

Aus der Klinik für Thorax- und Kardiovaskularchirurgie  
am Herzzentrum Duisburg,  
Evangelisches und Johanniter Klinikum Niederrhein gGmbH  
Chefarzt: Professor Dr. A. Krian

**Eine softwarebasierte Methodik zur Wahrung der Konsistenz  
von DRG- und qualitätssicherungs-relevanten Daten am  
Beispiel des BQS-Moduls „Herzchirurgie“**

**Dissertation**

**zur Erlangung des Grades eines Doktors der Medizin  
Der Medizinischen Fakultät  
der Heinrich-Heine-Universität Düsseldorf**

**vorgelegt von**

Arnd Benninghoff

**2009**

Als Inauguraldissertation gedruckt mit Genehmigung der  
Medizinischen Fakultät der Heinrich-Heine-Universität Düsseldorf

gez.: Univ.-Prof. Dr. med. Joachim Windolf

Dekan

Referent: Prof. Dr. med Arno Krian

Korreferent: Univ.-Prof. Dr. med. Christian Ohmann





# Inhaltsverzeichnis

1. Einleitung und Fragestellung.....	11
1.1. Ziele.....	13
1.2. Gliederung.....	13
2. Material und Methoden.....	15
2.1. Analyse des „Datensatzes HCH“.....	15
2.2. ER-Modellierung mit dem „Oracle Designer“.....	15
2.3. Implementierung des Datenmodells via SQL.....	16
2.4. Erhebung der Daten.....	16
2.5. Formulierung der Abfragen und Definition von Sichten („Views“)......	16
2.6. Konkrete Nutzung des Datenmodells: Entwicklung der Beispielanwendung „Qualitätssicherungs-Assistent“.....	17
2.7. Hinweis zum Volumen des Anhangs.....	17
3. Ergebnisse.....	17
3.1. Analyse des Codebezugs (ICD, OPS301) der Felder des ab dem 01.01.2007 gültigen „Datensatzes Herzchirurgie“ der „Bundesgeschäftsstelle Qualitätssicherung gGmbH“ (BQS).....	17
3.1.1. Feld [Institutionskennzeichen].....	18
3.1.1.1. Feld [Betriebsstätten-Nummer].....	18
3.1.2. Feld [Fachabteilung].....	18
3.1.3. Feld [Identifikationsnummer des Patienten].....	18
3.1.4. Feld [Aufnahmedatum Krankenhaus].....	18
3.1.5. Feld [Geburtsdatum].....	18
3.1.6. Feld [Geschlecht].....	18
3.1.7. Feld [Körpergröße].....	19
3.1.8. Feld [Körpergewicht bei Aufnahme].....	19
3.1.9. Feld [klinischer Schweregrad der Herzinsuffizienz (NYHA-Klassifikation)].....	19
3.1.10. Feld [Angina Pectoris (nach CCS)].....	19
3.1.11. Feld [Infarkt(e)].....	19
3.1.12. Feld [kardiogener Schock / Dekompensation].....	19
3.1.13. Feld [Reanimation].....	19
3.1.14. Feld [Patient wird beatmet].....	20
3.1.15. Feld [Pulmonale Hypertonie].....	20
3.1.16. Feld [Herzrhythmus bei Aufnahme].....	20
3.1.17. Feld [Patient ist Schrittmacher-/Defi-Träger].....	20
3.1.18. Feld [Einstufung nach ASA-Klassifikation].....	20
3.1.19. Feld [LVEF].....	20
3.1.20. Feld [Koronarangiographiebefund].....	20
3.1.21. Feld [signifikante Hauptstammstenose].....	20
3.1.22. Feld [PCI].....	20
3.1.23. Feld [Vor-OP(s) an Herz/Aorta].....	21
3.1.24. Feld [akute Infektion(en)].....	21
3.1.25. Feld [Diabetes mellitus].....	21
3.1.26. Feld [arterielle Gefäßerkrankung].....	21
3.1.27. Feld [periphere AVK (Extremitäten)].....	21
3.1.28. Feld [Arteria Carotis].....	21
3.1.29. Feld [Aortenaneurysma].....	21

3.1.30. Feld [sonstige arterielle Gefäßerkrankung(en)].....	21
3.1.31. Feld [Lungenerkrankungen].....	21
3.1.32. Feld [neurologische Erkrankung(en)].....	21
3.1.33. Feld [Schweregrad der Behinderung].....	21
3.1.34. Feld [präoperative Nierenersatztherapie].....	22
3.1.35. Feld [Kreatininwert i.S. in mg/dl (präoperativ)].....	22
3.1.36. Feld [Kreatininwert i.S. in µmol/l (präoperativ)].....	22
3.1.37. Feld [OP-Protokoll-Nummer].....	22
3.1.38. Feld [OP-Datum].....	22
3.1.39. Feld [Koronarchirurgie].....	22
3.1.40. Feld [Aortenklappenchirurgie].....	22
3.1.41. Feld [sonstige OP].....	22
3.1.42. Feld [Operation (OPS)].....	22
3.1.43. Feld [Dringlichkeit].....	22
3.1.44. Feld [Nitrate (i.v.)].....	23
3.1.45. Feld [Troponin positiv].....	23
3.1.46. Feld [Inotrope (i.v.)].....	23
3.1.47. Feld [(präoperativ) mechanische Kreislaufunterstützung].....	23
3.1.48. Feld [Wundkontaminationsklassifikation (nach Definition der CDC)].....	23
3.1.49. Feld [OP-Vorgehen].....	23
3.1.50. Feld [Zugang].....	23
3.1.51. Feld [OP-Zeit (Schnitt-Nahtzeit)].....	23
3.1.52. Feld [Bypasszeit].....	23
3.1.53. Feld [Aortenabklemmzeit].....	23
3.1.54. Feld [Anzahl der Grafts].....	23
3.1.55. Feld [Art der Grafts - Vene].....	24
3.1.56. Feld [Art der Grafts – ITA links].....	24
3.1.57. Feld [Art der Grafts – ITA rechts].....	24
3.1.58. Feld [Art der Grafts – A. radialis].....	24
3.1.59. Feld [Art der Grafts – sonstige Grafts].....	24
3.1.60. Feld [zentrale Anastomose(n)].....	24
3.1.61. Feld [periphere Anastomose(n), arteriell].....	24
3.1.62. Feld [periphere Anastomose(n), venös].....	24
3.1.63. Feld [TEA].....	24
3.1.64. Feld [Versorgte Koronararterie(n) – LAD und / oder Äste].....	24
3.1.65. Feld [Versorgte Koronararterie(n) – RCA und / oder Äste].....	24
3.1.66. Feld [Versorgte Koronararterie(n) – RCX und / oder Äste].....	24
3.1.67. Feld [anderer Koronareingriff].....	25
3.1.68. Feld [Aortenklappenchirurgie - Stenose].....	25
3.1.69. Feld [Aortenklappenchirurgie - Insuffizienz].....	25
3.1.70. Feld [Aortenklappenchirurgie - Klappeneingriff].....	25
3.1.71. Feld [Aortenklappenchirurgie - Ventil- bzw. Ringtyp].....	25
3.1.72. Feld [Aortenklappenchirurgie - Durchmesser].....	25
3.1.73. Feld [Dauer Intensivaufenthalt].....	25
3.1.74. Feld [Mobilisation].....	26
3.1.75. Feld [Mediastinitis].....	26
3.1.76. Feld [Reanimation].....	26
3.1.77. Feld [Myokardinfarkt].....	26
3.1.78. Feld [Low Cardiac Output].....	26
3.1.79. Feld [Rethorakotomie / Grund].....	27

3.1.80. Feld [respiratorische Insuffizienz].....	27
3.1.81. Feld [zerebrovaskuläres Ereignis bis zur Entlassung].....	27
3.1.82. Feld [Dauer des zerebrovaskulären Ereignisses].....	27
3.1.83. Feld [Schweregrad eines neurologischen Defizits bei Entlassung].....	27
3.1.84. Feld [Psychosyndrom].....	28
3.1.85. Feld [therapiepflichtige gastrointestinale Komplikationen].....	28
3.1.86. Feld [Kreatininwert i.S. in mg/dl (postoperativ)].....	28
3.1.87. Feld [Kreatininwert i.S. in µmol/l (postoperativ)].....	28
3.1.88. Feld [postoperative Nierenersatztherapie].....	28
3.1.89. Feld [Erythrozytenkonzentrat(e)].....	28
3.1.90. Feld [FFP].....	28
3.1.91. Feld [Thrombozytenkonzentrat].....	29
3.1.92. Feld [Herzrhythmus bei Entlassung].....	29
3.1.93. Feld [Patient trägt Schrittmacher / Defibrillator].....	29
3.1.94. Feld [Entlassungsdiagnose(n)].....	30
3.1.95. Feld [Entlassungsdatum Krankenhaus].....	30
3.1.96. Feld [Entlassungsgrund (§301-Vereinbarung)].....	30
3.1.97. Feld [Follow-Up - Erhebungsdatum].....	30
3.1.98. Feld [Status des Patienten am 30. postoperativen Tag].....	30
3.1.99. Feld [Todesdatum].....	30
3.1.100. Zusammenfassung der Analyse.....	30
3.1.100.1. Direkt ableitbare Felder.....	30
3.1.100.2. Wechselseitig anhand der Schlüssel (ICD / OPS) prüfbare Felder.....	31
3.2. Strukturen für den ICD- sowie den OPS-Katalog.....	32
3.2.1. Beschreibung der Entitäten und der Beziehungen zwischen diesen.....	32
3.2.1.1. Versionen: Tabellen [ICD_VERSION] und [OPS301_VERSION].....	32
3.2.1.2. Schlüssel: Tabellen [ICD] und [OPS301].....	33
3.2.1.3. Synonyme Begriffe: Tabellen [ICD_THESAURUS] und [OPS301_THESAURUS].....	33
3.2.1.4. Lokalisation: Tabellen [ICD_LOKALISATION] und [OPS301_LOKALISATION].....	33
3.2.1.5. Suchtexte: Tabellen [ICD_SUCHTEXT] und [OPS301_SUCHTEXT].....	33
3.2.2. Grafische Darstellung des Entitäts-Beziehungs-Modells (ER-Modell).....	34
3.2.3. Die aus dem Entitäts-Beziehungs-Modell resultierenden SQL-Anweisungen..	35
3.2.4. Die aus dem Entitäts-Beziehungs-Modell resultierenden konkreten Tabellen..	37
3.3. Strukturen für codeunabhängige Werte.....	39
3.4. Strukturen für die Falldaten.....	40
3.4.1. Basisdaten der Fälle.....	40
3.4.2. ICD- und OPS-Daten der Fälle.....	41
3.4.3. Codeunabhängige Daten der Fälle.....	43
3.4.5. ICD- , OPS- und codeunabhängige Daten der Operationen.....	43
3.5. Ableitung von Werten aus codierten Diagnosen und Leistungen.....	45
3.5.2. Ableitung von einfachen Werten.....	46
3.5.3. Ableitung von Kombinationswerten.....	48
3.5.4. Ableitung von Werten aus Komplexen.....	50
3.5.5. Kontext-Zuordnung.....	52
3.5.6. Phasen-Zuordnung.....	55
3.5.7. Abfragen zur Ermittlung abgeleiteter Werte.....	57
3.5.7.1. ICD-Schlüssel eines Falles – Sicht [Fall_ICD_INKL_OP].....	58
3.5.7.2. Einfache Diagnose-Werte - Sicht [FALL_KONTEX_DIAG_WERT_ICD].	58

3.5.7.3. Diagnose-Normal-Werte - Sicht [FALL_KONTEXT_DIAG_WERT_NON_ICD].....	58
3.5.7.4. Bestimmung der Zahl einfacher Diagnose-Werte je kombiniertem Wert - Sicht [DIAGNOSE_WERT_KOMB_ELEM_COUNT].....	58
3.5.7.5. Kombinierte Diagnose-Werte - Sicht [FALL_KONTEXT_DIAG_WERT_KOMB].....	59
3.5.7.6. Alle Diagnose-Werte - Sicht [FALL_KONTEXT_DIAGNOSE_RANG].....	59
3.5.7.7. Konkrete Werte der Diagnosen – Sicht [FALL_KONTEXT_DIAGNOSE]..	59
3.5.7.8. Ableitung von Prozedur-Werten.....	59
3.5.7.9. Ableitung von Komplex-Werten.....	60
3.5.7.9.1. Einfache Werte.....	60
3.5.7.9.2. Kombinierte Werte.....	60
3.5.7.9.3. Vereinigung der einfachen und kombinierten Werte.....	61
3.5.7.9.4. Zählung diagnostischer und prozeduraler Elemente.....	61
3.5.7.9.5. Zulässige Werte der Komplexe.....	61
3.5.7.9.6. Konkrete, schlüsselbasierte Werte der Komplexe.....	61
3.5.7.9.7. Werte der Komplexe.....	61
3.5.7.10. Abgeleitete Werte der Operationen.....	61
3.6. Plausibilitätsprüfung.....	63
3.6.1. Pufferung von Daten.....	63
3.6.2. Einfache Zuordnung von Schlüssel zu Diagnosen und Prozeduren.....	64
3.6.3. Elemente.....	65
3.6.4. Prüfung von Regeln.....	68
3.6.4.1. Existenz suspekter Elemente.....	69
3.6.4.2. Obligate Koppelung von Elementen.....	70
3.6.4.3. Gegenseitiger Ausschluss von Elementen.....	71
3.7. „Modell EJK“: Anwendung des Datenmodells.....	72
3.7.1. Ausgangssituation .....	72
3.7.2. Ziele .....	72
3.7.3. Lösung .....	73
3.7.3.1. Akquisition der Daten aus dem KIS.....	73
3.7.3.2. Übertragung der Rohdaten in die Strukturen des Datenmodells.....	74
3.7.3.3. Ableitung von Angaben aus den ICD- und OPS-Codes.....	75
3.7.3.4. Erzeugung und Anzeige des Prüfberichts.....	76
3.7.3.5. Übertragung des kompletten HCH-Datensatzes ins KIS.....	77
3.7.3.6. Pflege der Hinterlegungen.....	79
3.7.3.7. Menüsystem und Pflege der System-Konfiguration.....	81
4. Diskussion und Ausblick.....	83
4.1. Das konkrete System und dessen praktischer Einsatz.....	83
4.2. Flexibilität und mögliche Erweiterungen.....	84
4.3. Unabhängigkeit von spezifischen Datenbank-Systemen.....	84
4.4. Unabhängigkeit von spezifischen Krankenhaus-Informations-Systemen.....	85
4.4.1 Akquisition von Daten aus dem KIS.....	85
4.4.2. Rückübertragung von Daten an das KIS.....	85
4.5. Verwendung freier Werkzeuge und Komponenten.....	86
4.6. Ausblick auf die zukünftige Entwicklung .....	86
Abkürzungsverzeichnis.....	88
Literaturverzeichnis.....	89
Anhang 1 – Modul HCH der BQS (Musterbögen).....	93
Anhang 2 – Beispiele für zur Ableitung von Werten verwendbaren Codes.....	103



Anhang 3 – SQL92-Skript zur Implementierung des Datenmodells.....	106
Anhang 4 – Datenbank-Abfragen zu Ableitung von Daten aus ICD- und OPS-Schlüsseln	
.....	121
A4.1. View [Fall_ICD_Inkl_Op].....	121
A4.2. View [View Fall_Kontext_Diag_Wert_ICD].....	123
A4.3. View [Fall_Kontext_Diag_Wert_Non_ICD].....	128
A4.4. View [Diagnose_Wert_Komb_Elem_Count].....	129
A4.5. View [Fall_Kontext_Diag_Wert_Komb].....	130
A4.6. View [Fall_Kontext_Diagnose_Rang].....	132
A4.7. View [Fall_Kontext_Diagnose].....	134
A4.8. View [Fall_OPS_Inkl_OP].....	135
A4.9. View [Fall_Kontext_Proz_Wert_OPS].....	137
A4.10. View [Fall_Kontext_Proz_Wert_Non_OPS].....	142
A4.11. View [Prozedur_Wert_Komb_Elem_Count].....	143
A4.12. View [Fall_Kontext_Proz_Wert_Komb].....	144
A4.13. View [Fall_Kontext_Prozedur_Rang].....	146
A4.14. View [Fall_Kontext_Prozedur].....	148
A4.15. View [Fall_Kont_Komp_Diag_Wert_ICD].....	149
A4.16. View [Fall_Kont_Komp_Proz_Wert_OPS].....	154
A4.17. View [Fall_Kont_Komp_DWK_DW_ICD].....	159
A4.18. View [Fall_Kont_Komp_PWK_PW_OPS].....	164
A4.19. View [Fall_Kont_Komp_Diag_Wert_Komb].....	169
A4.20. View [Fall_Kont_Komp_Proz_Wert_Komb].....	171
A4.21. View [Fall_Kont_Komp_DW_PW_DWK_PWK].....	173
A4.22. View [Fall_KPW].....	176
A4.23. View [Fall_Kont_Komp_W_DW_PW_DWK_PWK].....	177
A4.24. View [Fall_Kont_Komp_Wert_D_P_Count].....	178
A4.25. View [Fall_Kontext_Komplex_Wert_Rang].....	179
A4.26. View [Fall_Kontext_Komplex_Min_Rang].....	180
A4.27. View [Fall_Kontext_Komplex].....	181
A4.28. View [Fall_ICD_OP].....	183
A4.29. View [Fall_Kontext_Diag_Wert_ICD_Op].....	184
A4.30. View [Fall_Kontext_Diag_W_Non_ICD_OP].....	189
A4.31. View [Fall_Kontext_Diag_Wert_Komb_OP].....	190
A4.32. View [Fall_Kontext_Diagnose_Rang_OP].....	192
A4.33. View [Fall_Kontext_Diagnose_OP].....	194
A4.34. View [Fall_OPS_OP].....	195
A4.35. View [Fall_Kontext_Proz_Wert_OPS_OP].....	196
A4.36. View [Fall_Kontext_Proz_W_Non_OPS_OP].....	201
A4.37. View [Fall_Kontext_Proz_Wert_Komb_OP].....	202
A4.38. View [Fall_Kontext_Prozedur_Rang_OP].....	204
A4.39. View [Fall_Kontext_Prozedur_OP].....	206
A4.40. View [Fall_Kont_Komp_D_Wert_ICD_OP].....	207
A4.41. View [Fall_Kont_Komp_P_Wert_OPS_OP].....	212
A4.42. View [Fall_Kont_Komp_DWK_DW_ICD_OP].....	217
A4.43. View [Fall_Kont_Komp_PWK_PW_OPS_OP].....	223
A4.44. View [Fall_Kont_Komp_D_Wert_Komb_OP].....	229
A4.45. View [Fall_Kont_Komp_P_Wert_Komb_OP].....	231
A4.46. View [Fall_KT_Komp_DW_PW_DWK_PWK_OP].....	233
A4.47. View [Fall_KPW_OP].....	236

A4.48. View [Fall_KT_KP_W_DW_PW_DWK_PWK_OP].....	237
A4.49. View [Fall_KT_KP_Wert_D_P_Count_OP].....	239
A4.50. View [Fall_Kontext_Komp_Wert_Rang_OP].....	240
A4.51. View [Fall_Kontext_Komp_Min_Rang_OP].....	241
A4.52. View [Fall_Kontext_Komplex_OP].....	242
Anhang 5 – Datenbank-Abfragen zur Plausibilitätsprüfung.....	244
A5.1. View [Puffer_Fall_Element].....	244
A5.2. View [Puffer_Fall_OP_Element].....	249
A5.3. View [Fall_Pruefung_Element].....	254
A5.4. View [Fall_OP_Pruefung_Element].....	259
A5.5. View [Fall_Pruefung_Koppelung].....	264
A5.6. View [Fall_OP_Pruefung_Koppelung].....	266
A5.7. View [Fall_Pruefung_Ausschluss].....	268
A5.8. View [Fall_OP_Pruefung_Ausschluss].....	269
Anhang 6: Exkurs – Design relationaler Datenbanken und Normalisierung.....	271
Anhang 7: JavaDoc der Klasse AgentDatabase (Auszug).....	288

# 1. Einleitung und Fragestellung

Eine bundesweit angelegte externe Qualitätssicherungsmaßnahme wird im Bereich der Herzchirurgie seit Dezember 1991 (Gründung der „Bundesarbeitsgemeinschaft Qualitätssicherung Herzchirurgie“) durchgeführt [1]; diese Dimension sowie die Existenz einer vertraglich fixierten formalen Grundlage stellten zu jener Zeit bemerkenswerte Neuerungen dar.

Der Tradition historischer Vorläufer folgend und vergleichbar mit ähnlichen Projekten anderer Fächer wurde und wird dabei das Ziel verfolgt, definierte Krankheitsbilder bzw. -Verläufe, die zugehörigen therapeutischen Regimes und schließlich deren Ergebnisse so detailliert wie möglich abzubilden, um damit eine klinikübergreifende Vergleichbarkeit auf der Basis wissenschaftlicher Auswertungen herzustellen. Die so gewonnenen Erkenntnisse dienen der Optimierung von Behandlungsstrategien und auf diese Weise letztlich der Verbesserung der Qualität.

Trotz zahlreicher Modifikationen hinsichtlich der Art und des Umfangs der erhobenen Daten, einer Reduktion des betrachteten Kollektivs sowie des Übergangs der Verantwortung für die Abwicklung des Verfahrens von der ursprünglich zuständigen, bei der Ärztekammer Nordrhein angesiedelten Projektgeschäftsstelle auf die „Bundesgeschäftsstelle Qualitätssicherung gGmbH“ (BQS [2]) zeigt sich Konstanz im Hinblick auf eine Eigenschaft der gesammelten Informationen: ein erheblicher Teil der Felder bezieht sich direkt oder indirekt auf Diagnosen und (so weit es um den intra- oder postoperativen Teil geht) Prozeduren bzw. Leistungen [3].

Besondere Bedeutung kommt diesem Punkt im Zusammenhang mit der im Jahr 2001 begonnenen Einführung eines pauschalierenden Entgeltsystems (G-DRG, [4]) zu. Die im ersten Absatz des § 17 b KHG geforderte Abbildung von Komplexitäten und Comorbiditäten bedeutet für die Dokumentation in den Kliniken, dass das Erzielen angemessener Erlöse nun in stärkerem Maße von einer vollständigen Verschlüsselung von Haupt- und Neben-Diagnosen sowie allen erbrachten Leistungen abhängig ist (Abbildung 1). Diese durchgreifende Änderung lässt sich als Adaption des fundamentalen Prinzips medizinischer Qualitätssicherung interpretieren: die Charakteristika und Determinanten einer Behandlung so präzise wie möglich darzustellen. Angesichts der gleichsinnigen Entwicklung war der Weg zur redundanten Sammlung von Angaben für Qualitätssicherung und Abrechnung förmlich vorgezeichnet.

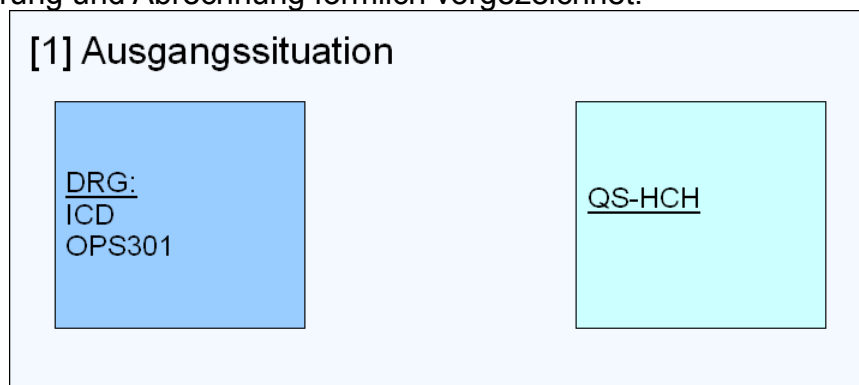
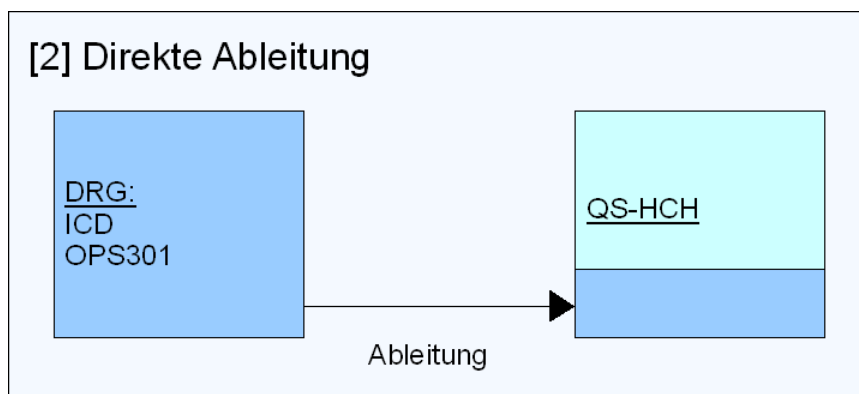


Abbildung 1: bisheriges Nebeneinander von DRG- und QS-relevanten Daten

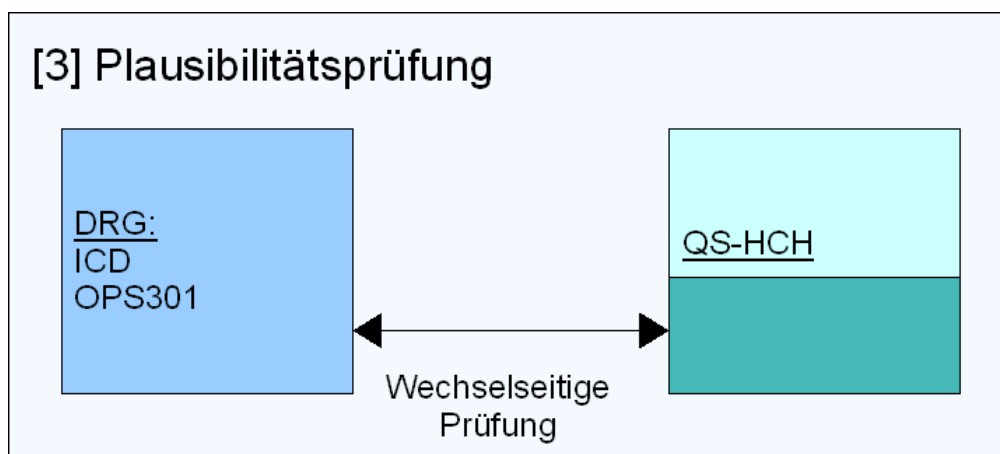
Für die Codierung von Diagnosen und Leistungen werden die durch das „Deutsche Institut für Medizinische Dokumentation und Information“ (DIMDI, [5]) veröffentlichten ICD- [6] und OPS301-Kataloge [7] in der jeweils gültigen, derzeit jährlich aktualisierten Fassung verwendet. Durch die „Deutschen Kodierrichtlinien“ wird definiert, unter welchen Bedingungen Schlüssel aus den beiden Verzeichnissen für Abrechnungszwecke angegeben werden können, um die Erkrankungsschwere eines konkreten Falles bzw. den dadurch verursachten Aufwand korrekt abzubilden [8].

Die für die Dokumentation verantwortlichen ärztlichen Mitarbeiter sehen sich durch diese beiden Entwicklungen mit einer Situation konfrontiert, die für einen Teil der Pflichtangaben eine Doppelerfassung gleicher Informationen erfordert – nämlich in denjenigen Fällen, in denen sich eine Antwort auf eine Frage des „Datensatzes Herzchirurgie“ nach dem Vorliegen einer bestimmten Diagnose oder nach der Durchführung einer bestimmten Leistung (oder nach Kombinationen aus beiden) direkt aus der Analyse der für die Abrechnung erfassten ICD- und OPS301-Schlüssel erschließt (Abbildung 2).



**Abbildung 2: Ableitung von QS-Angaben aus den DRG-relevanten Daten**

Bei einem weiteren Teil der Felder des „Datensatzes Herzchirurgie“ lässt sich – wenn schon keine direkte Ableitung des jeweiligen Wertes möglich ist – zumindest eine Prüfung auf wechselseitige Plausibilität der abrechnungs- und QS-relevanten Angaben vornehmen. Fehler in einem der beiden Bereiche ließen sich so ggf. leichter entdecken und korrigieren (Abbildung 3).



**Abbildung 3: gegenseitige Prüfung der DRG- und QS-relevanten Daten**

Mit Hilfe der beiden zuvor geschilderten Prinzipien (direkte Ableitung von QS-Angaben aus den codierten ICD- und OPS301-Schlüsseln; wechselseitige Prüfung der Plausibilität) lässt

sich der verständliche Wunsch nach Entlastung vom durch die Pflicht zur Dokumentation entstehenden Aufwand bei Steigerung der Qualität eben dieser Dokumentation realisieren; sie waren bereits Gegenstand einer früheren, jedoch auf den präoperativen Teil des „Datensatzes Herzchirurgie“ sowie das Abteilungsinformationssystem „ASKTHIS“ [9] beschränkten Arbeit [10] des Autors.

## **1.1. Ziele**

Aufbauend auf diesen Erkenntnissen werden mit der vorliegenden Arbeit folgende Ziele verfolgt:

- Schaffung eines Datenmodells, das die direkte Ableitung von Angaben des kompletten „Datensatzes Herzchirurgie“ aus den erfassten ICD- und OPS301-Schlüsseln eines Falles ebenso erlaubt wie eine wechselseitige Plausibilitätsprüfung zwischen den für die Abrechnung bzw. die Qualitätssicherung relevanten Daten; so soll eine integrative Sicht auf zwei Bereiche (QS- und DRG-relevante Dokumentation) ermöglicht werden, die in den auf dem deutschen Markt erhältlichen Krankenhausinformationssystemen bisher noch einer strikten Trennung unterliegen.
- Identifikation und Berücksichtigung generell anwendbarer Prinzipien bei der Gestaltung von Teilmodellen, die eine spätere Nutzung des Datenmodells auch im Rahmen von Qualitätssicherungsmaßnahmen anderer Fachdisziplinen ermöglichen.
- Realisation einer von existierenden Krankenhaus- oder Abteilungs-Informationssystemen und damit auch von speziellen Datenbanken unabhängigen, d.h. breit einsetzbaren Lösung, die eine Integration mit anderen Systemen durch Berücksichtigung offener Standards ermöglicht (Abstraktion des Modells von einer bestimmten bzw. konkreten/speziellen Anwendung). Diesem Ziel dient eine Beschränkung der aus dem Bestand des KIS zu gewinnenden Daten auf diejenigen Informationen, die Gegenstand der Vereinbarungen zum Datenaustausch gemäß § 301 SGB V, somit also in jedem KIS vorzuhaltende „Pflichtangaben“ sind [11].
- Beachtung bzw. Umsetzung von Design-Richtlinien, die eine Reduktion des für die Wartung der Datenbestandes (beispielsweise im Zuge der jährlichen Überarbeitung des ICD- bzw. des OPS301-Kataloges) erforderlichen Aufwandes auf das notwendige Minimum ermöglichen.
- Implementierung einer in das Krankenhausinformationssystem „ORBIS“ des Herstellers Agfa HealthCare [12] integrierten Anwendung des Datenmodells, die das mit der Dokumentation betraute Personal entlastet und auf diese Weise die Tauglichkeit des gefundenen Ansatzes demonstriert.
- Verwendung freier Werkzeuge („open source“), wo immer dies möglich und sinnvoll ist, um Interessenten ggf. eine kostengünstige Weiterentwicklung des Systems zu ermöglichen.

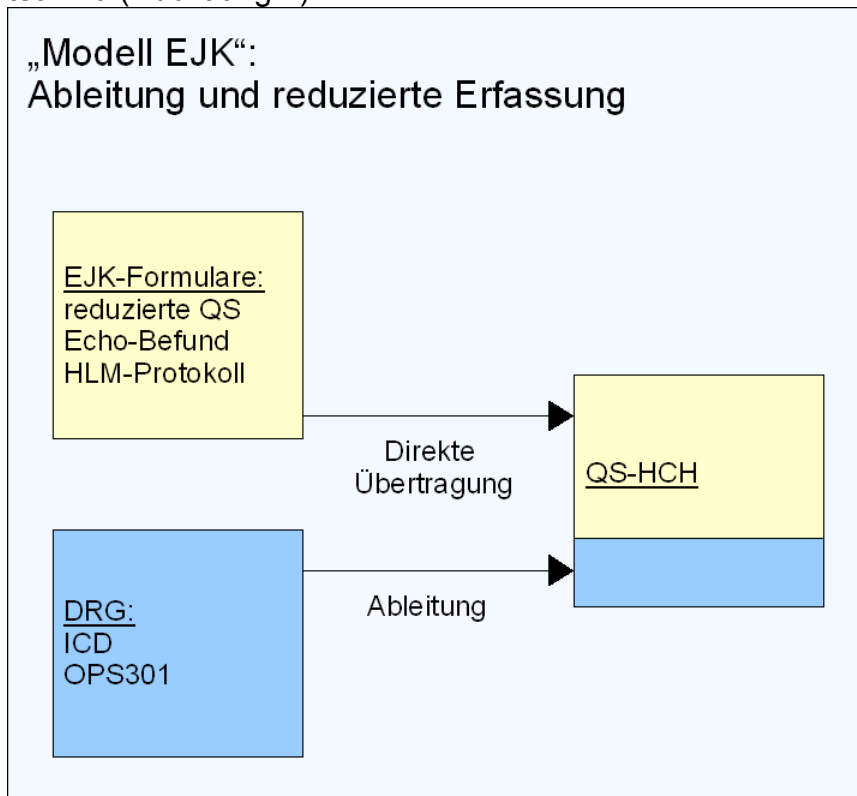
## **1.2. Gliederung**

Im ersten Teil (3.1.) der Arbeit werden die Elemente aller Teile des „Datensatzes Herzchirurgie“ einer Analyse im Hinblick auf eine potentielle Ableitbarkeit von Inhalten aus den codierten Diagnosen und Leistungen unterzogen; wo dies nicht möglich ist, wird sekundär die Möglichkeit zur Etablierung einer wechselseitigen Plausibilitätsprüfung evaluiert.

Darauf folgt im zweiten Teil die Beschreibung des Datenmodells, das den beiden

Einsatzzielen (direkte Herleitung von Angaben bzw. Prüfung der Plausibilität von Angaben) gerecht wird (3.2. - 3.6.).

Schließlich wird im dritten Teil (3.7.) die Nutzung des Datenmodells innerhalb einer in das KIS „ORBIS“ integrierten Applikation und die hierdurch zu erzielende Reduktion des Aufwandes für die abrechnungs- und QS-relevante Dokumentation (bei gleichzeitiger Automatisierung und damit Verbesserung der Prüfung von Plausibilitätsregeln) demonstriert. Dies gelingt unter anderem durch den Einsatz speziell entwickelter elektronischer Formulare des KIS, innerhalb derer auf das Ausfüllen von Feldern, deren Inhalte sich direkt aus den erfassten ICD- oder OPS301-Schlüsseln eines Falles ableiten lassen, verzichtet wird (Abbildung 4).



**Abbildung 4: Ableitung und reduzierte Erfassung von QS-Daten im EJK**

Wegen ihrer Mächtigkeit bzw. ihrer weltweit / langjährig bewiesenen Tauglichkeit und nicht zuletzt angesichts des enormen Marktanteils relationaler Datenbanken [13] gerade als Fundament von Krankenhausinformationssystemen wurde entschieden, die Datenbasis nach den Prinzipien des „relationalen Modells“ sowie der Normalisierung zu entwerfen [14], [15].

Aus vergleichbaren Gründen und zusätzlich wegen ihrer Kompaktheit bzw. einfachen Lesbarkeit wurde für die grafische Darstellung des Datenmodells bzw. seiner Teile die sog. „Martin-Notation“ [16] gewählt.

Für den Entwurf von Datenmodellen werden heute üblicherweise Designwerkzeuge eingesetzt, um den dafür erforderlichen Aufwand möglichst gering zu halten. Aus der Fülle der am Markt erhältlichen, leistungsfähigen Produkte wurde der „Oracle Designer“ [17] gewählt, da diese Software bereits seit einer Dekade in der Klinik für Thorax- und Kardiovaskularchirurgie am Herzzentrum Duisburg eingesetzt wird und dementsprechend umfangreiche Erfahrungen existieren.

Für die Entwicklung der Applikation, die als Beispiel für die Nutzung des Datenmodells Gegenstand des 3. Teils dieser Arbeit ist, fiel die Wahl auf die Programmiersprache Java [18] [19]. Für diese Entscheidung sprachen neben ihrer mittlerweile weiten Verbreitung vor allem Eigenschaften wie Mächtigkeit, leichte Erlernbarkeit und kostenlose bzw. freie Verfügbarkeit von Laufzeitsystemen und Programmierwerkzeugen für alle relevanten Betriebssysteme (Plattformunabhängigkeit) [20]. Da auf umfangreiche, positive Erfahrungen mit dem Einsatz der „Netbeans“-Entwicklungsumgebung zurückgegriffen werden konnte, fiel die Entscheidung für die Verwendung dieser IDE („integrated development environment“) [21].

---

## **2. Material und Methoden**

### **2.1. Analyse des „Datensatzes HCH“**

Jedes Feld des Datensatzes Herzchirurgie wurde einer Analyse dahingehend unterzogen, ob eine direkte Ableitbarkeit aus ICD- bzw. OPS301-Schlüsseln oder zumindest eine auf diese Daten gestützte Plausibilitätsprüfung möglich ist.

Neben einer Beachtung der Ausfüllhinweise zum Datensatz [22] war hierzu für jedes diagnose- oder prozedurbezogenes Feld eine auf hinreichend differenzierte Schlüssel zielende Suche in den amtlichen Katalogen erforderlich.

Um das Kriterium der „Ableitbarkeit“ zu erfüllen, mussten die in Frage kommenden Schlüssel einerseits für die möglichen Werte des Feldes codieren, durften jedoch andererseits nicht so wenig spezifisch sein, dass auch über die eigentlich zulässigen Ausprägungen hinausgehende bzw. davon verschiedene Erkrankungen oder Leistungen im Falle einer Erfassung für den jeweils betrachteten Fall zu einem dann falschen Wert des Zielfeldes führen. Diese Problematik wird im Ergebnisteil für jedes Feld des Datensatzes genauer erläutert.

Die Durchführung von Plausibilitätsprüfungen stellt hingegen keine derart restriktiven Anforderungen an die Genauigkeit der Übereinstimmung von Feldinhalt und Katalogschlüssel, da das Ergebnis dieses Prozesses Meldungen an den Benutzer des Systems sind, die über mögliche Fehler im Bereich der Dokumentation eines Falles informieren und daher stets weiterer Interaktion des Anwenders bedürfen.

### **2.2. ER-Modellierung mit dem „Oracle Designer“**

Die Entwicklung des sowohl direkte Ableitung wie auch Plausibilitätsprüfung ermöglichenden Datenmodells erforderte die auf die Ergebnisse der Analyse des Datensatzes (2.1.) gestützte Identifikation der Entitäten sowie der Beziehungen zwischen diesen; zum Einsatz kam dabei das Modul „ER-Modeler“ des „Oracle Designer“.

Das Ergebnis dieses Schrittes stellt das ER-Modell des Systems dar, welches die Tabellen, ihre Eigenschaften (Typ und vorgesehener Inhalt der Felder, Identifikatoren bzw. Primärschlüssel) sowie die Art der Beziehungen zwischen den Tabellen beschreibt.

So weit sie sich auf zentrale Elemente des Datenmodells beziehen, wurden Abbildungen

mit „Entitäts-Beziehungs-Diagrammen“ in den Text der folgenden Abschnitte eingebettet.

### **2.3. Implementierung des Datenmodells via SQL**

Mit dem „Repository Object Navigator“ des „Oracle Designer“ wurden aus dem ER-Modell die konkreten Anweisungen an das relationale Datenbank-Managementsystem in der Sprache SQL („structured query language“) [23] erzeugt, die zur Implementierung des Datenbank-Schemas erforderlich sind.

Ergänzend zu den im ER-Modell definierten Eigenschaften der Tabellen bzw. der Beziehungen zwischen diesen wurden in diesem Schritt Regeln formuliert, denen die im Schema schließlich hinterlegten Daten unterliegen. So weit diese nicht als routinemäßige Maßnahmen, sondern als relevant und spezifisch für das Datenmodell anzusehen sind, wird später explizit Bezug auf sie genommen. Die während dieser Phase zur Steigerung der Performanz späterer Abfragen ebenfalls angelegten Indizes erfahren – angesichts ihres Charakters als Standardtechnik - im weiteren Text jedoch keine besondere Würdigung mehr.

Eine kurze Einführung in die Grundlagen des Designs relationaler Datenbanken sowie die Prinzipien der Normalisierung findet sich in Anhang 6 [24].

Der vollständige, in der Sprache SQL formulierte Quelltext, mit dessen Hilfe das entworfene Schema der Datenbank – bestehend aus allen Tabellen inklusive der durch so genannte „Constraints“ bzw. Regeln abgebildeten Beziehungen zwischen diesen – angelegt wurde, findet sich im Anhang 3.

### **2.4. Erhebung der Daten**

Nach der Implementierung des Schemas der Datenbank mussten die Einträge der Tabellen angelegt werden; insbesondere mussten die korrespondierenden Elemente der Felder des „Datensatzes Herzchirurgie“ aus dem ICD- bzw. OPS301-Katalog ermittelt und hinterlegt werden. In weiteren Schritten wurden die Beziehungen zwischen den einzelnen Daten bzw. Tabellen definiert und als Teil der Datenbasis eingefügt.

Wegen des großen Umfangs der Hinterlegungen werden im Text nur jeweils wenige Beispiele diskutiert; weitere Aufstellungen verwendbarer Schlüssel finden sich exemplarisch im Anhang 2.

### **2.5. Formulierung der Abfragen und Definition von Sichten („Views“)**

Abfragen zur Nutzung des Datenmodells im Zuge der QS- und DRG-relevanten Dokumentation von Behandlungsfällen im KIS wurden in SQL formuliert. Ein Hilfsmittel zur besseren Strukturierung insbesondere komplexer Auswertungen stellen dabei sog. „Sichten“ („Views“) dar. Dabei handelt es sich um permanent in der Datenbank verfügbare Abfragen, deren Ergebnismengen ihrerseits wie Tabellen (bspw. als Zwischenschritte innerhalb weiterer Abfragen) verwendet werden können.

Die Datenbank-Auswertungen, die letztlich der direkten Ableitung von QS-Daten aus den codierten Diagnosen und Prozeduren bzw. der wechselseitigen Prüfung der Plausibilität von Angaben dienen, wurden schrittweise aufgebaut, wobei die jeweiligen Ergebnismengen dieser Zwischenschritte als Sichten angelegt wurden.



Eine Aufstellung der für die Anlage der erforderlichen Sichten verwendeten SQL-Anweisungen findet sich in verschiedenen, im Text referenzierten Abschnitten des Anhangs.

## **2.6. Konkrete Nutzung des Datenmodells: Entwicklung der Beispielanwendung „Qualitätssicherungs-Assistent“**

Als konkrete Anwendung des Datenmodells wurde anschließend die Java-Anwendung namens „Qualitätssicherungs-Assistent“ entwickelt, die aus dem KIS exportierte Daten eines Behandlungsfalles (ICD- und OPS301-Schlüssel) mit Hilfe der im Datenmodell hinterlegten Angaben auswertet und so die direkt ableitbaren QS-Daten wie auch das Ergebnis der Plausibilitätsprüfungen (in Form einer Menge von Einzelmeldungen) gewinnt.

Die direkt abgeleiteten QS-Daten werden mit einer Fernsteuerungseinheit übertragen, die die Tätigkeit eines Datentypisten simuliert, indem alle für die (ursprünglich manuelle) Eingabe der abgeleiteten Informationen notwendigen Maus- und Tastatureingaben erzeugt und an den Teil des KIS gesendet werden, in dem die Dokumentation der QS-Angaben erfolgt. Auf diese Weise kann eine automatisierte Datenübertragung auch für diejenigen Systeme realisiert werden, deren Hersteller keine eigens dafür konzipierte Schnittstelle zur Verfügung stellen.

Da die Erzeugung der Steuerkommandos ihrerseits weitgehend über in Tabellen hinterlegte Daten determiniert wird, kann durch Manipulation der entsprechenden Einträge der „Qualitätssicherungs-Assistent“ mit geringem Aufwand und weitgehend ohne Änderung der Java-Quelltexte an andere Krankenhausinformationssysteme ebenso wie an weitere Qualitätssicherungsmaßnahmen (anderer Fachgebiete) adaptiert werden (3.7.).

## **2.7. Hinweis zum Volumen des Anhangs**

Einen wesentlichen Teil der Ergebnisse (2.2., 2.3., 2.5.) dieser Arbeit stellen das Datenmodell sowie die auf diesem operierenden Abfragen (gerade auch als Grundlage von Sichten!) dar. Die SQL-Anweisungen, mit deren Hilfe sowohl das Datenmodell auf einem RDBMS mit entsprechenden Eigenschaften implementiert als auch notwendige Auswertungen vorgenommen werden können, sind komplex und – damit einhergehend – äußerst umfangreich. Da ihre Integration in den eigentlichen Text dessen Volumen weit über ein noch vertretbares Maß hinaus ausweiten würde, erfolgte eine Verlagerung der Skripte in den Anhang - hieraus wiederum erklärt sich sein großer Umfang.

# **3. Ergebnisse**

## **3.1. Analyse des Codebezugs (ICD, OPS301) der Felder des ab dem 01.01.2007 gültigen „Datensatzes Herzchirurgie“ der „Bundesgeschäftsstelle Qualitätssicherung gGmbH“ (BQS)**

Die für den „Datensatz Herzchirurgie“ zu erfassenden Angaben werden durch die Bögen des Moduls HCH (Anhang 1) und die diesen zugeordneten Ausfüllhinweise definiert.

Die einzelnen Felder wurden im Hinblick auf die Möglichkeit einer direkten Ableitung ihrer Inhalte aus ICD- oder OPS-codierten Diagnosen bzw. Leistungen oder alternativ auf eine

wechselseitige Prüfbarkeit der Plausibilität hin analysiert.

Beide Verfahren (Ableitung, Plausibilitätsprüfung) werden in nachfolgenden Abschnitten ausführlich vor- und exemplarisch dargestellt; ferner finden sich im Anhang komplette Listen der geeigneten Katalogeinträge (Anhang 2: eine nach den Feldern des Datensatzes HCH aufgeschlüsselte Aufstellung der ICD- und OPS-Katalogeinträge, die sich im Zuge der Analyse als verwendbar für die beiden skizzierten Zwecke erwiesen) sowie der für die Durchführung von Plausibilitätsprüfungen verwendeten Daten(strukturen). Soweit also für die Angaben des Moduls HCH Potential zur Anwendung einer der beiden Methodiken besteht, wird das in den Passagen dieses Kapitels nur in jeweils knapper Form erwähnt.

Für die Ableitung von Werten aus den Diagnose- oder Leistungsdaten ist neben der Frage, ob überhaupt ein ICD- oder OPS-Katalogeintrag existiert, von ausschlaggebender Bedeutung, dass der in Betracht kommende Schlüssel hinreichend spezifisch ist, also ausschließlich für die Erkrankung bzw. die Maßnahme codiert, die eine mögliche Ausprägung des jeweils analysierten Feldes darstellt - und nicht zusätzlich noch für Krankheitsentitäten oder Prozeduren, die eben nicht durch die jeweilige Option abgebildet werden. Nicht hinreichend spezifische Katalogeinträge bieten sich – bei fehlender Möglichkeit zur direkten Ableitung von Werten – als Gegenstand von Plausibilitätsprüfungen an.

Eine zusätzlich zu berücksichtigende Dimension dieses Problems stellen Fälle dar, in denen die zur Auswahl stehenden möglichen Werte eines QS-Feldes ihrerseits Kombinationen aus mehreren Diagnosen (bzw. Prozeduren) oder gar von Diagnosen und Prozeduren darstellen. Diese „Spezialfälle“ werden in den später folgenden Abschnitten thematisiert, die der Herleitung von Daten gewidmet sind. Der letztgenannte Punkt (Kombinationen aus Diagnosen und Leistungen) wird als im weiteren Kontext dieser Arbeit „Komplex“ bezeichnetes Konstrukt behandelt und später dargestellt.

Die Ergebnisse dieser Analyse werden nachfolgend dargestellt, wobei die Gliederung der Unterpunkte sich am Index der jeweils behandelten Felder innerhalb des Datensatzes orientiert (Musterbögen der BQS für das Modul HCH: Anhang 1, Abbildungen A-1.1. – A-1.10.).

### **3.1.1. Feld [Institutionskennzeichen]**

#### **3.1.1.1. Feld [Betriebsstätten-Nummer]**

### **3.1.2. Feld [Fachabteilung]**

### **3.1.3. Feld [Identifikationsnummer des Patienten]**

### **3.1.4. Feld [Aufnahmedatum Krankenhaus]**

### **3.1.5. Feld [Geburtsdatum]**

### **3.1.6. Feld [Geschlecht]**

Die Inhalte der vorgenannten Felder beziehen sich auf administrative Daten zum stationären Aufenthalt; es existiert keinerlei Bezug zu ICD- bzw. OPS-codierten Diagnosen oder Leistungen.

### **3.1.7. Feld [Körpergröße]**

### **3.1.8. Feld [Körpergewicht bei Aufnahme]**

Hierbei handelt es sich um Messwerte ohne direkten Bezug zu Einträgen des ICD- oder OPS-Kataloges. Zwar bestünde die Möglichkeit, aus diesen Angaben den sog. „body-mass-index“ (BMI) zu berechnen und bei Überschreitung zu definierender Grenzwerte eine Prüfung auf die Existenz eines für die Erkrankung „Adipositas“ codierenden Schlüssels durchzuführenden – angesichts der fehlenden Relevanz (sei es durch Auswirkungen auf die Abrechnung des Falles oder durch Reduktion des Aufwandes für die Dokumentation) wurde jedoch darauf verzichtet.

### **3.1.9. Feld [klinischer Schweregrad der Herzinsuffizienz (NYHA-Klassifikation)]**

Dieses Feld bezieht sich auf eine Stadieneinteilung einer ggf. vorhandenen Herzinsuffizienz auf Grundlage der NYHA-Klassifikation [25].

Für die Linksherzinsuffizienz codieren hinreichend differenzierte, den 4 Kategorien der Klassifikation entsprechende Schlüssel (I50.11, I50.12, I50.13, I50.14).

Anders verhält es sich hingegen für den Fall einer Rechtsherz- oder Globalinsuffizienz.

Da eine direkte Ableitung der für dieses Feld anzugebenden Option somit nicht durchführbar ist, besteht die alternative Möglichkeit, die wechselseitige Plausibilität zwischen dem Wert des Feldes und der (fehlenden) Existenz entsprechender Schlüssel in den Falldaten zu prüfen.

### **3.1.10. Feld [Angina Pectoris (nach CCS)]**

In diesem Feld soll eine eventuell vorhandene Angina pectoris klassifiziert werden, wobei die Einteilung [26] der Canadian Cardiovascular Society (CCS) [27] zur Anwendung kommt.

Es existieren keine ICD-Schlüssel, die den Definitionen der CCS entsprechen – jedoch besteht die Möglichkeit, die wechselseitige Plausibilität der Angaben nach folgendem Schema zu prüfen: liegt eine Angina pectoris – also eine vom Grad 0 („nein“) abweichende Symptomatik – vor, muss einer der dafür codierenden ICD-Schlüssel Element der Falldaten sein.

### **3.1.11. Feld [Infarkt(e)]**

### **3.1.12. Feld [kardiogener Schock / Dekompensation]**

### **3.1.13. Feld [Reanimation]**

Im ICD-Katalog existieren keine hinreichend differenzierten Schlüssel, die eine Abgrenzung der in diesen Feldern zur berücksichtigenden verstrichenen Intervalle zulassen; so weit die Definition von Zeitspannen Bestandteil der vorhandenen Codes sind, unterscheiden sie sich deutlich von denjenigen der im Modul HCH vorgegebenen Ausprägungen.

Mit dem Repertoire von entsprechenden Schlüsseln des ICD-Kataloges lässt sich jedoch eine Prüfung der wechselseitigen Plausibilität durchführen.

### **3.1.14. Feld [Patient wird beatmet]**

Zwar existieren 3 für die verschiedenen Formen der Entität „respiratorische Insuffizienz“ codierende ICD-Schlüssel (J96.0, J96.1, J96.9), jedoch sind auch Fälle denkbar, in denen ein Patient bzw. eine Patientin bei bestehender respiratorischer Insuffizienz (noch) nicht beatmet wird; der Code „Langzeitige Abhängigkeit vom Respirator“ (Z99.1) bezeichnet eine langfristige Abhängigkeit von mindestens 3 Monaten Dauer.

Daraus ergibt sich, dass der Inhalt von Feld 14 nicht implizit auf Basis der erfassten ICD-Schlüssel eines Falles ermittelt werden kann. Es bietet sich jedoch an, mittels der erwähnten Katalogeinträge die Plausibilität der Angaben zu prüfen.

### **3.1.15. Feld [Pulmonale Hypertonie]**

Im ICD-Katalog existieren Einträge für die primäre sowie die sekundäre Form – eine Ableitung der Angabe aus den für einen Fall erfassten ICD-Schlüsseln ist also möglich.

### **3.1.16. Feld [Herzrhythmus bei Aufnahme]**

Sowohl für die mögliche Ausprägung „Vorhofflimmern“ wie auch für die Option „andere Rhythmusstörungen“ existieren entsprechende Schlüssel im ICD-Katalog; damit kann der Inhalt dieses Feldes direkt – basierend auf den ICD-Schlüsseln eines Falles – ermittelt werden.

### **3.1.17. Feld [Patient ist Schrittmacher-/Defi-Träger]**

Der einzige in Betracht kommende ICD-Schlüssel Z95.0 „Vorhandensein eines implantierten Herzschrittmachers oder eines implantierten Kardiodéfibrillators“ lässt die hier geforderte Unterscheidung bezüglich der beiden Typen von Aggregaten nicht zu, kann jedoch im Rahmen einer durchzuführenden Plausibilitätsprüfung berücksichtigt werden.

### **3.1.18. Feld [Einstufung nach ASA-Klassifikation]**

Es existieren keine ICD-Schlüssel, die sich auf die ASA-Klassifikation beziehen; folglich ist weder eine ICD-gestützte Ableitung des Feld-Inhaltes noch eine Prüfung der Plausibilität möglich.

### **3.1.19. Feld [LVEF]**

Da für die hier geforderte Kategorisierung der linksventrikulären Funktion keine adäquaten ICD-Schlüssel existieren, sind weder Ableitung noch Plausibilitätsprüfung möglich.

### **3.1.20. Feld [Koronarangiographiebefund]**

Es existieren hinreichend differenzierte ICD-Schlüssel (I25.11, I25.12, I25.13), die eine direkte Ableitung des Inhaltes dieses Feldes ermöglichen.

### **3.1.21. Feld [signifikante Hauptstammstenose]**

Ist der Code I25.14 Element der für einen Fall erfassten Menge von ICD-Schlüsseln, lässt sich der Inhalt dieses Feldes direkt ableiten; das setzt allerdings voraus, dass eine Codiervorschrift etabliert wird, die einen Stenosegrad von 50% oder mehr als Mindestvoraussetzung für die Verwendung dieses Katalogeintrags definiert.

### **3.1.22. Feld [PCI]**

Der Wert dieses sich auf in der Vergangenheit durchgeführte Maßnahmen beziehenden Feldes kann nicht direkt aus den codierten ICD-Schlüsseln abgeleitet werden.

Es besteht jedoch die Möglichkeit, eine Plausibilitätsprüfung durchzuführen, die die

Existenz von interventionell applizierten koronaren Implantaten beschreibende bzw. einschließende Codes berücksichtigt.

### **3.1.23. Feld [Vor-OP(s) an Herz/Aorta]**

Es existieren keine Einträge im ICD-Katalog, die Rückschlüsse auf die hier geforderte genaue Anzahl der in der Vergangenheit durchgeführten Eingriffe am Herzen bzw. an der Aorta zulassen – jedoch besteht die Möglichkeit, eine Plausibilitätsprüfung durchzuführen, die sich auf post-operative Zustände beschreibende Schlüssel erstreckt.

### **3.1.24. Feld [akute Infektion(en)]**

Mit Ausnahme der beiden Optionen „Wundinfektion Thorax [7]“ und „Wundinfektion untere Extremität [11]“, für die keine hinreichend differenzierten Einträge innerhalb des ICD-Kataloges existieren, können alle Elemente des hier anzuwendenden „Schlüssels 2“ (eine Abbildung des BQS-Muster-Dokumentationsbogens inklusive Schlüsselerzeichnis findet sich in Anhang 1) direkt aus den ICD-codierten Daten eines Falles abgeleitet werden.

### **3.1.25. Feld [Diabetes mellitus]**

Da hier zusätzliche Aussagen zur ggf. bereits eingeleiteten Therapie der Erkrankung erwartet werden, ist eine Herleitung aus den erfassten ICD-Schlüsseln nicht möglich; jedoch besteht die Möglichkeit, die diesbezügliche wechselseitige Plausibilität der Angaben zu prüfen.

### **3.1.26. Feld [arterielle Gefäßerkrankung]**

### **3.1.27. Feld [periphere AVK (Extremitäten)]**

### **3.1.28. Feld [Arteria Carotis]**

### **3.1.29. Feld [Aortenaneurysma]**

### **3.1.30. Feld [sonstige arterielle Gefäßerkrankung(en)]**

Es existieren hinreichend differenzierte ICD-Schlüssel, die eine direkte Ableitung der Inhalte dieser Felder ermöglichen.

### **3.1.31. Feld [Lungenerkrankungen]**

Eine Herleitung aus den für den Fall erfassten, ICD-codierten Aufnahme-Diagnosen ist nur für die Ausprägungen „nein“ und „ja, andere Lungenerkrankungen möglich“; im Fall der COPD werden zusätzlich Angaben zur bisher erfolgten Therapie erwartet, die sich aus den in Frage kommenden ICD-Schlüsseln nicht erschließen lassen. Eine Plausibilitätsprüfung, die das Vorliegen einer COPD zum Gegenstand hat, ist allerdings möglich.

### **3.1.32. Feld [neurologische Erkrankung(en)]**

Die möglichen Ausprägungen für dieses Feld lassen sich aus den erfassten ICD-Schlüsseln eines Falles ableiten. Eine beachtenswerte Besonderheit stellt dabei die Option „ja, Kombination“ dar: Kombinationen aus Diagnosen bedürfen einer besonderen Berücksichtigung im Datenmodell, die später erläutert wird.

### **3.1.33. Feld [Schweregrad der Behinderung]**

Die möglichen Inhalte dieses Feldes, die sich auf die modifizierte Rankin-Skala zur Qualifizierung des neurologischen Defizits nach Schlaganfall [28] beziehen, lassen sich nicht direkt aus den ICD-codierten Aufnahmediagnosen eines Falles ableiten; es besteht jedoch die Möglichkeit, eine auf die Folgezustände von Schlaganfällen beschreibenden

Schlüssel gestützte Plausibilitätsprüfung durchzuführen.

### **3.1.34. Feld [präoperative Nierenersatztherapie]**

Die möglichen Inhalte dieses Feldes lassen sich aus den ICD-Schlüsseln eines Falles ableiten, sofern für die Ausprägung „akut“ einer der beiden Schlüssel Z49.1 („Extrakorporale Dialyse“) bzw. Z49.2 („sonstige Dialyse“), für die Option „chronisch“ hingegen Z99.2 („langzeitige Abhängigkeit von der Dialyse bei Niereninsuffizienz“) zur Anwendung kommt.

### **3.1.35. Feld [Kreatininwert i.S. in mg/dl (präoperativ)]**

### **3.1.36. Feld [Kreatininwert i.S. in µmol/l (präoperativ)]**

Diese Werte lassen sich nicht direkt aus Einträgen des ICD-Katalogs ableiten; es besteht aber die Möglichkeit zur Prüfung der Plausibilität, indem die Überschreitung von Grenzwerten mit der Existenz eines für eine Störung der Nierenfunktion codierenden Schlüssels verglichen wird.

---

*Hier endet der „präoperative Teil“ des Datensatzes Herzchirurgie; die folgenden Felder beziehen sich auf „intraoperative“ Angaben, die separat für jeden durchgeführten Eingriff zu dokumentieren sind.*

---

### **3.1.37. Feld [OP-Protokoll-Nummer]**

### **3.1.38. Feld [OP-Datum]**

Bei diesen beiden Feldern handelt es sich um administrative Angaben ohne Bezug zu Diagnose- oder Leistungsschlüsseln.

### **3.1.39. Feld [Koronarchirurgie]**

### **3.1.40. Feld [Aortenklappenchirurgie]**

### **3.1.41. Feld [sonstige OP]**

Die Inhalte dieser 3 Felder lassen sich direkt aus den für die betreffende Operation angegebenen OPS-Schlüsseln ableiten.

### **3.1.42. Feld [Operation (OPS)]**

In diesem Feld werden die für eine Operation erfassten OPS-Schlüssel übermittelt; eine auf die Menge der Codes eines Eingriffs gestützte Ableitbarkeit ist implizit gegeben.

### **3.1.43. Feld [Dringlichkeit]**

Der Inhalt dieses Feldes lässt sich weder aus Einträgen des ICD-Kataloges noch aus denen des OPS-Kataloges direkt ableiten.

Eine Plausibilitätsprüfung begrenzter Aussagekraft lässt sich auf Basis der Annahme realisieren, dass bestimmte Maßnahmen (bspw. Reanimation, IABP-Implantation) mit einer dringlichen bzw. Notfall-Indikation assoziiert sind. Dabei muss jedoch berücksichtigt werden, dass nicht zwangsläufig nur vorbestehende, sondern auch erst im Verlauf des Eingriffs auftretende Symptome Ziel einer therapeutischen Prozedur werden können.

### **3.1.44. Feld [Nitrate (i.v.)]**

### **3.1.45. Feld [Troponin positiv]**

### **3.1.46. Feld [Inotrope (i.v.)]**

Die Inhalte dieser Felder lassen sich mangels hinreichend differenzierter ICD- oder OPS-Schlüssel nicht direkt ableiten. Prinzipiell ist es jedoch möglich, eine auf die Existenz einer ICD-codierten Indikation (Nitrate i.v., Inotrope i.v.) oder einer mit positivem Troponin-Test einhergehenden Erkrankung zielende Plausibilitätsprüfung durchzuführen.

### **3.1.47. Feld [(präoperativ) mechanische Kreislaufunterstützung]**

Die zutreffende Option in diesem Feld kann nicht direkt aus den Schlüsseln der Operation bzw. des Falles abgeleitet werden – insbesondere die ICD-Codes, die entsprechende Abhängigkeiten beschreiben (bspw. Z99.8 „Langzeitige Abhängigkeit von sonstigen unterstützenden Apparaten, medizinischen Geräten oder Hilfsmitteln“), sind nicht spezifisch genug für diesen Zweck.

Eine Auswertung präoperativ durchgeführter Maßnahmen im Hinblick auf solche Prozeduren, die die Anwendung kreislaufunterstützender Systeme beschreiben, liefert in den Fällen falsche Resultate, in denen die Maßnahme vor Auf-/Übernahme in die herzchirurgische Fachabteilung durchgeführt wurde.

### **3.1.48. Feld [Wundkontaminationsklassifikation (nach Definition der CDC)]**

Es existieren keine hinreichend differenzierte Schlüssel im ICD-Katalog, die eine direkte Ableitung des Feld-Inhaltes oder die Durchführung einer Plausibilitätsprüfung zulassen.

### **3.1.49. Feld [OP-Vorgehen]**

Da keine Einträge im OPS-Katalog existieren, die für die Option „Umstieg von OP ohne HLM auf OP mit HLM“ codieren, kann lediglich eine Plausibilitätsprüfung mit Hilfe der Schlüssel durchgeführt werden, die Aussagen zum (fehlenden) Einsatz der Herz-Lungen-Maschine implizieren und so einen Abgleich mit den beiden anderen Alternativen („mit HLM“/„ohne HLM“) zulassen.

### **3.1.50. Feld [Zugang]**

Die für dieses Feld definierten Alternativen lassen sich aus den zur Operation erfassten OPS-Schlüsseln ableiten.

### **3.1.51. Feld [OP-Zeit (Schnitt-Nahtzeit)]**

### **3.1.52. Feld [Bypasszeit]**

### **3.1.53. Feld [Aortenabklemmzeit]**

Die Inhalte dieser Felder lassen sich weder aus Schlüsseln ableiten noch besteht die Möglichkeit zur Durchführung einer Plausibilitätsprüfung.

### **3.1.54. Feld [Anzahl der Grafts]**

Eine Ableitung dieser Angabe mittels der für die Operation erfassten OPS-Schlüssel ist nicht möglich, jedoch könnte eine auf die Frage „Ist die Anzahl der Grafts von 0 verschieden?“ zielende Plausibilitätsprüfung durchgeführt werden.

### **3.1.55. Feld [Art der Grafts - Vene]**

Der Inhalt dieses Feldes lässt sich auf Basis der OPS-Codes der Operation direkt ermitteln.

### **3.1.56. Feld [Art der Grafts – ITA links]**

### **3.1.57. Feld [Art der Grafts – ITA rechts]**

### **3.1.58. Feld [Art der Grafts – A. radialis]**

Es existieren keine Einträge im OPS-Katalog, die hinreichend genau nach den für diese Felder relevanten Typen von arteriellen Grafts differenzieren; eine direkte Ableitung der Feld-Inhalte ist daher nicht möglich, die Prüfung der Plausibilität hingegen schon.

### **3.1.59. Feld [Art der Grafts – sonstige Grafts]**

Der Wert dieses Feldes lässt sich direkt aus der Menge der für die Operation erfassten OPS-Schlüssel ableiten.

### **3.1.60. Feld [zentrale Anastomose(n)]**

### **3.1.61. Feld [periphere Anastomose(n), arteriell]**

### **3.1.62. Feld [periphere Anastomose(n), venös]**

Diese Angaben lassen sich nicht aus der Menge der für eine Operation dokumentierten OPS-Schlüssel ableiten.

Analog zu den unter 3.1.54. bzw. 3.1.56. - 3.1.58. geschilderten Konstellationen bestünde die Möglichkeit, eine auf die Frage „Wurden Bypässe angelegt bzw. Grafts verwendet?“ konzentrierte Plausibilitätsprüfung durchzuführen. Dadurch könnte jedoch praktisch keine zusätzliche „Sicherheit“ im Hinblick auf das Ziel einer möglichst fehler- bzw. widerspruchsfreien Dokumentation gewonnen werden, da identische Sachverhalte berücksichtigt würden.

### **3.1.63. Feld [TEA]**

Diese Angabe lässt sich aus den für die Operation angegebenen OPS-Schlüsseln direkt ableiten; hierzu muss auch die zum Code erfasste Seiten-Lokalisation berücksichtigt werden (das Verfahren zur Auswertung dieses Merkmals wird in später folgenden Abschnitten beschrieben).

### **3.1.64. Feld [Versorgte Koronararterie(n) – LAD und / oder Äste]**

### **3.1.65. Feld [Versorgte Koronararterie(n) – RCA und / oder Äste]**

### **3.1.66. Feld [Versorgte Koronararterie(n) – RCX und / oder Äste]**

Es besteht keine Möglichkeit, die in diesen Feldern geforderten Angaben aus den angegebenen OPS-Schlüsseln einer Operation abzuleiten, da im Prozeduren-Katalog die Versorgung bestimmter Koronararterien bzw. ihrer Äste nicht differenziert berücksichtigt wird.

Hinsichtlich der möglichen, jedoch keinen zusätzlichen Nutzen stiftenden Durchführung einer Plausibilitätsprüfung gilt die bereits unter 3.1.62. getroffene Aussage.



### **3.1.67. Feld [anderer Koronareingriff]**

Der Wert dieses Feldes lässt sich direkt aus der Menge der für die Operation erfassten OPS-Schlüssel ableiten.

### **3.1.68. Feld [Aortenklappenchirurgie - Stenose]**

### **3.1.69. Feld [Aortenklappenchirurgie - Insuffizienz]**

Die Inhalte beider Felder lassen sich aus den ICD-Schlüsseln der Operation ableiten; dabei muss beachtet werden, dass jeweils auf einzelne Herzklappen bezogene Katalogeinträge verwendet werden, da mehrere Klappen einschließende Kombinationscodes keine Differenzierung zwischen Stenosen und / oder Insuffizienzen der verschiedenen Klappen zulassen.

### **3.1.70. Feld [Aortenklappenchirurgie - Klappeneingriff]**

Der Großteil der durch einen speziellen Katalog definierten möglichen Werte für dieses Feld lässt sich direkt aus den OPS-Codes der Operationen ableiten. Ausnahmen bilden dabei die Einträge 28 („Ersatz einer mechanischen Klappe durch eine andere Klappe“), 38 („Ersatz einer Bioklappe durch eine andere Klappe“) sowie 45 („Prothesenreinigung /Thrombenentfernung“).

Die Werte 28 und 38 können ebenfalls auf Basis der erfassten OPS-Schlüssel ermittelt werden, sofern zusätzlich die Art des Explantats dokumentiert und berücksichtigt wird.

Lediglich die „Prothesenreinigung / Thrombenentfernung“ (45) muss – in Ermangelung eines äquivalenten Eintrags im OPS-Katalog – explizit / separat erfasst werden.

### **3.1.71. Feld [Aortenklappenchirurgie - Ventil- bzw. Ringtyp]**

Die einschlägigen Schlüssel des OPS-Kataloges sind hinsichtlich des verwendeten Implantats nicht spezifisch genug, als dass sich mit ihrer Hilfe der Eintrag direkt ermitteln ließe.

Eine Plausibilitätsprüfung wäre auf dieser Basis zwar möglich, ein zusätzlicher Informationsgewinn im Hinblick auf eine wechselseitig widerspruchsfreie Dokumentation kann aber nicht erzielt werden, da eine Vorwegnahme dieser Kontrolle bereits durch das Verfahren zu Bestimmung des Wertes in Feld 70 (siehe Punkt 3.1.70.) erfolgt.

### **3.1.72. Feld [Aortenklappenchirurgie - Durchmesser]**

Der Inhalt dieses Feldes lässt sich weder aus Schlüsseln ableiten noch besteht die Möglichkeit zur Durchführung einer Plausibilitätsprüfung.

---

*Hier endet der separat für den durchgeführten Eingriff zu dokumentierende „intraoperative Teil“ des Datensatzes Herzchirurgie; die folgenden Felder beziehen sich auf „postoperative“ Angaben.*

---

### **3.1.73. Feld [Dauer Intensivaufenthalt]**

Der Inhalt dieses Feldes lässt sich weder aus Schlüsseln ableiten noch besteht die Möglichkeit zur Durchführung einer Plausibilitätsprüfung; es sollte jedoch möglich sein,

aus den administrativen Daten jedes am Markt erhältlichen KIS den hier geforderten Eintrag zu gewinnen.

### **3.1.74. Feld [Mobilisation]**

Es besteht keine Möglichkeit, den Inhalt dieses Feldes aus den Diagnose- oder Prozedurenschlüsseln abzuleiten; auch eine Plausibilitätsprüfung kann nicht auf dieser Basis realisiert werden.

### **3.1.75. Feld [Mediastinitis]**

Im ICD-Katalog existieren lediglich 2 Einträge, mit denen diese Erkrankung codiert werden kann: J85.3 („Abszeß des Mediastinums“) und J95.8 („Krankheiten des Mediastinums, anderenorts nicht spezifiziert“).

Da diese Schlüssel nicht für die Mediastinitis im engeren Sinne spezifisch sind, ist eine direkte Ableitung des Feldinhaltes aus den Diagnosedaten nicht möglich.

Auch eine Plausibilitätsprüfung kann nur für folgende „suspekte“ Konstellation durchgeführt werden: es wurde im Feld 75 das Auftreten einer postoperativen Mediastinitis angegeben, ein zugeordneter ICD-Code fehlt jedoch.

Der umgekehrte Fall ist hingegen plausibel: die Verwendung eines der oben genannten unspezifischen Schlüssel bedeutet nicht zwangsläufig, dass eine Mediastinitis vorlag (da auch andere Erkrankungen eingeschlossen werden); im Feld 75 „darf“ dann durchaus „Mediastinitis = nein“ dokumentiert worden sein.

### **3.1.76. Feld [Reanimation]**

Der Wert dieses Feldes lässt sich direkt aus der Menge der für den Fall erfassten OPS-Schlüssel ableiten.

### **3.1.77. Feld [Myokardinfarkt]**

Diese Angabe kann direkt aus den dokumentierten ICD-Schlüsseln eines Falles ermittelt werden, soweit sich diese auf während der postoperativen Phase neu aufgetretene Erkrankungen beziehen.

Dabei ist zu beachten, dass – sofern ein Myokardinfarkt bereits zu den Aufnahmediagnosen zählte – für ein weiteres Ereignis im Behandlungsverlauf einer der für Rezidivinfarkte (I22) codierenden Katalogeinträge verwendet wird.

### **3.1.78. Feld [Low Cardiac Output]**

Der Inhalt dieses Feldes lässt sich aus den postoperativen Diagnosen (OPS) und Prozeduren (OPS) ermitteln; dazu muss das kombinierte Vorliegen von postoperativen Diagnosen und Leistungen ausgewertet werden, was nur – wie in diesem Fall – bei einer strengen Koppelung zwischen Diagnose bzw. Indikation und Leistung (IABP oder sonstige Kreislaufunterstützung) möglich ist. Eine Ausnahme bildet dabei die „konservative Therapie“, für die kein passender OPS-Code existiert; für diese Ausprägung ist der ICD-Schlüssel alleine maßgeblich, so lange keine weiterführende Maßnahme dokumentiert wurde.

Dieses Feld ist ein Beispiel für ein eingangs erwähntes, im Kontext dieser Arbeit als „Komplex“ bezeichnetes Konstrukt; die Datenstrukturen und Algorithmen, die einer Bestimmung von Werten auf Basis von „Komplexen“ dienen, werden später ausführlich

dargestellt.

### **3.1.79. Feld [Rethorakotomie / Grund]**

Durch Berücksichtigung der für Folgeeingriffe erfassten ICD- und OPS-Codes kann der Inhalt dieses Feldes direkt bestimmt werden; es gelten die bereits unter 3.1.78. getroffenen Aussagen zur „komplex“-basierten Ableitung von Daten.

Einige der in Betracht kommenden OPS-Katalogeinträge enthalten bereits Aussagen zur Indikation („Grund“); in solchen Fällen kann auf die zusätzliche Auswertung der ICD-Schlüssel verzichtet werden.

### **3.1.80. Feld [respiratorische Insuffizienz]**

Diese Angabe lässt sich mit Hilfe der kombinierten Auswertung postoperativer ICD- und OPS-Schlüssel ableiten („Komplex“); für 2 der 3 möglichen „pathologischen“ Ausprägungen - „ja, Reintubation“ und „ja, Tracheotomie“ - müssen die zugehörigen OPS-Codes Teil der Falldaten sein. Da für die „forcierte Atemtherapie“ kein korrespondierender Eintrag im OPS-Katalog existiert, wird dieser Wert zugewiesen, wenn zwar eine „respiratorische Insuffizienz“ via ICD, jedoch weder einer Reintubation noch eine Tracheotomie (OPS!) dokumentiert wurden.

### **3.1.81. Feld [zerebrovaskuläres Ereignis bis zur Entlassung]**

Der Wert dieses Feldes lässt sich aus den ICD-Schlüsseln eines Falles ableiten, soweit sich diese auf im (postoperativen) Behandlungsverlauf neu aufgetretene Erkrankungen bzw. Ereignisse beziehen.

### **3.1.82. Feld [Dauer des zerebrovaskulären Ereignisses]**

Eine direkte Ableitung des Wertes dieses Feldes ist auf Basis von ICD-Schlüsseln nicht möglich, da lediglich eine kleine Teilmenge der Erkrankungen aus dem Formenkreis „zerebrovaskuläre Ereignisse“ beschreibenden Codes überhaupt eine Angabe zur Dauer einer Symptomatik einschließt.

Soweit es sich bei den zerebrovaskulären Ereignissen (3.1.81.) um transitorische Ischämien handelt, enthält der ICD-Katalog eine Anzahl von Einträgen (G45), deren implizite Aussagen zur Dauer der Symptomatik mit der Option „bis einschließlich 24 Stunden“übereinstimmt.

Trotzdem ist die Durchführung einer Plausibilitätsprüfung hier nicht sinnvoll, da das Feld 82 direkt vom Feld 81 abhängig ist und ein Kontrollverfahren sich lediglich auf eine Teilmenge der Schlüssel erstrecken würde, die bereits zur Ableitung des Inhaltes von Feld 81 berücksichtigt werden; ein zusätzlicher Gewinn im Hinblick auf die Frage nach einer wechselseitigen Vereinbarkeit von DRG- und QS-relevanten Daten lässt sich auf diese Weise nicht erzielen.

### **3.1.83. Feld [Schweregrad eines neurologischen Defizits bei Entlassung]**

Es existieren keine hinreichend differenzierten Einträge im ICD-Katalog, die eine Ableitung des Wertes für dieses Feld zulassen.

Zudem besteht die Möglichkeit, dass ein neu aufgetretenes neurologisches Defizit sich im weiteren Verlauf der Behandlung komplett zurückbildet.

Eine Plausibilitätsprüfung kann sich jedoch auf den Zusammenhang zwischen der

Persistenz einer entsprechenden Symptomatik (Wert  $\geq$  Rankin 1) bis zum Ende der Behandlung und der Angabe eines ICD-Codes für die Residuen (bspw. Hemiplegie, Gesichtsfelddefekte, Dysphagie, Dys- oder Aphasie) erstrecken.

### **3.1.84. Feld [Psychosyndrom]**

In diesem Feld werden Fragen nach dem Vorliegen einer Erkrankung und einer ggf. notwendigen Therapie verknüpft. Da für die prozedurale Komponente keine adäquaten OPS-Schlüssel im Katalog existieren, ist eine direkte Ableitung des Feldinhaltes nicht möglich. Eine Plausibilitätsprüfung auf Basis der ICD-Schlüssel, die für die verschiedenen Formen des Psychosyndroms codieren, kann jedoch realisiert werden.

### **3.1.85. Feld [therapiepflichtige gastrointestinale Komplikationen]**

Auch in diesem Feld ist eine Verknüpfung von Angaben erforderlich, die sich auf die Diagnose wie auch auf die Therapie von postoperativ neu auftretenden gastrointestinalen Erkrankungen beziehen.

Dabei ist besonders zu berücksichtigen, dass eine interventionelle oder chirurgische Behandlung dieser Komplikationen häufig (ggf. nach einer Verlegung) in anderen Fachabteilungen (Viszeral-Chirurgie, Innere Medizin) erfolgt, die „ihre“ Prozeduren außerhalb des herzchirurgischen Kontext dokumentieren; für eine potentielle Ableitung des Feldinhaltes relevante OPS-Schlüssel stehen somit nicht sicher bzw. nicht zeitnah zur Verfügung.

Eine Plausibilitätsprüfung ist dagegen realisierbar, wobei sowohl ICD-Schlüssel (gastrointestinale Erkrankungen, die als Komplikationen auftreten können) als auch Elemente des OPS-Kataloges (Interventionen oder chirurgische Eingriffe; für konservative Maßnahmen existieren keine Codes) berücksichtigt werden können.

### **3.1.86. Feld [Kreatininwert i.S. in mg/dl (postoperativ)]**

### **3.1.87. Feld [Kreatininwert i.S. in $\mu$ mol/l (postoperativ)]**

Analog zu den vergleichbaren präoperativen Angaben (3.1.35. & 3.1.36.) gilt auch hier: diese Werte lassen sich nicht direkt aus Einträgen des ICD-Katalogs ableiten. Es besteht aber die Möglichkeit zur Prüfung der Plausibilität, indem die Überschreitung von Grenzwerten mit der Existenz eines für eine Störung der Nierenfunktion codierenden Schlüssels verglichen wird; bei dieser Störung kann es sich um eine vorbestehende oder im Verlauf neu aufgetretene Erkrankung handeln.

### **3.1.88. Feld [postoperative Nierenersatztherapie]**

Die möglichen Inhalte dieses Feldes lassen sich aus der Kombination von OPS-Codes (postoperative Durchführung einer Nierenersatztherapie) und ggf. ICD-Schlüsseln eines Falles ableiten, sofern für die Ausprägung „chronisch“ Z99.2 („langzeitige Abhängigkeit von der Dialyse bei Niereninsuffizienz“) angegeben wird. Dabei ist es unerheblich, ob die assoziierte Erkrankung der Nieren sich erstmals im Verlauf der Behandlung manifestierte oder bereits bei Behandlungsbeginn vorlag.

### **3.1.89. Feld [Erythrozytenkonzentrat(e)]**

### **3.1.90. Feld [FFP]**

Die hier geforderten exakten Zahlen lassen sich aus den Einträgen des OPS-Kataloges, die Wertebereiche von transfundierten Einheiten definieren, nicht direkt ableiten.

Es besteht jedoch die Möglichkeit, bei von der Zahl 0 verschiedenen Angaben zu prüfen, ob per ICD eine Indikation dokumentiert und ein korrespondierender OPS-Schlüssel für die Transfusion erfasst wurde.

### **3.1.91. Feld [Thrombozytenkonzentrat]**

Im Gegensatz zu den beiden vorherigen Feldern (3.1.89. & 3.1.90.) muss hier lediglich angegeben werden, ob Thrombozytenkonzentrate transfundiert wurden, nicht jedoch deren exakte Anzahl; der einzutragende Wert lässt sich daher aus den OPS-Daten des Falles ableiten.

Zusätzlich besteht die Möglichkeit, eine sich auf die Existenz einer zugehörigen Indikation (ICD) erstreckende Plausibilitätsprüfung durchzuführen.

### **3.1.92. Feld [Herzrhythmus bei Entlassung]**

Die ICD-Verschlüsselung einer Erkrankung aus diesem Formenkreis bedeutet nicht zwangsläufig, dass sie bis zur Entlassung persistiert; wird hier jedoch angegeben, dass eine Rhythmusstörung (Optionen „Vorhofflimmern“ oder „anderer Rhythmus“) zu diesem Zeitpunkt vorliegt, kann der Frage nachgegangen werden, ob eine Diagnose dieser Kategorie entweder prä- oder postoperativ dokumentiert wurde.

Eine Plausibilitätsprüfung darf sich also nur auf die Konstellation „pathologischer Wert“ für Feld 92 bei gleichzeitigem Fehlen eines zugeordneten ICD-Schlüssels, nicht jedoch auf den umgekehrten Fall (Kombination aus vorhandenem ICD-Schlüssel „Rhythmusstörung“ und der Angabe „Sinusrhythmus bei Entlassung“) erstrecken.

### **3.1.93. Feld [Patient trägt Schrittmacher / Defibrillator]**

Bei Berücksichtigung der Angabe im Feld 17 (3.1.17. / „Patient ist Schrittmacher-/Defi-Träger“) lässt sich der Inhalt dieses Feldes auf Basis der OPS-Daten des Falles ermitteln.

Hierzu müssen folgende Konstellationen unterschieden werden:

- Falls bereits bei Aufnahme ein implantiertes Aggregat existierte und während des aktuellen Aufenthaltes keine Prozedur der Kategorie "reine Explantation" durchgeführt wurde, kann der Wert des Feldes 17 übernommen werden.
- Falls bereits bei Aufnahme ein implantiertes Aggregat existierte und während des aktuellen Aufenthaltes eine Prozedur der Kategorie "reine Explantation" durchgeführt wurde, resultiert der Wert „nein“ (eine Ausnahme von dieser Regel bildet der unmittelbar nachfolgende Punkt!).
- Falls bereits bei Aufnahme ein implantiertes Aggregat existierte und während des aktuellen Aufenthaltes eine Prozedur der Kategorien „Implantation“ bzw. "Systemwechsel" durchgeführt wurde, kann der anzugebende Wert aus den OPS-Codes abgeleitet werden.
- Falls bei Aufnahme kein implantiertes Aggregat existierte und während des aktuellen Aufenthaltes eine Prozedur der Kategorie "Implantation" durchgeführt wurde, kann der anzugebende Wert aus den OPS-Codes abgeleitet werden.
- In allen anderen Fällen erfolgt eine direkte Übernahme des Wertes aus Feld 17 - insbesondere dann, wenn ein Aggregatwechsel oder eine Sondenrevision durchgeführt wurden (denn in diesen Fällen ändert sich an der Art des Implantats nichts).

### **3.1.94. Feld [Entlassungsdiagnose(n)]**

In diesem Feld werden die ICD-Schlüssel der Entlassungsdiagnosen übermittelt; eine auf die Menge der Codes eines Falles gestützte Ableitbarkeit ist implizit gegeben.

### **3.1.95. Feld [Entlassungsdatum Krankenhaus]**

### **3.1.96. Feld [Entlassungsgrund (§301-Vereinbarung)]**

Die Inhalte dieser Felder beziehen sich auf administrative Daten zum stationären Aufenthalt; es existiert keinerlei Bezug zu ICD- bzw. OPS-codierten Diagnosen oder Leistungen.

### **3.1.97. Feld [Follow-Up - Erhebungsdatum]**

### **3.1.98. Feld [Status des Patienten am 30. postoperativen Tag]**

### **3.1.99. Feld [Todesdatum]**

Für keines dieser Felder besteht ein Bezug zum ICD- oder OPS-Katalog, der die Ableitung der jeweiligen Angabe oder die Durchführung einer Plausibilitätsprüfung ermöglichen würde.

## **3.1.100. Zusammenfassung der Analyse**

### 3.1.100.1. Direkt ableitbare Felder

Die Inhalte folgender 32 Felder (von insgesamt 99 Elementen des Datensatzes HCH) lassen sich direkt aus den codierten Diagnosen und Prozeduren (ICD / OPS) ableiten:

- [15] pulmonale Hypertonie
- [16] Herzrhythmus bei Aufnahme
- [20] Koronarangiographiefund
- [21] Signifikante Hauptstammstenose
- [24] akute Infektion(en) *{Ausnahmen: Optionen „Wundinfektion Thorax“ bzw. „Wundinfektion untere Extremität“}*
- [26] arterielle Gefäßerkrankung(en)
- [27] periphere AVK
- [28] Arteria carotis
- [29] Aortenaneurysma
- [30] sonstige arterielle Gefäßerkrankung(en)
- [32] Neurologische Erkrankung(en)
- [34] Präoperative Nierenersatztherapie
- [39] Koronarchirurgie
- [40] Aortenklappenchirurgie
- [41] Sonstige OP
- [50] Zugang
- [55] Art der Grafts
- [59] Art der Grafts
- [63] TEA
- [67] anderer Koronareingriff
- [68] Aortenklappenchirurgie - Stenose
- [69] Aortenklappenchirurgie – Insuffizienz

- [70] Klappeneingriff *{Ausnahmen: siehe 3.1.70.}*
- [76] Reanimation
- [77] Myokardinfarkt postoperativ
- [78] Low cardiac output
- [79] Rethorakotomie / Grund
- [81] Zerebrovaskuläres Ereignis bis zur Entlassung
- [80] Respiratorische Insuffizienz
- [88] Postoperative Nierenersatztherapie
- [91] Thrombozytenkonzentrat
- [93] Pat. trägt Schrittmacher/Defibrillator

### 3.1.100.2. Wechselseitig anhand der Schlüssel (ICD / OPS) prüfbare Felder

Die Inhalte folgender 43 Felder (von insgesamt 99 Elementen des Datensatzes HCH) können zum Gegenstand einer wechselseitigen Prüfung auf Basis der codierten Diagnosen und Prozeduren (ICD / OPS) werden:

- [9] klinischer Schweregrad der Herzinsuffizienz (NYHA-Klassifikation)
- [10] Angina pectoris (nach CCS)
- [11] Infarkt(e) {präoperativ}
- [12] kardiogener Schock / Dekompensation
- [13] Reanimation
- [14] Patient wird beatmet
- [17] Patient ist Schrittmacher-/Defi-Träger
- [18] Einstufung nach ASA-Klassifikation und Leistungseinschränkung (W)
- [22] PCI
- [23] Vor-OP(s) an Herz/Aorta (Anzahl)
- [24] akute Infektion(en) *{Betrifft nur die Optionen „Wundinfektion Thorax“ bzw. „Wundinfektion untere Extremität“}*
- [25] Diabetes mellitus
- [31] Lungenerkrankung(en)
- [33] Schweregrad der Behinderung
- [35] Kreatininwert i.S. präoperativ [mg/dl]
- [36] Kreatininwert i.S. präoperativ [µmol/l]
- [43] Dringlichkeit
- [44] Nitrate (i.v.)
- [45] Troponin positiv
- [46] Inotrope (i.v.)
- [47] (präoperativ) mechanische Kreislaufunterstützung
- [49] OP-Vorgehen
- [54] Anzahl der Grafts
- [56] Art der Grafts: ITA links
- [57] Art der Grafts: ITA rechts
- [58] Art der Grafts: A. radialis
- [60] zentrale Anastomose(n)
- [61] periphere Anastomose(n), arteriell
- [62] periphere Anastomose(n), venös
- [64] Versorgte Koronararterien: LAD und / oder Äste
- [65] Versorgte Koronararterien: RCA und / oder Äste
- [66] Versorgte Koronararterien: RCX und / oder Äste
- [71] Aortenklappenchirurgie - Ventil- bzw. Ringtyp
- [75] Mediastinitis

- [82] Dauer des zerebrovaskulären Ereignisses
- [83] Schweregrad eines neurologischen Defizits bei Entlassung
- [84] Psychosyndrom
- [85] therapiepflichtige gastrointestinale Komplikation(en)
- [86] Kreatininwert i.S. (postoperativ) [mg/dl]
- [87] Kreatininwert i.S. (postoperativ) [µmol/l]
- [89] Erythrozytenkonzentrat(e) [Einheiten]
- [90] FFP [Einheiten]
- [92] Herzrhythmus bei Entlassung

### **3.2. Strukturen für den ICD- sowie den OPS-Katalog**

Die Daten der beiden Kataloge bilden einen essentiellen Bestandteil des hier vorgestellten Datenmodells. Sowohl das Verfahren zur Ableitung von Feldinhalten wie auch die Durchführung von Plausibilitätsprüfungen fußen im wesentlichen auf der Definition von Mengen aus Codes, die im Hinblick auf eine bestimmte Aussage relevant sind.

Zur Gewährleistung der sog. „referentiellen Integrität“ ist eine vollständige Abbildung des ICD- wie auch des OPS-Katalogs notwendig.

Dabei wird – sehr verkürzt bzw. vereinfacht dargestellt – durch anzulegende Datenbankregeln („Constraints“) garantiert, dass innerhalb des Systems, das Gegenstand dieser Arbeit ist, nur solche Schlüssel verwendet werden können, die tatsächlich Bestandteile der offiziellen Klassifikationen sind.

Konkret wird die Ausführung von Operationen auf dem Datenbestand unterbunden, die zur Übernahme fehlerhaft spezifizierter Schlüssel (Beispiel: „Schreibfehler“) führen würden; dabei prüft das Datenbanksystem auf Grundlage der oben erwähnten Integritätsbedingung („Regeln“), ob der angegebene Code Bestandteil der Hinterlegung aller zulässigen Katalogeinträge ist.

Abgesehen von dem Umstand, dass die formale Korrektheit der Datenbasis in Bezug auf die rechtlichen Grundlagen (offizieller Charakter der durch das DIMDI veröffentlichten und zur Verfügung gestellten Kataloge) auf die soeben beschriebene Weise sichergestellt wird, stellen die Tabellen mit den Katalogdaten (Schlüssel, Klartext-Angaben und synonyme Begriffe) die recherchierbare Basis für Suchwerkzeuge dar, mit deren Hilfe die Pflege und Fortentwicklung der Datenbestände komfortabel gestaltet werden kann. Dieser Punkt wird bei der späteren Vorstellung der das Datenmodell nutzenden Beispielanwendung näher erläutert.

#### **3.2.1. Beschreibung der Entitäten und der Beziehungen zwischen diesen**

Die für die Daten beider Kataloge bestimmten Tabellen sind strukturell identisch aufgebaut. Identifizierbare Einheiten bilden jeweils Versionen, die Schlüssel selbst im engeren Sinne sowie synonyme Begriffe der Diagnosen bzw. Prozeduren. Zusätzlich muss die Seitenlokalisierung in systematisierter Form spezifiziert werden.

##### 3.2.1.1. Versionen: Tabellen [ICD\_VERSION] und [OPS301\_VERSION]

Gegenwärtig erscheinen jährlich aktualisierte Versionen der beiden Kataloge. Neben Eigenschaften wie Bezeichnung oder Kürzel ist der Zeitpunkt, ab dem eine Version gültig



ist, von besonderer Bedeutung, lässt sich doch mit Hilfe dieser Angabe in Kombination mit dem Behandlungsbeginn die für einen Fall anzuwendende Version ermitteln.

*Hinweis:*

*wie ausnahmslos alle Tabellen des Datenmodells enthalten auch [ICD\_VERSION] und [OPS\_VERSION] Felder, die jeweils den Primärschlüssel - also den eindeutigen Identifikator für jeden Datensatz – darstellen (Anhang 6). Konkret sind das [ICDVER\_ID] in der Tabelle [ICD\_VERSION] bzw. [OPSVER\_ID] in der Tabelle [OPS301\_VERSION], anhand derer sich das Muster für die Benennung aller Primärschlüssel-Felder nachvollziehen lässt: einer an der Bezeichnung der Tabelle orientierten Abkürzung folgt die Sequenz [\_ID]. Die Tatsache, dass jede der im weiteren Verlauf vorgestellten Tabellen einen aus einem einzelnen Feld bestehenden Primärschlüssel enthält, der nach der soeben beschriebenen Konvention bezeichnet wird, findet künftig – als allgemeines, Datenmodell-weit gültiges Designprinzip - keine explizite Erwähnung mehr.*

### 3.2.1.2. Schlüssel: Tabellen [ICD] und [OPS301]

Diese Tabellen enthalten die Angaben zu den „eigentlichen“ Schlüsseln, die ihrerseits genau einer Version des jeweiligen Kataloges zugeordnet sind (3.2.1.1).

*Hinweis:*

*Abgebildet wird diese Zuordnung durch die Aufnahme des Primärschlüssels der Versions-Tabelle als Feld in die Tabellen mit den Daten zu den ICD- bzw. OPS-Katalogelementen und dessen Definition als Fremdschlüssel mit Hilfe einer Regel. Dieses auf sog. „foreign-key-constraints“ gestützte Verfahren kommt innerhalb des Datenmodells in ausnahmslos allen Fällen zur Anwendung, in denen Beziehungen des zuvor geschilderten Typs berücksichtigt bzw. etabliert werden müssen (Anhang 6).*

Daneben bilden Felder für den „offiziellen“ Code (bspw. „I25.13“) sowie die die jeweilige Diagnose beschreibenden Texte (Kurztext, Volltext, interne Bezeichnung) weitere Bestandteile.

### 3.2.1.3. Synonyme Begriffe: Tabellen [ICD\_THESAURUS] und [OPS301\_THESAURUS]

Zu jedem Katalogeintrag können mehrere synonyme Begriffe existieren (bspw. „3G-KHK“, „3-Gefäß-KHK“ oder „koronare 3-Gefäß-Erkrankung“ für den ICD-Code I25.13). Für diese zusätzlichen Informationen sind die mit dem Namen [ALTERNATIVTEXT] bezeichneten Felder der Tabellen [ICD\_THESAURUS] bzw. [OPS301\_THESAURUS] vorgesehen.

### 3.2.1.4. Lokalisation: Tabellen [ICD\_LOKALISATION] und [OPS301\_LOKALISATION]

In diesen Tabellen werden die Angaben zur Lokalisation (links, rechts, beidseits etc.) definiert, mit denen Fall-bezogene ICD- und OPS-codierte Diagnosen bzw. Leistungen bei Bedarf ergänzt werden können.

### 3.2.1.5. Suchtexte: Tabellen [ICD\_SUCHTEXT] und [OPS301\_SUCHTEXT]

Nicht mit allen am Markt erhältlichen RDBMS lässt sich eine performante Suche nach Fragmenten oder kompletten Texten in den „langen“ Feldern der Tabellen [ICD] oder [OPS301] durchführen. Ausschließlich für die Lösung dieses rein technischen Problems (das allerdings mit dem hier verwendeten Produkt des Herstellers Oracle nicht auftrat) wurden die beiden oben genannten Tabellen angelegt, deren in Bezug auf die Länge begrenzten Felder namens [SUCHTEXT] im Bedarfsfall die Realisation effizienter

Suchoperationen zulassen.

### 3.2.2. Grafische Darstellung des Entitäts-Beziehungs-Modells (ER-Modell)

Zur Darstellung der einzelnen Teile des Datenmodells wird in den folgenden Abschnitten die sog. „Krähenfuß-Notation“ benutzt, wie sie standardmäßig in den Werkzeugen des Herstellers Oracle Anwendung findet; eine schematische Darstellung ihrer grafischen Elemente findet sich im Anhang 6.

*Hinweise:*

*Die weiter oben erwähnten Fremdschlüssel werden in dieser Form der Darstellung nicht explizit in der Feldliste der sie „beherbergenden“ Tabelle aufgeführt; vielmehr wird der so definierte Bezug durch die Linien zwischen den Entitäten/Tabellen abgebildet. Für das Verständnis der Abbildungen ist zusätzlich das Wissen um die Kennzeichnung der gelisteten Felder relevant:*

- *[#]* kennzeichnet Felder, aus denen sich der Primärschlüssel einer Entität/Tabelle zusammensetzt.
- *[\*]* kennzeichnet Felder, die einen Wert enthalten müssen („Pflichtfelder“)
- *[o]* kennzeichnet Felder, die nicht zwangsläufig einen Wert enthalten müssen.

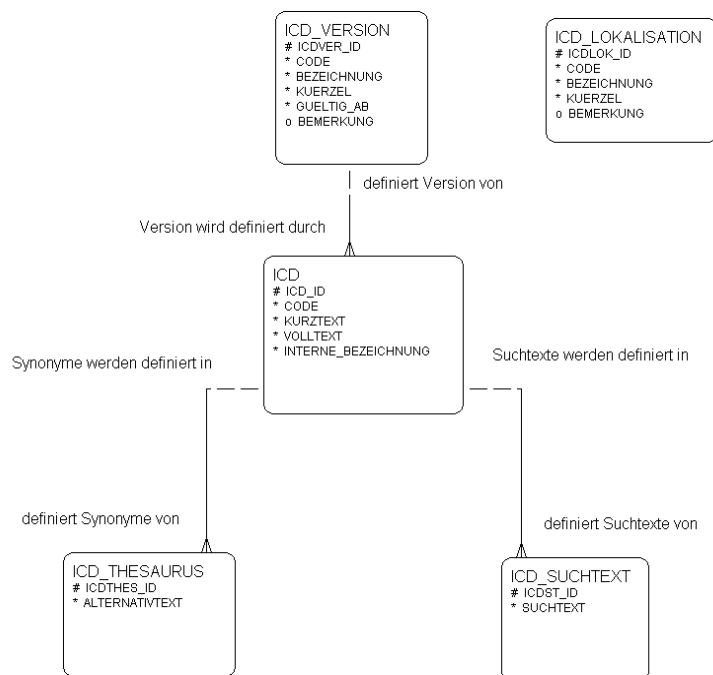
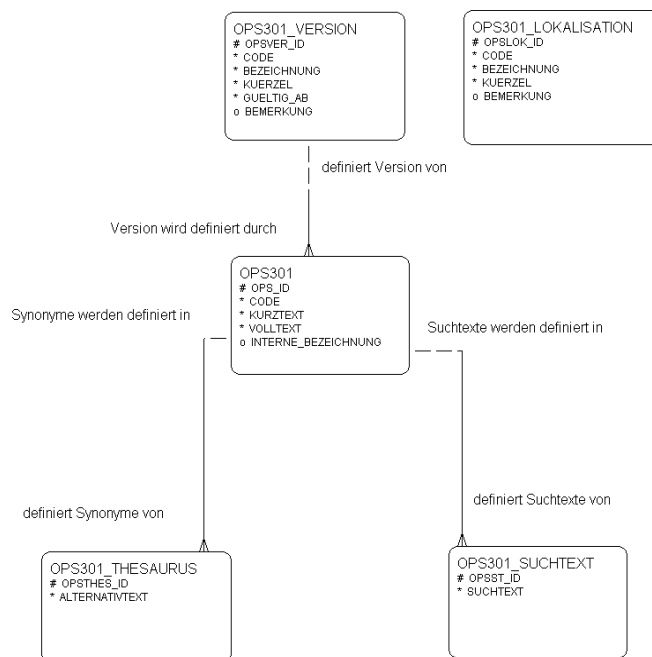


Abbildung 5: ER-Modell ICD-Katalog



**Abbildung 6: ER-Modell OPS-Katalog**

### 3.2.3. Die aus dem Entitäts-Beziehungs-Modell resultierenden SQL-Anweisungen

Mit Hilfe der Oracle-Software „Designer“ lassen sich aus den zuvor dargestellten ER-Modellen die in der Datenbank-Sprache SQL-92 [29] formulierten Anweisungen erzeugen, die in relationalen Datenbank-Systemen die Tabellen ebenso wie die vorgestellten Regeln anlegen. Letztere definieren unter anderem die Beziehungen zwischen den Tabellen und werden als „Constraints“ etabliert.

Der Standard SQL-92 wurde gewählt, weil der Großteil der am Markt erhältlichen kommerziellen wie freien relationalen Datenbankmanagementsysteme die dieser Syntax folgenden Anweisung interpretieren (im Sinne von „verstehen und umsetzen“) können.

Die zur Implementierung verwendeten Befehle werden nur in diesem Abschnitt – und, aufgrund des identischen Aufbaus der Datenstrukturen für die beiden Schlüsselwerke – nur für den ICD-Katalog exemplarisch aufgeführt. Um den Umfang des Textes auf ein vertretbares Maß zu begrenzen, wurde das Skript, das das komplette Datenmodell anlegt, in den Anhang 3 verlagert.

**Hinweis:**

*Nicht alle Constraints, die durch dieses Teil- bzw. durch das Gesamt-Skript im Anhang angelegt werden, lassen sich aus den ER-Modellen herleiten. Sie können daher nicht automatisch durch Funktionen des „Designer“ erzeugt werden, sondern sind vielmehr Ergebnisse separat durchzuführender Modellierungsschritte. Von besonderer Bedeutung sind dabei die „unique“-Constraints, die auch im Beispiel vorkommen. Mit ihrer Hilfe lassen sich Sekundärschlüssel definieren – also Felder bzw. Kombinationen aus Feldern, deren (kombinierte) Inhalte innerhalb einer Tabelle eindeutig (im Sinne von einmalig) sein müssen. Anders ausgedrückt stellen „unique“-Regeln sicher, dass – bezogen auf eine bestimmte Tabelle – nicht zwei oder mehr Datensätze existieren, die denselben Wert im betreffenden Feld oder der Kombination von Feldern aufweisen. Beispielsweise würde es*

keinen Sinn ergeben, wenn es in der Tabelle [ICD\_VERSION] zwei Einträge mit der [BEZEICHNUNG] „2007“ geben würde, da es ganz offensichtlich nur eine Ausgabe des Kataloges mit diesem Namen gibt. „Unique“-Constraints stellen also sicher, dass Manipulationen am Datenbestand, die Regeln der zuvor beschriebenen Art verletzen, vom Datenbank-System zurückgewiesen werden. Daneben stellen Sekundärschlüssel zusätzliche, d.h. neben dem Primärschlüssel nutzbare, eindeutige Identifikatoren der Datensätze einer Tabelle dar.

```

CREATE TABLE ICD_VERSION
(
ICDVER_ID      INTEGER          NOT NULL,
CODE           VARCHAR(20)      NOT NULL,
BEZEICHNUNG   VARCHAR(200)     NOT NULL,
KUERZEL       VARCHAR(20)      NOT NULL,
GUELTIG_AB   DATE              NOT NULL,
BEMERKUNG     VARCHAR(2000)
);

CREATE TABLE ICD
(
ICD_ID         INTEGER          NOT NULL,
ICDVER_ID     INTEGER          NOT NULL,
CODE           VARCHAR(10)      NOT NULL,
KURZTEXT      VARCHAR(2000)    NOT NULL,
VOLLTEXT      VARCHAR(2000)    NOT NULL,
INTERNE_BEZEICHNUNG VARCHAR(200) NOT NULL
);

CREATE TABLE ICD_THESAURUS
(
ICDTHES_ID    INTEGER          NOT NULL,
ICD_ID        INTEGER          NOT NULL,
ALTERNATIVTEXT VARCHAR(255)    NOT NULL
);

CREATE TABLE ICD_SUCHTEXT
(
ICDST_ID      INTEGER          NOT NULL,
ICD_ID        INTEGER          NOT NULL,
SUCHTEXT      VARCHAR(255)    NOT NULL
);

CREATE TABLE ICD_LOKALISATION
(
ICDLOK_ID     INTEGER          NOT NULL,
CODE           VARCHAR(20)      NOT NULL,
BEZEICHNUNG   VARCHAR(200)     NOT NULL,
KUERZEL       VARCHAR(20)      NOT NULL,
BEMERKUNG     VARCHAR(2000)
);

ALTER TABLE ICD_VERSION
ADD CONSTRAINT ICD_VERSION_PK
PRIMARY KEY(ICDVER_ID);

ALTER TABLE ICD
ADD CONSTRAINT ICD_PK
PRIMARY KEY(ICD_ID);

ALTER TABLE ICD_THESAURUS
ADD CONSTRAINT ICD_THESAURUS_PK
PRIMARY KEY(ICDTHES_ID);

ALTER TABLE ICD_LOKALISATION
ADD CONSTRAINT ICD_LOKALISATION_PK

```

```

PRIMARY KEY(ICDLOK_ID);

ALTER TABLE ICD_SUCHTEXT
ADD CONSTRAINT ICD_SUCHTEXT_PK
PRIMARY KEY(ICDST_ID);

ALTER TABLE ICD_VERSION
ADD CONSTRAINT ICD_VERSION_CODE_UK
UNIQUE(CODE);

ALTER TABLE ICD_VERSION
ADD CONSTRAINT ICD_VERSION_BEZEICHNUNG_UK
UNIQUE(BEZEICHNUNG);

ALTER TABLE ICD_VERSION
ADD CONSTRAINT ICD_VERSION_KUERZEL_UK
UNIQUE(KUERZEL);

ALTER TABLE ICD_VERSION
ADD CONSTRAINT ICD_VERSION_GUELTIG_AB_UK
UNIQUE(GUELTIG_AB);

ALTER TABLE ICD
ADD CONSTRAINT ICD_ICDVER_ID_CODE_UK
UNIQUE(ICDVER_ID,CODE);

ALTER TABLE ICD_THESAURUS
ADD CONSTRAINT ICD_THES_ICD_ID_ALTTEXT_UK
UNIQUE(ICD_ID,ALTERNATIVTEXT);

ALTER TABLE ICD_LOKALISATION
ADD CONSTRAINT ICD_LOKALISATION_CODE_UK
UNIQUE(CODE);

ALTER TABLE ICD_LOKALISATION
ADD CONSTRAINT ICD_LOKALISATION_BEZEICH_UK
UNIQUE(BEZEICHNUNG);

ALTER TABLE ICD_LOKALISATION
ADD CONSTRAINT ICD_LOKALISATION_KUERZEL_UK
UNIQUE(KUERZEL);

ALTER TABLE ICD
ADD CONSTRAINT ICD_ICD_VER_FK
FOREIGN KEY (ICDVER_ID) REFERENCES ICD_VERSION(ICDVER_ID);

ALTER TABLE ICD_THESAURUS
ADD CONSTRAINT ICD_THESAURUS_ICD_FK
FOREIGN KEY (ICD_ID) REFERENCES ICD(ICD_ID);

ALTER TABLE ICD_SUCHTEXT
ADD CONSTRAINT ICD_SUCHTEXT_ICD_FK
FOREIGN KEY (ICD_ID) REFERENCES ICD(ICD_ID);

```

### 3.2.4. Die aus dem Entitäts-Beziehungs-Modell resultierenden konkreten Tabellen

Zum besseren Verständnis, welche Ergebnisse insbesondere die „CREATE TABLE“-Befehle des obigen Beispiels zeitigen, werden hier beispielhaft die konkreten, auf diese Weise erzeugten Tabellen des ER-Modells „ICD-Katalog“ schematisch abgebildet.

In den folgenden Abschnitten sind die Darstellungen des Datenmodells jedoch auf die ER-Modelle beschränkt. Die Details der Tabellen-Strukturen lassen sich dem Skript, das das komplette Datenmodell anlegt (und hier speziell den Feldlisten der „CREATE TABLE“-Kommandos, siehe oben), im Anhang 3 entnehmen.

<b>ICD_VERSION</b>		
Feld	Typ	Pflichtfeld?
ICDVER_ID	INTEGER	ja
CODE	VARCHAR(20)	ja
BEZEICHNUNG	VARCHAR(200)	ja
KUERZEL	VARCHAR(20)	ja
GUELTIG_AB	DATE	ja
BEMERKUNG	VARCHAR(2000)	nein

<b>ICD</b>		
Feld	Typ	Pflichtfeld?
ICD_ID	INTEGER	ja
ICDVER_ID	INTEGER	ja
CODE	VARCHAR(10)	ja
KURZTEXT	VARCHAR(2000)	ja
VOLLTEXT	VARCHAR(2000)	ja
INTERNE_BEZEICHNUNG	VARCHAR(200)	nein

<b>ICD_THESAURUS</b>		
Feld	Typ	Pflichtfeld?
ICDTHES_ID	INTEGER	ja
ICD_ID	INTEGER	ja
ALTERNATIVTEXT	VARCHAR(200)	ja

<b>ICD_SUCHTEXT</b>		
Feld	Typ	Pflichtfeld?
ICDST_ID	INTEGER	ja
ICD_ID	INTEGER	ja
SUCHTEXT	VARCHAR(200)	ja

ICD_LOKALISATION		
Feld	Typ	Pflichtfeld?
ICDLOK_ID	INTEGER	ja
CODE	VARCHAR(20)	ja
BEZEICHNUNG	VARCHAR(200)	ja
KUERZEL	VARCHAR(20)	ja
BEMERKUNG	VARCHAR(2000)	nein

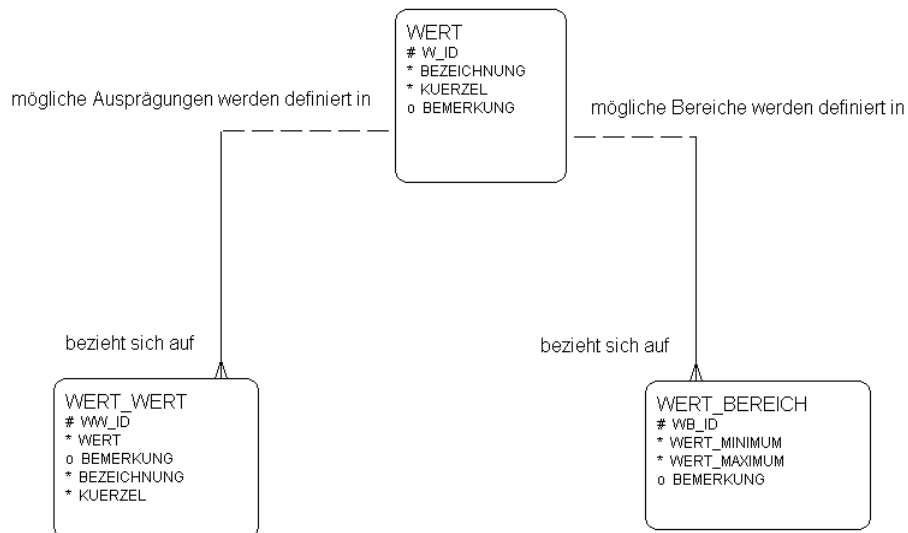
### 3.3. Strukturen für codeunabhängige Werte

Neben den ICD- und OPS-Daten eines Falles sind hauptsächlich im Hinblick auf die Durchführung von Plausibilitätsprüfungen solche Angaben bedeutsam, die sich nicht direkt aus den Schlüsseln der beiden Kataloge ermitteln lassen (Ausführungen zu den Punkten 3.1.1. - 3.1.99.).

Die Informationen dieser Kategorie sind heterogen: numerische Angaben wie Laborwerte (Beispiel: Kreatinin i.S., 3.1.35.) zählen ebenso dazu wie Felder, für die nur ein begrenzter Wertevorrat - im Sinne zulässiger Optionen - zur Verfügung steht (Beispiel: Mobilisation, 3.1.74.).

Allen gemeinsam ist, dass sie in textbasierter Form (als alphanumerische Zeichenketten) aus dem KIS übernommen und so auch temporär (Punkt 3.4.) gespeichert werden.

Um die nicht ICD- bzw. OPS-bezogenen Daten im Rahmen der Plausibilitätsprüfung trotz ihrer Vielgestaltigkeit strukturiert auswerten zu können, wurden – zusätzlich zur die Eigenschaften der codeunabhängigen Werte beschreibenden Tabelle [WERT] - Tabellen angelegt, mit deren Hilfe Intervalle für numerische Angaben ([WERT\_BEREICH]) bzw. konkrete Ausprägungen für bestimmte Felder ([WERT\_WERT]) definiert werden können, die eine Rolle im Kontext zu berücksichtigender Regeln spielen (Abbildung 7). Die Nutzung der auf diese Weise hinterlegten Definitionen wird im Abschnitt „Plausibilitätsprüfung“ (3.6.) vorgestellt.



**Abbildung 7: ER-Modell codeunabhängige Werte**

### 3.4. Strukturen für die Falldaten

#### 3.4.1. Basisdaten der Fälle

Das hier vorgestellte Datenmodell soll keine Parallelstrukturen zu etablierten und mehr oder weniger ausgereiften Krankenhausinformationssystemen schaffen und deshalb fallbezogene Informationen nicht dauerhaft vorhalten. Vielmehr werden Angaben zu Aufenthalten und Operationen, die sich aus den ICD- und OPS-Schlüsseln (3.2.) sowie den codeunabhängigen Werten (3.3.) zusammensetzen, lediglich temporär gespeichert.

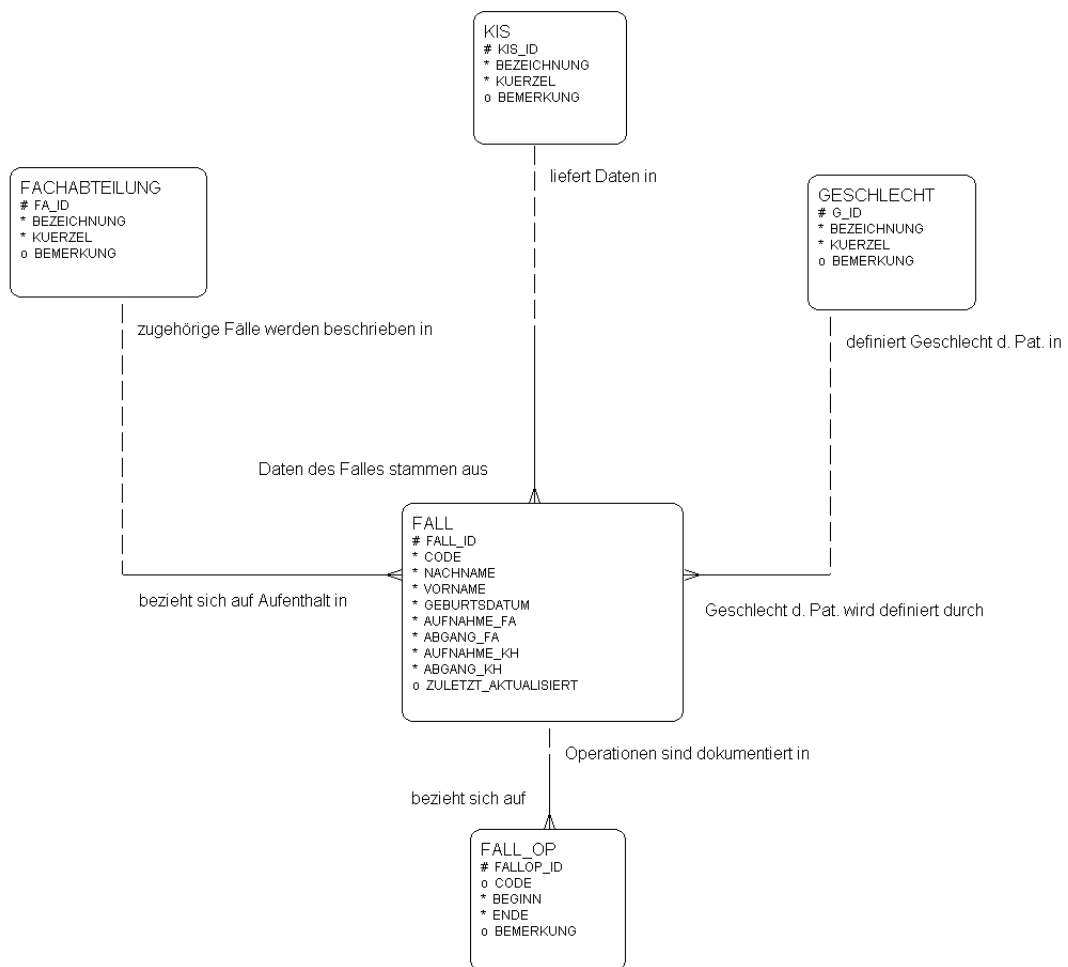
Von Relevanz sind dabei vor allem Daten, die der „Identifizierbarkeit“ von Fällen (Name, Vorname, KIS-spezifische Fallnummer etc.) dienen und eine zeitliche Einordnung der (erstmaligen) Erhebung von Diagnosen bzw. der Durchführung von Operationen und anderen Maßnahmen zulassen. Entsprechend reduziert stellt sich der Umfang der in der „zentralen“ Tabelle [FALL] berücksichtigten Angaben dar, die vor allem Beginn und Ende der Aufenthalte in der Fachabteilung bzw. innerhalb des Gesamt-Klinikums abbilden.

Jedem Fall können eine oder mehrere Operationen in der Tabelle [FALL\_OP] zugeordnet sein, deren wesentliches Charakteristikum der Zeitpunkt ihres Beginns ist.

*Hinweis:*

*Die bislang erwähnten Basisdaten der Fälle (ICD- und OPS-Schlüssel, zeitliche Eckdaten etc.) sind Gegenstand der Vereinbarungen zum Datenaustausch gemäß § 301 SGB V [11], gehören also zum Repertoire von „Pflichtangaben“, das in der Datenbasis jedes derzeit am Markt erhältlichen KIS Berücksichtigung finden muss.*





**Abbildung 8: ER-Modell Falldaten (Basis)**

### 3.4.2. ICD- und OPS-Daten der Fälle

Die beiden Tabellen [FALL\_ICD] (Abbildung 9) und [FALL\_OPS301] (Abbildung 10) enthalten die Diagnose- und Leistungsschlüssel der Fälle. Neben dem Bezug zum Fall sowie dem Element des betreffenden Schlüsselwerks (ICD- bzw. OPS-Katalog) wird die Seiten-Lokalisation (links, rechts, beidseits, nicht angegeben) definiert.

Für die chronologische Zuordnung zu den Phasen des Behandlungsverlaufes (z.B. prä-, intra- und postoperativ) relevant ist die Angabe des Datums: im Fall von ICD-Codes handelt es sich um den Zeitpunkt, an dem die Diagnose gestellt wurde, während bei OPS-Daten der Beginn der Prozedur beschrieben wird.

Ferner wird jeder Schlüssel eines Falles einer ICD- bzw. OPS-Kategorie zugeordnet; vor allem bei der in den später folgenden Abschnitten beschriebenen Bewertung von Diagnosen ist von besonderem Interesse, ob es sich um bereits zum Beginn des Aufenthaltes bestehende („Aufnahmediagnosen“) oder erst im weiteren Verlauf der Behandlung auftretende Erkrankungen („Behandlungs- oder Nachfolgediagnosen“) handelt. Neben der Angabe „Kategorie“ spielt dabei das bereits zuvor erwähnte Feld „Datum“ eine wichtige Rolle.

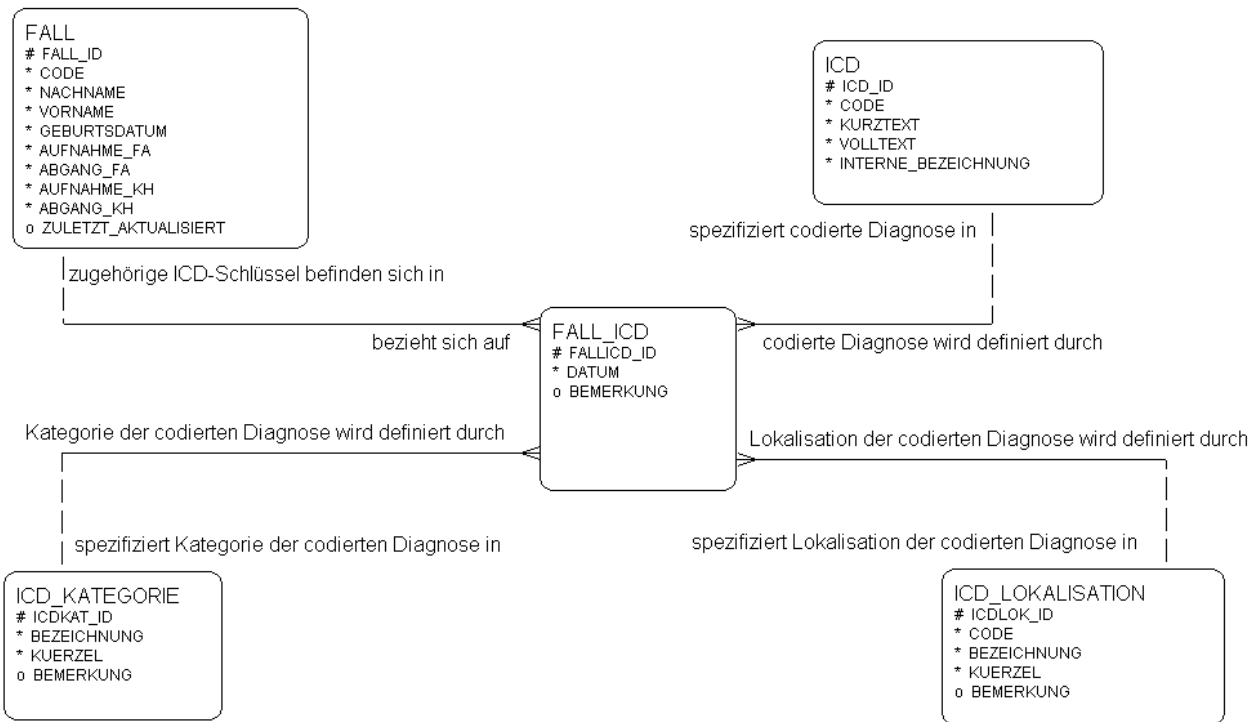


Abbildung 9: ER-Modell ICD-Daten der Fälle

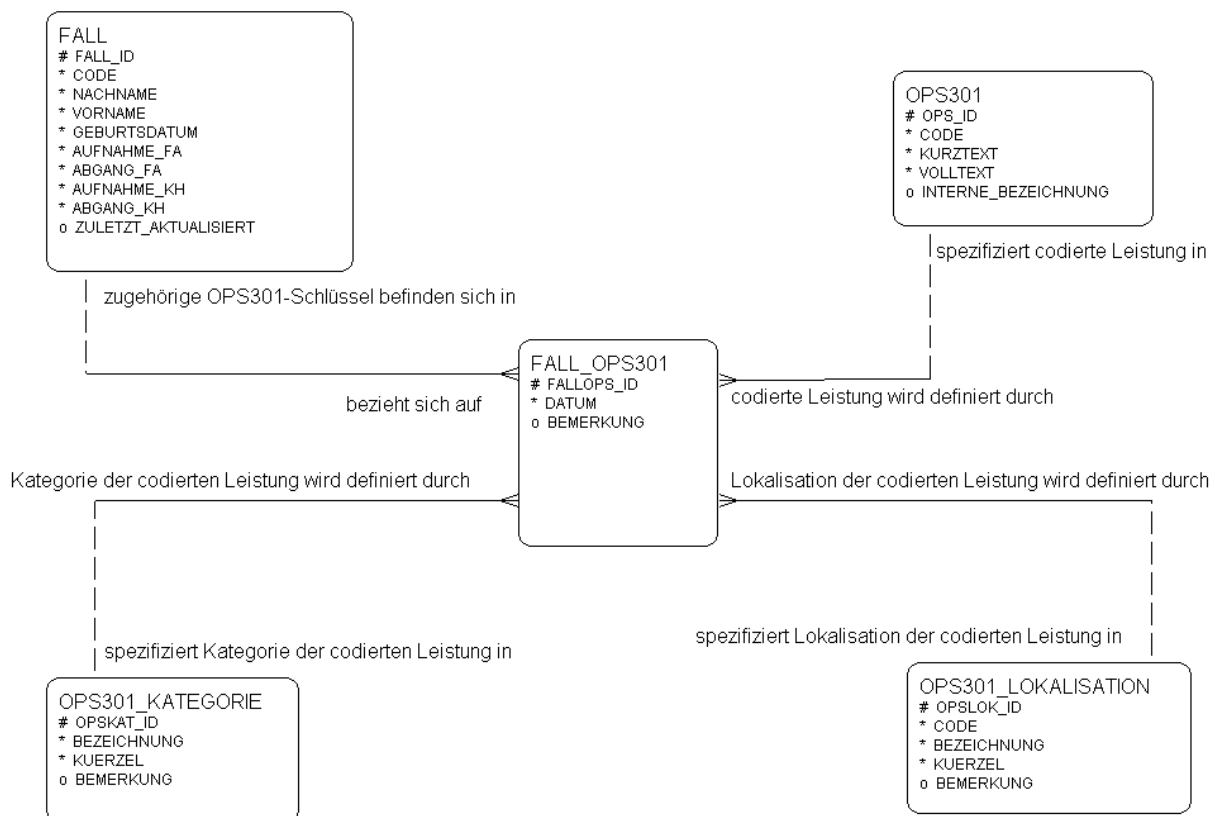


Abbildung 10: ER-Modell OPS-Daten der Fälle

### 3.4.3. Codeunabhängige Daten der Fälle

Für fallbezogene Angaben, die sich nicht direkt aus codierten Diagnosen (ICD) oder Leistungen (OPS) herleiten lassen, ist die Tabelle [FALL\_WERT] (Abbildung 11) vorgesehen.

Neben dem Bezug zum jeweiligen Fall und der Qualifikation der Information („Um welche Angabe handelt es sich?“) durch den Verweis auf einen Eintrag der Tabelle [WERT] (3.3.) bildet die konkrete Ausprägung des spezifizierten Wertes ihren wesentlichen Bestandteil (Beispiel: „1.9“, wenn der Verweis auf den Eintrag in der Tabelle [WERT] definiert, dass es sich um den Wert des Feldes „Kreatininwert i.S. in mg/dl (präoperativ)“ handelt; 3.1.35.).

Um der Heterogenität der hier zu hinterlegenden Daten gerecht zu werden, werden die eigentlichen Werte ausnahmslos als alphanumerische Zeichenketten, d.h. in Form von Texten, gespeichert.

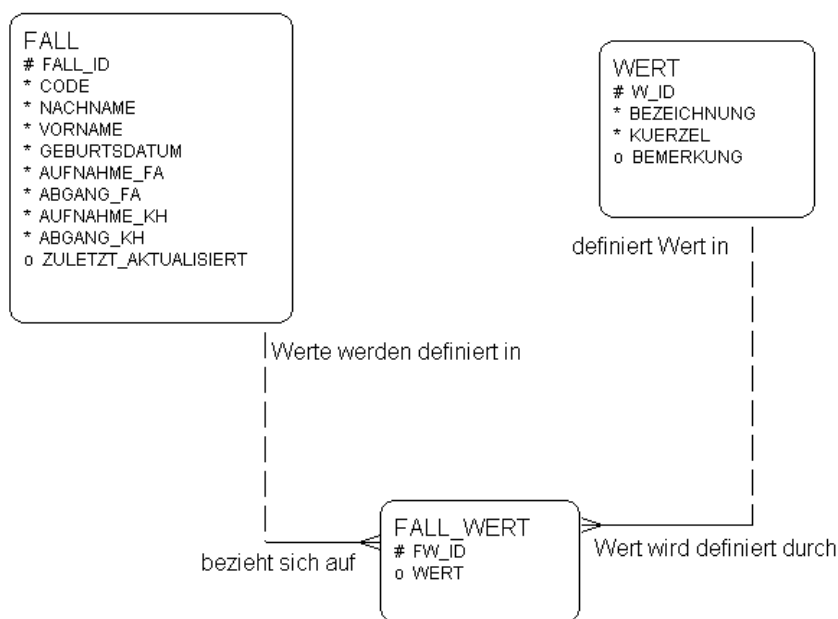
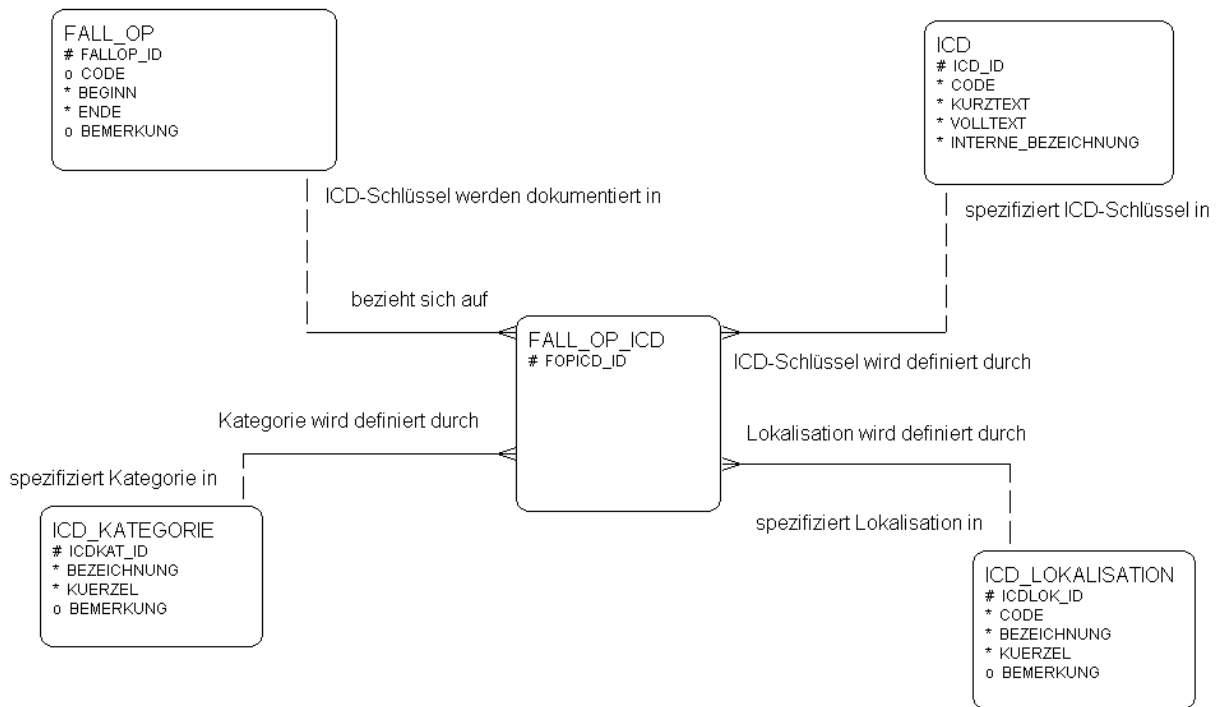


Abbildung 11: ER-Modell codeunabhängige Werte der Fälle

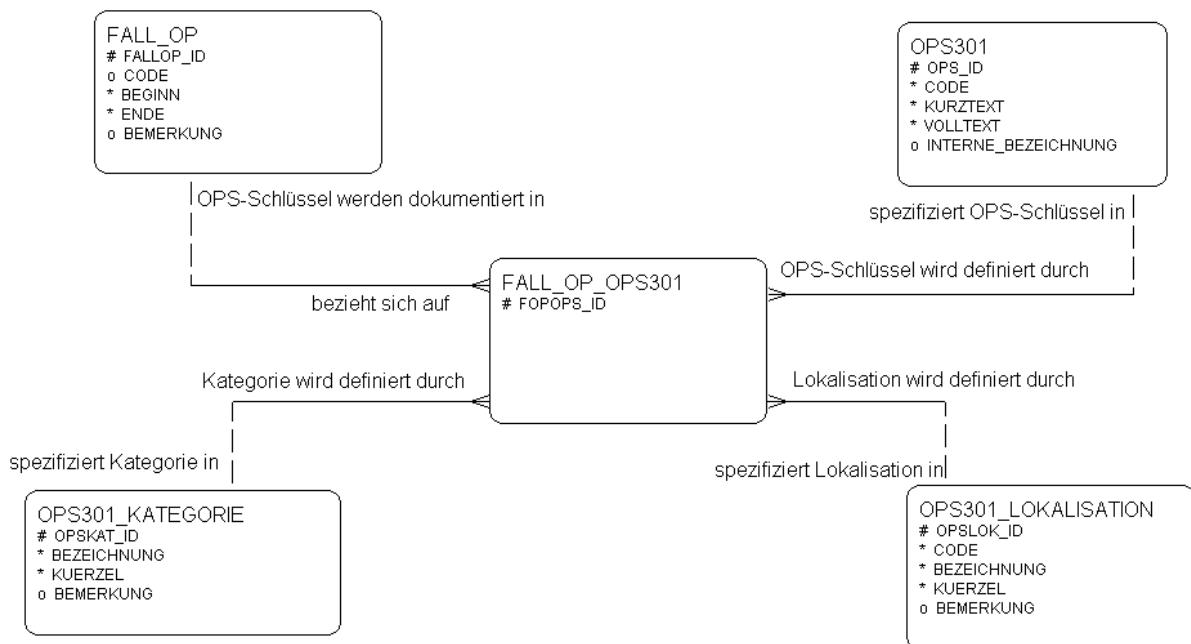
### 3.4.5. ICD-, OPS- und codeunabhängige Daten der Operationen

Die Strukturen der für operationsbezogene Informationen bestimmten Tabellen [FALL\_OP\_ICD] (Abbildung 12), [FALL\_OP\_OPS301] (Abbildung 13) und [FALL\_OP\_WERT] (Abbildung 14) ähneln denjenigen ihrer fallbezogenen Gegenstücke; die Ausführungen dazu (3.4.2., 3.4.3.) gelten analog und werden daher nicht wiederholt.

Im Unterschied zu diesen Pendanten wird der Zeitpunkt der Diagnose-Erhebung bzw. des Beginns einer Prozedur nicht für jeden Schlüssel einzeln vorgehalten, da diese Angabe bereits durch den Beginn der zugeordneten Operation definiert ist.



**Abbildung 12: ER-Modell ICD-Daten der Operationen**



**Abbildung 13: ER-Modell OPS-Daten der Operationen**

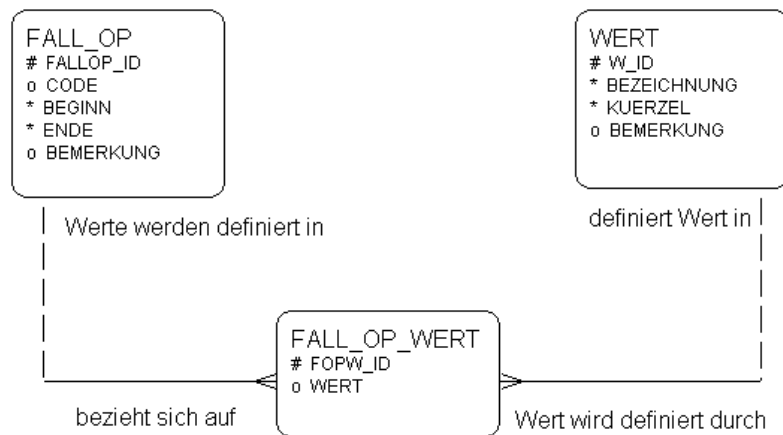


Abbildung 14: ER-Modell codeunabhängige Daten der Operationen

### 3.5. Ableitung von Werten aus codierten Diagnosen und Leistungen

Eines der zentralen Ziele, die mit der Entwicklung des Datenmodells verfolgt wurden, stellt die Möglichkeit dar, aus den ICD- und OPS-Daten eines Fall diagnose- bzw. leistungsbezogene Aussagen direkt ermitteln zu können.

Prinzipiell können 4 Konstellationen im Hinblick auf die Herleitung unterschieden werden:

- **„Einfache“ Werte:** eine diagnose- oder leistungsbezogene Aussage – beispielsweise die konkrete Aussage „1-Gefäßerkrankung“ für das Feld „Koronarangiographiebefund“ (3.1.20.) - folgt implizit aus der Dokumentation eines Schlüssels, der einer definierten Teilmenge des jeweiligen Katalogs entstammt (im genannten Beispiel: einer Teilmenge des ICD-Katalogs, die lediglich das Element „I25.11“ umfasst)
- **Kombinierte Werte:** eine diagnose- oder leistungsbezogene Option für ein Feld basiert auf der kombinierten, also gleichzeitigen bzw. gemeinsamen Angabe von Schlüsseln, die Elemente von 2 oder mehr verschiedenen Teilmengen des jeweiligen Katalogs sind; ein Beispiel hierfür ist die mögliche Aussage „ja, Kombination“ für das Feld „neurologische Erkrankung(en)“ (3.1.32.).
- **„Komplexe“:** dieser Begriff bezeichnet im Kontext dieser Arbeit Ausprägungen von Feldern, die sich gleichzeitig auf Diagnosen wie auch auf durchgeführte Leistungen beziehen, wobei die Diagnosen in der Regel die Indikationen für die Prozeduren darstellen. Im Grunde sind „Komplexe“ ihrerseits Kombinationen entweder aus den zuvor beschriebenen einfachen oder aus kombinierten Werten, jedoch mit dem Unterschied, dass übergreifend Angaben der Kategorien „Diagnose“ und „Therapie“ gleichzeitig berücksichtigt werden. Die Option „ja, Tracheotomie“ des Feldes „respiratorische Insuffizienz“ (3.1.80.) stellt ein Beispiel für ein solches Konstrukt dar.
- **„Normalwerte“:** das sind die Aussagen, die sich für Felder ergeben, deren Inhalt sich nach einem der 3 zuvor skizzierten Prinzipien ermitteln lassen, wenn die notwendigen Schlüssel – seien es einzelne oder Kombinationen von Codes – nicht Teil der für einen Fall dokumentierten Menge von Katalogeinträgen sind. So weit es sich um abzuleitende Angaben mit Bezug zu Diagnosen handelt, ist der Normalwert mit der Abwesenheit (dem „Nicht-Vorliegen“) einer Erkrankung des beschriebenen Formenkreises gleichzusetzen. Vergleichbar gilt für Felder mit Bezug zu

(therapeutischen) Maßnahmen, dass der Normalwert gleichbedeutend mit der nicht erfolgten Durchführung ist.

Nachfolgend werden die für die „Ableitung“ erforderlichen Datenstrukturen sowie das eigentliche Verfahren im engeren Sinne am Beispiel von Datenbank-Abfragen vorgestellt.

### **3.5.2. Ableitung von einfachen Werten**

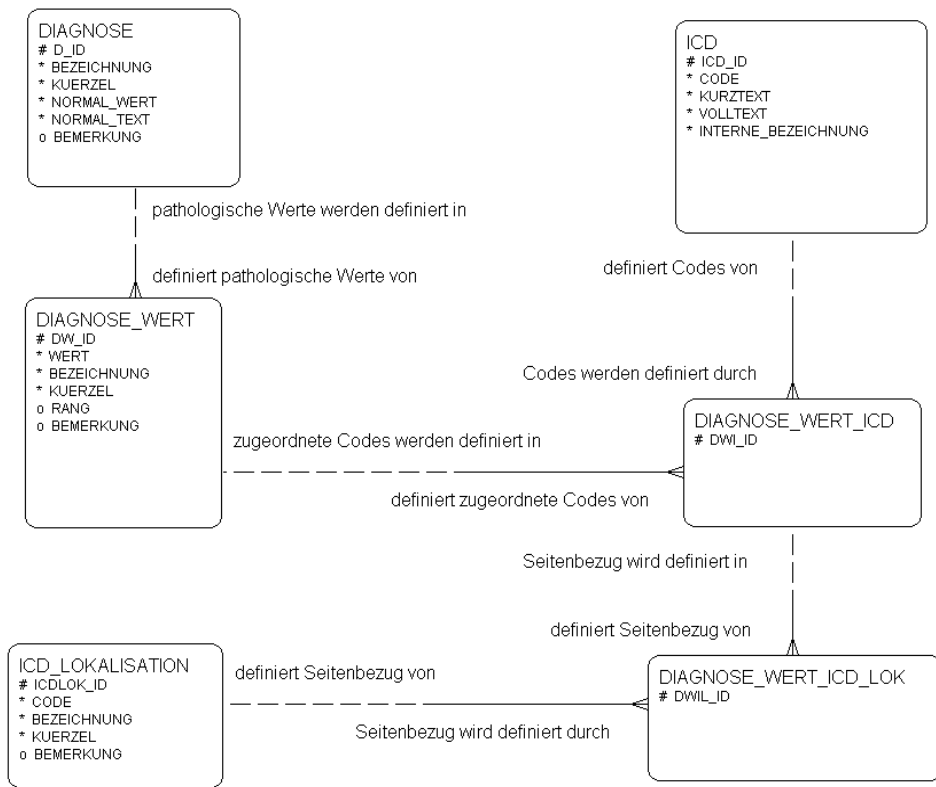
Den Diagnosen (Tabelle [DIAGNOSE], Abbildung 15) und Prozeduren (Tabelle [PROZEDUR], Abbildung 16) werden ihre zulässigen Werte in den Tabellen [DIAGNOSE\_WERT] bzw. [PROZEDUR\_WERT] zugeordnet.

Die Schlüssel, die für diese Ausprägungen codieren, finden sich in den Tabellen [DIAGNOSE\_WERT\_ICD] bzw. [PROZEDUR\_WERT\_OPS].

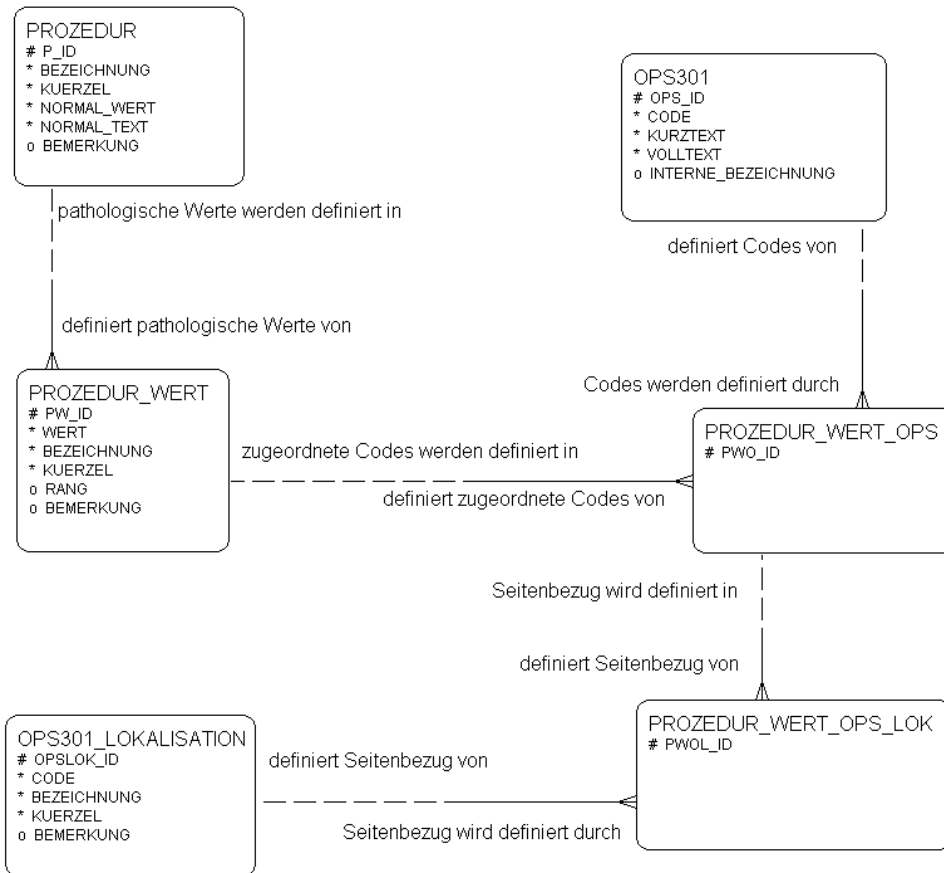
Sofern ein bestimmter Wert – wie beispielsweise im Fall der Option „ja, links“ für das Feld „TEA“ (3.1.63.) - eine Beschränkung auf eine Seiten-Lokalisation erfordert, kann diese durch einen Eintrag in den Tabellen [DIAGNOSE\_WERT\_ICD\_LOK] oder [PROZEDUR\_WERT\_OPS\_LOK] spezifiziert werden.

Durch die Angabe geeigneter Werte in den Feldern namens [RANG] in den Tabellen [DIAGNOSE\_WERT] und [PROZEDUR\_WERT] besteht die Möglichkeit, festzulegen, welchem Wert der Vorzug zu geben ist, wenn sich aus den dokumentierten Schlüsseln zwei oder mehr konkurrierende Werte für ein diagnose- oder prozedurbezogenes Feld ermitteln lassen – in diesen Fällen kann eine darauf zielende Abfrage denjenigen Eintrag in [DIAGNOSE\_WERT] bzw. [PROZEDUR\_WERT] wählen, der den jeweils niedrigsten Wert für [RANG] aufweist.

Die „Normalwerte“ für Diagnosen und Prozeduren (Nicht-Vorliegen einer Erkrankung bzw. Nicht-Durchführung einer Maßnahme) werden in den Tabellen [DIAGNOSE] und [PROZEDUR] durch die mit [NORMAL\_WERT] bzw. [NORMAL\_TEXT] bezeichneten Felder definiert.



**Abbildung 15: ER-Modell Ableitung von „einfachen“ Diagnose-Werten**



## Abbildung 16: ER-Modell Ableitung von „einfachen“ Prozedur-Werten

### 3.5.3. Ableitung von Kombinationswerten

Die zulässigen Optionen für manche Felder basieren auf der kombinierten Existenz von Codes in der Menge aller Schlüssel eines Falles, die differierenden „einfachen“ Werten (Tabellen [DIAGNOSE\_WERT\_ICD] bzw. [PROZEDUR\_WERT\_OPS]; 3.5.2.) zugeordnet sind.

Anders ausgedrückt: diese Einträge des ICD- oder OPS-Katalogs beziehen sich auf verschiedene, voneinander abgrenzbare Erkrankungs- oder prozedurale Entitäten, die jedoch gemeinsam angegeben worden sein müssen, um daraus den „Kombinationswert“ für das „betroffene“ Feld ableiten zu können.

Das Feld „neurologische Erkrankung(en)“ (3.1.32.) illustriert dieses Prinzip: neben den einfachen Werten „ja, ZNS, zerebrovaskulär (Blutung, Ischämie)“, „ja, ZNS, andere“ und „ja, peripher“ bildet die Ausprägung „ja, Kombination“ eine weitere zulässige Option, die sich auf pathologische Angaben bezieht. Die Voraussetzung für die Ableitung der Aussage „ja, Kombination“ besteht in der gleichzeitigen Angabe von mindestens 2 der 3 zuerst genannten, einfachen Werten (ja, ZNS, zerebrovaskulär (Blutung, Ischämie) / „ja, ZNS, andere“ / „ja, peripher“).

Kombinationen der beschriebenen Art finden im Datenmodell ihren Niederschlag in Gestalt der Tabellen [DIAGNOSE\_WERT\_KOMBINATION] (Abbildung 17) sowie [PROZEDUR\_WERT\_KOMBINATION] (Abbildung 18). Die einfachen Werte, die die Bestandteile der Kombinationen bilden, beziehen sich auf Einträge der Tabellen [DIAGNOSE\_WERT] oder [PROZEDUR\_WERT] und sind in den Tabellen [DIAG\_WERT\_KOMB\_DIAG\_WERT] bzw. [PROZ\_WERT\_KOMB\_PROZ\_WERT] definiert.

Durch Spezifikation des Typs von Kombinationen („mindestens 2 Elemente obligat“ vs. „alle Elemente obligat“) wird festgelegt, ob sich aus den ICD- bzw. OPS-Schlüsseln eines Falles ein Minimum von 2 oder aber alle der einer Kombination zugeordneten einfachen Werte ermitteln lassen müssen, damit das Mindestkriterium für die Herleitung des Kombinationswertes erfüllt ist (Tabellen [DIAGNOSE\_WERT\_KOMBINATIONSTYP] {Abbildung 17} bzw. [PROZEDUR\_WERT\_KOMBINATIONSTYP] {Abbildung 18}).

Im Fall des Beispiels „neurologische Erkrankung(en)“ wurde als Kombinationstyp „mindestens 2 Elemente obligat“ gewählt – das bedeutet, dass sich auf Basis der dokumentierten ICD-Codes mindestens 2 der 3 einfachen Werte (ja, ZNS, zerebrovaskulär (Blutung, Ischämie) / „ja, ZNS, andere“ / „ja, peripher“) ableiten lassen müssen, um zur Aussage „ja, Kombination“ als resultierende Angabe zu gelangen.

Die Kombinationswerte stehen in derselben Beziehung zu den Diagnosen bzw. Prozeduren wie ihre „einfachen“ Gegenstücke. Alleine die Tatsache, dass Kombinationen ihrerseits auf mehreren einfachen Werten basieren, impliziert, dass für die zugehörige Diagnose oder Prozedur die Zuweisung verschiedener Optionen möglich ist – wie bereits unter 3.5.2. erwähnt kann die definitive Auswahl des korrekten Inhalts („Welcher konkurrierende einfache oder Kombinationswert hat Priorität?“) über den Wert des auch hier implementierten Feldes [RANG] gesteuert werden.



Wie diese Rang-determinierte Steuerung in der Abfragesprache SQL formuliert wird, ist Gegenstand des Abschnitts 3.5.7.

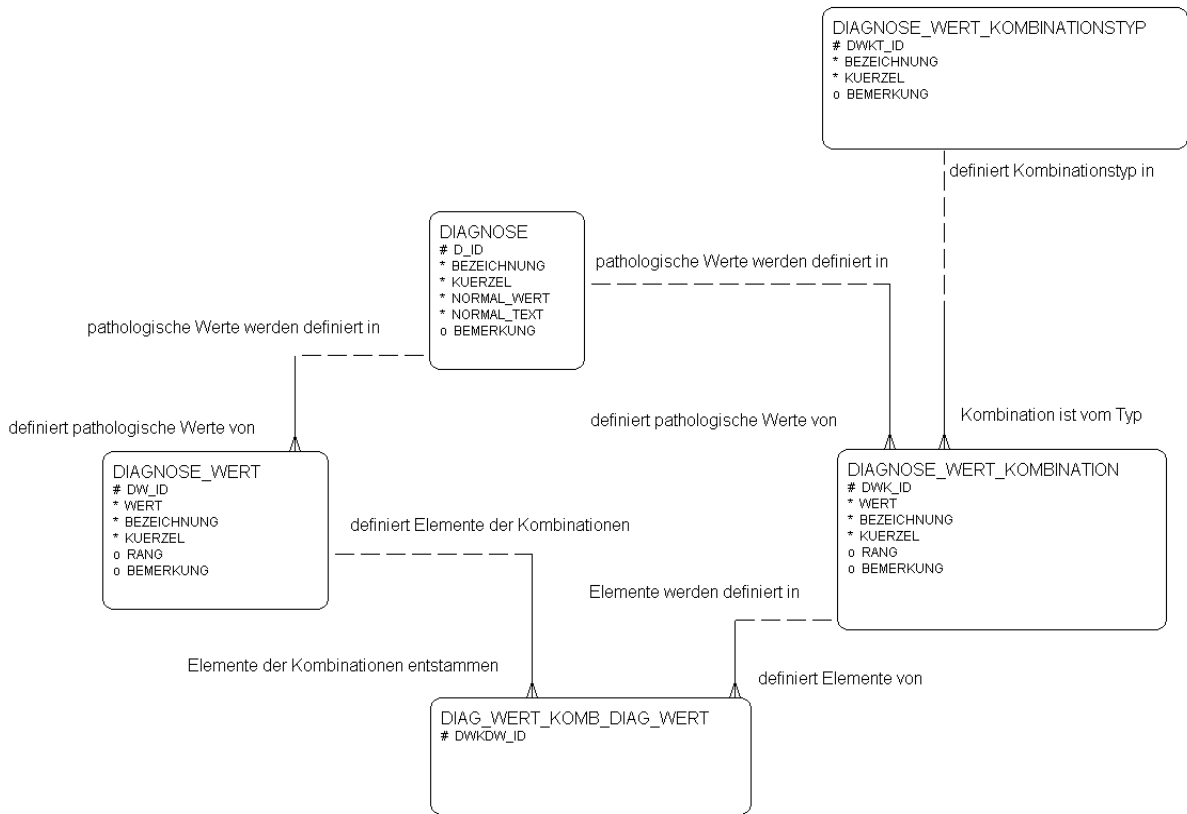
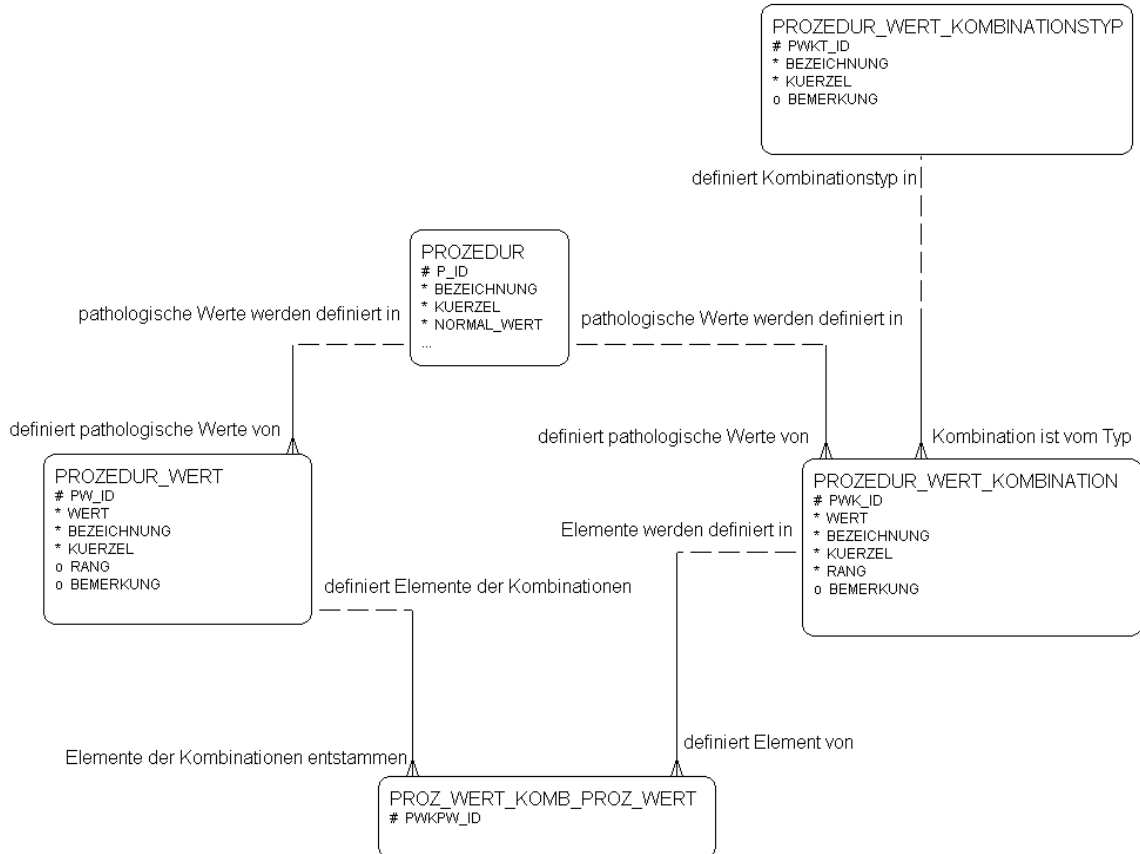


Abbildung 17: ER-Modell Ableitung von Diagnose-Kombinationswerten



## Abbildung 18: ER-Modell Ableitung von Prozedur-Kombinationswerten

### 3.5.4. Ableitung von Werten aus Komplexen

Die „dritte Säule der Ableitung“ bilden die „Komplexe“ getauften Konstrukte, derer sich die Methode zur Bestimmung derjenigen Feldinhalte bedient, die sich auf das mit der Durchführung einer therapeutischen Maßnahme assoziierte Vorliegen einer Erkrankung (bei der es sich beispielsweise um eine Komplikation handeln kann) beziehen.

Anders ausgedrückt: Komplexe dienen der Abbildung einer durch eine Option eines Feldes definierten Verknüpfung von Diagnose(n) und Prozedur(en).

Den konkreten, in der Tabelle [KOMPLEX\_WERT] (Abbildung 19, Abbildung 20) beschriebenen Werten eines Komplexes sind daher Ausprägungen von Diagnosen (Abbildung 19) und Prozeduren (Abbildung 20) zugeordnet, die ihrerseits einfache (Tabellen [KOMPLEX\_WERT\_D\_WERT], Abbildung 19, bzw. [KOMPLEX\_WERT\_P\_WERT], Abbildung 20) oder kombinierte Werte sein können (Tabellen [KOMPLEX\_WERT\_D\_WERT\_KOMB], Abbildung 19 bzw. [[KOMPLEX\_WERT\_P\_WERT\_KOMB], Abbildung 20).

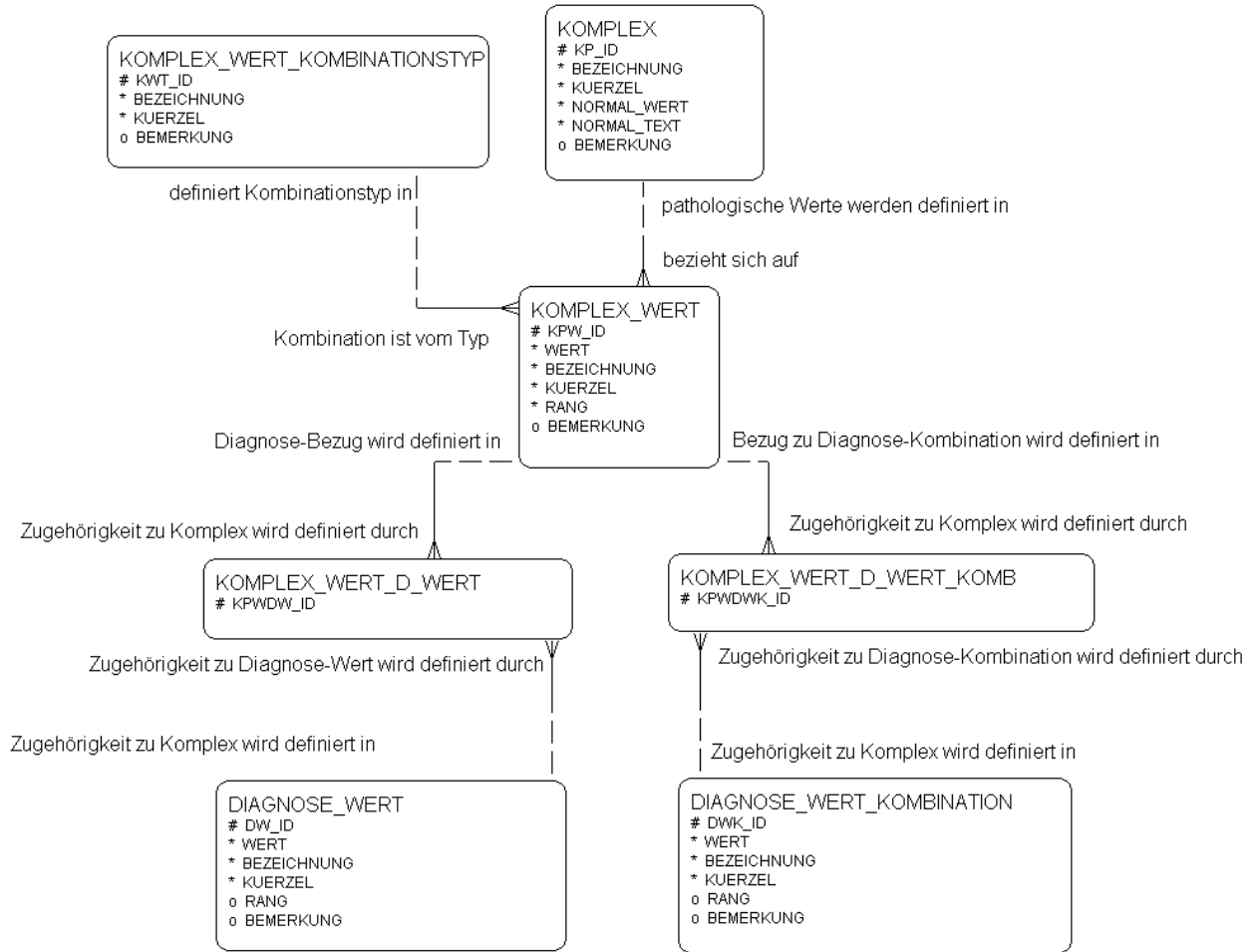
Durch Spezifikation des Kombinationstyps ([KOMPLEX\_WERT\_KOMBINATIONSTYP], Abbildung 19) für einen Eintrag kann bestimmt werden, welche Elemente (einfache oder kombinierte Diagnose- bzw. Prozedur-Werte) Teil der Daten eines Falls sein müssen, um den Wert des betreffenden Eintrags der Tabelle [KOMPLEX\_WERT] ableiten zu können.

Die beiden Typen „mindestens 1 Element“ und „mindestens 1 Diagnose + 1 Prozedur“ reichen aus, um alle denkbaren Konstellation abzubilden - vor allem im Hinblick auf solche Felder, bei denen lediglich ein Teil der zulässigen Optionen auf dem zwingend erforderlichen gleichzeitigen Vorliegen von diagnostischen und therapeutischen Komponenten basiert. Im Gegensatz zur Ableitung der Kombinationswerte (3.5.3.) wird keine Option der Art „alle Elemente sind Pflicht“ benötigt, da sich Komplexe ihrerseits auf kombinierte Werte beziehen können, die durch diese Eigenschaft charakterisiert sind. Durch entsprechend spezifizierte Kombinationswerte kann also der obligate Einschluss von mehr als 2 „einfachen“ Diagnose- oder Prozedur-Werten als Voraussetzung für die Ableitbarkeit eines Komplex-Wertes realisiert werden.

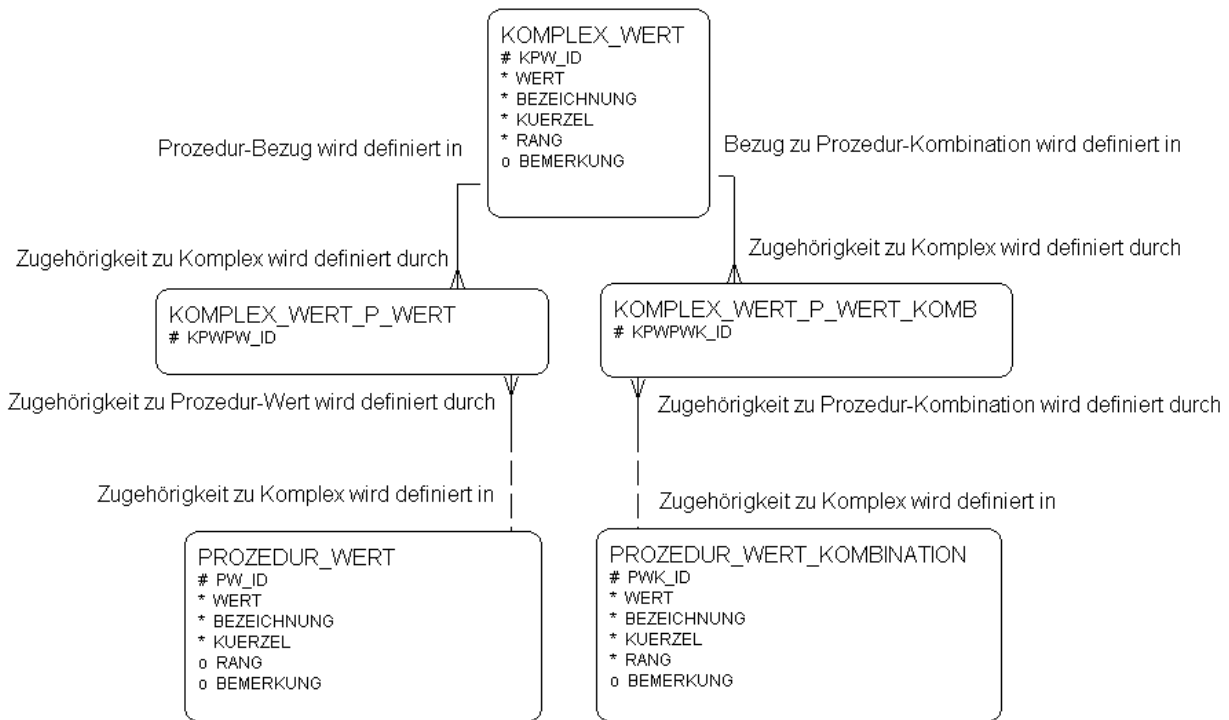
Als Beispiel zur Verdeutlichung dieser Zusammenhänge soll das Feld „Respiratorische Insuffizienz“ (3.1.80.) dienen, welches im Datenmodell als Komplex Berücksichtigung findet. Als mögliche Optionen stehen die Werte „[0] nein“, „[1] ja, mit forcierter Atemtherapie“, „[2] ja, Reintubation“ und „[3] ja, Tracheotomie“ zur Verfügung. Während [2] und [3] die gleichzeitige Existenz von für die respiratorische Insuffizienz codierenden ICD-Schlüsseln neben den Einträgen des OPS-Katalogs für die Maßnahmen „Reintubation“ und „Tracheotomie“ erfordern, lässt sich [1] alleine aus der Tatsache des Vorliegens einer respiratorischen Insuffizienz ableiten - und zwar dann, wenn keine Reintubation oder Tracheotomie durchgeführt wurde (dabei ist die durch den Wert des Feldes [RANG] in [KOMPLEX\_WERT] determinierte Reihenfolge der Auswertung bedeutsam: 3.5.2., 3.5.3.). Daraus ergibt sich, dass die Einträge für „ja, Reintubation“ und „ja, Tracheotomie“ in der Tabelle [KOMPLEX\_WERT] vom Typ „mindestens 1 Diagnose + 1 Prozedur“ sind, während „ja, mit forcierter Atemtherapie“ als Typ „mindestens 1 Element“ klassifiziert ist,

sich also letztlich nur auf einen einfachen Diagnose-Wert stützt..

Die Tabelle [KOMPLEX] ähnelt strukturell den Tabellen [DIAGNOSE] (3.5.2., Abbildung 15) bzw. [PROZEDUR] (3.5.2., Abbildung 16) – vor allem im Hinblick auf die Spezifikation des „Normalwertes“, der durch das Fehlen zugeordneter Diagnosen und/oder Prozeduren gekennzeichnet ist (im obigen Beispiel „Respiratorische Insuffizienz: „[0] nein“).



**Abbildung 19: ER-Modell Diagnose-Bezug von Komplexen**



**Abbildung 20: ER-Modell Prozedur-Bezug von Komplexen**

### 3.5.5. Kontext-Zuordnung

Die Einträge der Tabellen [DIAGNOSE], [PROZEDUR], [KOMPLEX] und [WERT] entsprechen jeweils einem Feld des Datensatzes Herzchirurgie (3.3., 3.5.2., 3.5.3, 3.5.4.).

So genannte „Kontexte“ (repräsentiert durch die Tupel der Tabelle [KONTEXT], Abbildung 21) dienen der Gliederung bzw. Einordnung der „verteilten“ Elemente des Moduls HCH.

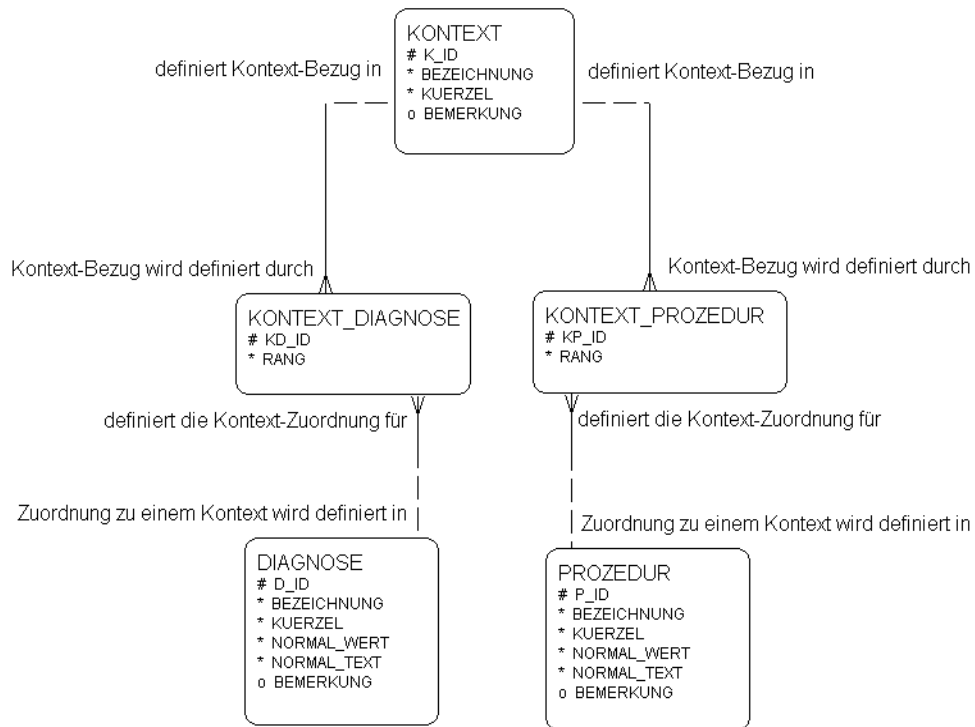
Die Beziehungen zwischen Kontexten auf der einen und Diagnosen, Prozeduren, Komplexen bzw. codeunabhängigen Werten auf der anderen Seite werden in den Tabellen [KONTEXT\_DIAGNOSE] (Abbildung 21), [KONTEXT\_PROZEDUR] (Abbildung 21), [KONTEXT\_KOMPLEX] (Abbildung 22) und [KONTEXT\_WERT] (Abbildung 22) hinterlegt.

Auf diese Weise kann durch einen Eintrag in der Tabelle [KONTEXT\_DIAGNOSE] beispielsweise definiert werden, dass das diagnosebezogene Feld „arterielle Gefäßerkrankung“ (3.1.26.) zum Kontext „präoperativer Teil des Datensatzes Herzchirurgie“ gehört. Zusätzlich besteht die Möglichkeit, die Position der Elemente innerhalb des Moduls mittels Zuweisung von Werten in den mit [RANG] bezeichneten Feldern zu beschreiben.

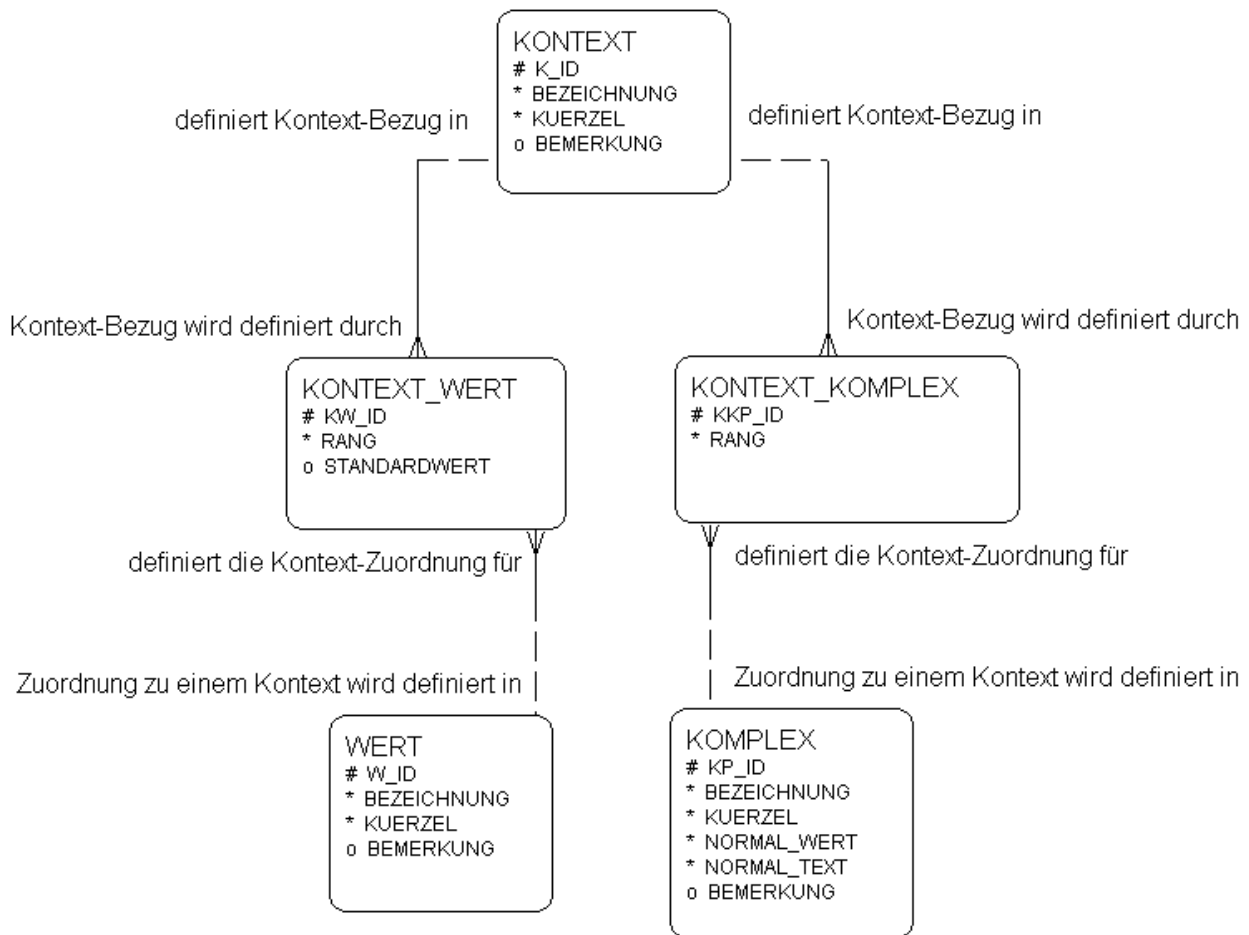
Kontexte können ihrerseits Strukturen namens Superkontexten zugeordnet werden (Tabellen [SUPERKONTEXT] und [SUPERKONTEXT\_KONTEXT], Abbildung 23). Im vorliegenden Fall ist das sinnvoll, weil der „Datensatz Herzchirurgie“ aus den genau einmal je Behandlung zu erfassenden prä- bzw. postoperativen Teilen sowie dem intraoperativen Abschnitt besteht, der unter Umständen – abhängig von der Zahl der durchgeführten Eingriffe – mehrfach angelegt werden muss. Als Superkontext ist also der „Datensatz Herzchirurgie“ als ganzes zu bezeichnen, während prä-, intra- und

postoperativer Teil als Kontexte Berücksichtigung finden.

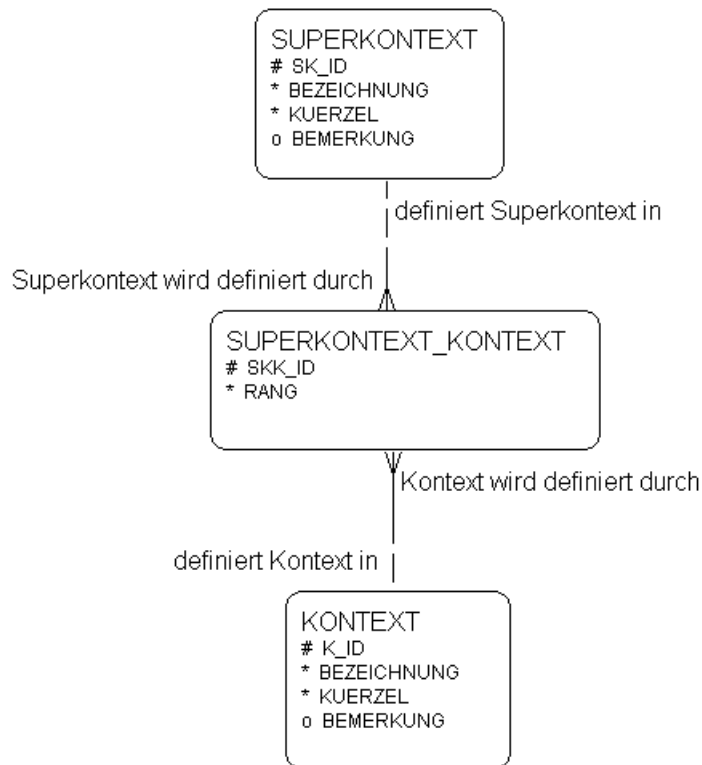
Wie bereits einleitend erwähnt wurde, soll das Datenmodell so flexibel gestaltet sein, dass seine Anwendung u.a. auch auf die Qualitätssicherungsmaßnahmen anderer Fachgebiete möglich wird. Die Spezifikation von Superkontexten und Kontexten – analog zum „Modul HCH“ - bildet gemeinsam mit der Zuweisung von Feldern unterschiedlicher Herkunft zu diesen ein wichtiges Instrument zur Realisation dieses Ziels.



**Abbildung 21: ER-Modell Kontext-Bezug von Diagnosen und Prozeduren**



**Abbildung 22: ER-Modell Kontext-Bezug von Komplexen und codeunabhängigen Werten**



**Abbildung 23: ER-Modell Superkontext-Bezug von Kontexten**

### 3.5.6. Phasen-Zuordnung

Neben den intraoperativen Diagnosen und Leistungen werden innerhalb des „Moduls HCH“ vor allem Betrachtungen angestellt, die sich auf bereits vor Beginn der Behandlung bestehende sowie im Laufe des Aufenthaltes neu auftretende Erkrankungen erstrecken. Um Aussagen aus ICD- und OPS-Schlüsseln herleiten zu können, sind somit zusätzliche Informationen erforderlich, die eine zeitliche Einordnung der codierten Angaben erlauben.

Im Hinblick auf diese Bedingung ist es hilfreich, eine Gliederung der Behandlung in verschiedene Phasen vorzunehmen, die durch Einträge der Tabelle [PHASE] (Abbildungen 24 und 25) repräsentiert werden. Es können grob folgende Abschnitte eines stationären Aufenthaltes unterschieden werden: Aufnahme mit der sich daran anschließenden präoperativen Behandlung, Haupteingriff und postoperativer Verlauf (zum letztgenannten zählen ggf. durchgeführte Folgeeingriffe).

Während im Fall von operationsbezogenen ICD- bzw. von OPS-Schlüsseln im Allgemeinen die Zuweisung zu einem der zuvor genannten Abschnitte implizit erfolgt, muss im Fall aller anderen Diagnosen eine Zuordnung zu einer der Phasen „Aufnahme“ (alle Erkrankungen, die zu Behandlungsbeginn vorlagen), „Behandlung“ (alle Erkrankungen, die bei Behandlungsbeginn vorlagen **oder** im Laufe des Aufenthaltes neu auftreten) oder „postoperativ“ (ausschließlich im Laufe der Behandlung neu aufgetretene Erkrankungen) erfolgen.

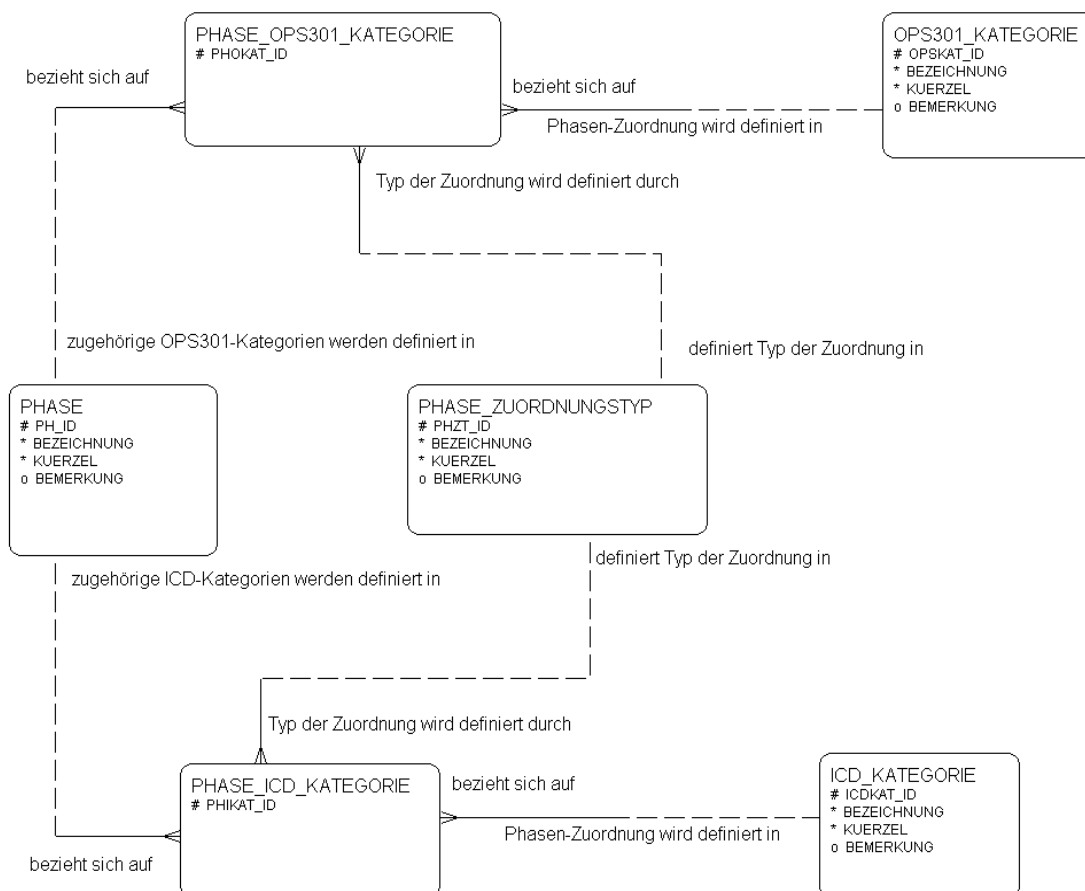
Um beispielsweise die im postoperativen Teil des Datensatzes gestellte Frage nach einem „zerebrovaskulären Ereignis bis zur Entlassung“ (3.1.81.) zu beantworten, darf eine dafür

berücksichtigte Diagnose nicht bereits Teil der Aufnahmediagnosen des Falles sein, die der Phase „Aufnahme“ zugeordnet werden. Ebenso muss der Zeitpunkt, an dem die Diagnose gestellt wurde, chronologisch hinter dem Beginn der Behandlung liegen.

Die – je nach KIS unterschiedlich bezeichneten – Kategorien von ICD- und OPS-Codes werden in den Tabellen [PHASE\_OPS301\_KATEGORIE] und [PHASE\_ICD\_KATEGORIE] (Abbildung 24) den einzelnen Phasen zugeordnet. Über die Angabe des durch Datensätze der Tabelle [PHASE\_ZUORDNUNGSTYP] definierten Typs kann bestimmt werden, ob eine Kategorie von Schlüsseln einer Phase zugewiesen werden soll (Beispiel: ICD-Schlüssel der Kategorie „Aufnahmediagnose“ sollen der Phase „Aufnahme“ zugerechnet werden) oder im Gegenteil zu einem Ausschluss der Entität (Diagnose oder Prozedur) führt (Beispiel: ein Schlüssel der Kategorie „Aufnahmediagnose“ führt dazu, dass derselbe Schlüssel nicht mehr der Phase „postoperativ“ zugeschlagen werden kann – relevant beispielsweise für das bereits zuvor erwähnte Beispiel „zerebrovaskuläres Ereignis bis zur Entlassung“ (3.1.81.)).

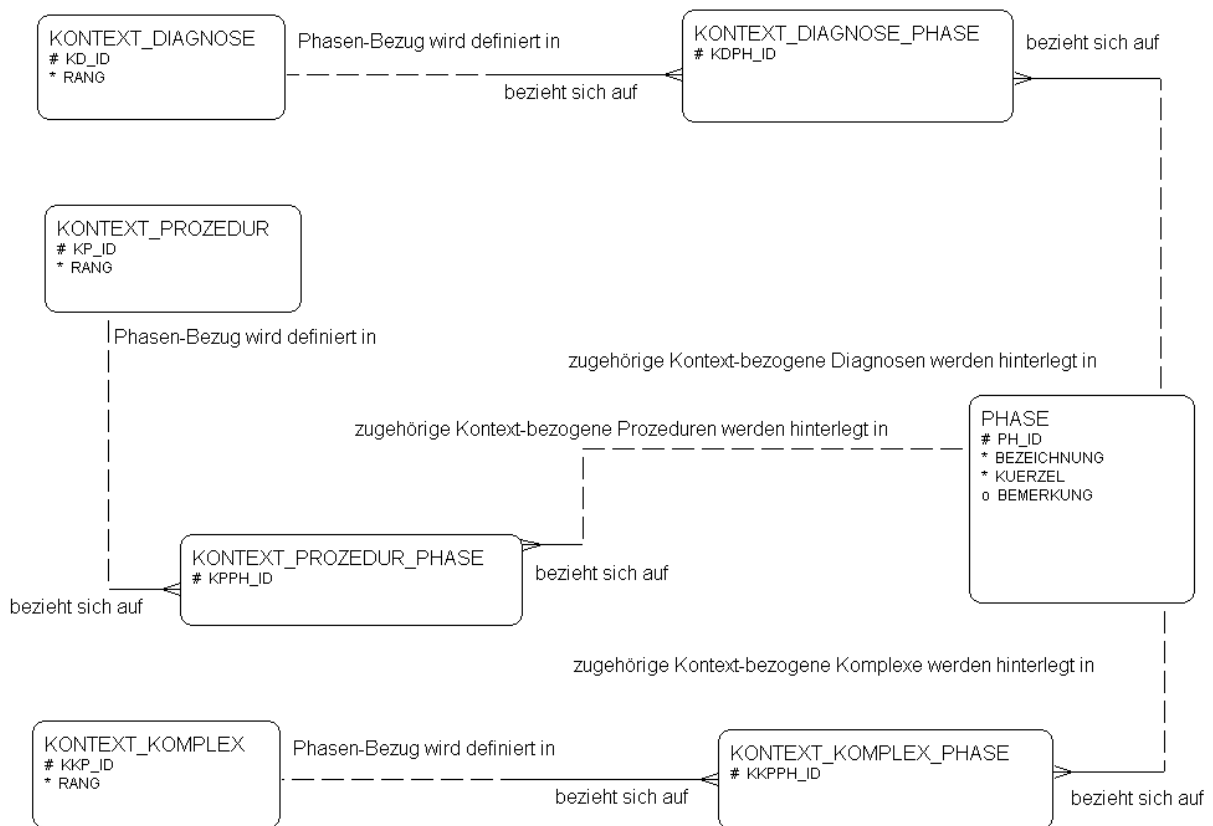
Für im Datenmodell kontextabhängig (3.5.5.) als Diagnosen, Prozeduren oder Komplexe klassifizierte Felder wird die Zuordnung zu einer der Phasen in den Tabellen [KONTEXT\_DIAGNOSE\_PHASE], [KONTEXT\_PROZEDUR\_PHASE] oder [KONTEXT\_KOMPLEX\_PHASE] definiert (Abbildung 25).

Auf welche Weise die Informationen bezüglich des Phasenbezugs innerhalb von Abfragen konkret Berücksichtigung finden, wird im Abschnitt 3.5.7. dargestellt.



**Abbildung 24: ER-Modell Phasen-Zuordnung von ICD-/OPS-Kategorien zu Phasen**





**Abbildung 25: ER-Modell Phasen-Zuordnung von Code-basierten Werten zu Phasen**

### 3.5.7. Abfragen zur Ermittlung abgeleiteter Werte

Die Abfragen, mit deren Hilfe die aus den ICD- bzw. OPS-Schlüssel ableitbaren Daten gewonnen werden, wurden – wie bereits unter Punkt 2.5. erwähnt - in Form von Sichten („Views“) permanent in der Datenbank abgelegt. Auf diese Weise kann die Komplexität der aufeinander aufbauenden Abfragen reduziert werden, da ohne diesen Mechanismus die in SQL formulierten Anweisungen bereits nach wenigen der nachfolgend beschriebenen Schritten einen unüberschaubaren und nicht mehr mit vertretbarem Aufwand künftig fortzuentwickelnden Umfang annehmen würden.

Selbst unter Nutzung von Sichten („Views“) als Abstraktions- bzw. Ordnungselement erreichen die SQL-Anweisungen Volumina, die eine adäquate Integration in den Haupttext dieser Arbeit nicht zulassen. Die Quellen wurden daher in den Anhang 4 verlagert, während in den nun folgenden Abschnitten jeweils auf die dortigen, mit Zeilennummern versehenen Texte verwiesen wird.

Detailliert wird die Herleitung von Diagnose- sowie von Komplex-bezogenen Daten dargestellt; da das sich auf prozedurale Schlüssel (OPS) erstreckende Verfahren auf denselben Prinzipien wie das ICD- bzw. Diagnose-gestützte basiert, wird es in deutlich kürzerer Form berücksichtigt (die dazugehörigen Abfragen finden sich jedoch ebenfalls vollständig im Anhang 4).

#### 3.5.7.1. ICD-Schlüssel eines Falles – Sicht [Fall\_ICD\_INKL\_OP]

Die Abfrage, die die Grundlage dieser Sicht bildet, liefert alle ICD-Schlüssel eines Falles unter Einschluss der operationsbezogenen Codes zurück. Die Angaben zu den ICD-Katalogeinträgen werden – von den allgemeinen Angaben zum Fall abgesehen - um Eigenschaften wie Kategorie, Datum und ggf. Lokalisation ergänzt, die für die nachfolgend beschriebenen Auswertungen von Bedeutung sind (Anhang 4, A4.1.).

#### 3.5.7.2. Einfache Diagnose-Werte - Sicht [FALL\_KONTEX\_DIAG\_WERT\_ICD]

Mit dieser Abfrage (Anhang 4, A4.2.) werden die einfachen, kontextbezogenen (3.5.5.) Diagnose-Werte (3.5.2.) eines Falles aus seinen ICD-Schlüsseln abgeleitet.

Dabei wird eine ggf. definierte, obligatorische Seitenlokalisierung ebenso berücksichtigt (ab Zeile 35) wie Ein- bzw. Ausschlüsse, die sich aus den Zusammenhängen zwischen ICD-Kategorie (ab Zeile 98), Zeitpunkt (ab Zeile 159) und Behandlungs-Phase (3.5.6.) ergeben.

Im Hinblick auf die spätere endgültige Bestimmung des Wertes einer Diagnose ist von Bedeutung, dass diese Sicht durchaus mehrere **mögliche** Werte je Diagnose liefern kann, die sich in ihrem Rang unterscheiden und zusätzlich mit den Kombinationswerten „konkurrieren“ (3.5.3. bzw. 3.5.7.5.).

#### 3.5.7.3. Diagnose-Normal-Werte - Sicht [FALL\_KONTEXT\_DIAG\_WERT\_NON\_ICD]

Sofern keiner der einem (pathologischen) Einfach-Wert einer Diagnose zugewiesenen ICD-Schlüssel Teil der Daten eines Falles ist, nimmt diese Diagnose ihren (physiologischen) „Normalwert“ an, bei dem es sich in der Regel um die Abwesenheit der beschriebenen Erkrankung handelt (Anhang 4, A4.3.).

Damit stellen die diesem Kriterium entsprechenden Diagnosen eines Falles die komplementären Daten zu dem durch die Sicht [FALL\_KONTEX\_DIAG\_WERT\_ICD] repräsentierten Ergebnis dar (dortiges Kriterium: die den einfachen, pathologischen Diagnose-Werten zugeordneten ICD-Schlüssel müssen Teil der Daten eines Falles sein, 3.5.7.2.).

Dieser Sachverhalt findet seinen Ausdruck in dem Umstand, dass die Abfrage für die Diagnose-Normal-Werte sich gerade auf diejenigen Diagnosen bezieht, die nicht Elemente der durch die Sicht [FALL\_KONTEX\_DIAG\_WERT\_ICD] definierten Menge sind (Zeilen 46 – 57).

#### 3.5.7.4. Bestimmung der Zahl einfacher Diagnose-Werte je kombiniertem Wert - Sicht [DIAGNOSE\_WERT\_KOMB\_ELEM\_COUNT]

Um einer Diagnose ggf. einen Kombinationswert zuweisen zu können, muss geprüft werden, ob sich die notwendige Zahl von Einfach-Werten (3.5.2.), aus denen sich der kombinierte Wert zusammensetzt, aus der Menge der ICD-Schlüssel eines Falles ableiten lässt.

Diese hilfswise angelegte Sicht (Anhang 4, A4.4.) liefert für jeden Diagnose-Kombinationswert (3.5.3.) die Anzahl der einfachen Werte zurück, auf denen er basiert.

Sie – bzw. die ihr zugrunde liegende Abfrage – liefert Angaben, die für den nachfolgenden

Schritt relevant sind.

#### 3.5.7.5. Kombinierte Diagnose-Werte - Sicht [FALL\_KONTEXT\_DIAG\_WERT\_KOMB]

Die Abfrage, auf der diese View basiert (Anhang 4, A4.5.), liefert Kombinationswerte für Diagnosen zurück, sofern die für deren Zuweisung definierten Kriterien erfüllt sind (3.5.3.).

Neben der Existenz der sich aus den zugeordneten Schlüsseln innerhalb der Daten eines Falles herleitenden einfachen Diagnose-Werte muss auch deren jeweilige Mindestzahl „erreicht“ werden. Abhängig vom Typ der Kombination müssen sich wenigstens 2 (Zeilen 61 - 79) oder aber die Gesamtzahl (3.5.7.4. und Zeilen 41 – 60;) aller einzelnen, einfachen Werte aus den ICD-Codes ermitteln lassen.

Im Hinblick auf die spätere endgültige Bestimmung des Wertes einer Diagnose ist von Bedeutung, dass diese Sicht durchaus mehrere **mögliche** Kombinations-Werte je Diagnose liefern kann, die sich in ihrem Rang unterscheiden und zusätzlich mit den einfachen Werten „konkurrieren“ (3.5.2. bzw. 3.5.7.2).

#### 3.5.7.6. Alle Diagnose-Werte - Sicht [FALL\_KONTEXT\_DIAGNOSE\_RANG]

Das Ergebnis, das die Abfrage dieser Sicht (Anhang 4, A.4.6.) liefert, umfasst alle normalen (3.5.), einfachen (3.5.2.) und kombinierten Diagnose-Werte (3.5.3.), die sich aus den ICD-Schlüsseln eines Falles – bzw. aus deren Fehlen - ableiten lassen.

Da diese vereinigte Menge noch mehrere mögliche bzw. zulässige, normale (Zeilen 1 - 25), einfache (Zeilen 27 - 50) oder kombinierte Werte (Zeilen 52 - 75) je Diagnose enthalten kann, stellt sie lediglich einen Zwischenschritt auf dem Weg zur endgültigen Bestimmung des konkreten Wertes jeder Diagnose im nächsten Abschnitt (3.5.7.7.) dar.

#### 3.5.7.7. Konkrete Werte der Diagnosen – Sicht [FALL\_KONTEXT\_DIAGNOSE]

Die Abfrage, die das Fundament dieser Sicht bildet (Anhang 4, A4.7.), liefert die konkreten Werte der einzelnen Diagnosen der Fälle zurück, bei denen es sich um normale (physiologische) oder pathologische einfache bzw. kombinierte Werte handeln kann.

Sofern – gestützt auf die Auswertung der dokumentierten ICD-Schlüssel – mehrere einfache oder kombinierte pathologische Werte in Betracht kommen, wird die Alternative aus den Datensätzen der im vorhergehenden Abschnitt beschriebenen View [FALL\_KONTEXT\_DIAGNOSE\_RANG] gewählt, deren Wert im Feld [RANG] das jeweilige, auf die Gruppe der möglichen Ausprägungen bezogene Minimum darstellt (Zeilen 29 – 50; zur Bedeutung des Feldes [RANG]: 3.5.2., 3.5.3.).

#### 3.5.7.8. Ableitung von Prozedur-Werten

Die Ableitung von Prozedur-Werten folgt denselben Prinzipien, die zuvor für die Gewinnung von Daten aus Diagnosen bzw. ICD-Schlüsseln formuliert wurden. Auf eine (erneute) detaillierte Darstellung wird daher verzichtet.

Analog gelten die Aussagen folgender Abschnitte für die komplementären Abfragen bzw. Sichten:

Sicht / Abfrage	Quelltext	Referenz-Abschnitt
FALL_OPS_INKL_OP	A4.8.	3.5.7.1.
FALL_KONTEXT_PROZ_WERT_OPS	A4.9.	3.5.7.2.
FALL_KONTEXT_PROZ_WERT_NON_OPS	A4.10.	3.5.7.3.
PROZEDUR_WERT_KOMB_ELEM_COUNT	A4.11.	3.5.7.4.
FALL_KONTEXT_PROZ_WERT_KOMB	A4.12.	3.5.7.5.
FALL_KONTEXT_PROZEDUR_RANG	A4.13.	3.5.7.6.
FALL_KONTEXT_PROZEDUR	A4.14.	3.5.7.7.

### 3.5.7.9. Ableitung von Komplex-Werten

Das Verfahren zur Bestimmung von Werten auf der Grundlage von Komplexen ist in großen Teilen identisch mit demjenigen zur Ableitung der Diagnose- oder Prozedur-Werte.

Einfache und kombinierte Werte von Diagnosen bzw. Prozeduren, die ihrerseits die Basis der Komplexe bilden (3.5.4.), werden in voneinander getrennten Abfragen ermittelt, die die Bezüge zu den Komplexen abbilden und auf den bereits zuvor vorgestellten Prinzipien (3.5.7.2. - 3.5.7.8.) basieren.

Unterschiede ergeben sich in den Teilen der Auswertungen, die die definierten Kombinationen aus diagnostischen und prozeduralen Elementen berücksichtigen.

#### **3.5.7.9.1. Einfache Werte**

In den Abfragen der Sichten [FALL\_KONT\_KOMP\_DIAG\_WERT\_ICD] (Anhang 4, A4.15.) und [FALL\_KONT\_KOMP\_PROZ\_WERT\_OPS] (Anhang 4, A4.16.) werden einfache Werte von Diagnosen und Prozeduren aus den ICD- und OPS-Schlüsseln der Fälle abgeleitet, soweit sie mögliche Bestandteile von Komplexen sind (A4.15., Zeilen 38 – 48 bzw. A4.16., Zeilen 38 – 48). Von diesem Unterschied abgesehen, gelten die Ausführungen zu Punkt 3.5.7.2. analog.

#### **3.5.7.9.2. Kombinierte Werte**

Sofern Bezüge zu Komplexen existieren, werden einfache Werte, die selbst Bestandteile kombinierter Werte von Diagnosen und Prozeduren sind, als Ergebnis der Abfragen, die durch die Sichten [FALL\_KONT\_KOMP\_DWK\_DW\_ICD] (Anhang 4, A4.17.) und [FALL\_KONT\_KOMP\_PWK\_PW\_OPS] (Anhang 4, A4.18.) repräsentiert werden, zurückgeliefert (Bezug zu den Komplexen: jeweils in den Zeilen 49 - 67).

Gestützt auf diese beiden Mengen werden die konkreten Kombinationswerte durch die Abfragen der Sichten [FALL\_KONT\_KOMP\_DIAG\_WERT\_KOMB] (Anhang 4, A4.19.) und [FALL\_KONT\_KOMP\_PROZ\_WERT\_KOMB] (Anhang 4, A4.20.) ermittelt. Dabei kommt das bereits im Abschnitt 3.5.7.5. vorgestellte Verfahren zur Anwendung – jedoch insofern modifiziert, als die Zahl der einfachen Werte nicht separat (3.5.7.4.), sondern innerhalb der Anweisung selbst ermittelt wird (jeweils Zeilen 81 – 115).

### **3.5.7.9.3. Vereinigung der einfachen und kombinierten Werte**

Die einfachen und kombinierten Werte von Diagnosen und Prozeduren werden durch die Abfrage der Sicht [FALL\_KONT\_KOMP\_DW\_PW\_DWK\_PWK] (Anhang 4, A4.21.) vereinigt und mittels der Felder der Sicht [FALL\_KPW] (Anhang 4, A4.22.) um zusätzliche Informationen zur Sicht [FALL\_KONT\_KOMP\_W\_DW\_PW\_DWK\_PWK] (Anhang 4, A4.23.) ergänzt, in der die Zuordnung der Elemente zu den konkreten Ausprägungen der Komplexe realisiert wird (Zeile 44 – 65).

### **3.5.7.9.4. Zählung diagnostischer und prozeduraler Elemente**

Wie bereits zuvor erwähnt (3.5.4.) wurde, kann durch die Zuweisung eines des beiden Kombinationstypen „mindestens 1 Element“ oder „mindestens 1 Diagnose + 1 Prozedur“ für jeden Wert eines Komplexes definiert werden, welcher Art das Kriterium für die „Zulässigkeit“ seiner Ableitung ist.

Um diese Entscheidung treffen zu können, liefert die Sicht [FALL\_KONT\_KOMP\_WERT\_D\_P\_COUNT] (Anhang 4, A4.24.) für jede Ausprägung eines Komplexes die Zahlen der aus den ICD- und OPS-Schlüsseln eines Falles herleitbaren diagnostischen und prozeduralen Elemente zurück (Zeilen 24 und 25).

### **3.5.7.9.5. Zulässige Werte der Komplexe**

Durch Rückgriff auf die Angaben der im vorhergehenden Abschnitt beschriebenen Abfrage bzw. Sicht kann das durch den Kombinationstyp spezifizierte Kriterium in der SQL-Anweisung der Sicht [FALL\_KONTEXT\_KOMPLEX\_WERT\_RANG] (Anhang 4, A4.25.) geprüft (Zeilen 27 – 45) und so die Menge aller zulässigen Werte der Komplexe ermittelt werden.

### **3.5.7.9.6. Konkrete, schlüsselbasierte Werte der Komplexe**

Da die Menge aller zulässigen, schlüsselbasierten (ICD, OPS) Werte der Komplexe, die die zuvor beschriebene Sicht „liefert“, durchaus mehrere mögliche, alternative Werte je Komplex umfassen kann, besteht die Notwendigkeit, den jeweils konkret zu verwendenden „Non-Normal-Wert“ anhand seines (minimalen) Ranges zu ermitteln (analog zu den Ausführungen unter Punkt. 3.5.7.7.).

Dieser Schritt wird in der Abfrage der Sicht [FALL\_KONTEXT\_KOMPLEX\_MIN\_RANG] (Anhang 4, A4.26) vollzogen (Zeile 18).

### **3.5.7.9.7. Werte der Komplexe**

Die Abfrage der Sicht [FALL\_KONTEXT\_KOMPLEX] (Anhang 4, A4.27.) schließlich liefert die „tatsächlichen“ Werte der Komplexe zurück. Die Ergebnismenge der unmittelbar zuvor beschriebenen Sicht wird dabei um die „Normal-Werte“ derjenigen Komplexe ergänzt, für die sich keine „Non-Normal-Werte“ aus den ICD- bzw. OPS-Schlüsseln der Fälle herleiten lassen (Zeile 35 – 82).

### **3.5.7.10. Abgeleitete Werte der Operationen**

Das innerhalb der Abschnitte 3.5.7.1. - 3.5.7.9. beschriebenen Abfragen beziehen sich auf alle ICD- und OPS-Schlüssel eines Aufenthaltes.

Um Operationen zugeordnete Werte aus der diagnostischen bzw. prozeduralen Codierung

ableiten zu können, muss die Menge der dokumentierten Katalogeinträge auf diese beschränkt und um spezifische Angaben des jeweiligen Eingriffs ergänzt werden.

Von diesen Besonderheiten abgesehen, sind die OP-bezogenen Abfragen mit denjenigen, die die Gesamtheit der Schlüssel eines Falles berücksichtigen, identisch. Der nachfolgend dargestellten Tabelle können die Referenzen auf die Passagen entnommen werden, die sinngemäß auch für die eingriffsspezifischen Sichten gelten.

<b>Sicht bzw. Abfrage / (passendes Pendant)</b>	<b>Quelltext</b>	<b>Referenz-Abschnitt</b>
FALL_ICD_OP / (FALL_ICD_INKL_OP)	A4.28.	3.5.7.1.
FALL_KONTEXT_DIAG_WERT_ICD_OP / (FALL_KONTEXT_DIAG_WERT_ICD)	A4.29.	3.5.7.2.
FALL_KONTEXT_DIAG_W_NON_ICD_OP / (FALL_KONTEXT_DIAG_WERT_NON_ICD)	A4.30.	3.5.7.3.
FALL_KONTEXT_DIAG_WERT_KOMB_OP / (FALL_KONTEXT_DIAG_WERT_KOMB)	A4.31.	3.5.7.5.
FALL_KONTEXT_DIAGNOSE_RANG_OP / (FALL_KONTEXT_DIAGNOSE_RANG)	A4.32.	3.5.7.6.
FALL_KONTEXT_DIAGNOSE_OP / (FALL_KONTEXT_DIAGNOSE)	A4.33.	3.5.7.7.
FALL_OPS_OP / (FALL_OPS_INKL_OP)	A4.34.	3.5.7.1. 3.5.7.8.
FALL_KONTEXT_PROZ_WERT_OPS_OP / (FALL_KONTEXT_PROZ_WERT_OPS)	A4.35.	3.5.7.2. 3.5.7.8.
FALL_KONTEXT_PROZ_W_NON_OPS_OP / (FALL_KONTEXT_PROZ_WERT_NON_OPS)	A4.36.	3.5.7.3. 3.5.7.8.
FALL_KONTEXT_PROZ_WERT_KOMB_OP / (FALL_KONTEXT_PROZ_WERT_KOMB)	A4.37.	3.5.7.5. 3.5.7.8.
FALL_KONTEXT_PROZEDUR_RANG_OP / (FALL_KONTEXT_PROZEDUR_RANG)	A4.38.	3.5.7.6. 3.5.7.8.
FALL_KONTEXT_PROZEDUR_OP / (FALL_KONTEXT_PROZEDUR)	A4.39.	3.5.7.7. 3.5.7.8.
FALL_KONT_KOMP_D_WERT_ICD_OP / (FALL_KONT_KOMP_DIAG_WERT_ICD)	A4.40.	3.5.7.9.1.
FALL_KONT_KOMP_P_WERT_OPS_OP / (FALL_KONT_KOMP_PROZ_WERT_OPS)	A4.41.	3.5.7.9.1.
FALL_KONT_KOMP_DWK_DW_ICD_OP / (FALL_KONT_KOMP_DWK_DW_ICD)	A4.42.	3.5.7.9.2.
FALL_KONT_KOMP_PWK_PW_OPS_OP / (FALL_KONT_KOMP_PWK_PW_OPS)	A4.43.	3.5.7.9.2.
FALL_KONT_KOMP_D_WERT_KOMB_OP / (FALL_KONT_KOMP_DIAG_WERT_KOMB)	A4.44.	3.5.7.9.2.

FALL_KONT_KOMP_P_WERT_KOMB_OP / (FALL_KONT_KOMP_PROZ_WERT_KOMB)	A4.45.	3.5.7.9.2.
FALL_KT_KOMP_DW_PW_DWK_PWK_OP / (FALL_KONT_KOMP_DW_PW_DWK_PWK)	A4.46.	3.5.7.9.3.
FALL_KPW_OP / (FALL_KPW)	A4.47.	3.5.7.9.3.
FALL_KT_KP_W_DW_PW_DWK_PWK_OP / (FALL_KONT_KOMP_W_DW_PW_DWK_PWK)	A4.48.	3.5.7.9.3.
FALL_KT_KP_WERT_D_P_COUNT_OP / (FALL_KONTT_KOMP_WERT_D_P_COUNT)	A4.49.	3.5.7.9.4.
FALL_KONTEXT_KOMP_WERT_RANG_OP / (FALL_KONTEXT_KOMPLEX_WERT_RANG)	A4.50.	3.5.7.9.5.
FALL_KONTEXT_KOMP_MIN_RANG_OP / (FALL_KONTEXT_KOMPLEX_MIN_RANG)	A4.51.	3.5.7.9.6.
FALL_KONTEXT_KOMPLEX_OP / (FALL_KONTEXT_KOMPLEX)	A4.52.	3.5.7.9.7.

### 3.6. Plausibilitätsprüfung

Die Schaffung von Strukturen innerhalb des Datenmodells, die eine Prüfung der wechselseitigen Plausibilität von QS- und DRG-relevanten Daten erlauben, war eines der beiden ursprünglich in dieser Arbeit verfolgten Designziele.

Durch Einführung des Konzepts der „Elemente“ (3.6.3.), bei denen es sich **wahlweise** um Diagnosen, Prozeduren, Komplexe oder codeunabhängige Werte handeln kann, konnte das Prüfverfahren indes so flexibel gestaltet werden, dass die Überwachung einer großen Bandbreite von Regeln möglich wird.

So kann definiert werden, dass bestimmte Ausprägungen suspekt (im Sinne von „zulässig, jedoch möglicherweise falsch“) sind bzw. dass verschiedene Elemente ausschließlich gekoppelt existieren können oder sich im Gegenteil gegenseitig ausschließen. Dieses Thema wird in den Abschnitten 3.6.3. und 3.6.4. ausführlich dargestellt und anhand von Beispielen illustriert.

Zunächst werden aber 2 essentielle strukturelle Voraussetzungen für die Etablierung des „tabellengestützten“ Prüfmechanismus beschrieben (3.6.1., 3.6.2.).

#### 3.6.1. Pufferung von Daten

Die Prüfung der Plausibilität basiert auf Datenbank-Abfragen, mit deren Hilfe Aussagen zur Existenz oder eben dem Fehlen eines bestimmten Wertes gemacht werden. Diese Werte können ihrerseits in Form von Diagnosen, Prozeduren oder Komplexen (3.5.) das Ergebnis von SQL-Befehlen als Grundlage von Sichten (3.5.7.) sein, die die für einen Aufenthalt dokumentierten ICD- und OPS-Schlüssel auswerten.

Da viele Werte Gegenstand verschiedener Regeln sind und somit innerhalb des Prüfverfahrens mehrfach berücksichtigt werden, ist es sinnvoll, diese unabhängig von ihrer

Herkunft (ICD- oder OPS-bezogen bzw. codeunabhängig) temporär abzulegen, um die Geschwindigkeit der im Rahmen der Plausibilitätsprüfung durchzuführenden Abfragen zu erhöhen bzw. deren Kosten zu verringern.

Als Zwischenspeicher dienen die Tabellen [PUFFER\_FALL\_KONTEXT] (Anhang 3, Zeilen 53-64) für alle Werte eines Aufenthaltes und [PUFFER\_FALL\_OP\_KONTEXT] (Anhang 3, Zeilen 65-78) für diejenigen, die sich auf Operationen beziehen. Ihr Aufbau kann den referenzierten Passagen des SQL-Skripts entnommen werden, mit dessen Hilfe das Datenmodell implementiert wird.

### 3.6.2. Einfache Zuordnung von Schlüsseln zu Diagnosen und Prozeduren

Gruppierte (Tabellen [DIAGNOSE\_ICD], Abbildung 26, und [PROZEDUR\_OPS301], Abbildung 27) und ggf. um die Definition einer Seitenlokalisierung ergänzte ICD- oder OPS301-Schlüssel (Tabellen [DIAGNOSE\_ICD\_LOK], Abbildung 26, und [PROZEDUR\_OPS301\_LOK], Abbildung 27) können als Diagnosen oder Prozeduren Konstrukte darstellen, die Gegenstand einer Plausibilitätsprüfung werden.

Dieses Prinzip ähnelt demjenigen, mit dessen Hilfe aus den Codes konkrete Werte abgeleitet werden (3.5.2.) - allerdings um die Tabellen reduziert, die der Vorhaltung der Werte im engeren Sinne dienen. Insofern ist es in struktureller Hinsicht weniger kompliziert und seine Inhalte auf diese Weise mit geringerem Aufwand zu pflegen. Im folgenden Abschnitt (3.6.3.) wird die Rolle dieser simplen Zuordnungen deutlich.

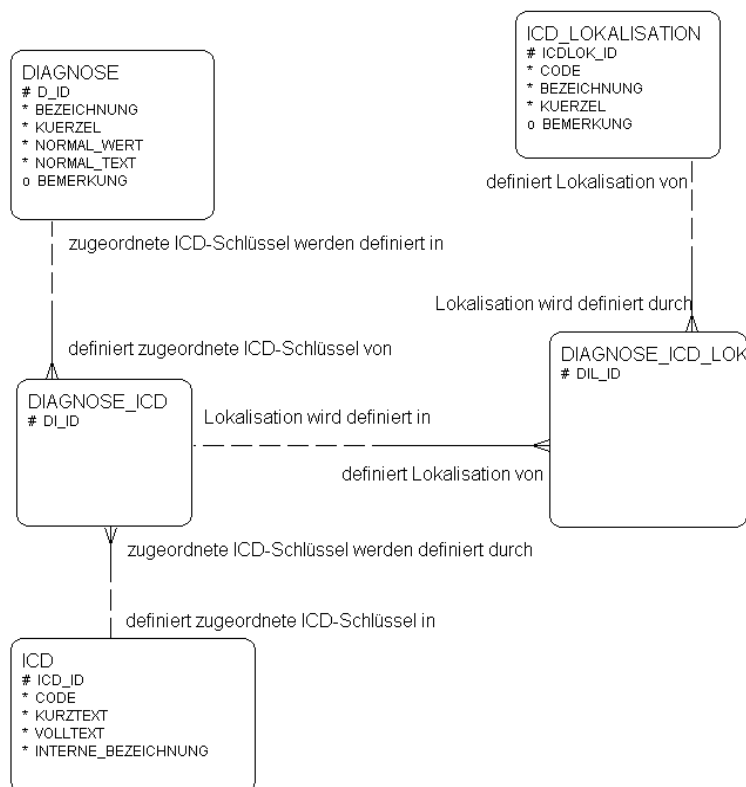
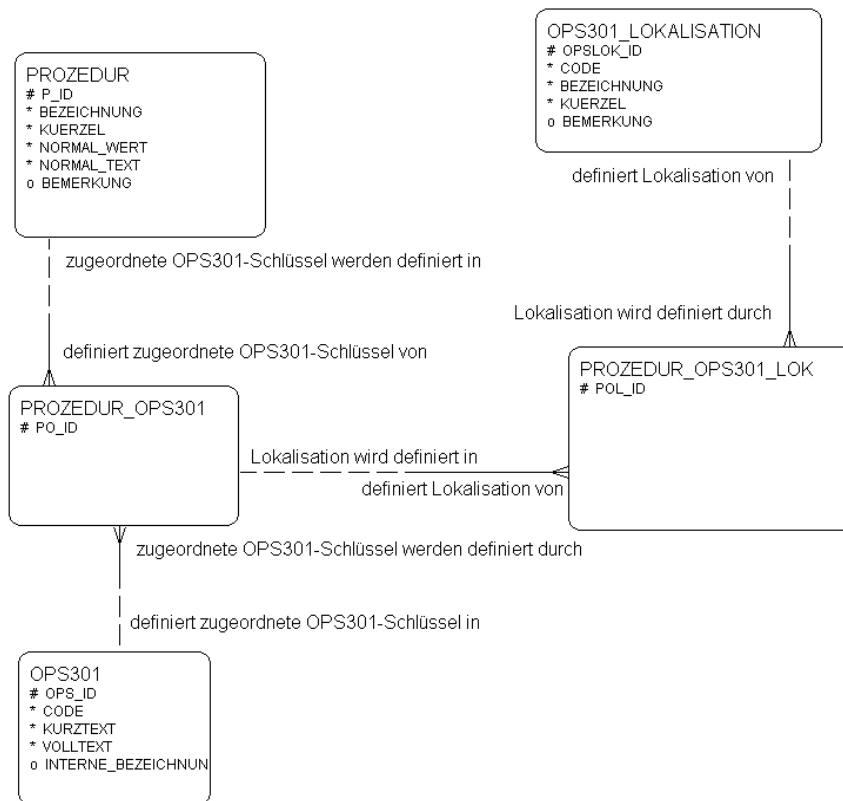


Abbildung 26: Zuordnung von ICD-Schlüsseln zu Diagnosen





**Abbildung 27: Zuordnung von OPS-Schlüsseln zu Prozeduren**

### 3.6.3. Elemente

Regeln, auf deren Einhaltung sich Plausibilitätsprüfungen erstrecken, beziehen sich auf Diagnosen, Prozeduren, Komplexe (3.5.) oder codeunabhängige Werte (3.3.).

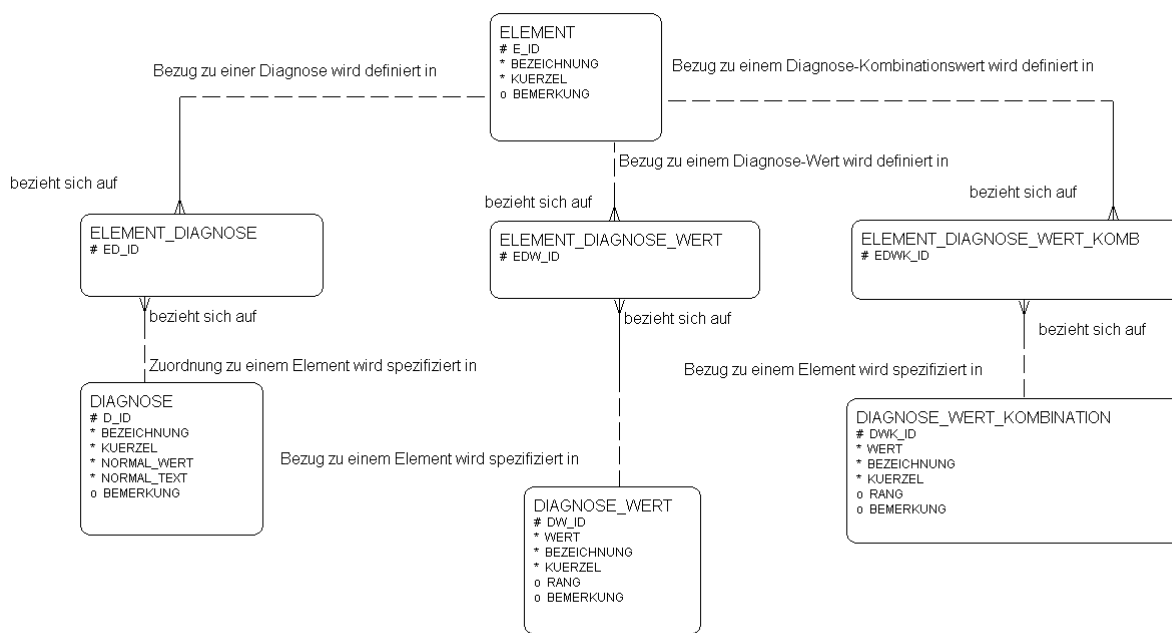
Um die Formulierung bzw. Hinterlegung von Regeln im Datenmodell ebenso einfach wie flexibel zu gestalten, bedarf es einer „Superstruktur“, die den Zugang einer Regel zu „ihren“ sich aus den 4 Komponenten (Diagnosen, Prozeduren, Komplexe, codeunabhängige Werte) rekrutierenden Objekten vereinheitlicht.

Dieses Konstrukt wird innerhalb des Datenmodells als Element bezeichnet; kurz gesagt: ein Element kann sowohl die Ausprägung einer Diagnose, einer Prozedur, eines Komplexes oder eines codeunabhängigen Wertes sein. Auf diese Weise können diese Entitäten trotz ihrer Verschiedenartigkeit miteinander verbunden werden.

Die Deklaration einfacher (3.5.2.) oder kombinierter (3.5.3.) Werte von Diagnosen als Elemente erfolgt in den Tabellen [ELEMENT\_DIAGNOSE\_WERT] bzw. [ELEMENT\_DIAGNOSE\_WERT\_KOMB] (Abbildung 28). Die Einträge der Tabelle [ELEMENT\_DIAGNOSE] definieren entweder Elemente, die sich auf die Normal-Werte von Diagnosen (3.5.) stützen oder alternativ zu „Diagnosen“ gruppierte Mengen von ICD-Schlüsseln repräsentieren (3.6.2.).

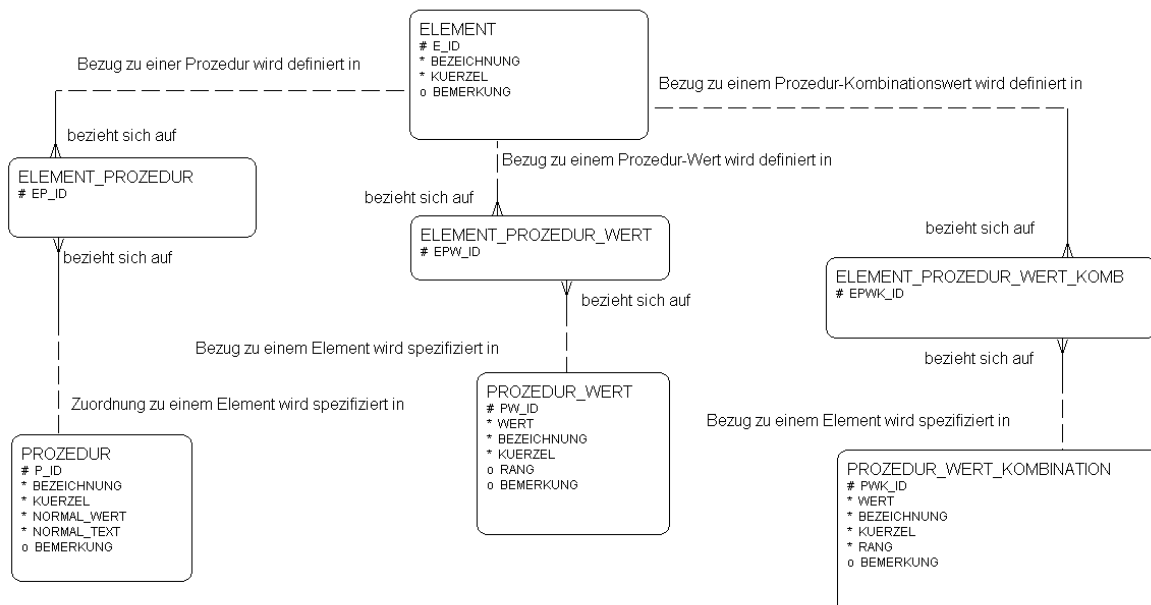
Die Abfragen, die die Basis der Sichten [PUFFER\_FALL\_ELEMENT] (Anhang 5, A5.1.) und [PUFFER\_FALL\_OP\_ELEMENT] (Anhang 5, A5.2.) bilden, liefern Fall- bzw. Eingriffsbezogene Angaben zu denjenigen Elementen zurück, deren Existenz sich aus den

temporär abgelegten Daten der Behandlungen (3.6.1.) herleiten lässt.



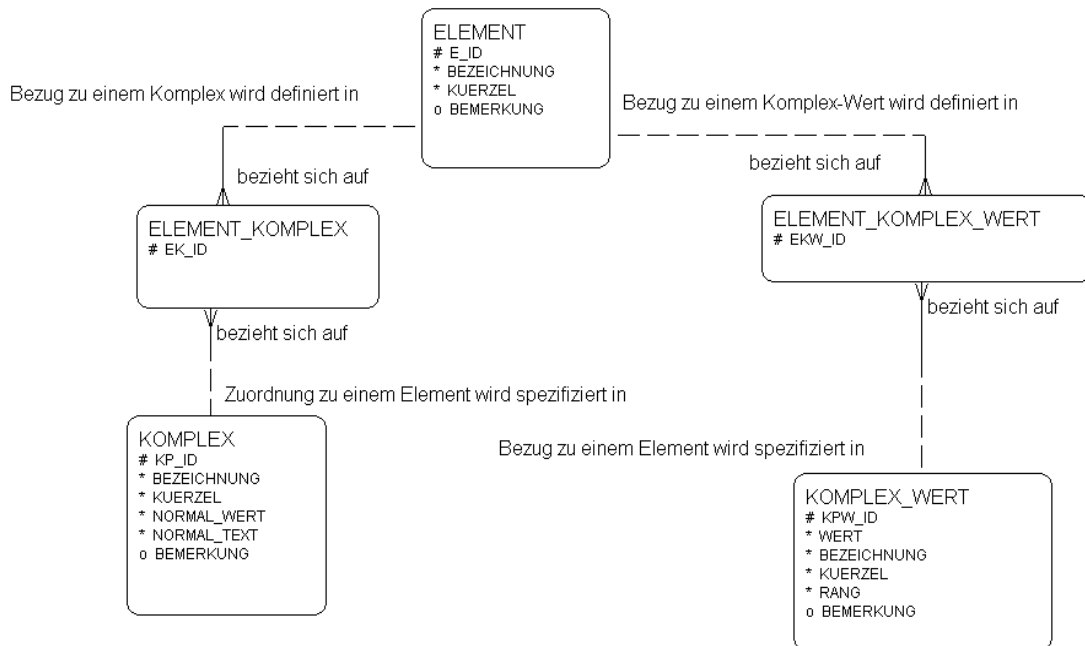
**Abbildung 28: ER-Modell Diagnose-bezogene Konstrukte als Elemente**

Analog erfolgt die Zuordnung einfacher bzw. kombinierter Werte von Prozeduren zu Elementen in den Tabellen [ELEMENT\_PROZEDUR\_WERT] bzw. [ELEMENT\_PROZEDUR\_WERT\_KOMB], während Normalwerte bzw. gruppierte OPS-Schlüssel Gegenstand der Datensätze der Tabelle [ELEMENT\_PROZEDUR] sind (Abbildung 29).



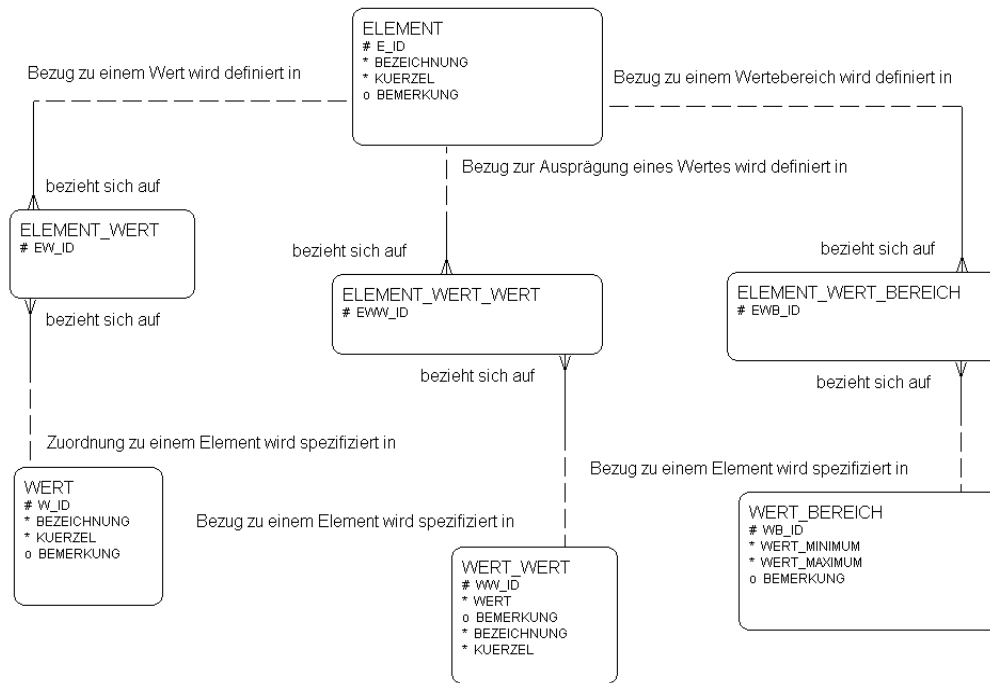
**Abbildung 29: ER-Modell Prozedur-bezogene Konstrukte als Elemente**

Die Bezüge zwischen den Ausprägungen von Komplexen (3.5.4.) einerseits und Elementen andererseits werden in der Tabelle [ELEMENT\_KOMPLEX\_WERT] abgebildet, während gleiches für die Normalwerte von Komplexen und die Tabelle [ELEMENT\_KOMPLEX] gilt (Abbildung 30).



**Abbildung 30: ER-Modell Komplex-bezogene Konstrukte als Elemente**

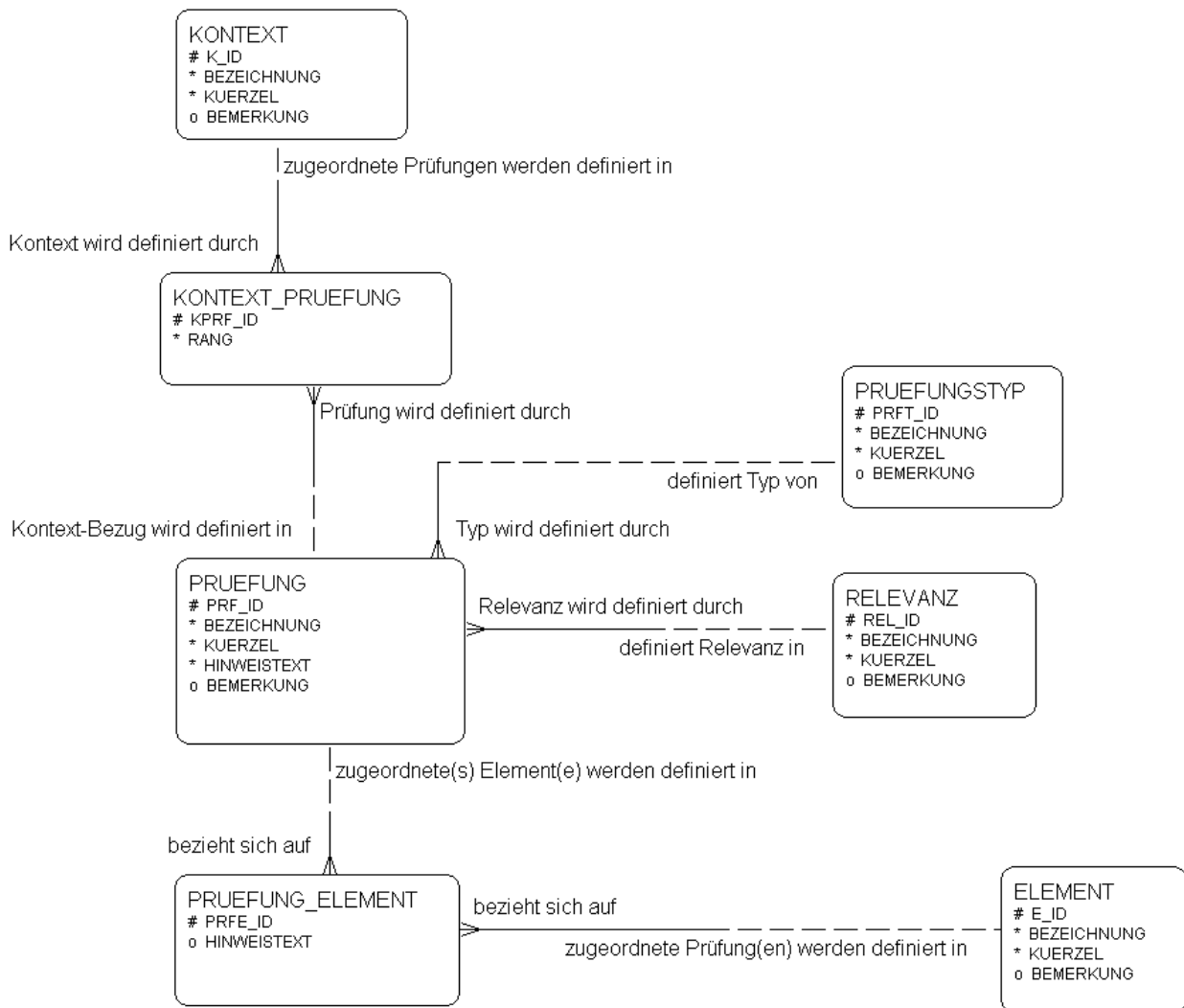
Der Bezug zwischen konkreten codeunabhängigen Werten (3.3.) und Elementen kann in der Tabelle [ELEMENT\_WERT\_WERT] hinterlegt werden (Abbildung 31). Stellen codeunabhängige Werte – wie bspw. Laborparameter - numerische Angaben dar, besteht die Möglichkeit, dafür definierte Bereiche (Intervalle) durch Einträge in der Tabelle [ELEMENT\_WERT\_BEREICH] Elementen zuzuordnen. Ist die schiere Existenz eines codeunabhängigen Wertes von Interesse, kann [ELEMENT\_WERT] verwendet werden.



**Abbildung 31: ER-Modell codeunabhängige Konstrukte als Elemente**

### 3.6.4. Prüfung von Regeln

Zentraler Bestandteil des durch das Datenmodell realisierbaren Prüfmechanismus ist die Tabelle [PRUEFUNG] (Abbildung 32), deren Einträge durch korrespondierende Datensätze der Tabelle [KONTEXT\_PRUEFUNG] einem Kontext (3.5.5.) zugeordnet werden können. Aus den Tabellen [PRUEFUNGSTYP] und [RELEVANZ] stammende Fremdschlüssel ermöglichen eine weitere Qualifikation der Tupel, die im Hinblick auf die Berücksichtigung bzw. Positionierung der Ergebnisse einzelner Prüfungen – beispielsweise im Prüfbericht (3.7.3.4) sowie im Hinblick auf künftige mögliche Erweiterungen des Datenmodells von Bedeutung ist.



**Abbildung 32: ER-Modell Prüfung & Prüfung suspekter Werte**

Prüfungen, deren Gegenstand die im Datenmodell abgebildeten Informationen sind, können sich auf 3 Sachverhalte erstrecken:

- Existenz suspekter Elemente
- obligate Koppelung von Elementen
- gegenseitiger Ausschluss von Elementen.

### 3.6.4.1. Existenz suspekter Elemente

Als „suspekte Elemente“ werden Angaben definiert, die prinzipiell zwar zulässig sind, da sie – wie beispielsweise die Ausprägung „Infarkt(e): unbekannt“ (3.1.11.) - Bestandteil der BQS-Spezifikation des Moduls Herzchirurgie sind, tatsächlich in der Praxis jedoch höchst selten vorkommen, weil sich - um beim Beispiel zu bleiben - in fast allen Fällen spätestens in der Retrospektive eine konkrete Aussage für das genannte Feld ermitteln lässt.

Einträge der Tabelle [PRUEFUNG\_ELEMENT] (Abbildung 23) spezifizieren Elemente, deren Existenz innerhalb der Daten eines Falles verdächtig im Sinne von „höchstwahrscheinlich falsch“ ist. Die beispielhaft genannte Aussage „Infarkte: unbekannt“ kann detektiert werden, indem durch Einträge in den Tabellen [ELEMENT] und [ELEMENT\_WERT\_WERT] (3.6.3., Abbildung 31) ein Bezug zu dieser speziellen, in der

Tabelle [WERT\_WERT] (3.3., Abbildung 7) hinterlegten Ausprägung des codeunabhängigen Wertes „Infarkt(e)“ (=Datensatz der Tabelle [WERT]) hergestellt wird.

Die Datenbank-Abfragen, mit deren Hilfe suspekt Elemente innerhalb der Daten eines Falles oder eines Eingriffs entdeckt werden können, bilden die Basis der Sichten [FALL\_PRUEFUNG\_ELEMENT] bzw. [FALL\_OP\_PRUEFUNG\_ELEMENT]; die zugehörigen Quelltexte finden sich im Anhang 5 (A5.3., A5.4.).

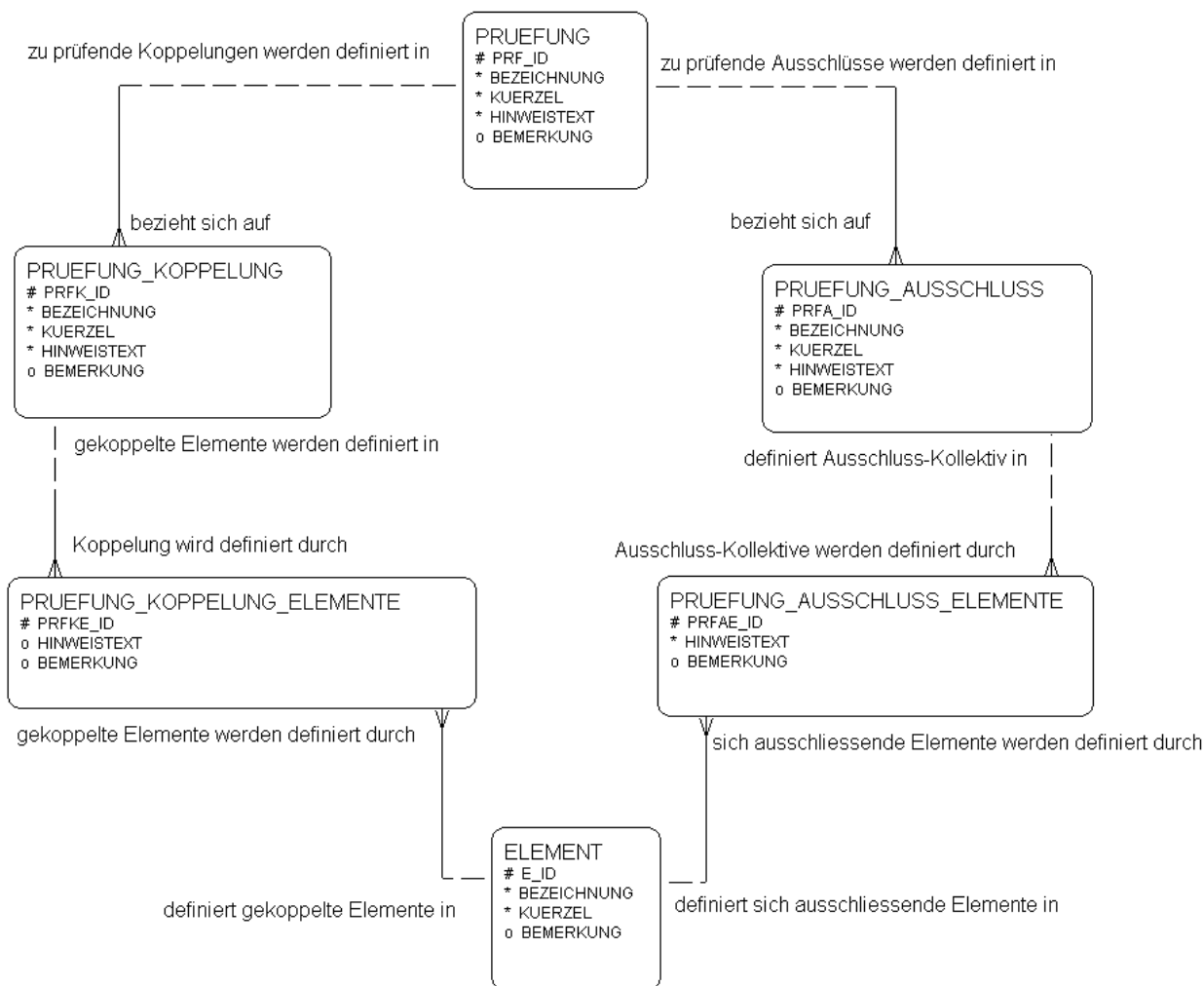
#### 3.6.4.2. Obligate Koppelung von Elementen

Regeln dieser Kategorie erstrecken sich auf üblicherweise gemeinsam existierende Angaben; typische Beispiele hierfür sind Kombinationen aus therapeutischen Maßnahmen und deren Indikation (OPS-Schlüssel für die Transfusion von Blutprodukten und ICD-Schlüssel für Erkrankungen wie „akute Blutungsanämie“) oder gleichsinnige Angaben aus den beiden Bereichen „Abrechnung“ und „Qualitätssicherung“ (die vorgenannten Transfusions-bezogenen OPS-Schlüssel im Verein mit der Erfassung ggf. übertragener Erythrozytenkonzentrate im Feld [89], 3.1.89.).

Solche Koppelungen können für die gezielte Suche nach Elementen genutzt werden, deren definierter Gegenpart fehlt, um Hinweise auf widersprüchliche bzw. fehlende Daten zu liefern.

Die hier beschriebenen Regeln werden in der Tabelle [PRUEFUNG\_KOPPELUNG] hinterlegt, während „ihre“ Elemente, zwischen denen eine inhaltliche Verbindung der oben genannten Art besteht, durch Einträge der Tabelle [PRUEFUNG\_KOPPELUNG\_ELEMENTE] repräsentiert werden (Abbildung 33).

Die im Anhang 5 zu findenden Quelltexte der Fall- bzw. OP-bezogenen Sichten [FALL\_PRUEFUNG\_KOPPELUNG] und [FALL\_OP\_PRUEFUNG\_KOPPELUNG] (A5.5. bzw. A5.6.) beschreiben die Ausgestaltung entsprechender Abfragen.



**Abbildung 33: ER-Modell Prüfung obligate Koppelung & gegenseitiger Ausschluss**

### 3.6.4.3. Gegenseitiger Ausschluss von Elementen

Die „Koppelung“ genannten Zusammenhänge, die Gegenstand des vorhergehenden Abschnitts (3.6.4.3.) sind, lassen sich prinzipiell auch in entgegengesetzter Weise formulieren.

So steht – um beim dortigen Beispiel zu bleiben – eine über die Angabe eines OPS-Schlüssels dokumentierte Transfusion von Erythrozytenkonzentraten im Konflikt mit einer Angabe des Wertes „0“ im Feld „Erythrozytenkonzentrat(e)“ des Moduls HCH.

Für Regeln, die sich auf den gegenseitigen Ausschluss von Elementen beziehen, ist die Tabelle [PRUEFUNG\_AUSSCHLUSS] bestimmt, während die Informationen zu sich einander widersprechenden Angaben in der Tabelle [PRUEFUNG\_AUSSCHLUSS\_ELEMENTE] hinterlegt werden (Abbildung 33).

Die Abfragen, die die Basis der Sichten [FALL\_PRUEFUNG\_AUSSCHLUSS] (fallbezogen; Anhang 5, A5.7.) und [FALL\_OP\_PRUEFUNG\_AUSSCHLUSS] (eingriffsbezogen; Anhang 5, A5.8.) darstellen, liefern – gestützt auf den Nachweis der gleichzeitigen Existenz eigentlich nicht miteinander vereinbarer Elemente - Hinweise auf potentielle Mängel innerhalb der Dokumentation.

## **3.7. „Modell EJK“: Anwendung des Datenmodells**

### **3.7.1. Ausgangssituation**

Im „Herzzentrum Duisburg“ als Betriebsteil der „Evangelischen und Johanniter Klinikum Niederrhein gGmbH“ (EJK) und damit auch in der „Klinik für Thorax- und Kardiovaskularchirurgie“ wird das Krankenhausinformationssystem „ORBIS“ des Herstellers „Agfa Healthcare“ eingesetzt.

Neben der für die Abrechnung relevanten, für alle Fachabteilungen vorgesehenen Erfassung von ICD- und OPS-Schlüsseln werden die im Verlauf einer herzchirurgischen Behandlung anfallenden Daten (hierzu zählen insbesondere die für das Modul HCH erforderlichen Angaben!) in spezifischen Formularen dokumentiert (Abbildung 4).

Diese aufgeteilte Datenerhebung versetzt die verschiedenen Teams (ärztlicher Dienst auf den Normal- und Intensivstation{en}, Operateure, Mitarbeiter der Echokardiographie-Abteilung, Kardiotechniker) in die Lage, ihre jeweiligen, gebietsbezogenen Angaben in kooperativer Weise zu erfassen, ohne dabei auf die Nutzung bereits vorhandener Informationen verzichten zu müssen.

Konsequenz daraus ist, dass die für einen kompletten Datensatz des Moduls HCH erforderlichen Daten in verschiedenen Tabellen innerhalb der Datenbasis des „ORBIS“-Systems vorgehalten werden.

Wichtig ist ferner der Umstand, dass der Hersteller des KIS keine Schnittstelle für das für die QS-Daten bestimmte Modul namens „QSOM“ zur Verfügung stellt, mit deren Hilfe eine Übertragung auf direktem Wege hätte realisiert werden können. Inwiefern dieser Punkt von Bedeutung ist, wird in den folgenden Abschnitten erläutert.

### **3.7.2. Ziele**

Mit Hilfe von Software-Komponenten, die sich auf das Datenmodell stützten, sollten folgende Ziele erreicht werden:

- Vermeidung redundanter Erfassung von Daten und Reduktion des damit einhergehenden Aufwandes: wenn sich beispielsweise aus den erfassten ICD-Schlüsseln eindeutig ableiten lässt, dass eine zerebrovaskuläre Erkrankung des ZNS vorlag, macht es keinen Sinn, diese Angabe ein weiteres Mal separat zu erfassen.
- Verbesserung der Güte der erfassten Daten: gegenseitige Prüfung der Angaben aus den Bereichen „DRG-Abrechnung“ und „Qualitätssicherung“, um potentielle Fehler (in beiden Kompartimenten!) identifizieren und korrigieren zu können.
- Behandlungsbegleitende, kooperative Erhebung der Daten: jeder Mitarbeiter erfasst diejenigen Angaben, die in direkter Verbindung mit seiner Tätigkeit stehen; die geringere Latenz zwischen „Entstehung“ und Dokumentation von Informationen soll zu einer größeren Validität der Daten führen, zumal ein Großteil der Angaben sich nur schwierig mit größerem zeitlichen Abstand retrospektiv gewinnen lässt.
- Vereinigung der aufgeteilten Informationen und Erzeugung des Datensatzes für das Modul HCH durch automatisierte Übertragung aller relevanten Angaben in das dafür vorgesehene Modul des KIS (im Fall von ORBIS: „QSOM“).



### 3.7.3. Lösung

#### 3.7.3.1. Akquisition der Daten aus dem KIS

Das Datenmodell und die Routinen, die seinen Nutzen für die Anwender erschließen, wurden für einen im Dokumentationsprozess unmittelbar der DRG-relevanten Erfassung bzw. Komplettierung der ICD- und OPS-Codes folgenden Einsatz entworfen.

Zu diesem Zeitpunkt wird über eine Menüoption innerhalb von „ORBIS“ die im Rahmen dieser Arbeit entwickelte JAVA-Applikation aufgerufen, wobei der aktuell durch den Mitarbeiter bearbeitete Fall durch die Übergabe von Aufruf-Parametern spezifiziert wird.

Das Programm führt daraufhin verschiedene – der einfacheren Wartbarkeit halber ihrerseits in Tabellen hinterlegte – SQL-Anweisungen aus, mit deren Hilfe aus dem Bestand des KIS Stamm- und OP-Daten sowie ICD- bzw. OPS-Schlüssel des Falles ebenso wie die auf verschiedene Spezial-Formulare (reduzierte QS-Formulare, HLM-Protokoll) verteilten, für das Modul HCH relevanten Angaben gelesen werden (Abbildung 34).

Die Akquisition der Daten sowie deren weitere Verarbeitung wird durch die Java-Applikation in Form eines für den Anwender nachvollziehbaren Protokolls visualisiert (Abbildung 35).

*Hinweis: alle SQL-Operationen, die als Bestandteil des Projekts entwickelte Java-Programme oder -Bibliotheken ausführen, machen Gebrauch von der in die Java-Laufzeitumgebung bzw. in die Programmiersprache integrierten JDBC-Funktionalität [30]. Diese Softwaretechnologie ermöglicht es, das Datenmodell auf allen freien oder kommerziell erhältlichen RDBMS einzusetzen, für die entsprechende Treiber zur Verfügung stehen und die den verwendeten SQL-Standard (SQL92) unterstützen. Gleiches gilt für die an die Datenbasis des KIS gerichteten Abfragen. Ein Großteil der JDBC-basierten Routinen der hier vorgestellten Java-Applikation sind in der Klasse AgentDatabase implementiert, deren via JavaDoc [31] beschriebenes API sich in Anhang 7 findet.*

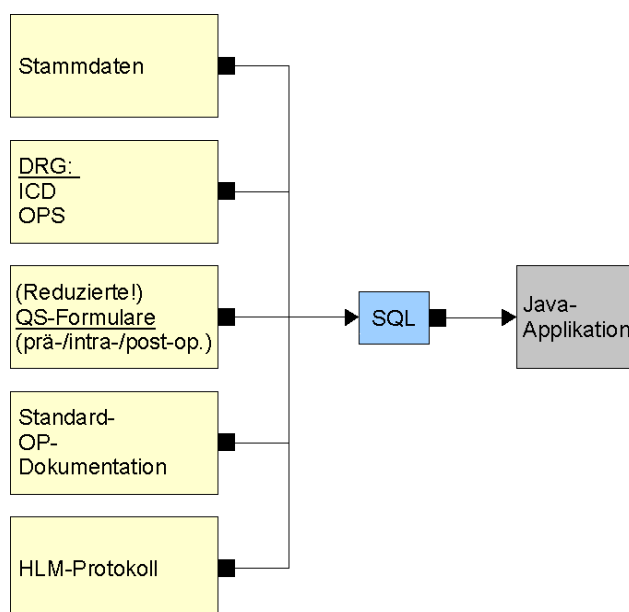


Abbildung 34: Akquisition der Daten aus dem KIS

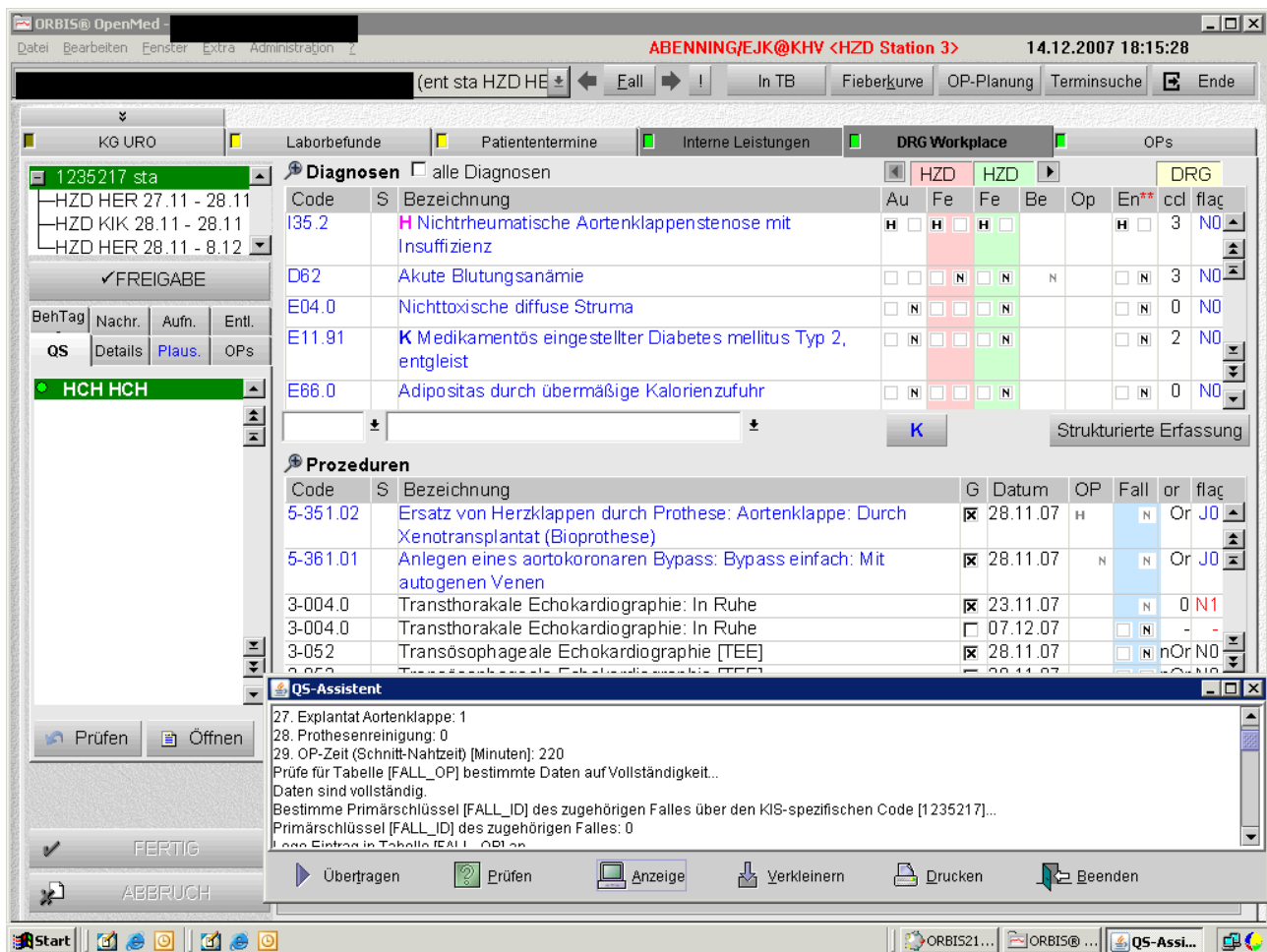


Abbildung 35: Darstellung des Protokolls der Daten-Akquisition und -Verarbeitung

### 3.7.3.2. Übertragung der Rohdaten in die Strukturen des Datenmodells

Die direkt übernommenen ICD- und OPS-Codes sowie die codeunabhängigen Daten des spezifizierten Falles werden in die dafür vorgesehenen Datenstrukturen des Modells (3.4.) übertragen (Abbildung 36).

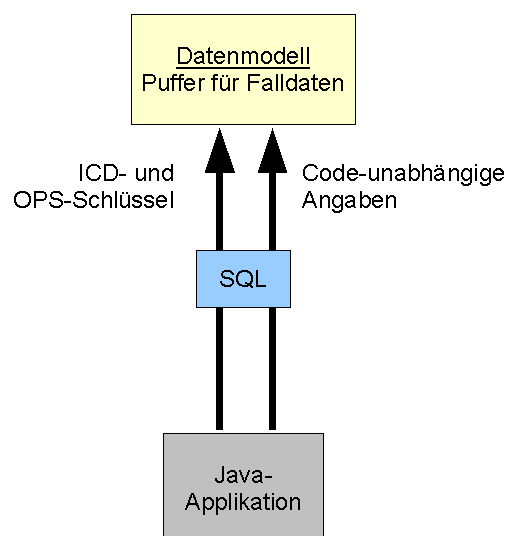
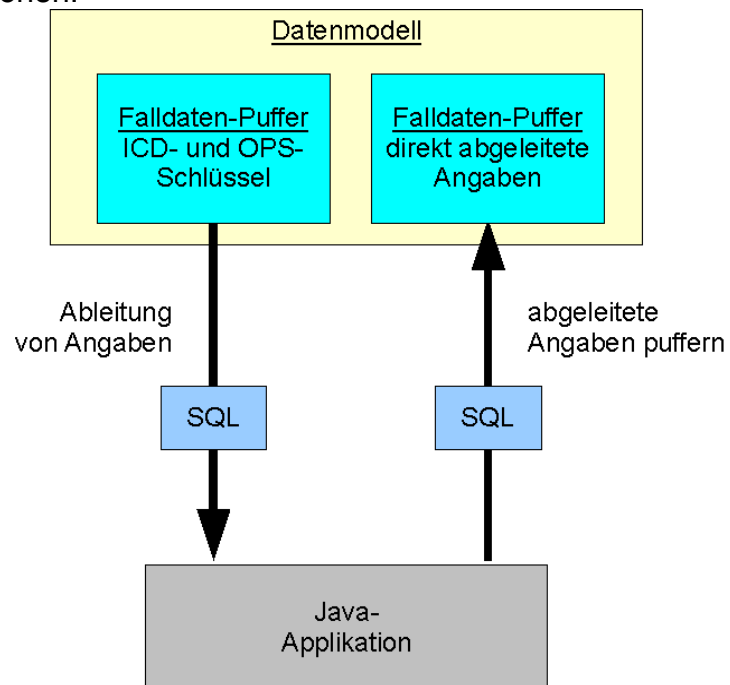


Abbildung 36: Übertragung der Rohdaten in die Strukturen des Datenmodells

### 3.7.3.3. Ableitung von Angaben aus den ICD- und OPS-Codes

Die Java-Applikation führt anschließend die auf den in Kapitel 3.5. beschriebenen Methoden fußenden Abfragen aus und speichert die abgeleiteten Angaben in den als Puffer dienenden Datenstrukturen (Abbildung 37). Dieses Procedere dient – wie bereits in Kapitel 3.6.1 erwähnt - der Steigerung der Performanz bei wiederholten Zugriffen auf dieselben Informationen.



**Abbildung 37: Ableitung von Angaben und deren anschließende Pufferung**

Beide Kategorien von Daten – die direkt aus dem KIS übernommenen wie auch die als Ergebnis der Ableitung gewonnenen Feld-Inhalte – werden durch das Programm dargestellt, wodurch sich dem Anwender die Möglichkeit bietet, die Angaben zu kontrollieren (Abbildung 38).

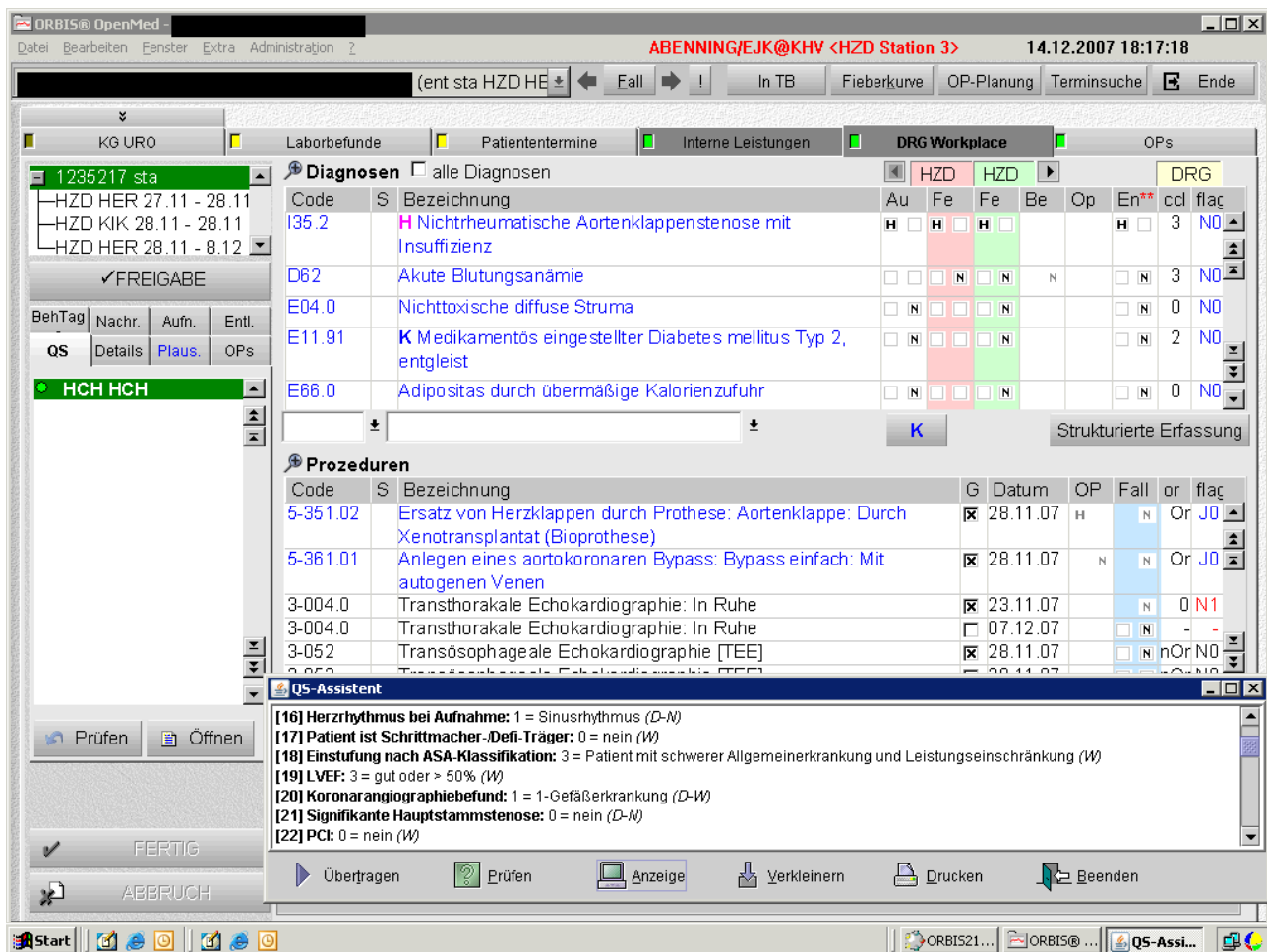


Abbildung 38: Darstellung der direkt übernommenen sowie der abgeleiteten Angaben

### 3.7.3.4. Erzeugung und Anzeige des Prüfberichts

Im nächsten Schritt werden die in Kapitel 3.6. beschriebenen Auswertungen auf Basis der gepufferten Falldaten durch das Java-Programm ausgeführt (Abbildung 39) und deren Ergebnisse textlich aufbereitet als sog. Prüfbericht dem Benutzer zur Verfügung gestellt.

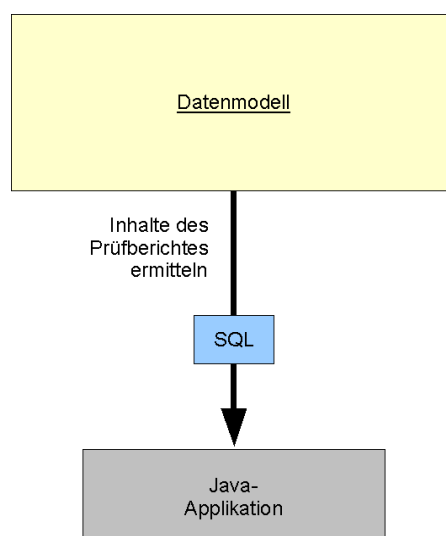


Abbildung 39: Inhalte des Prüfberichtes ermitteln

Der Bericht kann am Bildschirm dargestellt (Abbildung 40) oder in äquivalenter Form gedruckt werden und so als Grundlage für eine auf den wechselseitigen Vergleich gestützte Kontrolle der Dokumentation durch den bearbeitenden Mitarbeiter dienen.

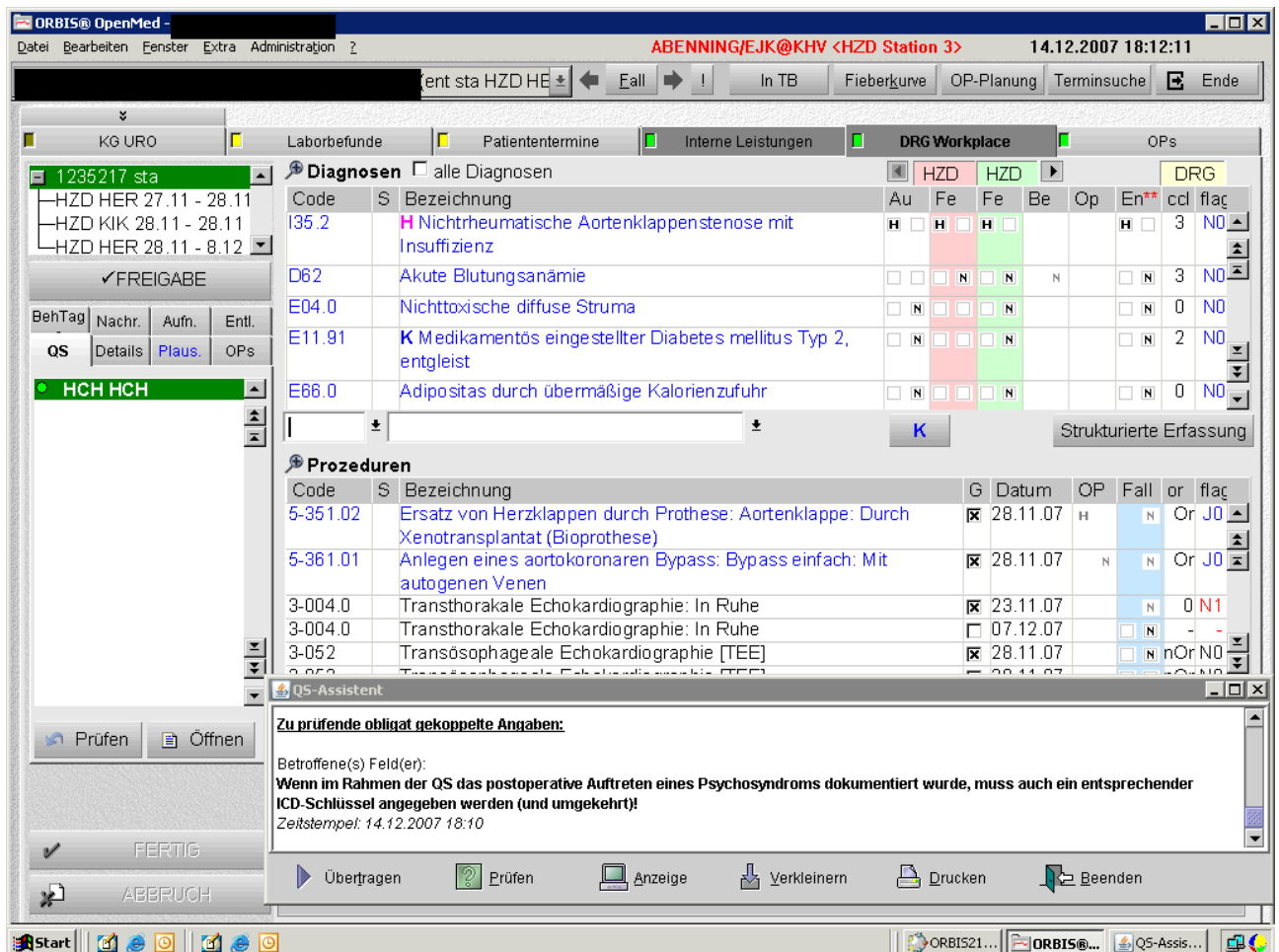


Abbildung 40: Darstellung des Prüfberichts; Druck-Option

### 3.7.3.5. Übertragung des kompletten HCH-Datensatzes ins KIS

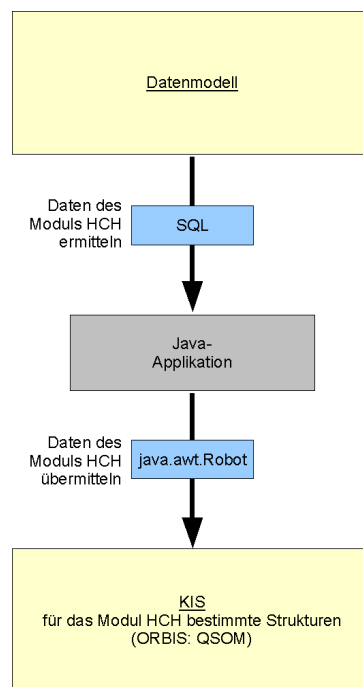
Wenn – wie im Fall des Produkts „ORBIS“ - das eingesetzte KIS keine Schnittstelle bietet, mit deren Hilfe die Inhalte des Moduls HCH von Fremdprogrammen wie der hier vorgestellten Java-Applikation direkt übertragen werden können, verbleibt alternativ die Möglichkeit, die Interaktionen eines menschlichen Anwenders zur Eingabe der Daten über die jeweils zur Verfügung stehende grafische Oberfläche (im Fall von „ORBIS“ ist dies das sog. „QSOM-Formular“ des betreffenden Falls) zu simulieren.

Dabei werden die zu übertragenden Informationen in eine Sequenz von Tastatur- und/oder Maus-Befehlen umgesetzt, die das jeweils für eine Einzel-Angabe bestimmte Oberflächen-Element aktiviert und den zugehörigen Wert „einträgt“. So können – Feld für Feld – alle Inhalte des Datensatzes Herzchirurgie transferiert werden (Abbildungen 41 und 42).

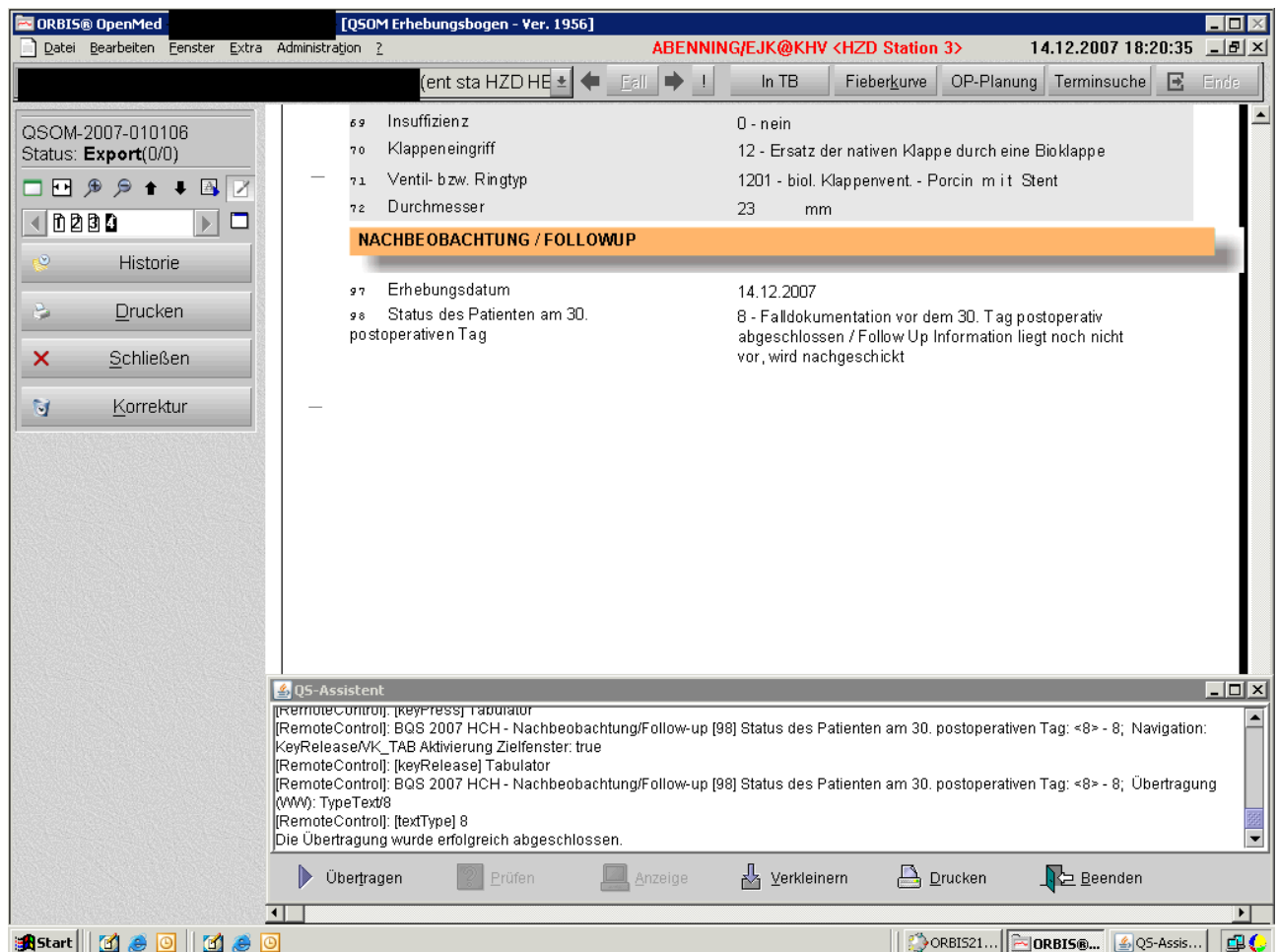
Diese Fremdsteuerung von Programmen durch andere (Java-)Applikationen über ihre grafische Benutzeroberfläche wird mit Hilfe der durch die Standardklasse `java.awt.Robot` zur Verfügung gestellten Funktionen realisiert [32].

Die Transformation der Feld-Inhalte des Datensatzes Herzchirurgie in Steuersequenzen

gelingt auf der Basis von in Tabellen hinterlegten Informationen. Die Determination des Umwandlungsprozesses durch Datenbank-basierte Parameter eröffnet die Möglichkeit, die Anwendung ohne Änderung der Java-Quelltexte an neue Versionen der eingesetzten KIS-Module oder sogar an die Produkte anderer Hersteller anzupassen.



**Abbildung 41: Übertragung der Daten als Sequenz von Tastatur-/Maus-Befehlen**



**Abbildung 42: Abschluss der Datenübertragung via java.awt.Robot**

### 3.7.3.6. Pflege der Hinterlegungen

Für jede Tabelle des Datenmodells existiert eine Java-Applikation, die eine komfortable Bearbeitung bzw. Pflege der dort hinterlegten Daten ermöglicht. Die grafischen Oberflächen dieser Programme sind ähnlich aufgebaut und umfassen folgende Elemente:

- Menü für den Aufruf von Programmfunktionen (wie bspw. „Änderungen speichern“ etc.)
- Werkzeugleiste für den raschen Zugriff auf Optionen des Menüs über Schaltflächen, deren Symbole einheitlich gestaltet sind und denjenigen im Bereich des Menüs entsprechen
- Bereich für die Definition von ggf. anzuwendenden Suchkriterien
- Arbeitsbereich für die Bearbeitung der Tabellen-Inhalte (Anlegen neuer, Änderung oder Entfernung bestehender Datensätze)
- Statuszeile zur Darstellung von Informationen zum augenblicklichen Zustand der Applikation.

Die standardisierte Anmutung der Benutzeroberflächen ist das Ergebnis eines Merkmals des Software-Designs: alle Module leiten sich jeweils von einer abstrakten Vorgängerklasse ab, von deren Methoden lediglich einige wenige angewendet werden müssen, um eine Anpassung an die Struktur der jeweiligen Tabelle bei sonst weitgehend identischem Funktionsumfang zu realisieren. Auf diese Weise können Modifikationen bzw. später ggf. als notwendig erachtete Erweiterungen auf einfache Weise umgesetzt werden.

Die für den Aufbau der Datenbank-Verbindung benötigten Angaben können den Einzelprogrammen als Aufruf-Parameter übergeben oder innerhalb der Anwendung in einem dafür vorgesehenen Dialog interaktiv spezifiziert werden. Auf diese Weise wird die als Entwicklungsziel definierte Unabhängigkeit von der verwendeten Datenbank - bzw. genauer: dem verwendeten RDBMS – erreicht (1.1.).

Auf den unten dargestellten Bildschirmfotos (Abbildungen 43 und 44) sind beispielhaft die Module zur Bearbeitung der Datensätze der Tabellen [DIAGNOSE\_WERT\_ICD] (3.5.2.) und [PRUEFUNG\_KOPPELUNG\_ELEMENTE] (3.6.4.2.) dargestellt.

In beiden Abbildungen ist ein weiteres, in zahlreichen weiteren Modulen eingesetztes Oberflächen-Element sichtbar: für die Angabe von Fremdschlüsseln wurden Auswahllisten in die Tabellen-Zeilen integriert, die statt der numerischen Werte die leichter interpretierbaren Inhalte der [BEZEICHNUNG] genannten Felder der referenzierten Tabellen enthalten. Diese Bezeichnungen werden ihrerseits durch während der Programm-Initialisierung ausgeführte SQL-Abfragen ermittelt – im Fall der Abbildung 43 aus der Tabelle [DIAGNOSE\_WERT] bzw. im Beispiel der Abbildung 44 aus der Tabelle [ELEMENT].

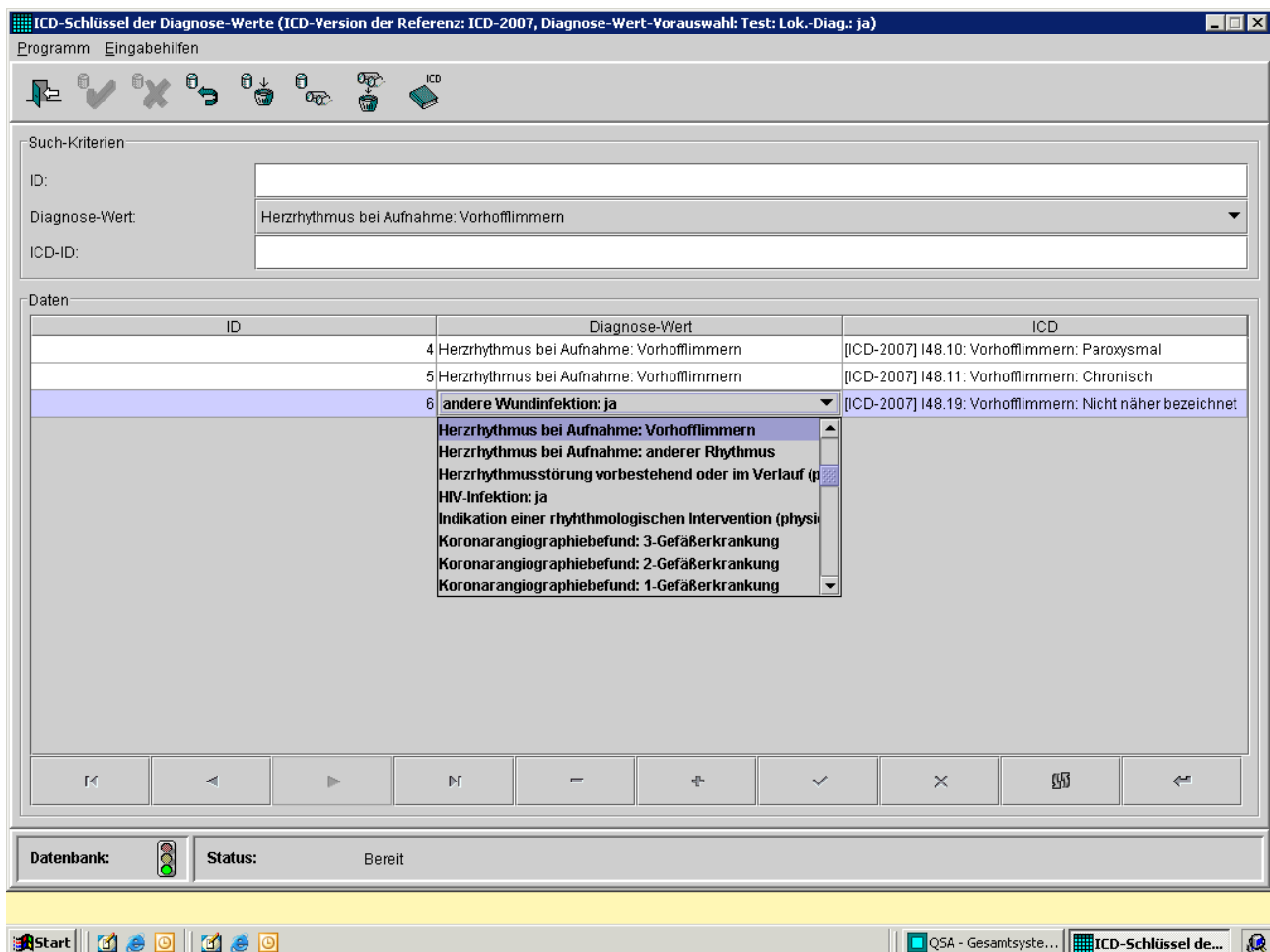
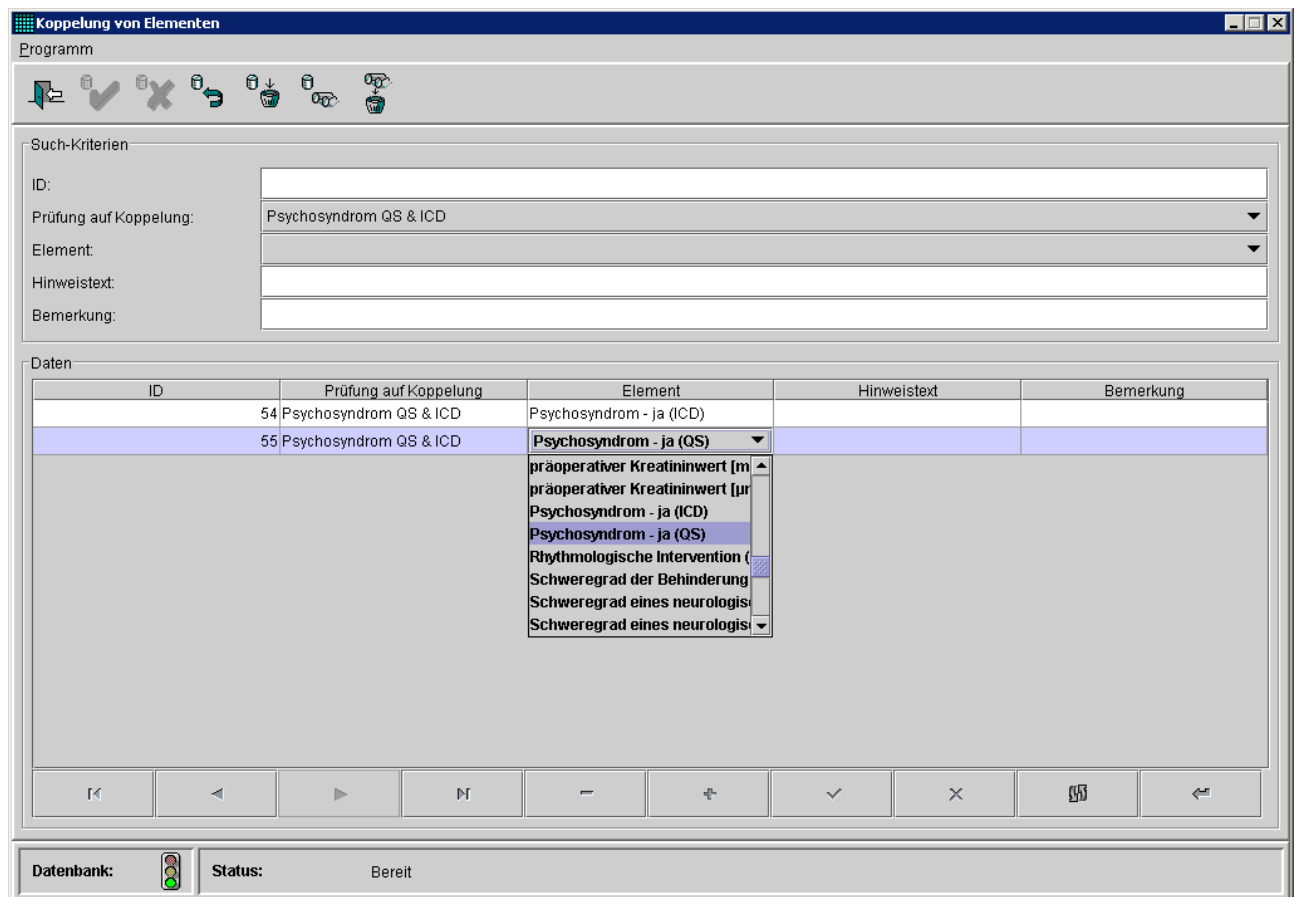


Abbildung 43: Modul zur Pflege der Tabelle [DIAGNOSE\_WERT\_ICD]





**Abbildung 44: Modul zur Pflege der Tabelle [PRUEFUNG\_KOPPELUNG\_ELEMENTE]**

### 3.7.3.7. Menüsystem und Pflege der System-Konfiguration

Alle für die Pflege von Tabellen des Datenmodells vorgesehenen Module (3.7.3.6.) sind in ein zentrales Menüsystem eingebunden (Abbildung 45). Eine Baumstruktur dient der thematischen Kategorisierung der Module, wobei sowohl die Kategorien als auch die dort verwendeten Bezeichnungen durch den Administrator des System frei wählbar sind.

Möglich wird das durch die Hinterlegung aller zur Konfiguration und Steuerung des Menüsystems verwendeten Daten in Tabellen, deren Einträge ihrerseits durch Module bearbeitet werden können, die denen im vorhergehenden Abschnitt (3.7.3.6.) vergleichbar sind.

So kann das Menüsystem mit geringem Aufwand um neue Module und damit neue Funktionen erweitert werden, ohne hierzu den Java-Quelltext modifizieren zu müssen.

In Abbildung 46 ist exemplarisch das Modul zur Pflege derjenigen Tabelle dargestellt, die die Informationen aller im Menüsystem verfügbaren Module beherbergt.

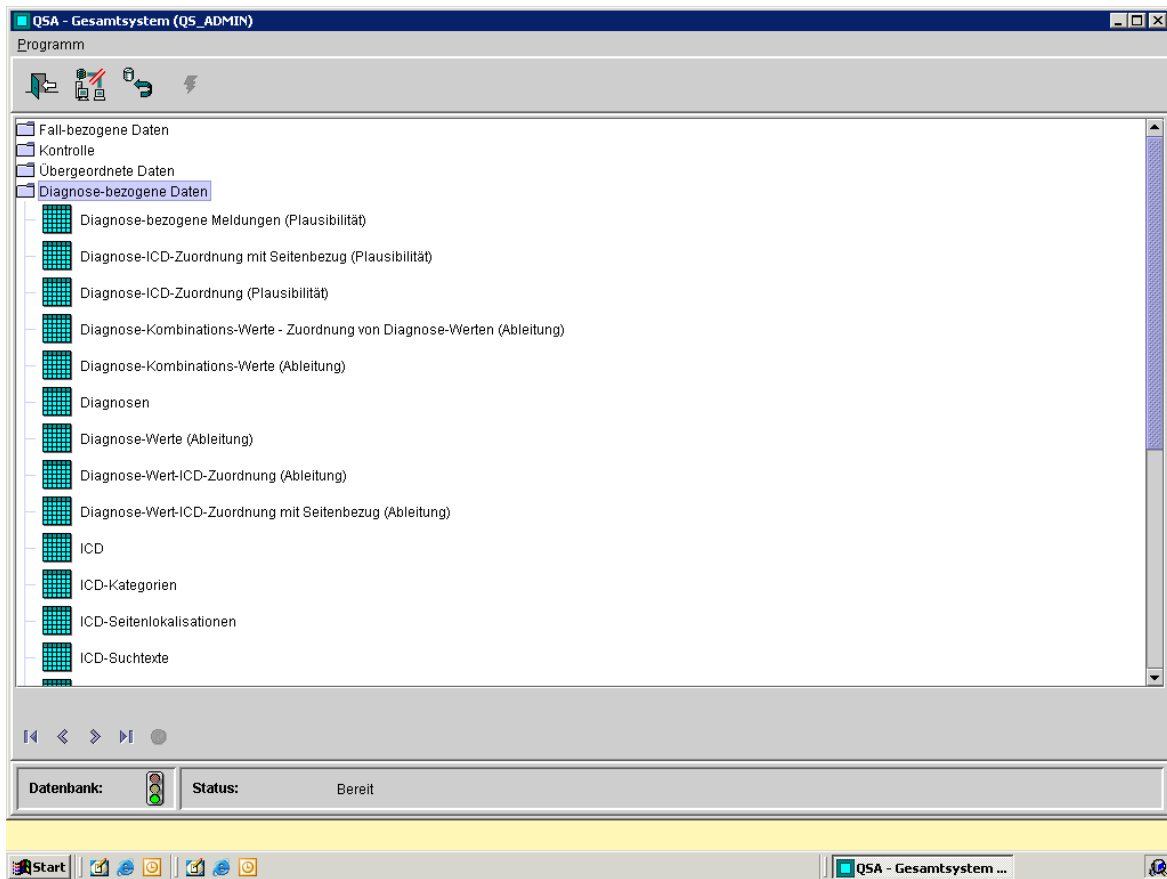


Abbildung 45: Baumstruktur des zentralen Menüsystems

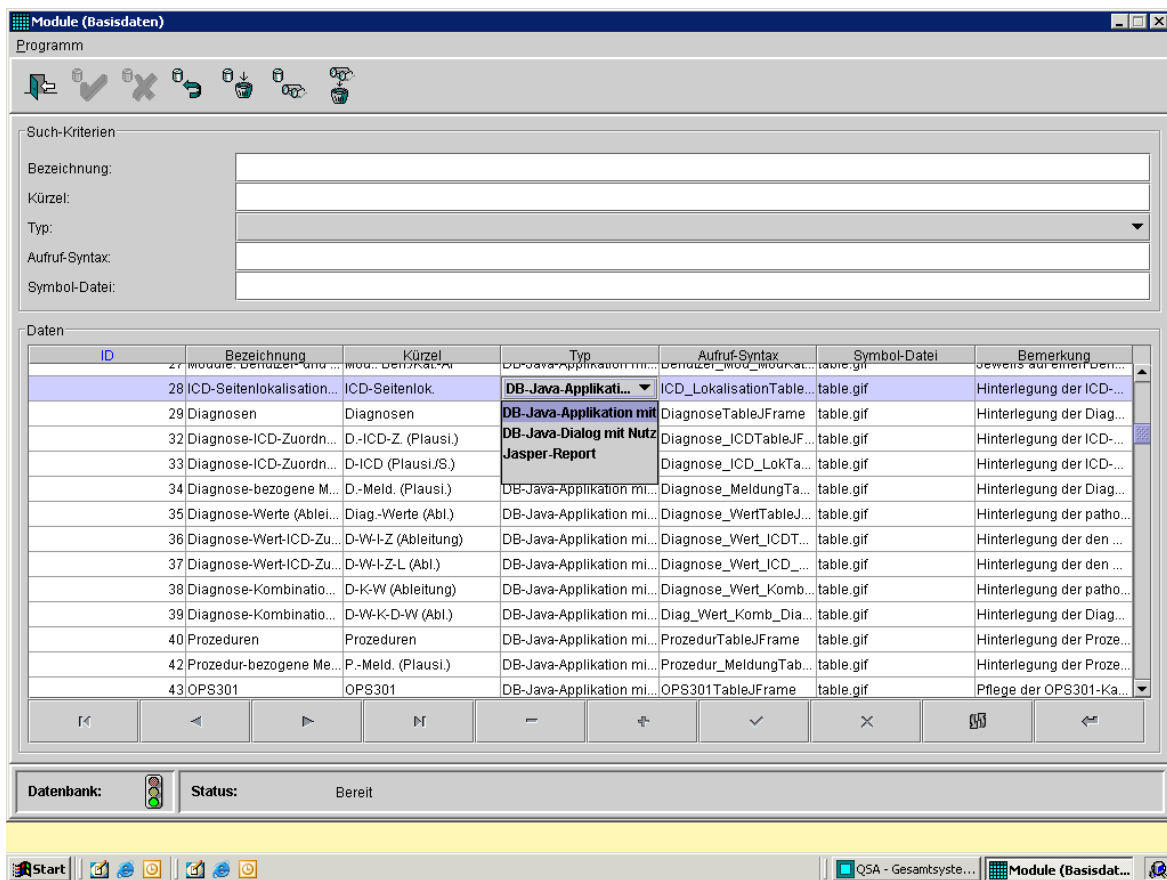


Abbildung 46: Modul-Verwaltung des Menüsystems

## 4. Diskussion und Ausblick

Orientiert an den in Abschnitt 1.1. formulierten Zielen und unter Berücksichtigung der ersten, allerdings noch als vorläufig zu bezeichnenden Erfahrungen mit der Anwendung des Datenmodells und den sich darauf stützenden Applikationen (3.7.) in der Praxis erfolgt nun eine Bewertung der erreichten Ergebnisse.

### 4.1. Das konkrete System und dessen praktischer Einsatz

Wie die in den Abschnitten 3.5. und 3.6. beschriebenen, auf die Strukturen des Datenmodells gestützten Abfrage-Strategien ebenso wie der mittlerweile routinemäßige Einsatz in der Klinik für Thorax- und Kardiovaskularchirurgie des EJK zeigen, gelingen sowohl die Ableitung als auch die wechselseitige Plausibilitätsprüfung der QS- und der DRG-relevanten Daten zuverlässig.

Dieses Ergebnis setzt allerdings – im Abschnitt 3.1. dokumentierte - Recherchen in den amtlichen Schlüssel-Katalogen (ICD, OPS) voraus, die insbesondere auch die vom DIMDI veröffentlichten synonymen Bezeichnungen berücksichtigen. Die differenzierte Betrachtung beider Datenmengen im Hinblick auf Möglichkeiten zur Kontrolle der Konsistenz von Angaben bildet die Grundlage eines nutzbringenden Prüfmechanismus, dessen Güte nicht zuletzt von der Präzision und Verständlichkeit der den Anwendern schließlich präsentierten Rückmeldungen abhängig ist.

In diesem Zusammenhang ist der Hinweis bedeutsam, dass ausnahmslos alle Definitionen für die Ableitung und Prüfung von Angaben in der in den Abschnitten 3.5. und 3.6. beschriebenen Weise in Form von Einträgen in den Tabellen des Datenmodells angelegt wurden. Notwendige Eingaben bzw. Änderungen wurden ausschließlich mit Hilfe der im Abschnitt 3.7.3.6. vorgestellten Applikationen zur Pflege der Hinterlegungen durchgeführt – in keinem Fall waren bzw. sind hierzu Anpassungen der Java-Quelltexte der Applikationen notwendig. Die Wartung und – in Grenzen – die Weiterentwicklung des Systems setzt also eben nicht die Expertise eines Softwareentwicklers mit Kenntnissen der Programmiersprache Java voraus.

Ausdrücklich als subjektiv zu etikettierenden Berichten der Anwender des in dieser Arbeit vorgestellten Systems zufolge führt dessen Einsatz zu einer spürbaren Entlastung vom früher durch die notwendige separate Kontrolle der Datenmengen verursachten Aufwand. Die direkte Erschließung der Inhalte von 32 Feldern des Moduls HCH (3.1.100.11.) sowie die gegenseitige Prüfung der Angaben zu 43 weiteren Feldern (3.1.100.2.) auf Basis der DRG-relevanten ICD- und OPS-Codes bzw. auch jeweils untereinander ist demnach mit einer geringeren Fehlerträchtigkeit des Dokumentationsprozesses verbunden, wobei die Reduktion der durch die früher notwendige, explizite Erfassung der erwähnten 32 ableitbaren Angaben bedingten Belastung ebenfalls willkommen ist.

Um diesen noch wenigstens teilweise spekulativen Eindrücken ein gesichertes Fundament zu verschaffen, drängt sich die Durchführung einer Untersuchung auf, die – unter Berücksichtigung des Aufwandes - die Güte der Ergebnisse herkömmlicher Dokumentation mit denjenigen vergleicht, die unter Inanspruchnahme der neu entwickelten Lösung erzielbar sind.

Die mangels einer durch den Hersteller Agfa zur Verfügung gestellten Schnittstelle zur Übertragung der Daten des Moduls HCH in die Datenbasis des KIS „ORBIS“ realisierte

Lösung, die die Bedienung der Oberfläche der KIS-Teil-Applikation namens „QSOM“ (3.7.3.5.) durch einen „menschlichen Benutzer“ simuliert, erlaubt zwar eine Anpassung an jede vorstellbare, HCH-Daten-empfangende Software – insbesondere an die Produkte anderer Anbieter von Krankenhaus-Informationssystemen; jedoch ist dieser innerhalb der Arbeitssitzungen der System-Benutzer ablaufende Transfer prinzipiell durch Seiteneffekte anderer, gleichzeitig ausgeführter Applikationen störfähig. Bisher gab es allerdings keinen Fall einer solchen, damit also eher theoretisch relevanten Interferenz.

Ableitung und Prüfung der Daten beanspruchen mehrere (etwa 2-3) Minuten je Fall; damit ist genügend Potential für eine Optimierung der in den Abschnitten 3.5.7. und 3.6. vorgestellten Auswertungen vorhanden.

## **4.2. Flexibilität und mögliche Erweiterungen**

Die Identifikation und Berücksichtigung genereller Prinzipien waren von Anfang an bedeutsame Ziele während der Entwicklung des Datenmodells und der darauf fußenden Applikationen.

Ihren Niederschlag fanden sie nicht nur im bereits (4.1.) erwähnten Merkmal einer weitgehend auf Tabellen-Einträgen gestützten, demnach nicht von der Verfügbarkeit programmiersprachlicher Expertise abhängigen Konfigurierbarkeit des resultierenden Gesamtsystems, sondern ebenso in der problemlos möglichen Anwendbarkeit auf die Erfordernisse weiterer Qualitätssicherungsmaßnahmen.

Exemplarisch sei das BQS-Modul „Koronarangiographie und Perkutane Koronarintervention (PCI)“ [33] genannt: schon ein orientierender erster Blick offenbart, dass hier ebenfalls ein bedeutender Anteil der Feld-Inhalte entweder aus den DRG-relevanten Schlüsseln abgeleitet werden kann oder alternativ für ein zweiseitiges Prüfverfahren geeignet ist.

Potentielle künftige Einsatzfelder sind jedoch nicht auf das „QS-DRG-Duo“ beschränkt: mit vergleichbar geringem Aufwand kann die gegenseitige Plausibilität von mit den Mitteln des ICD- bzw. OPS-Katalogs codierten Angaben geprüft werden, wobei die Kombination aus Datenmodell und zugehörigen Programmen als eine Art „Intra-DRG-Prüfmodul“ fungieren würde.

Auch eine Ausweitung der für die Herleitung oder Kontrolle in Betracht kommenden Datenbasis auf weitere im klinischen Bereich eingesetzte Produkte wie beispielsweise Labor-Informationssysteme ließe sich mit geringem Aufwand realisieren: sobald die Akquisition der Daten – zum Beispiel über SQL oder Dateioperationen – gelingt, kann der Nutzen von Angaben nahezu jeglicher Art in Form von „Elementen“ erschlossen werden (3.6.3.). Eine gewisse Kooperation der jeweiligen Hersteller ist allerdings Voraussetzung für jeden der beiden letztgenannten Punkte.

## **4.3. Unabhängigkeit von spezifischen Datenbank-Systemen**

Das Ziel der „Datenbank-Unabhängigkeit“ von Datenmodell und zugehörigen Applikationen wurde erreicht: es ist unerheblich, welches RDBMS zur Implementierung des Datenmodell verwendet wird, solange das Produkt folgende 2 Bedingungen erfüllt

- es muss ein JDBC-Treiber verfügbar sein (derzeit existiert kein frei oder kommerziell erhältliches RDBMS, das dieses Kriterium nicht erfüllt)

- es muss in der Lage sein, den Sprachstandard SQL-92 korrekt zu interpretieren, der die formale Grundlage für das Skript im Anhang 3 bildet.

Im Rahmen der in dieser Arbeit vorgestellten Entwicklung wurde das Datenmodell auf einem – kommerziell erhältlichen - RDBMS des Herstellers Oracle (Version 9i) implementiert; tatsächlich befinden sich alle seine Strukturen in einem separaten Schema des Servers, der auch die Datenbasis des KIS „ORBIS“ beherbergt.

Probatorisch zeigte sich, dass eine Instanz des freien Systems „MaxDB“ für diesen Zweck ebenso gut geeignet ist [34].

#### **4.4. Unabhängigkeit von spezifischen Krankenhaus-Informationssystemen**

Dieses Kriterium ist im Hinblick auf die Interaktion zwischen dem Datenmodell bzw. den zugehörigen Programmen einerseits sowie dem KIS andererseits relevant. Entscheidend für die Betrachtung ist der Prozess der Akquisition von KIS-Daten und – so weit diese Fähigkeit eingesetzt wird – die Rückübertragung von abgeleiteten Daten und ggf. Prüfberichten an das KIS.

##### **4.4.1 Akquisition von Daten aus dem KIS**

Die Gewinnung von Daten aus dem KIS kann auf verschiedenen Wegen erfolgen: für diese Arbeit wurden beispielsweise via JDBC übermittelte SQL-Abfragen an das Oracle-RDBMS von ORBIS eingesetzt. Da der Großteil der am Markt erhältlichen KIS auf relationalen Datenbanken basiert, dürfte dieses Verfahren sich prinzipiell auch für die Produkte anderer Anbieter eignen; allerdings ist in diesem Fall der Willen zur Kooperation auf Seiten des Herstellers unverzichtbare Voraussetzung für eine erfolgreiche Realisierung - ist es doch nahezu unmöglich, ohne wenigstens basale Kenntnisse der jeweiligen Datenstrukturen mit vertretbarem Aufwand geeignete Datenbank-Auswertungen zu formulieren.

Existiert keine Möglichkeit, einen direkten SQL-Zugriff auf die notwendigen Daten zu erhalten, können auch Nachrichten des KIS im HL7-Format [35] verarbeitet und ausgewertet werden. Dieses Verfahren erfordert jedoch – ebenso wie der Rückgriff auf Export-Dateien, die auf alternativen Wegen (zum Beispiel mit speziellen Werkzeugen des Herstellers) erzeugt wurden – die Entwicklung zusätzlicher Programm-Komponenten bzw. Software-Routinen.

##### **4.4.2. Rückübertragung von Daten an das KIS**

Die Nutzung der durch die Java-Klasse `java.awt.Robot` zur Verfügung gestellten Funktionen ermöglicht die automatisierte Bedienung des für die Entgegennahme der Qualitätssicherungsdaten bestimmten Moduls des KIS durch das unter Punkt 3.7.3.5. beschriebene Programm. Dieses Übertragungsverfahren lässt sich an jede beliebige Benutzeroberfläche und damit an jedes am Markt erhältliche Produkt (KIS) anpassen. Da die Erzeugung der für die „Fernsteuerung“ notwendigen Sequenzen aus Tastatur- und/oder Maus-Ereignissen ihrerseits durch die Inhalte spezieller Tabellen determiniert wird, ist eine Modifikation mit geringem Aufwand verbunden und setzt zudem keine programmiersprachlichen Kenntnisse – etwa der Sprache Java – voraus. Als besonders attraktiv mag zudem die Unabhängigkeit von der Kooperationsbereitschaft des KIS-Anbieters erscheinen.

Trotz der genannten Vorteile ist der Schaffung einer Schnittstelle zur strukturierten und kontrollierten Übergabe der Daten auf Seiten des KIS der Vorzug zu geben. Abgesehen davon, dass sich auf diese Weise der Grad der Automatisierung erhöhen bzw. das Ausmaß der notwendigen „menschlichen“ Interaktion reduzieren ließe, ist eine Interferenz durch andere Programme innerhalb derselben Betriebssystem-Sitzung dann nicht zu fürchten. Langfristig dürfte auch die Pflege einer sich auf eine Schnittstelle stützenden Lösung einfacher sein, weil – sofern die Erfahrungen während der Entwicklungsphase repräsentativ sind - Zusammensetzung und Inhalte der QS-Datensätze seltener Änderungen unterworfen sind als die Gestaltung und Benutzerführung von grafischen Oberflächen.

Formale Grundlage einer noch hypothetischen Schnittstelle könnte ein Standard wie HL7 (4.4.1.) oder ein anderes Format bilden, während der Austausch von Dateien über durch beide Systeme genutzte Datenträger beziehungsweise Verfahren zur direkten Kommunikation als Medien für den Transport der Daten dienen.

#### **4.5. Verwendung freier Werkzeuge und Komponenten**

Es gelang, den Großteil der Entwicklung mit Hilfe freier Werkzeuge zu realisieren – im einzelnen seien hier noch einmal die plattformunabhängige Laufzeitumgebung „Java“, die Entwicklungsumgebung „Netbeans“ sowie die beiden datenbankunabhängigen SQL-Clients „Execute Query“ [36] und „Squirrel SQL“ [37] genannt.

Ausnahmen stellen das für den Entwurf des Datenmodells verwendete Programm „Oracle Designer“ sowie das zu seiner Implementierung genutzte, gleichzeitig die Basis des KIS bildende Oracle-RDBMS dar.

Als Datenbank eignen sich jedoch prinzipiell auch andere SQL92-fähige, freie Systeme wie beispielsweise „MaxDB“ (4.3.), während die künftige Pflege und Erweiterung des Modells – gestützt auf die Bestandteil des Anhangs dieser Arbeit bildenden SQL-DDL-Skripte - alternativ durch Zuhilfenahme einer open-source-Applikation wie „DB Designer Fork“ [38] erfolgen kann. Eine – bei ausschließlicher Betrachtung der Kosten für die Software-Werkzeuge – kostengünstige Fortentwicklung des Systems durch Interessenten sollte daher möglich sein.

#### **4.6. Ausblick auf die zukünftige Entwicklung**

Obwohl das Ziel dieser Arbeit primär die Schaffung einer auf das herzchirurgische BQS-Modul fokussierten Lösung darstellte, sind die hier vorgestellten Strukturen und Algorithmen – beispielsweise Abfragen und Java-Komponenten – genereller Natur. Die Anpassung an andere Erhebungen ist – wie unter 4.1. und 4.2. bereits erwähnt – ohne Änderung des Datenmodells und mit allenfalls geringfügigen Änderungen an den Quelltexten der Java-Programme möglich; entsprechende Entwicklungsaktivitäten sind also folgerichtig die nächsten Schritte in der Evolution des Systems.

Nach Möglichkeit soll auch auf den Hersteller von „ORBIS“ eingewirkt werden, um eine weitergehende Integration der Bestandteile der Applikation in das KIS zu erreichen. Erstrebenswert wäre unter anderem die Schaffung einer Schnittstelle für die letztlich zu übertragenden Elemente des Datensatzes HCH (3.7.3.5.) sowie einer Möglichkeit zur Rückübertragung von Berichten als Ergebnis durchgeführter Prüfungen (3.6. und 3.7.3.4.). Gerade der letztgenannte Punkt ist von Bedeutung, wenn das System lediglich als

Prüfinstanz Verwendung finden soll.

Einige Passagen dieser Arbeit thematisieren – wenn auch nur marginal - die Vorzüge freier Software [39]. Dieses Modell der Software-Entwicklung stellt auch für die im Rahmen dieser Arbeit entwickelte Lösung eine attraktive Option dar, die sorgfältiger Abwägung bedarf. Die Perspektive, Lasten der künftigen Entwicklung auf die Mitglieder einer noch zu bildenden „Community“ zu verteilen, von deren Ergebnissen im Gegenzug alle Beteiligten profitieren, ist äußerst verführerisch.

Bei allem Optimismus darf nicht unerwähnt bleiben, dass der vorgestellte Ansatz geeignet ist, seine Anwender von stets wiederkehrenden, gleichförmigen Kontrollpflichten zu entlasten und den mit der Erhebung von Daten verbundenen Aufwand zu reduzieren. Gleichwohl wird er die Arbeit seiner Benutzer auch in Zukunft keineswegs vollständig überflüssig machen, da kein noch so ausgereiftes Modell mit maximal differenzierten Regeln alle Konstellationen berücksichtigen (im Sinne von „vorwegnehmen“) kann, die für den Geist eines (geschulten) Menschen allenfalls unbedeutende Herausforderungen bilden.

## Abkürzungsverzeichnis

API	<b>A</b> pplication <b>p</b> rogramming <b>i</b> nterface (Beschreibung der Schnittstelle einer Programm-Bibliothek; Zielgruppe: Software-Entwickler)
BQS	<b>B</b> undesgeschäftsstelle <b>Q</b> ualitätssicherung gGmbH
CCS	<b>C</b> anadian <b>C</b> ardiovascular <b>S</b> ociety
DIMDI	<b>D</b> eutsche Institut für <b>M</b> edizinische <b>D</b> okumentation und <b>I</b> nformation
DRG	<b>D</b> iagnosis <b>R</b> elated <b>G</b> roups
EJK	<b>E</b> vangelisches und <b>J</b> ohanniter <b>K</b> linikum Niederrhein gGmbH
ER(-Modell)	<b>E</b> ntity- <b>r</b> elationship-model = Entitäts-Beziehungs(-Modell)
G-DRG	<b>G</b> erman Refined - <b>D</b> iagnosis <b>R</b> elated <b>G</b> roups
HCH	Modul „ <b>HCH</b> “ für den BQS-Leistungsbereich „Herzchirurgie“
ICD	<b>I</b> nternational <b>c</b> lassification of <b>d</b> iseases = internationale Klassifikation der Krankheiten
IDE	<b>I</b> ntegrated <b>d</b> evelopment <b>e</b> nvironment = integrierte Entwicklungsumgebung
KHG	<b>K</b> rankenhausgesetz
KIS	<b>K</b> rankenhausinformationssystem
NYHA	<b>N</b> ew <b>Y</b> ork <b>H</b> eart <b>A</b> ssociation
OPS / OPS301	<b>O</b> perationen- und <b>P</b> rozedurenschlüssel
Oracle	Kommerzieller Anbieter von Datenbankprodukten
QS	<b>Q</b> ualitätssicherung
RDBMS	<b>R</b> elationales <b>D</b> atenbank- <b>M</b> anagement- <b>S</b> ystem
SQL	<b>S</b> tructured <b>Q</b> uery <b>L</b> anguage = strukturierte Abfragesprache
SQL-DDL	<b>S</b> tructured <b>Q</b> uery <b>L</b> anguage – <b>D</b> ata <b>D</b> efinition <b>L</b> anguage (zur Implementierung von Datenstrukturen)
SQL-DML	<b>S</b> tructured <b>Q</b> uery <b>L</b> anguage – data <b>M</b> anipulation <b>L</b> anguage (zur Manipulation von Daten)



## Literaturverzeichnis

1.  
Jaster HJ, Kalmar P, Struck E  
Qualitätssicherung in der Herzchirurgie  
In: Dokumentationsverfahren in der Herzchirurgie  
1. Auflage  
Steinkopff, Darmstadt 1995; 63-73
2.  
Bundesgeschäftsstelle Qualitätssicherung gGmbH:  
<http://www.bqs-online.com/>
3.  
Datensatz Herzchirurgie:  
<http://www.bqs-online.com/download/boegen-2008/HCH.pdf>
4.  
§ 17 b KHG:  
[http://www.g-drg.de/systemgrundlagen/drg\\_17khg.php?m=1](http://www.g-drg.de/systemgrundlagen/drg_17khg.php?m=1)
5.  
DIMDI:  
<http://www.dimdi.de/dynamic/de/index.html>
6.  
ICD-10-GM 2007:  
<http://www.dimdi.de/static/de/klassi/diagnosen/icd10/htmlgm2007/fr-icd.htm>
7.  
OPS 2007:  
<http://www.dimdi.de/static/de/klassi/prozeduren/ops301/opshtml2007/fr-ops.htm>
8.  
Deutsche Kodierrichtlinien, Version 2007:  
[http://www.g-drg.de/deutschesdrg/drg\\_kodier.php?m=11](http://www.g-drg.de/deutschesdrg/drg_kodier.php?m=11)
9.  
Jeibmann M, Radu M, Krian A  
Das Duisburger Modell eines Klinikinformationssystems  
In: Dokumentationsverfahren in der Herzchirurgie  
1. Auflage  
Steinkopff, Darmstadt 1995; 25-33

10.  
Benninghoff A, Beckmann A, Radu M, Jeibmann M, Krian A  
Normalisierte Datenstrukturen für die "Qualitätssicherung Herzchirurgie" nach dem relationalen Modell  
In: Informationsmanagement in der Herzchirurgie VII  
1. Auflage  
Schöningh Verlag, Münster 2002; 79-87
11.  
Vereinbarung zum Datenaustausch gemäß § 301 SGB V:  
<http://www.dkgev.de/pdf/1951.pdf>
12.  
Das KIS ORBIS der Fa. Agfa HealthCare:  
[http://www.agfa.com/germany/de/he/solutions/krankenhausweite\\_it/index.jsp](http://www.agfa.com/germany/de/he/solutions/krankenhausweite_it/index.jsp)
13.  
Meier A  
Relationale Datenbanken - Eine Einführung für die Praxis  
2. Auflage  
Springer-Verlag, Berlin, Heidelberg, New York 1995
14.  
Codd, E. F.  
A Relational Model of Data for Large Shared Data Banks  
Communications of the ACM 1970; Vol. 13, No. 6: 377-387
15.  
Chen P  
The entity-relationship model—toward a unified view of data  
ACM Transactions on Database Systems (TODS) 1976; Special issue: papers from the international conference on very large data bases: September 22-24, 1975, Framingham, MA, Volume 1, Issue 1: 9-36
16.  
Martin J  
Information Engineering, Book III: Design & Construction  
1. Auflage  
Prentice Hall, Englewood Cliffs/NJ 1990
17.  
Oracle Designer:  
<http://www.oracle.com/technology/products/designer/index.html>
18.  
Ullmann C  
Java ist auch eine Insel  
2., aktualisierte und erweiterte Auflage 2003  
Galileo Computing 2003

19.  
Middendorf S, Singer R, Heid J  
Java Programmierhandbuch und Referenz  
3., überarbeitete und erweiterte Auflage 2003  
dpunkt.Verlag, Heidelberg 2003
20.  
Java SE:  
<http://java.sun.com/javase/>
21.  
Netbeans:  
<http://www.netbeans.org/>
22.  
Ausfüllhinweise HCH:  
[http://www.bqs-online.com/download/boegen-2007/Ausfuellhinweise\\_HCH.htm](http://www.bqs-online.com/download/boegen-2007/Ausfuellhinweise_HCH.htm)
23.  
Kuhlmann G, Müllmerstadt F  
Datenbanksprache SQL für DOS, Windows, OS/2 und Unix - Eine strukturierte Einführung  
Originalausgabe  
Rowohlt Taschenbuch Verlag GmbH, Reinbek bei Hamburg 1994
24.  
Benninghoff A  
Kurs Datenbankdesign/SQL - Kursteil Datenbankdesign  
Schulungsunterlage für den Workshop "Dokumentationsverfahren in der Herzchirurgie"  
2000
25.  
NYHA-Klassifikation:  
<http://flexikon.doccheck.com/NYHA-Stadium>
26.  
CCS-Klassifikation:  
<http://flexikon.doccheck.com/CCS-Klassifikation>
27.  
CCS:  
[http://www.ccs.ca/home/index\\_e.aspx](http://www.ccs.ca/home/index_e.aspx)
28.  
Modifizierte Rankin-Skala:  
[http://www.bqs-outcome.de/2006/ergebnisse/leistungsbereiche/karotis/buaw/index\\_html/0006\\_Def\\_Modifizierte\\_Rankin-Skala\\_zur\\_Qualif.html](http://www.bqs-outcome.de/2006/ergebnisse/leistungsbereiche/karotis/buaw/index_html/0006_Def_Modifizierte_Rankin-Skala_zur_Qualif.html)
29.  
BNF Grammar for ISO/IEC 9075:1992 - Database Language SQL (SQL-92):  
<http://savage.net.au/SQL/sql-92.bnf.html>

30.  
JDBC-Spezifikation:  
<http://java.sun.com/products/jdbc/overview.html>
31.  
JavaDoc-Homepage:  
<http://java.sun.com/j2se/javadoc/>
32.  
API der Klasse java.awt.Robot:  
<http://java.sun.com/javase/6/docs/api/java/awt/Robot.html>
33.  
Koronarangiographie und Perkutane Koronarintervention (PCI):  
<http://www.bqs-online.com/download/boegen-2008/21-3.pdf>
34.  
MaxDB:  
<https://www.sdn.sap.com/irj/sdn/maxdb-downloads>
35.  
HL7:  
<http://www.hl7.org/>
36.  
Execute Query:  
<http://www.executequery.org/>
37.  
SQuirreL SQL:  
<http://www.squirrelsql.org/>
38.  
DB Designer Fork:  
<http://sourceforge.net/projects/dbdesigner-fork/>
39.  
Free Software Foundation (FSF):  
<http://www.fsf.org/>

# Anhang 1 – Modul HCH der BQS (Musterbögen)

MUSTER - Nicht zur Dokumentation verwenden



## Datensatz Herzchirurgie

HCH

Basis	ANAMNESE / BEFUND	
<b>PATIENTENBASISDATEN</b>		
1 Institutionskennzeichen □□□□□□□□	9 klinischer Schweregrad der Herzinsuffizienz (NYHA-Klassifikation)  1 = (I): Beschwerdefreiheit, normale körperliche Belastbarkeit 2 = (II): Beschwerden bei stärkerer körperlicher Belastung 3 = (III): Beschwerden bei leichter körperlicher Belastung 4 = (IV): Beschwerden in Ruhe	15 pulmonale Hypertonie  0 = nein 1 = ja 9 = unbekannt
1.1 Betriebsstätten-Nummer □□	10 Angina Pectoris (nach CCS)  0 = nein 1 = CCS I (Angina pectoris bei schwerer Belastung) 2 = CCS II (Angina pectoris bei mittlerer Belastung) 3 = CCS III (Angina pectoris bei leichter Belastung) 4 = CCS IV (Angina pectoris in Ruhe)	16 Herzrhythmus bei Aufnahme  1 = Sinusrhythmus 2 = Vorhofflimmern 8 = anderer Rhythmus
2 Fachabteilung § 301-Vereinbarung □□□□	11 Infarkt(e)  0 = nein 1 = ja, letzte(r) innerhalb der letzten 48 Stunden 2 = ja, letzte(r) innerhalb der letzten 21 Tage 3 = ja, letzte(r) länger als 21 Tage, weniger als 91 Tage zurück 4 = ja, letzte(r) länger als 91 Tage zurück 8 = ja, letzter Zeitpunkt unbekannt 9 = unbekannt	17 Patient ist Schrittmacher- / Defi-Träger  0 = nein 1 = ja, Schrittmacher 2 = ja, Defibrillator
Schlüssel 1	12 kardiogener Schock / Dekompensation  0 = nein 1 = ja, letzte(r) innerhalb der letzten 48 Stunden 2 = ja, letzte(r) innerhalb der letzten 21 Tage 3 = ja, letzte(r) länger als 21 Tage 8 = ja, letzter Zeitpunkt unbekannt 9 = unbekannt	18 Einstufung nach ASA-Klassifikation  1 = normaler, ansonsten gesunder Patient 2 = Patient mit leichter Allgemeinerkrankung und Leistungseinschränkung 3 = Patient mit schwerer Allgemeinerkrankung und Leistungseinschränkung 4 = Patient mit inaktivierender Allgemeinerkrankung, ständige Lebensbedrohung 5 = moribunder Patient
3 Identifikationsnummer des Patienten □□□□□□□□□□□□□□□□ □□	13 Reanimation  0 = nein 1 = ja, letzte(r) innerhalb der letzten 48 Stunden 2 = ja, letzte(r) innerhalb der letzten 21 Tage 3 = ja, letzte(r) länger als 21 Tage 8 = ja, letzter Zeitpunkt unbekannt 9 = unbekannt	
4 Aufnahmedatum Krankenhaus TT.MM.JJ.JJ □□.□□.□□□□	14 Patient wird beatmet  0 = nein 1 = ja	
5 Geburtsdatum TT.MM.JJ.JJ □□.□□.□□□□		
6 Geschlecht  1 = männlich 2 = weiblich		
7 Körpergröße  999 = unbekannt cm □□□□		
8 Körpergewicht bei Aufnahme  999 = unbekannt kg □□□□		

Stand 30.06.2008 - gültig ab dem 01.01.2007  
© BQS Bundesgeschäftsstelle Qualitätssicherung gGmbH

Abbildung A-1.1.: Muster „Datensatz Herzchirurgie“ der BQS, Bogen 1

KARDIALE BEFUNDE	VORAUSGEGANGENE(R) INTERVENTIONELLE(R) KORONAREINGRIFF(E)	AKTUELLE WEITERE ERKRANKUNG(EN)	
<p>19 LVEF <input type="checkbox"/></p> <p>1 = schlecht oder &lt;30% 2 = mittel oder 30% - 50% 3 = gut oder &gt;50%</p> <p>20 Koronarangiographiebefund <input type="checkbox"/></p> <p>0 = keine KHK 1 = 1-Gefäßerkrankung 2 = 2-Gefäßerkrankung 3 = 3-Gefäßerkrankung</p> <p>21 signifikante Hauptstammstenose <input type="checkbox"/></p> <p>0 = nein 1 = ja, gleich oder größer 50% 9 = unbekannt</p>	<p>22 PCI <input type="checkbox"/></p> <p>0 = nein 1 = ja</p> <tr> <th style="background-color: #cccccc;">VOR-OP(S) AN HERZ/AORTA</th> </tr> <p>23 Anzahl <input type="checkbox"/></p> <p>8 = genaue Anzahl unbekannt (aber mind. 1) 9 = unbekannt</p>	VOR-OP(S) AN HERZ/AORTA	<p><b>weitere Begleiterkrankungen</b></p> <p>24 akute Infektion(en) <input type="checkbox"/></p> <p>1. <input type="checkbox"/> <input type="checkbox"/> 2. <input type="checkbox"/> <input type="checkbox"/> 3. <input type="checkbox"/> <input type="checkbox"/></p> <p>Schlüssel 2</p> <p>25 Diabetes mellitus <input type="checkbox"/></p> <p>0 = nein 1 = ja, diätetisch behandelt 2 = ja, orale Medikation 3 = ja, mit Insulin behandelt 4 = ja, unbehandelt 9 = unbekannt</p> <p>26 arterielle Gefäßerkrankung <input type="checkbox"/></p> <p>0 = nein 1 = ja 9 = unbekannt</p> <p><b>wenn Feld 26 = 1</b></p> <p>27 periphere AVK (Extremitäten) <input type="checkbox"/></p> <p>0 = nein 1 = ja 9 = unbekannt</p> <p>28 Arteria Carotis <input type="checkbox"/></p> <p>0 = nein 1 = ja 9 = unbekannt</p> <p>29 Aortenaneurysma <input type="checkbox"/></p> <p>0 = nein 1 = ja 9 = unbekannt</p> <p>30 sonstige arterielle Gefäßerkrankung(en) <input type="checkbox"/></p> <p>0 = nein 1 = ja 9 = unbekannt</p>
VOR-OP(S) AN HERZ/AORTA			

Stand 30.06.2006 - gültig ab dem 01.01.2007

© BQS Bundesgeschäftsstelle Qualitätssicherung gGmbH

Abbildung A-1.2.

31	Lungenerkrankung(en)	<input type="checkbox"/>
	0 = nein 1 = ja, COPD mit Dauermedikation 2 = ja, COPD ohne Dauermedikation 8 = ja, andere Lungenerkrankungen 9 = unbekannt	
32	neurologische Erkrankung(en)	<input type="checkbox"/>
	0 = nein 1 = ja, ZNS, zerebrovaskulär (Blutung, Ischämie) 2 = ja, ZNS, andere 3 = ja, peripher 4 = ja, Kombination 9 = unbekannt	
<b>wenn Neurologische Erkrankungen = 1 (ZNS, zerebrovaskulär):</b>		
33	Schweregrad der Behinderung	<input type="checkbox"/>
	0 = Rankin 0: kein neurologisches Defizit nachweisbar 1 = Rankin 1: Apoplex mit funktionell irrelevantem neurologischen Defizit 2 = Rankin 2: leichter Apoplex mit funktionell geringgradigem Defizit und / oder leichter Aphasie 3 = Rankin 3: mittelschwerer Apoplex mit deutlichem Defizit mit erhaltener Gehfähigkeit und / oder mittelschwerer Aphasie 4 = Rankin 4: schwerer Apoplex, Gehen nur mit Hilfe möglich und / oder komplette Aphasie 5 = Rankin 5: invalidisierender Apoplex: Patient ist bettlägerig bzw. rollstuhlpflichtig	
34	präoperative Nierenersatztherapie	<input type="checkbox"/>
	0 = nein 1 = akut 2 = chronisch	
<b>wenn Feld 34 = 0</b>		
35	Kreatininwert i. S. in mg/dl (präoperativ)	mg/dl <input type="text"/> <input type="text"/> <input type="text"/> , <input type="text"/> <input type="text"/>
36	Kreatininwert i. S. in µmol/l (präoperativ)	µmol/l <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>

Abbildung A-1.3.

Operation	
Mindestens ein Bogen muss ausgefüllt werden	
OP-BASISDATEN	
37	OP-Protokoll Nummer <input type="text"/> <input type="text"/>
38	OP-Datum TT.MM.JJJJ <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>
KATEGORIEN DURCHGEFÜHRTER EINGRIFFE	
39	Koronarchirurgie <input type="checkbox"/> 0 = nein 1 = ja
40	Aortenklappenchirurgie <input type="checkbox"/> 0 = nein 1 = ja
41	sonstige OP <input type="checkbox"/> 0 = nein 1 = ja
42	Operation (OPS) 1. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/> 2. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/> 3. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/> 4. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/> 5. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/> 6. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/> 7. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/> 8. <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> : <input type="text"/>
43	Dringlichkeit <input type="checkbox"/> 1 = elektiv 2 = dringlich 3 = Notfall 4 = Notfall (Reanimation / ultima ratio)
44	Nitrate (i. v.) <input type="checkbox"/> 0 = nein 1 = ja
45	Troponin positiv <input type="checkbox"/> 0 = nein 1 = ja
46	Inotrope (i. v.) <input type="checkbox"/> 0 = nein 1 = ja
47	(präoperativ) mechanische Kreislaufunterstützung <input type="checkbox"/> 0 = nein 1 = ja, IABP 2 = ja, andere
48	Wundkontaminationsklassifikation (nach Definition der CDC) Erläuterungen i. d. Ausfüllhinweisen <input type="checkbox"/> 1 = aseptische Eingriffe 2 = bedingt aseptische Eingriffe 3 = kontaminierte Eingriffe 4 = septische Eingriffe
49	OP-Vorgehen <input type="checkbox"/> 1 = mt HLM 2 = ohne HLM 3 = Umstieg von OP ohne HLM auf OP mit HLM
50	Zugang <input type="checkbox"/> 1 = konventionelle Sternotomie 2 = andere
51	OP-Zeit (Schnitt-Nahtzeit) <input type="text"/> <input type="text"/> <input type="text"/> Minuten 999 = unbekannt
52	Bypasszeit <input type="text"/> <input type="text"/> <input type="text"/> min 999 = unbekannt
53	Aortenabklemmzeit <input type="text"/> <input type="text"/> <input type="text"/> min 999 = unbekannt
KORONARCHIRURGIE	
wenn Feld 39 = 1	
54	Anzahl der Grafts <input type="text"/>
Art der Grafts	
bei Graft(s)	
55	Vene <input type="checkbox"/> 1 = ja
56	ITA links <input type="checkbox"/> 1 = ja
57	ITA rechts <input type="checkbox"/> 1 = ja
58	A. radialis <input type="checkbox"/> 1 = ja
59	sonstige Grafts <input type="checkbox"/> 1 = ja
wenn Feld 39 = 1	
60	zentrale Anastomose(n) <input type="text"/> Anzahl
61	periphere Anastomose(n), arteriell <input type="text"/> Anzahl
62	periphere Anastomose(n), venös <input type="text"/> Anzahl
63	TEA <input type="checkbox"/> 0 = nein 1 = ja, links 2 = ja, rechts 3 = ja, beidseits
Versorgte Koronararterie(n)	
bei Graft(s)	
64	LAD und / oder Äste <input type="checkbox"/> 1 = ja

Abbildung A-1.4.



<p>65 RCA und / oder Äste <input type="checkbox"/></p> <p style="padding-left: 20px;">1 = ja</p> <p>66 RCX und / oder Äste <input type="checkbox"/></p> <p style="padding-left: 20px;">1 = ja</p> <hr/> <p><b>wenn Feld 39 = 1</b></p> <p>67 anderer Koronareingriff <input type="checkbox"/></p> <p style="padding-left: 20px;">0 = nein 1 = ja</p>	<p><b>AORTENKLAPPENCHIRURGIE</b></p> <p><b>wenn Feld 40 = 1</b></p> <p>68 Stenose <input type="checkbox"/></p> <p style="padding-left: 20px;">0 = nein 1 = ja</p> <p>69 Insuffizienz <input type="checkbox"/></p> <p style="padding-left: 20px;">0 = nein 1 = ja</p> <p>70 Klappeneingriff <input type="checkbox"/> <input type="checkbox"/></p> <p style="padding-left: 20px;">Schlüssel 3</p> <hr/> <p><b>bei Ersatz der Aortenklappe</b></p> <p>71 Ventil- bzw. Ringtyp <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/></p> <p style="padding-left: 20px;">Schlüssel 4</p> <p>72 Durchmesser <input type="checkbox"/> <input type="checkbox"/></p> <p style="padding-left: 20px;">mm</p>
--	---

**Abbildung A-1.5.**

Basis					
POSTOPERATIVER VERLAUF					
73	Dauer Intensivaufenthalt  Tage <input type="text"/> <input type="text"/> <input type="text"/>	79	Rethorakotomie / Grund  1. <input type="text"/> <input type="text"/> 2. <input type="text"/> <input type="text"/> 3. <input type="text"/> <input type="text"/> 4. <input type="text"/> <input type="text"/>  0 = keine Rethorakotomie / kein weiterer Grund 1 = Blutung/Hämatom 2 = Low Cardiac Output 3 = Tamponade 4 = Gratprobleme, Ischämie 5 = Wundinfektion, Mediastinitis 6 = Dissektion 7 = Prothesendysfunktion 8 = instabiles Sternum 9 = Chylothorax 88 = anderer Grund	<b>wenn Dauer über 24 Stunden</b>	
74	Mobilisation  0 = Patient ist/wurde nicht mobilisiert 1 = Aufstehen am 1. postoperativen Tag 2 = Aufstehen am 2. postoperativen Tag 3 = Aufstehen am 3. postoperativen Tag oder später 9 = nicht beurteilbar	80	respiratorische Insuffizienz  0 = nein 1 = ja, mit forcierter Atemtherapie 2 = ja, Reintubation 3 = ja, Tracheotomie	83 <b>Schweregrad eines neurologischen Defizits bei Entlassung</b>  0 = Rankin 0: kein neurologisches Defizit nachweisbar 1 = Rankin 1: Apoplex mit funktionell irrelevantem neurologischem Defizit 2 = Rankin 2: leichter Apoplex mit funktionell geringgradigem Defizit und / oder leichter Aphasie 3 = Rankin 3: mittelschwerer Apoplex mit deutlichem Defizit mit erhaltener Gehfähigkeit und / oder mittelschwerer Aphasie 4 = Rankin 4: schwerer Apoplex, Gehen nur mit Hilfe möglich und / oder komplette Aphasie 5 = Rankin 5: invalidisierender Apoplex: Patient ist bettlägerig bzw. rollstuhlpflichtig 6 = Rankin 6: Apoplex mit tödlichem Ausgang	
75	Mediastinitis  0 = nein 1 = ja	81	zerebrovaskuläres Ereignis bis zur Entlassung  0 = nein 1 = ja	84 <b>Psychosyndrom</b>  0 = nein 1 = ja, flüchtig, ohne Therapie 2 = ja, therapierelevant, mit Therapie	
76	Reanimation  0 = nein 1 = ja	<b>wenn Feld 81 = 1</b>		85 <b>therapiepflichtige gastrointestinale Komplikation(en)</b>  0 = nein 1 = ja, konservativ 2 = ja, interventionell 3 = ja, Laparotomie	
77	Myokardinfarkt  0 = nein 1 = ja	82	Dauer des zerebrovaskulären Ereignisses  1 = bis einschl. 24 Stunden 2 = mehr als 24 Stunden bis einschl. 72 Stunden 3 = über 72 Stunden	86 <b>Kreatininwert i.S. in mg/dl (postoperativ)</b>  mg/dl <input type="text"/> <input type="text"/> , <input type="text"/>	
78	Low Cardiac Output (bei erneuter OP/IEKZ bitte weiteren OP-Bogen ausfüllen)  0 = nein 1 = ja, konservativ therapiert 2 = ja, IABP 3 = ja, andere mechanische Kreislaufunterstützung			87 <b>Kreatininwert i.S. in µmol/l (postoperativ)</b>  µmol/l <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	
				88 <b>postoperative Nierenersatztherapie</b>  0 = nein 1 = akut 2 = chronisch	
				89 <b>Erythrozytenkonzentrat(e)</b>  Einheiten <input type="text"/> <input type="text"/> 98 = genaue Zahl nicht bekannt (aber mind. 1) 99 = unbekannt	
				90 <b>FFP</b> 00 = keine  Einheiten <input type="text"/> <input type="text"/>	

Abbildung A-1.6.

<p>91 Thrombozytenkonzentrat <input type="checkbox"/></p> <p>0 = nein 1 = ja</p>	<p><b>BEI ENDE DER BEHANDLUNG</b></p> <p>92 Herzrhythmus bei Entlassung <input type="checkbox"/></p> <p>1 = Sinusrhythmus 2 = Vorhofflimmern 8 = anderer Rhythmus</p> <p>93 Patient trägt Schrittmacher / Defibrillator <input type="checkbox"/></p> <p>0 = nein 1 = ja, Schrittmacher 2 = ja, Defibrillator</p>	<p><b>ENTLASSUNG / VERLEGUNG</b></p> <p>94 Entlassungsdiagnose(n) (ICD-10)</p> <p>1. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>2. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>3. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>4. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>5. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>6. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>7. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>8. <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>95 Entlassungsdatum Krankenhaus TT.MM.JJJJ <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>96 Entlassungsgrund § 301-Vereinbarung <input type="checkbox"/> <input type="checkbox"/></p> <p>Schlüssel 5</p>
--	--	---

Stand 30.06.2008 - gültig ab dem 01.01.2007  
© BQS Bundesgeschäftsstelle Qualitätssicherung ggmbH

**Abbildung A-1.7.**

Follow-Up	
NACHBEOBACHTUNG / FOLLOW-UP	
97	<b>Erhebungsdatum</b> TT.MM.JJJJ  <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>
98	<b>Status des Patienten am 30. postoperativen Tag</b> (nach der ersten OP)  <input type="checkbox"/> 0 = Patient ist verstorben 1 = Patient lebt 8 = Falldokumentation vor dem 30. Tag postoperativ abgeschlossen / Follow Up Information liegt noch nicht vor, wird nachgeschickt 9 = Situation des Patienten ist unbekannt  <b>wenn Feld 98 = 0</b>
99	<b>Todesdatum</b> TT.MM.JJJJ  <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> . <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>

Stand 30.06.2006 - gültig ab dem 01.01.2007  
 © BQS Bundesgeschäftsstelle Qualitätssicherung gGmbH

**Abbildung A-1.8.**

## Schlüssel 1

100 = Innere Medizin  
 102 = Innere Medizin/Schwerpunkt Geriatrie  
 103 = Innere Medizin/Schwerpunkt Kardiologie  
 104 = Innere Medizin/Schwerpunkt Nephrologie  
 105 = Innere Medizin/Schwerpunkt Hämatologie und internistische Onkologie  
 106 = Innere Medizin/Schwerpunkt Endokrinologie  
 107 = Innere Medizin/Schwerpunkt Gastroenterologie  
 108 = Innere Medizin/Schwerpunkt Pneumologie  
 109 = Innere Medizin/Schwerpunkt Rheumatologie  
 114 = Innere Medizin/Schwerpunkt Lungen- und Bronchialheilkunde  
 150 = Innere Medizin/Tumorforschung  
 151 = Innere Medizin/Schwerpunkt Coloproktologie  
 152 = Innere Medizin/Schwerpunkt Infektionskrankheiten  
 153 = Innere Medizin/Schwerpunkt Diabetes  
 154 = Innere Medizin/Schwerpunkt Naturheilkunde  
 156 = Innere Medizin/Schwerpunkt Schlaganfallpatienten  
 200 = Geriatrie  
 224 = Geriatrie/Schwerpunkt Frauenheilkunde  
 260 = Tagesklinik (für teilstationäre Pflegesätze)  
 261 = Nachtambulanz (für teilstationäre Pflegesätze)  
 300 = Kardiologie  
 400 = Nephrologie  
 410 = Nephrologie/Schwerpunkt Pädiatrie  
 500 = Hämatologie und internistische Onkologie  
 510 = Hämatologie und internistische Onkologie/Schwerpunkt Pädiatrie  
 524 = Hämatologie und internistische Onkologie/Schwerpunkt Frauenheilkunde  
 533 = Hämatologie und internistische Onkologie/Schwerpunkt Strahlenheilkunde  
 600 = Endokrinologie  
 607 = Endokrinologie/Schwerpunkt Gastroenterologie  
 610 = Endokrinologie/Schwerpunkt Pädiatrie  
 700 = Gastroenterologie  
 706 = Gastroenterologie/Schwerpunkt Endokrinologie  
 710 = Gastroenterologie/Schwerpunkt Pädiatrie  
 800 = Pneumologie  
 900 = Rheumatologie  
 910 = Rheumatologie/Schwerpunkt Pädiatrie  
 1000 = Pädiatrie  
 1004 = Pädiatrie/Schwerpunkt Nephrologie  
 1005 = Pädiatrie/Schwerpunkt Hämatologie und internistische Onkologie  
 1006 = Pädiatrie/Schwerpunkt Endokrinologie  
 1007 = Pädiatrie/Schwerpunkt Gastroenterologie  
 1009 = Pädiatrie/Schwerpunkt Rheumatologie  
 1011 = Pädiatrie/Schwerpunkt Kinderkardiologie  
 1012 = Pädiatrie/Schwerpunkt Neonatologie  
 1014 = Pädiatrie/Schwerpunkt Lungen- und Bronchialheilkunde  
 1028 = Pädiatrie/Schwerpunkt Kinderneurologie  
 1050 = Pädiatrie/Schwerpunkt Perinatalmedizin  
 1051 = Langzeitbereich Kinder  
 1100 = Kinderkardiologie  
 1136 = Kinderkardiologie/Schwerpunkt Intensivmedizin  
 1200 = Neonatologie  
 1300 = Kinderchirurgie  
 1400 = Lungen- und Bronchialheilkunde  
 1410 = Lungen- und Bronchialheilkunde/Schwerpunkt Pädiatrie  
 1500 = Allgemeine Chirurgie

1513 = Allgemeine Chirurgie/Schwerpunkt Kinderchirurgie  
 1516 = Allgemeine Chirurgie/Schwerpunkt Unfallchirurgie  
 1518 = Allgemeine Chirurgie/Schwerpunkt Gefäßchirurgie  
 1519 = Allgemeine Chirurgie/Schwerpunkt Plastische Chirurgie  
 1520 = Allgemeine Chirurgie/Schwerpunkt Thoraxchirurgie  
 1523 = Chirurgie/Schwerpunkt Orthopädie  
 1536 = Allgemeine Chirurgie/Intensivmedizin  
 1550 = Allgemeine Chirurgie/Schwerpunkt Abdominal- und Gefäßchirurgie  
 1551 = Allgemeine Chirurgie/Schwerpunkt Handchirurgie  
 1600 = Unfallchirurgie  
 1700 = Neurochirurgie  
 1800 = Gefäßchirurgie  
 1900 = Plastische Chirurgie  
 2000 = Thoraxchirurgie  
 2021 = Thoraxchirurgie/Schwerpunkt Herzchirurgie  
 2036 = Thoraxchirurgie/Intensivmedizin  
 2050 = Thoraxchirurgie/Schwerpunkt Herzchirurgie Intensivmedizin  
 2100 = Herzchirurgie  
 2120 = Herzchirurgie/Schwerpunkt Thoraxchirurgie  
 2136 = Herzchirurgie/Intensivmedizin (§ 13 Abs. 2 Satz 3 2. Halbsatz BPIV '95)  
 2150 = Herzchirurgie/Schwerpunkt Thoraxchirurgie Intensivmedizin  
 2200 = Urologie  
 2300 = Orthopädie  
 2309 = Orthopädie/Schwerpunkt Rheumatologie  
 2315 = Orthopädie/Schwerpunkt Chirurgie  
 2400 = Frauenheilkunde und Geburtshilfe  
 2402 = Frauenheilkunde/Schwerpunkt Geriatrie  
 2405 = Frauenheilkunde/Schwerpunkt Hämatologie und internistische Onkologie  
 2406 = Frauenheilkunde/Schwerpunkt Endokrinologie  
 2425 = Frauenheilkunde  
 2500 = Geburtshilfe  
 2600 = Hals-, Nasen-, Ohrenheilkunde  
 2700 = Augenheilkunde  
 2800 = Neurologie  
 2810 = Neurologie/Schwerpunkt Pädiatrie  
 2856 = Neurologie/Schwerpunkt Schlaganfallpatienten  
 2900 = Allgemeine Psychiatrie  
 2928 = Allgemeine Psychiatrie/Schwerpunkt Neurologie  
 2930 = Allgemeine Psychiatrie/Schwerpunkt Kinder- und Jugendpsychiatrie  
 2931 = Allgemeine Psychiatrie/Schwerpunkt Psychosomatik/Psychotherapie  
 2950 = Allgemeine Psychiatrie/Schwerpunkt Suchtblindung  
 2951 = Allgemeine Psychiatrie/Schwerpunkt Gerontopsychiatrie  
 2952 = Allgemeine Psychiatrie/Schwerpunkt Forensische Behandlung  
 2960 = Allgemeine Psychiatrie/Tagesklinik (für teilstationäre Pflegesätze)  
 2961 = Allgemeine Psychiatrie/Nachtambulanz (für teilstationäre Pflegesätze)  
 3000 = Kinder- und Jugendpsychiatrie  
 3060 = Kinder- und Jugendpsychiatrie/Tagesklinik (für teilstationäre Pflegesätze)  
 3061 = Kinder- und Jugendpsychiatrie/Nachtambulanz (für teilstationäre Pflegesätze)  
 3100 = Psychosomatik/Psychotherapie  
 3200 = Nuklearmedizin  
 3233 = Nuklearmedizin/Schwerpunkt Strahlenheilkunde  
 3300 = Strahlenheilkunde

3305 = Strahlenheilkunde/Schwerpunkt Hämatologie und internistische Onkologie  
 3350 = Strahlenheilkunde/Schwerpunkt Radiologie  
 3400 = Dermatologie  
 3460 = Dermatologie/Tagesklinik  
 3500 = Zahn- und Kieferheilkunde, Mund- und Kieferchirurgie  
 3600 = Intensivmedizin  
 3601 = Intensivmedizin/Schwerpunkt Innere Medizin  
 3603 = Intensivmedizin/Schwerpunkt Kardiologie  
 3610 = Intensivmedizin/Schwerpunkt Pädiatrie  
 3617 = Intensivmedizin/Schwerpunkt Neurochirurgie  
 3618 = Intensivmedizin/Schwerpunkt Chirurgie  
 3621 = Intensivmedizin/Schwerpunkt Herzchirurgie  
 3622 = Intensivmedizin/Schwerpunkt Urologie  
 3624 = Intensivmedizin/Schwerpunkt Frauenheilkunde und Geburtshilfe  
 3626 = Intensivmedizin/Schwerpunkt Hals-, Nasen-, Ohrenheilkunde  
 3628 = Intensivmedizin/Schwerpunkt Neurologie  
 3650 = Operative Intensivmedizin/Schwerpunkt Chirurgie  
 3651 = Intensivmedizin/Thorax-Herzchirurgie  
 3652 = Intensivmedizin/Herz-Thoraxchirurgie  
 3700 = sonstige Fachabteilung  
 3750 = Angiologie  
 3751 = Radiologie/Röntgendiagnostik  
 3752 = Palliativmedizin  
 3753 = Schmerztherapie  
 3754 = Heiltherapeutische Abteilung  
 3755 = Wirbelsäulenchirurgie

## Schlüssel 2

0 = keine  
 1 = Mediastinitis  
 2 = Bakteriämie  
 3 = broncho-pulmonale Infektion  
 4 = oto-laryngologische Infektion  
 5 = floride Endokarditis  
 6 = Peritonitis  
 7 = Wundinfektion Thorax  
 8 = Pleuraempyem  
 9 = Venenkatheterinfektion  
 10 = Harnwegsinfektion  
 11 = Wundinfektion untere Extremitäten  
 12 = HIV-Infektion  
 13 = Hepatitis B oder C  
 18 = andere Wundinfektion  
 88 = sonstige Infektion

## Schlüssel 3

11 = Ersatz der nativen Klappe durch mechanische Klappe  
 12 = Ersatz der nativen Klappe durch eine Bioklappe  
 18 = Ersatz der nativen Klappe durch eine andere Klappe  
 21 = Ersatz einer mechanischen Klappe durch eine mechanische Klappe  
 22 = Ersatz einer mechanischen Klappe durch eine Bioklappe  
 28 = Ersatz einer mechanischen Klappe durch eine andere Klappe  
 31 = Ersatz einer Bioklappe durch eine mechanische Klappe  
 32 = Ersatz einer Bioklappe durch eine Bioklappe  
 38 = Ersatz einer Bioklappe durch eine andere Klappe  
 41 = offene Exploration der Klappe ohne Korrektur

Abbildung A-1.9.

- 42 = Sprengung der Klappe (auch Kommissurotomie) und Rekonstruktion
- 43 = Entkalkung
- 44 = Prothesenfixation
- 45 = Prothesenreinigung (Thrombenentfernung)
- 51 = isolierte Sprengung der Klappe (auch Kommissurotomie)
- 52 = isolierte Rekonstruktion mit Implantat
- 53 = isolierte Rekonstruktion ohne Implantat
- 88 = anderer Klappeneingriff

- 17 = interne Verlegung mit Wechsel zwischen den Entgeltbereichen der DRG-Fallpauschalen, nach der BPFIV oder für besondere Einrichtungen nach § 17b Abs. 1 Satz 15 KHG
- 18 = Rückverlegung
- 19 = Entlassung vor Wiederaufnahme mit Neueinstufung
- 20 = Entlassung vor Wiederaufnahme mit Neueinstufung wegen Komplikation
- 21 = Entlassung mit nachfolgender Wiederaufnahme

#### Schlüssel 4

- 1101 = mech. Klappenvent. - Kugelventil
- 1102 = mech. Klappenvent. - Hubdeckelventil
- 1103 = mech. Klappenvent. - Kippscheibventil
- 1104 = mech. Klappenvent. - Zweiflügelklappe
- 1105 = mech. Klappenvent. - Dreiflügelklappe
- 1188 = mech. Klappenvent. - andere mechanische Klappe
- 1201 = biol. Klappenvent. - Porcin mit Stent
- 1202 = biol. Klappenvent. - Porcin ohne Stent
- 1203 = biol. Klappenvent. - Xeno-Perikard mit Stent
- 1204 = biol. Klappenvent. - Xeno-Perikard stentless
- 1205 = biol. Klappenvent. - andere Hetero mit Stent
- 1206 = biol. Klappenvent. - andere Hetero ohne Stent
- 1207 = biol. Klappenvent. - Homologe mit Stent
- 1208 = biol. Klappenvent. - Homologe ohne Stent
- 1209 = biol. Klappenvent. - Autologe mit Stent
- 1210 = biol. Klappenvent. - Autologe ohne Stent
- 1288 = biol. Klappenvent. - andere biologische Klappenventile
- 1388 = hybride Klappenventile / andere Ventile
- 2101 = mech. Ringimpl. - geschlossen flexibel
- 2102 = mech. Ringimpl. - geschlossen starr
- 2103 = mech. Ringimpl. - offen flexibel
- 2104 = mech. Ringimpl. - offen starr
- 2201 = biol. Ringimpl. - geschlossen
- 2202 = biol. Ringimpl. - offen
- 2301 = hybride Ringimpl. / andere Ringimpl. - geschlossen
- 2302 = hybride Ringimpl. / andere Ringimpl. - offen

#### Schlüssel 5

- 1 = Behandlung regulär beendet
- 2 = Behandlung regulär beendet, nachstationäre Behandlung vorgesehen
- 3 = Behandlung aus sonstigen Gründen beendet
- 4 = Behandlung gegen ärztlichen Rat beendet
- 5 = Zuständigkeitswechsel des Kostenträgers
- 6 = Verlegung in ein anderes Krankenhaus
- 7 = Tod
- 8 = Verlegung in ein anderes Krankenhaus im Rahmen einer Zusammenarbeit (§ 14 Abs. 5 Satz 2 BPFIV in der am 31.12.2003 geltenden Fassung)
- 9 = Entlassung in eine Rehabilitationseinrichtung
- 10 = Entlassung in eine Pflegeeinrichtung
- 11 = Entlassung in ein Hospiz
- 12 = interne Verlegung
- 13 = externe Verlegung zur psychiatrischen Behandlung
- 14 = Behandlung aus sonstigen Gründen beendet, nachstationäre Behandlung vorgesehen
- 15 = Behandlung gegen ärztlichen Rat beendet, nachstationäre Behandlung vorgesehen
- 16 = externe Verlegung mit Rückverlegung oder Wechsel zwischen den Entgeltbereichen der DRG-Fallpauschalen, nach der BPFIV oder für besondere Einrichtungen nach § 17b Abs. 1 Satz 15 KHG mit Rückverlegung

Stand 30.06.2006 - gültig ab dem 01.01.2007

© BQS Bundesgeschäftsstelle Qualitätssicherung gGmbH

## Abbildung A-1.10.

## Anhang 2 – Beispiele für zur Ableitung von Werten verwendbaren Codes

Diagnose	Wert	Rang	ICD-Code	ICD-Text
Herzrhythmus bei Aufnahme	Vorhofflimmern	1	I48.10	Vorhofflimmern: Paroxysmal
Herzrhythmus bei Aufnahme	Vorhofflimmern	1	I48.11	Vorhofflimmern: Chronisch
Herzrhythmus bei Aufnahme	Vorhofflimmern	1	I48.19	Vorhofflimmern: Nicht näher bezeichnet
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I42.80	Arrhythmogene rechtsventrikuläre Kardiomyopathie [ARVCM]
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I44.1	Atrioventrikulärer Block 2. Grades
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I44.2	Atrioventrikulärer Block 3. Grades
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I44.3	Sonstiger und nicht näher bezeichneter atrioventrikulärer Block
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I45.4	Unspezifischer intraventrikulärer Block
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I47.0	Ventrikuläre Arrhythmie durch Re-entry
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I47.1	Supraventrikuläre Tachykardie
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I47.2	Ventrikuläre Tachykardie
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I47.9	Paroxysmale Tachykardie, nicht näher bezeichnet
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I48.00	Vorhofflattern: Paroxysmal
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I48.01	Vorhofflattern: Chronisch
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I48.09	Vorhofflattern: Nicht näher bezeichnet
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I49.3	Ventrikuläre Extrasystolie
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I49.5	Sick-Sinus-Syndrom
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I49.8	Sonstige näher bezeichnete kardiale Arrhythmien
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	I49.9	Kardiale Arrhythmie, nicht näher bezeichnet
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	R00.0	Tachykardie, nicht näher bezeichnet
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	R00.1	Bradykardie, nicht näher bezeichnet
Herzrhythmus bei Aufnahme	anderer Rhythmus	2	R00.8	Sonstige und nicht näher bezeichnete Störungen des Herzschlages
Koronarangiographiebefund	3-Gefäßerkrankung	1	I25.13	Atherosklerotische Herzkrankheit: Drei-Gefäßerkrankung
Koronarangiographiebefund	2-Gefäßerkrankung	2	I25.12	Atherosklerotische Herzkrankheit: Zwei-Gefäßerkrankung
Koronarangiographiebefund	1-Gefäßerkrankung	3	I25.11	Atherosklerotische Herzkrankheit: Ein-Gefäßerkrankung

Prozedur	Wert	Rang	OPS-Code	OPS-Text
anderer Koronareingriff	ja	1	5-357.6	Operationen bei kongenitalen Gefäßanomalien: Koronargefäße
anderer Koronareingriff	ja	1	5-360.x	Desobliteration (Enderarteriektomie) der Koronararterien: Sonstige
anderer Koronareingriff	ja	1	5-360.y	Desobliteration (Enderarteriektomie) der Koronararterien: N.n.bez.
anderer Koronareingriff	ja	1	5-360.0	Desobliteration (Enderarteriektomie) der Koronararterien: Enderarteriektomie, offen chirurgisch
anderer Koronareingriff	ja	1	5-360.1	Desobliteration (Enderarteriektomie) der Koronararterien: Enderarteriektomie, offen chirurgisch, mit Patch
anderer Koronareingriff	ja	1	5-360.2	Desobliteration (Enderarteriektomie) der Koronararterien: Enderarteriektomie, offen chirurgisch, mit Einbringen eines Stent in eine Koronararterie
anderer Koronareingriff	ja	1	5-360.3	Desobliteration (Enderarteriektomie) der Koronararterien: Enderarteriektomie, offen chirurgisch, mit Einbringen von zwei Stents in eine Koronararterie
anderer Koronareingriff	ja	1	5-360.4	Desobliteration (Enderarteriektomie) der Koronararterien: Enderarteriektomie, offen chirurgisch, mit Einbringen eines Stent in mehrere Koronararterien
anderer Koronareingriff	ja	1	5-363.0	Anderer Revaskularisation des Herzens: Koronararterienpatch
anderer Koronareingriff	ja	1	5-363.1	Anderer Revaskularisation des Herzens: Koronararterienbypass-Revision
anderer Koronareingriff	ja	1	5-363.2	Anderer Revaskularisation des Herzens: Koronararterienbypass-Neuanlage
anderer Koronareingriff	ja	1	5-363.3	Anderer Revaskularisation des Herzens: Koronararterientransposition
anderer Koronareingriff	ja	1	5-369.x	Anderer Operationen an den Koronargefäßen: Sonstige
anderer Koronareingriff	ja	1	5-369.y	Anderer Operationen an den Koronargefäßen: N.n.bez.
anderer Koronareingriff	ja	1	5-369.0	Anderer Operationen an den Koronargefäßen: Naht (nach Verletzung)
anderer Koronareingriff	ja	1	5-369.1	Anderer Operationen an den Koronargefäßen: Korrektur eines Aneurysmas
anderer Koronareingriff	ja	1	5-369.2	Anderer Operationen an den



				Koronargefäßen: Verschluss einer erworbenen koronaren Fistel
anderer Koronareingriff	ja	1	5-369.3	Anderer Operationen an den Koronargefäßen: Rekonstruktion des Koronarostiums
anderer Koronareingriff	ja	1	5-369.4	Anderer Operationen an den Koronargefäßen: Sympathektomie der Koronararterien
anderer Koronareingriff	ja	1	5-369.5	Anderer Operationen an den Koronargefäßen: Verschluss von Kollateralgefäßen
Reanimation	ja	1	8-771	Kardiale oder kardiopulmonale Reanimation
Reanimation	ja	1	8-772	Operative Reanimation
Reanimation	ja	1	8-779	Anderer Reanimationsmaßnahmen

## Anhang 3 – SQL92-Skript zur Implementierung des Datenmodells

```
1 CREATE TABLE PRUEFUNG_AUSSCHLUSS (PRFA_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT
2 NULL, KUERZEL VARCHAR(20) NOT NULL, HINWEISTEXT VARCHAR(2000) NOT NULL, BEMERKUNG
3 VARCHAR(2000), PRF_ID INTEGER NOT NULL) ;
4 CREATE TABLE KONTEXT_KOMPLEX_WERT (KKW_ID INTEGER NOT NULL, K_ID INTEGER NOT NULL, KPW_ID
5 INTEGER NOT NULL, RANG INTEGER) ;
6 CREATE TABLE OPS301_VERSION (OPSVER_ID INTEGER NOT NULL, CODE VARCHAR(20) NOT
7 NULL, BEZEICHNUNG VARCHAR(200) NOT NULL, KUERZEL VARCHAR(20) NOT NULL, GUELTIG_AB DATE NOT
8 NULL, BEMERKUNG VARCHAR(2000)) ;
9 CREATE TABLE KIS (KIS_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT NULL, KUERZEL
10 VARCHAR(20) NOT NULL, BEMERKUNG VARCHAR(2000)) ;
11 CREATE TABLE ELEMENT_WERT (EW_ID INTEGER NOT NULL, W_ID INTEGER NOT NULL, E_ID INTEGER NOT
12 NULL) ;
13 CREATE TABLE STATUS (STATUS VARCHAR(1) NOT NULL, BEZEICHNUNG VARCHAR(100) NOT
14 NULL, KUERZEL VARCHAR(20) NOT NULL) ;
15 CREATE TABLE GESCHLECHT (G_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT NULL, KUERZEL
16 VARCHAR(20) NOT NULL, BEMERKUNG VARCHAR(2000)) ;
17 CREATE TABLE DIAGNOSE_WERT_ICD (DWI_ID INTEGER NOT NULL, DW_ID INTEGER NOT NULL, ICD_ID
18 INTEGER NOT NULL) ;
19 CREATE TABLE WERT_WERT (WW_ID INTEGER NOT NULL, WERT VARCHAR(20) NOT NULL, BEMERKUNG
20 VARCHAR(2000), W_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT NULL, KUERZEL VARCHAR(20)
21 NOT NULL) ;
22 CREATE TABLE KONTEXT_KOMPLEX_PHASE (KKPPH_ID INTEGER NOT NULL, PH_ID INTEGER NOT
23 NULL, KKP_ID INTEGER NOT NULL) ;
24 CREATE TABLE WERT (W_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT NULL, KUERZEL
25 VARCHAR(20) NOT NULL, BEMERKUNG VARCHAR(2000)) ;
26 CREATE TABLE ELEMENT_PROZEDUR_WERT (EPW_ID INTEGER NOT NULL, PW_ID INTEGER NOT NULL, E_ID
27 INTEGER NOT NULL) ;
28 CREATE TABLE DIAGNOSE_WERT_KOMBINATIONSTYP (DWKT_ID INTEGER NOT NULL, BEZEICHNUNG
29 VARCHAR(200) NOT NULL, KUERZEL VARCHAR(20) NOT NULL, BEMERKUNG VARCHAR(2000)) ;
30 CREATE TABLE WERT_BEREICH (WB_ID INTEGER NOT NULL, WERT_MINIMUM NUMERIC(10,3) NOT
31 NULL, WERT_MAXIMUM NUMERIC(10,3) NOT NULL, BEMERKUNG VARCHAR(2000), W_ID INTEGER NOT NULL) ;
32 CREATE TABLE KONTEXT_DIAGNOSE (KD_ID INTEGER NOT NULL, K_ID INTEGER NOT NULL, D_ID INTEGER
33 NOT NULL, RANG INTEGER NOT NULL) ;
34 CREATE TABLE FALL_KONTEXT_PRUEFUNG_HINWEIS (FKPHW_ID INTEGER NOT NULL, FKP_ID INTEGER NOT
35 NULL, HWT_ID INTEGER NOT NULL, REIHENFOLGE INTEGER NOT NULL, HINWEIS VARCHAR(2000) NOT NULL)
36 ;
37 CREATE TABLE IMPORTDATEI_FELD (IMPDATF_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT
38 NULL, KUERZEL VARCHAR(20) NOT NULL, REIHENFOLGE INTEGER NOT NULL, BEMERKUNG
39 VARCHAR(2000), IMPDAT_ID INTEGER NOT NULL, DATTYP_ID INTEGER NOT NULL) ;
40 CREATE TABLE PHASE_ZUORDNUNGSTYP (PHZT_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT
41 NULL, KUERZEL VARCHAR(20) NOT NULL, BEMERKUNG VARCHAR(2000)) ;
42 CREATE TABLE OPS301_THESAURUS (OPSTHES_ID INTEGER NOT NULL, OPS_ID INTEGER NOT
43 NULL, ALTERNATIVTEXT VARCHAR(200) NOT NULL) ;
44 CREATE TABLE FALL_OPS301 (FALLOPS_ID INTEGER NOT NULL, FALL_ID INTEGER NOT NULL, OPS_ID
45 INTEGER NOT NULL, OPSKAT_ID INTEGER NOT NULL, OPSLOK_ID INTEGER NOT NULL, DATUM DATE NOT
46 NULL, BEMERKUNG VARCHAR(2000)) ;
47 CREATE TABLE ELEMENT_PROZEDUR (EP_ID INTEGER NOT NULL, P_ID INTEGER NOT NULL, E_ID INTEGER
48 NOT NULL) ;
49 CREATE TABLE RELEVANZ (REL_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT NULL, KUERZEL
50 VARCHAR(20) NOT NULL, BEMERKUNG VARCHAR(2000)) ;
51 CREATE TABLE IMPORTDATEI_FELD_WERT (IMPDATFW_ID INTEGER NOT NULL, IMPDATF_ID INTEGER NOT
52 NULL, W_ID INTEGER NOT NULL, STANDARDWERT VARCHAR(200)) ;
53 CREATE TABLE PUFFER_FALL_KONTEXT (ID INTEGER NOT NULL, FALL_ID INTEGER, FALL_CODE
54 VARCHAR(200), NACHNAME VARCHAR(200), VORNAME VARCHAR(200), GEBURTSDATUM
55 TIMESTAMP, AUFNAHME_FA TIMESTAMP, ABGANG_FA TIMESTAMP, AUFNAHME_KH TIMESTAMP, ABGANG_KH
56 TIMESTAMP, K_ID INTEGER, KONTEXT_BEZEICHNUNG VARCHAR(200), KONTEXT_RANG NUMERIC, KD_ID
57 INTEGER, D_ID INTEGER, DW_ID INTEGER, DWK_ID INTEGER, DIAGNOSE_BEZEICHNUNG
58 VARCHAR(200), DIAGNOSE_WERT_BEZEICHNUNG VARCHAR(200), DIAGNOSE_WERT
59 VARCHAR(20), KONTEXT_PROZEDUR_KP_ID INTEGER, P_ID INTEGER, PW_ID INTEGER, PWK_ID
60 INTEGER, PROZEDUR_BEZEICHNUNG VARCHAR(200), PROZEDUR_WERT_BEZEICHNUNG
61 VARCHAR(200), PROZEDUR_WERT VARCHAR(20), KKP_ID INTEGER, KOMPLEX_KP_ID INTEGER, KPW_ID
```

```

INTEGER,KOMPLEX_BEZEICHNUNG VARCHAR(200),KOMPLEX_WERT_BEZEICHNUNG 62
VARCHAR(200),KOMPLEX_WERT VARCHAR(20),KW_ID INTEGER,W_ID INTEGER,WERT_BEZEICHNUNG 63
VARCHAR(200),WERT_WERT VARCHAR(20)) ; 64
CREATE TABLE PUFFER_FALL_OP_KONTEXT (ID INTEGER NOT NULL,FALL_ID INTEGER,FALL_CODE 65
VARCHAR(200),NACHNAME VARCHAR(200),VORNAME VARCHAR(200),GEBURTSDATUM 66
TIMESTAMP,AUFNAHME_FA TIMESTAMP,ABGANG_FA TIMESTAMP,AUFNAHME_KH TIMESTAMP,ABGANG_KH 67
TIMESTAMP,FALLOP_ID INTEGER,OP_CODE VARCHAR(200),OP_BEGINN TIMESTAMP,OP_ENDE 68
TIMESTAMP,K_ID INTEGER,KONTEXT_BEZEICHNUNG VARCHAR(200),KONTEXT_RANG INTEGER,KD_ID 69
INTEGER,D_ID INTEGER,DW_ID INTEGER,DWK_ID INTEGER,DIAGNOSE_BEZEICHNUNG 70
VARCHAR(200),DIAGNOSE_WERT_BEZEICHNUNG VARCHAR(200),DIAGNOSE_WERT 71
VARCHAR(20),KONTEXT_PROZEDUR_KP_ID INTEGER,P_ID INTEGER,PW_ID INTEGER,PWK_ID 72
INTEGER,PROZEDUR_BEZEICHNUNG VARCHAR(200),PROZEDUR_WERT_BEZEICHNUNG 73
VARCHAR(200),PROZEDUR_WERT VARCHAR(20),KKP_ID INTEGER,KOMPLEX_KP_ID INTEGER,KPW_ID 74
INTEGER,KOMPLEX_BEZEICHNUNG VARCHAR(200),KOMPLEX_WERT 75
VARCHAR(20),KOMPLEX_WERT_BEZEICHNUNG VARCHAR(200),KW_ID INTEGER,W_ID 76
INTEGER,WERT_BEZEICHNUNG VARCHAR(200),WERT_WERT VARCHAR(20)) ; 77
CREATE TABLE FALL_KONTEXT_PRUEFUNG (FKP_ID INTEGER NOT NULL,FALL_ID INTEGER NOT 78
NULL,KPRF_ID INTEGER NOT NULL,ZULETZT_AKTUALISIERT DATE NOT NULL) ; 79
CREATE TABLE PROZEDUR_WERT_KOMBINATIONSTYP (PWKT_ID INTEGER NOT NULL,BEZEICHNUNG 80
VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ; 81
CREATE TABLE DIAGNOSE_ICD (DI_ID INTEGER NOT NULL,D_ID INTEGER NOT NULL,ICD_ID INTEGER 82
NOT NULL) ; 83
CREATE TABLE FALL_OP_OPS301 (FOPOPS_ID INTEGER NOT NULL,FALLOP_ID INTEGER NOT 84
NULL,OPS_ID INTEGER NOT NULL,OPSKAT_ID INTEGER NOT NULL,OPSLOK_ID INTEGER NOT NULL) ; 85
CREATE TABLE ELEMENT_BEZIEHUNG (EBEZ_ID INTEGER NOT NULL,E_ID_1 INTEGER NOT NULL,E_ID_2 86
INTEGER NOT NULL,EBEZTYP_ID INTEGER NOT NULL,BEMERKUNG VARCHAR(2000) NOT NULL) ; 87
CREATE TABLE WERT_DIAGNOSE (WD_ID INTEGER NOT NULL,W_ID INTEGER NOT NULL,D_ID INTEGER 88
NOT NULL) ; 89
CREATE TABLE ELEMENT_WERT_WERT (EWW_ID INTEGER NOT NULL,WW_ID INTEGER NOT NULL,E_ID 90
INTEGER NOT NULL) ; 91
CREATE TABLE OPS301 (OPS_ID INTEGER NOT NULL,OPSVER_ID INTEGER NOT NULL,CODE VARCHAR(20) 92
NOT NULL,KURZTEXT VARCHAR(2000) NOT NULL,VOLLTEXT VARCHAR(2000) NOT 93
NULL,INTERNE_BEZEICHNUNG VARCHAR(200)) ; 94
CREATE TABLE KOMPLEX_WERT_D_WERT (KPWDW_ID INTEGER NOT NULL,KPW_ID INTEGER NOT 95
NULL,DW_ID INTEGER NOT NULL) ; 96
CREATE TABLE ELEMENT_DIAGNOSE_WERT (EDW_ID INTEGER NOT NULL,DW_ID INTEGER NOT NULL,E_ID 97
INTEGER NOT NULL) ; 98
CREATE TABLE FALL_WERT (FW_ID INTEGER NOT NULL,WERT VARCHAR(20) NOT NULL,W_ID INTEGER 99
NOT NULL,FALL_ID INTEGER NOT NULL) ; 100
CREATE TABLE PROZEDUR_WERT (PW_ID INTEGER NOT NULL,P_ID INTEGER NOT 101
NULL,EJN_ID_AUSWERTUNG INTEGER NOT NULL,WERT VARCHAR(20) NOT NULL,BEZEICHNUNG 102
VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,RANG INTEGER,BEMERKUNG 103
VARCHAR(2000)) ; 104
CREATE TABLE PROZEDUR_WERT_KOMBINATION (PWKT_ID INTEGER NOT NULL,P_ID INTEGER NOT 105
NULL,EJN_ID_AUSWERTUNG INTEGER NOT NULL,WERT VARCHAR(20) NOT NULL,BEZEICHNUNG 106
VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,RANG INTEGER NOT NULL,BEMERKUNG 107
VARCHAR(2000),PWKT_ID INTEGER NOT NULL) ; 108
CREATE TABLE ICD_THESAURUS (ICDTHES_ID INTEGER NOT NULL,ICD_ID INTEGER NOT 109
NULL,ALTERNATIVTEXT VARCHAR(255) NOT NULL) ; 110
CREATE TABLE ELEMENT (E_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL 111
VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ; 112
CREATE TABLE KONTEXT_DIAGNOSE_PHASE (KDPH_ID INTEGER NOT NULL,PH_ID INTEGER NOT 113
NULL,KD_ID INTEGER NOT NULL) ; 114
CREATE TABLE PRUEFUNG_KOPPELUNG_ELEMENTE (PRFKE_ID INTEGER NOT NULL,HINWEISTEXT 115
VARCHAR(2000),BEMERKUNG VARCHAR(2000),PRFK_ID INTEGER NOT NULL,E_ID INTEGER NOT NULL) ; 116
CREATE TABLE KONTEXT_WERT (KW_ID INTEGER NOT NULL,K_ID INTEGER NOT NULL,W_ID INTEGER NOT 117
NULL,RANG INTEGER NOT NULL,EJN_ID_PFLICHTANGABE INTEGER NOT NULL,STANDARDWERT 118
VARCHAR(200)) ; 119
CREATE TABLE KOMPLEX_WERT_D_WERT_KOMB (KPWDWK_ID INTEGER NOT NULL,KPW_ID INTEGER NOT 120
NULL,DWK_ID INTEGER NOT NULL) ; 121
CREATE TABLE FACHABTEILUNG (FA_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT 122
NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ; 123
CREATE TABLE KOMPLEX_WERT (KPW_ID INTEGER NOT NULL,KP_ID INTEGER NOT 124
NULL,EJN_ID_AUSWERTUNG INTEGER NOT NULL,WERT VARCHAR(20) NOT NULL,BEZEICHNUNG 125
VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,RANG INTEGER NOT NULL,BEMERKUNG 126
VARCHAR(2000),KWT_ID INTEGER NOT NULL) ; 127
CREATE TABLE PROZEDUR (P_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL 128
VARCHAR(20) NOT NULL,NORMAL_WERT VARCHAR(20) NOT NULL,NORMAL_TEXT VARCHAR(200) NOT 129

```

```

130 NULL,BEMERKUNG VARCHAR(2000)) ;
131 CREATE TABLE ICD (ICD_ID INTEGER NOT NULL,ICDVER_ID INTEGER NOT NULL,CODE VARCHAR(10)
132 NOT NULL,KURZTEXT VARCHAR(2000) NOT NULL,VOLLTEXT VARCHAR(2000) NOT
133 NULL,INTERNE_BEZEICHNUNG VARCHAR(200) NOT NULL) ;
134 CREATE TABLE KOMPLEX_WERT_P_WERT_KOMB (KPWPWK_ID INTEGER NOT NULL,KPW_ID INTEGER NOT
135 NULL,PWK_ID INTEGER NOT NULL) ;
136 CREATE TABLE DIAGNOSE_WERT_ICD_LOK (DWIL_ID INTEGER NOT NULL,DWI_ID INTEGER NOT
137 NULL,ICDLOK_ID INTEGER NOT NULL) ;
138 CREATE TABLE ENTSCHEIDUNG_JA_NEIN (EJN_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
139 NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
140 CREATE TABLE ELEMENT_KOMPLEX (EK_ID INTEGER NOT NULL,KP_ID INTEGER NOT NULL,E_ID INTEGER
141 NOT NULL) ;
142 CREATE TABLE FALL_ICD (FALLICD_ID INTEGER NOT NULL,FALL_ID INTEGER NOT NULL,ICD_ID
143 INTEGER NOT NULL,ICDKAT_ID INTEGER NOT NULL,ICDLOK_ID INTEGER NOT NULL,DATUM DATE NOT
144 NULL,BEMERKUNG VARCHAR(2000)) ;
145 CREATE TABLE PRUEFUNG_KOPPELUNG (PRFK_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
146 NULL,KUERZEL VARCHAR(20) NOT NULL,HINWEISTEXT VARCHAR(2000) NOT NULL,BEMERKUNG
147 VARCHAR(2000),PRF_ID INTEGER NOT NULL) ;
148 CREATE TABLE PROZEDUR_WERT_OPS_LOK (PWOL_ID INTEGER NOT NULL,PWO_ID INTEGER NOT
149 NULL,OPSLOK_ID INTEGER NOT NULL) ;
150 CREATE TABLE KOMPLEX_WERT_KOMBINATIONSTYP (KWT_ID INTEGER NOT NULL,BEZEICHNUNG
151 VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
152 CREATE TABLE DIAGNOSE_WERT (DW_ID INTEGER NOT NULL,D_ID INTEGER NOT
153 NULL,EJN_ID_AUSWERTUNG INTEGER NOT NULL,WERT VARCHAR(20) NOT NULL,BEZEICHNUNG
154 VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,RANG INTEGER,BEMERKUNG
155 VARCHAR(2000)) ;
156 CREATE TABLE PROZEDUR_WERT_OPS (PWO_ID INTEGER NOT NULL,PW_ID INTEGER NOT NULL,OPS_ID
157 INTEGER NOT NULL) ;
158 CREATE TABLE OPS301_SUCHTEXT (OPSST_ID INTEGER NOT NULL,OPS_ID INTEGER NOT NULL,SUCHTEXT
159 VARCHAR(200) NOT NULL) ;
160 CREATE TABLE PHASE (PH_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL
161 VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
162 CREATE TABLE KONTEXT_KOMPLEX (KKP_ID INTEGER NOT NULL,K_ID INTEGER NOT NULL,KP_ID
163 INTEGER NOT NULL,RANG INTEGER NOT NULL) ;
164 CREATE TABLE PRUEFUNG (PRF_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL
165 VARCHAR(20) NOT NULL,HINWEISTEXT VARCHAR(2000) NOT NULL,BEMERKUNG VARCHAR(2000),PRFT_ID
166 INTEGER NOT NULL,REL_ID INTEGER NOT NULL) ;
167 CREATE TABLE SUPERKONTEXT (SK_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
168 NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
169 CREATE TABLE ELEMENT_DIAGNOSE_WERT_KOMB (EDWK_ID INTEGER NOT NULL,DWK_ID INTEGER NOT
170 NULL,E_ID INTEGER NOT NULL) ;
171 CREATE TABLE KONTEXT_WERT_WERT (KWW_ID INTEGER NOT NULL,K_ID INTEGER NOT NULL,WW_ID
172 INTEGER NOT NULL,RANG INTEGER) ;
173 CREATE TABLE BEARBEITER (BA_ID INTEGER NOT NULL,NAME VARCHAR(200) NOT NULL,VORNAME
174 VARCHAR(200) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
175 CREATE TABLE KONTEXT_PROZEDUR_PHASE (KPPH_ID INTEGER NOT NULL,PH_ID INTEGER NOT
176 NULL,KP_ID INTEGER NOT NULL) ;
177 CREATE TABLE OPS301_KATEGORIE (OPSKAT_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
178 NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
179 CREATE TABLE PRUEFUNGSTYP (PRFT_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
180 NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
181 CREATE TABLE OPS301_LOKALISATION (OPSLOK_ID INTEGER NOT NULL,CODE VARCHAR(20) NOT
182 NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG
183 VARCHAR(2000)) ;
184 CREATE TABLE KOMPLEX_WERT_P_WERT (KPWPWK_ID INTEGER NOT NULL,KPW_ID INTEGER NOT
185 NULL,PW_ID INTEGER NOT NULL) ;
186 CREATE TABLE PROZ_WERT_KOMB_PROZ_WERT (PWKPWK_ID INTEGER NOT NULL,PWK_ID INTEGER NOT
187 NULL,PW_ID INTEGER NOT NULL) ;
188 CREATE TABLE KONTEXT_PROZEDUR (KP_ID INTEGER NOT NULL,K_ID INTEGER NOT NULL,P_ID INTEGER
189 NOT NULL,RANG INTEGER NOT NULL) ;
190 CREATE TABLE KONTEXT_DIAG_WERT_KOMB (KDWK_ID INTEGER NOT NULL,K_ID INTEGER NOT
191 NULL,DWK_ID INTEGER NOT NULL,RANG INTEGER) ;
192 CREATE TABLE SUPERKONTEXT_KONTEXT (SKK_ID INTEGER NOT NULL,SK_ID INTEGER NOT NULL,K_ID
193 INTEGER NOT NULL,RANG INTEGER NOT NULL) ;
194 CREATE TABLE PRUEFUNG_AUSSCHLUSS_ELEMENTE (PRFAE_ID INTEGER NOT NULL,HINWEISTEXT
195 VARCHAR(2000) NOT NULL,BEMERKUNG VARCHAR(2000),E_ID INTEGER NOT NULL,PRFA_ID INTEGER NOT
196 NULL) ;
197 CREATE TABLE ELEMENT_PROZEDUR_WERT_KOMB (EPWK_ID INTEGER NOT NULL,PWK_ID INTEGER NOT

```

```

NULL,E_ID INTEGER NOT NULL) ;
CREATE TABLE PHASE_ICD_KATEGORIE (PHIKAT_ID INTEGER NOT NULL,PH_ID INTEGER NOT
NULL,ICDKAT_ID INTEGER NOT NULL,PHZT_ID INTEGER NOT NULL) ;
CREATE TABLE FALL_OP_WERT (FOPW_ID INTEGER NOT NULL,WERT VARCHAR(20) NOT NULL,W_ID
INTEGER NOT NULL,FALLOP_ID INTEGER NOT NULL) ;
CREATE TABLE PROZEDUR_OPS301 (PO_ID INTEGER NOT NULL,P_ID INTEGER NOT NULL,OPS_ID
INTEGER NOT NULL) ;
CREATE TABLE IMPORTDATEI (IMPDAT_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
NULL,KUERZEL VARCHAR(20) NOT NULL,DATEINAME VARCHAR(256) NOT NULL,DATEINAMENERWEITERUNG
VARCHAR(256) NOT NULL,TRENNER VARCHAR(200) NOT NULL,BEMERKUNG VARCHAR(2000),KIS_ID
INTEGER NOT NULL) ;
CREATE TABLE ICD_KATEGORIE (ICDKAT_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
CREATE TABLE DIAGNOSE_WERT_KOMBINATION (DWK_ID INTEGER NOT NULL,D_ID INTEGER NOT
NULL,EJN_ID AUSWERTUNG INTEGER NOT NULL,WERT VARCHAR(20) NOT NULL,BEZEICHNUNG
VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,RANG INTEGER,BEMERKUNG
VARCHAR(2000),DWKT_ID INTEGER NOT NULL) ;
CREATE TABLE WERT_PROZEDUR (WP_ID INTEGER NOT NULL,W_ID INTEGER NOT NULL,P_ID INTEGER
NOT NULL) ;
CREATE TABLE ELEMENT_DIAGNOSE (ED_ID INTEGER NOT NULL,D_ID INTEGER NOT NULL,E_ID INTEGER
NOT NULL) ;
CREATE TABLE ELEMENT_WERT_BEREICH (EWB_ID INTEGER NOT NULL,WB_ID INTEGER NOT NULL,E_ID
INTEGER NOT NULL) ;
CREATE TABLE KONTEXT_PRUEFUNG (KPRF_ID INTEGER NOT NULL,RANG INTEGER NOT NULL,PRF_ID
INTEGER NOT NULL,K_ID INTEGER NOT NULL) ;
CREATE TABLE PHASE_OPS301_KATEGORIE (PHOKAT_ID INTEGER NOT NULL,PH_ID INTEGER NOT
NULL,OPSKAT_ID INTEGER NOT NULL,PHZT_ID INTEGER NOT NULL) ;
CREATE TABLE ICD_VERSION (ICDVER_ID INTEGER NOT NULL,CODE VARCHAR(20) NOT
NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,GUELTIG_AB DATE NOT
NULL,BEMERKUNG VARCHAR(2000)) ;
CREATE TABLE DIAG_WERT_KOMB_DIAG_WERT (DWKDW_ID INTEGER NOT NULL,DWK_ID INTEGER NOT
NULL,DW_ID INTEGER NOT NULL) ;
CREATE TABLE FALL (FALL_ID INTEGER NOT NULL,CODE VARCHAR(200) NOT NULL,NACHNAME
VARCHAR(200) NOT NULL,VORNAME VARCHAR(200) NOT NULL,GEBURTSDATUM DATE NOT
NULL,AUFNAHME_FA DATE NOT NULL,ABGANG_FA DATE NOT NULL,AUFNAHME_KH DATE NOT
NULL,ABGANG_KH DATE NOT NULL,G_ID INTEGER NOT NULL,BA_ID INTEGER NOT NULL,KIS_ID INTEGER
NOT NULL,FA_ID INTEGER NOT NULL,ZULETZT_AKTUALISIERT DATE,EJN_ID_PUFFER_BEREIT INTEGER
NOT NULL) ;
CREATE TABLE FALL_OP (FALLOP_ID INTEGER NOT NULL,FALL_ID INTEGER NOT NULL,CODE
VARCHAR(240),BEGINN DATE NOT NULL,ENDE DATE NOT NULL,BEMERKUNG VARCHAR(240)) ;
CREATE TABLE KONTEXT_PROZEDUR_WERT (KPW_ID INTEGER NOT NULL,K_ID INTEGER NOT NULL,PW_ID
INTEGER NOT NULL,RANG INTEGER) ;
CREATE TABLE PROZEDUR_OPS301_LOK (POL_ID INTEGER NOT NULL,PO_ID INTEGER NOT
NULL,OPSLOK_ID INTEGER NOT NULL) ;
CREATE TABLE FALL_OP_ICD (FOPICD_ID INTEGER NOT NULL,FALLOP_ID INTEGER NOT NULL,ICD_ID
INTEGER NOT NULL,ICDKAT_ID INTEGER NOT NULL,ICDLOK_ID INTEGER NOT NULL) ;
CREATE TABLE PRUEFUNG_ELEMENT (PRFE_ID INTEGER NOT NULL,HINWEISTEXT VARCHAR(2000),E_ID
INTEGER NOT NULL,PRF_ID INTEGER NOT NULL) ;
CREATE TABLE DIAGNOSE_ICD_LOK (DIL_ID INTEGER NOT NULL,DI_ID INTEGER NOT NULL,ICDLOK_ID
INTEGER NOT NULL) ;
CREATE TABLE ICD_LOKALISATION (ICDLOK_ID INTEGER NOT NULL,CODE VARCHAR(20) NOT
NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG
VARCHAR(2000)) ;
CREATE TABLE KONTEXT_WERT_BEREICH (KWB_ID INTEGER NOT NULL,K_ID INTEGER NOT NULL,WB_ID
INTEGER NOT NULL,RANG INTEGER) ;
CREATE TABLE KOMPLEX (KP_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL
VARCHAR(20) NOT NULL,NORMAL_WERT VARCHAR(20) NOT NULL,NORMAL_TEXT VARCHAR(200) NOT
NULL,BEMERKUNG VARCHAR(2000)) ;
CREATE TABLE KONTEXT_ELEMENT_BEZIEHUNG (KEB_ID INTEGER NOT NULL,K_ID INTEGER NOT
NULL,EBEZ_ID INTEGER NOT NULL,BEMERKUNG VARCHAR(2000),MELDUNG VARCHAR(2000)) ;
CREATE TABLE ICD_SUCHTEXT (ICDST_ID INTEGER NOT NULL,ICD_ID INTEGER NOT NULL,SUCHTEXT
VARCHAR(255) NOT NULL) ;
CREATE TABLE DATENTYP (DATTYP_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT
NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
CREATE TABLE ELEMENT_BEZIEHUNGSTYP (EBEZTYP_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200)
NOT NULL,KUERZEL VARCHAR(20) NOT NULL,BEMERKUNG VARCHAR(2000)) ;
CREATE TABLE DIAGNOSE (D_ID INTEGER NOT NULL,BEZEICHNUNG VARCHAR(200) NOT NULL,KUERZEL
VARCHAR(20) NOT NULL,NORMAL_WERT VARCHAR(20) NOT NULL,NORMAL_TEXT VARCHAR(200) NOT

```

```

266 NULL,BEMERKUNG VARCHAR(2000)) ;
267 CREATE TABLE KONTEXT (K_ID INTEGER NOT NULL, BEZEICHNUNG VARCHAR(200) NOT NULL, KUERZEL
268 VARCHAR(20) NOT NULL, BEMERKUNG VARCHAR(2000)) ;
269 CREATE TABLE KONTEXT_DIAGNOSE_WERT (KDW_ID INTEGER NOT NULL, K_ID INTEGER NOT NULL, DW_ID
270 INTEGER NOT NULL, RANG INTEGER) ;
271 CREATE TABLE ELEMENT_KOMPLEX_WERT (EKW_ID INTEGER NOT NULL, KPW_ID INTEGER NOT NULL, E_ID
272 INTEGER NOT NULL) ;
273 CREATE TABLE WERT_KOMPLEX (WK_ID INTEGER NOT NULL, W_ID INTEGER NOT NULL, KP_ID INTEGER
274 NOT NULL) ;
275 CREATE TABLE KONTEXT_PROZ_WERT_KOMB (KPWK_ID INTEGER NOT NULL, K_ID INTEGER NOT
276 NULL, PWK_ID INTEGER NOT NULL, RANG INTEGER) ;
277
278 ALTER TABLE PRUEFUNG_AUSSCHLUSS ADD CONSTRAINT PRUEF_AUSS_PK PRIMARY KEY(PRFA_ID);
279 ALTER TABLE KONTEXT_KOMPLEX_WERT ADD CONSTRAINT K_K_WERT_PK PRIMARY KEY(KKW_ID);
280 ALTER TABLE OPS301_VERSION ADD CONSTRAINT OPS_VERSION_PK PRIMARY KEY(OPSVER_ID);
281 ALTER TABLE KIS ADD CONSTRAINT KIS_PK PRIMARY KEY(KIS_ID);
282 ALTER TABLE ELEMENT_WERT ADD CONSTRAINT E_WERT_PK PRIMARY KEY(EW_ID);
283 ALTER TABLE STATUS ADD CONSTRAINT STATUS_PK PRIMARY KEY(STATUS);
284 ALTER TABLE GESCHLECHT ADD CONSTRAINT GESCHLECHT_PK PRIMARY KEY(G_ID);
285 ALTER TABLE DIAGNOSE_WERT_ICD ADD CONSTRAINT DIAGNOSE_WERT_ICD_PK PRIMARY KEY(DWI_ID);
286 ALTER TABLE WERT_WERT ADD CONSTRAINT WERT_WERT_PK PRIMARY KEY(WW_ID);
287 ALTER TABLE KONTEXT_KOMPLEX_PHASE ADD CONSTRAINT KT_KOMP_PH_PK PRIMARY KEY(KKPPH_ID);
288 ALTER TABLE WERT ADD CONSTRAINT WERT_PK PRIMARY KEY(W_ID);
289 ALTER TABLE ELEMENT_PROZEDUR_WERT ADD CONSTRAINT E_P_WERT_PK PRIMARY KEY(EPW_ID);
290 ALTER TABLE DIAGNOSE_WERT_KOMBINATIONSTYP ADD CONSTRAINT DW_KOMB_TYP_PK PRIMARY
291 KEY(DWKT_ID);
292 ALTER TABLE WERT_BEREICH ADD CONSTRAINT WERT_BER_PK PRIMARY KEY(WB_ID);
293 ALTER TABLE KONTEXT_DIAGNOSE ADD CONSTRAINT KONTEXT_DIAGNOSE_PK PRIMARY KEY(KD_ID);
294 ALTER TABLE FALL_KONTEXT_PRUEFUNG_HINWEIS ADD CONSTRAINT F_K_PR_F_HW_PK PRIMARY
295 KEY(FKPHW_ID);
296 ALTER TABLE IMPORTDATEI_FELD ADD CONSTRAINT IDAT_FELD_PK PRIMARY KEY(IMPDATEF_ID);
297 ALTER TABLE PHASE_ZUORDNUNGSTYP ADD CONSTRAINT PH_ZUOTYP_PK PRIMARY KEY(PH_ZT_ID);
298 ALTER TABLE OPS301_THESAURUS ADD CONSTRAINT OPS301_THESAURUS_PK PRIMARY KEY(OPSTHES_ID);
299 ALTER TABLE FALL_OPS301 ADD CONSTRAINT FALL_OPS_PK PRIMARY KEY(FALLOPS_ID);
300 ALTER TABLE ELEMENT_PROZEDUR ADD CONSTRAINT E_PROZEDUR_PK PRIMARY KEY(EP_ID);
301 ALTER TABLE RELEVANZ ADD CONSTRAINT RELEVANZ_PK PRIMARY KEY(REL_ID);
302 ALTER TABLE IMPORTDATEI_FELD_WERT ADD CONSTRAINT IDATF_WERT_PK PRIMARY KEY(IMPDATEFW_ID);
303 ALTER TABLE PUFFER_FALL_KONTEXT ADD CONSTRAINT PUF_F_KT_PK PRIMARY KEY(ID);
304 ALTER TABLE PUFFER_FALL_OP_KONTEXT ADD CONSTRAINT PUF_FOP_KT_PK PRIMARY KEY(ID);
305 ALTER TABLE FALL_KONTEXT_PRUEFUNG ADD CONSTRAINT FALL_K_PR_F_PK PRIMARY KEY(FKP_ID);
306 ALTER TABLE PROZEDUR_WERT_KOMBINATIONSTYP ADD CONSTRAINT PW_KOMB_TYP_PK PRIMARY
307 KEY(PWKT_ID);
308 ALTER TABLE DIAGNOSE_ICD ADD CONSTRAINT DIAGNOSE_ICD_PK PRIMARY KEY(DI_ID);
309 ALTER TABLE FALL_OP_OPS301 ADD CONSTRAINT FALLOP_OPS_PK PRIMARY KEY(FOPOPS_ID);
310 ALTER TABLE ELEMENT_BEZIEHUNG ADD CONSTRAINT ELEM_BEZ_PK PRIMARY KEY(EBEZ_ID);
311 ALTER TABLE WERT_DIAGNOSE ADD CONSTRAINT WERT_DIAG_PK PRIMARY KEY(WD_ID);
312 ALTER TABLE ELEMENT_WERT_WERT ADD CONSTRAINT E_W_WERT_PK PRIMARY KEY(EWW_ID);
313 ALTER TABLE OPS301_ADD CONSTRAINT OPS301_PK PRIMARY KEY(OPS_ID);
314 ALTER TABLE KOMPLEX_WERT_D_WERT ADD CONSTRAINT KOMPLEX_WERT_D_WERT_PK PRIMARY
315 KEY(KPW_DW_ID);
316 ALTER TABLE ELEMENT_DIAGNOSE_WERT ADD CONSTRAINT E_D_WERT_PK PRIMARY KEY(EDW_ID);
317 ALTER TABLE FALL_WERT ADD CONSTRAINT FALL_WERT_PK PRIMARY KEY(FW_ID);
318 ALTER TABLE PROZEDUR_WERT ADD CONSTRAINT PROZEDUR_WERT_PK PRIMARY KEY(PW_ID);
319 ALTER TABLE PROZEDUR_WERT_KOMBINATION ADD CONSTRAINT PROZEDUR_WERT_KOMBINATION_PK PRIMARY
320 KEY(PWK_ID);
321 ALTER TABLE ICD_THESAURUS ADD CONSTRAINT ICD_THESAURUS_PK PRIMARY KEY(ICDTHES_ID);
322 ALTER TABLE ELEMENT ADD CONSTRAINT ELEMENT_PK PRIMARY KEY(E_ID);
323 ALTER TABLE KONTEXT_DIAGNOSE_PHASE ADD CONSTRAINT KT_DIAG_PH_PK PRIMARY KEY(KDPH_ID);
324 ALTER TABLE PRUEFUNG_KOPPELUNG_ELEMENTE ADD CONSTRAINT PRF_KOPP_E_PK PRIMARY
325 KEY(PRFKE_ID);
326 ALTER TABLE KONTEXT_WERT ADD CONSTRAINT KONT_WERT_PK PRIMARY KEY(KW_ID);
327 ALTER TABLE KOMPLEX_WERT_D_WERT_KOMB ADD CONSTRAINT KOMPLEX_WERT_D_WERT_KOMB_PK PRIMARY
328 KEY(KPW_DWK_ID);
329 ALTER TABLE FACHABTEILUNG ADD CONSTRAINT FACHABT_PK PRIMARY KEY(FA_ID);
330 ALTER TABLE KOMPLEX_WERT ADD CONSTRAINT KOMPLEX_WERT_PK PRIMARY KEY(KPW_ID);
331 ALTER TABLE PROZEDUR ADD CONSTRAINT PROZEDUR_PK PRIMARY KEY(P_ID);
332 ALTER TABLE ICD ADD CONSTRAINT ICD_PK PRIMARY KEY(ICD_ID);
333 ALTER TABLE KOMPLEX_WERT_P_WERT_KOMB ADD CONSTRAINT KOMPLEX_WERT_P_WERT_KOMB_PK PRIMARY

```

KEY(KPWPWK\_ID); 334  
 ALTER TABLE DIAGNOSE\_WERT\_ICD\_LOK ADD CONSTRAINT D\_WERT\_ICD\_LOK\_PK PRIMARY KEY(DWIL\_ID); 335  
 ALTER TABLE ENTSCHEIDUNG\_JA\_NEIN ADD CONSTRAINT ENTSCH\_JA\_NEIN\_PK PRIMARY KEY(EJN\_ID); 336  
 ALTER TABLE ELEMENT\_KOMPLEX ADD CONSTRAINT E\_KOMPLEX\_PK PRIMARY KEY(EK\_ID); 337  
 ALTER TABLE FALL\_ICD ADD CONSTRAINT FALL\_ICD\_PK PRIMARY KEY(FALLICD\_ID); 338  
 ALTER TABLE PRUEFUNG\_KOPPELUNG ADD CONSTRAINT PRUEF\_KOPP\_PK PRIMARY KEY(PRFK\_ID); 339  
 ALTER TABLE PROZEDUR\_WERT\_OPS\_LOK ADD CONSTRAINT P\_WERT\_OPS\_LOK\_PK PRIMARY KEY(PWOL\_ID); 340  
 ALTER TABLE KOMPLEX\_WERT\_KOMBINATIONSTYP ADD CONSTRAINT KW\_KOMB\_TYP\_PK PRIMARY 341  
 KEY(KWT\_ID); 342  
 ALTER TABLE DIAGNOSE\_WERT ADD CONSTRAINT DIAGNOSE\_WERT\_PK PRIMARY KEY(DW\_ID); 343  
 ALTER TABLE PROZEDUR\_WERT\_OPS ADD CONSTRAINT PROZEDUR\_WERT\_OPS\_PK PRIMARY KEY(PWO\_ID); 344  
 ALTER TABLE OPS301\_SUCHTEXT ADD CONSTRAINT OPS301\_SUCHTEXT\_PK PRIMARY KEY(OPSST\_ID); 345  
 ALTER TABLE PHASE ADD CONSTRAINT PHASE\_PK PRIMARY KEY(PH\_ID); 346  
 ALTER TABLE KONTEXT\_KOMPLEX ADD CONSTRAINT KONT\_KOMP\_PK PRIMARY KEY(KKP\_ID); 347  
 ALTER TABLE PRUEFUNG ADD CONSTRAINT PRUEFUNG\_PK PRIMARY KEY(PRF\_ID); 348  
 ALTER TABLE SUPERKONTEXT ADD CONSTRAINT SUPERKON\_PK PRIMARY KEY(SK\_ID); 349  
 ALTER TABLE ELEMENT\_DIAGNOSE\_WERT\_KOMB ADD CONSTRAINT E\_D\_W\_KOMB\_PK PRIMARY KEY(EDWK\_ID); 350  
 ALTER TABLE KONTEXT\_WERT\_WERT ADD CONSTRAINT K\_W\_WERT\_PK PRIMARY KEY(KWW\_ID); 351  
 ALTER TABLE BEARBEITER ADD CONSTRAINT BEARBEITER\_PK PRIMARY KEY(BA\_ID); 352  
 ALTER TABLE KONTEXT\_PROZEDUR\_PHASE ADD CONSTRAINT KT\_PROZ\_PH\_PK PRIMARY KEY(KPPH\_ID); 353  
 ALTER TABLE OPS301\_KATEGORIE ADD CONSTRAINT OPS\_KAT\_PK PRIMARY KEY(OPSKAT\_ID); 354  
 ALTER TABLE PRUEFUNGSTYP ADD CONSTRAINT PRUEFTYP\_PK PRIMARY KEY(PRFT\_ID); 355  
 ALTER TABLE OPS301\_LOKALISATION ADD CONSTRAINT OPS\_LOKALISATION\_PK PRIMARY 356  
 KEY(OPSLOK\_ID); 357  
 ALTER TABLE KOMPLEX\_WERT\_P\_WERT ADD CONSTRAINT KOMPLEX\_WERT\_P\_WERT\_PK PRIMARY 358  
 KEY(KPWPWK\_ID); 359  
 ALTER TABLE PROZ\_WERT\_KOMB\_PROZ\_WERT ADD CONSTRAINT PROZ\_WERT\_KOMB\_PROZ\_WERT\_PK PRIMARY 360  
 KEY(PWKPW\_ID); 361  
 ALTER TABLE KONTEXT\_PROZEDUR ADD CONSTRAINT KONTEXT\_PROZEDUR\_PK PRIMARY KEY(KP\_ID); 362  
 ALTER TABLE KONTEXT\_DIAG\_WERT\_KOMB ADD CONSTRAINT K\_D\_W\_KOMB\_PK PRIMARY KEY(KDWK\_ID); 363  
 ALTER TABLE SUPERKONTEXT\_KONTEXT ADD CONSTRAINT SUPKON\_KON\_PK PRIMARY KEY(SKK\_ID); 364  
 ALTER TABLE PRUEFUNG\_AUSSCHLUSS\_ELEMENTE ADD CONSTRAINT PRF\_AUSS\_E\_PK PRIMARY 365  
 KEY(PRFAE\_ID); 366  
 ALTER TABLE ELEMENT\_PROZEDUR\_WERT\_KOMB ADD CONSTRAINT E\_P\_W\_KOMB\_PK PRIMARY KEY(EPWK\_ID); 367  
 ALTER TABLE PHASE\_ICD\_KATEGORIE ADD CONSTRAINT PH\_ICD\_KAT\_PK PRIMARY KEY(PHIKAT\_ID); 368  
 ALTER TABLE FALL\_OP\_WERT ADD CONSTRAINT FOP\_WERT\_PK PRIMARY KEY(FOPW\_ID); 369  
 ALTER TABLE PROZEDUR\_OPS301 ADD CONSTRAINT PROZEDUR\_OPS301\_PK PRIMARY KEY(PO\_ID); 370  
 ALTER TABLE IMPORTDATEI ADD CONSTRAINT IMPDATEI\_PK PRIMARY KEY(IMPDAT\_ID); 371  
 ALTER TABLE ICD\_KATEGORIE ADD CONSTRAINT ICD\_KAT\_PK PRIMARY KEY(ICDKAT\_ID); 372  
 ALTER TABLE DIAGNOSE\_WERT\_KOMBINATION ADD CONSTRAINT DIAGNOSE\_WERT\_KOMBINATION\_PK PRIMARY 373  
 KEY(DWK\_ID); 374  
 ALTER TABLE WERT\_PROZEDUR ADD CONSTRAINT WERT\_PROZ\_PK PRIMARY KEY(WP\_ID); 375  
 ALTER TABLE ELEMENT\_DIAGNOSE ADD CONSTRAINT E\_DIAGNOSE\_PK PRIMARY KEY(ED\_ID); 376  
 ALTER TABLE ELEMENT\_WERT\_BEREICH ADD CONSTRAINT E\_W\_BER\_PK PRIMARY KEY(EWB\_ID); 377  
 ALTER TABLE KONTEXT\_PRUEFUNG ADD CONSTRAINT KONT\_PRUEF\_PK PRIMARY KEY(KPRF\_ID); 378  
 ALTER TABLE PHASE\_OPS301\_KATEGORIE ADD CONSTRAINT PH\_OPS\_KAT\_PK PRIMARY KEY(PHOKAT\_ID); 379  
 ALTER TABLE ICD\_VERSION ADD CONSTRAINT ICD\_VERSION\_PK PRIMARY KEY(ICDVER\_ID); 380  
 ALTER TABLE DIAG\_WERT\_KOMB\_DIAG\_WERT ADD CONSTRAINT DIAG\_WERT\_KOMB\_DIAG\_WERT\_PK PRIMARY 381  
 KEY(DWKDW\_ID); 382  
 ALTER TABLE FALL ADD CONSTRAINT FALL\_PK PRIMARY KEY(FALL\_ID); 383  
 ALTER TABLE FALL\_OP ADD CONSTRAINT FALL\_OP\_PK PRIMARY KEY(FALLOP\_ID); 384  
 ALTER TABLE KONTEXT\_PROZEDUR\_WERT ADD CONSTRAINT K\_P\_WERT\_PK PRIMARY KEY(KPW\_ID); 385  
 ALTER TABLE PROZEDUR\_OPS301\_LOK ADD CONSTRAINT PROZEDUR\_OPS301\_LOK\_PK PRIMARY 386  
 KEY(POL\_ID); 387  
 ALTER TABLE FALL\_OP\_ICD ADD CONSTRAINT FALLOP\_ICD\_PK PRIMARY KEY(FOPICD\_ID); 388  
 ALTER TABLE PRUEFUNG\_ELEMENT ADD CONSTRAINT PRUEF\_ELEM\_PK PRIMARY KEY(PRFE\_ID); 389  
 ALTER TABLE DIAGNOSE\_ICD\_LOK ADD CONSTRAINT D\_ICD\_LOK\_PK PRIMARY KEY(DIL\_ID); 390  
 ALTER TABLE ICD\_LOKALISATION ADD CONSTRAINT ICD\_LOKALISATION\_PK PRIMARY KEY(ICDLOK\_ID); 391  
 ALTER TABLE KONTEXT\_WERT\_BEREICH ADD CONSTRAINT K\_W\_BER\_PK PRIMARY KEY(KWB\_ID); 392  
 ALTER TABLE KOMPLEX ADD CONSTRAINT KOMPLEX\_PK PRIMARY KEY(KP\_ID); 393  
 ALTER TABLE KONTEXT\_ELEMENT\_BEZIEHUNG ADD CONSTRAINT KONT\_E\_BEZ\_PK PRIMARY KEY(KEB\_ID); 394  
 ALTER TABLE ICD\_SUCHTEXT ADD CONSTRAINT ICD\_SUCHTEXT\_PK PRIMARY KEY(ICDST\_ID); 395  
 ALTER TABLE DATENTYP ADD CONSTRAINT DATENTYP\_PK PRIMARY KEY(DATTYP\_ID); 396  
 ALTER TABLE ELEMENT\_BEZIEHUNGSTYP ADD CONSTRAINT E\_BEZ\_TYP\_PK PRIMARY KEY(EBEZTYP\_ID); 397  
 ALTER TABLE DIAGNOSE ADD CONSTRAINT DIAGNOSE\_PK PRIMARY KEY(D\_ID); 398  
 ALTER TABLE KONTEXT ADD CONSTRAINT KONTEXT\_PK PRIMARY KEY(K\_ID); 399  
 ALTER TABLE KONTEXT\_DIAGNOSE\_WERT ADD CONSTRAINT K\_D\_WERT\_PK PRIMARY KEY(KDW\_ID); 400  
 ALTER TABLE ELEMENT\_KOMPLEX\_WERT ADD CONSTRAINT E\_K\_WERT\_PK PRIMARY KEY(EKW\_ID); 401

```

402 ALTER TABLE WERT_KOMPLEX ADD CONSTRAINT WERT_KOMP_PK PRIMARY KEY(WK_ID);
403 ALTER TABLE KONTEXT_PROZ_WERT_KOMB ADD CONSTRAINT K_P_W_KOMB_PK PRIMARY KEY(KPW_ID);
404 ALTER TABLE PRUEFUNG_AUSSCHLUSS ADD CONSTRAINT PRUEF_AUSS_KUERZEL_UK UNIQUE(KUERZEL);
405 ALTER TABLE PRUEFUNG_AUSSCHLUSS ADD CONSTRAINT PRUEF_AUSS_BEZEICHNUNG_UK
406 UNIQUE(BEZEICHNUNG);
407 ALTER TABLE KONTEXT_KOMPLEX_WERT ADD CONSTRAINT K_K_WERT_K_ID_KPW_ID_UK
408 UNIQUE(KPW_ID,K_ID);
409 ALTER TABLE OPS301_VERSION ADD CONSTRAINT OPS301_VERSION_GUELTIG_AB_UK
410 UNIQUE(GUELTIG_AB);
411 ALTER TABLE OPS301_VERSION ADD CONSTRAINT OPS301_VERSION_CODE_UK UNIQUE(CODE);
412 ALTER TABLE OPS301_VERSION ADD CONSTRAINT OPS301_VERSION_BEZEICHNUNG_UK
413 UNIQUE(BEZEICHNUNG);
414 ALTER TABLE OPS301_VERSION ADD CONSTRAINT OPS301_VERSION_KUERZEL_UK UNIQUE(KUERZEL);
415 ALTER TABLE KIS ADD CONSTRAINT KIS_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
416 ALTER TABLE KIS ADD CONSTRAINT KIS_KUERZEL_UK UNIQUE(KUERZEL);
417 ALTER TABLE ELEMENT_WERT ADD CONSTRAINT E_WERT_E_ID_W_ID_UK UNIQUE(E_ID,W_ID);
418 ALTER TABLE STATUS ADD CONSTRAINT STATUS_KUERZEL_UK UNIQUE(KUERZEL);
419 ALTER TABLE STATUS ADD CONSTRAINT STATUS_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
420 ALTER TABLE GESCHLECHT ADD CONSTRAINT GESCHLECHT_KUERZEL_UK UNIQUE(KUERZEL);
421 ALTER TABLE GESCHLECHT ADD CONSTRAINT GESCHLECHT_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
422 ALTER TABLE DIAGNOSE_WERT_ICD ADD CONSTRAINT DIAG_WERT_ICD_DW_ID_ICD_ID_UK
423 UNIQUE(DW_ID,ICD_ID);
424 ALTER TABLE WERT_WERT ADD CONSTRAINT WERT_WERT_W_ID_KUERZEL_UK UNIQUE(KUERZEL,W_ID);
425 ALTER TABLE WERT_WERT ADD CONSTRAINT WERT_WERT_W_ID_WERT_UK UNIQUE(W_ID,WERT);
426 ALTER TABLE WERT_WERT ADD CONSTRAINT WERT_WERT_W_ID_BEZEICHNUNG_UK
427 UNIQUE(BEZEICHNUNG,W_ID);
428 ALTER TABLE KONTEXT_KOMPLEX_PHASE ADD CONSTRAINT KT_KOMP_PH_PH_ID_KKP_ID_UK
429 UNIQUE(KKP_ID,PH_ID);
430 ALTER TABLE WERT ADD CONSTRAINT WERT_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
431 ALTER TABLE WERT ADD CONSTRAINT WERT_KUERZEL_UK UNIQUE(KUERZEL);
432 ALTER TABLE ELEMENT_PROZEDUR_WERT ADD CONSTRAINT E_P_WERT_E_ID_P_ID_UK
433 UNIQUE(PW_ID,E_ID);
434 ALTER TABLE DIAGNOSE_WERT_KOMBINATIONSTYP ADD CONSTRAINT DW_KOMB_TYP_KUERZEL_UK
435 UNIQUE(KUERZEL);
436 ALTER TABLE DIAGNOSE_WERT_KOMBINATIONSTYP ADD CONSTRAINT DW_KOMB_TYP_BEZEICHNUNG_UK
437 UNIQUE(BEZEICHNUNG);
438 ALTER TABLE WERT_BEREICH ADD CONSTRAINT WERT_BER_W_MAX_UK UNIQUE(W_ID,WERT_MAXIMUM);
439 ALTER TABLE WERT_BEREICH ADD CONSTRAINT WERT_BER_W_MIN_UK UNIQUE(WERT_MINIMUM,W_ID);
440 ALTER TABLE KONTEXT_DIAGNOSE ADD CONSTRAINT KONTEXT_DIAGNOSE_K_ID_D_ID_UK
441 UNIQUE(K_ID,D_ID);
442 ALTER TABLE KONTEXT_DIAGNOSE ADD CONSTRAINT KONTEXT_DIAGNOSE_K_ID_RANG_UK
443 UNIQUE(RANG,K_ID);
444 ALTER TABLE FALL_KONTEXT_PRUEFUNG_HINWEIS ADD CONSTRAINT F_K_PR_F_HW_FKP_ID_HWT_ID_RF_UK
445 UNIQUE(FKP_ID,HWT_ID,REIHENFOLGE);
446 ALTER TABLE IMPORTDATEI_FELD ADD CONSTRAINT IDAT_FELD_IMP DAT_ID_RF_UK
447 UNIQUE(REIHENFOLGE,IMP DAT_ID);
448 ALTER TABLE IMPORTDATEI_FELD ADD CONSTRAINT IDAT_FELD_IMP DAT_ID_KUERZEL_UK
449 UNIQUE(KUERZEL,IMP DAT_ID);
450 ALTER TABLE IMPORTDATEI_FELD ADD CONSTRAINT IDAT_FELD_IMP DAT_ID_BEZ_UK
451 UNIQUE(BEZEICHNUNG,IMP DAT_ID);
452 ALTER TABLE OPS301_THESAURUS ADD CONSTRAINT OPS301_THES_OPS_ID_ALTTEXT_UK
453 UNIQUE(OPS_ID,ALTERNATIVTEXT);
454 ALTER TABLE ELEMENT_PROZEDUR ADD CONSTRAINT E_PROZEDUR_E_ID_P_ID_UK UNIQUE(E_ID,P_ID);
455 ALTER TABLE RELEVANZ ADD CONSTRAINT RELEVANZ_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
456 ALTER TABLE RELEVANZ ADD CONSTRAINT RELEVANZ_KUERZEL_UK UNIQUE(KUERZEL);
457 ALTER TABLE IMPORTDATEI_FELD_WERT ADD CONSTRAINT IDATF_WERT_IMP DATF_ID_UK
458 UNIQUE(IMP DATF_ID);
459 ALTER TABLE FALL_KONTEXT_PRUEFUNG ADD CONSTRAINT FALL_K_PR_F_FALL_ID_KPRF_ID_UK
460 UNIQUE(FALL_ID,KPRF_ID);
461 ALTER TABLE PROZEDUR_WERT_KOMBINATIONSTYP ADD CONSTRAINT PW_KOMB_TYP_BEZEICHNUNG_UK
462 UNIQUE(BEZEICHNUNG);
463 ALTER TABLE PROZEDUR_WERT_KOMBINATIONSTYP ADD CONSTRAINT PW_KOMB_TYP_KUERZEL_UK
464 UNIQUE(KUERZEL);
465 ALTER TABLE DIAGNOSE_ICD ADD CONSTRAINT DIAGNOSE_ICD_D_ID_ICD_ID_UK UNIQUE(D_ID,ICD_ID);
466 ALTER TABLE ELEMENT_BEZIEHUNG ADD CONSTRAINT ELEM_BEZ_EBEZ_TYP_ID_E_ID_12_UK
467 UNIQUE(E_ID_1,EBEZ_TYP_ID,E_ID_2);
468 ALTER TABLE WERT_DIAGNOSE ADD CONSTRAINT WERT_DIAG_W_ID_D_ID_UK UNIQUE(W_ID,D_ID);
469 ALTER TABLE ELEMENT_WERT_WERT ADD CONSTRAINT E_W_WERT_E_ID_WW_ID_UK UNIQUE(WW_ID,E_ID);

```



```

ALTER TABLE OPS301 ADD CONSTRAINT OPS301_OPSVER_ID_CODE_UK UNIQUE(OPSVER_ID, CODE); 470
ALTER TABLE KOMPLEX_WERT_D_WERT ADD CONSTRAINT KOMPL_W_D_W_KPW_ID_DW_ID_UK 471
UNIQUE(KPW_ID, DW_ID); 472
ALTER TABLE ELEMENT_DIAGNOSE_WERT ADD CONSTRAINT E_D_WERT_E_ID_DW_ID_UK 473
UNIQUE(E_ID, DW_ID); 474
ALTER TABLE FALL_WERT ADD CONSTRAINT FALL_WERT_FALL_ID_W_ID_UK UNIQUE(FALL_ID, W_ID); 475
ALTER TABLE PROZEDUR_WERT ADD CONSTRAINT PROZ_W_P_ID_BEZEICHNUNG_UK 476
UNIQUE(P_ID, BEZEICHNUNG); 477
ALTER TABLE PROZEDUR_WERT ADD CONSTRAINT PROZ_W_P_ID_WERT_UK UNIQUE(P_ID, WERT); 478
ALTER TABLE PROZEDUR_WERT ADD CONSTRAINT PROZ_W_P_ID_KUERZEL_UK UNIQUE(P_ID, KUERZEL); 479
ALTER TABLE PROZEDUR_WERT_KOMBINATION ADD CONSTRAINT P_W_K_P_ID_WERT_UK 480
UNIQUE(P_ID, WERT); 481
ALTER TABLE PROZEDUR_WERT_KOMBINATION ADD CONSTRAINT P_W_K_P_ID_KUERZEL_UK 482
UNIQUE(P_ID, KUERZEL); 483
ALTER TABLE PROZEDUR_WERT_KOMBINATION ADD CONSTRAINT P_W_K_P_ID_BEZEICHNUNG_UK 484
UNIQUE(P_ID, BEZEICHNUNG); 485
ALTER TABLE ICD_THESAURUS ADD CONSTRAINT ICD_THES_ICD_ID_ALTTEXT_UK 486
UNIQUE(ICD_ID, ALTERNATIVTEXT); 487
ALTER TABLE ELEMENT ADD CONSTRAINT ELEMENT_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 488
ALTER TABLE ELEMENT ADD CONSTRAINT ELEMENT_KUERZEL_UK UNIQUE(KUERZEL); 489
ALTER TABLE KONTEXT_DIAGNOSE_PHASE ADD CONSTRAINT KT_DIAG_PH_PH_ID_KD_ID_UK 490
UNIQUE(PH_ID, KD_ID); 491
ALTER TABLE PRUEFUNG_KOPPELUNG_ELEMENTE ADD CONSTRAINT PRF_KOPP_PRFK_ID_E_ID_UK 492
UNIQUE(E_ID, PRFK_ID); 493
ALTER TABLE KONTEXT_WERT ADD CONSTRAINT KONT_WERT_K_ID_RANG_UK UNIQUE(RANG, K_ID); 494
ALTER TABLE KONTEXT_WERT ADD CONSTRAINT KONT_WERT_K_ID_W_ID_UK UNIQUE(W_ID, K_ID); 495
ALTER TABLE KOMPLEX_WERT_D_WERT_KOMB ADD CONSTRAINT KOMPL_W_D_W_K_KPW_ID_DWK_ID_UK 496
UNIQUE(KPW_ID, DWK_ID); 497
ALTER TABLE FACHABTEILUNG ADD CONSTRAINT FACHABT_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 498
ALTER TABLE FACHABTEILUNG ADD CONSTRAINT FACHABT_KUERZEL_UK UNIQUE(KUERZEL); 499
ALTER TABLE KOMPLEX_WERT ADD CONSTRAINT KOMPLEX_W_KP_ID_KUERZEL_UK UNIQUE(KP_ID, KUERZEL); 500
ALTER TABLE KOMPLEX_WERT ADD CONSTRAINT KOMPLEX_W_KP_ID_WERT_UK UNIQUE(KP_ID, WERT); 501
ALTER TABLE KOMPLEX_WERT ADD CONSTRAINT KOMPLEX_W_KP_ID_BEZEICHNUNG_UK 502
UNIQUE(KP_ID, BEZEICHNUNG); 503
ALTER TABLE PROZEDUR ADD CONSTRAINT PROZEDUR_KUERZEL_UK UNIQUE(KUERZEL); 504
ALTER TABLE PROZEDUR ADD CONSTRAINT PROZEDUR_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 505
ALTER TABLE ICD ADD CONSTRAINT ICD_ICDVER_ID_CODE_UK UNIQUE(ICDVER_ID, CODE); 506
ALTER TABLE KOMPLEX_WERT_P_WERT_KOMB ADD CONSTRAINT KOMPL_W_P_W_K_KPW_ID_PWK_ID_UK 507
UNIQUE(KPW_ID, PWK_ID); 508
ALTER TABLE DIAGNOSE_WERT_ICD_LOK ADD CONSTRAINT D_W_ICD_L_DWI_ID_ICDLOK_ID_UK 509
UNIQUE(DWI_ID, ICDLOK_ID); 510
ALTER TABLE ENTSCHEIDUNG_JA_NEIN ADD CONSTRAINT ENTSCH_JA_NEIN_BEZEICHNUNG_UK 511
UNIQUE(BEZEICHNUNG); 512
ALTER TABLE ENTSCHEIDUNG_JA_NEIN ADD CONSTRAINT ENTSCH_JA_NEIN_KUERZEL_UK 513
UNIQUE(KUERZEL); 514
ALTER TABLE ELEMENT_KOMPLEX ADD CONSTRAINT E_KOMPLEX_E_ID_KP_ID_UK UNIQUE(KP_ID, E_ID); 515
ALTER TABLE PRUEFUNG_KOPPELUNG ADD CONSTRAINT PRUEF_KOPP_KUERZEL_UK UNIQUE(KUERZEL); 516
ALTER TABLE PRUEFUNG_KOPPELUNG ADD CONSTRAINT PRUEF_KOPP_BEZEICHNUNG_UK 517
UNIQUE(BEZEICHNUNG); 518
ALTER TABLE PROZEDUR_WERT_OPS_LOK ADD CONSTRAINT PROZ_W_O_L_PWO_ID_OPSLOK_ID_UK 519
UNIQUE(PWO_ID, OPSLOK_ID); 520
ALTER TABLE KOMPLEX_WERT_KOMBINATIONSTYP ADD CONSTRAINT KW_KOMBTYP_KUERZEL_UK 521
UNIQUE(KUERZEL); 522
ALTER TABLE KOMPLEX_WERT_KOMBINATIONSTYP ADD CONSTRAINT KW_KOMBTYP_BEZEICHNUNG_UK 523
UNIQUE(BEZEICHNUNG); 524
ALTER TABLE DIAGNOSE_WERT ADD CONSTRAINT DIAGNOSE_WERT_D_ID_KUERZEL_UK 525
UNIQUE(D_ID, KUERZEL); 526
ALTER TABLE DIAGNOSE_WERT ADD CONSTRAINT DIAGNOSE_WERT_D_ID_BEZ_UK 527
UNIQUE(D_ID, BEZEICHNUNG); 528
ALTER TABLE DIAGNOSE_WERT ADD CONSTRAINT DIAGNOSE_WERT_D_ID_WERT_UK UNIQUE(D_ID, WERT); 529
ALTER TABLE PROZEDUR_WERT_OPS ADD CONSTRAINT PROZ_W_OPS_PW_ID_OPS_ID_UK 530
UNIQUE(PW_ID, OPS_ID); 531
ALTER TABLE PHASE ADD CONSTRAINT PHASE_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 532
ALTER TABLE PHASE ADD CONSTRAINT PHASE_KUERZEL_UK UNIQUE(KUERZEL); 533
ALTER TABLE KONTEXT_KOMPLEX ADD CONSTRAINT KONTEXT_KOMPLEX_K_ID_RANG_UK 534
UNIQUE(RANG, K_ID); 535
ALTER TABLE KONTEXT_KOMPLEX ADD CONSTRAINT _KONTEXT_KOMPLEX_K_ID_KP_ID_UK 536
UNIQUE(K_ID, KP_ID); 537

```

```

538 ALTER TABLE PRUEFUNG ADD CONSTRAINT PRUEFUNG_KUERZEL_UK UNIQUE(KUERZEL);
539 ALTER TABLE PRUEFUNG ADD CONSTRAINT PRUEFUNG_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
540 ALTER TABLE SUPERKONTEXT ADD CONSTRAINT SUPERKON_KUERZEL_UK UNIQUE(KUERZEL);
541 ALTER TABLE SUPERKONTEXT ADD CONSTRAINT SUPERKON_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
542 ALTER TABLE ELEMENT_DIAGNOSE_WERT_KOMB ADD CONSTRAINT E_D_W_KOMB_E_ID_DWK_ID_UK
543 UNIQUE(E_ID,DWK_ID);
544 ALTER TABLE KONTEXT_WERT_WERT ADD CONSTRAINT K_W_WERT_K_ID_WW_ID_UK UNIQUE(WW_ID,K_ID);
545 ALTER TABLE KONTEXT_PROZEDUR_PHASE ADD CONSTRAINT KT_PROZ_PH_PH_ID_KP_ID_UK
546 UNIQUE(KP_ID,PH_ID);
547 ALTER TABLE OPS301_KATEGORIE ADD CONSTRAINT OPS_KAT_KUERZEL_UK UNIQUE(KUERZEL);
548 ALTER TABLE OPS301_KATEGORIE ADD CONSTRAINT OPS_KAT_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
549 ALTER TABLE PRUEFUNGSTYP ADD CONSTRAINT PRUEFTYP_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
550 ALTER TABLE PRUEFUNGSTYP ADD CONSTRAINT PRUEFTYP_KUERZEL_UK UNIQUE(KUERZEL);
551 ALTER TABLE OPS301_LOKALISATION ADD CONSTRAINT OPS301_LOK_KUERZEL_UK UNIQUE(KUERZEL);
552 ALTER TABLE OPS301_LOKALISATION ADD CONSTRAINT OPS301_LOK_BEZEICHNUNG_UK
553 UNIQUE(BEZEICHNUNG);
554 ALTER TABLE OPS301_LOKALISATION ADD CONSTRAINT OPS301_LOK_CODE_UK UNIQUE(CODE);
555 ALTER TABLE KOMPLEX_WERT_P_WERT ADD CONSTRAINT KOMPL_W_P_W_KPW_ID_PW_ID_UK
556 UNIQUE(KPW_ID,PW_ID);
557 ALTER TABLE PROZ_WERT_KOMB_PROZ_WERT ADD CONSTRAINT P_W_K_P_W_PWK_ID_PW_ID_UK
558 UNIQUE(PWK_ID,PW_ID);
559 ALTER TABLE KONTEXT_PROZEDUR ADD CONSTRAINT KONTEXT_PROZEDUR_K_ID_P_ID_UK
560 UNIQUE(K_ID,P_ID);
561 ALTER TABLE KONTEXT_PROZEDUR ADD CONSTRAINT KONTEXT_PROZEDUR_K_ID_RANG_UK
562 UNIQUE(RANG,K_ID);
563 ALTER TABLE KONTEXT_DIAG_WERT_KOMB ADD CONSTRAINT K_D_W_KOMB_K_ID_DWK_ID_UK
564 UNIQUE(DWK_ID,K_ID);
565 ALTER TABLE SUPERKONTEXT_KONTEXT ADD CONSTRAINT SUPKON_KON_SK_ID_RANG_UK
566 UNIQUE(RANG,SK_ID);
567 ALTER TABLE SUPERKONTEXT_KONTEXT ADD CONSTRAINT SUPKON_KON_SK_ID_K_ID_UK
568 UNIQUE(K_ID,SK_ID);
569 ALTER TABLE PRUEFUNG_AUSSCHLUSS_ELEMENTE ADD CONSTRAINT PRF_AUSS_PRFA_ID_E_ID_UK
570 UNIQUE(E_ID,PRFA_ID);
571 ALTER TABLE ELEMENT_PROZEDUR_WERT_KOMB ADD CONSTRAINT E_P_W_KOMB_E_ID_PWK_ID_UK
572 UNIQUE(PWK_ID,E_ID);
573 ALTER TABLE PHASE_ICD_KATEGORIE ADD CONSTRAINT PH_ICD_KAT_PH_ID_ICDKAT_ID_UK
574 UNIQUE(ICDKAT_ID,PH_ID);
575 ALTER TABLE FALL_OP_WERT ADD CONSTRAINT FOP_WERT_FALL_OP_ID_W_ID_UK
576 UNIQUE(W_ID,FALL_OP_ID);
577 ALTER TABLE PROZEDUR_OPS301 ADD CONSTRAINT PROZEDUR_OPS301_P_ID_OPS_ID_UK
578 UNIQUE(P_ID,OPS_ID);
579 ALTER TABLE IMPORTDATEI ADD CONSTRAINT IMPDATEI_KIS_ID_KUERZEL_UK UNIQUE(KIS_ID,KUERZEL);
580 ALTER TABLE IMPORTDATEI ADD CONSTRAINT IMPDATEI_KIS_ID_BEZEICHNUNG_UK
581 UNIQUE(BEZEICHNUNG,KIS_ID);
582 ALTER TABLE ICD_KATEGORIE ADD CONSTRAINT ICD_KAT_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
583 ALTER TABLE ICD_KATEGORIE ADD CONSTRAINT ICD_KAT_KUERZEL_UK UNIQUE(KUERZEL);
584 ALTER TABLE DIAGNOSE_WERT_KOMBINATION ADD CONSTRAINT DIAG_WERT_KOMB_D_ID_BEZ_UK
585 UNIQUE(D_ID,BEZEICHNUNG);
586 ALTER TABLE DIAGNOSE_WERT_KOMBINATION ADD CONSTRAINT DIAG_WERT_KOMB_D_ID_WERT_UK
587 UNIQUE(D_ID,WERT);
588 ALTER TABLE DIAGNOSE_WERT_KOMBINATION ADD CONSTRAINT DIAG_WERT_KOMB_D_ID_KUERZEL_UK
589 UNIQUE(D_ID,KUERZEL);
590 ALTER TABLE WERT_PROZEDUR ADD CONSTRAINT WERT_PROZ_W_ID_P_ID_UK UNIQUE(W_ID,P_ID);
591 ALTER TABLE ELEMENT_DIAGNOSE ADD CONSTRAINT E_DIAGNOSE_E_ID_D_ID_UK UNIQUE(E_ID,D_ID);
592 ALTER TABLE ELEMENT_WERT_BEREICH ADD CONSTRAINT E_W_BER_E_ID_WB_ID_UK UNIQUE(WB_ID,E_ID);
593 ALTER TABLE KONTEXT_PRUEFUNG ADD CONSTRAINT KONT_PRUEF_K_ID_PRF_ID_UK
594 UNIQUE(PRF_ID,K_ID);
595 ALTER TABLE KONTEXT_PRUEFUNG ADD CONSTRAINT KONT_PRUEF_K_ID_RANG_UK UNIQUE(RANG,K_ID);
596 ALTER TABLE PHASE_OPS301_KATEGORIE ADD CONSTRAINT PH_OPS_KAT_PH_ID_OPSKAT_ID_UK
597 UNIQUE(OPSKAT_ID,PH_ID);
598 ALTER TABLE ICD_VERSION ADD CONSTRAINT ICD_VERSION_KUERZEL_UK UNIQUE(KUERZEL);
599 ALTER TABLE ICD_VERSION ADD CONSTRAINT ICD_VERSION_GUELTIG_AB_UK UNIQUE(GUELTIG_AB);
600 ALTER TABLE ICD_VERSION ADD CONSTRAINT ICD_VERSION_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG);
601 ALTER TABLE ICD_VERSION ADD CONSTRAINT ICD_VERSION_CODE_UK UNIQUE(CODE);
602 ALTER TABLE DIAG_WERT_KOMB_DIAG_WERT ADD CONSTRAINT D_W_K_D_W_DWK_ID_DW_ID_UK
603 UNIQUE(DWK_ID,DW_ID);
604 ALTER TABLE FALL ADD CONSTRAINT FALL_KIS_ID_CODE_UK UNIQUE(CODE,KIS_ID);
605 ALTER TABLE KONTEXT_PROZEDUR_WERT ADD CONSTRAINT K_P_WERT_K_ID_PW_ID_UK

```

```

UNIQUE(PW_ID,K_ID); 606
ALTER TABLE PROZEDUR_OPS301_LOK ADD CONSTRAINT PROZ_OPS_L_PO_ID_OPSLOK_ID_UK 607
UNIQUE(PO_ID,OPSLOK_ID); 608
ALTER TABLE PRUEFUNG_ELEMENT ADD CONSTRAINT PRUEF_ELEM_PRF_ID_E_ID_UK 609
UNIQUE(E_ID,PRF_ID); 610
ALTER TABLE DIAGNOSE_ICD_LOK ADD CONSTRAINT D_ICD_LOK_DI_ID_ICDLOK_ID_UK 611
UNIQUE(DI_ID,ICDLOK_ID); 612
ALTER TABLE ICD_LOKALISATION ADD CONSTRAINT ICD_LOKALISATION_BEZEICH_UK 613
UNIQUE(BEZEICHNUNG); 614
ALTER TABLE ICD_LOKALISATION ADD CONSTRAINT ICD_LOKALISATION_CODE_UK UNIQUE(CODE); 615
ALTER TABLE ICD_LOKALISATION ADD CONSTRAINT ICD_LOKALISATION_KUERZEL_UK UNIQUE(KUERZEL); 616
ALTER TABLE KONTEXT_WERT_BEREICH ADD CONSTRAINT K_W_BER_K_ID_WB_ID_UK UNIQUE(K_ID,WB_ID); 617
ALTER TABLE KOMPLEX ADD CONSTRAINT KOMPLEX_KUERZEL_UK UNIQUE(KUERZEL); 618
ALTER TABLE KOMPLEX ADD CONSTRAINT KOMPLEX_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 619
ALTER TABLE KONTEXT_ELEMENT_BEZIEHUNG ADD CONSTRAINT KONT_E_BEZ_K_ID_EBEZ_ID_UK 620
UNIQUE(EBEZ_ID,K_ID); 621
ALTER TABLE DATENTYP ADD CONSTRAINT DATENTYP_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 622
ALTER TABLE DATENTYP ADD CONSTRAINT DATENTYP_KUERZEL_UK UNIQUE(KUERZEL); 623
ALTER TABLE ELEMENT_BEZIEHUNGSTYP ADD CONSTRAINT E_BEZTYP_KUERZEL_UK UNIQUE(KUERZEL); 624
ALTER TABLE ELEMENT_BEZIEHUNGSTYP ADD CONSTRAINT E_BEZTYP_BEZEICHNUNG_UK 625
UNIQUE(BEZEICHNUNG); 626
ALTER TABLE DIAGNOSE ADD CONSTRAINT DIAGNOSE_KUERZEL_UK UNIQUE(KUERZEL); 627
ALTER TABLE DIAGNOSE ADD CONSTRAINT DIAGNOSE_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 628
ALTER TABLE KONTEXT ADD CONSTRAINT KONTEXT_BEZEICHNUNG_UK UNIQUE(BEZEICHNUNG); 629
ALTER TABLE KONTEXT ADD CONSTRAINT KONTEXT_KUERZEL_UK UNIQUE(KUERZEL); 630
ALTER TABLE KONTEXT_DIAGNOSE_WERT ADD CONSTRAINT K_D_WERT_K_ID_DW_ID_UK 631
UNIQUE(K_ID,DW_ID); 632
ALTER TABLE ELEMENT_KOMPLEX_WERT ADD CONSTRAINT E_K_WERT_E_ID_KPW_ID_UK 633
UNIQUE(E_ID,KPW_ID); 634
ALTER TABLE WERT_KOMPLEX ADD CONSTRAINT WERT_KOMP_W_ID_KP_ID_UK UNIQUE(KP_ID,W_ID); 635
ALTER TABLE KONTEXT_PROZ_WERT_KOMB ADD CONSTRAINT K_P_W_KOMB_K_ID_PWK_ID_UK 636
UNIQUE(K_ID,PWK_ID); 637

ALTER TABLE WERT_BEREICH ADD CONSTRAINT WERT_BER_MIN_MAX_CH CHECK (WERT_MAXIMUM >= 638
WERT_MINIMUM) ; 639
ALTER TABLE ELEMENT_BEZIEHUNG ADD CONSTRAINT ELEM_BEZ_E_ID_1_E_ID_2_CH CHECK (E_ID_1 != 640
E_ID_2) ; 641
ALTER TABLE FALL ADD CONSTRAINT FALL_AUFNAHME_ABGANG_CH CHECK (Aufnahme_KH <= Aufnahme_FA 642
And 643
Aufnahme_FA <= Abgang_FA 644
And 645
Abgang_FA <= Abgang_KH) ; 646
ALTER TABLE FALL_OP ADD CONSTRAINT FALL_OP_BEGINN_ENDE_CH CHECK (Beginn <= Ende) ; 647
ALTER TABLE PRUEFUNG_AUSSCHLUSS ADD CONSTRAINT PRUEF_AUSS_PRUEFUNG_FK FOREIGN KEY 648
(PRF_ID)REFERENCES PRUEFUNG(PRF_ID); 649
ALTER TABLE KONTEXT_KOMPLEX_WERT ADD CONSTRAINT K_K_WERT_KOMP_WERT_FK FOREIGN KEY 650
(KPW_ID)REFERENCES KOMPLEX_WERT(KPW_ID); 651
ALTER TABLE KONTEXT_KOMPLEX_WERT ADD CONSTRAINT K_K_WERT_KONTEXT_FK FOREIGN KEY 652
(K_ID)REFERENCES KONTEXT(K_ID); 653
ALTER TABLE ELEMENT_WERT ADD CONSTRAINT E_WERT_ELEMENT_FK FOREIGN KEY (E_ID)REFERENCES 654
ELEMENT(E_ID); 655
ALTER TABLE ELEMENT_WERT ADD CONSTRAINT E_WERT_WERT_FK FOREIGN KEY (W_ID)REFERENCES 656
WERT(W_ID); 657
ALTER TABLE DIAGNOSE_WERT_ICD ADD CONSTRAINT DIAGNOSE_WERT_ICD_ICD_FK FOREIGN KEY 658
(ICD_ID)REFERENCES ICD(ICD_ID); 659
ALTER TABLE DIAGNOSE_WERT_ICD ADD CONSTRAINT DIAGNOSE_WERT_ICD_DW_FK FOREIGN KEY 660
(DW_ID)REFERENCES DIAGNOSE_WERT(DW_ID); 661
ALTER TABLE WERT_WERT ADD CONSTRAINT WERT_WERT_WERT_FK FOREIGN KEY (W_ID)REFERENCES 662
WERT(W_ID); 663
ALTER TABLE KONTEXT_KOMPLEX_PHASE ADD CONSTRAINT KT_KOMP_PH_KONT_KOMP_FK FOREIGN KEY 664
(KKP_ID)REFERENCES KONTEXT_KOMPLEX(KKP_ID); 665
ALTER TABLE KONTEXT_KOMPLEX_PHASE ADD CONSTRAINT KT_KOMP_PH_PHASE_FK FOREIGN KEY 666
(PH_ID)REFERENCES PHASE(PH_ID); 667
ALTER TABLE ELEMENT_PROZEDUR_WERT ADD CONSTRAINT E_P_WERT_ELEMENT_FK FOREIGN KEY 668
(E_ID)REFERENCES ELEMENT(E_ID); 669
ALTER TABLE ELEMENT_PROZEDUR_WERT ADD CONSTRAINT E_P_WERT_PROZ_WERT_FK FOREIGN KEY 670
(PW_ID)REFERENCES PROZEDUR_WERT(PW_ID); 671
ALTER TABLE WERT_BEREICH ADD CONSTRAINT WERT_BER_WERT_FK FOREIGN KEY (W_ID)REFERENCES 672
WERT(W_ID); 673

```

```

674 WERT(W_ID);
675 ALTER TABLE KONTEXT_DIAGNOSE ADD CONSTRAINT KONTEXT_DIAGNOSE_KONTEXT_FK FOREIGN KEY
676 (K_ID)REFERENCES KONTEXT(K_ID);
677 ALTER TABLE KONTEXT_DIAGNOSE ADD CONSTRAINT KONTEXT_DIAGNOSE_DIAGNOSE_FK FOREIGN KEY
678 (D_ID)REFERENCES DIAGNOSE(D_ID);
679 ALTER TABLE FALL_KONTEXT_PRUEFUNG_HINWEIS ADD CONSTRAINT F_K_PRF_HW_FALL_K_PRF_FK FOREIGN
680 KEY (FKP_ID)REFERENCES FALL_KONTEXT_PRUEFUNG(FKP_ID);
681 ALTER TABLE FALL_KONTEXT_PRUEFUNG_HINWEIS ADD CONSTRAINT F_K_PRF_HW_HINWEISTYP_FK FOREIGN
682 KEY (HWT_ID)REFERENCES HINWEISTYP(HWT_ID);
683 ALTER TABLE IMPORTDATEI_FELD ADD CONSTRAINT IDAT_FELD_IMPORTDATEI_FK FOREIGN KEY
684 (IMPDAT_ID)REFERENCES IMPORTDATEI(IMPDAT_ID);
685 ALTER TABLE IMPORTDATEI_FELD ADD CONSTRAINT IDAT_FELD_DATENTYP_FK FOREIGN KEY
686 (DAT_TYP_ID)REFERENCES DATENTYP(DAT_TYP_ID);
687 ALTER TABLE OPS301_THESAURUS ADD CONSTRAINT OPS301_THESAURUS_OPS301_FK FOREIGN KEY
688 (OPS_ID)REFERENCES OPS301(OPS_ID);
689 ALTER TABLE FALL_OPS301 ADD CONSTRAINT FALL_OPS_OPS301_FK FOREIGN KEY (OPS_ID)REFERENCES
690 OPS301(OPS_ID);
691 ALTER TABLE FALL_OPS301 ADD CONSTRAINT FALL_OPS_OPS_LOK_FK FOREIGN KEY
692 (OPSLOK_ID)REFERENCES OPS301_LOKALISATION(OPSLOK_ID);
693 ALTER TABLE FALL_OPS301 ADD CONSTRAINT FALL_OPS_FALL_FK FOREIGN KEY (FALL_ID)REFERENCES
694 FALL(FALL_ID);
695 ALTER TABLE FALL_OPS301 ADD CONSTRAINT FALL_OPS_OPS_KAT_FK FOREIGN KEY
696 (OPSKAT_ID)REFERENCES OPS301_KATEGORIE(OPSKAT_ID);
697 ALTER TABLE ELEMENT_PROZEDUR ADD CONSTRAINT E_PROZEDUR_PROZEDUR_FK FOREIGN KEY
698 (P_ID)REFERENCES PROZEDUR(P_ID);
699 ALTER TABLE ELEMENT_PROZEDUR ADD CONSTRAINT E_PROZEDUR_ELEMENT_FK FOREIGN KEY
700 (E_ID)REFERENCES ELEMENT(E_ID);
701 ALTER TABLE IMPORTDATEI_FELD_WERT ADD CONSTRAINT IDATF_WERT_IDAT_FELD_FK FOREIGN KEY
702 (IMPDATF_ID)REFERENCES IMPORTDATEI_FELD(IMPDATF_ID);
703 ALTER TABLE IMPORTDATEI_FELD_WERT ADD CONSTRAINT IDATF_WERT_WERT_FK FOREIGN KEY
704 (W_ID)REFERENCES WERT(W_ID);
705 ALTER TABLE FALL_KONTEXT_PRUEFUNG ADD CONSTRAINT FALL_K_PRF_FALL_FK FOREIGN KEY
706 (FALL_ID)REFERENCES FALL(FALL_ID);
707 ALTER TABLE FALL_KONTEXT_PRUEFUNG ADD CONSTRAINT FALL_K_PRF_KONT_PRUEF_FK FOREIGN KEY
708 (KPRF_ID)REFERENCES KONTEXT_PRUEFUNG(KPRF_ID);
709 ALTER TABLE DIAGNOSE_ICD ADD CONSTRAINT DIAGNOSE_ICD_DIAGNOSE_FK FOREIGN KEY
710 (D_ID)REFERENCES DIAGNOSE(D_ID);
711 ALTER TABLE DIAGNOSE_ICD ADD CONSTRAINT DIAGNOSE_ICD_ICD_FK FOREIGN KEY
712 (ICD_ID)REFERENCES ICD(ICD_ID);
713 ALTER TABLE FALL_OP_OPS301 ADD CONSTRAINT FALLOP_OPS_OPS_KAT_FK FOREIGN KEY
714 (OPSKAT_ID)REFERENCES OPS301_KATEGORIE(OPSKAT_ID);
715 ALTER TABLE FALL_OP_OPS301 ADD CONSTRAINT FALLOP_OPS_OPS_LOK_FK FOREIGN KEY
716 (OPSLOK_ID)REFERENCES OPS301_LOKALISATION(OPSLOK_ID);
717 ALTER TABLE FALL_OP_OPS301 ADD CONSTRAINT FALLOP_OPS_FALL_OP_FK FOREIGN KEY
718 (FALLOP_ID)REFERENCES FALL_OP(FALLOP_ID);
719 ALTER TABLE FALL_OP_OPS301 ADD CONSTRAINT FALLOP_OPS_OPS301_FK FOREIGN KEY
720 (OPS_ID)REFERENCES OPS301(OPS_ID);
721 ALTER TABLE ELEMENT_BEZIEHUNG ADD CONSTRAINT ELEM_BEZ_ELEMENT_2_FK FOREIGN KEY
722 (E_ID_2)REFERENCES ELEMENT(E_ID);
723 ALTER TABLE ELEMENT_BEZIEHUNG ADD CONSTRAINT ELEM_BEZ_E_BEZTYP_FK FOREIGN KEY
724 (EBEZTYP_ID)REFERENCES ELEMENT_BEZIEHUNGSTYP(EBEZTYP_ID);
725 ALTER TABLE ELEMENT_BEZIEHUNG ADD CONSTRAINT ELEM_BEZ_ELEMENT_1_FK FOREIGN KEY
726 (E_ID_1)REFERENCES ELEMENT(E_ID);
727 ALTER TABLE WERT_DIAGNOSE ADD CONSTRAINT WERT_DIAG_DIAGNOSE_FK FOREIGN KEY
728 (D_ID)REFERENCES DIAGNOSE(D_ID);
729 ALTER TABLE WERT_DIAGNOSE ADD CONSTRAINT WERT_DIAG_WERT_FK FOREIGN KEY (W_ID)REFERENCES
730 WERT(W_ID);
731 ALTER TABLE ELEMENT_WERT_WERT ADD CONSTRAINT E_W_WERT_WERT_WERT_FK FOREIGN KEY
732 (WW_ID)REFERENCES WERT_WERT(WW_ID);
733 ALTER TABLE ELEMENT_WERT_WERT ADD CONSTRAINT E_W_WERT_ELEMENT_FK FOREIGN KEY
734 (E_ID)REFERENCES ELEMENT(E_ID);
735 ALTER TABLE OPS301 ADD CONSTRAINT OPS301_OPS301_VERSION_FK FOREIGN KEY
736 (OPSVER_ID)REFERENCES OPS301_VERSION(OPSVER_ID);
737 ALTER TABLE KOMPLEX_WERT_D_WERT ADD CONSTRAINT KOMPLEX_WERT_D_WERT_K_W_FK FOREIGN KEY
738 (KPW_ID)REFERENCES KOMPLEX_WERT(KPW_ID);
739 ALTER TABLE KOMPLEX_WERT_D_WERT ADD CONSTRAINT KOMPLEX_WERT_D_WERT_D_W_FK FOREIGN KEY
740 (DW_ID)REFERENCES DIAGNOSE_WERT(DW_ID);
741 ALTER TABLE ELEMENT_DIAGNOSE_WERT ADD CONSTRAINT E_D_WERT_ELEMENT_FK FOREIGN KEY

```

(E_ID)REFERENCES ELEMENT(E_ID);	742
ALTER TABLE ELEMENT_DIAGNOSE_WERT ADD CONSTRAINT E_D_WERT_DIAGNOSE_WERT_FK FOREIGN KEY	743
(DW_ID)REFERENCES DIAGNOSE_WERT(DW_ID);	744
ALTER TABLE FALL_WERT ADD CONSTRAINT FALL_WERT_WERT_FK FOREIGN KEY (W_ID)REFERENCES	745
WERT(W_ID);	746
ALTER TABLE FALL_WERT ADD CONSTRAINT FALL_WERT_FALL_FK FOREIGN KEY (FALL_ID)REFERENCES	747
FALL(FALL_ID);	748
ALTER TABLE PROZEDUR_WERT ADD CONSTRAINT PROZEDUR_WERT_EJN_AUSW_FK FOREIGN KEY	749
(EJN_ID_AUSWERTUNG)REFERENCES ENTSCHEIDUNG_JA_NEIN(EJN_ID);	750
ALTER TABLE PROZEDUR_WERT ADD CONSTRAINT PROZEDUR_WERT_PROZEDUR_FK FOREIGN KEY	751
(P_ID)REFERENCES PROZEDUR(P_ID);	752
ALTER TABLE PROZEDUR_WERT_KOMBINATION ADD CONSTRAINT PROZ_WERT_KOMB_EJN_AUSW_FK FOREIGN	753
KEY (EJN_ID_AUSWERTUNG)REFERENCES ENTSCHEIDUNG_JA_NEIN(EJN_ID);	754
ALTER TABLE PROZEDUR_WERT_KOMBINATION ADD CONSTRAINT PROZ_WERT_KOMB_PW_KOMBTYP_FK FOREIGN	755
KEY (PWKT_ID)REFERENCES PROZEDUR_WERT_KOMBINATIONSTYP(PWKT_ID);	756
ALTER TABLE PROZEDUR_WERT_KOMBINATION ADD CONSTRAINT PROZ_WERT_KOMB_PROZEDUR_FK FOREIGN	757
KEY (P_ID)REFERENCES PROZEDUR(P_ID);	758
ALTER TABLE ICD_THESAURUS ADD CONSTRAINT ICD_THESAURUS_ICD_FK FOREIGN KEY	759
(ICD_ID)REFERENCES ICD(ICD_ID);	760
ALTER TABLE KONTEXT_DIAGNOSE_PHASE ADD CONSTRAINT KT_DIAG_PH_PHASE_FK FOREIGN KEY	761
(PH_ID)REFERENCES PHASE(PH_ID);	762
ALTER TABLE KONTEXT_DIAGNOSE_PHASE ADD CONSTRAINT KT_DIAG_PH_KONT_DIAG_FK FOREIGN KEY	763
(KD_ID)REFERENCES KONTEXT_DIAGNOSE(KD_ID);	764
ALTER TABLE PRUEFUNG_KOPPELUNG_ELEMENTE ADD CONSTRAINT PRF_KOPP_E_ELEMENT_FK FOREIGN KEY	765
(E_ID)REFERENCES ELEMENT(E_ID);	766
ALTER TABLE PRUEFUNG_KOPPELUNG_ELEMENTE ADD CONSTRAINT PRF_KOPP_E_PRUEF_KOPP_FK FOREIGN	767
KEY (PRFK_ID)REFERENCES PRUEFUNG_KOPPELUNG(PRFK_ID);	768
ALTER TABLE KONTEXT_WERT ADD CONSTRAINT KONT_WERT_WERT_FK FOREIGN KEY (W_ID)REFERENCES	769
WERT(W_ID);	770
ALTER TABLE KONTEXT_WERT ADD CONSTRAINT KONT_WERT_ENTSCH_J_N_FK FOREIGN KEY	771
(EJN_ID_PFLICHTANGABE)REFERENCES ENTSCHEIDUNG_JA_NEIN(EJN_ID);	772
ALTER TABLE KONTEXT_WERT ADD CONSTRAINT KONT_WERT__FK FOREIGN KEY (K_ID)REFERENCES	773
KONTEXT(K_ID);	774
ALTER TABLE KOMPLEX_WERT_D_WERT_KOMB ADD CONSTRAINT KOMPLEX_WERT_D_W_K_D_W_K_FK FOREIGN	775
KEY (DWK_ID)REFERENCES DIAGNOSE_WERT_KOMBINATION(DWK_ID);	776
ALTER TABLE KOMPLEX_WERT_D_WERT_KOMB ADD CONSTRAINT KOMPLEX_WERT_D_W_K_K_W_FK FOREIGN KEY	777
(KPW_ID)REFERENCES KOMPLEX_WERT(KPW_ID);	778
ALTER TABLE KOMPLEX_WERT ADD CONSTRAINT KOMPLEX_WERT_KW_KOMBTYP_FK FOREIGN KEY	779
(KWT_ID)REFERENCES KOMPLEX_WERT_KOMBINATIONSTYP(KWT_ID);	780
ALTER TABLE KOMPLEX_WERT ADD CONSTRAINT KOMPLEX_WERT_KOMPLEX_FK FOREIGN KEY	781
(KP_ID)REFERENCES KOMPLEX(KP_ID);	782
ALTER TABLE KOMPLEX_WERT ADD CONSTRAINT KOMPLEX_WERT_EJN_FK FOREIGN KEY	783
(EJN_ID_AUSWERTUNG)REFERENCES ENTSCHEIDUNG_JA_NEIN(EJN_ID);	784
ALTER TABLE ICD ADD CONSTRAINT ICD_ICD_VER_FK FOREIGN KEY (ICDVER_ID)REFERENCES	785
ICD_VERSION(ICDVER_ID);	786
ALTER TABLE KOMPLEX_WERT_P_WERT_KOMB ADD CONSTRAINT KOMPLEX_WERT_P_W_K_K_W_FK FOREIGN KEY	787
(KPW_ID)REFERENCES KOMPLEX_WERT(KPW_ID);	788
ALTER TABLE KOMPLEX_WERT_P_WERT_KOMB ADD CONSTRAINT KOMPLEX_WERT_P_W_K_P_W_K_FK FOREIGN	789
KEY (PWK_ID)REFERENCES PROZEDUR_WERT_KOMBINATION(PWK_ID);	790
ALTER TABLE DIAGNOSE_WERT_ICD_LOK ADD CONSTRAINT D_WERT_ICD_ICD_LOK_FK FOREIGN KEY	791
(ICDLOK_ID)REFERENCES ICD_LOKALISATION(ICDLOK_ID);	792
ALTER TABLE DIAGNOSE_WERT_ICD_LOK ADD CONSTRAINT D_WERT_ICD_LOK_D_WERT_ICD_FK FOREIGN KEY	793
(DWI_ID)REFERENCES DIAGNOSE_WERT_ICD(DWI_ID);	794
ALTER TABLE ELEMENT_KOMPLEX ADD CONSTRAINT E_KOMPLEX_KOMPLEX_FK FOREIGN KEY	795
(KP_ID)REFERENCES KOMPLEX(KP_ID);	796
ALTER TABLE ELEMENT_KOMPLEX ADD CONSTRAINT E_KOMPLEX_ELEMENT_FK FOREIGN KEY	797
(E_ID)REFERENCES ELEMENT(E_ID);	798
ALTER TABLE FALL_ICD ADD CONSTRAINT FALL_ICD_ICD_LOK_FK FOREIGN KEY (ICDLOK_ID)REFERENCES	799
ICD_LOKALISATION(ICDLOK_ID);	800
ALTER TABLE FALL_ICD ADD CONSTRAINT FALL_ICD_ICD_KAT_FK FOREIGN KEY (ICDKAT_ID)REFERENCES	801
ICD_KATEGORIE(ICDKAT_ID);	802
ALTER TABLE FALL_ICD ADD CONSTRAINT FALL_ICD_ICD_FK FOREIGN KEY (ICD_ID)REFERENCES	803
ICD(ICD_ID);	804
ALTER TABLE FALL_ICD ADD CONSTRAINT FALL_ICD_FALL_FK FOREIGN KEY (FALL_ID)REFERENCES	805
FALL(FALL_ID);	806
ALTER TABLE PRUEFUNG_KOPPELUNG ADD CONSTRAINT PRUEF_KOPP_PRUEFUNG_FK FOREIGN KEY	807
(PRF_ID)REFERENCES PRUEFUNG(PRIF_ID);	808
ALTER TABLE PROZEDUR_WERT_OPS_LOK ADD CONSTRAINT P_WERT_OPS_OPS_LOK_FK FOREIGN KEY	809

```

810 (OPSLOK_ID)REFERENCES OPS301_LOKALISATION(OPSLOK_ID);
811 ALTER TABLE PROZEDUR_WERT_OPS_LOK ADD CONSTRAINT P_WERT_OPS_LOK_P_WERT_OPS_FK FOREIGN KEY
812 (PWO_ID)REFERENCES PROZEDUR_WERT_OPS(PWO_ID);
813 ALTER TABLE DIAGNOSE_WERT ADD CONSTRAINT DIAGNOSE_WERT_DIAGNOSE_FK FOREIGN KEY
814 (D_ID)REFERENCES DIAGNOSE(D_ID);
815 ALTER TABLE DIAGNOSE_WERT ADD CONSTRAINT DIAGNOSE_WERT_EJN_AUSW_FK FOREIGN KEY
816 (EJN_ID_AUSWERTUNG)REFERENCES ENTSCHEIDUNG_JA_NEIN(EJN_ID);
817 ALTER TABLE PROZEDUR_WERT_OPS ADD CONSTRAINT PROZEDUR_WERT_OPS_OPS_FK FOREIGN KEY
818 (OPS_ID)REFERENCES OPS301(OPS_ID);
819 ALTER TABLE PROZEDUR_WERT_OPS ADD CONSTRAINT PROZEDUR_WERT_OPS_PW_FK FOREIGN KEY
820 (PW_ID)REFERENCES PROZEDUR_WERT(PW_ID);
821 ALTER TABLE OPS301_SUCHTEXT ADD CONSTRAINT OPS301_SUCHTEXT_OPS301_FK FOREIGN KEY
822 (OPS_ID)REFERENCES OPS301(OPS_ID);
823 ALTER TABLE KONTEXT_KOMPLEX ADD CONSTRAINT KONTEXT_KOMPLEX_KOMPLEX_FK FOREIGN KEY
824 (KP_ID)REFERENCES KOMPLEX(KP_ID);
825 ALTER TABLE KONTEXT_KOMPLEX ADD CONSTRAINT KONTEXT_KOMPLEX_KONTEXT_FK FOREIGN KEY
826 (K_ID)REFERENCES KONTEXT(K_ID);
827 ALTER TABLE PRUEFUNG ADD CONSTRAINT PRUEFUNG_PRUEFTYP_FK FOREIGN KEY (PRFT_ID)REFERENCES
828 PRUEFUNGSTYP(PRFT_ID);
829 ALTER TABLE PRUEFUNG ADD CONSTRAINT PRUEFUNG_RELEVANZ_FK FOREIGN KEY (REL_ID)REFERENCES
830 RELEVANZ(REL_ID);
831 ALTER TABLE ELEMENT_DIAGNOSE_WERT_KOMB ADD CONSTRAINT E_D_W_KOMB_ELEMENT_FK FOREIGN KEY
832 (E_ID)REFERENCES ELEMENT(E_ID);
833 ALTER TABLE ELEMENT_DIAGNOSE_WERT_KOMB ADD CONSTRAINT E_D_W_KOMB_D_W_K_FK FOREIGN KEY
834 (DWK_ID)REFERENCES DIAGNOSE_WERT_KOMBINATION(DWK_ID);
835 ALTER TABLE KONTEXT_WERT_WERT ADD CONSTRAINT K_W_WERT_WERT_WERT_FK FOREIGN KEY
836 (WW_ID)REFERENCES WERT_WERT(WW_ID);
837 ALTER TABLE KONTEXT_WERT_WERT ADD CONSTRAINT K_W_WERT_KONTEXT_FK FOREIGN KEY
838 (K_ID)REFERENCES KONTEXT(K_ID);
839 ALTER TABLE KONTEXT_PROZEDUR_PHASE ADD CONSTRAINT KT_PROZ_PH_PHASE_FK FOREIGN KEY
840 (PH_ID)REFERENCES PHASE(PH_ID);
841 ALTER TABLE KONTEXT_PROZEDUR_PHASE ADD CONSTRAINT KT_PROZ_PH_KONT_PROZ_FK FOREIGN KEY
842 (KP_ID)REFERENCES KONTEXT_PROZEDUR(KP_ID);
843 ALTER TABLE KOMPLEX_WERT_P_WERT ADD CONSTRAINT KOMPLEX_WERT_P_WERT_K_W_FK FOREIGN KEY
844 (KPW_ID)REFERENCES KOMPLEX_WERT(KPW_ID);
845 ALTER TABLE KOMPLEX_WERT_P_WERT ADD CONSTRAINT KOMPLEX_WERT_P_WERT_P_W_FK FOREIGN KEY
846 (PW_ID)REFERENCES PROZEDUR_WERT(PW_ID);
847 ALTER TABLE PROZ_WERT_KOMB_PROZ_WERT ADD CONSTRAINT P_W_K_P_W_P_WERT_KOMB_FK FOREIGN KEY
848 (PWK_ID)REFERENCES PROZEDUR_WERT_KOMBINATION(PWK_ID);
849 ALTER TABLE PROZ_WERT_KOMB_PROZ_WERT ADD CONSTRAINT P_W_K_P_W_P_WERT_FK FOREIGN KEY
850 (PW_ID)REFERENCES PROZEDUR_WERT(PW_ID);
851 ALTER TABLE KONTEXT_PROZEDUR ADD CONSTRAINT KONTEXT_PROZEDUR_KONTEXT_FK FOREIGN KEY
852 (K_ID)REFERENCES KONTEXT(K_ID);
853 ALTER TABLE KONTEXT_PROZEDUR ADD CONSTRAINT KONTEXT_PROZEDUR_PROZEDUR_FK FOREIGN KEY
854 (P_ID)REFERENCES PROZEDUR(P_ID);
855 ALTER TABLE KONTEXT_DIAG_WERT_KOMB ADD CONSTRAINT K_D_W_KOMB_KONTEXT_FK FOREIGN KEY
856 (K_ID)REFERENCES KONTEXT(K_ID);
857 ALTER TABLE KONTEXT_DIAG_WERT_KOMB ADD CONSTRAINT K_D_W_KOMB_DIAG_W_K_FK FOREIGN KEY
858 (DWK_ID)REFERENCES DIAGNOSE_WERT_KOMBINATION(DWK_ID);
859 ALTER TABLE SUPERKONTEXT_KONTEXT ADD CONSTRAINT SUPKON_KON_SUPERKON_FK FOREIGN KEY
860 (SK_ID)REFERENCES SUPERKONTEXT(SK_ID);
861 ALTER TABLE SUPERKONTEXT_KONTEXT ADD CONSTRAINT SUPKON_KON_KONTEXT_FK FOREIGN KEY
862 (K_ID)REFERENCES KONTEXT(K_ID);
863 ALTER TABLE PRUEFUNG_AUSSCHLUSS_ELEMENTE ADD CONSTRAINT PRF_AUSS_E_PRUEF_AUSS_FK FOREIGN
864 KEY (PRFA_ID)REFERENCES PRUEFUNG_AUSSCHLUSS(PRFA_ID);
865 ALTER TABLE PRUEFUNG_AUSSCHLUSS_ELEMENTE ADD CONSTRAINT PRF_AUSS_E_ELEMENT_FK FOREIGN KEY
866 (E_ID)REFERENCES ELEMENT(E_ID);
867 ALTER TABLE ELEMENT_PROZEDUR_WERT_KOMB ADD CONSTRAINT E_P_W_KOMB_PWK_FK FOREIGN KEY
868 (PWK_ID)REFERENCES PROZEDUR_WERT_KOMBINATION(PWK_ID);
869 ALTER TABLE ELEMENT_PROZEDUR_WERT_KOMB ADD CONSTRAINT E_P_W_KOMB_ELEMENT_FK FOREIGN KEY
870 (E_ID)REFERENCES ELEMENT(E_ID);
871 ALTER TABLE PHASE_ICD_KATEGORIE ADD CONSTRAINT PH_ICD_KAT_PH_ZUOTYP_FK FOREIGN KEY
872 (PHZT_ID)REFERENCES PHASE_ZUORDNUNGSTYP(PHZT_ID);
873 ALTER TABLE PHASE_ICD_KATEGORIE ADD CONSTRAINT PH_ICD_KAT_PHASE_FK FOREIGN KEY
874 (PH_ID)REFERENCES PHASE(PH_ID);
875 ALTER TABLE PHASE_ICD_KATEGORIE ADD CONSTRAINT PH_ICD_KAT_ICD_KAT_FK FOREIGN KEY
876 (ICDKAT_ID)REFERENCES ICD_KATEGORIE(ICDKAT_ID);
877 ALTER TABLE FALL_OP_WERT ADD CONSTRAINT FOP_WERT_WERT_FK FOREIGN KEY (W_ID)REFERENCES

```

WERT(W_ID);	878
ALTER TABLE FALL_OP_WERT ADD CONSTRAINT FOP_WERT_FALL_OP_FK FOREIGN KEY	879
(FALLOP_ID)REFERENCES FALL_OP(FALLOP_ID);	880
ALTER TABLE PROZEDUR_OPS301 ADD CONSTRAINT PROZEDUR_OPS301_PROZEDUR_FK FOREIGN KEY	881
(P_ID)REFERENCES PROZEDUR(P_ID);	882
ALTER TABLE PROZEDUR_OPS301 ADD CONSTRAINT PROZEDUR_OPS301_OPS301_FK FOREIGN KEY	883
(OPS_ID)REFERENCES OPS301(OPS_ID);	884
ALTER TABLE IMPORDATEI ADD CONSTRAINT IMPDATEI_KIS_FK FOREIGN KEY (KIS_ID)REFERENCES	885
KIS(KIS_ID);	886
ALTER TABLE DIAGNOSE_WERT_KOMBINATION ADD CONSTRAINT DIAG_WERT_KOMB_EJN_AUSW_FK FOREIGN	887
KEY (EJN_ID_AUSWERTUNG)REFERENCES ENTSCHEIDUNG_JA_NEIN(EJN_ID);	888
ALTER TABLE DIAGNOSE_WERT_KOMBINATION ADD CONSTRAINT DIAG_WERT_KOMB_DIAG_FK FOREIGN KEY	889
(D_ID)REFERENCES DIAGNOSE(D_ID);	890
ALTER TABLE DIAGNOSE_WERT_KOMBINATION ADD CONSTRAINT DIAG_WERT_KOMB_DW_KOMBTYP_FK FOREIGN	891
KEY (DWKT_ID)REFERENCES DIAGNOSE_WERT_KOMBINATIONSTYP(DWKT_ID);	892
ALTER TABLE WERT_PROZEDUR ADD CONSTRAINT WERT_PROZ_PROZEDUR_FK FOREIGN KEY	893
(P_ID)REFERENCES PROZEDUR(P_ID);	894
ALTER TABLE WERT_PROZEDUR ADD CONSTRAINT WERT_PROZ_WERT_FK FOREIGN KEY (W_ID)REFERENCES	895
WERT(W_ID);	896
ALTER TABLE ELEMENT_DIAGNOSE ADD CONSTRAINT E_DIAGNOSE_DIAGNOSE_FK FOREIGN KEY	897
(D_ID)REFERENCES DIAGNOSE(D_ID);	898
ALTER TABLE ELEMENT_DIAGNOSE ADD CONSTRAINT E_DIAGNOSE_ELEMENT_FK FOREIGN KEY	899
(E_ID)REFERENCES ELEMENT(E_ID);	900
ALTER TABLE ELEMENT_WERT_BEREICH ADD CONSTRAINT E_W_BER_WERT_BER_FK FOREIGN KEY	901
(WB_ID)REFERENCES WERT_BEREICH(WB_ID);	902
ALTER TABLE ELEMENT_WERT_BEREICH ADD CONSTRAINT E_W_BER_ELEMENT_FK FOREIGN KEY	903
(E_ID)REFERENCES ELEMENT(E_ID);	904
ALTER TABLE KONTEXT_PRUEFUNG ADD CONSTRAINT KONT_PRUEF_KONTEXT_FK FOREIGN KEY	905
(K_ID)REFERENCES KONTEXT(K_ID);	906
ALTER TABLE KONTEXT_PRUEFUNG ADD CONSTRAINT KONT_PRUEF_PRUEFUNG_FK FOREIGN KEY	907
(PRF_ID)REFERENCES PRUEFUNG(PRF_ID);	908
ALTER TABLE PHASE_OPS301_KATEGORIE ADD CONSTRAINT PH_OPS_KAT_OPS_KAT_FK FOREIGN KEY	909
(OPSKAT_ID)REFERENCES OPS301_KATEGORIE(OPSKAT_ID);	910
ALTER TABLE PHASE_OPS301_KATEGORIE ADD CONSTRAINT PH_OPS_KAT_PHASE_FK FOREIGN KEY	911
(PH_ID)REFERENCES PHASE(PH_ID);	912
ALTER TABLE PHASE_OPS301_KATEGORIE ADD CONSTRAINT PH_OPS_KAT_PH_ZUOTYP_FK FOREIGN KEY	913
(PHZT_ID)REFERENCES PHASE_ZUORDNUNGSTYP(PHZT_ID);	914
ALTER TABLE DIAG_WERT_KOMB_DIAG_WERT ADD CONSTRAINT DIAG_WERT_KOMB_D_W_DWK_FK FOREIGN KEY	915
(DWK_ID)REFERENCES DIAGNOSE_WERT_KOMBINATION(DWK_ID);	916
ALTER TABLE DIAG_WERT_KOMB_DIAG_WERT ADD CONSTRAINT DIAG_WERT_KOMB_D_W_DW_FK FOREIGN KEY	917
(DW_ID)REFERENCES DIAGNOSE_WERT(DW_ID);	918
ALTER TABLE FALL ADD CONSTRAINT FALL_FACHABT_FK FOREIGN KEY (FA_ID)REFERENCES	919
FACHABTEILUNG(FA_ID);	920
ALTER TABLE FALL ADD CONSTRAINT FALL_KIS_FK FOREIGN KEY (KIS_ID)REFERENCES KIS(KIS_ID);	921
ALTER TABLE FALL ADD CONSTRAINT FALL_BEARBEITER_FK FOREIGN KEY (BA_ID)REFERENCES	922
BEARBEITER(BA_ID);	923
ALTER TABLE FALL ADD CONSTRAINT FALL_GESCHLECHT_FK FOREIGN KEY (G_ID)REFERENCES	924
GESCHLECHT(G_ID);	925
ALTER TABLE FALL ADD CONSTRAINT FALL_EJN_ID_PUFFER_BEREIT_FK FOREIGN KEY	926
(EJN_ID_PUFFER_BEREIT)REFERENCES ENTSCHEIDUNG_JA_NEIN(EJN_ID);	927
ALTER TABLE FALL_OP ADD CONSTRAINT FALL_OP_FALL_FK FOREIGN KEY (FALL_ID)REFERENCES	928
FALL(FALL_ID);	929
ALTER TABLE KONTEXT_PROZEDUR_WERT ADD CONSTRAINT K_P_WERT_PROZ_WERT_FK FOREIGN KEY	930
(PW_ID)REFERENCES PROZEDUR_WERT(PW_ID);	931
ALTER TABLE KONTEXT_PROZEDUR_WERT ADD CONSTRAINT K_P_WERT_KONTEXT_FK FOREIGN KEY	932
(K_ID)REFERENCES KONTEXT(K_ID);	933
ALTER TABLE PROZEDUR_OPS301_LOK ADD CONSTRAINT PROZ_OPS_LOK_OPS_LOK_FK FOREIGN KEY	934
(OPSLOK_ID)REFERENCES OPS301_LOKALISATION(OPSLOK_ID);	935
ALTER TABLE PROZEDUR_OPS301_LOK ADD CONSTRAINT PROZ_OPS_LOK_PROZ_OPS_FK FOREIGN KEY	936
(PO_ID)REFERENCES PROZEDUR_OPS301(PO_ID);	937
ALTER TABLE FALL_OP_ICD ADD CONSTRAINT FALLOP_ICD_FALL_OP_FK FOREIGN KEY	938
(FALLOP_ID)REFERENCES FALL_OP(FALLOP_ID);	939
ALTER TABLE FALL_OP_ICD ADD CONSTRAINT FALLOP_ICD_ICD_FK FOREIGN KEY (ICD_ID)REFERENCES	940
ICD(ICD_ID);	941
ALTER TABLE FALL_OP_ICD ADD CONSTRAINT FALLOP_ICD_ICD_KAT_FK FOREIGN KEY	942
(ICDKAT_ID)REFERENCES ICD_KATEGORIE(ICDKAT_ID);	943
ALTER TABLE FALL_OP_ICD ADD CONSTRAINT FALLOP_ICD_ICD_LOK_FK FOREIGN KEY	944
(ICDLOK_ID)REFERENCES ICD_LOKALISATION(ICDLOK_ID);	945

```

946 ALTER TABLE PRUEFUNG_ELEMENT ADD CONSTRAINT PRUEF_ELEM_PRUEFUNG_FK FOREIGN KEY
947 (PRF_ID)REFERENCES PRUEFUNG(PR_FID);
948 ALTER TABLE PRUEFUNG_ELEMENT ADD CONSTRAINT PRUEF_ELEM_ELEMENT_FK FOREIGN KEY
949 (E_ID)REFERENCES ELEMENT(E_ID);
950 ALTER TABLE DIAGNOSE_ICD_LOK ADD CONSTRAINT D_ICD_LOK_D_ICD_FK FOREIGN KEY
951 (DI_ID)REFERENCES DIAGNOSE_ICD(DI_ID);
952 ALTER TABLE DIAGNOSE_ICD_LOK ADD CONSTRAINT D_ICD_LOK_ICD_LOK_FK FOREIGN KEY
953 (ICDLOK_ID)REFERENCES ICD_LOKALISATION(ICDLOK_ID);
954 ALTER TABLE KONTEXT_WERT_BEREICH ADD CONSTRAINT K_W_BER_KONTEXT_FK FOREIGN KEY
955 (K_ID)REFERENCES KONTEXT(K_ID);
956 ALTER TABLE KONTEXT_WERT_BEREICH ADD CONSTRAINT K_W_BER_WERT_BER_FK FOREIGN KEY
957 (WB_ID)REFERENCES WERT_BEREICH(WB_ID);
958 ALTER TABLE KONTEXT_ELEMENT_BEZIEHUNG ADD CONSTRAINT KONT_E_BEZ_ELEM_BEZ_FK FOREIGN KEY
959 (EBEZ_ID)REFERENCES ELEMENT_BEZIEHUNG(EBEZ_ID);
960 ALTER TABLE KONTEXT_ELEMENT_BEZIEHUNG ADD CONSTRAINT KONT_E_BEZ_KONTEXT_FK FOREIGN KEY
961 (K_ID)REFERENCES KONTEXT(K_ID);
962 ALTER TABLE ICD_SUCHTEXT ADD CONSTRAINT ICD_SUCHTEXT_ICD_FK FOREIGN KEY
963 (ICD_ID)REFERENCES ICD(ICD_ID);
964 ALTER TABLE KONTEXT_DIAGNOSE_WERT ADD CONSTRAINT K_D_WERT_KONTEXT_FK FOREIGN KEY
965 (K_ID)REFERENCES KONTEXT(K_ID);
966 ALTER TABLE KONTEXT_DIAGNOSE_WERT ADD CONSTRAINT K_D_WERT_DIAG_WERT_FK FOREIGN KEY
967 (DW_ID)REFERENCES DIAGNOSE_WERT(DW_ID);
968 ALTER TABLE ELEMENT_KOMPLEX_WERT ADD CONSTRAINT E_K_WERT_ELEMENT_FK FOREIGN KEY
969 (E_ID)REFERENCES ELEMENT(E_ID);
970 ALTER TABLE ELEMENT_KOMPLEX_WERT ADD CONSTRAINT E_K_WERT_KOMPLEX_WERT_FK FOREIGN KEY
971 (KPW_ID)REFERENCES KOMPLEX_WERT(KPW_ID);
972 ALTER TABLE WERT_KOMPLEX ADD CONSTRAINT WERT_KOMP_WERT_FK FOREIGN KEY (W_ID)REFERENCES
973 WERT(W_ID);
974 ALTER TABLE WERT_KOMPLEX ADD CONSTRAINT WERT_KOMP_KOMPLEX_FK FOREIGN KEY
975 (KP_ID)REFERENCES KOMPLEX(KP_ID);
976 ALTER TABLE KONTEXT_PROZ_WERT_KOMB ADD CONSTRAINT K_P_W_KOMB_P_WERT_K_FK FOREIGN KEY
977 (PWK_ID)REFERENCES PROZEDUR_WERT_KOMBINATION(PWK_ID);
978 ALTER TABLE KONTEXT_PROZ_WERT_KOMB ADD CONSTRAINT K_P_W_KOMB_KONTEXT_FK FOREIGN KEY
979 (K_ID)REFERENCES KONTEXT(K_ID);

```



# Anhang 4 – Datenbank-Abfragen zu Ableitung von Daten aus ICD- und OPS-Schlüsseln

## A4.1. View [Fall\_ICD\_Inkl\_Op]

```
CREATE OR REPLACE VIEW FALL_ICD_INKL_OP AS
SELECT
FAL.FALL_ID AS FALL_ID,
FAL.CODE AS FALL_CODE,
FAL.NACHNAME AS NACHNAME,
FAL.VORNAME AS VORNAME,
FAL.GEBURTSDATUM AS GEBURTSDATUM,
FAL.AUFNAHME_FA AS AUFNAHME_FA,
FAL.ABGANG_FA AS ABGANG_FA,
FAL.AUFNAHME_KH AS AUFNAHME_KH,
FAL.ABGANG_KH AS ABGANG_KH,
FAL.G_ID AS G_ID,
FAL.BA_ID AS BA_ID,
BEA.NAME || ', ' || BEA.VORNAME AS BEARBEITER,
FAL.KIS_ID AS KIS_ID,
KIS.BEZEICHNUNG AS KIS_BEZEICHNUNG,
FAL.FA_ID AS FA_ID,
FAB.BEZEICHNUNG AS FACHABTEILUNG_BEZEICHNUNG,
NULL AS FALLOP_ID,
NULL AS OP_BEGINN,
NULL AS OP_ENDE,
FIC.FALLICD_ID AS FALLICD_ID,
NULL AS FOPICD_ID,
FIC.ICD_ID AS ICD_ID,
ICD.CODE AS ICD_CODE,
ICD.KURZTEXT AS ICD_KURZTEXT,
IVE.ICDVER_ID AS ICDVER_ID,
IVE.BEZEICHNUNG AS ICD_VERSION_BEZEICHNUNG,
FIC.ICDKAT_ID AS ICDKAT_ID,
IKA.BEZEICHNUNG AS ICD_KATEGORIE_BEZEICHNUNG,
FIC.ICDLOK_ID AS ICDLOK_ID,
ILO.BEZEICHNUNG AS ICD_LOKALISATION_BEZEICHNUNG,
FIC.DATUM AS DATUM
FROM
FALL FAL,
FALL_ICD FIC,
ICD ICD,
ICD_VERSION IVE,
ICD_KATEGORIE IKA,
ICD_LOKALISATION ILO,
KIS KIS,
FACHABTEILUNG FAB,
BEARBEITER BEA
WHERE
FAL.FALL_ID = FIC.FALL_ID
AND
FIC.ICD_ID = ICD.ICD_ID
AND
FIC.ICDKAT_ID = IKA.ICDKAT_ID
AND
FIC.ICDLOK_ID = ILO.ICDLOK_ID
AND
FAL.KIS_ID = KIS.KIS_ID
AND
FAL.FA_ID = FAB.FA_ID
AND
FAL.BA_ID = BEA.BA_ID
AND
```

```

59 ICD.ICDVER_ID = IVE.ICDVER_ID
60 UNION
61 SELECT
62 FAL.FALL_ID,
63 FAL.CODE,
64 FAL.NACHNAME,
65 FAL.VORNAME,
66 FAL.GEBURTSDATUM,
67 FAL.AUFNAHME_FA,
68 FAL.ABGANG_FA,
69 FAL.AUFNAHME_KH,
70 FAL.ABGANG_KH,
71 FAL.G_ID,
72 FAL.BA_ID,
73 BEA.NAME || ', ' || BEA.VORNAME,
74 FAL.KIS_ID,
75 KIS.BEZEICHNUNG,
76 FAL.FA_ID,
77 FAB.BEZEICHNUNG,
78 FOP.FALLOP_ID,
79 FOP.BEGINN,
80 FOP.ENDE,
81 NULL,
82 FOI.FOPICD_ID,
83 FOI.ICD_ID,
84 ICD.CODE,
85 ICD.KURZTEXT,
86 IVE.ICDVER_ID,
87 IVE.BEZEICHNUNG,
88 FOI.ICDKAT_ID,
89 IKA.BEZEICHNUNG,
90 FOI.ICDLOK_ID,
91 ILO.BEZEICHNUNG,
92 FOP.BEGINN
93 FROM
94 FALL                FAL,
95 FALL_OP            FOP,
96 FALL_OP_ICD        FOI,
97 ICD                ICD,
98 ICD_VERSION        IVE,
99 ICD_KATEGORIE      IKA,
100 ICD_LOKALISATION  ILO,
101 KIS                KIS,
102 FACHABTEILUNG     FAB,
103 BEARBEITER        BEA
104 WHERE
105 FAL.FALL_ID = FOP.FALL_ID
106 AND
107 FOP.FALLOP_ID = FOI.FALLOP_ID
108 AND
109 FOI.ICD_ID = ICD.ICD_ID
110 AND
111 FOI.ICDKAT_ID = IKA.ICDKAT_ID
112 AND
113 FOI.ICDLOK_ID = ILO.ICDLOK_ID
114 AND
115 FAL.KIS_ID = KIS.KIS_ID
116 AND
117 FAL.FA_ID = FAB.FA_ID
118 AND
119 FAL.BA_ID = BEA.BA_ID
120 AND
121 ICD.ICDVER_ID = IVE.ICDVER_ID;

```

## A4.2. View [View Fall\_Kontext\_Diag\_Wert\_ICD]

```
CREATE OR REPLACE VIEW FALL_KONTEXT_DIAG_WERT_ICD AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FAL.CODE             AS FALL_CODE,
FAL.NACHNAME         AS NACHNAME,
FAL.VORNAME          AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA      AS AUFNAHME_FA,
FAL.ABGANG_FA        AS ABGANG_FA,
FAL.AUFNAHME_KH      AS AUFNAHME_KH,
FAL.ABGANG_KH        AS ABGANG_KH,
DIA.D_ID             AS D_ID,
DIA.BEZEICHNUNG      AS DIAGNOSE_BEZEICHNUNG,
DIW.DW_ID            AS DW_ID,
DIW.BEZEICHNUNG      AS DIAGNOSE_WERT_BEZEICHNUNG,
DIW.WERT             AS DIAGNOSE_WERT,
DIW.RANG             AS DIAGNOSE_WERT_RANG,
DIW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
KDI.KD_ID            AS KD_ID,
KON.K_ID             AS K_ID,
KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
KDI.RANG             AS KONTEXT_RANG
FROM
FALL                 FAL,
DIAGNOSE             DIA,
DIAGNOSE_WERT        DIW,
KONTEXT_DIAGNOSE     KDI,
KONTEXT              KON
WHERE
DIA.D_ID = DIW.D_ID
AND
DIA.D_ID = KDI.D_ID
AND
KDI.K_ID = KON.K_ID
-----
-- Auflösung eines definierten oder eben
-- nicht definierten Seitenbezugs:
-- 1. Fall:
-- es wurde kein Seitenbezug definiert,
-- daher kann diese Eigenschaft
-- vernachlässigt werden
-- 2. Fall:
-- Seitenlokalisierung ist definiert UND
-- muss dann mit der des Fall-bezogenen
-- ICD-Codes übereinstimmen
-----
AND
(
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD  DWI,
FALL_ICD_INKL_OP   FIC
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
FIC.ICD_ID = DWI.ICD_ID
AND
DWI.DWI_ID NOT IN
(
SELECT
```

```

66 DWL.DWI_ID
67 FROM
68 DIAGNOSE_WERT_ICD_LOK DWL
69 WHERE
70 DWL.DWI_ID = DWI.DWI_ID
71 )
72 )
73 OR
74 DIW.DW_ID IN
75 (
76 SELECT
77 DWI.DW_ID
78 FROM
79 DIAGNOSE_WERT_ICD DWI,
80 DIAGNOSE_WERT_ICD_LOK DWL,
81 FALL_ICD_INKL_OP FIC
82 WHERE
83 DWI.DW_ID = DIW.DW_ID
84 AND
85 FIC.FALL_ID = FAL.FALL_ID
86 AND
87 FIC.ICD_ID = DWI.ICD_ID
88 AND
89 DWI.DWI_ID = DWL.DWI_ID
90 AND
91 FIC.ICDLOK_ID = DWL.ICDLOK_ID
92 )
93 )
94 -- -----
95 -- Ende: Auflösung des ggf. zu
96 -- berücksichtigenden Seitenbezugs
97 -- -----
98 -- -----
99 -- Kategorie-Auswertung:
100 -- Berücksichtigung von gemäß der Kontext-
101 -- Phasen-Zuordnung ein- und
102 -- auszuschließenden ICD-Kategorien
103 -- -----
104 AND
105 (
106 DIW.DW_ID IN
107 (
108 SELECT
109 DWI.DW_ID
110 FROM
111 DIAGNOSE_WERT_ICD DWI,
112 FALL_ICD_INKL_OP FIC,
113 PHASE_ICD_KATEGORIE PIK,
114 KONTEXT_DIAGNOSE_PHASE KDP
115 WHERE
116 DWI.DW_ID = DIW.DW_ID
117 AND
118 FIC.FALL_ID = FAL.FALL_ID
119 AND
120 DWI.ICD_ID = FIC.ICD_ID
121 AND
122 PIK.ICDKAT_ID = FIC.ICDKAT_ID
123 AND
124 KDP.KD_ID = KDI.KD_ID
125 AND
126 PIK.PH_ID = KDP.PH_ID
127 AND
128 PIK.PHZT_ID = 1 -- EINSCHLUSS!
129 )
130 AND
131 DIW.DW_ID NOT IN
132 (
133 SELECT

```

```

DWI.DW_ID 134
FROM 135
DIAGNOSE_WERT_ICD DWI, 136
FALL_ICD_INKL_OP FIC, 137
PHASE_ICD_KATEGORIE PIK, 138
KONTEXT_DIAGNOSE_PHASE KDP 139
WHERE 140
DWI.DW_ID = DIW.DW_ID 141
AND 142
FIC.FALL_ID = FAL.FALL_ID 143
AND 144
DWI.ICD_ID = FIC.ICD_ID 145
AND 146
PIK.ICDKAT_ID = FIC.ICDKAT_ID 147
AND 148
KDP.KD_ID = KDI.KD_ID 149
AND 150
PIK.PH_ID = KDP.PH_ID 151
AND 152
PIK.PHZT_ID = 2 -- AUSSCHLUSS! 153
) 154
) 155
-- ----- 156
-- Ende: Kategorie-Auswertung 157
-- ----- 158
-- ----- 159
-- Auswertung Zeitpunkt: 160
-- Berücksichtigung von chronologischen 161
-- Gesichtspunkten (Zeitpunkt im Beh.- 162
-- Verlauf, Zuordnung zu Erst- und ggf. 163
-- Folgeeingriffen etc.) in Abhängigkeit 164
-- von der definierten Phase 165
-- ----- 166
AND 167
( 168
-- ----- 169
-- Auswertung Zeitpunkt - 170
-- Phasen: Aufnahme 171
--      Behandlung 172
--      intraop. allgemein 173
--      Behandlung (neu) 174
-- Keine zeitliche Einschränkung, da durch 175
-- die ICD-Kategorie bereits eine 176
-- eindeutige Zuordnung möglich ist 177
-- (bei Beachtung der Ein- und Ausschlüsse!) 178
-- ----- 179
DIW.DW_ID IN 180
( 181
SELECT 182
DWI.DW_ID 183
FROM 184
DIAGNOSE_WERT_ICD DWI, 185
FALL_ICD_INKL_OP FIC, 186
KONTEXT_DIAGNOSE_PHASE KDP 187
WHERE 188
DWI.DW_ID = DIW.DW_ID 189
AND 190
FIC.FALL_ID = FAL.FALL_ID 191
AND 192
DWI.ICD_ID = FIC.ICD_ID 193
AND 194
KDP.KD_ID = KDI.KD_ID 195
AND 196
KDP.PH_ID IN (1, 2, 5, 6) 197
) 198
-- ----- 199
-- Auswertung Zeitpunkt - 200
-- Phase: postoperativ 201

```

```

202 -- Zeitliche Einschränkung: Zeitpunkt muss
203 -- mit dem Beginn des ersten Eingriffs
204 -- übereinstimmen oder hinter diesem liegen
205 -----
206 OR
207 DIW.DW_ID IN
208 (
209 SELECT
210 DWI.DW_ID
211 FROM
212 DIAGNOSE_WERT_ICD      DWI,
213 FALL_ICD_INKL_OP      FIC,
214 KONTEXT_DIAGNOSE_PHASE KDP
215 WHERE
216 DWI.DW_ID = DIW.DW_ID
217 AND
218 FIC.FALL_ID = FAL.FALL_ID
219 AND
220 DWI.ICD_ID = FIC.ICD_ID
221 AND
222 KDP.KD_ID = KDI.KD_ID
223 AND
224 KDP.PH_ID IN (7)
225 AND
226 FIC.DATUM >=
227 (
228 SELECT
229 MIN (FOP.BEGINN)
230 FROM
231 FALL_OP FOP
232 WHERE
233 FOP.FALL_ID = FAL.FALL_ID
234 )
235 )
236 -----
237 -- Auswertung Zeitpunkt -
238 -- Phase: intraop. Haupteingriff
239 -- Zeitliche Einschränkung: Zeitpunkt muss
240 -- mit dem Beginn des ersten Eingriffs
241 -- übereinstimmen
242 -----
243 OR
244 DIW.DW_ID IN
245 (
246 SELECT
247 DWI.DW_ID
248 FROM
249 DIAGNOSE_WERT_ICD      DWI,
250 FALL_ICD_INKL_OP      FIC,
251 KONTEXT_DIAGNOSE_PHASE KDP
252 WHERE
253 DWI.DW_ID = DIW.DW_ID
254 AND
255 FIC.FALL_ID = FAL.FALL_ID
256 AND
257 DWI.ICD_ID = FIC.ICD_ID
258 AND
259 KDP.KD_ID = KDI.KD_ID
260 AND
261 KDP.PH_ID IN (3)
262 AND
263 FIC.DATUM =
264 (
265 SELECT
266 MIN (FOP.BEGINN)
267 FROM
268 FALL_OP FOP
269 WHERE

```

```

FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Folgeeingriff(e)
-- Zeitliche Einschränkung: Zeitpunkt muss
-- hinter dem Beginn des ersten Eingriffs
-- liegen
-----
OR
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD          DWI,
FALL_ICD_INKL_OP          FIC,
KONTEXT_DIAGNOSE_PHASE  KDP
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
KDP.KD_ID = KDI.KD_ID
AND
KDP.PH_ID IN (4)
AND
FIC.DATUM >
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP  FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Ende: Auswertung Zeitpunkt
-----
);

```

270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313

### A4.3. View [Fall\_Kontext\_Diag\_Wert\_Non\_ICD]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAG_WERT_NON_ICD AS
2 SELECT
3 FAL.FALL_ID          AS FALL_ID,
4 FAL.CODE            AS FALL_CODE,
5 FAL.NACHNAME        AS NACHNAME,
6 FAL.VORNAME         AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
9 FAL.ABGANG_FA       AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH       AS ABGANG_KH,
12 DIA.D_ID            AS D_ID,
13 DIA.BEZEICHNUNG     AS DIAGNOSE_BEZEICHNUNG,
14 NULL                AS DW_ID,
15 DIA.NORMAL_TEXT     AS DIAGNOSE_WERT_BEZEICHNUNG,
16 DIA.NORMAL_WERT     AS DIAGNOSE_WERT,
17 9999                AS DIAGNOSE_WERT_RANG,
18 1                   AS EJN_ID_AUSWERTUNG,
19 KDI.KD_ID           AS KD_ID,
20 KON.K_ID            AS K_ID,
21 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
22 KDI.RANG            AS KONTEXT_RANG
23 FROM
24 FALL                FAL,
25 DIAGNOSE            DIA,
26 KONTEXT_DIAGNOSE   KDI,
27 KONTEXT             KON
28 WHERE
29 DIA.D_ID = KDI.D_ID
30 AND
31 KDI.K_ID = KON.K_ID
32 AND
33 -- Diagnosen müssen sich prinzipiell aus ICD-Codes ableiten lassen!
34 DIA.D_ID IN
35 (
36 SELECT
37 DIW.D_ID
38 FROM
39 DIAGNOSE_WERT       DIW,
40 DIAGNOSE_WERT_ICD   DWI
41 WHERE
42 DIW.D_ID = DIA.D_ID
43 AND
44 DIW.DW_ID = DWI.DW_ID
45 )
46 AND
47 DIA.D_ID NOT IN
48 (
49 SELECT
50 FDI.D_ID
51 FROM
52 FALL_KONTEXT_DIAG_WERT_ICD   FDI
53 WHERE
54 FDI.D_ID = DIA.D_ID
55 AND
56 FDI.FALL_ID = FAL.FALL_ID
57 );
```



#### **A4.4. View [Diagnose\_Wert\_Komb\_Elem\_Count]**

```
CREATE OR REPLACE VIEW DIAGNOSE_WERT_KOMB_ELEM_COUNT AS
SELECT
DKD.DWK_ID          AS DWK_ID,
COUNT (DKD.DWKDW_ID) AS ZAHL_ELEMENTE
FROM
DIAG_WERT_KOMB_DIAG_WERT  DKD
GROUP BY
DKD.DWK_ID;
```

1  
2  
3  
4  
5  
6  
7  
8

## A4.5. View [Fall\_Kontext\_Diag\_Wert\_Komb]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAG_WERT_KOMB AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
9 FAL.ABGANG_FA       AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH       AS ABGANG_KH,
12 DIA.D_ID            AS D_ID,
13 DIA.BEZEICHNUNG     AS DIAGNOSE_BEZEICHNUNG,
14 DWK.DWK_ID          AS DWK_ID,
15 DWK.BEZEICHNUNG     AS DIAGNOSE_WERT_BEZEICHNUNG,
16 DWK.WERT            AS DIAGNOSE_WERT,
17 DWK.RANG            AS DIAGNOSE_WERT_RANG,
18 DWK.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
19 KDI.KD_ID           AS KD_ID,
20 KON.K_ID            AS K_ID,
21 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
22 KDI.RANG            AS KONTEXT_RANG
23 FROM
24 FALL                 FAL,
25 DIAGNOSE             DIA,
26 DIAGNOSE_WERT_KOMBINATION DWK,
27 DIAGNOSE_WERT_KOMB_ELEM_COUNT DKE,
28 KONTEXT_DIAGNOSE    KDI,
29 KONTEXT              KON
30 WHERE
31 DIA.D_ID = DWK.D_ID
32 AND
33 DWK.DWK_ID = DKE.DWK_ID
34 AND
35 DIA.D_ID = KDI.D_ID
36 AND
37 KDI.K_ID = KON.K_ID
38 AND
39 (
40 (
41 -- Kombinationstyp: alle Diagnosen sind obligat!
42 DWK.DWKT_ID = 1
43 AND
44 DKE.ZAHL_ELEMENTE =
45 (
46 SELECT
47 COUNT (DKD.DWKDW_ID)
48 FROM
49 FALL_KONTEXT_DIAG_WERT_ICD FDI,
50 DIAG_WERT_KOMB_DIAG_WERT DKD
51 WHERE
52 FDI.FALL_ID = FAL.FALL_ID
53 AND
54 FDI.DW_ID = DKD.DW_ID
55 AND
56 DKD.DWK_ID = DWK.DWK_ID
57 )
58 )
59 OR
60 (
61 -- Kombinationstyp: mindestens 2 Diagnosen sind obligat!
62 DWK.DWKT_ID = 2
63 AND
64 1 <
65 (
```

SELECT	66
COUNT (DKD.DWKDW_ID)	67
FROM	68
FALL_KONTEXT_DIAG_WERT_ICD FDI,	69
DIAG_WERT_KOMB_DIAG_WERT DKD	70
WHERE	71
FDI.FALL_ID = FAL.FALL_ID	72
AND	73
FDI.DW_ID = DKD.DW_ID	74
AND	75
DKD.DWK_ID = DWK.DWK_ID	76
)	77
)	78
);	79

## A4.6. View [Fall\_Kontext\_Diagnose\_Rang]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAGNOSE_RANG AS
2 SELECT
3 FALL_ID                AS FALL_ID,
4 FALL_CODE              AS FALL_CODE,
5 NACHNAME              AS NACHNAME,
6 VORNAME               AS VORNAME,
7 GEBURTSDATUM         AS GEBURTSDATUM,
8 AUFNAHME_FA          AS AUFNAHME_FA,
9 ABGANG_FA             AS ABGANG_FA,
10 AUFNAHME_KH          AS AUFNAHME_KH,
11 ABGANG_KH            AS ABGANG_KH,
12 D_ID                 AS D_ID,
13 DW_ID                AS DW_ID,
14 NULL                 AS DWK_ID,
15 DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
16 DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
17 DIAGNOSE_WERT        AS DIAGNOSE_WERT,
18 DIAGNOSE_WERT_RANG   AS DIAGNOSE_WERT_RANG,
19 1                     AS EJN_ID_AUSWERTUNG,
20 KD_ID                AS KD_ID,
21 K_ID                 AS K_ID,
22 KONTEXT_BEZEICHNUNG  AS KONTEXT_BEZEICHNUNG,
23 KONTEXT_RANG         AS KONTEXT_RANG
24 FROM
25 FALL_KONTEXT_DIAG_WERT_NON_ICD
26 UNION
27 SELECT
28 FALL_ID,
29 FALL_CODE,
30 NACHNAME,
31 VORNAME,
32 GEBURTSDATUM,
33 AUFNAHME_FA,
34 ABGANG_FA,
35 AUFNAHME_KH,
36 ABGANG_KH,
37 D_ID,
38 DW_ID,
39 NULL,
40 DIAGNOSE_BEZEICHNUNG,
41 DIAGNOSE_WERT_BEZEICHNUNG,
42 DIAGNOSE_WERT,
43 DIAGNOSE_WERT_RANG,
44 EJN_ID_AUSWERTUNG,
45 KD_ID,
46 K_ID,
47 KONTEXT_BEZEICHNUNG,
48 KONTEXT_RANG
49 FROM
50 FALL_KONTEXT_DIAG_WERT_ICD
51 UNION
52 SELECT
53 FALL_ID,
54 FALL_CODE,
55 NACHNAME,
56 VORNAME,
57 GEBURTSDATUM,
58 AUFNAHME_FA,
59 ABGANG_FA,
60 AUFNAHME_KH,
61 ABGANG_KH,
62 D_ID,
63 NULL,
64 DWK_ID,
65 DIAGNOSE_BEZEICHNUNG,
```

DIAGNOSE_WERT_BEZEICHNUNG,	66
DIAGNOSE_WERT,	67
DIAGNOSE_WERT_RANG,	68
EJN_ID_AUSWERTUNG,	69
KD_ID,	70
K_ID,	71
KONTEXT_BEZEICHNUNG,	72
KONTEXT_RANG	73
FROM	74
FALL_KONTEXT_DIAG_WERT_KOMB;	75

## A4.7. View [Fall\_Kontext\_Diagnose]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAGNOSE AS
2 SELECT
3 T1.FALL_ID AS FALL_ID,
4 T1.FALL_CODE AS FALL_CODE,
5 T1.NACHNAME AS NACHNAME,
6 T1.VORNAME AS VORNAME,
7 T1.GEBURTSDATUM AS GEBURTSDATUM,
8 T1.AUFNAHME_FA AS AUFNAHME_FA,
9 T1.ABGANG_FA AS ABGANG_FA,
10 T1.AUFNAHME_KH AS AUFNAHME_KH,
11 T1.ABGANG_KH AS ABGANG_KH,
12 T1.D_ID AS D_ID,
13 T1.DW_ID AS DW_ID,
14 T1.DWK_ID AS DWK_ID,
15 T1.DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
16 T1.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
17 T1.DIAGNOSE_WERT AS DIAGNOSE_WERT,
18 T1.DIAGNOSE_WERT_RANG AS DIAGNOSE_WERT_RANG,
19 T1.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
20 T1.KD_ID AS KD_ID,
21 T1.K_ID AS K_ID,
22 T1.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
23 T1.KONTEXT_RANG AS KONTEXT_RANG
24 FROM
25 FALL_KONTEXT_DIAGNOSE_RANG T1
26 WHERE
27 T1.EJN_ID_AUSWERTUNG = 1
28 AND
29 (
30 T1.FALL_ID,
31 T1.D_ID,
32 T1.DIAGNOSE_WERT_RANG
33 )
34 IN
35 (
36 SELECT
37 T2.FALL_ID,
38 T2.D_ID,
39 MIN (T2.DIAGNOSE_WERT_RANG)
40 FROM
41 FALL_KONTEXT_DIAGNOSE_RANG T2
42 WHERE
43 T2.FALL_ID = T1.FALL_ID
44 AND
45 T2.D_ID = T1.D_ID
46 GROUP BY
47 T2.FALL_ID,
48 T2.D_ID
49 )
50 ;
```

## A4.8. View [Fall\_OPS\_Inkl\_OP]

```
CREATE OR REPLACE VIEW FALL_OPS_INKL_OP AS
SELECT
FAL.FALL_ID AS FALL_ID,
FAL.CODE AS FALL_CODE,
FAL.NACHNAME AS NACHNAME,
FAL.VORNAME AS VORNAME,
FAL.GEBURTSDATUM AS GEBURTSDATUM,
FAL.AUFNAHME_FA AS AUFNAHME_FA,
FAL.ABGANG_FA AS ABGANG_FA,
FAL.AUFNAHME_KH AS AUFNAHME_KH,
FAL.ABGANG_KH AS ABGANG_KH,
FAL.G_ID AS G_ID,
FAL.BA_ID AS BA_ID,
BEA.NAME || ', ' || BEA.VORNAME AS BEARBEITER,
FAL.KIS_ID AS KIS_ID,
KIS.BEZEICHNUNG AS KIS_BEZEICHNUNG,
FAL.FA_ID AS FA_ID,
FAB.BEZEICHNUNG AS FACHABTEILUNG_BEZEICHNUNG,
NULL AS FALLOP_ID,
NULL AS OP_BEGINN,
NULL AS OP_ENDE,
FAO.FALLOPS_ID AS FALLOPS_ID,
NULL AS FOPOPS_ID,
FAO.OPS_ID AS OPS_ID,
OPS.CODE AS OPS_CODE,
OPS.KURZTEXT AS OPS_KURZTEXT,
OVE.OPSVER_ID AS OPSVER_ID,
OVE.BEZEICHNUNG AS OPS_VERSION_BEZEICHNUNG,
FAO.OPSKAT_ID AS OPSKAT_ID,
OKA.BEZEICHNUNG AS OPS_KATEGORIE_BEZEICHNUNG,
FAO.OPSLOK_ID AS OPSLOK_ID,
OLO.BEZEICHNUNG AS OPS_LOKALISATION_BEZEICHNUNG,
FAO.DATUM AS DATUM
FROM
FALL
FALL_OPS301
OPS301
OPS301_VERSION
OPS301_KATEGORIE
OPS301_LOKALISATION
KIS
FACHABTEILUNG
BEARBEITER
BEA
WHERE
FAL.FALL_ID = FAO.FALL_ID
AND
FAO.OPS_ID = OPS.OPS_ID
AND
FAO.OPSKAT_ID = OKA.OPSKAT_ID
AND
FAO.OPSLOK_ID = OLO.OPSLOK_ID
AND
FAL.KIS_ID = KIS.KIS_ID
AND
FAL.FA_ID = FAB.FA_ID
AND
FAL.BA_ID = BEA.BA_ID
AND
OPS.OPSVER_ID = OVE.OPSVER_ID
UNION
SELECT
FAL.FALL_ID,
FAL.CODE,
FAL.NACHNAME,
FAL.VORNAME,
```

```

66 FAL.GEBURTSDATUM,
67 FAL.AUFNAHME_FA,
68 FAL.ABGANG_FA,
69 FAL.AUFNAHME_KH,
70 FAL.ABGANG_KH,
71 FAL.G_ID,
72 FAL.BA_ID,
73 BEA.NAME || ', ' || BEA.VORNAME,
74 FAL.KIS_ID,
75 KIS.BEZEICHNUNG,
76 FAL.FA_ID,
77 FAB.BEZEICHNUNG,
78 FOP.FALLOP_ID,
79 FOP.BEGINN,
80 FOP.ENDE,
81 NULL,
82 FOO.FOPOPS_ID,
83 FOO.OPS_ID,
84 OPS.CODE,
85 OPS.KURZTEXT,
86 OVE.OPSVER_ID,
87 OVE.BEZEICHNUNG,
88 FOO.OPSKAT_ID,
89 OKA.BEZEICHNUNG,
90 FOO.OPSLOK_ID,
91 OLO.BEZEICHNUNG,
92 FOP.BEGINN
93 FROM
94 FALL                FAL,
95 FALL_OP             FOP,
96 FALL_OP_OPS301     FOO,
97 OPS301              OPS,
98 OPS301_VERSION     OVE,
99 OPS301_KATEGORIE   OKA,
100 OPS301_LOKALISATION OLO,
101 KIS                 KIS,
102 FACHABTEILUNG      FAB,
103 BEARBEITER         BEA
104 WHERE
105 FAL.FALL_ID = FOP.FALL_ID
106 AND
107 FOP.FALLOP_ID = FOO.FALLOP_ID
108 AND
109 FOO.OPS_ID = OPS.OPS_ID
110 AND
111 FOO.OPSKAT_ID = OKA.OPSKAT_ID
112 AND
113 FOO.OPSLOK_ID = OLO.OPSLOK_ID
114 AND
115 FAL.KIS_ID = KIS.KIS_ID
116 AND
117 FAL.FA_ID = FAB.FA_ID
118 AND
119 FAL.BA_ID = BEA.BA_ID
120 AND
121 OPS.OPSVER_ID = OVE.OPSVER_ID;

```



## A4.9. View [Fall\_Kontext\_Proz\_Wert\_OPS]

```
CREATE OR REPLACE VIEW FALL_KONTEXT_PROZ_WERT_OPS AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FAL.CODE            AS FALL_CODE,
FAL.NACHNAME        AS NACHNAME,
FAL.VORNAME         AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA     AS AUFNAHME_FA,
FAL.ABGANG_FA       AS ABGANG_FA,
FAL.AUFNAHME_KH     AS AUFNAHME_KH,
FAL.ABGANG_KH       AS ABGANG_KH,
PRO.P_ID            AS P_ID,
PRO.BEZEICHNUNG     AS PROZEDUR_BEZEICHNUNG,
PRW.PW_ID           AS PW_ID,
PRW.BEZEICHNUNG     AS PROZEDUR_WERT_BEZEICHNUNG,
PRW.WERT            AS PROZEDUR_WERT,
PRW.RANG            AS PROZEDUR_WERT_RANG,
PRW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
KPR.KP_ID           AS KP_ID,
KON.K_ID            AS K_ID,
KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
KPR.RANG            AS KONTEXT_RANG
FROM
FALL                FAL,
PROZEDUR            PRO,
PROZEDUR_WERT      PRW,
KONTEXT_PROZEDUR  KPR,
KONTEXT            KON
WHERE
PRO.P_ID = PRW.P_ID
AND
PRO.P_ID = KPR.P_ID
AND
KPR.K_ID = KON.K_ID
-- -----
-- Auflösung eines definierten oder eben
-- nicht definierten Seitenbezugs:
-- 1. Fall:
-- es wurde kein Seitenbezug definiert,
-- daher kann diese Eigenschaft
-- vernachlässigt werden
-- 2. Fall:
-- Seitenlokalisierung ist definiert UND
-- muss dann mit der des Fall-bezogenen
-- ICD-Codes übereinstimmen
-- -----
AND
(
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS  PWO,
FALL_OPS_INKL_OP   FIN
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
FIN.OPS_ID = PWO.OPS_ID
AND
PWO.PWO_ID NOT IN
(
SELECT
```

```

66  PWL.PWO_ID
67  FROM
68  PROZEDUR_WERT_OPS_LOK  PWL
69  WHERE
70  PWL.PWO_ID = PWO.PWO_ID
71  )
72  )
73  OR
74  PRW.PW_ID IN
75  (
76  SELECT
77  PWO.PW_ID
78  FROM
79  PROZEDUR_WERT_OPS      PWO,
80  PROZEDUR_WERT_OPS_LOK PWL,
81  FALL_OPS_INKL_OP      FIN
82  WHERE
83  PWO.PW_ID = PRW.PW_ID
84  AND
85  FIN.FALL_ID = FAL.FALL_ID
86  AND
87  FIN.OPS_ID = PWO.OPS_ID
88  AND
89  PWO.PWO_ID = PWL.PWO_ID
90  AND
91  FIN.OPSLOK_ID = PWL.OPSLOK_ID
92  )
93  )
94  -- -----
95  -- Ende: Auflösung des ggf. zu
96  -- berücksichtigenden Seitenbezugs
97  -- -----
98  -- -----
99  -- Kategorie-Auswertung:
100 -- Berücksichtigung von gemäß der Kontext-
101 -- Phasen-Zuordnung ein- und
102 -- auszuschließenden ICD-Kategorien
103 -- -----
104 AND
105 (
106 PRW.PW_ID IN
107 (
108 SELECT
109 PWO.PW_ID
110 FROM
111 PROZEDUR_WERT_OPS      PWO,
112 FALL_OPS_INKL_OP      FIN,
113 PHASE_OPS301_KATEGORIE POK,
114 KONTEXT_PROZEDUR_PHASE KPP
115 WHERE
116 PWO.PW_ID = PRW.PW_ID
117 AND
118 FIN.FALL_ID = FAL.FALL_ID
119 AND
120 PWO.OPS_ID = FIN.OPS_ID
121 AND
122 POK.OPSKAT_ID = FIN.OPSKAT_ID
123 AND
124 KPP.KP_ID = KPR.KP_ID
125 AND
126 POK.PH_ID = KPP.PH_ID
127 AND
128 POK.PHZT_ID = 1  -- EINSCHLUSS!
129 )
130 AND
131 PRW.PW_ID NOT IN
132 (
133 SELECT

```

```

PWO.PW_ID 134
FROM 135
PROZEDUR_WERT_OPS PWO, 136
FALL_OPS_INKL_OP FIN, 137
PHASE_OPS301_KATEGORIE POK, 138
KONTEXT_PROZEDUR_PHASE KPP 139
WHERE 140
PWO.PW_ID = PRW.PW_ID 141
AND 142
FIN.FALL_ID = FAL.FALL_ID 143
AND 144
PWO.OPS_ID = FIN.OPS_ID 145
AND 146
POK.OPSKAT_ID = FIN.OPSKAT_ID 147
AND 148
KPP.KP_ID = KPR.KP_ID 149
AND 150
POK.PH_ID = KPP.PH_ID 151
AND 152
POK.PHZT_ID = 2 -- AUSSCHLUSS! 153
) 154
) 155
-- ----- 156
-- Ende: Kategorie-Auswertung 157
-- ----- 158
-- ----- 159
-- Auswertung Zeitpunkt: 160
-- Berücksichtigung von chronologischen 161
-- Gesichtspunkten (Zeitpunkt im Beh.- 162
-- Verlauf, Zuordnung zu Erst- und ggf. 163
-- Folgeeingriffen etc.) in Abhängigkeit 164
-- von der definierten Phase 165
-- ----- 166
AND 167
( 168
-- ----- 169
-- Auswertung Zeitpunkt - 170
-- Phasen: Aufnahme 171
--      Behandlung 172
--      intraop. allgemein 173
--      Behandlung (neu) 174
-- Keine zeitliche Einschränkung, da durch 175
-- die ICD-Kategorie bereits eine 176
-- eindeutige Zuordnung möglich ist 177
-- (bei Beachtung der Ein- und Ausschlüsse!) 178
-- ----- 179
PRW.PW_ID IN 180
( 181
SELECT 182
PWO.PW_ID 183
FROM 184
PROZEDUR_WERT_OPS PWO, 185
FALL_OPS_INKL_OP FIN, 186
KONTEXT_PROZEDUR_PHASE KPP 187
WHERE 188
PWO.PW_ID = PRW.PW_ID 189
AND 190
FIN.FALL_ID = FAL.FALL_ID 191
AND 192
PWO.OPS_ID = FIN.OPS_ID 193
AND 194
KPP.KP_ID = KPR.KP_ID 195
AND 196
KPP.PH_ID IN (1, 2, 5, 6) 197
) 198
-- ----- 199
-- Auswertung Zeitpunkt - 200
-- Phase: postoperativ 201

```

```

202 -- Zeitliche Einschränkung: Zeitpunkt muss
203 -- mit dem Beginn des ersten Eingriffs
204 -- übereinstimmen oder hinter diesem liegen
205 -----
206 OR
207 PRW.PW_ID IN
208 (
209 SELECT
210 PWO.PW_ID
211 FROM
212 PROZEDUR_WERT_OPS      PWO,
213 FALL_OPS_INKL_OP      FIN,
214 KONTEXT_PROZEDUR_PHASE KPP
215 WHERE
216 PWO.PW_ID = PRW.PW_ID
217 AND
218 FIN.FALL_ID = FAL.FALL_ID
219 AND
220 PWO.OPS_ID = FIN.OPS_ID
221 AND
222 KPP.KP_ID = KPR.KP_ID
223 AND
224 KPP.PH_ID IN (7)
225 AND
226 FIN.DATUM >=
227 (
228 SELECT
229 MIN (FOP.BEGINN)
230 FROM
231 FALL_OP  FOP
232 WHERE
233 FOP.FALL_ID = FAL.FALL_ID
234 )
235 )
236 -----
237 -- Auswertung Zeitpunkt -
238 -- Phase: intraop. Haupteingriff
239 -- Zeitliche Einschränkung: Zeitpunkt muss
240 -- mit dem Beginn des ersten Eingriffs
241 -- übereinstimmen
242 -----
243 OR
244 PRW.PW_ID IN
245 (
246 SELECT
247 PWO.PW_ID
248 FROM
249 PROZEDUR_WERT_OPS      PWO,
250 FALL_OPS_INKL_OP      FIN,
251 KONTEXT_PROZEDUR_PHASE KPP
252 WHERE
253 PWO.PW_ID = PRW.PW_ID
254 AND
255 FIN.FALL_ID = FAL.FALL_ID
256 AND
257 PWO.OPS_ID = FIN.OPS_ID
258 AND
259 KPP.KP_ID = KPR.KP_ID
260 AND
261 KPP.PH_ID IN (3)
262 AND
263 FIN.DATUM =
264 (
265 SELECT
266 MIN (FOP.BEGINN)
267 FROM
268 FALL_OP  FOP
269 WHERE

```

```

FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Folgeeingriff(e)
-- Zeitliche Einschränkung: Zeitpunkt muss
-- hinter dem Beginn des ersten Eingriffs
-- liegen
-----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_INKL_OP      FIN,
KONTEXT_PROZEDUR_PHASE KPP
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
PWO.OPS_ID = FIN.OPS_ID
AND
KPP.KP_ID = KPR.KP_ID
AND
KPP.PH_ID IN (4)
AND
FIN.DATUM >
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Ende: Auswertung Zeitpunkt
-----
);

```

270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313

## A4.10. View [Fall\_Kontext\_Proz\_Wert\_Non\_OPS]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZ_WERT_NON_OPS AS
2 SELECT
3 FAL.FALL_ID          AS FALL_ID,
4 FAL.CODE            AS FALL_CODE,
5 FAL.NACHNAME        AS NACHNAME,
6 FAL.VORNAME         AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
9 FAL.ABGANG_FA       AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH       AS ABGANG_KH,
12 PRO.P_ID            AS P_ID,
13 PRO.BEZEICHNUNG     AS PROZEDUR_BEZEICHNUNG,
14 NULL                AS PW_ID,
15 PRO.NORMAL_TEXT     AS PROZEDUR_WERT_BEZEICHNUNG,
16 PRO.NORMAL_WERT     AS PROZEDUR_WERT,
17 9999                AS PROZEDUR_WERT_RANG,
18 1                   AS EJN_ID_AUSWERTUNG,
19 KPR.KP_ID           AS KP_ID,
20 KON.K_ID            AS K_ID,
21 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
22 KPR.RANG            AS KONTEXT_RANG
23 FROM
24 FALL                FAL,
25 PROZEDUR            PRO,
26 KONTEXT_PROZEDUR   KPR,
27 KONTEXT             KON
28 WHERE
29 PRO.P_ID = KPR.P_ID
30 AND
31 KPR.K_ID = KON.K_ID
32 AND
33 -- Prozeduren müssen sich prinzipiell aus OPS301-Codes ableiten lassen!
34 PRO.P_ID IN
35 (
36 SELECT
37 PRW.P_ID
38 FROM
39 PROZEDUR_WERT       PRW,
40 PROZEDUR_WERT_OPS   PWO
41 WHERE
42 PRW.P_ID = PRO.P_ID
43 AND
44 PRW.PW_ID = PWO.PW_ID
45 )
46 AND
47 PRO.P_ID NOT IN
48 (
49 SELECT
50 FPR.P_ID
51 FROM
52 FALL_KONTEXT_PROZ_WERT_OPS   FPR
53 WHERE
54 FPR.P_ID = PRO.P_ID
55 AND
56 FPR.FALL_ID = FAL.FALL_ID
57 );
```

#### **A4.11. View [Prozedur\_Wert\_Komb\_Elem\_Count]**

```
CREATE OR REPLACE VIEW PROZEDUR_WERT_KOMB_ELEM_COUNT AS
SELECT
PKP.PWK_ID          AS PWK_ID,
COUNT (PKP.PWKPW_ID) AS ZAHL_ELEMENTE
FROM
PROZ_WERT_KOMB_PROZ_WERT PKP
GROUP BY
PKP.PWK_ID;
```

1  
2  
3  
4  
5  
6  
7  
8

## A4.12. View [Fall\_Kontext\_Proz\_Wert\_Komb]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZ_WERT_KOMB AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA      AS AUFNAHME_FA,
9 FAL.ABGANG_FA        AS ABGANG_FA,
10 FAL.AUFNAHME_KH      AS AUFNAHME_KH,
11 FAL.ABGANG_KH        AS ABGANG_KH,
12 PRO.P_ID             AS P_ID,
13 PRO.BEZEICHNUNG      AS PROZEDUR_BEZEICHNUNG,
14 PWK.PWK_ID           AS PWK_ID,
15 PWK.BEZEICHNUNG      AS PROZEDUR_WERT_BEZEICHNUNG,
16 PWK.WERT             AS PROZEDUR_WERT,
17 PWK.RANG             AS PROZEDUR_WERT_RANG,
18 PWK.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
19 KPR.KP_ID            AS KP_ID,
20 KON.K_ID             AS K_ID,
21 KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
22 KPR.RANG             AS KONTEXT_RANG
23 FROM
24 FALL                 FAL,
25 PROZEDUR             PRO,
26 PROZEDUR_WERT_KOMBINATION PWK,
27 PROZEDUR_WERT_KOMB_ELEM_COUNT PKE,
28 KONTEXT_PROZEDUR     KPR,
29 KONTEXT              KON
30 WHERE
31 PRO.P_ID = PWK.P_ID
32 AND
33 PWK.PWK_ID = PKE.PWK_ID
34 AND
35 PRO.P_ID = KPR.P_ID
36 AND
37 KPR.K_ID = KON.K_ID
38 AND
39 (
40 (
41 -- Kombinationstyp: alle Prozeduren sind obligat!
42 PWK.PWKT_ID = 1
43 AND
44 PKE.ZAHL_ELEMENTE =
45 (
46 SELECT
47 COUNT (PKP.PWKPW_ID)
48 FROM
49 FALL_KONTEXT_PROZ_WERT_OPS FDO,
50 PROZ_WERT_KOMB_PROZ_WERT PKP
51 WHERE
52 FDO.FALL_ID = FAL.FALL_ID
53 AND
54 FDO.PW_ID = PKP.PW_ID
55 AND
56 PKP.PWK_ID = PWK.PWK_ID
57 )
58 )
59 OR
60 (
61 -- Kombinationstyp: mindestens 2 Prozeduren sind obligat!
62 PWK.PWKT_ID = 2
63 AND
64 1 <
65 (
```



SELECT	66
COUNT (PKP.PWKPW_ID)	67
FROM	68
FALL_KONTEXT_PROZ_WERT_OPS FDO,	69
PROZ_WERT_KOMB_PROZ_WERT PKP	70
WHERE	71
FDO.FALL_ID = FAL.FALL_ID	72
AND	73
FDO.PW_ID = PKP.PW_ID	74
AND	75
PKP.PWK_ID = PWK.PWK_ID	76
)	77
)	78
);	79

### A4.13. View [Fall\_Kontext\_Prozedur\_Rang]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZEDUR_RANG AS
2 SELECT
3 FALL_ID                AS FALL_ID,
4 FALL_CODE              AS FALL_CODE,
5 NACHNAME              AS NACHNAME,
6 VORNAME              AS VORNAME,
7 GEBURTSDATUM        AS GEBURTSDATUM,
8 AUFNAHME_FA         AS AUFNAHME_FA,
9 ABGANG_FA           AS ABGANG_FA,
10 AUFNAHME_KH        AS AUFNAHME_KH,
11 ABGANG_KH          AS ABGANG_KH,
12 P_ID               AS P_ID,
13 PW_ID              AS PW_ID,
14 NULL                AS PWK_ID,
15 PROZEDUR_BEZEICHNUNG AS PROZEDUR_BEZEICHNUNG,
16 PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
17 PROZEDUR_WERT       AS PROZEDUR_WERT,
18 PROZEDUR_WERT_RANG AS PROZEDUR_WERT_RANG,
19 1                   AS EJN_ID_AUSWERTUNG,
20 KP_ID              AS KP_ID,
21 K_ID               AS K_ID,
22 KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
23 KONTEXT_RANG       AS KONTEXT_RANG
24 FROM
25 FALL_KONTEXT_PROZ_WERT_NON_OPS
26 UNION
27 SELECT
28 FALL_ID,
29 FALL_CODE,
30 NACHNAME,
31 VORNAME,
32 GEBURTSDATUM,
33 AUFNAHME_FA,
34 ABGANG_FA,
35 AUFNAHME_KH,
36 ABGANG_KH,
37 P_ID,
38 PW_ID,
39 NULL,
40 PROZEDUR_BEZEICHNUNG,
41 PROZEDUR_WERT_BEZEICHNUNG,
42 PROZEDUR_WERT,
43 PROZEDUR_WERT_RANG,
44 EJN_ID_AUSWERTUNG,
45 KP_ID,
46 K_ID,
47 KONTEXT_BEZEICHNUNG,
48 KONTEXT_RANG
49 FROM
50 FALL_KONTEXT_PROZ_WERT_OPS
51 UNION
52 SELECT
53 FALL_ID,
54 FALL_CODE,
55 NACHNAME,
56 VORNAME,
57 GEBURTSDATUM,
58 AUFNAHME_FA,
59 ABGANG_FA,
60 AUFNAHME_KH,
61 ABGANG_KH,
62 P_ID,
63 NULL,
64 PWK_ID,
65 PROZEDUR_BEZEICHNUNG,
```

PROZEDUR_WERT_BEZEICHNUNG,	66
PROZEDUR_WERT,	67
PROZEDUR_WERT_RANG,	68
EJN_ID_AUSWERTUNG,	69
KP_ID,	70
K_ID,	71
KONTEXT_BEZEICHNUNG,	72
KONTEXT_RANG	73
FROM	74
FALL_KONTEXT_PROZ_WERT_KOMB;	75

#### A4.14. View [Fall\_Kontext\_Prozedur]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZEDUR AS
2 SELECT
3 T1.FALL_ID AS FALL_ID,
4 T1.FALL_CODE AS FALL_CODE,
5 T1.NACHNAME AS NACHNAME,
6 T1.VORNAME AS VORNAME,
7 T1.GEBURTSDATUM AS GEBURTSDATUM,
8 T1.AUFNAHME_FA AS AUFNAHME_FA,
9 T1.ABGANG_FA AS ABGANG_FA,
10 T1.AUFNAHME_KH AS AUFNAHME_KH,
11 T1.ABGANG_KH AS ABGANG_KH,
12 T1.P_ID AS P_ID,
13 T1.PW_ID AS PW_ID,
14 T1.PWK_ID AS PWK_ID,
15 T1.PROZEDUR_BEZEICHNUNG AS PROZEDUR_BEZEICHNUNG,
16 T1.PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
17 T1.PROZEDUR_WERT AS PROZEDUR_WERT,
18 T1.PROZEDUR_WERT_RANG AS PROZEDUR_WERT_RANG,
19 T1.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
20 T1.KP_ID AS KP_ID,
21 T1.K_ID AS K_ID,
22 T1.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
23 T1.KONTEXT_RANG AS KONTEXT_RANG
24 FROM
25 FALL_KONTEXT_PROZEDUR_RANG T1
26 WHERE
27 T1.EJN_ID_AUSWERTUNG = 1
28 AND
29 (
30 T1.FALL_ID,
31 T1.P_ID,
32 T1.PROZEDUR_WERT_RANG
33 )
34 IN
35 (
36 SELECT
37 T2.FALL_ID,
38 T2.P_ID,
39 MIN (T2.PROZEDUR_WERT_RANG)
40 FROM
41 FALL_KONTEXT_PROZEDUR_RANG T2
42 WHERE
43 T2.FALL_ID = T1.FALL_ID
44 AND
45 T2.P_ID = T1.P_ID
46 GROUP BY
47 T2.FALL_ID,
48 T2.P_ID
49 )
50 ;
```

## A4.15. View [Fall\_Kont\_Komp\_Diag\_Wert\_ICD]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_DIAG_WERT_ICD AS
SELECT
FAL.FALL_ID           AS FALL_ID,
FAL.CODE              AS FALL_CODE,
FAL.NACHNAME          AS NACHNAME,
FAL.VORNAME           AS VORNAME,
FAL.GEBURTSDATUM     AS GEBURTSDATUM,
FAL.AUFNAHME_FA       AS AUFNAHME_FA,
FAL.ABGANG_FA         AS ABGANG_FA,
FAL.AUFNAHME_KH       AS AUFNAHME_KH,
FAL.ABGANG_KH         AS ABGANG_KH,
DIA.D_ID              AS D_ID,
DIA.BEZEICHNUNG       AS DIAGNOSE_BEZEICHNUNG,
DIW.DW_ID             AS DW_ID,
DIW.BEZEICHNUNG       AS DIAGNOSE_WERT_BEZEICHNUNG,
DIW.WERT              AS DIAGNOSE_WERT,
KKP.KKP_ID            AS KKP_ID,
KON.K_ID              AS K_ID,
KON.BEZEICHNUNG       AS KONTEXT_BEZEICHNUNG,
KKP.RANG              AS KONTEXT_RANG,
KPL.KP_ID             AS KP_ID,
KPL.BEZEICHNUNG       AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID            AS KPW_ID,
KPW.BEZEICHNUNG       AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.RANG              AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
KPWDW_ID              AS KPWDW_ID
FROM
FALL                  FAL,
DIAGNOSE              DIA,
DIAGNOSE_WERT         DIW,
KONTEXT_KOMPLEX       KKP,
KONTEXT               KON,
KOMPLEX               KPL,
KOMPLEX_WERT          KPW,
KOMPLEX_WERT_D_WERT   KWD
WHERE
KKP.KP_ID = KPL.KP_ID
AND
KKP.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWD.KPW_ID
AND
KWD.DW_ID = DIW.DW_ID
AND
DIW.D_ID = DIA.D_ID
AND
KKP.K_ID = KON.K_ID
-----
-- Auflösung eines definierten oder eben
-- nicht definierten Seitenbezugs:
-- 1. Fall:
-- es wurde kein Seitenbezug definiert,
-- daher kann diese Eigenschaft
-- vernachlässigt werden
-- 2. Fall:
-- Seitenlokalisierung ist definiert UND
-- muss dann mit der des Fall-bezogenen
-- ICD-Codes übereinstimmen
-----
AND
(
DIW.DW_ID IN
(
SELECT
```

```

66 DWI.DW_ID
67 FROM
68 DIAGNOSE_WERT_ICD DWI,
69 FALL_ICD_INKL_OP FIC
70 WHERE
71 DWI.DW_ID = DIW.DW_ID
72 AND
73 FIC.FALL_ID = FAL.FALL_ID
74 AND
75 FIC.ICD_ID = DWI.ICD_ID
76 AND
77 DWI.DWI_ID NOT IN
78 (
79 SELECT
80 DWL.DWI_ID
81 FROM
82 DIAGNOSE_WERT_ICD_LOK DWL
83 WHERE
84 DWL.DWI_ID = DWI.DWI_ID
85 )
86 )
87 OR
88 DIW.DW_ID IN
89 (
90 SELECT
91 DWI.DW_ID
92 FROM
93 DIAGNOSE_WERT_ICD DWI,
94 DIAGNOSE_WERT_ICD_LOK DWL,
95 FALL_ICD_INKL_OP FIC
96 WHERE
97 DWI.DW_ID = DIW.DW_ID
98 AND
99 FIC.FALL_ID = FAL.FALL_ID
100 AND
101 FIC.ICD_ID = DWI.ICD_ID
102 AND
103 DWI.DWI_ID = DWL.DWI_ID
104 AND
105 FIC.ICDLOK_ID = DWL.ICDLOK_ID
106 )
107 )
108 -- -----
109 -- Ende: Auflösung des ggf. zu
110 -- berücksichtigenden Seitenbezugs
111 -- -----
112 -- -----
113 -- Kategorie-Auswertung:
114 -- Berücksichtigung von gemäß der Kontext-
115 -- Phasen-Zuordnung ein- und
116 -- auszuschließenden ICD-Kategorien
117 -- -----
118 AND
119 (
120 DIW.DW_ID IN
121 (
122 SELECT
123 DWI.DW_ID
124 FROM
125 DIAGNOSE_WERT_ICD DWI,
126 FALL_ICD_INKL_OP FIC,
127 PHASE_ICD_KATEGORIE PIK,
128 KONTEXT_KOMPLEX_PHASE KPH
129 WHERE
130 DWI.DW_ID = DIW.DW_ID
131 AND
132 FIC.FALL_ID = FAL.FALL_ID
133 AND

```

```

DWI.ICD_ID = FIC.ICD_ID                                134
AND                                                       135
PIK.ICDKAT_ID = FIC.ICDKAT_ID                          136
AND                                                       137
KPH.KKP_ID = KKP.KKP_ID                                138
AND                                                       139
PIK.PH_ID = KPH.PH_ID                                  140
AND                                                       141
PIK.PHZT_ID = 1 -- EINSCHLUSS!                          142
)                                                         143
AND                                                       144
DIW.DW_ID NOT IN                                       145
(                                                         146
SELECT                                                    147
DWI.DW_ID                                                148
FROM                                                    149
DIAGNOSE_WERT_ICD          DWI,                        150
FALL_ICD_INKL_OP          FIC,                        151
PHASE_ICD_KATEGORIE      PIK,                        152
KONTEXT_KOMPLEX_PHASE    KPH                        153
WHERE                                                              154
DWI.DW_ID = DIW.DW_ID                                          155
AND                                                              156
FIC.FALL_ID = FAL.FALL_ID                                      157
AND                                                              158
DWI.ICD_ID = FIC.ICD_ID                                       159
AND                                                              160
PIK.ICDKAT_ID = FIC.ICDKAT_ID                                 161
AND                                                              162
KPH.KKP_ID = KKP.KKP_ID                                       163
AND                                                              164
PIK.PH_ID = KPH.PH_ID                                         165
AND                                                              166
PIK.PHZT_ID = 2 -- AUSSCHLUSS!                               167
)                                                         168
)                                                         169
-----                                                  170
-- Ende: Kategorie-Auswertung                               171
-- -----                                                  172
-----                                                  173
-- Auswertung Zeitpunkt:                                   174
-- Berücksichtigung von chronologischen                   175
-- Gesichtspunkten (Zeitpunkt im Beh.-                   176
-- Verlauf, Zuordnung zu Erst- und ggf.                   177
-- Folgeeingriffen etc.) in Abhängigkeit                   178
-- von der definierten Phase                               179
-----                                                  180
AND                                                       181
(                                                         182
-----                                                  183
-- Auswertung Zeitpunkt -                                  184
-- Phasen: Aufnahme                                       185
--           Behandlung                                     186
--           intraop. allgemein                             187
--           Behandlung (neu)                              188
-- Keine zeitliche Einschränkung, da durch                 189
-- die ICD-Kategorie bereits eine                           190
-- eindeutige Zuordnung möglich ist                         191
-- (bei Beachtung der Ein- und Ausschlüsse!)               192
-----                                                  193
DIW.DW_ID IN                                             194
(                                                         195
SELECT                                                    196
DWI.DW_ID                                                197
FROM                                                    198
DIAGNOSE_WERT_ICD          DWI,                        199
FALL_ICD_INKL_OP          FIC,                        200
KONTEXT_KOMPLEX_PHASE    KPH                        201

```

```

202 WHERE
203 DWI.DW_ID = DIW.DW_ID
204 AND
205 FIC.FALL_ID = FAL.FALL_ID
206 AND
207 DWI.ICD_ID = FIC.ICD_ID
208 AND
209 KPH.KKP_ID = KKP.KKP_ID
210 AND
211 KPH.PH_ID IN (1, 2, 5, 6)
212 )
213 -----
214 -- Auswertung Zeitpunkt -
215 -- Phase: postoperativ
216 -- Zeitliche Einschränkung: Zeitpunkt muss
217 -- mit dem Beginn des ersten Eingriffs
218 -- übereinstimmen oder hinter diesem liegen
219 -----
220 OR
221 DIW.DW_ID IN
222 (
223 SELECT
224 DWI.DW_ID
225 FROM
226 DIAGNOSE_WERT_ICD      DWI,
227 FALL_ICD_INKL_OP      FIC,
228 KONTEXT_KOMPLEX_PHASE KPH
229 WHERE
230 DWI.DW_ID = DIW.DW_ID
231 AND
232 FIC.FALL_ID = FAL.FALL_ID
233 AND
234 DWI.ICD_ID = FIC.ICD_ID
235 AND
236 KPH.KKP_ID = KKP.KKP_ID
237 AND
238 KPH.PH_ID IN (7)
239 AND
240 FIC.DATUM >=
241 (
242 SELECT
243 MIN (FOP.BEGINN)
244 FROM
245 FALL_OP FOP
246 WHERE
247 FOP.FALL_ID = FAL.FALL_ID
248 )
249 )
250 -----
251 -- Auswertung Zeitpunkt -
252 -- Phase: intraop. Haupteingriff
253 -- Zeitliche Einschränkung: Zeitpunkt muss
254 -- mit dem Beginn des ersten Eingriffs
255 -- übereinstimmen
256 -----
257 OR
258 DIW.DW_ID IN
259 (
260 SELECT
261 DWI.DW_ID
262 FROM
263 DIAGNOSE_WERT_ICD      DWI,
264 FALL_ICD_INKL_OP      FIC,
265 KONTEXT_KOMPLEX_PHASE KPH
266 WHERE
267 DWI.DW_ID = DIW.DW_ID
268 AND
269 FIC.FALL_ID = FAL.FALL_ID

```



```

AND 270
DWI.ICD_ID = FIC.ICD_ID 271
AND 272
KPH.KKP_ID = KKP.KKP_ID 273
AND 274
KPH.PH_ID IN (3) 275
AND 276
FIC.DATUM = 277
( 278
SELECT 279
MIN (FOP.BEGINN) 280
FROM 281
FALL_OP FOP 282
WHERE 283
FOP.FALL_ID = FAL.FALL_ID 284
) 285
) 286
----- 287
-- Auswertung Zeitpunkt - 288
-- Phase: intraop. Folgeeingriff(e) 289
-- Zeitliche Einschränkung: Zeitpunkt muss 290
-- hinter dem Beginn des ersten Eingriffs 291
-- liegen 292
----- 293
OR 294
DIW.DW_ID IN 295
( 296
SELECT 297
DWI.DW_ID 298
FROM 299
DIAGNOSE_WERT_ICD DWI, 300
FALL_ICD_INKL_OP FIC, 301
KONTEXT_KOMPLEX_PHASE KPH 302
WHERE 303
DWI.DW_ID = DIW.DW_ID 304
AND 305
FIC.FALL_ID = FAL.FALL_ID 306
AND 307
DWI.ICD_ID = FIC.ICD_ID 308
AND 309
KPH.KKP_ID = KKP.KKP_ID 310
AND 311
KPH.PH_ID IN (4) 312
AND 313
FIC.DATUM > 314
( 315
SELECT 316
MIN (FOP.BEGINN) 317
FROM 318
FALL_OP FOP 319
WHERE 320
FOP.FALL_ID = FAL.FALL_ID 321
) 322
) 323
----- 324
-- Ende: Auswertung Zeitpunkt 325
----- 326
); 327

```

## A4.16. View [Fall\_Kont\_Komp\_Proz\_Wert\_OPS]

```
1 CREATE OR REPLACE VIEW FALL_KONT_KOMP_PROZ_WERT_OPS AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
9 FAL.ABGANG_FA        AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH        AS ABGANG_KH,
12 PRO.P_ID             AS P_ID,
13 PRO.BEZEICHNUNG      AS PROZEDUR_BEZEICHNUNG,
14 PRW.PW_ID            AS PW_ID,
15 PRW.BEZEICHNUNG      AS PROZEDUR_WERT_BEZEICHNUNG,
16 PRW.WERT             AS PROZEDUR_WERT,
17 KKP.KKP_ID           AS KKP_ID,
18 KON.K_ID             AS K_ID,
19 KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
20 KKP.RANG             AS KONTEXT_RANG,
21 KPL.KP_ID            AS KP_ID,
22 KPL.BEZEICHNUNG      AS KOMPLEX_BEZEICHNUNG,
23 KPW.KPW_ID           AS KPW_ID,
24 KPW.BEZEICHNUNG      AS KOMPLEX_WERT_BEZEICHNUNG,
25 KPW.RANG             AS KOMPLEX_WERT_RANG,
26 KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
27 KPWPW_ID            AS KPWPW_ID
28 FROM
29 FALL                 FAL,
30 PROZEDUR             PRO,
31 PROZEDUR_WERT        PRW,
32 KONTEXT_KOMPLEX     KKP,
33 KONTEXT              KON,
34 KOMPLEX              KPL,
35 KOMPLEX_WERT         KPW,
36 KOMPLEX_WERT_P_WERT KWP
37 WHERE
38 KKP.KP_ID = KPL.KP_ID
39 AND
40 KKP.KP_ID = KPW.KP_ID
41 AND
42 KPW.KPW_ID = KWP.KPW_ID
43 AND
44 KWP.PW_ID = PRW.PW_ID
45 AND
46 PRW.P_ID = PRO.P_ID
47 AND
48 KKP.K_ID = KON.K_ID
49 -----
50 -- Auflösung eines definierten oder eben
51 -- nicht definierten Seitenbezugs:
52 -- 1. Fall:
53 -- es wurde kein Seitenbezug definiert,
54 -- daher kann diese Eigenschaft
55 -- vernachlässigt werden
56 -- 2. Fall:
57 -- Seitenlokalisierung ist definiert UND
58 -- muss dann mit der des Fall-bezogenen
59 -- OPS-Codes übereinstimmen
60 -----
61 AND
62 (
63 PRW.PW_ID IN
64 (
65 SELECT
```

```

PWO.PW_ID 66
FROM 67
PROZEDUR_WERT_OPS PWO, 68
FALL_OPS_INKL_OP FIN 69
WHERE 70
PWO.PW_ID = PRW.PW_ID 71
AND 72
FIN.FALL_ID = FAL.FALL_ID 73
AND 74
FIN.OPS_ID = PWO.OPS_ID 75
AND 76
PWO.PWO_ID NOT IN 77
( 78
SELECT 79
PWL.PWO_ID 80
FROM 81
PROZEDUR_WERT_OPS_LOK PWL 82
WHERE 83
PWL.PWO_ID = PWO.PWO_ID 84
) 85
) 86
OR 87
PRW.PW_ID IN 88
( 89
SELECT 90
PWO.PW_ID 91
FROM 92
PROZEDUR_WERT_OPS PWO, 93
PROZEDUR_WERT_OPS_LOK PWL, 94
FALL_OPS_INKL_OP FIN 95
WHERE 96
PWO.PW_ID = PRW.PW_ID 97
AND 98
FIN.FALL_ID = FAL.FALL_ID 99
AND 100
FIN.OPS_ID = PWO.OPS_ID 101
AND 102
PWO.PWO_ID = PWL.PWO_ID 103
AND 104
FIN.OPSLOK_ID = PWL.OPSLOK_ID 105
) 106
) 107
-- ----- 108
-- Ende: Auflösung des ggf. zu 109
-- berücksichtigenden Seitenbezugs 110
-- ----- 111
-- ----- 112
-- Kategorie-Auswertung: 113
-- Berücksichtigung von gemäß der Kontext- 114
-- Phasen-Zuordnung ein- und 115
-- auszuschließenden OPS-Kategorien 116
-- ----- 117
AND 118
( 119
PRW.PW_ID IN 120
( 121
SELECT 122
PWO.PW_ID 123
FROM 124
PROZEDUR_WERT_OPS PWO, 125
FALL_OPS_INKL_OP FIN, 126
PHASE_OPS301_KATEGORIE POK, 127
KONTEXT_KOMPLEX_PHASE KPH 128
WHERE 129
PWO.PW_ID = PRW.PW_ID 130
AND 131
FIN.FALL_ID = FAL.FALL_ID 132
AND 133

```

```

134 PWO.OPS_ID = FIN.OPS_ID
135 AND
136 POK.OPSKAT_ID = FIN.OPSKAT_ID
137 AND
138 KPH.KKP_ID = KKP.KKP_ID
139 AND
140 POK.PH_ID = KPH.PH_ID
141 AND
142 POK.PHZT_ID = 1 -- EINSCHLUSS!
143 )
144 AND
145 PRW.PW_ID NOT IN
146 (
147 SELECT
148 PWO.PW_ID
149 FROM
150 PROZEDUR_WERT_OPS      PWO,
151 FALL_OPS_INKL_OP      FIN,
152 PHASE_OPS301_KATEGORIE POK,
153 KONTEXT_KOMPLEX_PHASE KPH
154 WHERE
155 PWO.PW_ID = PRW.PW_ID
156 AND
157 FIN.FALL_ID = FAL.FALL_ID
158 AND
159 PWO.OPS_ID = FIN.OPS_ID
160 AND
161 POK.OPSKAT_ID = FIN.OPSKAT_ID
162 AND
163 KPH.KKP_ID = KKP.KKP_ID
164 AND
165 POK.PH_ID = KPH.PH_ID
166 AND
167 POK.PHZT_ID = 2 -- AUSSCHLUSS!
168 )
169 )
170 -----
171 -- Ende: Kategorie-Auswertung
172 -----
173 -----
174 -- Auswertung Zeitpunkt:
175 -- Berücksichtigung von chronologischen
176 -- Gesichtspunkten (Zeitpunkt im Beh.-
177 -- Verlauf, Zuordnung zu Erst- und ggf.
178 -- Folgeeingriffen etc.) in Abhängigkeit
179 -- von der definierten Phase
180 -----
181 AND
182 (
183 -----
184 -- Auswertung Zeitpunkt -
185 -- Phasen: Aufnahme
186 --      Behandlung
187 --      intraop. allgemein
188 --      Behandlung (neu)
189 -- Keine zeitliche Einschränkung, da durch
190 -- die OPS-Kategorie bereits eine
191 -- eindeutige Zuordnung möglich ist
192 -- (bei Beachtung der Ein- und Ausschlüsse!)
193 -----
194 PRW.PW_ID IN
195 (
196 SELECT
197 PWO.PW_ID
198 FROM
199 PROZEDUR_WERT_OPS      PWO,
200 FALL_OPS_INKL_OP      FIN,
201 KONTEXT_KOMPLEX_PHASE KPH

```

```

WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
PWO.OPS_ID = FIN.OPS_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (1, 2, 5, 6)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: postoperativ
-- Zeitliche Einschränkung: Zeitpunkt muss
-- mit dem Beginn des ersten Eingriffs
-- übereinstimmen oder hinter diesem liegen
-----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_INKL_OP      FIN,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
PWO.OPS_ID = FIN.OPS_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (7)
AND
FIN.DATUM >=
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP  FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Haupteingriff
-- Zeitliche Einschränkung: Zeitpunkt muss
-- mit dem Beginn des ersten Eingriffs
-- übereinstimmen
-----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_INKL_OP      FIN,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID

```

```

270 AND
271 PWO.OPS_ID = FIN.OPS_ID
272 AND
273 KPH.KKP_ID = KKP.KKP_ID
274 AND
275 KPH.PH_ID IN (3)
276 AND
277 FIN.DATUM =
278 (
279 SELECT
280 MIN (FOP.BEGINN)
281 FROM
282 FALL_OP FOP
283 WHERE
284 FOP.FALL_ID = FAL.FALL_ID
285 )
286 )
287 -----
288 -- Auswertung Zeitpunkt -
289 -- Phase: intraop. Folgeeingriff(e)
290 -- Zeitliche Einschränkung: Zeitpunkt muss
291 -- hinter dem Beginn des ersten Eingriffs
292 -- liegen
293 -----
294 OR
295 PRW.PW_ID IN
296 (
297 SELECT
298 PWO.PW_ID
299 FROM
300 PROZEDUR_WERT_OPS      PWO,
301 FALL_OPS_INKL_OP      FIN,
302 KONTEXT_KOMPLEX_PHASE KPH
303 WHERE
304 PWO.PW_ID = PRW.PW_ID
305 AND
306 FIN.FALL_ID = FAL.FALL_ID
307 AND
308 PWO.OPS_ID = FIN.OPS_ID
309 AND
310 KPH.KKP_ID = KKP.KKP_ID
311 AND
312 KPH.PH_ID IN (4)
313 AND
314 FIN.DATUM >
315 (
316 SELECT
317 MIN (FOP.BEGINN)
318 FROM
319 FALL_OP FOP
320 WHERE
321 FOP.FALL_ID = FAL.FALL_ID
322 )
323 )
324 -----
325 -- Ende: Auswertung Zeitpunkt
326 -----
327 );

```

## A4.17. View [Fall\_Kont\_Komp\_DWK\_DW\_ICD]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_DWK_DW_ICD AS
SELECT
FAL.FALL_ID           AS FALL_ID,
FAL.CODE             AS FALL_CODE,
FAL.NACHNAME         AS NACHNAME,
FAL.VORNAME          AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA      AS AUFNAHME_FA,
FAL.ABGANG_FA        AS ABGANG_FA,
FAL.AUFNAHME_KH      AS AUFNAHME_KH,
FAL.ABGANG_KH        AS ABGANG_KH,
DIA.D_ID             AS D_ID,
DIA.BEZEICHNUNG      AS DIAGNOSE_BEZEICHNUNG,
DIW.DW_ID            AS DW_ID,
DIW.BEZEICHNUNG      AS DIAGNOSE_WERT_BEZEICHNUNG,
DIW.WERT             AS DIAGNOSE_WERT,
DWK.DWK_ID           AS DWK_ID,
DWK.BEZEICHNUNG      AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
DWK.WERT             AS DIAGNOSE_WERT_KOMB,
DWK.DWKT_ID          AS DWKT_ID,
DKT.BEZEICHNUNG      AS DIAGNOSE_WERT_KOMBINATIONSTYP,
KKP.KKP_ID           AS KKP_ID,
KON.K_ID             AS K_ID,
KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
KKP.RANG             AS KONTEXT_RANG,
KPL.KP_ID            AS KP_ID,
KPL.BEZEICHNUNG      AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID           AS KPW_ID,
KPW.BEZEICHNUNG      AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.RANG             AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
KPW.KWT_ID           AS KWT_ID,
KWT.BEZEICHNUNG      AS KOMPLEX_WERT_KOMBINATIONSTYP,
KWK.KPWDWK_ID        AS KPWDWK_ID
FROM
FALL                 FAL,
DIAGNOSE             DIA,
DIAGNOSE_WERT        DIW,
DIAGNOSE_WERT_KOMBINATION DKD,
DIAGNOSE_WERT_KOMBINATIONSTYP DKT,
DIAG_WERT_KOMB_DIAG_WERT DKD,
KONTEXT_KOMPLEX      KKP,
KONTEXT              KON,
KOMPLEX              KPL,
KOMPLEX_WERT         KPW,
KOMPLEX_WERT_D_WERT_KOMB KWK,
KOMPLEX_WERT_KOMBINATIONSTYP KWT
WHERE
KKP.KP_ID = KPL.KP_ID
AND
KKP.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWK.KPW_ID
AND
KPW.KWT_ID = KWT.KWT_ID
AND
KWK.DWK_ID = DWK.DWK_ID
AND
DWK.DWK_ID = DKD.DWK_ID
AND
DWK.DWKT_ID = DKT.DWKT_ID
AND
DKD.DW_ID = DIW.DW_ID
AND
DIW.D_ID = DIA.D_ID
```

```

66 AND
67 KKP.K_ID = KON.K_ID
68 -- -----
69 -- Auflösung eines definierten oder eben
70 -- nicht definierten Seitenbezugs:
71 -- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
72 -- vernachlässigt werden
73 -- 2. Fall: Seitenlokalisierung ist definiert UND muss dann mit der des Fall-bezogenen
74 -- ICD-Codes übereinstimmen
75 -- -----
76 AND
77 (
78 DIW.DW_ID IN
79 (
80 SELECT
81 DWI.DW_ID
82 FROM
83 DIAGNOSE_WERT_ICD DWI,
84 FALL_ICD_INKL_OP FIC
85 WHERE
86 DWI.DW_ID = DIW.DW_ID
87 AND
88 FIC.FALL_ID = FAL.FALL_ID
89 AND
90 FIC.ICD_ID = DWI.ICD_ID
91 AND
92 DWI.DWI_ID NOT IN
93 (
94 SELECT
95 DWL.DWI_ID
96 FROM
97 DIAGNOSE_WERT_ICD_LOK DWL
98 WHERE
99 DWL.DWI_ID = DWI.DWI_ID
100 )
101 )
102 OR
103 DIW.DW_ID IN
104 (
105 SELECT
106 DWI.DW_ID
107 FROM
108 DIAGNOSE_WERT_ICD DWI,
109 DIAGNOSE_WERT_ICD_LOK DWL,
110 FALL_ICD_INKL_OP FIC
111 WHERE
112 DWI.DW_ID = DIW.DW_ID
113 AND
114 FIC.FALL_ID = FAL.FALL_ID
115 AND
116 FIC.ICD_ID = DWI.ICD_ID
117 AND
118 DWI.DWI_ID = DWL.DWI_ID
119 AND
120 FIC.ICDLOK_ID = DWL.ICDLOK_ID
121 )
122 )
123 -- -----
124 -- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs
125 -- -----
126 -- -----
127 -- Kategorie-Auswertung:
128 -- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und auszuschließenden
129 -- ICD-Kategorien
130 -- -----
131 AND
132 (
133 DIW.DW_ID IN

```



```

(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_INKL_OP      FIC,
PHASE_ICD_KATEGORIE   PIK,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
PIK.ICDKAT_ID = FIC.ICDKAT_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
PIK.PH_ID = KPH.PH_ID
AND
PIK.PHZT_ID = 1  -- EINSCHLUSS!
)
AND
DIW.DW_ID NOT IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_INKL_OP      FIC,
PHASE_ICD_KATEGORIE   PIK,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
PIK.ICDKAT_ID = FIC.ICDKAT_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
PIK.PH_ID = KPH.PH_ID
AND
PIK.PHZT_ID = 2  -- AUSSCHLUSS!
)
)
-- -----
-- Ende: Kategorie-Auswertung
-- -----
-- Auswertung Zeitpunkt:
-- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf,
-- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten
-- Phase
-- -----
AND
(
-- -----
-- Auswertung Zeitpunkt -
-- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu).
-- Keine zeitliche Einschränkung, da durch die ICD-Kategorie bereits eine
-- eindeutige Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!)
-- -----
DIW.DW_ID IN
(

```

```

202 SELECT
203 DWI.DW_ID
204 FROM
205 DIAGNOSE_WERT_ICD      DWI,
206 FALL_ICD_INKL_OP      FIC,
207 KONTEXT_KOMPLEX_PHASE KPH
208 WHERE
209 DWI.DW_ID = DIW.DW_ID
210 AND
211 FIC.FALL_ID = FAL.FALL_ID
212 AND
213 DWI.ICD_ID = FIC.ICD_ID
214 AND
215 KPH.KKP_ID = KKP.KKP_ID
216 AND
217 KPH.PH_ID IN (1, 2, 5, 6)
218 )
219 -- -----
220 -- Auswertung Zeitpunkt -
221 -- Phase: postoperativ
222 -- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
223 -- übereinstimmen oder hinter diesem liegen
224 -- -----
225 OR
226 DIW.DW_ID IN
227 (
228 SELECT
229 DWI.DW_ID
230 FROM
231 DIAGNOSE_WERT_ICD      DWI,
232 FALL_ICD_INKL_OP      FIC,
233 KONTEXT_KOMPLEX_PHASE KPH
234 WHERE
235 DWI.DW_ID = DIW.DW_ID
236 AND
237 FIC.FALL_ID = FAL.FALL_ID
238 AND
239 DWI.ICD_ID = FIC.ICD_ID
240 AND
241 KPH.KKP_ID = KKP.KKP_ID
242 AND
243 KPH.PH_ID IN (7)
244 AND
245 FIC.DATUM >=
246 (
247 SELECT
248 MIN (FOP.BEGINN)
249 FROM
250 FALL_OP FOP
251 WHERE
252 FOP.FALL_ID = FAL.FALL_ID
253 )
254 )
255 -- -----
256 -- Auswertung Zeitpunkt -
257 -- Phase: intraop. Haupteingriff
258 -- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
259 -- übereinstimmen
260 -- -----
261 OR
262 DIW.DW_ID IN
263 (
264 SELECT
265 DWI.DW_ID
266 FROM
267 DIAGNOSE_WERT_ICD      DWI,
268 FALL_ICD_INKL_OP      FIC,
269 KONTEXT_KOMPLEX_PHASE KPH

```

```

WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (3)
AND
FIC.DATUM =
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Folgeeingriff(e)
-- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs
-- liegen
-----
OR
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_INKL_OP      FIC,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (4)
AND
FIC.DATUM >
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Ende: Auswertung Zeitpunkt
-----
);

```

## A4.18. View [Fall\_Kont\_Komp\_PWK\_PW\_OPS]

```
1 CREATE OR REPLACE VIEW FALL_KONT_KOMP_PWK_PW_OPS AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME         AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
9 FAL.ABGANG_FA       AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH       AS ABGANG_KH,
12 PRO.P_ID            AS P_ID,
13 PRO.BEZEICHNUNG     AS PROZEDUR_BEZEICHNUNG,
14 PRW.PW_ID           AS PW_ID,
15 PRW.BEZEICHNUNG     AS PROZEDUR_WERT_BEZEICHNUNG,
16 PRW.WERT            AS PROZEDUR_WERT,
17 PWK.PWK_ID          AS PWK_ID,
18 PWK.BEZEICHNUNG     AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
19 PWK.WERT            AS PROZEDUR_WERT_KOMB,
20 PWK.PWKT_ID         AS PWKT_ID,
21 PKT.BEZEICHNUNG     AS PROZEDUR_WERT_KOMBINATIONSTYP,
22 KKP.KKP_ID          AS KKP_ID,
23 KON.K_ID            AS K_ID,
24 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
25 KKP.RANG            AS KONTEXT_RANG,
26 KPL.KP_ID           AS KP_ID,
27 KPL.BEZEICHNUNG     AS KOMPLEX_BEZEICHNUNG,
28 KPW.KPW_ID          AS KPW_ID,
29 KPW.BEZEICHNUNG     AS KOMPLEX_WERT_BEZEICHNUNG,
30 KPW.RANG            AS KOMPLEX_WERT_RANG,
31 KPW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
32 KPW.KWT_ID          AS KWT_ID,
33 KKT.BEZEICHNUNG     AS KOMPLEX_WERT_KOMBINATIONSTYP,
34 KWK.KPWPWK_ID       AS KPWPWK_ID
35 FROM
36 FALL                FAL,
37 PROZEDUR            PRO,
38 PROZEDUR_WERT       PRW,
39 PROZEDUR_WERT_KOMBINATION PWK,
40 PROZEDUR_WERT_KOMBINATIONSTYP PKT,
41 PROZ_WERT_KOMB_PROZ_WERT PKP,
42 KONTEXT_KOMPLEX    KKP,
43 KONTEXT            KON,
44 KOMPLEX            KPL,
45 KOMPLEX_WERT       KPW,
46 KOMPLEX_WERT_P_WERT_KOMB KWK,
47 KOMPLEX_WERT_KOMBINATIONSTYP KKT
48 WHERE
49 KKP.KP_ID = KPL.KP_ID
50 AND
51 KKP.KP_ID = KPW.KP_ID
52 AND
53 KPW.KPW_ID = KWK.KPW_ID
54 AND
55 KPW.KWT_ID = KKT.KWT_ID
56 AND
57 KWK.PWK_ID = PWK.PWK_ID
58 AND
59 PWK.PWK_ID = PKP.PWK_ID
60 AND
61 PWK.PWKT_ID = PKT.PWKT_ID
62 AND
63 PKP.PW_ID = PRW.PW_ID
64 AND
65 PRW.P_ID = PRO.P_ID
```

```

AND
KKP.K_ID = KON.K_ID
-----
-- Auflösung eines definierten oder eben
-- nicht definierten Seitenbezugs:
-- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
-- vernachlässigt werden
-- 2. Fall: Seitenlokalisierung ist definiert UND muss dann mit der des Fall-bezogenen
-- OPS-Codes übereinstimmen
-----
AND
(
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS PWO,
FALL_OPS_INKL_OP FIN
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
FIN.OPS_ID = PWO.OPS_ID
AND
PWO.PWO_ID NOT IN
(
SELECT
PWL.PWO_ID
FROM
PROZEDUR_WERT_OPS_LOK PWL
WHERE
PWL.PWO_ID = PWO.PWO_ID
)
)
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS PWO,
PROZEDUR_WERT_OPS_LOK PWL,
FALL_OPS_INKL_OP FIN
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
FIN.OPS_ID = PWO.OPS_ID
AND
PWO.PWO_ID = PWL.PWO_ID
AND
FIN.OPSLOK_ID = PWL.OPSLOK_ID
)
)
-----
-- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs
-----
-- Kategorie-Auswertung:
-- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und auszuschließenden
-- OPS-Kategorien
-----
AND
(
PRW.PW_ID IN

```

```

134 (
135 SELECT
136 PWO.PW_ID
137 FROM
138 PROZEDUR_WERT_OPS      PWO,
139 FALL_OPS_INKL_OP      FIN,
140 PHASE_OPS301_KATEGORIE POK,
141 KONTEXT_KOMPLEX_PHASE KPH
142 WHERE
143 PWO.PW_ID = PRW.PW_ID
144 AND
145 FIN.FALL_ID = FAL.FALL_ID
146 AND
147 PWO.OPS_ID = FIN.OPS_ID
148 AND
149 POK.OPSKAT_ID = FIN.OPSKAT_ID
150 AND
151 KPH.KKP_ID = KKP.KKP_ID
152 AND
153 POK.PH_ID = KPH.PH_ID
154 AND
155 POK.PHZT_ID = 1 -- EINSCHLUSS!
156 )
157 AND
158 PRW.PW_ID NOT IN
159 (
160 SELECT
161 PWO.PW_ID
162 FROM
163 PROZEDUR_WERT_OPS      PWO,
164 FALL_OPS_INKL_OP      FIN,
165 PHASE_OPS301_KATEGORIE POK,
166 KONTEXT_KOMPLEX_PHASE KPH
167 WHERE
168 PWO.PW_ID = PRW.PW_ID
169 AND
170 FIN.FALL_ID = FAL.FALL_ID
171 AND
172 PWO.OPS_ID = FIN.OPS_ID
173 AND
174 POK.OPSKAT_ID = FIN.OPSKAT_ID
175 AND
176 KPH.KKP_ID = KKP.KKP_ID
177 AND
178 POK.PH_ID = KPH.PH_ID
179 AND
180 POK.PHZT_ID = 2 -- AUSSCHLUSS!
181 )
182 )
183 -----
184 -- Ende: Kategorie-Auswertung
185 -----
186 -----
187 -- Auswertung Zeitpunkt:
188 -- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf,
189 -- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten
190 -- Phase
191 -----
192 AND
193 (
194 -----
195 -- Auswertung Zeitpunkt -
196 -- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu).
197 -- Keine zeitliche Einschränkung, da durch die OPS-Kategorie bereits eine
198 -- eindeutige Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!)
199 -----
200 PRW.PW_ID IN
201 (

```

```

SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_INKL_OP      FIN,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
PWO.OPS_ID = FIN.OPS_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (1, 2, 5, 6)
)
-- -----
-- Auswertung Zeitpunkt -
-- Phase: postoperativ
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen oder hinter diesem liegen
-- -----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_INKL_OP      FIN,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FIN.FALL_ID = FAL.FALL_ID
AND
PWO.OPS_ID = FIN.OPS_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (7)
AND
FIN.DATUM >=
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP  FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-- -----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Haupteingriff
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen
-- -----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_INKL_OP      FIN,
KONTEXT_KOMPLEX_PHASE KPH

```

```

270 WHERE
271 PWO.PW_ID = PRW.PW_ID
272 AND
273 FIN.FALL_ID = FAL.FALL_ID
274 AND
275 PWO.OPS_ID = FIN.OPS_ID
276 AND
277 KPH.KKP_ID = KKP.KKP_ID
278 AND
279 KPH.PH_ID IN (3)
280 AND
281 FIN.DATUM =
282 (
283 SELECT
284 MIN (FOP.BEGINN)
285 FROM
286 FALL_OP FOP
287 WHERE
288 FOP.FALL_ID = FAL.FALL_ID
289 )
290 )
291 -- -----
292 -- Auswertung Zeitpunkt -
293 -- Phase: intraop. Folgeeingriff(e)
294 -- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs
295 -- liegen
296 -- -----
297 OR
298 PRW.PW_ID IN
299 (
300 SELECT
301 PWO.PW_ID
302 FROM
303 PROZEDUR_WERT_OPS      PWO,
304 FALL_OPS_INKL_OP      FIN,
305 KONTEXT_KOMPLEX_PHASE KPH
306 WHERE
307 PWO.PW_ID = PRW.PW_ID
308 AND
309 FIN.FALL_ID = FAL.FALL_ID
310 AND
311 PWO.OPS_ID = FIN.OPS_ID
312 AND
313 KPH.KKP_ID = KKP.KKP_ID
314 AND
315 KPH.PH_ID IN (4)
316 AND
317 FIN.DATUM >
318 (
319 SELECT
320 MIN (FOP.BEGINN)
321 FROM
322 FALL_OP FOP
323 WHERE
324 FOP.FALL_ID = FAL.FALL_ID
325 )
326 )
327 -- -----
328 -- Ende: Auswertung Zeitpunkt
329 -- -----
330 );

```



## A4.19. View [Fall\_Kont\_Komp\_Diag\_Wert\_Komb]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_DIAG_WERT_KOMB AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FAL.CODE             AS FALL_CODE,
FAL.NACHNAME         AS NACHNAME,
FAL.VORNAME          AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA     AS AUFNAHME_FA,
FAL.ABGANG_FA       AS ABGANG_FA,
FAL.AUFNAHME_KH     AS AUFNAHME_KH,
FAL.ABGANG_KH       AS ABGANG_KH,
DIA.D_ID             AS D_ID,
DIA.BEZEICHNUNG     AS DIAGNOSE_BEZEICHNUNG,
DWK.DWK_ID          AS DWK_ID,
DWK.BEZEICHNUNG     AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
DWK.WERT            AS DIAGNOSE_WERT_KOMB,
KKP.KKP_ID          AS KKP_ID,
KON.K_ID            AS K_ID,
KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
KKP.RANG            AS KONTEXT_RANG,
KPL.KP_ID           AS KP_ID,
KPL.BEZEICHNUNG     AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID          AS KPW_ID,
KPW.BEZEICHNUNG     AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.WERT            AS KOMPLEX_WERT,
KPW.RANG            AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG
FROM
FALL                FAL,
DIAGNOSE            DIA,
DIAGNOSE_WERT_KOMBINATION DWK,
DIAGNOSE_WERT_KOMB_ELEM_COUNT DKE,
KONTEXT_KOMPLEX    KKP,
KONTEXT             KON,
KOMPLEX             KPL,
KOMPLEX_WERT        KPW,
KOMPLEX_WERT_D_WERT_KOMB KWK
WHERE
KKP.K_ID = KON.K_ID
AND
KKP.KP_ID = KPL.KP_ID
AND
KPL.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWK.KPW_ID
AND
KWK.DWK_ID = DWK.DWK_ID
AND
DWK.DWK_ID = DKE.DWK_ID
AND
DWK.D_ID = DIA.D_ID
AND
(
(
-- Kombinationstyp: alle Diagnosen sind obligat!
DWK.DWKT_ID = 1
AND
DKE.ZAHL_ELEMENTE =
(
SELECT
COUNT (DISTINCT (KDK.DW_ID))
FROM
FALL_KONT_KOMP_DWK_DW_ICD  KDK
WHERE
KDK.FALL_ID = FAL.FALL_ID
```

```
66 AND
67 KDK.DWK_ID = DWK.DWK_ID
68 )
69 )
70 OR
71 (
72 -- Kombinationstyp: mindestens 2 Diagnosen sind obligat!
73 DWK.DWKT_ID = 2
74 AND
75 1 <
76 (
77 SELECT
78 COUNT (DISTINCT (KDK.DW_ID))
79 FROM
80 FALL_KONT_KOMP_DWK_DW_ICD KDK
81 WHERE
82 KDK.FALL_ID = FAL.FALL_ID
83 AND
84 KDK.DWK_ID = DWK.DWK_ID
85 )
86 )
87 );
```

## A4.20. View [Fall\_Kont\_Komp\_Proz\_Wert\_Komb]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_PROZ_WERT_KOMB AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FAL.CODE            AS FALL_CODE,
FAL.NACHNAME        AS NACHNAME,
FAL.VORNAME         AS VORNAME,
FAL.GEBURTSDATUM   AS GEBURTSDATUM,
FAL.AUFNAHME_FA    AS AUFNAHME_FA,
FAL.ABGANG_FA      AS ABGANG_FA,
FAL.AUFNAHME_KH    AS AUFNAHME_KH,
FAL.ABGANG_KH      AS ABGANG_KH,
PRO.P_ID           AS P_ID,
PRO.BEZEICHNUNG     AS PROZEDUR_BEZEICHNUNG,
PWK.PWK_ID         AS PWK_ID,
PWK.BEZEICHNUNG     AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
PWK.WERT           AS PROZEDUR_WERT_KOMB,
KKP.KKP_ID         AS KKP_ID,
KON.K_ID           AS K_ID,
KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
KKP.RANG           AS KONTEXT_RANG,
KPL.KP_ID          AS KP_ID,
KPL.BEZEICHNUNG     AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID         AS KPW_ID,
KPW.BEZEICHNUNG     AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.WERT           AS KOMPLEX_WERT,
KPW.RANG           AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG
FROM
FAL          FAL,
PROZEDUR    PRO,
PROZEDUR_WERT_KOMBINATION PWK,
PROZEDUR_WERT_KOMB_ELEM_COUNT PKE,
KONTEXT_KOMPLEX KKP,
KONTEXT      KON,
KOMPLEX      KPL,
KOMPLEX_WERT KPW,
KOMPLEX_WERT_P_WERT_KOMB KWK
WHERE
KKP.K_ID = KON.K_ID
AND
KKP.KP_ID = KPL.KP_ID
AND
KPL.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWK.KPW_ID
AND
KWK.PWK_ID = PWK.PWK_ID
AND
PWK.PWK_ID = PKE.PWK_ID
AND
PWK.P_ID = PRO.P_ID
AND
(
(
-- Kombinationstyp: alle Prozeduren sind obligat!
PWK.PWKT_ID = 1
AND
PKE.ZAHL_ELEMENTE =
(
SELECT
COUNT (DISTINCT (KPK.PW_ID))
FROM
FALL_KONT_KOMP_PWK_PW_OPS  KPK
WHERE
KPK.FALL_ID = FAL.FALL_ID
```

```
66 AND
67 KPK.PWK_ID = PWK.PWK_ID
68 )
69 )
70 OR
71 (
72 -- Kombinationstyp: mindestens 2 Prozeduren sind obligat!
73 PWK.PWKT_ID = 2
74 AND
75 1 <
76 (
77 SELECT
78 COUNT (DISTINCT (KPK.PW_ID))
79 FROM
80 FALL_KONT_KOMP_PWK_PW_OPS   KPK
81 WHERE
82 KPK.FALL_ID = FAL.FALL_ID
83 AND
84 KPK.PWK_ID = PWK.PWK_ID
85 )
86 )
87 );
```

## A4.21. View [Fall\_Kont\_Komp\_DW\_PW\_DWK\_PWK]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_DW_PW_DWK_PWK AS
-----
-- Diagnose-Werte
-----
SELECT
DWE.FALL_ID AS FALL_ID,
DWE.FALL_CODE AS FALL_CODE,
DWE.NACHNAME AS NACHNAME,
DWE.VORNAME AS VORNAME,
DWE.GEBURTSDATUM AS GEBURTSDATUM,
DWE.AUFNAHME_FA AS AUFNAHME_FA,
DWE.ABGANG_FA AS ABGANG_FA,
DWE.AUFNAHME_KH AS AUFNAHME_KH,
DWE.ABGANG_KH AS ABGANG_KH,
DWE.KKP_ID AS KKP_ID,
DWE.K_ID AS K_ID,
DWE.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
DWE.KONTEXT_RANG AS KONTEXT_RANG,
DWE.KP_ID AS KP_ID,
DWE.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
DWE.KPW_ID AS KPW_ID,
DWE.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
DWE.KOMPLEX_WERT_RANG AS KOMPLEX_WERT_RANG,
DWE.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
DWE.D_ID AS D_ID,
DWE.DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
DWE.DW_ID AS DW_ID,
DWE.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
DWE.DIAGNOSE_WERT AS DIAGNOSE_WERT,
NULL AS DWK_ID,
NULL AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
NULL AS DIAGNOSE_WERT_KOMB,
NULL AS P_ID,
NULL AS PROZEDUR_BEZEICHNUNG,
NULL AS PW_ID,
NULL AS PROZEDUR_WERT_BEZEICHNUNG,
NULL AS PROZEDUR_WERT,
NULL AS PWK_ID,
NULL AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
NULL AS PROZEDUR_WERT_KOMB
FROM
FALL_KONT_KOMP_DIAG_WERT_ICD DWE
UNION
-----
-- Prozedur-Werte
-----
SELECT
PWE.FALL_ID,
PWE.FALL_CODE,
PWE.NACHNAME,
PWE.VORNAME,
PWE.GEBURTSDATUM,
PWE.AUFNAHME_FA,
PWE.ABGANG_FA,
PWE.AUFNAHME_KH,
PWE.ABGANG_KH,
PWE.KKP_ID,
PWE.K_ID,
PWE.KONTEXT_BEZEICHNUNG,
PWE.KONTEXT_RANG,
PWE.KP_ID,
PWE.KOMPLEX_BEZEICHNUNG,
PWE.KPW_ID,
PWE.KOMPLEX_WERT_BEZEICHNUNG,
PWE.KOMPLEX_WERT_RANG,
```

```

66  PWE.KPW_EJN_ID_AUSWERTUNG,
67  NULL,
68  NULL,
69  NULL,
70  NULL,
71  NULL,
72  NULL,
73  NULL,
74  NULL,
75  PWE.P_ID,
76  PWE.PROZEDUR_BEZEICHNUNG,
77  PWE.PW_ID,
78  PWE.PROZEDUR_WERT_BEZEICHNUNG,
79  PWE.PROZEDUR_WERT,
80  NULL,
81  NULL,
82  NULL
83  FROM
84  FALL_KONT_KOMP_PROZ_WERT_OPS  PWE
85  UNION
86  -----
87  -- Diagnose-Kombinationswerte
88  -----
89  SELECT
90  DWK.FALL_ID,
91  DWK.FALL_CODE,
92  DWK.NACHNAME,
93  DWK.VORNAME,
94  DWK.GEBURTSDATUM,
95  DWK.AUFNAHME_FA,
96  DWK.ABGANG_FA,
97  DWK.AUFNAHME_KH,
98  DWK.ABGANG_KH,
99  DWK.KKP_ID,
100 DWK.K_ID,
101 DWK.KONTEXT_BEZEICHNUNG,
102 DWK.KONTEXT_RANG,
103 DWK.KP_ID,
104 DWK.KOMPLEX_BEZEICHNUNG,
105 DWK.KPW_ID,
106 DWK.KOMPLEX_WERT_BEZEICHNUNG,
107 DWK.KOMPLEX_WERT_RANG,
108 DWK.KPW_EJN_ID_AUSWERTUNG,
109 DWK.D_ID,
110 DWK.DIAGNOSE_BEZEICHNUNG,
111 NULL,
112 NULL,
113 NULL,
114 DWK.DWK_ID,
115 DWK.DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
116 DWK.DIAGNOSE_WERT_KOMB,
117 NULL,
118 NULL,
119 NULL,
120 NULL,
121 NULL,
122 NULL,
123 NULL,
124 NULL
125 FROM
126 FALL_KONT_KOMP_DIAG_WERT_KOMB  DWK
127 UNION
128 -----
129 -- Prozedur-Kombinationswerte
130 -----
131 SELECT
132 PWK.FALL_ID,
133 PWK.FALL_CODE,

```

PWK.NACHNAME,	134
PWK.VORNAME,	135
PWK.GEBURTSDATUM,	136
PWK.AUFNAHME_FA,	137
PWK.ABGANG_FA,	138
PWK.AUFNAHME_KH,	139
PWK.ABGANG_KH,	140
PWK.KKP_ID,	141
PWK.K_ID,	142
PWK.KONTEXT_BEZEICHNUNG,	143
PWK.KONTEXT_RANG,	144
PWK.KP_ID,	145
PWK.KOMPLEX_BEZEICHNUNG,	146
PWK.KPW_ID,	147
PWK.KOMPLEX_WERT_BEZEICHNUNG,	148
PWK.KOMPLEX_WERT_RANG,	149
PWK.KPW_EJN_ID_AUSWERTUNG,	150
NULL,	151
NULL,	152
NULL,	153
NULL,	154
NULL,	155
NULL,	156
NULL,	157
NULL,	158
PWK.P_ID,	159
PWK.PROZEDUR_BEZEICHNUNG,	160
NULL,	161
NULL,	162
NULL,	163
PWK.PWK_ID,	164
PWK.PROZEDUR_WERT_KOMB_BEZEICHNUNG,	165
PWK.PROZEDUR_WERT_KOMB	166
FROM	167
FALL_KONT_KOMP_PROZ_WERT_KOMB PWK;	168

## A4.22. View [Fall\_KPW]

```
1 CREATE OR REPLACE VIEW FALL_KPW AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM     AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA      AS AUFNAHME_FA,
9 FAL.ABGANG_FA        AS ABGANG_FA,
10 FAL.AUFNAHME_KH      AS AUFNAHME_KH,
11 FAL.ABGANG_KH        AS ABGANG_KH,
12 KKP.KKP_ID           AS KKP_ID,
13 KON.K_ID             AS K_ID,
14 KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
15 KKP.RANG             AS KONTEXT_KOMPLEX_RANG,
16 KPL.KP_ID            AS KP_ID,
17 KPL.BEZEICHNUNG      AS KOMPLEX_BEZEICHNUNG,
18 KPW.KPW_ID           AS KPW_ID,
19 KPW.BEZEICHNUNG      AS KOMPLEX_WERT_BEZEICHNUNG,
20 KPW.RANG             AS KOMPLEX_WERT_RANG,
21 KPW.WERT             AS KOMPLEX_WERT,
22 KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
23 KPW.KWT_ID           AS KWT_ID
24 FROM
25 FALL                 FAL,
26 KONTEXT_KOMPLEX     KKP,
27 KONTEXT              KON,
28 KOMPLEX              KPL,
29 KOMPLEX_WERT         KPW
30 WHERE
31 KKP.K_ID = KON.K_ID
32 AND
33 KKP.KP_ID = KPL.KP_ID
34 AND
35 KPL.KP_ID = KPW.KP_ID;
```



## A4.23. View [Fall\_Kont\_Komp\_W\_DW\_PW\_DWK\_PWK]

```

CREATE OR REPLACE VIEW FALL_KONT_KOMP_W_DW_PW_DWK_PWK AS
SELECT
FKP.FALL_ID AS FALL_ID,
FKP.FALL_CODE AS FALL_CODE,
FKP.NACHNAME AS NACHNAME,
FKP.VORNAME AS VORNAME,
FKP.GEBURTSDATUM AS GEBURTSDATUM,
FKP.AUFNAHME_FA AS AUFNAHME_FA,
FKP.ABGANG_FA AS ABGANG_FA,
FKP.AUFNAHME_KH AS AUFNAHME_KH,
FKP.ABGANG_KH AS ABGANG_KH,
FKP.KKP_ID AS KKP_ID,
FKP.K_ID AS K_ID,
FKP.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
FKP.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG,
FKP.KP_ID AS KP_ID,
FKP.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
FKP.KPW_ID AS KPW_ID,
FKP.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
FKP.KOMPLEX_WERT_RANG AS KOMPLEX_WERT_RANG,
FKP.KOMPLEX_WERT AS KOMPLEX_WERT,
FKP.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
FKP.KWT_ID AS KWT_ID,
DWE.D_ID AS DWE_D_ID,
DWE.DIAGNOSE_BEZEICHNUNG AS DWE_DIAGNOSE_BEZEICHNUNG,
DWE.DW_ID AS DW_ID,
DWE.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
DWE.DIAGNOSE_WERT AS DIAGNOSE_WERT,
DWK.D_ID AS DWK_D_ID,
DWK.DIAGNOSE_BEZEICHNUNG AS DWK_DIAGNOSE_BEZEICHNUNG,
DWK.DWK_ID AS DWK_ID,
DWK.DIAGNOSE_WERT_KOMB_BEZEICHNUNG AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
DWK.DIAGNOSE_WERT_KOMB AS DIAGNOSE_WERT_KOMB,
PWE.P_ID AS PWE_P_ID,
PWE.PROZEDUR_BEZEICHNUNG AS PWE_PROZEDUR_BEZEICHNUNG,
PWE.PW_ID AS PW_ID,
PWE.PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
PWE.PROZEDUR_WERT AS PROZEDUR_WERT,
PWK.P_ID AS PWK_P_ID,
PWK.PROZEDUR_BEZEICHNUNG AS PWK_PROZEDUR_BEZEICHNUNG,
PWK.PWK_ID AS PWK_ID,
PWK.PROZEDUR_WERT_KOMB_BEZEICHNUNG AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
PWK.PROZEDUR_WERT_KOMB AS PROZEDUR_WERT_KOMB
FROM
FALL_KPW FKP,
FALL_KONT_KOMP_DIAG_WERT_ICD DWE,
FALL_KONT_KOMP_DIAG_WERT_KOMB DWK,
FALL_KONT_KOMP_PROZ_WERT_OPS PWE,
FALL_KONT_KOMP_PROZ_WERT_KOMB PWK
WHERE
FKP.FALL_ID = DWE.FALL_ID (+)
AND
FKP.KPW_ID = DWE.KPW_ID (+)
AND
FKP.FALL_ID = DWK.FALL_ID (+)
AND
FKP.KPW_ID = DWK.KPW_ID (+)
AND
FKP.FALL_ID = PWE.FALL_ID (+)
AND
FKP.KPW_ID = PWE.KPW_ID (+)
AND
FKP.FALL_ID = PWK.FALL_ID (+)
AND
FKP.KPW_ID = PWK.KPW_ID (+);

```

## A4.24. View [Fall\_Kont\_Komp\_Wert\_D\_P\_Count]

```
1 CREATE OR REPLACE VIEW FALL_KONT_KOMP_WERT_D_P_COUNT AS
2 SELECT
3   KKW.FALL_ID                AS FALL_ID,
4   KKW.FALL_CODE             AS FALL_CODE,
5   KKW.NACHNAME              AS NACHNAME,
6   KKW.VORNAME               AS VORNAME,
7   KKW.GEBURTSDATUM         AS GEBURTSDATUM,
8   KKW.AUFNAHME_FA          AS AUFNAHME_FA,
9   KKW.ABGANG_FA            AS ABGANG_FA,
10  KKW.AUFNAHME_KH           AS AUFNAHME_KH,
11  KKW.ABGANG_KH             AS ABGANG_KH,
12  KKW.KKP_ID                AS KKP_ID,
13  KKW.K_ID                  AS K_ID,
14  KKW.KONTEXT_BEZEICHNUNG   AS KONTEXT_BEZEICHNUNG,
15  KKW.KONTEXT_KOMPLEX_RANG  AS KONTEXT_KOMPLEX_RANG,
16  KKW.KP_ID                 AS KP_ID,
17  KKW.KOMPLEX_BEZEICHNUNG   AS KOMPLEX_BEZEICHNUNG,
18  KKW.KPW_ID                AS KPW_ID,
19  KKW.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
20  KKW.KOMPLEX_WERT_RANG     AS KOMPLEX_WERT_RANG,
21  KKW.KOMPLEX_WERT         AS KOMPLEX_WERT,
22  KKW.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
23  KKW.KWT_ID                AS KWT_ID,
24  COUNT (KKW.DWE_D_ID) + COUNT (KKW.DWK_D_ID) AS D_COUNT,
25  COUNT (KKW.PWE_P_ID) + COUNT (KKW.PWK_P_ID) AS P_COUNT
26 FROM
27  FALL_KONT_KOMP_W_DW_PW_DWK_PWK  KKW
28 GROUP BY
29  KKW.FALL_ID,
30  KKW.FALL_CODE,
31  KKW.NACHNAME,
32  KKW.VORNAME,
33  KKW.GEBURTSDATUM,
34  KKW.AUFNAHME_FA,
35  KKW.ABGANG_FA,
36  KKW.AUFNAHME_KH,
37  KKW.ABGANG_KH,
38  KKW.KKP_ID,
39  KKW.K_ID,
40  KKW.KONTEXT_BEZEICHNUNG,
41  KKW.KONTEXT_KOMPLEX_RANG,
42  KKW.KP_ID,
43  KKW.KOMPLEX_BEZEICHNUNG,
44  KKW.KPW_ID,
45  KKW.KOMPLEX_WERT_BEZEICHNUNG,
46  KKW.KOMPLEX_WERT_RANG,
47  KKW.KOMPLEX_WERT,
48  KKW.KPW_EJN_ID_AUSWERTUNG,
49  KKW.KWT_ID;
```

## A4.25. View [Fall\_Kontext\_Komplex\_Wert\_Rang]

```
CREATE OR REPLACE VIEW FALL_KONTEXT_KOMPLEX_WERT_RANG AS
SELECT
DPC.FALL_ID AS FALL_ID,
DPC.FALL_CODE AS FALL_CODE,
DPC.NACHNAME AS NACHNAME,
DPC.VORNAME AS VORNAME,
DPC.GEBURTSDATUM AS GEBURTSDATUM,
DPC.AUFNAHME_FA AS AUFNAHME_FA,
DPC.ABGANG_FA AS ABGANG_FA,
DPC.AUFNAHME_KH AS AUFNAHME_KH,
DPC.ABGANG_KH AS ABGANG_KH,
DPC.KKP_ID AS KKP_ID,
DPC.K_ID AS K_ID,
DPC.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
DPC.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG,
DPC.KP_ID AS KP_ID,
DPC.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
DPC.KPW_ID AS KPW_ID,
DPC.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
DPC.KOMPLEX_WERT_RANG AS KOMPLEX_WERT_RANG,
DPC.KOMPLEX_WERT AS KOMPLEX_WERT,
DPC.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
DPC.KWT_ID AS KWT_ID
FROM
FALL_KONT_KOMP_WERT_D_P_COUNT DPC
WHERE
-----
-- es muss mindestens ein Element vorhanden sein
-----
(
DPC.KWT_ID = 1
AND
(DPC.D_COUNT + DPC.P_COUNT) > 0
)
-----
-- es muss mindestens ein diagnostisches UND ein prozedurales Element vorhanden sein
-----
OR
(
DPC.KWT_ID = 2
AND
DPC.D_COUNT > 0
AND
DPC.P_COUNT > 0
);
```

## A4.26. View [Fall\_Kontext\_Komplex\_Min\_Rang]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_KOMPLEX_MIN_RANG AS
2 SELECT
3   KWR.FALL_ID                AS FALL_ID,
4   KWR.FALL_CODE             AS FALL_CODE,
5   KWR.NACHNAME              AS NACHNAME,
6   KWR.VORNAME               AS VORNAME,
7   KWR.GEBURTSDATUM         AS GEBURTSDATUM,
8   KWR.AUFNAHME_FA          AS AUFNAHME_FA,
9   KWR.ABGANG_FA            AS ABGANG_FA,
10  KWR.AUFNAHME_KH           AS AUFNAHME_KH,
11  KWR.ABGANG_KH             AS ABGANG_KH,
12  KWR.KKP_ID                AS KKP_ID,
13  KWR.K_ID                  AS K_ID,
14  KWR.KONTEXT_BEZEICHNUNG   AS KONTEXT_BEZEICHNUNG,
15  KWR.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG,
16  KWR.KP_ID                 AS KP_ID,
17  KWR.KOMPLEX_BEZEICHNUNG  AS KOMPLEX_BEZEICHNUNG,
18  MIN (KWR.KOMPLEX_WERT_RANG) AS MINIMUM_RANG
19 FROM
20 FALL_KONTEXT_KOMPLEX_WERT_RANG KWR
21 GROUP BY
22   KWR.FALL_ID,
23   KWR.FALL_CODE,
24   KWR.NACHNAME,
25   KWR.VORNAME,
26   KWR.GEBURTSDATUM,
27   KWR.AUFNAHME_FA,
28   KWR.ABGANG_FA,
29   KWR.AUFNAHME_KH,
30   KWR.ABGANG_KH,
31   KWR.KKP_ID,
32   KWR.K_ID,
33   KWR.KONTEXT_BEZEICHNUNG,
34   KWR.KONTEXT_KOMPLEX_RANG,
35   KWR.KP_ID,
36   KWR.KOMPLEX_BEZEICHNUNG;
```

## A4.27. View [Fall\_Kontext\_Komplex]

```
CREATE OR REPLACE VIEW FALL_KONTEXT_KOMPLEX AS
-- -----
-- Non-Normal-Werte
-- -----
SELECT
KMR.FALL_ID           AS FALL_ID,
KMR.FALL_CODE        AS FALL_CODE,
KMR.NACHNAME         AS NACHNAME,
KMR.VORNAME          AS VORNAME,
KMR.GEBURTSDATUM     AS GEBURTSDATUM,
KMR.AUFNAHME_FA      AS AUFNAHME_FA,
KMR.ABGANG_FA        AS ABGANG_FA,
KMR.AUFNAHME_KH      AS AUFNAHME_KH,
KMR.ABGANG_KH        AS ABGANG_KH,
KMR.KKP_ID           AS KKP_ID,
KMR.K_ID             AS K_ID,
KMR.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
KMR.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG,
KMR.KP_ID            AS KP_ID,
KMR.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
KPW.WERT              AS KOMPLEX_WERT,
KPW.BEZEICHNUNG      AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.KPW_ID           AS KPW_ID
FROM
FALL_KONTEXT_KOMPLEX_MIN_RANG KMR,
KOMPLEX                       KPL,
KOMPLEX_WERT                   KPW
WHERE
KMR.KP_ID = KPW.KP_ID
AND
KMR.KP_ID = KPL.KP_ID
AND
KMR.MINIMUM_RANG = KPW.RANG
UNION
-- -----
-- Normal-Werte
-- -----
SELECT
FAL.FALL_ID,
FAL.CODE,
FAL.NACHNAME,
FAL.VORNAME,
FAL.GEBURTSDATUM,
FAL.AUFNAHME_FA,
FAL.ABGANG_FA,
FAL.AUFNAHME_KH,
FAL.ABGANG_KH,
KKP.KKP_ID,
KKP.K_ID,
KPL.BEZEICHNUNG,
KKP.RANG,
KKP.KP_ID,
KPL.BEZEICHNUNG,
KPL.NORMAL_WERT,
KPL.NORMAL_TEXT,
NULL
FROM
FALL          FAL,
KONTEXT_KOMPLEX KKP,
KONTEXT       KON,
KOMPLEX       KPL
WHERE
KKP.K_ID = KON.K_ID
AND
KKP.KP_ID = KPL.KP_ID
```

```
66 AND
67 (
68 FAL.FALL_ID,
69 KKP.KP_ID
70 )
71 NOT IN
72 (
73 SELECT
74 KMR.FALL_ID,
75 KMR.KP_ID
76 FROM
77 FALL_KONTEXT_KOMPLEX_MIN_RANG KMR
78 WHERE
79 KMR.FALL_ID = FAL.FALL_ID
80 AND
81 KMR.KP_ID = KKP.KP_ID
82 );
```

## A4.28. View [Fall\_ICD\_OP]

```
CREATE OR REPLACE VIEW FALL_ICD_OP AS
SELECT
FAL.FALL_ID AS FALL_ID,
FAL.CODE AS FALL_CODE,
FAL.NACHNAME AS NACHNAME,
FAL.VORNAME AS VORNAME,
FAL.GEBURTSDATUM AS GEBURTSDATUM,
FAL.AUFNAHME_FA AS AUFNAHME_FA,
FAL.ABGANG_FA AS ABGANG_FA,
FAL.AUFNAHME_KH AS AUFNAHME_KH,
FAL.ABGANG_KH AS ABGANG_KH,
FAL.G_ID AS G_ID,
FAL.BA_ID AS BA_ID,
BEA.NAME || ', ' || BEA.VORNAME AS BEARBEITER,
FAL.KIS_ID AS KIS_ID,
KIS.BEZEICHNUNG AS KIS_BEZEICHNUNG,
FAL.FA_ID AS FA_ID,
FAB.BEZEICHNUNG AS FACHABTEILUNG_BEZEICHNUNG,
FOP.FALLOP_ID AS FALLOP_ID,
FOP.BEGINN AS OP_BEGINN,
FOP.ENDE AS OP_ENDE,
NULL AS FALLICD_ID,
FOI.FOPICD_ID AS FOPICD_ID,
FOI.ICD_ID AS ICD_ID,
ICD.CODE AS ICD_CODE,
ICD.KURZTEXT AS ICD_KURZTEXT,
IVE.ICDVER_ID AS ICDVER_ID,
IVE.BEZEICHNUNG AS ICD_VERSION_BEZEICHNUNG,
FOI.ICDKAT_ID AS ICDKAT_ID,
IKA.BEZEICHNUNG AS ICD_KATEGORIE_BEZEICHNUNG,
FOI.ICDLOK_ID AS ICDLOK_ID,
ILO.BEZEICHNUNG AS ICD_LOKALISATION_BEZEICHNUNG,
FOP.BEGINN AS DATUM
FROM
FALL FAL,
FALL_OP FOP,
FALL_OP_ICD FOI,
ICD ICD,
ICD_VERSION IVE,
ICD_KATEGORIE IKA,
ICD_LOKALISATION ILO,
KIS KIS,
FACHABTEILUNG FAB,
BEARBEITER BEA
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
FOP.FALLOP_ID = FOI.FALLOP_ID
AND
FOI.ICD_ID = ICD.ICD_ID
AND
FOI.ICDKAT_ID = IKA.ICDKAT_ID
AND
FOI.ICDLOK_ID = ILO.ICDLOK_ID
AND
FAL.KIS_ID = KIS.KIS_ID
AND
FAL.FA_ID = FAB.FA_ID
AND
FAL.BA_ID = BEA.BA_ID
AND
ICD.ICDVER_ID = IVE.ICDVER_ID;
```

## A4.29. View [Fall\_Kontext\_Diag\_Wert\_ICD\_Op]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAG_WERT_ICD_OP AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM     AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA      AS AUFNAHME_FA,
9 FAL.ABGANG_FA        AS ABGANG_FA,
10 FAL.AUFNAHME_KH      AS AUFNAHME_KH,
11 FAL.ABGANG_KH        AS ABGANG_KH,
12 FOP.FALLOP_ID       AS FALLOP_ID,
13 FOP.CODE             AS OP_CODE,
14 FOP.BEGINN          AS OP_BEGINN,
15 FOP.ENDE            AS OP_ENDE,
16 DIA.D_ID            AS D_ID,
17 DIA.BEZEICHNUNG     AS DIAGNOSE_BEZEICHNUNG,
18 DIW.DW_ID           AS DW_ID,
19 DIW.BEZEICHNUNG     AS DIAGNOSE_WERT_BEZEICHNUNG,
20 DIW.WERT            AS DIAGNOSE_WERT,
21 DIW.RANG            AS DIAGNOSE_WERT_RANG,
22 DIW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
23 KDI.KD_ID           AS KD_ID,
24 KON.K_ID            AS K_ID,
25 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
26 KDI.RANG            AS KONTEXT_RANG
27 FROM
28 FALL                FAL,
29 FALL_OP            FOP,
30 DIAGNOSE           DIA,
31 DIAGNOSE_WERT     DIW,
32 KONTEXT_DIAGNOSE  KDI,
33 KONTEXT            KON
34 WHERE
35 FAL.FALL_ID = FOP.FALL_ID
36 AND
37 DIA.D_ID = DIW.D_ID
38 AND
39 DIA.D_ID = KDI.D_ID
40 AND
41 KDI.K_ID = KON.K_ID
42 -----
43 -- Auflösung eines definierten oder eben
44 -- nicht definierten Seitenbezugs:
45 -- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
46 -- vernachlässigt werden
47 -- 2. Fall: Seitenlokalisierung ist definiert UND muss dann mit der des Fall-bezogenen
48 -- ICD-Codes übereinstimmen
49 -----
50 AND
51 (
52 DIW.DW_ID IN
53 (
54 SELECT
55 DWI.DW_ID
56 FROM
57 DIAGNOSE_WERT_ICD DWI,
58 FALL_ICD_OP       FIC
59 WHERE
60 DWI.DW_ID = DIW.DW_ID
61 AND
62 FIC.FALL_ID = FAL.FALL_ID
63 AND
64 FIC.FALLOP_ID = FOP.FALLOP_ID
65 AND
```



```

FIC.ICD_ID = DWI.ICD_ID                                66
AND                                                    67
DWI.DWI_ID NOT IN                                    68
(                                                    69
SELECT                                              70
DWL.DWI_ID                                          71
FROM                                              72
DIAGNOSE_WERT_ICD_LOK   DWL                      73
WHERE                                              74
DWL.DWI_ID = DWI.DWI_ID                          75
)                                                    76
)                                                    77
OR                                                    78
DIW.DW_ID IN                                        79
(                                                    80
SELECT                                              81
DWI.DW_ID                                          82
FROM                                              83
DIAGNOSE_WERT_ICD           DWI,                 84
DIAGNOSE_WERT_ICD_LOK     DWL,                 85
FALL_ICD_OP                FIC                 86
WHERE                                              87
DWI.DW_ID = DIW.DW_ID                            88
AND                                                89
FIC.FALL_ID = FAL.FALL_ID                        90
AND                                                91
FIC.FALLOP_ID = FOP.FALLOP_ID                   92
AND                                                93
FIC.ICD_ID = DWI.ICD_ID                          94
AND                                                95
DWI.DWI_ID = DWL.DWI_ID                          96
AND                                                97
FIC.ICDLOK_ID = DWL.ICDLOK_ID                   98
)                                                    99
)                                                    100
-- -----                                         101
-- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs 102
-- -----                                         103
-- -----                                         104
-- Kategorie-Auswertung:                             105
-- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und auszuschließenden 106
-- ICD-Kategorien                                    107
-- -----                                         108
AND                                                    109
(                                                    110
DIW.DW_ID IN                                        111
(                                                    112
SELECT                                              113
DWI.DW_ID                                          114
FROM                                              115
DIAGNOSE_WERT_ICD           DWI,                 116
FALL_ICD_OP                FIC,                 117
PHASE_ICD_KATEGORIE       PIK,                 118
KONTEXT_DIAGNOSE_PHASE    KDP                 119
WHERE                                              120
DWI.DW_ID = DIW.DW_ID                            121
AND                                                122
FIC.FALL_ID = FAL.FALL_ID                        123
AND                                                124
FIC.FALLOP_ID = FOP.FALLOP_ID                   125
AND                                                126
DWI.ICD_ID = FIC.ICD_ID                          127
AND                                                128
PIK.ICDKAT_ID = FIC.ICDKAT_ID                   129
AND                                                130
KDP.KD_ID = KDI.KD_ID                            131
AND                                                132
PIK.PH_ID = KDP.PH_ID                            133

```

```

134 AND
135 PIK.PHZT_ID = 1 -- EINSCHLUSS!
136 )
137 AND
138 DIW.DW_ID NOT IN
139 (
140 SELECT
141 DWI.DW_ID
142 FROM
143 DIAGNOSE_WERT_ICD      DWI,
144 FALL_ICD_OP            FIC,
145 PHASE_ICD_KATEGORIE   PIK,
146 KONTEXT_DIAGNOSE_PHASE KDP
147 WHERE
148 DWI.DW_ID = DIW.DW_ID
149 AND
150 FIC.FALL_ID = FAL.FALL_ID
151 AND
152 FIC.FALLOP_ID = FOP.FALLOP_ID
153 AND
154 DWI.ICD_ID = FIC.ICD_ID
155 AND
156 PIK.ICDKAT_ID = FIC.ICDKAT_ID
157 AND
158 KDP.KD_ID = KDI.KD_ID
159 AND
160 PIK.PH_ID = KDP.PH_ID
161 AND
162 PIK.PHZT_ID = 2 -- AUSSCHLUSS!
163 )
164 )
165 -----
166 -- Ende: Kategorie-Auswertung
167 -----
168 -----
169 -- Auswertung Zeitpunkt:
170 -- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf,
171 -- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten
172 -- Phase
173 -----
174 AND
175 (
176 -----
177 -- Auswertung Zeitpunkt -
178 -- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu)
179 -- Keine zeitliche Einschränkung, da durch die ICD-Kategorie bereits eine eindeutige
180 -- Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!)
181 -----
182 DIW.DW_ID IN
183 (
184 SELECT
185 DWI.DW_ID
186 FROM
187 DIAGNOSE_WERT_ICD      DWI,
188 FALL_ICD_OP            FIC,
189 KONTEXT_DIAGNOSE_PHASE KDP
190 WHERE
191 DWI.DW_ID = DIW.DW_ID
192 AND
193 FIC.FALL_ID = FAL.FALL_ID
194 AND
195 FIC.FALLOP_ID = FOP.FALLOP_ID
196 AND
197 DWI.ICD_ID = FIC.ICD_ID
198 AND
199 KDP.KD_ID = KDI.KD_ID
200 AND
201 KDP.PH_ID IN (1, 2, 5, 6)

```

```

)
-----
-- Auswertung Zeitpunkt -
-- Phase: postoperativ
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen oder hinter diesem liegen
-----
OR
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_OP            FIC,
KONTEXT_DIAGNOSE_PHASE KDP
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
FIC.FALLOP_ID = FOP.FALLOP_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
KDP.KD_ID = KDI.KD_ID
AND
KDP.PH_ID IN (7)
AND
FIC.DATUM >=
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Haupteingriff
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen
-----
OR
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_OP            FIC,
KONTEXT_DIAGNOSE_PHASE KDP
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
FIC.FALLOP_ID = FOP.FALLOP_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
KDP.KD_ID = KDI.KD_ID
AND
KDP.PH_ID IN (3)
AND
FIC.DATUM =

```

```

270 (
271 SELECT
272 MIN (FOP.BEGINN)
273 FROM
274 FALL_OP FOP
275 WHERE
276 FOP.FALL_ID = FAL.FALL_ID
277 )
278 )
279 -- -----
280 -- Auswertung Zeitpunkt -
281 -- Phase: intraop. Folgeeingriff(e)
282 -- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs liegen
283 -- -----
284 OR
285 DIW.DW_ID IN
286 (
287 SELECT
288 DWI.DW_ID
289 FROM
290 DIAGNOSE_WERT_ICD          DWI,
291 FALL_ICD_OP                FIC,
292 KONTEXT_DIAGNOSE_PHASE   KDP
293 WHERE
294 DWI.DW_ID = DIW.DW_ID
295 AND
296 FIC.FALL_ID = FAL.FALL_ID
297 AND
298 FIC.FALLOP_ID = FOP.FALLOP_ID
299 AND
300 DWI.ICD_ID = FIC.ICD_ID
301 AND
302 KDP.KD_ID = KDI.KD_ID
303 AND
304 KDP.PH_ID IN (4)
305 AND
306 FIC.DATUM >
307 (
308 SELECT
309 MIN (FOP.BEGINN)
310 FROM
311 FALL_OP FOP
312 WHERE
313 FOP.FALL_ID = FAL.FALL_ID
314 )
315 )
316 -- -----
317 -- Ende: Auswertung Zeitpunkt
318 -- -----
319 );

```

## A4.30. View [Fall\_Kontext\_Diag\_W\_Non\_ICD\_OP]

```
CREATE OR REPLACE VIEW FALL_KONTEXT_DIAG_W_NON_ICD_OP AS
SELECT
FAL.FALL_ID      AS FALL_ID,
FAL.CODE        AS FALL_CODE,
FAL.NACHNAME    AS NACHNAME,
FAL.VORNAME     AS VORNAME,
FAL.GEBURTSDATUM AS GEBURTSDATUM,
FAL.AUFNAHME_FA AS AUFNAHME_FA,
FAL.ABGANG_FA   AS ABGANG_FA,
FAL.AUFNAHME_KH AS AUFNAHME_KH,
FAL.ABGANG_KH   AS ABGANG_KH,
FOP.FALLOP_ID  AS FALLOP_ID,
FOP.CODE       AS OP_CODE,
FOP.BEGINN     AS OP_BEGINN,
FOP.ENDE       AS OP_ENDE,
DIA.D_ID       AS D_ID,
DIA.BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
NULL           AS DW_ID,
DIA.NORMAL_TEXT AS DIAGNOSE_WERT_BEZEICHNUNG,
DIA.NORMAL_WERT AS DIAGNOSE_WERT,
9999          AS DIAGNOSE_WERT_RANG,
1             AS EJM_ID_AUSWERTUNG,
KDI.KD_ID      AS KD_ID,
KON.K_ID       AS K_ID,
KON.BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
KDI.RANG       AS KONTEXT_RANG
FROM
FALL           FAL,
FALL_OP       FOP,
DIAGNOSE      DIA,
KONTEXT_DIAGNOSE KDI,
KONTEXT       KON
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
DIA.D_ID = KDI.D_ID
AND
KDI.K_ID = KON.K_ID
AND
DIA.D_ID IN -- Diagnosen müssen sich prinzipiell aus ICD-Codes ableiten lassen!
(
SELECT
DIW.D_ID
FROM
DIAGNOSE_WERT      DIW,
DIAGNOSE_WERT_ICD DWI
WHERE
DIW.D_ID = DIA.D_ID
AND
DIW.DW_ID = DWI.DW_ID
)
AND
DIA.D_ID NOT IN
(
SELECT
FDI.D_ID
FROM
FALL_KONTEXT_DIAG_WERT_ICD_OP FDI
WHERE
FDI.D_ID = DIA.D_ID
AND
FDI.FALL_ID = FAL.FALL_ID
AND
FDI.FALLOP_ID = FOP.FALLOP_ID
);
```

## A4.31. View [Fall\_Kontext\_Diag\_Wert\_Komb\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAG_WERT_KOMB_OP AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA      AS AUFNAHME_FA,
9 FAL.ABGANG_FA        AS ABGANG_FA,
10 FAL.AUFNAHME_KH      AS AUFNAHME_KH,
11 FAL.ABGANG_KH        AS ABGANG_KH,
12 FOP.FALLOP_ID       AS FALLOP_ID,
13 FOP.CODE             AS OP_CODE,
14 FOP.BEGINN          AS OP_BEGINN,
15 FOP.ENDE            AS OP_ENDE,
16 DIA.D_ID            AS D_ID,
17 DIA.BEZEICHNUNG     AS DIAGNOSE_BEZEICHNUNG,
18 DWK.DWK_ID          AS DWK_ID,
19 DWK.BEZEICHNUNG     AS DIAGNOSE_WERT_BEZEICHNUNG,
20 DWK.WERT            AS DIAGNOSE_WERT,
21 DWK.RANG            AS DIAGNOSE_WERT_RANG,
22 DWK.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
23 KDI.KD_ID           AS KD_ID,
24 KON.K_ID            AS K_ID,
25 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
26 KDI.RANG            AS KONTEXT_RANG
27 FROM
28 FALL                FAL,
29 FALL_OP             FOP,
30 DIAGNOSE            DIA,
31 DIAGNOSE_WERT_KOMBINATION DWK,
32 DIAGNOSE_WERT_KOMB_ELEM_COUNT DKE,
33 KONTEXT_DIAGNOSE    KDI,
34 KONTEXT             KON
35 WHERE
36 FAL.FALL_ID = FOP.FALL_ID
37 AND
38 DIA.D_ID = DWK.D_ID
39 AND
40 DWK.DWK_ID = DKE.DWK_ID
41 AND
42 DIA.D_ID = KDI.D_ID
43 AND
44 KDI.K_ID = KON.K_ID
45 AND
46 (
47 (
48 -- Kombinationstyp: alle Diagnosen sind obligat!
49 DWK.DWKT_ID = 1
50 AND
51 DKE.ZAHL_ELEMENTE =
52 (
53 SELECT
54 COUNT (DKD.DWKDW_ID)
55 FROM
56 FALL_KONTEXT_DIAG_WERT_ICD_OP FDI,
57 DIAG_WERT_KOMB_DIAG_WERT     DKD
58 WHERE
59 FDI.FALL_ID = FAL.FALL_ID
60 AND
61 FDI.FALLOP_ID = FOP.FALLOP_ID
62 AND
63 FDI.DW_ID = DKD.DW_ID
64 AND
65 DKD.DWK_ID = DWK.DWK_ID
```

)	66
)	67
OR	68
(	69
-- Kombinationstyp: mindestens 2 Diagnosen sind obligat!	70
DWK.DWKT_ID = 2	71
AND	72
1 <	73
(	74
SELECT	75
COUNT (DKD.DWKDW_ID)	76
FROM	77
FALL_KONTEXT_DIAG_WERT_ICD_OP FDI,	78
DIAG_WERT_KOMB_DIAG_WERT DKD	79
WHERE	80
FDI.FALL_ID = FAL.FALL_ID	81
AND	82
FDI.FALLOP_ID = FOP.FALLOP_ID	83
AND	84
FDI.DW_ID = DKD.DW_ID	85
AND	86
DKD.DWK_ID = DWK.DWK_ID	87
)	88
)	89
);	90

## A4.32. View [Fall\_Kontext\_Diagnose\_Rang\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAGNOSE_RANG_OP AS
2 SELECT
3 FALL_ID                AS FALL_ID,
4 FALL_CODE              AS FALL_CODE,
5 NACHNAME              AS NACHNAME,
6 VORNAME               AS VORNAME,
7 GEBURTSDATUM         AS GEBURTSDATUM,
8 AUFNAHME_FA          AS AUFNAHME_FA,
9 ABGANG_FA             AS ABGANG_FA,
10 AUFNAHME_KH          AS AUFNAHME_KH,
11 ABGANG_KH            AS ABGANG_KH,
12 FALLOP_ID            AS FALLOP_ID,
13 OP_CODE              AS OP_CODE,
14 OP_BEGINN            AS OP_BEGINN,
15 OP_ENDE              AS OP_ENDE,
16 D_ID                 AS D_ID,
17 DW_ID                AS DW_ID,
18 NULL                 AS DWK_ID,
19 DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
20 DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
21 DIAGNOSE_WERT        AS DIAGNOSE_WERT,
22 DIAGNOSE_WERT_RANG   AS DIAGNOSE_WERT_RANG,
23 1                    AS EJN_ID_AUSWERTUNG,
24 KD_ID                AS KD_ID,
25 K_ID                 AS K_ID,
26 KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
27 KONTEXT_RANG         AS KONTEXT_RANG
28 FROM
29 FALL_KONTEXT_DIAG_W_NON_ICD_OP
30 UNION
31 SELECT
32 FALL_ID,
33 FALL_CODE,
34 NACHNAME,
35 VORNAME,
36 GEBURTSDATUM,
37 AUFNAHME_FA,
38 ABGANG_FA,
39 AUFNAHME_KH,
40 ABGANG_KH,
41 FALLOP_ID,
42 OP_CODE,
43 OP_BEGINN,
44 OP_ENDE,
45 D_ID,
46 DW_ID,
47 NULL,
48 DIAGNOSE_BEZEICHNUNG,
49 DIAGNOSE_WERT_BEZEICHNUNG,
50 DIAGNOSE_WERT,
51 DIAGNOSE_WERT_RANG,
52 EJN_ID_AUSWERTUNG,
53 KD_ID,
54 K_ID,
55 KONTEXT_BEZEICHNUNG,
56 KONTEXT_RANG
57 FROM
58 FALL_KONTEXT_DIAG_WERT_ICD_OP
59 UNION
60 SELECT
61 FALL_ID,
62 FALL_CODE,
63 NACHNAME,
64 VORNAME,
65 GEBURTSDATUM,
```



AUFNAHME_FA,	66
ABGANG_FA,	67
AUFNAHME_KH,	68
ABGANG_KH,	69
FALLOP_ID,	70
OP_CODE,	71
OP_BEGINN,	72
OP_ENDE,	73
D_ID,	74
NULL,	75
DWK_ID,	76
DIAGNOSE_BEZEICHNUNG,	77
DIAGNOSE_WERT_BEZEICHNUNG,	78
DIAGNOSE_WERT,	79
DIAGNOSE_WERT_RANG,	80
EJN_ID_AUSWERTUNG,	81
KD_ID,	82
K_ID,	83
KONTEXT_BEZEICHNUNG,	84
KONTEXT_RANG	85
FROM	86
FALL_KONTEXT_DIAG_WERT_KOMB_OP;	87

### A4.33. View [Fall\_Kontext\_Diagnose\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_DIAGNOSE_OP AS
2 SELECT
3 T1.FALL_ID AS FALL_ID,
4 T1.FALL_CODE AS FALL_CODE,
5 T1.NACHNAME AS NACHNAME,
6 T1.VORNAME AS VORNAME,
7 T1.GEBURTSDATUM AS GEBURTSDATUM,
8 T1.AUFNAHME_FA AS AUFNAHME_FA,
9 T1.ABGANG_FA AS ABGANG_FA,
10 T1.AUFNAHME_KH AS AUFNAHME_KH,
11 T1.ABGANG_KH AS ABGANG_KH,
12 T1.FALLOP_ID AS FALLOP_ID,
13 T1.OP_CODE AS OP_CODE,
14 T1.OP_BEGINN AS OP_BEGINN,
15 T1.OP_ENDE AS OP_ENDE,
16 T1.D_ID AS D_ID,
17 T1.DW_ID AS DW_ID,
18 T1.DWK_ID AS DWK_ID,
19 T1.DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
20 T1.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
21 T1.DIAGNOSE_WERT AS DIAGNOSE_WERT,
22 T1.DIAGNOSE_WERT_RANG AS DIAGNOSE_WERT_RANG,
23 T1.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
24 T1.KD_ID AS KD_ID,
25 T1.K_ID AS K_ID,
26 T1.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
27 T1.KONTEXT_RANG AS KONTEXT_RANG
28 FROM
29 FALL_KONTEXT_DIAGNOSE_RANG_OP T1
30 WHERE
31 T1.EJN_ID_AUSWERTUNG = 1
32 AND
33 (
34 T1.FALL_ID,
35 T1.FALLOP_ID,
36 T1.D_ID,
37 T1.DIAGNOSE_WERT_RANG
38 )
39 IN
40 (
41 SELECT
42 T2.FALL_ID,
43 T2.FALLOP_ID,
44 T2.D_ID,
45 MIN (T2.DIAGNOSE_WERT_RANG)
46 FROM
47 FALL_KONTEXT_DIAGNOSE_RANG_OP T2
48 WHERE
49 T2.FALL_ID = T1.FALL_ID
50 AND
51 T2.D_ID = T1.D_ID
52 GROUP BY
53 T2.FALL_ID,
54 T2.FALLOP_ID,
55 T2.D_ID
56 )
57 ;
```

## A4.34. View [Fall\_OPS\_OP]

```
CREATE OR REPLACE VIEW FALL_OPS_OP AS
SELECT
FAL.FALL_ID AS FALL_ID,
FAL.CODE AS FALL_CODE,
FAL.NACHNAME AS NACHNAME,
FAL.VORNAME AS VORNAME,
FAL.GEBURTSDATUM AS GEBURTSDATUM,
FAL.AUFNAHME_FA AS AUFNAHME_FA,
FAL.ABGANG_FA AS ABGANG_FA,
FAL.AUFNAHME_KH AS AUFNAHME_KH,
FAL.ABGANG_KH AS ABGANG_KH,
FAL.G_ID AS G_ID,
FAL.BA_ID AS BA_ID,
BEA.NAME || ', ' || BEA.VORNAME AS BEARBEITER,
FAL.KIS_ID AS KIS_ID,
KIS.BEZEICHNUNG AS KIS_BEZEICHNUNG,
FAL.FA_ID AS FA_ID,
FAB.BEZEICHNUNG AS FACHABTEILUNG_BEZEICHNUNG,
FOP.FALLOP_ID AS FALLOP_ID,
FOP.BEGINN AS OP_BEGINN,
FOP.ENDE AS OP_ENDE,
NULL AS FALLOPS_ID,
FOO.FOPOPS_ID AS FOPOPS_ID,
FOO.OPS_ID AS OPS_ID,
OPS.CODE AS OPS_CODE,
OPS.KURZTEXT AS OPS_KURZTEXT,
OVE.OPSVER_ID AS OPSVER_ID,
OVE.BEZEICHNUNG AS OPS_VERSION_BEZEICHNUNG,
FOO.OPSKAT_ID AS OPSKAT_ID,
OKA.BEZEICHNUNG AS OPS_KATEGORIE_BEZEICHNUNG,
FOO.OPSLOK_ID AS OPSLOK_ID,
OLO.BEZEICHNUNG AS OPS_LOKALISATION_BEZEICHNUNG,
FOP.BEGINN AS DATUM
FROM
FALL
FALL_OP
FALL_OP_OPS301
OPS301
OPS301_VERSION
OPS301_KATEGORIE
OPS301_LOKALISATION
KIS
FACHABTEILUNG
BEARBEITER
BEA
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
FOP.FALLOP_ID = FOO.FALLOP_ID
AND
FOO.OPS_ID = OPS.OPS_ID
AND
FOO.OPSKAT_ID = OKA.OPSKAT_ID
AND
FOO.OPSLOK_ID = OLO.OPSLOK_ID
AND
FAL.KIS_ID = KIS.KIS_ID
AND
FAL.FA_ID = FAB.FA_ID
AND
FAL.BA_ID = BEA.BA_ID
AND
OPS.OPSVER_ID = OVE.OPSVER_ID;
```

## A4.35. View [Fall\_Kontext\_Proz\_Wert\_OPS\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZ_WERT_OPS_OP AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA      AS AUFNAHME_FA,
9 FAL.ABGANG_FA        AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH       AS ABGANG_KH,
12 FOP.FALLOP_ID       AS FALLOP_ID,
13 FOP.CODE             AS OP_CODE,
14 FOP.BEGINN          AS OP_BEGINN,
15 FOP.ENDE            AS OP_ENDE,
16 PRO.P_ID            AS P_ID,
17 PRO.BEZEICHNUNG     AS PROZEDUR_BEZEICHNUNG,
18 PRW.PW_ID           AS PW_ID,
19 PRW.BEZEICHNUNG     AS PROZEDUR_WERT_BEZEICHNUNG,
20 PRW.WERT            AS PROZEDUR_WERT,
21 PRW.RANG            AS PROZEDUR_WERT_RANG,
22 PRW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
23 KPR.KP_ID           AS KP_ID,
24 KON.K_ID            AS K_ID,
25 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
26 KPR.RANG            AS KONTEXT_RANG
27 FROM
28 FALL                FAL,
29 FALL_OP             FOP,
30 PROZEDUR            PRO,
31 PROZEDUR_WERT       PRW,
32 KONTEXT_PROZEDUR    KPR,
33 KONTEXT             KON
34 WHERE
35 FAL.FALL_ID = FOP.FALL_ID
36 AND
37 PRO.P_ID = PRW.P_ID
38 AND
39 PRO.P_ID = KPR.P_ID
40 AND
41 KPR.K_ID = KON.K_ID
42 -- -----
43 -- Auflösung eines definierten oder eben
44 -- nicht definierten Seitenbezugs:
45 -- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
46 -- vernachlässigt werden
47 -- 2. Fall:
48 -- Seitenlokalisierung ist definiert UND muss dann mit der des Fall-bezogenen
49 -- ICD-Codes übereinstimmen
50 -- -----
51 AND
52 (
53 PRW.PW_ID IN
54 (
55 SELECT
56 PWO.PW_ID
57 FROM
58 PROZEDUR_WERT_OPS PWO,
59 FALL_OPS_OP       FOO
60 WHERE
61 FOO.FALLOP_ID = FOP.FALLOP_ID
62 AND
63 PWO.PW_ID = PRW.PW_ID
64 AND
65 FOO.FALL_ID = FAL.FALL_ID
```

```

AND
FOO.OPS_ID = PWO.OPS_ID
AND
PWO.PWO_ID NOT IN
(
SELECT
PWL.PWO_ID
FROM
PROZEDUR_WERT_OPS_LOK PWL
WHERE
PWL.PWO_ID = PWO.PWO_ID
)
)
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS PWO,
PROZEDUR_WERT_OPS_LOK PWL,
FALL_OPS_OP FOO
WHERE
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.OPS_ID = PWO.OPS_ID
AND
PWO.PWO_ID = PWL.PWO_ID
AND
FOO.OPSLOK_ID = PWL.OPSLOK_ID
)
)
-- -----
-- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs
-- -----
-- Kategorie-Auswertung:
-- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und auszuschließenden
-- ICD-Kategorien
-- -----
AND
(
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS PWO,
FALL_OPS_OP FOO,
PHASE_OPS301_KATEGORIE POK,
KONTEXT_PROZEDUR_PHASE KPP
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.OPS_ID = FOO.OPS_ID
AND
POK.OPSKAT_ID = FOO.OPSKAT_ID
AND
KPP.KP_ID = KPR.KP_ID
AND

```

```

134 POK.PH_ID = KPP.PH_ID
135 AND
136 POK.PHZT_ID = 1 -- EINSCHLUSS!
137 )
138 AND
139 PRW.PW_ID NOT IN
140 (
141 SELECT
142 PWO.PW_ID
143 FROM
144 PROZEDUR_WERT_OPS      PWO,
145 FALL_OPS_OP           FOO,
146 PHASE_OPS301_KATEGORIE POK,
147 KONTEXT_PROZEDUR_PHASE KPP
148 WHERE
149 PWO.PW_ID = PRW.PW_ID
150 AND
151 FOO.FALL_ID = FAL.FALL_ID
152 AND
153 FOO.FALLOP_ID = FOP.FALLOP_ID
154 AND
155 PWO.OPS_ID = FOO.OPS_ID
156 AND
157 POK.OPSKAT_ID = FOO.OPSKAT_ID
158 AND
159 KPP.KP_ID = KPR.KP_ID
160 AND
161 POK.PH_ID = KPP.PH_ID
162 AND
163 POK.PHZT_ID = 2 -- AUSSCHLUSS!
164 )
165 )
166 -- -----
167 -- Ende: Kategorie-Auswertung
168 -- -----
169 -- -----
170 -- Auswertung Zeitpunkt:
171 -- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf,
172 -- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten
173 -- Phase
174 -- -----
175 AND
176 (
177 -- -----
178 -- Auswertung Zeitpunkt -
179 -- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu)
180 -- Keine zeitliche Einschränkung, da durch die ICD-Kategorie bereits eine
181 -- eindeutige Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!)
182 -- -----
183 PRW.PW_ID IN
184 (
185 SELECT
186 PWO.PW_ID
187 FROM
188 PROZEDUR_WERT_OPS      PWO,
189 FALL_OPS_OP           FOO,
190 KONTEXT_PROZEDUR_PHASE KPP
191 WHERE
192 PWO.PW_ID = PRW.PW_ID
193 AND
194 FOO.FALL_ID = FAL.FALL_ID
195 AND
196 FOO.FALLOP_ID = FOP.FALLOP_ID
197 AND
198 PWO.OPS_ID = FOO.OPS_ID
199 AND
200 KPP.KP_ID = KPR.KP_ID
201 AND

```

```

KPP.PH_ID IN (1, 2, 5, 6)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: postoperativ
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen oder hinter diesem liegen
-----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_OP           FOO,
KONTEXT_PROZEDUR_PHASE KPP
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.OPS_ID = FOO.OPS_ID
AND
KPP.KP_ID = KPR.KP_ID
AND
KPP.PH_ID IN (7)
AND
FOO.DATUM >=
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Haupteingriff
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen
-----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_OP           FOO,
KONTEXT_PROZEDUR_PHASE KPP
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.OPS_ID = FOO.OPS_ID
AND
KPP.KP_ID = KPR.KP_ID
AND
KPP.PH_ID IN (3)
AND

```

```

270 FOO.DATUM =
271 (
272 SELECT
273 MIN (FOP.BEGINN)
274 FROM
275 FALL_OP FOP
276 WHERE
277 FOP.FALL_ID = FAL.FALL_ID
278 )
279 )
280 -----
281 -- Auswertung Zeitpunkt -
282 -- Phase: intraop. Folgeeingriff(e)
283 -- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs liegen
284 -- -----
285 OR
286 PRW.PW_ID IN
287 (
288 SELECT
289 PWO.PW_ID
290 FROM
291 PROZEDUR_WERT_OPS          PWO,
292 FALL_OPS_OP              FOO,
293 KONTEXT_PROZEDUR_PHASE  KPP
294 WHERE
295 PWO.PW_ID = PRW.PW_ID
296 AND
297 FOO.FALL_ID = FAL.FALL_ID
298 AND
299 PWO.OPS_ID = FOO.OPS_ID
300 AND
301 KPP.KP_ID = KPR.KP_ID
302 AND
303 KPP.PH_ID IN (4)
304 AND
305 FOO.DATUM >
306 (
307 SELECT
308 MIN (FOP.BEGINN)
309 FROM
310 FALL_OP FOP
311 WHERE
312 FOP.FALL_ID = FAL.FALL_ID
313 )
314 )
315 -----
316 -- Ende: Auswertung Zeitpunkt
317 -- -----
318 );

```



## A4.36. View [Fall\_Kontext\_Proz\_W\_Non\_OPS\_OP]

```
CREATE OR REPLACE VIEW FALL_KONTEXT_PROZ_W_NON_OPS_OP AS
SELECT
FAL.FALL_ID      AS FALL_ID,
FAL.CODE        AS FALL_CODE,
FAL.NACHNAME    AS NACHNAME,
FAL.VORNAME     AS VORNAME,
FAL.GEBURTSDATUM AS GEBURTSDATUM,
FAL.AUFNAHME_FA AS AUFNAHME_FA,
FAL.ABGANG_FA   AS ABGANG_FA,
FAL.AUFNAHME_KH AS AUFNAHME_KH,
FAL.ABGANG_KH   AS ABGANG_KH,
FOP.FALLOP_ID  AS FALLOP_ID,
FOP.CODE       AS OP_CODE,
FOP.BEGINN    AS OP_BEGINN,
FOP.ENDE      AS OP_ENDE,
PRO.P_ID      AS P_ID,
PRO.BEZEICHNUNG AS PROZEDUR_BEZEICHNUNG,
NULL         AS PW_ID,
PRO.NORMAL_TEXT AS PROZEDUR_WERT_BEZEICHNUNG,
PRO.NORMAL_WERT AS PROZEDUR_WERT,
9999        AS PROZEDUR_WERT_RANG,
1           AS EJM_ID_AUSWERTUNG,
KPR.KP_ID     AS KP_ID,
KON.K_ID      AS K_ID,
KON.BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
KPR.RANG      AS KONTEXT_RANG
FROM
FALL          FAL,
FALL_OP      FOP,
PROZEDUR     PRO,
KONTEXT_PROZEDUR KPR,
KONTEXT      KON
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
PRO.P_ID = KPR.P_ID
AND
KPR.K_ID = KON.K_ID
AND
PRO.P_ID IN -- Prozeduren müssen sich prinzipiell aus OPS301-Codes ableiten lassen!
(
SELECT
PRW.P_ID
FROM
PROZEDUR_WERT      PRW,
PROZEDUR_WERT_OPS PWO
WHERE
PRW.P_ID = PRO.P_ID
AND
PRW.PW_ID = PWO.PW_ID
)
AND
PRO.P_ID NOT IN
(
SELECT
FPR.P_ID
FROM
FALL_KONTEXT_PROZ_WERT_OPS_OP FPR
WHERE
FPR.P_ID = PRO.P_ID
AND
FPR.FALL_ID = FAL.FALL_ID
AND
FPR.FALLOP_ID = FOP.FALLOP_ID
);
```

## A4.37. View [Fall\_Kontext\_Proz\_Wert\_Komb\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZ_WERT_KOMB_OP AS
2 SELECT
3 FAL.FALL_ID          AS FALL_ID,
4 FAL.CODE            AS FALL_CODE,
5 FAL.NACHNAME        AS NACHNAME,
6 FAL.VORNAME         AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
9 FAL.ABGANG_FA       AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH       AS ABGANG_KH,
12 FOP.FALLOP_ID      AS FALLOP_ID,
13 FOP.CODE            AS OP_CODE,
14 FOP.BEGINN         AS OP_BEGINN,
15 FOP.ENDE            AS OP_ENDE,
16 PRO.P_ID           AS P_ID,
17 PRO.BEZEICHNUNG     AS PROZEDUR_BEZEICHNUNG,
18 PWK.PWK_ID         AS PWK_ID,
19 PWK.BEZEICHNUNG     AS PROZEDUR_WERT_BEZEICHNUNG,
20 PWK.WERT           AS PROZEDUR_WERT,
21 PWK.RANG           AS PROZEDUR_WERT_RANG,
22 PWK.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
23 KPR.KP_ID          AS KP_ID,
24 KON.K_ID           AS K_ID,
25 KON.BEZEICHNUNG    AS KONTEXT_BEZEICHNUNG,
26 KPR.RANG           AS KONTEXT_RANG
27 FROM
28 FALL                FAL,
29 FALL_OP             FOP,
30 PROZEDUR            PRO,
31 PROZEDUR_WERT_KOMBINATION PWK,
32 PROZEDUR_WERT_KOMB_ELEM_COUNT PKE,
33 KONTEXT_PROZEDUR    KPR,
34 KONTEXT             KON
35 WHERE
36 FAL.FALL_ID = FOP.FALL_ID
37 AND
38 PRO.P_ID = PWK.P_ID
39 AND
40 PWK.PWK_ID = PKE.PWK_ID
41 AND
42 PRO.P_ID = KPR.P_ID
43 AND
44 KPR.K_ID = KON.K_ID
45 AND
46 (
47 (
48 -- Kombinationstyp: alle Prozeduren sind obligat!
49 PWK.PWKT_ID = 1
50 AND
51 PKE.ZAHL_ELEMENTE =
52 (
53 SELECT
54 COUNT (PKP.PWKPW_ID)
55 FROM
56 FALL_KONTEXT_PROZ_WERT_OPS_OP FDO,
57 PROZ_WERT_KOMB_PROZ_WERT      PKP
58 WHERE
59 FDO.FALL_ID = FAL.FALL_ID
60 AND
61 FDO.FALLOP_ID = FOP.FALLOP_ID
62 AND
63 FDO.PW_ID = PKP.PW_ID
64 AND
65 PKP.PWK_ID = PWK.PWK_ID
```

)	66
)	67
OR	68
(	69
-- Kombinationstyp: mindestens 2 Prozeduren sind obligat!	70
PWK.PWKT_ID = 2	71
AND	72
1 <	73
(	74
SELECT	75
COUNT (PKP.PWKPW_ID)	76
FROM	77
FALL_KONTEXT_PROZ_WERT_OPS_OP FDO,	78
PROZ_WERT_KOMB_PROZ_WERT PKP	79
WHERE	80
FDO.FALL_ID = FAL.FALL_ID	81
AND	82
FDO.FALLOP_ID = FOP.FALLOP_ID	83
AND	84
FDO.PW_ID = PKP.PW_ID	85
AND	86
PKP.PWK_ID = PWK.PWK_ID	87
)	88
)	89
);	90

### A4.38. View [Fall\_Kontext\_Prozedur\_Rang\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZEDUR_RANG_OP AS
2 SELECT
3 FALL_ID                AS FALL_ID,
4 FALL_CODE              AS FALL_CODE,
5 NACHNAME              AS NACHNAME,
6 VORNAME              AS VORNAME,
7 GEBURTSDATUM        AS GEBURTSDATUM,
8 AUFNAHME_FA         AS AUFNAHME_FA,
9 ABGANG_FA           AS ABGANG_FA,
10 AUFNAHME_KH        AS AUFNAHME_KH,
11 ABGANG_KH          AS ABGANG_KH,
12 FALLOP_ID          AS FALLOP_ID,
13 OP_CODE            AS OP_CODE,
14 OP_BEGINN         AS OP_BEGINN,
15 OP_ENDE           AS OP_ENDE,
16 P_ID              AS P_ID,
17 PW_ID            AS PW_ID,
18 NULL              AS PWK_ID,
19 PROZEDUR_BEZEICHNUNG AS PROZEDUR_BEZEICHNUNG,
20 PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
21 PROZEDUR_WERT      AS PROZEDUR_WERT,
22 PROZEDUR_WERT_RANG AS PROZEDUR_WERT_RANG,
23 1                  AS EJN_ID_AUSWERTUNG,
24 KP_ID             AS KP_ID,
25 K_ID              AS K_ID,
26 KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
27 KONTEXT_RANG      AS KONTEXT_RANG
28 FROM
29 FALL_KONTEXT_PROZ_W_NON_OPS_OP
30 UNION
31 SELECT
32 FALL_ID,
33 FALL_CODE,
34 NACHNAME,
35 VORNAME,
36 GEBURTSDATUM,
37 AUFNAHME_FA,
38 ABGANG_FA,
39 AUFNAHME_KH,
40 ABGANG_KH,
41 FALLOP_ID,
42 OP_CODE,
43 OP_BEGINN,
44 OP_ENDE,
45 P_ID,
46 PW_ID,
47 NULL,
48 PROZEDUR_BEZEICHNUNG,
49 PROZEDUR_WERT_BEZEICHNUNG,
50 PROZEDUR_WERT,
51 PROZEDUR_WERT_RANG,
52 EJN_ID_AUSWERTUNG,
53 KP_ID,
54 K_ID,
55 KONTEXT_BEZEICHNUNG,
56 KONTEXT_RANG
57 FROM
58 FALL_KONTEXT_PROZ_WERT_OPS_OP
59 UNION
60 SELECT
61 FALL_ID,
62 FALL_CODE,
63 NACHNAME,
64 VORNAME,
65 GEBURTSDATUM,
```

AUFNAHME_FA,	66
ABGANG_FA,	67
AUFNAHME_KH,	68
ABGANG_KH,	69
FALLOP_ID,	70
OP_CODE,	71
OP_BEGINN,	72
OP_ENDE,	73
P_ID,	74
NULL,	75
PWK_ID,	76
PROZEDUR_BEZEICHNUNG,	77
PROZEDUR_WERT_BEZEICHNUNG,	78
PROZEDUR_WERT,	79
PROZEDUR_WERT_RANG,	80
EJN_ID_AUSWERTUNG,	81
KP_ID,	82
K_ID,	83
KONTEXT_BEZEICHNUNG,	84
KONTEXT_RANG	85
FROM	86
FALL_KONTEXT_PROZ_WERT_KOMB_OP;	87

## A4.39. View [Fall\_Kontext\_Prozedur\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_PROZEDUR_OP AS
2 SELECT
3 T1.FALL_ID AS FALL_ID,
4 T1.FALL_CODE AS FALL_CODE,
5 T1.NACHNAME AS NACHNAME,
6 T1.VORNAME AS VORNAME,
7 T1.GEBURTSDATUM AS GEBURTSDATUM,
8 T1.AUFNAHME_FA AS AUFNAHME_FA,
9 T1.ABGANG_FA AS ABGANG_FA,
10 T1.AUFNAHME_KH AS AUFNAHME_KH,
11 T1.ABGANG_KH AS ABGANG_KH,
12 T1.FALLOP_ID AS FALLOP_ID,
13 T1.OP_CODE AS OP_CODE,
14 T1.OP_BEGINN AS OP_BEGINN,
15 T1.OP_ENDE AS OP_ENDE,
16 T1.P_ID AS P_ID,
17 T1.PW_ID AS PW_ID,
18 T1.PWK_ID AS PWK_ID,
19 T1.PROZEDUR_BEZEICHNUNG AS PROZEDUR_BEZEICHNUNG,
20 T1.PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
21 T1.PROZEDUR_WERT AS PROZEDUR_WERT,
22 T1.PROZEDUR_WERT_RANG AS PROZEDUR_WERT_RANG,
23 T1.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
24 T1.KP_ID AS KP_ID,
25 T1.K_ID AS K_ID,
26 T1.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
27 T1.KONTEXT_RANG AS KONTEXT_RANG
28 FROM
29 FALL_KONTEXT_PROZEDUR_RANG_OP T1
30 WHERE
31 T1.EJN_ID_AUSWERTUNG = 1
32 AND
33 (
34 T1.FALL_ID,
35 T1.FALLOP_ID,
36 T1.P_ID,
37 T1.PROZEDUR_WERT_RANG
38 )
39 IN
40 (
41 SELECT
42 T2.FALL_ID,
43 T2.FALLOP_ID,
44 T2.P_ID,
45 MIN (T2.PROZEDUR_WERT_RANG)
46 FROM
47 FALL_KONTEXT_PROZEDUR_RANG_OP T2
48 WHERE
49 T2.FALL_ID = T1.FALL_ID
50 AND
51 T2.P_ID = T1.P_ID
52 GROUP BY
53 T2.FALL_ID,
54 T2.FALLOP_ID,
55 T2.P_ID
56 );
```

## A4.40. View [Fall\_Kont\_Komp\_D\_Wert\_ICD\_OP]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_D_WERT_ICD_OP AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FAL.CODE            AS FALL_CODE,
FAL.NACHNAME        AS NACHNAME,
FAL.VORNAME         AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA     AS AUFNAHME_FA,
FAL.ABGANG_FA       AS ABGANG_FA,
FAL.AUFNAHME_KH     AS AUFNAHME_KH,
FAL.ABGANG_KH       AS ABGANG_KH,
FOP.FALLOP_ID      AS FALLOP_ID,
FOP.CODE            AS OP_CODE,
FOP.BEGINN          AS OP_BEGINN,
FOP.ENDE            AS OP_ENDE,
DIA.D_ID            AS D_ID,
DIA.BEZEICHNUNG     AS DIAGNOSE_BEZEICHNUNG,
DIW.DW_ID           AS DW_ID,
DIW.BEZEICHNUNG     AS DIAGNOSE_WERT_BEZEICHNUNG,
DIW.WERT            AS DIAGNOSE_WERT,
KKP.KKP_ID          AS KKP_ID,
KON.K_ID            AS K_ID,
KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
KKP.RANG            AS KONTEXT_RANG,
KPL.KP_ID           AS KP_ID,
KPL.BEZEICHNUNG     AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID          AS KPW_ID,
KPW.BEZEICHNUNG     AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.RANG            AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
KWD.KPWDW_ID        AS KPWDW_ID
FROM
FALL
FALL_OP
DIAGNOSE
DIAGNOSE_WERT
KONTEXT_KOMPLEX
KONTEXT
KOMPLEX
KOMPLEX_WERT
KOMPLEX_WERT_D_WERT
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
KKP.KP_ID = KPL.KP_ID
AND
KKP.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWD.KPW_ID
AND
KWD.DW_ID = DIW.DW_ID
AND
DIW.D_ID = DIA.D_ID
AND
KKP.K_ID = KON.K_ID
-- -----
-- Auflösung eines definierten oder eben nicht definierten Seitenbezugs:
-- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
-- vernachlässigt werden
-- 2. Fall: Seitenlokalisierung ist definiert UND muss dann mit der des Fall-bezogenen
-- ICD-Codes übereinstimmen
-- -----
AND
(
DIW.DW_ID IN
```

```

66  (
67  SELECT
68  DWI.DW_ID
69  FROM
70  DIAGNOSE_WERT_ICD   DWI,
71  FALL_ICD_OP         FIC
72  WHERE
73  DWI.DW_ID = DIW.DW_ID
74  AND
75  FIC.FALL_ID = FAL.FALL_ID
76  AND
77  FIC.FALLOP_ID = FOP.FALLOP_ID
78  AND
79  FIC.ICD_ID = DWI.ICD_ID
80  AND
81  DWI.DWI_ID NOT IN
82  (
83  SELECT
84  DWL.DWI_ID
85  FROM
86  DIAGNOSE_WERT_ICD_LOK   DWL
87  WHERE
88  DWL.DWI_ID = DWI.DWI_ID
89  )
90  )
91  OR
92  DIW.DW_ID IN
93  (
94  SELECT
95  DWI.DW_ID
96  FROM
97  DIAGNOSE_WERT_ICD       DWI,
98  DIAGNOSE_WERT_ICD_LOK   DWL,
99  FALL_ICD_OP             FIC
100 WHERE
101 DWI.DW_ID = DIW.DW_ID
102 AND
103 FIC.FALL_ID = FAL.FALL_ID
104 AND
105 FIC.FALLOP_ID = FOP.FALLOP_ID
106 AND
107 FIC.ICD_ID = DWI.ICD_ID
108 AND
109 DWI.DWI_ID = DWL.DWI_ID
110 AND
111 FIC.ICDLOK_ID = DWL.ICDLOK_ID
112 )
113 )
114 -- -----
115 -- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs
116 -- -----
117 -- -----
118 -- Kategorie-Auswertung:
119 -- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und auszuschließenden
120 -- ICD-Kategorien
121 -- -----
122 AND
123 (
124 DIW.DW_ID IN
125 (
126 SELECT
127 DWI.DW_ID
128 FROM
129 DIAGNOSE_WERT_ICD       DWI,
130 FALL_ICD_OP             FIC,
131 PHASE_ICD_KATEGORIE     PIK,
132 KONTEXT_KOMPLEX_PHASE   KPH
133 WHERE

```



```

DWI.DW_ID = DIW.DW_ID                                134
AND                                                    135
FIC.FALL_ID = FAL.FALL_ID                            136
AND                                                    137
FIC.FALLOP_ID = FOP.FALLOP_ID                       138
AND                                                    139
DWI.ICD_ID = FIC.ICD_ID                             140
AND                                                    141
PIK.ICDKAT_ID = FIC.ICDKAT_ID                       142
AND                                                    143
KPH.KKP_ID = KKP.KKP_ID                             144
AND                                                    145
PIK.PH_ID = KPH.PH_ID                               146
AND                                                    147
PIK.PHZT_ID = 1 -- EINSCHLUSS!                       148
)                                                    149
AND                                                    150
DIW.DW_ID NOT IN                                    151
(                                                    152
SELECT                                               153
DWI.DW_ID                                           154
FROM                                                 155
DIAGNOSE_WERT_ICD          DWI,                    156
FALL_ICD_OP                FIC,                    157
PHASE_ICD_KATEGORIE       PIK,                    158
KONTEXT_KOMPLEX_PHASE     KPH                    159
WHERE                                                                160
DWI.DW_ID = DIW.DW_ID                                             161
AND                                                                162
FIC.FALL_ID = FAL.FALL_ID                                       163
AND                                                                164
FIC.FALLOP_ID = FOP.FALLOP_ID                                   165
AND                                                                166
DWI.ICD_ID = FIC.ICD_ID                                         167
AND                                                                168
PIK.ICDKAT_ID = FIC.ICDKAT_ID                                  169
AND                                                                170
KPH.KKP_ID = KKP.KKP_ID                                         171
AND                                                                172
PIK.PH_ID = KPH.PH_ID                                           173
AND                                                                174
PIK.PHZT_ID = 2 -- AUSSCHLUSS!                                  175
)                                                                176
)                                                                177
-- -----                                                    178
-- Ende: Kategorie-Auswertung                                  179
-- -----                                                    180
-- -----                                                    181
-- Auswertung Zeitpunkt:                                       182
-- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf, 183
-- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten 184
-- Phase                                                         185
-- -----                                                    186
AND                                                            187
(                                                            188
-- -----                                                    189
-- Auswertung Zeitpunkt -                                       190
-- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu). 191
-- Keine zeitliche Einschränkung, da durch die ICD-Kategorie bereits eine 192
-- eindeutige Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!) 193
-- -----                                                    194
DIW.DW_ID IN                                                195
(                                                            196
SELECT                                               197
DWI.DW_ID                                           198
FROM                                                 199
DIAGNOSE_WERT_ICD          DWI,                    200
FALL_ICD_OP                FIC,                    201

```

```

202 KONTEXT_KOMPLEX_PHASE KPH
203 WHERE
204 DWI.DW_ID = DIW.DW_ID
205 AND
206 FIC.FALL_ID = FAL.FALL_ID
207 AND
208 FIC.FALLOP_ID = FOP.FALLOP_ID
209 AND
210 DWI.ICD_ID = FIC.ICD_ID
211 AND
212 KPH.KKP_ID = KKP.KKP_ID
213 AND
214 KPH.PH_ID IN (1, 2, 5, 6)
215 )
216 -- -----
217 -- Auswertung Zeitpunkt -
218 -- Phase: postoperativ
219 -- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
220 -- übereinstimmen oder hinter diesem liegen
221 -- -----
222 OR
223 DIW.DW_ID IN
224 (
225 SELECT
226 DWI.DW_ID
227 FROM
228 DIAGNOSE_WERT_ICD      DWI,
229 FALL_ICD_OP            FIC,
230 KONTEXT_KOMPLEX_PHASE KPH
231 WHERE
232 DWI.DW_ID = DIW.DW_ID
233 AND
234 FIC.FALL_ID = FAL.FALL_ID
235 AND
236 FIC.FALLOP_ID = FOP.FALLOP_ID
237 AND
238 DWI.ICD_ID = FIC.ICD_ID
239 AND
240 KPH.KKP_ID = KKP.KKP_ID
241 AND
242 KPH.PH_ID IN (7)
243 AND
244 FIC.DATUM >=
245 (
246 SELECT
247 MIN (FOP.BEGINN)
248 FROM
249 FALL_OP FOP
250 WHERE
251 FOP.FALL_ID = FAL.FALL_ID
252 )
253 )
254 -- -----
255 -- Auswertung Zeitpunkt -
256 -- Phase: intraop. Haupteingriff
257 -- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
258 -- übereinstimmen
259 -- -----
260 OR
261 DIW.DW_ID IN
262 (
263 SELECT
264 DWI.DW_ID
265 FROM
266 DIAGNOSE_WERT_ICD      DWI,
267 FALL_ICD_OP            FIC,
268 KONTEXT_KOMPLEX_PHASE KPH
269 WHERE

```

```

DWI.DW_ID = DIW.DW_ID                                270
AND                                                    271
FIC.FALL_ID = FAL.FALL_ID                            272
AND                                                    273
FIC.FALLOP_ID = FOP.FALLOP_ID                       274
AND                                                    275
DWI.ICD_ID = FIC.ICD_ID                             276
AND                                                    277
KPH.KKP_ID = KKP.KKP_ID                             278
AND                                                    279
KPH.PH_ID IN (3)                                     280
AND                                                    281
FIC.DATUM =                                          282
(                                                    283
SELECT                                              284
MIN (FOP.BEGINN)                                  285
FROM                                              286
FALL_OP FOP                                       287
WHERE                                             288
FOP.FALL_ID = FAL.FALL_ID                         289
)                                                 290
)                                                 291
-- -----                                          292
-- Auswertung Zeitpunkt -                          293
-- Phase: intraop. Folgeeingriff(e)                294
-- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs liegen 295
-- -----                                          296
OR                                                    297
DIW.DW_ID IN                                         298
(                                                    299
SELECT                                              300
DWI.DW_ID                                         301
FROM                                              302
DIAGNOSE_WERT_ICD          DWI,                  303
FALL_ICD_OP                FIC,                  304
KONTEXT_KOMPLEX_PHASE     KPH                    305
WHERE                                             306
DWI.DW_ID = DIW.DW_ID                             307
AND                                                    308
FIC.FALL_ID = FAL.FALL_ID                         309
AND                                                    310
FIC.FALLOP_ID = FOP.FALLOP_ID                    311
AND                                                    312
DWI.ICD_ID = FIC.ICD_ID                           313
AND                                                    314
KPH.KKP_ID = KKP.KKP_ID                           315
AND                                                    316
KPH.PH_ID IN (4)                                   317
AND                                                    318
FIC.DATUM >                                        319
(                                                    320
SELECT                                              321
MIN (FOP.BEGINN)                                  322
FROM                                              323
FALL_OP FOP                                       324
WHERE                                             325
FOP.FALL_ID = FAL.FALL_ID                         326
)                                                 327
)                                                 328
-- -----                                          329
-- Ende: Auswertung Zeitpunkt                       330
-- -----                                          331
);                                                  332

```

## A4.41. View [Fall\_Kont\_Komp\_P\_Wert\_OPS\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONT_KOMP_P_WERT_OPS_OP AS
2 SELECT
3 FAL.FALL_ID           AS FALL_ID,
4 FAL.CODE             AS FALL_CODE,
5 FAL.NACHNAME         AS NACHNAME,
6 FAL.VORNAME          AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA      AS AUFNAHME_FA,
9 FAL.ABGANG_FA        AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH        AS ABGANG_KH,
12 FOP.FALLOP_ID       AS FALLOP_ID,
13 FOP.CODE             AS OP_CODE,
14 FOP.BEGINN          AS OP_BEGINN,
15 FOP.ENDE            AS OP_ENDE,
16 PRO.P_ID            AS P_ID,
17 PRO.BEZEICHNUNG     AS PROZEDUR_BEZEICHNUNG,
18 PRW.PW_ID           AS PW_ID,
19 PRW.BEZEICHNUNG     AS PROZEDUR_WERT_BEZEICHNUNG,
20 PRW.WERT            AS PROZEDUR_WERT,
21 KKP.KKP_ID          AS KKP_ID,
22 KON.K_ID            AS K_ID,
23 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
24 KKP.RANG            AS KONTEXT_RANG,
25 KPL.KP_ID           AS KP_ID,
26 KPL.BEZEICHNUNG     AS KOMPLEX_BEZEICHNUNG,
27 KPW.KPW_ID          AS KPW_ID,
28 KPW.BEZEICHNUNG     AS KOMPLEX_WERT_BEZEICHNUNG,
29 KPW.RANG            AS KOMPLEX_WERT_RANG,
30 KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
31 KWP.KPWPW_ID        AS KPWPW_ID
32 FROM
33 FALL                 FAL,
34 FALL_OP              FOP,
35 PROZEDUR             PRO,
36 PROZEDUR_WERT        PRW,
37 KONTEXT_KOMPLEX     KKP,
38 KONTEXT              KON,
39 KOMPLEX              KPL,
40 KOMPLEX_WERT         KPW,
41 KOMPLEX_WERT_P_WERT KWP
42 WHERE
43 FAL.FALL_ID = FOP.FALL_ID
44 AND
45 KKP.KP_ID = KPL.KP_ID
46 AND
47 KKP.KP_ID = KPW.KP_ID
48 AND
49 KPW.KPW_ID = KWP.KPW_ID
50 AND
51 KWP.PW_ID = PRW.PW_ID
52 AND
53 PRW.P_ID = PRO.P_ID
54 AND
55 KKP.K_ID = KON.K_ID
56 -----
57 -- Auflösung eines definierten oder eben
58 -- nicht definierten Seitenbezugs:
59 -- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
60 -- vernachlässigt werden
61 -- 2. Fall:
62 -- Seitenlokalisation ist definiert UND muss dann mit der des Fall-bezogenen
63 -- OPS-Codes übereinstimmen
64 -----
65 AND
```

```

(
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS PWO,
FALL_OPS_OP FOO
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
FOO.OPS_ID = PWO.OPS_ID
AND
PWO.PWO_ID NOT IN
(
SELECT
PWL.PWO_ID
FROM
PROZEDUR_WERT_OPS_LOK PWL
WHERE
PWL.PWO_ID = PWO.PWO_ID
)
)
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS PWO,
PROZEDUR_WERT_OPS_LOK PWL,
FALL_OPS_OP FOO
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
FOO.OPS_ID = PWO.OPS_ID
AND
PWO.PWO_ID = PWL.PWO_ID
AND
FOO.OPSLOK_ID = PWL.OPSLOK_ID
)
)
-- -----
-- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs
-- -----
-- Kategorie-Auswertung:
-- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und auszuschließenden
-- OPS-Kategorien
-- -----
AND
(
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS PWO,
FALL_OPS_OP FOO,
PHASE_OPS301_KATEGORIE POK,

```

```

134 KONTEXT_KOMPLEX_PHASE    KPH
135 WHERE
136 PWO.PW_ID = PRW.PW_ID
137 AND
138 FOO.FALL_ID = FAL.FALL_ID
139 AND
140 FOO.FALLOP_ID = FOP.FALLOP_ID
141 AND
142 PWO.OPS_ID = FOO.OPS_ID
143 AND
144 POK.OPSKAT_ID = FOO.OPSKAT_ID
145 AND
146 KPH.KKP_ID = KKP.KKP_ID
147 AND
148 POK.PH_ID = KPH.PH_ID
149 AND
150 POK.PHZEIT_ID = 1  -- EINSCHLUSS!
151 )
152 AND
153 PRW.PW_ID NOT IN
154 (
155 SELECT
156 PWO.PW_ID
157 FROM
158 PROZEDUR_WERT_OPS          PWO,
159 FALL_OPS_OP                FOO,
160 PHASE_OPS301_KATEGORIE    POK,
161 KONTEXT_KOMPLEX_PHASE    KPH
162 WHERE
163 PWO.PW_ID = PRW.PW_ID
164 AND
165 FOO.FALL_ID = FAL.FALL_ID
166 AND
167 FOO.FALLOP_ID = FOP.FALLOP_ID
168 AND
169 PWO.OPS_ID = FOO.OPS_ID
170 AND
171 POK.OPSKAT_ID = FOO.OPSKAT_ID
172 AND
173 KPH.KKP_ID = KKP.KKP_ID
174 AND
175 POK.PH_ID = KPH.PH_ID
176 AND
177 POK.PHZEIT_ID = 2  -- AUSSCHLUSS!
178 )
179 )
180 -----
181 -- Ende: Kategorie-Auswertung
182 -----
183 -----
184 -- Auswertung Zeitpunkt:
185 -- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf,
186 -- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten
187 -- Phase
188 -----
189 AND
190 (
191 -----
192 -- Auswertung Zeitpunkt -
193 -- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu).
194 -- Keine zeitliche Einschränkung, da durch die OPS-Kategorie bereits eine
195 -- eindeutige Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!)
196 -----
197 PRW.PW_ID IN
198 (
199 SELECT
200 PWO.PW_ID
201 FROM

```

```

PROZEDUR_WERT_OPS      PWO,          202
FALL_OPS_OP            FOO,          203
KONTEXT_KOMPLEX_PHASE KPH          204
WHERE                  205
PWO.PW_ID = PRW.PW_ID  206
AND                   207
FOO.FALL_ID = FAL.FALL_ID  208
AND                   209
FOO.FALLOP_ID = FOP.FALLOP_ID  210
AND                   211
PWO.OPS_ID = FOO.OPS_ID  212
AND                   213
KPH.KKP_ID = KKP.KKP_ID  214
AND                   215
KPH.PH_ID IN (1, 2, 5, 6)  216
)                      217
-- ----- 218
-- Auswertung Zeitpunkt - 219
-- Phase: postoperativ 220
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs 221
-- übereinstimmen oder hinter diesem liegen 222
-- ----- 223
OR 224
PRW.PW_ID IN 225
( 226
SELECT 227
PWO.PW_ID 228
FROM 229
PROZEDUR_WERT_OPS      PWO,          230
FALL_OPS_OP            FOO,          231
KONTEXT_KOMPLEX_PHASE KPH          232
WHERE 233
PWO.PW_ID = PRW.PW_ID  234
AND 235
FOO.FALL_ID = FAL.FALL_ID  236
AND 237
FOO.FALLOP_ID = FOP.FALLOP_ID  238
AND 239
PWO.OPS_ID = FOO.OPS_ID  240
AND 241
KPH.KKP_ID = KKP.KKP_ID  242
AND 243
KPH.PH_ID IN (7) 244
AND 245
FOO.DATUM >= 246
( 247
SELECT 248
MIN (FOP.BEGINN) 249
FROM 250
FALL_OP FOP 251
WHERE 252
FOP.FALL_ID = FAL.FALL_ID  253
) 254
) 255
-- ----- 256
-- Auswertung Zeitpunkt - 257
-- Phase: intraop. Haupteingriff 258
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs 259
-- übereinstimmen 260
-- ----- 261
OR 262
PRW.PW_ID IN 263
( 264
SELECT 265
PWO.PW_ID 266
FROM 267
PROZEDUR_WERT_OPS      PWO,          268
FALL_OPS_OP            FOO,          269

```

```

270 KONTEXT_KOMPLEX_PHASE KPH
271 WHERE
272 PWO.PW_ID = PRW.PW_ID
273 AND
274 FOO.FALL_ID = FAL.FALL_ID
275 AND
276 FOO.FALLOP_ID = FOP.FALLOP_ID
277 AND
278 PWO.OPS_ID = FOO.OPS_ID
279 AND
280 KPH.KKP_ID = KKP.KKP_ID
281 AND
282 KPH.PH_ID IN (3)
283 AND
284 FOO.DATUM =
285 (
286 SELECT
287 MIN (FOP.BEGINN)
288 FROM
289 FALL_OP FOP
290 WHERE
291 FOP.FALL_ID = FAL.FALL_ID
292 )
293 )
294 -- -----
295 -- Auswertung Zeitpunkt -
296 -- Phase: intraop. Folgeeingriff(e)
297 -- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs
298 -- liegen
299 -- -----
300 OR
301 PRW.PW_ID IN
302 (
303 SELECT
304 PWO.PW_ID
305 FROM
306 PROZEDUR_WERT_OPS PWO,
307 FALL_OPS_OP FOO,
308 KONTEXT_KOMPLEX_PHASE KPH
309 WHERE
310 PWO.PW_ID = PRW.PW_ID
311 AND
312 FOO.FALL_ID = FAL.FALL_ID
313 AND
314 FOO.FALLOP_ID = FOP.FALLOP_ID
315 AND
316 PWO.OPS_ID = FOO.OPS_ID
317 AND
318 KPH.KKP_ID = KKP.KKP_ID
319 AND
320 KPH.PH_ID IN (4)
321 AND
322 FOO.DATUM >
323 (
324 SELECT
325 MIN (FOP.BEGINN)
326 FROM
327 FALL_OP FOP
328 WHERE
329 FOP.FALL_ID = FAL.FALL_ID
330 )
331 )
332 -- -----
333 -- Ende: Auswertung Zeitpunkt
334 -- -----
335 );

```



## A4.42. View [Fall\_Kont\_Komp\_DWK\_DW\_ICD\_OP]

```

CREATE OR REPLACE VIEW FALL_KONT_KOMP_DWK_DW_ICD_OP AS
SELECT
FAL.FALL_ID           AS FALL_ID,
FAL.CODE             AS FALL_CODE,
FAL.NACHNAME         AS NACHNAME,
FAL.VORNAME          AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA      AS AUFNAHME_FA,
FAL.ABGANG_FA        AS ABGANG_FA,
FAL.AUFNAHME_KH      AS AUFNAHME_KH,
FAL.ABGANG_KH        AS ABGANG_KH,
FOP.FALLOP_ID        AS FALLOP_ID,
FOP.CODE             AS OP_CODE,
FOP.BEGINN           AS OP_BEGINN,
FOP.ENDE             AS OP_ENDE,
DIA.D_ID             AS D_ID,
DIA.BEZEICHNUNG      AS DIAGNOSE_BEZEICHNUNG,
DIW.DW_ID            AS DW_ID,
DIW.BEZEICHNUNG      AS DIAGNOSE_WERT_BEZEICHNUNG,
DIW.WERT             AS DIAGNOSE_WERT,
DWK.DWK_ID           AS DWK_ID,
DWK.BEZEICHNUNG      AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
DWK.WERT             AS DIAGNOSE_WERT_KOMB,
DWK.DWKT_ID          AS DWKT_ID,
DKT.BEZEICHNUNG      AS DIAGNOSE_WERT_KOMBINATIONSTYP,
KKP.KKP_ID           AS KKP_ID,
KON.K_ID             AS K_ID,
KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
KKP.RANG             AS KONTEXT_RANG,
KPL.KP_ID            AS KP_ID,
KPL.BEZEICHNUNG      AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID           AS KPW_ID,
KPW.BEZEICHNUNG      AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.RANG             AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
KPW.KWT_ID           AS KWT_ID,
KWT.BEZEICHNUNG      AS KOMPLEX_WERT_KOMBINATIONSTYP,
KWK.KPWDWK_ID        AS KPWDWK_ID
FROM
FALL
FALL_OP
DIAGNOSE
DIAGNOSE_WERT
DIAGNOSE_WERT_KOMBINATION
DIAGNOSE_WERT_KOMBINATIONSTYP
DIAG_WERT_KOMB_DIAG_WERT
KONTEXT_KOMPLEX
KONTEXT
KOMPLEX
KOMPLEX_WERT
KOMPLEX_WERT_D_WERT_KOMB
KOMPLEX_WERT_KOMBINATIONSTYP
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
KKP.KP_ID = KPL.KP_ID
AND
KKP.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWK.KPW_ID
AND
KPW.KWT_ID = KWT.KWT_ID
AND
KWK.DWK_ID = DWK.DWK_ID
AND

```

```

66 DWK.DWK_ID = DKD.DWK_ID
67 AND
68 DWK.DWKT_ID = DKT.DWKT_ID
69 AND
70 DKD.DW_ID = DIW.DW_ID
71 AND
72 DIW.D_ID = DIA.D_ID
73 AND
74 KKP.K_ID = KON.K_ID
75 -- -----
76 -- Auflösung eines definierten oder eben
77 -- nicht definierten Seitenbezugs:
78 -- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
79 -- vernachlässigt werden
80 -- 2. Fall: Seitenlokalisierung ist definiert UND muss dann mit der des Fall-bezogenen
81 -- ICD-Codes übereinstimmen
82 -- -----
83 AND
84 (
85 DIW.DW_ID IN
86 (
87 SELECT
88 DWI.DW_ID
89 FROM
90 DIAGNOSE_WERT_ICD DWI,
91 FALL_ICD_OP FIC
92 WHERE
93 DWI.DW_ID = DIW.DW_ID
94 AND
95 FIC.FALL_ID = FAL.FALL_ID
96 AND
97 FIC.FALLOP_ID = FOP.FALLOP_ID
98 AND
99 FIC.ICD_ID = DWI.ICD_ID
100 AND
101 DWI.DWI_ID NOT IN
102 (
103 SELECT
104 DWL.DWI_ID
105 FROM
106 DIAGNOSE_WERT_ICD_LOK DWL
107 WHERE
108 DWL.DWI_ID = DWI.DWI_ID
109 )
110 )
111 OR
112 DIW.DW_ID IN
113 (
114 SELECT
115 DWI.DW_ID
116 FROM
117 DIAGNOSE_WERT_ICD DWI,
118 DIAGNOSE_WERT_ICD_LOK DWL,
119 FALL_ICD_OP FIC
120 WHERE
121 DWI.DW_ID = DIW.DW_ID
122 AND
123 FIC.FALL_ID = FAL.FALL_ID
124 AND
125 FIC.FALLOP_ID = FOP.FALLOP_ID
126 AND
127 FIC.ICD_ID = DWI.ICD_ID
128 AND
129 DWI.DWI_ID = DWL.DWI_ID
130 AND
131 FIC.ICDLOK_ID = DWL.ICDLOK_ID
132 )
133 )

```

```

-----
-- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs
-----
-----
-- Kategorie-Auswertung:
-- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und auszuschließenden
-- ICD-Kategorien
-----
AND
(
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_OP            FIC,
PHASE_ICD_KATEGORIE   PIK,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
FIC.FALLOP_ID = FOP.FALLOP_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
PIK.ICDKAT_ID = FIC.ICDKAT_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
PIK.PH_ID = KPH.PH_ID
AND
PIK.PHZT_ID = 1  -- EINSCHLUSS!
)
AND
DIW.DW_ID NOT IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_OP            FIC,
PHASE_ICD_KATEGORIE   PIK,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
FIC.FALLOP_ID = FOP.FALLOP_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
PIK.ICDKAT_ID = FIC.ICDKAT_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
PIK.PH_ID = KPH.PH_ID
AND
PIK.PHZT_ID = 2  -- AUSSCHLUSS!
)
)
-----
-- Ende: Kategorie-Auswertung
-----
-----

```

134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201

```

202 -- Auswertung Zeitpunkt:
203 -- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf,
204 -- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten
205 -- Phase
206 -- -----
207 AND
208 (
209 -- -----
210 -- Auswertung Zeitpunkt -
211 -- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu).
212 -- Keine zeitliche Einschränkung, da durch die ICD-Kategorie bereits eine
213 -- eindeutige Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!)
214 -- -----
215 DIW.DW_ID IN
216 (
217 SELECT
218 DWI.DW_ID
219 FROM
220 DIAGNOSE_WERT_ICD      DWI,
221 FALL_ICD_OP            FIC,
222 KONTEXT_KOMPLEX_PHASE KPH
223 WHERE
224 DWI.DW_ID = DIW.DW_ID
225 AND
226 FIC.FALL_ID = FAL.FALL_ID
227 AND
228 FIC.FALLOP_ID = FOP.FALLOP_ID
229 AND
230 DWI.ICD_ID = FIC.ICD_ID
231 AND
232 KPH.KKP_ID = KKP.KKP_ID
233 AND
234 KPH.PH_ID IN (1, 2, 5, 6)
235 )
236 -- -----
237 -- Auswertung Zeitpunkt -
238 -- Phase: postoperativ
239 -- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
240 -- übereinstimmen oder hinter diesem liegen
241 -- -----
242 OR
243 DIW.DW_ID IN
244 (
245 SELECT
246 DWI.DW_ID
247 FROM
248 DIAGNOSE_WERT_ICD      DWI,
249 FALL_ICD_OP            FIC,
250 KONTEXT_KOMPLEX_PHASE KPH
251 WHERE
252 DWI.DW_ID = DIW.DW_ID
253 AND
254 FIC.FALL_ID = FAL.FALL_ID
255 AND
256 FIC.FALLOP_ID = FOP.FALLOP_ID
257 AND
258 DWI.ICD_ID = FIC.ICD_ID
259 AND
260 KPH.KKP_ID = KKP.KKP_ID
261 AND
262 KPH.PH_ID IN (7)
263 AND
264 FIC.DATUM >=
265 (
266 SELECT
267 MIN (FOP.BEGINN)
268 FROM
269 FALL_OP FOP

```

```

WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Haupteingriff
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen
-----
OR
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_OP            FIC,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
FIC.FALLOP_ID = FOP.FALLOP_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (3)
AND
FIC.DATUM =
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP  FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Folgeeingriff(e)
-- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs
-- liegen
-----
OR
DIW.DW_ID IN
(
SELECT
DWI.DW_ID
FROM
DIAGNOSE_WERT_ICD      DWI,
FALL_ICD_OP            FIC,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
DWI.DW_ID = DIW.DW_ID
AND
FIC.FALL_ID = FAL.FALL_ID
AND
FIC.FALLOP_ID = FOP.FALLOP_ID
AND
DWI.ICD_ID = FIC.ICD_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND

```

```
338 KPH.PH_ID IN (4)
339 AND
340 FIC.DATUM >
341 (
342 SELECT
343 MIN (FOP.BEGINN)
344 FROM
345 FALL_OP FOP
346 WHERE
347 FOP.FALL_ID = FAL.FALL_ID
348 )
349 )
350 -----
351 -- Ende: Auswertung Zeitpunkt
352 -----
353 );
```

### A4.43. View [Fall\_Kont\_Komp\_PWK\_PW\_OPS\_OP]

```

CREATE OR REPLACE VIEW FALL_KONT_KOMP_PWK_PW_OPS_OP AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FAL.CODE             AS FALL_CODE,
FAL.NACHNAME         AS NACHNAME,
FAL.VORNAME          AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA      AS AUFNAHME_FA,
FAL.ABGANG_FA        AS ABGANG_FA,
FAL.AUFNAHME_KH      AS AUFNAHME_KH,
FAL.ABGANG_KH        AS ABGANG_KH,
FOP.FALLOP_ID       AS FALLOP_ID,
FOP.CODE             AS OP_CODE,
FOP.BEGINN           AS OP_BEGINN,
FOP.ENDE             AS OP_ENDE,
PRO.P_ID             AS P_ID,
PRO.BEZEICHNUNG      AS PROZEDUR_BEZEICHNUNG,
PRW.PW_ID            AS PW_ID,
PRW.BEZEICHNUNG      AS PROZEDUR_WERT_BEZEICHNUNG,
PRW.WERT             AS PROZEDUR_WERT,
PWK.PWK_ID           AS PWK_ID,
PWK.BEZEICHNUNG      AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
PWK.WERT             AS PROZEDUR_WERT_KOMB,
PWK.PWKT_ID          AS PWKT_ID,
PKT.BEZEICHNUNG      AS PROZEDUR_WERT_KOMBINATIONSTYP,
KKP.KKP_ID           AS KKP_ID,
KON.K_ID             AS K_ID,
KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
KKP.RANG             AS KONTEXT_RANG,
KPL.KP_ID            AS KP_ID,
KPL.BEZEICHNUNG      AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID           AS KPW_ID,
KPW.BEZEICHNUNG      AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.RANG             AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS EJN_ID_AUSWERTUNG,
KPW.KWT_ID           AS KWT_ID,
KKT.BEZEICHNUNG      AS KOMPLEX_WERT_KOMBINATIONSTYP,
KWK.KPWPWK_ID        AS KPWPWK_ID
FROM
FALL
FALL_OP
PROZEDUR
PROZEDUR_WERT
PROZEDUR_WERT_KOMBINATION
PROZEDUR_WERT_KOMBINATIONSTYP
PROZ_WERT_KOMB_PROZ_WERT
KONTEXT_KOMPLEX
KONTEXT
KOMPLEX
KOMPLEX_WERT
KOMPLEX_WERT_P_WERT_KOMB
KOMPLEX_WERT_KOMBINATIONSTYP
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
KKP.KP_ID = KPL.KP_ID
AND
KKP.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWK.KPW_ID
AND
KPW.KWT_ID = KKT.KWT_ID
AND
KWK.PWK_ID = PWK.PWK_ID
AND

```

```

66 PWK.PWK_ID = PKP.PWK_ID
67 AND
68 PWK.PWKT_ID = PKT.PWKT_ID
69 AND
70 PKP.PW_ID = PRW.PW_ID
71 AND
72 PRW.P_ID = PRO.P_ID
73 AND
74 KKP.K_ID = KON.K_ID
75 -- -----
76 -- Auflösung eines definierten oder eben
77 -- nicht definierten Seitenbezugs:
78 -- 1. Fall: es wurde kein Seitenbezug definiert, daher kann diese Eigenschaft
79 -- vernachlässigt werden
80 -- 2. Fall: Seitenlokalisierung ist definiert UND muss dann mit der des Fall-bezogenen
81 -- OPS-Codes übereinstimmen
82 -- -----
83 AND
84 (
85 PRW.PW_ID IN
86 (
87 SELECT
88 PWO.PW_ID
89 FROM
90 PROZEDUR_WERT_OPS PWO,
91 FALL_OPS_OP FOO
92 WHERE
93 PWO.PW_ID = PRW.PW_ID
94 AND
95 FOO.FALL_ID = FAL.FALL_ID
96 AND
97 FOO.FALLOP_ID = FOP.FALLOP_ID
98 AND
99 FOO.OPS_ID = PWO.OPS_ID
100 AND
101 PWO.PWO_ID NOT IN
102 (
103 SELECT
104 PWL.PWO_ID
105 FROM
106 PROZEDUR_WERT_OPS_LOK PWL
107 WHERE
108 PWL.PWO_ID = PWO.PWO_ID
109 )
110 )
111 OR
112 PRW.PW_ID IN
113 (
114 SELECT
115 PWO.PW_ID
116 FROM
117 PROZEDUR_WERT_OPS PWO,
118 PROZEDUR_WERT_OPS_LOK PWL,
119 FALL_OPS_OP FOO
120 WHERE
121 PWO.PW_ID = PRW.PW_ID
122 AND
123 FOO.FALL_ID = FAL.FALL_ID
124 AND
125 FOO.FALLOP_ID = FOP.FALLOP_ID
126 AND
127 FOO.OPS_ID = PWO.OPS_ID
128 AND
129 PWO.PWO_ID = PWL.PWO_ID
130 AND
131 FOO.OPSLOK_ID = PWL.OPSLOK_ID
132 )
133 )

```



```

-----
-- Ende: Auflösung des ggf. zu berücksichtigenden Seitenbezugs
-----
-----
-- Kategorie-Auswertung:
-- Berücksichtigung von gemäß der Kontext-Phasen-Zuordnung ein- und
-- auszuschließenden OPS-Kategorien
-----
AND
(
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_OP           FOO,
PHASE_OPS301_KATEGORIE POK,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.OPS_ID = FOO.OPS_ID
AND
POK.OPSKAT_ID = FOO.OPSKAT_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
POK.PH_ID = KPH.PH_ID
AND
POK.PHZT_ID = 1  -- EINSCHLUSS!
)
)
AND
PRW.PW_ID NOT IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_OP           FOO,
PHASE_OPS301_KATEGORIE POK,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.OPS_ID = FOO.OPS_ID
AND
POK.OPSKAT_ID = FOO.OPSKAT_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
POK.PH_ID = KPH.PH_ID
AND
POK.PHZT_ID = 2  -- AUSSCHLUSS!
)
)
-----
-- Ende: Kategorie-Auswertung
-----
-----

```

134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201

```

202 -- Auswertung Zeitpunkt:
203 -- Berücksichtigung von chronologischen Gesichtspunkten (Zeitpunkt im Beh.-Verlauf,
204 -- Zuordnung zu Erst- und ggf. Folgeeingriffen etc.) in Abhängigkeit von der definierten
205 -- Phase
206 -- -----
207 AND
208 (
209 -- -----
210 -- Auswertung Zeitpunkt -
211 -- Phasen: Aufnahme, Behandlung, intraop. allgemein, Behandlung (neu).
212 -- Keine zeitliche Einschränkung, da durch die OPS-Kategorie bereits eine
213 -- eindeutige Zuordnung möglich ist (bei Beachtung der Ein- und Ausschlüsse!)
214 -- -----
215 PRW.PW_ID IN
216 (
217 SELECT
218 PWO.PW_ID
219 FROM
220 PROZEDUR_WERT_OPS      PWO,
221 FALL_OPS_OP            FOO,
222 KONTEXT_KOMPLEX_PHASE KPH
223 WHERE
224 PWO.PW_ID = PRW.PW_ID
225 AND
226 FOO.FALL_ID = FAL.FALL_ID
227 AND
228 FOO.FALLOP_ID = FOP.FALLOP_ID
229 AND
230 PWO.OPS_ID = FOO.OPS_ID
231 AND
232 KPH.KKP_ID = KKP.KKP_ID
233 AND
234 KPH.PH_ID IN (1, 2, 5, 6)
235 )
236 -- -----
237 -- Auswertung Zeitpunkt -
238 -- Phase: postoperativ
239 -- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
240 -- übereinstimmen oder hinter diesem liegen
241 -- -----
242 OR
243 PRW.PW_ID IN
244 (
245 SELECT
246 PWO.PW_ID
247 FROM
248 PROZEDUR_WERT_OPS      PWO,
249 FALL_OPS_OP            FOO,
250 KONTEXT_KOMPLEX_PHASE KPH
251 WHERE
252 PWO.PW_ID = PRW.PW_ID
253 AND
254 FOO.FALL_ID = FAL.FALL_ID
255 AND
256 FOO.FALLOP_ID = FOP.FALLOP_ID
257 AND
258 PWO.OPS_ID = FOO.OPS_ID
259 AND
260 KPH.KKP_ID = KKP.KKP_ID
261 AND
262 KPH.PH_ID IN (7)
263 AND
264 FOO.DATUM >=
265 (
266 SELECT
267 MIN (FOP.BEGINN)
268 FROM
269 FALL_OP FOP

```

```

WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Haupteingriff
-- Zeitliche Einschränkung: Zeitpunkt muss mit dem Beginn des ersten Eingriffs
-- übereinstimmen
-----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_OP           FOO,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.OPS_ID = FOO.OPS_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (3)
AND
FOO.DATUM =
(
SELECT
MIN (FOP.BEGINN)
FROM
FALL_OP  FOP
WHERE
FOP.FALL_ID = FAL.FALL_ID
)
)
-----
-- Auswertung Zeitpunkt -
-- Phase: intraop. Folgeeingriff(e)
-- Zeitliche Einschränkung: Zeitpunkt muss hinter dem Beginn des ersten Eingriffs liegen
-----
OR
PRW.PW_ID IN
(
SELECT
PWO.PW_ID
FROM
PROZEDUR_WERT_OPS      PWO,
FALL_OPS_OP           FOO,
KONTEXT_KOMPLEX_PHASE KPH
WHERE
PWO.PW_ID = PRW.PW_ID
AND
FOO.FALL_ID = FAL.FALL_ID
AND
FOO.FALLOP_ID = FOP.FALLOP_ID
AND
PWO.OPS_ID = FOO.OPS_ID
AND
KPH.KKP_ID = KKP.KKP_ID
AND
KPH.PH_ID IN (4)

```

```
338 AND
339 FOO.DATUM >
340 (
341 SELECT
342 MIN (FOP.BEGINN)
343 FROM
344 FALL_OP FOP
345 WHERE
346 FOP.FALL_ID = FAL.FALL_ID
347 )
348 )
349 -- -----
350 -- Ende: Auswertung Zeitpunkt
351 -- -----
352 );
```

#### A4.44. View [Fall\_Kont\_Komp\_D\_Wert\_Komb\_OP]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_D_WERT_KOMB_OP AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FAL.CODE            AS FALL_CODE,
FAL.NACHNAME        AS NACHNAME,
FAL.VORNAME         AS VORNAME,
FAL.GEBURTSDATUM   AS GEBURTSDATUM,
FAL.AUFNAHME_FA     AS AUFNAHME_FA,
FAL.ABGANG_FA       AS ABGANG_FA,
FAL.AUFNAHME_KH     AS AUFNAHME_KH,
FAL.ABGANG_KH       AS ABGANG_KH,
FOP.FALLOP_ID       AS FALLOP_ID,
FOP.CODE            AS OP_CODE,
FOP.BEGINN          AS OP_BEGINN,
FOP.ENDE            AS OP_ENDE,
DIA.D_ID            AS D_ID,
DIA.BEZEICHNUNG     AS DIAGNOSE_BEZEICHNUNG,
DWK.DWK_ID          AS DWK_ID,
DWK.BEZEICHNUNG     AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
DWK.WERT            AS DIAGNOSE_WERT_KOMB,
KKP.KKP_ID          AS KKP_ID,
KON.K_ID            AS K_ID,
KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
KKP.RANG            AS KONTEXT_RANG,
KPL.KP_ID           AS KP_ID,
KPL.BEZEICHNUNG     AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID          AS KPW_ID,
KPW.BEZEICHNUNG     AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.WERT            AS KOMPLEX_WERT,
KPW.RANG            AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG
FROM
FALL                FAL,
FALL_OP             FOP,
DIAGNOSE            DIA,
DIAGNOSE_WERT_KOMBINATION DWK,
DIAGNOSE_WERT_KOMB_ELEM_COUNT DKE,
KONTEXT_KOMPLEX     KKP,
KONTEXT             KON,
KOMPLEX             KPL,
KOMPLEX_WERT        KPW,
KOMPLEX_WERT_D_WERT_KOMB KWK
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
KKP.K_ID = KON.K_ID
AND
KKP.KP_ID = KPL.KP_ID
AND
KPL.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWK.KPW_ID
AND
KWK.DWK_ID = DWK.DWK_ID
AND
DWK.DWK_ID = DKE.DWK_ID
AND
DWK.D_ID = DIA.D_ID
AND
(
(
-- Kombinationstyp: alle Diagnosen sind obligat!
DWK.DWKT_ID = 1
AND
DKE.ZAHL_ELEMENTE =
```

```

66  (
67  SELECT
68  COUNT (DISTINCT (KDK.DW_ID))
69  FROM
70  FALL_KONT_KOMP_DWK_DW_ICD_OP  KDK
71  WHERE
72  KDK.FALL_ID = FAL.FALL_ID
73  AND
74  KDK.FALLOP_ID = FOP.FALLOP_ID
75  AND
76  KDK.DWK_ID = DWK.DWK_ID
77  )
78  )
79  OR
80  (
81  -- Kombinationstyp: mindestens 2 Diagnosen sind obligat!
82  DWK.DWKT_ID = 2
83  AND
84  1 <
85  (
86  SELECT
87  COUNT (DISTINCT (KDK.DW_ID))
88  FROM
89  FALL_KONT_KOMP_DWK_DW_ICD_OP  KDK
90  WHERE
91  KDK.FALL_ID = FAL.FALL_ID
92  AND
93  KDK.FALLOP_ID = FOP.FALLOP_ID
94  AND
95  KDK.DWK_ID = DWK.DWK_ID
96  )
97  )
98  );

```

## A4.45. View [Fall\_Kont\_Komp\_P\_Wert\_Komb\_OP]

```
CREATE OR REPLACE VIEW FALL_KONT_KOMP_P_WERT_KOMB_OP AS
SELECT
FAL.FALL_ID           AS FALL_ID,
FAL.CODE             AS FALL_CODE,
FAL.NACHNAME         AS NACHNAME,
FAL.VORNAME          AS VORNAME,
FAL.GEBURTSDATUM    AS GEBURTSDATUM,
FAL.AUFNAHME_FA      AS AUFNAHME_FA,
FAL.ABGANG_FA        AS ABGANG_FA,
FAL.AUFNAHME_KH      AS AUFNAHME_KH,
FAL.ABGANG_KH        AS ABGANG_KH,
FOP.FALLOP_ID        AS FALLOP_ID,
FOP.CODE             AS OP_CODE,
FOP.BEGINN           AS OP_BEGINN,
FOP.ENDE             AS OP_ENDE,
PRO.P_ID             AS P_ID,
PRO.BEZEICHNUNG      AS PROZEDUR_BEZEICHNUNG,
PWK.PWK_ID           AS PWK_ID,
PWK.BEZEICHNUNG      AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
PWK.WERT             AS PROZEDUR_WERT_KOMB,
KKP.KKP_ID           AS KKP_ID,
KON.K_ID             AS K_ID,
KON.BEZEICHNUNG      AS KONTEXT_BEZEICHNUNG,
KKP.RANG             AS KONTEXT_RANG,
KPL.KP_ID            AS KP_ID,
KPL.BEZEICHNUNG      AS KOMPLEX_BEZEICHNUNG,
KPW.KPW_ID           AS KPW_ID,
KPW.BEZEICHNUNG      AS KOMPLEX_WERT_BEZEICHNUNG,
KPW.WERT             AS KOMPLEX_WERT,
KPW.RANG             AS KOMPLEX_WERT_RANG,
KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG
FROM
FALL                FAL,
FALL_OP             FOP,
PROZEDUR            PRO,
PROZEDUR_WERT_KOMBINATION PWK,
PROZEDUR_WERT_KOMB_ELEM_COUNT PKE,
KONTEXT_KOMPLEX    KKP,
KONTEXT             KON,
KOMPLEX             KPL,
KOMPLEX_WERT        KPW,
KOMPLEX_WERT_P_WERT_KOMB KWK
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
KKP.K_ID = KON.K_ID
AND
KKP.KP_ID = KPL.KP_ID
AND
KPL.KP_ID = KPW.KP_ID
AND
KPW.KPW_ID = KWK.KPW_ID
AND
KWK.PWK_ID = PWK.PWK_ID
AND
PWK.PWK_ID = PKE.PWK_ID
AND
PWK.P_ID = PRO.P_ID
AND
(
(
-- Kombinationstyp: alle Prozeduren sind obligat!
PWK.PWKT_ID = 1
AND
PKE.ZAHL_ELEMENTE =
```

```

66  (
67  SELECT
68  COUNT (DISTINCT (KPK.PW_ID))
69  FROM
70  FALL_KONT_KOMP_PWK_PW_OPS_OP  KPK
71  WHERE
72  KPK.FALL_ID = FAL.FALL_ID
73  AND
74  KPK.FALLOP_ID = FOP.FALLOP_ID
75  AND
76  KPK.PWK_ID = PWK.PWK_ID
77  )
78  )
79  OR
80  (
81  -- Kombinationstyp: mindestens 2 Prozeduren sind obligat!
82  PWK.PWKT_ID = 2
83  AND
84  1 <
85  (
86  SELECT
87  COUNT (DISTINCT (KPK.PW_ID))
88  FROM
89  FALL_KONT_KOMP_PWK_PW_OPS_OP  KPK
90  WHERE
91  KPK.FALL_ID = FAL.FALL_ID
92  AND
93  KPK.FALLOP_ID = FOP.FALLOP_ID
94  AND
95  KPK.PWK_ID = PWK.PWK_ID
96  )
97  )
98  );

```



## A4.46. View [Fall\_KT\_Komp\_DW\_PW\_DWK\_PWK\_OP]

```
CREATE OR REPLACE VIEW FALL_KT_KOMP_DW_PW_DWK_PWK_OP AS
-----
-- Diagnose-Werte
-----
SELECT
DWE.FALL_ID AS FALL_ID,
DWE.FALL_CODE AS FALL_CODE,
DWE.NACHNAME AS NACHNAME,
DWE.VORNAME AS VORNAME,
DWE.GEBURTSDATUM AS GEBURTSDATUM,
DWE.AUFNAHME_FA AS AUFNAHME_FA,
DWE.ABGANG_FA AS ABGANG_FA,
DWE.AUFNAHME_KH AS AUFNAHME_KH,
DWE.ABGANG_KH AS ABGANG_KH,
DWE.FALLOP_ID AS FALLOP_ID,
DWE.OP_CODE AS OP_CODE,
DWE.OP_BEGINN AS OP_BEGINN,
DWE.OP_ENDE AS OP_ENDE,
DWE.KKP_ID AS KKP_ID,
DWE.K_ID AS K_ID,
DWE.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
DWE.KONTEXT_RANG AS KONTEXT_RANG,
DWE.KP_ID AS KP_ID,
DWE.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
DWE.KPW_ID AS KPW_ID,
DWE.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
DWE.KOMPLEX_WERT_RANG AS KOMPLEX_WERT_RANG,
DWE.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
DWE.D_ID AS D_ID,
DWE.DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
DWE.DW_ID AS DW_ID,
DWE.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
DWE.DIAGNOSE_WERT AS DIAGNOSE_WERT,
NULL AS DWK_ID,
NULL AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
NULL AS DIAGNOSE_WERT_KOMB,
NULL AS P_ID,
NULL AS PROZEDUR_BEZEICHNUNG,
NULL AS PW_ID,
NULL AS PROZEDUR_WERT_BEZEICHNUNG,
NULL AS PROZEDUR_WERT,
NULL AS PWK_ID,
NULL AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
NULL AS PROZEDUR_WERT_KOMB
FROM
FALL_KONT_KOMP_D_WERT_ICD_OP DWE
UNION
-----
-- Prozedur-Werte
-----
SELECT
PWE.FALL_ID,
PWE.FALL_CODE,
PWE.NACHNAME,
PWE.VORNAME,
PWE.GEBURTSDATUM,
PWE.AUFNAHME_FA,
PWE.ABGANG_FA,
PWE.AUFNAHME_KH,
PWE.ABGANG_KH,
PWE.FALLOP_ID,
PWE.OP_CODE,
PWE.OP_BEGINN,
PWE.OP_ENDE,
PWE.KKP_ID,
```

```

66  PWE.K_ID,
67  PWE.KONTEXT_BEZEICHNUNG,
68  PWE.KONTEXT_RANG,
69  PWE.KP_ID,
70  PWE.KOMPLEX_BEZEICHNUNG,
71  PWE.KPW_ID,
72  PWE.KOMPLEX_WERT_BEZEICHNUNG,
73  PWE.KOMPLEX_WERT_RANG,
74  PWE.KPW_EJN_ID_AUSWERTUNG,
75  NULL,
76  NULL,
77  NULL,
78  NULL,
79  NULL,
80  NULL,
81  NULL,
82  NULL,
83  PWE.P_ID,
84  PWE.PROZEDUR_BEZEICHNUNG,
85  PWE.PW_ID,
86  PWE.PROZEDUR_WERT_BEZEICHNUNG,
87  PWE.PROZEDUR_WERT,
88  NULL,
89  NULL,
90  NULL
91  FROM
92  FALL_KONT_KOMP_P_WERT_OPS_OP  PWE
93  UNION
94  -----
95  -- Diagnose-Kombinationswerte
96  -----
97  SELECT
98  DWK.FALL_ID,
99  DWK.FALL_CODE,
100 DWK.NACHNAME,
101 DWK.VORNAME,
102 DWK.GEBURTSDATUM,
103 DWK.AUFNAHME_FA,
104 DWK.ABGANG_FA,
105 DWK.AUFNAHME_KH,
106 DWK.ABGANG_KH,
107 DWK.FALLOP_ID,
108 DWK.OP_CODE,
109 DWK.OP_BEGINN,
110 DWK.OP_ENDE,
111 DWK.KKP_ID,
112 DWK.K_ID,
113 DWK.KONTEXT_BEZEICHNUNG,
114 DWK.KONTEXT_RANG,
115 DWK.KP_ID,
116 DWK.KOMPLEX_BEZEICHNUNG,
117 DWK.KPW_ID,
118 DWK.KOMPLEX_WERT_BEZEICHNUNG,
119 DWK.KOMPLEX_WERT_RANG,
120 DWK.KPW_EJN_ID_AUSWERTUNG,
121 DWK.D_ID,
122 DWK.DIAGNOSE_BEZEICHNUNG,
123 NULL,
124 NULL,
125 NULL,
126 DWK.DWK_ID,
127 DWK.DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
128 DWK.DIAGNOSE_WERT_KOMB,
129 NULL,
130 NULL,
131 NULL,
132 NULL,
133 NULL,

```

NULL,	134
NULL,	135
NULL	136
FROM	137
FALL_KONT_KOMP_D_WERT_KOMB_OP DWK	138
UNION	139
-----	140
-- Prozedur-Kombinationswerte	141
-----	142
SELECT	143
PWK.FALL_ID,	144
PWK.FALL_CODE,	145
PWK.NACHNAME,	146
PWK.VORNAME,	147
PWK.GEBURTSDATUM,	148
PWK.AUFNAHME_FA,	149
PWK.ABGANG_FA,	150
PWK.AUFNAHME_KH,	151
PWK.ABGANG_KH,	152
PWK.FALLOP_ID,	153
PWK.OP_CODE,	154
PWK.OP_BEGINN,	155
PWK.OP_ENDE,	156
PWK.KKP_ID,	157
PWK.K_ID,	158
PWK.KONTEXT_BEZEICHNUNG,	159
PWK.KONTEXT_RANG,	160
PWK.KP_ID,	161
PWK.KOMPLEX_BEZEICHNUNG,	162
PWK.KPW_ID,	163
PWK.KOMPLEX_WERT_BEZEICHNUNG,	164
PWK.KOMPLEX_WERT_RANG,	165
PWK.KPW_EJN_ID_AUSWERTUNG,	166
NULL,	167
NULL,	168
NULL,	169
NULL,	170
NULL,	171
NULL,	172
NULL,	173
NULL,	174
PWK.P_ID,	175
PWK.PROZEDUR_BEZEICHNUNG,	176
NULL,	177
NULL,	178
NULL,	179
PWK.PWK_ID,	180
PWK.PROZEDUR_WERT_KOMB_BEZEICHNUNG,	181
PWK.PROZEDUR_WERT_KOMB	182
FROM	183
FALL_KONT_KOMP_P_WERT_KOMB_OP PWK;	184

## A4.47. View [Fall\_KPW\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KPW_OP AS
2 SELECT
3 FAL.FALL_ID          AS FALL_ID,
4 FAL.CODE            AS FALL_CODE,
5 FAL.NACHNAME        AS NACHNAME,
6 FAL.VORNAME         AS VORNAME,
7 FAL.GEBURTSDATUM    AS GEBURTSDATUM,
8 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
9 FAL.ABGANG_FA       AS ABGANG_FA,
10 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
11 FAL.ABGANG_KH       AS ABGANG_KH,
12 FOP.FALLOP_ID      AS FALLOP_ID,
13 FOP.CODE            AS OP_CODE,
14 FOP.BEGINN         AS OP_BEGINN,
15 FOP.ENDE            AS OP_ENDE,
16 KKP.KKP_ID         AS KKP_ID,
17 KON.K_ID           AS K_ID,
18 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
19 KKP.RANG            AS KONTEXT_KOMPLEX_RANG,
20 KPL.KP_ID          AS KP_ID,
21 KPL.BEZEICHNUNG     AS KOMPLEX_BEZEICHNUNG,
22 KPW.KPW_ID         AS KPW_ID,
23 KPW.BEZEICHNUNG     AS KOMPLEX_WERT_BEZEICHNUNG,
24 KPW.RANG            AS KOMPLEX_WERT_RANG,
25 KPW.WERT           AS KOMPLEX_WERT,
26 KPW.EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
27 KPW.KWT_ID         AS KWT_ID
28 FROM
29 FALL                FAL,
30 FALL_OP            FOP,
31 KONTEXT_KOMPLEX    KKP,
32 KONTEXT            KON,
33 KOMPLEX            KPL,
34 KOMPLEX_WERT       KPW
35 WHERE
36 FAL.FALL_ID = FOP.FALL_ID
37 AND
38 KKP.K_ID = KON.K_ID
39 AND
40 KKP.KP_ID = KPL.KP_ID
41 AND
42 KPL.KP_ID = KPW.KP_ID;
```

#### A4.48. View [Fall\_KT\_KP\_W\_DW\_PW\_DWK\_PWK\_OP]

```
CREATE OR REPLACE VIEW FALL_KT_KP_W_DW_PW_DWK_PWK_OP AS
SELECT
FKP.FALL_ID AS FALL_ID,
FKP.FALL_CODE AS FALL_CODE,
FKP.NACHNAME AS NACHNAME,
FKP.VORNAME AS VORNAME,
FKP.GEBURTSDATUM AS GEBURTSDATUM,
FKP.AUFNAHME_FA AS AUFNAHME_FA,
FKP.ABGANG_FA AS ABGANG_FA,
FKP.AUFNAHME_KH AS AUFNAHME_KH,
FKP.ABGANG_KH AS ABGANG_KH,
FKP.FALLOP_ID AS FALLOP_ID,
FKP.OP_CODE AS OP_CODE,
FKP.OP_BEGINN AS OP_BEGINN,
FKP.OP_ENDE AS OP_ENDE,
FKP.KKP_ID AS KKP_ID,
FKP.K_ID AS K_ID,
FKP.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
FKP.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG,
FKP.KP_ID AS KP_ID,
FKP.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
FKP.KPW_ID AS KPW_ID,
FKP.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
FKP.KOMPLEX_WERT_RANG AS KOMPLEX_WERT_RANG,
FKP.KOMPLEX_WERT AS KOMPLEX_WERT,
FKP.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
FKP.KWT_ID AS KWT_ID,
DWE.D_ID AS DWE_D_ID,
DWE.DIAGNOSE_BEZEICHNUNG AS DWE_DIAGNOSE_BEZEICHNUNG,
DWE.DW_ID AS DW_ID,
DWE.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
DWE.DIAGNOSE_WERT AS DIAGNOSE_WERT,
DWK.D_ID AS DWK_D_ID,
DWK.DIAGNOSE_BEZEICHNUNG AS DWK_DIAGNOSE_BEZEICHNUNG,
DWK.DWK_ID AS DWK_ID,
DWK.DIAGNOSE_WERT_KOMB_BEZEICHNUNG AS DIAGNOSE_WERT_KOMB_BEZEICHNUNG,
DWK.DIAGNOSE_WERT_KOMB AS DIAGNOSE_WERT_KOMB,
PWE.P_ID AS PWE_P_ID,
PWE.PROZEDUR_BEZEICHNUNG AS PWE_PROZEDUR_BEZEICHNUNG,
PWE.PW_ID AS PW_ID,
PWE.PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
PWE.PROZEDUR_WERT AS PROZEDUR_WERT,
PWK.P_ID AS PWK_P_ID,
PWK.PROZEDUR_BEZEICHNUNG AS PWK_PROZEDUR_BEZEICHNUNG,
PWK.PWK_ID AS PWK_ID,
PWK.PROZEDUR_WERT_KOMB_BEZEICHNUNG AS PROZEDUR_WERT_KOMB_BEZEICHNUNG,
PWK.PROZEDUR_WERT_KOMB AS PROZEDUR_WERT_KOMB
FROM
FALL_KPW_OP FKP,
FALL_KONT_KOMP_D_WERT_ICD_OP DWE,
FALL_KONT_KOMP_D_WERT_KOMB_OP DWK,
FALL_KONT_KOMP_P_WERT_OPS_OP PWE,
FALL_KONT_KOMP_P_WERT_KOMB_OP PWK
WHERE
FKP.FALL_ID = DWE.FALL_ID (+)
AND
FKP.FALLOP_ID = DWE.FALLOP_ID (+)
AND
FKP.KPW_ID = DWE.KPW_ID (+)
AND
FKP.FALL_ID = DWK.FALL_ID (+)
AND
FKP.FALLOP_ID = DWK.FALLOP_ID (+)
AND
FKP.KPW_ID = DWK.KPW_ID (+)
```

```
66 AND
67 FKP.FALL_ID = PWE.FALL_ID (+)
68 AND
69 FKP.FALLOP_ID = PWE.FALLOP_ID (+)
70 AND
71 FKP.KPW_ID = PWE.KPW_ID (+)
72 AND
73 FKP.FALL_ID = PWK.FALL_ID (+)
74 AND
75 FKP.FALLOP_ID = PWK.FALLOP_ID (+)
76 AND
77 FKP.KPW_ID = PWK.KPW_ID (+);
```

#### A4.49. View [Fall\_KT\_KP\_Wert\_D\_P\_Count\_OP]

```
CREATE OR REPLACE VIEW FALL_KT_KP_WERT_D_P_COUNT_OP AS 1
SELECT 2
KKW.FALL_ID AS FALL_ID, 3
KKW.FALL_CODE AS FALL_CODE, 4
KKW.NACHNAME AS NACHNAME, 5
KKW.VORNAME AS VORNAME, 6
KKW.GEBURTSDATUM AS GEBURTSDATUM, 7
KKW.AUFNAHME_FA AS AUFNAHME_FA, 8
KKW.ABGANG_FA AS ABGANG_FA, 9
KKW.AUFNAHME_KH AS AUFNAHME_KH, 10
KKW.ABGANG_KH AS ABGANG_KH, 11
KKW.FALLOP_ID AS FALLOP_ID, 12
KKW.OP_CODE AS OP_CODE, 13
KKW.OP_BEGINN AS OP_BEGINN, 14
KKW.OP_ENDE AS OP_ENDE, 15
KKW.KKP_ID AS KKP_ID, 16
KKW.K_ID AS K_ID, 17
KKW.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG, 18
KKW.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG, 19
KKW.KP_ID AS KP_ID, 20
KKW.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG, 21
KKW.KPW_ID AS KPW_ID, 22
KKW.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG, 23
KKW.KOMPLEX_WERT_RANG AS KOMPLEX_WERT_RANG, 24
KKW.KOMPLEX_WERT AS KOMPLEX_WERT, 25
KKW.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG, 26
KKW.KWT_ID AS KWT_ID, 27
COUNT (KKW.DWE_D_ID) + COUNT (KKW.DWK_D_ID) AS D_COUNT, 28
COUNT (KKW.PWE_P_ID) + COUNT (KKW.PWK_P_ID) AS P_COUNT 29
FROM 30
FALL_KT_KP_W_DW_PW_DWK_PWK_OP KKW 31
GROUP BY 32
KKW.FALL_ID, 33
KKW.FALL_CODE, 34
KKW.NACHNAME, 35
KKW.VORNAME, 36
KKW.GEBURTSDATUM, 37
KKW.AUFNAHME_FA, 38
KKW.ABGANG_FA, 39
KKW.AUFNAHME_KH, 40
KKW.ABGANG_KH, 41
KKW.FALLOP_ID, 42
KKW.OP_CODE, 43
KKW.OP_BEGINN, 44
KKW.OP_ENDE, 45
KKW.KKP_ID, 46
KKW.K_ID, 47
KKW.KONTEXT_BEZEICHNUNG, 48
KKW.KONTEXT_KOMPLEX_RANG, 49
KKW.KP_ID, 50
KKW.KOMPLEX_BEZEICHNUNG, 51
KKW.KPW_ID, 52
KKW.KOMPLEX_WERT_BEZEICHNUNG, 53
KKW.KOMPLEX_WERT_RANG, 54
KKW.KOMPLEX_WERT, 55
KKW.KPW_EJN_ID_AUSWERTUNG, 56
KKW.KWT_ID; 57
```

## A4.50. View [Fall\_Kontext\_Komp\_Wert\_Rang\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_KOMP_WERT_RANG_OP AS
2 SELECT
3 DPC.FALL_ID AS FALL_ID,
4 DPC.FALL_CODE AS FALL_CODE,
5 DPC.NACHNAME AS NACHNAME,
6 DPC.VORNAME AS VORNAME,
7 DPC.GEBURTSDATUM AS GEBURTSDATUM,
8 DPC.AUFNAHME_FA AS AUFNAHME_FA,
9 DPC.ABGANG_FA AS ABGANG_FA,
10 DPC.AUFNAHME_KH AS AUFNAHME_KH,
11 DPC.ABGANG_KH AS ABGANG_KH,
12 DPC.FALLOP_ID AS FALLOP_ID,
13 DPC.OP_CODE AS OP_CODE,
14 DPC.OP_BEGINN AS OP_BEGINN,
15 DPC.OP_ENDE AS OP_ENDE,
16 DPC.KKP_ID AS KKP_ID,
17 DPC.K_ID AS K_ID,
18 DPC.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
19 DPC.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG,
20 DPC.KP_ID AS KP_ID,
21 DPC.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
22 DPC.KPW_ID AS KPW_ID,
23 DPC.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
24 DPC.KOMPLEX_WERT_RANG AS KOMPLEX_WERT_RANG,
25 DPC.KOMPLEX_WERT AS KOMPLEX_WERT,
26 DPC.KPW_EJN_ID_AUSWERTUNG AS KPW_EJN_ID_AUSWERTUNG,
27 DPC.KWT_ID AS KWT_ID
28 FROM
29 FALL_KT_KP_WERT_D_P_COUNT_OP DPC
30 WHERE
31 -----
32 -- es muss mindestens ein Element vorhanden sein
33 -----
34 (
35 DPC.KWT_ID = 1
36 AND
37 (DPC.D_COUNT + DPC.P_COUNT) > 0
38 )
39 -----
40 -- es muss mindestens ein diagnostisches UND ein prozedurales Element vorhanden sein
41 -----
42 OR
43 (
44 DPC.KWT_ID = 2
45 AND
46 DPC.D_COUNT > 0
47 AND
48 DPC.P_COUNT > 0
49 );
```



## A4.51. View [Fall\_Kontext\_Komp\_Min\_Rang\_OP]

```
CREATE OR REPLACE VIEW FALL_KONTEXT_KOMP_MIN_RANG_OP AS 1
SELECT 2
KWR.FALL_ID AS FALL_ID, 3
KWR.FALL_CODE AS FALL_CODE, 4
KWR.NACHNAME AS NACHNAME, 5
KWR.VORNAME AS VORNAME, 6
KWR.GEBURTSDATUM AS GEBURTSDATUM, 7
KWR.AUFNAHME_FA AS AUFNAHME_FA, 8
KWR.ABGANG_FA AS ABGANG_FA, 9
KWR.AUFNAHME_KH AS AUFNAHME_KH, 10
KWR.ABGANG_KH AS ABGANG_KH, 11
KWR.FALLOP_ID AS FALLOP_ID, 12
KWR.OP_CODE AS OP_CODE, 13
KWR.OP_BEGINN AS OP_BEGINN, 14
KWR.OP_ENDE AS OP_ENDE, 15
KWR.KKP_ID AS KKP_ID, 16
KWR.K_ID AS K_ID, 17
KWR.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG, 18
KWR.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG, 19
KWR.KP_ID AS KP_ID, 20
KWR.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG, 21
MIN (KWR.KOMPLEX_WERT_RANG) AS MINIMUM_RANG 22
FROM 23
FALL_KONTEXT_KOMP_WERT_RANG_OP KWR 24
GROUP BY 25
KWR.FALL_ID, 26
KWR.FALL_CODE, 27
KWR.NACHNAME, 28
KWR.VORNAME, 29
KWR.GEBURTSDATUM, 30
KWR.AUFNAHME_FA, 31
KWR.ABGANG_FA, 32
KWR.AUFNAHME_KH, 33
KWR.ABGANG_KH, 34
KWR.FALLOP_ID, 35
KWR.OP_CODE, 36
KWR.OP_BEGINN, 37
KWR.OP_ENDE, 38
KWR.KKP_ID, 39
KWR.K_ID, 40
KWR.KONTEXT_BEZEICHNUNG, 41
KWR.KONTEXT_KOMPLEX_RANG, 42
KWR.KP_ID, 43
KWR.KOMPLEX_BEZEICHNUNG; 44
```

## A4.52. View [Fall\_Kontext\_Komplex\_OP]

```
1 CREATE OR REPLACE VIEW FALL_KONTEXT_KOMPLEX_OP AS
2 -- -----
3 -- Non-Normal-Werte
4 -- -----
5 SELECT
6 KMR.FALL_ID AS FALL_ID,
7 KMR.FALL_CODE AS FALL_CODE,
8 KMR.NACHNAME AS NACHNAME,
9 KMR.VORNAME AS VORNAME,
10 KMR.GEBURTSDATUM AS GEBURTSDATUM,
11 KMR.AUFNAHME_FA AS AUFNAHME_FA,
12 KMR.ABGANG_FA AS ABGANG_FA,
13 KMR.AUFNAHME_KH AS AUFNAHME_KH,
14 KMR.ABGANG_KH AS ABGANG_KH,
15 KMR.FALLOP_ID AS FALLOP_ID,
16 KMR.OP_CODE AS OP_CODE,
17 KMR.OP_BEGINN AS OP_BEGINN,
18 KMR.OP_ENDE AS OP_ENDE,
19 KMR.KKP_ID AS KKP_ID,
20 KMR.K_ID AS K_ID,
21 KMR.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
22 KMR.KONTEXT_KOMPLEX_RANG AS KONTEXT_KOMPLEX_RANG,
23 KMR.KP_ID AS KP_ID,
24 KMR.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
25 KPW.WERT AS KOMPLEX_WERT,
26 KPW.BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
27 KPW.KPW_ID AS KPW_ID
28 FROM
29 FALL_KONTEXT_KOMP_MIN_RANG_OP KMR,
30 KOMPLEX KPL,
31 KOMPLEX_WERT KPW
32 WHERE
33 KMR.KP_ID = KPW.KP_ID
34 AND
35 KMR.KP_ID = KPL.KP_ID
36 AND
37 KMR.MINIMUM_RANG = KPW.RANG
38 UNION
39 -- -----
40 -- Normal-Werte
41 -- -----
42 SELECT
43 FAL.FALL_ID,
44 FAL.CODE,
45 FAL.NACHNAME,
46 FAL.VORNAME,
47 FAL.GEBURTSDATUM,
48 FAL.AUFNAHME_FA,
49 FAL.ABGANG_FA,
50 FAL.AUFNAHME_KH,
51 FAL.ABGANG_KH,
52 FOP.FALLOP_ID,
53 FOP.CODE,
54 FOP.BEGINN,
55 FOP.ENDE,
56 KKP.KKP_ID,
57 KKP.K_ID,
58 KON.BEZEICHNUNG,
59 KKP.RANG,
60 KKP.KP_ID,
61 KPL.BEZEICHNUNG,
62 KPL.NORMAL_WERT,
63 KPL.NORMAL_TEXT,
64 NULL
65 FROM
```

FALL	FAL,	66
FALL_OP	FOP,	67
KONTEXT_KOMPLEX	KKP,	68
KONTEXT	KON,	69
KOMPLEX	KPL	70
WHERE		71
FAL.FALL_ID = FOP.FALL_ID		72
AND		73
KKP.K_ID = KON.K_ID		74
AND		75
KKP.KP_ID = KPL.KP_ID		76
AND		77
(		78
FAL.FALL_ID,		79
FOP.FALLÖP_ID,		80
KKP.KP_ID		81
)		82
NOT IN		83
(		84
SELECT		85
KMR.FALL_ID,		86
KMR.FALLOP_ID,		87
KMR.KP_ID		88
FROM		89
FALL_KONTEXT_KOMP_MIN_RANG_OP KMR		90
WHERE		91
KMR.FALL_ID = FAL.FALL_ID		92
AND		93
KMR.FALLOP_ID = FOP.FALLOP_ID		94
AND		95
KMR.KP_ID = KKP.KP_ID		96
);		97

# Anhang 5 – Datenbank-Abfragen zur Plausibilitätsprüfung

## A5.1. View [Puffer\_Fall\_Element]

```
1 CREATE OR REPLACE VIEW PUFFER_FALL_ELEMENT AS
2 SELECT
3   PFK.FALL_ID           AS FALL_ID,
4   ELE.E_ID             AS E_ID,
5   ELE.BEZEICHNUNG      AS ELEMENT_BEZEICHNUNG,
6   PFK.K_ID             AS K_ID,
7   PFK.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
8   PFK.FALL_CODE        AS FALL_CODE,
9   PFK.NACHNAME         AS NACHNAME,
10  PFK.VORNAME          AS VORNAME,
11  PFK.GEBURTSDATUM     AS GEBURTSDATUM,
12  PFK.AUFNAHME_FA      AS AUFNAHME_FA,
13  PFK.ABGANG_FA        AS ABGANG_FA,
14  PFK.AUFNAHME_KH      AS AUFNAHME_KH,
15  PFK.ABGANG_KH        AS ABGANG_KH
16 FROM
17   PUFFER_FALL_KONTEXT PFK,
18   ELEMENT             ELE
19 WHERE
20   (
21     -- -----
22     -- Ausprägung eines Wertes stimmt mit der einem Element zugeordneten
23     -- Ausprägung eines Wertes überein
24     -- -----
25     (
26       PFK.W_ID IS NOT NULL
27       AND
28       ELE.E_ID IN
29       (
30         SELECT
31           EWW.E_ID
32         FROM
33           ELEMENT_WERT_WERT EWW,
34           WERT_WERT        WEW
35         WHERE
36           ELE.E_ID = EWW.E_ID
37           AND
38           EWW.WW_ID = WEW.WW_ID
39           AND
40           WEW.W_ID = PFK.W_ID
41           AND
42           WEW.WERT = PFK.WERT_WERT
43       )
44     )
45     -- -----
46     -- (Numerischer) Wert (CAVE: mögliche Konversionsprobleme bei Umwandlung
47     -- stringbasierter Angaben!) liegt innerhalb eines Intervalls, das einem
48     -- Element zugeordnet ist
49     -- -----
50     OR
51     (
52       PFK.W_ID IS NOT NULL
53       AND
54       ELE.E_ID IN
55       (
56         SELECT
57           EWB.E_ID
58         FROM
59           ELEMENT_WERT_BEREICH EWB,
60           WERT_BEREICH        WEB
61         WHERE
```

```

EWB.E_ID = ELE.E_ID                                62
AND                                                  63
EWB.WB_ID = WEB.WB_ID                              64
AND                                                  65
WEB.W_ID = PFK.W_ID                                66
AND                                                  67
PFK.WERT_WERT BETWEEN                              68
WEB.WERT_MINIMUM                                  69
AND                                                  70
WEB.WERT_MAXIMUM                                  71
)                                                    72
)                                                    73
-- -----                                         74
-- Element bezieht sich auf den Normal-Wert einer Diagnose 75
-- -----                                         76
OR                                                  77
(                                                    78
PFK.D_ID IS NOT NULL                              79
AND                                                  80
PFK.DW_ID IS NULL                                 81
AND                                                  82
PFK.DWK_ID IS NULL                                83
AND                                                  84
ELE.E_ID IN                                        85
(                                                    86
SELECT                                             87
EDI.E_ID                                          88
FROM                                              89
ELEMENT_DIAGNOSE EDI                             90
WHERE                                             91
EDI.E_ID = ELE.E_ID                              92
AND                                              93
EDI.D_ID = PFK.D_ID                              94
)                                                  95
)                                                  96
-- -----                                         97
-- Element bezieht sich auf den path. Wert einer Diagnose (einfach) 98
-- -----                                         99
OR                                                  100
(                                                    101
PFK.DW_ID IS NOT NULL                             102
AND                                                  103
ELE.E_ID IN                                        104
(                                                    105
SELECT                                             106
EDW.E_ID                                          107
FROM                                              108
ELEMENT_DIAGNOSE_WERT EDW                       109
WHERE                                             110
EDW.E_ID = ELE.E_ID                              111
AND                                              112
EDW.DW_ID = PFK.DW_ID                            113
)                                                  114
)                                                  115
-- -----                                         116
-- Element bezieht sich auf den path. Wert einer Diagnose (Kombination) 117
-- -----                                         118
OR                                                  119
(                                                    120
PFK.DWK_ID IS NOT NULL                             121
AND                                                  122
ELE.E_ID IN                                        123
(                                                    124
SELECT                                             125
EDK.E_ID                                          126
FROM                                              127
ELEMENT_DIAGNOSE_WERT_KOMB EDK                 128
WHERE                                             129

```

```

130 EDK.E_ID = ELE.E_ID
131 AND
132 EDK.DWK_ID = PFK.DWK_ID
133 )
134 )
135 -- -----
136 -- Element bezieht sich auf den Normal-Wert einer Prozedur
137 -- -----
138 OR
139 (
140 PFK.P_ID IS NOT NULL
141 AND
142 PFK.PW_ID IS NULL
143 AND
144 PFK.PWK_ID IS NULL
145 AND
146 ELE.E_ID IN
147 (
148 SELECT
149 EPR.E_ID
150 FROM
151 ELEMENT_PROZEDUR EPR
152 WHERE
153 EPR.E_ID = ELE.E_ID
154 AND
155 EPR.P_ID = PFK.P_ID
156 )
157 )
158 -- -----
159 -- Element bezieht sich auf den non-normal-Wert einer Prozedur (einfach)
160 -- -----
161 OR
162 (
163 PFK.PW_ID IS NOT NULL
164 AND
165 ELE.E_ID IN
166 (
167 SELECT
168 EPW.E_ID
169 FROM
170 ELEMENT_PROZEDUR_WERT EPW
171 WHERE
172 EPW.E_ID = ELE.E_ID
173 AND
174 EPW.PW_ID = PFK.PW_ID
175 )
176 )
177 -- -----
178 -- Element bezieht sich auf den non-normal-Wert einer Prozedur (Kombination)
179 -- -----
180 OR
181 (
182 PFK.PWK_ID IS NOT NULL
183 AND
184 ELE.E_ID IN
185 (
186 SELECT
187 EPK.E_ID
188 FROM
189 ELEMENT_PROZEDUR_WERT_KOMB EPK
190 WHERE
191 EPK.E_ID = ELE.E_ID
192 AND
193 EPK.PWK_ID = PFK.PWK_ID
194 )
195 )
196 -- -----
197 -- Element bezieht sich auf den Normal-Wert eines Komplexes

```

```

-- -----
OR
(
PFK.KOMPLEX_KP_ID IS NOT NULL
AND
PFK.KPW_ID IS NULL
AND
ELE.E_ID IN
(
SELECT
EZX.E_ID
FROM
ELEMENT_KOMPLEX EKX
WHERE
EKX.E_ID = ELE.E_ID
AND
EKX.KP_ID = PFK.KOMPLEX_KP_ID
)
)
-- -----
-- Element bezieht sich auf den non-normal-Wert eines Komplexes
-- -----
OR
(
PFK.KPW_ID IS NOT NULL
AND
ELE.E_ID IN
(
SELECT
EKW.E_ID
FROM
ELEMENT_KOMPLEX_WERT EKW
WHERE
EKW.E_ID = ELE.E_ID
AND
EKW.KPW_ID = PFK.KPW_ID
)
)
-- -----
-- Element bezieht sich auf einen ICD-Wert
-- -----
OR
(
ELE.E_ID IN
(
SELECT
EDI.E_ID
FROM
FALL_KONTEXT_DIAGNOSE_ICD   FKD,
ELEMENT_DIAGNOSE             EDI
WHERE
EDI.E_ID = ELE.E_ID
AND
FKD.FALL_ID = PFK.FALL_ID
AND
EDI.D_ID = FKD.D_ID
)
)
-- -----
-- Element bezieht sich auf einen OPS-Wert
-- -----
OR
(
ELE.E_ID IN
(
SELECT
EPR.E_ID
FROM

```

198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265

```
266 FALL_KONTEXT_PROZEDUR_OPS FKP,  
267 ELEMENT_PROZEDUR EPR  
268 WHERE  
269 EPR.E_ID = ELE.E_ID  
270 AND  
271 FKP.FALL_ID = PFK.FALL_ID  
272 AND  
273 EPR.P_ID = FKP.P_ID  
274 )  
275 )  
276 );
```



## A5.2. View [Puffer\_Fall\_OP\_Element]

```
CREATE OR REPLACE VIEW PUFFER_FALL_OP_ELEMENT AS
SELECT
PFK.FALL_ID           AS FALL_ID,
PFK.FALLÖP_ID        AS FALLÖP_ID,
ELE.E_ID             AS E_ID,
ELE.BEZEICHNUNG      AS ELEMENT_BEZEICHNUNG,
PFK.K_ID             AS K_ID,
PFK.KONTEXT_BEZEICHNUNG AS KONTEXT_BEZEICHNUNG,
PFK.FALL_CODE        AS FALL_CODE,
PFK.NACHNAME         AS NACHNAME,
PFK.VORNAME          AS VORNAME,
PFK.GEBURTSDATUM     AS GEBURTSDATUM,
PFK.AUFNAHME_FA      AS AUFNAHME_FA,
PFK.ABGANG_FA        AS ABGANG_FA,
PFK.AUFNAHME_KH      AS AUFNAHME_KH,
PFK.ABGANG_KH        AS ABGANG_KH,
PFK.OP_CODE          AS OP_CODE,
PFK.OP_BEGINN        AS OP_BEGINN,
PFK.OP_ENDE          AS OP_ENDE
FROM
PUFFER_FALL_OP_KONTEXT PFK,
ELEMENT                ELE
WHERE
(
-- -----
-- Ausprägung eines Wertes stimmt mit der einem Element zugeordneten
-- Ausprägung eines Wertes überein
-- -----
(
PFK.W_ID IS NOT NULL
AND
ELE.E_ID IN
(
SELECT
EWW.E_ID
FROM
ELEMENT_WERT_WERT EWW,
WERT_WERT         WEW
WHERE
ELE.E_ID = EWW.E_ID
AND
EWW.WW_ID = WEW.WW_ID
AND
WEW.W_ID = PFK.W_ID
AND
WEW.WERT = PFK.WERT_WERT
)
)
-- -----
-- (Numerischer) Wert (CAVE: mögliche Konversionsprobleme bei Umwandlung
-- stringbasierter Angaben!) liegt innerhalb eines Intervalls, das einem
-- Element zugeordnet ist
-- -----
OR
(
PFK.W_ID IS NOT NULL
AND
ELE.E_ID IN
(
SELECT
EWB.E_ID
FROM
ELEMENT_WERT_BEREICH EWB,
WERT_BEREICH        WEB
WHERE
```

```

66  EWB.E_ID = ELE.E_ID
67  AND
68  EWB.WB_ID = WEB.WB_ID
69  AND
70  WEB.W_ID = PFK.W_ID
71  AND
72  PFK.WERT_WERT BETWEEN
73  WEB.WERT_MINIMUM
74  AND
75  WEB.WERT_MAXIMUM
76  )
77  )
78  -- -----
79  -- Element bezieht sich auf den Normal-Wert einer Diagnose
80  -- -----
81  OR
82  (
83  PFK.D_ID IS NOT NULL
84  AND
85  PFK.DW_ID IS NULL
86  AND
87  PFK.DWK_ID IS NULL
88  AND
89  ELE.E_ID IN
90  (
91  SELECT
92  EDI.E_ID
93  FROM
94  ELEMENT_DIAGNOSE EDI
95  WHERE
96  EDI.E_ID = ELE.E_ID
97  AND
98  EDI.D_ID = PFK.D_ID
99  )
100 )
101 -- -----
102 -- Element bezieht sich auf den path. Wert einer Diagnose (einfach)
103 -- -----
104 OR
105 (
106 PFK.DW_ID IS NOT NULL
107 AND
108 ELE.E_ID IN
109 (
110 SELECT
111 EDW.E_ID
112 FROM
113 ELEMENT_DIAGNOSE_WERT EDW
114 WHERE
115 EDW.E_ID = ELE.E_ID
116 AND
117 EDW.DW_ID = PFK.DW_ID
118 )
119 )
120 -- -----
121 -- Element bezieht sich auf den path. Wert einer Diagnose (Kombination)
122 -- -----
123 OR
124 (
125 PFK.DWK_ID IS NOT NULL
126 AND
127 ELE.E_ID IN
128 (
129 SELECT
130 EDK.E_ID
131 FROM
132 ELEMENT_DIAGNOSE_WERT_KOMB EDK
133 WHERE

```

```

EDK.E_ID = ELE.E_ID                                134
AND                                                  135
EDK.DWK_ID = PFK.DWK_ID                            136
)                                                    137
)                                                    138
-- -----                                         139
-- Element bezieht sich auf den Normal-Wert einer Prozedur 140
-- -----                                         141
OR                                                  142
(                                                  143
PFK.P_ID IS NOT NULL                               144
AND                                                  145
PFK.PW_ID IS NULL                                  146
AND                                                  147
PFK.PWK_ID IS NULL                                 148
AND                                                  149
ELE.E_ID IN                                        150
(                                                  151
SELECT                                             152
EPR.E_ID                                          153
FROM                                              154
ELEMENT_PROZEDUR EPR                             155
WHERE                                             156
EPR.E_ID = ELE.E_ID                              157
AND                                              158
EPR.P_ID = PFK.P_ID                              159
)                                                  160
)                                                  161
-- -----                                         162
-- Element bezieht sich auf den non-normal-Wert einer Prozedur (einfach) 163
-- -----                                         164
OR                                                  165
(                                                  166
PFK.PW_ID IS NOT NULL                              167
AND                                                  168
ELE.E_ID IN                                        169
(                                                  170
SELECT                                             171
EPW.E_ID                                          172
FROM                                              173
ELEMENT_PROZEDUR_WERT EPW                       174
WHERE                                             175
EPW.E_ID = ELE.E_ID                              176
AND                                              177
EPW.PW_ID = PFK.PW_ID                            178
)                                                  179
)                                                  180
-- -----                                         181
-- Element bezieht sich auf den non-normal-Wert einer Prozedur (Kombination) 182
-- -----                                         183
OR                                                  184
(                                                  185
PFK.PWK_ID IS NOT NULL                            186
AND                                                  187
ELE.E_ID IN                                        188
(                                                  189
SELECT                                             190
EPK.E_ID                                          191
FROM                                              192
ELEMENT_PROZEDUR_WERT_KOMB EPK                 193
WHERE                                             194
EPK.E_ID = ELE.E_ID                              195
AND                                              196
EPK.PWK_ID = PFK.PWK_ID                          197
)                                                  198
)                                                  199
-- -----                                         200
-- Element bezieht sich auf den Normal-Wert eines Komplexes 201

```

```

202  -- -----
203  OR
204  (
205  PFK.KOMPLEX_KP_ID IS NOT NULL
206  AND
207  PFK.KPW_ID IS NULL
208  AND
209  ELE.E_ID IN
210  (
211  SELECT
212  EKX.E_ID
213  FROM
214  ELEMENT_KOMPLEX EKX
215  WHERE
216  EKX.E_ID = ELE.E_ID
217  AND
218  EKX.KP_ID = PFK.KOMPLEX_KP_ID
219  )
220  )
221  -- -----
222  -- Element bezieht sich auf den non-normal-Wert eines Komplexes
223  -- -----
224  OR
225  (
226  PFK.KPW_ID IS NOT NULL
227  AND
228  ELE.E_ID IN
229  (
230  SELECT
231  EKW.E_ID
232  FROM
233  ELEMENT_KOMPLEX_WERT EKW
234  WHERE
235  EKW.E_ID = ELE.E_ID
236  AND
237  EKW.KPW_ID = PFK.KPW_ID
238  )
239  )
240  -- -----
241  -- Element bezieht sich auf einen ICD-Wert
242  -- -----
243  OR
244  (
245  ELE.E_ID IN
246  (
247  SELECT
248  EDI.E_ID
249  FROM
250  FALL_OP_KONTEXT_DIAGNOSE_ICD   FKD,
251  ELEMENT_DIAGNOSE               EDI
252  WHERE
253  EDI.E_ID = ELE.E_ID
254  AND
255  FKD.FALL_ID = PFK.FALL_ID
256  AND
257  EDI.D_ID = FKD.D_ID
258  )
259  )
260  -- -----
261  -- Element bezieht sich auf einen OPS-Wert
262  -- -----
263  OR
264  (
265  ELE.E_ID IN
266  (
267  SELECT
268  EPR.E_ID
269  FROM

```

```
FALL_OP_KONTEXT_PROZEDUR_OPS   FKP,      270
ELEMENT_PROZEDUR                 EPR      271
WHERE                             272
EPR.E_ID = ELE.E_ID              273
AND                               274
FKP.FALL_ID = PFK.FALL_ID        275
AND                               276
EPR.P_ID = FKP.P_ID              277
)                                  278
)                                  279
);                                 280
```

### A5.3. View [Fall\_Pruefung\_Element]

```
1 CREATE OR REPLACE VIEW FALL_PRUEFUNG_ELEMENT AS
2 SELECT
3 PFK.FALL_ID AS FALL_ID,
4 PRF.PRF_ID AS PRF_ID,
5 PRF.HINWEISTEXT AS PRUEFUNG_HINWEISTEXT,
6 ELE.BEZEICHNUNG AS ELEMENT_BEZEICHNUNG,
7 PRE.HINWEISTEXT AS PRUEFUNG_ELEMENT_HINWEISTEXT,
8 PFK.D_ID AS D_ID,
9 PFK.DW_ID AS DW_ID,
10 PFK.DWK_ID AS DWK_ID,
11 PFK.DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
12 PFK.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
13 PFK.DIAGNOSE_WERT AS DIAGNOSE_WERT,
14 PFK.P_ID AS P_ID,
15 PFK.PW_ID AS PW_ID,
16 PFK.PWK_ID AS PWK_ID,
17 PFK.PROZEDUR_BEZEICHNUNG AS PROZEDUR_BEZEICHNUNG,
18 PFK.PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
19 PFK.PROZEDUR_WERT AS PROZEDUR_WERT,
20 PFK.KOMPLEX_KP_ID AS KP_ID,
21 PFK.KPW_ID AS KPW_ID,
22 PFK.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
23 PFK.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
24 PFK.KOMPLEX_WERT AS KOMPLEX_WERT,
25 PFK.W_ID AS W_ID,
26 PFK.WERT_BEZEICHNUNG AS WERT_BEZEICHNUNG,
27 PFK.WERT_WERT AS WERT_WERT,
28 SUK.SK_ID AS PRF_SK_ID,
29 SUK.BEZEICHNUNG AS PRF_SUPERKONTEXT_BEZEICHNUNG,
30 SKK.RANG AS PRF_SUPERKONTEXT_KONTEXT_RANG,
31 KPR.K_ID AS PRF_K_ID,
32 KON.BEZEICHNUNG AS PRF_KONTEXT_BEZEICHNUNG,
33 KPR.RANG AS PRF_KONTEXT_RANG,
34 PFK.K_ID AS ELEMENT_K_ID,
35 PFK.KONTEXT_BEZEICHNUNG AS ELEMENT_KONTEXT_BEZEICHNUNG,
36 PFK.KONTEXT_RANG AS ELEMENT_KONTEXT_RANG,
37 PFK.FALL_CODE AS FALL_CODE,
38 PFK.NACHNAME AS NACHNAME,
39 PFK.VORNAME AS VORNAME,
40 PFK.GEBURTSDATUM AS GEBURTSDATUM,
41 PFK.AUFNAHME_FA AS AUFNAHME_FA,
42 PFK.ABGANG_FA AS ABGANG_FA,
43 PFK.AUFNAHME_KH AS AUFNAHME_KH,
44 PFK.ABGANG_KH AS ABGANG_KH
45 FROM
46 PUFFER_FALL_KONTEXT PFK,
47 KONTEXT_PRUEFUNG KPR,
48 KONTEXT KON,
49 SUPERKONTEXT_KONTEXT SKK,
50 SUPERKONTEXT SUK,
51 DATENEBENE_KONTEXT DEK,
52 PRUEFUNG PRF,
53 PRUEFUNG_ELEMENT PRE,
54 ELEMENT ELE
55 WHERE
56 PFK.K_ID = KPR.K_ID
57 AND
58 KPR.PRF_ID = PRF.PRF_ID
59 AND
60 KPR.K_ID = KON.K_ID
61 AND
62 KON.K_ID = SKK.K_ID
63 AND
64 SKK.SK_ID = SUK.SK_ID
65 AND
```

DEK.K_ID = KON.K_ID	66
AND	67
DEK.DATEB_ID = 1	68
AND	69
PRF.PRF_ID = PRE.PRF_ID	70
AND	71
PRE.E_ID = ELE.E_ID	72
AND	73
(	74
-- Normalwert einer Diagnose	75
(	76
PFK.D_ID IS NOT NULL	77
AND	78
PFK.DW_ID IS NULL	79
AND	80
PFK.DWK_ID IS NULL	81
AND	82
ELE.E_ID IN	83
(	84
SELECT	85
EDI.E_ID	86
FROM	87
ELEMENT_DIAGNOSE EDI	88
WHERE	89
EDI.E_ID = ELE.E_ID	90
AND	91
EDI.D_ID = PFK.D_ID	92
)	93
)	94
-- pathologischer (Einfach-)Wert einer Diagnose	95
OR	96
(	97
PFK.D_ID IS NOT NULL	98
AND	99
PFK.DW_ID IS NOT NULL	100
AND	101
PFK.DWK_ID IS NULL	102
AND	103
ELE.E_ID IN	104
(	105
SELECT	106
EDW.E_ID	107
FROM	108
ELEMENT_DIAGNOSE_WERT EDW	109
WHERE	110
EDW.E_ID = ELE.E_ID	111
AND	112
EDW.DW_ID = PFK.DW_ID	113
)	114
)	115
-- pathologischer (Kombinations-)Wert einer Diagnose	116
OR	117
(	118
PFK.D_ID IS NOT NULL	119
AND	120
PFK.DW_ID IS NULL	121
AND	122
PFK.DWK_ID IS NOT NULL	123
AND	124
ELE.E_ID IN	125
(	126
SELECT	127
EDK.E_ID	128
FROM	129
ELEMENT_DIAGNOSE_WERT_KOMB EDK	130
WHERE	131
EDK.E_ID = ELE.E_ID	132
AND	133

```

134 EDK.DWK_ID = PFK.DWK_ID
135 )
136 )
137 -- Normalwert einer Prozedur
138 OR
139 (
140 PFK.P_ID IS NOT NULL
141 AND
142 PFK.PW_ID IS NULL
143 AND
144 PFK.PWK_ID IS NULL
145 AND
146 ELE.E_ID IN
147 (
148 SELECT
149 EPR.E_ID
150 FROM
151 ELEMENT_PROZEDUR EPR
152 WHERE
153 EPR.E_ID = ELE.E_ID
154 AND
155 EPR.P_ID = PFK.P_ID
156 )
157 )
158 -- pathologischer (Einfach-)Wert einer Prozedur
159 OR
160 (
161 PFK.P_ID IS NOT NULL
162 AND
163 PFK.PW_ID IS NOT NULL
164 AND
165 PFK.PWK_ID IS NULL
166 AND
167 ELE.E_ID IN
168 (
169 SELECT
170 EPW.E_ID
171 FROM
172 ELEMENT_PROZEDUR_WERT EPW
173 WHERE
174 EPW.E_ID = ELE.E_ID
175 AND
176 EPW.PW_ID = PFK.PW_ID
177 )
178 )
179 -- pathologischer (Kombinations-)Wert einer Prozedur
180 OR
181 (
182 PFK.P_ID IS NOT NULL
183 AND
184 PFK.PW_ID IS NULL
185 AND
186 PFK.PWK_ID IS NOT NULL
187 AND
188 ELE.E_ID IN
189 (
190 SELECT
191 EPK.E_ID
192 FROM
193 ELEMENT_PROZEDUR_WERT_KOMB EPK
194 WHERE
195 EPK.E_ID = ELE.E_ID
196 AND
197 EPK.PWK_ID = PFK.PWK_ID
198 )
199 )
200 -- Normalwert eines Komplexes
201 OR

```



```

(
PFK.KOMPLEX_KP_ID IS NOT NULL
AND
PFK.KPW_ID IS NULL
AND
ELE.E_ID IN
(
SELECT
EKX.E_ID
FROM
ELEMENT_KOMPLEX EKX
WHERE
EKX.E_ID = ELE.E_ID
AND
EKX.KP_ID = PFK.KOMPLEX_KP_ID
)
)
-- Non-Normalwert eines Komplexes
OR
(
PFK.KOMPLEX_KP_ID IS NOT NULL
AND
PFK.KPW_ID IS NOT NULL
AND
ELE.E_ID IN
(
SELECT
EKW.E_ID
FROM
ELEMENT_KOMPLEX_WERT EKW
WHERE
EKW.E_ID = ELE.E_ID
AND
EKW.KPW_ID = PFK.KPW_ID
)
)
-- Exakte Übereinstimmung eines codeunabhängigen Wertes
OR
(
PFK.W_ID IS NOT NULL
AND
PFK.WERT_WERT IS NOT NULL
AND
ELE.E_ID IN
(
SELECT
EWW.E_ID
FROM
ELEMENT_WERT_WERT EWW,
WERT_WERT WEW
WHERE
EWW.E_ID = ELE.E_ID
AND
EWW.WW_ID = WEW.WW_ID
AND
WEW.W_ID = PFK.W_ID
AND
WEW.WERT = PFK.WERT_WERT
)
)
-- Codeunabhängiger Wert liegt in einem definierten Intervall
OR
(
PFK.W_ID IS NOT NULL
AND
PFK.WERT_WERT IS NOT NULL
AND
ELE.E_ID IN

```

```
270 (
271 SELECT
272 EWB.E_ID
273 FROM
274 ELEMENT_WERT_BEREICH EWB,
275 WERT_BEREICH WEB
276 WHERE
277 EWB.E_ID = ELE.E_ID
278 AND
279 EWB.WB_ID = WEB.WB_ID
280 AND
281 WEB.W_ID = PFK.W_ID
282 AND
283 PFK.WERT_WERT BETWEEN
284 WEB.WERT_MINIMUM
285 AND
286 WEB.WERT_MAXIMUM
287 )
288 )
289 );
```

## A5.4. View [Fall\_OP\_Pruefung\_Element]

```

CREATE OR REPLACE VIEW FALL_OP_PRUEFUNG_ELEMENT AS
SELECT
PFK.FALL_ID AS FALL_ID,
PFK.FALLÖP_ID AS FALLÖP_ID,
PRF.PRF_ID AS PRF_ID,
PRF.HINWEISTEXT AS PRUEFUNG_HINWEISTEXT,
ELE.BEZEICHNUNG AS ELEMENT_BEZEICHNUNG,
PRE.HINWEISTEXT AS PRUEFUNG_ELEMENT_HINWEISTEXT,
PFK.D_ID AS D_ID,
PFK.DW_ID AS DW_ID,
PFK.DWK_ID AS DWK_ID,
PFK.DIAGNOSE_BEZEICHNUNG AS DIAGNOSE_BEZEICHNUNG,
PFK.DIAGNOSE_WERT_BEZEICHNUNG AS DIAGNOSE_WERT_BEZEICHNUNG,
PFK.DIAGNOSE_WERT AS DIAGNOSE_WERT,
PFK.P_ID AS P_ID,
PFK.PW_ID AS PW_ID,
PFK.PWK_ID AS PWK_ID,
PFK.PROZEDUR_BEZEICHNUNG AS PROZEDUR_BEZEICHNUNG,
PFK.PROZEDUR_WERT_BEZEICHNUNG AS PROZEDUR_WERT_BEZEICHNUNG,
PFK.PROZEDUR_WERT AS PROZEDUR_WERT,
PFK.KOMPLEX_KP_ID AS KP_ID,
PFK.KPW_ID AS KPW_ID,
PFK.KOMPLEX_BEZEICHNUNG AS KOMPLEX_BEZEICHNUNG,
PFK.KOMPLEX_WERT_BEZEICHNUNG AS KOMPLEX_WERT_BEZEICHNUNG,
PFK.KOMPLEX_WERT AS KOMPLEX_WERT,
PFK.W_ID AS W_ID,
PFK.WERT_BEZEICHNUNG AS WERT_BEZEICHNUNG,
PFK.WERT_WERT AS WERT_WERT,
SUK.SK_ID AS PRF_SK_ID,
SUK.BEZEICHNUNG AS PRF_SUPERKONTEXT_BEZEICHNUNG,
SKK.RANG AS PRF_SUPERKONTEXT_KONTEXT_RANG,
KPR.K_ID AS PRF_K_ID,
KON.BEZEICHNUNG AS PRF_KONTEXT_BEZEICHNUNG,
KPR.RANG AS PRF_KONTEXT_RANG,
PFK.K_ID AS ELEMENT_K_ID,
PFK.KONTEXT_BEZEICHNUNG AS ELEMENT_KONTEXT_BEZEICHNUNG,
PFK.KONTEXT_RANG AS ELEMENT_KONTEXT_RANG,
PFK.FALL_CODE AS FALL_CODE,
PFK.NACHNAME AS NACHNAME,
PFK.VORNAME AS VORNAME,
PFK.GEBURTSDATUM AS GEBURTSDATUM,
PFK.AUFNAHME_FA AS AUFNAHME_FA,
PFK.ABGANG_FA AS ABGANG_FA,
PFK.AUFNAHME_KH AS AUFNAHME_KH,
PFK.ABGANG_KH AS ABGANG_KH,
PFK.OP_CODE AS OP_CODE,
PFK.OP_BEGINN AS OP_BEGINN,
PFK.OP_ENDE AS OP_ENDE
FROM
PUFFER_FALL_OP_KONTEXT PFK,
KONTEXT_PRUEFUNG KPR,
KONTEXT KON,
SUPERKONTEXT_KONTEXT SKK,
SUPERKONTEXT SUK,
DATENEBENE_KONTEXT DEK,
PRUEFUNG PRF,
PRUEFUNG_ELEMENT PRE,
ELEMENT ELE
WHERE
PFK.K_ID = KPR.K_ID
AND
KPR.PRF_ID = PRF.PRF_ID
AND
KPR.K_ID = KON.K_ID
AND

```

```

66 KON.K_ID = SKK.K_ID
67 AND
68 SKK.SK_ID = SUK.SK_ID
69 AND
70 KON.K_ID = DEK.K_ID
71 AND
72 DEK.DATEB_ID = 2
73 AND
74 PRF.PRF_ID = PRE.PRF_ID
75 AND
76 PRE.E_ID = ELE.E_ID
77 AND
78 (
79 -- Normalwert einer Diagnose
80 (
81 PFK.D_ID IS NOT NULL
82 AND
83 PFK.DW_ID IS NULL
84 AND
85 PFK.DWK_ID IS NULL
86 AND
87 ELE.E_ID IN
88 (
89 SELECT
90 EDI.E_ID
91 FROM
92 ELEMENT_DIAGNOSE EDI
93 WHERE
94 EDI.E_ID = ELE.E_ID
95 AND
96 EDI.D_ID = PFK.D_ID
97 )
98 )
99 -- pathologischer (Einfach-)Wert einer Diagnose
100 OR
101 (
102 PFK.D_ID IS NOT NULL
103 AND
104 PFK.DW_ID IS NOT NULL
105 AND
106 PFK.DWK_ID IS NULL
107 AND
108 ELE.E_ID IN
109 (
110 SELECT
111 EDW.E_ID
112 FROM
113 ELEMENT_DIAGNOSE_WERT EDW
114 WHERE
115 EDW.E_ID = ELE.E_ID
116 AND
117 EDW.DW_ID = PFK.DW_ID
118 )
119 )
120 -- pathologischer (Kombinations-)Wert einer Diagnose
121 OR
122 (
123 PFK.D_ID IS NOT NULL
124 AND
125 PFK.DW_ID IS NULL
126 AND
127 PFK.DWK_ID IS NOT NULL
128 AND
129 ELE.E_ID IN
130 (
131 SELECT
132 EDK.E_ID
133 FROM

```

ELEMENT_DIAGNOSE_WERT_KOMB EDK	134
WHERE	135
EDK.E_ID = ELE.E_ID	136
AND	137
EDK.DWK_ID = PFK.DWK_ID	138
)	139
)	140
-- Normalwert einer Prozedur	141
OR	142
(	143
PFK.P_ID IS NOT NULL	144
AND	145
PFK.PW_ID IS NULL	146
AND	147
PFK.PWK_ID IS NULL	148
AND	149
ELE.E_ID IN	150
(	151
SELECT	152
EPR.E_ID	153
FROM	154
ELEMENT_PROZEDUR EPR	155
WHERE	156
EPR.E_ID = ELE.E_ID	157
AND	158
EPR.P_ID = PFK.P_ID	159
)	160
)	161
-- pathologischer (Einfach-)Wert einer Prozedur	162
OR	163
(	164
PFK.P_ID IS NOT NULL	165
AND	166
PFK.PW_ID IS NOT NULL	167
AND	168
PFK.PWK_ID IS NULL	169
AND	170
ELE.E_ID IN	171
(	172
SELECT	173
EPW.E_ID	174
FROM	175
ELEMENT_PROZEDUR_WERT EPW	176
WHERE	177
EPW.E_ID = ELE.E_ID	178
AND	179
EPW.PW_ID = PFK.PW_ID	180
)	181
)	182
-- pathologischer (Kombinations-)Wert einer Prozedur	183
OR	184
(	185
PFK.P_ID IS NOT NULL	186
AND	187
PFK.PW_ID IS NULL	188
AND	189
PFK.PWK_ID IS NOT NULL	190
AND	191
ELE.E_ID IN	192
(	193
SELECT	194
EPK.E_ID	195
FROM	196
ELEMENT_PROZEDUR_WERT_KOMB EPK	197
WHERE	198
EPK.E_ID = ELE.E_ID	199
AND	200
EPK.PWK_ID = PFK.PWK_ID	201

```

202 )
203 )
204 -- Normalwert eines Komplexes
205 OR
206 (
207 PFK.KOMPLEX_KP_ID IS NOT NULL
208 AND
209 PFK.KPW_ID IS NULL
210 AND
211 ELE.E_ID IN
212 (
213 SELECT
214 EKX.E_ID
215 FROM
216 ELEMENT_KOMPLEX EKX
217 WHERE
218 EKX.E_ID = ELE.E_ID
219 AND
220 EKX.KP_ID = PFK.KOMPLEX_KP_ID
221 )
222 )
223 -- Non-Normalwert eines Komplexes
224 OR
225 (
226 PFK.KOMPLEX_KP_ID IS NOT NULL
227 AND
228 PFK.KPW_ID IS NOT NULL
229 AND
230 ELE.E_ID IN
231 (
232 SELECT
233 EKW.E_ID
234 FROM
235 ELEMENT_KOMPLEX_WERT EKW
236 WHERE
237 EKW.E_ID = ELE.E_ID
238 AND
239 EKW.KPW_ID = PFK.KPW_ID
240 )
241 )
242 -- Exakte Übereinstimmung eines codeunabhängigen Wertes
243 OR
244 (
245 PFK.W_ID IS NOT NULL
246 AND
247 PFK.WERT_WERT IS NOT NULL
248 AND
249 ELE.E_ID IN
250 (
251 SELECT
252 EWW.E_ID
253 FROM
254 ELEMENT_WERT_WERT EWW,
255 WERT_WERT WEW
256 WHERE
257 EWW.E_ID = ELE.E_ID
258 AND
259 EWW.WW_ID = WEW.WW_ID
260 AND
261 WEW.W_ID = PFK.W_ID
262 AND
263 WEW.WERT = PFK.WERT_WERT
264 )
265 )
266 -- Codeunabhängiger Wert liegt in einem definierten Intervall
267 OR
268 (
269 PFK.W_ID IS NOT NULL

```

AND	270
PFK.WERT_WERT IS NOT NULL	271
AND	272
ELE.E_ID IN	273
(	274
SELECT	275
EWB.E_ID	276
FROM	277
ELEMENT_WERT_BEREICH EWB,	278
WERT_BEREICH WEB	279
WHERE	280
EWB.E_ID = ELE.E_ID	281
AND	282
EWB.WB_ID = WEB.WB_ID	283
AND	284
WEB.W_ID = PFK.W_ID	285
AND	286
PFK.WERT_WERT BETWEEN	287
WEB.WERT_MINIMUM	288
AND	289
WEB.WERT_MAXIMUM	290
)	291
)	292
);	293

## A5.5. View [Fall\_Pruefung\_Koppelung]

```
1 CREATE OR REPLACE VIEW FALL_PRUEFUNG_KOPPELUNG AS
2 SELECT
3 FAL.FALL_ID          AS FALL_ID,
4 PRF.PRF_ID          AS PRF_ID,
5 PRF.HINWEISTEXT     AS PRUEFUNG_HINWEISTEXT,
6 PKO.PRFK_ID        AS PRFK_ID,
7 PKO.BEZEICHNUNG     AS KOPPELUNG_BEZEICHNUNG,
8 PKO.HINWEISTEXT     AS KOPPELUNG_HINWEISTEXT,
9 KPR.K_ID           AS K_ID,
10 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
11 SUK.SK_ID          AS SK_ID,
12 SUK.BEZEICHNUNG    AS SUPERKONTEXT_BEZEICHNUNG,
13 KPR.RANG           AS PRUEFUNG_KONTEXT_RANG,
14 SKK.RANG           AS SUPERKONTEXT_KONTEXT_RANG,
15 FAL.CODE           AS FALL_CODE,
16 FAL.NACHNAME       AS NACHNAME,
17 FAL.VORNAME        AS VORNAME,
18 FAL.GEBURTSDATUM   AS GEBURTSDATUM,
19 FAL.AUFNAHME_FA    AS AUFNAHME_FA,
20 FAL.ABGANG_FA      AS ABGANG_FA,
21 FAL.AUFNAHME_KH    AS AUFNAHME_KH,
22 FAL.ABGANG_KH      AS ABGANG_KH
23 FROM
24 FALL                FAL,
25 KONTEXT_PRUEFUNG   KPR,
26 KONTEXT            KON,
27 SUPERKONTEXT_KONTEXT SKK,
28 SUPERKONTEXT      SUK,
29 DATENEBENE_KONTEXT DEK,
30 PRUEFUNG           PRF,
31 PRUEFUNG_KOPPELUNG PKO
32 WHERE
33 KPR.PRF_ID = PRF.PRF_ID
34 AND
35 KPR.K_ID = KON.K_ID
36 AND
37 KON.K_ID = SKK.K_ID
38 AND
39 SKK.SK_ID = SUK.SK_ID
40 AND
41 DEK.K_ID = KON.K_ID
42 AND
43 DEK.DATEB_ID = 1
44 AND
45 PRF.PRF_ID = PKO.PRF_ID
46 AND
47 (
48 -- -----
49 -- mindestens eines der gekoppelten Elemente muss dem
50 -- Fall zugeordnet sein = vorhandene(s) Element(e)
51 -- -----
52 PKO.PRFK_ID IN
53 (
54 SELECT
55 PKE.PRFK_ID
56 FROM
57 PRUEFUNG_KOPPELUNG_ELEMENTE PKE,
58 PUFFER_FALL_ELEMENT        PFE
59 WHERE
60 PFE.FALL_ID = FAL.FALL_ID
61 AND
62 PKE.PRFK_ID = PKO.PRFK_ID
63 AND
64 PFE.E_ID = PKE.E_ID
65 )
```



```

-- -----
-- mindestens eines der gekoppelten Elemente darf dem
-- Fall nicht zugeordnet sein = fehlende(s) Element(e)
-- -----
AND
(
PKO.PRFK_ID IN
(
SELECT
PKE.PRFK_ID
FROM
PRUEFUNG_KOPPELUNG_ELEMENTE PKE
WHERE
PKE.PRFK_ID = PKO.PRFK_ID
AND
PKE.E_ID NOT IN
(
SELECT
PFE.E_ID
FROM
PUFFER_FALL_ELEMENT PFE
WHERE
PFE.FALL_ID = FAL.FALL_ID
AND
PFE.E_ID = PKE.E_ID
)
)
)
);

```

66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94

## A5.6. View [Fall\_OP\_Pruefung\_Koppelung]

```
1 CREATE OR REPLACE VIEW FALL_OP_PRUEFUNG_KOPPELUNG AS
2 SELECT
3 FAL.FALL_ID          AS FALL_ID,
4 FOP.FALLÖP_ID       AS FALLÖP_ID,
5 PRF.PRF_ID          AS PRF_ID,
6 PRF.HINWEISTEXT     AS PRUEFUNG_HINWEISTEXT,
7 PKO.PRFK_ID         AS PRFK_ID,
8 PKO.BEZEICHNUNG     AS KOPPELUNG_BEZEICHNUNG,
9 PKO.HINWEISTEXT     AS KOPPELUNG_HINWEISTEXT,
10 KPR.K_ID            AS K_ID,
11 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
12 SUK.SK_ID          AS SK_ID,
13 SUK.BEZEICHNUNG     AS SUPERKONTEXT_BEZEICHNUNG,
14 KPR.RANG            AS PRUEFUNG_KONTEXT_RANG,
15 SKK.RANG            AS SUPERKONTEXT_KONTEXT_RANG,
16 FAL.CODE            AS FALL_CODE,
17 FAL.NACHNAME        AS NACHNAME,
18 FAL.VORNAME         AS VORNAME,
19 FAL.GEBURTSDATUM   AS GEBURTSDATUM,
20 FAL.AUFNAHME_FA     AS AUFNAHME_FA,
21 FAL.ABGANG_FA       AS ABGANG_FA,
22 FAL.AUFNAHME_KH     AS AUFNAHME_KH,
23 FAL.ABGANG_KH       AS ABGANG_KH,
24 FOP.CODE            AS OP_CODE,
25 FOP.BEGINN          AS OP_BEGINN,
26 FOP.ENDE            AS OP_ENDE
27 FROM
28 FALL                FAL,
29 FALL_OP             FOP,
30 KONTEXT_PRUEFUNG   KPR,
31 KONTEXT             KON,
32 SUPERKONTEXT_KONTEXT SKK,
33 SUPERKONTEXT       SUK,
34 DATENEBENE_KONTEXT DEK,
35 PRUEFUNG            PRF,
36 PRUEFUNG_KOPPELUNG PKO
37 WHERE
38 FAL.FALL_ID = FOP.FALL_ID
39 AND
40 KPR.PRF_ID = PRF.PRF_ID
41 AND
42 KPR.K_ID = KON.K_ID
43 AND
44 KON.K_ID = SKK.K_ID
45 AND
46 SKK.SK_ID = SUK.SK_ID
47 AND
48 DEK.K_ID = KON.K_ID
49 AND
50 DEK.DATEB_ID = 2
51 AND
52 PRF.PRF_ID = PKO.PRF_ID
53 AND
54 (
55  -- -----
56  -- mindestens eines der gekoppelten Elemente muss dem
57  -- Fall zugeordnet sein = vorhandene(s) Element(e)
58  -- -----
59  PKO.PRFK_ID IN
60  (
61  SELECT
62  PKE.PRFK_ID
63  FROM
64  PRUEFUNG_KOPPELUNG_ELEMENTE PKE,
65  PUFFER_FALL_OP_ELEMENT     PFE
```

WHERE	66
PFE.FALL_ID = FAL.FALL_ID	67
AND	68
PFE.FALLOP_ID = FOP.FALLOP_ID	69
AND	70
PKE.PRFK_ID = PKO.PRFK_ID	71
AND	72
PFE.E_ID = PKE.E_ID	73
)	74
-- -----	75
-- mindestens eines der gekoppelten Elemente darf dem	76
-- Fall nicht zugeordnet sein = fehlende(s) Element(e)	77
-- -----	78
AND	79
(	80
PKO.PRFK_ID IN	81
(	82
SELECT	83
PKE.PRFK_ID	84
FROM	85
PRUEFUNG_KOPPELUNG_ELEMENTE PKE	86
WHERE	87
PKE.PRFK_ID = PKO.PRFK_ID	88
AND	89
PKE.E_ID NOT IN	90
(	91
SELECT	92
PFE.E_ID	93
FROM	94
PUFFER_FALL_OP_ELEMENT PFE	95
WHERE	96
PFE.FALL_ID = FAL.FALL_ID	97
AND	98
PFE.FALLOP_ID = FOP.FALLOP_ID	99
AND	100
PFE.E_ID = PKE.E_ID	101
)	102
)	103
)	104
);	105

## A5.7. View [Fall\_Pruefung\_Ausschluss]

```
1 CREATE OR REPLACE VIEW FALL_PRUEFUNG_AUSSCHLUSS AS
2 SELECT
3 FAL.FALL_ID          AS FALL_ID,
4 PRF.PRF_ID          AS PRF_ID,
5 PRF.HINWEISTEXT     AS PRUEFUNG_HINWEISTEXT,
6 PAU.PRFA_ID        AS PRFA_ID,
7 PAU.BEZEICHNUNG     AS AUSSCHLUSS_BEZEICHNUNG,
8 PAU.HINWEISTEXT     AS AUSSCHLUSS_HINWEISTEXT,
9 KPR.K_ID           AS K_ID,
10 KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
11 SUK.SK_ID          AS SK_ID,
12 SUK.BEZEICHNUNG     AS SUPERKONTEXT_BEZEICHNUNG,
13 KPR.RANG           AS PRUEFUNG_KONTEXT_RANG,
14 SKK.RANG           AS SUPERKONTEXT_KONTEXT_RANG,
15 FAL.CODE           AS FALL_CODE,
16 FAL.NACHNAME       AS NACHNAME,
17 FAL.VORNAME        AS VORNAME,
18 FAL.GEBURTSDATUM   AS GEBURTSDATUM,
19 FAL.AUFNAHME_FA    AS AUFNAHME_FA,
20 FAL.ABGANG_FA      AS ABGANG_FA,
21 FAL.AUFNAHME_KH    AS AUFNAHME_KH,
22 FAL.ABGANG_KH      AS ABGANG_KH
23 FROM
24 FALL                FAL,
25 KONTEXT_PRUEFUNG   KPR,
26 KONTEXT            KON,
27 SUPERKONTEXT_KONTEXT SKK,
28 SUPERKONTEXT       SUK,
29 DATENEBENE_KONTEXT DEK,
30 PRUEFUNG            PRF,
31 PRUEFUNG_AUSSCHLUSS PAU
32 WHERE
33 KPR.PRF_ID = PRF.PRF_ID
34 AND
35 KPR.K_ID = KON.K_ID
36 AND
37 KON.K_ID = SKK.K_ID
38 AND
39 SKK.SK_ID = SUK.SK_ID
40 AND
41 DEK.K_ID = KON.K_ID
42 AND
43 DEK.DATEB_ID = 1
44 AND
45 PRF.PRF_ID = PAU.PRF_ID
46 AND
47 (
48  -- -----
49  -- es müssen mindestens 2 einander ausschliessende Elemente dem Fall zugeordnet sein
50  -- -----
51  1 <
52  (
53  SELECT
54  COUNT (DISTINCT (PAE.E_ID))
55  FROM
56  PRUEFUNG_AUSSCHLUSS_ELEMENTE PAE,
57  PUFFER_FALL_ELEMENT          PFE
58  WHERE
59  PFE.FALL_ID = FAL.FALL_ID
60  AND
61  PAE.PRFA_ID = PAU.PRFA_ID
62  AND
63  PFE.E_ID = PAE.E_ID
64  )
65  );
```

## A5.8. View [Fall\_OP\_Pruefung\_Ausschluss]

```

CREATE OR REPLACE VIEW FALL_OP_PRUEFUNG_AUSSCHLUSS AS
SELECT
FAL.FALL_ID          AS FALL_ID,
FOP.FALLÖP_ID       AS FALLÖP_ID,
PRF.PRF_ID          AS PRF_ID,
PRF.HINWEISTEXT     AS PRUEFUNG_HINWEISTEXT,
PAU.PRFA_ID        AS PRFA_ID,
PAU.BEZEICHNUNG     AS AUSSCHLUSS_BEZEICHNUNG,
PAU.HINWEISTEXT     AS AUSSCHLUSS_HINWEISTEXT,
KPR.K_ID           AS K_ID,
KON.BEZEICHNUNG     AS KONTEXT_BEZEICHNUNG,
SUK.SK_ID          AS SK_ID,
SUK.BEZEICHNUNG     AS SUPERKONTEXT_BEZEICHNUNG,
KPR.RANG           AS PRUEFUNG_KONTEXT_RANG,
SKK.RANG           AS SUPERKONTEXT_KONTEXT_RANG,
FAL.CODE           AS FALL_CODE,
FAL.NACHNAME       AS NACHNAME,
FAL.VORNAME        AS VORNAME,
FAL.GEBURTSDATUM   AS GEBURTSDATUM,
FAL.AUFNAHME_FA    AS AUFNAHME_FA,
FAL.ABGANG_FA      AS ABGANG_FA,
FAL.AUFNAHME_KH    AS AUFNAHME_KH,
FAL.ABGANG_KH      AS ABGANG_KH,
FOP.CODE           AS OP_CODE,
FOP.BEGINN         AS OP_BEGINN,
FOP.ENDE           AS OP_ENDE
FROM
FAL
      FALL
FALL_OP
      FOP,
KONTEXT_PRUEFUNG
      KPR,
KONTEXT
      KON,
SUPERKONTEXT_KONTEXT
      SKK,
SUPERKONTEXT
      SUK,
DATENEBENE_KONTEXT
      DEK,
PRUEFUNG
      PRF,
PRUEFUNG_AUSSCHLUSS
      PAU
WHERE
FAL.FALL_ID = FOP.FALL_ID
AND
KPR.PRF_ID = PRF.PRF_ID
AND
KPR.K_ID = KON.K_ID
AND
KON.K_ID = SKK.K_ID
AND
SKK.SK_ID = SUK.SK_ID
AND
DEK.K_ID = KON.K_ID
AND
DEK.DATEB_ID = 2
AND
PRF.PRF_ID = PAU.PRF_ID
AND
(
-- -----
-- es müssen mindestens 2 einander ausschliessende Elemente dem Fall zugeordnet sein
-- -----
1 <
(
SELECT
COUNT (PAE.E_ID)
FROM
PRUEFUNG_AUSSCHLUSS_ELEMENTE PAE,
PUFFER_FALL_OP_ELEMENT       PFE
WHERE

```

```
66 PFE.FALL_ID = FAL.FALL_ID
67 AND
68 PFE.FALLOP_ID = FOP.FALLOP_ID
69 AND
70 PAE.PRFA_ID = PAU.PRFA_ID
71 AND
72 PFE.E_ID = PAE.E_ID
73 )
74 );
```

# Anhang 6: Exkurs – Design relationaler Datenbanken und Normalisierung

Der Text in diesem Abschnitt entstammt einer vom Autor erstellten Schulungsunterlage für die Teilnehmer eines von ihm durchgeführten Seminars, das anlässlich des Workshops „Dokumentationsverfahren in der Herzchirurgie“ im Jahr 2000 stattfand.

---

## **Kursteil Datenbankdesign: Übersicht**

- Datenbanken / Datenbankmanagementsysteme
- Grundbegriffe des Relationenmodells
- Graphische Darstellung: „Krähenfuß – Notation“
- Relationale Datenbanken
- Vom Entitäts – Beziehungsmodell zum Datenbankschema
- Normalisierung
- Mutationsanomalien
- Normalformen (1. – 3.)

Für die Erstellung dieses Skriptes wurden folgende Quellen verwendet:

- „Datenbanksysteme“ von Gunter Schlageter, Fernuniversität / Gesamthochschule Hagen\_
- „Relationale Datenbanken – Eine Einführung für die Praxis“ von Andreas Meier (Springer Verlag)
- ORACLE – Schulung - CASE 1 / Theorie: Modellierungstechniken

## **Definition Datenbank:**

*Integrierte Ansammlung von Daten, die allen Benutzern eines Anwendungsbereiches als gemeinsame Basis aktueller Informationen dient.*

*„integriert“:*

Daten sind den natürlichen Zusammenhängen in der Anwendungswelt (=Ausschnitt der realen Welt, auch als „Miniwelt“ bezeichnet) entsprechend strukturiert (und nicht etwa danach, wie einzelne Anwendungen diese Daten benötigen!); eine Datenbank modelliert bestimmte Aspekte der Realwelt.

*„gemeinsame Basis“:*

Daten in der Datenbank können durch viele Benutzer genutzt werden, u.U. sogar gleichzeitig (paralleler Zugriff), wobei das Datenbanksystem eine gegenseitige Störung der Anwender verhindert. Verschiedene Anwendungen haben unterschiedliche Sichten auf ihre gemeinsame Datenbasis.

Daten werden nicht länger als Anhängsel der Anwendungsprogramme angesehen, sondern als eigenständige wesentliche Betriebsmittel des Unternehmens.

## **Definition Datenbankmanagementsystem (DBMS):**

*Softwaresystem, welches es ermöglicht, eine Datenbank zu definieren (Informationseinheiten, Beziehungen zwischen diesen), Daten zu speichern, zu verändern und zu löschen sowie diese abzufragen.*



## **Vorteile des Datenbank- / DBMS - Konzeptes:**

### **Vermeidung von Redundanz und Inkonsistenz**

Daten werden zentral verwaltet und nur einmal definiert und erfasst (Konsistenz: logische Übereinstimmung bzw. Widerspruchsfreiheit von Daten).

### **Datenunabhängigkeit der Anwendungen**

Anwendungsprogramme greifen nicht direkt auf die Datenbank zu, sondern kommunizieren mit dem DBMS, das Anfragen in entsprechende Zugriffe auf die Datenbank umsetzt und die Daten in der jeweils geeigneten Form (Sicht!) zur Verfügung stellt (nicht wie bisher in Form von ganzen Sätzen). Auf diese Weise wird die Datenbank von den Anwendungsprogrammen isoliert, so dass der Entwickler deren Details, insbesondere Details über die physische Organisation der Daten auf den Trägern, nicht kennen muss („Black Box“).

### **Flexibilität**

Innerhalb der Datenbank liegen sämtliche verfügbaren Daten in überschaubarer Form vor und können daher unter Verwendung eines standardisierten Zugriffsverfahrens im Prinzip in beliebiger – und vor allem nicht vorausgeplanter - Weise ausgewertet werden.

### **Gewährleistung der Integrität der Daten**

Das DBMS prüft zentral die Integrität von Daten und verwehrt solche Änderungen von Daten, die zur Verletzung der vorgegebenen Integritätsbedingungen führen würden (Integrität: Korrektheit und Vollständigkeit der abgespeicherten Daten).

### **Schutz der Daten gegen unberechtigten Zugriff**

Mechanismen zur Gewährung bzw. Verweigerung von Zugriffen können auf Ebene des DBMS implementiert werden.

### **Koordination von gleichzeitig auf der Datenbank arbeitenden Benutzern**

Das DBMS gewährleistet den störungsfreien Betrieb der parallel ablaufenden Anwendungen.

### **Datensicherheit**

Mechanismen zur Wiederherstellung einer korrekten Datenbank nach dem Auftreten von Fehlern können auf Ebene des DBMS implementiert werden.

## Grundbegriffe des Relationenmodells

### Entität (engl. „entity“) / Entitäts - Typ / Attribut:

Entität = identifizierbares Objekt der Realwelt (oder unserer Vorstellung von dieser); der Entitäts - Typ repräsentiert die Menge aller Objekte, die die gleichen, durch ihn charakterisierten Eigenschaften oder Merkmale (=Attribute) besitzen.

Beispiel:

- Entität <PATIENT>, charakterisiert durch die Attribute <PATNR>, <NAME> etc.
- Entität <BEHANDLUNG>, charakterisiert durch die Attribute <PATNR> <BEHNR>, <AUFNAHMEGRUND> etc.

### Beziehung (engl. „relationship“):

Beschreibung des Verhältnisses, in dem Entitäten zueinander stehen; die Beziehungen zwischen Entitäten können ihrerseits selbst Mengen bilden (also Entitäten sein!) und durch (zusätzliche) Attribute beschrieben werden.

Folgende Beziehungsgrade werden unterschieden:

- 1 : 1 - Beziehung: einer Entität aus der Menge 1 ist **genau eine** Entität aus der Menge 2 zugeordnet (Beispiel: „ein Patient hat höchstens einen Patienten-Telefonanschluß“ bzw. „ein Patienten-Telefonanschluß ist höchstens einem Patienten zugeordnet“).
- 1 : n - Beziehung: einer Entität aus der Menge 1 sind **eine oder mehrere** Entitäten aus der Menge 2 zugeordnet (Beispiel: „ein Patient wird einmal oder mehrfach behandelt“).
- n : m - Beziehung: einer Entität aus der Menge 1 sind **eine oder mehrere** Entitäten aus der Menge 2 zugeordnet **und umgekehrt** (Beispiel: „zu einer Behandlung bzw. einem Krankenhausaufenthalt werden eine oder mehrere ICD - Diagnosen angegeben, umgekehrt kann ein und dieselbe ICD – Diagnose in einer oder mehreren Behandlungen auftauchen“).

n : m – Beziehungen werden in der Regel in zwei Beziehungen vom Typ 1 : n aufgelöst (dies wird später am oben genannten Beispiel demonstriert!).

## **Relation:**

Eine Relation ist eine Menge von Tupeln (=„Datensätze“), die tabellenförmig (Spalten bzw. „Felder“ entsprechen hierbei den Attributen!) dargestellt werden kann und sich durch folgende Eigenschaften auszeichnet:

- besitzt einen eindeutigen Namen
- innerhalb der Tabelle ist jeder Merkmals- oder „Feld“ – Name eindeutig und bezeichnet eine bestimmte „Spalte“ mit einer bestimmten Eigenschaft
- die Anzahl der Merkmale ist beliebig
- die Ordnung der Spalten innerhalb der Relation ist bedeutungslos
- jedes Tupel beschreibt ein unterscheidbares Objekt der Anwendungswelt
- für jedes Tupel innerhalb der Tabelle existiert ein Identifikator, der als Primärschlüssel bezeichnet wird (siehe auch Definition „Primärschlüssel“)
- die Anzahl der Tupel innerhalb der Relation ist beliebig
- die Ordnung der Tupel innerhalb der Tabelle ist bedeutungslos.

Eine Relation ist demnach als eine Menge ungeordneter Tupel aufzufassen; aufgrund dieses Mengenbegriffs kann ein und dasselbe Tupel nur jeweils einmal in einer Relation vorkommen.

## **Primärschlüssel:**

*Eines der Merkmale oder eine Kombination von mehreren Merkmalen dient als eindeutiger Identifikator für jedes einzelne Tupel innerhalb der Tabelle und wird als Primärschlüssel bezeichnet (= „Eindeutigkeit“).*

*Falls der Schlüssel aus einer Kombination von Merkmalen besteht, so muss diese Kombination minimal sein, d.h. kein Merkmal kann aus dieser Kombination entfernt werden, ohne dass die Eindeutigkeit verloren geht (= „Minimalität“).*

### Beispiel:

Gegeben sei die folgende Relation mit Patienten – Daten....

### **PATIENT**

<b>PATNR</b>	<b>NAME</b>	<b>VORNAME</b>	<b>GEBURTSDATUM</b>	<b>GESCHLECHT</b>
1	Korbma che r	Karl	12.07.1948	m
2	Drechsler	Heinrich	12.07.1948	m
3	Hoffmann	Irene	22.11.1932	w

Die Patienten – Nummern werden fortlaufend vergeben. Zunächst wird (fälschlich!) die Kombination aus PATNR und GEBURTSDATUM als Primärschlüssel gewählt; anschließend wird geprüft, ob der Schlüssel minimal ist: testweise wird das Attribut PATNR aus dem Schlüssel gestrichen – dies führt jedoch – da z.B. mehrere Patienten am gleichen Tag geboren worden sein können (im obigen Beispiel die Patienten „1“ und „2“) - dazu, dass nicht mehr jedes Tupel eindeutig identifiziert werden kann (der Wert „12.07.1948“ identifiziert z.B. 2 Tupel!). Umgekehrt ist es jedoch sehr wohl möglich, das Attribut GEBURTSDATUM aus dem Primärschlüssel zu streichen, da anschließend immer noch jedes Tupel eindeutig über seine Patienten – Nummer identifiziert werden. Der zusammengesetzte Schlüssel PATNR | GEBURTSDATUM erfüllt das Gebot der Minimalität also nicht, der lediglich aus dem Attribut PATNR bestehende Primärschlüssel hingegen schon.

*Statt eines natürlichen Merkmals (Beispiel Patient: Name des Patienten) bzw. einer Kombination von natürlichen Merkmalen kann auch ein künstliches Merkmal (oder eine Kombination von natürlichen Merkmalen) verwendet werden (Beispiel Patient: Patienten – Nr.).*

Ein künstlicher Schlüssel sollte anwendungsneutral sein und keine eigene Aussagekraft besitzen; auf diese Weise werden Interpretationsspielräume eingeschränkt und mit der Zeit auftretende Bedeutungsverluste von ursprünglich festgelegten Definitionen unterbunden.

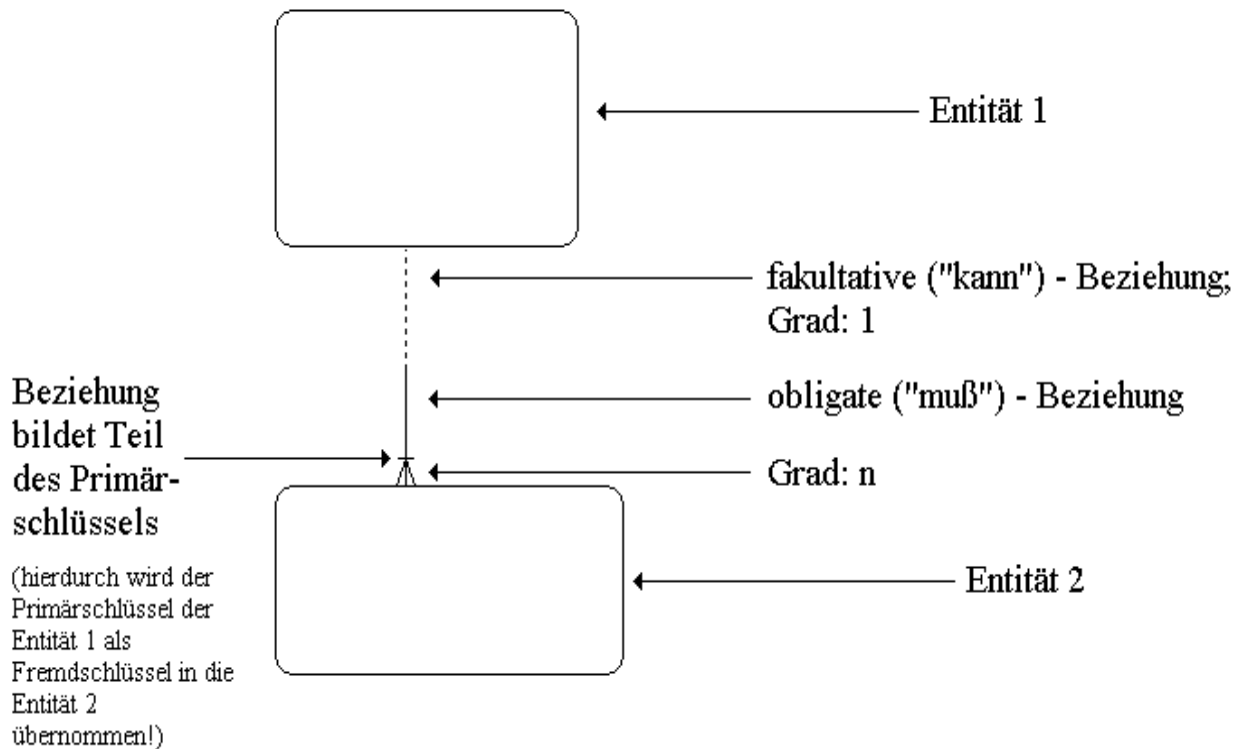
## **Fremdschlüssel:**

*Attribut(e) einer Tabelle, welche(s) als Primärschlüssel in einer anderen Tabelle auftritt / auftreten.*

## Entitäts - Beziehungsmodell:

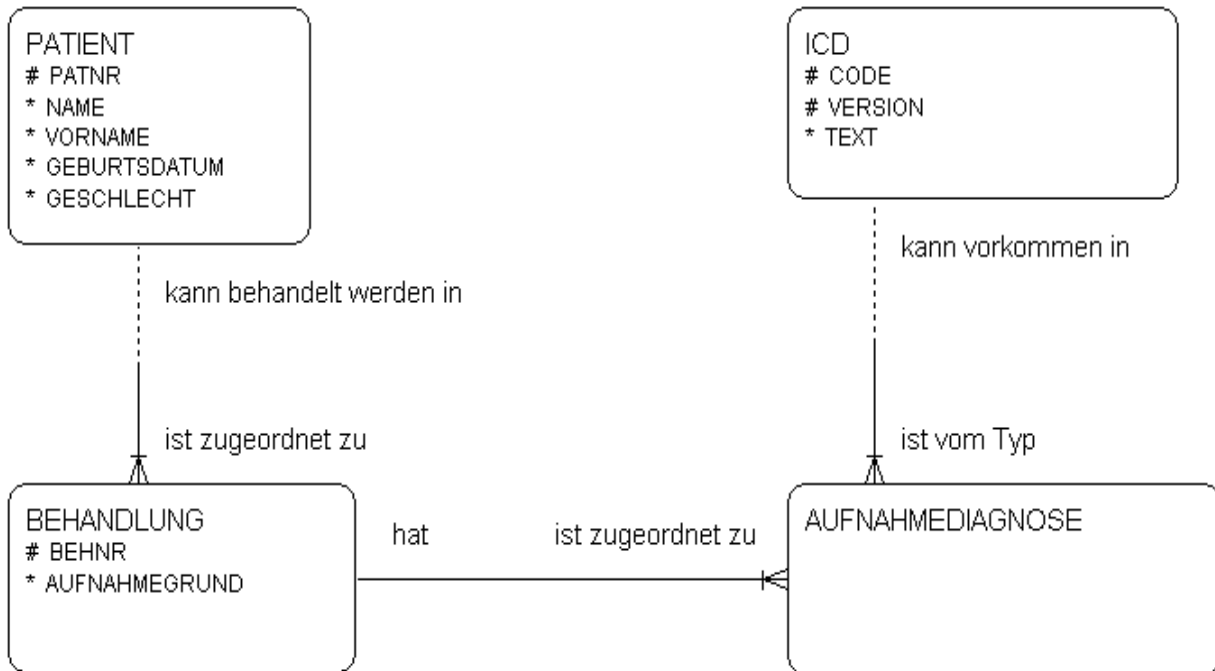
*In einem solchen Modell wird ein Bereich der Realwelt als Sammlung von Entitätsmengen und Beziehungen zwischen diesen dargestellt .*

Zur Darstellung von Entitäts – Beziehungs - (engl.: entity / relationship = E/R -) Modellen existieren verschiedene graphische Notationen, von denen nachfolgend exemplarisch die sogenannte „Krähenfuß – Notation“ vorgestellt werden soll.



## Beispiel für ein mit Hilfe der sog. „Krähenfuß – Notation“ dargestelltes Entitäts - Beziehungsmodell

Das unten abgebildete Entitäts – Beziehungsmodell soll Datenstrukturen für die Erfassung der Aufnahmediagnosen nach § 301 SGB V beschreiben. (Hierbei wird jedoch kein Anspruch auf Korrektheit oder Vollständigkeit erhoben!)



## **Relationale Datenbanken:**

*Die Daten der Realwelt werden als eine Sammlung von Relationen dargestellt; eine Relation kann man sich anschaulich als eine Tabelle vorstellen.*

*Sämtliche Informationen, d.h. sowohl Entitäten als auch die Beziehungen zwischen diesen, werden über solche Relationen bzw. Tabellen modelliert, wobei die einzelnen Realwelt - Objekte durch Tupel („Zeilen“ oder „Datensätze“ der Tabellen) repräsentiert werden.*

*Die Eigenschaften der Entitäten werden durch das Relationsschema beschrieben, das die Attribute der jeweiligen Relation spezifiziert; hierbei wird jedem Attribut ein bestimmter Wertebereich (=„domain“) zugeordnet.*

*Die Gesamtheit der Relationenschemata heißt Schema der relationalen Datenbank; zu diesem zählen darüber hinaus die Integritätsbedingungen, die Aussagen über die Korrektheit von Werten machen bzw. diese gewährleisten.*

*Entitäten ebenso wie Beziehungen werden durch Relationen ausgedrückt.*

*Die Definition des DBMS wird um folgende Punkte zum Konstrukt RDBMS (=relationales Datenbankmanagementsystem) erweitert:*

- *basiert auf einer klaren formalen Grundlage („alle Informationen werden in Relationen abgespeichert“)*
- *unterstützt eine mengenorientierte Sprachschnittstelle zur Datendefinition, - Manipulation und – Selektion (z.B. SQL)*
- *bei Abfragen formuliert der Anwender seine Selektionsbedingung, während das RDBMS nach Absetzen des entsprechenden Befehls die Durchsuchung der Datenbank sowie die Bereitstellung der Ergebnis – Relation (=eine Menge von Tupeln!) übernimmt.*

# Vom Entitäts – Beziehungsmodell zum Datenbankschema

## Schritt 1 – Abbildung der Entitätsmengen:

- **Jede** (zuvor identifizierte) Entitätsmenge wird als eigenständige Tabelle mit einem eindeutigen Schlüssel angelegt.
- Die Merkmale der Entitätsmenge werden in die entsprechenden Attribute der Tabelle umgesetzt, wobei für jedes Attribut der jeweilige Wertebereich / Wertevorrat („domain“) festgelegt wird.

### Beispiel:

Entitätsmenge PATIENT mit den Attributen PATNR, NAME, VORNAME, GEBURTSDATUM, GESCHLECHT:

- das Attribut PATNR bildet den Primärschlüssel für die Tabelle PATIENT
- die Wertemenge für PATNR ist als numerischer Wert, die Wertemengen für NAME, VORNAME und GESCHLECHT sind als Zeichenketten und die Wertemenge für GEBURTSDATUM ist als Datumswert definiert.

### **PATIENT**

<b>PATNR</b>	<b>NAME</b>	<b>VORNAME</b>	<b>GEBURTSDATUM</b>	<b>GESCHLECHT</b>
11	Maier	Hans	13.05.1951	m
12	Müller	Rudolf	22.11.1938	m
13	Schmidt	Dieter	04.01.1960	w



## Schritt 2 - Abbildung der Beziehungsmengen:

- Jede Beziehungsmenge **kann** als eigenständige Tabelle angelegt werden, wobei in diesem Fall die Primärschlüssel der beteiligten Entitätsmengen bzw. Tabellen als Fremdschlüssel auftauchen **müssen**.
  - Der Primärschlüssel für eine solche, eine Beziehung abbildende Tabelle kann sich aus den Fremdschlüsseln zusammensetzen (=Regelfall, trifft auch im o.g. Beispiel zu) oder künstlich gebildet werden.
  - Weitere Merkmale der Beziehungsmenge erscheinen als zusätzliche Felder in der Tabelle (z.B. könnte in der Tabelle AUFNAHMEDIAGNOSE noch ein Feld DATUM für die Dokumentation des Zeitpunktes der Erfassung aufgenommen werden).
1. 1 : 1 – Beziehungen **können ohne eigenständige Tabellen** abgebildet werden, indem der Primärschlüssel der einen beteiligten Tabelle als Fremdschlüssel in die andere Tabelle aufgenommen wird.
  2. 1 : n – Beziehungen **können ebenfalls ohne eigenständige Tabellen** abgebildet werden, indem der Primärschlüssel der Grad 1 – Tabelle als Fremdschlüssel in die Grad n – Tabelle aufgenommen wird (Beispiel: der Primärschlüssel PATNR der Tabelle Patient tritt als Fremdschlüssel und hierbei **als Teil** des Primärschlüssels der Tabelle BEHANDLUNG auf).
  3. n : m – Beziehungen **müssen** als eigenständige Tabellen abgebildet werden, wobei die Primärschlüssel der beteiligten Tabellen als Fremdschlüssel in die Beziehungs – Tabelle aufgenommen werden und dort in der Regel den Primärschlüssel bilden (z.B. müssen in der Tabelle AUFNAHMEDIAGNOSE folgende Fremdschlüssel – Attribute existieren: PATNR und BEHANDNR {Primärschlüssel der Tabelle BEHANDLUNG} sowie CODE und VERSION {Primärschlüssel der Tabelle ICD}).

## **Normalisierung**

*Entwurfs- bzw. Prüfungsverfahren zur Untersuchung von Abhängigkeiten innerhalb von Tabellen, mit dessen Hilfen Redundanz von Informationen und damit zusammenhängende (Mutations-) Anomalien vermieden werden sollen (ein Attribut einer Tabelle ist redundant, wenn bei einzelnen Tupeln dieser Tabelle Werte dieses Attributs wegfallen können, ohne dass hierbei ein Informationsverlust entsteht).*

Für die meisten Praxisfälle genügt eine Beschränkung auf die ersten 3 Normalformen.

## **Mutationsanomalien**

Zur Demonstration der verschiedenen Anomalie – Formen soll die unten dargestellte (in vielfacher Hinsicht falsch modellierte) Tabelle dienen, in der die Informationen von Patienten und Hausärzten gespeichert werden; (künstlicher) Primärschlüssel der Tabelle soll das Attribut PATNR sein.

PATNR	NAME	VORNAME	HAUSARZT
11	Maier	Hans	Dr. Müller, Bahnfostr. 22, 42657 Redundanz
12	Müller	Rudolf	Dr. Schulz, Allee 12, 42658 Redundanz
13	Schmidt	Dieter	Dr. Schulz, Allee 12, 42658 Redundanz

## **Einfügeanomalien**

*Beim Vorliegen einer Einfügeanomalie ist es unmöglich, die Daten eines Realwelt – Objektes (=einer Entität) unabhängig von den Daten eines anderen Realwelt – Objektes festzuhalten.*

Bei Verwendung des obigen Relationsschemas ist es beispielsweise nicht möglich, die Daten eines weiteren Hausarztes zu erfassen, wenn nicht gleichzeitig die Daten eines Patienten eingetragen werden.

## **Löschanomalien**

*Beim Vorliegen einer Löschanomalie kommt es – gleichsam als Seiteneffekt - zu ungewollten Verlusten von Sachverhalten.*

Würde im obigen Beispiel der Patient „Maier“ mit der Patienten – Nr. „11“ gelöscht, wären gleichzeitig die Daten des Hausarztes „Dr. Müller“ verloren.

## **Änderungsanomalien**

*Von Änderungsanomalien spricht man, wenn bei Änderung nur eines einzigen Sachverhaltes Manipulationen an mehreren Tupeln notwendig wären.*

Im obigen Beispiel würde ein Umzug des Hausarztes „Dr. Schulz“ mit entsprechender Änderung der Anschrift Korrekturen an den Tupeln mit den Patienten – Nummern „12“ und „13“ erfordern.

# Die Normalformen

## 1. Normalform (1NF)

*Eine Relation befindet sich in erster Normalform, wenn jedes seiner Attribut atomar, d.h. nicht weiter teilbar ist; insbesondere dürfen einzelne Attribute nicht aus zusammengesetzten Werten, Mengen, Aufzählungstypen oder Wiederholungsgruppen bestehen.*

Beispiel:

Nicht erlaubt wäre ein Relationsschema folgender Art:

PATNR	NAME_VORNAME
11	Maier Hans
12	Müller#Rudolf
13	Schmidt / Dieter

Ausgehend vom obigen Beispiel wäre folgendes Relationsschema hingegen korrekt im Sinne der ersten Normalform:

PATNR	NAME	VORNAME
11	Maier	Hans
12	Müller	Rudolf
13	Schmidt	Dieter

## **2. Normalform (2NF)**

*Ein Relation befindet sich in zweiter Normalform, wenn sie sich in erster Normalform befindet und zusätzlich jedes Tupel einer Relation über einen Schlüssel (=entweder ein einzelnes Attribut oder eine Kombination aus mehreren) einwandfrei identifiziert werden kann; anders ausgedrückt: jedes Attribut, das nicht Bestandteil des Primärschlüssels der Relation ist, muss voll funktional abhängig vom Primärschlüssel sein – jeder Identifikator der Tabelle beschreibt also **genau einen** Wert eines beliebigen Nichtschlüsselmerkmals.*

### Beispiele:

<PATNR> ist Schlüssel für PATIENT; in PATIENT können nicht zwei oder mehr Einträge mit dem gleichen Wert für <PATNR> vorkommen. Die Patienten - Nummer „13“ beschreibt genau einen Wert des Attributs VORNAME, nämlich „Dieter“.

### **PATIENT**

<b>PATNR</b>	<b>NAME</b>	<b>VORNAME</b>
11	Maier	Hans
12	Müller	Rudolf
13	Schmidt	Dieter

<PATNR, BEHANDNR> ist Schlüssel für BEHANDLUNG; in BEHANDLUNG können nicht zwei oder mehr Einträge mit dem gleichen Wert für <PATNR, BEHANDNR> vorkommen. Die Kombination aus der Patienten - Nummer „13“ und der Behandlungen – Nummer „1“ beschreibt genau einen Wert des Attributs AUFNAHMEGRUND, nämlich „01“.

### **BEHANDLUNG**

<b>PATNR</b>	<b>BEHNR</b>	<b>AUFNAHMEGRUND</b>
11	1	01
11	2	07
12	1	01
13	1	01

Besondere Beachtung verdient der Begriff der vollen funktionalen Abhängigkeit bei zusammengesetzten Schlüsseln.

*Damit eine Relation mit einem solchen Schlüssel die Kriterien der zweiten Normalform erfüllt, darf nur über die Gesamtheit der Schlüsselattribute eine eindeutige Bestimmung der Nichtschlüsselattribute möglich sein; anders ausgedrückt: die volle funktionale Abhängigkeit eines Merkmales von diesem zusammengesetzten Schlüssel verbietet es, dass es lediglich von einem Teil dieses Schlüssels abhängig ist.*

Beispiel:

In der unten dargestellten Tabelle sollen die Aufnahme – Diagnosen zu den einzelnen Behandlungen eines Patienten erfasst werden; der Primärschlüssel dieser Relation wird von der Kombination der Attribute PATNR, BEHNR, CODE und VERSION gebildet.

Das Merkmal (ICD - Diagnose-) TEXT ist jedoch funktional abhängig von nur einem Teil des Primärschlüssels, nämlich den Attributen (ICD-) CODE und (ICD-) VERSION.

PATNR	BEHNR	CODE	VERSION	TEXT
11	1	I25.1	10	KHK
11	1	I08.0	10	Mitral- und Aortenvitium
12	1	I25.1	10	KHK
13	1	I65.2	10	Stenose der A. carotis

Um diese Relation in die zweite Normalform zu überführen, muss eine Zerlegung vorgenommen werden:

CODE	VERSION	TEXT
I25.1	10	KHK
I08.0	10	Mitral- und Aortenvitium
I65.2	10	Stenose der A. carotis

PATNR	BEHNR	CODE	VERSION
11	1	I25.1	10
11	1	I08.0	10
12	1	I25.1	10
13	1	I65.2	10

### **3. Normalform (3NF)**

*Eine Relation befindet sich in dritter Normalform, wenn sie sich in zweiter Normalform befindet und zusätzlich kein Nichtschlüsselmerkmal transitiv („über Umwege“) funktional abhängig von irgendeinem Schlüssel ist.*

*Ein Mermal ist dann transitiv abhängig vom Primärschlüssel einer Relation, wenn es funktional abhängig von einem weiteren Schlüssel innerhalb der Relation ist, der selbst wiederum („direkt“) abhängig vom Primärschlüssel ist.*

Beispiel:

In der folgenden Tabelle werden die Fachabteilungen erfaßt, die die einzelnen Behandlungen eines Patienten durchführen; der Primärschlüssel dieser Relation wird von den Attributen PATNR und BEHNR gebildet.

<b>PATNR</b>	<b>BEHNR</b>	<b>FACHABTEILUNG</b>	<b>BEZEICHNUNG</b>
11	1	01	Herzchirurgie
12	1	01	Herzchirurgie
13	1	02	Gefäßchirurgie

Zwar ist das Attribut BEZEICHNUNG funktional abhängig vom Primärschlüssel der Relation (die Kombination von PATNR = 11 und BEHNR = 1 identifiziert die BEZEICHNUNG „Herzchirurgie“ eindeutig, d.h., diese Behandlung des Patienten findet in dieser Fachabteilung statt!), jedoch ist die BEZEICHNUNG zusätzlich funktional abhängig von einem weiteren Schlüssel, nämlich dem Attribut FACHABTEILUNG, welches seinerseits wiederum funktional abhängig vom Primärschlüssel PATNR | BEHNR ist.

Die Auflösung dieser transitiven Abhängigkeit und damit die Überführung dieser Relation in die dritte Normalform erfolgt durch Zerlegung:

<b>FACHABTEILUNG</b>	<b>BEZEICHNUNG</b>
01	Herzchirurgie
02	Gefäßchirurgie

<b>PATNR</b>	<b>BEHNR</b>	<b>FACHABTEILUNG</b>
11	1	01
12	1	01
13	1	02

## Anhang 7: JavaDoc der Klasse AgentDatabase (Auszug)

```
public class AgentDatabase extends java.lang.Object
```

Diese Instanz implementiert die Datenbank-bezogenen Funktionalitäten des Qualitätssicherungs-Assistenten im engeren Sinne.

**Author:**

abenning

Field Summary	
protected static java.lang.String	<b><u>className</u></b> Bezeichnung dieser Klasse (zur Verwendung in Meldungen).
protected static int	<b><u>ERRORCODE</u></b> Allgemeiner Fehlercode beispielsweise als Rückgabewert von Datenbank- Routinen.
static int	<b><u>FDST_ALL</u></b> Konstante, die die Herkunft des Inhaltes eines Kontext-bezogenen Feldes beschreibt: alle Inhalte (unabhängig von ihrer Herkunft).
static int	<b><u>FDST_DIAG</u></b> Konstante, die die Herkunft des Inhaltes eines Kontext-bezogenen Feldes beschreibt: Inhalt basiert auf einer "einfachen" Diagnose.
static int	<b><u>FDST_KOMP</u></b> Konstante, die die Herkunft des Inhaltes eines Kontext-bezogenen Feldes beschreibt: Inhalt basiert auf einem Komplex.
static int	<b><u>FDST_NON_CODE</u></b> Konstante, die die Herkunft des Inhaltes eines Kontext-bezogenen Feldes beschreibt: Inhalt ist nicht code-basiert (Diagnose/ICD, Prozedur/ OPS301).
static int	<b><u>FDST_PROC</u></b> Konstante, die die Herkunft des Inhaltes eines Kontext-bezogenen Feldes beschreibt: Inhalt basiert auf einer "einfachen" Prozedur.
static int	<b><u>HWT_ID_CHECK_RESULT</u></b> Konstante, die den Primärschlüssel eines Hinweistyps spezifiziert: (unbefriedigendes) Ergebnis einer Prüfung.
static int	<b><u>HWT_ID_FIELD_INFO</u></b> Konstante, die den Primärschlüssel eines Hinweistyps spezifiziert: Feld-Bezug einer Prüfung.
static int	<b><u>ICDL0K_ID_STANDARD</u></b> Konstante, die den Vorgabewert einer (nicht genauer



	definierten) ICD-Lokalisation spezifizierte (Primärschlüssel des entsprechenden Eintrages in der gleichnamigen Tabelle).
static int	<u><a href="#">IMPDAT_ID_HCH_FALL_ICD_DATEN</a></u> Konstante für den Primärschlüssel des Eintrages der Tabelle IMPORTDATEI, der die "HCH-Fall-ICD-Daten" spezifiziert.
static int	<u><a href="#">IMPDAT_ID_HCH_FALL_OPS_DATEN</a></u> Konstante für den Primärschlüssel des Eintrages der Tabelle IMPORTDATEI, der die "HCH-Fall-OPS-Daten" spezifiziert.
static int	<u><a href="#">IMPDAT_ID_HCH_FALLDATEN</a></u> Konstante für den Primärschlüssel des Eintrages der Tabelle IMPORTDATEI, der die "HCH-Falldaten" spezifiziert.
static int	<u><a href="#">IMPDAT_ID_HCH_OP_ICD_DATEN</a></u> Konstante für den Primärschlüssel des Eintrages der Tabelle IMPORTDATEI, der die "HCH-OP-ICD-Daten" spezifiziert.
static int	<u><a href="#">IMPDAT_ID_HCH_OP_OPS_DATEN</a></u> Konstante für den Primärschlüssel des Eintrages der Tabelle IMPORTDATEI, der die "HCH-OP-OPS-Daten" spezifiziert.
static int	<u><a href="#">IMPDAT_ID_HCH_OPDATEN</a></u> Konstante für den Primärschlüssel des Eintrages der Tabelle IMPORTDATEI, der die "HCH-OP-Daten" spezifiziert.
static int	<u><a href="#">K_ID_BQS_HCH_1</a></u> Konstante für den Kontext "BQS Modul HCH 2007 Teil 1 (präop.)".
static int	<u><a href="#">K_ID_BQS_HCH_2</a></u> Konstante für den Kontext "BQS Modul HCH 2007 Teil 2 (intraop.)".
static int	<u><a href="#">K_ID_BQS_HCH_3</a></u> Konstante für den Kontext "BQS Modul HCH 2007 Teil 3 (postop.)".
static int	<u><a href="#">K_ID_BQS_HCH_4</a></u> Konstante für den Kontext "BQS Modul HCH 2007 Nachbeobachtung/Follow-up".
static int	<u><a href="#">KIS_ID_ORBIS</a></u> Konstante für den Primärschlüssel in der Tabelle KIS, der ORBIS bezeichnet.
static int	<u><a href="#">KISTM_ID_ORBIS_STANDARD</a></u> Konstante für den KIS-transfermodus "ORBIS-Standard".
static int	<u><a href="#">OPSLOK_ID_STANDARD</a></u> Konstante, die den Vorgabewert einer (nicht genauer definierten) OPS-Lokalisation spezifiziert (Primärschlüssel des entsprechenden Eintrages in der gleichnamigen Tabelle).
static int	<u><a href="#">PRF_ID_CONJUNCTION_INTRAOP</a></u> Konstante, die eine durchzuführende Plausibilitätsprüfung

	spezifiziert: Prüfung auf obligate Koppelung von intraoperativen Elementen.
static int	<a href="#"><u>PRF_ID_CONJUNCTION_POSTOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf obligate Koppelung von postoperativen Elementen.
static int	<a href="#"><u>PRF_ID_CONJUNCTION_PRAEOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf obligate Koppelung von präoperativen Elementen.
static int	<a href="#"><u>PRF_ID_DISJUNCTION_INTRAOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf gegenseitigen Ausschluss von intraoperativen Elementen.
static int	<a href="#"><u>PRF_ID_DISJUNCTION_POSTOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf gegenseitigen Ausschluss von postoperativen Elementen.
static int	<a href="#"><u>PRF_ID_DISJUNCTION_PRAEOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf gegenseitigen Ausschluss von präoperativen Elementen.
static int	<a href="#"><u>PRF_ID_KCH_SCORE_INTRAOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung der KCH-Score relevanten Angaben.
static int	<a href="#"><u>PRF_ID_KCH_SCORE_PRAEOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung der KCH-Score relevanten Angaben.
static int	<a href="#"><u>PRF_ID_MISSING_MANDATORY_INTRAOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf Vollständigkeit der (codeunabhängigen) intraoperativen Pflichtangaben.
static int	<a href="#"><u>PRF_ID_MISSING_MANDATORY_POSTOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf Vollständigkeit der (codeunabhängigen) postoperativen Pflichtangaben.
static int	<a href="#"><u>PRF_ID_MISSING_MANDATORY_PRAEOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf Vollständigkeit der (codeunabhängigen) präoperativen Pflichtangaben.
static int	<a href="#"><u>PRF_ID_SUSP_INTRAOP</u></a> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf zweifelhafte intraoperative Angaben.
static int	<a href="#"><u>PRF_ID_SUSP_POSTOP</u></a>

	Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf zweifelhafte postoperative Angaben.
static int	<u><a href="#">PRF_ID_SUSP_PRAEOP</a></u> Konstante, die eine durchzuführende Plausibilitätsprüfung spezifiziert: Prüfung auf zweifelhafte präoperative Angaben.
static int	<u><a href="#">SK_ID_BQS_HCH_2007</a></u> Konstante für den Super-Kontext "BQS Modul Herzchirurgie 2007".
static int	<u><a href="#">SQLA_ID_FALL_ADMINISTRATIVE_DATEN</a></u> Konstante für die SQL-Anweisung, die bei Angabe der FALL_ID die administrativen Falldaten zurückliefert
static int	<u><a href="#">SQLA_ID_FALL_TITEL</a></u> Konstante für die SQL-Anweisung, die bei Angabe der FALL_ID einen Falltitel, wie er beispielsweise für Ausdrucke verwendet werden kann, zurückliefert
static int	<u><a href="#">SQLA_ID_ORBIS_HCH_FALL_ICD</a></u> Konstante für die auf der ORBIS-Datenbank auszuführende SQL-Anweisung, mit deren Hilfe die ICD-Daten eines Falles ermittelt werden können.
static int	<u><a href="#">SQLA_ID_ORBIS_HCH_FALL_OPS</a></u> Konstante für die auf der ORBIS-Datenbank auszuführende SQL-Anweisung, mit deren Hilfe die OPS-Daten eines Falles ermittelt werden können.
static int	<u><a href="#">SQLA_ID_ORBIS_HCH_FALLDATEN</a></u> Konstante für die SQL-Anweisung, mit deren Hilfe die HCH-Falldaten aus der ORBIS-Datenbank geladen werden (bezieht sich auf BQS-Modul HCH, Spezifikation 2007).
static int	<u><a href="#">SQLA_ID_ORBIS_HCH_IOP_QUELLE</a></u> Konstante für die auf der ORBIS-Datenbank auszuführende SQL-Anweisung, mit deren Hilfe für jeden Eingriff bestimmt werden kann, ob für den intraoperativen Teil des BQS-Moduls HCH (Spezifikation 2007) ein Teil der Daten dem HLM-Protokoll oder aber dem "intraoperativen Bogen" entnommen werden müssen; ihr Ergebnis entscheidet darüber, welche SQL-Anweisung(en) letztlich die HCH-OP-Daten liefert.
static int	<u><a href="#">SQLA_ID_ORBIS_HCH_OP_HLM</a></u> Konstante für die auf der ORBIS-Datenbank auszuführende SQL-Anweisung, mit deren Hilfe die HLM-Protokoll-basierten OP-Daten für das BQS-Modul HCH (Spezifikation 2007) ermittelt werden können.
static int	<u><a href="#">SQLA_ID_ORBIS_HCH_OP_ICD</a></u> Konstante für die auf der ORBIS-Datenbank auszuführende SQL-Anweisung, mit deren Hilfe die OP-bezogenen ICD-Daten eines Falles ermittelt werden können.
static int	<u><a href="#">SQLA_ID_ORBIS_HCH_OP_IOP</a></u>

	Konstante für die auf der ORBIS-Datenbank auszuführende SQL-Anweisung, mit deren Hilfe die IOP-Bogen-basierten OP-Daten für das BQS-Modul HCH (Spezifikation 2007) ermittelt werden können.
static int	<a href="#"><u>SQLA_ID_ORBIS_HCH_OP_OPS</u></a> Konstante für die auf der ORBIS-Datenbank auszuführende SQL-Anweisung, mit deren Hilfe die OP-bezogenen OPS-Daten eines Falles ermittelt werden können.
static int	<a href="#"><u>SQLA_ID_ORBIS_QSOM_WINDOW_CAPTION</u></a> Konstante für die SQL-Anweisung, mit deren Hilfe der Fenstertitel der ORBIS-QSOM-Zielapplikation bestimmt wird.
static int	<a href="#"><u>STS_ORBIS_QSOM_ENDLESS</u></a> Konstante für die zur Steuersequenz, mit deren Hilfe das ORBIS-QSOM- Formular in den "Endlosmodus" versetzt wird.
static int	<a href="#"><u>STS_ORBIS_QSOM_FUP_IOP2</u></a> Konstante für die Steuersequenz, mit deren Hilfe im ORBIS-QSOM-Formular der Wechsel zwischen dem Abschnitt "Nachbeobachtung/Follow-up" und einem ggf. vorhandenen 2. intraoperativen Abschnitt (=Folgeeingriff) ausgeführt wird.
static int	<a href="#"><u>STS_ORBIS_QSOM_INIT</u></a> Konstante für die zur Initialisierung des ORBIS-QSOM-Formulars notwendige Steuersequenz.
static int	<a href="#"><u>STS_ORBIS_QSOM_IOPX_IOPY</u></a> Konstante für die Steuersequenz, mit deren Hilfe im ORBIS-QSOM-Formular der Wechsel zwischen zwei auf Folgeeingriffe bezogenen intraoperativen Abschnitten (zwischen zweitem und drittem intraoperativen Abschnitt, zwischen drittem und viertem - und so weiter) ausgeführt wird.
static int	<a href="#"><u>STS_ORBIS_QSOM_NEW_FUP</u></a> Konstante für die Steuersequenz, mit deren Hilfe im ORBIS-QSOM-Formular das Follow-Up-Subformular angelegt wird.
static int	<a href="#"><u>STS_ORBIS_QSOM_NEW_OP</u></a> Konstante für die Steuersequenz, mit deren Hilfe im ORBIS-QSOM-Formular ein OP-Subformular neu angelegt wird.
static int	<a href="#"><u>TXT_ID_SOURCE_LEGEND</u></a> Konstante, die einen Primärschlüssel der Tabelle TEXT beschreibt: Legende für die Herkunft (Quelle) von Angaben.
static int	<a href="#"><u>VOVEPT_ID_IMPORT_ERROR</u></a> Konstant, die den Primärschlüssel des Typs eines Vorverarbeitungs- Protokolleintrages definiert: Import-Fehler.

## Constructor Summary

[AgentDatabase](#) ()

## Method Summary

static boolean	<p><a href="#"><u>containsNamedField</u></a>(java.sql.ResultSet resultSet, java.lang.String fieldName)</p> <p>Liefert true zurück, wenn die Datenmenge ein Feld mit der spezifizierten Bezeichnung (zwischen Groß- und Kleinschreibung wird nicht unterschieden) enthält, sonst false.</p>
static java.lang.Integer	<p><a href="#"><u>countBufferRecords</u></a>(java.sql.Connection connection, <a href="#"><u>Schema</u></a> schema)</p> <p>Liefert Anzahl der Fälle im "Puffer" der QSA-Datenbank zurück.</p>
static java.lang.Integer	<p><a href="#"><u>countRecordsToDo</u></a>(java.sql.Connection connection, <a href="#"><u>Schema</u></a> schema, java.lang.Integer kis_ID)</p> <p>Liefert die Anzahl der noch zu verarbeitenden Einträge der Tabelle VORVERARBEITUNG zurück; Hinweis: erfolgreich bearbeitete Datensätze werden aus der Tabelle gelöscht, während Einträge, bei deren Verarbeitung (Import-)Fehler auftraten, um Einträge in der Tabelle VORVERARBEITUNG_PROTOKOLL (und den entsprechenden weiteren Detail-Tabellen) ergänzt werden.</p>
static void	<p><a href="#"><u>createToDoProtocol</u></a>(<a href="#"><u>MessageProcessor</u></a> messageProcessor, java.sql.Connection connection, <a href="#"><u>Schema</u></a> schema, java.lang.Integer kis_ID, java.lang.String code, java.lang.Integer vovept_ID, java.lang.String text)</p> <p>Legt einen Protokoll-Eintrag des spezifizierten Typ zmm bezeichneten Eintrag der Tabelle VORVERARBEITUNG mit dem übergebenen Text an; CAVE: bestehende Protokolleinträge zu diesem Datensatz der Tabelle VORVERARBEITUNG werden dabei komplett gelöscht!</p>
static java.lang.String	<p><a href="#"><u>decode_KIS_WERT_WERT</u></a>(<a href="#"><u>MessageProcessor</u></a> messageProcessor, java.sql.Connection connection, <a href="#"><u>Schema</u></a> schema, java.lang.Integer kis_ID, java.lang.Integer impDat_ID, java.lang.Integer reihenfolge, java.lang.String originalValue)</p> <p>TODO: Übersetzung von direkt importierten Werten - KIS-spezifische Codes werden hier in die konkreten Ausprägungen von Werten umgewandelt, wie sie in der QSA-Datenbank benötigt werden; wurde für den jeweiligen aus dem KIS importierten Wert kein korrespondierender Wert definiert, wird der ursprüngliche, direkt aus dem KIS importierte Wert zurück geliefert (also das Orginal ohne Umwandlung); die entsprechenden Hinterlegungen finden sich in den Tabellen KIS_WERT_WERT, IMPORTDATEI_FELD, IMPORTDATEI_FELD_WERT und WERT_WERT.</p>

static void	<p><b><u>deleteFalldaten</u></b> (java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID)</p> <p>Löscht alle Einträge des über seinen Primärschlüssel identifizierten Falles aus den fallbezogenen Tabellen.</p>
static void	<p><b><u>deleteFalldaten</u></b> (java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String code)</p> <p>Löscht alle Einträge des über seinen KIS-spezifischen Code identifizierten Falles aus den fallbezogenen Tabellen.</p>
static void	<p><b><u>deleteToDoRecord</u></b> (java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String code)</p> <p>Ermittelt den zugehörigen Primärschlüssel des durch KIS und KIS-spezifischen Fall-Code definierten Eintrages der Tabelle VORVERARBEITUNG; wird kein entsprechender Datensatz gefunden, wird null zurückgeliefert.</p>
static java.lang.Integer	<p><b><u>get_FALL_KONTEXT_PRUEFUNG_HINWEIS_RF</u></b> (MessageProcessor messageProcessor, java.sql.Connection connection, <u>Schema</u> schema, int fkp_ID, java.lang.Integer hwt_ID)</p> <p>Ermittelt für einen neuen Eintrag in der Tabelle FALL_KONTEXT_PRUEFUNG_HINWEIS mit Hilfe der übergebenen Parameter den "nächsten" Wert für das Feld REIHENFOLGE; existieren noch keine vergleichbaren Einträge (in Bezug auf Werte für die Felder FKP_ID und HWT_ID), wird standardmäßig 1 zurückgeliefert.</p>
static <u>AdmissionData</u>	<p><b><u>getAdmissionData</u></b> (java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID)</p> <p>Liefert die administrativen Daten des spezifizierten Falles zurück; hierzu wird die in der Datenbank hinterlegte SQL-Abfrage verwendet, die durch den Primärschlüssel SQLA_ID_FALL_ADMINISTRATIVE_DATEN spezifiziert wird; werden keine entsprechenden Daten gefunden, wird null zurückgeliefert.</p>
static java.lang.Integer	<p><b><u>getBuffered_Fall_ID</u></b> (java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String fallCodeKIS)</p> <p>Liefert die zugehörige FALL_ID zurück, wenn sich die Daten des durch den Primärschlüssel des KIS (in der gleichnamigen Tabelle) und seinen KIS-spezifischen Code (in ORBIS: "Fallnummer") identifizierten Falles bereits in den Puffer-Tabellen befinden, sonst null.</p>
static java.lang.Integer	<p><b><u>getCorrespondingID</u></b> (java.sql.Connection connection, <u>Schema</u> schema, java.lang.String tableName, java.lang.String pkFieldName, java.lang.Integer kis_ID,</p>

	<p>java.lang.String code)</p> <p>Liefert den Wert des benannten Primärschlüssel-Feldes der bezeichneten Tabelle (dabei muss es sich um die korrespondierende Tabelle mit den hinterlegten KIS-spezifischen Codes handeln) zurück, der der übergebenen Kombination aus ID des KIS sowie dem KIS-spezifischen Code des Eintrages entspricht; null, wenn kein entsprechender Wert ermittelt werden konnte.</p>
static java.lang.Integer	<p><a href="#"><b>getCorrespondingID</b></a>(java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.String tableName, java.lang.String pkFieldName, java.lang.Integer kis_ID, java.lang.String code, java.lang.Integer standardResult)</p> <p>Liefert den Wert des benannten Primärschlüssel-Feldes der bezeichneten Tabelle (dabei muss es sich um die korrespondierende Tabelle mit den hinterlegten KIS-spezifischen Codes handeln) zurück, der der übergebenen Kombination aus ID des KIS sowie dem KIS-spezifischen Code des Eintrages entspricht; standardResult, wenn kein entsprechender Wert ermittelt werden konnte.</p>
static java.lang.String	<p><a href="#"><b>getDeleteStatementFall</b></a>(<a href="#">Schema</a> schema, java.lang.String tableName, java.lang.Integer fall_ID)</p> <p>Liefert die auf den durch seine ID definierten Fall bezogene Delete- Anweisung für die spezifizierte Tabelle zurück.</p>
static java.lang.String	<p><a href="#"><b>getDeleteStatementFallOp</b></a>(<a href="#">Schema</a> schema, java.lang.String tableName, java.lang.Integer fall_ID)</p> <p>Liefert die auf den durch seine ID definierten Fall bezogene Delete- Anweisung für die spezifizierte Tabelle zurück.</p>
static java.lang.String	<p><a href="#"><b>getDeleteStatementPufFallMap</b></a>(<a href="#">Schema</a> schema, java.lang.String tableName, java.lang.Integer fall_ID)</p> <p>Liefert die auf den durch seine ID definierten Fall bezogene Delete- Anweisung für die spezifizierte Tabelle zurück.</p>
static java.lang.String	<p><a href="#"><b>getDeleteStatementPufFallOpMap</b></a>(<a href="#">Schema</a> schema, java.lang.String tableName, java.lang.Integer fall_ID)</p> <p>Liefert die auf den durch seine ID definierten Eingriff bezogene Delete- Anweisung für die spezifizierte Tabelle zurück.</p>
static <a href="#">InputSetList</a>	<p><a href="#"><b>getEnterInputSetList</b></a>(java.sql.Connection conne</p>

	<pre> ction, Schema schema, java.lang.Integer kistm_ID, java.lang.String id_PFM, java.lang.String id_PFOM, java.lang.String k_ID, java.lang.String kd_ID, java.lang.String dw_ID, java.lang.String dwk_ID, java.lang.String diagnose_Wert, java.lang.String kp_ID, java.lang.String pw_ID, java.lang.String pwk_ID, java.lang.String prozedur_Wert, java.lang.String kkp_ID, java.lang.String kpw_ID, java.lang.String komplex_Wert, java.lang.String kw_ID, java.lang.String wert_Wert, java.lang.String mainComment) </pre> <p>Liefert eine Instanz vom Typ InputSetList zurück, die die Ausfüll- Steuersequenzen (Eintragen der Feld-Inhalte im engeren Sinne) des für das angegebene Diagnose-, Prozedur, Komplex- oder codeunabhängige Feld des (implizit durch kd_ID, kp_ID, kkp_id oder kw_id) bezeichneten Kontext enthält; der jeweils einzutragende Wert wird durch die zur Ausprägung passende Ausfüllsequenz, die durch die übergebenen Primärschlüssel bestimmt wird, definiert; kann keine Ausfüllsequenz ermittelt werden, wird der Wert selbst in eine Steuer-Zeichenfolge umgewandelt und "direkt" eingetragen.</p>
static java.lang.Integer	<pre> getFall_ID (java.sql.Connection connection, Schema schema, java.lang.Integer kis_ID, java.lang.String code) </pre> <p>Liefert den passenden Wert des Feldes FALL_ID der Tabelle FALL (Primärschlüssel) zur übergebenen Kombination aus KIS-spezifischem Code und ID des KIS zurück; null, falls kein entsprechender Eintrag existiert.</p>
static java.lang.Integer	<pre> getFallOp_ID (java.sql.Connection connection, Schema schema, java.lang.Integer kis_ID, java.lang.String fall_Code, java.lang.String op_Code) </pre> <p>Liefert den Primärschlüssel der durch die Angabe des KIS, des KIS-spezifischen Codes des zugehörigen Falles sowie ihren KIS-spezifischen Code spezifizierten Operation in der Tabelle FALL_OP zurück.</p>
static java.lang.Integer	<pre> getFallOp_ID (java.sql.Connection connection, Schema schema, java.lang.Integer kis_ID, java.lang.String fall_Code, java.lang.String op_Code, java.sql.Timestamp beginn) </pre>



	<p>Liefert den Primärschlüssel der durch die Angabe des KIS, des KIS-spezifischen Codes des zugehörigen Falles und ihren Code innerhalb des KIS ODER ihren Beginn spezifizierten Operation in der Tabelle FALL_OP zurück.</p>
static java.lang.Integer	<p><b><u>getFallOp_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String code, java.sql.Timestamp beginn)</p> <p>Liefert den Primärschlüssel der durch die Angabe des KIS, des KIS-spezifischen Codes des zugehörigen Falles sowie ihren Beginn spezifizierten Operation in der Tabelle FALL_OP zurück.</p>
static StringList	<p><b><u>getFormatTemplates</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.Integer impDat_ID)</p> <p>Liefert eine Instanz vom Typ StringList mit den Format-Templates für die Felder der spezifizierten Import-Datei zurück.</p>
static java.lang.Integer	<p><b><u>getICD_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer icdVer_ID, java.lang.String code)</p> <p>Liefert den Primärschlüssel (Feld ICD_ID der Tabelle ICD) des über seinen Code sowie seine ICD-Version spezifizierten ICD- Katalogeintrages zurück.</p>
static java.lang.String	<p><b><u>getICD_Version_Bezeichnung</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer icdVer_ID)</p> <p>Liefert das Label der spezifizierten ICD-Version zurück (null, falls dieses nicht ermittelt werden konnte).</p>
static java.lang.Integer	<p><b><u>getICDVer_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID)</p> <p>Liefert den Primärschlüssel der für den angegebenen Fall anzuwendenden ICD-Version (Feld ICDVER_ID der Tabelle ICD_VERSION) zurück; die gesuchte ICD-Version ist dabei die zum Zeitpunkt der Aufnahme des Falles in das Krankenhaus gültige; kann der Wert nicht ermittelt werden, wird null zurückgeliefert.</p>
static java.lang.Integer	<p><b><u>getICDVer_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String code)</p> <p>Liefert den Primärschlüssel der für den angegebenen Fall anzuwendenden ICD-Version (Feld ICDVER_ID der Tabelle ICD_VERSION) zurück; die gesuchte ICD-Version ist dabei die zum Zeitpunkt der Aufnahme des Falles in das Krankenhaus gültige; kann der Wert nicht ermittelt werden, wird null zurückgeliefert; der Fall wird über die Angabe des KIS sowie seinen KIS-spezifischen Code ("Fallnummer" oder ähnliches) identifiziert.</p>

<pre>static StringList</pre>	<pre><b><u>getImportFileFieldNames</u></b>(java.sql.Connection connection,                              <a href="#">Schema</a> schema,                              java.lang.Integer impDat_ID)     Liefert eine StringList-Instanz mit den Bezeichnungen der Felder der spezifizierten Importdatei zurück, wobei diese unter Beachtung ihrer Reihenfolge gelistet werden.</pre>
<pre>static <a href="#">InputSetList</a></pre>	<pre><b><u>getInputSetList</u></b>(java.sql.Connection connection,                    <a href="#">Schema</a> schema,                    java.lang.Integer sts_ID)     Einzelschritte der über den angegebenen Primärschlüssel (STS_ID) spezifizierten Steuersequenz aus der gleichnamigen Tabelle sowie der Tabelle STEUERSEQUENZ_SCHRITT auslesen und als Instanz vom Typ InputSetList zurückliefern (diese enthält keine Einträge, wenn eine Steuersequenz mit diesem Primärschlüssel nicht existiert).</pre>
<pre>static <a href="#">InputSetList</a></pre>	<pre><b><u>getInputSetList</u></b>(java.sql.Connection connection,                    <a href="#">Schema</a> schema,                    java.lang.Integer fall_ID,                    java.lang.Integer fallOp_ID,                    java.lang.Integer[] k_ID,                    java.lang.Integer kistm_ID)     Liefert eine Instanz vom Typ InputSetList zurück, die die Steuersequenzen für die Übertragung der QS-Daten des spezifizierten Falles - oder, falls fallOp_ID ungleich null ist: der spezifizierten Operation - innerhalb der angegebenen Kontexte (es kann eine Menge von IDs in Form eines Arrays übergeben werden, die in eine entsprechende Passage des Where-Abschnitts umgesetzt wird) enthält; diese umfassen die Ziel- und gegebenenfalls die Post-Navigation sowie die eigentlichen Ausfüll-Anweisungen; es reicht aus, entweder fall_ID oder aber fallOp_ID anzugeben, da über fallOp_ID auch eine Identifikation des zugehörigen Falles möglich ist.</pre>
<pre>static <a href="#">InputSetList</a></pre>	<pre><b><u>getInputSetList</u></b>(java.sql.Connection connection,                    <a href="#">Schema</a> schema,                    java.lang.Integer fall_ID,                    java.lang.Integer fallOp_ID,                    java.lang.Integer k_ID,                    java.lang.Integer kistm_ID)     Liefert eine Instanz vom Typ InputSetList zurück, die die Steuersequenzen für die Übertragung der QS-Daten des spezifizierten Falles innerhalb des angegebenen Kontext enthält; diese umfassen die Ziel- und gegebenenfalls die Post-Navigation sowie die eigentlichen Ausfüll-Anweisungen.</pre>
<pre>static java.lang.String</pre>	<pre><b><u>getInText</u></b>(java.lang.String fieldName,              java.lang.Integer[] elements,              boolean literal)     Liefert einen String nach dem Schema "Feldname In (ID1, ID2, IDn)" zurück, wobei hinsichtlich des übergebenen Arrays mit den Identifikatoren zwischen numerischen (literal = false) und literalen Elementen (literal = true) unterschieden wird (letztere werden innerhalb der Klammer um</pre>

	einschliessende Hochkommata ergänzt).
static java.lang.String	<b><u>getInText</u></b> (java.lang.String fieldName, java.lang.String[] elements, boolean literal) Liefert einen String nach dem Schema "Feldname In (ID1, ID2, IDn)" zurück, wobei hinsichtlich des übergebenen Arrays mit den Identifikatoren zwischen numerischen (literal = false) und literalen Elementen (literal = true) unterschieden wird (letztere werden innerhalb der Klammer um einschliessende Hochkommata ergänzt).
static java.lang.String	<b><u>getKontext_Bezeichnung</u></b> (java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer k_ID) Liefert Bezeichnung des spezifizierten Kontext zurück.
static <a href="#">KontextFieldDataList</a>	<b><u>getKontextFieldDataList</u></b> (java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer fall_ID, java.lang.Integer fallOp_ID, java.lang.Integer k_ID, int fieldDataSourceType) Liefert einen Vektor mit Instanzen vom Typ KontextFieldData zurück, die die Kontext-bezogenen Felder eines Falles oder einer Operation eines Falles (abhängig davon, ob fall_ID oder fallOp_ID angegeben wurde) beschreiben; die Herkunft der Felder kann über den Parameter fieldDataSourceType spezifiziert werden (siehe hierzu die Kommentare zu den Konstanten, die der Notation FDST_X entsprechen).
static <a href="#">InputSetList</a>	<b><u>getNavigationInputSetList</u></b> (java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer kistm_ID, java.lang.Integer navtyp_ID, java.lang.String kd_ID, java.lang.String kp_ID, java.lang.String kkp_ID, java.lang.String kw_ID, java.lang.String mainComment) Liefert eine Instanz vom Typ InputSetList zurück, die die Navigations- Steuersequenzen des spezifizierten Navigationstyps (beispielsweise Ziel-Navigation) für das angegebene Diagnose-, Prozedur, Komplex- oder codeunabhängige Feld des (implizit durch kd_ID, kp_ID, kkp_id oder kw_id) bezeichneten Kontext enthält.
static java.lang.String	<b><u>getNextCodeToDo</u></b> (java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer kis_ID) Liefert den KIS-spezifischen Fall-Code des als nächstes zu verarbeitenden Eintrages der Tabelle VORVERARBEITUNG für das spezifizierte KIS zurück; Hinweis: erfolgreich (also ohne Importfehler oder ähnliches) verarbeitete Einträge werden aus

	<p>der Tabelle (und deren Detail-Tabellen) gelöscht, während für mit Fehlern verarbeitete Datensätze anschließend ein zugehöriger Datensatz in der Tabelle VORVERARBEITUNG_PROTOKOLL existiert - zu verarbeiten sind also als dem spezifizierten KIS zugeordneten Datensätze, für die noch kein Protokoll existiert (in der durch die Felder REIHENFOLGE und VOVE_ID determinierten Reihenfolge; siehe hierzu auch Methode countRecordToDo).</p>
static java.lang.Integer	<p><b><u>getOPS_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer opsVer_ID, java.lang.String code)  Liefert den Primärschlüssel (Feld OPS_ID der Tabelle OPS301) des über seinen Code sowie seine OPS301-Version spezifizierten OPS301- Katalogeintrages zurück.</p>
static java.lang.String	<p><b><u>getOPS_Version_Bezeichnung</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer opsVer_ID)  Liefert das Label der spezifizierten OPS301-Version zurück (null, falls dieses nicht ermittelt werden konnte).</p>
static java.lang.Integer	<p><b><u>getOPSVer_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID)  Liefert den Primärschlüssel der für den angegebenen Fall anzuwendenden OPS301-Version (Feld OPSVER_ID der Tabelle OPS301_VERSION) zurück; die gesuchte OPS301-Version ist dabei die zum Zeitpunkt der Aufnahme des Falles in das Krankenhaus gültige; kann der Wert nicht ermittelt werden, wird null zurückgeliefert.</p>
static java.lang.Integer	<p><b><u>getOPSVer_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String code)  Liefert den Primärschlüssel der für den angegebenen Fall anzuwendenden OPS301-Version (Feld OPSVER_ID der Tabelle OPS301_VERSION) zurück; die gesuchte OPS301-Version ist dabei die zum Zeitpunkt der Aufnahme des Falles in das Krankenhaus gültige; kann der Wert nicht ermittelt werden, wird null zurückgeliefert.</p>
static java.lang.String	<p><b><u>getPruefung_Bezeichnung</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer prf_ID)  Liefert die Bezeichnung der spezifizierten Prüfung zurück.</p>
static <u>RuleOffenceDataList</u>	<p><b><u>getRuleOffenceDataList</u></b>(<u>MessageProcessor</u> messageProcessor, java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID, java.lang.Integer sk_ID, java.lang.Integer prf_ID)  Liefert einen Vektor mit Instanzen vom Typ</p>

	<p>RuleOffenceData zurück, die die Superkontext-bezogenen, nicht-befriedigenden Ergebnisse der Plausibilitätsprüfung für den spezifizierten Fall (siehe hierzu die Kommentare zu den Konstanten, die der Notation PRF_ID_X entsprechen); existieren keine nicht-befriedigenden Ergebnisse, enthält der Vektor keine Ergebnisse; ist prf_ID null, wird der gesamte Prüfbericht in den Vektor übertragen (also die unbefriedigenden Ergebnisse aller kontextbezogenen Prüfungen für den Fall).</p>
<pre>static <a href="#">RuleOffenceDataList</a></pre>	<pre><a href="#">getRuleOffenceDataList</a>(<a href="#">MessageProcessor</a> messageProcessor,     java.sql.Connection connection,     <a href="#">Schema</a> schema,     java.lang.Integer fall_ID,     java.lang.Integer sk_ID,     java.lang.Integer[] prf_ID)</pre> <p>Liefert einen Vektor mit Instanzen vom Typ RuleOffenceData zurück, die die Superkontext-bezogenen, nicht-befriedigenden Ergebnisse der Plausibilitätsprüfung für den spezifizierten Fall (siehe hierzu die Kommentare zu den Konstanten, die der Notation PRF_ID_X entsprechen); existieren keine nicht-befriedigenden Ergebnisse, enthält der Vektor keine Ergebnisse; es kann eine einzelne oder aber eine Menge von Prüfungen über den Parameter prf_ID angegeben werden; enthält das Array keine Elemente, wird der gesamte Prüfbericht in den Vektor übertragen (also die unbefriedigenden Ergebnisse aller kontextbezogenen Prüfungen für den Fall).</p>
<pre>protected static <a href="#">RuleOffenceDataList</a></pre>	<pre><a href="#">getRuleOffenceDataList</a>(<a href="#">MessageProcessor</a> messageProcessor,     java.sql.Connection connection,     <a href="#">Schema</a> schema,     java.lang.Integer fall_ID,     java.lang.Integer sk_ID,     java.lang.String whereText)</pre> <p>Liefert einen Vektor mit Instanzen vom Typ RuleOffenceData zurück, die die Superkontext-bezogenen, nicht-befriedigenden Ergebnisse der Plausibilitätsprüfung für den spezifizierten Fall; existieren keine nicht-befriedigenden Ergebnisse, enthält der Vektor keine Ergebnisse; ob und wie weit eine Einschränkung auf bestimmte Prüfungen erfolgt, wird durch eine entsprechende Formulierung in whereText definiert.</p>
<pre>static java.lang.String</pre>	<pre><a href="#">getSQLStatement</a>(java.sql.Connection connection,     <a href="#">Schema</a> schema,     int sqlA_ID)</pre> <p>Liefert den Text der über ihren Primärschlüssel in der Tabelle SQL_ANWEISUNG (Feld SQLA_ID) spezifizierten SQL-Anweisung zurück.</p>
<pre>static <a href="#">TitleTextList</a></pre>	<pre><a href="#">getStatementTitleTextList_Pruefung_Element</a>(<a href="#">Schema</a> schema,     java.lang.Integer fall_ID,     java.lang.Integer k_ID,     java.lang.Integer prf_ID)</pre>

	<p>Liefert eine TitleTextList-Instanz zurück, die die SQL-Anweisungen enthält, deren sequentielle Ausführung die für die spezifizierte Kombination von Fall, Kontext und Prüfung eine Ergebnismenge zurückliefert, die ihrerseits die Daten liefert, die in die Tabellen FALL_KONTEXT_PRUEFUNG und FALL_KONTEXT_PRUEFUNG_HINWEIS eingetragen werden; die Anweisungen beziehen sich ausschließlich auf Einzel-Elemente, die einer Prüfung unterzogen werden sollen; diese Routine ist für den Einsatz innerhalb der Methode processRuleOffenceCheck bestimmt.</p>
static java.lang.String	<p><b><u>getSuperkontext_Bezeichnung</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer sk_ID)</p> <p>Liefert Bezeichnung des spezifizierten Super-Kontext zurück.</p>
static java.lang.String	<p><b><u>getText</u></b>(java.sql.Connection connection, <u>Schema</u> schema, int txt_ID)</p> <p>Liefert den über seinen Primärschlüssel in der Tabelle TEXT (Feld TXT_ID) spezifizierten Text zurück.</p>
static java.lang.String	<p><b><u>getToDoComment</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String code)</p> <p>Ermittelt die Bemerkung zum über den Primärschlüssel des KIS sowie den KIS-spezifischen Fall-Code definierten Eintrag (null, falls kein Eintrag existiert oder das Feld keinen Wert enthält).</p>
static java.lang.Integer	<p><b><u>getVOVE_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kis_ID, java.lang.String code)</p> <p>Ermittelt den zugehörigen Primärschlüssel des durch KIS und KIS- spezifischen Fall-Code definierten Eintrages des Tabelle VORVERARBEITUNG; wird kein entsprechender Datensatz gefunden, wird null zurückgeliefert.</p>
static java.lang.Integer	<p><b><u>getW_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer impDat_ID, java.lang.Integer reihenfolge)</p> <p>Liefert die ID des Wertes zurück (Primärschlüssel der Tabelle WERT, W_ID), der dem über seine Position bezeichneten Feld der spezifizierten Importdatei zugeordnet ist; null, wenn kein passender Eintrag gefunden wird.</p>
static void	<p><b><u>insert_FALL_ICD</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID, java.lang.Integer icd_ID, java.lang.Integer icdKat_ID, java.lang.Integer icdLok_ID, java.sql.Timestamp datum, java.lang.String bemerkung)</p>

	<p>Legt einen neuen Eintrag in der Tabelle FALL_ICD mit den übergebenen Werten an (der Primärschlüssel wird innerhalb der Methode ermittelt).</p>
static int	<p><b><u>insert_FALL_KON_PRUF_FALL_KON_PRUF_HINWEIS</u></b> (<a href="#">MessageProcessor</a> messageProcessor,  <a href="#">java.sql.Connection</a> connection, <a href="#">Schema</a> schema,  int fkp_ID, <a href="#">java.lang.Integer</a> fall_ID,  <a href="#">java.lang.Integer</a> kprf_ID,  <a href="#">java.lang.Integer</a> hwt_ID,  <a href="#">java.lang.String</a> hinweis)</p> <p>Legt einen neuen Eintrag mit den übergebenen Werten in der Detail- Tabelle FALL_KONTEXT_PRUEFUNG_HINWEIS an; existiert noch kein zugehöriger Eintrag in der Master-Tabelle FALL_KONTEXT_PRUEFUNG (zur aktuell betrachteten kontextbezogenen Prüfung), so wird dieser zunächst (mit den ebenfalls übergebenen Werten) angelegt und der dabei implizit ermittelte Primärschlüssel für die Einfügeoperation auf FALL_KONTEXT_PRUEFUNG_HINWEIS verwendet; liefert den (gegebenenfalls neu ermittelten) Primärschlüssel des zugehörigen Eintrages in FALL_KONTEXT_PRUEFUNG zurück (FKP_ID).</p>
static int	<p><b><u>insert_FALL_KONTEXT_PRUEFUNG_HINWEIS</u></b> (<a href="#">java.sql.Connection</a> connection, <a href="#">Schema</a> schema,  <a href="#">java.lang.Integer</a> fkp_ID,  <a href="#">java.lang.Integer</a> hwt_ID,  <a href="#">java.lang.Integer</a> reihenfolge,  <a href="#">java.lang.String</a> hinweis)</p> <p>Neuen Eintrag in der Tabelle FALL_KONTEXT_PRUEFUNG_HINWEIS mit den übergebenen Werten anlegen; der Primärschlüssel wird dabei implizit ermittelt und als Ergebnis zurückgeliefert.</p>
static int	<p><b><u>insert_FALL_KONTEXT_PRUEFUNG</u></b> (<a href="#">java.sql.Connection</a> connection, <a href="#">Schema</a> schema,  <a href="#">java.lang.Integer</a> fall_ID,  <a href="#">java.lang.Integer</a> kprf_ID)</p> <p>Neuen Eintrag in der Tabelle FALL_KONTEXT_PRUEFUNG mit den übergebenen Werten anlegen; der Primärschlüssel FP_ID wird dabei ebenso wie der Inhalt des Feldes ZULETZT_AKTUALISIERT implizit bestimmt; liefert den Primärschlüssel des neu angelegten Datensatzes zurück.</p>
static void	<p><b><u>insert_FALL_OP_ICD</u></b> (<a href="#">java.sql.Connection</a> connection, <a href="#">Schema</a> schema,  <a href="#">java.lang.Integer</a> fallOp_ID,  <a href="#">java.lang.Integer</a> icd_ID,  <a href="#">java.lang.Integer</a> icdKat_ID,  <a href="#">java.lang.Integer</a> icdLok_ID)</p>

	<p>Legt einen neuen Eintrag in der Tabelle FALL_OP_ICD mit den übergebenen Werten an (der Primärschlüssel wird innerhalb der Methode ermittelt).</p>
static void	<pre><u>insert_FALL_OP_OPS301</u>(java.sql.Connection connection,                         <u>Schema</u> schema,                         java.lang.Integer fallOp_ID,                         java.lang.Integer ops_ID,                         java.lang.Integer opsKat_ID,                         java.lang.Integer opsLok_ID)</pre> <p>Legt einen neuen Eintrag in der Tabelle FALL_OP_OPS301 mit den übergebenen Werten an (der Primärschlüssel wird innerhalb der Methode ermittelt).</p>
static void	<pre><u>insert_FALL_OP_WERT</u>(java.sql.Connection connection,                         <u>Schema</u> schema,                         java.lang.Integer fallOp_ID,                         java.lang.Integer impDat_ID,                         java.lang.Integer reihenfolge,                         java.lang.String wert)</pre> <p>Legt einen Eintrag in der Tabelle FALL_WERT an, wobei ein Teil der Feld-Inhalte mit Hilfe der übergebenen Parameter in vorgeschalteten Schritten zunächst ermittelt werden müssen (und zwar aus den Angaben zur Importdatei und der Position des Feldes innerhalb Importdatei).</p>
static java.lang.Integer	<pre><u>insert_FALL_OP</u>(java.sql.Connection connection,                 <u>Schema</u> schema,                 java.lang.Integer fall_ID,                 java.lang.String code,                 java.sql.Timestamp beginn,                 java.sql.Timestamp ende)</pre> <p>Legt einen neuen Eintrag in der Tabelle FALL_OP mit den übergebenen Werte an und liefert den Primärschlüssel des Datensatzes (der innerhalb der Methode ermittelte Wert des Feldes FALLOP_ID) zurück.</p>
static void	<pre><u>insert_FALL_OPS301</u>(java.sql.Connection connection,                     <u>Schema</u> schema,                     java.lang.Integer fall_ID,                     java.lang.Integer ops_ID,                     java.lang.Integer opsKat_ID,                     java.lang.Integer opsLok_ID,                     java.sql.Timestamp datum,                     java.lang.String bemerkung)</pre> <p>Legt einen neuen Eintrag in der Tabelle FALL_OPS301 mit den übergebenen Werten an (der Primärschlüssel wird innerhalb der Methode ermittelt).</p>
static void	<pre><u>insert_FALL_WERT</u>(java.sql.Connection connection,                   <u>Schema</u> schema,                   java.lang.Integer fall_ID,                   java.lang.Integer impDat_ID,                   java.lang.Integer reihenfolge,                   java.lang.String wert)</pre> <p>Legt einen Eintrag in der Tabelle FALL_WERT an, wobei</p>



	<p>ein Teil der Feld-Inhalte mit Hilfe der übergebenen Parameter in vorgeschalteten Schritten zunächst ermittelt werden müssen (und zwar aus den Angaben zur Importdatei und der Position des Feldes innerhalb Importdatei).</p>
<pre>static java.lang.Integer</pre>	<p><b><u>insert_FALL</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.String code, java.lang.String nachname, java.lang.String vorname, java.sql.Timestamp geburtsdatum, java.sql.Timestamp aufnahme_FA, java.sql.Timestamp abgang_FA, java.sql.Timestamp aufnahme_KH, java.sql.Timestamp abgang_KH, java.lang.Integer g_ID, java.lang.Integer ba_ID, java.lang.Integer kis_ID, java.lang.Integer fa_ID)</p> <p>Legt einen neuen Eintrag in der Tabelle FALL mit den übergebenen Werten an und liefert dessen (erst innerhalb dieser Methode ermittelten!)</p>
<pre>static void</pre>	<p><b><u>insert_PUFFER_FALL_MAPPING</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer id, java.lang.Integer reihenfolge, java.lang.String kc_ID_KeyPress, java.lang.String kc_ID_KeyRelease, java.lang.String bc_ID_MousePress, java.lang.String bc_ID_MouseRelease, java.lang.String mouseMove_X, java.lang.String mouseMove_Y, java.lang.String delay, java.lang.String typeText)</p> <p>Legt einen neuen Datensatz in der Tabelle PUFFER_FALL_MAPPING mit den übergebenen Parametern an; der Primärschlüssel wird innerhalb der Methode bestimmt.</p>
<pre>static boolean</pre>	<p><b><u>isEnterableElement</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer kistm_ID, java.lang.String k_ID, java.lang.String fall_ID, java.lang.String fallOp_ID, java.lang.String d_ID, java.lang.String dw_ID, java.lang.String dwk_ID, java.lang.String p_ID, java.lang.String pw_ID, java.lang.String pwk_ID, java.lang.String kp_ID, java.lang.String kpw_ID, java.lang.String w_ID, java.lang.String wert_Wert)</p>

	<p>Prüfung, ob Feld überhaupt ausgefüllt werden muss beziehungsweise kann; dies ist dann NICHT der Fall, wenn der durch die übergebenen Parameter bezeichnete zugehörige Inhalt (Diagnose-Kombinationswert, Diagnose-Wert, Diagnose-Normalwert et cetera) als zweites Element einer Beziehung zwischen 2 Elementen definiert ist, deren zweites vom Vorhandensein des ersten abhängig ist, wobei das erste Element NICHT in der Menge der Elemente des angegebenen Falles vorkommt; beim Vorliegen der beschriebenen Konstellation wird false zurückgeliefert, in allen anderen Fällen true; ist der Parameter fallOp_ID null, soll die sich die Abfrage hinsichtlich des ersten Elements auf die Tabelle PUFEEER_FALL_KONTEXT erstrecken, anderenfalls auf PUFEEER_FALL_OP_KONTEXT.</p>
static boolean	<p><b><u>isMandatory</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer sk_ID, java.lang.Integer impDat_ID, java.lang.Integer reihenfolge)</p> <p>Liefert true zurück, falls der korrespondierende Wert des über seine "REIHENFOLGE" bezeichneten Feldes der spezifizierten Importdatei innerhalb mindestens eines Kontextes des angegebenen Superkontext eine obligate Angabe darstellt ("Pflichtfeld"), sonst false.</p>
static boolean	<p><b><u>isRuleOffenceChecked</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID)</p> <p>Liefert true zurück, falls für den spezifizierten Fall bereits Einträge (und damit ein Prüfprotokoll) in der Tabelle FALL_KONTEXT_PRUEFUNG existieren, anderenfalls false.</p>
static boolean	<p><b><u>isValidFall_ID</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID)</p> <p>Liefert true zurück, wenn ein Eintrag in der Tabelle FALL mit dem angegebenen Wert von FALL_ID existiert, sonst false.</p>
static void	<p><b><u>markAsBuffered</u></b>(java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer fall_ID)</p> <p>Führt eine Update-Operation auf das Feld EJN_ID_PUFFER_BEREIT der Tabelle FALL durch, wodurch vermerkt wird, dass die Puffer (nunmehr) gefüllt sind.</p>
static void	<p><b><u>presetEmptyImportFields</u></b>(<u>MessageProcessor</u> messageProcessor, java.sql.Connection connection, <u>Schema</u> schema, java.lang.Integer impDat_ID, StringList fields)</p> <p>Prüft die aus dem spezifizierten Importdatensatz (via JDBC oder Dateiimport importierte Daten) stammende Feldliste und trägt in etwaige "leere" Felder jeweils den für diese in der Tabelle IMPORT_DATEI_FELD_WERT definierten Standardwert ein, so weit ein solcher angegeben wurde.</p>

<pre>static void</pre>	<pre><a href="#">processKontextSpecificItems_1_HCH_PraeOp</a> (<a href="#">MessageProcessor</a> messageProcessor, java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer fall_ID, java.lang.Integer kistm_ID)</pre> <p>Kontext-Spezialitäten für den präoperativen Teil des BQS-Moduls HCH implementieren - hier müssen die Werte der Felder 24 ("akute Infektionen") sowie 31 ("Lungenerkrankung(en)") ermittelt werden (CAVE: gegebenenfalls muss hier später der Bezug auf einen KIS-Transfermodus - durch Auswertung des Parameters kistm_ID - "nachgerüstet" werden).</p>
<pre>static void</pre>	<pre><a href="#">processKontextSpecificItems_2_HCH_IntraOp</a> (<a href="#">MessageProcessor</a> messageProcessor, java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer fall_ID, java.lang.Integer kistm_ID)</pre> <p>Kontext-Spezialitäten für den intraoperativen Teil des BQS-Moduls HCH implementieren - hier muss der Wert des Feldes 70 ("Klappeneingriff") ermittelt werden - und zwar für alle Operationen eines Falles, weswegen lediglich der Parameter fall_ID übergeben wird (CAVE: gegebenenfalls muss hier später der Bezug auf einen KIS-Transfermodus - durch Auswertung des Parameters kistm_ID - "nachgerüstet" werden).</p>
<pre>static void</pre>	<pre><a href="#">processKontextSpecificItems_3_HCH_PostOp</a> (<a href="#">MessageProcessor</a> messageProcessor, java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer fall_ID, java.lang.Integer kistm_ID)</pre> <p>Kontext-Spezialitäten für den postoperativen Teil des BQS-Moduls HCH implementieren - hier muss der Wert des Feldes 93 ("Patient trägt Schrittmacher/Defibrillator") ermittelt werden (CAVE: gegebenenfalls muss hier später der Bezug auf einen KIS-Transfermodus - durch Auswertung des Parameters kistm_ID - "nachgerüstet" werden).</p>
<pre>static void</pre>	<pre><a href="#">processKontextSpecificItems</a> (<a href="#">MessageProcessor</a> messageProcessor, java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer[] k_ID, java.lang.Integer fall_ID, java.lang.Integer kistm_ID)</pre> <p>Kontext-Spezialitäten für spezifische Felder eines Falles implementieren: hier werden "zusammengesetzte" Feld-Inhalte erzeugt, die sich nach speziellen Algorithmen sowohl aus codierten Diagnosen oder Prozeduren wie auch aus codeunabhängigen Werten bestimmen lassen, bei denen eine direkte Bestimmung lediglich auf Grundlage der Informationen einer der beiden Datengruppen jedoch nicht möglich ist; hier</p>

	kann eine Menge von Kontexten über ihre Primärschlüssel definiert werden.
static void	<p><b><u>processKontextSpecificItems</u></b> (<a href="#">MessageProcessor</a> messageProcessor, java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer k_ID, java.lang.Integer fall_ID, java.lang.Integer kistm_ID)</p> <p>Kontext-Spezialitäten für spezifische Felder eines Falles implementieren: hier werden "zusammengesetzte" Feld-Inhalte erzeugt, die sich nach speziellen Algorithmen sowohl aus codierten Diagnosen oder Prozeduren wie auch aus codeunabhängigen Werten bestimmen lassen, bei denen eine direkte Bestimmung lediglich auf Grundlage der Informationen einer der beiden Datengruppen jedoch nicht möglich ist.</p>
static void	<p><b><u>processMessage</u></b> (<a href="#">MessageProcessor</a> messageProcessor, java.lang.String message)</p> <p>Gibt die übergeben Botschaft mit Hilfe der ebenfalls übergebenen MessageProcessor-Instanz aus, sofern diese initialisiert wurde.</p>
static void	<p><b><u>processRuleOffenceCheck</u></b> (<a href="#">MessageProcessor</a> messageProcessor, java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer fall_ID, java.lang.Integer sk_ID)</p> <p>Plausibilitätsprüfung durchführen und (nicht-befriedigende) Ergebnisse in den Tabellen FALL_KONTEXT_PRUEFUNG beziehungsweise FALL_KONTEXT_PRUEFUNG_HINWEIS protokollieren; gegebenenfalls dort vorhandene "alte" Einträge zum spezifizierten Fall werden zuvor gelöscht.</p>
static java.lang.String	<p><b><u>processSQLSelectStringStatement_FALL_ID</u></b> (java.sql.Connection connection, <a href="#">Schema</a> schema, int sqlA_ID, java.lang.Integer fall_ID)</p> <p>Präpariert das das via sqlA_ID bezeichnete mit der übergebenen FALL_ID, führt es aus und liefert den Ergebnis-String zurück.</p>
static void	<p><b><u>processSuperKontextSpecificItems</u></b> (<a href="#">MessageProcessor</a> messageProcessor, java.sql.Connection connection, <a href="#">Schema</a> schema, java.lang.Integer sk_ID, java.lang.Integer fall_ID, java.lang.Integer kistm_ID)</p> <p>Super-Kontext-Spezialitäten für spezifische Felder eines Falles implementieren: hier werden "zusammengesetzte" Feld-Inhalte erzeugt, die sich nach speziellen Algorithmen sowohl aus codierten Diagnosen oder Prozeduren wie auch aus codeunabhängigen Werten bestimmen lassen, bei denen eine direkte Bestimmung lediglich auf Grundlage der Informationen</p>

	einer der beiden Datengruppen jedoch nicht möglich ist; es werden sequenziell alle betroffenen Felder der einem Superkontext zugeordneten Kontexte "behandelt".
static int	<u><a href="#">transfer_FALL_KONTEXT_DIAGNOSE_OP</a></u> (java.sql.Connection connection, <u><a href="#">Schema</a></u> schema, java.lang.Integer fall_ID) Übertragung der abgeleiteten, Diagnose- und OP-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.
static int	<u><a href="#">transfer_FALL_KONTEXT_DIAGNOSE</a></u> (java.sql.Connection connection, <u><a href="#">Schema</a></u> schema, java.lang.Integer fall_ID) Übertragung der abgeleiteten, Diagnose- und Fall-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.
static int	<u><a href="#">transfer_FALL_KONTEXT_KOMPLEX_OP</a></u> (java.sql.Connection connection, <u><a href="#">Schema</a></u> schema, java.lang.Integer fall_ID) Übertragung der abgeleiteten, Komplex- und OP-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.
static int	<u><a href="#">transfer_FALL_KONTEXT_KOMPLEX</a></u> (java.sql.Connection connection, <u><a href="#">Schema</a></u> schema, java.lang.Integer fall_ID) Übertragung der abgeleiteten, Komplex- und Fall-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.
static int	<u><a href="#">transfer_FALL_KONTEXT_PROZEDUR_OP</a></u> (java.sql.Connection connection, <u><a href="#">Schema</a></u> schema, java.lang.Integer fall_ID) Übertragung der abgeleiteten, Prozedur- und OP-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.
static int	<u><a href="#">transfer_FALL_KONTEXT_PROZEDUR</a></u> (java.sql.Connection connection, <u><a href="#">Schema</a></u> schema, java.lang.Integer fall_ID) Übertragung der abgeleiteten, Prozedur- und Fall-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.
static int	<u><a href="#">transfer_FALL_KONTEXT_WERT_OP</a></u> (java.sql.Connection connection, <u><a href="#">Schema</a></u> schema,

	<pre>java.lang.Integer fall_ID)</pre> <p>Übertragung der direkt importierten, OP-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.</p>
<pre>static int</pre>	<pre><b><u>transfer_FALL_KONTEXT_WERT</u></b>(java.sql.Connection connection,                                <u>Schema</u> schema, java.lang.Integer fall_ID)</pre> <p>Übertragung der direkt importierten, Fall-bezogenen QS-Daten in die entsprechende Puffer-Tabelle; liefert die Anzahl der in der Puffer-Tabelle angelegten Einträge zurück.</p>

## **Danksagung**

Besonders möchte ich mich bei Herrn Professor Dr. med. Arno Krian, Chefarzt der Klinik für Thorax- und Kardiovaskularchirurgie am Herzzentrum Duisburg, Evangelisches und Johanniter Klinikum Niederrhein gGmbH, für die freundliche Überlassung des Themas und die stets großzügige Unterstützung bedanken.





## Lebenslauf

Name: Arnd Benninghoff

Wohnort: Mauerstr. 1-3  
47137 Duisburg

Geburtsdatum: 07.10.1966

Geburtsort: Wesel

Eltern: Bernhard und Käthe Benninghoff, geborene Platt

Familienstand: verheiratet mit Petra – Susanne Benninghoff (geb. Dengler) seit dem 31.10.1997

Staatsangehörigkeit: Deutsch

Konfession: Evangelisch

Schulbildung 1985 Abitur

Wehrdienst 1985 - 1988 Soldat auf Zeit in der Teilstreitkraft Luftwaffe, Laufbahn als Offiziersanwärter im Truppendienst, Entlassung im September 1988 als Leutnant der Reserve

Studium 1988 - 1994 Humanmedizinisches Studium an der Heinrich - Heine - Universität zu Düsseldorf

1994 - 1995 Praktisches Jahr im Evangelischen und Johanniter Klinikum Niederrhein

August 1990 Ärztliche Vorprüfung

August 1991 1. Abschnitt der ärztlichen Prüfung

April 1994 2. Abschnitt der ärztlichen Prüfung

Mai 1995 3. Abschnitt der ärztlichen Prüfung

Beruflicher Werdegang Juli 1995 – heute Tätigkeit als Arzt im Praktikum bzw. als Arzt in der Klinik für Thorax- und Kardiovaskularchirurgie am Herzzentrum Duisburg, Evangelisches und Johanniter Klinikum Niederrhein gGmbH

Februar 1997 Approbation als Arzt

April 2003 Zuerkennung der Zusatzbezeichnung "Medizinische Informatik" durch die Ärztekammer Nordrhein



Abstract der Dissertation von Arnd Benninghoff

**Eine softwarebasierte Methodik zur Wahrung der Konsistenz von DRG- und qualitätssicherungs-relevanten Daten am Beispiel des BQS-Moduls „Herzchirurgie“**

Zum gegenwärtigen Zeitpunkt sind die für die medizinische Qualitätssicherung und die für die Abrechnung nach dem pauschalierten Entgeltsystem (G-DRG) benötigten Informationen selbst innerhalb integrierter Krankenhausinformationssysteme strukturell und funktionell weitgehend voneinander getrennt.

Eine exemplarische Betrachtung des Moduls „Herzchirurgie“ der Bundesgeschäftsstelle Qualitätssicherung zeigt, dass sich aus den mit Hilfe der amtlichen Kataloge (ICD, OPS) verschlüsselten Diagnosen und Leistungen, die die Basis für die Bestimmung der abzurechnenden Pauschale bilden, zumindest ein Teil der für die Qualitätssicherung benötigten Angaben direkt ableiten lässt, während sich ein weiterer Teil der Elemente beider Datenmengen für eine gegenseitige Kontrolle auf Konsistenz eignet.

Die Arbeit beschreibt ein Datenmodell sowie die darauf operierenden Auswertungen und Software-Routinen, mit deren Hilfe beide Varianten einer integrativen Sichtweise – die Gewinnung von für die Qualitätssicherung relevanten Aussagen aus ICD- und OPS-Codes ebenso wie die wechselseitige Prüfung auf Plausibilität – realisiert werden können. Während der Entwicklung der Strukturen wurde - trotz der Orientierung am Beispiel des Moduls „Herzchirurgie“ - deren Design so flexibel bzw. generell gültigen Prinzipien gehorchend gestaltet, dass eine Anpassung an die Maßnahmen anderer Fächer ebenso wie an die verschiedenen am Markt erhältlichen Krankenhausinformationssysteme mit geringem Aufwand und weitgehend ohne programmiersprachliche Kenntnisse möglich ist. Dies gelingt nicht zuletzt durch die Verwendung standardisierter Sprachen (SQL, Java) und teilweise frei verfügbarer Software-Werkzeuge.

Zunächst werden alle Felder des Datensatzes „Herzchirurgie“ im Hinblick auf eine Eignung zur direkten Ableitung aus den Schlüsseln der Kataloge oder alternativ zur Sicherung der Korrektheit analysiert.

Daran schließt sich die Formulierung der an das Datenmodell zu adressierenden Forderungen an, dessen Bestandteile schrittweise – geleitet von diesen Kriterien - entwickelt und dargestellt werden.

Es folgt die Vorstellung der Datenbank-Abfragen, mit deren Hilfe die Extraktion konkreter Feldinhalte des Moduls Herzchirurgie aus der Menge via ICD- bzw. OPS-codierter Diagnosen und Leistungen wie auch die auf die Entdeckung möglicher Fehler während der Dokumentation zielende Untersuchung gelingen. Letztere setzt die Definition von Prüfregeln voraus, deren Eigenschaften und Implementierung in den dafür vorgesehenen Bereichen des Modells in diesem Kontext erläutert werden.

Schließlich wird die praktische Anwendung des Datenmodells innerhalb einer im Bereich der Klinik im Einsatz befindlichen Anwendung vorgestellt. Erste Erfahrungen mit dieser Lösung legen nahe, dass primäre Ziele wie die Reduktion des mit der Datenerfassung verbundenen Aufwandes ebenso wie die Verbesserung der Qualität der dokumentierten Angaben selbst erreicht werden können.

Im Anhang finden sich unter anderem das zur Implementierung des Datenmodells notwendige SQL-Skript sowie die Quelltexte aller erforderlichen und im Text referenzierten Auswertungen.

Herzzentrum Duisburg  
Klinik für Thorax- u. Kardiovaskularchirurgie  
Chefarzt Prof. Dr. med. A. Krian  
Gerrickstraße 21, 47137 Duisburg

*Arnd Benninghoff, 18. 8. 08*  
