

# **Technische Dokumentation zur BQS-Spezifikation für QS-Dokumentationssoftware 12.0**

**zur Datenerfassung, Plausibilitätsprüfung und  
Datenübermittlung im Rahmen der externen  
vergleichenden Qualitätssicherung**

Stand: 01.07.2008 Version: 12.0, gültig ab 01.01.2009

BQS Bundesgeschäftsstelle Qualitätssicherung gGmbH

Kanzlerstr. 4

40472 Düsseldorf

Telefon 0211/28 07 29-0

Fax 0211/28 07 29-99

E-mail [spezifikation@bqs-online.de](mailto:spezifikation@bqs-online.de)

Homepages:

<http://www.bqs-online.de>

<http://www.bqs-qualitaetsreport.de>

<http://www.bqs-outcome.de>

<http://www.bqs-qualitaetsindikatoren.de>

Die BQS Bundesgeschäftsstelle Qualitätssicherung gGmbH übernimmt keinerlei Haftung für Fehler und Irrtümer, die in dieser Spezifikation enthalten sind.

# Inhaltsverzeichnis

<b>1</b>	<b>EINLEITUNG .....</b>	<b>7</b>
1.1	NEUESTE NACHRICHTEN	7
1.2	ZIELSETZUNG DER SPEZIFIKATION	7
1.2.1	Bereitstellung valider und vergleichbarer Daten	8
1.2.2	Förderung der Prozess- und Datenintegration im Krankenhaus	8
1.3	ZIELSETZUNG DER TECHNISCHEN DOKUMENTATION	8
1.4	HINWEISE FÜR DEN BENUTZER	8
1.5	ALLGEMEINE ANMERKUNGEN ZUR STRUKTUR DER SPEZIFIKATION	9
1.5.1	Abfragen der Datenbank	10
1.5.2	Tabellenstruktur der Datenbank	12
<b>2</b>	<b>DATENFELDBESCHREIBUNG.....</b>	<b>15</b>
2.1	AUFBAU DER DATENFELDBESCHREIBUNG	16
2.1.1	Ziele	16
2.1.2	Module (Datensätze)	16
2.1.3	Teildatensätze	18
2.1.4	Datenfelder (Bogenfelder)	24
2.1.5	Felder - ein erster Schritt zur Prozess- und Datenintegration	26
2.1.6	Basistypen	28
2.1.7	Schlüssel	29
2.1.8	Schlüsselwerte	31
2.1.9	Exklusionsschlüssel	34
2.2	ÜBERSCHRIFTEN	34
2.3	AUSFÜLLHINWEISE	35
2.4	VERWENDUNG DER DATENFELDBESCHREIBUNG FÜR DIE GESTALTUNG VON EINGABEMASKEN	37
<b>3</b>	<b>PLAUSIBILITÄTSPRÜFUNGEN .....</b>	<b>40</b>

3.1	ARTEN DER PLAUSIBILITÄTSPRÜFUNGEN	40
3.1.1	Harte Prüfungen	41
3.1.2	Weiche Plausibilitätsprüfungen	41
3.1.3	Warnende Plausibilitätsprüfungen	42
3.2	FELDBEZOGENE PRÜFUNGEN	42
3.2.1	Überprüfung des Formats	43
3.2.2	Überprüfung der Feldlänge	44
3.2.3	Überprüfung der Schlüsselkodes	44
3.2.4	Überprüfung numerischer Wertebereiche	45
3.2.5	Überprüfung der Muss-Felder	47
3.3	FELDÜBERGREIFENDE REGELN	49
3.3.1	Die Regeltabelle	49
3.3.2	Bogenfelder einer Regel	50
3.3.3	Mehrfach vorkommende Regeln	50
3.3.4	Weitere Regeln	51
3.4	REGELSYNTAX	51
3.5	TEILDATENSATZÜBERGREIFENDE REGELN	58
3.5.1	Klassifizierung der teildatensatzübergreifenden Regeln	58
3.5.2	Regeln mit Teildatensatz-Listefeldern	59
3.6	VERFAHREN FÜR DIE EVALUATION VON REGELN	60
3.7	FELDGRUPPEN	62
3.7.1	Struktur der Tabellen <code>FeldGruppe</code> und <code>FeldgruppeFelder</code>	63
3.7.2	Syntax der Feldgruppenregeln	64
3.7.3	Formale Definition von Feldgruppen	65
3.7.4	Feldgruppen mit mehreren Filterfeldern	68
3.7.5	Gestaltung von Eingabemasken mit Feldgruppen	68
3.8	LEITLINIEN FÜR DIE GESTALTUNG DER BENUTZEROBERFLÄCHEN VON ERFASSUNGSPROGRAMMEN	71
<b>4</b>	<b>LISTEN VON SCHLÜSSELKODES (OPS, ICD-10-GM) .....</b>	<b>73</b>
4.1	OPS-LISTEN	73
4.2	ICD-LISTEN	75

<b>5</b>	<b>VERSIONIERUNG .....</b>	<b>76</b>
5.1	GRUNDLEGENDE DEFINITIONEN	76
5.1.1	Finale Spezifikation - Zwischenstände	76
5.1.2	Historie der Versionen	77
5.2	DELTA-INFORMATIONEN ZUR VORHERGEHENDEN FINALEN VERSION	77
5.2.1	Neue Entitäten	77
5.2.2	Geänderte Entitäten	78
5.2.3	Gelöschte Entitäten	79
5.2.4	Konfiguration der Delta-Berechnung	80
5.3	ABGRENZUNG ZWISCHEN ERFASSUNGSJAHREN UND DATENSATZFORMATEN	80
5.4	VERSION DES EXPORTVERFAHRENS	81
5.5	VERSION DER EXPORTDATEIEN	81
<b>6</b>	<b>DATENEXPORT .....</b>	<b>82</b>
6.1	REGISTRIERUNG EINES DOKUMENTATIONSSYSTEMS	83
6.1.1	Registrierung bei einer Landesgeschäftsstelle Qualitätssicherung	83
6.1.2	Registrierung bei der BQS Bundesgeschäftsstelle Qualitätssicherung	83
6.2	IDENTIFIZIERUNG VON DATENSÄTZEN	84
6.3	DER EXPORTVORGANG	86
6.3.1	Die Steuerdatei	86
6.3.2	Erzeugung der Transaktionsdatei	89
6.4	EXPORT VON TEILDATENSÄTZEN	89
6.4.1	Anonymisierung	89
6.4.2	Aufbau der Exportdatei	92
6.5	REGELN FÜR DIE ENTGEGENNAHME VON DATENSÄTZEN UND TEILDATENSÄTZEN	96
6.5.1	Prüfungen und Datenrückbestätigung	96
6.5.2	Prüfung der warnenden Plausibilitätsregeln	98
6.5.3	Stornierung von Datensätzen	99
6.6	DIE ANTWORTDATEIEN	99
6.6.1	Die Bestätigungsdatei	100
6.6.2	Die Fehlerdatei	101

6.7 SONSTIGER DATEITRANSFER ZWISCHEN KRANKENHAUS, LQS UND BQS	105
---	-----

<b>ANHANG .....</b>	<b>106</b>
---------------------	------------

A HINWEISE ZUR IMPLEMENTIERUNG VON FUNKTIONEN	106
---	-----

B ERLÄUTERUNGEN ZUM DATENÜBERMITTLUNGSVERFAHREN PER E-MAIL	108
--	-----

C BESONDERHEITEN DER QUALITÄTSSICHERUNG IN HESSEN	109
---	-----

# 1 Einleitung

## 1.1 *Neueste Nachrichten*

Die Datensätze der neuen Spezifikation sind im Vergleich zum Vorjahr weitgehend stabil.

Bei den Datenfeldern ergeben sich insgesamt folgende Änderungen:

- 8 neue Datenfelder
- 17 gelöschte Datenfelder
- 12 aktualisierte Datenfelder

Betroffen sind die Module 10/2, 16/1, 17/1, 18/1, 20/1, 21/3, PNEU und die transplantiationsmedizinischen Module. Plausibilitätsregeln (inkl. Fehlertexte), Feldgruppen und Ausfüllhinweise, welche mit den veränderten Datenfeldern im Zusammenhang stehen, wurden überarbeitet.

Im Jahre 2009 ist das Modul 20/1 nur noch in Hamburg verpflichtend. Die BQS hat im Auftrag der [EQS](#) (Externe Qualitätssicherung Hamburg) Änderungswünsche umgesetzt. Das Modul 20/1 wird im Jahre 2009 als landespezifisches Modul „*Perkutane transluminale Angioplastie (PTA Hamburg)*“ fortgeführt.

Sämtliche Änderungen der Datenbank (im Vergleich zum Service Release 3 der BQS-Spezifikation 11.0) lassen sich in den Delta-Tabellen nachvollziehen.

Die technische Struktur der Spezifikation wurde nicht verändert. Jedoch wurde in der technische Dokumentation die Beispiele aktualisiert überarbeitet und die Anpassungen an das Jahr 2009 vorgenommen.

## 1.2 *Zielsetzung der Spezifikation*

Mit der vorliegenden Spezifikation für die Erstellung von Software zur Erfassung, Plausibilitätsprüfung und Übermittlung von Daten für die externe vergleichende Qualitätssicherung werden grundsätzlich zwei Ziele verfolgt:

### **1.2.1 Bereitstellung valider und vergleichbarer Daten**

Einerseits stellen externe Qualitätssicherungsmaßnahmen, die einen Vergleich der Qualität von Krankenhausleistungen zum Ziel haben, eine Reihe von methodischen Anforderungen an die Datenerhebung, Datenerfassung und Plausibilitätsprüfung, um valide, reliable und auch vergleichbare Daten gewinnen zu können. Die Erfassung und Plausibilitätsprüfung durch unterschiedliche Programme beinhaltet grundsätzlich die Gefahr einer Verzerrung der Daten. Die Vorgaben dieser Spezifikation sollen dazu dienen, durch einheitliche Festlegung von Datenfeldbeschreibungen, Plausibilitätsregeln, Grundsätzen der Benutzerschnittstellengestaltung und Datenübermittlungsformate dieser Gefahr entgegenzuwirken.

### **1.2.2 Förderung der Prozess- und Datenintegration im Krankenhaus**

Andererseits sollen die Krankenhäuser dabei unterstützt werden, die Mehrfacherfassung und redundante Datenhaltung für Qualitätssicherung und medizinische Dokumentation im Krankenhaus zu vermeiden und damit den Dokumentationsmehraufwand so weit wie möglich zu reduzieren. Dies kann am besten durch eine Integration der Datenerfassung und Datenverwaltung in die bestehenden medizinischen Dokumentationssysteme realisiert werden. Die Spezifikation soll bestehende medizinische Dokumentations- und Krankenhausinformationssysteme um eine Komponente für die Erfassung, Plausibilitätsprüfung und Übermittlung von Daten für die externe vergleichende Qualitätssicherung (evQS) ergänzen.

### **1.3 Zielsetzung der technischen Dokumentation**

Da die Spezifikation als Datenbank zur Verfügung gestellt wird, ist es das Hauptziel dieses Dokumentes, die Struktur der Datenbank zu erläutern und Hilfestellung bei der softwaretechnischen Umsetzung der Spezifikation zu geben.

### **1.4 Hinweise für den Benutzer**

Diese Dokumentation richtet sich an Hersteller und Entwickler von Krankenhaussoftware, interessierte Krankenhäuser, Softwareentwickler in den Landesgeschäftsstellen Qualitätssicherung (LQS) und im zentralen Datenverarbeitungsservice der BQS (ZDS) und an alle anderen Interessenten.

Hersteller von Krankenhaussoftware entwickeln Programme, welche die Erfassung aller qualitätsrelevanten Daten dieser Spezifikation ermöglichen, diese Daten auf Vollständigkeit und Plausibilität prüfen, im vorgegebenen Format an die datenentgegennehmenden Stellen (Datenstellen, Landesgeschäftsstellen Qualitätssicherung oder BQS) exportieren und feh-



lerhafte (von den Datenstellen abgelehnte Datensätze) in geeigneter Weise dem Anwender zur Korrektur und erneutem Export vorlegen. Sowohl die Landesgeschäftsstellen für Qualitätssicherung als auch die Hersteller von Krankenhaussoftware müssen die Daten auf Plausibilität und Vollständigkeit prüfen. Darüber hinaus bietet die Spezifikation den Softwareanbietern Hinweise für die Gestaltung der Benutzeroberfläche und die benutzerfreundliche Plausibilitätsprüfung unmittelbar bei der Dateneingabe.

Die Datenstellen nehmen die Qualitätsdaten von den Krankenhäusern entgegen und prüfen sie auf Vollständigkeit und Plausibilität. Nach bestandener Prüfung werden Datensätze von den Datenstellen als angenommen bestätigt. Fehlerhafte Datensätze werden in einem Fehlerprotokoll angezeigt. Folgende Übersicht listet die Bedeutung einzelner Themen dieses Dokumentes für die beiden Zielgruppen auf.

Tabelle 1: Bedeutung verschiedener Themen für die Zielgruppen der Spezifikation

Thema	relevant für Datenentgegen- nahme (LQS/BQS)	relevant für Softwareher- steller
Bogenfelder	ja	ja
Felder	weniger	ja
Basistypen	ja	ja
Regeln	ja	ja
Mehrfachregeln	ja	weniger
Syntax	ja	ja
RegelFelder	ja	ja
Feldgruppen	nein	ja
Exportformat	ja	ja

## 1.5 Allgemeine Anmerkungen zur Struktur der Spezifikation

Die Spezifikation ist in einer relationalen Datenbank abgelegt. Zurzeit wird sie ausschließlich als Access-Datenbank (MS Access 2000) zur Verfügung gestellt.

Der Name der Spezifikation richtet sich nach folgendem Schema:

spez-<Version>.mdb

<Version> bezeichnet die Versionsnummer (z.B. 110 für 11.0, 120 für 12.0).

Beispiel:

Im Erfassungsjahr 2009 ist die Spezifikation `spez-120.mdb` gültig.

Service-Releases zur Spezifikation sind am Namen erkennbar:

`spez-<Version>_SR<n>.mdb`

`<n>` bezeichnet die laufende Nummer (1, 2, 3) des Service-Releases.

Beispiel:

`spez-120_SR1.mdb` ist das erste Service-Release zur Spezifikation 12.0.

Eine Übersicht über die in der jeweiligen Version enthaltenen Module (Leistungsbereiche) liefert die Abfrage *Datensätze* der Access-Datenbank (siehe Kapitel 1.5.1).

### 1.5.1 Abfragen der Datenbank

Die Abfragen der Access-Datenbank geben einen vereinfachenden Überblick über die Spezifikation.

- *Datensätze:*  
Überblick über die in der Spezifikation enthaltenen Module (verpflichtende und freiwillige Module).
- *Datenfeldbeschreibung*  
Alle Bogenfelder der spezifizierten Module, sortiert nach Modulname, Bogenname und Zeilennummer der Bogenfelder (Kapitel 2.1.4).
- *DatenfeldbeschreibungFürEinModul*  
Wenn man diese Abfrage aufruft, so muss der Modulname (z.B. "12/1") angegeben werden und man erhält eine entsprechende modulbezogene Auswahl der Datenfeldbeschreibung.
- *Plausibilitätsregeln*  
Alle Plausibilitätsregeln der spezifizierten Module, sortiert nach Modulname und Nummer der Regel (Kapitel 3)
- *PlausibilitätsregelnFürEinModul*  
Wenn man diese Abfrage aufruft, so muss der Modulname (z.B. "12/1") angegeben werden und man erhält eine entsprechende modulbezogene Auswahl der Plausibilitätsregeln.

- *Teildatensätze*  
Überblick über die Teildatensätze und die Regeln für das Anlegen von Teildatensätzen (Kapitel 2.1.3)
- *Ersatzfelder*  
Auflistung der zu anonymisierenden Bogenfelder für alle spezifizierten Module (Kapitel 6.4)
- *OPSListen*  
Überblick über die Codes der OPS-Listen (Kapitel 4).
- *ICDListen*  
Überblick über die Codes der ICD-Listen (Kapitel 4).
- *ExportfelderFürEinModul*  
Überblick über die Exportfelder eines Moduls (Modulname ist explizit anzugeben). Man erhält eine Übersicht über die zu exportierenden Felder incl. Zuordnung zum Teildatensatz. (Kapitel 6.4)
- *Feldgruppen*  
Übersicht über alle Feldgruppen (Kapitel 3.7).
- *FeldgruppenFürEinModul*  
Wenn man diese Abfrage aufruft, so muss der Modulname (z.B. "12/1") angegeben werden und man erhält eine entsprechende modulbezogene Auswahl der Feldgruppen eines Moduls.
- *WertebereicheNumerischerFelder*  
Modulübergreifende Anzeige der numerischen Datenfelder (Typ ZAHLE und GANZEZAHLE) und ihrer Wertebereiche.
- *WertebereicheNumerischerFelderFürEinModul*  
Anzeige der numerischen Datenfelder (Typ ZAHLE und GANZEZAHLE) und ihrer Wertebereiche für ein Modul, welches direkt angegeben werden muss.
- *ÜberschriftenFürEinModul*  
Anzeige der Überschriften für das angegebene Modul. Angegeben werden Start- und Ende-Felder der Überschriften, sowie die Ebene der Überschriften.
- *Schlüsselcodes*  
Zeigt alle Schlüssel und die zugehörigen Schlüsselwerte an.
- *Ausfüllhinweise*  
Anzeige der Zuordnung von Ausfüllhinweisen (htm.Dateien) zu den Feldern in den einzelnen Modulen.
- *AusfüllhinweiseFürEinModul*

Anzeige der Zuordnung von Ausfüllhinweisen (htm.Dateien) zu den Feldern eines Modules, welches direkt angegeben werden muss.

## 1.5.2 Tabellenstruktur der Datenbank

Die Tabellen und Spalten (Attribut) unterliegen einem einheitlichen Namensschema. Erlaubte Zeichen sind die Buchstaben a-z, A-Z und die Ziffern 0-9. Umlaute und Sonderzeichen werden nicht verwendet. Das erste Zeichen eines Namens darf keine Ziffer sein.

Ein Tabellename beginnt immer mit einem Großbuchstaben und ein Attributname mit einem Kleinbuchstaben. Wenn ein Name aus mehreren Teilen (z.B. Substantiven) besteht, so beginnt jeder nachfolgende Namensteil mit einem Großbuchstaben.

### Beispiel:

```
BasisTyp (Tabelle)
idBasisTyp (Spalte)
```

Für jede Tabelle ist in der Spezifikation *höchstens ein Primärschlüssel* definiert, der nach folgendem Schema aufgebaut ist:

```
id<TabellenName>
```

Der Ausdruck in den eckigen Klammern ist ein Platzhalter für den Namen der Tabelle.

Die meisten Tabellen haben einen einfachen Primärschlüssel vom Typ `AUTOINCREMENT`. Zusätzlich enthalten derartige Tabellen mindestens ein *identifizierendes Attribut*<sup>1</sup>, welches durch Setzen eines weiteren, eindeutigen Index (bestehend aus einem oder mehreren Attributen) definiert ist.

### Beispiele:

- Identifizierendes Attribut: `Attribut name in Tabelle BasisTyp`
- Identifizierende Attributkombination: `Attribute code und fkSchluessel in Tabelle SchlüsselWert`

---

<sup>1</sup> oder eine identifizierende Attributkombination, die einen eindeutigen Schlüssel definiert

Es gibt auch Tabellen, deren einziger eindeutiger Schlüssel der Primärschlüssel ist. Ein Beispiel ist die Tabelle `MussKann` mit dem Primärschlüssel `idMussKann` vom Typ `TEXT(1)` (entspricht `VARCHAR(1)`). Diese Tabellen sind als einfache "Nachschlagtabellen" zu interpretieren. Im Fall der Tabelle `MussKann` soll im entsprechenden Fremdschlüsselfeld der verknüpften Detailtabelle durch das Datenbankschema gewährleistet werden, dass nur ein 'M' oder 'K' eingegeben werden darf.

Die Namen von Fremdschlüsseln sind analog zum Namen der Primärschlüssel aufgebaut:

```
fk<FremdTabellenName>
```

Die Namensgebung von Primär- und Fremdschlüsseln vereinfacht den Aufbau von komplexeren Abfragen, welche sich über mehrere Tabellen erstrecken (Inklusionsverknüpfungen, *Joins*).

Die Fremdschlüsselattribute (Namen beginnen mit `fk`) wurden in MS Access als Datenbankattribute zum Nachschlagen eingerichtet. Z.B. wird beim Fremdschlüsselattribut `fkModul` in der Tabelle `Bogen` nicht mehr der Primärschlüssel des jeweiligen Moduls, sondern der Name des Moduls angezeigt.<sup>2</sup> Sind zwei Tabellen mehrfach durch Schlüssel-Fremdschlüssel-Beziehungen miteinander verknüpft, so kann der Name eines Fremdschlüssels auch folgendermaßen aufgebaut sein:

```
fk<FremdTabellenName><Rolle>
```

`<Rolle>` ist der Platzhalter für eine zusätzliche Qualifizierung der Relation.

N-M-Beziehungen werden wie üblich über Verknüpfungstabellen realisiert. In der Spezifikation haben Verknüpfungstabellen gewöhnlich keinen Primärschlüssel<sup>3</sup>, jedoch einen eindeutigen Schlüssel, der über die Fremdschlüsselfelder definiert ist. Ein Beispiel hierfür ist die Tabelle `RegelFelder`, welche die Tabellen `BogenFeld` und `Regel` verknüpft.

Folgende Attribute treten in vielen Tabellen auf und seien hier kurz erläutert:

- `name` ist in der Regel als "technischer Name" zu verstehen. Z.B. wird `Feld.name` als Variablenname in den Plausibilitätsregeln verwendet.
- `bezeichnung` ist eine kurze Beschreibung. Z.B. ist `BogenFeld.bezeichnung` der Text, welcher ein Feld auf einem Eingabeformular beschreibt.

---

<sup>2</sup> Diese Änderung betrifft nur die Anzeige, nicht jedoch die Struktur der Datenbank!

<sup>3</sup> Hier: Primärschlüssel im Sinne der Access-Definition eines Primärschlüssels. Streng genommen wird über die beiden Fremdschlüssel ein neuer Primärschlüssel definiert.

- `bedingung` enthält einen logischen Ausdruck. Prominentester Vertreter dieses Attributtyps ist das Attribut `bedingung` in der Tabelle `Regeln`.

## 2 Datenfeldbeschreibung

Für alle Dokumentationsbögen<sup>4</sup> eines Moduls existiert jeweils eine eigene Datenfeldbeschreibung. Die Datenfeldbeschreibung spezifiziert alle auszufüllenden Datenfelder (Bogenfelder, auch Items genannt). Die Datenfeldbeschreibung besteht aus mehreren Tabellen (Abbildung 1), welche in den nachfolgenden Abschnitten erläutert werden.

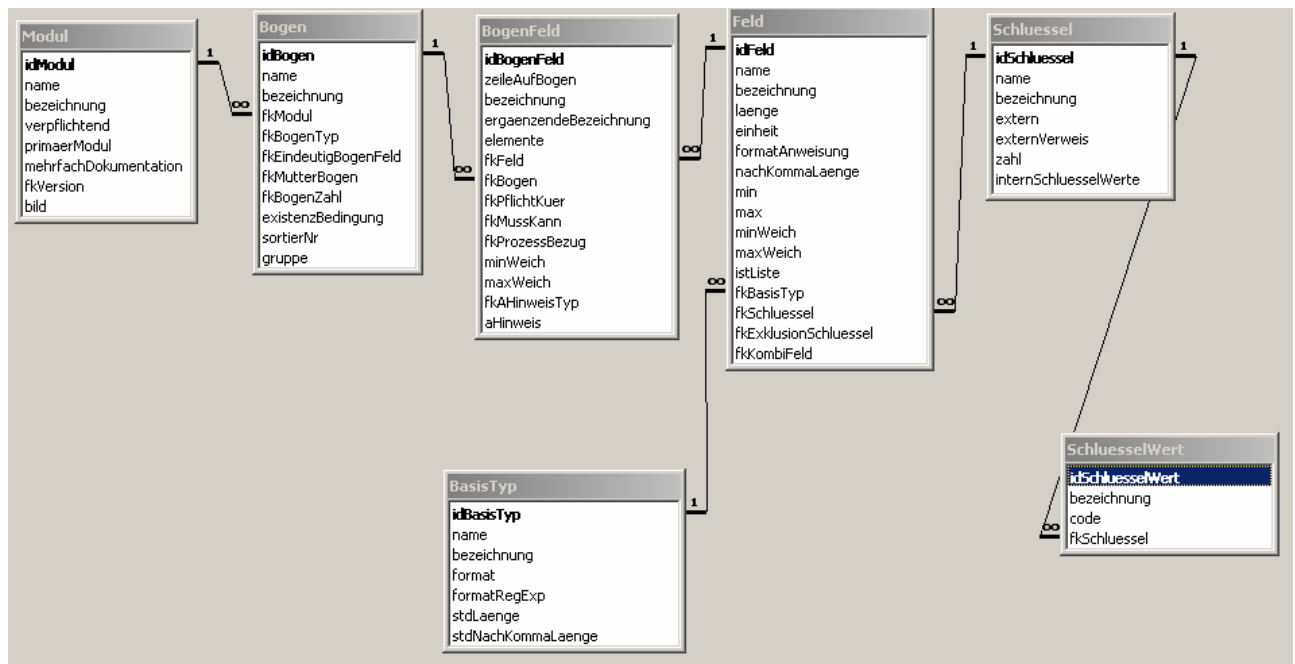


Abbildung 1 Tabellen und Relationen der Datenfeldbeschreibung

Die Abfragen *Datenfeldbeschreibung* und *DatenfeldbeschreibungFürEinModul* der Access-Datenbank ermöglichen den vereinfachenden Blick auf diese Struktur.

Das für den Anwender wichtigste Merkmal ist die Bezeichnung des Datenfeldes (Attribut `BogenFeld.bezeichnung`).

Die Datenfeldbeschreibung orientiert sich eng am Dokumentationsbogen („*Bogensicht*“). Im Kontext einer integrierten, prozessorientierten Krankenhaussoftware müssen die Teildatensätze nicht direkt in Eingabeformulare umgesetzt werden. Es ist sinnvoller, die Teile eines Doku-

<sup>4</sup> Ein Dokumentationsbogen oder kurz Bogen ist als eine Menge von auszufüllenden Datenfeldern zu verstehen. Die Papierform ist hier nur als eine Erscheinungsform des Dokumentationsbogens zu verstehen. Man spricht besser von Eingabeformular oder Eingabemaske.

mentationsbogens zu dem Zeitpunkt und in dem Dokumentationskontext zu erfragen, der sich in den Prozessablauf des Krankenhauses einordnet.

## 2.1 Aufbau der Datenfeldbeschreibung

### 2.1.1 Ziele

- Bereitstellung der Informationen, welche für die Programmierung des Eingabeformulars und die Sicherung der eingegebenen Daten nötig sind,
- Vermeidung von Redundanzen,
- Typisierung der Felder nach fachlichen und datentechnischen Kriterien.

### 2.1.2 Module (Datensätze)

Ein Modul der BQS-Spezifikation enthält die Datensatzdefinition mindestens eines medizinischen Leistungsbereiches (Beispiele: Geburtshilfe, Cholezystektomie)<sup>5</sup>. Die QS-Dokumentationssoftware kann für einen Behandlungsfall eine oder mehrere Moduldokumentationen anlegen, welche nach Dokumentationsabschluss an die datenentgegennehmenden Stellen übermittelt werden. Fehlerfreie Moduldokumentationen (verkürzt "Module") werden dem Krankenhaus von der datenentgegennehmenden Stelle bestätigt. Die BQS-Bundesauswertung basiert auf den Moduldokumentationen der Krankenhäuser.

Aus technischer Sicht ist das Modul durch einen eindeutigen Namen gekennzeichnet. Es umfasst mindestens einen Teildatensatz (Kapitel 2.1.3). In der Tabelle `Modul` (Tabelle 2) der BQS-Spezifikation finden sich die zentralen Definitionen eines Moduls.

Tabelle 2: Struktur der Tabelle `Modul`

Feldname	Feldtyp	Bemerkung
idModul	INTEGER	Primärschlüssel
name	TEXT(32)	Eindeutiger technischer Name

<sup>5</sup> In der Regel entspricht ein Modul genau einem Leistungsbereich (z.B. Perinatalmedizin). Das Modul Herzchirurgie enthält aber mehrere Leistungsbereiche: Koronarchirurgie, Aortenklappenchirurgie u.a. Die Module bzw. Leistungsbereiche werden von Fachgruppen der BQS erarbeitet (Überblick über die Fachgruppen unter [www.bqs-online.de](http://www.bqs-online.de)).



Feldname	Feldtyp	Bemerkung
bezeichnung	TEXT(255)	Erläuternde Bezeichnung
verpflichtend	BOOLEAN	Besteht für das Modul eine QS-Dokumentationsverpflichtung?
primaerModul	BOOLEAN	Ist das Modul ein Primärmodul?
mehrfachDokumentation	BOOLEAN	Ist ein mehrfaches Anlegen eines gleichartigen Datensatzes pro Krankenhausfall zulässig (ja/nein)?
direkt	BOOLEAN	handelt es sich um ein direktes Datenexportverfahren
indirekt	BOOLEAN	handelt es sich um ein indirektes Datenexportverfahren

### *Auslösung der Moduldokumentation*

Der auslösende Sachverhalt für die Dokumentationspflicht ist in der BQS-Spezifikation für QS-Filter-Software ([www.bqs-online.de](http://www.bqs-online.de)) definiert. Die QS-Filter-Software greift zu diesem Zweck auf administrative Routinedaten (z.B. Haupt- und Nebendiagnosen und Prozeduren) zurück, welche in jedem Krankenhausinformationssystem (KIS) verfügbar sind und von den Krankenhäusern auch für die Umsetzung der Datenübermittlungsvereinbarung gemäß § 301 Abs. 3 SGB V (kurz: DÜV-301) benötigt werden.

### *Primärmodule - Minimaldatensatz*

Für Primärmodule sind in der BQS-Spezifikation für QS-Filter-Software Auslösebedingungen definiert. Das Modul *MDS* (*Minimaldatensatz*) besitzt dagegen keinen definierten Auslöser für die Dokumentationspflicht.

*Der Anwender darf den Minimaldatensatz nur dann verwenden, wenn sich die erbrachte Leistung nicht im vorgesehenen Modul dokumentieren lässt (vgl. Ausfüllhinweise zum Minimaldatensatz).* Bei der Dokumentation des Minimaldatensatzes ist immer anzugeben, anstelle welchen Primärmoduls er angewendet wird (Datenfeld `ZUQSMODUL` "Zugehöriger QS-Datensatz").

Der Minimaldatensatz, welcher überwiegend administrative Daten des Behandlungsfalls enthält, ist erforderlich für den Vollständigkeitsabgleich der QS-Dokumentationen eines Krankenhauses.

### *Sekundärmodule*

Neben dem Minimaldatensatz sind die Follow-Up-Module, wie z.B. das Modul HTXFU (Follow-Up-Herztransplantation) Sekundärmodule. Der Datensatz HTXFU ist jeweils nach 1, 2 oder 3 Jahren von demjenigen Krankenhaus zu dokumentieren, in welchem die Transplantation erbracht worden ist. Da für das Follow-Up kein neuer OPS-Kode erbracht worden ist, wird dieser Datensatz nicht direkt vom QS-Filter ausgelöst.

### *Mehrfachdokumentation*

Pro Krankenhausfall darf höchstens ein Datensatz eines Moduls angelegt und exportiert werden, wenn in der Spalte `mehrfachDokumentation` der Tabelle `Modul` `nein` angegeben ist.

### Beispiel:

Werden während eines stationären Aufenthaltes bei einer Patientin zwei Brustoperationen durchgeführt, so dürfen hierfür nicht zwei Datensätze 18/1 angelegt werden. Stattdessen sind die Operationen in mehreren Teildatensätzen eines Datensatzes zu dokumentieren. Dagegen ist die doppelte Anlage eines Datensatzes 17/2 (Hüft-Endoprothesen-Erstimplantation) während eines stationären Aufenthaltes erforderlich, falls der Patient beidseitig operiert wird.

### Achtung:

Die QS-Dokumentationssoftware soll sicherstellen, dass die Mehrfachdokumentation gleichartiger Datensätze für einen Krankenhausfall unterbunden wird, falls diese nicht zulässig ist.

## **2.1.3 Teildatensätze**

Die Begriffe *Teildatensatz* und *Bogen* werden als *Synonyme* gebraucht. Bei der Papierdokumentation entspricht ein Teildatensatz einem Teil eines Dokumentationsbogens<sup>6</sup>. Man-

---

<sup>6</sup> Grundsätzlich ist die "Bogensicht" die Sicht der medizinischen Fachgruppen, welche die Module entwickeln. Bei verteilten Softwarelösungen für das Krankenhaus ist die Bogensicht dann nicht mehr adäquat, wenn die Bestandteile eines Bogens auf verschiedene Teilsysteme verteilt sind. Die Daten eines Bogens werden für den Export aus den einzelnen Teilsystemen zusammenge-

che Teildatensätze (z.B. Kind-Teildatensatz in Geburtshilfe, 16/1) müssen unter bestimmten Umständen mehrfach pro Datensatz ausgefüllt werden.

Ein Teildatensatz ..

- ist jeweils einem Modul zugeordnet,
- besitzt einen Namen, der innerhalb eines Moduls eindeutig ist,
- kann unter definierten Bedingungen mehrfach pro Fall erzeugt werden.

Die Teildatensätze der BQS-Spezifikation sind in der Tabelle `Bogen` definiert (Tabelle 3).

Tabelle 3: Struktur der Tabelle `Bogen`

Feldname	Feldtyp	Bemerkung
<code>idBogen</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name des Teildatensatzes
<code>bezeichnung</code>	TEXT	Beschreibender Text
<code>extistenzBedingung</code>	MEMO	Logische Bedingung (Regeln für das Anlegen von Teildatensätzen)
<code>fkModul</code>	INTEGER	Obligatorischer Fremdschlüssel zu einem Modul
<code>fkBogenZahl</code>	TEXT(1)	Anzahl der auszufüllenden Teildatensätze pro Patient (bezogen auf den Basisbogen oder ggf. auf den Mutterteildatensatz)
<code>fkMutterBogen</code>	INTEGER	Optionaler Fremdschlüssel, welcher den Mutter-Teildatensatz eines Teildatensatzes definiert
<code>fkBogenTyp</code>	TEXT(1)	spezifiziert den für den Export relevanten Bogentyp: Mögliche Werte B, K oder O. Die Angabe ist obligatorisch.
<code>fkEindeutigBogenFeld</code>	INTEGER	Fremdschlüssel auf ein Bogenfeld, welches mehrfach vorhandene Teildatensätze eines Datensatzes identifiziert

Die referenzierten Tabellen `BogenZahl` und `BogenTyp` sind in Kapitel 1.5.2 beschriebene "Nachschlagtabellen".

### *Benennung von Teildatensätzen*

---

stellt. Die Papierbögen werden lediglich als Layoutinformation zur Verfügung gestellt. Diese Papierbögen sind zur Dokumentation nicht zugelassen.

Ein Teildatensatz wird durch die folgende Kombination von Modulnamen und Bogennamen identifiziert und angesprochen:

<Modul.name>:<Bogen.name>

Beispiele:

- 01/1:B ist der Basisbogen des Moduls Dekompression bei Karpaltunnelsyndrom,
- 21/3:PCI ist der PCI-Teildatensatz des Moduls "Koronarangiographie und Perkutane Koronarintervention (PCI)",
- HCH:O ist der operative Teildatensatz des Moduls Herzchirurgie.

*Bogentyp - Kerndatensatz vs. optionaler Datensatz*

Das Attribut `fkBogenTyp` definiert für jeden Teildatensatz seine Rolle im und seine Zugehörigkeit zum Kerndatensatz. Der Basisteildatensatz ist immer Bestandteil des so genannten Kerndatensatzes.

Tabelle 4: Inhalte der Tabelle `BogenTyp`

idBogenTyp	Bezeichnung
B	Basisteildatensatz (Teil des Kerndatensatzes)
K	Teildatensatz ist Teil des Kerndatensatzes und kein Basisteildatensatzes
O	Teildatensatz ist Teil des optionalen Datensatzes <sup>7</sup>

*Hierarchie von Teildatensätzen*

Der Ausgangspunkt ("root") für die Teildatensatzhierarchie eines Moduls ist immer der Basisteildatensatz (Wert `B` des Attributes `fkBogenTyp`). Ein abhängiger Teildatensatz besitzt einen Mutterteildatensatz, der über das Attribut `fkMutterBogen` definiert ist<sup>8</sup>.

<sup>7</sup> Im Verfahrensjahr 2009 nicht relevant: Alle optionalen Teildatensätze bilden den optionalen Datensatz. Falls Plausibilitätsverletzungen auftreten, welche sich ausschließlich auf den optionalen Datensatz beziehen, so werden die plausiblen Teildatensätze des Kerndatensatzes trotzdem von der Datenstelle entgegengenommen und bescheinigt (Kapitel 6.5.1).

<sup>8</sup> Falls der Mutterteildatensatz nicht über das Attribut `fkMutterBogen` explizit definiert ist, so gilt implizit der Basisteildatensatz des Moduls als Mutterteildatensatz.

Auf diese Weise lässt sich für jedes Modul ein "Hierarchiebaum" der Teildatensätze aufbauen.

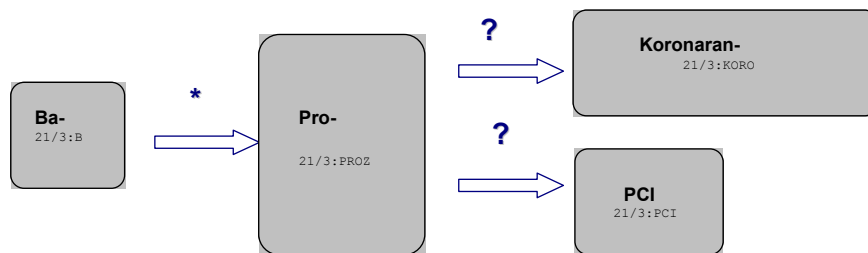


Abbildung 2 Teildatensatzstruktur des Datensatzes 21/3

Das Modul 21/3 (Abbildung 2) enthält die Teildatensätze:

21/3 : B = Basis

21/3 : PROZ = Prozedur

21/3 : KORO = Koronarangiographie

21/3 : PCI = PCI

In der Tabelle `Bogen` sind folgende Bezüge zum Mutterteildatensatz definiert:

21/3 : B hat keinen Mutterteildatensatz

21/3 : PROZ hat den Mutterteildatensatz 21/3 : B

21/3 : PCI hat den Mutterteildatensatz 21/3 : PROZ

21/3 : KORO hat den Mutterteildatensatz 21/3 : PROZ

### *Regeln für das Anlegen von Teildatensätzen*

Jedes Modul muss die Definition genau eines Basisteildatensatzes enthalten (Wert *B* des Attributes `fkBogenTyp`). Wenn die Dokumentation eines Moduls durchgeführt wird, muss der Basisteildatensatz genau einmal angelegt werden (z.B. in der Exportdatei).

Das Attribut `fkBogenZahl` gibt Auskunft darüber, wie oft ein Teildatensatz pro Vorgang angelegt werden darf. Folgende Werte des Attributs sind möglich:

- 1 = Genau ein Teildatensatz muss ausgefüllt werden
- + = Mindestens ein Teildatensatz muss ausgefüllt werden
- ? = Höchstens ein Teildatensatz darf ausgefüllt werden

\* = Eine beliebige Anzahl von Teildatensätzen kann ausgefüllt werden

Die Kardinalität eines abhängigen Teildatensatzes bezieht sich auf den Mutterteildatensatz. Der Basisteildatensatz hat immer die Kardinalität 1.

Z.B. definiert die Ausprägung `fkBogenZahl = *` eine 1-N-Beziehung.

Man beachte, dass das Attribut `fkBogenZahl` wichtig für das Verfahren der Entgegennahme von Datensätzen ist (Kapitel 6.5).

#### Beispiele:

- Der Teildatensatz `01/1:B` muss als Basisteildatensatz genau einmal ausgefüllt werden (`fkBogenZahl = 1`).
- Der Teildatensatz `HCH:O` muss mindestens einmal pro Datensatz angelegt werden (`fkBogenZahl = +`)
- Der Teildatensatz `21/3:PCI` im Modul „Koronarangiographie und Perkutane Koronarintervention (PCI)“ braucht nur dann angelegt zu werden, wenn auch wirklich eine PCI durchgeführt wurde. Es kann also eine beliebige Anzahl von Teildatensätzen angelegt werden. Trotzdem gilt `fkBogenZahl = ?`, da - bezogen auf jeden Mutterteildatensatz `21/3:PROZ` - maximal ein Teildatensatz existieren darf.

Man beachte weiterhin, dass die im Attribut `fkBogenZahl` definierten Kardinalitäten durch Definitionen in den nachfolgend beschriebenen Attributen `existenzBedingung` bzw. `fkEindeutigBogenFeld` eingeschränkt werden können.

#### *Inhaltliche Voraussetzung für das Anlegen von Teildatensätzen*

Das Attribut `existenzBedingung` ist eine logische Bedingung (Syntax gemäß Kapitel 3.4) für das Anlegen eines Teildatensatzes. Die referenzierten Bogenfelder der Existenzbedingung beziehen sich auf den Mutterteildatensatz.

Die Krankenhaussoftware muss die Existenzbedingung als Trigger für das Anlegen eines abhängigen Teildatensatzes nutzen.

Wenn die Existenzbedingung eines potenziellen Kindteildatensatzes erfüllt ist, so **muss** der Kindteildatensatz auch angelegt und übermittelt werden.

Andererseits gilt: Wenn die entgegennehmende Stelle einen Kindteildatensatz erhält, für den die zugehörige Existenzbedingung im Mutterteildatensatz nicht erfüllt ist, so ist das eine relationale Plausibilitätsverletzung.

Beispiel: Modul 21/3

Der Teildatensatz 21/3:KORO darf nur innerhalb eines Vorgangs angelegt werden, wenn im zugehörigen Mutterteildatensatz 21/3:PROZ folgende Bedingung erfüllt ist:

ARTPROZEDUR IN (1; 3).

Wenn ein Benutzer im Feld ARTPROZEDUR den Code 1 ("Diagnostische Koronarangiographie") auswählt, so

- 1) muss der abhängige Teildatensatz 21/3:KORO angelegt werden.
- 2) darf der Teildatensatz 21/3:PCI nicht angelegt werden.
- 3) muss ein bereits angelegter Teildatensatz 21/3:PCI wieder gelöscht werden.

*Identifizierende Attribute mehrfach vorhandener Teildatensätze*

Teildatensätze, welche mehr als einmal ausgefüllt werden dürfen (Werte + und \* des Attributes `fkBogenZahl`), sind nicht mehr durch die Vorgangsnummer voneinander unterscheidbar. Diese Teildatensätze benötigen ein zusätzliches identifizierendes Bogenfeld, welches im Attribut `fkEindeutigBogenFeld` festgelegt wird. Beim Teildatensatz 21/3:PCI ist es das Bogenfeld `LFDNRPTCA`.

Beim Anlegen einer Tabelle für die Speicherung eines mehrfach vorhandenen Teildatensatzes muss der Primärschlüssel mindestens die Attribute `VorgangsNr`, `VersionsNr`<sup>9</sup> und das in `fkEindeutigBogenFeld` definierte Feld umfassen.

Beispiel: Modul 10/1

Die Follow-Up-Dokumentation des Moduls 10/1 ist freiwillig. Daher hat der Teildatensatz 10/1:FUB die Kardinalität ?. Mindestens ein Follow-Up der Strombahn muss dann erhoben werden, wenn auch das Basis-Follow-Up existiert: Der Teildatensatz 10/1:FUS hat die Kardinalität \* und den Mutterteildatensatz 10/1:FUB.

Die Existenzbedingung des Teildatensatzes 10/1:FUS lautet:

FUERHEBDATUM <> LEER.

Dadurch ist sichergestellt, dass mindestens ein Teildatensatz 10/1:FUS erstellt wird (FUERHEBDATUM ist Muss-Feld).

Durch das identifizierende Attribut 10/1:FUS:FUOPSTROMBAHN ist gewährleistet, dass maximal vier verschiedene Teildatensätze angelegt werden (für jede Strombahn und jede Seite einen).

---

<sup>9</sup> Bei der entgegennehmenden Stelle kommt noch das Feld `RegistrierNr` hinzu, da dort Datensätze verschiedener Krankenhäuser gesammelt werden.

Beispiel: Modul HCH

Die Follow-Up-Dokumentation des Moduls `HCH` ist ebenfalls freiwillig. Daher hat der Teildatensatz `HCH:FU` die Kardinalität `?`. Höchstens ein Bogen darf ausgefüllt werden. Die Software muss dem Anwender alle freiwillig auszufüllenden Teildatensätze zur Verfügung stellen!

#### 2.1.4 Datenfelder (Bogenfelder)

Jedes auf einem Teildatensatz vorhandene und auszufüllende Feld wird als Datenfeld (Item, Bogenfeld) bezeichnet. Datenfelder sind charakterisiert durch ihren Namen (Bezeichnung) und die Spezifikation des einzutragenden Inhalts.

Die Bezeichnung<sup>10</sup> wird so gewählt, dass sie einem medizinischen Experten unmittelbar verständlich ist. Die Spezifikation des Inhaltes umfasst dagegen sowohl eine fachliche (medizinische) als auch datentechnische Typisierung. Dagegen repräsentieren die in der Tabelle `Feld` aufgelisteten Felder inhaltlich gleiche Dokumentationsfelder mehrerer Module (Kapitel 2.1.5). Der datentechnische Typ (`BasisTyp`) charakterisiert das Format des Feldes (Kapitel 2.1.6).

Jedes Datenfeld hat zwingend einen Bezug zu einem Teildatensatz und zu einem medizinischen Feld. Der Name des Datenfeldes ist identisch mit dem Namen des medizinisch-fachlichen Feldes. Weitere Eigenschaften sind der Text, welcher das Item auf dem Erhebungsformular kennzeichnet, und die fortlaufende Nummer im Teildatensatz. Die Datenfelder sind in der Tabelle `BogenFeld` gespeichert.

Identifizierendes Merkmal eines Datenfeldes ist eine Kombination aus `fkBogen` und `fkFeld`. Das bedeutet, dass das Datenbankschema gewährleistet, dass der technische Feldname (`Feld.name`) in einem Teildatensatz maximal einmal vorkommt.

Per definitionem muss ein Datenfeldname sogar innerhalb eines Moduls eindeutig sein. D.h., dass eine Abfrage mit dem Primärschlüsselpaar (`modulNr`, `feldNr`) höchstens einen Primärschlüssel `idBogenFeld` liefert.

Die Tabelle `BogenFeld` ist eine Verknüpfungstabelle, die wegen ihrer zentralen Bedeutung - im Gegensatz zu anderen Verknüpfungstabellen - eine Ausnahmeregelung besitzt: Diese äußert sich darin, dass hier ein Primärschlüssel `idBogenFeld` definiert ist.

---

<sup>10</sup> ggf. im Kontext der Überschriften (Kapitel 2.2)



Tabelle 5: Struktur der Tabelle BogenFeld

Feldname	Feldtyp	Bemerkung
idBogenFeld	INTEGER	Primärschlüssel
fkBogen	INTEGER	Fremdschlüssel zu dem Teildatensatz und zu dem
fkFeld	INTEGER	Feld, bilden zusammen die identifizierenden Merkmale
zeileAufBogen	DOUBLE	Zeile in dem Dokumentationsbogen. Bei mehreren Feldern in einer Zeile kann es auch die Form <nn,m> haben
bezeichnung	TEXT	Beschreibender Text zum Feld auf dem Dokumentationsbogen. Wenn der Feldinhalt leer ist, so wird der Inhalt des gleichnamigen Feldes in der Tabelle Feld genommen.
elemente	INTEGER	Anzahl der Elemente bei Listenfeldern
fkMussKann	TEXT(1)	„M“ oder „K“, Unterscheidung zwischen Muss- und Kann-Feldern
minWeich	DOUBLE	Weiche Untergrenze des Wertebereiches eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
maxWeich	DOUBLE	Weiche Obergrenze des Wertebereiches eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
aHinweis	TEXT(32)	Name des HTML-Ausfüllhinweises ohne Endung .htm (Kapitel 2.3)

### *Muss- und Kann-Felder*

Jedes Bogenfeld ist als Muss- oder Kann-Feld zu deklarieren:

- 1) Ein Muss-Feld muss innerhalb eines angelegten Teildatensatzes immer ausgefüllt sein (Kapitel 3.2.5)<sup>11</sup>.
- 2) Kann-Felder deklarieren dagegen abhängige Felder in Feldgruppen und müssen nur unter bestimmten Bedingungen ausgefüllt werden. Wenn also logische Sachverhalte dem Ausfüllen von Kann-Feldern entgegenstehen, so dürfen sie nicht ausgefüllt werden.

### *Anzahl der Elemente von Listenfeldern*

---

<sup>11</sup> In jedem Muss-Feld muss für jeden angelegten Teildatensatz einmal eine Angabe erfolgen.

Das Attribut `elemente` ist nur relevant bei von Listenfeldern (vgl. Attribut `Feld.istListe`) abgeleiteten Bogenfeldern (Bogenfeldlisten). Es gibt die Größe der Bogenfeldliste an. Wenn für eine Bogenfeldliste das Attribut `elemente` leer ist, so ist die Größe *per definitionem* 1.

Wenn ein Listenfeld als Muss-Feld deklariert ist, so ist nur das erste Exportfeld der Liste ein Muss-Feld, die restlichen Elemente sind dann Kann-Felder. Wenn ein Listenfeld als Kann-Feld deklariert ist, so sind alle exportierten Elemente ebenfalls Kann-Felder.

### 2.1.5 Felder - ein erster Schritt zur Prozess- und Datenintegration

Die Einführung der Tabelle `Feld` (Tabelle 6) erleichtert dem Softwarehersteller den Abgleich seines Datenmodells mit dem Datenmodell der BQS. Redundante Information auf der Menge aller Dokumentationsbögen muss dadurch nicht redundant abgebildet werden. Z.B. taucht das Feld `ENTLGRUND` (Entlassungsgrund) in den meisten Modulen auf. Um die mehrfache Pflege dieser Felder zu vermeiden, wird ein Feld mit dem Namen `ENTLGRUND` definiert und jeweils nur noch in der Tabelle `BogenFeld` referenziert.

Jedem Feld ist zwingend ein Basistyp zugeordnet (Kapitel 2.1.6). Bei Schlüsselfeldern muss auch ein Schlüssel assoziiert sein. Im Gegensatz zu den (technischen) Basistypen enthalten die Felder die medizinisch-fachliche Information der Datenfelder. Der fachliche Inhalt wird durch den Text im Attribut `bezeichnung`<sup>12</sup> beschrieben.

Identifizierendes Attribut eines Feldes ist allein sein technischer Name (Attribut `name`). Dieses ist ganz wichtig für die Eindeutigkeit von Feldnamen innerhalb eines Moduls. Felder mit unterschiedlichen Typen oder unterschiedlichen Schlüsseln müssen auch unterschiedliche Namen haben.

Ein Feld kann als Skalar oder als Liste definiert sein. Diese Eigenschaft wird über das Attribut `istListe` gesteuert. Jedes von einem Listenfeld abgeleitete Bogenfeld ist automatisch eine Liste<sup>13</sup>. Die Anzahl der Elemente des von einem Feld abgeleiteten Bogenfeldes wird über das Attribut `elemente` der Tabelle `BogenFeld` gesteuert.

#### Beispiel:

---

<sup>12</sup> Das Attribut `bezeichnung` ist ein Standardtext für das gleichnamige Attribut der Tabelle `BogenFeld`. Im Eingabeformular wird die Bezeichnung aus der Tabelle `BogenFeld` angezeigt.

<sup>13</sup> Man beachte die Besonderheiten der Listenfelder beim Datenexport und in der Syntax der Plausibilitätsregeln.

Das Feld `AUFNDIAG` (Aufnahmediagnosen) ist als Liste definiert. Im Modul 17/1 enthält das entsprechende Bogenfeld fünf Elemente, im Modul 16/1 ist das Attribut `BogenFeld.elemente` der Aufnahmediagnose leer. Auch im Modul 16/1 ist die Aufnahmediagnose ein Listenfeld, aber mit nur einem Element. Insbesondere bei der Verwendung der richtigen Operatoren in den Plausibilitätsregeln und Feldgruppen ist die Listendefinition eines Feldes wichtig.

Grundsätzlich gilt:

Die Entscheidung, ob ein Bogenfeld ein Skalar oder Listenfeld ist, wird in der Tabelle `Feld` getroffen. Alle von einem Listenfeld abgeleiteten Bogenfelder sind automatisch auch Listenfelder. Die Größe der Liste wird individuell in der Tabelle `BogenFeld` konfiguriert.

Die Tabelle `Feld` soll für den Softwarehersteller ein Hilfsmittel sein, um künftig nicht mehr in enger Anlehnung an die Bogenstruktur der Spezifikation das Datenmodell zu entwickeln. Die BQS favorisiert Software-Lösungen, welche sich gut an die Krankenhausprozesse anpassen lassen.

Die Tabelle `Feld` ist also ein Schritt weg von der bereits angesprochenen *Bogensicht*. Es können Felder identifiziert werden, welche nicht in einem Qualitätssicherungsbogen, also im Kontext einer Operation, dokumentiert werden. Ein Beispiel ist die *Aufnahmediagnose*, welche gestellt wird, bevor überhaupt klar ist, welche Art von Qualitätsdokumentation durchzuführen ist.

Tabelle 6 Struktur der Tabelle `Feld`

Feldname	Feldtyp	Bemerkung
<code>idFeld</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name (Erlaubte Zeichen: A-Z, 0-9, Ziffer nicht am Anfang)
<code>bezeichnung</code>	TEXT	Beschreibender Text auf dem Dokumentationsbogen (Standardwert für gleichnamiges Feld in Tabelle <code>BogenFeld</code> )
<code>laenge</code>	INTEGER	Anzahl der Zeichen in der Feldeingabemaske, enthält beim Typ <code>ZAHL</code> auch das Komma, bei <code>SCHLUESSEL</code> die Trennzeichen
<code>nachKommaLaenge</code>	INTEGER	Anzahl der Nachkommastellen in der Feldeingabemaske (muss kleiner als <code>laenge</code> sein)
<code>fkBasisTyp</code>	INTEGER	Fremdschlüssel zur Tabelle <code>Basistypen</code>

Feldname	Feldtyp	Bemerkung
fkSchluessel	INTEGER	Fremdschlüssel zur Tabelle Schlüsseltypen
istListe	BOOL	Wenn <code>istListe</code> wahr ist, so sind die vom betreffenden Feld abgeleiteten Bogenfelder Listenfelder.
fkKombiFeld	INTEGER	Optionaler Fremdschlüssel auf ein anderes Feld, welches Kombinationsfelder kennzeichnet
einheit	TEXT(50)	Einheit des Feldes (z.B. mm, Stunden)
formatAnweisung	TEXT	Formatanweisung zum Ausfüllen des Datenfeldes (z.B. TT.MM.JJJJ)
min	DOUBLE	Harte Untergrenze des Wertebereiches eines numerischen Datenfeldes (modulübergreifend). Die Definition ist optional.
max	DOUBLE	Harte Obergrenze des Wertebereiches eines numerischen Datenfeldes (modulübergreifend). Die Definition ist optional.
minWeich	DOUBLE	Weiche Untergrenze des Wertebereiches eines numerischen Datenfeldes (modulübergreifend). Die Definition ist optional.
maxWeich	DOUBLE	Weiche Obergrenze des Wertebereiches eines numerischen Datenfeldes (modulübergreifend). Die Definition ist optional.

### *Kombinationsfelder*

Für manche Bogenfelder ist zwingend vorgeschrieben, dass sie innerhalb eines Moduls in Kombination mit einem anderen Bogenfeld existieren.

Die Definition von Kombinationsfeldern geschieht mit Hilfe des optionalen Fremdschlüssels `fkKombiFeld` in der Tabelle `Feld`.

## 2.1.6 Basistypen

Das Hauptmerkmal eines Basistyps ist der technische Typ eines Eingabefeldes (z.B. Zeichenkette, numerischer Typ, Datum usw.). Wichtiges Charakteristikum ist die Beschreibung des Eingabeformats. Die Basistypen sind Voraussetzung für die Beschreibung einer formalen Regelsyntax (Kapitel 3.4).

Das identifizierende Merkmal eines Basistyps ist sein technischer Name (Attribut `name`).

Tabelle 7: Struktur der Tabelle `BasisTyp`

Feldname	Feldtyp	Bemerkung
----------	---------	-----------

Feldname	Feldtyp	Bemerkung
idBasistyp	INTEGER	Primärschlüssel
name	TEXT	Technischer Name (muss eindeutig sein)
bezeichnung	TEXT	
format	TEXT	Formatdefinition, z.B. TT.MM.JJJJ beim Basistyp Datum
formatRegExp	TEXT	Regulärer Ausdruck für die Formatprüfung
stdLaenge	INTEGER	Vorschlagsfeld für das gleichnamige Feld in der Tabelle Feld (einschließlich Vorzeichen und Komma)
stdNachKommaLaenge	INTEGER	Vorschlagsfeld für das gleichnamige Feld in der Tabelle Feld

#### Anmerkungen:

- In Zeichenketten (Basistyp `TEXT`) sind alle Zeichen des ASCII-Formats mit einem Kode > 32 erlaubt. Ausgenommen sind das Semikolon, die doppelten Anführungsstriche und Hochkommata.
- Es gibt zwei Arten von Schlüsseln: numerisch und nichtnumerisch (vgl. Kapitel 2.1.7).
- Das Komma trennt die Nachkommastellen, Vorzeichen + und – sind erlaubt.

### 2.1.7 Schlüssel

Identifizierendes Merkmal eines Schlüssels (Kodesystem) ist sein technischer Name. Die meisten Schlüsselcodes sind in der Tabelle `Schlüsselwert` (Kapitel 2.1.8) definiert.

Tabelle 8: Struktur der Tabelle `Schlüssel`

Feldname	Feldtyp	Bemerkung
idSchluessel	INTEGER	Primärschlüssel
name	TEXT	Technischer Name (muss eindeutig sein)
bezeichnung	TEXT	
extern	BOOL	Zeigt an, ob der Schlüssel in der Tabelle <code>Schlüssel</code> (=falsch) oder in einer externen Tabelle gespeichert (=wahr) ist.
externVerweis	TEXT	Verweis auf die Quelle des externen Schlüssels
zahl	BOOL	Wenn wahr, sind die Werte im Attribut <code>code</code> der zugehörigen Schlüsselwerte als ganze Zahl kodiert, ansonsten als Zeichenkette.
sortierNrVerwendet	BOOL	Flag, welches anzeigt, ob für die Reihenfolge das Attribut <code>sortierNr</code> der Tabelle <code>Schlüsselwert</code> herangezogen wird.

Feldname	Feldtyp	Bemerkung
fkMutterSchluessel	INTEGER	Referenz auf einen übergeordneten Schlüssel. Z.B. enthält der Schlüssel <i>MaSarkome</i> ausschließlich Codes des Schlüssels <i>ICDO3Mamma</i> . Abgeleitete Schlüssel enthalten in der Regel keine Bezeichnungen (Datenbanktabelle <i>SchluesselWert</i> ), da diese bereits im „Mutterschlüssel“ definiert sind.

Schlüsselcodes können auf zwei Arten interpretiert werden. Wenn das Attribut `zahl` gesetzt ist, so werden die Codes als ganze Zahl gedeutet. Ansonsten werden sie als Zeichenketten interpretiert. In der Syntax der Plausibilitätsregeln werden die letztgenannten Codes in einfache Hochkommata gesetzt (vgl. Kapitel 3.4).

#### *Externe Schlüsselkataloge*

Externe Schlüsselkataloge sind über das Attribut `extern` deklariert. Hinweise zu den Bezugsquellen sind in der Spalte `externVerweis` zu finden (z.B. <http://www.dimdi.de>). Externe Schlüsselkataloge werden nicht von der BQS bereitgestellt und somit auch nicht verantwortet.

#### Achtung:

Der Softwareanbieter oder die datenentgegennehmende Stelle hat dafür Sorge zu tragen, dass die aktuellen externen Schlüsselkataloge in der Software verwendet werden.

Der *Fachabteilungsschlüssel* (`Fachabt`) ist ein externer Schlüsselkatalog: Die Schlüsselcodes sind nicht mehr Bestandteil der Datenbank und müssen über die § 301-Vereinbarung (<http://www.dkgev.de>) bezogen werden. Der Fachabteilungsschlüssel wird wie bisher numerisch interpretiert. Es ist somit egal, ob der Wert *100* oder *0100* für die Abteilung *Innere Medizin* übermittelt wird.

Ebenso ist der *Entlassungsgrund* (`EntlGrund`) ein externer Schlüssel, der als Schlüssel 5 in Anlage 2 der § 301-Vereinbarung definiert ist: Die 1. und 2. Stelle dieses § 301-Schlüssels werden im Rahmen der BQS-Spezifikation numerisch kodiert (siehe Attribut `zahl`). Die Schlüsselcodes, welche der BQS zum Zeitpunkt der Publikation der BQS-Spezifikation bekannt sind, sind in der Tabelle `SchluesselWert` enthalten. Spätere

Schlüsseländerungen bzw. -fortschreibungen müssen vom Softwareanbieter und von der datenentgegennehmenden Stelle selbstständig und zeitnah übernommen werden.

## 2.1.8 Schlüsselwerte

Tabelle 9 gibt einen Überblick über die Datenbanktabelle `SchluesseWert`, in welcher die Codes und Bezeichnungen der Schlüssel hinterlegt sind.

Identifizierendes Merkmal ist hier eine Kombination der Spalten `fkSchluessel` und `code`. Das bedeutet, dass jeder Schlüsselcode innerhalb eines Schlüssels nur einmal vorkommen darf.

Tabelle 9: Struktur der Tabelle `SchluesseWert`

Feldname	Feldtyp	Bemerkung
<code>idSchluesselWert</code>	INTEGER	Primärschlüssel
<code>fkSchluessel</code>	INTEGER	Fremdschlüssel zur Tabelle Schlüssel
<code>code</code>	TEXT(50)	Schlüsselcode (entweder numerisch oder alphanumerisch kodiert)
<code>bezeichnung</code>	TEXT	Textliche Definition des Schlüsselwertes
<code>sortierNr</code>	INTEGER	optionale Angabe zur Reihenfolge der Schlüsselwerte: Wenn belegt, so ist diese Reihenfolge bei der Anzeige in der Erfassungssoftware einzuhalten.

Das Attribut `code` ist ein Textfeld, welches in Abhängigkeit vom Wert des Attributes `zahl` im zugeordneten Schlüssel entweder numerisch oder nichtnumerisch interpretiert wird.

Wenn in einer Plausibilitätsregel (Kapitel 3.2.2 und 3.4) Felder mit numerischen Schlüsseln (Basistyp `NUMSCHLUESSEL`) vorkommen, so werden bei der Evaluierung der Regel die Schlüsselcodes wie ganze Zahlen behandelt. Z.B. ist der numerische Schlüsselcode `07` gleichwertig mit `7`. Bei einem nichtnumerischen Schlüsselfeld (Basistyp `SCHLUESSEL`) hätten die beiden Codes eine unterschiedliche Bedeutung.

### *Sortierung der Codes*

Für die Codes (Attribut `SchluesseWert.code`) eines Schlüssels ist eine Sortierung definiert. Die Art der Sortierung wird über die Attribute `zahl` und `sortierNr` verwendet der Tabelle `Schluesse` festgelegt.

- Numerische Sortierung: Wenn `sortierNrVerwendet = nein` und `zahl = ja`, so sind die Codes nach der Spalte `code` der Tabelle `Schluessel` numerisch zu sortieren.
- Alphanumerische Sortierung: Wenn `sortierNrVerwendet = nein` und `zahl = nein`, so sind die Codes nach der Spalte `code` der Tabelle `Schluessel` alphanumerisch zu sortieren.
- Spezielle Sortierung: Wenn `sortierNrVerwendet = ja`, so sind die Codes nach den Werten in der Spalte `sortierNr` der Tabelle `Schluessel` numerisch zu sortieren.

Beispiel:

Das Datenfeld „pT“ des Datensatzes 18/1<sup>14</sup> besitzt den Schlüssel `pTMamma`, für welchen die spezielle Sortierung (`sortierNrVerwendet = ja`) definiert ist (Tabelle 10).

Tabelle 10 Schlüssel mit spezieller Sortierung (pTMamma)

Kode	sortierNr
pT0	1
pTis	2
pT1mic	3
pT1a	4
pT1b	5
pT1c	6
pT2	7
pT3	8
pT4a	9
pT4b	10
pT4c	11
..	..

*Suchfunktion bei Schlüssel mit einer großen Anzahl von Codes*<sup>15</sup>

Bei Schlüssel mit einer großen Anzahl von Codes (z.B. Schlüssel `ICDO3Mamma` mit 138 Einträgen) soll eine anwenderfreundliche Möglichkeit zur Auswahl der passenden Codes bereitgestellt werden. Die Umsetzung als Auswahlliste (z.B. Combobox) führt zu erhöhtem Dokumentationsaufwand, falls der Anwender über Pfeiltasten oder Schiebepalken zum passenden Code navigieren muss. Ergänzend soll daher eine Suchfunktion realisiert werden, welche eine Suche über die Attribute `SchluesselWert.code` oder `Schluessel-`

---

<sup>14</sup> Version 12.0



Wert.bezeichnung ermöglicht. Die zu realisierenden Anwendungsfälle werden in den folgenden Beispiele erläutert.

Beispiel (Suche über Kode):

Der Anwender möchte beim Datenfeld „maligne Neoplasie“ (Schlüssel ICDO3Mamma, Modul 18/1) einen ICDO3-Kode eingeben, welcher mit der Ziffernfolge 8523 beginnt. Über ein geeignetes Suchfenster gelangt der Anwender zu einer Teilliste, welche die nachfolgend aufgelisteten Codes und die hinterlegten Bezeichnungen anzeigt:

- 8523/3 = invasives duktales Karzinom gemischt mit anderen Karzinom-Typen
- 8523/6 = invasives duktales Karzinom gemischt mit anderen Karzinom-Typen, Metastase
- 8523/9 = invasives duktales Karzinom gemischt mit anderen Karzinom-Typen, unbestimmt ob Primärtumor oder Metastase

Beispiel (Suche über Bezeichnung):

Der Anwender möchte beim Datenfeld „maligne Neoplasie“ (Schlüssel ICDO3Mamma, Modul 18/1) einen ICDO3-Kode eingeben, dessen Bezeichnung die Zeichenfolge *Adenokarzinom* enthält. Über ein geeignetes Suchfenster gelangt der Anwender zu einer Teilliste, welche die nachfolgend aufgelisteten Codes und die hinterlegten Bezeichnungen anzeigt:

- 8140/3 = Adenokarzinom o.n.A.
- 8140/6 = Adenokarzinom-Metastase o.n.A.
- 8140/9 = Adenokarzinom o.n.A., unbestimmt ob Primärtumor oder Metastase
- 8211/3 = Tubuläres Adenokarzinom
- 8211/6 = Tubuläres Adenokarzinom, Metastase
- 8211/9 = Tubuläres Adenokarzinom, unbestimmt ob Primärtumor oder Metastase
- 8290/3 = Oxyphiles Adenokarzinom
- 8290/6 = Oxyphiles Adenokarzinom, Metastase
- 8290/9 = Oxyphiles Adenokarzinom, unbestimmt ob Primärtumor oder Metastase
- 8401/3 = Apokrines Adenokarzinom
- 8401/6 = Apokrines Adenokarzinom, Metastase
- 8401/9 = Apokrines Adenokarzinom, unbestimmt ob Primärtumor oder Metastase
- 8410/3 = Talgdrüsenadenokarzinom
- 8410/6 = Talgdrüsenadenokarzinom, Metastase
- 8410/9 = Talgdrüsenadenokarzinom, unbestimmt ob Primärtumor oder Metastase
- 8480/3 = Muzinöses Adenokarzinom
- 8480/6 = Muzinöses Adenokarzinom, Metastase
- 8480/9 = Muzinöses Adenokarzinom, unbestimmt ob Primärtumor oder Metastase
- 8503/2 = Nichtinvasives intraduktales papilläres Adenokarzinom
- 8503/3 = Intraduktales papilläres Adenokarzinom mit Invasion
- 8503/6 = Intraduktales papilläres Adenokarzinom mit Invasion, Metastase

---

<sup>15</sup> Diese Anforderungen wurden auf mehrfachen Anwenderwunsch neu in die Spezifikation aufgenommen.

8503/9 = Intraduktales papilläres Adenokarzinom mit Invasion, unbestimmt ob Primärtumor oder Metastase

8572/3 = Adenokarzinom mit Spindelzellmetaplasie

8572/6 = Adenokarzinom mit Spindelzellmetaplasie, Metastase

8572/9 = Adenokarzinom mit Spindelzellmetaplasie, unbestimmt ob Primärtumor oder Metastase

## 2.1.9 Exklusionsschlüssel

Wenige Datenfelder des Datensatzes HCH und der Tx-Datensätze genügen einer besonderen Systematik:

Die Werte einiger Bogenfelder vom Feldtyp `GANZEZAHL` oder `ZAHL` werden bei bestimmten Wertkonstellationen in besonderer Weise (nichtnumerisch) interpretiert.

### Beispiel:

Im Bogenfeld `KOERPERGROESSE` im Teildatensatz `HCH:B` (Zeile 9) ist die Körpergröße in cm anzugeben. Im Dokumentationsbogen gibt es den Hinweis, dass bei unbekannter Körpergröße der Wert 999 anzugeben ist.

7 Körpergröße

cm

999 = unbekannt

Abbildung 3 Numerisches Datenfeld (Datensatz HCH) mit Exklusionsschlüssel: Die Angabe 999 ist keine Körpergröße.

Die über das Attribut `fkExklusionSchluessel` (Tabelle `Feld`) referenzierten Schlüsselcodes umfassen die Zahlenwerte, welche nicht numerisch interpretiert werden. Die Exklusionsschlüssel sind in der Eingabemaske darzustellen.

## 2.2 Überschriften

Die Überschriften der Dokumentationsbögen in die Spezifikation sind in der Tabelle `Ab-schnitt` zu finden.

Tabelle 11: Struktur der Tabelle `Abschnitt`

Feldname	Feldtyp	Bemerkung
<code>idAbschnitt</code>	INTEGER	Primärschlüssel

Feldname	Feldtyp	Bemerkung
bezeichnung	TEXT	Text der Überschrift
ebene	INTEGER	Zeigt die Hierarchie der Überschriften an
fkStartBogenFeld	INTEGER	Fremdschlüssel auf das erste zur Überschrift gehörende Bogenfeld
fkEndeBogenFeld	INTEGER	Fremdschlüssel auf das letzte zur Überschrift gehörende Bogenfeld

Zu jeder Überschrift ist angegeben, bei welchem Bogenfeld sie beginnt und bei welchem Bogenfeld sie endet. Über das Attribut *ebene* lassen sich auch Teilüberschriften realisieren. Ein Bogenfeld kann somit mehreren Überschriften zugeordnet sein.

Die in der Spezifikationsdatenbank hinterlegten Überschriften sind in die Eingabemasken der QS-Dokumentationssoftware zu integrieren.

Achtung:

Viele Datenfelder sind für den Anwender erst im Kontext der Überschriften verständlich.

### 2.3 Ausfüllhinweise

Die Ausfüllhinweise zu den Datenfeldern sind in einem separaten ZIP-Archiv enthalten, welches nach folgendem Schema benannt ist:

Ausfuellhinweise-<Version>.zip (beim Hauptrelease)

bzw.

Ausfuellhinweise-<Version>\_SR<n>.zip (bei Service Releases)

Jeder Ausfüllhinweis ist ein HTML-Dokument.

Beispiel:

Ausfüllhinweis ENTLGRUND.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>ENTLGRUND</title>
  <link href="feldhinweis.css" type="text/css" rel="stylesheet">
</head>
<body>
<!--BLOCKANFANG-->
<div class="AH">
<p>Der Entlassungsgrund 09 = Entlassung in eine Rehabilitationseinrichtung ist auch
dann anzugeben, wenn der Patient zwischenzeitlich kurzfristig nach Hause entlassen
wird und seine Rehabilitationsmaßnahmen von dort aus antritt.</p>
</div>
<!--BLOCKENDE-->
</body>
</html>
```

In der Spalte `aHinweis` der Tabelle `BogenFeld` ist festgelegt, welcher HTML-Ausfüllhinweis mit einem Datenfeld verknüpft ist:

`<aHinweis>.htm` = Name der HTML-Datei

**Beispiel:**

Das Bogenfeld 1571 (Version 9.0) hat in der Spalte `aHinweis` den Eintrag „*OPSCHLUESSEL*“. Der zugeordnete Ausfüllhinweis des ZIP-Archives heißt *OPSCHLUESSEL.htm*.

Wenn der Eintrag in `aHinweis` leer ist, so existiert für das betreffende Bogenfeld kein Ausfüllhinweis.

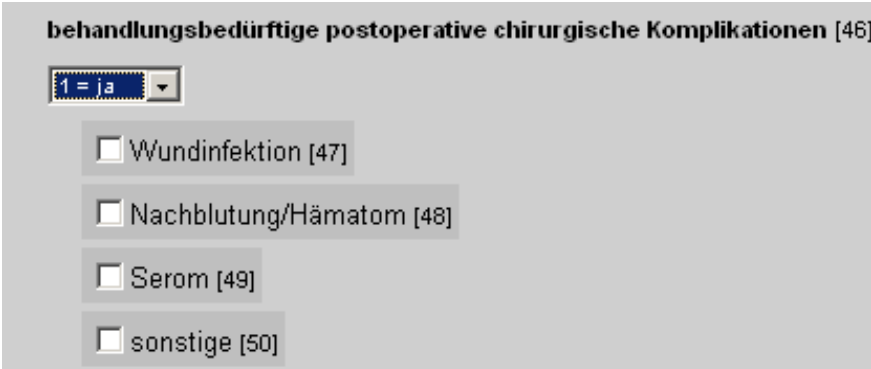
Die Zuordnung von Bogenfeldern und Ausfüllhinweisen ist auch in der Abfrage *Ausfüllhinweise* dargestellt. Sie zeigt Modul/Teildatensatz, Zeile, Feldname, Bezeichnung und den HTML-Dateinamen des Ausfüllhinweises zu dem Bogenfeld. Im Gegensatz zur Tabelle `Bogenfeld` ist hier die Endung `.htm` mit angegeben.

## 2.4 Verwendung der Datenfeldbeschreibung für die Gestaltung von Eingabemas- ken

Die Eingabemaske einer Erfassungssoftware (graphical user interface = GUI) soll ergonomisch und anwenderfreundlich gestaltet sein. Gestaltung und Layout des GUI sind Aufgabe der Softwarehersteller. Neben Anforderungen der Kunden werden üblicherweise firmeninterne Standards bzw. Vorgaben des Betriebssystems (z.B. Windows) für das „look and feel“ berücksichtigt.

### Umsetzungsbeispiel:

Abbildung 4 zeigt eine mögliche Umsetzung der Datenfelder unter der Überschrift „Komplikationen“ des Datensatzes 18/1 (Version 10.0) in einem webbasierten Produkt.<sup>16</sup>



**behandlungsbedürftige postoperative chirurgische Komplikationen [46]**

1 = ja ▾

Wundinfektion [47]

Nachblutung/Hämatom [48]

Serom [49]

sonstige [50]

Abbildung 4 Beispiel für webbasierte Oberfläche

Die Datenfelder 47 (Wundinfektion) bis 50 (sonstige) haben den Schlüssel `Jleer` und sind im HTML-Beispiel als Eingabefelder vom Typ Checkbox realisiert, während das Datenfeld 46 ein Optionsfeld ist.

Diese Spezifikation definiert als *Minimalstandard die für den Anwender sichtbaren Inhalte* der Dokumentationsformulare. Als Referenz für die sichtbaren Inhalte dienen die Dokumentationsbögen, welche zusammen mit der Spezifikation von der BQS veröffentlicht wer-

<sup>16</sup> Dieses Beispiel ist nicht als verbindliche Vorgabe für die Umsetzung zu verstehen, sondern dient der Veranschaulichung für den Leser. Insbesondere sei hier explizit darauf hingewiesen, dass ein Web-Frontend lediglich eine unter vielen Umsetzungsvarianten ist, welche von der BQS nicht präferiert wird.

den. Die Dokumentationsbögen werden als PDF-Dokumente bereitgestellt, welche aus der Spezifikationsdatenbank automatisch generiert worden sind.

Tabelle 12 gibt einen Überblick darüber, welche Informationen der Spezifikationsdatenbank (identifiziert durch Tabelle und Attribut) bei der Erstellung der Dokumentationsbögen berücksichtigt werden und somit auch in den Erfassungssystemen sichtbar sein sollen.

Tabelle 12 Informationen aus der Datenbank, welche im GUI verwendet werden

Tabelle	Attribut	Bemerkung	sichtbar für Anwender
Modul	name	Kürzel des Datensatzes (z.B. 18/1) , erscheint üblicherweise im Titel des Formulars	✓
Modul	bezeichnung	Bezeichnung des Datensatzes (z.B. Mammachirurgie), erscheint üblicherweise im Titel des Formulars	✓
Bogen	bezeichnung	Bezeichnung des Teildatensatzes	✓
BogenFeld	zeileAufBogen	Nummer des Datenfeldes. Die BQS empfiehlt zur besseren Orientierung, die Zeilennummer in den Eingabemasken anzuzeigen.	(✓)
BogenFeld	bezeichnung	Bezeichnung des Datenfeldes	✓
BogenFeld	ergaenzendeBezeichnung	Ergänzende Bezeichnung zum Datenfeld, kann z.B. durch Wahl der Schrift von der Bezeichnung abgesetzt werden.	✓
Feld	laenge	definiert die Länge des Eingabefeldes. Für die Gestaltung des Eingabefeldes sind weitere Informationen aus der Datenbank wichtig (z.B. Feld.nachKommaLaenge)	
Feld	einheit	Einheiten (wie z.B. ml) müssen angezeigt werden	✓
Feld	formatAnweisung	Formatanweisungen (wie z.B. TT.MM.JJJJ) sollen - sofern nicht durch übergeeignete Eingabefelder unterstützt – für den Anwender angezeigt werden.	(✓)
Feld	fkExklusionSchluessel	Die Exklusionsschlüssel (Kode und Bezeichnung) eines numerischen Datenfeldes sind anzuzeigen (Kapitel 2.1.9).	✓
SchlüsselWert	code	Bei Schlüsselfeldern sollen die Codes möglichst in Auswahllisten angezeigt werden. Bei einigen Realisierungsvarianten (z.B. Checkbox) kann auf die Anzeige der Codes verzichtet werden.	(✓)
SchlüsselWert	bezeichnung	Bei Schlüsselfeldern müssen die Textdefinitionen der	✓

		Kodes (z.B. in einer Auswahlliste) angezeigt werden. Für die Sortierung sind die Attribute <code>sortierNrVerwendet</code> und <code>zahl</code> der Tabelle <code>Schluesse1</code> relevant (Abschnitt 2.1.8).	
Abschnitt	bezeichnung	Die Überschriften sind wichtig für die Strukturierung und das Verständnis des Datensatzes und müssen deshalb in der QS-Dokumentationssoftware angezeigt werden.	✓

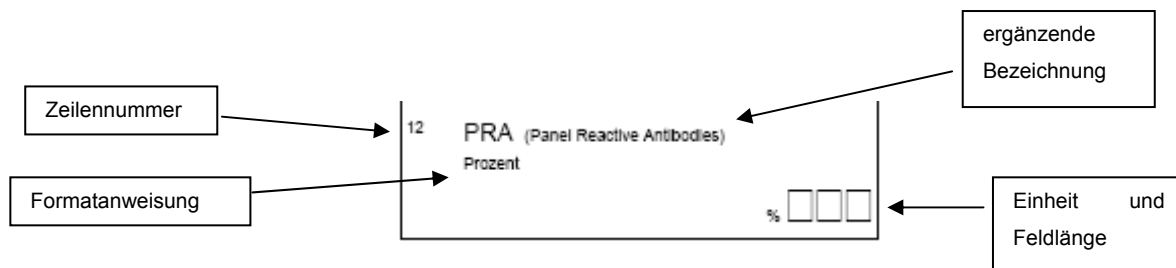


Abbildung 5 Beispiel für Informationen, die in der Oberfläche angezeigt werden sollen

Abbildung 5 zeigt für ein Datenfeld eines Dokumentationsbogens (PDF) den Zusammenhang zu den Informationen, welche in der Datenbank vorhanden sind.

Werden Datenfelder (z.B. OPS) eines BQS-Datensatzes aus Fremdsystemen<sup>17</sup> über Schnittstellen importiert, so sollen die übernommenen Daten auch in der Erfassungsoftware angezeigt werden. Es ist für den Anwender wichtig, die vollständigen QS-Daten im Kontext eines „BQS-Formulars“ zu sehen und auch auf Richtigkeit und Vollständigkeit zu prüfen.

#### *Umrechnung von Einheiten bei numerischen Feldern*

In Einzelfällen ist es aus Anwendersicht hilfreich, wenn die Erfassungsmaske die Eingabe von Messwerten in Einheiten ermöglicht, welche von den spezifizierten Einheiten abweichen. Beispielsweise soll laut Spezifikation das Datenfelder `DLDAUER` in Minuten mit einer Nachkommastelle (z.B. Modul 09/1) dokumentiert werden. Der Ausfüllhinweis stellt eine Umrechnungstabelle bereit, falls die Angaben in Minuten und Sekunden vorliegen. Diese

<sup>17</sup> Die BQS empfiehlt, möglichst viele Daten aus Fremdsystemen zu übernehmen, um Mehrfachdokumentation zu vermeiden.

Funktionalität sollte möglichst in die Erfassungssoftware integriert werden, um den Dokumentationsaufwand zu verringern.

### 3 Plausibilitätsprüfungen

Fehlende und widersprüchliche Angaben in den Datensätzen sollen durch umfangreiche Plausibilitätsprüfungen verhindert werden.

In der *QS-Dokumentationssoftware* muss die vollständige Plausibilitätsprüfung für jeden Datensatz *spätestens* bei Dokumentationsabschluss erfolgen. Teile der Plausibilitätsprüfungen sollen bereits während der Erfassung erfolgen. Dadurch wird sichergestellt, dass ein aufwändiges Korrekturverfahren (Übermitteln der Datensätze an die datenentgegennehmende Stelle, dort Prüfung, ggf. Übermitteln eines Fehlerberichts an das dokumentierende Krankenhaus, Korrektur der Dokumentation und erneutes Übermitteln des Datensatzes) weitgehend entfällt. Die *datenentgegennehmenden Stellen* führen für jeden Datensatz alle harten Plausibilitätsprüfungen der BQS-Spezifikation durch. Bei einer Regelverletzung ist der Datensatz zurückzuweisen<sup>18</sup>.

*Es gelten folgende Grundsätze der Plausibilitätsprüfungen:*

- Alle Felder müssen ausgefüllt sein, wenn dem nicht andere logische Sachverhalte entgegenstehen.
- Jedes Feld, das auszufüllen ist, muss einen sinnvollen Feldinhalt haben.
- Es wird jede harte Plausibilitätsprüfung vorgenommen, die definiert ist.
- Harte Plausibilitätsprüfungen werden nur vorgenommen, wenn Sachverhalte zwingend miteinander gekoppelt sind.
- Es werden keine Sachverhalte suggeriert (keine Default-Werte, keine Vorbelegungen, keine Profile; Fehlermeldungen werden vorgegeben).
- Keine Angabe (bzw. kein Feldinhalt) wird ergänzt oder gelöscht.

#### 3.1 Arten der Plausibilitätsprüfungen

---

<sup>18</sup> Achtung: Die datenentgegennehmenden Stellen dürfen keine zusätzlichen (in der Spezifikation nicht definierte) Plausibilitätsprüfungen durchführen.



Es wird zwischen drei Arten von Plausibilitätsprüfungen unterschieden:

- harte Prüfungen
- weiche Prüfungen in der QS-Dokumentationssoftware
- warnende Prüfungen bei der Datenentgegennahme

Die drei Arten der Prüfungen werden in unterschiedlichen Kontexten (QS-Software bzw. Datenentgegennahme) durchgeführt und haben unterschiedliche Konsequenzen. Tabelle 13 gibt einen Überblick:

Tabelle 13: Arten der Plausibilitätsprüfungen

Art der Prüfung	Kürzel	Prüfung durch QS-Software	Prüfung durch Datenstelle	Konsequenz: verhindert Dokumentationsabschluss oder Datenentgegennahme
hart	H	ja	ja	ja
weich	W	ja	nein	nein
warnend	D	optional	ja	nein

### 3.1.1 Harte Prüfungen

Harte Prüfungen sind sowohl in der QS-Dokumentationssoftware als auch bei der Datenentgegennahme anzuwenden. Bei einer harten Regelverletzung ist

- ein Dokumentationsabschluss der QS-Dokumentation eines Vorganges unzulässig oder
- ein Datensatz von der entgegennehmenden Stelle zurückzuweisen.

Die in der technischen Dokumentation und der Datenbank definierten Plausibilitätsprüfungen sind hart, außer wenn sie explizit als weich oder warnend gekennzeichnet sind.

### 3.1.2 Weiche Plausibilitätsprüfungen

Die weichen Plausibilitätsprüfungen der BQS-Spezifikation sind von der QS-Dokumentationssoftware bis spätestens zum Dokumentationsabschluss durchzuführen. Bei einer Regelverletzung erhält der Benutzer einen Warnhinweis, anhand dessen er entscheidet, ob eine Änderung von Feldinhalten notwendig ist.

### 3.1.3 Warnende Plausibilitätsprüfungen

Unterjährig (in Form von Service-Releases während des Erfassungsjahres) findet keine Verschärfung der harten Plausibilitätsprüfungen statt, um den Anwendern in den Krankenhäusern den Dokumentationsabschluss nach transparenten Regeln zu ermöglichen.

Trotz einer Vielzahl harter Plausibilitätsprüfungen kann es aber dennoch vorkommen, dass Krankenhäuser inhaltlich unplausible Datensätze liefern. Fallen derartige Lücken in der Datenqualität nach Start des Erfassungsjahres auf, so kann die im Krankenhaus installierte QS-Dokumentationssoftware den Anwender nicht auf diesen Mangel hinweisen.

Warnende Plausibilitätsprüfungen können, z.B. auf Anregung einer Landesgeschäftsstelle, von der BQS unterjährig in den Service-Releases publiziert werden, um die Krankenhäuser auf gravierende Mängel in der Datenqualität hinzuweisen<sup>19</sup>:

- Datenstellen setzen die warnenden Plausibilitätsprüfungen um.
- Krankenhäuser erhalten über das Fehlerprotokoll Informationen über die mangelhaften Datensätze.
- Die QS-Dokumentationssoftware zeigt die Warnungen den Anwendern nach der Rückprotokollierung an.
- Anwender haben dann die Möglichkeit, die mit Warnungen versehenen Datensätze zu korrigieren und erneut einzusenden.

## 3.2 Feldbezogene Prüfungen

Feldbezogene Prüfungen sind Plausibilitätsregeln, welche sich auf ein einziges Datenfeld beziehen.

Einige feldbezogene Prüfungen - insbesondere die Wertebereichsüberprüfungen - können auch in der formalen Regelsyntax formuliert werden (Beispiel: `OPDAUER > 600`), welche erst in nachfolgenden Abschnitten behandelt wird (ab Kapitel 3.3).

Die in diesem Abschnitt behandelten Prüfungen ergeben sich direkt aus der Datenfeldbeschreibung (Tabellen `Feld` und `BogenFeld`).

---

<sup>19</sup> Warnende Plausibilitätsprüfungen sind somit als **ernste Warnungen** zu verstehen!

Die QS-Dokumentationssoftware muss sowohl die harten als auch die weichen feldbezogenen Prüfungen ausführen. Die Evaluation soll direkt bei Dateneingabe geschehen. Fehleingaben sollen dem Benutzer direkt mitgeteilt werden. Einige Prüfungen erübrigen sich durch adäquate Gestaltung von Eingabemasken, z.B. durch Bereitstellung von Auswahlmenüs für Schlüsselcodes. Bei Regelverletzung muss die QS-Dokumentationssoftware dem Benutzer verständliche Fehlertexte anzeigen.

Bei der Datenentgegennahme sind alle harten feldbezogenen Prüfungen zu evaluieren und bei Regelverletzung die unten definierten standardisierten Fehlertexte im Rahmen des Korrekturverfahrens (Kapitel 6.6.2) an das Krankenhaus zu übermitteln.

Die nachfolgend beschriebenen feldbezogenen Prüfungen werden vor Evaluation der in Kapitel 3.3 beschriebenen feldübergreifenden Regeln durchgeführt.

### 3.2.1 Überprüfung des Formats

Die Formatprüfung bezieht sich auf den Exportdatensatz: Die QS-Dokumentationssoftware muss Daten im korrekten Format generieren, die datenentgegennehmenden Stellen prüfen auf Korrektheit des Formats.

Die Prüfung bezieht sich insbesondere darauf, ob der Feldinhalt dem in der BQS-Spezifikation definierten Basistyp entspricht. Z.B. sind Buchstaben beim Basistyp GANZE-ZAHL nicht erlaubt. Die weiteren Details zu den notwendigen Formatprüfungen finden sich im Abschnitt 6.4.2.

#### *Standardisierter Fehlertext für Formatfehler*

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<BogenFeld.name> "<BogenFeld.bezeichnung>" (Zeile <BogenFeld.zeileAufBogen>) ist kein gültiger <BasisTyp.name>-Wert (<BasisTyp.bezeichnung> <BasisTyp.format>).

#### Beispiel:

Der Wert '3A.06.2005' des Datenfeldes 01/1:B:AUFNDATUM "Aufnahmedatum Krankenhaus" (Zeile 6) ist kein gültiger DATUM-Wert (Zehnstelliges Datum TT.MM.JJJJ).

### 3.2.2 Überprüfung der Feldlänge

Überschreitet die Anzahl der Zeichen eines Wertes die spezifizierte Länge<sup>20</sup> (Attribut `laenge` in Tabelle `Feld`) des Feldes?

#### *Standardisierter Fehlertext für Längenfehler*

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<BogenFeld.name> "<BogenFeld.bezeichnung>" (Zeile <BogenFeld.zeileAufBogen>) überschreitet die zulässige Feldlänge <Feld.laenge>.

#### Beispiel:

Der Wert '31.06.20040' des Datenfeldes 01/1:B:AUFNDATUM "Aufnahmedatum Krankenhaus" (Zeile 6) überschreitet die zulässige Feldlänge 10.

### 3.2.3 Überprüfung der Schlüsselcodes

Werden bei Schlüsselfeldern nur zulässige Schlüsselcodes verwendet?

#### *Standardisierter Fehlertext bei unzulässigen Schlüsselcodes*

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<BogenFeld.name> "<BogenFeld.bezeichnung>" (Zeile <BogenFeld.zeileAufBogen>) ist kein zulässiger Code des Schlüssels <Schlüssel.name> (<Schlüssel.bezeichnung>).

#### Beispiel:

Der Wert 'xyz' des Datenfeldes 01/1:O:OPSCHLUESSEL "Operation OPS-" (Zeile 23) ist kein zulässiger Code des Schlüssels OPS301.

#### Besonderheiten bei externen Schlüsseln:

- Bei externen DIMDI-Schlüsseln (ICD-10-GM oder OPS) sind die jeweils gültigen amtlichen Kataloge zu verwenden. Alle Codes eines Behandlungsfalls müssen in derjenigen Katalogversion dokumentiert sein, welche am Aufnahmetag Krankenhaus des Patienten bzw. am Follow-Up-Erhebungsdatum bei den Datensätzen HTXFU, NTXFU, LTXFU, NLSFU und LLSFU gültig ist<sup>21</sup>.
- Nicht-terminale ICD- oder OPS-Kodes sind unzulässig!

<sup>20</sup> Wenn bei einem Export- oder Zusatzfeld die Länge nicht spezifiziert ist, so entfällt die Prüfung.

<sup>21</sup> Analog zur Regelung zur Abgrenzung von Verfahrensjahren in Kapitel 5.3.

- Fehlen bei OPS-Kodes Seitenlokalisationen, obwohl diese erforderlich sind, so ist der OPS-Kode fehlerhaft (siehe auch Kapitel 4.1).

### 3.2.4 Überprüfung numerischer Wertebereiche

Bei numerischen Datenfeldern (Typ ZAHL oder GANZEZAHL) ist *hart* zu überprüfen, ob der Zahlenwert im durch die Attribute `min` und `max` (Tabelle FELD) definierten Wertebereich liegt:

`wert < min` (nur prüfen, wenn `min`  $\neq$  LEER)

`wert > max` (nur prüfen, wenn `max`  $\neq$  LEER)

In der Tabelle `Feld` sind weitere Unter- und Obergrenzen (Attribute `minWeich` und `maxWeich`) für *weiche* Prüfungen definiert. In wenigen Fällen sind auch in der Tabelle `BogenFeld` weiche Grenzen definiert. Falls vorhanden (= *not null*) werden die in der Tabelle `BogenFeld` gesetzten speziellen weichen Grenzen statt der weichen Grenzen in der Tabelle `Feld` angewandt.

#### Beispiel:

Tabelle 14 zeigt für das Feld `OPDAUER`<sup>22</sup> die Wertebereichsdefinitionen auf Feld- und Bogenfeldebene für die Module 09/1, 16/1 und PNTX.

Tabelle 14 Beispiel für Wertebereichsgrenzen (Datenfeld `OPDAUER`)<sup>23</sup>

Modul	Feld.min	Feld.max	Feld.minWeich	BogenFeld.minWeich	Feld.maxWeich	BogenFeld.maxWeich
09/1	<b>1</b>	undefiniert	undefiniert	<b>5</b>	<b>240</b>	undefiniert
16/1	<b>1</b>	undefiniert	undefiniert	<b>10</b>	240	<b>90</b>
PNTX	<b>1</b>	undefiniert	undefiniert	undefiniert	<b>240</b>	undefiniert

Für alle Module ist folgende harte Prüfung anzuwenden:

`OPDAUER < 1` (hart)

Im Modul 09/1 sind folgende weiche Prüfungen anzuwenden:

`OPDAUER < 5` (weich)

`OPDAUER > 240` (weich)

Im Modul 16/1 sind folgende weiche Prüfungen anzuwenden:

<sup>22</sup> Beispiel aus BQS-Spezifikation 12.0

<sup>23</sup> Die fett hervorgehobenen Einträge sind gültige Wertebereichsgrenzen

OPDAUER < 10 (weich)

OPDAUER > 90 (weich)

Im Modul PNTX ist folgende weiche Prüfung anzuwenden:

OPDAUER > 240 (weich)

Eine Übersicht über die in numerischen Datenfeldern definierten harten und weichen Wertebereiche bietet die Abfrage *WertebereicheNumerischerFelder*.

Außerdem sind die Wertebereiche in den Ausfüllhinweisen angegeben. Z. B. für das Datenfeld OPDAUER in 09/1:

- Gültige Angabe:  $\geq 1$  Minuten
- Angabe ohne Warnung: 5 - 240 Minuten

#### Achtung:

Wenn für ein numerisches Datenfeld Exklusionsschlüssel (Abschnitt 2.1.9) definiert sind, so sind die Prüfungen so zu modifizieren, dass die Codes des Exklusionsschlüssels nicht als numerische Werte behandelt werden.

#### Beispiel:

Das Datenfeld HCH:B:KOERPERGROESSE hat die weiche Obergrenze 230, aber den Exklusionskodes 999 = unbekannt.

Die anzuwendende Plausibilitätsprüfung lautet:

KOERPERGROESSE > 230 UND KOERPERGROESSE <> 999

#### *Standardisierter Fehlertext bei Unterschreitung einer Wertebereichsgrenze*

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<BogenFeld.name> "<BogenFeld.bezeichnung>" (Zeile <BogenFeld.zeileAufBogen>) ist kleiner als '<Feld.min>'.

#### Beispiel:

Der Wert '-90' des Datenfeldes 17/5:B:POSTOPEXFLEXKNIE1 "Extension/Flexion 1" (Zeile 42,1) ist kleiner als '0'.

#### *Standardisierter Fehlertext bei Überschreitung einer Wertebereichsgrenze*

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<BogenFeld.name> "<BogenFeld.bezeichnung>" (Zeile <BogenFeld.zeileAufBogen>) ist größer als '<Feld.max>'.

Beispiel:

Der Wert '370' des Datenfeldes 17/5:B:POSTOPEXFLEXKNIE1 "Extension/Flexion 1" (Zeile 42,1) ist größer als '10'.

Bei weichen Plausibilitätsverletzungen ist dem Fehlertext das Wort „Hinweis:“ voranzustellen.

### 3.2.5 Überprüfung der Muss-Felder

#### 3.2.5.1 Muss-Felder des Datensatzes

Ein nicht ausgefülltes Muss-Feld (Kapitel 2.1.4) führt zu einer Regelverletzung.

*Standardisierter Fehlertext für Muss-Fehler*

Das Datenfeld <Modul.name>:<Bogen.name>:<BogenFeld.name> "<BogenFeld.bezeichnung>" (Zeile <BogenFeld.zeileAufBogen>) muss einen gültigen Wert enthalten.

Beispiel:

Das Datenfeld 01/1:B:AUFNDATUM "Aufnahmedatum Krankenhaus" (Zeile 6) muss einen gültigen Wert enthalten.

Achtung:

Wenn ein Listenfeld als Muss-Feld deklariert ist, so ist nur das erste Exportfeld der Liste ein Muss-Feld, die restlichen Elemente sind dann Kann-Felder. Wenn ein Listenfeld als Kann-Feld deklariert ist, so sind alle exportierten Elemente ebenfalls Kann-Felder.

#### 3.2.5.2 Muss-Felder des Exportdatensatzes

Verbindlich für die Muss/Kann-Prüfung ist die Definition in der Tabelle `BogenFeld`. Die Muss/Kann-Zuordnungen im Exportdatensatz werden hieraus abgeleitet:

- 1) Für Exportfelder, welche nicht pseudonymisiert werden und keine Listenfelder sind, entspricht die Muss/Kann-Zuordnung der Definitionen in der Tabelle `BogenFeld`.

- 2) Die Muss/Kann-Zuordnung der anonymisierten Datenfelder (Ersatzfelder) ergibt sich logisch aus der Berechnungsformel (Attribut `formel` in Tabelle `ErsatzFeld`). Z.B. ist ein Ersatzfeld ein Muss-Feld, wenn alle an der Berechnung beteiligten Bogenfelder Muss-Felder sind.
- 3) Bei Muss-Listefeldern der Tabelle `BogenFeld` ist grundsätzlich nur das erste Element (`FELD_1`) ein Muss-Feld, die weiteren Elementfelder (`FELD_2`, `FELD_3` etc.) sind Kann-Felder.
- 4) Zusatzfelder der Tabelle `ExportFormat` sind Muss-Felder, außer wenn sie Schlüsselfelder mit einem `Jleer`-Schlüssel (z.B. das Zusatzfeld `Storno`) sind.

Als inoffizielle Hilfestellung für datenentgegennehmende Stellen bei der Umsetzung wird das Attribut `fkMussKann` in der Tabelle `ExportFormat` eingeführt, deren Inhalte automatisch generiert werden. Allerdings funktioniert nach dem gegenwärtigen Stand diese automatische Zuordnung nur für die Fälle 1, 3 und 4. Für den Fall 2 bleibt die `fkMussKann`-Zuordnung leer. Datenentgegennehmende Stellen müssen trotzdem entsprechende Muss-Prüfungen implementieren.

#### Beispiel:

Das Feld `entlquartal` berechnet sich über die Formel `quartal(ENTLDATEUM)`. In Modulen, wo das Datenfeld `ENTLDATEUM` ein Muss-Feld ist, ist auch das Ersatzfeld ein Muss-Feld und die datenentgegennehmende Stelle ist verpflichtet, die Muss-Prüfung hier auch durchführen. Ansonsten ist das Ersatzfeld ein Kann-Feld. Achtung: Das Nichtausführen der erforderlichen Muss-Prüfungen kann gravierende Folgen für die Auswertung haben.

Für Exportfelder (Tabelle `ExportFormat`), welche einen Bezug zu einem Ersatzfeld (Tabelle `ErsatzFeld`) bzw. zu einem Zusatzfeld (Tabelle `ZusatzFeld`) haben, sind die standardisierten Fehlertexte anzupassen:

#### *Standardisierter Fehlertext für Muss-Fehler eines Ersatzfeldes*

Das Ersatzfeld des Datenfeldes `<Modul.name>:<Bogen.name>:<BogenFeld.name>` "`<BogenFeld.bezeichnung>`" (Zeile `<BogenFeld.zeileAufBogen>`) muss einen gültigen Wert enthalten.

#### *Standardisierter Fehlertext für Muss-Fehler eines Zusatzfeldes*



Das Zusatzfeld `<Modul.name>:<Bogen.name>:<ZusatzFeld.name>` "`<ZusatzFeld.bezeichnung>`" muss einen gültigen Wert enthalten.

### 3.3 Feldübergreifende Regeln

Die Regeln ..

- haben eine eigene Syntax.
- haben geringe Komplexität.
- haben einfache, dem Anwender verständliche Fehlertexte.
- enthalten alle Teilregeln der Feldgruppen.
- haben gewöhnlich den Bezug zu zwei oder mehreren Feldern.
- können zum Teil direkt nach der Benutzereingabe in ein Feld geprüft werden.
- enthalten Bedingungen für unplausible Angaben<sup>24</sup>.

Um die Regeln von den später definierten Feldgruppenregeln (Kapitel 3.7) zu unterscheiden, werden sie auch Einzelregeln genannt.

#### 3.3.1 Die Regeltabelle

Die Bedingungen für unplausible Angaben sind in der Tabelle `Regeln` abgelegt. Die Syntax ist in Kapitel 3.4 beschrieben. Die Bedingungen sind möglichst kurz gefasst (Vermeidung von durch `ODER` verknüpften Teilbedingungen). Jede Bedingung kommt nur einmal innerhalb eines Moduls vor.

Tabelle 15: Struktur der Tabelle `Regeln`

Feldname	Feldtyp	Bemerkung
<code>idRegeln</code>	INTEGER	Primärschlüssel
<code>fkModul</code>	INTEGER	Fremdschlüssel zur Tabelle <code>Modul</code>
<code>bedingung</code>	MEMO	entsprechend der Syntax definierte Regeln
<code>meldung</code>	MEMO	Fehlermeldung: Diese Texte sind bei Regeln mit Bezug zu Feldgruppen automatisch generiert.
<code>alternativMeldung</code>	MEMO	Alternative Fehlermeldung: Wenn hier ein Text vorhanden ist, so ist dieser anstelle des Textes in der Spalte <code>meldung</code>

<sup>24</sup> Eine Plausibilitätsregel müsste eigentlich "Unplausibilitätsregel" heißen, weil sie unplausible Zustände beschreibt, die zu Fehlermeldungen führen.

Feldname	Feldtyp	Bemerkung
fkMehrfachRegel	INTEGER	zu verwenden. Fremdschlüssel zur Tabelle mit mehrfach vorkommenden Regeln
fkFeldGruppe	INTEGER	Optionaler Fremdschlüssel zur Tabelle <code>FeldGruppe</code> : Indikator dafür, ob eine Regel aus einer Feldgruppe generiert wurde.
fkRegelTyp	TEXT(1)	Fremdschlüssel zur Tabelle <code>RegelTyp</code> : Die Regeltypen sind die in Kapitel 3.1 beschriebenen Arten der Plausibilitätsprüfungen: H, W oder D
gueltigNachExport	BIT	Regeln, welche den Wert nein haben, können von datenentgegennehmenden Stellen nicht evaluiert werden. Stattdessen werden die referenzierten Ersatzbedingungen der Tabelle <code>MehrfachRegel</code> evaluiert (falls definiert).

### 3.3.2 Bogenfelder einer Regel

Die Tabelle `RegelFelder` (Tabelle 16) ist eine Verknüpfungstabelle zwischen den Tabellen `Regeln` und `BogenFeld`. Durch geeignete Abfragen kann man unter Verwendung dieser Tabelle Folgendes herausfinden:

- 1) Bogenfelder, welche in einer Regel verwendet werden
- 2) Regeln, welche sich auf ein Bogenfeld beziehen.

Die Tabelle `RegelFelder` wird automatisch aus den Tabellen `Regeln` und `BogenFeld` generiert.

Tabelle 16: Struktur der Tabelle `RegelFelder`

Feldname	Feldtyp	Bemerkung
fkBogenFeld	INTEGER	Fremdschlüssel zu den Tabellen <code>Feld</code> und <code>Regeln</code> , bilden zusammen
fkRegeln	INTEGER	den Primärschlüssel

### 3.3.3 Mehrfach vorkommende Regeln

Wenn in einer Regel von der Anonymisierung betroffene Datenfelder benutzt werden, so kann diese von datenentgegennehmenden Stellen nicht evaluiert werden. Statt dessen wird für solche Regeln aus der Tabelle `Regeln` über einen Bezug zur Tabelle `MehrfachRegel` eine Ersatzbedingung definiert, welche von den datenentgegennehmenden Stellen zu evaluieren ist.

Tabelle 17: Struktur der Tabelle `MehrfachRegel`

Feldname	Feldtyp	Bemerkung
<code>idMehrfachRegel</code>	INTEGER	Primärschlüssel
<code>bedingung</code>	MEMO	entsprechend der Syntax definierte Regeln
<code>meldung</code>	MEMO	Kontextunabhängige Fehlermeldung
<code>ersatzBedingung</code>	MEMO	Ersatzregel für den anonymisierten Datensatz
<code>fkRegelTyp</code>	TEXT(1)	Fremdschlüssel zur Tabelle <code>RegelTyp</code> : Die Regeltypen sind die in Kapitel 3.1 beschriebenen Arten der Plausibilitätsprüfungen: H, W oder D

Beispiel:

Die Datenfelder der Regel 90<sup>25</sup> (`OPDATUM > ENTLDATUM`) werden nicht an die datenentgegennehmenden Stellen übermittelt. Bei der Entgegennahme ist die in der Tabelle `MehrfachRegel` definierte Ersatzregel (`idMehrfachRegel = 133`) anzuwenden:  
`poopvwdauer < 0.`

### 3.3.4 Weitere Regeln

Weitere feldübergreifende Regeln sind die in Kapitel 2.1.3 beschriebenen Existenzbedingungen für das Anlegen von abhängigen Teildatensätzen (Attribut `existenzBedingung` in Tabelle `Bogen`).

## 3.4 Regelsyntax

Bedingungen sind in den Tabellen `Regeln`, `MehrfachRegel` und `Bogen` definiert. Die den Bedingungen zugrunde liegende Regelsyntax ist in diesem Abschnitt beschrieben.

Jede Regel ist ein logischer Ausdruck, welcher das Ergebnis WAHR oder FALSCH hat. Jede Regel bezieht sich auf einen eingegebenen Datensatz eines Moduls, dessen Daten in Variablen, den Feldern, gespeichert sind.

---

<sup>25</sup> Beispiel aus BQS-Spezifikation 12.0

Die Regelsyntax lehnt sich an die logischen Ausdrücke in bekannten Programmiersprachen an. Jedoch haben die Operatoren deutsche Namen, z.B. UND statt AND oder ODER statt OR.

*Typen:*

Die möglichen Typen der Datenfelder sind in Tabelle 18 aufgelistet.

Tabelle 18: Datentypen der Datenfelder in den Plausibilitätsregeln

Datentyp	Bezeichnung	Beispiele (Literele)
BOOL	Boolsche Variable:	WAHR, FALSCH
TEXT	Zeichenkette (String)	"Spezifikation"
GANZEZAHL <sup>26</sup>	.. -2, -1, 0, 1, 2, 3, ..	1
ZAHL	Zahl (mit oder ohne Nachkommastellen)	25,4 oder -100,8
DATUM	Zehnstelliges Datum	'01.01.2006'
MONDATUM	Monatsdatum	'04.2006'
QUARTDATUM	Quartalsdatum	'3/2006'
JAHRDATUM	Jahresdatum	'2006'
NUMSCHLUESSEL	Numerisch kodierter Schlüssel (wie GANZEZAHL)	0
SCHLUESSEL	Alphanumerischer Schlüssel	'12.1'
UHRZEIT	Uhrzeit	'10:15'

Die BQS-Spezifikation unterscheidet zwischen NUMSCHLUESSEL und SCHLUESSEL:

- Die meisten Schlüsselwerte werden wie der Basistyp GANZEZAHL kodiert. D.h., dass die Codes dann nicht in Hochkommata gesetzt werden dürfen.
- Die OPS-Schlüssel (z.B. '5-282.0') oder die ICD-10-GM-Schlüssel haben dagegen den Basistyp SCHLUESSEL.

Achtung:

Jedes Datum (Datum, Jahresdatum, Monatsdatum, Quartalsdatum) muss in Hochkommata gesetzt werden.

---

<sup>26</sup> Beim Typ GANZEZAHL sind auch negative ganze Zahlen erlaubt.

### *Felder*

Feldnamen bestehen aus maximal 32 Zeichen und dürfen nur die Buchstaben A bis Z (Großbuchstaben) und die Ziffern 0 bis 9 enthalten. Ein Feldname muss immer mit einem Buchstaben beginnen. Umlaute und Sonderzeichen sind in Feldnamen nicht erlaubt. Ein Feldname darf auch nicht ein reserviertes Wort sein (z.B. LEER).

### Achtung:

In einer Regel dürfen nur die Feldnamen der im betreffenden Modul definierten Bogenfelder<sup>27</sup> enthalten sein. Bei der Evaluierung von Regeln werden die aktuellen Werte der referenzierten Bogenfelder eingesetzt. Kann-Bogenfelder können auch nicht ausgefüllt sein, also den Wert LEER haben.

### *Listenfelder*

Ein Bogenfeld wird dann als Liste interpretiert, wenn im referenzierten Feld (Tabelle `Feld`) der Wert des Attributes `Feld.istListe` ist. Ansonsten ist das Bogenfeld ein Skalar. Bei der Formulierung von Regeln ist darauf zu achten, dass Listenfelder nicht bei jedem Operator als Operand fungieren können. Listenfelder dürfen z.B. nicht voneinander subtrahiert werden.

Ein Beispiel ist das Feld `SSRISIKO`, welches z.B. im Modul 16/1 als Bogenfeld vorkommt. Hinter `SSRISIKO` verbirgt sich eine Liste mit neun Elementen, welche nachfolgend als Variable in einer Regel angesprochen wird:

```
ANZSSVORHER = 0 UND SSRISIKO EINSIN (23)
```

Andererseits können auch die Elemente einer Liste einzeln angesprochen werden:

```
SSRISIKO_1, SSRISIKO_2, .., SSRISIKO_9
```

Die Nummer des jeweiligen Elementes (zwischen 1 und `listenElementeMax`) wird mit vorhergehendem Unterstrich an den Schluss des Feldnamens gehängt.

### *Literale*

---

<sup>27</sup> Bei den Ersatzregeln in Tabelle `MehrfachRegel` sind stattdessen die Exportfelder des Moduls erlaubt.

Bis auf die numerischen Typen (GANZEZAHL und ZAHL) müssen Literale von einfachen Hochkommata eingeschlossen sein oder Zeichenketten von doppelten Anführungsstrichen.<sup>28</sup>

### Beispiele für Regeln:

```
OPDAUER < 10

ALLGEMEINANA = 0 UND REGIONALANAE = 0

POKOMPLIKAT = 1 UND WUNDINFABSZE = LEER UND ABSZESS = LEER UND HAEMATBLUTUN = LEER UND DEKUBITUS
= LEER UND SONSTKOMPLIK = LEER

HISTOL NICHTIN (17; 19; 22; 23; 24; 29; 31; 39; 72) UND TNMPT <> LEER

VOROPDATUM > aktuellesDatum()

ENTLGRUND <> 07
Dieser Ausdruck ist zu interpretieren wie ENTLGRUND <> 7, da der dem Feld ENTLGRUND zu Grunde liegende Schlüssel numerisch ist.
```

### *Listen von Literalen*

Literale können sowohl als Skalare als auch als Listen angesprochen werden. Der Separator einer Liste von Literalen ist das Semikolon.

### Beispiele:

- Liste von Literalen vom Typ GANZEZAHL oder SCHLUESSEL (numerisch) :  
(1; 2; 3)
- Liste von Literalen vom Typ SCHLUESSEL (nichtnumerisch) :  
( '5-282.0' ; '5-282.x' ; '5-282.y' )
- Ein Sonderfall ist die Liste mit dem Element LEER, welche z.B. benötigt wird, wenn man prüfen will, ob alle Listenfelder ausgefüllt sind:  
(LEER)

Längere Listen von Prozedurcodes (OPS) oder Diagnosecodes (ICD-GM-10) werden als Variable angesprochen, deren Namen einem festen Namensschema gehorchen (Kapitel 4).

### Beispiel:

---

<sup>28</sup> Beim Export entfallen die begrenzenden Zeichen!

In der Regel 323<sup>29</sup> aus Modul 12/3 (Leistenhernie) wird die OPS-Liste `OPS_RezidivHernien` verwendet.

`ARTHERNIE = 2 UND OPSCHLUESSEL KEINSIN OPS_RezidivHernien`

Außerdem gibt es Teildatensatz-Listenfelder, welche im Abschnitt 3.5.2 beschrieben werden.

### Operatoren

Tabelle 19 gibt einen groben Überblick über die in der Syntax zulässigen Operatoren. Der aktuelle Überblick über alle zulässigen Operationen (inkl. Operanden) ist in Tabelle `SyntaxOperator` der Spezifikation zu finden.

Tabelle 19: Präzedenz und Assoziativität der Operatoren<sup>30</sup>

Präzedenz	Assoziativität	Operator	Erläuterung
0	links	IN	Die Variable und die Feldelemente müssen gleichen Typs sein.
	links	NICHTIN	
	links	EINSIN	Mengenoperator
	links	JEDESIN	Mengenoperator
	links	EINSNICHTIN	Mengenoperator
	links	KEINSIN	Mengenoperator
1	links	*	Multiplikation
	links	/	Division
2	links	+	Addition
	links	-	Subtraktion
3	links	<	Vergleichsoperator „kleiner“
	links	>	Vergleichsoperator „größer“
	links	<=	Vergleichsoperator „kleiner gleich“
	links	>=	Vergleichsoperator „größer gleich“
4	links	=	Vergleichsoperator
	links	<>	Vergleichsoperator „ungleich“
5	rechts	NICHT	logisches Nicht

<sup>29</sup> Beispiel aus BQS-Spezifikation 12.0

<sup>30</sup> In dieser Übersichtstafel hat jeder einzelne Operator eine Präzedenzstufe (höchste Präzedenzstufe ist 0). Operatoren, welche die gleiche Stufe haben, werden stattdessen nach den Regeln der Assoziativität aufgelöst.

Präzedenz	Assoziativität	Operator	Erläuterung
6	links	UND	logisches Und
7	links	ODER	logisches Oder

### Anmerkungen:

- In der Tabelle `SyntaxOperator` wird nicht zwischen `NUMSCHLUESSEL` und `SCHLUESSEL` differenziert, da für beide Typen z.Z. die gleichen Operationen (`=`, `IN`, `<>`, `EINSIN`, `KEINSIN`, `JEDESIN`, `NICHTIN`) gelten.
- Folgende Operatoren erfordern entweder nur rechts oder links und rechts Listenfelder:  
`IN`, `NICHTIN` (nur rechts)  
`EINSIN`, `KEINSIN`, `JEDESIN`, `EINSNICHTIN` (links und rechts)
- Die Operation  
`GANZEZAHL := DATUM1 - DATUM2`  
liefert als Ergebnis die Differenz zwischen zwei Kalenderdaten in Tagen.
- Die Operation  
`ZAHL := UHRZEIT1 - UHRZEIT2`  
liefert als Ergebnis die Differenz zwischen zwei Uhrzeiten in Minuten.

### Operatoren mit beidseitigen Listenfeldern als Operanden:

- `EINSIN`: Wenn mindestens ein Element aus der linken Liste in der rechten Liste enthalten ist, so ist der Ausdruck wahr (nichtleere Schnittmenge)
- `KEINSIN`: Wenn kein Element der linken Liste in der rechten Liste enthalten ist, so ist der Ausdruck wahr (Leere Schnittmenge). Dieser Operator ist redundant, da er auch durch Negation des `EINSIN`-Operators abgedeckt ist.
- `JEDESIN`: Der Ausdruck ist dann wahr, wenn jedes Element der linken Liste in der rechten Liste enthalten ist (Teilmenge)
- `EINSNICHTIN`: Der Ausdruck ist dann wahr, wenn mindestens ein Element der linken Liste nicht in der rechten Liste enthalten ist (nichtleere Differenz)

### Beispiele:

- Folgende Regel prüft, ob kein Element des Listenfeldes `OPSCHLUESSEL` (4 Elemente) einen bestimmten Code besitzt:  
`OPSCHLUESSEL KEINSIN ('5-983')`  
Wenn z.B. `OPSCHLUESSEL := ('5-661.3y'; LEER; LEER; LEER)`, so ist die Re-



gel erfüllt.

Gleichwertig ist die Regel:

NICHT OPSCHLUESSEL EINSIN ('5-983')

- Eine Besonderheit bei Listenoperationen ist die Prüfung, ob alle Elemente einer Liste ausgefüllt sind:

NICHT OPSCHLUESSEL JEDESIN (LEER)

Diese Bedingung erfordert, dass zumindest ein Listenelement ausgefüllt ist. Z.B. erfüllt

OPSCHLUESSEL:= ('5-661.3y'; LEER; LEER; LEER) die Bedingung.

Gleichwertig ist die Regel:

OPSCHLUESSEL EINSNICHTIN (LEER)

### Hinweis

Folgende Operatoren sind komplementär:

- IN und NICHTIN
- EINSIN und KEINSIN
- JEDESIN und EINSNICHTIN

### Beispiel:

Folgende Ausdrücke sind gleich:

1) A EINSNICHTIN B

2) NICHT A JEDESIN B

### *Funktionen*

Eine Funktion ist gekennzeichnet durch ihren Namen, an den sich *unmittelbar* (ohne Leerzeichen) ein Listenausdruck anschließt. Funktionen ohne Übergabeparameter werden ähnlich wie in C oder Java durch ein Klammerpaar abgeschlossen.

Der aktuelle Stand der in der Syntax verwendeten Funktionen ist in der Tabelle *Syntax-Funktion* der Spezifikation zu finden.

Häufig wird nur die Signatur von Funktionen bereitgestellt. Weitere Hinweise für die Implementierung findet man in Anhang A. In den nachfolgenden Beispielen gilt folgende Notation für Funktionen:

```
<BASISTYP> <FUNKTIONSNAME> ([ <BASISTYP> <VARNAME> { ; <BASISTYP> <VARNAME> } ] )
```

mit

{ }           Wiederholung

[ ]           Option

<BASISTYP>   Basistyp der Variablen

<VARNAME> Name der Variablen

### Beispiele:

- DATUM aktuellesDatum()  
Funktion ohne Übergabeparameter und mit Ergebnistyp DATUM
- DATUM Minimum(DATUM DATUMLISTE)  
Funktion mit Ergebnis vom Typ DATUM, welche das Minimum einer Liste von Datumsangaben (DATUMLISTE) liefert.
- JAHRDATUM jahreswert(DATUM EINDATUM)  
Funktion mit Ergebnis vom Typ JAHRDATUM

Zurzeit kommen verschachtelte Funktionsaufrufe (z.B. funktionA(funktionB())) oder arithmetische Ausdrücke als Funktionsargumente (z.B. funktion(x+y)) nicht vor.

### *Plausibilitätsprüfungen mit OPS- und ICD-Listen*

Die OPS- und ICD-Listen enthalten ausschließlich Normcodes. Die im Krankenhaus dokumentierten Codes enthalten ggf. auch Zusatzkennzeichen. Bei der Evaluation der Regeln werden die dokumentierten Zusatzkennzeichen ignoriert (vergleiche auch Kapitel 4.1 und 4.2).

### Beispiel:

Die OPS-Liste KAT OPS enthält unter Anderem den Code 5-144.01. Die Evaluation der Regel OPSCHLUESSEL EINSIN KAT OPS führt auch dann zu einem positiven Ergebnis, wenn OPSCHLUESSEL = ('5-144.01:R'; LEER; LEER; LEER)

## **3.5 Teildatensatzübergreifende Regeln**

Eine Regel ist teildatensatzübergreifend, wenn die Datenfelder der Regel aus mehreren Teildatensätzen eines Moduls stammen.

### **3.5.1 Klassifizierung der teildatensatzübergreifenden Regeln**

Es gibt zwei Arten von teildatensatzübergreifenden Regeln:

- a) Die Felder sind auf verschiedenen Teildatensätzen eines Moduls definiert

#### Beispiel:

Regel Modul 21/3: OPDATUM > ENTLDATUM

Das Bogenfeld OPDATUM ist in Teildatensatz 21/3:PROZ, das Bogenfeld ENTLDATUM in 21/3:B definiert.

- b) Ein Feld der Regel ist in einem wiederholbaren Teildatensatz definiert und die Regel bezieht sich auf alle Werte des Datenfeldes innerhalb eines Datensatzes (= Summe aller Teildatensätze eines Vorgangs).

Beispiel:

Die Werte des Bogenfeldes LFDNREINGRIFF müssen - bezogen auf alle 21/3:PROZ-Teildatensätze einer QS-Dokumentation - eindeutig sein. Diese Regel findet sich nicht in der Tabelle Regeln, sondern ist über das Attribut fkEindeutigBogenFeld der Tabelle Bogen definiert (vgl. Kapitel 2.1.3).

### 3.5.2 Regeln mit Teildatensatz-Listefeldern

Zu jedem skalaren Datenfeld eines wiederholbaren Teildatensatzes existiert ein *Teildatensatz-Listefeld* (kurz TDS-Listefeld), welches über das @-Zeichen vor dem Feldnamen angesprochen wird. Das TDS-Listefeld enthält sämtliche Werte des betreffenden Datenfeldes, welche innerhalb der QS-Dokumentation eines Falles existieren.

Beispiel:

Der Teildatensatz 21/3:PROZ enthält das Datenfeld 51 EXITUS (Exitus im Herzkatheterlabor). Wurden während eines Aufenthaltes zwei Operationen (Prozedurnummer 1 und 2) durchgeführt, so werden zwei Teildatensätze 21/3:PROZ angelegt und das Datenfeld EXITUS muss zweimal dokumentiert:

21/3:PROZ[LFDNREINGRIFF=1]:EXITUS = 0 (nein, Patient überlebt den Eingriff)

21/3:PROZ[LFDNREINGRIFF=2]:EXITUS = 1 (ja, Patient verstirbt während des Eingriffs)

Das entsprechende TDS-Listefeld lautet:

@EXITUS = (0; 1)

TDS-Listefelder können in Feldgruppen und Plausibilitätsregeln verwendet werden. Bei Verwendung in Feldgruppen hat das Attribut tdsListe in der Tabelle FeldGruppeFelder den Wert wahr.

Beispiel:

Die Regel 12114<sup>31</sup>

@EXITUS EINSIN (1) UND ENTLGRUND <> 7

erzwingt die korrekte Dokumentation des Datenfeldes ENTLGRUND (Entlassungsgrund nach §301) auf dem Basis-Teildatensatz 21/3:B, wenn der Patient im Herzkatheterlabor verstorben ist.

### 3.6 Verfahren für die Evaluation von Regeln

Grundsätzlich muss jede gemäß Kapitel 3.4 formulierte Regel evaluiert werden, wenn keine der folgenden Bedingungen zutrifft:

- 1 Für mindestens ein referenziertes Bogenfeld<sup>32</sup> schlägt eine *harte* Feldprüfung (Kapitel 3.2) fehl<sup>33</sup>.
- 2 Ein Feld der Regel ist LEER (nicht ausgefüllt) und ist Bestandteil einer Operation, die einen Wert ungleich LEER erfordert (z.B. FELD > 0). Explizite Vergleiche gegen LEER (Ausdrücke wie FELD = LEER oder FELD <> LEER, siehe Abschnitt 3.4 und Tabelle *SyntaxOperator*) sind ausgenommen. Ausdrücke mit Listenoperatoren (EINSIN, KEINSIN, JEDESIN, EINSNICHTIN, IN, NICHTIN) sind ebenso nicht betroffen, da sie auch dann evaluierbar sind, wenn links ein Feld den Wert LEER hat oder rechts eine leere Liste steht ( (LEER) oder () ) steht. Für diese Bedingung gilt folgende Ausnahme<sup>34</sup>:  
 Innerhalb einer Regel steht vor einem Term (z.B. FELD > 0) ein Vergleich desselben Feldes gegen LEER (z.B. FELD = LEER ODER FELD > 0)  
 Dieser Teilausdruck ist auch für den Wert LEER der Variablen FELD zu evaluieren und hat das Ergebnis wahr.<sup>35</sup>

<sup>31</sup> Beispiel aus BQS-Spezifikation 12.0

<sup>32</sup> Die Verbindung zwischen Regeln und Bogenfeldern geschieht über die Tabelle *RegelFelder* (Kapitel 3.3.2)

<sup>33</sup> Erst bei Fehlerfreiheit der feldbezogenen Prüfungen werden die feldübergreifenden Prüfungen durchgeführt.

<sup>34</sup> Diese Ausnahmeregelung ist beschränkt auf Teilausdrücke der folgende Struktur:

FELD = LEER ODER FELD OPERATOR OPERAND

OPERATOR: Operator gemäß BQS-Syntax, z.B. >, <>, <=

OPERAND: zulässiger Operand

<sup>35</sup> Die komplette Evaluation des booleschen Ausdrucks würde zu einer Ausnahmebedingung führen. Die Teilevaluation des ersten Terms ist in diesem Fall ausreichend.

- 3 Eine Funktion der Regel hat das Ergebnis LEER und ist Bestandteil einer Operation, die einen Wert ungleich LEER Wert erfordert.

Beispiel:

```
gestAlter(abstGebterm; TRAGZEITKLIN; SSRISIKO) >= 224
```

Hintergrund der zweiten und dritten Bedingung ist, dass bei Kann-Feldern die Evaluation einer Regel zu einem Programmfehler führen kann und die Regel somit nicht evaluierbar ist. Diese Fehler können entweder vor der Regelevaluation durch Abfangen von Konstellationen undefinierter Werte<sup>36</sup> oder durch eine geeignete Ausnahmebehandlung<sup>37</sup> während der Evaluation vermieden werden.

*Evaluation teildatensatzübergreifender Regeln*

Teildatensatzübergreifende Regeln (Typ a) in Kapitel 3.5) müssen u. U. mehrfach evaluiert werden (für jede Kombination von Teildatensätzen, welche von der Regel betroffen sind).

Beispiel:

In Modul 21/3 (Koronarangiographie und Perkutane Koronarintervention (PCI)) gibt es die Regel 8867:

```
OPDATUM > ENTLDATUM
```

Diese Regel hat einen Bezug zum Teildatensatz 21/3:B (über das Feld ENTLDATUM) und zum Teildatensatz 21/3:PROZ (über das Feld OPDATUM). Es wird angenommen, dass bei einem Patienten folgende Eingriffe durchgeführt wurden:

03.05.2009: Eingriff (Eingriffsnummer 1) Koronarangiographie

06.05.2009: Eingriff (Eingriffsnummer 2) PCI

07.05.2009: Entlassung aus dem Krankenhaus

In der QS-Dokumentation wird das Modul 21/3 mit den Teildatensätzen 21/3:B, 21/3:PROZ[1] und 21/3:PROZ[2]<sup>38</sup> angelegt. In den eckigen Klammern findet sich der Wert des Datenfeldes Eingriffsnummer LFDNREINGRIFF aus dem Teildatensatz 21/3:PROZ.

Dann gibt es für die Felder ENTLDATUM und OPDATUM folgende Werte:

---

<sup>36</sup> durch *if*-Abfragen

<sup>37</sup> Exception-Handling

<sup>38</sup> Zusätzlich muss es auch die Teildatensätze 21/3:KORO[LFDNRKORO=1] und 21/3:PCI[LFDNRPCI=1] geben (Abbildung 2) .

21/3:B.ENTLDATEM = '07.05.2009'

21/3:PROZ[LFDNREINGRIFF=1].OPDATEM = '03.05.2009'

21/3:PROZ[LFDNREINGRIFF=2].OPDATEM = '06.05.2009'

Die Regel muss somit zweimal evaluiert werden. Nachfolgend sind die Datumswerte bereits eingesetzt:

'03.05.2009' > '07.05.2009'

'06.05.2009' > '07.05.2009'

#### Achtung:

In wenigen Einzelfällen beziehen sich Plausibilitätsregeln auf mehr als zwei Teildatensätze.

### 3.7 Feldgruppen

Bogenfelder eines Moduls können zu einer Feldgruppe zusammengefasst werden. Zwischen den Feldern einer Feldgruppe sind logische Abhängigkeiten definiert. In der Praxis bedeutet das, dass der Anwender daran gehindert wird, Felder mit Werten auszufüllen, welche der Logik der Feldgruppe widersprechen.

Feldgruppen ergeben sich indirekt aus der Definition von Plausibilitätsregeln. Ein Satz von Plausibilitätsregeln fasst Felder zu einer logischen Einheit zusammen. Die explizite Definition von Feldgruppen strukturiert sowohl die Bogenfelder als auch die Plausibilitätsregeln.

Feldgruppen betreffen eine logisch zusammenhängende Gruppe von Datenfeldern eines Moduls. Die möglichen Antworten<sup>39</sup> eines jeden Datenfeldes werden in zwei Gruppen aufgeteilt. Die erste Gruppe ist die Menge der positiven und die zweite Gruppe die Menge der negativen Antworten. Eine typische positive Antwort ist „Feld <> LEER“ und die komplementäre negative Antwort „Feld = LEER“.

Eine Feldgruppe kann ein *Filterfeld* haben. Wenn die Antwort dieses Filterfeldes negativ ausfällt, so darf keines der anderen Felder (*abhängige Felder*) positiv beantwortet werden.

Tabelle 20 gibt einen Überblick über die Typen von Feldgruppen. Der aktuelle Stand findet sich in der Tabelle `FeldGruppenTyp` der BQS-Spezifikation.

Tabelle 20: Typen von Feldgruppen

Name	Bemerkung
------	-----------

<sup>39</sup> Die Antworten eines Datenfeldes umfassen hier neben möglichen Werten (z.B. Schlüsselwerten) oder Wertemengen auch die Kategorie „nicht ausgefüllt“ (LEER)

MF_OPTIONAL_FILTER	Optionale Mehrfachauswahl mit Filterfeld
MF_MINDESTENS1_FILTER	Obligatorische Mehrfachauswahl mit Filterfeld
MF_ALLES_FILTER	Mehrfachauswahl mit Filterfeld, alle abhängigen Felder müssen ausgefüllt sein
EF_FILTER	Einfachauswahl mit Filter
EF_OPTIONAL_FILTER	Optionale Einfachauswahl mit Filter
EF	Einfachauswahl
MF_MINDESTENS1	Obligatorische Mehrfachauswahl ohne Filterfeld
UND	Einfache Regel mit Und-Verknüpfungen
MF_OPTIONAL	Optionale Mehrfachauswahl ohne Filterfeld

### 3.7.1 Struktur der Tabellen `FeldGruppe` und `FeldgruppeFelder`

Die Feldgruppen sind in den Tabellen `FeldGruppe` und `FeldgruppeFelder` definiert. In der Tabelle `FeldGruppe` (Tabelle 21) sind Name, Typ und Zuordnung zu einem Modul der Feldgruppen definiert. Die Verknüpfungstabelle `FeldgruppeFelder` (Tabelle 22) definiert die abhängigen Bogenfelder und die Filterbogenfelder der Feldgruppen.

Tabelle 21: Struktur der Tabelle `FeldGruppe`

Feldname	Feldtyp	Bemerkung
<code>idFeldGruppe</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT(64)	Technischer Name der Feldgruppe
<code>fkModul</code>	INTEGER	Obligatorischer Fremdschlüssel zu einem Modul
<code>fkFeldgruppenTyp</code>	INTEGER	Obligatorischer Fremdschlüssel zu einem Feldgruppentyp
<code>hinweis</code>	TEXT	Bei Filter-Feldgruppen relevant für die Gestaltung der Eingabemaske Der Hinweistext informiert den Anwender über die Bedingungen, welche das Ausfüllen von ein oder mehreren abhängigen Feldern erforderlich machen. Der Hinweistext kann bei der Erstellung der Eingabemasken verwendet werden. Beispiel: Der Hinweistext „Bei postoperativen Komplikationen“ wird oberhalb eines Blocks von Detailkomplikationen angezeigt.

fkFilterFeldTyp	CHAR(1)	Attribut wird bei Feldgruppen mit mehreren Filterfeldern gesetzt: O = Oder-Verknüpfung der positiven Filterbedingungen U = Und-Verknüpfung der positiven Filterbedingungen
fkRegelTyp	CHAR(1)	Fremdschlüssel zur Tabelle RegelTyp: Die Regeltypen sind die in Kapitel 3.1 beschriebenen Arten der Plausibilitätsprüfungen: H, W oder D Die generierten Einzelregeln der Feldgruppe haben den gleichen Regeltyp.
nurPositiv	BIT	Nur bei Filter-Feldgruppen wirksam: Wenn wahr, dann umfasst die Feldgruppe nur diejenigen Regeln, welche sich auf die positive Filterbedingung beziehen
grauWennNegativ	BIT	definiert eine Layout-Feldgruppe, wenn ja (siehe Abschnitt 3.7.5)

Tabelle 22: Struktur der Tabelle FeldgruppeFelder

Feldname	Feldtyp	Bemerkung
idFeldgruppeFelder	INTEGER	Primärschlüssel
fkFeldGruppe	INTEGER	Obligatorischer Fremdschlüssel zur Feldgruppe
fkBogenFeld	INTEGER	Obligatorischer Fremdschlüssel zum Bogenfeld
bedingung	TEXT	Positive Bedingung für das jeweilige Bogenfeld
istFilter	BOOL	legt fest, ob das jeweilige Bogenfeld ein Filterfeld ist
bezeichnungSchluesselListe	TEXT	Abkürzende Bezeichnung für eine Schlüsseliste in der Bedingung, wird beim Generieren von Fehlermeldungen verwendet.
tdsFilter	BIT	Das Bogenfeld wird in Regeln als TDS-Listenfeld (Kapitel 3.5.2) verwendet (Vorstellen des @-Zeichens vor Feldnamen)

### 3.7.2 Syntax der Feldgruppenregeln

In den Tabellen FeldGruppe bzw. FeldgruppeFelder sind die positiven Bedingungen für das Filterbogenfeld bzw. die abhängigen Bogenfelder einer Feldgruppe definiert. Jede Bedingung hat folgenden Aufbau:



<Operator> <Operand>

Der linke Operand wird hier weggelassen, weil er immer der Name des jeweiligen Bogenfeldes ist. Die komplette Bedingung für das Bogenfeld einer Feldgruppe lautet also:

<Bogenfeld> <Operator> <Operand>

Als Operator kann jeder dyadische Operator der Tabelle 19 verwendet werden.

Die auf der rechten Seite erlaubten Operanden sind nachfolgend aufgelistet:

- Literale (siehe Tabelle 18)
- LEER
- Kodelisten, in denen auch die Codes eines Schlüssels referenziert werden können ( (1;2;3) oder (MaDCIS) )
- ICD-Listen oder OPS-Listen (z.B. OPS\_Axilladissektion)

Hinweis: Der rechte Operand darf kein Bogenfeld sein, da sich eine Feldbedingung immer genau auf ein Bogenfeld bezieht.

#### Beispiele für Bedingungen von Feldgruppen:

- ERSCHWNEBDG = 1
- HERZFEHLER <> LEER
- OPSCHLUESSEL EINSIN OPS\_Axilladissektion
- POSTICDO3 IN (MaDCIS) ist gleichbedeutend mit POSTICDO3 IN ('8500/2'; '8503/2'; '8504/2'; '8507/2'; '8540/3'; '8543/3'), da der Schlüssel MaDCIS die aufgelisteten Codes umfasst.

### 3.7.3 Formale Definition von Feldgruppen

$A$  sei ein Bogenfeld einer Feldgruppe. Dann seien  $p(A)$  die positiven und  $n(A)$  die negativen Bedingungen, welche jeweils das Ergebnis wahr oder falsch haben können.

Eine Feldgruppe bestehe insgesamt aus  $n$  abhängigen Bogenfeldern:

$A_1, A_2, \dots, A_n$

Eine Feldgruppe kann ggf. ein Filterfeld haben, welches mit  $F$  bezeichnet wird.

Dann lässt sich eine Feldgruppe in folgender Tabelle darstellen:

Tabelle 23: Formale Definition einer Feldgruppe

Feld	Positive Bedingung	Negative Bedingung	Bemerkung
F	$p(F)$	$n(F)$	falls Feldgruppentyp mit Filter
$A_1$	$p(A_1)$	$n(A_1)$	
$A_2$	$p(A_2)$	$n(A_2)$	
$A_3$	$p(A_3)$	$n(A_3)$	
..			
$A_n$	$p(A_n)$	$n(A_n)$	

In Abhängigkeit von den Feldgruppentypen werden unterschiedliche Einzelregeln generiert.

### **Feldgruppen mit Filter**

#### *Regeln der Feldgruppe "Optionale Mehrfachauswahl mit Filterfeld"*

*(MF\_OPTIONAL\_FILTER)*

$$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$$

Insgesamt sind  $n$  Einzelregeln mit der Feldgruppe verknüpft.

#### *Regeln der Feldgruppe "Obligatorische Mehrfachauswahl mit Filterfeld"*

*(MF\_MINDESTENS1\_FILTER)*

$$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$$

$$p(F) \text{ UND } n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$$

Insgesamt sind  $n+1$  Einzelregeln mit der Feldgruppe verknüpft.

#### *Regeln der Feldgruppe "Mehrfachauswahl mit Filterfeld, alle abhängigen Felder müssen positiv beantwortet sein" (MF\_ALLES\_FILTER)*

$$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$$

$$p(F) \text{ UND } n(A_i) \quad i = 1, \dots, n$$

Insgesamt sind  $2^n$  Einzelregeln mit der Feldgruppe verknüpft.

**Regeln der Feldgruppe "Einfachauswahl mit Filter" (*EF\_FILTER*)**

$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$   
 $p(F) \text{ UND } n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$   
 $p(F) \text{ UND } p(A_j) \text{ UND } p(A_i) \text{ für alle unterschiedlichen } i, j = 1, \dots, n$

Insgesamt sind  $n(n+1)/2+1$  Einzelregeln mit der Feldgruppe verknüpft.

**Regeln der Feldgruppe "Optionale Einfachauswahl mit Filter" (*EF\_OPTIONAL\_FILTER*)**

$n(F) \text{ UND } p(A_i) \quad i = 1, \dots, n$   
 $p(F) \text{ UND } p(A_j) \text{ UND } p(A_i) \text{ für alle unterschiedlichen } i, j = 1, \dots, n$

Insgesamt sind  $n(n+1)/2$  Einzelregeln mit der Feldgruppe verknüpft.

**Achtung**

Wenn in einer Feldgruppe mit Filter das Attribut `nurPositiv` gesetzt ist, so sind nur die Einzelregeln mit positiver Filterbedingung Bestandteil der Feldgruppe.

Beispiel:

Die Feldgruppe *EF\_FILTER* mit `nurPositiv=ja` hat folgende Einzelregeln:

$p(F) \text{ UND } n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$   
 $p(F) \text{ UND } p(A_j) \text{ UND } p(A_i) \text{ für alle unterschiedlichen } i, j = 1, \dots, n$

**Feldgruppen ohne Filter**

**Regeln der Feldgruppe "Einfachauswahl" (*EF*)**

$n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$   
 $p(A_j) \text{ UND } p(A_i) \text{ für alle unterschiedlichen } i, j = 1, \dots, n$

Insgesamt sind  $n(n-1)/2+1$  Einzelregeln mit der Feldgruppe verknüpft.

### *Regeln der Feldgruppe "Obligatorische Mehrfachauswahl" (MF\_MINDESTENS1)*

$$n(A_1) \text{ UND } n(A_2) \text{ UND } \dots \text{ UND } n(A_n)$$

Insgesamt ist eine Einzelregel mit der Feldgruppe verknüpft.

### *Regeln der Feldgruppe "Und-Regel" (UND)*

$$p(A_1) \text{ UND } p(A_2) \text{ UND } \dots \text{ UND } p(A_n)$$

Insgesamt ist eine Einzelregel mit der Feldgruppe verknüpft.

## **3.7.4 Feldgruppen mit mehreren Filterfeldern**

Seit der BQS-Spezifikation 8.0 besteht die Möglichkeit, Feldgruppen mit mehr als einem Filterfeld zu definieren:

Formal gibt es dann die Filterfelder  $F_1, F_2, \dots, F_n$  mit den positiven bzw. negativen Bedingungen  $p(F_j)$  bzw.  $n(F_j)$ . Für alle Filterfelder wird eine positive Bedingung  $p(F_1, \dots, F_n)$  und eine negative Bedingung  $n(F_1, \dots, F_n)$  gebildet. Diese modifizierten Filterbedingungen ersetzen die im Kapitel 3.7.3 definierten Filterbedingungen  $p(F)$  und  $n(F)$  bei den Einzelregeln.

Die Filterfelder können entweder über eine ODER-Verknüpfung oder eine UND-Verknüpfung mit einander verbunden sein:

- $p(F_1, \dots, F_n) = p(F_1) \text{ ODER } p(F_2) \text{ ODER } \dots \text{ ODER } p(F_n)$  (ODER-Verknüpfung)
- $p(F_1, \dots, F_n) = p(F_1) \text{ UND } p(F_2) \text{ UND } \dots \text{ UND } p(F_n)$  (UND-Verknüpfung)

Der Verknüpfungstyp ist im Attribut `fkFilterFeldTyp` der Tabelle `FeldGruppe` hinterlegt.

## **3.7.5 Gestaltung von Eingabemasken mit Feldgruppen**

Feldgruppen, bei denen das Attribut `grauWennNegativ` in der Datenbanktabelle `FeldGruppe` (Erläuterung in Tabelle 21) gesetzt ist, sollen für die Gestaltung von Eingabemasken verwendet werden und werden nachfolgend auch als *Layout-Feldgruppen* bezeichnet.

Der Attributname `grauWennNegativ` wurde gewählt, weil die abhängigen Felder der Layout-Feldgruppen auf den generierten Dokumentationsbögen (Kapitel 2.4) eingegraut sind. Abbildung 8 zeigt die Layout-Feldgruppe `12/3:POPKOMP` mit dem Filterfeld 36 und den beiden abhängigen Feldern 37 und 38.

36	allgemeine behandlungsbedürftige postoperative Komplikation(en)	<input type="checkbox"/>
	0 = nein 1 = ja	
<b>wenn Feld 36 = 1</b>		
37	Wundhämatom/Nachblutung	<input type="checkbox"/>
	1 = ja	
38	sonstige	<input type="checkbox"/>
	1 = ja	

Abbildung 6 Feldgruppe `12/3:POPKOMP` auf dem Dokumentationsbogen

Layout-Feldgruppen haben folgende Eigenschaften:

- Sie haben ein oder mehrere Filterfelder.
- Jedes abhängige Feld hat die Bedingung = LEER oder EINSNICHTIN (LEER) (Attribut `bedingung` in Tabelle `FeldGruppeFelder`)
- Das Attribut `nurPositiv` hat den Wert nein

Bei Vorliegen dieser drei Eigenschaften müssen die abhängigen Felder leer bleiben, wenn die negative Filterbedingung bei der Dokumentation eines Falles erfüllt ist.

Beispiel:

Wenn in Datenfeld 36=0 (nein) angegeben ist, so müssen die Datenfelder 37 und 38 leer bleiben. Die ersten beiden Plausibilitätsprüfungen (siehe Abbildung 7) stellen dieses sicher.

Bedingung	Fehlermeldung	Regeltyp	fkFeldGruppe
POKOMPLIKAT <> 1 UND HAEMATBLUTUN <> LEER	"allgemeine behandlungsbedürftige postoperative Komplikation(en)" ist nicht "ja" [1], obwohl "Wundhämatom/Nachblutung" ausgefüllt ist	hart	12/3:POKOMPLIKAT
POKOMPLIKAT <> 1 UND SONSTKOMPLIK <> LEER	"allgemeine behandlungsbedürftige postoperative Komplikation(en)" ist nicht "ja" [1], obwohl "sonstige" ausgefüllt ist	hart	12/3:POKOMPLIKAT
POKOMPLIKAT = 1 UND HAEMATBLUTUN = LEER UND SONSTKOMPLIK = LEER	"allgemeine behandlungsbedürftige postoperative Komplikation(en)" ist "ja" [1], obwohl keines der Felder (Verlauf) "Wundhämatom/Nachblutung" und "sonstige" ausgefüllt ist	hart	12/3:POKOMPLIKAT

Abbildung 7 Plausibilitätsregeln der Feldgruppe 12/3:POPKOMP in Spezifikation 12.0 (Abfrage *PlausibilitätsregelnFürEinModul*)

Abbildung 8 zeigt die zugehörige Feldgruppeneffinition (Abfrage *FeldgruppeFürEinModul* = Zusammenschau der Tabellen *Feldgruppe* und *FeldgruppeFelder*)<sup>40</sup>.

Name Feldgruppe	Bogenfeld	bedingung	nurPositiv	fkFeldGruppenTyp
12/3:POKOMPLIKAT	12/3.0.36.POKOMPLIKAT	= 1	<input type="checkbox"/>	MF_MINDESTENS1_FILTER
12/3:POKOMPLIKAT	12/3.0.37.HAEMATBLUTUN	<> LEER	<input type="checkbox"/>	MF_MINDESTENS1_FILTER
12/3:POKOMPLIKAT	12/3.0.38.SONSTKOMPLIK	<> LEER	<input type="checkbox"/>	MF_MINDESTENS1_FILTER

Abbildung 8 Definition der Feldgruppe 12/3:POPKOMP in Spezifikation 12.0 (Abfrage *FeldgruppeFürEinModul*)

### Empfehlung zur Umsetzung von Layout-Feldgruppen

Ist dieses Attribut gesetzt, so darf die Benutzereingabe für die abhängigen Felder durch die Erfassungssoftware deaktiviert werden, falls die negative Filterbedingung zutrifft.

Bei der Umsetzung muss Folgendes sichergestellt werden:

- Nach jeder Änderung der Inhalte der Filterfelder im Erfassungsformular muss das Programm die Filterbedingung der Feldgruppe evaluieren und ggf. eine Aktualisierung der Oberfläche durchführen.
- Die Benutzereingabe für die abhängigen Felder darf nur dann deaktiviert werden, wenn keines dieser Felder ausgefüllt ist. Ansonsten ist der Anwender auf eine Plausibilitätsverletzung hinzuweisen.

- Wenn nach einer Benutzereingabe die positive Filterbedingung zutrifft, so sind ggf. vorher deaktivierte Eingabefelder wieder zu aktivieren. Deaktivierte Felder dürfen nicht versteckt werden.

### **3.8 Leitlinien für die Gestaltung der Benutzeroberflächen von Erfassungsprogrammen**

Die Grundsätze für die Plausibilitätsprüfungen wirken sich insbesondere auf die Gestaltung der Benutzeroberflächen in Erfassungsprogrammen aus. Die nachfolgend vorgestellten Regeln sind Ergebnis der Empfehlungen der Fachgruppen sowie der Arbeitsgruppe „Plausibilitätsprüfungen“. Grundsätzlich sollte durch die funktionale Gestaltung der Benutzeroberfläche ein Kompromiss zwischen Dateneingabekomfort einerseits und Zwang zur aktiven Eingabe korrekter Daten andererseits gefunden werden. Im Folgenden werden die Regeln für die Gestaltung von Benutzeroberflächen aufgeführt<sup>41</sup>:

- Keine Suggestion von Feldinhalten durch Vorbelegung (Defaults)<sup>42</sup>: Oberstes Prinzip bei der Gestaltung der Benutzeroberflächen ist, dass dem Anwender des Programms keine Angaben suggeriert werden. Insbesondere darf keine Vorbelegung mit Standardwerten erfolgen, die den „Nicht-Problemfall“ dokumentieren.
- Verwendung der vorgeschriebenen Fehler- und Warnmeldungen bei feldübergreifenden Regeln: Die Fehler- und Warnmeldungen sind so formuliert, dass sie möglichst nicht suggerieren, auf welche Weise widersprechende Angaben korrigiert werden sollen. Insofern sollen sie wörtlich übernommen werden.
- Keine zusätzlichen Ober-/Untergrenzen für Maße, Zeitdauern und Anzahlen: Außer den durch die Datenfeldbeschreibung und die Plausibilitätsregeln vorgegebenen Wertebereichen darf in Erfassungsprogrammen keine Einengung möglicher Merkmalsausprägungen in Wertefeldern erfolgen.
- Zwang zur aktiven Entscheidung zwischen „ja“ und „nein“: An solchen Stellen in den Dokumentationsbögen, bei denen die Auswahl „0“ (nein) und „1“ (ja) (vgl. z.B. Schlüssel JN) zu treffen ist, darf keine Voreinstellung des Wertes im Eingabefeld erfolgen. Es besteht somit der Zwang zur Eingabe eines Wertes. Nur an Stellen, an denen im Erfas-

---

<sup>40</sup> nach BQS-Spezifikation 10.0

<sup>41</sup> Externe Systeme, welche Daten an ein Erfassungsprogramm übergeben, sollten diese Grundsätze sinngemäß anwenden.

<sup>42</sup> Bei wenigen Feldern (Felder BSNR, IKNRKH, FACHABT) und bei den Zusatzfeldern darf von dieser Regel abgewichen werden.

sungsformular lediglich „1“ als Option angegeben wird, soll die Nicht-Eingabe eines Wertes als Verneinung interpretiert werden. Hintergrund dieser Differenzierung ist, dass einerseits in qualitätskritischen Bereichen eine Unterscheidung zwischen „keine Angabe“ und „nein“ erfolgen muss, es andererseits der Benutzerakzeptanz abträglich ist, wenn diese Systematik auch an allen anderen Stellen durchgängig verfolgt wird.

- Kein automatisches Verändern von Feldinhalten in Abhängigkeit von anderen Feldinhalten: Ein Beispiel: wenn die Transfusion von Blut zunächst bejaht und das Feld „Fremdblut“ angekreuzt worden ist, soll das Abkreuzen des übergeordneten Feldes „Bluttransfusion“ nicht automatisch zum Entfernen des Kreuzes bei „Fremdblut“ führen. Vielmehr soll eine Fehlermeldung erfolgen und der Anwender gezwungen sein, zunächst das Kreuz bei „Fremdblut“ zu entfernen, bevor „Blutverbrauch“ verneint werden kann.



## 4 Listen von Schlüsselkodes (OPS, ICD-10-GM)

In der Spezifikation sind Listen von OPS- bzw. ICD-10-GM-Kodes<sup>43</sup> in separaten Tabellen definiert. Jede Liste hat einen technischen Namen (z.B. `KARP_ICD`) und eine erläuternde Bezeichnung (z.B. „*Einschlussdiagnosen Dekompression bei Karpaltunnelsyndrom*“).

Die Listen sind in insgesamt vier Tabellen der Spezifikationsdatenbank definiert. Die technischen Namen und erläuternden Bezeichnungen sind in den Tabellen `OPSListe` bzw. `ICDListe` definiert. Die Codes finden sich in den Tabellen `OPSWert` bzw. `ICDWert`.

### Achtung

Einige Listen der Tabellen `OPSListe` oder `ICDListe` sind als QS-Filter-Listen deklariert (Attribut `qsFilter`). Die Definitionen der QS-Filter-Listen sind in der für das Verfahrensjahr gültigen BQS-Spezifikation für QS-Filter-Software zu finden. Dort existieren die gleichen Tabellen `OPSListe`, `ICDListe`, `OPSWert` und `ICDWert` mit den entsprechenden Listendefinitionen.

### Beispiel

Die Einschlussprozeduren Perinatalmedizin `GEB OPS` für das Erfassungsjahr 2009 sind in der BQS-Spezifikation für QS-Filter-Software 12.0 zu finden.

### 4.1 OPS-Listen

Jede OPS-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `OPSListe`), welcher per definitionem folgendem Schema gehorcht:

$$\{ \langle \text{TEXT} \rangle \_ \} \text{OPS} \{ \_ \langle \text{TEXT} \rangle \}$$

Hinter `<TEXT>` verbirgt sich ein frei wählbarer Name (Erlaubte Zeichen A-Z, a-z, 0-9,\_, Umlaute sind nicht erlaubt). Die `{ }`-Ausdrücke sind optional.

### Beispiele:

---

<sup>43</sup> Die aktuell gültigen Kataloge sind über das DIMDI (<http://www.dimdi.de>) zu beziehen.

OPS_Mastektomie	Prozeduren Mastektomie
OPS_primaereSectio	Prozeduren primäre Sectio
HCH OPS_EX	Ausschlussprozeduren Herzchirurgie
GEB OPS	Einschlussprozeduren Perinatalmedizin

### *Umgang mit Seitenlokalisationen*

Die Codes der OPS-Listen enthalten keine Seitenlokalisationen, obwohl die Zusatzkennzeichen für Seitenbezeichnung R, L oder B für Prozeduren an Lokalisationen, die paarig vorhanden sind (z.B. Leiste, Niere, Oberschenkel etc.), verpflichtend zu dokumentieren sind<sup>44</sup>. Für die Prüfung, ob zwei Codes identisch sind, genügt kein einfacher String-Vergleich. Stattdessen wird ein Stringvergleich der Normcodes<sup>45</sup> durchgeführt, um die Übereinstimmung zwischen dokumentiertem Code und dem Code einer OPS-Liste zu ermitteln (Tabelle 24).

Tabelle 24: Identitätsprüfung zwischen dokumentierten OPS-Kodes und Codes von OPS-Listen

Dokumentierter OPS-Kode	OPS-Kode der OPS-Liste	Bedingung für Gleichheit (= ist Stringvergleich)
<code>normCode<sub>Dok</sub>+seite<sub>Dok</sub></code>	<code>normCode<sub>Liste</sub></code>	<code>normCode<sub>Dok</sub> = normCode<sub>Liste</sub></code>
<code>normCode<sub>Dok</sub></code>	<code>normCode<sub>Liste</sub></code>	<code>normCode<sub>Dok</sub> = normCode<sub>Liste</sub></code>

<sup>44</sup> In der QS-Dokumentation wird das Zusatzkennzeichen für die Seitenbezeichnung getrennt durch einen Doppelpunkt dem OPS-Kode angehängt. Fehlt ein erforderliches Zusatzkennzeichen, so ist die Dokumentation unplausibel (Kapitel 3.2.3).

**Beispiel:**

Für den OPS-Kode 5-144.01 (Extrakapsuläre Exzision der Linse [ECCE]: Über sklerokornealen Zugang; Mit Einführung einer kapsel fixierten Hinterkammerlinse = Einschlussprozedur für den Datensatz Kataraktchirurgie) ist eine Seitenangabe erforderlich. Daher sind folgende Codes gültig:

- 5-144.01:R
- 5-144.01:L
- 5-144.01:B

<sup>45</sup> Jeder OPS-Kode `code` lässt sich entweder als Code mit Seitenlokalisation: `code=normCode+seite` oder als Code ohne Seitenlokalisation `code=normCode` darstellen.

Achtung:

Bei allen Prüfungen mit OPS-Listen (z.B. OPSCHLUESSEL EINSIN KAT OPS) sind diese Regeln zu beachten (Kapitel 3.4).

## 4.2 ICD-Listen

Jede ICD-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `ICDListe`), welcher *per definitionem* folgendem Schema gehorcht:

$$\{ \langle \text{TEXT} \rangle \_ \} \text{ICD} \{ \_ \langle \text{TEXT} \rangle \}$$

Hinter `<TEXT>` verbirgt sich ein frei wählbarer Name (Erlaubte Zeichen A-Z, a-z, 0-9, `_`, Umlaute sind nicht erlaubt).

Die in der Tabelle `ICDWert` (Attribut `code`) definierten Codes entsprechen der Systematik der Spalte `NormCode` aus Tabelle `Codes` in den Datenquellen des DIMDI):

Der ICD-10-GM wird 4/5-stellig kodiert, kann aber von einem Suffix bestehend aus `[A|V|Z][L|R|B]` (ohne Leerzeichen, z.B. "K41.9ZL") gefolgt sein. Achtung: *Die Suffixe \*, +, ! entfallen in der Spezifikation!*<sup>46</sup>

---

<sup>46</sup> Es ist zu beachten, dass im Krankenhaus dokumentierte ICD-Kodes die Suffixe \*, +, ! enthalten können.

Beispiel:

In der ICD-Liste `GEB_ICD` ist der Code `z37.9` definiert ist. Bei der Prüfung, ob der im Krankenhaus dokumentierte Code `z37.9!` in der Liste `GEB_ICD` enthalten ist, muss die Software zu einem positiven Ergebnis kommen.

## 5 Versionierung

### 5.1 Grundlegende Definitionen

Informationen zur Version der Spezifikationsdatenbank sind in der Tabelle `Version` zu finden. Die wichtigsten Eigenschaften einer Version sind der Versionsname (Attribut `name`) und die Gültigkeitszeiträume (Attribute `ab` und `bis`). Der Gültigkeitszeitraum einer Version ist in der Regel ein Erfassungsjahr (z.B. 1.1.2009 bis 31.12.2009).

#### Achtung:

QS-Dokumentationssoftware eines Erfassungsjahres wird für Behandlungsfälle verwendet, deren *Aufnahmedatum* ins Krankenhaus in den oben definierten Gültigkeitszeitraum fällt. Bei so genannten „Überliegern“ (Aufnahmedatum im alten Jahr, Entlassungsdatum im nachfolgenden Jahr) wird die QS-Dokumentationssoftware auch noch für Behandlungsfälle benutzt, welche nach dem in der Datenbank definierten Gültigkeitszeitraum (in der Regel nach dem 31.12.) entlassen worden sind.

Jedes Modul der Datenbank hat eine Version (vgl. Attribut `fkVersion` in Tabelle `Modul`). Über die in der Datenbank definierten Relationen sind auch für alle Bogenfelder (Tabelle `BogenFeld`), Exportfelder (Tabelle `ExportFormat`) und Plausibilitätsregeln (Tabelle `Regeln`) Versionen definiert.<sup>47</sup>

#### 5.1.1 Finale Spezifikation - Zwischenstände

Versionen können den Status "*in Entwicklung*" oder "*final*" haben. Diese Zustände werden in der Nachschlagetabelle `VersStatus` verwaltet. Das Attribut `gueltig` zeigt die gültige Version der Datenbank an. Nur eine einzige Version darf als gültig markiert sein.

---

<sup>47</sup> Ein expliziter Versionsbezug von Teildatensätzen, Bogenfeldern, Exportfeldern etc. ist nicht mehr vorgesehen, er ergibt sich aus der Version der Spezifikation.

Während der Entwicklung einer neuen Spezifikation können als zusätzliche Serviceleistung für Softwareanbieter und Krankenhäuser von der BQS Zwischenstände publiziert werden, in denen Module unterschiedlicher Version enthalten sind. Eine Spezifikationsdatenbank enthält in diesem Fall Module, denen eine Version mit dem Status "*in Entwicklung*" zugeordnet ist.

Beispiel:

Die Versionen 11.0 und 12.0 einschließlich ihrer Service-Releases sind finale Versionen. Eine Datenbank, in der 12.0 die gültige Version ist, darf keine Module mit der Version 11.0 enthalten.

### **5.1.2 Historie der Versionen**

Die Tabelle `Version` enthält auch einen Selbstbezug (Attribut `fkVersion`), welche die Identifizierung der Vorgängerversion ermöglicht.

Beispiel:

Die Vorgängerversion der Spezifikation 12.0 ist die Version 11.0.

### **5.2 Delta-Informationen zur vorhergehenden finalen Version**

Um den Nutzern der Spezifikation umfassende Informationen zu den jeweiligen Änderungen zur Verfügung zu stellen, enthält die neue Spezifikationsdatenbank Tabellen, die den Änderungsstand zur letzten finalen bzw. zum letzten Service Release einer finalen Version der Datenbank anzeigen. Diese so genannten Delta-Tabellen werden automatisch generiert.

Es gibt drei Delta-Tabellen, welche die neuen Entitäten, die gelöschten Entitäten und die geänderten Attributwerte weiter bestehender Entitäten aufzeigen.

#### **5.2.1 Neue Entitäten**

Über die Attribute `id` und `fkTabellenStruktur` in der Tabelle `DeltaNeu` (Tabelle 25) ist die Zuordnung zu den Entitäten der Datenbank möglich.

Tabelle 25: Struktur der Tabelle `DeltaNeu`

Feldname	Feldtyp	Bemerkung
<code>idDeltaNeu</code>	AutoWert	Primärschlüssel
<code>id</code>	Text	ID der Entität, die eingefügt wurde
<code>fkTabellenStruktur</code>	Zahl	Bezug zur Tabelle, in der die Entität eingefügt wurde
<code>anweisungDatenentgegennahme</code>	Memo	Anweisung für die datenentgegennehmenden Stellen (wird nur bei Service Releases gefüllt)
<code>bemerkung</code>	Memo	Begründung für die Ergänzung (wird nur bei Service Releases gefüllt)

### Beispiel<sup>48</sup>:

Die Zeile `idDeltaNeu=12` enthält folgende Werte:

`id=16680`

`fkTabellenStruktur=<SchluesselWert>`

`bemerkung=Schlüsselwert 8523/3=invasives duktales Karzinom gemischt mit anderen Karzinom-Typen [Schlüssel ICDO3Mamma]`

Dies bedeutet, dass in der Tabelle `SchluesselWert` ein neuer Eintrag mit `idSchluesselWert=16680` angelegt worden ist. Unter `bemerkung` finden sich weitere Erläuterungen zum neuen Tabelleneintrag.

## 5.2.2 Geänderte Entitäten

Über die Attribute `id` und `fkTabellenFeldStruktur` der Tabelle `DeltaAttribut` (Tabelle 26) ist die Zuordnung zu den Attributen der Entitäten der Datenbank möglich.

Tabelle 26: Struktur der Tabelle `DeltaAttribut`

Feldname	Feldtyp	Bemerkung
<code>idDeltaAttribut</code>	AutoWert	Primärschlüssel
<code>id</code>	Text	ID der Entität, die geändert wurde
<code>fkTabellenFeldStruktur</code>	Zahl	Bezug zum Attribut einer Tabelle, in der die Entität geändert wurde
<code>alterinhalt</code>	Memo	alter Inhalt der geänderten Entität in der letzten finalen Spezifikation
<code>neuerinhalt</code>	Memo	neuer Inhalt dieser Entität in der aktuellen Spezifikation
<code>anweisungDatenentgegennahme</code>	Memo	Anweisung für die datenentgegennehmenden Stellen (z.B. „LQS: Negative Zahlen zulassen“), wird nur bei Service

<sup>48</sup> Version 12.0

Feldname	Feldtyp	Bemerkung
		Releases gefüllt
bemerkung	Memo	Begründung für die Ergänzung (wird nur bei Service Releases eingefügt)

#### Beispiel<sup>49</sup>:

Die Zeile `idDeltaAttribut=354` enthält folgende Werte:

`id=8055`

`fkTabellenFeldStruktur = <Regeln.fkRegelTyp>`

`alterInhalt=W`

`neuerInhalt=H`

`bemerkung=Regel 8055 (Modul HCH,Typ=H): OPSCHLUESSEL EINSIN HCH_OPS_EX`

D.h., dass in der Zeile `idRegeln=8055` der Tabelle `Regeln` das Attribut `fkRegelTyp` von W auf H geändert wurde. Die Spalte `bemerkung` enthält weitere Hinweise zum Kontext der Änderung.

### 5.2.3 Gelöschte Entitäten

Über die Attribute `id` und `fkTabellenStruktur` in der Tabelle `DeltaGeloesch` (Tabelle 27) ist die Zuordnung zu den Entitäten der Datenbank möglich.

Tabelle 27: Struktur der Tabelle `DeltaGeloesch`

Feldname	Feldtyp	Bemerkung
<code>idDeltaGeloesch</code>	AutoWert	Primärschlüssel
<code>id</code>	Text	ID der Entität, die gelöscht wurde
<code>fkTabellenStruktur</code>	Zahl	Bezug zum Attribut einer Tabelle, in der die Entität gelöscht wurde
<code>anweisungDatenentgegennahme</code>	Memo	Anweisung für die datenentgegennehmenden Stellen (z.B. „LQS: Prüfung deaktivieren“), wird nur bei Service Releases gefüllt
<code>bemerkung</code>	Memo	Begründung für die Ergänzung (wird nur bei Service Releases eingefügt)

#### Beispiel<sup>50</sup>:

---

<sup>49</sup> Version 12.0

Die Zeile `idDeltaGeloescht=2` enthält folgende Werte:

```
id=5241
fkTabellenStruktur=<BogenFeld>
bemerkung=18/1:BRUST:REGIOLYMPHKNOTEN
```

D.h, dass aus der Tabelle `BogenFeld` der Vorgängerdatenbank (11.0 SR3) die Zeile `id-BogenFeld=5241` gelöscht wurde. Unter `bemerkung` finden sich weitere Hinweise zum Kontext der gelöschten Entität.

## 5.2.4 Konfiguration der Delta-Berechnung

Es werden nur dann Delta-Informationen zu einer Entität angezeigt, wenn für mindestens ein Attribut der Entität in der Tabelle `TabellenFeldStruktur` das Attribut `deltaAktiv = Ja` gesetzt ist.

Tabelle 28: Inhalt der Tabelle `TabellenFeldStruktur`

idTabelle	feldName	...	deltaAktiv	fk TabellenFeldStruktur
445	idRegeln		Nein	Regeln
450	fkMehrfachRegel		Nein	Regeln
451	fkFeldGruppe		Nein	Regeln
449	fkRegelTyp		Ja	Regeln
446	bedingung		Ja	Regeln
447	meldung		Ja	Regeln
448	fkModul		Ja	Regeln
454	gueltigNachExport		Ja	Regeln

### Beispiel:

Das Delta für die Tabelle „Regeln“ wird für die Attribute `fkRegelTyp`, `bedingung`, `meldung`, `fkModul` und `gueltigNachExport` erzeugt.

## 5.3 Abgrenzung zwischen Erfassungsjahren und Datensatzformaten

Die datenentgegennehmenden Stellen müssen Datensätze von Krankenhäusern entgegennehmen, falls die Aufnahmedaten Krankenhaus in den Gültigkeitszeitraum der Version einer finalen Datenbank fallen. Eine Datenbank ist dann final, wenn alle ihre Module die

---

<sup>50</sup> Version 12.0



gleiche finale Version haben und diese finale Version in der Tabelle `Version` als gültig markiert ist.

Das Abgrenzungskriterium zwischen den Erfassungsjahren<sup>51</sup> ist..

- das Aufnahmedatum in das Krankenhaus (`AUFNDATUM`) bei Datensätzen stationär aufgenommenener Patienten.
- das Erhebungsdatum des Follow-Up (`FUERHEBDATUM`) bei Datensätzen ohne Aufnahmedatum in das Krankenhaus: `HTXFU`, `NTXFU`, `LTXFU`, `NLSFU` und `LLSFU`.

Achtung:

Dem Erfassungsjahr 2009 zugeordnete Fälle müssen im Format der BQS-Spezifikation 12.0 an die datenentgegennehmenden Stellen gesandt werden, sonst ist die Datenlieferung zurückzuweisen.

#### **5.4 Version des Exportverfahrens**

Das Attribut `fkGuelzigExportVerfahren` gibt an, welche Versionskürzel (Attribut `Version.name`) in der ersten Zeile der Steuerdatei verwendet werden. Nur eine Version darf als gültig für das Exportverfahren markiert sein. Da das Exportverfahren nicht jährlich geändert wird, kann die Version des Exportverfahrens der aktuellen Version der Spezifikation "hinterherhinken".

Für das Erfassungsjahr 2009 gilt das Exportverfahren 12.0.

#### **5.5 Version der Exportdateien**

Die Version der Exportdateien entspricht der gültigen Version der Spezifikation.

---

<sup>51</sup> Das Abgrenzungskriterium definiert somit die Zuordnung des Datensatzes zu einer Version der BQS-Spezifikation bzw. das Format des Datensatzes.

## 6 Datenexport

Es gibt zwei Datenexportverfahren: Die meisten Module der BQS-Spezifikation werden von den Krankenhäusern in einem *indirekten Verfahren* (QSLQS) über die Landesgeschäftsstellen Qualitätssicherung (LQS) an die BQS übermittelt. Die Module HCH, HTX, HTXFU, PNTX, PNTXFU, LTX, LTXFU, LUTX, LUTXFU, NLS, NLSFU, LLS und LLSFU werden direkt an die Bundesgeschäftsstelle Qualitätssicherung BQS gesandt (*Direktverfahren*; QSBQS).

Die datenentgegennehmende Stelle ist somit entweder die zuständige LQS oder die BQS.

### Achtung:

Die QS-Dokumentationssoftware hat dafür zu sorgen, dass auch die Minimaldatensätze dem korrekten Verfahren zugeordnet werden. Entscheidend für die Zuordnung zum Exportverfahren ist hier das Datenfeld ZUQSMODUL „Zugehöriger QS-Datensatz“.

### Achtung:

Die datenentgegennehmenden Stellen haben Datensätze, welche nicht in die Zuständigkeit ihres Verfahrens fallen, mit Fehlermeldung zurückzuweisen.

### Beispiel:

Herzchirurgische Datensätze dürfen nicht von einer Landesgeschäftsstelle akzeptiert werden.

Aus technischer Sicht sind die Exportverfahren jedoch gleich. Für jeden Teildatensatz eines Moduls wird vom Dokumentationssystem eine eigene Exportdatei generiert, welche eine Kopfzeile mit den Feldnamen und nachfolgend die exportierten Datensätze enthält (ASCII-Format mit Semikolon als Trennzeichen). Die Struktur einer Exportdatei orientiert sich - unter Berücksichtigung der Anonymisierungsvorschriften - an der Datenfeldbeschreibung des Teildatensatzes und wird um die Zusatzfelder ergänzt.

Aus einer Steuerdatei und ein oder mehreren Exportdateien wird vom Dokumentationssystem eine komprimierte und verschlüsselte Transaktionsdatei für den Versand von abgeschlossenen Vorgängen (Datensätzen) an die entgegennehmende Stelle erzeugt.

Das Übertragungsverfahren wird in einer gesonderten Spezifikation der Datenübermittlung detailliert erläutert. Es ist nicht Bestandteil der technischen Dokumentation zur Spezifikation von QS-Dokumentationssoftware.

## 6.1 Registrierung eines Dokumentationssystems

Die Registrierung eines Dokumentationssystems bei der datenentgegennehmenden Stelle ist Voraussetzung für die Datenübermittlung.

### 6.1.1 Registrierung bei einer Landesgeschäftsstelle Qualitätssicherung

Es dürfen nur solche Dokumentationssysteme, welche bei der Landesgeschäftsstelle Qualitätssicherung registriert sind, Datensätze an die Landesebene (Qualitätssicherungsverfahren  $Q_{SLQS}$ ) übermitteln.

Registrierte Dokumentationssysteme bekommen von der Landesgeschäftsstelle Qualitätssicherung eine Registriernummer zugewiesen, welche folgendermaßen aufgebaut ist:

$\langle \text{Registriernummer} \rangle = \langle \text{Länderkode} \rangle \langle \text{Registrierkode} \rangle$

Jedes Dokumentationssystem im Krankenhaus ist somit bundesweit identifizierbar. Beim Wechsel eines Dokumentationssystems innerhalb eines Krankenhauses ist zu beachten, dass eine neue Registriernummer zu beantragen ist.

Beispiel:

**Registriernummer:**

**HH1234a**



Die Adressen der einzelnen Landesgeschäftsstellen Qualitätssicherung (LQS) werden auf der Homepage [www.bqs-online.de](http://www.bqs-online.de) unter der Rubrik „Qualitätssicherung in den Ländern“ aufgeführt.

### 6.1.2 Registrierung bei der BQS Bundesgeschäftsstelle Qualitätssicherung

Für die Direktverfahren ( $Q_{SBQS}$ ) muss ein Dokumentationssystem - ggf. zusätzlich - bei der BQS registriert werden.

Die Registriernummer ist analog zur Registriernummer eines Bundeslandes aufgebaut. Der Ländercode wird durch BQ ersetzt. Der Registriercode ist fünfstellig und besteht aus Ziffern und Kleinbuchstaben.

Hier gilt gleichermaßen, dass nur Dokumentationssysteme, welche BQS registriert sind, Datensätze übermitteln dürfen.

Beispiel:

bq1011a

Adresse und Ansprechpartner für die Registrierung im Direktverfahren werden auf der Homepage [www.bqs-online.de](http://www.bqs-online.de) unter der Rubrik „Fragen und Antworten“ aufgeführt.

## **6.2 Identifizierung von Datensätzen**

Die Vorgangsnummer kennzeichnet in eindeutiger Weise jeden dokumentierten Datensatz eines Dokumentationssystems, und zwar unabhängig vom angewandten Modul.

Im einfachsten Fall könnte also die Vorgangsnummer um 1 erhöht werden, wenn ein neuer Datensatz angelegt wird. Wenn während eines stationären Aufenthaltes zwei QS-Dokumentationen eines Krankenhausfalls angelegt werden, so müssen auch unterschiedliche Vorgangsnummern vergeben werden.

Insbesondere ist es falsch, einfach eine Patientenidentifikationsnummer oder die Krankenhaus-Fallnummer zu verwenden bzw. zu pseudonymisieren<sup>52</sup>. Bei der Umsetzung hat der Softwareanbieter weit gehende Freiheit, vorausgesetzt, dass die modulübergreifende Eindeutigkeit der Vorgangsnummer gewährleistet ist.

Die Vorgangsnummer, welche als Zusatzfeld mit jedem Teildatensatz an die entgegennehmende Stelle übermittelt wird, darf für die Datenstelle nicht personenbeziehbar sein. In der Vorgangsnummer sollte z.B. nicht das Geburtsdatum enthalten sein.

Die QS-Dokumentationssoftware verwaltet jahrgangsübergreifend die Vorgangsnummern der QS-Dokumentationen. Sie soll dem Krankenhaus eine Zuordnung der Vorgangsnummern zu krankenhausinternen Fall- oder Patientennummern (vgl. nicht übermitteltes Datenfeld IDNRPAT) ermöglichen.

*Annahme oder Ablehnung von unterschiedlichen Versionen eines Datensatzes*

Bei der entgegennehmenden Stelle eingehende Datensätze werden anhand der Kombination aus Registriernummer und Vorgangsnummer als *ein Vorgang* identifiziert. Der für den Vorgang gespeicherte Datensatz kann durch eine neuere Version (mit höherer Versionsnummer) überschrieben werden.<sup>53</sup>

Unterschiedliche Versionen eines Datensatzes müssen demselben Primärmodul<sup>54</sup> zugeordnet sein. Findet ein Wechsel des Primärmoduls statt, so ist ein neuer Datensatz mit unveränderter Vorgangsnummer immer abzulehnen. Tabelle 29 gibt einen detaillierten Überblick, unter welchen Voraussetzungen ein neuerer Datensatz einen alten Datensatz überschreiben darf.

**Achtung:**

Das Modul HTXFU ist ein Sekundärmodul zum Primärmodul HTX. Statt HTXFU darf also nicht der Minimaldatensatz übermittelt werden.

Tabelle 29: Regeln für die Annahme oder Ablehnung von unterschiedlichen Versionen eines Datensatzes

Art des alten Datensatzes	Art des neuen Datensatzes	Änderung Nummer des zugeordneten Primärmoduls	Aktion
MDS	MDS	=	überschreiben
MDS	MDS	<>	ablehnen
MDS	Primärmodul	<>	ablehnen
Primärmodul	MDS	<>	ablehnen
MDS	Primärmodul	=	überschreiben
Primärmodul	MDS	=	überschreiben
Primärmodul	Primärmodul	<>	ablehnen
Primärmodul	Primärmodul	=	überschreiben

**Zusammenfassung**

QS-Dokumentationen der Krankenhäuser sind über die Kombination aus Registriernummer und Vorgangsnummer bundesweit eindeutig identifizierbar.

<sup>52</sup> Denn für jeden Patienten können unter Umständen mehrere Datensätze angelegt werden.

<sup>53</sup> Ggf. ist der geänderte Datensatz mit einer neuen Versionsnummer zu übermitteln.

<sup>54</sup> Jedem bei der entgegennehmenden Stelle eingehenden Datensatz ist ein Primärmodul (Definition in Kapitel 2.1.2) zugeordnet. Auch dem Minimaldatensatz ist ein Primärmodul zugeordnet: Das im Bogenfeld ZUQSMODUL eingetragene Primärmodul.

### 6.3 Der Exportvorgang

Bei einem Exportvorgang wird von einem Dokumentationssystem eine Transaktionsdatei für die jeweilige datenentgegennehmende Stelle erzeugt.

Die Daten der zu exportierenden QS-Dokumentationen werden vom Dokumentationssystem in Exportdateien geschrieben und die entsprechenden Vorgänge (identifiziert durch Vorgangsnummern) im Dokumentationssystem als "exportiert" markiert.

Jeder Exportvorgang erfolgt gemäß dem jeweils gültigen Exportverfahren, dessen Version in Kapitel 5.4 näher erläutert ist.

Jeder Übermittlung an eine datenentgegennehmende Stelle ordnet das registrierte Dokumentationssystem eine *eindeutige Transaktionsnummer* zu. Die Eindeutigkeit der Transaktionsnummer muss innerhalb eines Erfassungsjahres hinweg gewährleistet sein.

#### Achtung:

Es ist insbesondere nicht erlaubt, für den erneuten Export korrigierter Datensätze die gleiche Transaktionsnummer wie für den ersten Export zu verwenden.

#### Nummernkreis der Transaktionsnummern

Die Transaktionsnummer ist als eindeutige, fortlaufende, positive und ganze Zahl zu wählen: 1, 2, 3 etc.

#### 6.3.1 Die Steuerdatei

Die Steuerdatei - das Inhaltsverzeichnis eines Exportvorganges - wird nach folgendem Schema benannt:

HEADER.<Transaktionsnummer><sup>55</sup>

#### Beispiel:

HEADER.1

HEADER.001

HEADER.0815

HEADER.88815

---

<sup>55</sup> Wenn die Transaktionsnummer Bestandteil des Dateinamens ist, so dürfen aus historischen Gründen bei Nummern führende Nullen vorangestellt werden: 001, 002... 099...

Die erste Zeile der Steuerdatei besteht aus folgenden Feldern:

- Verfahrenskennung `QSLQS` (für die Module der indirekten Verfahren) oder `QSBQS` (für die Module der Direktverfahren)
- Version des Datenexportverfahrens (vgl. Kapitel 5.4).
- Kennung der Software, mit der die Exportdateien erzeugt wurden,
- Datum und Uhrzeit der Erstellung der Exportdatei,
- Institutionskennzeichen des Absenders (Krankenhaus) und
- Registriernummer des Dokumentationssystems (Vergleiche Kapitel 6.1)
- Namenszeichen des Ansprechpartners zu Fragen der Datenübermittlung.

Jede nachfolgende Zeile spezifiziert eine Exportdatei und setzt sich aus folgenden Feldern zusammen:

- Modulbezeichnung (Attribut `name` in Tabelle `Modul`)
- Name des Teildatensatzes (Attribut `name` in Tabelle `Bogen`)
- Versionsnummer der gültigen Spezifikation (vgl. Kapitel 5.5)
- Name der Datei (zurzeit fest vorgegeben) und
- Anzahl der Datensätze in diesem Modul.

Die Felder sind durch Semikola voneinander getrennt. Jede Zeile wird durch `<CR><LF>` (ASCII 13, 10) abgeschlossen.

### Beispiele:

#### a) Steuerdatei des Verfahrens `QSLQS`:

```
QSLQS;12.0;12.0#SR1#QS-EasyDoc#7.5;11.11.2009
11:11:11;261800267;HH1234A;CV<CR><LF>
01/1;B;12.0;M01N1B.001;3<CR><LF>
01/2;B;12.0;M01N2B.001;2<CR><LF>
10/1;B;12.0;M10N1B.001;15<CR><LF>
12/1;B;12.0;M12N1B.001;14<CR><LF>
16/1;M;12.0;M16N1M.001;8<CR><LF>
16/1;K;12.0;M16N1K.001;9<CR><LF>
MDS;B;12.0;MMDSB.001;1<CR><LF>
```

#### b) Steuerdatei des Verfahrens `QSBQS`:

```
QSBQS;12.0;12.0#SR1#QS-Heartbreaker#10.0;11.11.2009
11:11:11;261800267;HH1234A;CV<CR><LF>
HCH;B;12.0;MHCHB.001;3<CR><LF>
HCH;O;12.0;MHCHO.001;5<CR><LF>
```

```
HCH;FU;12.0;MHCHFU.001;1<CR><LF>  
MDS;B;12.0;MMDSB.001;1<CR><LF>
```

### Achtung:

Es ist verboten, Exportdateien unterschiedlicher Verfahrenskennungen und unterschiedlicher Versionen in einer einzigen Transaktion zu übermitteln.

### *Softwarekennung*

Die Softwarekennung setzt sich aus folgenden Teilen zusammen:

- Version der BQS-Spezifikation für QS-Dokumentationssoftware, auf deren Basis die QS-Dokumentationssoftware entwickelt wurde
- Releasekennung der BQS-Spezifikation, auf deren Basis die QS-Dokumentationssoftware entwickelt wurde
- Name der QS-Dokumentationssoftware
- Releasekennung der QS-Dokumentationssoftware

Die Teile der Softwarekennung sind durch das Zeichen # voneinander getrennt:

```
<Version Spez.>#<Release Spez.>#<Name Software>#<Release Software>
```

### Beispiele:

```
12.0##QSEASYDOC#001
```

```
12.0#SR1#QSEASYDOC#099
```

Exporte, deren Softwarekennung nicht dem neuen Format genügt, sind fehlerhaft (Fehlerart STEUER):

### Standardisierter Fehlertext:

```
Softwarekennung in Steuerdatei fehlerhaft: Angabe im Format <Version Spez.>#<Release  
Spez.>#<Name Software>#<Release Software> erforderlich!
```



### 6.3.2 Erzeugung der Transaktionsdatei

Die Steuerdatei (Kapitel 7.3.1) und die zugehörigen Exportdateien (Kapitel 7.4.2) werden vor dem Export mit Hilfe eines pkzip 2.04g-kompatiblen Archivierungsprogramms komprimiert und anschließend mit Triple-DES verschlüsselt (vgl. Anhang B).

Das Krankenhaus erhält bei der Registrierung seiner Dokumentationssysteme von der zuständigen Landesgeschäftsstelle einen Verschlüsselungskode, der bei der Erzeugung der Archivdatei verwendet werden muss. Die Transaktionsdatei wird wie folgt benannt:

```
T-<Länder-Kode><Registrierkode>-<Transaktionsnummer>.ZIP.3DES
```

#### Beispiel:

```
T-HH1234A-001.ZIP.3DES
```

### 6.4 Export von Teildatensätzen

Beim Export einer QS-Dokumentation durch ein Dokumentationssystem werden die Inhalte der für den betreffenden Vorgang angelegten Teildatensätze in passende Exportdateien geschrieben. Jede Exportdatei ist einem in der Spezifikation definierten Teildatensatz (vgl. Tabelle `Bogen`) zugeordnet.

#### 6.4.1 Anonymisierung

Für die Übermittlung der Qualitätssicherungsdaten an die jeweilige Auswertungsstelle müssen personenbeziehbare Daten ausreichend anonymisiert werden. Die Anonymisierung betrifft alle Kalendertagesdaten.

Sie werden z.B. umgerechnet in die Differenz in Tagen zwischen dem Kalendertagesdatum und dem OP-Datum. Die Differenzen sind positiv anzugeben. Ob der Zeitraum prä- oder postoperativ liegt, ergibt sich aus der Bezeichnung des Zeitraums. Für das Entlassungsdatum wird zusätzlich das Quartal berechnet (z.B. "2/2003" für das zweite Quartal 2003.). Die Vorschriften der Anonymisierung von Bogenfeldern findet man in den Tabellen `Ersatz` und `ErsatzFuerFeld` (Kapitel 6.4.1.2).

##### 6.4.1.1 Ersatzfelder für den Datenexport

Ersatzfelder werden aus ein oder mehreren Feldern der Datenfeldbeschreibung berechnet. Mit einem Ersatzfeld verknüpfte Bogenfelder werden nicht exportiert, wenn sie nicht als `<bleibt>` gekennzeichnet wurden. Stattdessen werden ein oder mehrere Ersatzfelder ex-

portiert. Vorrangig dienen Ersatzfelder der Anonymisierung beim Datenexport. Die verwendeten Ersatzfelder sind in der Tabelle `ErsatzFeld` gespeichert.

Tabelle 30: Struktur der Tabelle `ErsatzFeld`

Feldname	Feldtyp	Bemerkung
<code>idErsatzFeld</code>	INTEGER	Primärschlüssel
<code>name</code>	TEXT	Technischer Name (muss eindeutig sein)
<code>bezeichnung</code>	TEXT	
<code>formel</code>	TEXT	Berechnungsformel der Ersatzfelder
<code>fkBasisTyp</code>	INTEGER	Obligatorischer Fremdschlüssel zum Basistyp
<code>fkSchluessel</code>	INTEGER	Optionaler Fremdschlüssel zum Schlüssel

#### Beispiel:

Das Ersatzfeld `postoperative Verweildauer` wird folgendermaßen berechnet:

$poopvwdauer = ENTLDATUM - OPDATUM$

### 6.4.1.2 Anonymisierungsvorschriften

Die Anonymisierung von Datenfeldern wird in der Tabelle `ErsatzFuerFeld` konfiguriert. Für die Programmierung der Exportfelder ist dieser Abschnitt weniger wichtig, da die Exportfelder direkt über die Abfrage `ExportFelderFürEinModul` bzw. die Tabelle `ExportFormat` ermittelt werden können (Kapitel 6.4.2).

Die Tabelle `ErsatzFuerFeld` (Tabelle 31) ordnet einem Feld (Tabelle `Feld`) oder Bogenfeld ein oder mehrere Ersatzfelder zu. Die über das Attribut `fkFeld` definierte Anonymisierung ist die Standardanonymisierung für alle Module. Sie kann jedoch durch eine modulspezifische Anonymisierung überschrieben werden: Wenn ein Ersatzfeld mit einem Bogenfeld (über Attribut `fkBogenFeld`) verknüpft ist, so wird statt des Bogenfeldes das berechnete Ersatzfeld in die Exportdatei des Teildatensatzes geschrieben.

Tabelle 31: Struktur der Tabelle `ErsatzFuerFeld`

Feldname	Feldtyp	Bemerkung
<code>idErsatzFuerFeld</code>	INTEGER	Primärschlüssel
<code>fkFeld</code>	INTEGER	Optionaler Fremdschlüssel zur Tabelle <code>Feld</code>
<code>fkBogenFeld</code>	TEXT	Optionaler Fremdschlüssel zur Tabelle <code>BogenFeld</code>

fkErsatzFeld	TEXT	Obligatorischer Fremdschlüssel zur Tabelle Er- satzFeld
parametrierbar	BOOL	

### Beispiele:

- Das Feld `ENTLDATEUM` (Modul 15/1) hat die Ersatzfelder *poopvwdauer*, *entlquartal* und *entlwochttag*<sup>56</sup>.
- Wenn das Ersatzfeld `<entfällt>` mit einem Bogenfeld verknüpft ist, so entfällt das Bogenfeld in der Exportdatei.
- Wenn das Ersatzfeld `<bleibt>` mit einem Bogenfeld verknüpft ist, so wird das Bogenfeld unverändert in die Exportdatei übernommen. Dieses Ersatzfeld ist nur als parametrierbare Verknüpfung sinnvoll (s.u.).

Was ist zu tun, wenn ein einziges Ersatzfeld sowohl über `fkFeld` als auch über `fkBogenFeld` definiert ist ("doppelte Definition")?

In diesem Fall hat die spezielle Anonymisierung (über `fkBogenFeld`) Vorrang, die allgemeinen Anonymisierungen (`fkFeld`) werden ignoriert. Die allgemeine Definition kommt nur in den Modulen zur Anwendung, in denen keine spezielle Anonymisierung vorliegt.

### Beispiel:

Dem Feld `ENTLDATEUM` sind für alle Module der indirekten Verfahren über das Attribut `fkFeld` die Ersatzfelder *vwdauer*, *entlwochttag* und *entlquartal* zugewiesen. Eine Ausnahme bildet das Modul 16/1. Hier kann in einigen Bundesländern das Entlassungsdatum auch direkt übermittelt werden. Daher sind in `fkBogenFeld` das parametrierbare Ersatzfeld `<bleibt>` und die anonymisierten Ersatzfelder *vwdauer*, *entlwochttag* und *entlquartal* definiert.

### *Parametrierung*

Die Verknüpfung zwischen Feld bzw. Bogenfeld und Ersatzfeld kann parametrierbar werden (Attribut `parametrierbar`). Parametrierbare Ersatzfelder erscheinen immer als eigene Spalte in der Exportdatei. Es ist aber über die Dokumentationssoftware konfigurierbar, ob

---

<sup>56</sup> z.B. in Hessen

die Werte auch tatsächlich exportiert werden. Auf diese Weise werden spezifische Erfordernisse zum Datenschutz auf der Landesebene berücksichtigt.

Beispiel:

Im Modul 16/1 gibt es für das Feld `PLZ` (Postleitzahl) drei parametrierbare Ersatzfelder:

`<bleibt>`, `PLZ3stellig` und `PLZ4stellig`

In manchen Bundesländern dürfen Postleitzahlen unverändert exportiert werden, in anderen gibt es schärfere Anforderungen, so dass nicht alle Stellen einer Postleitzahl exportiert werden.

Achtung:

Die Übermittlung von parametrierbaren Exportfeldern wird von der jeweils zuständigen Landesebene bzw. der Bundesebene bei Direktverfahren festgelegt. Wenn keine Festlegung getroffen worden ist, so bleiben die parametrierbaren Exportfelder leer<sup>57</sup>.

*Ersatzfelder, die nicht berechnet werden können*

Es kann auch vorkommen, dass Ersatzfelder für einen Datensatz nicht berechnet werden können, weil die der Berechnung zu Grunde liegenden Bogenfelder nicht ausgefüllt (`LEER`) sind. Folgende allgemeine Regeln gelten:

1. Wenn die Bogenfelder, aus denen ein Ersatzfeld berechnet wird, dokumentiert sind (`<>` `LEER`), so ist das entsprechende Ersatzfeld zu berechnen und zu exportieren.
2. Wenn eines der beteiligten Bogenfelder nicht ausgefüllt ist und somit auch kein Ersatzfeld berechnet werden kann, so wird kein Wert für das Ersatzfeld exportiert (bleibt `LEER`).

## 6.4.2 Aufbau der Exportdatei

Die Struktur der Exportdateien ergibt sich im Wesentlichen aus der Datenfeldbeschreibung der einzelnen Module und den oben beschriebenen Anonymisierungsmaßnahmen zur Erfüllung der datenschutzrechtlichen Voraussetzungen.

Die Exportdateien sind wie folgt aufgebaut:

---

<sup>57</sup> Die entsprechenden Exportfelder existieren zwar inkl. Feldnamen in der Exportdatei, die Werte werden jedoch nicht eingetragen.

- ASCII-Format, erweiterter IBM-PC-Zeichensatz (8 Bit, DOS/OS/2),
- die erste Zeile enthält die Feldnamen, die nachfolgenden Zeilen enthalten die Werte der Exportfelder eines Teildatensatzes,
- Zeilen werden durch <CR><LF> (ASCII 13, 10) abgeschlossen,
- Feldnamen in der ersten und Werte der Exportfelder in den folgenden Zeilen werden durch Semikola getrennt und können variabel lang sein<sup>58</sup>.

### *Felder der Exportdatei*

Einen Überblick über die zu exportierenden Felder eines Moduls liefert die Abfrage *ExportFelderFürEinModul*. Am Anfang eines jeden Teildatensatzes findet man die Zusatzfelder. Danach folgen die Bogenfelder des Teildatensatzes, wobei die zu anonymisierenden Bogenfelder durch die Ersatzfelder ersetzt sind.

### *Format der Exportfelder*

In Abhängigkeit vom Basistyp des Exportfeldes werden die Werte der Exportfelder exportiert. Die Basistypen, deren Literale in Regeln von doppelten Anführungszeichen oder Hochkommata umschlossen werden, werden ohne diese Zeichen exportiert (TEXT, SCHLUESSEL, DATUM, MONDATUM, JAHRDATUM und UHRZEIT). Die Formatanweisungen entsprechen (bis auf die Anführungszeichen und Hochkommata) den in Tabelle 18 aufgelisteten Beispielen.

Folgende Grundsätze für das Format von Werten der Exportfelder sind nachfolgend noch einmal explizit aufgelistet:

- Semikola dürfen innerhalb eines Exportfeldes nicht vorkommen,
- Dezimaltrennzeichen ist das Komma, es werden keine Tausendertrennzeichen verwendet,
- Datumentrennzeichen ist der Punkt (Datumsangaben sind immer 10-stellig)
- Uhrzeittrennzeichen ist der Doppelpunkt,
- Die Codes nach ICD-10-GM und OPS werden mit Trennzeichen übermittelt.

### Beispiel:

Dieses Beispiel zeigt den Anfang einer Exportdatei für das Modul 01/1. Es sind nur die ersten 11 Felder dargestellt:

```
RegistrierNr;Vorgangsnr;VersionNr;Storno;Modul;Bogen;DokAbschlDat;IKNRKH;BSNR;FACHABT;gebjahr;.. <CR><LF>
HH1234A;123456789;1;;01/1;B;31.01.2007;260100023;1;1700;1989;.. <CR><LF>
```

### *Export von Listenfeldern*

Alle Elemente von Listenfeldern (Kapitel 2.1.5) werden exportiert, wobei die Nummer des Listenfeldes im Namen des Exportfeldes an den Namen des Listenfeldes angehängt wird (inkl. vorangestelltem Unterstrich)<sup>59</sup>.

#### Beispiel:

Das Listenfeld `SSRISIKO` hat in 16/1 die Exportfelder `SSRISIKO_1`, `SSRISIKO_2`, ..

### *Zusatzfelder des Datenexports*

Zusatzfelder, welche nicht in der Datenfeldbeschreibung (Tabelle `BogenFeld`) eines Moduls enthalten sind, werden von der QS-Dokumentationssoftware ausgefüllt<sup>60</sup>.

Einige der in der Tabelle `ZusatzFeld` definierten Zusatzfelder werden nachfolgend erläutert:

- Das übertragene Speicherdatum `DokAbschlDat` (*Datum des Dokumentationsabschlusses bzw. der Freigabe des Datensatzes für den Export*) ist nicht Teil der Datenbank für Auswertungen und wird nur für organisatorische Zwecke verwendet.
- Die *Versionsnummer* (`VersionNr`) gibt an, die wievielte Version des Datensatzes übertragen wird.

#### Erläuterung:

In der Regel wird die Versionsnummer 1 lauten. D.h., dass der nach dem ersten Dokumentationsabschluss freigegebene Datensatz übertragen wird. Muss ein korrigierter Datensatz erneut eingesandt werden, so muss die Versionsnummer vom dokumentierenden System um eins erhöht werden. Die neue Version des Datensatzes wird bei der Entgegennahme geprüft und überschreibt bei Korrektheit die alte Version des Datensatzes.

#### Achtung:

Wenn die entgegennehmende Stelle einen Datensatz mit derselben Versionsnummer ein zweites Mal erhält, so wird dieser zurückgewiesen.

---

<sup>59</sup> Auch Listebogenfeldern mit nur einem Element wird die Nummer angehängt, da die Listeneigenschaft im Attribut `Feld.istListe` festgelegt wird.

<sup>60</sup> Hier gilt also nicht der Grundsatz, dass Felder nicht vorbelegt sein dürfen.

- Der Eintrag 1 im Zusatzfeld `Storno` veranlasst die datenentgegennehmende Stelle, den übermittelten Datensatz einschließlich seiner Vorversion(en) als „storniert“ zu kennzeichnen.
- Das Zusatzfeld `IdBogenFeldMutter` wird bei Teildatensätzen eingefügt, welche einen mehrfach anlegbaren Mutterteildatensatz (Attribut `fkBogenZahl = '*'` oder `'+'`) haben. In diesem Fall wird die identifizierende Nummer des Mutterteildatensatzes (konfiguriert über `Bogen.fkEindeutigBogenFeld`) im Kindteildatensatz übermittelt.

**Beispiel:**

Beim Übermitteln des Teildatensatzes `21/3:KORO` muss als `IdBogenFeldMutter` der Wert des Bogenfeldes `LFDNREINGRIFF` des Mutterteildatensatzes `21/3:PROZ` eingetragen werden.

Eine vollständige Liste der möglichen Zusatzfelder findet sich in der Tabelle `ZusatzFeld` der Spezifikationsdatenbank.

***Benennung der Exportdateien***

Exportdateien sollen nach folgendem Schema benannt werden<sup>61</sup>:

`M<Modulname><Teildatensatzname>.<Transaktionsnummer>`

**Erläuterung:**

- `<Modulname>` ist der Name des Moduls (Attribut `name` der Tabelle `Modul`), wobei der Schrägstrich durch `N` ersetzt wird
- `<Teildatensatzname>` ist in Attribut `name` der Tabelle `Bogen` zu finden.
- `<Transaktionsnummer>`: `1, 2, .., 99, 100, .., 999, 1000, 1001, ..`

**Beispiele:**

`M16N1M.001`

`M16N1K.001`

`M15N1B.001`

---

<sup>61</sup> Im Gegensatz zu früheren Jahren können auch Dateinamen verwendet werden, welche nicht der 8.3-Konvention entsprechen. Im Modul 18/1 ergibt sich zum Beispiel für die Exportdatei des Teildatensatzes `BRUST` ein längerer Dateiname.

MHCHO.001

oder

M16N1M.1

M16N1K.1

M15N1B.1

MHCHO.1

## **6.5 Regeln für die Entgegennahme von Datensätzen und Teildatensätzen**

### **6.5.1 Prüfungen und Datenrückbestätigung**

Wenn der Datensatz eines Vorgangs aus mehreren Teildatensätzen besteht, so wird der Datensatz des Moduls akzeptiert, wenn alle erforderlichen Teildatensätze fehlerfrei sind.

Die entgegennehmende Stelle kann nach der Prüfung eines Datensatzes zu folgenden Ergebnissen (*Bestätigungsstatus*) kommen:

- OK  
Der Datensatz ist fehlerfrei.
- FEHLER  
Der Kerndatensatz enthält Fehler und ggf. der optionale Datensatz bzw. ein Storno schlägt fehl.
- STORNO  
Der Datensatz (inkl. aller Teildatensätze) wurde von der daten-entgegennehmenden Stelle storniert.

#### Achtung:

*Bei jeder Übermittlung vom Krankenhaus an die entgegennehmende Stelle muss der gesamte Datensatz<sup>62</sup> neu übermittelt werden (auch wenn er vorher schon bestätigt worden ist)!*

*Wurden Datensätze mit OK bestätigt, so muss die Erfassungssoftware auch die Möglichkeit anbieten, eine nachträgliche Aktualisierung zu übermitteln<sup>63</sup>.*

---

<sup>62</sup> Der komplette Datensatz mit allen Teildatensätzen

<sup>63</sup> Anwender haben berichtet, dass einzelne Softwareprodukte diese Funktionalität nicht bereitstellen



### *Definitionen der Fehlerfreiheit von Teildatensätzen und Datensätzen*

- Ein Teildatensatz ist dann fehlerfrei, wenn keines seiner Bogenfelder eine Wertebereichsverletzung aufweist und keine der verletzten Plausibilitätsregeln des Moduls einen Bezug zu einem Bogenfeld des Teildatensatzes hat<sup>64</sup>.
- Der Kerndatensatz ist dann fehlerfrei, wenn seine Teildatensätze fehlerfrei sind.
- Der optionale Datensatz ist dann fehlerfrei, wenn seine Teildatensätze fehlerfrei sind.

*Folgende Prüfungen beziehen sich auf den Exportdatensatz und werden von den datenentgegennehmenden Stellen durchgeführt:*

*Prüfung auf Lesbarkeit und Virenfreiheit* des Datenträgers und der übermittelten Transaktionsdateien. Hier erfolgt keine Protokollierung in Form von Protokoll- und Fehlerdateien, sondern der Absender erhält eine entsprechende formlose Nachricht.

Die *formale Prüfung der Steuerdatei* (Kapitel 6.3.1) umfasst die Korrektheit des Formats.

Die *formale Prüfung der Exportdateien* (Kapitel 6.4.2) umfasst die Existenz der in der Steuerdatei aufgelisteten Exportdateien und die Korrektheit des Formats.

Die *Vollständigkeits- und Versionsprüfung* bezieht sich auf einen übermittelten Datensatz, der durch Registriernummer, Vorgangsnummer und Versionsnummer identifizierbar ist:

- Zulässigkeit der Versionsnummer: wurde der Datensatz bereits übermittelt, muss die Versionsnummer des neu übermittelten Datensatzes größer als die des bereits gespeicherten Datensatzes sein.
- Teildatensätze mit dem Attribut `fkBogenZahl + oder 1` müssen (bezogen auf den Mutterteildatensatz) mindestens einmal oder genau einmal in einem Datensatz existieren (Fehlerart `TDS` in Kapitel 6.6.2).
- Für mehrfach angelegte Teildatensätze mit der Kardinalität `+ oder *` (Attribut `fkBogenZahl Tabelle Bogen`) muss die Eindeutigkeit des Feldes `fkEindeutigBogenFeld` gewährleistet sein (Fehlerart `TDS` in Kapitel 6.6.2).
- Für jeden Teildatensatz mit mehrfach anlegbaren Mutterteildatensätzen ist zu prüfen, ob ein Mutterteildatensatz existiert, für den die Inhalte des Attributes `IdBogenFeld-`

---

<sup>64</sup> gilt auch für teildatensatzübergreifende Regeln vgl. 3.5

Mutter und des identifizierenden Attributes des Mutterteildatensatzes (definiert durch `fkEindeutigBogenFeld` in der Tabelle `Bogen`) gleich sind (Kapitel Fehlerart TDS in Kapitel 6.6.2).

#### *Prüfungen auf doppelte Datensätze*

Ein übermittelter Datensatz wird von der datenentgegennehmenden Stelle nicht angenommen, wenn bereits vorher ein anderer Datensatz mit derselben Registriernummer, Vorgangsnummer *und Versionsnummer* übermittelt worden ist.

Die Wertebereichsüberprüfungen einschließlich Muss-/Kann-Prüfung und die Plausibilitätsprüfungen (Kapitel 3) werden für jeden übermittelten Datensatz durchgeführt.

### **6.5.2 Prüfung der warnenden Plausibilitätsregeln**

Bei der Datenentgegennahme werden neben den harten Plausibilitätsprüfungen auch die warnenden Plausibilitätsprüfungen (Wert D in Attribut `fkRegelTyp` der Tabelle `Regeln`, Kapitel 3.1.3) durchgeführt.

Warnungen werden genauso wie harte Plausibilitätsverletzungen in der Fehlerdatei (inkl. Fehlermeldung, Regelnummer und Regeltyp) an das Krankenhaus übermittelt. Sie sind unterscheidbar anhand des neuen Feldes `Regeltyp` in der Fehlerdatei (Kapitel 6.6.2).

Warnungen führen nicht zur Ablehnung von Datensätzen durch die Datenstelle.
---

#### Beispiel:

Wenn ein Datensatz keine harte Plausibilitätsprüfung, aber ein oder mehrere warnende Plausibilitätsprüfungen („Warnhinweis“) verletzt, so wird er von der datenentgegennehmenden Stelle angenommen und bestätigt.

Für diesen Datensatz werden dem Krankenhaus Warnungen übermittelt, so dass das Krankenhaus die Möglichkeit hat, einen korrigierten Datensatz zu übermitteln.

### 6.5.3 Stornierung von Datensätzen

Datensätze können storniert werden, damit der Vollständigkeitsabgleich der gelieferten Datensätze korrekt geführt werden kann.

Dazu ist das Zusatzfeld `storno = 1` zu setzen. Die datenentgegennehmende Stelle wird dadurch veranlasst, den betreffenden Datensatz einschließlich aller Vorversionen und Teildatensätze als „storniert“ zu kennzeichnen. Der Stornovorgang wird in der Datenbestätigung protokolliert.

#### Achtung:

Der mit dem Storno-Zusatz gekennzeichnete Datensatz muss ebenfalls eine um eins erhöhte Versionsnummer enthalten, um die Stornierung unabhängig von der Reihenfolge der Verarbeitung von Datensätzen sicherzustellen. Ein Storno mit einer bereits verwendeten Versionsnummer wird zurückgewiesen (Bestätigungsstatus `FEHLER`, Fehlerart `DOPPELT`). Ein Stornoversuch eines noch nicht übermittelten Datensatzes wird ebenfalls zurückgewiesen.

### 6.6 Die Antwortdateien

Die Antwortdatei ist ein mit *Triple-DES* (3DES) verschlüsseltes ZIP-Archiv und wird wie folgt benannt:

```
A-<Länder-Kode><Registrierkode>-<Transaktionsnummer>.ZIP.3DES
```

#### Beispiel:

```
A-BW1234a-001.ZIP.3DES
```

Das unverschlüsselte ZIP-Archiv enthält die Protokolldatei und die Fehlerdatei, welche von der datenentgegennehmenden Stelle als Antwort auf eine eingegangene Transaktionsdatei generiert werden, und hat einen Namen nach folgendem Schema:

```
A-<Länder-Kode><Registrierkode>-<Transaktionsnummer>.ZIP
```

#### Beispiel:

A-BW1234a-001.ZIP

## 6.6.1 Die Bestätigungsdatei

Die Bestätigungsdatei enthält für jeden übermittelten Datensatz das Ergebnis der Prüfung auf Plausibilität (vgl. Kapitel 6.5.1 und Kapitel 6.5.2) und einen Namen nach folgendem Schema:

```
B<Länder-Kode><Registrierkode>.<Transaktionsnummer>
```

### Beispiel:

**B**BW1234a.001

Der einzelne Datensatz kann über die Kombination von Registriernummer, Vorgangsnummer und Versionsnummer eindeutig reidentifiziert werden. Im Fehlerfall (*Bestätigungsstatus FEHLER*) werden die Fehler in einer separaten Datei (Kapitel 6.6.2) aufgelistet:

```
<Re-  
gistrierNr>;<VorgangsNr>;<VersionNr>;<Modul>;<SPEZVERSION>;<BestätigungsStatus><CR  
><LF>
```

mit:	
{ }	Wiederholung
[ ]	Option
<CR>	Carriage Return (ASCII 13)
<LF>	Line Feed (ASCII 10)
<BestätigungsStatus>	Werte sind OK, STORNO oder FEHLER
<RegistrierNr>	Registriernummer des Dokumentationssystem
<VorgangsNr>	Vorgangsnummer
<VersionNr>	
<Modul>	Name des übermittelten Moduls (Attribut Modul.name)
<SPEZVERSION>	Version der gültigen Spezifikation (siehe Attribut Version.name )

### Beispiel:

```
HH1234A;261012309;1;21/3;12.0;FEHLER<CR><LF>
HH1234A;261012309;1;16/1;12.0;OK<CR><LF>
```

## 6.6.2 Die Fehlerdatei

Die Erfassungssoftware muss in der Lage sein, die Fehlermeldungen und Warnungen der datenentgegennehmenden Stelle einzulesen und den Anwender durch eine möglichst komfortable Nachbearbeitungsfunktion für die betreffenden Datensätze zu unterstützen.<sup>65</sup>

Die Fehlerdatei hat einen Namen nach folgendem Schema:

```
F<Länder-Kode><Registrierkode>.<Transaktionsnummer>
```

### Beispiel:

FBW1234a.001

Voraussetzung für diese Funktionalität ist, dass von der datenentgegennehmenden Stelle die Fehlermeldungen in einer Fehlerdatei übermittelt werden.

Die Zeilen der Fehlerdatei sind folgendermaßen aufgebaut:

```
<Re-
gistrierNr>; [<Modul>]; [<VorgangsNr>]; [<VersionNr>]; [<SPEZVERSION>]; <Fehlerart>; [<R
egelnr>]; [<Regeltyp>]; [<Liste>]; [<Meldung>] <CR><LF>
```

mit:

{ }	Wiederholung
[ ]	Option
<CR>	Carriage Return (ASCII 13)
<LF>	Line Feed (ASCII 10)
<Modul>	Modul
<RegistrierNr>	Registriernummer des Dokumentationssystem
<VorgangsNr>	Vorgangsnummer
<VersionNr>	Version des Spezifikation
<SPEZVERSION>	Version des Spezifikation
<Fehlerart>	STEUER = Formatfehler der Steuerdatei EXPORT = Formatfehler der Exportdatei

<sup>65</sup> Besonders wichtig, falls die Erfassungssoftware die Plausibilitätsprüfungen nicht vollständig umgesetzt hat!

	DOPPELT = bereits vorhandener Datensatz wird erneut übermittelt
	TDS = Vollständigkeit und Version der Teildatensätze
	WERT = Wertebereichsverletzung
	REGEL = Plausibilitätsverletzung
<Regelnr>	Nummer der Regel (idRegeln in Tabelle Regeln), nur wenn Fehlerart=REGEL vergeben, ansonsten leer
<Regeltyp>	nimmt die Werte H (=hart) oder D (=Warnung)
<Liste>	In Abhängigkeit von der Fehlerart entweder Liste von Teildatensätzen oder von Bogenfeldern
<Meldung>	Fehlermeldung als Freitext, z.Z. nicht fest vorgegeben

Für jeden aufgetretenen Fehler wird eine eigene Zeile angehängt!

Die Fehlerarten beziehen sich auf die in Kapitel 6.5.1 und Kapitel 6.5.2 beschriebenen Prüfungen. Es gibt für jeden Fehler auszufüllende Spalten bzw. Felder (<RegistrierNr> und <Fehlerart>). Andere Spalten bleiben bei einzelnen Fehlerarten leer. Z.B. braucht bei einem Fehler der Steuerdatei nur die Fehlermeldung übermittelt werden. Tabelle 32 gibt einen Überblick darüber, unter welchen Bedingungen in den Feldern der Fehlerdatei Angaben erforderlich sind.

Die *Bogenliste* umfasst ein oder mehrere Namen von Teildatensätzen, welche einen Bezug zu einer Regel haben. Entscheidend für den Bogenbezug sind die in der Tabelle Regeln formulierten Regeln, nicht die für den Exportdatensatz umformulierten Regeln.

Die *Bogenfeldliste* umfasst ein oder mehrere Namen von Bogenfeldern, welche einen Bezug zum Fehler haben. Bei der Fehlerart WERT enthält die Liste nur ein Element. Der Bogenfeldname umfasst auch den Namen des zugehörigen Teildatensatzes<sup>66</sup> (Beispiele: K[2].FLDOSISKORO, B.AUFNDATUM).

Für jede Regel gibt es eine Liste von Bogenfeldern (identifiziert über die Feldnamen der Regeln). Damit die Liste nicht durch Parsen ermittelt werden muss, wird sie auch über die Tabelle RegelFelder zur Verfügung gestellt. Über die Regelnummer können die Teildatensätze, welche Bezug zu einer Regel haben, durch folgende Abfrage identifiziert werden:

```
SELECT DISTINCT Bogen.name FROM (Modul INNER JOIN (Feld INNER JOIN (Bogen INNER
JOIN BogenFeld ON Bogen.idBogen = BogenFeld.fkBogen) ON Feld.idFeld = Bogen-
Feld.fkFeld) ON Modul.idModul = Bogen.fkModul) INNER JOIN RegelFelder ON Bogen-
```

---

<sup>66</sup> Der Bezug zum Modul kann entfallen, da es über die Vorgangsnummer identifiziert werden kann.

Feld.idBogenFeld = RegelFelder.fkBogenFeld WHERE RegelFelder.fkRegeln=<Regelnummer>;

Für das Feld <Regelnummer> ist die entsprechende Nummer (Attribut idRegeln) der Tabelle Regeln anzugeben.

Bei Teildatensätzen, welche mehrfach angelegt werden können, muss die Nummer des betreffenden Teildatensatzes in eckigen Klammern angehängt werden (z.B. P[1], P[2] usw.) Mit Nummer des betreffenden Teildatensatzes ist der Inhalt des in der Tabelle BogenFeld unter „fkEindeutigBogenFeld“ definierten Feldes gemeint.

### Beispiel:

Die Regel 2044 der Spezifikation 12.0 hat den Bezug zu den Teildatensätzen M und K des Moduls 16/1. Da der Teildatensatz K mehrfach angelegt werden kann, muss in der Bogenliste auch der betreffende Teildatensatz angegeben werden, z.B. M|K[1].

Tabelle 32: Ausfüllen der Felder der Fehlerdatei in Abhängigkeit von den Fehlerarten (Minuszeichen bedeutet "kein Wert" bzw. Nicht-Ausfüllen)

Fehlerart	<Modul>	<Vorgangsnr>/<VersionNr>	<SPEZVERSION>	<Regelnr>	<Regeltyp>	<Liste>	<Meldung>
STEUER	-	-	-	-	-	-	ja
EXPORT	ja	-	ja	-	-	<Bogen>	ja
DOPPELT	ja	ja	ja	-	-	-	ja
TDS	ja	ja	ja	-	-	<Bogenliste>	ja
WERT	ja	ja	ja	-	-	<Bogenfeldliste> <sup>67</sup>	ja
REGEL	ja	ja	ja	ja	ja	<Bogenliste>	ja

### Beispiel einer Fehlerdatei:

```
XY10012;17/5;62;1;12.0;WERT;;B.POSTOPEXFLEXKNIE1;Der Wert '370' des Datenfeldes
17/5:B.POSTOPEXFLEXKNIE1 "Extension/Flexion 1" (Zeile 40) ist größer als
'10'<CR><LF>
```

<sup>67</sup> In der Regel wird hier nur ein Bogenfeld aufgeführt. Ausnahme ist, wenn Kombinationsfelder geprüft werden: ENTLDIAG|ENTDIAGVERS u. a.

XY10012;16/1;69;1;12.0;REGEL;2171;K[1];operativer Entbindungsmodus kodiert ohne Indikationen zur operativen Entbindung<CR><LF>  
 XY10012;16/1;69;1;12.0;REGEL;2175;K[1];Extraktion bzw. Beckenendlage-Entbindung trotz Schädellage<CR><LF>  
 XY10012;16/1;69;1;12.0;REGEL;2166;K[1];Extraktion bzw. Beckenendlage-Entbindung trotz anderer Lageangabe<CR><LF>

### *Standardisierung der Meldungen bei Bestätigungsstatus mit Fehlerart **DOPPELT***

*Es wurde bereits ein anderer Datensatz mit derselben Registriernummer und Versionsnummer übermittelt.*

### *Standardisierung der Meldungen bei der Fehlerart **WERT***

Bei feldbezogenen Fehlern sind die standardisierten Fehlermeldungen aus Kapitel 3.2 zu verwenden.

### *Standardisierung der Meldungen bei Bestätigungsstatus mit Fehlerart **TDS***

- Wenn ein obligatorischer Teildatensatz (Attribut `Bogen.fkBogenZahl` ist + oder 1, oder ein zu einem Kindteildatensatz zugehöriger Mutterteildatensatz) eines Vorganges in den Exportdateien einer Transaktion nicht vorkommt, so ist folgende Fehlermeldung auszugeben:

*Erforderlicher Teildatensatz <Bogen.name> ("**<Bogen.bezeichnung>**") existiert nicht.*

- Wenn die Existenzbedingung eines Kindteildatensatzes im zugehörigen Mutterteildatensatz erfüllt ist, aber kein Kindteildatensatz vorhanden ist, so ist folgende Fehlermeldung auszugeben (Kapitel 2.1.3):

*Die Angaben im Datensatz erfordern einen Teildatensatz <Bogen.name> ("**<Bogen.bezeichnung>**"). Dieser fehlt.*

- Wenn die Existenzbedingung eines Kindteildatensatzes im zugehörigen Mutterteildatensatz nicht erfüllt ist, aber trotzdem ein Kindteildatensatz existiert, so ist folgende Fehlermeldung zu erzeugen (Kapitel 2.1.3):

*Die Angaben im Datensatz lassen keinen weiteren Teildatensatz <Bogen.name> ("**<Bogen.bezeichnung>**") zu, obwohl ein solcher übermittelt wurde.*

Dieser Fehler kann auf zwei Arten hervorgerufen werden: (a) im TDS `21/3:PROZ` wird ein TDS `21/3:PCI[1]` erwartet, aber ein `21/3:KORO[1]` geliefert; (b) es wird ein TDS `21/3:PCI[1]` erwartet, aber **zwei** TDS `21/3:PCI[1]` werden geliefert (gleicher Eintrag in `fkEindeutigBogenFeld`).



## 6.7 **Sonstiger Dateitransfer zwischen Krankenhaus, LQS und BQS**

Transaktions- und Antwortdateien dürfen ausschließlich transaktionsbezogene Dateien enthalten. Um auch sonstige, z.B. von der datenentgegennehmenden Stelle initiierte Dateitransfers in *verschlüsselter* Form durchführen zu können, werden *optional* folgende Konventionen definiert.

Beliebige Dateien werden in einem Zip-Archiv mit dem Namen

N-<Länder-Kode><Registrierkode>-<Inhaltsbezeichnung>.ZIP

verpackt und nach der Triple-DES-Verschlüsselung als

N-<Länder-Kode><Registrierkode>-<Inhaltsbezeichnung>.ZIP.3DES

an den Partner verschickt („N“ steht für „Nachrichten“). Dieses Verfahren kann in beiden Richtungen eingesetzt werden (von LQS zu Krankenhaus und umgekehrt). Es gelten keine Konventionen für den Inhalt des Zip-Archivs. Mit diesem Verfahren können auch nicht spezifizierte Dateien zwischen den einzelnen Institutionen verschlüsselt übermittelt werden. Hierbei ist der Datenschutz zu beachten!

### Beispiel:

N-HH1234a-Importübersicht-2009-08-01.ZIP.3DES

Dateien im Zip-Archiv:

Importübersicht-2009-08-01.xls Newsletter-2009-08.rtf
--

# ANHANG

## A Hinweise zur Implementierung von Funktionen

### Pseudocode für Funktionen:

Als Hilfestellung für die Ausprogrammierung wird bei manchen Funktionen Pseudocode bereitgestellt. Der Pseudocode ergänzt die Syntax der Plausibilitätsregeln um folgende Sprachelemente<sup>68</sup>:

1. Befehlszeilen werden mit Semikolon abgeschlossen.
2. Wertzuweisungen mit dem Operator :=

```
A := B + C;
```

3. Auswahanweisungen

```
if (<Bedingung>){  
  ..  
} else {  
  ..  
}
```

Hinter <Bedingung> verbirgt sich ein logischer Ausdruck, welcher der Syntax der Plausibilitätsregeln gehorcht.

4. Blöcke werden durch geschweifte Klammern definiert:

```
{  
  ..  
}
```

5. Innerhalb einer Funktion sind die Argumentvariablen verfügbar.
6. Eine Variable, welche den gleichen Namen wie die Funktion hat, muss am Ende mit `return` zurückgegeben werden.

---

<sup>68</sup> Der Pseudocode erhebt nicht den Anspruch auf formale Korrektheit

## Beispiel:

### Funktion zur Berechnung des Gestationsalter im Modul 16/1 (Geburtshilfe)<sup>69</sup>

Die Funktion `gestAlter` berechnet das Gestationsalter in Tagen und hat folgende Signatur:

GANZEZAHL `gestAlter`(GANZEZAHL `abstGebterm`; GANZEZAHL `TRAGZEITKLIN`; SCHLUESSEL `SSRISIKO`)

Die Formel (Pseudocode) ist in der Tabelle `SyntaxFunktion` der Spezifikationsdatenbank hinterlegt:

```

if ((abstGebterm <> LEER) UND ((abstGebterm + 280) >= 98) UND ((abstGebterm + 280) <= 336))
{
  if (SSRISIKO EINSIN (38))
  {
    if ((TRAGZEITKLIN <> LEER) UND (TRAGZEITKLIN > 18) UND (TRAGZEITKLIN <= 44))
    {
      gestAlter := TRAGZEITKLIN * 7;
    }
    else
    {
      gestAlter := abstGebterm + 280;
    }
  }
  else
  {
    if ((TRAGZEITKLIN <> LEER) UND (TRAGZEITKLIN > 18) UND (TRAGZEITKLIN <= 44))
    {
      if ((ABS(TRAGZEITKLIN * 7 - (abstGebterm + 280)) < 14) UND ((abstGebterm + 280) >= 126) UND ((abstGebterm +
        280) <= 308))
      {
        gestAlter := abstGebterm + 280;
      }
      else
      {
        gestAlter := TRAGZEITKLIN * 7;
      }
    }
    else
    {
      gestAlter := abstGebterm + 280;
    }
  }
}
else
{
  if ((TRAGZEITKLIN <> LEER) UND (TRAGZEITKLIN > 18) UND (TRAGZEITKLIN <= 44))
  {
    gestAlter := TRAGZEITKLIN * 7;
  }
  else
  {
    gestAlter := LEER;
  }
}

return gestAlter;

```

**Achtung:** Die Funktion `gestAlter` kann den Wert `LEER` zurückliefern. Daher sind bei der Evaluation der Plausibilitätsregeln die Besonderheiten des Kapitels 3.6 zu beachten.

<sup>69</sup> Die Formel zur Berechnung des Gestationsalters wurde in der Spezifikation 12.0 überarbeitet.

## **B Erläuterungen zum Datenübermittlungsverfahren per E-mail**

Die im Rahmen des E-Mail-Verfahrens notwendige ZIP-Komprimierung und 3DES-Verschlüsselung wird mit Hilfe einer von der BQS entwickelten DLL (Dynamic Link Library) durchgeführt.

Download unter <http://www.bqs-online.com/download/QuantPackerLibrary-1.0.zip>.

Die weiteren technischen Details finden sich im ZIP-Archiv.

## C Besonderheiten der Qualitätssicherung in Hessen

Im Auftrag der GQH (Geschäftsstelle Qualitätssicherung Hessen bei der Hessischen Krankenhausgesellschaft e.V.) wurden die BQS-Spezifikationen für QS-Dokumentationssoftware und QS-Filter-Software ergänzt.

- 1) Die seit dem Jahre 2004 bundesweit im Modul 15/1 verpflichtend zu dokumentierenden gynäkologischen Eingriffe (Hysterektomien, Adnexeingriffe und Konisationen) decken ca. 55% aller stationär durchgeführten gynäkologischen Operationen ab. In Hessen ist seit 1997 die Totalerhebung in der operativen Gynäkologie verpflichtend umgesetzt. Diese Unterschiede im Umfang der Erhebung werden in der BQS-Spezifikation 12.0 berücksichtigt.
- 2) Die Übermittlung von echten Datumsangaben in den Modulen 15/1 und 18/1 wird ermöglicht.

### C.1 Anpassung des Datensatzes 15/1 für Totalerhebung gynäkologischer Operationen

Die BQS-Spezifikation 12.0 ermöglicht die Totalerhebung gynäkologischer Operationen in Hessen. Zu diesem Zweck wurde im Teildatensatz Operation (15/1:0) mit der Spezifikation 8.0 das folgende Datenfeld eingeführt:

*Eingriff im Rahmen der Totalerhebung (d.h. keine Hysterektomie, kein Adnexeingriff und keine Konisation)<sup>70</sup>*

Eine Angabe (=ja) in diesem Datenfeld ist ausschließlich in Hessen möglich. In allen anderen Bundesländern darf das Datenfeld nicht ausgefüllt werden.

*Das Datenfeld GYNTOTAL soll nur in Erfassungssystemen hessischer Krankenhäuser sichtbar sein, um Anwender aus Krankenhäusern anderer Länder vor Fehldokumentation zu bewahren.*

---

<sup>70</sup> Das Datenfeld heißt GYNTOTAL und hat den Schlüssel Jleer

**Achtung:**

- 1) Wurde für den Eingriff mindestens eine bundesweit verpflichtende Prozedur (definiert in der OPS-Liste GYN\_OPS der BQS-Spezifikation für QS-Filter-Software) erbracht, so muss das Datenfeld GYNTOTAL leer bleiben.
- 2) Wurden für den Eingriff ausschließlich hessenweit verpflichtende Prozeduren (definiert in der OPS-Liste GYN\_OPS\_HESSEN der BQS-Spezifikation für QS-Filter-Software) erbracht, so muss das Datenfeld GYNTOTAL mit einer 1 (=ja) dokumentiert werden.

**C.2 Übermittlung von 15/1er-Datensätzen an die Bundesebene**

Die Datenstelle der GQH leitet Teildatensätze 15/1:O nicht weiter, wenn sie ausschließlich im Rahmen der hessischen Totalerhebung dokumentiert worden sind:

- 1) Ein Datensatz 15/1 wird von der GQH an die Bundesebene übermittelt, wenn für mindestens einen Teildatensatz 15/1:O die Bedingung GYNTOTAL=LEER zutrifft. In diesem Fall werden der Teildatensatz 15/1:B sowie alle Teildatensätze 15/1:O übertragen, für die diese Bedingung zutrifft.
- 2) Datensätze 15/1, bei denen für alle Teildatensätze 15/1:O die Bedingung GYNTOTAL=1 zutrifft, werden nicht übermittelt.

Beispiele

Folgende Beispiele zeigen für drei Fälle, welche Teildatensätze einer Dokumentation an die Bundesebene weitergeleitet werden.

Fall 1

Dokumentierte Teildatensätze	GYNTOTAL=1	TDS an BQS weiterleiten?
15/1:B	-	<b>ja</b>
15/1:O (1. Eingriff)	ja	nein
15/1:O (2. Eingriff)	nein	<b>ja</b>

Fall 2

Dokumentierte Teildatensätze	GYNTOTAL=1	TDS an BQS weiterleiten?
15/1:B	-	nein
15/1:O (1. Eingriff)	ja	nein
15/1:O (2. Eingriff)	ja	nein

### Fall 3

Dokumentierte Teildatensätze	GYNTOTAL=1	TDS an BQS weiterleiten?
15/1:B	-	<b>ja</b>
15/1:O (1. Eingriff)	nein	<b>ja</b>

### C.3 Ist-Bescheinigung

Die GQH bescheinigt die Datensätze den hessischen Krankenhäusern getrennt nach den QS-Filter-Leistungsbereichen:

- **15/1 B** (= gelieferte Datensätze, welche an die Bundesebene weitergeleitet werden)
- **15/1 L** (= gelieferte Datensätze, welche nicht an die Bundesebene weitergeleitet werden)

#### Beispiel

Die Fälle 1 und 3 des letzten Beispiels gehören zum Leistungsbereich 15/1, der Fall 2 zum Leistungsbereich 15/1H.

### C.4 Übermittlung von Datumsangaben in den Exportdatensätzen 15/1 und 18/1

In den Exportdatensätzen 15/1 und 18/1 sind nachfolgend aufgelistete Datumsangaben als parametrierbare Ersatzfelder definiert:

15/1:B:GEBDATUM  
15/1:B:AUFNDATUM  
15/1:O:OPDATUM  
15/1:B:ENTLDATUM  
18/1:B:GEBDATUM  
18/1:B:AUFNDATUM  
18/1:BRUST:VORHISTODATUM  
18/1:O:OPDATUM  
18/1:B:ENTLDATUM

Die Festlegung, ob parametrierbare Exportfelder auch wirklich vom Krankenhaus an die Datenstelle übermittelt werden, wird – in Abstimmung mit den zuständigen Landesdatenschutzbeauftragten - von der jeweils zuständigen Landesebene getroffen<sup>71</sup>.

---

<sup>71</sup> Dieses gilt generell für alle parametrierbaren Exportfelder.