

# **Technische Dokumentation zur Spezifikation der einrichtungsbezoge- nen QS-Dokumentation**

**Spezifikationsjahr 2026**

# Informationen zum Bericht

## BERICHTSDATEN

---

**Technische Dokumentation zur Spezifikation der einrichtungsbezogenen QS-Dokumentation. Spezifikationsjahr 2026**

Datum der Veröffentlichung 28. Januar 2026

## AUFTAGSDATEN

---

Auftraggeber Gemeinsamer Bundesausschuss (G-BA)

## DATENQUELLEN UND BETRACHTETE ZEITRÄUME

---

QS-Dokumentationsdaten Spezifikationsjahr 2026

# Inhaltsverzeichnis

Tabellenverzeichnis.....	7
Abbildungsverzeichnis .....	9
Abkürzungsverzeichnis .....	10
Änderungsindex.....	12
Leseanleitung .....	13
1 Einleitung .....	15
1.1 Spezifikationsbegriff .....	15
1.2 Zielsetzung und Zielgruppe .....	16
1.3 Releaseplanung.....	16
A Prozesse .....	18
1 QS-Dokumentation.....	19
1.1 Auslösung .....	19
1.2 Erfassung .....	19
1.2.1 Gestaltung von Eingabemasken .....	19
1.2.2 Einrichtungsidentifizierende Daten .....	23
1.2.3 QS-Daten .....	24
1.2.4 Plausibilitätsprüfungen .....	24
1.3 Export der Daten .....	25
1.3.1 Erzeugen der Exportdatei .....	26
1.3.2 Datenprüfung.....	29
1.3.3 XML-Verschlüsselung.....	29
1.3.4 Ausgangsvalidierung .....	30
1.3.5 Beispiele für Exportdateien.....	30
1.4 Datenübermittlung .....	31
1.4.1 Registrierung .....	31
1.4.2 Dateibenennung.....	32
1.4.3 Datenversand via E-Mail .....	34
1.5 Pseudonymisierung der LE-Daten.....	34
1.6 Rückprotokollierung .....	36

1.6.1	Funktion von Empfangsbestätigung und Datenflussprotokoll im Datenfluss.....	36
2	Sollstatistik.....	41
B	Komponenten .....	42
1	QS-Filter.....	47
1.1	Anmerkungen zur Struktur der Spezifikationsdatenbank für QS-Filter .....	47
1.2	Grundlegende Tabellen der Datenbank .....	47
1.2.1	Module (Datensätze der QS-Dokumentation) .....	47
1.2.2	Struktur der Datensatzdefinitionen .....	48
1.2.3	Funktionen.....	53
1.2.4	OPS-Listen .....	55
1.2.5	ICD-Listen.....	55
1.2.6	FAB-Listen .....	56
1.2.7	EBM-Listen .....	56
1.2.8	Entgelt-Listen.....	56
1.2.9	Versionsverwaltung.....	56
1.2.10	Meta-Tabellen .....	57
1.2.11	DB-Änderungen gegenüber der Vorgängerversion .....	58
1.3	Der QS-Filter-Datensatz .....	58
1.3.1	Der QS-Filter-Eingangsdatensatz.....	58
1.4	Der Algorithmus zur Ermittlung der Dokumentationspflicht .....	61
1.4.1	Einleitung und Überblick.....	61
1.4.2	Verfahrensbezogene Einschlusskriterien.....	61
1.4.3	Administrative Einschlusskriterien.....	63
1.4.4	Struktur und Syntax der Auslösebedingungen.....	64
2	QS-Dokumentation.....	68
2.1	Anmerkungen zur Struktur der Spezifikation zur QS-Dokumentation.....	68
2.2	Datenfeldbeschreibung .....	69
2.2.1	Dokumentationsmodule (Datensätze) .....	70
2.2.2	Teildatensätze.....	71
2.2.3	Datenfelder (Bogenfelder).....	72
2.2.4	Überschriften.....	80

2.2.5	Ausfüllhinweise.....	80
2.3	Plausibilitätsprüfungen .....	81
2.3.1	Die Regelabelle.....	81
2.3.2	Regelsyntax .....	83
2.3.3	Funktionen.....	87
2.3.4	Syntaxvariablen .....	88
2.3.5	Einzelregeln .....	88
2.3.6	Teildatensatzübergreifende Regeln.....	88
2.3.7	Feldgruppenregeln.....	89
2.3.8	Prüfung von Feldeigenschaften .....	95
2.3.9	Verfahren für die Evaluation von Regeln .....	98
2.4	Relationstabellen ICD- und OPS-Listen .....	99
2.5	Exportfeldbeschreibung .....	100
2.5.1	Exportmodule .....	100
2.5.2	Modulauslöser, Dokumentations- und Exportmodule .....	101
2.5.3	Über das Datenfeld „Status des Leistungserbringers“ im Modul SEPE gibt der Dokumentierende an, ob sich der Datensatz auf den vertragsärztlichen Sektor oder auf den Krankenhaussektor bezieht. Exportdatensatz.....	102
2.5.4	Umgang mit Umlaut-Domains und E-Mail-Adressen .....	102
2.6	Versionierung.....	102
2.6.1	Grundlegende Definitionen .....	102
2.6.2	Delta-Informationen zur vorhergehenden Version.....	104
2.7	Administrative Objekte .....	104
2.7.1	XML-Mapping in der Spezifikationsdatenbank (QS DOK) .....	104
2.7.2	Datenservices .....	107
2.7.3	Prüfschritte .....	108
3	XML-Schema .....	110
3.1	KompositionsmodeLL.....	110
3.2	Darstellung der XML-Struktur.....	111
3.3	Aufbau der XML-Exportdatei.....	112
3.3.1	Namensräume .....	112

3.3.2	Wurzelement <root> .....	113
3.3.3	Header-Bereich .....	114
3.3.4	Body-Bereich.....	126
4	Tools .....	135
4.1	Datenprüfprogramm .....	135
4.1.1	Umfang der Prüfungen .....	135
4.1.2	Ausgangskontrolle vor Versand .....	135
4.1.3	Programmaufruf .....	136
4.1.4	Verzeichnisstruktur .....	140
4.1.5	Ausgabe.....	140
4.1.6	Grafische Oberfläche .....	141
4.1.7	Programmierschnittstelle – API.....	141
4.2	Verschlüsselungspaket.....	142
4.3	LE-Pseudonymisierungsprogramm .....	142
C	Anhang .....	143
	Glossar .....	143
	Impressum.....	148

# Tabellenverzeichnis

Tabelle 1: Informationen aus der Datenbank, welche im GUI verwendet werden .....	20
Tabelle 2: XML-Schemata zur Prüfung vor der Verschlüsselung.....	29
Tabelle 3: Asymmetrische Verschlüsselung der XML-Elemente .....	30
Tabelle 4: XML-Schemata zur Prüfung nach der Verschlüsselung .....	30
Tabelle 5: Benennungselemente der Exportdateien.....	32
Tabelle 6: Pseudonymisierung der LE-Daten .....	35
Tabelle 7: Struktur der Tabelle Modul .....	48
Tabelle 8: Struktur der Tabelle Ds .....	49
Tabelle 9: Struktur der Tabelle Tds .....	49
Tabelle 10: Struktur der Tabelle BasisTyp .....	51
Tabelle 11: Struktur der Tabelle Schluessel .....	51
Tabelle 12: Struktur der Tabelle SchluesselWert .....	53
Tabelle 13: Struktur der Tabelle Version .....	57
Tabelle 14: Ausschnitt der Tabelle Ds .....	58
Tabelle 15: Felder des QS-Filter-Eingangsdatensatzes nach § 301 .....	59
Tabelle 16: Struktur der Tabelle ModulAusloeser .....	62
Tabelle 17: Struktur der Tabelle AdminKriterium .....	63
Tabelle 18: Basistypen der Variablen .....	64
Tabelle 19: Präzedenz und Assoziativität der Operatoren .....	66
Tabelle 20: Struktur der Tabelle Modul .....	70
Tabelle 21: Inhalte der Tabelle BogenTyp .....	72
Tabelle 22: Struktur der Tabelle BogenFeld .....	73
Tabelle 23: Struktur der Tabelle Feld .....	75
Tabelle 24: Struktur der Tabelle BasisTyp .....	76
Tabelle 25: Struktur der Tabelle Schluessel .....	77
Tabelle 26: Struktur der Tabelle SchluesselWert .....	79
Tabelle 27: Struktur der Tabelle Abschnitt .....	80
Tabelle 28: Arten von Hinweistypen .....	80
Tabelle 29: Tabelle RegelTyp .....	81
Tabelle 30: Struktur der Tabelle Regeln .....	82
Tabelle 31: Struktur der Tabelle RegelFelder .....	82
Tabelle 32: Basistypen der Datenfelder in den Plausibilitätsregeln .....	83
Tabelle 33: Präzedenz und Assoziativität der Operatoren .....	85
Tabelle 34: Typen von Feldgruppen .....	89
Tabelle 35: Struktur der Tabelle FeldGruppe .....	90
Tabelle 36: Struktur der Tabelle FeldgruppeFelder .....	91
Tabelle 37: Formale Definition einer Feldgruppe .....	92

Tabelle 38: Struktur der Tabelle ExportModul.....	100
Tabelle 39: Struktur der Tabelle ExportFormatExportModul .....	105
Tabelle 40: Felder der Abfrage vPruefung .....	108
Tabelle 41: Root-Element – Attribute.....	114
Tabelle 42: Kind-Elemente des Elements document.....	115
Tabelle 43: Kind-Elemente des Elements software.....	117
Tabelle 44: Kind-Elemente des Elements information_system.....	118
Tabelle 45: Angabe des betreffenden Datenflusses .....	118
Tabelle 46: Angabe des betreffenden Datenfluss-Ziels .....	118
Tabelle 47: Attribute des Elements header/provider .....	119
Tabelle 48: Attribute des Elements header/protocol .....	120
Tabelle 49: Attribute des Elements validation_provider .....	121
Tabelle 50: Attribute des Elements validation_item .....	122
Tabelle 51: Attribute des Elements status_document.....	123
Tabelle 52: Attribut des Elements status .....	124
Tabelle 53: Attribut des Elements error .....	124
Tabelle 54: Kind-Elemente des Elements error .....	124
Tabelle 55: Leistungserbringeridentifizierende Daten im vertragsärztlichen und stationären Bereich .....	127
Tabelle 56: Element care_provider in den DAS-Schemata interface_LQS.xsd und interface_LQS_psn.xsd.....	127
Tabelle 57: Attribute des Elements cases .....	128
Tabelle 58: Kind-Elemente des Elements case_admin.....	129
Tabelle 59: Kind-Element des Elements statistic.....	131
Tabelle 60: Attribut des Elements sent.....	132
Tabelle 61: Kind-Elemente des Elements statistic/sent.....	132
Tabelle 62: Attribute des Elements statistic .....	133
Tabelle 63: Kind-Elemente des Elements statistic/processed .....	133

# Abbildungsverzeichnis

Abbildung 1: Serielles Datenflussmodell für die sektorenübergreifende Qualitätssicherung .....	25
Abbildung 2: Beispiel einer Registrierungsnummer .....	31
Abbildung 3: Beispiel einer Empfangsbestätigung von einer DAS an einen LE.....	37
Abbildung 4: Tabellen und Relationen der Datenfeldbeschreibung .....	69
Abbildung 5: Algorithmus zur Evaluation von Plausibilitätsregeln .....	99
Abbildung 6: Tabelle Datenservice.....	107
Abbildung 7: Dateiordner der Schnittstellen-Schemata .....	110
Abbildung 8: Root-Element und Kind-Elemente header und body .....	114
Abbildung 9: Aufbau des Elements header .....	115
Abbildung 10: Aufbau des Elements document.....	115
Abbildung 11: Aufbau des Elements software.....	117
Abbildung 12: Aufbau des Elements information_system .....	118
Abbildung 13: Aufbau des Elements provider.....	119
Abbildung 14: Aufbau des Elements header/protocol .....	120
Abbildung 15: Aufbau und Kind-Elemente des Elements validation_provider .....	121
Abbildung 16: Aufbau und Kind-Elemente des Elements validation_item .....	122
Abbildung 17: Aufbau des Elements status_document .....	123
Abbildung 18: Aufbau und Kind-Elemente des Elements status .....	123
Abbildung 19: Aufbau des Elements error .....	124
Abbildung 20: Aufbau und Attribute des Elements encryption(Krankenhaus) .....	125
Abbildung 21: Aufbau des Elements body .....	126
Abbildung 22: Aufbau des Elements body/data_container .....	126
Abbildung 23: Aufbau des Elements care_provider – Krankenhaus .....	126
Abbildung 24: Aufbau des Elements cases .....	128
Abbildung 25: Aufbau des Elements case.....	128
Abbildung 26: Aufbau des Elements case_admin .....	129
Abbildung 27: Aufbau des Elements case_admin/protocol .....	130
Abbildung 28: Aufbau des Elements statistic .....	131
Abbildung 29: Aufbau des Elements sent .....	132
Abbildung 30: Aufbau des Elements processed.....	133
Abbildung 31: Weiche Schemavarianten für das DPP .....	139
Abbildung 32: Verzeichnisstruktur eines DPP-Ordners .....	140
Abbildung 33: Beispiel für eine Index.html Datei im Ordner <output_path>/html .....	141
Abbildung 34: Grafische Oberfläche des Datenprüfprogramms .....	141

## Abkürzungsverzeichnis

Abkürzung	Bedeutung
AG	Arbeitsgruppe
BAS	Bundesauswertungsstelle
BE	Bundesebene
BfArM	Bundesinstitut für Arzneimittel und Medizinprodukte
DAS	Datenannahmestelle
DeQS-RL	Richtlinie zur datengestützten einrichtungsübergreifenden Qualitätssicherung
DRG	Diagnosis Related Groups (diagnosebezogene Fallgruppen)
EBM	Einheitlicher Bewertungsmaßstab
FES	fortgeschrittene elektronische Signatur
FU	Follow-up
G-BA	Gemeinsamer Bundesausschuss
GKV	Gesetzliche Krankenversicherung
GOP	Gebührenordnungsposition
GUID	Globally Unique Identifier
ICD	International Statistical Classification of Diseases and Related Health Problems (Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme)
ICD-10-GM	Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme - 10. Revision - German Modification
ID	Identifikationsnummer
IK	Institutionskennzeichen
IKNR	Institutionskennzeichennummer
iv	Indirekte Verfahren
KH	Krankenhaus, Krankenhäuser
KHEntgG	Gesetz über die Entgelte für voll- und teilstationäre Krankenhausleistungen (Krankenhausentgeltgesetz)
KIS	Krankenhausinformationssystem
KVDT	Kassenärztliche Vereinigung-Datentransfer (Datenformat)
LE	Leistungserbringer
LKG	Landeskrankenhausgesellschaft(en)

LQS	Landesgeschäftsstelle(n) für Qualitätssicherung
MVZ	Medizinisches Versorgungszentrum
OPS	Operationen- und Prozedurenschlüssel
PB	Probefließ
QS	Qualitätssicherung
RL	Richtlinie
RSA	Verfahren zur Datenverschlüsselung, entwickelt von R. Rivest, A. Shamir und L. Adleman
SGB	Sozialgesetzbuch
SGB V	Sozialgesetzbuch Fünftes Buch
SQG	Sektorenübergreifende Qualität im Gesundheitswesen
SWA	Softwareanbieter
TB	Testbetrieb
TDS	Teildatensatz
VST	Vertrauensstelle
XML	Extensible Markup Language (maschinenlesbare Auszeichnungssprache)
XSLT	Extensible Stylesheet Language Transformation (Programmiersprache zur Transformation von XML-Dokumenten)

# Änderungsindex

Änderungen der Datenbanken im Vergleich zur Vorversion lassen sich anhand der Delta-Tabellen nachvollziehen.

## Kapitelübergreifende Änderungen

- Redaktionelle Änderungen
- Konkretisierungen und Optimierung von Formulierungen
- Anpassung und Ergänzung von Jahreszahlen, Beispielen und Abbildungen
- Sofern möglich, Löschen von Inhalten mit dem Bezug zum Verfahren QS WI (einschl. Beschreibungen, die den ambulanten und/oder vertragsärztlichen Bereich betreffen)

Konkrete Informationen zu den inhaltlichen Änderungen sind der Spezifikationskomponente Uebersicht\_Aenderungen zu entnehmen.

Die spezifischen Änderungen der vorliegenden Technischen Dokumentation werden im Folgenden mit Bezug zur jeweiligen Version dargestellt.

Änderung	Kapitel/Abschnitt	Version
Entfernen von einem Listenpunkt, wonach Felder mit Vorjahreswerten vorbelegt werden dürfen.	1.2.1	2026 V01
Entfernen von Umsetzungsanforderungen an die Sollstatistik - <u>Übermittlung erfolgt im Zuge der fallbezogenen Dokumentation!</u>	2	2026 V01
Entfernen von Umsetzungsanforderungen an die Konformitätserklärung - <u>Übermittlung erfolgt im Zuge der fallbezogenen Dokumentation!</u>	2	2026 V01
Entfernen des Kapitels zur Fehlerprüfung	1.4	2026 V01

# Leseanleitung

Diese Technische Dokumentation orientiert sich in ihrem Aufbau an den Abläufen der Erfassung und Übermittlung der Qualitätssicherungsdaten. Ziel dieser Struktur ist es, eine nachvollziehbare und logische Sicht auf die Umsetzung und Durchführung der beschriebenen Schritte zu gewährleisten. Die Prozesse und Unterprozesse werden im Abschnitt A Prozesse beschrieben und spiegeln die reale, chronologische Abfolge wider. Jede Prozessbeschreibung berücksichtigt zudem die unterschiedlichen Komponenten, die für die Umsetzung benötigt und in Abschnitt B Komponenten detailliert beschrieben werden. Im Abschnitt C Anhang wird ein Glossar mit den wichtigen Begriffen zum Themenbereich der Spezifikation zur Verfügung gestellt.

Für eine korrekte Umsetzung der Spezifikation ist es notwendig, die Dokumentation entsprechend ihrer Anordnung von Prozessen zu Komponenten zu befolgen. Einige Bereiche, die sich ausschließlich an bestimmte Zielgruppen richten, sind entsprechend gekennzeichnet.

## Legende

Die in dieser Dokumentation verwendeten Symbole heben bestimmte Aspekte bei der Umsetzung der Spezifikation hervor.



### Achtung

Beschreibt Ursache, Folge und Vermeidung einer besonderen Fehlanwendung, die zu Problemen bei der Implementierung oder Ähnlichem führen kann.



### Hinweis

Nützliche Informationen, Tipps oder Ratschläge zur Anwendung. Keine wesentlichen oder für das korrekte Funktionieren erforderlichen Informationen.

## Beispiel:

Beispiele sind ein Hilfsmittel, um zuvor vermittelte Informationen oder konkrete Abschnitte der Anwendung zu verdeutlichen.

LE

### Zielgruppe LE

Markiert nur für folgende Zielgruppe relevanten Bereich:

Leistungserbringer (LE) im Allgemeinen

**KH**

**Zielgruppe KH**

Markiert nur für folgende Zielgruppe relevanten Bereich:  
Krankenhaus (KH)

**DAS**

**Zielgruppe DAS**

Markiert nur für folgende Zielgruppe relevanten Bereich:  
Datenannahmestelle (DAS) im Allgemeinen

**LKG**

**Zielgruppe LKG**

Markiert nur für folgende Zielgruppe relevanten Bereich:  
Landesgeschäftsstelle für Qualitätssicherung (LQS)  
Landeskrankenhausgesellschaften (LKG)  
Landesarbeitsgemeinschaften (LAG)

# 1 Einleitung

Die vorliegende Technische Dokumentation beschreibt die Spezifikation zur einrichtungs- und sektorenübergreifenden Qualitätssicherung gemäß §§ 136 ff. SGB V.

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation umfasst alle Komponenten im Zusammenhang mit der Datenerfassung, d.h. von der Bestimmung einer Dokumentationspflicht bis hin zur Rückprotokollierung übermittelter Datensätze. Die vorliegende Technische Dokumentation beschreibt die hierbei relevanten Prozesse und Komponenten.

Regelungsbereich der Spezifikation der einrichtungsbezogenen QS-Dokumentation ist das QS-Verfahren *Diagnostik und Therapie der Sepsis (QS Sepsis)* gemäß der Richtlinie zur datengestützten einrichtungsübergreifenden Qualitätssicherung (DeQS-RL)<sup>1</sup>, die vom Gemeinsamen Bundesauschuss (G-BA) beschlossen wird. Im Spezifikationsjahr 2026 beinhaltet die Spezifikation der einrichtungsbezogenen QS-Dokumentation daher Module des Verfahrens *Diagnostik und Therapie der Sepsis (QS Sepsis)*.

Das QS-Verfahren wird über verschiedene Erhebungsinstrumente abgebildet:

- die fallbezogene Dokumentation beim Leistungserbringer
- die Erhebung von Sozialdaten bei den Krankenkassen
- die jährliche einrichtungsbezogene QS-Dokumentation

Die vorliegende Spezifikation bildet das Erhebungsinstrument der jährlichen einrichtungsbezogene QS-Dokumentation beim Leistungserbringer ab. Für die Nutzung von Sozialdaten bei den Krankenkassen gemäß § 299 SGB V sowie die fallbezogene QS-Dokumentation beim Leistungserbringer werden unabhängige Spezifikationen veröffentlicht. Für eine spezifikationskonforme Umsetzung von QS-Softwareprodukten sind alle Spezifikationskomponenten zu berücksichtigen.

## 1.1 Spezifikationsbegriff

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation ist die Gesamtheit aller Vorgaben, nach denen die Bestimmung der dokumentationspflichtigen Einrichtungen, die QS-Dokumentation selbst und die Übermittlung der Daten erfolgen sollen, bezogen auf ein Spezifikationsjahr. Die Dokumentationspflicht einer Einrichtung wird ausgelöst, wenn mindestens ein Fall den definierten Auslösekriterien entspricht. Die Zuordnung eines Falles zu einer Spezifikation richtet sich nach dem Kalenderjahr der Aufnahme in das Krankenhaus.

Um die komplexen Anforderungen an die sektorenübergreifende QS-Dokumentation sowie die zugehörigen Datenflüsse zu erfüllen, besteht die Spezifikation aus verschiedenen Komponenten, die je nach Anwender spezifisch zusammengestellt werden. Als Komponenten werden dabei Access-Datenbanken, Technische Dokumentationen, Ausfüllhinweise und anderes bezeichnet. Jeder Anwender bekommt damit das für ihn Relevante in einem eigenen Spezifikationspaket als

---

<sup>1</sup><https://www.g-ba.de/richtlinien/105/>

Download zur Verfügung gestellt. Jedes dieser Pakete kann auf diese Weise auch unabhängig von den anderen aktualisiert werden.

Damit gibt es ein Spezifikationspaket für

- den Regelbetrieb
- ggf. Sonderexporte
- ggf. Probebetriebe
- ggf. Testbetriebe

Sowohl die Spezifikationspakete als auch die einzelnen Komponenten werden nach einem einheitlichen Schema benannt, das bereits im Namen übersichtlich die relevanten Informationen wie Betriebsart, Exportformat und Versionierung enthält. Dieses Schema wird in der gesonderten Spezifikationskomponente „TechDok\_Benennungsschema“ vorgegeben. Durch die Versionierung sowohl auf der Ebene der Pakete als auch auf der Ebene der Komponenten ist gewährleistet, dass der aktuelle Stand leicht ersichtlich ist. Zudem wird die Kommunikation über die anzuwendenden Bestandteile der Spezifikation erleichtert.

Jedem Paket liegt eine Auflistung der einzelnen Komponenten und ggf. eine Übersicht über die Änderungen zur vorhergehenden Version bei.

## 1.2 Zielsetzung und Zielgruppe

Die Spezifikation der einrichtungsbezogenen QS-Dokumentation ist ein komplexes Regelwerk, das mithilfe verschiedener Komponenten verbindliche Grundlagen für alle Prozesse im Zusammenhang mit der Erfassung und Übermittlung von QS-Daten bei den unterschiedlichen Verfahrensteilnehmern (Leistungserbringer, Datenannahmestellen, Bundesauswertungsstelle) vorgibt und beschreibt. Dadurch werden die Bereitstellung valider und vergleichbarer Daten sowie ein unter datenschutzrechtlichen Gesichtspunkten sicherer Datenfluss gewährleistet.

## 1.3 Releaseplanung

Die Veröffentlichung der finalen Version der Spezifikation zur einrichtungsbezogenen QS-Dokumentation erfolgt nach Beschluss der Spezifikation sowie der entsprechenden Richtlinie durch das Plenum des G-BA. Die Spezifikation wird spätestens ein halbes Jahr vor Beginn des Dokumentationsstarts auf der IQTIG-Website (<http://www.iqtig.org>) zur Verfügung gestellt.

Da die Dokumentation der einrichtungsbezogenen QS-Dokumentation erst nach Ablauf des Spezifikationsjahres erfolgen kann, entsprechen die ICD-, OPS- und EBM-Kodes in der finalen Spezifikationsversion bereits der für dieses Spezifikationsjahr gültigen Version des BfArM. Ein separates Update (BfArM-Umstieg) entfällt daher. Es erfolgt eine regelmäßige Abstimmung mit Verfahrensteilnehmern (z. B. Softwareherstellern, Datenannahmestellen) in Form von Informati-onstreffen, Workshops und Kommunikationsplattform.

### Change- und Fehlermanagement

Das IQTIG empfiehlt für die Optimierung der Zusammenarbeit mit den beteiligten Stellen die folgenden Aktivitäten:

- Meldung von festgestellten Fehlern (z. B. Spezifikations- und Softwarefehlern)
- Verbreitung von Änderungsvorschlägen
- Abstimmung von Terminen und Umsetzungen im Rahmen der Releaseplanung
- Erfahrungsaustausch, um eine möglichst einheitliche Vorgehensweise zu ermöglichen
- Abstimmung der Spezifikationsänderungen

Vorschläge, Fehlermeldungen und Diskussionspunkte können per E-Mail an den Verfahrenssupport mitgeteilt werden. Sollten Sie keine Zugangsdaten zur Kommunikationsplattform haben, obwohl Sie eine beteiligte Institution (z. B. Softwareanbieter, Datenannahmestelle) sind, lassen Sie sich bitte bei uns registrieren.

**Ihr Ansprechpartner:**

Institut für Qualitätssicherung und Transparenz im Gesundheitswesen

Katharina-Heinroth-Ufer 1

10787 Berlin

Telefon: (+49) 30 58 58 26 340

Fax: (+49) 30 58 58 26 341

[verfahrenssupport@iqtig.org](mailto:verfahrenssupport@iqtig.org)

[www.iqtig.org](http://www.iqtig.org)

# A Prozesse

Im Folgenden werden die einzelnen Prozesse und Unterprozesse der Spezifikation beschrieben.

# 1 QS-Dokumentation

In diesem Kapitel werden die Prozessschritte sowie die in jedem Prozessschritt benötigten Werkzeuge der QS-Dokumentation in Bezug auf die Auslösung, Erfassung, Verarbeitung und Datenübermittlung der QS-Dokumentation beschrieben.

## 1.1 Auslösung

Für jeden Fall im Krankenhaus wird geprüft, welche Datensätze (Module) der externen Qualitätsicherung dokumentationspflichtig sind. Entspricht mindestens ein Fall (Tracer-Operation) den definierten Auslösekriterien, wird die einrichtungsbezogene QS-Dokumentation ausgelöst. Mit der vorliegenden Spezifikation können Systeme entwickelt werden, die eine zeit- und prozessnahe Auslösung von QS-Dokumentationen ermöglichen.

Der Algorithmus (Abschnitt B 1.4) trifft seine Entscheidung auf der Grundlage der medizinischen Routinedokumentationen bzw. Abrechnungsdaten (Kodierung der Diagnosen [Haupt- und Nebendiagnosen, Primär- und Sekundärdiagnosen], Prozeduren, Gebührenordnungspositionen gemäß EBM-Katalog).

KH

### Einrichtungen, die stationäre Fälle am Krankenhaus behandeln

Die medizinischen Routinedokumentationen sind in jedem Krankenhausinformationssystem (KIS) verfügbar, da sie auch für die Umsetzung der Datenübermittlungsvereinbarung gemäß § 301 Abs. 3 SGB V (kurz: DÜV-301) nötig sind. Die Informationen werden in einzelnen Fällen auch über spezialisierte Systeme, die auf die individuellen Anforderungen im jeweiligen Bereich zugeschnitten sind, zur Auslösung herangezogen.

Der QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V ist in Abschnitt B 1.3.1 beschrieben.

LE

## 1.2 Erfassung

Nach Auslösung eines Moduls durch den QS-Filter folgt der Prozess der Erfassung. Neben QS-Daten (Abschnitt 1.2.3) sind auch einrichtungsidentifizierende Daten wie das Institutionskennzeichen bei Krankenhäusern zu dokumentieren (Abschnitt 1.2.2). Die Erfassung ist abgeschlossen, wenn alle zu dokumentierenden Datenfelder unter Berücksichtigung von Abhängigkeiten und Plausibilitätsprüfungen vollständig erfasst wurden (Abschnitt 1.2.4). Die Regelungen zur Gestaltung von Eingabemasken sind Abschnitt 1.2.1 zu entnehmen.

### 1.2.1 Gestaltung von Eingabemasken

Die Benutzeroberfläche einer Erfassungssoftware (Graphical User Interface = GUI) oder einer webbasierten Lösung soll ergonomisch und anwenderfreundlich gestaltet sein. Gestaltung und Layout der Eingabemaske sind Aufgabe der Softwarehersteller bzw. der Webportal-Entwickler.

Neben Anforderungen der Kunden werden üblicherweise firmeninterne Standards bzw. Vorgaben des Betriebssystems (z. B. Windows) für das „look and feel“ berücksichtigt.

Diese Spezifikation definiert als Minimalstandard die für den Anwender sichtbaren Inhalte der Dokumentationsformulare. Als Referenz für die sichtbaren Inhalte dienen die Dokumentationsbögen, die als Bestandteil der Spezifikation durch das IQTIG veröffentlicht werden. Die Dokumentationsbögen werden als PDF-Dokumente bereitgestellt, die aus der Spezifikationsdatenbank automatisch generiert worden sind.



### **Achtung Verbindliche Grundlage zur Implementierung**

Bei den Dokumentationsbögen handelt es sich um Formulare zur Ansicht (Muster), die nicht zur Dokumentation zu verwenden sind. Die Papierform ist hier nur als eine Abbildung eines denkbaren Eingabeformulars oder einer denkbaren Eingabemaske zu verstehen.

Verbindlich sind nur die Inhalte der Datenbank zur QS-Dokumentation.

Tabelle 1 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 müssen.

*Tabelle 1: Informationen aus der Datenbank, welche im GUI verwendet werden*

<b>Tabelle</b>	<b>Attribut</b>	<b>Bemerkung</b>	<b>sichtbar für Anwender</b>
Modul	name	Kürzel des Datensatzes (z. B. 18 / 1), erscheint üblicherweise im Titel des Formulars	ja
Modul	bezeichnung	Bezeichnung des Datensatzes (z. B. Mammachirurgie), erscheint üblicherweise im Titel des Formulars	ja
Bogen	bezeichnung	Bezeichnung des Teildatensatzes	ja
BogenFeld	gliederungAufBogen	Nummer des Datenfeldes, dient bei umfangreicherer Bögen zur besseren Orientierung	ja
BogenFeld	bezeichnung	Bezeichnung des Datenfelds	ja
BogenFeld	ergaenzendeBezeichnung	Ergänzende Bezeichnung zum Datenfeld, kann z. B. durch Wahl der Schrift von der Bezeichnung abgesetzt werden	ja
Feld	laenge	Definiert die Länge des Eingabefelds. Für die Gestaltung des Eingabefelds sind weitere Informationen aus der Datenbank wichtig (z. B. Feld.nachKommaLaenge).	ja

Feld	einheit	Einheiten (wie z. B. ml) müssen angezeigt werden.	ja
BasisTyp	format	Formatanweisungen (wie z. B. TT.MM.JJJJ) sollen – sofern nicht durch übergeeignete Eingabefelder unterstützt – für den Anwender angezeigt werden.	ja
Schlues-selWert	code	Bei Schlüsselfeldern sollen die Kodes möglichst in Auswahllisten angezeigt werden. Bei einigen Realisierungsvarianten (z. B. Checkbox) kann auf die Anzeige der Kodes verzichtet werden.	(ja)
Schlues-selWert	bezeichnung	Bei Schlüsselfeldern müssen die Textdefinitionen der Kodes (z. B. in einer Auswahlliste) angezeigt werden. Für die Sortierung sind die Attribute sortierNrVewendet und zahl der Tabelle Schluessel relevant (Abschnitt B 2.2.3).	ja
Abschnitt	bezeichnung	Die Überschriften sind wichtig für die Strukturierung und das Verständnis des Datensatzes und müssen deshalb in der QS-Dokumentationssoftware angezeigt werden.	ja

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

### Allgemeine Grundsätze für die Gestaltung der Eingabemaske

Die Grundsätze für die Plausibilitätsprüfungen wirken sich insbesondere auf die Gestaltung der Benutzeroberflächen in Erfassungssoftware oder Weblösung aus. Durch die funktionale Gestaltung sollte 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>2</sup>

- Keine Suggestion von Feldinhalten durch Vorbelegung (Defaults): 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.

---

<sup>2</sup> Externe Systeme, die Daten an ein Erfassungsprogramm übergeben, sollten diese Grundsätze sinngemäß anwenden.



### Hinweis

Eine Vorbelegung mit Standardwerten ist nicht zulässig. Die Übernahme von im KIS vorhandenen Angaben in die QS-Dokumentation ist hingegen zulässig.  
Eine Ausnahme besteht für die in Tabelle Feld der QS-Dokumentationsdatenbank definierten Strukturparameter, die durch den Softwareanbieter im Bogen vorbelegt werden dürfen.  
Beispielsweise können die Angaben des Datenfeldes IKNRKH als Strukturparameter vorbelegt werden, wenn diese Angaben vorliegen.  
Die Frage, ob z. B. während des stationären Aufenthaltes Komplikationen aufgetreten sind, darf hingegen nicht mit einem Standardwert „nein“ vorbelegt werden, wenn hierzu keine Angaben im KIS enthalten sind.

- 
- 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 müssen sie wörtlich übernommen werden.
- 



### Achtung Verbindliche Grundlage zur Implementierung

Plausibilitätsregeln sind nicht in allen Fällen den Dokumentationsbögen zu entnehmen. Verbindlich sind nur die Inhalte der Datenbank zur QS-Dokumentation.

- 
- 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 entsprechenden 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 Erfassungsformular 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.
- 

### Beispiel<sup>3</sup>:

Wenn die Transfusion von Blut zunächst bejaht und das Feld „Fremdblut“ angekreuzt worden ist, soll das Entfernen des Kreuzes des übergeordneten Feldes „Bluttransfusion“ nicht automatisch zum Entfernen des

---

<sup>3</sup> Die im Beispiel genannten Felder sind NICHT Bestandteil der Spezifikationsdatenbank für die QS-Dokumentation. Sie sind frei erfunden und dienen lediglich der Erläuterung.

Kreuzes bei „Fremdblut“ führen. Vielmehr soll eine Fehlermeldung erfolgen und der Anwender gezwungen sein, zunächst das Kreuz bei „Fremdblut“ zu entfernen, bevor „Bluttransfusion“ verneint werden kann.

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

In Einzelfällen ist es aus Anwendersicht hilfreich, wenn die Eingabemaske die Eingabe von Messwerten in Einheiten ermöglicht, die von den spezifizierten Einheiten abweichen. Beispielsweise soll im Modul SEPE laut Spezifikation das Datenfeld ANZPFLEGEZVK in Vollkräften mit zwei Nachkommastellen dokumentiert werden. Die Funktionalität einer automatischen Umrechnung sollte möglichst in die Erfassungssoftware oder die Weblösung integriert werden, um den Dokumentationsaufwand zu verringern. Exportiert werden die Datenfelder in der gemäß Spezifikation gültigen Einheit.

### **Gestaltung von Eingabemasken mit Layout-Feldgruppen**

Sogenannte Layout-Feldgruppen sollen in der Erfassungsmaske separat kenntlich gemacht werden. Hierbei handelt es sich um Datenfelder, die zu einer logischen Gruppe zusammengefasst werden können. Feldgruppen können Filterfelder oder abhängige Felder beinhalten. Abhängige Felder von Layout-Feldgruppen werden auf den generierten Dokumentationsbögen eingegraut dargestellt. Hierbei handelt es sich um Feldgruppen, bei denen das Attribut grauWennNegativ in der Datenbanktabelle FieldGruppe auf TRUE gesetzt ist. Diese sollen für die Gestaltung von Eingabemasken verwendet werden. Die Eigenschaften von Layout-Feldgruppen sind in Abschnitt B 2.3.7 erläutert.

### **Empfehlung zur Umsetzung von Layout-Feldgruppen**

Ist das Attribut grauWennNegativ 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 ausgeblendet werden.

#### **1.2.2 Einrichtungsidentifizierende Daten**

In Hinblick auf eine einrichtungsbezogene Auswertung und Berichterstattung sind einrichtungsidentifizierende Daten in der QS-Dokumentation zu erfassen.

## Einrichtungsidentifizierende Daten des Krankenhauses

Wichtigstes Merkmal zur Identifikation der Einrichtung Krankenhaus ist das Institutionskennzeichen.

### 1.2.3 QS-Daten

Es sind alle erforderlichen Daten zu dokumentieren. Hierbei kann eine automatische Übertragung der QS-Daten aus dem KIS möglich sein.

### 1.2.4 Plausibilitätsprüfungen

Fehlende und widersprüchliche Angaben in den Datensätzen sollen durch umfangreiche Plausibilitätsprüfungen verhindert werden. In der QS-Dokumentation 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 aufwendiges Korrekturverfahren weitgehend entfällt.

Es gelten folgende Grundsätze für die Plausibilitätsprüfung:

- Alle Felder müssen ausgefüllt sein, wenn andere logische Sachverhalte dem nicht 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.

### Arten der Plausibilitätsprüfungen

Es wird zwischen folgenden Arten von Plausibilitätsprüfungen unterschieden:

- **harte Prüfungen:** Bei einer harten Regelverletzung ist ein Dokumentationsabschluss der QS-Dokumentation eines Vorgangs unzulässig. Die in der Technischen Dokumentation und der Datenbank definierten Plausibilitätsprüfungen sind hart, außer wenn sie explizit als weich gekennzeichnet sind.
- **weiche Prüfungen:** Die weichen Plausibilitätsprüfungen der Spezifikation sind bis spätestens zum Dokumentationsabschluss durchzuführen. Bei einer Regelverletzung erhält der Benutzer einen Warnhinweis, anhand dessen er entscheidet, ob eine Änderung von Feldinhalten notwendig ist.

### Einzelregeln

Einzelregeln sind in der Datenbank zur QS-Dokumentation in Regelsyntax in der Tabelle Regeln hinterlegt. Außerdem gibt es Regeln, die nur in Form von Feldeigenschaften – nicht aber in Regelsyntax – in der Datenbank hinterlegt sind. Die standardisierten Fehlertexte dieser Prüfungen sind Abschnitt B 2.3.8 zu entnehmen.

Es müssen sowohl die harten als auch die weichen feldbezogenen Prüfungen ausgeführt werden. Die Evaluation soll direkt bei der 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 Schlüsselkodes.

### Feldgruppenregeln

Datenfelder (Bogenfelder) eines Moduls können zu einer Feldgruppe zusammengefasst werden, um logische Abhängigkeiten von Bogenfeldern abzubilden. Das bedeutet in der Praxis, dass der Anwender daran gehindert wird, Felder mit Werten auszufüllen, die der Logik der Feldgruppe widersprechen.

Die explizite Definition von Feldgruppen strukturiert sowohl die Bogenfelder als auch die Plausibilitätsregeln, indem diese die Bogenfelder eines Moduls zu einer logisch zusammenhängenden Gruppe von Feldern zusammenfassen. Die Feldgruppen ergeben sich dabei indirekt aus der Definition von Plausibilitätsregeln.

Die Abbildung von Feldgruppenregeln in der Datenbank für QS-Dokumentation ist in Abschnitt B 2.3.7 erläutert.

## 1.3 Export der Daten

Die dokumentierten QS-Daten müssen bei sektorenübergreifenden Verfahren gem. DeQS-RL in Abhängigkeit vom Abrechnungskontext der Leistungserbringung verschiedenen Datenflüssen zugeordnet werden.

Abbildung 1 stellt das serielle Datenflussmodell gem. DeQS-RL dar. Hierbei sind je nach Leistungserbringer bzw. Abrechnungskontext unterschiedliche Datenannahmestellen vorgesehen:

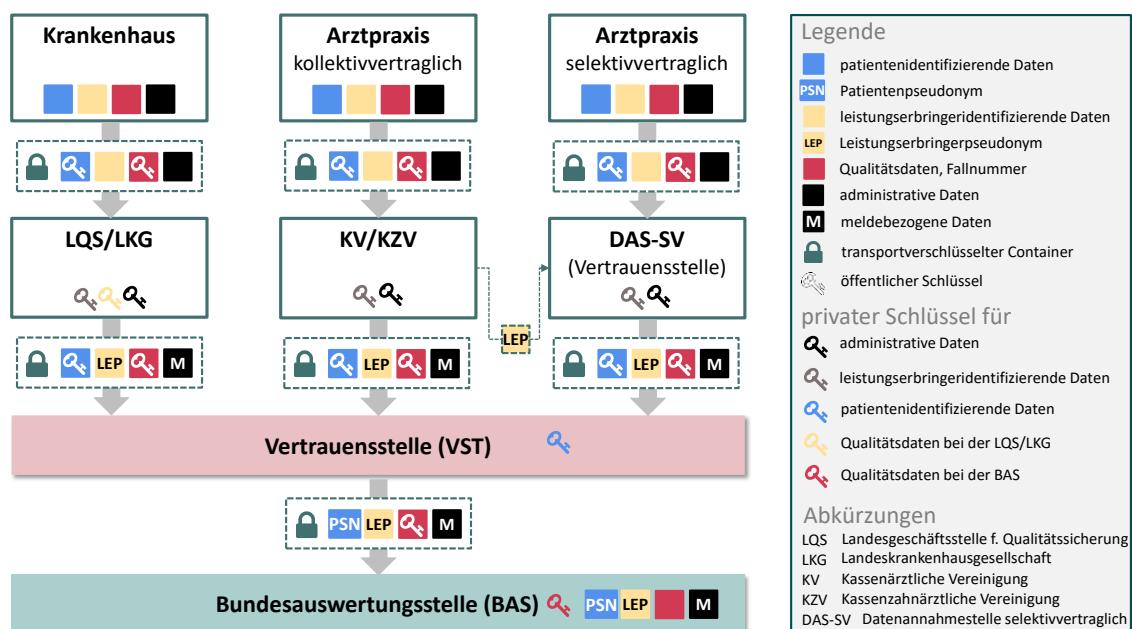


Abbildung 1: Serielles Datenflussmodell für die sektorenübergreifende Qualitätssicherung

Die Zuordnung der einzelnen Fälle zu den Leistungserbringern gemäß Datenfluss nach DeQS-RL richtet sich primär nach dem Abrechnungskontext und nicht nach dem Ort der Leistungserbringung.



### Hinweis zum Datenfluss

Die einrichtungsbezogenen QS-Daten nach § 3 Absatz 2 der DeQS-RL werden direkt von der Datenannahmestelle an die Bundesauswertungsstelle weitergeleitet, da keine Pseudonymisierung durch die Vertrauensstelle erfolgt.

Die Exportdateien werden beim Leistungserbringer erstellt und dann an die zuständige Datenannahmestelle (DAS) versendet. Datenannahmestellen nehmen die Datenlieferungen im Rahmen der DeQS-RL entgegen, führen die Leistungserbringerpseudonymisierung durch und leiten anschließend die Daten weiter an die Bundesauswertungsstelle. Der Export und die Übermittlung sämtlicher Daten finden im XML-Format statt.

#### 1.3.1 Erzeugen der Exportdatei

Die zu exportierenden Daten der einrichtungsbezogenen QS-Dokumentation werden vom Dokumentationssystem in Exportdateien geschrieben und die entsprechenden Vorgänge (identifiziert durch Vorgangsnummern, vgl. „Identifizierung von Datensätzen“ im Abschnitt 1.4.2) im absenden Dokumentationssystem als „exportiert“ markiert.

#### Inhalte der Exportdatei

Aufgrund der einheitlichen Leistungserbringerpseudonymisierung dürfen nur Datensätze der Module gemeinsam exportiert werden, die einem gemeinsamen Pseudonymisierungsverfahren unterliegen.



### Hinweis zum Datenfluss

Die Datensätze der fall- und einrichtungsbezogenen QS-Dokumentation können aufgrund der unterschiedlichen Datenflüsse nicht gemeinsam exportiert werden, obwohl sie demselben Pseudonymisierungsverfahren unterliegen.

Die einrichtungsbezogenen QS-Daten werden direkt von der Datenannahmestelle an die Bundesauswertungsstelle weitergeleitet, da keine Pseudonymisierung durch die Vertrauensstelle erfolgt. Der Datenfluss für die fallbezogenen QS-Daten (SEP) erfolgt hingegen unter Einbeziehung einer Datenannahmestelle und einer Vertrauensstelle an die Bundesauswertungsstelle.

Dateien, in denen diese Trennung nicht eingehalten wird, werden von der Datenannahmestelle zurückgewiesen.

#### Anonymisierung

Die Vorschriften zur Anonymisierung von Bogenfeldern sind in den Tabellen ErsatzFeld und ErsatzFuerFeld in der Spezifikationsdatenbank für QS-Dokumentation zu finden (Abschnitt

B 2.5.3). Da im Rahmen der einrichtungsbezogenen QS-Dokumentation keine personenbezogenen Daten erhoben werden, ist eine Anonymisierung durch die QS-Dokumentationssoftware derzeit nicht notwendig.

### Aufbau der Exportdatei

Die innere Struktur der Exportdatei ergibt sich unter anderem aus der Datenfeldbeschreibung der einzelnen Module und den oben beschriebenen Anonymisierungsmaßnahmen zur Erfüllung datenschutzrechtlicher Vorgaben. Aufbauend auf dieser Beschreibung wird ein XML-Schema abgeleitet. Die Struktur der Exportdatei wird durch entsprechende XML-Schemata festgelegt (Kapitel B 3).

Das XML-Schema beschreibt und definiert die Struktur des XML-Dokuments (Exportdatei) sowie den Inhaltstyp (Datentypen der einzelnen Bögen und Felder).

Die Exportdateien sind wie folgt aufgebaut:

- Header-Bereich enthält Metadaten (administrative Daten)
- Body-Bereich enthält die einrichtungsidentifizierenden (1.2.2) und die QS-Daten (1.2.3)



#### Hinweis

Es ist zu beachten, dass die Zeichenkodierung deklariert wird und diese der UTF-8-Kodierung entsprechen muss.

### Sonderzeichen in XML

Folgende Zeichen, die bei der XML-Syntax besondere Bedeutung haben, müssen geschützt werden, falls Sie als Inhalte zu übermittelnder Informationen (Attributwerte) vorkommen sollen:

- & &amp;
- < &lt;
- > &gt;

Folgende Zeichen (Apostroph bzw. Anführungszeichen) müssen nur dann entsprechend geschützt werden, wenn sie selbst zur Übermittlung von Attributwerten verwendet werden.

- ' &apos; (muss nur dann geschützt werden, wenn das Apostroph selbst zur Übermittlung von Attributwerten verwendet wird.)
- " &quot; (muss nur dann geschützt werden, wenn das Anführungszeichen selbst zur Übermittlung von Attributwerten verwendet wird.)

### Felder der Exportdatei

Einen Überblick über die zu exportierenden Felder eines Moduls liefert die Abfrage ExportFelderFürEinModul.<sup>4</sup>

<sup>4</sup> Metadaten des Datencontainers sind nicht Teil der Spezifikationsdatenbank und werden daher nicht durch die Abfrage ausgegeben.

## Export von Listenfeldern

Alle Elemente von Listenfeldern werden exportiert, ohne die Nummer des Listenfelds (im Namen des Exportfelds) an den Namen des Listenfelds anzuhängen (zu Listenfeldern siehe Abschnitt B 2.2.3).



### Hinweis

In der vorliegenden Spezifikation zur einrichtungsbezogenen QS-Dokumentation sind derzeit keine Listenfelder vorgesehen.

## Zusatzfelder des Datenexports

Zusatzfelder und administrative Felder im Header, die nicht in der Datenfeldbeschreibung (Tabelle BogenFeld) eines Moduls enthalten sind, sind von der QS-Dokumentationssoftware zu füllen.<sup>5</sup>

Die Zusatzfelder sind in der Tabelle ZusatzFeld definiert. 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.

Bei der Organisation im XML-Dokument ist weiter die Abfrage vExportZieleXml aus den administrativen Objekten zu berücksichtigen (Abschnitt B 2.7). Die Informationen in dieser Abfrage schließen aus, dass bestimmte Felder dem Element <qs\_data> zugeordnet werden, indem anhand eines XPath-Ausdrucks (xmlXPath) auf die richtige Stelle im XML-Dokument verwiesen wird.

## Organisation der Exportfelder im XML-Dokument

Die Exportfelder sind abhängig von der Modul- und Teildatensatzzugehörigkeit des Datensatzes im Element <qs\_data>, einem Unterelement des Elements <case> unterzubringen (Element qs\_data, Kapitel B 3).

## Stornierung

Um den Datensatz zu stornieren, muss <case>/<case\_admin><action> auf „delete“ gesetzt werden. Die Datenannahmestelle wird dadurch veranlasst, den betreffenden Datensatz einschließlich aller Vorversionen und Teildatensätze als „storniert“ zu kennzeichnen. Der Stornovorgang wird in der Datenbestätigung protokolliert.

Der zu stornierende Datensatz muss ebenfalls eine hochgezählte/fortgeschriebene Versionsnummer enthalten, um die Stornierung unabhängig von der Reihenfolge der Verarbeitung von Datensätzen sicherzustellen. Ein Storno mit einer bereits verwendeten Versionsnummer wird zurückgewiesen (Bestätigungsstatus ERROR, Fehlerart DOPPELT). Ein Stornoversuch eines noch nicht übermittelten Datensatzes wird ebenfalls zurückgewiesen.

<sup>5</sup> Hier gilt demnach nicht der Grundsatz, dass Felder nicht vorbelegt sein dürfen.

Zur Stornierung eines Datensatzes (Vorgang) genügt der Export der entsprechenden administrativen Daten <case>/<case\_admin>. Die QS-Daten (<qs\_data>) des zu stornierenden Datensatzes sind nicht erneut zu übermitteln.

### 1.3.2 Datenprüfung

Zusätzlich zu bereits im Rahmen der Erfassung durchzuführenden feldbezogenen und -übergreifenden Plausibilitätsprüfungen kann nach dem Datenexport die gesamte Struktur der XML-Datei durch aus der Spezifikationsdatenbank abgeleitete Schemata geprüft werden (Kapitel B 3). Welches Schema für einen Leistungserbringer vorgesehen ist, zeigt Tabelle 2

Tabelle 2: XML-Schemata zur Prüfung vor der Verschlüsselung

Richtlinie	Bereich	Schnittstelle	Schema
DeQS	Krankenhaus	interface_LE	interface_LE_lqs.xsd

Die einfachste Lösung für die Prüfung der Datenstruktur und der Plausibilität liegt in der Nutzung eines Datenprüfprogramms, das auf der Basis von XSLT die aus der Spezifikationsdatenbank QSDOK ausgeleiteten Plausibilitätsregeln in dem XML-Dokument prüft und Verstöße entsprechend im XML-Code dokumentiert. Ein solches Datenprüfprogramm setzt einen Standard für die Güte der Daten, der unbedingt einzuhalten ist. Die Datenprüfung muss an der Exportdatei vor der nachfolgend beschriebenen XML-Verschlüsselung vorgenommen werden.

Das Datenprüfprogramm ersetzt jedoch nicht die Verpflichtung der Softwareanbieter, schon bei der Eingabe der Daten eines Datensatzes, d.h. dokumentationsbegleitend, die Einhaltung der Plausibilitätsregeln sicherzustellen.

Details zur Verwendung eines Datenprüfprogramms können der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer bzw. Abschnitt B 4.1 entnommen werden.

### 1.3.3 XML-Verschlüsselung

Nach Fertigstellung des Dokuments und seiner Prüfung mit dem Datenprüfprogramm müssen einzelne Datenbereiche verschlüsselt werden. Die Verschlüsselung erfolgt asymmetrisch mit einem öffentlichen Schlüssel. Die öffentlichen Schlüssel der Datenannahmestellen (LKG) und der Bundesauswertungsstelle stehen den Anwendern auf der IQTIG-Website<sup>6</sup> zur Verfügung.

Diese Verschlüsselung baut auf dem W3C-XML-Encryption-Standard auf. Das IQTIG stellt ein Verschlüsselungsprogramm bereit, mit dem eine verfahrenskonforme XML-Verschlüsselung nach diesem Standard durchgeführt werden kann. Das Einbinden der Funktionen des Verschlüsselungsprogramms erfolgt u.a. über einen Befehlszeilenauftruf mit Parametern.

---

<sup>6</sup> <https://iqtig.org/datenerfassung/datenerfassung-und-uebermittlung/servicedateien/>

Tabelle 3 zeigt, in Abhängigkeit (erste Spalte) von der Stelle im Datenfluss (LE, DAS) und dem Weg des Datenflusses (LKG), welche Datenbereiche (Kopfzeile) mit welchem öffentlichen Schlüssel verschlüsselt werden müssen.

Tabelle 3: Asymmetrische Verschlüsselung der XML-Elemente

	<b>/qs_data</b>	<b>/care_provider</b>
Verschlüsselung beim LE für Versand an LKG	mit öffentlichem Schlüssel der LKG	keine Verschlüsselung
Verschlüsselung durch LKG nach LE-Pseudonymisierung vor Weiterleitung	mit öffentlichem Schlüssel der BAS	mit öffentlichem Schlüssel der BAS

#### 1.3.4 Ausgangsvalidierung

Als letzte Maßnahme vor der Weiterleitung muss das Dokument gegen das Übertragungsschema auf Gültigkeit geprüft werden. Durch diese Prüfung wird sichergestellt, dass die richtigen Bereiche des XML-Codes verschlüsselt sind, und sie schließt aus, dass kritische Daten versehentlich unverschlüsselt den Leistungserbringer verlassen.

Das für einen bestimmten Leistungserbringer geltende Schema kann der gesonderten Beschreibung der Schema-Familie entnommen werden.

Tabelle 4: XML-Schemata zur Prüfung nach der Verschlüsselung

<b>Richtlinie</b>	<b>Bereich (Versand von)</b>	<b>Schnittstelle</b>	<b>Schema</b>
DeQS	Krankenhaus (LE)	interface_LE_DAS	interface_LE_IQS
	LKG (DAS)	interface_DAS_BAS	interface_DAS_BAS

LE



#### Hinweis

Die für die Ausgangsvalidierung zu verwendenden Schemata sind dieselben, wie die bei der Eingangsvalidierung der DAS verwendeten Schemata. Bei nicht bestandener Schemaprüfung wird die Annahme des Dokuments verweigert.

#### 1.3.5 Beispiele für Exportdateien

Beispieldateien unterschiedlicher Schnittstellen können der Spezifikationskomponente „XML\_Schema“ entnommen werden. Im Ordner XML-Beispiele sind für alle Schnittstellen entsprechende Ordner hinterlegt, deren Beispieldateien jeweils im Unterordner „export“ abgelegt sind. Die Benennung der Beispieldateien ist an die Benennung der zugehörigen Schemata angelehnt.

## 1.4 Datenübermittlung

Im folgenden Abschnitt werden die abschließend zu unternehmenden Arbeitsschritte Dateibennung und Datenversand beschrieben.

Vor der Datenübermittlung muss die XML-Verschlüsselung erfolgreich durchgeführt worden sein. Ob sie erfolgreich durchgeführt wurde, kann mit den entsprechenden Schemata für die Ausgangskontrolle geprüft werden.

Für die sektorenübergreifenden Verfahren werden die Datenlieferfristen in den Themenspezifischen Bestimmungen der **DeQS-RL** festgelegt.

Die Übermittlung von Daten muss grundsätzlich gesichert sein, unabhängig von der XML-Verschlüsselung gezielter, einzelner Inhalte. Diese gesicherte Übermittlung kann einerseits über gesicherte Übertragungswege (z. B. durch eine SSL-Verschlüsselung einer Webschnittstelle oder Übertragung mittels SFTP) gewährleistet werden sowie alternativ über eine zusätzliche Sicherung der zu übertragenden Informationen (Verschlüsselung der Pakete).

### 1.4.1 Registrierung

LE →

#### Registrierung eines Dokumentationssystems (LKG)

Die Registrierung des Dokumentationssystems eines Leistungserbringers bei der DAS ist Voraussetzung für die Datenübermittlung. Die Registrierung unterstützt die sichere Übertragung von Daten per E-Mail.

#### Registrierung bei einer Datenannahmestelle

Im stationären Bereich bekommen registrierte Dokumentationssysteme von der Datenannahmestelle eine Registrierungsnummer zugewiesen, die folgendermaßen aufgebaut ist:

<Registrierungsnummer> = <Ländercode><Registriercode>

Jedes Dokumentationssystem des stationären Leistungserbringers ist auf diese Weise bundesweit identifizierbar. Beim Wechsel des Dokumentationssystems eines Leistungserbringers ist zu beachten, dass eine neue Registrierungsnummer zu beantragen ist.



Abbildung 2: Beispiel einer Registrierungsnummer

Die Adressen der einzelnen DAS werden auf der IQTIG-Homepage aufgeführt.

## Registrierung von Datenannahmestellen bei der Bundesauswertungsstelle

DAS →

Datenannahmestellen müssen bei der Bundesauswertungsstelle registriert sein. Sofern dies bereits für Datenübertragungen fallbezogener QS-Daten unter Anwendung der Basisspezifikation erfolgt ist, können die bereits vorhandenen Registriernummern zur Anwendung kommen.

## Registrierung von Softwareanbietern beim IQTIG für Testzwecke

Softwareanbieter, die an Testbetrieben teilnehmen wollen, müssen sich bei den Stellen registrieren, die den Test-Datenservice bereitstellen.

### 1.4.2 Dateibenennung

Die Exportdatei muss, analog zur Basisspezifikation nach einem bestimmten Schema benannt werden.

Syntax: <GUID>\_<Inhaltskennung><Protokolltyp>\_<Rolle Absender>.xml

Tabelle 5: Benennungselemente der Exportdateien

Element	Bedeutung
GUID	Die verwendete GUID ist die im Dokument verwendete ID des Datenpakets (Abschnitt A „Eindeutige Kennzeichnung der XML-Exportdateien“ auf S. 32).
Inhaltsken-nung	Q → QS-Daten
Protokoll-typ	T → Transaktionsprotokoll bzw. Empfangsbestätigung D → Datenflussprotokoll
Rolle Ab-sender	LE → Leistungserbringer DAS → Datenannahmestelle BAS → Bundesauswertungsstelle

#### Beispiele:

QS-Daten eines LE

47d16341-9e27-4e75-a27e-b791fbbd2dc8\_0.LE.xml

QS-Daten nach Verarbeitung durch eine DAS

47d16341-9e27-4e75-a27e-b791fbbd2dc8\_0.DAS.xml

#### Eindeutige Kennzeichnung der XML-Exportdateien

Jede Exportdatei wird durch eine universell eindeutige ID (GUID) von der QS-Software gekennzeichnet.

Ein Globally Unique Identifier (GUID) ist eine global eindeutige Zahl mit 128 Bit, die eine Implementation des Universally Unique Identifier Standards (UUID) darstellt.

GUIDs haben das Format XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX, wobei jedes X für ein Zeichen aus dem Hexadezimalsystem steht und damit eine Ziffer 0–9 oder ein Buchstabe A–F sein kann.

Erläuterung zur GUID:

- Die GUID wird im Exportprozess von der QS-Software einer bestimmten Exportdatei zugewiesen.
- Das registrierte Dokumentationssystem ordnet jeder an eine Datenannahmestelle übermittelten Exportdatei eine eindeutige GUID zu.
- Diese GUID wird im XML-Code des Dokuments als ID gesetzt. Sie muss bei dateibasierten Übertragungsverfahren in der Dateibenennung verwendet werden.
- Eine GUID wird durch eine gelungene Transaktion zwischen den jeweils beteiligten Übertragungspartnern verbraucht. Jede Datenannahmestelle muss dafür sorgen, dass ein eindeutig über die GUID identifizierbares Dokument nur einmal denselben Verarbeitungsschritt durchlaufen kann. Andernfalls ist die Verarbeitung mit einem entsprechenden Fehlerprotokoll abzulehnen.



### Achtung

Aufgrund von fehlerhaften, automatisierten Verarbeitungen in der DAS oder BAS kann der Ausnahmestatus entstehen, dass ein bereits verarbeitetes Dokument nebst Rückprotokollierung bis hin zum Leistungserbringer erneut verarbeitet werden muss.

Für den Fall, dass ein erneuter Datenfluss von QS-Daten ab der zuständigen DAS erfolgt, muss auch die Dokumentationssoftware beim Leistungserbringer über die Möglichkeit verfügen, mehr als ein Datenflussprotokoll zu einer GUID aufnehmen können.

## Identifizierung von Datensätzen

Die Vorgangsnummer (auch Datensatz-ID oder ID genannt) kennzeichnet in eindeutiger Weise jeden dokumentierten Vorgang eines registrierten Dokumentationssystems. Die Vorgangsnummer ist daher allein ein Merkmal des QS-Dokumentationssystems, um einen Datensatz innerhalb des registrierten Dokumentationssystems eindeutig identifizieren zu können. Eine Vorgangsnummer darf keine Rückschlüsse auf Personen oder Einrichtungen ermöglichen. Für die einrichtungsbezogene QS-Dokumentation wird empfohlen, die Modulkennung („SEPE“) als Vorgangsnummer zu verwenden. Die Modulkennung ist in Kombination mit der LE-Kennung sowie dem Spezifikationsjahr (Jahresangabe der Spezifikationsversion) eindeutig.

Mit der Spezifikation 2020 wird zusätzlich ein 36-stelliger pseudozufälliger Globally Unique Identifier (GUID) eingeführt, der die Datensätze bundesweit eindeutig identifiziert ohne Rückschlüsse auf den Leistungserbringer oder Patienten zuzulassen. Dadurch ist es nicht mehr nötig, dass die Datenannahmestellen aus den Vorgangsnummern bundesweit eindeutige Vorgangsnummern machen.

## Je Leistungserbringer und Modul ein Datensatz

Im Rahmen der einrichtungsbezogenen QS-Dokumentation darf je Leistungserbringer (IKNR) jährlich nur ein Datensatz je (Export-)Modul übermittelt werden. Ein übermittelter Datensatz darf

innerhalb der gültigen Lieferfrist beliebig oft aktualisiert werden (Update). Ebenso ist ein Storno möglich (Delete). Die Zuordnung von Update bzw. Delete erfolgt hierbei über die bei allen Datenübertragungen einheitliche Vorgangsnummer und kann nur in Verbindung mit einer jeweils höheren Versionsnummer akzeptiert werden, vgl. Erläuterungen zu dem XML-Element case\_admin (Tabelle 58).

### 1.4.3 Datenversand via E-Mail

Für die Übermittlung der Daten über nicht gesicherte Übertragungswege (E-Mail) muss die Exportdatei in eine Transaktionsdatei (verpackte und symmetrisch verschlüsselte AES-Datei) umgewandelt werden.

Die Erstellung der Transaktionsdatei kann mit dem vom IQTIG bereitgestellten Verschlüsselungsprogramm (TPacker) durchgeführt werden. Das Einbinden der Funktionen des Verschlüsselungsprogramms erfolgt über einen Befehlszeilenaufruf mit Parametern. Das Verschlüsselungsprogramm übernimmt auch die Dateibenennung der Transaktionsdatei mithilfe von übergebenen Parametern

**KH**

Die Voraussetzung für die Transportverschlüsselung zwischen LE und DAS ist die Registrierung der Dokumentationssoftware bei der zuständigen DAS. Bei der Registrierung erhält der Leistungserbringer einen Verschlüsselungskode, der bei der Erzeugung der Archivdatei verwendet werden muss.

**DAS**

Für die Übermittlung von DAS zu BAS ist ausschließlich das E-Mail-Verfahren vorgesehen. Die Übermittlung im Produktivbetrieb erfolgt an die E-Mail-Adresse [eb-daten@iqtig.org](mailto:eb-daten@iqtig.org). Eine Übermittlung im Testbetrieb wird an die E-Mail-Adresse [eb-daten-test@iqtig.org](mailto:eb-daten-test@iqtig.org) adressiert.

Die Transaktionsdatei wird nach einem vordefinierten Schema benannt (Abschnitt A 1.4 „Übermittlung der Daten im Datenfluss“):

---

**Beispiel:**

Transaktionsdatei

T-BU1234a-2016\_01\_19\_160945.zip.aes

T-BU1234a-2016\_01\_23\_114113\_045.zip.aes (millisekundengenau)

---

## 1.5 Pseudonymisierung der LE-Daten

Zu den Aufgaben der DAS gehört die Pseudonymisierung der leistungserbringeridentifizierenden Daten (LID) vor der Weiterleitung an die BAS. Hier ist die eindeutige Kennzeichnung der Leistungserbringer (Institutionskennzeichen bei Krankenhäusern [IKNRKH]) durch ein Pseudonym zuersetzen.

Das Pseudonym des Leistungserbringers (LE-Pseudonym) wird unter Entfernung der IKNRKH des Leistungserbringers in den XML-Code an vorgegebener Stelle eingetragen. DAS müssen in der Lage sein, den Leistungserbringer zu depseudonymisieren. Zudem muss das Pseudonymisierungsverfahren konstant sein, um Datenzusammenführungen mit Vorjahren zu erlauben.

Bei der systemweit einheitlichen Leistungserbringerpseudonymisierung ist zu beachten, dass gemäß DeQS-RL diese verfahrensspezifisch zu erfolgen hat. Für die systemweit einheitliche Leistungserbringerpseudonymisierung gemäß DeQS-RL ist das Pseudonymisierungsprogramm (PSP) zu verwenden (Abschnitt B Kapitel 4.3). Welche Elemente in XML durch welche DAS zu pseudonymisieren sind, wird in der folgenden Tabelle aufgeführt.

Tabelle 6: Pseudonymisierung der LE-Daten

Datenannahmestelle	Beschreibung
LKG	<care_provider>/<IKNR> ist durch <pseudonym> zu ersetzen. Für die Pseudonymisierung wird nur die IKNR verwendet.
	DeQS-RL Die Pseudonymisierung ist verfahrensspezifisch. Pseudonymisierung mittels des systemweit einheitlichen Verfahrens (PSP).



### Hinweis

Ob die Pseudonymisierung erfolgreich durchgeführt wurde, kann auch mit den entsprechenden Schemata unter /interface\_DAS/ interface\_<DAS>\_psn.xsd geprüft werden.

Nach der Verarbeitung von XML-Dateien durch die DAS sind noch folgende Änderungen vorzunehmen:

### Aktualisierung der Absenderinformationen

Die Absenderinformationen (/header/provider) müssen durch eigene Informationen (z. B. Funktion, Registriernummer) ersetzt werden.

### Eindeutigkeit der Vorgangsnummern

Die Leistungserbringer erzeugen eine in ihrer QS-Software eindeutige Vorgangsnummer, die ihnen die Reidentifikation der Fälle ermöglicht. 2020 wird zusätzlich ein 36-stelliger pseudozufälliger Globally Unique Identifier (GUID) eingeführt, der die Datensätze bundesweit eindeutig identifiziert ohne Rückschlüsse auf den Leistungserbringer oder Patienten zuzulassen.

Für den stationären Bereich gewährleisteten bisher Datenannahmestellen die Übersetzung der Leistungserbringer-Vorgangsnummern in bundesweit eindeutige Vorgangsnummern. Infolge der Einführung der GUID ist dies ab 2020 nicht mehr erforderlich und die Datenannahmestellen sollen die Vorgangsnummern nicht mehr ändern oder ergänzen.

Bei der Durchführung von Stellungnahmeverfahren, bei denen die BAS mit dem Leistungserbringer in Kontakt tritt, werden die originären Vorgangsnummern der Leistungserbringer zur Identifizierung der Vorgänge verwendet. Zusätzlich nennt die BAS jeweils die GUID, so dass die Leistungserbringer auch diese oder Teile davon zur Identifizierung von Fällen verwenden können.

Dies könnte nötig werden, falls bei einem Leistungserbringer, die eigenen Vorgangsnummern nicht eindeutig sind, weil für ein QS-Verfahren mehrere Software-Instanzen verwendet werden.



## Hinweise

Die Datenannahmestellen sollen ab 2020 die von den Leistungserbringern erstellten Vorgangsnummern nicht mehr ergänzen oder ändern. Wenn ein Leistungserbringer Vorgänge zu dem gleichen Modul mithilfe von unterschiedlichen QS-Systemen erfasst, kann dies mit einem erhöhten Suchaufwand bei der Fallidentifizierung, z. B. Fällen aus Listen mit auffälligen Vorgängen, einhergehen.

# 1.6 Rückprotokollierung

In diesem Kapitel wird die Rückprotokollierung in Bezug auf die Funktion, den Aufbau und die Erstellung beschrieben.

## 1.6.1 Funktion von Empfangsbestätigung und Datenflussprotokoll im Datenfluss

### Empfangsbestätigung

Die Empfangsbestätigung wird nach Erhalt und abschließender erfolgreicher Eingangsverarbeitung und Weiterleitung eines Dokuments über den Eingangskanal an den Absender übermittelt. Sie bestätigt dem Absender den Übergang der Verantwortung für das Dokument an den Aussteller.

Eine Empfangsbestätigung ist nur für den Absender bestimmt und wird nicht weitergeleitet. Bei Ausbleiben ist von einer fehlgeschlagenen Übermittlung auszugehen. Es gibt zurzeit keine verbindliche Vereinbarung, in welchem zeitlichen Rahmen eine Empfangsbestätigung erwartet werden kann. Angestrebt werden soll allerdings eine Echtzeitverarbeitung, sodass allein die Verarbeitungsdauer eines Dokuments die Verzögerung einer Empfangsbestätigung bedingt. Empfangsbestätigungen werden im Rahmen der einrichtungsbezogenen QS-Dokumentation lediglich von den DAS an die LE übermittelt. Empfangsbestätigungen von der BAS an die DAS sind aufgrund der Übermittlung eines Datenflussprotokolls (bzw. Miniprotokolls) nicht erforderlich.

### Beispiel:

Wie die folgende Abbildung zeigt, enthält die Empfangsbestätigung lediglich eine Benachrichtigung, dass die vom Leistungserbringer übermittelte XML-Datei verarbeitbar war und an die nachfolgende Stelle weitergeleitet wurde:

```

<?xml version="1.0" encoding="utf-8"?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" container_version="2.0" content_version="1.0" xsi:noNamespaceSchemaLocation="../../../../../../interface_LE_DAS/response_receipt.xsd">
  <header>
    <document>
      <id V="{f83f11a8-7d45-4f81-961c-dbef7e14fale}" />
      </document>
      <provider address="Virchowstraße 1, 10123 Musterstadt" email="email@domain.tld" fax="0123456789" function="Datenannahmestelle" name="Datenannahmestelle Beispieldname" phone="0123456789" registration="bu365478" />
    </header>
    <body>
      <nachricht>Ihre Dateneinsendung konnte erfolgreich eingelesen und an die Bundesauswertungsstelle weitergeleitet werden. Ein umfassendes Datenflussprotokoll über die Prüfergebnisse der Datenannahmestelle und der Bundesauswertungsstelle wird folgen.</nachricht>
    </body>
  </root>

```

Abbildung 3: Beispiel einer Empfangsbestätigung von einer DAS an einen LE

Die Empfangsbestätigung soll ohne Verschlüsselung an den Sender geschickt werden. Für eine weitere Vereinfachung der Benachrichtigung können die Datenannahmestellen den Inhalt der spezifizierten Benachrichtigung z. B. in die E-Mail übernehmen (betreff/body). Die vollständige Benachrichtigung lautet:

*Ihre Dateneinsendung konnte erfolgreich eingelesen und an die Bundesauswertungsstelle weitergeleitet werden.*

*Ein umfassendes Datenflussprotokoll über die Prüfergebnisse der Datenannahmestelle und der Bundesauswertungsstelle wird folgen.*

Die Benennung der unverschlüsselten Datei leitet sich aus dem Benennungsschema ab (Abschnitt A 1.4.2).

### Datenflussprotokoll

Ein Datenflussprotokoll (DPP) wird erstellt, wenn das Dokument keine weitere Verarbeitung mehr erlaubt. Das ist dann der Fall, wenn das Dokument durch einen der vorgesehenen Prüfungs- und Verarbeitungsschritte den Status `ERROR` erhält oder wenn das Dokument in der Bundesauswertungsstelle vollständig und erfolgreich verarbeitet wurde und den Status `WARNING` oder `OK` trägt.

Das Datenflussprotokoll dokumentiert alle an dem Dokument durchgeföhrten Prüfungen und deren Ergebnisse.

Ein Datenflussprotokoll wird in der Regel bis zum Leistungserbringer zurückübermittelt. Aus diesem Grund muss die Datenannahmestelle (DAS) nach der Prüfung des erhaltenen Datenflussprotokolls den Leistungserbringer depseudonymisieren.

Die BAS liefert ihre Datenflussprotolle an die DAS. Im Unterschied zur QS-Basispezifikation übermittelt die BAS die Datenflussprotolle nicht an die DAS-E-Mail-Adresse aus der Registrierung, sondern an die Absender-E-Mail-Adresse der Datenlieferung. Ist der Absender `a@das.de`, die in der BAS registrierte E-Mail-Adresse `b@das.de` und `/root/header/provider/@email` entspricht `c@das.de`, dann liefert die BAS das DPP an `a@das.de`.

Die im Protokoll im Attribut „`originator`“ ggf. enthaltenen Hinweise auf den Verursacher eines Fehlers müssen von der Datenannahmestelle so ausgewertet werden, dass Verarbeitungsfehler,

die weder auf Fehler des Leistungserbringers noch auf die Software/Weblösung des Leistungserbringers zurückzuführen sind, nicht an den Leistungserbringer weitergeleitet werden. Dieses ist daran erkennbar, dass im Attribut „originator“ ein von Leistungserbringer oder Softwareanbieter<sup>7</sup> abweichender Wert steht. In einem solchen Fall soll verhindert werden, dass das Datenflussprotokoll an den Leistungserbringer verschickt wird.

Statt der Weiterleitung sollte die fehlerhafte Verarbeitung des auslösenden Beteiligten (Datenannahmestelle, Bundesauswertungsstelle) korrigiert werden und ein korrekt verarbeitetes Dokument, dessen Stand in Abstimmung mit der Bundesauswertungsstelle über die ID zurückgesetzt wurde, erneut in den Datenfluss eingebracht werden. Für das Zurücksetzen eines Dokuments steht kein automatisierter Prozess zur Verfügung. Dieses kann über die E-Mail-Adresse [verfahrenssupport@iqtig.org](mailto:verfahrenssupport@iqtig.org) vereinbart werden.

Die Datenflussprotokolle werden durch eine Reduktion der erhaltenen QS-Export-Daten erstellt. Dabei wird das Element <qs\_data> aus der Datei entfernt. Es verbleiben Header, Protokoll und Admin-Daten in der Datei. Das übriggebliebene XML enthält innerhalb der ursprünglichen Struktur des Dokuments die bis dahin protokollierten Prüfungen und die sich daraus ergebenden Statusmeldungen der Datensätze und des Dokuments. Der Aufbau des Protokolls ist in der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer detailliert beschrieben.

Für das Datenflussprotokoll wird das Attribut `feedback_range` des Elements <protocol> auf `dataflow` gesetzt.

Für diesen Verarbeitungsschritt kann das Datenprüfprogramm eingesetzt werden.

Das XML-Protokoll kann von der datenentgegennehmenden Stelle oder von der QS-Software mithilfe einer Template-Definition z. B. nach HTML transformiert werden. Den zuvor beschriebenen Plausibilitätsprüfungen schließen sich auf Bundesebene weitergehende Prüfungen an. Diese führen zu einem weiteren Anwachsen des Prüfprotokolls und der Datenqualität.

Bezogen auf einen bestimmten Datensatz ist es erst nach der letzten abgeschlossenen Prüfung auf Bundesebene möglich, eine Aussage darüber zu treffen, ob sich dieser Datensatz für die Aufnahme in den Bundesdatenpool eignet oder nicht.

Um dem Leistungserbringer das konkrete Ergebnis seiner Datenlieferung in Bezug auf den Bundesdatenpool mitteilen zu können, wird auch das bis zum Schluss weitergeföhrte Dokument von QS-Daten befreit und als Datenflussprotokoll an die DAS versandt, die dann die Aufgabe hat, dieses an die – nur am Pseudonym erkennbaren – Leistungserbringer zu übermitteln.

Die Datenflussprotokolle werden jedoch auch von der Datenannahmestelle für den Fall erstellt, dass eine Weiterleitung der QS-Exporte nicht möglich war. In diesem Fall wird keine Eingangsbestätigung an den Absender verschickt. Um das Dokument als Datenflussprotokoll kenntlich zu machen, muss das Attribut `root/header/protocol/@feedback_range` auf den Wert `dataflow` gesetzt werden.

---

<sup>7</sup> meint auch Anbieter einer Webportal-Software

## Miniprotokoll

Die Erstellung eines vollständigen Datenflussprotokolls durch die DAS ist nur möglich, wenn die ursprüngliche XML-Datei lesbar ist und nach Entfernung der QS-Daten schemakonform bleibt. Andernfalls ist durch die annehmende DAS an dieser Stelle ein reduziertes Datenflussprotokoll („Miniprotokoll“) zu erstellen, das die ID (GUID) des Dokuments (sofern diese zur Verfügung stand und lesbar war, sonst wird diese nicht angegeben) und die konkrete Fehlermeldung enthält. Ein Miniprotokoll ist z. B. zu erstellen, wenn die Exportdatei nicht wohlgeformt ist.



### Hinweis

Regelungen zu Fehlermeldungen, dem Prüfungsprozess und der Ergebnisprotokollierung, Prüfungsergebnissen, Beziehungen der Vorgangsebene/Dokumentenebene und Rückprotokollen (Bereitstellung eines XSLT für die Transformation) sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

## Regelungen für ein Vorgehen bei Verarbeitungsabbrüchen im besonderen Fehlerfall

Diese Regelungen dienen dazu, ein abgestimmtes Vorgehen durch alle am Datenfluss beteiligten Stellen zu etablieren, um die Datenbestände in allen Instanzen zu konsolidieren, wenn es zu unerwarteten Störungen in der Verarbeitung von Datenlieferungen in der DAS oder BAS kommt. Solche Störungen sind selten und stellen einen Ausnahmefall dar.

Solche Störungen können in äußerst seltenen Fällen aufgrund einer fehlerhaften automatisierten Verarbeitung von Datenlieferungen in der DAS oder BAS dazu führen, dass fehlerhafte Datenflussprotokolle versandt werden. Dadurch können beim Leistungserbringer unter Umständen mehrere Datenflussprotokolle zu einer GUID eintreffen, die widersprüchliche Einstufungen der Fälle enthalten. Um diesen Umstand an allen am Datenfluss beteiligten Stellen aufzulösen, müssen die involvierten Vorgänge erneut verarbeitet werden.

Das präferierte Vorgehen in dieser Situation ist ein erneuter Versand aller betroffenen Datensätze, die unter der betroffenen GUID gesandt wurden, durch den Leistungserbringer in einer höheren Version. Dieser Versand muss nach Rücksprache mit der Stelle, bei der die fehlerhafte Verarbeitung aufgetreten ist, und unter Einbindung der anderen Stationen im Datenfluss geschehen und wird deshalb telefonisch von der zuständigen DAS beim betroffenen Leistungserbringer initiiert. Für den erneuten Versand ist beim Leistungserbringer ein Export aller betroffenen Vorgangsnummern einschließlich der Erhöhung der zugehörigen Versionsnummer unter einer neuen GUID durchzuführen. Das im Zuge des Exports erstellte XML-Dokument enthält alle Fälle als Update in der jeweils höchsten Versionsnummer, die das Softwaresystem beim Leistungserbringer erstellt, und erhält eine neue und damit unverbrauchte GUID. Somit können alle Datensätze im Datenfluss über alle folgenden Instanzen (DAS, BAS) einschließlich der zugehörigen Rückprotokollierung mittels des Datenflussprotokolls bis hin zum Leistungserbringer regulär verarbeitet werden.

Sollte der Leistungserbringer nicht in der Lage sein, die Daten erneut zu exportieren, so ist der alternative Weg zu wählen, bei dem ein durch alle Instanzen bereits verarbeitetes Dokument (einschließlich Rückprotokollierung mittels Datenflussprotokoll bis hin zum Leistungserbringer) – und

somit eine bereits verarbeitete GUID – erneut verarbeitet wird. Es bedarf ebenfalls einer telefonischen Abstimmung aller Stationen im Datenfluss. Auf Leistungserbringerseite muss die Software in der Lage sein, zu einer GUID mehr als ein Datenflussprotokoll aufzunehmen, um die Einstufung der Daten gemäß dem zweiten Datenflussprotokoll analog zu den anderen Verarbeitungsstellen anzupassen.

## 2 Sollstatistik

Mit der Sollstatistik soll nach Abschluss eines Spezifikationsjahres unabhängig von den tatsächlich erfolgten Dokumentationen ermittelt werden, ob ein Leistungserbringer nach den definierten Bedingungen dokumentationspflichtig war. Sie bildet damit eine wichtige Grundlage zur Bestimmung der Vollständigkeit und Vollzähligkeit. Die rechtliche Grundlage zur Ermittlung der Sollstatistik wird durch § 15 der Richtlinie zur datengestützten einrichtungsübergreifenden Qualitätssicherung (DeQS-RL) definiert.



### Achtung

Für die einrichtungsbezogene Dokumentationen beim Verfahren QS-Sepsis gilt folgende Vorgabe:

Eine Aufstellung (Soll) nach Teil 1 § 15 der DeQS-RL einschließlich der Konformitätserklärung nach Teil 1 § 15 DeQS-RL ist für die einrichtungsbezogene Dokumentation nicht erforderlich; die Dokumentationspflicht für die einrichtungsbezogenen Daten ergibt sich aus der Aufstellung (Soll) der fallbezogenen Dokumentation.

## B Komponenten

In diesem Kapitel werden die einzelnen Komponenten der Spezifikation beschrieben. Ein Spezifikationspaket bildet die Gesamtheit seiner Spezifikationskomponenten ab. In einem Spezifikationspaket können beispielsweise folgende Komponenten enthalten sein:

- **TechDok** – bezeichnet alle Technischen Dokumentationen; diese geben detaillierte Erläuterungen zur Funktionsweise und Verwendung der einzelnen Komponenten. Da es verschiedene spezifisch verfasste TechDoks gibt, wird eine entsprechende Kennzeichnung gleich im Kürzel vermerkt.
  - **TechDok\_EDOK** – Technische Dokumentation für die einrichtungsbezogene QS-Dokumentation
  - **TechDok\_Benennungsschema** – Technische Dokumentation für das Benennungsschema
- **QSDOK** – bezeichnet die Access-Datenbank, in der die QS-Dokumentation spezifiziert wird.
- **QSF** – bezeichnet die Access-Datenbank, in der der QS-Filter spezifiziert wird.
- **Schema** – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Sie enthält einzelne XML-Schemata, die festlegen, in welcher Struktur XML-Daten an Schnittstellen im Datenfluss vorliegen müssen.
- **Precheck** – auf der Komponentenebene ist dies eine ZIP-Datei, die Schemata für alle administrativen Daten enthält, die ab dem Export dieser Spezifikation gültig sind. Diese Schemata sind ausschließlich für die Verarbeitung der Datenservices der DAS und BAS relevant.
- **Ausfuehlhinweise** – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Sie enthält einzelne HTML-Dateien für jedes Modul, die mit den Kürzeln der einzelnen Module benannt sind.
- **Anwenderinformationen** – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Die ZIP-Datei enthält einzelne HTML-Dateien für jedes im QS-Filter definiertes Modul, die mit den Kürzeln der einzelnen Modulauslöser benannt sind.
- **Dokuboege**n – auf der Komponentenebene ist dies eine ZIP-Datei, die die Versionierung und vollständige Bezeichnung enthält. Sie beinhaltet die Dokumentationsbögen als einzelne PDF-Dateien für jedes Modul, die mit den Kürzeln der einzelnen Module benannt sind. Die Dokumentationsbögen bilden einige wichtige Datenbankinhalte<sup>8</sup> ab.
- **UebersichtAenderungen** – ist eine PDF-Datei mit der Übersicht über die Änderungen in der Spezifikation zu allen Vorversionen des Spezifikationsjahres und zur letzten Version des Vorjahres. Hier werden die Änderungen übersichtlich zusammengefasst und erläutert.
- **Protocol** – bezeichnet das Stylesheet zur Darstellung des Rückprotokolls (XSLT).
- **Komponentenübersicht** – bezeichnet die Auflistung aller im Spezifikationspaket enthaltenen Komponenten mit Bezug zu Versionsnummer und Veröffentlichungsdatum (CSV).

Hilfsprogramme werden ggf. ebenfalls als Komponenten in ein Spezifikationspaket aufgenommen. Hierbei kann es sich beispielsweise um **Verschlüsselungsprogramme** oder ein

---

<sup>8</sup> Die Papierform ist hier nur als Abbildung des Eingabeformulars oder der Eingabemaske zu verstehen. Verbindlich sind daher nur die Inhalte der Datenbank zur QS-Dokumentation.

**Datenprüfung** handeln. Die Angaben der Betriebsart und des Exportformats erfolgen nur dann, wenn sich die Komponenten durch diese Merkmale unterscheiden.

- **Datenprüfung** – ist auf Komponentenebene eine ZIP-Datei, welche die Bestandteile des Datenprüfprogramms enthält.

**V<Versionsnummer>**: Die Versionierung erfolgt in ganzen Zahlen, die zweistellig angegeben sind (unter 10 mit einer vorstehenden 0, z. B. V01).

Weitere mögliche Spezifikationskomponenten sind der Technischen Dokumentation zur Basispezifikation für Leistungserbringer zu entnehmen.

### **Spezifikationsdatenbanken**

Als Access-Datenbanken zur Verfügung gestellte Spezifikationskomponenten dienen der (automatisierten) Erstellung von z. B. Software für QS-Filter, QS-Dokumentation. Folgende Spezifikationskomponenten werden als Access-Datenbanken (MS Access 2007-2013) zur Verfügung gestellt:

#### **Datenbank zur QS-Dokumentation (QSDOK)**

Die Datenbank zur QS-Dokumentation dient der Spezifikation von Datenerhebung und -erfassung unter Berücksichtigung von Plausibilitätsprüfungen und zu exportierenden Datenfeldern. Einige wichtige Datenbankinhalte werden über die Dokumentationsbögen (Dokuboege) abgebildet.

Folgende Spezifikationskomponente ergänzt die Datenbank um Hinweise und Informationen:

- Die Ausfüllhinweise (Ausfuellhinweise) dienen als Hilfestellung bei der Dokumentation durch den Anwender. Die Namen der HTML-Dateien für einzelne Datenfelder sind in der Datenbank (BogenFeld.ahinweis) hinterlegt.

Die Datenbank zur QS-Dokumentation ist in Kapitel B 2 erläutert.

#### **Datenbank zum QS-Filter (QSF)**

Die Datenbank zur QS-Dokumentation dient der Spezifikation zur Auslösung von Dokumentationsmodulen. Die Auslösebedingungen pro Modul werden als Übersicht in Form der Anwenderinformationen (Anwenderinformationen) zur Verfügung gestellt.

Die Datenbank zum QS-Filter ist in Kapitel B 1 erläutert.

#### **Datenbank zu Datenserviceinformationen**

Seit der Spezifikation 2019 werden relevante Datenserviceinformationen in einer separaten Datenbank gepflegt. Die Datenbank zu Datenserviceinformationen ist keine dem Spezifikationspaket zugehörige Komponente, verfügt jedoch über dieselbe Verbindlichkeit. Da sie eine eigenständige Spezifikationsdatenbank darstellt kann sie außerhalb des Releasezyklus angepasst werden.

## Verschlüsselungsprogramme

Seit der Spezifikation 2020 werden die Verschlüsselungsprogramme nicht mehr als Spezifikationskomponente veröffentlicht. Sie sind daher keine dem Spezifikationspaket zugehörige Komponente, verfügen jedoch über dieselbe Verbindlichkeit. Da sie ein eigenständiges Paket in Form einer ZIP-Datei darstellen, können sie außerhalb des Releasezyklus angepasst werden.

In der ZIP-Datei enthalten sind der X- und der TPacker sowie die zum Veröffentlichungszeitpunkt aktuellen öffentlichen Schlüssel der Datenservices im Datenfluss.

## Pseudonymisierungsprogramm

Ab der Spezifikation 2021 wird das Pseudonymisierungsprogramm nicht mehr als Spezifikationskomponente veröffentlicht. Sie ist daher keine dem Spezifikationspaket zugehörige Komponente, verfügt jedoch über dieselbe Verbindlichkeit. Sie kann außerhalb des Releasezyklus zur Verfügung gestellt werden.

- **PSP** – ist eine ZIP-Datei, die neben dem Pseudonymisierungsprogramm ein Informationsmerkblatt beinhaltet.

## Tabellenstruktur der Datenbanken

Die Tabellen und deren Spalten (Attribute) 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 Tabellenname 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.

BasisTyp (Tabelle)

idBasisTyp (Spalte)

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

id<TabellenName>

Der Ausdruck in spitzen 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>9</sup>, welches durch Setzen eines weiteren, eindeutigen Indexes (bestehend aus einem oder mehreren Attributen) definiert ist.

---

<sup>9</sup> Oder sie enthalten eine identifizierende Attributkombination, die einen eindeutigen Schlüssel definiert.

**Beispiele:**

Identifizierendes Attribut: Attribut `name` in Tabelle BasisTyp

Identifizierende Attributkombination: Attribute `code` und `fkSchluessel` in Tabelle SchluesselWert

Es gibt auch Tabellen, deren einziger eindeutiger Schlüssel der Primärschlüssel ist. Ein Beispiel ist die Tabelle MussKann mit dem Primärschlüssel `idMussKann` vom Typ KURZER TEXT (1) (entspricht VARCHAR (1)). Diese Tabellen sind als einfache „Nachschlagetabellen“ 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 Entitäten erstrecken (Inklusionsverknüpfungen, Joins).

Die Fremdschlüsselattribute (Namen beginnen mit `fk`) wurden als Datenbankattribute zum Nachschlagen eingerichtet. Zum Beispiel wird beim Fremdschlüsselattribut `fkModul` in der Tabelle Tds nicht mehr der Primärschlüssel des jeweiligen Moduls, sondern der Name des Moduls angezeigt.

**Hinweis**

Diese Änderung betrifft nur die Anzeige, nicht jedoch die Struktur der Datenbank.

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>10</sup>, jedoch einen eindeutigen Schlüssel, der über die Fremdschlüsselfelder definiert ist.

Folgende Attribute treten in vielen Tabellen auf:

- 1 `name` ist in der Regel als technischer Name zu verstehen. Zum Beispiel wird `Feld.name` als Variablenname in den Plausibilitätsregeln verwendet.
- 2 `bezeichnung` ist eine kurze Beschreibung. Zum Beispiel ist `TdsFeld.bezeichnung` der Text, welcher ein Feld auf einem Eingabeformular beschreibt.

<sup>10</sup> Hier: Primärschlüssel im Sinne der Access-Definition eines Primärschlüssels. Streng genommen wird über die beiden Fremdschlüssel ein neuer Primärschlüssel definiert.

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

# 1 QS-Filter

Der QS-Filter definiert, unter welchen Bedingungen ein Modul ausgelöst wird; in der Datenbank zum QS-Filter und den dazugehörigen Anwenderinformationen sind die Informationen hierzu hinterlegt. Die Auslösekriterien werden für jeden Fall geprüft. Sind die Kriterien für mindestens einen Fall erfüllt, gilt das jeweilige Modul der externen Qualitätssicherung (SEPE) für eine Einrichtung als dokumentationspflichtig.

## 1.1 Anmerkungen zur Struktur der Spezifikationsdatenbank für QS-Filter

Die QS-Filter-Spezifikation ist in einer relationalen Datenbank abgelegt. Zurzeit wird sie ausschließlich als Access-Datenbank (MS Access 2007-2013) zur Verfügung gestellt. Der Name der QS-Filter-Spezifikation richtet sich nach folgendem Schema:

<Spezifikationsjahr>\_DeQS\_QSF\_EDOK\_V<Versionsnummer>.mdb

<Versionsnummer> bezeichnet die 2-stellige Versionsnummer (z. B. 01).

---

### Beispiel:

Im Spezifikationsjahr 2020 ist die QS-Filter-Spezifikation 2020\_DeQS\_QSF\_EDOK\_V01.mdb<sup>11</sup> gültig. Die Kennung 2020 gilt für das Spezifikationsjahr 2020.

---

Weiterführende Erläuterungen zum Benennungsschema für Spezifikationskomponenten sind der Technischen Dokumentation für das Benennungsschema zu entnehmen.

Die Tabellenstruktur der Spezifikationsdatenbank wird in Abschnitt B Komponenten beschrieben.

## 1.2 Grundlegende Tabellen der Datenbank

Dieser Abschnitt beinhaltet die Darstellung der grundlegenden Tabellen der Spezifikationsdatenbank mit ihren zugehörigen Datensätzen ausgehend von den jeweiligen Modulen.

### 1.2.1 Module (Datensätze der QS-Dokumentation)

In der Tabelle Modul sind Referenzen auf die Module hinterlegt, deren Dokumentationspflicht durch den QS-Filter ausgelöst werden kann.

Hinter jedem Modul verbirgt sich ein Datensatz der korrespondierenden Spezifikation für die QS-Dokumentation. Diese Datensätze bilden in der Spezifikation für QS-Dokumentation den auszulösenden Dokumentationsbogen (z. B. der Software) ab.

---

<sup>11</sup> Die Versionsnummer der gültigen Spezifikation (z. B. V01, V02, usw.) ist dem zuletzt veröffentlichten Update zu entnehmen.

**Beispiele:**

Einrichtungen, die stationäre Fälle am Krankenhaus behandeln

- Der Modulauslöser SEPE löst den QS-Datensatz SEPE aus.
- 

Die Module der Spezifikation für QS-Dokumentationssoftware sind in der gleichnamigen Tabelle definiert. Eine Zuordnung ist über das Attribut Modul.name möglich. Die Datensätze der QS-Dokumentation werden in Abschnitt B 2.2.1 beschrieben.

Die Struktur der Tabelle Modul wird im Folgenden beschrieben.

Tabelle 7: Struktur der Tabelle Modul

Feldname	Datentyp	Bemerkung
idModul	INTEGER	Primärschlüssel
name	KURZER TEXT	Technischer Name des Moduls (Identifizierendes Attribut)
bezeichnung	KURZER TEXT	Beschreibender Text für den QS-Datensatz
fkSchluesselWert	INTEGER	Verweis auf denjenigen Kode des Schlüssels Modul (Tabelle SchluesselWert), welcher dem betreffenden Datensatz zugeordnet ist.
fkModulParent	INTEGER	Verweis auf das Eltern-Modul eines Zählleistungsbereichs  Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
zaehlLb	BOOLEAN	Wenn WAHR, existiert kein entsprechender QS-Datensatz, es handelt sich um einen „Zählleistungsbereich“ zur separaten Darstellung in der Sollstatistik.  Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
ausloeseModul	BOOLEAN	Wenn WAHR, existiert ein entsprechender QS-Datensatz, der ausgelöst werden kann. Dieses Modul kann gleich dem Exportmodul sein, z. B. SEPE.
exportModul	BOOLEAN	Wenn WAHR, handelt es sich um ein Exportmodul, z. B. SEPE.

### 1.2.2 Struktur der Datensatzdefinitionen

Die Module werden über den definierten Modulauslöser als dokumentationspflichtig erkannt, indem die hinterlegte Bedingung mit den im KIS gespeicherten Daten geprüft wird. Ist die Bedin-

gung erfüllt, wird das Modul ausgelöst. Da die in der Bedingung enthaltenen Felder im KIS vorliegen müssen, ist der Eingangsdatensatz gemäß § 301 SGB V in der Spezifikationsdatenbank hinterlegt. Die definierten Felder der Datensätze und der Filterbedingungen sind analog zur Spezifikation für die QS-Dokumentation aufgebaut.

In den nachfolgenden Abschnitten dieses Unterkapitels wird die grundlegende Struktur der Tabellen der Datensatzdefinitionen beschrieben und die Definitionen der Datenfelder erläutert.

## Datensätze

Jeder Datensatz besteht aus Teildatensätzen, welche ausgehend von einem Basisteildatensatz hierarchisch angeordnet sind. Folgende Datensätze sind definiert:

- QS-Filter-Datensätze

Für jeden dieser Datensätze wird in der Tabelle *Ds* ein Eintrag angelegt.

*Tabelle 8: Struktur der Tabelle Ds*

Feldname	Datentyp	Bemerkung
<i>idDs</i>	INTEGER	Primärschlüssel
<i>name</i>	KURZER TEXT	Technischer Name des Datensatzes (Identifizierendes Attribut)
<i>bezeichnung</i>	KURZER TEXT	Beschreibender Text

## Teildatensätze

Die Definition von Teildatensätzen befindet sich in der Tabelle *Tds* der Datenbank (Tabelle 9). Jeder Teildatensatz ist eindeutig durch seinen Namen (z. B. *FALL*, *PROZ* oder *DIAG*) charakterisiert.

*Tabelle 9: Struktur der Tabelle Tds*

Feldname	Datentyp	Bemerkung
<i>idTds</i>	INTEGER	Primärschlüssel
<i>name</i>	KURZER TEXT	Technischer Name des Teildatensatzes (Identifizierendes Attribut)
<i>bezeichnung</i>	KURZER TEXT	Beschreibender Text
<i>fkTds</i>	INTEGER	Optionaler Fremdschlüssel zu einem Mutterteildatensatz
<i>fkDs</i>	INTEGER	Bezug des Teildatensatzes zum übergeordneten Datensatz in der Tabelle <i>Ds</i> , z. B. QS-Filter-Datensatz oder QS-Filter-Datensatz für Sollstatistik

Feldname	Datentyp	Bemerkung
fkRelation-Typ	KURZER TEXT (1)	<p>Relationstyp, bezieht sich auf die Relation zum Mutterteildatensatz:</p> <ul style="list-style-type: none"> <li>* Eine beliebige Anzahl von Teildatensätzen darf angelegt werden!</li> <li>? Höchstens ein Teildatensatz darf angelegt werden!</li> <li>+ Mindestens ein Teildatensatz muss angelegt werden!</li> <li>! Genau ein Teildatensatz muss angelegt werden!</li> </ul>
fkEindeutigTdsFeld	INTEGER	Fremdschlüssel auf ein TdsFeld, das mehrfach vorhandene Teildatensätze eines Datensatzes identifiziert
sortierNr	INTEGER	Angabe zur Reihenfolge der Teildatensätze

In der Tabelle Tds ist eine Hierarchie der Teildatensätze definiert. Der Ausgangspunkt („root“) für die Teildatensatzhierarchie eines Datensatzes (z. B. QS-Filter-Datensatz) ist immer der Basisteildatensatz (daraus folgt: Teildatensatz<sup>12</sup> fkTds = NULL in der Tabelle Tds). Über die in den restlichen Teildatensätzen des Moduls definierten Bezüge zu den Mutterteildatensätzen und Relationstypen lässt sich ein Hierarchiebaum der Teildatensätze aufbauen.

Jeder Datensatz besteht aus

- genau einem Basisteildatensatz
- ggf. einem oder mehreren weiteren Teildatensätzen (= Kindteildatensätze)

### Beispiel:

Der QS-Filter-Datensatz DATENSATZ\_301 besitzt den Basisteildatensatz FALL und die sechs Kindteildatensätze DIAG, PROZ und ENTGEILT

### Felder der Teildatensätze

Die Felder eines Teildatensatzes sind in der Tabelle TdsFeld definiert. Jedes Feld eines Teildatensatzes (kurz TDS-Feld) ist eindeutig charakterisiert durch die Zugehörigkeit zu einem Teildatensatz und zum referenzierten Feld. Jedes Feld darf also nur einmal in einem Teildatensatz verwendet werden. Listenfelder erfordern einen Wert > 1 beim Attribut elemente.

### Felder

Ein Feld wird eindeutig über seinen technischen Namen definiert. Jedes Feld hat einen Basistyp (z. B. SCHLUESSEL, NUMSCHLUESSEL, ZAHL). Schlüsselfelder erfordern zusätzlich einen Schlüssel (siehe Tabelle 11).

<sup>12</sup> Es darf nur eine Definition eines Basisteildatensatzes existieren.

## Basistypen

Das Hauptmerkmal eines Basistyps ist der technische Typ eines Eingabefeldes (z. B. Zeichenkette, numerischer Typ, Datum usw.). Weiteres Charakteristikum ist die Beschreibung des Eingabeformats. Die Basistypen sind Voraussetzung für die Beschreibung einer formalen Regelsyntax. Das identifizierende Merkmal eines Basistyps ist sein technischer Name (Attribut `name`).

Tabelle 10: Struktur der Tabelle BasisTyp

Feldname	Datentyp	Bemerkung
<code>idBasisTyp</code>	INTEGER	Primärschlüssel
<code>name</code>	KURZER TEXT	Technischer Name (muss eindeutig sein)
<code>bezeichnung</code>	KURZER TEXT	Beschreibender Text
<code>formatAnweisung</code>	KURZER TEXT	Regulärer Ausdruck für die Formatprüfung



### Hinweis

In Zeichenketten (Basistyp `TEXT`) sind alle Zeichen des ASCII-Formats mit einem Kode  $\geq 32$  erlaubt. Ausgenommen sind das Semikolon, die doppelten Anführungsstriche und Hochkomata.

- Es gibt zwei Arten von Schlüsseln: numerische und nichtnumerische.
- Das Komma trennt die Nachkommastellen, Vorzeichen + und – sind erlaubt.
- Das Datumstrennzeichen ist der Punkt.

## Schlüssel

Identifizierendes Merkmal eines Schlüssels ist sein technischer Name. Die meisten Schlüsselcodes sind in der Tabelle `SchlüsselWert` (Tabelle 12) definiert. Externe Schlüsselkataloge (z. B. OPS, EBM oder ICD10) sind von den entsprechenden Anbietern zu beziehen.

Tabelle 11: Struktur der Tabelle Schluessel

Feldname	Datentyp	Bemerkung
<code>idSchluessel</code>	INTEGER	Primärschlüssel
<code>name</code>	KURZER TEXT	Technischer Name (muss eindeutig sein)
<code>bezeichnung</code>	KURZER TEXT	Beschreibender Text

Feldname	Datentyp	Bemerkung
extern	BOOLEAN	Zeigt an, ob der Schlüssel in der Tabelle Schluessel oder in einer externen Tabelle gespeichert ist.
externVerweis	KURZER TEXT	Verweis auf externe Quelle des Schlüsselkataloges
zahl	BOOLEAN	Wenn WAHR, wird das Attribut code der zugehörigen Schlüsselwerte als ganze Zahl interpretiert, ansonsten als Zeichenkette.

Schlüsselkodes können auf zwei Arten kodiert werden. Wenn das Attribut `zahl = WAHR`, so werden die Kodes als ganze Zahl interpretiert. Ansonsten werden sie als Zeichenketten angesehen. In der Syntax der Auslösebedingungen werden die letztgenannten Kodes in einfache Hochkommata gesetzt.

### Beispiel:

Attribut `zahl` bei Schlüsselfeldern

Felder des Basistyps NUMSCHLUESSEL haben das Attribut `zahl = TRUE`.

Felder des Basistyps SCHLUESSEL haben das Attribut `zahl = FALSE`. Es handelt sich um alphanumerische Schlüssel, die Buchstaben, Ziffern oder Sonderzeichen verwenden (z. B. `ypN0`). Hierbei kann es sich auch um Werte handeln, die lediglich Ziffern verwenden, aber mit führender Null beginnen (z. B. `01`).

### Externe Schlüsselkataloge

Externe Schlüsselkataloge sind über das Attribut `extern` deklariert. Externe Schlüsselkataloge werden nicht vom IQTIG bereitgestellt und daher auch nicht verantwortet.

Hinweise zu den Bezugsquellen sind in der Spalte `externVerweis` zu finden (z. B. <http://www.bfarm.de>). Ein Verweis auf eine Bezugsquelle kann unabhängig vom Attribut `extern` angegeben werden (siehe Schlüssel JN).



### Hinweis

Das Feld PERSONENKREIS referenziert zwei externe Quellen, da dieses Feld sowohl die Schlüsselwerte nach § 301 abbildet. Es gelten die folgenden externen Verweise:

- besonderer Personenkreis gemäß Anlage 2 zur Vereinbarung nach § 301 Abs. 3 SGB V (Taxvariable DATENSATZ\_301:FALL:PERSONENKREIS)  
externer Verweis: § 301-Vereinbarung: <http://www.dkgev.de>,



### Achtung

---

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

---

## Schlüsselwerte

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

Tabelle 12: Struktur der Tabelle SchluesselWert

Feldname	Datentyp	Bemerkung
<code>idSchluesselWert</code>	INTEGER	Primärschlüssel
<code>fkSchluessel</code>	INTEGER	Fremdschlüssel zur Tabelle Schlüssel
<code>code</code>	INTEGER	0, 1, 2 ...
<code>bezeichnung</code>	KURZER TEXT	Textliche Definition des Schlüsselwertes



### Hinweis

Die Schlüsselwerte lassen sich am einfachsten über die Abfrage `Schlüsselkodes` ermitteln.

### 1.2.3 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. Funktionen können nicht nur in Regeln, sondern auch zur Berechnung von Exportfeldern genutzt werden (Abschnitt B 2.5.3).

Der aktuelle Stand der in der Syntax verwendeten Funktionen ist in der Tabelle `SyntaxFunktion` der Spezifikation zu finden. In der QSDOK-Spezifikationsdatenbank der einrichtungsbezogenen QS-Dokumentation sind derzeit keine Funktionen hinterlegt. In der QSF-Spezifikationsdatenbank der einrichtungsbezogenen QS-Dokumentation finden verschiedene Funktionen Anwendung.

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<sup>13</sup>:**

DATUM aktuellesDatum()

Funktion ohne Übergabeparameter und mit Ergebnistyp DATUM

DATUM Minimum (DATUM DATUMLISTE)

Funktion mit Ergebnis vom Typ DATUM, die das Minimum einer Liste von Datumsangaben (DATUMLISTE) liefert.

JAHRDATUM jahreswert (DATUM EINDATUM)

Funktion mit Ergebnis vom Typ JAHRDATUM

Es kommen auch verschachtelte Funktionsaufrufe (z. B. funktionA(funktionB())) oder arithmetische Ausdrücke als Funktionsargumente (z. B. funktion(x+y)) vor. Häufig wird nur die Signatur von Funktionen bereitgestellt.

**Hinweise für die Implementierung von Funktionen**

Als Hilfestellung für die Ausprogrammierung wird bei manchen Funktionen ein Pseudocode bereitgestellt. Der Pseudocode ergänzt die Syntax der Plausibilitätsregeln um folgende Sprachelemente:<sup>14</sup>

- Befehlszeilen werden mit Semikolon abgeschlossen ;
- Wertzuweisungen mit dem Operator :=

A := B + C;

- Auswahlanweisungen

if (<Bedingung>) {

...

}

else {

...

}

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

- Blöcke werden durch geschweifte Klammern definiert.

{

...

<sup>13</sup> Die im Beispiel genannten Felder sind NICHT Bestandteil der Spezifikationsdatenbank für den QS-Filter und dienen an dieser Stelle lediglich der Erläuterung.

<sup>14</sup> Der Pseudocode erhebt nicht den Anspruch auf formale Korrektheit.

}

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

#### 1.2.4 OPS-Listen

Die Struktur der Tabellen `OPSListe` und `OPSWert` entsprechen den gleichnamigen Tabellen der Spezifikationsdatenbank für die QS-Dokumentation. Aufgrund des Einrichtungsbezugs sind für das Verfahren SEPE nur die entsprechenden Tabellen in der Spezifikationsdatenbank für den QS-Filter relevant.

#### 1.2.5 ICD-Listen

Die Struktur der Tabellen `ICDListe` und `ICDWert` entsprechen den gleichnamigen Tabellen der Spezifikationsdatenbank für die QS-Dokumentation. Aufgrund des Einrichtungsbezugs sind für das Verfahren SEPE nur die entsprechenden Tabellen in der Spezifikationsdatenbank für den QS-Filter relevant.

#### Umgang mit Kennzeichen zur Diagnosesicherheit

Das Zusatzkennzeichen zur Diagnosesicherheit ist in den ICD-Listen der Spezifikationsdatenbanken nicht enthalten.

ICD-Kodes mit dem Zusatzkennzeichen A (Ausschluss) sind bei der Auslösung nicht zu berücksichtigen. Bei der Auslösung von ambulanten Fällen ist die Auslösung nur mithilfe von gesicherten Einschlussdiagnosen zulässig.



#### Achtung

#### Auslösung

Bei einer prozessbegleitenden Auslösung über eine QS-Filtersoftware ist sicherzustellen, dass Aktualisierungen im KIS zwischen der Anwendungssoftware und der QS-Dokumentationssoftware kommuniziert werden. Grundsätzlich ist hier eine Übereinstimmung aller Diagnosen mit der Entlassungsanzeige des Behandlungsfalls sicherzustellen.

ICD-Kodes mit dem Zusatzkennzeichen A (Ausschluss) sind bei der Prüfung auf Ausschlusslisten nicht zu berücksichtigen. Im QS-Filter-Algorithmus wird die Syntaxvariable `DIAG` daher daraufhin geprüft, ob es sich um eine gesicherte Diagnose handelt. Hierfür wird die folgende Teilbedingung verwendet:

```
formatListe(DIAG; ' [a-zA-Z] [0-9]{2} (\.\.[0-
9]{1,2})? ([#\t\*\+\!])? ([AVZavz]) {1} ([RLBrlb]) ?' ) ;FALSCH)
KEINSIN <Ausschlussliste>
```

### 1.2.6 FAB-Listen

Jede FAB-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `FABListe`), welcher nach folgendem Schema gebildet wird:

```
{<TEXT>} FAB{<TEXT>}
```

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

### 1.2.7 EBM-Listen

Jede EBM-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `EBMListe`), welcher nach folgendem Schema gebildet wird:

```
{<TEXT>} EBM{<TEXT>}
```

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

### String-Vergleich bei EBM-Kodes

EBM-Ziffern können auf Landesebene um weitere Zeichen ergänzt werden. Das Feld Gebührenordnungsziffer gemäß EBM-Katalog [EBM] hat daher die Feldlänge 7.

In der Spezifikationsdatenbank sind die EBM-Kodes in Tabelle `EBMWert` fünfstellig – ohne ergänzende Zeichen – definiert. Für die Prüfung, ob zwei Kodes identisch sind, genügt kein einfacher Stringvergleich. Stattdessen wird ein Stringvergleich der Normkodes<sup>15</sup> durchgeführt, um die Übereinstimmung zwischen dem dokumentierten Kode und dem einer EBM-Liste zu ermitteln.

### 1.2.8 Entgelt-Listen

Jede Entgelt-Liste ist charakterisiert durch ihren Namen (Attribut `name` in Tabelle `EntgeltListe`), welcher nach folgendem Schema gebildet wird:

```
<Fallart>{<TEXT>} ENTGELT{<TEXT>}
```

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

### 1.2.9 Versionsverwaltung

Jede Spezifikationsdatenbank hat eine Version. Die Versionsinformation ist in der Tabelle `Version` der Eintrag, welcher den Attributwert `gueltig = WAHR` besitzt.

Die wichtigsten Eigenschaften einer Version sind der Versionsname (Attribut `name`) und der Gültigkeitszeitraum (Attribute `ab` und `bis`).

---

<sup>15</sup> Jeder EBM-Kode lässt sich entweder als Kode mit ergänzenden Zeichen (Normcode + ergänzende Zeichen) oder als Kode ohne ergänzende Zeichen (Normkode) darstellen.



### Achtung

Der Gültigkeitszeitraum einer Version ist in der fallbezogenen QS-Basis spezifikation für Leistungserbringer in der Regel ein Spezifikationsjahr (z. B. Aufnahme zwischen dem 01.01.2026 und dem 31.12.2026).

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. Darüber hinaus verwaltet die Tabelle `Version` die Historie der Versionen des aktuellen Spezifikationsjahres<sup>16</sup>: Welche Vorgängerversion vorher gültig war, kann über das Attribut `fkVersion` ermittelt werden. Für die erste hinterlegte Version des Spezifikationsjahrs ist das Attribut `fkVersion` leer.

Tabelle 13: Struktur der Tabelle Version

Feldname	Datentyp	Bemerkung
<code>idVersion</code>	INTEGER	Primärschlüssel
<code>name</code>	KURZER TEXT	Technischer Name der Version (Identifizierendes Attribut)
<code>bezeichnung</code>	KURZER TEXT	Beschreibender Text
<code>ab</code>	DATUM	Anfang des Gültigkeitszeitraumes
<code>bis</code>	DATUM	Ende des Gültigkeitszeitraumes
<code>pub</code>	DATUM	Datum der Publikation
<code>gueltig</code>	BOOLEAN	gültige Version (nur ein Eintrag darf als gültig markiert sein)
<code>fkVersion</code>	INTEGER	Bezug zur Vorgängerversion
<code>fkVersStatus</code>	CHAR	Bezug zum Status einer Version (Tabelle <code>VersStatus</code> ): E = in Entwicklung F = finale Version U = Update der finalen Spezifikation

Das Attribut `Modul.fkVersion` wird verwendet, um die aktuelle Version eines Moduls kenntlich zu machen.

#### 1.2.10 Meta-Tabellen

In den Tabellen `TabellenStruktur` und `TabellenFeldStruktur` werden die Tabellen und ihre Attribute aufgelistet. Der Inhalt wird automatisch generiert (Abschnitt B 2.6.2).

<sup>16</sup> Die Inhalte der Vorversionen sind nicht Teil der aktuellen Spezifikationsdatenbank.

### 1.2.11 DB-Änderungen gegenüber der Vorgängerversion

Die Tabellen DeltaNeu, DeltaGeloescht und DeltaAttribut zeigen die Änderungen zur letzten gültigen Spezifikation des Vorjahres und zur Vorversion an (Abschnitt B 2.6.2).

## 1.3 Der QS-Filter-Datensatz

Verbindlich für ein Spezifikationsjahr ist die Datenfeldbeschreibung, welche in der gültigen Spezifikationsdatenbank definiert ist. Hier wird der Datensatz DATENSATZ\_301 dargestellt. Dieser bildet den entsprechenden QS-Filter-Eingangsdatensatz ab (Tabelle 14).

Tabelle 14: Ausschnitt der Tabelle Ds

<b>idDs</b>	<b>name</b>	<b>Bezeichnung</b>
1	DATEN-SATZ_301	Datenübermittlung nach § 301 Abs. 3 SGB V, ergänzt um die Festlegungen der Vereinbarung nach § 120 Abs. 3 SGB V



### Achtung

#### Datenverlust durch Fehlbezug und falsche Falleinschränkung

Die Bezeichnungen des Datensatzes sowie deren Felder nehmen Bezug auf § 301 SGB V. Diese Gesetzesgrundlagen beziehen sich lediglich auf den Kontext der Gesetzlichen Krankenversicherung. Der QS-Filter-Datensatz hingegen bezieht sich prinzipiell auf alle behandelten Patienten/Fälle.

Da in das DeQS-Verfahren SEPSIS nur gesetzlich versicherte Patienten einbezogen werden, kann eine Dokumentationspflicht nur identifiziert werden, wenn das Institutionskennzeichen der Krankenkasse der Versichertenkarte mit der Zeichenkette '10' beginnt.

### 1.3.1 Der QS-Filter-Eingangsdatensatz

Der QS-Filter-Eingangsdatensatz ist je nach Richtlinien- und Abrechnungskontext zu wählen:

**DATENSATZ\_301 enthält den QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V**

Die Dokumentationspflicht einer Einrichtung wird ausgelöst, wenn mindestens ein Fall den definierten Auslösebedingungen entspricht. Die für den QS-Filter maßgeblichen administrativen und medizinischen Datenfelder zur Identifikation dokumentationspflichtiger Fälle im Rahmen der DeQS-RL für durch ein Krankenhaus erbrachte Fälle finden sich im Aufnahmedatensatz bzw. in der Entlassungsanzeige des § 301-Datenformats<sup>17</sup>. Daher ist der QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V (DATENSATZ\_301) so beschaffen, dass fast alle seine Datenfelder aus der Struktur des § 301-Datensatzes abgeleitet werden können. Einzige Ausnahmen sind die Felder PATALTER und FALLART (lassen sich aus Inhalten des § 301-Datensatzes berechnen).

Der QS-Filter-Eingangsdatensatz nach § 301 Abs. 3 SGB V besteht aus folgenden Teilen:

- Behandlungsfall (Teildatensatz FALL)

<sup>17</sup> Datenübermittlung nach § 301 Abs. 3 SGB V, ergänzt um die Festlegungen der Vereinbarung nach § 120 Abs. 3 SGB V

- Prozedurangaben des Behandlungsfalls (Teildatensatz PROZ, wiederholbar)
- Diagnoseangaben des Behandlungsfalls (Teildatensatz DIAG, wiederholbar)
- Entgeltangaben des Behandlungsfalls (Teildatensatz ENTGELT, wiederholbar)

Die hier definierten Teildatensätze finden sich im § 301-Datensatz bzw. dem DRG-Datenformat nach § 21 KHEntgG<sup>18</sup> wieder. Der QS-Filter-Eingangsdatensatz basiert zum größten Teil auf einer Teilmenge der Inhalte dieser Datensätze.

Tabelle 15: Felder des QS-Filter-Eingangsdatensatzes nach § 301

Feld	Beschreibung	M/K	Zeichen-länge	BasisTyp
<b>Behandlungsfall (FALL)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
FALLART	Fallart 1 : ENTGELTARTAMB <> LEER UND ENTGELTARTSTAT = LEER 2 : ENTGELTARTSTAT <> LEER UND ENTGELTARTAMB = LEER	M	1	NUMSCHLUESSEL (Fallart)
AUFNGRUND	Aufnahmegrund gem. § 301 SGB V	K	2	SCHLUESSEL (AufnGrund)
AUFNDATUM	Datum der Aufnahme Krankenhaus	K	10	DATUM
ENTLDATUM	Datum der Entlassung Krankenhaus	K	10	DATUM
PATALTER	Alter in Jahren am Aufnahmetag	M	3	GANZEZAHL
ENTLGRUND	Entlassungsgrund § 301	K	2	SCHLUESSEL (EntlGrund)
KASSEIKNR	Institutionskennzeichen der Krankenkasse der Versichertenkarte	K	9	GANZEZAHL
VERSI-CHERTENIDNEU	eGK-Versichertennummer	K	10	TEXT
PERSONENKREIS	besonderer Personenkreis	K	1	SCHLUESSEL

<sup>18</sup> Vereinbarung über die Übermittlung von DRG-Daten nach § 21 KHEntgG. Stand: 30. November 2012. Siegburg: Institut für das Entgeltsystem im Krankenhaus (InEK gGmbH). Online verfügbar unter [http://www.gdrg.de/cms/Datenlieferung\\_gem.\\_21\\_KHEntgG/Dokumente\\_zur\\_Datenlieferung/Datensatzbeschreibung](http://www.gdrg.de/cms/Datenlieferung_gem._21_KHEntgG/Dokumente_zur_Datenlieferung/Datensatzbeschreibung).

Feld	Beschreibung	M/K	Zeichen-länge	BasisTyp
				(Personenkreis)
<b>Prozedurangaben (wiederholbar) (PROZ)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
OPDATUM	Datum der Operation	M	10	DATUM
OPS	Prozedurschlüssel (OPS-Kode)	M	13	SCHLUESSEL(OPS)
<b>Diagnoseangaben (wiederholbar) (DIAG)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
ICD	Diagnoseschlüssel (ICD-Kode)	M	9	SCHLUESSEL(ICD)
DIAGART	Diagnoseart	M	2	SCHLUESSEL(DiagArt)
SEKDIAG	Sekundär-Diagnoseschlüssel (Sekundär-Kode)	K	9	SCHLUESSEL(ICD)
<b>Entgeltangaben (wiederholbar) (ENTGELT)</b>				
FALLNUMMER	Fallnummer	M	15	TEXT
ENTGELTARTSTAT	Entgeltschlüssel stationär gem. § 301 SGB V (1.-3. Stelle)	K	3	NUMSCHLUESSEL(EntgeltSchluesselStat)
EBMDATUM	Leistungstag (Entgelt)	K	10	DATUM

In dem Datenfeld ENTGELTARTSTAT sind die Kodes zu dokumentieren, welche im Schlüssel Entgeltarten des Datenübermittlungsverfahrens nach § 301 Abs. 3 SGB V definiert sind<sup>19</sup>.

Für den externen Schlüssel EntgeltSchluesselStat sind dreistellige Schlüsselwerte hinterlegt.

### Primär und Sekundärdiagnosen

Für das Modul sind Einschlussdiagnosen nach Primär- und Sekundärdiagnosen im Sinne der Kreuz-Stern-Notation zu differenzieren. Die Zusammengehörigkeit bestimmter Sekundärdiagnosen zu bestimmten Primärdiagnosen ist hierbei zu berücksichtigen. Die in den Auslösekriterien referenzierten Felder DIAG bzw. HDIAG beinhalten weiterhin sowohl Primär- als auch Sekundärdiagnosen.

<sup>19</sup> Aktuelle Version abrufbar im Download-Bereich der DKG-Homepage (<http://www.dkgev.de>)

## 1.4 Der Algorithmus zur Ermittlung der Dokumentationspflicht

Dieser Abschnitt beschreibt den Algorithmus zur Ermittlung der Dokumentationsverpflichtung von Modulen und die erforderlichen Fehlerprüfungen.

### 1.4.1 Einleitung und Überblick

Für jedes Modul ist ein formaler, logischer Ausdruck definiert, welcher sich aus einer verfahrensbezogenen Teilbedingung (Abschnitt B 1.4.2) und einer administrativen Teilbedingung (Abschnitt B 1.4.3) zusammensetzt:

Auslösebedingung für ein Modul = `ModulAusloeser.bedingung UND AdminKriterium.bedingung`

Wenn eine der beiden Teilbedingungen nicht erfüllt ist, so löst der QS-Filter für das jeweilige Modul keine Dokumentationspflicht aus.

Für jeden Fall evaluiert der QS-Filter-Algorithmus sämtliche der in der Spezifikationsdatenbank hinterlegten Auslösebedingungen. Wird die Auslösebedingung erfüllt, so ist der Fall für das verknüpfte Modul dokumentationspflichtig.



#### Achtung

Generell gilt die Einschränkung, dass ein gleiches Modul pro Einrichtung nur einmal ausgelöst wird. Es können aber mehrere verschiedenartige Module ausgelöst werden.

### Dokumentationspflicht einer Einrichtung

Eine Einrichtung ist dokumentationspflichtig, wenn für sie mindestens ein QS-Modul dokumentationspflichtig ist. Ein QS-Modul wird als dokumentationspflichtig identifiziert, wenn mindestens ein Fall den definierten Auslösekriterien entspricht.

### Dokumentationspflicht eines Moduls

Ein Modul (Datensatz) ist dann dokumentationspflichtig, wenn folgende Bedingungen zutreffen:

- Bei der Fehlerprüfung tritt kein Fehler auf.
- Die jeweilige modulspezifische Auslösebedingung ist erfüllt (Abschnitt B 1.4.2).
- Die administrativen Bedingungen sind erfüllt (Abschnitt B 1.4.3).

## 1.4.2 Verfahrensbezogene Einschlusskriterien

### Überblick:

Für jede QS-Datensatz-Definition (= Modul) sind in der Tabelle `ModulAusloeser` eine oder mehrere Auslösebedingungen (Attribut `bedingung`) hinterlegt.

### Definition:

Jede Auslösebedingung der Tabelle `ModulAusloeser` definiert einen verfahrensspezifischen QS-Filter-Modulauslöser.

Ein QS-Filter-Modulauslöser bildet somit ein Komplex von Leistungen ab, welche über den QS-Filter-Algorithmus zu einer Dokumentationsverpflichtung führen.



### Achtung

#### Einschlusskriterien mit KASSEIKNR, VERSICHERENIDNEU und PERSONENKREIS

Verfahrensbezogene Einschlusskriterien können die Felder KASSEIKNR, VERSICHERENIDNEU und PERSONENKREIS enthalten, mithilfe dessen geprüft wird, ob es sich um einen GKV-Versicherten handelt. Liegt die elektronische Gesundheitskarte (eGK) bei der Prüfung der definierten Kriterien noch nicht vor, erlischt **nicht** die Dokumentationspflicht. Die Auslösebedingung ist bei Vorliegen der Angaben KASSEIKNR/VERSICHERENIDNEU/PERSONENKREIS/eGK erneut zu prüfen.

Ein QS-Filter-Modulauslöser identifiziert in Kombination mit dem administrativen Kriterium die Dokumentationspflicht für immer genau einen QS-Datensatz.

- Die QS-Filter-Modulauslöser SEPE lösen jeweils den QS-Datensatz SEPE aus.

#### Definition in der Datenbank

Tabelle 16 gibt einen Überblick über die Tabelle ModulAusloeser der Spezifikationsdatenbank.

Tabelle 16: Struktur der Tabelle ModulAusloeser

Feldname	Datentyp	Bemerkung
idModulAusloeser	INTEGER	Primärschlüssel
Name	KURZER TEXT	Technischer Name des QS-Filter-Modulauslösers
bezeichnung	KURZER TEXT	Bezeichnungstext der Auslösebedingung
bedingung	LANGER TEXT	Auslösebedingung für den QS-Filter-Modulauslöser
textDefinition	LANGER TEXT	Medizinisch-inhaltliche Definition bzw. Erläuterung der Auslösebedingung
fkModul	INTEGER	Bezug zum Modul
verpflichtend	BOOLEAN	Wenn WAHR, ist das betreffende Modul bundesweit verpflichtend
fkAdminKriterium	INTEGER	Bezug zu den administrativen Abgrenzungskriterien
fkDs	INTEGER	Bezug zum DATENSATZ_301 oder zum DATENSATZ_295
internBild	KURZER TEXT	Bild zum Modul

Feldname	Datentyp	Bemerkung
ambulant	BOOLEAN	Ist die Fallart ambulant (bezogen auf Fälle einer Einrichtung)?
stationär	BOOLEAN	Ist die Fallart stationär (bezogen auf Fälle einer Einrichtung)?
krankenhausabrechnung	BOOLEAN	Wird der Fall stationär abgerechnet (bezogen auf Fälle einer Einrichtung)?
selektiv	BOOLEAN	Wird der Fall ambulant selektivvertraglich abgerechnet (bezogen auf Fälle einer Einrichtung)?
kollektiv	BOOLEAN	Wird der Fall ambulant kollektivvertraglich abgerechnet (bezogen auf Fälle einer Einrichtung)?

### 1.4.3 Administrative Einschlusskriterien

#### Überblick

Die administrativen Regelungen beschreiben diejenigen Auslösebedingungen, welche über die modulspezifischen ICD/OPS/EBM-Kriterien (Abschnitt B 1.4.2) hinausgehen. Moduldokumentationen werden für eine Einrichtung (Prüfung auf Fallebene) nur dann ausgelöst, wenn

1. der Patient stationär aufgenommen wird,
2. der Behandlungsfall keine Verbringungsleistung für Dritte ist.

#### Zu 1.

Stationär aufgenommen: Die Bedingung wird aus der Angabe im Datenfeld Aufnahmegrund (Abschnitt B 1.3.1) abgeleitet:

```
AUFNGRUND NICHTIN ('03';'04') UND AUFNGRUND <> LEER
```

Diese Teilbedingung ist Teil der administrativen Kriterien (siehe Attribut `bedingung` in Tabelle 17). Der zeitliche Rahmen wird bei stationären Fällen über das Aufnahme- und das Entlassungsdatum definiert.

#### Zu 2.

Dieses Kriterium ist nicht aus ein oder mehreren Datenfeldern des QS-Filter-Eingangsdatensatzes ableitbar. Hier ist seitens des Krankenhauses für eine organisatorische Regelung zu sorgen.

#### Definition in der Datenbank

Tabelle 17 gibt einen Überblick über wichtige Attribute der Tabelle AdminKriterium, welche die administrativen Abgrenzungskriterien definiert.

Tabelle 17: Struktur der Tabelle AdminKriterium

Feldname	Datentyp	Bemerkung
idAdminKriterium	INTEGER	Primärschlüssel

Feldname	Datentyp	Bemerkung
name	KURZER TEXT	Technischer Name des administrativen Kriteriums
bedingung	LANGER TEXT	Administrative Auslösebedingung
info	KURZER TEXT	Bezeichnungstext des administrativen Kriteriums
sollBedingung	KURZER TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
sollBedingungUeberlieger	KURZER TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.

#### 1.4.4 Struktur und Syntax der Auslösebedingungen

##### Die Variablen der Auslösebedingungen

Die in den Auslösebedingungen erlaubten Variablen sind in der Tabelle `SyntaxVariable` definiert.

Die Variablennamen (Attribut `SyntaxVariable.name`) bestehen aus maximal 32 Zeichen. Sie 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. Namen von Operatoren wie `EINSIN`, siehe Tabelle 19).

##### Typen

Jede Variable hat einen Basistyp. Tabelle 18 gibt einen Überblick über die möglichen Basistypen.

Jeder der in der Tabelle `SyntaxVariable` definierten Variablen ist über den Wert des Attributs `SyntaxVariable.fkTdsFeld` ein Feld des QS-Filter-Eingangsdatensatzes zugeordnet. Jedes dieser Felder besitzt einen Basistyp.

Tabelle 18: Basistypen der Variablen

Basistyp	Bezeichnung	Beispiele (Literale)
BOOL	Boolesche Variable	WAHR, FALSCH
TEXT	Zeichenkette (String)	„Spezifikation“
GANZEZAHL	... -2, -1, 0, 1, 2, 3, ...	1
ZAHL	Zahl (mit oder ohne Nachkommastellen)	Erfassung: 25, 4 oder –100, 8 Export: 25.4 oder 100.8

<b>Basistyp</b>	<b>Bezeichnung</b>	<b>Beispiele (Literale)</b>
DATUM	Zehnstelliges Datum	'01.01.2013'
NUMSCHLUESSEL	Numerisch kodierter Schlüssel (wie GANZEZAHL)	0
SCHLUESSEL	Alphanumerischer Schlüssel	'19.1', '07'
UHRZEIT	Uhrzeit	'10:15'
JAHRDATUM	Jahresdatum	2017
QUARTDATUM	Quartalsdatum (ohne Angabe von Tagen oder Monaten)	'1/2017'

Die meisten Schlüsselwerte werden als GANZEZAHL kodiert, d. h., dass die Kodes nicht in Hochkommata gesetzt werden dürfen. Ausnahmen sind z. B. die OPS-Schlüssel (z. B. '5.282.0') oder die ICD-10-Schlüssel (z. B. 'C11.3').

### **Listen**

Eine Variable wird als Liste interpretiert, wenn der Wert des Attributs SyntaxVariable.istListe = WAHR ist.

### **Literale**

Alphanumerische Literale (z. B. SCHLUESSEL) werden von einfachen Hochkommata eingeschlossen, während Zeichenketten vom Basistyp TEXT in doppelte Anführungsstriche gesetzt werden müssen.

Dies gilt nicht für numerischen Literale (GANZEZAHL, ZAHL, NUMSCHLUESSEL) und Literale des Datentyps BOOL (Wahrheitswerte).

### **Diagnose-, Prozeduren-, Abrechnungs- und Fachabteilungslisten**

Fest definierte Listen von Diagnosen, Prozeduren oder Abrechnungsziffern werden in der Syntax ähnlich wie die Variable SEP\_OPS\_OD angesprochen.

Die erlaubten Namen von Diagnoselisten sind in der Tabelle ICDListe (Attribut name) zu finden. Analog sind die zulässigen Namen von Prozedurlisten in der Tabelle OPSListe (Attribut name) und von Abrechnungslisten in der Tabelle EBMListe bzw. EntgeltListe (Attribut name) definiert. Die Namen der Fachabteilungslisten sind in der Tabelle FABListe zu entnehmen.

Die jeweiligen Kodes (nach ICD-10-GM, OPS bzw. EBM) sind in den durch Fremdschlüsselbeziehungen verknüpften Tabellen ICDWert, OPSWert und EBMWert enthalten. Alle Kodes sind terminal. Die Diagnosekodes enthalten weder die Suffixe +, \* oder ! noch die Zusatzkennzeichen zur Diagnosesicherheit A, G, V und Z.

Die Kodes zu Entgelten und Fachabteilungen sind in den durch Fremdschlüsselbeziehungen verknüpften Tabellen EntgeltWert und FABWert enthalten.

## Die Operatoren der Auslösebedingungen

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

Tabelle 19: Präzedenz und Assoziativität der Operatoren<sup>20</sup>

Präzedenz	Assoziativität	Operator	Erläuterung
0	links	EINSIN	Mengenoperator „EINSIN“
	links	KEINSIN	Mengenoperator „KEINSIN“
3	links	<	Vergleichsoperator „kleiner“
	links	>	Vergleichsoperator „größer“
	links	<=	Vergleichsoperator „kleiner gleich“
	links	>=	Vergleichsoperator „größer gleich“
4	links	=	Vergleichsoperator „gleich“
	links	<>	Vergleichsoperator „ungleich“
5	rechts	NICHT	Logischer Operator „NICHT“
6	links	UND	Logischer Operator „UND“
7	links	ODER	Logischer Operator „ODER“

### Anmerkungen:

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)<sup>21</sup>.

---

### Beispiel:

Folgende Regel prüft, ob ein Element der Listenvariable `PROZ` einen bestimmten Kode besitzt:

---

```
PROZ EINSIN SEP_OPS_OD_BEATMUNG
```

---

<sup>20</sup> In dieser Übersichtstafel hat jeder einzelne Operator eine Präzedenzstufe (höchste Präzedenzstufe ist 0). Operatoren, welche die gleiche Stufe haben, werden nach den Regeln der Assoziativität aufgelöst.

<sup>21</sup> Dieser Operator ist redundant, da er auch durch Negation des `EINSIN`-Operators abgedeckt ist.

### String-Vergleich bei Diagnosen, Prozeduren und Abrechnungsziffern

Die Operation `EINSIN` basiert für Diagnosen, Prozeduren oder Abrechnungsziffern auf einem Vergleich von Zeichenketten (String-Vergleich): Für jedes Element der linken Liste wird über einen String-Vergleich geprüft, ob es in der rechten Liste enthalten ist.

Eine Modifikation des String-Vergleichs ist beim String-Vergleich von Diagnosen nötig: Hier werden die Suffixe `+`, `*` und `!` sowie die Zusatzkennzeichen `A`, `G`, `V` und `Z` ignoriert. Nicht ignoriert wird das Sonderzeichen `.` (und `-` bei OPS).



#### Achtung

In der QS-Filter-Datenbank sind nur terminale Kodes enthalten. Entsprechend muss sichergestellt sein, dass auch die Kodes aus dem Eingangsdatensatz terminal sind. Darauf ist besonders zu achten, wenn Daten ungeprüft aus Subsystemen übernommen wurden.

---

#### Beispiel:

Folgende Operationen führen zum gleichen Ergebnis

```
('Z37.9') EINSIN ('Z37.9';'Z37.0';'Z37.1';'Z37.2';'Z37.3')  
('Z37.9! ') EINSIN ('Z37.9';'Z37.0';'Z37.1';'Z37.2';'Z37.3')
```

---

Eine analoge Regelung gilt für die Seitenlokalisierungen der OPS-Kodes.

## 2 QS-Dokumentation

Die vorliegenden Spezifikationskomponenten für die QS-Dokumentation definieren die Vorgaben zur Datenerfassung, Plausibilitätsprüfung und Datenübermittlung für die externe vergleichende Qualitätssicherung im Rahmen der DeQS-RL. Diese sollen die Bereitstellung valider und vergleichbarer Daten gewährleisten. Neben der Datenbank zur QS-Dokumentation zählen zu den Spezifikationskomponenten der QS-Dokumentation die Ausfüllhinweise und die Dokumentationsbögen. Die Delta-Tabellen in der Datenbank zur QS-Dokumentation stellen geänderte, gelöschte und neue Datenbankinhalte im Vergleich zur letzten gültigen Version des Vorjahres sowie, sofern vorhanden, zur letzten Version des aktuellen Spezifikationsjahres dar.

### 2.1 Anmerkungen zur Struktur der Spezifikation zur QS-Dokumentation

Die Spezifikation zur QS-Dokumentation ist in einer relationalen Datenbank abgelegt. Zurzeit wird sie ausschließlich als Access-Datenbank (MS Access 2007-2016) zur Verfügung gestellt. Der Name der Spezifikation richtet sich nach folgendem Schema:

<Spezifikationsjahr>\_DeQS\_QSDOK\_EDOK\_V<Versionsnummer>.mdb

<Spezifikationsjahr> bezeichnet das Jahr, in dem die QS-Dokumentation stattfindet.

<Versionsnummer> bezeichnet die 2-stellige Versionsnummer (z. B. 01).

#### **Beispiel:**

Im Spezifikationsjahr 2026 ist die Spezifikation 2026\_DeQS\_QSDOK\_EDOK\_V01.mdb<sup>22</sup> gültig.

Weiterführende Erläuterungen zum Benennungsschema für Spezifikationskomponenten sind der Technischen Dokumentation für das Benennungsschema zu entnehmen. Die Tabellenstruktur der Spezifikationsdatenbank wird in Abschnitt B Komponenten beschrieben.

Folgende Attribute treten in vielen Tabellen auf:

- name ist in der Regel als technischer Name zu verstehen. Zum Beispiel wird Feld.name als Variablenname in den Plausibilitätsregeln verwendet.
- bezeichnung ist eine kurze Beschreibung. Zum Beispiel ist BogenFeld.bezeichnung der Text, welcher ein Feld auf einem Eingabeformular beschreibt.
- bedingung enthält einen logischen Ausdruck. Prominentester Vertreter dieses Attributtyps ist das Attribut bedingung in der Tabelle Regeln.

<sup>22</sup> Die Versionsnummer der gültigen Spezifikation (z. B. EDOK\_V01, EDOK\_V02 usw.) ist dem zuletzt veröffentlichten Update zu entnehmen.

## 2.2 Datenfeldbeschreibung

Für jedes Modul existiert eine eigene Datenfeldbeschreibung. Sie spezifiziert alle auszufüllenden Datenfelder (Bogenfelder, auch Items genannt) und besteht aus mehreren Tabellen (Abbildung 4), die in den nachfolgenden Abschnitten erläutert werden.

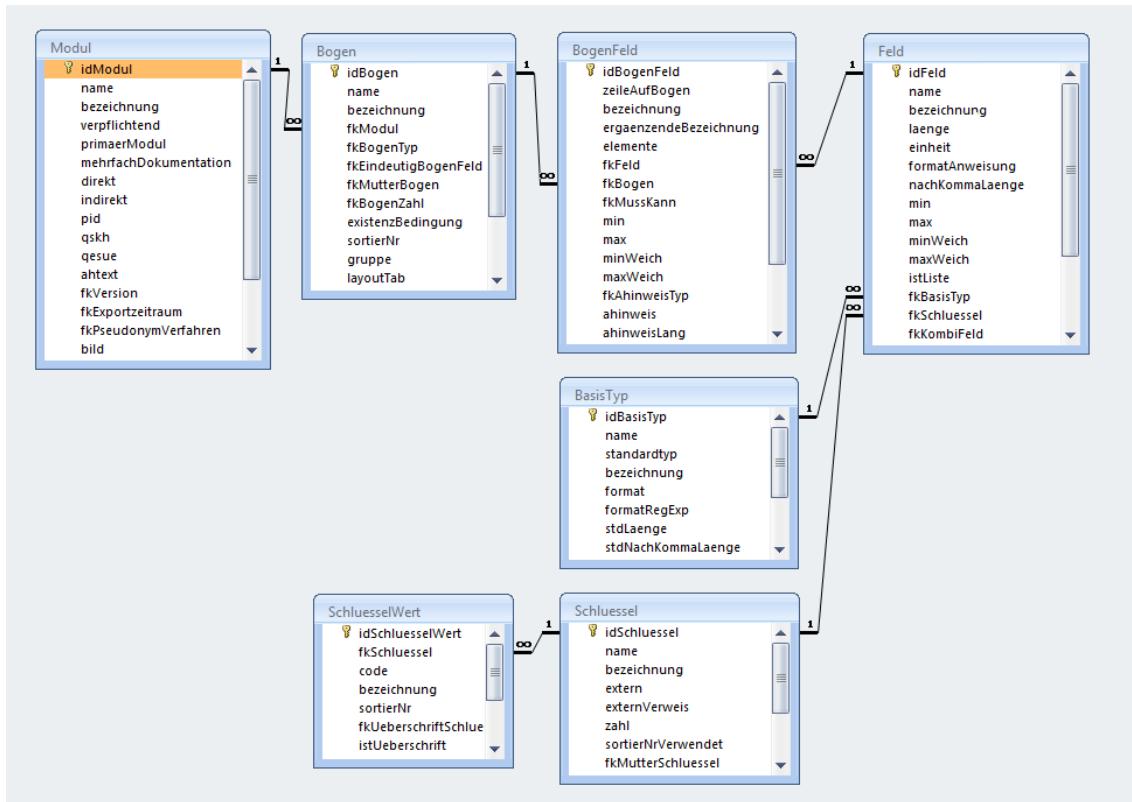


Abbildung 4: Tabellen und Relationen der Datenfeldbeschreibung

Die Abfragen `Datenfeldbeschreibung` und `DatenfeldbeschreibungFürEinModul` der Access-Datenbank ermöglichen den Überblick über diese Struktur.

Die Beschreibung der Datenfelder hat folgende Ziele:

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

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

Die Datenfeldbeschreibung ist teilweise auf dem jeweiligen Dokumentationsbogen eines Moduls („Bogensicht“) abgebildet. Zu beachten ist dabei, dass die „Bogensicht“ lediglich die Sicht der medizinischen Fachgruppen, die die Module entwickeln, darstellt. Bei verteilten Softwarelösungen<sup>23</sup>

<sup>23</sup> Die Kassenärztlichen Vereinigungen als DAS stellen den Vertragsärzten in ihrem Zuständigkeitsbereich Anwendungen zur Erfassung und Übertragung von Daten der einrichtungsbezogenen QS-Dokumentation, z. B.

für einen Leistungserbringer hingegen ist die Bogensicht dann nicht mehr adäquat, wenn die Bestandteile eines Bogens auf verschiedene Teilsysteme verteilt sind. Die Daten eines Bogens werden in diesen Fällen für den Export aus den einzelnen Teilsystemen zusammengestellt.



### Hinweis

Die Papierbögen werden lediglich zu Illustrationszwecken zur Verfügung gestellt. Sie sind zur Dokumentation nicht zugelassen.

Im Kontext einer integrierten, prozessorientierten Dokumentationssoftware müssen die Teildatensätze nicht direkt in Eingabeformulare umgesetzt werden. Es ist sinnvoller, die Teile eines Dokumentationsbogens zu dem Zeitpunkt und in dem Dokumentationskontext zu erfragen, der sich in den Prozessablauf<sup>24</sup> eines Leistungserbringers einordnet.

#### 2.2.1 Dokumentationsmodule (Datensätze)

Ein Modul der Spezifikation enthält die Datensatzdefinition von einem oder mehreren Verfahren. Abhängig von (inhaltlich oder organisatorisch) abzugrenzenden Bereichen kann ein Verfahren mehrere Module umfassen. Für eine Einrichtung kann eine oder mehrere Moduldokumentationen angelegt werden, die nach Dokumentationsabschluss an die Datenannahmestelle übermittelt werden. Fehlerfreie Moduldokumentationen (verkürzt „Module“), die die Basis der Datenauswertungen bilden, werden dem Leistungserbringer von der Datenannahmestelle bestätigt. Aus technischer Sicht ist ein Modul durch einen eindeutigen Namen gekennzeichnet. Es umfasst mindestens einen Teildatensatz. In der Tabelle `Modul` der QS-Spezifikation finden sich die zentralen Definitionen eines Moduls.

Tabelle 20: Struktur der Tabelle `Modul`

Feldname	Datentyp	Bemerkung
<code>idModul</code>	INTEGER	Primärschlüssel
<code>name</code>	KURZER TEXT (32)	Eindeutiger technischer Name
<code>bezeichnung</code>	KURZER TEXT (255)	Erläuternde Bezeichnung
<code>verpflichtend</code>	BOOLEAN	Besteht für das Modul eine QS-Dokumentationsverpflichtung?
<code>primaerModul</code>	BOOLEAN	Ist das Modul ein Primärmodul?
<code>mehrfachDokumentation</code>	BOOLEAN	Ist ein mehrfaches Anlegen eines gleichartigen Datensatzes pro Einrichtung zulässig (WAHR / FALSCH)?  Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.

webbasiert, zur Verfügung (ggf. mit Unterstützung durch die Kassenärztliche Bundesvereinigung). Im Krankenhaussektor wird von Softwareherstellern eine QS-Dokumentationssoftware umgesetzt.

<sup>24</sup> Zum Beispiel in den Prozessablauf eines Krankenhauses.

Feldname	Datentyp	Bemerkung
		Eine Mehrfachdokumentation ist jederzeit unzulässig.
ahtext	LANGER TEXT	Einleitender Text für den Ausfüllhinweis eines Moduls
fkVersion	INTEGER	Gültige Version des jeweiligen Moduls
fkExportzeitraum	INTEGER	Exportzeitraum (Exportzeitraum.exportBis) eines Moduls
Bild	KURZER TEXT (20)	Modulspezifisches Bild

In der Tabelle Modul wird der Exportzeitraum für ein Modul angegeben (Attribut Modul.fkExportzeitraum). Die hinterlegten Zeitpunkte sind in der Tabelle Exportzeitraum (Attribut exportBis) definiert.

### Auslösung der Moduldokumentation

Der auslösende Sachverhalt für die Dokumentationspflicht ist in der Spezifikation für den QS-Filter definiert. Dieser greift zu diesem Zweck unter anderem auf ICD- und OPS-Kodes (z. B. Haupt- und Nebendiagnosen und Prozeduren) zurück, die z. B. im Krankenhausinformationssystem (KIS) verfügbar sind. Die Dokumentationspflicht der Module des QS-Verfahrens SEPSIS wird unter anderem über Einschlussdiagnosen ermittelt, die nach Primär- und Sekundärdiagnosen im Sinne der Kreuz-Stern-Notation zu differenzieren sind. Die Zusammengehörigkeit bestimmter Sekundärdiagnosen zu bestimmten Primärdiagnosen ist bei der Auslösung zu berücksichtigen.

### Primärmodule

Für Primärmodule sind in der Spezifikation für QS-Filter-Software Auslösebedingungen definiert. In der vorliegenden Spezifikation der einrichtungsbezogenen QS-Dokumentation sind alle Module als Primärmodule gekennzeichnet.

#### 2.2.2 Teildatensätze

Die Begriffe „Teildatensatz“ und „Bogen“ werden synonym gebraucht. In den der Illustration dienten Dokumentationsbögen werden alle Teildatensätze aufgeführt.

#### Benennung von Teildatensätzen

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

<Modul.name>:<Bogen.name>

#### Beispiele:

SEPE:B ist der Basisbogen des Moduls Einrichtungsbezogene QS-Dokumentation Sepsis

## Bogentyp

Der Kerndatensatz besteht aus mindestens einem Basisteildatensatz und kann durch einen oder mehrere Teildatensätze ergänzt werden. Das Attribut `Bogen.fkBogenTyp` definiert für jeden Teildatensatz seine Rolle im und seine Zugehörigkeit zum Kerndatensatz. In Tabelle 21 sind die Bezeichnungen der einzelnen Bogentypen definiert.

Tabelle 21: Inhalte der Tabelle `BogenTyp`

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

Das Modul SEPE verfügt jeweils nur über einen Basisteildatensatz.

Weiterführende Erläuterungen zu Teildatensätzen sind in der Technischen Dokumentation zur Basisspezifikation wird Leistungserbringer zu entnehmen.

### 2.2.3 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>25</sup> wird so gewählt, dass sie einem medizinischen Experten unmittelbar verständlich ist. Die Spezifikation des Inhalts umfasst hingegen sowohl eine fachliche (medizinische) als auch datentechnische Typisierung. Dagegen repräsentieren die in der Tabelle `Feld` aufgelisteten Felder inhaltlich gleiche Dokumentationsfelder mehrerer Module (Abschnitt B 2.2.1), der datentechnische Typ (`BasisTyp`) charakterisiert das Format des Feldes (Abschnitt B 2.2.1).

Jedes Datenfeld hat zwingend einen Bezug zu einem Teildatensatz und zu einem technischen Feld. Weitere Eigenschaften sind die Bogenfeldbezeichnung und die fortlaufende Nummer im Teildatensatz. Die Datenfelder sind in der Tabelle `BogenFeld` gespeichert.

Identifizierendes Merkmal eines Datenfelds 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`) genau einen Primärschlüssel `idBogenFeld` liefert.

<sup>25</sup> Gegebenenfalls im Kontext der Überschriften (Abschnitt B 2.2.4).

Tabelle 22: Struktur der Tabelle *BogenFeld*

Feldname	Datentyp	Bemerkung
idBogenFeld	INTEGER	Primärschlüssel
zeileAufBogen	DOUBLE	Zeile auf dem Dokumentationsbogen.
gliederungAufBogen	KURZER TEXT	Gliederungsnummer, die im Dokumentationsbogen angezeigt wird
bezeichnung	KURZER TEXT	Beschreibender Text zum Feld auf dem Dokumentationsbogen.
ergaenzendeBezeichnung	KURZER TEXT	Optionale ergänzende Bezeichnung zu einem Bogenfeld.
elemente	INTEGER	Anzahl der Elemente bei Listenfeldern  Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation, da aktuell keine Listenfelder verwendet werden.
fkFeld	INTEGER	Fremdschlüssel zu dem Teildatensatz und zu dem Feld, bilden zusammen die identifizierenden Merkmale
fkBogen	INTEGER	
fkMussKann	KURZER TEXT (1)	M oder K, Unterscheidung zwischen Muss- und Kann-Feldern
min	DOUBLE	Harte Untergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
max	DOUBLE	Harte Obergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
minWeich	DOUBLE	Weiche Untergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
maxWeich	DOUBLE	Weiche Obergrenze des Wertebereichs eines numerischen Datenfeldes (modulspezifisch). Die Definition ist optional.
ahinweis	KURZER TEXT (32)	Name des HTML-Ausfüllhinweises ohne Endung .htm (Abschnitt B 2.2.5)
fkICDListe	INTEGER	Zuordnung einer ICD-Liste
fkOPSListe	INTEGER	Zuordnung einer OPS-Liste
berechnungsregel	TEXT	Formel zur Generierung des BogenFeldes, z. B. durch Aufruf einer Syntaxfunktion

## Muss- und Kann-Felder

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

- Ein Muss-Feld ( $\text{M}$ ) muss innerhalb eines angelegten Teildatensatzes immer ausgefüllt sein (Abschnitt B 2.2.2).<sup>26</sup>
- Kann-Felder ( $\text{K}$ ) sind optionale Felder.
- Abhängige Muss-Felder ( $\text{K}$ ) müssen nur unter bestimmten Bedingungen ausgefüllt werden. Wenn also logische Sachverhalte dem Ausfüllen von Kann-Feldern entgegenstehen, so dürfen sie nicht ausgefüllt werden. Diese Felder unterliegen Feldgruppenregeln und verfügen wie optionale Felder über den Attributwert  $\text{K}$ .

## Anzahl der Elemente von Listenfeldern

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

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

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

Die Tabelle `Feld` (Tabelle 23) erleichtert dem Softwarehersteller den Abgleich seines Datenmodells mit dem Datenmodell des IQTIG. Gleiche Informationen in der Menge aller Dokumentationsbögen müssen dadurch nicht redundant abgebildet werden.

Beispielsweise taucht das Feld `SCHULDOKU` in dem Modul `SEPE` auf. Um die mehrfache Pflege dieser Felder zu vermeiden, wird ein Feld mit dem Namen `SCHULDOKU` definiert und jeweils nur noch in der Tabelle `BogenFeld` referenziert.

Jedem Feld ist zwingend ein Basistyp zugeordnet (Abschnitt B 2.2.1). 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>27</sup> beschrieben.

Identifizierendes Attribut eines Felds ist allein sein technischer Name (Attribut `name`). Dies ist wichtig für die Eindeutigkeit von Feldnamen innerhalb eines Moduls: Felder mit unterschiedlichen Typen oder unterschiedlichen Schlüsseln müssen unterschiedliche Namen haben. Bei der Wahl des technischen Feldnamens sind beispielsweise folgende Punkte zu berücksichtigen:

- Der technische Name eines Feldes muss eindeutig sein
- Bei der Erstellung des technischen Feldnamens ist im Idealfall eine Zeichenlänge von circa 1 bis 15 Zeichen zu nutzen. Eine Zeichenlänge von 25 sollte im Idealfall nicht überschritten werden.

<sup>26</sup> In jedem Muss-Feld muss für jeden angelegten Teildatensatz einmal eine Angabe erfolgen.

<sup>27</sup> Das Attribut `bezeichnung` ist ein Standardtext für das gleichnamige Attribut der Tabelle `BogenFeld`. Im Eingabeformular wird die Bezeichnung aus der Tabelle `BogenFeld` angezeigt.

- Sonderzeichen wie z. B. Unterstriche sollten nicht verwendet werden.
- Der Feldname wird in Großbuchstaben geschrieben.
- Der technische Name ist so zu wählen, dass dieser den abzufragenden Sachverhalt zwar grob umfasst, jedoch nicht so konkret, dass eine modulübergreifende Verwendung des Feldes nicht mehr möglich ist.
- Nachträgliche Anpassungen der Bezeichnung des technischen Feldes bei Änderungen der Bogenfeldbezeichnung werden i.d.R. nicht vorgenommen.

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>28</sup> Die Anzahl der Elemente des von einem Feld abgeleiteten Bogenfelds wird über das Attribut `elemente` der Tabelle `BogenFeld` gesteuert.

Insbesondere für die Verwendung der richtigen Operatoren in den Plausibilitätsregeln und Feldgruppen ist die Listendefinition eines Felds wichtig.

Grundsätzlich gilt: Die Festlegung, ob ein Bogenfeld ein Skalar oder Listenfeld ist, wird durch die Tabelle `Feld` vorgegeben. Alle von einem Listenfeld abgeleiteten Bogenfelder sind automatisch auch Listenfelder. Die Größe der Liste wird individuell in der Tabelle `BogenFeld` konfiguriert.

Die Tabelle `Feld` bietet über die „Bogensicht“ hinausgehende Informationen.

Tabelle 23: Struktur der Tabelle `Feld`

Feldname	Datentyp	Bemerkung
<code>idFeld</code>	INTEGER	Primärschlüssel
<code>name</code>	KURZER TEXT	Technischer Name
<code>bezeichnung</code>	KURZER TEXT	(Erlaubte Zeichen: A–Z, 0–9, Ziffer nicht am Anfang) 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>einheit</code>	KURZER TEXT (50)	Einheit des Felds (z. B. mm, Stunden)
<code>formatAnweisung</code>	KURZER TEXT	Regulärer Ausdruck für die Formatprüfung (z. B. [0–9] { 9 })
<code>Funktion</code>	KURZER TEXT	Formel zur Generierung des Feldinhaltes, z. B. durch Aufruf einer Syntaxfunktion
<code>nachKommaLaenge</code>	INTEGER	Anzahl der Nachkommastellen in der Feldeingabemaske (muss kleiner als <code>laenge</code> sein)

<sup>28</sup> Man beachte die Besonderheiten der Listenfelder beim Datenexport und in der Syntax der Plausibilitätsregeln.

Feldname	Datentyp	Bemerkung
min	DOUBLE	Harte Untergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
max	DOUBLE	Harte Obergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
minWeich	DOUBLE	Weiche Untergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
maxWeich	DOUBLE	Weiche Obergrenze des Wertebereichs eines numerischen Datenfelds (modulübergreifend). Die Definition ist optional.
istListe	BOOLEAN	Wenn istListe = WAHR, so sind die vom betreffenden Feld abgeleiteten Bogenfelder Listenfelder.
fkBasisTyp	INTEGER	Fremdschlüssel zur Tabelle Basistypen
fkSchluessel	INTEGER	Fremdschlüssel zur Tabelle Schlüsseltypen
fkKombiFeld	INTEGER	Optionaler Fremdschlüssel auf ein anderes Feld, welches Kombinationsfelder kennzeichnet
strukturParameter	BOOLEAN	Markierung von Strukturparametern, die durch die Software automatisch aus dem KIS in den Dokumentationsbogen übernommen werden können

### Kombinationsfelder

Für manche Bogenfelder ist zwingend vorgeschrieben, dass sie innerhalb eines Moduls in Kombination mit einem anderen Bogenfeld existieren. Die Definition von Kombinationsfeldern geschieht mithilfe des optionalen Fremdschlüssels fkKombiFeld in der Tabelle Feld.

### Basistypen

Das Hauptmerkmal eines Basistyps ist der technische Typ eines Eingabefelds (z. B. Zeichenkette, numerischer Typ, Datum usw.). Wichtiges Charakteristikum ist die Beschreibung des Eingabeformats. Die Basistypen sind Voraussetzung für die Beschreibung einer formalen Regelsyntax (Abschnitt B 2.3.2).

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

Tabelle 24: Struktur der Tabelle BasisTyp

Feldname	Datentyp	Bemerkung
idBasisTyp	INTEGER	Primärschlüssel
name	KURZER TEXT	Technischer Name (muss eindeutig sein)

standardtyp	KURZER TEXT	Entsprechender Standarddatentyp
bezeichnung	KURZER TEXT	Beschreibender Text
format	KURZER TEXT	Formatdefinition, z. B. TT.MM.JJJJ beim Basistyp Datum
formatRegExp	KURZER 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



### Hinweis

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

## Schlüssel

Identifizierendes Merkmal eines Schlüssels (Kodesystem) ist sein technischer Name. Die meisten Schlüsselkodes sind in der Tabelle SchluesselWert definiert.

Tabelle 25: Struktur der Tabelle Schluessel

Feldname	Datentyp	Bemerkung
idSchluessel	INTEGER	Primärschlüssel
name	KURZER TEXT	Technischer Name (muss eindeutig sein)
bezeichnung	KURZER TEXT	Beschreibender Text
extern	BOOLEAN	Zeigt an, ob der Schlüssel in der Tabelle Schluessel (= FALSCH) oder in einer externen Tabelle gespeichert (= WAHR) ist.
externVerweis	KURZER TEXT	Verweis auf die Quelle des externen Schlüssels
zahl	BOOLEAN	Wenn WAHR, sind die Werte im Attribut code der zu gehörigen Schlüsselwerte als ganze Zahl kodiert, ansonsten als Zeichenkette.

Feldname	Datentyp	Bemerkung
sortierNrVerwendet	BOOLEAN	Flag, das anzeigt, ob für die Reihenfolge das Attribut sortierNr der Tabelle SchluesselWert herangezogen wird.
fkMutterSchluessel	INTEGER	Referenz auf einen übergeordneten Schlüssel.

Schlüsselkodes können auf zwei Arten interpretiert werden: Wenn das Attribut `zahl` gesetzt ist, so werden die Kodes als ganze Zahl gedeutet, ansonsten werden sie als Zeichenketten interpretiert. In der Syntax der Plausibilitätsregeln werden die letztgenannten Kodes in einfache Hochkommata gesetzt (Abschnitt B 2.3.2).

### Beispiel:

Attribut **zahl** bei Schlüsselfeldern

- Felder des Basistyps NUMSCHLUESSEL haben das Attribut `zahl` = WAHR.
- Felder des Basistyps SCHLUESSEL haben das Attribut `zahl` = FALSCH. Es handelt sich um alphanumerische Schlüssel, die Buchstaben, Ziffern oder Sonderzeichen verwenden (z. B. ypN0). Hierbei kann es sich auch um Werte handeln, die lediglich Ziffern verwenden, aber mit führender Null beginnen (z. B. 01).

### Externe Schlüsselkataloge

Externe Schlüsselkataloge sind über das Attribut `extern` deklariert. Hinweise zu den Bezugsquellen sind in der Spalte `externVerweis` zu finden (z. B. [www.bfarm.de](http://www.bfarm.de)). Diese externen Schlüsselkataloge werden nicht vom IQTIG bereitgestellt und somit auch nicht verantwortet.

Hinweise zu den Bezugsquellen sind in der Spalte `externVerweis` zu finden. Ein Verweis auf eine Bezugsquelle kann unabhängig vom Attribut `extern` angegeben werden.

Die Schlüsselkodes, die dem IQTIG zum Zeitpunkt der Publikation der QS-Spezifikation bekannt sind, sind in der Tabelle `SchluesselWert` enthalten. Spätere Schlüsseländerungen bzw. Fortschreibungen müssen (z. B. vom Softwareanbieter und von der datenentgegennehmenden Stelle) selbstständig und zeitnah über die jeweilige Bezugsquelle bezogen werden.

Die Schlüsselkodes sind in der Tabelle `SchluesselWert` enthalten. Spätere Schlüsseländerungen bzw. -fortschreibungen werden vom IQTIG zeitnah übernommen.

### Schlüsselwerte

Tabelle 26 gibt einen Überblick über die Datenbanktabelle `SchluesselWert`, in der die Kodes und Bezeichnungen der Schlüssel hinterlegt sind. Identifizierendes Merkmal ist hier eine Kombination der Spalten `fkSchluessel` und `code`. Das bedeutet, dass jeder Schlüsselkode innerhalb eines Schlüssels nur einmal vorkommen darf.

Tabelle 26: Struktur der Tabelle SchluesselWert

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

Das Attribut `code` der Tabelle `SchluesselWert` ist ein Textfeld, das in Abhängigkeit vom Wert des Attributes `zahl` im zugeordneten Schlüssel entweder numerisch oder nichtnumerisch interpretiert wird. Wenn in einer Plausibilitätsregel (Abschnitt B 2.3.2 und B 2.3.7) Felder mit numerischen Schlüsseln (Basistyp `NUMSCHLUESSEL`) vorkommen, so werden bei der Evaluierung der Regel die Schlüsselkodes wie ganze Zahlen behandelt.

### Sortierung der Kodes

- Für die Kodes (Attribut `SchluesselWert.code`) eines Schlüssels ist eine Sortierung definiert. Die Art der Sortierung wird über die Attribute `zahl` und `sortierNrVerwendet` der Tabelle `Schluessel` festgelegt.
- Numerische Sortierung: Wenn `sortierNrVerwendet = FALSCH` und `zahl = WAHR`, so sind die Kodes nach der Spalte `code` der Tabelle `Schluessel` numerisch zu sortieren.
- Alphanumerische Sortierung: Wenn `sortierNrVerwendet = FALSCH` und `zahl = FALSCH`, so sind die Kodes nach der Spalte `code` der Tabelle `Schluessel` alphanumerisch zu sortieren.
- Spezielle Sortierung: Wenn `sortierNrVerwendet = WAHR`, so sind die Kodes nach den Werten in der Spalte `sortierNr` der Tabelle `Schluessel` numerisch zu sortieren.

### Suchfunktion bei Schlüsseln mit einer großen Anzahl von Kodes

Bei Schlüsseln mit einer großen Anzahl von Kodes soll eine anwenderfreundliche Möglichkeit zur Auswahl der passenden Kodes bereitgestellt werden. Die Umsetzung als Auswahlliste (z. B. Combobox) führt zu erhöhtem Dokumentationsaufwand, falls der Anwender über Pfeiltasten oder Schieberegler zum passenden Kode navigieren muss. Ergänzend soll daher eine Suchfunktion realisiert werden, die eine Suche über die Attribute `SchluesselWert.code` oder `SchluesselWert.bezeichnung` ermöglicht. Die zu realisierenden Anwendungsfälle werden in den folgenden Beispielen erläutert.

## 2.2.4 Überschriften

Die Überschriften der Dokumentationsbögen in der Spezifikation sind in der Tabelle `Abschnitt` zu finden.

Tabelle 27: Struktur der Tabelle `Abschnitt`

Feldname	Datentyp	Bemerkung
<code>idAbschnitt</code>	INTEGER	Primärschlüssel
<code>bezeichnung</code>	KURZER TEXT	Text der Überschrift
<code>ebene</code>	INTEGER	Zeigt die Hierarchie der Überschriften an
<code>fkStartBogenFeld</code>	INTEGER	Fremdschlüssel auf das erste zur Überschrift gehörende Bogenfeld
<code>fkEndeBogenFeld</code>	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.



### Achtung

Die in der Spezifikationsdatenbank hinterlegten Überschriften sind in die Eingabemasken (z. B. der QS-Dokumentationssoftware) zu integrieren. Viele Datenfelder sind für den Anwender erst im Kontext der Überschriften verständlich.

## 2.2.5 Ausfüllhinweise

Die Ausfüllhinweise zu den Datenfeldern sind in einem separaten ZIP-Archiv enthalten, das nach dem Benennungsschema für Spezifikationskomponenten bezeichnet wird (`TechDok_Benennungsschema`). Jeder Ausfüllhinweis ist ein HTML-Dokument.

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

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

Wenn der Eintrag in `ahinweis` leer ist, so existiert für das betreffende Bogenfeld kein Ausfüllhinweis. Das Attribut `fkA hinweisTyp` lässt die Differenzierung drei verschiedener Arten von Ausfüllhinweisen zu:

Tabelle 28: Arten von Hinwestypen

fkA hinweisTyp	Beschreibung	Beispiel
<code>Feldbezogen</code>	Der Ausfüllhinweis bezieht sich auf den entsprechenden Datensatz in der Tabelle <code>Feld</code> . Der Ausfüllhinweis ist modulunabhängig.	<code>IKNRKH.htm</code> Der Ausfüllhinweis bezieht sich auf verschiedene Module.

Modulspezifisch	Soll sich ein Ausfüllhinweis nur auf ein bestimmtes Modul beziehen, kann der Attributwert modulspezifisch ausgewählt werden.	
Speziell	Soll es für verschiedene Datenfelder der Tabelle Feld einen gemeinsamen Ausfüllhinweis geben, kann dieser als speziell deklariert werden. Der Attributwert ahinweis definiert den Namen des Ausfüllhinweises.	

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.

## 2.3 Plausibilitätsprüfungen

Es wird zwischen zwei Arten von Plausibilitätsprüfungen unterschieden, die in Tabelle `RegelTyp` definiert sind:

- harte Prüfungen
- weiche Prüfungen in der QS-Dokumentationssoftware

Tabelle 29: Tabelle `RegelTyp`

<code>idRegelTyp</code>	<code>bezeichnung</code>
H	hart
W	weich

Weiterhin wird zwischen sogenannten **Einzelregeln** (Abschnitt B 2.3.5) und **Feldgruppenregeln** (Abschnitt B 2.3.7) unterschieden.

### 2.3.1 Die Regeltablelle

Die Bedingungen für unplausible Angaben<sup>29</sup> sind in der Tabelle `Regeln` abgelegt. Die hier beschriebenen Prüfungen sind in der Spezifikationsdatenbank für QS-Dokumentation hinterlegt. Die Syntax ist in Abschnitt B 2.3.2 beschrieben. Die Bedingungen sind möglichst kurzgefasst (Vermeidung von durch ODER verknüpften Teilbedingungen). Jede Bedingung kommt nur einmal innerhalb eines Moduls vor.

---

<sup>29</sup> Eine Plausibilitätsregel müsste eigentlich „Unplausibilitätsregel“ heißen, weil sie unplausible Zustände beschreibt, die zu Fehlermeldungen führen.

Tabelle 30: Struktur der Tabelle Regeln

Feldname	Datentyp	Bemerkung
idRegeln	INTEGER	Primärschlüssel
fkModul	INTEGER	Fremdschlüssel zur Tabelle Modul
bedingung	LANGER TEXT	Entsprechend der Syntax definierte Regeln
meldung	LANGER TEXT	Fehlermeldung: Diese Texte sind bei Regeln mit Bezug zu Feldgruppen automatisch generiert.
alternativMeldung	LANGER TEXT	Alternative Fehlermeldung: Wenn hier ein Text vorhanden ist, so ist dieser anstelle des Textes in der Spalte meldung zu verwenden.
fkMehrfachRegel	INTEGER	Fremdschlüssel zur Tabelle mit mehrfach vorkommenden Regeln, die mithilfe von Ersatzbedingungen nach dem Export gültig sind. Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
fkFeldGruppe	INTEGER	Optionaler Fremdschlüssel zur Tabelle FeldGruppe: Indikator dafür, ob eine Regel aus einer Feldgruppe generiert wurde.
fkRegelTyp	KURZER TEXT (1)	Fremdschlüssel zur Tabelle RegelTyp: Die Regeltypen sind die in Abschnitt A 2.3 beschriebenen Arten der Plausibilitätsprüfungen: H oder W
gueltigNachExport	BOOLEAN	Regeln, die den Wert FALSCH haben, können von Datenannahmestellen nicht evaluiert werden. Stattdessen werden die referenzierten Ersatzbedingungen der Tabelle MehrfachRegel evaluiert (falls definiert).

### Regelfelder (Bogenfelder einer Regel)

Die Tabelle RegelFelder (Tabelle 31) ist eine Verknüpfungstabelle zwischen den Tabellen Regeln und BogenFeld. Durch gezielte Abfragen erhält man unter Verwendung dieser Tabelle einen Überblick über Folgendes:

- Bogenfelder, die in einer Regel verwendet werden.
- Regeln, die sich auf ein Bogenfeld beziehen.

Tabelle 31: Struktur der Tabelle RegelFelder

Feldname	Datentyp	Bemerkung
fkBogenFeld	INTEGER	Fremdschlüssel zu den Tabellen Feld und Regeln, bilden zusammen den Primärschlüssel
fkRegeln	INTEGER	

## Mehrfachregeln (Ersatzregeln zur Prüfung nach dem Export)

Wenn in einer Regel von der Pseudonymisierung betroffene Datenfelder benutzt werden, so kann diese von Datenannahmestellen nicht evaluiert werden. Stattdessen wird für solche Regeln in der Tabelle `MehrfachRegel` eine Ersatzbedingung definiert, deren Referenzierung in der Tabelle `Regeln` definiert ist. Die Ersatzbedingung ist von den Datenannahmestellen zu evaluieren.

Weitere Erläuterungen zu Mehrfachregeln sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

## Weitere Regeln

Weitere feldübergreifende Regeln sind die in Abschnitt B 2.2.2 beschriebenen Existenzbedingungen für das Anlegen von abhängigen Teildatensätzen (Attribut `existenzBedingung` in Tabelle `Bogen`).

### 2.3.2 Regelsyntax

Bedingungen sind in den Tabellen `Regeln`, `MehrfachRegel` und `Bogen` definiert. Die den Bedingungen zugrundeliegende Regelsyntax wird in diesem Abschnitt beschrieben. Jede Regel ist ein logischer Ausdruck, dessen Ergebnis WAHR oder FALSCH lautet. Jede Regel bezieht sich auf einen eingegebenen Datensatz eines Moduls, dessen Daten in Variablen gespeichert sind.

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. Die Regelsyntax ist als Pseudocode zu verstehen.

## Typen

Die möglichen Typen der Datenfelder sind in Tabelle 32 aufgelistet.

Tabelle 32: Basistypen der Datenfelder in den Plausibilitätsregeln

Basistyp	Bezeichnung	Beispiele (Literale)
BOOL	Boolesche Variable	WAHR, FALSCH
TEXT	Zeichenkette (String)	"Spezifikation"
GANZEZAHL <sup>30</sup>	... -2, -1, 0, 1, 2, 3, ...	1
ZAHL	Zahl (mit oder ohne Nachkommastellen)	25,4 oder -100,8
DATUM	Zehnstelliges Datum	'01.01.2026'
MONDATUM	Monatsdatum	'04.2026'
QUARTDATUM	Quartalsdatum	'3/2026'
JAHRDATUM	Jahresdatum	2026

<sup>30</sup> Beim Typ GANZEZAHL sind auch negative ganze Zahlen erlaubt.

<b>Basistyp</b>	<b>Bezeichnung</b>	<b>Beispiele (Literale)</b>
NUMSCHLUESSEL	Numerisch kodierter Schlüssel (wie GANZEZAHL)	1
SCHLUESSEL	Alphanumerischer Schlüssel	'19.1', '07'
UHRZEIT	Uhrzeit	'10:15'

In der Spezifikation für die QS-Dokumentation wird zwischen NUMSCHLUESSEL und SCHLUESSEL unterschieden:

- Schlüsselwerte verfügen über den Datentyp NUMSCHLUESSEL, wenn es sich bei den Kodes um ganze Zahlen handelt. Da dies ein numerischer Schlüssel ist, darf er nicht in Hochkommata gesetzt werden.
- Schlüsselwerte, die alphanumerische Kodes beinhalten, haben den Basistyp SCHLUESSEL. Die OPS-Schlüssel (z. B. '5-282.0') und die ICD-10-GM-Schlüssel verfügen über diesen Daten- typ, der in Hochkommata geschrieben wird.



### Achtung

Datumsangaben (Datum, Monats-, Quartalsdatum) müssen in Hochkommata gesetzt werden. Eine Ausnahme ist das Jahresdatum (JAHRDATUM), da es sich hierbei um eine ganze Zahl handelt.

---

## 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 kein reserviertes Wort sein (z. B. LEER).



### Achtung

In einer Regel dürfen nur die Feldnamen der im betreffenden Modul definierten Bogenfelder<sup>31</sup> enthalten sein. Bei der Evaluierung von Regeln werden die aktuellen Werte der referenzierten Bogenfelder eingesetzt. Kann-Bogenfelder können auch unausgefüllt sein, also den Wert LEER haben.

---

## Listenfelder

Ein Bogenfeld wird dann als Liste interpretiert, wenn im referenzierten Feld (Tabelle Feld) der Wert des Attributs Feld.istListe = WAHR ist. Andernfalls 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.

---

<sup>31</sup> Bei den Ersatzregeln in Tabelle MehrfachRegel sind stattdessen die Exportfelder des Moduls erlaubt.

## Literele

Alphanumerische Literale (z. B. SCHLUESSEL) werden von einfachen Hochkommata eingeschlossen, während Zeichenketten vom Datentyp TEXT in Anführungszeichen gesetzt werden müssen.<sup>32</sup>

Dies gilt nicht für numerische Literale (GANZEZAHL, ZAHL, NUMSCHLUESSEL und JAHRDATUM) und Literale des Datentyps BOOL (Wahrheitswerte).

## Listen von Literalen

Literele können sowohl als Skalare als auch als Listen angesprochen werden. Der Separator einer Liste von Literalen ist das Semikolon. Um zu prüfen, ob alle Listenfelder ausgefüllt sind, wird die Liste über den Wert LEER angesprochen.

Außerdem gibt es Teildatensatz-Listenfelder, die im Abschnitt B 2.3.6 beschrieben werden.

## Operatoren

Tabelle 33 gibt einen Überblick über die in der Syntax zulässigen Operatoren. Der aktuelle Überblick über alle zulässigen Operationen (inkl. Operanden) ist in Tabelle SyntaxOperator in der QSDOK-Datenbank zu finden.

In Tabelle 33 hat jeder einzelne Operator eine Präzedenzstufe (höchste Präzedenzstufe ist 0). Operatoren, die die gleiche Stufe haben, werden nach den Regeln der Assoziativität aufgelöst.

Tabelle 33: Präzedenz und Assoziativität der Operatoren

Präzedenz	Assoziativität	Operator	Erläuterung
0	links	IN	Operator zum Vergleich einer Variablen mit einer Liste (z. B. ein Datenfeld mit Schlüsselwerten). Die Variable und die Feldelemente müssen gleichen Typs sein.
	links	NICHTIN	
	links	EINSIN	Operator zum Vergleich einer Liste mit einer anderen Liste oder einem Listenelement (z. B. ein Listenfeld mit einem Schlüsselwert oder ein Listenfeld mit einer OPS-Liste). Die Listenelemente müssen gleichen Typs sein.
	links	JEDESIN	
	links	EINSNICHTIN	
	links	KEINSIN	
1	links	*	Operator für die Multiplikation „mal“
	links	/	Operator für die Division „geteilt“
2	links	+	Operator für die Addition „plus“
	links	-	Operator für die Subtraktion „minus“
3	links	<	Vergleichsoperator „kleiner“

<sup>32</sup> Beim Export entfallen die begrenzenden Zeichen.

	links	>	Vergleichsoperator „größer“
	links	<=	Vergleichsoperator „kleiner gleich“
	links	>=	Vergleichsoperator „größer gleich“
4	links	=	Vergleichsoperator „gleich“
	links	<>	Vergleichsoperator „ungleich“
5	rechts	NICHT	Logischer Operator: „NICHT“
6	links	UND	Logischer Operator: „UND“
7	links	ODER	Logischer Operator: „ODER“

### Prüfung auf LEER mit Vergleichsoperatoren

Die Prüfung auf LEER von in Regeln verwendeten Kann-Feldern, welche an anderer Stelle in der Regel mit einem anderen Operator als <> oder = geprüft werden sollen, findet auf der linken Seite einer ODER-Verknüpfung statt. Hintergrund dieser Syntaxregel ist, dass die Vermeidung von Laufzeitfehlern bei der Evaluation ermöglicht werden soll.

#### Beispiel:

FELD = LEER ODER FELD OPERATOR OPERAND

Beispielsweise kann bei leeren Feldwerten und der vorgegebenen Linksassoziativität des ODER-Operators die Evaluation bei leerem Feldwert vor der Evaluation des rechtsstehenden Ausdrucks mit der Rückgabe von WAHR abgebrochen werden. Ein Laufzeitfehler, der sich z. B. bei einem Vergleich von LEER < WERT ergeben würde, kann so nicht entstehen.

### Operatoren zum Vergleich einer Variablen mit einer Liste

Folgende Operatoren erfordern entweder nur rechts oder links und rechts Listenfelder:

- nur rechts: IN, NICHTIN
- links und rechts: EINSIN, KEINSIN, JEDESIN, EINSNICKTIN

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).
- EINSNICKTIN: Der Ausdruck ist dann wahr, wenn mindestens ein Element der linken Liste nicht in der rechten Liste enthalten ist (nichtleere Differenz).

**Beispiel:**

- 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.

---

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

**Beispiel:**

NICHT FELD JEDESIN (LEER)

Diese Bedingung erfordert, dass zumindest ein Listenelement ausgefüllt ist. Beispielsweise erfüllt FELD := (WERT;LEER;LEER;LEER) die Bedingung.

Gleichwertig ist die Regel:

FELD EINSNICKTIN (LEER)

---

Folgende Operatoren sind komplementär:

IN und NICHTIN

EINSIN und KEINSIN

JEDESIN und EINSNICKTIN

Folgende Ausdrücke sind gleich:

A EINSNICKTIN B

NICHT(A JEDESIN B)

### 2.3.3 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. Funktionen können nicht nur in Regeln, sondern auch zur Berechnung von Exportfeldern genutzt werden (Abschnitt B 2.5.3).

Der aktuelle Stand der in der Syntax verwendeten Funktionen ist in der Tabelle [SyntaxFunktion](#) der Spezifikation zu finden. In der QSDOK-Spezifikationsdatenbank der einrichtungsbezogenen QS-Dokumentation sind derzeit keine Funktionen hinterlegt.

### 2.3.4 Syntaxvariablen

Der Eingangsdatensatz bildet die medizinische Routinedokumentation ab, die in jedem Krankenhausinformationssystem (KIS) enthalten ist. Syntaxvariablen in der QS-Dokumentation dienen der technischen Darstellung der automatischen Generierung von Angaben aus dem Eingangsdatensatz. Diese sind in Tabelle **SyntaxVariable** hinterlegt. Weiterführende Erläuterungen zu Syntaxvariablen sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

### 2.3.5 Einzelregeln

Sogenannte Einzelregeln können sich als feldbezogene Prüfungen auf ein einziges Datenfeld oder als feldübergreifende Prüfungen auf mehrere Datenfelder beziehen. Einzelregeln sind von den in Abschnitt B 2.3.7 beschriebenen Feldgruppen zu unterscheiden.

**Feldbezogene Prüfungen** – beispielsweise Wertebereichsüberprüfungen – sind in der formalen Regelsyntax in Tabelle **Regeln** formuliert.

Unter feldbezogenen Prüfungen sind aber auch die in Abschnitt B 2.3.8 beschriebenen Prüfungen des Formates, der Feldlänge, der Wertebereiche, Prüfungen von Schlüsselkodes und von Muss-Feldern zu verstehen. Für diese Prüfungen gibt es keine formale Regelsyntax in Tabelle **Regeln**.

#### Feldübergreifende 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>33</sup>.

Feldübergreifende Regeln können auch teildatensatzübergreifende Regeln sein, wenn die Datenfelder der Regel aus mehreren Teildatensätzen eines Moduls stammen (Abschnitt B 2.3.6).

### 2.3.6 Teildatensatzübergreifende Regeln

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

Es gibt zwei Arten von teildatensatzübergreifenden Regeln:

1. Die Felder sind in verschiedenen Teildatensätzen eines Moduls definiert.
2. 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).

---

<sup>33</sup> Eine Plausibilitätsregel müsste eigentlich „Unplausibilitätsregel“ heißen, weil sie unplausible Zustände beschreibt, die zu Fehlermeldungen führen.

## Regeln mit Teildatensatz-Listenfeldern

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

### 2.3.7 Feldgruppenregeln

Logische Abhängigkeiten von Bogenfeldern werden über Feldgruppen dargestellt. Die Plausibilitätsregeln, die einen Bezug zu einer Feldgruppe aufweisen (Tabelle Regeln), werden anhand der Feldgruppdefinition (Tabelle FeldgruppeFelder) automatisch generiert. Die Menge der abgeleiteten Einzelregeln wird in diesem Abschnitt erläutert.

Die möglichen Antworten<sup>34</sup> eines jeden Datenfeldes werden in zwei Gruppen aufgeteilt. Die erste Gruppe ist die Menge der positiven, die zweite Gruppe die Menge der negativen Antworten.<sup>35</sup>

Typische positive Antworten sind beispielsweise:

Feld <> LEER oder Feld IN (2; 3)

Die komplementären negativen Antworten würden entsprechend wie folgt lauten:

Feld = LEER oder Feld NICHTIN (2; 3)

Eine Feldgruppe kann ein Filterfeld haben. Wenn die Antwort dieses Filterfeldes negativ ausfällt (Bspw. Bedingung: Feld = 3; Antwort: Feld <> 3), so darf keines der abhängigen Felder positiv beantwortet werden.

Tabelle 34 gibt einen Überblick über die Typen von Feldgruppen. Der aktuelle Stand findet sich in der Tabelle FeldGruppenTyp der Spezifikation.

Tabelle 34: Typen von Feldgruppen

Name	Bemerkung
<b>mit Filterfeld</b>	
EF_FILTER	Einfachauswahl, genau ein abhängiges Feld muss positiv beantwortet sein
EF_OPTIONAL_FILTER	Einfachauswahl, genau ein abhängiges Feld kann positiv beantwortet sein
MF_OPTIONAL_FILTER	Mehrfachauswahl, alle abhängigen Felder können positiv beantwortet sein
MF_MINDESTENS1_FILTER	Mehrfachauswahl, mindestens ein abhängiges Feld muss positiv beantwortet sein

<sup>34</sup> Die Antworten eines Datenfeldes umfassen hier neben möglichen Werten (z. B. Schlüsselwerten) oder Wertemengen auch die Kategorie „nicht ausgefüllt“ (LEER).

<sup>35</sup> Die negativen Antworten sind abhängig von der definierten Bedingung eines Feldes in der entsprechenden Feldgruppe.

Name	Bemerkung
MF_ALLES_FILTER	Mehrfachauswahl, alle abhängigen Felder müssen positiv beantwortet sein
ohne Filterfeld	
EF	Einfachauswahl, genau ein Feld muss positiv beantwortet sein
MF_OPTIONAL	Mehrfachauswahl, alle Felder können positiv beantwortet sein
MF_MINDESTENS1	Mehrfachauswahl, mindestens ein Feld muss positiv beantwortet sein
UND	Einfache Regel mit Und-Verknüpfungen

In der Tabelle `BogenFeld` sind abhängige Datenfelder einer Feldgruppe immer als Kann-Felder definiert. Nach Abhängigkeit der Feldgruppenlogik können/müssen diese Felder leer bleiben oder zwingend ausgefüllt werden. Im letztgenannten Fall können die Datenfelder auch als bedingte Muss-Felder bezeichnet werden.

Die Muss- oder Kann-Definition der Datenfelder (Bogen- und Ersatzfelder) im Exportformat unterliegt ebenfalls der Feldgruppenlogik. Ist die Berechnung eines Ersatzfeldes von bedingten Datenfeldern abhängig, so gilt die Feldgruppenlogik auch für diese Ersatzfelder. Wenn die bedingten Datenfelder zwingend ausgefüllt werden müssen, so muss auch das Ersatzfeld zwingend berechnet bzw. exportiert werden.

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

Die Feldgruppen sind in den Tabellen `Feldgruppe` und `FeldgruppeFelder` definiert. In der Tabelle `FeldGruppe` (Tabelle 35) sind Name, Typ und die Zuordnung zu einem Modul definiert. Die Verknüpfungstabelle `FeldgruppeFelder` (Tabelle 36) definiert die abhängigen Bogenfelder. Zusätzlich wird hier festgelegt, welche Bogenfelder der Feldgruppe als Filterfeld dienen.

Tabelle 35: Struktur der Tabelle `FeldGruppe`

Feldname	Datentyp	Bemerkung
idFeldGruppe	INTEGER	Primärschlüssel
name	KURZER TEXT (64)	Technischer Name der Feldgruppe
fkModul	INTEGER	Obligatorischer Fremdschlüssel zu einem Modul
fkFeldgruppenTyp	INTEGER	Obligatorischer Fremdschlüssel zu einem Feldgruppentyp
hinweis	KURZER 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.

Feldname	Datentyp	Bemerkung
		Der Hinweistext kann bei der Erstellung der Eingabemasken verwendet werden.
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 Abschnitt A 2.3 beschriebenen Arten der Plausibilitätsprüfungen: H oder W Die generierten Einzelregeln der Feldgruppe haben den gleichen Regeltyp.
nurPositiv	BOOLEAN	Nur bei Filter-Feldgruppen wirksam: Wenn WAHR, dann umfasst die Feldgruppe nur diejenigen Regeln, welche sich auf die positive Filterbedingung beziehen.
grauWennNegativ	BOOLEAN	Definiert eine Layout-Feldgruppe, wenn WAHR (siehe unten S. 94, Layout-Feldgruppen)

Tabelle 36: Struktur der Tabelle FeldgruppeFelder

Feldname	Datentyp	Bemerkung
idFeldgruppeFelder	INTEGER	Primärschlüssel
fkFeldGruppe	INTEGER	Obligatorischer Fremdschlüssel zur Feldgruppe
fkBogenFeld	INTEGER	Obligatorischer Fremdschlüssel zum Bogenfeld
bedingung	KURZER TEXT	Positive Bedingung für das jeweilige Bogenfeld
istFilter	BOOL	Legt fest, ob das jeweilige Bogenfeld ein Filterfeld ist
bezeichnungSchluesselListe	KURZER TEXT	Abkürzende Bezeichnung für eine Schlüsselliste in der Bedingung, wird beim Generieren von Fehlermeldungen verwendet.
tdsFilter	BOOLEAN	Das Bogenfeld wird in Regeln als TDS-Listenfeld (Abschnitt 2.3.6) verwendet (Voranstellen des @-Zeichens vor Feldnamen).

## 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 33 verwendet werden. Die auf der rechten Seite erlaubten Operanden sind nachfolgend aufgelistet:

- Literale (Tabelle 32)
- LEER
- Kodelisten, in denen auch die Kodes eines Schlüssels referenziert werden können;  
Beispiel: `(1;2;3)`
- ICD-Listen oder OPS-Listen



### Hinweis

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

## Formale Definition von Feldgruppen

A sei ein Bogenfeld in 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 kann ggf. ein Filterfeld haben, das mit F bezeichnet wird. Eine Feldgruppe lässt sich dann in folgender Tabelle darstellen:

Tabelle 37: Formale Definition einer Feldgruppe

Feld	Positive Bedingung	Negative Bedingung	Bemerkung
F	$p(F)$	$n(F)$	falls Feldgruppentyp mit Filter
A1	$p(A1)$	$n(A1)$	
A2	$p(A2)$	$n(A2)$	
A3	$p(A3)$	$n(A3)$	
...			
An	$p(An)$	$n(An)$	

Eine Feldgruppe besteht insgesamt aus  $n$  abhängigen Bogenfeldern:

$A_1, A_2, \dots, A_n$

In Abhängigkeit von den Feldgruppentypen werden unterschiedliche Einzelregeln generiert.

## Feldgruppen mit Filter

- Regeln der Feldgruppe „Optionale Mehrfachauswahl mit Filterfeld“ (MF\_OPTIONAL\_FILTER)

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

Insgesamt sind  $n$  Einzelregeln mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Obligatorische Mehrfachauswahl mit Filterfeld“ (MF\_MINDESTENS1\_FILTER)

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

$p(F) \text{ UND } n(A1) \text{ UND } n(A2) \text{ UND } ... \text{ UND } n(An)$

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)

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

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

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

- Regeln der Feldgruppe „Einfachauswahl mit Filter“ (EF\_FILTER)

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

$p(F) \text{ UND } n(A1) \text{ UND } n(A2) \text{ UND } ... \text{ UND } n(An)$

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

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

- Regeln der Feldgruppe „Optionale Einfachauswahl mit Filter“ (EF\_OPTIONAL\_FILTER)

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

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

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

## Feldgruppen mit Filter: Attribut nurPositiv

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:

$p(F) \text{ UND } n(A1) \text{ UND } n(A2) \text{ UND } ... \text{ UND } n(An)$

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

### Feldgruppen ohne Filter

- Regeln der Feldgruppe „Einfachauswahl“ (EF)

$n(A1) \text{ UND } n(A2) \text{ UND } \dots \text{ UND } n(An)$

$p(Aj) \text{ UND } p(Ai)$  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)

$n(A1) \text{ UND } n(A2) \text{ UND } \dots \text{ UND } n(An)$

Insgesamt ist eine Einzelregel mit der Feldgruppe verknüpft.

- Regeln der Feldgruppe „Und-Regel“ (UND)

$p(A1) \text{ UND } p(A2) \text{ UND } \dots \text{ UND } p(An)$

Insgesamt ist eine Einzelregel mit der Feldgruppe verknüpft.

### Feldgruppen mit mehreren Filterfeldern

Es 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 Abschnitt B 2.3.7 definierten Filterbedingungen  $p(F)$  und  $n(F)$  bei den Einzelregeln.

Die Filterfelder können entweder über eine ODER-Verknüpfung oder eine UND-Verknüpfung miteinander 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.

### Layout-Feldgruppen

Feldgruppen, bei denen das Attribut grauWennNegativ in der Datenbanktabelle FeldGruppe WAHR ist, werden nachfolgend als Layout-Feldgruppen bezeichnet. Der Attributname grauWennNegativ wurde gewählt, weil die abhängigen Felder der Layout-Feldgruppen auf den generierten Dokumentationsbögen eingegraut sind.

Layout-Feldgruppen haben folgende Eigenschaften:

- Sie haben mindestens ein Filterfeld.

- Jedes abhängige Feld hat die Bedingung <> LEER oder EINSNICKTIN (LEER) (Attribut bedingung in Tabelle FeldGruppeFelder).
- Das Attribut nurPositiv hat den Wert FALSCH.

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

### **2.3.8 Prüfung von Feldeigenschaften**

Die in diesem Abschnitt behandelten feldbezogenen Prüfungen ergeben sich direkt aus den Tabellen Feld (bzw. ErsatzFeld oder ZusatzFeld) und BogenFeld (bzw. ExportFormat) und werden vor Evaluation der in Abschnitt B 2.3.4 beschriebenen feldübergreifenden Regeln durchgeführt.

Die hier beschriebenen Prüfungen sind nur in Form von Feldeigenschaften – nicht aber in Regelsyntax – in der Datenbank für QS-Dokumentation hinterlegt.

#### **Überprüfung des Formats**

Die Formatprüfung bezieht sich auf den Exportdatensatz (Abschnitt B 2.5.3): Die QS-Dokumentations- bzw. Exportsoftware muss Daten im korrekten Format generieren, was durch die datenentgegennehmenden Stellen geprüft wird. Für Exportfelder (Tabelle ExportFormat), die einen Bezug zu einem Ersatzfeld (Tabelle ErsatzFeld) bzw. zu einem Zusatzfeld (Tabelle ZusatzFeld) haben, sind die standardisierten Fehlertexte anzupassen.

Die Prüfung bezieht sich insbesondere darauf, ob der Feldinhalt dem in der Spezifikation definierten Basistyp entspricht. Beispielsweise sind Buchstaben beim Basistyp GANZEZAHL nicht erlaubt.

#### **Standardisierter Fehlertext für Formatfehler eines Ersatzfeldes**

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<Ersatzfeld.name> '<ErsatzFeld.bezeichnung>' ist kein gültiger <Basistyp.name> Wert (<Basistyp.bezeichnung> <Basistyp.format>).

#### **Standardisierter Fehlertext für Formatfehler eines Bogenfeldes**

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

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

Das Zusatzfeld <Modul.name>:<Bogen.name>:<ZusatzFeld.name> '<ZusatzFeld.bezeichnung>' ist kein gültiger <Basistyp.name> Wert (<Basistyp.bezeichnung> <Basistyp.format>).

## Überprüfung der Feldlänge

Die Feldlängenprüfung bezieht sich darauf, ob die Anzahl der Zeichen eines Wertes die spezifizierte Länge<sup>36</sup> (Attribut laenge in Tabelle Feld) des Feldes überschreitet.

### Standardisierter Fehlertext für Längenfehler eines Ersatzfeldes

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' überschreitet die zulässige Feldlänge <Feld.laenge>.

### Standardisierter Fehlertext für Längenfehler eines Bogenfeldes

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

### Standardisierter Fehlertext für Muss-Fehler eines Zusatzfeldes

Das Zusatzfeld <Modul.name>:<Bogen.name>:<ZusatzFeld.name> '<ZusatzFeld.bezeichnung>' überschreitet die zulässige Feldlänge <ZusatzFeld.laenge>.

## Überprüfung der Schlüsselkodes

Die Überprüfung von Schlüsselkodes bezieht sich darauf, ob bei Schlüsselfeldern nur zulässige Schlüsselkodes verwendet werden.

### Standardisierter Fehlertext bei unzulässigen Schlüsselkodes eines Ersatzfeldes

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' ist kein zulässiger Kode des Schlüssels <Schluessel.name> (<Schluessel.bezeichnung>).

### Standardisierter Fehlertext bei unzulässigen Schlüsselkodes eines Bogenfeldes

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<Feld.name> '<BogenFeld.bezeichnung>' (Zeile <BogenFeld.zeileAufBogen>) ist kein zulässiger Kode des Schlüssels <Schluessel.name> (<Schluessel.bezeichnung>).

### Standardisierter Fehlertext für Muss-Fehler eines Zusatzfeldes

Das Zusatzfeld <Modul.name>:<Bogen.name>:<ZusatzFeld.name> '<ZusatzFeld.bezeichnung>' ist kein zulässiger Kode des Schlüssels <Schluessel.name> (<Schluessel.bezeichnung>).

## 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:

---

<sup>36</sup> Wenn bei einem Ersatz die Länge nicht spezifiziert ist, so entfällt die Prüfung.

- wert < min (nur prüfen, wenn min <> LEER)
- wert > max (nur prüfen, wenn max <> LEER)

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

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

Außerdem sind die Wertebereiche in den Ausfüllhinweisen zu den jeweiligen Datenfeldern angegeben.

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

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' ist kleiner als '<Feld.min>'.

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

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

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

Der Wert '<WERT>' des Datenfeldes <Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' ist größer als '<Feld.max>'.

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

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

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

### **Überprüfung der Muss-Felder**

Ein nicht ausgefülltes Muss-Feld (Abschnitt B 2.2.3) führt zu einer Regelverletzung.

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

Das Datenfeld '<Modul.name>:<Bogen.name>:<ErsatzFeld.name> '<ErsatzFeld.bezeichnung>' muss einen gültigen Wert enthalten.

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

Das Datenfeld '`<Modul.name>:<Bogen.name>:<Feld.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.

#### **2.3.9 Verfahren für die Evaluation von Regeln**

Grundsätzlich muss jede gem. Abschnitt B 2.3.2 formulierte Regel evaluiert werden, wenn keine der folgenden drei Bedingungen zutrifft:

1. Für mindestens ein referenziertes Bogenfeld<sup>37</sup> schlägt eine harte Feldprüfung (Abschnitt B 2.3) fehl.<sup>38</sup>
2. Ein Feld der Regel ist nicht ausgefüllt (LEER) und **keine** der folgenden Teilbedingung trifft in Bezug auf das leere Feld zu:
  - Es ist in einer Liste enthalten, die mit einem Listenoperator (EINSIN, KEINSIN, JEDESIN, EINSNICKTIN) geprüft wird bzw. wird direkt gegen eine Liste geprüft (IN, NICHTIN).
  - Es wird in der Regel explizit mit `<>` oder `=` auf LEER geprüft.
  - Das Feld der Regel befindet sich auf einem vorhandenen Teildatensatz. (D.h. eine Regel wird nicht geprüft, wenn der entsprechende Teildatensatz eines leeren Fels optional und im konkreten Fall nicht vorhanden ist).
  - Jeder Operation auf einen Wert `<>` LEER ist eine ODER-Verknüpfte Prüfung auf LEER direkt vorgeschaltet (Feld = LEER ODER Feld Operator Operand).
3. Eine Funktion der Regel hat das Ergebnis LEER und wird in der Regel nicht explizit mit `<>` oder `=` auf LEER geprüft.

Der Algorithmus zur Evaluation einer Plausibilitätsregel ist in Abbildung 5 beschrieben.

### **Umgang mit Laufzeitfehlern**

Bei der Evaluation von Regeln können, z. B. bei einem Vergleich von LEER < WERT, Laufzeitfehler entstehen. Solche Laufzeitfehler sind bei der Evaluation zu verhindern.

Laufzeitfehler bei der Evaluation von Regeln nach den Vorgaben sind immer ein Hinweis auf Syntaxfehler in der Regel. Das Ausbleiben von Laufzeitfehlern ist noch kein Beweis für die syntaktische Fehlerfreiheit einer Regel, da es auch fehlertolerante Parser geben könnte, die beim Verlassen des definierten Wertebereichs der Regel ein FALSCH zurückgeben können. Tatsächlich liegt aber hier ein undefinierter Zustand vor, der weder FALSCH noch WAHR ist.

---

<sup>37</sup> Die Verbindung zwischen Regeln und Bogenfeldern geschieht über die Tabelle `Regelfelder`, siehe `Regelfelder` (Bogenfelder einer Regel), Abschnitt B 2.3.1.

<sup>38</sup> Erst bei Fehlerfreiheit der feldbezogenen Prüfungen werden die feldübergreifenden Prüfungen durchgeführt.

Ein undefinierter Zustand an einem beliebigen Punkt im Evaluationsprozess muss für alle dort noch möglichen Werte durch die Struktur der Regel immer vermieden werden.

#### Algorithmus zur Evaluation einer Plausibilitätsregel

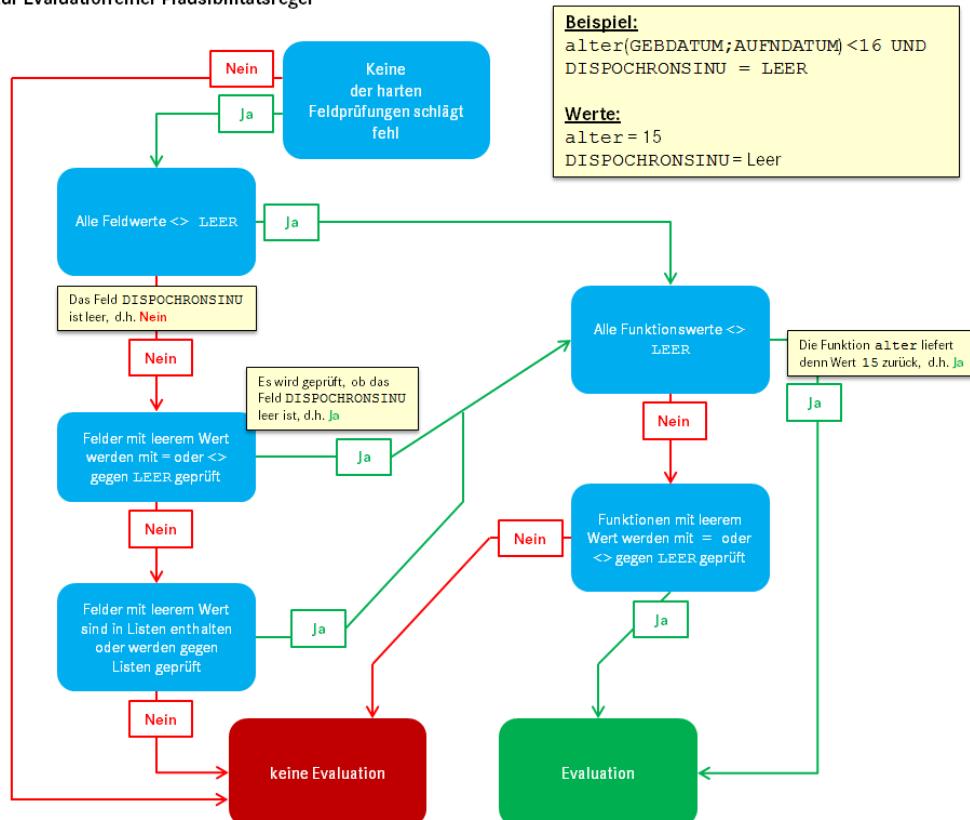


Abbildung 5: Algorithmus zur Evaluation von Plausibilitätsregeln

#### Teildatensatzübergreifende Regeln

Teildatensatzübergreifende Regeln (Abschnitt B 2.3.6) müssen u. U. mehrfach evaluiert werden (für jede Kombination von Teildatensätzen, die von der Regel betroffen ist).



#### Hinweis

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

## 2.4 Relationstabellen ICD- und OPS-Listen

Über die Tabellen ICDRelation und OPSRelation können Kode-Listen zu Gesamtlisten zusammengeführt werden, um diese z. B. für Plausibilitätsprüfungen zu nutzen oder um sie einem Bogenfeld zuzuordnen. Beispielsweise gibt es im Verfahren Cholezystektomie die Anforderung, dass in den Feldern zu Operationen und Diagnosen bestimmte Kodes, über die Einschlusskodes hinaus, aus definierten Listen übernommen werden müssen. Hierzu können die Einschlusslisten des QS-Filters mit der Liste der zusätzlichen Kodes zusammengeführt werden und die Gesamtliste in der Tabelle Bogenfeld referenziert werden.

**Beispiel:**

Die Liste `CHE_OPS_ProzAufenthalt` setzt sich zusammen aus den Einzellisten `CHE_OPS`, `OPS_Transfusionen` und `CHE_OPS_Aufenthalt`.

In den Tabellen `ICDWert` und `OPSWert` werden die entsprechenden Kodes nur für die Teillisten gepflegt. Die Werte der Gesamtlisten ergeben sich aus den Werten der Teillisten und werden in den Tabellen `ICDWert` und `OPSWert` nicht zusätzlich gepflegt.

**Hinweis**

Die Tabellen `ICDRelation` und `OPSRealation` wurden zur Angleichung der Datenbankstruktur an die Basisspezifikation in die Spezifikation der einrichtungsbezogenen QS-Dokumentation aufgenommen und werden derzeit dort nicht verwendet.

## 2.5 Exportfeldbeschreibung

Neben der Datenfeldbeschreibung (Abschnitt B 2.2) enthält die Spezifikationsdatenbank die Beschreibung der Exportfelder für ein Modul (Exportdatensatz). Diese werden zum Teil über Ersatzfelder berechnet. Es wird zudem zwischen Dokumentationsmodulen (Abschnitt B 2.2.1) und Exportmodulen unterschieden.

### 2.5.1 Exportmodule

Um die unterschiedlichen Abrechnungswege und davon abhängige Datenflüsse mit unterschiedlichen Datenannahmestellen für die Datenübermittlung zu berücksichtigen, wird zwischen den Dokumentationsmodulen (Tabelle `Modul`) und den Exportmodulen (Tabelle `Exportmodul`) unterschieden.

*Tabelle 38: Struktur der Tabelle `ExportModul`*

Feldname	Datentyp	Bemerkung
<code>idExportModul</code>	INTEGER	Primärschlüssel
<code>fkModul</code>	INTEGER	Bezug zum Dokumentationsmodul
<code>name</code>	KURZER TEXT	Technischer Name (muss eindeutig sein)
<code>bezeichnung</code>	KURZER TEXT	Bezeichnung des Exportmoduls
<code>existenzBedingung</code>	KURZER TEXT	Definiert, unter welcher Bedingung das Modul in ein definiertes Exportmodul transformiert wird.
<code>type_QS_data</code>	KURZER TEXT	Datentyp im XML-Schema
<code>ersatzBedingungMDS</code>	KURZER TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.

Feldname	Datentyp	Bemerkung
type_QS_data_mds	KURZER TEXT	Dieses Attribut hat derzeit keine Relevanz für die einrichtungsbezogene QS-Dokumentation.
fkExportzeitraumEntlSJ	INTEGER	Fremdschlüssel für die Tabelle mit den Exportzeiträumen
fkExportzeitraumEntlSJ1	INTEGER	Fremdschlüssel für die Tabelle mit den Exportzeiträumen
pid	BOOLEAN	Wird in diesem Modul die eGK-Versichertennummern zur Pseudonymisierung an die VST exportiert? <sup>39</sup>
vpb	BOOLEAN	Wird dieses Modul an die Versendestelle zur Patientenbefragung exportiert?
fkPseudonymVerfahren	INTEGER	Pseudonymisierung von FU-Verfahren <sup>40</sup>
fkRichtlinie	BOOLEAN	Verweis auf die für das Exportmodul gültige Richtlinie

Das Attribut `type_QS_data` gibt Auskunft darüber, welchem Datentyp dies im XML-Schema entspricht.

---

### Beispiel:

Beim Exportmodul SEPE ist der Datentyp im XML-Schema `qs_data_sepe_type`.

---

Softwareanbietern soll hiermit die Integration eines Mechanismus der automatischen Datentypzuweisung ermöglicht werden, um den Aufwand zu reduzieren und Fehler zu vermeiden.

## 2.5.2 Modulauslöser, Dokumentations- und Exportmodule

### Modulauslöser

In der Spezifikationsdatenbank für den QS-Filter (QSF) erfolgt z. B. für das Dokumentationsmodul SEPE eine Aufteilung der Modulauslöser analog zu dem Exportmodul SEPE (Abschnitt B 2.5.1).

- Der Modulauslöser SEPE löst den QS-Datensatz SEPE aus.

### Dokumentationsmodule

In der Spezifikationsdatenbank für die QS-Dokumentation ist das Dokumentationsmodul SEPE definiert. Sowohl für den Krankenhaussektor als auch für den vertragsärztlichen Sektor werden jeweils die gleichen Dokumentationsbögen ausgefüllt.

- Das Dokumentationsmodul SEPE wird von Einrichtungen, die stationäre Fälle am Krankenhaus

---

<sup>40</sup> Gemeint ist ein Follow-up-Verfahren anhand von PID-Daten.

behandeln, dokumentiert.

### **2.5.3 Über das Datenfeld „Status des Leistungserbringers“ im Modul SEPE gibt der Dokumentierende an, ob sich der Datensatz auf den vertragsärztlichen Sektor oder auf den Krankenhaussektor bezieht. Exportdatensatz**

Der Exportdatensatz enthält die Exportfelder für ein Modul. Welche Zusatzfelder, Bogenfelder und/oder Ersatzfelder den Exportdatensatz pro Modul bilden, ist in Tabelle ExportFormat definiert. Weiterführende Informationen zum Exportdatensatz sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

### **2.5.4 Umgang mit Umlaut-Domains und E-Mail-Adressen**

Die Nutzung von Umlauten ist durch die Internationalized Domain Names (IDN) möglich. Domannamen wie z. B. <http://www.qualitätsklinikverbund.de/> sind damit nutzbar. Da die Verwendung von Zeichen aus einem Unicode-Zeichensatz in Domannamen und E-Mail-Adressen immer noch Probleme in Anwendungsprogrammen bereiten, wurde das ASCII Compatible Encoding (ACE) entwickelt, in dem Regeln zur Überführung von Unicode-Zeichen in ASCII-Zeichen festgelegt wurden. Zu jeder Umlautdomain (IDN-Domain) wird parallel immer eine ACE-Domain registriert. Für das oben genannte Beispiel <http://www.qualitätsklinikverbund.de/> würde der ACE-Name xn--qualittsklinikverbund-b2b.de lauten.

E-Mail-Adressen dürfen im lokalen Teil (vor dem @-Zeichen) keine Umlaute enthalten. Die E-Mail-Adresse müller@muster.de wäre nicht möglich. Möglich ist lediglich mueller@muster.de. Es existieren experimentelle Request for Comments (RFCs)<sup>41</sup>, in denen UTF-8-codierte, internationale E-Mail-Adressen ermöglicht werden. Dies wird jedoch nicht flächendeckend im Produktiveinsatz genutzt, aus diesem Grund wird in dieser Spezifikation von der Annahme ausgegangen, dass der lokale Teil der E-Mail-Adresse immer durch ASCII-Zeichen repräsentiert wird.

In den E-Mail-Elementen des XML-Schemas (siehe Abbildung 5) sowie in der Auftragsdatei (siehe Abschnitt 5.1) sind ausschließlich Zeichen des ASCII-Zeichensatzes erlaubt. E-Mail-Adressen mit einem IDN Domannamen sind entsprechend den ACE-Regeln durch ASCII-Zeichen darzustellen und zu übermitteln.

## **2.6 Versionierung**

Im Folgenden werden die Tabelle Version, der Abgleich zu vorherigen Versionen, die Abgrenzung zwischen Spezifikationsjahren und Datensatzformaten sowie die Version von Exportverfahren und -dateien beschrieben.

### **2.6.1 Grundlegende Definitionen**

In der Tabelle Version finden sich Informationen zur Version der Spezifikationsdatenbank.

---

<sup>41</sup> <https://datatracker.ietf.org/doc/html/rfc5336>

Die wichtigsten Eigenschaften einer Version sind der Versionsname (Attribut `name`) und der Gültigkeitszeitraum (Attribute `ab` und `bis`).



### Achtung

Der Gültigkeitszeitraum einer Version ist in der fallbezogenen QS-Basis spezifikation für Leistungserbringer in der Regel ein Spezifikationsjahr (z. B. Aufnahme zwischen dem 01.01.2026 und dem 31.12.2026).

Die QS-Dokumentationssoftware eines Spezifikationsjahrs wird für diejenigen Einrichtungen verwendet, die Fälle mit Aufnahmedatum in das Krankenhaus (stationäre Fälle) im oben definierten Gültigkeitszeitraum behandeln.

Jedes Modul der Datenbank hat eine Version (vgl. Attribut `fkVersion` in Tabelle `Modul`). In einer Spezifikationsdatenbank können Module unterschiedlicher Versionen enthalten sein. Über die in der Datenbank definierten Relationen sind auch für alle Bogenfelder (Tabelle `BogenFeld`), Exportfelder (Tabelle `ExportFormat`) und Plausibilitätsregeln (Tabelle `Regeln`) Versionen definiert.



### Hinweis

Grundsätzlich baut eine QS-Dokumentationsdatenbank für ein Spezifikationsjahr auf der QS-Dokumentationsdatenbank aus dem vorherigen Spezifikationsjahr auf. Das Delta wird in den Tabellen `DeltaAttribut`, `DeltaGeloescht` und `DeltaNeu` gelöscht. Sollte nach der ersten Veröffentlichung der QS-Dokumentationsdatenbank für ein Spezifikationsjahr noch ein Update der QS-Dokumentationsdatenbank aus dem vorherigen Spezifikationsjahr notwendig werden, dann divergieren beide Datenbanken. Würde man beide Datenbanken kombinieren, dann wären die IDs nicht garantiert rechtseindeutig. Eine `idVersion` könnte dann zum Beispiel auf unterschiedliche Werte in `QSDOK.Version` verweisen.

## Status der Spezifikation

Versionen können den Status `in Entwicklung`, `final` oder `Update` der finalen Spezifikation haben. Diese Zustände werden zum Nachschlagen in der Tabelle `VersStatus` verwaltet. Das Attribut `gueltig` zeigt die gültige Version der Datenbank an. Es darf nur eine einzige Version als gültig markiert sein.

### Beispiel:

Die Version 2026 EDOK V01, einschließlich ihrer Updates, sind finale Versionen.

Hat eine Spezifikationsdatenbank den Status `in Entwicklung`, kann `Modul.fkVersion` als ungültig markierte Versionen enthalten, um Zwischenstände abzubilden.

## Historie der Versionen

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

### 2.6.2 Delta-Informationen zur vorhergehenden Version

Um den Benutzern der Spezifikation umfassende Informationen zu den jeweiligen Änderungen zur Verfügung zu stellen, enthält die Spezifikationsdatenbank Tabellen, die den Änderungsstand im Vergleich mit der letzten gültigen Version des Vorjahres und zur Vorversion der Datenbank anzeigen. Diese sogenannten Delta-Tabellen werden automatisch generiert. Weiterführende Erläuterungen zu den Delta-Informationen zur vorhergehenden Version sind der Technischen Dokumentation zur Basisspezifikation für Leistungserbringer zu entnehmen.

## 2.7 Administrative Objekte

Die Datenbank für QS-Dokumentation beinhaltet neben den Dokumentationsobjekten eine Gruppe von Tabellen, die Hilfestellungen für die Einhaltung eines korrekten Datenflusses und standardisierter Prüfprozesse geben.

Zu den administrativen Objekten gehören Mapping-Informationen mit denen QS-Daten in Form von Metainformationen zu administrativen Zwecken außerhalb des eigentlichen QS-Containers (dem Element `<qs-data>`) im XML verortet werden können. Außerdem gibt es mit der Abfrage `vPruefung` eine Auflistung von Prüfungen, die für den jeweiligen Datenservice zu implementieren sind. Um eine Ansicht der administrativen Objekte zu erhalten, ist in Access (Spezifikationsdatenbank zur QS-Dokumentation) oberhalb der Objektübersicht das Drop-Down-Menü zu öffnen und der Menüpunkt „Benutzerdefiniert“ auszuwählen.

Die für die Leistungserbringer relevanten Empfänger der Datenlieferungen sind in einer separaten Datenbank aufgelistet (Abschnitt B 2.7.2). Sie enthält auch Informationen über die zu verwendenden XML-Schlüssel.

### 2.7.1 XML-Mapping in der Spezifikationsdatenbank (QSDOK)

Exportfelder aus dem QS-Datensatz sind nicht immer Kindelemente von `case/qs_data` in der XML-Exportdatei. Die beiden Tabellen `ExportZiele` und `ExportZielXml` enthalten die Informationen, welche Exportfelder wohin verschoben, kopiert oder gelöscht werden. Die Tabelle `ExportFormatExportModul` zeigt neben diesen referenzierten Exportfeldern auch die Felder, die Kindelemente von `qs_data` bleiben.

Die über das Attribut `fkFeld` referenzierten Felder gelten für alle Module, in denen dieses Feld verwendet wird. Wird ein Feld (gleichzeitig) über unterschiedliche Feldarten referenziert, überlagern detailliertere Angaben die allgemeinen. Wird so beispielsweise in der Tabelle `ExportZiele` ein Feld allgemein über den Fremdschlüssel `fkFeld` und in einer weiteren Zeile modulspezifisch über `fkBogenFeld` referenziert, werden die allgemeinen Regelungen (`loeschenQS, exportWennLeer, xmlXPath` und `xmlGruppierung`) für das Feld mittels der

Referenzierung über `fkFeld` mit den spezifischeren Regelungen für das modulbezogene Feld mittels der Referenzierung über `fkBogenFeld` für das konkret aus der Referenzierung hervorgehende Modul überschrieben.

Dabei ist ebenfalls eine etwaige Konkretisierung über `fkExportModul` zu beachten, welche Ihrerseits konkretisierenden Charakter hat. Die über das Attribut `fkErsatzFeld` referenzierten Ersatzfelder gelten für alle Exportmodule, in denen dieses Ersatzfeld verwendet wird. Bei Referenzierung des gleichen Feldes (insbesondere bezogen auf `fkFeld`, `fkErsatzFeld` bzw. `fkZusatzfeld`) in zwei Zeilen – einmal ohne und einmal mit Angabe von `fkExportmodul` – so überschreibt der Eintrag mit Angabe den Eintrag ohne Angabe für das konkret benannte Exportmodul.

Die booleschen Spalten `loeschenQS` und `exportWennLeer` geben folgende Informationen an:

- `loeschenQS`: Das Datum wird nur an den/die alternativen Ort(e) geschrieben und taucht in den QS-Daten nicht mehr auf. Hier geht es zum Beispiel um vom Datenschutz betroffene Felder, die nur in bestimmte Bereiche des XML geschrieben werden dürfen, wo sie dann verschlüsselt werden können.
- `exportWennLeer`: Hier wird ein leeres Feld nur dann berücksichtigt und leer eingetragen, wenn dieser Wert auf `true` steht.

In der Spalte `xmlXPath` vom Typ String steht ein XPath-Ausdruck, der ein Mapping auf einen im XML liegenden Ort des betroffenen Feldes enthält. Ausgangspunkt des XPath-Ausdrucks ist immer der aktuelle Datensatz im XML, also das `<case>` Element.

Die boolesche Spalte `xmlGruppierung` gibt an, dass ein neuer XML-Knoten nur eingesetzt wird, wenn das Datum einen neuen Wert enthalten sollte.

Einen ergänzenden Überblick über die zu exportierenden Felder eines Exportmoduls liefert die Tabelle `ExportFormatExportModul`. Sie zeigt alle möglichen Exportfelder für einen Datensatz (`case`) und informiert über deren Häufigkeit, Typ und `xpath`.

*Tabelle 39: Struktur der Tabelle `ExportFormatExportModul`*

Feldname	Bemerkung
<code>idExportFormatExportModul</code>	Primärschlüssel
<code>Exportmodul_name</code>	name aus Tabelle <code>Exportmodul</code>
<code>qs_data_xsi_type</code>	<code>type_QS_data</code> oder <code>type_QS_data_mds</code> aus Tabelle <code>Exportmodul</code> (Wenn Wert null, dann führt <code>xpath</code> zu keinem Kindelement von <code>qs_data</code> .)
<code>Bogen_name</code>	name aus Tabelle <code>Bogen</code> (Wenn Wert null, dann führt <code>xpath</code> zu keinem Kindelement von <code>qs_data</code> .)
<code>fkBogen</code>	Fremdschlüssel zur Tabelle <code>Bogen</code>

Feldname	Bemerkung
BogenZahl_bzgl_mutterbogen	Anzahl der Teildatensätze/Bögen bezogen auf den Mutterbogen (QSDOK.Bogen .fkBogenZahl)
Bogen_mutterbogen_name	name aus Tabelle Bogen
fkMutterbogen	Schlüssel zum Mutterbogen
exportfeld	name entweder aus Tabelle Zusatzfeld, Ersatzfeld oder Feld
lfdNr	Sortierhilfe innerhalb eines Bogens; untere Grenze kann größer als 1 sein (lfdNr 4 kann bspw. dem xpath child::*[1] entsprechen); wenn Wert gleich 0, dann führt xpath zu keinem Kindelement von qs_data.
maxAnzahl	maximale Anzahl des Exportfeldes auf dem Bogen (Wenn fkBogen gleich null, dann maximale Anzahl im Datensatz.)
BasisTyp_name	name aus Tabelle BasisTyp
fkBasisTyp	Fremdschlüssel zur Tabelle BasisTyp
maxZeichenlaenge	maximale Zeichenlänge
xpath	xpath innerhalb der Exportdatei
fkMussKann	Fremdschlüssel zur Tabelle MussKann
fkBogenFeld	Fremdschlüssel zur Tabelle BogenFeld
BogenFeld_name	name aus Tabelle BogenFeld
BogenFeld_bezeichnung	bezeichnung aus Tabelle BogenFeld
fkErsatzFeld	Fremdschlüssel zur Tabelle ErsatzFeld
ErsatzFeld_name	name aus Tabelle ErsatzFeld
ErsatzFeld_bezeichnung	bezeichnung aus Tabelle ErsatzFeld
fkZusatzFeld	Fremdschlüssel zur Tabelle ZusatzFeld
ZusatzFeld_name	name aus Tabelle ZusatzFeld
ZusatzFeld_bezeichnung	bezeichnung aus Tabelle ZusatzFeld
fkModul	Fremdschlüssel zur Tabelle Modul
Modul_name	name aus Tabelle Modul
fkExportZiele	Fremdschlüssel zur Tabelle ExportZiele
fkExportmodul	Fremdschlüssel zur Tabelle ExportModul
fkExportZielXml	Fremdschlüssel zur Tabelle ExportZielXml

Feldname	Bemerkung
Bogen_sortierNr	sortierNr aus Tabelle Bogen
fkExportformat	Fremdschlüssel zur Tabelle ExportFormat

### Hinweis

XML-Kindeglemente von `qs_data`-Elementen werden in der Exportdatei immer gesetzt, sobald der Teil-datensatz in der Exportdatei vorhanden ist. Sollte bspw. kein Wert für Exportfeld `feld_2` vorliegen, dann ist die Abbildung in der Exportdatei:

```
<feld_1 V="foo"/>
```

```
<feld_2 />
```

```
<feld_3 />
```

Falsch wäre und schemaverletzend:

```
<feld_1 V="foo"/>
```

```
<feld_3 />
```

Die Optionalität wird also durch das XML-Attribut `V` abgebildet.

Bei `qs_data` Kindern mit `maxAnzahl > 0` muss nur das erste XML-Kindeglement geschrieben werden, wenn keine Werte zu exportieren sind.

### 2.7.2 Datenservices

In einer separaten Datenbank sind Angaben über die beim Export relevanten technischen Daten-services und ihre verfahrensbezogene und regionale Zuordnung zu finden. Diese Datenbank zu Datenserviceinformationen ist hier zu finden:

<https://iqtiq.org/datenerfassung/spezifikationen/spezifikation-zu-datenserviceinformationen/>

Die einzelnen Datenservices werden zusammen mit den E-Mail-Adressen, an die die Exportdateien zu versenden sind, und den für die Verschlüsselung der QS-Daten zu verwendenden XML-Schlüsseln in der Tabelle Datenservice abgebildet:

fklInstitution	funktion	sektor	datenabsender	email
QiG-BW	DAS	Krankenhaus	LE	daten-bw@unitrend.de
LAG-Bayern	DAS	Krankenhaus	LE	daten-ba@unitrend.de
LAG Sachsen	DAS	Krankenhaus	LE	daten@qs-sachsen.de
KGSH	DAS	Krankenhaus	LE	qsdaten@kgsh.de

fklInstitution	funktion	sektor	datenabsender	email
QiG-BW	DAS	Krankenhaus	LE	daten-bw@unitrend.de
LAG-Bayern	DAS	Krankenhaus	LE	daten-ba@unitrend.de
LAG Sachsen	DAS	Krankenhaus	LE	daten@qs-sachsen.de
BQS-Institut	DAS	Krankenhaus	LE	xmldaten@bqs-institut.de

Abbildung 6: Tabelle Datenservice

Die einzelnen Datenannahmestellen, die ihre Datenannahme mithilfe eines oder mehrerer dieser Datenservices realisieren, sind in der Tabelle Region hinterlegt, während die Zuordnung eines Datenservice zu einer Region abhängig vom Modul in der Tabelle DatenserviceModul festgelegt ist.

Eine zusammenfassende Darstellung der wichtigsten Merkmale ergeben sich aus den beiden Abfragen DatenserviceExportverfahren und DatenserviceExportModule. Die erste Abfrage aggregiert auf Ebene der Exportverfahren, während die DatenserviceExportModule eine weitergehende Differenzierung für die einzelnen Exportmodule anbietet.

### 2.7.3 Prüfschritte

In der Gruppe administrative Objekte der QSDOK-Datenbank befindet sich die Abfrage vPruefung. vPruefung zeigt für die Institutionsarten jeweils die Prüfungen, die wichtig für die Beurteilung der Güte der angelieferten Daten sind.

Die Datei datenflussprotokoll.xml in der Spezifikationskomponente XML-Schemata zeigt exemplarisch die Protokollierung der Prüfungen aus vPruefung. In der XML-Dokumentation ist die Protokollierung unter dem Xpath root/header/protocol zu finden.<sup>42</sup>

Tabelle 40: Felder der Abfrage vPruefung

Feldname	Bedeutung	Bezugselement im XML
Zielgruppe	Institutionsart, die die Prüfung ausführt	-
idPruefung	Referenz auf die konkrete einzelne Prüfung	-
pruefung	Kurzbeschreibung der Prüfung	optional unter: validation_provider/validation_item/@description
dpp	True/False, Prüfung ist im Datenprüfprogramm enthalten	-
meta_DS	meta_DS soll ausdrücken, ob das Prüfergebnis das Persistieren der Metadaten (case/case_admin) eines Datensatzes verhindert. Wenn eine verletzte Prüfung die Eigenschaft meta_DS = true hat, dann werden die Metadaten nichtpersistiert. Sind nur Prüfungen mit meta_DS =	

<sup>42</sup> In der XML-Dokumentation sind die Kind-Elemente in der Zeile Model verlinkt.

Feldname	Bedeutung	Bezugselement im XML
	false verletzt, müssen die Metadaten aus case/case_admin unabhängig von status_document und status_case persistiert werden.	
beschreibung	ggf. erweiterte Erläuterungen zur Prüfung	-
fkFehlermeldung	Fehlerreferenz	validation_item/error/rule_id/@v
Bereich	Einordnung der Prüfung	validation_item/@v
strenge	Konsequenz im Fehlerfall (Hinweis, Fehler)	validation_item/error/rule_type/@v In der Exportdatei ist ERROR auf H abzubilden und WARNING auf W.
regelverletzung	Bedingungen, die zu einer Regelverletzung führen	-
fehlermeldung	Standardisierte Fehlermeldung	validation_item/error/error_message/@v
parameter	Parameter für Fehlermeldung	-
protokoll-Level	Zuordnung der Prüfung: DK-> betrifft Dokument DS-> betrifft Datensatz	-
datenfluss	direkt, indirekt, pid, sv, vpb	
verursacher	Verursacher des Fehlers	error/@originator
sortierung	mögliche Prüfungsreihenfolge	

### Das Attribut Pruefung.dpp

Das Attribut dpp ist TRUE, wenn die entsprechende Fehlerprüfung über das Datenprüfprogramm abgebildet wird. Hierbei muss die Prüfung (mit der jeweiligen Prüfnummer idFehlermeldung) nicht eins-zu-eins im Datenprüfprogramm referenziert sein. Das Attribut ist auch dann TRUE, wenn es im Datenprüfprogramm eine analoge Prüfung gibt, die denselben Sachverhalt prüft und sicherstellt, dass keine Regelverletzung auftritt. Hierbei kann es sich auch um eine Schemaprüfung handelt. Das Attribut dpp dient dem Zweck (z. B. einer Datenannahmestelle) eine Übersicht über die Prüfungen zu geben, die bereits über das Datenprüfprogramm geprüft werden und keine separate (z. B. durch die Datenannahmestelle implementierte) Zusatzprüfung erfordern

## 3 XML-Schema

Die XML-Schema-Datei (XSD) ist eine Empfehlung des W3C<sup>43</sup> zum Definieren von Strukturen für XML-Dokumente.

In der Spezifikation der einrichtungsbezogenen QS-Dokumentation werden die XML-Schemata aus der Spezifikationsdatenbank abgeleitet und haben die Aufgabe, die aktuellen Datenflussmodelle der G-BA-Richtlinien abzubilden:

- Abbildung der zu exportierenden QS-Daten
- Abbildung der Schnittstellen an den Institutionsübergängen (LE und DAS, DAS und BAS)
- Abbildung der vorzunehmenden Datentransformation in den jeweiligen Einrichtungen des Datenflusses  
(z. B. LE-Pseudonymisierung bei der DAS)
- Abbildung der Rückprotokollierung

Aus diesem Grund gibt es bei der Erstellung von Schemata, welche die Konformität von Richtlinie und Datenschutz sicherstellen sollen, kein „Allround-Schema“, welches alle Anforderungen an alle Beteiligten abdeckt, sondern eine „Schema-Familie“, aus der heraus gezielt für jede Schnittstelle („Interface“) eine passende Datenstruktur definiert wird.

### 3.1 Kompositionsmodell

Bei der Schema-Erstellung wurde auf ein Kompositionsmodell zurückgegriffen, in dem sich alle Teilschemata am Ende einen Namensraum teilen, um diese Schema-Familie besser warten zu können und gleiche Teilstrukturen nur einmal definieren zu müssen. Als Bezeichnung des Namensraums wurde „urn:gba:sqg“ gewählt. Zu diesem Namensraum werden die Bausteine je nach Bedarf über „includes“ zusammengestellt.

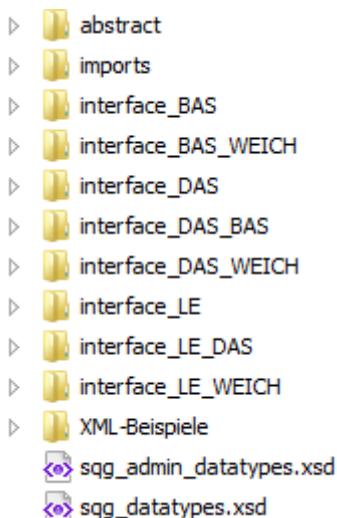


Abbildung 7: Dateiordner der Schnittstellen-Schemata

<sup>43</sup> <http://www.w3.org/XML/Schema>

Das KompositionsmodeLL ermöglicht es, Konzepte aus der objektorientierten Programmierung – darunter fallen die Konzepte abstrakter Typ, Ersetzbarkeit von Typen, Wiederverwendung und Polymorphismus – zu nutzen. Dadurch können Schemata erstellt oder genutzt werden, die generische Grundtypen definieren und diese Typen so erweitern, dass sie schnittstellenspezifisch sind, ohne das ursprüngliche Schema zu beeinflussen. Dieses KompositionsmodeLL wird hier näher erläutert.

### **Beispiel:**

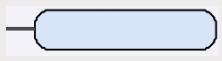
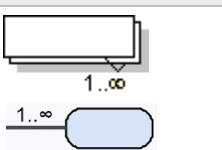
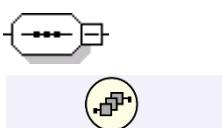
Die leistungserbringeridentifizierenden Daten existieren gemäß Datenflussmodell der G-BA-RL in drei Ausprägungen:

- im Klartext (Schnittstelle LE)
- pseudonymisiert (Schnittstelle DAS)
- verschlüsselt (Schnittstelle DAS-BAS)

Bei dieser Konzeption werden alle drei Ausprägungen vom selben Basisdatentyp geerbt, in einem zweiten Schritt die drei Ausprägungen konkretisiert und angepasst, und über „includes“ in die jeweilige Schnittstelle integriert.

## **3.2 Darstellung der XML-Struktur**

Zur Veranschaulichung der verwendeten XML-Schemata werden Diagramme verwendet, deren Symbole in der folgenden Tabelle kurz dargestellt und erläutert werden.

Symbol	Beschreibung
	<b>Optionales Element</b> Kardinalität 0..1 („0 oder 1“)
	<b>Obligatorisches Element</b> Kardinalität 1: das Element muss genau einmal vorkommen
	<b>Mehrfach wiederholbares Element</b> Kardinalität: die erlaubte Anzahl der Elemente wird unter dem Symbol dargestellt (Beispiel: 1..n, n..m).
	<b>Referenzelement</b> Das referenzierte globale Element ist an anderer Stelle im Schema definiert.
	<b>Eine Folge von Elementen</b> Die Elemente müssen genau in der Reihenfolge vorkommen, in der sie im Schemadiagramm angezeigt sind.
	<b>Eine Auswahl von Elementen</b> Nur ein einziges Element aus der Liste kann ausgewählt werden.

Symbol	Beschreibung
	Ein Element mit Kind-Elementen
	<b>Optionales Attribut</b> Kardinalität 0..1 („0 oder 1“)
	<b>Obligatorisches Attribut</b> Kardinalität 1: das Attribut muss genau einmal vorkommen
	<b>Komplexer Datentyp</b> Der komplexe Datentyp wird mit einem Rahmen mit einem weißen bzw. lilafarbenen Hintergrund angezeigt.

Die wesentlichen Bestandteile der XML-Schemata werden in den nachfolgenden Abschnitten dargestellt. Die Darstellung umfasst folgende Eigenschaften des betrachteten Elements:

- Grafische Abbildung der Kind-Elemente und -Attribute
- Auflistung der Kind-Elemente
- Auflistung der Kind-Attribute sowie ihrer Eigenschaften wie:
  - Name
  - XML-Datentyp (technische Bezeichnung: „Type“)
  - Muss-Kann-Feld (technische Bezeichnung: „Use“)
  - Konstante (technische Bezeichnung: „Fixed“)
  - Kurze Beschreibung (technische Bezeichnung: „Annotation“)

### 3.3 Aufbau der XML-Exportdatei

Grundsätzlich beginnt jede XML-Exportdatei mit einer Headerzeile gefolgt vom Wurzelement <root>, das den gesamten Inhalt einschließt. Als Zeichensatz wird „UTF-8“ (Unicode-Codierung) verwendet.

---

#### Beispiel:

Headerzeile

```
<?xml version="1.0" encoding="UTF-8"?>
<root> </root>
```

---

#### 3.3.1 Namensräume

Die Schemata für den XML-Datenfluss sind für den Namensraum mit der id "urn:gba:sqg" definiert. Dieser soll ohne Präfix-Mapping im <root>-Element in das XML eingebunden werden.

Falls ein Ablageort des zugrundeliegenden Schemas angegeben werden soll, wird dieses im <root>-Element mit dem Attribut "schemaLocation" vorgenommen. Da dieses Attribut ebenfalls aus einem externen Namensraum stammt, wird dieser Namensraum dem reservierten Präfix "xsi" zugeordnet, was sich dann als "xsi:schemaLocation='...'" liest:

- xmlns:xsi=<http://www.w3.org/2001/XMLSchema-instance>
- xsi:schemaLocation="urn:gba:sqg\_interface\_LE\_lqs.xsd"

Für die Verschlüsselung der XML-Elemente werden zwei externe Namensräume mit der id:

- " xmlns:ds=<http://www.w3.org/2000/09/xmldsig#>"  
und
- " xmlns:xenc=<http://www.w3.org/2001/04/xmlenc#>"

verwendet.

Diese sollen auf die reservierten Präfixe "ds" und "xenc" zugeordnet eingebunden werden. Es ist zu empfehlen, die externen Namensräume in das <root>-Element einzubinden, um lokale Wiederholungen auf Elementebene zu vermeiden.

---

**Beispiel:**

```
<root container_version="2.0" content_version="1.0"
      xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
      xmlns="urn:gba:sqg"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
</root>
```

---

### 3.3.2 Wurzelement <root>

Das Root-Element ist eine Art Umschlag oder Wurzelement für alle XML-Typen in den QS-Verfahren. Das Wurzelement besteht immer aus zwei Kind-Elementen (Zweige) <header> und <body>.

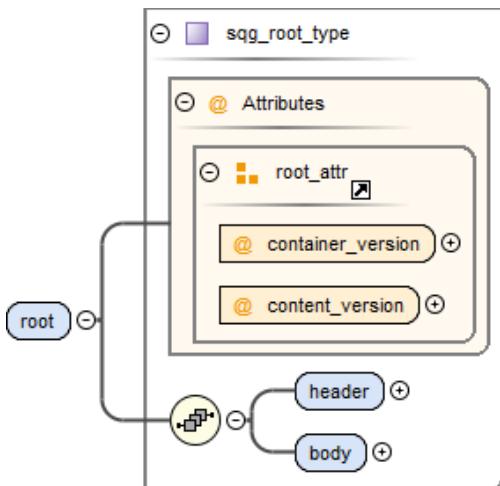


Abbildung 8: Root-Element und Kind-Elemente header und body

Das Root-Element hat zusätzlich zwei Attribute (Tabelle 41).

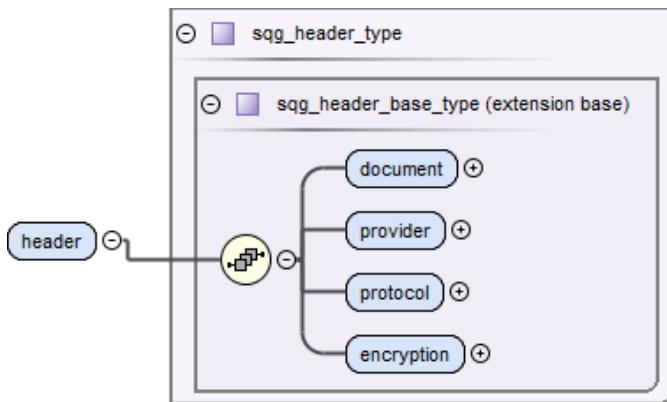
Tabelle 41: Root-Element – Attribute

Name	Type	Use	Fixed	Beschreibung
container_version	xs:string	required	2.0	<p>Ist ein fixer Wert und definiert die aktuell gültige Versionsnummer des Containers.</p> <p>Die Versionsnummer wird erhöht, wenn Änderungen am Schema des Containers (Umschlags) gemacht werden. Bei kleinen optionalen Änderungen wird die Versionsnummer beibehalten, um die Aufwärtskompatibilität zu gewährleisten.</p> <p>Ein XML-Dokument, das einen alten Wert dieses Attributs enthält, muss von der Datenannahmestelle zurückgewiesen werden.</p>
content_version	content_version_Datentyp	required	1.0	<p>Ist ein fixer Wert und definiert die aktuell gültige Versionsnummer des Inhalts der QS-Daten.</p> <p>Die Versionsnummer wird erhöht, wenn unterjährig das Schema unabhängig von der zugrunde liegenden Spezifikationsdatenbank geändert wird.</p>

### 3.3.3 Header-Bereich

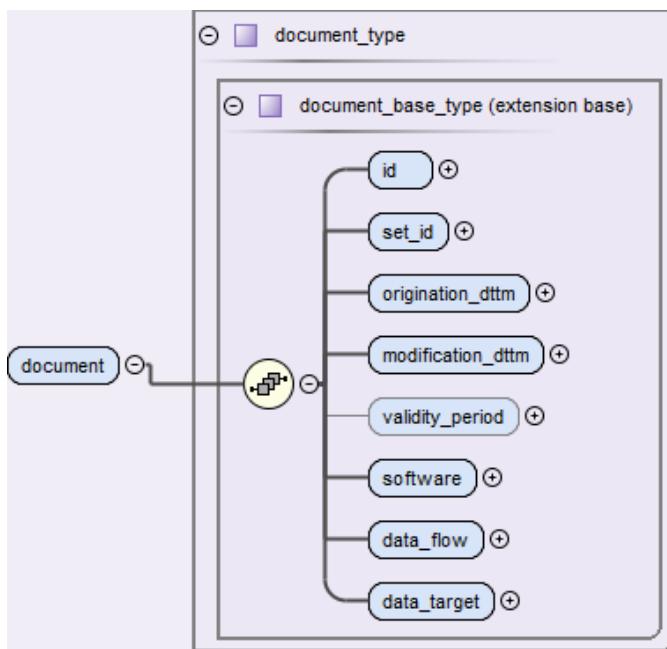
#### Element header

Das Element `Header` besteht aus Metadaten (administrative und meldebezogene Daten) zu den QS-Daten, die im `<body>` enthalten sind.

Abbildung 9: Aufbau des Elements *header*

### Element header/document

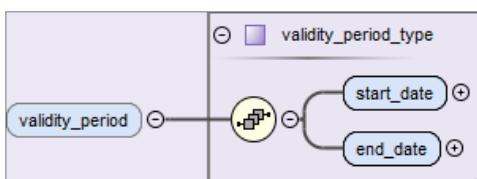
Das Element enthält allgemeine Informationen zum erstellten Dokument. Dieses Element ist weitestgehend über den gesamten Datenfluss hinweg beständig. Nur das Element <modification\_dttm> (Modifikationsdatum) wird bei jeder Bearbeitung neu gesetzt.

Abbildung 10: Aufbau des Elements *document*

Dieses Element hat weitere Kind-Elemente, die in der folgenden Tabelle beschrieben werden.

Tabelle 42: Kind-Elemente des Elements *document*

Kind-Elemente	Beschreibung
<id>	Nach Erstellung nicht mehr modifizierbar.

	Eindeutige ID des Dokuments, wird vom Dokumentenersteller als GUID erzeugt. <sup>44</sup>
<set_id>	Nach Erstellung nicht mehr modifizierbar. Eindeutige ID für mehrere Dokumente, die im selben Zusammenhang stehen; wird vom Leistungserbringer erzeugt. Dafür könnte z. B. die GUID vom ersten Dokument des Zusammenhangs verwendet werden.
<origination_dttm>	Das Element ist der Zeitstempel der ursprünglichen Dokumenterzeugung. Darf nach seiner Erstellung nicht mehr modifiziert werden. Format: CCYY-MM-DDThh:mm:ss
<modification_dttm>	Dieses Element ist ein Zeitstempel und muss bei jeder Modifikation des Dokuments aktualisiert werden. Das Modifikationsdatum darf nicht vor dem Erstelltdatum liegen. Format: CCYY-MM-DDThh:mm:ss
<validity_period>	 <p>Mithilfe des optionalen &lt;validity_period&gt;-Elements wird der Bezugszeitraum der Datei angegeben. Dies kann ein Jahr oder ein Quartal oder ein frei wählbarer Bereich sein. Zur Qualifizierung sind die Felder &lt;start_date&gt; und &lt;end_date&gt; entsprechend zu füllen</p>

<origination\_dttm> und <modification\_dttm> sind vom Datentyp `dateTime`, der einen Zeitpunkt darstellt (ISO 8601). Es handelt sich um das Format CCYY-MM-DDThh:mm:ss:

- „CC“ steht für das Jahrhundert,
- „YY“ steht für das Jahr,
- „MM“ steht für den Monat und
- „DD“ für den Tag.
- Der Buchstabe „T“ dient als Trennzeichen zwischen Datum und Zeit.
- „hh“, „mm“ und „ss“ repräsentieren jeweils Stunden, Minuten und Sekunden.

Dieser Darstellung kann direkt ein „Z“ nachgestellt werden, um anzugeben, dass es sich um die Universal Time Coordinated (UTC) handelt. Folgt der Zeitangabe statt eines „Z“ ein Plus- oder Minuszeichen bedeutet das, dass die darauffolgende Angabe im Format „hh:mm“ die Differenz zur UTC angibt (der Minutenanteil ist erforderlich).

### Beispiele:

- 2011-11-01T21:32:52
- 2011-11-01T21:32:52+02:00 (Zeitzonendifferenz von plus 2 Stunden)
- 2011-11-01T19:32:52Z

<sup>44</sup> Ein Globally Unique Identifier oder kurz GUID ist eine global eindeutige Zahl mit 128 Bit (16 Bytes), die ein Dokument weltweit eindeutig identifiziert.

## Element header/document/software

Sammelement für Angaben zur eingesetzten QS-Software.

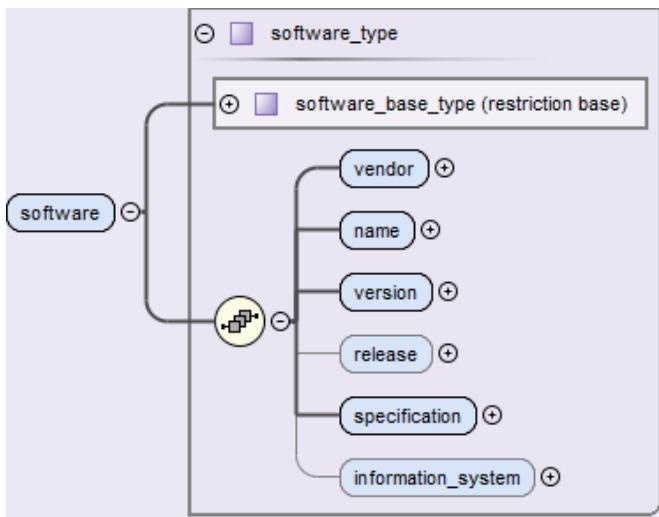


Abbildung 11: Aufbau des Elements software

Dieses Element enthält Kind-Elemente, die in der folgenden Tabelle beschrieben werden.

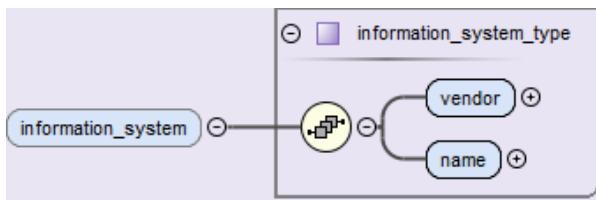
Tabelle 43: Kind-Elemente des Elements software

Kind-Elemente	Beschreibung
<vendor>	Enthält Informationen über den Softwarehersteller
<name>	Enthält den Softwarenamen der eingesetzten Software
<version>	Enthält die Version der eingesetzten Software
<release>	Enthält das Release der eingesetzten Software
<specification>	Enthält einen Wert aus der enumeration enum_spez_type: Version der Spezifikationsdatenbank (QSDOK)
<information_system>	Enthält Angaben zum eingesetzten Informationssystem (KIS).

Dem Attribut //software/vendor/@registration muss die vom IQTIG vergebene Registriernummer für die eingesetzte QS-Software zugewiesen werden.<sup>1</sup> Softwareanbieter können diese Registriernummer über [verfahrenssupport@iqtig.org](mailto:verfahrenssupport@iqtig.org) beantragen.

## Element header/document/software/information\_system

Sammelement für Angaben zum eingesetzten Informationssystem (KIS).

Abbildung 12: Aufbau des Elements *information\_system*Tabelle 44: Kind-Elemente des Elements *information\_system*

Kind-Elemente	Beschreibung
<vendor>	Enthält Informationen über den Softwarehersteller.
<name>	Enthält den Softwarenamen der eingesetzten Software.

### Element header/document/data\_flow

Dieses Element gibt an, für welchen Datenfluss (Datenannahmestelle) dieses Dokument erzeugt wurde.

Tabelle 45: Angabe des betreffenden Datenflusses

Ausprägung	Beschreibung
QS-Laenderbezogen	für eine Datenlieferung, welche über die Landesgeschäftsstelle (LKG) übermittelt werden muss

Über die beiden tabellarisch dargestellten Varianten hinaus existieren im XML-Schema noch weitere Ausprägungen, welche jedoch für den Datenfluss im Rahmen der einrichtungsbezogenen QS-Dokumentation keine Relevanz haben.

### Element header/document/data\_target

Dieses Element gibt an, welches Ziel der Datenfluss hat.

Tabelle 46: Angabe des betreffenden Datenfluss-Ziels

Datenfluss	Ziel
Echtdatenpool	Echtdaten für den Echtbetrieb
Probedatenpool	Echtdaten für vorläufige Auswertungen wie der Sonderexport
Testdatenpool	Testdaten für Testzwecke



#### Achtung Datenverlust

Nur Daten mit der Kennzeichnung „Echtdatenpool“ werden für die Erstellung von Auswertungen (z. B. Rückmeldeberichte) berücksichtigt.

### Element header/provider

Das Element <provider> gibt an, welche Institution dieses Dokument zuletzt bearbeitet hat. Es wird in jeder am Datenfluss beteiligten Instanz durch diese ersetzt und so zur nächsten Instanz geschickt.

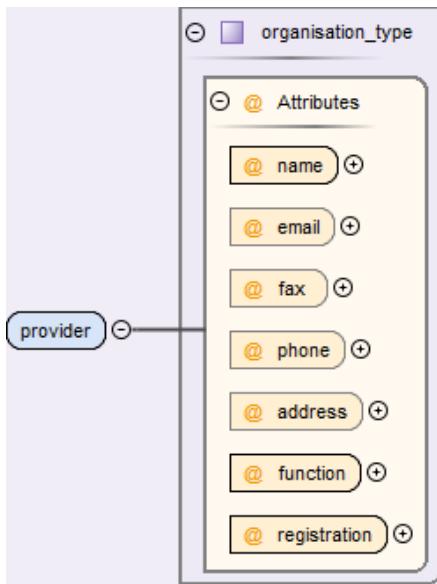


Abbildung 13: Aufbau des Elements provider

Tabelle 47: Attribute des Elements header/provider

Name	Type	Use	Beschreibung
name	xs:string	required	Name der Institution
email	emailAddress_type	optional	E-Mail-Adresse
fax	xs:string	optional	Faxnummer
phone	xs:string	optional	Telefonnummer
address	xs:string	optional	Adresse
function	enum_organization_type	required	Ausprägungen: Bundesauswertestelle, Datenannahmestelle, Softwarehersteller, undefined
registration	registration_type	required / optional	Registrierungsnummer.

### Element header/protocol

Das Element <protocol> nimmt Informationen zu Prüfungen auf, die im Datenfluss durchgeführt wurden. Es ist Teil der Rückprotokollierung. Dieses Element ist nicht optional und soll gemeinsam mit dem Unterelement <status\_document> von Anfang an im Datenfluss vorhanden sein, um nachfolgende im Datenfluss vorgenommene Prüfergebnisse aufzunehmen.

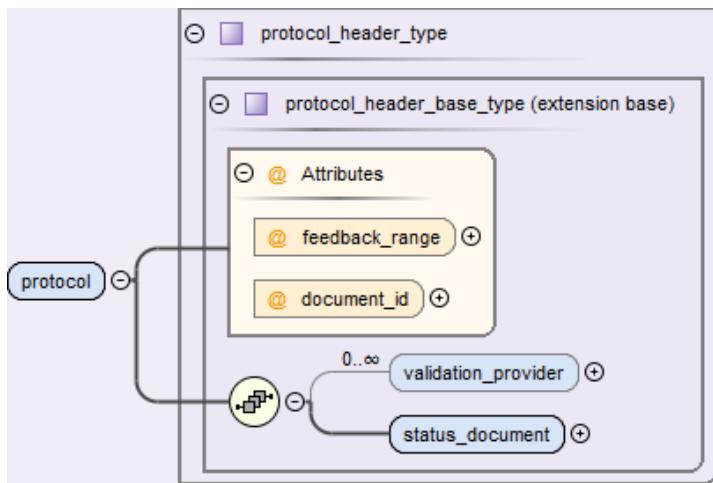


Abbildung 14: Aufbau des Elements header/protocol

Dieses Element hat zusätzlich zu den optionalen Attributen `feedback_range` und `document_id` zwei Kind-Elemente:

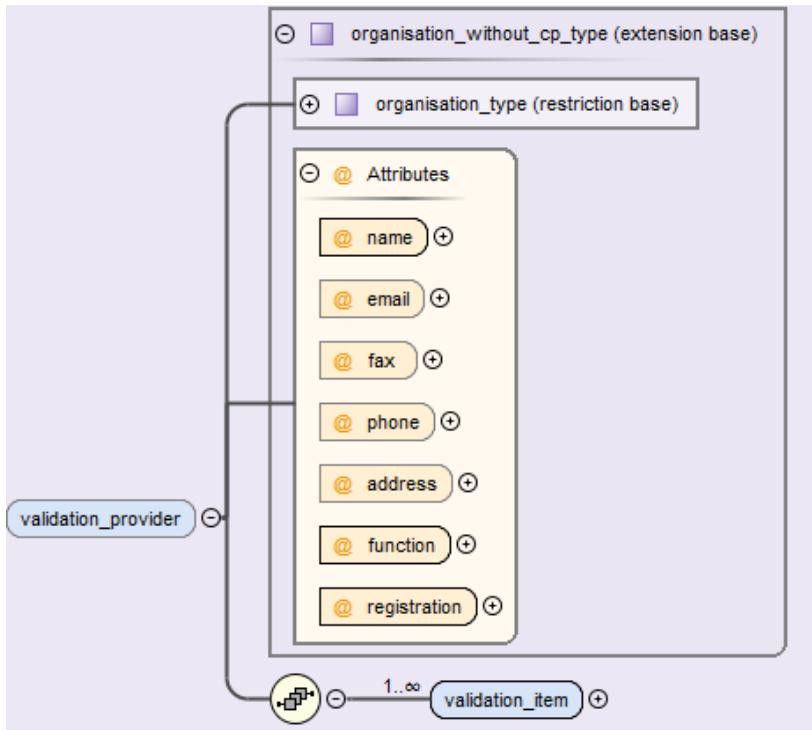
<`validation_provider`> und <`status_document`>

Tabelle 48: Attribute des Elements header/protocol

Name	Type	Use	Beschreibung
<code>feedback_range</code>		optional	Da die Transaktionsprotokolle durch die Empfangsbestätigungen ersetzt wurden, ist nur der Wert „dataflow“ zu verwenden.
<code>document_id</code>		optional	Soweit die GUID des Exportdokuments lesbar ist, muss sie in das Attribut <code>document_id</code> eingetragen werden.

### Element header/protocol/validation\_provider

Hier gibt sich die Stelle zu erkennen, die einen oder mehrere Prüfungsschritte durchgeführt hat. Die Ergebnisse der Prüfung werden in diesem Container abgelegt und werden Teil der Rückprotokollierung.

Abbildung 15: Aufbau und Kind-Elemente des Elements `validation_provider`Tabelle 49: Attribute des Elements `validation_provider`

Name	Type	Use
name	xs:string	required
email	emailAddress_type	optional
fax	xs:string	optional
phone	xs:string	optional
address	xs:string	optional
function	enum_validation_provider_type	required
registration	registration_type	required

### Element header/protocol/validation\_provider/validation\_item

Auf Dokumentenebene sind alle Prüfungen zu dokumentieren. Eine prüfende Einrichtung trägt sich als `<validation_provider>` in die entsprechende Auflistung ein und dokumentiert dann ihre durchgeführten Prüfungen in der Auflistung `<validation_item>`.

Es wird als Ergebnis jeder Prüfung eine der folgenden Aussagen über das geprüfte Objekt getroffen:

- OK (Keine Auffälligkeiten)
- WARNING (Auffälligkeiten, die einer Weiterverarbeitung nicht im Weg stehen)
- ERROR (Auffälligkeiten bzw. Fehler, die eine Weiterverarbeitung des Datensatzes oder des Dokumentes ausschließen)

Das Ergebnis der Prüfung wird in das Attribut @v des Elements <status> im Element <validation\_item> eingetragen.

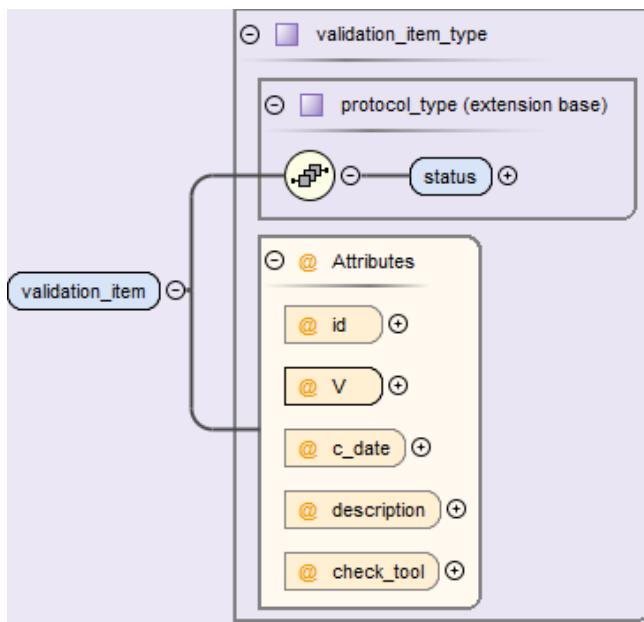


Abbildung 16: Aufbau und Kind-Elemente des Elements validation\_item

Tabelle 50: Attribute des Elements validation\_item

Name	Type	Beschreibung
id	xs:int	Diese ID ist dokumentweit gültig und darf im Header nur einmal vorkommen. Prüfungen auf Datensatzebene (Element <case>), die zu dieser Prüfung gehören, werden über diese ID zugeordnet. Die ID muss nur dann vergeben werden, wenn eine Prüfung auf Fallebene stattfindet.
v	enum_validation_type	Dieser Wert bezeichnet die durchgeführte Prüfung anhand einer Aufzählung, die in sqg_protocol.xsd definiert wird. Gültige Werte sind: Dechiffrierung, LE_Pseudonym, PID_Pseudonym, Protokoll, Schema, Spezifikation, Transaktion, sonstige Prüfung.
c_date	xs:dateTime	Hier kann ein Zeitstempel für die Verarbeitung angegeben werden.
description	xs:string	Prüfungsbeschreibung laut Spalte „Prüfung“ in Sicht vPruefung in QSDOK-Datenbank
check_tool	xs:string	Versionsnummer des Prüftools (z. B. das Datenprüfprogramm)

### Attribut /protocol/validation\_provider/validation\_item/@check\_tool

Hier können beim Einsatz eines Tools für die Prüfung der XML-Dateien der Name und die Versionsnummer des Tools hinterlassen werden (beim Einsatz des Datenprüfprogramms wird die Versionsnummer des XSLT-Skripts eingetragen).

Das Datenprüfprogramm trägt automatisch die Versionsnummer in das <validation\_item>-Element ein. Damit das Protokoll nicht unnötig groß wird, wird die Information über das Tool nur auf Dokumentenebene aufgenommen (header/protocol/validation\_provider/validation\_item/).

### Element header/protocol/status\_document

Hier wird der Gesamtstatus des Dokuments angegeben, das Attribut `v` kann also auf `OK`, `WARNING` oder `ERROR` stehen. Dieser Status kann nur geändert werden, wenn sich der Status des Dokuments verschlechtert oder gleich bleibt. `ERROR` bedeutet, dass das Dokument komplett zurückgewiesen werden muss.

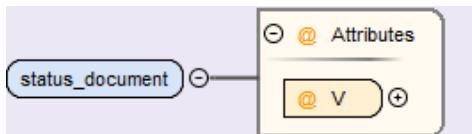


Abbildung 17: Aufbau des Elements `status_document`

Tabelle 51: Attribute des Elements `status_document`

Name	Type	Use	Beschreibung
<code>v</code>	<code>status_type</code>	required	Mögliche Werte: <code>OK/WARNING/ERROR</code>

### Element header/protocol/validation\_provider/validation\_item/status

Das Element gibt an, ob die betroffene Testeinheit ohne Fehler (`OK`), mit Fehlern (`WARNING`) oder mit fatalem Fehler (`ERROR`) abgeschlossen wurde. Der Gesamtstatus des Dokuments entspricht jeweils dem schlechtesten Prüfergebnis. Bei der ersten Prüfung mit dem Ergebnis `ERROR` muss die Weiterverarbeitung abgebrochen werden.

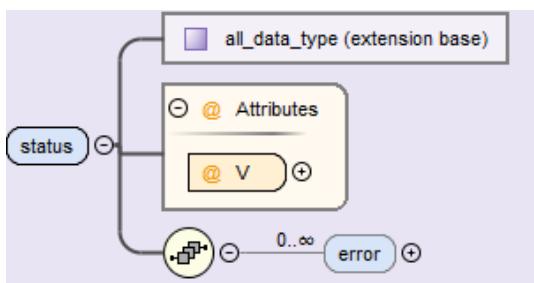


Abbildung 18: Aufbau und Kind-Elemente des Elements `status`

Tabelle 52: Attribut des Elements status

Name	Type	Use	Beschreibung
V	status_type	required	Status einer Prüfung mit folgenden, möglichen Werten: OK, WARNING oder ERROR.

Darüber hinaus gibt es die Möglichkeit, eine beliebige Anzahl von <error>-Elementen mit einer <error\_message> im <status>-Element unterzubringen.

#### Element header/protocol/validation\_provider/validation\_item/status/error

Ein <error>-Element nimmt Fehlerdaten auf. Als einzige verpflichtendes Unterelement gilt das <error\_message>-Element. Die Elemente <rule\_id> und <rule\_type> sind spezifisch für die Anwendung von Plausibilitätsregeln für die Spezifikation:

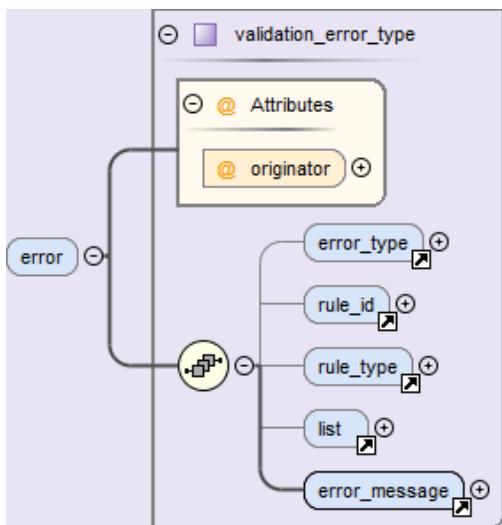


Abbildung 19: Aufbau des Elements error

Tabelle 53: Attribut des Elements error

Name	Type	Use	Beschreibung
originator	enum_organization_type	optional	Mögliche Werte: Bundesauswertestelle, Datenannahmestelle, Leistungserbringer, Softwarehersteller, undefined

Tabelle 54: Kind-Elemente des Elements error

Kind-Elemente	Beschreibung
<rule_id>	Nummer der Regel (idRegeln in Tabelle Regeln) oder

	Nummer der Fehlermeldung aus der Tabelle Fehlermeldung (id-Fehlermeldung)
<rule_type>	Werte H (=hart) oder W (=Warnung bzw. weich)
<liste>	In Abhängigkeit von der Fehlerart entweder Liste von Teildatensätzen oder von Bogenfeldern
<error_message>	Fehlermeldung als Freitext
<error_type>	Hat folgende Ausprägungen: EXPORT = Formatfehler der Exportdatei DOPPELT = bereits vorhandener Datensatz wird erneut übermittelt TDS = Vollständigkeit und Version der Teildatensätze WERT = Wertebereichsverletzung REGEL = Plausibilitätsverletzung KOLLISION = Patientenpseudonym mit unterschiedlichen Alters-/Geschlechtsangaben PID = PID nicht entschlüsselbar (in Kombination mit Dechiffrierung von „validation_item“) QS = QS-Daten nicht entschlüsselbar (in Kombination mit Dechiffrierung von „validation_item“) IST_Statistik = Fehlende/falsche Angaben zur IST-Statistik SOLL_Statistik = Fehlende/falsche Angaben zur SOLL-Statistik

### Element header/encryption

Das Element nimmt Informationen über den Schlüssel auf, mit dem die Daten verschlüsselt worden sind. Das Attribut `id` enthält den Namen des symmetrisch verschlüsselten XML-Knotens. Eine Beispielimplementierung dieser Spezifikation ist ein öffentliches Verschlüsselungsprogramm des IQTIG (XPacker).

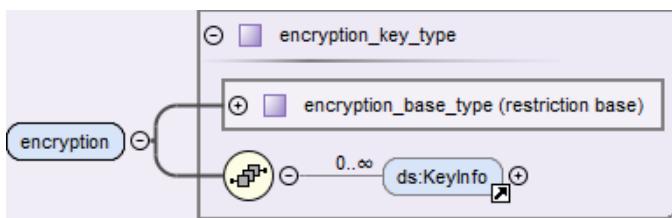


Abbildung 20: Aufbau und Attribute des Elements `encryption(Krankenhaus)`

Das Programm dient zur Ver- und Entschlüsselung einzelner XML-Elemente (Tags) innerhalb einer XML-Datei, basierend auf einem hybriden Verfahren, das aus folgenden Einzelschritten besteht:

- Ein zufälliger symmetrischer Schlüssel wird erzeugt.
- Mit diesem Schlüssel wird ein XML-Element (z. B. `qs_data`) chiffriert.
- Der Schlüssel wird nun mit dem „public Key“ des Empfängers (z. B. LKG) verschlüsselt.
- Der mit dem „public Key“ chiffrierte symmetrische Schlüssel wird dem Empfänger zusammen

mit den verschlüsselten Daten übergeben.

- Der Empfänger dechiffriert den Schlüssel mit seinem „private Key“ und erhält so den symmetrischen Schlüssel.
- Mit diesem symmetrischen Schlüssel dechiffriert der Empfänger die verschlüsselten Daten.

Weitere Informationen sind der Dokumentation des Verschlüsselungsprogramms zu entnehmen.

### 3.3.4 Body-Bereich

Im <body>-Element liegen die eigentlichen QS- und LE-Daten. Der Body-Bereich kann einen oder mehrere <data\_container> enthalten, die einem bestimmten Leistungserbringer zugeordnet sind.

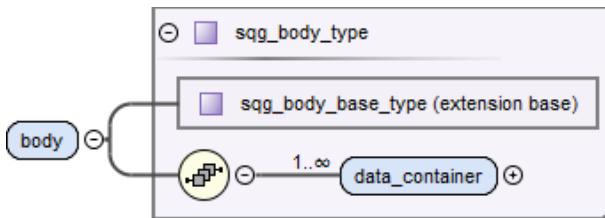


Abbildung 21: Aufbau des Elements body

#### Kind-Element body/data\_container

Ein <data\_container> ist einem bestimmten Leistungserbringer zuzuordnen. In der Regel sollte in einem Dokument nur ein <data\_container> vorhanden sein.

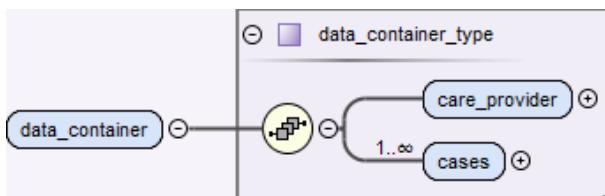


Abbildung 22: Aufbau des Elements body/data\_container

#### Element body/data\_container/care\_provider

Die Zuordnung zu einem Leistungserbringer erfolgt durch das Element care\_provider. Die Inhalte des <care\_provider>-Elements ergeben sich aus Anwendung der Abfrage vExportZieleXml der QSDOK-Datenbank.

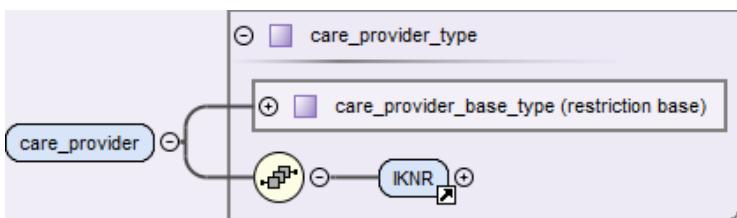


Abbildung 23: Aufbau des Elements care\_provider – Krankenhaus

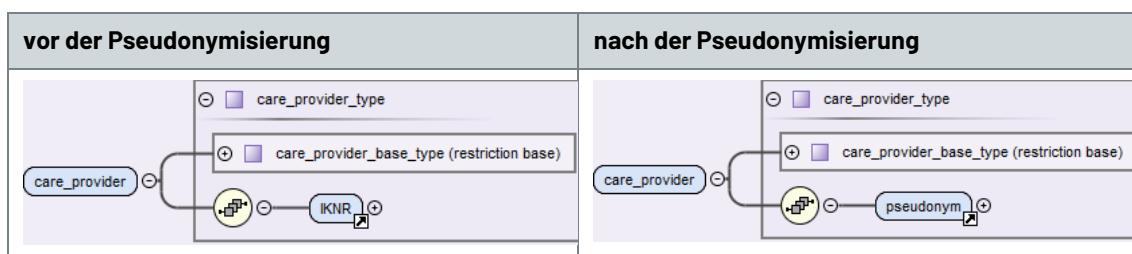
LKG →

Die Kind-Elemente für den Bereich Krankenhaus werden in der folgenden Tabelle beschrieben.

Tabelle 55: Leistungserbringeridentifizierende Daten im vertragsärztlichen und stationären Bereich

Kind-Elemente	Beschreibung
<b>Leistungserbringeridentifizierende Daten im Bereich Krankenhaus</b>	
<IKNR>	Institutionskennzeichen – IKNRKH. Gemäß §293 SGB V wird bei der Datenübermittlung zwischen den gesetzlichen Krankenkassen und den Leistungserbringern ein Institutionskennzeichen (IK) als eindeutige Identifizierung verwendet. Die IK werden durch die „Sammel- und Vergabestelle Institutionskennzeichen (SVI)“ der Arbeitsgemeinschaft Institutionskennzeichen in Sankt Augustin (SVI, Alte Heerstraße 111, 53757 Sankt Augustin) vergeben und gepflegt. Hier ist das bei der Registrierung für die Qualitätsicherung angegebene IK zu verwenden.

Tabelle 56: Element care\_provider in den DAS-Schemata interface\_LQS.xsd und interface\_LQS\_psn.xsd



### Element body/data\_container/cases

Container-Element für eine Liste von gleichartigen Fällen (Vorgängen). „Gleichartig“ meint hier Fälle des gleichen Primärmoduls<sup>45</sup>. Das Element enthält einen oder mehrere Vorgänge<sup>46</sup>.

Für unterschiedliche Module müssen jeweils mehrere <cases> angelegt werden. Die Ausweitung eines <cases>-Elements für Daten eines bestimmten Primärmoduls erfolgt über dessen Attribut module.

<sup>45</sup> In der Spezifikation zur einrichtungsbezogenen QS-Dokumentation kann je LE nur ein stationärer Bogen übermittelt werden.

<sup>46</sup> Fälle und Vorgänge werden hier als Synonyme verwendet.

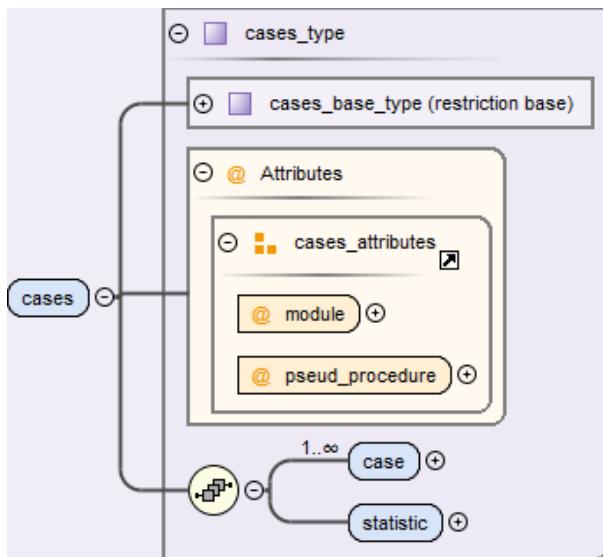


Abbildung 24: Aufbau des Elements cases

Tabelle 57: Attribute des Elements cases

Name	Type	Use	Beschreibung
module	enum_module_type	required	Das Erfassungsmodul
pseud_procedure	enum_procedure_type	required	Zuordnung des Moduls zu einem Pseudonymisierungsverfahren. Im Rahmen der Übermittlung von Daten zur einrichtungsbezogenen QS-Dokumentation ist hier lediglich der Wert „SEP“ zulässig.

### Element body/data\_container/cases/case

Das Element `<case>` entspricht einem Vorgang und enthält genau einen QS-Datensatz eines Moduls.

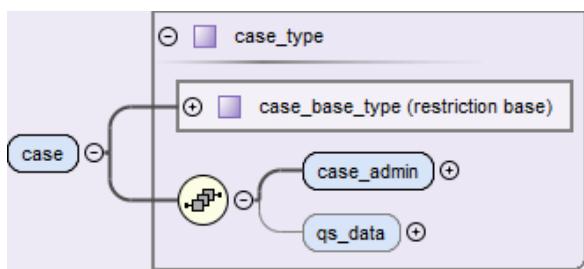


Abbildung 25: Aufbau des Elements case

### Element body/data\_container/cases/case/case\_admin

Das <case\_admin>-Element enthält weitere Elemente, die einen Vorgang identifizieren. Zusätzlich legt das Element fest, was mit dem Vorgang geschehen soll. Auf Vorgangsebene (Datensatzebene) werden von jeder Prüfstelle der Status der Prüfung und ggf. die Fehler in das Element <protocol> eingetragen.

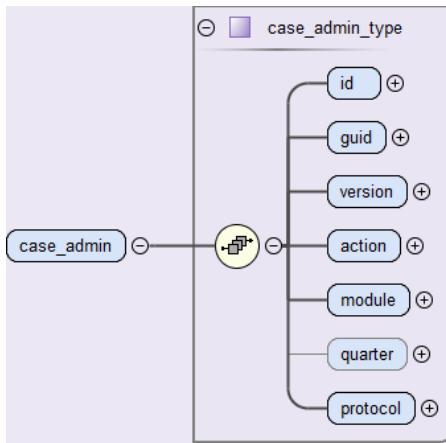


Abbildung 26: Aufbau des Elements case\_admin

Im Folgenden werden die einzelnen Kind-Elemente beschrieben.

Tabelle 58: Kind-Elemente des Elements case\_admin

Kind-Elemente	Beschreibung
<id>	<p>Vorgangsnummer - diese Nummer kennzeichnet jeden dokumentierten Vorgang eines Dokumentationssystems eindeutig und zwar unabhängig vom angewandten Modul.</p> <p>Bei der Umsetzung hat der Softwareanbieter weitgehende Freiheit, vorausgesetzt die modulübergreifende Eindeutigkeit der Vorgangsnummer ist gewährleistet. Da pro Modul jährlich nur eine Lieferung in der einrichtungsbezogenen Dokumentation erfolgt, kann auch der Modulname (SEPE) als Vorgangsnummer verwendet werden.</p> <p>Die Vorgangsnummer darf für die Datenannahmestelle nicht auf Personen zu beziehen sein.</p>
<guid>	36-stelliger pseudozufälliger Globally Unique Identifier (GUID) zur eindeutigen bundesweiten Identifizierung eines Vorganges. Die GUID lässt keine Rückschlüsse auf den Leistungserbringer oder den Patienten zu.
<version>	<p>Enthält eine Versionsnummer des Vorganges. Sie gibt an, die wievielte Version des Vorganges übertragen wird.</p> <p>In der Regel wird die Versionsnummer 1 lauten. D.h., dass der nach dem ersten Dokumentationsabschluss freigegebene Datensatz übertragen wird. Muss ein korrigierter Vorgang erneut eingesandt werden, so muss die Versionsnummer vom dokumentierenden System um 1 erhöht werden. Die neue Version des Vorganges wird bei der Entgegennahme geprüft und überschreibt bei Korrektheit die alte Version des Vorganges.</p>

	Wenn die Datenannahmestelle einen Vorgang mit derselben Versionsnummer ein zweites Mal erhält, so wird dieser zurückgewiesen.
<action>	<p>Definiert die gewünschte Aktion, kann „create“, „update“ und „delete“ sein. „create“ ist beim ersten Export des Vorganges zu verwenden, weitere Exporte des Datensatzes werden mit „update“ geliefert.</p> <p>Aus verschiedenen Gründen ist es im Datenfluss möglich, dass ein update eher die Datenannahmestelle erreicht als ein create. Daher sind create und update gleichzubehandeln. Entscheidend ist case_admin/version/@v.<sup>47</sup></p> <p>Um einen Vorgang zu stornieren, muss &lt;action&gt; auf „delete“ gesetzt werden.</p> <p>Die Datenannahmestelle wird dadurch veranlasst, den betreffenden Vorgang einschließlich aller Vorversionen und Teildatensätze als „storniert“ zu kennzeichnen. Der Stornovorgang wird in der Datenbestätigung protokolliert.</p> <p>Der zu stornierende Datensatz muss ebenfalls eine hochgezählte/fortgeschriebene Versionsnummer enthalten, um die Stornierung unabhängig von der Reihenfolge der Verarbeitung von Datensätzen sicherzustellen. Ein Storno mit einer bereits verwendeten Versionsnummer wird zurückgewiesen.</p> <p>Zur Stornierung eines Vorganges genügt der Export der entsprechenden administrativen Daten &lt;case&gt;/&lt;case_admin&gt;. Die QS-Daten (&lt;qs_data&gt;) des zu stornierenden Vorganges sind nicht erneut zu übermitteln.</p>
<module>	<p>Verfahrensabkürzung. Hier ist zu beachten, dass dieser Wert identisch zu den Attributwerten im Element &lt;cases&gt; und &lt;qs_data&gt; sein muss.</p> <p>Wenn die Datenannahmestelle unterschiedliche Modulbezeichnungen innerhalb eines &lt;cases&gt;-Elements erhält, wird die ganze Datenlieferung zurückgewiesen.</p>
<quarter>	Dieses Element ist optional und dient der Quartalszuordnung bei fallbezogenen QS-Dokumentationen, insbesondere dann, wenn die zuständige Datenannahmestelle keine Einsicht in die QS-Daten haben darf. Im Rahmen der einrichtungsbezogenen QS-Dokumentation kann und sollte dieses Element weggelassen werden.
<protocol>	Protokoll auf Vorgangsebene

### Element case\_Datentyp/case\_admin/protocol

Dieses Element hat eine ähnliche Struktur wie das oben beschriebene Element <protocol> im Header-Bereich.

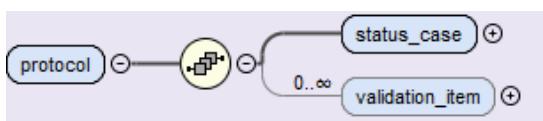


Abbildung 27: Aufbau des Elements case\_admin/protocol

Die Unterschiede sind:

<sup>47</sup> Sollte ein update mit Version 2 bereits in der Datenannahmestelle persistiert sein, dann ist ein später eintretendes create mit case\_admin/version/@v gleich 1 abzulehnen (siehe auch QSDOK.vPruefung mit idPruefung 37 und 48). Käme ein weiteres create mit Version 3, würde keine Ablehnung erfolgen. Ebenso würde danach ein weiteres create mit Version 4 zu keiner Ablehnung führen.

- Während das <protocol>-Element im Header Ergebnisse der Prüfungen, die das Dokument insgesamt betreffen, aufnimmt, nimmt das Protokoll-Element im Body-Bereich die Ergebnisse der Prüfungen auf, die auf Vorgangsebene (Datensatzebene) erfolgen.
- Für erfolgreiche Prüfergebnisse (`status="OK"`) wird nicht explizit das Element <validation\_item> erstellt. Für die Übermittlung des Status des Datensatzes dient weiterhin der implizite Wert des Elements <status\_case> (<code><status\_case v="OK"></code>).
- <status\_case> beinhaltet das schlechteste Ergebnis aller Prüfungen eines Datensatzes.
- Das Protokoll auf Vorgangsebene hat kein Element <validation\_provider> (Prüfstelle). Damit auch auf dieser Ebene die Ergebnisse der durchgeföhrten Prüfungen einer Prüfstelle zugeordnet werden können, müssen alle Ergebnisse einer Prüfung auf Fallebene mit einer gemeinsamen, dokumentweit eindeutigen ID im Attribut ID des Elements <validation\_item> eingetragen werden.

### **Element body/data\_container/cases/statistic**

Das Element <statistic> dient dazu, Statistiken über die Datenlieferung des Absenders und über deren Verarbeitung durch die Bundesauswertungsstelle aufzunehmen.

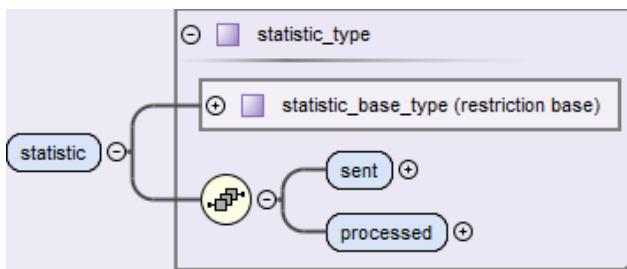


Abbildung 28: Aufbau des Elements statistic

Es ist nach der Prüfung bei der Bundesauswertungsstelle ein Teil des Rückprotokolls und besteht aus ähnlichen Kind-Elementen.

Tabelle 59: Kind-Element des Elements statistic

Kind-Element	Beschreibung
<sent>	Statistik über die von dem Datenlieferanten exportierten Datensätze. Es muss daher vom Datenlieferanten selbst ausgefüllt werden.
<processed>	Hat dieselbe Struktur wie <sent> und enthält das Ergebnis der Verarbeitung durch die Bundesauswertungsstelle.

### **Element statistic/sent**

Das Element nimmt Statistiken über die von dem Datenlieferanten exportierten Datensätze auf und muss vom Datenlieferanten selbst ausgefüllt werden.

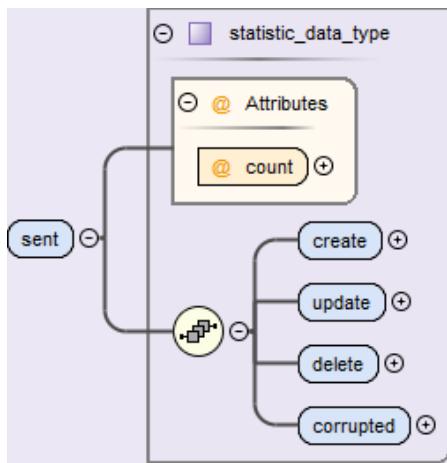


Abbildung 29: Aufbau des Elements sent

Es besteht aus vier Elementen und einem Attribut.

Tabelle 60: Attribut des Elements sent

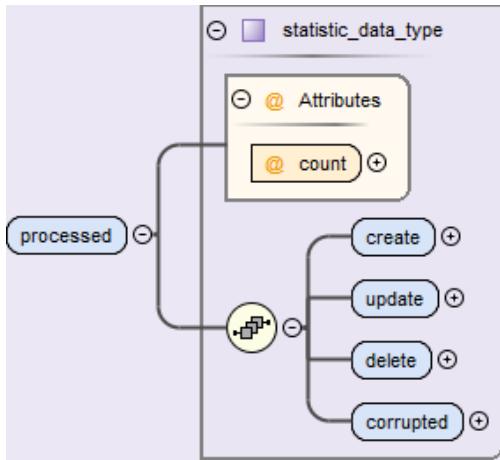
Name	Type	Use	Beschreibung
count	non_negative_integer_type	required	Gesamtzahl von Vorgängen (Ganze Zahl >= 0): Summe von (<create>+<update>+<delete>+<corrupted>)

Tabelle 61: Kind-Elemente des Elements statistic/sent

Kind-Element	Beschreibung
<create>	Anzahl der Datensätze, die neu importiert wurden.
<update>	Anzahl der Datensätze, die aktualisiert wurden (z. B. nach einer Korrektur).
<delete>	Anzahl der Datensätze, die von der Bundesauswertungsstelle storniert wurden.
<corrupted>	Anzahl der Datensätze, die fehlerhaft sind. Der Datenabsender trägt hier „0“ ein.

### Element statistic/processed

Das Element `<processed>` hat dieselbe Struktur wie das Element `<sent>` mit dem Unterschied, dass der Datenempfänger nach der Prüfung der Exportdatei in das Element `<processed>` eintragen soll, wie viele Datensätze er tatsächlich neu importiert, überschrieben und storniert hat und ggf. wie viele Datensätze fehlerhaft sind. Außerdem soll er im Attribut `count` des Elements `<processed>` die Gesamtsumme eintragen.

Abbildung 30: Aufbau des Elements *processed*

<processed> ist vom Datenlieferanten anzulegen und besteht aus vier Elementen und einem Attribut.

Tabelle 62: Attribute des Elements *statistic*

Name	Type	Use	Beschreibung
count	non_negative_integer_type	required	Gesamtzahl von Vorgängen (Datensätzen): Summe von (<create>+<update>+ <delete>+<corrupted>)

Tabelle 63: Kind-Elemente des Elements *statistic/processed*

Kind-Element	Beschreibung
<create>	Anzahl der Datensätze, die der Datenempfänger nach der Prüfung tatsächlich neu importieren konnte.
<update>	Anzahl der Datensätze, die der Datenempfänger nach der Prüfung tatsächlich aktualisieren konnte.
<delete>	Anzahl der Datensätze, die der Datenempfänger tatsächlich stornieren konnte.
<corrupted>	Anzahl der Datensätze, die wegen Mängeln nicht entsprechend der Absicht des Datenlieferanten in den Datenpool übernommen werden konnten.

Alle Werte müssen vom Leistungserbringer mit der Zahl „0“ vorbelegt werden.

Zusammenfassend kann man die Angaben im Element <sent> als Absichtserklärung der Datenlieferung betrachten. Die Angaben im Element <processed> sind als Ergebnis der Verarbeitung in Bezug auf den Zieldatenpool zu verstehen und werden daher nicht vom Leistungserbringer gesetzt.

### Element *qs\_data*

Das <qs\_data>-Element ist ein Container für die QS-Daten, die verfahrensspezifisch sind.

Der Datentyp der konkreten <qs\_data>-Instanz ist jeweils als Attribut "xsi:type" bei der Dokumenterstellung zu definieren.

Folgenden Varianten sind relevant:

```
<qs_data xsi:type="qs_data_sepe_type" module="SEPE">
```

Der Aufbau des XML-Elements <qs\_data> ist variabel und abhängig von der Struktur des jeweiligen Erfassungsmoduls. Die genaue Struktur eines Moduls (nach dem Export) ist von der Spezifikationsdatenbank vorgegeben. Im Allgemeinen gilt, dass jedes Modul immer über einen einzigen Basisdatensatz verfügt.

Die XML-Schemata der einzelnen Module sind in der Schnittstelle „interface\_LE“ zu finden.

## 4 Tools

Das vorliegende Kapitel beschreibt Spezifikationskomponenten, die als Hilfsprogramme die Prozesse in der Qualitätssicherung unterstützen. Zu den Hilfsprogrammen gehören im Rahmen der Basisspezifikation ein Verschlüsselungsmodul, ein Datenprüfprogramm und ein LE-Pseudonymisierungsprogramm. Das LE-Pseudonymisierungsprogramm ist im zugehörigen Anwendерhandbuch und in der dazugehörigen Datenflussspezifikation beschrieben. Das Verschlüsselungsmodul und Datenprüfprogramm werden hingegen in den folgenden Abschnitten erklärt.

### 4.1 Datenprüfprogramm

Das Datenprüfprogramm wurde implementiert, um bereits vor Ort beim Leistungserbringer eine von der Datenannahme-/Bundesauswertungsstelle implementierte Datenprüfung durchzuführen. Das Datenprüfprogramm bezieht die Plausibilitätsregeln direkt aus der Spezifikation und testet Daten vor deren Verschlüsselung.

Das Ausführen vom Datenprüfprogramm setzt Java in der Version 8 oder höher voraus.

#### 4.1.1 Umfang der Prüfungen

Es werden zwei wesentliche Bereiche mit diesem Programm geprüft:

- Schema-Konformität (Struktur)  
Die XML-Datei wird dabei auf Konformität mit dem zugrundeliegenden Schema (XSD) überprüft.
- Regel-Konformität (Inhalte)  
Die XML-Datei wird dabei auf Einhaltung der Regeln (XSLT) überprüft.

Die eigentliche Prüfung erfolgt in der XML-Export-Datei. Deren Struktur ist der entsprechenden Dokumentation bzw. dem gültigen XML-Schema zu entnehmen.

Die inhaltliche Prüfung selbst erfolgt über ein XSLT-Stylesheet und einen XSLT-Prozessor. Das Datenprüfprogramm verwendet die freie Version eines XSLT-Version 2.0-kompatiblen Programms (XSLT2). Die Einbindung von XSLT-Stylesheet und XSLT-Prozessor erfolgt über ein Java-Programm. Prinzipiell kann jeder XSLT2-fähige XSLT-Prozessor für die Prüfung auf dieser Grundlage verwendet werden. Das Datenprüfprogramm stellt eine Referenzimplementierung dar.

#### 4.1.2 Ausgangskontrolle vor Versand

Eine Schemavalidierung der Ausgangsdateien vor dem Versand der XML-Daten wird aus folgenden Gründen als notwendig erachtet:

- Sicherstellung der Datenintegrität nach der Verarbeitung der Daten
- Frühe Feststellung von Fehlerquellen in der eigenen Datenverarbeitung
- Entlastung des nachfolgenden Datenservices von nicht validen Daten
- Vermeidung des Versands von Daten, die gegen den Datenschutz verstößen

Aus diesen Gründen wurde das Datenprüfprogramm um die Möglichkeit erweitert die XML-Daten auch nach der Verschlüsselung der XML-Elemente auf Schemavalidität zu prüfen.

Der Leistungserbringer verwendet abhängig vom Datenfluss die Schemata zur Übergabe an die Datenannahmestelle (LE-DAS) unter:

```
interface_LE_DAS\
```

Die Datenannahmestelle verwendet das folgende Schema zur Übergabe an die Bundeauswertungsstelle (BAS):

```
interface_DAS_BAS\interface_DAS_BAS.xsd
```

Diese Schemata können einfach in den Config-Dateien im Parameter <xsd\_path> entsprechend angegeben werden. Dazu müssen nur mehrere Config-Dateien verwendet werden (je eine pro Schema). Des Weiteren besteht die Möglichkeit, beim Programmstart mit dem Parameter --xsd-path das Schema zu übergeben.

Diese Config-Dateien können entweder in verschiedenen Verzeichnissen abgelegt oder mit dem Parameter -c oder --config beim Start des Datenprüfprogramms über die Konsole angegeben werden. Da in diesem Fall keine inhaltliche Prüfung erfolgen soll (nur XSD, kein XSLT) muss zusätzlich der Parameter --no-spez-val angegeben werden:

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar -c config_schema.xml --no-spez-val
```

Diese notwendige Ausgangskontrolle können die Softwareanbieter/Webportal-Entwickler (bzw. die Leistungserbringer) und die Datenannahmestellen unabhängig von dem Datenprüfprogramm realisieren, indem sie eine Schemavalidierung gegen die o.g. Schemata durchführen.

Für eine Schemavalidierung gibt es zahlreiche Tools und Bibliotheken für alle bekannten Programmiersprachen (<http://www.w3.org/XML/Schema>).

#### 4.1.3 Programmaufruf

Das Datenprüfprogramm erzeugt eine Ausgabe/Output-Datei, die der Eingabe/Input-Datei entspricht, die jedoch um die Ergebnisse der Prüfungen erweitert wird. Die durchgeföhrten Prüfungen entsprechen einer Prüfung auf Dokumenten- und Vorgangsebene (Datensatzebene).

Das Datenprüfprogramm kann mehrere Dateien in einem Aufruf prüfen. Daher gibt es entsprechende Ordner für die Ein- und Ausgabedateien. Sollten diese Ordner nach der letzten Prüfung nicht geleert worden sein, so werden die Dateien des Eingabeordners erneut geprüft und der Ausgabeordner wird parallel mit Datum und Uhrzeit gesichert und ein neues leeres Ausgabeverzeichnis angelegt.

Die Prüfungen umfassen die Schemaprüfung und die Überprüfung der Feldinhalte (auch feldübergreifend).

### **Parameter -c oder --config**

Die Steuerung der Funktionen erfolgt über eine Konfigurationsdatei, deren Dateipfad dem Programm beim Programmstart mit dem Parameter **-c** oder **--config** beim Programmaufruf übergeben werden kann.

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar -c C:/konfiguration/config.xml
```

Wenn keine Konfigurationsdatei übergeben wird, wird die Datei `./config.xml` gesucht und geladen. Wenn diese Datei nicht gefunden wird, wird eine Standard-`config.xml`-Datei im Start-Order angelegt.

### **Parameter --no-spez-val**

Mit diesem Parameter wird das Prüfscript ausgeschaltet.

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar --no-spez-val
```

### **Parameter --no-schema-val**

Mit diesem Parameter wird die Schemaprüfung ausgeschaltet.

```
java -jar datenpruefprogramm-4.2.0-jar-with-dependencies.jar --no-schema-val
```

### **Batch-Dateien**

Beim Datenprüfprogramm werden beispielhafte Batchdateien mitgeliefert:

- `datenpruefprogramm_schema.bat`  
Hier wird eine reine Schemaprüfung (Nur XSD) anhand einer Konfigurationsdatei „`config_schema.xml`“ durchgeführt.
- `datenpruefprogramm_<Datenfluss>_<Verfahrensart>.bat`  
Hier wird sowohl die Schemaprüfung (XSD) als auch die Inhaltliche Prüfung (XSLT) anhand einer Konfigurationsdatei „`config_<Datenfluss>_<Verfahrensart>.xml`“ durchgeführt.

### **Beispiel einer Konfigurationsdatei:**

```
<?xml version= "1.0 " encoding= "UTF-8 " standalone= "no "?>
<config>
  <provider>
    <address>12345 Musterstadt</address>
    <email>max.mustermann@musterfirma.de</email>
    <fax>0123/456798</fax>
    <function>Softwarehersteller</function>
    <name>Mustermann</name>
    <phone>0123/456789</phone>
    <registration>1234567</registration>
```

```

</provider>
<gui>false</gui>
<input_path recursive= "true">input</input_path>
<output_path>output</output_path>
<xsd_path>xsd/interface_LE_WEICH/interface.xsd</xsd_path>
<xsl_path>xsl/2017_DPP_EB_V01.aqxsl</xsl_path>
</config>

```

---

Die Konfigurationsdatei besteht aus den folgenden Bereichen:

### **Provider (Softwarehersteller/Webportal-Entwickler)**

Im Element <provider> werden Daten benötigt, aus denen hervorgeht, wer das Prüfmodul einbindet und ausführt. In der Regel ist dies der Softwarehersteller. Zu beachten ist, dass die Auswahlmöglichkeit im Element <function> auf Softwarehersteller eingeschränkt ist. Die Elemente <fax>, <phone> und <address> sind optional, die anderen sind Pflichtelemente.



#### **Achtung**

In den Elementen <fax>, <phone> und <address> dürfen auf keinen Fall die Angaben des Leistungserbringens eingetragen werden!

---

### **GUI (Konsole)**

Für ein vereinfachtes Debugging gibt es die Möglichkeit, eine Konsole mit detaillierten Programm-ausgaben während der Verarbeitung über das Element <gui> und den Wert true zu öffnen. Der Standard-Wert ist false.

### **Input\_Path (Eingabeverzeichnis) – überschreibbar mit Parameter –input**

Im Element <input\_path> kann der Eingabeordner für die zu überprüfenden Exportdateien festgelegt werden. Das Element ist optional. Ohne diesen Parameter ist der Ordner <arbeitsverzeichnis>\input\ der Standard-Eingabe-Ordner. Es werden alle Dateien mit der Dateiendung .xml verarbeitet. Wenn das Attribut recursive auf true steht, werden auch alle entsprechenden Dateien in Unterordnern berücksichtigt. Die Standard- Einstellung von recursive ist false.

### **Output\_Path (Ausgabeverzeichnis) – überschreibbar mit Parameter –output**

Im Element <output\_path> kann der Ausgabeordner festgelegt werden. Das Element ist optional. Ohne diesen Parameter ist der Standard-Ausgabeordner <arbeitsverzeichnis>\output\. Der Dateiname der Ausgabedatei ist dabei gleich dem der Eingabedatei. Ein ggf. nicht vorhandener Ordner wird angelegt.

### XSD\_Path (Schemaordner) – überschreibbar mit Parameter --xsd-path

Im Element <xsd\_path> wird der Schemapfad gesetzt. Es wird dabei entweder auf ein Verzeichnis gezeigt, in dem genau eine Schemadatei erwartet wird, oder es wird direkt auf eine xsd-Datei gezeigt. Das Element ist optional. Ohne diesen Parameter wird das Schema im Verzeichnis <arbeitsverzeichnis>\xsd\interface\_LE\_WEICH gesucht. Eine Spezifikationskonforme Protokollierung kann vom Datenprüfprogramm sichergestellt werden, wenn die weiche Schemavariante verwendet wird.

Um nach einer Schemavalidierung der XML-Dateien, die Weiterverarbeitung und dementsprechend die spezifikationskonforme Protokollierung auf Datensatzebene weiterhin zu ermöglichen, wurden neben der harten Schemavariante ein weiches Schema für die Schnittstellen LE und DAS eingeführt. Diese weiche Variante wird ausschließlich mit dem Datenprüfprogramm verwendet (Abbildung 31). Eine detaillierte Übersicht über die Anwendung weicher Schemata mit dem Datenprüfprogramm sind im Abschnitt B zu finden.



#### Hinweis

Neben den Leistungserbringern sind nur die LKG und die BAS (DAS) jene Stellen im Datenfluss, welche QS-Daten gemäß den G-BA-RL entschlüsseln dürfen und die weiche Variante benötigen.

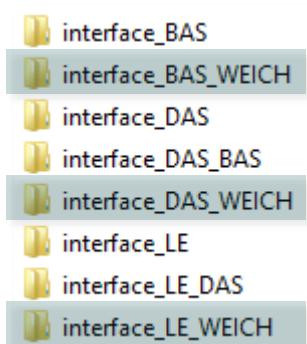


Abbildung 31: Weiche Schemavarianten für das DPP

### XSL\_Path (XSLT-Stylesheet-Ordner) – überschreibbar mit Parameter --xsl-path

Im Element <xsl\_path> kann der Quellordner für die XSLT-Stylesheets festgelegt werden. Das Element ist optional. Ohne diesen Parameter wird im Standard XSL-Ordner <arbeitsverzeichnis>\xsl\ nach den XSLT-Stylesheets gesucht.

Mehrere Konfigurationsdateien können für dasselbe Datenprüfprogramm angelegt werden, um beispielsweise Dateien unterschiedlicher Spezifikationen zu validieren oder die Durchführung einer Eingangs- bzw. einer Ausgangskontrolle jeweils vor der Entschlüsselung und nach der Verschlüsselung zu ermöglichen.

#### 4.1.4 Verzeichnisstruktur

Für das korrekte Funktionieren des Prüfprogramms ist neben den erforderlichen Dateien auch eine korrekte Verzeichnisstruktur notwendig.

In der config.xml wird der <xsl\_path> definiert. Wenn der Parameter <xsl\_path> auf ein Verzeichnis zeigt, muss in diesem Verzeichnis eine Stylesheet-Datei der folgenden Art vorliegen:

```
xsl\<Spezifikationsversion>.aqxsl
```

Hierbei handelt es sich um ein kompiliertes Haupt-XSL-Stylesheet, welches die Prüfung entsprechend der Spezifikation durchführt. Es enthält alle Tests auf Regeln und Wertebereichsverletzungen. Ansonsten kann der Parameter <xsl\_path> auf eine beliebige Stylesheet-Datei verweisen.

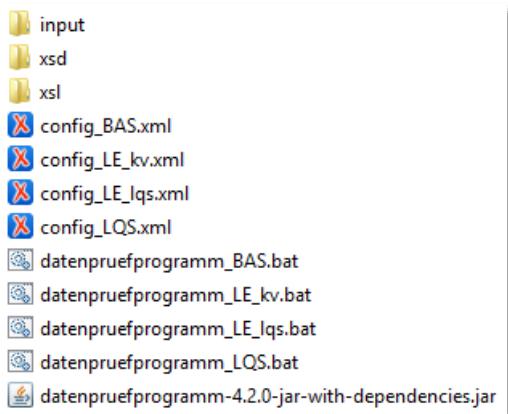


Abbildung 32: Verzeichnisstruktur eines DPP-Ordners

Die Verzeichnisstruktur kann über eine config.xml-Datei modifiziert werden. Ohne diese config.xml wird eine Standardkonfiguration angenommen, die das Prüfprogramm selbst in einer config.xml neu schreibt.

In Abbildung 32 ist eine typische Verzeichnisstruktur mit allen gemäß RL-Vorgaben sinnvollen config- und zugehörigen Batch-Dateien sowie dem Programm abgebildet. Innerhalb dieses Grundordners befinden sich die Verzeichnisse, welche in der Beschreibung der Konfiguration erläutert werden.

#### 4.1.5 Ausgabe

Nach dem Prüflauf wird ein Ordner <output\_path> erzeugt, der die geprüften Dateien und deren Datenflussprotokolle beinhaltet.

##### Geprüfte Dateien

Im Ordner <output\_path>/files liegen nun die geprüften Quell-Dateien, welche um das Ergebnis der Prüfung erweitert worden sind. Jeder Fall wird innerhalb der XML-Datei geprüft und in der XML-Struktur abgelegt. Zudem wird ein neuer Eintrag als validation\_provider erzeugt.

## Protokolle

Im Ordner <output\_path>/protocol liegen die Datenflussprotokolle, die den entsprechenden Dateien im Ordner <output\_path>/files entsprechen, in denen QS-Daten entfernt wurden.

## HTML-Protokolle

Im Ordner <output\_path>/html liegt eine index.html, in der auf vereinfachte Sichten der im Ordner <output\_path>/protocol erstellten Protokolle verwiesen wird.

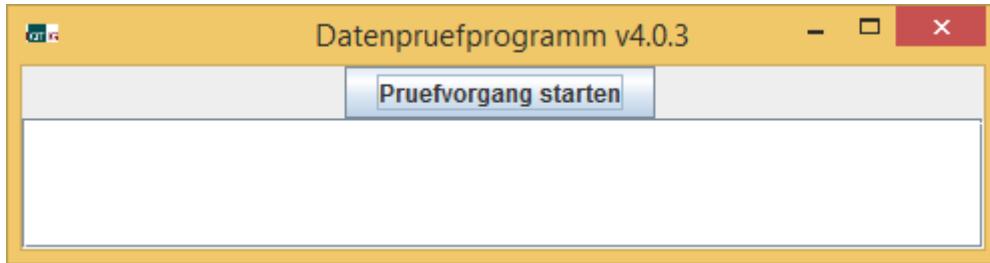
<a href="#">5d29aac7-d480-4377-ad2b-070596b27dda.xml</a>	5d29aac7-d480-4377-ad2b-070596b27dda	OK
<a href="#">65238192-1879-4fa2-84e6-a4323f52522c.xml</a>	65238192-1879-4fa2-84e6-a4323f52522c	OK
<a href="#">8e5e8197-5478-46b5-8fd3-cb527d0141a2.xml</a>	8e5e8197-5478-46b5-8fd3-cb527d0141a2	OK
<a href="#">9afe7686-7f73-43f8-a5ce-27d5139064f4.xml</a>	9afe7686-7f73-43f8-a5ce-27d5139064f4	OK
<a href="#">9d6359cf-2b70-4706-8670-5dd51b67306f.xml</a>	9d6359cf-2b70-4706-8670-5dd51b67306f	WARNING

Abbildung 33: Beispiel für eine Index.html Datei im Ordner <output\_path>/html

### 4.1.6 Grafische Oberfläche

Wird der Parameter GUI in der Konfigurationsdatei auf „true“ gesetzt, wird das Datenprüfprogramm mit einer einfachen grafischen Oberfläche gestartet.

Abbildung 34: Grafische Oberfläche des Datenprüfprogramms



Bei der grafischen Oberfläche muss zum Starten die Schaltfläche „Pruefvorgang starten“ gedrückt werden.

Die grafische Oberfläche zeigt die Ausgabe im Fensterbereich direkt an. Am Inhalt des Ausgabe-Ordners ändert sich nichts; beide Laufvarianten (grafische Ausgabe oder Konsolenausgabe) erzeugen den gleichen Output.

### 4.1.7 Programmierschnittstelle – API

Sämtliche Funktionen des Datenprüfprogramms können über eine Programmierschnittstelle (API) aufgerufen und direkt in einem Java-Umfeld verwendet werden.

## 4.2 Verschlüsselungspaket

Das IQTIG stellt ein Ver- und Entschlüsselungspaket für die Anwendung in der externen Qualitäts-sicherung entsprechend DeQS-RL zur Verfügung. Das Verschlüsselungspaket besteht aus dem TPacker für die Transportverschlüsselung und dem XPack für die XML-Verschlüsselung. Für alle Anwender, die händisch die Verschlüsselung durchführen müssen, stellt der GPacker mit sei-ner grafischen Oberfläche eine interaktive Alternative zur Verwendung der Programme XPack und TPacker dar. Die Verschlüsselungsprogramme werden separat auf der auf der IQTIG-Website veröffentlicht.

## 4.3 LE-Pseudonymisierungsprogramm

Das LE-Pseudonymisierungsprogramm richtet sich ausschließlich an die Datenannahmestellen (DAS), um die leistungserbringeridentifizierenden Daten bei den DeQS-Modulen vor der Weiterlei-tung an das IQTIG nach einem einheitlichen Verfahren zu pseudonymisieren. Alle weiteren Infor-mationen und die Beschreibung zur LE-Pseudonymisierung sind dem ZIP-Archiv und den darin enthaltenen Dokumentationen (Anwenderhandbuch und Datenflussspezifikation) auf der IQTIG-Website zu entnehmen.

# C Anhang

## Glossar

Begriff	Beschreibung
Abrechnungsdaten	Daten, die von Leistungserbringern über von ihnen erbrachte Leistungen zum Zweck der Abrechnung mit den Kostenträgern dokumentiert und übermittelt wurden.
Anonymisierung	Verändern personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können. (§ 3 Abs. 6a BDSG)
Auslösekriterien	Algorithmus zur Auslösung der Dokumentationspflicht (QS-Filter).
Basisspezifikation	Gemeinsame Spezifikation für stationäre und sektorenübergreifende QS-Dokumentationen.
Bundesdatenpool	Zusammenführung aller bundesweit dokumentierten QS-Daten.
Datenannahmestellen	Stellen, an die die Leistungserbringer oder andere Daten liefernde Stellen (z. B. Krankenkassen) die erhobenen Daten übermitteln. Sie prüfen die übermittelten Daten auf Plausibilität, Vollständigkeit und Vollzähligkeit.
Datenbasis	Im Einzelfall festzulegende bzw. festgelegte Gesamtmenge von auszuwertenden bzw. ausgewerteten Daten.
Datenfeld	Kleinste Einheit eines Datensatzes (z. B. Angabe des Geschlechts im Dokumentationsbogen).
Datenfluss	Übermittlung der Daten der QS-Verfahren in einem festgelegten Format und Inhalt, die vom Leistungserbringer über eine Datenannahmestelle, Vertrauensstelle (nur sektorenübergreifende PID-Verfahren) bis zur Datenauswertungsstelle gelangen. Die Datenflüsse sind grundsätzlich in der DeQS-RL des G-BA festgelegt.
Datensatz	Eine zusammenhängende Menge von QS-Daten, die einem Fall (beispielsweise einem Patienten) zugeordnet wird.
Datenvierlagerung	Verfahren zur Überprüfung von QS-Daten einerseits auf Vollzähligkeit, Vollständigkeit und Plausibilität (statistische Basisprüfung), andererseits ihre Übereinstimmung (Konkordanz) mit einer Referenzquelle wie bspw. der Krankenakte (Stichprobenverfahren mit Datenabgleich).
Diagnosis Related Groups (DRG)	Diagnosebezogene Fallgruppen zur pauschalierten Abrechnung stationärer Behandlungsfälle. Wesentliche Grundlage für die Ermittlung einer DRG, das sogenannte Grouping, sind Diagnosen- (ICD) und Prozeduren-Kodes (OPS).
Dokumentation	siehe: QS-Dokumentation

Einrichtung	siehe: Leistungserbringer
Spezifikationsjahr	Das Jahr, in dem die Daten erhoben werden. Hierauf beruhen die Ergebnisse der Indikatoren. Die Kriterien für die Abgrenzung des Spezifikationsjahres sind in der Spezifikation zur Sollstatistik definiert.
Follow-up	auch: Mehrpunktmeßung, Mehrzeitpunktmeßung. Versorgungsleistungen, deren Qualität (z. B. ein Behandlungserfolg) nicht einmalig, sondern zu unterschiedlichen weiteren Zeitpunkten gemessen wird. Die Messungen können in einem bestimmten oder auch in unterschiedlichen Sektoren (vgl. DeQS-RL) stattfinden.
Gemeinsamer Bundesaus- schuss (G-BA)	Oberstes Beschlussgremium der gemeinsamen Selbstverwaltung der Ärzte, Zahnärzte, Psychotherapeuten, Krankenhäuser und Krankenkassen in Deutschland. Er bestimmt in Form von Richtlinien den Leistungskatalog der Gesetzlichen Krankenversicherung (GKV) für etwa 70 Millionen Versicherte und legt damit fest, welche Leistungen der medizinischen Versorgung von der GKV erstattet werden.
Indikator	„Anzeiger“. Auch: Qualitätsindikator. Quantitatives Maß zur Bewertung der Erreichung eines Qualitätsziels einer bestimmten gesundheitlichen Versorgungsleistung.
Kostenträger	Personen und Institutionen, die die Kosten für medizinische Versorgungsleistungen tragen. Im Rahmen der gesetzlich verankerten Qualitätssicherung sind dies die gesetzlichen Krankenkassen.
Leistungserbringer	Personen und Einrichtungen, die medizinische Versorgungsleistungen erbringen bzw. bereitstellen. Der Begriff wird im SGB V auch für Ärzte und ärztliche Einrichtungen sowie für zugelassene Krankenhäuser gem. § 108 SGB V genutzt.
Leistungserbringeridentifizierende Daten (LID)	Daten, die eindeutig einen bestimmten Leistungserbringer identifizieren (z. B. Institutionskennzeichen oder Betriebsstättennummer).
Missing Values	„Fehlende Werte“, z. B. fehlende Antworten und nicht auswertbare Antworten bei der Auswertung eines Fragebogens.
Operationen- und Prozedurenschlüssel (OPS)	Kodierungssystem für medizinische Leistungen bzw. verbindlicher Abrechnungsschlüssel zwischen medizinischen Leistungserbringern und Kostenträgern (Gesetzliche Krankenkassen).
Patientenidentifizierende Daten (PID)	Daten, die eindeutig einen bestimmten Versicherten identifizieren (z. B. Versichertennummer).
Plausibilitätsprüfung	Statistisches Verfahren, mit dem die Dokumentationsdaten auf erlaubte und/oder fehlende Werte, Widerspruchsfreiheit, Werteverteilung und bekannte Korrelationen geprüft werden.
Probabetrieb	Erprobung eines QS-Verfahrens in einer begrenzten Anzahl von Einrichtungen. Ziel ist die Prüfung, ob die für das QS-Verfahren benötigten Daten entsprechend der vom Auftragnehmer vorgesehenen Planungen für die vorgelegten Indikatoren und Instrumente erhebbar und die Ergebnisse umsetzbar sowie für die Durchführung der Qualitätssicherung verwertbar sind.

Pseudocode	Programmcode, der das zugrundeliegende Prinzip eines Algorithmus beschreibt, selbst aber nicht lauffähig ist. Er dient zur Veranschaulichung, unabhängig von der konkret zu verwendenden Programmiersprache.
Pseudonymisierung	Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren (§ 3 Abs. 6a BDSG).
QS-Auslösung	Initiierung einer Dokumentationspflicht zu Zwecken der Qualitätssicherung (QS-Dokumentation). Bei einer Erhebung vorhandener Daten (z. B. Sozialdaten bei den Krankenkassen) analog das Kriterium, das die Lieferung eines bestimmten Datensatzes auslöst.
QS-Daten	Sammelbegriff für alle Daten, die im Zuge eines QS-Verfahrens erhoben und ausgewertet werden.
QS-Dokumentation	Gesonderte Erhebungen der Leistungserbringer zu Diagnose- und Behandlungsdaten der Patienten durch die Leistungserbringer für die Qualitätssicherung.
QS-Filter	Algorithmus, der auf Grundlage festgelegter Kriterien die für die Qualitätssicherung durch die Leistungserbringer zu dokumentierenden Patienten und deren Daten „filtert“. Die Kriterien hierzu werden in einer Spezifikation definiert.
QS-Filter-Software	Implementierung der Spezifikation für den QS-Filter.
QS-Verfahren	hier: QS-Verfahren der gesetzlichen Qualitätssicherung. Medizinischer Themen- oder Behandlungsbereich, der im Rahmen der bundesweiten Qualitätssicherung dokumentationspflichtig ist.
Qualität	Bezogen auf die Gesundheitsversorgung: Grad, in dem versorgungsrelevante Ergebnisse, Prozesse und Strukturen bestimmte, definierte Anforderungen erfüllen.
Qualitätsindikator	siehe: Indikator
Qualitätssicherung	Sammelbegriff für unterschiedliche Ansätze und Maßnahmen zur Sicherstellung festgelegter Qualitätsanforderungen bzw. zur Erreichung bestimmter Qualitätsziele. Hier: Gesetzliche Qualitätssicherung im Gesundheitswesen nach §§ 135-139 SGB V.
Qualitätssicherung, externe stationäre	Einrichtungsübergreifende Qualitätssicherung für medizinisch-pflegerische Leistungen, die ausschließlich im stationären Sektor erbracht werden.
Qualitätssicherungsmaßnahmen	Strukturierte, in Richtlinien geregelte Vorgehensweise, die Leistungserbringer bei der kontinuierlichen Qualitätsverbesserung unterstützt. Auslöser der Qualitätssicherungsmaßnahmen sind rechnerische Auffälligkeiten im Ergebnis eines Qualitätsindikators.
Regelbetrieb	auch: Routinebetrieb oder Echtbetrieb. Verpflichtende und flächendeckende Umsetzung eines QS-Verfahrens.
Risikoadjustierung	Methode zur Ermittlung und Gewichtung wesentlicher Einflussfaktoren (individueller Risiken) auf die Ausprägung einer Messgröße. Eine Risikoadjustierung ist vor allem bei Qualitätsindikatoren erforderlich, die sich auf Behandlungsergebnisse und einen Einrichtungsvergleich beziehen.

	Um hier einen fairen Vergleich zu erhalten, dürfen nur in gleichem Umfang erkrankte Patienten in Einrichtungen mit einer vergleichbaren Klientel miteinander verglichen werden.
Routinedaten	hier: Daten, die wesentlich zur Abwicklung von Geschäfts- und Verwaltungsabläufen erhoben werden (z. B. Abrechnungsdaten, personenbezogene administrative Daten). Abseits des uneinheitlichen Sprachgebrauchs stehen die Sozialdaten bei den Gesetzlichen Krankenkassen (auch: GKV-Routinedaten) im Vordergrund des Interesses, da sie gem. § 299 Abs. 1a SGB V zu Zwecken der Qualitätssicherung verwendet werden dürfen. Diese beinhalten insbesondere die abrechnungsrelevanten Daten für ambulante und stationäre Versorgungsleistungen (§§295 und 301 SGB V), für Arznei-, Heil- und Hilfsmittel (§§ 300 und 302 SGB V) sowie die Versichertenstammdaten (§ 284 SGB V).
Sektor	Institutionell, d.h. durch unterschiedliche Abrechnungsmodalitäten getrennte Bereiche der medizinisch-therapeutischen Versorgung im deutschen Gesundheitswesen (z. B. ambulant/stationär).
Sollstatistik	Aufstellung der im Spezifikationsjahr zu dokumentierenden Leistungen, die vom Krankenhaus durch Konformitätserklärung schriftlich bestätigt wird. Sie gibt die zu erwartende Anzahl von Fällen in den einzelnen QS-Verfahren (Soll) an und bildet zusammen mit der Anzahl der tatsächlich durch die Leistungserbringer gelieferten Daten (Ist) die Grundlage der Vollzähligkeitsprüfung.
Sozialdaten	Einzelangaben über die persönlichen und sachlichen Verhältnisse (personenbezogene Daten), die von den sozialrechtlichen Leistungsträgern zur Erfüllung ihrer gesetzlichen Aufgaben gesammelt und gespeichert werden.
Spezifikation	Datensatzbeschreibung. Festlegung, welche Daten für die Qualitätssicherung erhoben bzw. übermittelt werden müssen, welche Prüfalgorithmen zur Anwendung kommen (z. B. für Plausibilitätsprüfungen) und wie die QS-Auslösung operationalisiert ist. Im Rahmen der Neuentwicklung von QS-Verfahren ist die Spezifikation als das Ergebnis der informationstechnischen Aufbereitung zu betrachten.
Stellungnahmeverfahren	Instrument der Qualitätsförderung. Strukturiertes Verfahren, das Einrichtungen bei der kontinuierlichen Verbesserung von Prozessen und Qualität unterstützt. Auslöser des Stellungnahmeverfahrens sind rechnerische Auffälligkeiten im Ergebnis eines Qualitätsindikators im Rahmen der statistischen Basisprüfung der übermittelten QS-Daten.
Systempflege	Routinemäßige und kontinuierliche Evaluation und Anpassung der Qualitätsindikatoren, der Softwarespezifikation usw.
Verfahren	siehe: QS-Verfahren
Vertrauensstelle	Institution, die im Rahmen der sektorenübergreifenden Qualitätssicherung erhobene patientenidentifizierende Daten pseudonymisiert.
Verweildauer	Dauer des stationären Aufenthalts eines Patienten, Abstand zwischen Aufnahme- und Entlassungsdatum.

Vollständigkeit	Erfassung aller zu einem einzelnen Behandlungsfall erforderlichen Angaben (Daten).
Vollzähligkeit	Erfassung aller dokumentationspflichtigen Behandlungsfälle zu einer bestimmten Versorgungsleistung.
Vollzähligkeitsprüfung	Abgleich der laut QS-Filter-Software zu erwartenden Anzahl von Fällen in einem QS-Verfahren (Soll) mit der Anzahl der tatsächlich durch die Leistungserbringer gelieferten Daten (Ist).
XSLT	Extensible Stylesheet Language Transformations. Programmiersprache zur Transformation von XML-Dokumenten in andere XML-Dokumente oder andere Dokumentformate wie HTML. Im QS-Kontext kann es auch für Datenprüfung und Protokollerstellung verwendet werden.

# Impressum

## HERAUSGEBER

---

IQTIG – Institut für Qualitätssicherung  
und Transparenz im Gesundheitswesen  
Katharina-Heinroth-Ufer 1  
10787 Berlin

Telefon: (030) 58 58 26-0

[info@iqtig.org](mailto:info@iqtig.org)

[iqtig.org](http://iqtig.org)