

**Fachhochschule Potsdam**

University of Applied Sciences

Fachbereich Informationswissenschaften

---

**DIPLOMARBEIT**

## **Archivierung relationaler Datenbanken auf der Grundlage von XML**

Konzeption eines OAIS-konformen Archivierungsmodells und Entwicklung  
eines neuen Ansatzes zur Archivierung von Datenbankabfragen

---

vorgelegt von: **Julia Glöde**  
Studiengang: Dokumentation  
Matrikelnummer: 6763  
E-Mail: [julia.gloede@fh-potsdam.de](mailto:julia.gloede@fh-potsdam.de)

Erstgutachter: Prof. Dr. Rolf Däßler  
Zweitgutachter: Prof. Dr. Günther Neher

Berlin, Juli 2009

## Inhaltsverzeichnis

<b>Abkürzungsverzeichnis.....</b>	<b>IV</b>
<b>Abbildungsverzeichnis.....</b>	<b>VI</b>
<b>Einleitung.....</b>	<b>1</b>
<b><u>I Grundlagen und Stand der Forschung und Entwicklung.....</u></b>	<b><u>2</u></b>
1 Digitale Langzeitarchivierung.....	2
1.1 Bedeutung und Herausforderungen der digitalen Langzeitarchivierung.....	2
1.2 Charakteristika.....	3
1.3 Verfahren.....	3
1.4 Infrastruktur: Das OAIS-Modell.....	5
1.4.1 Datenmodell: Informationspakete.....	6
1.4.1.1 Einlieferungspaket.....	8
1.4.1.2 Archivpaket.....	8
1.4.1.3 Auslieferungspaket.....	8
1.4.2 Funktionsmodell.....	9
1.4.2.1 Übernahme.....	9
1.4.2.2 Aufbewahrung.....	10
1.4.2.3 Datenverwaltung.....	10
1.4.2.4 Systemverwaltung.....	10
1.4.2.5 Planung der Langzeitarchivierung.....	11
1.4.2.6 Bereitstellung.....	11
2 Relationale Datenbanksysteme.....	12
2.1 Bedeutung.....	12
2.2 Komponenten eines Datenbanksystems.....	13
2.2.1 Die Datenbank und das relationale Datenmodell.....	13

2.2.2	Das Datenbankverwaltungssystem.....	15
2.2.3	Die Datenbankkommunikationsschnittstelle am Beispiel von SQL.....	16
3	XML als Archivformat.....	20
4	Stand der Langzeitarchivierung von Datenbanken.....	24
5	Fazit.....	26
<b>II</b>	<b>Das Archivierungskonzept.....</b>	<b>28</b>
1	Das Gesamtkonzept.....	28
1.1	Daten- und Datenflussmodelle.....	28
1.1.1	Einlieferungspaket.....	29
1.1.2	Archivpaket.....	31
1.1.2.1	Gegenüberstellung potentieller Archivformate.....	32
1.1.2.1.1	Datenbankstruktur und Datenbankdaten.....	32
1.1.2.1.2	Datenbankabfragen.....	35
1.1.2.2	Paketaufbau.....	37
1.1.3	Auslieferungspaket.....	39
1.1.3.1	Variante 1: SQL und generische Nutzeroberflächen.....	40
1.1.3.2	Variante 2: XML und XQuery.....	41
1.1.3.3	Variante 3: SQL.....	41
1.2	Prozessmodell.....	42
1.2.1	Vorübernahme.....	42
1.2.2	Übernahme.....	43
1.2.3	Bereitstellung und Nutzung.....	44
1.2.3.1	Variante 1: SQL und generische Nutzeroberflächen.....	45
1.2.3.2	Variante 2: XML und XQuery.....	47
1.2.3.3	Variante 3: SQL.....	48

<b>III Die XML-Dokumenttypdefinition für Datenbankabfragen.....</b>	<b>49</b>
1 Struktur der Dokumenttypdefinition für Datenbankabfragen.....	50
1.1 Strukturelemente.....	51
1.1.1 AUSDRUCK.....	53
1.1.2 TABELLENSPALTE und WERT.....	56
1.1.3 PROJEKTION.....	56
1.1.4 SELEKTION.....	57
1.1.5 VERKNÜPFUNG.....	58
1.1.6 GRUPPIERUNG.....	60
1.1.7 SORTIERUNG.....	60
1.1.8 LIMITIERUNG.....	61
2 Abfragebeispiel.....	63
2.1 Darstellung der Abfrageelemente im Archivformat.....	63
2.1.1 Projektion.....	65
2.1.2 Verknüpfung.....	66
2.1.3 Gruppierung.....	67
2.1.4 Sortierung.....	68
<b>Fazit.....</b>	<b>69</b>
<b>Anhang.....</b>	<b>71</b>
<b>Literaturverzeichnis.....</b>	<b>120</b>
<b>Glossar.....</b>	<b>128</b>
<b>Eidesstattliche Erklärung.....</b>	<b>140</b>

## Abkürzungsverzeichnis

<b>AIP</b>	Archival Information Package
<b>API</b>	Application Programming Interface (Anwendungsprogramm)
<b>ARELDA</b>	Archivierung elektronischer digitaler Daten und Akten
<b>CCSDS</b>	Consultative Committee for Space Data Systems
<b>CMS</b>	Content Management System
<b>CSV</b>	Comma-Separated Values
<b>DBCI</b>	Database Communication Interface (Datenbankkommunikations-schnittstelle)
<b>DBMS</b>	Database Management System (Datenbankverwaltungssystem)
<b>DBML</b>	Database Markup Language
<b>DBS</b>	Database System (Datenbanksystem)
<b>DCL</b>	Data Control Language (Datenkontrollsprache)
<b>DDL</b>	Data Definition Language (Datenbeschreibungssprache)
<b>DIP</b>	Dissemination Information Package
<b>DML</b>	Data Manipulation Language (Datenmanipulationssprache)
<b>DMS</b>	Dokumenten Management System
<b>DTD</b>	Document Type Definition
<b>ERA</b>	Electronic Records Archives
<b>METS</b>	Metadata Encoding and Transmission Standard
<b>NDAD</b>	The National Digital Archive of Datasets
<b>NARA</b>	National Archives and Records Administration
<b>OAIS</b>	Open Archival Information System
<b>PDI</b>	Preservation Description Information
<b>PREMIS</b>	Preservation Metadata: Implementation Strategies
<b>RDBMS</b>	Relational Database Management System (relationales Datenbank-verwaltungssystem)

## Abkürzungsverzeichnis

---

<b>RFC</b>	Request for Comments
<b>RODA</b>	Repository of Authentic Digital Objects
<b>SGML</b>	Standard Generalized Markup Language
<b>SIARD</b>	Software Independent Archiving of Relational Databases
<b>SIP</b>	Submission Information Package
<b>SQL</b>	Standard Query Language
<b>WCMS</b>	Web Content Management System
<b>XML</b>	Extensible Markup Language (erweiterbare Auszeichnungssprache)
<b>XSD</b>	XML Schema Definition
<b>XSL</b>	Extensible Stylesheet Language

## Abbildungsverzeichnis

Abb. I-1: Aufbau eines Informationspaketes.....	7
Abb. I-2: Datenflussmodell des OAIS-Standards .....	11
Abb. I-3: Aufbau eines Datenbanksystems.....	13
Abb. I-4: Elemente des relationalen Datenmodells .....	14
Abb. I-5: Datenmodell des DMS „KnowledgeTree“ (gekürzt).....	15
Abb. I-6: SQL als Kommunikationsschnittstelle.....	18
Abb. I-7: Allgemeiner Aufbau eines XML-Dokumentes.....	20
Abb. I-8: Komponenten von XML.....	21
Abb. I-9: Vor- und Nachteile von XML.....	23
Abb. II-1: Ausgangsdaten und Einlieferungspaket.....	29
Abb. II-2: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbank- daten (gekürzt).....	30
Abb. II-3: Datenbankfeldbeschreibung (gekürzt).....	31
Abb. II-4: Vergleich von Archivformaten für Datenbankstruktur und Datenbank- daten.....	34
Abb. II-5: Vergleich von Archivformaten für Datenbankabfragen.....	36
Abb. II-6: Archivpaket.....	37
Abb. II-7: Umsetzung der Datenbankstruktur mit Hilfe von XML.....	38
Abb. II-8: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten (gekürzt). .	38
Abb. II-9: Auslieferungspaket (Variante 1).....	40
Abb. II-10: Auslieferungspaket (Variante 2).....	41
Abb. II-11: Auslieferungspaket (Variante 3).....	41
Abb. II-12: Prozessmodell Vorübernahme.....	42
Abb. II-13: Prozessmodell Übernahme.....	43
Abb. II-14: Prozessmodell Bereitstellung (Variante 1).....	45

## Abbildungsverzeichnis

---

Abb. II-15: Prozessmodell Bereitstellung (Variante 2).....	47
Abb. II-16: Prozessmodell Bereitstellung (Variante 3).....	48
Abb. III-1: SQLite-Syntaxdiagramm des SELECT-Befehls.....	50
Abb. III-2: Strukturbaum der Dokumenttypdefinition für Datenbankabfragen.....	52
Abb. III-3: Struktur und Eigenschaften des Elements AUSDRUCK.....	54
Abb. III-4: Attribute des Elements AUSDRUCK.....	54
Abb. III-5: Strukturbaum und XML-Code der Beispieloperation.....	55
Abb. III-6: Attribute des Elements TABELLENSPALTE.....	56
Abb. III-7: Struktur, Eigenschaften und Attribut des Elements PROJEKTION.....	57
Abb. III-8: Struktur, Eigenschaften und Attribute des Elements SELEKTION.....	58
Abb. III-9: Struktur und Eigenschaften des Elements VERKNÜPFUNG.....	59
Abb. III-10: Ergänzendes Attribut des Elements TABELLENSPALTE.....	60
Abb. III-11: Struktur und Eigenschaften des Elements GRUPPIERUNG.....	60
Abb. III-12: Struktur, Eigenschaften und Attribut des Elements SORTIERUNG.....	61
Abb. III-13: Struktur und Eigenschaften des Elements LIMITIERUNG.....	62
Abb. III-14: Attribut des Elements DATENSATZNUMMER.....	62
Abb. III-15: XML-Strukturbaum des Abfragebeispiels.....	64
Abb. III-16: Abfragebeispiel: Strukturbaum und XML-Code der Strukturelemente PROJEKTION.....	66
Abb. III-17: Abfragebeispiel: Strukturbaum und XML-Code des Strukturelements VERKNÜPFUNG.....	67
Abb. III-18: Abfragebeispiel: Strukturbaum und XML-Code des Strukturelements GRUPPIERUNG.....	68
Abb. III-19: Abfragebeispiel: Strukturbaum und XML-Code des Strukturelements SORTIERUNG.....	68

### Einleitung

Datenbanken bilden das Rückgrat der modernen Informations- und Wissensgesellschaft. Ein hoher Prozentsatz der zunehmend digital produzierten Daten wird lokal oder online in Datenbanken verwaltet.

Die verwalteten Daten müssen als wertvolle Ressourcen und bedeutender Bestandteil des kulturellen Erbes langfristig in einem digitalen Archiv gesichert werden. Die langfristige Sicherung wird nicht durch die Erstellung eines einfachen Backups der verwaltenden Datenbank gewährleistet. Die Datenbank muss stattdessen anhand archivischer Kriterien und Bestandserhaltungskonzepte bewahrt werden.

Bisherige Verfahren im Bereich der Datenbankarchivierung gewährleisten in der Regel nur die Sicherung der Datenbankdaten. Die Datenbankfunktionalität in Form der Benutzeroberflächen oder Datenbankabfragen wird nicht berücksichtigt.

In dieser Arbeit wird ein Modell vorgestellt, das auf den bisherigen Konzepten aufbaut. Es beinhaltet Prozesse, mit deren Hilfe neben den Datenbankdaten auch die Datenbankfunktionalität in Form der Abfragen gesichert werden kann.

Als Ergebnisse dieser Arbeit sind:

- ein Konzept für die Archivierung der wesentlichen Komponenten relationaler Datenbanken,
- eine Dokumenttypdefinition für die Speicherung von Datenbankabfragen und
- eine komplette Beschreibung des Datenbankarchivformats zu nennen.

Die Arbeit gliedert sich insgesamt in drei Hauptteile. Teil I befasst sich mit den theoretischen Grundlagen und beleuchtet den aktuellen Stand der Forschung und Entwicklung im Bereich der Datenbankarchivierung. Im darauf folgenden Teil II wird das gesamte Archivierungskonzept vorgestellt. Das Einlieferungs-, Archiv- und Auslieferungsformat wird beschrieben, und die ablaufenden Prozesse werden erläutert. Im Teil III erfolgt abschließend eine detaillierte Beschreibung der XML-Dokumenttypdefinition für Datenbankabfragen.

## I **Grundlagen und Stand der Forschung und Entwicklung**

### 1 **Digitale Langzeitarchivierung**

Unter digitaler Langzeitarchivierung werden alle Maßnahmen zusammengefasst, deren Zweck darin besteht, digitale Objekte für die Nachwelt, also über einen langen Zeitraum hinweg, dauerhaft zu erhalten<sup>1</sup>.

In den folgenden Abschnitten wird auf Herausforderungen, Charakteristika und Verfahren der digitalen Langzeitarchivierung eingegangen. Anschließend werden das Daten- und Funktionsmodell des OAIS-Standards genauer beschrieben.

#### 1.1 **Bedeutung und Herausforderungen der digitalen Langzeitarchivierung**

In der heutigen Zeit wird die Mehrheit der Informationen in den verschiedensten digitalen Formen produziert. Einen besonders großen Anteil haben zunehmend originär digital entstandene Informationen, zu denen kein analoges Äquivalent mehr vorliegt. Sie werden täglich in Behörden, Institutionen und Unternehmen in Form von Publikationen, Geschäfts- und Verwaltungsunterlagen, wissenschaftlichen Daten und Forschungsergebnissen produziert. Digitale Informationen sind wichtige Güter der Informations- und Wissensgesellschaft und zunehmend relevant als Bestandteil des kulturellen Erbes. Sie gelten als wertvolle kulturelle, wissenschaftliche und wirtschaftliche Ressourcen, die für die Nachwelt bewahrt werden müssen. Die digitale Langzeitarchivierung ist Voraussetzung für die Konkurrenzfähigkeit von Bildung, Wissenschaft und Forschung.

Eine weitere Besonderheit der digitalen Gesellschaft ist die starke Heterogenität und Vergänglichkeit der digitalen Objekte. Digitale Informationen in Form von Bits sind eng an Hard- und Software gebunden. Nur die entsprechende technische Nutzungsumgebung ist in der Lage, diese Bits zu interpretieren und verständlich wiederzugeben. Aufgrund der rasanten technischen Entwicklung von Datenformaten, Datenträgern und Softwaresystemen kommt es zu einem schnellen Medien- und System-

---

<sup>1</sup> vgl. dazu nestor - Kompetenznetzwerk Langzeitarchivierung: nestor Glossar : Langzeitarchivierung, 20.02.2004, URL: <http://www.langzeitarchivierung.de/index.phpmodule=Encyclopedia&func=displayterm&id=38&vid=1>

wandel. Dieser Wandel führt zu einer Alterung der für den Zugriff unentbehrlichen Technik und zum Verlust der für die Gesellschaft relevanten digitalen Informationen.

Zusammenfassend ist festzustellen, dass die wesentliche Herausforderung für die digitale Langzeitarchivierung nicht allein in der Datensicherung, sondern auch in der Gewährleistung des zukünftigen Zugriffs auf die digitalen Objekte besteht.

## 1.2 Charakteristika

Basierend auf den gegebenen Bedingungen bedeutet Langzeitarchivierung digitaler Objekte nicht „die Abgabe einer Garantieerklärung über fünf oder fünfzig Jahre, sondern die verantwortliche Entwicklung von Strategien, die den beständigen, vom Informationsmarkt verursachten Wandel bewältigen können.“<sup>2</sup>. Der Ausdruck „Langzeit“ beschreibt einen nicht näher fixierten Zeitraum, der sich dadurch auszeichnet, dass wesentliche, nicht vorhersehbare technologische und soziokulturelle Veränderungen auftreten, die eine vollständige Umwälzung sowohl der Gestalt als auch der Nutzungssituation der digitalen Ressourcen zur Folge haben<sup>3</sup>.

Der Ausdruck „Archivierung“ impliziert zudem, dass es sich nicht um die reine Abspeicherung digitaler Informationen zum Zwecke der Substanzerhaltung handelt. Die Erhaltung der Langzeitverfügbarkeit der digitalen Objekte ist als bedeutende Teilkomponente der digitalen Langzeitarchivierung zu beachten.

Grundsätzlich ist eine Bewertung der zu archivierenden digitalen Informationen im Hinblick auf den Inhalt und die Kernfunktionalitäten erforderlich. Entscheidend für die spätere Nutzung ist, dass der originale Kontext zukünftig rekonstruiert werden kann.

## 1.3 Verfahren

Um den genannten Herausforderungen im Bereich der Langzeitverfügbarkeit zu begegnen werden verschiedene Verfahren angewandt. Grundlegend kann auf eine

---

<sup>2</sup> Schwens; Liegmann: Langzeitarchivierung digitaler Ressourcen, 2004, S. 567

<sup>3</sup> vgl. dazu Schwens; Liegmann: Langzeitarchivierung digitaler Ressourcen, 2004, S. 567

Trennung der heterogenen Informationen<sup>4</sup> von ihren originalen Trägern und oft proprietären Systemen sowie auf eine Überführung in ein homogenes Speichersystem nicht verzichtet werden. Durch diese Maßnahmen wird die Unabhängigkeit von spezifischen Systemen und deren Wandel erreicht. Möglich wird dies durch Archivierungsstrategien wie die Migration und die Emulation.

Bei der Migration wird das digitale Objekt nicht in seiner originalen Form bewahrt, sondern so verändert, dass die Inhalte auf aktuellen Systemen angezeigt und benutzt werden können. Diese Archivierungsstrategie eignet sich nicht für komplexe digitale Objekte, da die Gefahr des Informationsverlustes bei der Veränderung der Objekte zu hoch ist. Eine sehr häufig angewandte Migrationsart ist die Formatkonversion. Sie wird notwendig sobald die Gefahr besteht, dass digitale Objekte aufgrund ihres Formates nicht mehr benutzt werden können. Die Formatkonversion findet vor allem auch im Rahmen der Normalisierung<sup>5</sup> in einem digitalen Archiv Anwendung. Bei der Konversion ist der Einsatz nicht proprietärer, offener und gut dokumentierter Datenformate und Standards von wesentlicher Bedeutung. Sie gelten als relativ langlebig und durch ihren öffentlich bekannten Aufbau ist die Wahrscheinlichkeit ihrer zukünftigen Lesbarkeit und Systemkompatibilität relativ hoch.

Die Emulation ist eine Strategie, die bei komplexen digitalen Objekten angewandt werden kann. Bei diesem Verfahren wird die interpretierende, originale Nutzungsumgebung nachgebildet. Das digitale Objekt bleibt unverändert. Da für jede neue Plattform meist auch neue Emulatoren entwickelt werden müssen, ist das Emulationsverfahren, vor allem bei Hardware-Generationswechseln, mit einem sehr hohen technischen Aufwand verbunden.

Strategien, die bei der digitalen Langzeitarchivierung im Rahmen der Substanzerhaltung der digitalen Objekte angewandt werden, sind unter anderem das Refreshing<sup>5</sup> und die Replikation<sup>5</sup>.

Zur Beschreibung, Such- und Auffindbarkeit sowie zur Wahrung der Integrität<sup>5</sup> und Authentizität<sup>5</sup> ist eine umfassende und möglichst standardisierte Erfassung technischer und nicht technischer Metadaten entscheidend. Sie dienen als Grundlage für die beschriebenen Archivierungsstrategien und die spätere Bereitstellung der digita-

---

<sup>4</sup> Heterogenität bezogen auf ihre Quellen und Datenträger.

<sup>5</sup> s. dazu Glossar

len Objekte. Metadaten sollten soweit wie möglich automatisiert erhoben, verwaltet und ausgewertet werden.

#### 1.4 Infrastruktur: Das OAIS-Modell

Die Infrastruktur eines digitalen Archivs wird in Form eines Referenzmodells durch den OAIS-Standard beschrieben. Das als ISO 14721 verabschiedete „*Reference Model for an Archival Information System (OAIS)*“ des CCDSD (*Consultative Committee for Space Data Systems*) ist seit 2002 Bestandteil des ISO-Normenwerkes. Unter Berücksichtigung der Besonderheiten digitaler Objekte bietet es eine allgemeine Sicht auf die Kernfunktionen eines digitalen Archivs und benennt eindeutig Funktionseinheiten, Schnittstellen und Typen von Informationsobjekten. OAIS versteht sich als ein offener Standard und stellt einen hohen Anspruch an die Allgemeingültigkeit. Es existieren keinerlei Beschränkungen auf bestimmte Datentypen, Datenformate oder technische Systemarchitekturen. Außerdem ist das Modell auf Zukunftsfähigkeit und Nachhaltigkeit ausgerichtet. Die Verwendung der aktuell entwickelten Lösungen muss auch in Zukunft gewährleistet sein. Darüber hinaus müssen die Lösungen den unterschiedlichen Nutzergruppen und ihren Bedürfnissen entsprechen.

Das Modell wurde 2002 erstmals in der Niederländischen Nationalbibliothek in Den Haag angewandt. Weitere Anwendungsbeispiele lassen sich im nationalen Datenarchiv Großbritanniens (NDAD)<sup>6</sup>, im Projekt PANDORA<sup>7</sup> des australischen Nationalarchivs und im ERA-System<sup>8</sup> des amerikanischen Nationalarchivs (NARA) finden.

In den folgenden Abschnitten wird detailliert auf die erwähnten Funktionseinheiten und Typen von Informationsobjekten des OAIS-Modells eingegangen.

---

<sup>6</sup> NDAD ist die Abkürzung für „National Digital Archive of Datasets“. Für weitere Informationen s. The National Archives: NDAD : The National Digital Archive of Datasets, 2009, URL: <http://www.ndad.nationalarchives.gov.uk/>

<sup>7</sup> Für weitere Informationen s. National Library of Australia: PANDORA Archive, 16.05.2008, URL: <http://pandora.nla.gov.au/overview.html>

<sup>8</sup> ERA ist die Abkürzung für „Electronic Records Archives“. Für weitere Informationen s. NARA: The National Archives Electronic Records Archives (ERA), 30.06.2009, URL: <http://www.archives.gov/era/>

### 1.4.1 Datenmodell: Informationspakete

Das Datenmodell des OAIS-Standards basiert auf dem Konzept von Informationspaketen. Sie bilden die Informationsobjekte, also die logisch abgegrenzten Informations- und Verwaltungseinheiten des digitalen Archivs. Ein Informationspaket ist eine konzeptionelle Einheit, die aus dem eigentlichen Inhalt und den entsprechenden Metadaten besteht.

Die Inhaltsinformationen bilden das eigentliche Zielobjekt der Archivierung. Sie bestehen aus dem digitalen Datenobjekt und den mit ihnen verbundenen Repräsentationsinformationen. Das Datenobjekt liegt in Form einer Bitsequenz vor. Die Repräsentationsinformationen sind für die Interpretation dieser Sequenz notwendig<sup>9</sup>.

Das Informationspaket des OAIS-Standards verwendet drei Arten von Metadaten:

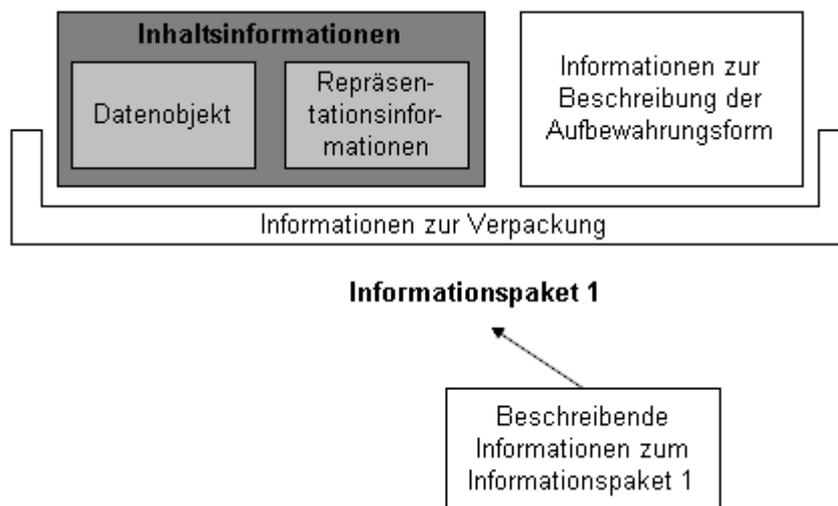
1. Informationen zur Beschreibung der Aufbewahrungsform enthalten Angaben zum Entstehungs- und Entwicklungsprozess (Provenienz<sup>10</sup>). Darüber hinaus beschreiben sie die benötigten Verfahren und technischen Voraussetzungen für die Sicherung (Informationen zur Stabilisierung<sup>10</sup>), eindeutige Identifikation (Informationen zu Beziehungen<sup>10</sup>), Kontexteinordnung (Kontextinformationen<sup>10</sup>) und die zukünftige Nutzbarmachung.
2. Informationen zur Verpackung werden verwendet, um die einzelnen Paketkomponenten zu verbinden und zu identifizieren.
3. Die beschreibenden Informationen dienen hauptsächlich der Auffindbarkeit des Informationspaketes und bilden die Grundlage für die spätere Recherche im Archiv.

Der grundlegende Aufbau eines Informationspaketes und die Beziehungen der einzelnen Paketbestandteile zueinander sind in der folgenden Abbildung I-1 schematisch dargestellt.

---

<sup>9</sup> Ein Beispiel für Repräsentationsinformationen ist die ASCII-Definition, die beschreibt, wie eine Bitsequenz als Symbol abgebildet wird. (vgl. dazu CCSDS: Reference Model for an Open Archival Information System (OAIS), Januar 2002, S. 1-13, URL: <http://public.ccsds.org/publications/archive/650x0b1.pdf>)

<sup>10</sup> s. dazu Glossar



**Abb. I-1:** Aufbau eines Informationspaketes

Die oben beschriebenen Arten von Metadaten können in einer Datei wie in einem Container zusammengefasst werden, in so genannten Containerformaten. Gängige Standards sind z. B. METS<sup>11</sup> (*Metadata Encoding and Transmission Standard*) und PREMIS<sup>11</sup> (*Preservation Metadata: Implementation Strategies*).

Die Metadaten werden gemeinsam mit den Inhaltsinformationen in einer einzelnen Datei gespeichert als auch separat von diesen in einem Datenbanksystem abgelegt.

Im OAIS-Modell werden folgende drei Arten von Informationspaketen unterschieden:

- das Einlieferungspaket (*Submission Information Package (SIP)*),
- das Archivpaket (*Archival Information Package (AIP)*) und
- das Auslieferungspaket (*Dissemination Information Package (DIP)*).

Diese Informationspakete stehen miteinander in Verbindung und sind abhängig von ihren Funktionen und den daraus resultierenden Anforderungen unterschiedlich aufgebaut.

---

<sup>11</sup> s. dazu Glossar

#### **1.4.1.1 Einlieferungspaket**

Das Einlieferungspaket beinhaltet die vom Produzenten übermittelten und vom Archiv übernommenen digitalen Daten. Neben den Inhaltsinformationen enthält das Paket Metadaten. Die Metadaten bestehen vor allem aus beschreibenden Informationen, die für die Aufnahme in das Archiv und die zukünftige Interpretierbarkeit notwendig sind.

#### **1.4.1.2 Archivpaket**

Das Archivpaket wird aus dem Einlieferungspaket generiert. Während dieses Prozesses erfolgen die Anreicherung mit weiteren Metadaten und mögliche Transformationen von Inhalt und Form. Das Archivpaket stellt im Kern die Form dar, in der die digitalen Informationen langfristig aufbewahrt werden.

Die Informationen zur Verpackung und die beschreibenden Informationen orientieren sich an festgelegten Standards und müssen, wie auch die Informationen zur Beschreibung der Aufbewahrungsform, vollständig und so ausführlich wie möglich vorliegen.

#### **1.4.1.3 Auslieferungspaket**

Das Auslieferungspaket stellt die Form da, in der das Archivpaket zugänglich gemacht wird. Die Umwandlung vom Archiv- zum Auslieferungspaket zeichnet sich mehrheitlich durch eine Formatänderung aus. Zu berücksichtigen ist außerdem die Bereitstellung der digitalen Informationen in einer für den Benutzer besonders komfortablen Form. Das Auslieferungspaket muss gezielt für unterschiedliche Nutzergruppen angepasst und anpassbar gehalten werden.

Der Aufbau des Auslieferungspaketes ist unterschiedlich. Das betrifft vor allem die Informationen zur Verpackung. Die Informationen zur Beschreibung der Aufbewahrungsform und die beschreibenden Informationen müssen nicht vollständig vorliegen.

## 1.4.2 Funktionsmodell

Das OAIS-Modell orientiert sich bei seinen Prozessen an den klassisch archivischen Arbeitsfeldern Erfassen, Bewerten, Aussondern, Übernehmen, Erschließen, Erhalten und Bereitstellen.

Im Bereich der Archivierungsstrategien wird die Migration in ihren verschiedenen Formen vorrangig eingesetzt. Das Konzept ist allerdings offen für andere Archivierungsstrategien.

Grundlegend unterscheidet das OAIS-Modell folgende sechs Funktionseinheiten, deren wesentliche Aufgaben in den folgenden Abschnitten kurz dargestellt werden<sup>12</sup>:

- Übernahme (*Ingest*),
- Aufbewahrung (*Archival Storage*),
- Datenverwaltung (*Data Management*),
- Systemverwaltung (*Administration*),
- Planung der Langzeitarchivierung (*Preservation Planning*) und
- Bereitstellung (*Access*).

### 1.4.2.1 Übernahme

Bei der Übernahme geht es darum, das Einlieferungspaket des Produzenten anzunehmen und für die Aufbewahrung und Verwaltung im Archiv vorzubereiten. Der Prozess der Übernahme beinhaltet Funktionen wie die Qualitätssicherung und die Erstellung der beschreibenden Informationen für die Datenverwaltung. Darüber hinaus wird die Aktualisierung des Archivspeichers und der Datenverwaltung koordiniert. Am Ende der Übernahme erfolgt die Generierung des Archivpaketes.

Abhängig von den Übernahmeanforderungen, die zwischen dem Produzenten und dem Archiv vereinbart wurden, ist unter Umständen eine Anpassung der zu archivierenden Daten notwendig. Prozesse, die in diesem Rahmen Anwendung finden, sind Datenexport, Formatkonversion und Metadatenerfassung. Die hier genannten vorbereitenden Arbeiten sind kein fester Bestandteil des OAIS-Modells und werden im Rahmen dieser Arbeit als Vorübernahme berücksichtigt.

---

<sup>12</sup> Eine detaillierte Darstellung und Beschreibung der einzelnen Funktionseinheiten erfolgt in: CCSDS: Reference Model for an Open Archival Information System (OAIS), Januar 2002.

#### **1.4.2.2 Aufbewahrung**

Die Funktionseinheit Aufbewahrung umfasst die Übernahme und Einlagerung des Archivpaketes.

Im Rahmen der Aufbewahrung wird die Auffindbarkeit des Archivpaketes geprüft. Das Speichersystem wird regelmäßig gewartet sowie eine Speicherhierarchie aufgebaut und verwaltet. Zusätzlich wird durch Prüfungen sichergestellt, dass die Archivpakete und Datenträger vollständig und lesbar vorliegen. Gegebenenfalls werden Archivierungsstrategien durchgeführt, um beispielsweise Datenträger zu erneuern.

Als Reaktion auf eine Nutzeranfrage erfolgt die Übergabe des Archivpaketes an die Bereitstellungseinheit.

#### **1.4.2.3 Datenverwaltung**

Innerhalb der Komponente Datenverwaltung werden Prozesse zur Verwaltung der beschreibenden Informationen und der administrativen Daten für die Archivverwaltung durchgeführt. Zudem werden auf Grundlage von Nutzeranfragen die gespeicherten Metadaten durchsucht, Ergebnisse generiert und Berichte erstellt.

#### **1.4.2.4 Systemverwaltung**

Im Bereich Systemverwaltung wird der Gesamtbetrieb des Archivsystems geregelt. Es werden die Hard- und Softwarekonfigurationen des Systems verwaltet und mit Hilfe von Systemanalysen Arbeitsabläufe überwacht und optimiert sowie die archivierten Informationen inventarisiert und aktualisiert. Darüber hinaus werden spezifische Archivstandards und -richtlinien festgelegt und deren Einhaltung kontrolliert.

Weitere Punkte sind die Kundenbetreuung und Vertragsverhandlungen mit dem Produzenten über die Umsetzung des Datentransfers, Fragen der erforderlichen Dateiformate und Datenformen sowie über rechtliche Aspekte.

#### 1.4.2.5 Planung der Langzeitarchivierung

Die Planung der Langzeitarchivierung ist zuständig für die Beobachtung der sich ständig ändernden technischen Umgebung und Nutzergewohnheiten. Die Archivinhalte werden evaluiert, Empfehlungen für Migrations-Zeitzyklen ausgesprochen sowie Vorschläge für Standards und Richtlinien entwickelt.

#### 1.4.2.6 Bereitstellung

Der Bereich Bereitstellung ermöglicht dem Nutzer Anfragen zu stellen, die relevanten Informationen zu finden und zu erhalten.

Nutzeranfragen werden entgegengenommen, die Suche nach Archivinhalten wird koordiniert und als Ergebnis das benutzerspezifische Auslieferungspaket, in Abhängigkeit der Zugriffsberechtigungen, generiert und verteilt.

Der Datenfluss zwischen den einzelnen Funktionseinheiten des OAIS-Modells ist in Abbildung I-2 schematisch dargestellt<sup>13</sup>.

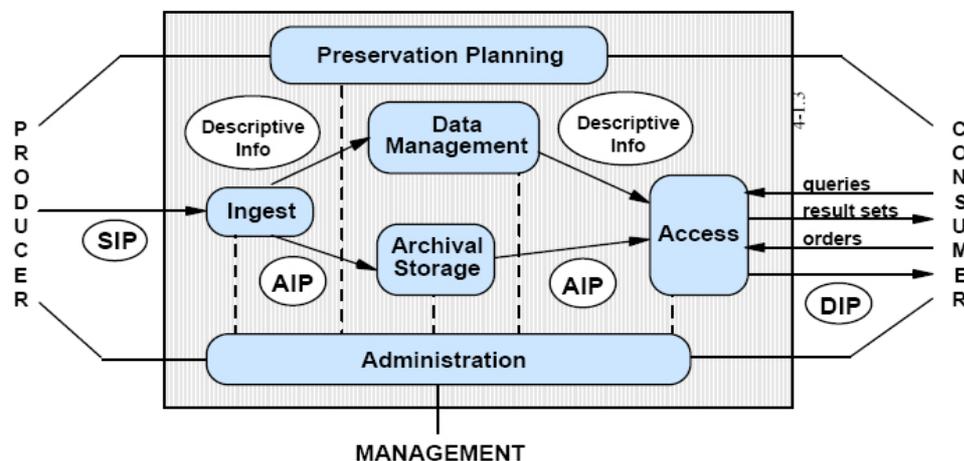


Abb. I-2: Datenflussmodell des OAIS-Standards<sup>14</sup>

<sup>13</sup> Ein detaillierteres Datenflussmodell ist im Anhang A-1 zu sehen.

<sup>14</sup> Quelle: CCSDS: Reference Model for an Open Archival Information System (OAIS), Januar 2002, S. 4-1

## 2 Relationale Datenbanksysteme

In einem Datenbanksystem werden Daten, die für einen spezifischen Zweck zusammengestellt wurden, strukturiert erfasst, gepflegt und verwaltet. Es werden Informationssuche und -zugriff ermöglicht.

Bei den für die Strukturierung der Daten relevanten Modellen ist das relationale Datenmodell marktbeherrschend. Aus diesem Grund konzentriert sich diese Arbeit auf relational organisierte Datenbanken.

In den folgenden Abschnitten werden die Bedeutung und der Aufbau von Datenbanksystemen sowie daraus resultierende Herausforderungen für die Archivierung betrachtet.

### 2.1 Bedeutung

Durch die zunehmende Produktion von Informationen in digitaler Form<sup>15</sup> wächst der Bedarf an Systemen wie Datenbanken, die große Datenmengen in unterschiedlichen Formaten strukturiert speichern, verwalten und den Zugriff auf diese sicherstellen. Datenbanken finden in fast allen gesellschaftlichen Bereichen Anwendung. Sie dienen der Speicherung von Registern, statistischen Informationen, Personal-, Unternehmens-, Mess- oder Konstruktionsdaten.

Verstärkt werden Datenbanken für die Regelung und Aufzeichnung von Geschäftsprozessen eingesetzt. Sie sind wesentlicher Bestandteil verschiedenster digitaler Informationssysteme, beispielsweise im Bereich des Daten- und Content-Managements.

Eine starke Verbreitung finden Datenbanken zudem im Internet. Webdatenbanken sind Voraussetzung für den Zugriff auf Informationsressourcen in einer verteilten Umgebung und für das Online-Angebot von Dienstleistungen. Viele Suchmaschinen, Bestell- und Katalogsysteme, die im Internet bereitgestellt werden, sind datenbankbasiert.

---

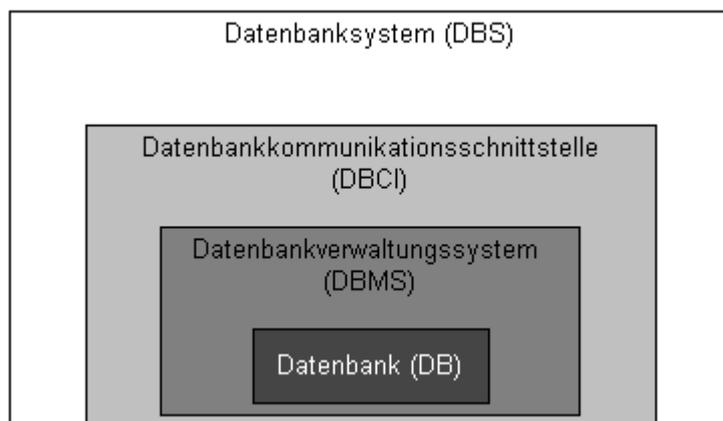
<sup>15</sup> s. dazu Kapitel I-1.1

Webinhalte werden zunehmend dynamisch generiert. Diese Entwicklung hat zu einer wachsenden Bedeutung von Web-Content-Management-Systemen (WCMS) wie TYPO3 geführt. Die Inhalte werden bei diesen Systemen ebenfalls in Datenbanken verwaltet.

## 2.2 Komponenten eines Datenbanksystems

In einem Datenbanksystem (*Database System (DBS)*) ist die Datenbank nur einer von drei Bestandteilen. Weitere Komponenten sind das Datenbankverwaltungssystem (*Database Management System (DBMS)*) und die Datenbankkommunikationsschnittstelle (*Database Communication Interface (DBC)*).

Abbildung I-3 zeigt schematisch den Aufbau eines Datenbanksystems.



**Abb. I-3:** Aufbau eines Datenbanksystems

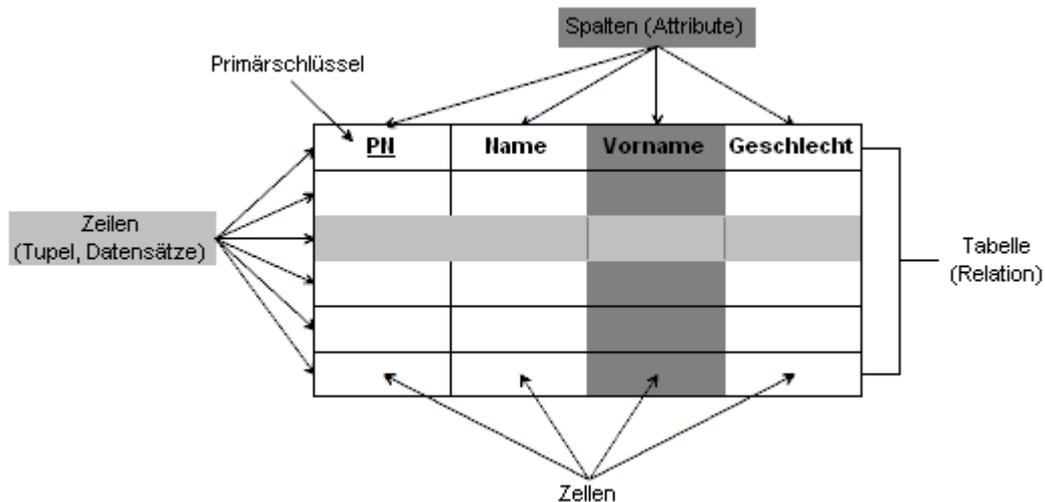
### 2.2.1 Die Datenbank und das relationale Datenmodell

Die Datenbank ist als Datenbasis ausschließlich für die systematische Speicherung der Inhalte zuständig und zu diesem Zweck in zwei Komponenten aufgeteilt. Der Basisbereich enthält die eigentlichen Daten, der Systemkatalog Angaben über die logische Datenstruktur.

Die Datenbankstruktur basiert auf dem relationalen Datenmodell. Das Grundkonzept des relationalen Modells beschreibt Tabellen, die zueinander in Beziehung stehen.

Grundsätzlich werden in einer relational organisierten Datenbank keine redundanten Daten gespeichert.

Die Elemente des relationalen Datenmodells werden in Abbildung I-4 dargestellt.



**Abb. I-4:** Elemente des relationalen Datenmodells

Eine Tabelle (Relation) stellt einen Objekttyp dar und enthält z. B. Angaben zu Personen. Die Tabellenzeilen (Tupel oder Datensätze) repräsentieren bestimmte Objekte (z. B. bestimmte Personen) mit gleichen Eigenschaften (z. B. Name, Vorname, Geschlecht). Diese Eigenschaften werden durch die Tabellenspalten (Attribute) beschrieben. Die eigentlichen Daten werden in den Tabellenzellen gespeichert und zeichnen sich durch einen bestimmten Wert und Datentyp aus. Mit Hilfe von Primärschlüsseln (z. B. Personenummer (PN)) wird jede Zeile eindeutig identifiziert.

Fremdschlüssel verweisen auf andere Tabellen. Die Verwendung von Fremdschlüsseln ist optional. Die Daten aus einzelnen Tabellen werden folglich erst durch Abfragen verknüpft. Durch diese Tabellenverknüpfung entsteht die eigentliche Information. Ohne die Verknüpfung ist eine relationale Datenbank lediglich eine zusammenhangslose Ansammlung von Daten.

In der Praxis setzen sich relationale Datenbanken aus hunderten von verknüpften Tabellen zusammen und zeichnen sich durch eine hohe Komplexität aus. Sind Datenbankstruktur und -inhalt nicht selbsterklärend ist die Interpretation, Verwaltung und Suche ohne Datenbankverwaltungssystem unmöglich.

Content- und Dokumenten-Management-Systeme besitzen eine hohe Komplexität hinsichtlich ihrer Datenbankstruktur und ihres Datenbankinhaltes. Abbildung I-5 zeigt einen Ausschnitt aus dem Datenmodell des Dokumenten-Management-Systems „KnowledgeTree“.

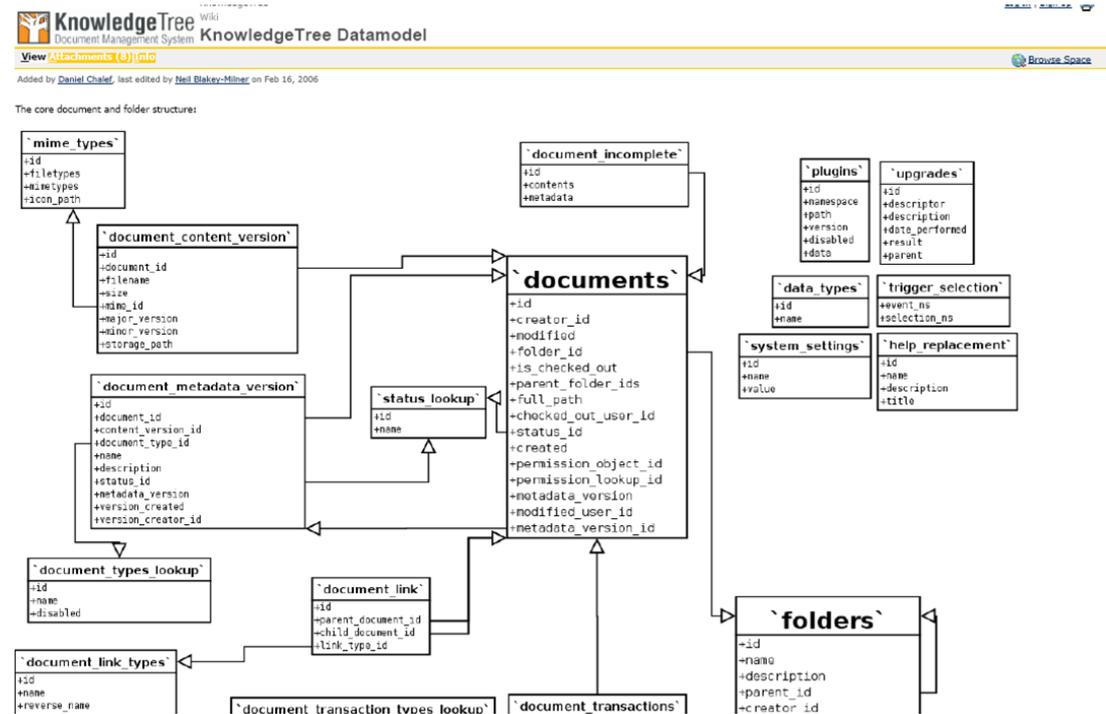


Abb. I-5: Datenmodell des DMS „KnowledgeTree“ (gekürzt)

Das Fehlen eindeutiger, klar definierter Bezeichnungen für Tabellen und Spalten wirkt sich nachteilig auf die zukünftige Interpretierbarkeit der Daten aus.

Datenbanken erlauben die Verwendung von Binary Large Objects (BLOBs)<sup>16</sup> sowie komplexer und benutzerdefinierter Typen. Diese Datentypen sind für die Archivierung problematisch.

### 2.2.2 Das Datenbankverwaltungssystem

Das Datenbankverwaltungssystem ist die zentrale Steuereinheit und Software-schnittstelle zwischen der physischen Datenbank und der Kommunikationsebene. Es enthält interne Mechanismen für die logische und physische Organisation des

<sup>16</sup> s. dazu Glossar

Datenmodells und ermöglicht durch entsprechende Werkzeuge die Strukturierung, Manipulation und Nutzung der Dateninhalte. Es werden nicht ausschließlich inhaltliche Datenbankabfragen verarbeitet, sondern auch Abfragen im Rahmen der Pflege der physischen Elemente und beschreibenden Informationen. Hinzu kommt die Verwaltung von Datentypen, Sichten (*Views*)<sup>17</sup>, Triggern<sup>17</sup>, Funktionen<sup>17</sup>, gespeicherten Prozeduren<sup>17</sup> und Zugriffsberechtigungen.

Ein Problem, das sich aus der Einbettung der Datenbank in ein Datenbanksystem ergibt, ist die zunehmende Abhängigkeit vom Datenbankverwaltungssystem. Heuscher meint dazu „[...] modern RDBMS move the physical storage of the data from the operating system (file level) to the application level (the internal storage of the RDBMS).“<sup>18</sup>.

Die Mehrheit der relationalen Datenbankverwaltungssysteme (*Relational Database Management System (RDBMS)*) verwendet die standardisierte Datenbanksprache SQL (*Structured Query Language*)<sup>19</sup>. Jedoch wird die Implementierung des SQL-Standards teilweise offen gelassen und dementsprechend in Datenbanksystemen in unterschiedlicher Form gestaltet. Oft wird nur der Kern von SQL verwendet, und es erfolgt eine ausgeprägte Anreicherung mit eigenen, nicht-standardisierten Elementen. Verschiedene produktspezifische SQL-Dialekte bilden sich heraus und finden Anwendung. Darüber hinaus werden proprietäre Programmiersprachen für die gespeicherten Routinen verwendet.

### 2.2.3 Die Datenbankkommunikationsschnittstelle am Beispiel von SQL

Der Datenbankkommunikationsschnittstelle kommt im Gesamtsystem eine bedeutende Rolle zu. Sie ermöglicht dem Nutzer die Kommunikation mit dem Datenbanksystem und den Zugriff auf die Datenbankinhalte. Abfragen sind für den Zugriff von wesentlicher Bedeutung. Erst durch Datenbankabfragen entstehen die eigentlichen Informationen. Die Bedeutung der gesamten Datenbank wird ersichtlich.

---

<sup>17</sup> s. dazu Glossar

<sup>18</sup> Heuscher [u. a.]: Providing Authentic Long-term Archival Access to Complex Relational Data, 2004, S. 3, URL: [http://arxiv.org/PS\\_cache/cs/pdf/0408/0408054v1.pdf](http://arxiv.org/PS_cache/cs/pdf/0408/0408054v1.pdf)

<sup>19</sup> SQL wird im Kapitel I-2.2.3 betrachtet.

Die Datenbankkommunikationsschnittstelle tauscht darüber hinaus Daten mit dem Datenbankverwaltungssystem aus und verfügt über automatisiert ablaufende Programme, die das System bei der Verarbeitung von Ein- und Ausgaben unterstützen. So wird mit Hilfe einer Protokolldatei das An- und Abmelden von Nutzern sowie deren Kommunikation mit dem System dokumentiert.

Die Kommunikation mit dem Datenbanksystem kann über Benutzeroberflächen oder direkt auf der Kommandoebene erfolgen. Für die Ausführung existieren zwei verschiedene Gruppen von Befehlen, die Datenbeschreibungssprache (*Data Definition Language (DDL)*) und die Datenmanipulationssprache (*Data Manipulation Language (DML)*). Die Unterscheidung dieser beiden Sprachen führt zu einer klaren Trennung zwischen Datenstrukturierung und -zugriff. Änderungen am Datenmodell können nur von befugten Personen durchgeführt und nicht versehentlich von Datenbanknutzern verursacht werden.

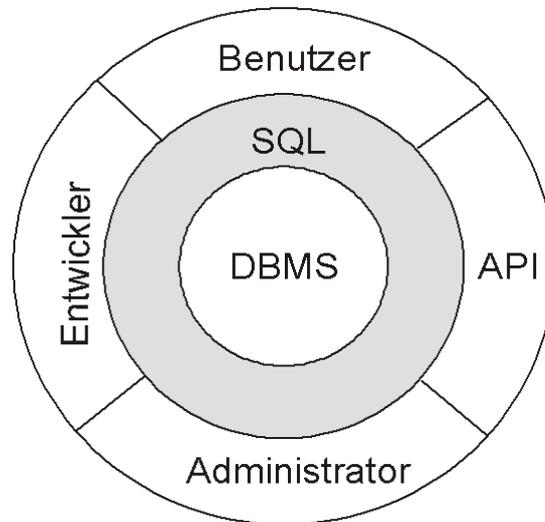
SQL (*Structured Query Language*)<sup>20</sup> ist eine Skriptsprache, in der die Datenbeschreibung und die Datenmanipulation integriert sind. Sie ermöglicht sowohl das Definieren, Anlegen und Verändern von Tabellen (Beschreibung) als auch das Hinzufügen, Ändern und Abfragen von Daten (Manipulation). Darüber hinaus können Zugriffsrechte festgelegt werden. SQL kann damit auch als Datenkontrollsprache verwendet werden.

SQL ist eine rein deskriptive Sprache. Sie beschreibt, was ausgeführt werden soll, aber nicht, wie etwas ausgeführt werden soll.

Abbildung I-6 stellt schematisch die Rolle von SQL als Kommunikationsschnittstelle dar. Eine Kommunikation kann sowohl zwischen Anwendern (Administrator, Entwickler, Benutzer) und dem Datenbankverwaltungssystem als auch zwischen Anwendungsprogrammen (*Application Programming Interface (API)*) und dem Verwaltungssystem stattfinden. Programmschnittstellen enthalten die Datenbanklogistik in Form von Abfragen, Datenverarbeitung und Nutzerschnittstellen und können nicht mit Hilfe von SQL gesichert werden.

---

<sup>20</sup> Die Entwicklung der Skriptsprache SQL begann in den 1970er Jahren. 1987 wurde sie als internationaler Standard ISO/IEC 9075 anerkannt und liegt aktuell in der Version SQL:2008 vor.



**Abb. I-6:** SQL als Kommunikationsschnittstelle<sup>21</sup>

Ein bedeutender SQL-Bestandteil im Bereich der Datenmanipulation ist die Formulierung von Abfragen mit Hilfe des SELECT-Befehls. Die SELECT-Anweisung beschreibt die Syntax, die für die Umsetzung von Abfrageoperationen benötigt wird. Diese Abfrageoperationen werden durch die relationale Algebra<sup>22</sup> beschrieben und vom Datenbankverwaltungssystem ausgeführt, um auf Daten zuzugreifen, die in verschiedenen Tabellen abgelegt sind. Die wichtigsten Operationen sind die Selektion und die Projektion. Sie werden im Teil III dieser Arbeit genauer betrachtet.

Der SQL-Standard wird durch eine Reihe von Erweiterungen ergänzt. Der Standard SQL/XML (ISO/IEC 9075-14) spielt dabei im Kontext der Archivierung eine entscheidende Rolle. Er befasst sich mit der Verwendung von SQL in Verbindung mit dem XML-Format. Der Standard ermöglicht die Speicherung von XML-Dokumenten in SQL-Datenbanken, die Einbindung der Abfragesprachen XPath und XQuery sowie den Export von SQL ins XML-Format.

Aus den Wechselwirkungen, die zwischen den Systemkomponenten Datenbank, Verwaltungssystem und Kommunikationsschnittstelle bestehen, ergibt sich eine weitere Herausforderung für die Archivierung, die Bestimmung des Informationsobjekts.

---

<sup>21</sup> Quelle: Däßler: Das Einsteigerseminar MySQL 5, 2005, S. 81

<sup>22</sup> s. dazu Glossar

Informationsobjekte können in folgenden Formen definiert werden:

- gesamtes Datenbanksystem
- einzelne Datenbank
- einzelne Tabellen
- einzelne Datensätze
- Inhalte, die über mehrere Tabellen verteilt sind.

### 3 XML als Archivformat

XML (*Extensible Markup Language*)<sup>23</sup> ist ein offener und lizenzfreier Standard des W3C (*World Wide Web Consortium*), der für den Datenaustausch im Internet von Bedeutung ist und sich auch hervorragend für die Archivierung eignet.

XML ist aus der Auszeichnungssprache SGML (*Standard Generalized Markup Language*)<sup>24</sup> hervorgegangen und stellt eine vereinfachte Teilmenge dieser Sprache dar. XML weist wesentliche Vorzüge von SGML auf, ist jedoch weniger komplex.

Als Auszeichnungssprache (*Markup Language*)<sup>24</sup> stellt XML die Syntax bereit, mit der die logische Struktur von elektronischen Dokumenten beschrieben wird. Die Inhaltsinformationen werden durch Zusatzinformationen ergänzt.

XML-Dokumente zeichnen sich durch eine gute und übersichtliche Strukturierung aus, die durch eine hierarchische Organisation erreicht wird. Ein XML-Dokument gliedert sich in zwei Bereiche. Der Prolog enthält unter anderem Angaben zur verwendeten XML-Version und zur Zeichencodierung. Der Bereich mit dem primären Inhalt besteht aus Elementen. Auf oberster Ebene befindet sich das Wurzelement, unter dem weitere Unterelemente verschachtelt angeordnet sind. Mit Hilfe von Attributen können die Elemente durch Zusatzinformationen genauer spezifiziert werden. Weitere mögliche Dokumentbestandteile sind Kommentare und Verarbeitungsanweisungen für andere Skriptsprachen wie PHP.

Abbildung I-7 zeigt den allgemeinen Aufbau eines XML-Dokumentes.

```
<?xml version="1.0" encoding="utf-8"?>
<WURZELEMENT>
  <UNTERELEMENT_1>Elementinhalt</UNTERELEMENT_1>
  <UNTERELEMENT_2 ATTRIBUT="Attributwert">
    <UNTERELEMENT_3>Elementinhalt</UNTERELEMENT_3>
  </UNTERELEMENT_2>
</WURZELEMENT>
```

**Abb. I-7:** Allgemeiner Aufbau eines XML-Dokumentes

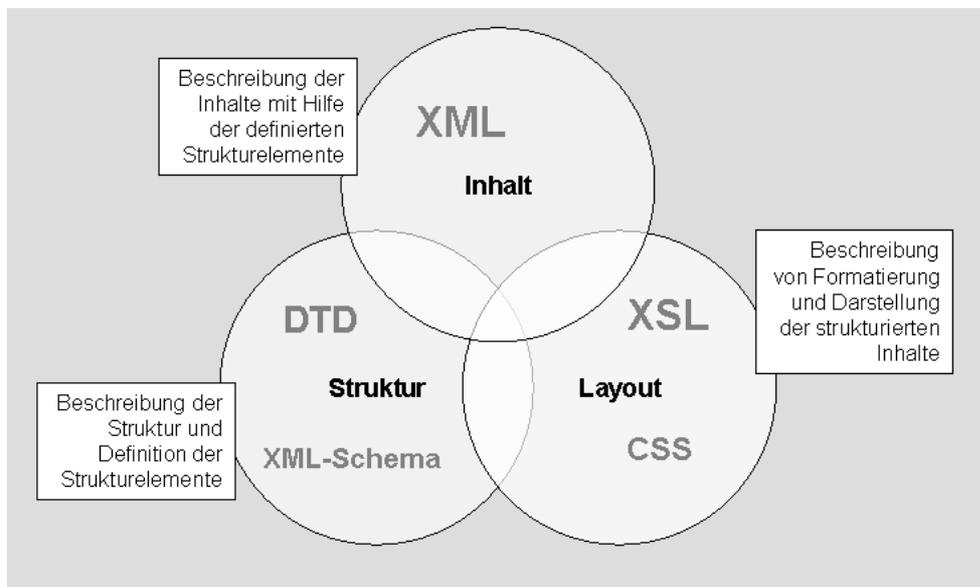
---

<sup>23</sup> Der XML-Standard wurde im Februar 1998 vom W3C verabschiedet und liegt aktuell in der fünften Ausgabe von 2008 vor.

<sup>24</sup> s. dazu Glossar

Mit Hilfe von Schemasprachen wie der Dokumenttypdefinition (*Document Type Definition (DTD)*) und der XML-Schema-Definition (*XML Schema Definition (XSD)*) wird sowohl die Dokumentenstruktur als auch die Semantik der Strukturelemente beschrieben. Alle potentiellen Elemente werden deklariert. Sie können bezüglich ihrer Syntax, ihrer Häufigkeit und Reihenfolge im Dokument, ihres Inhaltes (Standardwerte oder feste Werte) und der Anzahl ihrer Unterelemente spezifiziert werden. Außerdem besteht die Möglichkeit festzulegen, welche Attribute den Elementen zugeordnet werden dürfen. Ein XML-Schema erlaubt zusätzlich die Definition von Datentypen für Elemente und Attribute. Die Strukturdeklaration wird im Prolog des entsprechenden XML-Dokumentes eingebunden.

Für die Darstellung von XML-Dokumenten wird die Formatierungssprache XSL (*Extensible Stylesheet Language*) verwendet. Mit Hilfe dieser Sprache wird das Erscheinungsbild (*Layout*) des XML-Dokumentes genau beschrieben. Das Dokument kann in andere Formate umgewandelt werden.



**Abb. I-8:** Komponenten von XML

In Abbildung I-8 wird der Zusammenhang zwischen den XML-Komponenten schematisch veranschaulicht. Verdeutlicht wird dabei die mit XML umgesetzte Trennung von Inhalt, Struktur und Erscheinungsbild. Die Strukturbeschreibung und -definition hat keinerlei Auswirkung auf die Darstellung des Dokumentes.

XML ist eine erweiterbare Sprache, deren Aufbau nicht vordefiniert ist. Der Standard liefert lediglich Vorschriften für das Erstellen eigener Typdefinitionen. Mit einer eigenen Dokumenttypdefinition oder einem eigenen XML-Schema können neue Sprach-elemente beschrieben werden. „XML ist damit ein Standard zur Definition von beliebigen, in ihrer Grundstruktur jedoch stark verwandten Auszeichnungssprachen“<sup>25</sup> und wird folglich auch als Metasprache bezeichnet. Archivrelevante Praxisbeispiele sind die Metadaten-Containerformate METS und PREMIS. Werden eigene XML-Formate definiert, ist es im Kontext der Archivierung notwendig, die entsprechende Dokumenttypdefinition beziehungsweise das XML-Schema aufzubewahren.

XML ist textbasiert, kann durch die Verwendung von Unicode mit Sonderzeichen erkennen und somit von jedem Textverarbeitungsprogramm erstellt und gelesen werden. Aufgrund seiner übersichtlichen Struktur kann XML ohne weitere Softwarehilfsmittel vom Menschen inhaltlich interpretiert werden (Lesbarkeit). Es ist somit unabhängig von Hard- und Software und deshalb wesentlich beständiger als andere Formate.

Ein weiterer Aspekt, der für die Verwendung von XML als Archivformat spricht, ist der Umgang mit unterschiedlichen Objekttypen. XML-Elemente beschreiben Text, Grafiken oder auch abstraktes Wissen.

XML findet weltweit eine breite Anwendung und ist allgemein akzeptiert. Für die Nutzung von XML ist eine Vielzahl von Softwarewerkzeugen verfügbar. Softwarehersteller wie Oracle, Microsoft und IBM verwenden XML als Austauschformat.

Auf der Grundlage von XML wurden vom W3C weitere Sprachen definiert, die XML-Ausdrücke für häufig benötigte Funktionen anbieten. Ein Beispiel ist die Abfragesprache XQuery (*Extensible Query Language*), die das XML-Datenmodell verwendet.

In der folgenden Übersicht (s. Abbildung I-9) werden Vor- und Nachteile der Auszeichnungssprache XML im Hinblick auf die Archivierung dargestellt.

---

<sup>25</sup> nestor - Kompetenznetzwerk Langzeitarchivierung: nestor Glossar : XML, 01.01.2004, URL: <http://langzeitarchivierung.de/index.php?module=Encyclopedia&func=displayterm&id=20&vid=1>

<b>Vorteile</b>	<b>Nachteile</b>
offener und lizenzfreier Standard des W3C	XML-Dokumente können sehr groß und komplex werden
Vorteile von SGML, aber geringere Komplexität	mögliche Redundanzen bei den Strukturelementen innerhalb eines XML-Dokumentes
Trennung von Inhalt, Struktur und Erscheinungsbild	
gute und übersichtliche Strukturierung	
Erweiterbarkeit (Metasprache)	
Lesbarkeit	
Soft- und Hardwareunabhängigkeit	
Umgang mit unterschiedlichen Objekttypen	
Verfügbarkeit einer Vielzahl von Softwaretools	
Einbindung in viele Softwarelösungen	

**Abb. I-9:** Vor- und Nachteile von XML

## 4 Stand der Langzeitarchivierung von Datenbanken

Die Langzeitarchivierung von Datenbanken gewinnt zunehmend an Bedeutung. Das wird deutlich, wenn man den Stand der Forschung und Entwicklung betrachtet. Diverse kommerzielle Komplettlösungen sind bereits verfügbar. Darüber hinaus gibt es eine Reihe von Projekten, die sich mit dieser Thematik befassen.

CHRONOS<sup>26</sup>, ERPANET Solution<sup>27</sup> und die Hewlett Packard Database Archiving Software<sup>28</sup> sind Beispiele für kommerzielle Komplettlösungen. Diese Systeme sind weniger auf die Anforderungen von Archiven ausgerichtet. Sie verzichten zwar auf proprietäre Formate und verwenden stattdessen Archivformate, wie ASCII-Text und XML, berücksichtigen jedoch nicht die Datenbankapplikationsebene und bedeutende Aspekte wie die Bewertung oder Bestandserhaltung. Die Erfassung von Metadaten wird mehrheitlich unzureichend unterstützt. In der Regel sind die Systeme an ein bestimmtes Datenbankverwaltungssystem gebunden und können aus diesem Grund nicht mit verschiedenen, herstellerspezifischen SQL-Dialekten umgehen. Das CHRONOS-System ist darüber hinaus auf den Server der Herstellerfirma CSP angewiesen und muss von dieser verwaltet werden.

Generell sind die angebotenen, kommerziellen Komplettlösungen eher an den Erfordernissen der Wirtschaft oder dem Datenmanagement orientiert. Eine versionierte Datensicherung wird durchgeführt, die Daten werden lauffähig gehalten sowie revisionssicher und zeitbefristet aufbewahrt. Die Systeme dienen oft nur der Systementlastung und dem Einhalten vorgegebener Aufbewahrungsfristen.

---

<sup>26</sup> CHRONOS ist eine Serverlösung der CSP GmbH und Co. KG. Für weitere Informationen s. CSP: Datenbankarchivierung mit CHRONOS, URL: [http://www.csp-sw.de/de/inhalt.php?kategorie=c114\\_L\\_sungen\\_CHRONOS](http://www.csp-sw.de/de/inhalt.php?kategorie=c114_L_sungen_CHRONOS)

<sup>27</sup> ERPANET Solution ist ein Service des Projektes ERPANET. Für weitere Informationen s. ERPANET: Electronic Resource Preservation and Access Network, 2007, URL: <http://www.erp-net.org>

<sup>28</sup> Diese Archivierungssoftware wird von der Firma Hewlett-Packard (HP) angeboten. Es handelt sich dabei um eine komplette Serverlösung. Für weitere Informationen s. HP: HP Database Archiving software, 2009, URL: [http://h18006.www1.hp.com/products/software/im/governance\\_ediscovery/databasearchiving/index.html](http://h18006.www1.hp.com/products/software/im/governance_ediscovery/databasearchiving/index.html)

Neben kommerziellen Komplettlösungen existieren einige Projekte, die sich mit archivisch ausgerichteten Lösungen befassen.

Im Rahmen dieser Arbeit wurden sechs Projekte betrachtet<sup>29</sup>. Alle vorgestellten Modelle sind soft- und hardwareunabhängig und folglich nicht an ein bestimmtes Datenbankverwaltungssystem gebunden. Neben den Datenbankinhalten werden die Struktur, die notwendigen Metadaten sowie teilweise Sichten und andere Datenbankroutinen archiviert. Die Inhalte werden in XML oder flachen Datenformaten wie Unicode gespeichert. Für die Aufbewahrung der Struktur wird entweder SQL oder XML verwendet. Die Metadaten werden im XML-Format abgelegt.

Eine Übersicht, die für jedes Projekt zeigt, wie Daten, Struktur, Metadaten, Abfragen und Verknüpfungen archiviert und Archivdaten wieder bereitgestellt werden, ist im Anhang A-2 zu finden.

Vielversprechend sind das RODA-<sup>30</sup> und das ARELDA-Projekt<sup>31</sup>. Aus dem ARELDA-Projekt ist die Softwarelösung SIARD hervorgegangen. Beide Projekte orientieren sich am OAIS-Modell und verwenden für die Archivierung der Datenbankstruktur und Datenbankinhalte die Formate XML bzw. SQL.

Keines der betrachteten Modelle übernimmt die vollständige Datenbankfunktionalität und -nutzung. Nur das, was in der Datenbank abgelegt wurde, wird archiviert. Nicht gespeicherte Abfragen, Berichte und Routinen zur Bearbeitung und Repräsentation der Daten gehen bei der Archivierung verloren.

---

<sup>29</sup> Ein detaillierte Betrachtung der sechs Projekte erfolgt in der Vergleichsstudie: Glöde; Richter: Nicht-kommerzielle Datenbankarchivierung auf der Basis von XML, 2009

<sup>30</sup> RODA (*Repository of Authentic Digital Objects*) ist ein Gemeinschaftsprojekt des Portugiesischen National Archivs und der Universität Minho. Für weitere Informationen s. Ramalho [u. a.]: Relational Database Preservation through XML modelling, 2007, URL: <http://repositorium.sdum.minho.pt/bitstream/1822/7120/1/EML2007-final.pdf>

<sup>31</sup> ARELDA (Archivierung elektronischer digitaler Daten und Akten) ist ein Projekt des schweizerischen Bundes. Für weitere Informationen s. Heuscher [u. a.]: Providing Long-term Archival Access to Complex Relational Data, 2004

## 5 Fazit

Relationale Datenbanken haben aufgrund der zunehmenden Produktion digitaler Informationen einen hohen Stellenwert in allen Bereichen der Wirtschaft, Wissenschaft und Verwaltung. Dementsprechend besteht ein starkes Interesse an der langfristigen Aufbewahrung dieser Datenbanken.

Die Besonderheiten relationaler Datenbanken liegen in ihrer Datenstruktur sowie in der Tatsache, dass die eigentliche Information erst durch die Verknüpfung der Tabellen entsteht. Diese Verknüpfung wird durch Abfragen realisiert. Daten aus verschiedenen Tabellen werden so zu Datensätzen und Berichten zusammengesetzt und mit Zusatzinformationen, wie Spaltenbezeichnungen in Form von Aliasnamen, versehen.

Die Archivierung stößt im Bereich der Datenbanken auf eine Reihe von Problemen. Datenbanken sind in der Praxis sehr komplex. Gewöhnlich werden keine eindeutigen, klar definierten Tabellen- und Spaltenbezeichnungen verwendet und es fehlen Fremdschlüsseldefinitionen, Sichten und Feldbeschreibungen. Die Datenbankstruktur, die beschreibenden Informationen, Tabellenverknüpfungen sowie die Repräsentationen und Interpretationen der Daten sind überwiegend in den Programmschnittstellen für die Kommunikation hinterlegt. Diese Schnittstellen sind meist nicht zugänglich und können nur mit Hilfe der Emulation<sup>32</sup> archiviert werden. Darüber hinaus sind die Datenbanken an das jeweilige Datenbanksystem gebunden. Daten werden oft intern im Datenbanksystem durch Prozeduren, Funktionen und Trigger verändert, optimiert und verwaltet. Komponenten, die ebenfalls nur unter Anwendung der Emulation<sup>32</sup> archiviert werden könnten.

Die in Kapitel 4 genannten Projekte zeigen, dass die Archivierung von Datenbanken prinzipiell möglich ist. Die dort vorgestellten Archivierungsmodelle haben jedoch Probleme bei der Übernahme der vollständigen Datenbankfunktionalität und -nutzung.

---

<sup>32</sup> Die Emulation wird aufgrund ihrer Systemabhängigkeit als nicht geeignete Archivierungsstrategie angesehen. Die Vor- und Nachteile der Emulation werden im Kapitel I-1.3 genauer betrachtet.

Folgende Schlussfolgerungen können für die Datenbankarchivierung formuliert werden:

1. Eine reine Übernahme der Daten aus den Tabellen ist nicht ausreichend. Neben den Datenbankdaten müssen zwingend die Strukturinformationen (Tabellen, deren Spalten, Eigenschaften und Verknüpfungen) und beschreibenden Informationen (z. B. Feldbeschreibungen) archiviert werden. Die Tabellenverknüpfungen lassen sich über SQL-Abfragen ermitteln, falls sie nicht mit Hilfe von Fremdschlüsseln definiert wurden.
2. Die Abbildung des Nutzerverhaltens und die Bewahrung eines Teils der Datenbankfunktionalität sind mit Hilfe der Datenbankabfragen ohne Emulation möglich. Die Abfragen können aus Sichten oder SQL-Protokollen ermittelt werden.
3. Als Archivformat eignet sich XML aus Gründen der Standardisierung, Systemunabhängigkeit, übersichtlichen Strukturierung, Lesbarkeit und Erweiterbarkeit.
4. In jedem Fall ist eine Bewertung der Datenbank notwendig. Nicht alle Datenbanken sind archivierbar. Die Datenbanken von Dokumenten- und Content-Management-Systemen und die meisten Datenbanken mit geowissenschaftlichen Daten können beispielsweise derzeit nicht archiviert werden. Die gespeicherten Daten müssen durch SQL-Abfragen interpretierbar sein und dürfen bei der Repräsentation und Interpretation nicht an spezifische Programme gebunden sein.
5. Eine digitale Archivlösung für Datenbanken muss unbedingt OAIS-konform sein. Das OAIS-Modell berücksichtigt alle wesentlichen Aspekte der digitalen Langzeitarchivierung.

## II **Das Archivierungskonzept**

### 1 **Das Gesamtkonzept**

Unter Berücksichtigung der im Teil I abgeleiteten Schlussfolgerungen für die Datenbankarchivierung<sup>33</sup> wurde ein Konzept für die Langzeitarchivierung von relationalen Datenbanken entwickelt. Es basiert vollständig auf dem OAIS-Modell<sup>34</sup>, ist systemunabhängig und legt hohen Wert auf die Lesbarkeit der Daten.

In den folgenden Abschnitten wird das Gesamtkonzept anhand der Daten-, Datenfluss- und Prozessmodelle näher vorgestellt. Für die beispielhafte Veranschaulichung wird eine einfache Melderegister-Datenbank verwendet, dessen Struktur im Anhang A-3 genauer dokumentiert ist.

#### 1.1 **Daten- und Datenflussmodelle**

Den Ausgangspunkt des Archivierungskonzeptes bilden ein relationales Datenbanksystem (RDBS) und das Backup einer darin enthaltenen Datenbank. Die Datenbank stellt das Informationsobjekt der Archivierung dar. Übernommen werden sowohl Datenbankstruktur und -daten als auch an das System gestellte Abfragen. Darüber hinaus werden verschiedene Metadaten berücksichtigt. Finden in der zu archivierenden Datenbank keine eindeutigen, klar definierten Tabellen- und Spaltenbezeichnungen Anwendung, muss zusätzlich eine Felddescription archiviert werden.

In Anlehnung an den OAIS-Standard werden Informationspakete verwendet. Es wird zwischen dem Einlieferungspaket (SIP), dem Archivpaket (AIP) und dem Auslieferungspaket (DIP) unterschieden.

Prinzipiell bestehen die Informationspakete aus den Inhaltsinformationen und dazugehörigen Metadaten. Die Inhaltsinformationen werden in allen drei Paketen durch die Datenbankstrukturdaten und Datenbankdaten sowie durch die Abfragen, als Daten der Logistik, gebildet. Die Metadaten liegen in Form der Felddescription und weiteren Zusatzinformationen vor.

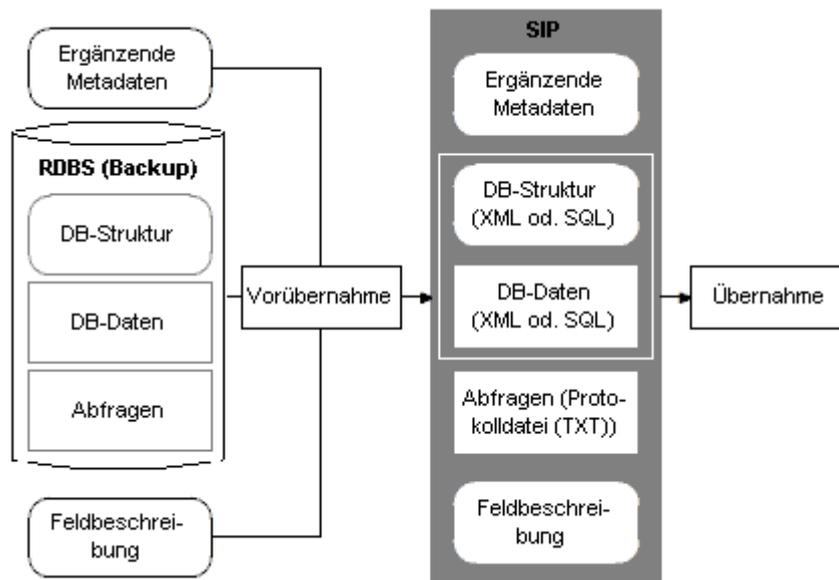
---

<sup>33</sup> s. dazu Kapitel I-5.

<sup>34</sup> Eine detaillierte Vorstellung des OAIS-Modells erfolgt im Kapitel I-1.4.

In den folgenden schematischen Darstellungen werden die Inhaltsinformationen als Rechtecke und die Metadaten durch Rechtecke mit abgerundeten Ecken gekennzeichnet. Detaillierte Datenflussmodelle sind im Anhang A-4 zu sehen.

### 1.1.1 Einlieferungspaket



**Abb. II-1:** Ausgangsdaten und Einlieferungspaket

Wie in Abbildung II-1 dargestellt, entsteht das Einlieferungspaket als Ergebnis der Vorübernahme und wird an die Übernahmeprozesse übergeben.

Die Datenbankstruktur (DB-Struktur) und die Datenbankdaten (DB-Daten) liegen, abhängig vom möglichen Datenbankexport, in einem systemspezifischen XML-Format oder in SQL vor. Die Angaben zur Datenbankstruktur bilden die strukturellen Repräsentationsinformationen des Informationspaketes und beinhalten Beschreibungen der vorhandenen Tabellen, Tabellenspalten sowie der Primär- und Fremdschlüssel.

Abbildung II-2 zeigt, wie die Struktur und die Daten einer einzelnen Datenbanktabelle des Melderegisters in einem XML-Dokument dargestellt werden, das aus dem Datenbanksystem MySQL exportiert wurde. Das vollständige XML-Dokument ist im Anhang A-5.1 hinterlegt.

```
<?xml version="1.0"?>
<mysqldump xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<database name="melderegister">
  <table_structure name="tabelle1">
    <field Field="feld1" Type="int(11)" Null="NO" Key="PRI" Extra="" />
    <field Field="feld2" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld3" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld4" Type="date" Null="YES" Key="" Extra="" />
    ...
    <key Table="tabelle1" Non_unique="0" Key_name="PRIMARY"
      Seq_in_index="1" Column_name="feld1" Collation="A" Cardinality="2"
      Null="" Index_type="BTREE" Comment="" />
    <options Name="tabelle1" Engine="InnoDB" Version="10"
      Row_format="Compact" Rows="10" Avg_row_length="1638"
      Data_length="16384" Max_data_length="0" Index_length="0"
      Data_free="4194304" Create_time="2009-06-21 10:19:02"
      Collation="latin1_swedish_ci" Create_options="" Comment="" />
  </table_structure>
  <table_data name="tabelle1">
    <row>
      <field name="feld1">1</field>
      <field name="feld2">Löwe</field>
      <field name="feld3">Ralf</field>
      <field name="feld4">1939-06-06</field>
      ...
    </row>
    ...
  </table_data>
</database>
</mysqldump>
```

**Abb. II-2:** Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten  
(gekürzt)

Die Datenbankabfragen sind im Protokoll des Datenbanksystems enthalten, das im Einlieferungspaket als Textdatei vorliegt. Ein Beispiel für eine MySQL-Protokolldatei ist im Anhang A-5.2 zu sehen.

Sowohl die Feldbeschreibung als auch die ergänzenden Metadaten können in unterschiedlichen digitalen Formaten oder in nicht-elektronischer Form geliefert werden.

In Abbildung II-3 ist der Ausschnitt der Feldbeschreibung des Melderegisterbeispiels zu sehen. Sie liegt in Form einer Liste vor und enthält Angaben zu den verwendeten Tabellen- und Spaltennamen, deren Klartext und Bedeutung sowie Angaben zu den vergebenen Datentypen. Die vollständige Datenbankfeldbeschreibung befindet sich im Anhang A-5.3.

tabelle1 Person			
feld1	Personennummer	Primärschlüssel	INTEGER, NOT NULL
feld2	Name		CHAR
feld3	Vorname		CHAR
feld4	Geburtsdatum		DATE
feld5	Geburtsort		CHAR
feld6	Geschlecht	w od. m	CHAR
feld7	Staatsangehörigkeit		CHAR
feld8	Religion		CHAR
feld9	Todesdatum		DATE
feld10	Todesort		CHAR
feld11	Wahlrecht	Ja od. Nein	CHAR
feld12	Kinder	Verweis auf Personennummer	INTEGER
feld13	Familienstand		CHAR

**Abb. II-3:** Datenbankfeldbeschreibung (gekürzt)

Bei den ergänzenden Metadaten handelt es sich vorwiegend um beschreibende Kontextinformationen, die Angaben zur institutionellen und organisatorischen Einordnung, zum Inhalt, Zweck und zur Bedeutung der Datenbank umfassen. Sie werden durch technische und rechtliche Informationen ergänzt.

In der Softwarelösung SIARD werden die Metadaten ausführlich beschrieben. Beispiele für Metadaten können daher der SIARD-Formatbeschreibung entnommen werden<sup>35</sup>.

### 1.1.2 Archivpaket

Das Archivpaket stellt die Form dar, in der die Datenbank langfristig aufbewahrt wird. Beim Aufbau des Paketes müssen Archivierungsanforderungen beachtet werden. Ein bedeutender Aspekt ist in diesem Zusammenhang die Verwendung geeigneter Formate.

---

<sup>35</sup> s. dazu Schweizerisches Bundesarchiv BAR: SIARD : Formatbeschreibung, 30.11.2008, URL: <http://www.bar.admin.ch/aktuell/00568/00702/00703/index.html?lang=de&download=M3wBPgDB/8ulI6Du36WenojQ1NTTjaXZnqWfVpzLhmfnapmmc7Zi6rZnqCkkIN0hHx+bKbXrZ6lhuDZz8mMps2g-pKfo>

In den folgenden Abschnitten werden potentielle Archivformate für Datenbankstruktur, Datenbankdaten sowie für Datenbankabfragen miteinander verglichen und es wird der Aufbau des verwendeten Archivpaketes genauer vorgestellt.

### **1.1.2.1 Gegenüberstellung potentieller Archivformate**

#### **1.1.2.1.1 Datenbankstruktur und Datenbankdaten**

Für die Speicherung der Datenbankstruktur und Datenbankdaten sind das SQL-, das CSV-Format (*Comma-Separated Values*) und ein XML-basiertes Format vorstellbar.

SQL ist ein internationaler Standard. Die Mehrheit der relationalen Datenbanksysteme basiert auf dieser Datenbanksprache. Datenbankexport und Bereitstellung sind daher leicht durchführbar.

In den Datenbanksystemen wird der standardisierte SQL-Kern allerdings durch individuelle, nicht-standardisierte Elemente ergänzt. Produktspezifische SQL-Dialekte finden Verwendung. Bei der Übernahme wird somit eine Konversion dieser Dialekte in standardkonformes SQL notwendig.

SIARD ist ein Beispiel für eine Softwarelösung, bei der das SQL-Format verwendet wird. Vor der Archivierung ist hier eine umfassende Analyse der produktspezifischen SQL-Formate unumgänglich. Nicht-standardisierte SQL-Elemente werden herausgefiltert und wenn möglich umgewandelt. Dieser Vorgang ist mit einem hohen Aufwand verbunden, da er teilweise manuell durchgeführt wird oder die Verwendung speziell entwickelter Werkzeuge erfordert.

Bei der Bereitstellung der archivierten Datenbank ist eine Anpassung des SQL-Archivformats an den Dialekt des jeweiligen Datenbanksystems notwendig.

SQL ist eine rein deskriptive Sprache und damit prinzipiell gut lesbar. Eine Einschränkung der Lesbarkeit ergibt sich dadurch, dass SQL-Kenntnisse erforderlich sind. Die Verarbeitung der SQL-Elemente gestaltet sich sehr aufwändig, da der Standard komplex und umfangreich ist.

Aufgrund der Systemabhängigkeit, eingeschränkter Lesbarkeit und aufwändiger Verarbeitung ist SQL nicht für die Archivierung der Datenbankstruktur und Datenbankdaten geeignet.

Das CSV-Format ist sehr kompakt und lässt sich leicht verarbeiten. Die Mehrheit der Datenbanksysteme unterstützt den direkten Datenexport in CSV.

Ein großer Nachteil des Formats ist die Trennung von Daten und Metadaten. Oft werden nur die Daten gespeichert. Strukturbeschreibungen sind daher zusätzlich erforderlich. Darüber hinaus werden in der Regel keine Informationen zu den Tabellenverknüpfungen und den Datenfeldern gesichert.

Die Struktur des CSV-Formates wird im RFC4180<sup>36</sup> grundlegend beschrieben. Ein allgemein gültiger Standard wurde bis zum jetzigen Zeitpunkt jedoch noch nicht verabschiedet. Die fehlende Standardisierung führt beispielsweise zu Unterschieden bei der Formatierung und der Zeichencodierung. Fehlt eine ausführliche Strukturbeschreibung, ist die Lesbarkeit eingeschränkt und das Format systemabhängig.

Das CSV-Format erfüllt nicht die notwendigen Formatanforderungen der digitalen Langzeitarchivierung und scheidet daher ebenfalls als Archivformat aus.

Für die Speicherung der Datenbankstruktur und Datenbankdaten ist zudem die Verwendung eines XML-basierten Formates, dessen Struktur durch ein XML-Schema oder eine Dokumenttypdefinition beschrieben wird, möglich.

Dieses Format ist rein deskriptiv, einfach strukturiert und systemunabhängig. Es zeichnet sich durch gute Lesbarkeit und einfache Verarbeitung aus.

Bisher wurde vom W3C noch kein XML-Standard für relationale Datenbankinhalte verabschiedet. Ein XML-basiertes Archivformat für Datenbankabfragen ist somit nicht standardisiert.

Aufgrund der guten Lesbarkeit, einfachen Verarbeitung und Systemunabhängigkeit erfüllt ein XML-basiertes Format alle wesentlichen Formatanforderungen der digitalen

---

<sup>36</sup> RFC ist die Abkürzung für „Request for Comments“ (s. Glossar). Für den RFC4180 s. Shafranovich: Common Format and MIME Type for Comma-Separated Values (CSV) Files, Oktober 2005, URL: <http://tools.ietf.org/html/rfc4180><http://tools.ietf.org/html/rfc4180>

Langzeitarchivierung. Es eignet sich am besten für die Archivierung der Datenbankinhalte und wird daher im Archivierungskonzept verwendet.

In Abbildung II-4 werden die wesentlichen Vor- und Nachteile der potentiellen Formate für die Archivierung der Datenbankstruktur und Datenbankdaten gegenübergestellt. Eine detailliertere Übersicht liegt als Anhang A-6 vor.

SQL	CSV	XML-basiertes Format
<ul style="list-style-type: none"> <li>+ internationaler Standard</li> <li>+ Basis der meisten RDBS</li> <li>+ deskriptiv</li> </ul>	<ul style="list-style-type: none"> <li>+ kompaktes Format</li> <li>+ direkter Datenbankexport</li> </ul>	<ul style="list-style-type: none"> <li>+ uneingeschränkte Lesbarkeit</li> <li>+ Systemunabhängigkeit</li> <li>+ einfache Verarbeitung</li> <li>+ Austauschbarkeit</li> <li>+ einfache Speicherung von Daten und Struktur</li> <li>+ deskriptiv</li> </ul>
<ul style="list-style-type: none"> <li>– Verwendung unterschiedlicher SQL-Dialekte in den einzelnen RDBS</li> <li>– bei der Übernahme: Konversion in standardkonformes SQL erforderlich (Systemabhängigkeit)</li> <li>– bei der Bereitstellung: Anpassung an den SQL-Dialekt des RDBS notwendig (Systemabhängigkeit)</li> <li>– eingeschränkte Lesbarkeit</li> <li>– aufwändige Verarbeitung</li> <li>– eingeschränkte Austauschbarkeit</li> </ul>	<ul style="list-style-type: none"> <li>– kein Standard</li> <li>– Trennung von Daten und Metadaten</li> <li>– eingeschränkte Austauschbarkeit</li> <li>– mögliche Einschränkung der Lesbarkeit</li> <li>– mögliche Systemabhängigkeit</li> <li>– keine Speicherung der Tabellenverknüpfungen und Feldbeschreibung</li> </ul>	<ul style="list-style-type: none"> <li>– kein Standard</li> </ul>

**Abb. II-4:** Vergleich von Archivformaten für Datenbankstruktur und Datenbankdaten

#### **1.1.2.1.2 Datenbankabfragen**

Für die Archivierung von Datenbankabfragen kommen SQL-, XQuery und ein spezifisches, XML-basiertes Format in Frage.

Die im vorherigen Abschnitt aufgeführten charakteristischen Merkmale von SQL haben im Bezug auf die Archivierung von Datenbankabfragen die gleichen Auswirkungen. Aufgrund der Systemabhängigkeit, der eingeschränkten Lesbarkeit und der aufwändigen Verarbeitung ist SQL als Format für die Sicherung von Abfragen nicht geeignet.

XQuery ist ein XML-basiertes Standardformat. Es ist systemunabhängig und ermöglicht die direkte Suche in den gespeicherten XML-Daten. Eine vorab durchzuführende Transformation der XQuery-Anweisungen ist dafür nicht erforderlich. Das ist ein großer Vorteil, da sich XQuery-Transformationen in beide Richtungen sehr aufwändig gestalten.

XQuery ist keine rein deskriptive Abfragesprache. Neben den Suchinhalten beschreibt das Format die auszuführenden Suchalgorithmen und ist demzufolge nur eingeschränkt lesbar.

Die Komplexität des XQuery-Standards führt zu einem hohen Aufwand bei der Verarbeitung der Befehle.

XQuery ist aufgrund der eingeschränkten Lesbarkeit und aufwändigen Verarbeitung nur bedingt als Archivformat für Datenbankabfragen geeignet.

Ein spezifisches, XML-basiertes Format für die Speicherung von Abfragen ist, wie bei der Struktur und den Daten, durch Lesbarkeit, Systemunabhängigkeit und einfache Verarbeitung gekennzeichnet.

Die direkte Suche in den archivierten XML-Daten ist ohne Transformation des Abfrageformates in SQL oder XQuery nicht möglich. Allerdings ist das Format bei der Bereitstellung an keine bestimmte Abfragesprache gebunden. SQL, XQuery oder andere Sprachen können relativ problemlos aus dem XML-basierten Format generiert werden.

Ein spezifisches, XML-basiertes Format erfüllt alle Formatanforderungen der digitalen Langzeitarchivierung und zeichnet sich zudem durch Flexibilität im Bereitstellungsprozess aus. Es ist am besten für die Archivierung von Datenbankabfragen geeignet und findet im Archivierungskonzept Anwendung.

Die Vor- und Nachteile der drei potentiellen Formate für die Abfragenspeicherung werden in der Übersicht in Abbildung II-5 zusammengefasst. Eine detailliertere Gegenüberstellung der Formate befindet sich im Anhang A-7.

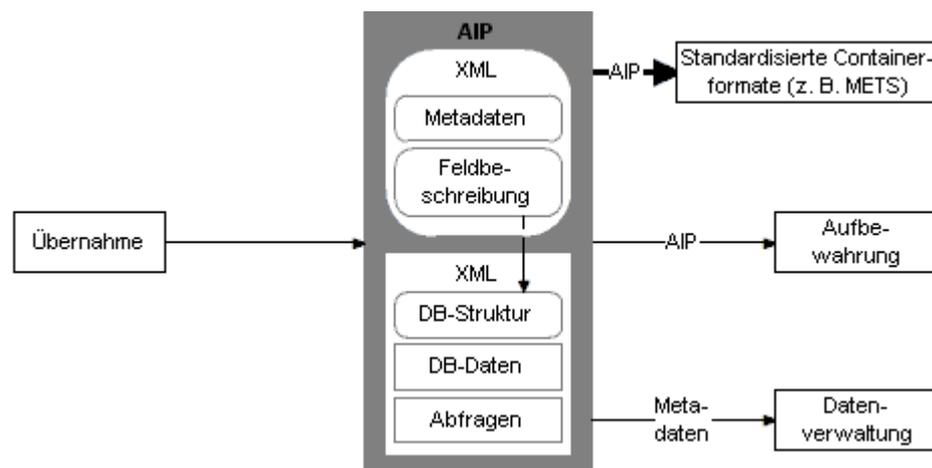
SQL	XQuery	XML-basiertes Format
<ul style="list-style-type: none"> <li>+ internationaler Standard</li> <li>+ Basis der meisten RDBS</li> <li>+ deskriptiv</li> </ul>	<ul style="list-style-type: none"> <li>+ internationaler Standard</li> <li>+ XML-basiert</li> <li>+ Systemunabhängigkeit</li> <li>+ direkte Suche in den Archivdaten ohne Transformation</li> <li>+ Austauschbarkeit</li> </ul>	<ul style="list-style-type: none"> <li>+ uneingeschränkte Lesbarkeit</li> <li>+ Systemunabhängigkeit</li> <li>+ einfache Verarbeitung</li> <li>+ Bei der Bereitstellung: Keine Bindung an eine bestimmte Abfragesprache (Flexibilität)</li> <li>+ Austauschbarkeit</li> <li>+ deskriptiv</li> </ul>
<ul style="list-style-type: none"> <li>– Verwendung unterschiedlicher SQL-Dialekte in den einzelnen RDBS</li> <li>– eingeschränkte Lesbarkeit</li> <li>– aufwändige Verarbeitung</li> <li>– bei der Übernahme: Konversion in standardkonformes SQL erforderlich (Systemabhängigkeit)</li> <li>– bei der Bereitstellung: Anpassung an den SQL-Dialekt des RDBS notwendig (Systemabhängigkeit)</li> <li>– eingeschränkte Austauschbarkeit</li> </ul>	<ul style="list-style-type: none"> <li>– eingeschränkte Lesbarkeit</li> <li>– erschwerte Verarbeitung</li> <li>– aufwändige Transformation (SQL ↔ XQuery)</li> </ul>	<ul style="list-style-type: none"> <li>– kein Standard</li> <li>– keine direkte Suche in den archivierten XML-Daten ohne Transformation</li> </ul>

**Abb. II-5:** Vergleich von Archivformaten für Datenbankabfragen

### 1.1.2.2 Paketaufbau

Sowohl die Datenbankstruktur, die Datenbankdaten als auch die Abfragen werden im Archivpaket im XML-Format gespeichert.

Abbildung II-6 veranschaulicht den Aufbau des Archivpaketes, das während des Übernahmeprozesses generiert wird.



**Abb. II-6:** Archivpaket

Für die Sicherung der Abfragen wird ein XML-basiertes Archivformat verwendet, das durch eine in dieser Arbeit entwickelte Dokumenttypdefinition beschrieben wird. Die Typdefinition und die sich daraus ergebende Formatstruktur werden im folgenden Teil III dieser Arbeit detailliert vorgestellt.

Bei der Archivierung der Datenbankstruktur und Datenbankdaten findet das im Rahmen des RODA-Projektes<sup>37</sup> entwickelte XML-Format Verwendung, das mit Hilfe des als Anhang A-8.1 enthaltenen XML-Schemas definiert wird. Das XML-Dokument ist durch die Elemente STRUCTURE und DATA zweigeteilt. Die Datenbankstruktur wird in Form der Tabellen, der jeweiligen Spalten und ihrer Eigenschaften sowie der Fremd- und Primärschlüssel beschrieben.

Abbildung II-7 zeigt, wie die genannten Strukturelemente in XML umgesetzt werden.

---

<sup>37</sup> Für weitere Informationen s. Ramalho [u. a.]: Relational Database Preservation through XML modelling, 2007

<b>Tabelle</b>	Element TABLE mit dem Attribut NAME
<b>Spalte</b>	Element COLUMN (gruppiert im übergeordneten Element COLUMNS) mit dem Attribut NAME
<b>Spalteneigenschaften</b>	Attribute des COLUMN-Elements, z. B. TYPE, NULL
<b>Primär- und Fremdschlüssel</b>	Element PKEY mit dem Attribut TYPE Element FKEY mit den Attributen IN (Verweis auf Zieltabelle) und REF (Verweis auf Zieldatensätze)

**Abb. II-7:** Umsetzung der Datenbankstruktur mit Hilfe von XML

Die Datenbankdaten werden als Elementinhalte dargestellt. Für jede Tabelle wird ein Element mit dem entsprechenden Tabellennamen, für jede Tabellenzeile ein weiteres Element mit dem Tabellennamen und der Nachsilbe -REG und für jede Spaltenspalte ein Element mit dem entsprechenden Spaltennamen erstellt.

Der Ausschnitt einer XML-Archivdatei mit den Struktur- und Dateninformationen ist in Abbildung II-8 zu sehen. Das vollständige XML-Dokument befindet sich im Anhang A-8.2 zu finden.

```

<?xml version="1.0" ?>
<DB>
  <STRUCTURE>
    <TABLE NAME="tabelle1" DESCRIPTION="Person">
      <COLUMNS>
        <COLUMN NAME="feld1" TYPE="int" SIZE="11" NULL="no"
          DESCRIPTION="Personennummer"/>
        ...
      </COLUMNS>
      <KEYS>
        <PKEY TYPE="simple">
          <FIELD NAME="feld1"/>
        </PKEY>
      </KEYS>
    </TABLE>
    ...
  </STRUCTURE>
  <DATA>
    <tabelle1>
      <tabelle1-REG>
        <feld1>1</feld1>
        ...
      </tabelle1-REG>
      ...
    </tabelle1>
  </DATA>
</DB>

```

**Abb. II-8:** XML-Archivdatei mit Datenbankstruktur und Datenbankdaten (gekürzt)

Für die Speicherung der Felddescription wird das beschriebene XML-Format für Datenbankstruktur und Datenbankdaten erweitert. Den Elementen TABLE und COLUMN wird das Attribut DESCRIPTION zugeordnet, das den Klartext zu den jeweiligen Tabellen- und Spaltenbezeichnungen enthält.

Die im Einlieferungspaket vorhandenen Metadaten werden während des Übernahmeprozesses ergänzt und müssen im Archivpaket vollständig vorliegen. Die Speicherung der Metadaten erfolgt im XML-Format. Folgende Arten von Metadaten sind unbedingt notwendig:

- beschreibende,
- technische und
- rechtliche Metadaten sowie
- Metadaten zum Archivierungsprozess.

Eine genaue Aufstellung der erforderlichen Metadaten kann der SIARD-Formatbeschreibung entnommen werden<sup>38</sup>.

Für die Speicherung der genannten Komponenten des Archivpaketes wird ein Containerformat verwendet. Eine Standardisierung ist an dieser Stelle, vor allem im Hinblick auf den Datenaustausch mit anderen Systemen, anzustreben. Das Modell stellt eine Schnittstelle bereit, die den Export des Archivpaketes in gängige Standards, wie METS, ermöglicht.

Das gesamte Archivpaket wird nach seiner Generierung an die Aufbewahrungseinheit übergeben. Die im Paket enthaltenen Metadaten werden kopiert und an die Datenverwaltung weitergeleitet.

### **1.1.3 Auslieferungspaket**

Das Auslieferungspaket wird in Folge von Nutzeranfragen generiert und für die Nutzung bereitgestellt. Es ist in seinem Aufbau an den Nutzerbedürfnissen orientiert.

Das Paket kann aufgrund der gewählten Archivformate flexibel gestaltet werden. Sowohl bei der Datenbankstruktur, den Datenbankdaten als auch den Abfragen liegt

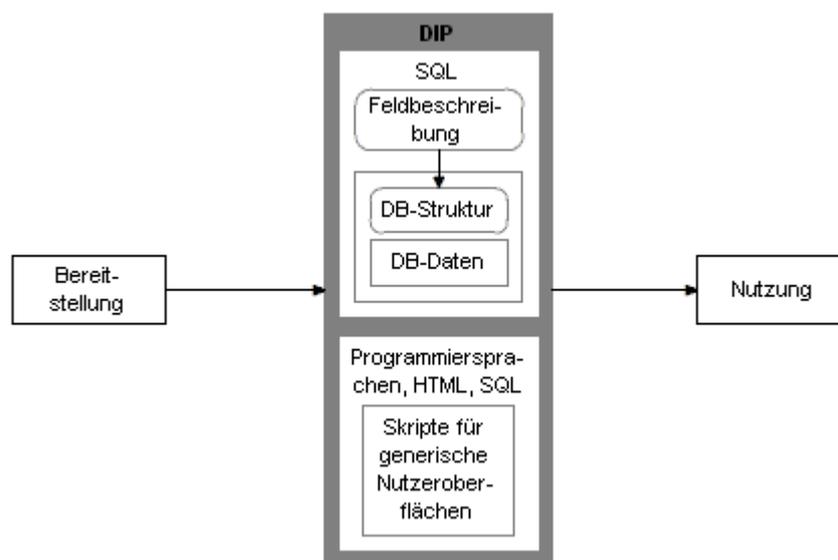
---

<sup>38</sup> s. dazu Schweizerisches Bundesarchiv BAR: SIARD : Formatbeschreibung, 2008

keinerlei Bindung an ein bestimmtes Ausgabeformat oder eine bestimmte Abfragesprache vor.

In den folgenden Abschnitten werden drei Varianten für den Aufbau des Auslieferungspaketes vorgestellt.

### 1.1.3.1 Variante 1: SQL und generische Nutzeroberflächen



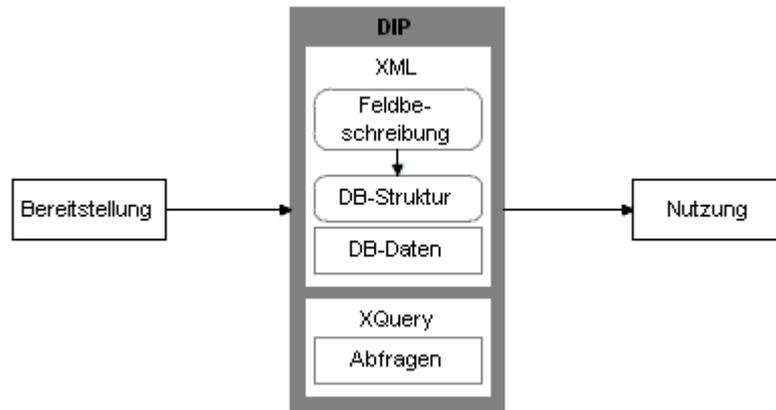
**Abb. II-9:** Auslieferungspaket (Variante 1)

Wie in Abbildung II-9 dargestellt, liegen bei der ersten Variante die Datenbankstruktur und Datenbankdaten sowie die Feldbeschreibung im SQL-Format vor. Die Feldbeschreibung ist in Form von SQL-Spaltenbezeichnungen oder Kommentarspalten in der Datenbankstruktur enthalten.

Die archivierten Abfragen werden in Form von Programmskripten für die Generierung von Nutzeroberflächen bereitgestellt. In den Skripten finden Programmiersprachen, HTML- und SQL-Anweisungen Verwendung.

### 1.1.3.2 Variante 2: XML und XQuery

Abbildung II-10 zeigt den Aufbau des Auslieferungspaketes der zweiten Variante.

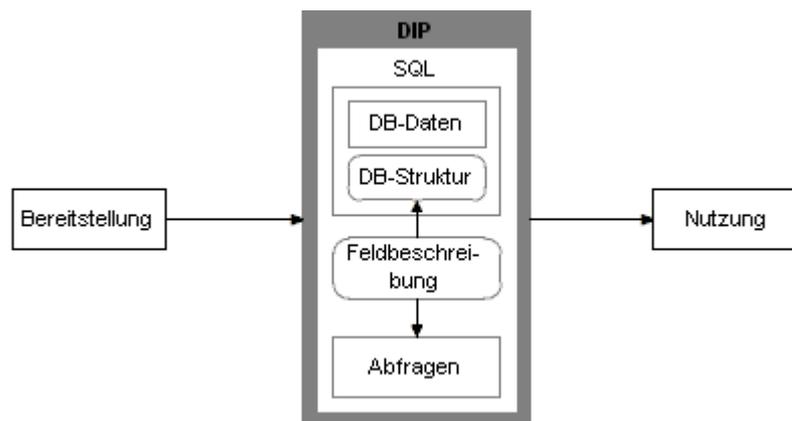


**Abb. II-10:** Auslieferungspaket (Variante 2)

Die Datenbankstruktur und Datenbankdaten sowie die Feldbeschreibung werden ohne Änderungen aus dem Archivpaket im XML-Format übernommen.

Die Datenbankabfragen werden im Auslieferungspaket in der Abfragesprache XQuery bereitgestellt.

### 1.1.3.3 Variante 3: SQL



**Abb. II-11:** Auslieferungspaket (Variante 3)

Der in Abbildung II-11 dargestellte Aufbau des Auslieferungspaketes zeigt, dass bei Variante 3 sowohl die Datenbankstruktur, die Datenbankdaten, die Abfragen als

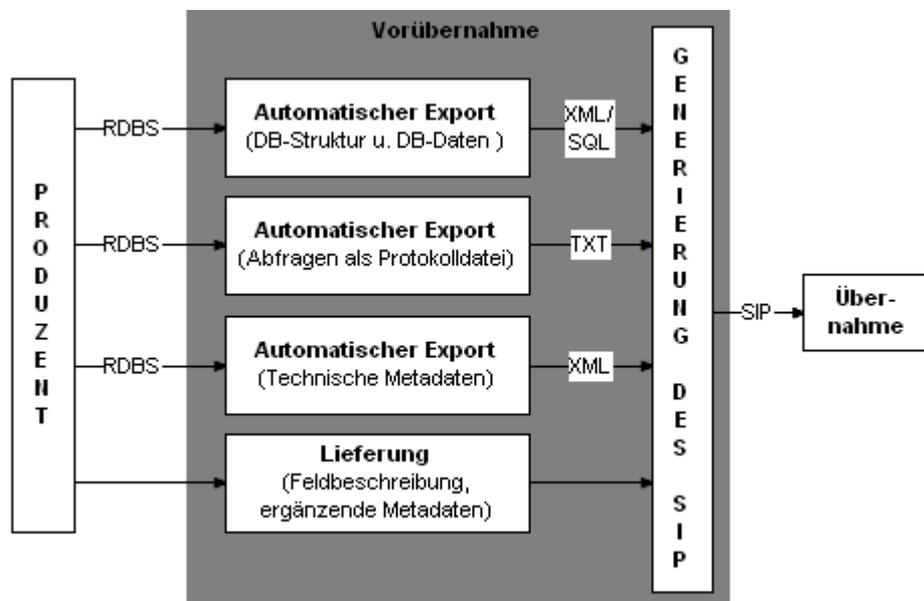
auch die Feldbeschreibung im SQL-Format vorliegen und durch die entsprechenden Befehle dargestellt werden. Die Feldbeschreibung wird mit Hilfe von SQL-Spaltenbezeichnungen, Aliasnamen oder Kommentarspalten umgesetzt. Es wird ausschließlich standardkonformes SQL verwendet.

## 1.2 Prozessmodell

Nachfolgend werden die Prozesse innerhalb der Funktionseinheiten Übernahme und Bereitstellung vorgestellt und die Vorgänge während der Vorübernahme und der Nutzung betrachtet. Auf die Funktionseinheiten Datenverwaltung, Aufbewahrung, Systemverwaltung und Planung der Langzeitarchivierung wird nicht genauer eingegangen.

Detaillierte Prozessmodelle sind im Anhang A-9 zu sehen.

### 1.2.1 Vorübernahme



**Abb. II-12:** Prozessmodell Vorübernahme

Die in Abbildung II-12 dargestellten Prozesse der Vorübernahme umfassen die wesentlichen Vorbereitungen für die Übernahme der Datenbank und der Metadaten in das Archiv. Unter Berücksichtigung der Übernahmeanforderungen, die zuvor zwi-

schen dem Produzenten und dem Archiv vereinbart wurden, wird das Einlieferungspaket generiert.

Die Datenbankstruktur und Datenbankdaten, die Abfragen als Protokolldatei sowie die technischen Metadaten können automatisch aus dem Datenbanksystem exportiert werden.

Die Feldbeschreibung und weitere Metadaten werden vom Produzenten geliefert und müssen, abhängig vom Lieferformat, gegebenenfalls manuell erfasst werden.

### 1.2.2 Übernahme

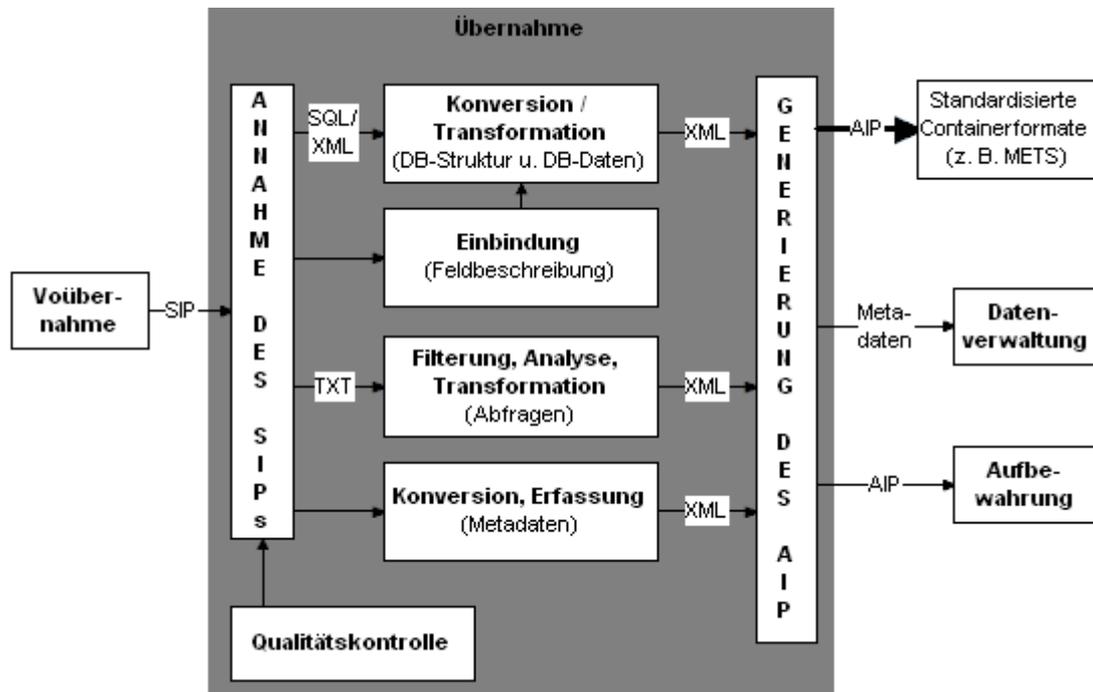


Abb. II-13: Prozessmodell Übernahme

Als erster Schritt wird bei der Übernahme das Einlieferungspaket angenommen. Darauf folgt die Qualitätskontrolle. Es wird überprüft, ob das Einlieferungspaket und die darin enthaltenen Komponenten den vereinbarten Übernahmeanforderungen entsprechen. Ist die Kontrolle erfolgreich, und liegen keine negativen Ergebnisse vor, laufen die Übernahmeprozesse ab, die in Abbildung II-13 zu sehen sind.

Die Datenbankstruktur und Datenbankdaten werden, abhängig vom Lieferformat, konvertiert oder transformiert. Liegen sie im SQL-Format vor, ist eine vollständige Transformation in das beschriebene XML-Archivformat notwendig. Werden die Struktur und die Daten bereits in XML geliefert, muss lediglich konvertiert werden.

Die Felddescription ist im Einlieferungspaket in elektronischer Form vorhanden und wird während der Übernahme mit Hilfe des Attributes DESCRIPTION an den jeweiligen Stellen des XML-Archivformates für die Struktur und die Daten eingebunden.

Das gelieferte Abfragenprotokoll enthält neben Informationen zu den Nutzern und deren An- und Abmeldungen alle SQL-Befehle, die das gesamte Datenbanksystem verarbeitet. Für die Archivierung werden nur die Suchanfragen benötigt, die an eine bestimmte Datenbank gestellt wurden. Als erster Schritt der Abfragenübernahme werden daher die relevanten SELECT-Anweisungen aus der Protokolldatei datenbankspezifischen herausgefiltert. Die als Ergebnis ausgegebene SQL-Datei wird anschließend einer Analyse unterzogen, bei der redundante Abfragen zusammengefasst werden. Zusätzlich wird die Häufigkeit der Abfragen erfasst, um so eine Aussage über die Abfragenrelevanz zu gewinnen. Diese Häufigkeitsangaben finden bei den Bereitstellungsprozessen weitere Verwendung. Die beschriebene Analyse läuft voll automatisiert ab.

Weitere Maßnahmen während der Übernahme sind die Konversion der gelieferten Metadaten in XML und die Erfassung weiterer Metadaten.

Den Abschluss der Übernahmeprozesse bildet die Generierung des Archivpaketes. Das Archivpaket wird an die Aufbewahrungseinheit, und die Metadaten werden an die Datenverwaltung übergeben.

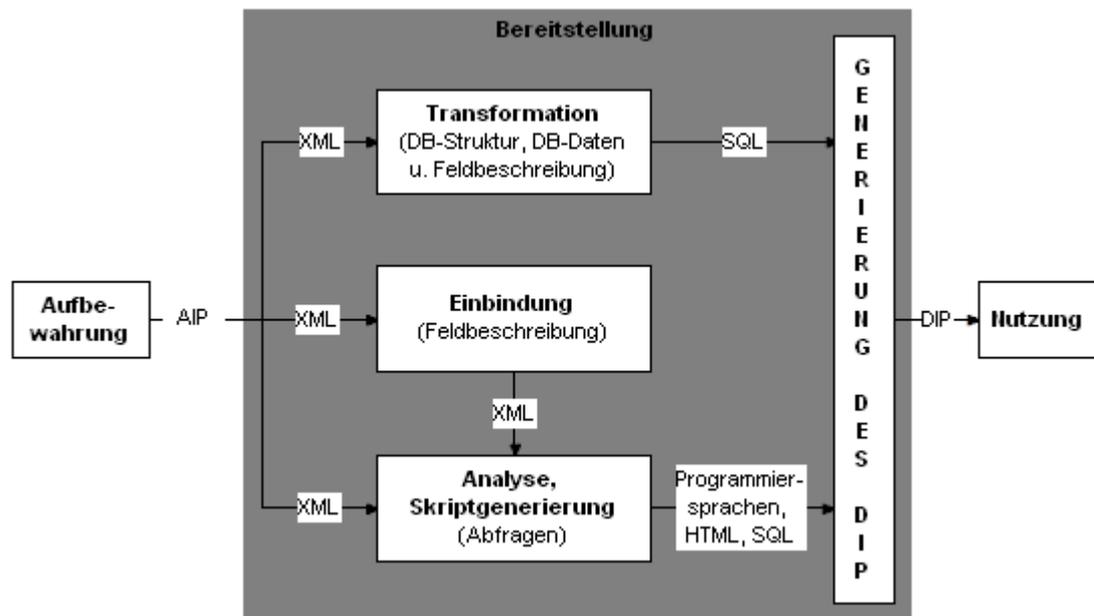
### **1.2.3 Bereitstellung und Nutzung**

Die Bereitstellungseinheit erhält aus dem Archivspeicher die für den Nutzer relevanten Archivpakete, generiert das Auslieferungspaket und stellt dieses anschließend für die Nutzung zur Verfügung.

In den folgenden Abschnitten werden die Prozesse vorgestellt, die für die Generierung der im Kapitel 1.1.3 beschriebenen Auslieferungspakete erforderlich sind. Alle

drei Varianten berücksichtigen die Datenbankstruktur, die Datenbankdaten, die Feldbeschreibung und die Abfragen. Durch die Bereitstellung der Datenbankabfragen wird ein Teil der Tabellenverknüpfung und damit ein wesentlicher Bestandteil der Datenbankstruktur rekonstruiert.

### 1.2.3.1 Variante 1: SQL und generische Nutzeroberflächen



**Abb. II-14:** Prozessmodell Bereitstellung (Variante 1)

Abbildung II-14 zeigt die bei Variante 1 ablaufenden Bereitstellungsprozesse. Die Datenbankstruktur und Datenbankdaten sowie die enthaltene Feldbeschreibung werden mit Hilfe eines Parsers<sup>39</sup> in das SQL-Format transformiert.

Der erste Schritt der Bereitstellung der im Archivpaket enthaltenen Abfragen ist eine Analyse, die sich in folgende drei Teilprozesse gliedert:

1. Verallgemeinerung: Es wird auf spezielle Suchparameter verzichtet. Die Parameter werden durch die jeweiligen Spaltenbezeichnungen ersetzt.

Dieser Vorgang bildet die Grundlage für die Rekonstruktion der generischen Datenbankfunktionalität.

---

<sup>39</sup> s. dazu Glossar

2. Häufigkeitsanalyse: Abfragen, die häufig vorkommen, sind wichtige Abfragen im Nutzerverhalten und werden deshalb in ihrer ursprünglichen Form beibehalten. Diese Teilanalyse ermöglicht die Rekonstruktion des Nutzerverhaltens.
3. Strukturanalyse: Es werden die in den Abfragen verwendeten Tabellen und Spalten ermittelt. Für die eindeutigen Tabellen- und Spaltenbezeichnungen wird auf die Feldbeschreibung zugegriffen. Dieses Vorgehen stellt sicher, dass später die Suchparameter der Nutzeroberflächen verständlich sind. Mit Hilfe der Funktionsbibliothek einer externen SQL-Referenz werden in den Abfragen enthaltene Datenbankfunktionen erkannt.

Die Strukturanalyse ist Voraussetzung für die Wiederherstellung der ursprünglichen Datenbankfunktionalität.

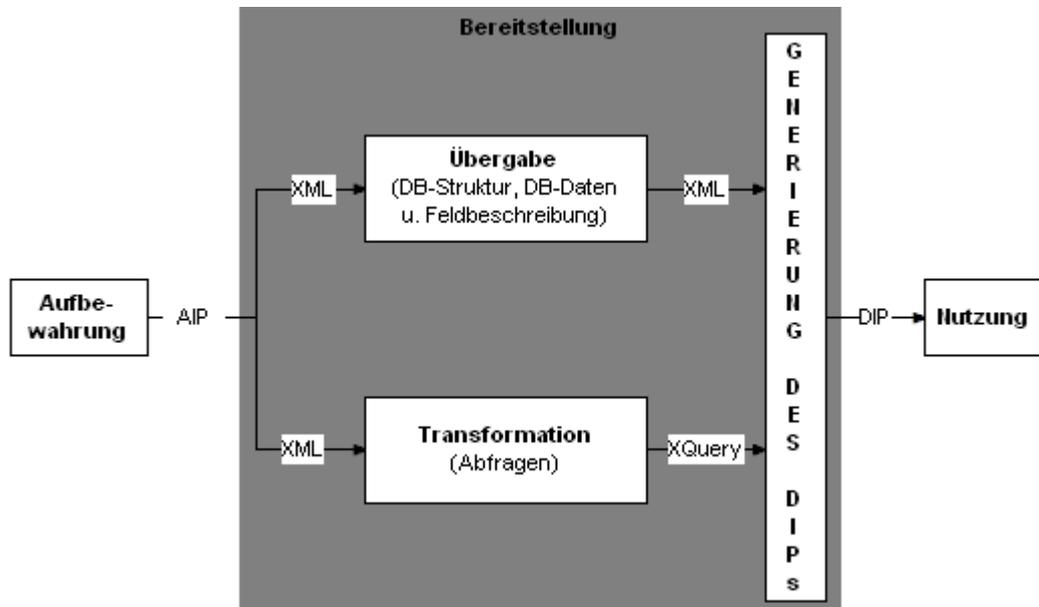
Auf Grundlage der Analyseergebnisse werden die Programmskripte für die späteren Nutzeroberflächen erstellt. Über eine Schnittstelle können die Ergebnisse in XML exportiert werden.

Im Rahmen der Nutzung der bereitgestellten Archivinhalte werden die Struktur und die Daten der Datenbank in ein relationales Datenbanksystem importiert, das sich auf einem SQL-Server befindet.

Die Programmskripte finden bei der Erstellung der Oberflächen für die Suche und die Ergebnisausgabe Anwendung. Den ersten Schritt bildet die Generierung der Suchoberfläche. Mit Hilfe von SQL werden die möglichen Suchparameter in der wiederhergestellten, relationalen Datenbank abgefragt. Über die Suchoberfläche formuliert der Nutzer, durch Auswahl der gewünschten Parameter, die Abfrage. Abschließend wird die Ergebnisoberfläche erzeugt. Bei diesem Vorgang findet ebenfalls ein Austausch mit dem relationalen Datenbanksystem statt.

Mit dieser Variante wird ein Teil des Nutzerverhaltens rekonstruiert. Durch die Datenbanksuche können die wesentlichen Aspekte der Datenbankfunktionalität, die in den Skripten der Anwendungsprogramme programmiert wurden, ebenfalls wiederhergestellt werden. Ein Nachteil dieser Lösung besteht darin, dass die Nutzung der archivierten Datenbank an ein relationales Datenbanksystem gebunden ist.

### 1.2.3.2 Variante 2: XML und XQuery



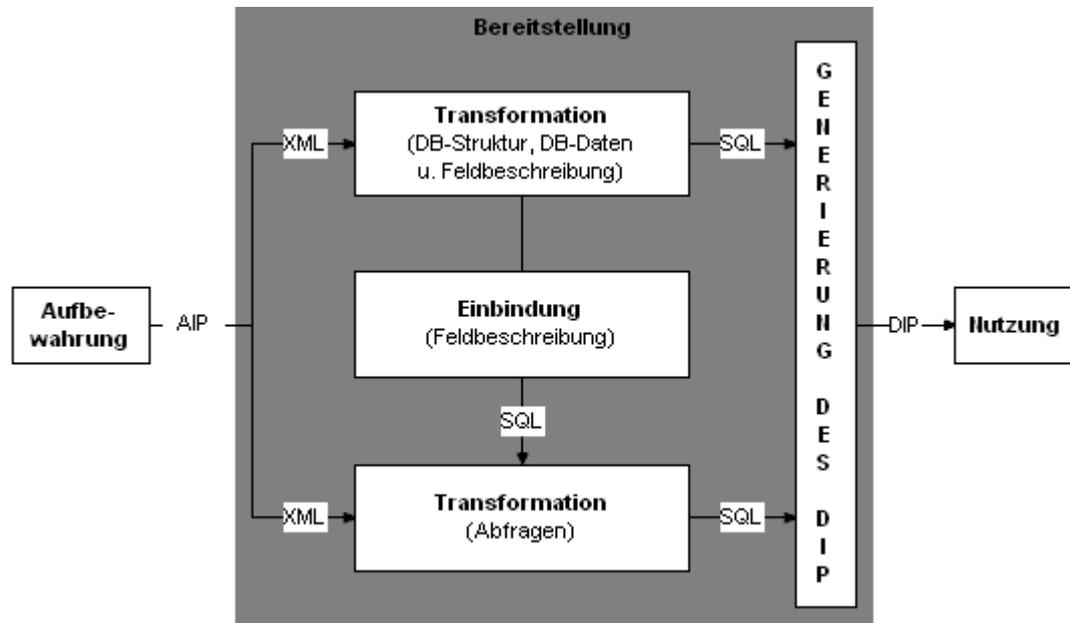
**Abb. II-15:** Prozessmodell Bereitstellung (Variante 2)

Wie in Abbildung II-15 dargestellt, werden bei Variante 2 sowohl die Datenbankstruktur, die Datenbankdaten als auch die integrierte Feldbeschreibung ohne Änderungen zur Verfügung gestellt. Es erfolgt lediglich eine Transformation der Abfragen von XML nach XQuery.

Für die Nutzung des Datenbestandes werden XML-Datenverwaltungssysteme verwendet. Beispiele für Systeme dieser Art sind relationale Datenbanksysteme mit XML-Unterstützung, native XML-Datenbanksysteme und XQuery-Server.

Diese Variante ermöglicht es, auch die Abfragen und einen Teil des Nutzerverhaltens abzubilden. Der Zugriff auf die Datenbankdaten erfolgt darüber hinaus systemunabhängig.

### 1.2.3.3 Variante 3: SQL



**Abb. II-16:** Prozessmodell Bereitstellung (Variante 3)

Abbildung II-16 zeigt die Bereitstellungsprozesse der dritten Variante. Sie zeichnet sich dadurch aus, dass sowohl die Datenbankstruktur, die Datenbankdaten, die enthaltenen Feldbeschreibung als auch die Abfragen mit Hilfe eines Parsers nach SQL transformiert werden.

Als weiterer Prozess wird die Feldbeschreibung ebenfalls in die SQL-Dateien, die die Datenbankabfragen enthalten, eingebunden.

Um die in dieser Form bereitgestellte Datenbank wieder nutzen zu können, müssen die Datenbankstruktur und die Datenbankdaten in ein relationales Datenbanksystem eingelesen werden, das sich auf einem SQL-Server befindet. Mit Hilfe der zur Verfügung gestellten SQL-Abfragen kann auf den Datenbestand zugegriffen werden. Die Erstellung von Berichten wird somit beispielsweise möglich.

Diese Bereitstellungs- und Nutzungsvariante gewährleistet ebenfalls die Abbildung eines Teils der Abfragen und das Nutzerverhaltens. Als Nachteil dieser Variante ist zu erwähnen, dass sie bei der Anwendung auf ein relationales Datenbanksystem angewiesen ist.

### III Die XML-Dokumenttypdefinition für Datenbankabfragen

Wie in der Beschreibung des im Konzept verwendeten Archivpaketes<sup>40</sup> bereits erwähnt wurde, findet für die Speicherung der Datenbankabfragen ein XML-basiertes Archivformat Anwendung, das auf einer selbst entwickelten Dokumenttypdefinition basiert und in das die Abfragen bei der Übernahme transformiert werden.

Die Überführung von SQL-Abfragen in andere Formate ist eine Thematik, mit der sich bereits einige Projekte befassen haben. Ziel dieser Projekte war die genaue Abbildung der SQL-Syntax. Aufgrund der hohen Komplexität des SQL-Standards ist der Ansatz einer quasi 1:1-Abbildung mit einer Reihe von Schwierigkeiten und einem hohen Aufwand verbunden.

Beispielhafte Projekte wurden an der Universität „Tecnológico de Monterrey“ in Mexiko<sup>41</sup> und der Universität in Rostock<sup>42</sup> durchgeführt. Den Schwerpunkt der Projektarbeit bildete hier die Transformation von SQL-Abfragen in entsprechende XQuery-Anweisungen.

Das im Archivierungskonzept verwendete Archivformat verfolgt einen neuen Ansatz. Es beschreibt und bewahrt<sup>43</sup> die grundlegenden Elemente einer Datenbankabfrage und damit die wesentlichen Operationen der relationalen Algebra.

Die komplexe SQL-Syntax wird auf Grundlage von XML vereinfacht. Ausgehend von dieser XML-Struktur können leichter als bei den anderen Ansätzen Abfragen in anderen Abfragesprachen wie beispielsweise XQuery generiert werden.

---

<sup>40</sup> s. dazu Kapitel II-1.1.2.2

<sup>41</sup> Für weitere Informationen s. Cartujano Escobar; Espinosa; Lozano: XML Information Retrieval Using SQL2XQuery, 2002, URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.4444&rep=rep1&type=pdf>

<sup>42</sup> Für weitere Informationen s. Jahnkuhn; Bruder; Balouch: Transformation von SQL in XQuery-Anfragen innerhalb föderierter Informationssysteme, 2005, URL: [http://dbs.informatik.uni-halle.de/GvD2005/beitraege/gvd05\\_JahBruBal.pdf](http://dbs.informatik.uni-halle.de/GvD2005/beitraege/gvd05_JahBruBal.pdf)

<sup>43</sup> Der Aspekt der Bewahrung wird durch die Verwendung des XML-Formates berücksichtigt. Die Begründung dafür ist im Kapitel II-1.1.2.1.2 zu finden.

In den folgenden Abschnitten wird die neu entwickelte Dokumenttypdefinition anhand der dort deklarierten Strukturelemente vorgestellt und dessen Anwendung im XML-Archivformat anhand eines Abfragebeispiels veranschaulicht.

## 1 Struktur der Dokumenttypdefinition für Datenbankabfragen

Die Struktur des im Konzept verwendeten Archivformats für Datenbankabfragen wird mit Hilfe einer Dokumenttypdefinition beschrieben. Sie ist im Anhang A-8.3 zu finden.

Die Dokumenttypdefinition orientiert sich in ihrem Aufbau an der SELECT-Anweisung des SQL-Standards. Für die Entwicklung des XML-Schemas wurde auf SQL-Syntaxdiagramme zurückgegriffen. Als Beispiel ist das Diagramm für den Kern des SELECT-Befehls von SQLite in Abbildung III-1 dargestellt. Das Diagramm zeigt die Struktur der SELECT-Anweisung. Durch diesen Aufbau wird vorgegeben, welche grundlegenden Elemente in einer Datenbankabfrage enthalten sein müssen. Das vollständige Syntaxdiagramm ist im Anhang A-10 zu sehen.

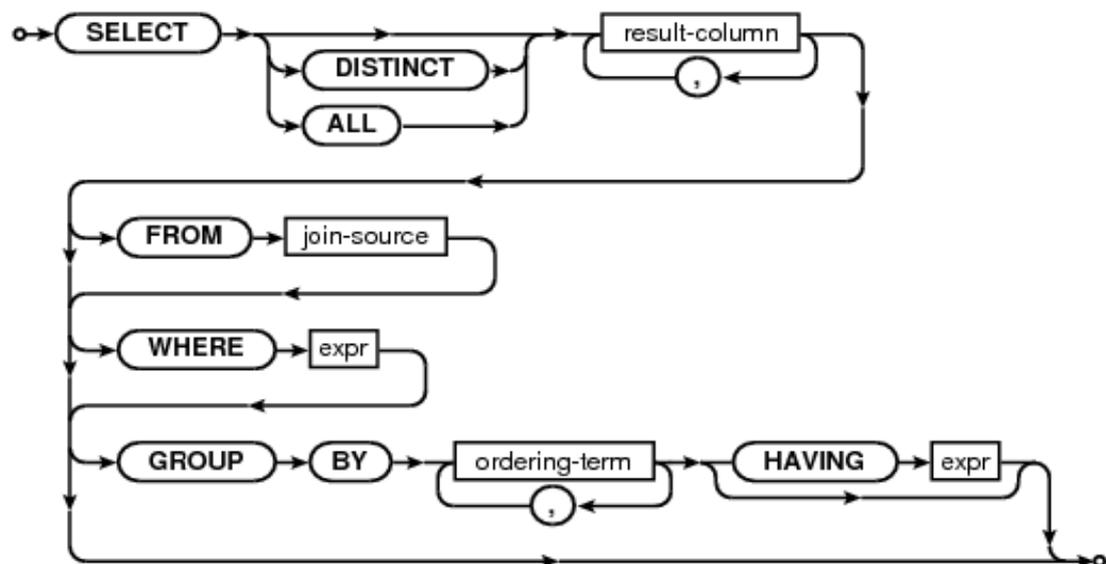


Abb. III-1: SQLite-Syntaxdiagramm des SELECT-Befehls<sup>44</sup>

<sup>44</sup> Quelle: SQLite SQL Language Syntax Diagrams, 09.10.2008, URL: <http://www.sqlite.org/syntaxdiagrams.html#select-core>

Die neu entwickelte Dokumenttypdefinition berücksichtigt folgende, aus der SELECT-Syntax abgeleitete Abfrageelemente:

- Projektion,
- Selektion,
- Tabellenverknüpfung,
- Gruppierung,
- Sortierung und
- Limitierung.

### **1.1 Strukturelemente**

Abbildung III-2 zeigt den Strukturbaum der XML-Dokumenttypdefinition und veranschaulicht die hierarchischen Zusammenhänge der definierten Strukturelemente.

Auf der obersten Ebene der Struktur ist das Wurzelement ABFRAGENSAMMLUNG definiert, das beliebig viele Elemente vom Typ ABFRAGE beinhalten kann.

Das Element ABFRAGE enthält die Unterelemente:

- PROJEKTION,
- SELEKTION,
- VERKNÜPFUNG,
- GRUPPIERUNG,
- SORTIERUNG und
- LIMITIERUNG.

Mit Hilfe dieser sechs Strukturelemente werden alle Abfragen mit den genannten Elementen beschrieben.

In den folgenden Abschnitten werden diese sechs ABFRAGE-Unterelemente einzeln betrachtet. Dabei werden jeweils die Besonderheiten des zugrunde liegenden Abfrageelements erläutert und daraus die Eigenschaften des entsprechenden XML-Strukturelements abgeleitet und beschrieben.

Darüber hinaus wird auf die Elemente AUSDRUCK, TABELLENSPALTE und WERT eingegangen, die mehrfach in der XML-Struktur enthalten sind.

III Die XML-Dokumenttypdefinition für Datenbankabfragen  
 1 Struktur der Dokumenttypdefinition für Datenbankabfragen

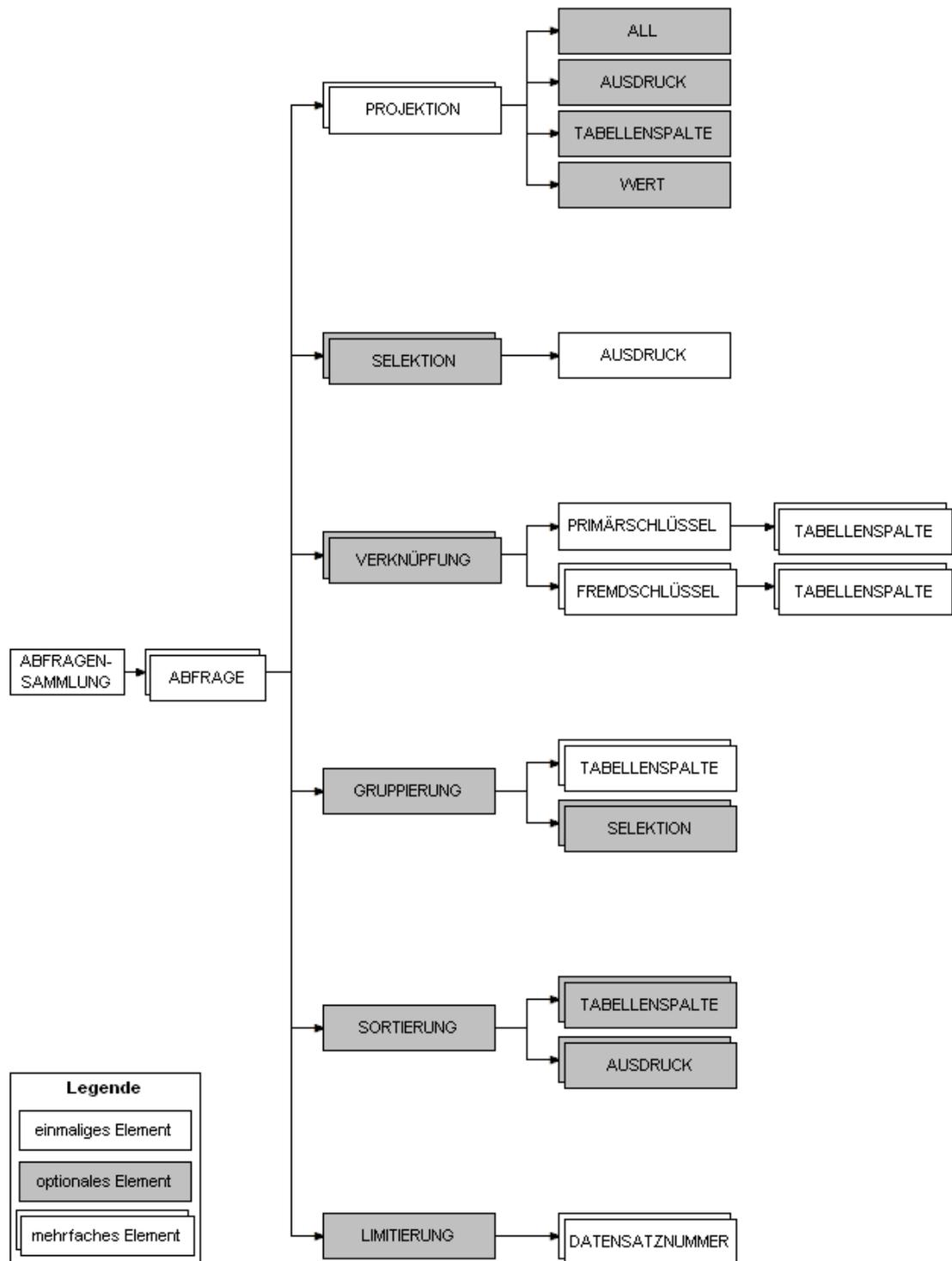


Abb. III-2: Strukturbaum der Dokumenttypdefinition für Datenbankabfragen<sup>45</sup>

<sup>45</sup> Ein detaillierter Strukturbaum mit zusätzlichen Angaben zu den Attributen ist im Anhang A-11 zu sehen.

### 1.1.1 AUSDRUCK

Das Element AUSDRUCK beschreibt Datenbankoperationen und -funktionen.

Eine Operation besteht dabei immer aus Operanden und Operatoren.

Als Operanden sind Tabellenspalten, Ausdrücke und Werte möglich. Mindestens ein Operand muss in einer Operation enthalten sein. Die Vergleichsoperation stellt eine Ausnahme dar. Sie beinhaltet immer zwei Operanden.

Die verwendeten Operatoren zeichnen sich durch unterschiedliche Bedeutungen und Prioritäten aus. Aufgrund dessen sind die richtige Reihenfolge der Operanden und die notwendige Klammersetzung bei zusammengesetzten Operationen zu beachten.

Eine Operation kann beispielsweise in folgenden Formen vorliegen:

- Rechenoperation ( $1 + 2$ ,  $-(+4)$ ,  $2 * 3$ ) und
- Vergleichsoperation ( $A = B$ ,  $B > C$ ,  $C \leq D$ ).

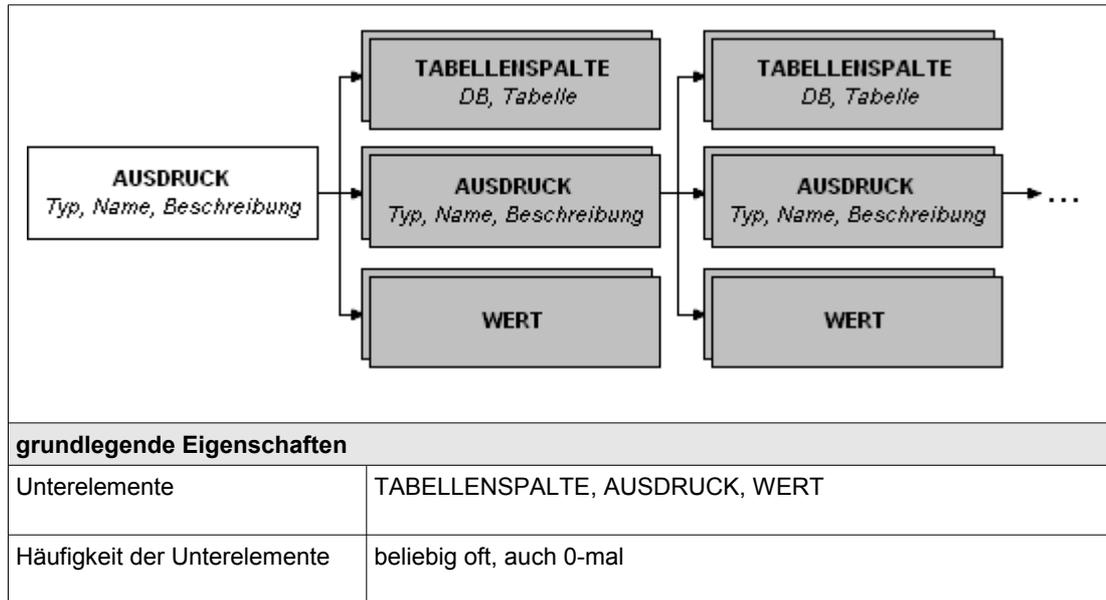
Eine Datenbankfunktion vereinigt mehrere Operationen.

Sie besteht aus einer Liste von Parametern, deren Reihenfolge durch die Funktionssyntax vorgegeben ist und beachtet werden muss. Die Parameter können durch Tabellenspalten, Ausdrücke und Werte gebildet werden. Die Parameterliste kann sowohl keinen als auch beliebig viele Einträge beinhalten.

Grundlegend können folgenden Funktionsgruppen unterschieden werden:

- mathematische Funktionen (z. B. `round()`),
- statistische Funktionen (z. B. `avg()`),
- Zeichenkettenfunktionen (z. B. `concat()`) sowie
- Datums- und Zeitfunktionen (z. B. `now()`).

In Abbildung III-3 wird die Struktur des Elements AUSDRUCK dargestellt, die sich aus den Eigenschaften von Operationen und Funktionen ergibt.



**Abb. III-3:** Struktur und Eigenschaften des Elements AUSDRUCK

Das Element AUSDRUCK kann als Unterelement von PROJEKTION, SELEKTION und SORTIERUNG verwendet werden.

Die Operanden und Funktionsparameter werden durch die Unterelemente repräsentiert. Die Reihenfolge dieser Elemente wird dabei automatisch durch die Reihenfolge der Strukturelemente in der XML-Struktur festgelegt.

Da jedes Element AUSDRUCK wiederum aus einem Unterelement AUSDRUCK bestehen kann, ist die beschriebene Struktur rekursiv.

Das Element enthält folgende Attribute (s. Abbildung III-4).

Attributname	Attributdefinition	Beschreibung
Typ	erforderlich, Aufzählung	Angabe des Ausdruckstyps Zur Auswahl stehen Operation und Funktion.
Name	erforderlich, Zeichenkette	Angabe des verwendeten Operators oder des Funktionsnamens
Beschreibung	erforderlich, Zeichenkette	Spezifikation der Operation (Rechenoperation, Vergleichsoperation ...) nähere Beschreibung von Funktionen mit Hilfe der Funktionsbedeutung (z. B. round(): Rundung)

**Abb. III-4:** Attribute des Elements AUSDRUCK

### III Die XML-Dokumenttypdefinition für Datenbankabfragen

#### 1 Struktur der Dokumenttypdefinition für Datenbankabfragen

Die Struktur des AUSDRUCK-Elements wird nun beispielhaft an folgender Rechenoperation veranschaulicht:

$$1 + 2 * 3 + 4 * 5 * 6 - 7.$$

Diese Operation setzt sich aus Werten und insgesamt vier weiteren, ineinander verschachtelten Operationen zusammen. Unter Beachtung der unterschiedlichen Prioritäten der verwendeten Rechenoperatoren können Klammern gesetzt und die einzelnen Operationen voneinander abgegrenzt werden:

$$(1 + (2 * 3) + (4 * 5 * 6)) - 7.$$

Abbildung III-5 zeigt, wie das Beispiel im Strukturbaum der Dokumenttypdefinition und im XML-Format dargestellt wird.

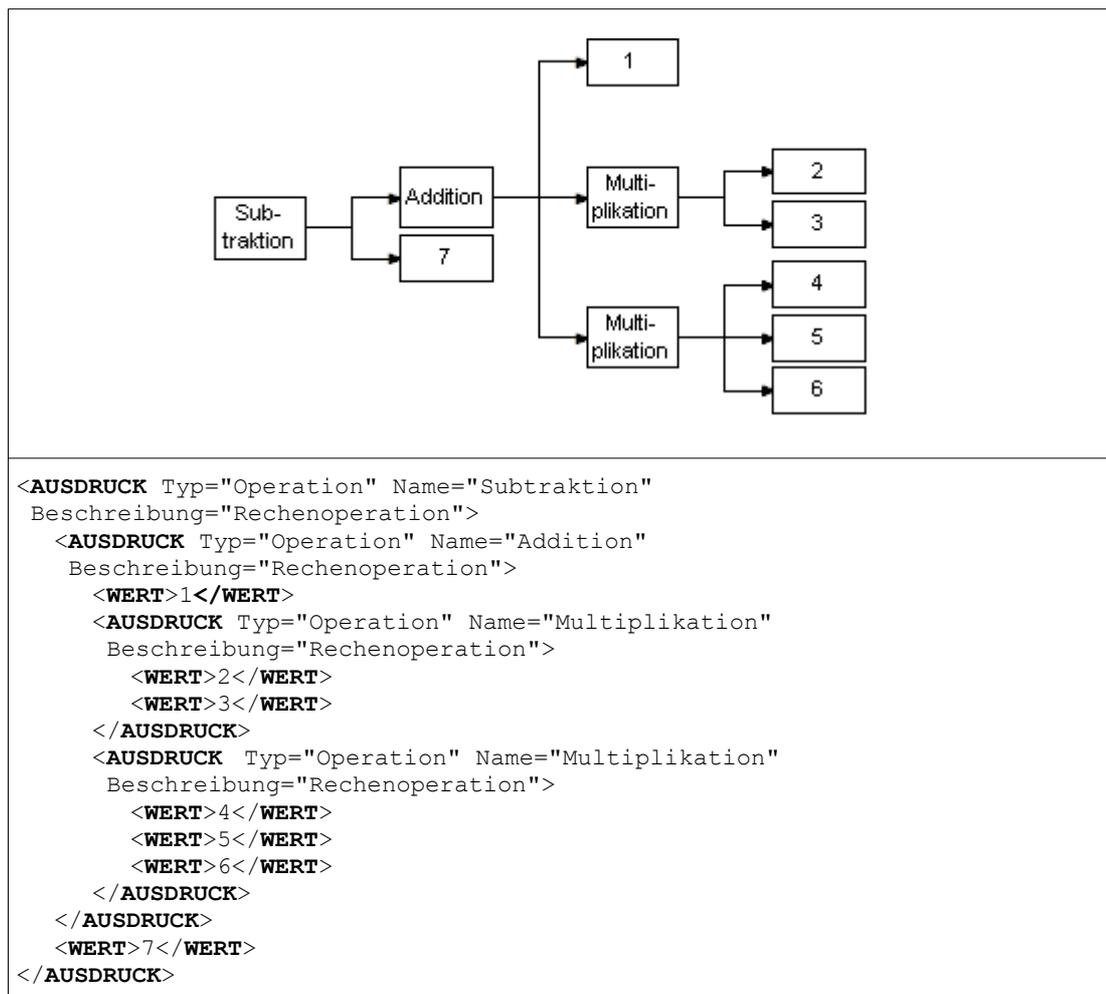


Abb. III-5: Strukturbaum und XML-Code der Beispieloperation

### 1.1.2 TABELLENSPALTE und WERT

Die Elemente TABELLENSPALTE und WERT sind in der Dokumenttypdefinition als optionale Unterelemente von PROJEKTION und AUSDRUCK deklariert. Das Element TABELLENSPALTE kann darüber hinaus den Elementen PRIMÄRSCHLÜSSEL, FREMDSCHLÜSSEL, GRUPPIERUNG und SORTIERUNG zugeordnet werden.

Beide Elemente können Zeichenketten beinhalten.

Das Element TABELLENSPALTE enthält die in der folgenden Übersicht (s. Abbildung III-6) aufgelisteten Attribute.

Attributname	Attributdefinition	Beschreibung
DB	erforderlich, Zeichenkette	Angabe der Datenbank, in der sich die Tabelle mit der Spalte befindet.
Tabelle	erforderlich, Zeichenkette	Angabe der Tabelle, in der sich die Spalte befindet.

**Abb. III-6:** Attribute des Elements TABELLENSPALTE

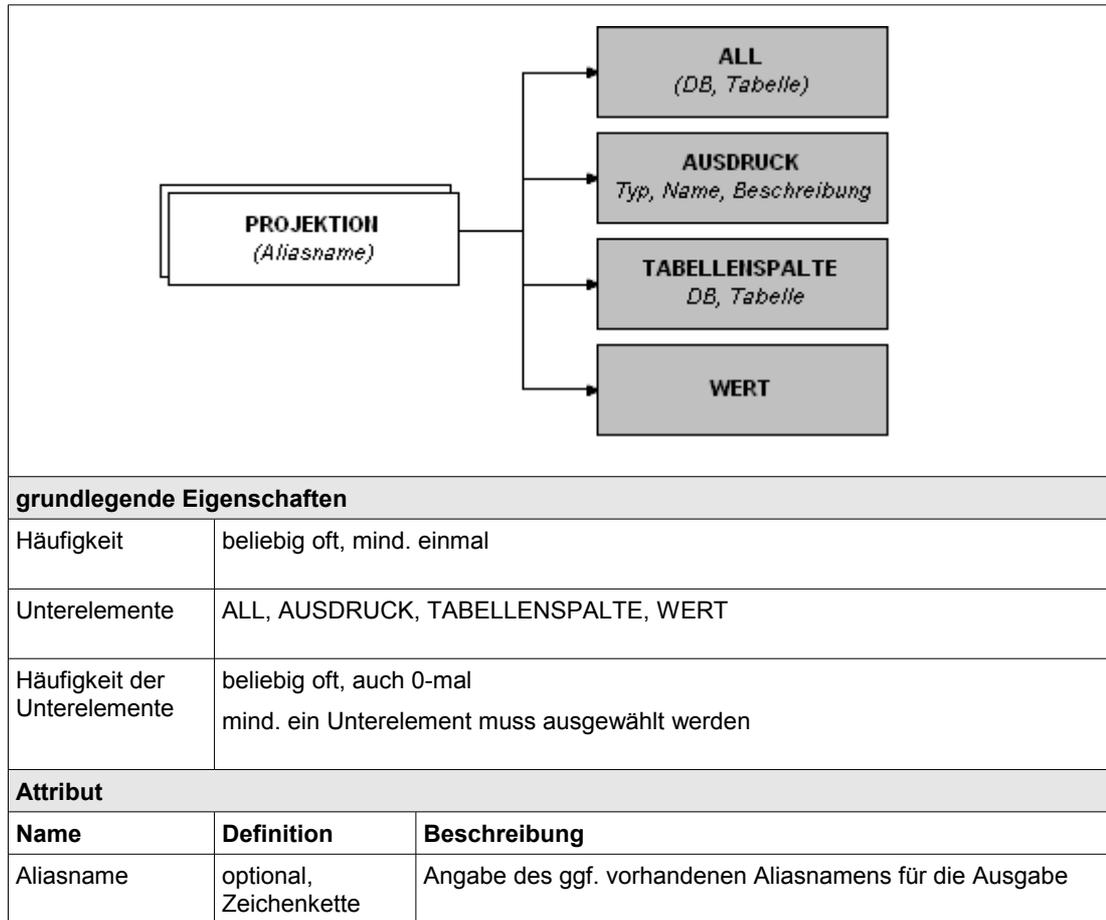
### 1.1.3 PROJEKTION

Das Abfrageelement Projektion ermöglicht die Auswahl der auszugebenden Tabellenspalten und muss mindestens einmal in einer Datenbankabfrage enthalten sein.

Neben einer Tabellenspalte kann eine Projektion auch aus einem Ausdruck oder einem Wert bestehen. Mit dem Zeichen „\*“ können alle Spalten einer Tabelle ausgewählt werden.

Abbildung III-7 enthält den Strukturbaum für das entsprechende Strukturelement PROJEKTION und listet die wesentlichen Elementeigenschaften auf.

Als Unterelement kann unter anderem das bisher noch nicht beschriebene ALL-Element enthalten sein. Es beschreibt ein leeres Element, das bei der Ausgabe aller Tabellenspalten einer Tabelle („\*“) zum Einsatz kommt. Ihm sind äquivalent zu TABELLENSPALTE die Attribute DB und Tabelle zugeordnet. Diese Attribute sind nur erforderlich, wenn durch eine Abfrage mehrere Tabellen miteinander verknüpft werden.



**Abb. III-7:** Struktur, Eigenschaften und Attribut des Elements PROJEKTION

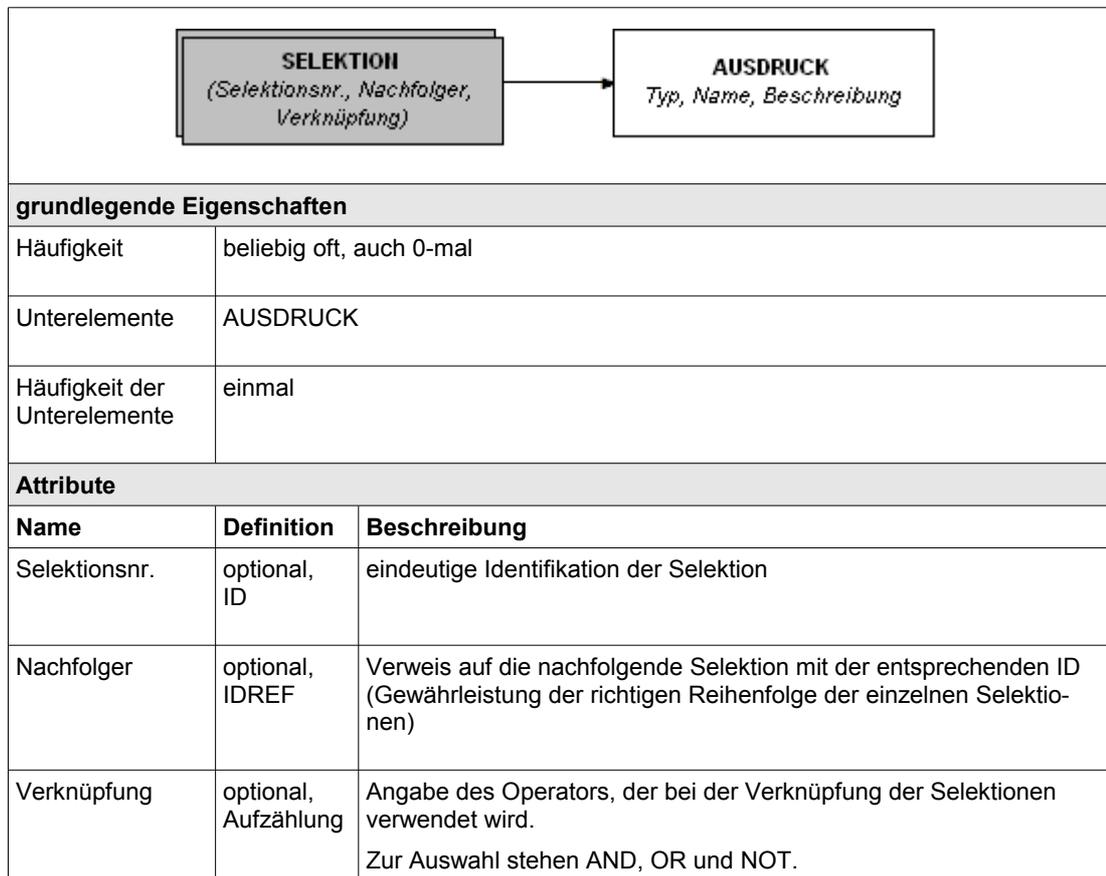
### 1.1.4 SELEKTION

Mit Hilfe des Abfrageelements Selektion werden Bedingungen festgelegt, die dazu führen, dass eine bestimmte Menge von Datensätzen aus einer Tabelle herausgefiltert wird.

Eine Selektion ist kein zwingender Bestandteil einer Datenbankabfrage, kann jedoch mehrfach enthalten sein. Ist dies der Fall, werden die einzelnen Selektionen durch Operatoren miteinander verknüpft. Dabei ist aufgrund der unterschiedlichen Prioritäten der Verknüpfungsoperatoren die Reihenfolge der Selektionen zu beachten.

Da mit einer Selektion eine Bedingung festgelegt wird, besteht sie immer aus einer Vergleichsoperation.

Die abgeleiteten Struktureigenschaften des XML-Elements SELEKTION werden in Abbildung III-8 dargestellt.



**Abb. III-8:** Struktur, Eigenschaften und Attribute des Elements SELEKTION

Das als Unterelement verwendete Element AUSTRUCK muss in jedem Fall durch seine Attribute Typ, Name und Beschreibung als Vergleichsoperation definiert werden.

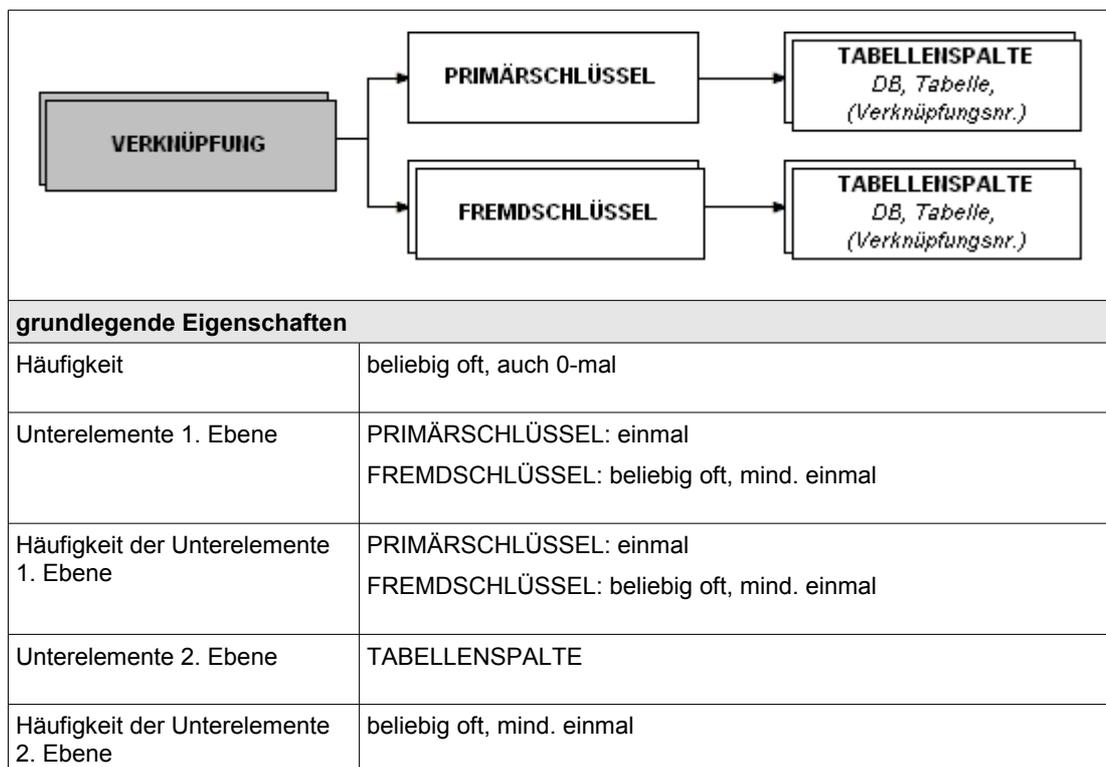
Die in Abbildung III-8 beschriebenen Attribute finden nur Anwendung, wenn das Element SELEKTION mehr als einmal vorhanden ist.

### 1.1.5 VERKNÜPFUNG

Das Element VERKNÜPFUNG beschreibt die in den Abfragen realisierten Tabellenverknüpfungen. Es beinhaltet die Primär- und Fremdschlüssel, über die eine Tabelle mit beliebig vielen anderen Tabellen in Beziehung gesetzt wird.

In einer Tabelle können mehrere Spalten als Primärschlüssel bzw. als Fremdschlüssel definiert sein.

Tabellenverknüpfungen sind als Abfrageelement nicht zwingend erforderlich.



**Abb. III-9:** Struktur und Eigenschaften des Elements VERKNÜPFUNG

Wie in Abbildung III-9 unter dem Punkt Häufigkeit vermerkt, kann das Element VERKNÜPFUNG beliebig oft verwendet werden. Das ist damit zu begründen, dass das Element die Verknüpfungen tabellenweise zusammenfasst und auch mehrere Tabellen über ihre Primärschlüssel mit anderen Tabellen verknüpft werden können.

Das Unterelement FREMDSCHLÜSSEL ist mehr als einmal enthalten, wenn eine Tabelle mit mehr als einer weiteren verknüpft wird.

Dem Unterelement TABELLENSPALTE wird im Kontext der Tabellenverknüpfung ein weiteres Attribut zugeordnet (s. Abbildung III-10).

Attributname	Attributdefinition	Beschreibung
Verknüpfungsnr.	optional, Zeichenkette	Identifikation der Verknüpfung Die für die Verknüpfung verwendeten Spalten enthalten jeweils die gleiche Nummer.

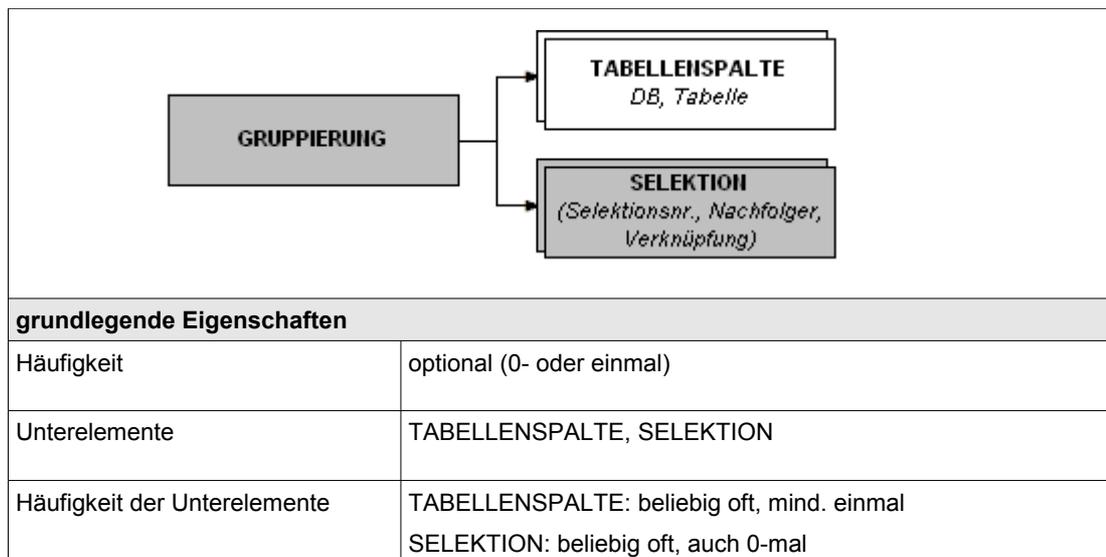
**Abb. III-10:** Ergänzendes Attribut des Elements TABELLENSPALTE

### 1.1.6 GRUPPIERUNG

Das optionale Abfrageelement Gruppierung erlaubt das Zusammenfassen von Datensätzen mit gleichen Einträgen in bestimmten Spalten.

Für die Gruppierung ist zwingend eine Liste mit Spaltennamen erforderlich, die aus mindestens einem Eintrag besteht. Darüber hinaus können Bedingungen festgelegt werden, die die Ergebnismenge der Gruppierung weiter einschränken.

Die Struktur des Elements GRUPPIERUNG ist in Abbildung III-11 zu sehen.



**Abb. III-11:** Struktur und Eigenschaften des Elements GRUPPIERUNG

### 1.1.7 SORTIERUNG

Mit Hilfe des optionalen Abfrageelements Sortierung können die Ergebnisdatensätze anhand von beliebig vielen Spalten oder Ausdrücken auf- und absteigend sortiert ausgegeben werden.

Zu diesem Zweck müssen die Sortierungsrichtung und die entsprechenden Spaltennamen oder Ausdrücke angegeben werden.

Dementsprechend ergeben sich die in Abbildung III-12 gezeigten Struktureigenschaften des XML-Elements SORTIERUNG.

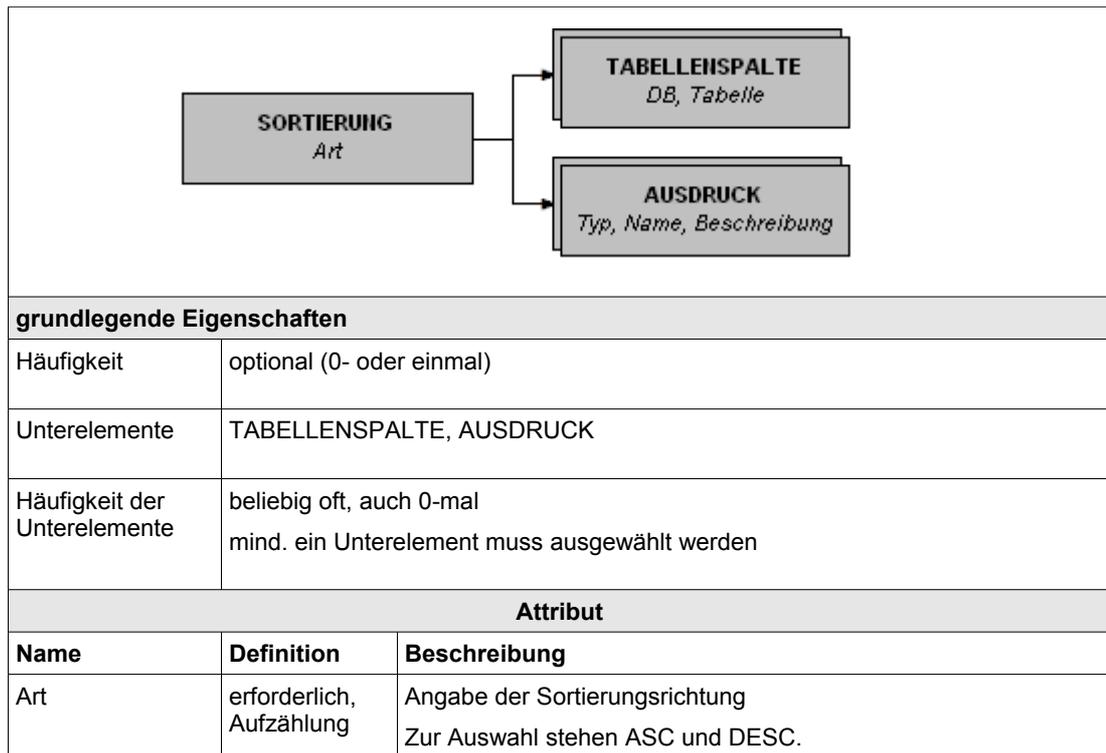
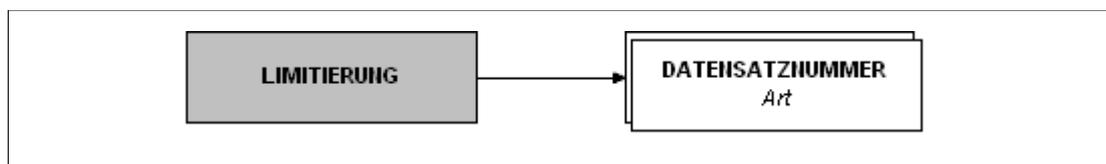


Abb. III-12: Struktur, Eigenschaften und Attribut des Elements SORTIERUNG

### 1.1.8 LIMITIERUNG

Das optionale Abfrageelement Limitierung begrenzt der Anzahl der Datensätze, die als Abfrageergebnis ausgegeben werden sollen. Die Begrenzung erfolgt durch die Angabe des ersten und des letzten anzuzeigenden Datensatzes.

Abbildung III-13 zeigt die Struktur des Elements LIMITIERUNG, die sich aus den Eigenschaften des entsprechenden Abfrageelements ergibt.



### III Die XML-Dokumenttypdefinition für Datenbankabfragen

#### 1 Struktur der Dokumenttypdefinition für Datenbankabfragen

---

grundlegende Eigenschaften	
Häufigkeit	optional (0- oder einmal)
Unterelemente	DATENSATZNUMMER
Häufigkeit der Unterelemente	beliebig oft, mind. einmal

**Abb. III-13:** Struktur und Eigenschaften des Elements *LIMITIERUNG*

Dem Unterelement DATENSATZNUMMER wird das Attribut Art zugeordnet, das in der folgenden Übersicht (s. Abbildung III-14) beschrieben wird.

Attributname	Attributdefinition	Beschreibung
Art	erforderlich, Aufzählung	Angabe der Art des Datensatzes im Kontext der Limitierung Zur Auswahl stehen Start und Ende.

**Abb. III-14:** Attribut des Elements *DATENSATZNUMMER*

## 2 Abfragebeispiel

Anhand der folgenden SQL-Abfrage wird in diesem Abschnitt die durch die Dokumenttypdefinition beschriebene Formatstruktur beispielhaft erläutert.

```
SELECT concat(a.feld3,"(",a.feld4,")") AS Wohnort,  
       a.feld5 AS Adresse,  
       count(b.feld1) AS Personenanzahl  
FROM   tabelle2 a, tabelle3 b  
WHERE  a.feld1=b.feld3  
GROUP BY a.feld3, a.feld4, a.feld5  
ORDER BY count(b.feld1) DESC;
```

Mit Hilfe dieser SELECT-Anweisung wird aus der bisher bereits verwendeten Melde-register-Datenbank die Anzahl der gemeldeten Personen herausgefiltert und anhand der Wohnorte und Adressen gruppiert. Die entsprechende Ergebnistabelle wird im Anhang A-12 gezeigt.

### 2.1 Darstellung der Abfrageelemente im Archivformat

Das Beispiel enthält folgende fünf grundlegende Abfrageelemente, die im Archivformat durch die entsprechenden Strukturelemente repräsentiert werden:

- Projektion,
- Verknüpfung,
- Gruppierung und
- Sortierung.

Die Abfrageelemente bestehen aus Ausdrücken in Form von Funktionen, aus Tabellenspalten und Werten.

Die XML-Struktur der Abfrage ist in Abbildung III-15 zu sehen. Das entsprechende XML-Dokument befindet sich im Anhang A-8.4.

Ein weiteres Abfragebeispiel mit komplexeren, rekursiven Strukturen ist im Anhang A-13 dokumentiert.

In den folgenden Abschnitten werden die Elemente der Abfrage einzeln betrachtet. Es wird beschrieben, wie sie im Beispiel verwendet und mit Hilfe der entsprechenden Strukturelemente im Archivformat umgesetzt werden.

III Die XML-Dokumenttypdefinition für Datenbankabfragen  
 2 Abfragebeispiel

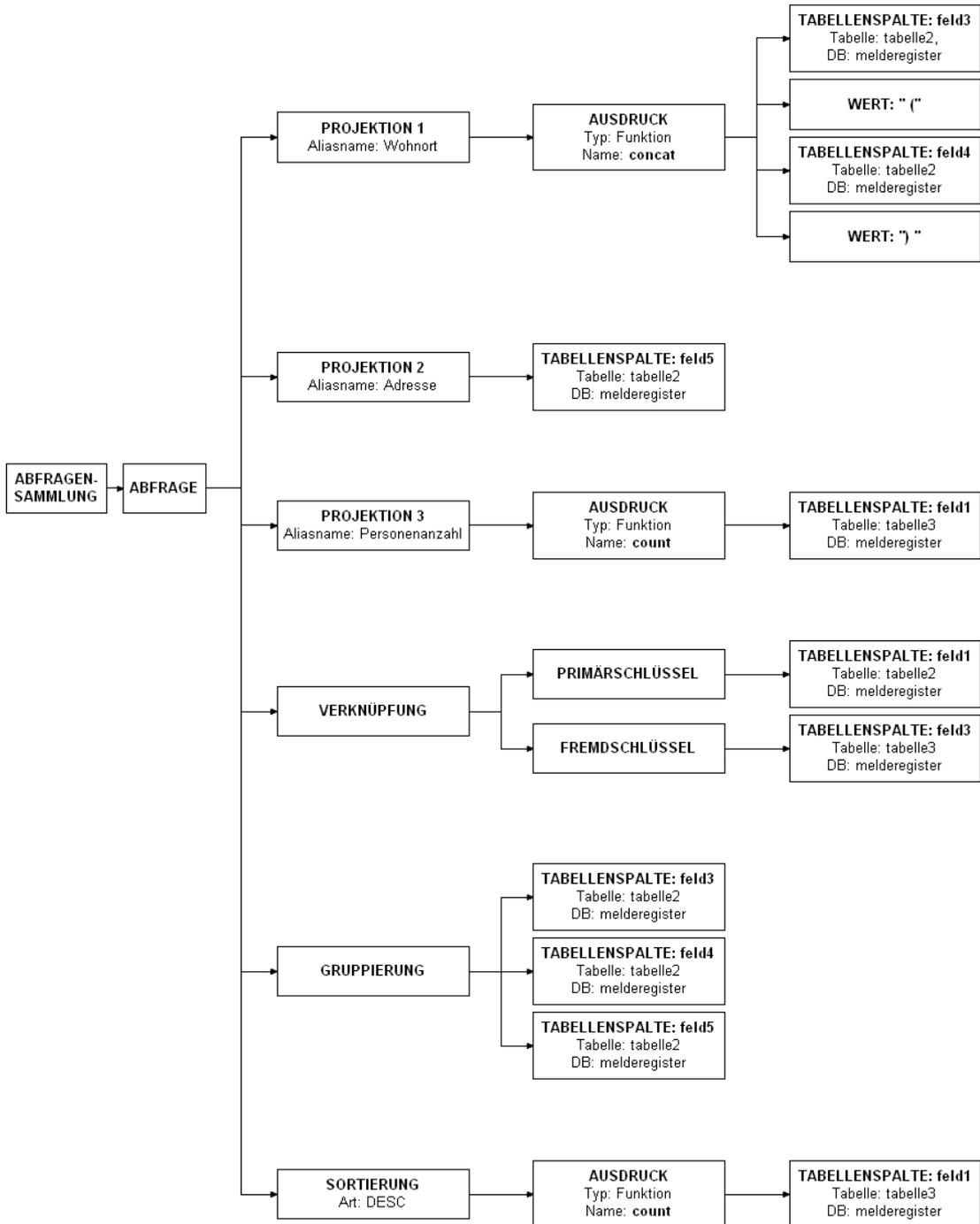


Abb. III-15: XML-Strukturbaum des Abfragebeispiels

### 2.1.1 Projektion

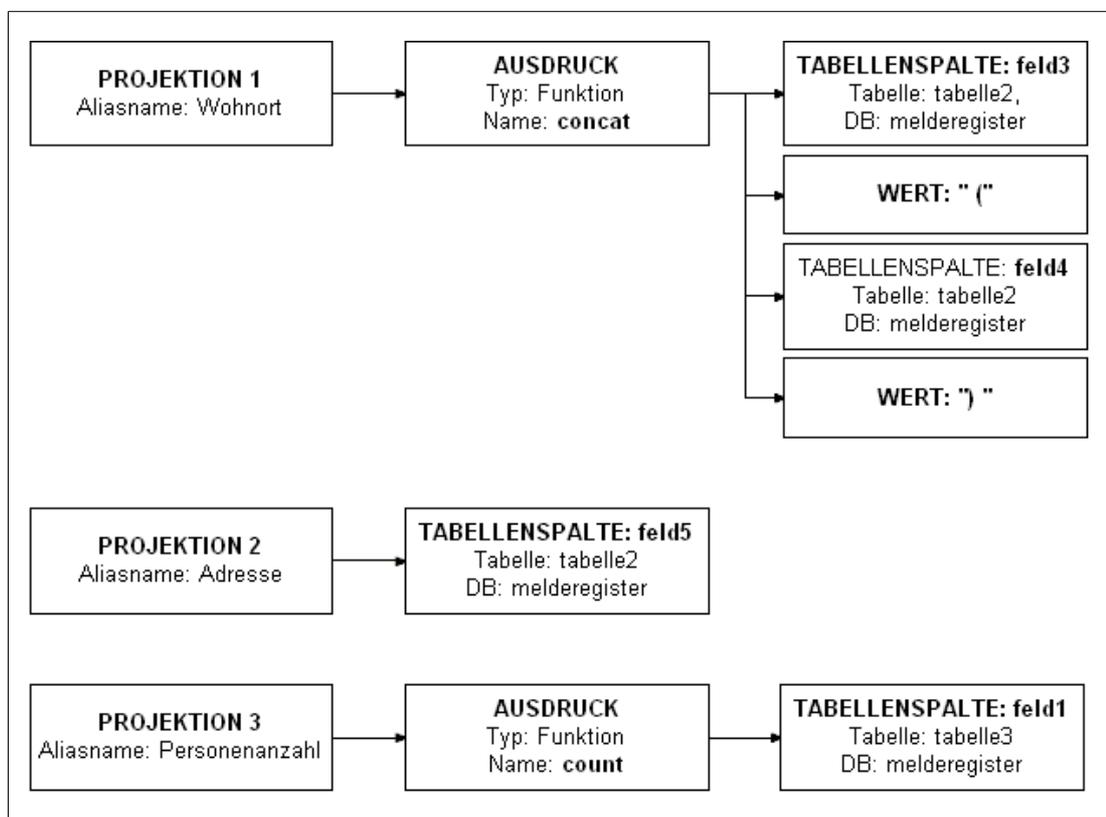
Folgende drei Projektionen sind in der Beispielabfrage enthalten:

- Projektion 1: **concat**(a.feld3," (" ,a.feld4," )") **AS** Wohnort
- Projektion 2: a.feld5 **AS** Adresse
- Projektion 3: **count**(b.feld1) **AS** Personenanzahl.

Projektion 1 hat den Aliasnamen „Wohnort“. Sie liegt als concat-Funktion und somit in Form eines Ausdrucks vor. Die Parameter dieser Funktion werden durch die Spalten „feld3“ und „feld4“ der Tabelle „tabelle2“ sowie durch die Werte „ („ und „)“ gebildet.

Der Projektion 2 ist der Aliasname „Adresse“ zugeordnet. Sie besteht aus der Spalte „feld5“ der Tabelle „tabelle2“.

Projektion 3 hat den Aliasnamen „Personenanzahl“ und beinhaltet ebenfalls einen Ausdruck. Es handelt sich um die Funktion count, die als Parameter die Spalte „feld1“ der Tabelle „tabelle3“ enthält.



```
<PROJEKTION Aliasname="Wohnort">
  <AUSDRUCK Typ="Funktion" Name="concat"
    Beschreibung="Verbindet mehrere Zeichenketten">
    <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
      feld3
    </TABELLENSPALTE>
    <WERT> (</WERT>
    <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
      feld4
    </TABELLENSPALTE>
    <WERT>)</WERT>
  </AUSDRUCK>
</PROJEKTION>
<PROJEKTION Aliasname="Adresse">
  <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
    feld5
  </TABELLENSPALTE>
</PROJEKTION>
<PROJEKTION Aliasname="Personenanzahl">
  <AUSDRUCK Typ="Funktion" Name="count"
    Beschreibung="Ermittelt die Anzahl von Werten">
    <TABELLENSPALTE Tabelle="tabelle3" DB="melderegister">
      feld1
    </TABELLENSPALTE>
  </AUSDRUCK>
</PROJEKTION>
```

**Abb. III-16:** Abfragebeispiel: Strukturbaum und XML-Code der Strukturelemente  
*PROJEKTION*

Im Archivformat wird die Projektion der Beispielabfrage mit Hilfe von drei Elementen PROJEKTION umgesetzt. Die genaue Struktur dieser Elemente wird in Abbildung III-16 dargestellt.

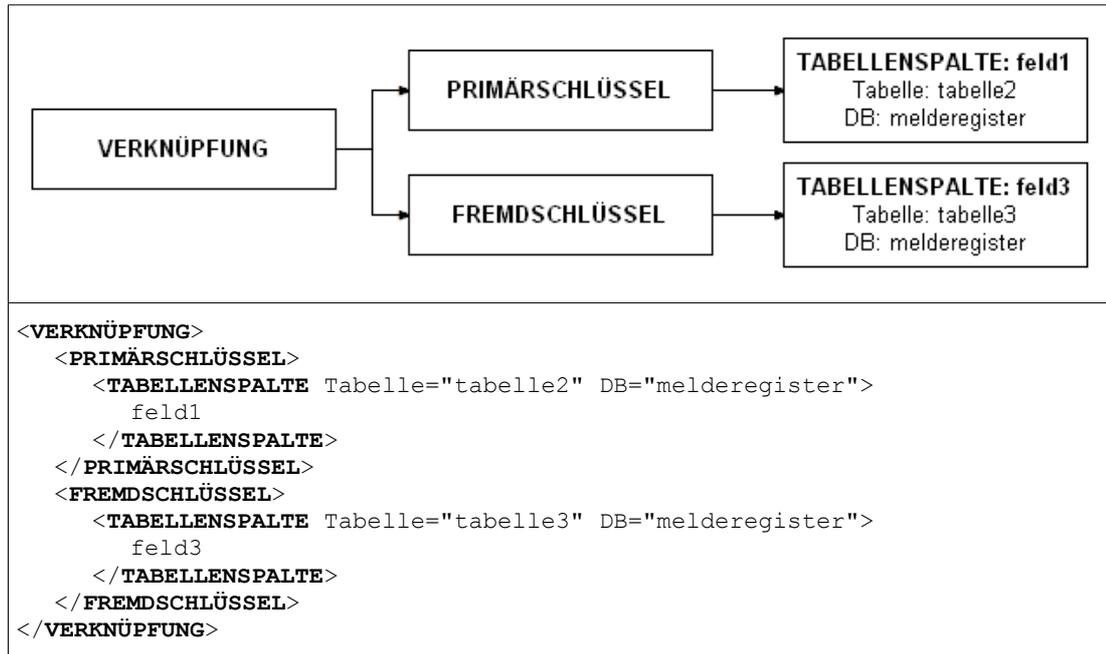
### 2.1.2 Verknüpfung

Die Tabellenverknüpfung wird in der Beispielabfrage durch folgende Vergleichsoperation realisiert:

a.feld1=b.feld3.

Die Tabelle „tabelle2“ wird mit der Tabelle „tabelle3“ verknüpft. Die Spalte „feld1“ der Tabelle „tabelle2“ dient als Primärschlüssel, die Spalte „feld3“ der Tabelle „tabelle3“ als Fremdschlüssel.

Abbildung III-17 zeigt, wie die Tabellenverknüpfung des Abfragebeispiels mit Hilfe des Strukturelements VERKNÜPFUNG beschrieben wird.



**Abb. III-17:** Abfragebeispiel: Strukturbaum und XML-Code des Strukturelements VERKNÜPFUNG

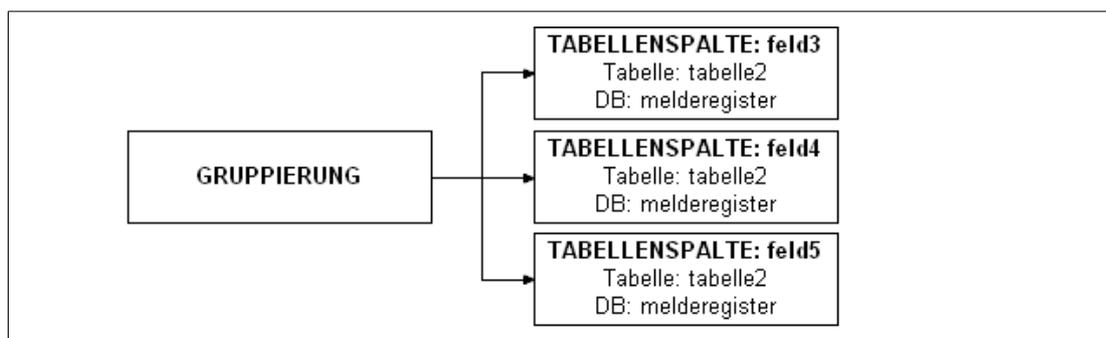
### 2.1.3 Gruppierung

Das Abfrageelement Gruppierung wird im Beispiel durch folgende SQL-Zeile vorgegeben:

```
GROUP BY a.feld3, a.feld4, a.feld5.
```

Die Datensätze werden bei der Ausgabe anhand der Spalten „feld3“, „feld4“ und „feld5“ der Tabelle „tabelle2“ zusammengefasst.

In Abbildung III-18 ist die Struktur des entsprechenden Strukturelements GRUPPIERUNG zu sehen.



```

<GRUPPIERUNG>
  <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
    feld3
  </TABELLENSPALTE>
  <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
    feld4
  </TABELLENSPALTE>
  <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
    feld5
  </TABELLENSPALTE>
</GRUPPIERUNG>
  
```

**Abb. III-18:** Abfragebeispiel: Strukturbaum und XML-Code des Strukturelements  
 GRUPPIERUNG

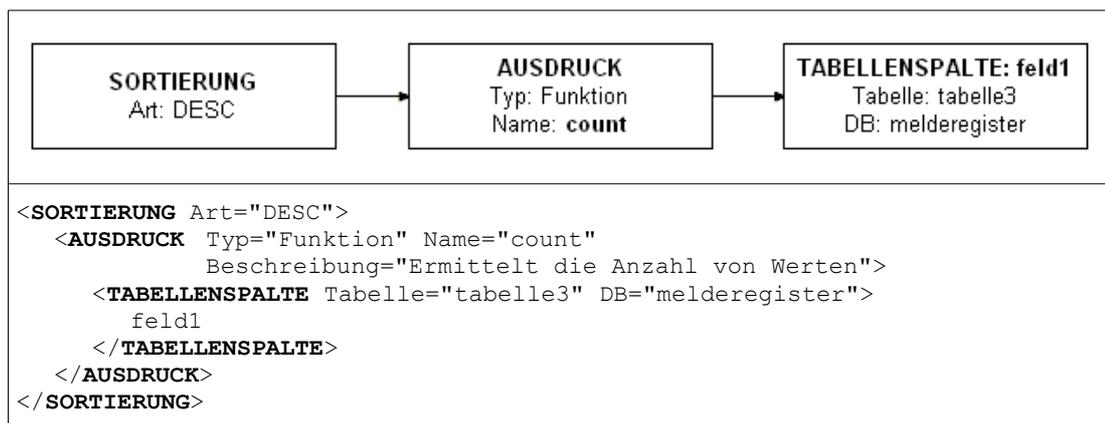
### 2.1.4 Sortierung

Das Abfrageelement Sortierung wird wie folgt im Abfragebeispiel beschrieben:

**ORDER BY** *count*(b.feld1) **DESC**.

Die Datensätze werden anhand der Ergebnisse eines Ausdrucks absteigend sortiert. Dieser Ausdruck liegt in Form einer count-Funktion vor und beinhaltet als Parameter die Spalte „feld1“ der Tabelle „tabelle3“.

Abbildung III-19 zeigt, wie diese Struktur im Archivformat umgesetzt wird.



**Abb. III-19:** Abfragebeispiel: Strukturbaum und XML-Code des Strukturelements  
 SORTIERUNG

### Fazit

Im Rahmen dieser Arbeit wurde ein Modell entwickelt, auf dessen Grundlage relationale Datenbanken in ein OAIS-konformes Archivsystem überführt und somit langfristig gesichert werden können.

Das hier vorgestellte Modell erweitert bekannte Archivierungskonzepte, die sich bisher nur mit der Archivierung der in den Datenbanken verwalteten Daten beschäftigt haben, um den wichtigen Aspekt einer vollständigen Sicherung der Datenbankabfragen. Durch die Archivierung der Datenbankabfragen können sowohl Teile des relationalen Datenmodells als auch ein Teil des vorarchivischen Nutzerverhaltens nachhaltig gesichert werden. Dieser Archivierungsansatz ermöglicht es, bei Bedarf die Datenbankfunktionalität, zumindest teilweise, wiederherzustellen.

Im Zusammenhang mit der Archivierung von Datenbankabfragen wurde eine XML-Dokumenttypdefinition entwickelt, die Abfragen semantisch abbildet. Das resultierende Archivformat für Datenbankabfragen ist lesbar, systemunabhängig und einfach strukturiert. Es kann relativ problemlos in andere standardisierte Abfragesprachen transformiert werden und zeichnet sich durch eine hohe Flexibilität im Bereitstellungsprozess aus.

Das vorgestellte Modell ist durch einige Einschränkungen gekennzeichnet. Diese werden im Folgenden aufgelistet:

1. Das im Modell beschriebene Verfahren kann nur die Datenbankfunktionalität sichern, die mit Hilfe von SQL-Abfragen abgebildet wird. Funktionalität, die durch komplexe, auf SQL aufsetzende Anwendungsprogramm bereitgestellt wird, kann nicht mit diesem Modell archiviert werden.
2. Mit der in dieser Arbeit entwickelten XML-Dokumenttypdefinition für Datenbankabfragen wird nur ein grundlegender Teil der Funktionalität der SQL-SELECT-Anweisung beschrieben. Um die gesamte Funktionalität einer SQL-Datenbankabfrage bei der Archivierung erfassen zu können, muss die XML-Dokumenttypdefinition weiter verfeinert werden.
3. Die XML-Dokumenttypdefinition für Datenbankabfragen ist nicht standardisiert.

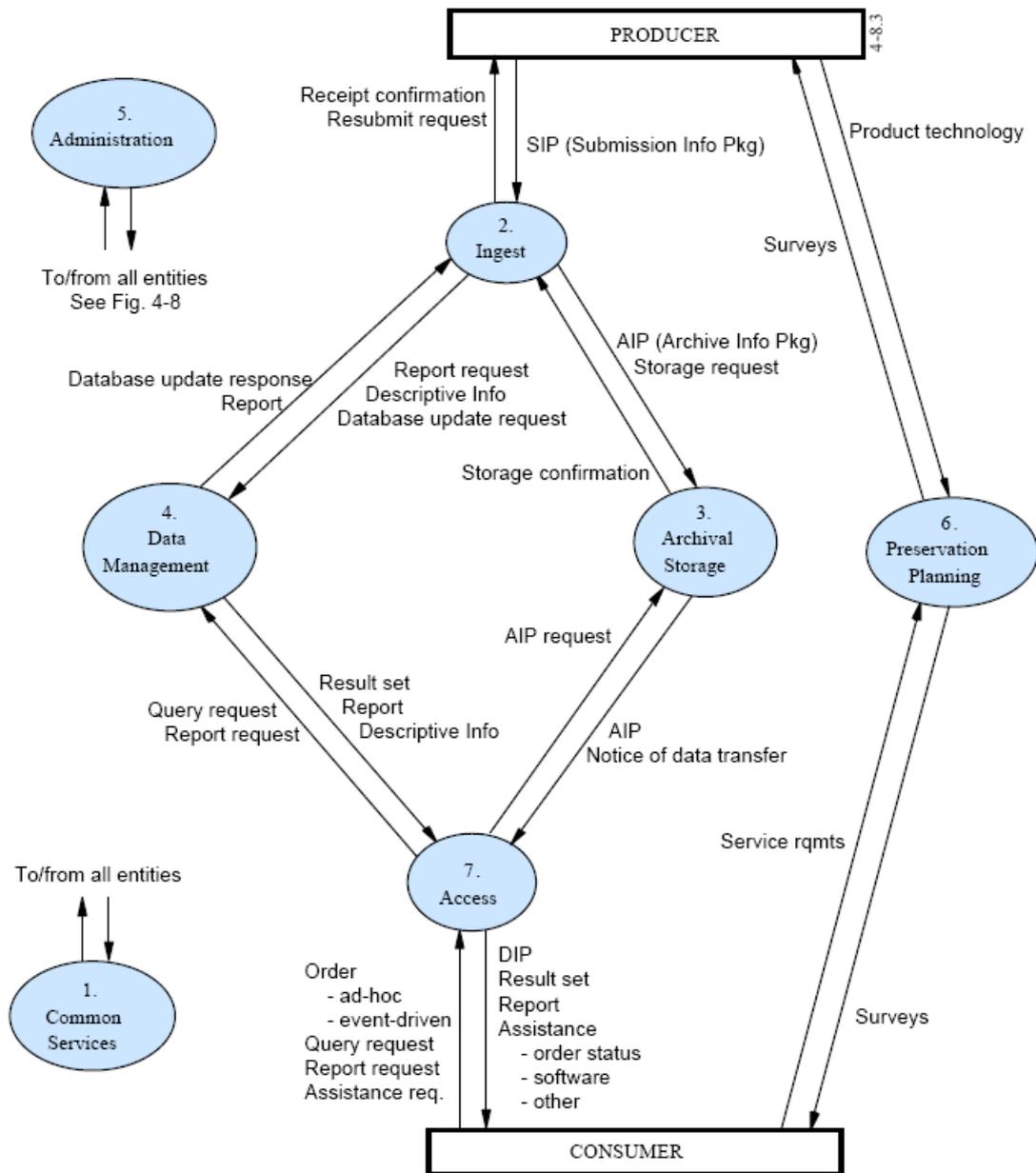
4. Für die Transformation der SQL-Abfragen in das XML-basierte Archivformat und für Bereitstellung des Archivformats in Form von standardisierten Abfragesprachen wie beispielsweise SQL und XQuery müssen noch Programme entwickelt werden.

## Anhang

A-1	Detailliertes OAIS-Datenflussmodell.....	73
A-2	Übersicht zu Projekten im Bereich der Archivierung relationaler Datenbanken.....	74
A-3	Entity-Relationship-Modell der Melderegister-Datenbank.....	75
A-4	Datenflussmodelle.....	76
A-4.1	Variante 1: SQL und generische Nutzeroberflächen.....	76
A-4.2	Variante 2: XML und XQuery.....	77
A-4.3	Variante 3: SQL.....	78
A-5	Einlieferungspaket.....	79
A-5.1	Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten.....	79
A-5.2	Beispiel für eine MySQL-Protokolldatei.....	87
A-5.3	Datenbankfeldbeschreibung.....	88
A-6	Vergleich von Archivformaten für Datenbankstruktur und Datenbankdaten.....	89
A-7	Vergleich von Archivformaten für Datenbankabfragen.....	91
A-8	Archivpaket.....	92
A-8.1	XML-Schema des Archivformates für Datenbankstruktur und Datenbankdaten.....	92
A-8.2	XML-Archivdatei mit Datenbankstruktur und Datenbankdaten.....	97
A-8.3	Dokumenttypdefinition des XML-basierten Archivformates für Abfragen.....	105
A-8.4	XML-Archivdatei für das Abfragebeispiel.....	106
A-9	Prozessmodelle.....	108
A-9.1	Variante 1: SQL und generische Nutzeroberflächen.....	108

A-9.2	Variante 2: XML und XQuery.....	109
A-9.3	Variante 3: SQL.....	110
A-10	SQLite-Syntaxdiagramm des SELECT-Befehls.....	111
A-11	Detaillierter Strukturbaum der XML-Dokumenttypdefinition für Daten- bankabfragen.....	114
A-12	Ergebnistabelle des Abfragebeispiels.....	115
A-13	Ergänzendes Abfragebeispiel mit komplexeren Strukturen.....	116
A-13.1	SELECT-Anweisung und Ergebnistabelle.....	116
A-13.2	XML-Strukturbaum.....	117
A-13.3	XML-Archivdatei.....	118

## A-1 Detailliertes OAIS-Datenflussmodell

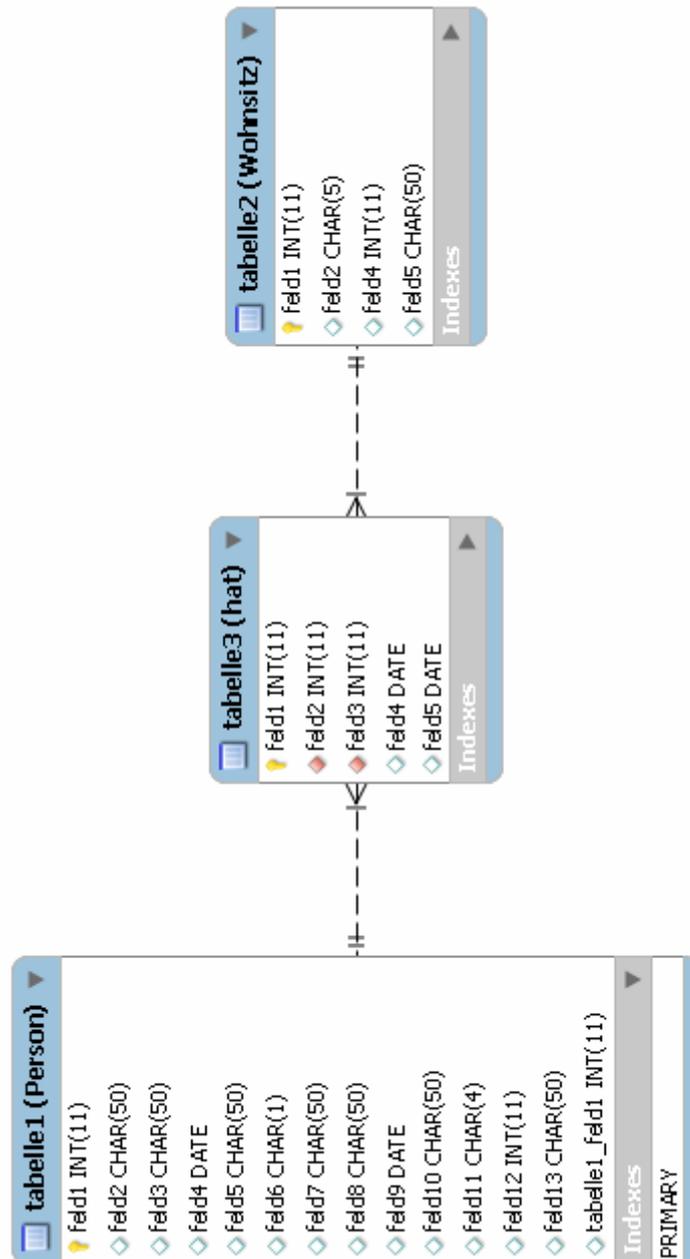


Quelle: CCSDS: Reference Model for an Open Archival Information System (OAIS), Januar 2002, S. 4-17

## A-2 Übersicht zu Projekten im Bereich der Archivierung relationaler Datenbanken

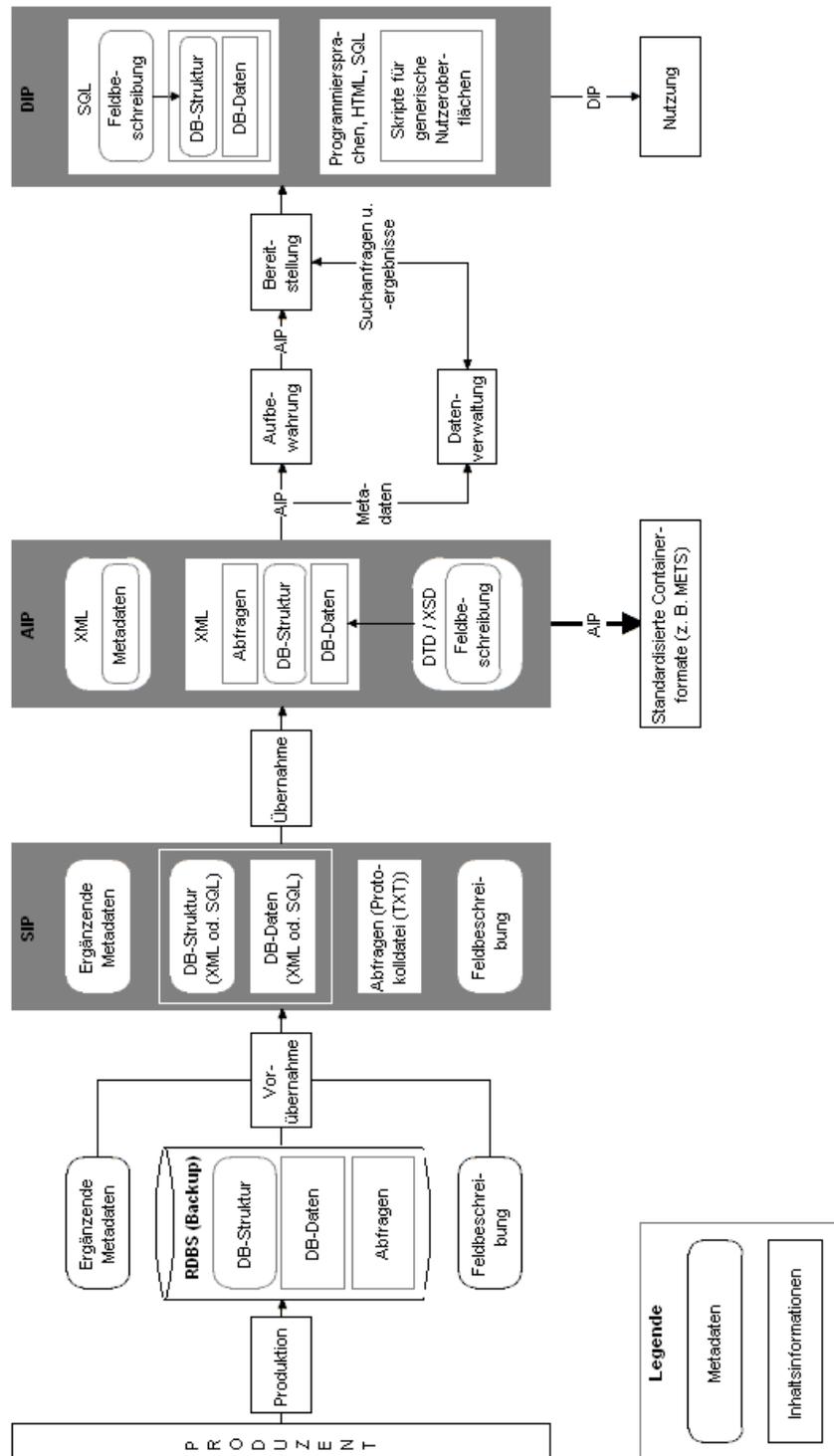
Projekte	Archivierung von					Bereitstellung der Datenbank
	Datenbankdaten	Datenbankstruktur	Metadaten	Datenbankabfragen	Verknüpfungen	
<b>ARELDA</b> mit der Softwarelösung <b>SIARD</b> (Schweizerisches Bundesarchiv)	XML-Format	SQL-DDL-Dateien (ein DDL-Datei pro Tabelle)	XML-Format – Datenbankstruktur – Informationen über die Archivierungsprozesse – beschreibende Metadaten – gespeicherte Routinen, Views usw. – ...	Views können übernommen werden.	realisiert durch die SQL-Struktur	Data Browser: Darstellung im Webbrowser mit Hilfe von XSL-Dateien
<b>Digital Preservation Testbed</b> (Nationalarchiv der Niederlande)	XML-Format (eine Datei pro Tabelle)	XML-Datei mit Tabellenliste und Beschreibung der wichtigsten Eigenschaften (Datentyp, Primärschlüssel, Fremdschlüssel)	keine Angaben	Views können übernommen werden.	keine Angaben	keine Angaben
<b>LWL-Archivamt für Westfalen</b>	CSV-Format mit ASCII / Unicode oder XML-Format mit Unicode	Dokumentation (Codebuch): – Tabellenspalten – Auflösung von Zahlencodes	keine Angaben	keine Angaben	Dokumentation	keine Angaben
<b>Migrationsprogramm der Universität Koblenz, Bundesarchiv</b>	Export im ASCII-Format Archivierung im XML-Format	keine Angaben	Erfassung in einer Erschließungsdatei	nicht berücksichtigt Die archivierten Daten können mit SQL abgefragt werden.	nicht berücksichtigt Es wurde nur eine einfache Datenbank mit einer Tabelle betrachtet	Darstellung im Webbrowser
<b>Rahmenkonzept zur Langzeitarchivierung digitaler Daten</b> (Stadtarchiv Stuttgart)	Flache Dateien: CSV-Format mit Unicode	SQL-Format mit ASCII / Unicode	XML-Format mit ASCII / Unicode und zugänglicher DTD	keine Angaben	keine Angaben	keine Angaben
<b>RODA</b> (Universität Minho, Portugiesisches Nationalarchiv)	XML-Format (DBML-Dateien ( <i>Database Markup Language</i> ))	XML-Format (DBML-Dateien ( <i>Database Markup Language</i> ))	XML-Format (beschreibende Metadaten)	keine Angaben	Dokumentation der Primär- und Fremdschlüssel in der XML-Datei für Inhalt und Struktur	SQL-Format: Verwendung in Verbindung mit einem RDBS HTML: grafische Benutzeroberfläche

### A-3 Entity-Relationship-Modell der Melderegister-Datenbank

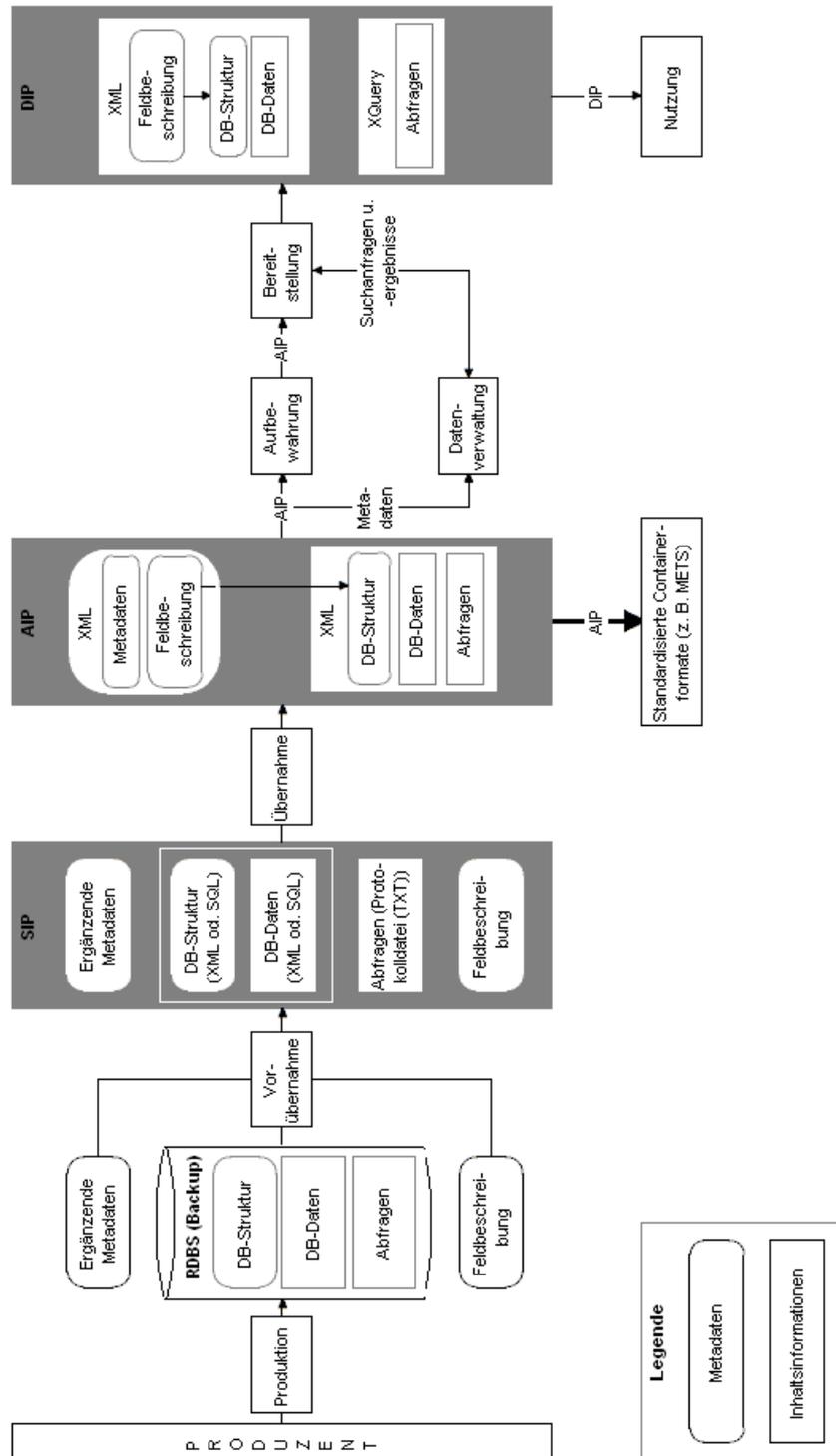


## A-4 Datenflussmodelle

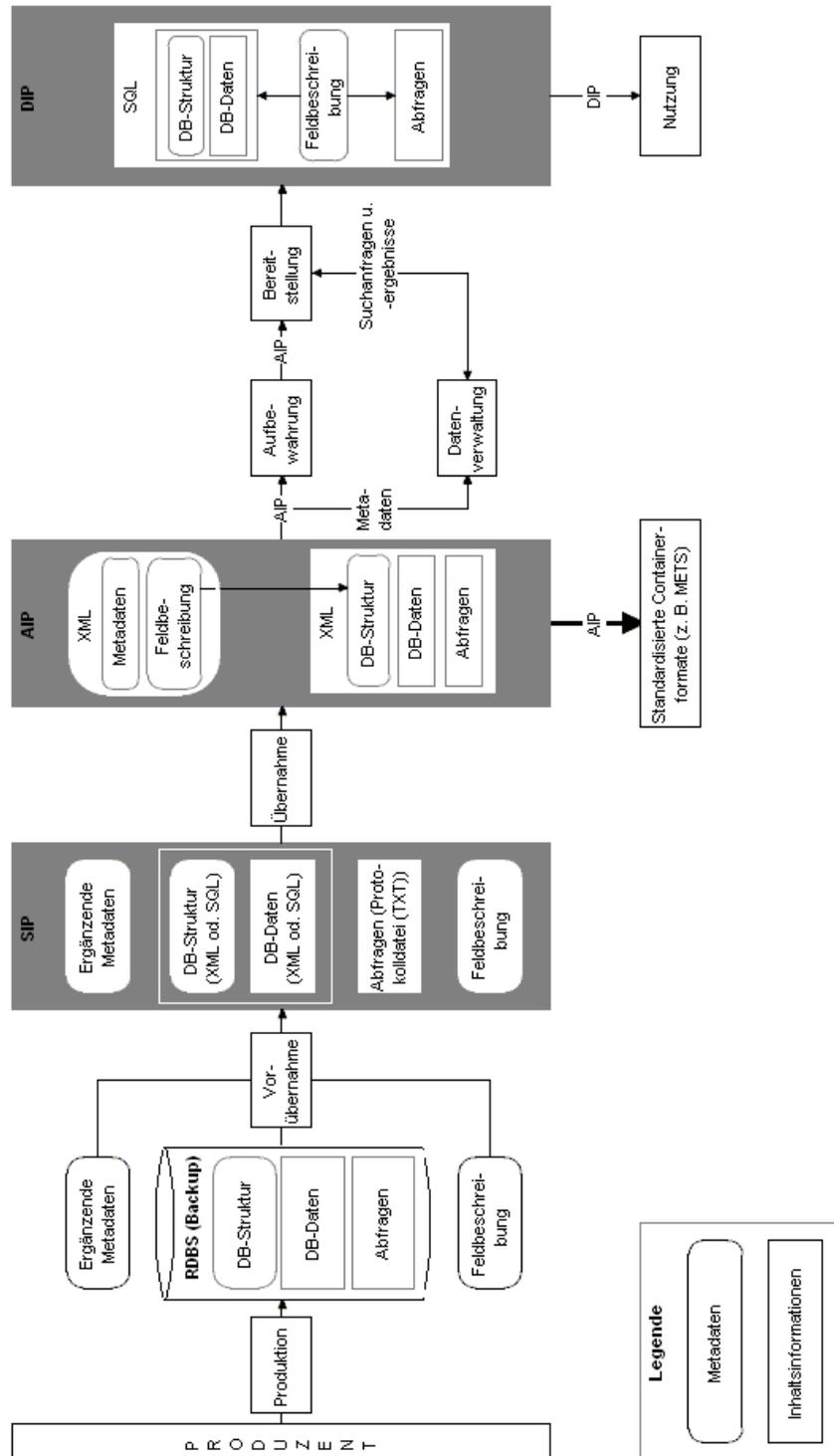
### A-4.1 Variante 1: SQL und generische Nutzeroberflächen



### A-4.2 Variante 2: XML und XQuery



### A-4.3 Variante 3: SQL



## A-5 Einlieferungspaket

### A-5.1 Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

```
<?xml version="1.0"?>
<mysqldump xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<database name="melderegister">
  <table_structure name="tabelle1">
    <field Field="feld1" Type="int(11)" Null="NO" Key="PRI" Extra="" />
    <field Field="feld2" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld3" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld4" Type="date" Null="YES" Key="" Extra="" />
    <field Field="feld5" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld6" Type="char(1)" Null="YES" Key="" Extra="" />
    <field Field="feld7" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld8" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld9" Type="date" Null="YES" Key="" Extra="" />
    <field Field="feld10" Type="char(50)" Null="YES" Key="" Extra="" />
    <field Field="feld11" Type="char(4)" Null="YES" Key="" Extra="" />
    <field Field="feld12" Type="int(11)" Null="YES" Key="" Extra="" />
    <field Field="feld13" Type="char(50)" Null="YES" Key="" Extra="" />
    <key Table="tabelle1" Non_unique="0" Key_name="PRIMARY"
      Seq_in_index="1" Column_name="feld1" Collation="A" Cardinality="10"
      Null="" Index_type="BTREE" Comment="" />
    <options Name="tabelle1" Engine="InnoDB" Version="10"
      Row_format="Compact" Rows="10" Avg_row_length="1638"
      Data_length="16384" Max_data_length="0" Index_length="0"
      Data_free="4194304" Create_time="2009-06-28 09:44:58"
      Collation="latin1_swedish_ci" Create_options="" Comment="" />
  </table_structure>
  <table_data name="tabelle1">
  <row>
    <field name="feld1">1</field>
    <field name="feld2">Löwe</field>
    <field name="feld3">Ralf</field>
    <field name="feld4">1939-06-06</field>
    <field name="feld5">Aspenstedt</field>
    <field name="feld6">w</field>
    <field name="feld7">deutsch</field>
    <field name="feld8">Atheist</field>
    <field name="feld9">1993-08-26</field>
    <field name="feld10">Aspenstedt</field>
    <field name="feld11">Ja</field>
    <field name="feld12" xsi:nil="true" />
    <field name="feld13">ledig</field>
  </row>
  <row>
    <field name="feld1">2</field>
    <field name="feld2">Lange</field>
    <field name="feld3">Johanna</field>
    <field name="feld4">1984-03-16</field>
    <field name="feld5">Potsdam</field>
    <field name="feld6">w</field>
    <field name="feld7">deutsch</field>
    <field name="feld8">Katholik</field>
    <field name="feld9" xsi:nil="true" />
  </row>
  </table_data>
</database>
</mysqldump>
```

## Anhang

### A-5 Einlieferungspaket: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

---

```
<field name="feld10" xsi:nil="true" />
<field name="feld11">Nein</field>
<field name="feld12" xsi:nil="true" />
<field name="feld13">verheiratet</field>
</row>
<row>
  <field name="feld1">3</field>
  <field name="feld2">Schreiber</field>
  <field name="feld3">Monika</field>
  <field name="feld4">1951-02-13</field>
  <field name="feld5">Potsdam</field>
  <field name="feld6">w</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Katholik</field>
  <field name="feld9" xsi:nil="true" />
  <field name="feld10" xsi:nil="true" />
  <field name="feld11">Ja</field>
  <field name="feld12" xsi:nil="true" />
  <field name="feld13">verheiratet</field>
</row>
<row>
  <field name="feld1">4</field>
  <field name="feld2">Weiss</field>
  <field name="feld3">Luca</field>
  <field name="feld4">1968-06-03</field>
  <field name="feld5">Berlin</field>
  <field name="feld6">w</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Protestant</field>
  <field name="feld9" xsi:nil="true" />
  <field name="feld10" xsi:nil="true" />
  <field name="feld11">Ja</field>
  <field name="feld12" xsi:nil="true" />
  <field name="feld13">verheiratet</field>
</row>
<row>
  <field name="feld1">5</field>
  <field name="feld2">Theiss</field>
  <field name="feld3">Mario</field>
  <field name="feld4">1970-01-02</field>
  <field name="feld5">Eldingen</field>
  <field name="feld6">m</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Atheist</field>
  <field name="feld9" xsi:nil="true" />
  <field name="feld10" xsi:nil="true" />
  <field name="feld11">Ja</field>
  <field name="feld12" xsi:nil="true" />
  <field name="feld13">verheiratet</field>
</row>
<row>
  <field name="feld1">6</field>
  <field name="feld2">Fuhrmann</field>
  <field name="feld3">Ulrike</field>
  <field name="feld4">1983-07-15</field>
  <field name="feld5">Aitrang</field>
  <field name="feld6">w</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Protestant</field>
  <field name="feld9">2003-12-19</field>
```

## Anhang

### A-5 Einlieferungspaket: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

---

```
<field name="feld10">Berlin</field>
<field name="feld11">Ja</field>
<field name="feld12" xsi:nil="true" />
<field name="feld13">verwitwet</field>
</row>
<row>
  <field name="feld1">7</field>
  <field name="feld2">Maur</field>
  <field name="feld3">Klaudia</field>
  <field name="feld4">1973-09-15</field>
  <field name="feld5">Behringen</field>
  <field name="feld6">w</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Atheist</field>
  <field name="feld9" xsi:nil="true" />
  <field name="feld10" xsi:nil="true" />
  <field name="feld11">Ja</field>
  <field name="feld12">10</field>
  <field name="feld13">ledig</field>
</row>
<row>
  <field name="feld1">8</field>
  <field name="feld2">Kunze</field>
  <field name="feld3">Jan</field>
  <field name="feld4">1948-01-28</field>
  <field name="feld5">Würzburg</field>
  <field name="feld6">m</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Atheist</field>
  <field name="feld9" xsi:nil="true" />
  <field name="feld10" xsi:nil="true" />
  <field name="feld11">Ja</field>
  <field name="feld12" xsi:nil="true" />
  <field name="feld13">ledig</field>
</row>
<row>
  <field name="feld1">9</field>
  <field name="feld2">Aachen</field>
  <field name="feld3">Kevin</field>
  <field name="feld4">1978-12-08</field>
  <field name="feld5">Berlin</field>
  <field name="feld6">m</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Buddhist</field>
  <field name="feld9" xsi:nil="true" />
  <field name="feld10" xsi:nil="true" />
  <field name="feld11">Ja</field>
  <field name="feld12" xsi:nil="true" />
  <field name="feld13">verheiratet</field>
</row>
<row>
  <field name="feld1">10</field>
  <field name="feld2">Bieber</field>
  <field name="feld3">Julia</field>
  <field name="feld4">1989-04-05</field>
  <field name="feld5">Behringen</field>
  <field name="feld6">w</field>
  <field name="feld7">deutsch</field>
  <field name="feld8">Katholik</field>
  <field name="feld9" xsi:nil="true" />
```

## Anhang

### A-5 Einlieferungspaket: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

---

```
<field name="feld10" xsi:nil="true" />
<field name="feld11">Ja</field>
<field name="feld12" xsi:nil="true" />
<field name="feld13">ledig</field>
</row>
</table_data>
<table_structure name="tabelle2">
  <field Field="feld1" Type="int(11)" Null="NO" Key="PRI" Extra="" />
  <field Field="feld2" Type="char(5)" Null="YES" Key="" Extra="" />
  <field Field="feld3" Type="char(50)" Null="YES" Key="" Extra="" />
  <field Field="feld4" Type="int(11)" Null="YES" Key="" Extra="" />
  <field Field="feld5" Type="char(50)" Null="YES" Key="" Extra="" />
  <key Table="tabelle2" Non_unique="0" Key_name="PRIMARY"
    Seq_in_index="1" Column_name="feld1" Collation="A" Cardinality="14"
    Null="" Index_type="BTREE" Comment="" />
  <options Name="tabelle2" Engine="InnoDB" Version="10"
    Row_format="Compact" Rows="14" Avg_row_length="1170"
    Data_length="16384" Max_data_length="0" Index_length="0"
    Data_free="4194304" Create_time="2009-06-28 09:45:10"
    Collation="latin1_swedish_ci" Create_options="" Comment="" />
</table_structure>
<table_data name="tabelle2">
<row>
  <field name="feld1">1</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Aspenstedt</field>
  <field name="feld4">38822</field>
  <field name="feld5">Feldstr. 5</field>
</row>
<row>
  <field name="feld1">2</field>
  <field name="feld2">Neben</field>
  <field name="feld3">Aspenstedt</field>
  <field name="feld4">38823</field>
  <field name="feld5">Feldstr. 5</field>
</row>
<row>
  <field name="feld1">3</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Aitrang</field>
  <field name="feld4">87648</field>
  <field name="feld5">Ufnaustr. 24</field>
</row>
<row>
  <field name="feld1">4</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Behringen</field>
  <field name="feld4">99326</field>
  <field name="feld5">Königstr. 52</field>
</row>
<row>
  <field name="feld1">5</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Eldingen</field>
  <field name="feld4">29351</field>
  <field name="feld5">Luckenwalder Str. 86</field>
</row>
<row>
  <field name="feld1">6</field>
  <field name="feld2">Neben</field>
```

## Anhang

### A-5 Einlieferungspaket: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

---

```
<field name="feld3">Eisfeld</field>
<field name="feld4">98671</field>
<field name="feld5">Eschenweg 46</field>
</row>
<row>
  <field name="feld1">7</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Berlin</field>
  <field name="feld4">10439</field>
  <field name="feld5">Trellenborger Str. 3</field>
</row>
<row>
  <field name="feld1">8</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Berlin</field>
  <field name="feld4">10179</field>
  <field name="feld5">Klosterstr. 3</field>
</row>
<row>
  <field name="feld1">9</field>
  <field name="feld2">Neben</field>
  <field name="feld3">Berlin</field>
  <field name="feld4">13089</field>
  <field name="feld5">Indunastr.105</field>
</row>
<row>
  <field name="feld1">10</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Potsdam</field>
  <field name="feld4">14469</field>
  <field name="feld5">Aalsteig 31</field>
</row>
<row>
  <field name="feld1">11</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Potsdam</field>
  <field name="feld4">14469</field>
  <field name="feld5">Florastr. 1</field>
</row>
<row>
  <field name="feld1">12</field>
  <field name="feld2">Neben</field>
  <field name="feld3">Potsdam</field>
  <field name="feld4">14469</field>
  <field name="feld5">Aalsteig 31</field>
</row>
<row>
  <field name="feld1">13</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Berlin</field>
  <field name="feld4">13355</field>
  <field name="feld5">Usedomerstr. 23</field>
</row>
<row>
  <field name="feld1">14</field>
  <field name="feld2">Haupt</field>
  <field name="feld3">Würzburg</field>
  <field name="feld4">97015</field>
  <field name="feld5">Mühlenstr. 49</field>
</row>
```

## Anhang

### A-5 Einlieferungspaket: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

---

```
</table_data>
<table_structure name="tabelle3">
  <field Field="feld1" Type="int(11)" Null="NO" Key="PRI" Extra="" />
  <field Field="feld2" Type="int(11)" Null="NO" Key="" Extra="" />
  <field Field="feld3" Type="int(11)" Null="NO" Key="" Extra="" />
  <field Field="feld4" Type="date" Null="YES" Key="" Extra="" />
  <field Field="feld5" Type="date" Null="YES" Key="" Extra="" />
  <key Table="tabelle3" Non_unique="0" Key_name="PRIMARY"
    Seq_in_index="1" Column_name="feld1" Collation="A" Cardinality="16"
    Null="" Index_type="BTREE" Comment="" />
  <options Name="tabelle3" Engine="InnoDB" Version="10"
    Row_format="Compact" Rows="16" Avg_row_length="1024"
    Data_length="16384" Max_data_length="0" Index_length="0"
    Data_free="4194304" Create_time="2009-06-28 09:45:27"
    Collation="latin1_swedish_ci" Create_options="" Comment="" />
</table_structure>
<table_data name="tabelle3">
<row>
  <field name="feld1">1</field>
  <field name="feld2">1</field>
  <field name="feld3">1</field>
  <field name="feld4">1954-08-15</field>
  <field name="feld5">1993-08-26</field>
</row>
<row>
  <field name="feld1">2</field>
  <field name="feld2">6</field>
  <field name="feld3">2</field>
  <field name="feld4">2001-08-14</field>
  <field name="feld5">2003-12-22</field>
</row>
<row>
  <field name="feld1">3</field>
  <field name="feld2">6</field>
  <field name="feld3">3</field>
  <field name="feld4">1983-07-15</field>
  <field name="feld5">2003-12-22</field>
</row>
<row>
  <field name="feld1">4</field>
  <field name="feld2">2</field>
  <field name="feld3">7</field>
  <field name="feld4">2000-04-07</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">5</field>
  <field name="feld2">3</field>
  <field name="feld3">10</field>
  <field name="feld4">1951-02-13</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">6</field>
  <field name="feld2">3</field>
  <field name="feld3">8</field>
  <field name="feld4">1970-09-14</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
```

## Anhang

### A-5 Einlieferungspaket: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

---

```
<field name="feld1">7</field>
<field name="feld2">4</field>
<field name="feld3">10</field>
<field name="feld4">1990-11-08</field>
<field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">8</field>
  <field name="feld2">5</field>
  <field name="feld3">5</field>
  <field name="feld4">1988-03-09</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">9</field>
  <field name="feld2">5</field>
  <field name="feld3">9</field>
  <field name="feld4">2001-09-01</field>
  <field name="feld5">2008-06-16</field>
</row>
<row>
  <field name="feld1">10</field>
  <field name="feld2">7</field>
  <field name="feld3">4</field>
  <field name="feld4">1980-10-24</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">11</field>
  <field name="feld2">8</field>
  <field name="feld3">14</field>
  <field name="feld4">1948-01-28</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">12</field>
  <field name="feld2">9</field>
  <field name="feld3">13</field>
  <field name="feld4">1986-01-27</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">13</field>
  <field name="feld2">9</field>
  <field name="feld3">6</field>
  <field name="feld4">2003-06-27</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">14</field>
  <field name="feld2">10</field>
  <field name="feld3">12</field>
  <field name="feld4">2008-08-01</field>
  <field name="feld5" xsi:nil="true" />
</row>
<row>
  <field name="feld1">15</field>
  <field name="feld2">10</field>
  <field name="feld3">4</field>
  <field name="feld4">1989-04-05</field>
```

## Anhang

### A-5 Einlieferungspaket: Exportiertes XML-Dokument mit Datenbankstruktur und Datenbankdaten

---

```
    <field name="feld5" xsi:nil="true" />
  </row>
  <row>
    <field name="feld1">16</field>
    <field name="feld2">3</field>
    <field name="feld3">11</field>
    <field name="feld4">1967-11-01</field>
    <field name="feld5" xsi:nil="true" />
  </row>
</table_data>
</database>
</mysqldump>
```

## Anhang

### A-5 Einlieferungspaket: Beispiel für eine MySQL-Protokolldatei

---

#### A-5.2 Beispiel für eine MySQL-Protokolldatei

```
C:\Programme\MySQL\MySQL Server 5.1\bin\mysqld, Version: 5.1.35-community-
log (MySQL Community Server (GPL)). started with:
TCP Port: 3306, Named Pipe: (null)
Time          Id Command      Argument
090707 17:57:17    1 Connect      root@localhost on
              1 Query  SELECT @@sql_mode
              1 Query  SET SESSION
sql_mode='STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
              1 Query  SET NAMES utf8
090707 17:57:19    2 Connect      user_1@localhost on
              2 Query  select @@version_comment limit 1
090707 17:57:25    2 QuerySELECT DATABASE()
              2 Init  DBmelderegister
090707 17:57:32    2 Queryshow tables
090707 17:58:17    2 QuerySELECT concat(a.feld3," (",a.feld4,")") AS
Wohnort, a.feld5 AS Adresse, count(b.feld1) AS Personenanzahl
FROM tabelle2 a, tabelle3 b
WHERE a.feld1=b.feld3
GROUP BY a.feld3, a.feld4, a.feld5
ORDER BY count(b.feld1) DESC
090707 17:59:07    2 QuerySELECT concat(b.feld5," ",b.feld4," ",b.feld3)
AS Wohnsitz, b.feld2 AS "Art des Wohnsitzes", a.*
FROM tabelle1 a, tabelle2 b, tabelle3 c
WHERE b.feld5 ="Aalsteig 31" AND b.feld4=14469 AND a.feld1=c.feld2 AND
c.feld3=b.feld1
090707 17:59:43    2 QuerySELECT * FROM tabelle1
WHERE feld11="Nein"
090707 18:00:35    2 QuerySELECT concat(b.feld5," ",b.feld2," ",b.feld3)
AS Nebenwohnsitz, concat(a.feld3," ",a.feld2) AS Person, a.feld4 AS
Geburtsdatum
FROM tabelle1 a, tabelle2 b, tabelle3 c
WHERE b.feld2="Neben" AND a.feld1=c.feld2 AND c.feld3=b.feld1
ORDER BY a.feld2 ASC
090707 18:01:40    2 QuerySELECT * FROM tabelle1
WHERE feld2="Kunze" AND feld3="Jan" AND feld4="1948-01-28"
090707 18:01:45    2 Quit
```

### A-5.3 Datenbankfeldbeschreibung

<b>tabelle1 Person</b>			
feld1	Personennummer	Primärschlüssel	INTEGER, NOT NULL
feld2	Name		CHAR
feld3	Vorname		CHAR
feld4	Geburtsdatum		DATE
feld5	Geburtsort		CHAR
feld6	Geschlecht	w od. m	CHAR
feld7	Staatsangehörigkeit		CHAR
feld8	Religion		CHAR
feld9	Todesdatum		DATE
feld10	Todesort		CHAR
feld11	Wahlrecht	Ja od. Nein	CHAR
feld12	Kinder	Verweis auf Personennummer	INTEGER
feld13	Familienstand		CHAR
<b>tabelle2 Wohnsitz</b>			
feld1	Wohnsitznummer	Primärschlüssel	INTEGER, NOT NULL
feld2	Art	Haupt- od. Nebenwohnung	CHAR
feld3	Stadt		CHAR
feld4	Postleitzahl		INTEGER
feld5	Adresse	Straße u. Hausnummer	CHAR
<b>tabelle3 hat</b>			
		<b>Hilfstabelle</b>	
feld1	Hat-Nummer	Primärschlüssel	INTEGER, NOT NULL
feld2	Personennummer	Fremdschlüssel	INTEGER, NOT NULL
feld3	Wohnsitznummer	Fremdschlüssel	INTEGER, NOT NULL
feld4	Einzugsdatum		DATE
feld5	Auszugsdatum		DATE

## A-6 Vergleich von Archivformaten für Datenbankstruktur und Datenbankdaten

	SQL	XML-basiertes Format	CSV (Textformat)
<b>Standard</b>	Ja (Basis der meisten RDBS)	Nein (XML: ja, XSD/DTD: nein)	Nein, nur grundlegend beschrieben (Folge: Unterschiede bei der Formatierung (z. B. Trennzeichen, Zeichenkodierung))
<b>Systemunabhängigkeit</b>	Prinzipiell ja, da ein standardisierter Kern von SQL in den RDBS verwendet wird. Jedoch entstehen durch nicht-standardisierte Erweiterungen produktspezifische SQL-Dialekte.	Ja	Ja. Es kann von jedem Textverarbeitungsprogramm gelesen und erstellt werden. (Voraussetzung ist eine zusätzliche Formatbeschreibung.)
<b>Lesbarkeit</b>	Ja, da rein deskriptiv Einschränkung durch erforderliche SQL-Kenntnisse	Ja, da rein deskriptiv und sehr gut strukturiert	Ja, aber eingeschränkt aufgrund fehlender Standardisierung und Strukturbeschreibung
<b>Verarbeitung</b>	aufwändig aufgrund der Komplexität und des Umfangs von SQL	einfach aufgrund der einfachen Strukturierung	leicht, da es sich um ein sehr kompaktes Format handelt und nur die Daten verarbeitet werden müssen.
<b>Konversion/Transformation bei der Übernahme</b>	SQL kann aus der Datenbank exportiert werden, aber eine Konversion in standardkonformes SQL ist notwendig.	XML kann aus der Datenbank exportiert werden, aber eine Konversion in ein XML-Archivformat ist notwendig.	Nein. Ein direkter Export aus der Datenbank ist möglich. Allerdings muss eine Formatbeschreibung erstellt werden.
<b>Bereitstellung und Datenzugriff</b>	Konversion (individuelle Anpassung des gespeicherten SQL an den SQL-Dialekt des jeweiligen RDBS)	Transformation (bei der Bereitstellung im SQL-Format)  keine Transformation (bei der Bereitstellung der Abfragen im XQuery-Format oder in einem RDBS mit XML-Funktionalität)	Eine Transformation ist notwendig.
<b>Austauschbarkeit</b>	eingeschränkt durch SQL-Dialekte der RDBS	Ja	Ja, eingeschränkt aufgrund der fehlenden Standardisierung

**Anhang**A-6 Vergleich von Archivformaten für Datenbankstruktur und Datenbankdaten

---

<b>Was wird gespeichert?</b>	Daten und Struktur verbunden im SQL-Befehl (CREATE TABLE, INSERT INTO u. a. m.)	Daten, Struktur und Datenbeschreibung (DTD, XSD)	Nur Daten, Strukturbeschreibungen sind zusätzlich erforderlich (Trennung von Daten und Metadaten)  in der Regel keine Speicherung der Tabellenverknüpfungen und Feldbeschreibung
<b>Sonstiges</b>		Umfangreiche Struktur durch mögliche Elementredundanzen	Kompaktes Format

## A-7 Vergleich von Archivformaten für Datenbankabfragen

	SQL	XML-basiertes Format	XQuery
<b>Standard</b>	Ja (Basis der meisten RDBS)	Nein (XML: ja, XSD/DTD: nein)	Ja und XML-basiert
<b>Systemunabhängigkeit</b>	Prinzipiell ja, da ein standardisierter Kern von SQL in den RDBS verwendet wird. Jedoch entstehen durch nicht-standardisierte Erweiterungen produkt-spezifische SQL-Dialekte.	Ja	Ja
<b>Lesbarkeit</b>	Ja, da rein deskriptiv, Einschränkung durch erforderliche SQL-Kenntnisse	Ja, da rein deskriptiv und sehr gut strukturiert	Ja, aber eingeschränkt. XQuery ist nicht rein deskriptiv, da auch die Suchalgorithmen beschrieben werden.
<b>Verarbeitung</b>	aufwändig aufgrund der Komplexität und des Umfangs von SQL	einfach aufgrund der einfachen Strukturierung	aufwändig aufgrund der Komplexität
<b>Konversion/ Transformation bei der Übernahme</b>	SQL kann aus der Datenbank exportiert werden, aber eine Konversion in standardkonformes SQL ist notwendig.	Ja (Transformation von SQL in XML-Format)	Ja (aufwändige Transformation zu XQuery)
<b>Bereitstellung und Datenzugriff</b>	Konversion (individuelle Anpassung des gespeicherten SQL an den SQL-Dialekt des jeweiligen RDBS)	Eine Transformation ist notwendig	aufwändige Transformation in SQL
	starke Bindung an SQL als Abfragesprache	keine Bindung an eine bestimmte Abfragesprache	starke Bindung an XQuery als Abfragesprache
	keine direkte Suche in den archivierten XML-Daten möglich	keine direkte Suche in den archivierten XML-Daten möglich	direkte Suche in archivierten XML-Daten möglich
<b>Austauschbarkeit</b>	eingeschränkt durch SQL-Dialekte der RDBS	Ja	Ja
<b>Sonstiges</b>		Umfangreiche Struktur durch mögliche Elementredundanzen	

## A-8 Archivpaket

### A-8.1 XML-Schema des Archivformates für Datenbankstruktur und Datenbankdaten

#### struktur\_daten.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--W3C Schema erstellt mit XMLSpy v2009 sp1 (http://www.altova.com)-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:attribute name="SIZE">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value=""/>
        <xs:enumeration value="1"/>
        <xs:enumeration value="11"/>
        <xs:enumeration value="4"/>
        <xs:enumeration value="5"/>
        <xs:enumeration value="50"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="NULL">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="no"/>
        <xs:enumeration value="yes"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:element name="tabelle3-REG">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="feld1"/>
        <xs:element ref="feld2"/>
        <xs:element ref="feld3"/>
        <xs:element ref="feld4"/>
        <xs:element ref="feld5"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="tabelle3">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="tabelle3-REG" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="tabelle2-REG">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="feld1"/>
        <xs:element ref="feld2"/>
        <xs:element ref="feld3"/>
        <xs:element ref="feld4"/>
        <xs:element ref="feld5"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## Anhang

### A-8 Archivpaket: XML-Schema des Archivformates für Datenbankstruktur und Datenbankdaten

---

```
</xs:complexType>
</xs:element>
<xs:element name="tabelle2">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tabelle2-REG" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="tabelle1-REG">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="feld1"/>
      <xs:element ref="feld2"/>
      <xs:element ref="feld3"/>
      <xs:element ref="feld4"/>
      <xs:element ref="feld5"/>
      <xs:element ref="feld6"/>
      <xs:element ref="feld7"/>
      <xs:element ref="feld8"/>
      <xs:element ref="feld9"/>
      <xs:element ref="feld10"/>
      <xs:element ref="feld11"/>
      <xs:element ref="feld12"/>
      <xs:element ref="feld13"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="tabelle1">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tabelle1-REG" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="feld9">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value=""/>
      <xs:enumeration value="1993-08-26"/>
      <xs:enumeration value="2003-12-19"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="feld8">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Atheist"/>
      <xs:enumeration value="Buddhist"/>
      <xs:enumeration value="Katholik"/>
      <xs:enumeration value="Protestant"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="feld7">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="deutsch"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

## Anhang

### A-8 Archivpaket: XML-Schema des Archivformates für Datenbankstruktur und Datenbankdaten

---

```
</xs:element>
<xs:element name="feld6">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="m"/>
      <xs:enumeration value="w"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="feld5" type="xs:string"/>
<xs:element name="feld4" type="xs:string"/>
<xs:element name="feld3" type="xs:string"/>
<xs:element name="feld2" type="xs:string"/>
<xs:element name="feld13">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ledig"/>
      <xs:enumeration value="verheiratet"/>
      <xs:enumeration value="verwitwet"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="feld12">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value=""/>
      <xs:enumeration value="10"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="feld11">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Ja"/>
      <xs:enumeration value="Nein"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="feld10">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value=""/>
      <xs:enumeration value="Aspenstedt"/>
      <xs:enumeration value="Berlin"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="feld1" type="xs:byte"/>
<xs:element name="feld">
  <xs:complexType>
    <xs:attribute name="NAME" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="feld1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="TABLE">
```

## Anhang

### A-8 Archivpaket: XML-Schema des Archivformates für Datenbankstruktur und Datenbankdaten

---

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="COLUMNS"/>
    <xs:element ref="KEYS"/>
  </xs:sequence>
  <xs:attribute name="NAME" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="tabelle1"/>
        <xs:enumeration value="tabelle2"/>
        <xs:enumeration value="tabelle3"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="DESCRIPTION" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Hilfstabelle hat"/>
        <xs:enumeration value="Person"/>
        <xs:enumeration value="Wohnsitz"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="STRUCTURE">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="TABLE" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PKEY">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="feld"/>
    </xs:sequence>
    <xs:attribute name="TYPE" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="simple"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="KEYS">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="PKEY"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DB">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="STRUCTURE"/>
      <xs:element ref="DATA"/>
    </xs:sequence>
  </xs:complexType>
</xs:complexType>
```

## Anhang

### A-8 Archivpaket: XML-Schema des Archivformates für Datenbankstruktur und Datenbankdaten

---

```
</xs:element>
<xs:element name="DATA">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tabelle1"/>
      <xs:element ref="tabelle2"/>
      <xs:element ref="tabelle3"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="COLUMNS">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="COLUMN" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="COLUMN">
  <xs:complexType>
    <xs:attribute name="TYPE" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="char"/>
          <xs:enumeration value="date"/>
          <xs:enumeration value="int"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute ref="SIZE" use="required"/>
    <xs:attribute ref="NULL" use="required"/>
    <xs:attribute name="NAME" type="xs:string" use="required"/>
    <xs:attribute name="DESCRIPTION" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

## A-8.2 XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

```
<?xml version="1.0"?>
<DB xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="struktur_daten.xsd">
  <STRUCTURE>
    <TABLE NAME="tabelle1" DESCRIPTION="Person">
      <COLUMNS>
        <COLUMN NAME="feld1" TYPE="int" SIZE="11" NULL="no"
          DESCRIPTION="Personennummer"/>
        <COLUMN NAME="feld2" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Name"/>
        <COLUMN NAME="feld3" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Vorname"/>
        <COLUMN NAME="feld4" TYPE="date" SIZE="" NULL="yes"
          DESCRIPTION="Geburtsdatum"/>
        <COLUMN NAME="feld5" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Geburtsort"/>
        <COLUMN NAME="feld6" TYPE="char" SIZE="1" NULL="yes"
          DESCRIPTION="Geschlecht"/>
        <COLUMN NAME="feld7" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Staatsangehörigkeit"/>
        <COLUMN NAME="feld8" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Religion"/>
        <COLUMN NAME="feld9" TYPE="date" SIZE="" NULL="yes"
          DESCRIPTION="Todesdatum"/>
        <COLUMN NAME="feld10" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Todesort"/>
        <COLUMN NAME="feld11" TYPE="char" SIZE="4" NULL="yes"
          DESCRIPTION="Wahlrecht"/>
        <COLUMN NAME="feld12" TYPE="int" SIZE="11" NULL="yes"
          DESCRIPTION="Kinder"/>
        <COLUMN NAME="feld13" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Familienstand"/>
      </COLUMNS>
      <KEYS>
        <PKEY TYPE="simple">
          <feld NAME="feld1"/>
        </PKEY>
      </KEYS>
    </TABLE>
    <TABLE NAME="tabelle2" DESCRIPTION="Wohnsitz">
      <COLUMNS>
        <COLUMN NAME="feld1" TYPE="int" SIZE="11" NULL="no"
          DESCRIPTION="Wohnsitznummer"/>
        <COLUMN NAME="feld2" TYPE="char" SIZE="5" NULL="yes"
          DESCRIPTION="Art"/>
        <COLUMN NAME="feld3" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Stadt"/>
        <COLUMN NAME="feld4" TYPE="int" SIZE="11" NULL="yes"
          DESCRIPTION="Postleitzahl"/>
        <COLUMN NAME="feld5" TYPE="char" SIZE="50" NULL="yes"
          DESCRIPTION="Adresse"/>
      </COLUMNS>
      <KEYS>
```

## Anhang

### A-8 Archivpaket: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

---

```
<PKEY TYPE="simple">
  <feld NAME="feld1"/>
</PKEY>
</KEYS>
</TABLE>
<TABLE NAME="tabelle3" DESCRIPTION="Hilfstabelle hat">
  <COLUMNS>
    <COLUMN NAME="feld1" TYPE="int" SIZE="11" NULL="no"
      DESCRIPTION="Hat-Nummer"/>
    <COLUMN NAME="feld2" TYPE="int" SIZE="11" NULL="no"
      DESCRIPTION="Personennummer"/>
    <COLUMN NAME="feld3" TYPE="int" SIZE="11" NULL="no"
      DESCRIPTION="Wohnsitznummer"/>
    <COLUMN NAME="feld4" TYPE="date" SIZE="" NULL="yes"
      DESCRIPTION="Einzugsdatum"/>
    <COLUMN NAME="feld5" TYPE="date" SIZE="" NULL="yes"
      DESCRIPTION="Auszugsdatum"/>
  </COLUMNS>
  <KEYS>
    <PKEY TYPE="simple">
      <feld NAME="feld1"/>
    </PKEY>
  </KEYS>
</TABLE>
</STRUCTURE>
<DATA>
  <tabelle1>
    <tabelle1-REG>
      <feld1>1</feld1>
      <feld2>Löwe</feld2>
      <feld3>Ralf</feld3>
      <feld4>1939-06-06</feld4>
      <feld5>Aspenstedt</feld5>
      <feld6>w</feld6>
      <feld7>deutsch</feld7>
      <feld8>Atheist</feld8>
      <feld9>1993-08-26</feld9>
      <feld10>Aspenstedt</feld10>
      <feld11>Ja</feld11>
      <feld12/>
      <feld13>ledig</feld13>
    </tabelle1-REG>
    <tabelle1-REG>
      <feld1>2</feld1>
      <feld2>Lange</feld2>
      <feld3>Johanna</feld3>
      <feld4>1984-03-16</feld4>
      <feld5>Potsdam</feld5>
      <feld6>w</feld6>
      <feld7>deutsch</feld7>
      <feld8>Katholik</feld8>
      <feld9/>
      <feld10/>
      <feld11>Nein</feld11>
      <feld12/>
      <feld13>verheiratet</feld13>
    </tabelle1-REG>
    <tabelle1-REG>
      <feld1>3</feld1>
      <feld2>Schreiber</feld2>
```

## Anhang

### A-8 Archivpaket: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

---

```
<feld3>Monika</feld3>
<feld4>1951-02-13</feld4>
<feld5>Potsdam</feld5>
<feld6>w</feld6>
<feld7>deutsch</feld7>
<feld8>Katholik</feld8>
<feld9/>
<feld10/>
<feld11>Ja</feld11>
<feld12/>
<feld13>verheiratet</feld13>
</tabelle1-REG>
<tabelle1-REG>
  <feld1>4</feld1>
  <feld2>Weiß</feld2>
  <feld3>Luca</feld3>
  <feld4>1968-06-03</feld4>
  <feld5>Berlin</feld5>
  <feld6>w</feld6>
  <feld7>deutsch</feld7>
  <feld8>Protestant</feld8>
  <feld9/>
  <feld10/>
  <feld11>Ja</feld11>
  <feld12/>
  <feld13>verheiratet</feld13>
</tabelle1-REG>
<tabelle1-REG>
  <feld1>5</feld1>
  <feld2>Theiss</feld2>
  <feld3>Mario</feld3>
  <feld4>1970-01-02</feld4>
  <feld5>Eldingen</feld5>
  <feld6>m</feld6>
  <feld7>deutsch</feld7>
  <feld8>Atheist</feld8>
  <feld9/>
  <feld10/>
  <feld11>Ja</feld11>
  <feld12/>
  <feld13>verheiratet</feld13>
</tabelle1-REG>
<tabelle1-REG>
  <feld1>6</feld1>
  <feld2>Fuhrmann</feld2>
  <feld3>Ulrike</feld3>
  <feld4>1983-07-15</feld4>
  <feld5>Aitrang</feld5>
  <feld6>w</feld6>
  <feld7>deutsch</feld7>
  <feld8>Protestant</feld8>
  <feld9>2003-12-19</feld9>
  <feld10>Berlin</feld10>
  <feld11>Ja</feld11>
  <feld12/>
  <feld13>verwitwet</feld13>
</tabelle1-REG>
<tabelle1-REG>
  <feld1>7</feld1>
  <feld2>Maur</feld2>
```

## Anhang

### A-8 Archivpaket: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

---

```
<feld3>Klaudia</feld3>
<feld4>1973-09-15</feld4>
<feld5>Behringen</feld5>
<feld6>w</feld6>
<feld7>deutsch</feld7>
<feld8>Atheist</feld8>
<feld9/>
<feld10/>
<feld11>Ja</feld11>
<feld12>10</feld12>
<feld13>ledig</feld13>
</tabelle1-REG>
<tabelle1-REG>
  <feld1>8</feld1>
  <feld2>Kunze</feld2>
  <feld3>Jan</feld3>
  <feld4>1948-01-28</feld4>
  <feld5>Würzburg</feld5>
  <feld6>m</feld6>
  <feld7>deutsch</feld7>
  <feld8>Atheist</feld8>
  <feld9/>
  <feld10/>
  <feld11>Ja</feld11>
  <feld12/>
  <feld13>ledig</feld13>
</tabelle1-REG>
<tabelle1-REG>
  <feld1>9</feld1>
  <feld2>Aachen</feld2>
  <feld3>Kevin</feld3>
  <feld4>1978-12-08</feld4>
  <feld5>Berlin</feld5>
  <feld6>m</feld6>
  <feld7>deutsch</feld7>
  <feld8>Buddhist</feld8>
  <feld9/>
  <feld10/>
  <feld11>Ja</feld11>
  <feld12/>
  <feld13>verheiratet</feld13>
</tabelle1-REG>
<tabelle1-REG>
  <feld1>10</feld1>
  <feld2>Bieber</feld2>
  <feld3>Julia</feld3>
  <feld4>1989-04-05</feld4>
  <feld5>Behringen</feld5>
  <feld6>w</feld6>
  <feld7>deutsch</feld7>
  <feld8>Katholik</feld8>
  <feld9/>
  <feld10/>
  <feld11>Ja</feld11>
  <feld12/>
  <feld13>ledig</feld13>
</tabelle1-REG>
</tabelle1>
<tabelle2>
  <tabelle2-REG>
```

## Anhang

### A-8 Archivpaket: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

---

```
<feld1>1</feld1>
<feld2>Haupt</feld2>
<feld3>Aspenstedt</feld3>
<feld4>38822</feld4>
<feld5>Feldstr. 5</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>2</feld1>
  <feld2>Neben</feld2>
  <feld3>Aspenstedt</feld3>
  <feld4>38823</feld4>
  <feld5>Feldstr. 5</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>3</feld1>
  <feld2>Haupt</feld2>
  <feld3>Aitrang</feld3>
  <feld4>87648</feld4>
  <feld5>Ufnaustr. 24</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>4</feld1>
  <feld2>Haupt</feld2>
  <feld3>Behringen</feld3>
  <feld4>99326</feld4>
  <feld5>Königstr. 52</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>5</feld1>
  <feld2>Haupt</feld2>
  <feld3>Eldingen</feld3>
  <feld4>29351</feld4>
  <feld5>Luckenwalder Str. 86</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>6</feld1>
  <feld2>Neben</feld2>
  <feld3>Eisfeld</feld3>
  <feld4>98671</feld4>
  <feld5>Eschenweg 46</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>7</feld1>
  <feld2>Haupt</feld2>
  <feld3>Berlin</feld3>
  <feld4>10439</feld4>
  <feld5>Trellenborger Str. 3</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>8</feld1>
  <feld2>Haupt</feld2>
  <feld3>Berlin</feld3>
  <feld4>10179</feld4>
  <feld5>Klosterstr. 3</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>9</feld1>
  <feld2>Neben</feld2>
  <feld3>Berlin</feld3>
  <feld4>13089</feld4>
```

## Anhang

### A-8 Archivpaket: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

---

```
<feld5>Indunastr.105</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>10</feld1>
  <feld2>Haupt</feld2>
  <feld3>Potsdam</feld3>
  <feld4>14469</feld4>
  <feld5>Aalsteig 31</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>11</feld1>
  <feld2>Haupt</feld2>
  <feld3>Potsdam</feld3>
  <feld4>14469</feld4>
  <feld5>Florastr. 1</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>12</feld1>
  <feld2>Neben</feld2>
  <feld3>Potsdam</feld3>
  <feld4>14469</feld4>
  <feld5>Aalsteig 31</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>13</feld1>
  <feld2>Haupt</feld2>
  <feld3>Berlin</feld3>
  <feld4>13355</feld4>
  <feld5>Usedomerstr. 23</feld5>
</tabelle2-REG>
<tabelle2-REG>
  <feld1>14</feld1>
  <feld2>Haupt</feld2>
  <feld3>Würzburg</feld3>
  <feld4>97015</feld4>
  <feld5>Mühlenstr. 49</feld5>
</tabelle2-REG>
</tabelle2>
<tabelle3>
  <tabelle3-REG>
    <feld1>1</feld1>
    <feld2>1</feld2>
    <feld3>1</feld3>
    <feld4>1954-08-15</feld4>
    <feld5>1993-08-26</feld5>
  </tabelle3-REG>
  <tabelle3-REG>
    <feld1>2</feld1>
    <feld2>6</feld2>
    <feld3>2</feld3>
    <feld4>2001-08-14</feld4>
    <feld5>2003-12-22</feld5>
  </tabelle3-REG>
  <tabelle3-REG>
    <feld1>3</feld1>
    <feld2>6</feld2>
    <feld3>3</feld3>
    <feld4>1983-07-15</feld4>
    <feld5>2003-12-22</feld5>
  </tabelle3-REG>
</tabelle3>
```

## Anhang

### A-8 Archivpaket: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

---

```
<tabelle3-REG>
  <feld1>4</feld1>
  <feld2>2</feld2>
  <feld3>7</feld3>
  <feld4>2000-04-07</feld4>
  <feld5/>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>5</feld1>
  <feld2>3</feld2>
  <feld3>10</feld3>
  <feld4>1951-02-13</feld4>
  <feld5>1970-09-14</feld5>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>6</feld1>
  <feld2>3</feld2>
  <feld3>8</feld3>
  <feld4>1970-09-14</feld4>
  <feld5/>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>7</feld1>
  <feld2>4</feld2>
  <feld3>10</feld3>
  <feld4>1990-11-08</feld4>
  <feld5/>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>8</feld1>
  <feld2>5</feld2>
  <feld3>5</feld3>
  <feld4>1988-03-09</feld4>
  <feld5/>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>9</feld1>
  <feld2>5</feld2>
  <feld3>9</feld3>
  <feld4>2001-09-01</feld4>
  <feld5>2008-06-16</feld5>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>10</feld1>
  <feld2>7</feld2>
  <feld3>4</feld3>
  <feld4>1980-10-24</feld4>
  <feld5/>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>11</feld1>
  <feld2>8</feld2>
  <feld3>14</feld3>
  <feld4>1948-01-28</feld4>
  <feld5/>
</tabelle3-REG>
<tabelle3-REG>
  <feld1>12</feld1>
  <feld2>9</feld2>
  <feld3>13</feld3>
```

## Anhang

### A-8 Archivpaket: XML-Archivdatei mit Datenbankstruktur und Datenbankdaten

---

```
        <feld4>1986-01-27</feld4>
        <feld5/>
    </tabelle3-REG>
    <tabelle3-REG>
        <feld1>13</feld1>
        <feld2>9</feld2>
        <feld3>6</feld3>
        <feld4>2003-06-27</feld4>
        <feld5/>
    </tabelle3-REG>
    <tabelle3-REG>
        <feld1>14</feld1>
        <feld2>10</feld2>
        <feld3>12</feld3>
        <feld4>2008-08-01</feld4>
        <feld5/>
    </tabelle3-REG>
    <tabelle3-REG>
        <feld1>15</feld1>
        <feld2>10</feld2>
        <feld3>4</feld3>
        <feld4>1989-04-05</feld4>
        <feld5/>
    </tabelle3-REG>
    <tabelle3-REG>
        <feld1>16</feld1>
        <feld2>3</feld2>
        <feld3>11</feld3>
        <feld4>1967-11-01</feld4>
        <feld5/>
    </tabelle3-REG>
</tabelle3>
</DATA>
</DB>
```

## A-8.3 Dokumenttypdefinition des XML-basierten Archivformates für Abfragen

### abfragen.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ABFRAGENSAMMLUNG (ABFRAGE+)>
  <!ELEMENT ABFRAGE (PROJEKTION+, SELEKTION*, VERKNÜPFUNG*, GRUPPIERUNG?,
    SORTIERUNG?, LIMITIERUNG?)>
    <!ELEMENT PROJEKTION (ALL | AUSDRUCK | TABELLENSPALTE | WERT)>
      <!ELEMENT ALL EMPTY>
      <!ELEMENT AUSDRUCK (AUSDRUCK | TABELLENSPALTE | WERT)*>
        <!ELEMENT TABELLENSPALTE (#PCDATA)>
        <!ELEMENT WERT (#PCDATA)>
      <!ELEMENT SELEKTION (AUSDRUCK)>
      <!ELEMENT VERKNÜPFUNG (PRIMÄRSCHLÜSSEL, FREMDSCHLÜSSEL+)>
        <!ELEMENT PRIMÄRSCHLÜSSEL (TABELLENSPALTE+)>
        <!ELEMENT FREMDSCHLÜSSEL (TABELLENSPALTE+)>
      <!ELEMENT GRUPPIERUNG (TABELLENSPALTE+, SELEKTION*)>
      <!ELEMENT SORTIERUNG (TABELLENSPALTE | AUSDRUCK)+>
      <!ELEMENT LIMITIERUNG (DATENSATZNUMMER+)>

<!ATTLIST PROJEKTION
  Aliasname CDATA #IMPLIED>

<!ATTLIST ALL
  DB CDATA #IMPLIED>

<!ATTLIST AUSDRUCK
  Typ (Operation | Funktion) #REQUIRED
  Name CDATA #REQUIRED
  Beschreibung CDATA #REQUIRED>

<!ATTLIST TABELLENSPALTE
  DB CDATA #REQUIRED
  Tabelle CDATA #REQUIRED
  Verknüpfungsnummer CDATA #IMPLIED>

<!ATTLIST SELEKTION
  Selektionsnr. ID #IMPLIED
  Nachfolger IDREF #IMPLIED
  Verknüpfung (AND | OR | NOT) #IMPLIED>

<!ATTLIST SORTIERUNG
  Art (ASC | DESC) #REQUIRED>

<!ATTLIST DATENSATZNUMMER
  Art (Anfang | Ende) #REQUIRED>
```

**A-8.4 XML-Archivdatei für das Abfragebeispiel**

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ABFRAGENSAMMLUNG SYSTEM "abfragen.dtd">
<ABFRAGENSAMMLUNG>
  <ABFRAGE>
    <PROJEKTION Aliasname="Wohnort">
      <AUSDRUCK Typ="Funktion" Name="concat" Beschreibung="Verbindet
        mehrere Zeichenketten">
        <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
          feld3
        </TABELLENSPALTE>
        <WERT> (</WERT>
        <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
          feld4
        </TABELLENSPALTE>
        <WERT>)</WERT>
      </AUSDRUCK>
    </PROJEKTION>
    <PROJEKTION Aliasname="Adresse">
      <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
        feld5
      </TABELLENSPALTE>
    </PROJEKTION>
    <PROJEKTION Aliasname="Personenanzahl">
      <AUSDRUCK Typ="Funktion" Name="count" Beschreibung="Ermittelt die
        Anzahl von Werten">
        <TABELLENSPALTE Tabelle="tabelle4" DB="melderegister">
          feld1
        </TABELLENSPALTE>
      </AUSDRUCK>
    </PROJEKTION>
    <VERKNÜPFUNG>
      <PRIMÄRSCHLÜSSEL>
        <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
          feld1
        </TABELLENSPALTE>
      </PRIMÄRSCHLÜSSEL>
      <FREMSCHLÜSSEL>
        <TABELLENSPALTE Tabelle="tabelle4" DB="melderegister">
          feld3
        </TABELLENSPALTE>
      </FREMSCHLÜSSEL>
    </VERKNÜPFUNG>
    <GRUPPIERUNG>
      <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
        feld3</TABELLENSPALTE>
      <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
        feld4</TABELLENSPALTE>
      <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
        feld5
      </TABELLENSPALTE>
    </GRUPPIERUNG>
    <SORTIERUNG Art="DESC">
      <AUSDRUCK Typ="Funktion" Name="count" Beschreibung="Ermittelt die
        Anzahl von Werten">
        <TABELLENSPALTE Tabelle="tabelle4" DB="melderegister">
          feld1
        </TABELLENSPALTE>

```

## Anhang

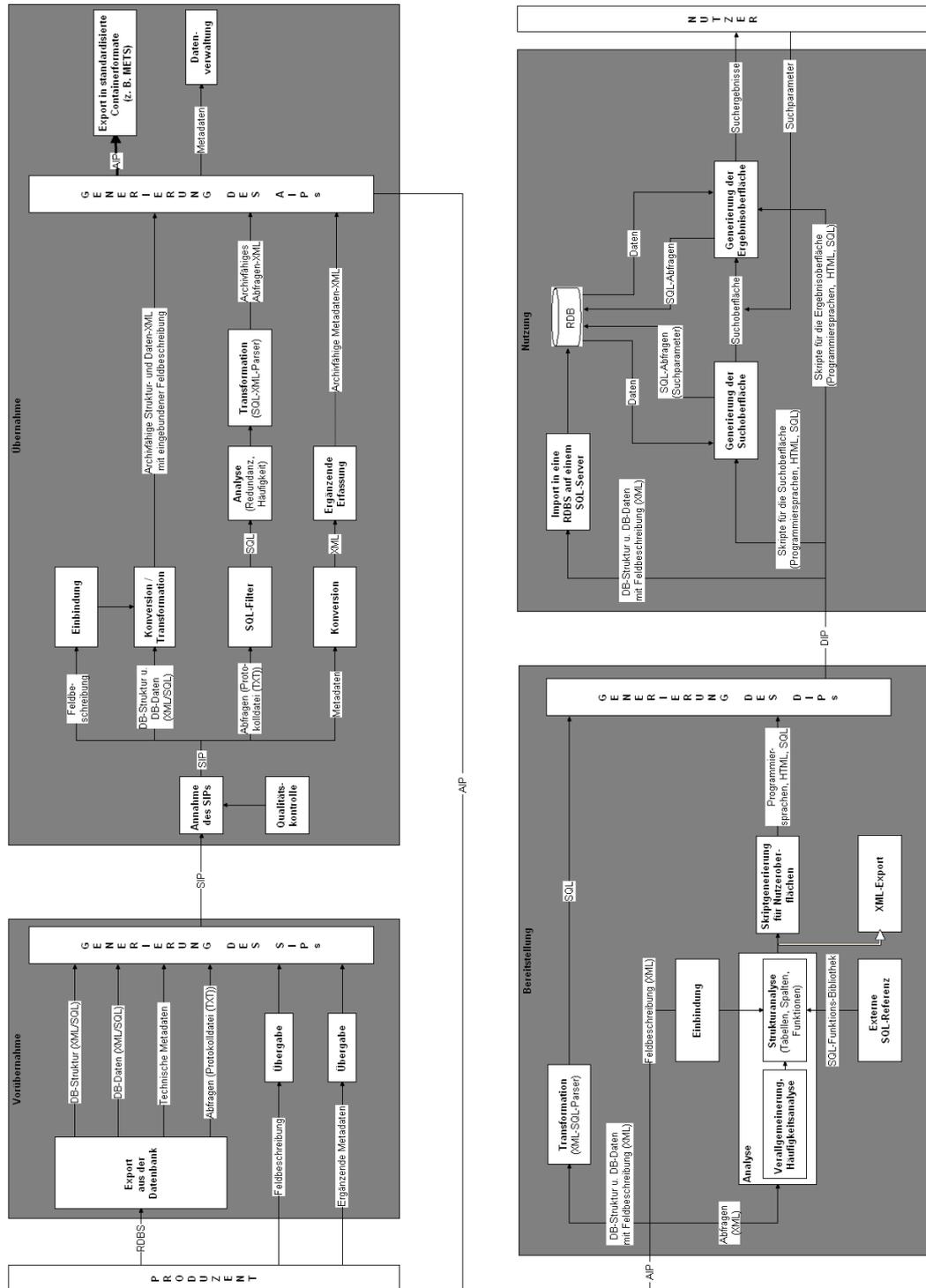
### A-8 Archivpaket: XML-Archivdatei für das Abfragebeispiel

---

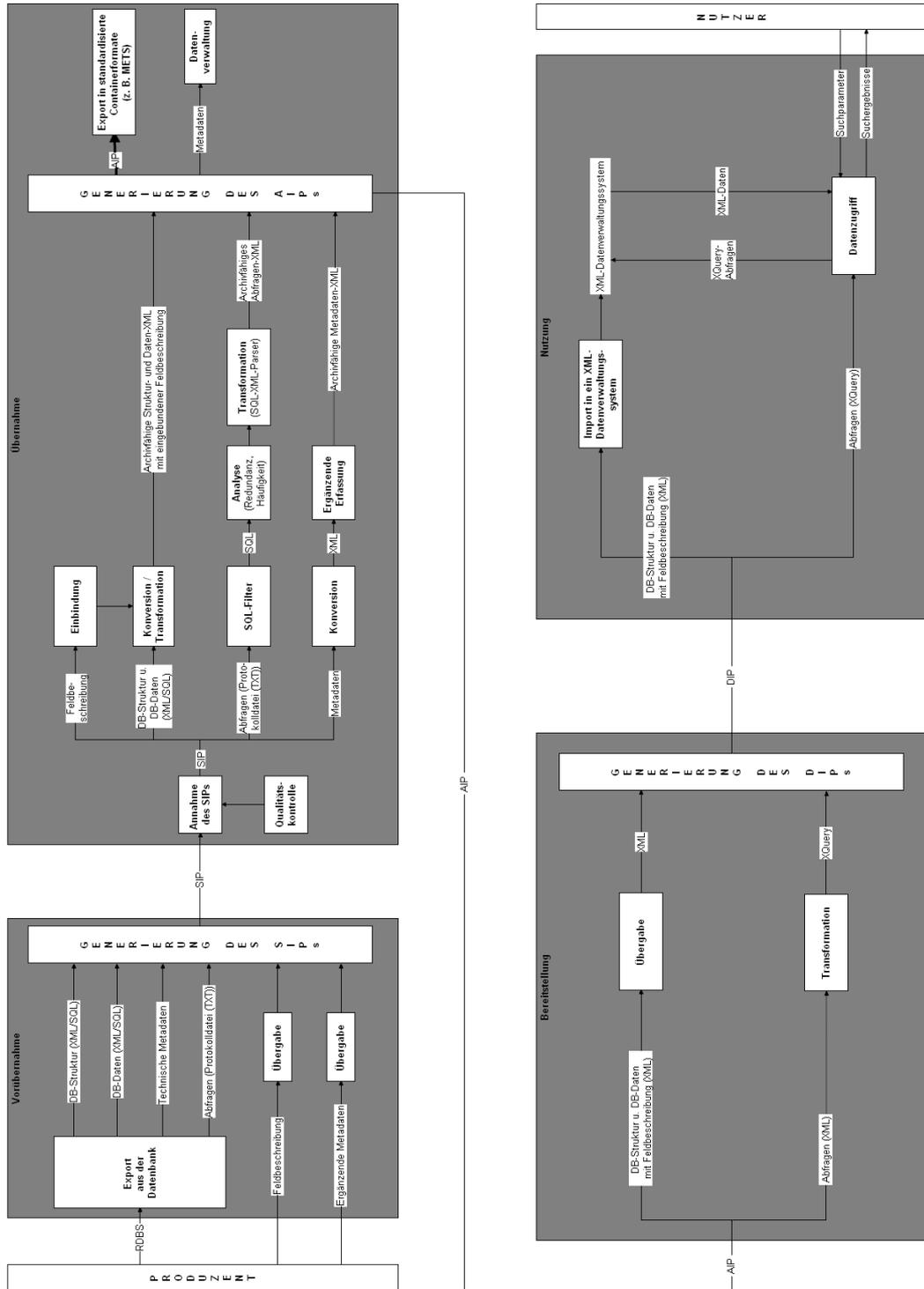
```
</AUSDRUCK>  
</SORTIERUNG>  
</ABFRAGE>  
</ABFRAGENSAMMLUNG>
```

## A-9 Prozessmodelle

### A-9.1 Variante 1: SQL und generische Nutzeroberflächen



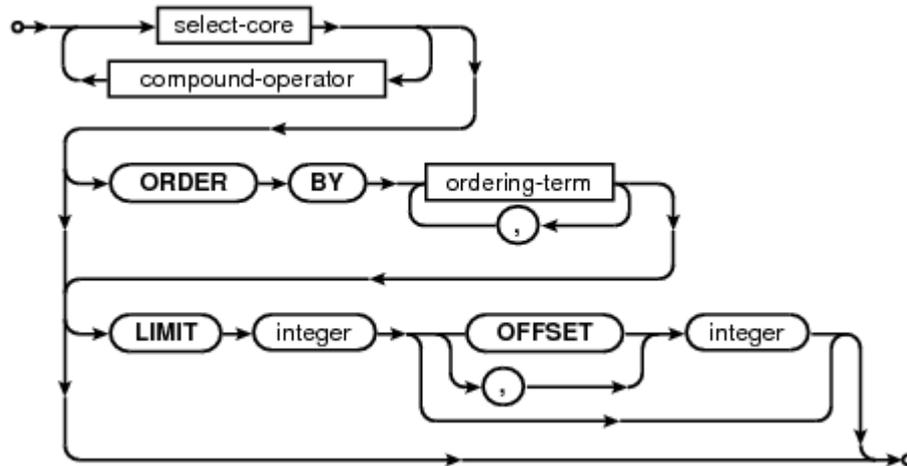
A-9.2 Variante 2: XML und XQuery



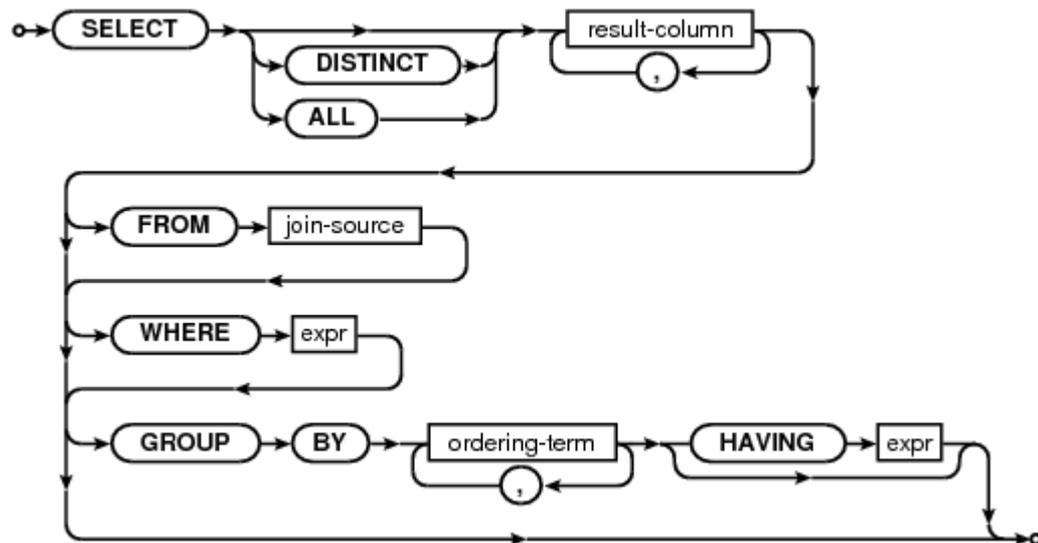


## A-10 SQLite-Syntaxdiagramm des SELECT-Befehls

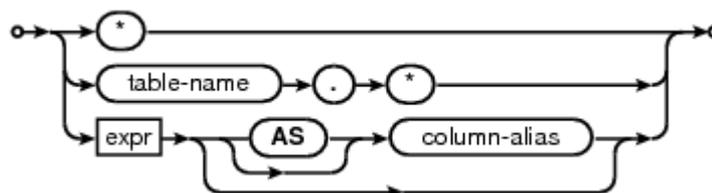
select-stmt:



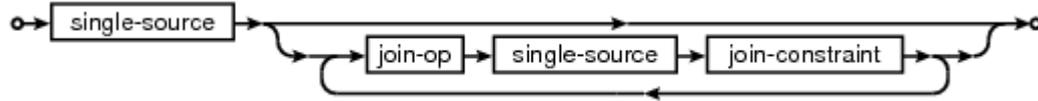
select-core:



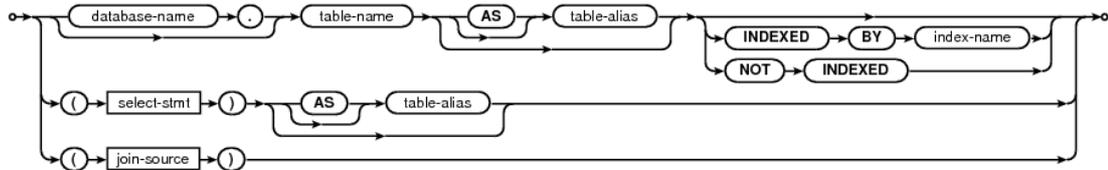
result-column:



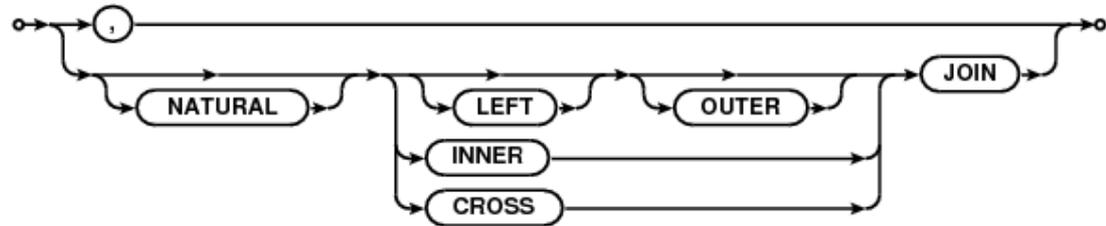
join-source:



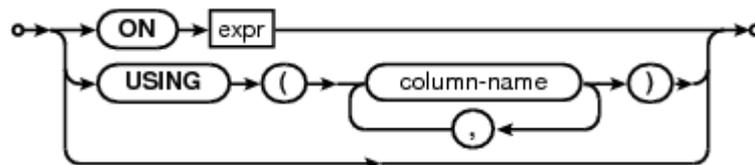
single-source:



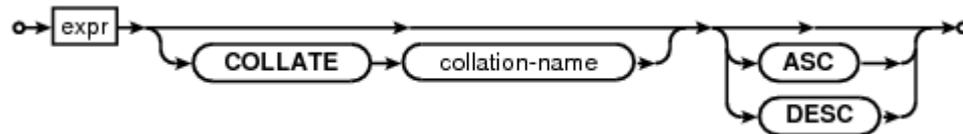
join-op:



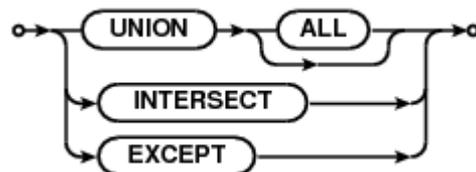
join-constraint:



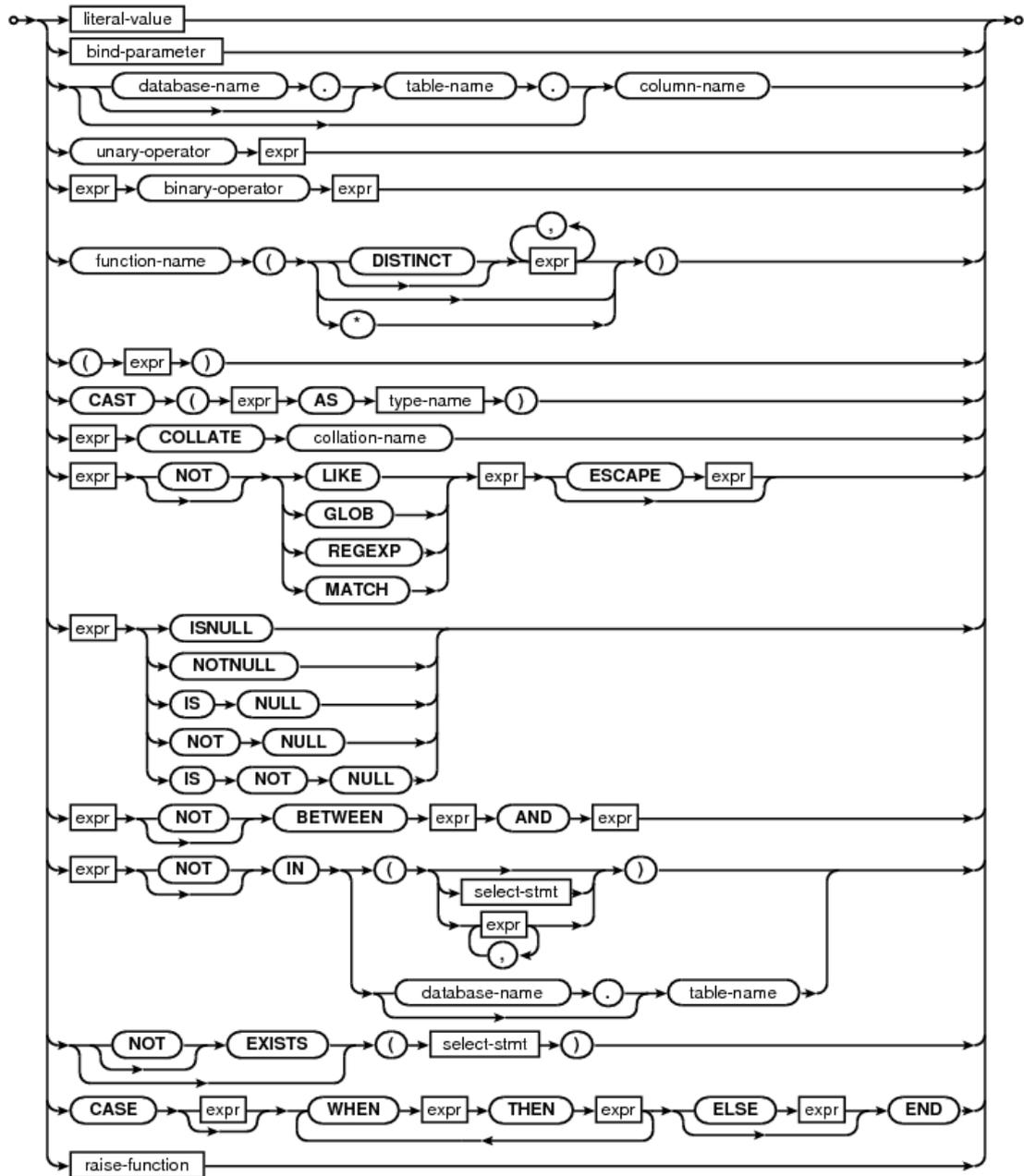
ordering-term:



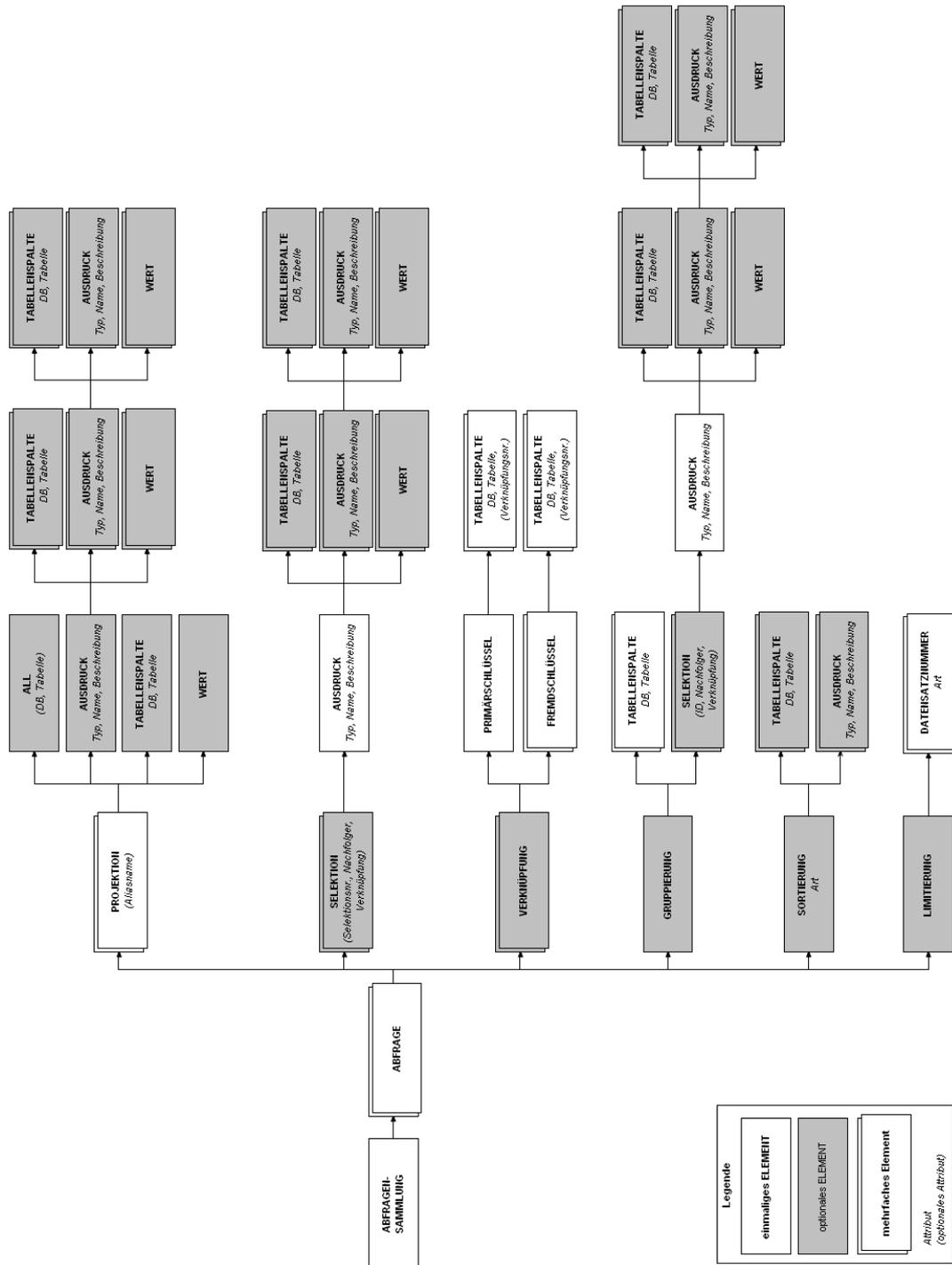
compound-operator:



expr:



## A-11 Detaillierter Strukturbaum der XML-Dokumenttypdefinition für Datenbankabfragen



## A-12 Ergebnistabelle des Abfragebeispiels

Wohnort	Adresse	Personenanzahl
Potsdam (14469)	Aalsteig 31	3
Behringen (99326)	Königstr. 52	2
Aspenstedt (38823)	Feldstr. 5	1
Berlin (10179)	Klosterstr. 3	1
Würzburg (97015)	Mühlenstr. 49	1
Aspenstedt (38822)	Feldstr. 5	1
Potsdam (14469)	Florastr. 1	1
Berlin (10439)	Trellenborger Str. 3	1
Berlin (13089)	Indunastr.105	1
Eisfeld (98671)	Eschenweg 46	1
Aitrang (87648)	Ufnaustr. 24	1
Eldingen (29351)	Luckenwalder Str. 86	1
Berlin (13355)	Usedomerstr. 23	1

## A-13 Ergänzendes Abfragebeispiel mit komplexeren Strukturen

### A-13.1 SELECT-Anweisung und Ergebnistabelle

#### Inhalt der Abfrage

Abfrage des Durchschnittsalters aller, als verheiratet gemeldeter Personen, gruppiert nach Wohnorten.

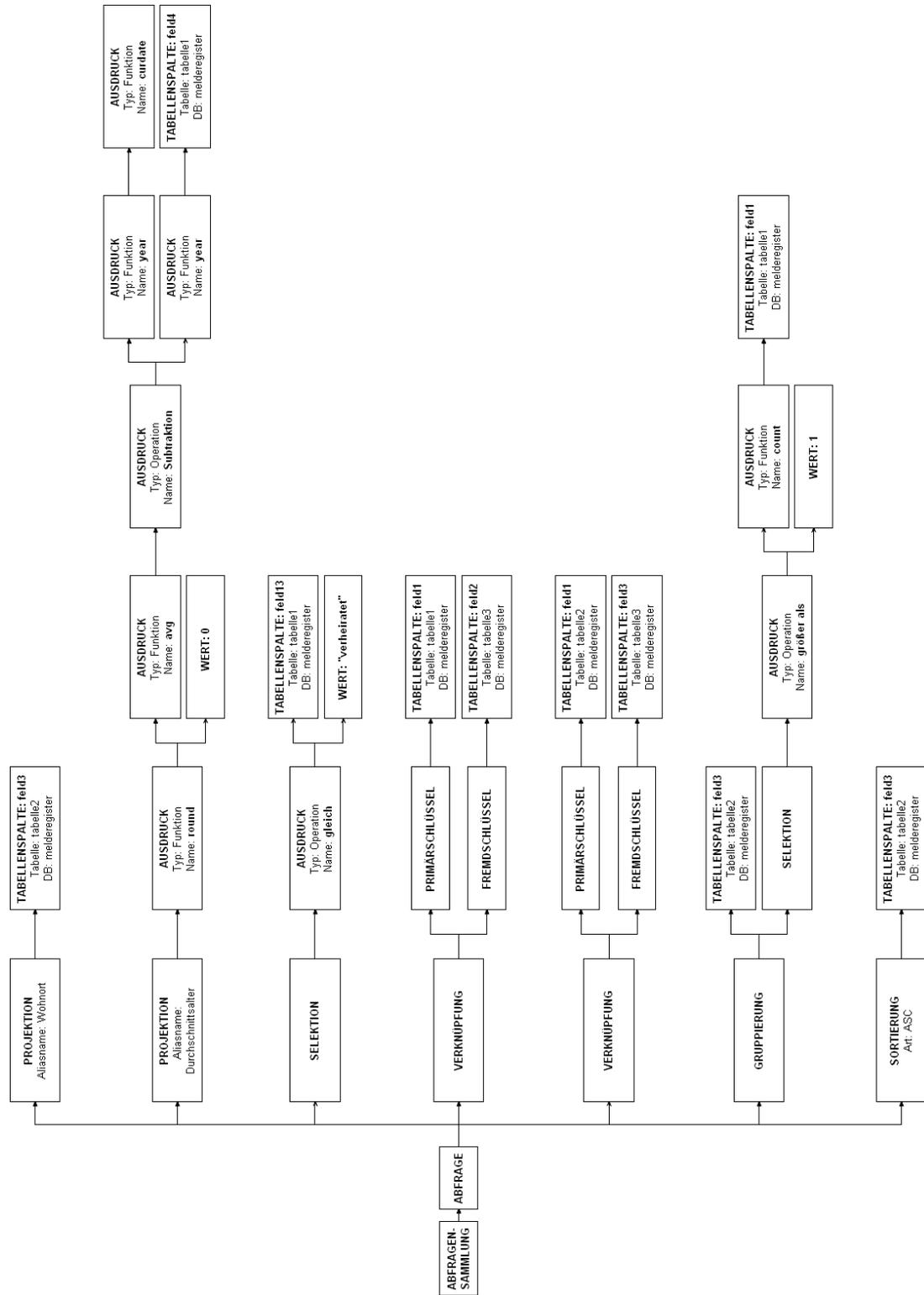
#### SELECT-Anweisung

```
SELECT b.feld3 AS Wohnort,  
       round(avg(year(curdate())-year(a.feld4)),0) AS Durchschnittsalter  
FROM tabelle1 a, tabelle2 b, tabelle3 c  
WHERE a.feld13="verheiratet" AND a.feld1=c.feld2 AND c.feld3=b.feld1  
GROUP BY b.feld3 HAVING count(a.feld1)>1  
ORDER BY b.feld3 ASC;
```

#### Ergebnistabelle

```
+-----+-----+  
| Wohnort | Durchschnittsalter |  
+-----+-----+  
| Berlin  | 38 |  
| Potsdam | 52 |  
+-----+-----+
```

### A-13.2 XML-Strukturbaum



### A-13.3 XML-Archivdatei

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ABFRAGENSAMMLUNG SYSTEM "Abfragen.dtd">
<ABFRAGENSAMMLUNG>
  <ABFRAGE>
    <PROJEKTION Aliasname="Wohnort">
      <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
        feld3
      </TABELLENSPALTE>
    </PROJEKTION>
    <PROJEKTION Aliasname="Durchschnittsalter">
      <AUSDRUCK Typ="Funktion" Name="round" Beschreibung="Rundung">
        <AUSDRUCK Typ="Funktion" Name="avg"
          Beschreibung="Durchschnittsberechnung">
          <AUSDRUCK Typ="Operation" Name="Subtraktion"
            Beschreibung="Rechenoperation">
            <AUSDRUCK Typ="Funktion" Name="year"
              Beschreibung="Extrahiert die Jahreszahl">
              <AUSDRUCK Typ="Funktion" Name="curdate"
                Beschreibung="Ermittelt das aktuellen Datums"/>
            </AUSDRUCK>
            <AUSDRUCK Typ="Funktion" Name="year"
              Beschreibung="Extrahiert die Jahreszahl">
              <TABELLENSPALTE Tabelle="tabelle1" DB="melderegister">
                feld4
              </TABELLENSPALTE>
            </AUSDRUCK>
          </AUSDRUCK>
        </AUSDRUCK>
      </AUSDRUCK>
    </PROJEKTION>
    <SELEKTION>
      <AUSDRUCK Typ="Operation" Name="gleich"
        Beschreibung="Vergleichsoperation">
        <TABELLENSPALTE Tabelle="tabelle1" DB="melderegister">
          feld13
        </TABELLENSPALTE>
        <WERT>verheiratet</WERT>
      </AUSDRUCK>
    </SELEKTION>
    <VERKNÜPFUNG>
      <PRIMÄRSCHLÜSSEL>
        <TABELLENSPALTE Tabelle="tabelle1" DB="melderegister">
          feld1
        </TABELLENSPALTE>
      </PRIMÄRSCHLÜSSEL>
      <FREMDSCHLÜSSEL>
        <TABELLENSPALTE Tabelle="tabelle3" DB="melderegister">
          feld2
        </TABELLENSPALTE>
      </FREMDSCHLÜSSEL>
    </VERKNÜPFUNG>
    <VERKNÜPFUNG>
      <PRIMÄRSCHLÜSSEL>
        <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
          feld1
        </TABELLENSPALTE>

```

## Anhang

### A-13 Ergänzendes Abfragebeispiel mit komplexeren Strukturen: XML-Archivdatei

---

```
</PRIMÄRSCHLÜSSEL>
<FREMSCHLÜSSEL>
  <TABELLENSPALTE Tabelle="tabelle3" DB="melderegister">
    feld3
  </TABELLENSPALTE>
</FREMSCHLÜSSEL>
</VERKNÜPFUNG>
<GRUPPIERUNG>
  <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
    feld3
  </TABELLENSPALTE>
  <SELEKTION>
    <AUSDRUCK Typ="Operation" Name="größer als"
      Beschreibung="Vergleichsoperation">
      <AUSDRUCK Typ="Funktion" Name="count" Beschreibung="Ermittelt
        die Anzahl von Werten">
        <TABELLENSPALTE Tabelle="tabelle1" DB="melderegister">
          feld1
        </TABELLENSPALTE>
      </AUSDRUCK>
      <WERT>1</WERT>
    </AUSDRUCK>
  </SELEKTION>
</GRUPPIERUNG>
<SORTIERUNG Art="ASC">
  <TABELLENSPALTE Tabelle="tabelle2" DB="melderegister">
    feld3
  </TABELLENSPALTE>
</SORTIERUNG>
</ABFRAGE>
</ABFRAGENSAMMLUNG>
```

## Literaturverzeichnis

### Hochschulschriften

- (1) Dehms, Björn:  
Langzeitarchivierung einfacher, relationaler Datenbanken : Entwicklung eines Prototypen zur Migration nach XML. Koblenz-Landau, Univer. - Fachbereich Informatik, Dipl.-Arb., 2000
  
- (2) Glöde, Julia; Richter Doreen:  
Nichtkommerzielle Datenbankarchivierung auf der Basis von XML. Potsdam, FH - Fachbereich Informationswissenschaften, Projektbericht, 2009

### Internetdokumente

- (1) Bundesamt für Sicherheit in der Informationstechnik (Hrsg.):  
M 4.170 Auswahl geeigneter Datenformate für die Archivierung von Dokumenten. Stand: 2005. URL: <http://www.bsi.bund.de/gshb/deutsch/m/m04170.htm> (letzter Zugriff: 13.07.2009). - online im Internet
  
- (2) Consultative Committee for Space Data Systems CCSDS (Hrsg.):  
Reference Model for an Open Archival Information System (OAIS). Stand: Januar 2002. (BLUE BOOK, 1). URL: <http://public.ccsds.org/publications/archive/650x0b1.pdf> (letzter Zugriff: 13.07.2009). - PDF-Dokument
  
- (3) CSP:  
Datenbankarchivierung mit CHRONOS. URL: [http://www.csp-sw.de/de/inhalt.-php?kategorie=c114\\_L\\_sungen\\_CHRONOS](http://www.csp-sw.de/de/inhalt.-php?kategorie=c114_L_sungen_CHRONOS) (letzter Zugriff: 13.07.2009). - online im Internet.

- (4) Digital Preservation Testbed (Hrsg.):  
From digital volatility to digital permanence: Preserving databases. Stand: Dezember 2003. URL: <http://www.digitaleduurzaamheid.nl/bibliotheek/docs/volatility-permanence-databases-en.pdf> (letzter Zugriff: 13.07.2009). - PDF-Dokument.
- (5) ERPANET:  
Electronic Resource Preservation and Access Network. Stand: 2007. URL: <http://www.erpanet.org> (letzter Zugriff: 13.07.2009). - online im Internet
- (6) Hewlett-Packard HP:  
HP Database Archiving software : Data management solutions for application and database environments. Stand: 2009. URL: [http://h18006.www1.hp.com/products/software/im/governance\\_ediscovery/databasearchiving/index.html](http://h18006.www1.hp.com/products/software/im/governance_ediscovery/databasearchiving/index.html) (letzter Zugriff: 13.07.2009). - online im Internet
- (7) National Library of Australia:  
PANDORA Archive. Stand: 16.05. 2008. URL: <http://pandora.nla.gov.au/overview.html> (letzter Zugriff: 13.07.2009). - online im Internet
- (8) nestor-Arbeitsgruppe Vertrauenswürdige Archive - Zertifizierung (Hrsg.):  
nestor-Kriterien : Kriterienkatalog vertrauenswürdiger digitaler Langzeitarchive. Version 2. Stand: 2008. (nestor-materialien, 8). URL: [http://www.langzeitarchivierung.de/downloads/mat/nestor\\_mat\\_08.pdf](http://www.langzeitarchivierung.de/downloads/mat/nestor_mat_08.pdf) (letzter Zugriff: 13.07.2009). - PDF-Dokument
- (9) nestor - Kompetenznetzwerk Langzeitarchivierung (Hrsg.):  
nestor-Glossar. URL: <http://www.langzeitarchivierung.de/index.php?module=Encyclopedia&func=lettersearch&letterget=All&vid=1> (letzter Zugriff: 13.07.2009). - online im Internet
- Langzeitarchivierung. Stand: 20.02.2004. URL: <http://www.langzeitarchivierung.de/index.php?module=Encyclopedia&func=displayterm&id=38&vid=1> (letzter Zugriff: 13.07.2009)

- Langzeitverfügbarkeit. Stand: 01.01.2004. URL: <http://www.langzeitarchivierung.de/index.php?module=Encyclopedia&func=displayterm&id=7&vid=1> (letzter Zugriff: 13.07.2009)
  - Refreshing. Stand: 01.01.2004. URL: <http://langzeitarchivierung.de/index.-php?module=Encyclopedia&func=displayterm&id=10&vid=1> (letzter Zugriff: 13.07.2009)
  - XML. Stand: 01.01.2004. URL: <http://www.langzeitarchivierung.de/index.-php?module=Encyclopedia&func=displayterm&id=20&vid=1> (letzter Zugriff: 13.07.2009)
- (10) nestor - Kompetenznetzwerk Langzeitarchivierung (Hrsg.):  
Memorandum zur Langzeitarchivierung digitaler Informationen in Deutschland.  
Stand: 31.03.2006. URL: <http://www.langzeitarchivierung.de/downloads/memo2006.pdf> (letzter Zugriff: 13.07.2009). - PDF-Dokument
- (11) o. V.:  
Digital Preservation. 23.02.2009. URL: [http://en.wikipedia.org/wiki/Digital\\_preservation](http://en.wikipedia.org/wiki/Digital_preservation) (letzter Zugriff: 13.07.2009). - online im Internet
- (12) o. V.:  
Extensible Markup Language. Stand: 12.03.2009. URL: [http://de.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://de.wikipedia.org/wiki/Extensible_Markup_Language) (letzter Zugriff: 13.07.2009). - online im Internet
- (13) o. V.:  
Langzeitarchivierung. Stand: 21.03.2009. URL: <http://de.wikipedia.org/wiki/Langzeitarchivierung> (letzter Zugriff: 13.07.2009). - online im Internet
- (14) o. V.:  
Request for Comments. Stand: 22.06.2009. URL: [http://de.wikipedia.org/wiki/Request\\_for\\_Comments](http://de.wikipedia.org/wiki/Request_for_Comments) (letzter Zugriff: 10.07.2009). - online im Internet

- (15) o. V.:  
SQLite SQL Language Syntax Diagrams. Stand: 09.10.2008. URL: <http://www.sqlite.org/syntaxdiagrams.html> (letzter Zugriff: 13.07.2009). - online im Internet
- (16) Schweizerisches Bundesarchiv BAR:  
SIARD : Formatbeschreibung. Stand: 30.11.2008. URL: <http://www.bar.admin.ch/aktuell/00568/00702/00703/index.html?lang=de&download=M3wB-PgDB/8u1l6Du36WenojQ1NTTjaXZnqWfVpzLhmfhnapmmc7Zi6rZnqCkklN0h-Hx+bKbXrZ6lhuDZz8mMps2gpKfo> (letzter Zugriff: 13.07.2009). - PDF-Dokument
- (17) Shafranovich, Yakov:  
Common Format and MIME Type for Comma-Separated Values (CSV) Files. Stand: Oktober 2005. URL: <http://tools.ietf.org/html/rfc4180> (letzter Zugriff: 13.07.2009). - online im Internet
- (18) The U.S. National Archives and Records Administration:  
The National Archives Electronic Records Archives (ERA). Stand: 30.06.2009. URL: <http://www.archives.gov/era/> (letzter Zugriff: 13.07.2009). - online im Internet
- (19) The National Archives:  
NDAD : The National Digital Archive of Datasets. Stand: 2009. URL: <http://www.ndad.nationalarchives.gov.uk/> (letzter Zugriff: 13.07.2009). - online im Internet
- (20) Worm, Peter:  
Handreichung für die Archivierung von digitalen Daten aus Kommunalverwaltungen. Version 0.3. Stand: 12.07.2007. URL: [http://www.archive.nrw.de/Archivaemter/WAA/BilderKartenLogosDateien/Handbuch\\_Archivierung\\_von\\_Fachverfahren\\_\\_Vers\\_0\\_3.pdf](http://www.archive.nrw.de/Archivaemter/WAA/BilderKartenLogosDateien/Handbuch_Archivierung_von_Fachverfahren__Vers_0_3.pdf) (letzter Zugriff: 13.07.2009). - PDF-Dokument

### Konferenz- und Tagungsbeiträge

- (1) Cartujano Escobar, Francisco Javier; Espinosa, Enrique David; Lozano, Rafael:  
XML Information Retrieval Using SQL2XQuery. St. Thomas, 2002 (Bericht im Rahmen der Konferenz "Information and Knowledge Sharing" in St. Thomas, US Virgin Islands 2002). (Verfügbar im Internet: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.4444&rep=rep1&type=pdf>, letzter Zugriff: 13.07.2009, PDF-Dokument)
- (2) Dehms, Björn; Engel, Andreas; MeYER, Ulrich:  
XML - ein Format zur dauerhaften Aufbewahrung und Nutzung einfacher, relationaler Datenbanken?. München, 2003 (Bericht im Rahmen der 5. Tagung des Arbeitskreises "Archivierung von Unterlagen aus digitalen Systemen" in München, Deutschland 2001). (Verfügbar im Internet: [http://www.uni-koblenz.de/~meyer/XML\\_RDB\\_Archivierung.pdf](http://www.uni-koblenz.de/~meyer/XML_RDB_Archivierung.pdf), letzter Zugriff: 13.07.2009, PDF-Dokument)
- (3) Heuscher, Stephan [u. a.]:  
Providing Authentic Long-term Archival Access to Complex Relational Data. Frascati, 2004 (Bericht im Rahmen des European Space Symposiums „Ensuring Long-term Preservation and Adding Value to Scientific and Technical Data“ in Frascati, Italien 2004). (Verfügbar im Internet: [http://arxiv.org/PS\\_cache/cs/pdf/0408/0408054v1.pdf](http://arxiv.org/PS_cache/cs/pdf/0408/0408054v1.pdf), letzter Zugriff: 13.07.2009, PDF-Dokument)
- (4) Jahnkuhn, Heiko; Bruder, Ilvio; Balouch, Ammar S.:  
Transformation von SQL in XQuery-Anfragen innerhalb föderierter Informationssysteme. Wörlitz, 2005 (Bericht im Rahmen des 17. GI-Workshops über Grundlagen von Datenbanken in Wörlitz, Deutschland 2005). (Verfügbar im Internet: [http://dbs.informatik.uni-halle.de/GvD2005/beitraege/gvd05\\_JahBru-Bal.pdf](http://dbs.informatik.uni-halle.de/GvD2005/beitraege/gvd05_JahBru-Bal.pdf), letzter Zugriff: 13.07.2009, PDF-Dokument)

- (5) Ramalho, José Carlos [u. a.]:  
Relational database preservation through XML modelling. Montréal, 2007 (Bericht im Rahmen der Konferenz "Extreme Markup Languages" in Montréal, Québec 2007). (Verfügbar im Internet: <http://repositorium.sdum.uminho.pt/bitstream/1822/7120/1/EML2007-final.pdf>, letzter Zugriff: 13.07.2009, PDF-Dokument)
- (6) Scheffel, Regine:  
Einführung in die digitale Langzeitarchivierung. Staufen, 2009 (Vortrag auf der nestor Spring-School in Staufen, Deutschland 2009)
- (7) Verdegem, Remco:  
Database Preservation Issues. Bern, 2003. (Bericht im Rahmen des ERPANET Workshops „Long-term Preservation of Databases“ in Bern, Schweiz 2003). (Verfügbar im Internet: [http://www.digitaleduurzaamheid.nl/bibliotheek/docs/longterm\\_preservation\\_of\\_databases.pdf](http://www.digitaleduurzaamheid.nl/bibliotheek/docs/longterm_preservation_of_databases.pdf), letzter Zugriff: 13.07.2009, PDF-Dokument)

### Monografien

- (1) Däßler, Rolf:  
Das Einsteigerseminar MySQL 5. Heidelberg : bhv, 2005
- (2) Kazakos, Wassilios; Schmidt, Andreas; Tomczyk, Peter:  
XML und Datenbanken : Konzepte, Anwendungen, Systeme. Berlin: Springer, 2002
- (3) Regionales Rechenzentrum für Niedersachsen (Hrsg.):  
XML 1.1 : Grundlagen. 6., unveränd. Aufl.. Hannover: Leibniz Universität, 2006

### Sammelwerkbeiträge

- (4) Neuroth, Heike [u. a.] (Hrsg.):  
nestor Handbuch : Eine kleine Enzyklopädie der digitalen Langzeitarchivierung. Version 1.5. Stand: Dezember 2008. (Verfügbar im Internet: <http://nestor.sub.uni-goettingen.de/handbuch/nestor-handbuch.pdf>, letzter Zugriff: 13.07.2009, PDF-Dokument)
- Brübach, Nils: Das Referenzmodell OAIS - Open Archival Information System, S. 7-1 – 7-12
  - Funk, Stefan E.: Emulation, S. 12-16 – 12-23
  - Funk, Stefan E.: Migration, S. 12-10 – 12-15
  - Liegmann, Hans; Neuroth, Ulrike: Einführung, S. 1-1 – 1-9
- (5) Schwens, Ute; Liegmann, Hans:  
Langzeitarchivierung digitaler Ressourcen. In: Rainer Kuhlen [u. a.] (Hrsg.): Handbuch zur Einführung in die Informationswissenschaft und -praxis. 5., völlig neu gefasste Ausg.. München : Saur, 2004, S. 567-570 (Grundlagen der praktischen Information und Dokumentation, Bd. 1)

### Glossar

#### **Anwendungsprogramm (*Application Programming Interface (API)*)**

Eine Programmschnittstelle ermöglicht die Kommunikation mit dem Datenbanksystem. Sie enthält die Datenbanklogistik in Form der Abfragen, Datenverarbeitung und Nutzerschnittstellen.

#### **Archivpaket (*Archival Information Package (AIP)*)**

Das Archivpaket ist eines von drei verschiedenen Informationspaketen des OAIS-Modells. Es stellt im Kern die Form da, in der die Information langfristig aufbewahrt werden soll. Der Aufbau ist an festgelegten Standards orientiert, die Informationen zur Beschreibung der Aufbewahrungsform und die beschreibenden Informationen sind ausführlich und vollständig vorhanden.

#### **Attribut**

Als Attribut wird im relationalen Datenmodell eine Tabellenspalte bezeichnet, die die Eigenschaft eines bestimmten Objektes beschreibt.

#### **Auslieferungspaket (*Dissemination Information Package (DIP)*)**

Das Auslieferungspaket ist eines von drei verschiedenen Informationspaketen des OAIS-Modells. Es stellt die Form dar, in der das Archivpaket zugänglich gemacht wird. Die Generierung und Auslieferung des Paketes erfolgt auf Anfrage der Nutzer.

#### **Auszeichnungssprache (*Markup Language*)**

Eine Auszeichnungssprache dient der Dokumentenrepräsentation. Sie formalisiert die Auszeichnung von Dokumenten (*Markup*) und löst sie von System- und Verarbeitungsabhängigkeit. Bedeutende Beispiele sind XML, HTML und SGML. (vgl. dazu Kazakos; Schmidt; Tomczyk: XML und Datenbanken : Konzepte, Anwendungen, Systeme, 2002, S. 8)

### **Authentizität<sup>46</sup>**

Unter Authentizität wird eine Objekteigenschaft verstanden. Ein Objekt ist authentisch, wenn es das darstellt, was es vorgibt darzustellen. Wichtige Aspekte sind die Quelle und Entstehungszeit sowie ein lückenloser Nachweis aller am Objekt durchgeführten Veränderungen.

### **Beschreibende Informationen**

Beschreibende Informationen sind als Metadaten Bestandteil eines Informationspaketes des OAIS-Modells. Sie werden an die Datenverwaltung übergeben und dienen dem Auffinden und Ordnen der archivierten Informationen.

### **Binary Large Object (BLOB)**

BLOBs beschreiben große binäre Datenobjekte, wie Audiodateien oder Grafiken.

### **CSV (*Comma-Separates Values*)**

CSV ist ein kompaktes Dateiformat, das den Aufbau von Textdokumenten beschreibt. Es eignet sich für die Speicherung und den Austausch einfach strukturierter Daten.

### **Datenbank**

Eine Datenbank ist eine Sammlung von Daten, die zu einem bestimmten Zweck zusammengestellt wurde und ist Bestandteil des Datenbanksystems.

### **Datenbankfunktion**

Unter einer Datenbankfunktion werden mehrere Datenbankoperationen unter einem Namen zusammengefasst. Beispiele sind das Runden von Zahlen (`round()`), das Ermitteln des arithmetischen Mittelwerts (`avg()`) und die Ausgabe des aktuellen Datums (`curdate()`).

---

<sup>46</sup> vgl. dazu nestor-Arbeitsgruppe Vertrauenswürdige Archive - Zertifizierung: nestor-Kriterien : Kriterienkatalog vertrauenswürdiger digitaler Langzeitarchive, 2008, S. 24, URL:[http://www.langzeitarchivierung.de/downloads/mat/nestor\\_mat\\_08.pdf](http://www.langzeitarchivierung.de/downloads/mat/nestor_mat_08.pdf)

### **Datenbankkommunikationsschnittstelle (*Database Communication Interface (DCI)*)**

Die Datenbankkommunikationsschnittstelle ist ein Bestandteil des Datenbanksystems und ermöglicht die Kommunikation der Anwender oder Anwenderprogramme mit dem Datenbanksystem.

### **Datenbankoperationen**

Mit Hilfe von Operationen können Berechnungen oder Vergleiche durchgeführt werden. Sie finden in relationalen Datenbank bei mathematischen Aufbereitungen von Tabelleninhalten und beim Formulieren von Bedingungen für die Auswahl von Datensätzen Anwendung. Eine Operation besteht dabei grundlegend aus Operanden und Operatoren.

### **Datenbanksystem (DBS)**

Das Datenbanksystem definiert das Gesamtsystem und besteht aus den drei Komponenten Datenbank, Datenbankverwaltungssystem und Kommunikationsschnittstelle.

### **Datenbankverwaltungssystem (*Database Management System (DBMS)*)**

Das Datenbankverwaltungssystem ist Bestandteil des Datenbanksystems. Es ist die zentrale Steuerungseinheit des Gesamtsystems und zudem die Softwareschnittstelle zwischen der physischen Datenbank und der Kommunikationsebene.

### **Datenbeschreibungssprache (*Data Definition Language (DDL)*)**

Die Datenbeschreibungssprache wird vom Datenbankentwickler und -modellierer für die Definition der Datenbankstruktur verwendet.

### **Datenkontrollsprache (*Data Control Language (DCL)*)**

Die Datenkontrollsprache wird unter anderem für die Vergabe von Zugriffsrechten angewandt.

### **Datenmanipulationssprache (*Data Manipulation Language (DML)*)**

Mit Hilfe der Datenmanipulationssprache erfolgt die Eingabe, Änderung und Löschung von Daten sowie die Formulierung von Abfragen durch den Datenbankadministrator und -nutzer.

### **Datenobjekt**

Das Datenobjekt stellt den eigentlichen Kern der Archivierung dar und ist in digitaler Form aus einer Bitsequenz aufgebaut.

### **DBMA (*Database Markup Language*)**

DBML bezeichnet die im Rahmen des RODA-Projekts zur Datenbankarchivierung entwickelte Auszeichnungssprache für die Beschreibung der Datenbankstruktur und Datenbankdaten.

### **Digitale Langzeitarchivierung**

Unter dem Begriff digitale Langzeitarchivierung werden alle Maßnahmen zusammengefasst, deren Zweck darin besteht, digitale Objekte für die Nachwelt, also über einen langen Zeitraum hinweg, dauerhaft zu erhalten.

### **Dokumenttypdefinition (*Document Type Definition (DTD)*)**

Bei der Dokumenttypdefinition handelt es sich um eine Schemasprache, mit deren Hilfe die Struktur eines XML-Dokumentes und die Semantik der enthaltenen Strukturelemente beschrieben werden. Eine Dokumenttypdefinition ist selbst kein XML-Dokument. Sie verwendet eine SGML-verwandte Syntax.

### **Einlieferungspaket (*Submission Information Package (SIP)*)**

Das Einlieferungspaket ist eines von drei verschiedenen Informationspaketen des OAIS-Modells. Es stellt die Form dar, in der die zu archivierenden Daten vom Produzenten an das Archiv übergeben werden.

### **Emulation**

Die Emulation ist eine Archivierungsstrategie, die sich durch die Bewahrung des digitalen Objekts in seiner ursprünglichen Form auszeichnet. Es wird die interpretierende, originale Nutzungsumgebung nachgebildet.

### **Formatkonversion**

Die Formatkonversion ist eine Form der Migration zur Überführung in ein anderes Datenformat. Sie wird beispielsweise im Rahmen der Normalisierung in einem digitalen Archiv angewandt.

### **Fremdschlüssel**

Fremdschlüssel sind bestimmte Spalten (Attribute) im relationalen Datenmodell, die dem Verweis auf andere Tabellen (Relationen) dienen.

### **Gruppierung**

Die Gruppierung ist ein grundlegendes Element einer Datenbankabfrage, das das Zusammenfassen von Datensätzen mit gleichen Einträgen in bestimmten Spalten erlaubt. Dieses Element findet Anwendung, wenn statistische Berechnungen mit mehreren Datensätzen ausgeführt werden sollen.

### **Informationen zu den Paketbeziehungen**

Die Informationen zu den Paketbeziehungen dienen der eindeutigen Identifikation der Inhaltsinformationen und der entsprechenden Metadaten. Sie stellen eines von vier Elementen der Informationen zur Beschreibung der Aufbewahrungsform dar.

### **Informationen zur Beschreibung der Aufbewahrungsform**

Informationen zur Beschreibung der Aufbewahrungsform sind eine Metadatenart des OAIS-Informationspaketes. Sie bestehen aus vier Elementen, den Provenienz- und Kontextinformationen sowie den Informationen zu den Paketbeziehungen und der Stabilisierung.

### **Informationen zur Stabilisierung**

Informationen zur Stabilisierung stellen sicher, dass keine undokumentierten Veränderungen an den digitalen Objekten durchgeführt werden. Sie stellen eines von vier Elementen der Informationen zur Beschreibung der Aufbewahrungsform dar.

### **Informationen zur Verpackung**

Informationen zur Verpackung sind eine Metadatenart des OAIS-Informationspaketes. Sie werden verwendet um die einzelnen Paketkomponenten zu verbinden und zu identifizieren.

### **Informationsobjekt**

Als Informationsobjekt wird die logisch abgrenzbare Informations- und Verwaltungseinheit in einem Archiv verstanden. Sie werden im OAIS-Modell durch die verschiedenen Informationspakete gebildet.

### **Informationspaket**

Das Informationspaket bildet die Basis des OAIS-Datenmodells. Es besteht aus dem eigentlichen Inhalt (Inhaltsinformationen) und verschiedenen Metadaten (Informationen zur Beschreibung der Aufbewahrungsform, Informationen zu Verpackung).

### **Inhaltsinformationen**

Inhaltsinformationen bestehen aus einem digitalen Datenobjekt und den mit ihm verbundenen Repräsentationsinformationen. Sie bilden das eigentliche Zielobjekt der Archivierung.

### **Integrität<sup>47</sup>**

Unter Integrität wird sowohl die Vollständigkeit als auch die Unversehrtheit der digitalen Objekte verstanden

---

<sup>47</sup> vgl. dazu nestor-Arbeitsgruppe Vertrauenswürdige Archive - Zertifizierung: nestor-Kriterien : Kriterienkatalog vertrauenswürdiger digitaler Langzeitarchive, 2008, S.21

### **Kontextinformationen**

Kontextinformationen beschreiben die Beziehung der Inhaltsinformationen zu ihrer Umgebung. Das beinhaltet Angaben zum Zweck sowie zur Verbindung mit anderen Inhalten. Sie stellen eines von vier Elementen der Informationen zur Beschreibung der Aufbewahrungsform dar.

### **Limitierung**

Die Limitierung ist ein grundlegendes Element einer Datenbankabfrage, mit dem die Anzahl der Datensätze, die als Abfrageergebnis ausgegeben werden sollen, begrenzt werden kann.

### **Metadaten<sup>48</sup>**

Metadaten sind strukturierter Daten über (digitale) Objekte, die der Beschreibung, der Suchbarkeit und dem Auffinden dieser Objekte dienen. Folgende Arten von Metadaten werden unterschieden:

- beschreibende,
- technische,
- administrative,
- rechtliche sowie
- Struktur- und Archivierungsmetadaten.

### **METS (*Metadata Encoding and Transmission Standard*)**

METS ist ein standardisiertes XML-Format, das durch ein XML-Schema definiert wird und der Beschreibung verschiedener digitaler Objektsammlungen dient.

### **Migration**

Die Migration ist eine Archivierungsstrategie, bei der das digitale Objekt so verändert wird, dass es von aktuellen Systemen interpretiert werden kann. Wichtige Arten der Migration sind die Formatkonversion, das Refreshing, die Replikation und das Repackaging.

---

<sup>48</sup> vgl. dazu Scheffel: Einführung in die digitale Langzeitarchivierung, 2009

### **Normalisierung**

Bei der Normalisierung wird die Anzahl der Formate in einem Archiv auf wenige standardisierte Formate reduziert. Dieser Vorgang findet beispielsweise bei der Übernahme und Bereitstellung in einem digitalen Archiv statt.

### **OAIS (*Open Archival Information System*)**

OAIS ist ein internationaler Standard (ISO 1471) in Form eines Referenzmodells, der eine allgemeine Sicht auf die Kernfunktionen eines digitalen Archivs bietet und eindeutig Funktionseinheiten, Schnittstellen und Typen von Informationsobjekten benennt.

### **Parser**

Ein Parser ist ein Programm, das beliebige Eingaben zerlegt und in die Formate umwandelt, die für die Weiterverarbeitung benötigt werden. Während dieses Vorgangs wird die Semantik der Eingabe erschlossen.

### **PREMIS (*Preservation Metadata: Implementation Strategies*)**

PREMIS ist ein XML-Format, das durch ein XML-Schema definiert wird. Es enthält Vorschläge und Empfehlungen zur Implementierung von Metadaten für die Langzeitarchivierung.

### **Primärschlüssel**

Primärschlüssel sind bestimmte Spalten (Attribute) im relationalen Datenmodell, die der eindeutigen Identifikation einer Tabellenzeile (Tupel, Datensatz) dienen.

### **Projektion**

Bei der Projektion handelt es sich um eine Operation der relationalen Algebra, die die Auswahl bestimmter Attribute einer Tabelle erlaubt. Sie wird häufig in Verbindung mit der Selektion angewandt.

### **Protokolldatei**

Die Protokolldatei wird automatisch vom Datenbanksystem gepflegt und verzeichnet Angaben zu den Nutzern und deren An- und Abmeldezeiten. Darüber hinaus dokumentiert sie alle SQL-Befehle, die vom System verarbeitet werden.

### **Provenienz-Informationen**

Provenienz-Informationen dokumentieren den Entstehungs- und Entwicklungsprozess der digitalen Objekte. Sie stellen eines von vier Elementen der Informationen zur Beschreibung der Aufbewahrungsform dar.

### **Prozedur**

Eine Prozedur ist ein gespeichertes und ausgeführtes Programm innerhalb des Datenbankverwaltungssystems, das grundlegende und entscheidende Aufgaben erfüllt.

### **Refreshing<sup>49</sup>**

Das Refreshing ist eine Form der Migration, die der Substanzerhaltung digitaler Objekte dient. Der begrenzten Haltbarkeit von Datenträgern wird durch das Umkopieren auf neue Datenträger(-generationen) begegnet.

### **Relation**

Als Relation wird eine Tabelle im relationalen Datenmodell bezeichnet, die die Daten speichert. Sie beschreibt einen Objekttyp, der mehrere Objekte (Zeilen, Tupel) mit gleichen Eigenschaften (Spalten, Attributen) umfasst.

### **Relationale Algebra**

Die relationale Algebra beschreibt Operationen, die das Datenbankverwaltungssystem ausführt, um auf Daten zuzugreifen, die in verschiedenen Tabellen abgelegt sind.

---

<sup>49</sup> vgl. dazu nestor - Kompetenznetzwerk Langzeitarchivierung: Glossar : Refreshing, 01.01.2004, URL: <http://langzeitarchivierung.de/index.php?module=Encyclopedia&func=displayterm&id=10&vid=1>

### **Repackaging**

Das Repackaging ist eine Form der Migration. Sie ist mit Veränderungen der Informationen zur Verpackung verbunden.

### **Replikation**

Die Replikation ist eine Form der Migration, die im Rahmen der Substanzerhaltung von Bedeutung ist und die mehrfache Speicherung von Daten an typischerweise unterschiedlichen Standorten bezeichnet.

### **Repräsentationsinformationen**

Die Repräsentationsinformationen bilden zusammen mit dem digitalen Datenobjekt die Inhaltsinformationen eines Informationspaketes. Sie versehen das Datenobjekt mit einer ergänzenden Bedeutung. Als Beispiel kann die ASCII-Definition genannt werden. Sie beschreibt, wie eine Bitsequenz (digitales Datenobjekt) als ein Symbol dargestellt wird. Es werden strukturelle und semantische Repräsentationsinformationen unterschieden.

### **Request for Comments (RFC)<sup>50</sup>**

Bei den Request for Comments handelt es sich um eine Reihe von technischen und organisatorischen Dokumenten zum Internet. Der RFC4810 beschreibt das CSV-Format.

### **Selektion**

Die Selektion ist eine der wichtigsten Operationen der relationalen Algebra. Sie ermöglicht das Herausfiltern einer bestimmten Menge von Datensätzen, die bestimmten Bedingungen entsprechen.

### **SGML**

SGML ist ein umfangreicher, internationaler Standard (ISO 8879) für die Beschreibung der Inhaltsstruktur von Dokumenten. Er wurde 1985 verabschiedet.

---

<sup>50</sup> o. V.: Request for Comments, 22.06.2009, URL: [http://de.wikipedia.org/wiki/Request\\_for\\_Comments](http://de.wikipedia.org/wiki/Request_for_Comments)

### **Sicht (*View*)**

Eine Sicht ist eine virtuelle Tabelle, die als Resultat einer vordefinierten Abfrage entsteht.

### **Sortierung**

Die Sortierung ist ein grundlegendes Element einer Datenbankabfrage, mit dem die Ergebnisdatensätze anhand von beliebig vielen Spalten oder Ausdrücken auf- und absteigend sortiert ausgegeben werden können.

### **SQL (*Structured Query Language*)**

SQL ist eine international standardisierte (ISO/IEC 9075), umfassende Skriptsprache zur Definition, Manipulation, Abfrage und Kontrolle von relationalen Datenbanken. Bei der Kommunikation mit dem Datenbanksystem agiert SQL gleichzeitig als Datenbeschreibungs-, Datenmanipulations- und Datenkontrollsprache.

### **SQL/XML**

SQL/XML ist ein internationaler Standard (ISO/IEC 9075-14), der sich als Erweiterung des SQL-Standards damit befasst, wie SQL in Verbindung mit dem XML-Format verwendet werden kann.

### **Trigger**

Trigger sind initiierte und zeitlich festgelegte Prozeduren, die beim Auftreten bestimmter Ereignisse automatisch vom Datenbankverwaltungssystem ausgeführt werden. Ein Beispiel ist das Aufzeichnen und Prüfen von Nutzeraktivitäten.

### **Tupel**

Als Tupel wird im relationalen Datenmodell eine Tabellenzeile bezeichnet, die ein bestimmtes Objekt mit bestimmten Eigenschaften (Attributen) repräsentiert.

### **XML (*Extensible Markup Language*)**

XML ist eine Auszeichnungssprache, die die Syntax vorgibt, mit der die logische Struktur von elektronischen Dokumenten beschrieben werden kann. XML wird auch als Metasprache bezeichnet, da das Format die Definition von weiteren Auszeichnungssprachen ermöglicht.

### **XML-Schema (*XML Schema Definition (XSD)*)**

Das XML-Schema ist eine Schemasprache, mit deren Hilfe die Struktur eines XML-Dokumentes und die Semantik der enthaltenen Strukturelemente beschrieben werden kann. Für die Strukturdefinition wird die XML-Syntax verwendet.

### **XPath (*Extensible Path Language*)**

XPath ist eine Abfragesprache, mit der Teile eines XML-Dokumentes adressiert werden können.

### **XSL (*Extensible Stylesheet Language*)**

XSL ist eine Formatierungssprache, die das Erscheinungsbild von XML-Dokumenten beschreibt.

### **XQuery (*Extensible Query Language*)**

XQuery ist eine Abfragesprache, die XML als Datenmodell verwendet.

## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Alle aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche gekennzeichnet.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht

Berlin, 13. Juli 2009

Julia Glöde