JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Datenqualitätsanalyse beim Ladevorgang in Data Warehouses am Beispiel der ETL-Prozesse der OÖGKK

Diplomarbeit zur Erlangung des akademischen Grades

Magister rer. soc. oec.

im Diplomstudium Wirtschaftsinformatik

angefertigt am

Institut für Wirtschaftsinformatik –
Data & Knowledge Engineering

Eingereicht von

Paul Leitner

Begutachter

o. Univ.-Prof. Dr. Michael Schrefl

Mitbetreuer

Mag. Stefan Berger

Linz, September 2008

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die Diplomarbeit mit dem Titel *Datenqualitätsanalyse beim Ladevorgang in Data Warehouses am Beispiel der ETL-Prozesse der OÖGKK* selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, den 18.09.2008

(Paul Leitner)

Zusammenfassung

Automatisierte Plausibilitäts- und Fehlerkontrollen gewinnen im Zuge der immer größer werdenden Menge an zu verarbeitenden Daten vermehrt an Bedeutung, da dieses Datenaufkommen manuell nicht mehr kontrolliert werden kann. Zu diesem Zweck müssen Fehlerquellen in der eigenen Datenhaltung bekannt sein, sodass sie in weiterer Folge identifiziert und bereinigt werden können. Die Oberösterreichische Gebietskrankenkasse (OÖGKK) sieht durch das wachsende und zu bewältigende Datenmengenaufkommen daher Handlungsbedarf als gegeben an, um die eintreffenden Daten im Zuge des ETL-Prozesses schneller und effizienter analysieren zu können und um auftretende Anomalien zu identifizieren und zu beseitigen.

In dieser Arbeit wird der Begriff Datenqualität definiert. Hierbei wird eine Einteilung in unterschiedliche Datenqualitätsdimensionen vorgenommen. Es werden verschiedene Ursachen für mangelnde Datenqualität vorgestellt und eine Klassifizierung dieser vorgenommen. Außerdem wird ein möglicher Ablauf des Data Cleaning Prozesses dargestellt.

Es werden fünf ausgewählte Werkzeuge auf ihre Funktionalitäten in der Datenqualitätsanalyse sowie der Datenbereinigung untersucht. In weiterer Folge wird die Einsatzfähigkeit der untersuchten Werkzeuge in der OÖGKK diskutiert. Die Arbeit stellt das Konzept eines Softwarewerkzeuges für automatisierte Plausibilitätskontrolle (SofaP) mit dem Einsatzgebiet Oberösterreichische Gebietskrankenkasse vor. SofaP bedient sich vor allem gespeicherter bzw. berechneter Referenz- und Grenzwerte zur Durchführung der Datenbereinigung. Mit Hilfe eines modularen Aufbaus wird eine leichte Erweiterbarkeit des Softwareprogramms zur Verfügung gestellt.

Abstract

Automated data cleaning is becoming more and more important because of the rapidly increasing amounts of data. Manual control of data is very difficult and time consuming. Therefore one must know about the different anomalies that may occur in their data to be able to identify and clean occurring anomalies. The "Oberösterreichische Gebietskrankenkasse" is aware of that problem. Due to that fact it is important to implement a new tool for ETL process control and to handle the huge amount of data faster and more efficient.

This work summarizes data cleaning techniques and reviews how the term data quality is defined in current literature. Therefore a classification of data quality dimensions is made. Furthermore this work presents different occurrences of poor data quality and categorizes them. It also illustrates that data cleaning is an iterative process in all state-of-the-art data cleaning methods.

From the practical viewpoint, this work analyzed and summarizes the methods and processes implemented in five different data cleaning tools. Each tool was evaluated, how efficiently it supports the goals of the ETL process control planned by the OÖGKK. This work contains the specification of a framework for an automated data cleaning process aligned for the OÖGKK (SofaP). Computed or saved reference values and boundary values are used for analyzing and data cleaning. The framework provides modularity. Therefore adaptations can be made easily.

Vorwort

Da geschlechtsspezifische Formulierungen häufig den Lesefluss behindern, wird in der vorliegenden Diplomarbeit auf eine derartige Differenzierung verzichtet. Der Autor möchte jedoch ausdrücklich darauf hinweisen, dass entsprechende Begriffe im Sinne der Gleichbehandlung von Mann und Frau grundsätzlich geschlechtsneutral gemeint sind und für beide Geschlechter gelten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ASCII	American Standard Code for Information Interchange
BI	Business Intelligence
bzw.	beziehungsweise
DB	Datenbank
DBMS	Datenbank-Managementsystem
d.h.	das heißt
DQM	Data Quality Mining
DQMS	Data Quality Management Systems
DS	Datenschnittstelle
DWH	Data Warehouse
ETL	Extract, Transform, Load
FOKO	Folge Kostenanalyse
FTP	File Transfer Protocol
GB	Gigabyte
IS	Information System
LVA	Lehrveranstaltung
OLAP	Online Analytical Processing
OLTP	Online Transactional Processing
OÖGKK	Oberösterreichische Gebietskrankenkasse
PDF	Portable Document Format
PIN	Persönliche Identifikationsnummer
SQL	Structured Query Language
SSIS	SQL Server^{TM}2005 Integration Services
u.ä.	und ähnliches
u.a.	und andere
vgl.	vergleiche
W-LAN	Wireless Lokal Area Network
z.B.	zum Beispiel

1 Einleitung

1.1 Gegenstand und Motivation

Mit Hilfe von Data Warehouses (DWH) sollen gesammelte Daten bereitgestellt und analysiert werden können. Es wird eine homogene und integrierte Datenbasis zur Verfügung gestellt, welche geeignet aufbereitet die Möglichkeit bietet effiziente und zielorientierte Analysen durchzuführen (vgl. [BG04, S. 13]).

Die Oberösterreichische Gebietskrankenkasse (OÖGKK) betreibt ein Data Warehouse. Unterschiedliche Datenmengen werden von verschiedenen Datenquellen im Data Warehouse integriert. Neue Daten werden regelmäßig geladen, ältere können aktualisiert werden. Dieser Integrationsprozess (Extraktion, Transformation, Laden - ETL-Prozess) wird in dieser Arbeit als Black-Box angesehen.

Die stetig wachsende Anzahl von komplexen und laufzeitintensiven ETL-Jobs und die hohen Qualitätsansprüche der „Kunden“ bewirken hohe Anforderungen an die Abwicklung der Befüllung des DWH. Dies erfordert die Durchführung von Plausibilitätskontrollen an den Daten. Die automatisierte Befüllung in den verschiedensten zeitlichen Intervallen kann jedoch nicht durch manuelle Plausibilitätskontrollen geprüft werden. Aus diesem Grund muss ein automatisiertes Framework entwickelt bzw. ein vorhandenes Framework integriert werden, um die gewünschte Qualitätssicherung durchzuführen.

In einem ersten Schritt ist es notwendig mögliche Fehlerquellen zu identifizieren (Kapitel 4), um entsprechende Methoden zur Gegensteuerung finden zu können. Diese Arbeit zeigt auf, wie fünf verschiedene Werkzeuge mit dieser Problematik umgehen (Kapitel 6), um so für die OÖGKK geeignete Funktionen zu identifizieren, welche eine automatisierte Plausibilitätskontrolle gewährleisten können. Es wird untersucht, ob eines dieser Werkzeuge in seinem Umfang für den Einsatz in der OÖGKK geeignet ist (Kapitel 7), oder ob ein neues Framework spezifiziert werden muss (Kapitel 8).

1.2 Aufgabenstellung und Zielsetzung

Aufgabenstellung dieser Diplomarbeit ist die Analyse von Werkzeugen in Bezug auf ihre Funktionen zur Datenbereinigung. Es werden Funktionen zur Datenanalyse und Bereinigung herausgearbeitet und übersichtlich gegenübergestellt. Auf Grund dieser Gegenüberstellung wird ein Werkzeug identifiziert, das am Besten mit den zuvor in der Ist-Analyse identifizierten Fehlerschwerpunkten übereinstimmt. Ist es nicht möglich ein geeignetes Werkzeug zu finden bzw. anzupassen, ist es notwendig ein eigenes Framework zu konzipieren, das die gewünschten Fähigkeiten in sich vereint und in weiterer Folge von der OÖGKK implementiert wird.

Zu diesem Zweck ist es nötig im Vorfeld zuerst die relevanten Grundlagen zu erläutern, sodass diese auf die spezifische Situation der OÖGKK angewendet werden können. Es werden hierbei die in der Literatur vorgestellten Fehlerquellen dargestellt und der Begriff Datenqualität erläutert.

Es wird der OÖGKK mit dieser Arbeit ein Werkzeug zur Datenanalyse und -bereinigung aus einer Auswahl vorgeschlagen oder ein eigenes Framework spezifiziert, welches auf die Anforderungen der OÖGKK abgestimmt ist. Dieses Werkzeug stellt ein Framework dar, das die im Zuge der ETL-Prozesse der OÖGKK auftretenden Fehlerquellen identifiziert und und eine Bearbeitung / Behebung erleichtert. Die Implementierung und Umsetzung einer Eigenlösung ist nicht mehr Teil dieser Diplomarbeit.

1.3 Aufbau der Arbeit

In Kapitel 2 werden die relevanten Grundlagen von Data Warehouses und des Extract-Transform-Load (ETL) Prozesses zusammengefasst. Diese sind die Basis für die Thematik der Diplomarbeit.

Um einen besseren Überblick über die Thematik Datenqualität zu bekommen wird in Kapitel 3 dieser Begriff näher erläutert. In diesem Zuge wird auch ein möglicher Ablauf des Datenbereinigungsprozesses dargestellt. In Kapitel 4 werden zusätzliche Fehlerquellen identifiziert und zur besseren Übersicht klassifiziert.

Anschließend wird in Kapitel 5 eine Ist-Analyse des DWH-Umfeldes OÖGKK präsentiert. Überblicksmäßig wird der jetzige Stand der Technik dargestellt und aufgezeigt, wie der Datenfluss von statten geht, welche Fehlerquellen vorliegen und wie diese zur Zeit behandelt werden.

Kapitel 6 beschäftigt sich mit der Analyse der ausgewählten Datenbereinigungswerkzeuge. Es werden die zur Verfügung gestellten Funktionen je Werkzeug herausgefiltert. Diese werden in Kapitel 7 gegenübergestellt, sodass eine Empfehlung für die weitere Vorgehensweise an die OÖGKK abgegeben werden kann.

Um ein geeignetes Werkzeug für die OÖGKK zur Verfügung zu stellen, wird anschließend in Kapitel 8 ein Softwarewerkzeug spezifiziert. Dieses Werkzeug ist in der Lage die zu verarbeitenden Daten zu analysieren, gegebenenfalls Warnungsmeldungen zu generieren und an den Benutzer zur weiteren Bearbeitung weiterzuleiten. Kapitel 9 beinhaltet eine abschließende Betrachtung.

2 Begriffsbestimmungen

Die Grundlage für diese Arbeit stellt der Bereich des Data Warehousing dar, der in folgendem Abschnitt erläutert wird. In weiterer Folge wird auch auf den „Extract, Transform, Load“ (ETL) Prozess eingegangen, der den Ausgangspunkt der Problemstellung darstellt.

2.1 Data Warehouse

Motivation der Einführung eines Data Warehouse ist eine bessere Unterstützung der Geschäftsabläufe durch Informationstechnologie, da eine Unterstützung für Entscheidungsprozesse durch bereits eingesetzte Informationstechnologie in den meisten Fällen zu gering ist. Operative Informationssysteme unterstützen das tägliche Geschäft. Sie verarbeiten das täglich in den Geschäftsprozessen anfallende Datenaufkommen. Durch eine Datenhaltung in verschiedenen operativen Systemen ist eine integrierte Sicht auf die Daten schwierig. Deshalb ist es notwendig für eine raschere Verfügbarkeit der Daten in Entscheidungsprozessen diese integriert und analysegerecht in einem Data Warehouse abzulegen. Data Warehouses sind analytische Informationssysteme. Abbildung 1 zeigt den Aufbau eines Analytischen Informationssystems nach Chamoni und Gluchowski.

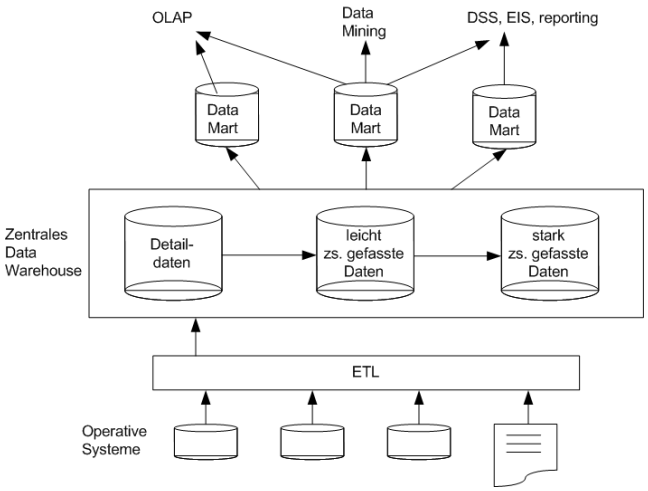

Abbildung 1: Analytisches Informationssystem in Anlehnung an [CG98]

Eine eindeutige Definition des Begriffes Data Warehouse findet sich jedoch in der Literatur nicht. Eine der ersten und auch gebräuchlichsten Definitionen ist jene nach Bill Inmon:

„A data warehouse is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions.“ [IH94, S. 34]

Inmon stellt an ein Data Warehouse somit vier Forderungen. Diese werden hier kurz vorgestellt:

- Themenorientierung (subject-oriented): Daten werden nach den Subjekten eines Unternehmens strukturiert z.B. Kunden, Produkte, Firma. Es wird somit eine Unterstützung von Analysen und Entscheidungsprozessen anstelle von bestimmten Anwendungen bzw. operativen Prozessen zur Verfügung gestellt.
- Vereinheitlichung (integrated): Ein Data Warehouse enthält Daten aus verschieden (operativen) Quellsystemen in integrierter und vereinheitlichter Form.
- Zeitorientierung (time-variant): Mit Hilfe eines Data Warehouses werden Analysen über zeitliche Veränderungen und Entwicklungen durchgeführt. Dazu ist eine langfristige Speicherung der Daten (einige Jahre) unbedingt notwendig. Dies führt zur Einführung der Dimension „Zeit“.
- Beständigkeit (non-volatile): Daten werden in einem Data Warehouse dauerhaft d.h. nicht-flüchtig gespeichert. Daten werden durch Laden aus operativen Systemen in das Data Warehouse nur eingefügt. Der Benutzer hat nur lesenden Zugriff auf die Daten.

Nach Bauer und Günzel [BG04] ist diese Definition einerseits nicht aussagekräftig genug um in Theorie und Praxis verwendet werden zu können. Andererseits schränkt sie dermaßen ein, dass viele Ansätze und Anwendungsgebiete herausfallen. Daher stellen sie folgende Definition auf:

„Ein Data Warehouse ist eine physische Datenbank, die eine integrierte Sicht auf beliebige Daten zu Analysezwecken ermöglicht.“ [BG04, S. 7]

Auch Wieken merkt an, dass es sich bei den Anforderungen von Inmon im Grunde um Anforderungen an die Datenbasis handelt. Anforderungen, die aus Anwendersicht und Informatiksicht zu berücksichtigen sind, sind in den meisten Fällen umfassender [Wie99, S. 16f]. Ein Data Warehouse stellt eine Architektur zur Datenintegration und Analyse von Daten dar. Da sich eine solche zentralistische Lösung in manchen Fällen konzeptuell und technisch als schwer durchsetzbar erweist, kann das Konzept eines Data Mart eingeführt werden [BG04, S. 59ff].

Ein Data Mart stellt im Kern eine Untergruppe der in einem Data Warehouse liegenden Daten dar. Es wird dadurch ein eingeschränkter Fokus des Data Warehouse dargestellt und in einer eigenen Datenbank gespeichert. Dabei ist es wichtig, dass in einem Data Mart alle benötigten Daten gespeichert sind, die für eine Anfrage benötigt werden. Durch die Verwendung von Data Marts ist es möglich die Abarbeitung von Abfragen zu beschleunigen [AM97, S69ff]. Bauer und Günzel [BG04, S. 60] zeigen weitere Vorteile auf:

- Eigenständigkeit (z.B. Mobilität).
- Datenschutzaspekte durch Teilsicht auf die Daten.
- Organisatorische Aspekte (z.B. Unabhängigkeit von Abteilungen).
- Verringerung des Datenvolumens.
- Performanzgewinn durch Aggregationen.
- Verteilung der Last.
- Unabhängigkeit von den Aktualisierungszyklen des Data Warehouse.

Die zu Grunde liegende Motivation steckt in der Verringerung des Datenvolumens und der Komplexität des Datenmodells. Als Modellierungsbasis bieten sich zwei verschiedene Modelle an, zum einen abhängige Data Marts und zum anderen unabhängige Data Marts.

Abhängige Data Marts stellen Extrakte eines Data Warehouse dar, welche auch aggregiert sein können. Die Daten werden aus dem zu Grunde liegenden Data Warehouse entnommen und stimmen inhaltlich sowie strukturell zu jedem Zeitpunkt überein. Dies erleichtert die Bildung von Data Marts, da einfache vom System zur Verfügung gestellte Mechanismen zur Generierung herangezogen werden können [BG04]. Abbildung 2 erläutert den Aufbau von abhängigen Data Marts graphisch:

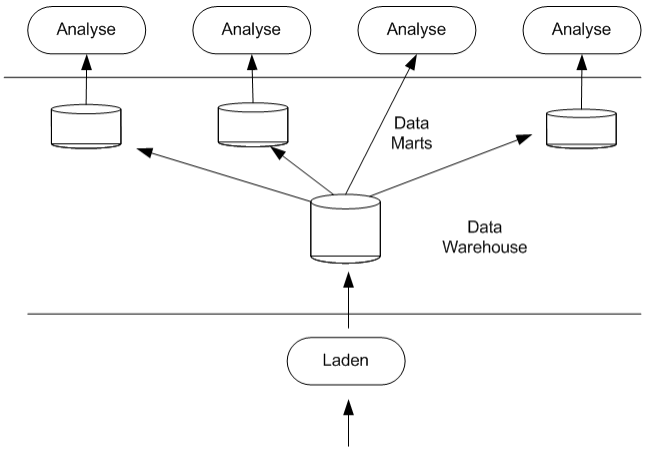

Abbildung 2: Architektur bei Verwendung abhängiger Data Marts [BG04, S. 61]

Unabhängige Data Marts entstehen, wenn keine Basisdatenbank zur Verfügung steht. Die Daten liegen meist in kleinen von einander unabhängigen Data Warehouses. Diese sind oft einzelnen Abteilungen zugewiesen. Dies beinhaltet zwar den Vorteil, dass die Komplexität zu Anfang niedrig gehalten werden kann und nach kurzen Anlaufzeiten für die einzelnen Abteilungen nutzbare Ergebnisse produziert

werden können. Werden aber Analysen gewünscht, die über einen solchen Bereich hinausgehen, so erweist sich diese Architektur als problematisch. Durch unterschiedliche Transformationsregeln kann eine übergreifenende Analyse oft nicht konsistent berechnet werden [BG04, S. 62f]. Die Data Marts liegen in dieser Form der Architektur als bereits vorverdichtete und bereinigte Datenquellen vor dem endgültigen Data Warehouse (siehe Abbildung 3).

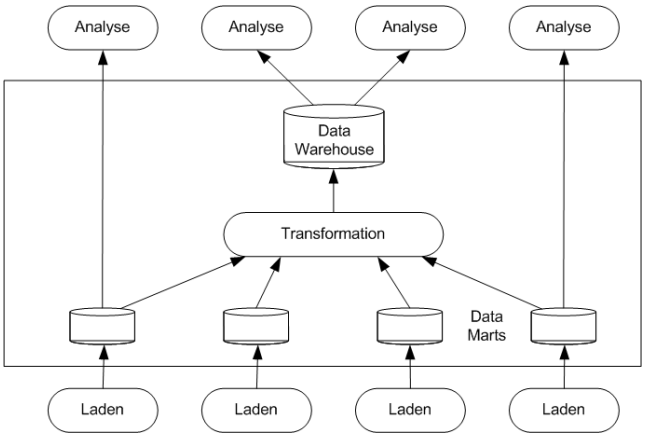

Abbildung 3: Architektur bei Verwendung unabhängiger Data Marts [BG04, S. 63]

In einem weiteren Punkt zeigen sich Unterschiede zwischen einem transaktionalen datenbankgestützten Anwendungssystem („Online Transactional Processing“ - OLTP) zur operativen Datenhaltung und einer „Online Analytical Processing“ (OLAP) [CCS93] Anwendung wie z.B. einem Data Warehouse. Eine OLAP Anwendung stellt ein exploratives interaktives System dar, das eine Datenanalyse auf Grund eines konzeptuellen multidimensionalem Datenmodells [KRRT98] vornimmt. [BG04, S. 8ff], [Leh03, S. 16ff]. Tabelle 1 fasst diese Unterschiede zusammen.

		Transaktional ausgerichtete Anwendungssysteme	Analytisch orientierte Data Warehouse Systeme
Perspektive der Anwendung	Anwendertyp und -anzahl	Ein- / Ausgabe durch Sachbearbeiter sehr viele Anwender	Auswertungen durch Manager, Controller, Analysten
	Interaktions-dauer und -typ	Lesen, Einfügen Aktualisieren, Löschen (kurze Lese- / Schreibtransaktionen)	Lesen periodisches Hinzufügen neue Datenbestände
	Anfragestruktur	einfach strukturiert	komplex, jedoch überwiegend bestimmten Mustern folgend
	Bereich einer Anfrage	wenige Datensätze (überwiegend Einzeltupelzugriff)	viele Datensätze (überwiegend Bereichsanfragen)
	Anzahl gleichzeitiger Zugriffe	sehr viele (bis in die Tausende)	wenige (bis in die Hunderte)
	Anwenderzahl	sehr viele	wenige, bis einige Hundert
Perspektive der Datenhaltung	Datenquellen	zentraler Datenbestand	mehrere unabhängige Datenquellen
	Schemaentwurf	anfrageneutrale Datenmodellierung	analysebezogene Datenmodellierung
	Eigenschaften des Datenbestandes	originär, zeitaktuell autonom, dynamisch	abgeleitet / konsolidiert historisiert, integriert, stabil teilweise (vor-)aggregiert
	Datenvolumen	Megabyte - Gigabyte	Gigybyte - Terabyte
	Typische Antwortzeit	ms - s	s - min

Tabelle 1: Vergleich von transaktional zu analyseorientierten Anwendungssysteme in Anlehnung an [Leh03, S. 18]

2.2 Extraktion, Transformation, Laden (ETL)

Der ETL-Prozess stellt einen wichtigen Teilschritt für die Datenmigration in ein Data Warehouse dar. Aufgabe des ETL-Prozesses ist die Bereitstellung der Daten in der geforderten Struktur, Aktualität und Qualität. Die dafür notwendigen Funktionen findet man häufig in ETL-Werkzeugen zusammengefasst [Wie99]. Der ETL-Prozess zeichnet sich durch folgende nach Kurz [Kur99] grob dargestellten Arbeitsschritte aus:

- Analyse und Dokumentation der Quelldatenbanksysteme.
- Extrahieren der ausgewählten Objekte.
- Transformieren der ausgewählten Objekte.
- Validieren und Bereinigen der transformierten Objekte.
- Vorbereiten der DWH-Routinen
- Laden der bereinigten und transformierten Objekte in das DWH.

Kurz [Kur99] unterteilt den ETL-Prozess in zwei Phasen: einerseits in die Definitionsphase und andererseits in die Ausführungsphase. Die Definitionsphase ist für die Definition bzw. Festlegung aller Objekte verantwortlich. Außerdem wird der Ablauf des ETL-Prozesses festgelegt. In einem ersten Schritt werden die operativen Quelldatensysteme analysiert. Danach werden die notwendigen Transformationen festgelegt. Abschließend wird die ETL-Laderoutine erstellt. Diese liest die entsprechenden Objekte ein, verarbeitet sie auf Grund der Transformationsregeln und legt die Objekte in der „Staging Area“ ab. Abbildung 4 zeigt dies in übersichtlicher Form.

Abbildung 5 stellt die technischen Prozessabläufe des ETL-Prozesses = „Ausführungsphase“ dar. Diese werden auf Grund der in der Definitionsphase festgelegten Metadatenbeschreibung, ETL-Job und -Laderoutinen sowie den erstellten Transformationsregeln und Übernahmeregeln durchgeführt.

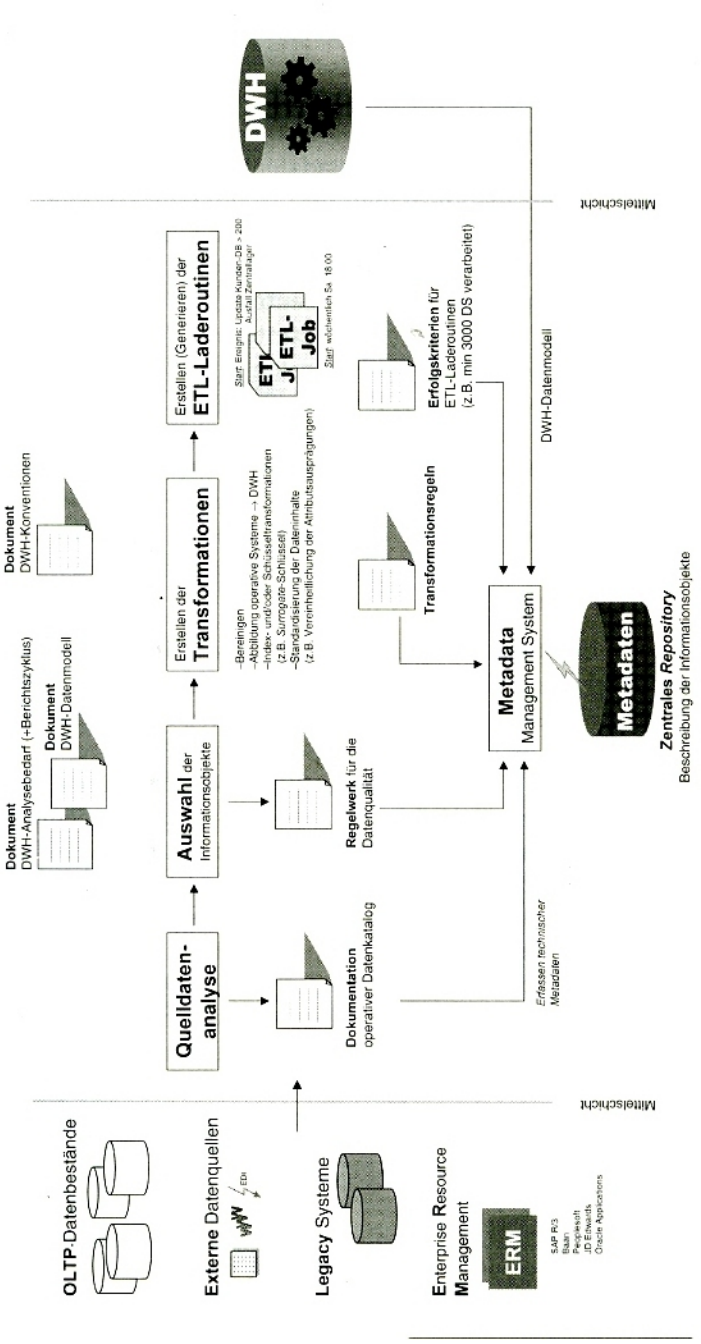

Abbildung 4: Definitionsphase des ETL-Prozesses [Kur99, S. 269]

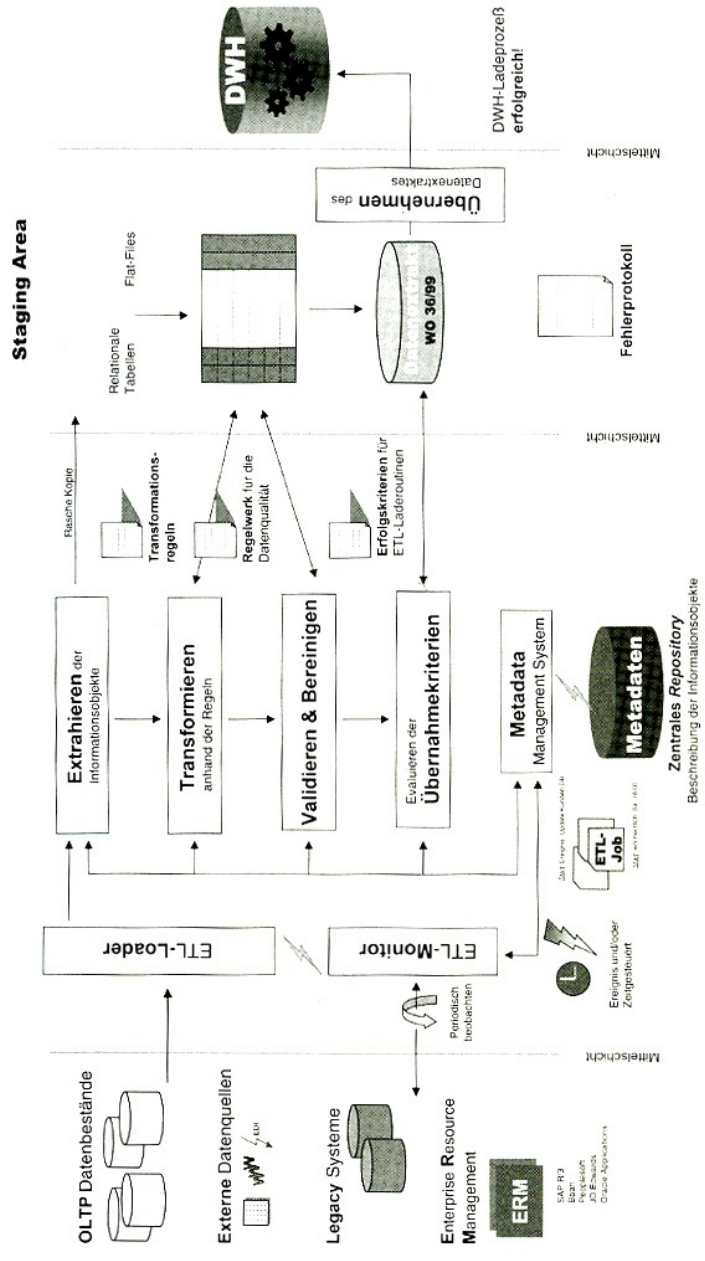

Abbildung 5: Ausführungsphase des ETL-Prozesses [Kur99, S. 271]

2.2.1 Extraktion

Die Extraktion von Daten beschreibt das Auslesen der selektierten bzw. der geänderten Daten aus den Quellsystemen. Da es sich hierbei um unterschiedlich große Datenmengen handelt, kann eine Komprimierung der zu transferierenden Daten sinnvoll sein [Kur99]. Wieken [Wie99, S. 190f] stellt drei verschiedene Zugriffsmöglichkeiten dar.

- Periodisch kompletter Abzug der Daten.
- Periodischer Abzug der geänderten Daten (Übertragung der Delta-Daten).
- Protokollierung aller Änderungen (Log).

Bei kompletten Abzügen werden die relevanten Objekte gesamt aus den zu Grunde liegenden operativen Datenbanksystemen entladen. Dies bedeutet gleichzeitig den einfachsten Lösungsweg und durch das zu erwartende hohe Datenaufkommen inkl. den Ladezeiten auch den aufwendigsten Lösungsweg. Durch eine Beschränkung des Abzuges auf die Delta-Daten[1] kann eine meist erhebliche Reduzierung des Datenvolumens erreicht werden. Wird ein Wert zwischen zwei Abzügen mehrmals geändert, so wird immer nur die letzte Änderung in das Data Warehouse übernommen. Beim Log-Verfahren werden alle Änderungen mitprotolliert und anschließend in das Data Warehouse übernommen [Wie99].

Bauer und Günzel stellen nach Kimball et al. [KRRT98] verschiedene Zeitpunkte vor, an denen der Extraktionsvorgang angestoßen werden kann [BG04]:

- Periodisch - Die Extraktion wird periodisch durchgeführt. Hierbei muss jedoch die Volatilität der Daten berücksichtigt werden. Wetterdaten z.B. müssen ständig aktualisiert werden. Produktdaten ändern sich hingegen kaum.
- Anfragegesteuert - Es wird eine explizite Anfrage abgesetzt, um den Extraktionsvorgang anzustoßen.
- Ereignisgesteuert - Der Extraktionsvorgang wird durch ein Zeit-, Datenbank- oder externes Ereignis angestoßen. Beispielsweise kann der Extraktionsvorgang nach einer festgelegte Anzahl von Änderungen begonnen werden (Datenbank). Beide zuvor aufgezeigten Zeitpunkte können per Definition auch in diese Gruppe eingeordnet werden.
- Sofort - Ist eine besondere Aktualität der Daten erforderlich, so kann bei jeder Änderung ein Extraktionsvorgang vorgenommen werden. Dies ist z.B. bei Börsenkursen der Fall. Das Data Warehouse muss genauso aktuell wie die operative Datenbank sein.

[1] Delta-Daten stellen den Unterschied der Daten vom letzten Extrahierungszeitpunkt zum aktuellen Zeitpunkt dar. Dieser Unterschied muss allerdings vor der eigentlichen Extrahierung berechnet werden.

2.2.2 Transformation

Mit Hilfe der Datentransformation werden ein einzelnes oder mehrere Felder der operativen Datenhaltung in ein einzelnes oder mehrere Felder des Data Warehouse transformiert. Dies bedeutet, dass sowohl Daten und Schemata als auch die Datenqualität an die Anforderungsanwendungen angepasst werden [BG04],[Wie99].

Im einfachsten Fall können Werte direkt übernommen werden. Wenn dies nicht möglich ist, müssen entsprechende Transformationen vorgenommen werden. Diese Transformationen behandeln Schemakonflikte (siehe Kapitel 4.1) sowie inhaltliche Aspekte wie Datenintegration und Datenbereinigung.

Da Daten oft aus heterogenen Datenquellen stammen, müssen diese zuerst auf ein einheitliches Format gebracht werden. Dazu sind nach Kimball et al. [KRRT98] meist folgende Transformationen für die sogenannte Datenmigration notwendig:

- Anpassung von Datentypen (z.B. Speicherung von Datumsangaben im Format Date).
- Konvertierung von Kodierungen (z.B. unterschiedliche Kodierung des Geschlechts wie 1 (weiblich), 2 (männlich) hin zu f (female) und m (male)).
- Vereinheitlichung von Zeichenketten (z.B. Darstellung des Geschlechts immer in Kleinbuchstaben).
- Vereinheitlichung von Datumsangaben (z.B. numerische Darstellung des Datums in der Form von YYYYMMDD).
- Umrechnung von Maßeinheiten (z.B unterschiedliche Darstellung von Volumina wie Hektoliter hin zu Liter).
- Kombinierung und Separierung von Attributen (z.B. Aufspaltung von „Vorname Zuname“ in „Vorname“ und „Zuname“).

Abbildung 6 stellt den Ablauf der Transformationsfunktion vereinfacht dar. Die „operativen Strukturen“ stellen hierbei die operativen Daten dar, nachdem sie aus dem operativen System entnommen wurden. Bei der virtuellen Tabelle kann es sich je nach ETL-Werkzeug auch um eine echte Datei handeln, an welcher die Transformationsregeln angewendet werden [Wie99].

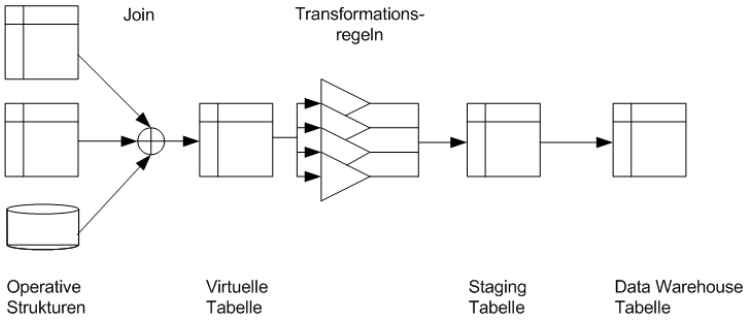

Abbildung 6: Grundstruktur der Transformationsstruktur [Wie99, S. 196]

2.2.3 Laden

Mit Hilfe des Ladevorgangs werden die transformierten Daten in das Data Warehouse eingelesen. Hierbei wird davon ausgegangen, dass die Struktur der Datensätze der Struktur der Tabelle im Data Warehouse entspricht und die Daten somit geladen werden können [Wie99].

Der Ladevorgang hat auf alle beteiligten Systeme Auswirkungen; einerseits durch die großen Datenmengen, die bewegt werden müssen, andererseits werden während dieser Phase andere Systeme (z.B. durch die hohe Auslastung) teilweise eingeschränkt bzw. für den Gebrauch kurzfristig gänzlich gesperrt [BG04]. Deshalb ist es vorteilhaft Ladevorgänge zu Zeitpunkten durchzuführen, an denen das System durch Anfragen der Benutzer nicht ausgelastet ist.

Beim Ladevorgang ist zu unterscheiden, ob das Data Warehouse gänzlich mit allen Daten aufgefüllt werden muss, oder ob es sich um Aktualisierungen handelt, bei denen nur Änderungen geladen werden müssen (siehe Kapitel 2.2.1 - je nach ursprünglich gewählter Strategie) Es werden hierbei nicht nur die aktuellen Daten gespeichert, sondern es erfolgt eine Historisierung der Daten. Geänderte Datensätze werden nicht einfach überschrieben, sondern zusätzlich in der Datenbank abgelegt [BG04, S. 95], [Inm02, S. 34f].

Es ist möglich, dass während des Ladeprozesses Fehler auftreten. Für diesen Fall muss das System in der Lage sein, darauf zu reagieren. Es kann z.B. den Fehler melden und / oder eine Maßnahme vom Anwender verlangen. Eine weitere Möglichkeit besteht darin anzugeben, welches das führende System ist bzw. wie Default-Werte erkannt werden können [AM97, S. 40], [Wie99, S. 197].

3 Grundlagen der Datenanalyse und -bereinigung

Dieses Kapitel definiert den Begriff der Datenbereinigung als die Erhöhung der Datenqualität. Dazu wird in einem ersten Schritt der Begriff „Datenqualität“ definiert. In einem nächsten Schritt wird dargestellt, welche Arten von unsauberen Daten („Dirty Data“) auftreten können. Weiters wird aufgezeigt, in welche Dimensionen und Regeln sich Datenqualität aufschlüsseln lässt. In einem letzten Schritt wird in Anlehnung an Müller und Freytag [HM03] gezeigt, wie ein möglicher Datenbereinigungsprozess („Data Cleaning Process“) aufgebaut wird.

3.1 Datenqualität (Data Quality)

Orr [Orr98] definiert Datenqualität als Übereinstimmung der abgebildeten Daten in einem „Information System“ (IS) mit den Daten aus der Wirklichkeit. Dies bedeutet: bei einer 100%igen Datenqualität stimmen die abgebildeten Daten in einem IS mit den aktuellen Daten der Wirklichkeit völlig überein und dies stellt eine exakte Wiedergabe der Realität dar. Bei einer Datenqualität von 0% ist keine Übereinstimmung zu sehen. Dies bedeutet, dass die anstehenden Entscheidungen auf Basis von falschen Daten getroffen werden, wodurch negative Konsequenzen entstehen können.

Tayi und Ballou [TB98, S. 54] beschreiben die Datenqualität mit einer Redewendung: „fitness for use“. Die Autoren wollen hiermit zum Ausdruck bringen, dass das Konzept rund um Datenqualität relativ ist. Es kann nicht davon ausgegangen werden, dass die Daten, sobald sie für eine Person verwendbar sind, auch von Dritten verarbeitet werden können, denn hier ist ein Handlungsspielraum für Fehlinterpretationen gegeben.

Bei genauerer Betrachtung der Inhalte von Datenbanken oder Data Warehouses (DW) kann festgestellt werden, dass diese mit Inkonsistenzen und Fehlern behaftet sind. Diese fehlerbehafteten Daten werden oftmals auch als „Dirty Data“ bezeichnet. Nach Kim et al. [KCH+03, S. 83] tritt „Dirty Data“ in drei Ausprägungen auf:

1. Fehlende Daten
2. Nicht fehlende aber falsche Daten
3. Nicht fehlende, nicht falsche aber unbrauchbare Daten

Data Cleaning, das auch synonym mit den Begriffen „data cleansing“, „data scrubbing“ oder „Datenbereinigung“ bezeichnet wird, wird dazu eingesetzt, diese Inkonsistenzen und Fehler in Datenbeständen zu erkennen. In einem weiteren Schritt

werden die unsauberen Daten anschließend entfernt bzw. korrigiert [Ger05, S. 1]. Um die zuvor genannten Fehler zu vermeiden, werden die Daten schon vorverarbeitet. Dazu eignen sich besonders folgende Methoden [HK00, S. 105ff]:

- Datenbereinigung (Data cleaning)
- Datenintegration (Data integration)
- Datentransformation (Data transformation)
- Datenverdichtung (Data reduction)

Datenintegration bedeutet das Zusammenführen von verschiedenen Datenquellen in eine einzige zusammenhängende Datenquelle wie z.B. Data Warehouses oder „Data Marts“. Unter Datentransformation wird die Normalisation der Daten verstanden, welche die Genauigkeit und Effizienz der Datenhaltung erhöht. Mit Hilfe der Datenverdichtung werden die Daten kompakter dargestellt und gespeichert. Dabei müssen jedoch dieselben analytischen Ergebnisse produziert werden, sodass kein Informationsgehalt verloren geht. Die oben erwähnten Methoden der Datenbereinigung erkennen Fehler und Unstimmigkeiten in der Datenhaltung und korrigieren diese. Abbildung 7 zeigt die Anwendung dieser Methoden in anschaulicher Form.

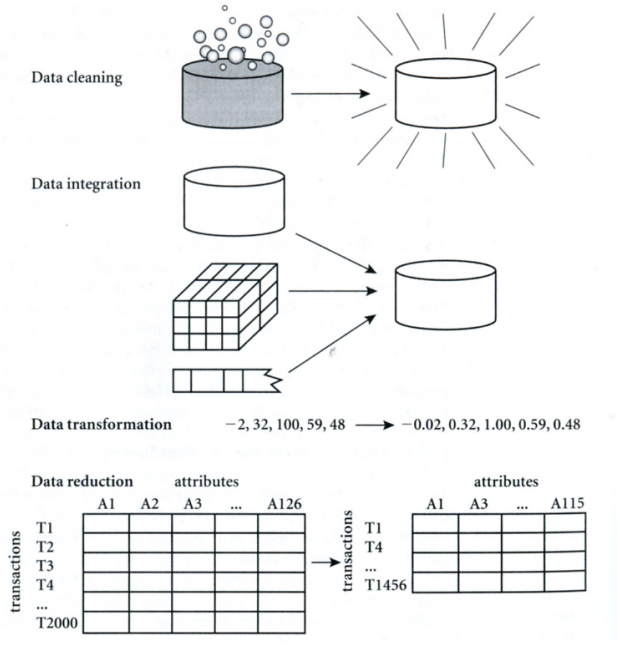

Abbildung 7: Formen der Datenvorverarbeitung [HK00, S. 108]

Typischerweise werden die Datenintegration und Datenbereinigung im Vorfeld von Datenbankzusammenschlüssen durchgeführt. Data cleaning findet sich aber auch

als Methode zur Kontrolle nach erfolgten Datenbankintegrationen. Hier wird das Hauptaugenmerk auf die Fehler gelegt, die auf Grund dieser Zusammenschlüsse entstehen können [HK00, S. 107].

3.2 Dimensionen der Datenqualität

Um Datenqualität zu beschreiben, ist es wichtig dabei nicht nur an Richtigkeit (accuracy) der Daten zu denken, sondern es müssen mehr Dimensionen bedacht werden [SMB05]. In der Literatur ([HM03], [LGJ03], [SMB05], [TB98]) finden sich vor allem drei weitere wichtige Dimensionen:

- Richtigkeit (accuracy)
- Vollständigkeit (completeness)
- Konsistenz (consistency)
- Aktualität (currency/timeliness)

In den folgenden Abschnitten werden diese Dimensionen näher erläutert.

3.2.1 Richtigkeit (accuracy)

Die Richtigkeit gibt an, ob ein Wert korrekt in einer Datenbank eingetragen worden ist. Hierbei wird zwischen syntaktischer und semantischer Richtigkeit unterschieden. Unter syntaktischer Richtigkeit versteht man Werte, die dem vorgesehenen Datentyp (Wertebereich) entsprechen. Es wird eine formale Übereinstimmung der verwendeten Objekte und Notationen mit den definierten Vorgaben gefordert. Semantische Richtigkeit bedeuted, dass die Werte der Wirklichkeit korrekt abgebildet werden und somit der reale Sachverhalt stimmig widergespiegelt wird.

Syntaktische Richtigkeit kann mit Hilfe von Vergleichsmethoden überprüft werden. Es wird z.B. die Distanz zweier Werte v und v' berechnet und verglichen. Elfeky et al. [EEV02, S. 21ff] schlägt dazu verschiedene Methoden wie z.B. die Hamming-Distanz, die Edit-Distanz oder N-grams vor. Hierbei ist anzumerken, dass die syntaktische Richtigkeit leichter als die semantische festgestellt werden kann [SMB05, S. 7]. In Fällen von syntaktischer Richtigkeit handelt es sich meistens um Fehler in der Schreibweise z.B. „Mayr“ anstatt „Mayer“.

Bei einem semantischen Fehler wurde eine falsche Information gespeichert: z.B. ein falscher Autor zu einem Buchtitel. Um einen besseren Überblick über die Richtigkeit der Daten zu gewinnen, ist die Berechnung des Verhältnisses der korrekt eingetragenen Werte zu allen Werten einer Datenbank sinnvoll. Dadurch ergibt sich ein Prozentsatz, der Aufschluss über die korrekt eingetragenen Werte liefert.

3.2.2 Vollständigkeit (completeness)

Die Vollständigkeit gibt an, ob alle relevanten Daten gesammelt und in der Datenbank eingetragen sind. Eine Berechnung kann durch das Verhältnis zwischen den Elementen einer Miniwelt, die in einer Datenbank repräsentiert werden, zu der Gesamtanzahl von Elementen in einer Miniwelt durchgeführt werden. Dies ist vor allem in Datenbanken interessant, in denen Null-Werte erlaubt sind. Hier muss die Frage beantwortet werden, warum ein Wert fehlt. Dies kann drei Gründe haben:

- Fall 1: Der Wert existiert nicht;
- Fall 2: Der Wert existiert, ist aber unbekannt;
- Fall 3: Es ist unbekannt, ob der Wert existiert;

Nachstehende Tabelle 2 veranschaulicht das Vorkommen von Null-Werten graphisch.

ID	Vorname	Nachname	E-mail
1	Alex	Mayer	alex.mayer(at)students.jku.at
2	Franz	Maier	NULL [nicht vorhanden] (Fall 1)
3	Fritz	Müller	NULL [vorhanden, nicht bekannt] (Fall 2)
4	Karl	Schweitzer	NULL [unbekannt] (Fall 3)

Tabelle 2: Beispiele zur Interpretation von Null-Werten

Wenn nun angenommen wird, dass ID 2 keine E-mail-Adresse besitzt, besteht keine Unvollständigkeit. Falls Person 3 eine E-mail-Adresse besitzt, aber keine bekannt ist, so ist der Datensatz unvollständig. Im dritten Fall wird die Möglichkeit dargestellt, dass nicht darüber entschieden werden kann, ob der Datensatz unvollständig ist, da nicht bekannt ist, ob eine E-mail-Adresse existiert [SMB05, S. 8].

Die Vollständigkeit ist kein primäres Problem der Datenbereinigung sondern vielmehr der Datenintegration. Dennoch trägt die Datenbereinigung einen großen Teil zur Vollständigkeit bei, indem fehlerhafte Tupel korrigiert (siehe Kapitel 6) und nicht einfach nur aus dem System gelöscht werden [HM03, S. 9].

3.2.3 Konsistenz (consistency)

Die Konsistenz bestimmt in Datenbanken die Widerspruchsfreiheit. Es werden hierbei Integritätsbedingungen festgelegt, die garantieren, dass keine semantischen Regeln verletzt werden und somit Widerspruchsfreiheit vorliegt. Integritätsbedingungen müssen von allen Instanzebenen des Datenbankschemas erfüllt werden. Es wird zwischen zwei verschiedenen Kategorien von Integritätsbedingungen unterschieden, zum einen die inter-relationalen und zum anderen die intra-relationalen Bedingungen.

Als Beispiel für eine intra-relationale Bedingung sei die einer Veröffentlichung eines Kinofilms auf DVD angeführt. Das Jahr der Veröffentlichung muss mindestens dasselbe Jahr bzw. ein späteres sein, als das Jahr der Erstveröffentlichung. Bei der inter-relationalen Bedingung muss z.B. das Veröffentlichungsdatum eines Films dem Datum eines eventuell gewonnen Oskars entsprechen [SMB05, S. 10].

Da die Widerspruchsfreiheit bzw. die Integritätsbedingungen sehr gut erforscht sind, sind diese in vielen Datenbanksystemen bereits durch Formulierungen in DB-spezifischer Syntax beschrieben und dadurch sehr leicht überprüfbar. In den meisten Datenbereinigungstools werden ebenso Regeln bestimmt, die eine Widerspruchsfreiheit herstellen sollen. Dies kann problemlos durch automatische Kontrolle erfolgen [SMB05, S. 10].

3.2.4 Aktualität (currency / timeliness)

Ein wichtiger Punkt in Bezug auf Datenqualität ist die Aktualität. Auf der einen Seite gibt es Dateninhalte, bei denen keine Veränderungen auftreten wie z.B. Vorname und Geburtsdatum von Personen. Diese Attribute von Entitäten werden als „stabil“ betrachtet. Auf der anderen Seite jedoch gibt es eine Vielzahl von Attribut-Werten, die einer ständigen Aktualisierung bedürfen („volatil“). Als Beispiele finden sich hier vor allem die Adresse und die Telefonnummer [SMB05, S. 9].

Die Aktualität bestimmt, wie oft gespeicherte Daten überprüft und gegebenenfalls geändert werden müssen. Sie kann vor allem durch den Zeitpunkt der letzten Aktualisierung gemessen werden. Hierbei ist festzuhalten, dass sich Daten in verschiedenen Zeiträumen unterschiedlich ändern können. Bei gleichbleibender Veränderungsrate kann die Aktualität leicht bestimmt werden. Findet die Veränderung unregelmäßig statt, kann die Berechnung des Durchschnittswertes der Aktualisierungszeiträume hilfreich sein. Um die Aktualität zu bestimmen, müssen die beiden Werte (Durchschnittszeitraum der Aktualisierungen und Zeitraum von der letzten Aktualisierung) verglichen werden. Liegt der Zeitraum seit der letzten Aktualisierung über der durchschnittlichen Aktualisierungsrate, darf angenommen werden, dass wahrscheinlich neuere Daten zur Verfügung stehen [SMB05, S. 9].

Es kann aber auch der Fall eintreten, dass die vorhandenen Daten zwar aktuell aber dennoch für einen speziellen Fall wertlos sind. Als Beispiel sei der Zeitplan einer Lehrveranstaltung (LVA) angeführt. Wenn die eingetragenen Stunden festgelegt sind, aber sie erst nach Lehrveranstaltungsbeginn für den Studenten einsehbar sind, so sind sie zwar aktuell, aber nicht rechtzeitig vorhanden. Weiters ist denkbar, dass Inhalte einer Lehrveranstaltung zwar bekannt sind, aber die Bekanntgabe erst nach Beginn der LVA erfolgt, so ist dies wohl zu spät. In diesem geschilderten Fall ist es wichtig, dass die Daten nicht nur aktuell, sondern auch rechtzeitig für einen interessierten Studenten abrufbar sind.

Weiters ist anzumerken, dass Aktualität und Richtigkeit oftmals ähnlich wahrgenommen werden. Nicht aktuelle Dateninhalte werden in der Praxis oft als unvollständig bezeichnet. Nichtsdestotrotz müssen diese beiden Dimensionen aber stets getrennt betrachtet werden, da sie unterschiedliche Methoden zur Behebung der Fehlerquellen benötigen [SMB05, S. 9f].

3.2.5 Hierarchiebildung von Qualitätsdimensionen

Um eine Vereinfachung der Dimensionen vorzunehmen, zeigen Müller und Freytag [HM03, S. 8] in welcher Art und Weise die Dimensionen hierarchisch aufgebaut werden können. Tabelle 3 veranschaulicht dies. Ohne Kenntnis der Dimensionen ist eine effiziente Fehlerbearbeitung nur erschwert möglich [TB98, S. 56].

Richtigkeit		
	Integrität	
		Vollständigkeit
		Validität (Gültigkeit)
	Konsistenz	
		Schemakonformität
		Einheitlichkeit
	Dichte	
Eindeutigkeit		

Tabelle 3: Hierarchie der Datenqualitätskriterien in Anlehnung an [HM03, S. 8]

Strong et al. [SLW97] nehmen eine andere Klassifizierung der Dimensionen vor. In einer Studie wurden über 100 Punkte identifiziert, die zu 20 Dimensionen zusammengefasst wurden. Diese wurden anschließend in vier Kategorien aufgeteilt. Tabelle 4 stellt eine Übersicht über die festgelegten Kategorien und die zugeteilten Dimensionen dar.

Kategorie	Dimension
Intrinsisch	Richtigkeit, Objektivität, Glaubwürdigkeit, Vertrauenswürdigkeit
Zugänglichkeit	Zugang, Sicherheit
Kontextabhängig	Relevanz, Mehrwert, Rechtzeitigkeit, Vollständigkeit, Datenmenge
Repräsentation	Interpretierbarkeit, Verständlichkeit, einheitliche und prägnante Darstellung

Tabelle 4: Datenqualitätsdimensionen in Anlehnung an [SLW97, S. 104]

Scannapieco et al. [SMB05] zeigt in einer Zusammenfassung (siehe Tabelle 5), dass die oben erwähnten Dimensionen nur als ein Auszug aus einer großen Menge an

Dimensionen zu betrachten sind. Da in der Literatur eine Vielzahl von Einteilungen in diverse Hierarchien und Dimensionen vorliegt, wird mit den ausgewählten Dimensionen ein möglichst guter Querschnitt dargestellt und dieser näher beschrieben.

	WandWang 1996 [WW96]	WangStrong 1996 [WS96]	Redman 1996 [Red96]	Jarke 1999 [JLVV99]	Bovee 2001 [BSM03]
Accuracy	X	X	X	X	X
Completeness	X	X	X	X	X
Consistency / Representational Consistency	X	X	X	X	X
Time-related Dimensions	X	X	X	X	X
Interpretability		X	X	X	X
Ease of Understanding / Understandability		X			
Reliability	X			X	
Credibility				X	X
Believability		X			
Reputation		X			
Objectivity		X			
Relevancy / Relevance		X	X		X
Accessibility		X		X	X
Security / Access Security		X		X	
Value-added		X			
Concise representation		X			
Appropriate amount of data / amount of		X	X		
Availability				X	
Portability			X	X	
Responsiveness / Response Time				X	

Tabelle 5: Qualitätsdimensionen in ausgewählten Veröffentlichungen [SMB05, S. 12]

3.3 Datenbereinigungsprozess (Data cleaning process)

Der Prozess, der die unsauberen Daten in saubere und qualitative Daten umwandelt, ist komplex und wird deshalb in mehrere Einzelschritte unterteilt. Die in

Abbildung 8 graphisch zusammengefassten Phasen werden von unterschiedlichen Autoren ([RD00], [HM03], [MM00], [Ger05]) verwendet.

In der Phase des „Data auditing“ wird nach möglichen vorhandenen Fehlern gesucht. Im nächsten Schritt, der „Workflow specification“, werden geeignete Methoden gesucht, die die Fehler automatisch aufspüren und eliminieren können. Anschließend werden in der „Workflow execution“ die ausgesuchten Methoden an einem Datenauszug getestet, sodass nach erfolgreicher Testphase die Methoden schließlich auf die ganze Datenmenge angewendet werden. In der abschließenden Phase - der Kontrollphase („Post-processig / Control“) - werden die Daten von einem Experten überprüft, sodass insbesondere nicht erkannte Fehler durch den Domänenexperten mittels seines speziellen Wissens noch zusätzlich behoben werden.

Dieser Prozess kann nicht als einmalig angesehen werden. Durch mehrere Iterationen wird ein zufriedenstellenderes Ergebnis erreicht. Hierbei kann mit jedem Schleifendurchlauf eine „bessere“ Qualität der Daten erreicht werden, wobei nach mehreren Durchläufen sich die Qualität nicht mehr signifikant verbessern lässt. Müller und Freytag [HM03, S. 11] merken an, dass im Zuge von Bereinigungen neue Fehler auftreten können. Dadurch ist es nötig den Prozess mehrmals durchlaufen zu lassen.

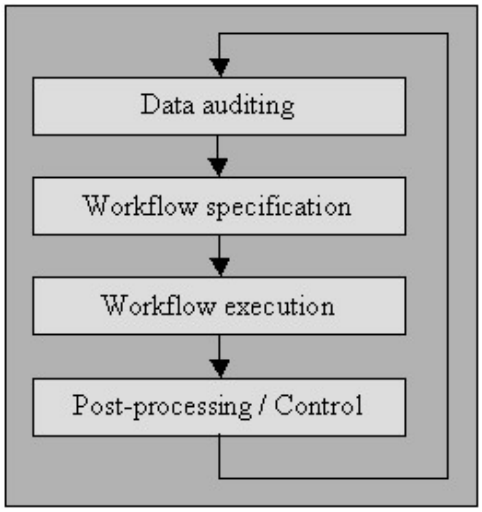

Abbildung 8: Data Cleaning process. [HM03, S. 11]

In der Folge werden die typischen Phasen des Data Cleaning Prozesses beschrieben.

3.3.1 Datenprüfung (Data auditing)

Aufgabe des „Data auditing“ ist das Analysieren der vorhandenen Daten. Dieser Prozess ist dafür verantwortlich, dass soviel Information wie möglich über die

Daten gesammelt wird, sodass mögliche Fehlerquellen leichter identifiziert werden können. Es ist hierbei nötig auch die Metadaten zu erheben. Metadaten sind mit Daten über Daten gleichzusetzen. Auf diese Art und Weise wird mehr Information über die einzelnen Attribute gesammelt. Dadurch ist es möglich, Attribute im vorliegenden Schema unabhängig von ihrer Bezeichnungen besser miteinander in Verbindung zu bringen und eventuelle Abhängigkeiten festzustellen. [Ger05, S. 4].

Grundsätzlich wird zwischen zwei Ansätzen unterschieden: zum einen „Data profiling“ und zum anderen „Data mining“. Mit Hilfe des „Data profiling“ werden die Attribute auf Instanzebene analysiert. Dabei werden möglichst viele Metadaten erfasst, wie z.B. der Datentyp, Wertebereiche oder auch statistische Werte wie Varianz, Minimum- oder Maximumwerte. „Data mining“-Verfahren schaffen einen globalen Zusammenhang. Durch diesen Gesamteindruck ist es möglich Korrelationen mehrerer Attribute zueinander leichter aufzuzeigen. Diese Technik erleichtert die Erstellung von Regeln bzw. Integritätsbedingungen, damit fehlende Werte ergänzt und falsche Werte korrigiert werden können [HM03, Ger05].

3.3.2 Ablaufspezifikation (Workflow specification)

Im zweiten Schritt des „data cleaning process“ muss entschieden werden, welche Transformationen und Reinigungsmethoden angewendet werden (z.B. die Transformation für Fuzzygruppierung oder die Transformation für Fuzzysuche (siehe Kapitel 6.2.2)). Die Auswahl der Methoden hängt vor allem von der Anzahl der Datenquellen und vom Grad der Heterogenität bzw. der „Verschmutzung“ der Daten ab [RD00, S. 5]. Beispielsweise kann es sinnvoll sein mit Hilfe einer Transformationsmethode Datenwerte eines Attributwertes zu extrahieren, um den Inhalt auf mehrere Felder aufzuteilen, wie z.B. bei der Adresse in Postleitzahl und Ort (siehe Kapitel 6.2.2.3). Weiters ist die Standardisierung von Werten eine Möglichkeit Daten zu bereinigen, so z.B. einheitliche Formatierungen für die Uhrzeit, das Datum oder Maßeinheiten.

Die ausgewählten Methoden bilden in der Sequenz, in der sie ausgeführt werden, die Spezifikation bzw. den „Data cleaning workflow“. Die Methodenauswahl erfolgt hierbei so, dass viele Problemsituationen damit abgedeckt werden. Dennoch ist eine zu spezifische Auswahl ungünstig, da dadurch manche Problemstellungen nicht erkannt werden würden, z.B. Abhängigkeiten von einzelnen Spaltenwerten.

Es ist zu beachten, dass manche Fehler erst im Zuge der Fehlerkorrektur auftreten können, d.h. die Behebung eines Fehlers forciert erst die Entstehung eines anderen [Ger05, S. 4]. Beispielsweise entsteht dies, wenn ein Wert falsch gesetzt wurde und abhängige Werte nicht dynamisch an diesen gebunden sind. So muss jede einzelne Berechnung überprüft werden.

Bei der Bereinigung von Fehlern ist die Betrachtung der Gründe für mögliche Fehler ein wichtiger Schritt, da dadurch manche Probleme leichter korrigiert bzw.

im Vorhinein vermieden werden können. Wenn z.B. bei der Dateneingabe häufig dieselben Tippfehler vorherrschen, kann hierbei das Layout der Tastatur (Englisch oder Deutsch) Aufschluss über manche Fehler geben und so eine leichtere Fehlerkorrektur ermöglichen [HM03, S. 12]. Rahm und Do [RD00] schlagen eine Vorabtestung der Methoden an ausgewählten Datensätzen vor, um dadurch noch Verbesserungen an den Transformationsschritten vorzunehmen, da wie zuvor aufgezeigt Fehler erst nach mehrmaliger Anwendung auftreten können.

3.3.3 Ablaufdurchführung (Workflow execution)

Nach Festlegung und erfolgreichem Testlauf der Workflow Spezifikation kann diese ausgeführt werden [RD00, S. 5]. Ziel des Workflows ist für jeden untersuchten Attributwert zu entscheiden, ob er korrekt oder fehlerhaft ist und gegebenfalls die Korrektur durchzuführen. Hierbei ist mit großer Rechenintensität zu rechnen.

Weiters besteht die Notwendigkeit, dass Domänenexperten nach Durchsicht der fehlerhaft markierten Attribute Entscheidungen über die Korrektheit treffen, da nicht alle Attribute, die durch den „Data cleaning“-Vorgang als fehlerhaft markiert wurden, auch fehlerhaft sind. Bei dieser schwierigen Entscheidung, ob ein Attribut fehlerhaft ist oder nicht, hat der Experte das letzte Wort. Da diese Interaktionen zeitaufwendig und kostspielig sind, werden Protokolle mitgeführt, die eine nachträgliche Bearbeitung gewährleisten [HM03, S. 12]. Nach der Korrektur der fehlerhaften Werte können nach Kübart et al. [KGH05, S. 24] vier Kategorien für die Einteilung der Ergebnisse bestimmt werden:

- „eine fehlerhafte Instanz wird als Fehler erkannt (richtig positiv);
- eine fehlerhafte Instanz bleibt unerkannt (falsch negativ);
- eine nicht fehlerbehaftete Instanz wird als Fehler eingeordnet (falsch positiv);
- eine nicht fehlerbehaftete Instanz wird nicht als Fehler eingeordnet (richtig negativ).“[KGH05, S. 24]

Tabelle 6 zeigt die vier Kategorien noch einmal in graphischer Darstellung:

	Testergebnis positiv	Testergebnis negativ
Instanz fehlerhaft	richtig positiv (RP)	falsch negativ (FN)
Instanz korrekt	falsch positiv (FP)	richig negativ (RN)

Tabelle 6: Kategorieeinteilung der Testfälle in Anlehnung an [KGH05, S. 24]

3.3.4 Nachbearbeitung (Post-processing / Control)

Als letzter Schritt des „Data cleaning process“ werden die Ergebnisse des getätigten Workflow noch auf Richtigkeit überprüft. Jene Werte, die nicht automatisch korrigiert werden konnten, werden nachträglich manuell geändert. Lernfähige Systeme nehmen diese Änderungen in den nächsten Durchlauf des Prozesses auf. Im Anschluss daran kann der Prozess zur weiteren Fehlerfindung/-behebung erneut angestoßen werden [HM03, S. 12].

4 Ursachen für mangelnde Datenqualität

In diesem Kapitel werden die möglichen Fehlerquellen, welche die Datenqualität negativ beeinflussen, kurz dargestellt. Hierbei wird eine Unterteilung nach der Herkunft von Fehlerquellen getroffen. Fehler können in einzelnen Datenquellen auftreten oder auch erst bei der Zusammenführung von mehreren Datenquellen entstehen. In einem weiteren Schritt wird eine genauere Klassifizierung der Fehlermöglichkeiten durchgeführt und diese übersichtlich dargestellt.

4.1 Klassifikation der Datenbankprobleme

In erster Linie wird zwischen vier Problemfeldern unterschieden. Probleme treten als „single-source“ oder „multi-source“ bzw. auf Schema- oder Instanzebene auf. Folgende Tabelle veranschaulicht die Einteilung in „single-source“-Schemaebene (A), „single-source“-Instanzebene (B), „multi-source“-Schemaebene (C) und „multi-source“-Instanzebene (D):

Schema	A	C
Instanz	B	D
	„single-source“	„multi-source“

Tabelle 7: Klassifikation von Datenbankproblemen

Abbildung 9 zeigt diese Klassifizierung in übersichtlicherer Form. Es werden im Zuge der Klassifizierung auch gleich mögliche Fehlerquellen beschrieben. Zusammenfassend lässt sich feststellen, dass sich auf der Ebene des Datenbankschemas Probleme aus schlechtem Datenbankdesign ableiten und sich mittels eines verbesserten Schemadesigns vermindern lassen. Probleme der Instanzebene spiegeln hingegen Fehler und Inkonsistenzen der Daten wider und sind auf der Ebene des Schemas auch nicht sichtbar. Auf der Bereinigung von Instanzproblemen liegt ein Hauptaugenmerk des „Data cleaning“ Prozesses [RD00, S. 2].

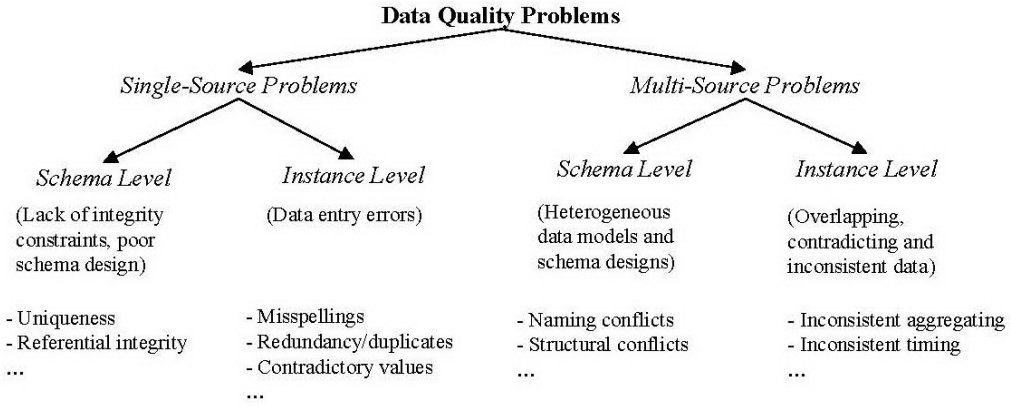

Abbildung 9: Klassifikation der Datenqualitätsprobleme [RD00, S. 3]

4.1.1 Single-source Probleme

Die Wahrscheinlichkeit des Auftretens von Single-source Problemen wird von der Art der Datenhaltung beeinflusst. Steht ein Datenbank-Managementsystem (DBMS) hinter der Datenbank, so ist es möglich sehr viele Probleme auf Grund von Schemadefinitionen und Integritätsbedingungen abzuwenden. Bei unstrukturierten bzw. offenen Datenhaltungen existieren keine Beschränkungen in Form eines Schemas. In diesen Fällen ist eine höhere Fehlerwahrscheinlichkeit gegeben [RD00, S. 3].

Scope / Problem		Dirty Data	Reasons/Remarks
Attribute	Illegal values	bdate=30.13.70	values outside of domain range
Record	Violated attribute dependencies	age=22, bdate=12.02.70	age = (current date - birth date) should hold
Record type	Uniqueness violation	emp1=(name="John Smith", SSN="123456") emp2=(name="Peter Miller", SSN="123456")	uniqueness for SSN (social security number) violated
Source	Referential integrity violation	emp=(name="John Smith", deptno=127)	referenced department (127) not defined

Tabelle 8: Darstellung von Single-source Problemen auf Schemaebene [RD00, S. 3]

Tabelle 8 zeigt eine mögliche Einteilung von Fehlern, die auf Schemaebene auftreten. Rahm und Do [RD00, S. 3] stellen dabei auch dar, dass sich der Fehler auf ein einzelnes Attribut konzentrieren kann. Es ist möglich, dass aber auch mehrere Attribute betroffen sind, wenn diese direkt in Beziehung zueinander stehen (z.B. Geburtsdatum und Alter). Tabelle 9 (siehe unten) zeigt mögliche Problemfälle auf

der Instanzebene auf. Diese umfassen z.B. einfache Schreibfehler für ein einzelnes Attribut sowie Fehler die auf die Eingabe von mehreren Attributen und ihrer Beziehungen verweisen.

Scope / Problem		Dirty Data	Reasons/Remarks
Attribute	Missing values	phone=9999-999999	unavailable values during data entry(dummy values or null)
	Misspellings Cryptic values, Abbreviations	city=“Liipzig“ experience=“B“; occupation=“DB Prog.“	usually typos, phonetic errors
	Embedded values	name=“J. Smith 12.02.70 New York“	multiple values entered in one attribute (e.g. in a free-form field)
	Misfielded values	city=“Germany“	
Record	Violated attribute dependencies	city=“Redmond“, zip=77777	city and zip code should correspond
Record type	Word transpositions	name1= „J. Smith“, name2=“Miller P.“	usually in a free-form field
	Duplicated records	emp1=(name=“John Smith“,...); emp2=(name=“J. Smith“,...)	same employee represented twice due to some data entry errors
	Contradicting records	emp1=(name=“John Smith“, bdate=12.02.70); emp2=(name=“John Smith“, bdate=12.12.70)	the same real world entity is described by different values
Source	Wrong references	emp=(name=“John Smith“, deptno=17)	referenced department (17) is defined but wrong

Tabelle 9: Darstellung von Single-source Problemen auf Instanzebene [RD00, S. 3]

4.1.2 Multi-source Probleme

Da Daten, die in mehreren Datenbanken abgelegt sind, oftmals nicht nach dem gleichen Schema verwaltet werden, entstehen besonders bei der Zusammenführung von Daten aus mehreren Datenquellen Fehler. Allgemeines Ziel der Integration von DBs ist neue und zusätzliche Information zu speichern, gleichzeitig aber Redundanzen zu verhindern [Ger05, S. 2]. Jede Datenquelle für sich kann schon Fehler enthalten bzw. die Darstellung der Daten kann anders erfolgen [RD00, S. 4]. Diese schon vorhandenen Fehler behindern oder verhindern das Zusammenführen der Datenquellen. Es ist dadurch auch leicht möglich unerwünschte Duplikate zu erzeugen. Dieses Problem zeigt sich besonders bei der unterschiedlichen Darstellung von Attributen bzw. Währungs- und Maßeinheiten [RD00, S. 4], wie z.B.

- Geschlecht: m/w; 0/1
- Euro/Dollar
- Kilo/Pfund
- Zentimeter/Zoll

Zusätzliche Schwierigkeiten bereitet die notwendige Integration der unterschiedlichsten Schemata. Hierbei entstehen vor allem Namens- und Strukturkonflikte [KS91]. Namenskonflikte deuten darauf hin, dass ein und derselbe Name in den Datenquellen für verschiedene Attribute verwendet wird bzw. das Attribut eine andere Bezeichnung besitzt. Strukturkonflikte zeigen sich in unterschiedlichen Datentypen bzw. legen unterschiedlich definierte Integritätsbedingungen zu Grunde [Ger05, S. 2].

4.2 Klassifikation der Anomalien / Fehlerquellen

In diesem Kapitel werden die Anomalien bzw. Fehler kurz vorgestellt, nach denen im Rahmen des „data cleaning“ gesucht wird. Es werden hierbei drei Fehlerkategorien unterschieden:

1. Syntaktische Fehler
2. Semantische Fehler
3. „Coverage“ Probleme

Unter syntaktische Anomalien fallen „Lexical error“, „Domain format error“ und „Irregularities“. Diese Fehler beziehen sich auf Fehler in Bezug auf das Format der Attribute sowie auf die Darstellung von Werten. „Constraint violation“, „Duplicates“ und „Invalid tuple“ sind die Hauptkomponenten von semantischen Anomalien. Diese Fehler sind dafür verantwortlich, dass die Datenrepräsentation nicht ausreichend und unter Umständen redundant ist. Durch Coverage Probleme wie „Missing value“ und „Missing tuple“ werden schließlich die Anzahl von Werten, die den realen Daten entsprechen, verringert [HM03, S. 6]. In den folgenden Unterkapiteln erfolgt eine genauere Erläuterung dieser Fehlermöglichkeiten.

Tabelle 10 bringt die in Folge vorgestellten Anomalien mit den in Kapitel 3.2 dargestellten Qualitätsdimensionen übersichtlich in Verbindung.

	Completeness	Validity	Schema conform.	Density	Uniqueness
Lexical error		-	•	-	-
Domain format error		-	•		-
Irregularities		-			-
Constraint Violation		•			
Misssing Value				•	-
Missing Tuple	•				
Duplicates					•
Invalid Tuple		•			

Tabelle 10: Fehlerquellen und Qualitätskriterien in Anlehnung an [HM03, S. 10]

Jeder • bedeutet eine direkte Abstufung des Qualitätskriteriums; während - bedeutet, dass beim Auftreten dieser Anomalie das Finden von anderen Anomalien als Abstufung des Qualitätskriteriums beeinträchtigt wird [HM03, S. 10].

4.2.1 Strukturprobleme (Lexical error)

„Lexical errors“ zeigen sich in Strukturproblemen in Bezug auf das Format der Daten. Es ist möglich, dass Daten zum Zeitpunkt des Datentransfers nicht verfügbar sind. Dadurch werden einige Datensätze nicht richtig abgebildet. Dies tritt häufig dann auf, wenn kein Schema der Daten verfügbar ist und die Attributwerte durch einzelne Trennzeichen separiert sind.

Tabelle 11 zeigt ein Beispiel eines solchen Fehlers. Hierbei wird angenommen, dass die Tabelle vier Spalten hat und eine Übertragung der Datenwerte ohne Schema erfolgt. Da jedoch einige Werte nicht existieren, ist es möglich, dass vorhandene Werte in falsche Spalten geschrieben werden. In diesem Fall fehlt im dritten Datensatz der Wert für das Geschlecht, dadurch wird der Wert der Größe um eine Spalte zu früh eingetragen. Es stimmt so die aktuelle Struktur mit dem definierten Schema nicht mehr überein [HM03, S. 10].

Name	Alter	Geschlecht	Größe
Alex Mayer	23	M	194
Markus, Mair	27	M	187
Karl, Maier	34	M	
Josef, Meier	21	187	

Tabelle 11: Beispiel für einen „Lexical error“

4.2.2 Fehler auf Grund des Domänenformates (Domain format error)

„Domain format errors“ stellen Fehler dar, bei denen der Wert eines Attributes nicht mit dem definierten Format übereinstimmt. Hierbei gibt das System eine bestimmte Formatierung vor, welche aber von den vorhandenen bzw. eingespielten Daten nicht befolgt wird.

Als Beispiel dient ebenfalls Tabelle 11. Im ersten Moment ist kein Fehler zu erkennen. Bei näherer Betrachtung des Attributes Name und seines Formates, welches mit $\{\sum D^*, \sum D^*\}$ spezifiziert ist, wird der Fehler erkennbar [HM03, S. 10]. Diese Beschreibung bedeutet, dass zuerst beliebig viele Buchstaben geschrieben sind, danach folgt ein Beistrich, welcher wieder von beliebig vielen Buchstaben gefolgt wird. Das Attribut Name müsste somit in Zeile eins statt des Wertes „Alex Mayer“ den Wert „Alex, Mayer“ besitzen, da hier der vom Format geforderte Beistrich fehlt.

4.2.3 Widersprüche (Irregularities)

„Irregularities“ spiegeln Widersprüche im Gebrauch von Werten, Abkürzungen und Einheiten wider. Dies ist ein häufiges Problem bei der Zusammenführung von mehreren Datenquellen (siehe Kapitel 4.1.2).

Tabelle 12 stellt dieses Problem übersichtlich dar. In diesem Beispiel wird die Währung nicht mit dem Preis angegeben. In einer Datenbank wird der Preis in Euro angegeben und in der zweiten Datenbank in Yen. Durch die Zusammenführung beider Datenbanken werden beide Preise angeführt. Die Preise werden korrekt angezeigt, aber ohne Hintergrundwissen werden die Daten falsch interpretiert.

ID	Titel	Preis	Altersbeschränkung
4654546	Shrek	9.99	ohne
4895646	Shrek II	1,659.65	ohne

Tabelle 12: Beispiel für „Irregularities“

4.2.4 Verletzung von Integritätsbedingungen (Integrity Constraint Violation)

„Integrity constraint violation“ beschreibt die Verletzung von Integritätsbedingungen durch Tuple. Jede Bedingung besteht aus einer Regel, die Wissen über die Wirklichkeit transportiert. Als Beispiel sei hier das Alter angeführt: Das Alter muss z.B. immer ≥ 0 sein [HM03, S. 6].

In diesem Zusammenhang sind auch die „Contradictions“ zu erwähnen. Es handelt sich hierbei um Fehler, die auf Grund von Beziehungen zwischen zwei Attributen

auftreten (siehe Kapitel 3.2.3). Diese Fehler sind entweder Verletzungen von Integritätsbedingungen oder es handelt sich dabei um Duplikate, die jedoch mit falschen Werten abgebildet sind. Deshalb werden sie in weiterer Folge nicht als eigenständige Fehlerquellen betrachtet [HM03, S. 7].

4.2.5 Duplikate (Duplicates)

Unter „Duplicates“ werden zwei oder mehrere Tupel verstanden, die ein und denselben Wert aus der realen Welt repräsentieren. Diese Duplikate müssen nicht genau dieselben Werte annehmen, sondern besitzen in ein oder mehreren Attributen Abweichungen. Sie müssen nur für dasselbe Objekt in der realen Welt stehen [HM03, S. 7]. Da dieses Problem in der Literatur schon lange behandelt wird, ist die Behandlung unter mehreren Begriffen auffindbar (z.B. „merge/purge problem“ [HS95], „data deduplication and instance identification“ [EIV07]). In dieser Arbeit wird unter Anomalie der Duplikate das Finden und Aufspüren von Duplikaten verstanden.

Tabelle 13 zeigt das Problem eines Duplikates. Eine Person befindet sich zweimal in einer Tabelle, weil Straße und PLZ und Ort unterschiedliche Werte aufweisen und somit durch das System angenommen wurde, dass es sich um zwei verschiedene Personen handelt.

ID	Name	Geschlecht	GebDatum	Straße	PLZ	Ort
4465	Max Mustermann	M	17.11.83	Marktplatz 6	4170	Haslach
4864	Max Mustermann	M	17.11.83	Hauptplatz 42	4040	Linz

Tabelle 13: Beispiel für „Duplicates“

4.2.6 Falscher Datensatz (Invalid Tuple)

„Invalid tuples“ zeigen keinen offensichtlichen Fehler, repräsentieren aber keinen Wert der realen Welt. Sie resultieren vielmehr aus der Unfähigkeit heraus, die reale Welt formal mit Hilfe von geeigneten Regeln zu beschreiben. Da die Auffindung dieser Fehler sehr schwierig ist, gestaltet sich die Korrektur dieser noch schwieriger bzw. ist fast unmöglich, weil keine Regeln zur Korrektur bekannt sind [HM03, S. 7].

Als Beispiel dient Tabelle 11. In dieser ist der Datensatz „Markus, Mair; 27; M; 187“ eingetragen. Voraussetzung für eine Aufnahme in diese Tabelle ist, dass ein Kundenkontakt zu dieser Person besteht. Dieser Datensatz repräsentiert jedoch eine Person, mit der noch nie ein Kontakt bestanden hat und dürfte somit nicht eingetragen sein.

4.2.7 Fehlender Wert (Missing Value)

„Missing values“ sind oft das Resultat von Einsparungen bei der Suche nach Daten. Dieser Fehler äußert sich oft im Gebrauch von NULL-Werten, welche einer besonderen Behandlung bedürfen (siehe Kapitel 3.2.2). Fehlende Werte können zusätzlich auch eine Integritätsverletzung bedeuten, sofern NOT NULL als „constraint“ definiert wurde [HM03, S. 7].

Das Fehlen von Werten erschwert die Durchführung von Analysen besonders. Es sind oftmals zu wenig Daten vorhanden, um die Berechnung eines „richtigen“ Ergebnisses zu ermöglichen. Dadurch ist es leichter möglich, Entscheidungen auf Grund falscher Information zu treffen, da statistische Berechnungen falsch durchgeführt wurden bzw. Ergebnisse eine Verfälschung erfahren.

Wie in Tabelle 14 dargestellt, bilden NULL-Werte oft die Basis für fehlende Werte, da diese dann als Platzhalter verwendet werden.

ID	Vorname	Nachname	Email
1	Alex	Mayer	alex.mayer(at)students.jku.at
2	Franz	Maier	NULL

Tabelle 14: Beispiel für fehlende Werte (Auszug aus Tabelle 2)

4.2.8 Fehlende Tupel (Missing Tuple)

„Missing Tuples“ bezeichnet den Fehler, wenn Daten der realen Welt nicht in der Datenbank repräsentiert werden [HM03, S. 10]. Dieser Fehler tritt auf, wenn ein falscher „Join“-Operator bei der Zusammenführung von zwei oder mehreren Tabellen verwendet wird. Fehlende Tuple sind ein häufiges Problem bei der Zusammenführung von verschiedenen Tabellen. Daher ist es wichtig, den passenden „Join“-Operator zu wählen, um Fehler von vornherein zu vermeiden.

Wählt man z.B. den „Äußeren“ Verbund rechts (right outer join), wie in Tabelle 15 ersichtlich, so bildet dieser eine Kombination aller rechten Tupel (Relation R) mit allen passenden linken Tupel (Relation S), wobei links notfalls mit leeren Feldern aufgefüllt wird. Unter passenden Tupel sind in diesem Fall jene Datensätze der Relation S zu verstehen, deren Wert der Spalte C mit den Werten der Spalte C der Relation R übereinstimmen. Dies ergibt dann NULL-Werte in der Endrelation (Relation erg).

S	A	B	C		R	C	D	E		ERG	A	B	C	D	E
	1	1	1	Right Outer		1	1	1			1	1	1	1	1
	2	2	2	Join		3	2	2			-	-	3	2	2

Tabelle 15: Beispiel für fehlende Tupel

4.3 Schlussfolgerungen

In Kapitel 3 und 4 wurden die in der Literatur relevanten Begriffe erläutert. Diese Erläuterungen stellen die Grundlagen für die weitere Arbeit dar. Anhand dieser ist es möglich die IST-Situation der OÖGKK (siehe Kapitel 5) entsprechend zu analysieren und die spezifischen Anforderungen der OÖGKK zu erarbeiten, sodass ein Überblick über die benötigte Funktionalität des einzusetzenden Frameworks geschaffen wird.

5 Ist-Analyse OÖGKK

Die Oberösterreichische Gebietskrankenkasse (OÖGKK) bietet im Rahmen von Dienstleistungsbeziehungen mit anderen Sozialversicherungsträgern sowie für Projekte Leistungen zur Erstellung und zum Betrieb von DWHs an. Es werden als interner Dienstleister für die Sozialversicherung mehrere Data Warehouse Produkte betrieben. Für die Bearbeitung und Speicherung der Daten werden DWH-Produkte auf Basis der SAS® Business Intelligence (BI) Plattform [SAS08c] eingesetzt.

Dieses Kapitel zeigt die Ist-Situation der OÖGKK als Fallstudie, sodass entsprechend der vorhandenen Anforderungen und Voraussetzungen ein Werkzeug spezifiziert werden kann, welches identifizierte Fehlerquellen (siehe Kapitel 5.3.2) bearbeitet und eine Plausibilitätskontrolle automatisch durchführt. Die OÖGKK tritt hierbei als Auftraggeber auf. Zu diesem Zweck teilt sich dieses Kapitel in folgende Themengebiete auf:

- Organisationsstruktur
- IT-Architektur
- DWH-Produkt FOKO
  - FOKO - Datenaufbau
  - FOKO - Fehlerquellen
  - FOKO - Beispieldaten
- Anforderungen

Die Erarbeitung dieser Kapitel erfolgt auf Basis der von der OÖGKK zur Verfügung gestellten Dokumente und Datensätze.

5.1 Organisationsstruktur

Für die Durchführung der Aufgaben der Oberösterreichischen Gebietskrankenkasse im Rahmen der Selbstverwaltung ist das Büro zuständig. Dieses Büro ist in die vier Teilbereiche Ressourcen & Information, Strategie & Führung, Vertragspartner sowie Kundenbetreuung & Gesundheit gegliedert (siehe Abbildung 10). Die OÖGKK beschäftigt rund 2.000 Arbeitnehmer.

Für die Entwicklung und Betreuung der DWH-Produkte ist ein spezielles Team verantwortlich, welches Teil der IT-Entwicklungsabteilung der OÖGKK (Abteilung IT-E) ist. Dieses DWH-Team umfasst derzeit 9 interne und 6 externe Mitarbeiter (Stand 06/2008). Die fachliche Betreuung der jeweiligen DWH-Produkte erfolgt

durch Projektteams aus anderen Fachabteilungen (z.B. Kundenservice, Behandlungsökonomie, usw.) sowie anderer Krankenversicherungsträger.

Eine Zusammenfassung des Aufbaus des Büros der OÖGKK zeigt das folgende Organigramm[2].

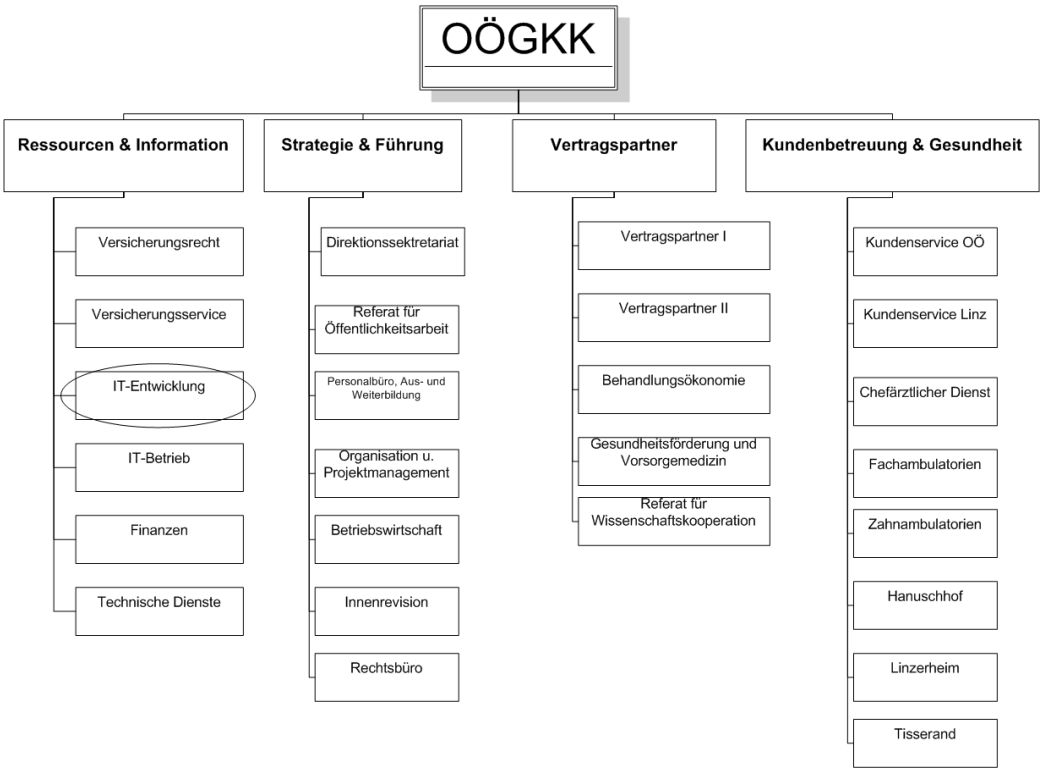

Abbildung 10: Vereinfachtes Organigramm der OÖGKK

5.2 IT-Architektur

Sämtliche DWH-Produkte der OÖGKK werden auf Basis der SAS® Business Intelligence (BI) Plattform entwickelt und betrieben. Das dem SAS® Server zugrunde liegende Betriebssystem ist Unix/AIX. Hierbei ist die komplette SAS® BI-Produktpalette [SAS08c] wie folgt im Einsatz: Als Frontend dienen den Benutzern das SAS® Web Report Studio [SAS08f] / SAS® Information Delivery Portal [SAS08e] - ein SAS® Base Client[SAS08a] mit implementierten Applikationen - und der SAS® Enterprise Guide [SAS08d]. Abbildung 11 zeigt den Aufbau der SAS® Client- / Server Architektur in übersichtlicher Form.

[2]vgl. http://www.ooegkk.at/mediaDB/134350.PDF, download am 23.07.08

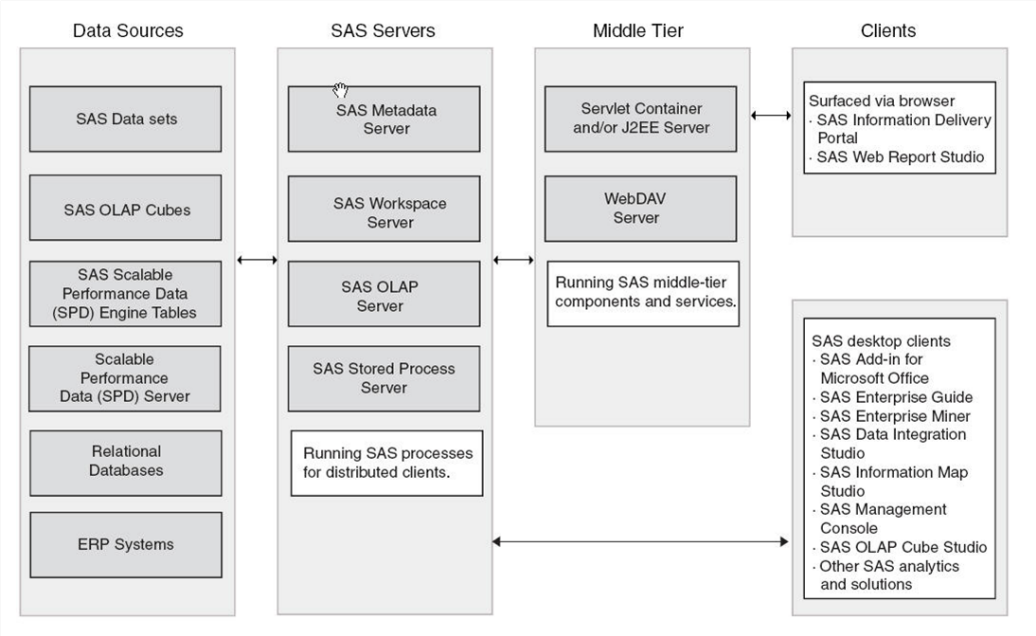

Abbildung 11: Übersicht: Aufbau SAS®Business Intelligence (BI) Plattform [SAS08g]

Die Folge-Kostenanalyse (siehe FOKO Kapitel 5.3) ist eine Client- / Serveranwendung. Alle Batch- und Reportfunktionen laufen auf dem Server. Alle Funktionen, die Ergebnisse auf den Computer liefern, bzw. alle Dialogfunktionen (Steuerfunktionen) sind auf den Clients installiert. Die Ausgangsdaten bilden verschiedene ASCII-Flatfiles, die von jedem Krankenversicherungsträger geliefert werden (siehe Abbildung 12).

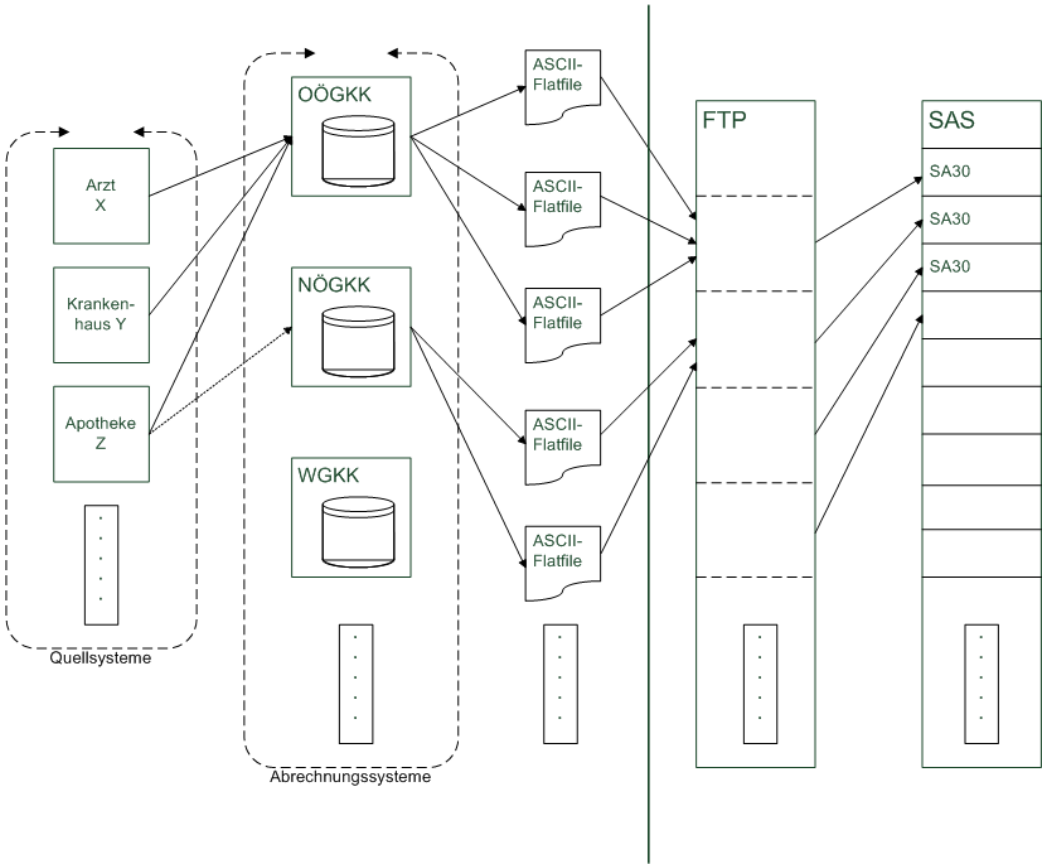

Abbildung 12: Übersicht Datenstrom

5.3 DWH-Produkt FOKO

Das DWH-Produkt FOKO (Folgekosten) ist ein Werkzeug, das die Analyse verschiedener Vertragspartnerabrechnungsdaten (z.B. Anzahl der Krankentransporte, Notarzteinsätze usw.) ermöglicht. Als Vertragspartner der Krankenversicherungsträger gelten:

- Apotheken
- Ambulanzen
- Ambulatorien
- Kur- und Erholungsheime
- Krankenhäuser
- Transportunternehmer
- Ärzte

Durch den Einsatz von FOKO wird vor allem Kostentransparenz geschaffen. Es stehen aber auch andere Ziele im Fokus, um Einsparungspotentiale bzw. eine Übersicht über alle getätigten Leistungen zu erhalten. Im Einzelnen verfolgte Ziele sind:

1. Eigen- und Folgekostentransparenz bei Vertragsärzten (Praktiker, Fachärzte).
2. Eigenkostentransparenz bei Zahnärzten und Wahlärzten.
3. Kostentransparenz bei Spitälern, Spitalsambulanzen, Transportunternehmen, Bandagisten, Optikern, Orthopädie- und Schuhmachern, sowie Apotheken, um bei Bedarf Markt- bzw. regulative Kostensenkungsprogramme erarbeiten zu können.
4. Entwicklungsbeobachtung bei Eigen- und Folgekosten (z.B. Vergleich zwischen 4. Quartal 2007 und 4. Quartal 2006 je Arzt; Istwert aus dem früheren Quartal wird zum Sollwert des späteren Quartals).
5. Häufigkeit der Inanspruchnahme aller Leistungen aus dem Vertragspartnerbereich (je Vertragspartner und gesamt) auf Basis eines Vergleiches zweier Beobachtungszeiträume.
6. Informationen über den nicht verdichteten Datenbereich mittels ad hoc Abfragen (Queries).

FOKO wird von allen Krankenversicherungsträgern eingesetzt. Daher ist es ein so genanntes Standardprodukt der Sozialversicherung. Es werden Analysen über Anzahl und Kosten der von den Vertragspartnern erbrachten Leistungen durchgeführt. Dies trägt wesentlich zu einer Erhöhung der Kostentransparenz bei. Dabei muss jedoch sichergestellt sein, dass alle für die Analysen benötigten historischen Daten gespeichert sind.

Hierbei gibt es strukturelle Trends, die selten gebrochen werden. Zur Veranschaulichung sowie zur Erarbeitung dieser Diplomarbeit wurden von der OÖGKK anonymisierte Daten zur Verfügung gestellt.

5.3.1 FOKO - Datenaufbau

Um FOKO mit Daten zu befüllen ist eine Datenschnittstelle (DS) nötig. Als DS sind ASCII-Textdateien definiert, deren Zeilen (Datensätze) einem definierten Schnittstellenformat entsprechen. Die DS wird mit Rohdaten aus allen die FOKO betreffenden Bereichen beschickt. Diese Daten kommen von den operativen Quellsystemen, die bei den unterschiedlichen Krankenversicherungsträgern im Einsatz sind.

Die DS bildet die Eingabedatei, auf dem FOKO aufsetzt. Die Datensätze setzen sich aus verschiedenen Satzarten zusammen. Eine taxative Aufzählung der Satzarten findet sich im Anhang unter Punkt A. Insgesamt werden je Quartal gesamt

circa 40 GB geladen. Dabei wird je nach Satzart eine unterschiedliche Anzahl von Sätzen geladen.

Die Übermittlung der Daten erfolgt mittels Aufteilung auf die Satzarten <200 (Leistungsdaten) und ≥ 200 (Stammdaten), d.h. die Leistungsdaten werden von den Stammdaten getrennt übernommen. Erstere werden in einer ASCII-Datei mit Namen „LEISTLAD“ vom zentralen FTP-Server eingespielt. Alle übrigen Stammsätze werden mit der Datei „STAMM“ übergeben. Diese Aktion wird zentral veranlasst.

Bei der Einspielung erfolgt eine Historisierung der Stammdatensätze, d.h. es erfolgt ein Update der DB. Bestehende Stammdatensätze, die im neuen Input nicht mehr vorhanden sind, werden mit den eingehenden Daten nicht gelöscht [Inm02, S. 34f]. Das System FOKO übernimmt alle Datensätze, in denen zumindest die Satzart und der Zeitraum angegeben sind. Diese Daten werden dann in SAS-Tabellen abgelegt, welche mit Hilfe eines Starschemas [KR02, S. 21f] aufgebaut sind.

Dem System wird auch noch eine Datei „LEISTNV“ übergeben, welche Nachverrechnungen zu bereits vorhandenen Beständen enthält. Im File 'LEISTNV' muss immer der gesamte Bestand einer Satzart übergeben werden, da die bereits im System vorhandenen Daten zur Gänze gelöscht und mit den Nachverrechnungsdaten neu aufgebaut werden. Bevor die Daten durch den Anwender endgültig in die DATA-Library übergeben werden, liegen alle in einer temporären Datei, in der sie auf Grund von Prüfroutinen noch einmal auf Fehler überprüft werden können.

Die Daten werden zur weiteren Verwendung verdichtet, d.h. die Daten werden je Bereich und Beobachtungszeitraum verdichtet und in eigenen Dateien abgespeichert, sodass sie für die Standardabfragen verwendbar sind. Alle Satzartentabellen (Schnittstellendaten) und die verdichteten Daten liegen in verschiedenen Dateien in der sogenannten DATA-Library (z.B DATA.SA10). Der genaue Aufbau dieser DATA-Library findet sich im Anhang unter Abschnitt B. Die DATA-Library ist auch zur Sicherung der Daten notwendig. Die Sicherung muss über das Betriebssystem durchgeführt werden. Die Verantwortung liegt somit bei den einzelnen Anwendern.

Abbildung 13 veranschaulicht den Aufbau der Datenübernahme sowie der Dateneinspielung in die FOKO.

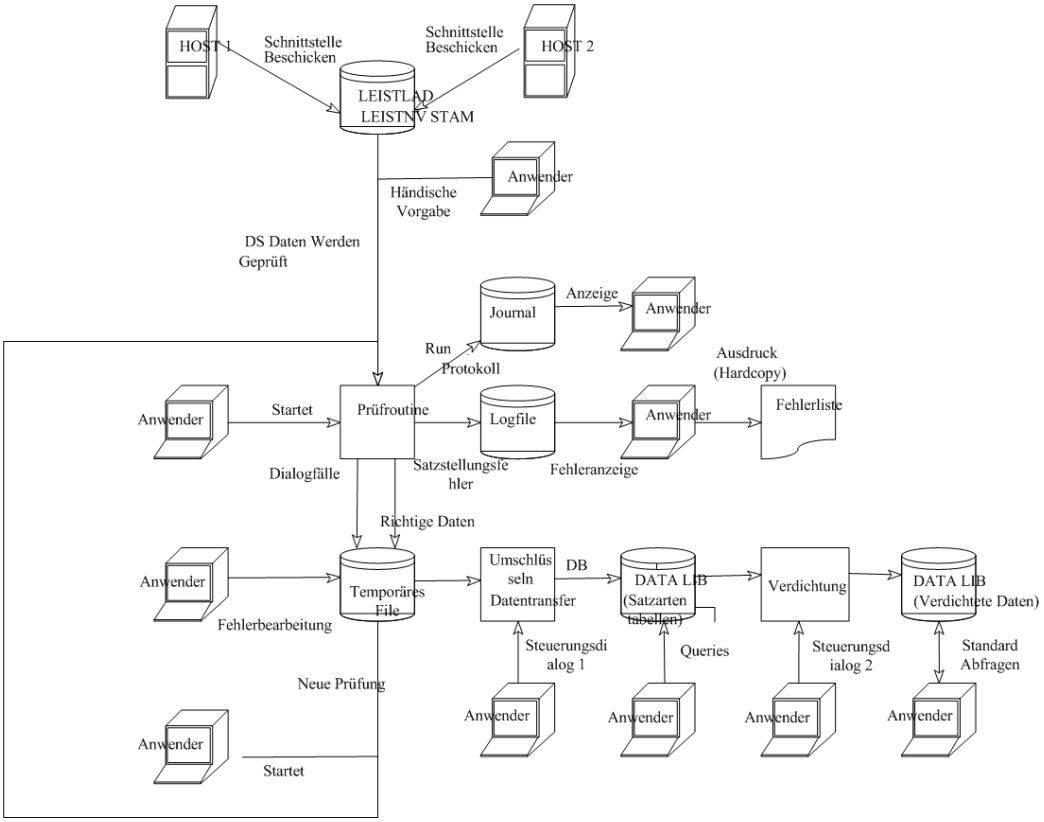

Abbildung 13: Ablauf der Dateneinspielung

5.3.2 FOKO - Fehlerquellen

Um in einem ersten Schritt mögliche Fehlerquellen identifizieren zu können, ist eine Klassifizierung der möglichen Auswertungen notwendig, sodass sowohl allgemeine Ursachen für Fehler gefunden als auch für jeden Punkt speziell zutreffende Fehler identifiziert werden können. Eine Auswertung der Daten kann auf Grund dieser Aufzählung vorgenommen werden:

- Kategorie - Darunter ist eine Zusammenfassung gleichartig zu behandelnder und bewertender Positionen zu verstehen. In der Statistik stellt die Kategorie eine Verdichtungsebene dar, für die Kennzahlen errechnet werden.
- Arzt - Je Arzt werden Ist- und Sollwerte im Vergleich mit dem auf die jeweilige Vergleichskategorie (z.B. Alterskategorie) bezogenen Durchschnitten ermittelt und allfällige Abweichungen dargestellt.
- Vertragspartnerauswertungen - Bei diesen Auswertungen gibt es keine Durchschnittsbildung sondern einen Vergleich des gewählten Beobachtungszeitraumes mit dem gleichen Zeitraum aus den Vorjahren je Vertragspartner.

- Mengen- / Preiskomponente - Alle Auswertungen basieren grundsätzlich auf diesen beiden Größen, damit kann jede Statistik in Richtung „zu viel“ oder „zu teuer“ interpretiert werden.
- Altersklassen - Alle Daten und Berechnungen im Eigen- und Folgekostenbereich werden auf Alterskategorien der Patienten (10-Jahressprünge) aufgeteilt.
- Beobachtungszeitraum - Der Beobachtungszeitraum ist als Parameter definiert und kann ein Monat, ein Quartal oder ein Jahr sein. Es muss darauf geachtet werden, dass bei den zu vergleichenden Daten jeweils derselbe Parameter zur Berechnung herangezogen wird, da ansonsten Entwicklungsbeobachtungen nicht möglich sind.

Im Zusammenhang mit dieser Diplomarbeit werden lediglich Fehlerursachen sowie auftretende Fehler dargestellt, welche auf Grund der zur Verfügung gestellten Testdaten identifiziert wurden. Dies sind folgende fünf Hauptfehlerquellen:

- Tupel der Zieldateien werden vom ETL-Job nicht erzeugt, weil die dazu nötigen Daten (z.B Stammdaten) nicht in den Quelldateien vorhanden sind. Konsequenz: Es werden zu wenige Datensätze auf Grund der Analyse angezeigt.
- Problem von Mehrfach-Datensätzen (Duplikate). Konsequenz: Es werden zu viele Datensätze auf Grund der Analyse angezeigt.
- Tupel der Quelldateien werden vom ETL-Job nicht geladen, weil sie von diesem nicht als zu ladend erkannt werden.
- Es werden falsche Werte angezeigt (Tippfehler).
- Es werden Tupel gespeichert, denen kein Wert aus der Wirklichkeit zu Grunde liegt.

Werden Tupel nicht erzeugt, weil die Daten nicht in den zugehörigen Quelldateien vorhanden sind, so ist dieses Problem nicht im Ladeprozess zu beheben. Da in diesem Fall fehlerhafte Daten zur Verfügung gestellt wurden, muss eine neue Datei angefordert werden.

Weiters kann auf Grund von statistischen Methoden (z.B. Standardabweichung) festgestellt werden, ob bei einer Analyse eine dieser Fehlervarianten aufgetreten ist. Durch die Festsetzung von Referenzwerten bzw. der Festsetzung von Grenzwerten kann eine Filterung der Daten vorgenommen werden. Referenzwerte geben eindeutige Werte an, die von den Daten angenommen werden müssen. Grenzwerte bilden eine obere bzw. untere Schranke, in denen die analysierten Werte liegen müssen. Strukturtreue deutet darauf hin, dass die untersuchten Werte sich in dem

festgelegten Grenzbereich befinden bzw. die geforderten Werte repräsentieren. In Abbildung 14 kann eine solche Strukturtreue festgestellt werden. In dem Beobachtungszeitraum wurden keine Werte entdeckt, die auf Problemstellen hindeuten.

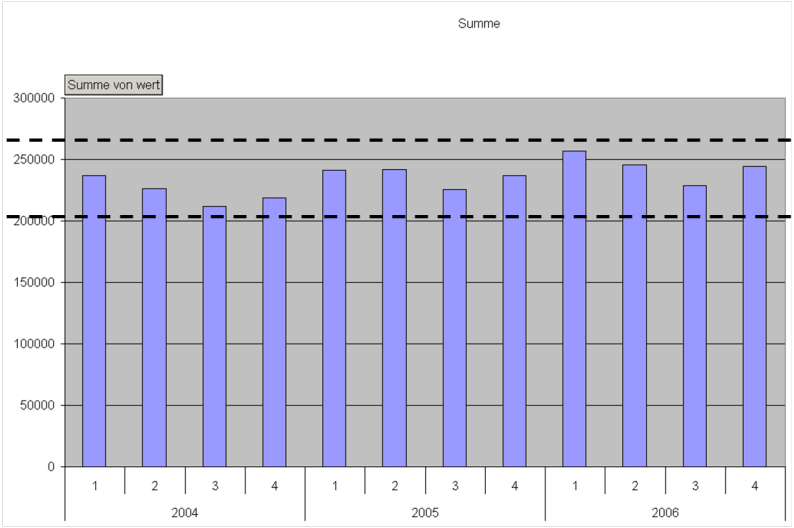

Abbildung 14: „Konsistenz“ der Werte

Die Abbildungen 15 und 16 zeigen Beispiele von Ausreißern auf. Hierbei finden die beschriebenen Grenzwerte ihren Einsatz. Befindet sich eine Vergleichsperiode nicht innerhalb dieser Schranken, wird eine vorher definierte Aktion (z.B Generierung einer Warnmeldung) veranlasst, sodass der Fehler genauer analysiert und die zu Grunde liegende Fehlerursache aufgedeckt werden kann.

In Abbildung 16 lässt sich dennoch nicht eindeutig feststellen, welche Werte die Ausreißer darstellen, da jeweils in den dritten Quartalen erhöhte Werte auftreten. In diesem speziellen Fall kann nicht ad hoc entschieden werden, in welchen Quartalen sich die Ausreißer befinden. Dazu ist noch eine weitere Überprüfung der historischen Daten notwendig, um ausschließen zu können, dass die Werte im dritten Quartal immer stärker ausgeprägt sind.

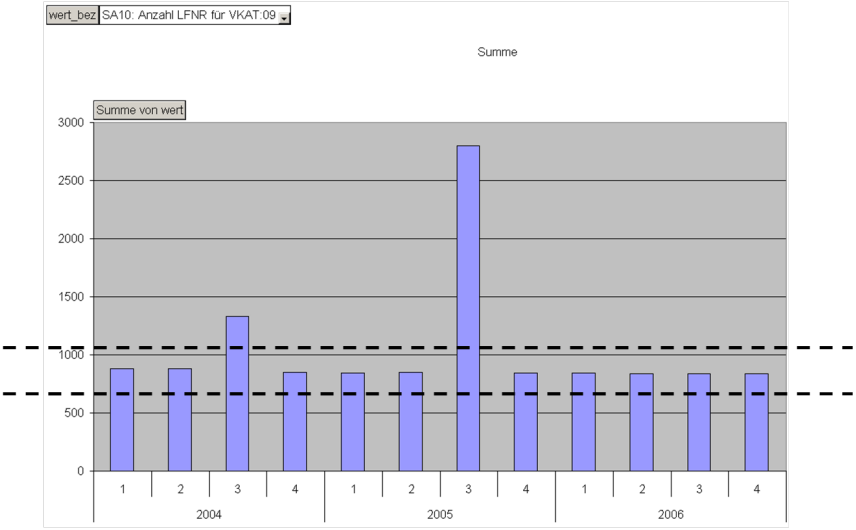

Abbildung 15: Beispiel für Ausreißer (1/2)

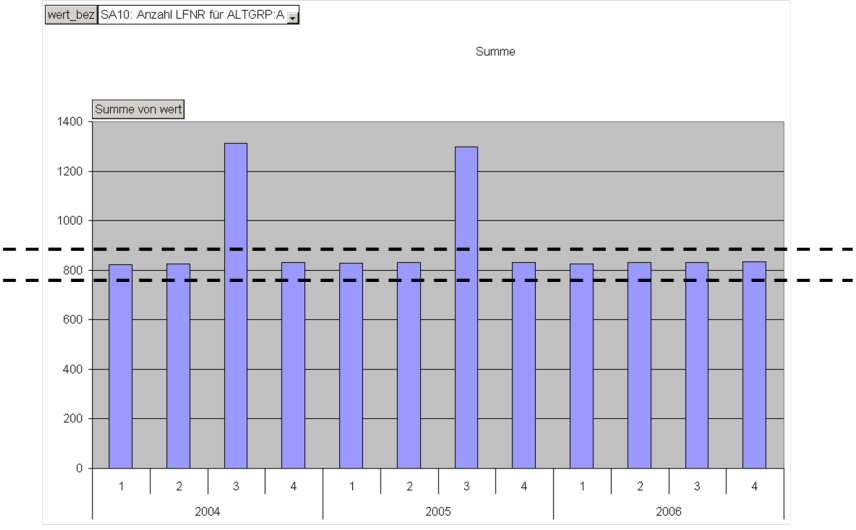

Abbildung 16: Beispiel für Ausreißer (2/2)

Die erkannten Abweichungen werden dem Benutzer mitgeteilt, sodass eine genauere Überprüfung und gegebenenfalls eine Änderung bzw. Nachbesserung der Werte vorgenommen werden kann. Diese Mitteilung erfolgt mittels eines Portals, auf das die Mitarbeiter Zugriffsrechte besitzen. In diesem Portal können die Abweichungen graphisch präsentiert werden, sodass eine Abarbeitung erleichtert wird. Zurzeit wird die Überprüfung der Daten nur durch eine manuelle Durchführung von Prüfabfragen, welche in SQL ([Dat94], [Mel93])[3] bzw. SAS-Code vorliegen, vorgenommen. Die manuelle Durchführung der Überprüfung bedeutet einen hohen zeitlichen und administrativen Aufwand.

5.3.3 FOKO - Beispieldaten

Für eine bessere Vergleichbarkeit der Werkzeuge wird folgender Auszug aus den von der OÖGKK zur Verfügung gestellten Beispieldaten verwendet:

nr	zeitraum	sa_key	ALT GRP	VNUMV	BETR	TART	BE FART	leidat	vtr	abrvtr	TRDAT	gesl	mwst bet
1	15979	1592981	2	2897	28,3	01	1	15996	14	14	18102003	W	2,83
2	15979	1641521	0	2897	28,3	07	1	16000	14	14	22102003	w	5,83
3	15979	1592989	2	2377	59,3	02	2	16033	14	14	24112003	M	5,93
4	15979	1592995	7	1212	59,3	00	2	15992	14	14	14102003	W	5,93
5	15979	1593032	5	1353	59,3	00	2	16009	14	14	31102003	W	5,93
6	15979	1593023	5	1353	59,3	00	2	16009	14	14	31102003	W	5,93
7	15979	1593066	5	1497	59,3	02	2	16028	14	14	19112003	M	5,33
8	15979	1593088	4	1700	59,3	00	2	15984	14	14	06102003	W	5,93
9	15979	1593100	5	3376	59,3	02	2	16017	14	14	08112003	M	5,93
10	15979	1678822	6	1600	59,3	02	2	16035	14	14	26112003	W	5,93
11	15979	1593122	6	9850	59,3	00	2	16022	14	14	13112003	W	5,93
12	15979	1593129	9	5450	59,3	00	2	15984	14	14	06102003	W	5,93
13	15979	1593143	8	4570	59,3	02	2	16049	14	14	10122003	W	5,93
14	15979	1593184	6	6200	59,3	00	2	16005	14	14	27102003	W	5,93
15	15979	1718047	5	2035	59,3	00	2	16018	14	14	09112003	W	5,93
16	15979	1653111	2	2640	59,3	00	2	15989	14	14	11102003	W	5,93
17	15949	1593241	0	2092	59,3	00	2	16024	14	14	15112003	W	5,93
18	15979	1593261	3	3713	59,3	00	2	16011	14	14	02112003	M	5,93
19	15979	1593272	2	3045	59,3	02	2	16054	14	15	15122003	M	5,93
20	15979	1593362	6	1257	59,3	00	2	16027	14	l4	18112003	M	5,93

Tabelle 16: Auszug aus den Beispieldaten

[3]SQL Standard - SQL:1999

Die genaue Bezeichnung der Spaltenköpfe lautet wie folgt:

Spaltenbezeichnung	Beschreibung
ALTGRP	Altersgruppe
VNUMV	Versicherungsnummer Versicherter
BETR	Kosten (Betrag ohne Mwst))
TART	Transportart
BEFART	Art der Beförderung
leidat	Leistungsdatum
vtr	Versicherungsträger
abrvtr	Abrechnender Versicherungsträger
TRDAT	Transportdatum)
gesl	Patienten Geschlecht
mwstbet	Mehrwertsteuerbetrag

Tabelle 17: Spaltenbezeichnung der Beispieldaten

In den Beispieldaten (Tabelle 16) sind folgende Fehler enthalten:

Fehlernummer	Datensatz	Spalte	Fehlerausprägung	richtiger Wert
1	2	gesl	w (Kleinschreibung)	W
2	2	mwstbet	5,83 (Falscher Wert)	2,83
3	5, 6	-	Duplikate	Datensatz 5
4	17	zeitraum	15949 (Falscher Zeitraum)	15979
5	19	abrvtr	15 (15 statt 14)	14
6	20	abrvtr	l4 („l“ statt 1)	14

Tabelle 18: Beispieldaten - Enthaltene Fehler

5.4 Anforderungen

Die OÖGKK betreibt ein SAS® Data Warehouse, in welches in unregelmäßigen Abständen auf Grund von Anfragen eines autorisierten Mitarbeiters Daten importiert werden. Dies kann z.B. eine Datenintegration von 10 GB monatlich bedeuten. Zur Zeit finden manuelle Kontrollen mit Hilfe von Prüfungsabfragen, welche in SQL oder auch SAS-Code vorliegen, statt, um Fehler in den Daten herauszufiltern und im Anschluss zu bearbeiten. Diese manuelle Kontrolle gestaltet sich zeit- und kostenintensiv. Auf Grund dieser Problematik ist es notwendig eine automatisierte Kontrolle vorzunehmen, welche die vorliegenden Daten anhand von festgelegten Kriterien (z.B Durchschnittswerte, Toleranzgrenzen) untersucht, Fehlerkorrekturen aufzeigt sowie eine übersichtliche Darstellung vornimmt, sodass die anschließende Fehlerbehandlung effizienter durchgeführt werden kann.

Zu diesem Zweck sind die vorhandenen Daten zu analysieren, um mögliche Fehlerquellen zu identifizieren. Hierzu ermittelt man im Vorfeld historische statistische Verteilungen und analysiert die vorliegenden Attributwerte auf ihr Vorkommen, damit Referenz- und Grenzwerte zur Datenüberprüfung zur Verfügung stehen.

Von den in Kapitel 4.2 vorgestellten Fehlerarten ist im Wesentlichen die Behandlung folgender drei Fehlerarten im Rahmen dieser Diplomarbeit zweckmäßig, weil diese nicht oder nur erschwert von den zur Zeit der OÖGKK zur Verfügung stehenden Überprüfungsmethoden gefunden werden. Diese stehen auch in Bezug zu den in Kapitel 5.3.2 identifizierten Fehlerquellen:

- Duplikate - Duplicates (siehe Kapitel 4.2.5)
- Falscher Datensatz - Invalid Tuple (siehe Kapitel 4.2.6)
- Fehlende Tupel - Missing Tuple (siehe Kapitel 4.2.8)

Im Vorfeld werden seitens der OÖGKK Abfragen getätigt, welche die Einhaltung von Integritätsbedingungen sowie Formatvorschriften sicherstellen. Das zur Verfügung gestellte Überprüfungstool überprüft im Anschluss an den Ladeprozess die eingelesenen Daten. In Kapitel 6 werden fünf verschiedene Werkzeuge auf ihre Funktionsweise in Bezug auf den Umgang mit der Fehlerfindung und anschließender Fehlerbehandlung von vorgegebenen Daten untersucht.

In Kapitel 7 werden die identifizierten Methoden je ausgewähltem Werkzeug (siehe Kapitel 6) den bearbeitenden Fehlerquellen zugeordnet, sodass herausgefiltert wird, welches Werkzeug welche Fehlerquellen bearbeiten kann. Wird ein passendes Werkzeug gefunden, das die soeben beschriebenen Fehlerquellen in ausreichendem Maße bearbeiten kann, muss eine Einbindung in das bestehende System überlegt werden. Gibt es kein passendes bzw. kein ausreichend qualifiziertes Werkzeug, so muss in weiterer Folge ein Framework konzipiert werden, in dem deklarative Regeln zur Kontrolle der Daten und die Methoden für die Behebung von Fehlern vorhanden sind.

Abbildung 17 zeigt den Aufbau der geforderten Lösung seitens der OÖGKK.

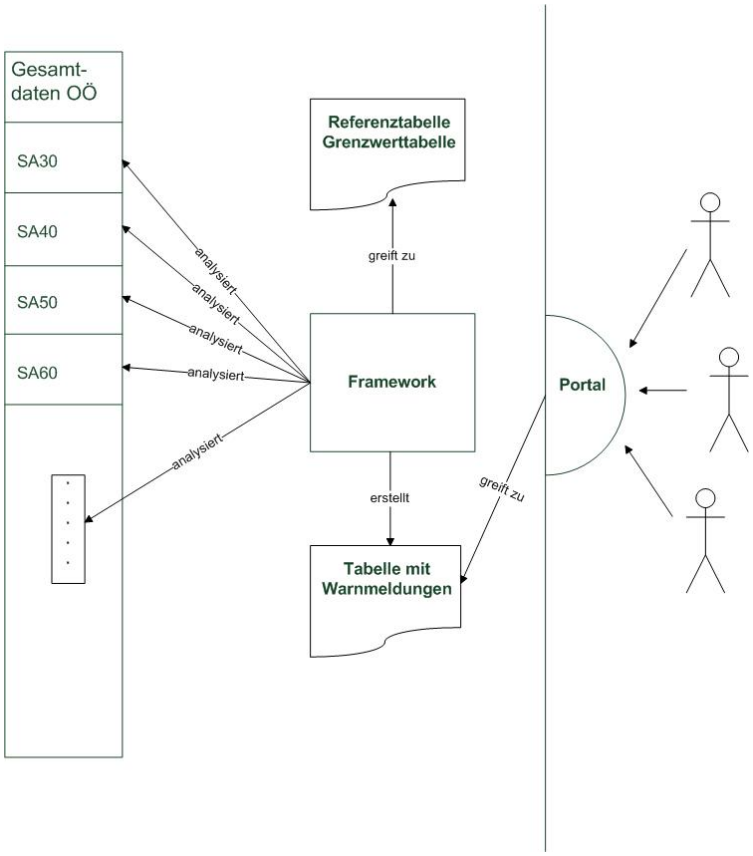

Abbildung 17: Ablauf: Datenimport und Datenüberprüfung

Durch vorher festgelegte Referenzwerte, die am Server abgespeichert sind, wird eine lückenlose Überprüfung der Daten gewährleistet. Bei diesen Werten handelt es sich um historisch ermittelte Ober- und Untergrenzen sowie eingetragene Referenzwerte einzelner Attribute, die von der OÖGKK ermittelt und eingetragen werden. In Abfragen werden durch Zuhilfenahme dieser festgesetzten Werte die Daten überprüft und potentielle Fehler kennzeichnet.

In einem weiteren Schritt werden alle erkannten Warnungsmeldungen gesammelt und in einer Ereignistabelle abgespeichert. Um eine Bearbeitung zu erleichtern bzw. zutreffende Warnungsmeldungen von falschen unterscheiden zu können, werden alle Warnungen im DWH-Portal visualisiert. Dabei stellen diese Warnungen die Abweichung der Prüfungsergebnisse zu den in der Referenztabelle gespeicherten historischen Prüfergebnissen dar. In weiterer Folge können die Anwender auf die Warnungsmeldungen reagieren und gegebenenfalls Änderungen an den Datenbeständen durchführen.

Ein ebenfalls wichtiges Kriterium stellt die Erweiterbarkeit des Prüfprogrammes dar. Es muss gewährleistet werden, dass das Werkzeug auf einfache Weise um

zusätzliche Prüfungen erweitert werden kann. Idealerweise ist dies in Form eines SAS-Codes (Datei) durchführbar, um die nachträgliche Entwicklung und somit auch die Einbindung in die Software zu erleichtern.

Zusammenfassend muss ein passendes Werkzeug für die OÖGKK mindestens folgende Anforderungen erfüllen:

- Schnittstellen für
  - SAS-Tabellen (siehe Kapitel 5.3.1).
  - Generierung und Abspeicherung von Warnungsmeldungen.
  - Mitprotokollierung der Arbeitsschritte.
- Identifizierung und Bearbeitung folgender Fehlerquellen (siehe Kapitel 5.3.2 und 5.3.3):
  - Duplikate - Duplicates.
  - Falscher Datensatz - Invalid Tuple.
  - Fehlende Tupel - Missing Tuple.
- Durchführung von vorgefertigten Abfragen.
- Einfache Erweiterbarkeit des Prüfprogrammes.
- Arbeiten mit Referenz- und Grenzwerten.

6 Ausgewählte Methoden für die Datenbereinigung anhand von bestehenden Lösungen

In diesem Kapitel werden fünf bestehende Lösungen zur Datenbereinigung näher beschrieben und die darin vorkommenden Methoden zur Fehlerbehebung vorgestellt. Es handelt sich hierbei um:

- Microsoft®SQL Server™2005 Integration Services (SSIS) [Mic08]
- Oracle®Warehouse Builder 10g Release 2 (10.2.0.2) [Ora08]
- SAS®9.1.2 Data Quality Server [SAS08b]
- WinPure ListCleaner Pro [Win08b]
- WizRule®(Demo) [Wiz08a]

Die Auswahl der ersten drei genannten Softwarepakete wurde auf Grund der vorherrschenden Marktstellung und des Bekanntheitsgrades durchgeführt[4]. Bei den beiden Letzteren handelt es sich um Softwarelösungen, die in eingeschränktem Maße frei zugänglich sind und zwei unterschiedliche Ansätze repräsentieren, wie an die Problemstellung herangegangen werden kann. Zum einen wird mittels WinPure ListCleaner Pro ein Werkzeug zur Verfügung gestellt, das sich vor allem auf die Bearbeitung der Attributwerte spezialisiert, und zum anderen stellt WizRule® ein Werkzeug dar, das zur Auffindung von Regeln (Wenn-Dann-Beziehungen) vorgesehen ist.

Ziel dieses Kapitels ist die Durchsprache der verwendeten Methoden, soweit diese aus den Dokumentationen und der zusätzlich auf den Hompages der Firmen zur Verfügung stehenden Unterlagen ersichtlich sind. Es wird dadurch eine Grundlage für eine Gegenüberstellung der beschriebenen Funktionen geschaffen, anhand dieser in weiterer Folge eine zweckmäßige Lösung vorgeschlagen bzw. ausgearbeitet wird.

6.1 Grundgerüst für den späteren Vergleich

In Kapitel 5.4 werden drei Fehlerquellen präsentiert, deren Behandlung von der OÖGKK gefordert wird. Es sind dies:

[4] vgl. Marktdaten von www.olapreport.com [NP08]

- Duplikate - Duplicates (Fehlerquelle 1)
  - Problem von Mehrfach-Datensätzen (Duplikate). Konsequenz: Es werden zu viele Datensätze auf Grund der Analyse angezeigt.
- Falscher Datensatz - Invalid Tuple (Fehlerquelle 2)
  - Es werden Tupel gespeichert, denen kein Wert der realen Welt zu Grunde liegt.
- Fehlende Tupel - Missing Tuple (Fehlerquelle 3)
  - Tupel der Zieldateien werden vom ETL-Job nicht erzeugt, weil die dazu nötigen Daten nicht in den Quelldateien vorhanden sind. Konsequenz: Es werden zu wenig Datensätze auf Grund der Analyse angezeigt.
  - Tupel der Quelldateien werden vom ETL-Job nicht geladen, weil sie von diesem nicht als zu ladend erkannt werden.

Diese Fehlerquellen müssen auf jeden Fall von dem geforderten Werkzeug bearbeitet werden. Zur besseren Vergleichbarkeit wird eine vierte Fehlerquelle - sonstiger Fehler (z.B Anzeigen von falschen Werten / Tippfehler) - eingeführt.

Für eine zusätzliche Einteilung der Funktionen wird folgende Gruppierung vorgenommen:

- Methoden zum Analysieren von Werten (A)
- Methoden zum Verändern von Werten (V)
- Methoden zum Darstellen der Ergebnisse (D)
- Regeln für Werte (R)

Um eine bessere Vergleichbarkeit der Methoden herzustellen, wird in weiterer Folge in den Schlussfolgerungen der einzelnen Werkzeuge das vorgestellte Einteilungsschema, welches in Tabelle 19 noch einmal übersichtlich dargestellt wird, angewendet.

Gruppe		Fehlerquelle	
A	Methoden zum Analysieren von Werten	1	Duplicates
V	Methoden zum Verändern von Werten	2	Invalid Tuple
D	Methoden zum Darstellen der Ergebnisse	3	Missing Tuple
R	Regeln für Werte	4	sonstiger Fehler

Tabelle 19: Legende zur Einteilung der Methoden

6.2 Microsoft® SQL Server™ 2005 Integration Services (SSIS)

Zu Beginn dieses Kapitels werden die in Microsoft® SQL Server™ 2005 Integration Services (in Folge SSIS genannt) verwendeten Methoden und Vorgehensweisen vorgestellt, welche sich mit den Problemen der Datenbereinigung und der Datenzusammenführung beschäftigen. Die zu Grunde liegenden Daten wurden aus der Online-Dokumentation des SSIS entnommen [Mic08]. Hierbei findet sich eine ähnliche Vorgehensweise zur Bereinigung von unsaubereren Daten wie jene des Data Cleaning Prozesses, der in Kapitel 3.3 bereits vorgestellt wurde.

Mit den SSIS werden im Wesentlichen folgende drei Teilbereiche abgedeckt:

- Profiling - Es werden erste grobe Untersuchungen an den Daten durchgeführt, um festzustellen, ob die Daten den festgelegten Anforderungen (Qualitätsmerkmalen, Integritätsbedingungen,...) entsprechen, sodass sie vom System bearbeitet werden können. Dadurch wird vermieden, dass das System durch vorhersehbare Fehler gestoppt wird.
- Cleaning - Nachdem die Daten den „Profiling-Standards“ entsprechen, müssen weiterhin Datenbereinigungsmethoden angewandt werden, um die Richtigkeit und Vollständigkeit der Daten gewährleisten zu können.
- Auditing - Mit dem Auditing werden die vorgenommen Operationen überwacht und mitprotokolliert. Dadurch kann die Qualität der Daten überwacht sowie festgestellt werden, in wie weit die getroffenen Maßnahmen den erwarteten Ergebnissen entsprechen.

In den folgenden drei Unterkapiteln wird die Vorgehensweise der Datenbereinigung des SSIS näher vorgestellt.

6.2.1 SSIS - Profiling

Das Profiling stellt sicher, dass die neuen Daten den vorhandenen Daten entsprechen. Dies passiert vor allem auf Schema-Ebene (siehe Kapitel 4.1). Es wird dabei ermittelt, ob es sinnvoll ist mit der Datenintegration überhaupt zu beginnen, oder ob die Daten im Vorhinein noch einmal bearbeitet werden müssen.

Beim Profiling spezifiziert der Anwender zuerst einige Metriken und Bedingungen, die erfüllt werden müssen, damit die Datenintegration gestartet werden kann. Hierbei liegt das Hauptaugenmerk auf der Qualität der gesamten Daten und nicht auf der eines einzigen Datensatzes. Insbesonders werden Fehler beachtet, welche das System während des Integrationsvorganges stoppen und somit einen Neuanfang verursachen würden.

Beispiele solcher Qualitätsindikatoren sind:

- Anzahl der Datensätze - Überprüfung der eingelesen Daten auf die Anzahl der tatsächlichen Datensätze. Sind überhaupt Datensätze bzw. die erwartete Anzahl von Datensätzen eingelesen worden?
- Prozentsatz von fehlenden Attributen - Datensätze besitzen oft Attribute, welche keine Werte enthalten. Sollte aber ein vergleichsweise hoher Prozentanteil an fehlenden Werten (NULL, NA, UNKNOWN,...) vorliegen, ist dies ein Signal für nicht aussagekräftige Daten. Dies weist auf einen Fehler im Ladevorgang bzw. in den Urspungsdaten hin.
- Prozentsatz der Integritätsverletzungen - Abhängig von der SQL-Server DB können Integritätsverletzungen als nicht relevant angesehen und somit verarbeitet werden (in Folge ein Fall für Datenbereinigung). Ebenso können diese Verletzungen von der DB von vornherein nicht zugelassen werden (Unique-Bedingung).
- Indikator für fehlende Dateistruktur - Liegt die Datei im angegebenen Verzeichnis? Dies ist vor allem wichtig, wenn mehrere unterschiedliche von einander abhängige Dateien eingelesen werden.
- Kennzeichnung von verdächtigen Datenwerten - Oftmals beinhalten Daten Fehler, die keine Bedingungen verletzen. Als Beispiel sei hier ein Kind mit Doktortitel genannt. Es besteht hier keine Verletzung einer Integritätsbedingung sowie eines NULL-Wertes. Dennoch kann hierbei auf einen Fehler geschlossen werden. Durch Miteinbeziehung von Data-Mining-Komponenten in SSIS können Vergleiche mit Referenzwerten durchgeführt und somit verdächtige Werte gekennzeichnet werden.

In einem weiteren Schritt werden a priori Grenzwerte für diese Indikatoren mitgeben, sodass feststellbar ist, ob eine Verletzung vorliegt oder nicht. Voraussetzung dafür ist eine sorgfältige Auswahl dieser Grenzwerte. In einem letzten Schritt werden Maßnahmen gesetzt, mit denen auf Fehler reagiert wird:

- Versenden eines E-mails mit einer genauen Beschreibung der Fehlerursache, um somit den Fehler beheben zu können.
- Verändern der Grenzwerte.
- Falls ein Problem besteht aber der Grenzwert gerade nicht überschritten wird, ist es nötig, die Situation länger zu beobachten. Anhand dieser Analysen können anschließend entsprechende Maßnahmen wie z.B. eine Anpassung der Grenzwerte ergriffen werden, welche in weiterer Folge ernstere Fehler verhindern.

6.2.2 SSIS - Cleaning

Der nächste Schritt, der mit Hilfe des SSIS vorgenommen wird, ist die Datenbereinigung. Sobald ein Fehler gefunden wird, gibt es drei Möglichkeiten diesem Fehler zu begegnen.

1. Behebung des Fehlers auf Grund einer vorgegebenen Geschäftslogik
2. Löschen des betreffenden Datensatzes und Fortsetzung der Verarbeitung
3. Beenden der Verarbeitung

Diese Möglichkeiten bieten eine einfache Auswahl, dennoch ist es wichtig die richtige Entscheidung für das jeweilige Problem zu treffen. Bei schwerwiegenden Fehlern wird die Verarbeitung abgebrochen. Dennoch ist es sinnvoll, den Fehler bestmöglich durch die bestehenden Geschäftslogiken zu beheben und im Anschluss die Daten fertig abzuarbeiten. Mögliche Fehlerursachen finden sich in Kapitel 4.2. Als wichtigste Probleme erscheinen Duplikate (siehe Kapitel 4.2.5), Fehlende Werte (siehe Kapitel 4.2.7) und Formatfehler (siehe Kapitel 4.2.2). Weiters kann es auch nötig sein keine Aktion durchzuführen. Es ist nicht immer vorteilhaft den entsprechenden Datensatz gleich zu löschen, da somit alle Daten, eventuell auch zum Einfügen relevante, verloren gehen. Hier bietet SSIS die Möglichkeit diese in einer Fehlertabelle abzuspeichern.

Eine zusätzliche Alternative ist den Fehler einfach zu übergehen. Dies ist dann sinnvoll, wenn der Fehler als zu unbedeutend eingeschätzt wird. Dabei sollten allerdings die Fehler gekennzeichnet bzw. mit Hilfe des Auditings (siehe Kapitel 6.2.3) genauer untersucht werden. Damit können in weiterer Folge Entscheidungen getroffen werden, wie in Zukunft mit einem gleichartigen Fehler zu verfahren ist. Für die Behandlung von hereinkommenden Daten und zur Durchführung der Datenbereinigung stellt SSIS im Wesentlichen drei Methoden bereit. Es sind dies:

- Reassigning column values - Mit Hilfe einer Suchfunktion werden eintreffende Daten einer verifizierten Datenquelle gegenübergestellt (siehe Kapitel 6.2.2.1), damit NULL-Werte, fehlende oder falsche Werte entdeckt werden. Zusätzlich ist es SSIS mit Hilfe der Transformation für abgeleitete Spalten möglich neue Spaltenwerte zuzuordnen.
- Handling data duplicates - Zur Erkennung von Duplikaten stellt SSIS drei Methoden zur Verfügung: zum einen die Transformation für Fuzzygruppierung und zum anderen die Transformationen für Suche bzw. Fuzzysuche. Da sich die Transformationen für Suche bzw. Fuzzysuche sehr ähneln wird in dieser Arbeit nur die Transformation für Fuzzsuche behandelt. Die Transformationen stellen den wichtigsten Teil der Datenbereinigung dar, deshalb werden sie in den Kapiteln 6.2.2.1 und 6.2.2.2 näher beschrieben.

- Extracting data - Sind mehrere Werte in einer Spalte untergebracht, stellt SSIS Funktionen zur Manipulation von Zeichenketten zur Verfügung, mit denen einzelne Datenwerte aus Zeichenketten extrahiert werden. Als Beispiel dienen hier Ort und Postleitzahl (PLZ). Werden beide in einem Feld gespeichert, aber in zwei Feldern benötigt, so wird mittels der Transformation für abgeleitete Spalten (siehe Kapitel 6.2.2.3) eine Aufspaltung dieses Feldes durchgeführt.

Weitere Informationen finden sich in der Onlinedokumentation[5]. Zudem werden die drei wichtigsten Funktionen des SSIS in Zusammenhang mit der Datenbereinigung in den folgenden Unterkapiteln kurz zusammengefasst.

6.2.2.1 Transformation für Fuzzysuche

Hauptaufgaben der Transformation für Fuzzysuche sind das Korrigieren von Daten, das Standardisieren von Werten und das Bereitstellen fehlender Werte. Unter dem Begriff Fuzzy-Logik versteht man eine Erweiterung der klassischen zweiwertigen Booleschen Logik. Es werden nicht nur die Zustände wahr (true, 1) und falsch (false, 0) akzeptiert, sondern die Fuzzy-Logik arbeitet auch mit Zwischenabstufungen. Dadurch ist die Fuzzy-Logik besonders zur Behandlung von nichtpräzisen Daten und Problemstellungen geeignet, zu denen es mehr als nur eine Lösung gibt. Dabei gilt: Je näher der Wert an 1 liegt, desto wahrscheinlicher ist die Richtigkeit der Aussage [Zad65].

Im Gegensatz zur Transformation für Suche, welche eine Gleichheitsverknüpfung zum Herausfiltern von übereinstimmenden Datensätzen in der Verweistabelle anwendet, kommt bei der Transformation für Fuzzygruppierung die Fuzzyübereinstimmung zur Anwendung. Diese versucht eine nahe Übereinstimmung zu den Werten der Verweistabelle herzustellen. In einem ersten Schritt wird die Transformation für Suche angewendet. Findet dies wider Erwarten keine direkte Übereinstimmung, wird mittels der Transformation für Fuzzysuche eine annähernde Übereinstimmung gesucht.

Die Transformation für Fuzzysuche wird mit Hilfe dreier Funktionen angepasst:

- Maximale Suche nach Übereinstimmungen pro Eingabezeile
- Suche nach kleineren Einheiten mittels Token-Trennzeichen
- Schwellenwerte für Ähnlichkeit

Die maximale Suche nach Übereinstimmungen pro Eingabezeile gibt Null oder mehr Übereinstimmungen zurück. Es wird jedoch die maximale Anzahl, die eingegeben wurde nicht überschritten bzw. dargestellt. Dies bedeutet: sind mehr Übereinstimmungen als der Maximalwert vorhanden, werden diese nicht angezeigt.

[5] vgl. http://msdn.microsoft.com/de-de/library/ms141026.aspx, letzter Zugriff am 18.06.2008

Um die Überprüfbarkeit zu erleichtern, werden die Daten in Token zerlegt. Die Transformation für Fuzzysuche stellt hierzu eine Vielzahl von Standardtrennzeichen zur Verfügung. Mit Hilfe dieser Zerlegung werden kleinere Einheiten der Daten überzeugt. Durch eine Vielzahl von Token ist es leichter Gemeinsamkeiten festzustellen bzw. Attribute zu identifizieren, die einen ähnlichen Inhalt haben.

Der Schwellenwert der Ähnlichkeit stellt eine Dezimalzahl zwischen 0 und 1 dar. Dieser Schwellenwert kommt zum Einsatz, wenn die Transformation für Fuzzysuche eine Fuzzyüberstimmung zwischen den Spalten/Attributen der Eingabe- und Verweistabelle durchführt. Je näher ein Wert bei 1 liegt, desto mehr wird eine Ähnlichkeit zu Grunde gelegt. Um ein Maß an Mindestähnlichkeit vorauszusetzen wird ein Minimalwert (MinSimilarity) festgelegt. Der Schwellenwert der Ähnlichkeit muss über diesem Wert liegen, sodass eine geforderte Ähnlichkeit zu Grunde liegt, und der Wert ausgegeben werden kann. Bei ausgewiesenen Übereinstimmungen werden auch noch ein Ähnlichkeits- und Vertrauensergebnis berechnet. Das Ähnlichkeitsergebnis beschreibt die strukturelle Ähnlichkeit zwischen dem Eingabedatensatz und dem Datensatz, der als Vergleichsdatensatz diente. Das Vertrauensergebnis gibt die Wahrscheinlichkeit an, mit der ein bestimmter Wert die beste Übereinstimmung unter all den gefundenen Übereinstimmungen darstellt.

Die Ausgabespalten der Transformation enthalten zusätzlich zu den Eingabespalten die ausgewählten Spalten der Suchtabelle sowie zwei zusätzliche Spalten, welche die zuvor beschriebenen Werte für Ähnlichkeit (_Similarity) und Qualität der Übereinstimmung (_Confidence) beinhalten.

6.2.2.2 Transformation für Fuzzygruppierung

Die Transformation für Fuzzygruppierung stellt eine Funktion zum Identifizieren von Duplikaten dar. Sie erkennt Datenzeilen, bei denen es sich wahrscheinlich um Duplikate handelt und führt diese dann in eine repräsentative Datenzeile zusammen. Hierbei wird eine Unterscheidung vorgenommen, ob nur nach genauen Übereinstimmungen gesucht wird, oder ob auch Zeilen gruppiert werden, die annähernd dieselben Werte enthalten. Es wird vom Benutzer ein Ähnlichkeitsgrad definiert. Um einen Ähnlichkeitsgrad wiedergeben zu können, stellt die Transformationsausgabe eine Spalte zur Verfügung, die das Ähnlichkeitsergebnis enthält. Dieses kann als Dezimalwert zwischen 0 und 1 liegen, wobei der Wert 1 eine absolute Übereinstimmung darstellt. Je näher das Ergebnis bei 1 liegt, desto genauer stimmen die Werte überein.

Um die bei der Transformation durchgeführte Gruppierung anpassen zu können, werden zwei Funktionen zur Verfügung gestellt, zum einen das Token-Trennzeichen und zum anderen der Schwellenwert der Ähnlichkeit. Die Zerlegung der Daten in Token erleichtert das Finden von Duplikaten. Hierbei stellt die Transformati-

on für Fuzzygruppierung eine Vielzahl von möglichen Trennzeichen in Token zur Verfügung.

Mit Hilfe des Schwellenwertes der Ähnlichkeit wird die Genauigkeit festgelegt, mit der nach Ähnlichkeiten der Token gesucht wird. Dies erfolgt sowohl auf Attributebene als auch auf Spaltenebene. Hier erfolgt ebenfalls eine Ergebniszuteilung zwischen 0 und 1. Ein Wert, der näher bei 1 liegt, stellt einen höheren Grad der Ähnlichkeit dar.

Die Transformation für Fuzzygruppierung berechnet die Ähnlichkeiten über Werte bzw. Spalten und führt diese zu einem Ergebniswert zusammen. Liegt der Wert unter dem des eingegebenen Schwellenwertes, so wird das entsprechende Attribut oder die entsprechende Spalte nicht zur Gruppierung hinzugefügt und als nicht ähnlich behandelt.

6.2.2.3 Transformation für abgeleitete Spalten

Mit Hilfe der Transformation für abgeleitete Spalten werden neue Spaltenwerte erzeugt. Diese können entweder in vorhandene Spalten dazu oder als Ersatzwert eingetragen werden. Zusätzlich ist es möglich auch neue Spalten damit zu füllen. Diese neuen Spaltenwerte werden durch die Anwendung von Ausdrücken auf die Transformationseingabespalten erstellt. Ein solcher Ausdruck kann eine Kombination von Variablen, Funktionen, Operatoren und Spalten enthalten. Die Variablen, Funktionen, Operatoren und Spalten können in beliebig vielen Ausdrücken verwendet werden. Durch die Definition von Ausdrücken werden folgende Aufgaben erfüllt:

- Verketten von Werten aus verschiedenen Spalten in einer neuen Spalte.
- Extrahieren von Zeichen aus einzelnen Zeichenketten.
- Anwenden von mathematischen Funktionen auf einzelne Spalten sowie speichern der Ergebnisse in abgeleiteten Spalten.
- Vergleichen von Spalten und Variablen und eventuell Ausgabe des „richtigen“ Wertes in einer neuen abgeleiteten Spalte.
- Extrahieren von Werten eines Datum-Wertes („Datetime“), z.B. Herausfiltern einer Jahreszahl.

Zusätzlich bietet SSIS Funktionen an, die die Durchführung der Datenbereinigung erleichtern:

- Transformation für Datenkonvertierung - Diese Transformation konvertiert den Datentyp einer Spalte in einen anderen Datentyp.

- Transformation für das Kopieren von Spalten - Diese Transformation fügt der Transformationsausgabe Kopien von Eingabespalten hinzu.
- Transformation zum Sortieren - Diese Transformation sortiert Daten.

Durch diese Funktionen ist es möglich die Daten in einer Art und Weise zu bearbeiten, sodass sie für eine Weiterverarbeitung in geeeigneter Form vorliegen.

6.2.3 SSIS - Auditing

Mit Hilfe des Auditings wird aufgezeigt, dass die Datenintegration erfolgreich abgeschlossen wurde. Durch das Auditing werden alle möglichen Operationen erfasst (einfügen, ändern, löschen), sodass sichergestellt ist, dass alle festgelegten Aufgaben auch tatsächlich durchgeführt worden sind. Dies wird vor allem durch Statistiken gewährleistet. SSIS stellt mehrere Tabellen zur Verfügung, um auf verschiedenen Ebenen (Instanz, Global) die aufgezeichneten Details einsehen zu können. Im Folgenden findet sich eine Auflistung von Tabellen, mit denen detailierte und aggregierte Übersichten erstellt werden.

- Audit Errors table - Speichert alle Fehler und Warnungen, welche einer sofortigen Bearbeitung bedürfen.
- Audit Detail table - Zeigt die genauen Details der Operationen auf (z.B. Anzahl der verarbeiteten Datensätze sowie der Einfüge-, Änderungs- und Löschoperationen).
- Audit Error Records table(s) - Speichert alle fehlerhaften Datensätze.

Durch die Auflistung der vorgenommen Aktionen in diese Tabellen wird eine Überwachung der Änderungen erleichtert. Somit wird die Transparenz gewährleistet und alle Aktionen und Änderungen sind entsprechend nachvollziehbar.

6.2.4 Microsoft® SQL Server™2005 Integration Services - Schlussfolgerungen

Die nachfolgende Tabelle 20 stellt die zur Verfügung gestellten Funktionen noch einmal übersichtlich dar.

Name	Beschreibung	Gruppe	Fehlerquelle
Qualitätsindikatoren	Anzahl der Datensätze	A	1, 2, 3
	Prozentsatz von fehlenden Attributen	A	4
	Prozentsatz von Integritätsverletzungen	A	4
	Indikator für fehlende Dateistruktur	A	4
	Kennzeichnung von verdächtigen Datenwerten	A	1, 2, 3, 4
	Führen von Grenzwerten zur Analyse	A	2, 4
Fuzzysuche	Überprüfung der Attribute anhand einer Referenztabelle	A	2
Fuzzygruppierung	Duplikatensuche	A	1
Reassigning column values		A, V	4
Transformation für abgeleitete Spalten		A, V	4
Transformation für Datenkonvertierung	Konvertiert den Datentyp einer Spalte in einen anderen Datentyp	V	1, 4
Transformation zum Sortieren	Sortiert Daten	A, D	4
Transformation für das Kopieren von Spalten	Fügt Kopien von Eingabespalten hinzu	A, V	4
Funktionen	Verketten von Werten aus verschieden Spalten in einer neuen Spalte	A, V	4
	Extrahieren von Zeichen aus einzelnen Zeichenketten	V	2, 4
	Anwenden von mathematischen Funktionen auf einzelne Spalten, speichern der Ergebnisse in abgeleiteten Spalten	A, V	4
	Vergleichen von Spalten und Variablen, Ausgabe des „richtigen“ Wertes in einer neuen abgeleiteten Spalte	A, V	1, 4
	Extrahieren von Werten eines Datum-Wertes	V	4
E-mail	Versenden einer E-mail mit Fehlerursache	A, D	

Tabelle 20: Methodeneinteilung SSIS

Microsoft®SQL Server™ 2005 Integration Services stellt drei grundlegende Funktionen zur Verfügung. Mit Hilfe des Profiling werden zunächst die Daten untersucht und mit Hilfe von Qualitätsindikatoren bewertet und überprüft. Hierbei werden verdächtige Werte gekennzeichnet und weitergeleitet.

Im nächsten Schritt wird mit Hilfe des Bereinigungsprozesses mit der Fehlerbehandlung begonnen. Dabei werden die vorliegenden Fehler auf Grund der vorgegeben Geschäftslogik bereinigt oder gelöscht bzw. kann auch die ganze Verarbeitung abgebrochen werden. Kernstücke des Bereinigungsprozesses stellen die Transformationen für Fuzzysuche, Fuzzygruppierung und abgeleitete Spalten dar. Mit Hilfe dieser drei Transformationen wird der Hauptanteil der Fehler behoben.

Den Abschluss bildet das Auditing. Dadurch wird sichergestellt, dass die Datenintegration vollständig abgeschlossen wurde. Dazu stellt das Auditing verschiedenste Tabellen zur Verfügung, mit deren Hilfe alle Aktionen überwacht und kontrolliert werden.

6.3 Oracle® Warehouse Builder 10g Release 2 (10.2.0.2)

Um Datenqualität beim Zusammenführen von mehreren Datenquellen sicherzustellen, orientiert sich Oracle® Warehouse Builder an vier Phasen:

- Qualitätsbewertung (Quality Assessment)
- Qualitätsdesign (Quality Design)
- Qualitätstransformationen (Quality Transformation)
- Qualitätsüberwachung (Quality Monitoring)

Diese vier Phasen werden in Abbildung 18 graphisch veranschaulicht. Hierbei konzentriert sich die Datenbereinigung bzw. das Auffinden von möglichen Fehlern, Inkonsistenzen und Mehrfachspeicherung auf drei wesentliche Teilbereiche dieser Phasen. Es liegt eine ähnliche Einteilung vor, wie sie schon in Kapitel 3.3 vorgenommen wird. Im Bereich des Data Profilings werden die Daten analysiert, um zu erkennen, wo sich Fehler eingeschlichen haben bzw. Inkonsistenzen vorherrschen. Dies bildet auch den Grundstock, um im späteren Data Monitoring die Daten zu überwachen.

Abbildung 18: Oracle - Phasen zur Sicherstellung von Datenqualität [Ora06, 10-2]

In einem weiteren Schritt werden mittels Data Rules Abhängigkeiten zwischen Datenobjekten und Attributen ermittelt bzw. in weiterer Folge auch festgesetzt, sodass eine Überprüfung anhand dieser Kriterien vollzogen wird. Als dritte Möglichkeit werden in der sogenannten Quality Transformation Phase eigene Funktionen ausgeführt, um Korrekturen vorzunehmen. Im Speziellen stellt Oracle hier zwei Operatoren zur Verfügung, zum einen den „Match-Merge Operator“ (siehe Kapitel 6.3.3.1) und zum anderen den „Name and Address Operator“ (siehe Kapitel 6.3.3.2).

Oracle® Warehouse Builder stellt zum Importieren und Exportieren folgende alternative Datenbanken und Dateiformate zur Verfügung:

Location Node in the Connection Explorer	Supported Sources	Supported Targets
Databases/Oracle	Oracle db 8.1, 9.0, 9.2, 10.1, 10.2	Oracle db 9.2, 10.1, 10.2
Databases/Non-Oracle	Any database accessible through Oracle Heterogeneous Services, including but not limited to DB2, DRDA, Informix, SQL Server, Sybase, and Teradata. Any data store accessible through the ODBC Data Source Administrator, including but not limited to Excel and MS Access.	To load data into spreadsheets or third-party databases, first deploy to a comma-delimited or XML format flat file.
Files	Delimited and fixed-length flat files.	Comma-delimited and XML format flat files.

Tabelle 21: Verwendete Dateiformate in Oracle Warehouse Builder 10.2. [Ora06, 5-2]

In den folgenden Abschnitten dieses Kapitels werden die einzelnen Möglichkeiten näher beschrieben, welche Oracle® Warehouse Builder zur Verfügung stellt, um eine bessere Datenqualität zu erreichen bzw. eine Datenbereinigung durchzuführen.

6.3.1 Oracle - Data Profiling

Das Data Profiling des Oracle® Warehouse Builder hilft, Fehler und Missstände in den Daten zu evaluieren und zu entdecken, noch bevor mit den Daten weiter gearbeitet wird, sie also in das System übernommen werden. Mit den in Kapitel 6.3.2 und 6.3.3 vorgestellten Regeln und Transformationen bietet der Oracle® Warehouse Builder die Möglichkeit automatisch Inkonsistenzen, Redundanzen und Ungenauigkeiten zu korrigieren [Ora06, 10-3ff].

Durch das Data Profiling, welches sich durch das genaue Analysieren der Daten auszeichnet, ist es möglich eine Vielzahl von Informationen über die Daten zu gewinnen, wie z.B.:

- Definition von zugelassenen Datenwerten.
- Beziehungen zwischen Spalten (Primärschlüssel, Abhängigkeiten,...).
- Anomalien und Ausreißer innerhalb von Spalten.
- Spalten, in denen nach Muster mögliche E-mail Adressen stehen.

Es können jene Datenformate analysiert werden, welche auch von Oracle erkannt und verarbeitet werden können (siehe Tabelle 21). Die gewonnenen Erkenntnisse werden tabellarisch oder graphisch analysiert. Dabei ist es möglich sich die Daten zu jedem Resultat anzusehen. In diesem Zusammenhang werden die entsprechenden Data Rules entweder automatisch oder manuell generiert. Der Oracle® Warehouse Builder ermöglicht es auch den Qualitätsindex six-sigma zu berechnen [Ora06, 10-9f]. Six-sigma entspricht dabei einem statistischen Qualitätsziel, welches im Falle des Oracle® Warehouse Builders für jede Spalte die Anzahl der Null-Werte (entspricht den Mängeln) zu der Gesamtanzahl von Zeilen ins Verhältnis setzt [Ora06, 20-22].

Typen von Data Profiling

Um die Daten zu analysieren, bietet das Data Profiling drei verschiedene Möglichkeiten an (siehe Abbildung 19). Diese richten sich vor allem nach der Art wie die Daten analysiert werden: innerhalb einer Spalte („Attribute Analysis“), in Abhängigkeiten von Spalten („Functional Dependency“) oder in Abhängigkeiten von Attributen/Spalten in verschiedenen Tabellen („Referential Analysis“). Durch selbst festgelegte Profilingprozesse, welche Data Rules benutzen, werden weitere Überprüfungsmöglichkeiten geschaffen.

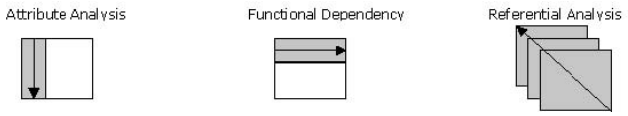

Abbildung 19: Oracle - Drei Typen des Data Profiling [Ora06, 10-4]

Abbildung 20 zeigt eine Aufschlüsselung der generell möglichen Analysen:

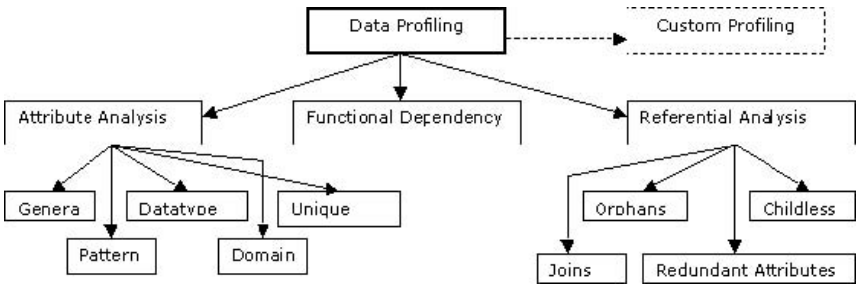

Abbildung 20: Oracle - Data Profiling / Attribut Analyse [Ora06, 10-4]

Die möglichen Analyseverfahren werden nachfolgend erläutert:

- Attribute Analysis - Mit Hilfe der Attributanalyse werden allgemeine und detailierte Informationen über die Struktur und den Inhalt einer Tabelle gewonnen, d.h. der in der Tabelle vorkommenden Spalten und Werte. Es werden dabei Informationen in folgenden Bereichen gefunden:

  - Pattern Analysis - Die Pattern Analysis erkennt Muster bzw. allgemeingültige Darstellungen durch eine Analyse der Attribute. Als erstes werden die Werte nach möglichen Mustern durchsucht. Danach werden die Werte mit den zuvor herausgefilterten Mustern identifiziert und in Beziehung gesetzt. Die dadurch errechneten Prozentsätze geben über die Gültigkeit der Muster Aufschluss. Im Anschluss daran können Regeln erstellt werden, die so erkannte Probleme beheben. Mögliche erkannte Muster sind z.B. der Aufbau von E-mail Adressen, Sozialversicherungsnummern, Telefonnummern,....
  - Domain Analysis - Diese gibt Aufschluss über mögliche Wertebereiche bzw. über Werte, die häufig vorkommen. Als Beispiel dient hier eine Spalte „Familienstand“. Nach Untersuchung dieser Spalte wird festgestellt, dass über 95% der darin vorkommenden Werte unter den folgenden zu finden sind: „alleinstehend“, „verheiratet“, „geschieden“ oder „verwitwet“. Bei genauerer Betrachtung der restlichen Menge finden sich jeweils falsch geschriebene Versionen dieser Werte. Somit können diese als einsetzbare Werte definiert werden und das Programm nimmt automatisch entsprechende Korrekturen vor. Dennoch sollte vorsichtig bei der Festlegung solcher Werte/Wertebereiche vorgegangen werden.
  - Data Type Analysis - Diese Analyse wertet die Datentypen der Werte genauer aus. Ziel ist das Herausfinden von Metriken wie z.B. Maximum, Minimum oder Länge, damit dadurch z.B. Unstimmigkeiten in der Speicherung herausgefunden werden. Sind in einem „VARCHAR“ Feld z.B. nur Ziffern vorhanden, ist es möglicherweise von Vorteil den Datentyp für eine effizientere Verarbeitung zu ändern. Dadurch ist es

möglich eine Regel zu definieren, mit der sichergestellt wird, dass alle Werte denselben Datentyp besitzen.

- Unique Key Analysis - Mit dieser Analyse wird festgestellt, ob ein Attribut einen „Unique Key“ darstellt oder nicht. Hierbei werden die prozentuellen Vorkommnisse der Werte zueinander in Beziehung gebracht. Stellt sich hier z.B. heraus, dass 95% aller Werte nur einmal vorkommen, müssen nur die weiteren 5% einer Untersuchung zugeführt werden. Dabei kann sich herausstellen, dass es sich z.B. um NULL-Werte oder Duplikate der vorangegangen Datensätze handelt. Somit ist es erforderlich eine Regel zu erstellen, die sicherstellt, dass alle eingetragenen Werte weder doppelt vorhanden sind noch NULL-Werte enthalten.

Oracle® Warehouse Builder stellt die wichtigsten Aggregationsfunktion (Minimum/Maximum Wert der Spalte, Anzahl der Werte, NULL-Werte,...) zur Verfügung. Diese Berechnung stützt sich auf den Datentyp. Da die Berechnung der Aggregationsfunktion vom Datentyp abhängig ist, können eventuell nicht alle Aggregationen berechnet werden, da bestimmte Berechnungen nicht zulässig sind (Durchschnitt, Standardabweichung,...) [Ora06, 20-16f].

- Functional Dependency - Diese Analyse ermittelt die Abhängigkeiten zwischen einzelnen Spalten. Hierbei wird herausgefunden, welche Attribute durch andere Werte errechnet bzw. abgeleitet werden können (Beispiel: Field A = Field B x Field C) [Ora06, 20-23f].

- Referential Analysis - Bei dieser Analyse werden Verbindungen zwischen mehreren Objekten (in verschiedenen Relationen) hergestellt. Es werden nach Möglichkeit speziellere Abhängigkeiten herausgefiltert. Hierbei finden die Ausdrücke parent und child zur Kennzeichnung der zu überprüfenden Objekte Verwendung. Es ergeben sich vier mögliche Ergebnisse:

  - Orphans - Die Werte werden nur in den child-Objekten gefunden
  - Childless - Die Werte werden nur in den parent-Objekten gefunden
  - Redundant Attributes - Die Werte existieren in beiden Tabellen
  - Joins

  Mit Hilfe der Ergebnisse dieser Analyse können Referenzregeln festgelegt / berechnet werden, die die Kardinalitäten zwischen den zu untersuchenden Tabellen genau festlegen [Ora06, 20-24f]. Diese Vorgehensweise findet sich auch im Konzept von Primär- und Fremdschlüsseln in der Literatur wieder [HS00].

6.3.2 Oracle - Data Rules

Data Rules beinhalten Regeln/Definitionen für gültige Datenwerte oder Beziehungen zwischen Attributen und Tabellen, welche im Oracle® Warehouse Builder gesetzt werden, um die Datenqualität zu erhöhen. Es ist möglich diese Data Rules im Zuge des Data Profiling generieren zu lassen oder selbst festzuschreiben. Data Rules finden eine breite Einsatzmöglichkeit, wie folgende Anwendungsgebiete aufzeigen [Ora06, 20-35ff]:

- Data Rules - Data Profiling
- Data Rules - Daten- und Schemabereinigung
- Data Rules - Data Auditing

Um eine bessere Übersicht über Data Rules zu erhalten, werden sie in in folgende Kategorien eingeteilt [Ora06, 20-35ff]:

- Domain List - Die Domain List Regel gibt eine Liste von Werten an, die ein Attribut annehmen kann. Als Beispiel: Die Werte, die in der Spalte „gesl“ (entspricht dem Geschlecht einer Person) zugelassen sind: M und W
- Domain Pattern List - Die Domain pattern list gibt ein Muster vor, nachdem ein Attributwert aufgebaut sein muss. Ein Muster kann wie unter Kapitel 4.2.2 schon beschrieben: $\{\sum D^*, \sum D^*\}$ sein, es muss jedoch die Konformität mit der von Oracle® Warehouse Builder zur Verfügung gestellten Syntax gewährleistet sein. Eine mögliche Syntax für den Aufbau einer Telefonnummer in Form eines regulären Ausdruckes [Fri03] mittels einer Oracle konformen Syntax lautet:

  (^[[:space:]]*[0-9]{ 3 }[[:punct:]|:space:]]?[0-9]{ 4 }[[:space:]]*$)

  Durch diese Syntax wird sichergestellt, dass die Telefonnummer folgenden Aufbau hat: 3 Ziffern, dann ein Leer- oder Punktuationszeichen, gefolgt von 4 Ziffern und abschließend folgen beliebig viele Leerzeichen.
- Domain Range - Die Domain range Regel gibt ein Intervall an, indem die Werte eines Attributes liegen dürfen. Als Beispiel müssen die Werte der Spalte „altdeka“ (entspricht den Altersdekaden einer Person) z.B. zwischen 0 und 9 liegen.
- Common Format - Diese Regel weist Attributen ein allgemein bekanntes Format zur Darstellung zu. Oracle® Warehouse Builder stellt hierbei mehrere Untertypen zur Formatzuweisung zur Verfügung, so z.B. Formatdefinitionen für die Telefonnummer, IP-Adresse, Sozialversicherungsnummer sowie für E-mail Adressen.

- No Nulls - Diese Regel legt fest, dass die Ausprägung eines Attributes keinen Null-Wert annehmen darf. So z.B. kann das Geburtsdatum eines Mitarbeiters keinen NULL-Wert annehmen.
- Functional Dependency - Diese Regel gibt an, dass es sich um normalisierte Daten handeln kann.
- Unique Key - Durch die Unique Key Regel wird festgelegt, dass die Werte eines Attributes unique also einzigartig sind. Z.B. existiert eine Sozialversicherungsnummer nur einmal. Wird diese ein zweites Mal eingetragen, so wird automatisch eine Fehlermeldung ausgegeben.
- Referential - Diese Regel gibt den Typ einer Beziehung (1:n) an, den ein Wert zu einem bestimmten Wert haben muss. (Z.B. sollte das Attribut Abteilungs_id einer Tabelle Abteilung eine 1:n Beziehung zu dem Attribut Abteilungs_id einer Mitarbeitertabelle haben.)
- Name and Address - Diese Regel benutzt die von Oracle® Warehouse Builder zur Verfügung gestellte Unterstützung zur Erkennung von Namen und Adressen, welche käuflich erworben werden kann (siehe Kapitel 6.3.3.2).
- Custom - Durch diese Regel wird ein SQL-Ausdruck festgelegt, welcher durch seine Eingabeparameter den zu untersuchenden Wert überprüft: Als Beispiel zur Validierung eines Eingabedatums dient folgende Regel VALID_DATE mit den zwei Input-Parametern, START_DATE und END_DATE. Ein somit gültiger Ausdruck für diese Regel ist:
  „THIS“.“END_DATE“ > „THIS“.“START_DATE“.

6.3.3 Oracle - Quality Transformation

Der Oracle® Warehouse Builder ermöglicht es, Korrekturfunktionen auf Grund der Profiling Ergebnisse zu erstellen. Diese können in weiterer Folge automatisch angewendet und durchgeführt werden. Bei dieser Transformation der Daten werden die in Folge vorgestellten Operatoren zu Hilfe genommen.

6.3.3.1 Match-Merge Operator

Durch den Match-Merge Operator ist es möglich aus einer Liste von Datensätzen jene heraus zu filtern, die sich auf dieselbe Entität beziehen, obwohl einige Attribute unterschiedliche Ausprägungen haben. Dies geschieht durch die Festlegung von Regeln, die bestimmen, welche Datenteile (= Attribute) übereinstimmen müssen und in weiterer Folge auch festlegen, welche Daten in den neuen Datensatz aufgenommen werden. Hierbei steht dem Oracle® Warehouse Builder bei der Evaluation der Regeln die OR-Logik zur Verfügung [Ora06, 21-1ff]. Sollten z.B. bei drei Datensätzen (A,B,C) auf Grund von zwei verschiedenen Regeln die Datensätze B

und C auf Grund von Regel 1 und A und B auf Grund von Regel 2 übereinstimmen, so würden auf Grund der vom Oracle® Warehouse Builder zur Verfügung gestellten OR-Logik alle drei Datensätze übereinstimmen [Ora06, 21-2f].

Durch die Benutzung dieser Regeln ist es möglich, dass Daten von den ausgewählten und somit mehrfach vorhandenen Datensätzen in den neuen Datensatz übernommen werden. Es muss kein „richtiger Datensatz“ ausgewählt werden, sondern es ist möglich, dass sich der neue Datensatz aus Attributwerten von verschiedenen Datensätzen zusammensetzt. In einem ersten Schritt werden den erstellten Regeln ein Name und die Ausführungspositionsnummer zugewiesen. In einem nächsten Schritt werden die Regeltypen zugewiesen. Oracle® Warehouse Builder stellt dazu folgende Typen von Übereinstimmungsregeln zur Verfügung (siehe Tabelle 22) [Ora06, 21-1ff]:

Übereinstimmungsregel	Beschreibung
All Match	Matches all the rows within the match bin.
None Match	Turns off matching. No rows match within the match bin.
Conditional	Matches rows based on an algorithm you select.
Weight	Matches rows based on scores that you assign to the attributes.
Person	Matches records based on people's names.
Firm	Matches records based on business names.
Address	Matches records based on postal addresses.
Custom	Create a custom comparison algorithm.

Tabelle 22: Übereinstimmungsregeln in Oracle® vgl. [Ora06, 21-9]

In einem weiteren Schritt ist es notwendig, jene Attributwerte festzulegen, welche in dem neuen Datensatz übernommen werden. Hierbei stehen folgende Möglichkeiten zur Auswahl um den geeignetsten Wert auszuwählen (siehe Tabelle 23):

Merge Rule	Description
Any	Uses the first non-blank value.
Match ID	Merges records that have already been output from another Match-Merge operator.
Rank	Uses the ranked values in a second attribute to select the preferred value.
Sequence	Uses the values in a sequence to generate unique keys as the data is loaded.
Min Max	Uses the first value based on the order of another attribute.
Copy	Uses the values from another merged attribute.
Custom	Uses the PL/SQL code that you provide as the criteria for merging records.
Any Record	Identical to the Any rule, except that an Any Record rule applies to multiple attributes.
Rank Record	Identical to the Rank rule, except that a Rank Record rule applies to multiple attributes
Min Max Record	Identical to the Min Max rule, except that a Min Max Record rule applies to multiple attributes.
Custom Record	Identical to the Custom rule, except that a Custom Record rule applies to multiple attributes.

Tabelle 23: Vereinigungsregeln in Oracle® vgl. [Ora06, 21-21]

Abschließend kann die Bereinigung durchgeführt werden, sodass alle vom System erkannten Mehrfachspeicherungen eliminiert werden.

6.3.3.2 Name and Address Operator in a Mapping

Mit Hilfe des Name and Address Operator in a Mapping ist es dem Oracle® Warehouse Builder möglich eine Bereinigung von Namen- und Adressdaten durchzuführen. Hierbei werden Fehler und Inkonsistenzen anhand von Datenbibliotheken erkannt und korrigiert. Diese Bibliotheken müssen von Drittanbietern zugekauft werden.

Der Oracle®Warehouse Builder stellt dazu zwei Optionen zur Verfügung. Zum einen kann ein neuer Adressoperator definiert werden. Um die Erstellung dieses Adressoperators zu erleichtern, wird vom Oracle® Warehouse Builder ein Softwa-

reassistentenprogramm zur Verfügung gestellt. Zum anderen kann mit Hilfe des Mapping-Editors ein existierender Adressoperator bearbeitet und editiert werden [Ora06, 10-11ff].

Die erkannten Fehler und Inkonsistenzen beziehen sich auf Variationen in den Adressformaten, Benützung von Abkürzungen, Rechtschreibfehler, veraltete Informationen und vertauschte Vor- und Nachnamen. Die Fehler werden vor allem durch folgende drei Maßnahmen erkannt[Ora06, 10-12]:

- Zerlegung der hereinkommenden Namen- und Adressdaten in einzelne Elemente (Token).
- Standardisierung der Namen- und Adressdaten durch standardisierte Versionen von Rufnamen und Firmennamen sowie der Benützung von Standardabkürzungen von Straßennamen, welche von der zuständigen Postgesellschaft des jeweiligen Landes empfohlen sind. Diese werden durch zugekaufte Bibliotheken definiert.
- Korrektur von Adressinformationen wie Straßennamen und Städtenamen, sodass keine inkorrekten und somit unzustellbaren Adressen eingetragen sind.

6.3.3.3 Oracle - Nützliche Transformationen

Zusätzlich zu den in Kapitel 6.3.3.1 und 6.3.3.2 vorgestellten Operatoren stellt der Oracle® Warehouse Builder eine Vielzahl von weiteren Transformationen zur Verfügung, welche zur weiteren Bearbeitung bzw. auch zu Vorarbeiten (z.B. Sicherstellen der Einheitlichkeit) herangezogen werden können. Durch diese Transformationen werden Änderungen an den Daten vorgenommen, sodass eine Bereinigung bzw. Anpassung durchgeführt werden kann. Im Einzelnen handelt es sich um nachstehende Transformation für Zeichen (siehe Tabelle 24), die für die Datenbereinigung im Rahmen dieser Arbeit interessant sind [Ora06, 27-11ff].

Bezeichnung	Beschreibung
CONCAT [Ora06, 27-11]	Fügt zwei Attribute in Einem zusammen, solange die Datentypen CHAR und VARCHAR vorliegen.
INITCAP [Ora06, 27-12]	Gibt den einzelnen Wörtern von einem Attribut einen großen Anfangsbuchstaben. Trennung erfolgt mittels Leerzeichen oder mit Nicht-alphanumerischen Zeichen.
LENGTH, LENGTH2, LENGTH4, LENGTHB, LENGTHC [Ora06, 27-13]	Ermittelt die Zeichenanzahl eines Attributes.
LOWER [Ora06, 27-14]	Gibt das Attribut in Kleinbuchstaben aus.
LTRIM [Ora06, 27-15]	Löscht von links beginnend alle definierten Zeichen bis die erste Ausnahme auftritt.
REPLACE [Ora06, 27-17]	Durch diese Transformationen werden ausgewählte Zeichen bzw. Zeichenketten durch ebensolche Festgelegte ausgetauscht.
RPAD [Ora06, 27-23]	Fügt am Ende des Attributes ein Zeichen/eine Zeichenkette so oft an, bis die neuen Zeichen eine gewisse Länge erreicht haben.
RTRIM [Ora06, 27-24]	Analog zu LTRIM, nur auf der rechten Seite des Attributes
SOUNDEX [Ora06, 27-24]	Gibt Attribute aus, die zwar verschieden geschrieben sind, aber phonetisch gleich klingen, (Bsp. Smith - Smyth). Nur in Englisch.
SUBSTR, SUBSTR2, SUBSTR4, SUBSTRB, SUBSTRC [Ora06, 27-15]	Gibt eine bestimmte Anzahl von Zeichen einer Zeichenkette aus.
TRIM [Ora06, 27-27]	Löscht alle vorangehenden und nachfolgenden Leerzeichen eines Attributwertes.
UPPER [Ora06, 27-27]	Gibt das Attribut in Großbuchstaben aus.

Tabelle 24: Oracle - Zeichentransformationen

6.3.4 Oracle® Warehouse Builder - Schlussfolgerungen

Nachfolgende Tabelle 25 stellt die zur Verfügung gestellten Funktionen des Oracle® Warehouse Builder noch einmal in tabellarischer Form dar.

Name	Beschreibung	Gruppe	Fehlerquelle
Pattern Analysis	Mustererkennung	A	4
Domain Analysis	Mögliche (oft vorkommende) Wertebereiche	A	4
Data Type Analysis	Genauere Auswertung des Datentyps	A	4
Unique Key Analysis	Primärschlüssel	A	2
Aggregationen	Minimum, Maximum ...	A	4
Functional Dependency	Abhängigkeiten von Spalten	A	4
Referential Analysis	Abhängigkeiten von Tabellen	A	4
Domain List	Liste von möglichen Werten	R	4
Domain Pattern List	Muster wie ein Attribut aufgebaut sein muss	R	4
Domain Range	Wertebereich	R	4
Common Format	Zuweisung von vordefinierten Formaten (z.B Sozialversicherungsnummer,...)	R	4
No Nulls	Null-Werte dürfen nicht vorkommen	R	4
Custom	Selbstdefinierte Regeln (SQL mit Parameterübergabe)	R	1, 2, 3, 4
Match-Merge Operator	Zusammenführen von Duplikaten	A, V	1
Name and Address Operator in a Mapping	Käufliche Adressenüberprüfung	A, V	4
CONCAT	Fügt zwei Attribute in Einem zusammen, solange die Datentypen CHAR und VARCHAR vorliegen.	V	4
INITCAP	Gibt den einzelnen Wörtern von einem Attribut einen großen Anfangsbuchstaben.	V	4
LENGTH, LENGTH2, LENGTH4, LENGTHB, LENGTHC	Ermittelt die Zeichenanzahl eines Attributes.	V	4
LOWER	Gibt das Attribut in Kleinbuchstaben aus.	V	4
LTRIM	Löscht von links beginnend alle definierten Zeichen bis die erste Ausnahme auftritt.	V	4
REPLACE	Ausgewählte Zeichen bzw. Zeichenketten werden durch ebensolche Festgelegte ausgetauscht.	V	4
RPAD	Fügt am Ende des Attributes ein Zeichen/eine Zeichenkette so oft an, bis die neuen Zeichen eine gewisse Länge erreichen.	V	4
RTRIM	Analog zu LTRIM, nur auf der rechten Seite des Attributes	V	4
SUBSTR, SUBSTR2, SUBSTR4, SUBSTRB, SUBSTRC	Gibt eine bestimmte Anzahl von Zeichen einer Zeichenkette aus	V	4
TRIM	Löscht alle vorangehenden und nachfolgenden Leerzeichen eines Attributwertes	V	4
UPPER	Gibt das Attribut in Großbuchstaben aus.	V	4

Tabelle 25: Methodeneinteilung Oracle®

Der Oracle® Warehouse Builder stellt eine Vielzahl von Möglichkeiten zur Analyse von Daten zur Verfügung, um eventuelle Fehler und Inkonsistenzen zu entdecken. Ein wichtiger Teil wird dabei durch das Data Profiling übernommen, welches im Vorfeld die Daten untersucht und diese durch eine graphische und tabellarische Unterstützung entsprechend darstellt. Durch eine Anwendung von festgelegten Regeln werden die Zieldaten bereinigt. Weiters bietet der Oracle® Warehouse

Builder durch die zur Verfügungstellung des „Match-Merge Operators“ und des „Name and Address Operator in a Mapping“ eine weitere Möglichkeit mit Duplikaten und Adressen auf einfache Art umzugehen. Der Name and Address Operator ist durch den Zukauf von Datenmaterial durch Drittanbieter realisierbar. Durch die in Kapitel 6.3.3.3 vorgestellten Transformationen können außerdem die Daten vorverarbeitet werden, um so eventuell vorhandene Fehler leichter zu erkennen. Als Beispiel dienen hier die vorangehenden und nachfolgenden Leerzeichen eines Attributwertes.

6.4 SAS® 9.1.2 Data Quality Server

Dieses Kapitel beschreibt welche Funktionen der SAS® Data Quality Server zur Verfügung stellt, um vorliegende Daten zu analysieren, zu bereinigen, zu transformieren und zu standardisieren. Dadurch werden die Daten so aufbereitet, dass durch die Steigerung der Fehlerfreiheit mehr Nutzen aus den Daten gewonnen werden kann. Die zugrundeliegenden Daten wurden aus der PDF-Ausgabe der Onlinedokumentation gewonnen [SAS08b]. SAS® ermöglicht den Umgang mit Dateien in SAS Format sowie dem „Blue Fusion Data“ Format. Das BFD Format wird von SAS® und der dfPower Studio Software[6] erkannt. Der SAS® Data Quality Server stellt folgende Funktionen zur Bearbeitung der Daten zur Verfügung, welche im Rahmen dieser Arbeit von Interesse sind [SAS08b, S. 1]:

- DQMATCH
- DQCASE
- DQPARSE
- DQPATTERN
- DQSTANDARDIZE

In den folgenden Unterkapiteln werden diese Prozeduren näher erläutert.

6.4.1 SAS - DQMATCH Funktion

Ziel der DQMATCH Funktion ist die Generierung eines Matchcodes für einen Attributwert. Dieser Matchcode repräsentiert eine verdichtete Version des Attributwertes. Die Qualität der Information wird durch ein Sensitivitätslevel, das bei der Generierung mitgegeben wird, festgelegt. Bei einem hohen Sensitivitätslevel müssen zwei Werte sich sehr ähnlich sein, um denselben Matchcode zu produzieren. Ist das Sensitivitätslevel niedrig angesetzt, so wird, obwohl beide Werte sehr unterschiedlich sind, der gleiche Matchcode generiert [SAS08b, S. 56f].

[6] vgl. http://www.dataflux.com/Technology/Products/dfPower-Studio.asp

Hierbei kommt folgende Syntax zur Anwendung:

DQMATCH(char,'match-definition'<,sensitivity,'locale'>)

„Char“ steht hierbei für den Wert für den der Matchcode zu der festgelegten „match-definition“ generiert wird. Die „match-definition“ legt den Namen der Vergleichsdefinition fest, welcher auch in der spezifizierten „locale“ enthalten sein muss. Die „locale“ spezifiziert die Länderabhängigkeit des Wertes. „Locale“ sind auf spezielle Gegebenheiten (Formatierungen, Schreibweisen, Abkürzungen) einzelner Länder abgestimmt und müssen gegebenenfalls in das Programm eingespielt werden. „ENUSA“ stellt Definitionen zur Verfügung, welche speziell auf „Englisch“ und die Region der Vereinigten Staaten von Amerika abgestimmt sind[7]. Die „sensitivity“ stellt den Gegenwert dar, wie viel Informationsgehalt der Matchcode beinhaltet. Der Standardwert beträgt hierbei 85, wobei ein Wertebereich zwischen 50 und 95 zulässig ist. Bei einem Wert >85 werden somit mehr Informationen in den Matchcode inkludiert [SAS08b, S. 56].

Im folgenden Beispiel wird ein Matchcode generiert, welcher den höchsten Informationsgehalt über den Inputparameter enthält:

mcName=dqMatch('Dr. Jim Goodnight', 'NAME', 95, 'ENUSA');

Mit Hilfe der DQMATCH Prozedur ist es möglich eine Tabelle von Referenzwerten mit den dazugehörigen Matchcodes zu erstellen [SAS08b, S. 19ff]. Damit ist es möglich, Werte miteinander zu vergleichen, um damit etwaige Duplikate identifizieren zu können.

6.4.2 SAS - DQCASE Funktion

Die DQCASE Funktion gibt die zu bearbeitende Zeichenkette mit einer einheitlichen Großschreibung zurück. Hierbei ist es möglich jegliche Buchstabenfolgen zu verarbeiten, wie z.B. Namen, Organisationen und Adressen. Zusätzlich können mit Hilfe dieser Funktion mehrere angrenzende Leerzeichen durch ein einziges ersetzt werden, sodass ein einheitliches Schriftbild generiert wird.

Bei der DQCASE Funktion kommt folgende Syntax zur Anwendung:

DQCASE(char,'case-definition'<,'locale'>)

„Char“ repräsentiert hier den Wert, der auf Grund der in der „case-definition“ angeführten Formatvorlage transformiert wird. Da sich die „case-definition“ wiederum aus der festgelegten „locale“ ableitet, muss auf diese speziell geachtet werden (siehe Kapitel 6.4.1).

Das folgende Beispiel zeigt den Aufbau der DQCASE Funktion:

[7] vgl. ISO 3166-Codes („ENUSA“ = „en-us“ nach ISO 3166); http://www.iso.org/iso/country_codes, letzter Aufruf am 16.09.2008

orgname=dqCase(“BILL’S PLUMBING & HEATING“,’Proper’,’ENUSA’);

Als Ergebnis wird der Variable „orgname“ folgender Wert zugewiesen: „Bill’s Plumbing & Heating“ [SAS08b, S. 48f].

6.4.3 SAS - DQPARSE Funktion

Die DQPARSE Funktion analysiert einen Attributwert und gibt entsprechend den eingetragenen Optionen einen Wert zurück, in dem nun mittels Trennzeichen jene einzelnen Werte (Subwerte) gekennzeichnet sind, die auf Grund der „parse-definition“ herausgefunden werden konnten. Durch dieses Trennzeichen wird mit Hilfe von zwei weiteren Funktionen der Zugriff auf die einzelnen Token gewährleistet. Es handelt sich hierbei um die Funktionen DQPARSETOKENGET und DQTOKEN.

Die DQPARSE Funktion verwendet folgende SAS® typische Syntax.

DQPARSE(char,’parse-definition’<, ’locale’>)

Im nachfolgenden Beispiel werden zuerst durch die Funktion DQPARSE die Trennzeichen gesetzt und danach durch die Funktion DQPARSETOKENGET die einzelnen Werte zugewiesen und ausgegeben, sodass „prefix = Mrs.“ und „given = Sallie“ ist [SAS08b, S. 59ff].

parsedValue=dqParse(Mrs. Sallie Mae Pravlik’, ’NAME, ’ENUSA’);
prefix=dqPaarseTokenGet(parsedValue, ’Name Prefix’, ’NAME’, ’ENUSA’);
given=dqPaarseTokenGet(parsedValue, ’Given Name’, ’NAME’, ’ENUSA’);

6.4.4 SAS - DQPATTERN Funktion

Die DQPATTERN Funktion gibt zurück, ob ein Zeichen bzw. eine Zeichenkette aus Ziffern, Buchstaben, Sonderzeichen oder aus einem Zusammenspiel dieser Zeichen besteht. Die Wahl der „pattern-analysis-definition“ gibt Auskunft über die Art der Analyse, welche wiederum von der „locale“ abhängig ist. Mögliche Rückgabewerte sind [SAS08b, S. 64f]:

- $*$ Sonderzeichen
- A Buchstaben
- M Mixtur aus Buchstaben, Ziffern und Sonderzeichen
- N Ziffern

Die Syntax lässt sich wie folgt abbilden:

DQPATTERN(char,’pattern-analysis-definition’<,’locale’>);

Folgendes Beispiel verdeutlicht die Funktion DQPATTERN :

pattern=dqPattern('Widgets 5“ 32CT', 'WORD', 'ENUSA');
put pattern;

Das Attribut pattern würde nach Aufruf dieser Funktion folgenden Wert haben: A N∗ M. Würde zum Beispiel CHARACTER als „pattern-analysis-definition“ verwendet, würde der Wert folgendermaßen lauten: AAAAAAA N∗ NNAA.

6.4.5 SAS - DQSTANDARDIZE Funktion

Mit Hilfe der „locale“ werden Standarddefinitionen für verschiedene Kontexte, wie z.B. für Namen, Datumsangaben oder auch Postleitzahlen zur Verfügung gestellt. Dies ist abhängig davon welche 'locale' verwendet und geladen ist. Mittels der DQSTANDARDIZE Funktion ist es möglich Werte in Bezug auf ihre Darstellung zu verändern (Leerzeichen, Format, Verwendung von standardisierten Abkürzungen). Nach Anwendung dieser Funktion wird der entsprechende Wert in angemessener Darstellung ohne Zeichensetzung und mit vernachlässigbaren Leerräumen zur Verfügung gestellt [SAS08b, S. 73f].

SAS ® stellt dazu folgende Syntax zur Verfügung:

DQSTANDARDIZE(char, 'standardize-definition'<,locale>)

Folgendes von SAS® zur Verfügung gestellte Beispiel wird dies noch einmal verdeutlichen:

Eingabe	Ausgabe
data _null_;	Name: House, KEN
length name stNaame $ 50;	StdName: Ken House
input name $char50.;	
stdName=dqStandardize(name, 'Name');	Name: House, Kenneth
put 'Name:' @10 name /	StdName: Kenneth House
'StdName:' @10 stdName /	
datalines;	Name: House, Mr. Ken W.
HOUSE, KEN	StdName: Mr Ken W House
House, Kenneth	
House, Mr. Ken W.	Name: MR. Ken W. House
MR. Ken W. House	StdName: Mr Ken W House
;	
run;	

Tabelle 26: Beispiel für die Funktion SAS® - DQSTANDARDIZE in Anlehnung an [SAS08b, S. 74]

6.4.6 SAS® 9.1.2 Data Quality Server - Schlussfolgerungen

Die zur Verfügung gestellten Funktionen des SAS® Data Quality Server werden in Tabelle 27 noch einmal zusammengefasst.

Name	Beschreibung	Gruppe	Fehlerquelle
DQMATCH Funktion	Erzeugt Matchcodes zur Duplikatenüberprüfung	A	1
DQMATCH Prozedur	Erzeugt eine gesamte Tabelle mit Matchcodes zur Duplikatenüberprüfung	A	1
DQCASE	Vereinheitlichung des Schriftbildes	V	1, 4
DQPARSE	Aufsplittung des Attributwertes in Subwerte	A,V	1, 4
DQPATTERN	Gibt an, ob ein Wert eine Zahl, Buchstabe, Sonderzeichen oder eine Mixtur aus Zeichen ist.	A	4
DQSTANDARDIZE	Überprüfung von Datum, Namen und Postleitzahlen nach vorgegeben Standards.	A, V	4

Tabelle 27: Methodeneinteilung SAS®

Der SAS® Data Quality Server stellt vor allem ein Werkzeug zur Analyse der vorhandenen Daten dar. Durch die Erstellung von Referenztabellen und der DQMATCH Funktion ist das Auffinden von Duplikaten sehr leicht möglich. Des Weiteren wird mit den Funktionen DQCASE, DQPATTERN und DQSTANDARDIZE die Möglichkeit zur Verfügung gestellt, Informationen über die Beschaffenheit der Daten zu bekommen und diese dann zu standardisieren und einheitlich darzustellen. Zusätzlich besteht auch die Möglichkeit Attributwerte aufzuspalten, um einen besseren Überblick über die Daten zu gewinnen.

Der SAS® Data Quality Server stellt eine Vielzahl von Funktionen zur Bearbeitung der Daten zur Verfügung, dennoch ist es für eine intensivere Bearbeitung der Daten notwendig auf weitere von SAS® bzw. Partnerfirmen käuflich erwerbbare Werkzeuge zurückzugreifen. Hierbei handelt es sich vor allem um DataFlux[8] und den SAS® Enterprise Data Integration Server[9], welcher auch auf die DataFlux Technologie aufgebaut ist.

6.5 WinPure ListCleaner Pro

WinPure ListCleaner Pro [Win08a] stellt ein Werkzeug dar, mit dem Listen / Tabellen bereinigt werden können. Hierbei ist es möglich Datensätze zu ändern, zu korrigieren bzw. zu löschen, Duplikate zu finden und die Listen zu standardisieren. WinPure ListCleaner Pro ist in der Lage verschiedene Datenformate zu lesen und zu bearbeiten (MS Access, MS SQL Server Dateien, MS Excel, Text und DBase Formate). Um die Daten zu bearbeiten, stellt WinPure ListCleaner Pro acht verschiedene Module zur Verfügung. Es sind dies im Einzelnen:

[8] vgl. http://www.dataflux.com/, letzter Aufruf am 16.07.2008

[9] vgl. http://www.sas.com/technologies/dw/entdiserver/index.html, letzter Aufruf am 16.07.2008

- Data Table
- Statistics
- Case Converter
- Text Cleaner
- Column Cleaner
- E-mail Cleaner
- Dupe Remover
- Table Matcher

Diese Module erlauben die Bearbeitung der Daten unter anderem mit Hilfe folgender Operationen: Ändern / Löschen von Datensätzen, Bearbeiten von Zeichen einzelner Attributwerte, Ändern ganzer Attributwerte oder dem Finden und Löschen von Duplikaten. Dabei können vor allem fehlende Daten mit Hilfe eines Scoring Systems identifiziert werden. In WinPure ListCleaner Pro können eine oder mehrere Listen korrigiert bzw. in weiterer Folge zusammengeführt werden. E-mail Adressen können auf einfache Weise korrigiert werden, da das System selber Korrekturvorschläge nennt. Ebenso können Namen und Adressen einfach standardisiert werden. In den folgenden Kapiteln werden die Bearbeitungs- und Darstellungsmöglichkeiten die WinPure ListCleaner Pro zur Verfügung stellt, näher erläutert.

6.5.1 WinPure ListCleaner Pro - Data Table

Mit dem Modul Data Table werden im WinPure ListCleaner Datensätze bearbeitet. Beim Import ist darauf zu achten, ob mittels Setzung eines Häkchens die Spaltennamen richtig importiert werden. Mit Hilfe diese Modules können verschiedene Operationen durchgeführt werden: ändern / löschen von Datensätzen bzw. sortieren (siehe Abbildung 21). Mit diesem Werkzeug ist es außerdem möglich, eine zweite Tabelle gleichzeitig zu bearbeiten und anzusehen. Abgeschlossene Transformationen können anschließend gespeichert werden. Hierzu können die Daten überschrieben oder in einer neuen Access, Excel, Text oder DBase Datei gespeichert werden.

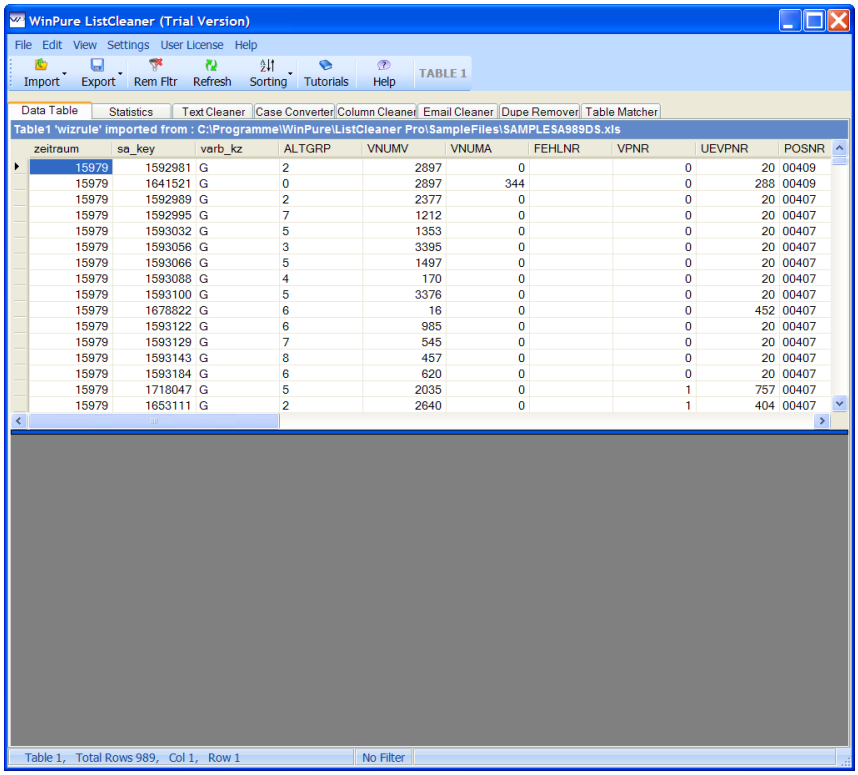

Abbildung 21: WinPure ListCleaner Pro - Data Table

6.5.2 WinPure ListCleaner Pro - Statistics

Es handelt sich hierbei um ein Modul, das dem Benutzer auf schnelle und effiziente Weise alle Zellen nach fehlenden Werten absucht und das Ergebnis danach graphisch aufbereitet. Der Benutzer hat dabei die Möglichkeit eine Auswahl betreffend der Darstellung (chart type), der zu analysierenden Spalten und der Untersuchung nach leeren / belegten Zellen zu treffen. Eine entsprechende Anfrage nach den leeren Zellen ist in Abbildung 22 ersichtlich.

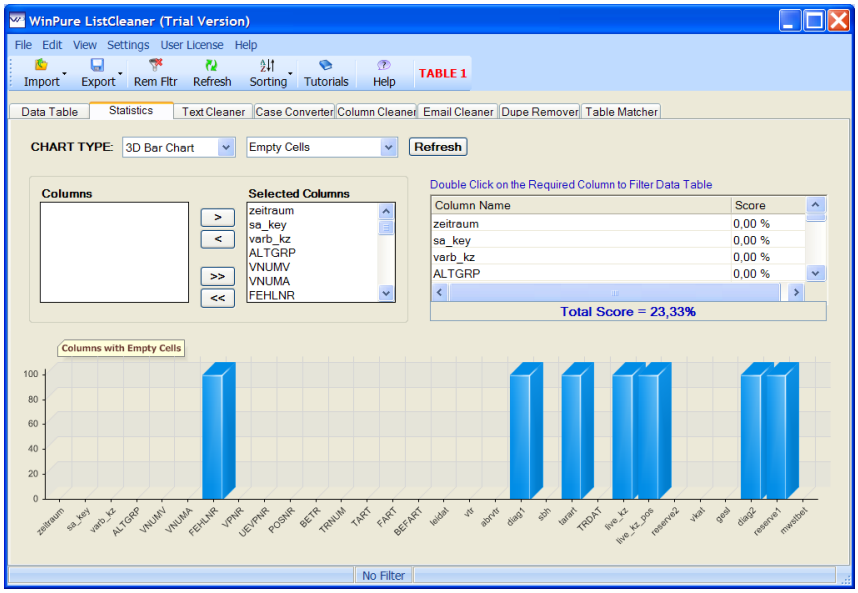

Abbildung 22: WinPure ListCleaner Pro - Statistics

6.5.3 WinPure ListCleaner Pro - Text Cleaner

Das Text Cleaner Modul stellt Operationen zur Verfügung, mit denen verschiedenste unerwünschte Zeichen entfernt werden können. Dies ist sehr hilfreich um nachfolgend die Suche nach Duplikaten zu vereinfachen. Mit Hilfe von Filtern können die Daten entsprechend eingeschränkt werden (Spalten, Zeichen (alpha), Ziffern (numeric), Sonderzeichen (puncts) und allfällige andere). Das Text Cleaner Modul bietet folgende Möglichkeiten zur Datenbearbeitung:

- Remove Leading and Trailing Spaces - Löschen aller Leerzeichen vor dem ersten Zeichen und nach dem letzten Zeichen
- Remove Double Spaces and Repetition of non-digit non-alpha chars - Löschen von doppelten (Sonder-) Zeichen: Bsp.: @@, !!, zwei Leerzeichen, u.ä.
- Remove Non-Printable Characters - Löschen von allfälligen Sonderzeichen (Zeilenschaltung), welche nicht unmittelbar sichtbar sind.
- Alpha only column cleaning - Bearbeitung von Buchstaben zur Bereinigung von Fehlern
  - Convert noughts to O's and ones to L's - Umschreiben der Nullen in O's und der 1 in L's.
  - Remove all non-digit, non-alpha characters (like $, „, etc.) except spaces, dot, comma, hyphen, apostrophe - Löschen aller Zeichen, bei denen es sich nicht um Zahlen und Buchstaben handelt ausgenommen Leerzeichen, Komma, Bindestrich und Apostroph.

  - Remove all digits - Löschen aller Ziffern.

- Numeric only column cleaning - Bearbeitung von Ziffern zur Bereinigung von Fehlern.
  - Convert 'O's to noughts, 'L's and i's to ones - Oftmals treten O's und L's anstelle von „0“ und „1“. Durch dieses Methode kann dies korrigiert werden (Beispiel: 3l45O4 wird zu 314504).
  - Remove all non-digit, except spaces, dot, comma, hyphen, apostrophe - Löschen aller alphanumerischen Zeichen ausgenommen Leerzeichen, Punkt, Komma, Bindestrich und Apostroph.

- Remove all dots, commas, hyphens, apostrophes - Löschen aller Punkte, Beistriche, Gedankenstriche und Apostrophe.
- Remove all Spaces - Löschen aller Leerzeichen.

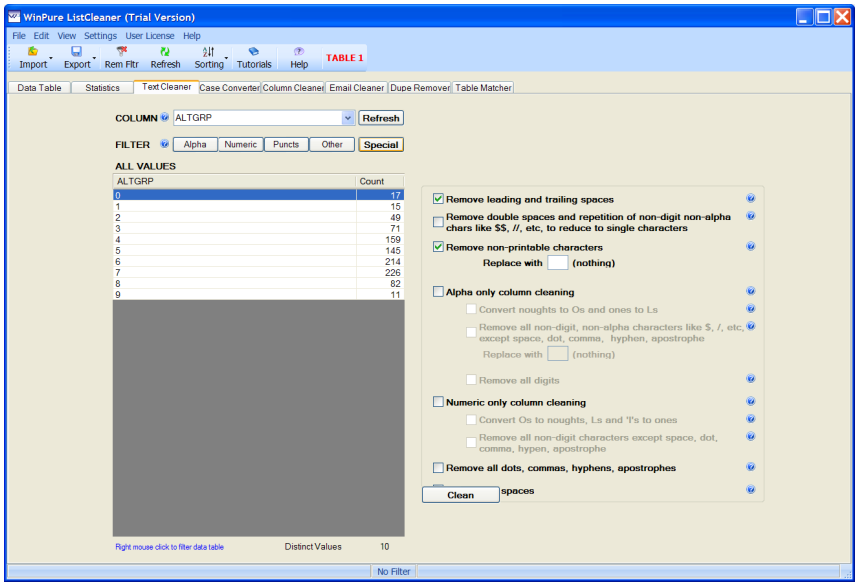

Abbildung 23: WinPure ListCleaner Pro - Text Cleaner

6.5.4 WinPure ListCleaner Pro - Case Converter

Das Modul Case Converter Modul kann eine einheitliche Rechtschreibung der Daten gewährleisten. Es stellt dieselben Filter wie das Text Cleaner Modul (siehe Kapitel 6.5.3) zur Verfügung. Nachdem die entsprechenden Zellen ausgewählt wurden, können drei verschiedene Schreibweisen ausgewählt werden. Zum einen können die Daten entweder alle groß bzw. klein geschrieben werden (siehe Abbildung 24), zum

anderen gibt es Mischformen, wie z.B. in Namen (schottisch: McCartney oder Eigennamen: UNO). Hierzu ist es nötig, das entsprechende Wissen über mögliche Ausnahmen und Prefixe (z.B. Mc) dem System mitzuteilen. Dafür werden eigene Listen mitgeführt.

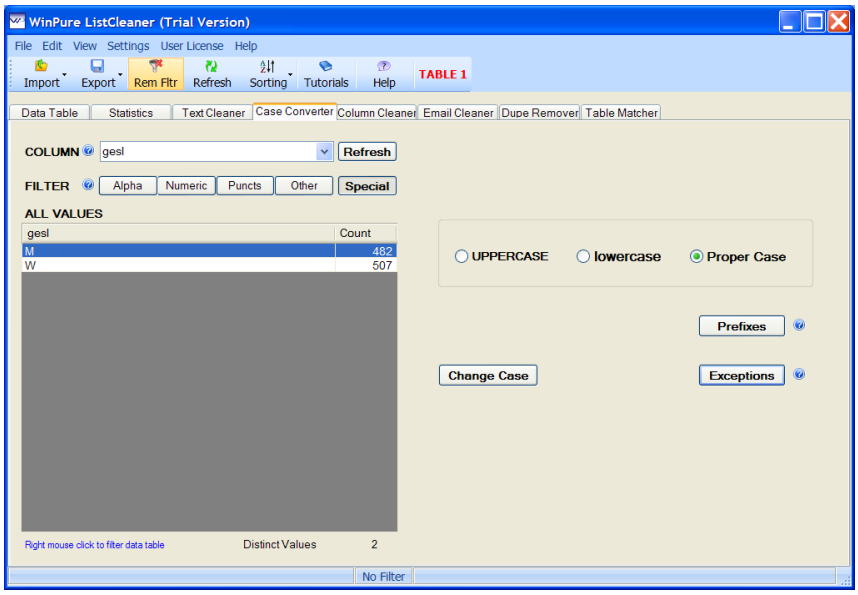

Abbildung 24: WinPure ListCleaner Pro - Case Converter

6.5.5 WinPure ListCleaner Pro - Column Cleaner

Das Column Cleaner Modul hilft dabei, falsche Werte in Zellen pro Spalte leicht zu erkennen und auch auszubessern. Durch die Auswahl der zu ändernden Werte und durch die Zuweisung des neuen Wertes ist es möglich eine Spalte sehr schnell und effizient zu bereinigen. Hierzu können die in Kapitel 6.5.3 vorgestellten Filter zu Hilfe genommen werden. Als anschauliches Beispiel dient hier die Spalte „gesl“ (siehe Abbildung 25). In diesem Fall ist „maennlich“ anstatt von „M“ eingetragen (siehe Kapitel 6.3.2). Durch diese von WinPure ListCleaner Pro zur Verfügung gestellte Bereinigungsform werden alle 482 Fälle auf einmal, schnell und einfach auf den richtigen Wert gesetzt.

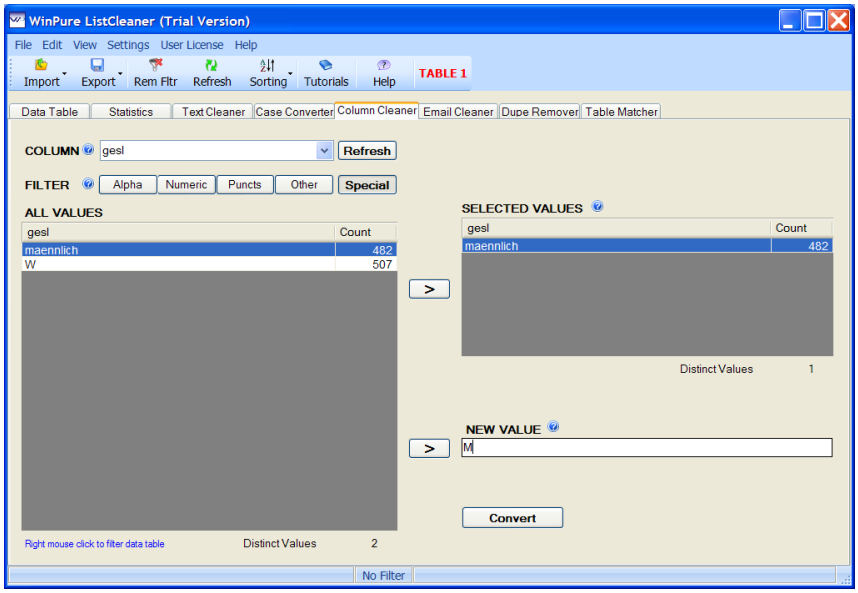

Abbildung 25: WinPure ListCleaner Pro - Column Cleaner

6.5.6 WinPure ListCleaner Pro - E-mail Cleaner

Das Modul E-mail Cleaner ermöglicht eine einfache Trennung von korrekten und falschen E-mail Adressen. Es erfolgt hierbei eine Aufspaltung der Adresse in Account / Sub-domain / Domain / Country. Weiters wird vom Programm eine Verwaltung der Top-Level Domains zur Verfügung gestellt. Zusätzlich liefert das Programm Vorschläge wie falsche E-mail Adressen korrigiert werden können und ermöglicht es, die falschen Adressen gleich durch die Vorschläge zu ersetzen. Durch einen zusätzlichen Filter können alle falschen Adressen angezeigt und noch einmal händisch im Data Table (siehe Kapitel 6.5.1) nachgebessert werden. Im Anschluss daran besteht die Möglichkeit durch nochmalige Kontrolle im E-mail Cleaner Modul eine Neuberechnung durchzuführen und etwaige falsche Adressen, falls nötig, schnell und sauber zu löschen.

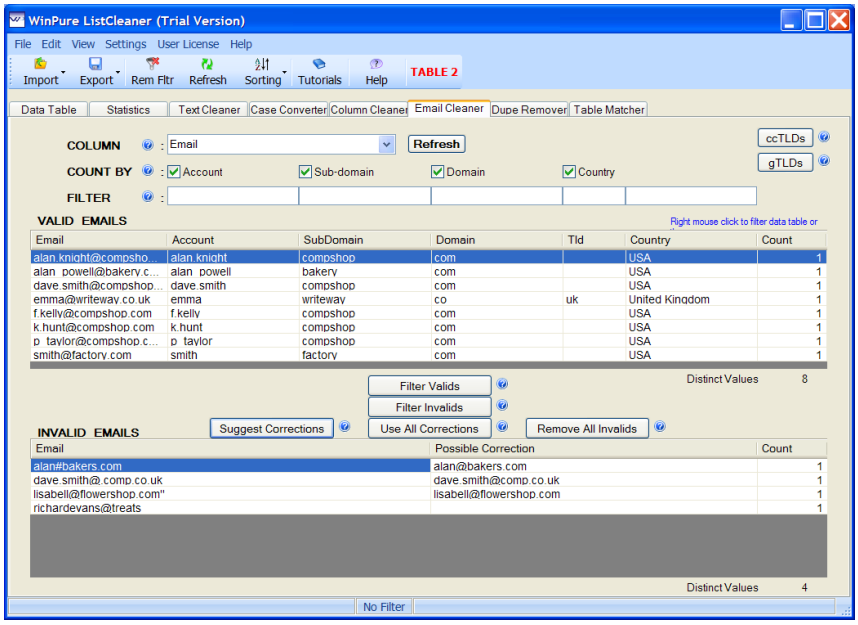

Abbildung 26: WinPure ListCleaner Pro - E-mail Cleaner

6.5.7 WinPure ListCleaner Pro - Dupe Remover

Das Duplicate Remover Modul findet auf einfache Weise Duplikate und hilft diese entsprechend zu reduzieren. Dazu müssen die Tabellen ausgewählt werden, die zur Überprüfung herangezogen werden sollen. WinPure ListCleaner Pro zeigt in graphischer Unterscheidung auf, welche Datensätze möglicherweise Duplikate darstellen. In weiterer Folge bietet WinPure ListCleaner Pro drei mögliche Aktionen zur Behandlung der Duplikate an (siehe Abbildung 27):

1. Exlude - Durch Auswahl einzelner Datensätze werden diese von der Duplikatenliste gestrichen.

2. Delete - Die Duplikate werden gelöscht (Auswahl „Alle“: Es werden alle gelöscht bis auf einen zufällig vom System ausgewählten Datensatz)

3. Filter & Export - Durch diese Auswahl werden entweder die durch die Abfrage herausgefundenen Duplikate oder das Komplementär zur Weiterverarbeitung dargestellt.

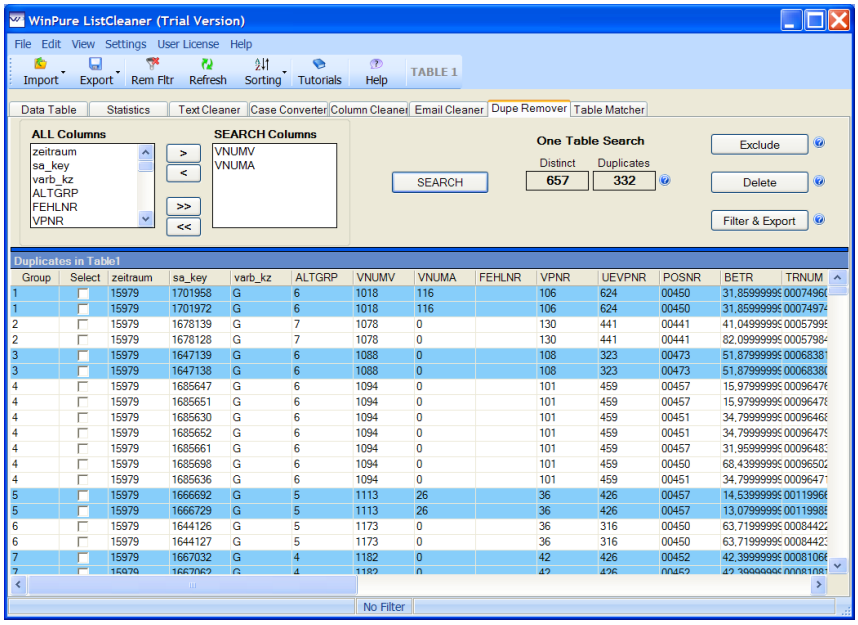

Abbildung 27: WinPure ListCleaner Pro - Dupe Remover

6.5.8 WinPure ListCleaner Pro - Table Matcher

Im Table Matcher Modul werden zwei Listen miteinander verglichen. Der WinPure ListCleaner Pro ist in der Lage bis zu 100 Spalten und 150.000 Zeilen zu importieren. Nachdem die entsprechenden Spalten, welche die Attributwerte beinhalten, die zum Herausfiltern der Duplikate herangezogen werden sollen, selektiert wurden, listet WinPure ListCleaner Pro alle möglichen Duplikate in Gruppen sortiert auf. Hier kann auf die in Kapitel 6.5.7 genannten Operationen zurückgegriffen werden, sodass eine zusammengesetzte Liste ohne Duplikate erzeugt wird (siehe Abbildung 28).

1. Exlude - Datensatz ist kein Duplikat
2. Delete (Selected, Matching) - Ausgewählte bzw. zusammenpassende Datensätze werden gelöscht
3. Filter & Export - Erstellen einer neuen Tabelle

Nach erfolgreicher Bearbeitung (Exportierung) der Daten ist es möglich mit Hilfe des Data Table Moduls (siehe Kapitel 6.5.1) die neue Tabelle nochmals zu bearbeiten bzw. durchzusehen.

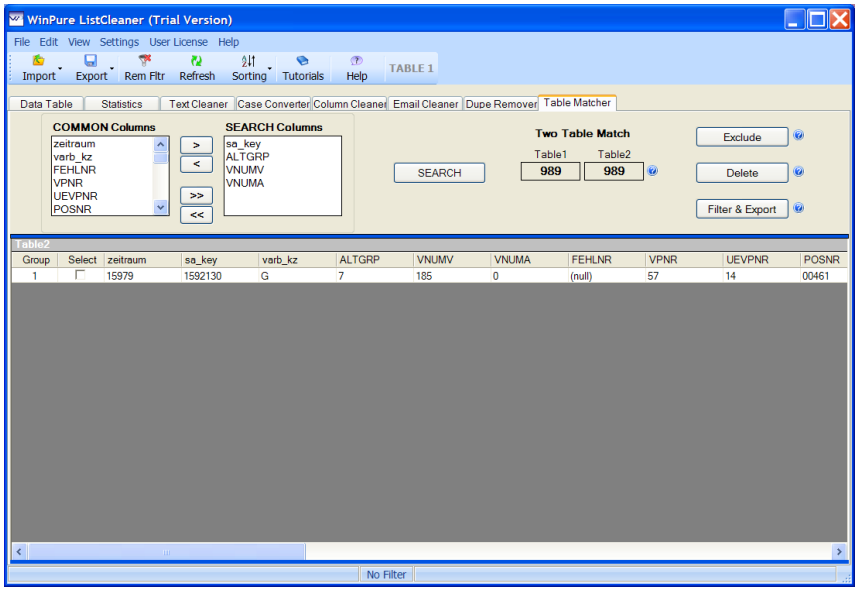

Abbildung 28: WinPure ListCleaner Pro - Table Matcher

6.5.9 Unterschied ListCleaner Pro - Clean and Match 2007

WinPure bietet für einen größeren Einsatzbereich das Werkzeug Clean und Match 2007 an. Im wesentlichen werden dieselben Elemente unterstützt, jedoch sind diese teilweise unterschiedlich aufgebaut. Der größte Unterschied findet sich in der Anzahl der Spalten bzw. Zeilen, die bearbeitet werden können. In Clean und Match 2007 wird der Import von bis zu 250 Tabellenspalten unterstützt sowie eine maximale Anzahl von 2 mal 250.000 Tupel. ListCleaner Pro ist auf 100 Spalten und 2 mal 150.000 Tupel limitiert. Clean and Match 2007 stellt eine ausgereiftere Duplikatenfindung in den Bereichen: Personennamen, Adressen, Firmennamen und Telefonnummern zur Verfügung. Außerdem besteht die Möglichkeit die Datenbereinigung und Anzeige der Ergebnisse durch eine Dual Screen Fähigkeit gleichzeitig durchzuführen. [Win08a]

Zusammenfassend kann festgestellt werden, dass WinPure Clean and Match 2007 mehr Fähigkeiten in Bezug auf die Auffindung und Zusammenführung von Duplikaten besitzt und eine größere Anzahl von Daten handhaben kann.

6.5.10 WinPure ListCleaner Pro - Schlussfolgerungen

Tabelle 28 zeigt die von WinPure ListCleaner Pro zur Verfügung gestellten Funktionen zur Datenbereinigung noch einmal in übersichtlicher Form.

Name	Beschreibung	Gruppe	Fehlerquelle
Data Table	Gleichzeitiges bearbeiten von 2 Tabellen	A,V,D	4
Statistics	Gibt die Anzahl der Leerzellen aus	A	4
Transformationen	Leerzeichenbereinigung (Anfang / Ende)	V	4
	Löschen doppelter Zeichen	V	4
	Löschen nicht sichtbarer Sonderzeichen	V	4
	Anpassen von Ziffern Buchstaben L und 1 sowie O und 0(Null)	V	4
	Löschen aller Ziffern in einem Attributwert	V	4
	Löschen aller Buchstaben in einem Attributwert	V	4
	Löschen aller Sonderzeichen	V	4
	Löschen aller Zeichensetzungen	V	4
	Löschen aller Leerzeichen	V	4
	Vereinheitlichung des Schriftbildes (case)	V	4
	Massenänderung in einer Spalte	V	4
E-mail Cleaner	Trennung von korrekten und falschen E-Mail Adressen	A,V	4
Dupe Remover	Zeigt die gefundenen Duplikate und gibt eine Auswahl, wie gelöscht werden kann	A,V	1
Table Matcher	Zusammenführen beider Tabellen ohne Duplikate	A,V	1, 4

Tabelle 28: Methodeneinteilung WinPure ListCleaner Pro

WinPure eignet sich gut, um zwei Tabellen gegenüberzustellen und diese zusammenzuführen. Dabei wird besonders darauf Wert gelegt, dass einheitliche Formate geschaffen werden, um so eine bessere Überprüfbarkeit, insbesondere auf Duplikate, zu erreichen. Ein weiterer Schwerpunkt von WinPure liegt in der Bearbeitung von E-mail Adressen, sodass falsche Adressen in den Datensätzen erkannt und gelöscht werden können. WinPure ListCleaner Pro stellt mit 150.000 gleichzeitig bearbeitbaren Tupel ein ansprechendes Werkzeug zur Auffindung von Duplikaten dar. Des Weiteren wird eine Überprüfung erleichtert, da auf einfache Weise Formate und Darstellungen von Attributen bearbeitet werden können.

6.6 WizRule®

Bei WizRule® [Wiz08a] handelt es sich vor allem um ein Werkzeug zur Generierung von Regeln (Wenn-Dann Bedingungen), welche in Relationen vorherrschen. Es zeigt dadurch mögliche Fehler auf, die durch nachträgliche Kontrolle überprüft werden können. In diesem Fall ist es möglich, die erkannten Fehler in externe Datenquellen zu transferieren. Im folgenden Abschnitt wird das Programm genauer beschrieben und die Fähigkeiten des Programms näher erörtert. Dazu teilt sich dieses Kapitel in folgende Abschnitte:

- Einführung
- Dateneingabe
- Rule Report
- Spelling Report

- Deviation Report
- Schlussfolgerungen

Die Information wurde anhand des Demoprogramms sowie der Online-Hilfe bzw. des Online-Training-Videos erstellt.

6.6.1 WizRule® - Einführung

WizRule® ist ein Werkzeug, das auf Grund von mathematischen Analysen Inkonsistenzen in einem eingelesenen Datenbestand feststellen kann. Es basiert auf der Annahme, dass in vielen Fällen Fehler nur eine Abweichung von der Norm sind. Zur Berechnung der Fehler / Abweichungen findet WizRule® zuerst alle möglichen Regeln (Wenn-Dann Bedingungen) heraus, die auf den eingelesenen Datenbestand anwendbar sind. Die Berechnung erfolgt auf Basis von mathematischen Grundlagen. Als Ergebnis liefert WizRule® eine Liste von Fällen die auf Grund der festgestellten Regeln etwaige Fehler bzw. Datensätze darstellen, die näher betrachtet werden sollen.

Im Zuge der Datenanalyse nimmt WizRule® folgende Operationen vor:

1. Lesen der Datensätze - Tätigen von Einstellungen, um WizRule® genauer zu konfigurieren.
2. Berechnen der Regeln sowie deren Zuverlässigkeit.
3. Analysieren eines jeden Datenfeldes in Bezug auf die gefundenen Regeln.
4. Auflisten der Datensätze mit möglichen Fehlern nach deren Wahrscheinlichkeit.

Die Ergebnisse listet WizRule® in drei verschiedenen Berichten auf - Rule (Regeln), Spelling (Rechtschreibfehler) und Deviation (Abweichungen) Report. Die genauere Beschreibung der Vorgehensweise und Ergebnispräsentation findet sich in den folgenden Kapiteln.

6.6.2 WizRule® - Dateneingabe

Bevor die Analyse startet, müssen einige Eingaben vorgenommen werden, um ein aussagekräftiges Ergebnis zu erreichen. In einem ersten Schritt muss die Datenquelle, welche überprüft werden soll, festgelegt werden. WizRule® stellt dazu vier mögliche Eingabevarianten für verschiedene Datenformate zur Verfügung:

Eingabevariante	Datenbank
Direkt	dBase, FoxPro, Clipper und andere Datenbanken mit *.dbf Format Microsoft Access (*.mdb), Microsoft SQL Server Tabellen, Oracle
ODBC (Open DataBase Connectivity) verträgliche	z.B. Access, SQL, Oracle, Sybase, Informix, DB\2,...
OLE DB verträgliche	z.B. Access, SQL, Oracle, Sybase, Informix, DB\2,...
ASCII	ASCII Textdateien

Tabelle 29: Verwendbare Datenformate für WizRule®

Nachdem die Auswahl getroffen wurde, die festlegt, welche Relation bearbeitet wird, kann mittels eines Viewers noch einmal überprüft werden, ob die Anzahl der eingelesenen Datensätze stimmt und alle Daten soweit korrekt vorliegen. Den einzelnen Feldern werden WizRule® spezifische Typen zugewiesen (Auswahl: Number, Quality, Quantity, Money). Diese müssen ebenso überprüft werden. Zusätzlich ist es möglich zu entscheiden, ob das Feld auch analysiert wird, wenn es leer ist, oder ob das Feld bei der WENN bzw. DANN Analyse ignoriert wird (siehe Abbildung 29). Unter dem Reiter „Data Format“ werden Einstellungen zur Darstellung von Zahlen und Währungen getroffen sowie das Ausgabeformat bestimmt (siehe Abbildung 30).

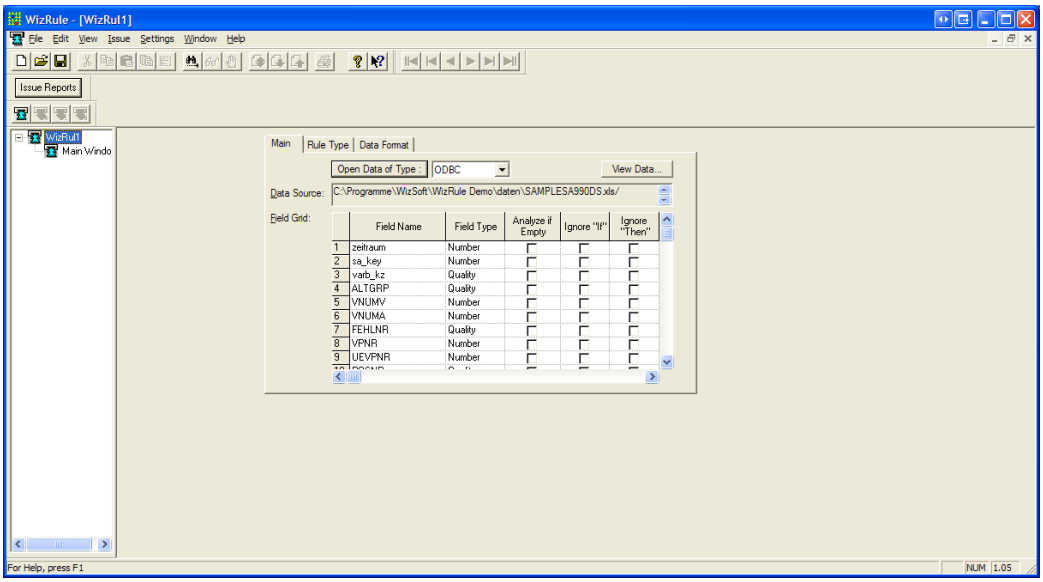

Abbildung 29: Hauptauswahl WizRule®

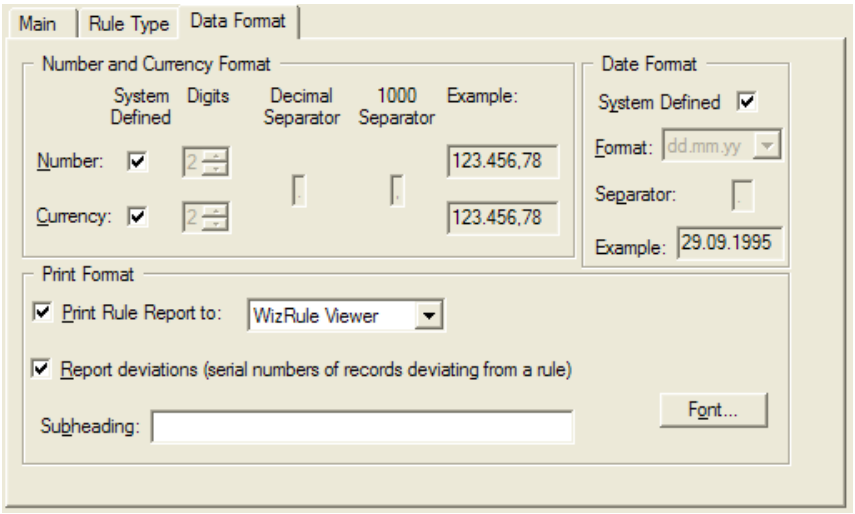

Abbildung 30: WizRule® Formateinstellungen

Die wichtigsten Einstellungen für die Generierung der Regeln können unter dem Reiter „Rule Type“ getroffen werden (siehe Abbildung 31):

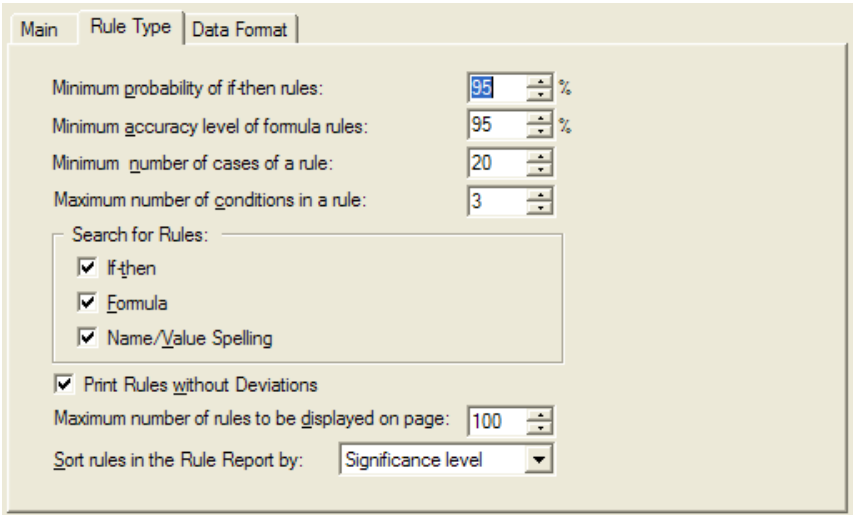

Abbildung 31: WizRule® Regeleinstellungen

- Minimum probability of if-then rules - Dieser Wahrscheinlichkeitswert gibt an, mit welcher Wahrscheinlichkeit die gefundene Wenn-Dann Regel auf jeden Datensatz zutreffen muss. Sie gibt an, mit welchem Prozentwert die jeweiligen Teile in Bezug auf die Bedingung vorkommen. Beispiel: Wenn die Wenn-Bedingung in 40 Datensätzen und die Dann-Bedingung in 38 von diesen 40 Datensätzen vorkommt, ergibt dies eine Wahrscheinlichkeit von 95%.

Der Begriff Rule Probability kann auch mit dem gebräuchlicheren Confidence Level gleichgesetzt werden [KGH05, S. 23]. Wird die Minimumwahrscheinlichkeit auf 100% festgesetzt, findet WizRule® nur jene Regeln bei denen keine Abweichungen bestehen. Je niedriger dieser Wert gesetzt wird, desto mehr Regeln wird WizRule® finden.

- Minimum accuracy level of formula rules - Mit Hilfe dieser Einstellung wird der Genauigkeitsgrad der mathematischen Formeln festgelegt. Als Beispiel für eine solche Formel kann folgendes verwendet werden:

  Field A = Field B x Field C

  Der Genauigkeitsgrad drückt die Relation zwischen dem gesamten Vorkommen der Regel und der Anzahl der Datensätze in denen diese Regel zutrifft aus.

- Minimum number of cases - Mit der Minimumanzahl der Treffer wird festgelegt, wie viele positive Datensätze mindestens erreicht werden müssen, damit eine Regel festgesetzt werden kann. Wenn also 40 Datensätze festgelegt werden, muss in mindestens so vielen Fällen die Regel positiv getestet werden. Der Minimumwert, der hierbei festgesetzt werden kann, beträgt 4. Die Minimumanzahl kann auch mit dem gebräuchlicherem Support Level gleichgesetzt werden [KGH05, S. 23].

  Das Zusammenspiel dieser drei Möglichkeiten erlaubt ein gezieltes Beeinflussen der Anzahl der gefundenen Regeln durch WizRule®.

- Maximum number of conditions - Die maximale Anzahl der Wenn-Bedingungen legt fest, wie viel Bedingungen eine Wenn-Dann Regel haben kann. Je höher diese Zahl ist, desto mehr Regeln werden durch WizRule® entdeckt. Wenn zum Beispiel maximal drei Bedingungen festgelegt werden, werden jene mit einer, zwei oder drei Bedingungen gefunden.

- Search for Rule Types

  Hier kann festgelegt werden, nach welchen Regeln WizRule® sucht:

  - Wenn-Dann-Beziehung

    Miteinbeziehen von Regeln, welche eine Wenn-Dann Beziehung repräsentiert:

    Wenn (Bedingung[en]), Dann (Wert von Attribut X) ist Y.

  - Mathematische Formel

    Suchen nach Regeln, die einen mathematischen Zusammenhang zwischen mehreren Attributen herstellen:

    Attribut A = Attribut B + Attribut C.

– Rechtschreibung

  Diese Regel zeigt eine allgemeine Rechtschreibung von Wörter bzw. Zeichenketten auf, welche in bis zu drei unterschiedlichen Darstellungsformen vorkommt.

Durch eine Auswahl mit der rechten Maustaste - Properties ist es möglich noch weitere Verfeinerungen durchzuführen, welche aber in dieser Arbeit nicht näher erläutert werden [Wiz08b][10].

Zusätzlich kann noch angegeben werden, ob Regeln angezeigt werden, bei denen keine Abweichungen festgestellt wurden, die Anzahl der Regeln pro Seite und ob eine Sortierung vorgenommen werden soll.

6.6.3 WizRule® - Rule Report

WizRule® stellt drei Arten von Regeln zur Verfügung, welche sukzessive abgearbeitet werden. Es handelt sich hierbei um Wenn-Dann Regeln, mathematische Formeln und Rechtschreibregeln (siehe Kapitel 6.6.2), welche auf falsch geschriebene Wörter hindeuten. WizRule® wirft dabei die gefundenen Abweichungen zu den jeweiligen Regeln aus.

Der Bericht gliedert sich in drei Teilbereiche: Zuerst werden die allgemeinen Einstellungen angezeigt. Als nächstes folgen jene Regeln, die immer zutreffen, sogenannte Then-Regeln (Beispiel: „gesl“ is W or M - Rule's probability: 1,000 - The rule exists in 989 records). Als nächstes werden die herausgefundenen Wenn-Dann Regeln dargestellt. In einem letzten Schritt werden die mathematischen Regeln, falls vorhanden, angezeigt.

Zusätzlich kann im Fenster rechts unten (siehe Abbildung 33) leicht herausgefunden werden, welches Feld in welchen Regeln vorkommt. In WizRule® ist es auch möglich, sich die einzelnen Regeln graphisch darstellen zu lassen. Um die Darstellung der Regeln übersichtlicher zu gestalten, kann man mit Rechtsklick auf die Seite und anschließender Auswahl des Punktes Display Rule Options jene Optionen zusätzlich auswählen, welche eine eingeschränkte Darstellungsweise präsentiert (siehe Abbildung 32). Dadurch werden die Regeln nach den eigenen Vorstellungen gefiltert (Beispiel: Top 20 Regeln für ein Feld). Die abweichenden Datensätze werden in der rechten oberen Programmhälfte dargestellt. Dadurch ist eine sofortige formale Überprüfung der Daten möglich (siehe Abbildung 33).

[10] WizRule® Training Video, http://www.wizsoft.com/ruletvideo.asp, download am 15.07.2008

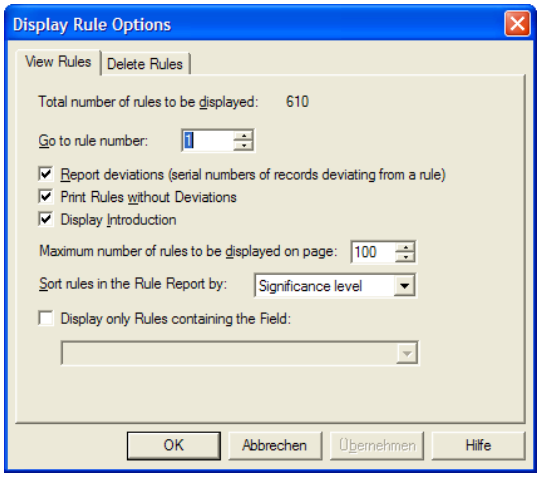

Abbildung 32: WizRule® Display Rule Options

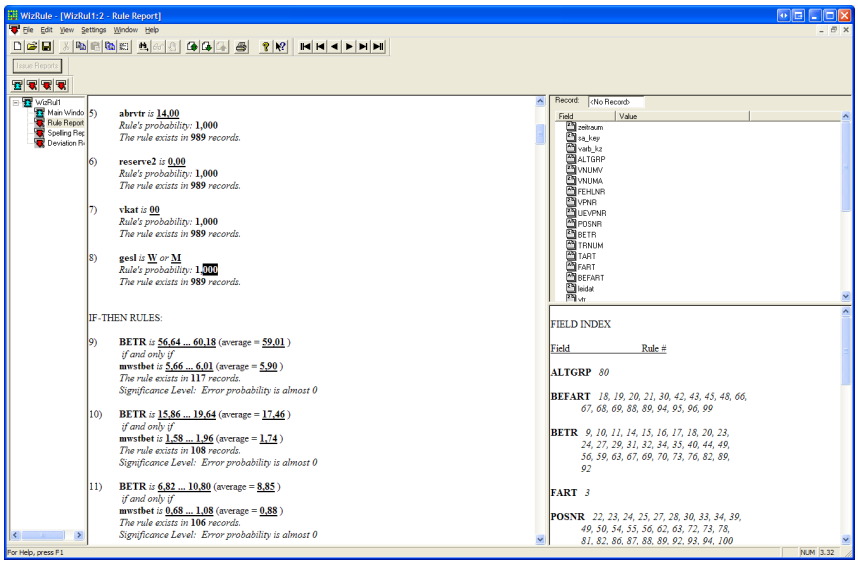

Abbildung 33: Beispiel für einen WizRule® Rule Report

Im folgenden Teil wird nun eine Regel, die aus den Beispieldatensätzen, welche von der OÖGKK zur Verfügung gestellt wurden, generiert wurde, näher erläutert. WizRule® bringt anhand der Beispieldaten folgende Regel zu Tage:

If	POSNR is 00473
Then	BETR is 51,88
Rule's probability:	0,968
The rule exists in 61 records.	
Significance Level:	Error probability is almost 0
Deviations (records' serial numbers):	868, 869

Tabelle 30: Beispielregel geniert von WizRule®

Diese Regel besagt, dass mit einer 96,8%igen Wahrscheinlichkeit, der Wert von BETR 51,88 ist, wenn der Wert von POSNR 00473 beträgt. Bei 61 Einträgen ergibt dies zwei Abweichungen. Die Konfidenz dieser Regel, also das Significance Level, macht nahezu 100% aus. Somit ist eine große Sicherheit für die Richtigkeit dieser Regel gegeben. Die Abweichungen weisen jeweils einen Wert von 25,94 auf (siehe Abbildung 33) und sollen nun auf ihre Fehlerhaftigkeit geprüft werden.

6.6.4 WizRule® - Spelling Report

Mit Hilfe dieses speziellen Berichtes werden Fehler in der Schreibweise aufgedeckt. Hierbei stellt WizRule® die in Abbildung 34 ersichtliche Maske zur Verfügung. Es erfolgt dabei eine Indexierung nach dem Feld. Durch Auswahl über das Drop-down Menü bzw. der Buttons Previous and Next sind alle Fehler einzeln durchsehbar. Linker Hand wird die entsprechende Regel angezeigt und durch Doppelklicken auf die entsprechende Abweichung wird rechter Hand der abweichende Datensatz angezeigt.

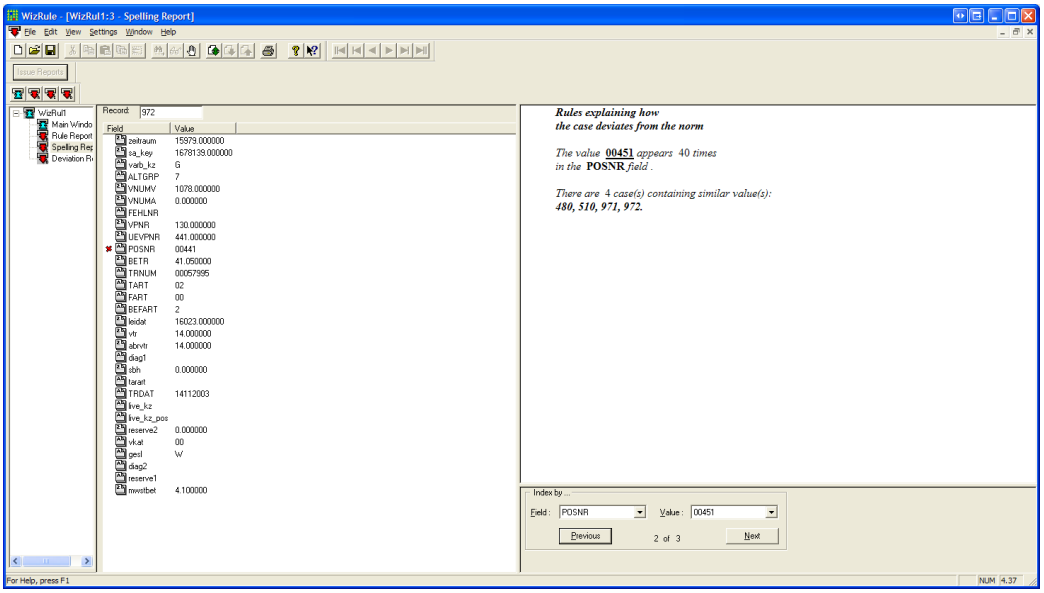

Abbildung 34: Beispiel für einen WizRule® Spelling Report

6.6.5 WizRule® - Deviation Report

Der Abweichungsbericht zeigt die ausgewählte Abweichung (X-Markierung bei VNUMA) sowie die dazugehörige Regel an. WizRule® stellt zur Veranschaulichung der Abweichung drei Möglichkeiten zur Verfügung, wie diese Abweichungen indexiert werden können:

- Level of Unlikelihood - Wie wahrscheinlich ist eine Abweichung
- Datenfeld
- Auswahl des Datensatzes selber

Bei dem Level of Unlikelihood kann entlang der Graphik (Abbildung 35 unten links) abgelesen werden, mit welcher Wahrscheinlichkeit es sich um eine Abweichung handelt, die auch von Interesse für den Betrachter ist. Je weiter links diese Abweichung zu finden ist, desto wahrscheinlicher handelt es sich um einen Fehler, bei dem eine nähere Betrachtung nötig ist. Um eine Überprüfung der Abweichungen zu erleichtern, können diese in eine ASCII, RTF oder Microsoft ACCESS Datei exportiert werden. Dabei kann eine Auswahl der zu exportierenden Abweichungen getroffen werden.

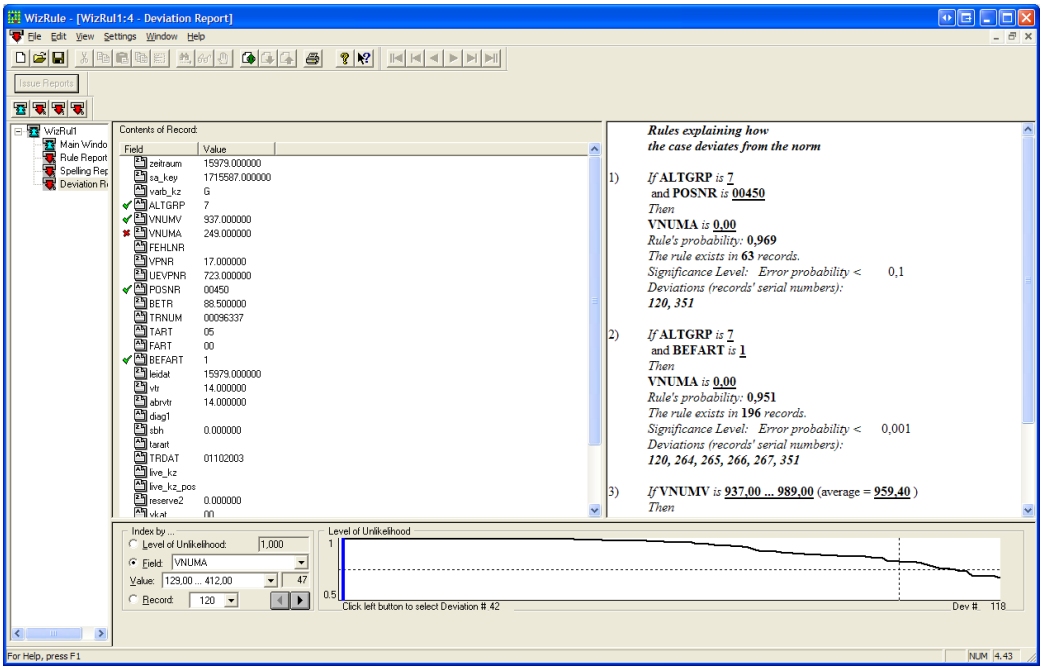

Abbildung 35: Beispiel für einen WizRule® Deviation Report

6.6.6 WizRule® - Schlussfolgerungen

Die nachfolgende Tabelle 31 fasst die von WizRule® zur Verfügung gestellten Funktionen zur Datenbereinigung noch einmal zusammen.

Name	Beschreibung	Gruppe	Fehlerquelle
Regel: Wenn-Dann-Beziehung	Berechnet Beziehungen von Spalten	A,D	4
Regel: Mathematische Formel	Berechnet mathematische Zusammenhänge	A,D	4
Regel: Regeln für die Rechtschreibung	Überprüft die Rechtschreibung der Spaltenwerte untereinander	A,D	4

Tabelle 31: Methodeneinteilung WizRule®

WizRule® eignet sich gut um Daten zu untersuchen und um Beziehungen zwischen einzelnen Datenfeldern herzustellen. Es erleichtert die Suche von Normabweichungen. Durch verschiedene Einstellungsmöglichkeiten (siehe Kapitel 6.6.2) ist es möglich, die Ergebnisse nach den eigenen Wünschen zu verändern, einzugrenzen oder auszuweiten. WizRule® liefert aber nur Vorschläge bzw. Ergebnisse von eventuellen Abweichungen. Alle Daten müssen dennoch ein zweites Mal kontrolliert werden. Eine eigentliche Bereinigung der Daten kann nicht mit Hilfe des Programms vorgenommen werden. Es bietet jedoch die Möglichkeit, die Ergebnisse in verschiedenen Ausgabeformaten abzuspeichern, um die Daten so nachbearbeiten zu können. WizRule® stellt somit ein Werkzeug dar, das Abweichungen und Fehler erkennen aber eine Datenbereinigung selbst nicht durchführen kann.

6.7 Schlussfolgerungen / Zusammenfassung

Im Rahmen dieses Kapitels wurden fünf Werkzeuge zur Behebung von Anomalien und Fehlerquellen in Datenbanken dargestellt. Die nach Marktanteil [NP08] ausgewählten Standardprodukte (Microsoft®, Oracle® und SAS®) bieten sehr ähnliche Lösungsmethoden. Es werden vor allem Methoden zur Transformation der vorliegenden Daten und zur Auffindung und Eliminierung von Duplikaten zur Verfügung gestellt. Microsoft® und Oracle® bieten zur Analyse und anschließenden Bearbeitung der Daten, wie auch in der Literatur diskutiert (siehe Kapitel 3.3), ein vorgegebenes Konzept an (siehe Kapitel 6.2 und 6.3) nach dem systematisch vorgegangen werden kann.

Schwerpunkt des Microsoft® SSIS liegt mit Hilfe der Transformation für Fuzzysuche und der Transformation für Fuzzygruppierung bei der Duplikatenbereinigung. Oracle® bietet vor allem durch die Data Rules (siehe Kapitel 6.3.2) eine Vielzahl von Möglichkeiten, um die Daten zu analysieren sowie in weiterer Folge durch Transformationen (siehe Kapitel 6.3.3) entsprechend zu bearbeiten. SAS® ist durch das Zusammenspiel der DQMATCH Funktion (generieren eines Matchcodes) sowie der DQMATCH Prozedur (generieren einer Matchtabelle) ebenfalls für die Behandlung von Duplikaten geeignet.

Winpure® stellt wie die obengenannten Werkzeuge eine Vielzahl von Transformationen für die Bearbeitung der Attributwerte zur Verfügung, sodass eine Vereinheitlichung der Daten für die anschließende Überprüfung geschaffen wird. Ebenso liegt der Schwerpunkt bei der Auffindung von Duplikaten sowie einer Bereinigung von E-mail Adressen.

Anders als die bisher vorgestellten Werkzeuge ist WizRule® nur auf das Analysieren der Daten spezialisiert. Durch Finden von Wenn-Dann Regeln werden Unregelmäßigkeiten in der Datenhaltung erkannt und textuell sowie graphisch nach deren Vorkommen dargestellt, sodass eine Nachprüfung und Kontrolle der Werte leicht fällt. Zu diesem Zweck besteht die Möglichkeit die gefundenen Regeln zu exportieren (z.B. Microsoft® Access), um dort eine etwaige weitere Aufbereitung durchzuführen. Zusätzlich können durch diese gefundenen Regeln vorkommende Rechtschreibfehler erkannt werden. Ein weiteres Kriterium dieses Werkzeuges ist das Finden von Abhängigkeiten einzelner Spalten, dies ist jedoch auf mathematische Berechnungen begrenzt.

Jedes dieser Werkzeuge bietet verschiedene Möglichkeiten, wie unsaubere Daten gefunden und anschließend bearbeitet bzw. bereinigt werden können. Nun ist es notwendig, die einzelnen Möglichkeiten gegenüberzustellen und das passende Werkzeug oder die notwendigen Funktionen herauszufiltern, um eine für die OÖGKK zweckmäßige Lösung vorschlagen zu können. Eine Aufarbeitung dieser Punkte findet sich in Kapitel 7.

7 Gegenüberstellung der Werkzeuge

Nachdem in Kapitel 6 die fünf ausgewählten Werkzeuge, ihre Methoden und Funktionen zum Analysieren von Datenmaterial sowie Beheben von Fehler vor- bzw. dargestellt wurden, folgt in diesem Kapitel eine Gegenüberstellung dieser Methoden, welche auf Grund der beschriebenen Fähigkeiten erstellt wird. Dadurch wird das Werkzeug identifiziert, welches am besten den Anforderungen der OÖGKK entspricht.

In weiterer Folge wird eine Empfehlung für die weitere Vorgehensweise ausgesprochen, ob ein bestehendes Produkt eingesetzt und eventuell erweitert oder ein neues Framework mit einer Auswahl an geeigneten Funktionen und Methoden spezifiziert werden soll.

7.1 Vergleich der Mächtigkeit der einzelnen Werkzeuge

Die folgenden Tabellen 32 und 33 stellen eine vollständige tabellarische Auflistung der in Kapitel 6 dargestellten Funktionen der einzelnen Werkzeuge dar. Um eine bessere Übersicht zu generieren, werden die einzelnen Funktionen in Themengebiete zusammengefasst.

Funktionsgruppe	SSIS	Oracle	SAS	WinPure	WizRule
Überprüfung von Referenzwerten	Fuzzysuche		DQMATCH Prozedur		
Duplikatensuche	Fuzzygruppierung	Match-Merge Operator	DQMATCH Funktion	Dupe Remover	
				Table Matcher	
Abhängigkeitsüberprüfung		Functional Dependency			Wenn-Dann Beziehung
		Referential Analysis			
Datentyp- sowie Formaterkennung und Formatumwandlung	Transformation für Datenkonvertierung	Pattern Analysis			
Transformationen für Alphanumerische Zeichen		Data Type Analysis			
		Domain Pattern List	DQCASE	Leerzeichenbereinigung (Anfang / Ende)	
		Common Format	DQPATTERN	Löschen doppelter Zeichen	
		No Nulls		Löschen nicht sichtbarer Sonderzeichen	
		INITCAP		Anpassen von Ziffern, Buchstaben	
		LENGTH, LENGTH2, LENGTH4,		L und 1 sowie O und 0(Null)	
		LENGTHB, LENGTHC		Löschen aller Ziffern in einem Attributwert	
		TRIM, LTRIM, RTRIM		Löschen aller Buchstaben in einem Attributwert	
		REPLACE		Löschen aller Sonderzeichen	
		RPAD		Löschen aller Zeichensetzungen	
		LOWER, UPPER		Löschen aller Leerzeichen	
				Vereinheitlichung des Schriftbildes (case)	

Tabelle 32: Zuteilung der Funktionen 1 / 2

Funktionsgruppe	SSIS	Oracle	SAS	WinPure	WizRule
Transformationen zur Bearbeitung von Spalten	Neuzuweisung von Spaltewerten	CONCAT	DQPARSE	Data Table	
	Transformation für abgeleitete Spalten	SUBSTR, SUBSTR2, SUBSTR4,	DQSTANDARDIZE	Massenänderung in Spalten	
	Transformation für das Kopieren von Spalten	SUBSTRB, SUBSTRC			
	Verketten von Werten aus verschieden Spalten in einer neuen Spalte				
	Extrahieren von Zeichen aus einzelnen Zeichenketten				
	Anwenden von mathematischen Funktionen				
	Vergleichen von Spalten und Variablen, Ausgabe eines Wertes				
	Extrahieren von Werten eines Datum-Wertes				
Mathematische Regelfindung	Mathematische Funktionen				Mathematische Formel
Aggregationsbildung		Aggregationen			
selbstdefinierte Regeln		Custom			
E-mail Bereinigung				E-mail Cleaner	
Qualitätsindikatoren	Anzahl der Datensätze	Unique Key Analysis		Statistics	
	Prozentsatz von fehlenden Attributen	Domain Analysis			
	Prozentsatz von Integritätsverletzungen	Domain List			
	Indikator für fehlende Dateistruktur	Domain Range			
	Kennzeichnung von verdächtigen Datenwerten				
	Führen von Grenzwerten zur Analyse				
nicht verwendet	Transformation zum Sortieren	Name and Address Operator in a Mapping			Regeln für die Rechtschreibung
	E-mail versenden				

Tabelle 33: Zuteilung der Funktionen 2 / 2

7.2 Vergleich der Funktionen anhand von Beispieldaten

Im folgenden Abschnitt wird gezeigt, welche Funktionen der einzelnen Werkzeuge eingesetzt werden können, um die Fehlerquellen (siehe Kapitel 5.4) in den von der OÖGKK zur Verfügung gestellten Beispieldaten (siehe Kapitel 5.3.3) zu identifizieren.

7.2.1 Beispieldaten Duplikate

nr	zeitraum	sa_key	ALT GRP	VNUMV	BETR	TART	BE FART	leidat	vtr	abrvtr	TRDAT	gesl	mwst bet
5	15979	1593032	5	1353	59,3	00	2	16009	14	14	31102003	W	5,93
6	15979	1593023	5	1353	59,3	00	2	16009	14	14	31102003	W	5,93

Tabelle 34: Beispieldaten - Duplikate

- **SSIS**
  SSIS stellt mit der Fuzzygruppierung eine Funktion zur Verfügung, die Duplikate erkennen kann. In SSIS definiert der Benutzer einen Operator „Transformation für Fuzzygruppierung“ auf die ausgewählte Tabelle „XY“. Hierzu werden die zu untersuchenden Spalten ausgewählt. In weitere Folge müssen die minimale Ähnlichkeit (Wert zwischen 0 und 1) und der Schwellenwert angepasst werden (siehe Kapitel 6.2.2.2), sodass die Duplikate ermittelt werden können. Eine genauere Beschreibung des Arbeitsablaufes findet sich unter „Identifizieren ähnlicher Datenzeilen mithilfe der Transformation für Fuzzygruppierung“[11]. Die Beispieldatensätze 5 und 6 werden als mögliche Duplikate erkannt, da hier nur der „sa_key“ unterschiedlich ist. Zusätzlich kann mit dem Qualitätsindikator „Anzahl der Datensätze“ ein grober Überblick geschaffen werden, wie viele Datensätze vorhanden sind. Hierbei kann bei einer unüblich großen Anzahl angenommen werden, dass Duplikate vorhanden sind.

- **Oracle**
  Oracle stellt mit dem „Match-Merge“ Operator eine Funktion zur Verfügung, die Duplikate erkennen kann. In Oracle definiert der Benutzer einen Operator „Match-Merge“ auf die Tabelle „XY“. In weiterer Folge werden die Eingabedaten, die Übereinstimmungsregel (z.B. „Any“) und die Ausgabeart festgelegt (siehe Kapitel 6.3.3.1). Oracle stellt hierbei dem Benutzer einen Softwareassistenten zur Verfügung der bei den Einstellungen hilft. Eine genauere Beschreibung des Arbeitsablaufes findet sich unter „Using the Match-Merge

[11] vgl. http://msdn.microsoft.com/de-de/library/ms142155.aspx

Operator“[12]. Im Beispiel werden die Beispieldatensätze 5 und 6 werden als mögliche Duplikate erkannt und in einen Datensatz zusammengeführt.

- **SAS**
  SAS besitzt mit der DQMATCH Funktion sowie mit der DQMATCH Prozedur Funktionen, die auf Grund der eingegebenen Daten Matchcodes und Cluster generieren, in denen Datensätze zusammengefasst sind, welche identische Matchcodes besitzen. Die Matchcodes werden mit Hilfe des Sensitivitätslevels beeinflusst. Je höher dieses Level angesetzt wird desto ähnlicher müssen die Datensätze sein. Die Darstellung eines Beispielcodes findet sich in [SAS08b, S. 19ff]. Im Beispiel werden die Beispieldatensätze 5 und 6 werden als mögliche Duplikate erkannt und demselben Cluster zugeteilt.

- **WinPure**
  WinPure besitzt mit dem Table Matcher Modul ebenfalls eine Funktion, mit der mögliche Duplikate erkannt werden können. Im Modul Dupe Remover werden die Spalten angegeben, die zum Auffinden von Duplikaten herangezogen werden sollen (Im Beispiel alle außer „sa_key“.). Mit Hilfe der Suche werden die Beispieldatensätze 5 und 6 als mögliche Duplikate erkannt. In weiterer Folge können die erkannten Duplikate zusammengeführt bzw. gelöscht werden.

[12] vgl. [Ora06, 21-1ff]

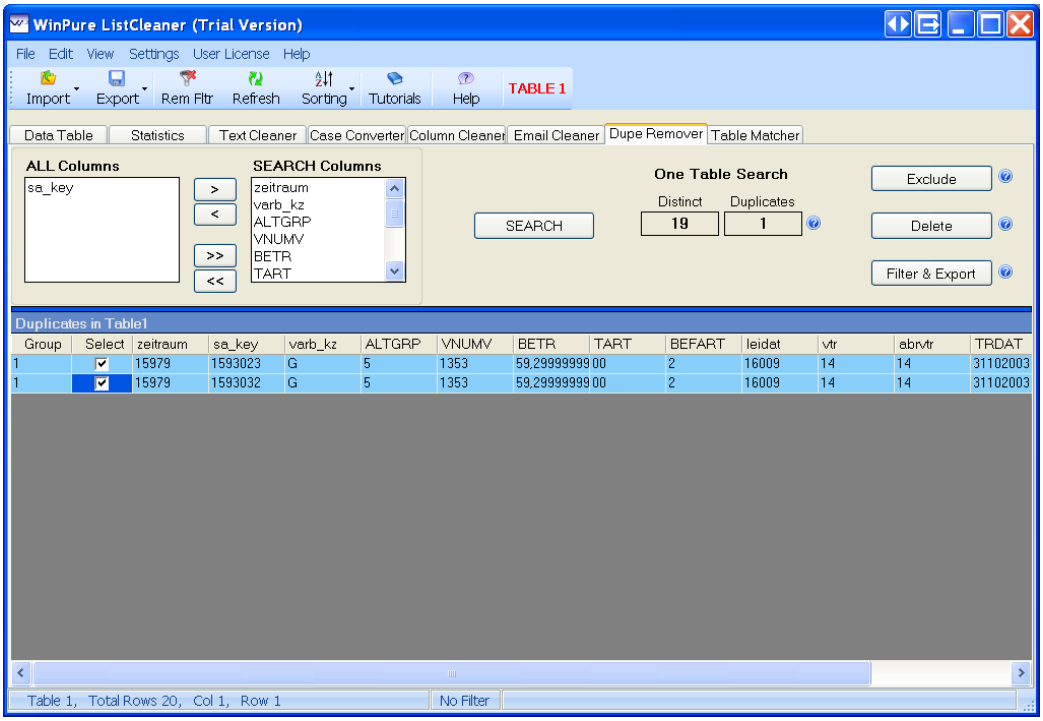

Abbildung 36: WinPure ListCleaner Pro - Beispiel Dupe Remover

- **WizRule®**
  Wizrule ist in der Lage, Wenn-Dann-Beziehungen zwischen Attributwerten zu erkennen, mathematische Zusammenhänge zwischen Feldern als Formeln auszudrücken oder die Rechtschreibung von Attributwerten über gesamte Tupelmengen hinweg auf Konsistenz zu testen. Es ist hingegen nicht möglich, die genannten Funktionen so zu konfigurieren, dass das Werkzeug doppelt vorhandene Datensätze aufzuspüren vermag. Man würde dazu eine Operation benötigen, die auf Basis einer Menge von Attributwerten (d.h. von mehreren Attributwerten eines Tupels) die Ähnlichkeit von Tupel-Paaren bestimmt. Daher ist WizRule® nicht für die Behebung von Duplikaten geeignet.

7.2.2 Beispieldaten falsche Tupel

nr	zeitraum	sa_key	ALT GRP	VNUMV	BETR	TART	BE FART	leidat	vtr	abrvtr	TRDAT	gesl	mwst bet
17	15949	1593241	0	2092	59,3	00	2	16024	14	14	15112003	W	5,93
19	15979	1593272	2	3045	59,3	02	2	16054	14	15	15122003	M	5,93
20	15979	1593362	6	1257	59,3	00	2	16027	14	14	18112003	M	5,93

Tabelle 35: Beispieldaten - falsche Tupel

- **SSIS**
  SSIS stellt mit der Fuzzysuche eine Funktion zur Verfügung, die auf Grund von gespeicherten Referenzwerten die Daten dahingehend überprüft, ob zulässige Werte verarbeitet werden. Bedingung dafür ist, dass die benötigten Referenzwerte bekannt sind. Diese werden in einer Verweisdatenquelle gespeichert. In SSIS definiert der Benutzer einen Operator „Transformation für Fuzzysuche“ auf die ausgewählte Tabelle „XY“. In einem weiteren Schritt müssen die Einstellungen für die drei unter Kapitel 6.2.2.1 beschriebenen Funktionen (Maximale Suche nach Übereinstimmungen pro Eingabezeile, Suche nach kleineren Einheiten mittels Token-Trennzeichen, Schwellenwerte für Ähnlichkeit) vorgenommen werden, sodass in einem nächsten Schritt die Überprüfung auf die Referenzwerte durchgeführt und das Ergebnis gespeichert werden kann. Eine genauere Beschreibung des Arbeitsablaufes findet sich unter „Transformation für Fuzzysuche“[13]. Die Beispieldatensätze 17, 19 und 20 werden als mögliche Fehler erkannt, da für mögliche Werte von „zeitraum“, „vtr“ und „abrvtr“ Referenzwerte eingetragen sind (zeitraum = 15979, vtr = 14 und abvtr = 14). Qualitätsindikatoren in Form von SQL-Abfragen können dabei behilflich sein, einen Überblick über die Daten zu gewinnen (z.B. Anzahl der Datensätze). Jedoch können sie keine Problemlösung vornehmen.

- **Oracle**
  Oracle bietet mit selbstdefinierten Regeln ebenfalls die Möglichkeit abzuprüfen, ob „korrekte“ Werte verarbeitet werden. Bedingung dafür ist, dass die „korrekten“ Werte bekannt sind. Hierbei wird mittels einer selbstdefinierten SQL-Abfrage, welche auf die zu überprüfende Tabelle angewendet wird, herausgefunden, ob Attributwerte in der Referenztabelle vorhanden sind. Die Beispieldatensätze 17, 19 und 20 werden als mögliche Fehler erkannt, da für mögliche Werte von „zeitraum“, „vtr“ und „abrvtr“ Referenzwerte bekannt sind (zeitraum = 15979, vtr = 14 und abvtr = 14).

- **SAS**
  Auch in SAS kann mit Hilfe des Matchcodes (entspricht den Referenzwerten) überprüft werden, ob „korrekte“ Werte verarbeitet werden. Hierbei findet sich eine ähnliche Vorgehensweise wie unter Kapitel 7.2.1 beschrieben. Es ist jedoch unbedingt notwendig, dass das Sensitivitätslevel auf 1 gesetzt ist, sodass nur genaue Übereinstimmungen akzeptiert werden. Die Beispieldatensätze 17, 19 und 20 werden als mögliche Fehler erkannt, da für mögliche Werte von „zeitraum“, „vtr“ und „abrvtr“ „korrekte“ Werte bekannt sind (zeitraum = 15979, vtr = 14 und abvtr = 14).

[13] vgl. http://msdn.microsoft.com/de-de/library/ms137786.aspx

- **WinPure**
  WinPure stellt kein Modul zur Verfügung, welche so konfiguriert werden kann, dass das Werkzeug falsche Datensätze aufzuspüren vermag. Man würde dazu eine Operation benötigen, die auf Basis einer Menge von Attributwerten feststellen kann, ob diesen Attributwerten ein Wert in der realen Welt zu Grunde liegt. Daher ist WinPure nicht für die Findung falscher Tupel geeignet.

- **WizRule®**
  WizRule® stellt keine Funktion zur Verfügung, welche so konfiguriert werden kann, dass das Werkzeug falsche Datensätze aufzuspüren vermag. Man würde dazu eine Operation benötigen, die auf Basis einer Menge von Attributwerten feststellen kann, ob diesen Attributwerten ein Wert in der realen Welt zu Grunde liegt. Daher ist WizRule® nicht für die Findung falscher Tupel geeignet.

7.2.3 Beispieldaten fehlende Tupel

- **SSIS**
  Qualitätsindikatoren in Form von SQL-Abfragen können dabei behilflich sein, einen Überblick über die Daten zu gewinnen (z.B. Anzahl der Datensätze), um festzustellen, ob genügend Daten vorhanden sind. Jedoch können sie keine Problemlösung vornehmen. Es sind Funktionen nötig, die Zugriff auf den ETL-Prozess haben, um überprüfen zu können, ob die „richtige“ Anzahl von Tupel geladen wurde.

- **Oracle**
  Oracle stellt mit der Möglichkeit Regeln selbst zu definieren, einen Weg zur Verfügung, die Daten zu analysieren, um feststellen zu können, ob genügend Daten geladen worden sind. Diese Regeln bestehen in Form von SQL-Abfragen welche auf die zu analysierendenn Daten angewendet werden. Jedoch können sie keine Problemlösung vornehmen. Es sind Funktionen nötig, die Zugriff auf den ETL-Prozess haben, um überprüfen zu können, ob die „richtige“ Anzahl von Tupel geladen wurde.

- **SAS, WinPure und WizRule®**
  Die genannten Werkzeuge stellen keine Funktion zur Verfügung, welche so konfiguriert werden kann, dass die Werkzeuge fehlende Tupel aufzuspüren vermögen. Man würde dazu zumindest eine Operation benötigen, die auf Basis einer Anzahl von Attributwerten feststellen kann, ob die „richtige“ Anzahl von Tupel geladen wurde. Des Weiteren sind Funktionen nötig, die Zugriff auf den ETL-Prozess haben, um überprüfen zu können, ob die „richtige“ Anzahl von Tupel geladen wurde.

7.2.4 Beispieldaten sonstige Fehler

nr	zeitraum	sa_key	ALT GRP	VNUMV	BETR	TART	BE FART	leidat	vtr	abrvtr	TRDAT	gesl	mwst bet
2	15979	1641521	0	2897	28,3	07	1	16000	14	14	22102003	w	5,83
20	15979	1593362	6	1257	59,3	00	2	16027	14	l4	18112003	M	5,93

Tabelle 36: Beispieldaten - sonstige Fehler

Fehlernummer	Datensatz	Spalte	Fehlerausprägung	richtiger Wert
1	2	gesl	w (Kleinschreibung)	W
2	2	mwstbet	5,83 (Falscher Wert)	2,83
3	20	abrvtr	l4 („l“ statt 1)	14

Tabelle 37: Beispieldaten - Fehlerbeschreibung

Alle fünf Werkzeuge stellen Funktionen zur Verfügung, welche sich mit dem Aufbau und der Umformung von Attributwerten beschäftigen. Durch diese Funktionen wird die allgemeine Erkennung von Schreib-, Format- und Abhängigkeitsfehlern erleichtert. Sie stellen aber keine Grundvoraussetzung für die Funktionalität des gewünschten Werkzeuges für die OÖGKK dar. Dennoch bieten sie interessante Möglichkeiten, um eine weitere Verbesserung der Datenqualität vornehmen zu können. Nachstehend findet sich eine Aufstellung der zur Verfügung gestellten Funktionen je Werkzeug, welche die Möglichkeit bieten, sonstige Fehler zu erkennen und gegebenenfalls zu bereinigen (siehe Tabelle 38).

- **SSIS**
  SSIS stellt mit der mathematischen Regelfindung eine Funktion zur Verfügung, mit der überprüft werden kann, ob die Spalte „mwstbet“ den „richtigen“, in Abhängigkeit zur Spalte „BETR“ berechneten, Mehrwertsteuersatz enthält. Zur Überprüfung kann dieser berechnet werden und gegebenfalls gespeichert werden.

  Weiters stellt SSIS folgende Funktionen zur Behebung sonstiger Fehler zur Verfügung: Datentyp- sowie Formaterkennung und Formatumwandlung, Transformationen zur Bearbeitung von Spalten, Extrahierung von Daten, mathematische Regelfindung und Qualitätsindikatoren.

- **Oracle**
  Oracle ist durch die Anwendung der Funktion „UPPER“ auf die Spalte „gesl“ in der Lage die geforderte Einheitsschreibweise herzustellen. Weiters ist es Oracle möglich, durch die Abhängigkeitsüberprüfung auf die Spalten „BETR“ und „mwstbet“ festzustellen, dass beim Beispieldatensatz 2 in der Spalte „mwst“ nicht die „üblichen“ 10% Mehrwertsteuer eingetragen sind.

Weiters stellt Oracle folgende Funktionen zur Behebung sonstiger Fehler zur Verfügung: Abhängigkeitsüberprüfung, Datentyp- sowie Formaterkennung und Formatumwandlung, Transformationen für alphanumerische Zeichen, Transformationen zur Bearbeitung von Spalten, Extrahierung von Daten, Aggregationsbildung, Qualitätsindikatoren und selbstdefinierte Regeln.

- **SAS**
  SAS stellt mit der DQCASE Funktion die Möglichkeit zur Verfügung die geforderte Einheitsschreibweise in der Spalte „gesl“ (siehe Fehlerdatensatz 2) sicherzustellen. Mit Anwendung dieser Funtkion auf einzelne Spalten ist die Einhaltung und Sicherstellung einer einheitlichen Schreibweise (z.B. Großbuchstaben) leicht durchführbar.

  SAS stellt folgende Funktionen zur Behebung sonstiger Fehler zur Verfügung: Transformationen für alphanumerische Zeichen, Transformationen zur Bearbeitung von Spalten und Extrahierung von Daten.

- **WinPure**
  WinPure stellt mit dem Modul Case Converter die möglichkeit zur Verfügung die Groß- und Kleinschreibung zu vereinheitlichen. Mit Hilfe dieses Modules ist es einfach das kleingeschriebene „w“ in Datensatz 2 auf das korrekte„W“ umzuformulieren.

  Winpure stellt folgende Funktionen zur Behebung sonstiger Fehler zur Verfügung: Transformationen für alphanumerische Zeichen, Transformationen zur Bearbeitung von Spalten und E-mail Bereinigung.

- **WizRule®**
  Durch die Mathematische Regelfindung ist WizRule® in der Lage die Abweichung der Mehrwertsteuer, welche im Datensatz 2 nicht 10% beträgt, zu erkennen und diese nach erfolgter Regelberechnung im Deviation Report darzustellen.

  WizRule® stellt folgende Funktionen zur Behebung sonstiger Fehler zur Verfügung: Abhängigkeitsüberprüfung, mathematische Regelfindung.

7.3 Gegenüberstellung von Funktionsgruppen und Fehlerquellen

Für die Herstellung einer direkten Vergleichsbasis wurden ähnliche Funktionsweisen zusammengefasst um eine gröbere Gesamtgliederung zusammenzustellen. Tabelle 38 veranschaulicht diese Gegenüberstellung nach den zusammengefassten Funktionen und den in Kapitel 5.4 spezifizierten Fehlerquellen auf Grund der Erkenntnisse der Bearbeitung der Beispieldaten.

Fehlerquelle / Funktionsgruppe	SSIS	Oracle	SAS	WinPure	WizRule
1 Duplikate - Mehrfach-Datensätze					
• Duplikatensuche	x	x	x	x	
2 Falsche Tuple					
• Überprüfung von Referenzwerten	x		x		
• selbstdefinierte Regeln		x			
3 Fehlende Tuple - Daten fehlen					
• Qualitätsindikatoren	*				
• selbstdefinierte Regeln		*			
3 Fehlende Tuple - nicht als zu ladend erkannt					
• Qualitätsindikatoren	*				
• selbstdefinierte Regeln		*			
4 Sonstige Fehler					
• Abhängigkeitsüberprüfung		x			x
• Datentyp- sowie Formaterkennung und -umwandlung	x	x			
• Transformationen für Alphanumerische Zeichen		x	x	x	
• Transformationen zur Bearbeitung von Spalten	x	x	x	x	
• Extrahierung von Daten	x	x	x		
• Mathematische Regelfindung	x				x
• Aggregationsbildung		x			
• E-mail Bereinigung				x	
• Qualitätsindikatoren	x	x			
• selbstdefinierte Regeln		x			

Tabelle 38: Gegenüberstellung der aggregierten Funktionen

Jedes x bedeutet: ist vorhanden; während * bedeutet: ist vorhanden, aber nicht ausreichend.

Mit Hilfe dieser Tabelle 38 ist eine Beurteilung der Werkzeuge in Bezug auf die vorhandenen Fehlerquellen möglich, sodass eine Entscheidung über die weitere Vorgehensweise getroffen werden kann.

7.4 Beurteilung

Durch die Angaben in Tabelle 38 wird ein Überblick gegeben, in welcher Art und Weise Daten untersucht bzw. Fehler behoben werden können. Der Microsoft® SQL Server™ 2005 Integration Services sowie der Oracle® Warehouse Builder 10g Release 2 stellen sich als die beiden mächtigsten Werkzeuge heraus. Sie besitzen umfangreiche Analysemethoden und können die vorhandenen Daten auf verschiedenste Arten bearbeiten, sodass entdeckte Fehler behoben werden können.

Der SAS® 9.1.2 Data Quality Server bietet die Möglichkeit der Duplikatenanalyse und Überprüfung der Attributwerte anhand von Referenzwerten, in diesem Fall generierten Matchcodes. Des Weiteren ermöglicht der Data Quality Server die Extrahierung von einzelnen Subwerten in den Attributwerten. Dies erleichtert die Überprüfung auf Duplikate, da die einzelnen Datenwerte genauer überprüft werden können. Ein weiterer Vorteil den SAS® bietet, ist die Möglichkeit die verwendeten

Datei- und Speicherformate beizubehalten, ohne weitere Programme und dafür notwendige Lizenzgebühren aufbieten zu müssen.

Durch ihre Spezialisierungen sind WinPure ListCleaner Pro und WizRule® für die Anforderungen der OÖGKK nicht geeignet. WinPure ListCleaner Pro erfüllt durch die Fähigkeit Duplikate zu bereinigen zumindest einen geforderten Punkt, ist aber durch seine weiteren Fähigkeiten, gesamte Spalten und einzelne Datenwerte zu analysieren und zu verändern, als Werkzeug für die angegeben Anforderungen nicht geeignet. WizRule® eignet sich mit seiner Fähigkeit von Wenn-Dann Beziehungen zum Analysieren von Daten, ist aber auf Grund seiner eingeschränkten Fähigkeiten nicht empfehlenswert.

Mit Ausnahme des SAS® 9.1.2 Data Quality Server werden keine Schnittstellen zu den SAS-Tabellen (siehe Kapitel 5.3.1) bereitgestellt. Des Weiteren ist es notwendig, Schnittstellen zur Verfügung zu stellen, die die Warnungsmeldungen generieren, abspeichern und die Mitprotokollierung des Arbeitsfortschrittes gewährleisten.

Alle analysierten Werkzeuge bieten im Sinne der Mindestanforderungen der OÖGKK (siehe Kapitel 5.4) unzureichende Methoden (z.B. Arbeiten mit Referenz- und Grenzwerten), um falsche Datensätze (Fehlerquelle 2) und Fehlende Tupel (Fehlerquelle 3) erkennen und bearbeiten zu können (siehe Tabelle 38). Der SSIS bietet mit seinem Qualitätsindikator „Anzahl der Datensätze“ als einziges Werkzeug die Möglichkeit abschätzen zu können, wie viele Datensätze fehlen oder zu viel sind, sofern die Anzahl der zu importierenden Datensätze bekannt ist.

7.5 Schlussfolgerungen des Vergleichs

Durch die vorgenommene Aufstellung (siehe Tabelle 38) und der Zuhilfenahme der festgelegten Anforderungen (siehe Kapitel 5.4) kann kein einzelnes Werkzeug als alleinige Lösung angesehen werden. Auf Grund dessen stehen zwei alternative Möglichkeiten zur weiteren Vorgehensweise zur Verfügung. Zum einen besteht die Möglichkeit ein bestehendes Werkzeug auszuwählen und weiterzuentwickeln, zum anderen gibt es die Möglichkeit ein neues Werkzeug zu spezifizieren, das den Anforderungen der OÖGKK entspricht.

Bei den in Frage kommenden Werkzeugen handelt es sich um den Microsoft® SQL Server™ 2005 Integration Services sowie den Oracle® Warehouse Builder 10g Release 2. Beide bieten umfassende Möglichkeiten Daten zu analysieren bzw. zu bearbeiten. Dennoch werden gewünschte Mindestanforderungen (siehe Kapitel 5.3.2 und 5.4) von beiden Werkzeugen nicht erfüllt (z.B. SAS - Identifizierung und Bearbeitung von fehlenden Tupel; Oracle - Arbeiten mit Referenz- und Grenzwerten). Da beide eigenständige und umfangreiche Werkzeuge darstellen, ist eine Erweiterung der Funktionalität schwer durchführbar. In weiterer Folge ist eine Einbindung in die vorhandene Infrastruktur schwierig. Ebenso ist für den Betrieb dieser Systeme ein weiteres Lizenzaufkommen zu bestreiten.

Für die Eigenerstellung eines Werkzeuges sprechen folgende Punkte:

- Einarbeitung nur von notwendigen Funktionen.
- Leichte Erweiterbarkeit für zukünftige Problemstellungen.
- Abstimmung auf die vorhandene Infrastruktur.
- leichtere Wartungsmöglichkeit.
- bessere Benutzungsschnittstellen.
- Mitarbeiter kennen das Produkt.
- Bestehendes Lizenzaufkommen wird nicht erhöht.
- Einarbeitung in die bestehende Produktpalette.
- Abstimmung der Benutzeroberfläche.

Auf Grund dieser Analyse wird die Spezifikation eines neuen Werkzeuges empfohlen. In weiter Folge wird in Kapitel 8 diese Konzipierung vorgenommen. Die Implementierung und Umsetzung dieser Eigenlösung ist nicht mehr Teil dieser Diplomarbeit.

8 Spezifikation

In diesem Kapitel wird die Spezifizierung eines Werkzeuges für die OÖGKK vorgenommen. Dieses Werkzeug ist in der Lage, die in SAS-Tabellen vorliegenden Daten zu analysieren und bei festgestellten Fehlern Warnungsmeldungen zu generieren und abzuspeichern. In weiterer Folge werden diese Warnungsmeldungen mit Hilfe des firmeninternen Portals dargestellt. Die Warnungsmeldungen werden den Mitarbeitern der IT-Entwicklungsabteilung zugeteilt, sodass eine nähere Untersuchung der Daten angestoßen wird. In Anlehnung an Balzert [Bal98] wird das Werkzeug anhand folgender Punkte näher beschrieben:

- Name
- Einsatzbereich
- Zielbestimmung
  - Musskriterien
  - Sollkriterien
  - Kannkriterien
  - Abgrenzungskriterien
- Werkzeugeinsatz
  - Anwendungsbereiche
  - Zielgruppen
  - Betriebsbedingungen
- Werkzeugübersicht
- Prozessablauf
- Werkzeugfunktionen
- Qualitätsanforderungen
- Benutzeroberfläche
- Technische Werkzeugumgebung
  - Software
  - Hardware
  - Werkzeugschnittstellen
- Anforderungen an die Entwicklungsumgebung

8.1 Name

Das Werkzeug stellt eine „Software für die automatisierte Plausibilitätskontrolle“ der Ladeprozesse der OÖGKK dar. Zur besseren Referenzierbarkeit wird dieser Software das Akronym „SofaP“ zugewiesen.

8.2 Einsatzbereich

Der Einsatzbereich dieses Werkzeuges ist die OÖGKK. Sie wird durch die Abteilung IT-Entwicklung betreut. SofaP hat direkten Zugriff auf das SAS-DWH der OÖGKK. SofaP wird auf den Servern der OÖGKK installiert.

8.3 Zielbestimmung

Dem Benutzer wird mit Hilfe der implementierten Abfragen eine Analyse der importierten Daten ermöglicht. Beschreibungen der dadurch festgestellten Anomalien werden zur späteren Bearbeitung abgespeichert. Diese Informationen werden mit Hilfe des DWH-Portals ausgelesen und dargestellt, sodass dem Benutzer die Warnungsmeldungen dargestellt werden. Anschließend kann dieser die Warnungsmeldungen abarbeiten und notwendige Änderungen an den Daten durchführen.

8.3.1 Musskriterien

Für das Werkzeug unabdingbare Kriterien, die in jedem Fall erfüllt werden müssen:

- Zugriff auf die in der SAS-Datei gespeicherten importierten Daten.
- Bereitstellung von Methoden zur Erkennung von Duplikaten („duplicates“) und fehlenden Datensätzen („missing tuple“).
- Durchführung von vorgefertigten Abfragen auf die zu analysierenden Daten.
- Generierung von Warnungsmeldungen auf Grund der getätigten Analysen und Abfragen.
- Modularer Aufbau, um nachträgliche Erweiterungen leicht einbauen zu können.
- Speicherung von Referenzdaten.
- Berechnung von Grenzwerten auf Grund von parametrisierten Abfragen.
- Speicherung von Grenzwerten.

8.3.2 Sollkriterien

Die Erfüllung dieser Kriterien wird angestrebt.

- Bereitstellung von Methoden zur Erkennung falscher Datensätze („invalid tuple“).
- Anstoß der Datenanalyse im Anschluss an den Datenimport.
- Speicherung von Parametern für Abfragen.
- Änderung von Parametern für Abfragen.
- Abarbeitung von parametrisierten Abfragen.
- Interface zum Verwalten der Abfragen und Parameter.
- Graphisches User-Interface zum Eintragen und Warten der Referenz- und Grenzwerte.
- Festlegung und Speicherung von Qualitätsindikatoren.
- Abspeicherung und Darstellung der erstellten Fehlermeldungen im DWH-Portal.

8.3.3 Kannkriterien

Es wurden keine Kriterien gefunden.

8.3.4 Abgrenzungskriterien

Diese Kriterien sollen bewusst nicht erreicht werden.

- Durchführung von Überprüfungen zur Integritätsprüfung.
- Durchführung von Datenbereinigungsmethoden zur Format- und Datentypbereinigung.
- Durchführung von Transformationen für alphanumerische Zeichen und zur Bearbeitung von Spalten.
- Extrahierung von Daten aus einzelnen Attributwerten.
- Funktionen für eine E-mail Bereinigung.

8.4 Werkzeugeinsatz

8.4.1 Anwendungsbereiche

Das Werkzeug stellt eine Anwendung zur Plausibilitätskontrolle dar. Im Anschluss an den Ladevorgang aller Datensätze eines Ladezyklus (ETL-Prozess) und Integrierung in SAS-Dateien wird eine Analyse dieser Dateien vorgenommen. Entsprechende Warnungsmeldungen, die auf Grund der abgearbeiteten Abfragen generiert werden, sind in Dateien abzuspeichern. Das DWH-Portal liest diese Warnungsmeldungen aus diesen Dateien aus und stellt diese dann für die Benutzer dar.

8.4.2 Zielgruppen

Zielgruppe ist die Organisationseinheit IT-Entwicklung der OÖGKK. Eine weitere Zielgruppe sind die Betreuer des DWH-Portals, welche mit der Bearbeitung der Fehlermeldungen betraut werden.

8.4.3 Betriebsbedingungen

Das Werkzeug wird nach jedem vollständigen ETL-Prozess gestartet und arbeitet automatisierte Analysen und Abfragen ab. Die ermittelten Fehlermeldungen werden dem DWH-Portal zur Verfügung gestellt bis die entsprechende Bearbeitung abgeschlossen ist.

- Standort: Das Einsatzgebiet beschränkt sich auf die OÖGKK mit Standort in Linz.
- Betriebsdauer: Nach jedem Ladezyklus, jedoch Zurverfügungstellung der Fehlermeldungen bis eine Bearbeitung vorgenommen wurde.
- Nach erfolgreicher Testphase ist der Betrieb wartungsfrei.
- Der Betrieb von SofaP erfolgt auf Grund der Automatisierung und des umfangreichen Datenvolumens unbeaufsichtigt. Es wird jedoch eine Logdatei erstellt.
- Ausführung erfolgt nur durch autorisierte Benutzer oder Administratoren. Daher ist kein Berechtigungssystem vorzusehen.

8.5 Werkzeugübersicht

Das Werkzeug ist in der Lage auf die importierten Daten, welche in einer SAS-Datei abgelegt sind, zuzugreifen. In weiterer Folge können Abfragen und Analysen, welche im Werkzeug definiert sind, auf die SAS-Daten abgesetzt werden. Hierbei ist es möglich auf gespeicherte Referenzwerte und Grenzwerte zuzugreifen, um mit diesen die Abfragen mit entsprechender Parametrisierung auszuführen. Dies

bedeutet, dass bei den Abfragen eingetragene Platzhalter entsprechend ersetzt werden.

Werden Datensätze gefunden, die die vorgegebenen Werte nicht einhalten bzw. als fehlerhaft eingestuft werden, so werden diese Datensätze vorgemerkt. Eine Hinweismeldung mit der entsprechenden Fehlerbezeichnung und Abweichungsdarstellung wird an das DWH-Portal übertragen. Mit Hilfe des DWH-Portals werden die Fehlermeldungen graphisch aufbereitet und den zuständigen Personen angezeigt, sodass diese Handlungen setzen können.

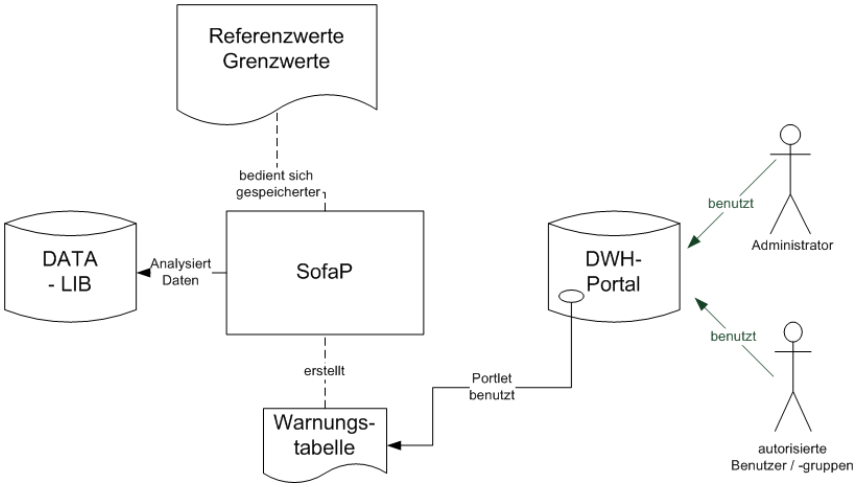

Abbildung 37: Graphische Darstellung Werkzeugübersicht

Im Folgenden werden einige zuvor verwendete Begriffe beschrieben und ihr Aufbau genauer erläuert:

Grenzwert

Grenzwerte stellen Unter- bzw. Obergrenzen von Werten dar, in denen sich diese befinden müssen. Auf Grund von Abweichungen können aber auch Werte außerhalb dieser Grenzen auftreten. Diese stellen Anomalien dar und werden mittels Warnungsmeldungen an den Benutzer weitergegeben. Hierbei wird die genaue Abweichung berechnet: bei Werten oberhalb des Maximalwertes: tatsächlicher Wert - Maximalwert = Abweichung; bei Werten unterhalb des Minimalwertes: Minimalwert - tatsächlicher Wert = Abweichung.

Referenzwert

Referenzwerte stellen einen Wert oder eine Auswahl von Werten dar, welche von dem zu untersuchenden Attributwert angenommen werden müssen. Dieser Referenzwert muss entweder vom Datentyp Char oder Datentyp Integer sein.

Unterschied berechneter und eingetragener Werte

In einigen Fällen ist es sinnvoll, keinen absoluten Referenz- bzw. Grenzwert einzutragen, sondern sich diesen aus vorhandenen, historischen Datensätzen berechnen zu lassen, um diesen dann einsetzen zu können. Hierbei können z.B. Durchschnittswerte von Spaltenwerten berechnet werden. Durch eine zusätzliche Berechnung der Standardabweichung werden unter Zuhilfenahme einer in den Referenzwerten abgespeicherten Zahl die Minimum- und Maximalwerte berechnet: Maximalwert = Durchschnittswert + (Referenzzahl * Standardabweichung). Analog ist dies für den Minimumwert zu handhaben: Minimum = Durchschnittswert - (Referenzzahl * Standardabweichung). Die Berechnung der Werte erfolgt über SQL-Statements, welche auf historischen Werten ausgeführt werden.

Warnungen

Wird eine Anomalie in den untersuchten Daten festgestellt, so wird eine Warnungsmeldung erstellt. Diese erstellten Warnungen werden in einer Datei gesammelt. Eine Warnung enthält eine eindeutige Warnungsnummer, den genauen Datensatz bzw. die Anzahl der verdächtigen Datensätze, die betreffende Abfrage sowie eine Fehlermeldung. Die Fehlermeldung setzt sich aus dem Attributnamen, dem Attributwert und der Abweichung sowie einer kurzen textuellen Fehlerbeschreibung zusammen.

Log-Datei

Die Log-Datei protokolliert alle Arbeitsschritte des Programmes mit und schreibt diese in eine Datei bzw. gibt diese während der Abarbeitung gegebenenfalls auch über das graphische User-Interface an den Benutzer weiter. Eine Log-Zeile enthält einen Zeitstempel, die jeweilige Operation, welche gerade ausgeführt wird, sowie den Status dieser Operation.

Beispiel der Log-Datei:

2008-07-30 12:28:28	Verbindungsherstellung zur Datenbank	OK
2008-07-30 12:28:30	Verbindungsherstellung zur SAS-Datei	OK
2008-07-30 12:28:32	Beginn Analyse Tabelle SA30	OK

Parameter

Parameter stellen eingesetzte Werte bei Where-Klauseln von SQL-Statements dar (z.B. WHERE gesl = [Parameter]). Diese Parameter werden in Form von Referenzwerten abgespeichert.

Graphische Aufbereitung der erläuterten Begriffe

Log-Datei	Referenzwert	Grenzwert	Warnungsmeldung
Zeitstempel	Referenzwertnummer	Grenzwertnummer	Warnungsmeldungsnummer
durchgeführte Operation	Referenzwertname	Grenzwertname	Tabelle
Status	Attributwert	untere Grenze	Abfragename
	(Char / Integer)	obere Grenze	Datensatz
	Abfragename	Abfragename	Attributname
	Beschreibung	Beschreibung	Attributwert
			berechnete Abweichung
			textuelle Fehlerbeschreibung

Tabelle 39: Übersicht: Log-Datei, Referenzwert, Grenzwert und Warnungsmeldung

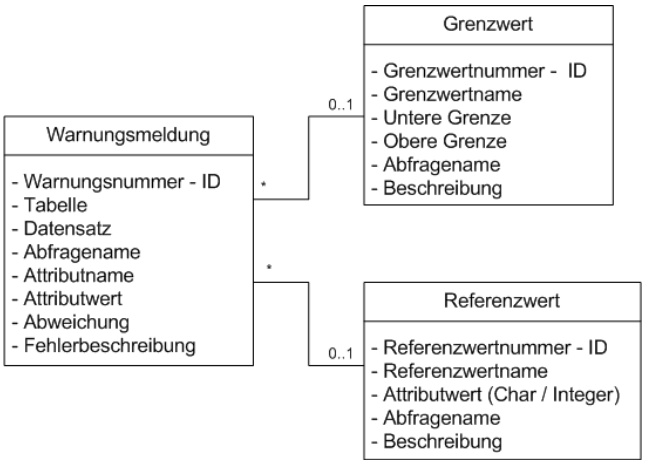

Abbildung 38: UML-Darstellung: Grenzwert, Referenzwert und Warnungshinweise

8.6 Prozessablauf

Der Prozessablauf von SofaP gestaltet sich folgendermaßen:

1. Herstellung der DB-Verbindung.
2. Herstellung Verbindung SAS-Datei.
3. Analyse der SAS-Datei.
4. Schließung der DB-Verbindung.

Da auf eine SAS-Datei eine oder mehrere Analysen gefahren werden, ist es möglich nach einer abgeschlossenen Analyse eine neue auf dieselbe Datei zu starten, eine neue SAS-Datei zu analysieren oder das Programm zu beenden.

Der Analysevorgang einer SAS-Datei gestaltet sich folgendermaßen:

In einem ersten Schritt werden auf Grund eines SQL-Statements ein Datensatz bzw. ein Set von Datensätzen berechnet. Im Anschluss werden die zugehörigen Referenz- oder Grenzwerte ermittelt. Im nächsten Schritt werden die Attributwerte des Datensatzes ausgelesen und mit den zuvor ermittelten Referenz- oder Grenzwerten verglichen. Wird eine Abweichung festgestellt, so wird eine Warnungsmeldung erstellt und abgespeichert. Wurde ein Set von Datensätzen zurückgegeben, wird der nächste Datensatz bearbeitet. Ansonsten ist diese Analyse abgeschlossen.

Abbildung 39 veranschaulicht dies noch einmal graphisch.

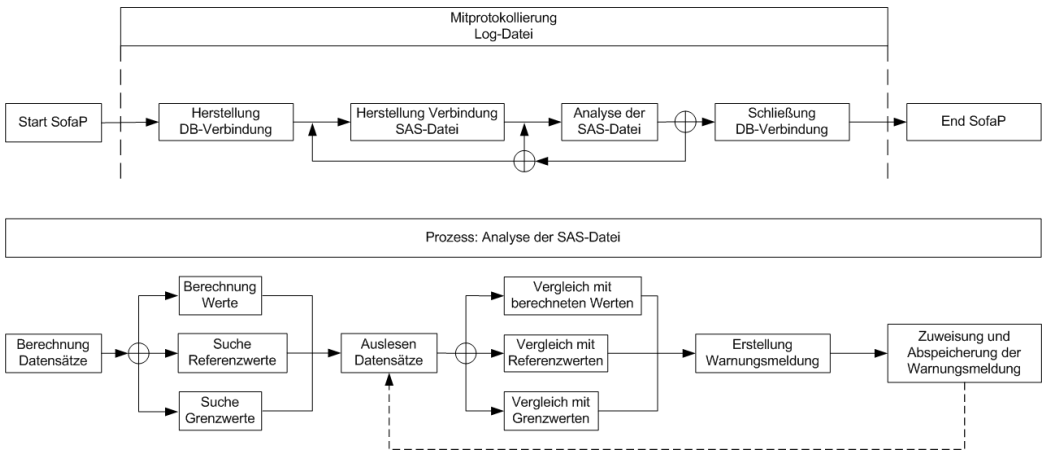

Abbildung 39: Prozessablauf SofaP

8.7 Werkzeugfunktionen

Das Programm unterstützt folgende Funktionen:

Allgemeine Funktionen

- Zur Analyse der Daten muss eine Datenbankverbindung hergestellt werden (Kapitel 8.7.1).
- Zur Analyse der Daten muss eine Verbindung zu der gewünschten SAS-Datei hergestellt werden (Kapitel 8.7.2).
- Nach Abschluss aller Transaktionen muss die Datenverbindung wieder geschlossen werden (Kapitel 8.7.3).
- Zur Nachverfolgung aller durchgeführten Aktivitäten muss eine Log-Datei verfasst werden (Kapitel 8.7.4).

Funktionen zur Datenanalyse und Fehlerfindung

- Zur Übersicht über die Anzahl einzelner Datensatzgruppen wird diese auf Grund von angegebenen Parametern berechnet (Kapitel 8.7.5).
- Wurden keine mathematischen Referenz- und Grenzwerte eingetragen, so werden diese ad hoc berechnet (Kapitel 8.7.6).
- Um Datensätze überprüfen zu können, muss das Auslesen von Datensätzen auf Grund spezieller Parameter ermöglicht werden (Kapitel 8.7.7).
- Zum Überprüfen der Datensätze müssen entsprechende Referenzwerte ausgelesen werden (Kapitel 8.7.8).
- Zum Überprüfen der Datensätze müssen entsprechende Grenzwerte ausgelesen werden (Kapitel 8.7.9).
- Zum Auffinden von Warnungen werden die Attributwerte eines Datensatzes mit festgelegten Referenzwerten verglichen (Kapitel 8.7.10).
- Zum Auffinden von Warnungen werden die Attributwerte eines Datensatzes mit festgelegten Grenzwerten verglichen (Kapitel 8.7.11).
- Zum Auffinden von Warnungen werden die Attributwerte eines Datensatzes mit ad hoc berechneten Referenz- bzw. Grenzwerten verglichen (Kapitel 8.7.12 und 8.7.13).

Funktionen zur Warnungsverarbeitung

- Wurde ein fehlerhafter Datensatz identifiziert, muss eine darauf abgestimmte Warnungsbeschreibung generiert werden. Der gefundenen Anomalie muss eine eindeutige Warnungsnummer zugewiesen werden. Abschließend werden die Warnungsdaten abgespeichert (Kapitel 8.7.14).
- Abgearbeitete Warnungsmeldungen müssen aus der Datenhaltung der Warnungen gelöscht werden (Kapitel 8.7.15).

Es folgt eine genaue und detaillierte Beschreibung der einzelnen Werkzeugfunktionen.

8.7.1 Verbindungsherstellung zur Datenbank

Funktionalität:	Aufbau einer Verbindung zur Datenbank.
Eingabedaten:	• Zeichenkette für den Standort der Servers. • Zeichenkette für den Benutzer. • Zeichenkette für das Passwort. • Zeichenkette für die gewünschte Tabelle.
Ausgabedaten:	Der Benutzer wird über eine erfolgreiche Anmeldung informiert.
Fehlerquellen:	• Die Anmeldung wurde nicht durchgeführt. • Die angegebene Tabelle konnte nicht gefunden werden.

8.7.2 Verbindungsherstellung zur SAS-Datei

Funktionalität:	Das Werkzeug stellt eine Verbindung zur SAS-Datei her.
Eingabedaten:	• Zeichenkette für die gewünschte SAS-Datei. • Zeichenkette für die gewünschte Tabelle.
Ausgabedaten:	Der Benutzer wird über einen erfolgreichen Zugriff informiert.
Fehlerquellen:	• Die Anmeldung wurde nicht durchgeführt. • Die angegebene Tabelle konnte nicht gefunden werden. • Auf die angegebene Tabelle konnte nicht zugegriffen werden.

8.7.3 Schließung aller Datenbankverbindungen

Funktionalität:	Das Werkzeug schließt alle vorhanden Datenbankverbindungen.
Eingabedaten:	keine
Ausgabedaten:	Der Benutzer wird über eine erfolgreiche Schließung informiert.
Fehlerquellen:	• Es besteht keine Verbindung zur Datenbank. • Die Verbindung wurde nicht geschlossen.

8.7.4 Schreiben der Log-Datei(en)

Funktionalität:	Zur Nachverfolgung und Kontrolle der durchgeführten Aktivitäten wird eine Log-Datei verfasst. Eine Protokollzeile besteht aus einem Zeitstempel, dem dazugehörigen Ereignis sowie des Status.
Eingabedaten:	• Zeitstempel. • Ereignis. • Status.
Ausgabedaten:	Der Benutzer wird über erfolgreiches Schreiben, Speichern und Schließen der LOG-Datei(en) informiert.
Fehlerquellen:	• Log-Datei kann nicht angelegt werden. • Daten können nicht in die Log-Datei geschrieben werden. • Zeitstempel kann nicht generiert werden. • Log-Datei kann nicht gespeichert werden.

8.7.5 Berechnen Anzahl Datensätze auf Grund von Parametern

Funktionalität:	Diese Funktion berechnet die Anzahl von Datensätzen, die die angegebenen Parameter erfüllen. Es können hierbei ein, mehrere oder kein Parameter überprüft werden. Wird kein Parameter übergeben, so stellt der berechnete Wert in Bezug zu früheren Importvorgängen einen Qualitätsindikator dar, ob im Vergleich zu früheren Importvorgängen zu viele oder zu wenig Datensätze importiert wurden.
Eingabedaten:	• Datensätze aus der SAS-Datei. • Parameter.
Ausgabedaten:	Integerwert der Anzahl der Datensätze, die auf Grund der mitgegeben Parameter ermittelt wurde.
Fehlerquellen:	Abfrage kann auf Grund von Namenskonflikten nicht ausgeführt werden.

8.7.6 Berechnen von Werten (Grenz- und Referenzwerte)

Funktionalität:	Diese Funktion berechnet mit Hilfe eines SQL-Statements Werte zu einzelnen Spalten der SAS-Datei. Es handelt sich hierbei um Minimum- und Maximalwerte, Durchschnittswerte sowie Standardabweichung.
Eingabedaten:	• Datensätze aus der SAS-Datei. • SQL-Statement.
Ausgabedaten:	Integerwert des berechneten Grenz- oder Referenzwertes.
Fehlerquellen:	Die angegebene Spalte enthält keine Integerwerte. Es können keine mathematischen Werte berechnet werden.

8.7.7 Auslesen von Datensätzen auf Grund spezieller Parameter

Funktionalität:	Diese Funktion entnimmt die einzelnen Attributwerte eines Datensatzes aus der SAS-Datei, um sie auf Grund der Referenz- bzw. Grenzwerte zu analysieren. Die Datensätze werden auf Grund keines, eines oder mehrerer Parameter selektiert. Rückgabe eines Datensatzes oder eines Sets von Datensätzen. Es wird ein Datensatz nach dem anderen analysiert. Werden Abweichungen festgestellt, wird eine Warnungsmeldung generiert.
Eingabedaten:	• Datensätze aus der SAS-Datei. • Parameter.
Ausgabedaten:	Einzelner Datensatz und seine Attributwerte aus der SAS-Datei.
Fehlerquellen:	• Daten können nicht aus der SAS-Datei entnommen werden. • Abfrage kann auf Grund von Namenskonflikten nicht ausgeführt werden.

8.7.8 Auffinden entsprechender Referenzwerte

Funktionalität:	Diese Funktion durchsucht die Liste aller Referenzwerte und gibt den identifizierten Referenzwert zurück.
Eingabedaten:	Parameter zur Identifizierung des Referenzwertes.
Ausgabedaten:	Identifizierter Referenzwerte.
Fehlerquellen:	• Es kann kein Referenzwert gefunden werden. Liste ist leer bzw. der gesuchte Referenzwert ist nicht vorhanden. • Auf die Liste kann nicht zugegriffen werden (existiert nicht).

8.7.9 Auffinden entsprechender Grenzwerte

Funktionalität:	Diese Funktion durchsucht die Liste aller Grenzwerte und gibt die identifizierten Grenzwerte zurück.
Eingabedaten:	Parameter zur Identifizierung des Grenzwertes.
Ausgabedaten:	Identifizierter Grenzwert bzw. identifizierte Grenzwerte.
Fehlerquellen:	• Es werden keine Grenzwerte gefunden. Liste ist leer bzw. die gesuchten Grenzwerte sind nicht vorhanden. • Auf die Liste kann nicht zugegriffen werden (existiert nicht).

8.7.10 Vergleich Datensatz - Referenzwerte

Funktionalität:	Diese Funktion dient dazu, die ausgewählten Attributwerte eines Datensatzes mit den zuvor ermittelten Referenzdaten (Kapitel 8.7.8) zu vergleichen. Falls vorhanden wird eine Abweichung vom Referenzwert registriert. Bei Textfeldern handelt es sich um Duplikate. Bei Integerwerten werden die festgesetzten Werte nicht eingehalten.
Eingabedaten:	• Referenzwert • Datensatz aus der SAS-Datei
Ausgabedaten:	• Bei Char-Datentypen handelt es sich um Duplikate - Ausgabe der Datensatznummer. • Bei Integer-Datentypen wird die Abweichung (nach oben bzw. unten) berechnet und zurückgegeben. Die Datensatznummer wird ebenfalls zurückgegeben. • Wird keine Abweichung festgestellt, so wird NULL zurückgegeben.
Fehlerquellen:	• Es kann kein Referenzwert gefunden werden. Liste ist leer bzw. der gesuchte Referenzwert ist nicht vorhanden. • Auf die Liste kann nicht zugegriffen werden (existiert nicht). • Die Abweichung kann nicht berechnet und zurückgegeben werden. • Der übergebene Datensatz enthält keine Attribute.

8.7.11 Vergleich Datensatz - Grenzwerte

Funktionalität:	Diese Funktion dient dazu, die ausgewählten Attributwerte eines Datensatzes mit den zuvor herausgefilterten Grenzwerten zu vergleichen. Falls vorhanden wird eine Unter- bzw. Überschreitung der Grenzwerte registriert. Das Ausmaß der Unter- bzw. Überschreitung wird berechnet und zusammen mit der Datensatznummer zurückgegeben.
Eingabedaten:	• Grenzwerte. • Datensatz aus der SAS-Datei.
Ausgabedaten:	• Es wird die Abweichung nach oben bzw. unten berechnet und zurückgegeben. Die Datensatznummer wird ebenfalls zurückgegeben. • Wird keine Abweichung festgestellt, so wird NULL zurückgegeben.
Fehlerquellen:	• Es werden keine Grenzwerte gefunden. Liste ist leer bzw. der gesuchte Grenzwert ist nicht vorhanden. • Auf die Liste kann nicht zugegriffen werden (existiert nicht). • Die Abweichung kann nicht berechnet und zurückgegeben werden. • Der übergebene Datensatz enthält keine Attribute.

8.7.12 Vergleich Datensatz - berechnete Referenzwerte

Funktionalität:	Diese Funktion dient dazu, die ausgewählten Attributwerte eines Datensatzes mit den ad hoc berechneten Referenzwerten zu vergleichen. Werden die Referenzwerte nicht eingehalten, so wird das Ausmaß der Über- bzw. Unterschreitung berechnet und zusammen mit der Datensatznummer zurückgegeben.
Eingabedaten:	• Berechnete mathematische Referenzwerte. • Datensatz aus der SAS-Datei.
Ausgabedaten:	• Es wird die Abweichung nach oben bzw. unten berechnet und zurückgegeben. Die Datensatznummer wird ebenfalls zurückgegeben. • Wird keine Abweichung festgestellt, so wird NULL zurückgegeben.
Fehlerquellen:	• Es werden keine Referenzwerte berechnet. Liste ist leer bzw. der gesuchte Referenzwert ist nicht vorhanden. • Die Abweichung kann nicht berechnet und zurückgegeben werden. • Der übergebene Datensatz enthält keine Attribute.

8.7.13 Vergleich Datensatz - berechnete mathematische Grenzwerte

Funktionalität:	Diese Funktion dient dazu, die ausgewählten Attributwerte eines Datensatzes mit den ad hoc berechneten mathematischen Grenzwerten zu vergleichen. Werden die Grenzwerte nicht eingehalten, so wird das Ausmaß der Über- bzw. Unterschreitung berechnet und zusammen mit der Datensatznummer zurückgegeben.
Eingabedaten:	• Berechnete mathematische Grenzwerte. • Datensatz aus der SAS-Datei.
Ausgabedaten:	• Es wird die Abweichung nach oben bzw. unten berechnet und zurückgegeben. Die Datensatznummer wird ebenfalls zurückgegeben. • Wird keine Abweichung festgestellt, so wird NULL zurückgegeben.
Fehlerquellen:	• Es werden keine Grenzwerte berechnet. Liste ist leer bzw. der gesuchte Grenzwert ist nicht vorhanden. • Die Abweichung kann nicht berechnet und zurückgegeben werden. • Der übergebene Datensatz enthält keine Attribute.

8.7.14 Zuweisen und Abspeichern der Warnungsmeldung

Funktionalität:	Diese Funktion nimmt die Ausgaben der Funktionen 8.7.10 bis 8.7.13. Bei Integerwerten wird die berechnete Abweichung in die Fehlermeldung eingebracht. Bei Char-Attributwerten wird die Duplikatenfehlermeldung eingetragen. Weiters wird der festgestellten Anomalie eine eindeutige Warnungsnummer (Integer) zugewiesen. Anschließend werden die Datensatznummer oder die Anzahl der identifizierten Datensätze, die Warnungsnummer und die Warnungsmeldung abgespeichert.
Eingabedaten:	Ausgabewerte der Funktionen 8.7.10 bis 8.7.13
Ausgabedaten:	• Datensatznummer des betreffen Datensatzes, oder Anzahl der identifizierten Datensätze. • Eindeutige Fehlernummer (Integerwert). • Warnungsmeldung.
Fehlerquellen:	• Die Funktionen 8.7.10 bis 8.7.13 liefern keine Rückgabewerte. • Die Daten können nicht abgespeichert werden.

8.7.15 Löschen bearbeiteter Warnungsmeldungen

Funktionalität:	Durch die Benutzer (im DWH-Portal) bearbeitete und zur Kenntnis genommene Fehlermeldungen, welche an das Werkzeug zurückgemeldet wurden, werden aus der Liste der zu bearbeiteten Warnungsmeldungen gelöscht.
Eingabedaten:	Warnungsmeldungsnummer aus dem DWH-Portal.
Ausgabedaten:	Bereinigte Liste der Warnungsmeldungen - es sind nur offene Warnungsmeldungen gespeichert.
Fehlerquellen:	Die Warnungsmeldung konnte nicht gelöscht werden.

8.8 Qualitätsanforderungen

Funktionalität

Gefordert wird ein Werkzeug, das automatisch nach Beendigung des Datenimportes seine Prüfroutinen startet und die Auswertungen durchführt. Erkannte Unregelmäßigkeiten werden gekennzeichnet, abgespeichert und an das DWH-Portal

weitergeleitet, sodass die Informationen den Benutzern zugänglich gemacht werden.

Zuverlässigkeit

Jede Komponente soll 98% der Laufzeit zur Verfügung stehen. Für eine robuste Anwendung muss besonders Augenmerk auf das Fehler- und Exceptionhandling gelegt werden.

Bedienbarkeit

Es ist vorerst keine graphische Oberfläche für Änderungen in den Prüfroutinen bzw. Änderung der Referenz- und Grenzwerte vorgesehen. Diese kann als „nice-to-have“ Kriterium in einer weiteren Version hinzugefügt werden. Änderungen sind vorerst in den Dateien durchzuführen.

Effizienz

Das Werkzeug soll effizient und wirtschaftlich gestaltet werden. Die Anforderungen an die Reaktionszeit des Werkzeuges müssen eingehalten werden.

Änderbarkeit

Das Werkzeug muss modular aufgebaut sein, um eventuelle Änderungen schnell und wirtschaftlich durchzuführen. Dies ist ebenso nötig um gegebenenfalls Erweiterungen rasch einbinden zu können. Der Programmcode muss den geltenden Normen der OÖGKK entsprechen.

8.9 Benutzeroberfläche

Zum Eintragen und Warten der Referenz- und Grenzwerte wird ein graphisches User-Interface entwickelt. Ebenso wird ein Interface zum Verwalten der Abfragen und Parameter zur Verfügung gestellt.

8.10 Technische Werkzeugumgebung

8.10.1 Software

Das Werkzeug muss mit der SAS® BI-Plattform kompatibel und auf dem Betriebssystem Unix/AIX lauffähig sein (siehe Kapitel 5.2). Es muss sichergestellt sein, dass die Anwendungen netzwerkfähig sind. Das Werkzeug wird ebenfalls über Unix/AIX installiert und gewartet.

8.10.2 Hardware

Das Werkzeug wird auf den Servern der OÖGKK installiert. Es müssen die Mindestvoraussetzungen für diese Server eingehalten werden.

8.10.3 Werkzeugschnittstellen

Die Daten werden in Dateien mittels SAS-Dateiformat gespeichert. Des Weiteren ist die gesamte SAS® BI-Plattform im Einsatz (siehe Kapitel 5.2). Es entstehen hierbei Schnittstellen zu beiden Systemen, zum einen zur SAS-Datei auf Grund des Datenaustausches und zum anderen zur SAS® BI-Produktpalette mit der Darstellung und Einbindung in das System der OÖGKK (DWH-Portal).

8.11 Anforderungen an die Entwicklungsumgebung

Die Anforderungen an die Entwicklungsumgebung entsprechen denen der technischen Werkzeugumgebung (siehe Kapitel 8.10).

9 Resümee

Abschließend werden die zentralen Punkte der vorliegenden Arbeit nochmals aufgezeigt und das Ergebnis zusammengefasst.

Um einen Einblick in den Themenschwerpunkt zu geben, wurde zunächst ein Überblick über die zwei zentralen Ausgangspunkte - Data Warehousing und den damit in Verbindung stehenden ETL-Prozess - gegeben. In weiterer Folge wurde der Begriff Datenqualität definiert und verschiedene Ausprägungen aufgezeigt. Es wurde dargestellt, wie ein möglicher Ablauf des Data Cleaning Prozesses aufgebaut werden kann. Außerdem wurden verschiedene Fehlerquellen erläutert, welche die Ursache für fehlende Datenqualität darstellen. Zum einen kann die Ursache in einer einzigen Datenbank auf Schema- oder Instanzebene liegen, zum anderen entstehen Datenqualitätsprobleme bei der Zusammenführung von verschiedenen Datenquellen.

Den zentralen Teil dieser Arbeit stellt die Auffindung eines geeigneten Werkzeuges für die automatisierte Plausibilitätskontrolle in der Oberösterreichischen Gebietskrankenkasse dar. Zu diesem Zweck wurde eine Ist-Analyse der OÖGKK durchgeführt. Fünf ausgewählte Werkzeuge wurden analysiert, sodass eine Auflistung ihrer Funktionen und Methoden zur Datenbereinigung aufgestellt werden konnte. Es wurde ein Vergleich der Funktionen der fünf Werkzeuge vorgenommen, um ein geeignetes Werkzeug für die Zielvorstellungen der OÖGKK zu finden.

Es konnte kein Werkzeug identifiziert werden, das den Anforderungen der OÖGKK entspricht. Deshalb wurde in weiterer Folge ein Werkzeug konzipiert, das speziell auf die Bedürfnisse der OÖGKK abgestimmt ist. Es werden die benötigten Funktionen dargestellt, die zur automatisierten Plausibilitätskontrolle der Datenhaltung der OÖGKK notwendig sind. Kernkomponenten dieses Frameworks sind die Analyse der Daten der OÖGKK, sowie das Überprüfen dieser auf Grund von festgesetzten bzw. ad hoc berechneten Grenz- und Referenzwerten. Das Framework stellt einen modularen Aufbau zur Verfügung, sodass in weiterer Folge Erweiterungen einfach hinzugefügt werden können.

Ein weiterer Teil dieser Arbeit war die Spezifikation eines Softwarewerkzeuges zur automatisierten Plausibilitätskontrolle (SofaP). Die Implementierung, Installierung und Wartung von SofaP wird von der OÖGKK übernommen.

A Taxative Aufzählung der Satzarten

Satzarten: 010: Arzt-Behandlungsscheine
011: Behandlungsscheine - Einzelpositionen
015: Diagnosen je Behandlungsschein
020: Arzt-Umsatzdaten
030: Verordnungen von Heilmitteln
040: Verordnungen von Heilbehelfen / Hilfsmittel
041: Einzelverordnungen Heilbehelfe / Hilfsmittel
050: Veranlaßte Transporte
060: Krankschreibungen
070: Veranlaßte Krankenhausaufenthalte (ohne Leistungshonorar)
071: Veranlaßte Krankenhausaufenthalte (mit Einzelleistungshonorar)
072: Veranlaßte Krankenhausaufenthalte mit leistungsorientierter Finanzierung → Abteilungen, Diagnosen
073: Veranlaßte Krankenhausaufenthalte mit leistungsorientierter Finanzierung → Einzelleistungen, Diagnosefallgruppe
080: Zuweisungen zu Spitalsambulanzen
081: Spitalsambulanzen - Einzelpositionen
090: Veranlaßte Kur- und Erholungsaufenthalte
100: Zuweisungen zu kasseneigenen Ambulatorien + Vorsorgeuntersuchung
101: Ambulatorien - Einzelpositionen (+ VU)

200: Vertragspartnerstamm
210: Arzt-Leistungspositionen (Stamm)
220: Arzt-Abrechnungskategorien
230: Heilmittelstamm
240: Heilbehelfe- / Hilfsmittelstamm
250: Ambulatorien - Positionsstamm
255: Behandlungsgruppen-Kataster
260: Transporte - Positionsstamm
270: Krankenhaus - Positionsstamm bzw. Diagnosefallgruppen
280: Indikationsgruppen für Heilmittel
290: Versichertenstamm
300: Dienstgeberstamm
310: Beitragsgruppen
320: Wirtschaftsklassen
330: Diagnosestamm

B Aufbau der DATA-Library

Satzartentabellen (Schnittstellendaten):

Arztabrechnung:	DATA.SA10	Kopfdaten je Fall
	DATA.SA11	Leistungsdaten je Fall
	DATA.SA20	Limitierungen
Heilmittel:	DATA.SA30	Rezeptdaten je Fall
Heilbehelfe/Hilfsmittel:	DATA.SA40	Kopfdaten je Fall
	DATA.SA41	Leistungsdaten je Fall
Transporte:	DATA.SA50	Transportdaten je Fall
Arbeitsunfähigkeit:	DATA.SA60	Arbeitsunfähigkeitsdaten je Fall
Krankenhausaufenthalte:	DATA.SA70	Daten des Krankenhausaufenthalte je Fall
Kur und Erholung:	DATA.SA90	Daten der Kur oder Erholung je Fall
Hauseigene Ambulatorien:	DATA.SA100	Kopfdaten je Fall
	DATA.SA101	Leistungsdaten je Fall

Tabelle 40: Aufbau Satzartentabellen

Literatur

[AM97] Sam Anahory and Dennis Murray. *Data Warehouse / Planung, Implementierung und Administration.* Addison-Wesley-Longman, Bonn [u.a.], 1997.

[Bal98] Helmut Balzert. *Lehrbuch der Softwaretechnik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung.* Spektrum, Akademischer Verlag, Heidelberg, Berlin, 1998.

[BG04] Andreas Bauer and Holger Günzel. *Data Warehouse Systeme - Architektur, Entwicklung, Anwendung, 2., überarb. u. aktualisierte Aufl.* dpunkt-Verlag, Heidelberg, 2004.

[BSM03] Matthew Bovee, Rajendra P. Srivastava, and Brenda Mak. A conceptual framework and belief-function approach to assessing overall information quality. *Int. J. Intell. Syst.*, 18(1):51–74, 2003.

[CCS93] E. F. Codd, S. B. Codd, and C. T. Salley. Providing olap (on-line analytical processing) to user-analysts: An it mandate. *White Paper, Arbor Software Cooperation*, 1993.

[CG98] Peter Chamoni and Peter Gluchowski. *Analytische Informationssysteme.* Springer, Berlin [u.a.], 1998.

[Dat94] C. J. Date. *A Guide to the SQL Standard.* Addison-Wesley, Reading, Mass., 1994.

[EEV02] Mohamed G. Elfeky, Ahmed K. Elmagarmid, and Vassilios S. Verykios. Tailor a record linkage tool box. In *ICDE*, pages 17–28, 2002.

[EIV07] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[Fri03] Jeffrey E. F. Friedl. *Reguläre Ausdrücke.* O'Reilly, Köln, 2003.

[Ger05] Jens Gerken. Data cleaning in data mining and warehousing I. *Universität Konstanz*, 2005.

[HK00] Jiawei Han and Micheline Kamber. *Data mining: concepts and techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[HM03] J.C. Freytag H. Müller. Problems, methods, and challenges in comprehensive data cleansing. *In Technical Report HUB-IB-164*, 2003.

[HS95] Mauricio A. Hernández and Salvatore J. Stolfo. The merge/purge problem for large databases. In *SIGMOD Conference*, pages 127–138, 1995.

[HS00] Andreas Heuer and Gunter Saake. *Datenbanken - Konzepte und Sprachen.* MITP Verlag, Bonn, 2000.

[IH94] W. H. Inmon and R. D. Hackathorn. *Using the Data Warehouse.* John Wiley and Sons, Inc., New York, NY, USA, 1994.

[Inm02] William H. Inmon. *Building the Data Warehouse.* John Wiley and Sons, Inc., New York, NY, USA, third edition, 2002.

[JLVV99] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis. *Fundamentals of Data Warehouses.* Springer-Verlag, 1999.

[KCH+03] Won Y. Kim, Byoung-Ju Choi, Eui Kyeong Hong, Soo-Kyung Kim, and Doheon Lee. A taxonomy of dirty data. *Data Min. Knowl. Discov.*, 7(1):81–99, 2003.

[KGH05] Joachim Kübart, Udo Grimmer, and Jochen Hipp. Regelbasierte Ausreissersuche zur Datenqualitätsanalyse. *Datenbank-Spektrum*, 5(14):22–28, 2005.

[KR02] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit.* John Wiley and Sons, Inc., New York, NY, USA, second edition, 2002.

[KRRT98] R. Kimball, L. Reeves, M. Ross, and W. Thornwaite. *The Data Warehouse Lifecycle Toolkit.* John Wiley and Sons, Inc., New York, NY, USA, 1998.

[KS91] Won Kim and Jungyun Seo. Classifying schematic and data heterogeneity in multidatabase systems. *Computer*, 24(12):12–18, 1991.

[Kur99] Andreas Kurz. *Data Warehousing Enabling Technology.* MITP-Verlag, Bonn, 1999.

[Leh03] Wolfgang Lehner. *Datenbanktechnologie für Data-Warehouse-Systeme: Konzepte und Methoden.* dpunkt-Verlag, Heidelberg, 2003.

[LGJ03] Dominik Lübbers, Udo Grimmer, and Matthias Jarke. Systematic development of data mining-based data quality tools. In *VLDB*, pages 548–559, 2003.

[Mel93] J. Melton. *Understanding the new SQL.* Morgan Kaufmann Publishers, San Mateo, Calif., 1993.

[Mic08] Microsoft® SQL Server™ 2005. Microsoft® Sql Server™ 2005 Documentation. http://msdn2.microsoft.com/en-us/library/ms130214.aspx, 2008. download am 15.04.2008.

[MM00] Jonathan I. Maletic and Andrian Marcus. Data cleansing: Beyond integrity analysis. In *IQ*, pages 200–209, 2000.

[NP08] The OLAP Report Nigel Pendse. Olap market share analysis. http://www.olapreport.com/market.htm, 2008. download am 15.07.2008.

[Ora06] Oracle® Warehouse Builder. Oracle® Warehouse Builder User's Guide 10g Release 2 (10.2.0.2) B28223-03 Pdf-Dokument. http://www.dke.jku.at/manuals/owb10g/doc/owb.102/b28223.pdf, 2006. download am 15.04.2008.

[Ora08] Oracle® Warehouse Builder. Oracle® Warehouse Builder User's Guide 10g Release 2 (10.2.0.2) B28223-03. http://www.dke.jku.at/manuals/owb10g/doc/owb.102/b28223/toc.htm, 2008. download am 15.04.2008.

[Orr98] Ken Orr. Data quality and system theory. *Commun. ACM*, 41(2):66–71, 1998.

[RD00] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[Red96] T. C. Redman. Data quality for dthe information age. *Artech House*, 1996.

[SAS08a] SAS. BASE SAS® Documentation. http://support.sas.com/documentation/onlinedoc/base/index.html, 2008.

[SAS08b] SAS. SAS® 9.1.2 Data Quality Server Documentation. http://support.sas.com/91doc/docMainpage.jsp?_topic=dqclref.hlp/a002282656.htm, 2008. download am 15.04.2008.

[SAS08c] SAS. SAS® Enterprise BI Server Documentation. http://support.sas.com/documentation/onlinedoc/portal/index.html, 2008. download am 15.04.2008.

[SAS08d] SAS. SAS® Enterprise Guide Documentation. http://support.sas.com/documentation/onlinedoc/guide/index.html, 2008. download am 15.04.2008.

[SAS08e] SAS. SAS® Information Delivery Portal Documentation. http://support.sas.com/documentation/onlinedoc/portal/index.html, 2008. download am 15.04.2008.

[SAS08f] SAS. SAS® Web Report Studio Documentation. http://support.sas.com/documentation/onlinedoc/wrs/index.html, 2008. download am 15.04.2008.

[SAS08g] SAS®. Sas Institute Inc. http://www.sas.com/, 2008. download am 22.04.2008.

[SLW97] Diane M. Strong, Yang W. Lee, and Richard Y. Wang. Data quality in context. *Commun. ACM*, 40(5):103–110, 1997.

[SMB05] Monica Scannapieco, Paolo Missier, and Carlo Batini. Data Quality at a Glance. *Datenbank-Spektrum*, 5(14):6–14, 2005.

[TB98] Giri Kumar Tayi and Donald P. Ballou. Examining data quality. *Commun. ACM*, 41(2):54–57, 1998.

[Wie99] John-Harry Wieken. *Der Weg zum Data Warehouse / Wettbewerbsvorteile durch strukturierte Unternehmensinformationen.* Addison-Wesley-Longman, München [u.a.], 1999.

[Win08a] WinPure. Vergleich ListCleaner Pro und Clean and Match 2007. http://www.winpure.com/compare.html, 2008. download am 22.04.08.

[Win08b] WinPure. WinPure ListCleaner Pro. http://www.winpure.com/, 2008. download am 22.04.2008.

[Wiz08a] WizSoft. WizRule®. http://www.wizsoft.com/, 2008. download am 22.04.2008.

[Wiz08b] WizSoft. WizRule® Trainingsvideo. http://www.wizsoft.com/ruletvideo.asp, 2008. download am 22.04.2008.

[WS96] Richard Y. Wang and Diane M. Strong. Beyond accuracy: what data quality means to data consumers. *J. Manage. Inf. Syst.*, 12(4):5–33, 1996.

[WW96] Yair Wand and Richard Y. Wang. Anchoring data quality dimensions in ontological foundations. *Commun. ACM*, 39(11):86–95, 1996.

[Zad65] L.A. Zadeh. Fuzzy sets. *Information and Control*, 3(8):338–353, 1965.