

Diplomarbeit I98

Materialisierte Views

Projektbeschreibung

Dokumentnummer: **I98MATVIEW-INIT02**
Version: **0.4**
Datum: **05.07.2002**
Autor(en): **Benjamin Regez**
Ersetzt Version: **0.3**
Verteiler: **P. Fierz, Dr. J. Fulde, Dr. F. Flueckiger**
Bemerkungen:

Änderungskontrolle

Version	Datum	Name	Bemerkungen
V0.1	07.05.2002	Benjamin Regez	Dokument erstellt
V0.2	13.05.2002	Benjamin Regez	Dokument erweitert
V0.3	02.06.2002	Benjamin Regez	Dokument überarbeitet
V0.4	05.07.2002	Benjamin Regez	Dokument überarbeitet (Konzeptwechsel)

Anmerkungen zum Dokument

Dokument erstellt mit: Microsoft Word 2000
Dateiname : Projektbeschreibung.doc

Inhaltsverzeichnis

1	Einleitung	4
1.1	Zweck des Dokumentes	4
1.2	Begriffe und Abkürzungen	4
2	Aufgabenstellung und grundlegende Ziele	5
2.1	Aufgabenstellung	5
2.2	Grundlegende Ziele	5
3	Angestrebtes Vorgehen	6
4	Technologien	7
5	Organisation	7
6	Terminplanung	8
7	Zusätzliche Informationen	9
7.1	Projektverlauf und Dokumente	9
Anhang	10
A.	Verzeichnisse	10
B.	Quellen	10
C.	Änderungen	10
D.	Zusätzliche Unterlagen	12

1 Einleitung

1.1 Zweck des Dokumentes

Dieses Dokument stellt das Projektpflichtenheft für die Diplomarbeit *Materialisierte Views* dar. Da es im Zusammenhang mit dem Thema der Diplomarbeit nicht möglich sein wird, den genauen Gegenstand sowie die Messbaren Ziele von Beginn weg exakt festzulegen, muss eine Begriffskorrektur vorgenommen werden.

Die Korrektur des Begriffs Pflichtenheft erfolgt im Einverständnis von Herrn Fierz, der dieses im Rahmen der Initialsitzung vom 6. Mai 2002 gegeben hat. Dieses Dokument nennen wir demzufolge **Projektbeschreibung**, anstelle des üblichen Begriffs **Projektpflichtenheft**.

So entspricht auch der Inhalt und somit der Zweck des vorliegenden Dokumentes nicht ganz dem eines üblichen Pflichtenheftes. Im wesentlichen werden die folgenden Punkte behandelt:

- ? Welches sind die grundlegenden Ziele, die durch Bearbeitung der Aufgabe erreicht werden sollen?
- ? Wie ist das Vorgehen, das zur Lösung der gestellten Aufgabe und somit zum Erreichen der genannten Ziele dienen soll?
- ? Basierend auf welchen Technologien und Techniken soll die Aufgabenstellung bearbeitet werden?
- ? Wie ist die Organisation innerhalb des Bearbeiterteams, d.h. wer übernimmt welche Rolle innerhalb des Bearbeitungsprozesses?
- ? Wie sieht die grobe Terminplanung aus und was für Meilensteine sind geplant?

Die Antworten auf diese Fragen liefert die vorliegende Projektbeschreibung. Da auch die in diesem Dokument gemachten Aussagen und Definition nicht finaler Natur sind, unterliegen diese einem gewissen Änderungspotential. Dieses ist vor allem im Bereich der eingesetzten Technologien und der Terminplanung anzusiedeln.

1.2 Begriffe und Abkürzungen

Die Begriffe und Abkürzungen sind im Glossar definiert, das der Einfachheit halber in einem eigenen Dokument zusammengefasst wurde. Das entsprechende Dokument ist mit der Dokument-Nummer I98MATVIEW-GLOS01 versehen.

2 Aufgabenstellung und grundlegende Ziele

2.1 Aufgabenstellung

Die Aufgabenstellung schlägt laut publiziertem Themenvorschlag [BOI02] folgendes Vorgehen vor:

1. Analysieren für welche art von Views eine automatische Generierung der Triggers in Frage kommt.
2. Für einige ausgewählte Views die Triggers von Hand schreiben um die vorherige Analyse zu verifizieren.
3. Schreiben eines Parsers, der die Triggers automatisch erstellt.
4. Integrieren der materialisierten Views im Datenbanksystem. Materialisierte Views gehören zur physikalischen Ebene der 3-Schichten Architektur. Das heisst, eine gegebene View kann materialisiert bzw. entmaterialisiert werden, ohne dass die Anwendungsprogramme verändert werden müssen.

Aus dem aufgeführten Vorschlag für die Vorgehensweise lassen sich die drei Teilaufgaben bilden:

1. Analyse und Klassifizierung verschiedener Viewtypen und definieren eines Mechanismus zur Aktualisierung der Viewdaten.
2. Erstellen eines Parsers zur Automatisierung der Trigger Erstellung.
3. Integration in ein DBMS.

Auf diesen drei Teilaufgaben, die mit der vorgeschlagenen Vorgehensweise gelöst werden sollen, basiert der vorliegende Projektbeschreibung.

2.2 Grundlegende Ziele

Aus dem Themenvorschlag [BOI02] kann ein grob umrissenes Hauptziel entnommen werden:

- ? Aus einer vorhandenen transienten View soll eine materialisierte View erstellt werden, die mittels automatisch generierter Triggers bei Veränderung der Datenbestände in den Basisentitäten automatisch aktualisiert wird.

Die Generierung der Triggers bzw. der Triggerdefinitionen in SQL soll mittels eines Parsers erfolgen. Dieses relativ grob formulierte Hauptziel lässt sich noch etwas verfeinern und in einzelne Teilziele unterteilen. Diese werden klassifiziert, um eine Abgrenzung zwischen erforderlichen und optionalen Zielen zu erhalten

Nr.	Zielformulierung	Zielklassifizierung
1	Analysieren & Klassifizieren der verschiedenen Viewtypen als Voraussetzung für die Evaluation eines entsprechenden View Maintenance Algorithmus (Ziel Nr. 5 und 6).	Mussziel
2	Transiente Views in materialisierte Form überführen.	Mussziel
3	Erstellen eines SQL Parsers zur automatisierten Überführung transienter Views in die materialisierte Form. View Maintenance erfolgt dabei auf der Basis vollständiger Neuberechnung der Viewdaten.	Mussziel
4	Integration in ein DBMS durch Implementation einer Schnittstelle, die die erforderlichen Aktionen auf das DBMS ausführt.	Mussziel
5	Evaluation und Implementation eines View Maintenance Algorithmus.	Sollziel
6	Evaluation und Implementation eines zusätzlichen View Maintenance Algorithmus, der in Kombination mit dem bestehenden Algorithmus die Effizienz der View Aktualisierung steigert.	Kannziel
7	Erstellen einer Benutzerschnittstelle bzw. -oberfläche, welche die Erstellung und Verwaltung der Views steuert.	Kannziel

8	Erstellen weiterer Schnittstellen für ein oder mehrere zusätzliche DBMS.	Kannziel
9	Funktionalität gekapselt als Service für andere Applikationen oder Dienste zur Verfügung stellen. Stichwort <i>Webservices</i> .	Kannziel
10	Kompatibilität zu älteren oder neueren SQL Spezifikationen als der aktuell gewählten herstellen.	Kannziel
11	Messen, untersuchen und auswerten der Geschwindigkeitsunterschiede, die sich aus der Materialisierung der Views ergeben.	Kannziel

Tabelle 1: Zielformulierung

Die Zielklassifizierung ist wie folgt definiert.

Mussziel	Ziele diesen Typs stellen Kernkriterien für die Erfüllung der grundlegenden Aufgabenstellung dar. Das Erreichen aller Mussziele stellt dabei die minimale Anforderung an die angestrebte Lösung dar.
Sollziel	Sollziele stellen Anforderungen dar, die die Brauchbarkeit der angestrebten Lösung stark beeinflussen. Das Erreichen aller Sollziele erfüllt die Aufgabenstellung im gestellten Rahmen.
Kannziel	Kannziele sind per Definition optional. Diese sind nicht direkt in der Aufgabenstellung aufgeführt. Daher richten sich diese auch in die Bereiche der Erweiterbarkeit, Benutzerfreundlichkeit sowie der erhöhten Brauchbarkeit der Lösung.

Tabelle 2: Definition Zielklassifizierung

Die Nummerierung der Ziele hat dabei keinen direkten Zusammenhang mit deren Priorisierung. Die Reihenfolge ergibt sich in erster Linie durch die Zielklassifizierung, innerhalb einer Klassifizierung können die Ziele dabei in beliebiger Reihenfolge angegangen werden. Einschränkungen müssen nur gemacht werden, falls Abhängigkeiten innerhalb der Ziele ein solches Vorgehen verunmöglichen.

Die Ziele sollen in einem iterativen Vorgehen schrittweise erarbeitet werden. Wie dies im einzelnen aussieht, wird im folgenden Kapitel detailliert erläutert.

3 Angestrebtes Vorgehen

Das in der Aufgabenstellung (Kapitel 2.1) aufgeführte Vorgehen wird von den Projektbearbeitern als sinnvoll erachtet und in den Grundzügen übernommen.

Es soll jedoch, gestützt auf die Gliederung in Teilaufgaben, eine Unterteilung in Phasen vorgenommen werden. Eine solche Unterteilung soll sicherstellen, dass am Ende jeder Phase prüf- und brauchbare Resultate vorhanden sind, die als Grundlage für die nächste Phase dienen oder in dieser entsprechend weiter ausgebaut werden.

Dadurch wird sichergestellt, dass die Gesamtlösung in mehreren Zyklen schrittweise erarbeitet wird. Der Gefahr einer Übergewichtung des Funktionsumfangs gegenüber der Stabilität und Funktionalität der Gesamtlösung wird so vorgebeugt.

Phase	Aufgabenteile	Zielerfüllung
1	Analyse & Klassifizierung der Viewtypen für die Evaluation entsprechender View Maintenance Algorithmen.	1
2	Überführen transienter Views in materialisierte Form, automatisiert und in DBMS integriert.	2, 3, 4
3	Erweitern der Funktionalität durch Implementation eines View Maintenance Algorithmus.	5
4	Stabilisieren und Testen der aktuell erstellten Lösung.	-
5	Implementation einzelner Kannziele im Rahmen der verbleibenden Zeitressourcen.	[6, 7, 8, 9, 10, 11]
6	Stabilisierung der implementierten Kannziele und der Gesamtlösung, Schlusstests.	-
7	Schlussdokumentation, Abschlussarbeiten	-

Tabelle 3: Unterteilung in Phasen

Bei der fünften Phase sind die in der Spalte *Zielerfüllung* aufgeführten Ziele in Eckige klammern gesetzt. Diese Notation soll darauf Hinweisen, dass eine Auswahl aus den aufgeführten Zielen umgesetzt werden kann. Diese Auswahl erfolgt wahlfrei oder in Absprache mit den Betreuern.

Die entsprechenden Dokumentationsarbeiten sind in den einzelnen Phasen enthalten.

4 Technologien

Zum aktuellen Zeitpunkt kann die Festlegung auf einzelne Technologien als schwierig eingestuft werden. Grundlegende Definitionen sollen jedoch gemacht werden, um die Stossrichtung der angestrebten Lösung aufzuzeigen.

Da der Aspekt der Forschung in der Projektarbeit eine zentrale Rolle spielt, hat sich das Bearbeiterteam vorgenommen, auch im Bereich der technischen Entwicklung auf neue Technologien zu setzen. Für die Umsetzung des Parsers sowie der gesamten Automation und Benutzerschnittstelle wird deshalb auf die Programmiersprache C# gesetzt. C# ist eine neu entwickelte objektorientierte Programmiersprache, die Java relativ stark ähnelt und die folgenden wichtigsten Merkmale hat[MOS02]:

- ? C# besteht zu 70% aus Java, 10% C++, 5% Visual Basic und 15% sind neu.
- ? C# ist voll objektorientiert und implementiert dieselben Konzepte wie Java.
- ? C# verfügt über einige sehr interessante Neuerungen wie beispielsweise. Versionisierung, Referenzparameter und Komponentenunterstützung.

Da C# Bestandteil des Microsoft .NET Framework ist, liegt auch die Wahl eines passenden DBMS nahe. Der MS SQL Servers 2000 implementiert den kompletten SQL 92 Standard und eignet sich daher bestens für den Einsatz im Rahmen der vorliegenden Projekt- bzw. Diplomarbeit.

Die nachfolgende Tabelle fasst die Technologien zusammen, die das Bearbeiterteam in den einzelnen Bereichen einzusetzen gedenkt:

Bereich	Technologie(n) und/oder Komponente(n)
DBMS	MS SQL Server 2000
SQL	SQL 92
Parser/Compiler Generator	Coco/R oder ANTLR für C#
Parser/Compiler Sprache	C#
Benutzerschnittstelle bzw. -oberfläche	C# in ASP.NET
Webservices	XML und SOAP

Tabelle 4: Für Realisierung vorgesehene Technologien und Komponenten

Die bisher getätigten Recherchearbeiten für den Einsatz von C# ergeben ein durchwegs positives Bild. Viele Universitäten und Hochschulen beschäftigen sich intensiv mit dieser Programmiersprache.

Unsicherheiten bestehen jedoch noch in der Wahl des Compiler/Parser Generators. Sowohl Coco/R [MOSCOCO] als auch ANTLR [ANTLR02] können Parser in C# generieren. Unklar ist jedoch noch, inwiefern sich die SQL Grammatik mit den Parser Generatoren verträgt.

Falls Änderungen im Bereich der Wahl der eingesetzten Technologien stattfinden oder unumgänglich sein sollten, dann werden diese in Rücksprache mit den Betreuern vorgenommen.

5 Organisation

Bezüglich der Projektorganisation wurden an der Sitzung vom 6. Mai 2002 bereits einige Beschlüsse getroffen.

Da die Zahl der beteiligten Personen klein und daher überschaubar ist, wird an dieser Stelle auf eine detaillierte Übersicht verzichtet. Die beteiligten Personen sind in Tabelle 5 mit der entsprechenden Rolle aus Projektexterner Sicht aufgeführt.

Rolle	Angaben zur Person
Bearbeiter	Benjamin Regez (benjamin.regez@isolutions.ch) Stefan Voss (svoss@gmx.ch)
Betreuer	P. Fierz (fierz@hta-be.bfh.ch) Dr. J. Fulde (fulde@hta-be.bfh.ch)
Experte	Dr. F. Flueckiger (federico.flueckiger@dir.supsi.ch)

Tabelle 5: Projektorganisation - beteiligte Personen

Zusätzlich sollen auch die Rollen innerhalb des Bearbeiterteams definiert werden, damit die Ansprechpartner und Verantwortlichkeiten für die Betreuer und den Experten klar geregelt sind.

Die Rollenverteilung bedeutet nicht, dass der Bearbeiter, der nicht in der Zuteilung der Rolle aufgeführt ist, keine Arbeiten in diesem Bereich tätigen wird. Die Rollenzuteilung regelt lediglich, wer im entsprechenden Bereich die Koordination der Arbeiten, die zeitliche Planung und die Verantwortung übernimmt.

Gegen aussen sind beide Bearbeiter als Team verantwortlich für den Projektfortschritt und das erfolgreiche Gelingen der Projekt- bzw. Diplomarbeit.

Rolle	Zuteilung
Verantwortlich für Administration und Ansprechpartner für Betreuer und Experte.	Benjamin Regez
Verantwortlich für Analyse und Spezifikation	Stefan Voss
Verantwortlich für Design und Implementation	Benjamin Regez, Stefan Voss
Verantwortlich für Stabilisierung und Tests	Stefan Voss
Verantwortlich für Dokumentation	Benjamin Regez
Verantwortlich für Projektwebseite	Benjamin Regez

Tabelle 6: Rollenverteilung innerhalb des Bearbeiterteams

6 Terminplanung

Die Terminplanung gibt Aufschluss zu den Arbeitsschritten und Aufgaben, die im zeitlichen Ablauf der Projekt- bzw. Diplomarbeit getätigt werden. Entsprechende Meilensteine definieren den Abschluss wichtiger Phasen und dienen als Messpunkte für die Kontrolle des Projektfortschritts.

Die Arbeitszeiten, die für die Realisierung der Projekt- bzw. Diplomarbeit investiert werden, können bedingt durch die Auslastung am Arbeitsplatz und durch Ferien- oder Krankheitsbedingte Abwesenheiten variieren. Grundsätzlich ist jedoch ein zeitlicher Einsatz von 12 Stunden pro Woche und Bearbeiter vorgesehen. Dies entspricht dem durch die Schule vorgesehenen Zeitfenster von 16 Lektionen pro Woche.

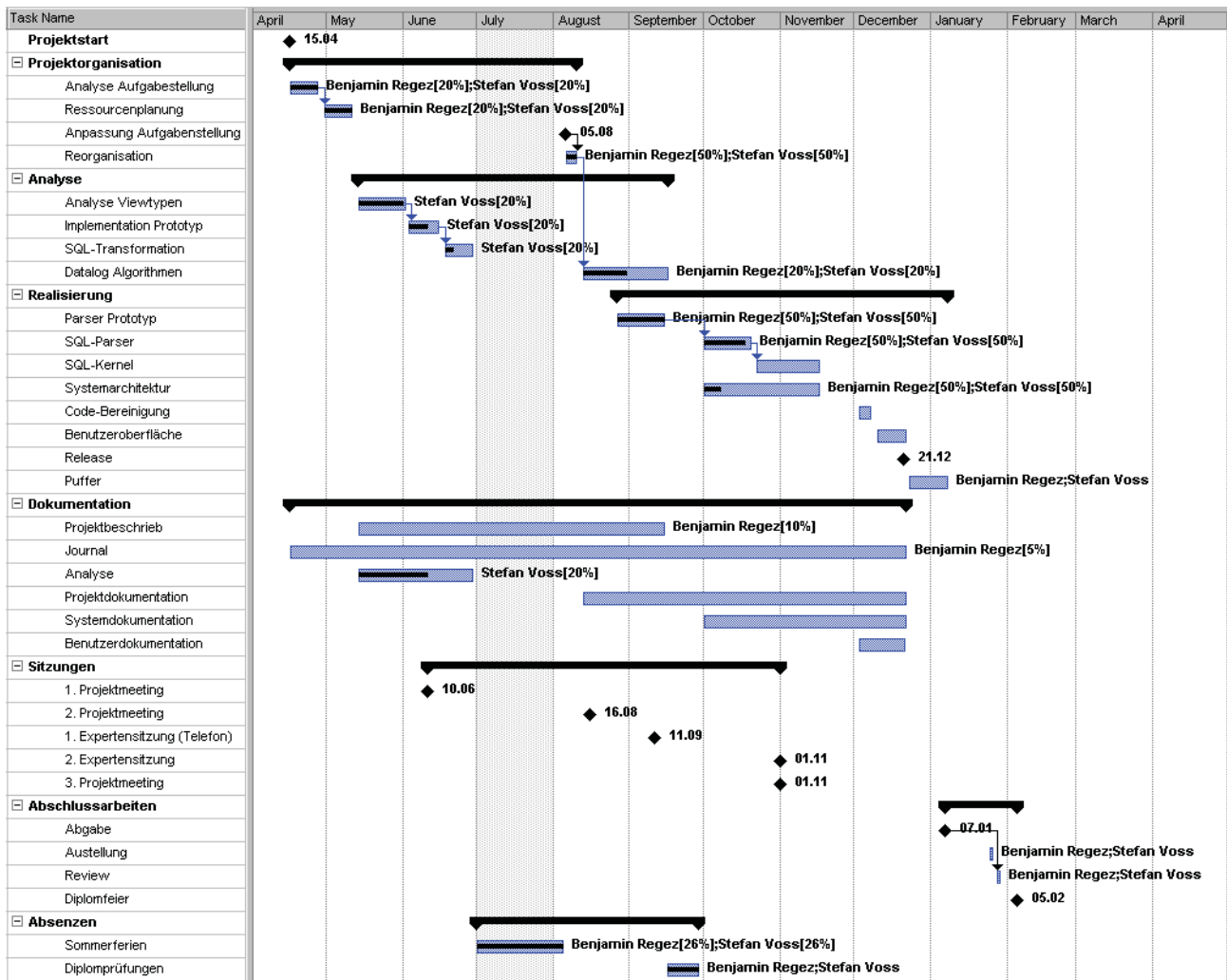


Abbildung 1: Terminplanung

Die Terminplanung ist, wie weiter oben bereits erwähnt, einem gewissen Änderungspotential unterworfen. Dieses resultiert hauptsächlich aus der Aufgabenstellung, die einige weitgehend unbekannte Bereiche, wie beispielsweise die Implementation verschiedener View Maintenance Algorithmen, aufweist. Das Abschätzen des Zeitbedarfs für die Bearbeitung einzelner Aufgaben innerhalb eines solchen Bereichs kann nicht mit einer akzeptablen Genauigkeit vorgenommen werden.

Aus diesem Grund verzichtet die Terminplanung auf das Abschätzen des Zeitbedarfs für die Erfüllung einzelner Ziele. Auch mit der in Phase fünf (s. Kapitel 3) vorgesehenen optionalen Auswahl eines zu implementierenden Zieltes liesse sich eine solche Zeitplanung schlecht in Einklang bringen.

Aus diesem Grund sind die einzelnen Ziele in Analyse- und Realisierungsabschnitt integriert und nicht einzeln aufgeführt.

7 Zusätzliche Informationen

7.1 Projektverlauf und Dokumente

Jeweils aktuelle Informationen zum Projektverlauf und die erarbeiteten Dokumente können auf der Projektwebseite unter www.matviews.ch abgerufen werden. Auch sämtlicher Quellcode und alle Arbeitsdokumente werden auf dieser Webseite verfügbar gemacht.

Der Zugriff auf die Projektwebseite erfordert ein gültiges Benutzerkonto. Ein solches kann bei Bedarf jederzeit angefordert werden. Für den öffentlichen Zugriff existiert ein Standardkonto *public*, das ohne Passwort den Zugriff auf die Projektwebseite gestattet.

ANHANG

A. Verzeichnisse

A1. Abbildungsverzeichnis

Abbildung 1: Terminplanung9

A2. Tabellenverzeichnis

Tabelle 1: Zielformulierung6
 Tabelle 2: Definition Zielklassifizierung6
 Tabelle 3: Unterteilung in Phasen.....7
 Tabelle 4: Für Realisierung vorgesehene Technologien und Komponenten7
 Tabelle 5: Projektorganisation - beteiligte Personen8
 Tabelle 6: Rollenverteilung innerhalb des Bearbeiterteams8
 Tabelle 8: Grundlegende Ziele vor dem Konzeptwechsel 11
 Tabelle 9: Geplantes Vorgehen vor dem Konzeptwechsel 11
 Tabelle 10: Gewichtung der Arbeitsschritte 12
 Tabelle 11: Definition Notenskala 12

B. Quellen

[ANTLR02] 'ANTLR, Complete Language Translation Solutions', Online: <http://www.antlr.org>, 2002

[BOI02] Boillat, J., 'Abteilung Informatik HTA-BE. Themenvorschlag für eine Diplomarbeit.', Online: <http://www.hta-be.bfh.ch/~wwwinfo/di/02/matview.shtml>, Hochschule für Technik und Architektur, Bern, 2002.

[MOS02] Mössenböck, H., 'C# - die neue Sprache für Microsoft .NET', Online: <http://www.ssw.uni-linz.ac.at/Teaching/Lectures/CSharp/01.Ueberblick.pdf>, System Software Group der Johannes Kepler Universität, Linz, 2002

[MOSCOCO] Mössenböck, H., 'Coco/R for C#', Online: <http://www.ssw.uni-linz.ac.at/Research/Projects/Coco/CSharp>, Institute of Practical Computer Science, Johannes Kepler University, Linz

C. Änderungen

Durch den Konzeptwechsel, der aus den gewonnenen Erkenntnissen vorgenommen wurde, mussten an den grundlegenden Zielen und dem daraus resultierenden angestrebten Vorgehen Änderungen vorgenommen werden. Die vor dem Konzeptwechsel in der Projektbeschreibung enthaltenen Ziele und das damit verbundene Vorgehen werden hier der Vollständigkeit halber aufgeführt.

C1. Grundlegende Ziele

Die untenstehende Tabelle zeigt die grundlegenden Ziele vor dem vorgenommenen Konzeptwechsel.

Nr.	Zielformulierung	Zielklassifizierung
1	Analysieren & Klassifizieren der verschiedenen Viewtypen.	Mussziel
2	Views einfachen Typs in materialisierte Form überführen.	Mussziel
3	Erstellen eines Parsers zur automatisierten Überführung einfacher Viewtypen in materialisierte Form.	Mussziel

4	Integration in ein bestimmtes DBMS durch Implementation einer Schnittstelle, die die erforderlichen Aktionen auf das DBMS ausführt und die materialisierten Views verwaltet.	Mussziel
5	Views komplexeren Typs in materialisierte Form überführen (wo überhaupt möglich bzw. sinnvoll).	Sollziel
6	Erweitern des Parsers zur Überführung komplexerer Viewtypen in materialisierte Form.	Sollziel
7	Erweitern der DBMS-Schnittstelle zur Verwaltung von Views des komplexeren Typs auf Stufe DBMS	Sollziel
8	View Optimierung. Die zu materialisierende View soll analysiert und - falls möglich - während dem Analyseprozess optimiert werden.	Kannziel
9	Erstellen einer Benutzerschnittstelle bzw. -oberfläche, welche die Erstellung und Verwaltung der Views steuert.	Kannziel
10	Erstellen weiterer Schnittstellen für zusätzliche DBMS.	Kannziel
11	Funktionalität gekapselt als Service für andere Applikationen oder Dienste zur Verfügung stellen. Stichwort <i>Webservices</i> .	Kannziel
12	Kompatibilität zu älteren oder neueren SQL Spezifikationen als der aktuell gewählten herstellen.	Kannziel
13	Messen, untersuchen und auswerten der Geschwindigkeitsunterschiede, die sich aus der Materialisierung der Views ergeben.	Kannziel

Tabelle 7: Grundlegende Ziele vor dem Konzeptwechsel

C2. Angestrebtes Vorgehen

Die untenstehende Tabelle zeigt das angestrebte Vorgehen mit den in Anhang C1 aufgeführten Zielen als Grundlage.

Phase	Aufgabenteile	Zielerfüllung
1	Analyse & Klassifizierung der Viewtypen.	1
2	Überführen von Views einfacheren Typs in materialisierte Form, automatisiert und in DBMS integriert.	2, 3, 4
3	Erweitern der Funktionalität von Phase für Views komplexeren Typs.	5, 6, 7
4	Stabilisierung und Testen der aktuell erstellten Lösung.	-
5	Erfüllung einzelner Kannziele im Rahmen der noch verfügbaren Zeitressourcen.	[8, 9, 10, 11, 12]
6	Stabilisierung der implementierten Kannziele und der Gesamtlösung, Schlusstests.	-
7	Schlussdokumentation, Abschlussarbeiten	-

Tabelle 8: Geplantes Vorgehen vor dem Konzeptwechsel

D. Zusätzliche Unterlagen

D1. Gewichtung der Arbeitsschritte

Die Betreuer der vorliegenden Diplomarbeit definieren die in nachfolgender Tabelle aufgeführte Gewichtung der einzelnen Arbeitsschritte.

Arbeitsschritt	Gewicht
Vorbereitungsphase	
Projektbeschreibung	4
Durchführung	
Zeitplanung und Führung des Laborjournals	3
Kreativität, Initiative, Selbständigkeit	3
Wahl des Vorgehens und der Entwurfswerkzeuge	3
Implementation, Robustheit, Programmierstil, Erweiterbarkeit	4
Systemtests (Verfahren, Durchführung, Bericht)	2
Kommunikation mit Experte und Betreuer	1
Qualität der Resultate (sowohl positive wie negative)	2
Projektbericht	
Inhalt korrekt, vollständig, verständlich Quellenangaben sowohl für Text wie auch für Code vollständig	4
Texte wie auch für Code vollständig	1
Klare, aussagekräftige Zusammenfassung	1

Tabelle 9: Gewichtung der Arbeitsschritte

Notenskala	
A	6 (hervorragend)
B	5.5 (sehr gut)
C	5 (gut)
D	4.5 (befriedigend)
E	4 (ausreichend)
F	< 4 (nicht bestanden)

Tabelle 10: Definition Notenskala