



## **Datenbank Live Migration**

*Implementierung eines Frameworks für die Datenbankmigration*

## **Bakkelaureatsarbeit Proposal**

**Autor:** Karl Hannes Stückler

**Mat. Nr.:** 0651676

**Betreuer:** Dipl.-Ing. Mag. Dr. Albert Weichselbraun

# I. Inhaltsverzeichnis

<b>I.</b>	<b>INHALTSVERZEICHNIS .....</b>	<b>2</b>
<b>1.</b>	<b>MOTIVATION .....</b>	<b>3</b>
<b>2.</b>	<b>PROBLEMSTELLUNG .....</b>	<b>3</b>
<b>3.</b>	<b>TECHNISCHE BEGRIFFE .....</b>	<b>3</b>
3.1	POSTGRESSQL .....	3
3.2	ORACLE DATABASE .....	4
3.3	JAVA .....	4
<b>4.</b>	<b>STRUKTUR DER ARBEIT .....</b>	<b>4</b>
<b>5.</b>	<b>PROJEKTPLAN .....</b>	<b>6</b>
5.1	MEILENSTEINPLAN .....	6
5.2	BALKENPLAN.....	7
5.3	KLASSENDIAGRAMM.....	8
<b>II.</b>	<b>LITERATUR .....</b>	<b>9</b>

## 1. Motivation

Datenbanken sind aus dem heutigen Zeitalter nicht mehr wegzudenken und werden in vielen Bereichen eingesetzt. In den letzten Jahren hat sich die Technik laufend weiterentwickelt, sodass Datenbanken immer größere Datenmenge bewältigen können. Deshalb ist es von großer Wichtigkeit, bestehende Datenbestände in neue Datenbanksysteme schnell und einfach zu migrieren, damit die Vorteile der neuen Entwicklungen auch in der Praxis verwendet werden.

## 2. Problemstellung

Bei großen Datenbeständen benötigt deren Migration viel Zeit und Ressourcen. Der Inhalt dieser Arbeit behandelt zunächst verschiedene theoretische Datenbankmigrationsansätze. Als nächsten Schritt werden verschiedene Anbieter derartiger Migrationslösungen miteinander verglichen und diskutiert. Dabei wird grundsätzlich zwischen Open Source und kommerziellen Lösungen differenziert. Im dritten Teil werden die theoretischen Kenntnisse anhand eines praktischen Beispiels erörtert. Das Ziel dieser Arbeit ist es, ein Framework für die Migration zwischen beliebigen Datenbanksystemen zu entwickeln sowie die Bereitstellung einer Referenzimplementierung für die Migration von PostgreSQL 8.2 auf die Version 8.3. Die Umsetzung erfolgt mit der Programmiersprache Java.

### Wichtige Anforderungen:

- Performance
- Keine bzw. minimale Stillstandszeiten während der Migration
- Bereitstellung von Schnittstellen für die Unterstützung von beliebigen Datenbanken

### Zukünftige Erweiterungen:

- Unterstützung für Stored Procedures und Functions
- Unterstützung für Views

## 3. Technische Begriffe

Im folgenden werden die Technologien dargestellt, die im Projekt zur Anwendung kommen.

### 3.1 PostgreSQL

PostgreSQL ist ein objektrelationales Datenbankmanagementsystem mit Unterstützung für SQL92 und SQL99. Die Entwicklung begann in den frühen achtziger Jahren an der University of California. Seit 1997 wird die Software von der Open Source Community weiterentwickelt und zählt heute zu den beliebtesten frei verfügbaren Datenbanksystemen.

## 3.2 Oracle Database

Oracle Database ist ein relationales Datenbankmanagementsystem von der Firma Oracle. Diese Datenbank eignet sich aufgrund seiner Architektur und Skalierbarkeit für eine große Anzahl von gleichzeitig aktiven Benutzern. Weitere wichtige Eigenschaften sind die plattformübergreifende Unterstützung von verteilten Datenbanken sowie Data Mining und Datawarehouse Funktionalität. Dieses System wird vorwiegend in großen Unternehmen mit sehr großen Datenmengen eingesetzt.

## 3.3 Java

Java ist eine objektorientierte Programmiersprache, die von der Firma Sun entwickelt wird. Java Programme werden in Bytecode, eine plattformübergreifende Maschinensprache übersetzt und von einem Interpreter ausgeführt.

## 4. Struktur der Arbeit

Die nachfolgenden Punkte stellen das voraussichtliche Inhaltsverzeichnis der Bachelorarbeit dar:

**Abstract** (Deutsch und Englisch)

### 1. Inhaltsverzeichnis

- a. Abkürzungsverzeichnis
- b. Abbildungsverzeichnis
- c. Tabellenverzeichnis

### 2. Datenbanken

- a. Anforderungen
- b. Kriterien für die Auswahl
- c. Einsatz in der Praxis
- d. Konkrete DB Vorstellen: PostgreSQL

### 3. Traditionelle Migrationsstrategien

- a. Anforderungen an eine Migration
- b. „Chicken Little“ Ansatz
  - i. Database First
  - ii. Database Last
- c. „Cold Turkey“/Big Bang – Ansatz
- d. Butterfly

- e. Vergleich der Ansätze
4. Anbieter von Migrationslösungen
- a. Slony (Open Source für PostgreSQL → Punkte sind noch nicht genau fixiert)
    - i. Einführung
    - ii. Replizieren
    - iii. Beispiel
  - b. Oracle Replication (→ Punkte sind noch nicht genau fixiert)
    - i. Einführung
    - ii. Replizieren
    - iii. Beispiel
  - c. RedGate für SQL Server (→ Punkte sind noch nicht genau fixiert)
    - i. Einführung
    - ii. Replizieren
    - iii. Beispiel
5. Projektplanung
- a. Motivation
  - b. Anforderungen
  - c. Projektmanagement
    - i. Projektzieleplan
    - ii. Projektstrukturplan
    - iii. Projektmeilensteinplan
    - iv. Projektbalkenplan
6. Praktische Umsetzung
- a. Anforderungen
  - b. Design
    - i. Use Case Diagramme

- ii. Klassendiagramme
- c. Tests
  - i. Aufbau der Testumgebung
  - ii. Performance Tests
- 7. Zusammenfassung und Ausblick
- 8. Literatur

## 5. Projektplan

Im folgenden wir ein Überblick über den Ablauf des Projektes gegeben. Die Darstellung erfolgt mit Instrumenten des Projektmanagements und der Softwareentwicklung.

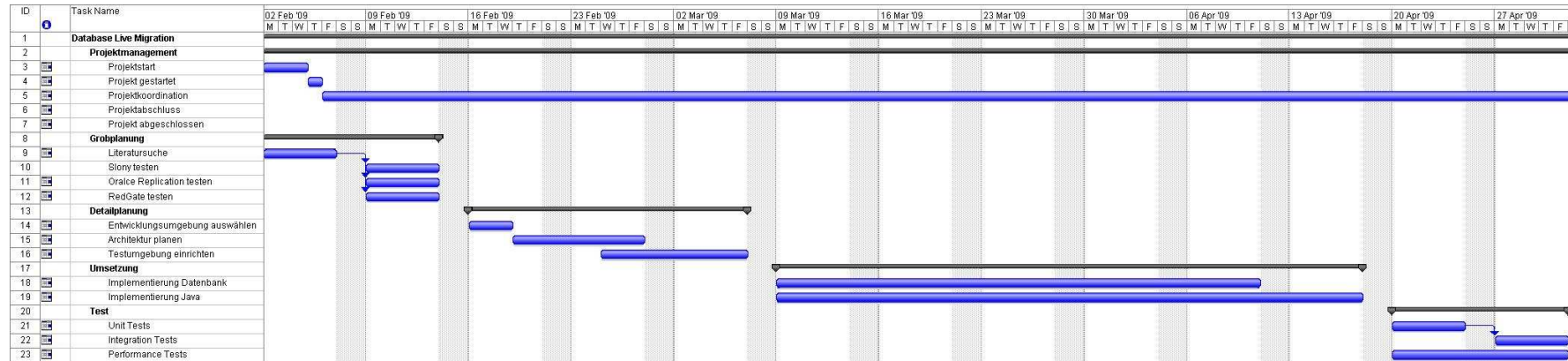
### 5.1 Meilensteinplan

Dieser Plan zeigt, zu welchen Terminen welche Phase abgeschlossen sein wird. Die Planungsphase dauert circa einen Monat. Die Umsetzung und die Tests bilden den Hauptteil des Projektes. Am 1. Mai wird das Projekt beendet.

Meilenstein	Basistermine	Aktuelle Plantermine	Ist Termine
Projekt gestartet	02.02.2009		
Grobplanung abgeschlossen	13.02.2009		
Detailplanung abgeschlossen	06.03.2009		
Umsetzung abgeschlossen	17.04.2009		
Test durchgeführt	01.05.2008		
Projekt abgeschlossen	01.05.2008		

## 5.2 Balkenplan

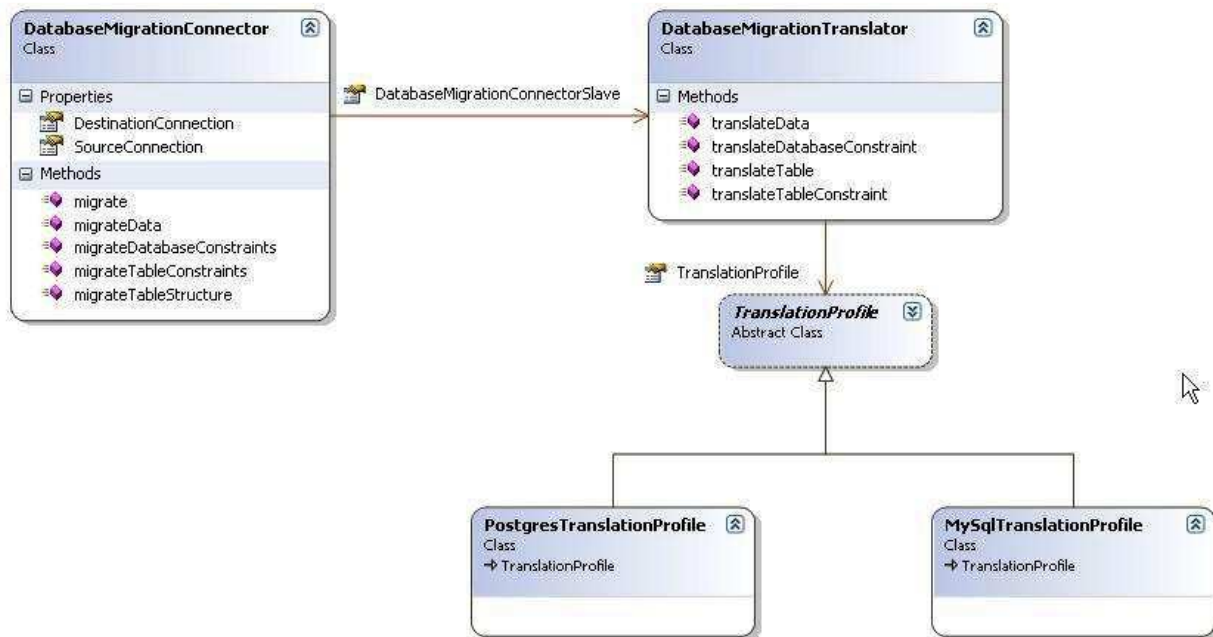
Die einzelnen Phasen des Meilensteinplanes werden hier detaillierter dargestellt.



### 5.3 Klassendiagramm

Die Klasse „*DatabaseMigrationConnector*“ kapselt die Verbindungen zur Quell- und Zieldatenbank. Dort werden die Daten von der Quelldatenbank gelesen und mithilfe der Klasse „*DatabaseMigrationTranslator*“ in den SQL Dialekt der Zieldatenbank übersetzt. Im Ergebnis wird pro SQL Befehl eine „translate“ Methode implementiert. Die abstrakte Klasse „*TranslationProfile*“ definiert eine Schnittstelle, mit der Klassen für beliebige SQL Dialekte implementiert werden können.

Um eine optimale Performance zu erreichen werden verschiedene Operationen, die unabhängig voneinander ausgeführt werden, in verschiedene Threads aufgeteilt.





## II. Literatur

1. **Broy, Prof. Dr. Dr. h.c. Manfred.** *Legacy Migrationsstrategien*. 2005. Seminararbeit.
2. **Oralce.** *Oracle Migration Workbench*. 2005.
3. **Slony.** Slony-I. [Online] January 25, 2009. <http://www.slony.info/>.
4. *Designing wrapper components for e-services*. **M. Mecella, B. Pernici.** s.l. : The VLDB Journal, 2001, The VLDB Journal.