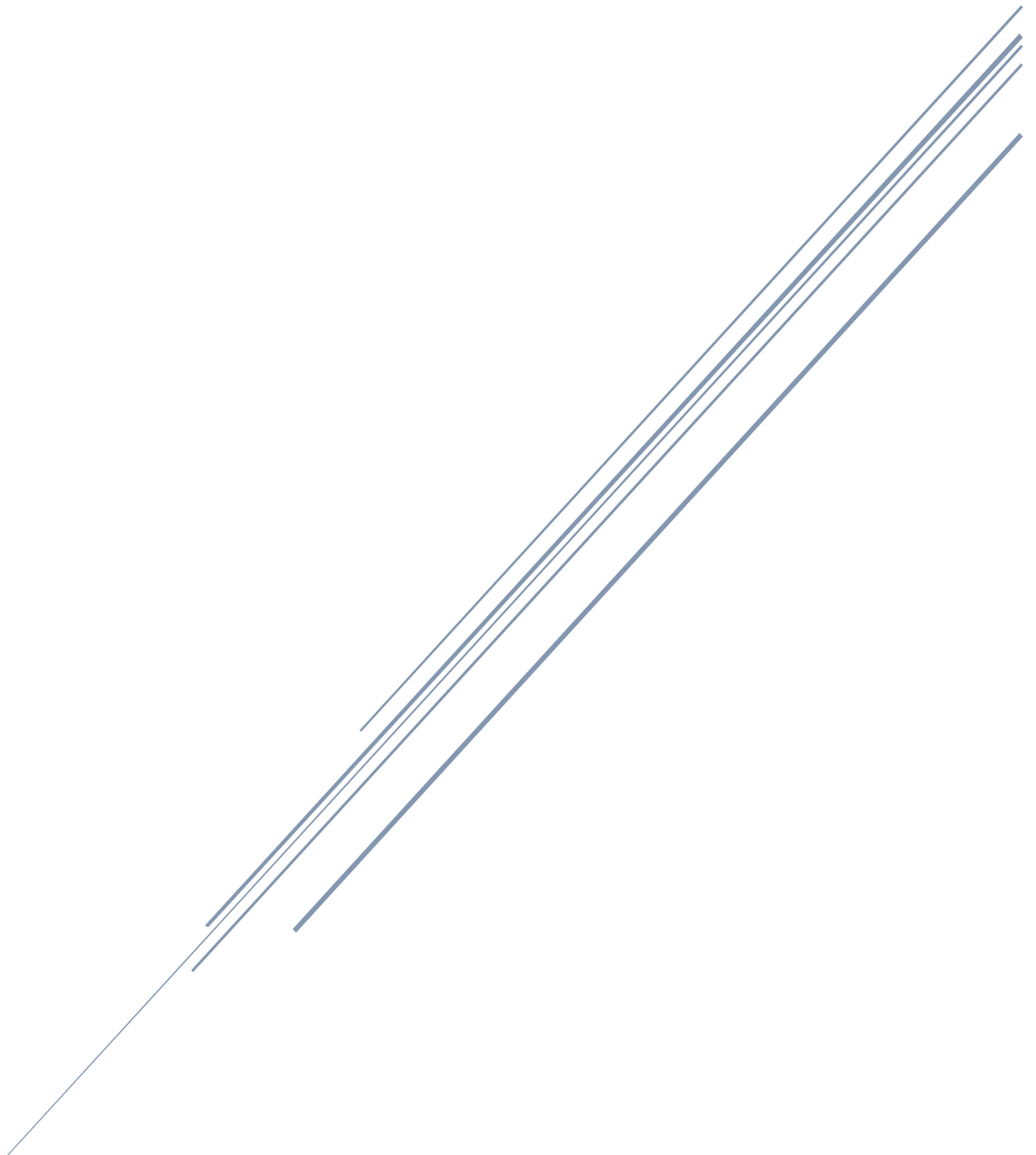


SYNCHRONISATION VON HETEROGENEN DATENBANKEN

DEZSYS-03



Brunner Helmuth, Pitirut Stefan
2014 - 11 - 13

Inhaltsverzeichnis

Aufgabenstellung.....	2
Designüberlegung.....	3
Arbeitsaufteilung.....	5
Aufwandsschätzung.....	5
Arbeitsdurchführung	5
Erfolge:	5
Niederlagen:	5
Verwendete Technologien	6
JDBC:.....	6
Strukturelle Heterogenität (Heterogene Datenbanken):	6
Mapping:	6
Start the Program	6
Quellen	7

Aufgabenstellung

Dokumentieren Sie Ihren Versuch zwei heterogene Datenbanksysteme (MySQL, Postgresql) zu synchronisieren. Verwenden Sie dabei unterschiedliche Schemata (verschiedene Tabellenstruktur) und zeigen Sie auf, welche Schwierigkeiten bei den unterschiedlichen Heterogenitätsgraden auftreten können (wie im Unterricht besprochen) [2Pkt].

Implementieren Sie eigenständig eine geeignete Middleware [8Pkt]. Testen Sie Ihr gewähltes System mit mehr als einer Tabelle [4Pkt] (Synchronisation bei Einfügen, Ändern und Löschen von Einträgen) und dokumentieren Sie die Funktionsweise, sowie auch die Problematiken bzw. nicht abgedeckte Fälle [2Pkt].

Das PDF soll ausführlich beschreiben, welche Annahmen getroffen wurden. Der Source-Code muss den allgemeinen Richtlinien entsprechen und ebenfalls abgegeben werden.

Gruppengröße: 2

Gesamtpunkte: 16 [Aufteilung in eckigen Klammern ersichtlich]

Punkte:

Dokumentation der Synchronisation [2Pkt]

Implementierung der Middleware [8Pkt]

- Zeittrigger bzw. Listener für Synchronisation bzw. DBMS Logs
- Konfiguration bez. Mapping der Tabellen und Attribute
- Konfliktlösung bei Zeitüberschneidung bzw. Datenproblemen (Log)
- LostUpdate-Problem

Test mit mehr als einer Tabelle und mindestens 10 Datensätze pro Tabelle [4Pkt]

- Uni- und Bidirektionale Änderungen mehrerer Tabellen
- Einfügen, Ändern und Löschen

Dokumentation der Funktionsweise, Problematiken und Problemfälle [2Pkt]

- Designdokumentation (Code + DB)
- Synchronisationsverhalten
- unbehandelte Problemfälle

Protokoll und Sourcecodedokumentation [0..-6Pkt]

Designüberlegung

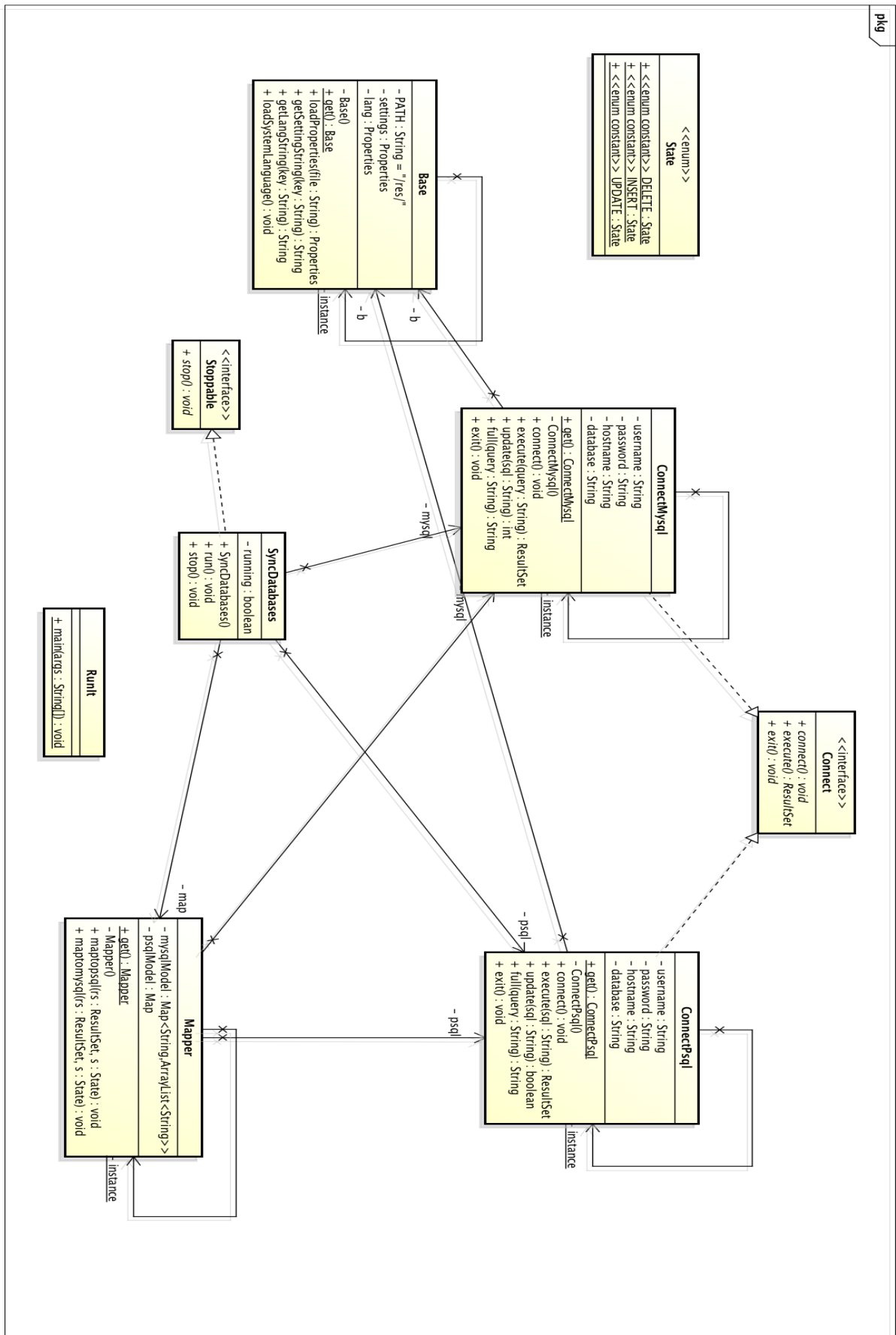
Wir haben uns am Anfang gedacht, dass das „update databases“ unser größtes problem sein wird, somit haben wir begonnen an einer Lösung zu arbeiten bzw. den Weg zur Lösung zu überlegen. Unsere erste vorrangsgehensweise war, das Update auf nicht heterogenen Datenbanken zu ermöglichen, dies wollten wir durch die Verwendung von `time_stamp()`; ermöglichen. Nach einigen Versuchen sind wir zur Überzeugung gekommen, dass wir diesen Task auf diese Variante nicht Lösen können. Als nächste vorrangsgehensweise haben wir uns für Versionierung entschieden. Unsere Idee war, beim updaten die zu Veränderten Datensätze in der Spalte version, um 1 zu erhöhen, und in einem Trigger dies zu verfolgen und bei einer Erhöhung, den selben Datensatz auf der andere Seite auch zu verändern. Damit haben wir weit mehr erreicht bei den nicht heterogenen Datenbanken als mit der vorher erwähnten vorangehensweise. Somit haben wir uns für diesen Weg entschieden für das Update zwischen zwei heterogene Datenbanken zu ermöglichen.

Wir haben am Anfang vermutet, dass „delete entry“ und „insert entry“ ungefähr gleich funktionieren. Wir dachten, dass wir versuchen könnten, diese zwei Sachen auf die selbe Art zu Lösen. Eine extra Tabelle (pro Funktion) erstellen, und diese auf `psql/mysql` Seite durch Synchronisation das selbe anzeigen zu lassen. Wir hatten noch eine andere Vorangehensweise, aber die hat sich als fehl am Platz herausgestellt, somit sind wir bei unserer ersten Annahme geblieben.

Die heterogene Datenbank haben wir wie folgt aufgebaut, wir haben uns hierbei für die strukturelle Heterogenität entschieden:

Leider haben wir dies nicht zum Abgabetermin geschafft, unsere Datenbank hat keine strukturelle Heterogenität, es ist eine einfache Datenbank.

Unser Konzept für diese Aufgabe sieht wie Folgt aus:



Arbeitsaufteilung

	Brunner	Pitirut
Dokumentation	X	X
Designüberlegung	X	X
Connection	X	
Middleware	X	X
Synchronisation	X	
Datenbanken		X

Aufwandsschätzung

Brunner	Geschätzte Zeit	Tatsächliche Zeit
Dokumentation	2 h	30 m
Designüberlegung	3 h	3 h 30 m
Connection	1 h	30 m
Middleware	3 h	3 h 30 m
Synchronisation	3 h	4 h
Gesamt:	12 h	12 h

Pitirut	Geschätzte Zeit	Tatsächliche Zeit
Dokumentation	2 h	2h 30 m
Designüberlegung	1 h	3 h
Middleware	2 h 30 m	1 h 30 m
Datenbanken	1 h 30 m	3 h
Gesamt:	7 h	9 h

Arbeitsdurchführung

Erfolge:

- Die „connection“ zu erstellen war schneller ermöglicht als erwartet.

Niederlagen:

- Das erstellen der Trigger für die jeweiligen Funktionen der Datenbanken waren aufwendiger als gedacht
- Das erstellen der Synchronisation verlief langsamer als angenommen, da die Trigger komplexer waren als angenommen
- Die Designüberlegung hat länger gedauert als geplant, da wir vieles Versucht haben, aber nicht alles Sinn ergab, somit auch einiges sofort wieder verworfen wurde.

Verwendete Technologien

JDBC:

Zum erstellen der jeweiligen “connections” wurde JDBC verwendet. Um das Auslesen aus der jeweiligen Datenbank, bzw. das Beschreiben der jeweiligen Datenbank zu ermöglichen wurde auch JDBC verwendet. Diese Technologie ist uns seit dem Vorjahr bekannt, in diesem Fall möchte ich auf die JDBC – Aufgabe aus dem Kontainer INSY, Fach DBSY, verweisen.

Strukturelle Heterogenität (Heterogene Datenbanken):

„Bei der Strukturellen Heterogenität handelt es sich um ein und den selben Sachverhalt, der von verschiedenen Personen erfasst und in einer Datenbank definiert wurden.

Beide Datenbanken unterscheiden sich nur vom Aufbau, die gespeicherten Daten sind jedoch gleichwertig.“ - Adeyemi Paul [1]

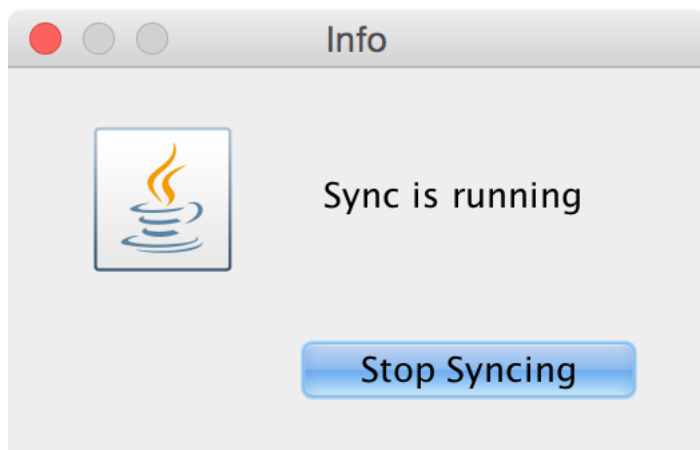
Mapping:

„Unter Mapping versteht man den Prozess der es 2 oder mehreren heterogenen Systeme erlaubt miteinander zu Kommunizieren bzw. Daten zu verschicken und verstehen.“ - Adeyemi Paul [1]

Start the Program

Es ist ein Ant-File zur Verfügung gestellt worden von uns, dieses kann man mit „ant run“. Damit man das Programm überhaupt starten kann, muss man vor dem ausführen des ant Befehls im Properties-file im res folder, den dbusername, dbpassword, dbhostname und dbname, für mysql und für psql ändern.

Damit die Verbindung zur Datenbank hergestellt werden kann.



Um das Programm zu beenden, muss der „Stop Syncing“ – Button betätigt werden.
Don't touch the EXIT (RED X) BUTTON!!!

Quellen

[1] : Adeyemi Paul, Synchronisation von Heterogenen Datenbanken , zuletzt bearbeitet: 17. Oktober 2014, zuletzt geöffnet: 13. November 2014, <https://elearning.tgm.ac.at/course/view.php?id=875>