

# Synchronisation von heterogenen Datenbanken

## SYT - 5A HIT

Ahmed Aly, Martin Haidn

November 13, 2014

### Contents

<b>1</b>	<b>Aufgabenstellung</b>	<b>2</b>
<b>2</b>	<b>Aufwandsschätzung</b>	<b>3</b>
<b>3</b>	<b>Datenbankentwurf</b>	<b>4</b>
3.1	ERD DB1 . . . . .	4
3.1.1	Create-Script . . . . .	4
3.2	ERD DB2 . . . . .	5
3.2.1	Create-Script . . . . .	5
<b>4</b>	<b>Middleware</b>	<b>6</b>
<b>5</b>	<b>Arbeitsdurchführung</b>	<b>6</b>
5.1	Peer Installation . . . . .	6
5.2	Propel Installation . . . . .	6

# 1 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 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

Dokumenation der Funktionsweise, Problematiken und Problemfälle [2Pkt]

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

Protokoll und Sourcecodedokumentation [0..-6Pkt]

## 2 Aufwandsschätzung

Für die Durchführung dieser Übung sind elf Stunden vorgesehen. Die Aufgabenstellung wurde in Arbeitspakete untergliedert und für die jeweiligen Tasks wurde eine Zeitaufwandsschätzung erstellt.

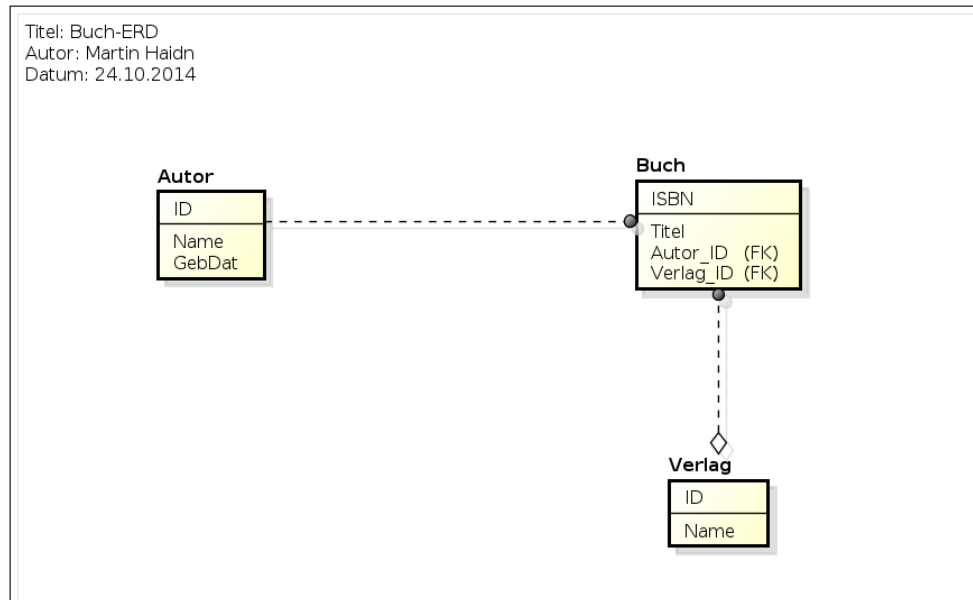
Arbeitszeitaufzeichnung - DBSync			
Task:	Mitglied:	Geschätzte Zeit:	Benötigte Zeit:
Vorbereitung & Evaluierung		1.5	
Datenbankentwurf (Bleistift ERD u. RM)		0.75	
Create-Script		0.5	
Middleware Konzept		1	
Realisierung der Middleware		5	
Testing		1	
Dokumentation & Organisation		1.5	
<b>Gesamt:</b>		<b>11.25</b>	
Task...	Durchzuführendes Arbeitspaket		
Mitglied...	Bearbeitendes Teammitglied		
Geschätzte Zeit...	Geschätzte Zeit für den jeweiligen task in (Stunden)		
Benötigte Zeit...	Tatsächlich investierte Zeit in den jeweiligen Task (Stunden)		

### 3 Datenbankentwurf

#### 3.1 ERD DB1

Bei dem Entwurf der Datenbank wurde auf ein simples Beispiel, bestehend aus drei Tabellen, zurückgegriffen. Es handelt sich um ein Buch, dass von einem bestimmtem Autor verfasst wurde, von dem eine eindeutige ID, der Name und das Geburtsdatum bekannt ist.

Jedes Buch wird von einem Verlag verlegt, der ebenfalls über eine eindeutige ID und einem Namen besitzt.



##### 3.1.1 Create-Script

Das Create-Script wurde über das Export-Tool von Astah an Hand des ER-Diagramms erstellt.

```

CREATE TABLE Autor (
  Autoren_ID INT NOT NULL,
  Name VARCHAR(25),
  Geburtsdatum DATE
);

ALTER TABLE Autor ADD CONSTRAINT PK_Autor PRIMARY KEY (Autoren_ID);

CREATE TABLE Verlag (
  ID INT NOT NULL,
  Verlagsname VARCHAR(30)
);

ALTER TABLE Verlag ADD CONSTRAINT PK_Verlag PRIMARY KEY (ID);

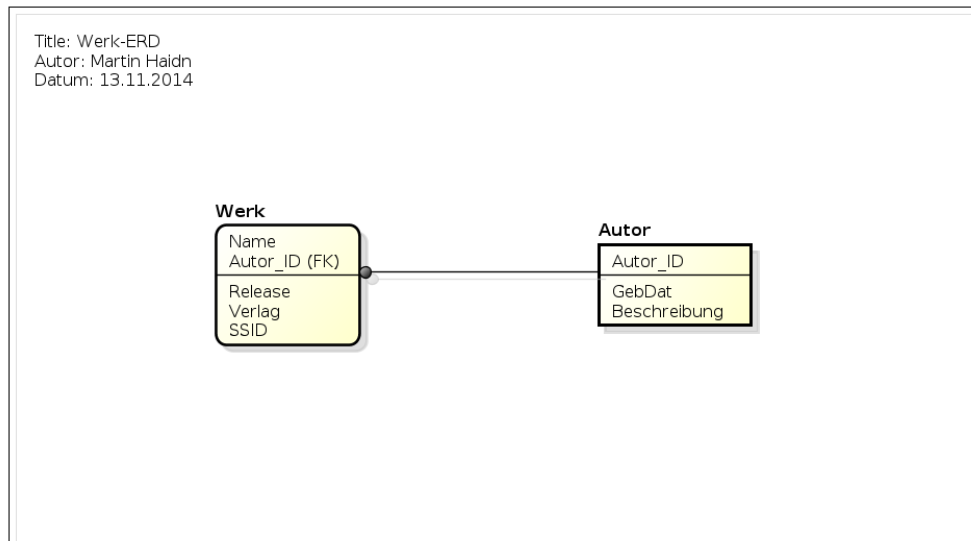
CREATE TABLE Buch (
  Internationale_Standard_Buchnummer VARCHAR(20) NOT NULL,
  Buchtitel VARCHAR(30),
  Autoren_ID INT NOT NULL,
  Verlags_ID INT
);

ALTER TABLE Buch ADD CONSTRAINT PK_Buch PRIMARY KEY (Internationale_Standard_Buchnummer);

ALTER TABLE Buch ADD CONSTRAINT FK_Buch_0 FOREIGN KEY (Autoren_ID) REFERENCES Autor (Autoren_ID);
ALTER TABLE Buch ADD CONSTRAINT FK_Buch_1 FOREIGN KEY (Verlags_ID) REFERENCES Verlag (ID);
  
```

## 3.2 ERD DB2

Ähnlich wie bei der ersten Datenbank, enthält die Zweite, Tabellen über das Halten von Informationen von Büchern, allerdings besitzt der Verlag keine eigene Identität mehr.



### 3.2.1 Create-Script

Das Create-Script wurde über das Export-Tool von Astah and Hand des ER-Diagrammes erstellt.

```

CREATE TABLE Autor (
  id INT NOT NULL,
  GebDat DATE,
  Beschreibung CHAR(10)
);

ALTER TABLE Autor ADD CONSTRAINT PK_Autor PRIMARY KEY (id);

CREATE TABLE Werk (
  name VARCHAR(50) NOT NULL,
  id INT NOT NULL,
  release DATE,
  verlag VARCHAR(50),
  SSID CHAR(10)
);

ALTER TABLE Werk ADD CONSTRAINT PK_Werk PRIMARY KEY (name, id);

ALTER TABLE Werk ADD CONSTRAINT FK_Werk_0 FOREIGN KEY (id) REFERENCES Autor (id);
  
```

## 4 Middleware

## 5 Arbeitsdurchführung

Für das Mappen auf die jeweiligen Datenbanken soll auf propel zurückgegriffen werden, ein Tool das letztes Jahr im Zuge einer Übung zum objektrelationalen Mapping verwendet wurde.

Der Vorteil der sich daraus ergibt liegt darin, dass die PHP-Klassen für das Mappen in die Datenbanken automatisch aus einem XML-Schema generiert werden und so nur noch die Synchronisation dazwischen implementiert werden muss.

Da die Installation von Propel über den Peer-PHP-Installer um einiges zügiger geht, wird dieser vorher installiert.

### 5.1 Peer Installation

Unter Linux kann Peer einfach mit dem folgenden Befehl installiert werden:

```
apt-get install php5-xsl php-pear
```

Da für manche Anwendungsfälle Phing benötigt wird macht es Sinn, auch Dies noch zu installieren:

```
sudo pear channel-discover pear.phing.info  
sudo pear install phing/phing
```

### 5.2 Propel Installation

Sind die vorherigen Punkte geglückt, kann nun Propel installiert werden:

```
pear channel-discover pear.propelorm.org  
pear install -a propel/propel-generator
```

Da für das Generieren der PHP-Klassen der Generator ausreicht, ignorieren wir die Propel-Runtime.