

# Datenbanken 1

## Proseminar Abgabe 2

SoSe 2015

### 1 Aufgabestellung

Das Ziel dieses zweiten Teils des Projektes ist es

1. das ER-Modell des ersten Teiles zu korrigieren und in ein relationales Schema umzusetzen;
2. dieses Schema in Form von Tabellen in SQL zu implementieren;
3. die Tabellen in der so entstandenen Datenbank mit Daten zu füllen;
4. die Datenbank wieder vollständig (Schemas und Daten) zu entfernen.

Die einzelnen Punkte werden nachfolgend im Detail beschrieben.

**Hinweis:** Die Namen der Entitäten und Attribute des ER-Modells sollten unbedingt mit den Namen der Relationen bzw. Tabellen und deren Attribute (wo möglich) übereinstimmen.

#### 1.1 ER-Diagramm in relationales Schema umsetzen

Korrigieren und verfeinern Sie, wo nötig, Ihr ER-Modell aus dem ersten Teil des Projektes und zeichnen Sie das ER-Diagramm. Beachten Sie dabei die Rückmeldung, die Sie zu Ihrer ER-Modellierung erhalten haben.

Setzen Sie das ER-Modell in das relationale Modell um. Geben Sie außer den Relationen und deren Attribute auch Schlüssel- und Fremdschlüssel-Bedingungen an.

**Abgabe:** PDF-Datei (`er-rel.pdf`) mit dem verbesserten ER-Diagramm und dessen relationale Umsetzung (Team-Name, sowie Namen und Matrikelnummern aller Mitglieder nicht vergessen).

#### 1.2 ER-Modell mit SQL umsetzen

Schreiben Sie ein SQL Skript zur Erstellung der Tabellen (ein SQL Skript ist eine Textdatei mit SQL Befehlen, die durch Strichpunkte getrennt sind).

Ihre Tabellendefinitionen sollte folgende Punkte abdecken:

1. passende Typen für die Attribute definieren und Default-Werte, wo sinnvoll;
2. Primärschlüssel festlegen;
3. Fremdschlüssel-Beziehungen definieren (überlegen Sie, welche Fremdschlüssel NULL-Werte erlauben sollen);
4. etwaige andere Integritätsbedingungen auf Attribute (z.B. NULL-Werte verbieten, Wertebereiche angeben) und/oder Tabellen festlegen.

Wo Ihre Wahl (z.B. einer Integritätsbedingung) nicht offensichtlich ist, bitte einen Kommentar hinzufügen. Für einzeilige Kommentare wird der doppelte Bindestrich (`-- Kommentar`) verwendet, für mehrzeilige Kommentare C-artige Bruchstrich-Stern Paare (`/* Kommentar */`).

**Abgabe:** SQL Skript `create.sql`, das alle Tabellen zum ER-Diagramm wie oben beschrieben erstellt.

### 1.3 Daten laden

Laden Sie mithilfe von `INSERT` Befehlen Daten in Ihre Datenbank. Achten Sie dabei darauf, dass durch Fremdschlüsselbedingungen keine Fehler auftreten.

**Abgabe:** SQL Skript `load.sql`, das die vorher erzeugten Tabellen mit Daten füllt.

### 1.4 Tabellen löschen

Löschen Sie alle Tabellen aus Ihrer Datenbank. Achten Sie darauf, dass durch Fremdschlüsselbedingungen keine Fehler auftreten.

**Abgabe:** SQL Skript `clean.sql`, das alle Tabellen löscht, die von `create.sql` erzeugt und von `load.sql` mit Daten gefüllt wurden.

## 2 Zugang zum Datenbankserver

Für das Proseminar wird ein PostgreSQL Datenbankserver zur Verfügung gestellt.

**Die Zugangsdaten werden Ihnen via Email zugesandt.**

Eine Starthilfe zur Verwendung des PostgreSQL Clients `psql` finden Sie hier:

<http://www.postgresql.org/docs/9.1/static/tutorial-accessdb.html>

Natürlich können Sie die Übung auch auf Ihrem eigenen PostgreSQL Server durchführen. PostgreSQL steht kostenlos zur Verfügung und ist einfach zu installieren.

Sie sollten Ihre Skripts jedoch unbedingt auf dem von uns zu Verfügung gestellten Server testen.

### 3 Abgabeformat

Die Abgabe erfolgt via E-Mail an `dbabgabe@cosy.sbg.ac.at` mit dem Betreff `PS Datenbanken <teamname>`.

Bitte halten Sie sich an folgendes Format für die Abgabe:

1. PDF des ER-Diagramms mit relationaler Umsetzung und die drei SQL Skripts werden in eine ZIP Datei mit dem Namen `<teamname>_2.zip` gepackt, wobei `<teamname>` durch den Namen Ihres Teams zu ersetzen ist, z.B. `a3_2.zip`, `b4_2.zip`.
2. Die PDF Datei und alle SQL Dateien enthalten einen Kopfteil mit folgenden Informationen (als Kommentar in SQL Skripts):
  - Name des Teams
  - Name und Matrikelnummer aller Mitglieder

Die Einteilung in Teams finden Sie auf der Website des Proseminars.

**Achtung:** Verspätete Abgaben oder Abgaben, die das oben angegebene Format nicht beachten, werden zurückgewiesen und nicht bewertet.

### 4 Bewertungskriterien

Die SQL Skripts werden wie folgt auf einer leeren Datenbank getestet. Stellen Sie sicher, dass alle Ihre SQL Skripts mit Ihrem `<user>` diesen Test ohne Fehler bestehen.

```
psql -h <host> -d <datenbank> -U <user> -f create.sql
```

```
psql -h <host> -d <datenbank> -U <user> -f load.sql
```

```
psql -h <host> -d <datenbank> -U <user> -f clean.sql
```

Zusätzlich zur Ausführbarkeit der SQL-Skripts fließen folgende Kriterien in die Bewertung ein:

- *Relationale Umsetzung:* Korrektheit der Umsetzung des ER-Modells in Relationen, einschließlich Schlüssel- und Fremdschlüssel-Bedingungen.
- *SQL Umsetzung:* Die Tabellen mit den angegebenen Integritätsbedingungen sollen das ER-Modell möglichst getreu wiedergeben. Die Attributtypen sollten sinnvoll und nachvollziehbar sein (im Zweifelsfall bitte kommentieren).
- *Zusätzliche Bedingungen* (über Schlüssel-/Fremdschlüsselbedingungen hinaus) auf Attribute und Tabellen, sofern sinnvoll, werden positiv bewertet.

Details zur Syntax der SQL Befehle in PostgreSQL (z.B. Integritätsbedingungen auf Spalten und Tabellen) finden Sie in der offiziellen Dokumentation:

<http://www.postgresql.org/docs/9.1/interactive/sql.html>