



Übung zur Vorlesung *Grundlagen: Datenbanken im WS20/21*

Christoph Anneser, Josef Schmeißer, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)
<https://db.in.tum.de/teaching/ws2021/grundlagen/>

Blatt Nr. Z2

In den folgenden Zusatzblättern wird der gesamte Prozess von der Modellierung eines Datenbankschemas, der Extraktion und Bereinigung der Daten, sowie der Erstellung eines geeigneten Datenbankschemas mittels SQL-DDL Statements und dem anschließenden Import der Daten in die Datenbank anhand des Beispiels der Fußball-Weltmeisterschaften der Jahre 1930 bis einschließlich 2014 erklärt.

Alle in diesem Blatt behandelten datenbankspezifischen Aspekte sind selbstverständlicherweise auch *klausurrelevant*.

Hausaufgabe 1

Überführen Sie das von Ihnen erstellte ER-Modell in ein entsprechendes Datenbankschema, indem Sie die jeweiligen DDL Anweisungen für die Entitäten und Relationen angeben. Achten Sie auch auf die Deklaration von Primär- und Fremdschlüsseln. Falls Sie zu keiner zufriedenstellenden eigenen Lösung im vorangegangenen Übungsblatt gekommen sein sollten, dann gehen Sie von der am Freitag veröffentlichten Beispiellösung aus.

```
create table worldcups
(
    year int primary key,
    country varchar
);
```

```
-----
create table teams
(
    key varchar(3) primary key,
    name varchar
);
```

```
-----
create table stadiums
(
    id int primary key,
    name varchar,
    city varchar
);
```

```
-----
create table referees
(
    id int primary key,
    name varchar,
```

```
    nation varchar(3)
);
```

```
-----
create table stages
(
    id int primary key,
    worldcup int references worldcups,
    stage varchar,
    knockout bool
);
```

```
-----
create table matches
(
    id int primary key,
    time timestamp,
    stage_id int references stages,
    stadium_id int references stadiums,
    home_team varchar(3) references teams,
    away_team varchar(3) references teams,
    referee int references referees,
    assistant1 int references referees,
    assistant2 int references referees,
    attendance int,
    half_time_home int,
    half_time_away int,
    goals_home int,
    goals_away int,
    win_on_extra_time bool,
    win_on_penalties bool,
    home_penalty_goals int null,
    away_penalty_goals int null
);
```

```
-----
create table coaches
(
    id int primary key,
    name varchar,
    nation varchar(3)
);
```

```
-----
create table coachings
(
    match_id int references matches,
    team varchar(3) references teams,
    coach_id int references coaches,
    primary key (match_id, team, coach_id)
);
```

```
create table players
(
  id int primary key,
  name varchar,
  nation varchar(3)
);
```

```
create table lineups
(
  player_id int references players,
  match_id int references matches,
  team varchar(3) references teams,
  starting bool, -- in starting 11?
  shirt_number int,
  captain bool,
  primary key (player_id, match_id, team)
);
```

```
create table goals
(
  id int primary key,
  player_id int references players,
  match_id int references matches,
  penalty bool,
  own_goal bool,
  minute int
);
```

```
create table substitutions
(
  id int primary key,
  player_id int references players,
  match_id int references matches,
  substitute_in bool, -- or substitute out?
  minute int
);
```

```
create table cards
(
  id int primary key,
  player_id int references players,
  match_id int references matches,
  type varchar(2), -- see description in kaggle dataset Y=Yellow, R=Red,
  SY=SecondYellow
  minute int
);
```

Hausaufgabe 2

In dieser Aufgabe setzen Sie einen lokalen Datenbankserver auf - mit diesem werden Sie Ihre oben erstellten DDL-Statements ausführen und testen.

Hinweis: DDL-Statements können nicht im Hyper-Webinterface ausgeführt werden. Dafür benötigen Sie weitergehende Rechte auf der Datenbank. Daher empfehlen wir die Installation eines Datenbankservers auf Ihrem lokalen Rechner!

Wir empfehlen hierfür Postgres, da es die gleiche Syntax wie HyPer benutzt, quelloffen (open-source) ist und vergleichsweise einfach installiert werden kann.

Die folgende Installationsanweisungen beziehen sich auf *Ubuntu 20.04*, wobei Postgres auch auf Windows und MacOS installiert werden kann.

```
$ sudo apt install postgresql
```

Den Datenbankserver kann man nun mit folgenden Befehlen steuern:

```
$ sudo systemctl (start|stop|restart|reload|status) postgresql
```

Wir legen nun einen neuen Benutzer an, mit dem wir uns künftig in `psql` einloggen werden. Als Nutzernamen benutzen Sie am besten Ihren Username, mit dem Sie auch in Ubuntu eingeloggt sind.

```
sudo -u postgres createuser -P <nutzernamen>
```

Im nächsten Schritt wechseln wir wieder zum User `postgres` und können nun unsere erste Datenbank anlegen, die wir `universitaet` nennen. Außerdem müssen wir den neuen Benutzer berechtigen, diese Datenbank verwenden zu dürfen:

```
$ sudo -u postgres psql
postgres=# CREATE DATABASE universitaet;
postgres=# GRANT ALL ON DATABASE universitaet to <nutzernamen>;
```

Die `psql`-Umgebung kann mit `\q` wieder verlassen werden.

Wir importieren nun das aus der Vorlesung bekannte Unischema. Dazu verlassen wir `psql` und laden zunächst die folgende SQL Datei von der Vorlesungswebsite herunter:

```
$ wget https://db.in.tum.de/teaching/ws1920/grundlagen/uni_mysql.sql
```

Nun erstellen wir ein neues Schema in der Datenbank und importieren alle Tabellen und Daten, die in der Datei `uni_mysql.sql` angegeben sind.

```
$ psql universitaet < uni_mysql.sql
```

Ab sofort können wir uns über `psql` mit der Datenbank verbinden:

```
$ psql universitaet
universitaet=> SELECT * FROM Studenten;
```

Nun können wir in der Datenbank neue Tabellen erstellen, diese mit Daten befüllen und Anfragen ausführen.

Aufgabe: Erstellen Sie nun eine Datenbank für die Fußball-Weltmeisterschaften und führen Sie die von Ihnen definierten DDL-Anweisungen aus, um alle Relationen zu erstellen.