

# Introduction to Data and Knowledge Engineering

## 6. Übung

### SQL



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

# Aufgabe 6.1

## Datenbank-Schema

Buch

	ISBN	Titel	Preis
PK	x		
FK			

Autor

	ID	Vorname	Nachname
PK	x		
FK			

BuchAutor

	ISBN	ID
PK	x	x
FK	Buch.ISBN	Autor.ID

# Aufgabe 6.1

## Datenbank-Schema

### Bibliothek

	Name	Ort	Telefon
PK	x		
FK			

### Exemplar

	Exemplarnr	ISBN	Bibliothek	Leser
PK	x	x		
FK		Buch.ISBN	Bibliothek.Name	Leser.ID

### Leser

	ID	Vorname	Nachname
PK	x		
FK			

# Aufgabe 6.1

a)



```
CREATE TABLE Buch (  
    ISBN CHAR(10),  
    Titel VARCHAR(50),  
    Preis DECIMAL(10, 2),  
    PRIMARY KEY (ISBN) );
```

```
CREATE TABLE Autor (  
    ID INT,  
    Vorname VARCHAR(30),  
    Nachname VARCHAR(30),  
    PRIMARY KEY (ID) );
```

```
CREATE TABLE BuchAutor (  
    ISBN CHAR(10),  
    ID INT,  
    PRIMARY KEY (ISBN, ID),  
    FOREIGN KEY (ISBN) REFERENCES Buch (ISBN),  
    FOREIGN KEY (ID) REFERENCES Autor (ID) );
```

# Aufgabe 6.1

a)



```
CREATE TABLE Bibliothek (  
    Name VARCHAR(30),  
    Ort   VARCHAR(30),  
    Telefon   VARCHAR(20),  
    PRIMARY KEY (Name) );
```

```
CREATE TABLE Exemplar (  
    Exemplarnr INT,  
    ISBN CHAR(10),  
    Bibliothek VARCHAR(30),  
    Leser INT,  
    PRIMARY KEY (Exemplarnr, ISBN),  
    FOREIGN KEY (ISBN) REFERENCES Buch(ISBN),  
    FOREIGN KEY (Bibliothek) REFERENCES Bibliothek(Name),  
    FOREIGN KEY (Leser) REFERENCES Leser(ID) );
```

```
CREATE TABLE Leser (  
    ID INT,  
    Vorname VARCHAR(30),  
    Nachname VARCHAR(30),  
    PRIMARY KEY (ID) );
```

# Aufgabe 6.1

b)

Löschen Sie das Attribut Ort aus der Tabelle Bibliothek und fügen Sie es anschließend wieder ein.

```
ALTER TABLE Bibliothek DROP COLUMN Ort;
```

```
ALTER TABLE Bibliothek ADD Ort varchar(30);
```

# Aufgabe 6.1

c)

Welches Problem tritt auf, wenn Sie versuchen die Tabelle Autor zu löschen? Wie können sie das Problem beheben?

Normalerweise würde man die Tabelle wie folgt löschen:

**DROP TABLE Autor;**

Dies funktioniert aber nicht, da ein Primärschlüsselattribut dieser Relation durch einen Fremdschlüssel in BuchAutor referenziert wird.

Folgende Anfrage dagegen hat Erfolg:

**DROP TABLE Autor CASCADE;**

Dabei ist aber zu beachten, dass durch obigen Befehl nicht das Attribut ID aus BuchAutor oder gar die ganze Tabelle BuchAutor gelöscht wird, sondern BuchAutor.ID einfach kein Fremdschlüssel mehr ist, sondern ein normales Attribut. Es wird also nur das Constraint FOREIGN KEY auf BuchAutor.ID gelöscht.

# SQL Ausprobieren

## postgresql

postgresql-8.4

<http://www.postgresql.de/>

<http://www.postgresql.org/>

user und datenbank anlegen:

<http://www.cyberciti.biz/faq/howto-add-postgresql-user-account/>

los gehts:

<http://user.it.uu.se/~rsomla/db2001/tips.html>

# Aufgabe 6.2

## a)-d)

- Geben Sie die Vor- und Nachnamen aller Leser aus.

```
SELECT Vorname, Nachname FROM Leser;
```

- Finden Sie den Titel des Buchs mit der ISBN "0201542633".

```
SELECT Titel FROM Buch WHERE ISBN='0201542633';
```

- Finden Sie den Titel des Buchs mit der ISBN "0201542633".

```
SELECT Nachname FROM Autor WHERE Nachname LIKE 'S%';
```

- Finden Sie die Vor- und Nachnamen der Autoren des Buchtitels „Objektdatenbanken“

```
SELECT Vorname, Nachname FROM Autor A, BuchAutor BA, Buch B
```

```
WHERE A.ID=BA.ID AND
```

```
BA.ISBN=B.ISBN AND B.Titel='Objektdatenbanken';
```

## Aufgabe 6.2

e)

- Geben Sie die Exemplarnr und die ISBN aller Exemplare aus und, falls das Exemplar ausgeliehen ist, den Nachnamen des Lesers, der es ausgeliehen hat.

```
SELECT E.Exemplarnr, E.ISBN, L.Nachname  
  
FROM Exemplar E LEFT OUTER JOIN Leser L ON E.Leser = L.ID;
```

Hinweis: Der LEFT OUTER JOIN bewirkt, dass alle Tupel der linken Seite des Joins (also alle Exemplare) in die Ergebnismenge gelangen, auch wenn sich kein passendes Tupel auf der rechten Seite findet (also in diesem Fall kein Leser, der das Exemplar ausgeliehen hat).

## Aufgabe 6.2

f)

- Finden Sie die IDs aller Leser, die aus der Informatik-Bibliothek Exemplare geliehen haben, jedoch nicht aus der Mathematik-Bibliothek.

```
SELECT DISTINCT Leser FROM Exemplar  
  
WHERE Bibliothek ='Informatik' AND NOT Leser IN  
  
(SELECT Leser FROM Exemplar  
  
WHERE Bibliothek='Mathematik' AND Leser IS NOT NULL);
```

Hinweis: Die Klausel "Leser IS NOT NULL" ist notwendig, damit in der verschachtelten Anfrage keine NULL-Werte auftauchen. Beispielsweise liefert die Frage, ob der Leser mit der ID "2" in der Tabelle NICHT enthalten ist, ein negatives Ergebnis. Das ist offensichtlich falsch, da der Leser "2" tatsächlich nicht darin vorkommt. Der in der Tabelle enthaltene NULL-Wert verfälscht das Ergebnis aber, so dass man ihn vorher eliminieren muss. Dieses Beispiel zeigt, dass NULL-Werte schwer auffindbare Fehler produzieren können und daher vermieden werden sollten.

# Aufgabe 6.2

## g)-j)

- Was kostet das billigste Buch?

```
SELECT MIN(Preis) FROM Buch;
```

- Wie lautet der Titel des teuersten Buchs und was kostet es?

```
SELECT Titel, Preis FROM Buch
```

```
WHERE Preis = ( SELECT MAX(Preis) FROM Buch );
```

- Welchen Wert hat der gesamte Exemplar-Bestand?

```
SELECT SUM(Preis) FROM Buch B, Exemplar E WHERE B.ISBN=E.ISBN;
```

- Was kostet ein Exemplar der Informatikbibliothek im Durchschnitt?

```
SELECT AVG(Preis) FROM Buch B, Exemplar E
```

```
WHERE B.ISBN=E.ISBN AND E.Bibliothek='Informatik';
```

## Aufgabe 6.2

### k)-m)

- Formulieren Sie die folgende Anfrage ohne die Verwendung von Aggregationsfunktionen:  
Wie lautet der Titel des teuersten Buchs und was kostet es?

```
SELECT Titel, Preis FROM Buch WHERE Preis >= ALL  
(SELECT Preis FROM Buch WHERE Preis IS NOT NULL);
```

- Wie viele Exemplare umfasst der Gesamtbestand der Bibliotheken?

```
SELECT COUNT(*) FROM Exemplar;
```

- Wie viele Autoren sind in der Informatikbibliothek vertreten?

```
SELECT COUNT(DISTINCT BA.ID) FROM BuchAutor BA, Exemplar E  
WHERE BA.ISBN = E.ISBN AND E.Bibliothek = 'Informatik';
```

## Aufgabe 6.2

### n)-o)

- Erstellen Sie eine Liste aller Leser, die Bücher aus der Informatikbibliothek entliehen haben, sortiert nach Nachnamen.

```
SELECT DISTINCT L.Nachname, L.Vorname FROM Leser L, Exemplar E  
WHERE E.Leser=L.ID AND E.Bibliothek ='Informatik' ORDER BY  
L.Nachname;
```

- Geben Sie für jeden Leser, der wenigstens ein Exemplar ausgeliehen hat, die ID und die Anzahl aller ausgeliehenen Bücher an.

```
SELECT L.ID, COUNT(E.ISBN) FROM Exemplar E, Leser L  
WHERE E.Leser = L.ID GROUP BY L.ID;
```

L.ID darf im SELECT-Statement auftauchen, da nach diesem Attribut gruppiert wird. L.Nachname darf hier beispielsweise nicht verwendet werden, womit es deutlich schwieriger wird, wenn man sich statt der ID den Nachnamen anzeigen lassen möchte.

## Aufgabe 6.2

p)

- Geben Sie für jeden Leser, der wenigstens ein Exemplar ausgeliehen hat, den Nachnamen und die Anzahl aller ausgeliehenen Bücher an. Verwenden Sie dazu eine verschachtelte Anfrage im FROM-Statement (Achtung: Diese Art der Verschachtelung erlauben einige Datenbanken nicht!).

```
SELECT L.Nachname, Anzahl FROM Leser L,  
  
(SELECT L.ID AS ID, COUNT(E.ISBN) AS Anzahl  
  
FROM Exemplar E, Leser L WHERE E.Leser = L.ID  
  
GROUP BY L.ID) Temp WHERE L.ID = Temp.ID;
```

## Aufgabe 6.2

q), r)

- Geben Sie für jeden Leser, der wenigstens zwei Exemplare ausgeliehen hat, die ID und die Anzahl aller ausgeliehenen Bücher an.

```
SELECT L.ID, COUNT(E.ISBN) FROM Exemplar E, Leser L
WHERE E.Leser = L.ID GROUP BY L.ID HAVING COUNT(E.ISBN) > 1;
```

- Die Buchpreise erhöhen sich um 2 Prozent. Ändern Sie die Relation Buch entsprechend.

```
UPDATE Buch SET Preis=Preis*1.02;
```

## Aufgabe 6.2

s), t)

- Herr Meier hat alle Bücher, die er ausgeliehen hat, zurückgegeben. Ändern sie die Datenbank entsprechend.

```
UPDATE Exemplar SET Leser=NULL WHERE Leser IN  
  
(SELECT ID FROM Leser WHERE Nachname='Meier');
```

- Frau Schmidt hat ihren Leseausweis zurückgegeben. Löschen Sie sie aus der Datenbank.

```
DELETE FROM Leser WHERE Nachname='Schmidt';
```