

Fügen sie folgende weitere Daten in ihre Datenbank ein!

```
INSERT INTO Person VALUES (1000001,'Meier','Max',2,3);
INSERT INTO Person VALUES (3012312,'Schmidt','Peter',2,1);
INSERT INTO Person VALUES (3923134,'Martin','Laura',2,5);
INSERT INTO Person VALUES (3423452,'Schneider','Fritz',2,4);
INSERT INTO Person VALUES (1344582,'Franz','Simone',5,8);
INSERT INTO Person VALUES (5498223,'Metzner','Sabine',2,6);
INSERT INTO Person VALUES (2394353,'Gottwald','Yvonne',3,4);
INSERT INTO Person VALUES (9832982,'Meißner','Karl',1,7);
```

```
INSERT INTO Fakultaet VALUES (1,'Wirtschaftswissenschaften');
INSERT INTO Fakultaet VALUES (2,'Informatik');
INSERT INTO Fakultaet VALUES (3,'Mathematik');
INSERT INTO Fakultaet VALUES (4,'Chemie');
INSERT INTO Fakultaet VALUES (5,'Physik');
```

```
INSERT INTO Veranstaltungen VALUES (1,'Rechnungswesen II-Kostenrechnung',1);
INSERT INTO Veranstaltungen VALUES (2,'BWL I-Marketing',1);
INSERT INTO Veranstaltungen VALUES (3,'BWL I-Produktionswirtschaft',1);
INSERT INTO Veranstaltungen VALUES (4,'Mathematik II',3);
INSERT INTO Veranstaltungen VALUES (5,'Methoden der Modellierung betrieblicher
Informationssysteme',1);
INSERT INTO Veranstaltungen VALUES (6,'Datenstrukturen',2);
INSERT INTO Veranstaltungen VALUES (7,'Statistik',3);
INSERT INTO Veranstaltungen VALUES (8,'Mikroökonomie',1);
INSERT INTO Veranstaltungen VALUES (9,'Rechnernetze/Betriebssysteme',2);
INSERT INTO Veranstaltungen VALUES (10,'Theoretische Informatik',2);
INSERT INTO Veranstaltungen VALUES (11,'Mathematik',2);
```

```
INSERT INTO Referent VALUES (1,234234,'I','Münch','Falko','',2000);
INSERT INTO Referent VALUES (2,234233,'I','Meier','Gerd','',1500);
INSERT INTO Referent VALUES (3,893049,'I','Günter','Franz','',2600);
INSERT INTO Referent VALUES (4,,'E','Kaufmann','Gerda','E-Soft',1700);
INSERT INTO Referent VALUES (5,1344582,'I','Franz','Simone','',500);
```

```
INSERT INTO Veranstaltungsbesuche VALUES (1344582,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (1000001,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (3012312,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (3923134,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (3423452,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (1344582,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (5498223,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (2394353,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (9832982,1,3,to_date('17.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (1000001,4,2,to_date('04.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (3012312,4,2,to_date('04.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (3923134,4,2,to_date('04.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (1344582,4,2,to_date('04.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (5498223,4,2,to_date('04.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (2394353,4,2,to_date('04.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (9832982,4,2,to_date('04.04.2002'));
INSERT INTO Veranstaltungsbesuche VALUES (1000001,2,1,to_date('10.05.2002'));
```

```
INSERT INTO Pflichtfaecher VALUES (2,1);
INSERT INTO Pflichtfaecher VALUES (2,2);
INSERT INTO Pflichtfaecher VALUES (2,3);
INSERT INTO Pflichtfaecher VALUES (2,4);
INSERT INTO Pflichtfaecher VALUES (2,5);
INSERT INTO Pflichtfaecher VALUES (2,6);
INSERT INTO Pflichtfaecher VALUES (2,7);
```

```
INSERT INTO Pflichtfaecher VALUES (2,8);  
INSERT INTO Pflichtfaecher VALUES (2,9);
```

Was macht folgendes Beispiel?

```
CREATE VIEW Wirtschaftsinformatiker AS  
  SELECT v.V_Nr, v.Bezeichnung,v.F_Nr  
  FROM veranstaltungen v, Pflichtfaecher p  
  WHERE v-v_nr = p.v_nr  
  AND S_Nr = 2;
```

```
SELECT * FROM Veranstaltungen;
```

```
SELECT * FROM Wirtschaftsinformatiker;
```

Erstellen sie folgende Sichten nach obigen Muster!

Erstellen sie eine Sicht welche nur die Veranstaltungen des Referenten Günter, Franz anzeigt!

```
CREATE VIEW Veranstaltungen_Guenter AS  
SELECT  
FROM Veranstaltungsbesuche vb, Referent r, Veranstaltungen v  
WHERE r.Name = 'Günter'  
AND r.R_Nr = vb.R_Nr  
AND vb.v_nr = v.v_nr;
```

Fragen sie die Daten ihres Views ab!

```
Select * from Veranstaltungen_Guenter;
```

Was fällt ihnen auf ?

Die Veranstaltungen erscheinen mehrmals

Begründen sie, warum die Veranstaltungsbesuche mehrmals in die Tabelle Veranstaltungsbesuche aufgenommen wurden!

Es könnte passieren, dass die Daten, welcher Student, welche Veranstaltungen besucht hat, sehr häufig abgefragt werden.

Mit dieser Lösung entfällt das Joinen

Die Sekretärin hat sich vertippt, es war nicht der Herr Günter, welcher die Veranstaltung „Rechnungswesen II-Kostenrechnung“ gehalten hat, sondern der Herr Münch!

Das Ändern bestehender Datensätze erfolgt mit der Anweisung „UPDATE“;

```
UPDATE Veranstaltungsbesuche  
SET R_nr =  
  (SELECT R_nr  
  FROM Referent  
  WHERE name = 'Münch')  
WHERE R_nr =  
  (SELECT R_nr  
  FROM Referent  
  WHERE name = 'Günter')  
;
```

Was müssen sie beim Ändern beachten?

Das auch wirklich alle Daten geändert werden, sonst kommt es zu einem inkonsistent Datenbestand.

Erhöhen sie das Gehalt von allen internen Referenten um 10%!

```
UPDATE Referent  
SET Gehalt = Gehalt * 1,1  
WHERE S = 'I';
```

Geben sie die Personalnummer, den Namen und das Jahresgehalt aus. Die Überschrift des Jahresgehalt soll nicht als Formel erscheinen, sondern als Name „Jahresgehalt“.

```
SELECT P_Nr, Name, Gehalt * 12 Jahresgehalt  
FROM Referent;
```

Für Berechnungen stehen diverse Funktionen zur Verfügung (Min, MAX, COUNT). Das Funktionen sind Gruppenfunktionen, welche auf alle Datensätze einer Tabelle angewendet werden.

COUNT	= zählt alle Datensätze
MIN	= ermittelt den Maximumwert
MAX	= ermittelt den Minimumwert
SUM	= Summe alle Datensätze
AVG	= Durchschnitt aller Datensätze

Geben sie die Anzahl (COUNT) der Veranstaltungsbesuche, das erste (MIN) und das letzte (MAX) Veranstaltungsdatum aus!

```
SELECT COUNT(*) Anzahl, Min(Datum) Erste, Max(Datum) Letzte  
FROM Veranstaltungsbesuche;
```

Mit DISTINCT Werte, welche mehrmals vorkommen nur einmal aufgeführt.

Ermittelnd sie die Anzahl der besuchten Kurse!

```
SELECT COUNT(DISTINCT V_NR) Veranstaltungen  
FROM Veranstaltungsbesuche;
```

Listen sie alle Referenten, außer den Referenten Günter auf!

```
SELECT *  
FROM Referent  
WHERE NOT (name='Günter');  
  
Bzw. Name <> 'Günter'
```

Sortieren sie die Bezeichnungen der Veranstaltungen!

```
SELECT *  
FROM Veranstaltungen  
ORDER BY Bezeichnung;
```

Ordnen sie nun die Bezeichnungen in absteigender Reihenfolge (DESC)!

```
SELECT *  
FROM Veranstaltungen  
ORDER BY Bezeichnung DESC;
```

Geben sie das Durchschnittsgehalt aus!

```
SELECT AVG(Gehalt)
FROM REFERENT;
```

Geben sie das Durchschnittsgehalt der internen bzw. externen Referenten aus!

```
SELECT AVG(Gehalt)
FROM REFERENT
GROUP BY S;
```

Verschachtelte Abfragen

Diese benötigt man für den Fall, dass Abfragen auf dem Ergebnis vorgängiger Abfragen aufbauen.

Wir möchten nun wissen, welche Veranstaltungen Peter Schmidt besucht hat. Ermitteln sie in einem ersten Schritt die Matrikelnummer des Peter Schmidt und in einem weiteren dann die besuchten Veranstaltungen, welche zu dieser Matrikelnummer gehören und in einem dritten Schritt ermittelnd sie die Kursbezeichnungen!

```
SELECT V_nr, Bezeichnung
FROM Veranstaltungen
WHERE V_NR IN (
    SELECT V_NR
    FROM Veranstaltungsbesuche
    WHERE Matrikel_nr = (
        SELECT Matrikel_nr
        FROM Person
        WHERE Name='Schmidt'
        AND Vorname = 'Peter'));
```

Verschachtelte Abfragen werden immer von unten nach oben abgearbeitet.

Wie würde diese Abfrage als JOIN aussehen?

```
SELECT V_nr, Bezeichnung
FROM Veranstaltungen v, Veranstaltungsbesuche vb, Person p
WHERE v.V NR = vb.V NR
AND vb.Matrikel_nr = p.Matrikel_nr
AND Name='Schmidt'
AND Vorname = 'Peter';
```

Erstellen sie nun eine View welche die Veranstaltungen der Studenten auflistet!

```
CREATE VIEW Veranstaltungen_Studenten AS
SELECT V_nr, Bezeichnung
FROM Veranstaltungen
WHERE V_NR IN (
    SELECT V_NR
    FROM Veranstaltungsbesuche);
```

Fragen sie nun von dieser View die Veranstaltungen ab, welcher der Schmidt, Peter besucht hat.

```
SELECT *
FROM Veranstaltungen_Studenten
WHERE Name='Schmidt';
```

```
AND Vorname = 'Peter';
```