

**Erstellen sie die Statements für die Tabellen! Achten sie auf die richtige Reihenfolge beim Erstellen!**

**Diskutieren sie, welche Spalten der Tabellen Nullwerte aufweisen dürfen!**

**Definition Nullwert:**

**Wenn ein Attribut eines Tupels einen Nullwert enthält, bedeutet das, dass dieses Attribut keinen Attributwert besitzt und somit keine Information beinhaltet.**

**Der Nullwert darf nicht mit der Zahl Null verwechselt werden!**

```
CREATE TABLE Person (  
    Matrikel_Nr number,  
    Name varchar(50) not null,  
    Vorname varchar(30) not null,  
    S_ID number constraint Person_FK1 references Studiengang  
    (S_ID),  
    Semesteranzahl number not null,  
    constraint Person_Pk primary key(Matrikel_nr));
```

```
CREATE TABLE Studiengang  
    (S_ID number,  
    Studiengang varchar(40) not null);
```

```
alter table Studiengang  
    add constraint Studiengang_Pk primary key(S_ID);
```

```
CREATE TABLE Veranstaltungsbesuche (  
    Matrikel_nr number  
        constraint VeranstaltungsbesucheFK1 references Person  
    (Matrikel_nr),  
    V_Nr number  
        constraint VeranstaltungsbesucheFK2 references  
    Veranstaltung (V_Nr),  
    R_Nr number  
        constraint VeranstaltungsbesucheFK3 references  
    Referent (R_Nr),  
    Datum date);
```

**Vorher: Person, Veranstaltung, Referent**

```
CREATE TABLE Pflichtfaecher (  
    S_NR number constraint PflichtfaecherFK1 references  
    Studiengang (S_ID),  
    V_NR number constraint PflichtfaecherFK2 references  
    Veranstaltung (V_nr));
```

**Vorher: Veranstaltung, Studiengang = Fremdschlüsseltabellen!**

```
CREATE TABLE Referent (  
    R_Nr number,  
    P_Nr number,  
    S varchar(1) not null,  
    Name varchar(30) not null,  
    Vorname varchar(30) not null,  
    Firma varchar(50),  
    constraint Referent_pk primary key(R_Nr));
```

```
CREATE TABLE Veranstaltungen (  
    V_Nr number,  
    Bezeichnung varchar (30),  
    F_Nr number constraint VeranstaltungenFK references Fakultaet  
(F_Nr),  
    constraint Veranstaltungen_pk primary key(V_Nr));
```

```
CREATE TABLE Fakultaet (  
    F_Nr number,  
    Fakultaet varchar(20) not null,  
    constraint Fakultaet _pk primary key(F_Nr));
```

1. Person
2. Studiengang
3. Fakultät
4. Veranstaltungen
5. Referent
6. Veranstaltungsbesuche
7. Pflichtfächer

**Warum muss die Tabelle Studiengang vor der Tabelle Person angelegt werden?**

Weil die Tabelle Person einen Fremdschlüssel auf die Tabelle Studiengang enthält. Dies bezeichnet man auch als referentielle Integrität.

**Sie stellen fest, dass sie in der Tabelle Referent noch deren Gehalt benötigen. Ergänzen sie die Spalte mit folgendem Befehl!**

```
ALTER TABLE tabellennamen ADD  
(neuer_Attributname Datentyp [NOT NULL]);
```

```
ALTER TABLE referent ADD  
(gehalt number);
```

Fügen sie folgenden Datensatz ein:

```
INSERT INTO Person  
VALUES (15, Meier, Juergen, 2, 1);
```

Warum lässt sich dieser Datensatz nicht einfügen?

Weil der Fremdschlüssel in der Tabelle Studiengang fehlt!

```
INSERT INTO Studiengang VALUES (1,'BWL');
INSERT INTO Studiengang VALUES (2,'Wirtschaftsinformatik');
INSERT INTO Studiengang VALUES (3,'Mathematik');
INSERT INTO Studiengang VALUES (4,'Physik');
INSERT INTO Studiengang VALUES (5,'Chemie');
```

**Fügen sie erneut folgenden Datensatz ein!**

```
INSERT INTO Person
VALUES (15, Meier, Juergen, 2, 1);
```

**Warum lässt er sich nun einfügen?**

**Der benötigte Fremdschlüssel existiert nun!**

**Versuchen sie den Datensatz des Studiengangs Wirtschaftsinformatik zu löschen!**

```
DELETE FROM Studiengang
WHERE S_ID = 2;
```

**Warum lässt sich dieser Datensatz nicht löschen?**

**Warum lässt sich folgender Datensatz nicht anlegen?**

```
INSERT INTO Person
VALUES (15, Forner, Jan, 2, 1);
```

**Weil die Matrikel\_nr (Primärschlüssel) schon vergeben ist!**

**Warum lässt sich folgender Datensatz nicht anlegen?**

```
INSERT INTO Person (Matrikel_Nr, Vorname,S_ID,Semesteranzahl)
VALUES (20, Jan, 2, 1);
```

**Weil wir das Namensfeld als „not null“ - Feld definiert haben!**

**Fügen sie die Datensätze der ihnen vorliegenden Tabellen mit folgendem Befehl ein:**

```
INSERT INTO Person (Matrikel_Nr,Name,Vorname,S_ID,Semesteranzahl)
VALUES (&Matrikel_Nr, &Name, &Vorname,S_ID, &Semesteranzahl);
```

Diskutieren sie mit folgender Tabelle welche Dienststelle welche Rechte erhalten soll:

- Daten abfragen (S)
- Daten einfügen (I)
- Daten ändern (U)
- Daten löschen (D)

	Privilegien	Person	Studiengang	Fakultät	Veranstaltungsbesuche	Referent		Pflichtfächer
						Gehalt	Rest	
Studentensekretariat	S	√				√	√	
	I	√				√	√	
	U	√				√	√	
	D	√				√	√	√
Planbüro	S		√	√	√			√
	I		√	√	√			
	U		√	√	√			
	D		√	√	√			
Registrierbüro	S	√						√
	I							√
	U	√						√
	D							√

Connect

Mit dem Recht Connect darf man in die Datenbank einsteigen und Datenverändern, aber keine Tabellen erstellen

**Legen sie einen neuen Nutzer Studentensekretariat an!**

X = Platznummer

**GANZ WICHTIG Platznummer X!!!**

GRANT CONNECT TO StudentensekretariatX  
 IDENTIFIED BY StudentensekretariatPasswort;

Geben sie dem Studentensekretariat die Rechte, die Studiengänge abzufragen!

```
GRANT SELECT
ON Studiengang
TO Studentensekretariat;
```

Logen sie sich aus und als Studentensekretariat ein! (Achtung StudentensekretariatX)

**Fragen sie die Daten des Studiengangs ab!**

Select \* from studiengang;

**Warum kommt hier eine Fehlermeldung?**

Weil der Nutzer Studentensekretariat keine Tabelle Studiengang angelegt hat!

```
Select * from userX.studiengang;
```

userX.studiengang = bedeutet, er greift auf die Tabelle Studeingang des Benutzers UserX zu.

Fragen sie den Inhalt der Tabelle Studenten ab!

Warum ist das nicht möglich?

Weil wir dem Nutzer Studentensekretariat keine Rechte auf diese Tabelle erteilt haben!

Da dies sehr aufwendig ist, vergeben wir ein Synonym:

```
CREATE SYNONYM Studiengang  
FOR userX.studiengang;
```

Warum funktioniert jetzt die Abfrage `Select * from studiengang`?

Weil wir für „userX.studiengang“ ein Synonym vergeben haben