

Eine Universität möchte den Studienverlauf ihrer Studenten für Auswertungszwecke verwalten. Es soll erfasst werden, welche Studenten, welche Veranstaltungen wann besucht haben und wer die Veranstaltung gehalten hat sowie die Pflichtfächer für jeden Studiengang. Die Studentendaten werden im Studentensekretariat, die Veranstaltungen vom Planbüro und die Veranstaltungsbesuche vom Registrierbüro verwaltet. Es gelten folgende weitere Bedingungen:

- Jede Person ist mit Personalnummer, Name und Vorname einzutragen.
- Für alle Studenten sind ergänzend Studiengang und Semesteranzahl zu erfassen.
- Einige Studenten können auch selbst Referenten sein (Als Hiwi ☺)
- Bei uni-eigenen Referenten sollen die Dienstjahre mit gespeichert werden.
- Einige Veranstaltungen können auch von externen Betreuern gehalten werden. Hier sollen der Name, der Vorname und der Firmenname erfasst werden. Diese sollen aber erst aufgenommen werden, wenn sie schon eine Veranstaltung in der Uni gehalten haben.
- Für die Veranstaltungen sind die Veranstaltungsnummer, die Bezeichnung und die Fakultät zu speichern.
- Jede Veranstaltung ist mit dem Teilnehmern, dem Referenten und dem Datum des Veranstaltungsbeginn festzuhalten.

1. Bilden sie Entitätsmengen

- Personen, Veranstaltung, Kursleiter

2. Legen sie die Beziehungen fest. Schreiben sie sich unklare Beziehungen auf!

- Lasse sie folgende Annahmen sammeln!

- rekursive Beziehung für uni-eigene Referenten
- eine Person kann Kursleiter sein oder nicht
- ein uni-eigener Referent entspricht genau einer Person
- Wie oft kann ein Referent bestimmte Kurse geben → Annahme beliebig oft
- Kann ein Kurs von beliebig vielen internen Referenten durchgeführt werden? → Annahme: ja (z. B. ABWL)
- Externe Referenten müssen mindestens einen Kurs durchgeführt haben.
- Kann ein Kurs von beliebig vielen externen Referenten durchgeführt werden? → Annahme: ja
- Muss ein Kursleiter mindestens die Veranstaltung vor einer Person gehalten haben? → Annahme: ja

- Achtung Beziehungsfalle:

- Beispiel: m-mc Beziehung Person – Veranstaltung
- Würde bedeuten, dass sofort jeder Person eine Veranstaltung zugeordnet werden müsste!
- Praxis: es werden erst alle Kurse aufgenommen!
- Nachteil: es können Veranstaltungen ohne Teilnehmer durchgeführt werden!
 - Muss in der Anwendung dann verhindert werden

3. Legen sie Identifikationsschlüssel fest!

- Für jede Entitätsmenge ist ein natürlicher oder ein künstlicher Schlüssel festzulegen.
- Natürlicher Schlüssel: es wird ein vorhandenes Attribut verwendet (z.B. Personalnummer, Matrikelnummer)
- Künstlicher Schlüssel: es wird eine neues Attribut eingeführt
- Person → Personalnummer (P_Nr)
- Kurs → Kursnummer (K_Nr)

4. Legen sie für jede Entitätsmenge die Attribute fest!

- Person (Matrikel_Nr,Name,Vorname,S_ID,Semesteranzahl)
- Studiengang (S_ID,Studiengang)
- Fakultät (F_Nr,Fakultät)
- Veranstaltungsbesuche (V_Nr,Bezeichnung,F_Nr)
- Referent (R_Nr,P_Nr,S,Name,Vorname,Firma)
- Veranstaltungsbesuche (P-Nr,V_Nr,R_Nr,Datum)
- Pflichtfächer (S_NR, V_NR)

5. Welche Transaktionen können für die Anwendung definiert werden?

- Daten abfragen
- Daten einfügen
- Daten ändern
- Daten löschen

- Beispiel-Transaktionen
- Einfügen. Löschen und Nachführen von Personen
- Einfügen. Löschen und Nachführen von Kursdaten
- Einfügen. Löschen und Nachführen von Daten in die Tabelle Pflichtfächer
- Einfügen eines Referenten in die Tabelle Referent
- Abfragen der Daten der Referenten (Erklären lassen, dass hiermit der Join entfällt und s.u.)

Praxis: Es werden erst die geforderten Abfragen erstellt
Warum? Damit man ein sinnvolles Datenbankdesign erhält.
Man kann bei häufigen Anfragen schon vorher das Joinen umgehen.

Warum wurde die doppelte Erfassung der Referenten, welche auch Studenten sind in Kauf genommen?

- ist bei der praktischen Implementierung günstiger und benutzerfreundlicher
- Vereinfachung der Abfragen (Beispiel: Es sollen die Namen aller Referenten ausgegeben werden)
- Reduzierung von Fremdschlüsseln

Unterstreichen sie in den ihnen vorliegenden Tabellen die Primärschlüssel einfach die Fremdschlüssel doppelt!

Legen sie für alle Tabellen den Datentyp fest!

Erstellen sie die Staments für die Tabellen!

```
CREATE TABLE Studiengang  
(S_ID number,  
Studiengang varchar(40));
```

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

```
CREATE TABLE Person (  
Matrikel_number,  
Name,Vorname varchar(30),  
S_ID number,  
Semesteranzahl number);
```

```
alter table Person  
add constraint Person_Pk primary key(Matrikel_number);
```

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.

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)

| | <u>Privilegien</u> | <u>Person</u> | Studiengang | Fakultät | Veranstaltungs besuche | Referent | Pflichtfächer |
|--------------------------|--------------------|---------------|-------------|----------|---------------------------|----------|---------------|
| Studenten sekretariat | S | | | | | | |
| | I | | | | | | |
| | U | | | | | | |
| | D | | | | | | |
| Planbüro | S | | | | | | |
| | I | | | | | | |
| | U | | | | | | |
| | D | | | | | | |
| Studenten sekretariat | S | | | | | | |
| | I | | | | | | |
| | U | | | | | | |
| | D | | | | | | |