

Die Online-Befragung zur genutzten alternativen Veranstaltungsform und zur Lehrevaluation ist online. Bitte ausfüllen: <https://forms.gle/ZzvVvTy6fqgHVmg6>. Sie werden eventuell aufgefordert sich bei Google anzumelden, das ist nur notwendig, wenn Sie in der Bearbeitung eine Pause machen wollen und das Teilergebnis zwischenspeichern wollen. Die Befragung endet am 22.12., die Ergebnisse stehen in einem nachfolgenden Fragen&Antworten-Dokument auf der Webseite der Veranstaltung.

Hinweis: Lesen Sie vor der Entwicklung den Leitfaden (<http://kleuker.iui.hs-osnabrueck.de/querschnittlich/DBNutzung.pdf>) von der Veranstaltungswebseite zur Entwicklung von Prozeduren und Trigger.

In diesem Praktikum werden folgende Tabellen genutzt. Der SQL-Code ist mit Beispieldatensätzen von der Veranstaltungsseite erhältlich. Überlegen Sie sich, wie Sie Ihre Funktionen und Prozeduren testen können. Der Testcode gehört zur Abnahme dazu.

```
CREATE TABLE Studierend(  
  matNr INTEGER,  
  name VARCHAR(20) NOT NULL,  
  CONSTRAINT PK_Studierend PRIMARY KEY (matNr)  
);
```

```
CREATE TABLE Modul(  
  mid INTEGER,  
  titel VARCHAR(32) NOT NULL,  
  semester INTEGER NOT NULL,  
  cp INTEGER NOT NULL,  
  CONSTRAINT PK_Modul PRIMARY KEY (mid),  
  CONSTRAINT bachelorsemester CHECK (semester > 0 AND semester < 7),  
  CONSTRAINT leistungspunkte CHECK (cp > 0 AND cp < 30)  
);
```

```
CREATE TABLE Pruefung (  
  matNr INTEGER,  
  mid INTEGER,  
  datum DATE NOT NULL,  
  versuch INTEGER,  
  note INTEGER NOT NULL,  
  CONSTRAINT PK_Pruefung PRIMARY KEY (matNr,mid,versuch),  
  CONSTRAINT FK_Pruefung_Studierend FOREIGN KEY (matNr)  
    REFERENCES Studierend(matNr),  
  CONSTRAINT FK_Pruefung_Modul FOREIGN KEY (mid)  
    REFERENCES Modul(mid),  
  CONSTRAINT Pruefung_eineProTag UNIQUE (matNr,mid,datum),  
  CONSTRAINT Pruefung_Versuch CHECK (versuch in (1,2,3)),  
  CONSTRAINT Pruefung_Note  
    CHECK (note in (100,130,170,200,230,270,300,330,370,400,500))  
);
```

Aufgabe 27 (9 Punkte)

- a) Schreiben Sie eine Prozedur `ein fuegenStudierend(Matrikelnummer, Name)`, die einen Wert in die Tabelle Studierend einfügt. Was passiert, wenn Sie

- keinen oder einen ungültigen Wert für Name oder einen bereits vergebenen Schlüssel eingeben?
- Schreiben Sie eine Prozedur **einFuegenStudierend2 (Name)**, die einen Wert in die Tabelle Studierend einfügt. Dabei soll die Matrikelnummer automatisch berechnet werden, überprüfen Sie vorher, ob es überhaupt schon einen Tabelleneintrag gibt und reagieren Sie wenn nötig.
 - Schreiben Sie eine Funktion **modulGescheitert (Modul_ID)**, die für ein mit seiner mid identifiziertes Modul die Anzahl der nicht bestandenen Prüfungen in diesem Modul als Ergebnis liefert.
 - Schreiben Sie eine Funktion **studierendLeistungspunkte (Matrikelnummer)**, die für einen Studierenden seine bisher erreichten Leistungspunkte berechnet, dabei werden nur die Punkte in bestandenen Fächern berücksichtigt.
 - Schreiben Sie eine Funktion **imDrittversuch (Modul_ID)**, die für ein mit seiner mid identifiziertes Modul die Anzahl der Studierenden ausgibt, die noch einen dritten Versuch machen müssen, also den zweiten Versuch nicht bestanden und den dritten Versuch noch nicht durchgeführt haben.
 - Schreiben Sie eine Funktion **studierendschnitt (Matrikelnummer)**, die für einen Studierenden seine Durchschnittsnote berechnet, dabei werden nur Noten in bestandenen Fächern berücksichtigt. Beachten Sie, dass die Note mit der Leistungspunktzahl (CP) gewichtet wird. Hat ein Studi z. B. ein Fach mit 10 CP mit einer 330, ein Fach mit 5 CP mit 130 bestanden und ein Fach mit 5 CP mit 170 bestanden, ist die Durchschnittsnote $(10*(330) + 5*(130+170)) / (1*10 + 2*5)$, also 240.

Hinweise:

Überlegen Sie bei d) bis h) zunächst eine passende SQL-Anfrage, die Sie dann in ein Programm einbetten.

Aufrufmöglichkeit für Funktionen:

```
SELECT modulschnitt(101) FROM SYSIBM.SYSDUMMY1;
```

Zur Umwandlung von Datentypen kann CAST, z. B. CAST(note as DOUBLE) genutzt werden, z. B.: **SELECT AVG(CAST (note AS DOUBLE))**

Sie dürfen selbstgeschriebene Prozeduren und Funktionen in anderen Aufgabenteilen nutzen; diese können auch in SQL-Anfragen eingebaut werden.

Ergebnisse:

Für a)-b) überlegen Sie sich ein eigenes Vorgehen, für die restlichen Methoden sollen folgende Anfragen und Ergebnisse zur Selbstüberprüfung genutzt werden, wenn *nur* die Daten aus der Ausgangsdatei eingespielt sind.

```
SELECT mid, modulGescheitert(mid) FROM Modul;
```

MID	2
11	8
12	11
13	0
14	0
21	1
22	1
23	0
24	0
25	0
31	0
32	0

33 0
34 0
35 0
36 0

```
SELECT matNr, studierendLeistungspunkte(matNr) FROM Studierend;
```

MATNR 2

444 55
445 40
446 20
447 40
448 90
449 15
450 15
451 0

```
SELECT mid, imDrittversuch(mid) FROM Modul;
```

MID 2

11 1
12 4
13 0
14 0
21 0
22 0
23 0
24 0
25 0
31 0
32 0
33 0
34 0
35 0
36 0

```
SELECT matNr, studierendschnitt(matNr) FROM Studierend;
```

MATNR 2

444 265.45454545454544
445 216.25
446 342.5
447 187.5
448 202.22222222222223
449 343.33333333333333
450 376.66666666666667
451 <NULL>