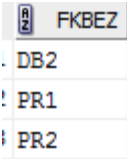
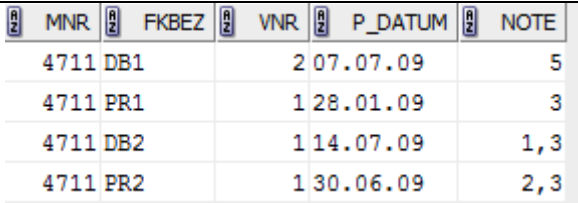


**Aufgabe 1**

Gib das Ergebnis folgender SQL-Anweisungen an (Tabellen s. gesondertes Datenblatt):

	Ergebnis <b>hier</b> eintragen
a) SELECT MAX(COUNT(*)) FROM KLAUSUR GROUP BY FKBEZ HAVING AVG(NOTE) < 4;	<b>2</b>
b) SELECT FKBEZ FROM STUDENT S CROSS JOIN KLAUSUR K WHERE S.MNR = 4712 MINUS SELECT FKBEZ FROM KLAUSUR K WHERE MNR = 4712 ORDER BY FKBEZ;	
c) SELECT MNR, FKBEZ, VNR, P_DATUM, NOTE FROM KLAUSUR WHERE (MNR, FKBEZ, VNR) IN (SELECT MNR, FKBEZ, MAX(VNR) FROM KLAUSUR GROUP BY MNR, FKBEZ) AND MNR = 4711;	

**Aufgabe 2**

Gegeben sei das beigegefügte Datenblatt mit den Tabellen STUDENT, FACH und KLAUSUR.

Es soll eine Liste aller Studenten ausgegeben werden, die in der Klausur DB1 besser abgeschnitten haben als der Durchschnitt der anderen Studenten in der gleichen Klausur zum gleichen Termin.

Ausgabe von *mnr*, *nachname*, *fkbez*, *p\_datum*, *note* sortiert nach *mnr*.

Trage Deine Lösung **hier** ein:

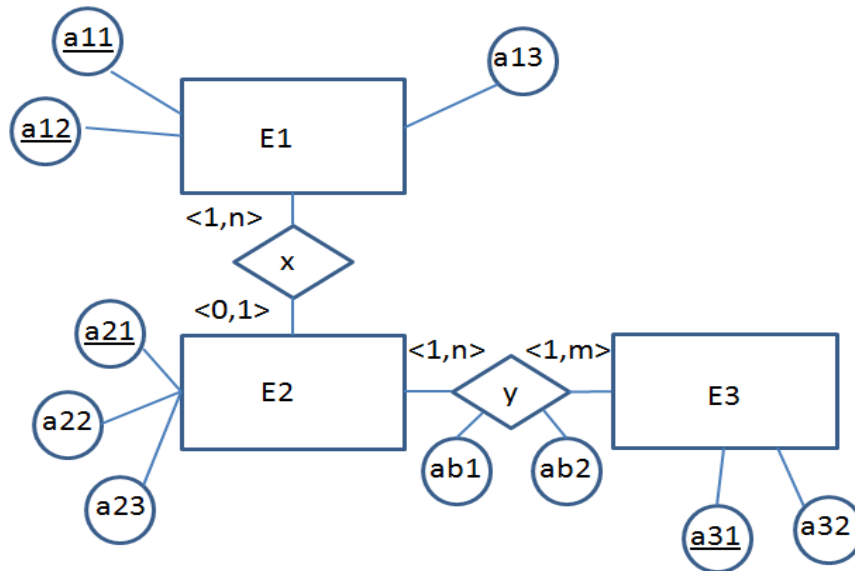
```
select s.mnr, s.nachname, k1.fkbez, k1.p_datum, k1.note
from student s
join klausur k1
on k1.mnr = s.mnr
where k1.note < (select avg(note)
                 from klausur k2
                 where k2.p_datum = k1.p_datum
                   and k2.fkbez = k1.fkbez
                   and k2.mnr <> k1.mnr)
and k1.fkbez = 'DB1';
```

**Aufgabe 3**

Gib durch ankreuzen in der Spalte 'richtig' bzw. 'falsch' an, ob die Behauptung zutrifft oder nicht.

Nr	Aussage	richtig	falsch
1	DB-Tabellen müssen einen Primärschlüssel haben.		x
2	MS Access ist eine Datenbank.		x
3	SQL ist eine Programmiersprache.		x
4	Ein Datenbankschema kann mit Hilfe der Anweisung UPDATE geändert werden		x
5	Integritätsbedingungen gehören zum Schema einer Datenbank.	x	
6	Jeder Join lässt sich aus Kreuzprodukt und Selektion zusammensetzen	x	
7	Ein UNIQUE Constraint hat die gleiche Wirkung wie ein Primary Key		x
8	Es sind mehrere UNIQUE Constraints pro Tabelle zulässig	x	
9	Pro Tabelle darf es nur einen einzigen Primary Key geben	x	
10	Der Vergleich zweier Spaltenwerte, die beide Null sind, ergibt wahr.		x
11	Bei COUNT(S) auf einer einzelnen Spalte S werden Nullwerte nicht gezählt.	x	
12	Mit Hilfe des ERM wird eine 100 %ige Speicherauslastung ermöglicht.		x
13	Beziehungen zwischen Relationen werden durch Kardinalitäten angegeben		x
14	Indexe führen i.d.R. zu einer besseren Performance bei Schreibzugriffen.		x
15	Ein LEFT Join liefert immer ein anderes Ergebnis als ein RIGHT Join		x
16	In einer Join-Bedingung darf man den Operator BETWEEN nicht verwenden		x
17	Gegeben sei eine Tabelle X mit einer Spalte x und dem einzigen Wert '*' Die Anweisung SELECT '2'    x    '2 = '    3*2 FROM X; liefert folgendes Ergebnis: '2*2 = 6		x
18	Gegeben sei eine Tabelle X mit einer Spalte x und dem einzigen Wert 'x' Die Anweisung SELECT COUNT(*) FROM (SELECT COUNT(null+null) FROM X); liefert folgendes Wert als Ergebnis: null		x
19	Die Berechnung der Zahl Pi wäre eine typische Anwendung, die den Einsatz von Transaktionen erfordert		x
20	Eine Schemaänderung an einer Datenbank wären eine typische Anwendung, die den Einsatz von Transaktionen erfordert		x
21	Die Anweisung Revoke dient zum morgendlichen Start der Datenbank		x
22	Eine Abteilung, ohne Mitarbeiter ist ein typisches Beispiel für eine schwache Entität.		x
23	1:N Assoziationen können keine Attribute haben.		x
24	Unter „Abgeschlossenheit“ von SQL versteht man, dass man SQL-Anweisungen nicht aus anderen Sprachen aufrufen kann		x
25	Fremdschlüssel sind immer gleichzeitig Primärschlüssel in einer anderen Tabelle		x

**Aufgabe 4** Gegeben sei folgendes ERD



Dieses ERD soll in eine relationale Datenbank umgesetzt werden. Gib die CREATE TABLE Anweisungen einschließlich NOT NULL, PRIMARY KEY und FOREIGN KEY Constraints an. Alle Attribute seien vom Typ INT.  
Trage Deine Lösung **hier** ein:

```
create table E1(
  a11  int  not null,
  a12  int  not null,
  a13  int  not null,
  constraint PK_E1 primary key(a11,a12)
);

create table E2(
  a21  int  not null,
  a22  int  not null,
  a23  int  not null,
  a2_fk1 int null,
  a2_fk2 int null,
  constraint PK_E2 primary key(a21),
  constraint FK_E1 foreign key(a2_fk1,a2_fk2)
  references E1(a11,a12)
);

create table E3(
  a31  int  not null,
  a32  int,
  constraint PK_E3 primary key(a31)
);
```

**Datenblatt****Bedeutung der Attribute:**

EXM_DATUM	Exmatrikulationsdatum
FBESCHREIBUNG	Langbezeichnung eines Fachs
FKBEZ	Kurzbezeichnung eines Fachs
GEB_DATUM	Geburtsdatum
IMM_DATUM	Immatrikulationsdatum
MNR	Matrikelnummer
NOTE	Ergebnis der Prüfung eines Studierenden
P_DATUM	Zeitpunkt, zu dem die Prüfung (z.B. Klausur) erfolgte.
VNR	Versuchsnummer; gibt an, um den wievielten Versuch eines Studenten, einen Leistungsnachweis zu erwerben, es sich handelt.

**Tabelle STUDENT**

MNR	NACHNAME	VORNAME	GEB_DATUM	IMM_DATUM	EXM_DATUM
4711	Lehmann	Heini	13.03.89	01.09.08	(null)
4712	Huber	Sven	14.07.89	01.09.08	(null)
4713	Meier	Swantje	11.04.88	01.03.09	(null)
4714	Tunix	Ole	15.03.88	01.03.09	(null)
4715	Kannix	Peter	02.11.89	01.03.09	(null)
4716	Weissnix	Axel	15.12.88	01.03.09	(null)

**Tabelle FACH**

FKBEZ	FBESCHREIBUNG
PR2	Programmieren 2
PR1	Programmieren 1
DB3	Datenbanksysteme 3
DB2	Datenbanksysteme 2
DB1	Datenbanksysteme 1

**Tabelle KLAUSUR**

MNR	FKBEZ	VNR	P_DATUM	NOTE
4711	DB1	1	02.02.08	4,7
4711	DB1	2	07.07.09	5
4711	PR1	1	28.01.09	3
4712	DB1	1	02.02.08	3,7
4713	DB1	1	02.02.08	1,7
4713	DB2	1	02.02.09	3,7
4714	PR1	1	28.01.09	2
4715	DB1	1	02.02.08	5
4711	DB2	1	14.07.09	1,3
4711	PR2	1	30.06.09	2,3