

DB1 Probeklausur

(basierend auf einer Vorlage von Prof. Dr. Ingo Stengel)

Name.....

MatrNo.....

Wichtig!

Die Klausur besteht aus vier Aufgaben. Für eine maximale Punktezahl (90 Punkte) sind alle Aufgaben zu lösen.

Zur Lösung der Aufgabe 4 benötigen Sie die Beschreibung der Tabellen der Mondial-Datenbank, siehe letzte Seite.

Zulässige Hilfsmittel – Notizzettel (DIN-A4, Vorder- und Rückseite) mit handgeschriebenen Notizen.

Aufgabe 1 – Relationale Algebra (25 Punkte)

Es seien folgende Tabellen gegeben:

Tabelle Firma

FirmenID	FirmenName	AuftragsNr
FirmaA	HSStartup	1
FirmaA	HSStartup	2
FirmaB	DBStartup	3
FirmaB	DBStartup	4

Tabelle Auftrag

AuftragsNr	Auftragstitel	TätigkeitsNr
1	Auftrag1	10
2	Auftrag2	11
3	Auftrag3	10
4	Auftrag4	11

Tabelle Tätigkeit

TätigkeitsNr	TätigkeitsTitel	TätigkeitsBeschreibung
10	Programmieren	Programmieren im Kundenauftrag
11	Testen	Testen der erstellten Software

Wichtige Anmerkung: Felder die den gleichen Namen in unterschiedlichen Tabellen haben enthalten die gleichen Werte in der gesamten Datenbank!

z.B. AuftragsNr in der Tabelle Firma ist identisch mit AuftragsNr in der Tabelle Auftrag, usw.

- Stellen sie folgende relationale Abfrage (Formel): Listen Sie alle Firmennamen.

$$\pi_{FirmenName}(Firma)$$

- Stellen sie eine relationale Abfrage (Formel): In welchen Firmen wird programmiert?

$$\pi_{FirmenName} \left(\sigma_{T\u00e4tigkeitsTitel="Programmieren"}(Firma \bowtie Auftrag \bowtie T\u00e4tigkeit) \right)$$

- Stellen sie eine relationale Abfrage (Formel): Listen Sie den Firmennamen und immer zwei T\u00e4tigkeitsTitel von unterschiedlichen von der jeweiligen Firma ausgef\u00fchrten T\u00e4tigkeiten pro Zeile.

Anmerkung: Unter Umst\u00e4nden kann der Firmenname doppelt vorkommen.

$$Temp \leftarrow Firma \bowtie Auftrag \bowtie T\u00e4tigkeit$$

$$\pi_{Temp.FirmenName, Temp.T\u00e4tigkeitstitel, Temp2.T\u00e4tigkeitstitel} \left(\sigma_{\substack{(Temp.FirmenID=Temp2.FirmenID) \\ \wedge (Temp.T\u00e4tigkeitstitel \neq Temp2.T\u00e4tigkeitstitel)}} \left(Temp \times_{Q_{Temp2}}(Temp) \right) \right)$$

Aufgabe 2 – Modellierung (25 Punkte)

Modellieren Sie folgenden Sachverhalt aus der Hochschullandschaft:

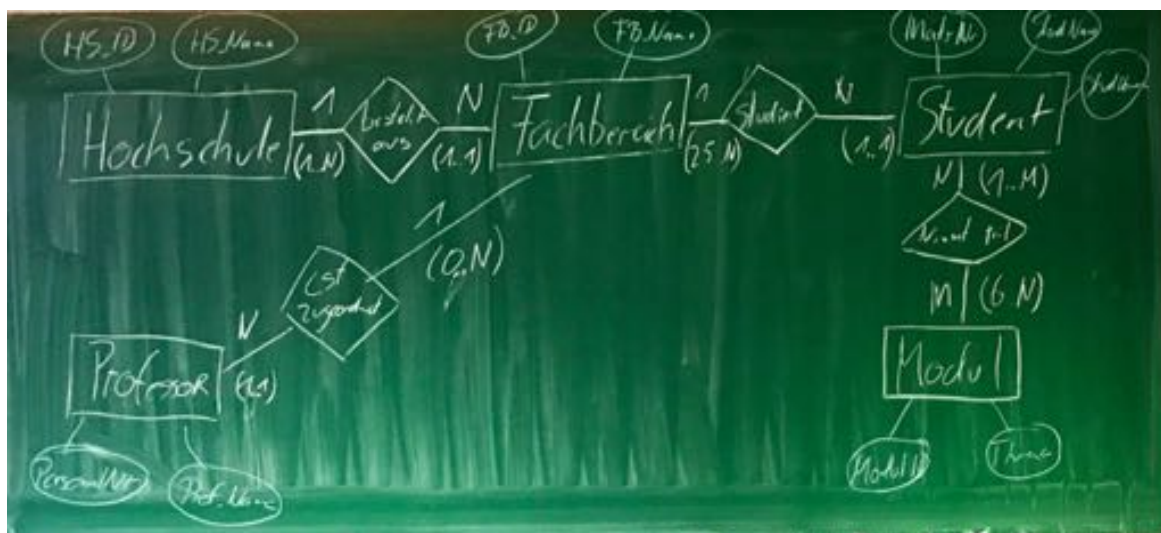
Jede Hochschule besteht aus mehreren Fachbereichen. Dabei muss jede Hochschule mindestens einen Fachbereich besitzen. Jede Hochschule hat Studenten und Professoren.

Jeder Student kann – zur gleichen Zeit - nur in einem Fachbereich studieren. Jedoch hat jeder Fachbereich mindestens 25 Studenten. Jeder Student kann mehrere Module belegen, muss aber immer mindesten ein Modul belegen. An einem Modul nehmen mindesten 6 Studierende teil. In der Regel sind es mehr Studierende.

Jeder Professor ist einem und nur einem Fachbereich zugeordnet. Dabei hat jeder Fachbereich in der Regel mehrere Professoren.

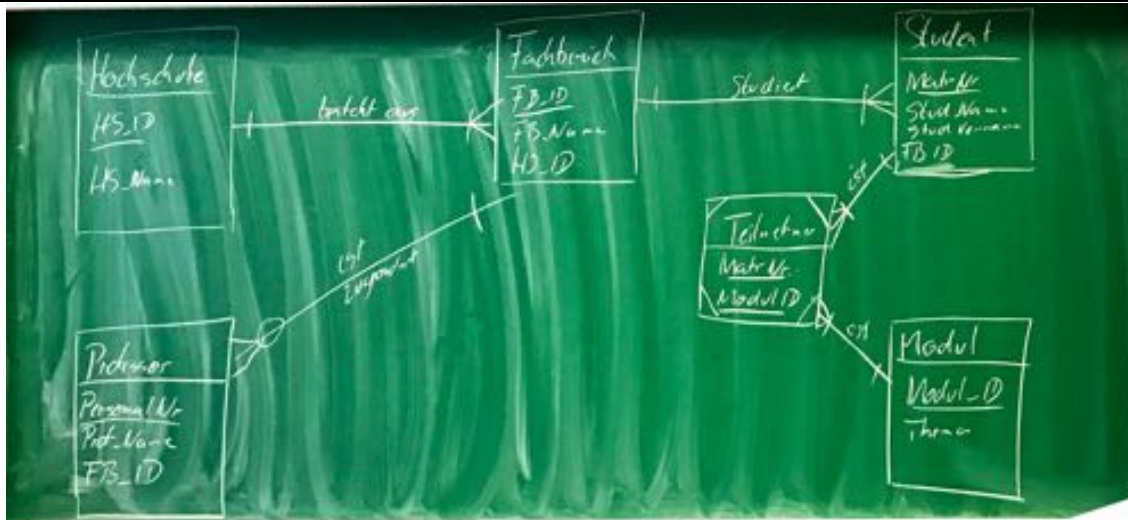
- Identifizieren Sie die Objekte (und deren Attribute), Relationen und Kardinalitäten und Konnektivitäten. Zeichnen Sie ein ER-Diagramm für ein konzeptionelles Modell in Chen-Notation.

ER-Diagramm konzeptionelles Modell



(Anmerkung: Beziehungen Hochschule <-> Professor, Hochschule <-> Student nicht eingezeichnet)

- Erweitern Sie das Modell zu einem internen Modell für ein Relationales Datenbankmanagementsystem und zeichnen Sie dieses in Crow's Foot Notation. Ergänzen Sie dabei Primär- und Fremdschlüssel.



(Anmerkung: Beziehungen Hochschule <-> Professor, Hochschule <-> Student nicht eingezeichnet)

Aufgabe 3 – Abhängigkeitsdiagramm und Normalformen (20 Punkte)

! Lesen Sie aufmerksam die Beschreibung!

Es sei folgende Tabelle zur Verwaltung der Strafgebühren an Ihrer Hochschule gegeben:

HS ID	HS Name	Stud Matr Nr	Stud Name	Stud Vorn	Ort	PLZ	Nicht rechtz Zurück Bücher	Gebühr proBuch	Insgesamt Gebühr
-------	---------	--------------	-----------	-----------	-----	-----	----------------------------	----------------	------------------

Beschreibung:

HSID – Stellt die ID der Hochschule dar. Sie ist Deutschlandweit eindeutig.

HSName – enthält den Namen der Hochschule.

StudMatrNr – enthält die Matrikelnummer eines Studenten und ist nur innerhalb einer Hochschule eindeutig.

StudName – enthält den Namen des Studenten.

StudVorn – enthält den Vornamen des Studenten.

Ort – enthält den Wohnort des Studenten.

PLZ – enthält die Postleitzahl des Wohnortes des Studenten.

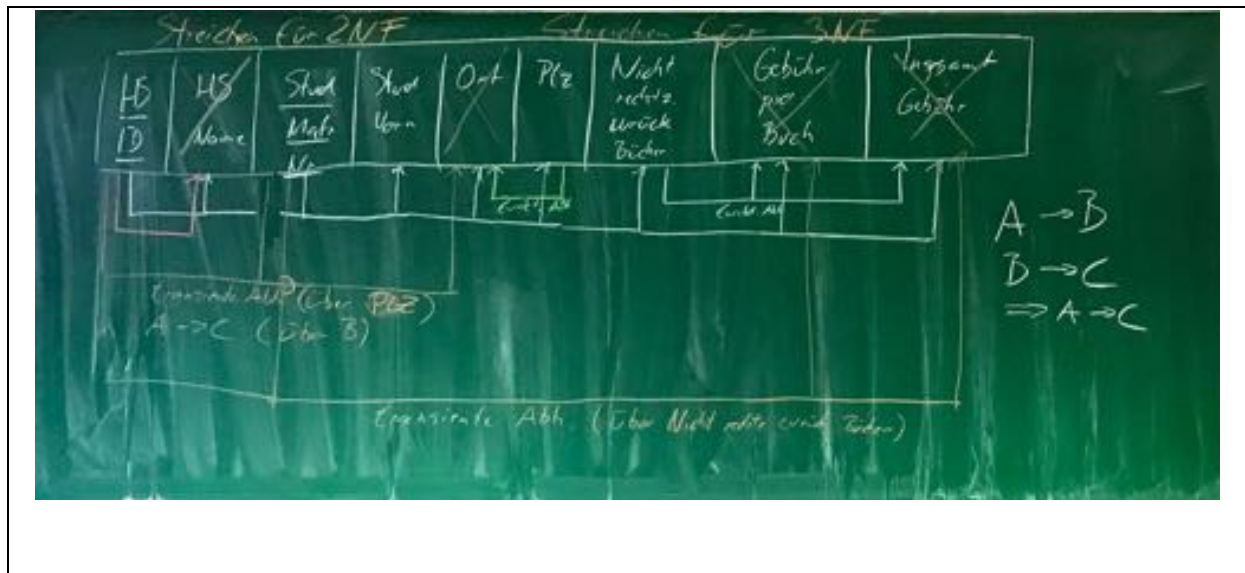
NichtrechtzZurückBücher – enthält die Anzahl der ausgeliehenen Bücher die der Student nicht rechtzeitig zurückgegeben hat.

GebührproBuch – ist die Strafgebühr pro Buch die erhoben wird. Diese ist von der Anzahl der nicht rechtzeitig zurückgegebenen Bücher abhängig. Bei 1-3 Bücher ist sie 3 Euro pro Buch, bei mehr als 4 Bücher ist sie 4 Euro pro Buch (für alle Bücher).

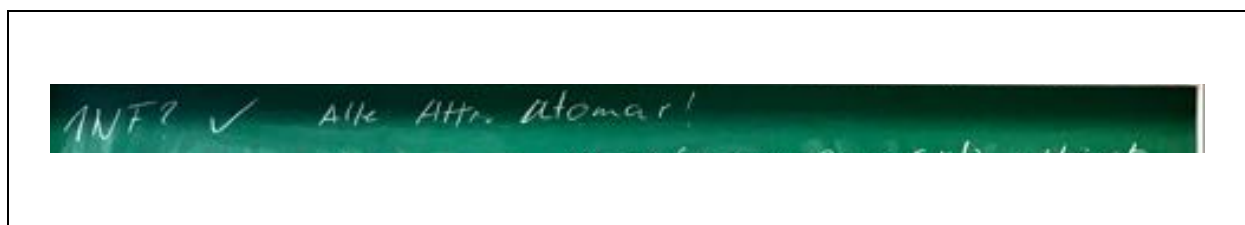
InsgesamtGebühr – Diese Gebühr berechnet sich aus der NichtrechtzZurückBücher multipliziert mit GebührproBuch.

Aufgaben:

- Erstellen Sie ein Abhängigkeitsdiagramm indem Sie:
 - die primären Schlüssel identifizieren (unterstreichen)
 - alle möglichen Abhängigkeiten eintragen.



- Ist die Tabelle in der 1NF? Falls nicht, so bringen Sie die Tabelle in die erste Normalform.



- Bringen Sie die Tabelle in die zweite Normalform. Listen Sie Tabellen, Attribute und Schlüssel.

2NF? X HS_Name von HS_ID (Teil des Primär-Schl!) abhängt

<u>HS_ID</u>	HS_Name
--------------	---------

(Anmerkung: Zur Lösung gehört auch die Tabelle aus dem 3. Aufgabenteil das Attribut HS_Name)

- Bringen Sie die Tabellen in die dritte Normalform. Listen Sie Tabellen, Attribute und Schlüssel.

3NF? X u.A. Ort trans von P.S. abhängt (über PLZ)

<u>PLZ</u>	ORT
------------	-----

Nicht rückg. Buch	Gebühr pro Buch	Insgesamt Gebühr
-------------------------	-----------------------	---------------------

(Anmerkung: Zur Lösung gehört auch die Tabelle aus dem 1. Aufgabenteil ohne die Attribute Ort, Gebühr pro Buch, Insgesamt Gebühr sowie ohne HS_Name aus 3. Aufgabenteil)

Aufgabe 4 – SQL Abfragen (20 Punkte)

Für die Bearbeitung dieser Aufgabe ist der Anhang Referential Dependencies of the Mondial Database (letzte Seite) notwendig.

- Definieren Sie eine SQL-Abfrage die alle Seen (Lakes) mit einer Tiefe (Depth) kleiner als 100m listet. Geben Sie das Ergebnis mit der Überschrift „Seen“ aus.

```
SELECT Name AS Seen
FROM Lake
WHERE Depth < 100;
```

- Definieren Sie eine SQL-Abfrage die alle Seen aus Deutschland in einer absteigenden alphabetischen Reihenfolge ausgibt.

Hinweis: Achten Sie darauf das Germany nur in der Tabelle Country im Namen steht. In anderen Tabellen finden sich Kürzel. Die Attributen Code und Country verwendet werden solche Kürzel. Das Kürzel für Deutschland ist 'D'.

```
SELECT Lake
FROM Geo_Lake
WHERE Country = 'D'
ORDER BY Lake DESC;
```

- Definieren Sie eine SQL-Abfrage die uns die Inseln mit der Stadt mit der kleinsten Bevölkerung (population) findet. Nutzen Sie hierzu die Funktion Min(). Geben Sie dabei den Namen der Insel, den Namen der Stadt auf der Insel und die Bevölkerung der Stadt aus.

Hinweis: diese Aufgabe kann nicht mit einem einfachen Select gelöst werden!

```
WITH CityIsland AS
  (SELECT LocatedOn.Island, City, Population
   FROM City JOIN LocatedOn ON City.Name = LocatedOn.City)
SELECT * FROM CityIsland
WHERE Population =
  (SELECT MIN(Population) FROM CityIsland);
```

Referential Dependencies of the Mondial Database

