

Name	Vorname	MatrNr
		Stud.-Gang

Bearbeitungszeit: 90 Minuten, Uploadzeit: 15 Minuten

Laden Sie alle Lösungsdateien in Moodle hoch.

Die Aufgaben können auf Papier oder mit den in den Übungen verwendeten Softwaretools bearbeitet werden.

Erlaubte Dateitypen: JERM-Datei, .jpg, .gif, .tiff, pdf, .txt, .sql

Verwenden Sie die Aufgabennummer als Dateiname, z.B. A123.pdf beinhaltet die Aufgaben 1-3

Erlaubte Hilfsmittel: alle

Aufgabe 1 (30 Punkte)

Ein Pizza-Service möchte ein Onlinesystem zur Bestellung von Pizzas entwickeln. Zu diesem Zweck soll ein Datenbanksystem entwickelt werden, in dem die angebotenen Pizzas, deren Preise und Bestellungen gespeichert werden.

Die Pizzeria hat 20 Pizzas, alle haben einen eindeutigen Namen, z.B. Margherita, Salami. Da Pizzas Allergene und Zusatzstoffe (z.B. Gluten, Geschmacksverstärker) beinhalten können, soll gespeichert werden, welche Pizzas welche solcher Stoffe enthalten.

Alle Pizzas gibt es in drei Größen: Mini, Standard und XL. Für alle Pizza/Größen-Kombinationen sollen die jeweiligen Preise gespeichert werden.

Ein Kunde kann sich mit Mailadresse und Adresse registrieren und anschließend Bestellungen aufgeben. Bei einer Bestellung kann er mehrere Pizzas in verschiedenen Größen und Anzahl bestellen. Zu jeder Bestellung wird die Bestellzeit und die Lieferzeit gespeichert.

Es gibt Geschenkgutscheine über einen bestimmten Eurobetrag, die bei der Bestellung eingelöst werden können. Ein Geschenkgutschein besitzt neben dem Eurobetrag einen eindeutigen Code, den man zum Einlösen bei der Bestellung angeben muss. Da der Eurobetrag des Geschenkgutscheins höher als der Bestellpreis sein kann, ist es möglich, einen Geschenkgutschein für mehrere Bestellungen zu verwenden. Genauso ist es möglich, dass bei einer Bestellung mehrere Gutscheine eingelöst werden. Aus diesem Grund muss pro Bestellung gespeichert werden, welche Gutscheine für jeweils wie viel Euro in Anspruch genommen werden.

Aufgaben:

- Stellen Sie den beschriebenen Sachverhalt in einem erweiterten Entity Relationship-Modell grafisch dar. Verwenden Sie dabei Kardinalitätsangaben.
- Geben Sie für das Modell Relationen in der Boyce Codd-Normalform an.

Aufgabe 2 (35 Punkte)

Die folgenden Relationen speichern Klausurergebnisse von Studierenden.

```
Studierende = ({matrikelnr, name}, { })  
Vorlesung    = ({vnr, name}, { })  
Klausur      = ({matrikelnr, vnr, datum, note}, { })
```

Die folgenden Tabellen verdeutlichen deren Bedeutung mit Beispieldaten:

Studierende	
matrikelnr	name
123456	Felix Maier
123459	Gaby Müller
123477	Hans Kunz
...	...

Vorlesung	
vnr	name
101	Datenbanksysteme
102	Betriebssysteme
104	Softwaremodellierung
...	...

Klausur			
matrikelnr	vnr	datum	note
123456	101	24.06.2020	1,0
123456	102	20.02.2020	1,3
123477	101	24.06.2020	3,3
...

Erstellen Sie SQL-Statements gemäß SQL-Standard für folgende Datenbank-Anfragen. Verwenden sie zur Suche und zur Ausgabe immer die entsprechenden Bezeichnungen aus der Aufgabenstellung und nicht die nur intern verwendeten Attribute (hier `matrikelnr`, `vnr`). Verwenden Sie ggfs. die Funktionen `TO_DATE` bzw. `TO_CHAR` um auf Datumsangaben zuzugreifen. Versuchen Sie die Anfragen so effizient und verständlich wie möglich zu formulieren.

Sie können diese Aufgabe auf Papier oder mit Oracle19c lösen. Eine Beispieldatenbank für dieses Datenmodell ist unter dem Schema „eck“ gespeichert. Die Attribute `vnr` und `matrikelnr` haben den Datentyp `int`. Sie besitzen darauf Leserechte.

Beispiel: `SELECT * FROM eck.Klausur;`

- In welchen Jahren ist in Datenbanksysteme niemand durchgefallen (niemand hat Note 5,0 erzielt)?
- Welche Studierenden haben in Datenbanksysteme bereits eine 1,0 als auch eine 5,0 erzielt?
- In welcher Vorlesung wurden 2020 durchschnittlich die besten Noten erzielt? Falls mehrere Vorlesungen den besten Notenschnitt erzielt haben, sollen alle ausgegeben werden.
- Wie viele Studierende haben noch keine Klausur in Datenbanken geschrieben?
- Stellen Sie pro Kalenderjahr die Durchschnittsnoten über sämtliche Vorlesungen hinweg zusammen? Sortieren Sie nach Kalenderjahr absteigend.
- Wie oft kommt es vor, dass zwei oder mehrere Studierende den gleichen Namen haben? Es soll die Anzahl der betreffenden Namen ausgegeben werden.

Geben Sie die folgende Anfrage in der relationalen Algebra an:

- Welche Note hat Gaby Müller in Datenbanken am 24.06.2020 erzielt?

Aufgabe 3 (10 Punkte)

Gegeben sind die folgenden Tabellen und Beispieldaten aus der Vorlesung:

Pers = ({pnr, name, jahrg, eindat, gehalt, beruf, anr, vnr})

Abt = ({anr, aname, ort})

Pers							
<u>pnr</u>	name	jahrg	eindat	gehalt	beruf	<u>anr</u>	<u>vnr</u>
406	Coy	1950	01.03.86	80.000	Kaufmann	K55	123
123	Mueller	1958	01.09.80	68.000	Programmierer	K51	
829	Schmidt	1960	01.06.90	74.000	Kaufmann	K53	123
874	Abel		01.05.94	62.000	Softw.Entwickler	K55	829
503	Junghans	1975		55.000	Programmierer	K51	123

Abt		
<u>anr</u>	aname	ort
K51	Entwicklung	Erlangen
K53	Buchh	Nürnberg
K55	Personal	Nürnberg

- a) Schreiben Sie eine SQL-Query für die folgende Anfrage: Ermitteln Sie die Namen aller gemeinsamen Chefs von Hr. Abel und Hr. Junghans. Dabei sollen nicht nur die direkten Chef-Beziehungen, sondern auch indirekt die Chefs der Chefs, etc. berücksichtigt werden.
- b) Die folgende SQL-Query sollte alle Abteilungsnamen ausgeben, in denen Software Entwickler oder Programmierer arbeiten. Leider enthält der Befehl einen Fehler und so wie er definiert ist, werden fälschlicherweise **alle** Abteilungsnamen ausgegeben. Beschreiben den Fehler und korrigieren Sie die Anfrage.

```
SELECT distinct a.aname
FROM Pers p, Abt a
WHERE p.anr = a.anr
AND p.beruf = 'Softw.Entwickler'
OR p.beruf = 'Programmierer';
```

Aufgabe 4 (4 Punkte)

Gegeben seien die folgenden Tabellen L und R und deren Attribute a und b bzw. c und d. Geben Sie für die folgenden Join-Verknüpfungen die Ergebnistabellen an.

L	
a	b
7	20
5	6
4	6

R	
c	d
20	20
7	4
7	6

- a) L RIGHT OUTER JOIN R ON L.b = R.d
- b) L FULL OUTER JOIN R ON L.a = R.c

Aufgabe 5 (6 Punkte)

Gegeben ist die Tabelle aus Aufgabe 3 und die folgende Anfrage

```
SELECT name, gehalt/12, eindat  
FROM Pers  
WHERE beruf = 'Programmierer'  
ORDER BY gehalt;
```

- a) Auf welchem Attribut würden Sie einen Index definieren um die SQL-Anfrage zu beschleunigen?
- b) Geben Sie einen SQL-Befehl an, der durch den Index verlangsamt werden würde.

Aufgabe 6 (12 Punkte)

Ein Kino mit mehreren Kinosälen verwendet ein Datenbanksystem um ihr Kinoprogramm zu speichern. Der Programmierer der Datenbank, der leider noch nie eine Vorlesung in Datenbanken gehört hat, hat folgende Relation gebildet:

```
Vorstellung = ({vorführungsnummer, filmtitel, altersfreigabe,  
                datum, uhrzeit, saal, anzSitze})
```

Die Attribute der Relation haben folgende Bedeutung:

- `vorführungsnummer`: Eindeutige Nummer einer Vorführung
 - `filmtitel`: Name des gezeigten Films (Filmtitel sind eindeutig)
 - `altersfreigabe`: Altersfreigabe des Films
 - `datum`: Datum der Filmvorführung
 - `uhrzeit`: Uhrzeit der Filmvorführung
 - `saal`: Eindeutige Bezeichnung des Kinosaals
 - `anzSitze`: Anzahl Sitzplätze des Kinosaals
- a) In welcher Normalform ist die angegebene Relation? Geben Sie eine kurze Begründung an.
- a) Geben Sie die voll funktionalen Abhängigkeiten an. Überprüfen Sie dabei, ob man auf die Vorführungsnummer verzichten könnte.
- b) Überführen Sie die Relation ggfs. in die Boyce Codd-Normalform.

Aufgabe 7 (3 Punkte)

Sind die folgenden Aussagen korrekt oder falsch? Es reicht „korrekt“ oder „falsch“ als Antwort.

- a) Ein Index sollte nur auf Attributen definiert werden, die sich nicht zu häufig ändern.
- b) Das Data Dictionary kann dazu beitragen, SQL-Injections zu verhindern.
- c) Prepared Statements können dazu beitragen, SQL-Injections zu verhindern.