Klausur Datenbanken
Wintersemester 2019/2020
Prof. Dr. Wolfgang May
27. Februar 2020, 14-16:30 Uhr
Bearbeitungszeit: 120 Minuten

Vorname:

Nachname:

Matrikelnummer:

Bei der Klausur sind **keine Hilfsmittel** (Skripten, Taschenrechner, etc.) erlaubt. Mobiltelefone müssen ausgeschaltet sein. Papier wird gestellt. Benutzen Sie die **ausgeteilten**, zusammengehefteten **Blätter** für Ihre Antworten. Schreiben Sie mit blauem/schwarzem Kugelschreiber, Füller, etc.; Bleistift ist nicht erlaubt.
Zwecks besserer Lesbarkeit (insbesondere auch für Nicht-{Mut/d/Va}tersprachler*innen) wird in der Aufgabenstellung auf gegenderte Sprache verzichtet.

Zum **Bestehen** der Klausur sind **45** Punkte hinreichend.

(Unterschrift (DSGVO))

Meine Note soll mit meiner persönlichen Codezahl: 1234 so bald wie möglich auf der Vorlesungs-Webseite veröffentlicht werden.

☐ Meine Note soll nicht veröffentlicht werden; ich erfahre sie dann aus FlexNow oder beim zuständigen Prüfungsamt.

	Max. Punkte	Erreichte Punkte
Aufgabe 1 (ER-Modell)	14	
Aufgabe 2 (Transformation in das Relationale Modell)	18	
Aufgabe 3 (SQL und Relationale Algebra)	45	
Aufgabe 4 (Verschiedenes)	14	
Summe	91	

Note:

Themenstellung: Essenslieferdienst

Alle Klausuraufgaben basieren auf einem gemeinsamen "Auftrag": In der Klausur soll eine Datenbank für einen Lieferdienst für Speisen entworfen werden.

1. Der Lieferdienst umfasst mehrere *Anbieter*.

   Jeder Anbieter hat einen *Namen*, z.B., *Pizzawelt* oder *Kalle's Imbiss* und eine *Adresse*. Jede Adresse besteht aus *Straßenname+Hausnummer* und gehört zu einem Postleitzahl-*Gebiet*. Straßenname + Hausnummer sind innerhalb eines solchen Gebiets eindeutig.

   Es kann mehrere Anbieter mit demselben Namen an verschiedenen Adressen geben. So gibt es eine *Pizzawelt*-Filiale sowohl an der Adresse *Um's Eck 5* im Gebiet *30425* als auch mit der Adresse *Laange Straße 404* im Gebiet *30431*. Neben dieser, im Haus *Laange Straße 406* befindet sich *Kalle's Imbiss*.

   Optional können Anbieter einen *Mindestbestellwert* für Bestellungen bei ihnen angeben.

   Bei *Pizzawelt Um's Eck 5* in 30425 beträgt der Mindestbestellwert 10 EUR.

2. Jeder Anbieter bietet eine Reihe von *Gerichten* an.

   Jedes Gericht hat einen *Namen*, z.B. *Pizza Margarita*, *Pizza Salami* oder *Pommes Frites*, und gehört zu einer Kategorie (z.B. *Pizza* oder *Snack*).

   Jeder Anbieter bietet einige Gerichte zu einem bestimmten *Preis* an: Bei *Pizzawelt Um's Eck 5* im Gebiet 30425 gibt es *Pizza Margarita* für 4.00 Euro und *Pizza Salami* für 4.50 EUR, während die andere *Pizzawelt* an der *Laangen Straße 404* die *Pizza Margarita* für nur 3.80 EUR anbietet. *Pommes Frites* kosten bei *Kalle's Imbiss* 3 EUR.

   Die Preise und die angebotenen Gerichte eines Anbieters können sich zudem im Laufe der Zeit ändern.

3. Für den Lieferdienst arbeiten mehrere *Fahrer*. Sie werden über ihren eindeutigen internen *Nickname* identifiziert, außerdem ist ihre Handy-Nummer gespeichert. Jeder Fahrer fährt in einem oder mehreren Postleitzahl-*Gebieten*. Für jeden Fahrer ist gespeichert, ob er gerade Dienst hat, oder nicht (ja/nein).

   Der Fahrer *Timo*, Handynummer 0160-4711, fährt in den Gebieten 30423, 30424 und 30425 und hat gerade Dienst.

   Der Fahrer *Tilo*, Handynummer 0171-0815, fährt in den Gebieten 30425, 30426 und 30431 und hat gerade frei.

4. Im Mittelpunkt der Datenbank stehen natürlich die *Bestellungen*, die von Kunden (per Web-Frontend, aber das spielt hier keine Rolle) aufgegeben werden.

   Jeder Bestellung wird eine eindeutige *ID* zugewiesen und das aktuelle *Datum* und die *Uhrzeit* werden erfasst. Der Kunde gibt eine Liefer*adresse* (Straße+Hausnr., Postleitzahl, sowie den Namen, bei dem geklingelt werden soll) an, und wählt dann einen *Anbieter* aus.

   Jede Bestellung besteht aus einer Liste der bestellten *Gerichte* des ausgewählten Anbieters mit der bestellten *Anzahl* zu deren derzeitigem *Kaufpreis*.

Später wird der Bestellung dann ein *Fahrer* zugeordnet, der sowohl den entsprechenden Anbieter als auch die Lieferadresse des Kundens erreicht. Dabei wird der *Status* der Bestellung (Neu, Zugeordnet (zu einem Fahrer), Abgeholt, Ausgeliefert) verfolgt. Es wird auch gespeichert, wann die Bestellung ausgeliefert wurde.

Alice gab soeben (27.02.2020, 14:05 Uhr) eine *Bestellung* auf, die die ID 777 bekommen hat. Sie bestellte von *Pizzawelt* an der Adresse *Um's Eck 5* im Gebiet 30425:

- 1 x *Pizza Margarita* für 4.00 Euro
- 2 x *Pizza Salami* für jeweils 4.50 Euro

Als Lieferadresse gab sie *Schmidt*, *Hinter dem Mond 15* im Gebiet 30423 an. Bisher wurde die Bestellung noch keinem Fahrer zugeteilt.

Name: MatNr.:

Aufgabe 1 (ER-Modell [14 Punkte])
Entwickeln Sie ein ER-Modell für das Szenario. Geben Sie darin die Schlüsselattribute sowie die Beziehungskardinalitäten an.

Name: MatNr.:

Aufgabe 2 (Transformation in das Relationale Modell [18 Punkte])

a) Geben Sie an, welche Tabellen (mit Attributen, Schlüsseln etc.) Ihre Datenbank enthält (keine SQL `CREATE TABLE`-Statements, sondern einfach grafisch). (12 P)

Markieren Sie dabei auch Schlüssel (durch unterstreichen) und Fremdschlüssel (durch überstreichen).

Geben Sie die Tabellen mit jeweils mindestens zwei Beispieltupeln (z.B. denen, die sich aus dem Aufgabentext ergeben, und/oder weiteren erfundenen) an.

b) Geben Sie das `CREATE TABLE`-Statement für diejenige Tabelle, in der das Angebot der Anbieter gespeichert wird, so vollständig wie möglich (d.h. mit allen notwendigen Constraints) an (6 P).

Name: MatNr.:

Aufgabe 3 (SQL und Relationale Algebra [45 Punkte])
Verwenden Sie für diese Aufgabe die von Ihnen entworfene relationale Datenbasis. Keine der Antworten soll Duplikate enthalten.

a) Geben Sie eine SQL-Anfrage *und* einen Algebra-Ausdruck oder -Baum an, die die Namen aller Pizzas angeben, die zum aktuellen Zeitpunkt bei mindestens einem Anbieter für weniger als 5 EUR zu bekommen sind. (2+2 P)

b) Geben Sie eine SQL-Anfrage *und* einen Algebra-Ausdruck oder -Baum an, die für die Lieferadresse *Hinter dem Mond 15, 30423* alle Anbieter ausgeben bei denen bestellt werden kann (d.h., für die es einen Fahrer gibt, der die Lieferadresse und den Anbieter erreichen kann, und gerade Dienst hat). (3+3 P)

c) Geben Sie eine SQL-Anfrage an, die für jeden Anbieter berechnet, wieviel Umsatz er im 2. Halbjahr 2019 über den Lieferservice gemacht hat. (3 P)

d) Geben Sie eine SQL-Anfrage *und* einen Algebra-Ausdruck oder -Baum an, die für jeden Anbieter alle von ihm aktuell angebotenen Gerichte auflisten, die bei ihm noch *nie* bestellt wurden. (3+3 P)

e) Geben Sie eine SQL-Anfrage an, die für jede Lieferadresse deren "Lieblingsanbieter" ausgibt, d.h., denjenigen Anbieter (bzw. diejenigen Anbieter, falls mehrere gleich beliebt sind), bei dem/denen (zu dieser Lieferadresse) am *häufigsten* bestellt wurde. (5 P)

f) Alice macht gleich nach der in der Themenstellung beschriebenen Bestellung noch weitere Bestellungen bei anderen Anbietern *an dieselbe Lieferadresse (und denselben Empfängername)* (z.B. bestellt sie noch 2 Portionen *Pommes Frites* bei *Kalle's Imbiss*). Aus diesem Grund wartet das System nach Eingang einer Bestellung immer einige Minuten, ob noch weitere Bestellungen mit demselben Ziel eingehen.

   Wenn dann eine solche –noch offene– Bestellung einem Fahrer zugeordnet werden soll, wird im optimalen Fall ein einziger Fahrer für alle *noch nicht bearbeiteten* Bestellungen, die an dieselbe Lieferadresse ausgeliefert werden sollen, zugeordnet.

   Geben Sie eine SQL-Anfrage *und* einen Algebra-Ausdruck oder -Baum an, die Namen derjenigen zur Zeit diensthabenden Fahrer ausgibt, die *jede* der neuen, noch nicht zugeordneten Bestellungen an *Schmidt*, *Hinter dem Mond 15, 30423*, abholen und ausliefern können. (6+6 P)

g) Ein bisschen Theorie. (10 P)

   Seien $R(\bar{A}, B)$ eine Relation, und Q eine Anfrage mit Format $[B]$ mit nicht-leerem Ergebnis.

   Betrachten Sie die Ausdrücke

   (1) $\pi[\bar{A}](R \ltimes Q)$ (Hinweis: $\ltimes$ ist ein linkes Semijoin[1])

[1] von rechten Semijoins distanzieren wir uns hiermit und lehnen jede Zusammenarbeit mit ihnen ab.

(2) $R \div Q$

Hinweis: Q könnte z.B. die Anfrage aus (3a) sein (die die Namen aller einfachen Pizzen ergibt), und R eine geeignete Projektion der Relation, in der gespeichert ist, welche Anbieter welche Gerichte anbieten.

- Zeigen sie, dass (1) = (2) nicht immer gilt. (2 P)
- Ist wenigstens eine der beiden folgenden Aussagen allgemeingültig? (4 P)

ja	nein	
☐	☐	(1) $\subseteq$ (2)
☐	☐	(1) $\supseteq$ (2)

  (mit Beweis oder fundierter Begründung)
- Was bedeutet es, für den Zusammenhang der Relation R mit einer festen Ergebnisrelation B_0 von Q (die o.E.d.A. mehr als ein Element enthält), wenn die Gleichheit (1) = (2) im aktuellen Datenbankzustand gilt. (3 P)

Name: MatNr.:

Aufgabe 4 (Verschiedenes [14 Punkte])

a) Fahrer Timo bekommt einen Studienplatz und kündigt.

a1) Welche Daten von ihm müssen in der Datenbank erhalten bleiben, und warum? (2P)

a2) Welche Daten über ihn werden sinnvollerweise gelöscht?
Geben Sie an, mit welchem SQL-Statement(s) dies auf dem Datenbankserver getan werden kann. (3P)

a3) Die diese Statement(s) ausführende Person verschreibt sich beim Tippen des SQL-Statements und hat statt "Timo" "Tilo" geschrieben, den es auch gibt. Er merkt es sofort, als die Datenbank ausgibt

```
sql> ... delete or update-statement ...
successfully deleted/updated ... rows
sql>
```

Was ist jetzt das Problem? (1P, kurz)

– Wie kann man am besten die Auswirkungen beheben? (2P)

b) Bei der Auswertung der Anfrage in Aufgabe 3f (der letzten Anfrage) müssen alle Bestellungen, auch die bereits schon lange erledigten, geprüft werden, ob sie die Bedingungen erfüllen.

Das sollte man effizienter lösen.

b1) Warum ist die Antwort "das sind doch sowieso die neuesten, die sind ganz schnell zu finden, meistens sogar noch im Cache" falsch? (2P)

b2) Beschreiben Sie eine Möglichkeit (es gibt einige, ganz verschiedene), wie man das System in diesem Aspekt effizienter gestalten kann? (3P)