Unterlagen für die Lehrkraft

Abiturprüfung 2018

Informatik, Grundkurs

1. Aufgabenart

Analyse, Modellierung und Abfrage relationaler Datenbanken

2. Aufgabenstellung[1]

siehe Prüfungsaufgabe

3. Materialgrundlage

entfällt

4. Bezüge zum Kernlehrplan und zu den Vorgaben 2018

Die Aufgaben weisen vielfältige Bezüge zu den Kompetenzerwartungen und Inhaltsfeldern des Kernlehrplans bzw. zu den in den Vorgaben ausgewiesenen Fokussierungen auf.
Im Folgenden wird auf Bezüge von zentraler Bedeutung hingewiesen.

1. *Inhaltsfelder und inhaltliche Schwerpunkte*
   Daten und ihre Strukturierung
   - Datenbanken

   Formale Sprachen und Automaten
   - Syntax und Semantik einer Programmiersprache
     - SQL

2. *Medien/Materialien*
   - entfällt

5. Zugelassene Hilfsmittel

- Taschenrechner (graphikfähiger Taschenrechner / CAS-Taschenrechner)
- Wörterbuch zur deutschen Rechtschreibung

[1] Die Aufgabenstellung deckt inhaltlich alle drei Anforderungsbereiche ab.

6. Modelllösungen

Die jeweilige Modelllösung stellt eine mögliche Lösung bzw. Lösungsskizze dar. Der gewählte Lösungsansatz und -weg der Schülerinnen und Schüler muss nicht identisch mit dem der Modelllösung sein. Sachlich richtige Alternativen werden mit entsprechender Punktzahl bewertet (Bewertungsbogen: Zeile „Sachlich richtige Lösungsalternative zur Modelllösung“).

Teilaufgabe a)

Die in Abbildung 1 gegebene Teilmodellierung besteht aus vier Entitätstypen und drei Beziehungstypen.

In den Entitätstypen `Restaurant`, `Gericht` und `Zutat` wird der Primärschlüssel in Form einer ID modelliert (Attribute: `RestaurantID`, `GerichtID` und `ZutatID`). Beim Entitätstyp `Richtung` wird die Bezeichnung verwendet (Attribut: `Bezeichnung`). Im Entitätstyp `Restaurant` kommt noch ein Name (Attribut: `Name`) und eine Adresse (Attribut: `Adresse`) dazu; bei Gerichten, Zutaten und kulinarischen Richtungen eine Bezeichnung (Attribute: `Bezeichnung`). Für jedes Gericht wird zusätzlich gespeichert, welchen Brennwert es hat (Attribut: `Brennwert`) und für jede Richtung, ob sie gerade im Angebot ist (Attribut: `istImAngebot`).

Der Beziehungstyp `bietetAn` modelliert, welches Restaurant welche Gerichte anbietet und zu welchem Preis das geschieht (Attribut: `Preis`). Da es sich um einen `n:m`-Beziehungstyp handelt, kann ein Restaurant mehrere Gerichte anbieten und auch ein Gericht von mehreren Restaurants bezogen werden.

Der Beziehungstyp `verwendet` modelliert, welche Zutaten für ein Gericht verarbeitet werden und in welcher Menge (Attribut: `Menge`). Da es sich auch hier um einen `n:m`-Beziehungstyp handelt, kann ein Gericht aus mehreren Zutaten bestehen und eine Zutat auch in mehreren Gerichten vorkommen.

Der Beziehungstyp `gehoertZu` modelliert, welches Gericht zu welcher kulinarischen Richtung gehört. Da es sich um einen `1:n`-Beziehungstyp handelt, muss ein Gericht genau einer Richtung zugeordnet werden, wobei eine Richtung mehrere Gerichte umfassen kann.

Das Attribut `Menge` muss dem Beziehungstyp `verwendet` zugeordnet sein, da die Menge, in der eine Zutat verwendet wird, von Gericht zu Gericht verschieden sein kann. Wäre das Attribut dem Entitätstyp `Zutat` zugeordnet, könnte z. B. die Zutat Salz nur noch in der dort pauschal angegebenen Menge verwendet werden, da die Datenbank die Mengen nicht mehr zwischen verschiedenen Gerichten differenzieren könnte.

Teilaufgabe b)

Die folgenden SQL-Anweisungen realisieren die Anfragen:

```
(i)
SELECT Gericht.Bezeichnung
FROM Gericht
WHERE Gericht.Brennwert < 200

(ii)
SELECT Gericht.Bezeichnung, bietetAn.Preis
FROM Restaurant
  INNER JOIN bietetAn
    ON Restaurant.RestaurantID = bietetAn.RestaurantID
  INNER JOIN Gericht
    ON bietetAn.GerichtID = Gericht.GerichtID
WHERE Restaurant.Name = "Lukullischer Luxus" AND
      bietetAn.Preis < 20

(iii)
SELECT Gericht.GerichtID, Gericht.Bezeichnung,
       COUNT(Bestellung.GerichtID)
FROM Gericht
  INNER JOIN Bestellung
    ON Gericht.GerichtID = Bestellung.GerichtID
GROUP BY Gericht.GerichtID
```

Teilaufgabe c)

(i) Die erste Abfrage erstellt das Kreuzprodukt der Relationen `Bestellung` und `bietetAn`. Aus diesem Kreuzprodukt werden diejenigen Datensätze selektiert, bei denen die Werte für `GerichtID` und `BestellungID`, die aus den Relationen `Bestellung` und `bietetAn` stammen, übereinstimmen.
In die Ergebnisrelation wird die ID des Restaurants (`RestaurantID`) und ein Wert mit dem Alias `A` aufgenommen, der für jedes Restaurant als Summe aller Produkte aus Angebotspreis (`bietetAn.Preis`) und Bestellmenge (`Bestellung.Anzahl`) berechnet wird. Dieser Wert stellt die Summe der Einnahmen eines Restaurants dar.
Da eine Aggregatfunktion verwendet wird, müssen die Datensätze nach `RestaurantID` gruppiert werden.
Im Sachzusammenhang ermittelt die Anweisung für jedes Restaurant, wie viel Geld es mit der Bestellplattform eingenommen hat. Dabei werden die ID jedes Restaurants und die Summe seiner Einnahmen zurückgegeben.

(ii) Die zweite Abfrage ermittelt mit einer Unterabfrage in den Zeilen 4 und 5 alle IDs von Gerichten aus der Relation `Bestellung`, d. h. die IDs derjenigen Gerichte, die schon einmal bestellt wurden. Anschließend ermittelt die äußere SQL-Anweisung die Bezeichnungen aller Gerichte aus der Relation `Gericht`, deren IDs nicht in der von der Unterabfrage ermittelten Menge sind.
Im Sachzusammenhang ermittelte die Anweisung, welche Gerichte noch nie bestellt wurden.

Teilaufgabe d)

Das folgende Datenbankschema trägt den Kritikpunkten Rechnung:

Kunde(KundeID, Vorname, Nachname, ↑AdresseID)

Restaurant(RestaurantID, Name, ↑AdresseID)

bietetAn(↑RestaurantID, ↑GerichtID, Preis)

Gericht(GerichtID, Bezeichnung, Brennwert)

Bestellung (↑KundeID, ↑GerichtID, ↑RestaurantID, Datum, Anzahl, istAbgearbeitet)

Adresse(AdresseID, Strassenname, Hausnummer, PLZ, Ortsname)

Um speichern zu können, ob eine Bestellung bereits abgearbeitet ist oder nicht, wird das boolesche Attribut `istAbgearbeitet` im Relationenschema `Bestellung` eingefügt.
Um die dritte Normalform herzustellen, muss jedes Attribut des Datenbankschemas atomar sein. Das ist bei der Adresse nicht der Fall, so dass diese in Einzelattribute aufgeteilt werden muss. Das Relationenschema `Adresse` fasst diese Attribute zusammen und deren Primärschlüssel `AdresseID` wird statt der Adresse selbst in den Relationenschemata `Restaurant` und `Kunde` eingetragen.
Des Weiteren müssen sich bei zusammengesetzten Primärschlüsseln alle Nicht-Schlüsselattribute desselben Relationenschemas auf alle Teile des Schlüssels beziehen. Dies ist bei den gegebenen Relationen bereits der Fall.
Letztlich darf es keine funktionale Abhängigkeit zwischen Nicht-Schlüsselattributen desselben Relationenschemas geben. Das ist beim gegebenen Datenbankschema auch nirgends der Fall.

Hinweise:
Je nach Sachkenntnis des Prüflings könnte fälschlich eine funktionale Abhängigkeit zwischen `PLZ` und `Ortsname` angenommen werden. Eine Lösung, die auf diesem Gedanken basiert, ist ebenfalls als richtig zu werten.

Teilaufgabe e)

Ob ein Gericht vegetarisch ist, hängt allein von seiner Zutatenliste ab und nicht vom Restaurant, das es anbietet. Das Attribut `istVegetarisch` ist also nicht in die Relation `bietetAn` einzufügen. Es würde sich daraus auch ein Verstoß gegen die zweite Normalform ergeben. Das Attribut `istVegetarisch` könnte in die Relation `Gericht` eingefügt werden, allerdings könnte ein Gericht dann als vegetarisch markiert werden, obwohl es eine offensichtlich fleischhaltige Zutat hat. Auch könnten widersprüchliche Eintragungen vorgenommen werden, wenn z. B. zwei Gerichte mit gleicher Zutatenliste einmal als vegetarisch und einmal als nicht vegetarisch markiert werden.
Zu bevorzugen ist daher, alle fleischhaltigen bzw. nicht fleischhaltigen Zutaten mit dem Attribut `istVegetarisch` zu kennzeichnen und die Frage nach einem vegetarischen Gericht mit einer entsprechenden Abfrage zu beantworten.

7. Teilleistungen – Kriterien / Bewertungsbogen zur Prüfungsarbeit

Name des Prüflings:_____ Kursbezeichnung:_____

Schule: _____

Teilaufgabe a)

	Anforderungen	**Lösungsqualität**			
	Der Prüfling	**maximal erreichbare Punktzahl**	**EK[2]**	**ZK**	**DK**
1	erläutert die gegebene Teilmodellierung und geht auf die Beziehungstypen ein.	7			
2	begründet, warum das Attribut `Menge` dem Beziehungstyp `verwendet` zugeordnet ist und nicht dem Entitätstyp `Zutat`.	3			
Sachlich richtige Lösungsalternative zur Modelllösung: (10) …… ……					
	Summe Teilaufgabe a)	**10**			

Teilaufgabe b)

	Anforderungen	**Lösungsqualität**			
	Der Prüfling	**maximal erreichbare Punktzahl**	**EK**	**ZK**	**DK**
1	entwirft für die erste Anfrage eine SQL-Anweisung.	3			
2	entwirft für die zweite Anfrage eine SQL-Anweisung.	4			
3	entwirft für die dritte Anfrage eine SQL-Anweisung.	5			
Sachlich richtige Lösungsalternative zur Modelllösung: (12) …… ……					
	Summe Teilaufgabe b)	**12**			

[2] EK = Erstkorrektur; ZK = Zweitkorrektur; DK = Drittkorrektur

Teilaufgabe c)

	Anforderungen	**Lösungsqualität**			
	Der Prüfling	**maximal erreichbare Punktzahl**	**EK**	**ZK**	**DK**
1	analysiert und erläutert die erste SQL-Anweisung.	4			
2	erläutert im Sachzusammenhang, welche Information die erste SQL-Anweisung ermittelt.	2			
3	analysiert und erläutert die zweite SQL-Anweisung.	4			
4	erläutert im Sachzusammenhang, welche Information die zweite SQL-Anweisung ermittelt.	2			
Sachlich richtige Lösungsalternative zur Modelllösung: (12) …… ……					
	Summe Teilaufgabe c)	**12**			

Teilaufgabe d)

	Anforderungen	**Lösungsqualität**			
	Der Prüfling	**maximal erreichbare Punktzahl**	**EK**	**ZK**	**DK**
1	überführt die gegebenen Relationenschemata in ein Datenbankschema das den Kritikpunkten Rechnung trägt.	6			
2	erläutert, wie sein Datenbankschema die aufgeführten Probleme löst.	4			
Sachlich richtige Lösungsalternative zur Modelllösung: (10) …… ……					
	Summe Teilaufgabe d)	**10**			

Teilaufgabe e)

	Anforderungen	**Lösungsqualität**			
	Der Prüfling	**maximal erreichbare Punktzahl**	**EK**	**ZK**	**DK**
1	entscheidet und begründet, in welche der Relationen das Attribut aufgenommen werden sollte.	6			
Sachlich richtige Lösungsalternative zur Modelllösung: (6) …… ……					
	Summe Teilaufgabe e)	**6**			

	Summe insgesamt	**50**			

Festlegung der Gesamtnote (Bitte nur bei der letzten bearbeiteten Aufgabe ausfüllen.)

	Lösungsqualität			
	maximal erreichbare Punktzahl	EK	ZK	DK
Übertrag der Punktsumme aus der ersten bearbeiteten Aufgabe	**50**			
Übertrag der Punktsumme aus der zweiten bearbeiteten Aufgabe	**50**			
Punktzahl der gesamten Prüfungsleistung	**100**			
aus der Punktsumme resultierende Note gemäß nachfolgender Tabelle				
Note ggf. unter Absenkung um bis zu zwei Notenpunkte gemäß § 13 Abs. 2 APO-GOSt				
Paraphe				

Berechnung der Endnote nach Anlage 4 der Abiturverfügung auf der Grundlage von § 34 APO-GOSt

Die Klausur wird abschließend mit der Note _____ (____ Punkte) bewertet.

Unterschrift, Datum:

Grundsätze für die Bewertung (Notenfindung)

Für die Zuordnung der Notenstufen zu den Punktzahlen ist folgende Tabelle zu verwenden:

Note	**Punkte**	**Erreichte Punktzahl**
sehr gut plus	15	100 – 95
sehr gut	14	94 – 90
sehr gut minus	13	89 – 85
gut plus	12	84 – 80
gut	11	79 – 75
gut minus	10	74 – 70
befriedigend plus	9	69 – 65
befriedigend	8	64 – 60
befriedigend minus	7	59 – 55
ausreichend plus	6	54 – 50
ausreichend	5	49 – 45
ausreichend minus	4	44 – 40
mangelhaft plus	3	39 – 33
mangelhaft	2	32 – 27
mangelhaft minus	1	26 – 20
ungenügend	0	19 – 0