

*Unterlagen für die Lehrkraft***Abiturprüfung 2016***Informatik, Grundkurs***1. Aufgabenart**

Aufgabenart	Aufgabenstellungen aus dem Bereich Relationale Datenbanken
Syntaxvariante	–

2. Aufgabenstellung¹

siehe Prüfungsaufgabe

3. Materialgrundlage

- entfällt

4. Bezüge zu den Vorgaben 2016**1. Inhaltliche Schwerpunkte**

Relationale Datenbanken

- Modellieren kontextbezogener Problemstellungen als Datenbanken mit dem Entity-Relationship Modell
- Datenbankschemata
- Normalisierung: Überführung einer Datenbank in die 1. bis 3. Normalform
- Relationenalgebra (Selektion, Projektion, Umbenennung, Join)
- SQL-Abfragen über eine und mehrere verknüpfte Tabellen

2. Medien/Materialien

- entfällt

5. Zugelassene Hilfsmittel

- Wörterbuch zur deutschen Rechtschreibung
- Taschenrechner (wissenschaftlicher Taschenrechner ohne oder mit Grafikfähigkeit/CAS-Taschenrechner)

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)

Der Datenbankentwurf aus Abbildung 1 umfasst die vier Entitätstypen Nachhilfeschueler, Tutor, Fach und Raum.

Für jeden Nachhilfeschüler und jeden Tutor werden ein Vorname, ein Nachname und ein Geburtsjahr gespeichert. Darüber hinaus sehen beide Entitätstypen ein ID-Attribut als Primärschlüssel vor.

Ein Fach ist mit einem ID-Eintrag als Primärschlüssel und einer Bezeichnung modelliert und ein Raum mit einer Raumnummer als Primärschlüssel und einer Bezeichnung.

Zwischen dem Entitätstyp Nachhilfeschueler und dem Entitätstyp Fach wird durch die Angabe des Beziehungstyps `brauchtHilfeIn` eine n:m-Beziehung modelliert, was bedeutet, dass ein Nachhilfeschüler in mehreren Fächern Hilfe benötigen kann und ein Fach auch bei mehreren Nachhilfeschülern als ein Problem eingetragen werden kann.

Der Beziehungstyp `trifftSichMit` modelliert zwischen dem Entitätstyp Nachhilfeschueler und dem Entitätstyp Tutor eine n:m-Beziehung, was bedeutet, dass ein Nachhilfeschüler von mehreren Tutoren Unterstützung in Anspruch nehmen kann und ein Tutor sich ggf. mehrerer Nachhilfeschüler annimmt. Des Weiteren ist eingetragen, an welchem Termin ein Treffen zwischen Nachhilfeschüler und Tutor stattfindet.

Der Beziehungstyp `istZugewiesen` zwischen den Entitätstypen Raum und Tutor stellt die Modellierung einer 1:n-Beziehung dar, woraus zu ersehen ist, dass einem Tutor genau ein Raum zukommt, diesem Raum aber noch andere Tutoren zugewiesen sein können.

Das zum Entity-Relationship-Diagramm in Abbildung 1 gehörende Datenbankschema ist das folgende:

```
Nachhilfeschueler(NachhilfeschuelerID, Vorname, Nachname,  
Geburtsjahr)  
Tutor(TutorID, Vorname, Nachname, Geburtsjahr, ↑Raumnummer)  
Fach(FachID, Bezeichnung)  
Raum(Raumnummer, Bezeichnung)  
brauchtHilfeIn(↑NachhilfeschuelerID, ↑FachID)  
trifftSichMit(↑NachhilfeschuelerID, ↑TutorID, Termin)
```

Die Beziehungstypen `brauchtHilfeIn` und `trifftSichMit` modellieren n:m-Beziehungen und werden daher jeweils mit einem separaten Relationenschema realisiert. Dabei werden die Primärschlüssel der beiden beteiligten Relationen als Fremdschlüssel eingetragen und ergeben den kombinierten Primärschlüssel der Beziehungsrelation. Beim Relationenschema `trifftSichMit` wird noch der Termin ergänzt. Da es sinnvoll ist, mehr als ein Treffen zwischen den gleichen Personen eintragen zu können, ist das Attribut `Termin` ebenfalls Teil des Primärschlüssels.

Der Beziehungstyp `istZugewiesen` modelliert eine 1:n-Beziehung. Die Primärschlüsselinträge der Relation zum Relationenschema `Raum` werden als Fremdschlüssel in die Relation zum Relationenschema `Tutor` eingetragen.

Teilaufgabe b)

Das Relationenschema `Sprechstunde` entspricht nicht den Kriterien der dritten Normalform, da das Attribut `Name` keinen atomaren Wertebereich hat, was der ersten und somit auch der dritten Normalform widerspricht. Darüber hinaus ist das Attribut `Anfangszeit` abhängig vom Attribut `Stunde`. Beide Attribute gehören nicht zum Primärschlüssel des Relationenschemas `Sprechstunde`. Dies ist ein weiterer Verstoß gegen die dritte Normalform.

Folgendes Datenbankschema entspricht der dritten Normalform:

Sprechstunde(Kuerzel, Vorname, Nachname, Tag, ↑Stunde, Raum)
Stundentafel(Stunde, Anfangszeit)

Das Attribut `Name` wird in `Vorname` und `Nachname` aufgeteilt, so dass die Wertebereiche atomar sind. Da das Datenbankschema über keinen kombinierten Primärschlüssel verfügt, ist es automatisch in zweiter Normalform.

Darüber hinaus wird ein zusätzliches Relationenschema `Stundentafel` eingeführt. `Anfangszeit` ist nun vom Primärschlüssel `Stunde` dieses Relationenschemas abhängig. Im Relationenschema `Sprechstunde` wird eine 1:n-Beziehung zu diesem neuen Relationenschema aufgebaut, indem der Fremdschlüssel `Stunde` eingetragen wird. Eine Abhängigkeit zwischen Attributen, die keine Schlüssel sind, liegt somit nicht mehr vor.

Teilaufgabe c)

Die folgenden SQL-Anweisungen realisieren die gesuchten Anfragen:

I.

```
SELECT Tutor.Vorname, Tutor.Nachname, Tutor.Grundentlohnung
FROM Tutor
WHERE Tutor.Grundentlohnung > 10
ORDER BY Tutor.Grundentlohnung DESC
```

II.

```
SELECT Tutor.Nachname
FROM Tutor JOIN (
    SELECT hat.TutorID, Count(hat.TutorID) as Anzahl
    FROM hat
    GROUP BY hat.TutorID
) AS tmp ON Tutor.TutorID = tmp.TutorID
WHERE tmp.Anzahl >= 2
```

III.

```
SELECT DISTINCT Tutor.Nachname, Tutor.Vorname, Auszeichnung.Fach
FROM Tutor
    LEFT JOIN hat ON Tutor.TutorID = hat.TutorID
    LEFT JOIN Auszeichnung
        ON hat.AuszeichnungID = Auszeichnung.AuszeichnungID
ORDER BY Tutor.Nachname, Tutor.Vorname
```

Teilaufgabe d)

Die Unterabfrage in den Zeilen 3 bis 5 liefert eine Ergebnistabelle mit den ID-Einträgen der Auszeichnungen unter dem Alias a und den dazugehörigen Boni aus der Tabelle Rang unter dem Alias b. In der Tabelle ist also aufgeführt, welchen Bonus es für welche Auszeichnung gibt. Die Tabelle wird mit dem Alias tmp benannt.

Bei Anwendung auf die Beispieldatensätze in der Anlage liefert die Abfrage folgendes Ergebnis:

Ergebnis:	
a	b
1	5
2	10
3	15
4	5

Die äußere SQL-Anweisung liefert eine Ergebnistabelle mit Nachnamen und Vornamen von Tutoren. In der dritten Spalte wird mittels der Aggregatfunktion SUM aus der Hilfstabelle tmp die Summe aller Bonuszahlungen angegeben, die zum jeweiligen Tutor gehört. Um die Summe der Bonuszahlungen berechnen zu können, müssen zunächst die Tutoren mit den von ihnen erhaltenen Bonuszahlungen in Verbindung gebracht werden. Hierzu wird in der WHERE-Klausel (Z. 6) eine Verbundoperation (JOIN) auf den relevanten Schlüsseln der Tabellen Tutor, hat und tmp durchgeführt.

In Zeile 7 wird eine Gruppierung nach Vor- und Nachname der Tutoren vorgenommen, um die Aggregatfunktion auf die einzelnen Tutoren und nicht auf die Gesamtdatenmenge anwenden zu können.

Im Sachzusammenhang ermittelt die Abfrage die Information, welchen Betrag an Bonuszahlung jeder Tutor ausgezahlt bekommt. Tutoren, die keine Bonuszahlung bekommen, werden in der Anfrage nicht berücksichtigt.

Teilaufgabe e)

Die Variante mit zwei Relationen ist im gegebenen Beispielkontext hinsichtlich einer sinnvollen Datenbankmodellierung nicht vertretbar.

Die Modellierung der Boni mit einem eigenen Relationenschema wäre nur dann zu begründen, wenn ein Bonus eine eigene Entität wäre, der selbst wieder Attribute zukommen. Das ist im gegebenen Beispiel aber nicht der Fall.

Hinweis: Abweichende Bearbeitungen sind möglich.

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 den Datenbankentwurf und berücksichtigt dabei die Kardinalitäten der Beziehungen.	4			
2	entwickelt ein Datenbankschema zum Datenbankentwurf.	4			
3	erläutert, wie die Beziehungen im Datenbankschema umgesetzt sind.	2			
Sachlich richtige Lösungsalternative zur Modelllösung: (10)					
	Summe Teilaufgabe a)	10			

Teilaufgabe b)

	Anforderungen	Lösungsqualität			
	Der Prüfling	maximal erreichbare Punktzahl	EK	ZK	DK
1	begründet, dass das Relationenschema nicht den Kriterien der dritten Normalform entspricht, da nicht alle Attribute einen atomaren Wertebereich haben.	2			
2	begründet, dass das Relationenschema nicht den Kriterien der dritten Normalform entspricht, da eine funktionale Abhängigkeit zwischen zwei Nichtschlüsselattributen besteht.	2			
3	entwickelt zum Relationenschema ein Datenbankschema, das sich in dritter Normalform befindet.	4			
4	erläutert die Änderungen, die zur Überführung in die dritte Normalform nötig sind.	4			
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	entwickelt die SQL-Anweisung zur ersten Anfrage.	3			
2	entwickelt die SQL-Anweisung zur zweiten Anfrage.	5			
3	entwickelt die SQL-Anweisung zur dritten Anfrage.	5			
Sachlich richtige Lösungsalternative zur Modelllösung: (13)					
.....					
.....					
	Summe Teilaufgabe c)	13			

Teilaufgabe d)

	Anforderungen	Lösungsqualität			
	Der Prüfling	maximal erreichbare Punktzahl	EK	ZK	DK
1	erläutert die Unterabfrage.	3			
2	ermittelt das Ergebnis bei Anwendung auf die Beispieldatensätze.	2			
3	erläutert die gesamte SQL-Anweisung.	3			
4	erläutert, welche Information durch die SQL-Anweisung im Sachzusammenhang ermittelt wird.	2			
Sachlich richtige Lösungsalternative zur Modelllösung: (10)					
.....					
.....					
	Summe Teilaufgabe d)	10			

Teilaufgabe e)

	Anforderungen	Lösungsqualität			
	Der Prüfling	maximal erreichbare Punktzahl	EK	ZK	DK
1	beurteilt, ob der Vorschlag eine sinnvolle Datenbankmodellierung darstellt.	5			
Sachlich richtige Lösungsalternative zur Modelllösung: (5)					
.....					
.....					
	Summe Teilaufgabe e)	5			

	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 – 34
mangelhaft	2	33 – 27
mangelhaft minus	1	26 – 20
ungenügend	0	19 – 0