

Name: _____

Abiturprüfung 2016

Informatik, Leistungskurs

Aufgabenstellung:

Um Schülerinnen und Schülern der Sekundarstufe I bei Problemen im Unterricht zu helfen, organisiert am Ada-Lovelace-Gymnasium in Köln der Förderverein der Schule ein Nachhilfeprogramm. Vertrauenswürdige Schülerinnen und Schüler der Sekundarstufe II, sogenannte Tutoren, bieten Unterstützung an und bekommen dafür neben einem kleinen Entgelt auch Auszeichnungen, sofern sie sich in einem Fach, das von ihnen angeboten wird, besonderes engagiert haben.



Die folgende Abbildung zeigt einen Auszug aus dem für die Verwaltung der anfallenden Daten vorgesehenen Datenmodell. Auszeichnungen werden hier noch nicht berücksichtigt.

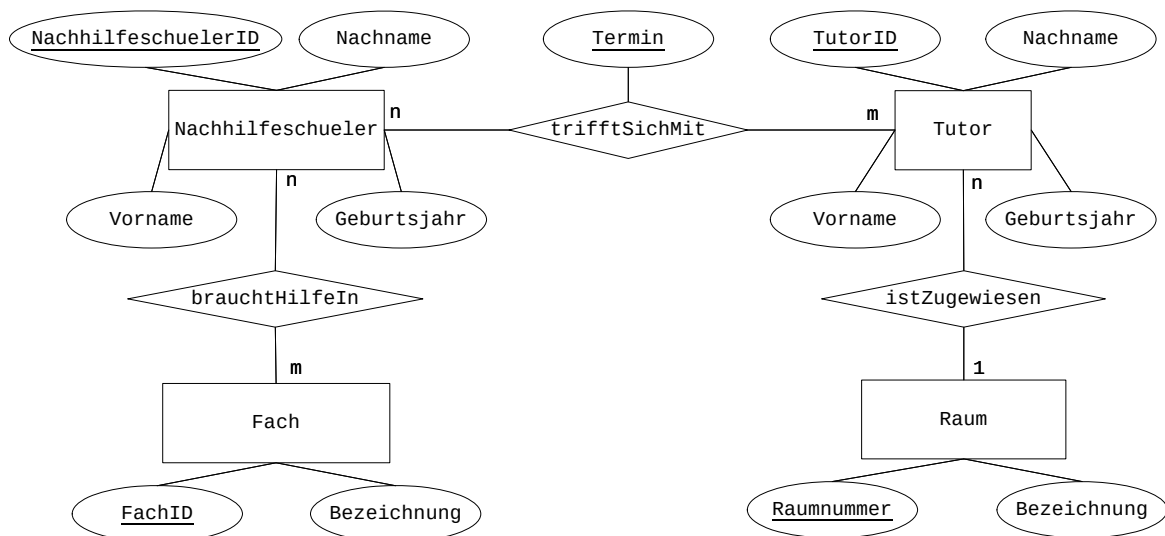


Abbildung 1: Auszug aus dem Datenmodell zum Nachhilfeprogramm



Name: _____

- a) *Erläutern Sie den in Abbildung 1 angegebenen Datenbankentwurf und berücksichtigen Sie dabei auch die Kardinalitäten der Beziehungen.*

Entwickeln Sie zum Datenbankentwurf aus Abbildung 1 ein Datenbankschema und erläutern Sie, wie die Beziehungen in Ihrem Schema umgesetzt sind.

(10 Punkte)

Da die Tutoren immer wieder Rückfragen an Lehrerinnen und Lehrer haben, müssen sie oft deren wöchentliche Sprechstunde aufsuchen. Um das zu vereinfachen, sollen die Sprechstundentermine aller Lehrenden in die Datenbank aufgenommen werden. Dazu soll das Datenbankschema um das folgende Relationenschema ergänzt werden:

Sprechstunde(Kuerzel, Name, Tag, Stunde, Anfangszeit, Raum)

Abbildung 2 zeigt eine Beispielrelation zum Relationenschema Sprechstunde.

Sprechstunde					
<u>Kuerzel</u>	Name	Tag	Stunde	Anfangszeit	Raum
Hu	David Hume	Fr	1	08:00	204
Lo	John Locke	Do	3	09:55	112

Abbildung 2: Beispielrelation mit Sprechstundenterminen

In einer Relation zum Relationenschema Sprechstunde ist enthalten, welche bzw. welcher Lehrende mit welchem Kürzel an welchem Tag in welcher Stunde in welchem Raum ihre bzw. seine Sprechstunde hat. Die angegebene Uhrzeit ergibt sich aus der Anfangszeit der in Stunde angegebenen Schulstunde. Beim Ausdrucken der Sprechstunden sollen die Datensätze nach den Nachnamen der Lehrenden sortiert werden.

- b) *Begründen Sie, inwieweit das Relationenschema Sprechstunde nicht den Kriterien der dritten Normalformen entspricht. Geben Sie alle Gründe an.*

Entwickeln Sie zum Relationenschema Sprechstunde ein Datenbankschema, das sich in der dritten Normalform befindet, und erläutern Sie die Änderungen, die zur Überführung in die dritte Normalform nötig sind.

(12 Punkte)



Name: _____

Um den Schülerinnen und Schülern, die als Tutoren an diesem System teilnehmen wollen, einen Anreiz zu bieten, sollen zwei Maßnahmen ergriffen werden. Zum einen werden die Tutoren für ihre Tätigkeit mit einer festen, von ihrer Jahrgangsstufe abhängigen Grundentlohnung bezahlt und zum anderen können sie sich je nach Engagement verschiedene Auszeichnungen verdienen, die deutlich machen, in welchen Fächern sie sich besonders hervorgetan haben. Die Auszeichnungen werden in verschiedenen Rangstufen (Bronze, Silber, Gold und Platin) vergeben. Abhängig vom Rang wird pro Auszeichnung ein Bonus auf die Grundentlohnung gezahlt. Natürlich können Tutoren auch mehrere Auszeichnungen in verschiedenen Fächern erhalten und somit auch mehrere Boni bekommen.

Zur Verwaltung dieses Entlohnungssystems wird eine Datenbank mit dem folgenden Datenbankschema verwendet:

Tutor(TutorID, Vorname, Nachname, Grundentlohnung,
Kontonummer, ↑BankID)
Bank(BankID, Name)
Auszeichnung(AuszeichnungID, Fach, ↑RangID)
Rang(RangID, Bezeichnung, Bonus)
hat(↑TutorID, ↑AuszeichnungID)

c) Auf dem obigen Datenbankschema sollen die folgenden Anfragen realisiert werden:

- I. Gesucht sind die Vor- und Nachnamen aller Tutoren, die eine höhere Grundentlohnung bekommen als der Durchschnitt aller Grundentlohnungen.
- II. Gesucht sind die Nach- und Vornamen aller Tutoren und die Fächer, in denen sie eine Auszeichnung haben. Das Ergebnis soll alphabetisch nach Nachnamen und ggf. Vorname sortiert sein und auch die Tutoren beinhalten, die keine Auszeichnungen haben.
- III. Gesucht sind die Nachnamen aller Tutoren, die über **keine** Auszeichnung mit einem Rang verfügen, welcher die Bezeichnung „Silber“ hat.

Entwickeln Sie jeweils eine SQL-Anweisung für die obigen Anfragen.

(12 Punkte)



Name: _____

d) Die folgende SQL-Anweisung wird ebenfalls auf dem obigen Datenbankschema ausgeführt:

```
1  SELECT Tutor.Nachname, Tutor.Vorname, SUM(tmp.b)
2  FROM Tutor
3     LEFT JOIN hat ON Tutor.TutorID = hat.TutorID
4     LEFT JOIN (
5         SELECT Auszeichnung.AuszeichnungID AS a,
6                Rang.Bonus AS b
7         FROM Auszeichnung JOIN Rang
8              ON Auszeichnung.RangID = Rang.RangID
9     ) AS tmp ON hat.AuszeichnungID = tmp.a
10 GROUP BY Tutor.Nachname, Tutor.Vorname
11 ORDER BY Tutor.Nachname, Tutor.Vorname
```

Anmerkung: In einigen Datenbanksystemen muss in Zeile 6 das Schlüsselwort JOIN durch INNER JOIN ersetzt werden.

Erläutern Sie die Unterabfrage in den Zeilen 5 bis 7 und ermitteln Sie ihr Ergebnis bei Anwendung auf die Beispieldatensätze in der Anlage.

Erläutern Sie anschließend die gesamte SQL-Anweisung und welche Information im Sachzusammenhang ermittelt wird.

(10 Punkte)



Name: _____

- e) In einem später verworfenen Modellierungsvorschlag der Datenbank sollte für jede Nachhilfeschülerin und jeden Nachhilfeschüler auch noch die aktuelle Klassenlehrerin bzw. der aktuelle Klassenlehrer gespeichert werden (vgl. Abbildung 3).

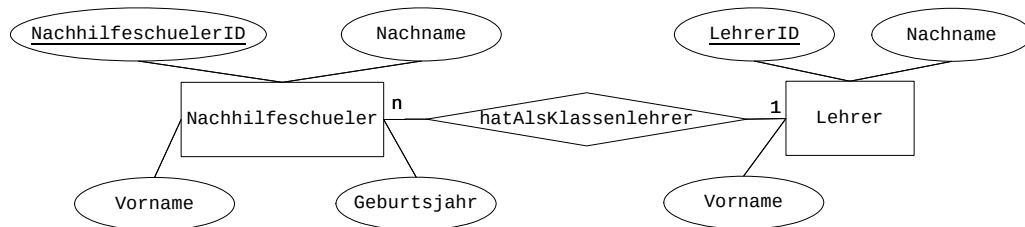


Abbildung 3: Auszug aus dem Datenbankentwurf mit Modellierung der Klassenleitung

Der Beziehungstyp **hatAlsKlassenlehrer** modelliert eine 1:n-Beziehung, die mit folgendem Relationenschema umgesetzt werden sollte:

hatAlsKlassenlehrer(↑NachhilfeschueelerID, ↑LehrerID)

*Begründen Sie, warum das Relationenschema **hatAlsKlassenlehrer** eine mögliche Umsetzung der entsprechenden 1:n-Beziehung aus Abbildung 3 darstellt.*

Nehmen Sie zu dieser Umsetzung Stellung.

(6 Punkte)

Zugelassene Hilfsmittel:

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



Name: _____

Anlage:

Tutor					
<u>TutorID</u>	Nachname	Vorname	Grundentlohnung	↑BankID	Kontonummer
1	Herrmann	Elena	20	3704736	45623413
2	Yildiz	Tolga	30	3704736	98237456
3	Schmidt	Thomas	20	4723434	89273459
4	Taubert	Tina	30	3704736	42348756
5	Baumann	Tom	10	3704736	76376234

Auszeichnung		
<u>AuszeichnungID</u>	Fach	↑RangID
1	Mathematik	1
2	Deutsch	2
3	Mathematik	3
4	Physik	1

Rang		
<u>RangID</u>	Bezeichnung	Bonus
1	Bronze	5
2	Silber	10
3	Gold	15
4	Platin	20

hat	
↑ <u>TutorID</u>	↑ <u>AuszeichnungID</u>
1	1
2	4
3	2
4	1
1	4

Bank	
<u>BankID</u>	Name
3704736	Kreisbank Köln
4723434	Bezirksbank Köln

*Unterlagen für die Lehrkraft***Abiturprüfung 2016***Informatik, Leistungskurs***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 gehörige 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 DISTINCT Tutor.Vorname, Tutor.Nachname
 FROM Tutor, (
 SELECT SUM(Tutor.Grundentlohnung)/COUNT(Tutor.Grundentlohnung)
 AS Durchschnitt
 FROM Tutor
) AS tmp
 WHERE Tutor.Grundentlohnung > tmp.Durchschnitt

II.
 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

III.
 SELECT Tutor.Nachname
 FROM Tutor
 WHERE Tutor.TutorID NOT IN (
 SELECT DISTINCT Tutor.TutorID
 FROM Tutor
 JOIN hat ON Tutor.TutorID = hat.TutorID
 JOIN Auszeichnung ON
 hat.AuszeichnungID = Auszeichnung.AuszeichnungID
 JOIN Rang ON Auszeichnung.RangID = Rang.RangID
 WHERE Rang.Bezeichnung = "Silber"
)

Teilaufgabe d)

Die Unterabfrage in den Zeilen 5 bis 7 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 ein JOIN zwischen den Tabellen Tutor, hat und tmp durchgeführt. Der JOIN zwischen allen drei Tabellen wird jeweils als LEFT JOIN durchgeführt, so dass in der Ergebnistabelle am Ende auch diejenigen Tutoren aufgeführt werden, die gar keine Auszeichnungen haben (z. B. Tom Baumann in den vorliegenden Beispieltabellen).

In Zeile 8 wird eine Gruppierung nach Vor- und Nachname der Tutoren vorgenommen, um die Aggregatfunktion SUM auf die einzelnen Tutoren und nicht auf die Gesamtdatenmenge anwenden zu können. Anschließend wird in Zeile 9 die Tabelle noch nach Nachnamen und ggf. Vornamen aufsteigend sortiert.

Im Sachzusammenhang ermittelt die Abfrage die Information, welchen Betrag an Bonuszahlung jeder Tutor ausgezahlt bekommt. Tutoren, die keine Bonuszahlung bekommen, werden mit NULL aufgeführt.

Teilaufgabe e)

Das Relationenschema hatAlsKlassenlehrer setzt die Primärschlüssel NachhilfeschuelerID aus dem Relationenschema Nachhilfeschueler und LehrerID aus dem Relationenschema Lehrer im Sinne einer Verknüpfungstabelle zueinander in Beziehung. Da der Schlüssel NachhilfeschuelerID ein Primärschlüssel ist, LehrerID aber nicht zum Primärschlüssel gehört, kann jeder Nachhilfeschüler nur einmal in eine entsprechende Tabelle eingetragen und ihm ein Lehrer zugewiesen werden. Die ID eines Lehrers kann jedoch mehrmals vorkommen. Das entspricht der 1:n-Beziehung aus Abbildung 3, die somit korrekt umgesetzt ist.

Die 1:n-Beziehung aus Abbildung 3 könnte ebenso realisiert werden, indem der Primärschlüssel aus dem Relationenschema Lehrer als Fremdschlüssel in das Relationenschema Nachhilfeschueler eingetragen wird. Eine zusätzliche Verknüpfungstabelle anzulegen, ist daher nicht notwendig.

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.	4			
2	entwickelt die SQL-Anweisung zur zweiten Anfrage.	4			
3	entwickelt die SQL-Anweisung zur dritten Anfrage.	4			
Sachlich richtige Lösungsalternative zur Modelllösung: (12)					
	Summe Teilaufgabe c)	12			

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	begründet, inwieweit die Relation eine korrekte Umsetzung der Beziehung ist.	3			
2	nimmt Stellung zur Umsetzung der Beziehung.	3			
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 Punktschme aus der ersten bearbeiteten Aufgabe	50			
Übertrag der Punktschme aus der zweiten bearbeiteten Aufgabe	50			
Übertrag der Punktschme aus der dritten bearbeiteten Aufgabe	50			
Punktzahl der gesamten Prüfungsleistung	150			
aus der Punktschme 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	150 – 143
sehr gut	14	142 – 135
sehr gut minus	13	134 – 128
gut plus	12	127 – 120
gut	11	119 – 113
gut minus	10	112 – 105
befriedigend plus	9	104 – 98
befriedigend	8	97 – 90
befriedigend minus	7	89 – 83
ausreichend plus	6	82 – 75
ausreichend	5	74 – 68
ausreichend minus	4	67 – 60
mangelhaft plus	3	59 – 50
mangelhaft	2	49 – 40
mangelhaft minus	1	39 – 30
ungenügend	0	29 – 0