

---

# Klausur der Lehrveranstaltung "Datenbanksysteme"

31. Juli 2013

---

Hier bitte Namensschild aufkleben

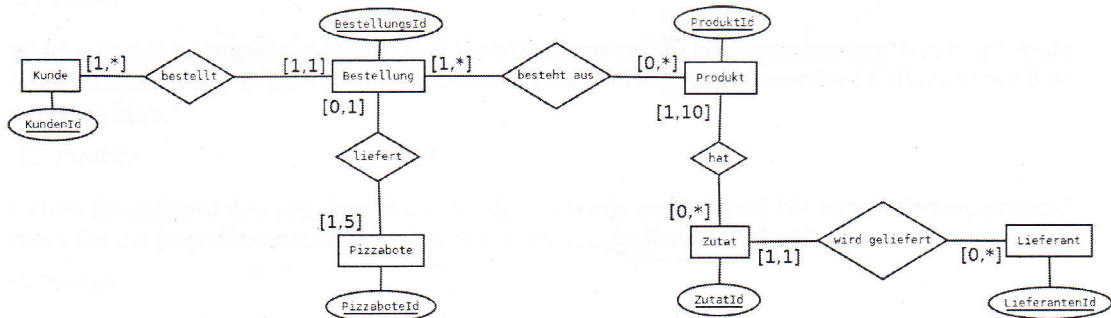
## Hinweis

Kleben Sie zunächst das Namensschild auf das Titelblatt und versehen Sie alle Blätter mit Ihrem Namen und Ihrer Matrikelnummer. Schreiben Sie die Lösungen nur auf das jeweilige zur Aufgabe passende Lösungsblatt im Anhang der Klausur. Multiple-Choice-Fragen werden direkt auf den Frageblättern beantwortet. Sie können die Lösungsblätter auf beiden Seiten beschreiben. Sollte Ihnen wider Erwarten der Platz auf einem Blatt nicht ausreichen, so können Sie von der Klausuraufsicht weitere Blätter erhalten, die Sie wiederum mit Ihrem Namen und Ihrer Matrikelnummer kennzeichnen. Geben Sie alle Blätter dieser Klausur wieder ab, da nachträglich abgegebene Blätter nicht korrigiert werden können.

Aufgabe:	1	2	3	4	$\Sigma$
Punkte:					
Note:					

## Aufgabe 1 – ER-Modellierung (15 Punkte)

Gegeben sei folgendes ER-Modell der Datenbank einer Pizzeria:



Alle Kardinalitätsangaben wurden in Teilnehmerkardinalität notiert.

- a) Welche der folgenden Aussagen werden durch diese Modellierung impliziert? Richtige Antworten werden mit 0,5 Punkten bewertet, bei falschen Antworten werden 0,5 Punkte abgezogen. Insgesamt wird die Aufgabe mit mindestens 0 Punkten bewertet.

	WAHR	FALSCH
Die Pizzeria bezieht eine bestimmte Zutat immer von genau einem Lieferant.		
Alle Bestellungen werden ausgeliefert.		
Ein Kunde hat genau eine Bestellung.		
Produkte haben keine identischen Zutaten.		
Eine Zutat kann in mehreren Produkten verwendet werden.		
Sei $N(E)$ die Anzahl der Instanzen des Entity-Typs $E$ . Es gilt immer $N(Zutat) \geq N(Produkt)$		
Ein Kunde wird immer vom gleichen Pizzaboten beliefert.		

(3,5 Punkte)

- b) Bilden Sie das gegebene ER-Modell auf ein entsprechendes Relationenmodell ab. Halten Sie sich dabei an folgende Notation:

- **Relationenschemata** sollen durch folgende Notation angegeben werden:

$$R = \{A_1, \dots, A_n\}$$

wobei  $A_1, \dots, A_n$  die Attribute des Relationschemas sind. Sie dürfen statt  $R$  aussagekräftigere Namen für die Relationenschemata verwenden.

- **Schlüssel** sind durch die mengenartige Notation aus der Vorlesung zu kennzeichnen:

$$K_R = \{\{A_i, \dots, A_k\}, \dots\}$$

- **Fremdschlüsselbedingungen** für jedes Relationenschema  $R$  werden in folgender Form angegeben:

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

$$X(R) \rightarrow Y(R')$$

Dabei ist  $X$  eine Attributliste im Relationenschema  $R$  und  $Y$  die entsprechende Attributliste, die Schlüssel von  $R'$  ist.

(5 Punkte)

- c) Welche Probleme ergeben sich aus dem Verschmelzen von Relationenschemata, wenn optionale Beziehungen wie z. B. *liefert* auf das Relationenmodell abgebildet werden? Erläutern Sie Ihre Antwort kurz.

(1,5 Punkte)

- d) Geben Sie anhand des gegebenen ER-Modells jeweils ein Beispiel für eine Relation, anhand derer Sie die Begriffe kapazitätsvermindernd sowie kapazitätserhöhend erläutern.

(5 Punkte)

## Aufgabe 2 – Funktionale Abhängigkeiten (15 Punkte)

Gegeben sei eine Relation  $R(A, B, C, D, E, F, G, H, I, J)$  mit den folgenden funktionalen Abhängigkeiten:

$$F = \{ \begin{array}{l} A \rightarrow CE, \\ AC \rightarrow F, \\ AI \rightarrow J, \\ BE \rightarrow F, \\ C \rightarrow AE, \\ D \rightarrow BE, \\ E \rightarrow B, \\ EF \rightarrow G, \\ FG \rightarrow H, \\ FJ \rightarrow A, \\ G \rightarrow B \end{array} \}$$

Alle Attribute der Relation sind atomar.

- a) • Welches Problem wird durch den RAP-Algorithmus gelöst?
- Zeigen Sie, dass gilt:  $AD \rightarrow H \in F^+$ ; verwenden Sie hierbei den RAP-Algorithmus. Geben Sie in jedem Teilschritt jeweils an, welche  $f \in F$  verwendet wurde, welche Attribute in der aktuellen Hülle  $\{AD\}^*$  enthalten sind sowie welche Regel verwendet wurde.
- Verwenden Sie hierfür die Tabelle auf dem Lösungsblatt.

(5 Punkte)

- b) Geben Sie für folgende funktionalen Abhängigkeiten  $f$  an, ob diese aus  $F$  ableitbar sind, d. h. ob  $F \models f$  gilt.

FUNKTIONALE ABHÄNGIGKEIT $f$	ABLEITBAR	NICHT ABLEITBAR
$A \rightarrow BCEH$	<input type="checkbox"/>	<input type="checkbox"/>
$BDG \rightarrow BCDEFG$	<input type="checkbox"/>	<input type="checkbox"/>
$DJ \rightarrow ABCDEFGIJ$	<input type="checkbox"/>	<input type="checkbox"/>
$ABCDEFGHJ \rightarrow ABCDEFGHIJ$	<input type="checkbox"/>	<input type="checkbox"/>
$ABCD \rightarrow ABCDGF$	<input type="checkbox"/>	<input type="checkbox"/>
$EFI \rightarrow AEFGHI$	<input type="checkbox"/>	<input type="checkbox"/>
$CI \rightarrow ABCEFGHIJ$	<input type="checkbox"/>	<input type="checkbox"/>
$GI \rightarrow BDEFG$	<input type="checkbox"/>	<input type="checkbox"/>

Richtige Antworten werden mit 0,5 Punkten bewertet, bei falschen Antworten werden 0,5 Punkte abgezogen. Insgesamt wird die Aufgabe mit mindestens 0 Punkten bewertet.

(4 Punkte)



Name:

Matrikelnummer:

- c) Bestimmen Sie die Menge aller Schlüssel der Relation  $R$ . In welcher höchsten Normalform befindet sich  $R$ ? Geben Sie eine funktionale Abhängigkeit  $f \in F$  an, die verhindert, dass sich  $R$  in einer höheren Normalform befindet.

(4 Punkte)

- d) Welche der Eigenschaften

- VEBUNDTREUE
- MINIMALITÄT
- ABHÄNGIGKEITSTREUE
- NP-VOLLSTÄNDIGKEIT

gelten für den Dekompositionsalgorithmus und/oder den Synthesealgorithmus bzw. für deren Ergebnis? Gehen Sie von den Algorithmen aus der Vorlesung aus. Füllen Sie hierfür die folgende Lösungstabelle aus:

ALGORITHMUS	VEBUNDTREUE	MINIMALITÄT	ABHÄNGIG- KEITSTREUE	NP-VOLL- STÄNDIGKEIT
DEKOMPOSITION	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SYNTHESE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(2 Punkte)

### Aufgabe 3 – SQL (15 Punkte)

In einem SQL-basierten Datenbanksystem seien Fußballspiele, Experten und deren Ergebnistipps durch folgende Relationen modelliert:

```
CREATE TABLE Spiel (
  id          INTEGER,
  team_a      CHAR(3),
  team_b      CHAR(3),
  tore_a      INTEGER,
  tore_b      INTEGER,
  PRIMARY KEY (id)
);
```

```
CREATE TABLE Experte (
  id          INTEGER,
  name        VARCHAR(20),
  titelzahl   INTEGER,
  PRIMARY KEY (id)
);
```

```
CREATE TABLE Tipp (
  s_id        INTEGER,
  e_id        INTEGER,
  tore_a      INTEGER,
  tore_b      INTEGER,
  FOREIGN KEY (e_id) REFERENCES Experte(id),
  FOREIGN KEY (s_id) REFERENCES Spiel(id)
);
```

**Hinweise:** team\_a bezeichnet die Heimmannschaft, team\_b die Auswärtsmannschaft eines Spiels. In einem Spiel bzw. Tipp bezeichnet tore\_a die Anzahl der Tore der Heimmannschaft, tore\_b die Anzahl der Tore der Auswärtsmannschaft. titelzahl enthält die Anzahl errungener Meistertitel des jeweiligen Experten.

Die Inhalte der Relationen sind wie folgt:

Spiel				
id	team_a	team_b	tore_a	tore_b
1	'BVB'	'VfB'	4	4
2	'BVB'	'FCB'	1	2
3	'FCB'	'VfB'	3	2

Experte		
id	name	titelzahl
4	'Lattek'	8
5	'Matthäus'	7
6	'Beckenbauer'	5

Tipp			
s_id	e_id	tore_a	tore_b
1	4	0	0
1	5	2	1
1	6	0	2
2	4	1	2
2	5	1	2
2	6	0	2
3	4	3	2
3	5	2	1
3	6	0	1

Alle Aufgaben lassen sich mit SQL-Konstrukten lösen, die in der Vorlesung vorgestellt wurden. Die Lösungen müssen dem SQL-Standard folgen und unabhängig vom Datenbankinhalt sein.

- a) Formulieren Sie eine SQL-Anfrage, die die durchschnittliche Anzahl von Titeln aller Experten ausgibt.

(1 Punkt)

- b) Erstellen Sie eine Sicht namens `FCBSpiel`, die alle Heim- und Auswärtsspiele der Mannschaft 'FCB' enthält. Ausgegeben werden sollen jeweils der Gegner, ein 'H' für ein Heimspiel bzw. ein 'A' für ein Auswärtsspiel sowie die Tore beider Mannschaften (die Tore der Mannschaft 'FCB' immer zuerst). Vergeben Sie auch aussagekräftige Attributnamen.

*Ergebnis von SELECT \* FROM FCBSpiel bei gegebenem Datenbankinhalt:*

('BVB', 'A', 2, 1), ('VfB', 'H', 3, 2).

(3 Punkte)

- c) Der Experte Beckenbauer ist in der Vergangenheit durch mehrere widersprüchliche Tipps zu einem einzigen Spiel aufgefallen. Modifizieren Sie die bereits existierende Relation `Tipp` mit einer SQL-DDL-Anweisung derart, dass jeder Experte zu jedem Spiel nur maximal einen Tipp abgeben kann.

(1 Punkt)

- d) Formulieren Sie eine SQL-Anfrage, die alle Experten bestimmt, die bei mindestens einem Spiel das richtige Torverhältnis (`tore_a - tore_b`) getippt haben. Ausgegeben werden sollen die Namen dieser Experten sowie die Anzahl der jeweils korrekten Torverhältnistipps.

*Ergebnis bei gegebenem Datenbankinhalt:*

('Lattek', 3), ('Matthäus', 2).

(2 Punkte)

- e) Die Mitglieder der Tipprunde *Stolperpass* haben nach dem Konsum größerer Biermengen festgelegt, Punkte für Tipps gemäß folgender SQL-Anfrage zu vergeben. Listen Sie kurz in natürlicher Sprache die Kriterien auf, die in dieser Tipprunde für die Punktevergabe gelten.

**Hinweise:**

- Die Funktion `sign(number)` gibt an, ob `number` kleiner (-1), gleich (0) oder größer (1) als 0 ist.
- `UNION ALL` vereinigt zwei Mengen, ohne dabei Duplikate zu eliminieren.

```
SELECT id, SUM(punkte)/titelzahl AS score FROM (
  SELECT e.id, e.titelzahl, e.titelzahl AS punkte
  FROM Experte e
  WHERE 0 = 1
  UNION ALL
  SELECT e.id, e.titelzahl, count(*) AS punkte
  FROM Experte e, Tipp t
  WHERE e.id = t.e_id
  GROUP BY e.id, e.titelzahl
  UNION ALL
  SELECT e.id, e.titelzahl, 2 AS punkte
  FROM Experte e
  CROSS JOIN Spiel s
  LEFT JOIN Tipp t ON (t.e_id = e.id AND t.s_id = s.id)
  WHERE SIGN(t.tore_a - t.tore_b) = SIGN(s.tore_a - s.tore_b)
  UNION ALL
  SELECT e.id, e.titelzahl, 7 AS punkte
  FROM Experte e
  JOIN Tipp t ON t.e_id = e.id
  JOIN Spiel s ON s.id = t.s_id
  WHERE s.tore_a = t.tore_a AND s.tore_b = t.tore_b AND t.s_id IN (
    SELECT s.id
    FROM Spiel s
    JOIN Tipp t ON s.id = t.s_id
    WHERE s.tore_a = t.tore_a AND s.tore_b = t.tore_b
    GROUP BY s.id
    HAVING COUNT(*) = 1)
)
GROUP BY id, titelzahl
ORDER BY score DESC, id ASC;
```

(5 Punkte)

- f) Formulieren Sie eine SQL-Anfrage, die alle Paarungen ausgibt, bei denen die Mannschaft 'BVB' ein Heimspiel hatte und kein Experte auf einen Heimsieg getippt hat.

*Ergebnis bei gegebenem Datenbankinhalt:*

('BVB', 'FCB').

(2 Punkte)

- g) Experte Lattek hat seinen Rücktritt erklärt. Formulieren Sie eine gültige Folge von SQL-Anweisungen, die den Experten namens 'Lattek' und alle zugehörigen Tipps aus der Datenbank entfernt.

(1 Punkt)



**Aufgabe 4 – Histories (15 Punkte)**

Gegeben seien die folgenden Transaktionen:

$$T_1 = r_1[x] \ r_1[y] \ w_1[y] \ r_1[z] \ c_1$$

$$T_2 = r_2[y] \ w_2[z] \ r_2[z] \ w_2[x] \ a_2$$

$$T_3 = w_3[x] \ w_3[x] \ r_3[x] \ w_3[y] \ c_3$$

$$T_4 = w_4[y] \ w_4[z] \ w_4[z] \ w_4[x] \ c_4$$

Basierend auf diesen Transaktionen sei folgende History definiert:

$$H = \begin{array}{cccccccccc} r_1[x] & w_4[y] & r_1[y] & r_2[y] & w_2[z] & w_4[z] & w_3[x] & w_4[z] & w_3[x] & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \\ w_1[y] & w_4[x] & c_4 & r_1[z] & r_2[z] & c_1 & w_2[x] & a_2 & r_3[x] & w_3[y] & c_3 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \end{array}$$

*Hinweis: Die Zahlen unter den Operationen entsprechen der Indizierung, die für Aufgabenteil c) verwendet werden soll.*

a) Bilden Sie die committed Projection  $C(H)$  von  $H$ .

(1 Punkt)

b) Konstruieren Sie den Serialisierbarkeitsgraphen von  $H$ . Begründen Sie, warum  $H$  nicht serialisierbar ist. Zur Erinnerung: Der Serialisierbarkeitsgraph von  $H$  enthält Kanten entsprechend der Konfliktrelationen der committed Projection  $C(H)$ .

(4 Punkte)

c) Untersuchen Sie, ob  $H$  durch das Eliminieren einer einzelnen Operation in eine serialisierbare History  $H'$  überführt werden kann. Betrachten Sie dazu alle möglichen Histories  $H'$ , die durch das Weglassen einer einzigen Operation in  $H$  entstehen, und überlegen Sie sich die Auswirkungen auf den Serialisierbarkeitsgraphen. Gibt es mehrere Möglichkeiten eine serialisierbare History  $H'$  zu erhalten? Geben Sie für alle möglichen Lösungen jeweils den Index der Operation an, die ausgelassen werden soll. *Hinweis: Bitte nutzen Sie die oben in der Aufgabenstellung verwendete Indizierung und nicht eine Indizierung innerhalb der committed Projection.*

(5 Punkte)

d) Ermitteln Sie für  $H$  sämtliche *reads-from* Beziehungen. Halten Sie sich bei der Angabe der Lösung an die Notation:

$$T_i \text{ reads } \langle \text{Variable} \rangle \text{ from } T_j.$$

Analysieren Sie für **jede** *reads-from* Beziehung, ob die Rücksetzbarkeitseigenschaften *rücksetzbar* (RC) und *kaskadenfrei rücksetzbar* (ACA) erfüllt sind. Benutzen Sie für die Lösung die entsprechende Tabelle auf dem Lösungsblatt. Ist  $H$  in der Klasse der *strikt rücksetzbaren* Histories (ST)?

(5 Punkte)