

Klausur Datenbanksysteme I – Wintersemester 2014/2015

Dr. Arthur Zimek

Aufgabe 1

Ja/Nein Fragen, 1 Punkt pro richtiger Antwort, -1 Punkt pro falsche Antwort, 0 Punkte für nicht ausgefüllt und nicht unter 0 Punkte für die Gesamtaufgabe.

- Das I in ACID steht für Integration. — Richtige Antwort: Nein
- Gegeben 3 Transaktionen mit je 7, 8 und 5 Aktionen. Es gibt 20 serielle Schedules. — Richtige Antwort: Nein
- Der JOIN setzt sich aus den Grundoperationen Kreuzprodukt und Projektion zusammen. — Richtige Antwort: Nein
- Dirty Read und Dirty Update sind mögliche Anomalien im Mehrbenutzerbetrieb. — Richtige Antwort: Nein (Da es Dirty Update nicht gibt...)
- Serialisierbarer Schedule liefert stets gleiches Ergebnis wie serieller Schedule. — Richtige Antwort: Ja
- View kann Basisrelation neuer View sein. — Richtige Antwort: Ja
- Ziel der Normalisierung ist Beseitigung der funktionalen Abhängigkeiten. — Richtige Antwort: Nein
- In objektrelationalen Datenbanken können eigene Datentypen und Funktionen definiert werden. — Richtige Antwort: Ja
- Verwendung physischer Pointer erlaubt schnelleren Zugriff auf referenzierte Objekte als Fremdschlüssel. — Richtige Antwort: Ja
- Anlegen eines Index auf Attribut Ort beschleunigt folgendes Statement: UPDATE Kunde SET Kreditlimit = 0.1 * Kreditlimit WHERE Ort = "Grünwald"; — Richtige Antwort: Ja

Aufgabe 2

Relationen: R(A,B,C) mit 40 Tupeln und S(C,D,E,F) mit 15 Tupeln.

a)

A₁: SELECT * FROM R,S;

A₂: SELECT * FROM R NATURAL JOIN S;

- Wie viele Ergebnistupel in A₁? — $600 = 40 * 15$

- Wie viele Ergebnisattribute in A_1 ? — $7 = 3 + 4$
- Tupel mindestens in A_2 ? — 0
- Tupel maximal in A_2 ? — $600 = 40 * 15$
- Attribute in A_2 ? — 6, da NATURAL JOIN gleiche Attribute reduziert

b)

SELECT a,b FROM R GROUP BY a,b;

a mit 20 unterschiedlichen Ausprägungen, b mit 5 unterschiedlichen Ausprägungen.

- Tupel mindestens? — 20 = unterschiedliche Ausprägungen von a, wonach gruppiert wird
- Tupel maximal? — $100 = 20 * 5$

c)

Relationen:

Runde1(Matrikelnummer, OK1, OK2, OK3)

Runde2(Matrikelnummer, OK1, OK2, OK3)

Studenten nehmen an einem Auswahlverfahren teil. Verfahren hat 2 Runden, die jeweils als Relationen gegeben sind mit Matrikelnummer der Studenten und den Wertungen für die 3 Aufgaben, die ihnen gegeben werden.

Die Wertungen für die Aufgaben haben den Datentyp Integer und sind mit 0 vorbelegt. Wird die jeweilige Aufgabe richtig bearbeitet, so wird die Wertung auf 1 gesetzt.

Teilnahme an Runde2 ist zum Bestehen des Verfahrens nötig.

Bestanden hat ein Student, wenn er mindestens 2 Aufgabe in Runde1 erfolgreich bearbeitet hat oder mindestens 2 Aufgaben in Runde2 erfolgreich bearbeitet hat.

Geben Sie für die folgenden SQL Anfragen an, ob sie das korrekte Ergebnis liefern.

Pro Anfrage die richtig eingeschätzt wird gibt es 2 Punkte, pro falscher Antwort -2 Punkte. Nichtbearbeitet gibt 0 Punkte und Sie können nicht weniger als 0 Punkte in dieser Aufgabe bekommen.

- SELECT DISTINCT Matrikelnummer
FROM Runde1
WHERE (OK1=1 AND (OK2=1 OR OK3=1) OR (OK2=1 AND OK3=1))

```

UNION SELECT DISTINCT Matrikelnummer
FROM Runde2
WHERE (OK1=1 AND (OK2=1 OR OK3=1) OR (OK2=1 AND OK3=1);

```

Falsch, da so auch Bestehen von Runde1 ohne Teilnahme an Runde2 ausreichen würde.

- SELECT DISTINCT a.Matrikelnummer
FROM Runde2 a, Runde1 b
WHERE a.OK1 + a.OK2 + a.OK3 >= 2
OR (a.Matrikelnummer = b.Matrikelnummer
AND b.OK1 + b.OK2 + b.OK3 > 1);

Richtig.

- SELECT b.Matrikelnummer
FROM Runde1 a JOIN Runde2 b
ON (a.Matrikelnummer = b.Matrikelnummer)
WHERE a.OK1 + a.OK2 + a.OK3 >= 2
OR b.OK1 + b.OK2 + b.OK3 >= 2;

Falsch, da JOIN als INNER JOIN defaulted und damit keine Tupel ohne Matches erhält – wer an Runde2 erfolgreich teilgenommen hat aber nicht zu Runde1 erschienen ist fliegt hier raus.

- SELECT b.Matrikelnummer
FROM Runde1 a RIGHT JOIN Runde2 b
ON (a.Matrikelnummer = b.Matrikelnummer)
WHERE a.OK1 + a.OK2 + a.OK3 >= 2
OR b.OK1 + b.OK2 + b.OK3 >= 2;

Richtig, da gegenüber vorheriger Anfrage ein rechtsseitig erhaltender RIGHT (OUTER) JOIN verwendet wird.

Aufgabe 3

Stadt, Land, Fluss mal anders: Die relationale Datenbank eines geographischen Systems beinhaltet Städte, Länder und Flüsse mit folgenden Einzelheiten:

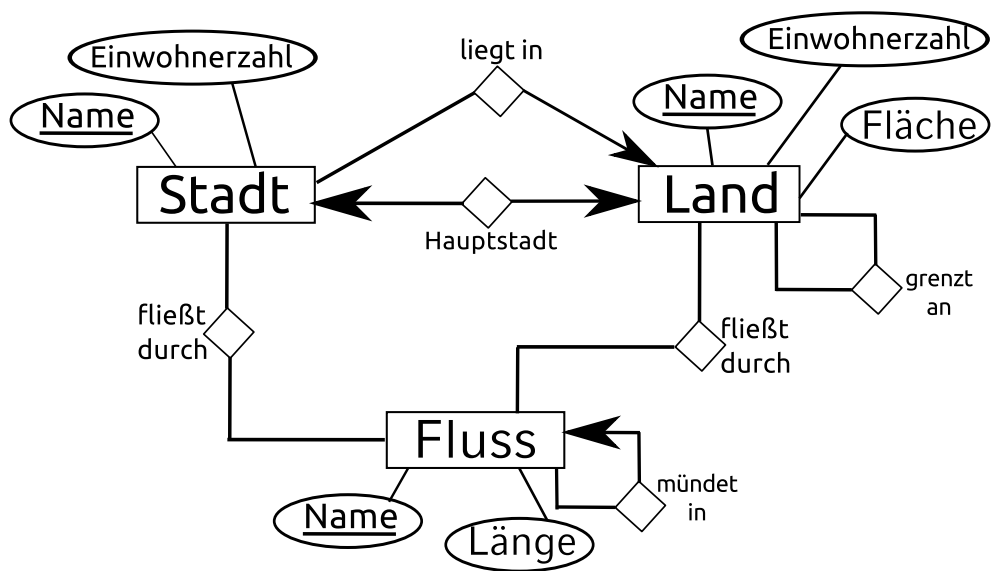
Eine Stadt wird durch ihren Namen und das Land gekennzeichnet, in dem sie liegt. Außerdem hat jede Stadt eine Einwohnerzahl.

Ein Land hat einen eindeutigen Namen, eine Fläche und ebenfalls eine Einwohnerzahl. Ein Land kann an mehrere Länder angrenzen. In einem Land liegt mindestens eine Stadt, wobei jede Stadt nur in genau einem Land liegt. Außerdem hat jedes Land eine Hauptstadt.

Ein Fluss wird durch seinen Namen bestimmt und hat eine gewisse Länge. Jeder Fluss kann durch mehrere Städte und auch durch mehrere Länder fließen und kann am Ende in einen anderen Fluss münden. Eine Stadt kann an mehreren Flüssen liegen und durch ein Land können mehrere Flüsse fließen.

Erstellen Sie ein ER-Diagramm für obige Datenbank. Markieren die die Funktionalitäten der Relationships und unterstreichen sie den Primärschlüssel jeder Entität.

Lösung:



Aufgabe 4

a)

Gegeben sind folgende Relationen:

X:

A	B	C	D
1	4	6	3
6	2	5	1
4	5	6	3
1	4	5	2
1	4	6	2
3	3	5	2
4	3	1	2
4	5	6	2

Y:

C	D	E
5	2	3
6	2	1
6	3	1

Z:

D	F
6	2
5	6
1	3

Geben Sie die Ergebnisrelationen folgender Ausdrücke der relationalen Algebra als Tabellen an.

- $X \bowtie \sigma_{F>2}(\pi_F(Z))$

A	B	C	D	F
1	4	6	3	3
4	5	6	3	3

- $(\pi_{C,D}(X) - \pi_{C,D}(Y)) \times \pi_D(Z)$

C	D	D
5	1	6
5	1	5
5	1	1
1	2	6
1	2	5
1	2	1

- $\pi_{A,D}(X) \div \pi_D(Y)$

A
1
4

b)

Gegeben sind die Relationen $R(a,b,c)$, $S(a,e,f)$ und $T(a,h)$, Wertebereich gleichnamiger Attribute stimmen überein.

Außerdem gegeben ist der Ausdruck: $\pi_{e,h}(\sigma_{b=10}((R \bowtie T) \bowtie S))$.

Entscheiden Sie ob folgende Ausdrücke mit dem gegebenen Ausdruck äquivalent sind.

- $\pi_{e,h}((\sigma_{b=10}(R)) \bowtie (\pi_{a,e}(S)) \bowtie T)$ — Ja
- $\pi_{e,h}(\sigma_{b=10}(((\pi_b(R)) \bowtie (\pi_{a,e}(S))) \bowtie (\pi_{a,h}(T))))$ — Nein
- $\pi_{e,h}(((\pi_{a,b}(\sigma_{b=10}(R))) \bowtie T) \bowtie (\pi_{a,e}(S)))$ — Ja
- $(\pi_{e,h}(R - \sigma_{b \neq 10}(R))) \bowtie (\pi_{e,h}(S \bowtie T))$ — Nein

Aufgabe 5

Gegeben sind die Relationen:

- $\text{Wirt}(\underline{\text{WNR}}, \text{WName}, \text{WStadt})$
- $\text{Speise}(\underline{\text{SNr}}, \text{Bezeichnung}, \text{Preis})$
- $\text{Gast}(\underline{\text{GNr}}, \text{GName}, \text{GStadt})$
- $\text{Bestellung}(\underline{\text{WNR}}, \underline{\text{SNr}}, \underline{\text{GNr}})$

Gesucht sind Name, Nummer und Stadt der Gäste, die niemals ein Schnitzel bestellt haben.

Formulieren Sie jeweils im Tupel- & Bereichskalkül diese Anfrage.

a) Tupelkalkül:

$\text{Schema}(g) = \text{Schema}(\text{Gast})$
 $\{g \mid \text{Gast}(g) \wedge \nexists s \in \text{Speise}:$
 $(s.\text{Bezeichnung} = \text{'Schnitzel'} \wedge \exists b \in \text{Bestellung}:$
 $b.\text{GNr} = g.\text{GNr} \wedge b.\text{SNr} = s.\text{SNr})\}$

b) Bereichskalkül:

$\{g_1, g_2, g_3 \mid \text{Gast}(g_1, g_2, g_3) \wedge \nexists bw, bs, sp:$
 $(\text{Bestellung}(bw, bs, g_1) \wedge \text{Speise}(bs, \text{'Schnitzel'}, sp))\}$

Aufgabe 6

Gegeben ist das Relationenschema $R(A, B, C, D, E, F)$ mit folgenden Abhängigkeiten:

- $C \rightarrow B$
- $B \rightarrow A$
- $C, E \rightarrow D$
- $E \rightarrow F$
- $C, E \rightarrow F$
- $C \rightarrow A$

a)

Bestimmen Sie alle Schlüsselkandidaten von R und begründen Sie, warum es keine weiteren Schlüsselkandidaten gibt.

$\{C, E\}$ ist einziger Schlüsselkandidat.

Ist eindeutig, weil $\{C, E\}^+ = \{A, B, C, D, E, F\}$.

Ist minimal, da man von C oder E alleine nicht alles erreichen kann: $\{C\}^+ = \{A, B, C\}$ und $\{E\}^+ = \{E, F\}$.

Es gibt keine weiteren Schlüsselkandidaten, da C und E nicht anders erreichbar sind.

b)

Überführen Sie R mit Hilfe des Synthesalgorithmus in die 3. Normalform. Führen Sie hierfür jeden der 4 Schritte durch und kennzeichnen Sie die Stellen, an denen nichts zu tun ist.

1. Kanonische Überdeckung

i) Linksreduktion • $C \rightarrow B$

- $B \rightarrow A$
- $C, E \rightarrow D$
- $E \rightarrow F$
- $C \rightarrow A$

ii) Rechtsreduktion • $C \rightarrow B$

- $B \rightarrow A$
- $C, E \rightarrow D$
- $E \rightarrow F$
- $C \rightarrow \emptyset$

iii) Entfernung von rechtsleeren Abhängigkeiten • $C \rightarrow B$

- $B \rightarrow A$
- $C, E \rightarrow D$
- $E \rightarrow F$

iv) Zusammenfassen von Abhängigkeiten mit gleicher linker Seite •

- $C \rightarrow B$
- $B \rightarrow A$
- $C, E \rightarrow D$
- $E \rightarrow F$

2. Erzeugen des neuen Relationenschemas

- $R1(\underline{C}, B)$
- $R1(\underline{B}, A)$
- $R1(\underline{C}, \underline{E}, D)$
- $R1(\underline{E}, F)$

3. Rekonstruktion des Schlüsselkandidaten

nichts zu tun

4. Elimination überflüssiger Relationen

nichts zu tun