

# Gedächtnisprotokoll der Klausur zur Vorlesung Datenbanksysteme I

im Wintersemester 2015/16

vom 27.01.2016

Prof. Christian Böhm

Bearbeitungszeit 90 Minuten.

**Notenschlüssel:** (max. 67 Punkte)

<b>Note</b>	5.0	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0
<b>Ab</b>	0	33.5	37	40.5	44	47.5	51	54.5	58	61.5	65

*In roter Farbe: Unsicherheiten bzgl. der Angabenstellung oder in meinem Lösungsvorschlag.*

## 1. Aufgabe: Allgemeine Fragen.

a) Nennen Sie die vier Bestandteile des ACID-Prinzips und erklären Sie diese kurz.

**A** – Atomicity: Eine Transaktion wird entweder ganz oder garnicht ausgeführt.

**C** – Consistency: Eine Transaktion überführt die DB von einem konsistenten Zustand in einen anderen konsistenten Zustand.

**I** – Isolation: Logischer Einbenutzerbetrieb.

**D** – Durability: Der Effekt einer erfolgreich abgeschlossenen Transaktion wird dauerhaft in der DB gespeichert.

b) Was versteht man unter einer Relation? (Es genügt eine Definition als Antwort)

*z.B.: Mathematische Definition:*

Relation  $r$  ist Teilmenge des kartesischen Produkts von  $k$  Wertebereichen  $D_1, \dots, D_k$

mit  $r \subseteq D_1 \times \dots \times D_k$

c) Gegeben seien  $R = (A,B,C)$  und  $S = (C,D,E)$ . Relation  $R$  enthalte 30 Tupel und Relation  $S$  enthalte 10 Tupel. Sei  $p$  ein beliebiges Selektionsprädikat. Gegeben seien folgende Aussagen:

```
A1: SELECT *  
FROM R, S  
WHERE p
```

```
A2: SELECT DISTINCT A  
FROM R, S
```

Wie viele Ergebnistupel liefert A1 mindestens?

0

Wie viele Ergebnistupel liefert A1 höchstens?

$30 \cdot 10 = 300$

Wie viele Ergebnistupel liefert A2 mindestens?

1

Wie viele Ergebnistupel liefert A2 höchstens?

**300**

- d) Nennen Sie drei mögliche Anomalien, die bei einem unkontrollierten Mehrnutzerbetrieb auftreten können.

Lost update  
dirty read  
non-repeatable read

- e) Gegeben seien Transaktionen T1, T2, T3, wobei T1 aus 5 Einzelaktionen, T2 aus 7 Einzelteilen und T3 aus 8 Einzelaktionen besteht. Wie viele mögliche **serielle** Schedules gibt es?

$3! = 6$  (Bei seriellen Schedules werden die Einzeloperationen der Tr. nicht verzahnt.)

- f) Was unterscheidet eine **materialisierte View** von einer **gewöhnlichen View**?

Materialisierte Views werden einmal berechnet und dann real gespeichert. Anfragen werden dann an diese gespeicherten Ergebnisrelationen gestellt und nicht jedes Mal neu berechnet. „Gewöhnliche“ Views werden bei jeder Anfrage neu berechnet.

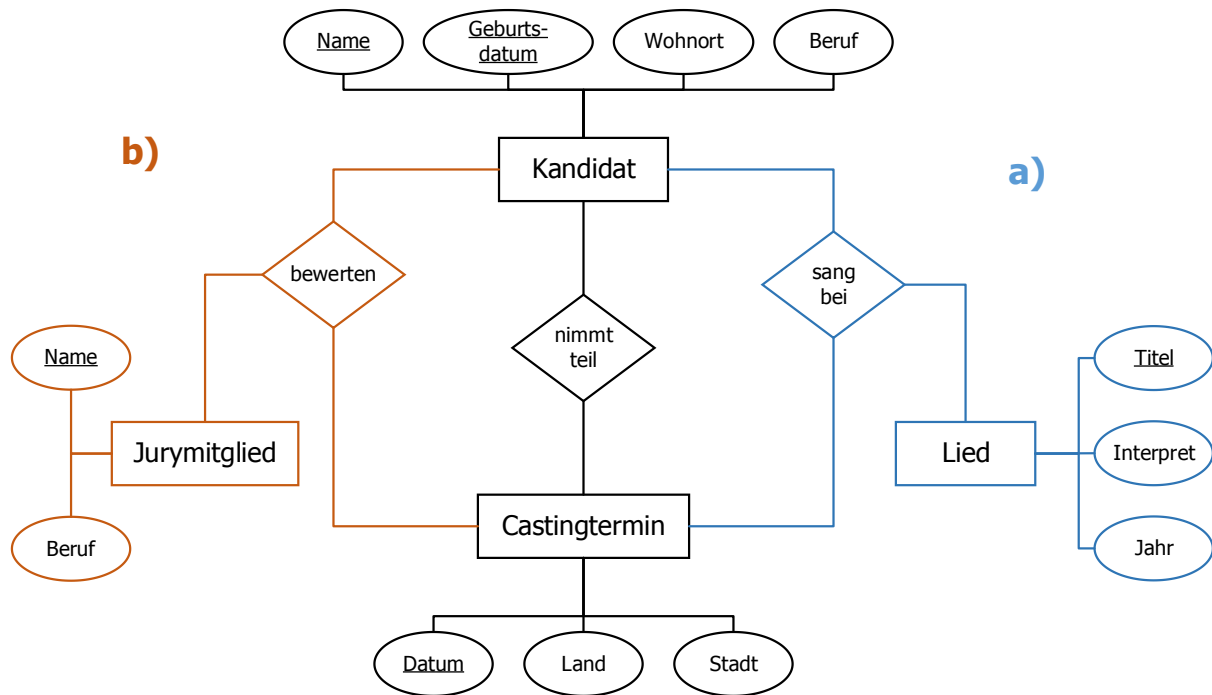
- g) Was ist ein Feature-Vektor und wozu kann er verwendet werden? Geben Sie auch ein Beispiel für einen Feature-Vektor an.

Ein Feature-Vektor ist eine Reihe an Merkmalen, die ein Objekt kennzeichnen. Bei der Suche nach ähnlichen Objekten in der DB werden nun die Eigenschaften der jeweiligen Feature-Vektoren verglichen.

Z.B. könnte ein Feature-Vektor für Autos folgendermaßen aussehen:  
(<PS>, <Gewicht>, <Farbe>)

## 2. Aufgabe: E/R-Diagramm.

In einer relationalen Datenbank sollen Informationen über die Castingshow *Deutschland sucht den Superstar* erfasst werden. Für jeden Castingtermin muss gespeichert werden, wann und wo er stattgefunden hat, wobei nicht mehrere Termine an einem Tag, aber mehrere zu unterschiedlichen Terminen am selben Ort veranstaltet werden können. Für jeden Kandidaten, identifiziert durch Name und Geburtsdatum, sollen Wohnort und Beruf hinterlegt werden, sowie, bei welchen Castingterminen er bisher aufgetreten ist. Zu diesem Zweck wurde zunächst folgendes E-/R-Diagramm entworfen:



Die Primärschlüssel waren in der Angabe nicht unterstrichen. Vermutlich gab es darauf Punkte.

- a) Zusätzlich soll nun erfasst werden, mit welchen Liedern der Kandidat beim Castingtermin aufgetreten ist. Zu jedem Lied soll dessen eindeutiger Titel, der (ursprüngliche) Interpret und das Erscheinungsjahr gespeichert werden. Erweitern Sie das E-/R-Diagramm entsprechend.
- b) Die Jury bewertet die Leistungen der Kandidaten beim Castingtermin (nicht bei einzelnen Liedern) mit „ja“ oder „nein“. Auch dies soll in einer Datenbank gespeichert werden. Integrieren Sie folgende Relationen in das E-/R-Diagramm:

**Jurymitglied** (Name, Beruf)

**bewerten** (Jurymitglied, Kandidat, Geburtsdatum, Castingtermin, Urteil)

### 3. Aufgabe: Relationale Algebra.

Bilden Sie folgende Operationen durch 5 relationale Grundoperationen (Vereinigung, Differenz, Kartesisches Produkt, Selektion, Projektion) nach:

a) Durchschnitt

$$A \cap B = A - (A - B)$$

b) Join = natural Join ?

c) Quotient

Sei (a-b) die Menge an Attributen, die in A enthalten sind, aber nicht in B.

$$A / B = \pi_{a-b}(A) - \pi_{a-b}(\pi_{a-b}(A) \times B - A)$$

## 4. Aufgabe: SQL-Anfragen

Gegeben sei folgendes Datenbankschema: Die Attribute Preis und Menge seien ganzzahlig, alle anderen Attribute seien Zeichenketten. Im folgenden Schema sind Primärschlüssel unterstrichen, Fremdschlüssel kursiv (überstrichen), wobei Fremdschlüssel referenzierte Relation in eckigen Klammern nach Fremdschlüsselattribut angegeben ist.

**Lieferant**(LNR, LName, LStadt)

**Ware**(WNR, Bezeichnung, Preis)

**Kunde**(KNR, KName, KStadt)

**Auftrag**(LNR[Lieferant], WNR[Ware], KNR[Kunde], Menge)

- a) Geben Sie die Anweisung in SQL-DDL an, um die Relation „Auftrag“ zu erzeugen. Achten Sie auf Fremdschlüsselbeziehungen. Das Attribut „Menge“ soll stets größer Null sein.

```
CREATE TABLE Auftrag (  
    LNR varchar(n) references Lieferant,  
    WNR varchar(n) references Ware,  
    KNR varchar(n) references Kunde,  
    Menge Integer check (Menge >= 0)  
    primary key (LNR, WNR, KNR)  
);
```

- b) Übersetzen Sie folgende verbal formulierte Anfrage in SQL:  
Gesucht sind die Nummer aller Waren, die Lieferant Huber liefert.

```
SELECT DISTINCT WNR  
FROM Lieferant natural join Auftrag natural join Ware  
WHERE Lieferant.LName = 'Huber'
```

- c) Übersetzen Sie folgende verbal formulierte Anfrage in SQL:  
Gesucht sind alle Städte, aus denen Münchener Lieferanten Aufträge erhalten haben.

```
SELECT DISTINCT KStadt  
FROM Kunde natural join Auftrag natural join Lieferant  
WHERE LStadt = 'München'
```

- d) Übersetzen Sie folgende verbal formulierte Anfrage in SQL:  
Gesucht ist die Liste der Namen aller Lieferanten mit dem Namen ihres jeweiligen Hauptabnehmers und dem Prozentsatz, den der Umsatz des Hauptabnehmers bei diesem Lieferanten am Gesamtumsatz des Lieferanten ausmacht. Der Hauptabnehmer des Lieferanten ist der Kunde, dessen Aufträge beim Lieferanten im Vergleich zu den Aufträgen seiner anderen Kunden den größten Umsatz ausmacht. Falls ein Lieferant mehrere Hauptabnehmer hat, deren Aufträge beim Lieferanten den gleichen Umsatz ausmachen, sollen sie alle in der Liste erscheinen.

```
CREATE VIEW kundenUmsatz AS
    SELECT LNR, KNR, sum(Preis*Menge) as proKunde
    FROM Auftrag natural join Ware
    GROUP BY LNR, KNR

CREATE VIEW gesamtUmsatz AS
    SELECT LNR, sum(proKunde) as gesProL
    FROM kundenUmsatz
    GROUP BY LNR

CREATE VIEW Hauptabnehmer AS
    SELECT LNR, KNR
    FROM kundenUmsatz ku1
    WHERE ku1.proKunde >= all
        (SELECT proKunde
         FROM kundenUmsatz ku2
         WHERE ku2.KNR=ku1.KNR AND ku1.LNR=ku2.LNR)
    GROUP BY LNR, KNR

SELECT LName, KName as Hauptabnehmer, Round(proKunde/gesProL*100, 2)
FROM Lieferant l, Kunde k, Hauptabnehmer ha, gesamtUmsatz gu,
kundenUmsatz ku
WHERE l.LNR=ha.LNR AND k.KNR=ha.KNR AND ku.KNR=ha.KNR AND
ka.LNR=gu.LNR
```

- e) Geben Sie für folgende Anfrage den äquivalenten Ausdruck im Tupelkalkül an:  
Gesucht ist Kundennummer, Name, Stadt der Kunden, die von allen nicht aus München stammenden Lieferanten beliefert werden.

$$\begin{aligned} \text{Schema}(k) &= \text{Schema}(\text{Kunde}) \\ \{ k \mid &\text{Kunde}(k) \wedge \exists l \in \text{Lieferant} \wedge (l.LStadt \neq \text{'München'} \wedge \\ &\forall a \in \text{Auftrag} (a.LNR=l.LNR \Rightarrow k.KNR=a.KNR)) \} \end{aligned}$$

- f) Es ist zusätzlich eine View gegeben, die durch folgenden SQL-Befehl erstellt wird:

```
CREATE VIEW L AS (
    SELECT LNR
    FROM Lieferant );
```

Ist folgender Befehl möglich? Begründen Sie.

```
INSERT INTO L VALUES (, 42');
```

Ja, da Projektionsviews (kein Join, nur Projektion="SELECT" in SQL), die kein 'DISTINCT' oder Arithmetik enthalten, updateable sind.

## 5. Aufgabe

Gegeben sei die Relation  $\mathcal{R}(A,B,C,D,E,F)$ , so wie die Menge der zugehörigen funktionalen Abhängigkeiten  $\mathcal{F}$ :

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

- a) Bestimmen Sie alle Schlüsselkandidaten der Relation R und begründen Sie, warum es keine weiteren Schlüsselkandidaten gibt.

$\{C,E\}$  ist SK, da  
 $\{C,E\}^+ = \{A,B,C,D,E,F\} \Rightarrow \{C,E\}$  ist eindeutig.  
 $\{C\}^+ = \{A,B,C\}$  und  $\{E\}^+ = \{E,F\} \Rightarrow \{C,E\}$  ist minimal.

$\{C,E\}$  ist einziger SK. Da sich C und E nicht herleiten lassen, müssen sie in jedem SK enthalten sein.  $\{C,E\}$  ist aber schon minimal.

- b) Überführen Sie die Relation R mit Hilfe des Synthesealgorithmus in die 3. Normalform. Führen Sie hierfür jeden der vier Schritte durch und kennzeichnen Sie Stellen, an denen nichts zu tun ist.

I. Kanonische Überdeckung bestimmen

a. Linksreduktion

$C \rightarrow B$   
 $B \rightarrow A$   
 $C,E \rightarrow D$   
 $E \rightarrow F$   
 $C,E \rightarrow F$  , da  $C \in \text{AttrHuelle}(\mathcal{F}, E) = \{E,F\}$   
 $C \rightarrow A$

b. Rechtsreduktion

$C \rightarrow B$   
 $B \rightarrow A$   
 $C,E \rightarrow D$   
 $E \rightarrow F$   
 $E \rightarrow F$   
 $C \rightarrow A$  , da  $A \in \text{AttrHuelle}(\mathcal{F} - (C \rightarrow A) \cup (C \rightarrow \{\}), C) = \{A,B,C\}$

c.  $X \rightarrow \{\}$  eliminieren

$C \rightarrow B$   
 $B \rightarrow A$   
 $C,E \rightarrow D$   
 $E \rightarrow F$   
 $E \rightarrow F$   
 $C \rightarrow \{\}$

d. Zusammenfassen gleicher linker Seiten

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

- II. Relationenschemata erzeugen
  - R1 (C, B)
  - R2 (B, A)
  - R3 (C, E, D)
- III. Schlüsselkandidaten rekonstruieren
  - {C,E} ist einziger SK und bereits in R3 enthalten.
- IV. Überflüssige Relationen löschen
  - Hier nichts zu tun.

$\Rightarrow \mathcal{R}(A,B,C,D,E,F)$  in 3. NF:

R1 (C, B)

R2 (B, A)

R3 (C, E, D)



## 6. Aufgabe

Gegeben seien T1, T2, T3 wie folgt:

T1 =  $\langle w(x), w(y), r(x) \rangle$

T2 =  $\langle w(y), r(z), w(z) \rangle$

T3 =  $\langle r(x), r(y), r(z) \rangle$

Außerdem seien folgende Aktionsfolgen S1, S2, S3 gegeben:

S1 =  $(w1(x), r3(x), w1(y), r3(y), w2(y), r3(z), r2(z), w2(z), r1(x))$

S2 =  $(r3(x), r3(y), r3(z), w2(y), r2(z), w2(z), w1(x), w1(y), r1(x))$

S3 =  $(r3(z), r2(z), r1(y), r3(x), r3(y), r2(z), w2(y), r1(x), r1(x))$

- a) Welche Aktionsfolgen von S1, S2, S3 stellen einen Schedule von T1, T2, T3 dar? Begründen Sie.

**S1** ist Schedule, da die Reihenfolge der Einzeloperationen in den Transaktionen bei er Verzahnung beibehalten wurde.

**S2** ist Schedule, da die Reihenfolge der Einzeloperationen in den Transaktionen beibehalten wurde.

**S3** ist kein Schedule, da die Reihenfolge der Einzeloperationen in den Transaktionen NICHT beibehalten wurde. Z.B.  $r3(z)$  vor  $r3(x)$   $\nleftrightarrow$

- b) Welche Aktionsfolgen von S1, S2, S3 stellen einen **seriellen** Schedule von T1, T2, T3 dar? Begründen Sie.

**S1** ist nicht seriell, da die Einzeloperationen der Tr. verzahnt sind. Aber evtl. serialisierbar.

**S2** ist seriell, da die Einzeloperationen der Tr. NICHT verzahnt sind. Die Tr. werden einfach nacheinander ausgeführt.

**S3** ist kein Schedule  $\Rightarrow$  auch kein serieller Schedule.

- c) Welche Aktionsfolgen von S1, S2, S3 stellen einen **serialisierbaren** Schedule von T1, T2, T3 dar? Begründen Sie (z.B. mittels eines Abhängigkeitsgraphen).

**S3** ist kein Schedule  $\Rightarrow$  kein serialisierbarer Schedule.

**S2** ist serieller Schedule  $\Rightarrow$  S2 ist trivialerweise serialisierbar.

**S1** ist serialisierbar, da der zugehörige Abhängigkeitsgraph (siehe unten) zyklensfrei ist.

**ABHÄNGIGKEITSGRAPH**