Klausur

Datenbankmanagementsysteme

Probeklausur Sommersemester 2010

<Datum>

Name: ..... Vorname: .....

Matr.Nr: ..... Studiengang: .....

Aufgabe Nr.	Max. Punkte	Erreichte Punkte
1	12	
2	42	
3	5	
4	12	
5	5	
6	14	
7	10	
Zusatzaufgabe	8	
Summe	108	
Note:		

- Bitte füllen Sie zuerst das Deckblatt aus und legen Sie **Studentenausweis** und **Personalausweis** bereit.
- Die **Bearbeitungszeit** beträgt **100 Minuten.**
- Falls der vorgesehene Platz nicht für Ihre Lösung ausreichen sollte, benutzen Sie bitte die Rückseiten bzw. die leeren Seiten am Ende der Klausur.

Wir wünschen Ihnen viel Erfolg!

Aufgabe 1: Struktur von Datenbanksystemen (12 Punkte)

Die unten angegebene Relation beschreibt Zulassungen des Straßenverkehrsamtes Essen.

- **PID** ist eine eindeutige Nummer für eine Person.
- **Nachname** und **Vorname** sind Attribute von Personen.
- **Kennzeichen** ist ein eindeutiges Kennzeichen von zugelassenen Autos.
- **AutoTypID** ist ein eindeutiger Bezeichner für ein Auto (d.h. für einen Autotypen und der dazugehörigen Bezeichnung).
- **Hersteller** beschreibt den Hersteller eines Autos.
- **Bezeichnung** gibt den vom Hersteller vergebenen Namen für ein Auto an, wobei es möglich ist, dass zwei Hersteller ihren Autos die gleichen Namen geben.
- **Baujahr** beschreibt das Baujahr eines Autos, wobei ein Autotyp unterschiedliche Baujahre haben kann.
- Das **Zulassungsdatum** beschreibt, wann das jeweilige Auto zugelassen wurde.

PID	Nachname	Vorname	Kennzeichen	AutoTypID	Hersteller	Bezeichnung	Baujahr	Zulassungsdatum
1	Matthäus	Lothar	E – DB 01	FoFi	Ford	Fiesta	1997	01.01.2001
1	Matthäus	Lothar	E – DB 02	FoMu	Ford	Mustang	1985	26.02.1988
2	Trappatoni	Giovanni	E – A 4711	OpCo	Opel	Corsa	1999	29.05.1999
2	Trappatoni	Giovanni	E – B 4711	VWPo	VW	Polo	2008	15.04.2008
3	Klinsmann	Jürgen	E – XY 777	VWPo	VW	Polo	2008	16.03.2008
3	Klinsmann	Jürgen	E – QM 12	VWGo	VW	Golf	2009	17.05.2009
3	Klinsmann	Jürgen	E – ST 44	AuTT	Audi	TT	2008	26.04.2009
4	Magath	Felix	E – ZZ 123	VWGo	VW	Golf	2009	21.08.2009
5	Magath	Margret	E – ZZ 123	VWGo	VW	Golf	2009	21.08.2009

a) Tragen Sie alle funktionalen Abhängigkeiten in die Liste unten ein

Anmerkung: Die Anzahl der Zeilen unten gibt keinen Hinweis auf die Anzahl der funktionalen Abhängigkeiten!

Lösung:

1. { PID } → { Nachname, Vorname }
2. { Kennzeichen } → { AutoTypID, Hersteller, Bezeichnung, Zulassungsdatum, Baujahr }
3. { AutoTypID } → { Hersteller, Bezeichnung }

1. {.....} → {.....}
2. {.....} → {.....}
3. {.....} → {.....}
4. {.....} → {.....}
5. {.....} → {.....}
6. {.....} → {.....}
7. {.....} → {.....}

b) Welche der oben angegebenen funktionalen Abhängigkeiten widersprechen den Normalformen 1-3? Schreiben Sie die Nummer der jeweiligen funktionalen Abhängigkeit hinter die entsprechende Normalform.

1. Normalform:

2. Normalform: 1 & 2

3. Normalform: 3

c) Bringen Sie die Tabelle in 3te Normalform. Tragen Sie dazu nur die Attributnamen (nicht die Daten) in die Tabellen unten ein.

- *Die Anzahl der Tabellen unten gibt keinen Hinweis auf die Anzahl der notwendigen Tabellen!*
- *Die Anzahl der Spalten in den Tabellen unten gibt keinen Hinweis auf die Anzahl der Spalten in den Ergebnistabellen!*

PID	Nachname	Vorname				
…	…	…	…	…	…	…

PID	Kenn-zeichen	AutoTypID	Zulassungs-datum	Baujahr		
…	…	…	…	…	…	…

AutoTypID	Hersteller	Bezeich-nung				
…	…	…	…	…	…	…

…	…	…	…	…	…	…

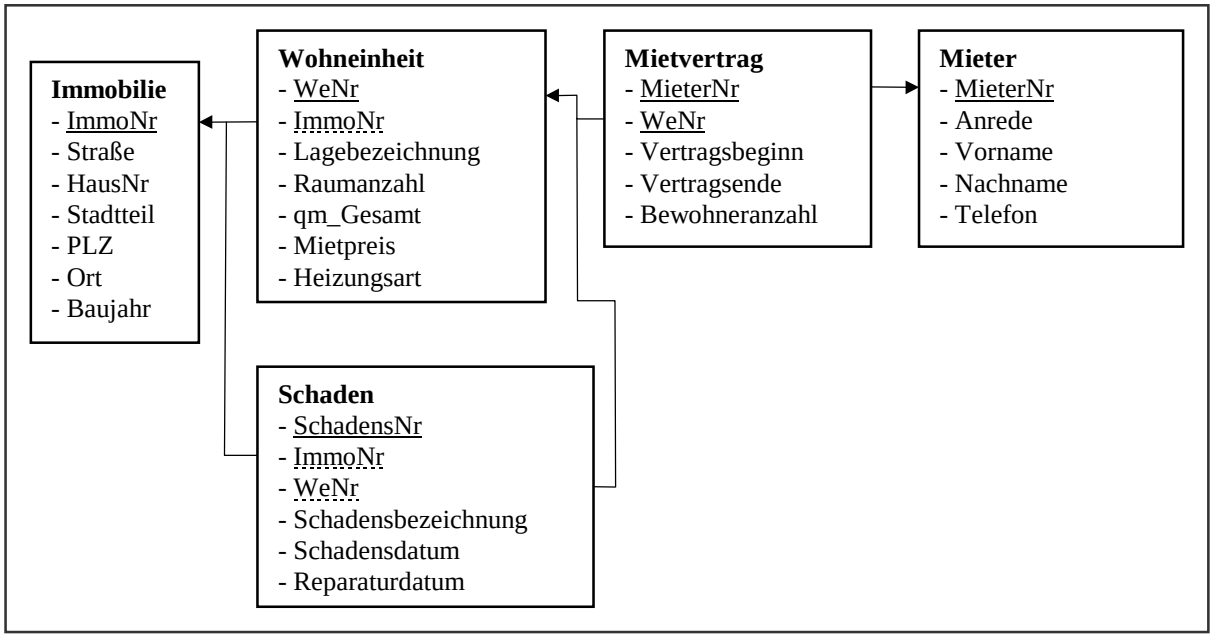
…	…	…	…	…	…	…

…	…	…	…	…	…	…

…	…	…	…	…	…	…

Aufgabe 2: SQL (42 Punkte)

Gegeben sei der folgende Auszug eines DB-Schemas einer Immobiliengesellschaft:



Bitte beachten Sie:

a) Die Pfeile in der Darstellung oben weisen immer auf das „1“-Element einer 1:n-Beziehung.
b) Der Primärschlüssel in Tabelle *Mietvertrag* setzt sich aus den Fremdschlüsselattributen *MieterNr* und *WeNr* zusammen.
c) Schäden können auch in gemeinsam genutzten Bereichen einer Immobilie auftreten (wie z.B. im Hausflur). Daher können Schäden nicht immer einer Wohneinheit zugeordnet werden. In diesem Fall ist der Wert von *WeNr* gleich *Null*.
d) Mieter können bereits in mehreren Wohnungen der Immobiliengesellschaft gewohnt haben, besitzen jedoch nie mehr als einen Mietvertrag gleichzeitig.
e) Alle Attribute mit Zeitpunktangaben sind mit dem Datentyp DATE definiert.
f) Alle Wohnungen befinden sich in derselben Stadt.

1. Geben Sie die folgenden DDL-Statements in SQL an

a) Erzeugen Sie die Tabelle *Mietvertrag* und beachten Sie hierbei die Primär- und Fremdschlüsselattribute. Wählen Sie zusätzlich geeignete Datentypen für die benötigten Attribute. **(6 Punkte)**

```
CREATE TABLE Mietvertrag (
        MieterNr            INTEGER,
        WeNr                INTEGER,
        Vertragsbeginn      DATE NOT NULL,
        Vertragsende        DATE,
        Bewohneranzahl  INTEGER,
CONSTRAINT PK PRIMARY KEY (MieterNr, WeNr),
CONSTRAINT FK1 FOREIGN KEY (MieterNr) REFERENCES Mieter,
CONSTRAINT FK2 FOREIGN KEY (WeNr) REFERENCES Wohneinheit);
```

b) Erzeugen Sie eine Sicht „Mietangebote“ (bestehend aus den Attributen *Wohneinheit.** und *Immobilie.**), die alle leerstehenden Wohneinheiten wiedergibt. D. h. es sollen sowohl die Wohneinheiten ermittelt werden, für die kein aktueller Mietvertrag vorliegt, als auch die Wohneinheiten, die noch nie vermietet wurden.
Hinweis: Das aktuelle Datum wird in SQL durch den Ausdruck CURRENT_DATE repräsentiert. **(6 Punkte)**

```
CREATE VIEW Mietangebote AS (
        SELECT Wohneinheit.*, Immobilie.*
          FROM Wohneinheit NATURAL JOIN Immobilie
        WHERE WeNr NOT IN (SELECT WeNr
                             FROM Mietvertrag
                           )
            AND NOT WeNr IN (SELECT WeNr
                               FROM Mietvertrag
                               WHERE Vertragsende IS NULL
                                     OR Vertragsende >= CURRENT_DATE)
);
```

2. Formulieren Sie, basierend auf dem oben gezeigten DB-Schema, die folgenden DRL- bzw. DML-Statements in SQL.

a) Geben Sie den Stadtteil an, in dem die Immobiliengesellschaft die meisten Wohneinheiten hat. **(3 Punkte)**

```
SELECT DISTINCT Stadtteil
  FROM Immobilie NATURAL JOIN Wohneinheit
GROUP BY Stadtteil
  HAVING COUNT(WeNr) >= ALL( SELECT COUNT(WeNr)
                               FROM Immobilie NATURAL JOIN Wohneinheit
                              GROUP BY Stadtteil
                             );
```

b) Geben Sie alle Immobilien an, in denen die Schadensbezeichnung „Glasbruch-Haustür“ aufgetreten ist und bei der der Schaden **nur** zu Lasten der Immobilie verbucht wurde. **(4 Punkte)**

```
SELECT DISTINCT Immobilie.*
  FROM Immobilie NATURAL JOIN Schaden
WHERE Schadensbezeichnung = 'Glasbruch-Haustür'
    AND WeNr IS NULL;
```

c) Erhöhen Sie die Bewohneranzahl des aktuellen Mietvertrages für die Wohneinheit mit der Lagebezeichnung „EG links“ in der „Hauptstrasse“ „3B“ (PLZ "45127") um eins, da die dort wohnhafte Familie ein weiteres Kind bekommen hat. **(4 Punkte)**

```
UPDATE Mietvertrag
    SET  Bewohneranzahl = Bewohneranzahl + 1
 WHERE WeNr IN ( SELECT WeNr
                   FROM Wohneinheit NATURAL JOIN Immobilie
                  WHERE Lagebezeichnung = 'EG links'
                    AND Straße = 'Haupstraße'
                    AND HausNr = '38'
                    AND PLZ = '45127'
                 )
AND Vertragsende IS NULL OR Vertragsende >= CURRENT_DATE;
```

d) Finden Sie die Mieter, die „Erstbezieher“ einer Wohnung sind (d. h. als erster Mieter einer Wohneinheit registriert sind) und vorher noch keine Wohnung der Immobiliengesellschaft bewohnt haben. **(5 Punkte)**

```
SELECT Mieter.*
  FROM Mieter NATURAL JOIN Mietvertrag
WHERE MieterNr IN ( SELECT MieterNr
                      FROM Mietvertrag
                    GROUP BY MieterNr
                      HAVING COUNT(MieterNr) = 1
                   )
   AND WeNr IN ( SELECT We_Nr
                   FROM Mietvertrag
                 GROUP BY WeNr
                   HAVING COUNT(WeNr) = 1
                );
```

e) Ermitteln Sie (ausgehend von einer erstellten und als korrekt angenommen Sicht „Mietangebote“ aus der Aufgabe 2.1b) eine Liste von freien Wohneinheiten, die zwischen 3 und 5 Räume besitzen und eine Größe zwischen 40 bis 70 m² haben.
Darüber hinaus sollen aber alle freien Wohnungen eliminiert werden, die speziell im Stadtteil „Vorort“ liegen und bisher noch nie bezogen wurden. **(6 Punkte)**

```
SELECT Mietangebote.*
  FROM Mietangebote
WHERE Raumanzahl BETWEEN 3 AND 5
   AND qm_Gesamt BETWEEN 40 AND 70
EXCEPT
        SELECT Mietangebote.*
          FROM Mietangebote
        WHERE Stadtteil = 'Vorort'
            AND WeNr IN (SELECT Wohneinheit.WeNr
                           FROM Wohneinheit LEFT OUTER JOIN Mietvertrag
                         WHERE Mietvertrag.WeNr IS NULL
                         );
```

f) Ermitteln Sie die Wohnungen, deren Quadratmeterpreis (€/m²) über dem durchschnittlichen Quadratmeterpreis aller Wohnungen in dem entsprechenden Stadtteil liegt. **(8 Punkte)**

```
SELECT Wohneinheit.*
  FROM Wohneinheit AS Wo1 NATURAL JOIN Immobilie AS Im1
WHERE (Mietpreis / qm_Gesamt) >= (
                                SELECT AVG(mietpreis/qm_Gesamt)
                                  FROM Wohneinheit AS Wo2 NATURAL JOIN Immobilie AS Im2
                                WHERE Im2.Stadtteil = Im1.Stadtteil
                                        );
```

Aufgabe 3: NULL-Marken (5 Punkte)

Gegeben seien die folgenden Datensätze in den Tabellen Mitarbeiter und Lehrveranstaltung:

Mitarbeiter	personID	gehalt	raum
	123	2000	SH420
	124	100	NULL
	125	2000	SH430
	126	4000	SH520
	127	NULL	SH530

Lehrveranstaltung	lv_nr	raum	status
	53004	SH601	'GS'
	53005	NULL	'HS'
	53006	SH420	'GS'
	53007	SE008	'GS'

a) Was ist das Ergebnis der folgenden Abfrage, die jene Mitarbeiter ausgeben soll, in deren Büro – fälschlicherweise – eine Lehrveranstaltung (des Hauptstudiums) stattfindet?

SELECT personID FROM Mitarbeiter WHERE raum =ANY (SELECT raum FROM Lehrveranstaltung WHERE status='HS');

Tragen Sie die alle zurückgegebenen Werte in die folgende Tabelle ein (beachten Sie: die Anzahl der Zeilen gibt keinen Hinweis auf die Anzahl der zurückgegebenen Werte!). Sollte eine leere Menge zurückgegeben werden, streichen Sie die Tabelle durch und schreiben ein "∅" rechts in den Freiraum neben der Tabelle.

Ergebnis	personID

leere Menge! ∅

Die Abfrage liefert die leere Menge zurück, weil die vom zweiten Operanden des xANY-Prädikats zurückgelieferte Menge nur NULL-Marken enthält und es folglich stets zu unknown evaluiert.

Aufgabe 4: Abfragetransformation (12 Punkte)

Gegeben seien folgende Relationen:

Person(personID, vorname, nachname, geburtsdatum)
Student (personID, matrikelnummer, fachbereich, studienfach, fachsemester)
Mitarbeiter (personID, wochenstunden, gehalt, raum)
Lehrveranstaltung(lv_nr, raum, gehalten_seit, status)
Leistung (personID, lv_nr, note)
Betreuung (personID, lv_nr)

Zeigen Sie mit Hilfe von Äquivalenzumformungen (die Transformationsregeln finden Sie am Ende der Klausur), dass die folgenden SQL-Ausdrücke äquivalent sind. Geben Sie dabei an, welche Transformationsregeln Sie in welcher Reihenfolge verwenden!

a) $\pi_{\text{nachname}}\, \sigma_{\text{nachname='Octopus'}}$
$(\pi_{\text{nachname,vorname,geburtsdatum}}\, (\text{Student} \bowtie (\sigma_{\text{vorname='Paul'}}\, \text{Person})))$

b) $\pi_{\text{nachname}}\, \sigma_{\text{nachname='Octopus'} \wedge \text{vorname='Paul'}}\, (\text{Person} \bowtie \text{Student})$

Lösung:

von a nach b:

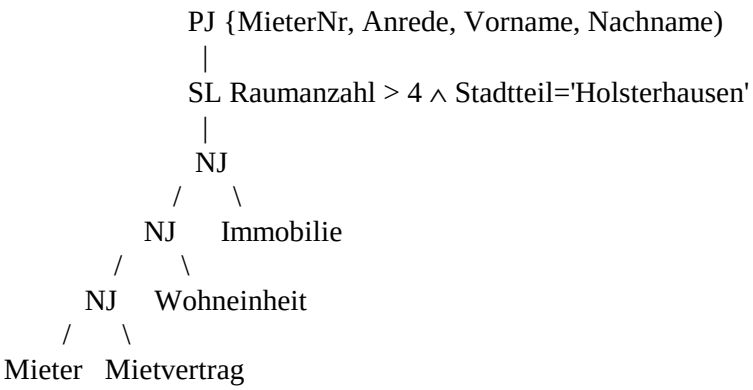
1. R2: $\pi_{\text{nachname}}\, \pi_{\text{nachname,vorname,geburtsdatum}}$
$(\sigma_{\text{nachname='Octopus'}}\, (\text{Student} \bowtie (\sigma_{\text{vorname='Paul'}}\, \text{Person})))$
2. R9: $\pi_{\text{nachname}}\, \sigma_{\text{nachname='Octopus'}}\, (\text{Student} \bowtie (\sigma_{\text{vorname='Paul'}}\, \text{Person}))$
3. R14: $\pi_{\text{nachname}}\, \sigma_{\text{nachname='Octopus'}}\, \sigma_{\text{vorname='Paul'}}\, (\text{Student} \bowtie \text{Person}))$
4. R10: $\pi_{\text{nachname}}\, \sigma_{\text{nachname='Octopus'} \wedge \text{vorname='Paul'}}\, (\text{Student} \bowtie \text{Person}))$
5. R5: $\pi_{\text{nachname}}\, \sigma_{\text{nachname='Octopus'} \wedge \text{vorname='Paul'}}\, (\text{Person} \bowtie \text{Student}))$

Aufgabe 5: Anfrageoptimierung (5 Punkte)

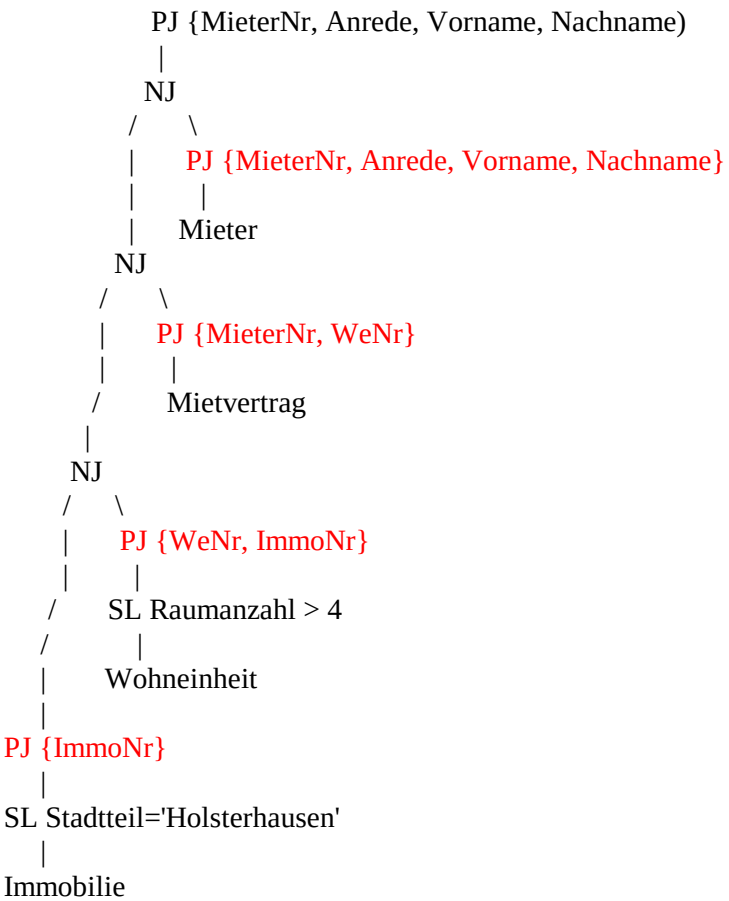
Erstellen Sie den Operatorbaum der folgenden Abfrage und optimieren Sie ihn.
(Geben Sie nur das Endergebnis Ihrer Optimierung an. Die bei der Optimierung durchgeführten Äquivalenzumformungen müssen nicht einzeln dokumentiert werden.)

$\pi_{\text{MieterNr, Anrede, Vorname, Nachname}}\ \sigma_{\text{Raumanzahl}>4 \wedge \text{Stadtteil='Holsterhausen'}}$
(Mieter $\bowtie$ Mietvertrag $\bowtie$ Wohneinheit $\bowtie$ Immobilie)

a) Operatorbaum:



b) Optimierter Operatorbaum:

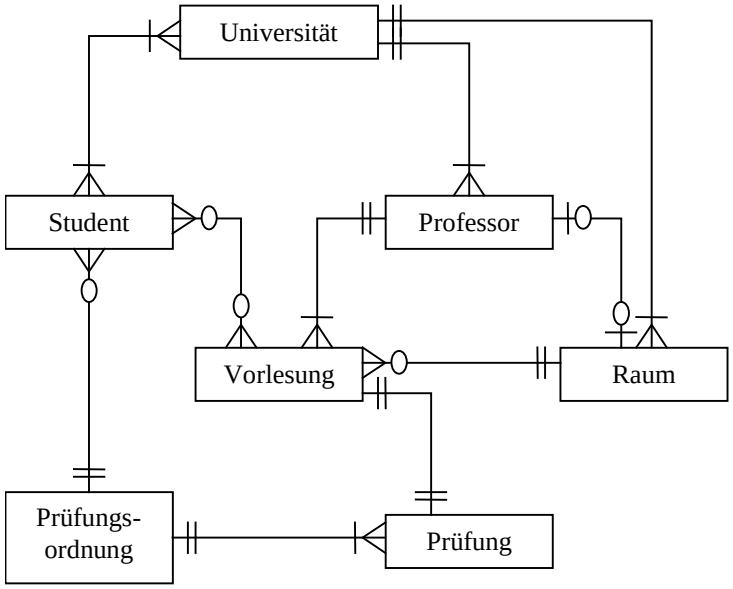


Aufgabe 6: Entity-Relationship-Modellierung (14 Punkte)

Bitte lesen Sie sich die folgende Aufgabenstellung zunächst ***vollständig*** durch, bevor Sie mit der Lösung beginnen!

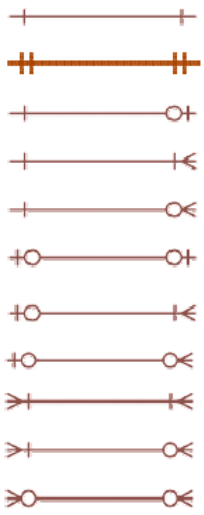
Modellieren Sie den folgenden Sachverhalt in einem ER-Diagramm. Verwenden Sie die Krähenfußnotation (siehe nächste Seite).

Eine Universität hat viele Studenten und viele Professoren und viele Räume. Studenten müssen an mindestens einer Universität eingeschrieben sein. Professoren lehren stets an genau einer Universität. Eine Universität ohne Professoren und/oder ohne Studenten und/oder ohne Räume gibt es nicht. Professoren bieten Vorlesungen an, an denen mehrere Studenten teilnehmen können. Professoren müssen mindestens eine Vorlesung anbieten. Studenten steht es frei, an Vorlesungen teilzunehmen oder nicht. Jede Vorlesung wird von genau einem Professor angeboten und findet auch dann statt, wenn kein Student an ihr teilnimmt. Jede Vorlesung findet stets in genau einem Raum statt. In einem Raum können zu unterschiedlichen Zeiten unterschiedliche Vorlesungen stattfinden, jedoch müssen nicht in allen Räumen Vorlesungen stattfinden. Denn einige Räume stehen den Professoren als Arbeitsraum zur Verfügung. Dabei ist jedem Professor maximal ein Arbeitsraum zugeordnet – und umgekehrt. Zu jeder Vorlesung gibt es genau eine Prüfung, in der ausschließlich die Inhalte der betreffenden Vorlesung abgefragt werden. Die Prüfung ist Teil genau einer Prüfungsordnung, die aus mehreren Prüfungen besteht. Eine Prüfungsordnung ohne Prüfungen gibt es nicht. Jeder Student studiert nach genau einer Prüfungsordnung, wobei nicht zu jeder Prüfungsordnung ein Student eingeschrieben sein muss.



Martin notation (crow's foot notation) by James Martin, Charles Bachmann and James Odell

Legend



1 - 1 (mandatory relationship – but exactly one)

- alternative presentation for 1 (double)

1 - c (either 0 or 1 – optional relationship)

1 – n (mandatory relationship – arbitrary many)

1 - cn (optional relationship – arbitrary many)

c - c (optional relationship – both sides)

c - m

c - cm

n - m

n - cn

cn - cm

Aufgabe 7: Synchronisationsverfahren (10 Punkte)

Gegeben sei die folgende Schedule:

R3(f) R4(e) W3(e) W4(h) R5(e) R1(g) R5(g) R2(f) R2(h) W1(g) W5(f) W1(e) W2(h)

Welche (um die Sperren erweiterte) Ergebnisschedule würde von dem Zweiphasensperrprotokoll mit **Preclaiming** erzeugt?

Tragen Sie die vollständige Schedule – inklusive der Sperren – in die folgende Tabelle ein! Beachten Sie dabei bitte, dass die Anzahl der Zeilen in der folgenden Tabelle ***keinen*** *Hinweis auf die richtige Lösung gibt.*

	e	f	g	h	1	2	3	4
pro Transaktion benötigte Sperren:								
T1 T2	x	 s	x	 x				
T3 T4	x s	s		 x				
T5	s	x	s					
los gehts...								
R3(f)	x3	s3R3u3						
R4(e)	×				R4(e)			
W3(e)	W3u3				R4(e)			
>R4(e)	s4R4u4			x4				
W4(h)				W4u4				
R5(e)	s5R5u5	x5	s5					
R1(g)			×		R1(g)			
R5(g)			R5u5		R1(g)			
>R1(g)	x1		x1R1					
R2(f)		×			R2(f)			

	e	f	g	h	1	2	3	4
R2(h)		×			R2(f)	R2(h)		
W1(g)			W1u1		R2(f)	R2(h)		
>R2(f)		×			R2(h)	R2(f)		
>R2(h)		×			R2(f)	R2(h)		
W5(f)		W5u5			R2(f)	R2(h)		
>R2(f)		s2R2u2		x2	R2(h)			
>R2(h)				R2				
W1(e)	W1u1							
W2(h)				W2u2				

Zusatzaufgabe (8 Punkte)

1. Der IBM InfoSphere Data Architect ist ein Werkzeug für Datendesign, -modellierung, integration und -standardisierung. Hierbei unterstützt er die Arbeit am **logischen Datenmodel**, am **physikalischen Datenmodell** und am **Domänenmodell**.

   a) Beschreiben Sie in kurzen Worten die entsprechenden Modellarten. **(4 Punkte)**

   Logisches Datenmodell:

   Setzt die Abstraktionsregeln fest und überführt die Realwelt in einen ersten Datenentwurf.

   Physikalisches Datenmodell:

   Legt die internen Speicherungsstrukturen und Zugriffspfade fest.

   Domänenmodell:

   Legt die Wertebereiche der Entitäten und Attribute fest.

2. In realen Datenbanksystemen werden häufig Entwicklungs- und Produktivsysteme verwendet. Nachdem ein Datenbanksystem in der Entwicklungsumgebung erweitert wurde, sollen nun durch den DB-Administrator alle Änderungen in das Produktivsystem übernommen werden.

   a) Welche Aufgaben hat der DB-Administrator zu erledigen, um beide Systeme auf den gleichen Stand zu bringen? Wählen Sie die benötigten Arbeitsschritte aus der folgenden Liste aus und bringen Sie diese in die richtige Reihenfolge. **(2 Punkte)**

   Schreiben Sie vor jeden Eintrag in der Liste unten die entsprechende Reihenfolgenummer. Sollte ein Arbeitsschritt nicht notwendig sein, so streichen Sie ihn bitte durch.

   /… Daten in das Testsystem importieren

   /… Vergleich zwischen den logischen Datenbankmodellen

   /… COMMIT im Testsystem absetzen

   1… Vergleich zwischen den physikalischen Datenbankmodellen

   2… Strukturänderungen in einem Datenbank - Skript zusammenfassen

   /… Datenbank-Skript im Testsystem ausführen

   4… COMMIT im Produktivsystem absetzen

   3… Datenbank-Skript im Produktivsystem ausführen

   b) Warum ist die Trennung zwischen Entwicklungs- und Produktivsystemen sinnvoll? **(2 Punkte)**

   Da auf diese Weise die Operationsfähigkeit des Produktivsystems nicht unmittelbar durch eventuelle Entwicklungsfehler gefährdet wird.

Anhang: Transformationsregeln

Rainer Unland
ICB
University of
Duisburg-Essen

22
2-Jun-10

2.2 Query Optimization

Logically equivalent transformations (part I)

Logically equivalent transformations of relational algebra expressions		
No	**Rule**	**side condition**
1	$SL_{F1}\ (SL_{F2}\ R) \equiv SL_{F2}\ (SL_{F1}\ R)$	—
2	$SL_F\ (PJ_A\ R) \equiv PJ_A\ (SL_F\ R)$	$\leftarrow$: $Attr(F) \subseteq A$
3	R **UN** S ≡ S **UN** R	—
4	R **CP** S ≡ S **CP** R	—
5	$R\ JN_F\ S \equiv S\ JN_F\ R$	—
6	(R **UN** S) **UN** T ≡ R **UN** (S **UN** T)	—
7	(R **CP** S) **CP** T ≡ R **CP** (S **CP** T)	—
8	$(R\ JN_{F1}\ S)\ JN_{F2}\ T \equiv R\ JN_{F1}\ (S\ JN_{F2}\ T)$	$\rightarrow$: $Attr(F2) \subseteq (Attr(S) \cup Attr(T))$ $\leftarrow$: $Attr(F1) \subseteq (Attr(R) \cup Attr(S))$
9	$PJ_{A1}\ R \equiv PJ_{A1}\ PJ_{A2}\ R$	$A1 \subseteq A2 \subseteq Attr(R)$
10	$SL_F\ R \equiv SL_{F1}SL_{F2}R$	$F = F1 \wedge F2$
11	$SL_F\ (R\ UN\ S) \equiv (SL_F\ R)\ UN\ (SL_F\ S)$	—
12	$SL_F\ (R\ DF\ S) \equiv (SL_F\ R)\ DF\ (SL_F\ S)$	—
13	$SL_F(R\ CP\ S) \equiv (SL_{F1}\ R)\ CP\ (SL_{F2}\ S)$	$\rightarrow$: $(F = F1 \wedge F2) \wedge Attr(F1) \subseteq Attr(R)) \wedge (Attr(F2) \subseteq Attr(S))$ $\leftarrow$: $F = F1 \wedge F2$

Rainer Unland
ICB
University of
Duisburg-Essen

23
2-Jun-10

2.2 Query Optimization

Logically equivalent transformations (part II)

14	$SL_F(R\ JN_{F3}\ S) \equiv (SL_{F1}R)\ JN_{F3}\ (SL_{F2}\ S)$	$\rightarrow$: $(F = F1 \wedge F2) \wedge Attr(F1) \subseteq Attr(R)) \wedge (Attr(F2) \subseteq Attr(S))$ $\leftarrow$: $F = F1 \wedge F2$
15	$PJ_A\ (R\ UN\ S) \equiv (PJ_A\ R)\ UN\ (PJ_A\ S)$	—
16	$PJ_A\ (R\ CP\ S) \equiv (PJ_{A1}\ R)\ CP\ (PJ_{A2}\ S)$	$\rightarrow$: $(A1 = A \cap Attr(R)) \wedge (A2 = A \cap Attr(S))$ $\leftarrow$: $A = A1 \cup A2$
17	$PJ_A(R\ JN_F\ S) \equiv (PJ_{A1}\ R)\ JN_F\ (PJ_{A2}\ S)$	$\rightarrow$: $(Attr(F) \subseteq A) \wedge (A1 = A \setminus Attr(S)) \wedge (A2 = A \setminus Attr(R))$ $\leftarrow$: $A = A1 \cup A2$
18	R **NJN** R ≡ R	—
19	R **UN** R ≡ R	—
20	R **DF** R ≡ 0,/	—
21	$R\ NJN\ SL_F\ R \equiv SL_F\ R$	—
22	$R\ UN\ SL_F\ R \equiv R$	—
23	$R\ DF\ SL_F\ R \equiv SL_{\neg F}\ R$	—
24	$(SL_{F1}\ R)\ NJN\ (SL_{F2}\ R) \equiv SL_{F1 \wedge F2}\ R$	—
25	$(SL_{F1}\ R)\ UN\ (SL_{F2}\ R) \equiv SL_{F1 \vee F2}\ R$	—
26	$(SL_{F1}\ R)\ DF\ (SL_{F2}\ R) \equiv SL_{F1 \wedge \neg F2}\ R$	—