



Universität Heidelberg
Institut für Informatik
Lehrstuhl für Datenbanksysteme
Prof. Dr. Michael Gertz

18.7.2011
Datenbanken I
Sommersemester 2011
Klausur

Klausur zur Vorlesung Datenbanken I im SS 2011

Name, Vorname: _____

Matrikelnummer: _____

Studiengang: _____

| Aufgabe | Max. Punkte | Punkte |
|----------|-------------|--------|
| 1 | 5 | |
| 2 | 19 | |
| 3 | 8 | |
| 4 | 12 | |
| 5 | 14 | |
| 6 | 8 | |
| Σ | 66 | |

Hinweise:

- Bearbeitungsdauer: 90 Minuten
- Sie können die Folienkopien, das Textbuch sowie Ihre Unterlagen zur Vorlesung verwenden. Notebooks, PDAs und andere elektronische Geräte dürfen nicht verwendet werden.
- Schalten Sie für die Dauer der Klausur Ihr Mobiltelefon aus.
- Die Klausur besteht aus 6 Aufgaben auf 8 Seiten. Bitte überprüfen Sie die Vollständigkeit.
- Bitte benutzen Sie den zu jeder Aufgabe vorgegebenen Platz; dieser sollte zur Lösung der Aufgabe ausreichen. Wenn Sie noch die Rückseite verwenden, markieren Sie bei der Aufgabe klar, dass es eine Fortsetzung auf der Rückseite gibt.
- Tauschen Sie auf keinen Fall irgendwelche Notizen etc. mit den Nachbarn aus.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist.
- Schreiben Sie deutlich.
- Benutzen Sie keinen Bleistift.



Beispieldatenbank:

Eine kleine Videothek in Wiesloch hat sich entschieden, ihren Kunden eine Datenbank zur Suche von Filmen (auf DVDs) und Darstellern anzubieten. Hierzu wird das folgende Datenbankschema verwendet:

DVD(did, titel, jahr, genre, preis)

Sequel(fortid, vorid)

Actor(vname, nname, alter)

Cast(did, vname, nname, rolle)

In der Relation DVD werden Informationen zu den verschiedenen DVDs verwaltet. Eine DVD ist eindeutig über eine Nummer (did) identifiziert und hat einen Filmtitel (titel) und ein Erscheinungsjahr (jahr). Zu jeder DVD werden noch das Genre (genre, z.B. 'Sci-Fi', 'Dokumentation', 'Krimi', 'Action' etc.) sowie der Preis der DVD (preis) gespeichert.

Informationen über Fortsetzungen von Filmen werden in der Relation Sequel gespeichert. Z.B. ist die DVD mit dem Titel 'Rambo II' eine Fortsetzung der DVD mit dem Titel 'Rambo I'. Beachten Sie, dass hier nur die IDs der entsprechenden DVDs gespeichert werden und eine Fortsetzung (fortid) genau einen Vorgänger (vorid) hat.

Die Relation Actor verwaltet Informationen zu Darstellern. Ein Darsteller ist eindeutig über den Vor- und Nachnamen (vname, nname) identifiziert und hat ein Alter (alter). Ein Darsteller kann in einem Film eine oder mehrere Rollen spielen (z.B. 'Diener', 'Postbote1', 'Postbote2'). Die entsprechenden Informationen hierzu werden in der Relation Cast verwaltet.

Machen Sie sich klar, welche Attribute in den obigen Relationen Fremdschlüssel sind !

Aufgabe 1: SQL-DDL Befehle

(5 Punkte)

- Angenommen, die Relationen DVD und Actor sind schon angelegt worden. Geben Sie den `create table` Befehl für die Relation Cast an, wiederum inklusive Primär- und Fremdschlüssel sowie eventueller einfacher `check`-Bedingungen.

```
create table Cast(
```



- b. Geben Sie den Preis der DVD aus, deren Film der direkte Nachfolger des Films mit dem Titel 'Toy Story' ist.



- c. Geben Sie für jede DVD den Filmtitel und die Anzahl der Darsteller in dem Film sowie das Alter des jüngsten Darstellers in dem Film aus. Die eventuell verschiedenen Rollen eines Darstellers in einem Film sind hierbei nicht relevant.
- d. Geben Sie den Titel und den Preis der ältesten DVD aus (basierend auf dem Attribut `jahr`), zu deren Film es keine Fortsetzung gibt.
- e. Geben Sie den Vor- und Nachnamen des Darstellers aus, der die meisten Rollen in einem Film gespielt hat.



Aufgabe 3: Algebra und Kalküle

(3+3+2=8 Punkte)

Gegeben sei die folgende umgangssprachliche Anfrage an die DVD-Datenbank: *Gesucht sind die IDs von DVDs des Genre 'Sci-Fi', in denen nur ein Darsteller spielt.*

a. Formulieren Sie die obige Anfrage als einen Ausdruck der Relationenalgebra.

b. Formulieren Sie die obige Anfrage als einen Ausdruck des Tupelkalküls.

c. Formulieren Sie die obige Anfrage als einen Ausdruck des Bereichskalküls.

Aufgabe 4: Anfragebearbeitung und -optimierung

(3+3+5+1=12 Punkte)

Gegeben sei die folgende Anfrage an die DVD-Datenbank als Ausdruck der Relationenalgebra:

$\pi_{\text{Cast.vname}, \text{Cast.nname}}(\sigma_{\text{Rolle}='Student' \wedge \text{Alter} > 50 \wedge \text{Cast.vname} = \text{Actor.vname} \wedge \text{Cast.nname} = \text{Actor.nname}}(\text{Cast} \times \text{Actor}))$

a. Geben Sie zu der obigen Anfrage den Operatorbaum an, den Sie nach Anwendung algebraischer Optimierungsregeln erhalten.



- b. Angenommen, die Relation `Actor` belegt mit 1000 Tupeln 200 Blöcke und die Relation `Cast` mit 5000 Tupeln 400 Blöcke. Keine der Relationen ist sortiert und es existiert auch kein Index. Im Hauptspeicher stehen pro Relation 2 Blöcke sowie ein Block für die Ausgabe zur Verfügung. Geben Sie die Berechnung der minimalen Anzahl an Blöcken an, die bei einem Block-Nested-Loops-Verbund gelesen und geschrieben werden müssen.
- c. Angenommen, es existiert ein B^+ -Baum Index $I(\text{Cast}(\text{Rolle}))$. Weiterhin sei angenommen, dass es zu dem Attribut `Rolle` 100 verschiedene Werte gibt, die gleichverteilt sind. Geben Sie einen Ausführungsplan zu der obigen Anfrage unter Verwendung dieses Indexes an und erläutern Sie kurz anhand dieses Planes die Bearbeitung der Anfrage.
- d. Wie müsste sich die Anzahl verschiedener Werte für das Attribut `Rolle` ändern, damit ein Full-Table Scan effizienter als ein Indexzugriff auf $I(\text{Cast}(\text{Rolle}))$ ist?

Aufgabe 5: ER-Modellierung

(10+4=14 Punkte)

Eine Wohnungsgenossenschaft möchte ihre Anliegen mit Hilfe einer Datenbank verwalten. Sie sollen dazu basierend auf den nachfolgenden Anforderungen ein entsprechendes ER-Schema entwickeln.

- Jedes im Besitz der Genossenschaft befindliche Gebäude verfügt über eine Adresse bestehend aus Strassenname, Hausnummer, Ort und Postleitzahl. Zusätzlich wird die Anzahl der Wohnungen im Haus gespeichert. Beispiele:
 - Das Gebäude in der Hauptstrasse 25 in 69115 Heidelberg hat 5 Wohnungen.
 - Das Gebäude am Mombertplatz 13 in 69126 Heidelberg hat 70 Wohnungen.
- Zu jeder Wohnung werden die Anzahl der Zimmer, die Quadratmeter und die monatliche Kaltmiete verwaltet. Beispiele:
 - Die Wohnung mit der Nummer 3 in der Hauptstrasse 25 in 69115 Heidelberg ist eine 3-Zimmer-Wohnung mit 60 m^2 für 700 Euro monatlich.



- Die Wohnung mit der Nummer 3 am Mombertplatz 13 in 69126 Heidelberg ist eine 4-Zimmer-Wohnung mit 55 m^2 für 480 Euro monatlich.
- Für jeden Mietvertrag ist gespeichert, wer welche Wohnung (und seit wann) gemietet hat. Wohnungen werden immer nur an einen Hauptmieter vermietet. Außerdem wird vermerkt, ob und zu welchen Zeitpunkt ein Mietverhältnis gekündigt wurde. Beispiele:
 - Egon Eisenbeis wohnt seit 04.07.2003 in der Wohnung Nummer 3 in der Hauptstrasse 25 in 69126 Heidelberg. Das Mietverhältnis ist nicht gekündigt.
 - Dagmar Ramgad wohnte seit 15.07.1969 in der Wohnung Nummer 50 am Mombertplatz 13 in 69126 Heidelberg. Das Mietverhältnis wurde zum 20.01.2010 aufgelöst.
- Mieter sind juristische Personen, die mit ihrem Namen und einer Telefonnummer vermerkt werden. Beispiel:
 - Mieter Egon Eisenbeis erreichbar unter 0315/328315

Basierend auf den obigen Informationen, führen Sie die folgenden Tätigkeiten aus:

- a. Geben Sie ein ER-Schema an, das die obigen Informationen entsprechend beschreibt. Spezifizieren Sie geeignete Primärschlüsselattribute sowie Kardinalitäten zu Beziehungstypen.



- b. Geben Sie umgangssprachlich zwei wichtige und notwendige Integritätsbedingungen an, die Sie nicht in Ihrem ER-Schema bzw. im ER-Modell ausdrücken können.

Aufgabe 6: Serialisierbarkeit

(2+6=8 Punkte)

- a. Geben Sie einen minimalen verzahnten Schedule für zwei Transaktionen T_1 und T_2 an, der nicht serialisierbar ist. (minimal heißt, dass der Schedule eine minimale Gesamtzahl von Lese/Schreiboperationen in T_1 und T_2 hat).

- b. Angenommen seien die beiden folgenden Schedules mit je drei verzahnten Transaktionen:

| T_1 | T_2 | T_3 |
|----------|----------|----------|
| | read(A) | read(A) |
| read(A) | | write(A) |
| write(A) | write(B) | |

| T_1 | T_2 | T_3 |
|----------|----------|----------|
| read(A) | | read(A) |
| | read(A) | write(A) |
| write(A) | write(A) | |

Geben Sie den Konfliktgraph für jeden der beiden Schedules an. Welche der Schedules ist konfliktserialisierbar? Wie würde die serielle Ausführung des seriellen Schedules aussehen?

Konfliktgraphen:



Universität Heidelberg
Institut für Informatik
Lehrstuhl für Datenbanksysteme
Prof. Dr. Michael Gertz

26.07.2012
Datenbanken I
Sommersemester 2012
Klausur

Vorlesung Datenbanken I im SoSe 2012

Klausur

Name, Vorname: _____

Matrikelnummer: _____

Studiengang: _____

| Aufgabe | Max. Punkte | Punkte |
|----------|-------------|--------|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| Σ | Sum | |

Hinweise:

- Bearbeitungsdauer: 90 Minuten
- Sie können die Folienkopien, das Textbuch sowie Ihre Unterlagen zur Vorlesung verwenden. Notebooks, PDAs und andere elektronische Geräte dürfen nicht verwendet werden.
- Schalten Sie für die Dauer der Klausur Ihr Mobiltelefon aus.
- Die Klausur besteht aus X Aufgaben auf Y Seiten. Bitte überprüfen Sie die Vollständigkeit.
- Bitte benutzen Sie den zu jeder Aufgabe vorgegebenen Platz; dieser sollte zur Lösung der Aufgabe ausreichen. Wenn Sie noch die Rückseite verwenden, markieren Sie bei der Aufgabe klar, dass es eine Fortsetzung auf der Rückseite gibt.
- Tauschen Sie auf keinen Fall irgendwelche Notizen etc. mit den Nachbarn aus.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist!
- Benutzen Sie keinen Bleistift.



Beispieldatenbank:

Zum Übungsbetrieb der Vorlesung Datenbanken 1 wird am Lehrstuhl eine Datenbank erstellt. In dieser Datenbank sollen im Wesentlichen die erzielten Punkte der Studenten gespeichert werden. Hierzu wird das folgende Datenbankschema verwendet:

```
Student (matnmr, nname, vname, fach, punkte)
Übungsgruppe (gid, tag, uhrzeit, tutor, limit)
Übungszettel (nummer, semester, aufgaben, punkte)
Abgabe (matnmr, gid, nummer, semester, ergebnis)
```

In der Relation Student werden die relevanten Informationen zu den teilnehmenden Studenten gespeichert. Ein Student lässt sich eindeutig durch seine Matrikelnummer (matnmr) identifizieren. Damit später die Scheine ausgestellt werden können, werden zusätzlich noch Studienfach (fach) sowie der volle Name des Studenten mitgespeichert.

Für den Übungsbetrieb ist es wichtig, dass die Studenten den Übungsgruppen zugeordnet werden. Hierzu wird die Relation Übungsgruppe angelegt. Jede Gruppe erhält eine gid, es werden des Weiteren auch der Termin der Übung durch tag und uhrzeit sowie die maximale Gruppengröße festgelegt.

Die Relation Übungszettel verwaltet die Informationen zu den Übungszetteln. Die Zettel werden durch ihre nummer im semester identifiziert. Zusätzlich werden die zu lösenden Aufgaben (aufgaben) und die maximal erreichbaren Punkte (punkte) mitgespeichert (es gibt in diesem Modell keine Extrapunkte).

Die Relation Abgabe verknüpft die Daten und speichert die erzielten Punkte des Studenten auf dem jeweiligen Übungszettel.

Machen Sie sich klar, welche Attribute in den obigen Relationen Fremdschlüssel sind !

Aufgabe 1: SQL-DDL Befehle

(5 Punkte)

Angenommen alle Relationen bis auf Abgabe sind schon angelegt worden. Geben Sie den `create table` Befehl für die Relation Abgabe an. Formulieren Sie hier auch Primär- und Fremdschlüssel sowie eventueller einfacher `check`-Bedingungen.

```
create table Abgabe (
```

Aufgabe 2: SQL-Anfragen

(3+3+4+5+4=19 Punkte)

Formulieren Sie die folgenden Anfragen an die obige Datenbank in SQL ohne dabei eine `limit`-Klausel zu benutzen.

- Geben Sie die Matrikelnummer und Nachnamen der Studenten an, die für die Klausur zugelassen werden; d.h. mehr als 50% der möglichen Punkte aus dem Übungsbetrieb haben.
- Zur Bereinigung der Datenbank sollen alle Studenten entfernt werden, die keine Abgabe ein-



gereicht haben.

- c. Für das nächste Semester werden Tutoren gesucht. Geben Sie den vollständigen Namen der beiden Studenten mit den meisten Punkten an.
- d. Geben Sie die Nummer des Übungszettels an, auf dem die Studenten (absolut) die wenigsten Punkte erzielt haben.
- e. Geben Sie die Übungsgruppe sowie deren Teilnehmerzahl an.
- f. Geben Sie ein update-Statement an, damit nach Abgabe eines weiteren Übungszettels, die Punktezahl des Studenten aktualisiert wird.

Aufgabe 2: Integritätsbedingungen

(3+2=5 Punkte)

- a. Welche Attribute in den obigen Relationen sind Fremdschlüsselattribute und welche Relationen referenzieren diese Attribute? Sie können hierzu folgende Notation verwenden, um zu beschreiben, dass ein Fremdschlüssel der Relation 1 auf den Primärschlüssel einer Relation 2 verweist: $\langle \text{Relation1} \rangle . \langle \text{Attribut(e)} \rangle \longrightarrow \langle \text{Relation2} \rangle$,
- b. Gegeben sei folgende Integritätsbedingung: *Die Gesamtzahl der LKWs einer Spedition muss mindestens gleich der Summe der LKWs an den Zweigstellen der Spedition sein.* Geben Sie alle Operationen an, die diese Bedingung verletzen können.

Aufgabe 3: SQL-Anfragen

(2+4+3+4+5+3+3=24 Punkte)

Formulieren Sie folgende Anfragen an die Beispieldatenbank in SQL:

- a. Geben Sie die Postleitzahl all der Zweigstellen an, die mindestens 50% aller LKWs einer Spedition haben.
- b. Welche Speditionen haben ihre Zweigstellen nur in einer Stadt (aber eventuell unter verschiedenen Postleitzahlen)? Formulieren Sie diese Anfrage ohne Aggregationen oder Gruppierung. Ausgegeben sind Name der Spedition und Name der Stadt.
- c. Wie viele Zweigstellen hat die Spedition mit den meisten LKWs? Ausgegeben sind der Name der Spedition und die Anzahl der Zweigstellen.
- d. Welche Speditionen haben ihren Hauptsitz in "Karlsruhe" und bedienen mehr als 50 verschiedene Routen?
- e. Gibt es Routen, die die gleichen Start- und Endpostleitzahlen aber verschiedene Entfernungen haben? Ausgegeben sind hierzu komplette Tupel der Relation ROUTE.
- f. Geben Sie für jede Route mit mehr als 1000 km die Anzahl der Speditionen an, die diese Route bedienen. Gesucht sind nur Routen, die von mehr als 5 Speditionen bedient werden. Ausgegeben sind neben der Anzahl der Speditionen die Start- und Endpostleitzahl.
- g. Die Spedition "HotWheels" hat zwei neue Zweigstellen in den Städten mit den Postleitzahlen "69120" und "76646" aufgemacht. Insgesamt hat die Spedition 20 neue LKWs, die gleich auf die beiden neuen Zweigstellen verteilt werden. Geben Sie alle Update-Operationen auf entsprechende Relationen an, um diese Änderungen durchzuführen.



Aufgabe 4: Anfragebearbeitung und -optimierung

(1+2+3+6=12 Punkte)

- a. Was wird durch die folgende SQL-Anfrage bestimmt?

```
select S.sname, T.name, R.km
from SPEDITION S, ROUTE R, STADT T
where S.sname = r.sname and
      R.end-plz = 69120 and R.start-plz = T.plz
```

- b. Übersetzen Sie diese Anfrage **direkt** in einen Ausdruck der Relationenalgebra. Verwenden Sie nur die Grundoperationen der Algebra (beachten Sie hierbei, dass z.B. der Join nur eine abgeleitete Operation ist).
- c. Führen Sie eine algebraische Optimierung für den unter b) bestimmten Ausdruck unter Verwendung der Transformationsregeln durch. Es reicht aus, wenn Sie hierzu nur das Endergebnis (Anfragebaum) angeben.
- d. Angenommen für die Primärschlüssel in den obigen Relationen existiert jeweils ein Clustered Index. Es sei weiterhin angenommen, dass ein Unclustered Index für das Attribut end-plz existiert.

Geben Sie einen effizienten Zugriffsplan für die von Ihnen unter c) beschriebene Anfrage an. Welche Algorithmen verwenden Sie für die Joins? Begründen Sie Ihre Wahl. Begründen Sie auch, warum der von Ihnen erstellte Zugriffsplan am effizientesten ist.

Aufgabe 5: Relationale Algebra

(4+2+2=8 Punkte)

Formulieren Sie folgende Anfragen als Ausdrücke der relationalen Algebra. Wenn möglich, wenden Sie die Transformationsregeln an, um eine effiziente Anfrage zu erhalten.

- a. Welche Spedition (sname) aus Mannheim hat die meisten LKWs?
- b. Geben Sie alle Routen an, die von solchen Speditionen bedient werden, die eine Zweigstelle in Sindelfingen haben.

Formulieren Sie die folgende Anfrage umgangssprachlich:

c. $\pi_{\text{sname}}(\sigma_{\text{lkw} > 100 \wedge \text{blkw} > 90}(\text{SPEDITION} \bowtie (\beta_{\text{blkw} \leftarrow \text{lkw}}(\pi_{\text{sname, lkw}}(\text{ZWEIGSTELLE}))))$

Aufgabe 6: Transaktionsverwaltung

(4+3=7 Punkte)

- a. Beschreiben Sie kurz die Verwendung des Redo-Log Buffers während der Ausführung einer Transaktion. Welche Informationen werden im Redo-Log Buffer verwaltet? Geben Sie zwei Szenarien an, in denen der Redo-Log Buffer verwendet wird.
- b. Angenommen sei die folgende Transaktion T . Zu Beginn der Transaktion hat das Objekt A den Wert 100.



| Zeit | Operation |
|------|------------|
| 1 | read(A,a) |
| 3 | a:=a + 100 |
| 5 | write(A,a) |
| 7 | read(B,b) |
| 9 | write(B,b) |
| 11 | commit |

- (i) Angenommen sei eine weitere Transaktion T' , die das Objekt A zum Zeitpunkt 4 lesen will. Welchen Wert für A sollte T' zu lesen bekommen? Woher nimmt das DBMS den entsprechenden Wert für A ?
- (ii) Angenommen, das System stürzt zum Zeitpunkt 9 ab. Welchen Wert hat A nach einem Neustart des Systems? Begründen Sie Ihre Antwort.

