Databases Exam

TDA357 (Chalmers), DIT620 (University of Gothenburg)

17 March 2017 at 8:30–12:30 in HA,HB,HC

Department of Computer Science and Engineering

Examiner: Aarne Ranta tel. 031 772 10 82.

Results: Will be published on 7 April.

Exam review: 18 April at 14-16 in room 6106, EDIT Building.

Grades: Chalmers: 24 for 3, 36 for 4, 48 for 5. GU: 24 for G, 42 for VG.

One English language dictionary is allowed. The exam contains a quick reference card on SQL and other concepts as a "cheat sheet". No cheat sheets of your own are allowed.

Specific instructions: You can answer in English, Danish, Dutch, Finnish, French, German, Italian, Norwegian, Spanish, or Swedish (in this exam; next time it can be another set of languages ;-) Begin the answer to each question (numbers 1 to 6) on a new page. The a,b,c,... parts with the same number can be on the same page. If you need many pages for one question, number the pages as well, for instance, Question 3 p. 2. Write the question number on every page.

Write clearly: unreadable = wrong! Fewer points are given for unnecessarily complicated solutions. Indicate clearly if you make any assumptions that are not given in the question. In particular: in SQL questions, use standard SQL or PostgreSQL. If you use any other variant (such as Oracle or MySQL), say this; but full points are not guaranteed since this may change the nature of the question.

.

1 Modelling (12p)

The domain to model is courses, exams, and exam questions:

- A course has a code and a number of credits. The code identifies the course uniquely.
- A course can have many exams, on different dates. Each exam has a total number of points.
- Each exam has a set of questions, identified by a question number. Each question has a topic and a maximum number of points.
- Each exam is held in one or more rooms.
- Rooms have unique names. Each room has a number of seats.

a. Draw an Entity-Relationship diagram for this domain. **Do not use multi-valued attributes**. (7p).

b. Write a database schema, carefully marking keys and referential constraints. The schema can be a direct translation of your diagram, which is an easy way to get it. However, if you are unsure about your diagram, you can write the schema separately. (5p)

2 Functional dependencies and normal forms (8p)

Consider a relation $R(A, B, C, D)$ with the functional dependencies $A \to B$ and $C \to D$ (but no others, except the derived and trivial ones).

a. Give a real example of attributes A, B, C, D with just these dependencies (no others, except derived or trivial ones). Justify your answer! (2p)

b. Show at least four derived functional dependencies, as well as all possible keys. (2p)

b. Convert the relation to 3NF or BCNF. Specify which of these forms you have chosen, and show the steps deriving the normal form. (4p)

3 SQL queries (12p)

Consider a table with the schema

```
ExamQuestions(courseCode,date,questionNumber,points,topic).
```

a. Show INSERT statements for two questions in today's database exam (i.e. the one that you are just working on). (2p)

b. Write an SQL query that returns the topics of today's database exam. (2p)

c. Write an SQL query that returns the total number of points of today's database exam. (3p)

d. Write an SQL query that shows the topics that appear in exams of at least three different courses. (5p)

4 Relational algebra (8p)

a. Write the query of Question 3d in relational algebra. (4p)

b. Assume a country such that

- there are 10 million persons and 10 thousand places to live
- every person lives in one place and has an age
- every place has a population, and for 90% of places it is less than 1000
- 90% of persons are younger than 75 years

The schemas are `Persons(_id,age,place)` and `Places(_name,population)`, where `Persons.place` refers to `Places.name`. Write an algebraic query that returns the number of people aged 75 years or more and that live in places whose population is less than 1000. Do this in such a way that the size of tables (i.e. number of tuples) considered at any point is as small as possible. What is the maximum size that needs to be considered? (Hint: you can first write a straightforward query and then apply the optimization of pushing conditions to products. But you must show the end result of the optimization.) (4p)

5 Constraints, triggers, and views (12p)

Database integrity can be improved by many techniques:

- views: virtual tables that show useful information that would create redundancy if stored in the actual tables
- constraints: conditions on attribute values and tuples
- triggers: automated checks and actions performed on entire tables

As a general rule, these methods should be applied in the above order: if a view can do the job, constraints are not needed, and if constraints can do the job, triggers are not needed.

The task in this question is to show how to guarantee integrity in an exam question database. The attributes that are needed are the following:

- an `Exam` has a date and a course code, as well as a total number of points
- a `Question` belongs to a certain exam, has its own question number, a number of points, and a text (the question itself)

(The domain is similar to Question 1 but not exactly the same.)

Your task is to give a database definition (tables, triggers, and views) that guarantees the following integrity conditions:

a. A course can have maximally one exam on the same day.

b. The number of points in an exam is the sum of the numbers of points in its questions.

c. An exam question is uniquely determined by its number, together with the exam to which it belongs.

d. An exam may not use the same question text twice.

e. The number of questions in an exam may not exceed 10.

f. The number of questions in an exam must be at least 5. (Hint: can be tricky!)

Write you answer as SQL code for tables, views, and triggers, marking with letters ((a), (b), etc) which part of the code guarantees which property. You can get 2 points for each of the properties. Unnecessarily complicated answers (e.g. using a trigger when a constraint would be enough) may reduce points.

6 Miscellaneous (8p)

a (Transactions, 2p). A common situation in flight booking is the following:

- User A wants to find a flight from X to Y.
- The system shows available flights. One of them has only one seat left.
- Before A makes a choice, user B also wants to find a flight from X to Y.
- The system shows the same list of flights to B.
- B selects immediately the flight that has only one seat left.
- After this, A tries to select the same flight, but this fails, because B has taken the last seat.

Explain this situation in transaction concepts:

- What kind of interference has taken place (dirty or unrepeatable read, phantom,...)?
- Which isolation levels would permit this situation?

b (Privileges, 3p). Assume an employee database with the following schema:

```
Persons(_id,name,address)
Employees(_id,position,salary)
  id -> Persons.id
```

What privileges are minimally needed

- by a customer who wants to find the names of the employees whose position is 'manager' ?
- by an administrator who wants to change the salary of an employee?

Show the required GRANT statements for customer and administrator, being careful not to grant more rights than necessary.

c (XML, 3p). The following XML document produces a validation error. What is the error? How could you fix it preserving as much data and possible and without changing the DTD?

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE FT [
<!ELEMENT FT (FT*)>
<!ATTLIST FT name CDATA #REQUIRED>
]>
<FT name="Victoria">
  <FT name="Estelle"/>
  <FT name="Oscar"/>
</FT>
<FT name="CarlPhilip">
  <FT name="Alexander"/>
</FT>
<FT name="Madeleine">
  <FT name="Leonore"/>
  <FT name="Nicolas"/>
</FT>
```