Institutionen för datavetenskap
Linköpings universitet

RETAKE EXAM
Databases for Bioinformatics
TDDD74

August 19, 2017
8.00 – 12.00

Grades

You can get max 30 points. To pass the exam, grade 3, you need 7.5 points in both the practical part (questions 1–3) and the theoretical part (questions 4–8) of the exam. For grade 4 and 5, you need 21 and 27 points, respectively.

Questions

Olaf Hartig will visit the room at 9.00 and at 10.30.

Instructions

- Write clearly.
- Use a separate page for every question.
- Answer in English.
- Give relevant and motivated answers only to the questions asked.
- State the assumptions you make besides those in the questions. None of these additional assumptions should change the spirit of the exercises.

Good luck!

Practical part (15 points)

Question 1. Data modeling with an EER diagram (5 p):

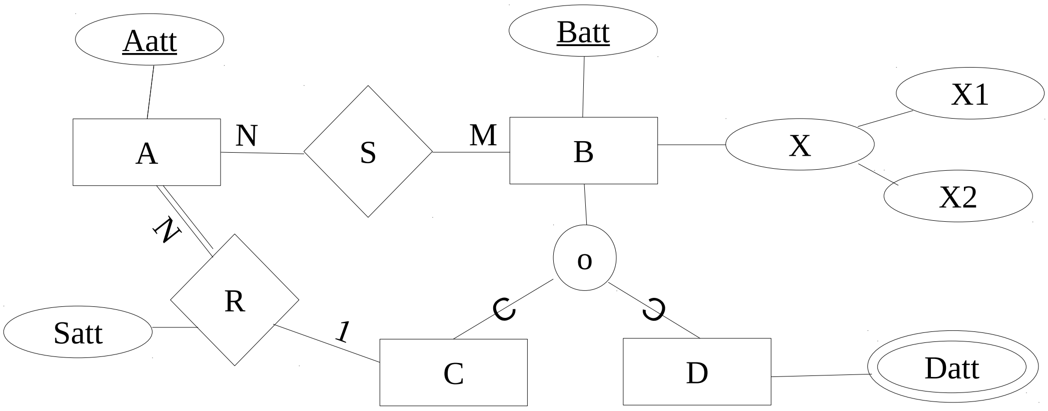
We want to create a database with the following information about figure skating events, skaters, and spectators.

- For the purpose of our database, each figure skating event has a unique event number. Furthermore, such events have a date, a start time, and an end time.
- Skaters and spectators are persons. Every person is identified by a social insurance number (SIN). Moreover, every person has a name and a birth date; the birth date is composed of a year, a month, and a day.
- Some persons are figure skaters who may perform in figure skating events.
- While not every skater performs in figure skating events, those who do, may perform in more than one of these events. On the other hand, every figure skating event must have one or more skaters performing in it.
- Every person (including figure skaters) may attend figure skating events as a spectator. However, not every person has to do so, and there may be events without any spectator. Of course, most events are attended by multiple spectators.
- For every person and every figure skating event that the person attends as a spectator, we want to record a ticket number of the ticket that the person used for entering the event.

Please draw an EER diagram that captures the aforementioned information (including cardinality constraints and participation constraints for participation of entities in relationships, as well as totalness constraints and disjointness constraints for specializations). Use the *notation as introduced in class*. Clearly write down your choices and assumptions in case you find that something in the information above is not clear.

Question 2. EER diagram and relational schema (5 p):

Translate the following EER diagram into an equivalent relational database schema, *by using the procedure introduced in class*. Your answer should be given in the form of a diagram that shows the relation schemas, including primary keys and foreign keys.



Question 3. SQL (1 + 2 + 2 = 5 p):

Consider the following database schema

Country(*Name*, *Code*, *Capital*, *Area*, *Population*)
Organization(*Name*, *Abbreviation*, *Established*)
IsMember(*Organization*, *Country*, *Joined*)

The attribute *Organization* in the table *IsMember* is a foreign key reference to *Abbreviation* in the table *Organization*. The attribute *Country* in table *IsMember* is a foreign key reference to *Code* in the table *Country*. Examples of the tuples for the above relational schema are as follows:

Country(Sweden, SWE, Stockholm, 449964, 9514000)
Organization(European Union, EU, 1952)
IsMember(EU, SWE, 1995-01-01)

Provide SQL queries to answer the following questions.

(a) List the names of all the organizations that the country named Zimbabwe is a member of.

(b) List the names of all organizations that do *not* have any members.

(c) List the name of every organization that has at least five members that have joined the organization before 1970. (Do not assume that organization names are unique.)

Theoretical part (15 points)

Question 4. Normalization (1 + 3 = 4 p):

Consider a relation schema *R*(*A*, *B*, *C*, *D*) for which the following functional dependencies exist:

FD1: $\{A,B\} \rightarrow \{C\}$
FD2: $\{C\} \rightarrow \{D\}$
FD3: $\{C\} \rightarrow \{A\}$

(a) Assume a relation state of *R* that contains the tuple *t* = (a1,b2,c6,d1). Name another tuple for *R* that, when inserted into *R* together with tuple *t*, would violate *both* FD2 and FD3.

(b) Normalize *R* up to Boyce-Codd normal form (BCNF). Explain your solution step by step. Bear in mind that a relation may have several candidate keys.

Question 5. Data structures (1 + 1 = 2 p):

Assume we have a sorted file with 1,000,000 records, a block size of 40,000 bytes, and unspanned allocation. Each record has a size of 40 bytes. The records have two fields, X and Y, where X is a key field (and Y is not). The file is sorted on X.

(a) Calculate the blocking factor of the file and the overall number of blocks that the file has.

(b) To speed up the retrieval we may use an index. Assume we want to speed up finding a record with a given value for X. Name **i)** the type of *single-level* index that we can use in this case and **ii)** the concrete number of index records that this index would have for our file.

Recall that $\log_2(2^x) = x$. That is, $\log_2(1) = 0$, $\log_2(2) = 1$, $\log_2(4) = 2$, $\log_2(8) = 3$, $\log_2(16) = 4$, $\log_2(32) = 5$, $\log_2(64) = 6$, $\log_2(128) = 7$, $\log_2(256) = 8$, $\log_2(512) = 9$, $\log_2(1024) = 10$, $\log_2(2048) = 11$, $\log_2(4096) = 12$, $\log_2(8192) = 13$, $\log_2(16384) = 14$, etc.

Question 6. Transactions and concurrency control (1 + 1 + 1 = 3 p):

(a) Remember that a schedule is a sequence of operations from multiple transactions. When do we say that such a schedule is *serial*? (i.e., define the notion of a serial schedule)

(b) Consider the following two schedules, S1 and S2. These two schedules are *not* conflict equivalent. Give a concrete reason why that is the case.

S1: $r_1(X)$, $w_1(X)$, $r_2(X)$, $w_2(X)$, $r_2(Z)$, $w_2(Z)$, $r_3(Z)$, $w_3(Z)$, $r_3(Y)$, $w_3(Y)$, $r_1(Y)$, $w_1(Y)$

S2: $r_1(X)$, $r_2(X)$, $w_1(X)$, $w_2(X)$

(c) Consider schedule S1 as given above. Is it serializable? Justify your claim.

Question 7. Database recovery (1 + 1 + 2 = 4 p):

(a) What is guaranteed to have happened when a transaction reaches its *commit point*?
(Note that this question is not about checkpoints.)

(b) Assume a DBMS applies the deferred update strategy without using checkpointing. Then,
i) when does the DBMS write out to disk the data blocks that a transaction has updated, and
ii) when does the DBMS write out the log buffers that contain log records about the transaction?

(c) Given the following log, apply each of the two recovery algorithms for the two immediate update strategies described in the course. In each of the two cases, list the operations that are performed during recovery in the order in which they are performed.

Start-transaction T2
Write-item T2, B, 3, 4
Start-transaction T3
Write-item T3, A, 7, 8
Checkpoint
Write-item T3, A, 8, 1
Commit T2
Checkpoint
Write-item T3, A, 1, 5
Start-transaction T4
Write-item T4, B, 4, 5
Write-item T4, B, 5, 10
Commit T3
Checkpoint
Start-transaction T1
Write-item T1, C, 8, 9
Commit T4
* system crash *

Question 8. Information Retrieval (1 + 1 = 2 p):

Assume that we use the vector model for information retrieval.

(a) Explain tf and idf in the vector model.

(b) Suppose that we are only interested in the words ‘gene’, ‘enzyme’, ‘protein’ and ‘signal’, and that we have two documents in our document base such that:

- Document 1 contains 'enzyme' 5 times, 'gene' 10 times, 'protein' 0 times and 'signal' 8 times.
- Document 2 contains 'enzyme' 0 times, 'gene' 0 times, 'protein' 7 times and 'signal' 1 time.

Give the document representations for Document 1 and Document 2 according to the tf-idf model.