Institutionen för datavetenskap
Linköpings universitet

RETAKE EXAM
Database Technology
TDDD37 – TDDD46

April 26, 2019
14.00 – 18.00

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

Patrick Lambrix will visit the room between 15.30 and 16.00.

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 information about figure skating events, skaters, and spectators.

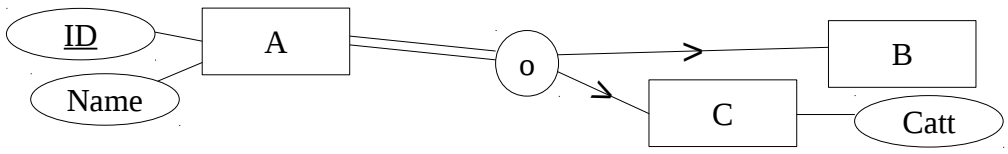
- For the purpose of our database, each figure skating event has a unique event number. Furthermore, such events have a date, a start time, and an end time.
- Skaters and spectators are persons. Every person is identified by a social insurance number (SIN). Moreover, every person has a name and a birth date; the birth date is composed of a year, a month, and a day.
- Some persons are 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 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 figure skating event that a spectator attends, we want to record a ticket number of the ticket that the spectator 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.

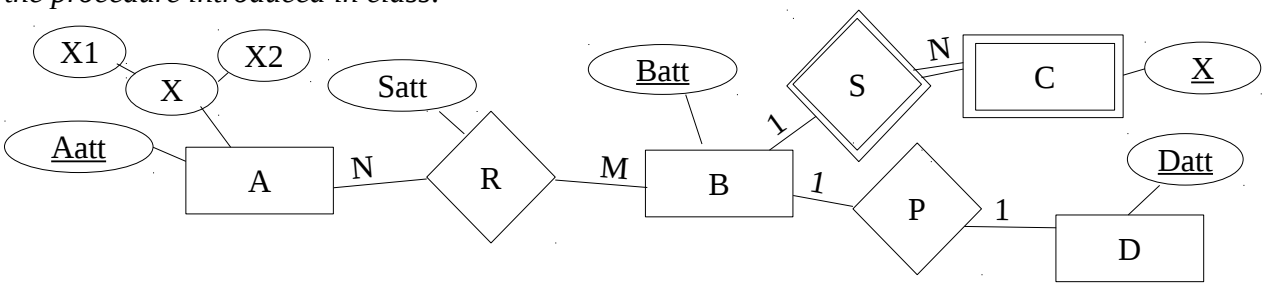
Question 2. EER diagram and relational schema (2 + 3 = 5 p):

For both of the following questions, your answer should be given in the form of a diagram that shows the relation schemas, including primary keys and foreign keys.

(a) Recall that there exist different approaches to translate specializations of entity types (i.e., super-classes with their sub-classes). Apply ***two*** possible approaches (from the approaches discussed in class) to the following example of such a specialization. That is, create two separate relational database schemas such that each of them illustrates the application of one of the approaches.



(b) Translate the following EER diagram into an equivalent relational database schema, *by using the procedure introduced in class*.



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

Consider a database created by the following SQL statements.

```
CREATE TABLE Continent ( cid INTEGER  PRIMARY KEY,
                         name VARCHAR(30)  );
CREATE TABLE Country ( code INTEGER  PRIMARY KEY,
                       name VARCHAR(30),
                       continent INTEGER,
                       CONSTRAINT fk_cont FOREIGN KEY (continent)
                                          REFERENCES Continent(cid)  );
CREATE TABLE IsMember ( country INTEGER,
                        organization VARCHAR(30),
                        CONSTRAINT PRIMARY KEY (country, organization),
                        CONSTRAINT fk_ism FOREIGN KEY (country)
                                          REFERENCES Country(code)  );
```

(a) Assume the database has been populated with some data such that none of the tables is empty (i.e., each of them contains at least one row) and the current state of the database is valid. Now, consider the following SQL statements. Something is wrong with each of them. That is, for each of them, you would get an error message when trying to execute it using a system that complies to the SQL standard. For each of the two statements, write down the reason for why it is wrong (i.e., what mistake has been made). If there are multiple reasons, it is sufficient to write down only one of them (no extra points for finding multiple mistakes).

(a.i)
```
SELECT continent, code, COUNT(*)
FROM Continent, Country
WHERE cid = continent AND name LIKE "A%"
GROUP BY continent;
```

(a.ii)
```
CREATE TABLE IsMember ( country INTEGER  PRIMARY KEY,
                        organization INTEGER,
                        CONSTRAINT fk_ism FOREIGN KEY (country)
                                          REFERENCES Country(continent)  );
```

(b) For the same database, provide an SQL query whose result consist of *a single column* that contains the names of all continents and of all countries.

(c) For the same database, provide an SQL query that lists (in a single column) the IDs of the continents that contain at least one country which is not a member of any organization. The list has to be duplicate-free.

Theoretical part (15 points)

Question 4. Normalization (1 + 1 + 2 = 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 $t = (1,2,6,1)$. Name another tuple for R that, when inserted into R together with tuple t, would violate *both* FD2 and FD3.

(b) Show that *R* is not in Boyce-Codd normal form (BCNF).

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

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

(a) What does *spanned allocation* mean (in the context of files that consist of blocks with records)?

(b) Assume we have a *heap file* with 1,000,000 records, a block size of 40,000 bytes, and unspanned allocation. Each record has a size of 400 bytes. The records contain only one field, X, which is a key field. For each of the following *four* points, provide only the numbers that are asked for; that is, *do **not** write any explanation/justification!*

Calculate: **i)** the blocking factor of the file and **ii)** the overall number of blocks that the file has.

Moreover, assume we want to find a record with a given value for X. How many block accesses are needed **iii)** in the best case and **iv)** in the worst case? (do not assume the existence of any index)

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) Consider the following schedule *S*. Is it *serializable*? Justify your claim.

S : $b_1, r_1(X), b_2, r_2(Y), w_1(X), b_3, w_2(Y), e_2, r_1(Y), r_3(X), e_3, w_1(Y), e_1$

(b) Consider again the same schedule *S*. Is this schedule *serial*? Justify your claim.

(c) What is the *isolation* property that is desired for transactions?
(Note that this is a general question; i.e., it is independent of the aforementioned schedule.)

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

(a) In the case of the deferred update strategy, one of the following is true: there is either no need to undo changes of non-committed transactions or no need to redo changes of committed transactions. *Which* one is it (no need to undo or no need to redo), and *why*?

(b) Something is wrong with the following log. What is it?

Start-transaction T1
Write-item T1, A, 4, 62
Start-transaction T2
Write-item T2, B, 8, 91
Checkpoint
Write-item T1, B, 91, 1
Commit T1
Checkpoint
Write-item T2, A, 1, 5
Commit T2

(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. For each operation in these two lists, indicate explicitly which value is written by the operation; you can do this by specifying the (new) log record resulting from the operation.

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. Query Processing (1 p):

Assume a relation *R* and let *pr* be the number of disk pages occupied by the file for this relation. What is the I/O cost (in terms of page reads) of performing the table scan algorithm over *R*? (Write only the answer to the question; no explanation is needed.)