Institutionen för datavetenskap
Linköpings universitet

EXAM
PART 1
Database Technology
TDDD12 – TDDD81

May 29, 2020
14:00 – **15:55**

Instructions

See: https://www.ida.liu.se/divisions/adit/dwis/dbcourse/distanceexam.shtml

Grades

You can get up to 15 points for this first part of the exam and another 15 points for the second part, which together may give you an overall of max 30 points.

To pass the exam (grade 3) you have to meet both of the following two conditions: First, you need to achieve at least 8 of the 15 points that can be achieved in this first part of the exam. Second, for both parts together, you need to achieve at least 15 of the 30 points that can be achieved overall. If you do not meet the first condition, your second part will *not* be considered for grading.

After fulfilling the aforementioned requirements to pass the exam, then for grade 4, you need at least 21 points (for both parts together), and for grade 5, you need at least 27 points.

Questions

If you have clarification questions regarding some of the exercises in the exam, the examiner will be available in the following Zoom meeting room throughout the whole time of the exam.

https://liu-se.zoom.us/j/67490211293?pwd=YkNHTEpLMS9BNGJWOXdtOGh0K2JHdz09
Meeting ID: 67490211293
Password: 556432

Notice that this Zoom meeting room has been set up using the waiting room feature of Zoom. Hence, when you enter, you will be put into the waiting room and, from there, you will then be admitted to the meeting room to ask your question.

Question 1. Data modeling with an EER diagram (3 p):

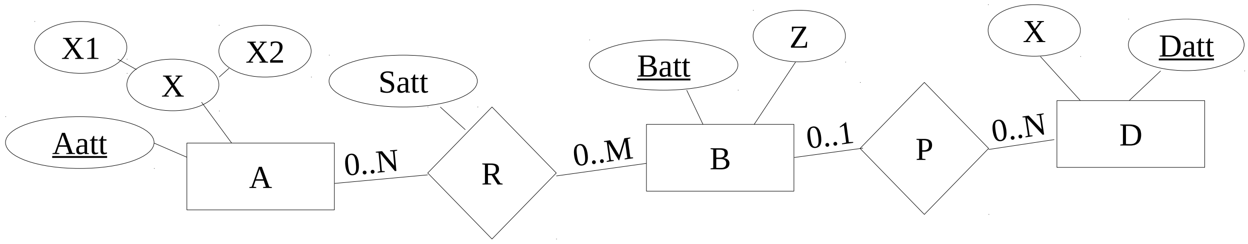
Consider the following *data requirements* for a database about chess competitions and participants.

- For the purpose of our database, each chess competition has a unique event number. Furthermore, such competitions have a date, a start time, and an end time.
- Chess players participate in such competitions. More specifically, every player may participate in one or more competitions (players who do not participate in any competition are irrelevant for our database); on the other hand, every competition may have multiple participating players but it may also have none (to be able to store competitions for which participants still have to register).
- Every chess player is identified by a social insurance number (SIN). Moreover, every player has a name and a birth date; the birth date is composed of a year, a month, and a day.
- Aside from participating in chess competitions, some chess players (but not all) have a favorite chess competition. Some chess competitions may be favored by multiple players, whereas others are not favored by any player.

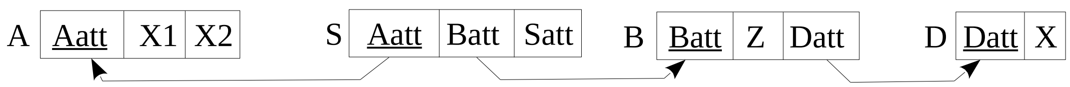
Draw an EER diagram that captures the aforementioned information as accurately as possible (including all types of constraints that can be captured in such diagrams). Use any of the two notations considered in class. Clearly write down your choices and assumptions in case you find that something in the information above is not clear.

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

Consider the following EER diagram.



A friend of you tried to translate this EER diagram into a relational database schema. The result is illustrated below. Unfortunately, your friend made two mistakes. Help your friend by identifying both mistakes. For each of the two mistakes, describe the mistake in one or two sentences.



Question 3. SQL (1 + 1 = 2 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 statement. Something is wrong with this statement. That is, you would get an error message when trying to execute it using a system that complies to the SQL standard. Explain in one or two sentences why the statement is wrong (i.e., describe 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).

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

(b) What is the intention of the following SQL query? To answer this question describe in one or two sentences what the query returns.

```
SELECT cid
FROM Continent
WHERE cid NOT IN ( SELECT continent
                   FROM Country, IsMember
                   WHERE code = country AND organization = "EU" );
```

Question 4. Functional Dependencies (1 + 1 = 2 p):

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

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

(a) Pick a random number between 0 and 10. Pick another random number between 10 and 20. Now, let $t = (x, y, y)$ be the tuple for *R* in which *x* is the first number you have picked, and *y* is your second number. Write down this tuple with your numbers. Next, name another tuple *t'* for *R* that, when inserted into *R* together with your tuple *t*, would violate FD3 but not FD1 and also not FD2. Hence, as answer to this question you only need to write your two tuples, *t* and *t'*.

(b) For the set $X = \{A\}$, the attribute closure of *X* with respect to the three functional dependencies given above is $X^+ = \{ A, C \}$. Provide a step-by-step description of a concrete process to determine this set X^+ (do not forget the steps that check whether you have reached a fixed point).

Question 5. Data structures (1 p):

Take the last digit of your LiU ID and multiply it by 1 000. Write down the resulting number and use it as the *block size* for the following task. For instance, if your LiU ID is `xyzbe146`, then you use a block size of 6 000 bytes. Additionally, multiple the last digit of your LiU ID by 10 000, write down the resulting number, and use this number as the *number of records* for the following task. Attention: If the last digit of your LiU ID is 0 (zero), assume it was 6 instead; hence, in this case, use a block size of 6 000 bytes and 60 000 records.

Now, assume a heap file with this block size and this number of records. Suppose that this heap file uses unspanned allocation, and each record in the file has a size of 100 bytes.

Determine the overall number of blocks that the file has and explain you reasoning step by step (just writing the number does no earn you any points even if your result is correct).

Question 6. Transactions and concurrency control (1 + 1 = 2 p):

Consider three transactions (T_1, T_2, and T_3) with the following sequences of operations:

T_1 : $r_1(X)$, $w_1(X)$, $r_1(Y)$, $w_1(Y)$

T_2 : $r_2(Y)$, $w_2(Y)$

T_3 : $r_3(X)$

(a) Create a schedule that contains all of these three transactions and that is *not serial*, and explain why your schedule is not serial.

(b) Take the operation $w_2(Y)$ of transaction T_2 and identify *every* operation in any of these three transactions that is in conflict with the operation $w_2(Y)$. This question can be answered by simply writing down all the conflicting operations that you have identified.

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

(a) Assume a DBMS that uses the deferred update strategy but no check pointing. When such a system has to recover a database after a system failure, which transactions does it have to redo? (This question can be answered independent of any specific log file.)

(b) Remember that after a transaction has reached its commit point, the DBMS has to make sure that the effects of this transaction are made permanent in the database even in the case of a system failure after the commit point. Describe in two to four sentences how this guarantee is achieved when using the strategy Immediate Update 1.

Question 8. Query Processing (1 p):

Assume we want to join two relations, *R1* and *R2*, where the file for *R1* occupies 100 disk pages (blocks) and the file for *R2* occupies 1 000 pages. Suppose the only join algorithm that is available in our DBMS is the nested loops join algorithm (NLJ).

What would be the lowest possible I/O cost (in terms of page reads) that our system can achieve by applying the NLJ to join the two relations? Explain your reasoning in about two to four sentences.

Hint: there are two different ways in which the NLJ can be used to join the two relations.