Information page for written examinations at Linköping University

Examination date	2019-08-24
Room (1)	TER3(18)
Time	8-12
Edu. code	TDDD74
Module	TEN1
Edu. code name **Module name**	Databases for Bioinformatics (Databaser för bioinformatik) Written examination (Skriftlig tentamen)
Department	IDA
Number of questions in the examination	8
Teacher responsible/contact person during the exam time	Olaf Hartig
Contact number during the exam time	5639
Visit to the examination room approximately	9.00 and 10.30
Name and contact details to the course administrator (name + phone nr + mail)	Annelie Almquist, 013-28 29 34 annelie.almquist@liu.se
Equipment permitted	Dictionary
Other important information	
Number of exams in the bag	

Institutionen för datavetenskap
Linköpings universitet

RETAKE EXAM
Databases for Bioinformatics
TDDD74

August 24, 2019
8.00 – 12.00

Grades

You can get max 30 points. To pass the exam, grade 3, you need 7.5 points both in the practical part (questions 1–3) and in 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 music concerts.

- Every concert has a unique number, a date, a start time, and an end time.
- A person is identified by a social insurance number (SIN), and has a name and a birth date; the birth date is composed of a year, a month, and a day.
- Every person in the database is either a musician or a concert-goer (but not both).
- Concert-goers attend concerts (at least one!). There may be concerts that nobody attends. Of course, most concerts are attended by multiple concert-goers.
- Musicians may perform concerts and may have multiple nicknames.
- While not every musician performs concerts, those who do, may perform more than one concert. On the other hand, every concert must have one or more musicians performing it.

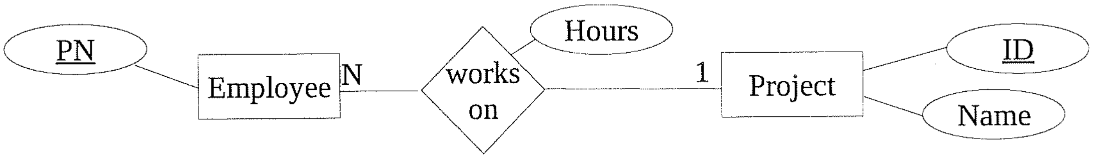
Please draw an EER diagram that captures the aforementioned information (including cardinality constraints and participation constraints for participation of entities in relationships, as well as totalness constraints and disjointness constraints for specializations).

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.

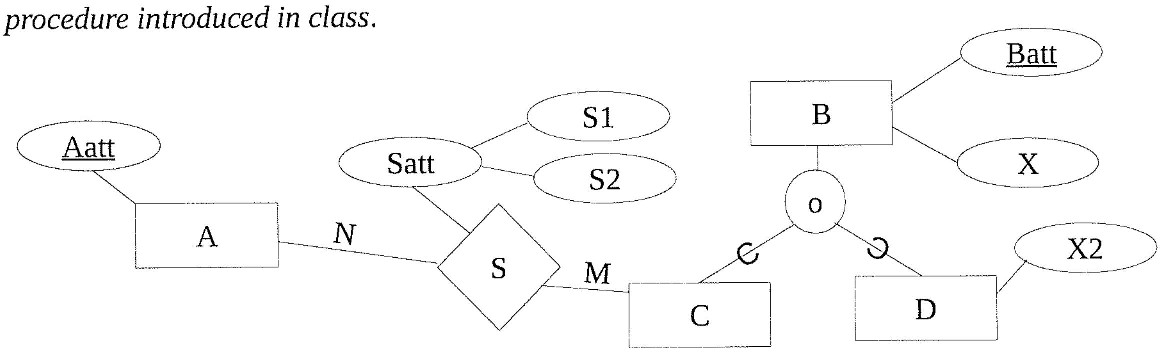
Question 2. EER diagram and relational schema (2 + 3 = 5 p):

For both of the following questions, your answers should be given in the form of diagrams that show the relation schemas, including primary keys and foreign keys.

(a) Recall that we have two different approaches to translate a 1:N relationship type during the translation of an ER diagram to a relational database schema. Apply *both* of these approaches to the following example of such a 1:N relationship type. That is, create two separate relational database schemas such that each of them illustrates the application of one of the two approaches.



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



Question 3. SQL (1 + 1 + 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) );
```

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.

(a) Consider the following SQL statement. Something is wrong with it. That is, you would get an error message when trying to execute the statement over the aforementioned database using a system that complies to the SQL standard. Write down a reason for why the statement is wrong (i.e., what mistake has been made). If there are multiple reasons (mistakes), it is sufficient to write down only one of them (no extra points for finding multiple mistakes).

```
SELECT continent, COUNT(DISTINCT country), COUNT(DISTINCT organization)
FROM Country, IsMember
WHERE code = country AND Continent.name LIKE "%America"
GROUP BY continent
HAVING organization LIKE "A%";
```

(b) For the same database, consider the following SQL statement. While the statement is syntactically correct, there are cases in which executing this statement would still fail with an error. Describe such a case in which the given statement would fail.

```
INSERT INTO Country VALUES (27, "Sweden", 3);
```

(c) For the same database, provide an SQL query that lists (in a single column) the names of all organizations mentioned in the database. The list has to be duplicate-free.

(d) For the same database, provide an SQL query that lists (in a single column) the names of the continents that contain at least one country which is not a member of any organization.

Theoretical part (15 points)

Question 4. Normalization (1 + 1 + 2 = 4 p):

Consider a relation schema *R*(*A*, *B*, *C*, *D*) with the following four functional dependencies:

FD1: $\{A\} \rightarrow \{B\}$
FD2: $\{B\} \rightarrow \{A\}$
FD3: $\{C\} \rightarrow \{D\}$
FD4: $\{D\} \rightarrow \{C\}$

(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 $\{A,C\}$ is a candidate key of *R*.

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

(d) 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 + 1 = 2 p):

Assume we have a sorted file with 100,000 records, a block size of 40,000 bytes, and unspanned allocation. Each record has a size of 400 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. For each of the following points, provide only the numbers that are asked for; that is, *do **not** write any explanation/justification.*

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

(b) Calculate the average number of block accesses needed to find records **i)** with a given value for X, and **ii)** with a given value for Y (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 + 1 = 4 p):

(a) Consider the operation $r_1(X)$, i.e., a read operation of data item *X* by transaction *1*. Name an arbitrary other operation such that this other operation conflicts with operation $r_1(X)$. (There is no need to provide any explanation/justification; simply name the operation.)

(b) 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)

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

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

(d) Specify the two-phase locking (2PL) protocol; what does a transaction have to do to follow the protocol? (Note that this is a general question; it is independent of the aforementioned schedule.)

Question 7. Database recovery (1 + 2 = 3 p):

(a) Which of the four ACID properties does a DBMS have to guarantee only for transactions that have reached their commit point (and not for any other transactions)? (There is no need to provide any explanation/justification; simply name the particular ACID property.)

(b) 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
Commit T4
* system crash *
```

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

Assume that we have two documents in our document base where:

- Document 1 contains the word "*enzyme*" 5 times, "*gene*" 10 times, "*protein*" 0 times, and "*signal*" 8 times;
- Document 2 contains the word "*enzyme*" 0 times, "*gene*" 0 times, "*protein*" 7 times, and "*signal*" 1 time.

Suppose we use the Boolean model for information retrieval, and we are only interested in the words "*gene*", "*enzyme*", "*protein*", and "*signal*".

(a) Give the document representations for Documents 1 and 2 according to the Boolean model.

(b) Represent the query to retrieve all documents that contain "*gene*" or "*protein*", but not "*signal*". Compute then the completed DNF (disjunctive normal form) of the query.