Institutionen för datavetenskap
Linköpings universitet

EXAM
TDDD37 Database Technology
TDDD46 Database Technology

January 16, 2016, 8.00-12.00

Help

Dictionary.

Grades

You can get max 32 points. To pass the exam, grade 3, you need 7.5 and 8.5 points in the practical and theoretical parts of the exam, respectively. For grade 4 and 5, you need 22 and 29 points, respectively.

Questions

Jose M. Peña will visit the room at 10.00.

Instructions

You can answer in Swedish or English. Write clearly. 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 EER diagram (5 p):

We want to create a database to store geographical information. Specifically, we want to store information about some countries, where each country is divided in regions and each region consists of some cities. For a country, we want to store its neighboring countries. For a region, we want to store its neighboring regions (even if they belong to other countries). For a city, we want to store its neighboring cities (even if they belong to other countries), as well as the road segment that connects each pair of neighboring cities (we assume that there is only one such segment). For a road segment, we want to store some additional information, such as number of lanes, length, maximum speed, etc. Since this information may change within a given segment, you may want to split the segment in sub-segments where the information does not change.

Draw an EER diagram for the description above. Feel free to add the attributes that you consider necessary. Clearly write down your choices and assumptions in case you find that something in the information above is not clear.

Question 2. MySQL (1 + 2 + 2 = 5 p):

Consider the Jonson Brothers' relational schema used in the labs. The following relations should suffice to answer the queries below. However, you are free to use any other relation in the relational model used in the labs.

Relation: jbemployee
An employee is identified by an id and described by name, salary, birthyear and startyear. The id of the manager of each employee is also supplied. A null value means that the employee has no manager.

Relation: jbitem
An item is identified by an id and described by its name, the department where it is sold, its price, the quantity on hand (qoh) and the identifier of the supplier that supplied it.

Relation: jbsupplier
A supplier (of items and parts) is identified by its id and described by its name and the city in which it is located.

Relation:jbparts
A part, used internally by the store, not sold to customers, is identified by its id and described by its name, color, weight, and the quantity on hand (qoh).

Produce the MySQL code to answer the following queries:

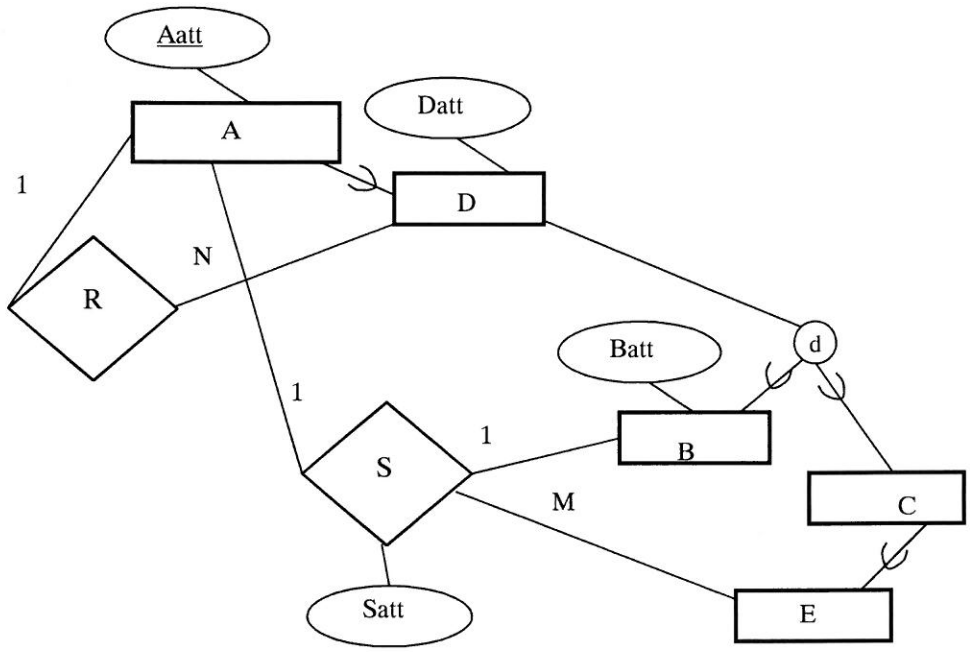
1. What was the age of each employee when they started working (startyear)?

2. Which items (note items, not parts) have been delivered by a supplier called Fisher-Price? Formulate this query using a subquery in the where-clause.

3. What is the name and color of the parts that are heavier than a card reader? Formulate this query without using a subquery in the where-clause.

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

Translate the EER diagram below into a relational schema. Use the algorithm you have seen in the course.



Theoretical part (15 points)

Question 4. Normalization (3 p):

Normalize up to Boyce-Codd normal form (BCNF) the relation R(A, B, C, D, E, F, G) with functional dependencies {ABC→DEFG, BC→DEF, C→D, F→G}. Explain your solution step by step. Bear in mind that a relation can have several candidate keys.

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

We have a file with 1000000 records. Each record is 10 bytes long. The records have two key attributes X and Y. The file is ordered on X. The database uses a block size of B=1000 bytes and unspanning allocation. Each index record is 2 bytes long.

1. Calculate the average (or the maximum, if you prefer) number of block access needed to find a record with a given value for X when using (a) the primary access method and (b) a single level index.
2. Calculate the average (or the maximum, if you prefer) number of block access needed to find a record with a given value for Y when using (a) the primary access method and (b) a single level index.
3. Describe in at most two sentences what a B+ tree is.

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 (2 + 1 + 1 = 4 p):

1. Give an example of a serializable schedule and an example of a non-serializable schedule.

2. Apply the two-phase locking protocol to the two schedules above.

3. Complete the following sentences:

    a. Two schedules are conflict equivalent if …

    b. A schedule is serializable if …

Question 7. Database recovery (3 p):

Consider the system log below:

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
Write-item T3, A, 1, 5
Checkpoint
Start-transaction T4
Write-item T4, B, 4, 5
Checkpoint
Write-item T4, B, 5, 10
Start-transaction T1
Write-item T1, C, 8, 9
Write-item T1, C, 9, 10
→system crash

Now, add somewhere in the system log above the instructions Commit T2, Commit T3, and Commit T4 so that recovery with deferred update performs as few operations as possible. Repeat the exercise for the other two recovery methods you have seen in the course.

Question 8. Optimization (2 p):

Let R(A,B), S(B, C), T(C,D), P(D,A) be four relations with the underlined attributes as keys. Optimize the following MySQL query:

```
SELECT S.B, T.C, P.A, R.A
FROM R, S, T, P
WHERE R.B = 5 AND S.C = 5 AND T.D = 5 AND P.A = R.A;
```