Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2011-01-07
Sal	
Tid	14-18
Kurskod	TDDD37
Provkod	TEN1
Kursnamn/benämning	Databasteknik / Database Technology
Institution	*IDA*
Antal uppgifter som ingår i tentamen	8
Antal sidor på tentamen (inkl. försättsbladet)	försättsblad + 6
Jour/Kursansvarig	Jose Pena / He Tan
Telefon under skrivtid	0708229596 / 0739092890
Besöker salen ca kl.	
Kursadministratör (namn + tfnnr + mailadress)	
Tillåtna hjälpmedel	Dictionary
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	
Vilken typ av papper ska användas, rutigt eller linjerat	
Antal exemplar i påsen	

Institutionen för datavetenskap
Linköpings universitet

TENTAMEN
TDDD37
Databasteknik / Database Technology

January 7, 2011, 14-18

Grades: The exam consists of 2 parts. For a pass grade you need to obtain 50% of the total points on **each** part. When a pass grade is obtained, the final grade is based on the total result and not on the different parts.

Instructions: In addition to the instructions on the cover page:

- Write clearly.
- Start the answers to a question on a new page.
- If you make assumptions that are not given in a question, then clearly describe these assumptions. (Of course, these assumptions cannot change the exercise.)
- You can answer in English or Swedish.

Tools: dictionary

LYCKA TILL!

Practical part (15 points)

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

Read the whole exercise before starting.

The second hand car company, "NiceCar", sells, buys and repairs cars. The company has much information which it must keep track of. Each of its employees has a name and a social security number. Salesmen sell cars to and buy cars from customers. The company keeps track of the date and price of selling and buying, and also customers' information, e.g. name, address and phone number. For selling, the information of salesmen commission is needs to be stored. The information about a car includes the registration number, the manufacturer, the model and the produced year. Mechanics repair cars. The information of each repairing, e.g. the date, a short description of the repairing, and the cost, need to be stored. The total repairing cost includes a base price and the cost for repairing the car parts. The base price for a repair and the price for repairing different car parts are determined for each year. The company keeps track of the base price and price for repairing different car parts for different years.

Draw an EER diagram for this second hand car dealer for the data described above.

Question 2. SQL (0.5+1+1.5+1.5+1.5=6p):

Study the following relations describing movies and movie theaters:

Movie:

Name	**ReleaseYear**	**Runtime**	**MPAA Rating**
Inception	2010	148 min	PG-13
Monsters, Inc.	2001	92 min	G
Lolita	1962	152 min	-
Lolita	1997	137 min	R
...	...	...	...

MovieTheater:

Id	**Name**	**Address**
1	Saga	Kungsgatan 24, Stockholm
2	Victoria	Götgatan 67, Stockholm
...	...	...

Playing:

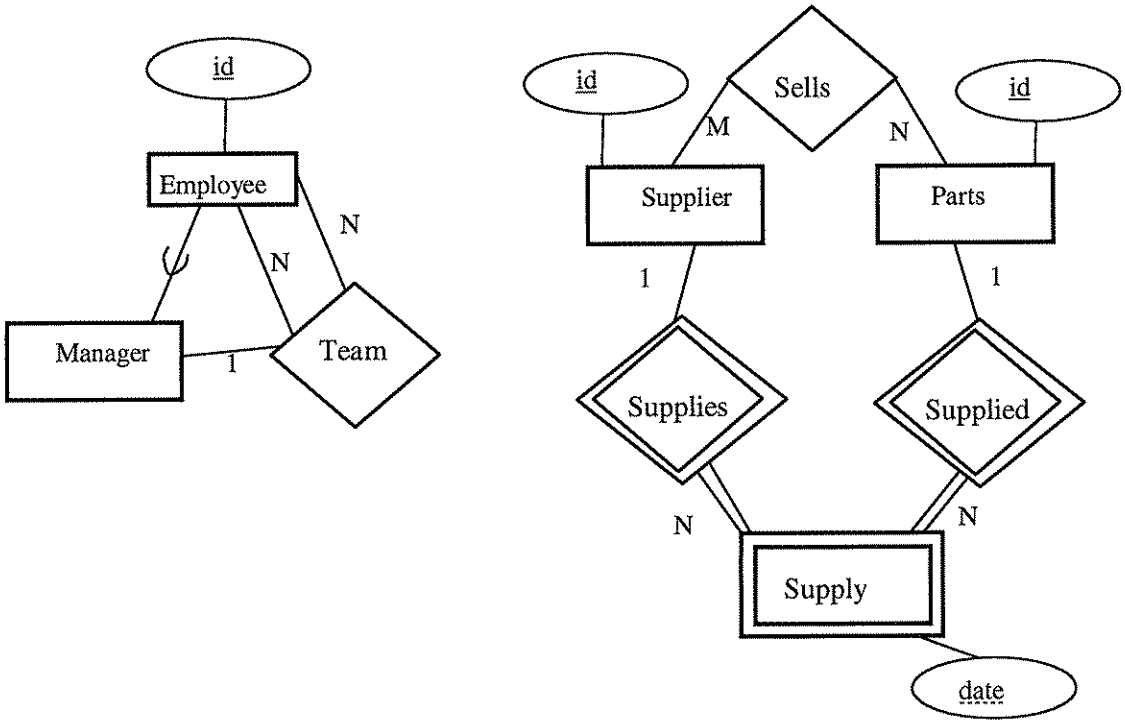
MovieName	**MovieReleaseYear**	**MovieTheater**
Inception	2010	1
Inception	2010	2
Monsters, Inc.	2001	2
...	...	...

(MovieName, MovieReleaseYear) is a foreign key referring to Movie(Name, ReleaseYear)
MovieTheater is a foreign key referring to MovieTheater(id)

Write SQL queries for the following:

a) List all the movies released in 2010. (0.5p)
b) List the names of all movies that are playing in Saga. (1p)
c) List the names of the theaters that are not playing the movie "Inception". (1.5p)
d) List the names of the theaters that are playing more than 6 different movies. (1.5p)
e) List the movies that are not playing in the same theater. (1.5p)

Question 3. Translation of EER to relational schema (4 p):



Translate the EER diagram above into a relational model (you have to follow the algorithm seen in the course). Mark the primary keys with solid underlining and the foreign keys with dotted underlining and an arrow from the foreign key to the attribute(s) pointed by the foreign key.

Theoretical part (16 points)

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

Assume the relation R(A,B,C,D,E) with the following functional dependencies:
AB → C D E and D →C E

a) Give a primary key for the relation.
b) Which normal form is the relation in? Motivate.
c) Normalize into BCNF. Show every step in the normalization procedure.

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

Assume a table with 1,000,000 records. The table is ordered on the key field ***X***. Each record is 400 bytes long. The database uses block size B = 4 096 bytes and the records are stored unspanning. Assume we create an index based on another key field ***Y***, where each index record uses 12 bytes (8 bytes for the search key and 4 bytes for the block pointer)

a) How many block accesses are needed to find a record with a given value for key field ***Y*** when no index is used?
b) How many block accesses are needed to find a record with a given value for key field ***Y*** when the index is used?

(You do not need to calculate it, if there is ***log*** in your answer.)

Question 6. Transactions and concurrency control (2+1=3p):

a. Use the two transactions below to give an example of a transaction schedule that is serializable but not serial. Explain why the schedule is serializable.

T1	**T2**
read(x)	read(x)
read(y)	read(y)
x:=x+y	y:=y+x
write(x)	write(y)

b. Apply the two-phase locking protocol to each of the transactions above.

Question 7. Database recovery (3p):

Apply the three recovery methods seen in the course to the system log below. Show all operations that are performed during the recovery. In the correct order!

Part of system log:

Start-transaction T1
Write-item T1, A, 1, 2
Start-transaction T2
Write-item T1, A, 2, 4
Write-item T2, B, 5, 6
Start-transaction T3
Start-transaction T4
Write-item T4, C, 6, 7
Write-item T3, C, 7, 8
Write-item T3, C, 8, 10
Checkpoint
Write-item T2, B, 6, 12
Checkpoint
Commit T1
Commit T2
→system crash

Question 8. Optimization (1+1+1=3p)

a. Let A, B, C and D be four tables with 10 attributes each. Each of the attributes has the UNIQUE constraint. Optimize the following MySQL query:

```
SELECT A.a
FROM A, B, C, D
WHERE A.pk=B.pk AND B.pk=C.pk AND C.pk=D.pk AND D.funnykey=A.pk;
```

b. Assume that the tables do not contain any NULL value. Assume also that each table contains 10 tuples and that each attribute is of size 1 byte. Show that the optimized query tree is more efficient than the canonical query tree.

c. Why does query optimization replace a selection followed by a Cartesian product with a join operation?