Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2010-10-18
Sal	TER3 - TER4
Tid	14-18
Kurskod	TDDD37 + TDDC94
Provkod	TEN1
Kursnamn/benämning	Databasteknik / Database Technology
Institution	*IDA*
Antal uppgifter som ingår i tentamen	8
Antal sidor på tentamen (inkl. försättsbladet)	8
Jour/Kursansvarig	Patrick Lambrix /Lena Strömbäck /Jose Pena
Telefon under skrivtid	2605 / 2324 / 1651
Besöker salen ca kl.	15.00, 16.30
Kursadministratör (namn + tfnnr + mailadress)	
Tillåtna hjälpmedel	dictionary, calculator
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	pass: 7.5 on part 1 + 8.5 on part 2
Vilken typ av papper ska användas, rutigt eller linjerat	
Antal exemplar i påsen	

Institutionen för datavetenskap
Linköpings universitet

TENTAMEN
TDDD37 / TDDC94
Databasteknik / Database Technology

October 18, 2010, 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, calculator.

LYCKA TILL!

Practical part (15 points)

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

An online photo forum has much information which it must keep track of. Each of its members has a unique id, user name, registration date and the photography equipment he/she is using. The photography equipment includes camera, lens and filter. Members can upload their photos for sharing. Each photo has its unique id, title, exposure date, post date, category, and whether it is manipulated. The forum keeps track of the member who uploads a photo, and the photography equipment that was used for shooting the photo. Every member can comment on photos. Each comment has its post date. The forum invites professional photographers, such as journalists, to become their expert members. Each expert member has a biography. Every week one photo is selected as "the photo of week". A photo can be "the photo of week" only once. The forum invites one of the expert members to write a short review on "the photo of week". For "the photo of week" the forum stores the week number, the review and the reviewer.

Draw an EER diagram for the photo forum for the data described above.

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

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
...	...	...	...

MovieTheater:

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

Playing:

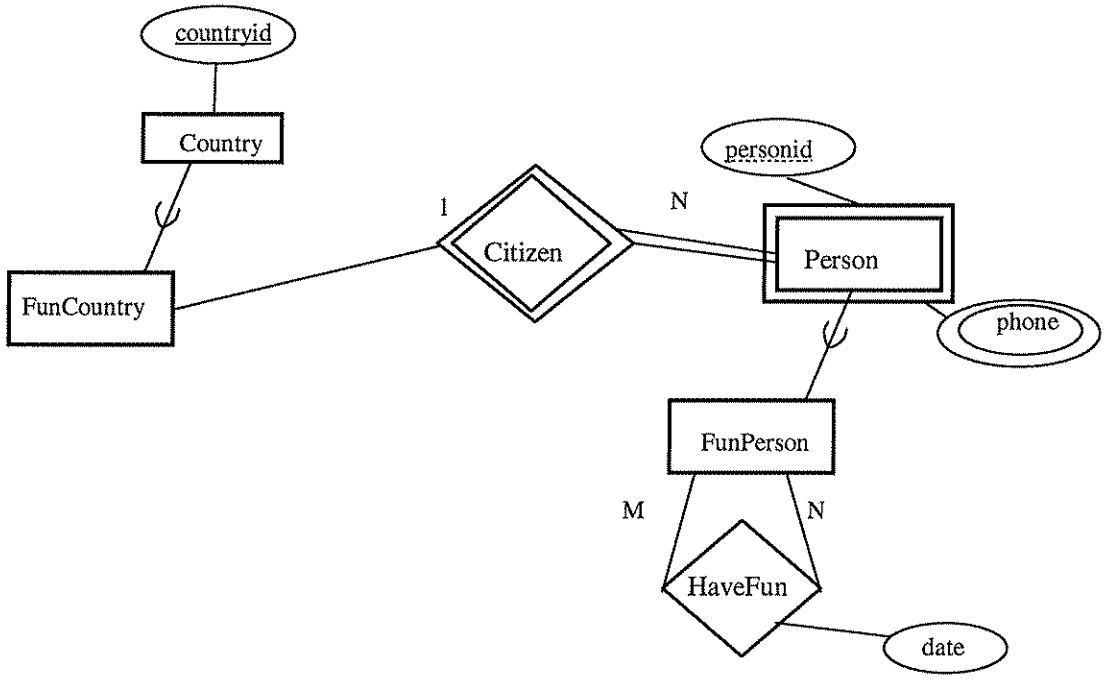
MovieName	**MovieReleaseYear**	**MovieTheater**
Inception	2010	Saga
...	...	...

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

Write SQL queries for the following:

a) List all movies that were released between 2006 and 2008, and have been played in the theater Saga. (1p)
b) List the theaters that are not playing the movie "Inception". (1.5p)
c) List the theaters that are playing more than 6 different movies. (1.5p)
d) List the movies that were released in the same year, but never have been played in the same theater? (2p)

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 dashed underlining and an arrow from the foreign key to the attribute(s) pointed by the foreign key.

Theoretical part (17 points)

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

A poken is a digital business card. As a user you register your contact information on a web page. When you meet a person that owns a poken you simple let the pokens touch each other to exchange information. Every user may have more than one poken. The following relation describes meetings between persons having pokens. You may assume the following functional dependencies between the attributes in the relation: Poken1 → UserID1, UserID1 → Name1, Poken2 → UserID2, UserID2 → Name2.

Tidpunkt	**Poken1**	**AnvID1**	**Namn1**	**Poken2**	**AnvID2**	**Namn2**
1009121315	POK1	lestr@ida	Lena	POK2	annva@ida	Anna
1009121320	POK2	annva@ida	Anna	POK3	sarst@ida	Sara
1009121311	POK1	lestr@ida	Lena	POK3	sarst@ida	Sara
1009121315	POK3	sarst@ida	Sara	POK5	marho@ida	Maria
1009121311	POK4	annva@ida	Anna	POK5	marho@ida	Maria

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 (2 + 2 = 4 p):

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.

i) Assume we create a secondary index based on another key field ***Y***, where each index record uses 8 bytes (4 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?

ii) By using B-tree index structure, how many block accesses are needed to find a record with a given value for key field ***Y***, if the base level index is the index we create on the key field ***Y***? We assume 4 bytes for the search key, 4 bytes for the block pointer, 8 bytes for a record pointer, and fill percentage is 70%.

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

a. Is the following transaction schedule serializable? Motivate your answer.

T1	T2	T3
	read(y)	
	y:=y+1	
	write(y)	
read(x)		
x:+x+1		
write(x)		
		read(y)
		y:=y+1
		write(y)
	read(x)	
	x:=x+1	
	write(x)	

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

c. Complete the following sentence with one of the four options given. If we use the two-phase locking protocol there can be a) starvation, b) deadlocks, c) both starvation and deadlocks, d) neither starvation nor deadlocks.

Question 7. Database recovery (3 p):

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, 3, 4
Write-item T2, B, 5, 6
Commit T1
Start-transaction T3
Start-transaction T4
Write-item T3, C, 7, 8
Write-item T3, C, 9, 10
Write-item T2, B, 11, 12
Checkpoint
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?