Försättsblad till skriftlig tentamen vid Linköpings universitet

	(fylls i av ansvarig)
Datum för tentamen	Januari 9. 2012
Sal	TER3
Tid	14.00-18.00
Kurskod	TDDD37
Provkod	TEN1
Kursnamn/benämning	Database Technology
Institution	*IDA*
Antal uppgifter som ingår i tentamen	8
Antal sidor på tentamen (inkl. försättsbladet)	5
Jour/Kursansvarig	Jose M. Peña
Telefon under skrivtid	0708229596
Besöker salen ca kl.	15.00 och 17.00
Kursadministratör (namn + tfnnr + mailadress)	Madeleine Häger Dahlqvist madeleine.hager.dahlqvist@liu.se 013 282360
Tillåtna hjälpmedel	Ordbok
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
TDD37 Database Technology
January 9, 2012, 14.00-18.00

Help

Dictionary.

Grades

You can get max 30 points. To pass the exam, grade 3, you need 7 and 8 points in the practical and theoretical part of the exam, respectively. For grade 4 and 5, you need 21 and 27 points, respectively.

Questions

During the exam, there exists the possibility of asking questions and clarifications from Jose M. Peña, tel. 013 281651 and Patrick Lambrix, tel. 013 282605. They will also visit the room at 15.00 and 17.00.

Instructions

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. Please, answer in English.

Good luck!

Practical part (14 points)

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

Read the whole exercise before you start.

We want to create a database to store information about the Swedish National Health System. Specifically, we want to store information about doctors, hospitals and patients. We also want to store information about which doctor or team of doctors treated which patient in which hospital. Notice that not only single doctors but also teams of doctors can treat patients. Furthermore, we assume doctors can work in several hospitals and that patients can be treated in several hospitals. Notice that doctors can also be patients. We also want to distinguish between senior and junior doctors. Every junior doctor has a senior doctor as mentor. We want to store who is the mentor of whom.

Your task is to build an EER model that they can use for creating the database. Clearly write down your choices and assumptions in case you find that something in the information above is not clear.

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

Team

id	name	Arena	founded

Player

id	name	position	age

Playing

id	team	player	year	points

team is a foreign key reference to *id* in the table Team.
player is a foreign key reference to *id* in the table Player.

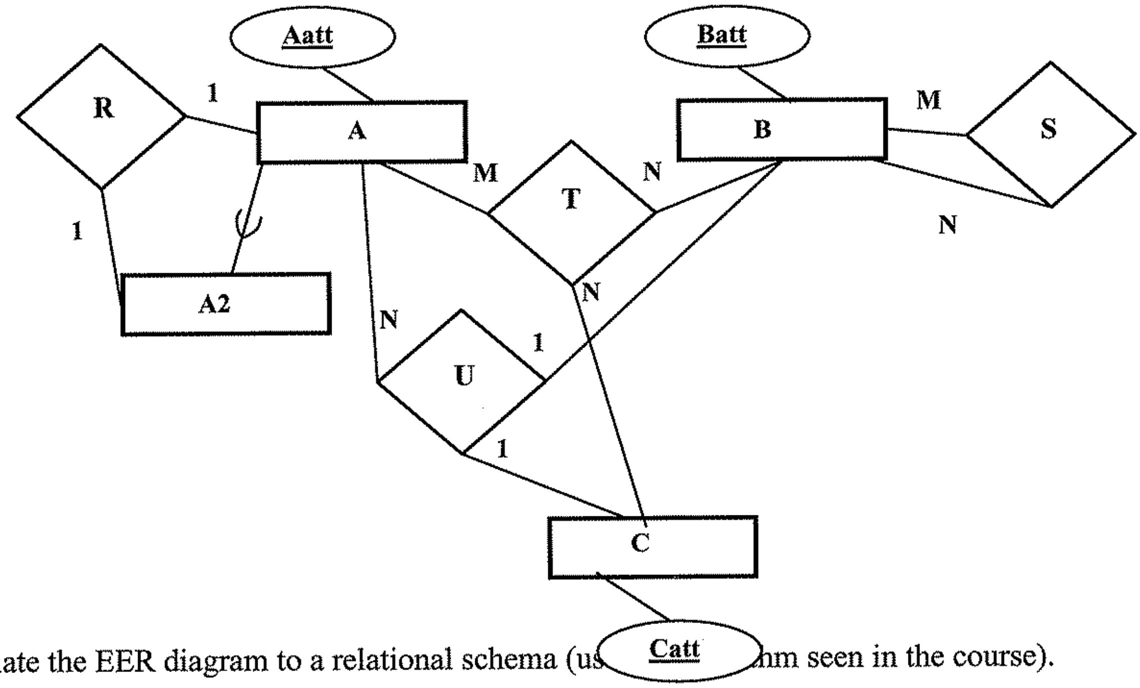
points is the total number of points a player scored for a team in a year.
Note that a player can play for more than one team in the same year.

1. List the names of the teams founded before 1980.

2. List the name of each player that has played for more than one team during the year 2011.

3. For each player, show the first year she played and the total number of points she scored in that year.

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



Translate the EER diagram to a relational schema (us[illegible]hm seen in the course).

Theoretical part (16 points)

Question 4. Normalization (2 p):

Normalize (1NF→2NF→3NF→BCNF) the relation R(A, B, C, D, E, F, G, H) with functional dependencies F={ABC→DEFGH, D→CEF, EF→GH}. *Explain your solution step by step.*

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

We have a file with 30000 records. Each record is 5 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=100 bytes and unspanning allocation. Each index record is 4 bytes long.

1. Calculate the average number of block access needed to find a record with a given value for X when using the primary access method and when using a single level index.
2. Calculate the average number of block access needed to find a record with a given value for Y when using the primary access method and when using a single level index.
3. Explain why you obtain different results in the question 2 depending on whether you use an index or not.

Recall that $\log_2 2^x = x$. That is, $\log_2 1 = 0$, $\log_2 2 = 1$, $\log_2 4 = 2$, $\log_2 8 = 3$, $\log_2 16 = 4$, $\log_2 32 = 5$, $\log_2 64 = 6$, $\log_2 128 = 7$, $\log_2 256 = 8$, $\log_2 512 = 9$, $\log_2 1024 = 10$, $\log_2 2048 = 11$, etc.

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

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

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

2. Does this schedule permit the two-phase locking protocol, i.e. can you apply the protocol so that the transactions interleave as in the schedule above ? Justify your answer.

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 T2
Write-item T2, B, 3, 4
Start-transaction T3
Write-item T3, A, 7, 8
Write-item T3, A, 8, 1
Commit T2
Start-transaction T4
Write-item T4, B, 4, 5
Write-item T4, B, 5, 10
Write-item T3, A, 1, 5
Checkpoint
Start-transaction T1
Commit T3
Write-item T1, C, 8, 9
→system crash

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

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

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

2. Assume that the tables do not contain any NULL value. Assume also that each table contains 1000 tuples and that each attribute is of size 4 byte. Show that the optimized query tree is more efficient than the canonical query tree.

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