Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	18e agusti 2012
Sal	TER3
Tid	8-12
Kurskod	TDDD37
Provkod	TEN1
Kursnamn/benämning	Database technology
Institution	*IDA*
Antal uppgifter som ingår i tentamen	8
Antal sidor på tentamen (inkl. försättsbladet)	6
Jour/Kursansvarig	Jose M. Peña
Telefon under skrivtid	0708229596
Besöker salen ca kl.	10
Kursadministratör (namn + tfnnr + mailadress)	Madeleine Häger Dahlqvist, 013 282360, madeleine.hager.dahlqvist@liu.se
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
TDDD37 Database Technology

August 18, 2012, 8.00-12.00

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

Jose M. Peña will visit the room at 10.00.

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 (15 points)

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

We want to create a database to store information about the last Olympic Games. Specifically, we want to store information about the athletes, in which sport they participated, and whether we won any medal in that sport. Mind that athletes can participate in many sports, individually or in teams. We also want to store the number of gold/silver/bronze medals won by the athletes of each country. Finally, we also want to store which athletes were family (couple, brothers, sisters, etc.).

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):

Consider the following database schema

Country(Name, Code, Capital, Area, Population)
Organization(Name, Abbreviation, Established)
IsMember(Organization, Country, Type)

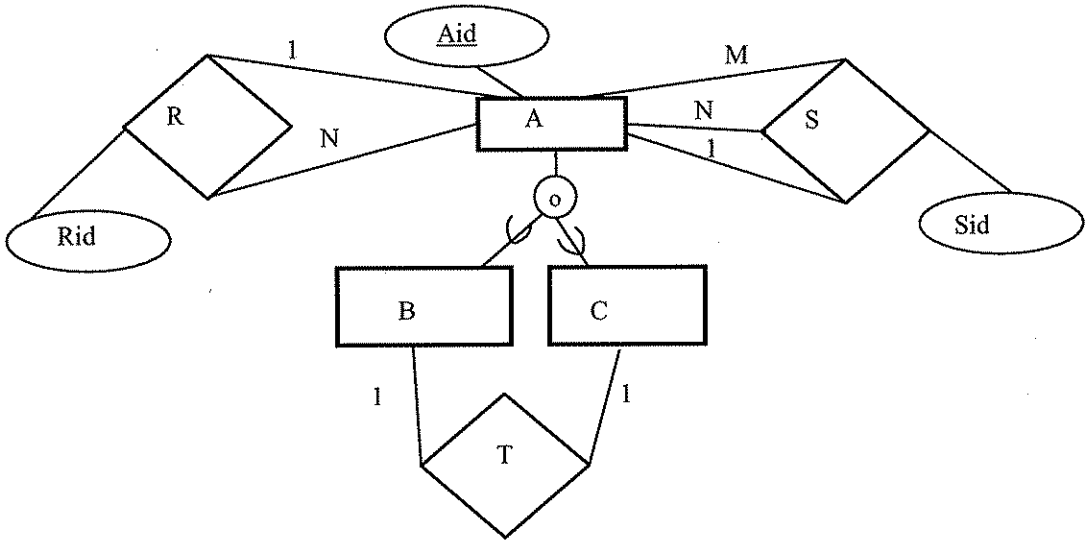
The attribute Organization in the table IsMember is a foreign key reference to Abbreviation in the table Organization.
The attribute Country in table IsMember is a foreign key reference to Code in the table Country.

1. List all the 'EU' member country names.

2. List all the country names which are not a member of any organization.

3. Compute the sum of the populations from all the 'UN' and 'EU' member countries. Note that if a country is a member of both organizations, its population should be counted only once.

Question 3. Translation of EER diagram into relational schema (5 p):



Translate the EER diagram to a relational schema (use the algorithm seen in the course).

Theoretical part (18 points)

Question 4. Normalization (2 + 1 = 3 p):

1. Normalize (1NF→2NF→3NF→BCNF) the relation R(A, B, C, D, E, F, G, H) with functional dependencies F={ABC→DEFGH, DEF→ABC, DE→A, G→H}. Explain your solution step by step. Bear in mind that a relation can have several candidate keys.
2. Do we always have to normalize every relation ? Explain why your answer is yes or no.

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

We have a file with 1000000 records. Each record is 8 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=4000 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 (1) the primary access method and (2) 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 (1) the primary access method, (2) a single level index and (3) static multi-level index.

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

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

Consider the following schedule:

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(z)		
z=z+1		
write(z)		
		read(y)
		y=y+1
		write(y)

1. Is the schedule serializable? Justify your claim.

2. Does this schedule permits 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 (2 + 1 + 1 = 4 p):

1. Apply the deferred update recovery method 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
   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
   Write-item T1, C, 8, 9
   Commit T4
   →system crash

2. Which database recovery strategy does not need the after images, i.e. the new value assigned to a variable ?
3. What happens if there is a system crash while you are recovering from a previous system crash ? How would you solve it ?

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

1. Let R(<u>A</u>,X), S(<u>B</u>, Y), and T(A, B, C) be three tables with the underlined attributes as keys. Optimize the following MySQL query:

   ```
   SELECT R.A, S.B
   FROM R, S, T
   WHERE R.A = T.A AND S.B = T.B AND T.C > 90;
   ```

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 ?