



Försättsblad till skriftlig tentamen vid Linköpings Universitet

(fylls i av ansvarig)

Datum för tentamen	28 maj 2009
Sal	Kåra och T1
Tid	14-18
Kurskod	TDDD12 och Tddb48
Provkod	TEN1
Kursnamn/benämning	Databasteknik
Institution	<i>IDA</i>
Antal uppgifter som ingår i tentamen	7
Antal sidor på tentamen (inkl. försättsbladet)	5
Jour/Kursansvarig	Jose M. Peña
Telefon under skrivtid	0708229596
Besöker salen ca kl.	Lena Strömbäck kl. 15 och Jose M. Peña kl. 17
Kursadministratör (namn + tfnr + mailadress)	Madeleine Häger Dahlqvist, 013-282360, madha@ida.liu.se
Tillåtna hjälpmedel	Lexikon, miniräknare
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	

LiTH, Linköpings tekniska högskola
IDA, Institutionen för datavetenskap
Jose M. Peña
2009-05-19

TENTAMEN

TDDD12 Databasteknik

TDDB48 Databasteknik

28 maj 2009, kl 14-18

Lokal

KÅRA och T1

Tillåtna hjälpmedel

Lexikon, miniräknare.

Poänggränser

Du kan få maximalt 30 poäng. För att få godkänt, betyg 3, krävs minst 7,5 poäng i respektive tentamensdel (Praktik och Teori). För betygen 4 och 5 krävs totalt 21 respektive 27 poäng.

Lärarjour

Under tentamenstiden finns möjlighet att ställa frågor och få förtydliganden från dels Lena Strömbäck, tel. 0709396776, som besöker salen kl. 15, och dels från Jose M. Peña, tel. 0708229596, som besöker salen kl. 17.

Instruktioner

Skriv klart och tydligt. Ge relevanta och motiverade svar på endast det som efterfrågas. Antaganden utöver de som står i uppgiften måste anges. Gjorda antaganden får naturligtvis inte förändra den givna uppgiften. Du kan svara på svenska eller engelska.

Lycka till!

Del 1: Praktik

Uppgift 1. EER-modellering (4 p):

Vi vill skapa en databas för att lagra information om schackspelare och schackmatcher. Databasen ska innehålla information om spelarna. Det finns två spelaretyper: Proffs och amatörer. Databasen ska innehålla hur mycket pengar varje proffsspelare har tjänat. För varje amatörspelare ska det registreras vilken proffsspelare är hans/hennes lärare.

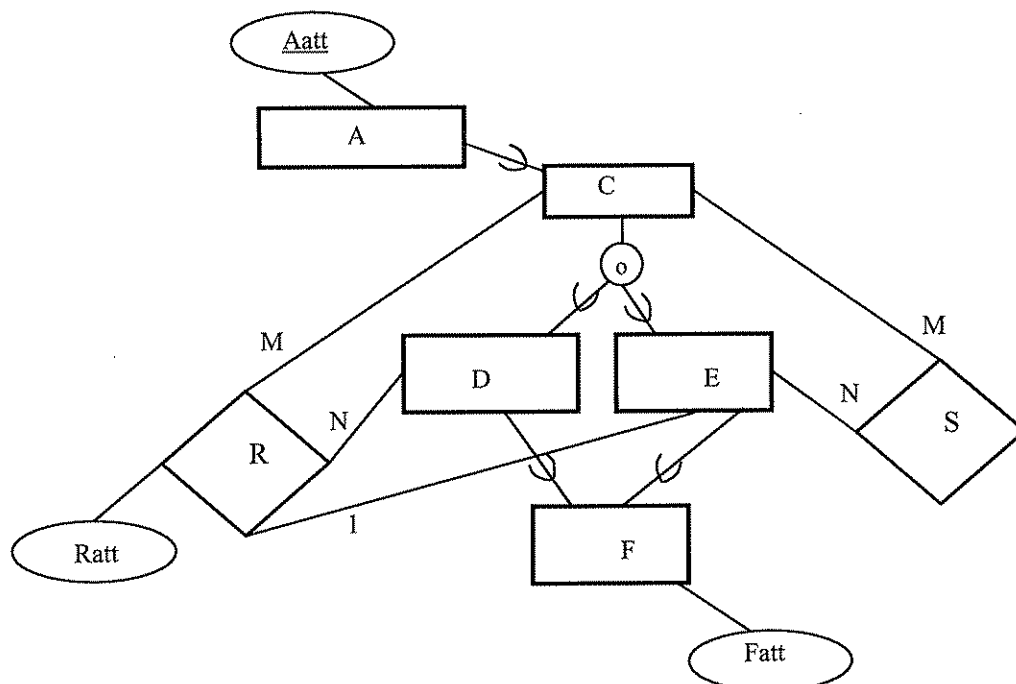
Databasen ska också innehålla information om matcherna. Det ska registreras när matchen skedde, vilka två spelare deltog i matchen, och alla schackdrag i matchen. Ett drag beskrivs av dragets ordningsnummer i matchen plus en kod. Koden följer alltid mönstret "pxy", med vilket menas att pjäsen p flyttas till fältet (x,y). Vi antar att varje spelare spelar max en match per dag.

Skapa en EER-modell för detta. Det räcker med att använda de här fem attributen:

- Spelarens personnummer.
- Pengarna som spelaren har tjänat.
- Matchens datum.
- Dragets ordningsnummer.
- Dragets kod.

Obs !!! Man kan använda andra attribut men en poäng dras av för detta. Motivera val av entiter och relationstyper samt attribut, skriv ned alla antaganden och ange alla kardinaliteter.

Uppgift 2. Översättning till relationer (4 p):



Översätt EER-diagrammet till relationer. Markera primärnycklar med heldragen understrykning och främmande nycklar med streckad understrykning i schemat, med en pekare från den främmande nyckeln till det/de attribut som den främmande nyckeln refererar till.

Uppgift 3. SQL (1 + 2 + 2 = 5p):

Lag

<u>id</u>	namn	arena	grundad
-----------	------	-------	---------

Spelare

<u>id</u>	namn	position	född
-----------	------	----------	------

Spelar

<u>id</u>	<u>lag</u>	spelare	säsong	poäng
-----------	------------	---------	--------	-------

lag i tabellen Spelar är en främmande nyckel som refererar till id i tabellen Lag.
spelare i tabellen Spelare är en främmande nyckel som refererar till id i tabellen Spelare.

En spelare kan endast spela för ett lag i en säsong.
poäng är antalet poäng en spelare får i en säsong.

1. Lista namnet på varje spelare som har spelat för "*LiU Club*".
2. För varje säsong, lista den genomsnittliga åldern på spelarna som spelar för "*LiU Club*".
3. För varje spelare som är yngre än *Magnus Johansson*, lista antalet säsonger de har spelat för "*LiU Club*". (En del yngre spelare har kanske inte spelat för någon säsong ännu !)

Uppgift 4. Normalisering (2p):

Normalizera (1NF → 2NF → 3NF → BCNF) relationen R1(A, B, C, D, E, F) med funktionella beroende (eng. functional dependencies) $F1 = \{A \rightarrow BCDEF, BC \rightarrow A, B \rightarrow DF, D \rightarrow E\}$. *Ange resultat efter varje steg.*

Del 2: Teori

Uppgift 5. Index (4 + 2 + 1 = 7p):

Antag att vi har en datafil har 1,000,000 poster. Filen är sorterad på nyckelfältet *A*. Storleken på varje post (eng. record) är 100 bytes. Ett diskblock har storlek 4096 bytes, och posterna är lagrade obrutna (eng. Unspanning). En indexpost har storlek 10 bytes. Antag att vi har en applikation som hämtar poster slumpmässigt enligt nyckelfältet *B*.

- Designa ett en-nivå-index som ger snabbare åtkomst av data för denna applikation. Gör en skiss av indexet samt beräkna dess storlek och hur många diskaccesser behövs för att hitta en post. (4p)
- Illustrera hur ett fler-nivå-index kan ge snabbare åtkomst till data än ett en-nivå-index för denna applikation. (2p)
- Ett index gör alltid databassökningar snabbare. Sant eller falskt? Varför?

Uppgift 6. Samtidighet (3 + 1 = 4p):

Givet nedanstående transaktioner

TRANSACTION1

```
Read-item(Other-account1);
Read-item(My-account);
Other-account1 = Other-account1 - 2000;
My-account = My-account + 2000;
Write-item(My-account);
Write-item(Other-account1);
```

TRANSACTION2

```
Read-item(Other-account2);
Read-item(My-account);
Other-account2 = Other-account2 + 3000;
Write-item(Other-account2);
My-account = My-account - 3000;
Write-item(My-account);
```

- (1) Ange alla möjliga konflikter, (2) rita ett serialiserbart (men ej seriellt) schema och (3) bevisa att schemat är serialiserbart.
- Kan en deadlock uppstå i ett schema med enbart de två givna transaktioner? Förklara ditt svar.

Uppgift 7. Databasåterställning (3 + 1 = 4p):

- Beskriv metoden för återställning med uppskjuten uppdatering (eng. recovery with deferred update). Använd systemloggen nedan för att exemplifiera metoden. Visa alla operationer som görs vid återställningen av databasen. I rätt ordning!

b) Behöver man lagra all information som lagrats i systemloggen nedan för återställning med uppskjuten uppdatering? Förklara ditt svar.

Part of system log:

Start-transaction T1
Write-item T1, A, 10, 20
Start-transaction T2
Write-item T1, B, 10, 20
Write-item T2, C, 10, 20
Commit T1
Start-transaction T3
Write-item T3, D, 20, 30
Checkpoint
Write-item T2, C, 20, 40
Commit T2
→system crash