

LiTH, Linköpings tekniska högskola
IDA, Institutionen för datavetenskap
Jose M. Peña
2009-08-05

TENTAMEN

TDDD12 Databasteknik
TDDDB48 Databasteknik
TDDC94 Databasteknik

12 augusti 2009, kl 14-18

Lokal

TER1 och TER2

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):

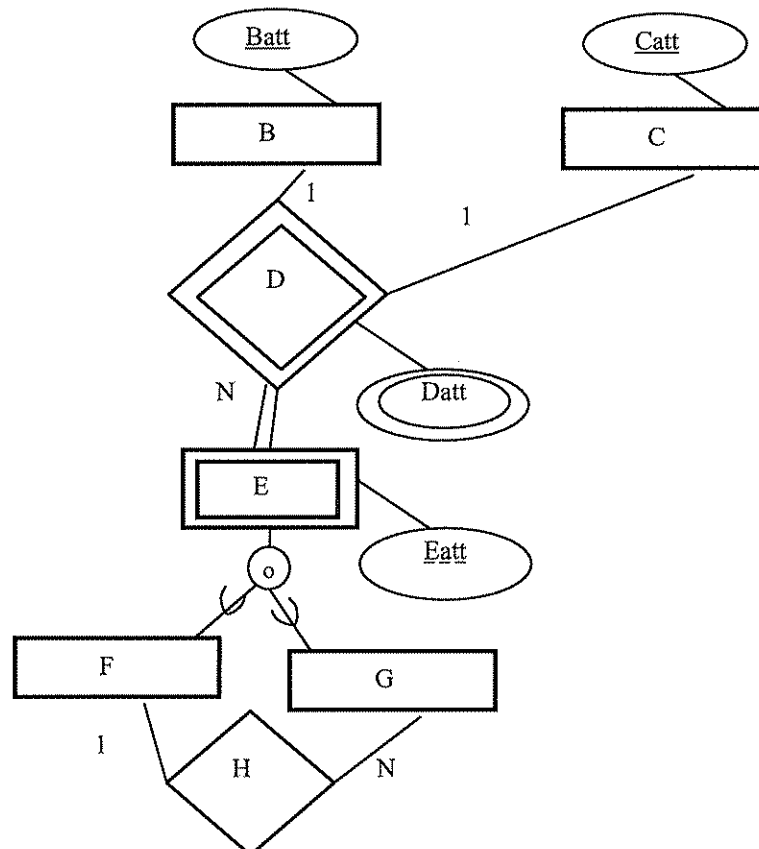
Vi vill skapa en databas för att lagra information om schackspelare och schackmatcher. Spelarna är svenska eller icke-svenska. Databasen ska innehålla information bara om matcherna där minst en av spelarna är svensk. Det ska registreras när matchen skedde, vilka två spelare deltog i matchen, och alla schackdrag i matchen samt alla slagna pjäserna 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. Om ett drag slår en pjäs, vilken pjäs har blivit slagen av draget ska lagras i databasen.

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

- Spelarens personnummer.
- Matchens datum.
- Dragets ordningsnummer.
- Dragets kod.
- Slagen pjäsens namn.

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 + 1 + 2 + 1 = 5p):

Lag

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

Spelare

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

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

En spelare kan endast spela för ett lag under en säsong.

poäng är antalet poäng en spelare får under en säsong.

1. Lista alla lag grundade före "*LiU Club*".
2. Lista den genomsnittliga åldern på spelarna som har spelat för "*LiU Club*" under säsongen 08/09.
3. För varje lag, lista den genomsnittliga åldern på spelarna som har spelat under säsongen 08/09.
4. För varje spelare som är yngre än *Magnus Johansson*, lista antalet säsonger han har spelat för "*LiU Club*". (En del yngre spelare har kanske inte spelat under någon säsong ännu !)

Uppgift 4. Normalisering (2p):

Normalizera (1NF→2NF→3NF→BCNF) relationen R(A, B, C, D, E, F) med funktionella beroende (eng. functional dependencies) $F = \{C \rightarrow A, A \rightarrow BCD, BE \rightarrow F\}$. Ange resultat efter varje steg.

Del 2: Teori

Uppgift 5. Index (4 + 2 = 6p):

Antag att 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*.

a) Designa ett en-nivå-index som ger snabbare åtkomst av data för denna tillämpning. Gör en skiss av indexet samt beräkna dess storlek och hur många blockaccesser behövs för att hitta en post.

b) Illustrera hur ett fler-nivå-index kan ge snabbare åtkomst till data än ett en-nivå-index för denna applikation. Gör en skiss av indexet samt beräkna dess storlek och hur många blockaccesser behövs för att hitta en post.

Uppgift 6. Samtidighet (1 + 1 + 2 + 1 = 5p):

- a) Inom databasområdet kräver man att en transaktion ska uppfylla ACID. Förklara kortfattat vad detta innebär och vilka egenskaper transaktionen måste ha.
- b) Studera följande transaktionsschema:

T1: read(A)
T1: A:=A+100
T2: read(A)
T2: A:=A-500
T2: write(A)
T1: write(A)

Varför uppfyller inte detta schema kraven för ACID? Ge minst ett exempel på problem som uppstår med utgångspunkt från egenskaperna i ACID.

- c) Förklara problemet med transaktionen i b genom att använda konfliktserialiserbarhet.
- d) Ge ett annat transaktionsschema för transaktionerna i b som uppfyller kraven på konfliktserialiserbarhet.

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

- a) Beskriv metoden för återställning med omedelbar uppdatering (eng. recovery with immediate update) som fungerar för systemloggen nedan. Använd systemloggen nedan för att exemplifiera metoden. Visa alla operationer som görs vid återställningen av databasen. I rätt ordning!
- b) Ger användningen av kontrollpunkter (eng. checkpoints) någon fördel i denna metod för återställning? Förklara ditt svar.

Part of system log:

Start-transaction T1
Write-item T1, A, 10
Start-transaction T2
Write-item T1, B, 10

Write-item T2, C, 10
Commit T1
Start-transaction T3
Write-item T3, D, 20
Write-item T2, C, 20
Commit T2
→system crash