

Databasesystemer

IT Universitetet i København

16. januar 2006

Eksamenssættet består af 5 opgaver med 16 spørgsmål, fordelt på 6 sider (inklusive denne side), *samt et svarark, hvor visse spørgsmål skal besvares.*

Vægten af hver opgave er angivet. Du har 4 timer til at besvare alle spørgsmål. Hvis du ikke er i stand til at give et fuldt svar på et spørgsmål, så prøv at give et delvist svar. Du kan vælge at skrive på engelsk eller dansk (evt. med engelske termer).

Siderne i besvarelsen skal være nummererede, og forsynet med navn, CPR nummer og kursuskode (DBS). Skriv kun på forsiden af arkene, og sortér dem inden nummereringen, så opgaverne forekommer i nummerrækkefølge.

“MDM” refererer i sættet til kursusbogen “Modern Database Management 7th edition” af Jeffery A. Hoffer, Mary B. Prescott and Fred R. McFadden.

Alle skriftlige hjælpemidler er tilladt.

1 Databasemodellering (30%)

Den akademiske verden er et interessant eksempel på internationalt samarbejde og udveksling. Denne opgave omhandler modellering af data om forskere, akademiske institutioner, og samarbejde mellem forskere. En forsker kan være ansat som VIP eller forskningsassistent. Der er tre typer af VIP: Adjunkter, lektorer og professorer. Følgende skal gemmes i databases:

- For hver forsker, hans/hendes navn, fødselsår, og aktuelle stilling (hvis den findes).
- For hver institution, dens navn, land, og indvielsesår, samt dens institutter (*f.eks. juridisk institut, datalogisk institut,...*). Et institut hører til præcis én institution.
- En ansættelsehistorie, der indeholder information om alle ansættelser (start- og slutdato, stilling, og på hvilket institut).
- Information om medforfatterskab, dvs. hvilke forskere der har skrevet en forskningsartikel sammen. For hver fælles artikel, skal titlen gemmes.
- For hver forsker skal der gemmes information om hans/hendes højeste akademiske grad (Bsc, MSc eller PhD), herunder hvem der var vejleder, og på hvilket institut.
- For hver VIP skal gemmes information om hvilke forskningsprojekter han/hun er involveret i (titel, startdato og slutdato), samt hvor mange penge han/hun har fået som hovedansøger i forskningsprojekter.

a) Tegn et EER diagram for den beskrevne database. Husk at indikere kardinaliteter på relationships. EER diagrammet bør ikke indeholde redundante entitetsmængder, relationships eller attributter. Det skal tilstræbes at bruge den modellering, der mest præcist afspejler data (brug f.eks. supertype/subtype relationships når det er relevant). Hvis du gør nogen antagelser om data, skal de beskrives i svaret.

b) Konverter EER diagrammet fra spørgsmål a) til relationer, og skriv SQL sætninger der skaber relationerne. Du kan lave et hvilket som helst rimeligt valg af datatyper. Husk at inkludere constraints, der følger af beskrivelsen af data eller dit EER diagram, inklusiv primærnøgle og fremmednøgle constraints.

2 Normalisering (15%)

Vi betragter følgende relation:

`Articles(ID, title, journal, issue, year, startpage, endpage, TR-ID)`

Relationen indeholder information om artikler publiceret i videnskabelige journaler. Hver artikel har et unikt ID, og information om, hvor den kan findes (navn på journal, hvilket nummer (issue), og på hvilke sider). Hvis resultater i en artikel tidligere er publiceret i en "teknisk rapport" (TR), kan et ID for denne tekniske rapport specificeres. Vi har følgende information om attributværdierne:

- For hver journal er et helt nummer altid publiceret indenfor et enkelt år (selv om det evt. har flere dele).
- Værdien af `endpage` er aldrig mindre end værdien af `startpage`.
- Der er aldrig dele af mere end én artikel på en enkelt side.

Følgende er en instans af relationen:

ID	title	journal	issue	year	startpage	endpage	TR-ID
42	Cuckoo Hashing	JAlg	51	2004	121	133	87
33	Deterministic Dictionaries	JAlg	41	2001	69	85	62
33	Deterministic Dictionaries	JAlg	41	2001	69	85	56
39	Dictionaries in less space	SICOMP	31	2001	111	133	47
57	P vs NP resolved	JACM	51	2008	1	3	99
77	What Gödel missed	SICOMP	51	2008	1	5	98
78	What Gödel missed	Nature	2222	2008	22	22	98

a) Baseret på ovenstående, indikér for hver af følgende mængder af attributter hvorvidt det er en nøgle for `Articles` eller ej. Brug svararket til dit svar.

1. {ID};
2. {ID, TR-ID};
3. {ID, title, TR-ID}
4. {title};
5. {title, year};
6. {startpage, journal, issue}

Du kan, hvis du vil, desuden skrive en kort forklaring til hvert svar. Denne vil blive taget i betragtning ved bedømmelsen, men er ikke nødvendig for at få fulde point.

b) Baseret på ovenstående, indikér for hver af følgende potentielle funktionelle afhængigheder (FDs), hvorvidt der er tale om en funktionel afhængighed eller ej. Brug svararket til dit svar.

1. ID \rightarrow title;
2. startpage \rightarrow endpage;
3. journal issue \rightarrow year
4. title \rightarrow ID;
5. ID \rightarrow startpage endpage journal issue;
6. TR-ID \rightarrow ID

Du kan, hvis du vil, desuden skrive en kort forklaring til hvert svar. Denne vil blive taget i betragtning ved bedømmelsen, men er ikke nødvendig for at få fulde point.

c) Foretag, baseret på a) og b), normalisering til BCNF, og angiv de resulterende relationer.

3 SQL forespørgsler (30%)

Vi betragter igen `Articles` fra opgave 2.

a) Indikér for hver af følgende udtryk hvorvidt det er et gyldigt SQL udtryk eller ej. Et gyldigt udtryk, som beskrevet i MDM, skal accepteres af en standard SQL fortolker, men et ugyldigt udtryk skal resultere i en fejlmeddelelse. Brug svararket til dit svar.

1. `SELECT * FROM Articles WHERE endpage-startpage>10;`
2. `SELECT * FROM Articles WHERE endpage-startpage<0;`
3. `SELECT SUM(name) FROM Articles;`
4. `SELECT AVG(year) FROM Articles WHERE title LIKE 'C%';`
5. `SELECT COUNT(*) FROM Articles GROUP BY year;`
6. `SELECT year,COUNT(*) FROM Articles WHERE COUNT(*)>10 GROUP BY year;`

b) Indikér for hver af følgende SQL forespørgsler, hvor mange tupler der ville blive returneret som resultat, hvis den blev kørt på instansen af `Articles` fra opgave 2. Brug svararket til dit svar.

1. `SELECT ID FROM Articles WHERE year<2006;`
2. `SELECT DISTINCT ID FROM Articles WHERE year<2006;`
3. `SELECT AVG(year) FROM Articles GROUP BY journal;`
4. `SELECT ID FROM Articles WHERE title LIKE '%d';`

Betragt relationerne `Authors(auID,name)` og `Authoring(articleID,authorID)`, der henholdsvis indeholder information om forfatternavne og forfattere til hver artikel.

c) Skriv en SQL forespørgsel der for hver artikel returnerer dens ID, titel, og antallet af forfattere.

d) Skriv en SQL forespørgsel der returnerer titlerne på artikler, der har 'Robert Tarjan' som forfatter.

e) Skriv en SQL forespørgsel der returnerer antallet af medforfattere til 'Robert Tarjan'. (Dvs. antallet af forfattere, der har skrevet mindst én artikel sammen med ham.)

4 Effektivitet og transaktioner (15%)

Betragt følgende seks forespørgsler på `Articles` fra opgave 2:

1. `SELECT title FROM Articles WHERE year=2005;`
2. `SELECT title FROM Articles WHERE endpage=100;`
3. `SELECT title FROM Articles WHERE year>1995 AND year<2000;`
4. `SELECT title FROM Articles WHERE journal='JACM' AND issue=55;`
5. `SELECT title FROM Articles WHERE issue=55 AND journal='JACM';`
6. `SELECT title FROM Articles WHERE endpage-startpage>50;`

a) Indikér hvilke af ovenstående forespørgsler, der sandsynligvis (baseret på din viden fra kurset) ville blive udført hurtigere, hvis *alle* følgende indekser blev oprettet. Brug svararket til dine svar.

```
CREATE INDEX Idx1 ON Articles(year,startpage);
CREATE INDEX Idx2 ON Articles(startpage,endpage);
CREATE INDEX Idx3 ON Articles(journal,issue,year);
```

Vi betragter nu nedenstående transaktioner på relationen `Authors(auID,name)`.

Time	User A	User B
1	<code>INSERT INTO Authors VALUES (42,'Donald Knuth');</code>	
2		<code>INSERT INTO Authors VALUES (43,'Guy Threeewood');</code>
3		<code>DELETE FROM Authors WHERE name LIKE 'Don%';</code>
4		<code>INSERT INTO Authors VALUES (44,'Donald E. Knuth');</code>
5	<code>DELETE FROM Authors WHERE name LIKE 'Guy%';</code>	
6	<code>COMMIT;</code>	
7		<code>COMMIT;</code>

b) Antag at `Authors` ikke indeholder nogen tupler, at transaktionerne kører på isoleringsniveau `READ COMMITTED`, og at de enkelte kommandoer udføres i den rækkefølge, der er vist ovenfor. Hvad er indholdet af `Authors` efter at begge transaktioner er fuldført?

c) Antag igen at `Authors` ikke indeholder nogen tupler. Hvad er det mulige indhold af `Authors` efter hver *seriel* udførelse af de to transaktioner?

5 Constraints (10%)

Antag at `Authoring` relationen fra opgave 3 blev oprettet som følger:

```
CREATE TABLE Authoring(  
  articleID INT REFERENCES Article(ID) ON DELETE SET NULL,  
  authorID INT REFERENCES Author(ID) ON DELETE CASCADE  
)
```

a) Indikér hvilke af følgende udsagn er sande, og hvilke er usande. Brug svararket til dine svar.

1. Hvis vi forsøger at slette en tupel fra `Authoring`, sker dette ikke. I stedet sættes værdien af `articleID` i tuplen til `NULL`.
2. Hvis vi sletter en tupel fra `Authoring`, så slettes også alle tupler i `Author` som den slettede tupel refererer til.
3. Hvis vi sletter en tupel fra `Article`, kan nogle attributter fra `Authoring` få deres værdi ændret til `NULL`.
4. Hvis vi forsøger at indsætte et tupel i `Author` med et `ID`, som der ikke refereres til i `Authoring`, så forkastes operationen (abort).
5. Hvis vi forsøger at indsætte et tupel i `Authoring` med et `ID`, som ikke eksisterer i `Author`, så forkastes operationen (abort).

b) Skriv `CHECK` constraints til `Articles` fra opgave 2 som sikrer at følgende er overholdt:

1. Værdier af `journal` attributten starter *ikke* med 'Journal'.
2. Værdien af `endpage` er aldrig mindre end værdien af `startpage`.
3. Værdien af `year` er givet i fuld længde (f.eks. forkortes 1999 ikke til 99). Du kan antage at `year` er af typen `int`, dvs. et heltal, og at ingen artikler er mere end 200 år gamle.

Svarark (afleveres)

Navn		Sidenummer	
CPR		Totalt sidetal	

Instruktioner. I alle spørgsmål, bortset fra 3.b (der spørger om tal), skal du sætte præcis ét X i hver søjle. Bemærk at retningen vil blive gjort på en måde, så tilfældige svar ikke betaler sig. For eksempel vil to korrekte svar og et forkert svar give det samme antal point som ét korrekt svar og to spørgsmålstegn.

Spørgsmål 2.a	1	2	3	4	5	6
Nøgle						
Ikke nøgle						
?						

Spørgsmål 2.b	1	2	3	4	5	6
FD						
Ikke FD						
?						

Spørgsmål 3.a	1	2	3	4	5	6
Gyldig						
Ugyldig						
?						

Spørgsmål 3.b	1	2	3	4
Antal tupler				

Spørgsmål 4.a	1	2	3	4	5	6
Hurtigere						
Samme						
?						

Spørgsmål 5.a	1	2	3	4	5
Sandt					
Usandt					
?					