UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF 212 — Databaseteori
Eksamensdag:	31. mai 2002
Tid for eksamen:	9.00 – 15.00
Oppgavesettet er på 7 sider.	
Vedlegg:	1 - Enkel grammatikk for parsing av spørsmål
Tillatte hjelpemidler:	Kalkulator

Kontroller at oppgavesettet er komplett før
du begynner å besvare spørsmålene.

Oppgave 1 Relasjonsalgebrateori

Klassisk relasjonsalgebra opererer, i tillegg til ρ (renavning, som burde ha vært med på listen, og som vi skal regne som den sjette), med fem fundamentale operatorer; de to første er unære, de tre siste binære (vi bruker her sett-versjonene av operatorene, ikke bag-versjonene):

- σ – seleksjon
- π – projeksjon
- $\cup$ – union
- – eller $\setminus$ – differans
- $\times$ – kartesisk produkt

1a Ikke-fundamentale operatorer

De tre viktigste ikke-fundamentale relasjonsoperatorene er $\cap$ (snitt), $\overset{\bowtie}{\theta}$ (θ-join) og $\bowtie$ (naturlig join). Vis hvordan du kan uttrykke hver av disse med de fem fundamentale operatorene.

1b Fundamentale operatorer

Forklar hvorfor de fem fundamentale operatorene utgjør et minimalt sett, det vil si at ingen av dem kan uttrykkes ved de andre fire.

Som en starthjelp skal vi her gi beviset for $\times$ (kartesisk produkt); de andre fire må du klare selv:

Siden $\times$ er den eneste av de fem som produserer et svar med flere attributter enn det operandene har, kan ikke $\times$ uttrykkes ved noen kombinasjon av $\sigma, \pi, \cup$ og $\setminus$.

(Fortsettes på side 2.)

1c En kommutativ lov

Hvilke betingelser må være oppfylt for at seleksjon og projeksjon skal kommutere (resultatet blir det samme selv om operatorene bytter rekkefølge)?

Mer presist: La R være en relasjon, la X være en komponent i R, og la θ være et seleksjonspredikat over R. Oppgi en nødvendig og tilstrekkelig betingelse for at $\pi_X(\sigma_\theta(R)) = \sigma_\theta(\pi_X(R))$. Svaret skal være begrunnet.

Oppgave 2 Implementasjon

Du skal nå vise hvordan forskjellige komponenter i en implementasjon av et databasesystem kan fungere. Vi skal her bare bruke følgende tabeller (primærnøkler er angitt med **fete** typer, andre nøkler er understreket og fremmednøkler fremgår av attributtnavnene):

PERSON(**personID**, kjønn, fornavn, mellomnavn, etternavn)
FILM(**filmID**, tittel, produksjonsår)
DELTAKELSE(**pID**, personID, funksjon, filmID)

Mot denne delen av databasen stiller vi spørsmålet (finn fornavn og etternavn på alle mannlige skuespillere som spilte i filmer produsert i 1999):

```
SELECT PERSON.fornavn, PERSON.etternavn
FROM PERSON, FILM, DELTAKELSE
WHERE PERSON.personID = DELTAKELSE.personID AND
      FILM.filmID = DELTAKELSE.filmID AND
      FILM.produksjonsår = ‘‘1999’’ AND
      PERSON.kjønn = ‘‘M’’ AND
      DELTAKELSE.funksjon = ‘‘cast’’
```

Databasen har “clustered” indekser på primærnøklene. I tillegg er det indekser på attributtet produksjonsår i FILM og attributtene personID og filmID i DELTAKELSE.

2a Parsing

i. Bruk den enkle grammatikken i vedlegg 1 til å lage et parsetre for spørsmålet over.

ii. Hvilke(n) hovedoppgave(r) har preprosessoren?

2b Logisk spørreplan

Konverter parsetreet fra forrige oppgave til en logisk spørreplan i relasjonsalgebra (tegn uttrykkstreet).

2c Optimering

i. Hvilke regler benyttes ofte (gir oftest stor ytelsesgevinst) for optimering av logiske spørreplaner?

ii. Optimer den logiske spørreplanen i deloppgave 2b (tegn det nye uttrykkstreet).

(Fortsettes på side 3.)

2d Eksekvering

Hvis datamengden er så stor at vi ikke kan holde hele datasettet i minnet samtidig, baserer vi eksekveringen av (full-relasjons) operatorene på to grunnprinsipper.

i. Hvilke er disse, og hvordan fungerer de i korte trekk?

ii. I hvilke tilfeller vil du bruke de forskjellige metodene? Begrunn svaret.

2e Datalagring

Anta at vi har et lagringssystem som bruker en disk med følgende spesifikasjoner:

Diskplater:	4
Spor:	10.000
Antall sektorer per spor:	500 (en ikke-sonet disk)
Byte per sektor:	512
Byte per "gap":	32
Gjennomsnittlig søketid:	5 ms
Spor-spor søk:	0,5 ms
Rotasjonshastighet:	15.000 RPM

Anta videre at hver relasjon er lagret samlet ("clustered") på disken. Databasen inneholder 100.000 personer, 100.000 filmer og deltakelse har 10.000.000 tupler. Hver blokk har et hode ("header") på 20 byte, hver post ("record") har et hode på 10 byte og hvert attributt trenger antall byte som følger:

PERSON:		FILM:		DELTAKELSE:	
personID	(4)	filmID	(4)	pID	(4)
kjønn	(1)	tittel	(50)	personID	(4)
fornavn	(20)	produksjonsår	(4)	funksjon	(20)
mellomnavn	(20)			filmID	(4)
etternavn	(20)				

i. Hva er den utnyttbare kapasitet på disken hvis platene brukes på begge sider?

ii. Hvilke faktorer inngår i det å aksessere en blokk på disken, og hva er gjennomsnittlig aksesstid for en vilkårlig 4 Kbyte blokk?

iii. Hvor stor plass trenger disse relasjonene på disken hvis hver post ("record") ikke er delt over flere blokker ("unspanned" lagring)?

iv. Hvor lang tid tar det å lese hele relasjonen DELTAKELSE (uavbrutt) hvis vi antar vilkårlig plassering av data i diskblokker på disken?

v. Hvordan kan man optimere en slik lesing av en relasjon ved å forandre blokkplassering, og hva blir den nye tiden for å lese hele DELTAKELSE (skriv ned eventuelle antagelser du gjør)?

(Fortsettes på side 4.)

2f Indekser

Anta at indeksene på tabellene er tette ("dense"), at de er implementert ved hjelp av B^+-trær, at rotnoden er holdt i minnet ("pinned"), at en peker tar 4 byte, og at en node i gjennomsnitt er 65 % full. Hvor mange diskaksesser spares i gjennomsnitt for å finne en PERSON med en gitt personID ved å bruke indeksen i forhold til å søke i selve relasjonene (lese hvert PERSON-tuppel)?

2g Overføring av mellomresultater mellom operatorer

Data kan sendes mellom to operatorer på hovedsaklig to forkjellige måter. Hvilke er disse, hva er forskjellen på dem, og hvilke fordeler/ulemper har de?

(Fortsettes på side 5.)

Oppgave 3 Transaksjoner

3a Egenskaper

En transaksjon kjennetegnes ved fire egenskaper eller krav. Hvilke er disse, og hva innebærer hver av dem?

3b Konflikter

i. Hva betyr det at to operasjoner er i konflikt med hverandre?

ii. Hva vil det si at to historier er konflikt-ekvivalente?

3c Serialiserbarhet

Anta at vi har tre transaksjoner T1, T2 og T3. Er følgende transaksjonshistorie serialiserbar (begrunn svaret)?

```
              T1              T2              T3

    |       begin1          begin2          begin3
    |
    |       read1(X)
    |       read1(Y)
  T |
  I |                       write2(X)
  D |
    |                                       write3(Y)
    |
    |                       read2(Y)
    |
    V       commit1         commit2         commit3
```

3d Låsing

Låsing brukes ofte for å sikre serialiserbarhet. Hvilken metode brukes oftest i praksis, og hvordan fungerer denne?

(Fortsettes på side 6.)

Oppgave 4 Spørring med SQL

I tillegg til datastrukturen brukt ovenfor, legger vi inn følgende tabell:

ROLLE(**rolleID**, **rollenavn**)

hvor rolleID er fremmednøkkel til DELTAKELSE, og hvor rollenavn er navnet på en rolle skuespilleren har spilt i filmen. Normalt skal følgende to regler gjelde:

- Alle forekomster i ROLLE skal peke på en forekomst i DELTAKELSE hvor funksjon har verdien "*cast*".
- Alle forekomster i DELTAKELSE hvor funksjon har verdien "*cast*", skal være pekt på av minst en forekomst i ROLLE.

Spørsmålene nedenfor skal alle besvares med SQL-spørringer.

4a Integritetskontroll

Lag to lister med brudd på reglene ovenfor (en liste for hver regel). I tillegg til primærnøkkelen i de postene som bryter regelen, skal filmens tittel og skuespillerens for- og etternavn skrives ut.

4b Statistikk-1

I hvor mange prosent av filmene finner vi regissøren på rollelisten?

4c Statistikk-2

Finn gjennomsnittet av antall roller flerrollehaverne i filmene har. (Du skal altså se bort fra alle de tilfellene hvor *en* skuespiller bare har en rolle i en film.)

4d Populære skuespillere

Lag en liste over skuespillere som har spilt i alle filmene til en regissør som har regissert minst 10 filmer. Listen skal inneholde for- og etternavn på både skuespiller og regissør.

Slutt på oppgavesettet
Lykke til!

Ragnar Normann & Pål Halvorsen

(Fortsettes på side 7.)

Vedlegg 1 - Grammatikk for parsing av spørsmål

```
<query> ::= <SFW>

<SFW> ::= SELECT <selList> FROM <fromList> WHERE <condition>

<selList> ::= <attribute>
<selList> ::= <attribute>, <selList>

<fromList> ::= <relation>
<fromList> ::= <relation, <fromList>

<condition> ::= <condition> AND <condition>
<condition> ::= <attribute> = <attribute>
<condition> ::= <attribute> = <pattern>
```

Elementære syntaktiske kategorier som <attribute>, <relation> og <pattern> har ingen regler, men oversettes henholdsvis med navnet på attributtet, navnet på relasjonen eller en streng i anførselstegn.