

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i : IN 212 — Databaseteori
Eksamensdag : Fredag 7. juni 1996
Tid for eksamen : 09.00 - 15.00
Oppgavesettet er på : 6 sider
Vedlegg : Ingen
Tillatte hjelpemidler : Alle trykte og skrevne

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

Les gjerne i gjennom hele settet før du begynner med besvarelsen.

Oppgave 1 (30 %)

Nedenfor følger (deler av) det begrepsmessige skjema for en relasjonsdatabase for registrering av pasienter og laboratorieprøver ved et sykehus:

PASIENT (PasNr, Navn, Adr, AvdKode)
AVDELING (AvdKode, AvdNavn, AntPas)
LAB_PRØVE (PrøveNr, PasNr, Dato, AvdKode, Prøvetaker, Type, Resultat)
ANSATT (AnsNr, AnsNavn, AnsAdr)

AntPas (antall pasienter) har domene heltall. Dato har domene dato. Alle andre attributter har domene tekst. Eksempler på verdi for AvdKode er 'HUD' for hudavdelingen og 'LUNGE' for lungeavdelingen. En pasient kan ta laboratorieprøver på andre avdelinger enn den han eller hun ligger på.

PasNr er primærnøkkel i PASIENT
AvdKode er primærnøkkel i AVDELING
(PrøveNr, PasNr) er primærnøkkel i LAB_PRØVE
AnsNr er primærnøkkel i ANSATT

AvdKode i PASIENT og LAB_PRØVE er fremmednøkler til AVDELING
PasNr i LAB_PRØVE er fremmednøkkel til PASIENT
Prøvetaker i LAB_PRØVE er fremmednøkkel til ANSATT

Oppg. 1A Bruk SQL til å finne pasientnummer og navn på alle pasienter ved hud- eller lungeavdelingen. Skriv også ut navn på avdelingen.

Oppg. 1B Bruk SQL til å finne hvilke pasienter (angitt ved PasNr) som har blitt tatt prøve av hvilke prøvetakere (angitt ved AnsNr) på hvilke datoer. Tabellen skal være sortert på prøvetaker med hensyn på antall prøver de har tatt, i avtakende rekkefølge. Dernest skal tabellen være sortert på dato i stigende rekkefølge.

Oppgave 1 fortsetter på neste side

Oppg. 1C Skriv en skisse av to prosedyrer (rutiner, metoder) i fritt valgt pseudokode med innbakt (embedded) SQL som brukes til å:

- (1) registrere nye pasienter (kall prosedyren NyPasient, du kan anta at en ny pasient ikke er registrert tidligere)
- (2) fjerne pasienter (kall prosedyren FjernPasient, pasienter fjernes på grunnlag av oppgitt pasientnummer)

Informasjonen i LAB_PRØVE påvirkes ikke. Du kan bruke ACCEPT(X) for å lese en verdi fra tastaturet inn i programvariabelen X og DISPLAY(X) for å skrive verdien av X til skjermen. DISPLAY('XXX') skriver den konstante teksten XXX til skjermen. NEW_PASNR(X) tilordner variabelen X en ny, unik verdi av PasNr.

Oppg. 1D Lag et objekt-orientert alternativ til skjemaet på side 1. Skriv skjemaet enten i O₂ eller ObjectStore syntaks. Inkluder også prosedyrene (metodene) i oppgave 1 C i skjemaet, men du skal ikke skrive implementasjonen, bare angi navnet og parametrene.

Vi vil utvide sykehus-databasen til å kunne holde informasjon om flere opphold for en pasient. Dette kan oppnås ved for eksempel å utvide PASIENT-relasjonen som angitt under alternativene (a) og (b) som følger.

(a) PASIENT (PasNr, Opphold, Navn, Adr), PasNr er primærnøkkel

Opphold er et attributt som selv inneholder en relasjon (mengde) som inneholder informasjon om alle oppholdene til en pasient. Attributtene til denne relasjonen er (OppholdNr, AvdKode, InnDato, UtDato). En pasient kan bare legges inn og utskrives én gang på en bestemt dato.

(b) PASIENT (PasNr, OppholdNr, Navn, Adr, AvdKode, InnDato, UtDato)

Alle attributtene i (b) er atomære. (PasNr, OppholdNr) er primærnøkkel

Oppg. 1E Relasjonen (a) er av en bestemt type. Hva kalles slike?

Hvilke FD-er (funksjonelle avhengigheter) er det rimelig å anta gjelder i relasjonen (b)?

Hvilken normalform har (b)?

Navn på pårørende til en pasient ønskes registrert i et tekst-attributt kalt Pårørende som er funksjonelt avhengig av PasNr:

PasNr → Pårørende

Oppg. 1F Lag en utvidelse av skjemaet på side 1 som i tillegg til informasjonen i dette opprinnelige skjemaet også inneholder informasjon om opphold og pårørende. Alle relasjonene skal være på minst tredje normalform (3NF). For hver tabell angi primær- og fremmednøkler. (For å spare skriving trenger du ikke skrive på nytt definisjonene av de relasjonene i det opprinnelige skjemaet du ikke ønsker å endre.)

Oppgave 2 (20 %)

Anta at X , Y , Z og W er delmengder av et relasjonsskjema R . Da gjelder følgende åtte slutningsregler for FD-er og MVD-er (læreboka s. 437 og 438):

- (IR1) Hvis $X \supseteq Y$, så har vi at $X \rightarrow Y$
- (IR2) Hvis $X \rightarrow Y$, så har vi at $XZ \rightarrow YZ$
- (IR3) Hvis $X \rightarrow Y$ og $Y \rightarrow Z$, så har vi at $X \rightarrow Z$
- (IR4) Hvis $X \twoheadrightarrow Y$, så har vi at $X \twoheadrightarrow R - XY$
- (IR5) Hvis $X \twoheadrightarrow Y$ og $W \supseteq Z$, så har vi at $XW \twoheadrightarrow YZ$
- (IR6) Hvis $X \twoheadrightarrow Y$ og $Y \twoheadrightarrow Z$, så har vi at $X \twoheadrightarrow (Z - Y)$
- (IR7) Hvis $X \rightarrow Y$, så har vi at $X \twoheadrightarrow Y$
- (IR8) Hvis $X \twoheadrightarrow Y$, $W \rightarrow Z$, $W \cap Y = \emptyset$ og $Y \supseteq Z$, så har vi at $X \rightarrow Z$.

Benytt de åtte reglene ovenfor (altså IKKE bruk definisjonen av MVD i læreboka s. 436) til å bevise at følgende tre regler også gjelder:

- (IR9) Hvis $X \twoheadrightarrow Y$ og $WY \twoheadrightarrow Z$, så har vi at $WX \twoheadrightarrow (Z - WY)$
- (IR10) Hvis $X \twoheadrightarrow Y$ og $XY \rightarrow Z$, så har vi at $X \rightarrow (Z - Y)$

Kan puffes: (IR11) Hvis $X \twoheadrightarrow Y$ og $X \twoheadrightarrow Z$, så har vi at $X \twoheadrightarrow (Y \cap Z)$,
 $X \twoheadrightarrow (Y - Z)$ og $X \twoheadrightarrow (Z - Y)$

Oppgave 3 (15 %)

Anta at du har en database med følgende tre relasjoner:

BESØKT (Person, Restaurant)
 SERVERER (Restaurant, Rett)
 LIKER (Person, Rett)

Den første er en registrering av hvilke restauranter hver person har besøkt, den andre hvilke retter hver restaurant serverer, og den siste forklarer hvilke retter hver person liker.

Uttrykk i tuppelkalkyle alle punktene A, B og C.

I tillegg løs B i relasjonsalgebra og domenekalkyle

- A) De restauranter som serverer en rett som Ole Olsen liker.
- B) De personer som har besøkt minst en restaurant som serverer minst en rett de liker.
- C) De personer som bare har besøkt restauranter som serverer minst en rett de liker. (Anta at hver person liker minst én rett og har besøkt minst én restaurant.)

Oppgave 4 (25 %)

Flyselskapet Fix skal begynne å fly på norske byer, og systemavdelingen deres skal lage et nytt bestillingssystem. Som datakonsulent skal du kritisere deres forslag til løsninger. De opererer med følgende meget enkle database som vi har godkjent:

RUTE (RuteNr, Fra, Til, Avgangstid)
 AVGANG (RuteNr, Dato, Ant_ledige_seter)
 BESTILLING (RuteNr, Dato, Navn, Ant_seter)

RUTE beskriver hvilke byer Fix flyr ruter fra og til og når flyet starter. AVGANG beskriver hver enkelt avgang på en rute (som bare kjøres én gang per dag) og hvor mange seter det er igjen som ikke er bestilt. BESTILLING beskriver registreringer av godkjente billett-reservasjoner.

Sentralt i oppgaven er låsing av databaser. Vi ser på låsing på tuppel-nivå i AVGANG-tabellen, og definerer at Leselås og Skrivelås tilsvare Shared og Exclusive Locks (s. 557 i læreboka). Vi antar at systemet tillater at en Leselås kan oppgraderes til Skrivelås og at skriveadgang også gir leseadgang. Lås av et tuppel oppnås ved å kalle låsefunksjoner med primærnøkkelen i AVGANG: (RuteNr, Dato) som parametre. En transaksjon venter i denne funksjonen til låsen er oppnådd.

I denne deloppgaven skal løsningen gå på ett, sentralisert anlegg. Systemavdelingen vurderer følgende to alternative bestillingstransaksjoner:

T1

```

"Snakk med kunden og finn ønsket R:= rutenr, D:= dato"
Leselås(R,D);
A:= Antall_ledige_seter i AVGANG(R,D);
"Snakk med kunden og finn ønsket S:= Antall_seter"
IF ( A >= S)
BEGIN
    Skrivelås(R,D);
    Senk Antall_ledige_seter med 'S' i AVGANG(R,D);
    Oppdater BESTILLING;
    FrigiSkrivelås(R,D);
END
ELSE "Si beklager ikke nok plasser ledige på denne avgangen"
FrigiLeselås(R,D);
  
```

T2

Som T1, men 'FrigiLeselås(R,D);' flyttes fra siste linje og settes inn mellom 3. og 4. linje i T1.

Oppgave 4 fortsetter på neste side

Oppg. 4A T1 og T2 kan hver medføre en alvorlig feil. Navngi feilsituasjonene og gi kortfattede beskrivelser med eksempler på hvordan de kan inntreffe.

Oppg. 4B Foreslå en bedre transaksjon T3 ved å gjøre få endringer på T1 eller T2, som ikke gir slike feilsituasjoner som T1 og T2. Hvis du ikke greier å finne en slik transaksjon, kan du nytte transaksjon T3B nedenfor, som ikke er optimal. Prøv uansett å forklare hvorfor T3B bør forbedres.

T3B

```

"Snakk med kunden og finn ønsket R:= rutenr, D:= dato"
Skrivelås(R,D);
A:= Antall_ledige_seter i AVGANG(R,D);
"Snakk med kunden og finn ønsket S:= Antall_seter"
IF ( A >= S)
BEGIN
    Senk Antall_ledige_seter med 'S' i AVGANG(R,D);
    Oppdater BESTILLING;
END
ELSE "Si beklager ikke nok plasser ledige på denne avgangen"
FrigiSkrivelås(R,D);

```

Vi kan nå anta at Fix bruker en riktig transaksjon T3 for ett sentralt anlegg, men nå vil de desentralisere databehandlingen. De kjøper inn databasemaskiner til hver by hvor de har flyvninger fra.

Oppg. 4C Som et første forsøk foreslår systemavdelingen å kopiere hele databasen og løsning forøvrig til hver slik maskin. De vil så la hver maskin lokalt behandle låser og bestillinger uten samarbeid med de andre maskinene. Forklar kortfattet med et eksempel for systemavdelingen hvorfor dette går meget galt.

Systemavdelingen innser nå at de må delvis fragmentere databasen og velger å gjøre det slik at alle maskinene har kopi av hele RUTE, men har horisontalt fragmentert AVGANG slik at hver by har de avganger som starter i vedkommende by. BESTILLINGER registreres og lagres fremdeles lokalt.

Systemavdelingen vurderer nå følgende alternative strategier for håndtering av låser:

- I. Alle låsene kan fremdeles behandles lokalt på det kontoret (den maskinen) som kunden ringer til for å bestille. Ingen kommunikasjon mellom maskinene om låser.
- II. Alle låsene behandles lokalt som I), men det sendes ut melding til alle andre maskiner om hvilket tuppel som låses før låseoperasjonen fullføres.
- III. Som II men i tillegg skal alle låseoperasjoner (både låsing og frigi lås) vente til de får kvittering tilbake som svar fra samtlige maskiner. Hvis en av de andre maskinene også har lås på dette tuppelet, kan det hende at den vil vente med å sende kvittering. Mottaker av en slik låse-melding nytter følgende algoritme for å vurdere venting:

```

IF <jeg ikke har lås ELLER mottatt melding = frigi_lås >
    'returner kvittering nå'
ELSE IF <jeg har leselås OG mottat melding = leselås>
    'returner kvittering nå'
ELSE IF <jeg har skrive-lås eller leselås>
    'returner kvittering først når jeg selv frigir denne låsen'

```

Oppgave 4 fortsetter på neste side

IV. Låsingen følger tuplene i AVGANG slik at hvis et tuppel skal låses, må forespørsel om lås og frigging sendes til den maskinen som har denne avgangen i AVGANG. Låsen administreres av denne maskinen, som da først sender kvittering tilbake når låsen er innvilget, og først da kan den som ba om lås fortsette.

Oppg. 4D Vurder både riktigheten og hensiktsmessigheten av disse fire forslagene og kom med din anbefaling til systemavdelingen i Fix om hvilket av dem de bør bruke. Du kan anta at alle sendte meldinger kommer fram til mottaker innen rimelig tid.

Oppgave 5 (10 %)

Oppg. 5A Finn en minimal overdekning for $F_1 = \{AB \rightarrow C, A \rightarrow B\}$

Oppg. 5B Finn en minimal overdekning for $F_2 = \{AB \rightarrow C, A \rightarrow D, BD \rightarrow C\}$

Slutt på oppgavesettet

Dag Sjøberg

Arne Maus