UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i :	INF3100/INF4100 — Databasesystemer
Eksamensdag :	Tirsdag 8. juni 2004
Tid for eksamen :	09.00 - 12.00
Oppgavesettet er på :	5 sider
Vedlegg :	Ingen
Tillatte hjelpemidler :	Kalkulator

Kontroller at oppgavesettet er komplett
før du begynner å besvare spørsmålene

Det er seks oppgaver

Alle har samme vekt, så beregn ca 30 minutter på hver

Oppgave 1 SQL og relasjonsalgebra

Gitt følgende datastruktur for en relasjonsdatabase (Primærnøkler er markert med **fete** typer, kandidatnøkler er i *kursiv*, fremmednøkler fremgår av attributt-navnene):

PERSON (**Fnr**, Etternavn, Fornavn, Adresse)
EKTESKAP (**Fnr-k** *Fnr-m*, ***Dato***, Etternavn-k, Etternavn-m)
NAVNESKIFTE (**Fnr**, **Dato**, Etternavn, Fornavn)

Nåværende navn lagres i tabellen PERSON. Følgelig lagres det navnet en person hadde umiddelbart før han/hun skiftet navn, i NAVNESKIFTE. Navnene i EKTESKAP er etternavnene etter vielsen. Dato er en standard SQL `DATE`, dvs. tekst i format `'2004-06-08'`, der rekkefølgen er år, måned og dag.

Løs følgende oppgave både med relasjonsalgebra og SQL:

> Finn navn og adresse til alle kvinner som ved en vielse i 2003 har byttet etternavn med sin brudgom. Ektefeller med samme etternavn skal ikke være med.

Slutt på oppgave 1

Oppgave 2 står på neste side

Oppgave 2 Normalisering

En liten ungdomsklubb tilbyr medlemmene sine (alle har både fasttelefon og mobiltelefon) en rekke kurs. For å holde oversikt over hvem som deltar på hva, har de laget en tabell (i Excel) med følgende 8 kolonner:

Medlemsnr, Navn, Adresse, Fasttelefon, Mobiltelefon, Kurskode, Kursnavn og Instruktør. Følgende funksjonelle avhengigheter gjelder:

Medlemsnr bestemmer Navn, Adresse, Fasttelefon og Mobiltelefon
Mobiltelefon bestemmer Medlemsnr, Navn, Adresse og Fasttelefon
Adresse og Fasttelefon bestemmer hverandre
Kurskode bestemmer Kursnavn og Instruktør

Oppgave 2 a Hvilke kandidatnøkler og hvilken normalform har tabellen?

Oppgave 2 b Normaliser tabellen til BCNF

Oppgave 3 Parsing, spørreplan og optimering

I denne oppgaven skal du vise din forståelse av hvordan en spørring parses, en logisk spørreplan lages, og hvordan spørreplanen kan optimeres. Vi skal bruke følgende relasjoner i denne oppgaven:

KUNDE(**kundeID**, kjønn, fornavn, etternavn, personnummer)
KONTO(**kontoNR**, kontotype, rentesats, opprettelsesdato)
KONTOEIERSKAP(**eierskapsID**, kundeID, eierskap, kontoNR)

Merk at primærnøkler er angitt med **dobbelunderstrekede og fete** typer. Andre kandidatnøkler er kun understreket.

Relasjonen KONTO inneholder bankkonti av flere typer og datoene kontoene ble opprettet. Det kan være flere kunder tilknyttet en konto. Dette er angitt i KONTOEIERSKAP der feltet eierskap enten er 'E' som betyr at vedkommende er eier av kontoen, eller 'D' som betyr at vedkommende disponerer den. Eieren er den som har opprettet kontoen. Attributtet opprettelsesdato i KONTO er en standard SQL DATE, dvs. tekst i format '2004-06-08', der rekkefølgen er år, måned og dag. Kjønn i KUNDE kan

være `'K'` eller `'M'`. Attributtet `kontotype` i `KONTO` er `'B'` for bruks-konto, `'S'` for sparekonto og `'L'` for lånekonto.

Vi skal bruke følgende spørring for å finne fornavn, etternavn og personnummer til kvinnelige kunder som har opprettet en sparekonto i 2003:

Oppgave 3 fortsetter på neste side

```
SELECT   KUNDE.fornavn, KUNDE.etternavn,
         KUNDE.personnummer
FROM     KUNDE, KONTO, KONTOEIERSKAP
WHERE    KUNDE.kundeID = KONTOEIERSKAP.kundeID AND
         KONTO.kontoNR = KONTOEIERSKAP.kontoNR AND
         KONTO.opprettelsesdato LIKE '2003%'   AND
         KONTO.kontotype = 'S'                 AND
         KUNDE.kjønn = 'K'
     AND
         KONTOEIERSKAP.eierskap = 'E'
```

Databasen har «clustered» indekser på primærnøklene. I tillegg er det indekser på attributtet `opprettelsesdato` i `KONTO` og på attributtene `kundeID` og `kontoNR` i `KONTOEIERSKAP`.

Oppgave 3 a Parsing

i) Bruk den enkle grammatikken på side 5 til å lage et parse-tre for spørringen ovenfor.

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

Oppgave 3 b Logisk spørreplan

Konverter parse-treet i oppgave 3 a ovenfor til en logisk spørreplan i relasjonsalgebra (tegn uttrykkstreet). NB! Denne oppgaven skal løses uten optimering. Optimering hører til neste oppgave!

Oppgave 3 c Optimering

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

ii) Optimer den logiske spørreplanen i deloppgave 3 b ovenfor (tegn det nye uttrykkstreet).

Oppgave 4 Indekser

Oppgave 4 a

Tegn og forklar hva tette (dense), sparsomme (sparse) og multi-nivå (multi-level) indekser er.

Oppgave 4 b

Forklar kort fordeler/ulemper med disse tre typene indekser, i hvilke tilfeller de anbefales brukt, og hvorfor.

Slutt på oppgave 4

Oppgave 5 og 6 står på neste side

Oppgave 5 Logging

Oppgave 5 a

Beskriv de to hovedtypene logger: optimistiske (Undo) og pessimistiske (Redo)

Oppgave 5 b

Fortell hva et sjekkpunkt er. Legg hovedvekt på loggbruken.

Oppgave 6 Transaksjonshåndtering

De to serialiserbarhetsbegrepene som er undervist i dette kurset, er basert på henholdsvis konflikt- og view-ekvivalens av eksekveringsplaner.

Oppgave 6 a

Definer konfliktekvivalens og view-ekvivalens.

Oppgave 6 b

Hva er det flest av, konflikt- eller view-serialiserbare eksekveringsplaner?

Hvilken tilleggsbetingelse må vi ha for at konflikt- og view-ekvivalens skal bli det samme?

Bevis at denne tilleggsbetingelsen sikrer at konflikt- og view-ekvivalens blir det samme.

Slutt på oppgavesettet

Naci Akkøk og Ragnar Normann

Vedlegg til oppgave 3 — Grammatikk for parsing av spørsmål

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

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

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

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

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

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