

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i INF3100 — Databasesystemer

Eksamensdag: 11. juni 2012

Tid for eksamen: 14.30–18.30

Oppgavesettet er på 5 sider.

Vedlegg: Ingen

Tillatte hjelpemidler: Kalkulator og ordbok

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

Les oppgavene nøye, og lykke til!

Supermarkedkjeden Rama består av stormarkeder som selger mat, klær, bøker, leketøy, elektroniske artikler og mye annet. Hver butikk består av flere avdelinger; to typiske avdelinger er grønnsakavdelingen og skotøyavdelingen.

Vi skal se på en liten database som holder orden på varene i en av butikkene. Alle varene er plassert i reoler; hver reol har 4–8 hyller. For enkelhets skyld antar vi at det på hver hylle bare er én type varer. Dessuten skal det aldri være varer fra to avdelinger i samme reol. Hver avdeling administrerer sine varer, dvs. avdelingen holder orden på varene i sine reoler og sørger for å kjøpe inn flere når det trengs.

Informasjon om varenes plassering i butikken og innkjøp av varer er samlet i følgende to relasjoner der primærnøkklene er understreket:

Vareplassering(reolnr, hyllenr, vare, antall, avdeling)

Innkjøp(vare, dato, antall, enhetspris, bestiller, avdeling)

I relasjonen `Vareplassering` er hver hylle identifisert ved hvilken reol (`reolnr`) den befinner seg i og hvilket hyllenummer (`hyllenr`) den har (nederste hylle er alltid nummer 1). `vare` og `antall` angir hvilken vare og hvor mange eksemplarer av varen som ligger i hyllen. Det er forøvrig ikke noe i veien for at det er så mye av en vare at den opptar mer enn én hylle. `avdeling` angir hvilken avdeling som har ansvar for reolen.

I relasjonen `Innkjøp` beskriver hvert tuppel ett innkjøp/én bestilling av en vare, hvilket tidspunkt bestillingen ble foretatt (`dato`), hvor mye som er

(Fortsettes på side 2.)

bestilt av varen (**antall**) og **enhetspris** for denne bestillingen . Videre står det hvilken ansatt (**bestiller**) som foretok bestillingen, og hvilken **avdeling** som har ansvaret for bestillingen.

For at databasen skal ha de rette egenskapene, må **Vareplassering** oppfylle følgende funksjonelle avhengigheter (FDer):

reolnr \rightarrow **avdeling**

vare \rightarrow **avdeling**

I tillegg skal **Innkjøp** oppfylle følgende regler:

- Hver vare er underlagt én avdeling, så det er alltid en og samme avdeling som har ansvaret for bestilling av en bestemt vare.
- Hver person er ansatt på én avdeling.

Oppgave 1 FDer og normalformer (30%)

- (10%) Finn ut hvilken normalform hver enkelt av FDene i **Vareplassering** har. Begrunn svaret.
- (5%) Dekomponer **Vareplassering** tapsfritt til BCNF. Er dekomposisjonen FD-bevarende?
- (5%) Angi hvilke funksjonelle avhengigheter som gjelder i **Innkjøp**.

Gitt en relasjon $R(A, B, C, D, E, F, G, H)$ med FDene $AE \rightarrow D$, $AFG \rightarrow C$, $B \rightarrow E$, $C \rightarrow AB$, $CD \rightarrow H$, $EF \rightarrow CDG$.

- (10%) Gitt dekomposisjonen $\{ABFG, ACFG, CDE, EFH\}$ av R . Finn ut om dekomposisjonen er tapsfri.

Oppgave 2 SQL (25%)

Ta utgangspunkt i relasjonene **Vareplassering** og **Innkjøp**. Besvar spørsmålene under ved hjelp av SQL. Du kan bruke views som del av besvarelsene.

- (5%) Finn den eller de varene som tar opp mest hylleplass. Skriv for hver ut navnet på varen og antall hyller varen befinner seg på.

(Fortsettes på side 3.)

- (ii) (5%) Finn gjennomsnittlig innkjøpspris pr. kilo for bananer i 2011. (Når det gjelder bananer og frukt generelt, inneholder attributtet **antall** i **Innkjøp** antall kilo som bestillingen omfatter.)
- (iii) (5%) Finn den eller de reolene der summen av alle varenes verdi i reolen er den høyeste. Vi måler en vares verdi ved å bruke nyeste innkjøpspris (enhetspris) for varen.
- (iv) (5%) Finn ut om det er noen reol der bestilling av alle varene som finnes i reolen, er foretatt av én person. Det kan være null, en eller flere slike reoler. Skriv ut reolnummeret til disse.
- (v) (5%) Finn ut om det er noen personer som har stått for mer enn ti innkjøp/bestillinger og der alle vedkommendes bestillinger gjelder forskjellige varer.

Oppgave 3 Relasjonsalgebra og optimering av spørringer (20%)

- (i) (10%) Bruk relasjonsalgebra til å skrive en spørring som for hver vare sjekker at den avdelingen som står oppført i **Vareplassering**, stemmer overens med den avdelingen som står oppført i **Innkjøp**.

Uttrykket under finner de varene hvor der er færre enn fem eksemplarer i en hylle og hvor det er blitt foretatt en bestilling på minst ti eksemplarer av varen i mai eller senere.

$$\pi_{\text{Vareplassering.vare}}(\sigma_{\text{Vareplassering.antall} < 5 \text{ and } \text{Innkjøp.dato} > '2012-04-30' \text{ and } \text{Innkjøp.antall} > 10}(\sigma_{\text{Innkjøp.vare} = \text{Vareplassering.vare}}(\text{Innkjøp} \times \text{Vareplassering})))$$

- (ii) (10%) Uttrykket utgjør en logisk spørreplan. Optimer den logiske spørreplanen (tegn den nye planen i form av et uttrykkstre).

Oppgave 4 Transaksjonsprotokoller (25%)

I denne oppgaven skal vi se på transaksjoner og låsing i et flerbrukersystem. En minimal relasjonsdatabase for å håndtere bestilling av flybilletter ser slik ut, der primærnøkler er understreket:

Rute(rutenr, fra, til)
 Flyavgang(rutenr, dato, antallSeter, ledigeSeter)
 Bestilling(rutenr, dato, passasjernavn)

(Fortsettes på side 4.)

Vi skal anta at kundebruk av systemet skjer via en app som har tilgang til følgende transaksjonstyper:

Type 1 – Bestilling(rutenr, dato, passasjernavn): Finn den aktuelle flyavgangen i `Flyavgang`. Hvis `ledigeSetter > 0`, så sett inn et nytt tuppel i `Bestilling` og minsk verdien til `ledigeSetter` med 1 i `Flyavgang`.

Type 2 – Avbestilling(rutenr, dato, passasjernavn): Fjern den aktuelle bestillingen fra `Bestilling`. Finn den aktuelle flyavgangen i `Flyavgang` og øk `ledigeSetter` med 1.

Anta at transaksjoner av de to typene ser slik ut fra transaksjonshåndterers ståsted:

Type 1: $r(a); w(b); w(a)$

Type 2: $r(b); w(b); r(a); w(a)$

Her representerer a et tuppel i `Flyavgang`. I type 1-transaksjonen representerer $w(b)$ innsetting av et nytt tuppel b i `Bestilling`. I type 2-transaksjonen representerer $r(b); w(b)$ oppletting av og deretter sletting av tuppelet b i `Bestilling`.

I tillegg til de transaksjonstypene som er tilgjengelige for kundeappen, fins noen administrative transaksjoner som vi ikke skal se nærmere på her, men som gjør at vi velger et låseregime som kan forhindre fantomtupler. Derfor skal vi bruke varsellåser. Vi har to låsbare enheter: relasjoner og tupler. Når vi i type 1 og 2-transaksjonene skal oppdatere, sette inn eller slette et tuppel, setter vi først varsellås på hele relasjonen, deretter setter vi en vanlig lås på det aktuelle tuppelet.

Kompatibilitetsmatrisen for varsellåser er som følger:

		IS	IX	S	X	(ønsket lås)
(holdt lås)	IS	Ja	Ja	Ja	Nei	
	IX	Ja	Ja	Nei	Nei	
	S	Ja	Nei	Ja	Nei	
	X	Nei	Nei	Nei	Nei	

Her er IS og IX varsellåser, mens S og X er vanlige lese- og skrivelåser. En skrivevarsellås på relasjonen `Flyavgang` angis f.eks. ved `IX(Flyavgang)`. Opplåsing av et element c angis ved `U(c)` (hvor c er en relasjon eller et tuppel).

(Fortsettes på side 5.)

- (i) (15%) Anta at vi bruker tofaselåsing. Vis hvordan hver av transaksjonstypene ser ut når transaksjonshåndtereren har lagt inn låser på passende steder.

To personer prøver å gjøre følgende samtidig:

- Person 1 bestiller en billett Oslo-Tromsø
- Person 2 avbestiller en billett Oslo-Tromsø

- (ii) (10%) Kan denne situasjonen gi vranglås (deadlock)? Begrunn svaret.
(Hvis du ikke har besvart spørsmål (i), kan du prøve å besvare spørsmål (ii) under antakelsen av at vi har vanlige eksklusive låser $l(c)$ i stedet.)

Ellen Munthe-Kaas