NTNU
Kunnskap for en bedre verden

Institutt for datateknologi og informatikk

Eksamensoppgave i TDT4145 Datamodellering og databasesystemer

Faglig kontakt under eksamen:

Roger Midtstraum: 995 72 420

Eksamensdato: 7. juni 2017

Eksamenstid (fra-til): 09:00-13:00

Hjelpemiddelkode/Tillatte hjelpemidler:

D – Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Annen informasjon:

Målform/språk: Norsk bokmål

Antall sider (uten forside): 6

Antall sider vedlegg: 0

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig ☐ **2-sidig** X

sort/hvit X **farger** ☐

skal ha flervalgskjema ☐

Kontrollert av:

23/5-2017 SEB

Dato Sign

Merk! Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

Oppgave 1 – Datamodeller (20 %)

Lag en ER-modell (du kan bruke alle virkemidler som er med i pensum) for følgende forenklede utgave av stortingsvalg.

Stortingsvalg er organisert med utgangspunkt i fylkene som har et antall stortingsrepresentanter hver. For hvert fylke stiller et antall partier med en fylkesliste som består av partiets kandidater, rangert i nummerert rekkefølge. Hvert fylke har et entydig fylkesnummer og et fylkesnavn. Hvert fylke er delt inn i kommuner som gjennomfører selve valget. Kommuner har entydig kommunenummer og et kommunenavn.

De politiske partiene som stiller til valg i minst ett fylke, tildeles et nasjonalt unikt partinummer, har en partikode og et partinavn. Et parti trenger ikke å stille til valg i flere fylker, men vil vanligvis stille liste i alle fylker. Alle partier har en registrert partileder som må være en person som er stemmeberettiget ved valget.

Vi holder oversikt over alle personer i landet. Personer er registrert med en unik personkode, for- og etternavn, fødselsdato, gateadresse, postnummer og poststed. Alle personer er tilknyttet en bostedskommune. For hver person skal det være registrert om vedkommende er stemmeberettiget ved valget. Personer som ikke er stemmeberettiget ved valget, er ikke valgbare og kan derfor ikke stå på noen fylkesliste over foreslåtte kandidater. Personer som ikke er stemmeberettigede, kan ikke være partiledere og kan heller ikke ha offisielle oppgaver i forbindelse med valget.

For å gjennomføre selve valget er kommunene delt inn i valgkretser og alle stemmeberettigede personer er tilknyttet en valgkrets der vedkommende må levere sin stemme. Valgkretser tildeles et valgkretsnummer som er unikt innenfor den kommunen valgkretsen tilhører. For hver valgkrets registreres navn, gateadresse, postnummer og poststed. I tillegg registrerer man antall stemmeberettigede personer og antall avgitte stemmer i valgkretsen. Når en person avgir stemme registreres det at denne personer har stemt, men det er hemmelig valg og det registreres ikke hvilket parti som fikk stemmen.

De avgitte stemmene telles opp i hver valgkrets. Valgkretsen registrerer hvor mange stemmer som ble avgitt for hver fylkesliste, og man registrerer også antall blanke stemmer som ble avgitt i valgkretsen. Når alle stemmer er talt opp i en valgkrets registreres det at denne valgkretsen er ferdig. Når alle valgkretser i et fylke er ferdige, registreres det hvor mange stortingsrepresentanter som ble valgt fra hver fylkesliste.

For hver valgkrets oppnevnes det et valgstyre bestående av stemmeberettigede personer som har ansvaret for gjennomføring av valget og opptelling av stemmer. En av disse personene utnevnes som valgstyrets leder og for akkurat denne personen registreres mobiltelefonnummer.

Personer kan bare stille som kandidat på en fylkesliste.

Gjør kort rede for eventuelle forutsetninger som du finner det nødvendig å gjøre.

NB! Oppgavesettet fortsetter på neste side.

Oppgave 2 – Relasjonsdatabaser, relasjonsalgebra og SQL (24 %)

Ta utgangspunkt i følgende relasjonsdatabase:

Harvest

PersonID	FruitType	Weight
1	1	100
3	1	100
2	1	200
1	3	300

Persons

PersonID	Name
1	Kari
2	Ola
3	Per
4	Liv
5	Liv

Fruits

FruitType	FruitName
1	Oranges
2	Apples
3	Bananas
4	Grapes

a) Hvilke *nøkler* (kandidatnøkler) og *fremmednøkler* har vi i de tre tabellene? Gjør kort rede for eventuelle forutsetninger som du finner det nødvendig å gjøre.

b) Tegn *tabellforekomsten* som blir resultatet av å utføre SQL-spørringen under:

```
select PersonID, Name, sum(Weight) AS KilosHarvested
from Persons natural join Harvest
group by PersonID, Name
order by KilosHarvested DESC, Name ASC
```

c) Lag en spørring i *relasjonsalgebra* som finner navn på personer som har plukket Oranges.

d) Lag en spørring i *SQL* som finner navn på frukttypene som personer med navn Kari har plukket.

e) Lag en spørring i *SQL* som finner PersonID og Name for personer som ikke har plukket noe frukt.

NB! Oppgavesettet fortsetter på neste side.

f) Lag en spørring i *SQL* som finner FruitType, FruitName og totalt antall kilo høstet av frukttypen. I resultatet skal vi bare ha med frukttyper der totalhøsten er mer enn 100 kilo.

g) Lag en spørring i *relasjonsalgebra* som finner den samme informasjonen som SQL-spørringen i oppgave b.

Oppgave 3 - Normaliseringsteori (16 %)

a) Forklar/definer begrepet *funksjonell avhengighet* (eng: functional dependency).

b) Gå ut fra at vi har tabellen PizzaOrders for å holde orden på pizzatyper, kunder og pizzabestillinger.

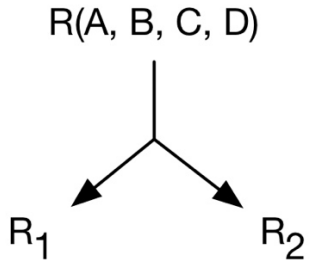
PizzaOrders(OrderNo, PizzaID, PizzaName, PizzaPrice,
CustNo, CustName, NumberOrdered)

der følgende funksjonelle avhengigheter gjelder

F = { OrderNo->PizzaID, CustomerNo, NumberOrdered;
PizzaID->PizzaName, PizzaPrice; CustNo->CustName }

Hvilke *innsettings-, oppdaterings- og slettingsproblemer* (eng: insertion, modification and deletion anomalies) har denne tabellen? Hva vil du foreslå å gjøre for å bli kvitt disse problemene? Gjør kort rede for eventuelle forutsetninger som du finner det nødvendig å gjøre.

c) Tabellen R(A, B, C, D) skal som vist i figuren under dekomponeres i to deltabeller, R_1 og R_2. Funksjonelle avhengigheter som gjelder er F = { C->B }.



Vi ønsker at dekomponeringen skal være attributtbevarende, ha tapsløst-join-egenskapen og bevare de funksjonelle avhengighetene. R_1 og R_2 skal være på Boyce-Codd normalform. Finn et forslag på tabeller, R_1 og R_2, som oppfyller disse kravene. Svaret må begrunnes.

NB! Oppgavesettet fortsetter på neste side.

d) Ta utgangspunkt i tabellforekomsten vist under. Anta at mvd-en Person ->> Sport gjelder for Interests-tabellen. Hvilke rader/tupler må legges til i tabellforekomsten for at dette skal være en gyldig tabellforekomst?

Interests

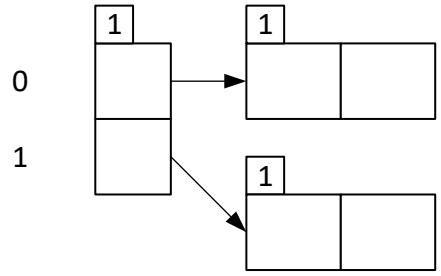
Person	Sport	Food
Kari	Athletics	Icecream
Ola	Tennis	Chocolate
Kari	Soccer	Biscuits

Oppgave 4 – Extendible hashing (5 %)

Vi skal sette inn følgende nøkler i en extendible hashing-struktur: 27, 18, 9, 7, 16, 13. Når vi starter innsettingen har vi kun to blokker, og pekervektoren har to innslag, slik at både global og lokal dybde er 1. Hver blokk har plass til to nøkler. Se figuren under. Du skal bruke hashfunksjonen:

h(K) = K MOD 4

Vis hvordan strukturen ser ut til slutt når du har satt inn alle nøklene. Husk å ta med både global og lokal dybde.



NB! Oppgavesettet fortsetter på neste side.

Oppgave 5 – Lagring og queries (16 %)

Vi har en database som lagrer brukervurdering av filmer. Dette er lagret i en tabell:

Movie (<u>movieId</u>, title, directorId, prodYear, nVotes, avgRating)

Hver film vil ha en post i denne databasen. Vi har lagret informasjon om 200 000 filmer.
Hver post (record) i Movie-tabellen er 100 byte. Hver blokk i databasen er 4096 byte (4 KB).

a) Anta tabellen er lagret i en heapfil. Hvor mange blokker vil heapfilen inneholde?
b) Anta tabellen i stedet er lagret i et clustered B+-tre med movieId som søkenøkkel. Anta movieId er 4 byte og en BlockId er 4 byte. Hvor mange blokker på hvert nivå vil B+-treet inneholde? Forklar evt. antagelser du gjør.
c) Vi har et query for å finne informasjon om en film:
   SELECT * FROM Movie WHERE movieId = 123456;
   Hvor mange blokker vil aksesseres for dette queriet ved å anta lagringen fra a) og hvor mange blokker vil aksesseres ved å anta lagringen fra b)? Skriv opp evt. antagelser du gjør.
d) Anta vi har en annen tabell Actor med 600 000 poster lagret i en heapfil med 16 000 blokker. Denne skal joines med en tabell Agency med 20 000 poster lagret i 500 blokker i en annen heapfil. Vi har kun nested-loop-join tilgjengelig og bufferet har plass til 52 blokker samtidig. Hvor mange blokker leses totalt i joinen?

Oppgave 6 – Transaksjoner: Recovery og låsing (11 %)

Vi har en historie:

H1: r2(A); w1(A); w1(B); w3(B); w2(B); c2; c1; c3

a) (5 %) For hver egenskap under forklar hvorfor historien har eller ikke har denne egenskapen:
   - Unrecoverable
   - Recoverable
   - ACA
   - Strict
b) (6 %) Anta at operasjonene i historien må sette lese- og skrivelåser og at vi har rigorous 2PL (tofaselåsing) og vranglåsoppdagelse (deadlock detection). Skriv om historien slik at den gjør bruk av låser. Innfør operasjonene rl1(X) – read_lock1(X), wl1(X) – write_lock1(X) og ul1(X) -- unlock1(X).

NB! Oppgavesettet fortsetter på neste side.

Oppgave 7 – Transaksjoner: Recovery – ARIES (8 %)

Vi har følgende logg som ble funnet etter en databasekrasj. A, B, C og D er dataelementer og loggpostene har formatet [LSN,Operation,Transaction,DataItem,BeforeImage,AfterImage]. Startverdiene for A, B, C og D står i første rad av tabellen.

	A	B	C	D
Startverdier	30	15	40	20
[101, end_ckpt]				
[102,start_trans,T3]				
[103,write,T3,B,15,11]		11		
[104,commit,T3]				
[105,start_trans,T2]				
[106,write,T2,B,11,19]		19		
[107,start_trans,T1]				
[108,write,T1,D,20,25]				25
[109, commit, T1]				
[110,write,T2,D,25,42]				42

a) (5 %) Anta A, B, C, D er datasider det gjøres recovery på. Hvilke loggposter blir det gjort REDO på under recovery når DPT (Dirty Page Table) har følgende tilstand ved start av REDO:
(B, recLSN=106),
(D, recLSN=110)
og dataelementene har følgende tilstand ved start av recovery:
(A, pageLSN=88,value=30)
(B, pageLSN=106, value=19)
(C, pageLSN=77, value=40)
(D, pageLSN=108, value=25)
For hver loggpost i loggen beskriv om det skjer REDO eller ikke og hvorfor/hvorfor ikke?
NB: (oppdatert 24.mai 2018) I ettertid har det vist seg at DPT ikke kan ha disse verdiene basert på den loggen som vises over. DPT kan høyst ha verdiene (B, recLSN=103) og (D, recLSN=108) basert på den loggen som finnes over. Dette forutsetter at B og D ikke finnes i DPT i sjekkpunktloggposten. Men selve oppgaven i seg selv er greit nok, fordi den tester om studenten har skjønt REDO-test-reglene.

b) (3 %) Hva er verdiene til A, B, C og D etter at hele recovery (også UNDO) er ferdig? Begrunn svaret ditt.