NTNU
Det skapende universitet

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4145 Datamodellering og databasesystemer

Faglig kontakt under eksamen:

Svein Erik Bratsberg: 73550382

Eksamensdato: 12. august 2013

Eksamenstid (fra-til): 15:00 - 19:00

Hjelpemiddelkode/Tillatte hjelpemidler:

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

Annen informasjon:

Målform/språk: Norsk bokmål

Antall sider: 5

Antall sider vedlegg: 0

Kontrollert av:

_____

Dato Sign.

Oppgave 1 – Datamodeller (20 %)

Lag en ER-modell (du kan bruke alle virkemidler som er med i pensum) for følgende situasjon:

I hopprenn er det to omganger. Det deltar et antall hoppere som tildeles et unikt startnummer for dette rennet. I første omgang starter hopperne i startnummer-rekkefølge. I andre omgang deltar bare de 30 beste hopperne etter første omgang. Disse hopperne starter i poengrekkefølge, slik at hopperen som leder etter første omgang, hopper helt til slutt. For hver hopper har vi et unikt utøvernummer, fornavn, etternavn, fødselsår og nasjonalitet.

Et hopp får en poengsum som er summen av lengdepoeng og stilpoeng. Lengdepoengene beregnes ut fra hoppets lengde og et sett parametere for den aktuelle bakken – konstruksjonspunkt og meterverdi. Stilpoengene tildeles av 5 dommere som hver gir fra 0 til 20 poeng. Den laveste og den høyeste poengsummen strykes. Summen av de tre gjenstående dommerpoengene utgjør hoppets stilpoeng. Resultatlisten for et hopprenn ordnes på samlet poengsum etter to hopp, slik at hopperen med høyest samlet poengsum vinner.

Et hopprenn har 5 oppnevnte dommere som har et unikt dommernummer, fornavn, etternavn og nasjonalitet. For å kunne gjøre statistiske analyser på dommerprestasjoner, skal alle dommerpoengsummer lagres. I tillegg til hopplengde og stilpoeng tildeles hvert hopp en status som kan ta verdiene stående, fall, DNS (Did Not Start) og DSQ (DiSQualified).

Et hopprenn arrangeres i en bestemt hoppbakke som har en unik FIS-bakke-ID, navn, by, land, bakkestørrelse (HS – Hill Size), konstruksjonspunkt (K-punkt), meterverdi og bakketype. Den lengste hopplengden som er oppnådd i bakken registreres som bakkerekord. I forbindelse med bakkerekord registreres alle hoppere som har oppnådd denne lengden, og dato det skjedde for hver hopper. Hoppbakker deles inn i bakketyper etter bakkestørrelse:

- Liten bakke (HS under 50 meter)
- Mellombakke (HS 50-84 meter)
- Normalbakke (HS 85-109 meter)
- Stor bakke (HS 110-184 meter)
- Skiflygingsbakke (HS over 184 meter)

Lengdepoengene regnes ut ved at bakkens K-punkt brukes som tabellpunkt. En hopplengde tilsvarende K-punktet gir 60 poeng. Forskjellen mellom hoppets målte lengde og K-punkt-lengden multipliseres med bakkens meterverdi og trekkes fra/legges til de 60 poengene.

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

Oppgave 2 – Relasjonsalgebra og SQL (20 %)

Ta utgangspunkt i følgende relasjonsdatabase (primærnøkler er understreket) for en enkel bedriftsdatabase:

Person(PID, Name, BirthYear, DepartmentID, RoomNo, DeskNo)
- DepartmentID er fremmednøkkel mot Department-tabellen. Attributtet kan ikke ha NULL-verdi.
- RoomNo, DeskNo er fremmednøkkel mot Desk-tabellen. Begge attributtene kan ha NULL-verdi.

Department(DID, Name, LeaderID)
- LeaderID er fremmednøkkel mot Person-tabellen. Attributtet kan ikke ha NULL-verdi.

Room(RoomNo, Type, Area, DepartmentID)
- DepartmentID er fremmednøkkel mot Department-tabellen. Attributtet kan ikke ha NULL-verdi

Desk(No, RoomNo, PhoneNo)
- RoomNo er fremmednøkkel mot Room-tabellen. Attributtet kan ikke ha NULL-verdi.
- PhoneNo er fremmednøkkel mot Phone-tabellen. Attributtet kan ha NULL-verdi.

Phone(PhoneNo, RoomNo)
- RoomNo er fremmednøkkel mot Room-tabellen. Attributtet kan ha NULL-verdi.

Relasjonsalgebra kan formuleres som tekst eller grafer. Hvis du behersker begge notasjonene foretrekker vi at du svarer med grafer, men du blir ikke trukket for å svare med tekst.

a) (4 %) Lag et ER-diagram som i størst mulig grad samsvarer med relasjonsskjemaet. Gjør rede for eventuelle antagelser du finner det nødvendig å gjøre.

b) (4 %) Lag en spørring i *relasjonsalgebra* som finner PID og Name for alle personer som har arbeidsplass på rom som disponeres av avdelingen (department) som heter "Økonomi".

c) (4 %) Skriv en *SQL-setning* som oppdaterer lederen for økonomiavdelingen til "Ole Hansen" som har PID = 100. Du kan forutsette at denne personen finnes i Person-tabellen.

d) (4 %) Skriv en *SQL-setning* som lager et view med alle ansatte i økonomiavdelingen. I dette viewet skal du ta med PID, Name og RoomNo.

e) (4 %) Lag en spørring i *SQL* som finner alle rom der mer enn 4 personer har sin arbeidsplass. Du skal ta med RoomNo og antall personer i resultatet. Resultatet skal være sortert etter antall personer i synkende rekkefølge.

Oppgave 3 – Teori (20 %)

a) (5 %) Ta utgangspunkt i tabellen

Athlete(ID, Name, BirthYear, Sex, Club, Region, StreetAddress, PostalCode, City)

der F = { ID → Name, BirthYear, Sex, Club, StreetAddress, PostalCode; Club → Region; PostalCode → City } er de funksjonelle avhengighetene som gjelder.

Forutsett at tabellen oppfyller kravene til første normalform (1NF). Bestem den høyeste normalformen som oppfylles av tabellen. Du må begrunne svaret.

b) (7,5 %) Gitt tabellen Athlete og funksjonelle avhengigheter som beskrevet i oppgave a. Dekomponer denne tabellen slik at du får en dekomponering med komponent-tabeller på så høy normalform som mulig. Vurder kvaliteten til den dekomponeringen du kommer fram til.

c) (7,5 %) Anta at F = { A → B, CD → A }. Fyll inn verdier i alle de tomme rutene i tabellen under. Dersom en rute kan ha en verdi som ikke er entydig bestemt, kan du bruke b_{ij} (i = radnummer og j = kolonnenummer) som symbol for denne verdien.

A	B	C	D
1	2	4	5
2		4	6
1			
		4	6
3		4	

Oppgave 4 – Lagring, indekser og queryutføring (20 %)

Vi har en tabell for å lagre øvingsgodkjenning:

Exercise (studno, exerno, status)

For hver av SQL-setningene nedenfor, foreslå hvordan du vil lagre og indeksere tabellen (heapfiler, B+-trær, hashing). Anta for hver SQL-setning at denne typen setning er mest brukt. Hvis du gjør bruk av en indeks, fortell hva søkenøkkelen for indeksen er.

a) INSERT INTO Exercise VALUES (123456, 5,'OK');

b) SELECT studno FROM Exercise WHERE status='OK';

c) SELECT studno FROM Exercise WHERE exerno=4 AND status='OK';

d) SELECT count(*) FROM Exercise WHERE studno=123456 AND status='OK';

e) SELECT studno FROM Exercise GROUP BY studno HAVING count(*) >
(SELECT count(*) FROM Exercise WHERE studno=123456 AND status='OK');

Begrunn hvert svar og forklar hvilke antagelser du evt. gjør.

Oppgave 5 – Transaksjoner (10 %)

a) Vis to motiverende eksempler for at vi innfører transaksjoner.

b) SQL har 4 isolasjonsnivåer for transaksjoner. Forklar egenskapene til de forskjellige nivåene.

Oppgave 6 – Join (10 %)

Anta følgende to tabeller:

Fakultet(faknavn,faknr,dekanus)
Student(studno,snavn,faknr)

Anta følgende SQL query:

```
SELECT snavn
FROM Fakultet f, Student s
WHERE f.faknr=s.fakr AND f.faknavn='IME';
```

Anta Fakultet har 500 poster lagret i 10 diskblokker og at Student har 80000 poster lagret i 2000 blokker. Anta videre at du har plass til 8 blokker samtidig i bufferet.
Regn ut antall I/Oer for queriet ved bruk av følgende joinmetoder:

a) Nested loop join.
b) Sort-merge join.