

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4145 Datamodellering og databasesystemer

Faglig kontakt under eksamen:

Roger Midtstraum: 995 72 420

Svein Erik Bratsberg: 995 39 963

Eksamensdato: 17. august 2016

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: 7

Antall sider vedlegg: 0

Kontrollert av:

Svein-Olaf Hvasshovd (sign.)

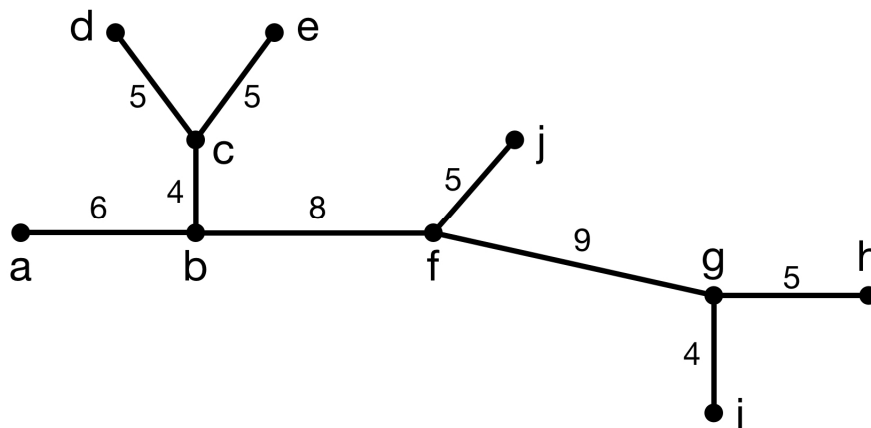
Dato

Sign.

Oppgave 1 – Datamodeller (18 %)

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

Det skal lages en applikasjon for å holde oversikt over veier, veistandard, fartsgrenser og ulykker. Veinettet er organisert som illustrert i figuren under. Vi har endepunkter (vist med bokstaver i figuren) som har en kjent posisjon (x-koordinat, y-koordinat og høyde). Mellom to endepunkter kan det være et veisegment som har et startpunkt, et endepunkt og en lengde (vist med tall i figuren). En vei består av ett eller flere veisegmenter, nummerert slik at vi får en sammenhengende vei fra start- til endepunkt. Et veisegment kan ikke være med i flere veier og det kan finnes veisegment som ikke er del av noen vei.



Vi har et antall forhåndsdefinerte fartsgrenser. Alle veisegment har en bestemt fartsgrense. Dersom man ønsker å differensiere fartsgrensen innenfor et veisegment, må dette veisegmentet splittes opp i flere veisegment, slik at hvert veisegment får en fartsgrense. Tilsvarende har vi et antall forhåndsdefinerte veistandarder, med en tilhørende beskrivelsestekst. Alle veisegment har en bestemt veistandard og splittes om nødvendig opp i flere veisegment for at det skal være slik. En vei har et entydig veinummer, et veinavn og en total veilengde.

Ulykker registreres med tid, dag, måned og år, samt en tekst som beskriver ulykken. Ulykker klassifiseres etter alvorlighetsgrad basert på et forhåndsdefinert klassifikasjonssystem. På den ene siden registreres ulykkens menneskelig konsekvens ("dødsulykke", "alvorlig personskade", "mindre personskade", "ubetydelig eller ingen personskade") og på den andre siden registreres ulykkens materiell konsekvens ("omfattende materielle skader", "mindre materielle skader", "ubetydelige eller ingen materielle skader"). Knyttet til hvert nivå finnes en beskrivende tekst som beskriver kriteriene for å klassifisere ulykken på dette nivået. For en ulykke skal man kunne registrere antall involverte kjøretøy og om ulykken var forårsaket av trafikkregelbrudd eller ikke. For å stedfeste en ulykke skal den plasseres i aktuelt veisegment, med en lengde (offset) som viser hvor langt fra veisegmentets startpunkt ulykken skjedde.

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

NB! Oppgavesettet fortsetter på neste side.

Oppgave 2 – Relasjonsdatabaser, ER-modeller, relasjonsalgebra og SQL (18 %)

Gitt følgende relasjonsdatabase-skjema for en smakstest-database for brus. Primærnøkler er understreket.

SoftDrink(SDID, ProductName, Taste)

TestPerson(TPID, Name, Sex, Age)

Ranking(RID, Description)

TasteTest(TTID, TPID, SDID, RID, TTYear, TTMonth, TTDay)

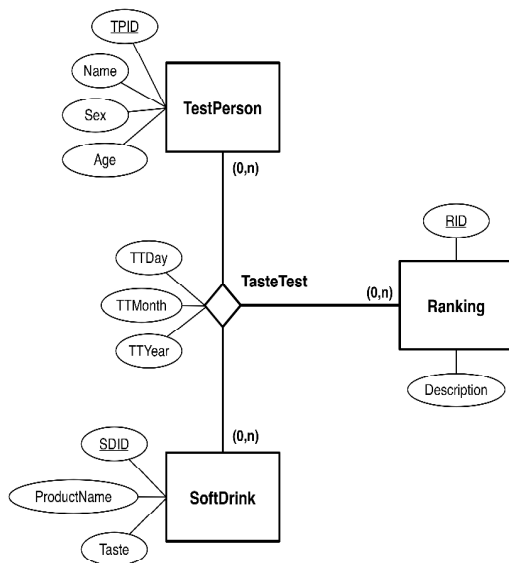
TPID er fremmednøkkel mot TestPerson-tabellen. Kan ikke ha NULL-verdi.

SDID er fremmednøkkel mot SoftDrink-tabellen. Kan ikke ha NULL-verdi.

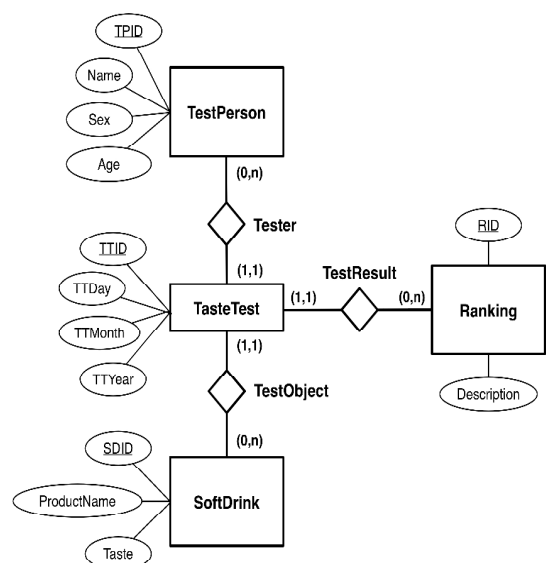
RID er fremmednøkkel mot Ranking-tabellen. Kan ikke ha NULL-verdi.

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) I figuren under har vi vist to alternative ER-modeller for denne databasen. Hvilket alternativ samsvarer best med relasjonsskjemaet? Du må begrunne svaret ditt.



Alternative 1



Alternative 2

NB! Oppgavesettet fortsetter på neste side.

- b) Lag en spørring i *relasjonsalgebra* som finner navn, kjønn og alder for alle som har testet brus (eng: soft drink) med pineapple-smak. Resultatet skal ikke inneholde duplikater (like rader).
- c) Man ønsker at alle testpersoner skal gjøre smaksvurdering av alle bruser. Lag en spørring i *relasjonsalgebra* som finner TPID og SDID for kombinasjoner av testperson og brus som mangler i databasen.
- d) Skriv en spørring i *SQL* som finner gjennomsnittsalder for testpersoner som har gitt brus med pineapple-smak rangering med beskrivelsen (eng: description) "awesome".
- e) Skriv en spørring i *SQL* som for hver rangering finner antall smakstester som har gitt denne karakteren. Resultatet skal inneholde RID, Description og antall smakstester, og skal være sortert på RID (i stigende rekkefølge).
- f) Skriv en spørring i *SQL* som finner alle bruser med orange-smak som har fått flere enn 5 "awesome" vurderinger fra kvinnelige testpersoner. Resultatet skal inneholde SDID, ProductName og antall "awesome" vurderinger.

Oppgave 3 – Datamodellering og relasjonsdatabaser (7 %)

- a) Hva vil det si at en spesialisering (eng: specialization) av en entitetsklasse er *disjunkt*?
- b) Hva vil det si at en spesialisering av en entitetsklasse er *total*?
- c) Hva vil det si at tabeller (relasjoner) i relasjonsdatabaser har *referanseintegritet* (eng: referential integrity)?

Oppgave 4 - Normaliseringsteori (5 %)

Under er vist en tabellforekomst for tabellen R. Finn alle funksjonelle avhengigheter (eng: functional dependencies) som *ikke* kan gjelde for tabellen dersom denne tabellforekomsten er gyldig.

R

a	b	c	d
1	1	1	1
2	3	1	2
1	1	1	3

NB! Oppgavesettet fortsetter på neste side.

Oppgave 5 - Normaliseringsteori (12 %)

- a) (2 %) Gitt tabellen $R(A,B,C,D)$ og $F=\{A \rightarrow C, B \rightarrow D\}$. Anta at tabellen oppfyller første normalform. Forklar hvorfor dette er den høyeste normalformen som oppfylles av tabellen.
- b) (5 %) Gitt tabellen $R(A,B,C,D)$ og $F=\{A \rightarrow C, B \rightarrow D\}$. Anta at tabellen oppfyller første normalform. Finn en dekomponering der alle komponenttabeller er på Boyce-Codd normalform (BCNF) og som ellers oppfyller alle ønskede egenskaper. Forklar hvorfor det er slik.
- c) (5 %) Gitt tabellen $R(A,B,C,D)$ og $F=\{A \rightarrow C\}$. En dekomponering i $R_1(A,B)$ og $R_2(B,C,D)$ har *ikke* tapsløst-join egenskapen. Lag et konkret eksempel (med tabellforekomster) som viser at det er slik.

Oppgave 6 – B+-trær (10 %)

Sett inn følgende nøkler i et B+-tre: 23, 11, 12, 16, 30, 17, 6, 4, 7, 13.

Anta det er plass til tre nøkler per blokk, både på løvnivå og lengre opp i treet (men 4 pekere til nivået under). Vi tilstanden til B+-treet hver gang du skal til å splitte en blokk, og vis treet til slutt.

NB! Oppgavesettet fortsetter på neste side.

Oppgave 7 – Queries og join (10 %)

- a) Vi har en tabell
Staff (staffNo, fName, lName, depNo, position, salary)

Det er to typer SQL-setninger som er vanlige i denne databasen:

- 1)

```
SELECT fName, lName, salary
FROM Staff
WHERE salary = (SELECT MAX(salary) FROM Staff);
```
- 2)

```
UPDATE Staff SET Salary = Salary + 0.01*Salary WHERE staffNo=123456;
```

Hvordan burde denne tabellen lagres og indekseres for at de to typene av queries skal kunne utføres effektivt.

Gi en begrunnelse for ditt valg.

- b) I tillegg til tabellen i a) har vi tabellen
Department (depNo, dName, managerNo, location)

Vi skal utføre en nested loop-join mellom de to tabellene. Staff er her lagret i en heapfil med 1000 blokker, mens Department er lagret i en heapfil med 30 blokker. Vi har et buffer tilgjengelig med 7 blokker. Hvor mye I/O får vi ved en nested loop-join? Gi svaret i antall blokker du må lese inn fra disk. Du trenger ikke regne med skriveoperasjoner her.

Oppgave 8 – Historie og tofaselåsing (2PL) (10 %)

Gitt følgende historie:

H1: r1(X); r2(Y); r1(Y); r3(X); r3(Y); w1(X); c1; w3(Y); c3; r2(X); w2(Z); c2;

- a) Tegn opp presedensgrafen og avgjør om historien er konfliktserialiserbar.
- b) Innfør tofaselåsing (2PL, rigorous) med lese- og skrive låser i H1. Skriv om historien H1 slik at den gjør bruk av låser og vis hvordan den da utføres. Når en transaksjon må vente på en lås, stopper denne transaksjonen, men andre transaksjoner kan fortsette. Den stoppede transaksjonen kan fortsette når de blokkerende låser slippes. Låsoppgradering er mulig.

NB! Oppgavesettet fortsetter på neste side.

Oppgave 9 – Recovery (10 %)

- a) Ved bruk av ARIES-recovery har vi tre forskjellige tester som vi kan gjøre slik at vi vet at en loggpost ikke trenger REDO. Hvilke tester er det?
- b) Hva skjer med logg- og databuffer i ARIES når en transaksjon committer?