

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4145 Datamodellering og databasesystemer

Faglig kontakt under eksamen:

Svein Erik Bratsberg: 99539963

Roger Midtstraum: 99572420

Eksamensdato: 6. august 2014

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: 5

Antall sider vedlegg: 0

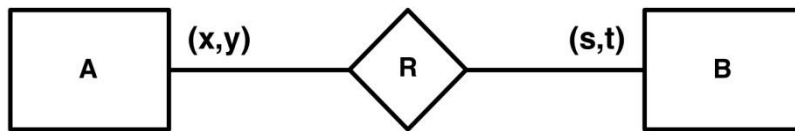
Kontrollert av:

Dato

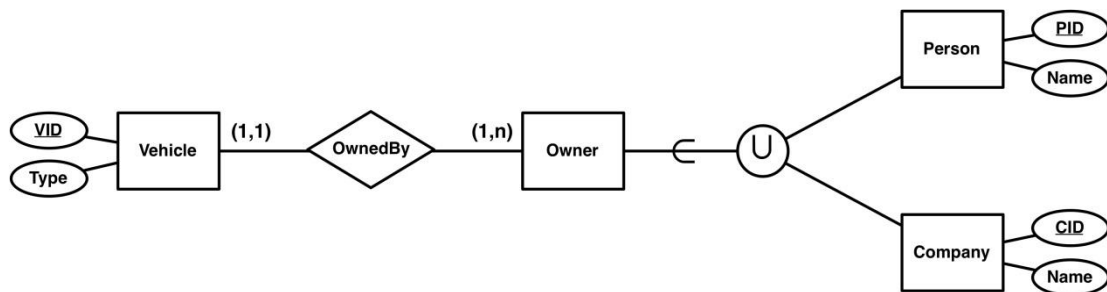
Sign.

Oppgave 1 – Datamodeller (20 %)

- a) (3 %) I ER-modeller kan vi uttrykke kardinalitets-restriksjoner som vist i figuren under. Forklar hvilke restriksjoner (eng: constraints) som kan uttrykkes ved hjelp av x, y, s og t.



- b) (5 %) I ER-diagrammet under har vi vist en ER-modell med bruk av kategori (eng: category). Vis hvordan dette ER-diagrammet kan oversettes til et relasjonsskjema som i størst mulig grad samsvarer med ER-modellen.



- c) (12 %) Lag en ER-modell (du kan bruke alle virkemidler som er med i pensum) for følgende situasjon:

I Nice er det mulig å leie bysykler. Rundt om i byen finnes en del sykkelstasjoner der det er låseterminaler med plass til et antall bysykler. For å kunne leie sykler registrerer man seg med mobiltelefonnummer, navn og et betalingskort. Når man ønsker å leie en sykkel bruker man den registrerte mobiltelefonen og ringer et telefonnummer som er oppgitt for den låseterminalen der sykkelen er låst. Dersom alt er i orden, blir sykkelen frigjort og så betaler man for tiden frem til man igjen låser sykkelen fast i en låseterminal. Alle sykler har en unik RFID-kode som gjenkjennes av låseterminalen når sykkelen leveres inn. Sykkel-leien avsluttes automatisk når låseprosedyren er fullført. De som driver sykkelutleien ønsker å holde oversikt over hvor syklene er låst fast, hvilke sykler som er i bruk, registrerte brukere, hvem som har en sykkel til låns og hvilke utlån som er fullført. Databasen må inneholde nok informasjon til å dokumentere kundenes bruk av syklene. Det finnes ulike typer sykler og man holder rede på når en sykkel settes i drift i systemet.

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 student-emne-eksamen-database:

Student(StudentNo, Name, Email)

Subject(SubjectNo, Name)

Exam(ExamNo, SubjectNo, Year, Month, Date)

- SubjectNo er fremmednøkkel mot Subject-tabellen. SubjectNo kan ikke ha NULL-verdi.

ExamRegistration(StudentNo, ExamNo, Grade)

- StudentNo er fremmednøkkel mot Student-tabellen. StudentNo kan ikke ha NULL-verdi.
- ExamNo er fremmednøkkel mot Exam-tabellen. ExamNo kan ikke ha NULL-verdi.
- Data legges inn ved oppmelding til eksamen. Grade settes til NULL inntil sensuren er klar.

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.

- (4 %) Lag et ER-diagram (du kan bruke alle virkemidler som er med i pensum) som i størst mulig grad samsvarer med relasjonsskjemaet. Gjør rede for eventuelle antagelser du finner det nødvendig å gjøre.
- (3 %) Lag en spørring i *relasjonsalgebra* som finner StudentNo, Name og Email for alle studenter som var oppmeldt til en av de eksamenene som ble arrangert 26. mai 2014.
- (3 %) Lag en spørring i *relasjonsalgebra* som finner alle studenter som ikke er oppmeldt til noen eksamen i 2014. I resultatet skal du ta med StudentNo, Name og Email.
- (3%) Lag en spørring i *SQL* som finner alle emner som er registrert med minst en eksamen i 2014. Resultatet skal bestå av SubjectNo og Name, det skal ikke være duplikater i resultatet, og resultattabellen skal være sortert etter SubjectNo i stigende rekkefølge.
- (3 %) Lag en spørring i *SQL* som finner antall ganger karakteren A har vært gitt siden emnet ble opprettet, i hvert emne som har hatt en eller flere eksamener i løpet av 2014. Resultattabellen skal ha kolonnene SubjectNo, Name og "Number of A". Resultatet skal sorteres etter "Number of A" i synkende rekkefølge.
- (4 %) Anta at alle fremmednøklerne er definert med restriksjonen "ON DELETE RESTRICT" eller "ON DELETE NO ACTION" (disse er likeverdige). Vi ønsker å slette studenten som har StudentNo 100. Vis hvordan du vil gjøre dette i SQL.

Oppgave 3 – Teori (20 %)

- a) (4 %) Gitt $R = \{A, B, C, D, E\}$ og $F = \{AB \rightarrow C, C \rightarrow E, AE \rightarrow B\}$. Tillukningen $AB^+ = ABCE$. Forklar hva dette betyr. Finn alle kandidatnøkler (eng: candidate key) for tabellen R. Svaret må begrunnes.
- b) (6 %) Tabellen $R = \{A, B, C, D\}$ dekomponeres i $R_1 = \{A, C\}$ og $R_2 = \{B, C, D\}$. Finn en mengde funksjonelle avhengigheter (eng: functional dependencies), F , som gjør at denne dekomponeringen har tapsløs-join-egenskapen (eng: lossless join property). Med utgangspunkt i din F , skal du finne alle kandidatnøkler og høyeste normalform for tabellene R , R_1 og R_2 . Alle svarene må begrunnes.
- c) (4 %) Gitt $R = \{A, B, C, D, E\}$ og $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E\}$. Gå ut fra at R oppfyller 1. normalform. Bestem den høyeste normalformen som oppfylles av R . Svaret må begrunnes.
- d) (6 %) Gitt $R = \{A, B, C, D, E\}$ og $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E\}$. Gå ut fra at R oppfyller 1. normalform. R skal om nødvendig dekomponeres slik at alle komponenter er på Boyce-Codd normalform (BCNF). Dekomponeringen skal ha tapsløs-join-egenskapen, attributtbevaring og bevaring av funksjonelle avhengigheter. Du må forklare hvordan din løsning oppfyller alle kravene.

Oppgave 4 – Extendible hashing (10 %)

Vi skal sette inn følgende nøkler i en hashfil med extendible hashing i den gitte rekkefølgen: 1, 2, 3, 4, 5, 6, 7, 8, 9 og 10. Det er plass til maksimalt to nøkler i hver blokk. Vis tilstanden til hashfila til slutt. Anta det er 4 blokker i hashfila når du starter med innsettingen og at hashfunksjonen er $h(K) = K \text{ MOD } 16$. Husk å ta med "global depth" og "local depth".

Oppgave 5 – Lagring og queryutføring (15 %)

Anta følgende tabell:

```
CREATE TABLE Employee (empno INT, lastname CHAR(30), firstname CHAR(30), email CHAR(30), startyear INT);
```

Vi har følgende typer lagring og indekser

1. Heapfil
2. Clustered B+-tre
3. Unclustered B+-tre og heapfil
4. Clustered hashfil
5. Unclustered hashfil og heapfil

For hvert av SQL-queryene under foreslå hvilken lagring/indekstype som er best. Hvis du gjør bruk av en indeks, fortell hvilket attributt som er indeksert. Begrunn svarene.

- i) INSERT INTO Employee VALUES (12123,'Hansen','Hans','hans@email.org',2013);
- ii) SELECT lastname, firstname, email, startyear FROM Employee WHERE startyear=2012;
- iii) SELECT * FROM Employee;
- iv) SELECT DISTINCT lastname FROM Employee ORDER BY lastname;
- v) UPDATE Employee SET startyear = 2012 WHERE empno=12123;

Oppgave 6 – Transaksjoner - SERIALIZABLE (5 %)

Forklar hvilke samtidighetsproblemer vi unngår ved å bruke SQLs isolasjonsnivå SERIALIZABLE.

Oppgave 7 – Transaksjoner – historie og låser (10 %)

- a) Se på sekvensen av operasjoner fra to transaksjoner, T1 og T2, beskrevet i tabellen under. Hva er det som er galt med denne sekvensen av operasjoner?

T1	T2
Read(X); X := X – N;	
	Read(X); X := X+M;
Write(X); Read(Y);	
	Write(X);
Y := Y – N; Write(Y);	

- b) Ved å innføre konservativ 2PL (tofaselåsing), vis hvordan sekvensen av operasjoner blir.