

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4145 Datamodellering og databasesystemer

Faglig kontakt under eksamen:

Svein Erik Bratsberg: 995 39 963

Roger Midtstraum: 995 72 420

Eksamensdato: 4. august 2015

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:

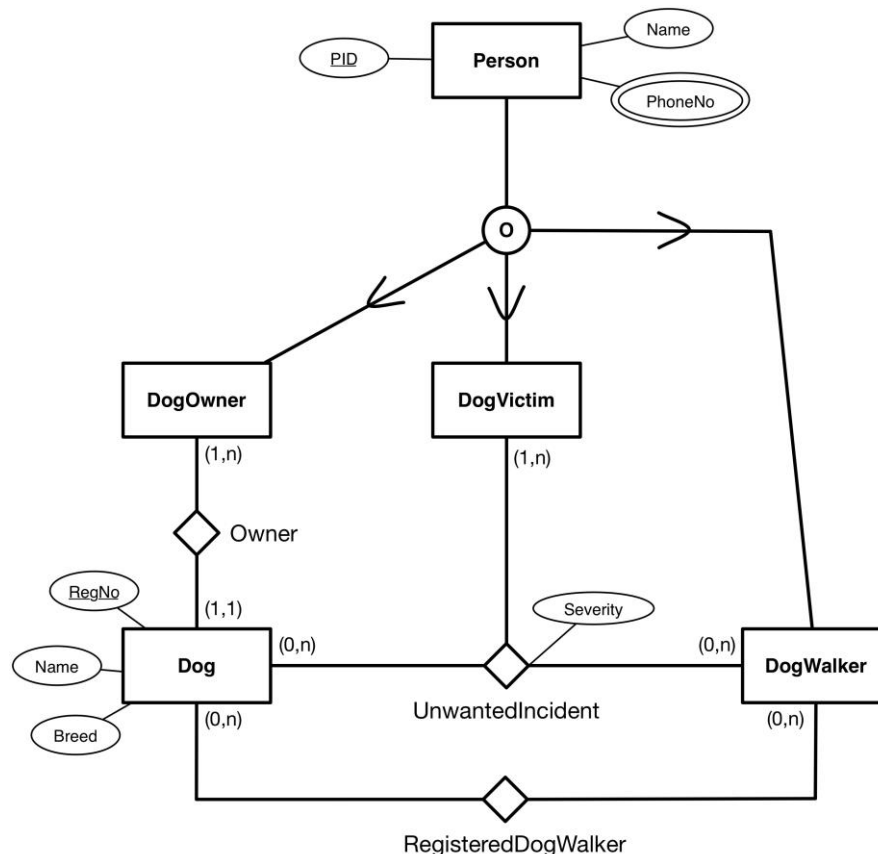
Svein-Olaf Hvasshovd (sign.)

Dato

Sign.

Oppgave 1 – Datamodeller (20 %)

- a) (6 %) Under er det vist et ER-diagram. I tillegg er det kjent at flere personer kan ha samme telefonnummer (PhoneNo). Vis hvordan du vil oversette dette til en "ekvivalent" relasjonsdatabase (vis hvilke tabeller du ønsker å ha, med attributter, primær- og fremmednøkler). Forklar kort hvilke alternative løsninger som finnes og begrunn ditt valg.



- b) (14 %) Lag en ER-modell (du kan bruke alle virkemidler som er med i pensum) for følgende situasjon:

Det arrangeres et sykkelløp, Fosen Rundt, som består av et antall etapper. Det er bare en klasse, og den som har den laveste totaltiden til slutt er sammenlagtvinner. For hver etappe har vi dato, starttid, startsted, målsted og lengde i kilometer. Det lages en resultatliste for hver etappe, rangert etter (korteste) kjøretid. Deltakere som ikke starter på en etappe, får statusen DNS (Did Not Start) for denne etappen. De som starter en etappe, men bryter underveis, får statusen DNF (Did Not Finish) for etappen. En syklist kan bli disket på en etappe og får i så fall statusen DSQ (DiSQualified) for etappen. Syklister med DNS, DNF eller DSQ på en eller flere etapper er ute av sammenlagt-konkurransen, men kan fortsette i Fosen Rundt og konkurrere om plasseringer på de gjenstående etappene.

Syklisterne er registrert som deltakere i Fosen Rundt, med unikt lisensnummer, navn, fødselsår og nasjonalitet. Hver syklist har et (unikt) startnummer og kjører for et lag

som har en unik lagkode på tre bokstaver og et fullstendig lagnavn. For hvert lag skal vi vite navn og mobiltelefonnummer for lagets "race director".

Det lages en resultatliste etter hver etappe med kjøretid (evt. status) for alle syklistene på denne etappen. Når det gjelder sammenlagtkonkurransen har vi en resultatliste som holdes fortløpende oppdatert (etter hver etappe), med total sykletid så langt i rundturen. Når den siste etappen er ferdig vil dette være den endelige resultatlisten for Fosen Rundt. Deltakere som ikke har pådratt seg DNS, DNF eller DSQ på noen etappe, har statusen *Aktiv* og tas med i sammenlagtlisten. Deltakere som pådrar seg DNS, DNF eller DSQ, får statusen *Brutt* og tas ikke med i sammenlagtlisten.

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

Oppgave 2 – Relasjonsalgebra og SQL (21 %)

Ta utgangspunkt i følgende relasjonsdatabase (primærnøkler er understreket) for en enkel billedelings-database:

Photos(PhotoID, PhotoDescription, PhotographerID)

PhotographerID er fremmednøkkel mot Users-tabellen. PhotographerID kan ikke ha NULL-verdi.

Users(UserID, UserName, UserDescription)

Likes(PhotoID, UserID)

PhotoID er fremmednøkkel mot Photos-tabellen. UserID er fremmednøkkel mot Users-tabellen.

Comments(PhotoID, UserID, SeqNo, Content)

PhotoID er fremmednøkkel mot Photos-tabellen. UserID er fremmednøkkel mot Users-tabellen.

Hashtags(TagID, Tag)

Metadata(PhotoID, TagID, UserID)

PhotoID er fremmednøkkel mot Photos-tabellen. TagID er fremmednøkkel mot Hashtags-tabellen. UserID er fremmednøkkel mot Users-tabellen.

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.

- (5 %) 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.
- (4 %) Lag en spørring i *relasjonsalgebra* som finner PhotoID, PhotoDescription og UserName for fotografen, for alle bilder som er tagget med emneknaggen (eng: hashtag) '#Trondheim'.
- (4 %) Lag en spørring i *SQL* som finner TagID og Tag for alle emneknagger som inneholder 'Trondheim'. Resultatet skal ordnes etter Tag i alfabetisk rekkefølge.

- d) (4 %) Brukeren (eng: user) med UserID 'xyz00' har røket uklar med brukeren 'abc11' og ønsker å fjerne alle *likes* som hun har gitt til bilder tatt av 'abc11'. Lag en *SQL-setning* som ordner dette.
- e) (4 %) Lag en spørring i *SQL* som finner UserID for alle brukere som har kommentert bilder som er merket med emneknaggen '#Beiarn'.

Oppgave 3 – Teori (19 %)

- a) (5 %) Definer *funksjonell avhengighet* (eng: functional dependency). Lag en tabell og gi ett eksempel på en funksjonell avhengighet som gjelder i denne tabellen, og ett eksempel på en funksjonell avhengighet som ikke gjelder i denne tabellen. Du må begrunne hvorfor du mener det er slik.
- b) (5 %) Hva vil det si at en dekomponering (eng: decomposition) har *tapsløst-join-egenskapen* (eng: lossless join property)? Vis ett eksempel på en dekomponering som har tapsløst-join-egenskapen, og ett eksempel på en dekomponering som ikke har den. Du må begrunne hvorfor du mener det er slik.
- c) (3 %) Gitt $R = \{A, B, C, D, E\}$ og $F = \{B \rightarrow C, E \rightarrow D\}$. 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 $F = \{A \rightarrow D, B \rightarrow D, C \rightarrow D, E \rightarrow D\}$. Gå ut fra at R oppfyller 1. normalform. Finn en god dekomponering av R slik at alle komponenttabellene er på Boyce-Codd normalform (BCNF). Svaret må begrunnes.

Oppgave 4 – Lagring og indeksering (10 %)

List opp noen alternative lagrings- og indekseringsmetoder brukt i databasesystemer for å lagre og indeksere en tabell. Forklar når de forskjellige metodene er fordelaktige å bruke.

Oppgave 5 – B+-trær (10 %)

Anta følgende tabell som er opprettet som et clustered B+-tre med søkenøkkel 'lastname'.

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

Anta B+-treet har en blokkstørrelse på 8192 bytes.

- a) Anta at tabellen er tom og så kjøres følgende følgende SQL-setning:
`INSERT INTO Employee VALUES`
`(123,'Hansen','Hans','hans@email.org',2012,100000),`
`(124,'Hanssen','Jens','jens@email.org',2013,90000),`
`(125,'Olsen','Ole','ole@mail.com',2008,80000),`
`(126,'Berntsen','Bernt','bernt@mail.com',2015,200000);`

Forklar hvordan B+-treet ser ut etter at denne setningen er utført. Dvs. tegn opp B+-treet med blokker og poster.

- b) I en blokksplitt senere blir det splittet mellom 'Hansen' og 'Hanssen'. Vis hvordan B+-treet ser ut nå, med fokus på posten som det splittes på. Vis blokker på både løvnivå (level 0) og nivået over (level 1) i B+-treet.

Oppgave 6 – Transaksjoner - serialiserbarhet (5 %)

Anta følgende historier:

H1: r1(A); w1(A); r2(B); r2(A); w2(A);

H2: r1(A); w1(A); w2(A);

H3: r1(A); r2(C); r1(C); r3(A); r3(B); w1(A); w3(B); r2(B); w2(C); w2(B);

Tegn opp presedensgrafene for disse historiene og bestem hvilke som er konfliktserialiserbare.

Oppgave 7 – Transaksjoner - recovery (15 %)

- a) I ARIES brukes det et begrep som kalles PageLSN. Hva brukes dette til?
- b) Hvilke fordeler har man ved å bruke NO-FORCE framfor FORCE?
- c) Når det i ARIES skal skrives «dirty» blokker fra databufferet til disk, hvilken sjekk og evt. hvilke aksjoner i forhold til loggen må den som skriver ut datablokkene gjøre?