Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

EKSAMENSOPPGAVE I TDT4145 – DATAMODELLERING OG DATABASESYSTEMER

Faglig kontakt under eksamen: Svein Erik Bratsberg og Roger Midtstraum

Tlf.: 99539963 (Bratsberg) og 99572420 (Midtstraum)

Eksamensdato:	**4. juni 2012**
Eksamenstid:	**15.00-19.00**
Tillatte hjelpemiddel:	**D: Ingen trykte eller håndskrevne hjelpemiddel tillatt. Bestemt, enkel kalkulator tillatt.**

Språkform: Bokmål

Sensurdato: 25. juni 2012

Oppgave 1 – Datamodellering – 20 %

Drømmereiser arrangerer gruppereiser med busser til forskjellige reisemål i Europa. Selskapet tilbyr et antall bussreiser som har tittel, tekstlig beskrivelse, startdato, sluttdato og fullpris per person. Registrerte kunder kan registrere interesse for en bussreise og i tilfelle tar selskapet vare på datoen dette ble gjort. Kunder er registrert med navn (obligatorisk), mobiltelefonnummer, e-postadresse og kategori (”barn”, ”voksen”, ”pensjonist”). For hver bussreise brukes et antall busser som er registrert med type og kapasitet (antall seter). Det selges billetter som reserverer et bestemt buss-sete for en bestemt kunde for den aktuelle bussreisen. Dersom en buss blir innblandet i en ulykke, må man kunne finne ut hvem som var med i bussen. Alle passasjerer må derfor være registrerte kunder. Hver billett har en pris som er avhengig av bussreisens fullpris og hvilken kategori kunde det er. Barn betaler for eksempel 50 % av fullpris.

Du skal lage en ER-modell (du kan bruke alle elementer som er undervist) for en database som holder oversikt over Drømmereisers virksomhet, som beskrevet over. Databasen skal blant annet kunne brukes for å finne svar på spørsmål som:

- Hvor mange billetter er solgt til bussreisen *Eksotiske Haparanda* som har reisenummer 12-2012?
- Hvilke passasjerer skal reise med bussen med registreringsnummer VH34576 på turen *Jämtland rundt* som starter 4. juni 2012?
- Hvor mange kunder har registrert interesse for turen med tittel *Fosen – et Norge i miniatyr*?
- Hva var gjennomsnittlig andel pensjonister per buss på reiser som ble gjennomført i 2011?
- Hvilke ti bussreiser var mest populære målt i antall solgte billetter i 2011?

Når det gjelder egenskaper som naturlig modelleres som attributter, skal du ta med de attributtene som fremgår av beskrivelsen over og de attributtene som du anser som viktigst å ha med. Det er ikke nødvendig å ta med alle attributter som ville fremkommet gjennom en virkelig modelleringsprosess.

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

Oppgave 2 – Relasjonsalgebra og SQL – 20 %

Ta utgangspunkt i følgende relasjonsdatabase (primærnøkler er understreket) for skøyteløp:

SkøyteLøp(LID, Klasse, Distanse, Dato, Tid)

SkøyteLøper(SID, Navn, FødselsÅr, Nasjonalitet, Klasse)

Passeringer(LID, SID, Lengde, PasseringsTid, RundeTid) – LID er fremmednøkkel mot SkøyteLøp, SID er fremmednøkkel mot SkøyteLøper.

Resultat(LID, SID, SluttTid, Plassering) – LID er fremmednøkkel mot SkøyteLøp, SID er fremmednøkkel mot SkøyteLøper.

Attributtet *Distanse* forteller hvor langt et skøyteløp er, for eksempel 500 m, 1500 m eller 5000 m. I passeringer forteller *Lengde* hvor mange meter som er tilbakelagt. Skøytebaner er rundbaner på 400 m; på en 1500 m vil man derfor ha passeringspunkt etter 300 m, 700 m, 1100 m og 1500 m. Rundetid er tid siden start for den første passeringen og tid siden forrige passering for de etterfølgende passeringene. En skøyteløper må fullføre et skøyteløp for å bli registrert i Resultat-tabellen for dette skøyteløpet.

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.

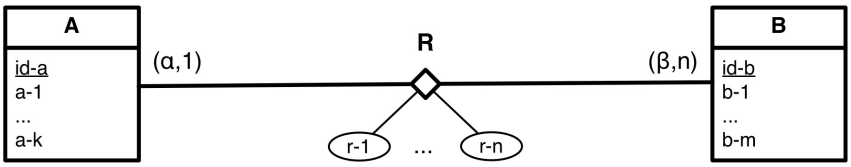
a) Lag en spørring i *relasjonsalgebra* som finner alle skøyteløpere (SID, Navn og Nasjonalitet) som *ikke* har fullført noen løp (dvs. at skøyteløperen ikke finnes i Resultat-tabellen).
b) Lag en spørring i *relasjonsalgebra* som finner alle norske skøyteløpere (SID, Navn og FødselsÅr) som har fullført (finnes i Resultat-tabellen) ett eller flere løp på 5000 m.
c) Lag en spørring i *SQL* som finner alle skøyteløpere (SID, Navn og Nasjonalitet) som har gått løp på 5000 m der alle rundetider var bedre enn 30 sekunder. Du kan anta at RundeTid lagres som sekunder – for eksempel 29,72 som betyr 29 sekunder og 72 hundredeler.
d) Lag en *SQL-spørring* som finner alle skøyteløp som hadde flere enn 20 deltakere som fullførte løpet (finnes i Resultat-tabellen). Den resulterende tabellen med LID og antall deltakere skal være sortert i synkende rekkefølge etter antall deltakere som fullførte.

Oppgave 3 – Teori – 15 %

a) La X være en mengde attributter. Forklar hva som menes med *tillukningen* (Engelsk: closure) til mengden attributter, X^+. Ta utgangspunkt i R = {a, b, c, d, e} og F = {a->c, bc->d, cd->e}, hva er ab^+?
b) Gitt en tabell R = {a, b, c, d, e, f} der det *ikke* finnes noen funksjonelle avhengigheter (Engelsk: functional dependencies). Under hvilke forutsetninger kan og bør tabellen dekomponeres i R_1 = {a, b} og R_2 = {a, c, d, e, f}?
c) Hva betyr det at relasjonsdatabaser skal ha *referanseintegritet* (Engelsk: referential integrity) og hvorfor er dette viktig?

Oppgave 4 – Mapping – 5 %

a) I figuren under er det vist en ER-modell, der id-a og id-b er nøkkelattributter, a-1 – a-k, b-1 – b-m og r-1 – r-n er attributter. α og β kan ha verdiene 0 (null) eller 1 (en). Hvilke alternativer har vi når denne ER-modellen skal oversettes til en relasjonsdatabase? Diskuter fordeler og ulemper ved de ulike alternativene og gi råd om når de forskjellige alternativene bør velges.



Oppgave 5 – Lagring, indekser og queryutføring – 25 %

a) **(5%) Statisk hashing**. Anta poster med nøklene gitt i følgende rekkefølge: 121, 204, 731, 547, 800, 221, 932, 145, 112, 721, 412. Anta videre at ei blokk har plass til tre poster og at det er 4 blokker i den statiske hashfila, og at det er brukt separat, lenket overløp. Bruk hashfunksjonen *h(K) = K MOD 4*. Vis hashfila etter at alle postene er satt inn. Hva er gjennomsnittlig antall blokker aksessert ved direkte søk etter nøklene i fila?

b) **(5%) Lineær hashing**: Sett inn de samme nøklene i samme rekkefølge som i oppgave a), men bruk lineær hashing. Start med 4 blokker. Anta det er plass til 3 nøkler per blokk. Bruk hashfunksjonen *h(K) = K MOD 8*. Gjør utvidelse av fila hver gang en overflytsblokk må lages. Vis sluttilstanden for hashstrukturen.

c) **(10%) B+-trær**: Sett inn de samme nøklene som i oppgave a) i den samme rekkefølgen. Anta det er plass til tre nøkler i hver blokk, også på indeksnivå. På indeksnivå er det plass til 4 blokkpekere i hver blokk. Vis B+-treet hver gang du skal til å splitte ei blokk og helt til slutt.

d) **(5%) Join**: Vi har følgende to tabeller

```
Employee(ssn,first_name,last_name,bdate,sex,salary,super_ssn,dno)
Department(dname,dnumber,mgr_ssn,mgr_start_date)
```

Anta at `Department` har 1000 poster lagret i 12 diskblokker og `Employee` har 60000 poster i 2000 diskblokker. Gitt det følgende queryet:

```
SELECT e.last_name, e.first_name
FROM Department d, Employee e
WHERE e.dno=d.dnumber AND
     d.dname='Accounting';
```

Anta du har plass for 10 diskblokker tilgjengelig i buffer. Hvor mange blokker må leses fra disken ved å bruke en nested-loop-join for `e.dno=d.number`?

Oppgave 6 – Transaksjoner – 15 %

a) **(5%) Rollback:** La A, B, C og D være dataelementer med angitte startverdier og gitt loggen under med loggposter på formatet:

[LSN,Operation,Transaction,DataItem,BeforeImage,AfterImage]

Noen av feltene er valgfrie.

	A	B	C	D
	30	15	40	20
[101,start_trans,T3]				
[102,write,T3,B,15,20]		20		
[103,start_trans,T2]				
[104,write,T2,B,20,18]		18		
[105,start_trans,T1]				
[106,write,T1,D,20,25]				25
[107,write,T2,D,25,26]				26

Hvis T2 rulles tilbake som en konsekvens av konflikten med T3, hvilke verdier vil dataelementene A, B, C, D ha etter at T2 har blitt rullet tilbake?

b) **(10%) ARIES-recovery:** Anta loggen over og at A, B, C, D er datasider det skal gjøres recovery på. Hvilke loggposter blir det gjort REDO på under recovery av denne loggen når Dirty Page Table (DPT) har følgende tilstand etter analysen:

(B,recLSN=104)
(D,recLSN=107)

Anta blokkene har følgende tilstand på disken:

(A,pageLSN=100,value=30),
(B,pageLSN=104,value=18),
(C,pageLSN=50,value=40),
(D,pageLSN=106,value=25)

For hver loggpost i loggen forklar hvorfor det blir gjort eller ikke gjort REDO.