

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: 4. juni 2016

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: 7

Antall sider vedlegg: 0

Kontrollert av:

Svein-Olaf Hvasshovd (sign.)

Dato

Sign.

Oppgave 1 – Datamodeller (20 %)

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

Butikkjeden FotoRental har en rekke butikker der registrerte kunder kan leie kamera og objektiver. Hver butikk har et antall kameraer og et antall objektiver, og kan bare leie ut utstyr som hører til i egen butikk. Butikkene har unike navn og registreres i datasystemene også med en adresse, et antall telefonnummer og en e-postadresse. Kundene skal registreres med e-postadresse som brukes som identifikator, navn, adresse og et mobilnummer. Kunder kan gjøre ulike leieavtaler med forskjellige butikker og kan gjøre mange leieavtaler med samme butikk.

En leieavtale gjelder minst ett kamera eller minst ett objektiv, men kan gjelde flere kamera og flere objektiver. En avtale har en registrert startdato og en registrert innleveringsdato. Alle kamera og objektiver har en fast døgnpris som er den samme i alle butikker og for alle kamera/objektiver av samme type, men kunder kan forhandle fram spesielle priser for hele leieavtaler. En avtalt, samlet utleiepris blir derfor registrert for hver leieavtale. Dersom leid utstyr leveres tilbake for sent eller i skadet tilstand, må kunden betale et ekstra straffegebyr som registreres sammen med en forklarende tekst. Når utstyret i en leieavtale er levert tilbake og kunden har betalt leie og evt. straffegebyr, settes leieavtalens status fra aktiv til avsluttet.

Alle kamera og objektiver har et unikt serienummer, en dato for når det ble tatt i bruk, status (fungerende eller defekt) og en tilstandskode (1-5, der 1 indikerer så godt som nytt og høyere tall indikerer økende grad av slitasje). Når et kamera eller et objektiv kommer tilbake etter utlån blir det sjekket, og status og tilstandskode blir endret hvis det er nødvendig.

Et kamera er av en bestemt kameratype, for eksempel 'Canon EOS 7d Mark II' og kameratypen er beskrevet med bildestørrelse i megapixels, vekt i gram, minnekort-type og døgnleie-pris i kroner. Hver kameratype produseres av en bestemt produsent. For produsenter, som kan produsere enten kamera, objektiver eller begge deler, skal vi kunne lagre produsentnavn og produsentland, samt navn på norsk importør og dennes telefonnummer og e-postadresse. Objektiver er av en bestemt objektivtype, for eksempel 'Sigma 24mm F1.4 A DG HSM Canon', som er objektiv produsert av Sigma som passer til Canon-kamera. For objektivtyper må vi altså kunne registrere produsent, så vel som alle kameratyper som objektiver av denne typen kan brukes sammen med. For objektivtyper er brennvidde (for eksempel 24 mm), største blenderåpning (for eksempel F1.4), vekt i gram, lengde i mm og filterdiameter i mm relevant informasjon, i tillegg til døgnleiepris (tilsvarende som for kameratyper).

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

NB! Oppgavesettet fortsetter på neste side.

Oppgave 2 – Relasjonsdatabaser og SQL (10 %)

R

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

S

B	D
1	4
3	4
5	4

- (5 %) Gitt to tabeller, R og S, som vist over. R og S kan blant annet sammenstilles ved å bruke (i) naturlig join, (ii) full-outer-join over B-kolonnene eller (iii) kartesisk produkt (cross join). Vis hva som blir resultatet i de tre tilfellene. Du skal vise resultat-tabeller med kolonner og rader.
- (3 %) Forklar hvordan vi i forbindelse med fremmednøkler (eng: foreign keys) kan spesifisere ulike typer restriksjoner (eng: constraints).
- (2 %) Hva vil det si at tabeller (relasjoner) i relasjonsdatabaser har *entitetsintegritet* (eng: entity integrity)?

NB! Oppgavesettet fortsetter på neste side.

Oppgave 3 – Relasjonsalgebra og SQL (15 %)

Gitt følgende relasjonsdatabase-skjema for en enkel eksamensdatabase. Primærnøkler er understreket.

Course(CID, CCode, Title, Credits, Description, CLevel)

Student(SNo, Name, Email)

Exam(ENo, CourseID, EDay, EMonth, EYear, Duration, EType)

CourseID er fremmednøkkel mot Course-tabellen. Kan ikke ha NULL-verdi

ExamRegistration(ExamNo, StudentNo, RDay, RMonth, RYear)

ExamNo er fremmednøkkel mot Exam-tabellen. Kan ikke ha NULL-verdi.

StudentNo er fremmednøkkel mot Student-tabellen. Kan ikke ha NULL-verdi.

ExamResult(ExamNo, StudentNo, Grade)

ExamNo er fremmednøkkel mot Exam-tabellen. Kan ikke ha NULL-verdi.

StudentNo er fremmednøkkel mot Student-tabellen. Kan ikke ha NULL-verdi.

GradeExplanation(ExamNo, StudentNo, ExpText)

ExamNo er fremmednøkkel mot Exam-tabellen. Kan ikke ha NULL-verdi.

StudentNo er fremmednøkkel mot Student-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.

- ExamRegistration-tabellen lagrer oppmeldinger til eksamen. Lag en spørring i *relasjonsalgebra* som for hver eksamen med 1 eller flere oppmeldinger gjort i 2016, finner antall studenter som har meldt seg opp til denne eksamenen i 2016. I resultatet skal du ha en kolonne for ExamNo og en kolonne for antall oppmeldte studenter. Resultatet skal være sortert etter antall oppmeldte i synkende rekkefølge.
- Lag en spørring i *relasjonsalgebra* som finner Sno, Name og Email for studenter som er oppmeldt til eksamen i emnet med CCode lik TDT4145 og eksamensdato 04-06-2016.
- Skriv en spørring i *SQL* som finner SNo, Name og Email for studenter som ikke var oppmeldt til noen eksamen som ble arrangert i 2015.
- Skriv en spørring i *SQL* som for hver karakter (som ble gitt til noen) finner antall studenter som fikk karakteren ved eksamen i TDT4145 (CCode) som ble arrangert 01-06-2015. Resultatet skal sorteres etter karakter i alfabetisk rekkefølge.
- Skriv en *SQL-setning* som registrerer en ny eksamen med disse dataene: ENo=999, CourseID=100, EDay=4, EMonth=6, EYear=2016, Duration=4, EType='Written'.

NB! Oppgavesettet fortsetter på neste side.

Oppgave 4 - Normaliseringsteori (6 %)

I øvelsen lengdehopp får hver utøver 6 forsøk (eng: trials). For hvert hopp måles lengde og vind, og det avgjøres om hoppet er gyldig (eng: approved) eller dødt (eng: foul). Under er vist en tabell med resultater fra seks lengdehopp.

LongJump

AthleteID	Name	Club	Class	Trial	Length	Wind	Status
1	Ola	BIL	B14	1	4,25	+2,0	Approved
2	Geir	SIL	B14	1	4,38	+0,1	Approved
3	Per	RIL	B14	1	2,98	-0,3	Approved
1	Ola	BIL	B14	2	NULL	+1,4	Foul
2	Geir	SIL	B14	2	4,40	+0,7	Approved
3	Per	RIL	B14	2	2,99	0,0	Approved

- Bestem hvilke funksjonelle avhengigheter (eng: functional dependencies) som vil gjelde for denne tabellen. Forklar de forutsetningene du trenger å gjøre for at det skal være slik.
- Bestem kandidatnøkler (eng: candidate keys) for denne tabellen. Du må begrunne svaret.
- LongJump-tabellen oppfyller bare første normalform (1NF). Hvilke problemer vil et databasedesign som dette kunne medføre?

Oppgave 5 - Normaliseringsteori (9 %)

- Gitt tabellen $R(A,B,C,D)$ og $F=\{A \rightarrow B, C \rightarrow D\}$. Finn en funksjonell avhengighet som hvis den blir lagt til i F , gjør at A blir en supernøkkel (eng: superkey) for R . Vi foretrekker svar der så få attributter som mulig inngår i den funksjonelle avhengigheten. Du må begrunne svaret ditt.
- Gitt tabellen $R(A,B,C,D)$ og $F=\{AB \rightarrow C, BC \rightarrow D\}$. Denne tabellen er på andre normalform (2NF), men ikke på tredje normalform (3NF). Forklar hvorfor det er slik.
- Gitt tabellen $R(A,B,C,D)$ og $F=\{AB \rightarrow C, BC \rightarrow D\}$. R kan dekomponeres i tabellene $R_1(A,B,C)$ og $R_2(B,C,D)$. Vurder kvaliteten til denne dekomponeringen.

NB! Oppgavesettet fortsetter på neste side.

Oppgave 6 – Lagring og queries (15 %)

Vi har en tabell for å lagre informasjon om ansatte:

Employee (ssn, fname, lname, salary, super_ssn, dno)

Attributtet ssn er primærnøkkel for tabellen. Vi har valgt å lagre og indeksere tabellen på følgende måte. Tabellen er lagret i en clustered, statisk hash-struktur hvor alle de 100 000 ansatte får plass i 1250 datablokker. Søkenøkkel er ssn. Overflyt er organisert slik at posten lagres i første etterfølgende blokk som har ledig plass. Gjennomsnittlig aksesseres da 1.2 blokker per søk på ssn.

I tillegg er tabellen indeksert gjennom et unclustered B+-tre med søkenøkkel 'lname' (lastname) som gir ssn til den/de ansatte. B+-treet har 3 nivåer og 600 blokker på løvnivå.

Gi et estimat på hvor mange blokker som aksesseres (dvs. leses og skrives) ved de følgende SQL-setningene:

- i) INSERT INTO Employee VALUES ('12345678901', 'Hans', 'Hansen', 500000, '98765432101', 101);
- ii) SELECT * FROM Employee WHERE ssn='12345678901';
- iii) SELECT * FROM Employee WHERE lname='Hansen';
- iv) SELECT lname FROM Employee WHERE salary > 600000; Anta 10% av de ansatte tilfredstiller betingelsen.
- v) SELECT DISTINCT lname FROM Employee ORDER BY lname;

Gi en forklaring for hvert svar.

Oppgave 7 – Historier og recoveryegenskaper (9 %)

Gitt følgende historier:

- 1. r1(X); r2(Y); w3(X); r2(X); r1(Y); c1; c2; c3;
- 2. r1(X); r1(Y); w1(X); r2(Y); w2(Y); w1(X); r3(Y); c1; c2; c3
- 3. r2(X); w3(X); c3; w1(Y); c1; r2(Y); w2(Z); c2;

Avgjør recoveryegenskapene (ikke gjenopprettbar, gjenopprettbar, ACA eller strikt) til historiene

NB! Oppgavesettet fortsetter på neste side.

Oppgave 8 – Historier, serialiserbarhet og tofaselåsing (2PL) (6 %)

Gitt følgende historie:

H4: r1(X); r2(Z); r1(Z); r3(X); r3(Y); w1(X); c1; w3(Y); c3; r2(X); w2(Z); w2(Y); c2;

- Tegn opp presedensgrafen og avgjør om historien er konfliktserialiserbar.
- Innfør tofaselåsing (2PL, rigorous) med lese- og skrive-låser i H4. Skriv om historien H4 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.

Oppgave 9 – Recovery (10 %)

Vi bruker ARIES-recovery og har følgende logg under restart etter krasj. Hver rad i tabellen under representerer en loggpost.

LSN	Prev_lsn	Transaction	Op_type	Page_id	Other_info
101			Begin_ckpt		...
102			End_ckpt		...
103	0	T1	Update	E	...
104	0	T2	Update	C	
105	104	T2	Commit		
106	103	T1	Update	C	...

- Anta at både transaksjonstabellen og DPT er tomme i sjekkpunktet funnet i loggposten med LSN 102. Hvordan ser transaksjonstabellen og DPT ut etter at analysefasen er ferdig? Transaksjonstabellen har følgende felter: Transaction, Last_lsn og Status. DPT har: Page_id og RecoveryLSN.
- Under redofasen trenger datablokk C redo av både loggpost 104 og 106, mens datablokk E trenger ikke redo. Hvilke verdier for PageLSNene kan datablokk C og E ha i det redofasen starter?