

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: 23. mai 2013

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: 6

Antall sider vedlegg: 0

Kontrollert av:

Dato

Sign.

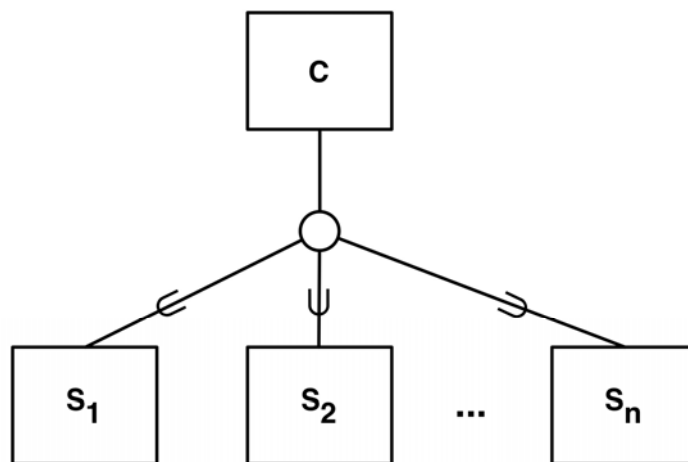
Oppgave 1 – Datamodeller (20 %)

- a) (5 %) Når er det hensiktsmessig å bruke en *svak entitetsklasse* (eng: weak entity class)? Lag et eksempel på et ER-diagram som viser bruk av en svak entitetsklasse. Bruk denne modellen til å forklare begrepene *identifiserende relasjonsklasse* (eng: identifying relationship type) og *delvis nøkkel* (eng: partial key).
- b) (10 %) Lag en ER-modell (du kan bruke alle virkemidler som er med i pensum) for følgende situasjon:

Vi har *emner* som er beskrevet med et unikt emnenummer og et emnenavn. Emner kan ha et antall *eksamener* som er beskrevet med eksamensdato og en oppgavetekst. Et emne har aldri flere eksamener på samme dag. En eksamen er for ett bestemt emne, og den samme oppgaveteksten blir aldri brukt for flere eksamener. *Studenter* er beskrevet med et entydig studentnummer, navn og et eller flere telefonnummer. Studenter kan ta en eller flere eksamener og oppnår i så fall en karakter for prestasjonen på hver eksamen. Dersom en student tar flere eksamener i samme emne, skal alle resultater lagres. En student kan gi en *vurdering* av et emne. En vurdering består av en rangering fra 1 til 6, der 1 er elendig og 6 er fremragende, og en kort, utfyllende tekst. En og samme student kan gjøre flere vurderinger av et og samme emne.

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

- c) (5 %) Forklar hvilke alternativer vi har for å oversette en ER-modell med spesialisering, som vist i Figur 1 under, til et relasjonsdatabaseskjema. Diskuter fordeler og ulemper ved de ulike alternativene.



Figur 1. Spesialisering av entitetsklasse. Legg merke til at figuren bare viser strukturen i spesialiseringen. Attributter og relasjonsklasser er utelatt i figuren, men du skal ta hensyn til slike modellelementer i svaret ditt.

Oppgave 2 – Relasjonsalgebra og SQL (20 %)

Ta utgangspunkt i følgende relasjonsdatabase (primærnøkler er understreket) for en enkel fotodatabase:

Photographer(PGID, Name, BirthYear, Nationality)

Photo(PhotoID, Title, Date, PhotographerID)

– PhotographerID er fremmednøkkel mot Photographer-tabellen.

Keyword(KID, Keyword, Description)

KeywordPhoto(KID, PhotoID, Date)

– KID er fremmednøkkel mot Keyword-tabellen. PhotoID er fremmednøkkel mot Photo-tabellen.

Person(PersonID, Name, Description, Nationality)

PersonInPhoto(PersonID, PhotoID, Description)

– PersonID er fremmednøkkel mot Person-tabellen. PhotoID er fremmednøkkel mot Photo-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 som i størst mulig grad samsvarer med relasjonsskjemaet. Gjør rede for eventuelle antagelser du finner det nødvendig å gjøre.
- (2 %) Lag en spørring i *relasjonsalgebra* som finner Name og BirthYear for norske fotografer.
- (3 %) Lag en spørring i *relasjonsalgebra* som finner fotografier som avbilder en eller flere norske personer, og som er tatt av en svensk fotograf. I resultatet skal du ta med PhotoID, Title og fotografens navn (Name).
- (3 %) Lag en spørring i *SQL* som finner PhotoID og Title for alle fotografier som ikke er beskrevet med noe nøkkelord (Keyword).
- (3 %) Lag en spørring i *SQL* som finner navn (Name) på alle personer som er avbildet i fotografier som har "terrengsykling" som nøkkelord (Keyword). Hvert navn skal forekomme bare en gang i resultatet.
- (4 %) Lag en spørring i *SQL* som finner PersonID, Name og antall fotografier for alle personer som er avbildet i mer enn 100 fotografier. Svaret skal ordnes i synkende rekkefølge etter antall fotografier.

Oppgave 3 – Teori (20 %)

- a) (2 %) Gitt $R = \{A, B, C, D, E, F\}$. Finn en mengde funksjonelle avhengigheter (eng: functional dependencies), G , slik at $AB^+ = ABCD$.
- b) (3 %) Hvilke krav må være oppfylt for at en mengde attributter skal være en *primærnøkkel* (eng: primary key) for en tabell?
- c) (5 %) Vi ønsker å dekomponere $R = \{A, B, C, D\}$ i $R_1 = \{A, B, C\}$ og $R_2 = \{B, C, D\}$. Finn en mengde funksjonelle avhengigheter, F , som garanterer at dekomponeringen har tapsløs-join-egenskapen (eng: lossless join property). Svaret må begrunnes.
- d) (5 %) Gitt $R = \{A, B, C, D, E, F\}$ og $G = \{C \rightarrow D, D \rightarrow E, E \rightarrow F\}$. En mulig dekomponering av R er $R_1 = \{A, B, C\}$, $R_2 = \{C, D\}$ og $R_3 = \{D, E, F\}$. Vurder kvaliteten på denne dekomponeringen.
- e) (5 %) Gitt $R = \{A, B, C, D\}$ og $F = \{B \rightarrow C, C \rightarrow D\}$. En av de funksjonelle avhengighetene bryter kravene til andre normalform (2NF). Hvilken avhengighet er dette og hva er problemet? Den andre avhengigheten bryter bare kravene til tredje normalform (3NF). Hvilken avhengighet er dette og hva er problemet?

Oppgave 4 – Lagring, indekser, queryutføring (15 %)

Vi har en tabell for å lagre sensur. Nøkkelen er understreket:

Grading(studno, courseno, grade)

Det er gitt følgende alternative måter å lagre og indeksere denne tabellen:

- i) Heapfil
- ii) Clustered B+-tre med søkenøkkel (studno, courseno)
- iii) Heapfil og unclustered B+-tre med søkenøkkel (studno, courseno)

Det er 200 000 poster i denne tabellen. Det er plass til 200 poster per blokk i heapfila. B+-treet bruker samme blokkstørrelse som heapfila. Du kan anta at en post i indeksen har samme størrelse som en post i tabellen. Dette gjelder både for clustered B+-tre og unclustered B+-tre. Alle B+-trær i denne oppgaven har høyde 3.

For hver av SQL-setningene nedenfor, regn ut hvor mange I/O-er vi får for hvert av de tre lagringsalternativene:

- a) `SELECT grade FROM Grading WHERE courseno='TDT4145' AND studno=123456;`
- b) `INSERT INTO Grading VALUES (123456, 'TDT4145', 'A');`
- c) `SELECT studno, grade FROM Grading WHERE studno < 10000;`

I c) kan du anta at 5 % av postene kvalifiserer. Svaret i denne oppgaven blir en tabell på 3 ganger 3 elementer med tall.

Oppgave 5 – Transaksjoner - gjenopprettbarhet (10 %)

Anta følgende historier:

H1: $r1(A); r2(A); w1(A); r1(B); w2(B); c2; c1;$

H2: $w1(A); w2(A); c1; c2;$

H3: $r1(A); r2(C); r1(C); r3(A); r3(B); w1(A); w3(B); r2(B); w2(C); w2(B); c1; c2; c3;$

Beskriv recoveryegenskapene til historiene (unrecoverable, recoverable, ACA, strict).

Oppgave 6 – Transaksjoner - låser (5 %)

Anta følgende historie:

H4: $r1(A); r2(B); w1(B); w1(C); r2(A); c1; c2;$

Innfør lese- og skrivelåser (2PL - rigorous) i H4. Skriv om historien H4 slik at den gjør bruk av låser.

Oppgave 7 – Transaksjoner - recovery (10 %)

- a) En transaksjon har oppdatert en tabell og gjort ei blokk i bufferet "skitten" (dirty). Hvilke metoder finnes for å la transaksjonen oppfylle D-en (durability) i ACID-egenskapene?
- b) Anta ARIES-recovery og 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]

	A	B	C	D
Start values	30	15	40	10
[101,start_trans,T3]				
[102,write,T3,B,15,12]		12		
[103,write,T3,C,40,45]			45	
[104,start_trans,T2]				
[105,write,T2,B,12,18]		18		
[106,start_trans,T1]				
[107,write,T1,D,10,15]				15
[108,write,T2,D,15,26]				26

Anta A, B, C, D er datasider det skal gjøres recovery på. DPT har følgende tilstand:

(B,recLSN=105),

(D,recLSN=108)

etter analysen. Blokkene har følgende tilstand på disken:

(A,pageLSN=100,value=30),

(B,pageLSN=105,value=18),

(C,pageLSN=103,value=45),

(D,pageLSN=107,value=15)

For hver write-loggpost i denne loggen forklar hvorfor (eller hvorfor ikke) det blir gjort redo under recovery.