

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i: IN212 Databaseteori
Eksamensdag: Lørdag 3. juni 2000
Tid for eksamen: 09.00 - 15.00
Oppgavesettet er på: 4 sider
Vedlegg: Ett ark med relasjonsdatabaseskjema
Tillatte hjelpemidler: Alle trykte og håndskrevne

Les hele oppgaven nøye og kontroller at oppgavesettet er komplett før du begynner å besvare det. Alle oppgavene bør besvares, så disponer tiden godt.

Oppgave 1 (65%) - Datamodellering og spørsmålsprosessering

Forelesningsdatabase

Universitetet har tatt initiativet til en database over alle forelesninger med tilhørende opplysninger om alle studenter som følger kursene. Databasen skal lagre informasjon om personer (ansatte eller studenter), kursinformasjon, forelesningsmaterieell, øvelser og eksamen.

Entitetstyper med egenskaper:

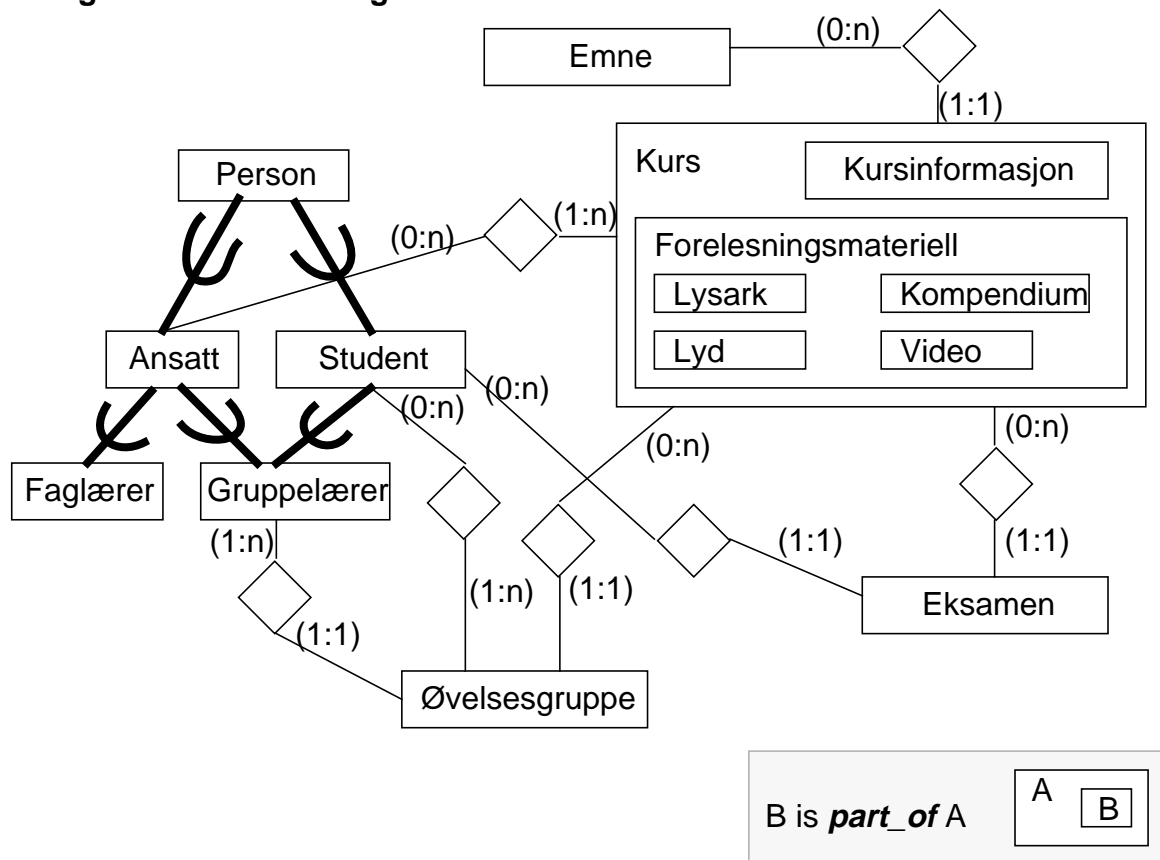
- **Person:** fødselsnummer, navn (består av for- og etternavn), adresse (består av gate, husnr., postnr., poststed, tlf.)
- **Student:** som person, i tillegg: hovedfag, epostadresse
- **Faglærer:** som person, i tillegg: tlf. jobb, epostadresse, arbeidssted (består av institutt navn og adresse), tittel/jobb-betegnelse
- **Gruppelærer:** som student og faglærer, i tillegg: kurskode, gruppenummer
- **Emne:** kode, navn, emnebeskrivelse (BLOB hentet fra studiehåndboken), forutsetninger (liste av emner), vekttall, liste av lærebøker
- **Kurs:** emne, semester (vår/høst og årstall), lærere, øvelsesgrupper (tid(er), sted(er), gruppelærer, studenter), forelesningstid(er), forelesningssted(er), eksamen (tid, sted), kursinformasjon (websider, forelesningskatalog), forelesningsmaterieell (lysark (=foiler) (kurs, kapittelnr.), kompendier (kurs, versjonsnr., navn, forfatter(e)), bilder, lydopptak, video)
- **Øvelsesgruppe:** gruppelærer, sted, tid, kurs, oppmeldte studenter (liste av studenter)
- **Eksamen:** kurs, student, resultat

Entitetene er allerede gruppert sammen til entitetstyper. Legg merke til at:

- noen entitetstyper inneholder også komplekse, sammensatte egenskaper i tillegg til de elementære
- noen entitetstyper inneholder andre entitetstyper
- noen entitetstyper arver alle egenskapene til andre entitetstyper

NB: Fordi vi skal realisere EER-diagrammet i ODL, har vi innført entitetstypen Ansatt. Merk at både Person og Ansatt skal realiseres som signaturklasser (interface).

EER-diagram for forelesningsdatabasen:



A. Beskriv en objektorientert skjemadesign (angi bare signaturen for metodene) i ODL-notasjon ut fra den gitte beskrivelsen av miniverden og EER-diagrammet.

B. Beskriv de integritetsregler som følger av, og modeller (beskriv i pseudo-JAVA- eller pseudo-SIMULA-notasjon) metoder for å håndheve, følgende forretningsregler:

- en student som tar et kurs, kan ikke samtidig være gruppelærer for dette kurset i dette semesteret
- en student skal bare ha registrert ett eksamensresultat for et emne (kurs) i eksamensprotokollen (beste avlagte eksamen)

C. Vedlagt er en mulig realisering av et utsnitt av EER-modellen som en relasjonsdatabase (se vedlegg). Løs følgende oppgaver med SQL (mot modellen i vedlegget) og OQL (mot ditt objektorienterte skjema):

- finn alle par av studenter med felles adresse
- finn alle data for å lage timeplanen for student X dette semester
- regn ut karaktergjennomsnittet for student X for alle avlagte eksamener

D. Sammenlign den relasjonelle løsningen med ditt objektorienterte skjema. Diskuter og drøft de brukte konsepter og nevnt fordeler og ulemper med de to realiseringene. Sammenlign de relasjonelle SQL-spørringene med OQL-spørringene, både med hensyn på uttrykkskraft og effektivitet.

E. Løs følgende to spørsmål i relasjonsalgebra og tuppelkalkyle:

- finn navnet på alle kompendier faglæreren med navn "X" har vært med på å skrive
- finn navn på faglærere som er (med)forfatter for alt kursmateriell for ett eller annet emne

F. Hvilken integritetsregler (avhengigheter) må vi introdusere i den nye tabellen dersom vi vil slå sammen de tre tabellene Emne, Kurs og Kursmateriell i den relasjonelle modellen til én tabell?

Oppgave 2 (20%) - Relasjonsteori

La R være en relasjon med attributter A_1, \dots, A_n . La $D = \{R_1, \dots, R_m\}$ være en dekomposisjon av R og la F være en mengde funksjonelle avhengigheter over R .

Da kan vi bruke følgende algoritme for å avgjøre om D er tapsfri med hensyn på F (algoritme 15.2 i læreboken):

1. Lag en matrise S med en rad for hver komponent R_i i D og en kolonne for hvert attributt A_j i R og sett $S(i,j) := b_{i,j}$ for alle matriseplasser
2. For hver rad i og kolonne j : Hvis R_i inneholder A_j , så sett $S(i,j) := a_j$
3. Gjenta følgende løkke helt til et fullstendig gjennomløp av løkken ikke endrer noe som helst i S :
 - {for hver funksjonell avhengighet $X \rightarrow Y$ i F
 - {for alle rader i S som har de samme symbolene i de kolonnene som svarer til attributtene i X
 - {gjør alle symboler i hver kolonne som svarer til et attributt i Y like i disse radene etter følgende regel:
 - Hvis en av radene har et a -symbol i kolonnen, sett dette a -symbolet inn i de andre radene også.
 - Hvis ingen av radene har et a -symbol i kolonnen, velg ett av de involverte b -symbolene og sett dette symbolet inn i denne kolonnen i de andre involverte radene } } }
4. Hvis (minst) en rad består av bare a -symboler, er D tapsfri; hvis ikke, er D ikke tapsfri

Som en øvelsesoppgave har du tidligere bevist at dersom algoritmen ikke gir noen rad med bare a -symboler, så er ikke D tapsfri. Du skal nå bevise det motsatte, at dersom algoritmen gir en rad med bare a -er, så sikrer det at D er tapsfri. Dette kan du gjøre ved å følge denne oppskriften:

Trinn A

Beskriv starttilstanden i algoritmen (etter punkt 2) ved hjelp av domenekalkyle. a_1, \dots, a_n skal være de frie og $b_{1,1}, \dots, b_{m,n}$ de bundne domenevariablene i domenepredikatet, og domenepredikatet skal uttrykke at hver linje i tabellen er en lovlig forekomst i R .

Trinn B

Begrunn kort at resultatet i trinn A også kan brukes som en domenekalkyledefinisjon av $\pi_{R_1}(R) * \dots * \pi_{R_m}(R)$ (der $*$ betegner naturlig join).

Trinn C

Det å bruke en funksjonell avhengighet i punkt 3 i algoritmen til å forandre en verdi i en kolonne kan beskrives som en endring av domenepredikatet. Beskriv denne endringen og begrunn hvorfor slike endringer ikke har betydning for resultatet i trinn B.

Trinn D

Bruk resultatet fra trinn C til å vise at dersom algoritmen produserer linjen $a_1 \dots a_n$, så er D en tapsfri dekomposisjon.

Oppgave 3 (15%) - Ytelsesanalyse

Du skal skrive en liten databaseapplikasjon for den nye, forbedrede håndholdte PDA-en (Personal Data Assistent) Hand-8 (H8). H8 har en begrenset mengde RAM som kan brukes for data. Den reelle mengden RAM du har lov å bruke, er 12400 bytes. RAM-typen valgt for dette produktet er designet for å bruke lite energi og er derfor ganske langsom. RAM leses i blokker av 100 byte og hver slik lesing tar 5 ms (0,005 sekunder). Dette er den desidert langsomste operasjonen i denne maskinen.

Du skal lage en database med en eneste tabell. Hver post (record) er eksakt 40 byte lang, og det er 200 poster. Siden kort utviklingstid er svært viktig for lønnsomheten for produktet, legger man vekt på å unngå ekstra programmering. Derfor er det ikke gjort noe for å optimere plassforbruket, så postene er lagret "unspanned" (det vil si at hele posten ligger i samme blokk). Den vanligste operasjonen er et enkelt likhetssøk på et ikke-unikt attributt (f.eks. navn = 'Lars'). Filen er ikke sortert på dette attributtet, og resultatet er den første forekomsten med riktig nøkkelverdi.

- a) Hva er gjennomsnittlig og lengste søketid for denne mest vanlige operasjonen hvis det ikke er noen indekser i systemet? (Svaret er avhengig av antall poster med denne nøkkelverdien. Det er nok å svare med den forutsetningen at nøkkelen er unik.)
- b) Hva er lengste søketid for denne operasjonen hvis du bruker en indeks? Lengden av navnefeltet som du ønsker å øke søkehastigheten på, er 10 byte. Det er ok å anta at indeksen er "unspanned".

Slutt på oppgavesettet

Lykke til!

Vera Goebel
Ragnar Normann
Lars Bratthall

Vedlegg (Relasjonell realisering av EER-modellen)

Den interessante delen av databasen er beskrevet nedenfor. Primærnøkler er angitt i forklaringene. Fremmednøkler er angitt ved at samme attributtnavn er brukt i flere tabeller.

Tabellnavn	Attributter	Forklaringer
Student	Fnr, Snavn, Hovedfag, Tlfnr, Gate, Husnr., Postnr, Poststed, Epost	Holder orden på data om studenter. Entydig identifisert av Fnr (fødselsnummer). Epost er også entydig.
Kurs	EmneID, Semester, Eksamenssted, Eksamenstid	Et emne holdt i ett semester, for eksempel IN212 (EmneID) holdt Vår2000 (Semester). Entydig bestemt av kombinasjonen av EmneID og Semester.
Kurslærer	EmneID, Semester, LærerFnr	Beskriver lærerne (ansatte og gruppelærere). Entydig bestemt av kombinasjonen av EmneID, Semester og LærerFnr.
KursGruppe	EmneID, Semester, LærerFnr, GruppeNr	Beskriver kursgruppene, f. eks. IN212-5. Entydig bestemt av kombinasjonen av EmneID, Semester og GruppeNr.
KursGruppe-Tid	EmneID, Semester, GruppeNr, GruppeTid, Sted	Beskriver hvor og når gruppen holdes. GruppeNr = 0 brukes for forelesninger. Entydig bestemt av kombinasjonen av EmneID, Semester, GruppeNr og GruppeTid.
KursGruppe-Student	EmneID, Semester, GruppeNr, Fnr	Beskriver hvilke studenter som følger hvilke kursgruppe. Entydig bestemt av kombinasjonen av alle fire feltene.
Emne	EmneID, EmneNavn, Vekttall	EmneID er entydig, f. eks. "IN212". EmneNavn er, f. eks., "Databaseteori".
KursInfo	EmneID, Semester, Kurs-Info	Beskriver administrativ kursinformasjon. Entydig bestemt av kombinasjonen av alle tre feltene.
Faglærer	LærerFnr, Lnavn, Tlfnr, Gate, Husnr., Postnr, Poststed, Epost, Arbeidssted, Tittel	Holder orden på data om faglærere (både forelesere og gruppelærere). Entydig identifisert av LærerFnr. Arbeidssted er f.eks. IFI. Tittel er f.eks. Dr.
Karakter	EmneID, Semester, Fnr, Karakter	Lagrer karakterene til alle studenter. Entydig bestemt av kombinasjonen av EmneID, Semester og Fnr.
Kursmaterieill	EmneID, Semester, MaterialID	Sier hvilke MaterialID-er som er brukt i et kurs. En MaterialID er enten et filnavn eller en web-adresse. Entydig bestemt av kombinasjonen av alle tre feltene.
Lysark	MaterialID, KapittelID	En samling lysark. Entydig bestemt av MaterialID. Hvis en bestemt MaterialID er i bruk i Lysark, er den verken i bruk i KursBLOB eller i Kurskompendium.
KursBLOB	MaterialID, Bloben, BlobType	Et lydopptak, et bilde eller en video. Entydig bestemt av MaterialID. Hvis en bestemt MaterialID er i bruk i KursBLOB, er den verken i bruk i Lysark eller i Kurskompendium.
Kurskompendium	MaterialID, Versjonsnummer, Materialnavn	En lokalt produsert tekst (Kompendium) brukt i et kurs. Entydig bestemt av MaterialID. Hvis en bestemt MaterialID er i bruk i Kurskompendium, er den verken i bruk i Lysark eller i KursBLOB. Versjonsnummer er f.eks. 1.0. Materialnavn kan være "Kompendium i IN212".
Forfatter	MaterialID, LærerFnr	Liste av forfatterne til kursmateriellet. Entydig bestemt av kombinasjonen av de to feltene.