

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i INF1300 — Introduksjon til databaser
Eksamensdag: 30. november 2012
Tid for eksamen: 09.00–15.00
Oppgavesettet er på 5 sider.
Vedlegg: ORM-ordliste, 1 side
Tillatte hjelpemidler: Halpin & Morgan: *Information Modeling and Relational Databases*. Second edition

Kontroller at oppgavesettet er komplett før
du begynner å besvare spørsmålene.

Eksamen har to deler, A (teller 60%) og B (40%). For å få bestått resultat må del A bestås. Sagt på en annen måte: Hvis del A sensureres til stryk, får man ikke bestått på hele eksamen uansett hvor godt del B er besvart.

A—Prosjektadministrasjon (vekt 60%)

I den organisasjonen vi skal studere, og du skal lage en modell for, er avdelinger ansvarlige for prosjekter. Både avdelinger og prosjekter har budsjettrammer. En budsjettamme er en totalsum penger i norske kroner (NOK). Avdelingenes og prosjektenes totalkostnader, må ikke overstige denne rammen.

Hver ansatt arbeider i en avdeling, men kan bli satt på ulike prosjekter, også prosjekter som kontrolleres av en annen avdeling. Alle prosjekter og avdelinger har navn. Ethvert prosjekt har en prosjektleder.

Det arbeid som den ansatte utfører på et prosjekt, blir omregnet i penger etter en viss timesats (i NOK). Timesatsen er bestemt av den ansattes funksjon (stilling) i organisasjonen. Antall timer multiplisert med timesatsen gir en kostnad som belastes prosjektet. Dette gjøres for alle som arbeider på samme prosjekt, slik at man til slutt har regnet ut totalkostnaden for hele prosjektet.

Enhver ansatt kan bli spurt om å skrive en eller flere rapporter om sin oppgave i prosjektet (sin del av «jobben»). Rapportene skilles fra hverandre ved at innleveringsuken påføres rapporten.

Det hender at organisasjonens ledelse vil ha oversikt over kostnadene både for avdelinger og prosjekter. Dette er for å kunne sammenlikne kostnadene med de respektive budsjettammer. Til dette trengs informasjon om kostnader pr. prosjekt pr. avdeling.

Av og til vil ledelsen også kontrollere arbeidet som utføres i prosjektene. For dette trengs informasjon om prosjektrapportene og hvilke stikkord som karakteriserer hver enkelt rapport.

Ytterligere forretningsregler:

(Fortsettes på side 2.)

- Ansatte kan utføre en jobb bare dersom deres funksjon og avdeling er kjent.
- En ansatt kan ikke både være prosjektleder og prosjektdeltaker på samme prosjekt.
- En ansatt kan gis en funksjon bare hvis funksjonens timesats er registrert.
- Før et prosjekt starter, må det være kjent hvilken avdeling som skal kontrollere prosjektet.
- De ansattes navn samt titlene på rapportene er ikke identifiserende. Alle andre numre og navn er identifiserende.
- Kostnadene til avdelinger og prosjekter kan registreres bare dersom det finnes tilsvarende budsjetttrammer.
- Summen av prosjektbudsjetttrammene må ikke overstige avdelingsbudsjetttrammen.

I oppgavene under er det viktig at det i besvarelsen skrives tydelig hvilken/hvilke deler av av ORM-diagrammet som dekker hver av oppgavene 1-3; skriv gjerne også kommentarer til. Til sammen skal modellene i de tre oppgavene dekke ovenforstående informasjon og forretningsregler. Legg vekt på å gi alle begreper en referansemåte. Alle faktatyper skal ha minst én entydighetspil.

Oppgave 1 (vekt 20%)

Lag en ORM-modell som beskriver personer, funksjoner og avdelinger i organisasjonen.

Oppgave 2 (vekt 20%)

Utvid modellen med prosjekter, budsjetttrammer og kostnader for avdelinger og prosjekter.

Oppgave 3 (vekt 5%)

Utvid modellen med rapporter og stikkord.

Oppgave 4 (vekt 15%)

Grupper den samlede ORM-modellen til et relasjonsdatabaseskjema. For hver relasjon, angi relasjonens navn og navnet til hvert attributt. Du skal ikke angi datatyper for attributtene og ikke bruke SQL i denne oppgaven. Understrek primærnøkklene. Markér andre kandidatnøkler med to streker. Undertrykte relasjoner skal ikke være med i skjemaet.

(Fortsettes på side 3.)

B—Studieaktivitetsdatabase (vekt 40%)

Studielaben vil gjerne undersøke om det er noen sammenheng mellom resultat (karakter) og studieaktivitet i noen kurs/emner ved Ifi. For å studere dette, har man tenkt å registrere informasjonen i en relasjonsdatabase. Før semesteret starter, har vi gitt følgende tabeller:

Student (bnavn, fdato, pnr, navn, studieprogram, adresse, postnr, sted, kjønn)

Opptak (bnavn, emne)

Ukeoppg (emne, oppgaveid)

Oblig (emne, oppgaveid)

Relasjonen Student inneholder informasjon om studentene. bnavn er studentens brukernavn (primærnøkkel), attributtparet (fdato, pnr) er fødselsnummer (kandidatnøkkel). Opptak inneholder informasjon om opptak til kurs (emner), emne er emnekoden, f.eks. 'INF1100'. Ukeoppg inneholder alle ukeoppgaver for alle emner. oppgaveid identifiserer den enkelte ukeoppgave og er på formen ukenr_oppg, f.eks. står tuppelet ('INF1000', '37_5c') for oppgave 5c gitt i uke 37 i emnet INF1000. Tilsvarende lister tabellen Oblig opp alle obliger i et emne. For INF1300 finner vi tre tupler, ('INF1300', 'oblig1'), ('INF1300', 'oblig2') og ('INF1300', 'oblig3'). Du kan forutsette at ingen emner har mer enn 7 obliger.

Arbeid med oppgaver skal registreres i tabellen

Oppgave (bnavn, emne, oppgaveid, status)

Like attributtnavn har samme betydning som over. status er en av 'fi' (forsøkt men ikke fått til), 'll' (løst), 'ld' (løst delvis), 'dg' (ikke forsøkt men deltatt på gjennomgang). De tre første skal bety at studenten har arbeidet med oppgaven. Den siste eller ingen registrering betyr at studenten ikke har arbeidet med oppgaven. For obliger (oppgaveid med verdi 'oblig1' osv.) registreres bare status som en av 'gk' (godkjent) eller 'ig' (ikke godkjent).

Øvrig studieaktivitet registreres i relasjonen

Aktivitet (bnavn, emne, dato, akt, min)

dato er dato for aktiviteten, akt (aktivitet) er en av 'f' (forelesning) 'g' (gruppe) 'u' (arbeid med ukeoppgaver) 'o3' (arbeid med oblig3) 'a' (øvrig arbeid med faget). min er tid for aktiviteten (i minutter).

Resultater for studentene i de forskjellige kursene registreres etter eksamen i tabellen

Eksamen (bnavn, emne, karakter)

Du kan anta at attributter har det domenet (der det ikke er opplagt) som er mest hensiktsmessig for spørringene, se særlig oppgavene

(Fortsettes på side 4.)

Student (bnavn, fdato, pnr, navn, studieprogram, adresse, postnr, sted, kjønn)

Opptak (bnavn, emne)

Ukeoppg (emne, oppgaveid)

Oblig (emne, oppgaveid)

Oppgave (bnavn, emne, oppgaveid, status)

Aktivitet (bnavn, emne, dato, akt, min)

Eksamen (bnavn, emne, karakter)

Oppgave 5 (vekt 5%)

a) Hvordan kan man sikre at alle studenter som registreres i Aktivitet og Oppgave har opptak til emnet det registreres aktivitet i? Bør det være slik at vi krever at studenter som har jobbet med en oppgave (registrert i Oppgave) også må være registrert i Aktivitet for dette arbeidet? Diskuter kort fordeler og ulemper ved å definere slike regler.

b) Definer tabellen Aktivitet med SQL (alle attributter er VARCHAR, bortsett fra dato (DATE) og min (INT)). Husk å definere primær- og fremmednøkler, og integritetsregler for attributter beskrevet ovenfor.

Oppgave 6 (vekt 5%)

Skriv en SQL-spørring som lister opp fullt navn for alle studenter som er tatt opp til INF1300 og som er født i april måned. *Hint: Hvis fdato er en tekst (VARCHAR) vil fdato være på formen 'xx04xx', hvor x står for et vilkårlig siffer.*

Oppgave 7 (vekt 5%)

Vi regner personer som har postnr >1299 som bosatt utenfor Oslo. Finn ut hvor mange som er tatt opp til INF1000 og til studieprogrammet som inneholder tekststrengen 'Design' som er bosatt henholdsvis i og utenfor Oslo (to heltall).

Oppgave 8 (vekt 5%)

Anta at attributtet kjønn i Student kan ha verdiene 'K', 'M', eller null. Fødselsnummeret (fdato, pnr) er laget slik at hvis det midterste sifferet i pnr er 0, 2, 4, 6 eller 8, så er kjønn 'K'. Skriv en SQL-setning som oppretter riktig kjønn (endrer verdien for attributtet kjønn) for personer der dette er null, basert på verdien av det tredje sifferet i personnummeret. Kjønn skal være 'M' hvis sifferet er 1, 3, 5, 7 eller 9. *Hint: hvis pnr er VARCHAR kan like brukes. Hvis heltall, bruk heltallsdivisjon (//) og modulofunksjonen (mod).*

(Fortsettes på side 5.)

Student (bnavn, fdato, pnr, navn, studieprogram, adresse, postnr, sted, kjønn)

Opptak (bnavn, emne)

Ukeoppg (emne, oppgaveid)

Oblig (emne, oppgaveid)

Oppgave (bnavn, emne, oppgaveid, status)

Aktivitet (bnavn, emne, dato, akt, min)

Eksamen (bnavn, emne, karakter)

Oppgave 9 (vekt 5%)

Skriv ut brukernavnene til studenter som har gjort alle ukeoppgavene (status med verdi 'fi', 'll' eller 'ld') i INF1000.

Oppgave 10 (vekt 5%)

Skriv ut antall studenter i hvert emne som bare har gjort obliger og som har strøket (karakter har verdi 'F').

Oppgave 11 (vekt 10%)

Lag en oversikt over alle emner. For hvert emne skriv en linje for hver karakter (A-F) og bak karakteren antall som fikk denne, deretter antall som bare har gjort obliger, så gjennomsnitt antall ukeoppgaver de som fikk karakteren har gjort og til slutt gjennomsnittlig hvor mange minutter de som fikk karakteren har arbeidet med emnet. Hver linje vil først ha emnekoden, så en karakter, deretter fire tall. Alle tallene gjelder karakteren og emnet først på linja.

Bonusspørsmål

Nøtt til dem som ikke synes det er nok å gjøre. Ikke forsøk på denne med mindre du er helt ferdig med oppgavene ovenfor. (De som får den til, får ekstra uttelling for dette. De som ikke prøver seg på den, eller ikke klarer den, taper ikke noe på det karaktermessig.)

Skriv en SQL-spørring som skriver ut brukernavnet til de som har «hull i» obliggodkjenningene, dvs. har en senere oblig godkjent og det ikke er registrert godkjenninger i alle forutgående (tidligere) obliger. (F.eks. hvis oblig3 er godkjent, må også 1 og 2 være det. Hvis begge eller en av dem mangler, er det et «hull»)

Oppgavesett slutt. Lykke til!

Dumitru Roman og Stein Michael Storleer