

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Andre deleksamen i :	INF1300 – Introduksjon til databaser
Eksamensdag :	Onsdag 10. desember 2008
Tid for eksamen :	14.30 – 17.30
Oppgavesettet er på :	2 sider
Vedlegg :	Ingen
Tillatte hjelpemidler :	Halpin & Morgan: <i>Information Modelling and Relational Databases</i> . Second Edition.

Scenario: Timeregnskap

I en organisasjon fører de ansatte hver dag ved arbeidshagens slutt timer på de prosjektene de har arbeidet på i løpet av dagen. Til dette brukes følgende tabell (primærnøkkelen er understreket):

Prosjekttimer(ansattID, prosjektID, dato, timer, status, godkjennerID)

Ved begynnelsen av hver ny uke går enten prosjektleder eller en stedfortreder igjennom samtlige nye timeanførsler for sitt prosjekt. Nye timer kan gjenkjennes ved at de har "ny" i attributtet status (og "null" i attributtet godkjennerID). De timeanførslene som godkjennes, får endret status til "godkjent". For de timene som ikke godkjennes, settes status til "merknad", og den prosjektmedarbeideren som har ført timene, får en melding om å sjekke og eventuelt rette timeantallet. Uansett legges ansattID for den som gjennomgikk timelistene, inn i godkjennerID. Når timetallet for de postene som fikk en merknad er revurdert av medarbeideren, kan status bli satt til "attestert" av prosjektleder, og prosjektleders ansattID legges i godkjennerID. Det er bare prosjektleder selv som kan endre status på de postene som har fått en merknad. Dette betyr at prosjektID, dato og status sammen bestemmer verdien i godkjennerID.

Oppgave 1 (10%) Bruk SQL til å lage en liste over prosjekttimer med status "merknad". Hver linje skal bestå av prosjektID, ansattID, dato og antall timer. Sorter listen på prosjektID og ansattID.

Oppgave 2 (15%) Angi hvilke funksjonelle avhengigheter (FDer) som gjelder i Prosjekttimer, og gi et begrunnet svar på hvilken normalform tabellen har.

Scenario: Maskindatabasen på Ifi

Det finnes mange mulige løsninger på hvordan maskindatabasen skal utformes. Vi skal her ta utgangspunkt i én mulig løsning og se på noen av de tabellene som inngår i denne løsningen:

Leverandør(navn, adresse, e-post, sentralbord, supporttelefon)
Ordre(leverandør, ordrenr, dato)
Ordrelinje(leverandør, ordrenr, linjenr, varenr, antall)
Faktura(leverandør, fakturanr, ordrenr, leveringsdato, beløp, konto)
Fakturalinje(leverandør, fakturanr, linjenr, varenr, antall, stykkpris, konto)
Detaljinje(leverandør, fakturanr, linjenr, antall, konto)

Her er primærnøkklene understreket.Attributtene med navn konto sier hvilken konto i universitetsregnskapet som skal belastes. I **Faktura** skal konto alltid oppgis. I **Fakturalinje** skal konto bare oppgis hvis den er forskjellig fra konto i **Faktura**. Tabellen **Detaljinje** brukes bare når deler av en fakturalinje skal belastes en annen konto enn resten av fakturalinjen. Husk at hver ordre kan bli oppfylt med flere delleveranser og at det er eksakt én faktura pr. delleveranse.

- Oppgave 3 (10%)** Gå ut fra at både leverandør, fakturanr og konto har datatype VARCHAR(30). Definer tabellen **Detaljinje** med SQL. Husk å definere primær- og fremmednøkler.
- Oppgave 4 (10%)** Beskriv fremmednøkkelene fra **Fakturalinje** til **Faktura** og forklar hva databasehåndteringssystemet må gjøre for å håndheve den.

De følgende oppgavene skal løses med SQL. Det er lov å bruke views.

- Oppgave 5 (10%)** Lag en liste over navn og e-postadresse på leverandører og det totale beløp de har fakturert for i år. Sorter listen slik at de som har solgt mest, kommer først.
- Oppgave 6 (10%)** For en gitt faktura, kontroller at totalbeløpet på fakturaen stemmer med det som er oppgitt på fakturalinjene. Skriv ut differansen mellom fakturaens totalbeløp og det beløpet fakturalinjene indikerer hvis tallene ikke stemmer overens.
- Oppgave 7 (15%)** Lag en liste over ordre som har minst en delleveranse, men hvor ikke hele ordren er levert.
- Oppgave 8 (20%)** For en gitt konto, finn ut hvor mye den er belastet for i år.