



Høgskolen i Østfold

EKSAMEN

Emnekode: ITF30307	Emne: Databaseadministrasjon og databasesystemer
Dato: 12.12.10	Eksamenstid: 09.00 - 12.00.
Hjelpemidler: ingen	Faglærer: Edgar Bostrøm / Lars V. Magnusson
Oppgavesettet består av 2 sider. På mange av oppgavene kan det lønne seg å svare punktvis. I noen tilfelle holder det med en setning eller tre, i andre tilfelle bør det gjøres en beskrivelse/kommentar/drøfting på hvert av disse punktene. De delene hvor det står Kort vil tillegges noe mindre vekt.	

***** Løsningsforslag – bare noen stikkord i all hast. En besvarelse bør være en god del mer omfattende. *****

Oppgave 1

En butikkjede vil opprette en ny database som skal benyttes både sentralt og i alle filialene. Systemet skal lagre data om varer, salg, kunder m.m. Siden butikkjeden er avhengig av at hvert salg går så raskt som mulig, må kunderegistreringen gjøres så effektiv som mulig. Samtidig skal data som er ferdig behandlet, automatisk kopieres over i andre systemer.

Forklar hvordan dette kan implementeres

1. Databasemessig, f.eks. sentralisert, distribuert, replikert

- **sentralisert**: er sikre på at det ikke blir duplikater, ingen problem med autoinkrementelle nøkler m.m., men kan gå tregt hvis ikke nettverket er svært raskt. Kopiering til andre systemer er også enkelt.
- **distribuert** (begrepet kan tolkes som fullstendig (alle data kun 1 sted) eller delvis distribuert (noe redundans): kan bli problemer som nevnt over, og f.eks. sjekk på FK må i verste fall gjøres distribuert. Ellers i utgangspunktet raskere hvis netthastighet kan være et problem. Men: mer komplisert, og mer komplisert mht. å få data over til andre systemer. Man kunne bruke 2PC – tofase-commit, men det har en del ulemper (se læreboka) – dette var ikke forelest.
- **replikert** (her kanskje mest naturlig å tenke på fullstendig replikasjon og asynkront, men med rask synkronisering): må tenke på autoinkrementell og faren for like data flere steder, men bør gå raskt mht. registreringen hvis denne skjer lokalt. Replikering gjør at de andre nodene får riktige data raskt (men ikke samtidig), og at en master kan sørge for viderekopiering.
- **kombinasjoner av disse**.

a) med hensyn til behandlingslogikk, f.eks. om behandlingen skal foregå i databasen (triggere m.m.), på applikasjonssiden eller på andre måter.

Noe av poenget er å si noe fornuftig om hvor behandlingslogikk bør ligge (i database, mellomvare, applikasjon).

- **triggere**: kan lage automatisk overføring til andre systemer, men umulig med unntak – det har både fordeler og ulemper.
- **mellomvare**: kan også definere automatisk overføring osv. Kan være flere lag med mellomvare. Jfr. gjesteforelesning.
- **applikasjonssiden**: også her kan man definere automatisk overføring, men er det flere applikasjoner mot samme base, må man gjøre det samme i alle.
Bla. Bla. Bla. Bla bla bla. Bla bla bla.

Oppgave 2

- a) Kjeden skal ansette en databaseadministrator. Hva er vedkommendes arbeidsoppgaver, hvilken kompetanse bør vedkommende ha, og hvilke hjelpemidler bør vedkommende ha til rådighet? Trekk gjerne inn momenter fra gjesteforeleser.
- her står det mye i boka. Under «hjelpemidler» bør man i tillegg nevne oppbygging av rutiner, monitoreringsverktøy og ikke minst kaffe ☺, jfr. gjesteforelesningen. Note: da jeg sensurerte dette, var kaffe det som flest hadde husket å skrive om!
- b) **Kort:** De vurderte også å ansette en dataadministrator, mens andre sa at «en dataadministrator og en databaseadministrator er jo det samme». Hva er i tilfelle forskjellen?
- **Stikkord:** DA som administrator av alle typer data (også manuelle), med vekt på det administrative og strategiske. DBA som i utgangspunktet mer tekniker, f.eks. optimalisering av DBMSer. Ofte vil DBAer i praksis involveres i data som strategisk ressurs, men har kanskje ikke kompetanse på det.

Oppgave 3

- a) Det ble diskutert om kjeden også trenger et datavarehus. Hva ville det kunne bruke det til?
- b) Sammenlign karakteristika for en «vanlig» database med karakteristika ved et datavarehus.
- c) **Kort:** Hva er ETL? Beskriv denne prosessen.

Hele oppgave 3: Se læreboka. Noe finnes også i forelesningsnotatene.

Oppgave 4

- a) Et slikt kundesystem vil opplagt trenge å utføre setninger av typen
- $$\pi_{\text{kundenr}, \text{kundenavn}, \text{ordredato}}(\sigma_{\text{filialnr}=719}(\sigma_{\text{ordrestatus}=\text{"Betalt"}}(\sigma_{\text{kunde.kundenr} = \text{ordre.kundenr}}(\text{Kunde} \times \text{Ordre}))))$$
- Forklar hva som skjer i denne setningen.
- Vi leser fra innerste parentes:
- Kunde × Ordre er et produkt, gir alle kombinasjoner mellom relasjonene, masse unødvendige data
 - $\sigma_{\text{kunde.kundenr} = \text{ordre.kundenr}}$ -restriksjon plukker ut de med like kundenr. Til sammen: eqvjoin.
 - De to andre restriksjonene plukker ut utvalgte ordrestatus og filialnr.
 - Til slutt gjøre det en prosjeksjon, slik at vi bare får utvalgte kolonner (kundenr,)
 - Til sammen: Man finner kundenr, kundenavn, ordredato for betalte order i filial nr. 719.
- b) En SQL-setning går gjennom en optimaliseringsfase før den utføres. Forklar hva som gjøres i denne fasen, helst ved å ta utgangspunkt i setningen over som en del av forklaringen. Begrepene fra relasjonsalgebra skal brukes i denne forklaringen.
- Se læreboka og notater. Bruk av disse prinsippene gjør at man bør omformulere til
- $$\pi_{\text{kundenr}, \text{kundenavn}, \text{ordredato}}(\sigma_{\text{filialnr}=719}(\text{Kunde} \bowtie \sigma_{\text{ordrestatus}=\text{"Betalt"}}(\text{Ordre}))), \text{ evt. projisere tidlig, dvs.}$$
- $$\pi_{\text{kundenr}, \text{kundenavn}}(\sigma_{\text{filialnr}=719}(\text{Kunde} \bowtie \pi_{\text{kundenr}, \text{ordredato}}(\sigma_{\text{ordrestatus}=\text{"Betalt"}}(\text{Ordre}))))$$
- c) **Kort:** Det er vanlig å snakke om regelbasert vs. kostbasert optimalisering. Hva ligger i dette?
- Se læreboka. Dette var ikke tatt opp i forelesningene, men var gitt som oppgave i en av innleveringene.

Oppgave 5

- a) Forklar forskjellen mellom et RDBMS, et ORDBMS og et OODBMS.
- Se læreboka og notater.
- b) **Kort:** Hva ville du anbefale for denne butikkkjeden? Begrunn svaret.
- RDBMS. De skal lage et standard system som ikke trenger noe spesielt innen OO. Da er RDBMSer velprøvde, standardiserte, og mange har kompetanse på det.
- c) **Kort:** Er XML-basert lagring et godt alternativt for dem? Begrunn svaret.
- Neppe. her er det antagelig svært strukturerte data, og mange av de samme argumentene som er nevnt over gjelder her også. Hastighet taler også imot XML. Muligens bør XML brukes til dataoverføringen nevnt i innledningen. Oppgaven spør imidlertid om lagring, ikke dataoverføring.

Oppgave 6

Det finnes flere ulike metoder for å koble seg til en database når man lager applikasjoner som er avhengig av data fra et DBMS. Forklar hvilke metoder vi har tilgjengelig, og hvilke fordeler og ulemper de har.

- Se lærebok og notater.