



# Høgskolen i Østfold

## EKSAMEN

Emnekode: ITF30307	Emne: Databaseadministrasjon og databasesystemer	
Dato: 04.12.13	Eksamenstid: 09.00 - 12.00.	
Hjelpemidler: ingen		Faglærer: Edgar Bostrøm / Per O. Bisseberg
Oppgavesettet består av 2 sider. Vedlegget består av 1 side.		
På mange av oppgavene kan det lønne seg å svare punktvis. I noen tilfeller holder det med en setning eller tre, i andre tilfeller bør det gjøres en beskrivelse/kommentar/drøfting på hvert av disse punktene.		
Tidsangivelsen pr. oppgave gir indikasjon på hvor mye man bør svare. Hver deloppgave teller likt.		
Sensurdato: <u>8. januar 2014</u>		
Karakterene er tilgjengelige for studenter på studentweb senest dagen etter oppgitt sensurfrist.		

### Oppgave 1. Tid: 60 minutter.

- a) Hvilke oppgaver har en databaseadministrator? Trekk gjerne inn stoff fra gjesteforeleser (utover det å drikke mye kaffe☺).  
**Kort:** Sammenlign rollen til en databaseadministrator og en dataadministrator.
- b) Hva er replikering, og hvilke fordeler og ulemper gir replikering i forhold til ikke-replikerte systemer?
- c) Hva menes med recovery av en database? Boka (og delvis også forelesningsnotatene) beskriver ulike teknikker for recovery – forklar!

### Oppgave 2. Tid: 60 minutter.

For delspørsmål a) og b) tar vi utgangspunkt i en forenklet ordre-ordrelinje-vare-struktur.

**ORDRE**  
**Ordrenr** Ordredato Kundenr

**VARE**  
**Varenr** Varenavn

**ORDRELINJE**  
**Ordrenr Varenr** Antall

- a) Lag utsagn i relasjonsalgebra for:
- Kundenr for den/de som har kjøpt 100 stk. "Grønne spikermatter" i en og samme ordre. Bruk natural join, og slik at du kobler sammen alle tabellene først («innerst»).
  - Samme som over, men vi ønsker i stedet å **ha med alt i relasjonen Ordre**, og vi ønsker at det skal gjøres **effektivt**. Tips: det kan også være lurt å bruke noe annet enn natural join (helt eller delvis) for å gjøre utsagnet enklere.
- b) Lag utsagn i relasjonsalgebra for:
- Varenr og varenavn på varer som ikke er solgt i det hele tatt (dvs. at de ikke finnes på noen ordrelinje).
  - Kundenr for de som har kjøpt alle varene som finnes i vare-relasjonen.
  - **Kort:** Læreboka og andre kilder definerer opp en grupperingsoperator. Gi et eksempel på et relasjonsalgebrauttrykk som bruker denne.
- c) Begrepene fra relasjonsalgebra kan brukes for å forklare hva som gjøres når man overfører/transformerer data fra et vanlig OLTP-system til et datavarehus. Forklar!

Tips for del a) og b): Se syntaks i vedlegget.

### Oppgave 3.      Tid: 60 minutter.

- a) Forklar og sammenlign virkemåtene til triggere, lagrede prosedyrer og lagrede funksjoner.
- b) Utred fordeler og ulemper ved plassering av programlogikk på databasenivå. Hvilke alternativer har vi til plassering av programlogikk?
- c) Forelesningene tok opp ulike databasemodeller/former (bl.a. hierarkisk/nettverks-, relasjonsdatabaser og OODB) og satte disse i sammenheng. Forklar, gjerne ved bruk av en tabell. Hvordan passer XML som databasemodell inn i dette mønsteret?

## Relasjonsalgebra - vanlige operasjoner.

<b>Mengdeoperasjoner:</b>	<i>Notasjon, variant 1</i>	<i>Notasjon, variant 2</i>
Union	$R \cup S$	R union S
Snitt	$R \cap S$	R intersect S
Mengdedifferanse	$R - S$ $R \setminus S$	R difference S R minus S
Mengdeprodukt, kartesisk produkt ("alle mot alle")	$R \times S$	R product S R times S
<b>Spesielt for relasjoner:</b>		
Horisontalt utvalg	$\sigma_{\langle \text{beting.} \rangle}(R)$	R where $\langle \text{bet.} \rangle$ R where $\langle \text{bet.} \rangle$
Vertikalt utvalg	$\pi_{\langle \text{feltliste} \rangle}(R)$	$R[\langle \text{feltliste} \rangle]$
Mengdedivisjon. (Gitt $R[c,d]$ og $S[d]$ . c er med i mengden R dividert med S hvis c i R forekommer sammen med alle d-er som finnes i S. )	$R \div S$ $R / S$	R divideby S
<b>Spesialiteter av produkt:</b>		
$\theta$ -join (produkt med en eller annen betingelse på kompatible attributter, f.eks. $>$ , $<$ , og kombinasjoner	$R \bowtie_{\langle \text{bet.} \rangle} S$	$R \text{ join}_{\langle \text{betingelse} \rangle} S$  $(R \text{ join } S) \text{ where } \langle \text{bet.} \rangle$
Equi-join ( $\theta$ -operasjonen er $=$ )	som over	som over
Natural join (Equi-join hvor felles attributt kommer bare en gang) ** den mest vanlige jointypen **	som over	som over
<b>Varianter for produkt:</b>		
Outer join, normalt venstre. (alle i R, samt alle fra S som oppfyller koblingsbetingelsen)	$R \bowtie_{\text{left}} S$	$R \text{ left join}_{\langle \text{bet.} \rangle} S$
Full join (alle i R, alle i S, samt alle som oppfyller koblingsbet.)	$R \bowtie_{\text{full}} S$	$R \text{ full join}_{\langle \text{bet.} \rangle} S$
Semijoin (de i R som tilfredsstiller $R \text{ join}_{\langle \text{betingelse} \rangle} S$ )	$R \bowtie_{\langle \text{bet.} \rangle} S$	$R \text{ semijoin}_{\langle \text{bet.} \rangle} S$

Legg merke til at operasjonene her er på mengder, slik at evt. duplikater tas bort – tilsvarende select distinct i SQL.

Dersom betingelsen er på entydige primær/fremmednøkkelkombinasjoner, droppes ofte  $\langle \text{bet} \rangle$ .