Høgskolen i Østfold

EKSAMEN

Emnekode:	**Emnenavn:**
ITF301416	Store datamengder – analyse og prosessering
Dato:	**Eksamenstid:**
16.12.16	09:00 – 13:00
Hjelpemidler:	**Faglærer:**
Ingen	Edgar Bostrøm

Om eksamensoppgaven og poengberegning:

Oppgavesettet består av 4 sider inklusiv denne forsiden og et vedlegg.

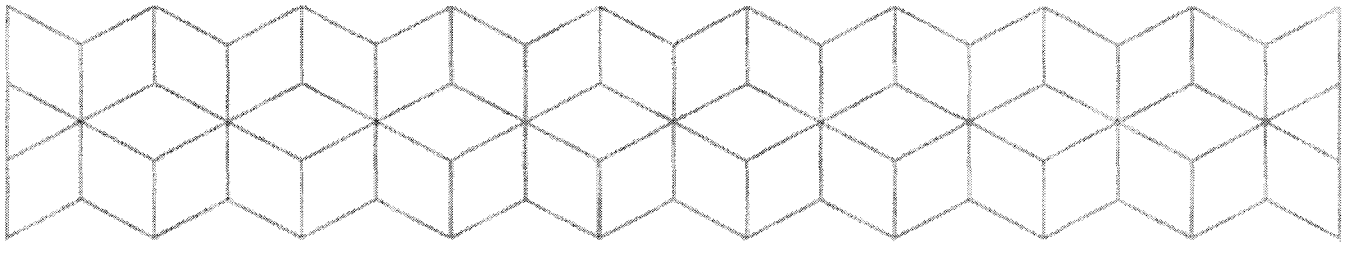
Kontroller at oppgaven er komplett før du begynner å besvare spørsmålene.

Alle de 4 oppgavene skal besvares og teller likt ved sensurering.

Les gjennom hele oppgavesettet med en gang, slik at du kan stille evt. spørsmål til hele settet når faglærer kommer.

Sensurfrist: 16.1.17

Karakterene er tilgjengelige for studenter på Studentweb senest 2 virkedager etter oppgitt sensurfrist. www.hiof.no/studentweb



Oppgave 1. Relasjonsalgebra og optimalisering. Tid: 1 time.

Vi tar utgangspunkt i samme system som i et av prosjektene som ble gitt i høst.

Prosjekt
Prosjektnr Prosjektnavn

Ansatt
Ansattnr Fødselsnr Etternavn Fornavn

ProsjektArbeide
Prosjektnr Ansattnr Timetall

a) Skriv ut hvor mange timer den ansatte med fødselsnr 15129911111 har brukt på prosjektet «Renovering av Svinesundbrua». Følgende forslag virker. Forklar hva som gjøres, hvorfor det ikke er lurt å bruke dette relasjonsalgebrautsagnet.

$\pi_{\text{Timetall}}(\sigma_{\text{Fødselsnr = 151299111111}}(\sigma_{\text{ Prosjektnavn = «Renovering av Svinesundsbrua»}}($

$\sigma_{\text{Prosjekt.Prosjektnr = ProsjektArbeide.Prosjektnr and Ansatt.Ansattnr = ProsjektAnsatt.Prosjektnr}}($

Prosjekt X ProsjektArbeide X Ansatt))))

b) Lag en mest mulig effektiv formulering av samme spørsmål som over.

c) Skriv ut prosjektnr og -navn på de prosjektene som ikke er satt i gang (dvs. prosjekter hvor det ikke finnes prosjektarbeide). Bruk helst mengdedifferanse, helst også semijoin.

d) Skriv ut Ansattnr, Etternavn, Fornavn på alle ansatte, sammen med eventuelle prosjekter de har vært med på, med Prosjektnr og Timetall, slik som vist under. Tips: outer join. Vi antar at det finnes en ***NULL*** som kan brukes på samme måte som i SQL.

Ansattnr	Etternavn	Fornavn	Prosjektnr	Timetall
712	Andersen	Anne	P_171	40
712	Andersen	Anne	P_131	70
712	Andersen	Anne	P_151	30
399	Bindersen	Binder	***NULL***	***NULL***
821	Caspersen	Casper	P_131	98
540	Davidsen	David	***NULL***	***NULL***

e) Skriv ut Ansattnr, Etternavn og Fornavn på de ansatte som har deltatt i alle prosjekter som er satt i gang.

f) I lærebøker og på nettet finner vi mange råd om «avoid using DISTINCT in SQL». Hvorfor? Og: er det en aktuell problemstilling i relasjonsalgebra? Hvorfor/hvorfor ikke?

Oppgave 2. Replikasjon. Tid: 1 time.

a) Beskriv hva distribusjon/replikasjon av data er, og hvilke fordeler og ulemper det innebærer. (Tid: 30 min.)

b) Forklar hva 2-fase-gjennomføring (2PC, 2 Two phase commit) av distribuerte transaksjoner er, og fordeler/begrensninger/alternativer til denne protokollen. (Tid: 15 min.)

c) Det finnes et kjent system innenfor Big Data som bruker datadistribusjon. Hvordan virker dette? (Tid: 15 min.)

Oppgave 3. Tid: 1 time.

a) Gi to gode grunner til fremveksten av Big Data teknologi

b) Graf-databaser er veldig gode til en spesiell bruk, hva slags? Gi gjerne et eksempel på bruk av graf-databaseteknologi.

c) Map Reduce er en sentral del av Hadoop. Gi et eksempel på problemer som Map Reduce er godt egnet for å løse, men også et eksempel på problemer Map Reduce ikke er egnet til å løse.

d) Hvorfor er Apache Spark i utgangspunktet raskere enn Apache Hadoop.

e) Sanntidsanalyse benytter seg typisk av tre tidsvinduer for å analysere data. Hva heter de og hvordan fungerer de?

Oppgave 4. Tid: 1 time.

a) Beskriv hovedelementene i en typisk Big Data Analysis / Data Science prosess.

b) Hvorfor kan det være viktig å rense data for outliers?

c) Beskriv forskjellen mellom Bias-feil og Variance-feil.

d) Tenk deg at vi har et klassifiseringsproblem med to klasser, A og B, og vi har 99000 eksempler av klasse A og 1000 eksempler av klass B i vårt datasett. Beskriv hvordan du vil gå frem for å lage og evaluere en klassifiseringsmodell, og hvordan du vil gå frem for å sammenligne to alternative modeller.

e) Beskriv hvordan k-nearest-neighbours modellen fungerer.

VEDLEGG: Relasjonsalgebra - vanlige operasjoner.

Mengdeoperasjoner:	*Notasjon, variant 1*	*Notasjon, variant -2*
Union	$R \cup S$	R union S
Snitt	$R \cap S$	R intersect S
Mengdedifferanse	R - S R \ S	R difference S R minus S
Mengdeprodukt, kartesisk produkt ("alle mot alle")	$R \times S$	R product S R times S
Spesielt for relasjoner:		
Horisontalt utvalg (sigma)	$\sigma_{<betingelse>}(R)$	R where <bet.> R where <bet.>
Vertikalt utvalg (pi)	$\pi_{<attributtliste>}(R)$	R[<attributtliste>]
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
Spesialiteter av produkt:		
θ-join (produkt med en eller annen betingelse på kompatible attributter, f.eks. >, <, og komb.)	$R \bowtie_{<betingelse>} S$	R join_{<betingelse>} S (R join S) where <bet.>
Equi-join (θ-opersjonen er =)	" "	" "
Natural join (Equi-join hvor felles attributt kommer bare en gang) ** den mest vanlige jointypen **	" "	" "
Varianter for produkt:		
Outer join, normalt venstre.. (alle i R, samt alle fra S som oppfyller koblingsbetingelsen)	$R \overset{\bowtie}{\subset} S$	R left join_{<bet.>} S
Full join (alle i R, alle i S, samt alle som oppfyller koblingsbet.)	$R \overset{\bowtie}{\supset\subset} S$	R full join_{<bet.>} S
Semijoin (de i R som tilfredsstiller R join_{<betingelse>} S)	$R \triangleright_{<betingelse>} S$	R semijoin_{<bet.>} S

Dersom betingelsen er på primær/fremmednøkkelkombinasjoner, droppes ofte <betingelse>.