Høgskolen i Østfold

EKSAMEN

Emnekode: ITF301415	Emne: Store datamengder: prosessering og analyse	
Dato: 01.12.2015	Eksamenstid: kl. 09:00 - kl. 12:00	
Hjelpemidler: Ingen		Faglærer: Edgar Bostrøm Erik Åsberg / Davide Roverso, eSmart
Oppgavesettet består av forside, 2 oppgavesider og en side vedlegg. Kontroller at oppgaven er komplett før du begynner å besvare spørsmålene. Hver av de 4 oppgavene vil telle likt ved bedømmingen.		
Sensurdato: 04.01.2016 Karakterene er tilgjengelige for studenter på studentweb senest 2 virkedager etter oppgitt sensurfrist. Følg instruksjoner gitt på: www.hiof.no/studentweb		

Oppgave 1. Tid 45 minutter.

Gitt følgende tabellstruktur (samme som i prosjekt/innlevering 1).

FYLKE
Fylkenr Fylkenavn

KOMMUNE
Kommunenr Kommunenavn Fylkenr

STEDSTYPE
StedkodeID Kodenavn

STED
StedID Stedsnavn StedkodeID Kommunenr

a) Lag et relasjonsalgebrautsagn for å finne stedID og stedsnavn for steder med kodenavn "Bru" i Hvaler kommune. Spørringen skal være godt optimalisert med hensyn til restriksjoner.

b) Finnes det stedstyper som ikke er i bruk? StedkodeID og Kodenavn skal være med. Det er et pluss hvis du bruker semijoin på en god måte for å løse denne oppgaven.

c) Finnes det kommuner som inneholder alle stedstyper? Alt om disse kommunene skal være med.

d) Hvordan kan relasjonsalgebra brukes til å forklare hva som skjer når man lager en distribuert database?

Oppgave 2. Tid 45 minutter.

a) Hva er forskjellene mellom datavarehus og tradisjonelle relasjonsdatabaser?

b) Forklar kort fordeler og ulemper med bruk av triggere.

c) Lag et eksempel på at en trigger trigger seg selv (ofte kalt «trigging i sirkel» eller «trigger hell»). Forklar hva som skjer.

d) Hva er cursorer/markører, og hva brukes de til i forbindelse med databaser?

Oppgave 3. Tid 45 minutter.

a) Big Data beskrives ofte med bokstaven V. Redegjør for de fire V'ene som beskriver Big Data.

b) Hadoop har en arkitektur bygget på to uavhengige rammeverk. Hvilke to og hva er formålet med hver av de?

c) Utfør en Map/Reduce algoritme på følgende tekst der målet er å telle antallet forekomster av hvert enkelt ord:

Deer Bear River
Car Car River
Deer Car Bear

d) Hvilken markant forskjell er det på Apache Spark og Apache Hadoop?

e) Hva er forskjellen på et Tumbling Window og et Hopping Window?

Oppgave 4. Tid 45 minutter.

a) Hva er forskjellene mellom «supervised learning» og «usupervised learning»?

b) Hva er forskjellene mellom «regression» og «classification»?

c) Forklar kort hva ligger bak begrepet «overfitting», og hvorfor det kan være et problem når man lager databaserte modeller med maskinlæring.

d) Forklar kort hva ligger bak begrepet «cross validation» og hva det brukes til.

e) Forklar hvorfor å se bare på «accuracy» kan være misvisende når man evaluerer klassifiseringsmodeller

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 $R \setminus S$	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.)	$R \overset{\bowtie}{\subset\supset} S$	R full $join_{<bet.>}$ S
Semijoin (de i R som tilfredsstiller R $join_{<betingelse>}$ S)	$R \triangleright_{<betingelse>} S$	R $semijoin_{<bet.>}$ S

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

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