

Institutt for datateknikk og informatikk

Eksamensoppgave i TDT4150 Avanserte databasesystemer

Faglig kontakt under eksamen: Jon Olav Hauglid

Tlf.: 93 80 58 51

Eksamensdato: Mandag 4. juni 2018

Eksamenstid (fra-til): 9.00 – 13.00

Hjelpemiddelkode/Tillatte hjelpemidler: D. Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Annen informasjon: Oppgavesettet inneholder 6 oppgaver. Det er angitt i prosent hvor mye hver (del-)oppgave teller ved sensur. Gjør rimelige antagelser der du mener oppgaveteksten er ufullstendig og skriv kort hva du antar. Lykke til!

Målform/språk: Bokmål

Antall sider (uten forside): 2

Antall sider vedlegg: 0

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig 2-sidig

sort/hvit farger

skal ha flervalgskjema

Kontrollert av:

14.05.18 Svein Erik Bratsberg (sign)

Dato

Sign

Oppgave 1 – Arkitektur (10 %)

- a) En «Relational Query Processor» har «Query Rewrite» og «Query Optimizer» som separate steg. Forklar kort hva hver av disse to stegene gjør og hva som er forskjellen på dem.
- b) Hvorfor er «Shared nothing» en populær parallell arkitektur for databasesystemer?

Oppgave 2 – NoSQL (20 %)

- a) Hva er de viktigste egenskapene ved et NoSQL-databasesystem?
- b) Beskriv Brewers CAP-teorem og forklar hvordan NoSQL-databaser forholder seg til dette.
- c) Hvordan komprimeres data i C-store basert på egenskapene ved datamengden i kolonnen?
- d) Beskriv hvordan «sharding» (partisjonering) og replikering foregår i MongoDB.

Oppgave 3 – Spørreoptimalisering (25 %)

- a) Kan seleksjon og projeksjon utføres i vilkårlig rekkefølge? Begrunn svaret.
- b) Anta at tre tabeller skal joines. Hvorfor kan rekkefølgen de joines i være viktig? Lag et eksempel som illustrerer dette.
- c) Anta at to tabeller skal joines. Kan indekser påvirke hvilken joinalgoritme som vil være mest effektiv? Begrunn svaret.
- d) Kan man bruke partisjonert parallellitet til å utføre join mer effektivt? Begrunn svaret.

Oppgave 4 – Distribuerte databasesystemer (20 %)

- a) Hva vil det si at oppdateringer i forbindelse med replikering utføres asynkront («lazy»)?
- b) Noen former for replikering kan medføre et «consistency» problem kalt «gamle data» («stale data»). Hva innebærer dette problemet? Hvilke former for replikering kan forårsake det? Lag et eksempel som illustrerer hvordan problemet kan oppstå.
- c) Hva er det som gjør at optimistisk samtidighetskontroll kan kalles optimistisk? Er låsing optimistisk eller pessimistisk? Begrunn svaret.

Oppgave 5 – Ranging og skyline (10 %)

- Hva er det som gjør at det er upraktisk å utføre Skyline vha. standard SQL?
- Du planlegger å dra til England for å se to av Liverpools fotballkamper neste sesong. Som en forberedelse har du hentet inn fly og hotellpriser for fire aktuelle kampsteder – se tabellene under. Beskriv steg for steg hvordan en top-2 spørring for å finne de to billigste reisemålene kan bli utført ved hjelp av Rank Join. Oppgi eventuelle antagelser du gjør.

Flypriser

Liverpool	1812
London	906
Manchester	1812
Leicester	4110

Hotellpriser

Liverpool	824
London	1580
Manchester	1219
Leicester	771

Oppgave 6 – Tema fra seminarartikler (15 %)

- Hva er det som gjør at Gorilla er spesielt effektiv til å lagre tidsserier?
- «Write-behind logging» er en foreslått protokoll for logging og gjenoppretting i databasesystemer som bruker «non-volatile memory» (NVM). Hva er det med NVM som gjør at en ny protokoll kan være gunstig?
- Er kostmodellen eller riktig estimering av f.eks. kardinaliteter, antall distinkte verdier e.l., viktigst for at spørreoptimalisatoren skal generere en effektiv utføringsplan? Har det noe å si hvor mange tabeller en spørring omfatter? Begrunn svaret.