

# T-106.5241 Distributed Databases

Examination, 14.12.2010

Please write down, on the top of each paper you return, the following information: "T-106.5241, 14.12.2010", your whole name, your student number, your degree programme, and the **total number of papers you returned**.

1. (6p) Shortly explain the following terms:

- a) vertical partitioning
- b) intraquery parallelism
- c) round-robin partitioning
- d) quorum consensus protocol
- e) transaction-processing monitor
- f) page server / data server with page shipping

2. Two-phase commit protocol:

- a) (4p) Describe the two-phase commit protocol as given by the course material (this means with the presumed abort ideology). What messages are sent between the different sites? What log messages are written? When is the log forced onto disk? Assume that the transaction commits successfully, and that none of the sites crash.
- b) (2p) What is the purpose of the *done* message that is sent at the end of the two-phase commit protocol?

3. Procedural replication:

- a) (2p) Explain how procedural replication differs from other replication methods?
- b) (2p) How does eager procedural replication work? What about lazy procedural replication?
- c) (1p) Give an example of a situation in which eager procedural replication is a good choice for a replication method.
- d) (1p) Give an example of a situation in which you should not use procedural replication (neither eager nor lazy).

4. a) (4p) Give an example of a query that is more efficiently calculated by using semijoins (optimizing with semijoins, planning with semijoins). Explain the situation and give the corresponding SQL query. Describe shortly how your query is executed, and explain where the semijoin is calculated.

b) (2p) How does asymmetric fragment-and-replicate join work?

5. How would you parallelize the following queries in a distributed database system? The values of the attribute  $A$  are uniformly distributed in the range  $[0..1000]$ . The relations  $r(\underline{A}, B)$  and  $s(\underline{A}, C)$  have been range-partitioned (using the same partitions) into all the processors. The results of the queries are needed at a single processor.

- a) (1p) **select count(distinct A) from r.**
- b) (2p) **select distinct B from r where A > 500.**
- c) (1p) **select count(\*) from r, s where r.A = s.A and B < 20 and C < 50.**
- d) (2p) **select distinct C from r, s where r.A = s.A and B < 20.**

# T-106.5241 Hajautetut tietokannat

Tentti, 14.12.2010

Kirjoita jokaisen palauttamasi paperin ylälaitaan selvästi: "T-106.5241, 14.12.2010", koko nimesi, opiskelijanumerosi ja koulutusohjelmasi sekä palauttamiesi **paperien kokonaismäärä**.

1. (6p) Selitä lyhyesti seuraavat käsitteet:

- pystysuora osittaminen (vertical partitioning)
- kyselyn sisäinen rinnakkaisuus (intraquery parallelism)
- kiertovuoro-ositus (round-robin partitioning)
- päätösvaltakäytäntö (quorum consensus protocol)
- transaktiomonitori (transaction-processing monitor)
- sivupalvelin (page server / data server with page shipping)

2. Kaksivaiheinen sitoutumiskäytäntö (two-phase commit protocol):

- (4p) Kuvaa kaksivaiheinen sitoutumiskäytäntö kurssimateriaalissa esitetyllä tavalla (eli keskeytysoletukseen, presumed abort, perustuen). Mitä viestejä eri pisteiden välillä liikkuu? Mitä lokikirjauksia tehdään? Milloin loki pakotetaan levyille? Oleta, että transaktio sitoutuu onnistuneesti eikä mikään piste romahda.
- (2p) Mihin kaksivaiheisen sitoutumiskäytännön lopuksi lähetettävää *done*-viestiä käytetään?

3. Proseduraalinen toisintaminen (procedural replication):

- (2p) Selitä, miten proseduraalinen toisintaminen eroaa muista toisintamismenetelmistä.
- (2p) Miten toimii innokas proseduraalinen toisintaminen? Entä laiska proseduraalinen toisintaminen?
- (1p) Anna esimerkki tilanteesta, jossa kannattaa käyttää innokasta proseduraalista toisintamista.
- (1p) Anna esimerkki tilanteesta, jossa ei kannata käyttää proseduraalista toisinnusta (innokasta tai lais-kaa).

4. a) (4p) Anna tilanne ja SQL-kysely, jonka laskemisessa kannattaa käyttää puoliliitosoptimointia (optimizing with semijoins, planning with semijoins). Kerro lyhyesti, miten kyselysi lasketaan ja missä kohdassa tehdään puoliliitos.

b) (2p) Miten toimii epäsymmetrinen paloittelu- ja toisinnusliitos (asymmetric fragment-and-replicate join)?

5. Miten seuraavat kyselyt kannattaa rinnakaistaa rinnakaistietokannassa? Attribuutin  $A$  arvot ovat jakautuneet tasaisesti välille  $[0..1000]$ . Relaatiot  $r(\underline{A}, B)$  ja  $s(\underline{A}, C)$  on osaväliositettu (range-partitioned) samoilla osaväleillä kaikille prosessoreille. Kyselyn tulos halutaan yhdelle prosessorille.

- (1p) **select count(distinct A) from r.**
- (2p) **select distinct B from r where A > 500.**
- (1p) **select count(\*) from r, s where r.A = s.A and B < 20 and C < 50.**
- (2p) **select distinct C from r, s where r.A = s.A and B < 20.**