

Kirjoitathan jokaiseen palauttamaasi paperin ylälaitaan selvästi 'T-106.5241, 18.12.2012', koko nimesi, opiskelijanumerosi, koulutusohjelmasi sekä montako paperia palautit yhteensä. Laskin on sallittu mutta sitä ei tarvita. *Please write on top of every answer sheet clearly 'T-106.5241, 18.12.2012', your full name, your student nr, your study program, and the total nr of sheets that you returned. Calculators are allowed but are not needed.*

1. Määritä lyhyesti ja selkeästi seuraavat käsitteet a-d. *Explain briefly and clearly the following terms a-d.*
 - (a) hajautettu tietokanta *distributed database* (b) transaktiomitoittuvuus *transaction scaleup*
 - (c) operaation sisäinen rinnakkaisuus, anna esimerkki *intraoperation parallelism*, given an example
 - (d) globaalinen atomisuus ja yksivaiheinen sitoutumiskäytäntö
global atomicity and the one-phase commit protocol
 - (e) Vertaa lyhyesti yhteismuistijärjestelmää yhteislevyjärjestelmään skaalautuvuuden, vikasietoisuuden sekä puskurieheyden kannalta. *Briefly contrast a shared memory architecture with a shared disk architecture from the point of view of scalability, fault-tolerance and buffer consistency.*
2. Tietokantakaaviomme koostuu kahdesta relaatiosta, avaimet alleviivattuina: *Our database schema consists of two relations, with keys underlined:*
 STAFF (StaffId, FirstName, LastName, DateJoined, ManagerId)
 MANAGERS (ManagerId, DateAppointed, DepartmentID)

Relaatio STAFF, jossa 12 000 riviä, sijaitsee pisteessä S_1 ja relaatio MANAGERS, jossa 200 riviä, pisteessä S_2 . *Relation STAFF, with 12 000 rows, is located at site S_1 and relation MANAGERS, with 200 rows, at site S_2 . Halutaan suorittaa pisteessä S_3 seuraava SQL kysely: We want to run the following SQL-query at Site S_3 :*

```
SELECT S.StaffId, S.FirstName, S.LastName, M.DateAppointed
FROM Staff S, Managers M WHERE S.ManagerId = M.ManagerId
```

- (a) Halutaan laskea ensin puoliliitos pisteessä S_1 . Mitä tulisi ensin lähettää pisteeseen S_1 ? Anna SQL-kysely. *We want to compute first the semi-join at Site S_1 . What should be first shipped to S_1 ? Give the SQL-query.*
 - (b) Olkoon F' monikkojoukko, joka lähetettiin pisteeseen S_1 . Kirjoita SQL-Select lauseke puoliliitokselle, joka käyttää saatua F' . Mihin pisteeseen näin saadut tulokset lähetetään? *Let F' be the tuples sent to S_1 . Write the SQL-Select clause for a semi-join that makes use of F' . To what site do we send the result set thus obtained?*
 - (c) Olkoon F'' edellisen puoliliitoksen tulosjoukko. Kirjoita lopullinen Select-lause, joka suoritetaan seuraavaksi, ja jossa hyödynnetään F'' . Lopputulos voidaan näin lähettää pisteeseen S_3 . *Let F'' denote the result set obtained from the previous semi-join. Write a Select-clause that should be performed next, making use of F'' . The final result can now be shipped to S_3 .*
3. Kohdat (a)-(d) liittyvät kaksivaiheiseen sitoutumiskäytäntöön (2PC), 'Presumed on Abort' muunnelmaan. Olkoon transaktion koordinaattorina C, ja kolmen alitransaktioiden osallistujina P_1 , P_2 sekä P_3 . *Items (a)-(d) refer to the two-phased commitment protocol (2PC), specifically its 'Presumed on Abort' version. Let the coordinator of the transaction be C and denote its three participants P_1 , P_2 and P_3 for the three subtransactions.*
 - (a) Kun koordinaattorin lisäksi on kolme osallistujaa, montako viestiä liikkuu *yhteensä* transaktion sitouttamista varten, jos 'done' kuittauksia ei lasketa? Montako näistä viesteistä on peräisin koordinaattorilta? *When in addition to the coordinator there are three participants, how many messages are exchanged in *total* to commit the transaction, if the 'done' acknowledgements are excluded? How many of these messages are sent by the coordinator?*
 - (b) Oletetaan nyt, että P_1 *ei* voi sitouttaa alitransaktiotansa. Jääkö P_1 odottamaan päätöstä C:ltä? Voiko se vapauttaa lukkonsa? Miksi tai miksi ei? *Assume now that P_1 *cannot* commit its subtransaction. Does P_1 need to wait for a decision from C? Can it release its locks? Why or why not?*