

Kirjoitathan jokaiseen palauttamaasi paperin ylälaitaan selvästi 'CS-E4630, 14.02.2017', 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 'CS-E4630, 14.02.2017', 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-e. Explain briefly and clearly the following terms a-e.

- (a) hajautettu tietokanta *distributed database*
- (b) erämitoituvuus *batch scaleup*
- (c) vahva keskinäinen johdonmukaisuus *strong mutual consistency*
- (d) keskinopea (toiseksi hitain) sivun levityskäytäntö yhteislevyjärjestelmässä
the medium page propagation scheme in a shared-disk system
- (e) johdettu vaakasuora ositus *derived horizontal fragmentation*

2. Tietokantakaavioomme kuuluu kaksi seuraavaa relaatiota, avaimet alleviivattuina:

Our database schema contains the following two relations, with keys underlined:

EMPLOYEES (EmployeeId, FirstName, LastName, DateJoined, RetirementId, AddressId)

RETIRED (RetirementId, DateRetired, RetirementPlan, BenefitsID)

Relaatio EMPLOYEES (10 000 riviä), sijaitsee pisteessä S_1 ja relaatio RETIRED (300 riviä), pisteessä S_2 .
Relation EMPLOYEES (10 000 rows), is located at site S_1 and relation RETIRED (300 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 E.EmployeeId, E.FirstName, E.LastName, R.DateRetired
FROM Retired R, Employees E WHERE R.RetirementId = E.RetirementId
```

- (a) Halutaan laskea ensin puoliliitosta 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-kysely puoliliitokselle, joka käyttää saatua F' . Mihin pisteeseen näin saadut tulokset lähetetään?
Let F' be the tuples sent to S_1 . Write the SQL query 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 SQL-kysely, 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 the SQL query that should be performed next, making use of F'' . The final result can now be shipped to S_3 .
- (d) Jos ' R ' viittaa relaatioon RETIRED ja ' E ' viittaa relaatioon EMPLOYEES alla olevassa yhtälössä (1), niin kuvaako kyseinen yhtälö (1) oikein puoliliitosta kuten sitä tässä sovellettiin? Perustele vastauksesi.
Eq. (1): $R \bowtie E = R \bowtie (E \times R)$
If ' R ' denotes relation RETIRED and ' E ' denotes relation EMPLOYEES in the above equation (1), is equation (1) a correct representation of how the semi-join was applied in-here? Justify your answer.

3. Tämä kysymys liittyy kaksivaiheiseen sitoutumiskäytäntöön (2PC), 'Presumed on Abort' muunnelmaan. Olkoon transaktion koordinaattorina C ja kolmen alitransaktioiden osallistujina P_1 , P_2 sekä P_3 . Oletetaan, että vain P_1 voi sitouttaa alitransaktionsa, kun taas P_2 ja P_3 *eivät* voi sitouttaa alitransaktiotansa.
This question relates 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. Assume that only P_1 can commit its subtransaction, while P_2 and P_3 are *not* able to commit their subtransactions.

- (a) Mitä viestejä koordinaattori C lähettää osallistujille P_1 :lle ja P_2 :lle? Entä mitä viestejä osallistujat P_1 ja P_2 lähettävät C :lle? Oletetaan, että mikään piste ei romahda.
What messages get sent by the Coordinator C to participants P_1 and P_2 ? What messages are sent by participants P_1 and P_2 to C ? Assume there are no crashes among the sites.
- (b) Ovatko mitkään osallistujista epävarmuustilassa äänestyksen aikana? Miksi tai miksi ei?
Do any of the participants enter a state of uncertainty during voting? Why or why not?
- (c) Jos koordinaattori C romahtaisi välittömästi sen jälkeen kun se on lähettänyt P_1 :lle viestin (*abort, T*), mitä P_1 :n tulisi tehdä vastaanotettuaan kyseisen viestin?
*If the coordinator C were to crash just after having sent P_1 the message (*abort, T*), what should P_1 do upon receiving that message?*
4. (a) Kuvaile lyhyesti miten toimii innokas missä-tahansa-päivittävä toisintaminen, ja mainitse yksi sen eduista sekä yksi sen heikkouksista.
Briefly describe how the eager update-anywhere replication works, mentioning one of its advantages and one of its disadvantages.
- (b) i) Luettele kolme kurssilla käsiteltyä rinnakkaisarkkitehtuuria ja mainitse yksi ominainen etu jokaiselle niistä.
ii) Mikä näistä kolmesta arkkitehtuureista on se, jossa kyselyn kuorman tasaaminen on kaikkein haastavinta?
iii) Entä mikä kolmesta arkkitehtuureista tukee helpoiten kyselyiden välistä rinnakkaisuutta?
i) List three parallel database-architectures covered in the course and mention one distinctive advantage for each one.
ii) For which one of these three architectures is load-balancing in a query the most challenging?
iii) Which of the three architectures most easily supports inter-query parallelism?
- (c) Käytetään sovittua päätösvaltakäytäntöä, jossa toisintojen lukumäärä n on aina parillinen ja p , lukuvallan koko, valitaan niin, että $p = n/2 + 1$.
Jos $n = 8$, mikä on pienin mahdollinen kirjoitusvallan koko q ? Salliiko kyseinen pienin q :n arvo tälle konfiguraatiolle kolmen pisteen romahtamisen? Perustele vastauksesi.
We use the Quorum Consensus protocol where the number of replica sites, n , is always even and p , the size of the read quorum, is chosen so that $p = n/2 + 1$.
If $n = 8$, what is the minimum value for the size of the write quorum q ? Will this minimum value of q allow this configuration to withstand the failure of three sites? Justify your answers.
5. Tarkastellaan asiakas-palvelin järjestelmää, joka perustuu tietoa kuljettavaan sivupalvelimeen, jossa on laiska (tarvehakuinen) sivun päivitysten levityskäytäntö.
Consider a Client-Server system based on a data-shipping page server using a lazy update (demand-driven) page propagation.
- (a) (i) Miten mainitussa asiakas-palvelinjärjestelmässä tulee toimia, kun asiakas haluaa päivittää tiettyä sivua p ja miten löytyy sivun p nykyversio? Mainitse myös miten tässä sovelletaan WAL-käytäntöä.
(ii) Miten transaktiorajat ylittävä puskurointi liittyy tähän?
(i) In the given C/S system, what is the procedure when a client needs to update a certain page p and how is the current version of page p found? Also mention how the WAL-policy gets applied.
(ii) How does intra-transaction caching relate to this?
- (b) Eräs opiskelija totesi ARIESista tällaisessa sivupalvelimessa seuraavasti: "Haasteena on se, että palvelimella voi olla lokikirjauksia päivityksille, vaikka sillä ei olisikaan hallussaan kyseisten päivitysten sivuja." Kommentoi opiskelijan väitettä ja sitä mihin se perustuu.
A student observed about ARIES in such a page-server by noting: "The challenge is that the server may have log records for updates even though it may not actually have the pages to which the updates refer to." Comment on the student's remark and on what it is based.