

T-106.5241 Hajautetut tietokannat Tentti, 12.09.2011 T-106.5241 Distributed databases Exam, 12.09.2011 1(2)

Kirjoita jokaiseen palauttamaasi paperin ylälaitaan selvästi 'T-106.5241, 12.09.2011', koko nimesi, opiskelijanumerosi, koulutusohjelmasi sekä montako paperia palautit yhteensä. Laskin on sallittu mutta sitä ei tarvita.

Write on top of every answer sheet clearly 'T-106.5241, 12.09.2011', 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 mutta selkeästi seuraavat käsitteet a-e (5p):

Explain briefly but clearly the following key-terms a-e: (5p):

[a] hajautettu tietokanta *distributed database*

[b] rinnakkainen tietokanta *parallel database*

[c] Kyselyiden välinen rinnakkaisuus *inter-query parallelism*

[d] pystysuora osittaminen *vertical partitioning*

[e] Round-robin kiertovuoro *round-robin partitioning*

[f] Kumpaan seuraavista SQL-kyselyihin round-robin menetelmä sopii parhaiten? Miksi? (1p)

Which of the following two SQL-clauses is better suited for round-robin? Why?(1p)

(I) SELECT * FROM Orders (II) SELECT orderid FROM Orders WHERE sales >= 1500 AND sales <=3000

[2] Tietokantakaaviomme koostuu kahdesta relaatiosta, avaimet alleviivattuina:

Our database schema consists of two relations, with keys underlined:

TOURS (Packageid, DurationDays, Destination, Hotelid) PRICES (Hotelid, DailyPrice, MaxDiscount)

Ote sisällöstä vain esimerkin vuoksi on alla. An excerpt for illustrative purposes only is given below.

TOURS

<u>Packageid</u>	DurationDays	Destination	Hotelid
301	3	London	LND01
302	7	London	LND01
303	3	Paris	PAR01
304	7	Paris	PAR01
305	3	Rome	RMA01

PRICES

<u>Hotelid</u>	DailyPrice	MaxDiscount
LND01	180	10
PAR01	190	15
RMA01	210	5

Relaatio TOURS sijaitsee pisteessä S_1 ja relaatio PRICES pisteessä S_2 .

Relation TOURS is located at site S_1 and relation PRICES 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 T.Packageid, P.DailyPrice* T.DurationDays, P.Hotelid FROM Tours T, Prices P

WHERE T.Hotelid = P.Hotelid

[a] Halutaan laskea ensin puoliliitos pisteessä S_1 . Mitä tulisi ensin lähettää pisteeseen S_1 ? Anna SQL-kysely. (1p)

We want to compute first the semi-join at Site S_1 . What should be first shipped to S_1 ? Give the SQL-query.(1p)

[b] Olkoon F' monikkojoukko, joka lähetettiin pisteeseen S_1 . Kirjoita SQL-Select lauseke puoliliitokselle, joka käyttää saatua F' . Mihin pisteeseen näin saadut tulokset lähetetään?(1p)

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? (1p)

[c] Olkoon F'' edellisen puoliliitoksen tulosjoukko. Kirjoita lopullinen Select-lause, joka suoritetaan seuraavaksi, ja jossa hyödynnetään F'' . Lopputulos voidaan näin lähettää pisteeseen S_3 . (2p)

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 . (2p)

[d] Miten toimii epäsymmetrinen paloittelu- ja toisinnusliitos? (2p)

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

[3] Kurssilla esitettiin kaksivaiheinen sitoutumiskäytäntö (2PC), josta 'Presumed on Abort' muunnelma.

In the course, you studied the two-phased commitment protocol (2PC), specifically the 'Presumed on Abort' version.

[a] Vertaa lyhyesti 2PC:n ja yksi-vaiheisen sitoutumiskäytäntöä. Briefly contrast 2PC with one-phased commitment. (2p)

T-106.5241 Hajautetut tietokannat Tentti, 12.09.2011 T-106.5241 Distributed databases Exam, 12.09.2011 2(2)

[b] Tietyn ajanjakson aikana osallistuja on epävarmuustilassa, jolloin se ei voi itsenäisesti päättää sitoutumisesta tai peruuttamisesta. Mistä viestistä (viesteistä) lähtien osallistuja siirtyy epävarmuustilaan? Mikä/Mitkä viesti(t) lopettavat osallistujan epävarmuustilan? (2p)

During a certain period, the participant is in an uncertainty state, during which it cannot decide by itself to commit or abort. What is(are) the message(s) that start(s) this uncertainty state for the participant and what is(are) the message(s) that end(s) this uncertainty state? (2p)

[c] Piste, joka ei pysty sitoutumaan lähettää vote (T,abort) ja lopulta sitouttaa peruutuksensa ja vie sen lokille. Miksi se voi lähettää peruutusviestin koordinoijalle *ennen* peruutuksen lokikirjausta? (1p)

*A site cannot commit, so it sends vote (T, abort) and then commits the abort and force-writes it on its log. Why can it send an abort message to the coordinator *before* the log write? (1p)*

[d] Jos kohdassa [c] osallistuja tai koordinaatiopiste romahtaisi ennen lokikirjausta, voisiko se vaikuttaa osallistujan päätökseen? (1p)

If in question [c] the participant or the coordinator were to crash, could that change the decision of the participant? (1p)

[4] Data-alkiota toisinnetaan neljässä pisteessä $\{S_1, \dots, S_4\}$, jossa jokaisella pisteellä on yksi äänimäärä, joten $n=4$.

A data item is replicated at four sites $\{S_1, \dots, S_4\}$ so that each site has one vote, so $n=4$.

[a] Olkoon nyt lukuvalian koko $p = 2$. Mikä on pienin mahdollinen toimiva kirjoitusvalta q ? (1p)

Let the read quorum size $p=2$. What is the minimum write quorum size q that can work? (1p)

[b] Jos nyt kaksi pistettä romahtaisi, voisiko silti päätösvaltakäytäntö edelleen toimia jollekin transaktiolle? (1p)

If two sites were now to fail, could a transaction still successfully get a read or write quorum? (1p)

[c] Miten yleisesti lasketaan maksimimäärä pisteitä n , jotka voivat romahtaa kun lukuvalian koko $= p$ ja kirjoitusvalian koko $= q$? *Given n sites, a read quorum size p and a write quorum size q , how do we generally find the maximum nr of sites that can fail? (2p)*

[d] Olkoon nyt $p = 1$ and $q = 4$. Leikkaavatko nyt luku- ja kirjoitusvallat toisiaan? Koska $q=n$ millä toisella nimellä tällaista luku-kirjoitusvaltaa yleisemmin kutsutaan? (2p) *Now let $p = 1$ and $q = 4$. Do the read and write quorums now overlap? By what other name is such a quorum system where $q=n$ commonly known as? (2p)*

[5] Käytössä on asiakaspalvelin, joka on transaktio-/kyselypalvelin ja toimii sivulähetyksiin perustuen, ja jossa ainoa fyysinen (pysyvä) loki on palvelimella. *Consider a client database server, which is a transaction/query server with page shipping where the only physical (stable) log is at the server.*

[a] Asiakas on saanut tarvittavat sivut tehdäkseen niihin päivitykset. Miten asiakas hoitaa näin lokitoimintonsa, eli miten hoituu asiakkaan WAL-käytäntö? (1p)

The client has received the necessary pages that it needs to update. How does the client handle its logging, that is, how is the WAL policy applied for the client? (1p)

[b] Myös asiakas-palvelin lokikirjauksissa tarvitaan LSN tai vastaava. Kannattaako asiakkaan tuottaa tämä paikallisesti vai kysellä se palvelimelta? (1p)

For each log record in a C/S environment, we need an LSN or its equivalent. Should the client produce this value locally or inquire the value from the server? (1p)

[c] Mitä palvelimen tulee WAL:in mukaan tehdä, jotta transaktio olisi sitoutunut? Mitä tarkoittaa asiakkaan 'älä-pakota käytäntö'? (2p)

What does the server need to do according to WAL before a transaction is considered committed? What does the client's 'no-force policy' mean at commitment? (2p)

[d] Voitko kuvitella tilanteen, jossa päivitetty sivu tulisi ARIESiltä ohitetuksi tapauksessa, jossa palvelin romahtaa ja käynnistyy uudelleen? (2p)

Can you think of a situation where a dirty page would be skipped by ARIES in case the server crashed and was restarted? (2p)