

*Please write on top of every answer sheet clearly 'CS-E4230, 07.04.2017', your full name, your student number, your study program, and the total nr of sheets that you returned. Calculators are allowed but are not needed. There are a total of five questions, each of which is worth six points.*

Kirjoitathan ystäväällisesti jokaiseen palauttamaasi paperin ylälaitaan selvästi 'CS-E4230, 07.04.2017', koko nimesi, opiskelijanumerosi, koulutusohjelmasi sekä montako paperia palautit yhteensä. Laskin on sallittu mutta sitä ei tarvita. Kysymyksiä on yhteensä viisi ja jokaisen kysymyksen maksimipistemääriä on kuusi.

1. Explain briefly but clearly the following terms a-e.

Määritä lyhyesti mutta selkeästi seuraavat käsitteet a-e.

- (a) *Transaction.* (b) *SQL isolation level 'Read Committed'.*
- (c) *Conditional lock request.* (d) *Phantom phenomenon.*
- (e) *WAL policy and its relation to committing a transaction.*

2. (a) Consider schedule  $S_1$  with its initial database state  $D$  as shown below. We apply the key-range locking protocol. What locks are acquired by the transactions in schedule  $S_1$  and when are those locks released under key-range locking? Show your results in a tabular format and mention what, if any, is the lock that cannot be granted.

Olkoon ajoitus  $S_1$  sekä tietokannan alkutila  $D$  kuten alla. Käytetään avainvälijulkitusta. Mitä lukkoja ajoituksen  $S_1$  transaktiot varaavat ja milloin lukot vapautetaan avainvälijulkituskäytännössä? Esitä vastauksesi taulukkomuodossa ja mainitse, mikäli jotain lukkoa ei voida myöntää.

$S_1: B_1R_1[x_1, > 2]B_2D_2[4]C_2B_3R_3[x_2, > 3]I_1[4, 2]C_1C_3$

Initial database (Tietokannan alkutila)  $D = \{(3, 0); (4, 1); (5, 2)\}$ .

- (b) Does the schedule  $S_1$  contain any isolation anomaly (dirty write, dirty read, unrepeatable read) or a phantom phenomenon? Justify briefly.

Mikä eristyyvyyss anomalia (likainen kirjoitus, likainen luku, toistokelvoton luku) tai haamuilmiö ajoituksessa  $S_1$  mahdollisesti esiintyy? Perustele lyhyesti.

- (c) Does schedule  $S_1$  contain a 'lost update'? Justify your answer.

Esiintyykö ajoituksessa  $S_1$  ns. kadonnut päivitys? Perustele lyhyesti.

3. (a) Classify each of the following statements (i)-(iv) as True (T) or False (F). If the statement is False, justify briefly why it is so.

Luokittele seuraavat väittämät (i)-(iv) joko toteksi (T) tai epätoteksi (E). Jos väittämä on epätosi, perustele lyhyesti miksi se on epätosi.

(i) Use of the Force & Steal policy will reduce the work needed by ARIES at recovery during the REDO-phase.

(ii) When using physiological logging and the No-Force & No-Steal policy, there should be no need for logical UNDO during recovery.

(iii) When using WAL together with a No-Force policy, the log buffer is forced to the log disk at commit time.

(iv) Under Force & Steal policy, dirty pages can get replaced in the buffer and so a dirty page that has just become clean should not have its entry removed from the Modified-Pages Table.

- (b) Is the following schedule  $S_2$  conflict-serializable? Justify briefly.

Onko seuraava ajoitus  $S_2$  konfliktisarjoittuva? Perustele lyhyesti.

$S_2: B_1B_2B_3R_1[x]R_3[y]W_1[x]R_2[y]W_3[x]W_2[y]R_3[y]C_1C_2C_3$ .

- (c) (i) Wait-Die and (ii) Wound-Wait are two deadlock prevention protocols.

Contrast how a transaction  $T_x$  that needs to obtain its locks on a 'hot-spot' data item  $t_i$  in a system  $S$  where other older transactions are also competing to access the same data item  $t_i$  would manage to obtain its locks using each of the prevention protocols (i) and (ii).

(i) Wait-Die ja (ii) Wound-Wait ovat kaksi lukkiutumisen estoprotokollia. Vertaile miten transaktio  $T_x$ , joka tarvitsee lukot 'hot-spot' data-alkioon  $t_i$  systeemissä  $S$ , jossa toiset vanhemmat transaktiot kilpailevat myös pääsystä samaan data-alkioon  $t_i$ , saisi tarvitsemansa lukot, kun käytetään (erikseen) estoprotokollaa (i) ja estoprotokollaa (ii).

4. The contents of the log saved on disk in a system crash are as shown below. The PageLSN value of the disk version for page  $p$  is 106. (It is assumed all the operations in the UNDO-phase can be performed physically.)

Lokin sisältö häiriön sattuessa on kuten alla. Levyversion PageLSN sivulle  $p$  on 106. (Oletetaan, että kaikki peruuutusvaiheen käänneisoperaatiot voidaan suorittaa fyysisesti.)

101:	<begin-checkpoint>	105:	< $T_1$ , B>
102:	<transaction-table, {}>	106:	< $T_1$ , I, p, $x_1$ , 15, 105>
103:	<page-table, {}>	107:	< $T_2$ , B>
104:	<end-checkpoint>	108:	< $T_2$ , I, p, $x_2$ , 3, 107>
		109:	< $T_3$ , B>
		110:	< $T_3$ , I, p, $x_3$ , 9, 109>
		111:	< $T_3$ , I, p, $x_4$ , 11, 110>
		112:	< $T_3$ , A>
		113:	< $T_3$ , $I^{-1}$ , p, $x_4$ , 110>
		114:	< $T_1$ , C>

- (a) What do we get as a result of the ARIES analysis phase?

Mitä saadaan ARIES analysivaiheen tuloksena?

- (b) Perform the ARIES REDO-phase, show your answer in a tabular format.

Suorita ARIESin TOISTO-vaihe, esitä vastauksesi taulukkomuodossa.

- (c) Perform the ARIES UNDO-phase, show your answer in a tabular format.

Suorita ARIESin PERUUTUS-vaihe, esitä vastauksesi taulukkomuodossa.

5. Consider a scenario with a database  $b$  and in it a relation  $r$ . Transaction  $T_1$  needs to read from relation  $r$  a certain tuple  $t_1$  while another transaction  $T_2$  needs to simultaneously update another tuple  $t_2$  from the same relation  $r$ .

Olkoon skenaariomme seuraava: meillä on tietokanta  $b$ , jossa on relaatio  $r$ . Transaktio  $T_1$  haluaa lukea kyseisestä relaatiosta  $r$  erään monikon  $t_1$ , kun taas toinen transaktio  $T_2$  haluaa samasta relaatiosta  $r$  samanaikaisesti päivittää toista monikkoa  $t_2$ .

- (a) What lock should  $T_2$  place on database  $b$ ? On relation  $r$ ? On tuple  $t_2$ ?

Mikä lukko  $T_2$ :n tulisi asettaa tietokantaan  $b$ ? Entä relaatioon  $r$ ? Entä monikkoon  $t_2$ ?

- (b) If transaction  $T_2$  were to place a SIX-lock on relation  $r$ , would this have any advantages/disadvantages for the read operation of transaction  $T_1$  or the update operation of transaction  $T_2$  in relation  $r$ ? Justify briefly.

Jos transaktio  $T_2$  asettaisikin SIX-lukon relaatioon  $r$ , olisiko tällä mitään etuja/haittoja transaktio  $T_1$ :n lukuoperaation tai transaktio  $T_2$ :n päivitysoperaation kannalta relaatiossa  $r$ ? Perustele lyhyesti.

- (c) If  $T_1$  has obtained an S-lock on tuple  $t_1$ , could transaction  $T_2$  then obtain an update-mode lock (U-lock) on the \*same\* tuple  $t_1$ ? Justify briefly. What is the primary purpose of the U-lock?

Jos  $T_1$ :lle on myönnetty S-lukko monikkoon  $t_1$ , voitaisiinko tuolloin  $T_2$ :lle myöntää ns. U-lukko \*samaan\* monikkoon  $t_1$ ? Perustele lyhyesti. Mikä on U-lukon perimmäinen tarkoitus?