

Kirjoitathan jokaiseen palauttamaasi paperin ylälaitaan selvästi 'T-106.5221, 21.05.2015', koko nimesi, opiskeajanumerosi, koulutusohjelmasi sekä montako paperia palautit yhteensä. Laskin on sallittu mutta sitä ei tarvita. Jokaisen kysymyksen maksimipistemäärä on kuusi.

Please write on top of every answer sheet clearly 'T-106.5221, 21.05.2015', 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. Each question is worth six points.

- Määritä lyhyesti mutta selkeästi seuraavat käsitteet a-e. *Explain briefly but clearly the following terms a-e.*

- (a) *Transaction* (b) *Isolation as part of ACID* (c) *Complete checkpoint*
- (d) *Physiological logging* (e) *The Update (U)-lock and its compatibility with a shared (S)-lock.*

- Olkoon ajoitus S_1 kuten alla. Käytetään avainvälikuitusta.

Let schedule S_1 be as below to which we apply the key-range locking protocol.

- (a) Tietokannan alkutila *Database state at onset* $D = \{(3,0);(4,1);(5,2)\}$
 $S_1: B_1R_1[x_1, > 1]B_2D_2[4]R_1[x_2, > 1], I_2[2]R_1[x_3, > 3]C_1C_2$.
Mitä lukkoja ajoituksen S_1 transaktiot varaavat ja milloin lukot vapautetaan
avainvälikuituskäytännössä? Esitä vastauksesi taulukkomuodossa ja mainitse mikä mahdollisesti on se lukko, jota ei voida myöntää.
What locks are acquired by the transactions in schedule S_1 and when are those locks released?
Show your results in a tabular format and mention what, if any, is the lock that cannot be granted.
- (b) Esiintyykö ajoituksessa S_1 jokin eristyvyysanomalia (likainen kirjoitus, likainen luku, toistokelvoton luku) ? Onko ajoitus mahdollinen avainvälikuituksessa? *Does schedule S_1 contain any isolation anomalies (dirty write, dirty read, non-repeatable read) ? Is the schedule possible in the key-range locking protocol?*
- (c) Seuraavaan tekstiin perustuen, kommentoi tekikö opiskelija N.N. mielestäsi oikean johtopäätöksen ajettuaan alla olevan SQL-kyselyn ja saatuaan mainitut tulokset.
Based on the following text, comment on whether you think student N.N. made the right conclusions after running the given SQL-query and getting the given result sets.

Student N.N. ran the query below at an SQL isolation level unknown to the student (the level was known to prevent dirty reads though) and obtained the following result set consisting of two tuples: $\{t_1 = 5153AB, t_2 = 5523AB\}$. Shortly after, the same query was run on the same database with the same isolation level and produced a result set with three tuples: $\{t_1 = 5153AB, t_2 = 5523AB, t_3 = 5559AB\}$

```
SELECT StudentId FROM Students WHERE Smajor = 'AstroBiology';
```

The student then concluded that the isolation level could have been Repeatable Read but definitely not Serializable.

- (a) Jos aineistomme sisältää ns. hot-spot sivun, jota useat transaktiot päivittävät toistuvasti, onko Steal/Force käytännöstä erityistä hyötyä ARIES-elvytyksen aikana? Auttaako kyseinen käytäntö pitämään sivua puhtaana? (Muistutus: sivu, joka poikkeaa levyllä olevasta sivusta on päivitetty sivu eikä likainen sivu).

If our data contains a so-called hot-spot page that is repeatedly modified by several transactions, would the Steal/Force policy provide any special benefits during ARIES recovery? Does such a policy help to maintain the page clean? (Recall that a page that differs from the page in the disk is a modified page and not a dirty page.)

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

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

- (i) Use of the Steal/Force policy eliminates the need for REDO at recovery.
- (ii) A transaction can be removed from the transactions table when all its log records and its commit record have been written to the permanent log.
- (iii) With the Steal/No-Force policy, the log is forced to disk only during checkpoints.
- (iv) Increasing the size of the database buffer can prevent the modified-pages table from becoming too large.

4. Lokin sisältö häiriön sattuessa on kuten alla. Levyversion PageLSN sivulle p on 106.

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.

105:	$\langle T_3, \text{B} \rangle$
106:	$\langle T_3, \text{I}, \text{p}, x_1, 10, 105 \rangle$
107:	$\langle T_2, \text{B} \rangle$
108:	$\langle T_2, \text{I}, \text{p}, x_2, 12, 107 \rangle$
109:	$\langle T_1, \text{B} \rangle$
110:	$\langle T_1, \text{I}, \text{p}, x_3, 23, 109 \rangle$
111:	$\langle T_1, \text{I}, \text{p}, x_4, 21, 110 \rangle$
112:	$\langle T_1, \text{A} \rangle$
113:	$\langle T_1, I^{-1}, \text{p}, x_4, 110 \rangle$
114:	$\langle T_3, \text{C} \rangle$

- (a) Mitä saadaan analyysivaiheen tuloksena? *What do we get as a result of the analysis phase?*

- (b) Suorita ARIESin TOISTO-vaihe, esitä vastauksesi taulukkomuodossa.

Perform the ARIES REDO- phase, show your answer in a table.

- (c) Suorita ARIESin PERUUTA-vaihe. *Perform the ARIES UNDO- phase.*

5. Olkoon tietokanta b ja siinä relaatio r , jossa on kaksi monikkoa t_1 ja t_2 . Transaktio T_1 , jonka aikaleima on $tmes_1 = 1100$ on kirjoittamassa monikkoa t_1 X-lukon turvin samalla kun toinen transaktio T_2 , jonka aikaleima on $tmes_2 = 9900$ on kirjoittamassa monikkoa t_2 (myös X-lukon turvin). Ennen kuin kyseiset X-lukot on vapautettu, T_1 haluaa lukea monikkoa t_2 .

Consider a database b and in it a relation r, with two tuples t_1 and t_2 . Transaction T_1 with a timestamp $tmes_1 = 1100$ is writing tuple t_1 under the protection of an X-lock while another transaction T_2 with a timestamp $tmes_2 = 9900$ is writing tuple t_2 (also under the protection of an X-lock). Before these X-locks are released, T_1 needs to read tuple t_2 .

- (a) Kun käytetään monirakeisuuslukituskäytäntöä, mitä aikomuslukkoja transaktio T_1 voi asettaa relaatioon r ? Entä tietokantaan b ? *When using multiple granularity locking, what intention locks can transaction T_1 place on relation r ? What about on database b ?*

- (b) Selitä lyhyesti kumpi protokollista, Wait-Die vaiko Wound-Wait antaa transaktiolle T_2 mahdollisuuden saada päivityksensä monikkoon t_2 valmiiksi nopeammin?

Briefly explain, which protocol, Wait-Die or Wound-Wait will allow transaction T_2 to complete its update on tuple t_2 faster?

- (c) Selitä lyhyesti ja selkeästi miten (i) Wait-Die ja (ii) Wound-Wait protokollat estävät silmukan muodostumisen wait-for-kaaviossa.

Explain briefly and clearly how the (i) Wait-Die and the (ii) Wound-Wait protocols prevent a cycle from being formed in the wait-for-graph.