

**Άσκηση 1η**  
**(αυστηρή) Ημερομηνία Παράδοσης: 05/04/2019, 16.59**

**ΘΕΜΑ 0<sup>ο</sup>**

Διαβάστε στο MySQL reference manual την ενότητα “[InnoDB Buffer Pool](https://dev.mysql.com/doc/refman/5.7/en/innodb-buffer-pool.html)” (for v. 5.7) και “[Buffer Pool](https://dev.mysql.com/doc/refman/8.0/en/innodb-buffer-pool.html)” (for v. 8.0).

(α) Γράψτε σε 5 γραμμές τι πρόβλημα υπάρχει με τα SELECT queries που δεν έχουν WHERE clause και πώς το λύνουμε.

(β) Σε οποιοδήποτε DBMS, η μνήμη που δίνουμε στους data buffers είναι κάτι μεταξύ 70%-80% της κύριας μνήμης σε ένα dedicated server. Στη MySQL αυτό αφορά τις InnoDB installations. Ψάξτε στο μηχανήμά σας πόση είναι η κύρια μνήμη και αναθέστε στη MySQL το αντίστοιχο νούμερο. Εξηγήστε στο χαρτί σας τα νούμερα και πώς το κάνατε (με ποια εντολή, ή, πιο εύκολα με ποια ρύθμιση του workbench).

(γ) Για μικρές βάσεις δεδομένων, οι οποίες χωρούν στην κύρια μνήμη ολόκληρες (όπως αυτή που έχετε εσείς στο project του μαθήματος), είναι εφικτό να καθορίσουμε τους data buffers σε κάτι σαν 10% παραπάνω από το μέγεθος της βάσης. Βρέστε το εν λόγω μέγεθος και ρυθμίστε ανάλογα με τη μνήμη του Η/Υ σας (αν δηλ., ισχύει η περίπτωση β, ή η περίπτωση γ). Εξηγήστε στο χαρτί τι κάνατε.

Θυμηθείτε ότι τα παραπάνω αφορούν σε dedicated servers – άρα, για το μηχανήμά σας που τρέχει και ένα σωρό άλλα πράγματα, πρέπει να είστε σε θέση να μειώσετε τη μνήμη αυτή όταν η MySQL δεν τρέχει. Επίσης, σε αυτή την περίπτωση, βοηθά να ορίσετε το DBMS να μη ξεκινά αυτόματα, αλλά manually ☺

**ΘΕΜΑ 1<sup>ο</sup>**

Για κάθε ένα από τα παρακάτω χρονοπρογράμματα (schedules) απαντήστε τις παρακάτω ερωτήσεις, εξηγώντας κάθε φορά την απάντηση:

- (α) Είναι το χρονοπρόγραμμα σειριοποιήσιμο σε σχέση με τις συγκρούσεις; Αν ναι δώστε ένα ισοδύναμο σειριακό πρόγραμμα.
- (β) Μπορεί να προκύψει το χρονοπρόγραμμα αυτό από πρωτόκολλο συντονισμού με κλείδωμα σε δυο φάσεις (two-phase locking);
- (γ) Μπορεί να προκύψει το χρονοπρόγραμμα αυτό από αυστηρό πρωτόκολλο συντονισμού με κλείδωμα σε δυο φάσεις (strict two-phase locking);
- (δ) Ποιο είναι το πιο αυστηρό επίπεδο απομόνωσης στην SQL από το οποίο μπορεί να έχει προκύψει ένα τέτοιο χρονοπρόγραμμα (αν μπορεί);
- S1: R1(X) R2(Z) R1(Z) R3(X) R3(Y) W1(X) W3(Y) C3 R2(Y) W2(Z) W2(Y) C1 C2  
S2: W2(A) R1(B) R2(D) W3(C) W1(B) W3(D) R3(B) R2(D) C2 C3 W1(D) R1(A) C1  
S3: R3(A) W2(B) R1(V) W3(B) R3(A) C3 R2(A) R1(Y) R2(V) W1(A) C1 C2

**ΘΕΜΑ 2<sup>ο</sup>**

Δίνονται οι παρακάτω ορισμοί:

1. *Ανανήψιμο χρονοπρόγραμμα*: ένα χρονοπρόγραμμα  $S$  στο οποίο καμία συναλλαγή  $T$  δεν επικυρώνεται (COMMIT) αν δεν επικυρωθούν όλες οι συναλλαγές  $T_i$  οι οποίες τροποποίησαν ένα δεδομένο πριν το διαβάσει η  $T$ .
2. *Χρονοπρόγραμμα χωρίς διάδοση ανακλήσεων*: ένα χρονοπρόγραμμα στο οποίο κάθε συναλλαγή διαβάζει μόνο στοιχεία που έχουν γραφεί από επικυρωμένες συναλλαγές.
3. *Αυστηρό χρονοπρόγραμμα*: ένα χρονοπρόγραμμα στο οποίο μια συναλλαγή δεν μπορεί ούτε να διαβάσει, ούτε να γράψει ένα στοιχείο  $X$ , έως ότου επικυρωθεί (ή ακυρωθεί) η τελευταία συναλλαγή που έγραψε το  $X$ .

Για το παρακάτω χρονοπρόγραμμα

R<sub>3</sub>(A) R<sub>1</sub>(B) R<sub>3</sub>(B) R<sub>2</sub>(A) R<sub>2</sub>(C) W<sub>3</sub>(A) W<sub>2</sub>(C) R<sub>1</sub>(C) W<sub>1</sub>(B) W<sub>1</sub>(C) R<sub>2</sub>(A) W<sub>2</sub>(A) W<sub>4</sub>(C)

- (i) προσθέστε πράξεις Commit για τις συναλλαγές ώστε το χρονοπρόγραμμα που προκύπτει να μην έχει δυνατότητα ανάκαμψης (non-recoverable)

- (ii) προσθέστε πράξεις Commit για τις συναλλαγές ώστε το χρονοπρόγραμμα που προκύπτει να έχει δυνατότητα ανάκαμψης (recoverable) αλλά με διάδοση ανακλήσεων (cascade aborts)
- (iii) προσθέστε πράξεις Commit για τις συναλλαγές ώστε το χρονοπρόγραμμα που προκύπτει να έχει δυνατότητα ανάκαμψης (recoverable), να μη διαδίδει τις ανακλήσεις αλλά να μην είναι αυστηρό
- (iv) προσθέστε πράξεις Commit για τις συναλλαγές ώστε το χρονοπρόγραμμα που προκύπτει να είναι αυστηρό.

Εξηγείστε την απάντησή σας.