

Οι Συναλλαγές στην SQL – Μέρος Α!

ΠΕΡΙΒΑΛΛΟΝ: [MySQL/MariaDB](#)

ΑΣΚΗΣΕΙΣ

1. Δημιουργήστε πίνακα T στο περιβάλλον MySQL:

```
CREATE TABLE T (id INT NOT NULL PRIMARY KEY, s VARCHAR(40), si SMALLINT);
```

Επιβεβαιώστε την ύπαρξή του και τη δομή του με την εντολή:
DESCRIBE T;

2. Εισάγετε τρεις νέες γραμμές στον πίνακα T και ακυρώστε την ενέργεια (συναλλαγή) με την εντολή ROLLBACK. Τι παρατηρείτε;

```
INSERT INTO T (id, s) VALUES (1, 'first');  
INSERT INTO T (id, s) VALUES (2, 'second');  
INSERT INTO T (id, s) VALUES (3, 'third');  
ROLLBACK;  
SELECT * FROM T;
```

3. Εισάγετε μία νέα γραμμή και ακυρώστε με ROLLBACK, εκτελώντας τις εξής εντολές:

```
START TRANSACTION;  
INSERT INTO T (id, s) VALUES (4, 'fourth');  
SELECT * FROM T ;  
ROLLBACK;  
SELECT * FROM T;
```

Συνεχίστε εισάγοντας μία νέα γραμμή και σχολιάστε τα αποτελέσματα των δύο SELECT:

```
INSERT INTO T (id, s) VALUES (5, 'fifth');  
ROLLBACK;  
SELECT * FROM T;
```

4. Δώστε τις εντολές:

```
DELETE FROM T;  
SET AUTOCOMMIT=0;
```

και επαναλάβετε τη σειρά των εντολών των παραπάνω 2-3, παραλείποντας την εντολή START TRANSACTION;

Σχολιάστε τα (νέα;) αποτελέσματα που προκύπτουν.

5. Συνεχίστε με τις εντολές:

```
INSERT INTO T (id, s) VALUES (9, 'will this be committed?');  
CREATE TABLE T2 (id INT);  
INSERT INTO T2 (id) VALUES (1);  
SELECT * FROM T2;  
ROLLBACK;  
SELECT * FROM T2;  
SELECT * FROM T;
```

Παρατηρείτε να έχει συμβεί κάτι το (κατ' αρχήν) μη αναμενόμενο; Αν ναι, πως το εξηγείτε;

6. Συνεχίστε με τις εντολές:

```
SET AUTOCOMMIT=1;
INSERT INTO T (id, s) VALUES (6, 'sixth');
ROLLBACK;
SELECT * FROM T;
```

Κατανοήστε και σχολιάστε: (α) τη λειτουργικότητα της εντολής SET AUTOCOMMIT=0/1, (β) τη διαφορά μεταξύ των SET AUTOCOMMIT και START TRANSACTION, (γ) την αρχική, προκαθορισμένη (default) κατάσταση της συνεδρίας εργασίας του περιβάλλοντος MySQL όσον αφορά στη διαχείριση των συναλλαγών.

7. Η σειρά και το είδος των εντολών που ακολουθούν στοχεύουν στο να διερευνηθεί το κατά πόσον ο διακομιστής MySQL εκτελεί αυτόματα εντολή ROLLBACK όταν συμβαίνουν σφάλματα κατά την εκτέλεση εντολών SQL. Σημειώστε την απόκριση του διακομιστή μετά την εκτέλεση της κάθε μίας εντολής:

```
SET AUTOCOMMIT=0;
INSERT INTO T (id, s) VALUES (2, 'Error test starts here');
-- division by zero should fail
SELECT (1/0) AS dummy FROM DUAL;
-- Now update a non-existing row
UPDATE T SET s = 'foo' WHERE id = 9999 ;
-- and delete an non-existing row
DELETE FROM T WHERE id = 7777 ;
SELECT * FROM T;
--
INSERT INTO T (id, s) VALUES (2, 'Hi, I am a duplicate');
INSERT INTO T (id, s) VALUES (3, 'How about inserting too long of a string value?');
INSERT INTO T (id, s, si) VALUES (4, 'Smallint overflow for 32769?', 32769);
SHOW ERRORS;
SHOW WARNINGS;
```

```
INSERT INTO T (id, s) VALUES (5, 'Is the transaction still active?');
SELECT * FROM T;
COMMIT;
```

Σχολιάστε τα αποτελέσματα της εκτέλεσης των παραπάνω. Έχει προκληθεί αυτόματη εκτέλεση ROLLBACK στο διακομιστή;

8. Χρήση της εντολής GET DIAGNOSTICS:

```
INSERT INTO T (id, s) VALUES (2, 'Hi, I am a duplicate');
GET DIAGNOSTICS @rowcount = ROW_COUNT;
GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE,
@sqlcode = MYSQL_ERRNO ;
SELECT @sqlstate, @sqlcode, @rowcount;
INSERT INTO T (id, s) VALUES (10, 'Smallint overflow for 32769?', 32769);
GET DIAGNOSTICS @rowcount = ROW_COUNT;
GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE,
@sqlcode = MYSQL_ERRNO ;
SELECT @sqlstate, @sqlcode, @rowcount;
INSERT INTO T (id,s,si) VALUES (11, 'Smallint overflow for 32769?', 32769);
GET DIAGNOSTICS @rowcount = ROW_COUNT;
GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE,
@sqlcode = MYSQL_ERRNO ;
SELECT @sqlstate, @sqlcode, @rowcount;
```

Ποιά είναι η απόκριση του διακομιστή στην κάθε μία περίπτωση; Διαφοροποιείται, για κάποιο λόγο, η απόκριση στην τρίτη εντολή SQL DML από τις άλλες δύο; Αναζητήστε στο διαδίκτυο την ερμηνεία των τιμών SQLCODE και SQLSTATE για την κάθε μία από τις τρεις περιπτώσεις.

9. Προσομοίωση βλάβης με Ctrl-C: η συναλλαγή ως μονάδα επαναφοράς από τη βλάβη. Εκτελέστε τις εντολές:
- ```
SELECT * FROM T;
SET AUTOCOMMIT = 0;
INSERT INTO T (id, s) VALUES (12, 'Let"s see what happens if ..');
SELECT * FROM T;
```

... αμέσως μετά διακόψτε τη συνεδρία βίαια, με την εντολή Ctrl-C (Control και μετά C).

Επανα-συνδεθείτε στη βάση MySQL και ελέγξτε το κατά πόσον η τελευταία συναλλαγή επιβίωσε ή όχι της βλάβης. Τι παρατηρείτε;

Παρακολουθήστε online τα τεκταινόμενα της παραπάνω (ερώτηση: soft ή hard;) βλάβης, ανοίγοντας ένα δεύτερο παράθυρο (συνεδρία) επικοινωνίας με το περιβάλλον MySQL και εκτελώντας τις εντολές με τη σειρά που αναγράφονται στον πίνακα που ακολουθεί.

| Συνεδρία A                                                                             | Συνεδρία B        |
|----------------------------------------------------------------------------------------|-------------------|
| SELECT * FROM T;                                                                       |                   |
|                                                                                        | SELECT * FROM T;  |
| SET AUTOCOMMIT=0;                                                                      |                   |
|                                                                                        | SET AUTOCOMMIT=0; |
| INSERT INTO T (id, s) VALUES (12, 'Let"s see what happens if ..');<br>SELECT * FROM T; |                   |
|                                                                                        | SELECT * FROM T;  |
| Cntrl-C                                                                                |                   |
|                                                                                        | SELECT * FROM T;  |

Σχολιάστε: (α) η συναλλαγή στη συνεδρία A, επιβίωσε ή όχι της βλάβης; (β) η βλάβη ήταν τοπικού (soft) ή καθολικού (hard) χαρακτήρα; (γ) ποιές από τις τέσσερις ιδιότητες των συναλλαγών (A.C.I.D.) βλέπετε να επαληθεύονται στα παραπάνω;

11. Ολοκληρώστε διαγράφοντας/καταργώντας τους πίνακες T, T2 και Accounts.

~~~~~