

Εργαστήριο Βάσεων Δεδομένων

Triggers

Triggers: Βασικές Έννοιες

- Ένας trigger είναι ένα κομμάτι κώδικα, μια ρουτίνα.
- Συνδέεται με ένα συγκεκριμένο πίνακα.
- Καλείται όταν συμβεί ένα γεγονός στον πίνακα.
- Συχνές χρήσεις:
 - Έλεγχος ορθότητας των δεδομένων που εισάγονται στον πίνακα.
 - Υλοποίηση λογικών περιορισμών.
 - Υπολογισμός παραγόμενων τιμών.
 - Συντήρηση logs πρόσβασης και αλλαγών στον πίνακα.
- Η υλοποίηση με triggers κοστίζει αρκετά, καθώς καθυστερούν την προσπέλαση του πίνακα.
- Στη MySQL
 - Υποστηρίζονται από την **5.0.2** έκδοση.
 - Πριν την **5.1.6** έκδοση έπρεπε να έχεις SUPER δικαιώματα για να δηλώσεις trigger.
 - Προσοχή! Η εξέταση πραγματοποιείται σε server όπου τρέχει η **5.0.5** έκδοση!

Βασικές εντολές

Δημιουργία

Εντολή CREATE TRIGGER <όνομα trigger>

Διαγραφή

Εντολή DROP TRIGGER <όνομα trigger>

Ανάκτηση Κώδικα

Εντολή SHOW CREATE TRIGGER <όνομα trigger>

Ανάκτηση Λίστας Triggers

Εντολή SHOW TRIGGERS

Κλήση

Δεν μπορείτε να καλέσετε έναν trigger. Καλείται μόνο όταν συμβεί το γεγονός με το οποίο συνδέεται.

Σχεδιασμός του trigger

- Για να δηλώσουμε έναν trigger πρέπει να αποφασίσουμε
 - Σε ποιόν πίνακα θα εφαρμοστεί:
 - Σε οποιοδήποτε πίνακα της βάσης στην οποία δημιουργείται.
 - Με τι είδος event θα συνδεθεί:
 - INSERT** Το event της εισαγωγής μιας εγγραφής στον πίνακα.
 - UPDATE** Το event της τροποποίησης μιας εγγραφής στον πίνακα.
 - DELETE** Το event της διαγραφής μιας εγγραφής από τον πίνακα.
 - Σε ποια χρονική στιγμή θα εκτελεστεί:
 - BEFORE** Πριν ξεκινήσει το event.
 - AFTER** Αφού ολοκληρωθεί το event.
 - Τη λειτουργία του:
 - Στο σώμα του trigger μπορούμε να γράψουμε κανονικά κώδικα.

Δημιουργία trigger

Δήλωση CREATE TRIGGER

```
CREATE TRIGGER trigger_name trigger_time trigger_event  
ON table_name  
FOR EACH ROW trigger_body
```

- Όπου:

trigger_name Το όνομα του trigger.

trigger_time Η χρονική στιγμή στην οποία θα κληθεί.

trigger_event Το event με το οποίο συνδέεται.

ON table_name Δήλωση του πίνακα στον οποίο ανήκει.

FOR EACH ROW Δήλωση ότι θα εκτελεστεί για κάθε γραμμή που περιλαμβάνεται στο event

trigger_body Το σώμα του trigger.

Δημιουργία ενός trigger - Παράδειγμα

- Δήλωση ενός trigger

Όνομα keep_count

Λειτουργία Κάθε φορά που εισάγεται ένα μάθημα, ενημερώνει μια μεταβλητή που κρατά το πλήθος των μαθημάτων.

```
mysql>CREATE TRIGGER keep_count  
->AFTER INSERT ON course  
->FOR EACH ROW  
->SET @courseCount=@courseCount+1;
```

- Γιατί AFTER?

Δημιουργία ενός trigger - Κλήση

- Κλήση trigger

```
mysql>SET @courseCount=(SELECT COUNT(*) FROM course);
mysql>SELECT @courseCount;
+-----+
| @courseCount |
+-----+
|             2 |
+-----+
1 row in set (0.00 sec)

mysql>INSERT INTO course(title,course_id,supervisor)
->VALUES
->('t1',NULL,'alex@upatras.gr'),
->('t2',NULL,'alex@upatras.gr');
```

- Επιστρέφει:

```
mysql>SELECT @courseCount;
+-----+
| @courseCount |
+-----+
|             4 |
+-----+
1 row in set (0.00 sec)
```

Trigger events

- Το event καταδεικνύει το είδος του γεγονότος που ενεργοποιεί τη σκανδάλη.
- Είδη events:
 - INSERT** Ενεργοποιείται κατά την εισαγωγή εγγραφών στον πίνακα.
 - Δηλαδή από τα statements: INSERT, LOAD DATA, REPLACE
 - UPDATE** Ενεργοποιείται κατά την τροποποίηση εγγραφών στον πίνακα.
 - Δηλαδή από τα statements UPDATE
 - DELETE** Ενεργοποιείται κατά τη διαγραφή εγγραφών στον πίνακα.
 - Δηλαδή από τα statements: DELETE και REPLACE
 - Η DROP TABLE και TRUNCATE δεν καλεί τον trigger
- Η διαγραφή ή τροποποίηση εγγραφών λόγω ξένων κλειδιών δεν ενεργοποιεί triggers!

Δυνατά triggers ενός πίνακα

- Κάθε trigger σε ένα συγκεκριμένο πίνακα πρέπει να έχει μοναδικό συνδυασμό event και χρονικής στιγμής.
- Άρα σε κάθε πίνακα μπορούμε να ορίσουμε το πολύ 6 triggers
 - BEFORE INSERT
 - AFTER INSERT
 - BEFORE UPDATE
 - AFTER UPDATE
 - BEFORE DELETE
 - AFTER DELETE

```
mysql> CREATE TRIGGER count_students
-> AFTER INSERT ON student
-> FOR EACH ROW
-> SET @stCount=@stCount+1;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TRIGGER count_students_alt
-> AFTER INSERT ON student
-> FOR EACH ROW
-> SET @studentsCount=@studentsCount+1;
```

```
ERROR 1235 (42000): This version of MySQL doesn't yet support 'multiple triggers with the same action time and event for one table'
```

Κώδικας μέσα στο σώμα του trigger

- Στο σώμα του trigger περιλαμβάνουμε τον κώδικα που εκτελείται όταν πυροδοτηθεί.
- Μπορούμε να συμπεριλάβουμε πολλαπλές ενέργειες χρησιμοποιώντας το block BEGIN...END.
 - Για τον ορισμό του σώματος του trigger επιβάλλεται η χρήση του delimiter.
- Όπως και στις stored procedures:
 - Μπορούμε να ορίσουμε μεταβλητές
 - Μπορούμε να χρησιμοποιήσουμε εντολές ελέγχου ροής
 - Μπορούμε να ανακτήσουμε τα δεδομένα της βάσης με SELECT INTO και CURSORS
 - Μπορούμε να καλέσουμε stored procedures με την CALL
- Δεν μπορούμε:
 - Να εκτυπώσουμε αποτελέσματα μιας SELECT.
 - Να επεξεργαστούμε τον πίνακα στον οποίο εφαρμόζεται ο trigger.
- Οι περιορισμοί που ισχύουν για τους triggers ισχύουν και για τις stored procedures που καλούνται από triggers.

Κώδικας στους triggers - Παράδειγμα

- Δήλωση ενός trigger

Όνομα init_book

Λειτουργία Κάθε φορά που εισάγεται ένα μάθημα, εισάγει και μια εγγραφή στα βιβλία για αυτό το μάθημα, χρησιμοποιώντας τον default τίτλο.

```
mysql>DELIMITER $
mysql>CREATE TRIGGER init_book
->AFTER INSERT ON course
->FOR EACH ROW
->BEGIN
-> DECLARE cid INT(4)
-> SELECT MAX(course_id) INTO cid FROM course;
-> INSERT INTO books(title,course_book)
-> VALUES (DEFAULT,cid);
->END$
mysql>DELIMITER ;
```

Κώδικας στους triggers - Κλήση

- Κλήση trigger

```
mysql>INSERT INTO course(title,course_id,supervisor)
->VALUES
->>('t3',NULL,'alex@upatras.gr'),
->>('t4',NULL,'alex@upatras.gr');
```

```
mysql>
```

```
mysql>SELECT books.title as Book, course.title as Course
->FROM books
->INNER JOIN course ON course_book=course_id;
```

```
+-----+-----+
| Book          | Course          |
+-----+-----+
| Databases 1   | Βάσεις Δεδομένων |
| Databases 1 2nd volume | Βάσεις Δεδομένων |
| Databases 2   | Βάσεις Δεδομένων II |
| Title        | t3              |
| Title        | t4              |
+-----+-----+
5 rows in set (0.00 sec)
```

Κώδικας στους triggers - Περιορισμοί

- Από τον trigger δεν επιτρέπεται να εκτυπωθούν αποτελέσματα
SELECT

```
mysql> DELIMITER $
mysql> CREATE TRIGGER init_book
-> AFTER INSERT ON course
-> FOR EACH ROW
-> BEGIN
-> DECLARE cid INT(4);
->
-> SELECT MAX(course_id) INTO cid FROM course;
-> INSERT INTO books(title, course_book)
-> VALUES (DEFAULT,cid);
->
-> SELECT title FROM books WHERE course_book=cid;
-> END$
ERROR 1415 (0A000): Not allowed to return a result set from a trigger
mysql> DELIMITER ;
```

Προσπέλαση του πίνακα στον οποίο ανήκει το trigger - Περιορισμοί

- Ο trigger δεν μπορεί να αλλάξει τον πίνακα στον οποίο εφαρμόζεται.
- Δεν είναι δυνατόν ακόμα και αν καλέσει stored procedure που τον αλλάζει.
- Παράδειγμα:

Όνομα overflow_registrations
Λειτουργία Αν τα registrations έχουν φτάσει σε ένα μέγιστο πλήθος, μόλις επειληφθεί μια νέα registration, σβώνονται τα παλιότερα.

```
mysql>DELIMITER $
mysql>CREATE TRIGGER overflow_registrations
->BEFORE INSERT ON registration
->FOR EACH ROW
->BEGIN
-> DECLARE regNum INT
-> DECLARE oldDate DATE
-> SELECT COUNT(*) INTO regNum FROM registration;
-> IF regNum>=5 THEN
-> SELECT MIN(reg_date) INTO oldDate
-> FROM registration;
-> DELETE FROM registration
-> WHERE reg_date=oldDate;
-> END IF;
->END$
mysql>DELIMITER ;
```

- Κλήση:

```
mysql> INSERT INTO registration(reg_date, reg_student, reg_course)
-> VALUES ('2012-12-03',2193,2);
ERROR 1442 (HY000): Can't update table 'registration' in stored function/trigger
because it is already used by statement which invoked this stored function/trigger.
```

Προσπέλαση των τιμών της τρέχουσας εγγραφής

- Κάθε κλήση της trigger αφορά μια εγγραφή (row) του πίνακα, η οποία δημιουργείται, ενημερώνεται ή διαγράφεται.
- Η εγγραφή έχει δύο καταστάσεις σε σχέση με το trigger:
 - Η κατάσταση της πριν την πραγματοποίηση του event.
 - Η κατάσταση της αφού ολοκληρωθεί το event.
- Χρησιμοποιούμε τα ψευδώνυμα OLD και NEW για να αναφερθούμε στην αρχική και την τελική κατάσταση αντίστοιχα.
- Μέσω των αναφορών OLD και NEW μπορούμε να προσπελάσουμε τα στοιχεία της τρέχουσας εγγραφής (row):

OLD.column_name Αναφέρεται στην τιμή της στήλης
column_name που είχε η εγγραφή πριν το event.

NEW.column_name Αναφέρεται στην τιμή της στήλης
column_name που απέκτησε η εγγραφή μετά το
event.

OLD και NEW αναφορές

- Διαθεσιμότητα OLD και NEW αναφορών ανά είδος event:
 - Σε έναν trigger για INSERT
 - Δεν υπάρχει OLD εγγραφή, το OLD δεν χρησιμοποιείται.
 - Το NEW αναφέρεται είτε στην προσωρινή εγγραφή που θα εισαχθεί, είτε στην εγγραφή που μόλις έγινε.
 - Σε έναν trigger για UPDATE
 - Το OLD αναφέρεται στις τιμές της εγγραφής πριν γίνει τροποποίηση.
 - Το NEW αναφέρεται είτε στις τιμές της εγγραφής που θα αποθηκευτούν, είτε στις τιμές που έχουν αποθηκευτεί ήδη.
 - Σε έναν trigger για DELETE
 - Το OLD αναφέρεται στην εγγραφή που θα διαγραφεί.
 - Δεν υπάρχει NEW εγγραφή, το NEW δεν χρησιμοποιείται.
- Η αναφορά OLD επιτρέπει μόνο ανάγνωση των παλαιότερων τιμών.
- Η αναφορά NEW επιτρέπει και την αλλαγή των τιμών που είναι προς αποθήκευση.
 - Επιτρέπει την αλλαγή μόνο **πριν** την αποθήκευση, δηλαδή σε BEFORE χρόνο.
 - Στα AUTO_INCREMENT πεδία η τιμή σε BEFORE χρόνο είναι μηδέν.

OLD και NEW αναφορές - Παράδειγμα INSERT BEFORE

- Παράδειγμα:

- Όνομα** checkRegDate
Λειτουργία Αν η ημερομηνία εγγραφής που θα εισαχθεί είναι μελλοντική, τότε την αντικαθιστά τη με τη σημερινή.

```
mysql>DELIMITER $
mysql>CREATE TRIGGER checkRegDate
->BEFORE INSERT ON registration
->FOR EACH ROW
->BEGIN
->  DECLARE currDate DATE
->
->  SET currDate=CURDATE();
->  IF NEW.reg_date>currDate THEN
->    SET NEW.reg_date=currDate;
->  END IF;
->END$
mysql>DELIMITER ;
```

OLD και NEW αναφορές - Παράδειγμα INSERT BEFORE

- Κλήση:

```
mysql>INSERT INTO registration(reg_date,reg_student,reg_course)
->VALUES('2030-04-17',2193,2);
```

```
mysql>
```

```
mysql>SELECT * FROM registration
->WHERE reg_course=2 AND reg_student=2193;
```

```
+-----+-----+-----+
| reg_date | reg_student | reg_course |
+-----+-----+-----+
| 2012-12-03 |          2193 |           2 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

OLD και NEW αναφορές - Παράδειγμα INSERT AFTER

- Παράδειγμα:

Όνομα returnId
Λειτουργία Μετά την εισαγωγή ενός φοιτητή, επιστρέφει σε μια μεταβλητή στο περιβάλλον το auto_increment id που πήρε η νέα εγγραφή.

```
mysql>DELIMITER $  
mysql>CREATE TRIGGER returnId  
->AFTER INSERT ON student  
->FOR EACH ROW  
->BEGIN  
-> SET @newStudent=NEW.AM;  
->END$  
mysql>DELIMITER ;
```

- Γιατί AFTER?

OLD και NEW αναφορές - Παράδειγμα INSERT AFTER

- Κλήση:

```
mysql> INSERT INTO student(AM,name,lastname)
-> VALUES(NULL,'Γιώργος','Αναστασίου');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM student WHERE lastname='Αναστασίου';
+-----+-----+-----+
| name   | lastname | am   |
+-----+-----+-----+
| Γιώργος | Αναστασίου | 2194 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT @newStudent;
+-----+
| @newStudent |
+-----+
|          2194 |
+-----+
1 row in set (0.00 sec)
```

OLD και NEW αναφορές - Παράδειγμα UPDATE

- Παράδειγμα:

- Όνομα** validateEmail
Λειτουργία Αν επιχειρηθεί η τροποποίηση του email ενός καθηγητή, ελέγχει εάν είναι νόμιμη η μορφή της νέας διεύθυνσης και αν όχι διατηρεί την παλιά.

```
mysql>DELIMITER $
mysql>CREATE TRIGGER validateEmail
->BEFORE UPDATE ON professor
->FOR EACH ROW
->BEGIN
-> IF NEW.email NOT LIKE '%@%.%' THEN
->     SET NEW.email=OLD.email;
-> END IF;
->END$
mysql>DELIMITER ;
```

OLD και NEW αναφορές - Παράδειγμα UPDATE - Κλήση

- Κλήση:

```
mysql> SELECT * FROM professor;
```

```
+-----+-----+-----+
| pr_name | pr_lastname | email |
+-----+-----+-----+
| Αλεξάνδρα | unknown | alex@upatras.gr |
| unknown | Δημητρίου | dim@upatras.gr |
| Μαρία | Παπαδοπούλου | pap@ceid.upatras.gr |
+-----+-----+-----+
3 rows in set (0.07 sec)
```

```
mysql> UPDATE professor SET email = 'alex@ceid.upatras.gr'
-> WHERE email='alex@upatras.gr';
```

```
Query OK, 1 row affected (0.07 sec)
Rows matched: 1 Changed: 1 warnings: 0
```

```
mysql> UPDATE professor SET email = 'dim.ceid.upatras.gr'
-> WHERE email='dim@upatras.gr';
```

```
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1 Changed: 0 warnings: 0
```

```
mysql> SELECT * FROM professor;
```

```
+-----+-----+-----+
| pr_name | pr_lastname | email |
+-----+-----+-----+
| Αλεξάνδρα | unknown | alex@ceid.upatras.gr |
| unknown | Δημητρίου | dim@upatras.gr |
| Μαρία | Παπαδοπούλου | pap@ceid.upatras.gr |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Use Case: Ανύρωση του event από τον trigger

- Μια συχνή χρήση των triggers είναι ο έλεγχος των τιμών της εγγραφής προς επεξεργασία.
- Στην περίπτωση που ο έλεγχος αποτυγχάνει σε χρόνο BEFORE, οι συνήθεις επιλογές είναι δύο:
 - Διόρθωση της τιμής που δεν ικανοποιεί τους περιορισμούς σε επιτρεπτή τιμή.
 - Ανύρωση της λειτουργίας που πυροδότησε το trigger.
- Για να ακυρωθεί το event που πυροδότησε το trigger υπάρχουν δύο επιλογές:
 - Η πραγματοποίηση μέσα στον κώδικα κάποιας μη επιτρεπτής ενέργειας έτσι ώστε να προκληθεί σφάλμα και να ακυρωθεί η ενέργεια.
 - Το πιο συχνό παράδειγμα είναι η εισαγωγή NULL σε κάποιο πεδίο που απαγορεύεται.
 - Η χρήση της εντολής SIGNAL.
 - Είναι η εντολή που προκαλεί την δημιουργία ενός error, το οποίο διακόπτει την εκτέλεση.
 - Υποστηρίζεται μόνο στην 5.5 έκδοση της MySQL!
 - Προσοχή! Η εξέταση πραγματοποιείται σε server όπου τρέχει η 5.0.5 έκδοση!

Ακύρωση του event - Invalid Operation - Παράδειγμα

- Παράδειγμα:

- Όνομα** checkProfessor
Λειτουργία Πριν την εισαγωγή του μαθήματος ελέγχει πόσα μαθήματα έχει ήδη αναλάβει ο supervisor. Αν έχει αναλάβει τον μέγιστο αριθμό, τότε αποτρέπει την εισαγωγή του μαθήματος.

```
mysql>DELIMITER $
mysql>CREATE TRIGGER validateEmail
->BEFORE INSERT ON course
->FOR EACH ROW
->BEGIN
-> DECLARE numOfCourses INT;
-> SELECT COUNT(*) INTO numOfCourses
-> FROM course
-> WHERE course.supervisor=NEW.supervisor;
->
-> IF numOfCourses>=2 THEN
-> SET NEW.supervisor=NULL;
-> END IF;
->END$
mysql>DELIMITER ;
```


Ακύρωση του event - Invalid Operation - Κλήση

- Κλήση:

```
mysql> SELECT COUNT(*) FROM course WHERE supervisor='pap@ceid.upatras.gr';
+-----+
| COUNT(*) |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO course(title,supervisor)
-> VALUES ('Lesson2','pap@ceid.upatras.gr');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO course(title,supervisor)
-> VALUES ('Lesson3','pap@ceid.upatras.gr');
ERROR 1048 (23000): column 'supervisor' cannot be null
```

Ακύρωση του event - SIGNAL - Παράδειγμα

- Παράδειγμα:

- Όνομα** validateRegistration
Λειτουργία Πριν την εγγραφή ενός φοιτητή σε μάθημα, έλεγχος στην ημερομηνία εγγραφής. Αν είναι πάνω από ένα χρόνο σε απόσταση από σήμερα, να αποτρέπεται η εγγραφή.

```
mysql>DELIMITER $
mysql>CREATE TRIGGER validateRegistration
->BEFORE INSERT ON registration
->FOR EACH ROW
->BEGIN
-> DECLARE currDate DATE;
-> DECLARE diff INT;
->
-> SET currDate=CURDATE();
-> SET diff=ABS(DATEDIFF(currDate,NEW.reg_date));
->
-> IF diff>=365 THEN
-> SIGNAL SQLSTATE VALUE '45000'
-> SET MESSAGE_TEXT = 'Invalid registration date! Must be within a
year.';
-> END IF;
->ENDS
mysql>DELIMITER ;
```

Ακύρωση του event - SIGNAL - Παράδειγμα

- Στην SIGNAL ορίσαμε:

SQLSTATE Ο κωδικός του σήματος.

- Το manual της MySQL ορίζει συγκεκριμένες τιμές και κανόνες για τον ορισμό τους.
- τιμή που θα χρειαστείτε είναι '45000', που σημαίνει 'User defined error'.

MESSAGE_TEXT Το μήνυμα του σήματος.

- Συναρτήσεις που χρησιμοποιήθηκαν:

ABS Επιστρέφει το απόλυτο.

DATEDIFF Αφαιρεί δύο ημερομηνίες και επιστρέφει τη διαφορά τους σε μέρες.

- Κλήση:

```
mysql> INSERT INTO registration(reg_date,reg_student,reg_course)
-> VALUES ('2030-04-17',2194,2);
ERROR 1644 (45000): Invalid registration date! Must be within a year.
mysql>
mysql> INSERT INTO registration(reg_date,reg_student,reg_course)
-> VALUES ('2013-04-17',2194,2);
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO registration(reg_date,reg_student,reg_course)
-> VALUES ('2012-04-17',2193,2);
Query OK, 1 row affected (0.12 sec)
```

Use Case: Υλοποίηση του AUTO_INCREMENT

- Σε κάποια RDBMS (όπως η ORACLE) δεν υποστηρίζεται η auto_increment δήλωση.
- Στα μερικά κλειδιά συμβαίνει να θέλουμε λειτουργικότητα αύξοντος αριθμού, αλλά όχι στο σύνολο του πίνακα.
- Σε αυτές τις περιπτώσεις η λειτουργικότητα auto increment υλοποιείται με trigger.
- Παράδειγμα:

Όνομα autoIncrementLecture
Λειτουργία Υπολογίζει αυτόματα τον επόμενο αύξοντα αριθμό διάλεξης για ένα συγκεκριμένο μάθημα.

```
mysql>DELIMITER $
mysql>CREATE TRIGGER autoIncrementLecture
->BEFORE INSERT ON lecture
->FOR EACH ROW
->BEGIN
-> DECLARE maxNum INT(2);
->
-> SET maxNum=0;
->
-> SELECT MAX(num_lecture) INTO maxNum
-> FROM lecture
-> WHERE course_lecture=NEW.course_lecture;
->
-> SET NEW.num_lecture=maxNum+1;
->ENDS
mysql>DELIMITER ;
```

Υλοποίηση του AUTO_INCREMENT

- Κλήση:

```
mysql> SELECT * FROM lecture WHERE course_lecture=2;
+-----+-----+-----+
| subject          | num_lecture | course_lecture |
+-----+-----+-----+
| Εισαγωγή σε βάσεις |          1 |                2 |
| Ανάλυση απαιτήσεων |          2 |                2 |
| ER-Σχεσιακό       |          3 |                2 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> INSERT INTO lecture(subject, num_lecture, course_lecture)
-> VALUES ('Εισαγωγή στη MySQL 1',0,2);
query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO lecture(subject, num_lecture, course_lecture)
-> VALUES ('Εισαγωγή στη MySQL 2',0,2);
query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM lecture WHERE course_lecture=2;
+-----+-----+-----+
| subject          | num_lecture | course_lecture |
+-----+-----+-----+
| Εισαγωγή σε βάσεις |          1 |                2 |
| Ανάλυση απαιτήσεων |          2 |                2 |
| ER-Σχεσιακό       |          3 |                2 |
| Εισαγωγή στη MySQL 1 |          4 |                2 |
| Εισαγωγή στη MySQL 2 |          5 |                2 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

- Σημείωση:

- Αν η SELECT δεν επιστρέφει τίποτα, τότε η maxNum θα διατηρήσει την προηγούμενη τιμή της, δηλαδή το 0.
- Η πρώτη διάλεξη ενός μαθήματος θα πάρει δηλαδή την τιμή 1.

Σημειώσεις για την εξέταση

- Η εξέταση πραγματοποιείται σε server όπου τρέχει η **5.0.5** έκδοση της MySQL, επομένως στην εξέταση:
 - Δεν θα έχετε δικαιώματα για να δημιουργήσετε trigger.
 - Θα φτιάξετε κανονικά το CREATE TRIGGER statement.
 - Θα το τρέξετε στην κονσόλα.
 - Αν έχει συντακτικά λάθη, η MySQL θα βγάλει το κατάλληλο μήνυμα.
 - Αν είναι σωστό συντακτικά, η MySQL θα βγάλει μήνυμα ότι δεν έχετε δικαιώματα για να δημιουργήσετε trigger.
 - Τη λογική θα την ελέγξετε μόνοι σας και εμείς θα το διορθώσουμε με τον ίδιο τρόπο.
 - Δεν θα δουλεύει η SIGNAL.
 - Αν σας ζητηθεί να ακυρώσετε ένα event, αναγκαστικά θα χρησιμοποιήσετε τον εναλλακτικό τρόπο, την πρόκληση σφάλματος.