

Εργαστήριο 4

Έναυσμα (trigger)

Ανάπτυξη και Επιμέλεια: Ζ. Γαροφαλάκη, Α. Τσολακίδης, Π. Ανδρίτσος

Στόχος

Εξοικείωση με την χρήση των έναυσμάτων (Triggers).

Εργαλεία

A. Trigger

Ένας trigger είναι μια ρουτίνα που σχετίζεται με ένα πίνακα. Θα ενεργοποιηθεί όταν πραγματοποιηθεί ένα γεγονός στον συγκεκριμένο πίνακα όπως INSERT, UPDATE και DELETE και μπορεί να κληθεί πριν ή μετά το συμβάν.

Συχνοί Λόγοι Χρήσης

- Υπολογισμός παραγόμενων τιμών.
- Καταχώρηση τιμών σε άλλους σχετιζόμενους πίνακες
- Έλεγχος περιορισμών πριν την μεταβολή των εγγραφών.
- Καταγραφή αυτοματοποιημένα των χρηστών και των ημερομηνιών που πραγματοποιούνται αλλαγές σε ένα πίνακα.

1. Η δημιουργία trigger γίνεται με την ακόλουθη γενική μορφή εντολής.

```
CREATE TRIGGER trigger_name  
{BEFORE | AFTER} {INSERT | UPDATE | DELETE }  
ON table_name FOR EACH ROW  
trigger_body;
```

2. Κάθε κλήση του trigger αφορά μια εγγραφή (row) του πίνακα, η οποία δημιουργείται, ενημερώνεται ή διαγράφεται. Κάθε εγγραφή μπορεί να έχει δύο καταστάσεις σε σχέση με την στιγμή που εκτελείτε ο trigger. Χρησιμοποιούμε τους τις Global μεταβλητές OLD και NEW για να αναφερθούμε στην αρχική και την τελική τιμή της εγγραφής αντίστοιχα.

- a. OLD.column_name. Αναφερόμαστε στην τιμή της στήλης που είχε η εγγραφή πριν το event.
- b. NEW.column_name Αναφερόμαστε στην τιμή της στήλης που θα έχει η εγγραφή μετά το event.

3. Πότε χρησιμοποιούμε OLD και NEW:

- a. Σε έναν trigger για INSERT
 - i. Δεν υπάρχει OLD εγγραφή.

- ii. Το NEW δείχνει στην εγγραφή που θα εισαχθεί, είτε στην εγγραφή που μόλις έγινε.
 - b. Σε έναν trigger για UPDATE
 - i. Το OLD αναφέρεται στις τιμές της εγγραφής πριν γίνει το update.
 - ii. Το NEW αναφέρεται είτε στις τιμές της εγγραφής που θα ισχύουν μετά το update.
 - c. Σε έναν trigger για DELETE
 - i. Το OLD αναφέρεται στην εγγραφή που θα διαγραφεί.
 - ii. Δεν υπάρχει NEW εγγραφή.
- 4. Ακολουθεί παράδειγμα εναύσματος (trigger) που αναλαμβάνει την εισαγωγή των ονομάτων των τμημάτων με κεφαλαία γράμματα στον πίνακα dept της βάσης δεδομένων. Το έναυσμα trigger dept_insert_update είναι αποθηκευμένο (stored) και εκτελείται πριν από την εισαγωγή των στοιχείων του πίνακα department και μεταγράφει τα στοιχεία αυτά με κεφαλαία γράμματα.

delimiter //	Όρισα ως delimiter τους χαρακτήρες //
CREATE TRIGGER dept_insert BEFORE INSERT ON department FOR EACH ROW BEGIN SET NEW.dname = UPPER(NEW.dname); END;	Ορίζω το αντικείμενο dept_insert
//	Ο ορισμός ολοκληρώθηκε και ο server δημιουργεί τον trigger
delimiter ;	Αλλάζω delimiter ώστε ο sever να περιμένει τον χαρακτήρα ; για να καταλάβει ότι ολοκληρώθηκε κάθε εντολή που θα του δώσω στη συνέχεια.

Επεξήγηση

- I. Δίνεται η δήλωση **INSERT INTO department VALUES(70, 'Learn');**
- II. Λόγω του γεγονότος “**BEFORE INSERT ON department**” που υπάρχει στον ορισμό του «αφυπνίζεται» ο trigger **dept_insert**.
- III. Επειδή δόθηκε δήλωση INSERT και ο trigger είναι row τα ζεύγη των global μεταβλητών που αντιστοιχούν στις στήλες του πίνακα Department έχουν τις παρακάτω τιμές:
 - a. OLD.deptno < - - NULL, OLD.dname < - - NULL
 - b. NEW.deptno < - - 70, NEW.dname < - - 'Learn'
- IV. Λόγω της εντολής SET NEW.dname = UPPER(NEW.dname); ο trigger μεταγράφει σε κεφαλαία την τιμή της μεταβλητής NEW.dname.
- V. Ο trigger τερματίζεται και εκτελείται η δήλωση:
 - a. **INSERT INTO department VALUES(70, 'LEARN');**

5. Η διαγραφή του **trigger** με όνομα dept_insert γίνεται με την εντολή:

```
DROP TRIGGER dept_insert;
```

Δραστηριότητες

Υλοποιήστε τις ακόλουθες δραστηριότητες. Η εντολή ή οι εντολές που απαιτούνται για την υλοποίηση του κάθε βήματος, καθώς και το αποτέλεσμα της εκτέλεσής της/τους θα πρέπει να ενταχθεί/-ούν σε ένα παραδοτέο αρχείο με τη μορφή κειμένου ή με τη μορφή στιγμιότυπου (screenshot). Το αρχείο ή τα αρχεία με τις απαντήσεις σας, θα πρέπει να συμπιεστούν σε ένα **xx_ZZZZZ_ΕΡΩΝΥΜΟ.zip**, όπου: (α) xx ο αριθμός του τμήματος στο οποίο ανήκετε (π.χ. για την ομάδα [02] ΔΕΥΤΕΡΑ 12:00-13:00, **xx = 02**) και (β) ZZZZZ ο Αριθμός Μητρώου σας. Το τελικό αυτό αρχείο θα υποβάλλεται στο χώρο του e-class -> ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ II -> Εργασίες.

1. **Συνδεθείτε** στην MySQL του συστήματός σας με όποιον από τους προαναφερόμενους τρόπους επιθυμείτε.
2. Ελέγξτε αν υπάρχει ΒΔ με την ονομασία **personnel**.
3. Επιλέξτε την ΒΔ personnel για **χρήση**.
4. Δημιουργήστε τον trigger με όνομα dept_update που αφυπνίζεται από το γεγονός "BEFORE UPDATE ON".

```
delimiter //
```

```
CREATE TRIGGER dept_update
```

```
BEFORE UPDATE ON department
```

```
FOR EACH ROW
```

```
BEGIN
```

```
SET NEW.dname = UPPER(NEW.dname);
```

```
END;
```

```
//
```

```
delimiter ;
```

Δοκιμή:

```
/* testing */
```

```
UPDATE department SET dname = 'Operations' WHERE deptno=70;
```

5. Στον πίνακα Department θα προσθέσουμε τη στήλη no_of_employees (αριθμός υπαλλήλων που έχει το τμήμα).

```
ALTER TABLE department ADD (no_of_employees INT);
```

6. Στον πίνακα Department θα ενημερώσουμε τη στήλη no_of_employees (αριθμός υπαλλήλων που έχει το τμήμα).

```
UPDATE department  
  
SET no_of_employees =  
  
        (SELECT COUNT(*)  
  
        FROM employee  
  
        WHERE employee.deptno = department.deptno);
```

7. Στον πίνακα Trigger που αφυπνίζεται από το γεγονός AFTER INSERT ON

```
delimiter //  
  
CREATE TRIGGER emp_insert  
  
AFTER INSERT ON employee  
  
FOR EACH ROW  
  
BEGIN  
  
UPDATE department  
  
SET no_of_employees = IFNULL(no_of_employees, 0) + 1  
  
WHERE deptno= NEW.deptno;  
  
END;  
  
//  
  
delimiter ;
```

8. Δοκιμή

```
INSERT INTO employee VALUES(7985, 'CLARKE', 10);  
  
SELECT * FROM employee;  
  
SELECT * FROM department;
```

9. Επεξήγηση

1. Ο trigger «αφυπνίζεται» λόγω της συνθήκης AFTER INSERT ON employee
2. OLD.empno<--NULL, OLD.ename<--NULL, OLD.deptno<--NULL
3. NEW.empno<-7985, NEW.ename<-'NAVATHE', NEW.deptno<-10

Άρα η δήλωση

```
UPDATE department
SET no_of_employees = NVL(no_of_employees, 0) + 1
WHERE deptno= :NEW.deptno;
```

Είναι ισοδύναμη με τη δήλωση

```
UPDATE department
SET no_of_employees = NVL(no_of_employees, 0) + 1
WHERE deptno= 10;
```

10. Δημιουργήστε, δοκιμάστε και εξηγήστε τον trigger που αφυπνίζεται από το γεγονός AFTER DELETE ON
11. Δημιουργήστε, δοκιμάστε και εξηγήστε τον trigger που αφυπνίζεται από το γεγονός AFTER UPDATE ON

12. Παρουσίαση πληροφοριών για τους Triggers

```
/* see triggers */
DESCRIBE Information_schema.TRIGGERS;

SELECT TRIGGER_NAME, EVENT_MANIPULATION, TRIGGER_SCHEMA
FROM INFORMATION_SCHEMA.TRIGGERS
WHERE TRIGGER_SCHEMA = 'my_first_triggers_db'
ORDER BY TRIGGER_NAME;
```

13. Διαγραφή των Triggers

```
DROP TRIGGER dept_insert;

DROP TRIGGER dept_update;

.....
```

ΒΔ personnel

Οι περιεχόμενοι πίνακες της ΒΔ personnel θα πρέπει να έχουν την ακόλουθη δομή και περιεχόμενα:

Στήλες	Τύπος δεδομένων
DEPT.DEPTNO, EMP.DEPTNO	numeric(2)
DNAME, JOB_DESCR	varchar(24)
LOC	char(23)
JOBCODE, JOBNO	numeric(3)
SAL, COMM	numeric(10,2)
EMPNO	numeric(4)
PROJECT.P_ID	int
PROJECT.P_NAME	varchar(255)

Πίνακας 1. Τύποι δεδομένων πινάκων EMP, JOB, DEPT

EMP

EMPNO	NAME	JOBNO	DEPTNO	COMM
10	CODD	100	50	
20	NAVATHE	200	50	450
30	ELMASRI	300	60	
40	DATE	100	50	

JOB

JOBCODE	JOB_DESCR	SAL
100	SALESMAN	2000
200	ANALYST	2000
300	DBA	3000

DEPT

DEPTNO	DNAME	LOC
50	SALES	ATHENS
60	ACCOUNTING	ATHENS
70	PAYROL	VOLOS

Εικόνα 1. Δεδομένα πινάκων EMP, JOB, DEPT