





ПЕРІЕХОМЕНА:

- 1. Δρομείς (Cursors)
- 2. Σκανδάλες (Triggers)
- 3. Γεγονότα (Events)

Ευάγγελος Μ.

Σμαραγδένιος Χορηγός Μαθήματος

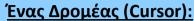
Κωνσταντίνος Μπλέτσης

Χρυσός Χορηγός Μαθήματος

MAOHMA 2.7: Cursors, Triggers και Events

1. Δρομείς (Cursors)





- Δίνει τη δυνατότητα επανάληψης επί ενός result set εντός μίας αποθηκευμένης διαδικασίας.
- Δήλωση δρομέα (αμέσως μετά τη δήλωση μεταβλητών)

DECLARE cursor **CURSOR FOR** select

- όπου cursor το όνομα που δίνουμε στο δρομέα και select το query με το οποίο σχετίζεται.
- Αρχικοποίηση του cursor με την εντολή:

OPEN cursor

• Παίρνουμε τα στοιχεία που έχουμε επιλέξει (στο select) με την εντολή:

FETCH cursor **INTO** variables

- η οποία μεταφέρει το δρομέα στην επόμενη γραμμη
- Ενεργοποιείται handler συνθήκης NOTFOUND, όταν φτάσει στο τέλος των εγγραφών του result set, αρκεί να τον έχουμε δηλώσει κατάλληλα:

DECLARE CONTINUE **HANDLER FOR** NOTFOUND statement;

• Κλείνουμε τον cursor με την εντολή:

CLOSE cursor

Παράδειγμα 1: DB: world, script: cursor.sql CREATE PROCEDURE total_population() BEGIN

```
DECLARE cur_pop INT;
DECLARE total_pop BIGINT DEFAULT 0;
DECLARE finished INT DEFAULT 0;
DECLARE cursor_population CURSOR FOR
SELECT population FROM country
WHERE population IS NOT NULL;
```

DECLARE CONTINUE HANDLER FOR NOT FOUND

```
SET finished=1;
OPEN cursor_population;
REPEAT
    FETCH cursor_population INTO cur_pop;
    SET total_pop = total_pop + cur_pop;
UNTIL finished=1
END REPEAT;
CLOSE cursor_population;
SELECT total_pop;
```

Περιορισμοί:

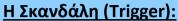
END

• Οι cursors είναι μόνο για ανάγνωση, οι εγγραφές έρχονται με τη σειρά του SELECT (όχι τυχαία προσπέλαση) και τα δεδομένα που φέρνουν είναι τα πραγματικά (άλλα ΣΔΒΔ δουλεύουν με αντίγραφα)

MAOHMA 2.7: Cursors, Triggers και Events

2. Σκανδάλες (Triggers)





- Εκτελεί αυτόματα ένα μπλοκ κώδικα πριν ή μετά από ένα INSERT, UPDATE ή DELETE μίας γραμμής του πίνακα.
- Συντακτικό:

```
CREATE TRIGGER trigger
(BEFORE | AFTER) (INSERT | UPDATE | DELETE) ON table
FOR EACH ROW
BEGIN
...
END;
```

Παρατηρήσεις:

- Οι σκανδάλες είναι αποθηκευμένα προγράμματα ειδικού σκοπού.
- Ο ορισμός τους απαιτεί αλλαγή του delimiter αντίστοιχα με τις stored procedures/functions.

```
DELIMITER $$
CREATE TRIGGER ...
END$$
DELIMITER ;
```

• ενώ βλέπουμε τα triggers επί ενός πίνακα και μέσω του MySQL Workbench (βλ.βίντεο)

```
city
Columns
Indexes
Foreign Keys
Triggers
```

Παράδειγμα 2: DB: emp, script: trigger.sql

```
CREATE TRIGGER tr_employees_after_insert

AFTER INSERT ON employees

FOR EACH ROW

BEGIN

DECLARE changes VARCHAR(200);

SET changes = CONCAT(NEW.emp_no,',',

NEW.birth_date,',',

NEW.first_name,',',

NEW.last_name,',',

NEW.gender,',',

NEW.hire_date);

INSERT INTO xafies

VALUES(CURRENT_USER(), 'INSERT', changes,

CURRENT_TIMESTAMP());

END
```

 Έχουμε πρόσβαση στις παλιές τιμές της γραμμής (πριν από UPDATE ή DELETE ως:

OLD.column

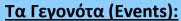
 Έχουμε πρόσβαση στις νέες τιμές της γραμμής (μετά από UPDATE ή INSERT ως:

NEW.column

MAOHMA 2.7: Cursors, Triggers και Events

3. Γεγονότα (Events)





- είναι μπλοκ κώδικα τα οποία τρέχουν σε προκαθορισμένες (επαναλαμβανόμενες) χρονικές στιγμές.
- Συντακτικό event που θα τρέξει μία φορά:

```
CREATE EVENT event
ON SCHEDULE AT timestamp
DO
BEGIN
...
END;
```

• Συντακτικό event που θα τρέξει πολλές φορές:

```
CREATE EVENT event
ON SCHEDULE EVERY n interval [STARTS timestamp]
[ENDS timestamp]
DO
BEGIN
...
END;
```

- interval είναι ένα από τα SECOND, MINUTE, HOUR, DAY,
 MONTH, YEAR
- timestamp είναι η χρονική στιγμή (αλλά μπορεί να καθοριστεί και ημερομηνία)

Παρατήρηση:

- O event scheduler της MySQL είναι αρχικά απενεργοποιημένος.
- Τον ενεργοποιούμε με την εντολή:

SET GLOBAL event scheduler=ON;

βλέπουμε τα events που έχουν οριστεί με την εντολή:

SHOW EVENTS;

ενώ διαγράφουμε ένα event με την εντολή:

DROP EVENT event;

Παράδειγμα 3: DB: emp, script: event.sql

CREATE EVENT event_erase_xafies
ON SCHEDULE EVERY 1 MINUTE STARTS now()
DO BEGIN
DELETE FROM xafies;
END

Παρατήρηση:

 Τα event αφορούν μόνο επεξεργασία των δεδομένων της βάσης με αντίστοιχες δυνατότητες με τις stored procedures. Για άλλες συστηματικές διαδικασίες (π.χ. backup) υπάρχουν άλλου είδους διαχειρίσεις (βλ. επόμενο μάθημα)