

**Ανοικτά Ακαδημαϊκά Μαθήματα**

**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας**

**Βάσεις Δεδομένων ΙΙ (Ε)**

**Ενότητα 5:** Χρήση stored procedures: cursors, functions, procedures, triggers

Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης

Τμήμα Μηχανικών Πληροφορικής ΤΕ

|  |  |
| --- | --- |
| Το περιεχόμενο του μαθήματος διατίθεται με άδεια Creative Commons εκτός και αν αναφέρεται διαφορετικά | Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους. |

Περιεχόμενα

[1. Δημιουργία βάσης δεδομένων training 3](#_Toc401504264)

[2. Παράδειγμα χρήσης Cursor 3](#_Toc401504265)

[3. Ορισμός συνάρτησης για τη διαχείριση Cursor 5](#_Toc401504266)

[4. Ορισμός triggers για την «αυτόματη» διαχείριση στοιχείων της βάσης 7](#_Toc401504267)

[5. Ορισμός trigger για τη διαχείριση τιμής στήλης (salary) 8](#_Toc401504268)

[6. Ορισμός trigger για τη διαχείριση τιμής στήλης (course\_name) 11](#_Toc401504269)

[7. Δοκιμή ορισμού δύο (2) εναυσμάτων (triggers) με την ίδια συνθήκη ενεργοποίησης 13](#_Toc401504270)

[8. Πως θα διορθώσουμε το πρόβλημα που προέκυψε κατά τη δοκιμή ορισμού δύο (2) triggers με την ίδια συνθήκη ενεργοποίησης 15](#_Toc401504271)

[9. Ορισμός trigger για την αυτόματη εισαγωγή στοιχείων σε πίνακα audit 17](#_Toc401504272)

Στόχος του πέμπτου εργαστηρίου είναι η κατανόηση σε βάθος της τεχνολογίας διαχείριση της βάσης με χρήση stored procedures:. Ειδικότερα εξετάζονται θέματα υλοποίησης με: cursors, procedures, functions, triggers. Μετά την επεξεργασία του εργαστηρίου ο ενδιαφερόμενος θα έχει κατανοήσει και θα έχει εμπεδώσει τον τρόπο χρήσης stored procedures: cursors, procedures, functions, triggers.

**Λέξεις Κλειδιά**: stored procedures, cursors, functions, procedures, triggers

# Δημιουργία βάσης δεδομένων training

Η βάση δεδομένων δε βρίσκεται στην Τρίτη Κανονική Μορφή

Lecturer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lecturer\_id | Lecturer\_surname | Lecturer\_name | City | salary | Course\_id | Course\_name |
|  |  |  |  |  |  |  |

TABLE course(course\_id int, course\_name varchar(50))

Course

|  |  |
| --- | --- |
| Course\_id | course\_name |
|  |  |

**Άσκηση**

Ανασχεδιάστε τη βάση στην Τρίτη Κανονική Μορφή.

# Παράδειγμα χρήσης Cursor

**Δημιουργία βάσης δεδομένων και πινάκων**

Drop database training;

CREATE DATABASE training;

USE training;

CREATE TABLE lecturer(lecturer\_id int(3),

 lecturer\_surname varchar(15), lecturer\_name varchar(15),

 city varchar(15), salary decimal (8,2), course\_id int, course\_name varchar(15));

CREATE TABLE course(course\_id int, course\_name varchar(50));



**Εισαγωγή στοιχείων**

INSERT INTO course VALUES (1, 'DATABASE');

INSERT INTO course VALUES (2, 'WEB DEVELOPMENT');

INSERT INTO course VALUES (3, 'DATA MINING');

INSERT INTO course VALUES (4, 'SEMANTIC WEB');

Select \* From COURSE;



INSERT INTO lecturer(lecturer\_id, lecturer\_name, lecturer\_surname, city, salary, course\_id) VALUES (1, 'CHRIS', 'DATE', 'LONDON', 2000, 1), (2, 'GIO', 'WIEDERHOLD', 'ATHENS', 1500, 1), (3, 'PETER', 'CHEN', 'ATHENS', 3500, 2), (4, 'JEFF', 'ULLMAN', 'ATHENS', 1700, 1), (5, 'TED', 'CODD', 'ATHENS', 2500, 2);

SELECT \* FROM LECTURER;

SELECT lecturer\_id, lecturer\_surname, lecturer\_name, course\_id, course\_name

FROM lecturer;



# Ορισμός συνάρτησης για τη διαχείριση Cursor

**Παρατηρήστε στη συνέχεια τις εντολές:**

* **Declare variables**
* **DECLARE CONTINUE HANDLER**
* **Open Cursor**
* **Fetch Cursor**
* **Close Cursor**

DELIMITER //
CREATE FUNCTION lecturer\_list() RETURNS VARCHAR(255) BEGIN
DECLARE record\_not\_found INTEGER DEFAULT 0;
DECLARE lecturer\_name\_var VARCHAR(150) DEFAULT ""; DECLARE lecturer\_surname\_var VARCHAR(150) DEFAULT "";
DECLARE lect\_list VARCHAR(255) DEFAULT "";
DECLARE my\_cursor CURSOR FOR SELECT lecturer\_name, lecturer\_surname FROM lecturer; DECLARE CONTINUE HANDLER FOR NOT FOUND SET record\_not\_found = 1;   OPEN my\_cursor;
 allLecturers: LOOP
 FETCH my\_cursor INTO lecturer\_name\_var, lecturer\_surname\_var;
 IF record\_not\_found THEN
 LEAVE allLecturers;
 END IF;
 SET lect\_list = CONCAT(lect\_list, lecturer\_surname\_var, ", ");
 END LOOP allLecturers;
 CLOSE my\_cursor;
 RETURN SUBSTR(lect\_list, 1, 70);
 END //


DELIMITER ;

**Χρήση της συνάρτησης**

SELECT lecturer\_list();



**Κατάργηση της συνάρτησης**

DROP FUNCTION lecturer\_list;



# Ορισμός triggers για την «αυτόματη» διαχείριση στοιχείων της βάσης

**Να η εικόνα της βάσης χωρίς triggers**

Select \* From COURSE;



SELECT lecturer\_id, lecturer\_surname, lecturer\_name, course\_id, course\_name

FROM lecturer;



**Πως ορίζεται η βάση δεδομένων, οι πίνακές της.**

Drop database training;

CREATE DATABASE training;

USE training;

CREATE TABLE lecturer(lecturer\_id int(3),

 lecturer\_surname varchar(15), lecturer\_name varchar(15),

 city varchar(15), salary decimal (8,2), course\_id int, course\_name varchar(15));

CREATE TABLE course(course\_id int, course\_name varchar(50));

# Ορισμός trigger για τη διαχείριση τιμής στήλης (salary)

drop trigger bi\_lecturer;

DELIMITER //

create trigger bi\_lecturer before insert on lecturer for each row

Begin

 declare name\_l int;

 set name\_l = length(new.lecturer\_name);

 set new.salary = new.salary + (new.salary \* name\_l)/10;

end;

//

DELIMITER ;



**Εισαγωγή στοιχείων. Επηρεάζεται από τον trigger**

INSERT INTO course VALUES (1, 'DATABASE');

INSERT INTO course VALUES (2, 'WEB DEVELOPMENT');

INSERT INTO course VALUES (3, 'DATA MINING');

INSERT INTO course VALUES (4, 'SEMANTIC WEB');

Select \* From COURSE;



INSERT INTO lecturer(lecturer\_id, lecturer\_name, lecturer\_surname, city, salary, course\_id) VALUES (1, 'CHRIS', 'DATE', 'LONDON', 2000, 1), (2, 'GIO', 'WIEDERHOLD', 'ATHENS', 1500, 1), (3, 'PETER', 'CHEN', 'ATHENS', 3500, 2), (4, 'JEFF', 'ULLMAN', 'ATHENS', 1700, 1), (5, 'TED', 'CODD', 'ATHENS', 2500, 2);

**Δείτε τις τιμές της στήλης salary**

SELECT lecturer\_id, lecturer\_surname, lecturer\_name, salary, course\_id

FROM lecturer;



**Δείτε τις τιμές της στήλης course\_name**

SELECT lecturer\_id, lecturer\_surname, lecturer\_name, course\_id, course\_name

FROM lecturer;



# Ορισμός trigger για τη διαχείριση τιμής στήλης (course\_name)

Drop database training;

CREATE DATABASE training;

USE training;

CREATE TABLE lecturer(lecturer\_id int(3),

 lecturer\_surname varchar(15), lecturer\_name varchar(15),

 city varchar(15), salary decimal (8,2), course\_id int, course\_name varchar(15));

CREATE TABLE course(course\_id int, course\_name varchar(50));

Drop trigger bi\_lect\_add\_dname;

DELIMITER //

create trigger bi\_lect\_add\_dname

before insert on lecturer

for each row

begin

declare dname\_var varchar(40);

 select course\_name into dname\_var from course where course\_id = new.course\_id;

 set new.course\_name = dname\_var;

end;

 //

DELIMITER ;

INSERT INTO course VALUES (1, 'DATABASE');

INSERT INTO course VALUES (2, 'WEB DEVELOPMENT');

INSERT INTO course VALUES (3, 'DATA MINING');

INSERT INTO course VALUES (4, 'SEMANTIC WEB');

Select \* From COURSE;



INSERT INTO lecturer(lecturer\_id, lecturer\_name, lecturer\_surname, city, salary, course\_id) VALUES (1, 'CHRIS', 'DATE', 'LONDON', 2000, 1), (2, 'GIO', 'WIEDERHOLD', 'ATHENS', 1500, 1), (3, 'PETER', 'CHEN', 'ATHENS', 3500, 2), (4, 'JEFF', 'ULLMAN', 'ATHENS', 1700, 1), (5, 'TED', 'CODD', 'ATHENS', 2500, 2);

**Να το αποτέλεσμα της εκτέλεσης του trigger κατά την εισαγωγή στοιχείων στον πίνακα lecturer**

SELECT lecturer\_id, lecturer\_surname, lecturer\_name, course\_id, course\_name

FROM lecturer;



# Δοκιμή ορισμού δύο (2) εναυσμάτων (triggers) με την ίδια συνθήκη ενεργοποίησης

Drop database training;

CREATE DATABASE training;

USE training;

CREATE TABLE lecturer(lecturer\_id int(3),

 lecturer\_surname varchar(15), lecturer\_name varchar(15),

 city varchar(15), salary decimal (8,2), course\_id int, course\_name varchar(15));

CREATE TABLE course(course\_id int, course\_name varchar(50));

drop trigger bi\_lecturer;

DELIMITER //

create trigger bi\_lecturer before insert on lecturer for each row

Begin

 declare name\_l int;

 set name\_l = length(new.lecturer\_name);

 set new.salary = new.salary + (new.salary \* name\_l)/10;

end;

//

DELIMITER ;

DELIMITER //

create trigger bi\_stud\_add\_dname

before insert on lecturer

for each row

begin

declare dnamev varchar(40);

 select course\_name into dnamev from course where course\_id = new.course\_id;

 set new.course\_name = dnamev;

end;

 //

DELIMITER ;

INSERT INTO course VALUES (1, 'DATABASE');

INSERT INTO course VALUES (2, 'WEB DEVELOPMENT');

INSERT INTO course VALUES (3, 'DATA MINING');

INSERT INTO course VALUES (4, 'SEMANTIC WEB');

Select \* From COURSE;

INSERT INTO lecturer(lecturer\_id, lecturer\_name, lecturer\_surname, city, salary, course\_id) VALUES (1, 'CHRIS', 'DATE', 'LONDON', 2000, 1), (2, 'GIO', 'WIEDERHOLD', 'ATHENS', 1500, 1), (3, 'PETER', 'CHEN', 'ATHENS', 3500, 2), (4, 'JEFF', 'ULLMAN', 'ATHENS', 1700, 1), (5, 'TED', 'CODD', 'ATHENS', 2500, 2);

SELECT lecturer\_id, lecturer\_surname, lecturer\_name, course\_id, course\_name

FROM lecturer;



**Μήνυμα που επιστρέφεται**





# Πως θα διορθώσουμε το πρόβλημα που προέκυψε κατά τη δοκιμή ορισμού δύο (2) triggers με την ίδια συνθήκη ενεργοποίησης

Drop database training;

CREATE DATABASE training;

USE training;

CREATE TABLE lecturer(lecturer\_id int(3),

 lecturer\_surname varchar(15), lecturer\_name varchar(15),

 city varchar(15), salary decimal (8,2), course\_id int, course\_name varchar(15));

CREATE TABLE course(course\_id int, course\_name varchar(50));

drop trigger bi\_lecturer;

DELIMITER //

create trigger bi\_lecturer before insert on lecturer for each row

Begin

 declare name\_length int;

 declare dname\_var varchar(40);

 select course\_name into dname\_var from course where course\_id = new.course\_id;

 set new.course\_name = dname\_var;

 set name\_length = length(new.lecturer\_name);

 set new.salary = new.salary + (new.salary \* name\_length)/10;

end;

 //

DELIMITER ;



INSERT INTO course VALUES (1, 'DATABASE');

INSERT INTO course VALUES (2, 'WEB DEVELOPMENT');

INSERT INTO course VALUES (3, 'DATA MINING');

INSERT INTO course VALUES (4, 'SEMANTIC WEB');

Select \* From COURSE;

INSERT INTO lecturer(lecturer\_id, lecturer\_name, lecturer\_surname, city, salary, course\_id) VALUES (1, 'CHRIS', 'DATE', 'LONDON', 2000, 1), (2, 'GIO', 'WIEDERHOLD', 'ATHENS', 1500, 1), (3, 'PETER', 'CHEN', 'ATHENS', 3500, 2), (4, 'JEFF', 'ULLMAN', 'ATHENS', 1700, 1), (5, 'TED', 'CODD', 'ATHENS', 2500, 2);

**Να τι βλέπουμε**

SELECT lecturer\_surname, lecturer\_name, salary, course\_id, course\_name

FROM lecturer;



# Ορισμός trigger για την αυτόματη εισαγωγή στοιχείων σε πίνακα audit

Drop database training;

CREATE DATABASE training;

USE training;

CREATE TABLE lecturer(lecturer\_id int(3),

 lecturer\_surname varchar(15), lecturer\_name varchar(15),

 city varchar(15), salary decimal (8,2), course\_id int, course\_name varchar(15));

CREATE TABLE course(course\_id int, course\_name varchar(50));

DELIMITER //

create table audit (user\_name varchar(30), table\_name varchar(30), update\_date date);

create trigger bi\_lecturer\_audit before insert on lecturer for each row

begin

 insert into audit (user\_name, table\_name, update\_date) values (current\_user(),'lecturer', now());

end;

//

DELIMITER ;

INSERT INTO course VALUES (1, 'DATABASE');

INSERT INTO course VALUES (2, 'WEB DEVELOPMENT');

INSERT INTO course VALUES (3, 'DATA MINING');

INSERT INTO course VALUES (4, 'SEMANTIC WEB');

Select \* From COURSE;

INSERT INTO lecturer(lecturer\_id, lecturer\_name, lecturer\_surname, city, salary, course\_id) VALUES (1, 'CHRIS', 'DATE', 'LONDON', 2000, 1), (2, 'GIO', 'WIEDERHOLD', 'ATHENS', 1500, 1), (3, 'PETER', 'CHEN', 'ATHENS', 3500, 2), (4, 'JEFF', 'ULLMAN', 'ATHENS', 1700, 1), (5, 'TED', 'CODD', 'ATHENS', 2500, 2);

SELECT lecturer\_surname, lecturer\_name, salary, course\_id, course\_name

FROM lecturer;

Select \* from audit;



|  |
| --- |
| **Ανοικτά Ακαδημαϊκά Μαθήματα****Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας** |
| **Τέλος Ενότητας** |
| **Χρηματοδότηση*** Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
* Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Αθήνας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
* Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

 |

**Σημειώματα**

**Σημείωμα Αναφοράς**

Copyright ΤΕΙ Αθήνας, Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης 2014. Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης. «Βάσεις Δεδομένων ΙΙ (Ε). Ενότητα 5: Χρήση stored procedures: cursors, functions, procedures, triggers». Έκδοση: 1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: [ocp.teiath.gr](https://ocp.teiath.gr/).

**Σημείωμα Αδειοδότησης**

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό. Οι όροι χρήσης των έργων τρίτων επεξηγούνται στη διαφάνεια «Επεξήγηση όρων χρήσης έργων τρίτων».

Τα έργα για τα οποία έχει ζητηθεί άδεια αναφέρονται στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

* που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
* που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
* που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

**Επεξήγηση όρων χρήσης έργων τρίτων**

|  |  |
| --- | --- |
| © | Δεν επιτρέπεται η επαναχρησιμοποίηση του έργου, παρά μόνο εάν ζητηθεί εκ νέου άδεια από το δημιουργό. |
| διαθέσιμο με άδεια CC-BY | Επιτρέπεται η επαναχρησιμοποίηση του έργου και η δημιουργία παραγώγων αυτού με απλή αναφορά του δημιουργού. |
| διαθέσιμο με άδεια CC-BY-SA | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού, και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια. |
| διαθέσιμο με άδεια CC-BY-ND | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η δημιουργία παραγώγων του έργου. |
| διαθέσιμο με άδεια CC-BY-NC | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η εμπορική χρήση του έργου. |
| διαθέσιμο με άδεια CC-BY-NC-SA | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια. Δεν επιτρέπεται η εμπορική χρήση του έργου. |
| διαθέσιμο με άδεια CC-BY-NC-ND | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η εμπορική χρήση του έργου και η δημιουργία παραγώγων του. |
| διαθέσιμο με άδεια CC0 Public Domain | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού. |
| διαθέσιμο ως κοινό κτήμα | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού. |
| χωρίς σήμανση | Συνήθως δεν επιτρέπεται η επαναχρησιμοποίηση του έργου. |

**Διατήρηση Σημειωμάτων**

* Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
* το Σημείωμα Αναφοράς
* το Σημείωμα Αδειοδότησης
* τη δήλωση Διατήρησης Σημειωμάτων
* το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει) μαζί με τους συνοδευόμενους υπερσυνδέσμους.