

Βάσεις Δεδομένων II (Ε)

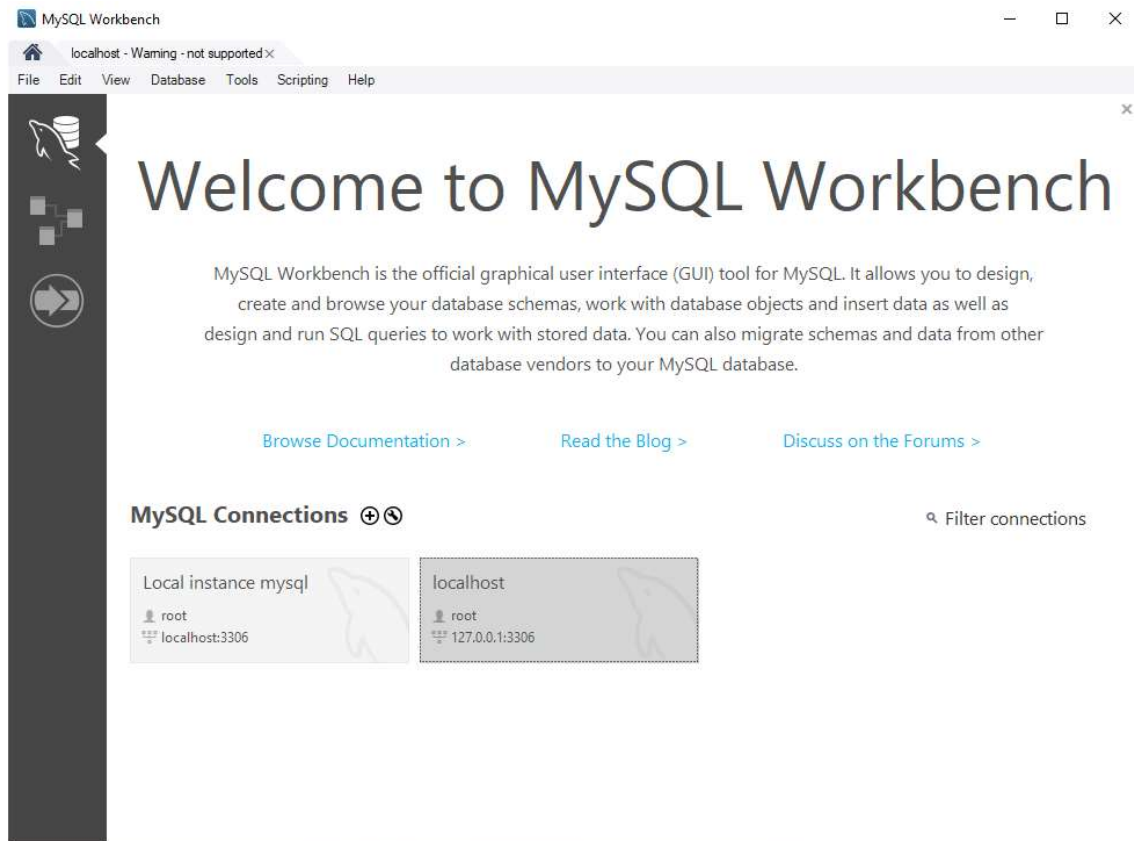
A.M.: ice18390094

Ονοματεπώνυμο: Άγγελος Τζώρτζης

Εργαστηριακή άσκηση 4

Τμήμα: [00] Χωρίς παρακολούθηση

1. Σύνδεση στην MySQL του συστήματος μας:



2. Ελέγχουμε εάν υπάρχει η ΒΔ με όνομα personnel:

Εμφάνιση όλων των ΒΔ του συστήματος μας

show databases;

ΒΔ στο σύστημα μας:

	Database
▶	information_schema
	mysql
	performance_schema
	personnel
	phpmyadmin
	test

Υπάρχει η ΒΔ με όνομα personnel στο σύστημα μας.

3. Επιλέγουμε την ΒΔ με όνομα personnel για χρήση:

```
use personnel;
```

Δημιουργήσαμε τους πίνακες με την δομή και τα περιεχόμενα με βάση την εκφώνηση της εργασίας.

4. Δημιουργία του trigger με όνομα dept_update που αφυπνίζεται από το γεγονός "BEFORE UPDATE ON":

```
create trigger dept_update
before update on DEPT
for each row
begin
set new.DNAME = upper(new.DNAME);
end;
//
delimiter ;
```

```
update DEPT set DNAME = 'Operations' WHERE deptno = 70;
select * from DEPT;
```

Τα στοιχεία του πίνακα DEPT:

	DEPTNO	DNAME	LOC
▶	50	SALES	ATHENS
	60	ACCOUNTING	ATHENS
	70	OPERATIONS	VOLOS
*	NULL	NULL	NULL

Ο trigger αυτός έχει σαν λειτουργία πριν κάποιο update να μετατρέπει τα γράμματα του νέου όνομα που ορίσαμε στο DEPTNO από πεζά σε κεφαλαία (όποια γράμματα είναι ήδη κεφαλαία μένουν ίδια). Όπως βλέπουμε στα περιεχόμενα του πίνακα DEPT μετά το update ενώ αλλάξαμε το όνομα του τμήματος 70 σε 'Operations' με την επίδραση του trigger, στον πίνακα εμφανίζεται ως 'OPERATIONS'.

5. Προσθήκη στον πίνακα Department τη στήλη no_of_employees (αριθμός υπαλλήλων που έχει το τμήμα).

```
alter table DEPT add (no_of_employees INT);
describe DEPT;
```

Δομή του πίνακα DEPT:

	Field	Type	Null	Key	Default	Extra
▶	DEPTNO	decimal(2,0)	NO	PRI	NULL	
	DNAME	varchar(24)	YES		NULL	
	LOC	char(23)	YES		NULL	
	no_of_employees	int(11)	YES		NULL	

Βλέπουμε την προσθήκη της στήλης no_of_employees με τον τύπο δεδομένων int, στο πίνακα DEPT.

6. Ενημέρωση στον πίνακα Department τη στήλη no_of_employees (αριθμός υπαλλήλων που έχει το τμήμα):

```
update DEPT
set no_of_employees = (
    select count(*)
    from EMP
    where EMP.DEPTNO = DEPT.DEPTNO
);
select * from DEPT;
```

Περιεχόμενα του πίνακα DEPT:

	DEPTNO	DNAME	LOC	no_of_employees
▶	50	SALES	ATHENS	3
	60	ACCOUNTING	ATHENS	1
	70	OPERATIONS	VOLOS	0
*	NULL	NULL	NULL	NULL

Έχει προστεθεί η στήλη στον πίνακα DEPT και βλέπουμε ότι “μετράει” σωστά τον αριθμό των υπαλλήλων σε κάθε τμήμα.

7. Δημιουργία του trigger με όνομα emp_insert που αφυπνίζεται από το γεγονός "AFTER INSERT ON":

```
delimiter //
create trigger emp_insert
after insert on EMP
for each row
begin
update DEPT
set no_of_employees = ifnull(no_of_employees, 0) + 1
where DEPT.DEPTNO = new.DEPTNO;
end;
//
delimiter ;
```

Ο trigger αυτός έχει σαν λειτουργία την αύξηση του αριθμού των υπαλλήλων κατα 1 σε ένα τμήμα μετά την εισαγωγή μιας νέας εγγραφής στον πίνακα EMP που ανήκει σε αυτό το τμήμα.

8. Δοκιμή του παραπάνω trigger:

```
insert into EMP(EMPNO, ENAME, JOBNO, DEPTNO, COMM)
values(7985, 'CLARKE', 100, 70, NULL);
select * from EMP;
select * from DEPT;
```

Πίνακας EMP:

	EMPNO	ENAME	JOBNO	DEPTNO	COMM
▶	10	CODD	100	50	NULL
	20	NAVATHE	200	50	450.00
	30	ELMASRI	300	60	NULL
	40	DATE	100	50	NULL
	7985	CLARKE	100	70	NULL
*	NULL	NULL	NULL	NULL	NULL

Πίνακας DEPT:

	DEPTNO	DNAME	LOC	no_of_employees
▶	50	SALES	ATHENS	3
	60	ACCOUNTING	ATHENS	1
	70	OPERATIONS	VOLOS	1
*	NULL	NULL	NULL	NULL

Βλέπουμε ότι έγινε η εισαγωγή της νέας εγγραφής στον πίνακα EMP. Επίσης βλέπουμε ότι στον πίνακα DEPT αυξήθηκε ο αριθμός υπαλλήλων του τμήματος 70 αφού εκεί έγινε η εισαγωγή της εγγραφής μας.

9. Εξήγηση της λειτουργίας του παραπάνω trigger:

Στο φυλλάδιο γίνεται επεξήγηση της λειτουργίας του παραπάνω trigger καθώς και δίνεται παράδειγμα αντικατάστασης της συνάρτησης ifnull με την NVL που λειτουργούν με τον ακριβώς ίδιο τρόπο.

10. Δημιουργία, δοκιμή και εξήγηση του trigger που αφυπνίζεται από το γεγονός AFTER DELETE ON:

```
create trigger emp_delete
after delete on EMP
for each row
begin
update DEPT
set no_of_employees = ifnull(no_of_employees, 0) - 1
where DEPT.DEPTNO = old.DEPTNO;
end;
//
delimiter ;

delete from EMP where EMPNO = 7985;
select * from EMP;
select * from DEPT;
```

Ο trigger αυτός έχει σαν λειτουργία την μείωση του αριθμού των υπαλλήλων κατα 1 σε ένα τμήμα μετά την διαγραφή μιας εγγραφής στον πίνακα EMP που ανήκει σε αυτό το τμήμα.

Πίνακας EMP:

	EMPNO	ENAME	JOBNO	DEPTNO	COMM
▶	10	CODD	100	50	NULL
	20	NAVATHE	200	50	450.00
	30	ELMASRI	300	60	NULL
	40	DATE	100	50	NULL
*	NULL	NULL	NULL	NULL	NULL

Πίνακας DEPT:

	DEPTNO	DNAME	LOC	no_of_employees
▶	50	SALES	ATHENS	3
	60	ACCOUNTING	ATHENS	1
	70	OPERATIONS	VOLOS	0
*	NULL	NULL	NULL	NULL

Βλέπουμε ότι έχει διαγραφεί ο υπάλληλος με κωδικό 7985 και ο αριθμός υπαλλήλων στο τμήμα που ανήκει μειώθηκε κατα 1.

11. Δημιουργία, δοκιμή και εξήγηση του trigger που αφυπνίζεται από το γεγονός AFTER DELETE ON:

```
delimiter //
create trigger emp_update
after update on EMP
for each row
begin
update DEPT
set no_of_employees = ifnull(no_of_employees, 0) - 1
where DEPT.DEPTNO = old.DEPTNO;
update DEPT
set no_of_employees = ifnull(no_of_employees, 0) + 1
where DEPT.DEPTNO = new.DEPTNO;
end;
//
delimiter ;

update EMP set DEPTNO = 70 where EMPNO = 10;
select * from EMP;
select * from DEPT;
```

Ο trigger αυτός μετά την ενημέρωση μιας εγγραφής, μειώνει κατά 1 τον αριθμό των υπαλλήλων στην γραμμή που ήταν ο παλιός αριθμός τμήματος του υπαλλήλου και στην συνέχεια αυξάνει κατά 1 τον αριθμό των υπαλλήλων στην γραμμή που είναι ο καινούργιος αριθμός τμήματος του υπαλλήλου.

Πίνακας EMP:

	EMPNO	ENAME	JOBNO	DEPTNO	COMM
▶	10	CODD	100	70	NULL
	20	NAVATHE	200	50	450.00
	30	ELMASRI	300	60	NULL
	40	DATE	100	50	NULL
*	NULL	NULL	NULL	NULL	NULL

Πίνακας DEPT:

	DEPTNO	DNAME	LOC	no_of_employees
▶	50	SALES	ATHENS	2
	60	ACCOUNTING	ATHENS	1
	70	OPERATIONS	VOLOS	1
*	NULL	NULL	NULL	NULL

Βλέπουμε ότι στον πίνακα EMP η εγγραφή με EMPNO ίσο με 10 πλέον έχει στο πεδίο DEPTNO την τιμή 70 αντί για 50. Αντίστοιχα στον πίνακα DEPT ο αριθμός των υπαλλήλων στο τμήμα 50 έχουν μειωθεί κατά 1 και στο τμήμα 70 αυξήθηκαν κατά 1.

12. Παρουσίαση πληροφοριών για τους Triggers:

```
describe Information_schema.Triggers;  
select TRIGGER_NAME, EVENT_MANIPULATION, TRIGGER_SCHEMA  
from Information_schema.Triggers  
where TRIGGER_SCHEMA = 'personnel'  
order by TRIGGER_NAME;
```

Εδώ φαίνονται οι πληροφορίες για τους triggers:

	TRIGGER_NAME	EVENT_MANIPULATION	TRIGGER_SCHEMA
▶	dept_update	UPDATE	personnel
	emp_delete	DELETE	personnel
	emp_insert	INSERT	personnel
	emp_update	UPDATE	personnel

13. Διαγραφή των triggers:

```
drop trigger dept_update;  
drop trigger emp_insert;  
drop trigger emp_delete;  
drop trigger emp_update;
```

Αποτέλεσμα όταν προσπαθήσουμε να εμφανίσουμε πληροφορίες για τους triggers:

	TRIGGER_NAME	EVENT_MANIPULATION	TRIGGER_SCHEMA

Προσπαθώντας να εμφανίσουμε πληροφορίες για τους triggers βλέπουμε ότι δεν εμφανίζεται τίποτα άρα πραγματοποιήθηκε η διαγραφή τους.