



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

DEPARTMENT OF INFORMATION AND COMPUTER ENGINEERING

4th LABORATORY EXERCISE

TRIGGER

WORK DETAILS

LABORATORY DEPARTMENT: [06] WEDNESDAY 13:00-14:00

LABORATORY RESPONSIBILITY : GAROFALAKI RANIA

DELIVERY DATE: 12/27/2024

SUBMISSION DEADLINE: 1/19/2024

DATA BASES II

STUDENT DETAILS

STUDENT PHOTO:



NAME: ATHANASIOU VASILEIOS EVANGELOS

STUDENT ID: 19390005

STUDENT SEMESTER: 9th

STUDENT STATUS : UNDERGRADUATE

PROGRAM OF STUDY : UNIWA

DATA BASES II

CONTENTS

DB personnel	6
personnel.sql	6
personnel. png	8
Activities	8
1. Connect to your system's	8
1.1. Statement	8
1.2. Result	9
1.3. Snapshot	9
2. Check if there is a DB named personnel	10
2.1. Statement	10
2.2. Result	10
2.3. Snapshot	10
3. Select the DB personnel to use	11
3.1. Statement	11
3.2. Result	11
3.3. Snapshot	11
4. Create the trigger named dept _ update which is awakened by the event “ BEFORE UPDATE ON ” ...	11
4.1. Statement	11
4.2. Result	12
4.3. Snapshot	13
5. In the Department table we will add the column no _ of _ employees (number of employees the department has)	14
5.1. Statement	14
5.2. Result	14
5.3. Snapshot	15
6. In the Department table we will add the column no _ of _ employees (number of employees the department has)	15
6.1. Statement	15
6.2. Result	16
6.3. Snapshot	16

DATA BASES II

7. In the Trigger table that is awakened by the AFTER event INSERT ON	17
7.1. Statement	17
7.2. Result	17
7.3. Snapshot	18
8. Test	18
8.1. Statement	18
8.2. Result	19
8.3. Snapshot	20
9. Explanation	20
9.1. Statement	20
9.2. Result	21
9.3. Snapshot	21
10. Create, test and explain the trigger that is awakened by the AFTER event DELETE ON	22
10.1. Statement	22
10.2. Result	24
10.3. Snapshot	26
10.4. Explanation	28
11. Create, test and explain the trigger that is awakened by the AFTER event UPDATE ON	29
11.1. Statement	29
11.2. Result	32
11.3. Snapshot	34
11.4. Explanation	38
12. Presentation of information on Triggers	39
12.1. Statement	39
12.2. Result	39
12.3. Snapshot	42
13. Deletion of Triggers	43
13.1. Statement	43
13.2. Result	43
13.3. Snapshot	44

DATA BASES II

DB personnel

personnel.sql

```
drop database if exists personnel?
```

```
create database personnel?
```

```
use personnel?
```

```
create table
```

```
DEPT (
```

```
DEPTNO numeric( 2),
```

```
DNAME varchar( 24 ),
```

```
LOC char( 23)
```

```
);
```

```
insert into
```

```
DEPT
```

```
(DEPTNO, DNAME, LOC)
```

```
values
```

```
(50, 'SALES', 'ATHENS'),
```

```
(60, 'ACCOUNTING', 'ATHENS'),
```

```
(70, 'PAYROL', 'VOLOS');
```

```
create table
```

```
JOB(
```

```
JOB_CODE numeric( 3),
```

```
JOB_DESCR varchar( 24 ),
```

```
SAL numeric( 10.2)
```

```
);
```

DATA BASES II

```
insert into  
JOB  
(JOBCODE, JOB_DESCR, SAL)  
values  
(100, 'SALESMAN', 2000),  
(200, 'ANALYST', 2000),  
(300, 'DBA', 3000);
```

```
create table  
EMP(  
EMPNO numeric( 4),  
NAME varchar( 255 ),  
JOBNO numeric( 3),  
DEPTNO numeric( 2),  
COMM numeric( 10,2)  
);
```

```
insert into  
EMP  
(EMPNO, NAME, JOBNO, DEPTNO, COMM)  
values  
(10, 'Codd', 100, 50, NULL),  
(20, 'NAVATHE', 200, 50, 450 ),  
(30, 'ELMASRI', 300, 60, NULL),  
(40, 'DATE', 100, 50, NULL);
```

DATA BASES II

personnel. png

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

Activities

1. Connect to your system's MySQL using any of the above methods you wish

1.1. Statement

```
"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --default-character-set=utf8mb4 -u root -p
```

Enter password: *****

1.2. Result

DATA BASES II

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 8

Server version: 8.0.35 MySQL Community Server - GPL

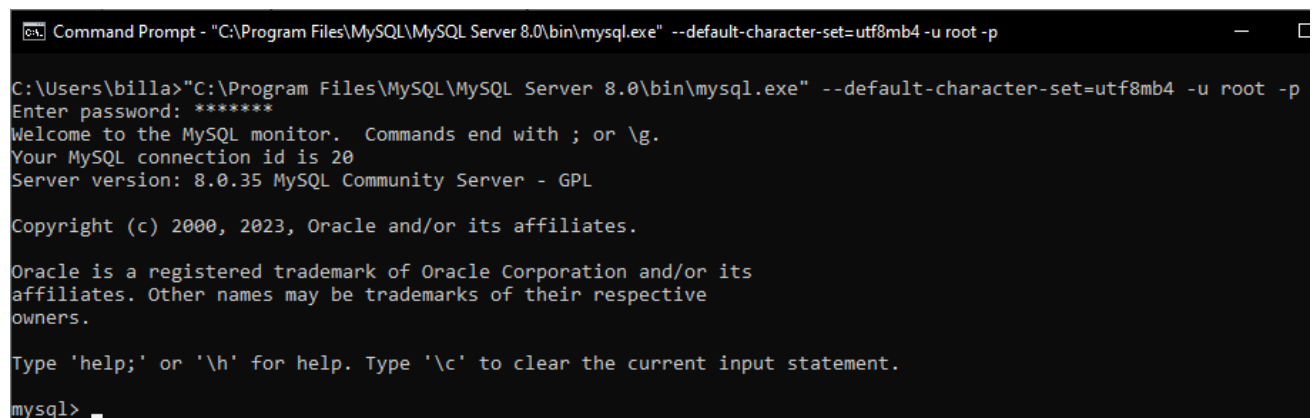
Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates . Other names may be trademarks of their respective companies owners .

Type 'help?' or '\h' for help. Type '\c' to clear the current input statement.

mysql >

1.3. Snapshot



```
Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --default-character-set=utf8mb4 -u root -p
C:\Users\bill>"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --default-character-set=utf8mb4 -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```


DATA BASES II

2 . Check if there is a DB named personnel

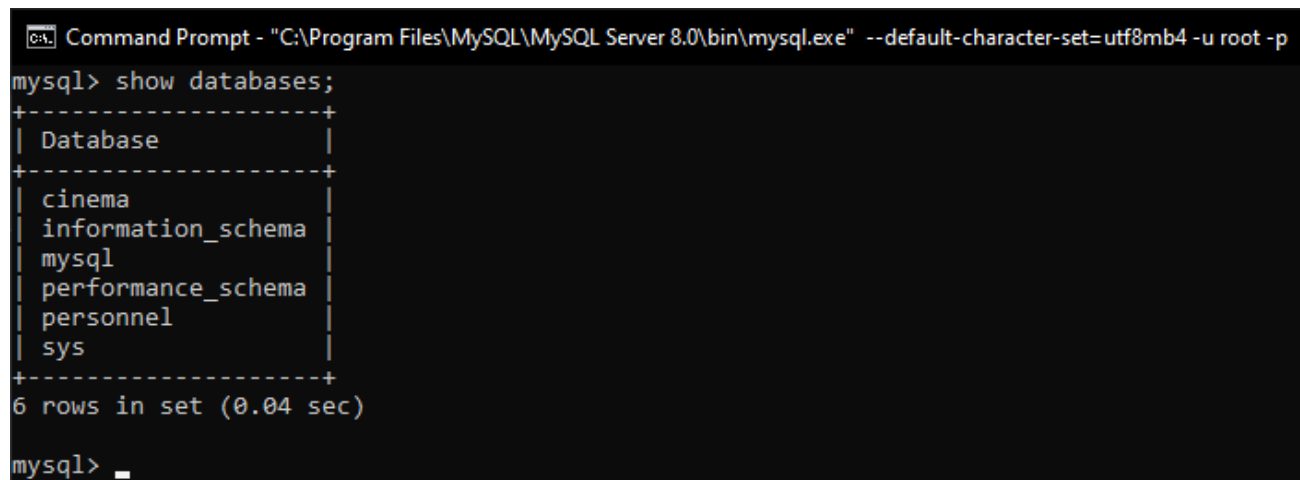
2.1. Statement

```
show databases ;
```

2.2. Result

```
+-----+
| Database |
+-----+
| cinema |
| information_schema |
| mysql |
| performance_schema |
| personnel |
| sys |
+-----+
6 rows in set (0.04 sec)
```

2.3. Snapshot



The screenshot shows a Windows Command Prompt window with the title "Command Prompt - 'C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe' --default-character-set=utf8mb4 -u root -p". The prompt is "mysql>". The command "show databases;" has been entered and executed. The output is a table with one column, "Database", listing the following databases: "cinema", "information_schema", "mysql", "performance_schema", "personnel", and "sys". The output is formatted with a header row and separator lines. Below the table, it says "6 rows in set (0.04 sec)". The prompt "mysql> _" is visible at the bottom.

```
C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --default-character-set=utf8mb4 -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| cinema |
| information_schema |
| mysql |
| performance_schema |
| personnel |
| sys |
+-----+
6 rows in set (0.04 sec)
mysql> _
```

DATA BASES II

3 . Select the DB personnel to use

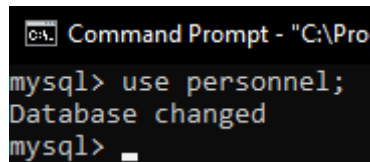
3.1. Statement

use personnel?

3.2. Result

Database changed

3.3. Snapshot



```
C:\Pro
mysql> use personnel;
Database changed
mysql> _
```

4. Create the trigger named dept _ update which is awakened by the event “ BEFORE UPDATE ON ”

4.1. Statement

```
delimiter //

...

CREATE TRIGGER

    dept_update

BEFORE UPDATE ON

    dept

FOR EACH ROW

BEGIN

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

END?

//
```

DATA BASES II

```
...
delimiter ;
...
UPDATE
    dept
SET
    dname = 'Operations'
WHERE
    deptno = 70;
//
...
SELECT * FROM department ;
```

4.2. Result

```
...
Query OK, 0 rows affected (0.01 sec)
...

...
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed : 1 Warnings: 0
...
+-----+-----+-----+
| DEPTNO | DNAME | LOC |
+-----+-----+-----+
| 50 | SALES | ATHENS |
| 60 | ACCOUNTING | ATHENS |
```

DATA BASES II

| 70 | OPERATIONS | VOLOS |

+-----+-----+-----+

3 rows in set (0.00 sec)

4.3. Snapshot

```
Command Prompt - "C:\Program Files\MySQL\MySQL Se
mysql> delimiter //
mysql> CREATE TRIGGER
-> dept_update
-> BEFORE UPDATE ON
-> dept
-> FOR EACH ROW
-> BEGIN
-> SET NEW.dname = UPPER(NEW.dname);
-> END;
-> //
Query OK, 0 rows affected (0.01 sec)

mysql> _
```

...

```
Command Prompt - "C:\Program Files\MySQL\MySQL S
mysql> delimiter ;
mysql> UPDATE
-> dept
-> SET
-> dname = 'Operations'
-> WHERE
-> deptno = 70
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM dept;
+-----+-----+-----+
| DEPTNO | DNAME      | LOC      |
+-----+-----+-----+
|      50 | SALES      | ATHENS   |
|      60 | ACCOUNTING | ATHENS   |
|      70 | OPERATIONS | VOLOS    |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

DATA BASES II

5 . In the Department table we will add the column no _ of _ employees (number of employees the department has)

5.1. Statement

```
ALTER TABLE
    dept
ADD
( no_of_employees INT)
;
...
DESCRIBE department ;
```

5.2. Result

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates : 0 Warnings: 0

...

```
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| DEPTNO | decimal( 2,0 ) | YES | | NULL | |
| DNAME | varchar( 24 ) | YES | | NULL | |
| LOC | char( 23 ) | YES | | NULL | |
| no_of_employees | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
```

4 rows in set (0.00 sec)

DATA BASES II

5.3. Snapshot

```
Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --default-cha
mysql> ALTER TABLE
    -> dept
    -> ADD
    -> (no_of_employees INT)
    -> ;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE dept;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| DEPTNO         | decimal(2,0)  | YES  |     | NULL    |       |
| DNAME          | varchar(24)    | YES  |     | NULL    |       |
| LOC            | char(23)       | YES  |     | NULL    |       |
| no_of_employees | int           | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _
```

6. In the Department table we will add the column no _ of _ employees (number of employees the department has)

6.1. Statement

UPDATE

dept

SET

no_of_employees =

(

SELECT

COUNT(*)

FROM

emp

WHERE

emp.deptno = dept.deptno

DATA BASES II

);

...

SELECT * FROM department ;

6.2. Result

Query OK, 3 rows affected (0.00 sec)

Rows matched: 3 Changed : 3 Warnings: 0

...

```
+-----+-----+-----+-----+
| DEPTNO | DNAME | LOC | no_of_employees |
+-----+-----+-----+-----+
| 50 | SALES | ATHENS | 3 |
| 60 | ACCOUNTING | ATHENS | 1 |
| 70 | OPERATIONS | VOLOS | 0 |
+-----+-----+-----+-----+

3 rows in set (0.00 sec)
```

6.3. Snapshot

```
C:\Program Files\MySQL\MySQL Server 8.0\bin\mys
mysql> UPDATE
-> dept
-> SET
-> no_of_employees =
-> (
-> SELECT
-> COUNT(*)
-> FROM
-> emp
-> WHERE
-> emp.deptno = dept.deptno
-> );
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3 Changed: 3 Warnings: 0

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME | LOC | no_of_employees |
+-----+-----+-----+-----+
| 50 | SALES | ATHENS | 3 |
| 60 | ACCOUNTING | ATHENS | 1 |
| 70 | OPERATIONS | VOLOS | 0 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

7 . In the Trigger table that is awakened by the AFTER event INSERT ON

7.1. Statement

```
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 deptno = NEW.deptno ;  
END?  
  
//  
  
...  
  
delimiter ;
```

7.2. Result

```
...  
  
Query OK, 0 rows affected (0.00 sec)  
  
...
```


DATA BASES II

7.3 . Snapshot

```
Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe"
mysql> delimiter //
mysql> 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 deptno = NEW.deptno;
-> END;
-> //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> _
```

8. Testing

8.1. Statement

INSERT INTO

emp

(empno , name, deptno)

VALUES

(7895, 'CLARKE', 70)

;

...

SELECT * FROM emp ;

...

SELECT * FROM department ;

8.2 . Result

Query OK, 1 row affected (0.00 sec)

...

EMPNO	NAME	JOBNO	DEPTNO	COMM
10	CODD	100	50	NULL
20	NAVATHE	200	50	450.00
30	ELMASRI	300	60	NULL
40	DATE	100	50	NULL
7895	CLARKE	NULL	70	NULL

5 rows in set (0.00 sec)

...

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

3 rows in set (0.00 sec)

DATA BASES II

8.3 . Snapshot

```
Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql"
mysql> INSERT INTO
-> emp
-> (empno, name, deptno)
-> VALUES
-> (7895, 'CLARKE', 70)
-> ;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM emp;
+-----+-----+-----+-----+-----+
| EMPNO | NAME   | JOBNO | DEPTNO | COMM   |
+-----+-----+-----+-----+-----+
| 10    | CODD   | 100   | 50     | NULL   |
| 20    | NAVATHE | 200   | 50     | 450.00 |
| 30    | ELMASRI | 300   | 60     | NULL   |
| 40    | DATE   | 100   | 50     | NULL   |
| 7895  | CLARKE | NULL  | 70     | NULL   |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME      | LOC    | no_of_employees |
+-----+-----+-----+-----+
| 50     | SALES      | ATHENS | 3               |
| 60     | ACCOUNTING | ATHENS | 1               |
| 70     | OPERATIONS | VOLOS  | 1               |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

9. Explanation

9.1. Statement

UPDATE

dept

SET

no_of_employees = IFNULL(no_of_employees , 0) + 1

WHERE

deptno = 70

DATA BASES II

;

...

SELECT * FROM department ;

9.2. Result

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed : 1 Warnings: 0

...

```
+-----+-----+-----+-----+
| DEPTNO | DNAME | LOC | no_of_employees |
+-----+-----+-----+-----+
| 50 | SALES | ATHENS | 3 |
| 60 | ACCOUNTING | ATHENS | 1 |
| 70 | OPERATIONS | VOLOS | 2 |
+-----+-----+-----+-----+

3 rows in set (0.00 sec)
```

9.3. Snapshot

```
Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --
mysql> UPDATE
-> dept
-> SET
-> no_of_employees = IFNULL(no_of_employees, 0) + 1
-> WHERE
-> deptno = 70
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME | LOC | no_of_employees |
+-----+-----+-----+-----+
| 50 | SALES | ATHENS | 3 |
| 60 | ACCOUNTING | ATHENS | 1 |
| 70 | OPERATIONS | VOLOS | 2 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

DATA BASES II

10 . Create, test, and explain the trigger that is fired by the AFTER event DELETE ON

10.1. Statement

Creation

```
delimiter //
...
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
    deptno = OLD.deptno ;
END?
//
...
delimiter ;
```

Essay

```
SELECT * FROM emp ;
...
```

DATA BASES II

```
SELECT * FROM department ;
```

```
...
```

```
DELETE
```

```
FROM
```

```
    emp
```

```
WHERE
```

```
    empno = 7895
```

```
;
```

```
...
```

```
SELECT * FROM emp ;
```

```
...
```

```
SELECT * FROM department ;
```

Illustration

```
SELECT * FROM department ;
```

```
...
```

```
UPDATE
```

```
    dept
```

```
SET
```

```
    no_of_employees = IFNULL( no_of_employees , 0) - 1
```

```
WHERE
```

```
    deptno = 70
```

```
;
```

```
...
```

```
SELECT * FROM department ;
```

10.2 . Result

DATA BASES II

Creation

...

Query OK , 0 rows affected (0.01 sec)

...

Essay

```
+-----+-----+-----+-----+-----+
| EMPNO | NAME  | JOBNO | DEPTNO | COMM |
+-----+-----+-----+-----+-----+
| 10    | CODD  | 100   | 50     | NULL |
| 20    | NAVATHE | 200   | 50     | 450.00 |
| 30    | ELMASRI | 300   | 60     | NULL |
| 40    | DATE  | 100   | 50     | NULL |
| 7895  | CLARKE | NULL  | 70     | NULL |
```

```
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

...

```
+-----+-----+-----+-----+-----+
| DEPTNO | DNAME  | LOC   | no_of_employees |
+-----+-----+-----+-----+-----+
| 50     | SALES  | ATHENS | 3 |
| 60     | ACCOUNTING | ATHENS | 1 |
| 70     | OPERATIONS | VOLOS | 2 |
```

```
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

...

DATA BASES II

Query OK, 1 row affected (0.01 sec)

...

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

4 rows in set (0.00 sec)

...

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

3 rows in set (0.00 sec)

Illustration

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

DATA BASES II

+-----+-----+-----+-----+

3 rows in set (0.00 sec)

...

Query OK, 1 row affected (0.02 sec)

Rows matched: 1 Changed : 1 Warnings: 0

...

+-----+-----+-----+-----+

| DEPTNO | DNAME | LOC | no_of_employees |

+-----+-----+-----+-----+

| 50 | SALES | ATHENS | 3 |

| 60 | ACCOUNTING | ATHENS | 1 |

| 70 | OPERATIONS | VOLOS | 0 |

+-----+-----+-----+-----+

3 rows in set (0.00 sec)

10.3. Snapshot

Creation

```
Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe"
mysql> delimiter //
mysql> 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
-> deptno = OLD.deptno;
-> END;
-> //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> _
```

Essay

DATA BASES II

```
C:\> Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe"

mysql> SELECT * FROM emp;
+-----+-----+-----+-----+-----+
| EMPNO | NAME   | JOBNO | DEPTNO | COMM   |
+-----+-----+-----+-----+-----+
| 10    | CODD   | 100   | 50     | NULL   |
| 20    | NAVATHE | 200   | 50     | 450.00 |
| 30    | ELMASRI | 300   | 60     | NULL   |
| 40    | DATE   | 100   | 50     | NULL   |
| 7895  | CLARKE | NULL  | 70     | NULL   |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME      | LOC      | no_of_employees |
+-----+-----+-----+-----+
| 50     | SALES      | ATHENS   | 3               |
| 60     | ACCOUNTING | ATHENS   | 1               |
| 70     | OPERATIONS | VOLOS    | 2               |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DELETE
-> FROM
-> emp
-> WHERE
-> empno = 7895
-> ;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM emp;
+-----+-----+-----+-----+-----+
| EMPNO | NAME   | JOBNO | DEPTNO | COMM   |
+-----+-----+-----+-----+-----+
| 10    | CODD   | 100   | 50     | NULL   |
| 20    | NAVATHE | 200   | 50     | 450.00 |
| 30    | ELMASRI | 300   | 60     | NULL   |
| 40    | DATE   | 100   | 50     | NULL   |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME      | LOC      | no_of_employees |
+-----+-----+-----+-----+
| 50     | SALES      | ATHENS   | 3               |
| 60     | ACCOUNTING | ATHENS   | 1               |
| 70     | OPERATIONS | VOLOS    | 1               |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

Illustration

DATA BASES II

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql.exe
mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME      | LOC      | no_of_employees |
+-----+-----+-----+-----+
| 50     | SALES      | ATHENS   | 3               |
| 60     | ACCOUNTING | ATHENS   | 1               |
| 70     | OPERATIONS | VOLOS    | 1               |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> UPDATE
-> dept
-> SET
-> no_of_employees = IFNULL(no_of_employees, 0) - 1
-> WHERE
-> deptno = 70
-> ;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME      | LOC      | no_of_employees |
+-----+-----+-----+-----+
| 50     | SALES      | ATHENS   | 3               |
| 60     | ACCOUNTING | ATHENS   | 1               |
| 70     | OPERATIONS | VOLOS    | 0               |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

10.4. Illustration

The [trigger](#) 'emp _ delete ' is woken due to the AFTER condition DELETE ON emp . Specifically, the [statement](#) " DELETE FROM emp WHERE empno = 7895?" which is the reason for the trigger to wake up .

Because, a DELETE statement was given and the trigger is FOR EACH ROW , the pairs of global variables corresponding to the columns of the emp table have the following values:

- i) OLD . empno 7895
OLD . name ← ' CLARKE '
- ii) NEW.empno ←NULL
NEW .name ←NULL
NEW.jobno ←NULL
NEW .deptno ←NULL
NEW.comm ←NULL

DATA BASES II

Due to her command "UPDATE dept SET no_of_employees = IFNULL(no_of_employees , 0) - 1 WHERE deptno = OLD.deptno ?", the trigger removes from him table 'DEPT' the ' no_of_employees ' by 1 unit and specifically from her recording with " OLD.deptno " for " deptno ".

11. Create, test and explain the trigger that is awakened by the AFTER event UPDATE ON

11.1. Statement

Creation

```
delimiter //  
  
...  
  
CREATE TRIGGER  
  
emp_update1  
  
AFTER UPDATE ON  
  
    emp  
  
FOR EACH ROW  
  
BEGIN  
  
UPDATE  
  
    dept  
  
SET no_of_employees = IFNULL( no_of_employees , 0) - 1;  
  
WHERE  
  
    deptno = OLD.deptno ;  
  
END?  
  
//  
  
...  
  
delimiter ;  
  
...  
  
delimiter //
```

DATA BASES II

```
CREATE TRIGGER
emp_update2
AFTER UPDATE ON
    emp
FOR EACH ROW
BEGIN
UPDATE
    dept
SET no_of_employees = IFNULL( no_of_employees , 0) + 1;
WHERE
    deptno = NEW.deptno ;
END?
//
...
delimiter ;
```

Essay

```
SELECT * FROM emp ;
...
SELECT * FROM department ;
...
UPDATE
    emp
SET
    deptno = 70
WHERE
    name = 'Codd'
;
```

DATA BASES II

...

```
SELECT * FROM emp ;
```

...

```
SELECT * FROM dept .
```

Illustration

```
SELECT * FROM department ;
```

...

```
UPDATE
```

```
    dept
```

```
    SET
```

```
    no_of_employees = IFNULL( no_of_employees , 0) - 1
```

```
    WHERE
```

```
    deptno = 70
```

```
    ;
```

...

```
UPDATE
```

```
    dept
```

```
    SET
```

```
    no_of_employees = IFNULL( no_of_employees , 0) + 1
```

```
    WHERE
```

```
    deptno = 50
```

```
    ;
```

...

```
SELECT * FROM department ;
```

11.2. Result

DATA BASES II

Creation

...

Query OK, 0 rows affected (0.01 sec)

...

...

...

Query OK, 0 rows affected (0.01 sec)

...

Essay

EMPNO	NAME	JOBNO	DEPTNO	COMM
-------	------	-------	--------	------

10	CODD	100	50	NULL
20	NAVATHE	200	50	450.00
30	ELMASRI	300	60	NULL
40	DATE	100	50	NULL

4 rows in set (0.00 sec)

...

DEPTNO	DNAME	LOC	no_of_employees
--------	-------	-----	-----------------

50	SALES	ATHENS	3
60	ACCOUNTING	ATHENS	1

DATA BASES II

```
| 70 | OPERATIONS | VOLOS | 0 |
```

```
+-----+-----+-----+-----+
```

3 rows in set (0.00 sec)

...

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed : 1 Warnings: 0

...

```
+-----+-----+-----+-----+
```

```
| EMPNO | NAME | JOBNO | DEPTNO | COMM |
```

```
+-----+-----+-----+-----+
```

```
| 10 | CODD | 100 | 70 | NULL |
```

```
| 20 | NAVATHE | 200 | 50 | 450.00 |
```

```
| 30 | ELMASRI | 300 | 60 | NULL |
```

```
| 40 | DATE | 100 | 50 | NULL |
```

```
+-----+-----+-----+-----+
```

4 rows in set (0.00 sec)

...

```
+-----+-----+-----+-----+
```

```
| DEPTNO | DNAME | LOC | no_of_employees |
```

```
+-----+-----+-----+-----+
```

```
| 50 | SALES | ATHENS | 2 |
```

```
| 60 | ACCOUNTING | ATHENS | 1 |
```

```
| 70 | OPERATIONS | VOLOS | 1 |
```

```
+-----+-----+-----+-----+
```

3 rows in set (0.00 sec)

Illustration

```
+-----+-----+-----+-----+
```


DATA BASES II

```
| DEPTNO | DNAME | LOC | no_of_employees |
+-----+-----+-----+-----+
| 50 | SALES | ATHENS | 2 |
| 60 | ACCOUNTING | ATHENS | 1 |
| 70 | OPERATIONS | VOLOS | 1 |
+-----+-----+-----+-----+
```

3 rows in set (0.00 sec)

...

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed : 1 Warnings: 0

...

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed : 1 Warnings: 0

...

```
+-----+-----+-----+-----+
| DEPTNO | DNAME | LOC | no_of_employees |
+-----+-----+-----+-----+
| 50 | SALES | ATHENS | 3 |
| 60 | ACCOUNTING | ATHENS | 1 |
| 70 | OPERATIONS | VOLOS | 0 |
+-----+-----+-----+-----+
```

3 rows in set (0.00 sec)

11.3. Snapshot

Creation

DATA BASES II

```
C:\> Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --defa
mysql> CREATE TRIGGER
-> emp_update1
-> AFTER UPDATE ON
-> emp
-> FOR EACH ROW
-> BEGIN
-> UPDATE
-> dept
-> SET no_of_employees = IFNULL(no_of_employees, 0) - 1
-> WHERE
-> deptno = OLD.deptno;
-> END;
-> //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> _
```

...

```
C:\> Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --de
mysql> delimiter //
mysql> CREATE TRIGGER
-> emp_update2
-> AFTER UPDATE ON
-> emp
-> FOR EACH ROW
-> BEGIN
-> UPDATE
-> dept
-> SET no_of_employees = IFNULL(no_of_employees, 0) + 1
-> WHERE
-> deptno = NEW.deptno;
-> END;
-> //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> _
```

DATA BASES II

Essay

```
C:\> Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe"

mysql> SELECT * FROM emp;
+-----+-----+-----+-----+-----+
| EMPNO | NAME   | JOBNO | DEPTNO | COMM   |
+-----+-----+-----+-----+-----+
| 10    | CODD   | 100   | 50     | NULL   |
| 20    | NAVATHE | 200   | 50     | 450.00 |
| 30    | ELMASRI | 300   | 60     | NULL   |
| 40    | DATE   | 100   | 50     | NULL   |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME      | LOC      | no_of_employees |
+-----+-----+-----+-----+
| 50     | SALES      | ATHENS   | 3               |
| 60     | ACCOUNTING | ATHENS   | 1               |
| 70     | OPERATIONS | VOLOS    | 0               |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> UPDATE
-> emp
-> SET
-> deptno = 70
-> WHERE
-> name = 'CODD'
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM emp;
+-----+-----+-----+-----+-----+
| EMPNO | NAME   | JOBNO | DEPTNO | COMM   |
+-----+-----+-----+-----+-----+
| 10    | CODD   | 100   | 70     | NULL   |
| 20    | NAVATHE | 200   | 50     | 450.00 |
| 30    | ELMASRI | 300   | 60     | NULL   |
| 40    | DATE   | 100   | 50     | NULL   |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME      | LOC      | no_of_employees |
+-----+-----+-----+-----+
| 50     | SALES      | ATHENS   | 2               |
| 60     | ACCOUNTING | ATHENS   | 1               |
| 70     | OPERATIONS | VOLOS    | 1               |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

DATA BASES II

Illustration

```
C:\> Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe"

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME          | LOC      | no_of_employees |
+-----+-----+-----+-----+
|      50 | SALES          | ATHENS   |                2 |
|      60 | ACCOUNTING     | ATHENS   |                1 |
|      70 | OPERATIONS     | VOLOS    |                1 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> UPDATE
-> dept
-> SET
-> no_of_employees = IFNULL(no_of_employees, 0) - 1
-> WHERE
-> deptno = 70
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE
-> dept
-> SET
-> no_of_employees = IFNULL(no_of_employees, 0) + 1
-> WHERE
-> deptno = 50
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM dept;
+-----+-----+-----+-----+
| DEPTNO | DNAME          | LOC      | no_of_employees |
+-----+-----+-----+-----+
|      50 | SALES          | ATHENS   |                3 |
|      60 | ACCOUNTING     | ATHENS   |                1 |
|      70 | OPERATIONS     | VOLOS    |                0 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

11.4 . Illustration

emp _ update 1

The [trigger](#) “ emp _ update 1 ” wakes up due to the AFTER condition UPDATE ON emp . Specifically, the [statement](#) " UPDATE emp SET deptno = 70 WHERE name = ' CODD '?" which is the reason for the trigger to wake up .

Because UPDATE statement was given and trigger is FOR EACH ROW , the pairs of global variables corresponding to the columns of the emp table have the following values:

- i) OLD . empno 10
OLD . name \leftarrow ' CODD '
OLD . jobno \leftarrow 100
OLD . deptno \leftarrow 50
OLD . comm \leftarrow NULL
- ii) NEW . empno \leftarrow 10
NEW . name \leftarrow ' CODD '
NEW . jobno \leftarrow 100
NEW . deptno \leftarrow 70
NEW . comm \leftarrow NULL

Because of the “ UPDATE dept SET no _ of _ employees = IFNULL (no _ of _ employees , 0) – 1 WHERE deptno = OLD . deptno ?”, the trigger removes from the table " DEPT " the " no _ of _ employees " by 1 unit and specifically from the record with " OLD . deptno ' for ' deptno ' .

emp _ update 2

The [trigger](#) " emp _ update 2" is awakened by the AFTER condition UPDATE ON emp . Specifically, the [statement](#) " UPDATE emp SET deptno = 70 WHERE name = ' CODD '?" which is the reason for the trigger to wake up .

Because UPDATE statement was given and trigger is FOR EACH ROW , the pairs of global variables corresponding to the columns of the emp table have the following values:

- i) OLD . empno 10
OLD . name \leftarrow ' CODD '
OLD . jobno \leftarrow 100
OLD . deptno \leftarrow 50
OLD . comm \leftarrow NULL
- ii) NEW . empno \leftarrow 10
NEW . name \leftarrow ' CODD '
NEW . jobno \leftarrow 100

DATA BASES II

NEW . deptno ← 70
NEW . comm ← NULL

Because of the “ UPDATE dept SET no _ of _ employees = IFNULL (no _ of _ employees , 0) + 1 WHERE deptno = NEW . deptno ?”, the trigger adds from the table " DEPT " the " no _ of _ employees " by 1 unit and specifically from the record with " NEW . deptno ' for ' deptno '.

12. Presentation of information about Triggers

12.1. Statement

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

12.2. Result

Field	Type	Null	Key	Default	Extra
TRIGGER_CATALOG	varchar(64)	YES		NULL	
TRIGGER_SCHEMA	varchar(64)	YES		NULL	
TRIGGER_NAME	varchar(64)	NO		NULL	
EVENT_MANIPULATION	enum ('INSERT','UPDATE','DELETE')	NO		NULL	

DATA BASES II

```
| EVENT_OBJECT_CATALOG | varchar( 64 ) | YES | | NULL | |
| EVENT_OBJECT_SCHEMA | varchar( 64 ) | YES | | NULL | |
| EVENT_OBJECT_TABLE | varchar( 64 ) | YES | | NULL | |
| ACTION_ORDER | int unsigned | NO | | NULL | |
| ACTION_CONDITION | binary( 0 ) | YES | | NULL | |
| ACTION_STATEMENT | longtext | NO | | NULL | |
| ACTION_ORIENTATION | varchar( 3 ) | NO | | | |
| ACTION_TIMING | enum ( 'BEFORE', 'AFTER' ) | NO | | NULL | |
| ACTION_REFERENCE_OLD_TABLE | binary( 0 ) | YES | | NULL | |
| ACTION_REFERENCE_NEW_TABLE | binary( 0 ) | YES | | NULL | |
| ACTION_REFERENCE_OLD_ROW | varchar( 3 ) | NO | | | |
| ACTION_REFERENCE_NEW_ROW | varchar( 3 ) | NO | | | |
| CREATED | timestamp( 2 ) | NO | | NULL | |
| SQL_MODE | set(
'REAL_AS_FLOAT','PIPES_AS_CONCAT','ANSI_QUOTES','IGNORE_SPACE',..., 'TIME_TRUNCAT
E_FRACTIONAL') | NO | | NULL | |
| DEFINER | varchar( 288 ) | NO | | NULL | |
| CHARACTER_SET_CLIENT | varchar( 64 ) | NO | | NULL | |
| COLLATION_CONNECTION | varchar( 64 ) | NO | | NULL | |
| DATABASE_COLLATION | varchar( 64 ) | NO | | NULL | |
22 rows in set (0.01 sec)

...

+-----+-----+-----+
| TRIGGER_NAME | EVENT_MANIPULATION | TRIGGER_SCHEMA |
+-----+-----+-----+
| dept_update | UPDATE | personnel |
| emp_delete | DELETE | personnel |
| emp_insert | INSERT | personnel |
| emp_update1 | UPDATE | personnel |
```

DATA BASES II

```
| emp_ update2 | UPDATE | personnel |
```

```
+-----+-----+-----+ +
```

```
5 rows in set (0.00 sec)
```

12.3 . Snapshot

DATA BASES II

Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" --default-character-set=utf8mb4 -u root -p

```
mysql> DESCRIBE Information_schema.TRIGGERS;
```

Field	Type	Null	Key	Default	Extra
TRIGGER_CATALOG	varchar(64)	YES		NULL	
TRIGGER_SCHEMA	varchar(64)	YES		NULL	
TRIGGER_NAME	varchar(64)	NO		NULL	
EVENT_MANIPULATION	enum('INSERT','UPDATE','DELETE')	NO		NULL	
EVENT_OBJECT_CATALOG	varchar(64)	YES		NULL	
EVENT_OBJECT_SCHEMA	varchar(64)	YES		NULL	
EVENT_OBJECT_TABLE	varchar(64)	YES		NULL	
ACTION_ORDER	int unsigned	NO		NULL	
ACTION_CONDITION	binary(0)	YES		NULL	
ACTION_STATEMENT	longtext	NO		NULL	
ACTION_ORIENTATION	varchar(3)	NO			
ACTION_TIMING	enum('BEFORE','AFTER')	NO		NULL	
ACTION_REFERENCE_OLD_TABLE	binary(0)	YES		NULL	
ACTION_REFERENCE_NEW_TABLE	binary(0)	YES		NULL	
ACTION_REFERENCE_OLD_ROW	varchar(3)	NO			
ACTION_REFERENCE_NEW_ROW	varchar(3)	NO			
CREATED	timestamp(2)	NO		NULL	
SQL_MODE: set('REAL_AS_FLOAT','PIPES_AS_CONCAT','ANSI_QUOTES','IGNORE_SPACE','NOT_USED','ONLY_FULL_GROUP_BY','NO_UNSIGNED_SUBTRACTION','NO_DIR_IN_CREATE','NOT_USED_9','NOT_USED_10','NOT_USED_11','NOT_USED_12','NOT_USED_13','NOT_USED_14','NOT_USED_15','NOT_USED_16','NOT_USED_17','NOT_USED_18','ANSI','NO_AUTO_VALUE_ON_ZERO','NO_BACKSLASH_ESCAPES','STRICT_TRANS_TABLES','STRICT_ALL_TABLES','NO_ZERO_IN_DATE','NO_ZERO_DATE','ALLOW_INVALID_DATES','ERROR_FO...					
DEFINER	varchar(288)	NO		NULL	
CHARACTER_SET_CLIENT	varchar(64)	NO		NULL	
COLLATION_CONNECTION	varchar(64)	NO		NULL	
DATABASE_COLLATION	varchar(64)	NO		NULL	

22 rows in set (0.00 sec)

```
mysql>
```

...

Command Prompt - "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe"

```
mysql> SELECT
-> TRIGGER_NAME, EVENT_MANIPULATION, TRIGGER_SCHEMA
-> FROM
-> INFORMATION_SCHEMA.TRIGGERS
-> WHERE
-> TRIGGER_SCHEMA = 'personnel'
-> ORDER BY
-> TRIGGER_NAME
-> ;
```

TRIGGER_NAME	EVENT_MANIPULATION	TRIGGER_SCHEMA
dept_update	UPDATE	personnel
emp_delete	DELETE	personnel
emp_insert	INSERT	personnel
emp_update1	UPDATE	personnel
emp_update2	UPDATE	personnel

5 rows in set (0.00 sec)

```
mysql>
```

13 . Deletion of Triggers

13.1. Statement

```
DROP TRIGGER dept_update ;  
  
...  
  
DROP TRIGGER emp_insert ;  
  
...  
  
DROP TRIGGER emp_delete ;  
  
...  
  
DROP TRIGGER emp_update1;  
  
...  
  
DROP TRIGGER emp_update2;  
  
...  
  
SELECT  
TRIGGER_NAME, EVENT_MANIPULATION, TRIGGER_SCHEMA  
FROM  
INFORMATION_SCHEMA.TRIGGERS  
WHERE  
TRIGGER_SCHEMA = 'personnel'  
ORDER BY  
TRIGGER_NAME  
;
```

13.2. Result

```
Query OK, 0 rows affected (0.01 sec)  
  
...  
  
Query OK, 0 rows affected (0.01 sec)  
  
...
```

DATA BASES II

Query OK, 0 rows affected (0.01 sec)

...

Query OK, 0 rows affected (0.01 sec)

...

Query OK, 0 rows affected (0.01 sec)

...

Empty set (0.00 sec)

13.3. Snapshot

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql.exe
mysql> DROP TRIGGER dept_update;
Query OK, 0 rows affected (0.01 sec)

mysql> DROP TRIGGER emp_insert;
Query OK, 0 rows affected (0.01 sec)

mysql> DROP TRIGGER emp_delete;
Query OK, 0 rows affected (0.01 sec)

mysql> DROP TRIGGER emp_update1;
Query OK, 0 rows affected (0.01 sec)

mysql> DROP TRIGGER emp_update2;
Query OK, 0 rows affected (0.01 sec)

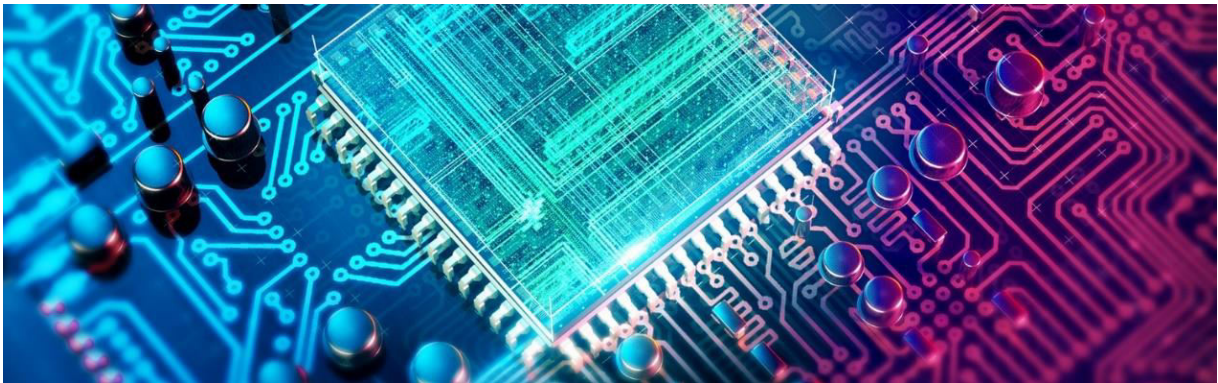
mysql> SELECT
  -> TRIGGER_NAME, EVENT_MANIPULATION, TRIGGER_SCHEMA
  -> FROM
  -> INFORMATION_SCHEMA.TRIGGERS
  -> WHERE
  -> TRIGGER_SCHEMA = 'personnel'
  -> ORDER BY
  -> TRIGGER_NAME
  -> ;
Empty set (0.00 sec)

mysql> _
```

DATA BASES II



Thank you for your attention.



DATA BASES II

