

ADVANCED DBMS LAB

TRIGGER

SWATHY KRISHNA P R
RMCA B
ROLL NO: 31

AIM

Create a Trigger for employee table it will update another table salary while updating values

OBJECTIVE

To develop and execute a Trigger for After update/Delete/Insert operations on a table

PROCEDURE

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

PROGRAM

sql>

```
CREATE TABLE `employee` (  
  `emp_id` int(11) NOT NULL,  
  `emp_name` varchar(45) DEFAULT NULL,  
  `dob` date DEFAULT NULL,  
  `address` varchar(45) DEFAULT NULL,  
  `designation` varchar(45) DEFAULT NULL,  
  `mobile_no` int(11) DEFAULT NULL,  
  `dept_no` int(11) DEFAULT NULL,  
  `salary` int(11) DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

Sql>

```
CREATE TABLE `salary` (  
  `employee_id` int(11) NOT NULL,  
  `old_sal` int(11) DEFAULT NULL,  
  `new_sal` int(11) DEFAULT NULL,  
  `rev_date` date DEFAULT NULL,  
  PRIMARY KEY (`employee_id`)  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER  
`db1`.`personal_updatations_AFTER_UPDATE_1`  
AFTER UPDATE ON `employee`  
FOR EACH ROW  
BEGIN  
  if(new.salary != old.salary)  
  then  
    INSERT INTO salary (employee_id,old_sal,new_sal,rev_date) values  
    (new.emp_id,old.salary,new.salary,sysdate());  
  END if;
```

end;

sql>

```
update employee set salary=234569 where emp_id=2;  
select * from salary;
```

The screenshot displays the MySQL Workbench interface. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' tree with 'employee db' selected, containing tables like 'employee', 'salary', 'student_table', and 'test'. The central 'Query Editor' shows a query with two statements: an update and a select. The 'Result Grid' below the query shows the results of the select statement, with one row for employee_id 2. The bottom 'Output' pane shows the execution log for the query, indicating that the update and select statements were executed successfully. The Windows taskbar at the bottom shows the system clock as 12:34 on 30-08-2021.

MySQL Workbench

connection1 - Warning - not s...

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

employee db

Tables

employee

salary

Views

Stored Procedures

Functions

phpmyadmin

student

student_table

test

Query 1 SQL File 3* student_table employee - Table employee - Table SQL File 5*

1 • update employee set salary=234569 where emp_id=2;
2 select * from salary;

Limit to 1000 rows

Result Grid

employee_id	old_sal	new_sal	rev_date
2	67000	234569	2021-08-30

Administration Schemas

Information

No object selected

salary 4 x

Read Only Context Help Snippets

Output

#	Time	Action	Message	Duration / Fetch
7	12:27:15	update employee set designation='ck' where emp_id=1	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
8	12:27:15	select * from promotions LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
9	12:31:15	update employee set designation='ck' where emp_id=2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.094 sec
10	12:31:15	select * from promotions LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
11	12:34:15	update employee set salary=234569 where emp_id=2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.062 sec
12	12:34:15	select * from salary LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

29°C

ENG

12:34

30-08-2021

AIM

Create a Trigger for employe table it will update another table personal_updates while updating values

OBJECTIVE

To develop and execute a Trigger for Before and After update/Delete/Insert operations on a table

PROCEDURE

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

PROGRAM

sql>

```
CREATE TABLE `employe` (  
  `emp_id` int(11) NOT NULL,  
  `emp_name` varchar(45) DEFAULT NULL,  
  `dob` date DEFAULT NULL,  
  `address` varchar(45) DEFAULT NULL,  
  `designation` varchar(45) DEFAULT NULL,  
  `mobile_no` int(11) DEFAULT NULL,  
  `dept_no` int(11) DEFAULT NULL,  
  `salary` int(11) DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

Sql>

```
CREATE TABLE `personal_updates` (  
  `emp_id` int(11) NOT NULL,  
  `old_phoneno` int(11) DEFAULT NULL,  
  `new_phoneno` int(11) DEFAULT NULL,  
  `rev_date` date DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER `db1`.`personal_updates_AFTER_UPDATE`  
AFTER UPDATE ON `employe`  
FOR EACH ROW  
BEGIN  
  if(new.mobile_no != old.mobile_no)  
  then  
    INSERT INTO personal_updates (emp_id,old_phoneno,new_phoneno,rev_date) values  
    (new.emp_id,new.mobile_no,old.mobile_no,sysdate());  
  END if;
```

end;

sql>

update employe set mobile_no=34566 where emp_id=2 ;

select * from personal_updatations;

The screenshot displays the MySQL Workbench interface. The top toolbar includes options like File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' panel with a tree view of the database structure, including 'employee db', 'Tables', 'Views', 'Stored Procedures', 'Functions', 'phpmyadmin', 'student', 'student_table', and 'test'. The main query editor shows two queries: 1. 'update employe set mobile_no=34566 where emp_id=2 ;' and 2. 'select * from personal_updatations;'. The 'Result Grid' shows the results of the second query, displaying a table with columns 'emp_id', 'old_phoneno', 'new_phoneno', and 'rev_date'. The table contains one row with values 2, 34566, 908756442, and 2021-08-30. The bottom panel shows the 'Output' tab with a table of execution logs, including timestamps, actions, messages, and durations.

emp_id	old_phoneno	new_phoneno	rev_date
2	34566	908756442	2021-08-30

#	Time	Action	Message	Duration / Fetch
9	12:31:15	update employe set designation='ckl' where emp_id=2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.094 sec
10	12:31:15	select * from promotions LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
11	12:34:15	update employe set salary=234569 where emp_id=2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.062 sec
12	12:34:15	select * from salary LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
13	12:35:22	update employe set mobile_no=34566 where emp_id=2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.046 sec
14	12:35:22	select * from personal_updatations LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

AIM

Create a Trigger for employe table it will update another table promotions while updating values

OBJECTIVE

To develop and execute a Trigger for Before and After update/Delete/Insert operations on a table

PROCEDURE

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

PROGRAM

sql>

```
CREATE TABLE `employe` (  
  `emp_id` int(11) NOT NULL,  
  `emp_name` varchar(45) DEFAULT NULL,  
  `dob` date DEFAULT NULL,  
  `address` varchar(45) DEFAULT NULL,  
  `designation` varchar(45) DEFAULT NULL,  
  `mobile_no` int(11) DEFAULT NULL,  
  `dept_no` int(11) DEFAULT NULL,  
  `salary` int(11) DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

Sql>

```
CREATE TABLE `personal_updatations` (  
  `emp_id` int(11) NOT NULL,  
  `old_phoneno` int(11) DEFAULT NULL,  
  `new_phoneno` int(11) DEFAULT NULL,  
  `rev_date` date DEFAULT NULL,  
  PRIMARY KEY (`emp_id`)  
);
```

sql>

```
CREATE DEFINER=`root`@`localhost` TRIGGER `db1`.`employe_AFTER_UPDATE_1`  
AFTER UPDATE ON `employe`  
FOR EACH ROW  
BEGIN  
  if(new.designation != old.designation)  
  then  
    INSERT INTO promotions (emp_id,old_designation,new_designation,rev_date) values  
    (new.emp_id,new.designation,old.designation,sysdate());  
  END if;
```

end;

sql>

update employe set designation='clk' where emp_id=2;

select * from promotions;

The screenshot displays the MySQL Workbench interface. The top toolbar includes icons for file operations, query execution, and database management. The left sidebar shows the 'SCHEMAS' panel with a tree view of the database structure, including tables like 'employee' and 'promotions'. The main query editor window shows a SQL script with two statements: an update and a select. The 'Result Grid' tab is active, displaying the results of the 'select * from promotions;' query. The results are shown in a table with columns 'emp_id', 'old_designation', 'new_designation', and 'rev_date'. The output shows one row with emp_id 2, old_designation 'ck', new_designation 'HR', and rev_date '2021-08-30'. The bottom status bar shows the current session information, including the time and date.

MySQL Workbench

connection1 - Warning - not s...

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employee db

Tables

employee

salary

Views

Stored Procedures

Functions

phpmyadmin

student

student_table

test

Query 1 SQL File 3* student_table employee - Table employee - Table SQL File 5*

1 • update employe set designation='clk' where emp_id=2;

2 select * from promotions;

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows: Export: Wrap Cell Contents

emp_id	old_designation	new_designation	rev_date
2	ck	HR	2021-08-30

Administration Schemas

Information

No object selected

promotions 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	12:26:16	update employe set salary=234569 where emp_id=1	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
6	12:26:16	select * from salary LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
7	12:27:15	update employe set designation='clk' where emp_id=1	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
8	12:27:15	select * from promotions LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
9	12:31:15	update employe set designation='clk' where emp_id=2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.094 sec
10	12:31:15	select * from promotions LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

File Explorer

29°C

12:32

30-08-2021