

Aim: Demonstrate the use of PL/SQL, conditional statements and control structures, Cursors, Triggers.

Objectives:

- To use PL/SQL triggers.
- To understand the different types of triggers.

Tools Used: MySQL Workbench

Concept:

A trigger is called a special procedure because it cannot be called directly like a stored procedure. The key distinction between the trigger and procedure is that a trigger is called automatically when a data modification event occurs against a table. These triggers have various types which follow:

- **Before insert triggers** are triggered prior to the insertion of data into a table.
- **After insert triggers** activate after data has been successfully inserted into a table.
- **Before update triggers** are invoked before any updates are made to the data in a table.
- **After update triggers** are executed once data in the table has been updated.
- **Before delete triggers** are activated before any data is removed from the table.
- **After delete triggers** are invoked after data has been successfully deleted from the table.

Syntax:

```
DELIMITER //
CREATE TRIGGER trigger_name
(AFTER | BEFORE)
(INSERT UPDATE | DELETE)
ON table_name FOR EACH ROW
BEGIN
--variable declarations
--trigger code
END;
```

Lab Assignment on Triggers

Problem Statement:

1) Question on Before Insert Trigger:

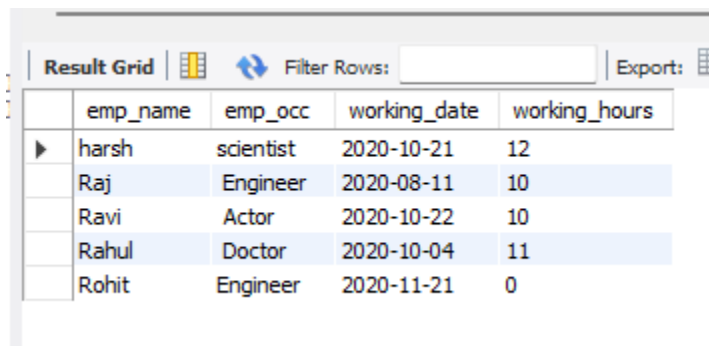
Write a trigger which ensures that if the user enters a negative value in Working_hours the value is set to 0.

Trigger :

```
CREATE DEFINER='root'@'localhost' TRIGGER `employee_BEFORE_INSERT`  
BEFORE INSERT ON `employee` FOR EACH ROW BEGIN  
IF new.working_hours<0 then set new.working_hours=0;  
end if;  
END
```

Sql Query :

```
insert into employee values('Rohit','Engineer', '2020-11-21','-15');  
select * from employee;
```



	emp_name	emp_occ	working_date	working_hours
▶	harsh	scientist	2020-10-21	12
	Raj	Engineer	2020-08-11	10
	Ravi	Actor	2020-10-22	10
	Rahul	Doctor	2020-10-04	11
	Rohit	Engineer	2020-11-21	0

Problem Statement:

2) Question on After Insert Trigger:

Create a table emp_audit(name,audit_description) Create a trigger to make sure If any employee information is inserted in emp table when trigger is inserting the row in emp_audit table automatically.

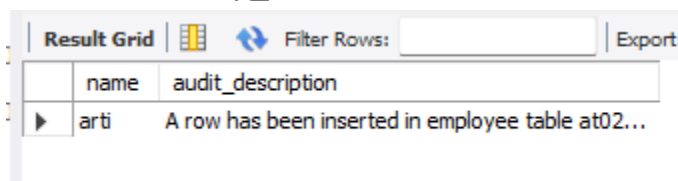
Trigger:

```
CREATE DEFINER='root'@'localhost' TRIGGER `emp_AFTER_INSERT` AFTER  
INSERT ON `emp` FOR EACH ROW BEGIN  
insert into emp_audit values (new.emp_name, concat('A row has been inserted in  
employee table at',
```

```
date_format(now(), '%d-%m-%y %h:%i:%s %p')));  
END
```

Sql Query :

```
insert into emp values  
( 'arti', 'professor', '2020-08-18', '10');  
select * from emp_audit;
```



	name	audit_description
▶	arti	A row has been inserted in employee table at 02...

Problem Statement :**3) Question on Before Update Trigger:**

Create a trigger if a new working date is greater than today's date to raise an error message.

Trigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `emp_BEFORE_UPDATE` BEFORE  
UPDATE ON `emp` FOR EACH ROW BEGIN  
declare ErrorMessage varchar(300);  
set ErrorMessage =concat("Working date is" , new.working_date, "cannot be greater  
than" ,old.working_date);  
if new.working_date> old.working_date then signal sqlstate '45000' set  
message_text=ErrorMessage;  
end if;  
END
```

Sql query :

```
update emp  
set working_date = "2022-12-28" where emp_name = 'arti';
```

Error Code: 1644. Working date is2022-12-28cannot be greater than2020-08-18

Problem Statement :**4) Question on After Update Trigger:**

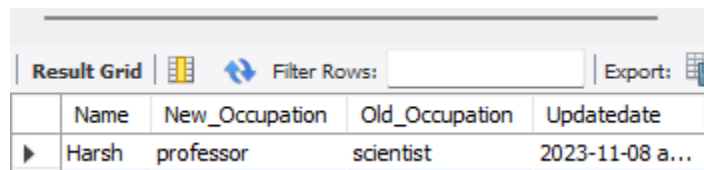
Create a table EmpChanges(Name, New Occupation, Old Occupation, Updatedat as shown in following output. Create a trigger that will keep history of changes in the EmpChange table when you change data in the Emp table.

Trigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `emps_AFTER_UPDATE` AFTER
UPDATE ON `emps` FOR EACH ROW BEGIN
if old.occupation<>new.occupation
then insert into empChanges values
(old.name,new.occupation,old.occupation,concat(current_date(),' at
',current_time()));
end if;
END
```

Sql query :

```
update emps set occupation="professor" where Name="harsh";
select * from EmpChanges;
```



	Name	New_Occupation	Old_Occupation	Updatedate
▶	Harsh	professor	scientist	2023-11-08 a...

Problem Statement :**5) Question on Before Delete Trigger:** Create a table Emp_archeives

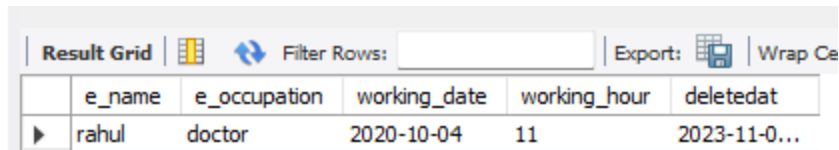
(Name,Occupation,Working_date,WorkingHours, Deletedat) Create trigger to ensure before removing data from Emp table, the record should be entered in Emp_archieves table.

Trigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `emps_BEFORE_DELETE` BEFORE
DELETE ON `emps` FOR EACH ROW BEGIN
insert into emp_archives
values(old.name,old.occupation,old.working_date,old.working_hours,concat(current_date(),'
at ',current_time()));
END
```

Sql query :

```
delete from emps where name='Rahul';
select * from emp_archives;
```



The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' search bar and an 'Export' button. The table has five columns: e_name, e_occupation, working_date, working_hour, and deletedat. One row is visible for an employee named 'rahul' who is a 'doctor', with a working date of '2020-10-04', 11 working hours, and a deletion date of '2023-11-0...'.

	e_name	e_occupation	working_date	working_hour	deletedat
▶	rahul	doctor	2020-10-04	11	2023-11-0...

Problem Statement :

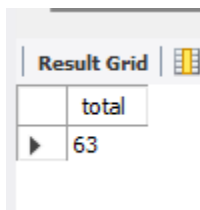
6) Question on After Delete Trigger: Consider you have two tables Emp Table(Original Table) and Total_working_hours_table which looks like

Triggers

```
CREATE DEFINER=`root`@`localhost` TRIGGER `emps_AFTER_DELETE` AFTER DELETE
ON `emps` FOR EACH ROW BEGIN
update total_working_hours
set total=total-old.working_hours;
END
```

Sql query :

```
delete from emps where name='Raj';
select * from total_working_hours;
```



The screenshot shows a database interface with a 'Result Grid' tab. The table has one column named 'total' and one row with the value '63'.

	total
▶	63

Observation :

In this practical , I understand to use different types of triggers which allows us to specify SQL actions that should be executed automatically when a specific event occurs in the database.