BHARATIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Academic Year: 2023– 2024

Class: F.Y.MCA **Course Code: MC502** **Semester: I**

Course: Database Management System

Name of the Student: Abhijit Namdeo Shirke

UCID: 2023510054

Class: FY MCA

Signature of the Student: Abhijit

BHARATIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Academic Year: 2023– 2024

Class: F.Y.MCA Course Code: MC502 Semester: I

Course: Database Management System

Experiment No.5

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

Tools Required: MySQL Workbench

Prior Concept: Database, Concept of Relational Database, SQL.

New Concept: Building blocks of PL/SQL, Conditional statement and Control structures, Cursors, Triggers.

Concept: Basic building block of PL/SQL:

```
DECLARE
<declarations section>
BEGIN
<executable command(s)>
EXCEPTION
<exception handling>
END;
```

PL/SQL Conditional Statements Syntax:

a) IF-THEN-ELSIF Syntax:

```
IF(boolean_expression 1)THEN
S1; -- Executes when the boolean expression 1 is true
ELSIF( boolean_expression 2) THEN
S2; -- Executes when the boolean expression 2 is true
ELSIF( boolean_expression 3) THEN
S3; -- Executes when the boolean expression 3 is true
ELSE
S4; -- executes when the none of the above condition is true
END IF;
```

b) CASE Syntax:

```
CASE selector
WHEN 'value1' THEN S1;
WHEN 'value2' THEN S2;
WHEN 'value3' THEN S3;
```

```
...
ELSE Sn; -- default case
END CASE;
```

PL/SQL Control Structure Syntax:

a) BASIC LOOP:

```
LOOP
        Sequence of statements;
END LOOP;
```

b) WHILE LOOP:

```
WHILE condition LOOP
        sequence_of_statements
END LOOP;
```

c) FOR LOOP:

```
FOR counter IN initial_value .. final_value LOOP
        sequence_of_statements;
END LOOP;
```

Explicit Cursor

There are four steps in using an Explicit Cursor.

- DECLARE the cursor in the declaration section.
- OPEN the cursor in the Execution Section.
- FETCH the data from cursor into PL/SQL variables or records in the Execution Section.
- CLOSE the cursor in the Execution Section before you end the PL/SQL Block.

Trigger Syntax:

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR]| UPDATE [OR]| DELETE}
[OF col_name]
ON table_name [FOR EACH
ROW]
WHEN (condition)
DECLARE
```

```
Declaration-statements BEGIN
Executable-statements
EXCEPTION
Exception-handling-statements END;
```

Lab Exercise:

Demonstrate all these syntax with the help on the problem statement given by instructor.

BHARATIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Academic Year: 2023– 2024

Class: F.Y.MCA **Course Code: MC502** **Semester: I**

Course: Database Management System

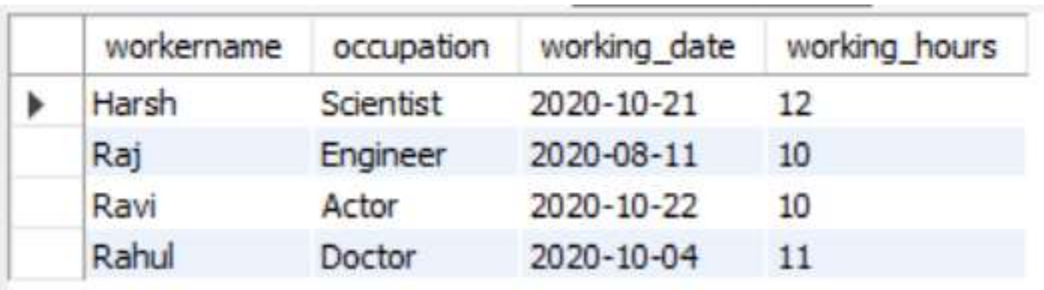
Lab Assignment on Triggers

Scenario:
Create following emp table and insert the specified values in the database using MySQL.

Query

```
create database trigger_54;
use trigger_54;
create table emp (workername varchar(20), occupation varchar(20),
                    working_date date, working_hours int);

insert into emp values('Harsh','Scientist','2020-10-21',12),
                              ('Raj','Engineer','2020-08-11',10),
                              ('Ravi','Actor','2020-10-22',10),
                              ('Rahul','Doctor','2020-10-04',11);
```

Ouput:



	workername	occupation	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

1. Question on Before Insert Trigger:
   a. Write a trigger which ensures that if user enters negative value in Working_hours the value is set to 0

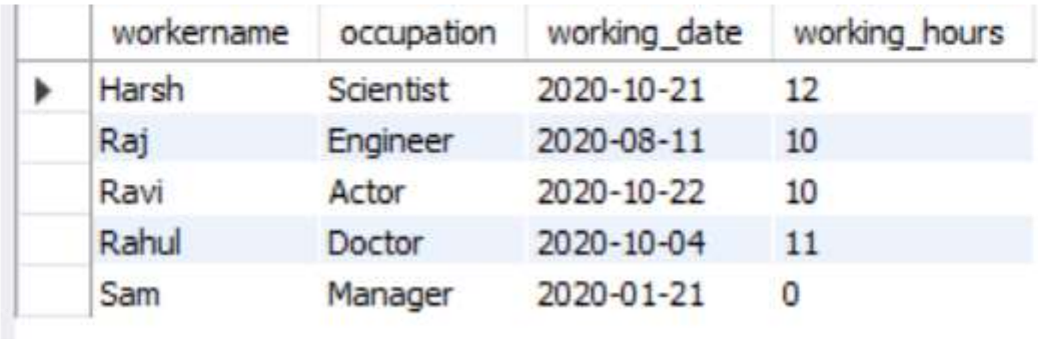
Tigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `emp_BEFORE_INSERT`
BEFORE INSERT ON `emp` FOR EACH ROW BEGIN
IF new.working_hours < 0
    then set new.working_hours=0;
END IF;
END
```

Query:

```
insert into emp values ('Sam','Manager','2020-01-21',-12);
```

OUTPUT:



	workername	occupation	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
	Sam	Manager	2020-01-21	0

2. Question on After Insert Trigger:
   a. Create a table emp_audit(name,audit_description)
   b. Create a trigger to make sure If any employee information is inserted in emp table then trigger is inserting the row in emp_audit table automatically.
   c. Output should look like:

Name	Audit_description
Arti	A row has been inserted in emp table at 2020-01-23 at 11:23:45 PM

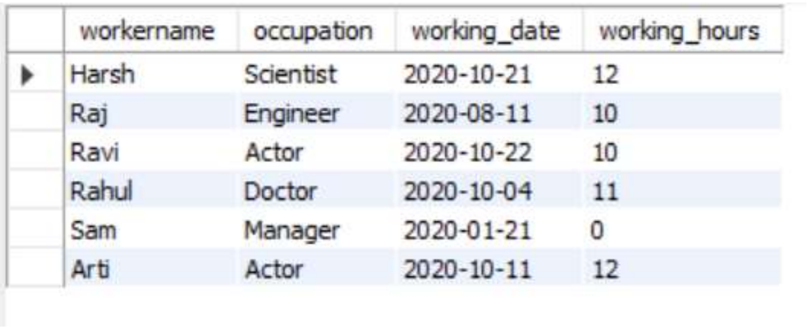
Tigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `people_AFTER_INSERT`
AFTER INSERT ON `emp` FOR EACH ROW BEGIN
insert into emp_audit values
    (new.workername,concat('A row has been inserted in people table at
    ',DATE_FORMAt(NOW(),'%d-%m-%Y-%h:%i:%s %p')));
END
```

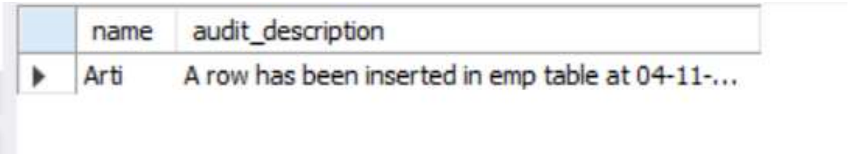
Query:

```
Create table emp_audit(name varchar(30), audit_description varchar(100));

insert into emp values ('Arti','Actor','2020-10-11',12);
```

OUTPUT:



	workername	occupation	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
	Sam	Manager	2020-01-21	0
	Arti	Actor	2020-10-11	12



	name	audit_description
▶	Arti	A row has been inserted in emp table at 04-11-...

3. Question on Before Update Trigger:
    a. Create a trigger if a new working date is greater than today’s date to raise an error message.

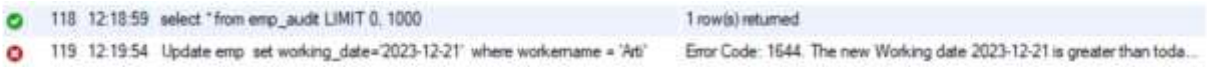
Tigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `people_BEFORE_UPDATE`
BEFORE UPDATE ON `emp` FOR EACH ROW BEGIN
declare today_date date;
declare errormessage varchar(250);
    SET today_date = CURDATE();
    SET errormessage = CONCAt("The new Working date ",new.working_date,
                                "is greater than today date ",today_date);
IF new.working_date > today_date
    then
    SIGNAL SQLSTATE '45000'
    SET message_text = errormessage;
END IF;
END
```

Query:

```
Update emp
 set working_date='2023-12-21'
 where workername = 'Arti';
```

OUTPUT:



```
118  12:18:59  select * from emp_audit LIMIT 0, 1000                              1 row(s) returned
119  12:19:54  Update emp  set working_date='2023-12-21'  where workername = 'Arti'    Error Code: 1644. The new Working date 2023-12-21 is greater than toda...
```

4. Question on After Update Trigger:
    a. Create a table EmpChanges(Name, New Occupation, Old Occupation, Updatedate as shown in following output.
    b. Create a trigger that will keep history of changes in the EmpChange table when you change data in Emp table.
    c. Output should look like

Name	New Occupation	Old Occupation	Updatedate
Harsh	Professor	Scientist	2020-01-23 at 11:23:45 PM

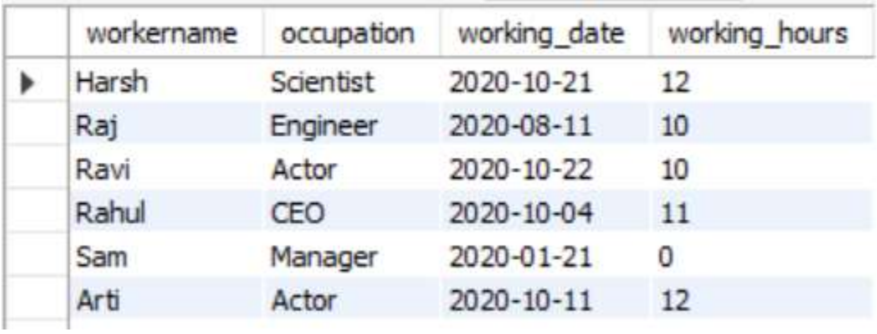
Tigger:

```
        CREATE DEFINER=`root`@`localhost` TRIGGER `emp_AFTER_UPDATE`
AFTER UPDATE ON `emp` FOR EACH ROW BEGIN

        insert into empchanges values (new.workername, new.occupation,
old.occupation,  concat(DATE_FORMAt(NOW(),'%d-%m-%Y at %h:%i:%s %p')) );

        END
```

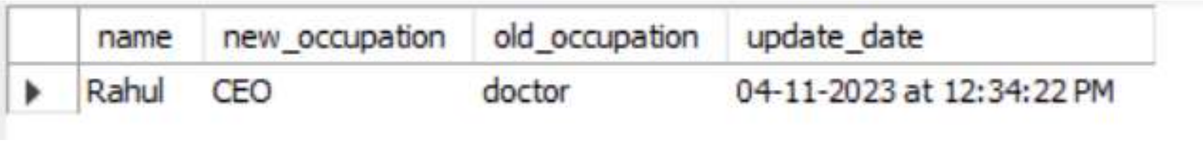
Query:

```
    create table empchanges(name varchar(20) , new_occupation varchar(20),
            old_occupation varchar(20), update_date varchar(50));

    Update emp
     set occupation='CEO'
     where workername = 'Rahul';
```

OUTPUT:



workername	occupation	working_date	working_hours
Harsh	Scientist	2020-10-21	12
Raj	Engineer	2020-08-11	10
Ravi	Actor	2020-10-22	10
Rahul	CEO	2020-10-04	11
Sam	Manager	2020-01-21	0
Arti	Actor	2020-10-11	12



name	new_occupation	old_occupation	update_date
Rahul	CEO	doctor	04-11-2023 at 12:34:22 PM

5. Question on Before Delete Trigger:
   a. Create a table Emp_archeives (Name, Occupation, Working_date, WorkingHours, Deletedate)
   b. Create trigger to ensure before removing data from Emp table, the record should be entered in Emp_archieves table.

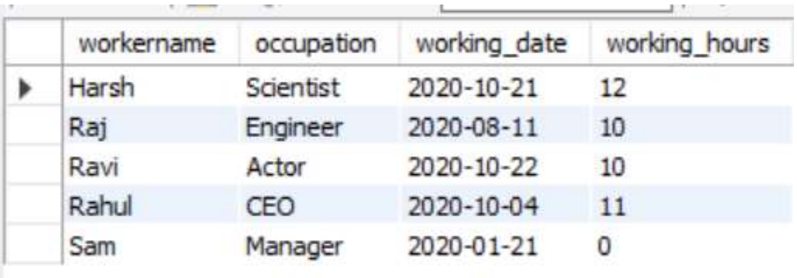
Tigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `emp_BEFORE_DELETE`
BEFORE DELETE ON `emp` FOR EACH ROW BEGIN
insert into emp_archives values
(old.workername,old.occupation,old.working_date,old.working_hours,concat(DA
TE_FORMAt(NOW(),'%d-%m-%Y at %h:%i:%s %p')) );
END
```

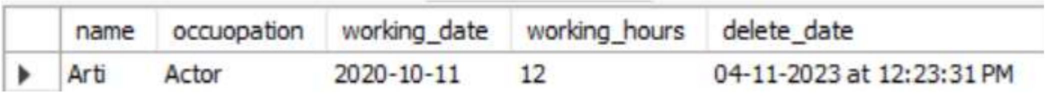
Query:

```
create table emp_archives( name varchar(20), occuopation varchar(20),
            working_date date, working_hours varchar(20),
            delete_date varchar(50));

delete from emp
where workername = 'Arti';
```

OUTPUT:



	workername	occupation	working_date	working_hours
▶	Harsh	Scientist	2020-10-21	12
	Raj	Engineer	2020-08-11	10
	Ravi	Actor	2020-10-22	10
	Rahul	CEO	2020-10-04	11
	Sam	Manager	2020-01-21	0



	name	occuopation	working_date	working_hours	delete_date
▶	Arti	Actor	2020-10-11	12	04-11-2023 at 12:23:31 PM

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

Total
43

   b. Create a trigger that changes the Total of above table when Emp leaves the company.

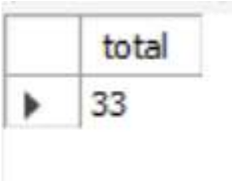
Tigger:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `emp_AFTER_DELETE`
AFTER DELETE ON `emp` FOR EACH ROW BEGIN
insert into total_working_hours_table values
 ( (select sum(working_hours) from emp));
END
```

Query:

```
create table total_working_hours_table (total int);

delete from emp
where workername='Raj';
```

OUTPUT:



	total
▶	33

Observation:

1. Triggers in MySQL are database objects that automatically execute a set of SQL statements when specific events occur within a table, such as INSERT, UPDATE, or DELETE operations.
2. MySQL Workbench's visual design capabilities make it easy to create and modify triggers, allowing to specify when the trigger should fire and what actions it should take in response to the specified events.
3. Triggers are commonly used to implement auditing, maintain history tables