Ex.No:01

DDL COMMANDS

Aim:

To create and work with DDL commands

Procedure:

Step1: Open Run SQL on Command line and connect to SQL .

Step 2: Then work with database using SQL queries

1.**CREATE:**

Syntax: CREATE TABLE tablename(column_name1 datatype,…,column_name_n datatype);

Code:

Create table programming(serial number(10), Name varchar2(100),joining date,shift number(1));

OUTPUT:

```
SQL> create table programming (Serial number (10), Name varchar2(100), Language varchar2(100), Joining date, Shift number (1));

Table created.

SQL> desc programming;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SERIAL                                             NUMBER(10)
 NAME                                               VARCHAR2(100)
 LANGUAGE                                           VARCHAR2(100)
 JOINING                                            DATE
 SHIFT                                              NUMBER(1)
```

2.Alter:

i) Alter table- Add Column:

Syntax: ALTER TABLE tablename ADD column_name datatype;

Code: alter table programming add language_type varchar2(15);

ii) Alter table-Drop Column:

Syntax: ALTER TABLE tablename DROP column_name;

Code: alter table programming drop language_type;

iii) Alter table-Rename Column:

Syntax: ALTER TABLE table_name RENAME COLUMN oldname TO newname;

Code: alter table programming rename column joining to date_of_joining;

iv) Alter table-Modify Datatype:

Syntax: ALTER TABLE table_name MODIFY column_name datatype;

Code: alter table programming serial varchar2(10);

OUTPUT:

```
SQL> desc programming;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SERIAL                                             VARCHAR2(10)
 NAME                                               VARCHAR2(100)
 LANGUAGE                                           VARCHAR2(100)
 DATE_OF_JOINING                                    DATE
 SHIFT                                              NUMBER(1)

SQL> alter table programming add language_type varchar2(15);

Table altered.

SQL> desc programming;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SERIAL                                             VARCHAR2(10)
 NAME                                               VARCHAR2(100)
 LANGUAGE                                           VARCHAR2(100)
 DATE_OF_JOINING                                    DATE
 SHIFT                                              NUMBER(1)
 LANGUAGE_TYPE                                      VARCHAR2(15)
```

```
SQL> desc programming;
 Name                                        Null?    Type
 ----------------------------------------- -------- ----------------------------
 SERIAL                                               NUMBER(10)
 NAME                                                 VARCHAR2(100)
 LANGUAGE                                             VARCHAR2(100)
 JOINING                                              DATE
 SHIFT                                                NUMBER(1)

SQL> alter table programming add Language_type varchar2(15);

Table altered.

SQL> alter table programming rename column joining to date_of_joining;

Table altered.

SQL> alter table programming drop column language_type;

Table altered.

SQL> alter table programming modify Serial varchar2(10);

Table altered.
```

3.Rename:

Syntax: RENAME old_name to new_name;

Code: Rename programming to programming_class;

OUTPUT:

```
SQL> desc programming;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SERIAL                                             VARCHAR2(10)
 NAME                                               VARCHAR2(100)
 LANGUAGE                                           VARCHAR2(100)
 DATE_OF_JOINING                                    DATE
 SHIFT                                              NUMBER(1)
 LANGUAGE_TYPE                                      VARCHAR2(15)

SQL> rename programming to programming_class;

Table renamed.
```

4.Truncate

Syntax: TRUNCATE TABLE table_name;

Code: truncate table programming_class;

OUTPUT:

```
SQL> truncate table programming_class;

Table truncated.

SQL> select * from programming_class;

no rows selected
```

5.Drop

Syntax: DROP TABLE table_name;

Code: drop table programming_class;

OUTPUT:

```
SQL> drop table programming_class;

Table dropped.

SQL>
```

RESULT:

Thus the DDL Commands has been executed successfully

Ex No:02

DML COMMANDS

Aim:

To work with DML commands

Procedure:

Step1: Open Run SQL on Command line and connect to SQL .

Step 2: Then work with database using SQL queries

1.INSERT:

Syntax: INSERT INTO table_name(column_name1,…,column_name_n) VALUES (value1, value2,…,valueN);

Code: insert into tution(rollno,name,standard,subjects,shift,fee) values(1,’Balan’,12,’Maths’,1,3500);

OUTPUT:

```
SQL> insert into tution(rollno, name,standard,subjects,shift,fee) values(01,'Balan',12,'maths',1,3500);

1 row created.

SQL> desc tution;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLLNO                                             NUMBER(15)
 NAME                                               VARCHAR2(25)
 STANDARD                                           NUMBER(2)
 SUBJECTS                                           VARCHAR2(45)
 SHIFT                                              NUMBER(2)
 FEE                                                NUMBER(5)

SQL> select * from tution;

    ROLLNO NAME                      STANDARD
---------- ------------------------ ----------
SUBJECTS                                      SHIFT        FEE
---------------------------------------------- ---------- ----------
         1 Balan                                   12
maths                                                   1       3500


SQL>
```

2.SELECT:

Syntax: SELECT column_name1, column_name2,…,column_name_n FROM table_name WHERE condition_expression;

Code: select rollno,name,standard,subjects from tution where fee>4500;

OUTPUT:

```
SQL> select Rollno,name,standard,subjects from tution where fee>4500;

    ROLLNO NAME                          STANDARD
---------- ------------------------- ----------
SUBJECTS
--------------------------------------------
         5 Vishwa                            12
Maths

         5 Vishwa                            12
Maths


SQL> select Rollno,name,standard,subjects from tution where fee<4500;

    ROLLNO NAME                          STANDARD
---------- ------------------------- ----------
SUBJECTS
--------------------------------------------
         1 Balan                             12
maths

         3 Dheepan                           12
Science


SQL>
```

3.UPDATE:

Syntax: UPDATE table_name SET column_name1+value1, column_name2=value,… WHERE condition;

Code: update tution set standard=12 where name= ’Niteesh’;

4.DELETE:

Syntax: DELETE FROM table_name WHERE condition;

Code: delete from tution where rollno=5;

OUTPUT:

```
SQL> update tution set standard=12 where name='Nitheesh';

1 row updated.

SQL> delete from tution where rollno=5;

2 rows deleted.

SQL> select * from tutuion;
select * from tutuion
              *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> select * from tution;

    ROLLNO NAME                          STANDARD
---------- ------------------------- ----------
SUBJECTS                                     SHIFT        FEE
-------------------------------------------- ---------- ----------
         1 Balan                             12
maths                                             1       3500

         1 Nitheesh                          12
Science                                           2       4500

         3 Dheepan                           12
Science                                           2       3800


    ROLLNO NAME                          STANDARD
---------- ------------------------- ----------
SUBJECTS                                     SHIFT        FEE
-------------------------------------------- ---------- ----------
         4 Aslam                             12
Social                                            2       4500
```

Ex No:03

DCL COMMANDS

Aim:

To work with DCL commands

Procedure:

Step1: Open Run SQL on Command line and connect to SQL .

Step 2: Then work with database using SQL queries

1.GRANT:

Syntax: GRANT privileges ON object_name TO user;

Code: grant all privileges on tution to baldb;

2.REVOKE:

Syntax: REVOKE privileges ON object_name FROM user;

Code: revoke all privileges on tution from baldb;

OUTPUT:

```
SQL> create user baldb identified by balandb;

User created.

SQL> grant all privileges on tution to baldb;

Grant succeeded.

SQL> revoke all privileges on tution from baldb;

Revoke succeeded.

SQL> |
```

Ex No:04

SUB QUERIES

Aim:

To work with Sub Queries

Procedure:

Step1: Open Run SQL on Command line and connect to SQL .

Step 2: Then work with database using SQL queries

1.SELECT :

Syntax: SELECT Column_name FROM table_name WHERE column_name expression operator(SELECT column_nameFROM table_name WHERE condition);

Code: select * from tution where rollno in (select rollno from tution where shift=1);

OUTPUT:

```
SQL> select * from tution where rollno in (select rollno from tution where shift=1);

    ROLLNO NAME                        STANDARD
---------- ------------------------ ----------
SUBJECTS                                          SHIFT        FEE
------------------------------------------- ---------- ----------
         1 Balan                            12
maths                                                  1       3500

         5 Raju                             10
maths                                                  1       5200

         6 vishwa                           10
science                                                1       5300


SQL>
```

2.INSERT:

Syntax: INSERT INTO table1 SELECT * FROM table2 WHERE condition;

Code: sql> create table tution3 as select * from tution where rollno=1;

sql>insert into tution3 (rollno,name,standard,subjects,shift,fee) select * from tution where rollno=2;

OUTPUT:

```
SQL> create table tution3 as select * from tution where rollno = 1;

Table created.

SQL> insert into tution3(rollno, name, standard, subjects, shift, fee) select * from tution where rollno = 2;

1 row created.

SQL> select * from tution3;

    ROLLNO NAME                        STANDARD
---------- ------------------------ ----------
SUBJECTS                                          SHIFT        FEE
------------------------------------------- ---------- ----------
         1 Balan                            12
maths                                                  1       3500

         2 Nitheesh                         12
Science                                                2       4500
```

3.UPDATE:

Syntax: UPDATE table1 SET column_name1=value WHERE column_name in (SELECT column_name FROM table2 WHERE condition);

Code: update tution3 set standard=10 where fee in (select fee from tution where fee=3500);

OUTPUT:

```
SQL> select * from tution;

    ROLLNO NAME                      STANDARD
---------- ------------------------ ----------
SUBJECTS                                     SHIFT        FEE
-------------------------------------------- ---------- ----------
         1 Balan                            12
maths                                                1       3500

         2 Nitheesh                         12
Science                                              2       4500

         3 Dheepan                          12
Science                                              2       3800


    ROLLNO NAME                      STANDARD
---------- ------------------------ ----------
SUBJECTS                                     SHIFT        FEE
-------------------------------------------- ---------- ----------
         4 Aslam                            12
Social                                               2       4500

         5 Raju                             10
maths                                                1       5200

         6 vishwa                           10
science                                              1       5300


6 rows selected.

SQL> select * from tution3;

    ROLLNO NAME                      STANDARD
---------- ------------------------ ----------
SUBJECTS                                     SHIFT        FEE
-------------------------------------------- ---------- ----------
         1 Balan                            12
maths                                                1       3500
```

```
SQL> update tution3 set standard = 10 where fee in ( select fee from tution where fee = 3500);

1 row updated.

SQL> select* from tution3;

    ROLLNO NAME                      STANDARD
---------- ------------------------ ----------
SUBJECTS                                     SHIFT        FEE
-------------------------------------------- ---------- ----------
         1 Balan                            10
maths                                                1       3500

         2 Nitheesh                         12
Science                                              2       4500


SQL>
```

4.DELETE:

Syntax: DELETE FROM table1 WHERE column_name in (SELECT column_name FROM table2 WHERE column_name=value);

Code: delete from tution3 where shift in(select shift from tution3 where fee=4500)

OUTPUT:

```
SQL> delete from tution3 where shift in (select shift from tution3 where fee=4500);

1 row deleted.

SQL> select * from tution3;

    ROLLNO NAME                      STANDARD
---------- ------------------------ ----------
SUBJECTS                                     SHIFT        FEE
-------------------------------------------- ---------- ----------
         1 Balan                            10
maths                                                1       3500
```

Ex.No:05 **JOINS**

Aim:

To create and work with MySQL Joins

Procedure:

Step1: Open Run SQL on Command line and connect to SQL .

Step 2: Then work with database using SQL queries

1.**Inner Join:**

Syntax: Select column_name() from table1 innerjoin table2 on table1.column_name= table2.column_name;

Code:

Select tuition.name,school.dept,tuition.shift from school inner join tuition on school.regno=tuition=reg.no;

OUTPUT:

```
SQL> select * from school;

     REGNO DEPT     SCHOOL
---------- -------- ----------
         1 commerce vhss
         2 biology  petit
         3 computer petit
         4 computer patrick
         5 commerce vhss

SQL> select * from tuition;

     REGNO NAME            SHIFT        FEE A
---------- ---------- ---------- ---------- -
         1 Balan               1       3500 P
         2 Aslam               1       3500 P
         3 Vishwa              1       4000 A
         4 Raju                1       4000 P
         5 Tamizh              2       4000 A
         6 Sandro              2       3800 A

6 rows selected.

SQL> |
```

```
SQL> select tuition.name,school.dept,tuition.shift from school inner join tuition on school.regno=tuition.regno;

NAME       DEPT          SHIFT
---------- -------- ----------
Balan      commerce          1
Aslam      biology           1
Vishwa     computer          1
Raju       computer          1
Tamizh     commerce          2
```

2.Left Join:

Syntax: Select column_names from table1 left join table2 on table1.column_name=table2.column_name;

Code: select tuition.name, school.dept from school left join tuition on school.regno=tuition.regno;

OUTPUT:

```
SQL> select tuition.name,school.dept from school left join tuition on school.regno=tuition.regno;

NAME       DEPT
---------- --------
Balan      commerce
Aslam      biology
Vishwa     computer
Raju       computer
Tamizh     commerce
```

3.Right Join:

Syntax: Select column_names from table1 right join table2 on table1.column_name=table2.column_name;

Code: select tuition.name, school.dept from school left join tuition on school.regno=tuition.regno;

OUTPUT:

```
SQL> select tuition.name,school.dept from school right join tuition on school.regno=tuition.regno;

NAME       DEPT
---------- --------
Balan      commerce
Aslam      biology
Vishwa     computer
Raju       computer
Tamizh     commerce
Sandro
```

4.Full Outer Join:

Syntax: Select column_names from table1 full outer join table2 on table1.column_name=table2.column_name;

Code: Select tuition.name, tuition.shift, school.school,school.dept from school full join tuition on school.regno=tuition.regno;

OUTPUT:

```
SQL> select tuition.name,tuition.shift,school.school,school.dept from school full join tuition on school.regno=tuition.regno;

NAME            SHIFT SCHOOL     DEPT
---------- ---------- ---------- --------
Balan               1 vhss       commerce
Aslam               1 petit      biology
Vishwa              1 petit      computer
Raju                1 patrick    computer
Tamizh              2 vhss       commerce
Sandro              2

6 rows selected.
```

RESULT:

Thus the Join queries has been executed successfully

Ex No:06

PL/SQL

Aim:

To work with PL/SQL

Procedure:

Step1: Open Run SQL on Command line and connect to SQL .

Step 2: Then work with database using PL/SQL Block commands

i. Declare
ii. Begin
iii. Exception
iv. End

SYNTAX:

DECLARE

<declarations section>

BEGIN

<executable section commands>

EXCEPTION

<exception handling>

END;

EXAMPLE:

1.Addition of Two Numbers:

Program Code:

```
SQL> SET SERVEROUTPUT ON;
SQL> declare
  2  x number(5);
  3  y number(5);
  4  z number(5);
  5  begin
  6  x:=50;
  7  y:=20;
  8  z:=x+y;
  9  dbms_output.put_line('sum is'||z);
 10  end;
 11  /
```

Output:

```
SQL> SET SERVEROUTPUT ON;
SQL> declare
  2  x number(5);
  3  y number(5);
  4  z number(5);
  5  begin
  6  x:=50;
  7  y:=20;
  8  z:=x+y;
  9  dbms_output.put_line('sum is'||z);
 10  end;
 11  /
sum is70

PL/SQL procedure successfully completed.

SQL>
```

2.Generating Series:

Program Code:

```
SQL> SET SERVEROUTPUT ON;
SQL> declare
 2  n number(5);
 3  begin
 4  n:=1;
 5  for i in 1..10 loop
 6  case n
 7  when 1 then
 8  dbms_output.put_line(i);
 9  when 2 then
10  if mod(i,2)=0 then
11  dbms_output.put_line(i);
12  end if;
13  when 3 then
14  if mod(i,2)!=0 then
15  dbms_output.put_line(i);
16  end if;
17  end case;
18  end loop;
19  end;
20  /
```

OUTPUT:

```
SQL> SET SERVEROUTPUT ON;
SQL> declare
  2  n number(5);
  3  begin
  4  n:=1;
  5  for i in 1..10 loop
  6  case n
  7  when 1 then
  8  dbms_output.put_line(i);
  9  when 2 then
 10  if mod(i,2)=0 then
 11  dbms_output.put_line(i);
 12  end if;
 13  when 3 then
 14  if mod(i,2)!=0 then
 15  dbms_output.put_line(i);
 16  end if;
 17  end case;
 18  end loop;
 19  end;
 20  /
1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.

SQL>
```

Ex no: 06 CURSOR PROCEDURE FUNCTIONS

AIM:

To write a SQL program to work with cursor, procedure and functions.

PROCEDURE:

Step 1: Open Run SQL on Command line and connect to SQL

Step 2: Then work with database using SQL queries.

PL/SQL PROCEDURE:

The PL/SQL stored procedure or simply a procedure is a PL/SQL block which performs one or more specific tasks. It is just like procedures in other programming languages.

The procedure contains a header and a body.

- **Header:** The header contains the name of the procedure and the parameters or variables passed to the procedure.
- **Body:** The body contains a declaration section, execution section and exception section similar to a general PL/SQL block.

Syntax for creating procedure:

```
CREATE [OR REPLACE] PROCEDURE procedure_name

   [ (parameter [,parameter]) ]

IS

   [declaration_section]

BEGIN

   executable_section

[EXCEPTION

   exception_section]

END [procedure_name];
```

TABLE QUERY:

```
create table employee(emp_id number(5)primary key, emp_name varchar2(20), city varchar2(20), salary number(7), age number(5));

 insert into employee values (1, 'Raju', 'Pdy', 800000, 20);

 insert into employee values (2, 'Niteesh', 'Pdy', 790000, 21);

 insert into employee values (3, 'Punith', 'AP', 750000, 20);

 insert into employee values (4, 'Sidharth', 'MP', 650000, 21);

 insert into employee values (5, 'Mantu', 'Delhi', 900000, 22);
```

PROGRAM CODE:

```
DECLARE

PROCEDURE pro

AS

BEGIN

  dbms_output.put_line('It is working perfectly!');

END;

BEGIN

pro();

END;

/
```

OUTPUT:

```
SQL> set serveroutput on;
SQL> ed pro;

SQL> @pro;
It is working perfectly!

PL/SQL procedure successfully completed.
```

PL/SQL – CURSORS:

A cursor is used to referred to a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors:

- Implicit Cursors
- Explicit Cursors

IMPLICIT CURSOR:

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement.

1 %FOUND

Returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows or a SELECT INTO statement returned one or more rows. Otherwise, it returns FALSE.

2 %NOTFOUND

The logical opposite of %FOUND. It returns TRUE if an INSERT, UPDATE, or DELETE statement affected no rows, or a SELECT INTO statement returned no rows. Otherwise, it returns FALSE.

3 %ISOPEN

Always returns FALSE for implicit cursors, because Oracle closes the SQL cursor automatically after executing its associated SQL statement.

4 %ROWCOUNT

Returns the number of rows affected by an INSERT, UPDATE, or DELETE statement, or returned by a SELECT INTO statement.

EXPLICIT CURSOR:

Explicit cursors are programmer-defined cursors for gaining more control over the context area.

The syntax for creating an explicit cursor is –

CURSOR cursor_name IS select_statement;

Working with an explicit cursor includes the following steps –

- Declaring the cursor for initializing the memory
- Opening the cursor for allocating the memory
- Fetching the cursor for retrieving the data
- Closing the cursor to release the allocated memory

PROGRAM CODE:

```
DECLARE

   e_id employee.emp_id%type;

   e_name employee.emp_name%type;

   e_city employee.city%type;

   cursor e_employee is

   select emp_id, emp_name, city from employee;

begin

   open e_employee;

   loop
```

```
  fetch e_employee into e_id, e_name, e_city;

  exit when e_employee%notfound;

  dbms_output.put_line(e_id || ' ' || e_name || ' ' || e_city);

  end loop;

  close e_employee;

end;

/
```

OUTPUT:

```
SQL> ed e

SQL> @e;
1 Raju Pdy
2 Niteesh Pdy
3 Punith AP
4 Sidharth MP
5 Mantu Delhi

PL/SQL procedure successfully completed.
```

PL/SQL FUNCTION:

The PL/SQL Function is very similar to PL/SQL Procedure. The main difference between procedure and a function is, a function must always return a value, and on the other hand a procedure may or may not return a value.

Syntax to create a function:

CREATE [OR REPLACE] FUNCTION function_name [parameters]

[(parameter_name [IN | OUT | IN OUT] type [, ...])]

```
RETURN return_datatype

{IS | AS}

BEGIN

  < function_body >

END [function_name];
```

PROGRAM CODE:

```
DECLARE

n number;

t number;

FUNCTION func

RETURN number IS

  total number(2) := 0;

BEGIN

  SELECT count(*) into total

  FROM employee;

   RETURN total;

END;

BEGIN

n:=2;

       t:=func();

       dbms_output.put_line(t);
```

END;

/

OUTPUT:

```
SQL> set serveroutput on;
SQL> ed func;

SQL> @func;
5

PL/SQL procedure successfully completed.
```

RESULT:

Thus the queries for Procedure, Cursors and Functions were successfully executed and the output is noted.

Ex.No:08 **TRIGGERS**

Aim:

To write a program and work with Triggers

Procedure:

Step1: Open MySQL workbench and connect to SQL .

Step 2: Then work with database using SQL queries and PI/SQL

1.**Trigger:**

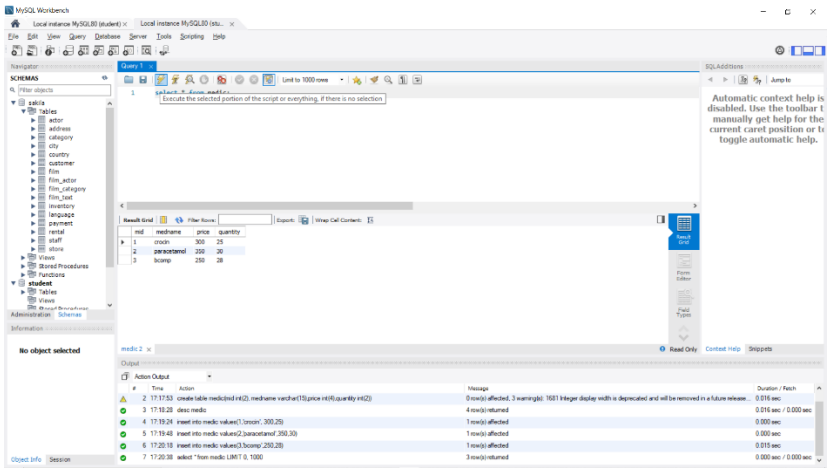
Syntax: CREATE [OR REPLACE] TRIGGER trigger_name. {BEFORE | AFTER | INSTEAD OF } {INSERT [OR] | UPDATE [OR] | DELETE} [OF col_name]

Code:

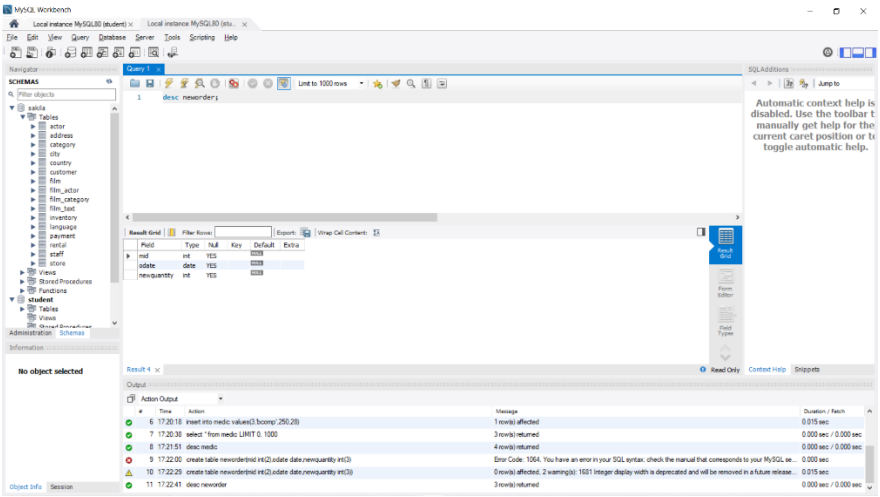
```
delimiter $$
create trigger neworderformedicine
after update on medic
for each row
begin
if new.quantity<20 then
insert into neworder values(new.mid,sysdate(),200);
end if;
end;
$$
```

OUTPUT:

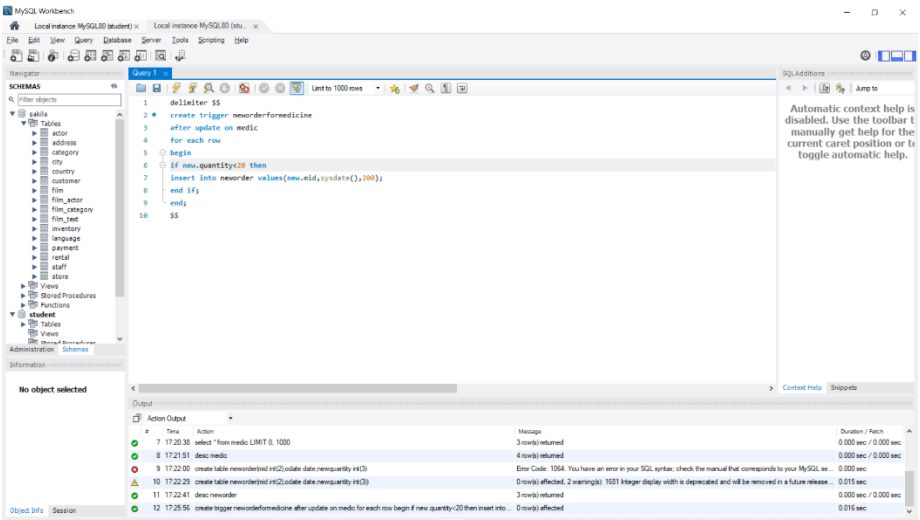
Medic table:



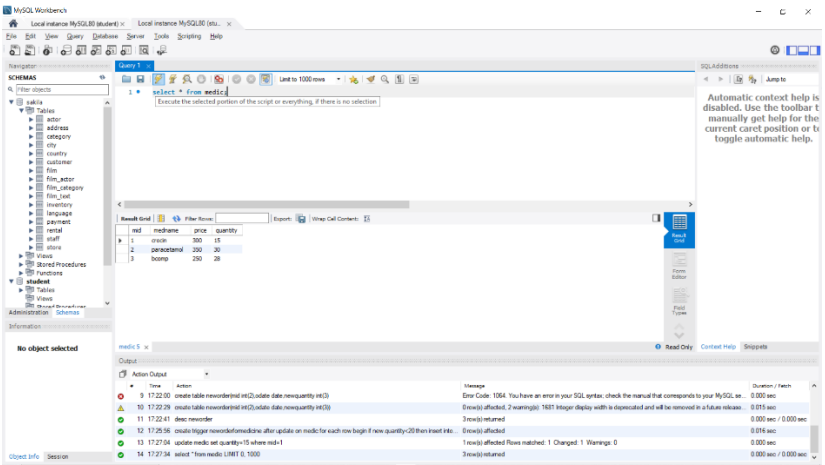
Neworder table:



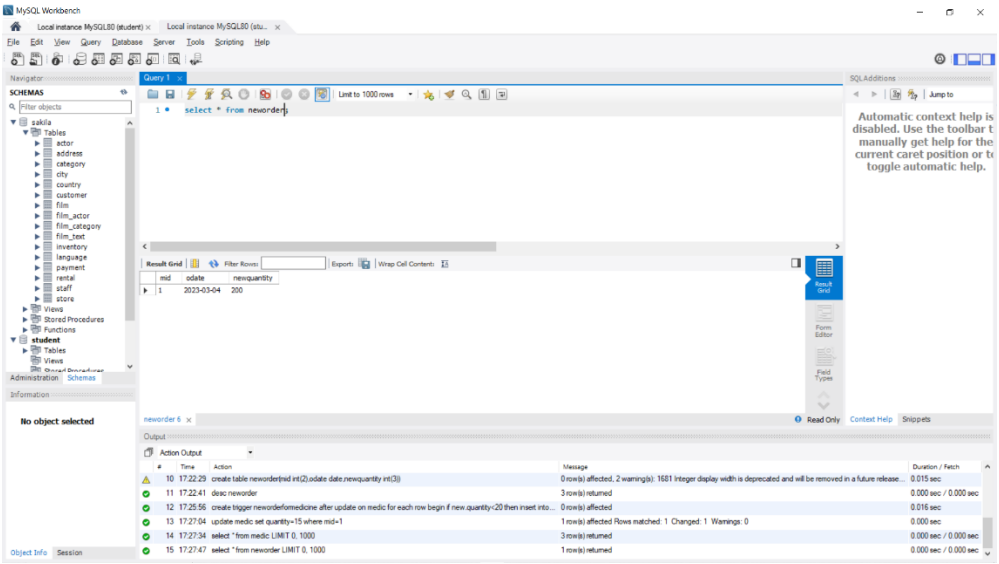
Trigger Query:



Updated table Medic:



Triggered table Neworder:



RESULT:

Thus the Trigger queries has been executed successfully

STUDENT LOGIN PAGE

ABSTRACT:

Creating an application project using SQL and PHP language. The application project is based on SQL and PHP language connected using XAMPP. The front-end of the program is written using HTML as markup language and CSS as style sheet.The back-end of the program is written using PHP programming language.The database connection is written using MySQL query language.The full development is executed using XAMPP localhost server.

MODULE:

The entire project consists of five modules

1. config.php
2. dashboard.php
3. index.php
4. registration.php
5. style.css

CODING:

I.Config.php:

```
<?php
define('DB_SERVER','localhost');
define('DB_USER','root');
define('DB_PASS' ,'');
define('DB_NAME', 'dbms');
$con = mysqli_connect(DB_SERVER,DB_USER,DB_PASS,DB_NAME);
// Check connection
if (mysqli_connect_errno())
{
 echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

II.Dashboard.php:

```
<?php
session_start();
error_reporting(0);
include("config.php");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
```

```
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

    <title>Balan | Dashboard</title>

</head>
<body>
   <div class="box">
    <div class="container">

        <div class="top">
<header>Hi</header>


        </div>

            <div class="top">
<header>
 <h1>
<?php $query=mysqli_query($con,"select name from student where register='".
$_SESSION['login']."'");
while($row=mysqli_fetch_array($query))
{
            echo $row['name'];
} ?>
</h1>
</header>
</div>
        </div>
    </div>
</div>
</body>
</html>
```

3.Index.php:

```
<?php
session_start();
error_reporting(0);
include("config.php");
if(isset($_POST['submit']))
{
$ret=mysqli_query($con,"SELECT * FROM student WHERE register='".$_POST['register']."' and
password='".$_POST['password']."'");
$num=mysqli_fetch_array($ret);
if($num>0)
{
$extra="dashboard.php";//
$_SESSION['login']=$_POST['register'];
$_SESSION['id']=$num['id'];
$host=$_SERVER['HTTP_HOST'];
$uip=$_SERVER['REMOTE_ADDR'];
$status=1;
```

```
// For stroing log if user login successfull
$log=mysqli_query($con,"insert into userlog(uid,username,userip,status) values('".
$_SESSION['id']."','".$_SESSION['login']."','$uip','$status')");
$uri=rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
header("location:http://$host$uri/$extra");
exit();
}
else
{
        // For stroing log if user login unsuccessfull
$_SESSION['login']=$_POST['register'];
$uip=$_SERVER['REMOTE_ADDR'];
$status=0;
mysqli_query($con,"insert into userlog(username,userip,status) values('".
$_SESSION['login']."','$uip','$status')");
$_SESSION['errmsg']="Invalid username or password";
$extra="index.php";
$host  = $_SERVER['HTTP_HOST'];
$uri  = rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
header("location:http://$host$uri/$extra");
exit();
}
}
?>


<!DOCTYPE html>
<html lang="en">
        <head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="style.css">
   <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

   <title>Balan | Login</title>

</head>
        <body class="box">
                <div class="container">
                        <div class="main-login col-xs-10 col-xs-offset-1 col-sm-8 col-sm-offset-2
col-md-4 col-md-offset-4">
                                <div class="logo margin-top-30">

                                </div>

                                        <form class="form-login " method="post">



                                                                <span style="color:red;"><?php echo
$_SESSION['errmsg']; ?><?php echo $_SESSION['errmsg']="";?></span>
```

```
<div class="form-group input-field">
        <input type="text" class="form-control input" name="register" placeholder="Username" >
        <i class='bx bx-user' ></i>
      </div>

 <div class="form-group input-field">
        <input type="Password" class="form-control input"  name="password"
placeholder="Password" id="">
        <i class='bx bx-lock-alt'></i>
      </div>

                                                <div class="input-field form-actions">

                                                        <button type="submit" class="submit
btn btn-primary pull-right" name="submit">
                                                                Login <i class="fa fa-arrow-
circle-right"></i>
                                                        </button>

                                                </div>



                                                <div class="new-account">
<br>

                                                        Don't have an account yet?
                                                        <a href="registration.php">
                                                                Create an account
                                                        </a>
                                                </div>


                                        </form>



                                </div>

                        </div>
                </div>
                <script src="vendor/jquery/jquery.min.js"></script>
                <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
                <script src="vendor/modernizr/modernizr.js"></script>
                <script src="vendor/jquery-cookie/jquery.cookie.js"></script>
                <script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
                <script src="vendor/switchery/switchery.min.js"></script>
                <script src="vendor/jquery-validation/jquery.validate.min.js"></script>
```

```
                <script src="assets/js/main.js"></script>

                <script src="assets/js/login.js"></script>
                <script>
                        jQuery(document).ready(function() {
                                Main.init();
                                Login.init();
                        });
                </script>

        </body>
        <!-- end: BODY -->
</html>
```

IV. Registration.php

```
<?php
include_once('config.php');
if(isset($_POST['submit']))
{
$name=$_POST['name'];
$register=$_POST['register'];
$gender=$_POST['gender'];
$email=$_POST['email'];
$password=$_POST['password'];
$query=mysqli_query($con,"insert into request(name,register,email,gender,password,requestDate)
values('$name','$register','$email','$gender','$password',curdate())");
if($query)
{
        echo "<script>alert('Successfully registered. You can login after few minutes ');</script>";
        //header('location:index.php');
}
else {
echo "<script>alert('request unsuccessfully');</script>";
}
}
?>


<!DOCTYPE html>
<html lang="en">

        <head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="style.css">
   <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

   <title>Balan | Register</title>
```

```
<script type="text/javascript">
function valid()
{
 if(document.registration.password.value!= document.registration.password_again.value)
{
alert("Password and Confirm Password Field do not match  !!");
document.registration.password_again.focus();
return false;
}
return true;
}
</script>


</head>

<body class="box">
<!-- start: REGISTRATION -->
<div class="container">

<div class="main-login col-xs-10 col-xs-offset-1 col-sm-8 col-sm-
offset-2 col-md-4 col-md-offset-4">
<!-- start: REGISTER BOX -->
<div class="box-register">
<form name="registration" id="registration"  method="post"
onSubmit="return valid();">

<legend>
Register
</legend>
<p>
Enter your details below:
</p>
<div class="input-field form-group">
<span>
<input type="text" class="input form-
control" name="name" placeholder="Full Name" required>
</span>
</div>

<div class="form-group input-field">
<span>
<input type="text" class=" input form-
control" name="register" placeholder="Register no." required>
</span>
</div>

<div class="form-group input-field">
<label class=" input-field">
Gender
</label>
```

```
                                                        <div class="clip-radio radio-primary">
<span>
                                                                <input type="radio" id="rg-
female" name="gender" value="female" >
                                                                <label for="rg-female">
                                                                        Female
                                                                </label>

                                                                <input type="radio" id="rg-male"
name="gender" value="male">
                                                                <label for="rg-male">
                                                                        Male
                                                                </label>
                                                        </div>
                                                </div>

                                                <div class="form-group input-field">
                                                        <span class="input-icon">
                                                                <input type="email"
class="form-control input" name="email" id="email" onBlur="userAvailability()"
placeholder="Email" required>
                                                                <i class="fa fa-envelope"></i>
</span>

                                                </div>
                                                <div class="form-group input-field">
                                                        <span class="input-icon">
                                                                <input type="password"
class="form-control input" id="password" name="password" placeholder="Password" required>
                                                                <i class="fa fa-lock"></i>
</span>
                                                </div>
                                                <div class="form-group input-field">
                                                        <span class="input-icon">
                                                                <input type="password"
class="form-control input"  id="password_again" name="password_again" placeholder="Password
Again" required>
                                                                <i class="fa fa-lock"></i>
</span>
                                                </div>

                                                <div class="form-actions input-field">
                                                        <p>
                                                                Already have an account?
                                                                <a href="index.php">
                                                                        Log-in
                                                                </a>
                                                        </p>
                                                        <button type="submit" class="submit
btn btn-primary pull-right" id="submit" name="submit">
                                                                Submit <i class="fa fa-arrow-
circle-right"></i>
```

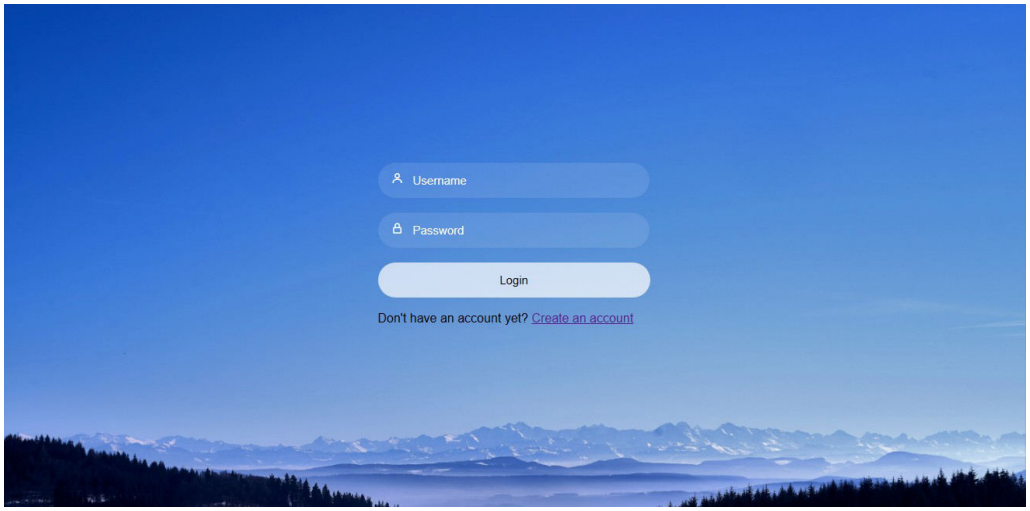
```
                                                                    </button>
                                                                </div>

                                                    </form>

                                                    <div class="copyright">
                                                            <span class="current-year"></span>.
<span>Balan&copy;All rights reserved</span>
                                                    </div>

                                            </div>

                                    </div>
                            </div>
                            <script src="vendor/jquery/jquery.min.js"></script>
                            <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
                            <script src="vendor/modernizr/modernizr.js"></script>
                            <script src="vendor/jquery-cookie/jquery.cookie.js"></script>
                            <script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
                            <script src="vendor/switchery/switchery.min.js"></script>
                            <script src="vendor/jquery-validation/jquery.validate.min.js"></script>
                            <script src="assets/js/main.js"></script>
                            <script src="assets/js/login.js"></script>
                            <script>
                                    jQuery(document).ready(function() {
                                            Main.init();
                                            Login.init();
                                    });
                            </script>

                    <script>
function userAvailability() {
$("#loaderIcon").show();
jQuery.ajax({
url: "check_availability.php",
data:'email='+$("#email").val(),
type: "POST",
success:function(data){
$("#user-availability-status1").html(data);
$("#loaderIcon").hide();
},
error:function (){}
});
}
</script>

                    </body>
                    <!-- end: BODY -->
</html>
```

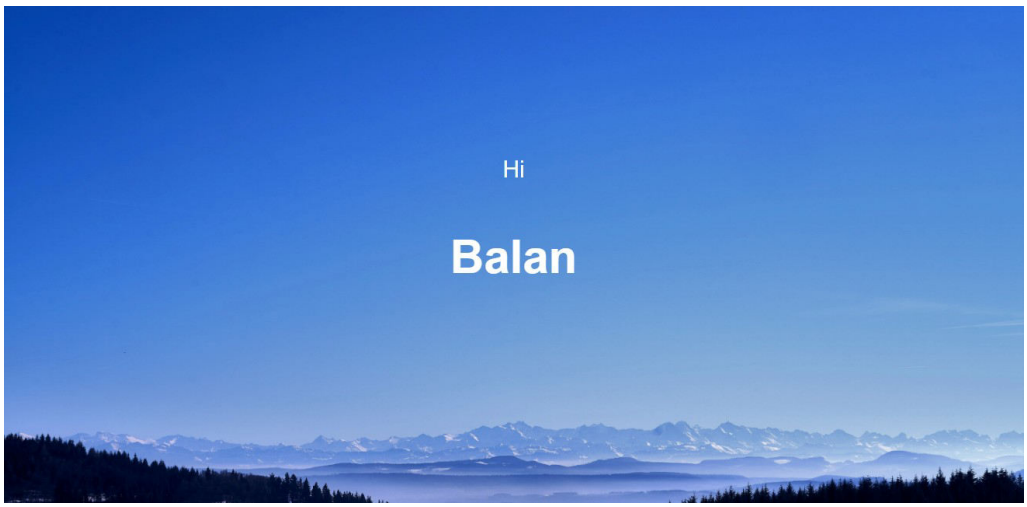
V. Style.css

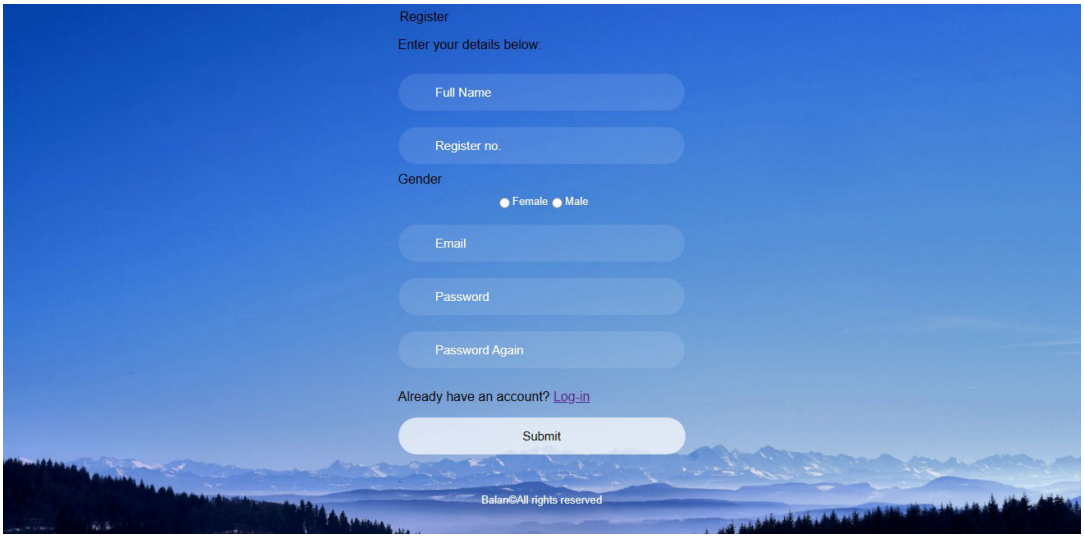
```
@import url('https://fonts.googleapis.com/css2?
family=Nunito:wght@400;600;800&display=swap');
*{
    font-family: 'poppins' ,sans-serif;
}
body{
    background-image: url("images/1.jpg");
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
    background-repeat: no-repeat;

}
.box{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 90vh;
}
.container{
    width: 350px;
    display: flex;
    flex-direction: column;
    padding: 0 15px 0 15px;

}
span{
    color: #fff;
    font-size: small;
    display: flex;
    justify-content: center;
    padding: 10px 0 10px 0;
}
header{
    color: #fff;
    font-size: 30px;
    display: flex;
    justify-content: center;
    padding: 10px 0 10px 0;
}

.input-field .input{
    height: 45px;
    width: 87%;
    border: none;
    border-radius: 30px;
    color: #fff;
    font-size: 15px;
    padding: 0 0 0 45px;
    background: rgba(255,255,255,0.1);
```

```
    outline: none;
}
i{
    position: relative;
    top: -33px;
    left: 17px;
    color: #fff;
}
::-webkit-input-placeholder{
    color: #fff;
}
.submit{
    border: none;
    border-radius: 30px;
    font-size: 15px;
    height: 45px;
    outline: none;
    width: 100%;
    color: black;
    background: rgba(255,255,255,0.7);
    cursor: pointer;
    transition: .3s ;
}
.submit:hover{
    box-shadow: 1px 5px 7px 1px rgba(0, 0, 0, 0.2);
}
.two-col{
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    color: #fff;
    font-size: small;
    margin-top: 10px;
}
.one{
    display: flex;
}
label a{
    text-decoration: none;
    color: #fff;
}
```

OUTPUT:


Username
Password
Login
Don't have an account yet? Create an account


Hi
Balan


Register
Enter your details below:
Full Name
Register no.
Gender
Female
Male
Email
Password
Password Again
Already have an account? Log-in
Submit
Balan©All rights reserved