DBMS LAB PROGRAMS

B.VANI(192472348)

1.To Create, Alter and Drop the table using Data Definition Language.

OUTPUT:

```
mysql> create table student(Regno int(3),Name char(15),gender char(1),Dob int(10),mobileno int(10),city char(10));
Query OK, 0 rows affected (0.14 sec)

mysql> desc student;
+----------+----------+------+-----+---------+-------+
| Field    | Type     | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| Regno    | int(3)   | YES  |     | NULL    |       |
| Name     | char(15) | YES  |     | NULL    |       |
| gender   | char(1)  | YES  |     | NULL    |       |
| Dob      | int(10)  | YES  |     | NULL    |       |
| mobileno | int(10)  | YES  |     | NULL    |       |
| city     | char(10) | YES  |     | NULL    |       |
+----------+----------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
mysql> create table department(deptno int(10),deptname char(10),depthead char(10));
Query OK, 0 rows affected (0.11 sec)

mysql> desc department;
+-----------+-----------+------+-----+---------+-------+
| Field     | Type      | Null | Key | Default | Extra |
+-----------+-----------+------+-----+---------+-------+
| deptno    | int(10)   | YES  |     | NULL    |       |
| deptname  | char(10)  | YES  |     | NULL    |       |
| depthead  | char(10)  | YES  |     | NULL    |       |
+-----------+-----------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

2.To add the constraints like primary key, foreign key, unique key and check using DDL commands.

OUTPUT:

```
mysql> alter table student add primary key(city);
Query OK, 0 rows affected (0.14 sec)
```

```
mysql> desc student;
+----------+----------+------+-----+---------+-------+
| Field    | Type     | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| Regno    | int(3)   | YES  |     | NULL    |       |
| Name     | char(15) | YES  |     | NULL    |       |
| gender   | char(1)  | YES  |     | NULL    |       |
| Dob      | int(10)  | YES  |     | NULL    |       |
| mobileno | int(10)  | YES  |     | NULL    |       |
| city     | char(10) | NO   | PRI | NULL    |       |
+----------+----------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
mysql> alter table department add primary key(deptno);
Query OK, 0 rows affected (0.11 sec)
```

```
mysql> desc department;
+----------+----------+------+-----+---------+-------+
| Field    | Type     | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| deptno   | int(10)  | NO   | PRI | NULL    |       |
| deptname | char(10) | YES  |     | NULL    |       |
| depthead | char(10) | YES  |     | NULL    |       |
+----------+----------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> desc student2;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| RegNo    | int         | NO   | PRI | NULL    |       |
| S1Name   | varchar(15) | YES  | UNI | NULL    |       |
| Age      | char(2)     | YES  |     | NULL    |       |
| MobileNo | int         | YES  |     | NULL    |       |
| address  | varchar(15) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> create table student3(RegNo int(3),Name varchar(15));
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> Alter table student3 ADD PRIMARY KEY(RegNo);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc student3;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| RegNo | int         | NO   | PRI | NULL    |       |
| Name  | varchar(15) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> Alter table student2 ADD FOREIGN KEY (RegNo) REFERENCES Student3(RegNo);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

3.To perform Data Manipulation Language (DML) Commands such as INSERT, SELECT, UPDATE, DELETE in the table.

```
mysql> SELECT * FROM college;
+------+---------+-----------+------+------+------+-------+
| sno  | name    | regno     | dept | age  | DOB  | facno |
+------+---------+-----------+------+------+------+-------+
|    1 | YUGA    | 191811316 | cse  |   19 | 2000 |   200 |
|    2 | SRI     | 191811382 | cse  |   19 | 2000 |   201 |
|    3 | ABDUL   | 191811362 | cse  |   20 | 1999 |   202 |
|    4 | SUPRIYA | 191811322 | cse  |   20 | 1999 |   203 |
+------+---------+-----------+------+------+------+-------+
4 rows in set (0.00 sec)
```

```
mysql> delete from faculty where Facno='2';
Query OK, 1 row affected (0.06 sec)

mysql> select * from faculty;
+-------+---------+--------+------+----------+------+
| Facno | FacName | gender | Dob  | mobileno | DOJ  |
+-------+---------+--------+------+----------+------+
|     1 | ramu    | M      | 2000 | 90515252 | 2001 |
+-------+---------+--------+------+----------+------+
1 row in set (0.00 sec)
```

```
mysql> select * from faculty;
+-------+---------+--------+------+----------+------+
| Facno | FacName | gender | Dob  | mobileno | DOJ  |
+-------+---------+--------+------+----------+------+
|     1 | ramu    | M      | 2000 | 90515252 | 2001 |
+-------+---------+--------+------+----------+------+
1 row in set (0.00 sec)

mysql> select * from course;
+----------+------------+------------+-------+--------+-------+
| courseno | coursedesc | coursetype | semno | hallno | Facno |
+----------+------------+------------+-------+--------+-------+
|        1 | prgm       | u          |  2000 |    201 |  2001 |
+----------+------------+------------+-------+--------+-------+
1 row in set (0.00 sec)

mysql> select * from department;
+--------+----------+----------+
| deptno | deptname | depthead |
+--------+----------+----------+
|      1 | cse      | vasu     |
+--------+----------+----------+
1 row in set (0.00 sec)
```

4.To view the records from the tables using SELECT commands with WHERE Clause and Pattern matching.

OUTPUT:

```
mysql> select * from course;
+----------+------------+------------+-------+--------+-------+
| courseno | coursedesc | coursetype | semno | hallno | Facno |
+----------+------------+------------+-------+--------+-------+
|        1 | prgm       | u          |  2000 |    201 |  2001 |
+----------+------------+------------+-------+--------+-------+
1 row in set (0.00 sec)
```

```
mysql> SELECT  * FROM student;
+------+--------------------+------+------+------+-------+--------+
| sno  | name               | dept | age  | DOB  | Facno | gender |
+------+--------------------+------+------+------+-------+--------+
|    1 | DHONI              | cse  |   19 | 2000 |     7 | M      |
|    2 | virat              | cse  |   19 | 2000 |    71 | M      |
|    3 | sindhu             | cse  |   18 | 2001 |    71 | F      |
|    4 | VIJAY_DEVARAKONDA  | cse  |   21 | 1998 |     3 | M      |
+------+--------------------+------+------+------+-------+--------+
4 rows in set (0.00 sec)
```

```
mysql> select * from course;
+----------+------------+------------+-------+--------+-------+
| courseno | coursedesc | coursetype | semno | hallno | Facno |
+----------+------------+------------+-------+--------+-------+
|        1 | prgm       | u          |  2000 |    201 |  2001 |
+----------+------------+------------+-------+--------+-------+
1 row in set (0.00 sec)
```

5.To view the records from the tables using SELECT commands with BETWEEN, IN, Aggregate functions.

OUTPUT:

```
mysql> select course from course;
ERROR 1054 (42S22): Unknown column 'course' in 'field list'
mysql> select * from course;
+----------+-----------+-------------+--------+
| courseno | regno     | coursedesc  | scores |
+----------+-----------+-------------+--------+
| C001     | 191811306 | mathematics |     93 |
| C001     | 191811307 | mathematics |     80 |
| C002     | 191811306 | OOAD        |     75 |
| C002     | 191811307 | OOAD        |     85 |
| C003     | 191811306 | DBMS        |     91 |
| C003     | 191811307 | DBMS        |     93 |
+----------+-----------+-------------+--------+
6 rows in set (0.00 sec)
```

```
mysql> select courseno from course where scores between 50 and 80;
+----------+
| courseno |
+----------+
| C001     |
| C002     |
+----------+
2 rows in set (0.00 sec)
```

```
mysql> select courseno,coursedesc,count(regno) from course where regno like '191811%';
+----------+-------------+--------------+
| courseno | coursedesc  | count(regno) |
+----------+-------------+--------------+
| C001     | mathematics |            6 |
+----------+-------------+--------------+
1 row in set (0.00 sec)
```

6.To view the records from the tables using SELECT commands with Group By, Having, Order By

OUTPUT:

```
mysql> select courseno,count(courseno) from course group by courseno;
+----------+-----------------+
| courseno | count(courseno) |
+----------+-----------------+
|        1 |               1 |
+----------+-----------------+
1 row in set (0.06 sec)
```

```
mysql> select courseno,count(score) from course group by courseno;
+----------+--------------+
| courseno | count(score) |
+----------+--------------+
|        1 |            1 |
+----------+--------------+
1 row in set (0.00 sec)
```

```
mysql> select name,DOB from student order by regno;
+------+--------+-----------+------+
| sno  | name   | regno     | DOB  |
+------+--------+-----------+------+
|    1 | mukesh | 191811309 | 2001 |
|    2 | nani   | 191811319 | 2001 |
|    3 | ram    | 191811320 | 2000 |
|    4 | ntr    | 191811321 | 2000 |
+------+--------+-----------+------+
```

7.To perform subquery and correlated query on the given relation.

OUTPUT:

```
mysql> select * from faculty order by doj limit 1;
+-------+-------------+--------+------------+------------+------------+--------+
| FacNo | FacultyName | gender | Dob        | Doj        | Mobileno   | DeptNo |
+-------+-------------+--------+------------+------------+------------+--------+
| fo1   | chaithu     | m      | 0000-00-00 | 0000-00-00 | 2147483647 | cse    |
+-------+-------------+--------+------------+------------+------------+--------+
1 row in set (0.00 sec)
```

```
mysql> select * from stud2 where marks>(select avg(marks) from stud2);
+-------+------+----------+-------+------------+-----------+
| regno | name | courseno | marks | facultydoj | assements |
+-------+------+----------+-------+------------+-----------+
|  1234 | nani | coo1     |    50 |       2001 |         5 |
|  1254 | ramu | coo1     |    50 |       2002 |         4 |
+-------+------+----------+-------+------------+-----------+
2 rows in set (0.00 sec)
```

8.To perform JOIN using EquiJoin, InnerJoin, OuterJoin on the given relation.

OUTPUT:

```
mysql> select faculty.facno,faculty.facname,department.deptno,department.deptname from faculty c
+-------+---------+--------+----------+
| facno | facname | deptno | deptname |
+-------+---------+--------+----------+
| 802   | Ratnam  | 11     | Sales    |
+-------+---------+--------+----------+
1 row in set (0.00 sec)
```

```
mysql> select student.name,student.marks,course.courseno from student inner join course on student.course=course.courseno;
+--------+-------+----------+
| name   | marks | courseno |
+--------+-------+----------+
| Ramu   |     0 | C00      |
| Geetha |     0 | C00      |
| Pooja  |     0 | C00      |
+--------+-------+----------+
3 rows in set (0.00 sec)
```

```
mysql> select faculty.facno,faculty.facname,department.deptno,department.deptname from faculty cross join department;
+-------+---------+--------+----------+
| facno | facname | deptno | deptname |
+-------+---------+--------+----------+
| 802   | Ratnam  | 11     | Sales    |
+-------+---------+--------+----------+
1 row in set (0.00 sec)
```

9.To create view, index and sequence on the given relation.

OUTPUT:

```
mysql> select * from employees1;
+-------------+-------------+-----------+---------------+--------+
| employee_id | first_name  | last_name | device_serial | salary |
+-------------+-------------+-----------+---------------+--------+
|           1 | John        | Smith     | ABC123        |  60000 |
|           2 | Jane        | Doe       | DEF456        |  65000 |
|           3 | Bob         | Johnson   | GHI789        |  70000 |
|           4 | Sally       | Fields    | JKL012        |  75000 |
|           5 | Michael     | Smith     | MNO345        |  80000 |
|           6 | Emily       | Jones     | PQR678        |  85000 |
|           7 | David       | Williams  | STU901        |  90000 |
|           8 | Sarah       | Johnson   | VWX234        |  95000 |
|           9 | James       | Brown     | YZA567        | 100000 |
|          10 | Emma        | Miller    | BCD890        | 105000 |
|          11 | William     | Davis     | EFG123        | 110000 |
|          12 | Olivia      | Garcia    | HIJ456        | 115000 |
|          13 | Christopher | Rodriguez | KLM789        | 120000 |
|          14 | Isabella    | Wilson    | NOP012        | 125000 |
|          15 | Matthew     | Martinez  | QRS345        | 130000 |
|          16 | Sophia      | Anderson  | TUV678        | 135000 |
|          17 | Daniel      | Smith     | WXY901        | 140000 |
|          18 | Mia         | Thomas    | ZAB234        | 145000 |
|          19 | Joseph      | Hernandez | CDE567        | 150000 |
|          20 | Abigail     | Smith     | FGH890        | 155000 |
+-------------+-------------+-----------+---------------+--------+
20 rows in set (0.00 sec)
```

```
mysql> desc employees1;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| employee_id   | int         | YES  | MUL | NULL    |       |
| first_name    | varchar(50) | YES  |     | NULL    |       |
| last_name     | varchar(50) | YES  |     | NULL    |       |
| device_serial | varchar(15) | YES  |     | NULL    |       |
| salary        | int         | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> insert into v1 values(101,100001);
Query OK, 1 row affected (0.01 sec)

mysql> select * from v1;
+--------------+---------+
| employee_id | salary |
+--------------+---------+
|            1 |   60000 |
|            2 |   65000 |
|            3 |   70000 |
|            4 |   75000 |
|            5 |   80000 |
|            6 |   85000 |
|            7 |   90000 |
|            8 |   95000 |
|            9 |  100000 |
|           10 |  105000 |
|           11 |  110000 |
|           12 |  115000 |
|           13 |  120000 |
|           14 |  125000 |
|           15 |  130000 |
|           16 |  135000 |
|           17 |  140000 |
|           18 |  145000 |
|           19 |  150000 |
|           20 |  155000 |
|          101 |  100001 |
+--------------+---------+
21 rows in set (0.00 sec)
```

```
mysql> select * from employees1;
+-------------+-------------+-----------+---------------+--------+
| employee_id | first_name  | last_name | device_serial | salary |
+-------------+-------------+-----------+---------------+--------+
|           1 | John        | Smith     | ABC123        |  60000 |
|           2 | Jane        | Doe       | DEF456        |  65000 |
|           3 | Bob         | Johnson   | GHI789        |  70000 |
|           4 | Sally       | Fields    | JKL012        |  75000 |
|           5 | Michael     | Smith     | MNO345        |  80000 |
|           6 | Emily       | Jones     | PQR678        |  85000 |
|           7 | David       | Williams  | STU901        |  90000 |
|           8 | Sarah       | Johnson   | VWX234        |  95000 |
|           9 | James       | Brown     | YZA567        | 100000 |
|          11 | William     | Davis     | EFG123        | 110000 |
|          12 | Olivia      | Garcia    | HIJ456        | 115000 |
|          13 | Christopher | Rodriguez | KLM789        | 120000 |
|          14 | Isabella    | Wilson    | NOP012        | 125000 |
|          15 | Matthew     | Martinez  | QRS345        | 130000 |
|          16 | Sophia      | Anderson  | TUV678        | 135000 |
|          17 | Daniel      | Smith     | WXY901        | 140000 |
|          18 | Mia         | Thomas    | ZAB234        | 145000 |
|          19 | Joseph      | Hernandez | CDE567        | 150000 |
|          20 | Abigail     | Smith     | FGH890        | 155000 |
|         101 | NULL        | NULL      | NULL          | 100001 |
+-------------+-------------+-----------+---------------+--------+
20 rows in set (0.00 sec)
```

10.To learn how to use various MySQL loop statements including while, repeat to run a block of code repeatedly based on a condition.

OUTPUT:

Program :-1

```
mysql> CREATE PROCEDURE test_mysql_while_loop()
    ->  BEGIN
    ->  DECLARE x  INT;
    ->  DECLARE str  VARCHAR(255);
    ->
    ->  SET x = 1;
    ->  SET str =  '';
    ->
    ->  WHILE x  <= 5 DO
    ->  SET  str = CONCAT(str,x,',');
    ->  SET  x = x + 1;
    ->  END WHILE;
    ->
    ->  SELECT str;
    ->  END
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> CALL test_mysql_while_loop() //
+------------+
| str        |
+------------+
| 1,2,3,4,5, |
+------------+
1 row in set (0.00 sec)

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

Program :-1

```
mysql> CREATE PROCEDURE test_mysql_while_loop()
    ->  BEGIN
    ->  DECLARE x  INT;
    ->  DECLARE str  VARCHAR(255);
    ->
    ->  SET x = 1;
    ->  SET str =  '';
    ->
    ->  WHILE x  <= 5 DO
    ->  SET  str = CONCAT(str,x,',');
    ->  SET  x = x + 1;
    ->  END WHILE;
    ->
    ->  SELECT str;
    ->  END
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> CALL test_mysql_while_loop() //
+------------+
| str        |
+------------+
| 1,2,3,4,5, |
+------------+
1 row in set (0.00 sec)

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

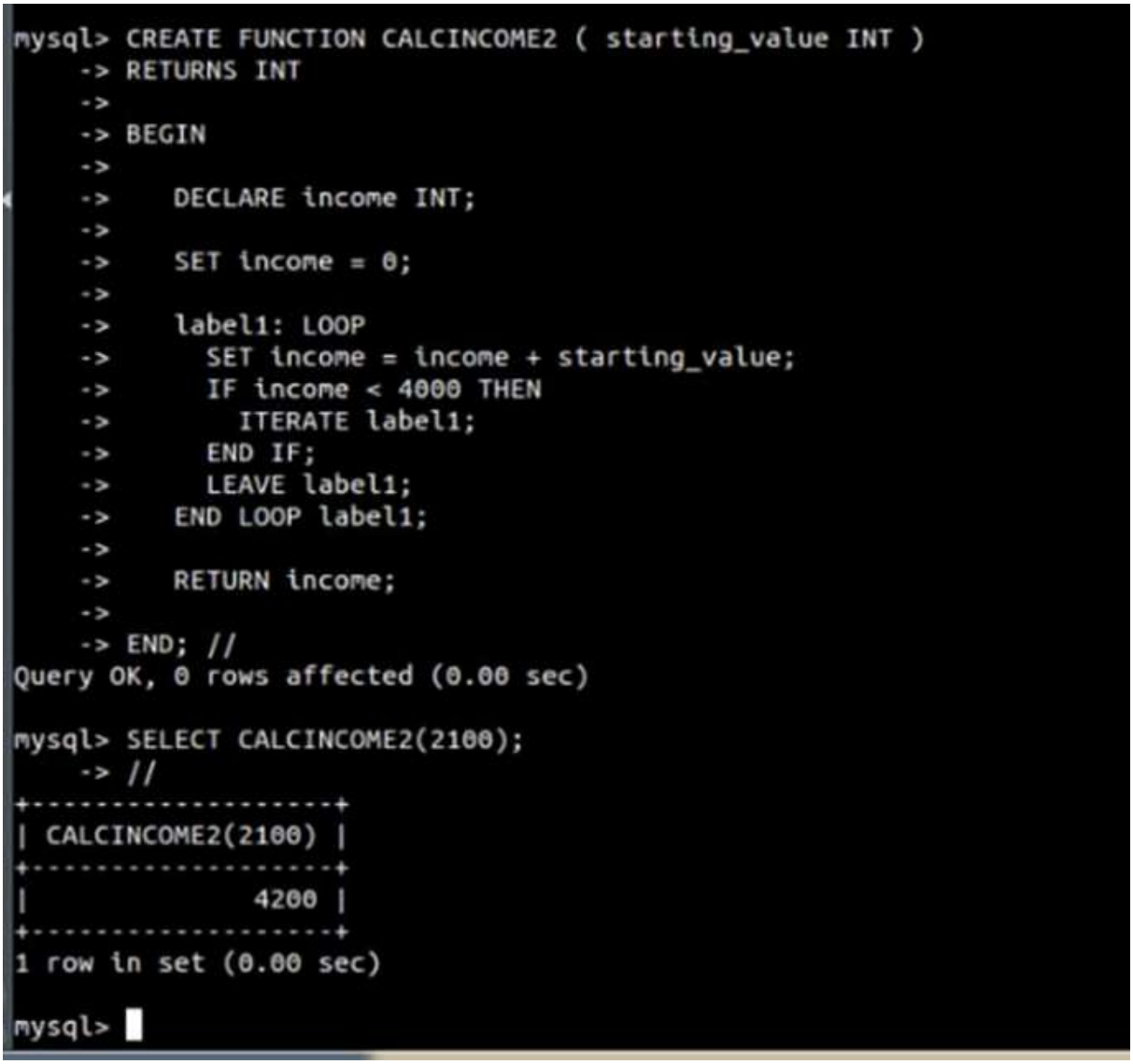
11.To learn how to use various MySQL loop statements including case and loop to run a block of code repeatedly based on a condition.

OUTPUT:

```
mysql> CREATE FUNCTION IncomeLevel ( monthly_value INT )
    -> RETURNS varchar(20)
    ->
    -> BEGIN
    ->
    ->    DECLARE income_level varchar(20);
    ->
    ->    CASE monthly_value
    ->       WHEN 4000 THEN
    ->         SET income_level = 'Low Income';
    ->
    ->       WHEN 5000 THEN
    ->         SET income_level = 'Avg Income';
    ->
    ->       ELSE
    ->         SET income_level = 'High Income';
    ->    END CASE;
    ->
    ->    RETURN income_level;
    ->
    -> END; //
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT INCOMELEVEL(5300); //
+------------------+
| INCOMELEVEL(5300) |
+------------------+
| High Income       |
+------------------+
1 row in set (0.00 sec)

mysql>
```



12.To learn how to use various TCL commands Commit, Rollback and Savepoint SQL commands

OUTPUT:

Questions:-

```
mysql> create table clss(name varchar(10),id int(5));
Query OK, 0 rows affected (0.19 sec)

mysql> insert into clss values("dj",5);
Query OK, 1 row affected (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.04 sec)

mysql> update class set name="bravo" where id="5";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> savepoint A;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into clss values("uppal",6);
Query OK, 1 row affected (0.00 sec)

mysql> savepoint B;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into clss values("balu",7);
Query OK, 1 row affected (0.00 sec)

mysql> savepoint C;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from clss;
+-------+------+
| name  | id   |
+-------+------+
| dj    |    5 |
| uppal |    6 |
| balu  |    7 |
+-------+------+
3 rows in set (0.00 sec)

mysql> ROLLBACK TO B;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from clss;
+-------+------+
| name  | id   |
+-------+------+
| dj    |    5 |
| uppal |    6 |
+-------+------+
2 rows in set (0.00 sec)

mysql> ROLLBACK TO A;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from clss;
+------+------+
| name | id   |
+------+------+
| dj   |    5 |
+------+------+
1 row in set (0.00 sec)
```

14.To learn how to use various DCL commands GRANT and REVOKE SQL commands

OUTPUT:

Program :-1

```
mysql> CREATE PROCEDURE student_info()
    -> select * from student_info;
Query OK, 0 rows affected (0.00 sec)

mysql> call student_info();
+-------+-------+-----------+---------+
| stuid | name  | area      | subject |
+-------+-------+-----------+---------+
|   201 | raj   | chennai   | dbms    |
|   202 | rahul | hyderabad | ooad    |
|   203 | rahim | mumbai    | java    |
|   204 | vikas | kochi     | python  |
+-------+-------+-----------+---------+
4 rows in set (0.00 sec)

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

Program :-2

```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetCustomerLevel(
    ->     in  p_customerNumber int(11),
    ->     out p_customerLevel  varchar(10))
    -> BEGIN
    ->     DECLARE creditlim double;
    ->
    ->     SELECT creditlimit INTO creditlim
    ->     FROM customers
    ->     WHERE customerNumber = p_custonerNunber;
    ->
    ->     IF creditlim > 50000 THEN
    ->  SET p_customerLevel = 'PLATINUM';
    ->     ELSEIF (creditlim <= 50000 AND creditlin >= 10000) THEN
    ->         SET p_customerLevel = 'GOLD';
    ->     ELSEIF creditlim < 10000 THEN
    ->         SET p_customerLevel = 'SILVER';
    ->     END IF;
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)
```

15.To implement Functions using program in MySQL.

OUTPUT:

```
mysql> DELIMITER //
mysql> CREATE FUNCTION CustomerLevel(p_CREDITLIMIT INT) RETURNS VARCHAR(10)
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE lvl VARCHAR(10);
    -> IF p_CREDITLIMIT > 50000 THEN
    -> SET lvl = 'PLATINUM';
    -> ELSEIF (p_CREDITLIMIT <= 50000 AND p_CREDITLIMIT >= 10000) THEN
    -> SET lvl = 'GOLD';
    -> ELSEIF p_CREDITLIMIT < 10000 THEN
    -> SET lvl = 'SILVER';
    -> END IF;
    -> RETURN (lvl);
    -> END
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT NAME,CustomerLevel(CREDITLIMIT)
    -> FROM CUSTOMER
    -> ORDER BY NAME
    -> //
ERROR 1054 (42S22): Unknown column 'NAME' in 'field list'
mysql> SELECT CNAME,CustomerLevel(CREDITLIMIT)
    -> FROM CUSTOMER
    -> ORDER BY NAME
    -> //
ERROR 1054 (42S22): Unknown column 'NAME' in 'order clause'
mysql> SELECT CNAME,CustomerLevel(CREDITLIMIT) FROM CUSTOMER ORDER BY CNAME//
+----------+---------------------------+
| CNAME    | CustomerLevel(CREDITLIMIT) |
+----------+---------------------------+
| DINESH   | GOLD                      |
| NAGENDRA | PLATINUM                  |
| RAJA     | GOLD                      |
| RAMU     | SILVER                    |
+----------+---------------------------+
4 rows in set (0.00 sec)
```

```
mysql> DELIMITER //
mysql> CREATE FUNCTION CustomerLevel(p_CREDITLIMIT INT) RETURNS VARCHAR(10)
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE lvl VARCHAR(10);
    -> IF p_CREDITLIMIT > 50000 THEN
    -> SET lvl = 'PLATINUM';
    -> ELSEIF (p_CREDITLIMIT <= 50000 AND p_CREDITLIMIT >= 10000) THEN
    -> SET lvl = 'GOLD';
    -> ELSEIF p_CREDITLIMIT < 10000 THEN
    -> SET lvl = 'SILVER';
    -> END IF;
    -> RETURN (lvl);
    -> END
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT NAME,CustomerLevel(CREDITLIMIT)
    -> FROM CUSTOMER
    -> ORDER BY NAME
    -> //
ERROR 1054 (42S22): Unknown column 'NAME' in 'field list'
mysql> SELECT CNAME,CustomerLevel(CREDITLIMIT)
    -> FROM CUSTOMER
    -> ORDER BY NAME
    -> //
```

16.Write a Cursor program using MySQL to retrieve the email-ids(build an email list) of employees from employees table.

OUTPUT:

Program :-1

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE build_email_list (INOUT email_list varchar(4000))
    -> BEGIN
    -> DECLARE v_finished INTEGER DEFAULT 0;
    ->     DECLARE v_email varchar(100) DEFAULT "";
    -> DECLARE email_cursor CURSOR FOR
    -> SELECT email FROM employees;
    -> DECLARE CONTINUE HANDLER FOR
    ->     NOT FOUND SET v_finished = 1;
    -> OPEN email_cursor;
    -> get_email:LOOP
    -> FETCH email_cursor INTO v_email;
    -> IF v_finished = 1 THEN
    -> LEAVE get_email;
    -> END IF;
    -> SET email_list = CONCAT(v_email,";",email_list);
    -> END LOOP get_email;
    -> CLOSE email_cursor;
    -> END $$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> SET @email_list = "";
Query OK, 0 rows affected (0.00 sec)

mysql> CALL build_email_list(@email_list);
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> select @email_list;
+-------------------------------------------------------------+
| @email_list                                                 |
+-------------------------------------------------------------+
| stony@marvel.com;kclark@decomics.com;pharry@warnerbros.com; |
+-------------------------------------------------------------+
1 row in set (0.00 sec)
```

17.To implement trigger in MySQL.

OUTPUT:

Program:-

```
mysql> DELIMITER //
mysql> CREATE TRIGGER before_student_update
    -> BEFORE UPDATE ON student
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO student_audit
    -> SET action ='update',
    -> student_id =OLD.id,
    -> lastname=OLD.Name,
    -> changedat = NOW();
    -> END //
Query OK, 0 rows affected (0.06 sec)

mysql> DELIMITER;
mysql> update student set name ='tony stark_c'Where id=3;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> select * from student_audit;
+----+------------+------------+---------------------+--------+
| id | student_id | lastname   | changedat           | action |
+----+------------+------------+---------------------+--------+
|  1 |          3 | tony stark | 2019-08-12 13:07:44 | update |
+----+------------+------------+---------------------+--------+
1 row in set (0.00 sec)
```

18.This function returns the numeric value of the leftmost character of the string str. Returns 0 if str is the empty string

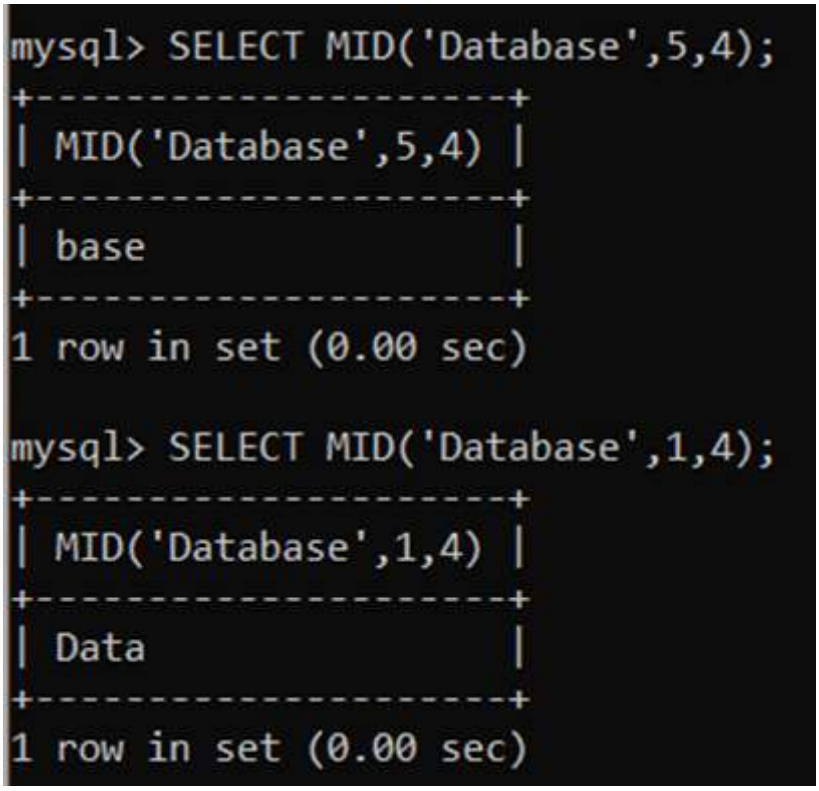
OUTPUT:

```
mysql> SELECT CONCAT('data','base','system');
+-----------------------------------+
| CONCAT('data','base','system') |
+-----------------------------------+
| databasesystem                    |
+-----------------------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT LEFT('Database', 5);
+----------------------+
| LEFT('Database', 5) |
+----------------------+
| Datab                |
+----------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT LPAD('Database',20,'$');
+--------------------------+
| LPAD('Database',20,'$') |
+--------------------------+
| $$$$$$$$$$Database    |
+--------------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT LPAD('Database',4,'$');
+------------------------+
| LPAD('Database',4,'$') |
+------------------------+
| Data                   |
+------------------------+
1 row in set (0.00 sec)
```


mysql> SELECT MID('Database',5,4);
+----------------------+
| MID('Database',5,4) |
+----------------------+
| base |
+----------------------+
1 row in set (0.00 sec)
mysql> SELECT MID('Database',1,4);
+----------------------+
| MID('Database',1,4) |
+----------------------+
| Data |
+----------------------+
1 row in set (0.00 sec)

```
mysql> select * from employees1;
+-------------+-------------+-----------+---------------+--------+
| employee_id | first_name  | last_name | device_serial | salary |
+-------------+-------------+-----------+---------------+--------+
|           1 | John        | Smith     | ABC123        |  60000 |
|           2 | Jane        | Doe       | DEF456        |  65000 |
|           3 | Bob         | Johnson   | GHI789        |  70000 |
|           4 | Sally       | Fields    | JKL012        |  75000 |
|           5 | Michael     | Smith     | MNO345        |  80000 |
|           6 | Emily       | Jones     | PQR678        |  85000 |
|           7 | David       | Williams  | STU901        |  90000 |
|           8 | Sarah       | Johnson   | VWX234        |  95000 |
|           9 | James       | Brown     | YZA567        | 100000 |
|          11 | William     | Davis     | EFG123        | 110000 |
|          12 | Olivia      | Garcia    | HIJ456        | 115000 |
|          13 | Christopher | Rodriguez | KLM789        | 120000 |
|          14 | Isabella    | Wilson    | NOP012        | 125000 |
|          15 | Matthew     | Martinez  | QRS345        | 130000 |
|          16 | Sophia      | Anderson  | TUV678        | 135000 |
|          17 | Daniel      | Smith     | WXY901        | 140000 |
|          18 | Mia         | Thomas    | ZAB234        | 145000 |
|          19 | Joseph      | Hernandez | CDE567        | 150000 |
|          20 | Abigail     | Smith     | FGH890        | 155000 |
|         101 | NULL        | NULL      | NULL          | 100001 |
+-------------+-------------+-----------+---------------+--------+
20 rows in set (0.14 sec)
```