DBMS EXPERIMENT 22:

Create a view with name 'std_view' using STUDENT table which holds the value of the following :

Reg_No	Stu_Name	Department	DBMS Mark
ECE0023	Reddy	Electronics	56
CSE0024	Sandy	Computer_Science	86
ECE0040	Hajira	Electronics	40

Perform the following

* which of the student scored low mark * List the stu_Name in Ascending Order

* Display the names Group by Department * list the students having the dept as "Electronics"

* List the students whose name ends with the substring "dy" * Find the students whose mark >50

* Drop the table

OUTPUT:

Query:-

```
mysql> create view stu_view as select regno,name and dob from stu;
Query OK, 0 rows affected (0.05 sec)

mysql> select * from stu_view;
+-----------+--------------+
| regno     | name and dob |
+-----------+--------------+
| 191811164 |            0 |
| 191811144 |            0 |
|  19181134 |            0 |
|  19181124 |            0 |
|  19181114 |            0 |
+-----------+--------------+
5 rows in set, 5 warnings (0.00 sec)
mysql> create index regno on stu(regno);
Query OK, 0 rows affected (0.36 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from stu;
+-------+------------+----------+--------------+-------------+-----------+-----------
-----------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinalit
ex_comment |
+-------+------------+----------+--------------+-------------+-----------+-----------
-----------+
| stu   |          1 | regno    |            1 | regno       | A         |
          |
+-------+------------+----------+--------------+-------------+-----------+-----------
-----------+
1 row in set (0.00 sec)
```

Progarm:-1

```
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 ='SLIVER';
    -> END IF;
    -> RETURN (lvl);
    -> END
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT NAME,customerLevel(CREDITLIMIT)
    -> FROM CUSTOMER
    -> ORDER BY NAME //
```

Progarm:-2

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE find_fact(IN n INT)
    -> BEGIN
    ->  SET @@GLOBAL.max_sp_recursion_depth=255;
    ->  SET @@session.max_sp_recursion_depth=255;
    ->  CALL factorial(n,@fact);
    ->  SELECT @fact;
    -> END
    -> $$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE PROCEDURE factorial(IN n INT,OUT fact INT)
    -> BEGIN
    ->  IF n=1 THEN
    ->     SET fact:=1;
    ->  ELSE
    ->    CALL factorial(n-1,fact);
    ->    SET fact:=n*fact;
    ->   END IF;
    -> END
    -> $$
Query OK, 0 rows affected (0.01 sec)

mysql> CALL find_fact(5);
    -> $$
+--------+
| @fact  |
+--------+
|    120 |
+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Progarm:-1

```
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 ='SLIVER';
    -> END IF;
    -> RETURN (lvl);
    -> END
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT NAME,customerLevel(CREDITLIMIT)
    -> FROM CUSTOMER
    -> ORDER BY NAME //
```

Progarm:-2

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE find_fact(IN n INT)
    -> BEGIN
    ->  SET @@GLOBAL.max_sp_recursion_depth=255;
    ->  SET @@session.max_sp_recursion_depth=255;
    ->  CALL factorial(n,@fact);
    ->  SELECT @fact;
    -> END
    -> $$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER $$
mysql> CREATE PROCEDURE factorial(IN n INT,OUT fact INT)
    -> BEGIN
    ->  IF n=1 THEN
    ->     SET fact:=1;
    ->  ELSE
    ->    CALL factorial(n-1,fact);
    ->    SET fact:=n*fact;
    ->   END IF;
    -> END
    -> $$
Query OK, 0 rows affected (0.01 sec)

mysql> CALL find_fact(5);
    -> $$
+--------+
| @fact  |
+--------+
|    120 |
+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 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
    -> //
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)
```