Practical No. 11

Aim: Implement set operations, case statement and view queries.

Theory:

1. **Set Operations**:
   - Set operations are fundamental database operations used to manipulate data.
   - Common set operations include union, intersection, difference, and Cartesian product.
   - These operations help combine or compare data from one or more database tables.
2. **Case Statement**:
   - A case statement is a conditional statement used in SQL to perform conditional logic.
   - It allows you to perform different actions based on specified conditions.
   - Commonly used for data transformation and customization in query results.
3. **View Queries**:
   - A view is a virtual table created by a query that can be used like a regular table.
   - View queries allow you to encapsulate complex queries into reusable objects.
   - They enhance security by limiting direct access to underlying tables and simplify query composition.

Queries:

(1) Implement “IF” Condition in Query

a) Put if condition on “price” attribute (IF Else)

```
mysql> SELECT IF((SELECT PRICE FROM product WHERE PRODUCT_NO=120) = (SELECT PRICE FROM product WHERE PRODUCT_NO=122),'YES','NO')AS IF_STATEMENT;
+--------------+
| IF_STATEMENT |
+--------------+
| NO           |
+--------------+
1 row in set (0.01 sec)

mysql> /*202203103510203*/
```

b) Try nested IF on the “price” attribute.

c) Display Price and quantity and there rating with “High” , “Medium” and “Low”

```
mysql> /*202203103510124*/
mysql> SELECT
    ->     IF(
    ->         (SELECT MAX(PRICE) FROM PRODUCT WHERE PRODUCT_NO = 120) = (SELECT MAX(PRICE) FROM PRODUCT WHERE PRODUCT_NO = 122),
    ->         'YES',
    ->         IF(
    ->             (SELECT MAX(PRICE) FROM PRODUCT WHERE PRODUCT_NO = 120) < (SELECT MAX(PRICE) FROM PRODUCT WHERE PRODUCT_NO = 122),
    ->             'LOW',
    ->             'HIGH'
    ->         )
    ->     ) AS NESTED_IF_STATEMENT;
+---------------------+
| NESTED_IF_STATEMENT |
+---------------------+
| HIGH                |
+---------------------+
```

(2) Create a view VProduct of the product‟s id, description and price.

```
mysql> /*202203103510124*/
mysql> CREATE VIEW VPRODUCT AS SELECT PRODUCT_NO , DESCRIPTION, PRICE FROM PRODUCT;
Query OK, 0 rows affected (0.04 sec)

mysql> SELECT * FROM VPRODUCT;
+------------+-----------------+----------+
| PRODUCT_NO | DESCRIPTION     | PRICE    |
+------------+-----------------+----------+
|        120 | REDUCER         |  1200.00 |
|        121 | PLATE           |  2000.00 |
|        122 | HANDLE          |   700.00 |
|        124 | WIDGET REMOVER  |   900.00 |
|        136 | SIZE WIDGET     |  1000.00 |
|        137 | SIZE WIDGET     | 15000.00 |
+------------+-----------------+----------+
```

(3) Create a view of Vorder to get orders (order_id, product_id, description, customer_name,quantity) placed by customer who belongs to “BRIXTON”.

```
mysql> CREATE VIEW VORDER AS SELECT O.CORDER_NO AS ORDER_ID, OL.PRODUCT_NO, P.DESCRIPTION, C.NAME A
NE OL ON O.CORDER_NO = OL.CORDER_NO JOIN PRODUCT P ON OL.PRODUCT_NO = P.PRODUCT_NO JOIN CUSTOMER C
RIXTON';
Query OK, 0 rows affected (0.04 sec)

mysql> /*202203103510124*/
mysql> CREATE VIEW VORDER AS SELECT O.CORDER_NO AS ORDER_ID, OL.PRODUCT_NO, P.DESCRIPTION, C.NAME A
NE OL ON O.CORDER_NO = OL.CORDER_NO JOIN PRODUCT P ON OL.PRODUCT_NO = P.PRODUCT_NO JOIN CUSTOMER C
RIXTON';
ERROR 1050 (42S01): Table 'VORDER' already exists
mysql> SELECT * FROM VORDER;
+----------+------------+-------------+---------------+----------+
| ORDER_ID | PRODUCT_NO | DESCRIPTION | CUSTOMER_NAME | QUANTITY |
+----------+------------+-------------+---------------+----------+
|      203 |        122 | HANDLE      | DRAKE         |       20 |
|      204 |        136 | SIZE WIDGET | GARRY SMITH   |       30 |
+----------+------------+-------------+---------------+----------+
```

(4) Insert a new row in VProduct 135, „Sofa“ and 35000.

```
mysql> /*202203103510124*/
mysql> INSERT INTO VProduct (PRODUCT_NO, DESCRIPTION, PRICE)
    -> VALUES (135, 'Sofa', 35000);
Query OK, 1 row affected (0.06 sec)

mysql> SELECT * FROM VProduct;
+------------+----------------+----------+
| PRODUCT_NO | DESCRIPTION    | PRICE    |
+------------+----------------+----------+
|        120 | REDUCER        |  1200.00 |
|        121 | PLATE          |  2000.00 |
|        122 | HANDLE         |   700.00 |
|        124 | WIDGET REMOVER |   900.00 |
|        135 | Sofa           | 35000.00 |
|        136 | SIZE WIDGET    |  1000.00 |
|        137 | SIZE WIDGET    | 15000.00 |
+------------+----------------+----------+
```

(5) Update product‟s quantity to 25 which is brought by customer „DRAKE‟ in Vorder.

```
mysql> UPDATE VORDER SET QUANTITY = 25 WHERE CUSTOMER_NAME = 'DRAKE';
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM VORDER;
+----------+------------+-------------+---------------+----------+
| ORDER_ID | PRODUCT_NO | DESCRIPTION | CUSTOMER_NAME | QUANTITY |
+----------+------------+-------------+---------------+----------+
|      203 |        122 | HANDLE      | DRAKE         |       25 |
|      204 |        136 | SIZE WIDGET | GARRY SMITH   |       30 |
+----------+------------+-------------+---------------+----------+
2 rows in set (0.00 sec)

mysql> /*202203103510124*/
```

(6) Delete details of product id 121 from VProduct.

```
mysql> DELETE FROM VProduct WHERE PRODUCT_NO = 121;
Query OK, 1 row affected (0.05 sec)

mysql> SELECT * FROM VProduct;
+------------+----------------+----------+
| PRODUCT_NO | DESCRIPTION    | PRICE    |
+------------+----------------+----------+
|        120 | REDUCER        |  1200.00 |
|        122 | HANDLE         |   700.00 |
|        124 | WIDGET REMOVER |   900.00 |
|        135 | Sofa           | 35000.00 |
|        136 | SIZE WIDGET    |  1000.00 |
|        137 | SIZE WIDGET    | 15000.00 |
+------------+----------------+----------+
6 rows in set (0.00 sec)

mysql> /*202203103510124*/
```

(7) Delete view VProduct.

```
mysql> /*202203103510124*/
mysql> DROP VIEW VProduct;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM VProduct;
ERROR 1146 (42S02): Table 'Tutorial1_Ankit.VProduct' doesn't exist
```

(8) Display the name of all customers and all suppliers with their id by using union operator.

```
mysql> /*202203103510124*/
mysql> SELECT NAME, CUSTOMER_NO
    -> FROM CUSTOMER
    -> UNION
    -> SELECT NAME, SUPPLIER_NO
    -> FROM SUPPLIER;
+--------------+-------------+
| NAME         | CUSTOMER_NO |
+--------------+-------------+
| GARRY SMITH  |          10 |
| PATEL        |          20 |
| DRAKE        |          30 |
| BOB SMITH    |          40 |
| JAMES        |          50 |
| NORTON       |          60 |
| JOHN MICHAEL |          70 |
| MICHAEL      |        1001 |
| RINGWORLD    |        1002 |
| BABYLON      |        1003 |
| JOHN         |        1004 |
| SMITH        |        1005 |
+--------------+-------------+
```

(9) List products which are not bought by any customer using the minus operator.

```
mysql> /*202203103510124*/
mysql> SELECT PRODUCT_NO, DESCRIPTION
    -> FROM PRODUCT
    -> MINUS
    -> SELECT PRODUCT_NO, DESCRIPTION
    -> FROM Vorder;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use n
ear 'SELECT PRODUCT_NO, DESCRIPTION
FROM Vorder' at line 4
```

(10) Give the name of suppliers who are also customers.

```
mysql> /*202203103510124*/
mysql> SELECT DISTINCT S.NAME AS SUPPLIER_NAME
    -> FROM SUPPLIER S
    -> JOIN CUSTOMER C ON S.NAME = C.NAME;
Empty set (0.00 sec)
```

Conclusion:

Incorporated set operations, case statements, and view queries to enhance data manipulation and retrieval capabilities. These features have significantly improved the flexibility and efficiency of database queries, allowing for more complex and customized data processing.