SQL Assignment: products Table

```
mysql> CREATE DATABASE ASSIGN_1;
Query OK, 1 row affected (0.05 sec)
```

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| assign_1           |
| courier_management |
| elakkiya           |
| hospital_management |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| virtualartgallery  |
| world              |
+--------------------+
11 rows in set (0.01 sec)
```

Created a database named assign_1.

Task 1: Create a Table Create a table named products with the following columns:

product_id (INT, Primary Key)

product_name (VARCHAR)

category (VARCHAR)

price (DECIMAL)

stock_quantity (INT)

added_date (DATE)

```
mysql> USE ASSIGN_1;
Database changed
mysql> CREATE TABLE PRODUCTS(
    -> PRODUCT_ID INT PRIMARY KEY,
    -> PRODUCT_NAME VARCHAR (100),
    -> CATEGORY VARCHAR (100),
    -> PRICE DECIMAL(7,2),
    -> STOCK_QTY INT,
    -> ADDED_DATE DATE);
Query OK, 0 rows affected (0.07 sec)

mysql> DESC PRODUCTS;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| PRODUCT_ID   | int          | NO   | PRI | NULL    |       |
| PRODUCT_NAME | varchar(100) | YES  |     | NULL    |       |
| CATEGORY     | varchar(100) | YES  |     | NULL    |       |
| PRICE        | decimal(7,2) | YES  |     | NULL    |       |
| STOCK_QTY    | int          | YES  |     | NULL    |       |
| ADDED_DATE   | date         | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

Task 2: Insert Records Insert at least 5 different products, each with a unique category and price range. Use realistic product names (e.g., headphones, mouse, laptop, etc.).

```
mysql> INSERT INTO PRODUCTS VALUES
    -> (1, 'MOUSE', 'ELECTRONICS', 400.00, 7, '2025-05-30'),
    -> (2, 'HEADPHONE', 'ELECTRONICS', 900.00, 5, '2025-06-19'),
    -> (3, 'YELLOW SALWAR', 'CLOTHING', 650.00, 3, '2025-7-08'),
    -> (4, 'SPEAKER', 'ELECTRONICS', 860.00, 12, '2025-06-15'),
    -> (5, 'BAGGY JEAN', 'CLOTHING', 1020.00, 6, '2025-07-11');
Query OK, 5 rows affected (0.04 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Task 3: Write Queries

1. List all products.

```
mysql> SELECT * FROM PRODUCTS;
+------------+--------------+-------------+---------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE   | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+---------+-----------+------------+
|          1 | MOUSE        | ELECTRONICS |  400.00 |         7 | 2025-05-30 |
|          2 | HEADPHONE    | ELECTRONICS |  900.00 |         5 | 2025-06-19 |
|          3 | YELLOW SALWAR | CLOTHING    |  650.00 |         3 | 2025-07-08 |
|          4 | SPEAKER      | ELECTRONICS |  860.00 |        12 | 2025-06-15 |
|          5 | BAGGY JEAN   | CLOTHING    | 1020.00 |         6 | 2025-07-11 |
+------------+--------------+-------------+---------+-----------+------------+
5 rows in set (0.01 sec)
```

2. Display only product_name and price .

```
mysql> SELECT PRODUCT_NAME, PRICE FROM PRODUCTS;
+---------------+---------+
| PRODUCT_NAME  | PRICE   |
+---------------+---------+
| MOUSE         |  400.00 |
| HEADPHONE     |  900.00 |
| YELLOW SALWAR |  650.00 |
| SPEAKER       |  860.00 |
| BAGGY JEAN    | 1020.00 |
+---------------+---------+
5 rows in set (0.00 sec)
```

3. Find products with stock_quantity less than 10.

```
mysql> SELECT * FROM PRODUCTS WHERE STOCK_QTY < 10;
+------------+--------------+-------------+---------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE   | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+---------+-----------+------------+
|          1 | MOUSE        | ELECTRONICS |  400.00 |         7 | 2025-05-30 |
|          2 | HEADPHONE    | ELECTRONICS |  900.00 |         5 | 2025-06-19 |
|          3 | YELLOW SALWAR | CLOTHING    |  650.00 |         3 | 2025-07-08 |
|          5 | BAGGY JEAN   | CLOTHING    | 1020.00 |         6 | 2025-07-11 |
+------------+--------------+-------------+---------+-----------+------------+
4 rows in set (0.01 sec)
```

4. Find products with price between 500 and 2000.

```
mysql> SELECT * FROM PRODUCTS WHERE PRICE BETWEEN 500 AND 2000;
+------------+--------------+-------------+---------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE   | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+---------+-----------+------------+
|          2 | HEADPHONE    | ELECTRONICS |  900.00 |         5 | 2025-06-19 |
|          3 | YELLOW SALWAR | CLOTHING    |  650.00 |         3 | 2025-07-08 |
|          4 | SPEAKER      | ELECTRONICS |  860.00 |        12 | 2025-06-15 |
|          5 | BAGGY JEAN   | CLOTHING    | 1020.00 |         6 | 2025-07-11 |
+------------+--------------+-------------+---------+-----------+------------+
4 rows in set (0.01 sec)
```

5. Show products added after 2025-07-01 .

```
mysql> SELECT * FROM PRODUCTS WHERE ADDED_DATE > '2025-07-01';
+------------+--------------+----------+---------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY | PRICE   | STOCK_QTY | ADDED_DATE |
+------------+--------------+----------+---------+-----------+------------+
|          3 | YELLOW SALWAR | CLOTHING |  650.00 |         3 | 2025-07-08 |
|          5 | BAGGY JEAN   | CLOTHING | 1020.00 |         6 | 2025-07-11 |
+------------+--------------+----------+---------+-----------+------------+
2 rows in set (0.01 sec)
```

6. List all products whose names start with ‘S’.

```
mysql> SELECT * FROM PRODUCTS WHERE PRODUCT_NAME LIKE 'S%';
+------------+--------------+-------------+--------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE  | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+--------+-----------+------------+
|          4 | SPEAKER      | ELECTRONICS | 860.00 |        12 | 2025-06-15 |
+------------+--------------+-------------+--------+-----------+------------+
1 row in set (0.01 sec)
```

7. Show all products that belong to either Electronics or Furniture .

```
mysql> SELECT * FROM PRODUCTS WHERE CATEGORY IN ('ELECTRONICS' , 'FURNITURE');
+------------+--------------+-------------+--------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE  | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+--------+-----------+------------+
|          1 | MOUSE        | ELECTRONICS | 400.00 |         7 | 2025-05-30 |
|          2 | HEADPHONE    | ELECTRONICS | 900.00 |         5 | 2025-06-19 |
|          4 | SPEAKER      | ELECTRONICS | 860.00 |        12 | 2025-06-15 |
+------------+--------------+-------------+--------+-----------+------------+
3 rows in set (0.01 sec)
```

Task 4: Update & Delete

1. Update the price of one product.

```
mysql> UPDATE PRODUCTS SET PRICE = 450.00 WHERE PRODUCT_NAME = 'MOUSE';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM PRODUCTS;
+------------+--------------+-------------+---------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE   | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+---------+-----------+------------+
|          1 | MOUSE        | ELECTRONICS |  450.00 |         7 | 2025-05-30 |
|          2 | HEADPHONE    | ELECTRONICS |  900.00 |         5 | 2025-06-19 |
|          3 | YELLOW SALWAR | CLOTHING    |  650.00 |         3 | 2025-07-08 |
|          4 | SPEAKER      | ELECTRONICS |  860.00 |        12 | 2025-06-15 |
|          5 | BAGGY JEAN   | CLOTHING    | 1020.00 |         6 | 2025-07-11 |
+------------+--------------+-------------+---------+-----------+------------+
5 rows in set (0.00 sec)
```

2. Increase stock of all products in a specific category by 5.

```
mysql> UPDATE PRODUCTS SET STOCK_QTY = STOCK_QTY+5 WHERE CATEGORY = 'CLOTHING';
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT * FROM PRODUCTS;
+------------+--------------+-------------+---------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE   | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+---------+-----------+------------+
|          1 | MOUSE        | ELECTRONICS |  450.00 |         7 | 2025-05-30 |
|          2 | HEADPHONE    | ELECTRONICS |  900.00 |         5 | 2025-06-19 |
|          3 | YELLOW SALWAR | CLOTHING    |  650.00 |         8 | 2025-07-08 |
|          4 | SPEAKER      | ELECTRONICS |  860.00 |        12 | 2025-06-15 |
|          5 | BAGGY JEAN   | CLOTHING    | 1020.00 |        11 | 2025-07-11 |
+------------+--------------+-------------+---------+-----------+------------+
5 rows in set (0.00 sec)
```

3. Delete one product based on its product_id .

```
mysql> DELETE FROM PRODUCTS WHERE PRODUCT_ID = 3;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM PRODUCTS;
+------------+--------------+-------------+---------+-----------+------------+
| PRODUCT_ID | PRODUCT_NAME | CATEGORY    | PRICE   | STOCK_QTY | ADDED_DATE |
+------------+--------------+-------------+---------+-----------+------------+
|          1 | MOUSE        | ELECTRONICS |  450.00 |         7 | 2025-05-30 |
|          2 | HEADPHONE    | ELECTRONICS |  900.00 |         5 | 2025-06-19 |
|          4 | SPEAKER      | ELECTRONICS |  860.00 |        12 | 2025-06-15 |
|          5 | BAGGY JEAN   | CLOTHING    | 1020.00 |        11 | 2025-07-11 |
+------------+--------------+-------------+---------+-----------+------------+
4 rows in set (0.00 sec)
```

The product with product id 3 is deleted.

4. Delete all products with stock_quantity = 0

```
mysql> DELETE FROM PRODUCTS WHERE STOCK_QTY = 0;
Query OK, 0 rows affected (0.00 sec)
```

There is no product with stock quantity 0.