DAY-01

SQL Assignment 1: products Table

 SQL Assignment: `products` Table

 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)

 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.).

 Task 3: Write Queries

1. List all products.
2. Display only `product_name` and `price`.
3. Find products with `stock_quantity` less than 10.
4. Find products with `price` between 500 and 2000.
5. Show products added after `2023-01-01`.
6. List all products whose names start with 'S'.
7. Show all products that belong to either `Electronics` or `Furniture`.

 Task 4: Update & Delete

1. Update the price of one product.
2. Increase stock of all products in a specific category by 5.
3. Delete one product based on its `product_id`.
4. Delete all products with `stock_quantity = 0`.

TASK – 01: Create a Table

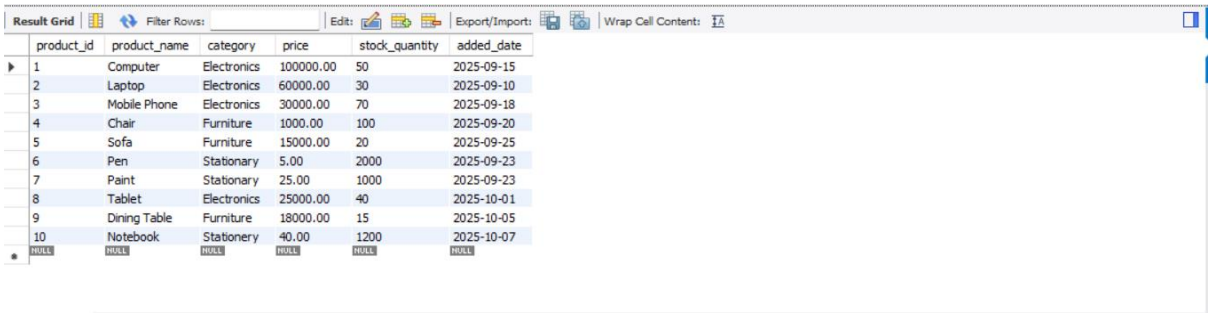
```
CREATE DATABASE sample;

USE sample;

CREATE TABLE products(

product_id INT PRIMARY KEY,

product_name VARCHAR(100),

category VARCHAR(100),

price DECIMAL(10,2),

stock_quantity INT,

added_date DATE);
```

TASK – 02: Insert Records

```
INSERT INTO products VALUES
(1, 'Computer', 'Electronics', 100000.00, 50, '2025-09-15' ),
(2, 'Laptop', 'Electronics', 60000.00, 30, '2025-09-10' ),
(3, 'Mobile Phone', 'Electronics', 30000.00, 70, '2025-09-18'),
(4, 'Chair', 'Furniture', 1000.00, 100, '2025-09-20'),
(5, 'Sofa', 'Furniture', 15000.00, 20, '2025-09-25'),
(6, 'Pen', 'Stationary', 5.00, 2000, '2025-09-23'),
(7, 'Paint', 'Stationary', 25.00, 1000, '2025-09-23'),
(8, 'Tablet', 'Electronics', 25000.00, 40, '2025-10-01'),
(9, 'Dining Table', 'Furniture', 18000.00, 15, '2025-10-05'),
(10, 'Notebook', 'Stationery', 40.00, 1200, '2025-10-07');
```

TASK – 03: Write Queries

1. SELECT * FROM products;



product_id	product_name	category	price	stock_quantity	added_date
1	Computer	Electronics	100000.00	50	2025-09-15
2	Laptop	Electronics	60000.00	30	2025-09-10
3	Mobile Phone	Electronics	30000.00	70	2025-09-18
4	Chair	Furniture	1000.00	100	2025-09-20
5	Sofa	Furniture	15000.00	20	2025-09-25
6	Pen	Stationary	5.00	2000	2025-09-23
7	Paint	Stationary	25.00	1000	2025-09-23
8	Tablet	Electronics	25000.00	40	2025-10-01
9	Dining Table	Furniture	18000.00	15	2025-10-05
10	Notebook	Stationery	40.00	1200	2025-10-07
NULL	NULL	NULL	NULL	NULL	NULL

2. SELECT product_name, price FROM products;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	price
Computer	100000.00
Laptop	60000.00
Mobile Phone	30000.00
Chair	1000.00
Sofa	15000.00
Pen	5.00
Paint	25.00
Tablet	25000.00
Dining Table	18000.00
Notebook	40.00

3. SELECT * FROM products WHERE stock_quantity < 10;

Result Grid | Filter Rows: | Edit: | Export/Import:

product_id	product_name	category	price	stock_quantity	added_date
11	Smartwatch	Electronics	12000.00	5	2025-10-10
13	Marker Set	Stationery	150.00	7	2025-10-15
NULL	NULL	NULL	NULL	NULL	NULL

Stationery

4. SELECT * FROM products WHERE price BETWEEN 500 AND 2000;

product_id	product_name	category	price	stock_quantity	added_date
4	Chair	Furniture	1000.00	100	2025-09-20
NULL	NULL	NULL	NULL	NULL	NULL

5. SELECT * FROM products WHERE added_date > '2023-01-01';

product_id	product_name	category	price	stock_quantity	added_date
1	Computer	Electronics	100000.00	50	2025-09-15
2	Laptop	Electronics	60000.00	30	2025-09-10
3	Mobile Phone	Electronics	30000.00	70	2025-09-18
4	Chair	Furniture	1000.00	100	2025-09-20
5	Sofa	Furniture	15000.00	20	2025-09-25
6	Pen	Stationary	5.00	2000	2025-09-23
7	Paint	Stationary	25.00	1000	2025-09-23
8	Tablet	Electronics	25000.00	40	2025-10-01
9	Dining Table	Furniture	18000.00	15	2025-10-05
10	Notebook	Stationery	40.00	1200	2025-10-07
11	Smartwatch	Electronics	12000.00	5	2025-10-10
12	Bookshelf	Furniture	7000.00	10	2025-10-12
13	Marker Set	Stationery	150.00	7	2025-10-15
NULL	NULL	NULL	NULL	NULL	NULL

6. SELECT * FROM products WHERE product_name LIKE 'S%';

product_id	product_name	category	price	stock_quantity	added_date
5	Sofa	Furniture	15000.00	20	2025-09-25
11	Smartwatch	Electronics	12000.00	5	2025-10-10
NULL	NULL	NULL	NULL	NULL	NULL

7. SELECT * FROM products WHERE category IN ('Electronics', 'Furniture');

product_id	product_name	category	price	stock_quantity	added_date
5	Sofa	Furniture	15000.00	20	2025-09-25
11	Smartwatch	Electronics	12000.00	5	2025-10-10
NULL	NULL	NULL	NULL	NULL	NULL

TASK – 04: Update & Delete:

1. UPDATE products SET price = 10000.00 WHERE product_id = 5;

2. UPDATE products SET stock_quantity = stock_quantity + 5 WHERE category = 'Electronics';

3. DELETE FROM products WHERE product_id = 3;

4. DELETE FROM products WHERE stock_quantity = 0;

SQL Assignment 2: Tables & Insert Statements

Tables & Insert Statements

1. departments Table

```
CREATE TABLE departments (
  dept_id INT PRIMARY KEY,
  dept_name VARCHAR(100)
);
```

```
INSERT INTO departments VALUES
(1, 'Human Resources'),
(2, 'Engineering'),
(3, 'Marketing');
```

2. employees Table

```
CREATE TABLE employees (
  emp_id INT PRIMARY KEY,
  emp_name VARCHAR(100),
  dept_id INT,
  salary INT
);
```

```
INSERT INTO employees VALUES
(101, 'Amit Sharma', 1, 30000),
(102, 'Neha Reddy', 2, 45000),
(103, 'Faizan Ali', 2, 48000),
(104, 'Divya Mehta', 3, 35000),
(105, 'Ravi Verma', NULL, 28000);
```

JOIN-Based Questions

1. Show all employees with their department names.
2. List employees who **do not belong** to any department.
3. Display the total number of employees in each department.
4. Show departments with **no employees**.
5. List employee names and department names for those who earn more than 40,000.

1.Show all employees with their department names.

```
SELECT e.emp_name, d.dept_name
FROM employees e
JOIN departments d
ON e.dept_id = d.dept_id;
```

	emp_name	dept_name
▶	Amit Sharma	Human Resourses
	Neha Reddy	Engineering
	Faizan Ali	Engineering
	Divya Mehta	Marketing

2.List employees who do not belong to any department.

```
SELECT e.emp_name
FROM employees e
LEFT JOIN departments d
ON e.dept_id = d.dept_id
WHERE e.dept_id IS NULL;
```

	emp_name
▶	Ravi Verma

3.Display the total number of employees in each department.

```
SELECT d.dept_name, COUNT(e.emp_id) AS total_no_employees
FROM departments d
JOIN employees e
ON e.dept_id = d.dept_id
GROUP BY dept_name;
```

	dept_name	total_no_employees
▶	Human Resourses	1
	Engineering	2
	Marketing	1

4.Show departments with no employees.

```
SELECT d.dept_name
FROM departments d
LEFT JOIN employees e
ON e.dept_id = d.dept_id
WHERE e.emp_id IS NULL;
```

5.List employee names and department names for those who earn more than 40,000.

```
SELECT e.emp_name, d.dept_name
FROM employees e
JOIN departments d
ON e.dept_id = d.dept_id
WHERE e.salary > 40000;
```

emp_name	dept_name
Neha Reddy	Engineering
Faizan Ali	Engineering