Name: Simantini M. Patil                                        Data Engineering Batch 2

# CRUD OPERATIONS

```
------Creation of table Product------
Use appdb;

CREATE TABLE Product (

   ProductID INT PRIMARY KEY,

   ProductName VARCHAR(100),

   Category VARCHAR(50),

   Price INT,

   StockQuantity INT,

   Supplier VARCHAR(100)

);

select * from Product;  ---View table---


----Inserting the values in table----

INSERT INTO Product (ProductID, ProductName, Category, Price, StockQuantity, Supplier)
VALUES

(1, 'Laptop', 'Electronics', 70000, 50, 'TechMart'),

(2, 'Office Chair', 'Furniture', 5000, 100, 'HomeComfort'),

(3, 'Smartwatch', 'Electronics', 15000, 200, 'GadgetHub'),

(4, 'Desk Lamp', 'Lighting', 1200, 300, 'BrightLife'),

(5, 'Wireless Mouse', 'Electronics', 1500, 250, 'GadgetHub');

select * from Product;    ---View table---


----1. CRUD Operations----

--- Query 1. Insert a product named "Gaming Keyboard" under the "Electronics" category,
priced at 3500, with 150 units in stock, supplied by "TechMart".---

INSERT INTO Product (ProductID, ProductName, Category, Price, StockQuantity, Supplier)
VALUES

(6, 'Gaming Keyboard', 'Electronics', 3500, 150, 'TechMart');

select * from Product;    ---View table---
```

```sql
---Query 2. Increase the price of all Electronics products by 10%.---
Update Product
set price = Price * 1.10
where Category='Electronics';
select * from Product;     ---View table---

---Query 3. Remove the product with the ProductID = 4 (Desk Lamp).---
Delete from Product
where ProductID = 4;

---Query 4. Display all products sorted by Price in descending order.---
SELECT * FROM Product
ORDER BY Price DESC;
----2. Sorting and Filtering----
--- Query 1. Sort products by stock quantity (ascending)---
SELECT * FROM Product
ORDER BY StockQuantity ASC;

-- Query 2. Filter products by category (Electronics)---
SELECT * FROM Product
WHERE Category = 'Electronics';

--- Query 3. Filter Electronics products priced above 5000---
SELECT * FROM Product
WHERE Category = 'Electronics' AND Price > 5000;

--Query 4. Filter Electronics OR products priced below 2000---
SELECT * FROM Product
WHERE Category = 'Electronics' OR Price < 2000;
----3. Aggregation and Grouping----
-- Query 1. Total stock value (Price × StockQuantity)---
```

```sql
SELECT SUM(Price * StockQuantity) AS TotalStockValue
FROM Product;

-- Query 2. Average price by category---
SELECT Category, AVG(Price) AS AveragePrice
FROM Product
GROUP BY Category;

-- Query 3. Total number of products supplied by GadgetHub---
SELECT COUNT(*) AS TotalProducts
FROM Product
WHERE Supplier = 'GadgetHub';
----4. Conditional and Pattern Matching----
---Query 1. Products whose name contains 'Wireless'---
SELECT * FROM Product
WHERE ProductName LIKE '%Wireless%';

--Query 2. Products supplied by TechMart or GadgetHub---
SELECT * FROM Product
WHERE Supplier IN ('TechMart', 'GadgetHub');

---Query 3. Products with price between 1000 and 20000---
SELECT * FROM Product
WHERE Price BETWEEN 1000 AND 20000;
----5. Advanced Queries----
---Query 1. Products with stock greater than average stock---
SELECT * FROM Product
WHERE StockQuantity > (
   SELECT AVG(StockQuantity) FROM Product
);
```

```sql
---Query 2. Top 3 most expensive products---
SELECT * FROM Product
ORDER BY Price DESC
LIMIT 3;


---Query 3. Duplicate supplier names---
SELECT Supplier, COUNT(*) AS Count
FROM Product
GROUP BY Supplier
HAVING COUNT(*) > 1;


--Query 4. Summary: number of products and total stock value per category---
SELECT Category, COUNT(*) AS ProductCount,
       SUM(Price * StockQuantity) AS TotalStockValue
FROM Product
GROUP BY Category;
----6. Join and Subqueries (based on Product table)----
---Query 1. Supplier who provides the maximum number of products---
SELECT Supplier, COUNT(*) AS ProductCount
FROM Product
GROUP BY Supplier
ORDER BY ProductCount DESC
LIMIT 1;
---Query 2. Most expensive product per category---
SELECT Category, ProductName, Price
FROM Product p
WHERE Price = (
    SELECT MAX(Price)
    FROM Product
    WHERE Category = p.Category);
```