CREATE TABLE Products ( product\_id INT PRIMARY KEY,

product\_name VARCHAR(100),

category VARCHAR(50),

unit\_price DECIMAL(10, 2)

);

INSERT INTO Products (product\_id, product\_name, category, unit\_price) VALUES (101, 'Laptop', 'Electronics', 500.00),

(102, 'Smartphone', 'Electronics', 300.00),

(103, 'Headphones', 'Electronics', 30.00),

(104, 'Keyboard', 'Electronics', 20.00),

(105, 'Mouse', 'Electronics', 15.00);

1. **Retrieve all columns from the product table.**

select \* from products;

1. **Retrieve the product\_name and unit\_price from the Products table.**

select product\_name,unit\_price from products;

1. **Filter the Products table to show only products in the 'Electronics' category.**

select \* from products where category='Electronics';

1. **Retrieve the product\_id and product\_name from the Products table for products with a unit\_price greater than $100.**

select \* from products where unit\_price >100;

1. **Calculate the average unit\_price of products in the Products table.**

SELECT AVG(unit\_price) AS average\_price FROM Products;

1. **Retrieve product\_name and unit\_price from the Products table with the Highest Unit Price**

SELECT product\_name, unit\_price FROM Products ORDER BY unit\_price DESC LIMIT 1;

1. **Retrieve the product\_name and unit\_price from the Products table, ordering the results by unit\_price in descending order.**

SELECT product\_name, unit\_price FROM Products ORDER BY unit\_price DESC;

1. **Retrieve the product\_name and unit\_price from the Products table, filtering the unit\_price to show only values between $20 and $600.**

SELECT product\_name, unit\_price FROM Products WHERE unit\_price BETWEEN 20 AND 600;

1. **Retrieve the product\_name and category from the Products table, ordering the results by category in ascending order.**

SELECT product\_name, category FROM Products ORDER BY category ASC;

SALES TABLE

CREATE TABLE Sales (

sale\_id INT PRIMARY KEY, product\_id INT, quantity\_sold INT,

sale\_date DATE,

total\_price DECIMAL(10, 2)

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

**);**

INSERT INTO Sales (sale\_id, product\_id, quantity\_sold, sale\_date, total\_price) VALUES (1, 101, 5, '2024-01-01', 2500.00),

(2, 102, 3, '2024-01-02', 900.00),

(3, 103, 2, '2024-01-02', 60.00),

(4, 104, 4, '2024-01-03', 80.00),

(5, 105, 6, '2024-01-03', 90.00);

**1. Retrieve all columns from the Sales table.**

Ans select \* from Sales;

**2. Retrieve the sale\_id and sale\_date from the Sales table.**

Ans select sale\_id,sale\_date from Sales;

**3. Filter the Sales table to show only sales with a total\_price greater than $100.**

Ans select \* from Sales where total\_price>100;

**4. Retrieve the sale\_id and total\_price from the Sales table for sales made on January 3, 2024.**

Ans select sale\_id,total\_price from Sales where sale\_date='2024-01-3';

**5. Calculate the total revenue generated from all sales in the Sales table.**

Ans select sum(total\_price) as revenue from Sales;

**6. Calculate the total quantity\_sold from the Sales table.**

Ans select sum(quantity\_sold) as total\_quatity from Sales;

**7. Retrieve the sale\_id, product\_id, and total\_price from the Sales table for sales with a**

**quantity\_sold greater than 4**

Ans select \* from Sales where quantity\_sold>4;

**8. Calculate the average total\_price of sales in the Sales table.**

Ans select avg(total\_price) from Sales ;