```
CREATE SCHEMA sales;
use sales;
CREATE TABLE Sales (
 sale_id INT PRIMARY KEY,
 product_id INT,
 quantity_sold INT,
 sale_date DATE,
 total_price DECIMAL(10, 2)
);
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);
-- Retrieve all columns from the Sales table.
SELECT * FROM sales;
-- 2. Retrieve the sale_id and sale_date from the Sales table.
SELECT sale_id, sale_date FROM sales;
-- 3. Filter the Sales table to show only sales with a total_price greater than $100.
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.
SELECT sale_id, total_price FROM Sales WHERE sale_date = '2024-01-03';
-- 5. Calculate the total revenue generated from all sales
SELECT SUM(total_price) AS total_revenue FROM Sales;
-- 6. Calculate the total quantity_sold from the Sales table:
SELECT SUM(quantity_sold) AS total_quantity FROM Sales;
-- 7. Retrieve the sale_id, product_id, and total_price for sales with quantity_sold greater than 4:
SELECT sale_id, product_id, total_price FROM Sales WHERE quantity_sold > 4;
-- 8. Calculate the average total_price of sales
SELECT AVG(total_price) AS average_price FROM Sales;
```

```
CREATE SCHEMA PP;
USE PP;
CREATE TABLE Products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    category VARCHAR(50),
    unit_price DECIMAL(10, 2)
);

-- Insert sample data into Products table
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 Products table
SELECT * FROM Products;

-- 2. Retrieve the product_name and unit_price from the Products table
SELECT product_name, unit_price FROM Products;

-- 3. Filter the Products table to show only products in the 'Electronics' category
SELECT * FROM Products WHERE category = 'Electronics';

-- 4. Retrieve the product_id and product_name from the Products table for products with a
unit_price greater than $100
SELECT product_id, product_name FROM Products WHERE unit_price > 100.00;
```

-- 5. Calculate the average unit_price of products in the Products table

SELECT AVG(unit_price) AS average_price FROM Products;

-- 6. Retrieve product_name and unit_price from the Products table with the Highest Unit Price

SELECT product_name, unit_price FROM Products WHERE unit_price = (SELECT MAX(unit_price) FROM Products);

-- 7. 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;

-- 8. 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 20.00 AND 600.00;

-- 9. 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;