PRODUCTS TABLE QUERIES

1. Retrieve all columns from the Products table:

```
SELECT * FROM Products;
```

2. Retrieve the product_name and unit_price:

```
SELECT product_name, unit_price FROM Products;
```

3. Filter products in the 'Electronics' category:

```
SELECT * FROM Products
WHERE category = 'Electronics';
```

4. product_id and product_name with unit_price > $100:

```
SELECT product_id, product_name FROM Products
WHERE unit_price > 100;
```

5. Calculate average unit_price:

```
SELECT AVG(unit_price) AS average_unit_price FROM Products;
```

6. product_name and unit_price with the Highest Unit Price:

```
SELECT product_name, unit_price FROM Products
ORDER BY unit_price DESC
LIMIT 1;
```

7. Order by unit_price in descending order:

```
SELECT product_name, unit_price FROM Products
ORDER BY unit_price DESC;
```

8. Filter unit_price between $20 and $600:

```
SELECT product_name, unit_price FROM Products
WHERE unit_price BETWEEN 20 AND 600;
```

9. Order by category in ascending order:

```
SELECT product_name, category FROM Products
ORDER BY category ASC;
```

SALES TABLE QUERIES

1. Retrieve all columns from the Sales table:

```
SELECT * FROM Sales;
```

2. Retrieve sale_id and sale_date:

```
SELECT sale_id, sale_date FROM Sales;
```

3. Sales with total_price > $100:

```
SELECT * FROM Sales
WHERE total_price > 100;
```

4. sale_id and total_price on '2024-01-03':

```
SELECT sale_id, total_price FROM Sales
WHERE sale_date = '2024-01-03';
```

5. Total revenue from all sales:

```
SELECT SUM(total_price) AS total_revenue FROM Sales;
```

6. Total quantity_sold:

```
SELECT SUM(quantity_sold) AS total_quantity_sold FROM Sales;
```

7. Sales with quantity_sold > 4:

```
SELECT sale_id, product_id, total_price FROM Sales
WHERE quantity_sold > 4;
```

8. Average total_price of sales:

```
SELECT AVG(total_price) AS average_total_price FROM Sales;
```