**SQL PRACTICE**

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

Select \* from Products;

| **product\_id** | **product\_name** | **category** | **unit\_price** |
| --- | --- | --- | --- |
| 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 |

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

select product\_name, unit\_price from Products;

| **product\_name** | **unit\_price** |
| --- | --- |
| Laptop | 500.00 |
| Smartphone | 300.00 |
| Headphones | 30.00 |
| Keyboard | 20.00 |
| Mouse | 15.00 |

**3. Filter the Products table to show only products in the ‘Electronics’ category.**

Select \* from products where category = ‘Electronics’

| product\_id | product\_name | category | unit\_price |
| --- | --- | --- | --- |
| 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 |

**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;

| **product\_id** | **product\_name** |
| --- | --- |
| 101 | Laptop |
| 102 | Smartphone |

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

select avg(unit\_price) from Products;

|  |
| --- |
| avg(unit\_price) |
| 173.00000 |

**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);

| **product\_name** | **unit\_price** |
| --- | --- |
| Laptop | 500.00 |

**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;

| **product\_name** | **unit\_price** |
| --- | --- |
| Laptop | 500.00 |
| Smartphone | 300.00 |
| Headphones | 30.00 |
| Keyboard | 20.00 |
| Mouse | 15.00 |

**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 >=20 and unit\_price<=600;

| product\_name | unit\_price |
| --- | --- |
| Laptop | 500.00 |
| Smartphone | 300.00 |
| Headphones | 30.00 |
| Keyboard | 20.00 |

**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;

| product\_name | category |
| --- | --- |
| Laptop | Electronics |
| Smartphone | Electronics |
| Headphones | Electronics |
| Keyboard | Electronics |
| Mouse | Electronics |

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

select \* from Sales;

| **sale\_id** | **product\_id** | **quantity\_sold** | **sale\_date** | **total\_price** |
| --- | --- | --- | --- | --- |
| 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 |

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

select sale\_id, sale\_date from Sales;

| **sale\_id** | **sale\_date** |
| --- | --- |
| 1 | 2024-01-01 |
| 2 | 2024-01-02 |
| 3 | 2024-01-02 |
| 4 | 2024-01-03 |
| 5 | 2024-01-03 |

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

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

| **sale\_id** | **product\_id** | **quantity\_sold** | **sale\_date** | **total\_price** |
| --- | --- | --- | --- | --- |
| 1 | 101 | 5 | 2024-01-01 | 2500.00 |
| 2 | 102 | 3 | 2024-01-02 | 900.00 |

**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';

| **sale\_id** | **total\_price** |
| --- | --- |
| 4 | 80.00 |
| 5 | 90.00 |

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

select sum(total\_price) from Sales;

| **sum(total\_price)** |
| --- |
| 3630.00 |

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

select sum(quantity\_sold) from Sales;

| **sum(quantity\_sold)** |
| --- |
| 20 |

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

**quantity\_sold greater than 4.**

select sale\_id, product\_id, total\_price from Sales where quantity\_sold>4;

| **sale\_id** | **product\_id** | **total\_price** |
| --- | --- | --- |
| 1 | 101 | 2500.00 |
| 5 | 105 | 90.00 |

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

select avg(total\_price) from Sales;

| **avg(total\_price)** |
| --- |
| 726.000000 |