DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Faculties of Computer Science



Bachelors of Science in Computer Science and Information Technology (B.Sc. CSIT)

Course: Database Management Systems (CSC 265)

Class of 2027/Semester: IV

A Lab Report On:

**IMPLEMENTATION MYSQL QUERIES**

**Submitted by:**

Name: Shaurav Bhandari

Roll No: 1333

**Submitted to:**

Er Satya Ram Suwal

Faculty, Department of Computer Science

**Definition of tables and insertion of all data for First part:**

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);

-- Create 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 sample data into Sales table

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);

**Queries:**

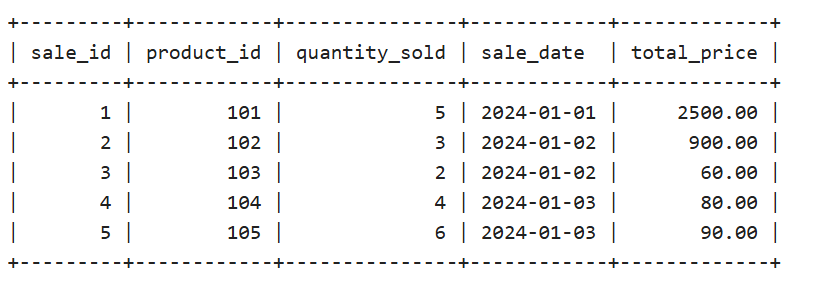
1. **Retrieve all columns from the Sales table:**

**Code:**

**SELECT \***

**FROM Sales;**

**Output:**

****

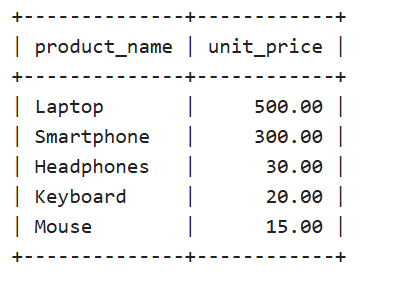
1. **Retrieve the product\_name and unit\_price from the Products table:**

**Code:**

**SELECT product\_name, unit\_price**

**FROM Products;**

**Output:**

****

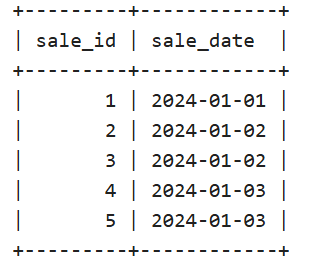
1. **Retrieve the sale\_id and sale\_date from the Sales table.**

**Code:**

**SELECT sale\_id, sale\_date**

**FROM Sales;**

**Output:**

****

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

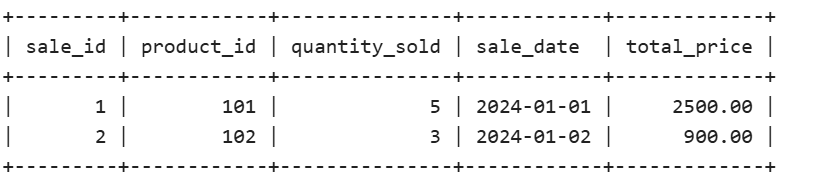
**Code:**

**SELECT \***

**FROM Sales**

**WHERE total\_price > 100;**

**Output:**

****

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

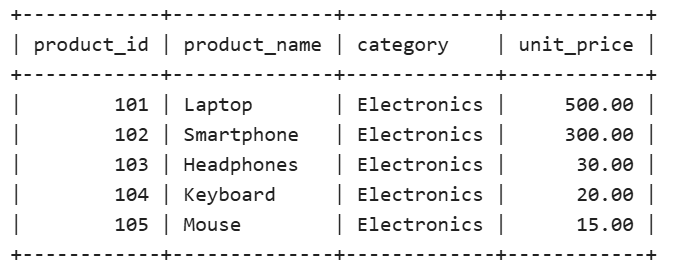
**Code:**

**SELECT \***

**FROM Products**

**WHERE category = 'Electronics';**

**Output:**

****

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

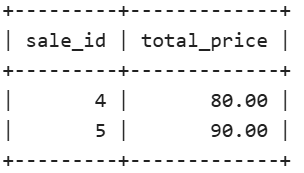
**Code:**

**SELECT sale\_id, total\_price**

**FROM Sales**

**WHERE sale\_date = '2024-01-03';**

**Output:**

****

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

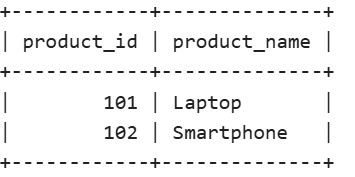
**Code:**

**SELECT product\_id, product\_name**

**FROM Products**

**WHERE unit\_price > 100;**

**Output:**

****

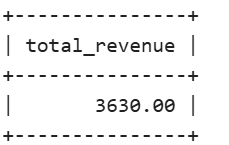
1. **Calculate the total revenue generated from all sales in the Sales table.**

**Code:**

SELECT SUM(total\_price) AS total\_revenue

FROM Sales;

**Output:**

****

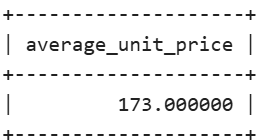
1. **Calculate the average unit\_price of products in the Products table.**

**Code:**

SELECT AVG(unit\_price) AS average\_unit\_price

FROM Products;

**Output:**

****

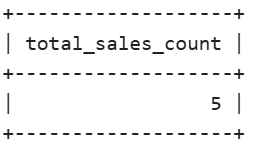
1. **Calculate the total quantity\_sold from the Sales table.**

**Code:**

SELECT COUNT(\*) AS total\_sales\_count

FROM Sales**;**

**Output:**

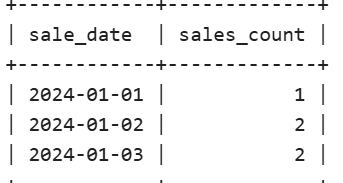
****

1. **Count Sales Per Day from the Sales table.**

**Code:**

SELECT sale\_date, COUNT(\*) AS sales\_count   
FROM Sales   
GROUP BY sale\_date   
ORDER BY sale\_date;

**Output:**

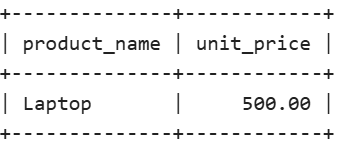
****

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

**Code:**

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

**Output:**

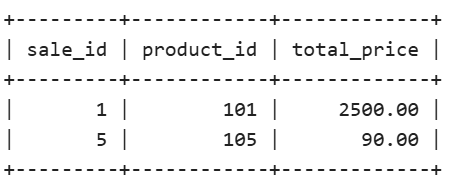
****

1. **Retrieve the sale\_id, product\_id, and total\_price from the Sales table for sales with a quantity\_sold greater than 4.**

**Code:**

SELECT sale\_id, product\_id, total\_price   
FROM Sales   
WHERE quantity\_sold > 4;

**Output:**

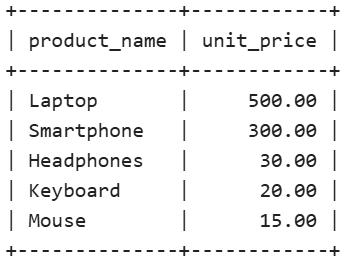
****

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

**Code:**

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

**Output:**

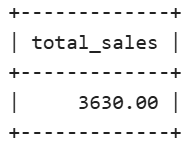
****

1. **Retrieve the total\_price of all sales, rounding the values to two decimal places.**

**Code:**

SELECT ROUND(SUM(total\_price), 2) AS total\_sales   
FROM Sales;

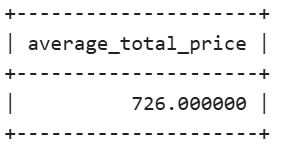
**Output:**

****

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

SELECT AVG(total\_price) AS average\_total\_price   
FROM Sales;

**Output:**

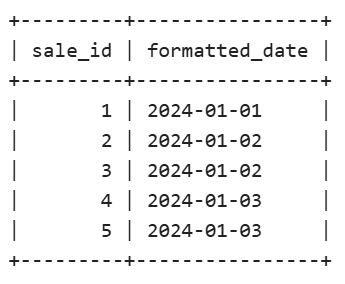
****

1. **Retrieve the sale\_id and sale\_date from the Sales table, formatting the sale\_date as 'YYYY-MM-DD'.**

**Code:**

SELECT sale\_id, DATE\_FORMAT(sale\_date, '%Y-%m-%d') AS formatted\_date   
FROM Sales;

**Output:**

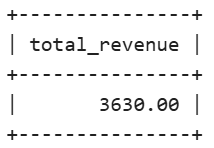
****

### **Calculate the total revenue generated from sales of products in the 'Electronics' category:**

**Code:**

SELECT SUM(Sales.total\_price) AS total\_revenue   
FROM Sales   
JOIN Products ON Sales.product\_id = Products.product\_id   
WHERE Products.category = 'Electronics';

**Output:**

****

**CONCLUSION:**

**Hence, various practices of implementation of DDL, DML, etc. were implemented using MYSQL.**