Internship Task - RDBMS and SQL Task #3

(1)create database

query=>

CREATE DATABASE Retailstores;

(2)Use Database

Query=>

use Retailstores;

(3)create table Customers

Query=>

CREATE TABLE Customers(

customer\_id INT Primary key,

first\_name VARCHAR(100) NOT NULL,

last\_name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

phone VARCHAR(20) NOT NULL,

address TEXT NOT NULL,

join\_date DATE

);

(4)create table Products

Query=>

CREATE TABLE Products(

product\_id INT Primary key,

product\_name VARCHAR(100) NOT NULL,

category VARCHAR(50),

price DECIMAL(10,2),

stock\_quantity INT);

(5)create table Orders

Query=>

CREATE TABLE Orders(

order\_id INT Primary key,

customer\_id INT,

Foreign key (customer\_id) References Customers(customer\_id),

order\_date DATE,

total\_amount DECIMAL(10,2),

order\_status VARCHAR(20)

);

(6)create table OrderDetails

Query=>

CREATE TABLE OrderDetails(

order\_detail\_id INT Primary key,

order\_id INT,

Foreign key (order\_id) References Orders(order\_id),

product\_id INT,

Foreign key (product\_id)References Products(product\_id),

quantity INT,

unit\_price DECIMAL(10,2)

);

(7)create table Payments

Query=>

CREATE TABLE Payments(

payment\_id INT Primary key,

order\_id INT,

Foreign key (order\_id)References Orders(order\_id),

payment\_date DATE,

payment\_amount DECIMAL(10,2),

payment\_method VARCHAR(20)

);

(8)Insert data in table Customers

Query=>

INSERT INTO Customers

VALUES

(1, 'seema', 'roy', 'seemaroy12@gmail.com', '8987574676', '123 Main St,indore', '2023-01-10'),

(2, 'anand', 'Sen', 'anandsen12@gmail.com', '8978490989', '456 new market,bhopal', '2022-12-15'),

(3, 'Anjali', 'Jangid', 'anjalijangid123@gmail.com', '8903454567', '789 vijay nagar,indore', '2021-08-20'),

(4, 'Sarah', 'solanki', 'sarahsolanki45@gmail.com', '7657489309', '101 college road,gwalior', '2022-05-05'),

(5, 'pankaj', 'solanki', 'pankajsolanki56@gmail.com', '9086456736', '234 main chouk,indore', '2021-11-12'),

(6, 'bhanu', 'sahu', 'bhanusahu32@gmail.com', '7849098756', '567 main bypass,rewa', '2023-03-01'),

(7, 'siya', 'Taylor', 'siyataylor@gmail.com', '9098678949', '890 bus stand,harda', '2022-07-18'),

(8, 'Matthew', 'roy', 'matthewroy@gmail.com', '9023456776', '123 station rd, dewas', '2021-10-25'),

(9, 'Sophia', 'roy', 'sophiaroy@gmail.com', '9098678909', '456 mahakal lok, ujjain', '2022-01-30'),

(10, 'David', 'barnar', 'davidbarnar@gmail.com', '9867556786', '789 main road,bhopal', '2023-06-05');

(9)Insert data in table Products

Query=>

INSERT INTO Products

VALUES

(1, 'Laptop', 'Electronics', 8990.99, 50),

(2, 'Smartphone', 'Electronics', 6990.49, 120),

(3, 'Headphones', 'Accessories', 99.99, 200),

(4, 'Bluetooth Speaker', 'Accessories', 490.99, 150),

(5, 'Coffee Maker', 'Appliances', 79.99, 80),

(6, 'Microwave Oven', 'Appliances', 7049.99, 60),

(7, 'LED TV', 'Electronics', 8499.99, 40),

(8, 'Gaming Console', 'Electronics', 299.99, 30),

(9, 'Washing Machine', 'Appliances', 6399.99, 20),

(10, 'Refrigerator', 'Appliances', 9799.99, 25);

(10)Insert data in table Orders

Query=>

INSERT INTO Orders

VALUES

(101, 1, '2023-01-12', 150.75, 'Shipped'),

(102, 2, '2023-02-05', 245.50, 'Delivered'),

(103, 3, '2023-03-20', 89.99, 'Pending'),

(104, 4, '2023-04-15', 320.00, 'Shipped'),

(105, 5, '2023-05-22', 499.99, 'Delivered'),

(106, 6, '2023-06-10', 119.50, 'Cancelled'),

(107, 7, '2023-07-01', 245.00, 'Shipped'),

(108, 8, '2023-08-15', 149.99, 'Pending'),

(109, 9, '2023-09-03', 300.00, 'Delivered'),

(110, 10, '2023-10-05', 59.95, 'Shipped');

(11)Insert data in table OrderDetails

Query=>

INSERT INTO OrderDetails

VALUES

(201, 101, 1, 1, 899.99),

(202, 102, 3, 2, 99.99),

(203, 103, 2, 1, 699.49),

(204, 104, 4, 1, 49.99),

(205, 105, 5, 1, 79.99),

(206, 106, 7, 1, 499.99),

(207, 107, 6, 2, 149.99),

(208, 108, 8, 1, 299.99),

(209, 109, 9, 1, 399.99),

(210, 110, 10, 1, 799.99);

(12)Insert data in table Payments

Query=>

INSERT INTO Payments

VALUES

(301, 101, '2023-01-13', 150.75, 'Credit Card'),

(302, 102, '2023-02-06', 245.50, 'PayPal'),

(303, 103, '2023-03-21', 89.99, 'Debit Card'),

(304, 104, '2023-04-16', 320.00, 'Credit Card'),

(305, 105, '2023-05-23', 499.99, 'Cash'),

(306, 106, '2023-06-11', 119.50, 'Credit Card'),

(307, 107, '2023-07-02', 245.00, 'PayPal'),

(308, 108, '2023-08-16', 149.99, 'Debit Card'),

(309, 109, '2023-09-04', 300.00, 'Credit Card'),

(310, 110, '2023-10-06', 59.95, 'Cash');

===========================================================================================================================================================

SQL Queries for the Case Study

1. Find the Total Number of Orders for Each Customer

Query=>

SELECT customer\_id, COUNT(\*) AS total\_orders

FROM Orders

GROUP BY customer\_id;

2. Find the Total Sales Amount for Each Product (Revenue per Product)

Query=>

SELECT products.product\_id, products.product\_name, SUM(orderdetails.quantity \* orderdetails.unit\_price) AS total\_sales

FROM OrderDetails

JOIN Products ON orderdetails.product\_id = products.product\_id

GROUP BY products.product\_id;

3. Find the Most Expensive Product Sold

Query=>

SELECT products.product\_name, MAX(orderdetails.unit\_price) AS most\_expensive\_price

FROM OrderDetails

JOIN Products ON orderdetails.product\_id = products.product\_id

GROUP BY products.product\_name

ORDER BY most\_expensive\_price DESC

LIMIT 1;

4. Get the List of Customers Who Have Placed Orders in the Last 30 Days

Query=>

SELECT customers.customer\_id, customers.first\_name, customers.last\_name, customers.email

FROM Customers

JOIN Orders ON customers.customer\_id = orders.customer\_id

WHERE orders.order\_date >= CURDATE() - INTERVAL 30 DAY;

5. Calculate the Total Amount Paid by Each Customer

Query=>

SELECT c.customer\_id, c.first\_name, c.last\_name, SUM(p.payment\_amount) AS total\_paid

FROM Customers c

JOIN Orders o ON c.customer\_id = o.customer\_id

JOIN Payments p ON o.order\_id = p.order\_id

GROUP BY c.customer\_id;

6. Get the Number of Products Sold by Category

Query=>

SELECT p.category, SUM(od.quantity) AS total\_products\_sold

FROM OrderDetails od

JOIN Products p ON od.product\_id = p.product\_id

GROUP BY p.category;

7. List All Orders That Are Pending (i.e., Orders that haven't been shipped yet)

Query=>

SELECT o.order\_id, o.customer\_id, o.order\_date, o.total\_amount, o.order\_status

FROM Orders o

WHERE o.order\_status = 'Pending';

8. Find the Average Order Value (Total Order Amount / Number of Orders)

Query=>

SELECT AVG(total\_amount) AS average\_order\_value

FROM Orders;

9. List the Top 5 Customers Who Have Spent the Most Money

Query=>

SELECT c.customer\_id, c.first\_name, c.last\_name, SUM(p.payment\_amount) AS total\_spent

FROM Customers c

JOIN Orders o ON c.customer\_id = o.customer\_id

JOIN Payments p ON o.order\_id = p.order\_id

GROUP BY c.customer\_id

ORDER BY total\_spent DESC

LIMIT 5;

10. Find the Products That Have Never Been Sold

Query=>

SELECT p.product\_name

FROM Products p

LEFT JOIN OrderDetails od ON p.product\_id = od.product\_id

WHERE od.product\_id IS NULL;

=============================================================================================================================================