SQL PROJECT

SALES MANAGEMENT SYSYTEM

SATYAM NIVRUTTI GAWALI

CONTENT

ABOUT

The Sales Management System will help a company track its sales performance, manage orders, monitor inventory, and generate sales reports

FEATURES

Product Management

Order Management

-

Customer Management

Inventory Management

-

Salesperson Management

Sales Reports

-

CREATE DATABASE

```
1 • CREATE DATABASE SALES ;
```

CREATING TABLES & INSERTING VALUES

TABLE 1 : Products

```sql
CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    product_name VARCHAR(255) NOT NULL,
    category VARCHAR(100),
    price DECIMAL(10 , 2 ) NOT NULL,
    stock_quantity INT NOT NULL
);
```

```sql
INSERT INTO Products (product_name, category, price, stock_quantity) VALUES
('Laptop', 'Electronics', 800.00, 50),
('Smartphone', 'Electronics', 500.00, 100),
('Headphones', 'Accessories', 100.00, 150),
('Desk Chair', 'Furniture', 150.00, 30),
('Mouse', 'Accessories', 25.00, 200);
```

TABLE 2 : Customers

```sql
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    phone_number VARCHAR(20),
    address VARCHAR(255)
);
```

```sql
INSERT INTO Customers (first_name, last_name, email, phone_number, address) VALUES
('Alice', 'Johnson', 'alice.johnson@email.com', '555-1234', '123 Maple St'),
('Bob', 'Smith', 'bob.smith@email.com', '555-5678', '456 Oak St'),
('Charlie', 'Brown', 'charlie.brown@email.com', '555-9876', '789 Pine St');
```

TABLE 3 : Salespersons

```sql
CREATE TABLE Salespersons (
    salesperson_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    commission_rate DECIMAL(5, 2) NOT NULL -- Commission rate as a percentage
);
```

```sql
INSERT INTO Salespersons (first_name, last_name, commission_rate) VALUES
('John', 'Doe', 5.00),
('Jane', 'Smith', 6.00),
('Bob', 'Brown', 4.00);
```

TABLE 4 : Orders

```sql
CREATE TABLE Orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    salesperson_id INT,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id),
    FOREIGN KEY (salesperson_id) REFERENCES Salespersons(salesperson_id)
);
```

```sql
INSERT INTO Orders (customer_id, salesperson_id, total_amount) VALUES
(1, 2, 1300.00),
(2, 1, 600.00),
(3, 3, 275.00);
```

TABLE 5 : Order_Details

```sql
CREATE TABLE Order_Details (
    order_detail_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    product_id INT,
    quantity INT NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

```sql
INSERT INTO Order_Details (order_id, product_id, quantity, price) VALUES
(1, 1, 1, 800.00),
(1, 3, 2, 100.00),
(2, 2, 1, 500.00),
(3, 4, 1, 150.00),
(3, 5, 5, 25.00);
```

TABLE 5 : Inventory_Transactions

```sql
CREATE TABLE Inventory_Transactions (
    transaction_id INT PRIMARY KEY AUTO_INCREMENT,
    product_id INT,
    transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    transaction_type VARCHAR(50), -- 'Sale', 'Restock', 'Return', etc.
    quantity INT NOT NULL,
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

```sql
INSERT INTO Inventory_Transactions (product_id, transaction_type, quantity) VALUES
(1, 'Sale', 1),
(3, 'Sale', 2),
(2, 'Sale', 1),
(4, 'Sale', 1),
(5, 'Sale', 5);
```

SQL QUERIES

01 : View total sales by salesperson

```sql
SELECT
    S.first_name,
    S.last_name,
    SUM(O.total_amount) AS total_sales
FROM
    Orders O
        JOIN
    Salespersons S ON O.salesperson_id = S.salesperson_id
GROUP BY S.salesperson_id;
```

Result Grid | Filter Rows:

first_name	last_name	total_sales
Jane	Smith	1300.00
John	Doe	600.00
Bob	Brown	275.00

02 : View product sales performance

```sql
SELECT P.product_name, SUM(OD.quantity) AS total_sold, SUM(OD.quantity * OD.price) AS total_sales
FROM Order_Details OD
JOIN Products P ON OD.product_id = P.product_id
GROUP BY P.product_id;
```

Result Grid | Filter Rows:

product_name	total_sold	total_sales
Laptop	1	800.00
Headphones	2	200.00
Smartphone	1	500.00
Desk Chair	1	150.00
Mouse	5	125.00

03 : View current inventory levels

```
SELECT P.product_name, P.stock_quantity
FROM Products P;
```

Result Grid | Filter Rows:

product_name	stock_quantity
Laptop	50
Smartphone	100
Headphones	150
Desk Chair	30
Mouse	200

04 : View customer purchase history

```sql
SELECT C.first_name, C.last_name, O.order_date, O.total_amount
FROM Orders O
JOIN Customers C ON O.customer_id = C.customer_id
WHERE C.customer_id = 1;
```

Result Grid | Filter Rows: | Export:

first_name	last_name	order_date	total_amount
Alice	Johnson	2024-11-24 13:15:37	1300.00

05 : Calculate salesperson commission

```sql
SELECT S.first_name, S.last_name, SUM(O.total_amount) * (S.commission_rate / 100) AS commission
FROM Orders O
JOIN Salespersons S ON O.salesperson_id = S.salesperson_id
GROUP BY S.salesperson_id;
```

Result Grid | Filter Rows:

first_name	last_name	commission
Jane	Smith	78.00000000
John	Doe	30.00000000
Bob	Brown	11.00000000

06 : Check for low stock products (less than 40)

```sql
SELECT P.product_name, P.stock_quantity
FROM Products P
WHERE P.stock_quantity < 40;
```

Result Grid | Filter Rows:

	product_name	stock_quantity
▶	Desk Chair	30

07 : View total sales per product category

```sql
SELECT P.category, SUM(OD.quantity * OD.price) AS total_sales
FROM Order_Details OD
JOIN Products P ON OD.product_id = P.product_id
GROUP BY P.category;
```

Result Grid | Filter Rows:

category	total_sales
Electronics	1300.00
Accessories	325.00
Furniture	150.00

08 : View inventory transactions (Sales, Restocks, Returns)

```sql
SELECT P.product_name, IT.transaction_type, IT.quantity, IT.transaction_date
FROM Inventory_Transactions IT
JOIN Products P ON IT.product_id = P.product_id;
```

Result Grid | Filter Rows: | Export: | Wrap

product_name	transaction_type	quantity	transaction_date
Laptop	Sale	1	2024-11-24 13:16:19
Headphones	Sale	2	2024-11-24 13:16:19
Smartphone	Sale	1	2024-11-24 13:16:19
Desk Chair	Sale	1	2024-11-24 13:16:19
Mouse	Sale	5	2024-11-24 13:16:19

CONCLUSION

The Sales Management System (SMS) is a comprehensive **SQL** project that covers key aspects of a sales operation, including product management, customer orders, inventory tracking, and sales performance analysis. It generates valuable insights, such as identifying the most popular products, calculating salesperson commissions, and tracking stock levels. This project is practical and scalable.

THANK'S FOR WATCHING