Data Digger – SQL Project

A complete SQL practice project for building and managing an E-Commerce Store Database.

This project covers:

- Database creation
- CRUD operations
- Keys & constraints
- Aggregate functions
- Filtering, Sorting, Group By
- Safe Update mode

Database Setup

```
CREATE DATABASE data_digger;

USE data_digger;
```

Customers Table

```
CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

Name VARCHAR(50),

Email VARCHAR(100),

Address VARCHAR(100)

);

INSERT INTO Customers (CustomerID, Name, Email, Address) VALUES

(1, 'Alice', 'alice@example.com', 'manpur'),
(2, 'Bob', 'bob@example.com', 'methan'),

(3, 'Charlie', 'charlie@example.com', 'gandhinagr'),

(4, 'David', 'david@example.com', 'kuntalpur'),

(5, 'Emma', 'emma@example.com', 'khanpur');
```

Important Queries:

```
SELECT * FROM Customers;

UPDATE Customers SET Address = 'pipali' WHERE Customerid = 5;

DELETE FROM Customers WHERE Customerid = 5;

SELECT * FROM Customers WHERE Name = 'Alice';

DROP TABLE Customers;
```

Orders Table

```
CREATE TABLE Orders (

Orderid INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

TotalAmount DECIMAL(10,2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES

(101, 1, '2025-10-20', 45.99),

(102, 2, '2025-10-25', 120.50),

(103, 3, '2025-11-05', 88.00),

(104, 4, '2025-11-10', 25.75),

(105, 5, '2025-11-12', 300.00);
```

Queries:

```
SELECT * FROM Orders WHERE CustomerID = 2;

UPDATE Orders SET TotalAmount = 3000 WHERE CustomerID = 2;

DELETE FROM Orders WHERE OrderID = 5;

SELECT * FROM Orders WHERE OrderDate >= (CURDATE() - INTERVAL 30 DAY);

SELECT MAX(TotalAmount), MIN(TotalAmount), AVG(TotalAmount) FROM Orders;
```

Products Table

```
CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Price DECIMAL(10, 2),

Stock INT

);

INSERT INTO Products VALUES

(1, 'Laptop', 45000.00, 15),

(2, 'Mouse', 750.00, 150),

(3, 'Keyboard', 1250.50, 80),

(4, 'Monitor', 15000.00, 25),

(5, 'Webcam', 450.00, 0);
```

Queries:

```
SELECT * FROM Products ORDER BY Price DESC;

UPDATE Products SET Price = 60000 WHERE ProductID = 2;

DELETE FROM Products WHERE Stock = 0;
```

```sql
SELECT * FROM Products WHERE Price BETWEEN 500 AND 2000;

SELECT MAX(Price), MIN(Price) FROM Products;
```

OrderDetails Table

```sql
CREATE TABLE OrderDetails (
OrderDetailID INT PRIMARY KEY,
OrderID INT,
ProductID INT,
Quantity INT,
SubTotal DECIMAL(10, 2),
FOREIGN KEY (ProductID) REFERENCES Products(ProductID),
FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
);

INSERT INTO OrderDetails VALUES
(1, 101, 1, 2, 1500.00),
(2, 102, 2, 1, 1250.50),
(3, 103, 3, 1, 45000.00),
(4, 104, 4, 5, 3750.00),
(5, 105, 5, 2, 5000.00);
```

Queries:

```sql
SELECT * FROM OrderDetails WHERE OrderID = 1;

SELECT ProductID, SUM(Quantity) AS TotalSold FROM OrderDetails GROUP BY ProductID ORDER BY TotalSold DESC LIMIT 3;

SELECT ProductID, COUNT(*) AS TimesSold FROM OrderDetails WHERE ProductID = 2 GROUP BY ProductID;
```

Project Completed – Full SQL functionality demonstration.