Advanced SQL Project: Data Analysis Using SQL Features

1. Dataset Selection

Dataset Overview

The dataset contains sales transaction details, including order numbers, product details, customer information, and sales revenue.

1.1 Download the Database

- The dataset contains order and customer details, including sales, quantity, order status, and customer information.

1.2 Import the Dataset into SQL

Create a new database in MySQL, PostgreSQL, or SQL Server.

- Created a database:

#Query

```
create database sales_data;
use sales_data;
```

- Created a table based on dataset columns:

#Query

```
CREATE TABLE orders (
   ORDERNUMBER INT,
   QUANTITYORDERED INT,
   PRICEEACH DECIMAL(10,2),
   ORDERLINENUMBER INT,
   SALES DECIMAL(10,2),
   ORDERDATE DATE,
   STATUS VARCHAR(50),
   QTR_ID INT,
   MONTH_ID INT,
   YEAR_ID INT,
   PRODUCTLINE VARCHAR(100),
   MSRP DECIMAL(10,2),
   PRODUCTCODE VARCHAR(50),
   CUSTOMERNAME VARCHAR(100),
   PHONE VARCHAR(50),
   ADDRESSLINE1 VARCHAR(255),
   ADDRESSLINE2 VARCHAR(255),
   CITY VARCHAR(100),
   STATE VARCHAR(100),
   POSTALCODE VARCHAR(20),
   COUNTRY VARCHAR(50),
   TERRITORY VARCHAR(50),
   CONTACTLASTNAME VARCHAR(50),
   CONTACTFIRSTNAME VARCHAR(50),
   DEALSIZE VARCHAR(20)
);
```

- Inserted the dataset using:

#Query

```
 INSERT INTO orders VALUES (10107,30,95.7,2,2871,'2003-02-24
00:00:00','Shipped',1,2,2003,'Motorcycles',95,'S10_1678','Land of Toys Inc.','2125557818','897
Long Airport Avenue',NULL,'NYC','NY','10022','USA','NA','Yu','Kwai','Small'),
        (10121,34,81.35,5,2765.9,'2003-05-07
00:00:00','Shipped',2,5,2003,'Motorcycles',95,'S10_1678','Reims Collectables','26.47.1555','59
rue de l''Abbaye',NULL,'Reims',NULL,'51100','France','EMEA','Henriot','Paul','Small'),
        (10134,41,94.74,2,3884.34,'2003-07-01
00:00:00','Shipped',3,7,2003,'Motorcycles',95,'S10_1678','Lyon Souveniers','+33 1 46 62
7555','27 rue du Colonel Pierre Avia',NULL,'Paris',NULL,'75508','France','EMEA','Da
Cunha','Daniel','Medium'),
        (10145,45,83.26,6,3746.7,'2003-08-25
00:00:00','Shipped',3,8,2003,'Motorcycles',95,'S10_1678','Toys4GrownUps.com','6265557265','
78934 Hillside Dr.',NULL,'Pasadena','CA','90003','USA','NA','Young','Julie','Medium'),
        (10159,49,100,14,5205.27,'2003-10-10
00:00:00','Shipped',4,10,2003,'Motorcycles',95,'S10_1678','Corporate Gift Ideas
Co.','6505551386','7734 Strong St.',NULL,'San
Francisco','CA',NULL,'USA','NA','Brown','Julie','Medium'),
        (10168,36,96.66,1,3479.76,'2003-10-28
00:00:00','Shipped',4,10,2003,'Motorcycles',95,'S10_1678','Technics Stores
Inc.','6505556809','9408 Furth
Circle',NULL,'Burlingame','CA','94217','USA','NA','Hirano','Juri','Medium'),
        (10180,29,86.13,9,2497.77,'2003-11-11
00:00:00','Shipped',4,11,2003,'Motorcycles',95,'S10_1678','Daedalus Designs
Imports','20.16.1555','184, chausse de
Tournai',NULL,'Lille',NULL,'59000','France','EMEA','Rance','Martine','Small'),
```

2. Data Cleaning & Preparation

2.1 Identify and Remove Duplicates

- Check for duplicate records:

# Query

```
SELECT ORDERNUMBER, COUNT(*)
FROM orders
GROUP BY ORDERNUMBER
HAVING COUNT(*) > 1;
```

- Remove duplicates:

Remove duplicates using SELECT DISTINCT and DELETE with ROW_NUMBER().

- Standardize text fields using UPPER(), LOWER(), and TRIM().

# Query

```
DELETE FROM orders
WHERE ORDERNUMBER IN (
  SELECT ORDERNUMBER FROM (
    SELECT ORDERNUMBER, ROW_NUMBER() OVER (PARTITION BY ORDERNUMBER ORDER BY
ORDERNUMBER) AS row_num
    FROM orders
  ) t WHERE row_num > 1
);
```

2.2 Standardize Text Fields

- Convert product and customer names to uppercase:

# Query

```
UPDATE orders
SET CUSTOMERNAME = UPPER(CUSTOMERNAME), PRODUCTLINE = UPPER(PRODUCTLINE);
```

- Trim spaces from addresses

# Query

```
UPDATE orders
SET ADDRESSLINE1 = TRIM(ADDRESSLINE1), ADDRESSLINE2 = TRIM(ADDRESSLINE2);
```

3. Data Exploration & Aggregations

3.1 Top 5 Highest-Selling Products

# Query

```
SELECT PRODUCTLINE, SUM(SALES) AS TotalSales
FROM orders
GROUP BY PRODUCTLINE
ORDER BY TotalSales DESC
LIMIT 5;
```

3.2 Top 3 Customers Based on Spending

# Query

```
SELECT CUSTOMERNAME, SUM(SALES) AS TotalSpent
FROM orders
GROUP BY CUSTOMERNAME
ORDER BY TotalSpent DESC
LIMIT 3;
```

4. Joins & Relationships

4.1 INNER JOIN: Customer & Order Data

# Query

```
SELECT s.CUSTOMERNAME, s.ORDERDATE, s.SALES
FROM orders s
INNER JOIN orders c ON s.CUSTOMERNAME = c.CUSTOMERNAME;
```

4.2 LEFT JOIN: Customers with or without Orders

# Query

```
SELECT c.CUSTOMERNAME, s.ORDERNUMBER, s.SALES
FROM orders c
LEFT JOIN orders s ON c.CUSTOMERNAME = s.CUSTOMERNAME;
```

5. Subqueries & CTEs

5.1 Find the Customer with the Highest Purchase

# Query

```
SELECT CUSTOMERNAME, SALES
FROM orders
WHERE SALES = (SELECT MAX(SALES) FROM orders);
```

5.2 CTE for High-Value Orders

# Query

```
WITH HighValueOrders AS (
   SELECT * FROM orders WHERE SALES > 500
)
SELECT * FROM HighValueOrders;
```

6. Performance Optimization

6.1 Indexing

# Query

```
CREATE INDEX idx_customer ON orders(CUSTOMERNAME);
```

6.2 Query Execution Plan Analysis

# Query

```
EXPLAIN ANALYZE
SELECT CUSTOMERNAME, SUM(SALES)
FROM orders
GROUP BY CUSTOMERNAME;
```

This project demonstrates SQL expertise through advanced queries, performance tuning, and data insights.