Sales & Customer Insights Analysis Using SQL

To analyze sales performance, customer behavior, product profitability, and employee productivity across offices, using SQL queries on the company's transactional database.

Tools Used: MySQL Workbench

Sneha Muley

GITHUB

Objective

- The project focuses on analyzing customer and product data using SQL.
- Perform data analysis using SQL queries.
- Understand relationships between Customers, Orders, and Products.
- Aims to derive meaningful business insights from various relational tables.
- Helps understand customer behavior, sales trends, and product performance.

Schema

customers → orders → orderdetails → products → productlines
customers → payments
employees → customers
employees → offices

Sales Performance Analysis

1.Total sales

```sql
#1 Total sales
select round(sum(quantityOrdered * priceEach),2) as total_sales
from orderdetails;
```

Result

total_sales
9604190.61

2. Monthly sales trends

```sql
#2 Monthly sales trends
select date_format(orderDate,'%Y-%m') as month,
round(sum(quantityOrdered * priceEach),2) as total_sales
from orders o
join orderdetails od on o.orderNumber=od.orderNumber
group by month
order by month;
```

Result

month	total_sales
2003-01	116692.77
2003-02	128403.64
2003-03	160517.14
2003-04	185848.59
2003-05	179435.55
2003-06	150470.77
2003-07	201940.36
2003-08	178257.11
2003-09	236697.85
2003-10	514336.21
2003-11	988025.15
2003-12	276723.25
2004-01	292385.21
2004-02	289502.84
2004-03	217691.26

Result 1

3 .Top 5 products by sales

```sql
#3 Top 5 products by sales
select p.productName,
round(SUM(od.quantityOrdered * od.priceEach),2) as total_sales
from orderdetails od
join products p on od.productCode=p.productCode
group by p.productName
order by total_sales desc
limit 5;
```

Result

productName	total_sales
1992 Ferrari 360 Spider red	276839.98
2001 Ferrari Enzo	190755.86
1952 Alpine Renault 1300	190017.96
2003 Harley-Davidson Eagle Drag Bike	170686.00
1968 Ford Mustang	161531.48

4.Revenue by productline

```sql
# Revenue by productline
SELECT pl.productLine,
       ROUND(SUM(od.quantityOrdered * od.priceEach), 2) AS total_revenue
FROM orderdetails od
JOIN products p ON od.productCode = p.productCode
JOIN productlines pl ON p.productLine = pl.productLine
GROUP BY pl.productLine
ORDER BY total_revenue DESC;
```

Result

productLine	total_revenue
Classic Cars	3853922.49
Vintage Cars	1797559.63
Motorcycles	1121426.12
Trucks and Buses	1024113.57
Planes	954637.54
Ships	663998.34
Trains	188532.92

Customer Analysis

1.Top 10 Customers by Total Spend

```sql
# Top 10 Customers by Total Spend
SELECT c.customerName,
       ROUND(SUM(od.quantityOrdered * od.priceEach), 2) AS total_spent
FROM customers c
JOIN orders o ON c.customerNumber = o.customerNumber
JOIN orderdetails od ON o.orderNumber = od.orderNumber
GROUP BY c.customerName
ORDER BY total_spent DESC
LIMIT 10;
```

Result

customerName	total_spent
Euro+ Shopping Channel	820689.54
Mini Gifts Distributors Ltd.	591827.34
Australian Collectors, Co.	180585.07
Muscle Machine Inc	177913.95
La Rochelle Gifts	158573.12
Dragon Souveniers, Ltd.	156251.03
Down Under Souveniers, Inc	154622.08
Land of Toys Inc.	149085.15
AV Stores, Co.	148410.09
The Sharp Gifts Warehouse	143536.27

2.Average Order Value (AOV) per Customer

```sql
#Average Order Value (AOV) per Customer
SELECT c.customerName,
       ROUND(AVG(o2.total_amount), 2) AS avg_order_value
FROM customers c
JOIN (
GROUP BY c.customerName;
```

3.Customers Without Payments (Potential Credit Issues)

```sql
SELECT c.customerName
FROM customers c
LEFT JOIN payments p ON c.customerNumber = p.customerNumber
WHERE p.customerNumber IS NULL;
```

Employee & Office Performance

1.Top Performing Sales Representatives

```sql
# Top Performing Sales Representative
SELECT e.firstName, e.lastName,
       ROUND(SUM(od.quantityOrdered * od.priceEach), 2) AS total_sale
FROM employees e
JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN orders o ON c.customerNumber = o.customerNumber
JOIN orderdetails od ON o.orderNumber = od.orderNumber
GROUP BY e.employeeNumber
ORDER BY total_sales DESC;
```

Result

	firstName	lastName	total_sales
▶	Gerard	Hernandez	1258577.81
	Leslie	Jennings	1081530.54
	Pamela	Castillo	868220.55
	Larry	Bott	732096.79
	Barry	Jones	704853.91
	George	Vanauf	669377.05
	Peter	Marsh	584593.76
	Loui	Bondur	569485.75
	Andy	Fixter	562582.59
	Steve	Patterson	505875.42
	Foon Yue	Tseng	488212.67
	Mami	Nishi	457110.07
	Martin	Gerard	387477.47
	Julie	Firrelli	386663.20
	Leslie	Thompson	347533.03

2.Sales by Office Location

```sql
SELECT o.city, o.country,
       ROUND(SUM(od.quantityOrdered * od.priceEach), 2) AS total_sales
FROM offices o
JOIN employees e ON o.officeCode = e.officeCode
JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN orders ord ON c.customerNumber = ord.customerNumber
JOIN orderdetails od ON ord.orderNumber = od.orderNumber
GROUP BY o.city, o.country
ORDER BY total_sales DESC;
```

Result

city	country	total_sales
Paris	France	3083761.58
London	UK	1436950.70
San Francisco	USA	1429063.57
NYC	USA	1157589.72
Sydney	Australia	1147176.35
Boston	USA	892538.62
Tokyo	Japan	457110.07

Product Profitability Analysis

1.Profit Margin by Product

```sql
# Profit Margin by Product
SELECT productName,
       ROUND(SUM(quantityOrdered * (priceEach - buyPrice)), 2) AS total_profit
FROM orderdetails od
JOIN products p ON od.productCode = p.productCode
GROUP BY productName
ORDER BY total_profit DESC;
```

Result

productName	total_profit
1992 Ferrari 360 Spider red	135996.78
1952 Alpine Renault 1300	95282.58
2001 Ferrari Enzo	93349.65
2003 Harley-Davidson Eagle Drag Bike	81031.30
1968 Ford Mustang	72579.26
1969 Ford Falcon	72399.77
1928 Mercedes-Benz SSK	68423.18
2002 Suzuki XREO	67641.47
1980s Black Hawk Helicopter	64599.11
1948 Porsche Type 356 Roadster	62725.78
1917 Grand Touring Sedan	60945.00
1957 Corvette Convertible	59910.22
1962 Volkswagen Microbus	55655.47
1976 Ford Gran Torino	54647.25
1932 Model A Ford J-Coupe	54026.65

Result 1

2.Most Ordered Products

```sql
#Most Ordered product
SELECT p.productName, SUM(od.quantityOrdered) AS total_quantity
FROM orderdetails od
JOIN products p ON od.productCode = p.productCode
GROUP BY p.productName
ORDER BY total_quantity DESC
LIMIT 5;
```

Result

productName	total_quantity
1992 Ferrari 360 Spider red	1808
1937 Lincoln Berline	1111
American Airlines: MD-11S	1085
1941 Chevrolet Special Deluxe Cabriolet	1076
1930 Buick Marquette Phaeton	1074

Insights & Observations

- Certain products contribute significantly to total revenue.
- A few customers account for a large portion of sales .
- Seasonal or monthly trends can be identified in order data.
- Helps in identifying high-value customers and optimizing inventory.

Conclusion

- SQL proved to be a powerful tool for structured data analysis.
- Helped uncover key patterns in customer and sales data.
- The project enhanced my understanding of joins, subqueries, and data relationships.