E-Commerce Sales Data Analysis

SOLUTIONS

Client: SwiftMarket

Sales Period: 2023

Solutions

1. Provide a visual representation of our annual sales performance over time?

```
SELECT
    MONTHNAME(SaleDate) AS Month,
    SUM(TotalPrice) AS DailySales
FROM Salesdata
GROUP BY MONTH(SaleDate),MONTHNAME(SaleDate)
ORDER BY MONTH(SaleDate),MONTHNAME(SaleDate);
```

Alteranatively

```
SELECT
    YEAR(SaleDate) AS Year,
    MONTH(SaleDate) AS Month,
    DAY(SaleDate) AS Day,
    SUM(TotalPrice) AS DailySales
FROM Sales
GROUP BY YEAR(SaleDate), MONTH(SaleDate), DAY(SaleDate)
ORDER BY YEAR(SaleDate), MONTH(SaleDate), DAY(SaleDate);
```

2. The client is interested in understanding how the sales have been fluctuating over time to identify any underlying patterns or anomalies Analyze the trend in our monthly sales using a moving average technique

```
select
months,
AVG(monthlysales) OVER (ORDER BY monthnum ROWS BETWEEN 3
PRECEDING AND CURRENT ROW) AS MovingAverage
from
(SELECT
     month(saledate) monthnum,
    monthname(SaleDate) as months,
    sum(totalprice) as monthlysales
FROM Salesdata
Group by month(saledate),monthname(saledate)) t;
```

3. Explore the month-over-month growth rate in sales.

```
SELECT
    MONTH(SaleDate) AS Monthnum,
    monthname(saledate) as months,
    100*(SUM(TotalPrice) - LAG(SUM(TotalPrice)) OVER (ORDER BY
MONTH(SaleDate)))/
    LAG(SUM(TotalPrice)) OVER (ORDER BY MONTH(SaleDate))AS
'growthrate in %'
FROM Salesdata
GROUP BY YEAR(SaleDate), MONTH(SaleDate),monthname(saledate)
ORDER BY YEAR(SaleDate), MONTH(SaleDate),monthname(saledate);
```

Alternatively

```
select
months,
100*(monthlysales - lag(monthlysales) over(order by monthnum))/
     lag(monthlysales) over(order by monthnum) as 'growthrate
in %'
from
(SELECT
     month(saledate) monthnum,
    monthname(SaleDate) as months,
    sum(totalprice) as monthlysales
FROM Salesdata
Group by month(saledate),monthname(saledate)) t
order by monthnum,months;
```

4. identify the top-performing sales representatives based on their total sales volume

```
select t.employeeid,e.firstname,t.totalsales
from
(select employeeid,sum(totalprice) as totalsales
from salesdata
group by EmployeeID) t
join employees e
on t.employeeid=e.EmployeeID
order by t.totalsales desc;
```

5. What are the total sales made by each employee?

This query is same as question4 but without order by
these questions can also be asked for monthly top performers.
Sometimes, different questions might result in same queries.

```
SELECT
    EmployeeID,
    SUM(TotalPrice) AS TotalSales
FROM Salesdata
GROUP BY EmployeeID;
```

6. Which category has the highest total sales?

```
SELECT
    c.CategoryName,
    SUM(s.TotalPrice) AS TotalSales
FROM Salesdata s
JOIN Subcategories sc ON s.SubcatID = sc.SubcatID
JOIN Categories c ON sc.CategoryID = c.CategoryID
GROUP BY c.CategoryName
ORDER BY TotalSales DESC
LIMIT 1;
```

The following is the Python code to represent the query as a text chart. Try it out!

```
query = """SELECT
    c.CategoryName,
    SUM(s.TotalPrice) AS TotalSales
FROM Salesdata s
JOIN Subcategories sc ON s.SubcatID = sc.SubcatID
JOIN Categories c ON sc.CategoryID = c.CategoryID
GROUP BY c.CategoryName
ORDER BY TotalSales DESC
LIMIT 1;"""
```

```
df = read_query(query)

categoryname = df['CategoryName'].values[0]
value = df['TotalSales'].astype(float).values[0]
value_fmt = f"{value:,}"

# Setting the plot size and background
plt.figure(figsize=(8, 3), facecolor='#00008B')

# # SETTING THE GRAPH BACKGROUND COLOR: BY 2 METHODS
# Method 1:
# setting using gca() method

# plt.gca().set_facecolor('#F0FFFF')


# Method 2
# Setting the background color of the plot using set_facecolor()
method
ax = plt.axes()
ax.set_facecolor("#FFF8DC")

# declaring the text properties in fontdict
font_props = {'fontsize': 20, 'color': 'blue',
'fontfamily':'cursive'}

# Plot text with custom font properties
plt.text(x=0.5, y=0.8, s='Highest Selling Category',
         fontdict={'fontsize': 25, 'color': '#00008B',
'fontfamily':'fantasy'},
         horizontalalignment='center')

plt.text(x=0.5, y=0.5, s=categoryname,
fontdict=font_props,horizontalalignment='center')
plt.text(x=0.5, y=0.3, s=value_fmt,
fontdict=font_props,horizontalalignment='center')

# to hide the xticks and yticks
plt.xticks([])
plt.yticks([])
```

```
plt.show()
```

Output:

7. How many customers are there in each state?

```sql
SELECT
    State,
    COUNT(*) AS TotalCustomers
FROM Customers
GROUP BY State
ORDER BY TotalCustomers desc;
```

8. What is the average quantity of each product sold?

```sql
SELECT
    s.SubcatID,
    sb.SubcatName,
    AVG(Quantity) AS AvgQuantitySold
FROM Salesdata s
join subcategories sb on s.subcatid=sb.SubcatID
GROUP BY s.SubcatID,sb.subcatname;
```

9. Which supplier has the highest total sales?

```
SELECT
    SupplierName,
    SUM(TotalPrice) AS TotalSales
FROM Salesdata s
JOIN Subcategories sc ON s.SubcatID = sc.SubcatID
JOIN Suppliers su ON sc.SupplierID = su.SupplierID
GROUP BY SupplierName
ORDER BY TotalSales DESC
LIMIT 1;
```

10. What is the total revenue generated from sales in each state?

```
SELECT
    c.State,
    SUM(s.TotalPrice) AS TotalRevenue
FROM Salesdata s
JOIN Customers c ON s.CustomerID = c.CustomerID
GROUP BY c.State;
```

11. How many sales were made by each customer?

```
SELECT
    s.CustomerID,
    c.firstname,
    COUNT(*) AS TotalSales
FROM Salesdata s
join customers c
on s.CustomerID=c.CustomerID
GROUP BY s.CustomerID,c.firstname
order by TotalSales desc;
```

12. What is the total salary expenditure on employees in each department?

```sql
SELECT
    Department,
    SUM(Salary) AS TotalSalaryExpenditure
FROM Employees
GROUP BY Department;
```

13. Which subcategory has the highest average unit price?

```sql
SELECT
    SubcatName,
    AVG(UnitPrice) AS AvgSellingPrice
FROM Subcategories
GROUP BY SubcatName
ORDER BY AvgSellingPrice DESC
LIMIT 3;
```

14. What is the total revenue generated from sales in each category?

```sql
SELECT
    c.CategoryName,
    SUM(s.TotalPrice) AS TotalRevenue
FROM Salesdata s
JOIN Subcategories sc ON s.SubcatID = sc.SubcatID
JOIN Categories c ON sc.CategoryID = c.CategoryID
GROUP BY c.CategoryName;
```