Sales Trend Analysis Report

Introduction:

Sales Trend Analysis Using MySQL

This document presents a series of SQL queries executed on the Superstore dataset to analyze sales performance trends. The goal is to uncover meaningful business insights from monthly revenue and order volume across different time periods.

MySQL's powerful aggregation functions, date extraction techniques, and sorting capabilities were used to answer key business questions. Each query was crafted to understand patterns in revenue generation, order volume, profitability, and seasonal variations.

The screenshots below show the query results for each analysis, providing a clear view of the trends across different years and months.

1. What is the total revenue generated each month?

```
31    -- 1. What is the total revenue generated each month?
32 •  SELECT
33        EXTRACT(YEAR FROM Order_Date) AS Year,
34        EXTRACT(MONTH FROM Order_Date) AS Month,
35        SUM(Sales) AS Total_Revenue
36    FROM Superstore
37    GROUP BY Year, Month
38    ORDER BY Year, Month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Year	Month	Total_Revenue
2011	1	13870.682999134064
2011	2	4410.481939911842
2011	3	55526.199467897415
2011	4	28139.56089258194
2011	5	23634.666870355606
2011	6	34508.99548578262
2011	7	33500.8727966547
2011	8	27603.51218366623
2011	9	81496.80730801821
2011	10	31394.94074845314
2011	11	78297.24052846432
2011	12	69379.83628046513
2012	1	18085.115518808365
2012	2	12183.727917432785
2012	3	38361.83618021011
2012	4	32640.482509732246
2012	5	29325.970824480057
2012	6	24659.683981239796
2012	7	28524.52092576027

2. How many distinct orders were placed each month?

```sql
-- 2. How many distinct orders were placed each month?
SELECT
    EXTRACT(YEAR FROM Order_Date) AS Year,
    EXTRACT(MONTH FROM Order_Date) AS Month,
    COUNT(DISTINCT Order_ID) AS Order_Volume
FROM Superstore
GROUP BY Year, Month
ORDER BY Year, Month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Year	Month	Order_Volume
2011	1	29
2011	2	29
2011	3	71
2011	4	66
2011	5	68
2011	6	66
2011	7	63
2011	8	72
2011	9	127
2011	10	77
2011	11	149
2011	12	141
2012	1	28
2012	2	38
2012	3	75
2012	4	72
2012	5	72
2012	6	66
2012	7	65

3. What were the top 5 highest revenue-generating months?

```sql
-- 3. What were the top 5 highest revenue-generating months?
SELECT
    EXTRACT(YEAR FROM Order_Date) AS Year,
    EXTRACT(MONTH FROM Order_Date) AS Month,
    SUM(Sales) AS Revenue
FROM Superstore
GROUP BY Year, Month
ORDER BY Revenue DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Year	Month	Revenue
2014	11	111262.02664148808
2013	12	96280.08081781864
2014	9	89121.13573098183
2014	12	87872.89762043953
2011	9	81496.80730801821
2013	11	81268.61861932278
2011	11	78297.2405284643 2
2014	10	77547.55502343178
2012	11	74699.033156991
2012	12	74478.47951388359
2013	9	71348.29879927635
2011	12	69379.83628046513
2012	9	63704.30225908756
2014	8	61038.605743050575
2013	5	56160.66828584671
2011	3	55526.199467897415
2013	10	54893.84360408783
2014	3	53775.83276462555
2013	3	50302.891105771065

4. Which month had the highest number of orders?

```sql
-- 4. Which month had the highest number of orders?
SELECT
    EXTRACT(YEAR FROM Order_Date) AS Year,
    EXTRACT(MONTH FROM Order_Date) AS Month,
    COUNT(DISTINCT Order_ID) AS Order_Count
FROM Superstore
GROUP BY Year, Month
ORDER BY Order_Count DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Year	Month	Order_Count
2014	11	250

5. Which month and year had the highest profit?

```sql
-- 5. Which month and year had the highest profit?
SELECT
    EXTRACT(YEAR FROM Order_Date) AS Year,
    EXTRACT(MONTH FROM Order_Date) AS Month,
    SUM(Profit) AS Total_Profit
FROM Superstore
GROUP BY Year, Month
ORDER BY Total_Profit DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Year	Month	Total_Profit
2013	12	17547.963401466608

6. Which year had the highest total sales?

```sql
-- 6. Which year had the highest total sales?
SELECT
    EXTRACT(YEAR FROM Order_Date) AS Year,
    SUM(Sales) AS Total_Sales
FROM Superstore
GROUP BY Year
ORDER BY Total_Sales DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Year	Total_Sales
2014	725726.326821804

7. How does revenue trend across months in a single year (e.g., 2014)?

```sql
-- 7. How does revenue trend across months in a single year (e.g., 2014)?
SELECT
    EXTRACT(MONTH FROM Order_Date) AS Month,
    SUM(Sales) AS Revenue
FROM Superstore
WHERE EXTRACT(YEAR FROM Order_Date) = 2014
GROUP BY Month
ORDER BY Month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month	Revenue
1	44592.05354332924
2	20244.702346920967
3	53775.83276462555
4	39611.0578520298
5	45485.47797238827
6	47520.58467233181
7	47654.39691078663
8	61038.605743050575
9	89121.13573098183
10	77547.55502343178
11	111262.02664148808
12	87872.89762043953

8. Which month consistently generates low sales across years?

```sql
-- 8. Which month consistently generates low sales across years?
SELECT
    Month,
    AVG(Monthly_Revenue) AS Avg_Revenue
FROM (
    SELECT
        EXTRACT(MONTH FROM Order_Date) AS Month,
        EXTRACT(YEAR FROM Order_Date) AS Year,
        SUM(Sales) AS Monthly_Revenue
    FROM Superstore
    GROUP BY Year, Month
) AS Monthly
GROUP BY Month
ORDER BY Avg_Revenue ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month	Avg_Revenue
2	14858.55584871769
1	23745.11076566577
4	34869.19898959994
6	36350.84125006199
7	36991.64765447378
5	38651.69598826766
8	39546.909589767456
10	48804.62368482351
3	49491.689879626036
9	76417.63602434099
12	82002.82355815172
11	86381.72973656654

9. What's the correlation between order volume and revenue per month?

```sql
-- 9. What's the correlation between order volume and revenue per month?
SELECT
    Year, Month, Revenue, Order_Count, (Revenue / Order_Count) AS Revenue_Per_Order
FROM (
    SELECT
        EXTRACT(YEAR FROM Order_Date) AS Year,
        EXTRACT(MONTH FROM Order_Date) AS Month,
        SUM(Sales) AS Revenue,
        COUNT(DISTINCT Order_ID) AS Order_Count
    FROM Superstore
    GROUP BY Year, Month
) AS Monthly;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Year	Month	Revenue	Order_Count	Revenue_Per_Order
2011	1	13870.682999134064	29	478.2994137632436
2011	2	4410.481939911842	29	152.08558413489112
2011	3	55526.199467897415	71	782.0591474351749
2011	4	28139.56089258194	66	426.3569832209385
2011	5	23634.666870355606	68	347.56863044640596
2011	6	34508.99548578262	66	522.8635679664034
2011	7	33500.8727966547	63	531.7598856611858
2011	8	27603.51218366623	72	383.382113662031
2011	9	81496.80730801821	127	641.7071441576237
2011	10	31394.94074845314	77	407.72650322666414
2011	11	78297.24052846432	149	525.4848357615055
2011	12	69379.83628046513	141	492.0555764571995
2012	1	18085.115518808365	28	645.8969828145845
2012	2	12183.727917432785	38	320.62441887981015
2012	3	38361.83618021011	75	511.4911490694682
2012	4	32640.482509732246	72	453.3400348573923

10. Which months should we focus marketing on (based on low order volume)?

```sql
-- 10. Which months should we focus marketing on (based on low order volume)?
SELECT
    Month,
    AVG(Order_Count) AS Avg_Orders
FROM (
    SELECT
        EXTRACT(YEAR FROM Order_Date) AS Year,
        EXTRACT(MONTH FROM Order_Date) AS Month,
        COUNT(DISTINCT Order_ID) AS Order_Count
    FROM Superstore
    GROUP BY Year, Month
) AS Monthly
GROUP BY Month
ORDER BY Avg_Orders ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month	Avg_Orders
2	40.2500
1	44.2500
7	83.0000
4	83.7500
8	84.2500
3	85.0000
6	87.0000
5	94.0000
10	102.5000
9	169.2500
12	175.2500
11	184.2500