# **Retail Sales Analysis**

By: Olutola Adunoluwa Toby

#### Aim

The analysis was performed to evaluate the sales performance of the retail company and find opportunities to increase its quarterly earnings. The analysis highlighted the top-performing segments the business should focus on and the underperforming products in terms of orders and revenue generation. The analysis also features sales and revenue forecasting based on historical data to predict future trends.

#### **Summary**

Furnishings was the best-performing category, with the highest revenue generated (\$226.8k) and sales between 2015 and 2016. However, office supplies performed the least, with about \$29.6k in revenue. The king-sized and double beds generated the highest revenues across both years. Finally, California, Virginia, and Missouri generated about 26% of the revenue and 20% of all orders, with Missouri having the highest average revenue per order.

#### **Data Model**

This data model's Entity Relationship Diagram (ERD) was drawn using the Crow's Foot Notation. The dataset consisted of three tables or entities uniquely related to each other: order details, products, and property info. The simple ERD was drawn using the following business rules:

An order could contain at least one or more products, but a product could be included in many orders or may not be ordered at all. Likewise, each order can come from one and only one location, i.e., state, but none or many orders could be made from one state.

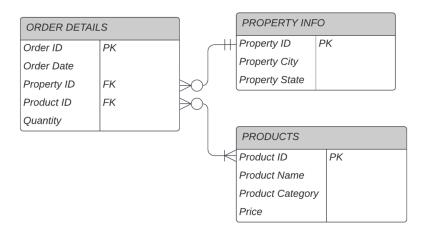


Figure 1: Data Model of the Retail Database

### **Retail Table Creation**

```
- DROP-TABLE IF EXISTS Retail;
- SELECT OrderID, OrderDate, O. ProductID,
- - - ProductName, ProductCategory,
- - - Quantity, Price, (Quantity * Price) Revenue,
- - - O. PropertyID, PropertyCity, PropertyState
- - - INTO Retail
- FROM-OrderDetails O
- LEFT JOIN Products P
- ON-o. productid = p. productid
- LEFT JOIN Propertyinfo I
- ON-o. propertyid = i. propertyid;
```

# Total Revenue, Quantity, Orders, and their Percentage Split per Segment

#### **Aggregation by Product Category**

```
*SELECT COALESCE(ProductCategory, 'Total') * ProductCategory, *SUM(Revenue) * Revenue, *
*CONCAT('%',CAST((SUM(Revenue) * 100.0)/(SELECT * SUM(Revenue) * FROM * retail) * AS * DEC(10,2))) * Rev_Percentage, *
*SUM(Quantity) * Quantity, *
*CONCAT('%',CAST((SUM(Quantity) * 100.0)/(SELECT * SUM(Quantity) * FROM * retail) * AS * DEC(10,2))) * Qty_Percentage, *
*COUNT(OrderID) * Transactions, *
*CONCAT('%',CAST((COUNT(OrderID) * 100.0)/(SELECT * COUNT(OrderID) * FROM * retail) * AS * DEC(10,2))) * Txn_Percentage * FROM * retail *
*GROUP * BY * ProductCategory * WITH * ROLLUP *
*ORDER * BY * Rev_Percentage * DESC
```

ProductCategory 🗸	Revenue 🗸	Rev_Percentage 🗸	Quantity 🗸	Qty_Percentage 🗸	Transactions 🗸	Txn_Percentage 🗸
Housekeeping	50413	%9.45	2158	%21.37	1063	%21.26
Office Supplies	29550	%5.54	1348	%13.35	664	%13.28
Furnishings	226865	%42.53	2760	%27.34	1368	%27.36
Public Areas	133076	%24.94	2234	%22.13	1101	%22.02
Maintenance	93575	%17.54	1596	%15.81	804	%16.08
Total	533479	%100.00	10096	%100.00	5000	%100.00

Figure 2: Query result showing revenue, quantity, and orders per product category



Figure 3: Visualization of revenue and orders per category on Power BI

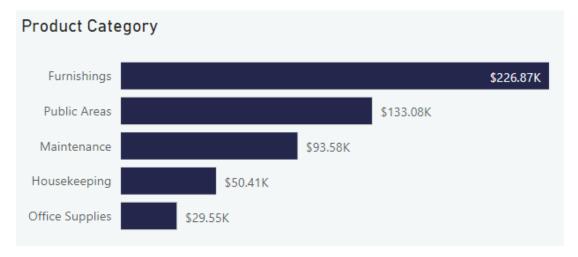


Figure 4: Revenue per product category

# **Aggregation by State**

```
By-State
--SELECT-
--COALESCE(PropertyState,'Total') - PropertyState,
--SUM(Revenue) - Revenue, --
--CONCAT('%', -CAST((SUM(Revenue) - *100.0)/(SELECT - SUM(Revenue) - FROM - retail) - AS - DEC(10,2))) - Revenue_Percentage,
--SUM(Quantity) - Quantity,
--CONCAT('%', -CAST((SUM(Quantity) - *100.0)/(SELECT - SUM(Quantity) - FROM - retail) - AS - DEC(10,2))) - Quantity_Percentage,
--COUNT(OrderID) - Transactions,
--CONCAT('%', -CAST((COUNT(OrderID) *100.0)/(SELECT - COUNT(OrderID) - FROM - retail) - AS - DEC(10,2))) - Txn_Percentage
---FROM - retail
---GROUP - BY - PropertyState - WITH - ROLLUP
---OFFSET - 1 - ROWS -
```

	PropertyState 🗸	Revenue 🗸	Revenue_Percentage 🗸	Quantity 🗸	Quantity_Percentage 🗸	Transactions 🗸	Txn_Percentage 🗸
1	Total	533479	%100.00	10096	%100.00	5000	%100.00
2	California	54529	%10.22	987	%9.78	505	%10.10
3	Virginia	53429	%10.02	1044	%10.34	504	%10.08
4	Missouri	32859	%6.16	545	%5.40	267	%5.34
5	Illinois	29218	%5.48	539	%5.34	257	%5.14
6	Louisiana	27483	%5.15	520	%5.15	260	%5.20

Figure 5:Query result showing revenue, quantity, and orders per state



Figure 6: Revenue generated per state

# Aggregation by Date - Month and Year

idx 🗸	Month 🗸	Revenue 🗸	Rev_Percentage 🗸	Quantity 🗸	Qty_Percentage 🗸	Transactions 🗸	Txn_Percentage 🗸
1	January	44965	%8.43	870	%8.62	431	%8.62
2	February	39215	%7.35	752	%7.45	381	%7.62
3	March	41601	%7.80	865	%8.57	427	%8.54
4	April	45859	%8.60	907	%8.98	436	%8.72
5	May	52814	%9.90	893	%8.85	445	%8.90
6	June	42418	%7.95	870	%8.62	431	%8.62
7	July	43818	%8.21	827	%8.19	425	%8.50
8	August	45347	%8.50	878	%8.70	436	%8.72
9	Septemb	43292	%8.12	795	%7.87	384	%7.68
10	October	41131	%7.71	842	%8.34	416	%8.32
11	November	50912	%9.54	839	%8.31	401	%8.02
12	December	42107	%7.89	758	%7.51	387	%7.74

Figure 7: Query result showing revenue, quantity, and orders per month

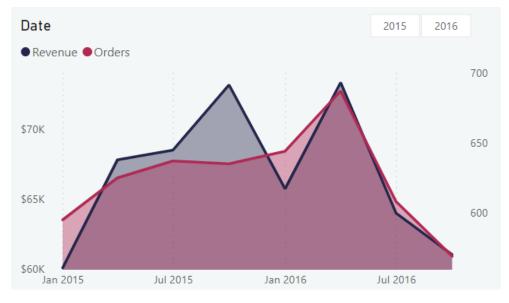


Figure 8: Time series analysis of revenue and orders

# Revenue Growth per Year

```
WITH Pivot_CTE AS

(SELECT*
FROM

(SELECT*DISTINCT*ProductCategory, DATEPART(yy,OrderDate)*Year,
SUM(Revenue)*OVER(PARTITION*BY*ProductCategory, DATEPART(yy,OrderDate))*Revenue

FROM*retail)*q
PIVOT(
```

	ProductCategory 🗸	2015 🗸	2016 🗸	Revenue_Growth 🗸
1	Furnishings	119501	107364	%-10.16
2	Housekeeping	23337	27076	%16.02
3	Maintenance	46169	47406	%2.68
4	Office Supplies	14473	15077	%4.17
5	Public Areas	65991	67085	%1.66



Figure 9: Trend analysis of average revenue for the furnishings category



Figure 10:Trend analysis of average revenue for the housekeeping category



Figure 11: Trend analysis of average revenue for the maintenance category

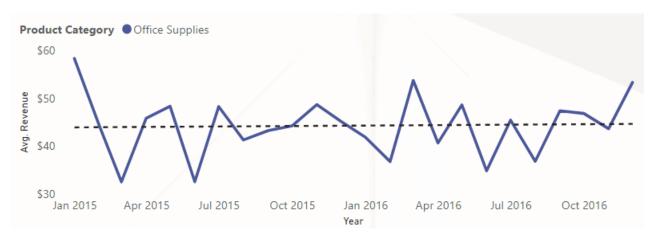


Figure 12: Trend analysis of average revenue for the office supplies category



Figure 13: Trend analysis of average revenue for the public areas category

### **Order Growth per Year**

```
SELECT ProductCategory, SUM([2015]) [2015], SUM([2016]) [2016],

CONCAT('%',CAST((SUM([2016])-SUM([2015]))*100.0/SUM([2015]) AS DEC(10,2))) Order_Growth

FROM

(SELECT DISTINCT ProductCategory,

CASE WHEN DATEPART(yy, OrderDate) = '2015' THEN COUNT(OrderID) ELSE 0 END AS [2015],

CASE WHEN DATEPART(yy, OrderDate) = '2016' THEN COUNT(OrderID) ELSE 0 END AS [2016]

FROM retail

GROUP BY ProductCategory, DATEPART(yy, OrderDate)) q

GROUP BY ProductCategory
```

	ProductCategory 🗸	2015 🗸	2016 🗸	Order_Growth 🗸
1	Furnishings	695	673	%-3.17
2	Housekeeping	501	562	%12.18
3	Maintenance	411	393	%-4.38
4	Office Supplies	325	339	%4.31
5	Public Areas	560	541	%-3.39



Figure 14: Trend analysis of orders for the furnishings category



Figure 15: Trend analysis of orders for the housekeeping category



Figure 16: Trend analysis of orders for the maintenance category



Figure 17: Trend analysis of orders for the office supplies category



Figure 18: Trend analysis of orders for the public areas category

#### **Top Products per Category**

```
SELECT DISTINCT ProductCategory,
FIRST_VALUE(Revenue) OVER(PARTITION BY ProductCategory ORDER BY Revenue DESC) Revenue,
FIRST_VALUE(ProductName) OVER(PARTITION BY ProductCategory ORDER BY Revenue DESC) Top_Product
FROM

SUM(Revenue) OVER(PARTITION BY ProductName,
SUM(Revenue) OVER(PARTITION BY ProductCategory, ProductName) Revenue,
COUNT(OrderID) OVER(PARTITION BY ProductCategory, ProductName) Quantity,
FROM PRODUCT OF COUNT(OrderID) OVER(PARTITION BY ProductCategory, ProductName) Transactions
FROM PROM Pretail)q
```

	ProductCategory 🗸	Revenue 🗸	Top_Product 🗸
1	Furnishings	32700	Bed (King)
2	Housekeeping	5160	Moisturizer
3	Maintenance	27300	Portable Drill
4	Office Supplies	7225	Printer Toner
5	Public Areas	22360	Sofa

# **Average Revenue Per Category**

```
---By Product Category
---SELECT DISTINCT ProductCategory,
---CAST(AVG(Revenue * 1.0) OVER(PARTITION BY ProductCategory) AS DEC(10,2)) AvgRevenue
---FROM Retail
----By State
----By State
----SELECT DISTINCT PropertyState,
----CAST(AVG(Revenue * 1.0) OVER(PARTITION BY PropertyState) AS DEC(10,2)) AvgRevenue
----FROM Retail
-----ORDER BY AvgRevenue DESC
```

	ProductCategory 🗸	AvgRevenue 🗸		PropertyState 🗸	AvgRevenue 🗸
1	Furnishings	165.84	1	Missouri	123.07
2	Public Areas	120.87	2	Massachusetts	114.32
3	Maintenance	116.39	3	Illinois	113.69
4	Housekeeping	47.43	4	Florida	111.11
5	Office Supplies	44.50	5	Oregon	110.06

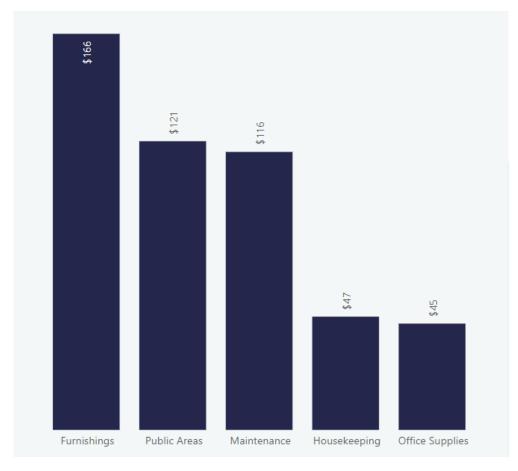


Figure 19: Average revenue per category

#### Average Revenue per Segment

```
CREATE OR ALTER PROCEDURE sp_Avg_Revenue (@segment nvarchar(15))

AS

BEGIN

COUNT(OrderID) OVER(PARTITION BY ' + @segment + ' ' ') Orders

COUNT(OrderID) OVER(PARTITION BY ' + @segment + ' ') Orders

FROM retail ORDER BY AvgRevenue DESC'

EXEC Sp_executesql @sql

END

GO

EXEC SP_Avg_Revenue 'PropertyState'
```

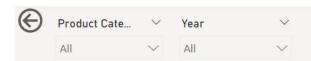
# Percentage Split for Quantity, Revenue, and Orders per Segment

```
· CREATE · OR · ALTER · PROCEDURE · sp_Percentage_split · @segment · NVARCHAR(15)
· AS
BEGIN
DECLARE @sql NVARCHAR(1000)
SET @sql == 'SELECT COALESCE( ''++ @segment ++ ', ''++ quotename('Total',''') ++ ') Segment,
SUM(Revenue) Revenue, CONCAT(''+ quotename('%',''')+'',
CAST((SUM(Revenue) *100.0)/(SELECT SUM(Revenue) FROM retail) AS DEC(10,2))) Rev_Percentage,
· · · · · · · · · · · · · · · · · SUM(Quantity) Quantity, CONCAT( · ' · + · quotename('%','''') · + · ' · ,
CAST((SUM(Quantity) *100.0)/(SELECT SUM(Quantity) FROM retail) AS DEC(10,2))) Qty Percentage,
COUNT(OrderID) Transactions, CONCAT(''+ quotename('%','''') + '',
CAST((COUNT(OrderID)*100.0)/(SELECT COUNT(OrderID) FROM retail) AS DEC(10,2))) Txn_Percentage
FROM retail
GROUP BY '++ @segment ++
ORDER BY Rev_Percentage DESC'
EXEC sp executesql @sql
END
• GO
```

#### Order Growth per Year

```
····CREATE ·OR · ALTER · PROCEDURE · sp_Order_Growth · @segment · NVARCHAR(15)
· · · · AS ·
· · · · BEGIN
DECLARE @sql NVARCHAR(1000)
SET @sql = 'SELECT *, CONCAT( ' + quotename('%',''') + ' )
  CAST((([2016]-[2015]) *100.0)/[2015] AS DEC(10,2))) Order Growth FROM
(SELECT DISTINCT '++@segment +-',
····· (SELECT COUNT(OrderID) FROM retail r1 WHERE DATEPART(yy,OrderDate) == ' ·
····· (SELECT COUNT(OrderID) FROM retail r1 WHERE DATEPART(yy,OrderDate) -- '
FROM retail r) q'
EXEC sp executesql @sql
- - - END
· · · · GO
EXEC sp Order Growth 'PropertyState'
· · · · GO
```

# Retail Dashboard

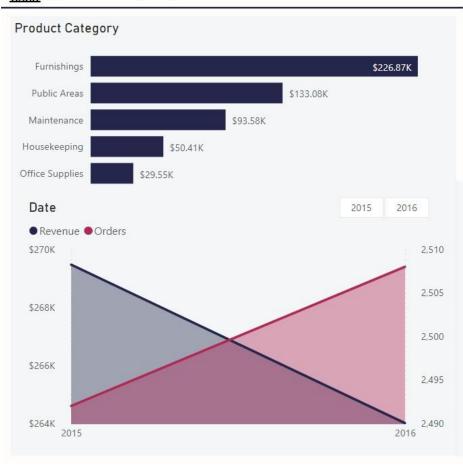




# Retail Sales Analysis



# Revenue by Segment







# Retail Dashboard

