blinkit

DAX Measures for BlinkIT Dashboard(Power BI)

1. **Average Sale:**

Average Sale = AVERAGE(blinkit_grocery_cpy[Item_Outlet_Sales])

- **Purpose:** Calculates the average sales per item.
- **Use Case:** Display on KPI cards or use as a benchmark for outlet or category comparisons.

---

2. **Avg MRP per Category:**

Avg MRP per Category = AVERAGEX('blinkit_grocery_cpy','blinkit_grocery_cpy'[Item_MRP])

- **Purpose:** Calculates the average price (MRP) of items across all categories.
- **Use Case:** Useful for pricing strategy analysis.

---

3. **Avg Revenue per Product:**

Avg Revenue per Product = AVERAGEX('blinkit_grocery_cpy','blinkit_grocery_cpy'[Item_Outlet_Sales])

- **Purpose:** Computes the average revenue for individual products.
- **Use Case:** Ideal for product performance evaluation.

---

4. **Avg_Price_Top_5:**

Avg_Price_Top_5 = AVERAGEX(TOPN(5, 'blinkit_grocery_cpy','blinkit_grocery_cpy'[Item_Outlet_Sales], DESC),'blinkit_grocery_cpy'[Item_MRP])

- **Purpose:** Calculates the average price (MRP) of the top 5 selling products.
- **Use Case:** Useful for identifying premium products that drive revenue.

---

5. **Avg_Sales_Per_Item:**

Avg_Sales_Per_Item =

DIVIDE(SUM('blinkit_grocery_cpy'[Item_Outlet_Sales]), SUM('blinkit_grocery_cpy'[Item_MRP]),0)

- **Purpose:** Determines the average sales relative to product price.
- **Use Case:** Helps evaluate pricing efficiency.

6. **Category Sales Contribution:**

Category Sales Contribution = DIVIDE(SUM('blinkit_grocery_cpy'[Item_Outlet_Sales]),

CALCULATE(SUM('blinkit_grocery_cpy'[Item_Outlet_Sales]), ALL('blinkit_grocery_cpy'))) * 100

- **Purpose:** Calculates each category's percentage contribution to total sales.
- **Use Case:** Ideal for a pie chart or treemap visualization.

---

7. **Outlet Performance:**

Outlet Performance =

SUMX('blinkit_grocery_cpy','blinkit_grocery_cpy'[Item_Outlet_Sales])

- **Purpose:** Computes total sales for each outlet.
- **Use Case:** Core metric for outlet performance comparisons.

---

8. **Revenue by Outlet Age:**

Revenue by Outlet Age =

SUMX(FILTER('blinkit_grocery_cpy','blinkit_grocery_cpy'[Outlet_Age] >= 10),
'blinkit_grocery_cpy'[Item_Outlet_Sales])

- **Purpose:** Aggregates revenue for outlets older than 10 years.
- **Use Case:** Analyze trends based on outlet age.

---

9. **Revenue by Price Tier:**

Revenue by Price Tier = SUMX('blinkit_grocery_cpy',IF('blinkit_grocery_cpy'[Price_Tier] =
"High",'blinkit_grocery_cpy'[Item_Outlet_Sales],0))

- **Purpose:** Calculates revenue generated by high-tier products.
- **Use Case:** Helps assess premium product performance.

---

10. **Sales by Outlet Size and Location:**

Sales by Outlet Size_Location =

SUMX(FILTER('blinkit_grocery_cpy','blinkit_grocery_cpy'[Outlet_Size] = "Small"
&&'blinkit_grocery_cpy'[Outlet_Location_Description] = "Urban
Area"),'blinkit_grocery_cpy'[Item_Outlet_Sales])

- **Purpose:** Measures sales in specific outlet size and location combinations.

- **Use Case:** Evaluates regional performance.

11. **Top Outlet Name:**

Top Outlet Name = VAR TopOutletTable = TOPN(1, SUMMARIZE('blinkit_grocery_cpy','blinkit_grocery_cpy'[Outlet_Identifier],"TotalSales", [Total_Sales]),[TotalSales], DESC)RETURN MAXX(TopOutletTable, 'blinkit_grocery_cpy'[Outlet_Identifier])

- **Purpose:** Identifies the outlet with the highest total sales.
- **Use Case:** Display in a KPI card or as part of a leaderboard.

12. **Top Product Category:**

Top Product Category =

VAR TopCategoryTable = TOPN(1,SUMMARIZE('blinkit_grocery_cpy','blinkit_grocery_cpy'[Item_Type],"CategoryRevenue", [Total_Sales]),[CategoryRevenue], DESC)RETURN MAXX(TopCategoryTable, 'blinkit_grocery_cpy'[Item_Type])

- **Purpose:** Finds the top-performing product category.
- **Use Case:** Use in KPI cards or charts.

13. **Top_Performing_Outlet:**

Top_Performing_Outlet = VAR TopOutletID = CALCULATE(FIRSTNONBLANK('blinkit_grocery_cpy'[Outlet_Identifier], 1),TOPN(1,SUMMARIZE('blinkit_grocery_cpy','blinkit_grocery_cpy'[Outlet_Identifier],"TotalSales", SUM('blinkit_grocery_cpy'[Item_Outlet_Sales])),[TotalSales], DESC))

VAR TopOutletSales =

CALCULATE(MAX('blinkit_grocery_cpy'[Item_Outlet_Sales]),FILTER('blinkit_grocery_cpy', 'blinkit_grocery_cpy'[Outlet_Identifier] = TopOutletID))RETURNTopOutletID & "-" & FORMAT(TopOutletSales, "0,0.00")

- **Purpose:** Combines the top outlet ID and its total sales in one string.
- **Use Case:** Display as a KPI.

14. **Worst_Performing_Outlet:**

Worst_Performing_Outlet = VAR WorstOutletID = CALCULATE(FIRSTNONBLANK('blinkit_grocery_cpy'[Outlet_Identifier], 1),TOPN(1,SUMMARIZE('blinkit_grocery_cpy','blinkit_grocery_cpy'[Outlet_Identifier],"TotalSales", SUM('blinkit_grocery_cpy'[Item_Outlet_Sales])),[TotalSales], ASC))VAR WorstOutletSales = CALCULATE(MIN('blinkit_grocery_cpy'[Item_Outlet_Sales]),FILTER('blinkit_grocery_cpy', 'blinkit_grocery_cpy'[Outlet_Identifier] = WorstOutletID))RETURNWorstOutletID & "-" & FORMAT(WorstOutletSales, "0,0.00")

- **Purpose:** Combines the worst-performing outlet ID and its sales in one string.
- **Use Case:** Display as a KPI.

15. **Total Outlets:**

Total Outlets = DISTINCTCOUNT(blinkit_grocery_cpy[Outlet_Identifier])

- **Purpose:** Counts the total number of unique outlets in the dataset.

16. **Total Sales:**

Total_Sales = SUM('blinkit_grocery_cpy'[Item_Outlet_Sales])

- **Purpose:** Aggregates the total sales across all outlets.