**Blinkit Data Analysis Report**

**1. Data Import and Cleaning**

**Data Import**

To begin the analysis, the dataset is imported using the following query:

SELECT \* FROM blinkit\_data;

**Data Cleaning**

Cleaning the Item\_Fat\_Content field ensures consistency and accuracy in the dataset. Multiple variations of the same category (e.g., 'LF', 'low fat' vs. 'Low Fat') can cause inconsistencies in reporting, aggregations, and filtering. Standardizing these values improves data quality and uniformity across analyses.

**Query to Standardize Item\_Fat\_Content**

UPDATE blinkit\_data

SET Item\_Fat\_Content =

CASE

WHEN Item\_Fat\_Content IN ('LF', 'low fat') THEN 'Low Fat'

WHEN Item\_Fat\_Content = 'reg' THEN 'Regular'

ELSE Item\_Fat\_Content

END;

After executing this update, verify the changes using:

SELECT DISTINCT Item\_Fat\_Content FROM blinkit\_data;

**2. Key Performance Indicators (KPIs)**

**2.1 Total Sales**

SELECT CAST(SUM(Total\_Sales) / 1000000.0 AS DECIMAL(10,2)) AS Total\_Sales\_Million

FROM blinkit\_data;

**2.2 Average Sales**

SELECT CAST(AVG(Total\_Sales) AS INT) AS Avg\_Sales

FROM blinkit\_data;

**2.3 Number of Items Sold**

SELECT COUNT(\*) AS No\_of\_Orders

FROM blinkit\_data;

**2.4 Average Rating**

SELECT CAST(AVG(Rating) AS DECIMAL(10,1)) AS Avg\_Rating

FROM blinkit\_data;

**3. Sales Analysis**

**3.1 Total Sales by Fat Content**

SELECT Item\_Fat\_Content, CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM blinkit\_data

GROUP BY Item\_Fat\_Content;

**3.2 Total Sales by Item Type**

SELECT Item\_Type, CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM blinkit\_data

GROUP BY Item\_Type

ORDER BY Total\_Sales DESC;

**3.3 Total Sales by Outlet Location and Fat Content**

SELECT Outlet\_Location\_Type,

ISNULL([Low Fat], 0) AS Low\_Fat,

ISNULL([Regular], 0) AS Regular

FROM

(

SELECT Outlet\_Location\_Type, Item\_Fat\_Content,

CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM blinkit\_data

GROUP BY Outlet\_Location\_Type, Item\_Fat\_Content

) AS SourceTable

PIVOT

(

SUM(Total\_Sales)

FOR Item\_Fat\_Content IN ([Low Fat], [Regular])

) AS PivotTable

ORDER BY Outlet\_Location\_Type;

**Explanation:**

* The query groups sales by Outlet\_Location\_Type and Item\_Fat\_Content.
* The PIVOT operation transforms row data into columns (Low Fat and Regular).
* NULL values are replaced with 0 for better readability.

**4. Outlet Sales Analysis**

**4.1 Total Sales by Outlet Establishment Year**

SELECT Outlet\_Establishment\_Year, CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM blinkit\_data

GROUP BY Outlet\_Establishment\_Year

ORDER BY Outlet\_Establishment\_Year;

**4.2 Percentage of Sales by Outlet Size**

SELECT

Outlet\_Size,

CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales,

CAST((SUM(Total\_Sales) \* 100.0 / SUM(SUM(Total\_Sales)) OVER()) AS DECIMAL(10,2)) AS Sales\_Percentage

FROM blinkit\_data

GROUP BY Outlet\_Size

ORDER BY Total\_Sales DESC;

**Explanation:**

* Calculates total sales per outlet size.
* Determines each outlet size’s percentage contribution to total sales.

**4.3 Total Sales by Outlet Location**

SELECT Outlet\_Location\_Type, CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM blinkit\_data

GROUP BY Outlet\_Location\_Type

ORDER BY Total\_Sales DESC;

**4.4 Comprehensive Outlet Metrics**

SELECT Outlet\_Type,

CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales,

CAST(AVG(Total\_Sales) AS DECIMAL(10,0)) AS Avg\_Sales,

COUNT(\*) AS No\_Of\_Items,

CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg\_Rating,

CAST(AVG(Item\_Visibility) AS DECIMAL(10,2)) AS Item\_Visibility

FROM blinkit\_data

GROUP BY Outlet\_Type

ORDER BY Total\_Sales DESC;

**5. Conclusion**

This report presents an analysis of sales trends, item attributes, and outlet performance within the Blinkit dataset. The queries provide insights into sales distribution by various factors, assisting in data-driven decision-making.