**Blinkit Analysis**

* See all the data imported:

SELECT \* FROM blinkit\_data

* **DATA CLEANING:**

Cleaning the Item\_Fat\_Content field ensures data consistency and accuracy in analysis. The presence of multiple variations of the same category (e.g., LF, low fat vs. Low Fat) can cause issues in reporting, aggregations, and filtering. By standardizing these values, we improve data quality, making it easier to generate insights and maintain uniformity in our datasets.

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 query check the data has been cleaned or not using below query

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



**A. KPI’s**

**1. TOTAL SALES:**

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

FROM blinkit\_data;

****

**2. AVERAGE SALES**

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

FROM blinkit\_data;

****

**3. NO OF ITEMS**

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

FROM blinkit\_data;

****

**4. AVG RATING**

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

FROM blinkit\_data;

****

**B. 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

****

**C. 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

****

**D. Fat Content by Outlet for Total Sales**

SELECT Outlet\_Location\_Type,

SUM(CASE WHEN Item\_Fat\_Content = 'Low Fat' THEN Total\_Sales ELSE 0 END) AS Low\_Fat,

SUM(CASE WHEN Item\_Fat\_Content = 'Regular' THEN Total\_Sales ELSE 0 END) AS Regular

FROM blinkit\_data

GROUP BY Outlet\_Location\_Type

ORDER BY Outlet\_Location\_Type;

****

**Query Explanations**

This query selects each Outlet\_Location\_Type and, for each type, computes two conditional sums: one summing Total\_Sales when Item\_Fat\_Content = 'Low Fat' (otherwise zero) as Low\_Fat, and another summing Total\_Sales when Item\_Fat\_Content = 'Regular' as Regular. It reads from the table blinkit\_data, groups the results by Outlet\_Location\_Type to aggregate per outlet, and finally orders the output by Outlet\_Location\_Type.

**E. Total Sales by Outlet Establishment**

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 Total\_Sales;

****

**F. 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;

**Query Explanation:**

* Select Outlet\_Size as the grouping category
* Compute total sales per outlet size: SUM(Total\_Sales) cast to decimal(10,2) as Total\_Sales
* Compute each outlet size’s share of overall sales: (sum for that size / total sum over all sizes) × 100, cast to decimal(10,2) as Sales\_Percentage
* Use OVER() window to get the grand total sum across all rows
* Group by Outlet\_Size
* Order results by descending Total\_Sales

****

**G. 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;

****

**H. All Metrics by Outlet Type:**

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;

****