

# Blinkit Analysis

See all the data imported:

-- *Selecting the database*

USE Blinkit\_Report;

SELECT \* FROM blinkit\_data

## Data Cleaning:

To ensure consistency and accuracy in analysis, the Item\_Fat\_Content field needs to be cleaned. Variations of the same category, such as LF, low fat, and Low Fat, can lead to discrepancies in reporting, filtering, and aggregations. Standardizing these values enhances data quality, streamlines analysis, and ensures uniformity across the dataset, making it easier to derive meaningful insights.

-- *Updating Item\_Fat\_Content to standardize category names*

UPDATE BlinkIT\_Data

SET Item\_Fat\_Content = CASE

-- *Replacing different variations of 'Low Fat' with a standardized value*

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

-- *Replacing shorthand 'reg' with the standardized value 'Regular'*

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

-- *Keeping all other values unchanged*

ELSE Item\_Fat\_Content

END;

After executing this query check the data has been cleaned or not using below query.

## Selecting distinct values from the Item\_Fat\_Content column

-- *This helps in identifying the unique categories present in the dataset*

SELECT DISTINCT

Item\_Fat\_Content -- *Retrieves only unique values from the Item\_Fat\_Content column*

FROM

BlinkIT\_Data; -- *Specifies the source table from which the data is being fetched*

	Item_Fat_Content
1	Low Fat
2	Regular

## KPI: Calculating Total Sales in Millions

SELECT

-- *Summing up the Total\_Sales column to get the overall sales*

-- *Dividing by 1,000,000 to convert the value into millions*

-- *Casting the result to DECIMAL(10,2) to ensure two decimal places for accuracy*

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

FROM

-- *Selecting data from the BlinkIT\_Data table*

BlinkIT\_Data;

	Total_Sales_Million
1	1.20

## KPI: Calculating Average Sales

SELECT

-- *Calculating the average value of the Total\_Sales column*

-- *Using CAST to convert the result into an integer (removing decimal values)*

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

FROM

-- *Selecting data from the BlinkIT\_Data table*

BlinkIT\_Data;

	Avg_Sales
1	140

## KPI: Counting the Total Number of Orders

SELECT

-- *COUNT(\*) counts the total number of rows in the BlinkIT\_Data table*

-- *Each row represents an order, so this gives the total number of orders*

COUNT(\*) AS No\_of\_Orders

FROM

-- *Selecting data from the BlinkIT\_Data table*

BlinkIT\_Data;

	No_of_Orders
1	8523

### KPI: Calculating the Average Rating of All Orders

SELECT

-- AVG(Rating) computes the average rating from the Rating column  
-- CAST(... AS DECIMAL(10,1)) ensures the result is displayed with one decimal place  
CAST(AVG(Rating) AS DECIMAL(10,1)) AS Avg\_Rating

FROM

-- Selecting data from the BlinkIT\_Data table  
BlinkIT\_Data;

Results		Messages	
	Avg_Rating		
1	4.0		

### Total Sales by Fat Content

SELECT

-- Selecting the fat content category  
Item\_Fat\_Content,  
  
-- Calculating total sales for each fat content category  
-- SUM(Total\_Sales) aggregates the total sales per fat content type  
-- CAST(... AS DECIMAL(10,2)) ensures the result is displayed with two decimal places  
CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM

-- Fetching data from the BlinkIT\_Data table  
BlinkIT\_Data

GROUP BY

-- Grouping the results by fat content to calculate sales per category  
Item\_Fat\_Content;

Results		Messages	
	Item_Fat_Content	Total_Sales	
1	Low Fat	776319.68	
2	Regular	425361.80	

### Total Sales by Item Type (sorted in descending order)

SELECT

-- Selecting the item type category  
Item\_Type,  
  
-- Calculating total sales for each item type  
-- SUM(Total\_Sales) aggregates the total sales per item type  
-- CAST(... AS DECIMAL(10,2)) ensures the result is displayed with two decimal places  
CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM

-- Fetching data from the BlinkIT\_Data table  
BlinkIT\_Data

GROUP BY

-- Grouping the results by item type to calculate sales per category  
Item\_Type

ORDER BY

-- Sorting the results in descending order to show item types with the highest sales first  
Total\_Sales DESC;

Results		Messages	
	Item_Type	Total_Sales	
1	Fruits and Vegetables	178124.08	
2	Snack Foods	175433.92	
3	Household	135976.53	
4	Frozen Foods	118558.88	
5	Dairy	101276.46	
6	Canned	90706.73	
7	Baking Goods	81894.74	
8	Health and Hygiene	68025.84	
9	Meat	59449.86	
10	Soft Drinks	58514.16	
11	Breads	35379.12	
12	Hard Drinks	29334.68	
13	Others	22451.89	
14	Starchy Foods	21880.03	
15	Breakfast	15596.70	
16	Seafood	9077.87	

### Fat Content by Outlet for Total Sales (Pivot Table)

```
SELECT
  -- Selecting the outlet location type
  Outlet_Location_Type,

  -- Using ISNULL() to replace NULL values with 0 for better readability
  ISNULL([Low Fat], 0) AS Low_Fat,
  ISNULL([Regular], 0) AS Regular
FROM (
  -- Subquery to calculate total sales based on outlet location type and fat content
  SELECT
    -- Selecting outlet location type
    Outlet_Location_Type,

    -- Selecting item fat content category
    Item_Fat_Content,

    -- Aggregating total sales per location type and fat content
    -- SUM(Total_Sales) computes the total sales
    -- CAST(... AS DECIMAL(10,2)) ensures the result has two decimal places
    CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales
  FROM
    -- Fetching data from the BlinkIT_Data table
    BlinkIT_Data
  GROUP BY
    -- Grouping by outlet location type and fat content category
    Outlet_Location_Type, Item_Fat_Content
) AS SourceTable
PIVOT (
  -- Pivoting data to create separate columns for 'Low Fat' and 'Regular' categories
  SUM(Total_Sales)
  FOR Item_Fat_Content IN ([Low Fat], [Regular])
) AS PivotTable
ORDER BY
  -- Sorting results alphabetically by outlet location type
  Outlet_Location_Type;
```

	Outlet_Location_Type	Low_Fat	Regular
1	Tier 1	215047.91	121349.90
2	Tier 2	254464.77	138685.87
3	Tier 3	306806.99	165326.03

#### Why Use ISNULL?

During a PIVOT operation, if there is no data for a specific combination of Outlet\_Location\_Type and Item\_Fat\_Content, the corresponding cell will return a NULL value. Applying ISNULL(column) ensures that these NULL values are replaced with 0, improving data readability and consistency.

### Total Sales by Outlet Establishment Year

```
-- Selecting the year when each outlet was established
SELECT Outlet_Establishment_Year,

  -- Calculating the total sales for each establishment year
  -- SUM(Total_Sales) aggregates the total sales for each year
  -- CAST ensures the result is formatted as a decimal with two decimal places
  CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales

-- Fetching data from the BlinkIT_Data table
FROM BlinkIT_Data

-- Grouping sales data by the establishment year of the outlets
GROUP BY Outlet_Establishment_Year

-- Sorting results in ascending order to display sales trends over time
ORDER BY Outlet_Establishment_Year;
```

	Outlet_Size	Total_Sales	Sales_Percentage
1	Medium	507895.73	42.27
2	Small	444794.17	37.01
3	High	248991.58	20.72

## All Metrics by Outlet Type

-- Selecting the outlet type for analysis

SELECT Outlet\_Type,

-- Calculating the total sales for each outlet type

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

-- Computing the average sales per item in each outlet type

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

-- Counting the total number of items sold for each outlet type

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

-- Calculating the average rating for each outlet type

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

-- Computing the average visibility of items for each outlet type

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

-- Fetching data from the BlinkIT\_Data table

FROM BlinkIT\_Data

-- Grouping sales data by outlet type

GROUP BY Outlet\_Type

-- Sorting in descending order to show the highest total sales first

ORDER BY Total\_Sales DESC;

Results		Messages				
	Outlet_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating	Item_Visibility
1	Supermarket Type1	787549.89	141	5577	3.96	0.06
2	Grocery Store	151939.15	140	1083	3.99	0.10
3	Supermarket Type2	131477.77	142	928	3.97	0.06
4	Supermarket Type3	130714.67	140	935	3.95	0.06