#### **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

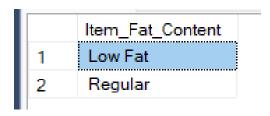
WHEN Item_Fat_Content IN ('LF', 'low fat') THEN 'Low Fat'

WHEN Item_Fat_Content = 'reg' THEN 'Regular'

ELSE Item_Fat_Content

END;
```

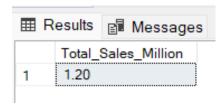
After executing this query check the data has been cleaned or not using below query SELECT DISTINCT Item Fat Content FROM blinkit data;



#### A. KPI's

#### 1. TOTAL SALES:

select cast(sum(Sales)/1000000 As decimal(10,2)) as Total\_Sales\_Millions from blinkit\_data;



### 2. AVERAGE SALES

select cast(avg(Sales) as decimal(10,0)) 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(Sales)as DECIMAL(10,2)) AS Total\_Sales

## FROM blinkit\_data

# GROUP BY Item\_Fat\_Content;

Item_Fat_Content Total_Sale	
	s
1 Low Fat 776319.6	3
2 Regular 425361.8	)

## C. Total Sales by Item Type

SELECT Item\_Type, CAST(SUM(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,

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

⊞ F	Results	■ Messages		
	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

## E. Total Sales by Outlet Establishment

```
SELECT Outlet_Establishment_Year, CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales
FROM blinkit_data
GROUP BY Outlet_Establishment_Year
ORDER BY Outlet_Establishment_Year
```

■ Results						
	Outlet_Establishment_Year2	Total_Sales				
1	1998	204522.26				
2	2000	131809.02				
3	2010	132113.37				
4	2011	78131.56				
5	2012	130476.86				
6	2015	130942.78				
7	2017	133103.91				
8	2020	129103.96				
9	2022	131477.77				

# F. Percentage of Sales by Outlet Size

**SELECT** 

Outlet\_Size,

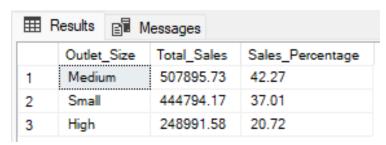
CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total\_Sales,

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

FROM blinkit\_data

**GROUP BY Outlet Size** 

ORDER BY Total\_Sales DESC;



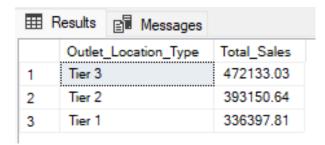
### **G.** Sales by Outlet Location

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

CAST(AVG(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

⊞ Results									
	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			