

Blinkit Queries

- See all 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 ;
```

	Item_Fat_Content
1	Low Fat
2	Regular

A.KPI's

1.TOTAL SALES

```
SELECT CAST(SUM(Total_Sales) / 1000000.0 AS DECIMAL(10,2)) AS
Total_Sales_Million
```

```
FROM blinkit_data;
```

Results		Messages
Total_Sales_Million		
1	1.20	

2. AVERAGE SALES

```
SELECT CAST(AVG(Total_Sales) AS INT) AS Avg_Sales  
FROM blinkit_data;
```

Results		Messages
Avg_Sales		
1	140	

3. NO OF ITEMS

```
SELECT COUNT(*) AS No_of_Orders  
FROM blinkit_data;
```

Results		Messages
No_of_Orders		
1	8523	

4. AVG RATING

```
SELECT CAST(AVG(Rating) AS DECIMAL(10,1)) AS Avg_Rating  
FROM blinkit_data;
```

Results		Messages
Avg_Rating		
1	4.0	

B. 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

	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

C.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
```

	Item_Fat_Content	Total_Sales
1	Low Fat	776319.68
2	Regular	425361.80

D. FAT CONTENT BY OUTLET FOR TOTAL SALES

```
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;

```

	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(Total_Sales) AS
DECIMAL(10,2)) AS Total_Sales
FROM blinkit_data
GROUP BY Outlet_Establishment_Year
ORDER BY Outlet_Establishment_Year

```

	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. 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
```

A screenshot of a SQL query results window. At the top, there are two tabs: "Results" (highlighted in orange) and "Messages". Below the tabs is a table with three rows of data. The table has three columns: "Outlet_Location_Type" (containing values 1, 2, and 3), "Total_Sales" (containing values 472133.03, 393150.64, and 336397.81), and a row number column (containing values 1, 2, and 3). The first row (Tier 3) is highlighted with a dashed border.

	Outlet_Location_Type	Total_Sales
1	Tier 3	472133.03
2	Tier 2	393150.64
3	Tier 1	336397.81