

Blinkit Analysis

- See all the data imported:

```
SELECT * FROM new_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

```
UPDATE new_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 new_data
```

	item_fat_content character varying (50) 
1	Regular
2	Low Fat

A. KPI's

1. TOTAL SALES:

```
SELECT ROUND(SUM(sales)) AS Total_Sales  
FROM new_data;
```

	total_sales numeric 
1	1201681

2. AVERAGE SALES

```
SELECT ROUND(AVG(sales), 2) AS Avg_Sales
FROM new_data;
```

	avg_sales numeric 🔒
1	140.99

3. NO OF ITEMS

```
SELECT COUNT(item_identifier) AS Count_Of_Items
FROM new_data;
```

	count_of_items bigint 🔒
1	8523

4. AVG RATING

```
SELECT ROUND(AVG(rating), 2) AS Average_Rating
FROM new_data;
```

	average_rating numeric 🔒
1	3.97

B. Total Sales by Fat Content:

```
SELECT
    item_fat_content,
    ROUND(SUM(sales)) AS Total_Sales,
    ROUND(AVG(sales)) AS Avg_Sales,
    COUNT(item_identifier) AS Item_Count,
    ROUND(AVG(rating), 2) AS Average_Rating
FROM new_data
GROUP BY item_fat_content
ORDER BY Total_Sales DESC;
```

	item_fat_content character varying (50) 🔒	total_sales numeric 🔒	avg_sales numeric 🔒	item_count bigint 🔒	average_rating numeric 🔒
1	Low Fat	776320	141	5517	3.97
2	Regular	425362	142	3006	3.97

C. Total Sales by Item Type

```
SELECT
    item_type,
    ROUND(SUM(sales)) AS Total_Sales,
    ROUND(AVG(sales)) AS Avg_Sales,
    COUNT(item_identifier) AS Item_Count,
    ROUND(AVG(rating), 2) AS Average_Rating
FROM new_data
GROUP BY item_type
ORDER BY Total_Sales DESC;
```

	item_type character varying (100) 🔒	total_sales numeric 🔒	avg_sales numeric 🔒	item_count bigint 🔒	average_rating numeric 🔒
1	Fruits and Vegetables	178124	145	1232	3.96
2	Snack Foods	175434	146	1200	3.95
3	Household	135977	149	910	4.00
4	Frozen Foods	118559	139	856	3.97
5	Dairy	101276	148	682	3.97
6	Canned	90707	140	649	3.99
7	Baking Goods	81895	126	648	3.98
8	Health and Hygiene	68026	131	520	3.99
9	Meat	59450	140	425	4.02
10	Soft Drinks	58514	131	445	3.92
11	Breads	35379	141	251	3.88
12	Hard Drinks	29335	137	214	3.91
13	Others	22452	133	169	3.95
14	Starchy Foods	21880	148	148	3.92
15	Breakfast	15597	142	110	3.93
16	Seafood	9078	142	64	3.96

D. Fat Content by Outlet for Total Sales

```
SELECT
    outlet_location_type,
    item_fat_content,
```

```

        ROUND(SUM(sales)) AS Total_Sales,
        ROUND(AVG(sales)) AS Avg_Sales,
        COUNT(item_identifier) AS Item_Count,
        ROUND(AVG(rating), 2) AS Average_Rating
FROM new_data
GROUP BY outlet_location_type, item_fat_content
ORDER BY Total_Sales DESC;

```

	outlet_location_type character varying (50) 🔒	item_fat_content character varying (50) 🔒	total_sales numeric 🔒	avg_sales numeric 🔒	item_count bigint 🔒	average_rating numeric 🔒
1	Tier 3	Low Fat	306807	142	2168	3.96
2	Tier 2	Low Fat	254465	141	1809	3.97
3	Tier 1	Low Fat	215048	140	1540	3.98
4	Tier 3	Regular	165326	140	1182	3.97
5	Tier 2	Regular	138686	142	976	3.95
6	Tier 1	Regular	121350	143	848	3.97

E. Total Sales by Outlet Establishment

```

SELECT
    outlet_establishment_year,
    ROUND(SUM(sales)) AS Total_Sales,
    ROUND(AVG(sales)) AS Avg_Sales,
    COUNT(item_identifier) AS Item_Count,
    ROUND(AVG(rating), 2) AS Average_Rating
FROM new_data
GROUP BY outlet_establishment_year
ORDER BY outlet_establishment_year ASC;

```

	outlet_establishment_year integer	total_sales numeric	avg_sales numeric	item_count bigint	average_rating numeric
1	2011	78132	141	555	3.98
2	2012	130477	140	930	3.99
3	2014	131809	141	932	3.95
4	2015	130943	141	929	3.96
5	2016	132113	142	930	3.96
6	2017	133104	143	930	3.94
7	2018	204522	140	1463	3.97
8	2020	129104	139	926	3.98
9	2022	131478	142	928	3.97

F. Percentage of Sales by Outlet Size

```

SELECT
    outlet_size,
    ROUND(SUM(sales), 2) AS Total_Sales,
    ROUND(SUM(sales) * 100.0 / SUM(SUM(sales)) OVER (), 2) AS
Sales_Percentage
FROM new_data
GROUP BY outlet_size
ORDER BY Total_Sales DESC;

```

	outlet_size character varying (20)	total_sales numeric	sales_percentage numeric
1	Medium	507895.74	42.27
2	Small	444794.17	37.01
3	High	248991.59	20.72

G. Sales by Outlet Location

```
SELECT
    outlet_location_type,
    ROUND(SUM(sales)) AS Total_Sales,
    ROUND(AVG(sales)) AS Avg_Sales,
    COUNT(item_identifier) AS Item_Count,
    ROUND(SUM(sales) * 100.0 / SUM(SUM(sales)) OVER (), 2) AS
Sales_Percentage,
    ROUND(AVG(rating), 2) AS Average_Rating
FROM new_data
GROUP BY outlet_location_type
ORDER BY Total_Sales DESC;
```

	outlet_location_type character varying (50) 🔒	total_sales numeric 🔒	avg_sales numeric 🔒	item_count bigint 🔒	sales_percentage numeric 🔒	average_rating numeric 🔒
1	Tier 3	472133	141	3350	39.29	3.96
2	Tier 2	393151	141	2785	32.72	3.96
3	Tier 1	336398	141	2388	27.99	3.98

H. All Metrics by Outlet Type:

```
SELECT
    outlet_type,
    ROUND(SUM(sales)) AS Total_Sales,
    ROUND(AVG(sales)) AS Avg_Sales,
    COUNT(item_identifier) AS Item_Count,
    ROUND(SUM(sales) * 100.0 / SUM(SUM(sales)) OVER (), 2) AS
Sales_Percentage,
    ROUND(AVG(rating), 2) AS Average_Rating
FROM new_data
GROUP BY outlet_type
ORDER BY Total_Sales DESC;
```

	outlet_type character varying (100) 🔒	total_sales numeric 🔒	avg_sales numeric 🔒	item_count bigint 🔒	sales_percentage numeric 🔒	average_rating numeric 🔒
1	Grocery Store	151939	140	1083	12.64	3.99
2	Supermarket Type1	787550	141	5577	65.54	3.96
3	Supermarket Type2	131478	142	928	10.94	3.97
4	Supermarket Type3	130715	140	935	10.88	3.95