

Sales Analysis

- See all the data imported:

```
SELECT * FROM sales_data
```

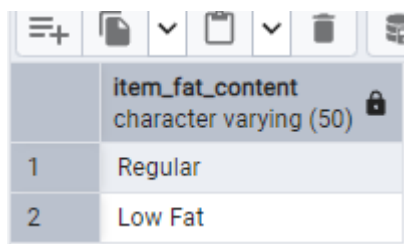
- **DATA CLEANING:**

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 sales_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 sales_data;
```



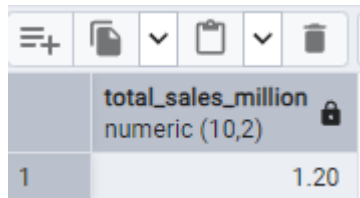
The screenshot shows a database interface with a toolbar at the top containing icons for menu, copy, dropdown, paste, delete, and refresh. Below the toolbar is a table with the following structure:

	item_fat_content character varying (50)
1	Regular
2	Low Fat

A. KPI's

1. TOTAL SALES:

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

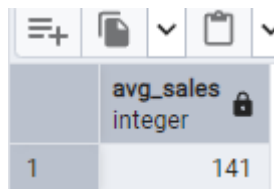


A screenshot of a database query result table. The table has two columns: an index column and a data column. The data column header is 'total_sales_million' with a data type of 'numeric (10,2)' and a lock icon. The first row has the value '1.20'.

	total_sales_million numeric (10,2)
1	1.20

2. AVERAGE SALES

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

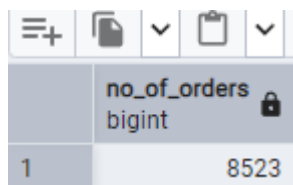


A screenshot of a database query result table. The table has two columns: an index column and a data column. The data column header is 'avg_sales' with a data type of 'integer' and a lock icon. The first row has the value '141'.

	avg_sales integer
1	141

3. NO OF ITEMS

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

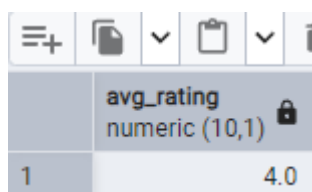


A screenshot of a database query result table. The table has two columns: an index column and a data column. The data column header is 'no_of_orders' with a data type of 'bigint' and a lock icon. The first row has the value '8523'.

	no_of_orders bigint
1	8523

4. AVG RATING

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

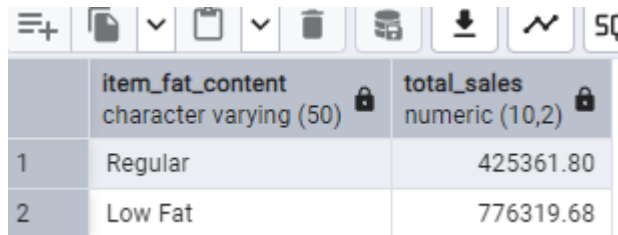


A screenshot of a database query result table. The table has two columns: an index column and a data column. The data column header is 'avg_rating' with a data type of 'numeric (10,1)' and a lock icon. The first row has the value '4.0'.

	avg_rating numeric (10,1)
1	4.0

B. Total Sales by Fat Content:

```
SELECT Item_Fat_Content, CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS  
Total_Sales  
FROM sales_data  
GROUP BY Item_Fat_Content
```



	item_fat_content character varying (50)	total_sales numeric (10,2)
1	Regular	425361.80
2	Low Fat	776319.68

C. Total Sales by Item Type

```
SELECT Item_Type, CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales  
FROM sales_data  
GROUP BY Item_Type  
ORDER BY Total_Sales DESC
```

	item_type character varying (50) 🔒	total_sales numeric (10,2) 🔒
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	58514.17
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

E. Total Sales by Outlet Establishment

```

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

```

	outlet_establishment_year integer	total_sales numeric (10,2)
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(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 sales_data

GROUP BY Outlet_Size

ORDER BY Total_Sales DESC;

	outlet_size character varying (20)	total_sales numeric (10,2)	sales_percentage numeric (10,2)
1	Medium	507895.73	42.27
2	Small	444794.17	37.01
3	High	248991.58	20.72

G. Sales by Outlet Location

SELECT Outlet_Location_Type, CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS

Total_Sales

FROM sales_data

GROUP BY Outlet_Location_Type

ORDER BY Total_Sales DESC

	outlet_location_type character varying (20) 🔒	total_sales numeric (10,2) 🔒
1	Tier 3	472133.03
2	Tier 2	393150.64
3	Tier 1	336397.81

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 sales_data
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC

```

	outlet_type character varying (50) 🔒	total_sales numeric (10,2) 🔒	avg_sales numeric (10) 🔒	no_of_items bigint 🔒	avg_rating numeric (10,2) 🔒	item_visibility numeric (10,2) 🔒
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