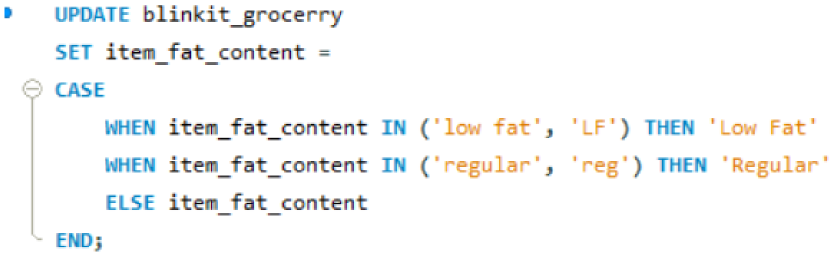
Blinkit Analysis

- See all the data imported:

  ```
  SELECT * FROM blinkit_data
  ```

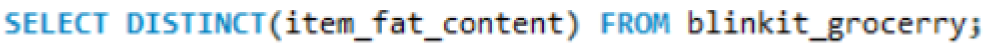
- **DATA CLEANING:**

  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.

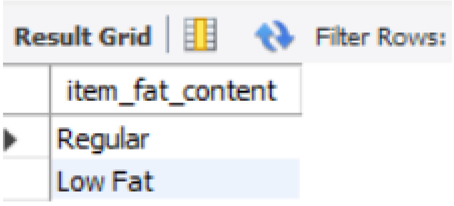


```
UPDATE blinkit_grocerry
SET item_fat_content =
CASE
    WHEN item_fat_content IN ('low fat', 'LF') THEN 'Low Fat'
    WHEN item_fat_content IN ('regular', 'reg') THEN 'Regular'
    ELSE item_fat_content
END;
```

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



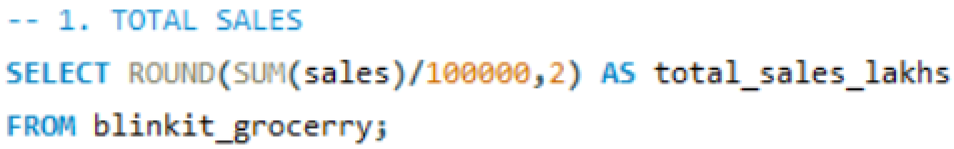
```
SELECT DISTINCT(item_fat_content) FROM blinkit_grocerry;
```



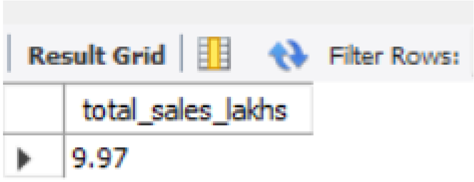
```
item_fat_content
Regular
Low Fat
```

A. KPI’s

1. TOTAL SALES:

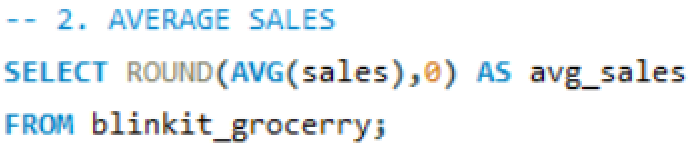


```
-- 1. TOTAL SALES
SELECT ROUND(SUM(sales)/100000,2) AS total_sales_lakhs
FROM blinkit_grocerry;
```

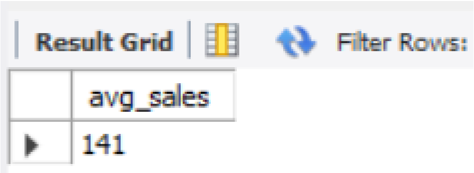


total_sales_lakhs
9.97

2. AVERAGE SALES

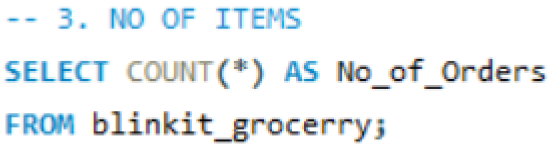


```
-- 2. AVERAGE SALES
SELECT ROUND(AVG(sales),0) AS avg_sales
FROM blinkit_grocerry;
```

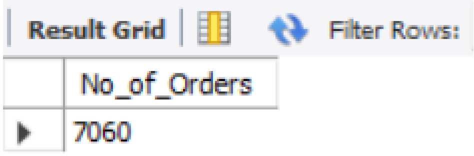


avg_sales
141

3. NO OF ITEMS

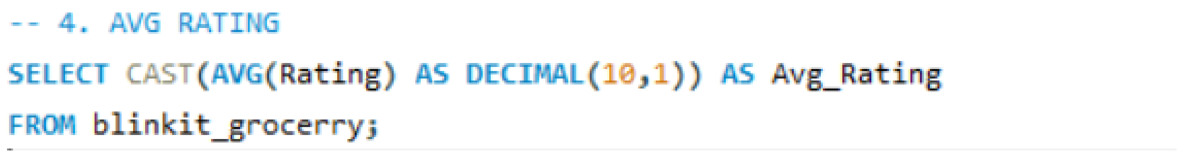


```
-- 3. NO OF ITEMS
SELECT COUNT(*) AS No_of_Orders
FROM blinkit_grocerry;
```

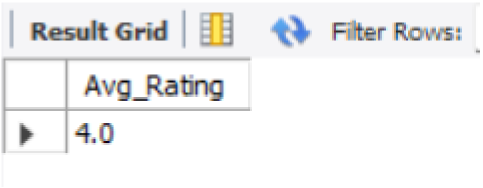


No_of_Orders
7060

4. AVG RATING

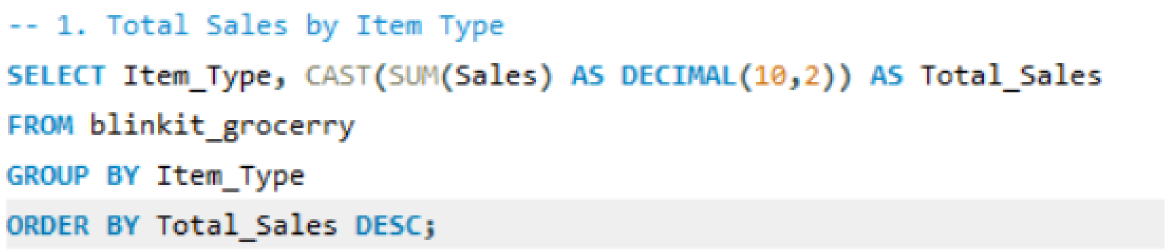


```sql
-- 4. AVG RATING
SELECT CAST(AVG(Rating) AS DECIMAL(10,1)) AS Avg_Rating
FROM blinkit_grocerry;
```

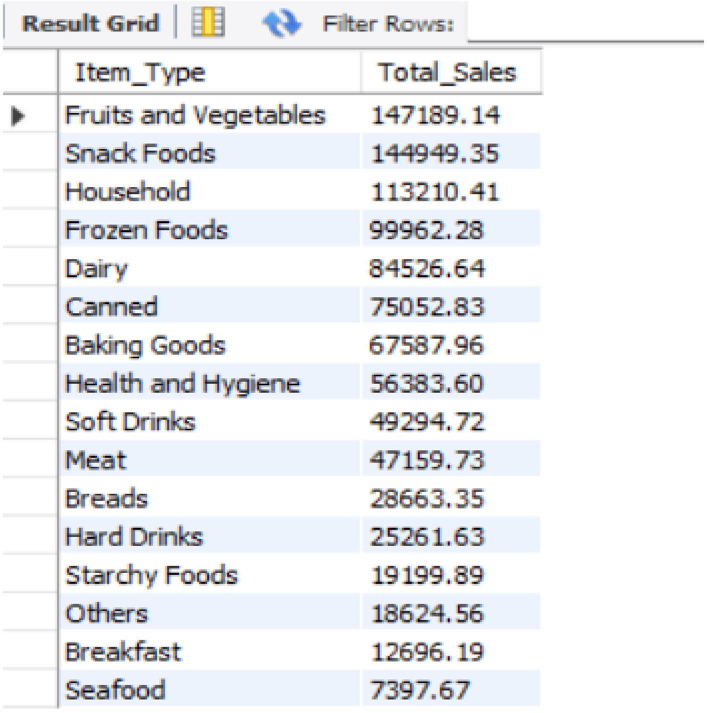


	Avg_Rating
▶	4.0

1. Total Sales by Item Type:

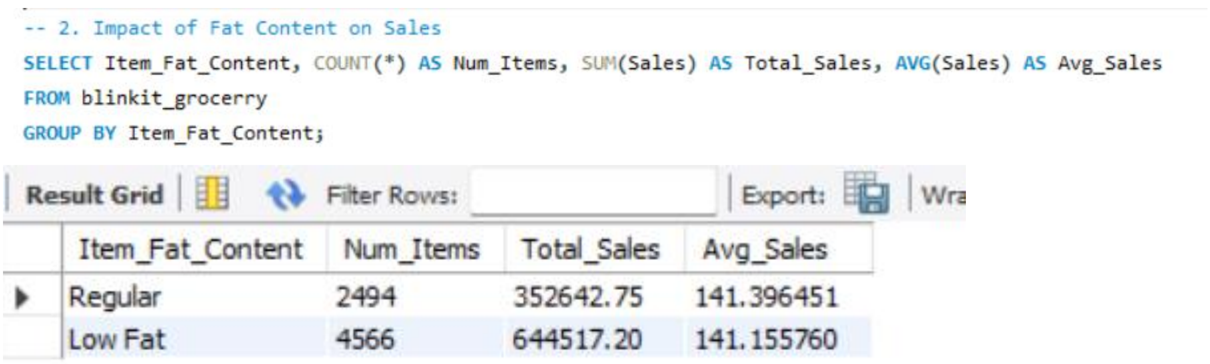


```sql
-- 1. Total Sales by Item Type
SELECT Item_Type, CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales
FROM blinkit_grocerry
GROUP BY Item_Type
ORDER BY Total_Sales DESC;
```



	Item_Type	Total_Sales
▶	Fruits and Vegetables	147189.14
	Snack Foods	144949.35
	Household	113210.41
	Frozen Foods	99962.28
	Dairy	84526.64
	Canned	75052.83
	Baking Goods	67587.96
	Health and Hygiene	56383.60
	Soft Drinks	49294.72
	Meat	47159.73
	Breads	28663.35
	Hard Drinks	25261.63
	Starchy Foods	19199.89
	Others	18624.56
	Breakfast	12696.19
	Seafood	7397.67

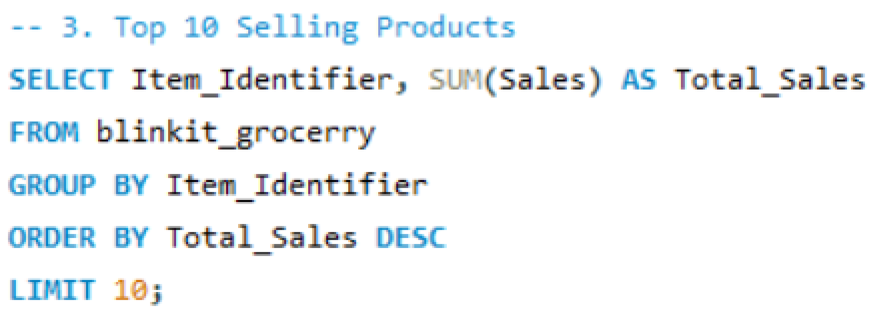
2. Impact of Fat Content on Sales



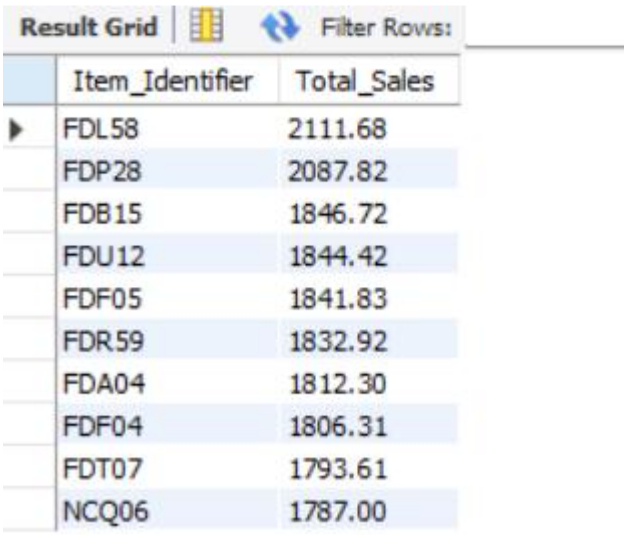
```
-- 2. Impact of Fat Content on Sales
SELECT Item_Fat_Content, COUNT(*) AS Num_Items, SUM(Sales) AS Total_Sales, AVG(Sales) AS Avg_Sales
FROM blinkit_grocerry
GROUP BY Item_Fat_Content;
```

Item_Fat_Content	Num_Items	Total_Sales	Avg_Sales
Regular	2494	352642.75	141.396451
Low Fat	4566	644517.20	141.155760

3. Top 10 Selling Products

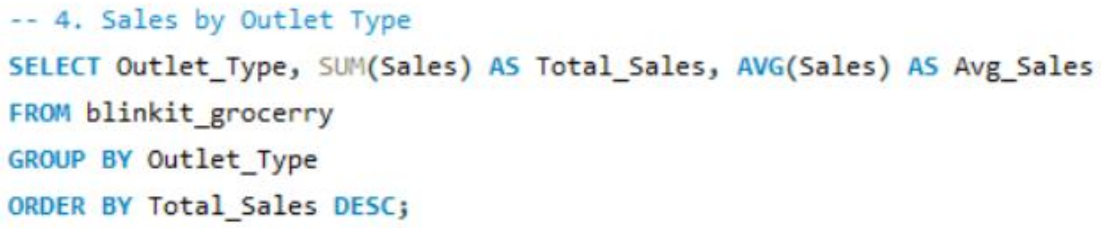


```
-- 3. Top 10 Selling Products
SELECT Item_Identifier, SUM(Sales) AS Total_Sales
FROM blinkit_grocerry
GROUP BY Item_Identifier
ORDER BY Total_Sales DESC
LIMIT 10;
```

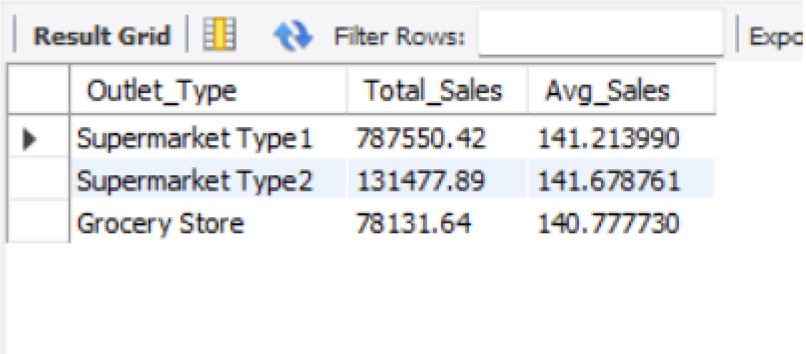


Item_Identifier	Total_Sales
FDL58	2111.68
FDP28	2087.82
FDB15	1846.72
FDU12	1844.42
FDF05	1841.83
FDR59	1832.92
FDA04	1812.30
FDF04	1806.31
FDT07	1793.61
NCQ06	1787.00

4.. Total Sales by Outlet Type

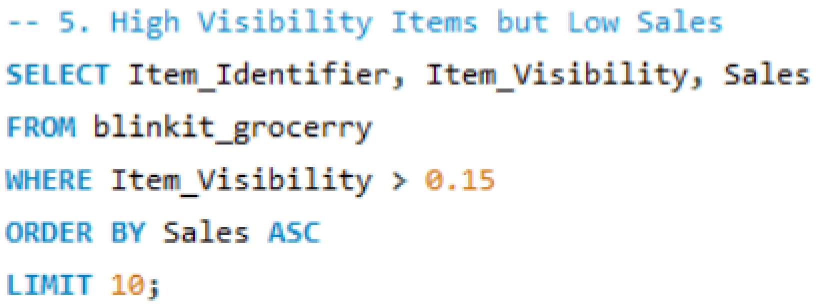


```
-- 4. Sales by Outlet Type
SELECT Outlet_Type, SUM(Sales) AS Total_Sales, AVG(Sales) AS Avg_Sales
FROM blinkit_grocerry
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC;
```

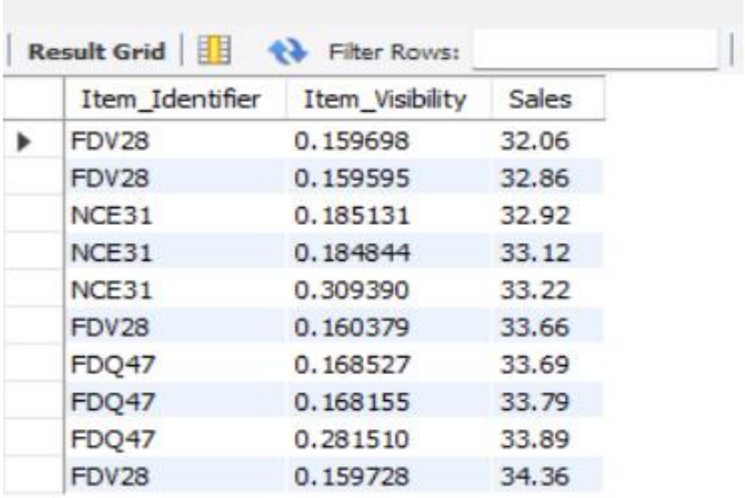


Outlet_Type	Total_Sales	Avg_Sales
Supermarket Type1	787550.42	141.213990
Supermarket Type2	131477.89	141.678761
Grocery Store	78131.64	140.777730

5. High visibility Items but Low Sales

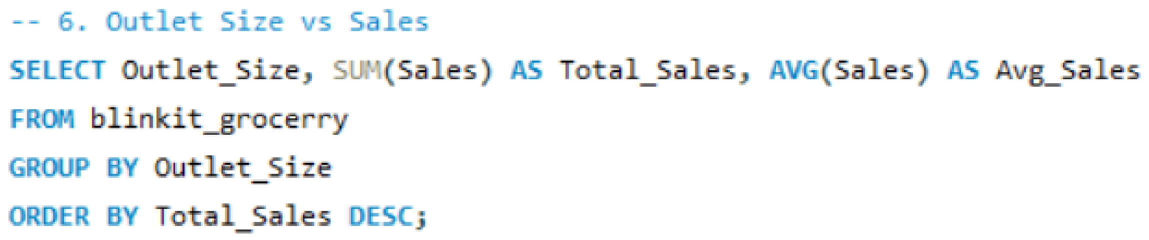


```
-- 5. High Visibility Items but Low Sales
SELECT Item_Identifier, Item_Visibility, Sales
FROM blinkit_grocerry
WHERE Item_Visibility > 0.15
ORDER BY Sales ASC
LIMIT 10;
```

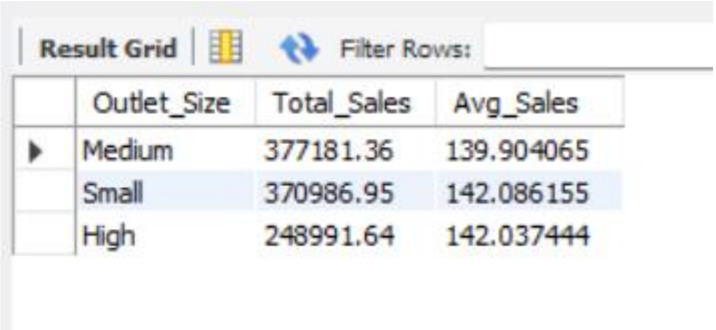


Item_Identifier	Item_Visibility	Sales
FDV28	0.159698	32.06
FDV28	0.159595	32.86
NCE31	0.185131	32.92
NCE31	0.184844	33.12
NCE31	0.309390	33.22
FDV28	0.160379	33.66
FDQ47	0.168527	33.69
FDQ47	0.168155	33.79
FDQ47	0.281510	33.89
FDV28	0.159728	34.36

6. Outlet Size vs Sales

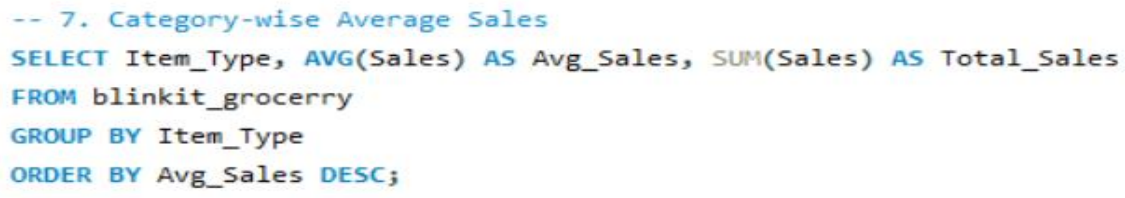


```
-- 6. Outlet Size vs Sales
SELECT Outlet_Size, SUM(Sales) AS Total_Sales, AVG(Sales) AS Avg_Sales
FROM blinkit_grocerry
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC;
```

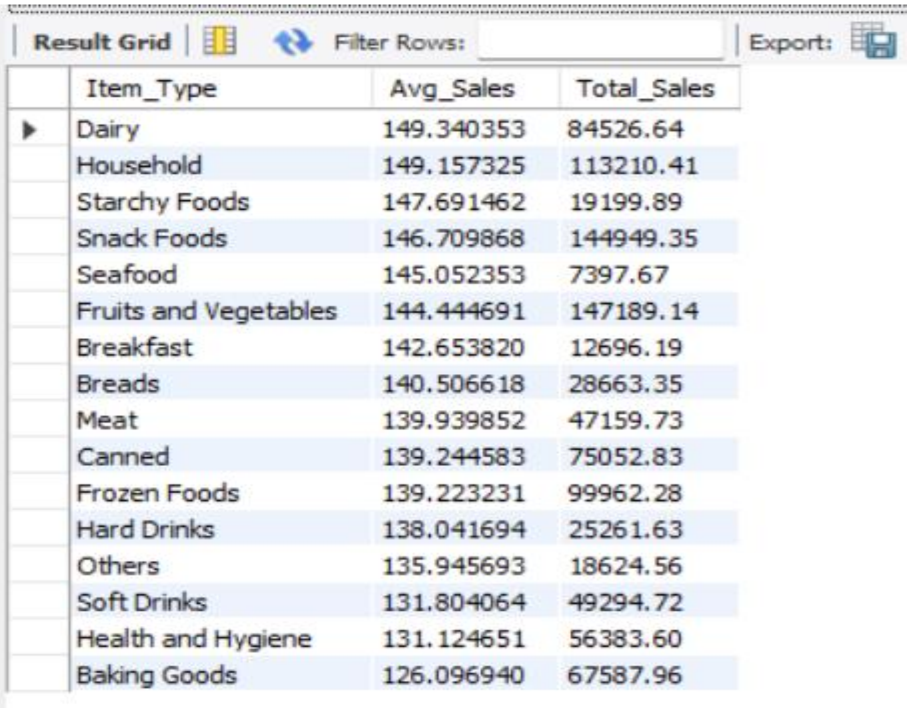


Outlet_Size	Total_Sales	Avg_Sales
Medium	377181.36	139.904065
Small	370986.95	142.086155
High	248991.64	142.037444

7.Category wise Average Sales:

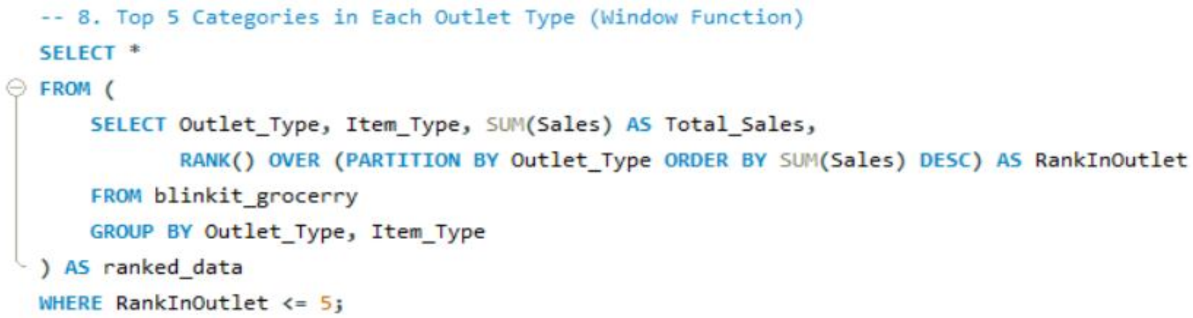


```
-- 7. Category-wise Average Sales
SELECT Item_Type, AVG(Sales) AS Avg_Sales, SUM(Sales) AS Total_Sales
FROM blinkit_grocerry
GROUP BY Item_Type
ORDER BY Avg_Sales DESC;
```

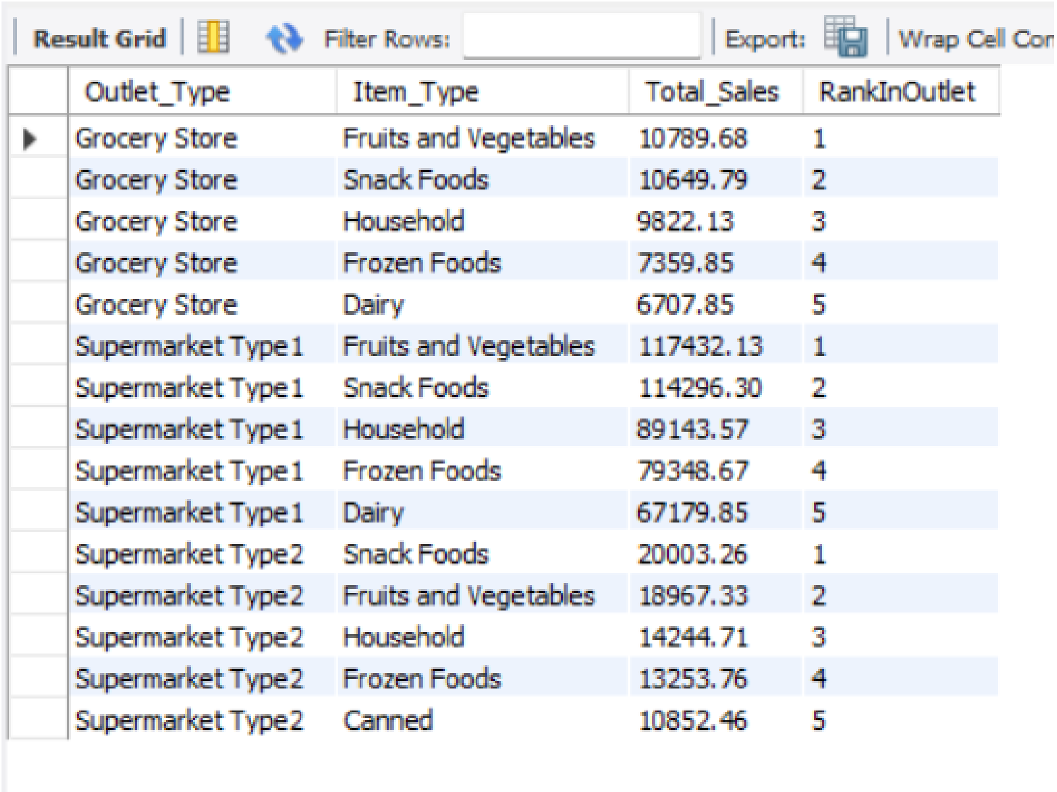


Item_Type	Avg_Sales	Total_Sales
Dairy	149.340353	84526.64
Household	149.157325	113210.41
Starchy Foods	147.691462	19199.89
Snack Foods	146.709868	144949.35
Seafood	145.052353	7397.67
Fruits and Vegetables	144.444691	147189.14
Breakfast	142.653820	12696.19
Breads	140.506618	28663.35
Meat	139.939852	47159.73
Canned	139.244583	75052.83
Frozen Foods	139.223231	99962.28
Hard Drinks	138.041694	25261.63
Others	135.945693	18624.56
Soft Drinks	131.804064	49294.72
Health and Hygiene	131.124651	56383.60
Baking Goods	126.096940	67587.96

8. Top 5 categories in each outlet type:

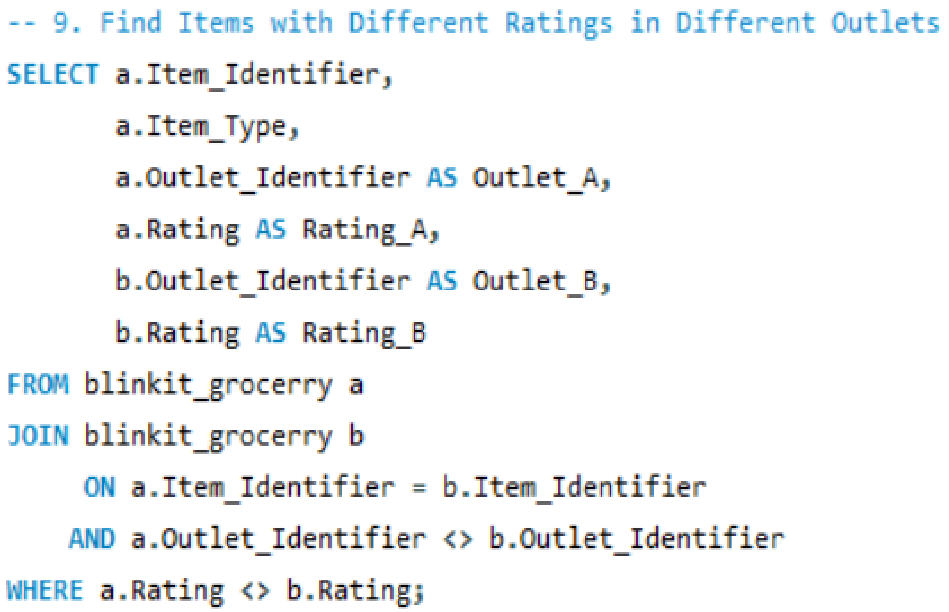


```
-- 8. Top 5 Categories in Each Outlet Type (Window Function)
SELECT *
FROM (
    SELECT Outlet_Type, Item_Type, SUM(Sales) AS Total_Sales,
           RANK() OVER (PARTITION BY Outlet_Type ORDER BY SUM(Sales) DESC) AS RankInOutlet
    FROM blinkit_grocerry
    GROUP BY Outlet_Type, Item_Type
) AS ranked_data
WHERE RankInOutlet <= 5;
```

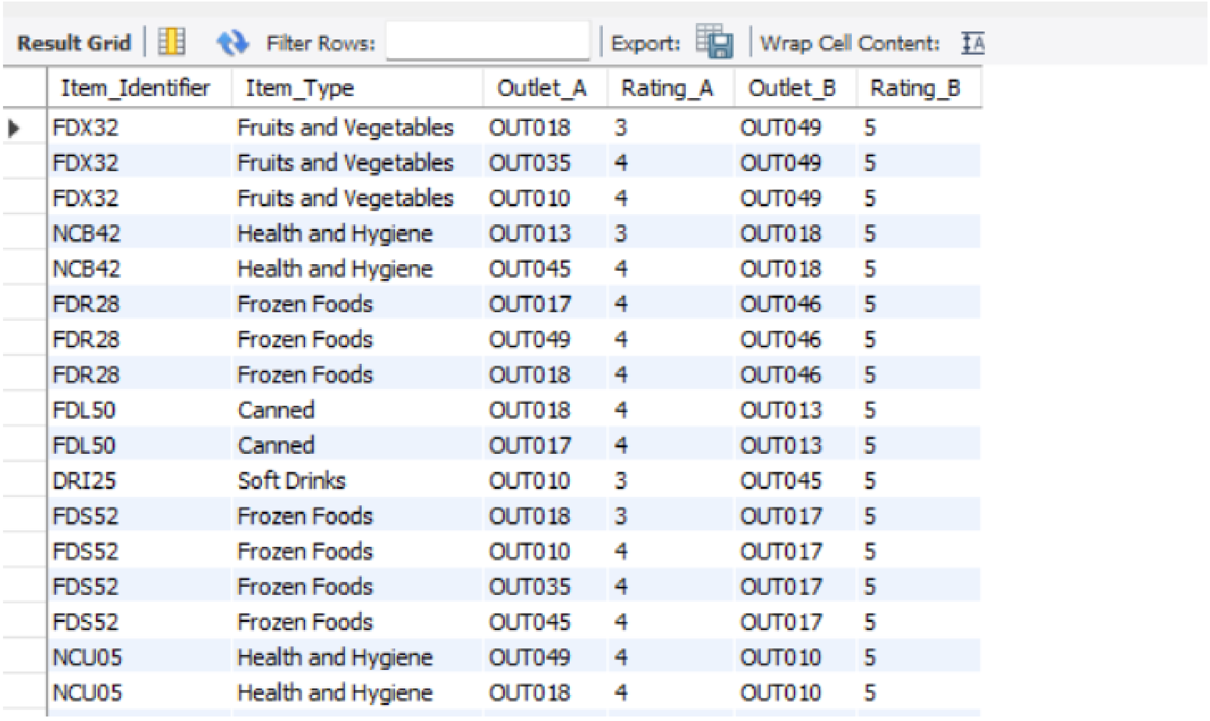


Outlet_Type	Item_Type	Total_Sales	RankInOutlet
Grocery Store	Fruits and Vegetables	10789.68	1
Grocery Store	Snack Foods	10649.79	2
Grocery Store	Household	9822.13	3
Grocery Store	Frozen Foods	7359.85	4
Grocery Store	Dairy	6707.85	5
Supermarket Type1	Fruits and Vegetables	117432.13	1
Supermarket Type1	Snack Foods	114296.30	2
Supermarket Type1	Household	89143.57	3
Supermarket Type1	Frozen Foods	79348.67	4
Supermarket Type1	Dairy	67179.85	5
Supermarket Type2	Snack Foods	20003.26	1
Supermarket Type2	Fruits and Vegetables	18967.33	2
Supermarket Type2	Household	14244.71	3
Supermarket Type2	Frozen Foods	13253.76	4
Supermarket Type2	Canned	10852.46	5

9.Find items with different ratings in different outlets:

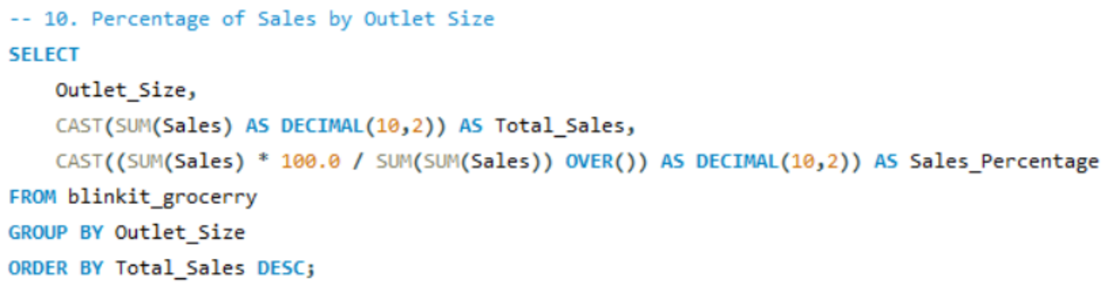


```sql
-- 9. Find Items with Different Ratings in Different Outlets
SELECT a.Item_Identifier,
       a.Item_Type,
       a.Outlet_Identifier AS Outlet_A,
       a.Rating AS Rating_A,
       b.Outlet_Identifier AS Outlet_B,
       b.Rating AS Rating_B
FROM blinkit_grocerry a
JOIN blinkit_grocerry b
     ON a.Item_Identifier = b.Item_Identifier
    AND a.Outlet_Identifier <> b.Outlet_Identifier
WHERE a.Rating <> b.Rating;
```

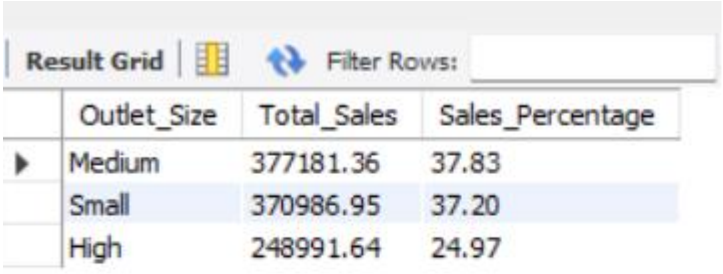


Item_Identifier	Item_Type	Outlet_A	Rating_A	Outlet_B	Rating_B
FDX32	Fruits and Vegetables	OUT018	3	OUT049	5
FDX32	Fruits and Vegetables	OUT035	4	OUT049	5
FDX32	Fruits and Vegetables	OUT010	4	OUT049	5
NCB42	Health and Hygiene	OUT013	3	OUT018	5
NCB42	Health and Hygiene	OUT045	4	OUT018	5
FDR28	Frozen Foods	OUT017	4	OUT046	5
FDR28	Frozen Foods	OUT049	4	OUT046	5
FDR28	Frozen Foods	OUT018	4	OUT046	5
FDL50	Canned	OUT018	4	OUT013	5
FDL50	Canned	OUT017	4	OUT013	5
DRI25	Soft Drinks	OUT010	3	OUT045	5
FDS52	Frozen Foods	OUT018	3	OUT017	5
FDS52	Frozen Foods	OUT010	4	OUT017	5
FDS52	Frozen Foods	OUT035	4	OUT017	5
FDS52	Frozen Foods	OUT045	4	OUT017	5
NCU05	Health and Hygiene	OUT049	4	OUT010	5
NCU05	Health and Hygiene	OUT018	4	OUT010	5

10. Percentage of Sales by outlet size:

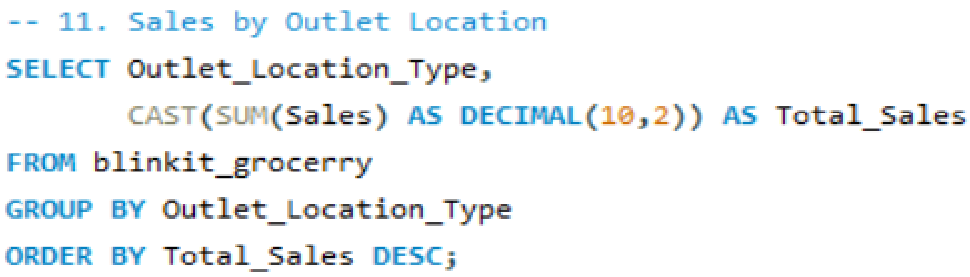


```sql
-- 10. Percentage of Sales by Outlet Size
SELECT
    Outlet_Size,
    CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales,
    CAST((SUM(Sales) * 100.0 / SUM(SUM(Sales)) OVER()) AS DECIMAL(10,2)) AS Sales_Percentage
FROM blinkit_grocerry
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC;
```

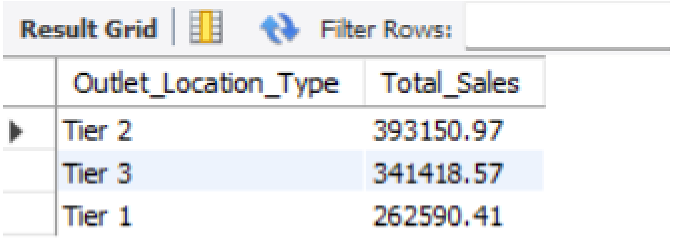


Outlet_Size	Total_Sales	Sales_Percentage
Medium	377181.36	37.83
Small	370986.95	37.20
High	248991.64	24.97

11. Sales by outlet location:

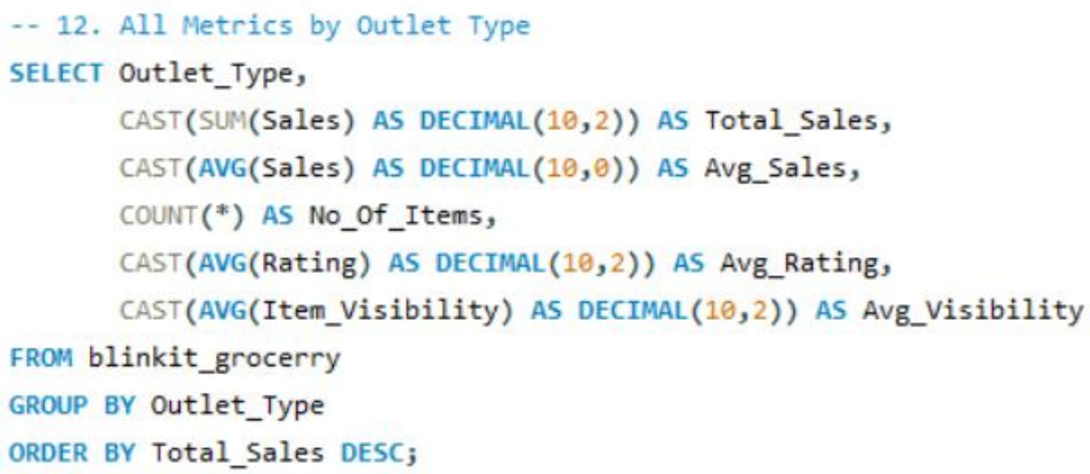


```sql
-- 11. Sales by Outlet Location
SELECT Outlet_Location_Type,
       CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales
FROM blinkit_grocerry
GROUP BY Outlet_Location_Type
ORDER BY Total_Sales DESC;
```

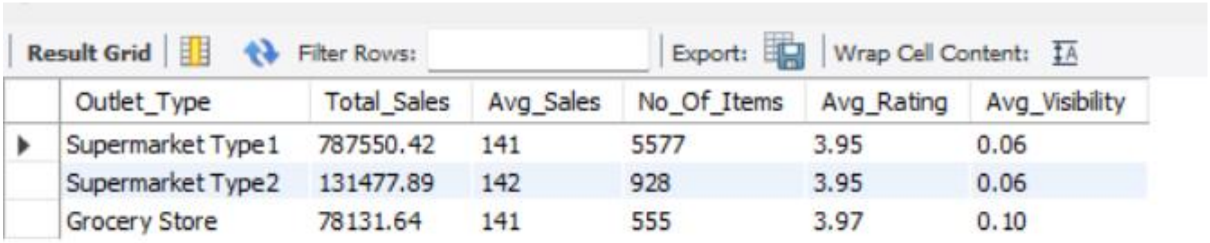


Outlet_Location_Type	Total_Sales
Tier 2	393150.97
Tier 3	341418.57
Tier 1	262590.41

12. All metrices by outlet type:



```
-- 12. 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 Avg_Visibility
FROM blinkit_grocerry
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC;
```



Outlet_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating	Avg_Visibility
Supermarket Type1	787550.42	141	5577	3.95	0.06
Supermarket Type2	131477.89	142	928	3.95	0.06
Grocery Store	78131.64	141	555	3.97	0.10