BLINKIT DATA ANALYSIS

- **See if all the Data is imported:**

```
SELECT * FROM BlinkIT_Grocery_Data
SELECT COUNT(*) FROM BlinkIT_Grocery_Data
```

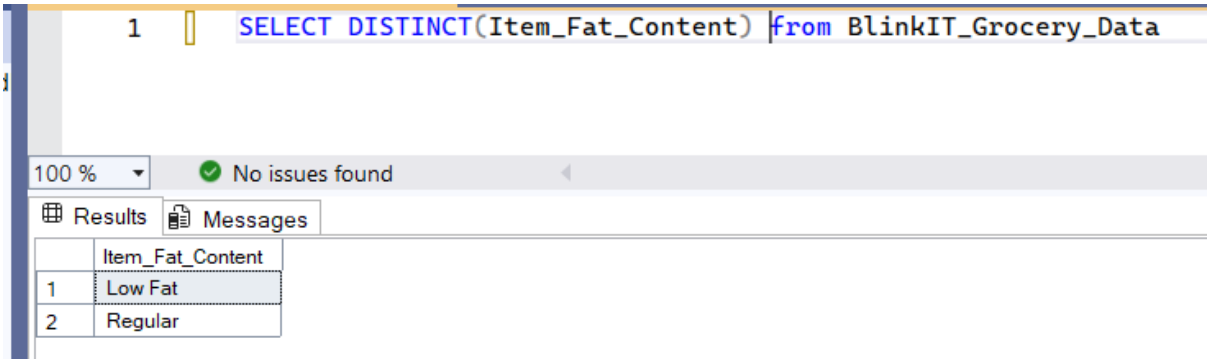
- **DATA CLEANING:**

  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_Grocery_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
```

- **Verify the changes:**

```
SELECT DISTINCT(Item_Fat_Content) from BlinkIT_Grocery_Data
```



A. KPI Requirements:

1. Total Sales:

```
SELECT SUM(Total_Sales) AS Total_Sales from BlinkIT_Grocery_Data
```

```
Total_Sales from BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	Total_Sales
1	1201681.47996712

- To view the Sales in Millions:

```
SELECT CAST(SUM(Total_Sales) /1000000 AS DECIMAL(10,2))
AS Total_Sales_In_Millions from BlinkIT_Grocery_Data
```

```
SELECT CAST(SUM(Total_Sales) /1000000 AS DECIMAL(10,2))
AS Total_Sales_In_Millions from BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	Total_Sales_In_Millions
1	1.20

2. **Average Sales:**

```
SELECT AVG(Total_Sales) AS Avg_Sales FROM BlinkIT_Grocery_Data
```

```
SELECT AVG(Total_Sales) AS Avg_Sales FROM BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	Avg_Sales
1	140.992781880455

- To view in Decimal:

```
SELECT CAST(AVG(Total_Sales) AS DECIMAL(10,2))
AS Avg_Sales FROM BlinkIT_Grocery_Data
```

```sql
SELECT CAST(AVG(Total_Sales) AS DECIMAL(10,2))
AS Avg_Sales FROM BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	Avg_Sales
1	140.99

- To view without Decimal:
  ```sql
  SELECT CAST(AVG(Total_Sales) AS INT)
  AS Avg_Sales FROM BlinkIT_Grocery_Data
  ```

```sql
SELECT CAST(AVG(Total_Sales) AS INT)
AS Avg_Sales FROM BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	Avg_Sales
1	140

3. **Number of Items:**
   ```sql
   SELECT COUNT(*) AS No_of_Items FROM BlinkIT_Grocery_Data
   ```

```sql
SELECT COUNT(*) AS No_of_Items FROM BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	No_of_Items
1	8523

4. **Average Rating:**
   ```sql
   SELECT AVG(Rating) AS Avg_Rating FROM BlinkIT_Grocery_Data
   ```

```sql
SELECT AVG(Rating) AS Avg_Rating FROM BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	Avg_Rating
1	3.96585709104848

- To view in Decimal:
```sql
SELECT CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
from BlinkIT_Grocery_Data
```

```sql
SELECT CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
from BlinkIT_Grocery_Data
```

100 % No issues found

Results | Messages

	Avg_Rating
1	3.97

B. Total Sales by Fat Content:

```sql
SELECT Item_Fat_Content, SUM(Total_Sales) AS Total_Sales
FROM BlinkIT_Grocery_Data
GROUP BY Item_Fat_Content
```

```sql
SELECT Item_Fat_Content, SUM(Total_Sales) AS Total_Sales
FROM BlinkIT_Grocery_Data
GROUP BY Item_Fat_Content
```

100 % No issues found

Results | Messages

	Item_Fat_Content	Total_Sales
1	Low Fat	776319.67764473
2	Regular	425361.802322388

- To view in Decimal:
```sql
SELECT Item_Fat_Content, CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS
Total_Sales
FROM BlinkIT_Grocery_Data
GROUP BY Item_Fat_Content
```

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

100 % No issues found

Results Messages

	Item_Fat_Content	Total_Sales
1	Low Fat	776319.68
2	Regular	425361.80

- All Metrics by Item Fat Content:

```sql
SELECT Item_Fat_Content,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Item_Fat_Content
ORDER BY Total_Sales DESC
```

```sql
SELECT Item_Fat_Content,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Item_Fat_Content
ORDER BY Total_Sales DESC
```

100 % No issues found

Results Messages

	Item_Fat_Content	Total_Sales	Avg_Sales	Avg_Rating	No_of_Items
1	Low Fat	776319.68	140.71	3.97	5517
2	Regular	425361.80	141.50	3.97	3006

C. Total Sales by Item Type:

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

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

100 % No issues found

Results Messages

	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

- Top 5 Sellers by Item Type:

```sql
SELECT TOP 5 Item_Type,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Item_Type
ORDER BY Total_Sales DESC
```

```sql
SELECT TOP 5 Item_Type,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Item_Type
ORDER BY Total_Sales DESC
```

100 % No issues found

Results Messages

	Item_Type	Total_Sales	Avg_Sales	Avg_Rating	No_of_Items
1	Fruits and Vegetables	178124.08	144.58	3.96	1232
2	Snack Foods	175433.92	146.19	3.95	1200
3	Household	135976.53	149.42	4.00	910
4	Frozen Foods	118558.88	138.50	3.97	856
5	Dairy	101276.46	148.50	3.97	682

D. Fat Content by Outlet for Total Sales:

```sql
SELECT Outlet_Location_Type, Item_Fat_Content,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Outlet_Location_Type, Item_Fat_Content
ORDER BY Total_Sales DESC
```

```sql
SELECT Outlet_Location_Type, Item_Fat_Content,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Outlet_Location_Type, Item_Fat_Content
ORDER BY Total_Sales DESC
```

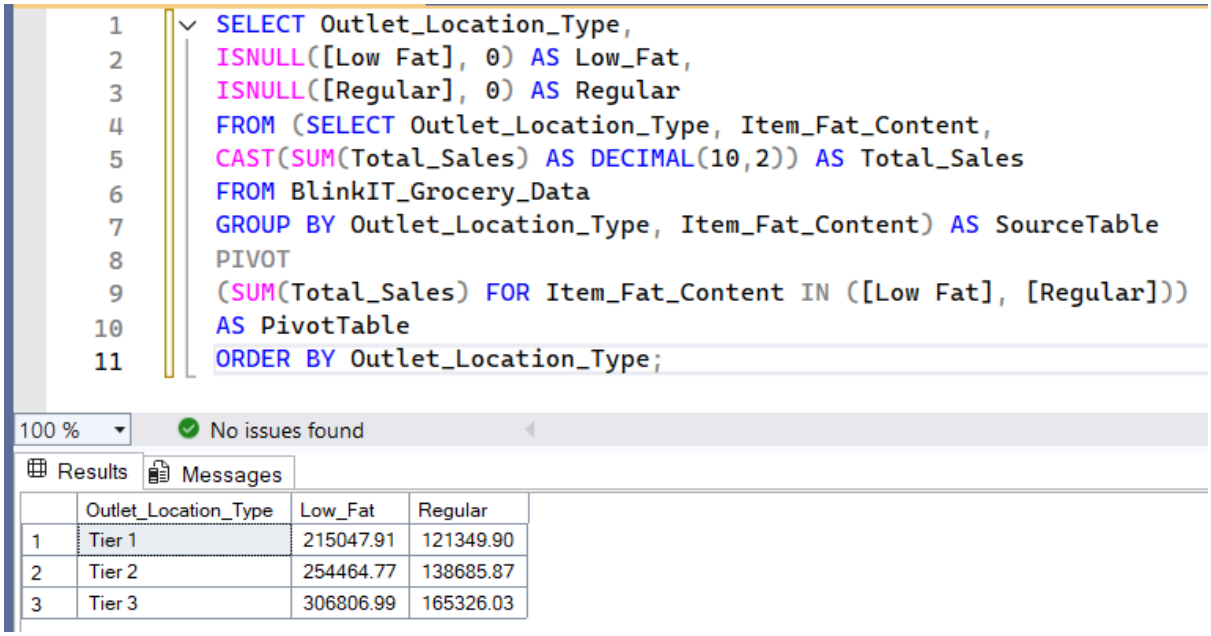
100 % No issues found

Results Messages

	Outlet_Location_Type	Item_Fat_Content	Total_Sales	Avg_Sales	Avg_Rating	No_of_Items
1	Tier 3	Low Fat	306806.99	141.52	3.96	2168
2	Tier 2	Low Fat	254464.77	140.67	3.97	1809
3	Tier 1	Low Fat	215047.91	139.64	3.98	1540
4	Tier 3	Regular	165326.03	139.87	3.97	1182
5	Tier 2	Regular	138685.87	142.10	3.95	976
6	Tier 1	Regular	121349.90	143.10	3.97	848

- Using Pivot Table:

```sql
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_Grocery_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

Query Explanations

This query aims to transform the `blinkit_data` table to display total sales (`Total_Sales`) for each combination of `Outlet_Location_Type` and `Item_Fat_Content`. The result will show `Outlet_Location_Type` as rows and `Item_Fat_Content` categories ("Low Fat" and "Regular") as columns. If there are no sales for a particular combination, the query will display `0` instead of `NULL`.

Detailed Explanation:

1. **Subquery**
   - **Aggregation:**
     sql
     CopyEdit
     ```sql
     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
     ```
     - **Purpose:** This subquery groups the data by `Outlet_Location_Type` and `Item_Fat_Content`, calculating the total sales for each combination.
     - **`CAST(SUM(Total_Sales) AS DECIMAL(10,2))`:** Sums the `Total_Sales` for each group and casts the result to a decimal with two decimal places for precision.

2. **PIVOT Operation:**
   - **Pivoting:**
     ```
     sql
     CopyEdit
     PIVOT
     (
         SUM(Total_Sales)
         FOR Item_Fat_Content IN ([Low Fat], [Regular])
     ) AS PivotTable
     ```
     - **Purpose:** Transforms the rows of `Item_Fat_Content` into columns (`[Low Fat]` and `[Regular]`).
     - **`SUM(Total_Sales)`:** Aggregates the `Total_Sales` for each `Item_Fat_Content` category within each `Outlet_Location_Type`.
3. **Main Query:**
   - **Selecting and Handling NULLs:**
     ```
     sql
     CopyEdit
     SELECT
         Outlet_Location_Type,
         ISNULL([Low Fat], 0) AS Low_Fat,
         ISNULL([Regular], 0) AS Regular
     FROM
         PivotTable
     ORDER BY
         Outlet_Location_Type;
     ```
     - **`ISNULL([Low Fat], 0) AS Low_Fat`:** Replaces any `NULL` values in the `[Low Fat]` column with `0` and renames the column to `Low_Fat`.
     - **`ISNULL([Regular], 0) AS Regular`:** Similarly, replaces `NULL` values in the `[Regular]` column with `0`.
     - **`ORDER BY Outlet_Location_Type`:** Sorts the final result set by `Outlet_Location_Type`.

Why Use `ISNULL`?

When performing a PIVOT operation, if a particular combination of `Outlet_Location_Type` and `Item_Fat_Content` doesn't exist in the data, the resulting cell will contain a `NULL` value. Using `ISNULL(column)`

E. Total Sales by Outlet Establishment:

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

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

100 % No issues found

Results Messages

	Outlet_Establishment_Year	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:

```sql
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 BlinkIT_Grocery_Data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC
```

```sql
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 BlinkIT_Grocery_Data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC
```

100 % No issues found

Results Messages

	Outlet_Size	Total_Sales	Sales_Percentage
1	Medium	507895.73	42.27
2	Small	444794.17	37.01
3	High	248991.58	20.72

Query Explanation:

Outlet_Size: This column represents the size category of the outlet (e.g., Small, Medium, Large).

CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales:

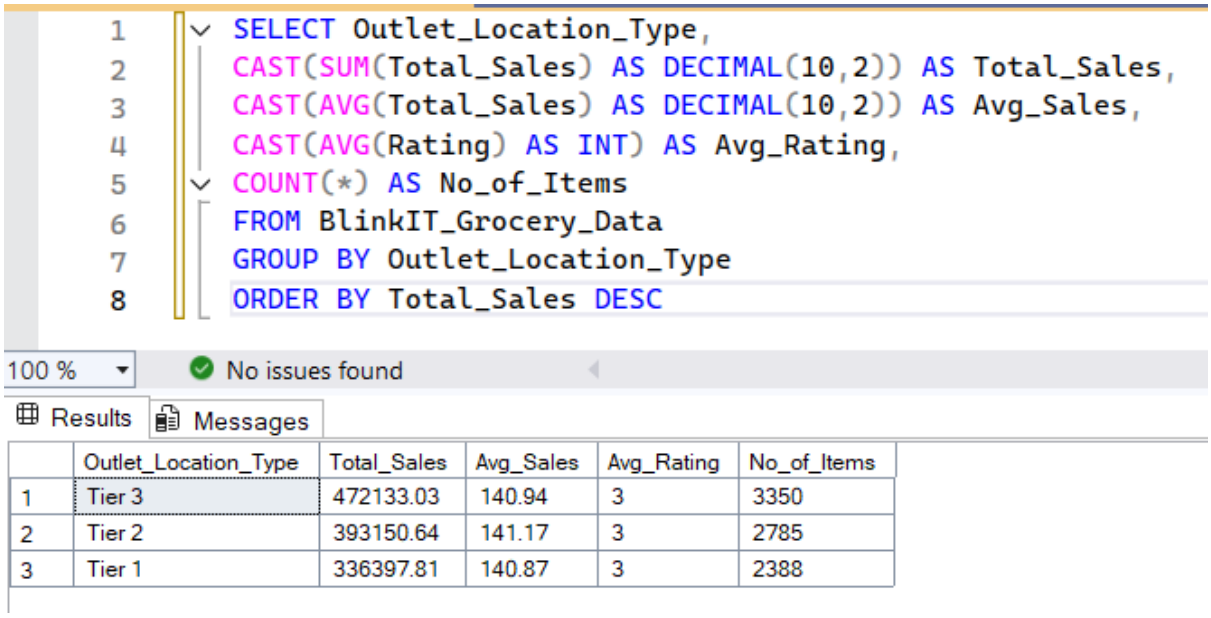
- **SUM(Total_Sales)**: Calculates the total sales for each Outlet_Size.
- **CAST(... AS DECIMAL(10,2))**: Formats the resulting sum to a decimal number with two decimal places for precision.

CAST((SUM(Total_Sales) * 100.0 / SUM(SUM(Total_Sales)) OVER()) AS DECIMAL(10,2)) AS Sales_Percentage:

- **SUM(Total_Sales) * 100.0**: Multiplies the total sales of the current Outlet_Size by 100 to prepare for percentage calculation.
- **SUM(SUM(Total_Sales)) OVER()**:
  - **SUM(Total_Sales)**: Within the GROUP BY context, this computes the total sales for each Outlet_Size.
  - **SUM(...) OVER()**: The outer SUM combined with the OVER() clause calculates the grand total of all Total_Sales across all outlet sizes without collapsing the result set.
- **SUM(Total_Sales) * 100.0 / SUM(SUM(Total_Sales)) OVER()**: Divides the total sales of the current Outlet_Size by the grand total sales and multiplies by 100 to get the percentage contribution of each outlet size to the overall sales.
- **CAST(... AS DECIMAL(10,2))**: Formats the resulting percentage to two decimal places.

G. Sales by Outlet Location:

```sql
SELECT Outlet_Location_Type,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS INT) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Outlet_Location_Type
ORDER BY Total_Sales DESC
```



	Outlet_Location_Type	Total_Sales	Avg_Sales	Avg_Rating	No_of_Items
1	Tier 3	472133.03	140.94	3	3350
2	Tier 2	393150.64	141.17	3	2785
3	Tier 1	336397.81	140.87	3	2388

H. All Metrics by Outlet Type:

```sql
SELECT Outlet_Type,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST((SUM(Total_Sales) * 100/ SUM(SUM(Total_Sales)) OVER()) AS DECIMAL(10,2))
AS Sales_Percentage,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS INT) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
```

```sql
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC
```

```sql
SELECT Outlet_Type,
CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales,
CAST((SUM(Total_Sales) * 100/ SUM(SUM(Total_Sales)) OVER()) AS DECIMAL(10,2)) AS Sales_Percentage,
CAST(AVG(Total_Sales) AS DECIMAL(10,2)) AS Avg_Sales,
CAST(AVG(Rating) AS INT) AS Avg_Rating,
COUNT(*) AS No_of_Items
FROM BlinkIT_Grocery_Data
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC
```

100 % No issues found

Results Messages

	Outlet_Type	Total_Sales	Sales_Percentage	Avg_Sales	Avg_Rating	No_of_Items
1	Supermarket Type1	787549.89	65.54	141.21	3	5577
2	Grocery Store	151939.15	12.64	140.29	3	1083
3	Supermarket Type2	131477.77	10.94	141.68	3	928
4	Supermarket Type3	130714.67	10.88	139.80	3	935