Blinkit Analysis Using SQL

- **See all the data imported:**

  SELECT * FROM blinkit_data;

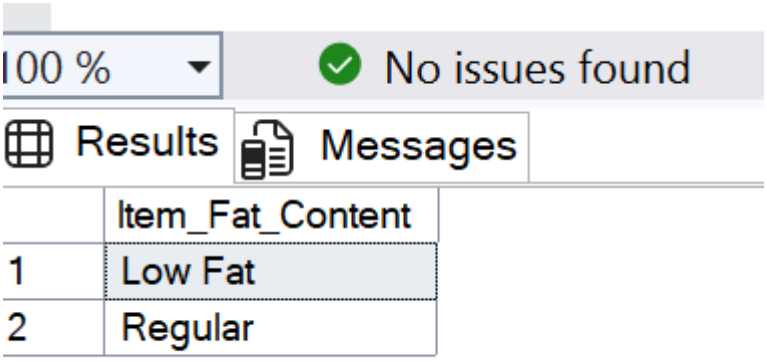
- **DATA CLEANING:**

  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_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 if the data has been cleaned or not using the query:

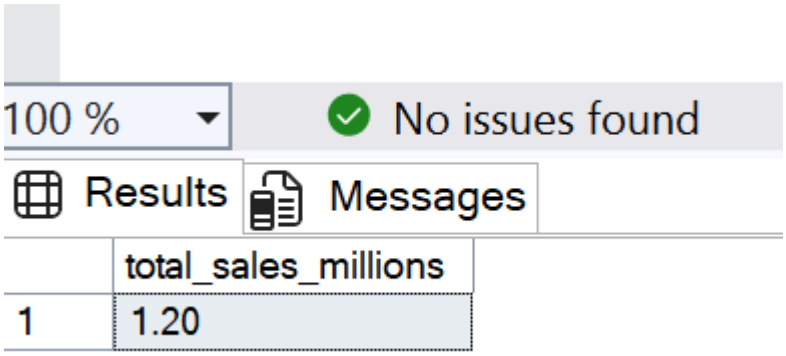
  SELECT DISTINCT Item_Fat_Content FROM blinkit_data;



A.KPI's(ALL METRICS)

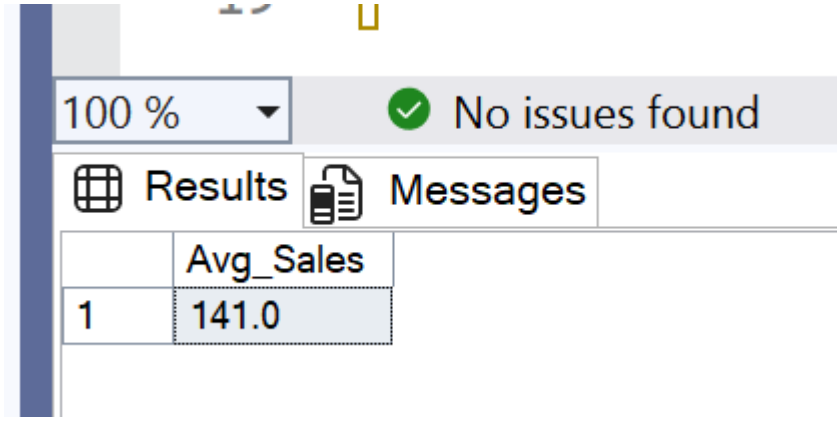
1.TOTAL SALES:

```
SELECT CAST(SUM(Sales)/ 1000000  AS DECIMAL(10,2)) AS total_sales_millions
 FROM blinkit_data;
```



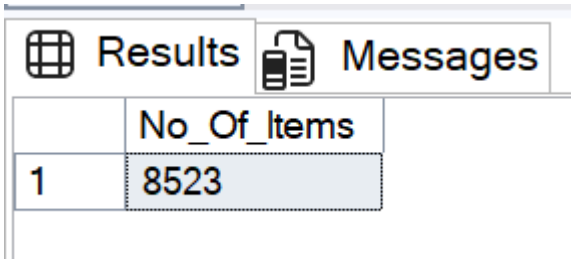
2.AVERAGE SALES:

```
SELECT CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales FROM blinkit_data;
```



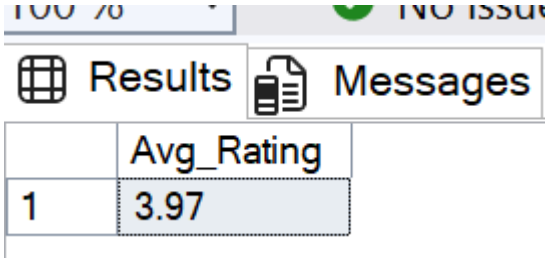
3.NO. OF ITEMS:

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



4.AVERAGE RATING:

SELECT CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating FROM blinkit_data;


Results Messages

	Avg_Rating
1	3.97

B.Granular Requirements:

1.ALL METRICS BY FAT CONTENT:

```
SELECT Item_Fat_Content,
        CAST(SUM(Sales)/1000 AS DECIMAL(10,2)) AS Total_Sales_Thousands,
        CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales,
        COUNT(*) AS No_Of_Items,
        CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
From blinkit_data
GROUP BY Item_Fat_Content
ORDER BY Total_Sales_Thousands DESC;
```

Results Messages

	Item_Fat_Content	Total_Sales_Thousands	Avg_Sales	No_Of_Items	Avg_Rating
1	Low Fat	776.32	140.7	5517	3.97
2	Regular	425.36	141.5	3006	3.97

2.ALL METRICS BY ITEM TYPE:

```
SELECT Item_Type,
        CAST(SUM(Sales)/1000 AS DECIMAL(10,2)) AS Total_Sales,
        CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales,
        COUNT(*) AS No_Of_Items,
```

```
        CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
From blinkit_data
GROUP BY Item_Type
ORDER BY Total_Sales DESC;
```

Results Messages

	Item_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Fruits and Vegetables	178.12	144.6	1232	3.96
2	Snack Foods	175.43	146.2	1200	3.95
3	Household	135.98	149.4	910	4.00
4	Frozen Foods	118.56	138.5	856	3.97
5	Dairy	101.28	148.5	682	3.97
6	Canned	90.71	139.8	649	3.99
7	Baking Goods	81.89	126.4	648	3.98
8	Health and Hygiene	68.03	130.8	520	3.99
9	Meat	59.45	139.9	425	4.02
10	Soft Drinks	58.51	131.5	445	3.92
11	Breads	35.38	141.0	251	3.88
12	Hard Drinks	29.33	137.1	214	3.91
13	Others	22.45	132.9	169	3.95
14	Starchy Foods	21.88	147.8	148	3.92
15	Breakfast	15.60	141.8	110	3.93
16	Seafood	9.08	141.8	64	3.96

--To find top 5 items

```
SELECT TOP 5 Item_Type,
        CAST(SUM(Sales)/1000 AS DECIMAL(10,2)) AS Total_Sales,
        CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales,
        COUNT(*) AS No_Of_Items,
        CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
From blinkit_data
GROUP BY Item_Type
ORDER BY Total_Sales DESC;
```

Results Messages

	Item_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Fruits and Vegetables	178.12	144.6	1232	3.96
2	Snack Foods	175.43	146.2	1200	3.95
3	Household	135.98	149.4	910	4.00
4	Frozen Foods	118.56	138.5	856	3.97
5	Dairy	101.28	148.5	682	3.97

3.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(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;
```

Results Messages

	Outlet_Location_Type	Low_Fat	Regular
1	Tier 1	215047.91	121349.90
2	Tier 2	254464.78	138685.87
3	Tier 3	306807.00	165326.04

4.ALL METRICS BY OUTLET ESTABLISHMENT:

```
SELECT Outlet_Establishment_Year,
    CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales,
    CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales,
    COUNT(*) AS No_Of_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
FROM blinkit_data
GROUP BY Outlet_Establishment_Year
ORDER BY Total_Sales DESC;
```

Results Messages

	Outlet_Establishment_Year	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	2018	204522.26	139.8	1463	3.97
2	2017	133103.91	143.1	930	3.94
3	2016	132113.37	142.1	930	3.96
4	2014	131809.02	141.4	932	3.95
5	2022	131477.78	141.7	928	3.97
6	2015	130942.78	141.0	929	3.96
7	2012	130476.86	140.3	930	3.99
8	2020	129103.96	139.4	926	3.98
9	2011	78131.57	140.8	555	3.98

5.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_data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC;
```

Results Messages

	Outlet_Size	Total_Sales	Sales_Percentage
1	Medium	507895.74	42.27
2	Small	444794.17	37.01
3	High	248991.59	20.72

6.ALL METRICS BY OUTLET LOCATION:

```
SELECT Outlet_Location_Type,
    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,
    CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales,
    COUNT(*) AS No_Of_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
FROM blinkit_data
WHERE Outlet_Establishment_Year = 2020
GROUP BY Outlet_Location_Type
ORDER BY Total_Sales DESC;
```

Results Messages

	Outlet_Location_Type	Total_Sales	Sales_Percentage	Avg_Sales	No_Of_Items	Avg_Rating
1	Tier 3	472133.03	39.29	140.9	3350	3.96
2	Tier 2	393150.65	32.72	141.2	2785	3.96
3	Tier 1	336397.81	27.99	140.9	2388	3.98

7.ALL METRICS BY OUTLET TYPE:

```
SELECT Outlet_Type,
    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,
    CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales,
    COUNT(*) AS No_Of_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating
FROM blinkit_data
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC;
```

Results Messages

	Outlet_Type	Total_Sales	Sales_Percentage	Avg_Sales	No_Of_Items	Avg_Rating
1	Supermarket Type1	787549.89	65.54	141.2	5577	3.96
2	Grocery Store	151939.15	12.64	140.3	1083	3.99
3	Supermarket Type2	131477.78	10.94	141.7	928	3.97
4	Supermarket Type3	130714.67	10.88	139.8	935	3.95