BlinkIT Analysis

- See all the imported Data:

```
SELECT *
FROM blinkit_data
```

	Item_Fat_Content	Item_Identifier	Item_Type	Outlet_Establishment_Year	Outlet_Identifier	Outlet_Location_Type	Outlet_Size	Outlet_Type	Item_Visibility	Item_Weight	Sales	Rating
1	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100013501942158	15.1000003814697	145.478607177734	5
2	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.00859605055302382	11.8000001907349	115.349197387695	5
3	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Supermarket Type1	0.0258964858949184	13.8500003814697	165.02099609375	5
4	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Supermarket Type1	0.0422778688371181	12.1499996185303	126.504600524902	5
5	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970195800066	19.6000003814697	55.1613998413086	5
6	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.00550548080354929	8.89000034332275	102.40160369873	5
7	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.0983124226331711	11.8000001907349	81.461799621582	5
8	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.0269037131220102	19.7000007629395	96.0726013183594	5
9	Low Fat	FDW20	Fruits and Vegetables	2014	OUT013	Tier 3	High	Supermarket Type1	0.024129331111908	20.75	124.172996520996	5
10	Low Fat	FDX25	Canned	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.101561568677425	NULL	181.92919921875	5
11	LF	FDX21	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.0845545679330826	NULL	109.89119720459	5
12	Low Fat	NCU41	Health and Hygiene	2017	OUT035	Tier 2	Small	Supermarket Type1	0.0520449765026569	18.8500003814697	192.184600830078	5
13	Low Fat	FDL20	Fruits and Vegetables	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.128937661647797	17.1000003814697	112.388603210449	5
14	Low Fat	NCR54	Household	2014	OUT013	Tier 3	High	Supermarket Type1	0.0904868245124817	16.3500003814697	195.210998535156	5
15	Low Fat	FDH19	Meat	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.032928239554166 8	NULL	173.173797607422	5
16	Regular	FDB57	Fruits and Vegetables	2017	OUT035	Tier 2	Small	Supermarket Type1	0.018801549449563	20.25	222.177200317383	5
17	Low Fat	FDO23	Breads	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.14702382683754	17.8500003814697	93.7435989379883	5
18	Low Fat	NCB07	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.0776280537247658	19.2000007629395	197.610992431641	5
19	Low Fat	FDJ56	Fruits and Vegetables	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.182514876127243	NULL	98.7699966430664	5

```
SELECT COUNT(*)
FROM blinkit_data
```

	(No column name)
1	8523

- Data Cleaning:
  Cleaning the Item_Fat_Content field ensures data consistency and accuracy in analysis. The presence of multiple versions 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 dataset.

```
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
```

```
(8523 rows affected)

Completion time: 2025-07-12T12:44:54.7683805+05:30
```

- After executing this query, check whether the data has been cleaned or not.

```
SELECT *
FROM blinkit_data
```

	Item_Fat_Content	Item_Identifier	Item_Type	Outlet_Establishment_Year	Outlet_Identifier	Outlet_Location_Type	Outlet_Size	Outlet_Type	Item_Visibility	Item_Weight	Sales	Rating
1	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100013501942158	15.1000003814697	145.478607177734	5
2	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.00859605055302382	11.8000001907349	115.349197387695	5
3	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Supermarket Type1	0.0258964858949184	13.8500003814697	165.02099609375	5
4	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Supermarket Type1	0.0422778688371181	12.1499996185303	126.504600524902	5
5	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970195800066	19.6000003814697	55.1613998413086	5
6	Low Fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.00550548080354929	8.89000034332275	102.40160369873	5
7	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.0983124226331711	11.8000001907349	81.461799621582	5
8	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.0269037131220102	19.7000007629395	96.0726013183594	5
9	Low Fat	FDW20	Fruits and Vegetables	2014	OUT013	Tier 3	High	Supermarket Type1	0.024129331111908	20.75	124.172996520996	5
10	Low Fat	FDX25	Canned	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.101561568677425	NULL	181.92919921875	5
11	Low Fat	FDX21	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.0845545679330826	NULL	109.89119720459	5
12	Low Fat	NCU41	Health and Hygiene	2017	OUT035	Tier 2	Small	Supermarket Type1	0.0520449765026569	18.8500003814697	192.184600830078	5
13	Low Fat	FDL20	Fruits and Vegetables	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.128937661647797	17.1000003814697	112.388603210449	5

```
SELECT DISTINCT(Item_Fat_Content)
FROM blinkit_data;
```

	Item_Fat_Content
1	Low Fat
2	Regular

KPI's

1. Total Sales:

```
SELECT SUM(Sales) AS Total_Sales
FROM blinkit_data
```

	Total_Sales
1	1201681.49196053

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

	Total_Sales_Millions
1	1.20

2. Average Sales

```
SELECT AVG(Sales) AS Avg_Sales
FROM blinkit_data
```

	Avg_Sales
1	140.992783287636

```
SELECT CAST(AVG(Sales) AS DECIMAL(10, 0)) AS Avg_Sales
FROM blinkit_data
```

	Avg_Sales
1	141

3. Number of Items

```
SELECT COUNT(*) FROM blinkit_data
```

	(No column name)
1	8523

4. Average Rating

```
SELECT AVG(Rating) AS Avg_Rating
FROM blinkit_data
```

	Avg_Rating
1	3.96585709104848

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

	Avg_Rating
1	3.97

Granular Requirements

1. Total Sales by Fat Content

```
SELECT Item_Fat_Content, SUM(Sales) AS Total_Sales
FROM blinkit_data
GROUP BY Item_Fat_Content
```

	Item_Fat_Content	Total_Sales
1	Low Fat	776319.687639236
2	Regular	425361.804321289

```
SELECT Item_Fat_Content, CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total_Sales
FROM blinkit_data
GROUP BY Item_Fat_Content
ORDER BY Total_Sales DESC
```

	Item_Fat_Content	Total_Sales
1	Low Fat	776319.69
2	Regular	425361.80

```
SELECT Item_Fat_Content,
    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
WHERE Outlet_Establishment_Year = 2022
GROUP BY Item_Fat_Content
ORDER BY Total_Sales DESC
```

	Item_Fat_Content	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Low Fat	84844.61	141.9	598	3.97
2	Regular	46633.17	141.3	330	3.98

```
SELECT Item_Fat_Content,
    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 Item_Fat_Content
ORDER BY Total_Sales DESC
```

	Item_Fat_Content	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Low Fat	776319.69	140.7	5517	3.97
2	Regular	425361.80	141.5	3006	3.97

```sql
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
```

	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. Total Sales by Item Type

```sql
SELECT Item_Type,
    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 Item_Type
ORDER BY Total_Sales DESC
```

Results Messages

	Item_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Fruits and Vegetables	178124.08	144.6	1232	3.96
2	Snack Foods	175433.92	146.2	1200	3.95
3	Household	135976.53	149.4	910	4.00
4	Frozen Foods	118558.88	138.5	856	3.97
5	Dairy	101276.46	148.5	682	3.97
6	Canned	90706.73	139.8	649	3.99
7	Baking Goods	81894.74	126.4	648	3.98
8	Health and Hygiene	68025.84	130.8	520	3.99
9	Meat	59449.86	139.9	425	4.02
10	Soft Drinks	58514.17	131.5	445	3.92
11	Breads	35379.12	141.0	251	3.88
12	Hard Drinks	29334.68	137.1	214	3.91
13	Others	22451.89	132.9	169	3.95
14	Starchy Foods	21880.03	147.8	148	3.92
15	Breakfast	15596.70	141.8	110	3.93
16	Seafood	9077.87	141.8	64	3.96

```sql
SELECT TOP 5 Item_Type,
    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 Item_Type
ORDER BY Total_Sales DESC
```

	Item_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Fruits and Vegetables	178124.08	144.6	1232	3.96
2	Snack Foods	175433.92	146.2	1200	3.95
3	Household	135976.53	149.4	910	4.00
4	Frozen Foods	118558.88	138.5	856	3.97
5	Dairy	101276.46	148.5	682	3.97

```sql
SELECT TOP 5 Item_Type,
    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 Item_Type
ORDER BY Total_Sales
```

	Item_Type	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Seafood	9077.87	141.8	64	3.96
2	Breakfast	15596.70	141.8	110	3.93
3	Starchy Foods	21880.03	147.8	148	3.92
4	Others	22451.89	132.9	169	3.95
5	Hard Drinks	29334.68	137.1	214	3.91

3. Fat Content by Outlet for Total Sales

```
SELECT Outlet_Location_Type, Item_Fat_Content,
    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_Location_Type, Item_Fat_Content
ORDER BY Total_Sales DESC
```

	Outlet_Location_Type	Item_Fat_Content	Total_Sales	Avg_Sales	No_Of_Items	Avg_Rating
1	Tier 3	Low Fat	306807.00	141.5	2168	3.96
2	Tier 2	Low Fat	254464.78	140.7	1809	3.97
3	Tier 1	Low Fat	215047.91	139.6	1540	3.98
4	Tier 3	Regular	165326.04	139.9	1182	3.97
5	Tier 2	Regular	138685.87	142.1	976	3.95
6	Tier 1	Regular	121349.90	143.1	848	3.97

```
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
ORDER BY Total_Sales ASC
```

	Outlet_Location_Type	Item_Fat_Content	Total_Sales
1	Tier 1	Regular	121349.90
2	Tier 2	Regular	138685.87
3	Tier 3	Regular	165326.04
4	Tier 1	Low Fat	215047.91
5	Tier 2	Low Fat	254464.78
6	Tier 3	Low Fat	306807.00

```
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;
```

	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. Total Sales by Outlet Establishment

```
SELECT Outlet_Establishment_Year,
    CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total_Sales
FROM blinkit_data
GROUP BY Outlet_Establishment_Year
ORDER BY Outlet_Establishment_Year ASC
```

	Outlet_Establishment_Year	Total_Sales
1	2011	78131.57
2	2012	130476.86
3	2014	131809.02
4	2015	130942.78
5	2016	132113.37
6	2017	133103.91
7	2018	204522.26
8	2020	129103.96
9	2022	131477.78

```
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
```

	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 / SUM(SUM(Sales)) OVER()) AS DECIMAL(10, 2)) AS Sales_Percentage
FROM blinkit_data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC
```

	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(Sales) AS DECIMAL(10, 2)) AS Total Sales:

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

CAST((SUM(Sales) * 100 / SUM(SUM(Sales)) OVER()) AS DECIMAL(10, 2)) AS Sales_Percentage:

- **SUM(Sales) * 100**: Multiplies the total sales of the current `Outlet_Size` by 100 to prepare for percentage calculation.
- **SUM(SUM(Sales)) OVER()**:
  - **SUM(Sales)**: Within the `GROUP BY` context, this computes the total sales for each `Outlet_Size`.
  - **SUM(SUM(Sales)) OVER()**: The outer `SUM` combined with the `OVER()` calculates the grand total of all the Sales across all `Outlet_Size` without collapsing the result set.
- **(SUM(Sales) * 100 / SUM(SUM(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.

6. Sales by Outlet Location:

```
SELECT Outlet_Location_Type,
    CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total_Sales,
    CAST((SUM(Sales) * 100 / 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_Location_Type
ORDER BY Total_Sales DESC
```

	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

```
SELECT Outlet_Location_Type,
    CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total_Sales,
    CAST((SUM(Sales) * 100 / 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 = 2022
GROUP BY Outlet_Location_Type
ORDER BY Total_Sales DESC
```

	Outlet_Location_Type	Total_Sales	Sales_Percentage	Avg_Sales	No_Of_Items	Avg_Rating
1	Tier 3	131477.78	100.00	141.7	928	3.97

7. All Metrics by Outlet Type:

```
SELECT Outlet_Type,
    CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total_Sales,
    CAST((SUM(Sales) * 100 / 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
```

	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

```
SELECT Outlet_Type,
    CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total_Sales,
    CAST((SUM(Sales) * 100 / 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 = 2022
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC
```

	Outlet_Type	Total_Sales	Sales_Percentage	Avg_Sales	No_Of_Items	Avg_Rating
1	Supermarket Type2	131477.78	100.00	141.7	928	3.97