Blinkit Analysis SQL Codes

Thilina Sasanka
UOP

Import Data

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

Data Cleaning

```
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 the data has been cleaned or not using below query

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

	Item_Fat_Content
1	Low Fat
2	Regular

KPI'S REQUIREMENTS

Total Sales

Create column as total sales in millions and get values in two decimal points

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

	Total_Sales_Millions
1	1.20

Average Sales

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

	Avg_Sales
1	141.0

No. of Items

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

	No_Of_Items
1	8523

Multiple Filter Conditions

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

```
FROM blinkit_data
WHERE Item_Fat_Content = 'Low Fat'
```

	Total_Sales_Millions
1	0.78

```
SELECT CAST(SUM(Sales)/ 1000000 AS DECIMAL(10,2)) AS Total_Sales_Millions
FROM blinkit_data
WHERE Outlet_Establishment_Year = 2022
```

	Total_Sales_Millions
1	0.13

```
SELECT CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg_Sales FROM blinkit_data
WHERE Outlet_Establishment_Year = 2022
```

	Avg_Sales
1	141.7

```
SELECT COUNT(*)AS No_Of_Items FROM blinkit_data
WHERE Outlet_Establishment_Year = 2022
```

	No_Of_Items
1	928

Average Rating

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

	Avg_Rating
1	3.97

Business Requirement

1.Total Sales 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
```

	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

```
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  (If you want to find bottom 5,  change DESC to ASC)
```

	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

3.Total 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 Sales
              FROM blinkit_data
              GROUP BY Outlet_Location_Type, Item_Fat_Content
) AS SourceTable
PIVOT
(
       SUM(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 Content 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 Outlet_Establishment_Year DESC
```

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

	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.Sales of 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
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

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
```

	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