

# GROCERY ANALYSIS

## *Blinkit Mart Grocery*

---

- a. See all grocery\_data in Blinkit Database

```
SELECT * FROM grocery_data
```

- b. Data Cleaning

Cleaning data pada kolom atau fitur Item\_Fat\_Content untuk konsistensi dan akurasi dalam analisis. Pada fitur tersebut terdapat multi variable pada kategori (LF, Low Fat, == Low Fat).

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

Execute :

```
SELECT DISTINCT Item_Fat_Content FROM grocery_data
```

	Item_Fat_Content
1	Low Fat
2	Regular

## 1. KPI's

- a. TOTAL SALES (The Overall revenue generated from all items sold)

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

Output :

	Total_Sales_Millions
1	0.98

- b. AVERAGE SALES (The Average revenue per sale)

```
SELECT CAST(AVG(Sales) AS DECIMAL(5,2)) AS Avg_Sales  
FROM grocery_data
```

Output :

	Avg_Sales
1	115.50

- c. NO OF ITEMS (The total count of different items sold)

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

Output :

	(No column name)
1	8523

- d. AVERAGE RATING (The Average Customer rating for items sold)

```
SELECT CAST(AVG(Rating) AS DECIMAL(3,2)) AS Avg_Rating  
FROM grocery_data
```

Output :

	Avg_Rating
1	3.97

## 2. GRANULAR REQUIREMENTS

- a. Total Sales by Fat Content

- Objective : Analyze impact of fat content on total sales
- additional KPI Mtrics : Average sales, Number of Items, Average Rating

```
SELECT Item_Fat_Content,  
CAST(SUM(Sales)/1000 AS DECIMAL(10,2)) AS
```

```

Total_Sales_Thousand,
        CAST(AVG(Sales) AS DECIMAL(10,2)) AS
Avarage_Sales,
        COUNT(*) AS Number_Of_Items,
        CAST(AVG(Rating) AS DECIMAL(10,2)) AS
Avarage_Rating
FROM grocery_data
GROUP BY Item_Fat_Content
ORDER BY Total_Sales_Thousand DESC

```

Output :

	Item_Fat_Content	Total_Sales_Thousand	Avarage_Sales	Number_Of_Items	Avarage_Rating
1	Low Fat	631.16	114.40	5517	3.97
2	Regular	353.27	117.52	3006	3.97

b. Total Sales by Item Type

- i. Objective : Identify the performance of diff item types in terms of total sales
- ii. additional KPI Mtrics : Average sales, Number of Items, Average Rating

```

SELECT Item_Type,
        CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales,
        CAST(AVG(Sales) AS DECIMAL(10,2)) AS
Avarage_Sales,
        COUNT(*) AS Number_Of_Items,
        CAST(AVG(Rating) AS DECIMAL(10,2)) AS
Avarage_Rating
FROM grocery_data
GROUP BY Item_Type
ORDER BY Total_Sales DESC

```

Output :

	Item_Type	Total_Sales	Avarage_Sales	Number_Of_Items	Avarage_Rating
1	Fruits and Vegetables	151647.82	123.09	1232	3.96
2	Snack Foods	143738.49	119.78	1200	3.95
3	Household	109389.42	120.21	910	4.00
4	Frozen Foods	97995.77	114.48	856	3.97
5	Dairy	83455.01	122.37	682	3.97
6	Canned	73791.59	113.70	649	3.99
7	Baking Goods	63979.65	98.73	648	3.98
8	Health and Hygiene	56491.55	108.64	520	3.99
9	Meat	49040.47	115.39	425	4.02
10	Soft Drinks	48625.19	109.27	445	3.92
11	Breads	31437.62	125.25	251	3.88
12	Hard Drinks	24869.66	116.21	214	3.91
13	Starchy Foods	17719.28	119.72	148	3.92
14	Others	13471.04	79.71	169	3.95
15	Breakfast	11976.89	108.88	110	3.93
16	Seafood	6801.59	106.27	64	3.96

c. Fat Content by Outlet for Total Sales

- i. Objective : Compare total sales across diff outlets segmented by fat content
- ii. additional KPI Mtrics : Average sales, Number of Items, Average Rating

```

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 grocery_data
    GROUP BY Outlet_Location_Type, Item_Fat_Content
) AS Source_Table
PIVOT(
    SUM(Total_Sales)
    FOR Item_Fat_Content IN ([Low Fat],[Regular])
) AS Pivot_Table
ORDER BY Outlet_Location_Type

```

Output :

	Outlet_Location_Type	Low_Fat	Regular
1	Tier 1	173162.46	101739.19
2	Tier 2	208357.28	114639.58
3	Tier 3	249640.54	136891.98

d. Total Sales by Outlet Establishment

- i. Objective : Evaluate how the age type of outlet establishment influencers total sales

```
SELECT Outlet_Establishment_Year,
       CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales,
       CAST(AVG(Sales) AS DECIMAL(10,2)) AS
Avarage_Sales,
       COUNT(*) AS Number_Of_Items,
       CAST(AVG(Rating) AS DECIMAL(10,2)) AS
Avarage_Rating
FROM grocery_data
GROUP BY Outlet_Establishment_Year
ORDER BY Outlet_Establishment_Year DESC
```

Output :

	Outlet_Establishment_Year	Total_Sales	Avarage_Sales	Number_Of_Items	Avarage_Rating
1	2022	105831.62	114.04	928	3.97
2	2020	107413.97	116.00	926	3.98
3	2018	165505.93	113.13	1463	3.97
4	2017	110023.83	118.31	930	3.94
5	2016	109274.34	117.50	930	3.96
6	2015	105559.07	113.63	929	3.96
7	2014	108424.65	116.34	932	3.95
8	2012	107467.53	115.56	930	3.99
9	2011	64930.11	116.99	555	3.98

### 3. CHART'S REQUIREMENTS

#### a. Percentage of Sales by Outlet Size

- i. Objective : Analyze the correlation between outlet size and total sales

```
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 grocery_data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC
```

Output :

	Outlet_Size	Total_Sales	Sales_Percentage
1	Medium	414339.18	42.09
2	Small	364346.73	37.01
3	High	205745.13	20.90

#### b. Sales by Outlet Location

- i. Objective : Assess the geographic distribution of sales across diff locations

```
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(10,2)) AS
Average_Sales,
    COUNT(*) AS Number_Of_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS
Avarage_Rating
FROM grocery_data
GROUP BY Outlet_Location_Type
```

Output :

	Outlet_Location_Type	Total_Sales	Sales_Percentage	Average_Sales	Number_Of_Items	Avarage_Rating
1	Tier 2	322996.86	32.81	115.98	2785	3.96
2	Tier 3	386532.52	39.26	115.38	3350	3.96
3	Tier 1	274901.66	27.92	115.12	2388	3.98

#### c. All Metrics by Outlet Type

- i. Objective : Provide a comprehensive view Metrics
1. Total Sales
  2. Average Sales
  3. Number of Items
  4. Average Rating

```

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(10,2)) AS
Average_Sales,
    COUNT(*) AS Number_Of_Items,
    CAST(AVG(Rating) AS DECIMAL(10,2)) AS
Avarage_Rating
FROM grocery_data
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC

```

Output :

	Outlet_Type	Total_Sales	Sales_Percentage	Average_Sales	Number_Of_Items	Avarage_Rating
1	Supermarket Type1	648163.38	65.84	116.22	5577	3.96
2	Grocery Store	123089.90	12.50	113.66	1083	3.99
3	Supermarket Type3	107346.13	10.90	114.81	935	3.95
4	Supermarket Type2	105831.62	10.75	114.04	928	3.97