# BlinkIT Analysis

## Objective:

- Analyse BlinkIT's sales performance using SQL.
- Identify key insights and trends from the data.
- Evaluate customer preferences using key KPIs.
- Understand how product and outlet features affect sales.
- Provide data-driven recommendations to improve business performance.

Tools Used: **Excel, MySQL.**

## Step 1: Importing Data from Excel to MySQL

The BlinkIT Excel file is first saved as a CSV file and then moved into MySQL's upload folder. Using the **LOAD DATA INFILE** command, the CSV file is imported into a table in MySQL. This prepares the data for cleaning and analysis.

## Step 2: Data Cleaning

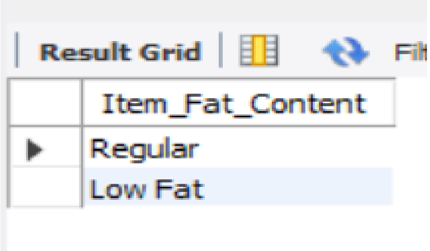
I cleaned the Item_Fat_Content column to make the values consistent. The dataset had different versions of the same category ( LF, low fat  Vs.  Low Fat), which could create confusion during analysis. By standardizing these values, the data becomes more accurate, easier to analyse, and better for creating clear reports.

QUERY:

```
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 |
|---|
| Regular |
| Low Fat |

## Step 3: Business Requirements

### A. KPI's Requirement:

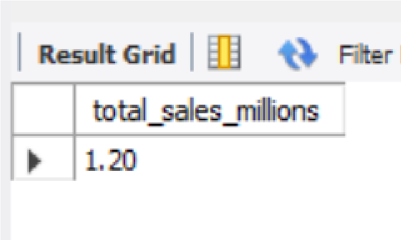
To evaluate BlinkIT's performance effectively, the following key metrics will be analyzed:

### 1. Total Sales:

Overall revenue generated from all products sold.

QUERY:

```
SELECT CAST(SUM(Sales) / 1000000.0 AS DECIMAL(10,2)) AS
Total_Sales_Million
FROM blinkit_data;
```



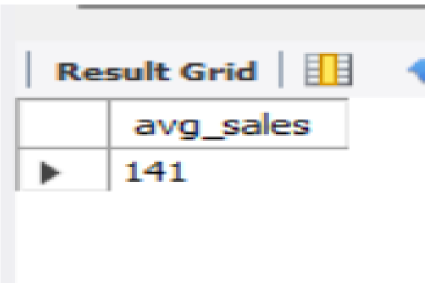
| total_sales_millions |
|---|
| 1.20 |

### 2. Average Sales:

The average revenue per sale or transaction.

QUERY:

```
SELECT CAST(AVG(Total_Sales) AS INT) AS Avg_Sales
FROM blinkit_data;
```



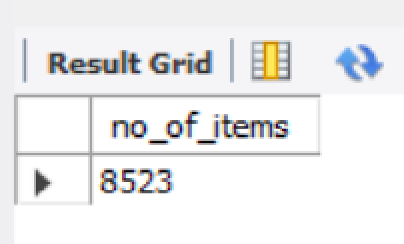
| avg_sales |
|---|
| 141 |

### 3. Number of Items:

Total count of items or SKUs included in sales data.

QUERY:

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



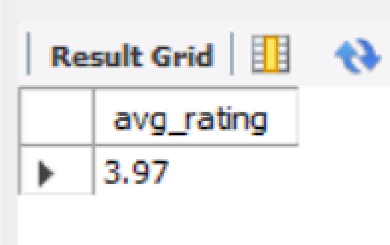
| no_of_items |
|---|
| 8523 |

### 4. Average Rating:

The average customer rating given to products, reflecting satisfaction and quality perception.

QUERY:

```
SELECT CAST(AVG(Rating) AS DECIMAL(10,1)) AS Avg_Rating
FROM blinkit_data;
```



| avg_rating |
|---|
| 3.97 |

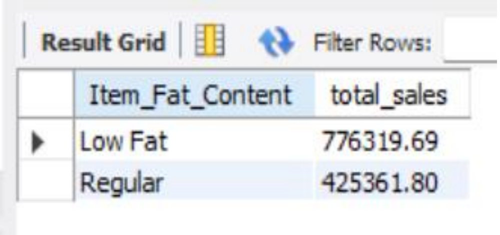
## B. Granular Requirements:

### 1. Total Sales by Fat Content

Analyze how different fat content categories influence total sales.

QUERY:

```
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;
```



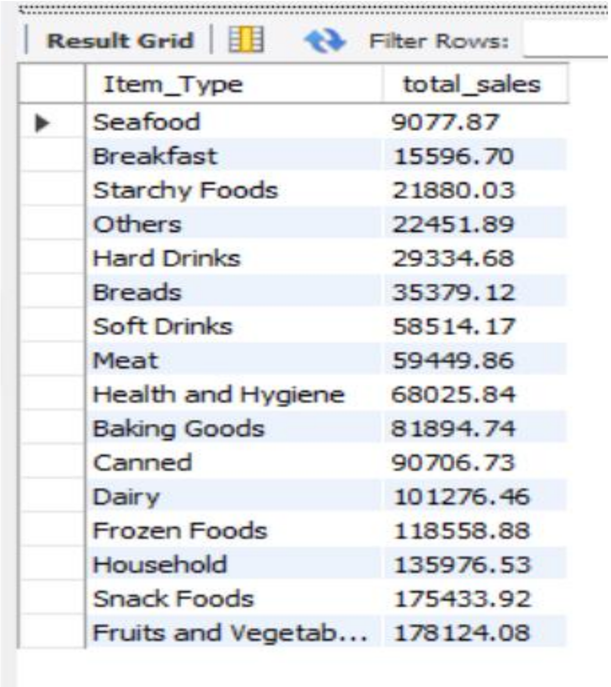
| Item_Fat_Content | total_sales |
|---|---|
| Low Fat | 776319.69 |
| Regular | 425361.80 |

## 2. Total Sales by Item Type

Evaluate which item categories contribute most to overall sales.

QUERY:

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



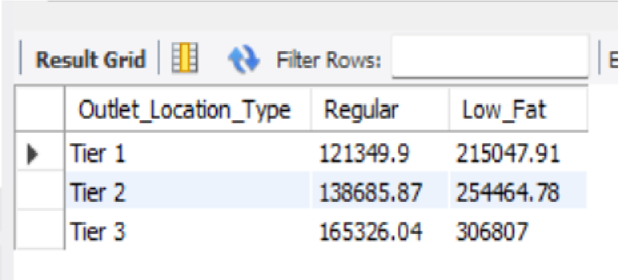
| Item_Type | total_sales |
|---|---|
| Seafood | 9077.87 |
| Breakfast | 15596.70 |
| Starchy Foods | 21880.03 |
| Others | 22451.89 |
| Hard Drinks | 29334.68 |
| Breads | 35379.12 |
| Soft Drinks | 58514.17 |
| Meat | 59449.86 |
| Health and Hygiene | 68025.84 |
| Baking Goods | 81894.74 |
| Canned | 90706.73 |
| Dairy | 101276.46 |
| Frozen Foods | 118558.88 |
| Household | 135976.53 |
| Snack Foods | 175433.92 |
| Fruits and Vegetab... | 178124.08 |

## 3. Fat Content by Outlet for Total Sales

Compare sales performance across outlets segmented by fat content.

QUERY:

```
SELECT Outlet_Location_Type,
ROUND(SUM(CASE WHEN Item_Fat_Content = 'Regular' THEN Sales ELSE 0
END), 2) AS Regular,
ROUND(SUM(CASE WHEN Item_Fat_Content = 'Low Fat' THEN Sales ELSE 0
END), 2) AS Low_Fat
FROM blinkit_data
GROUP BY Outlet_Location_Type
ORDER BY Outlet_Location_Type;
```



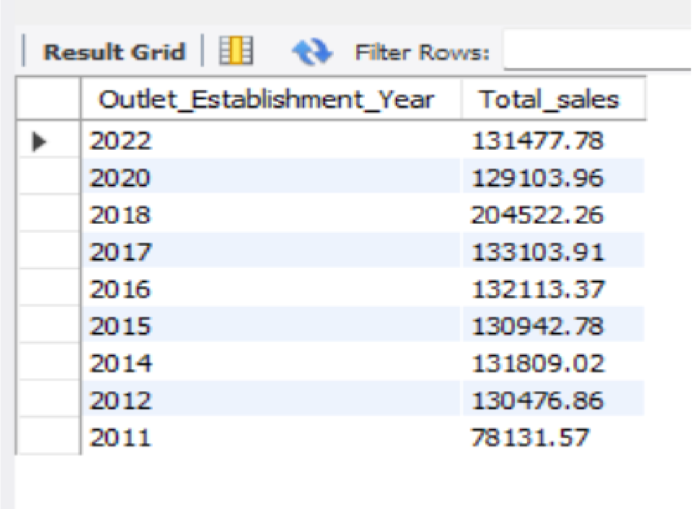
| | Outlet_Location_Type | Regular | Low_Fat |
|---|---|---|---|
| ▶ | Tier 1 | 121349.9 | 215047.91 |
| | Tier 2 | 138685.87 | 254464.78 |
| | Tier 3 | 165326.04 | 306807 |

## 4. Total Sales by Outlet Establishment Year

Understand how the age or type of the outlet influences revenue generation.

QUERY:

```
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;
```



| | Outlet_Establishment_Year | Total_sales |
|---|---|---|
| ▶ | 2022 | 131477.78 |
| | 2020 | 129103.96 |
| | 2018 | 204522.26 |
| | 2017 | 133103.91 |
| | 2016 | 132113.37 |
| | 2015 | 130942.78 |
| | 2014 | 131809.02 |
| | 2012 | 130476.86 |
| | 2011 | 78131.57 |

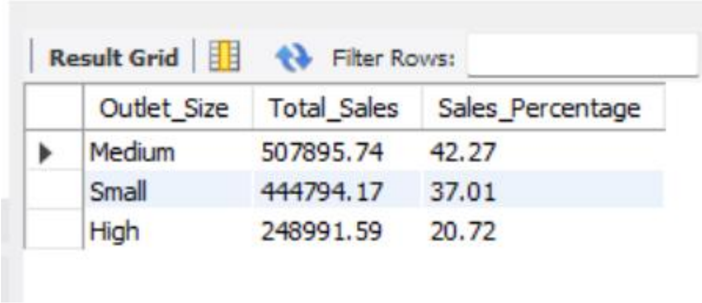
# C. Chart Requirements

## 5. Percentage of Sales by Outlet Size

Analyze the relationship between outlet size and overall sales contribution.

QUERY:

```
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;
```



Result Grid | Filter Rows:

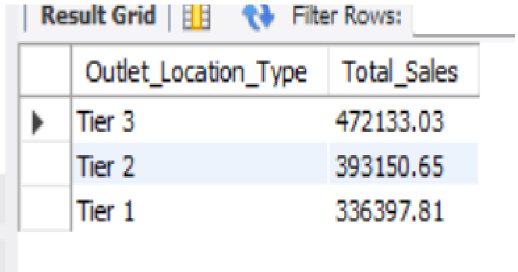
| Outlet_Size | Total_Sales | Sales_Percentage |
|---|---|---|
| Medium | 507895.74 | 42.27 |
| Small | 444794.17 | 37.01 |
| High | 248991.59 | 20.72 |

## 6. Sales by Outlet Location

Understand geographic sales distribution and identify high-performing locations.

QUERY:

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



Result Grid | Filter Rows:

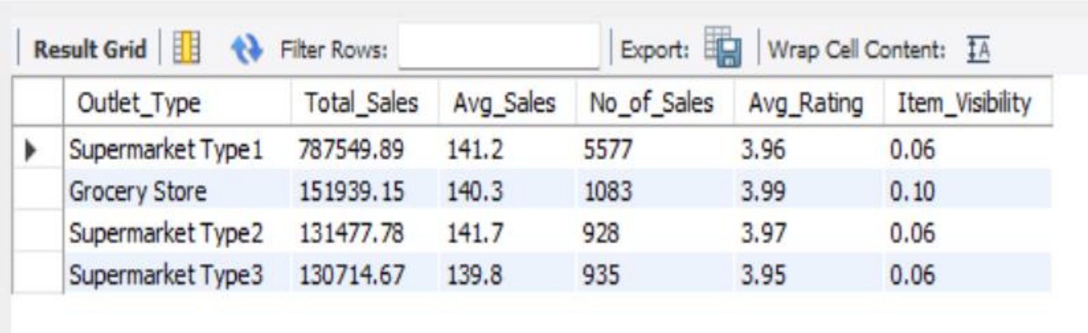
| Outlet_Location_Type | Total_Sales |
|---|---|
| Tier 3 | 472133.03 |
| Tier 2 | 393150.65 |
| Tier 1 | 336397.81 |

## 7.All Metrics by Outlet Type

Provide a consolidated view of Total Sales, Average Sales, Number of Items, and Average Rating based on outlet type.

QUERY:

```
SELECT Outlet_Type,
           CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales,
           CAST(AVG(Sales) AS DECIMAL(10,0)) AS Avg_Sales,
           COUNT(*) AS No_Of_Items,
           CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
           CAST(AVG(Item_Visibility) AS DECIMAL(10,2)) AS
Item_Visibility
FROM blinkit_data
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC;
```



| Outlet_Type | Total_Sales | Avg_Sales | No_of_Sales | Avg_Rating | Item_Visibility |
|---|---|---|---|---|---|
| Supermarket Type1 | 787549.89 | 141.2 | 5577 | 3.96 | 0.06 |
| Grocery Store | 151939.15 | 140.3 | 1083 | 3.99 | 0.10 |
| Supermarket Type2 | 131477.78 | 141.7 | 928 | 3.97 | 0.06 |
| Supermarket Type3 | 130714.67 | 139.8 | 935 | 3.95 | 0.06 |

## Conclusion:

This analysis helped identify important patterns in BlinkIT’s sales data, showing which products and outlets perform best. Using SQL the data was cleaned, analyzed that can support better decision-making. Overall, the insights gained can help improve sales, optimize inventory, and enhance customer satisfaction.