**Blink It Analysis -** Online Supermarket

**Business Requirements:**

To conduct comprehensive analysis off Blinkit’s sales performance, and inventory distribution to identify key insights and Opportunities for optimization using KPI’s and visualisation in power BI

**KPI requirements:**

1. **Total Sales:** the overall revenue generated from all items sold.
2. **Average Sales:** the average revenue per sale
3. **Number of items:** the total count of different items sold
4. **Average Rating:** the average customer rating for items sold

**Granular requirements:**

1. **Total sales by fat content:**
   1. Objective: analyse the impact of fat content on total sales
   2. Additional KPI metrics: access how other KPI’s (average sales, number of items, average rating) vary with fat content.
2. **Total sales by item type:**
   1. Objective: identify the performance of different item types in terms of total sales
   2. Additional KPI metrics: Asses how other KPI's average sales, number of items, have average rating vary with fat content.
3. **Fat content by outlet for total sales:**
   1. Objective: Compare total sales across different outlets segmented by fat content
   2. Additional KPI metrics: Asses how other KPI's average sales, number of items, have average rating vary with fat content.
4. **Total sales by outlet establishment:**
   1. Objective: evaluate how the age or type of outlet establishment influences total sales
5. **Percentage of sales by outlet size**
   1. Objective: analyse the correlation between outlet size and total sales
6. **Sales by outlet location:**
   1. Objective: Asses the geographic distribution of sales across different locations
7. **all metrics by outlet type:**
   1. Objective: Provide a comprehensive view of all key metrics (total sales, average sales, number of items, average rating) broken down by different outlet types.

**Blinkit Analysis SQL**

After Successful importing (.csv) data into SSMS select top 1000 Rows (or)

Use this simple query to see the entire data imported correctly or not.

|  |
| --- |
| SELECT \* FROM blinkit\_data |

**Data Cleaning:**

I encountered some of the names in column (Item\_Fat\_Content), Use of key words are not consistent, so updating the table column with the relevant characters in the same category as (LS, low fat) as ‘Low Fat’ and ‘reg’ as ‘Regular’ which allows the data to be more relevant and consistent. Ensuring the data in a perfect condition for further analysis.

|  |
| --- |
| 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 Updating the data using the above query check the data has been cleaned or not by using the below query.

|  |
| --- |
| SELECT DISTINCT Item\_Fat\_Content FROM blinkit\_data |

**A screenshot of a computer

AI-generated content may be incorrect.**

**KPI requirements:**

1. **What are the Total Sales?** the overall revenue generated from all items sold.

|  |
| --- |
| SELECT CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10, 2)), ' M') As Total\_Sales\_In\_Millons FROM blinkit\_data |

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **What is the Average Sale?** the average revenue per sale

|  |
| --- |
| SELECT CAST(AVG(Sales) AS DECIMAL (10,0)) AS Average\_Sale FROM blinkit\_data |

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Total Number of items:** the total count of different items sold

|  |
| --- |
| SELECT COUNT(\*) AS No\_Of\_Items FROM blinkit\_data |

A screenshot of a computer

AI-generated content may be incorrect.

1. **What is the Average Rating?** the average customer rating for items sold

|  |
| --- |
| SELECT CAST(AVG(Rating) AS DECIMAL(10,2)) AS Average\_Rating FROM blinkit\_data |

**A screenshot of a computer error

AI-generated content may be incorrect.**

1. **Total sales by fat content:**
   1. Objective: analyse the impact of fat content on total sales
   2. Additional KPI metrics: access how other KPI’s (average sales, number of items, average rating) vary with fat content.

|  |
| --- |
| SELECT Item\_Fat\_Content,  CONCAT(CAST(SUM(Sales)/1000 AS DECIMAL(10,2)),' K')AS Total\_Sales\_In\_Thousands,  CAST(AVG(Sales) AS DECIMAL(10,0)) AS Average\_Sale,  COUNT(\*) AS No\_Of\_Items,  CAST(AVG(Rating) AS DECIMAL(10,2)) AS Average\_Rating  FROM blinkit\_data  GROUP BY Item\_Fat\_Content; |

**A screenshot of a table

AI-generated content may be incorrect.**

1. **Total sales by item type:**
   1. Objective: identify the performance of different item types in terms of total sales
   2. Additional KPI metrics: Asses how other KPI's average sales, number of items, have average rating vary with fat content.

|  |
| --- |
| SELECT Item\_Type,  CAST(SUM(Sales) AS decimal (10, 2)) AS Total\_sales,  CAST(AVG(Sales) AS decimal(10, 0)) AS Average\_Sales,  COUNT(\*) AS No\_Of\_Items,  CAST(AVG(Rating) AS DECIMAL(10, 2)) AS Average\_Rating  FROM blinkit\_data  GROUP BY Item\_Type  ORDER BY Total\_sales DESC; |

A screenshot of a data

AI-generated content may be incorrect.

1. **Fat content by outlet for total sales:**
   1. Objective: Compare total sales across different outlets segmented by fat content
   2. Additional KPI metrics: Asses how other KPI's average sales, number of items, have average rating vary with fat content

|  |
| --- |
| SELECT Outlet\_Location\_Type,Item\_Fat\_Content,  CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total\_Sales FROM blinkit\_data  GROUP BY Item\_Fat\_Content, Outlet\_Location\_Type |

A screenshot of a computer

AI-generated content may be incorrect.

The above result shows the Outlet\_Location\_Type as multiple duplicates and, we can see similar in Item\_Fat\_Content. So, let’s break it down in - making Outlet\_Location\_Type AS Rows and Others in Columns which gives nice detail about the outlet information on multiple occasions like Total\_Sales, or Average\_Sales, Etc.

So, I am using a Pivot Table where changing g Rows into Columns so let’s write the query for this

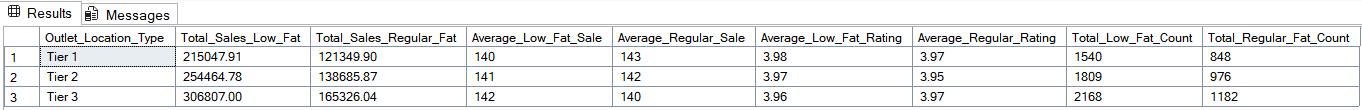
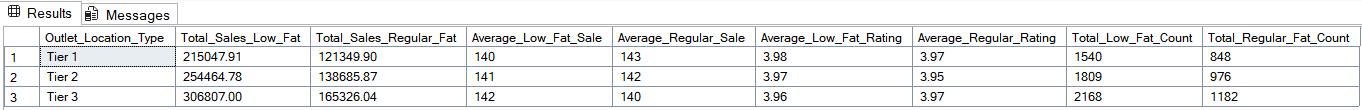
|  |
| --- |
| SELECT Outlet\_Location\_Type,  ISNULL([Low Fat],0) AS Total\_Low\_Fat\_Sales,  ISNULL([Regular],0) AS Total\_Regular\_Sales  FROM (  SELECT Outlet\_Location\_Type,Item\_Fat\_Content,  CAST(SUM(Sales) AS DECIMAL(10, 2)) AS Total\_Sales FROM blinkit\_data  GROUP BY Item\_Fat\_Content, Outlet\_Location\_Type) AS Source\_Table  PIVOT(  SUM(Total\_Sales)  FOR Item\_Fat\_Content IN ([Low Fat],[Regular])  ) AS Pivot\_Table |

A screenshot of a computer

AI-generated content may be incorrect.

Finally achieved with Ch, But We need KPI's average sales, number of items, have average rating and count also So the query should be like this below,

|  |
| --- |
| -- Pivot Total Sales  With  Total\_Sales AS (  SELECT Outlet\_Location\_Type,  ISNULL([Low Fat], 0) AS Total\_Sales\_Low\_Fat,  ISNULL([Regular], 0) AS Total\_Sales\_Regular\_Fat  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 TotalSales\_Source\_Table  PIVOT(  SUM(Total\_Sales)  FOR Item\_Fat\_Content IN ([Low Fat],[Regular])  ) AS Total\_Sales\_Pivot\_Table  ),  -- Pivot Average Sales  Average\_Sales AS (  SELECT Outlet\_Location\_Type,  ISNULL([Low Fat],0) AS Average\_Low\_Fat\_Sale,  ISNULL([Regular],0) AS Average\_Regular\_Sale  FROM  (SELECT Outlet\_Location\_Type,  Item\_Fat\_Content,  CAST(AVG(Sales) AS DECIMAL(10,0)) AS Average\_Sales  FROM blinkit\_data  GROUP BY Outlet\_Location\_Type,Item\_Fat\_Content  ) AS Average\_Sales\_Source\_Table  PIVOT(  MAX(Average\_Sales)  FOR Item\_Fat\_Content IN ([Low Fat],[Regular])  ) AS Pivot\_Average\_Sales\_Table  ),  -- Pivot Average Rating  Average\_Rating AS(  SELECT Outlet\_Location\_Type,  ISNULL([Low Fat],0) AS Average\_Low\_Fat\_Rating,  ISNULL([Regular],0) AS Average\_Regular\_Rating  FROM  (SELECT Outlet\_Location\_Type,  Item\_Fat\_Content,  CAST(AVG(Rating)AS DECIMAL(10,2)) AS Average\_Rating  FROM blinkit\_data  GROUP BY Outlet\_Location\_Type, Item\_Fat\_Content  ) As Rating\_Source\_Table  PIVOT(  MAX(Average\_Rating)  FOR Item\_Fat\_Content IN ([Low Fat],[Regular])  ) AS Pivot\_Average\_Rating\_Table  ),  -- Pivot Counting Fat\_Content  Sales\_Count AS (  SELECT Outlet\_Location\_Type,  ISNULL([Low Fat], 0) AS Total\_Low\_Fat\_Count,  ISNULL([Regular], 0) AS Total\_Regular\_Fat\_Count  FROM (  SELECT Outlet\_Location\_Type,  Item\_Fat\_Content  FROM blinkit\_data  ) AS Average\_Rating\_Source\_Table  PIVOT(  Count(Item\_Fat\_Content)  FOR Item\_Fat\_Content IN ([Low Fat],[Regular])  ) AS Pivot\_Count\_Table  )  -- Joining the CTE's For the Final Results  SELECT  TS.Outlet\_Location\_Type,  -- Total Sales  TS.Total\_Sales\_Low\_Fat,  TS.Total\_Sales\_Regular\_Fat,  -- Average Sales  ASL.Average\_Low\_Fat\_Sale,  ASL.Average\_Regular\_Sale,  -- Average Rating  AR.Average\_Low\_Fat\_Rating,  AR.Average\_Regular\_Rating,  -- Sales Count  SC.Total\_Low\_Fat\_Count,  SC.Total\_Regular\_Fat\_Count  FROM Total\_Sales TS  JOIN Average\_Sales ASL ON TS.Outlet\_Location\_Type = ASL.Outlet\_Location\_Type  JOIN Average\_Rating AR ON TS.Outlet\_Location\_Type = AR.Outlet\_Location\_Type  JOIN Sales\_Count SC ON TS.Outlet\_Location\_Type = SC.Outlet\_Location\_Type; |



The above results showing all what i expected, I am happy here. But in this I used CTE’s combined with PIVOTS for making the task possible. The result come in a length that cannot be visible, so the result is chopped in 2 pieces for easy readability

1. **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; |

A screenshot of a data

AI-generated content may be incorrect.

1. **percentage of sales by outlet size**

|  |
| --- |
| SELECT Outlet\_Size,  Round(SUM(Sales),2) AS Total\_Sales,  Concat(Round(SUM(Sales) \* 100.0 /  (SELECT SUM(Sales) FROM blinkit\_data),2),' %')  AS Total\_Percentage\_Sales  FROM blinkit\_data  GROUP BY Outlet\_Size; |

**A screenshot of a data table

AI-generated content may be incorrect.**

1. **Sales By Outlet Location**

|  |
| --- |
| SELECT Outlet\_Location\_Type,  ROUND(sum(Sales), 2)AS Total\_Sales,  Concat(Round(SUM(Sales) \* 100.0 /  (SELECT SUM(Sales) FROM blinkit\_data),2),' %')  AS Total\_Percentage\_Sales,  ROUND(AVG(Sales),2) AS Average\_Sales,  COUNT(\*) AS No\_Of\_Items,  ROUND(AVG(Rating),2) AS Average\_Rating  FROM blinkit\_data  GROUP BY Outlet\_Location\_Type |

**A screenshot of a graph

AI-generated content may be incorrect.**

1. **All Metrics by Outlet Type**

|  |
| --- |
| SELECT Outlet\_Type,  ROUND(sum(Sales), 2)AS Total\_Sales,  Concat(Round(SUM(Sales) \* 100.0 /  (SELECT SUM(Sales) FROM blinkit\_data),2),' %')  AS Total\_Percentage\_Sales,  ROUND(AVG(Sales),2) AS Average\_Sales,  COUNT(\*) AS No\_Of\_Items,  ROUND(AVG(Rating),2) AS Average\_Rating  FROM blinkit\_data  GROUP BY Outlet\_Type |

**A screenshot of a graph

AI-generated content may be incorrect.**