Saurabh Kumar

Blinkit Analysis

* See all the data imported

SELECT \* FROM blinkit\_data

After uploading the data check the uploaded data is correctly uploaded by running this quesry match the columns are numbers are same as the uploaded files

SELECT COUNT (\*) FROM blinkit\_data



* DATA CLEANING

Cleaning the Item\_Fat\_Content field ensures data consistency and accuracy in analysis. The presence of multiple variations of the same category (e.g., LF, low fat vs. Low Fat) can cause issues in reporting, aggregating and filtering. By standardizing these values we improve data quality making it easier to generate insights and maintain uniformity in our datasets.

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;



* COUNTING THE DATA

To particularly get the no of items each category has we can count query with conditions like if we need total number of items in Low fat or of rating 5 we can run query

SELECT COUNT(\*) AS No\_Of\_Items\_In\_Low\_Fat FROM blinkit\_data

WHERE Item\_Fat\_Content = 'Low Fat'



SELECT COUNT(\*) AS No\_Of\_Items\_of\_5\_Rating FROM blinkit\_data

WHERE Rating = 5



* TOTAL SALES

To calculate the total sales we run this query to get the data in millions

SELECT CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)), ' Millions') AS Total\_Sales\_in\_Millions

FROM blinkit\_data



To get the data for Items\_Fat\_Content Category wise data we can run these query

SELECT CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)), ' Millions') AS Total\_Sales\_in\_Millions

FROM blinkit\_data

WHERE Item\_Fat\_Content = 'Low Fat'



SELECT CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)), ' Millions') AS Total\_Sales\_in\_Millions

FROM blinkit\_data

WHERE Item\_Fat\_Content = 'Regular'

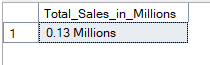


Also if we need sales data for Oultlet\_Establishment\_Year we can run query

SELECT CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)), ' Millions') AS Total\_Sales\_in\_Millions

FROM blinkit\_data

WHERE Outlet\_Establishment\_Year = 2017



* Average Sales

Similarly for average sales

SELECT CAST(AVG(Sales) AS DECIMAL(10,1)) AS Average\_Sales

FROM blinkit\_data



If we need average sales for peerticular category or particular Establishment we can run similar query as sales by just changing the SUM to AVG also we need to remove the million’s part because average sales is a very small number that we can read easily

* Average Rating

SELECT CAST(AVG(Rating) AS DECIMAL(10,1)) AS Average\_Rating

FROM blinkit\_data

Instead of getting all these data separately we can run query at one places to get all data at one place for better insights and visualization

SELECT Item\_Fat\_Content,

CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)), ' Millions') AS Total\_sales,

CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg\_Sales,

COUNT (\*) AS No\_Of\_Items,

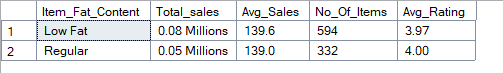
CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg\_Rating

FROM blinkit\_data

WHERE Outlet\_Establishment\_Year = 2020

GROUP BY Item\_Fat\_Content

ORDER BY Total\_sales DESC



* Total Sales by Item types

SELECT Item\_Type,

CONCAT(CAST(SUM(Sales) AS DECIMAL(10,2)), ' Millions') AS Total\_sales,

CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg\_Sales,

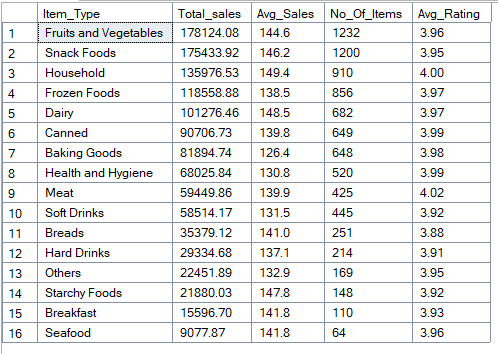
COUNT (\*) AS No\_Of\_Items,

CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg\_Rating

FROM blinkit\_data

GROUP BY Item\_Type

ORDER BY Total\_sales DESC



If we want data of best 5 Sold items we can run query

SELECT TOP 5 Item\_Type,

CONCAT(CAST(SUM(Sales) AS DECIMAL(10,2)), ' Millions') AS Total\_sales,

CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg\_Sales,

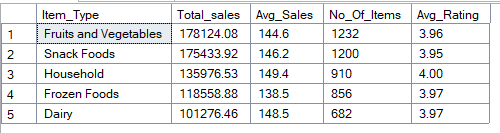
COUNT (\*) AS No\_Of\_Items,

CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg\_Rating

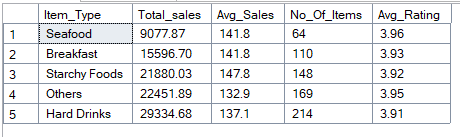
FROM blinkit\_data

GROUP BY Item\_Type

ORDER BY Total\_sales DESC



Similarly for bottom 5 we change DESC to ASC



* Fat 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 Total\_Sales

FROM blinkit\_data

GROUP BY Outlet\_Location\_Type, Item\_Fat\_Content

) AS SourceTable

PIVOT

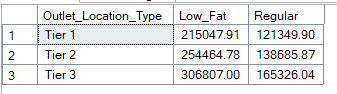
(

SUM(Total\_Sales)

FOR Item\_Fat\_Content IN ([Low Fat], [Regular])

) AS PivotTable

ORDER BY Outlet\_Location\_Type;



* Total Sales by Outlet Establishment Year

SELECT Outlet\_Establishment\_Year,

CAST(SUM(Sales) AS DECIMAL (10,2)) AS Total\_sales

CAST(AVG(Sales) AS DECIMAL(10,1)) AS Avg\_Sales,

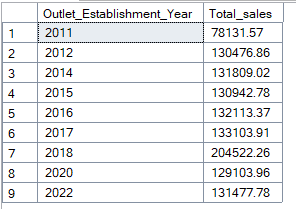
COUNT(\*) AS No\_Of\_Items,

CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg\_Rating

FROM blinkit\_data

GROUP BY Outlet\_Establishment\_Year

ORDER BY Outlet\_Establishment\_Year ASC



* Percentage of Sales by Outlet Size

SELECT Outlet\_Size,

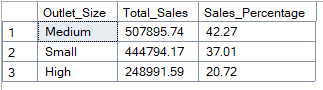
CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total\_Sales,

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;



* Sales by 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,

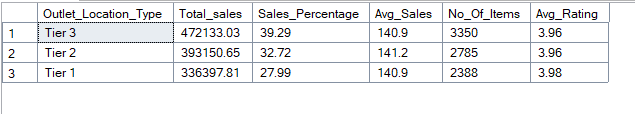
COUNT(\*) AS No\_Of\_Items,

CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg\_Rating

FROM blinkit\_data

GROUP BY Outlet\_Location\_Type

ORDER BY Total\_Sales DESC;



* 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;

