

# SQL Project : Zepto

## Step 1: Getting familiar with the dataset

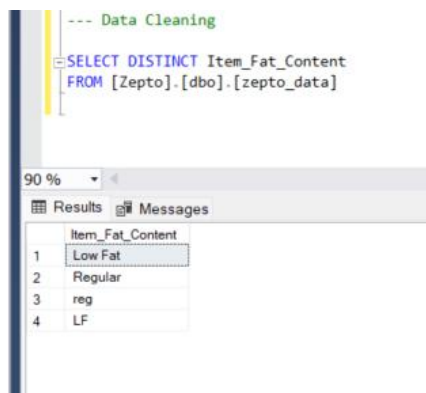
such as – number of columns, datatype, columns names, number of records, etc

```
SELECT * FROM [Zepto].[dbo].[zepto_data]
```

```
SELECT COUNT(*) FROM [Zepto].dbo.zepto_data
```

## Step 2: Data Cleaning

We can see in below image that the field Item\_Fat\_Content has multiple variations of the same category (e.g., LF, Low Fat) that can cause issues in reporting, aggregations, and filtering.



The screenshot shows a SQL query editor with the following query:

```
--- Data Cleaning  
SELECT DISTINCT Item_Fat_Content  
FROM [Zepto].[dbo].[zepto_data]
```

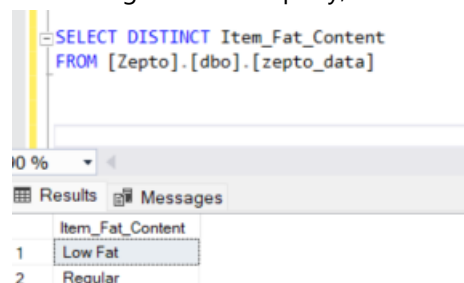
Below the query, the 'Results' tab is active, displaying a table with the following data:

	Item_Fat_Content
1	Low Fat
2	Regular
3	reg
4	LF

We need to clean this field to ensure data consistency and accuracy in analysis. By standardizing these values, we can improve data quality, making it easier to generate insights and maintain uniformity in our datasets.

```
UPDATE [zepto_data]  
SET Item_Fat_Content =  
CASE WHEN [Item_Fat_Content] = 'LF' THEN 'Low Fat'  
      WHEN [Item_Fat_Content] = 'reg' THEN 'Regular'  
      ELSE [Item_Fat_Content]  
END
```

After firing this above query, as a result we get below results:



The screenshot shows the same SQL query as before, but the results are now standardized:

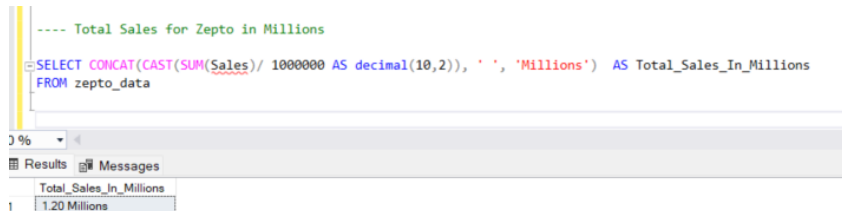
	Item_Fat_Content
1	Low Fat
2	Regular

## Step 3: Calculating KPIs

### Total Sales

(in Millions)

```
SELECT CONCAT(CAST(SUM(Sales)/ 1000000 AS decimal(10,2)), ' ', 'Millions') AS  
Total_Sales_In_Millions  
FROM zepto_data
```



The screenshot shows a SQL query editor with the following query: `SELECT CONCAT(CAST(SUM(Sales)/ 1000000 AS decimal(10,2)), ' ', 'Millions') AS Total_Sales_In_Millions FROM zepto_data`. Below the query, the 'Results' tab is active, displaying a single row with the value '1.20 Millions' under the column header 'Total\_Sales\_In\_Millions'.

Total_Sales_In_Millions
1.20 Millions

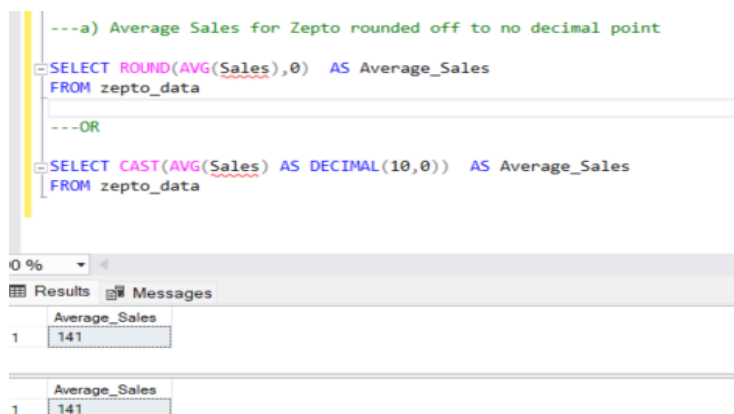
### Average Sales

(rounded off to zero decimal point)

```
SELECT ROUND(AVG(Sales),0) AS Average_Sales  
FROM zepto_data
```

---OR

```
SELECT CAST(AVG(Sales) AS DECIMAL(10,0)) AS Average_Sales  
FROM zepto_data
```



The screenshot shows two SQL queries in a query editor. The first query is `SELECT ROUND(AVG(Sales),0) AS Average_Sales FROM zepto_data`. The second query is `SELECT CAST(AVG(Sales) AS DECIMAL(10,0)) AS Average_Sales FROM zepto_data`. Below the queries, the 'Results' tab is active, displaying a single row with the value '141' under the column header 'Average\_Sales'.

Average_Sales
141

## Average Customer Rating

```
SELECT ROUND(AVG([Customer Rating]),2) AS Average_Customer_Rating
FROM zepto_data
```

The screenshot shows a SQL query in a query editor and its results in a table. The query calculates the average customer rating from the 'zepto\_data' table, rounded to two decimal places. The result is displayed in a table with one row and one column.

```
-- e) Average Customer Rating
SELECT ROUND(AVG([Customer Rating]),2) AS Average_Customer_Rating
FROM zepto_data
```

Average_Customer_Rating
3.97

## KPIs by Fat Content

```
SELECT [Item_Fat_Content]
      , CONCAT(CAST(SUM(Sales)/ 1000000 AS decimal(10,2)), ' ', 'Millions') AS Total_Sales_In_Millions
      , CAST(AVG(Sales) AS DECIMAL(10,0)) AS Average_Sales
      , COUNT(*) AS Number_of_Items
      , ROUND(AVG([Customer Rating]),2) AS Average_Customer_Rating
FROM zepto_data
GROUP BY [Item_Fat_Content]
ORDER BY Item_Fat_Content
```

The screenshot shows a SQL query in a query editor and its results in a table. The query calculates various KPIs grouped by 'Item\_Fat\_Content' from the 'zepto\_data' table. The results are displayed in a table with five columns: Item\_Fat\_Content, Total\_Sales\_In\_Millions, Average\_Sales, Number\_of\_Items, and Average\_Customer\_Rating.

```
-- KPIs by Fat Content
SELECT [Item_Fat_Content]
      , CONCAT(CAST(SUM(Sales)/ 1000000 AS decimal(10,2)), ' ', 'Millions') AS Total_Sales_In_Millions
      , CAST(AVG(Sales) AS DECIMAL(10,0)) AS Average_Sales
      , COUNT(*) AS Number_of_Items
      , ROUND(AVG([Customer Rating]),2) AS Average_Customer_Rating
FROM zepto_data
GROUP BY [Item_Fat_Content]
ORDER BY Item_Fat_Content
```

Item_Fat_Content	Total_Sales_In_Millions	Average_Sales	Number_of_Items	Average_Customer_Rating
Low Fat	0.78 Millions	141	5517	3.97
Regular	0.43 Millions	142	3006	3.97

## Top 5 selling items for Zepto

```
SELECT [Item_Type] Top_5_selling_items
from (
SELECT TOP 5 [Item_Type]
FROM zepto_data
GROUP BY [Item_Type]
ORDER BY CONCAT(CAST(SUM(Sales)/ 1000000 AS decimal(10,2)), ' ', 'Millions')
) p
ORDER BY [Item_Type]
ORDER BY Item_Type
```

```

---- Top 5 selling items

SELECT [Item_Type] Top_5_selling_items
from (
  SELECT TOP 5 [Item_Type]
  FROM zepto_data
  GROUP BY [Item_Type]
  ORDER BY CONCAT(CAST(SUM(Sales)/ 1000000 AS decimal(10,2)), ' ', 'Millions')
) p
ORDER BY [Item_Type]

```

90 %

Results Messages

	Top_5_selling_items
1	Breakfast
2	Hard Drinks
3	Others
4	Seafood
5	Starchy Foods

## Fat Content by Outlet for Total Sales

```

SELECT Outlet_Location_Type,
       ISNULL([Low Fat], 0) AS Low_Fat_Sales_In_Millions,
       ISNULL([Regular], 0) AS Regular_Sales_In_Millions
FROM
(
  SELECT Outlet_Location_Type, Item_Fat_Content,
         CAST(SUM(Sales)/ 1000000 AS decimal(10,2)) AS Total_Sales_Millions
  FROM zepto_data
  GROUP BY Outlet_Location_Type, Item_Fat_Content
) AS t1
PIVOT
(
  SUM(Total_Sales_Millions)
  FOR Item_Fat_Content IN ([Low Fat], [Regular])
) AS PivotTable
ORDER BY Outlet_Location_Type;

```

```

-- Fat Content by Outlet for Total Sales

SELECT Outlet_Location_Type,
       ISNULL([Low Fat], 0) AS Low_Fat_Sales_In_Millions,
       ISNULL([Regular], 0) AS Regular_Sales_In_Millions
FROM
(
  SELECT Outlet_Location_Type, Item_Fat_Content,
         CAST(SUM(Sales)/ 1000000 AS decimal(10,2)) AS Total_Sales_Millions
  FROM zepto_data
  GROUP BY Outlet_Location_Type, Item_Fat_Content
) AS t1
PIVOT
(
  SUM(Total_Sales_Millions)
  FOR Item_Fat_Content IN ([Low Fat], [Regular])
) AS PivotTable
ORDER BY Outlet_Location_Type;

```

90 %

Results Messages

	Outlet_Location_Type	Low_Fat_Sales_In_Millions	Regular_Sales_In_Millions
1	Tier 1	0.22	0.12
2	Tier 2	0.25	0.14
3	Tier 3	0.31	0.17

This query aims to transform the Zepto\_data table to display total sales (in millions) for each combination of Outlet\_Location\_Type and Item\_Fat\_Content. The result will show Outlet\_Location\_Type as rows and Item\_Fat\_Content categories ("Low Fat" and "Regular") as columns. If there are no sales for a particular combination, the query will display 0 instead of NULL.

### Total Sales by Outlet Establishment

```
SELECT Outlet_Establishment_Year , Outlet_Identifier
      , CAST(SUM(Sales) AS decimal(10,2)) AS Total_Sales
FROM zepto_data
GROUP BY Outlet_Establishment_Year, Outlet_Identifier
ORDER BY CAST(SUM(Sales) AS decimal(10,2)) DESC
```

The screenshot shows a SQL query window with the following query:

```
--- Total Sales by Outlet Establishment
SELECT Outlet_Establishment_Year , Outlet_Identifier
      , CAST(SUM(Sales) AS decimal(10,2)) AS Total_Sales
FROM zepto_data
GROUP BY Outlet_Establishment_Year, Outlet_Identifier
ORDER BY CAST(SUM(Sales) AS decimal(10,2)) DESC
```

Below the query, the 'Results' tab is active, displaying a table with 10 rows and 3 columns: Outlet\_Establishment\_Year, Outlet\_Identifier, and Total\_Sales. The results are sorted by Total\_Sales in descending order.

Outlet_Establishment_Year	Outlet_Identifier	Total_Sales
2017	OUT035	133103.91
2010	OUT046	132113.37
2000	OUT013	131809.02
2022	OUT018	131477.77
2015	OUT045	130942.78
1998	OUT027	130714.67
2012	OUT049	130476.86
2020	OUT017	129103.96
2011	OUT010	78131.56
1998	OUT019	73807.58

### Percentage of Sales by Outlet Size

```
SELECT
  Outlet_Size,
  CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales
  , ROUND((CAST(SUM(Sales) AS DECIMAL(10,2))/SUM(SUM(Sales)) OVER())*100,2) Sales_Percentage
FROM zepto_data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC;
```

The screenshot shows a SQL query window with the following query:

```
SELECT
  Outlet_Size,
  CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales
  , ROUND((CAST(SUM(Sales) AS DECIMAL(10,2))/SUM(SUM(Sales)) OVER())*100,2) Sales_Percentage
FROM zepto_data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC;
```

Below the query, the 'Results' tab is active, displaying a table with 3 rows and 3 columns: Outlet\_Size, Total\_Sales, and Sales\_Percentage. The results are sorted by Total\_Sales in descending order.

Outlet_Size	Total_Sales	Sales_Percentage
Medium	507895.73	42.27
Small	444794.17	37.01
High	248991.58	20.72

**SUM(Total\_Sales):** Within the GROUP BY context, this computes the total sales for each Outlet\_Size.

**SUM(... ) OVER():** The outer SUM combined with the OVER() clause calculates the grand total of all Total\_Sales across all outlet sizes without collapsing the result set.