**BlinkIT\_Analysis\_Using\_Sql\_Queries:**

**SELECT \* FROM blinkit\_db.blinkit\_grocery\_data;**

-- select count(\*) from blinkit\_db.blinkit\_grocery\_data;

* **DATA Cleaning:**

-- UPDATE blinkit\_db.blinkit\_grocery\_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;

//To check how many distinct columns are there in Item\_Fat\_Content;

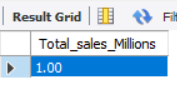
–SELECT DISTINCT(Item\_Fat\_Content) FROM blinkit\_db.blinkit\_grocery\_data;

* **-- KPI requirements:**

1. **Total Sales: the overall revenue generated from all of its items sold:**

select cast(sum(sales)/ 1000000 as DECIMAL(10,2)) as Total\_sales\_Millions

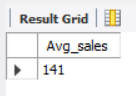
from blinkit\_db.blinkit\_grocery\_data;



1. **Average Sales: Average sales revenue per sale:**

select cast(AVG(sales) AS DECIMAL(10,0)) as Avg\_sales

from blinkit\_db.blinkit\_grocery\_data



1. **Number of items: the total count of items sold;**

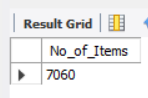
SELECT count(\*) as No\_of\_Items FROM blinkit\_db.blinkit\_grocery\_data;

**OR**

select cast(AVG(sales) AS DECIMAL(10,0)) as Avg\_sales

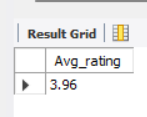
from blinkit\_db.blinkit\_grocery\_data

where Item\_Fat\_Content= 'Low fat';



1. **Average Rating : The average customer rating for items sold:**

select cast(avg(Rating) as DECIMAL(10,2)) as Avg\_rating from blinkit\_db.blinkit\_grocery\_data;



* **GRANULAR REQUIREMENTS:**

1. **Total sales by fats content:**

select Item\_Fat\_Content,

cast(sum(Sales)/1000 as decimal(10,2) )as Total\_sales\_Thousands,

cast(AVG(sales) AS DECIMAL(10,0)) as Avg\_sales,

count(\*) as No\_of\_Items,

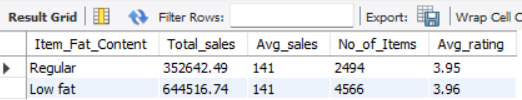
cast(avg(Rating) as DECIMAL(10,2)) as Avg\_rating

from blinkit\_db.blinkit\_grocery\_data

where Outlet\_Establishment\_Year= 2020

group by Item\_Fat\_Content

order by Total\_sales\_Thousands



1. **Tota; sales by item Type;**

select Item\_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

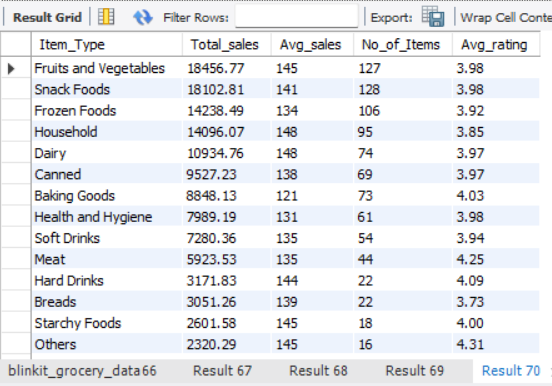
from blinkit\_db.blinkit\_grocery\_data

where Outlet\_Establishment\_Year= 2020

group by Item\_Type

order by Total\_sales desc

limit 5



1. **Fat content by outlet for total sales:**

select Outlet\_Location\_Type, Item\_Fat\_Content,

cast(sum(Sales) as decimal(10,2) )as Total\_sales,

cast(AVG(sales) AS DECIMAL(10,0)) as Avg\_sales,

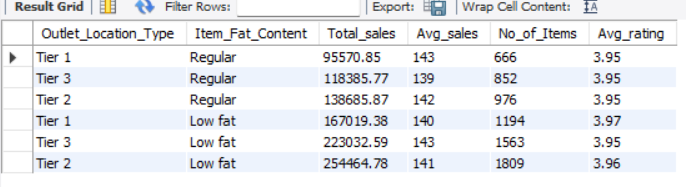
count(\*) as No\_of\_Items,

cast(avg(Rating) as DECIMAL(10,2)) as Avg\_rating

from blinkit\_db.blinkit\_grocery\_data

group by Outlet\_Location\_Type, Item\_Fat\_Content

order by Total\_sales asc;



-- 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\_db.blinkit\_grocery\_data

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

-- AS SourceTable

-- )

-- PIVOT

-- (

-- SUM(Sales)

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

-- )AS PivotTable

-- ORDER BY Outlet\_Location\_Type;

// CORRECTED ONE:

SELECT

Outlet\_Location\_Type,

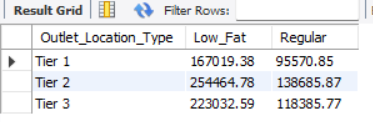
CAST(IFNULL(SUM(CASE WHEN Item\_Fat\_Content = 'Low Fat' THEN Sales ELSE 0 END), 0) AS DECIMAL(10,2)) AS Low\_Fat,

CAST(IFNULL(SUM(CASE WHEN Item\_Fat\_Content = 'Regular' THEN Sales ELSE 0 END), 0) AS DECIMAL(10,2)) AS Regular

FROM blinkit\_db.blinkit\_grocery\_data

GROUP BY Outlet\_Location\_Type

ORDER BY Outlet\_Location\_Type;



1. **Total sales by outlet establishment;**

select Outlet\_Establishment\_Year,

cast(sum(Sales) as decimal(10,2) )as Total\_sales,

cast(AVG(sales) AS DECIMAL(10,0)) as Avg\_sales,

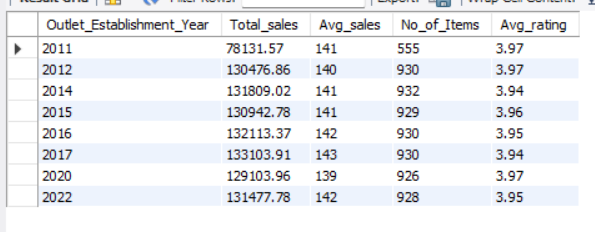
count(\*) as No\_of\_Items,

cast(avg(Rating) as DECIMAL(10,2)) as Avg\_rating

from blinkit\_db.blinkit\_grocery\_data

group by Outlet\_Establishment\_Year

order by Outlet\_Establishment\_Year asc;



1. **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\_db.blinkit\_grocery\_data

GROUP BY Outlet\_Size

ORDER BY Total\_Sales DESC;

\*/

SELECT

Outlet\_Size,

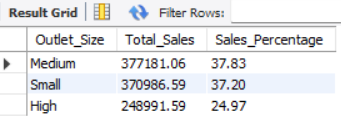
CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total\_Sales,

CAST((SUM(Sales) \* 100.0 / (SELECT SUM(Sales) FROM blinkit\_db.blinkit\_grocery\_data)) AS DECIMAL(10,2)) AS Sales\_Percentage

FROM blinkit\_db.blinkit\_grocery\_data

GROUP BY Outlet\_Size

ORDER BY Total\_Sales DESC;



1. **Sales by outlet location:**

select Outlet\_Location\_Type,

cast(sum(Sales) as decimal(10,2) )as Total\_sales,

cast(AVG(sales) AS DECIMAL(10,0)) as Avg\_sales,

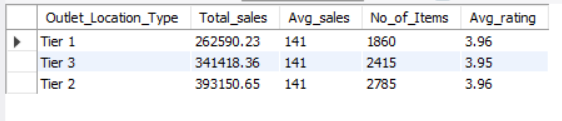
count(\*) as No\_of\_Items,

cast(avg(Rating) as DECIMAL(10,2)) as Avg\_rating

from blinkit\_db.blinkit\_grocery\_data

group by Outlet\_Location\_Type

order by Total\_sales asc;



1. **All metrics by outlet type:**

select Outlet\_Type,

cast(sum(Sales) as decimal(10,2) )as Total\_sales,

cast(AVG(sales) AS DECIMAL(10,0)) as Avg\_sales,

CAST((SUM(Sales) \* 100.0 / (SELECT SUM(Sales) FROM blinkit\_db.blinkit\_grocery\_data)) AS DECIMAL(10,2)) AS Sales\_Percentage,

count(\*) as No\_of\_Items,

cast(avg(Rating) as DECIMAL(10,2)) as Avg\_rating

from blinkit\_db.blinkit\_grocery\_data

-- where Outlet\_Establishment\_Year = 2020

group by Outlet\_Type

order by Total\_sales asc;

