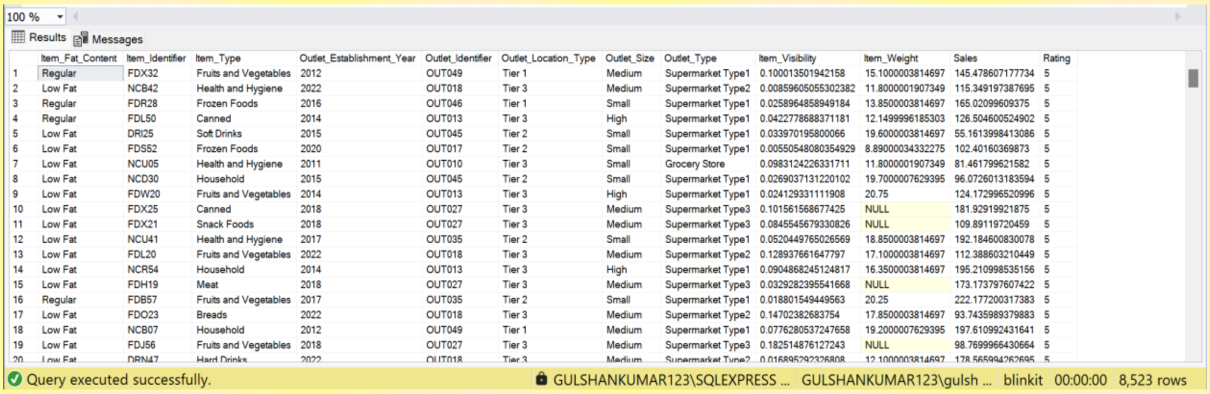
BLINKIT DATA ANALYSIS DOCUMENTATION (SQL)

1.Imported the Data into Sql

Then checked all the Data has been imported or not?

select * from blinkit_data



	Item_Fat_Content	Item_Identifier	Item_Type	Outlet_Establishment_Year	Outlet_Identifier	Outlet_Location_Type	Outlet_Size	Outlet_Type	Item_Visibility	Item_Weight	Sales	Rating
1	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100013501942158	15.1000003814697	145.478607177734	5
2	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.00859605055302382	11.8000001907349	115.349197387695	5
3	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Supermarket Type1	0.0258964858949184	13.8500003814697	165.02099609375	5
4	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Supermarket Type1	0.0422778688371181	12.1499996185303	126.504600524902	5
5	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970195800066	19.6000003814697	55.1613998413086	5
6	Low Fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.00550548080354929	8.89000034332275	102.4016036987 3	5
7	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.0983124226331711	11.8000001907349	81.461799621582	5
8	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.0269037131220102	19.7000007629395	96.0726013183594	5
9	Low Fat	FDW20	Fruits and Vegetables	2014	OUT013	Tier 3	High	Supermarket Type1	0.024129331111908	20.75	124.172996520996	5
10	Low Fat	FDX25	Canned	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.101561568677425	NULL	181.92919921875	5
11	Low Fat	FDX21	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.0845545679330826	NULL	109.89119720459	5
12	Low Fat	NCU41	Health and Hygiene	2017	OUT035	Tier 2	Small	Supermarket Type1	0.0520449765026569	18.8500003814697	192.184600830078	5
13	Low Fat	FDL20	Fruits and Vegetables	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.128937661647797	17.1000003814697	112.388603210449	5
14	Low Fat	NCR54	Household	2014	OUT013	Tier 3	High	Supermarket Type1	0.0904868245124817	16.3500003814697	195.210998535156	5
15	Low Fat	FDH19	Meat	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.0329282395541668	NULL	173.173797607422	5
16	Regular	FDB57	Fruits and Vegetables	2017	OUT035	Tier 2	Small	Supermarket Type1	0.018801549449563	20.25	222.177200317383	5
17	Low Fat	FDO23	Breads	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.14702382683754	17.8500003814697	93.7435989379883	5
18	Low Fat	NCB07	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.0776280537247658	19.2000007629395	197.610992431641	5
19	Low Fat	FDJ56	Fruits and Vegetables	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.182514876127243	NULL	98.7699966430664	5
20	Low Fat	DRN47	Hard Drinks	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.016895292326808	12.1000003814697	178.565994262695	5

✓ Query executed successfully. GULSHANKUMAR123\SQLEXPRESS ... GULSHANKUMAR123\gulsh ... blinkit 00:00:00 8,523 rows

Got all my 8523 rows.

Then

2. Data Cleaning has been Done on the column Item_Fat_Content

In that Low Fat and Regular are two distinct categories but there LF and low fat was also present so I updated them to Low Fat and same for Regular where reg was written I updated it to Regular

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'

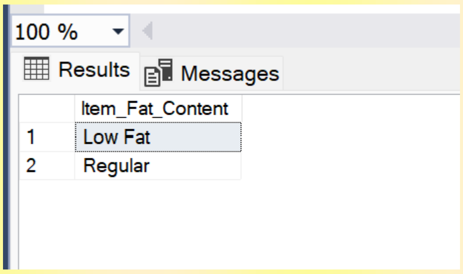
else Item_Fat_Content

End

Then I checked that updation has been done or not by using

select distinct(Item_Fat_Content) from blinkit_data

I got the output as

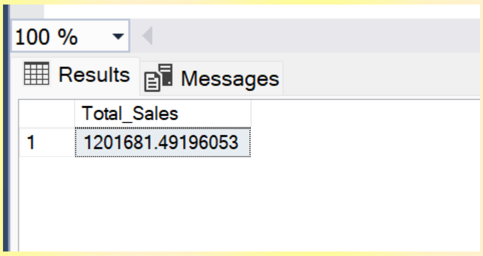


	Item_Fat_Content
1	Low Fat
2	Regular

3. Started With KPIs

To get the Total Sales

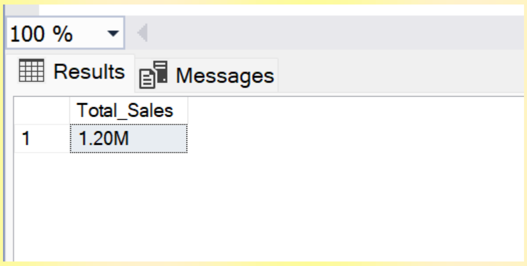
select SUM(Sales) as Total_Sales From blinkit_data



	Total_Sales
1	1201681.49196053

I casted as it into the decimal and also used concat(,’M’) for symbolising it as Million

select CONCAT(CAST(SUM(Sales)/1000000 as decimal(10,2)),'M') as Total_Sales From blinkit_data

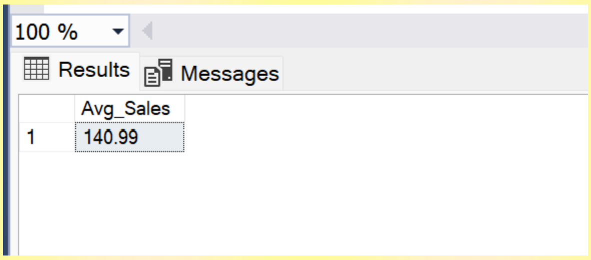


	Total_Sales
1	1.20M

4. To get the Average Sales

In that I rounded up to 2 places because I wanted value not so large in decimal point so I rounded it up

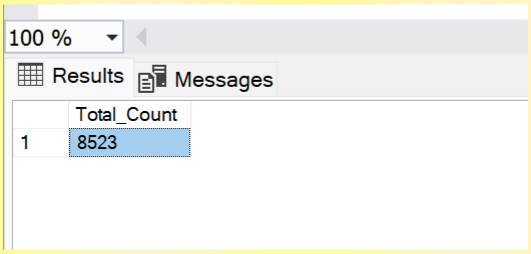
select round(avg(Sales),2) as Avg_Sales from blinkit_data



	Avg_Sales
1	140.99

5. To get the Total Numbers of Orders

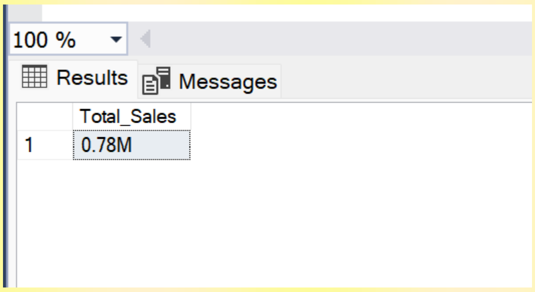
```
select count(Item_Fat_Content) as Total_Count from blinkit_data
```



	Total_Count
1	8523

6. To get the Sales of Low Fat

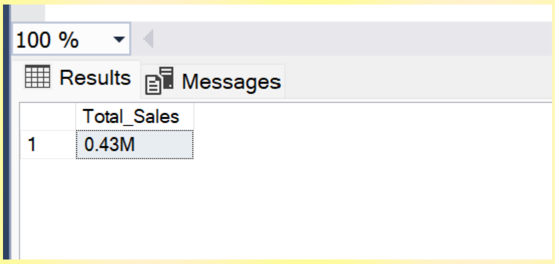
```
select CONCAT(CAST(SUM(Sales)/1000000 as decimal(10,2)),'M') as Total_Sales From
blinkit_data

where Item_Fat_Content='Low Fat';
```



	Total_Sales
1	0.78M

7. To get the Total sales by Regular fat

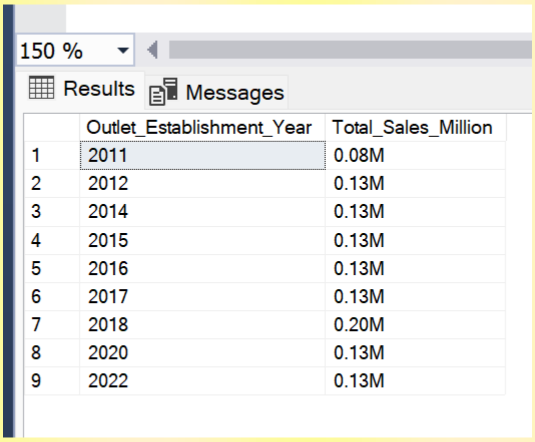
```
select CONCAT(CAST(SUM(Sales)/1000000 as decimal(10,2)),'M') as Total_Sales From
blinkit_data

where Item_Fat_Content='Regular';
```



	Total_Sales
1	0.43M

8. Total Sales Per Year

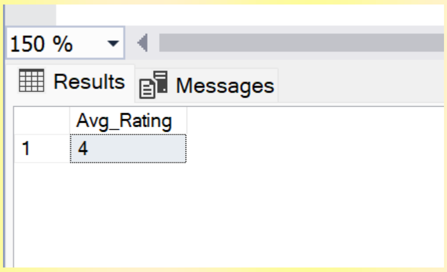
```
SELECT
  Outlet_Establishment_Year,
  CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)), 'M') AS Total_Sales_Million
FROM blinkit_data
WHERE Outlet_Establishment_Year IN (2011, 2012, 2014, 2015, 2016, 2017, 2018, 2020, 2022)
GROUP BY Outlet_Establishment_Year
ORDER BY Outlet_Establishment_Year;
```



	Outlet_Establishment_Year	Total_Sales_Million
1	2011	0.08M
2	2012	0.13M
3	2014	0.13M
4	2015	0.13M
5	2016	0.13M
6	2017	0.13M
7	2018	0.20M
8	2020	0.13M
9	2022	0.13M

9. Average Rating of Blinkit

```
select round(avg(Rating),1) as Avg_Rating from blinkit_data
```

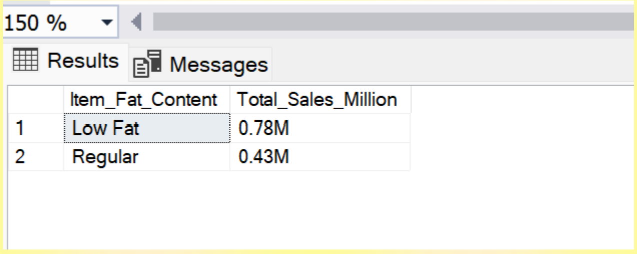


	Avg_Rating
1	4

10.Total Sales By Fat Content

```
SELECT
  Item_Fat_Content,
  CONCAT(CAST(SUM(Sales)/1000000 AS DECIMAL(10,2)), 'M') AS Total_Sales_Million
FROM blinkit_data
WHERE Item_Fat_Content IN ('Low Fat','Regular')
```

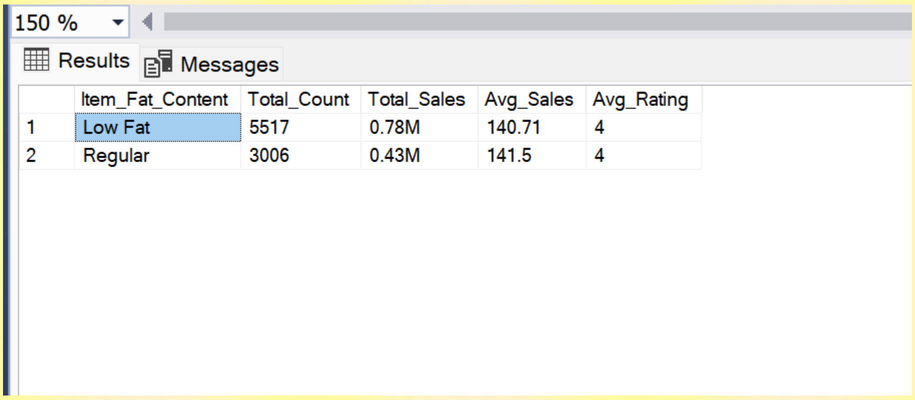
```
GROUP BY Item_Fat_Content
ORDER BY Item_Fat_Content;
```



	Item_Fat_Content	Total_Sales_Million
1	Low Fat	0.78M
2	Regular	0.43M

11.Variation in Data according to Fat Types

```
SELECT
  Item_Fat_Content,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000000,2),'M')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),1) as Avg_Rating
FROM blinkit_data
GROUP BY Item_Fat_Content
ORDER BY Avg_Sales;
```

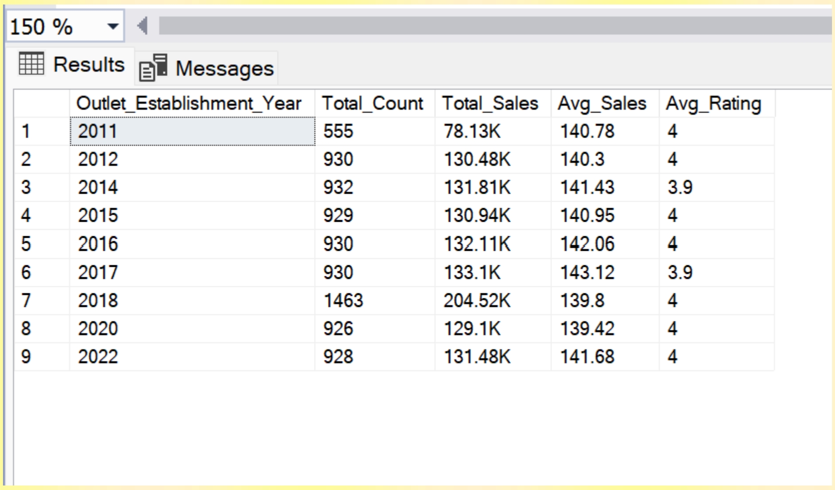


	Item_Fat_Content	Total_Count	Total_Sales	Avg_Sales	Avg_Rating
1	Low Fat	5517	0.78M	140.71	4
2	Regular	3006	0.43M	141.5	4

12.Variation in Data According to Year:

```
SELECT
```

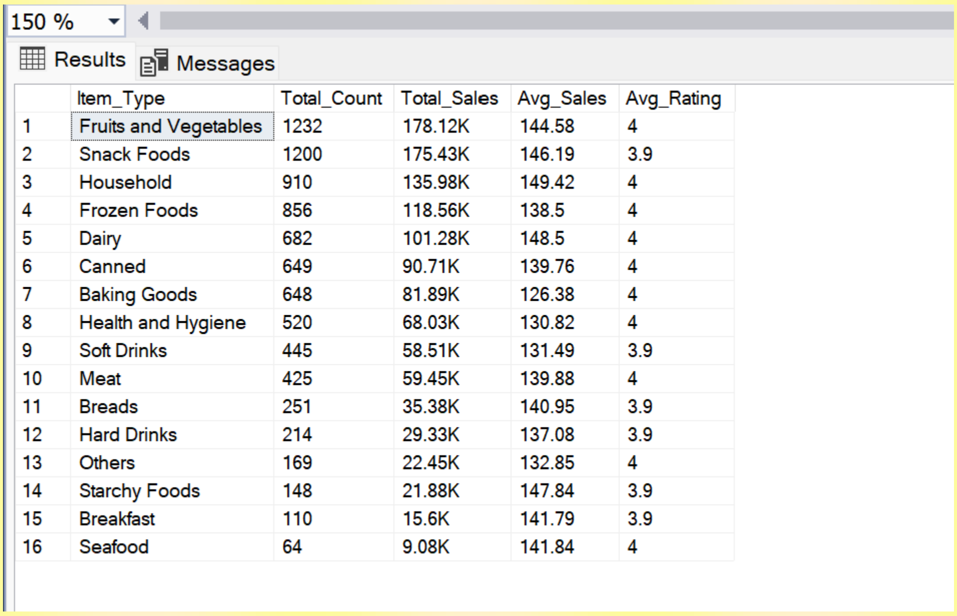
```
  Outlet_Establishment_Year,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000,2),'K')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),1) as Avg_Rating
FROM blinkit_data
WHERE Outlet_Establishment_Year IN (2011, 2012, 2014, 2015, 2016, 2017, 2018, 2020, 2022)
GROUP BY Outlet_Establishment_Year
ORDER BY Outlet_Establishment_Year;
```



	Outlet_Establishment_Year	Total_Count	Total_Sales	Avg_Sales	Avg_Rating
1	2011	555	78.13K	140.78	4
2	2012	930	130.48K	140.3	4
3	2014	932	131.81K	141.43	3.9
4	2015	929	130.94K	140.95	4
5	2016	930	132.11K	142.06	4
6	2017	930	133.1K	143.12	3.9
7	2018	1463	204.52K	139.8	4
8	2020	926	129.1K	139.42	4
9	2022	928	131.48K	141.68	4

13. Data Variation by Item_Type:

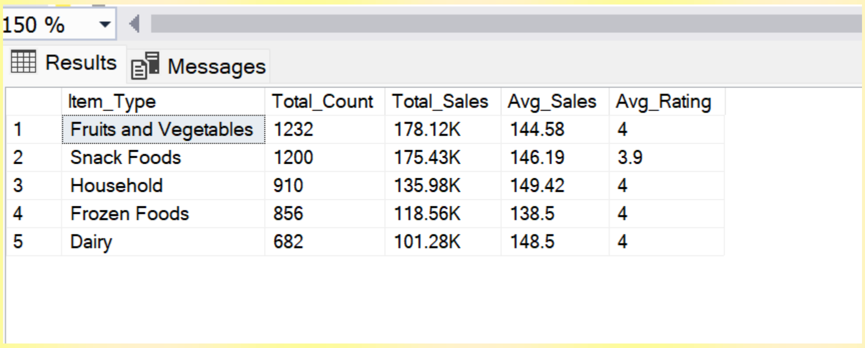
```
SELECT
  Item_Type,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000,2),'K')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),1) as Avg_Rating
FROM blinkit_data
GROUP BY Item_Type
ORDER BY Total_Count desc;
```



	Item_Type	Total_Count	Total_Sales	Avg_Sales	Avg_Rating
1	Fruits and Vegetables	1232	178.12K	144.58	4
2	Snack Foods	1200	175.43K	146.19	3.9
3	Household	910	135.98K	149.42	4
4	Frozen Foods	856	118.56K	138.5	4
5	Dairy	682	101.28K	148.5	4
6	Canned	649	90.71K	139.76	4
7	Baking Goods	648	81.89K	126.38	4
8	Health and Hygiene	520	68.03K	130.82	4
9	Soft Drinks	445	58.51K	131.49	3.9
10	Meat	425	59.45K	139.88	4
11	Breads	251	35.38K	140.95	3.9
12	Hard Drinks	214	29.33K	137.08	3.9
13	Others	169	22.45K	132.85	4
14	Starchy Foods	148	21.88K	147.84	3.9
15	Breakfast	110	15.6K	141.79	3.9
16	Seafood	64	9.08K	141.84	4

14. Top 5 Item_Type by count

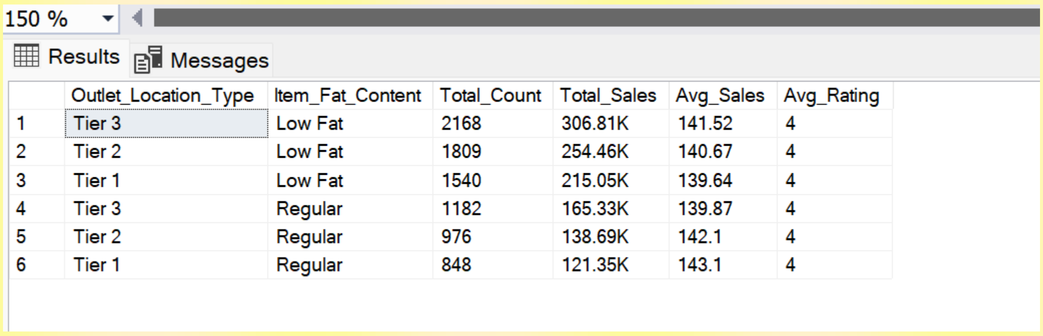
```
SELECT Top 5
  Item_Type,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000,2),'K')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),1) as Avg_Rating
FROM blinkit_data
GROUP BY Item_Type
ORDER BY Total_Count desc;
```



	Item_Type	Total_Count	Total_Sales	Avg_Sales	Avg_Rating
1	Fruits and Vegetables	1232	178.12K	144.58	4
2	Snack Foods	1200	175.43K	146.19	3.9
3	Household	910	135.98K	149.42	4
4	Frozen Foods	856	118.56K	138.5	4
5	Dairy	682	101.28K	148.5	4

15.Fat content by Outlet Type

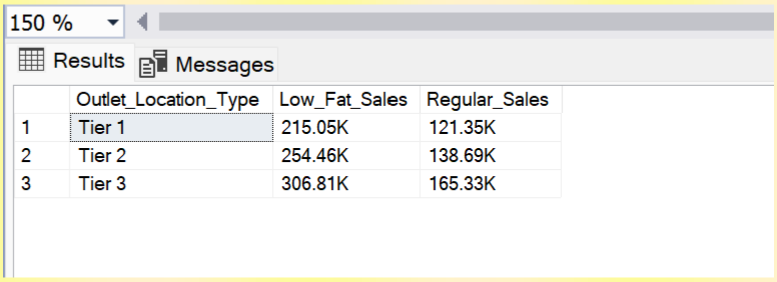
```
SELECT
  Outlet_Location_Type,Item_Fat_Content,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000,2),'K')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),1) as Avg_Rating
FROM blinkit_data
GROUP BY Outlet_Location_Type,Item_Fat_Content
ORDER BY Total_Sales desc;
```



	Outlet_Location_Type	Item_Fat_Content	Total_Count	Total_Sales	Avg_Sales	Avg_Rating
1	Tier 3	Low Fat	2168	306.81K	141.52	4
2	Tier 2	Low Fat	1809	254.46K	140.67	4
3	Tier 1	Low Fat	1540	215.05K	139.64	4
4	Tier 3	Regular	1182	165.33K	139.87	4
5	Tier 2	Regular	976	138.69K	142.1	4
6	Tier 1	Regular	848	121.35K	143.1	4

<u>Just beautifying it</u>

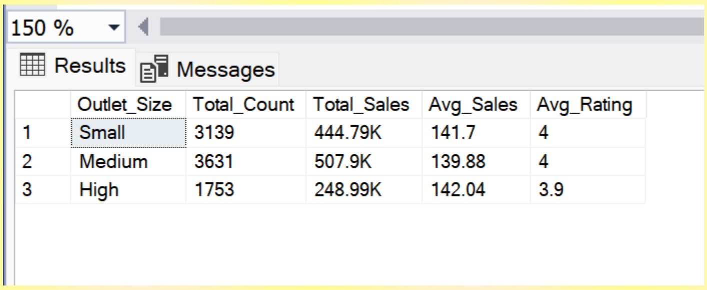
```
SELECT
  Outlet_Location_Type,

  CONCAT(ROUND(SUM(CASE
    WHEN Item_Fat_Content='Low Fat' THEN Sales
    ELSE 0 END) / 1000, 2), 'K') AS Low_Fat_Sales,

  CONCAT(ROUND(SUM(CASE
    WHEN Item_Fat_Content = 'regular' THEN Sales
    ELSE 0 END) / 1000, 2), 'K') AS Regular_Sales

FROM blinkit_data
WHERE Outlet_Location_Type IN ('Tier 1', 'Tier 2', 'Tier 3')
GROUP BY Outlet_Location_Type
ORDER BY Outlet_Location_Type;
```



	Outlet_Location_Type	Low_Fat_Sales	Regular_Sales
1	Tier 1	215.05K	121.35K
2	Tier 2	254.46K	138.69K
3	Tier 3	306.81K	165.33K

16. Data Variation by Outlet Size

```
SELECT
  Outlet_Size,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000,2),'K')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),1) as Avg_Rating
FROM blinkit_data
GROUP BY Outlet_Size
ORDER BY Outlet_Size desc;
```

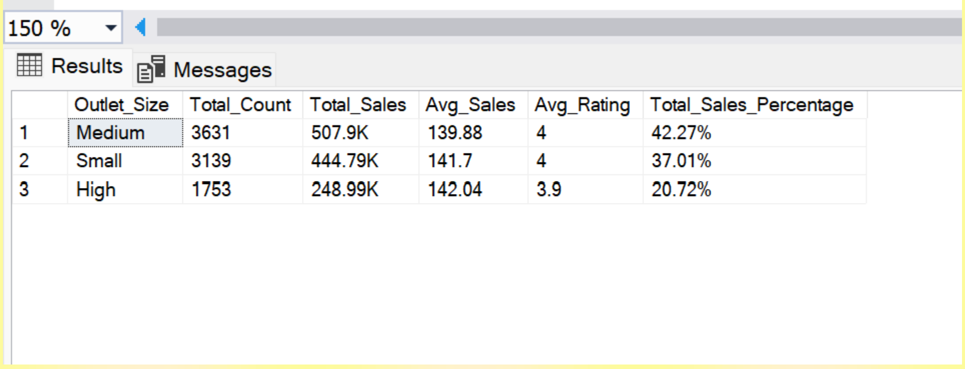


	Outlet_Size	Total_Count	Total_Sales	Avg_Sales	Avg_Rating
1	Small	3139	444.79K	141.7	4
2	Medium	3631	507.9K	139.88	4
3	High	1753	248.99K	142.04	3.9

Another Way of Doing It:

```
SELECT
  Outlet_Size,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000,2),'K')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),1) as Avg_Rating,
  concat(ROUND(SUM(Sales)*100/sum(sum(Sales)) over(),2),'%') AS
Total_Sales_Percentage
FROM blinkit_data
```

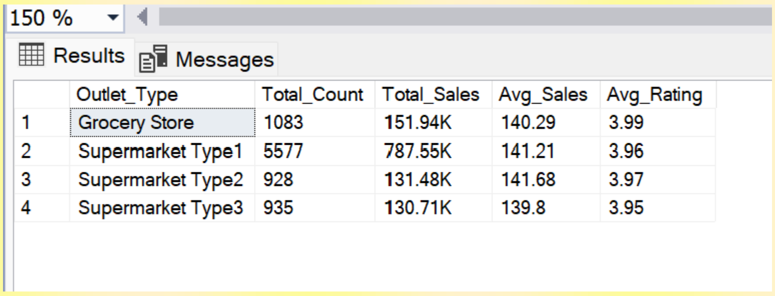
```
GROUP BY Outlet_Size
ORDER BY Total_Sales desc;
```



	Outlet_Size	Total_Count	Total_Sales	Avg_Sales	Avg_Rating	Total_Sales_Percentage
1	Medium	3631	507.9K	139.88	4	42.27%
2	Small	3139	444.79K	141.7	4	37.01%
3	High	1753	248.99K	142.04	3.9	20.72%

17.Data Variation by Outlet Type

```
SELECT
  Outlet_Type,
  count(*) as Total_Count,
  concat(ROUND(sum(Sales)/1000,2),'K')as Total_Sales,
  Round(Avg(Sales),2) AS Avg_Sales,
  Round(AVG(Rating),2) as Avg_Rating
FROM blinkit_data
GROUP BY Outlet_Type
ORDER BY Outlet_Type ;
```



	Outlet_Type	Total_Count	Total_Sales	Avg_Sales	Avg_Rating
1	Grocery Store	1083	151.94K	140.29	3.99
2	Supermarket Type1	5577	787.55K	141.21	3.96
3	Supermarket Type2	928	131.48K	141.68	3.97
4	Supermarket Type3	935	130.71K	139.8	3.95

THANKING YOU