

```
--Displaying data--
```

```
Select * from blinkit_grocery_data
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'blinkitdb' database selected. The right pane shows the 'SQLQuery1.sql' window with the query 'SELECT * FROM blinkit_grocery_data'. Below the query window, the 'Results' pane displays a table with 10 rows and 10 columns. The status bar at the bottom indicates 'Query executed successfully.' and 'DESKTOP-VL7MPET (16.0 RTM) DESKTOP-VL7MPET\Asus (74) blinkitdb 00:00:00 8,523 rows'.

	Item_Fat_Content	Item_Identifier	Item_Type	Outlet_Establishment_Year	Outlet_Identifier	Outlet_Location_Type	Outlet_Size	Outlet_Type	Item_Visibility	Item_Outlet_Identifier
1	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100013501942158	1
2	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.00859605055302382	1
3	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket Type1	0.0258964858949184	1
4	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket Type1	0.0422778688371181	1
5	Low Fat	DR125	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970195800066	1
6	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.00550548080354929	8
7	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.0983124226331711	1
8	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.0269037131220102	1
9	Low Fat	FDW20	Fruits and Vegetables	2000	OUT013	Tier 3	High	Supermarket Type1	0.024129331111908	2
10	Low Fat	FDX25	Canned	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.101561568677425	N

```
/* Cleaning the data (Data Preprocessing for the analysis)*/
```

```
-- Change the name of Item_Fat_Content values using UPDATE--
```

```
update 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
```

```
-- Checking the distinct values of Item_Fat_Content after updating--
```

```
select distinct(Item_Fat_Content) from blinkit_grocery_data
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'blinkitdb' database selected. The right pane shows the 'SQLQuery1.sql' window with the query 'update 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'. Below the query window, the 'Results' pane displays a table with 2 rows and 1 column. The status bar at the bottom indicates 'Query executed successfully.' and 'DESKTOP-VL7MPET (16.0 RTM) DESKTOP-VL7MPET\Asus (74) blinkitdb 00:00:00 2 rows'.

	Item_Fat_Content
1	Low Fat
2	Regular

```
-- Querying for KPI's --
```

```
-- 1. Finding totat sale--
```

```
select cast(cast(sum(Total_Sales) / 1000000 as decimal(10,2)) as varchar(20)) +
```

```
'M' as [Total Sales in Millions]
```

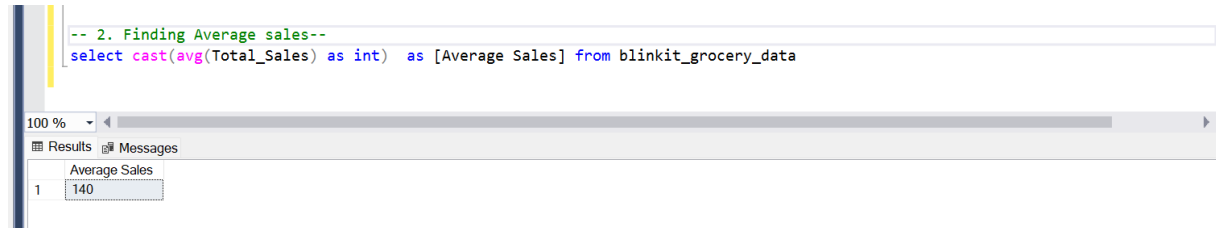
```
from blinkit_grocery_data
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'blinkitdb' database selected. The right pane shows the 'SQLQuery1.sql' window with the query 'select cast(cast(sum(Total_Sales) / 1000000 as decimal(10,2)) as varchar(20)) + 'M' as [Total Sales in Millions] from blinkit_grocery_data;'. Below the query window, the 'Results' pane displays a table with 1 row and 1 column. The status bar at the bottom indicates 'Query executed successfully.' and 'DESKTOP-VL7MPET (16.0 RTM) DESKTOP-VL7MPET\Asus (74) blinkitdb 00:00:00 1 rows'.

	Total Sales in Millions
1	1.20M

-- 2. Finding Average sales--

```
select cast(avg(Total_Sales) as int) as [Average Sales] from  
    blinkit_grocery_data
```

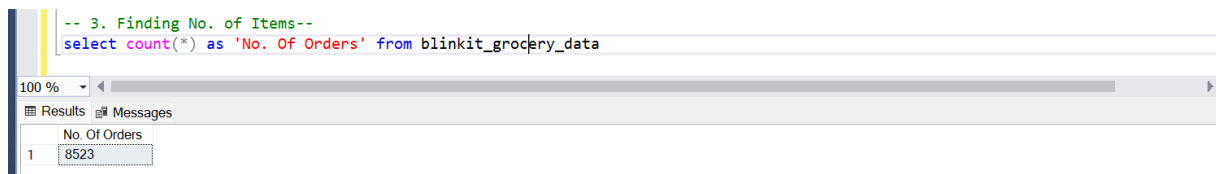


The screenshot shows a SQL query editor with the query: `-- 2. Finding Average sales--
select cast(avg(Total_Sales) as int) as [Average Sales] from blinkit_grocery_data`. Below the editor, the 'Results' tab is active, displaying a single row with the value 140 under the column header 'Average Sales'.

Average Sales
140

-- 3. Finding No. of Items--

```
select count(*) as 'No. Of Orders' from blinkit_grocery_data
```

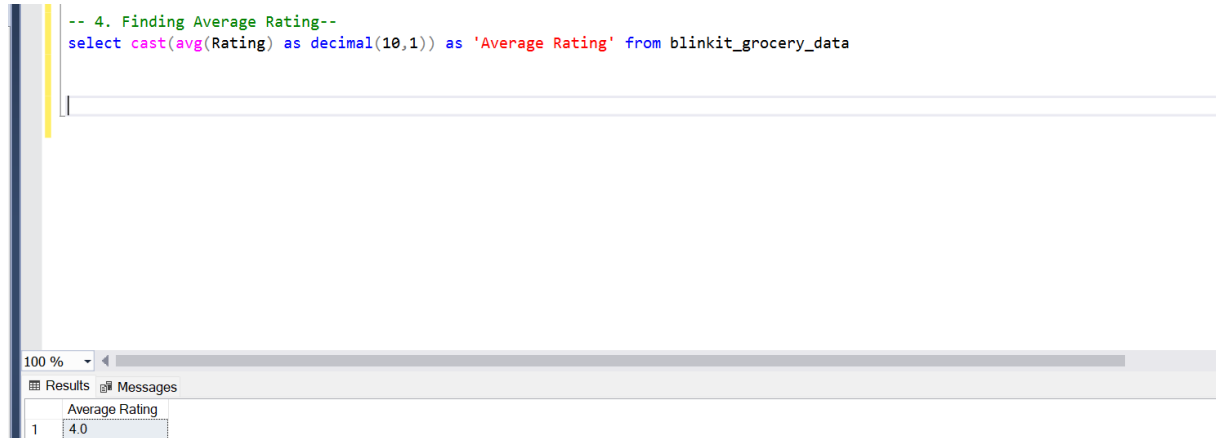


The screenshot shows a SQL query editor with the query: `-- 3. Finding No. of Items--
select count(*) as 'No. Of Orders' from blinkit_grocery_data`. Below the editor, the 'Results' tab is active, displaying a single row with the value 8523 under the column header 'No. Of Orders'.

No. Of Orders
8523

-- 4. Finding Average Rating--

```
select cast(avg(Rating) as decimal(10,1)) as 'Average Rating' from  
    blinkit_grocery_data
```



The screenshot shows a SQL query editor with the query: `-- 4. Finding Average Rating--
select cast(avg(Rating) as decimal(10,1)) as 'Average Rating' from blinkit_grocery_data`. Below the editor, the 'Results' tab is active, displaying a single row with the value 4.0 under the column header 'Average Rating'.

Average Rating
4.0

-- (A) Total Sales by Item_Fat_Content--

```
select Item_Fat_Content, cast(sum(Total_Sales) as decimal(10,2)) as 'Total  
    Sales'  
from blinkit_grocery_data group by Item_Fat_Content
```

```
-- (A) Total Sales by Item_Fat_Content--
select Item_Fat_Content, cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales'
from blinkit_grocery_data group by Item_Fat_Content
```

00 %

Results Messages

	Item_Fat_Content	Total Sales
1	Low Fat	776319.68
2	Regular	425361.80

```
-- (B) Total Sales by Item Type
select Item_Type, cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales'
from blinkit_grocery_data group by Item_Type order by [Total Sales] desc
```

```
-- (B) Total Sales by Item Type
select Item_Type, cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales'
from blinkit_grocery_data group by Item_Type order by [Total Sales] desc
```

100 %

Results Messages

	Item_Type	Total Sales
1	Fruits and Vegetables	178124.08
2	Snack Foods	175433.92
3	Household	135976.53
4	Frozen Foods	118558.88
5	Dairy	101276.46
6	Canned	90706.73
7	Baking Goods	81894.74
8	Health and Hygiene	68025.84
9	Meat	59449.86
10	Soft Drinks	58514.16
11	Breads	35379.12
12	Hard Drinks	29334.68
13	Others	22451.89
14	Starchy Foods	21880.03
15	Breakfast	15596.70
16	Seafood	9077.87

```
-- (C) 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(Total_Sales) as decimal(10,2)) as Total_Sales
from blinkit_grocery_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

```

```

-- (C) 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(Total_Sales) as decimal(10,2)) as Total_Sales
    from blinkit_grocery_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

```

	Outlet_Location_Type	Low_Fat	Regular
1	Tier 1	215047.91	121349.90
2	Tier 2	254464.77	138685.87
3	Tier 3	306806.99	165326.03

```
-- (D) Total Sales by Outlet Establishment--
```

```

select Outlet_Establishment_Year, cast(sum(Total_Sales) as decimal(10,2)) as
       'Total Sales'
from blinkit_grocery_data
group by Outlet_Establishment_Year
order by Outlet_Establishment_Year

```

```

-- E. Total Sales by Outlet Establishment--
select Outlet_Establishment_Year, cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales'
from blinkit_grocery_data
group by Outlet_Establishment_Year
order by Outlet_Establishment_Year

```

	Outlet_Establishment_Year	Total Sales
1	1998	204522.26
2	2000	131809.02
3	2010	132113.37
4	2011	78131.56
5	2012	130476.86
6	2015	130942.78
7	2017	133103.91
8	2020	129103.96
9	2022	131477.77

```
-- (E) Percentage of Sales by Outlet Size--
```

```

select
    Outlet_Size,
    cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales',
    cast((sum(Total_Sales) * 100.0 / sum(sum(Total_Sales)) over()) as decimal
    (10,2)) as 'Sales Percentage'

```

```

from blinkit_grocery_data
group by Outlet_Size
order by [Total Sales] desc

```

```

-- (F) Percentage of Sales by Outlet Size--
select
    Outlet_Size,
    cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales',
    cast((sum(Total_Sales) * 100.0 / sum(sum(Total_Sales)) over())) as decimal(10,2)) as 'Sales Percentage'
from blinkit_grocery_data
group by Outlet_Size
order by [Total Sales] desc

```

00 %

Results Messages

	Outlet_Size	Total Sales	Sales Percentage
1	Medium	507895.73	42.27
2	Small	444794.17	37.01
3	High	248991.58	20.72

```

-- (F) Sales by Outlet Location--
select Outlet_Location_Type, cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales'
from blinkit_grocery_data
group by Outlet_Location_Type
order by [Total Sales] desc

```

```

-- (F) Sales by Outlet Location--
select Outlet_Location_Type, cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales'
from blinkit_grocery_data
group by Outlet_Location_Type
order by [Total Sales] desc

```

100 %

Results Messages

	Outlet_Location_Type	Total Sales
1	Tier 3	472133.03
2	Tier 2	393150.64
3	Tier 1	336397.81

```

-- (G) All metrics by Outlet Type
select Outlet_Type,
    cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales',
    cast(avg(Total_Sales) as decimal(10,2)) as 'Average Sales',
    count(*) as 'No. of Items',
    cast(avg(Rating) as decimal(10,2)) as 'Average Rating',
    cast(sum(Item_Visibility) as decimal(10,2)) as 'Item Visibility'
from blinkit_grocery_data
group by Outlet_Type
order by [Total Sales] desc

```

```
-- (G) All metrics by Outlet Type
select Outlet_Type,
       cast(sum(Total_Sales) as decimal(10,2)) as 'Total Sales',
       cast(avg(Total_Sales) as decimal(10,2)) as 'Average Sales',
       count(*) as 'No. of Items',
       cast(avg(Rating) as decimal(10,2)) as 'Average Rating',
       cast(sum(Item_Visibility) as decimal(10,2)) as 'Item Visibility'
from blinkit_grocery_data
group by Outlet_Type
order by [Total Sales] desc
```

100 %

Results Messages

	Outlet_Type	Total Sales	Average Sales	No. of Items	Average Rating	Item Visibility
1	Supermarket Type1	787549.89	141.21	5577	3.96	338.65
2	Grocery Store	151939.15	140.29	1083	3.99	113.57
3	Supermarket Type2	131477.77	141.68	928	3.97	56.62
4	Supermarket Type3	130714.67	139.80	935	3.95	54.80