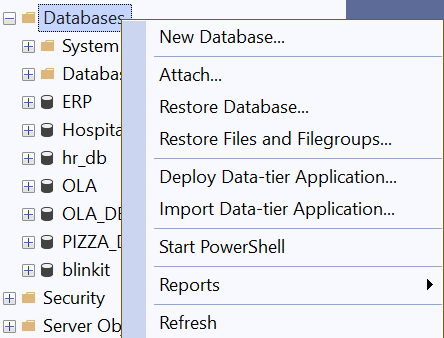
* **Extract File into SSMS:**

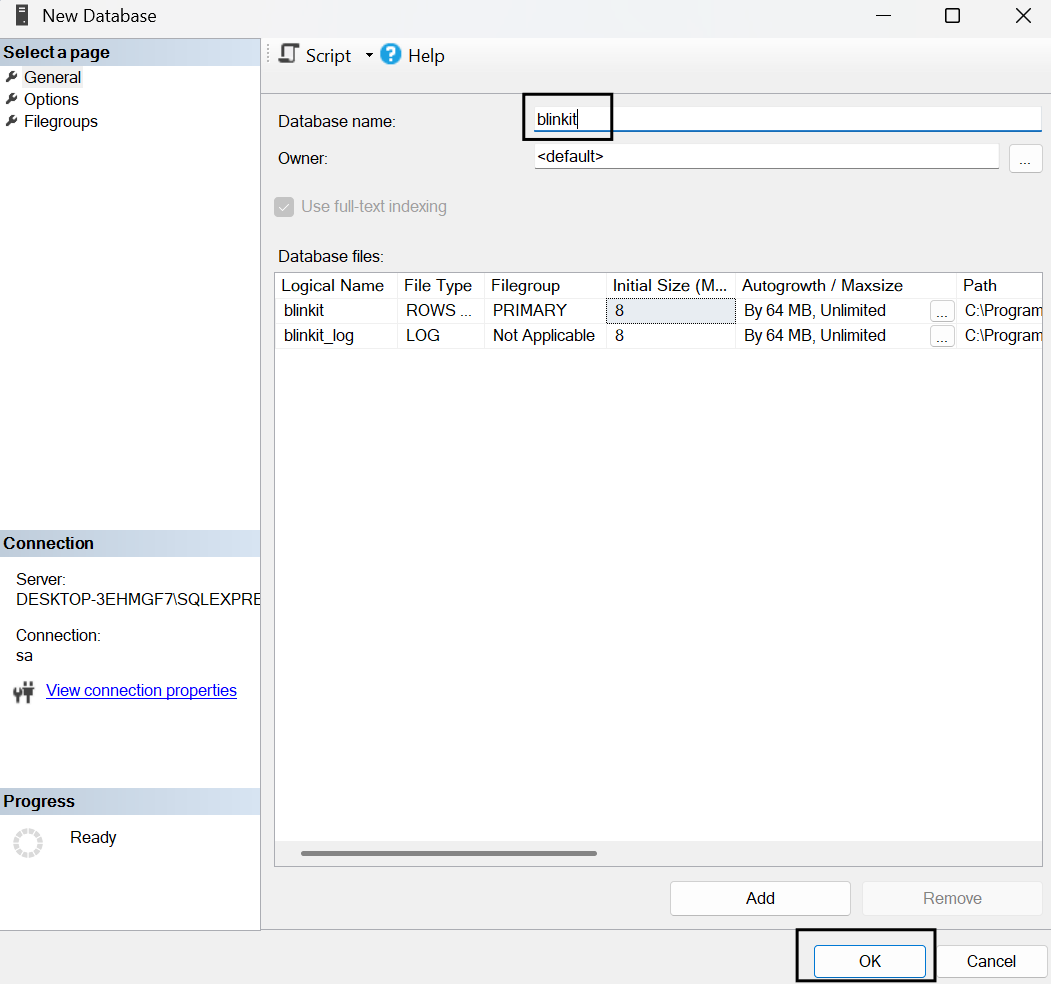
1. Open the SQL Server Management Studio (SSMS 2021) and connect to the server.

* ***Note*** *– You can open any SQL Server (MySQL, PostgreSQL, Oracle, etc.) and connect to your server.*

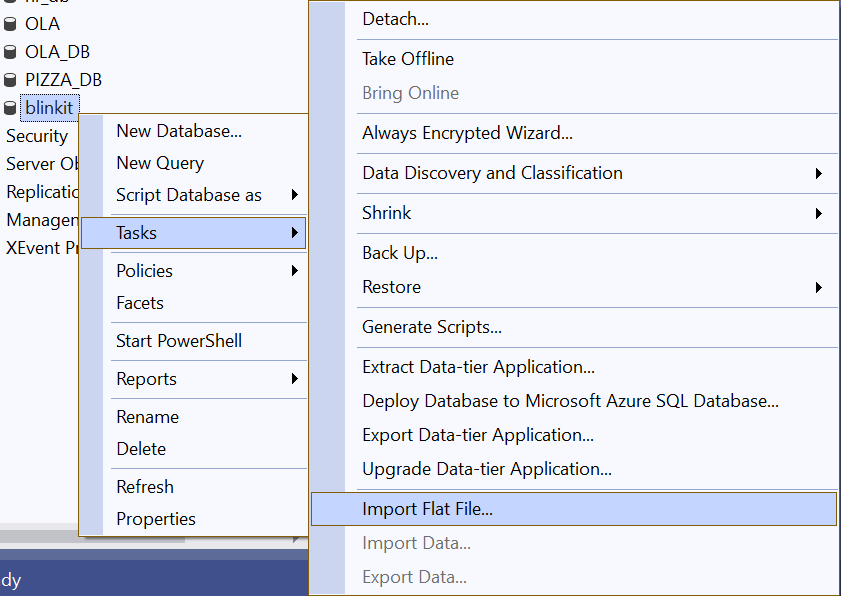
1. Go to the Databases on the left side and click right and select new Database.



1. Enter the database name as blinkit and click on Ok.



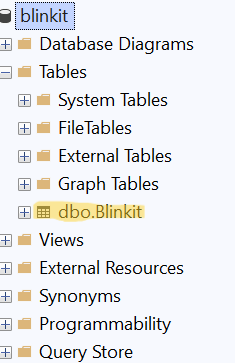
1. Select blinkit database and click right and select Task → Import Flat File…



1. The Blinkit.csv file was imported into SQL Server using the Import Flat File Wizard. The file path was selected, the table was named **Blinkit** under the **dbo** schema, and data types were auto-detected. After previewing and confirming the structure, the import was completed by clicking **Finish**, creating the table in our database.

(***Note*** *- The extracted file should be in .csv/.txt.)*

1. You can check that the table has been added to the Blinkit database and, if the table is not there, then refresh the database and then check.

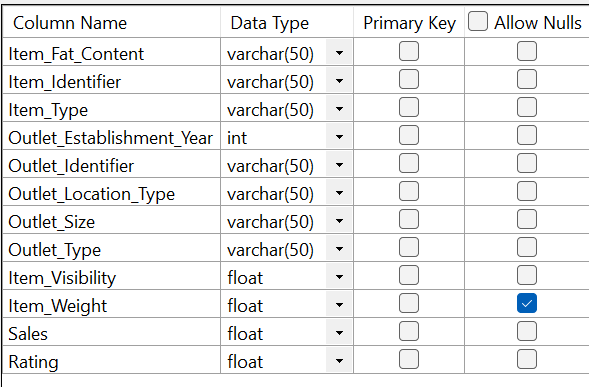


* **CSV data download or show:**

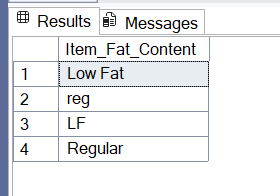


* **Data Transformation in SQL:**

1. While import the CSV file, data types were modified as per the data structure: text columns like Item\_Fat\_Content, Item\_Identifier, and Item\_Type were set to varchar, numeric columns like Item\_Visibility, Item\_Weight, Sales, and Rating were set to float, and Outlet\_Establishment\_Year was assigned as int. Only the Item\_Weight column was allowed to have NULL values.



1. Check the table in the database. There are 3 different values for Item\_Fat\_Content, like LF, low fat and reg, so it has to be changed to LF, low fat → ‘Low Fat’ & reg → ‘Regular’.



1. Update the table –

UPDATE Blinkit

SET Item\_Fat\_Content =

CASE

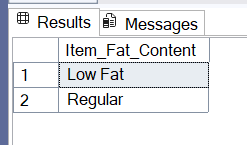
WHEN Item\_Fat\_Content IN ('LF', 'low fat') THEN 'Low Fat'

WHEN Item\_Fat\_Content = 'reg' THEN 'Regular'

ELSE Item\_Fat\_Content

END;

Output –



* **Requirements:**
  + 1. **KPI’s:**

1. **Total Sales:** 
   * + - * The overall revenue generated from all items sold.

***Input –***

select

Concat (

cast (

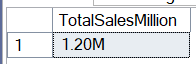
SUM(Sales)/1000000

as decimal (10,2)),

'M') TotalSalesMillion

from blinkit;

***Output –***



1. **Average Sales:**
   * + - * The average revenue per sale.

***Input –***

select

round (

AVG(Sales),0

) AS AvgSales

from blinkit;

***Output* –**



1. **No of Items:**
   * + - * The total count of different items sold.

***Input –***

select

Count (\*) as No\_of\_ITEMS

from blinkit;

***Output* –**



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

***Input –***

select

round (

avg(Rating), 2

) as AvegRating

from blinkit;

***Output* –**



* + 1. **Granular Requirements:**

1. **Total Sales by Fat Content:**

***Input* –**

select

Item\_Fat\_Content,

Concat (

ROUND (

SUM(Sales)/1000 ,2)

),

'K') as TotalSales

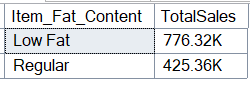
from

blinkit

group by

Item\_Fat\_Content;

***Output* –**

****

1. **Total Sales by Item Type:**

***Input* –**

select Top 5

Item\_Type,

round(SUM(Sales), 2) as TotalSales

from

blinkit

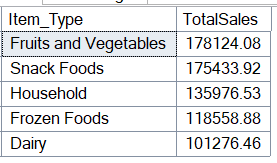
group by

Item\_Type

order by

TotalSales DESC;

***Output* –**

****

1. **Total sales by Item Fat Content and Outlet Location Type**

***Input* –**

select

Item\_Fat\_Content,

Outlet\_Location\_Type,

round(SUM(Sales), 2) AS TotalSales

from

blinkit

group by

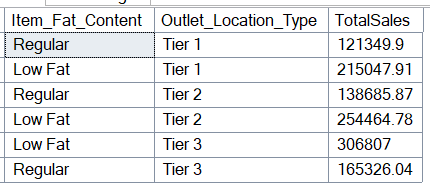
Item\_Fat\_Content,

Outlet\_Location\_Type

order by

Outlet\_Location\_Type;

***Output* –**

******

1. **Fat Content by Outlet for Total Sales:**

***Input* –**

select Outlet\_Location\_Type,

Coalesce ([Low Fat], 0) as Low\_Fat,

Coalesce ([Regular], 0) as Regular

from

(select

Item\_Fat\_Content,

Outlet\_Location\_Type,

round(SUM(Sales), 2) AS TotalSales

from

blinkit

group by

Item\_Fat\_Content,

Outlet\_Location\_Type

) AS SourceTable

PIVOT

(

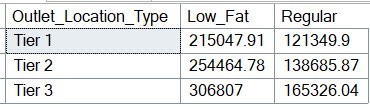
Sum(TotalSales)

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

) AS PivotTable

Order by Outlet\_Location\_Type;

***Output* –**

****

1. **Outlet by Fat Content for Total Sales:**

***Input –***

select

Item\_Fat\_Content,

Coalesce ([Tier 1], 0) as Tier\_1,

Coalesce ([Tier 2], 0) as Tier\_2,

Coalesce ([Tier 3], 0) as Tier\_3

From

(select

Item\_Fat\_Content,

Outlet\_Location\_Type,

round(SUM(Sales), 2) AS TotalSales

from

blinkit

group by

Item\_Fat\_Content,

Outlet\_Location\_Type) as SourceTable

PIVOT

(

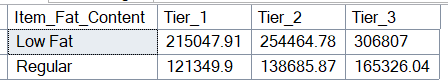
Sum(TotalSales)

For Outlet\_Location\_Type IN ([Tier 1], [Tier 2], [Tier 3])

) AS PivotTable

order by Item\_Fat\_Content;

***Output –***

****

1. **Total Sales by Outlet Establishment (Top 5):**

***Input –***

select TOP 5

Outlet\_Establishment\_Year,

round(SUM(Sales), 2) AS TotalSales

from

blinkit

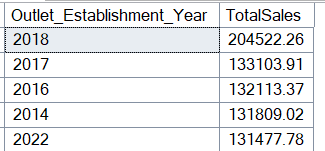
group by

Outlet\_Establishment\_Year

order by

TotalSales desc;

***Output –***



1. **Total Sales by Outlet Establishment (Bottom 5):**

***Input –***

select TOP 5

Outlet\_Establishment\_Year,

round(SUM(Sales), 2) AS TotalSales

from

blinkit

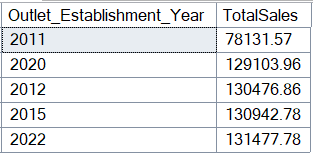
group by

Outlet\_Establishment\_Year

order by

TotalSaleS;

***Output –***



* + 1. **Charts Requirements:**

**Percentage of Sales by Outlet Size:**

***Input –***

select

Outlet\_Size,

round(SUM(sales),2) as TotalSales,

round((SUM(Sales) \* 100.0 / SUM(SUM(Sales))

OVER ()), 2) as PerSALES

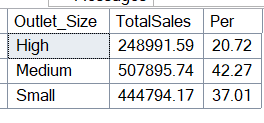
from

blinkit

group by

Outlet\_Size

***Output –***

****

**Sales by Outlet Location:**

***Input –***

select

Outlet\_Location\_Type,

ROUND(SUM(Sales),2) as TotalSales

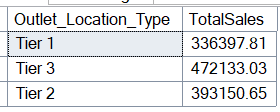
from

blinkit

group by

Outlet\_Location\_Type

***Output*** –

******

**All Metrics by Outlet Type:**

***Input –***

select

Outlet\_Type,

ROUND(SUM(Sales),2) as TotalSales,

round((SUM(Sales) \* 100.0 / SUM(SUM(Sales))

OVER ()), 0) as PerSales,

round(AVG(Sales),0) AvgSales,

Count (\*) AS No\_of\_items,

round(avg(Rating), 2) as AvegRating,

ROUND(max(Sales),2) as MaximumSales,

ROUND(min(Sales),2) as MinimumSales

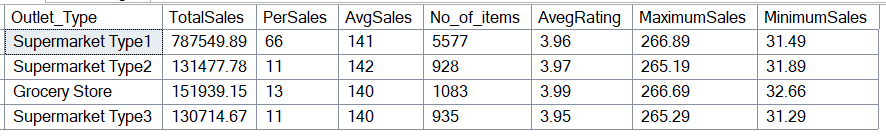
from

blinkit

group by

Outlet\_Type

***Output –***

******

* **Required All Data (SQL View):**

1. **KPI’s:**
2. **Total Sales:** The overall revenue generated from all items sold.

* **select \* from vw\_totalsales;**

1. **Average Sales:** The average revenue per sale.

* **select \* from vw\_AVGsales;**

1. **No of Items:** The total count of different items sold.

* **select \* from vw\_noofitems;**

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

* **select \* from vw\_AVGrating;**

1. **Granular Requirements:**
2. **Total Sales by Fat Content:**

* select \* from vw\_Sales\_FC;

1. **Total Sales by Item Type:**

* select \* from vw\_Sales\_it;

1. **Total sales by Item Fat Content and Outlet Location Type**

* select \* from vw\_Sales\_OLT;

1. **Fat Content by Outlet for Total Sales:**

* select \* from vw\_Sales\_FCO

Order by Outlet\_Location\_Type;

1. **Outlet by Fat Content for Total Sales:**

* select \* from vw\_Sales\_OFC

order by Item\_Fat\_Content;

1. **Total Sales by Outlet Establishment (Top 5):**

* select \* from vw\_Sales\_Oet5;

1. **Total Sales by Outlet Establishment (Bottom 5):**

* select \* from vw\_Sales\_Oeb5;

1. **Charts Requirements:**
2. **Percentage of Sales by Outlet Size:**

* select \* from vw\_PerSales\_OS;

1. **Sales by Outlet Location:**

* select \* from vw\_Sales\_OTL;

1. **All Metrics by Outlet Type:**

* select \* from vw\_Sales\_OT;