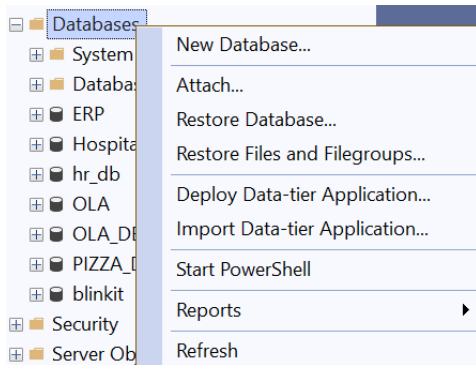
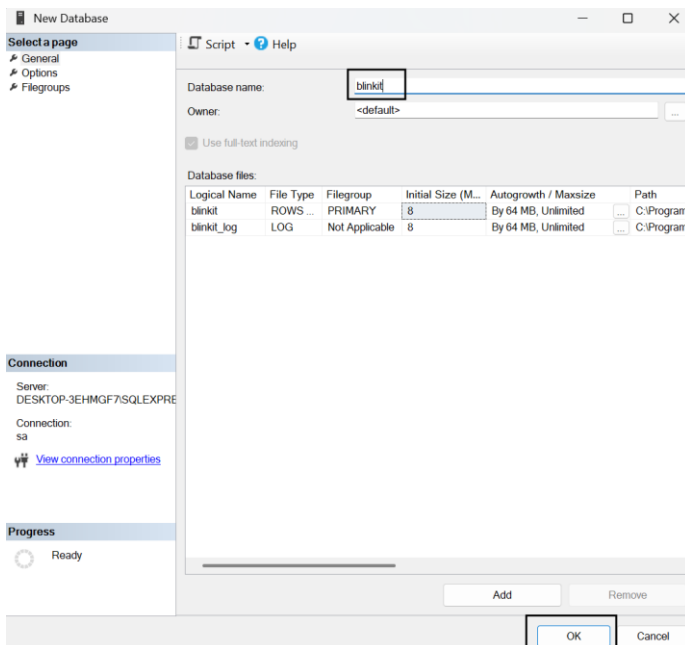


## ❖ Extract File into SSMS:

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

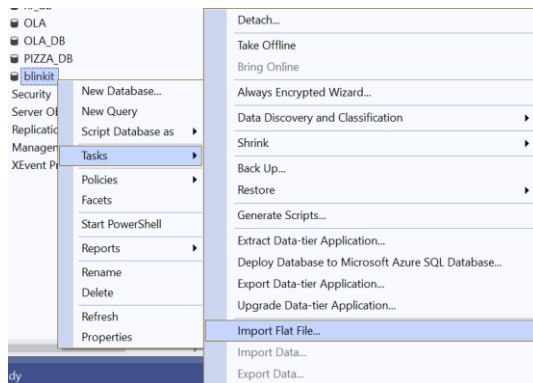


3. Enter the database name as blinkit and click on Ok.



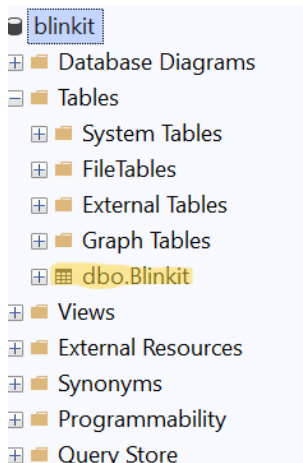
# Blinkit Analysis

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

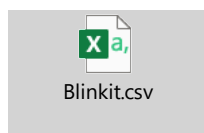


5. The Blinkit.csv file was imported into SQL Server using the Import Flat File Wizard. The file path was selected, the table was named **Blinkit** under the **dbo** schema, and data types were auto-detected. After previewing and confirming the structure, the import was completed by clicking **Finish**,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

6. 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:

- 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.

Column Name	Data Type	Primary Key	<input type="checkbox"/> Allow Nulls
Item_Fat_Content	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Item_Identifier	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Item_Type	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Outlet_Establishment_Year	int	<input type="checkbox"/>	<input type="checkbox"/>
Outlet_Identifier	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Outlet_Location_Type	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Outlet_Size	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Outlet_Type	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Item_Visibility	float	<input type="checkbox"/>	<input type="checkbox"/>
Item_Weight	float	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Sales	float	<input type="checkbox"/>	<input type="checkbox"/>
Rating	float	<input type="checkbox"/>	<input type="checkbox"/>

- 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'.

Results Messages	
	Item_Fat_Content
1	Low Fat
2	reg
3	LF
4	Regular

- 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 –

Results Messages	
	Item_Fat_Content
1	Low Fat
2	Regular

## ❖ Requirements:

### A. 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 –**

	TotalSalesMillion
1	1.20M

#### 2. Average Sales:

- The average revenue per sale.

**Input –**

```
SELECT
    ROUND (
        AVG(SALES),0
    ) AS AVGSALES
FROM BLINKIT;
```

**Output –**

	AvgSales
1	141

#### 3. No of Items:

- The total count of different items sold.

**Input –**

```
SELECT
    COUNT (*) AS NO_OF_ITEMS
FROM BLINKIT;
```

**Output –**

	No_of_Items
1	8523

#### 4. Average Rating:

- The average customer rating for items sold.

**Input –**

```
SELECT
    ROUND (
        AVG(RATING), 2
    ) AS AVEGRATING
FROM BLINKIT;
```

**Output –**

	AvegRating
1	3.97

## B. 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 –*

Item_Fat_Content	TotalSales
Low Fat	776.32K
Regular	425.36K

### 2. 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 –*

Item_Type	TotalSales
Fruits and Vegetables	178124.08
Snack Foods	175433.92
Household	135976.53
Frozen Foods	118558.88
Dairy	101276.46

### 3. 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 –*

Item_Fat_Content	Outlet_Location_Type	TotalSales
Regular	Tier 1	121349.9
Low Fat	Tier 1	215047.91
Regular	Tier 2	138685.87
Low Fat	Tier 2	254464.78
Low Fat	Tier 3	306807
Regular	Tier 3	165326.04

## 4. 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 –**

Outlet_Location_Type	Low_Fat	Regular
Tier 1	215047.91	121349.9
Tier 2	254464.78	138685.87
Tier 3	306807	165326.04

## 5. 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 –**

Item_Fat_Content	Tier_1	Tier_2	Tier_3
Low Fat	215047.91	254464.78	306807
Regular	121349.9	138685.87	165326.04

## 6. 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 –**

Outlet_Establishment_Year	TotalSales
2018	204522.26
2017	133103.91
2016	132113.37
2014	131809.02
2022	131477.78

## 7. 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 –**

Outlet_Establishment_Year	TotalSales
2011	78131.57
2020	129103.96
2012	130476.86
2015	130942.78
2022	131477.78

## C. Charts Requirements:

### 1. 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 –**

Outlet_Size	TotalSales	Per
High	248991.59	20.72
Medium	507895.74	42.27
Small	444794.17	37.01

### 2. Sales by Outlet Location:

**Input –**

```
SELECT
    OUTLET_LOCATION_TYPE,
    ROUND(SUM(SALES),2) AS TOTALSALES
FROM
    BLINKIT
GROUP BY
    OUTLET_LOCATION_TYPE
```

**Output –**

Outlet_Location_Type	TotalSales
Tier 1	336397.81
Tier 3	472133.03
Tier 2	393150.65

### 3. 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 –**

Outlet_Type	TotalSales	PerSales	AvgSales	No_of_items	AvegRating	MaximumSales	MinimumSales
Supermarket Type1	787549.89	66	141	5577	3.96	266.89	31.49
Supermarket Type2	131477.78	11	142	928	3.97	265.19	31.89
Grocery Store	151939.15	13	140	1083	3.99	266.69	32.66
Supermarket Type3	130714.67	11	140	935	3.95	265.29	31.29



## ❖ Required All Data (SQL View):

### A. KPI's:

1. **Total Sales:** The overall revenue generated from all items sold.  
→ `SELECT * FROM VW_TOTALSALES;`
2. **Average Sales:** The average revenue per sale.  
→ `SELECT * FROM VW_AVGSALES;`
3. **No of Items:** The total count of different items sold.  
→ `SELECT * FROM VW_NOOFITEMS;`
4. **Average Rating:** The average customer rating for items sold.  
→ `SELECT * FROM VW_AVGRATING;`

### B. Granular Requirements:

1. **Total Sales by Fat Content:**  
→ `SELECT * FROM VW_SALES_FC;`
2. **Total Sales by Item Type:**  
→ `SELECT * FROM VW_SALES_IT;`
3. **Total sales by Item Fat Content and Outlet Location Type**  
→ `SELECT * FROM VW_SALES_OLT;`
4. **Fat Content by Outlet for Total Sales:**  
→ `SELECT * FROM VW_SALES_FCO  
ORDER BY OUTLET_LOCATION_TYPE;`
5. **Outlet by Fat Content for Total Sales:**  
→ `SELECT * FROM VW_SALES_OFCL  
ORDER BY ITEM_FAT_CONTENT;`
6. **Total Sales by Outlet Establishment (Top 5):**  
→ `SELECT * FROM VW_SALES_OET5;`
7. **Total Sales by Outlet Establishment (Bottom 5):**  
→ `SELECT * FROM VW_SALES_OEB5;`

### C. Charts Requirements:

1. **Percentage of Sales by Outlet Size:**  
→ `SELECT * FROM VW_PERSALES_OS;`
2. **Sales by Outlet Location:**  
→ `SELECT * FROM VW_SALES_OTL;`
3. **All Metrics by Outlet Type:**  
→ `SELECT * FROM VW_SALES_OT;`