SQL Script Explanation for Sales Data Analysis

1. Creating and Using a Database

```
CREATE DATABASE mayank;
USE mayank;
```

- Creates a new database called 'mayank'.
- Switches to using this database for further operations.

2. Creating the SalesData Table

```
CREATE TABLE SalesData (
    Item_Fat_Content VARCHAR(50),
    Item_Identifier VARCHAR(50),
    Item_Type VARCHAR(100),
    Outlet_Establishment_Year INT,
    Outlet_Identifier VARCHAR(50),
    Outlet_Location_Type VARCHAR(100),
    Outlet_Size VARCHAR(50),
    Outlet_Type VARCHAR(100),
    Item_Visibility DOUBLE,
    Item_Weight DOUBLE DEFAULT NULL,
    Total_Sales DOUBLE,
    Rating INT
);
```

- Creates a table named 'SalesData' to store product sales information.
- Key columns include Item_Fat_Content, Item_Identifier, Total_Sales, and Rating.

3. Importing Data from a CSV File

```
LOAD DATA INFILE 'C:/.../BlinkIT Grocery Data.csv'
INTO TABLE SalesData
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES (...)
SET Item_Weight = CASE
    WHEN @Item_Weight REGEXP '^[0-9.]+$' THEN @Item_Weight
    ELSE NULL
END;
```

- Loads data from a CSV file into the 'SalesData' table.
- Ignores the header row and handles missing values in Item_Weight.

4. Cleaning and Normalizing Data

```
UPDATE SalesData
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;
```

- Standardizes values in 'Item_Fat_Content'.

5. Total Sales & Average Sales Calculation

```
SELECT CAST(SUM(Total_Sales)/1000000 AS DECIMAL(10,2)) AS total_sales_millions FROM SalesData;
```

- Finds total sales in millions.
- Other queries calculate avg sales and count total items.

6. Sales Analysis by Year (2022)

```
SELECT CAST(SUM(Total_Sales)/1000000 AS DECIMAL(10,2)) AS total_sales_millions
FROM SalesData WHERE Outlet_Establishment_Year = 2022;
```

- Filters total and avg sales for 2022 outlets.

7. Grouped Analysis by Item Fat Content

```
SELECT Item_Fat_Content, CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS total_sales FROM SalesData WHERE Outlet_Establishment_Year = 2020 GROUP BY Item_Fat_Content ORDER BY total_sales DESC;
```

- Groups sales data by Item_Fat_Content for outlets established in 2020.

8. Pivot Analysis by Outlet Location Type

```
SELECT Outlet_Location_Type, CAST(SUM(CASE WHEN Item_Fat_Content = 'Low Fat' THEN Total_Sales ELSE 0 END) AS DECIMAL(10,2)) AS Low_Fat FROM SalesData GROUP BY Outlet_Location_Type;
```

- Pivot-like query showing sales split by fat content and location type.

9. Sales Contribution by Outlet Size

```
SELECT Outlet_Size, CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS Total_Sales FROM SalesData GROUP BY Outlet_Size ORDER BY Total_Sales DESC;
```

- Calculates sales distribution by outlet size.

10. Final Sales & Rating Report by Outlet Type

```
SELECT Outlet_Type, CAST(SUM(Total_Sales) AS DECIMAL(10,2)) AS total_sales FROM SalesData GROUP BY Outlet_Type ORDER BY total_sales DESC;
```

- Summarizes sales and ratings by outlet type.