Blinkit Sales Analysis — SQL Project

Project Overview

This project analyzes the Blinkit grocery sales dataset. Goals: - Clean & standardize the dataset - Optimize data types for performance and accuracy - Compute KPIs and perform EDA - Produce insights for business decisions

---

1. Data Cleaning & Preparation

1.1 Rename incorrect column

```sql
ALTER TABLE blinkit_data
RENAME COLUMN `item_Fat_Content` TO Item_Fat_Content;
```

Explanation: Fix inconsistent casing/typos so downstream queries are reliable.

1.2 Standardize `Item_Fat_Content` values

```sql
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;
```

Explanation: Consolidates synonyms into canonical categories for accurate grouping.

1.3 Convert text → numeric and optimize types (MySQL)

```sql
ALTER TABLE blinkit_data
MODIFY COLUMN Item_Fat_Content ENUM('Low Fat', 'Regular'),
MODIFY COLUMN Item_Identifier VARCHAR(10),
MODIFY COLUMN Item_Type VARCHAR(50),
MODIFY COLUMN Outlet_Establishment_Year YEAR,
MODIFY COLUMN Outlet_Identifier VARCHAR(10),
MODIFY COLUMN Outlet_Location_Type ENUM('Tier 1', 'Tier 2', 'Tier 3'),
MODIFY COLUMN Outlet_Size ENUM('Small', 'Medium', 'High'),
MODIFY COLUMN Outlet_Type VARCHAR(50),
MODIFY COLUMN Item_Visibility DECIMAL(5,4),
MODIFY COLUMN Item_Weight DECIMAL(6,2),
MODIFY COLUMN Total_Sales DECIMAL(14,2),
MODIFY COLUMN Rating DECIMAL(3,1);
```

Explanation: Use appropriate types to save space and enable numeric aggregations.

---

2. Core KPIs (SQL)

2.1 Total Sales (in millions)

```sql
SELECT CAST(SUM(Total_Sales) / 1000000.0 AS DECIMAL(10,2)) AS
Total_Sales_Million
FROM blinkit_data;
```

Explanation: Shows total revenue scaled to millions for readable reporting.

2.2 Average Sales per Record

```sql
SELECT CAST(AVG(Total_Sales) AS DECIMAL(10,0)) AS Avg_Sales
FROM blinkit_data;
```

2.3 Number of Orders / Records

```sql
SELECT COUNT(*) AS No_of_Orders
FROM blinkit_data;
```

2.4 Average Rating

```sql
SELECT CAST(AVG(Rating) AS DECIMAL(10,1)) AS Avg_Rating
FROM blinkit_data;
```

---

3. Exploratory Data Analysis Queries

3.1 Total Sales by Fat Content

```sql
SELECT Item_Fat_Content, CAST(SUM(Total_Sales) AS DECIMAL(14,2)) AS
Total_Sales
FROM blinkit_data
GROUP BY Item_Fat_Content
ORDER BY Total_Sales DESC;
```

Insight: Identify which fat-content category drives revenue.

3.2 Total Sales by Item Type

```sql
SELECT Item_Type, CAST(SUM(Total_Sales) AS DECIMAL(14,2)) AS Total_Sales
FROM blinkit_data
GROUP BY Item_Type
ORDER BY Total_Sales DESC;
```

3.3 Fat Content Sales by Outlet Location

```sql
SELECT
    Outlet_Location_Type,
    SUM(CASE WHEN Item_Fat_Content = 'Low Fat' THEN Total_Sales ELSE 0 END)
AS Low_Fat,
```

```
    SUM(CASE WHEN Item_Fat_Content = 'Regular' THEN Total_Sales ELSE 0 END)
AS Regular
FROM blinkit_data
GROUP BY Outlet_Location_Type
ORDER BY Outlet_Location_Type;
```

Insight: Shows preference differences across Tier 1/2/3.

3.4 Sales by Outlet Establishment Year

```
SELECT Outlet_Establishment_Year, CAST(SUM(Total_Sales) AS DECIMAL(14,2)) AS
Total_Sales
FROM blinkit_data
GROUP BY Outlet_Establishment_Year
ORDER BY Outlet_Establishment_Year;
```

3.5 Sales Contribution (%) by Outlet Size

```
SELECT
    Outlet_Size,
    ROUND(SUM(Total_Sales),2) AS Total_Sales,
    ROUND((SUM(Total_Sales) * 100.0 / SUM(SUM(Total_Sales)) OVER()), 2) AS
Sales_Percentage
FROM blinkit_data
GROUP BY Outlet_Size
ORDER BY Total_Sales DESC;
```

Note: `SUM(SUM(...)) OVER()` is a SQL-standard window expression supported by MySQL 8+.

3.6 Total Sales by Outlet Location Type

```
SELECT Outlet_Location_Type, CAST(SUM(Total_Sales) AS DECIMAL(14,2)) AS
Total_Sales
FROM blinkit_data
GROUP BY Outlet_Location_Type
ORDER BY Total_Sales DESC;
```

3.7 Full Metrics by Outlet Type

```
SELECT Outlet_Type,
       CAST(SUM(Total_Sales) AS DECIMAL(14,2)) AS Total_Sales,
       CAST(AVG(Total_Sales) AS DECIMAL(10,0)) AS Avg_Sales,
       COUNT(*) AS No_Of_Items,
       CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating,
       CAST(AVG(Item_Visibility) AS DECIMAL(10,4)) AS Avg_Visibility
FROM blinkit_data
GROUP BY Outlet_Type
ORDER BY Total_Sales DESC;
```
