**Blinkit Sales Analysis – Database Documentation**

**1. Introduction**

This document provides a professional overview and detailed documentation of the **Blinkit** database schema, data cleaning procedures, and SQL queries used for various analytical purposes. The database and its associated queries enable insights into sales performance, customer behavior, and outlet analysis based on item characteristics, outlet types, establishment years, and location types.

**2. Database Schema: Blinkit**

**2.1 Table Structure**

The **Blinkit** table stores data related to items, outlets, and sales. Below is the schema for the table:

CREATE TABLE Blinkit (

ItemFatContent VARCHAR(20), -- Describes the fat content of the item (e.g., 'Low Fat', 'Regular')

ItemIdentifier VARCHAR(20), -- Unique identifier for each item

ItemType VARCHAR(50), -- Type of item (e.g., 'Snacks', 'Beverages')

OutletEstablishmentYear INT, -- Year the outlet was established

OutletIdentifier VARCHAR(20), -- Unique identifier for each outlet

OutletLocationType VARCHAR(20), -- Location type of the outlet (e.g., 'Tier 1', 'Tier 2', 'Tier 3')

OutletSize VARCHAR(20), -- Size of the outlet (e.g., 'Small', 'Medium', 'High')

OutletType VARCHAR(30), -- Type of outlet (e.g., 'Grocery Store', 'Supermarket Type1', 'Supermarket Type2', 'Supermarket Type3')

ItemVisibility DECIMAL(10,6), -- Visibility score of the item (from 0 to 1)

ItemWeight DECIMAL(10,2) NULL, -- Weight of the item (nullable)

Sales DECIMAL(10,4), -- Sales revenue for the item

Rating DECIMAL(3,1) -- Rating of the item (on a scale from 1 to 5)

);

**2.2 Column Definitions**

* **ItemFatContent**: Represents the fat content category of the item, such as "Low Fat" or "Regular".
* **ItemIdentifier**: A unique code used to identify each item.
* **ItemType**: Describes the category or type of item, such as "Snacks", "Beverages", etc.
* **OutletEstablishmentYear**: The year when the outlet was established.
* **OutletIdentifier**: A unique identifier for the outlet where the item is sold.
* **OutletLocationType**: Indicates the location tier of the outlet, which can be one of the following:
  + "Tier 1" – The highest-tier location.
  + "Tier 2" – The second-tier location.
  + "Tier 3" – The third-tier location.
* **OutletSize**: Describes the size of the outlet. Possible values include:
  + "Small"
  + "Medium"
  + "High"
* **OutletType**: Represents the type of outlet, which can be one of the following:
  + "Grocery Store"
  + "Supermarket Type1"
  + "Supermarket Type2"
  + "Supermarket Type3"
* **ItemVisibility**: A numerical score indicating the visibility of the item (scale from 0 to 1).
* **ItemWeight**: The weight of the item in kilograms, nullable if the data is missing.
* **Sales**: The total revenue generated from the sale of the item.
* **Rating**: The rating given by customers to the item, on a scale from 1 to 5.

**3. Data Cleaning Procedures**

**3.1 Standardizing Fat Content Data**

Before conducting any analysis, we standardize the ItemFatContent values to ensure consistency. The query below updates the fat content values to more readable formats:

-- Disable SQL Safe Updates to allow updates on the table

SET SQL\_SAFE\_UPDATES = 0;

-- Standardize the ItemFatContent values

UPDATE blinkit

SET

ItemFatContent = CASE

WHEN ItemFatContent IN ('LF', 'low fat') THEN 'Low Fat'

WHEN ItemFatContent = 'reg' THEN 'Regular'

ELSE ItemFatContent

END

WHERE

ItemFatContent IN ('LF', 'low fat', 'reg');

-- Re-enable safe updates

SET SQL\_SAFE\_UPDATES = 1;

This procedure ensures that values like 'LF', 'low fat', and 'reg' are replaced with a consistent and standardized format, namely 'Low Fat' and 'Regular'.

**4. Analytical Queries**

**4.1 Retrieve Unique Item Fat Content Values**

To retrieve all distinct values of ItemFatContent:

SELECT DISTINCT ItemFatContent

FROM blinkit;

**4.2 Total Sales for 2022**

The following query calculates the total sales of items from outlets established in 2022:

SELECT

ROUND(SUM(sales), 0) AS 'Total\_Sales'

FROM

blinkit

WHERE

OutletEstablishmentYear = 2022;

**4.3 Average Sales for 2022**

This query calculates the average sales for outlets established in 2022:

SELECT

ROUND(AVG(sales), 0) AS 'Avg\_Sales'

FROM

blinkit

WHERE

OutletEstablishmentYear = 2022;

**4.4 Total Number of Items for 2022**

To count the number of items sold in outlets established in 2022:

SELECT

COUNT(\*) AS 'No\_of\_items'

FROM

blinkit

WHERE

OutletEstablishmentYear = 2022;

**4.5 Average Rating for All Items**

To retrieve the average rating of all items:

SELECT

ROUND(AVG(rating), 0) AS 'Avg\_Rating'

FROM

blinkit;

**4.6 Item Fat Content Sales Analysis**

This query provides a detailed analysis of total sales, average sales, number of items, and average ratings based on the ItemFatContent category:

SELECT

ItemFatContent,

ROUND(SUM(sales), 2) AS 'Total\_Sales',

ROUND(AVG(sales), 2) AS 'Avg\_Sales',

COUNT(\*) AS 'No\_of\_items',

ROUND(AVG(rating), 2) AS 'Avg\_Rating'

FROM

blinkit

GROUP BY ItemFatContent

ORDER BY Total\_Sales DESC;

**4.7 Top 5 Item Types by Sales**

To retrieve the top 5 item types based on total sales:

SELECT

ItemType,

ROUND(SUM(sales), 2) AS 'Total\_Sales',

ROUND(AVG(sales), 2) AS 'Avg\_Sales',

COUNT(\*) AS 'No\_of\_items',

ROUND(AVG(rating), 2) AS 'Avg\_Rating'

FROM

blinkit

GROUP BY ItemType

ORDER BY Total\_Sales DESC

LIMIT 5;

**4.8 Sales by Fat Content and Outlet Location**

This query provides an analysis of sales for "Low Fat" and "Regular" items, broken down by outlet location type:

SELECT

OutletLocationType,

ROUND(SUM(CASE

WHEN ItemFatContent = 'Low Fat' THEN sales

ELSE 0

END), 2) AS 'Low\_Fat\_Sales',

ROUND(SUM(CASE

WHEN ItemFatContent = 'Regular' THEN sales

ELSE 0

END), 2) AS 'Regular\_Sales'

FROM

blinkit

GROUP BY OutletLocationType

ORDER BY OutletLocationType;

**4.9 Total Sales by Outlet Establishment Year**

This query summarizes total sales, average sales, number of items, and average ratings based on the year the outlet was established:

SELECT

OutletEstablishmentYear,

ROUND(SUM(sales), 2) AS 'Total\_Sales',

ROUND(AVG(sales), 2) AS 'Avg\_Sales',

COUNT(\*) AS 'No\_of\_items',

ROUND(AVG(rating), 2) AS 'Avg\_Rating'

FROM

blinkit

GROUP BY OutletEstablishmentYear

ORDER BY Total\_Sales DESC;

**4.10 Sales and Percentage by Outlet Size**

This query calculates the total sales and percentage of total sales contributed by each outlet size:

SELECT

OutletSize,

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

ROUND((SUM(Sales) / (SELECT SUM(Sales) FROM Blinkit)), 2) \* 100 AS SalesPercentage

FROM

Blinkit

GROUP BY OutletSize;

**4.11 Sales and Rating Summary by Outlet Location (2020)**

To analyze the total sales, sales percentage, and ratings by outlet location type for outlets established in 2020:

SELECT

OutletLocationType,

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

ROUND((SUM(Sales) / (SELECT SUM(Sales) FROM Blinkit)), 2) \* 100 AS SalesPercentage,

ROUND(AVG(sales), 2) AS 'Avg\_Sales',

COUNT(\*) AS 'No\_of\_items',

ROUND(AVG(rating), 2) AS 'Avg\_Rating'

FROM

Blinkit

WHERE

OutletEstablishmentYear = 2020

GROUP BY OutletLocationType

ORDER BY TotalSales DESC;

**4.12 Sales and Rating Summary by Outlet Type**

This query summarizes the total sales, sales percentage, and ratings by outlet type:

SELECT

OutletType,

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

ROUND((SUM(Sales) / (SELECT SUM(Sales) FROM Blinkit)), 2) \* 100 AS SalesPercentage,

ROUND(AVG(sales), 2) AS 'Avg\_Sales',

COUNT(\*) AS 'No\_of\_items',

ROUND(AVG(rating), 2) AS '

Avg\_Rating'  
FROM  
Blinkit  
GROUP BY OutletType  
ORDER BY TotalSales DESC;

**5. Conclusion**

The Blinkit database provides valuable insights into the sales dynamics, product performance, and outlet-level metrics. By using these SQL queries, businesses can evaluate their sales strategies, identify top-performing products, understand customer preferences, and optimize outlet performance.