**Market Basket Analysis & Store Optimization**

## Module: SQL Analysis

## Business Simulation Context

“SmartMart” is a **mid-size supermarket chain with 40+ outlets across metro cities**. Recently, the management has observed:

* While **sales have increased**, **profit margins are declining**.
* Popular products **run out of stock too quickly**.
* Some products are **never bought together**, leading to **poor shelf placement and unnecessary waste**.

To address these challenges, SmartMart has launched an initiative to **optimize store layout and promotions using Market Basket Analysis.** As part of the **Retail Data Strategy Team**, our group is responsible for delivering **data-driven, actionable insights using SQL**.

## Project Objectives

Using SQL, we aim to:

* **Identify products that are frequently bought together** to improve store layout and bundling.
* Help **reposition and bundle products** for a better customer experience and higher sales.
* **Optimize shelf space and stock placement** to reduce waste and stockouts.
* Recommend **pricing and promotion strategies using association rules** to improve profit margins.

## Tools & Technologies Used

**SQL (MySQL):** For querying and analyzing transactional sales and basket data to extract actionable insights.

**Dataset:** **market\_basket\_data** and **transactions** tables provided by SmartMart for analysis.

SQL Queries for Market Basket Analysis

## Goal of SQL Queries

To identify:

* **Top-selling products** driving sales but possibly reducing margins.
* **Frequently purchased product pairs** for bundling and shelf optimization.
* **Sales seasonality and day-of-week trends** to align promotions.
* **Basket size patterns** for upselling opportunities.

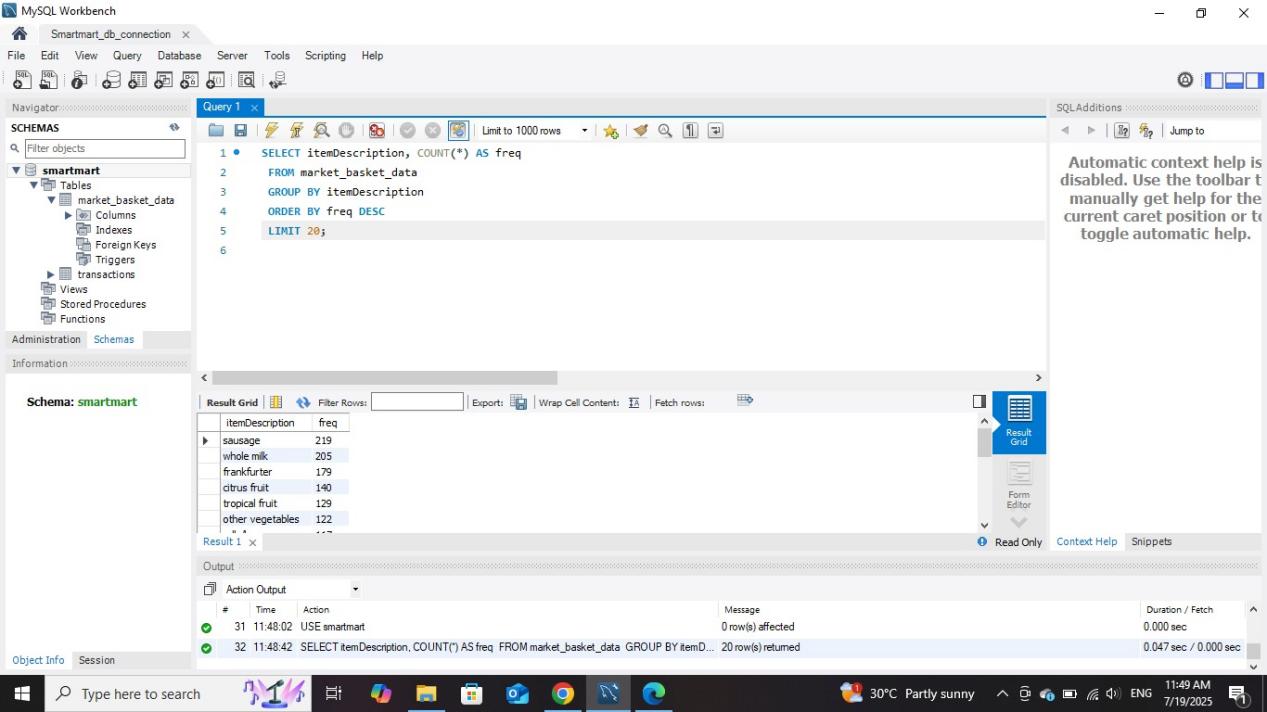
**SQL Queries Executed**1) SELECT itemDescription, COUNT(\*) AS freq

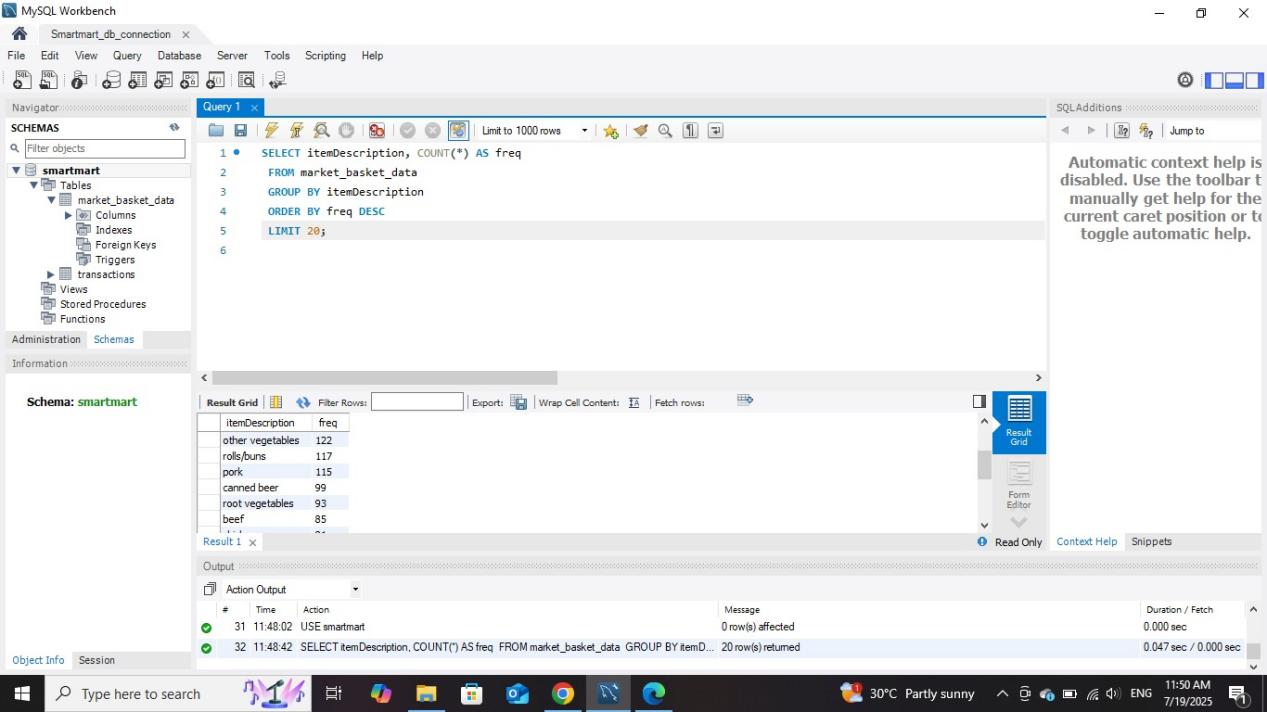
FROM market\_basket\_data

GROUP BY itemDescription

ORDER BY freq DESC

LIMIT 20;

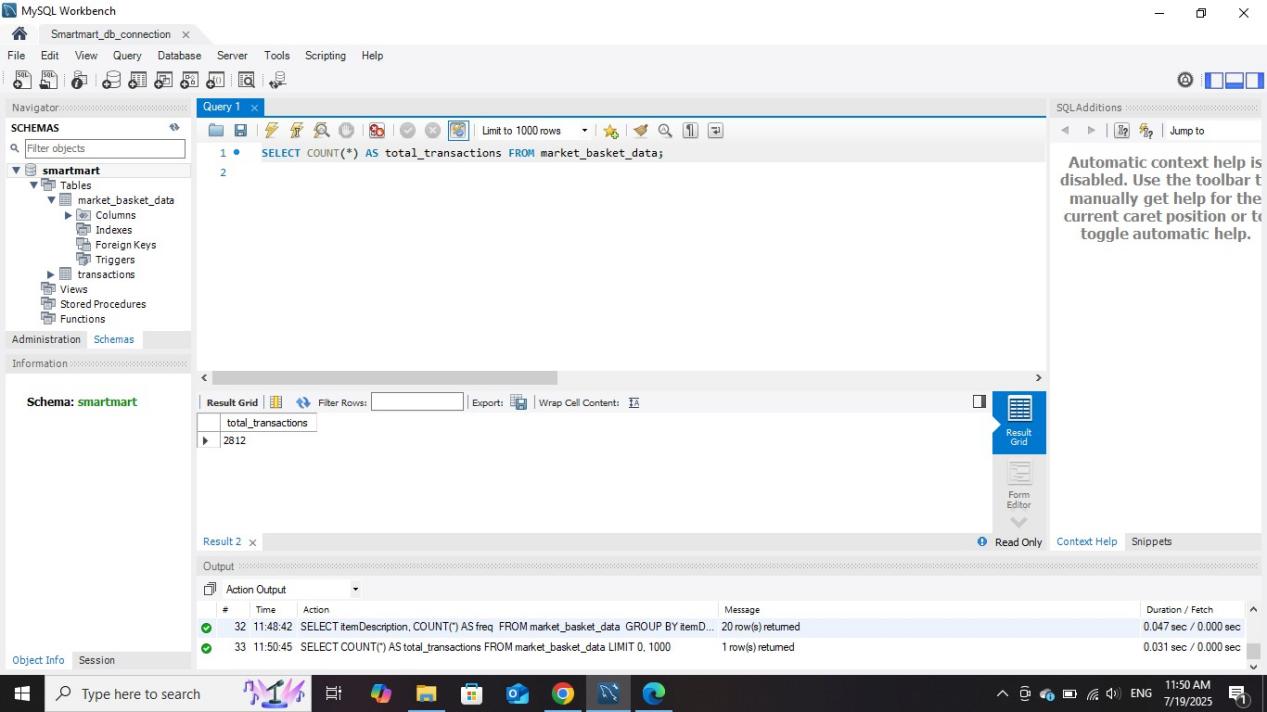
****



1. USE sales;

SELECT COUNT(\*) AS total\_transactions

FROM market\_basket\_data;

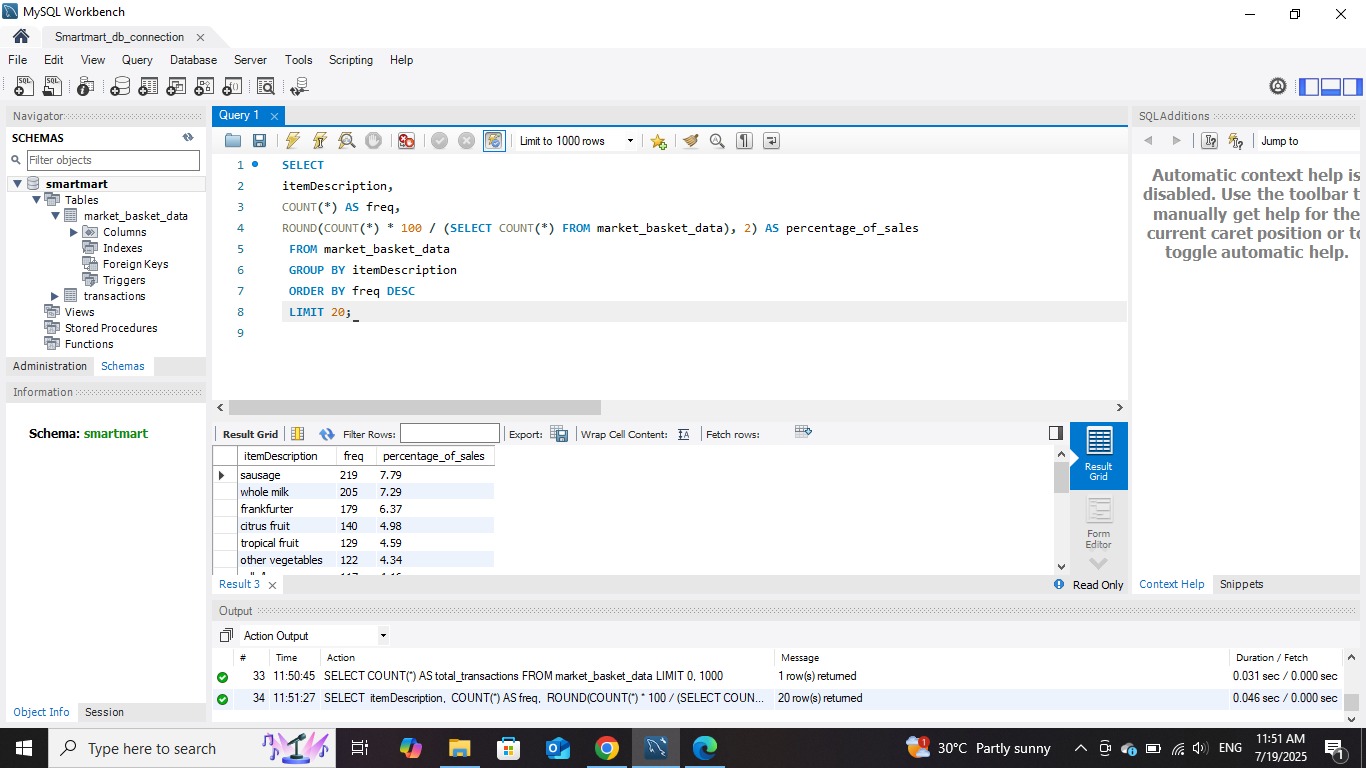


1. SELECT itemDescription, COUNT(\*) AS freq, ROUND(COUNT(\*) \* 100 / (SELECT COUNT(\*) FROM market\_basket\_data), 2) AS percentage\_of\_sales

FROM market\_basket\_data

GROUP BY itemDescription

ORDER BY freq DESC LIMIT 20;



1. SELECT

day\_of\_week,

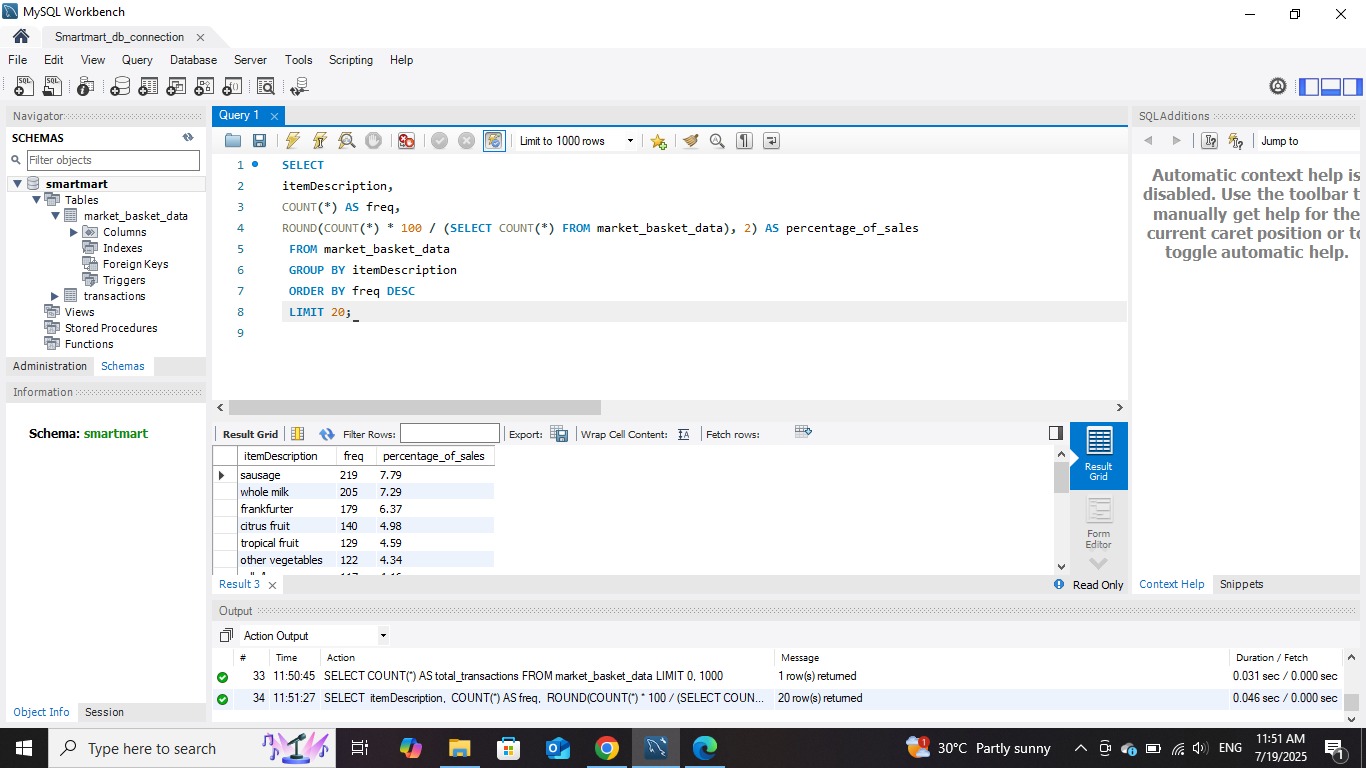
itemDescription,

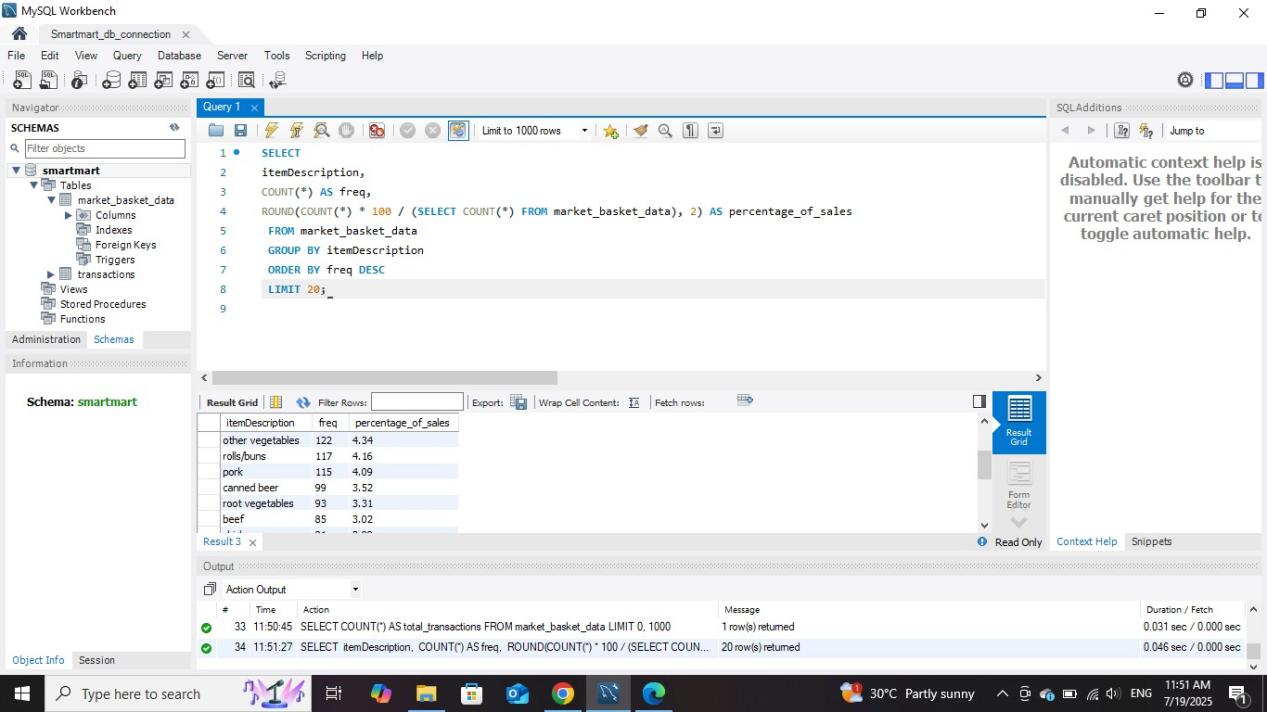
COUNT(\*) AS freq

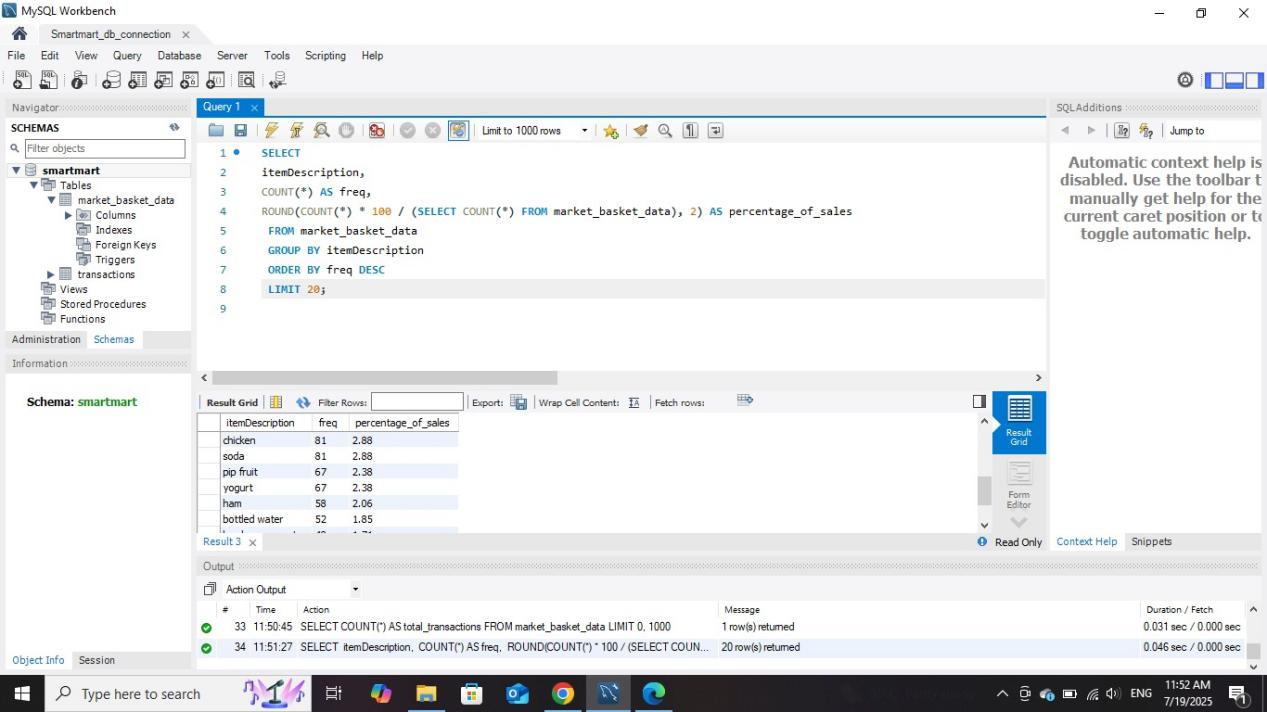
FROM market\_basket\_data

GROUP BY day\_of\_week, itemDescription

ORDER BY day\_of\_week ASC, freq DESC;



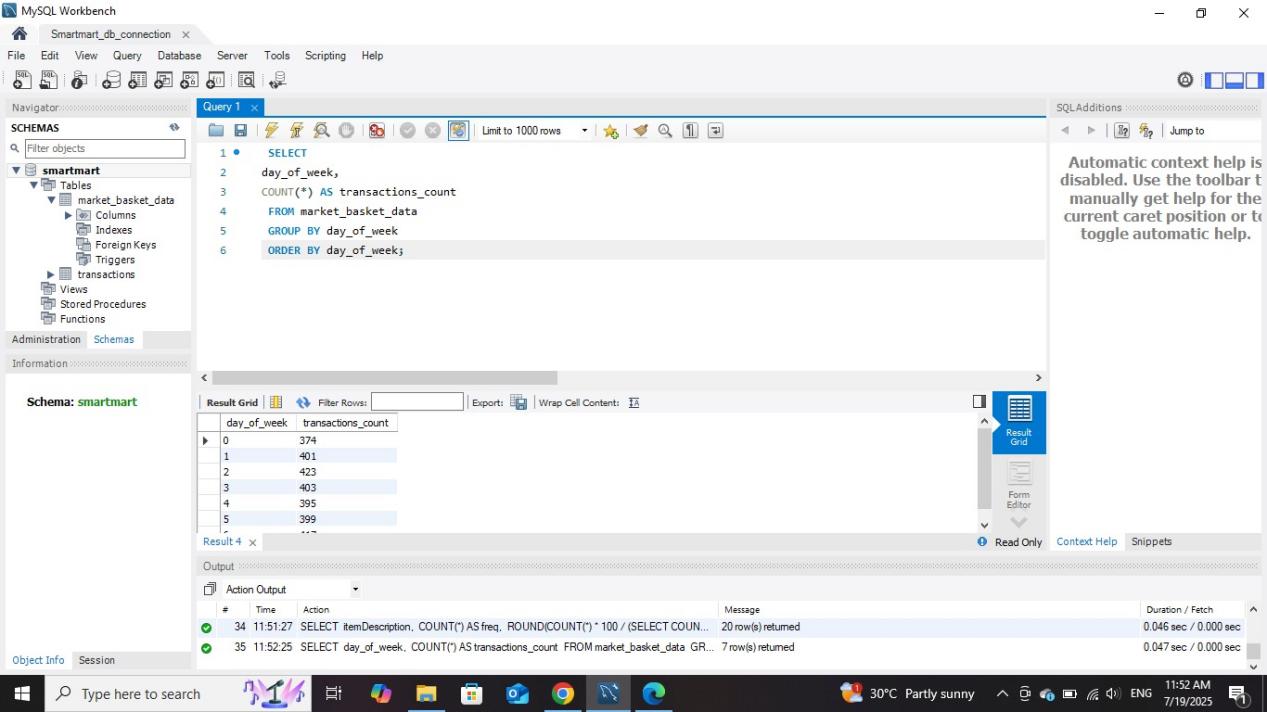


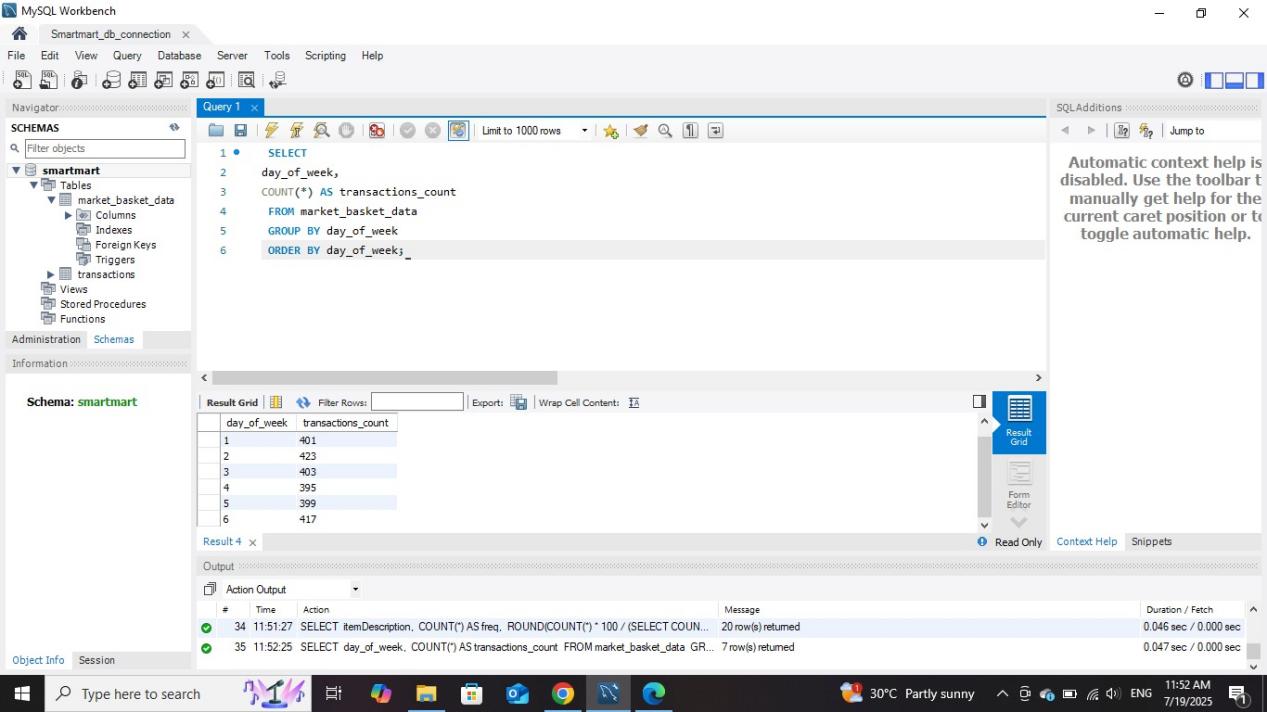


1. SELECT day\_of\_week, COUNT(\*) AS transactions\_count

FROM market\_basket\_data

GROUP BY day\_of\_week

ORDER BY day\_of\_week;  
  




1. SELECT

day\_of\_week,

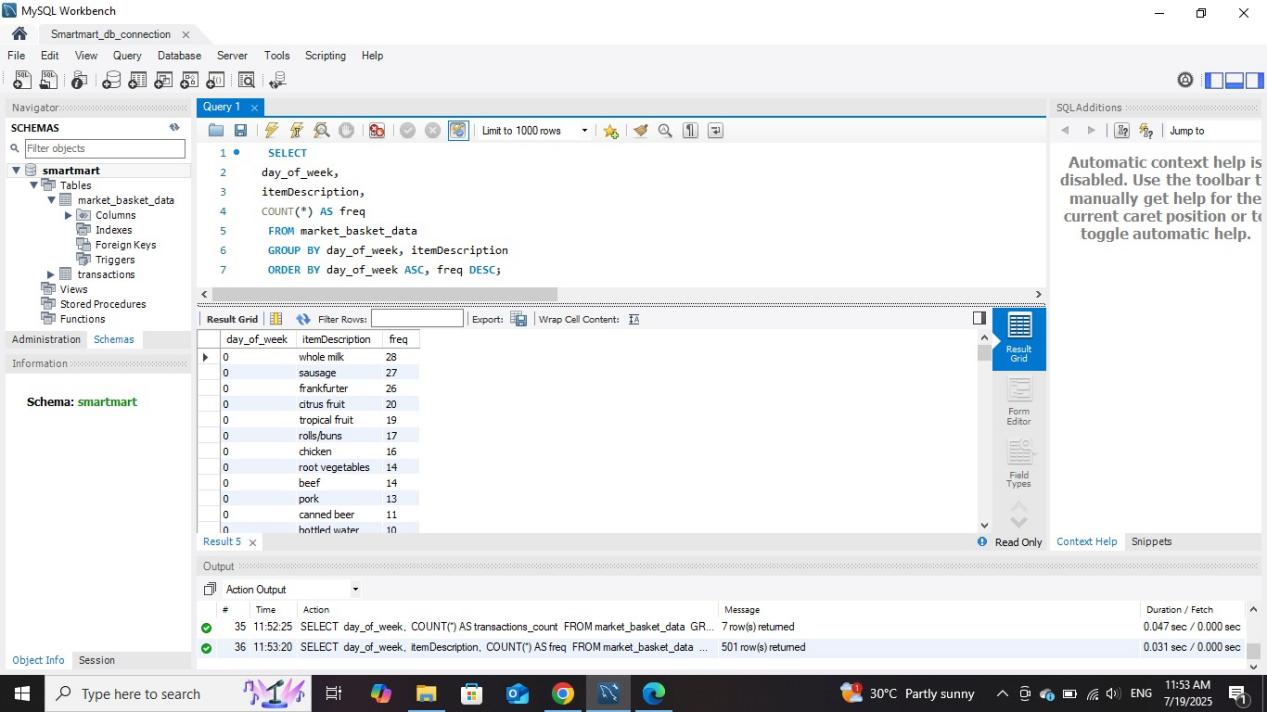
itemDescription,

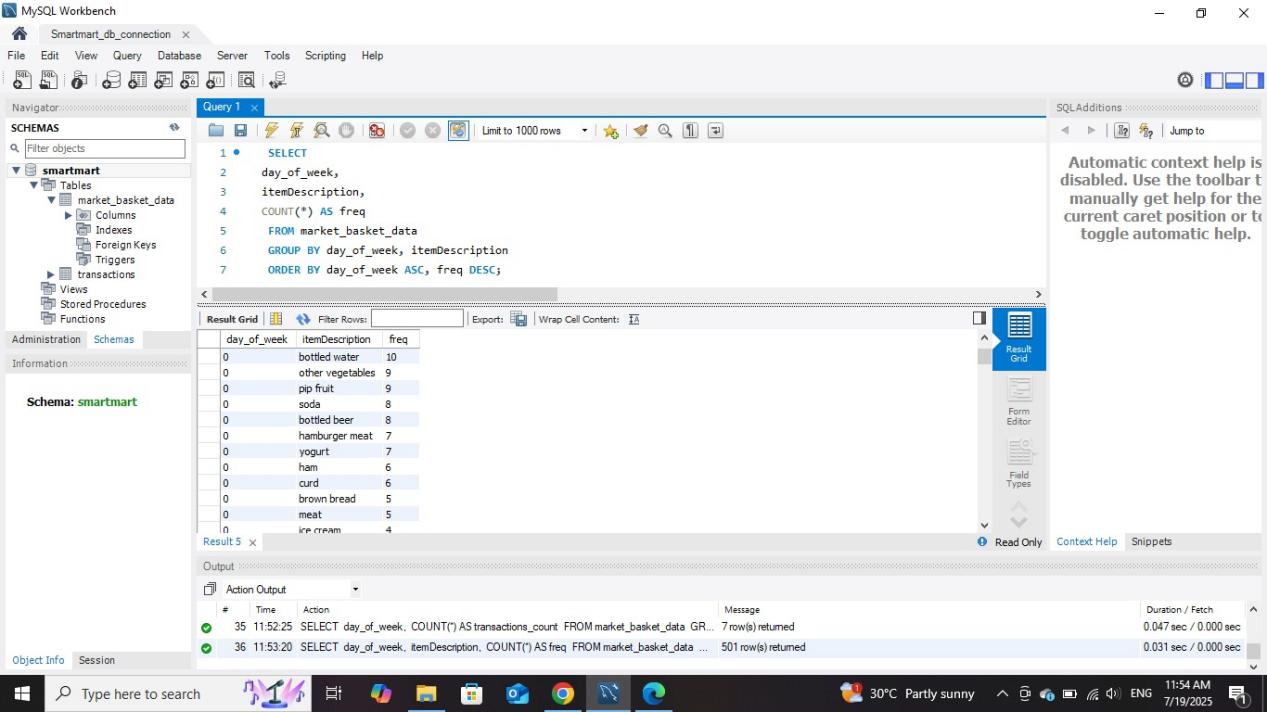
COUNT(\*) AS freq

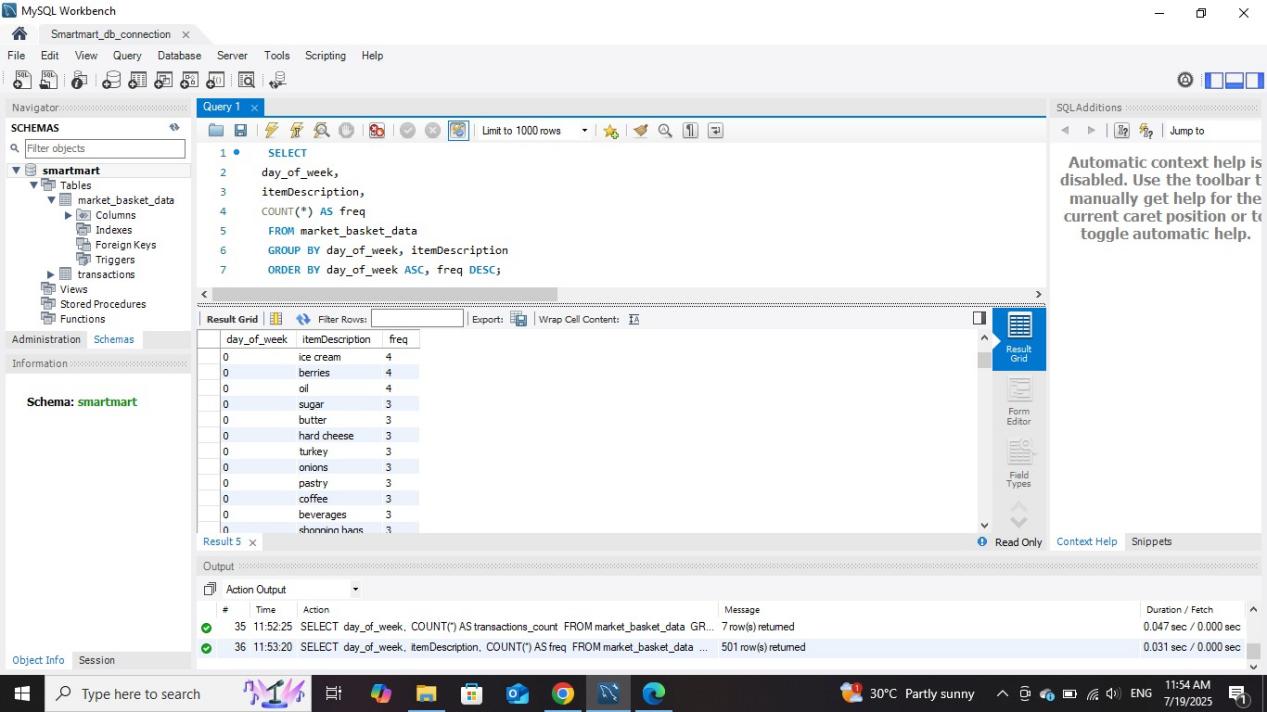
FROM market\_basket\_data

GROUP BY day\_of\_week, itemDescription

ORDER BY day\_of\_week ASC, freq DESC;





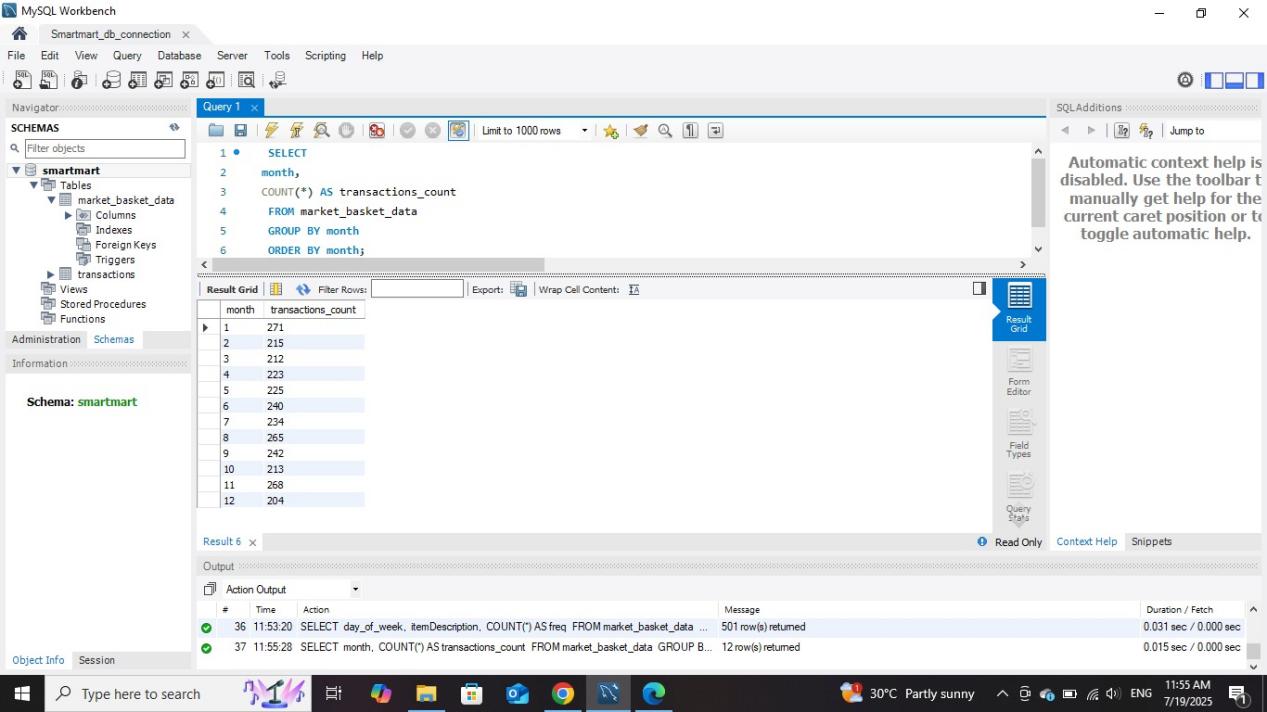


1. SELECT month, COUNT(\*) AS transactions\_count

FROM market\_basket\_data

GROUP BY month

ORDER BY month;



‘

1. SELECT

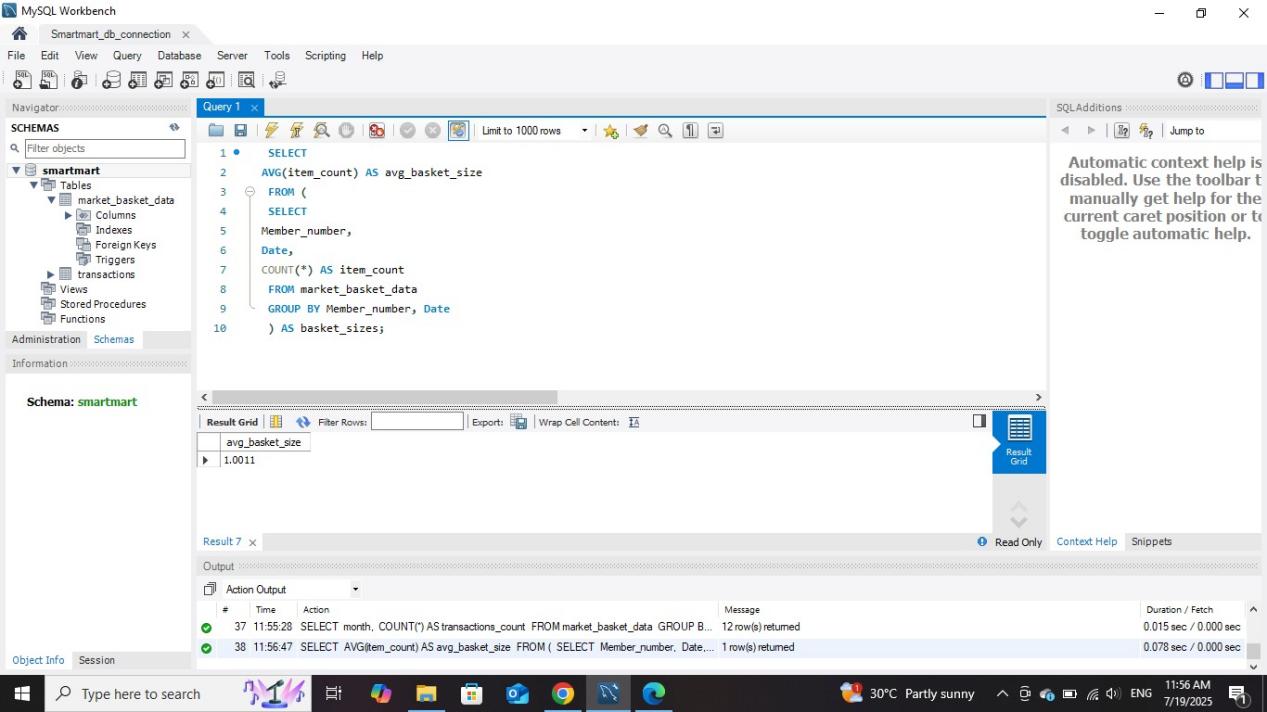
AVG(item\_count) AS avg\_basket\_size

FROM ( SELECT

Member\_number,

Date, COUNT(\*) AS item\_count

FROM market\_basket\_data

GROUP BY Member\_number, Date) AS basket\_sizes;  
  


1. SELECT

a.itemDescription AS item1,

b.itemDescription AS item2,

COUNT(\*) AS freq

FROM market\_basket\_data a

JOIN market\_basket\_data b

ON a.Member\_number = b.Member\_number

AND a.Date = b.Date AND a.itemDescription < b.itemDescription

GROUP BY item1, item2

ORDER BY freq DESC

LIMIT 20;

