

**Department of Computer Engineering**

**Course: DWMM**

**Mini-Project – Phase 3 Report**

**Guidance By - Kumavat Ma’am**

**Topic: Analytical Query Processing on Dairy Management System**

**By**

|  |  |  |
| --- | --- | --- |
| **Roll No** | **SRN** | **Name of Student** |
| 38 | 202200930 | Shyamal S. Patil |
| 39 | 202200931 | M. Asad Shaikh |

**Index**

|  |  |  |
| --- | --- | --- |
| **SNO** | **Topic** | **Page No** |
| **1** | **Analytical Query Processing** | **3** |
| **2** | **Outcome** | **8** |

**Analytical Queries:**

In our Dairy Management System with a multidimensional model, we can execute various analytical queries to extract valuable insights for decision-making. Here are a few samples analytical queries that we can run using SQL:

**Customer Analysis:**

1. Top Customers by Total Spending

**Category:** Customer Analysis

**Query:**

SELECT

cd.CustomerName,

SUM(ds.SalesAmount) AS TotalSpending

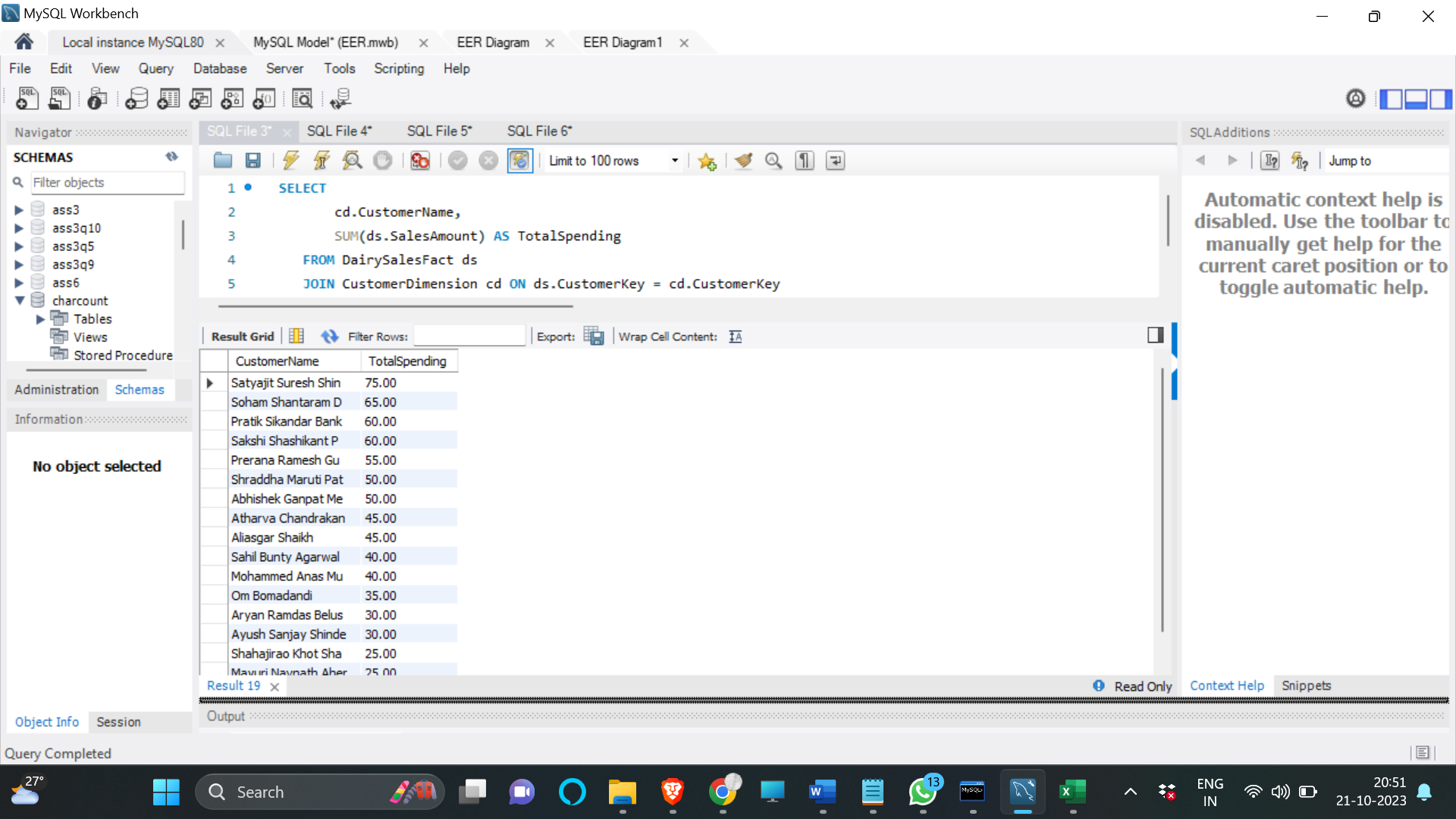
FROM DairySalesFact ds

JOIN CustomerDimension cd ON ds.CustomerKey = cd.CustomerKey

GROUP BY CustomerName

ORDER BY TotalSpending DESC;

**Output:**



**Time Analysis:**

2. Average Sales Amount per Month

**Category**: Time Analysis

**Query:**

SELECT

td.CalendarYear,

td.CalendarMonth,

AVG(ds.SalesAmount) AS AvgSalesAmount

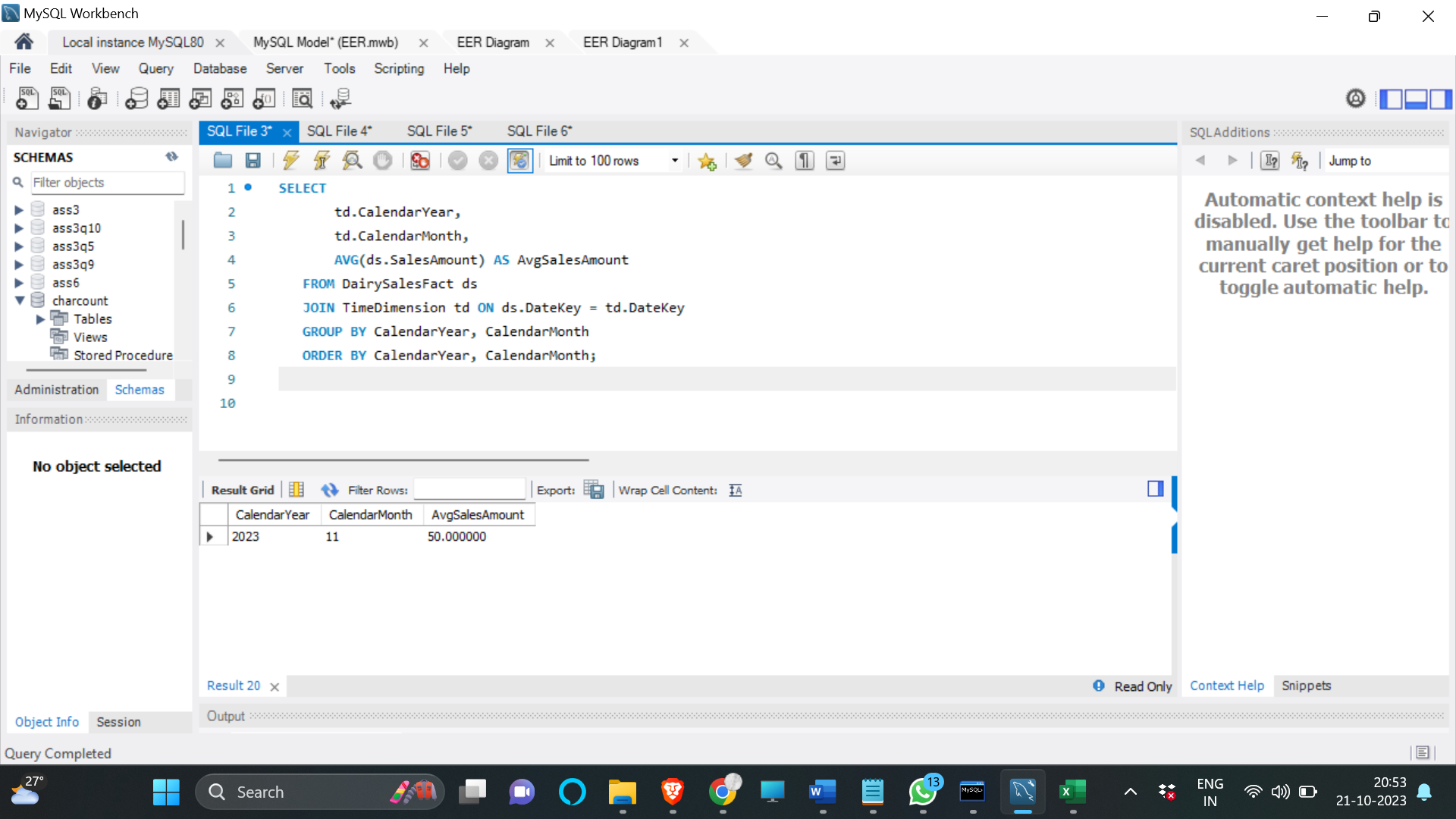
FROM DairySalesFact ds

JOIN TimeDimension td ON ds.DateKey = td.DateKey

GROUP BY CalendarYear, CalendarMonth

ORDER BY CalendarYear, CalendarMonth;

**Output:**



**Product Analysis:**

3. Top Selling Products:

To find the top-selling products

**Category:** Product Analysis

**Query:**

SELECT

pd.Product\_Name,

SUM(ds.QuantitySold) AS TotalQuantitySold

FROM DairySalesFact ds

JOIN Product\_Dim pd ON ds.ProductKey = pd.Product\_ID

GROUP BY Product\_Name

ORDER BY TotalQuantitySold DESC;

**Output:**



6. Products Needing Reorder

**Category:** Product Analysis

**Query:**

SELECT

pd.Product\_Name,

pd.Max\_Order\_Quantity - pd.Quantity\_In\_Stock AS ReorderQuantity

FROM Product\_Dim pd

WHERE pd.Quantity\_In\_Stock < pd.ReorderPoint;

**Output:**



**Sales Analysis:**

7. Total Sales Amount and Profit by Date:

This query will help us to calculate the total sales amount and profit for each date.

**Category:** Sales Analysis

**Query:**

SELECT

DateKey,

SUM(SalesAmount) AS TotalSalesAmount,

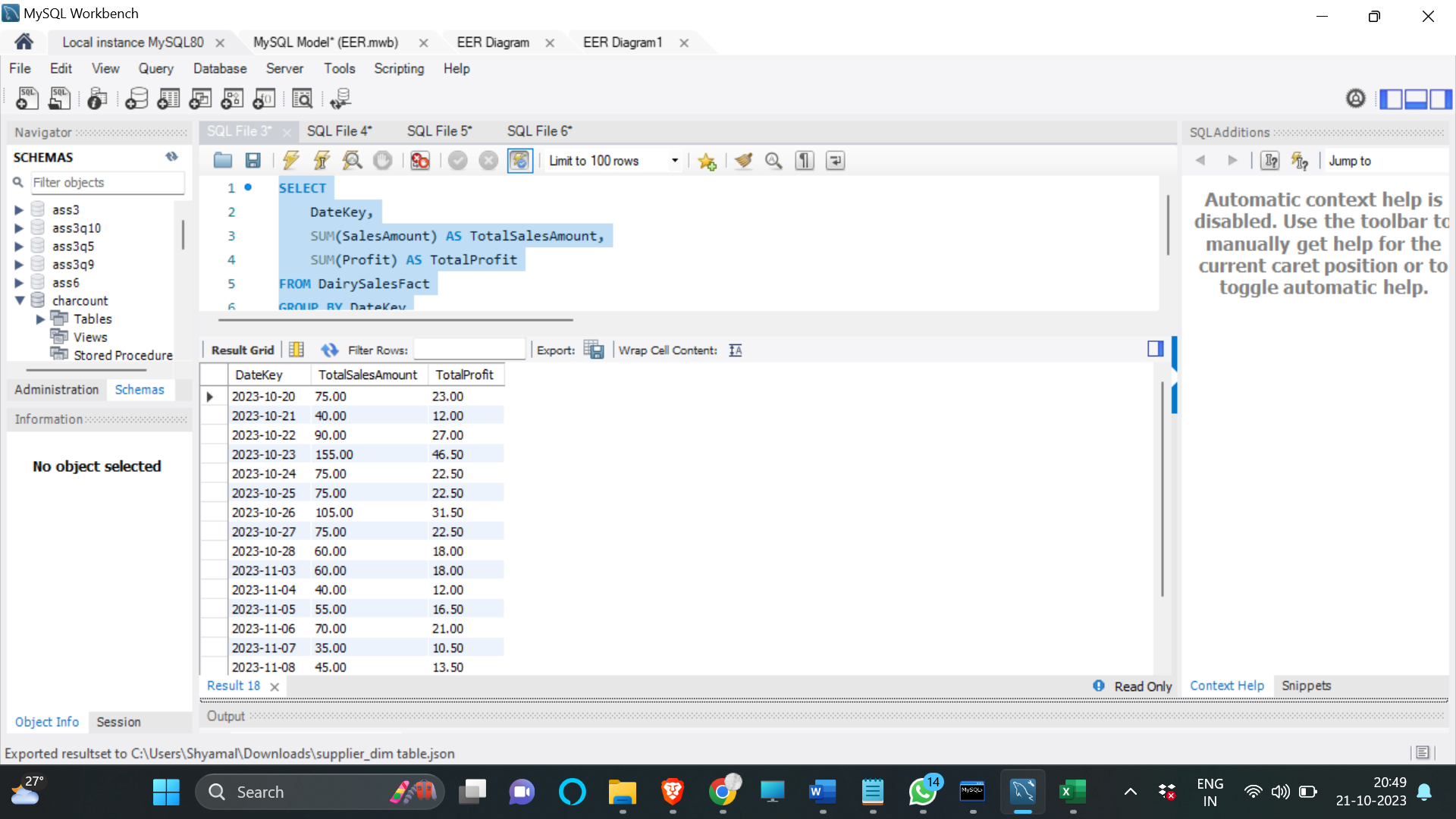
SUM(Profit) AS TotalProfit

FROM DairySalesFact

GROUP BY DateKey

ORDER BY DateKey;

**Output:**



**Inventory Analysis:**

8. Quarterly Profit Analysis

**Category:** Profit Analysis

**Query:**

SELECT

td.CalendarYear,

td.CalendarQuarter,

SUM(s.Profit) AS TotalProfit

FROM DairySalesFact s

JOIN TimeDimension td ON s.DateKey = td.DateKey

GROUP BY CalendarYear, CalendarQuarter

ORDER BY CalendarYear, CalendarQuarter;

**Output:**



**Customer Geography Analysis:**

9. Customer Geography Analysis

**Category:** Customer Analysis

**Query:**

SELECT

cd.CustomerCountry AS Country,

SUM(ds.SalesAmount) AS TotalSalesAmount

FROM DairySalesFact ds

JOIN CustomerDimension cd ON ds.CustomerKey = cd.CustomerKey

GROUP BY Country

ORDER BY TotalSalesAmount DESC;

**Output:**



**Outcome:**

**Indexing:**

Utilize indexing to enhance query performance. Create indexes on foreign keys, frequently queried columns, and dimension keys. This will significantly improve data retrieval from fact and dimension tables.

**Partitioning:**

Consider partitioning large tables to optimize query performance. For instance, partition the Sales\_Dim table based on time, creating separate partitions for different years or months. This reduces the amount of data scanned during queries.

**Aggregate Tables:**

Create aggregate tables to pre-compute summary statistics. For example, store pre-computed monthly or yearly sales totals. This accelerates complex aggregation queries and contributes to efficient decision-making within our Dairy Management System.