VISHWAKARMA UNIVERSITY

Maximising Human Potential

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:

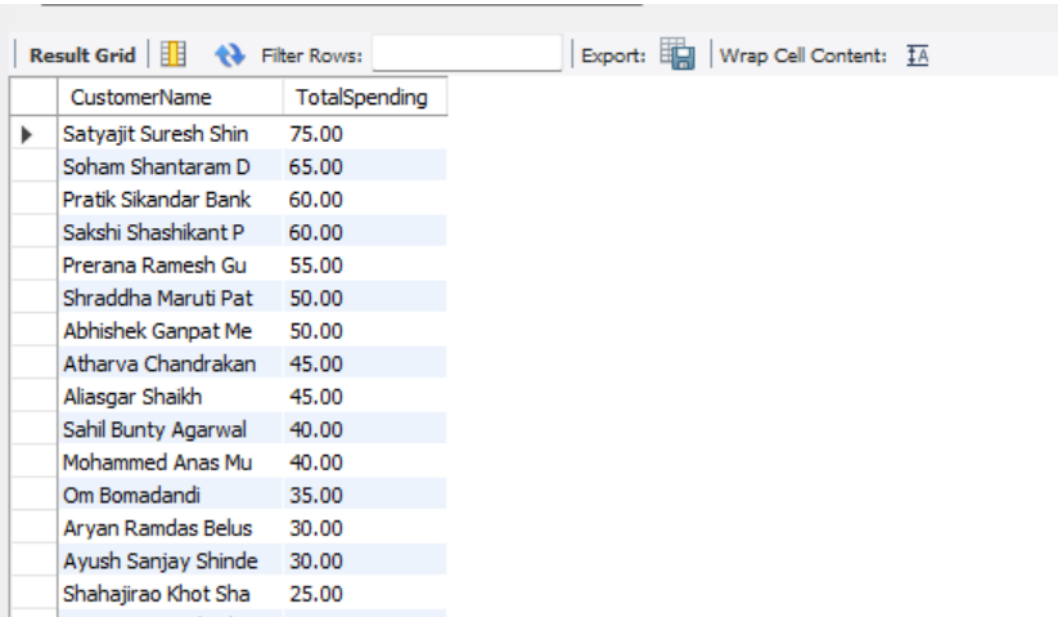
In our Dairy Management System with a multidimensional model, we can execute various analytical queries to extract valuable insights for decision-making. Here are a few samples analytical queries that we can run using SQL:

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:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

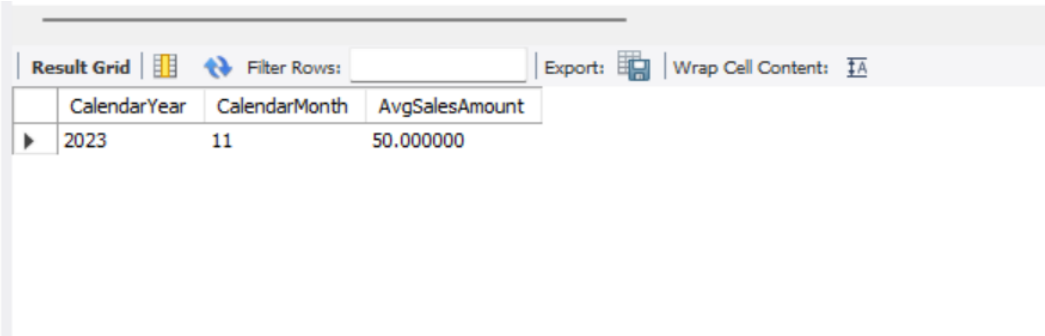
CustomerName	TotalSpending
Satyajit Suresh Shin	75.00
Soham Shantaram D	65.00
Pratik Sikandar Bank	60.00
Sakshi Shashikant P	60.00
Prerana Ramesh Gu	55.00
Shraddha Maruti Pat	50.00
Abhishek Ganpat Me	50.00
Atharva Chandrakan	45.00
Aliasgar Shaikh	45.00
Sahil Bunty Agarwal	40.00
Mohammed Anas Mu	40.00
Om Bomadandi	35.00
Aryan Ramdas Belus	30.00
Ayush Sanjay Shinde	30.00
Shahajirao Khot Sha	25.00

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:



CalendarYear	CalendarMonth	AvgSalesAmount
2023	11	50.000000

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:

Supplier_Name	TotalSalesAmount

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:

Supplier_Name	TotalPurchases	TotalAmountPurchased

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.

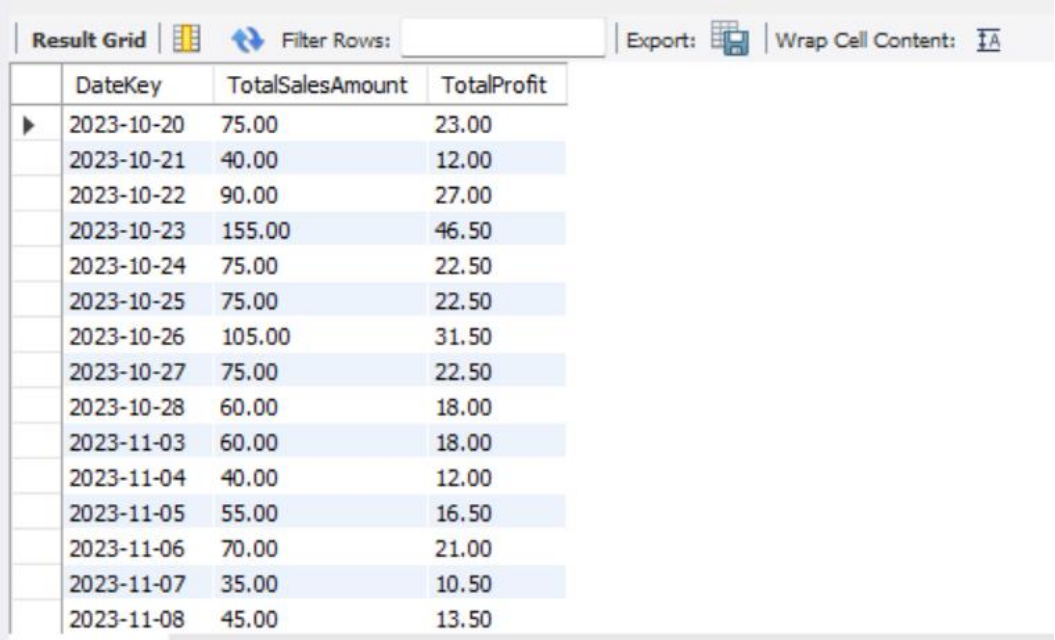
Category: Sales Analysis

Query:

```
SELECT
```

```
    DateKey,
    SUM(SalesAmount) AS TotalSalesAmount,
    SUM(Profit) AS TotalProfit
FROM DairySalesFact
GROUP BY DateKey
ORDER BY DateKey;
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

DateKey	TotalSalesAmount	TotalProfit
2023-10-20	75.00	23.00
2023-10-21	40.00	12.00
2023-10-22	90.00	27.00
2023-10-23	155.00	46.50
2023-10-24	75.00	22.50
2023-10-25	75.00	22.50
2023-10-26	105.00	31.50
2023-10-27	75.00	22.50
2023-10-28	60.00	18.00
2023-11-03	60.00	18.00
2023-11-04	40.00	12.00
2023-11-05	55.00	16.50
2023-11-06	70.00	21.00
2023-11-07	35.00	10.50
2023-11-08	45.00	13.50

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:

CalendarYear	CalendarQuarter	TotalProfit
2023	4	30

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:

Country	TotalSalesAmount
India	750

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.