Project Report

Data Warehousing & Business Intelligence

Submitted to: Dr. Asif Naeem

Submitted by: Abdullah Khan (20i-0894)

Table of Contents

[OVERVIEW 2](#_Toc120655282)

[SCHEMA FOR WAREHOUSE 2](#_Toc120655283)

[MESHJOIN ALGORITHM 3](#_Toc120655284)

[SHORTCOMINGS OF MESHJOIN ALGORITHM 3](#_Toc120655285)

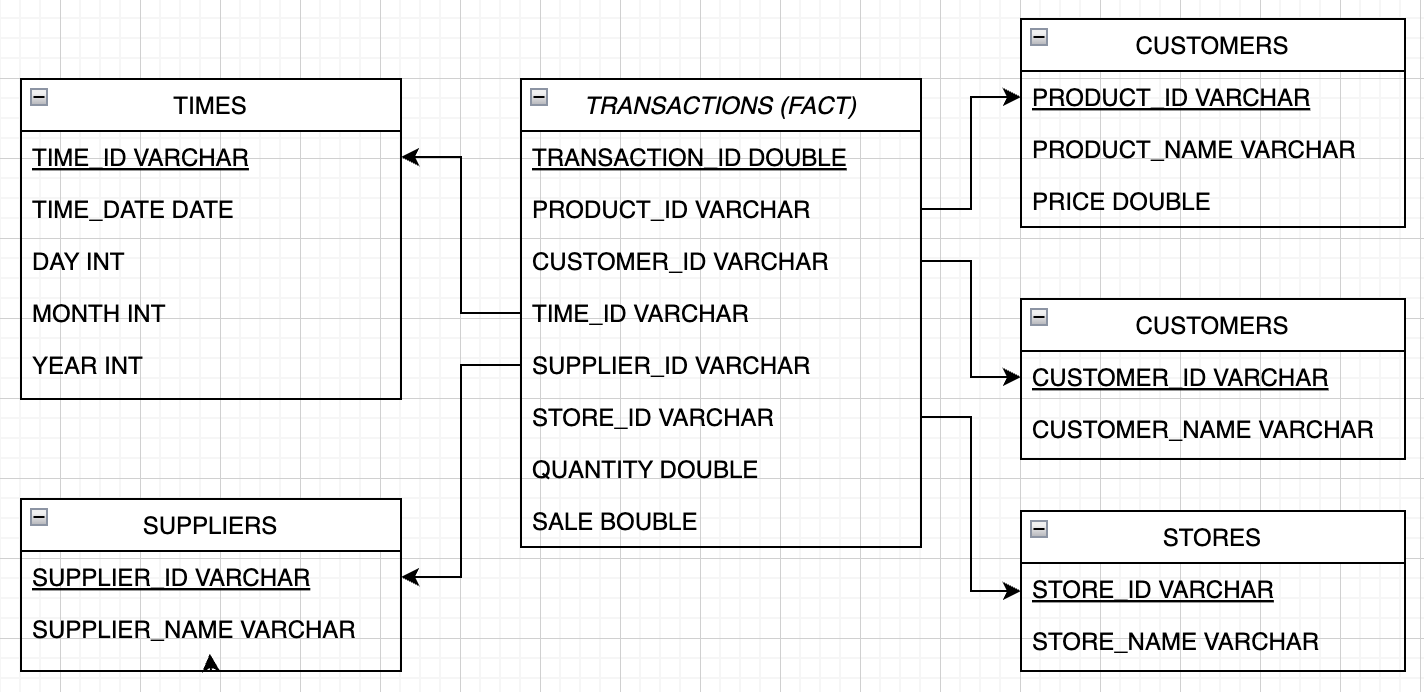
[LEARNING OUTCOMES 3](#_Toc120655286)

[QUERIES OUTPUT: 4](#_Toc120655287)

# OVERVIEW

This project is the course project for the course Data Warehousing and Business Intelligence for Fall ’22 semester. The goal of the project is to design a Data Warehouse for the Metro Cash and Carry transactions data. The data consists of transactional data from the database and the master data containing product details and customer details. To handle the data coming from the sources in a stream, we will implement a mesh Join algorithm to handle the join of master data to real-time data coming from the transactions database and implement a near real-time Data Warehouse for the business.

# SCHEMA FOR WAREHOUSE



# MESHJOIN ALGORITHM

1. Establish 3 connections with the database and the warehouse, one for master data, one for transactions, and the 3rd one to load to the warehouse.
2. Ask user whether he wants to use the default credentials or enter custom ones.
3. Write query to calculate the count of transactions in the source database.
4. Iterate on the number of transactions until the queue is not empty.
5. In each iteration,
   1. Load 100 transactions from the database.
   2. Put transaction id, product id, and customer id of the 100 transactions in a hash map key, and the whole tuple as value against the key.
   3. Put the above 100 keys in the queue as an array list of array list.
   4. Load 20 products and 10 customers from the master data to ensure the same number of partitions of the master data as that of queue.
   5. Compare the products and customers data with the keys of the hash map and append matched items to the value tuples in the hash map.
   6. Once the queue reaches 5 sets of 100 transactions, poll the first entry from the queue, get all the values against that queue array list of array list to the warehouse using update queries and reset the current products and customers to 0 to make a new cycle of iteration over the master data.
6. Display total elapsed time in the end.

# SHORTCOMINGS OF MESHJOIN ALGORITHM

Three Shortcoming of Mesh Join Algorithm:

1. The wasted cost generated by the mesh join is huge. It checks the transactions against the whole master data, even though most of the products have repetitive products which are sold the most.
2. As the data is loaded sequentially, there is a high chance that the incoming tuples do not match with the data present in the buffer at the time of arrival.
3. Optimizing the disk, I/O is difficult in mesh join. If we want to optimize the number of disk I/O, we need to increase the buffer size to read more data at once depending on the size of the master data as the whole master data must be loaded in each cycle of loading.

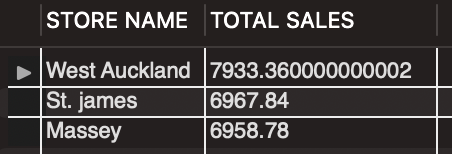
# LEARNING OUTCOMES

The main learnings from this project are:

* Learnt how to implement a data warehouse for real time transactional data.
* Implemented a join type that works on real time data to provide near real-time warehousing.
* Learnt that everything has its goods and bad and so does mesh join. Where it helps us provide a real time join, its disk I/O costs are huge compared to traditional joins.
* Wrote OLAP queries for a warehouse using roll up and drill down method and learnt how these methods allow the business intelligence to people to do business analysis at a larger scale.
* Documented a course project for the first time.
* Wrote a read me file which helped me understand how important it is to help the user of your project run the project on their own machines for further analysis, reference, and testing.

# QUERIES OUTPUT:

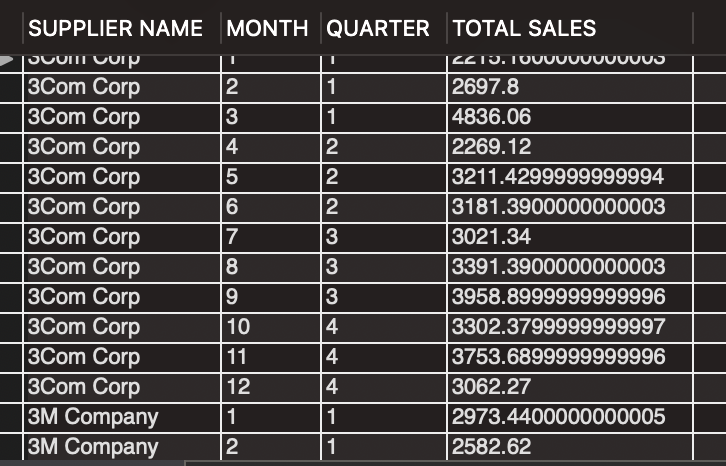
Q1:



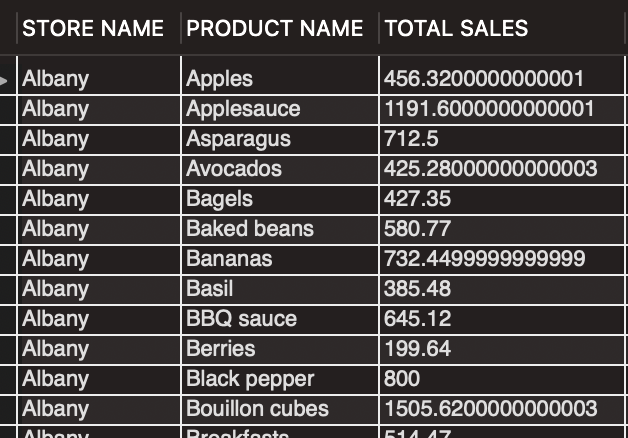
Q2(Top supplier for the next weekend can be found by the fact which suppliers have performed the best during the previous weekend and overall best performers on weekends.):



Q3:



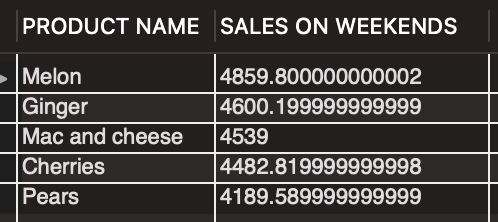
Q4:



Q5:



Q6:



Q7:



Q8:



Q9(One product (Tomatoes) has the same name but different product ID.):



Q10:

