Near Real Time WareHouse for Metro Store Project Report

Name : Muhammad Haris

Roll Number : 19I-0740

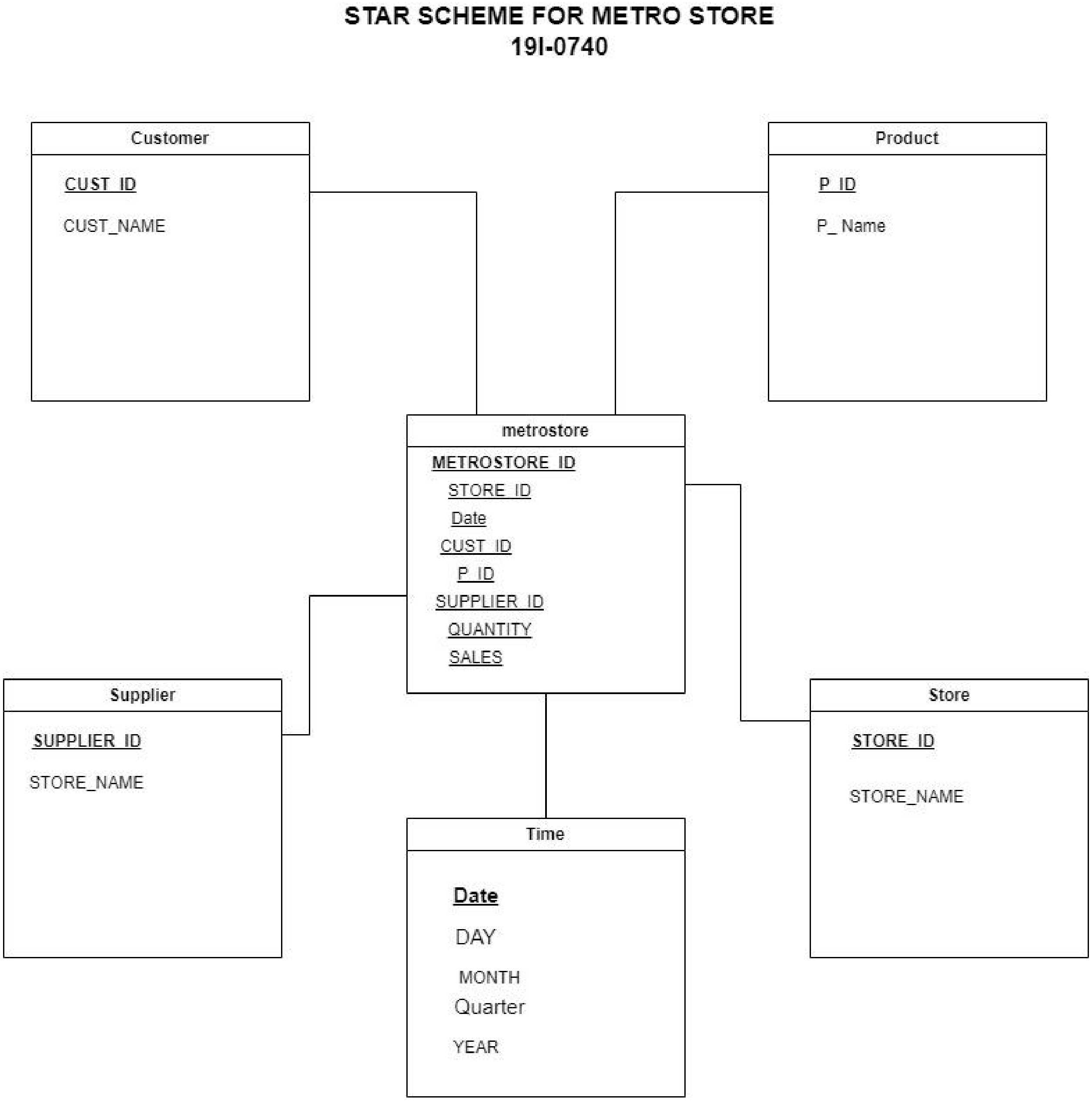
Section : CS-A

Project Overview

METRO is one of the biggest superstore chains in Pakistan. The store has thousands of customers and therefore it is important for the store to online analyze the shopping behavior of their customers. Based on that the store can optimize their selling strategies e.g. giving promotions on different products. Now, to make this analysis of shopping behavior practical there is a need of building a near-real time DW and customers' transactions from Data Sources (DSs) are required to reflect into DW as soon as they appear in DSs. To build a near real-time DW we need to implement a near-real-time ETL (Extraction, Transformation, and Loading) tools. Since the data generated by customers is incomplete as required by DW, it needs to be completed in the transformation layer of ETL To implement this enrichment feature in the transformation phase of ETL, we need a StreamRelation join operator.

The MESH JOIN (Mesh Join) algorithm was introduced by Polyzotis in 2008 with the objective of implementing the Stream- Relation join operation in the transformation phase of ETL. The main components of MESHJOIN are: The disk-buffer which will be an array and used to load the disk partitions in memory. Typically, MD is large, it has to be loaded in memory in partitions. Normally, the size of each partition in MD is equal to the size of the disk-buffer. Also MD is traversed cyclically in an endless loop. The hash table which stores the customers' transactions (tuples). The queue is used to keep the record of all the customers' transactions in memory with respect to their arrival times. The queue has the same number of partitions as MD to make sure that each tuple has joined with the whole MD before leaving the join operator. The stream-buffer will be an array and is used to hold the customer transaction meanwhile the algorithm completes one iteration

Schema



MESH JOIN

```
Method getTransactiondata:
    Read the transactions
    Load in the HashTable
    Return the index of the last transaction

Load the Transactions data chunk in the buffer
Load the Customers Master Data chunk in the Customer Buffer
Load the Products Master Data chunk in the Product Buffer

Iterating the chunk of the Transactions and Customer’s First chunk
Get the Hash value from the hash table
IF Transactions.C_ID =  Customer.C_ID
    Update the Transaction Tuple
ELSE
    move to the next chunk of the data of the Customer
Iterate the chunk of the Product table
IF Transaction.P_ID = Product.P_ID
    Update the Product name, Supplier ID, Supplier Name
    Calculate the Sale by multiplying the Price from Product and Quantity
ELSE
    Move to the next chunk

Load the Joined Tuple in the DW
```

Shortcomings of MESH JOIN

- If we are having huge Master Data the performance of the mesh join will be reduced as we have to check each chunk of the transaction will every master data chunk.
- We have to change the Master Data tuple's size(chunk) in case the transaction Data chunk changes

Learning

By working on this project, I get insight into how things work in a real time warehouse and how the big giants are managing, analyzing big data and by maintaining the warehouse and using Business intelligence they are growing in business and providing the user with best possible results. I get to know the working of mesh join algorithms in real time businesses.