

**DATA WAREHOUSE PROJECT**

**Name:** Faraz Saeed

**Roll #:** i160107

**Submitted to:** Sir Asif Naeem

**DATE:** 1st June 2020

**Project Overview**

The project implements the building of the Dara Warehouse for Metro Store, which is one of the biggest retail stores in Islamabad. The Data Warehouse is built so that the important business queries and analysis can be done by using the Data Warehouse.

The Transactional data is extracted from the two tables called TRANSACTIONS AND MASTERDATA and the relevant information is then transformed and loaded inside relevant tables in the Warehouse on daily basis. The data extracted from the two Data Source tables in enriched together and joined through the MESHJOIN which reads a chunk of Transaction data, joins it with the Master Data and then loads the relevant data into the relevant tables in the Warehouse.

In this case, the main components of MESHJOIN are queue, Hash table and Disk Buffer. The algorithm accept 50 transactions into the hash table, makes their entry in queue and then joins them with a chunk of Master Data of size 10 in the disk buffer. The processes is done cyclically until a transaction data chunk is joined with whole master data ( count maintained by the queue) and then it is loaded into the Relevant Warehouse tables. Before loading the tuple in the Warehouse, it checks if the dimension tables already contain the data, if yes, then only the fact table is updated, else, the required dimension tables as well as the fact table is updated. This loop breaks when all the data from transactions in loaded in the warehouse.

The schema used to implement the Data Warehouse is Star Schema. There are 5 dimension tables named DIM\_PRODUCT, DIM\_CUSTOMER, DIM\_DATE, DIM\_STORE, DIM\_SUPPLIER and 1 Fact Table named FACT\_SALE with 2 measures, QUANTITY and TOTAL\_SALE in this project. The next part of this report will guide you through the details of MESHJOIN algorithm, its shortcomings and the summary and learning of this project.

**MESHJOIN Algorithm**

DiskBuffer<MasterData>

HashTable<P\_ID,ArrayList<Joined node>>

Queue

While(true)

{

Populate diskbuffer with 10 masterdata records

Insert in queue if queue.size<10

If queue.peak.count == 10

{

Insert the transactions of this queue node into your star schema

{

If new ids already inserted in dimensions table then populate fact table only.

Else populate dimension tables as well as fact table

}

Remove transactions of this queue node from hashtable

}

For (i<50)

{

Load 50 records from transaction table and insert into hashtable

}

For (queue.size)

{

Increment the count value of each queue element

}

For(i<10)

{

Match diskBuffer records with hashtable entries and join them

}

Update loop variables

}

**Shortcomings in MeshJoin**

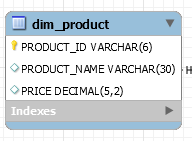
1. The dependency between different components of mesh join increases the complexity for variable quantity of data item.
2. Mesh join needs to be implemented on a different platform such as java or any other programming language. This makes it slower as compared to its alternatives such as INLJ which can be implemented in MYSQL hence shorter connection lag.
3. Mesh join being written in programming language makes it lengthier and language functions dependable instead of being dependent to only the sql.

**Schema For DW**

The schema used for this project is Star Schema.

**Dimension Tables:**

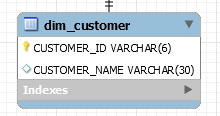
1. DIM\_PRODUCT



**Primary Key:** PRODUCT\_ID

**Foreign Keys:** NONE

1. DIM\_CUSTOMER



**Primary Key:** CUSTOMER\_ID

**Foreign Keys:** NONE

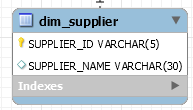
1. DIM\_STORE



**Primary Key:** STORE\_ID

**Foreign Keys:** NONE

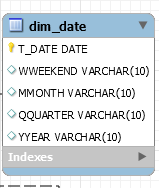
1. DIM\_SUPPLIER



**Primary Key:** SUPPLIER\_ID

**Foreign Keys:** NONE

1. DIM\_DATE

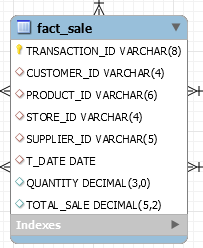


**Primary Key:** T\_DATE

**Foreign Keys:** NONE

**Fact Table:**

1. FACT\_SALE



**Primary Key:** TRANSACTION\_ID

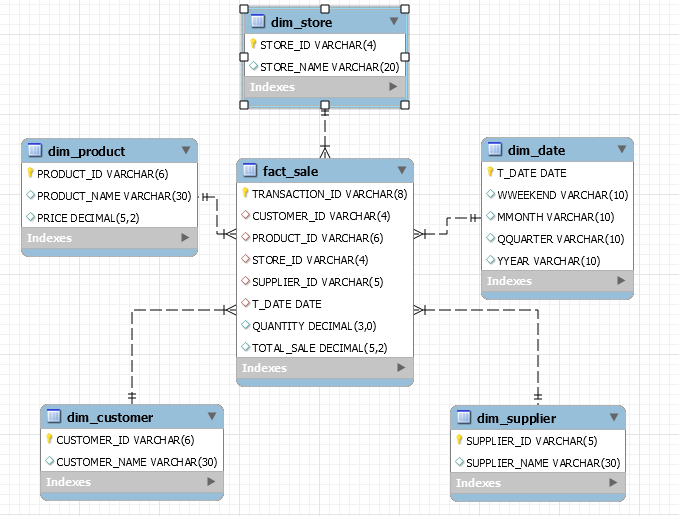
**Foreign Keys:** CUSTOMER\_ID, PRODUCT\_ID, STORE\_ID,

SUPPLIER\_ID, T\_DATE

**Measures:**

1. Quantity
2. Total\_Sale

**STAR SCHEMA:**

****

**Summary**

Doing this project was a good practical exercise for the students to learn practical skills relating to this course. From the very start of this course, I had many queries about how to manage everyday day that keeps on adding in the data sources which includes new records for dimension table. From the concept of applying mesh join to load transaction data I was able to get a clear idea of how to manage when loading new data everyday and not at once i.e. bulk load.

However, the algorithm looked like a problem in the start but as of the project ended, learning a new tool and using JDBC for implementing the Data Warehouse has further increased my practical skills. After doing the project, I have developed skills in 2 different tools (MYSQL and JAVA) by learning both in the Lab and while working on this project. Another thing that I have learned during the project was the in-depth knowledge of PL/SQL and database connectivity.

After successfully implementing the project using java and mysql and implementing those interesting queries, I believe the concepts of data warehouse along with a little practical experience, using any new tool would not create any problems for all of us now.