**Data Warehousing Report**

**Name: Muhammad Ameen**

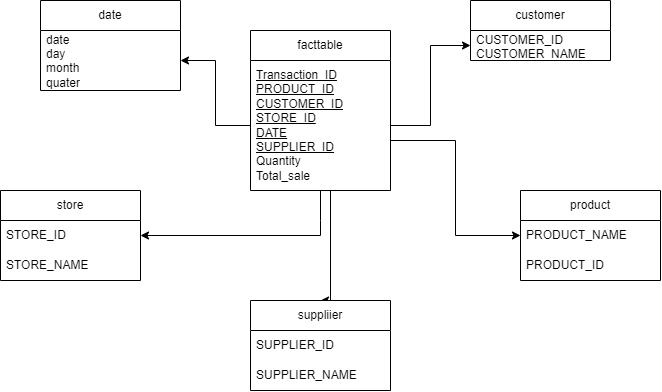
**Section: A**

**Roll# i190690**

**Project Overview:**

This project was based on new theory related to data warehouse. Which is called as near real time data warehouse where we load data to data warehouse very frequently and that causes issues, and how to tackle those issues is basically what mostly people find it interesting. We had to implement a near real time data warehouse prototype for a metro online shopping.

**Schema:**

****

**Mesh join:**

We had to implement mesh join algorithm which loads data stream and master data after joining and then we perform loading. This algorithm helps to create an ease to make a traditional warehouse to near real time data warehouse.

**Pseudo Code:**

Func MeshJoin():

    SET index to 0

    SET tempj to 0

    SET tempp to 0

    SET temp  to 0

    for i=1 to totalnumberoftransaction

        i=loadTransactionData(i)

        if(tempj equals to 0)

            j=0

        if(tempp equals to 0)

            p=0

        loadCustomerMasterData(p)

        loadProductMasterData(j)

        for x=0 to productMasterData.size

            product=productMasterData.get(x)

            maptoMatch=hashMap.get(product.product\_id)

            if(maptoMatch is not null)

                for a=0 to maptoMatch.size

                    check=false

                    maptoMatch.get(a).product\_name=product.product\_name

                    maptoMatch.get(a).supplier\_id=product.supplier\_id

                    maptoMatch.get(a).supplier\_name=product.supplier\_name

                    maptoMatch.get(a).Total\_sale=mapToMatch.get(a).quantity\*product.price

                    for u=0 to customerMasterData.size

                        customer=customerMasterData.get(u)

                        if(customer is not null)

                            if(maptoMatch.get(a).customer\_id equals to customer.customer\_id)

                                maptoMatch.get(a).customer\_name=customer.customer\_name

                                loadTuple()

        if(pointerQueue.size equals to 10)

            removehead()

        tempj+=10

        tempp+=50

    for i=1 to 9

        if(tempj equals to 0)

            j=0

        if(tempp equals to 0)

            p=0

        loadCustomerMasterData(p)

        loadProductMasterData(j)

        for x=0 to productMasterData.size

            product=productMasterData.get(x)

            maptoMatch=hashMap.get(product.product\_id)

            if(maptoMatch is not null)

                for a=0 to maptoMatch.size

                    check=false

                    maptoMatch.get(a).product\_name=product.product\_name

                    maptoMatch.get(a).supplier\_id=product.supplier\_id

                    maptoMatch.get(a).supplier\_name=product.supplier\_name

                    maptoMatch.get(a).Total\_sale=mapToMatch.get(a).quantity\*product.price

                    for u=0 to customerMasterData.size

                        customer=customerMasterData.get(u)

                        if(customer is not null)

                            if(maptoMatch.get(a).customer\_id equals to customer.customer\_id)

                                maptoMatch.get(a).customer\_name=customer.customer\_name

                                loadTuple()

        if(pointerQueue.size is greater than 0)

            removehead()

        tempj+=10

        tempp+=50

**Anomaly in dataset:**

The anomaly is that the transaction id is not giving the exact relationship with dates when the transaction took place. The transaction id is acting like a surrogate key, every new id would be generated when a transaction take place. The higher transaction id means a later date but that is not the case.

**Shortcomings in Mesh Join:**

* The algorithm reduces the access of disk to master data over a queue of stream tuples in memory.
* If the master data increase very large, the algorithm performance will be decreased.
* The stream can be infinite, and the algorithm has lesser removal of records from hash map.

**What did I learn?**

I learnt many different things about near real time data warehouse, which includes uses of threads in near real time warehouse, and uses of mesh join algorithm and handling of data, joining of data streams and table.