

**NAME: FAIZAN UL HAQ**

**SECTION: DS-M**

**ROLL: I21-1771**

**Project overview:**

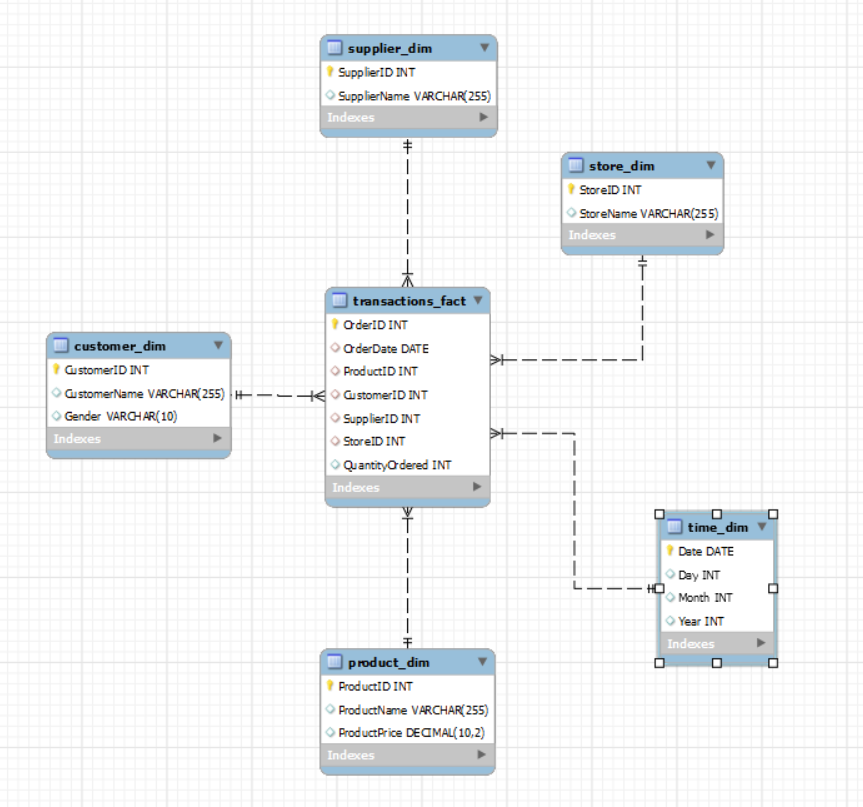
The purpose of this project was to efficiently load the data warehouse with the values form the databases in near realtime. We use the Hybridjoin invented by Dr.Asif Naeem to achieve this time result.

**TASK BREAKUP**

1. **Creating the Warehouse:**

**So to create the data warehouse we used the transactions dataset and the masterdata to take out their attributes and carefully design our datawarehouse. For the warehouse we used the starschema which was is very fast for the this specific datawarehouse as we don’t have to big table dimensions.**

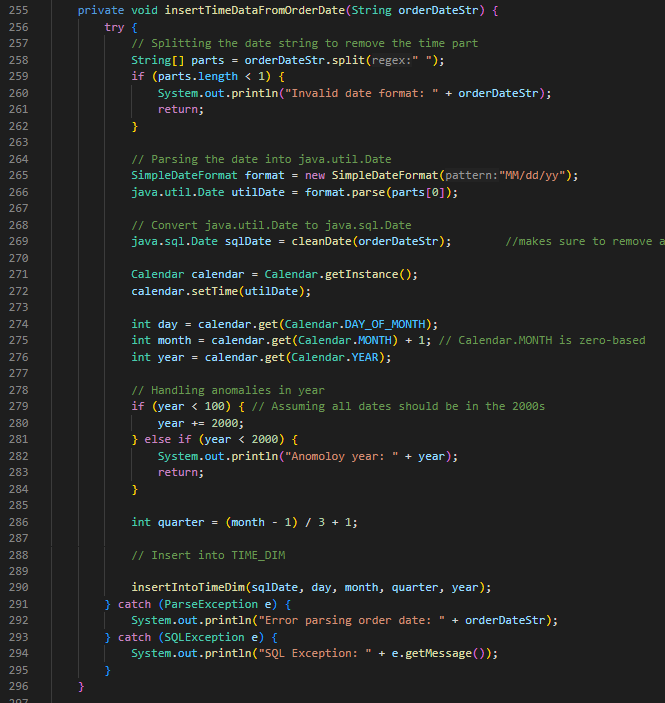
**The following is the star schema for our electronica-dw warehouse**

****

**The fact table incudes the following items:**

* **OrderID**: A unique identifier for each transaction.
* **OrderDate**: The date when the transaction occurred.
* **ProductID**: A foreign key referencing the **PRODUCT\_DIM** dimension table, representing the product sold.
* **CustomerID**: A foreign key referencing the **CUSTOMER\_DIM** dimension table, representing the customer who made the purchase.
* **SupplierID**: A foreign key referencing the **SUPPLIER\_DIM** dimension table, representing the supplier of the product.
* **StoreID**: A foreign key referencing the **STORE\_DIM** dimension table, representing the store where the sale took place.
* **Sales**: The amount of sales for each transaction.

1. **Loading date Dimension table in the form of calendar**

****

**As we can see lines 271 and onwards I created an object of calender, broke the date string into different parts such as day month year and quarter and then loaded it into the date variabls of the time\_dim**

**3) Implementing the HYBRIDJOIN algorithm using Java and successfully loading transactional**

**data into DW after joining it with MD.**

For the hybrid join part of the code, since we were required to implement this code using threading I created a shared memory. The shared memory comprises of a list of tuple data types. The tuple data type is my own class that consists of all the attributes in the transaction dataset. This way I send the list into both thread 1 for streaming and thread 2 for hybrid join.

The streaming then takes out the data from the database and then inserts it int the list tuple by tuple and once a certain amount of the list is ready it is sent into the hybrid join thread. The size of the list is controlled by the 3rd thread the controller thread.

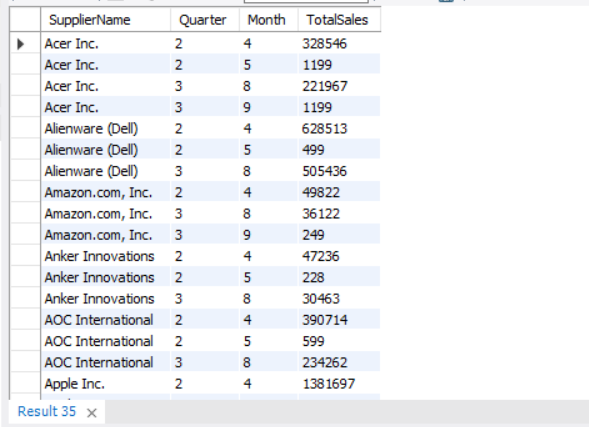
The hybrid join thread then takes the list, seperates each tuple and extracts the productID from it and send the productID into a doubly inked list queue and uses the same productID to hash the whole data into a ArrayListValuedHashMap hashtable the special thing about this hashtable is that it lets us hash multiple values to the same key. Thus this way the size of our hashtable remains smaller and more efficient.

Next we take out the front most key that is the productID from the linked list and use it to load a set of the masterData into a buffer. Once that is done we use the productID to compare the hashtable values with the masterData values and the onces that match are then processed and sent to the warehouse. The buffer for the master data is then emptied and the front most node of the linked list is also deleted.

The processing includes insertion into the specific dimension tables and fact table.

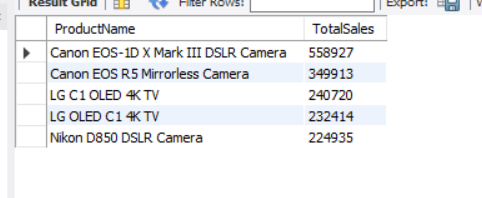
* 1. **OLAP QUERIES OUTPUTS**

Q1)

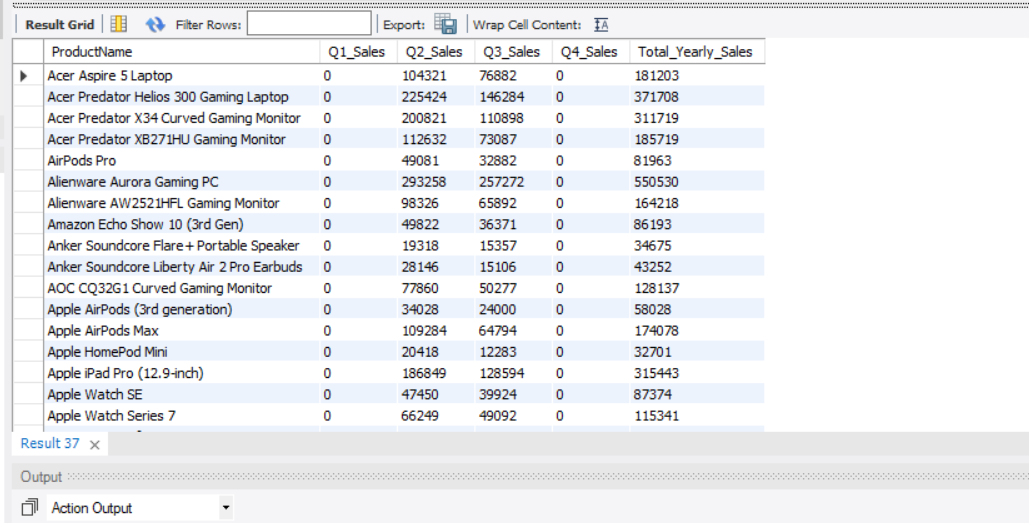


Q2) SKIPPED

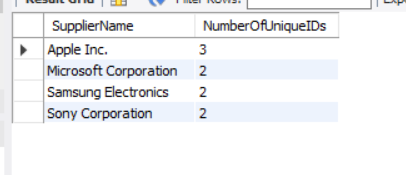
Q3)



Q4)

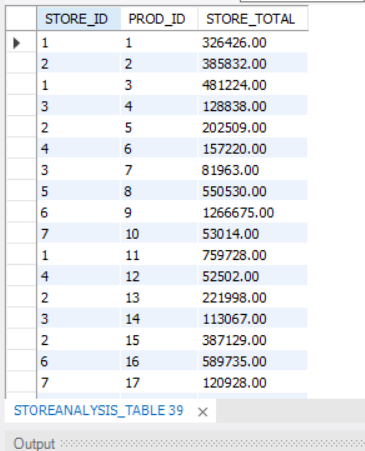


Q5)



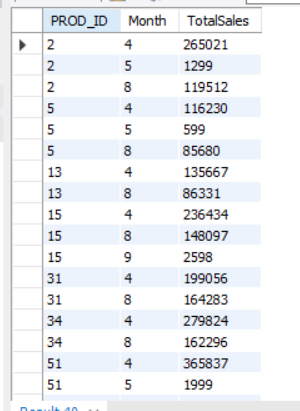
These are the anomalies In the data warehouse where a single supplier name has multiple supplier ids which should not be possible.

Q6)

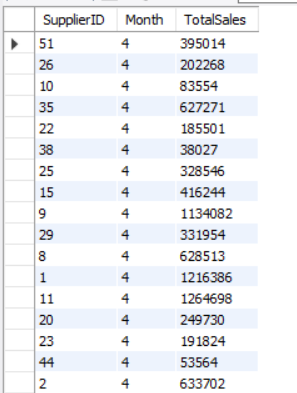


I created a normal view for this as materialized view cannot be made on mysql workbench

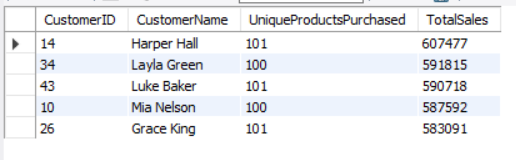
Q7)



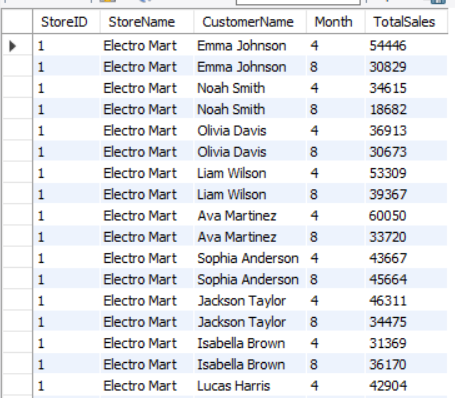
Q8)



Q9)



Q10)



**What did I learn from this project?**

This project was a really nice exercise to learn all about the working of a data warehouse and OLAP queries from roll up to slicing etc.

I learned how to deal with java and how to implement threading. I also learned how to effectively create a data warehouse that runs on near Realtime speeds.

**Shortcoming of hybrid join?**

The short coming of hybrid join in my opinion is that it takes a lot of storage space as we are essentially creating a doubly linked list and thus it includes more pointers. Eventhough the linked list is used to ensure that we get a hit from the comparision between masterdata buffer and hashtable however we are sacrificing more storage space for more speed.