**SEMESTER FINAL PROJECT FALL 2020  
DATAWAREHOUSING**

**Name:** Hamza Zafar

**Roll No:** 17i-0307

**Date:** November 11, 2020

**Course Instructor:** Prof Dr. Asif Naeem

# **DATAWEARHOUSE PROJECT OVERVIEW:**

This project started with a master data file containing records of thousands of transactions. This DDL file helped in the creating, designing and in analyzing the Metro shopping store in Islamabad.

## **Design or Extraction Phase:**

This movement outlines the reason of building a Datawarehouse. Being the most pivotal development, this movement is the finishing up stage to go either for Star Model or for Snowflake Model. For this circumstance, we will design the Datawarehouse with the Star Example as indicated by the need of this endeavor.

## **Implementing or Transformation-Loading Phase:**

This is the ensuing development, being the middle heart of the Datawarehouse. This movement consolidates the reasoning and key technique behind the working of your Datawarehouse. This movement concludes how to stack the data, when to stack the data and, in what amount to stack the data. This has been done using **Java SE 8 in Eclipse IDE**. The backend figuring used in this endeavor is **Hybrid Join Algorithm**.

## **Datawarehouse Analysis Phase:**

Last anyway not the least the fundamental development after the execution time of Data warehousing is analyzing the Datawarehouse. This movement outfits an information with respect to the business. Over the Data Stockroom various inquiries in **MySQL** are ran which gives the reason of uncovering, customer encounters and outlining a business understanding model for the association at whatever point required.

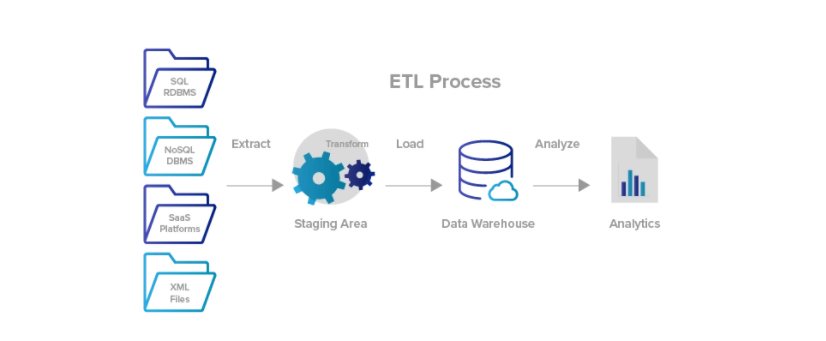
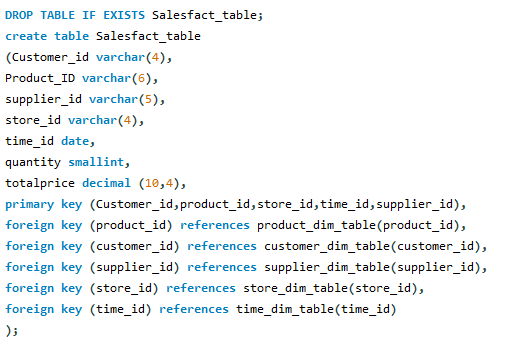
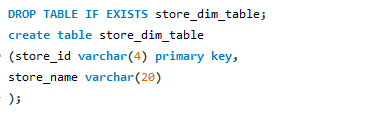
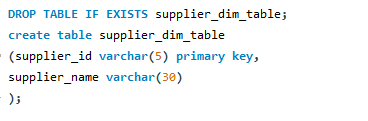
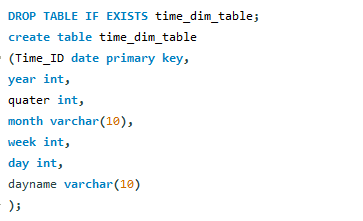
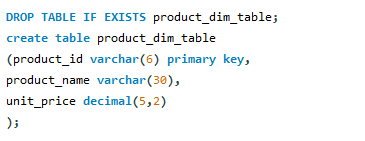
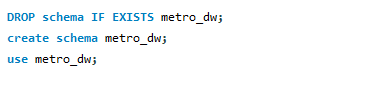


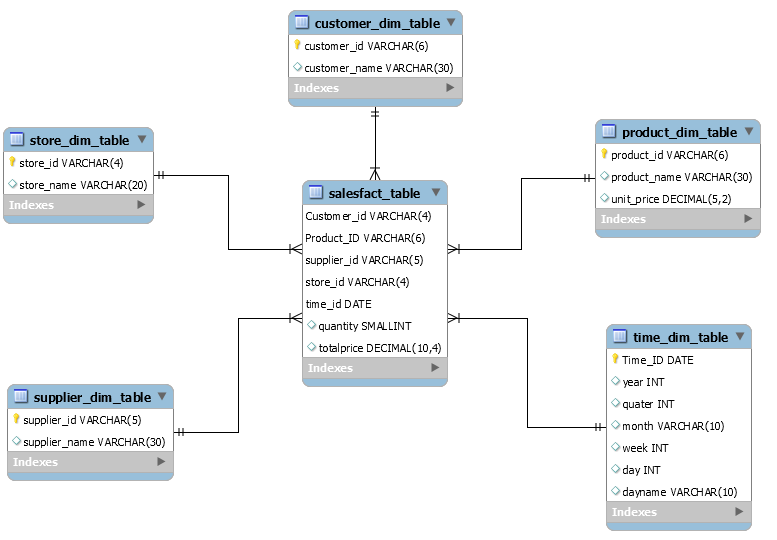
Fig 1 Overall ETL Overview of DWH Design, Implementation and Analysis.

# **SCHEMA FOR DW:**

**Creating a Star Schema :**



# **STAR SCHEMA FOR DW:**



# **HYBRID JOIN ALGORITHM:**

## **WHAT IS HYBRID JOIN ALGORITHM:**

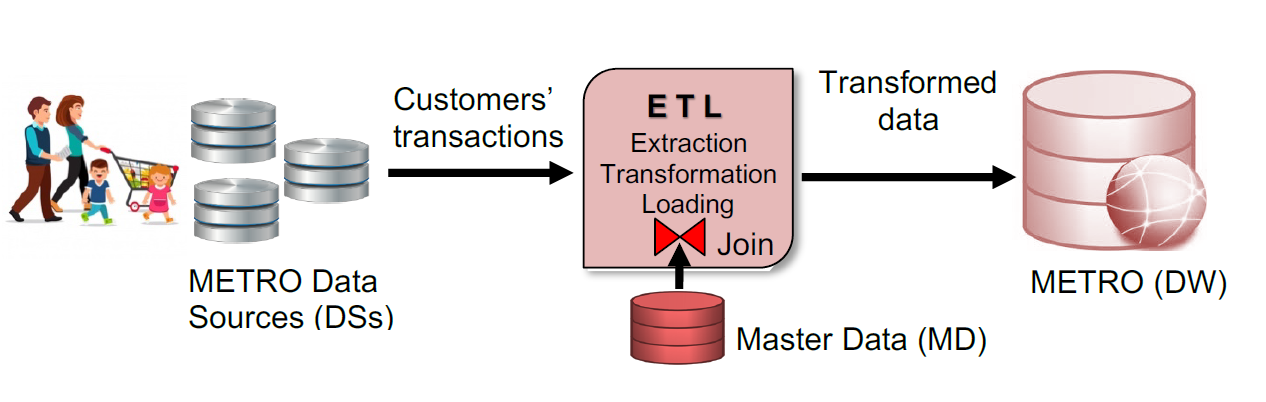
The HYBRIDJOIN (Hybrid Join) algorithm has been introduced by Naeem et. al. in 2011 with objective of implementing the join operation in the transformation phase of ETLThe main components of Hybrid JOIN are as follows:

**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.

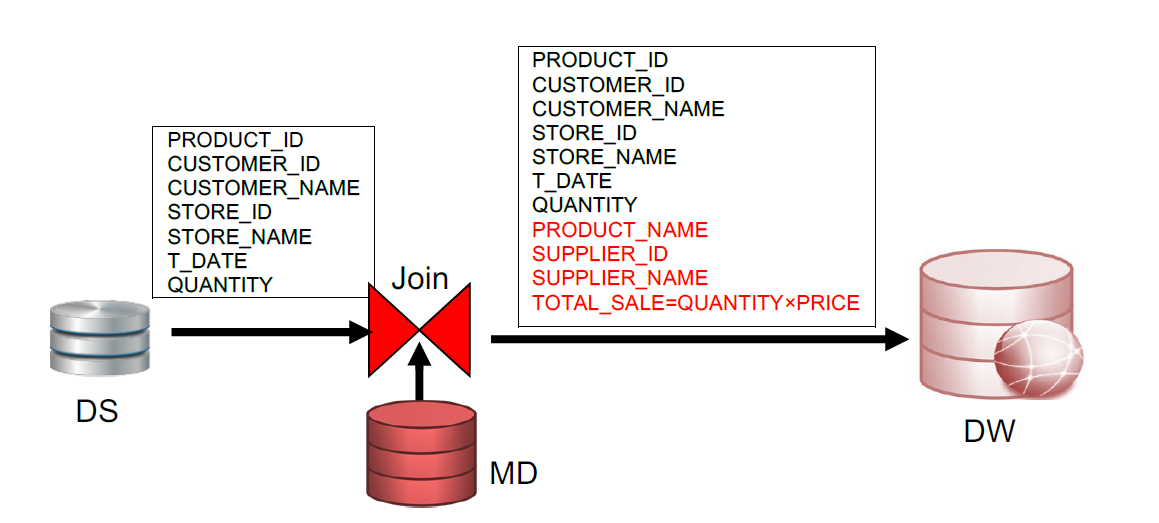
**The hash table** which stores the customers’ transactions (tuples)**.**

**The queue** is used to keep the record of all the customers’ transactions in memory with respect to their arrival times. The queue has 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. However, you don’t need the stream buffer in this project as we are not considering the stream of customers’ transactions.

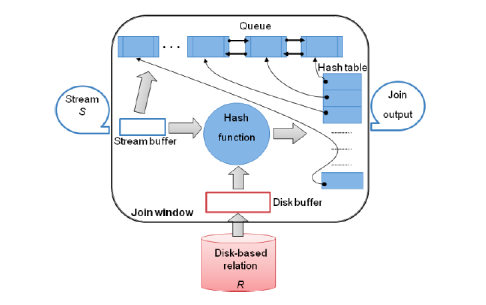


**Fig 2.0 Overview of Metro Store DWH**



**Fig 2.1 Enrichment of Metro Store DWH**

The crux of Hybrid Join is that with every loop step a new input chunk of customers’ transactions is read into main memory (the Hash table and the Queue) from DS. The size of the input chunk depends on the number of tuples deleted from the Hash table in the previous transaction. Also for every loop step a segment of MD tuples is loaded into the disk-buffer using join attribute value stored in the queue as an index. The MD tuples loaded in the disk buffer are then probed into the multi-hash table. If the tuple matches in the multi-hash table then the matched record is joined with the relevant MD tuple and the join output is produced. The matched tuple is then removed from the multi-hash table and the queue. Finally, the joined output tuple is loaded to DW “metro\_dw”.



**Fig 2.2 Execution Architecture of HYBRIDJOIN**

**IMPLEMENTATION OF HYBRID JOIN JAVA CODE:**

HybridJoin implementation all three phases of ETL –extraction of records from TRANSACTIONS table, transforming them with MD(masterdata) and then loading these records to DW “metro\_dw” successfully.

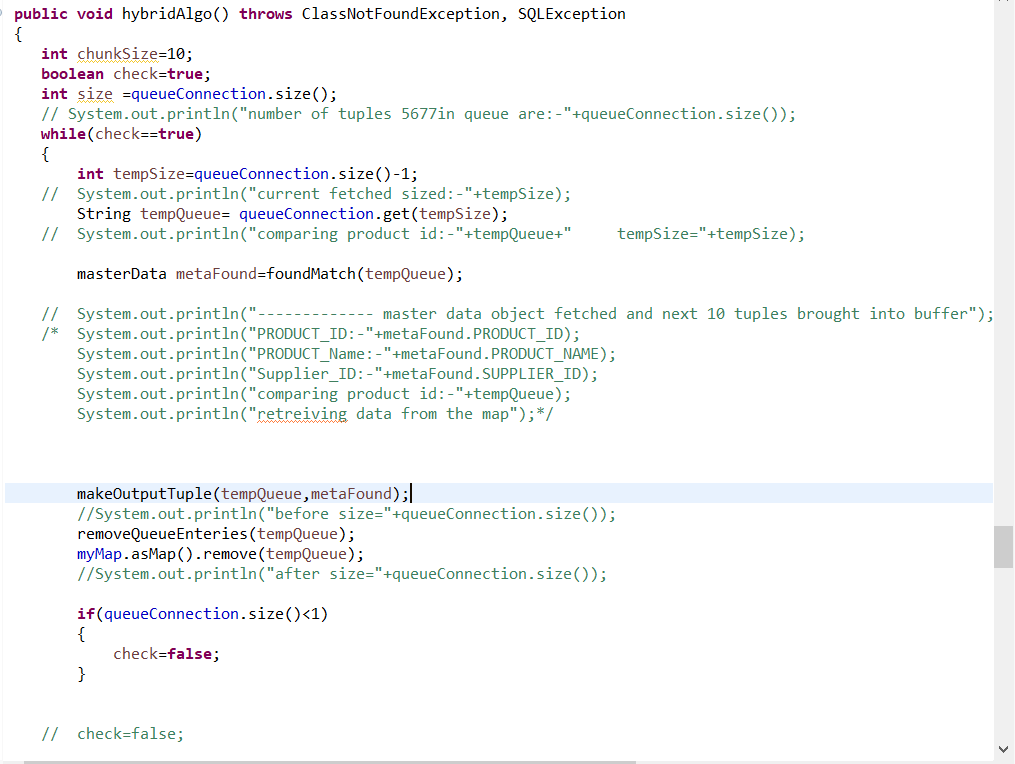
**LOADING THE DATA IN WEARHOUSE “metro\_dw”:**

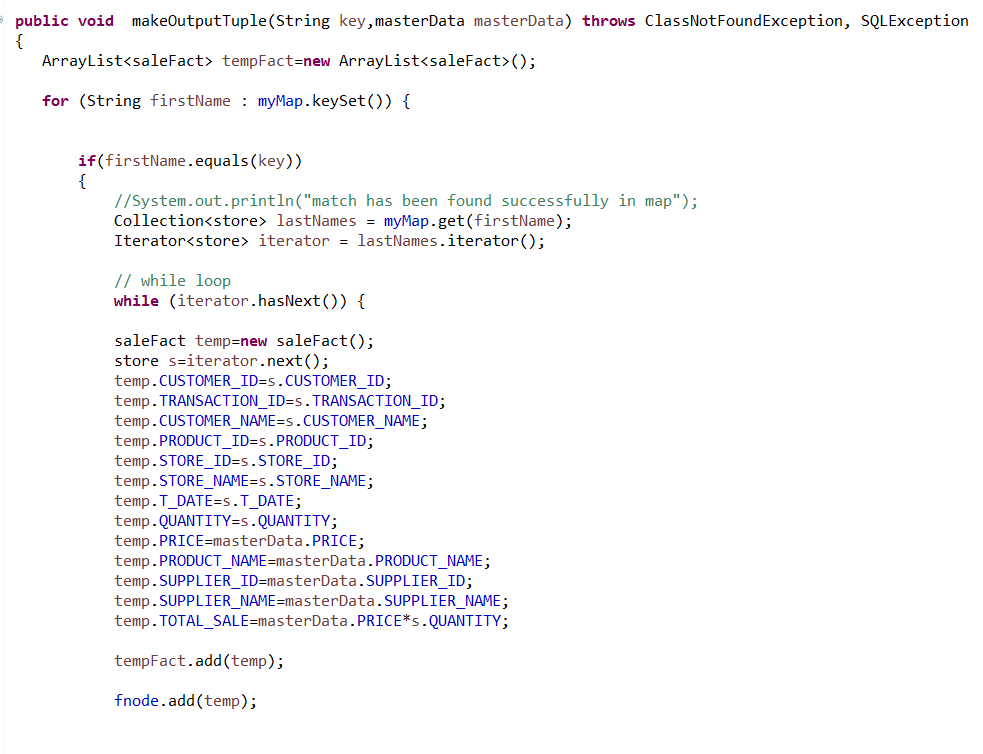
****

****

****

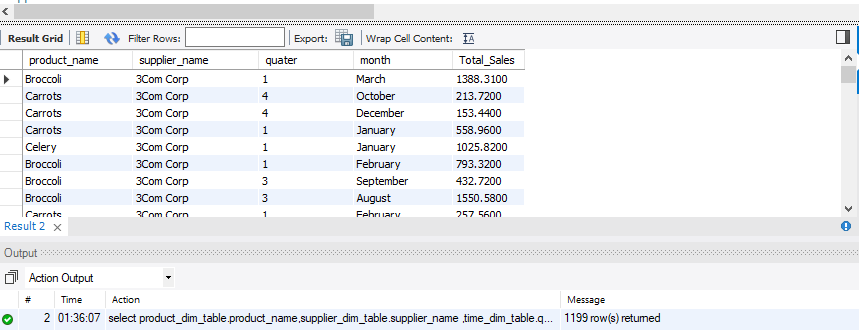
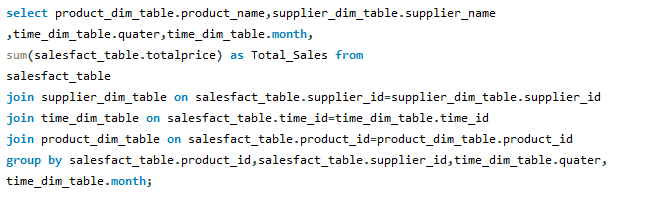
****

****

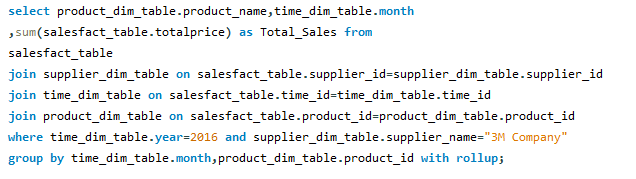
****

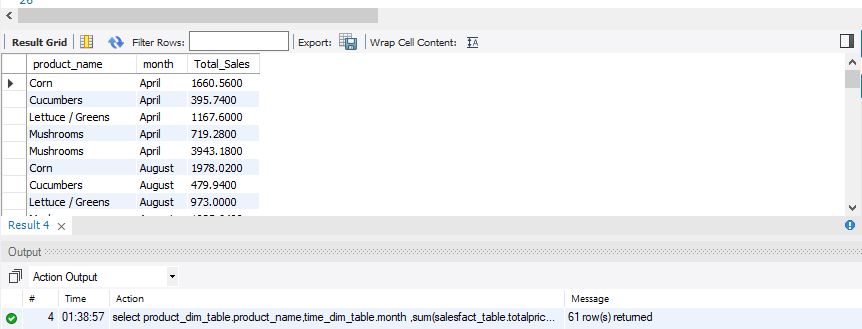
# **OUTPUT OF OLAP QUERIES:**

Query No. 1: Present total sales of all products supplied by each supplier with respect to quarter and month using drill down concept.

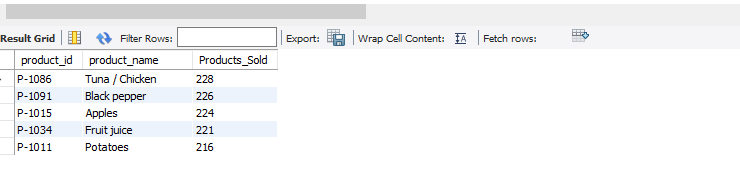
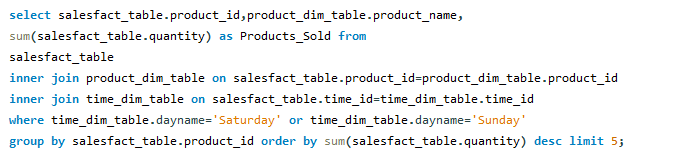


Query No. 2: Find total sales of product with respect to month using feature of rollup on month and feature of dicing on supplier with name "3M Company" and Year as "2016". You will use grouping sets feature to achieve rollup. Your output should be sequentially ordered according to product and month.

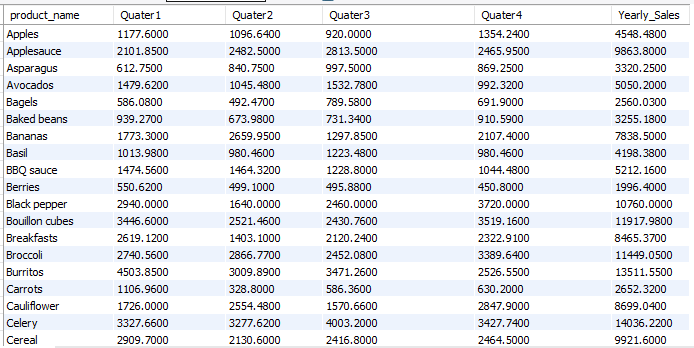
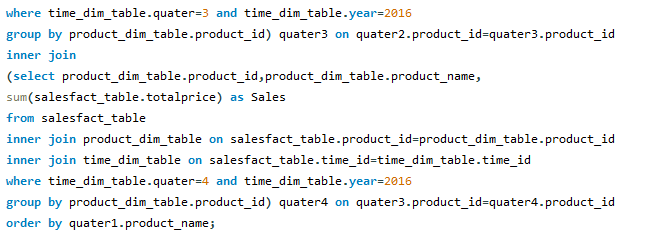
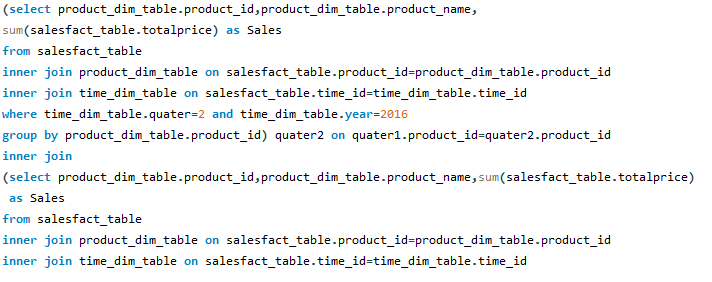
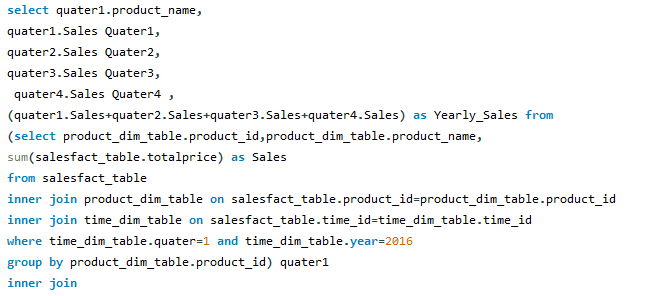




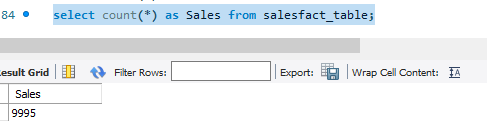
Query No. 3: Find the 5 most popular products sold over the weekends.



Query No. 4: Present the quarterly sales of each product for year 2016 along with its total yearly sales. Note: each quarter sale must be a column and yearly sale as well. Order result according to product



Query No. 5: Find an anomaly in the data warehouse dataset. write a query to show the anomaly and explain the anomaly in your project report.

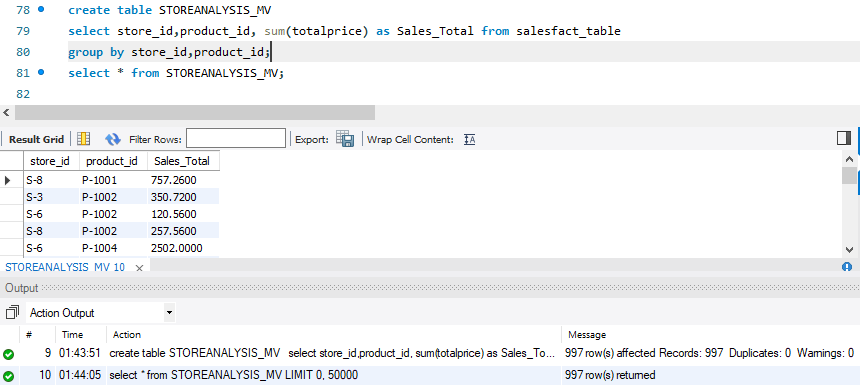


This query shows that after implementation of hybrid join and loading data into the wearhouse we get 9995 records in total. This means we have 9995 unique records in the salesfact\_table. The transaction table had 10000 records while we got 5 records less. This shows that the transaction table had a duplication of 5 records which were removed on applying the primary keys in the salesfact\_table. This occurred due to, like if a customer done a transaction of some product and then on the same day he came again to buy the same product and everything same except the transaction id, now in this case the record was all same except the transactionid which is different for both transactions. The data attribute was also same. If we had used the timestamp then that would have differentiated the both records but as we only had date so we don’t know the time from that. So due to the duplication of these records, which were removed when we applied the primary key in salesfact\_table we got 9995 rows in total.

Query No. 6: Q6 Create a materialised view with name “STOREANALYSIS\_MV” that presents the productwise sales analysis for each store.

STORE\_ID PROD\_ID STORE\_TOTAL

------------- ------------ -------------------



# **SHORTCOMINGS OF HYBRID JOIN:**

Near-real-time data warehousing plays an essential role for decision making in organizations where latest data is to be fed from various data sources on near-real-time basis. The stream of sales data coming from data sources needs to be transformed to the data warehouse format using disk-based master data. This transformation process is a challenging task due to **slow disk access** rate as compare to the fast stream data. For this purpose, an adaptive semi-stream join algorithm called HYBRIDJOIN (Hybrid Join) is presented in the literature. The algorithm uses a **single buffer** to load partitions from the master data. Therefore, the algorithm has to wait until the next disk partition overwrites the existing partition in the buffer. As the cost of loading the disk partition into the buffer is a **major cost in the total algorithm’s processing cost**, this leaves the performance of the algorithm **sub-optimal**.

# **LEARNING OUTCOME:**

This project has helped me understand how ETL is done for any Datawarehouse. This is helped me a lot in Excelling in this area of Field. This project helped me in understanding the model and structure and actually how the wearhouse is made. The process which produces a wearhouse. This project was helpful in understanding the ETL for building a wearhouse. Further it provided me the knowledge that how to connect our java code with the database and Understanding of structure and implementation of OLAP queries for the wearhouse analysis.

# **REFERENCES:**

* <https://aisel.aisnet.org/confirm2019/27/>
* <https://panoply.io/data-warehouse-guide/etl-tutorial/>
* https://geekforgeeks.com