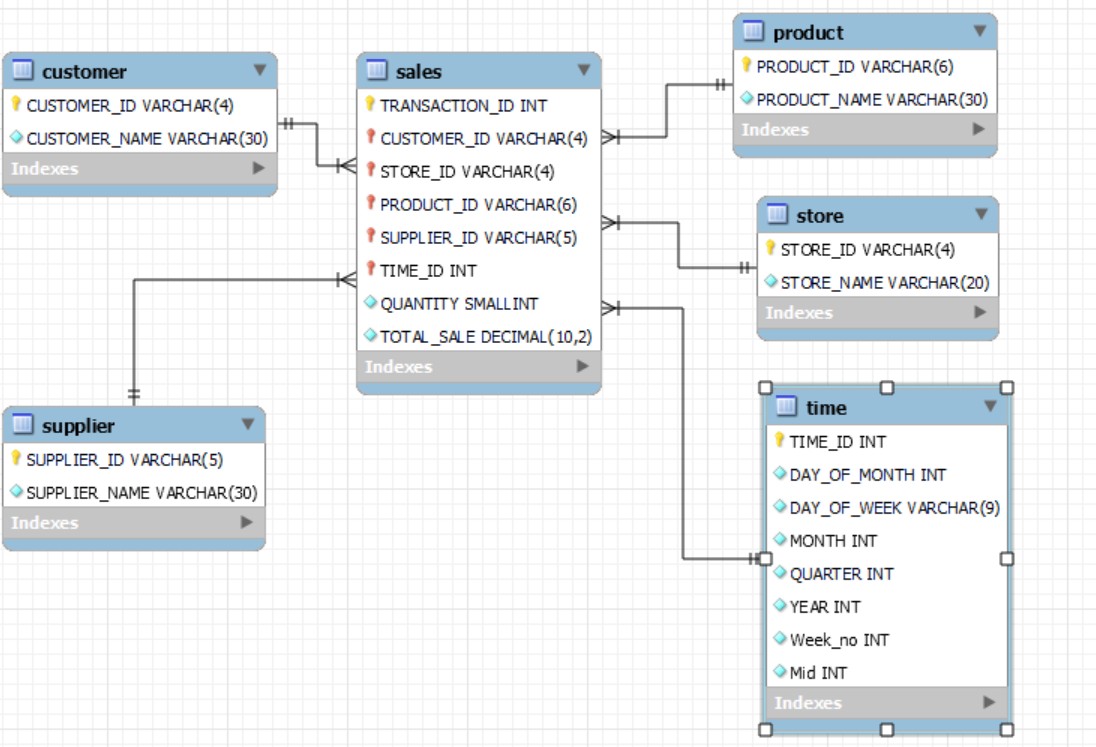
# Project Overview

This project is to design and create such a data warehouse system Metro Pakistan | Biggest Wholesale & Supermarket 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 techniques e.g. giving promotions on different products. Due to the data from one-year transaction data.Transaction table 10000 records and master table 100 products. Program write for Extract Transform Load (ETL) in JAVA. When joining the transaction data and master data in the Meshjoin algorithm. The Start schema is used to model the data structure in the data warehouse first design the star schema. The start schema has five dimensions customer, store, Time, Product, Supplier, and Fact table Sale have two measures Total Sold and Total\_Sale on the basis of the transaction. After Extracting and Transforming the data then loaded the data in the Data warehouse. Perform the Online Analytical Processing (OLAP) query.

# Schema for DW

The Start schema is used to model the data structure in the data warehouse. Star schema has five dimensions customer, Store, Product, Time, and Supplier. Fact sale table has two-measure Total sold and Total Sale.



# MESHJOIN Algorithm

The program dynamically loads the partition size and the stream partition’s records size. The partition size of Meshjoin-partition is 10 and the stream partition size is 50. Add transaction records in an input queue. Enqueue the partition when it gets full. Add all new records into the record list. Load the next master partition data. Compare every record product id with master partition product id. Put a new record into the output queue. Last dequeue the last.

## Shortcomings in Mesh Join

1. Meshjoin depends on the partitions for the internal queue for The stream data and the number of changes needed for the disk based relation In memory . This dependency hampers the optimal distribution of memory between the Join components. Normally the size of the disk-buffer differs with the size of the disk-Based relation and its not necessary.
2. Stream-based algorithm Mesh Join (meshjoin) has been proposed to amortize disk Access over the fast stream. Meshjoin makes no assumptions about the data distribution. In Actual world applications, however, skewed distributions can be found, e.g, some products are sold more often than the remaining products. The question that comes to mind is, how much does the performance of mesh join loose by not adapting to data Skew.

3 The main problem in stream-relation joins is the different nature of inputs; stream data is Fast and busty, where the disk-based relation is not as fast because of the high disk I/O cost.

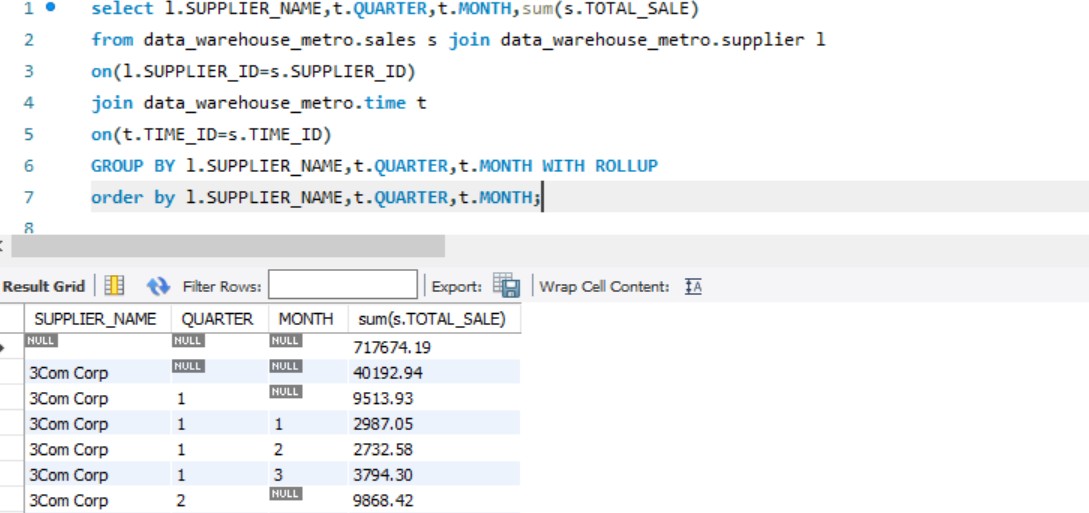
## Lessons Learned

* I learned java language programming and Logic builds Extraction Transform and Load (ETL) import data into Data warehouse.
* A real-time ETL program is to be implemented in reality.
* I learned OLAP querys Roll up , drill down

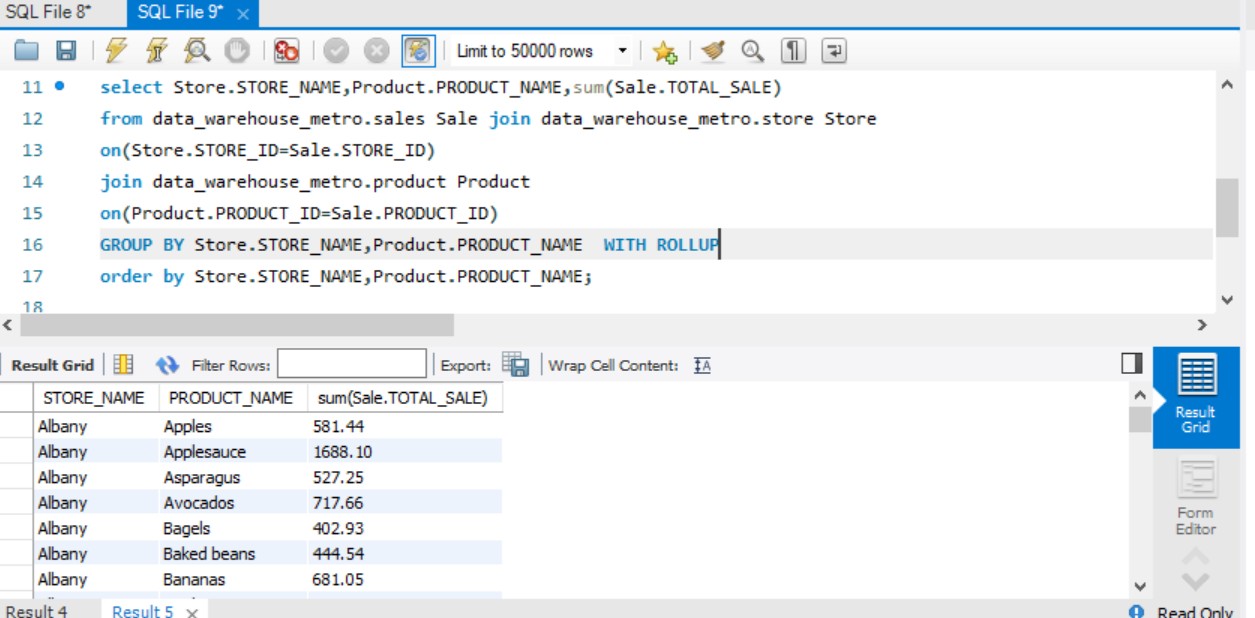
## DW analysis

**Q1 Present total sales of all products supplied by each supplier with respect to quarter and**

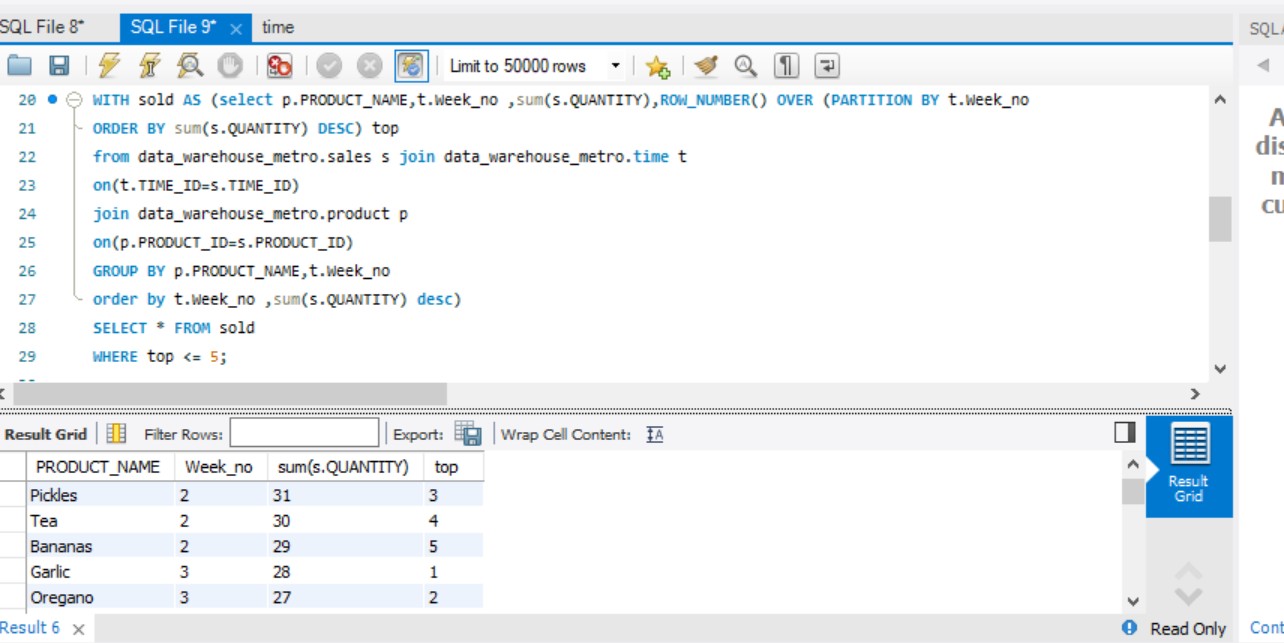
**month**



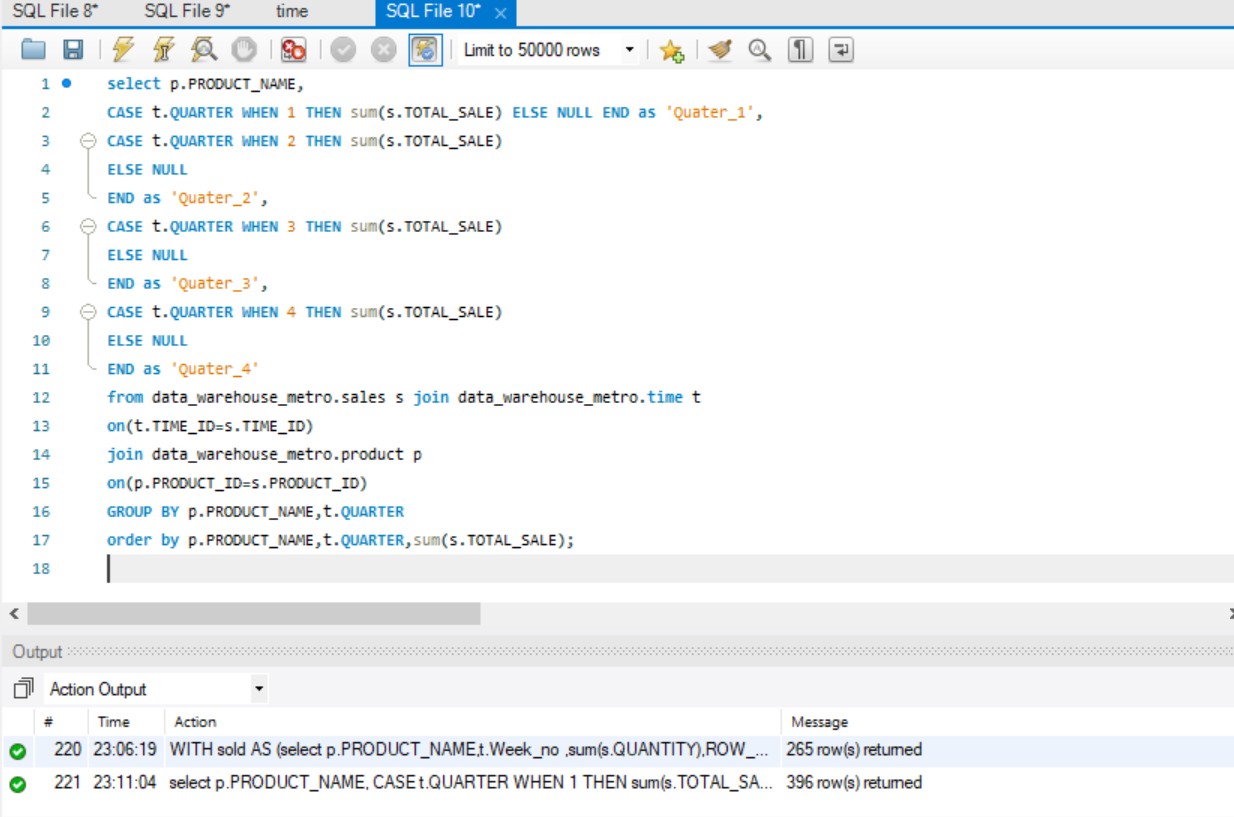
**Q2 Present total sales of each product sold by each store. The output should be organised store wise and then product wise under each store.**

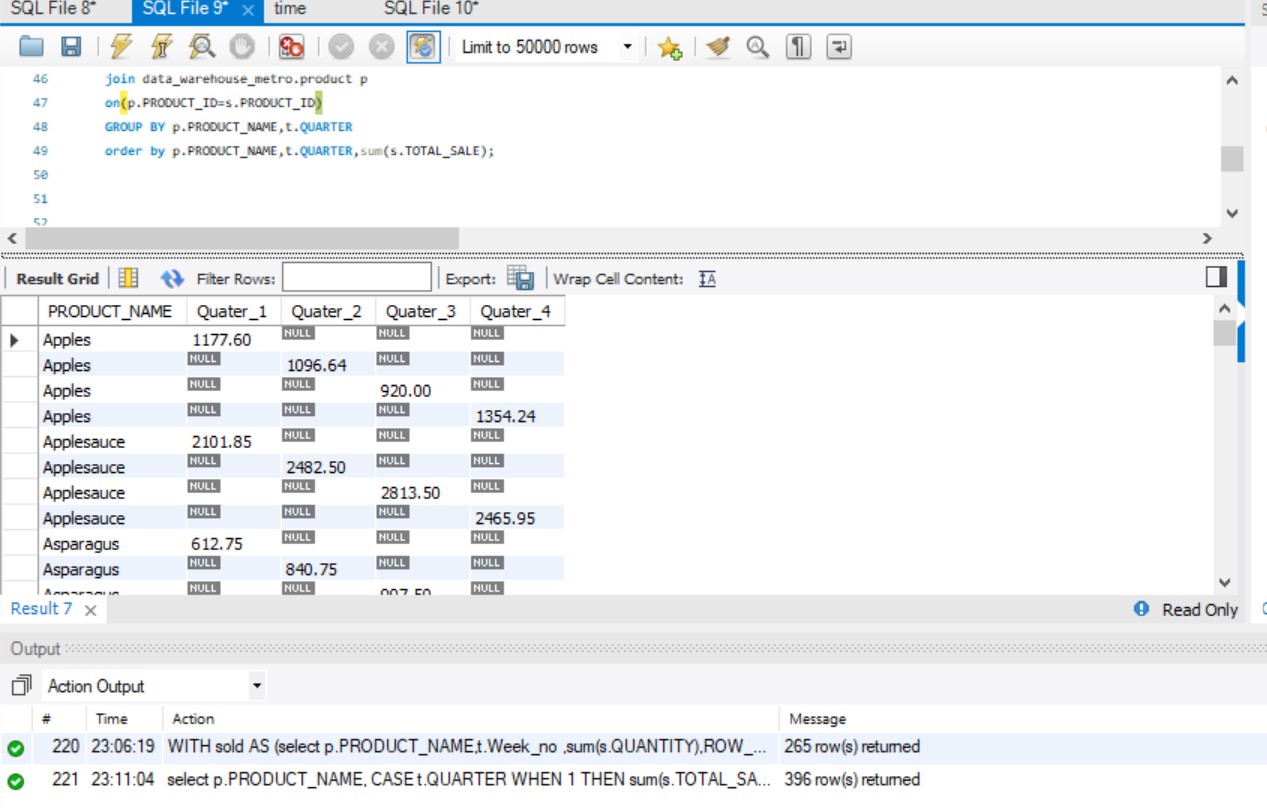


**Q3 Find the 5 most popular products sold over the weekends.**

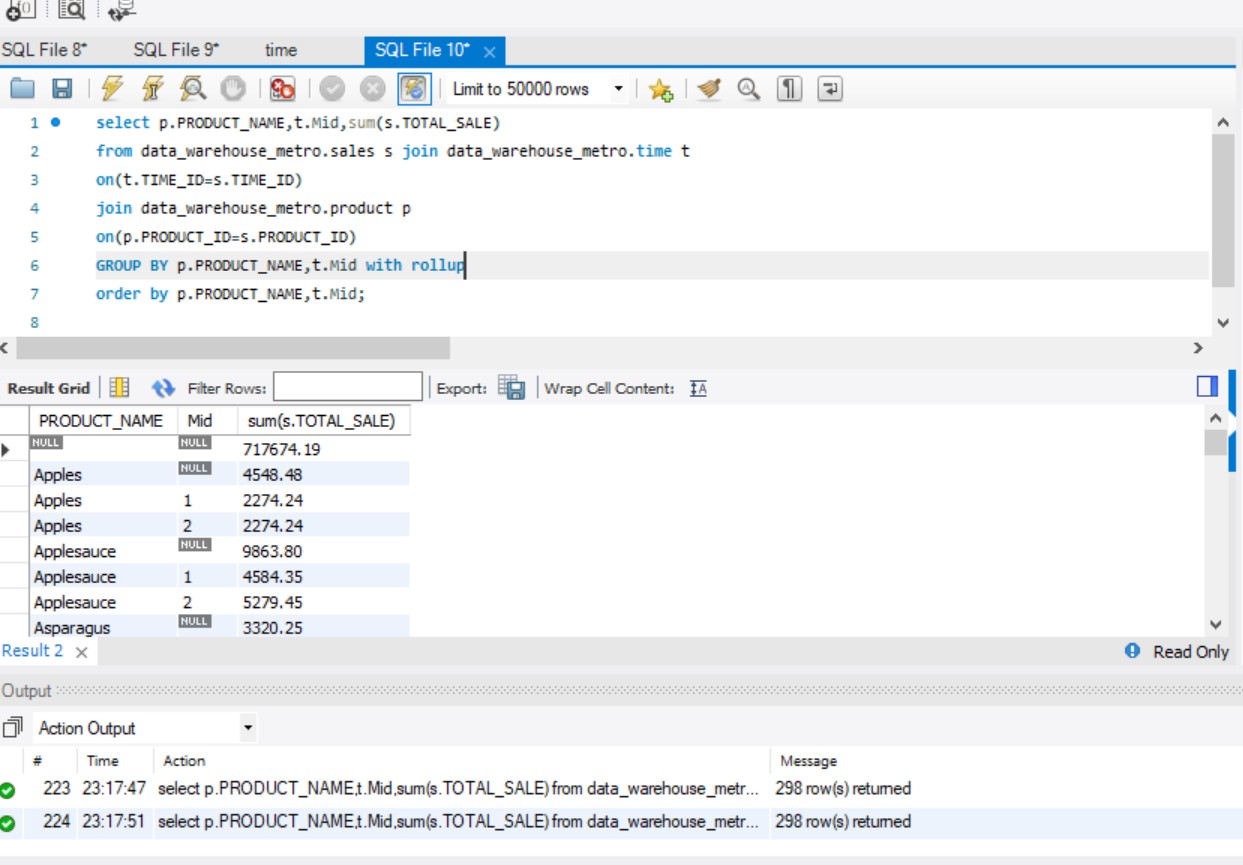


**Q4 Present the quarterly sales of each product for year 2016 using drill down query concept. Note: each quarter sale must be a column.**

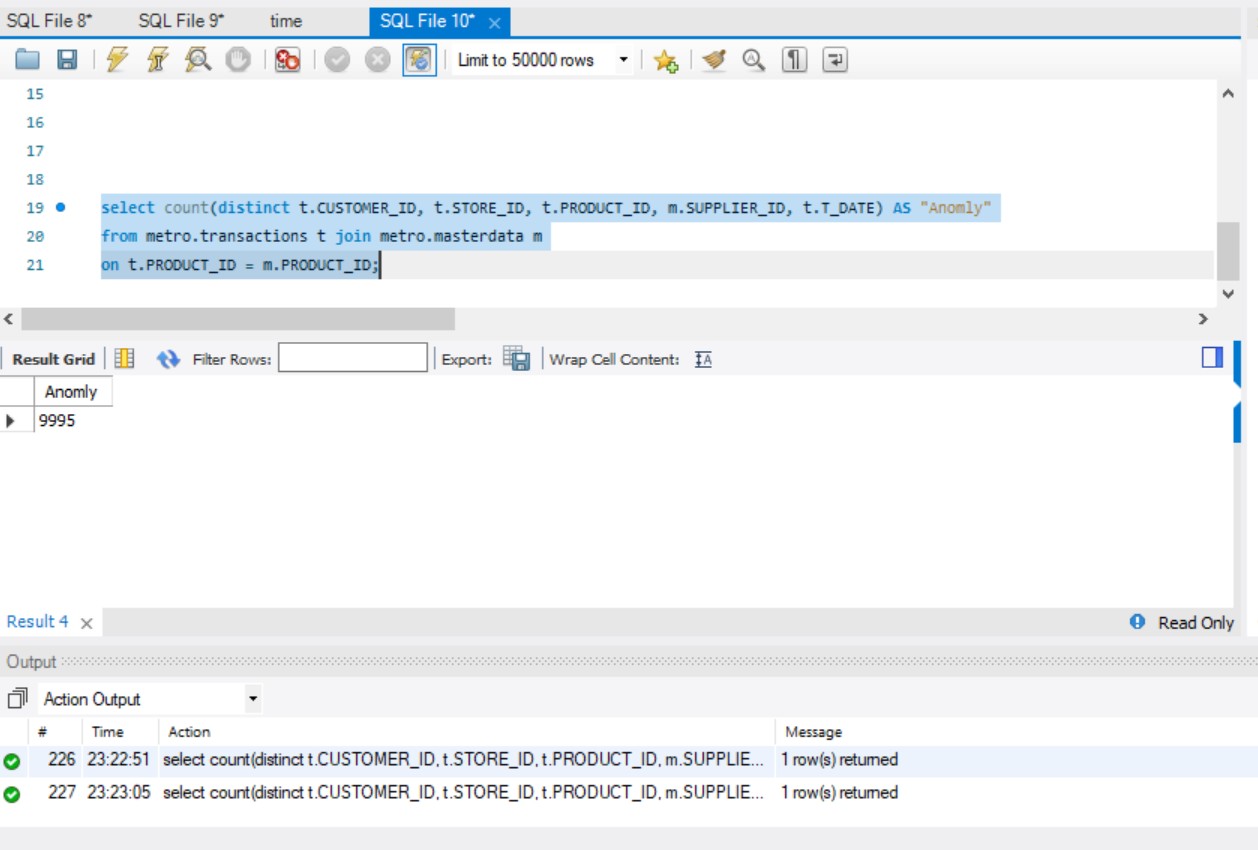




**Q5 Extract total sales of each product for the first and second half of year 2016 along with its total yearly sales.**



**Q6 Find an anomaly in the data warehouse dataset. write a query to show the anomaly and explain the anomaly in your project report.**



**Q7 Create a materialised view with the name “STOREANALYSIS\_MV” that presents the product- wise sales analysis for each store.**



SQLF,leS- SQLF,le9" time



**Q** 'ifirA Limit to **50000** rows • 0.. [ID l-'l

29 • CREATE VIEW STOREAI

38 SELECT s.store\_id STORE\_IO, p. product\_id PROO\_IO, SU1>1(l. total\_sale) S ORE\_TOTAL

31 **froa** data\_wareh.ouse\_metro.store s, data\_wareh.ouse\_metro.sales l,data\_wareh.ouse\_metro.product p

32 where s.store\_id = l.store\_id and l.product\_id = p.product\_id

33 GROUP by s.store\_na e, p.product\_naae

34 ORDER by store\_na e, product\_na e;

35

36 • select • **fr011** data\_l'!arehouse\_roetro.STOREAI LYSIS

<

I **Result Grid** I

►

**-t\** FilterRows:---- IExport: I**Wrap**Content:

ST0REANAL YSJS 5 X

|  |  |  |
| --- | --- | --- |
| STORE\_ID | PROD\_ID | STORE\_TOTAL |
| S-2 | P-1015 | 581.44 |
| S-2 | P-1080 | 1688.10 |
| S-2 | P-1000 | 527.25 |
| S-2 | P-1016 | 717.66 |
| S-2 | P-1030 | 402.93 |
| S-2 | P-1081 | 4-H.54 |
| S-2 | P-1017 | 681.05 |
| S-2 | P-1090 | 377.10 |

Output

oil Action Ouput

limo **Action Mt-.nage**

**0** 237 23:30:24 CREATE VIEW STOREANALYSIS AS SELECTs.store\_id STORE\_ID. p.produclJ ... 0IQw(s)affected

**0** 238 23:30:24 select "fromdata\_warehouse\_metro.STOREANALYSISLIMIT 0. 50000 987 row(s)retuned

**0** Reado