# **PROJECT REPORT**

# **Anas Zafar**

# **19i-1676**

## GOALS OF THE PROJECT

* Build a data warehouse prototype for transactional data of Metro
* Learn the implementation of ETL techniques for mapping data from database into data warehouse
* Perform OLAP queries for data analysis on the built data warehouse

## TASKS PERFORMED TO ACHIEVE THE GOALS

### Identifying the facts and dimensions

* Identify the facts and dimensions in the given data and list them down
* Define granularities for the chosen dimensions referring to the business questions asked
* Identify the measures for calculating the Total sales

The following Facts and Dimensions were identified in the project with granularities

**Facts:**

1. Sales
2. Quantity

**Dimensions:**

1. Customer Dimension
2. Store Dimension
3. Supplier Dimension
4. Product Dimension
5. Time Dimension
   1. Year
   2. Quarter
   3. Month
   4. Date of Month
   5. Date (unique)

The Sales fact value is calculated as a product of Quantity sold in transaction and Unit price of the product

Total\_Sales=Quantity\*Unit price

### Defining Attributes for Facts and Dimension Tables

The fact and dimension tables are composed of following attributes

**Dimensions:**

1. Customer
   1. Customer id (Unique)
   2. Customer name
2. Store
   1. Store id (Unique)
   2. Store name
3. Supplier
   1. Supplier id (Unique)
   2. Supplier name
4. Product
   1. Product id (Unique)
   2. Product name
   3. Unit price
5. Time
   1. Year (Unique)
   2. Quarter
   3. Month
   4. Date of Month
   5. Date (full date, unique)

**Fact:**

1. Fact\_Table(Table)
2. Customer id
3. Product id
4. Supplier id
5. Store id
6. Date
7. Quantity
8. Sales

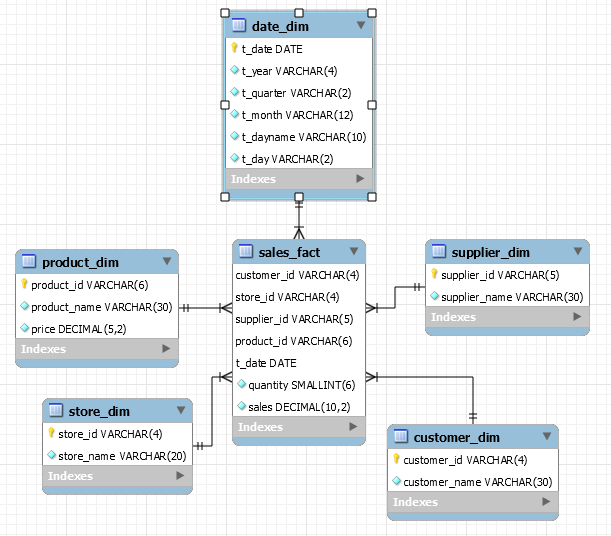
A materialized, view with following attributes, is also created as demanded in the problem statement

* Store Analysis Materialized View

1. Store id
2. Product id
3. Total sales

This view table contains the product wise total sales of each store

### Star Schema



## Extraction, Transformation and Loading processes

|  |  |
| --- | --- |
| Transactions | |
| Attributes | Methods |
| transaction id  product id  customer id  customer name  store id  store name  date  quantity | * Transactions () * Parametrized Constructor |

|  |  |
| --- | --- |
| Master | |
| Attributes | Methods |
| product id  product name  supplier id  supplier name  double price | * Master() * Parametrized Constructor |

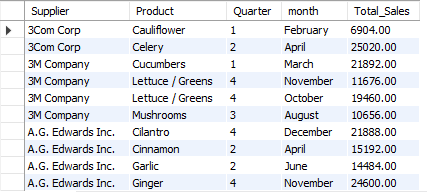
|  |  |
| --- | --- |
| Fact\_Table | |
| Attributes | Methods |
| transaction id  customer id  customer name  store id  store name  product id  product name  supplier id  supplier name  date  double price  quantity | * Fact\_Table() * Parametrized Constructor |

|  |  |
| --- | --- |
| Mesh\_Join | |
| Attributes | Methods |
| none | * Public Static Void Main (…)   Implements Mesh\_Join |

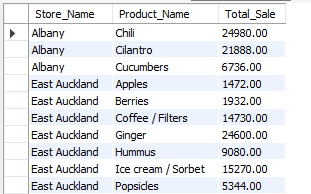
## Queries’ Results

The snaps below are the piece of results of the respective queries

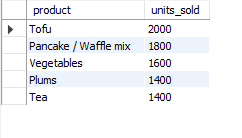
**Query 1**



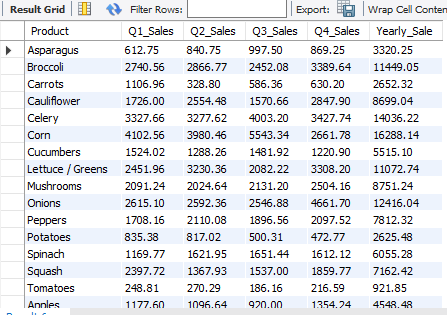
**Query 2**



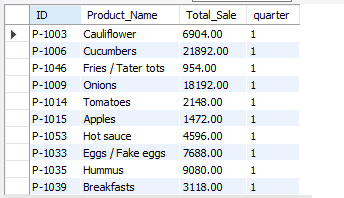
**Query 3**



**Query 4**

****

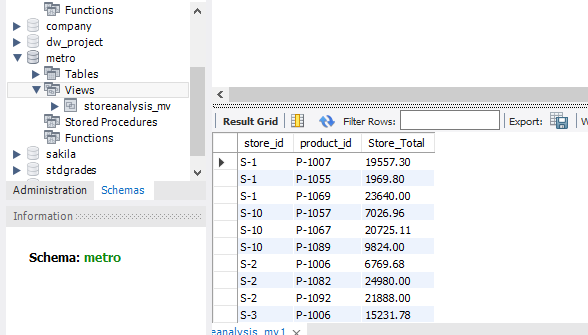
**Query 5**



**Query 6**

* There is Transition Id Given ,Which is of no need in data ware house, that's just a burden in data or adding just more space.

**Query 7**



## Shortcoming of Mesh\_Join and learnings from project

### Shortcomings

* In MESHJOIN there is a dependency between the size of partitions in an internal queue for the stream data and the number of iterations required to bring the disk-based relation into memory. This dependency hampers the optimal distribution of memory among the join components. In particular the size of the disk-buffer varies with the size of the disk-based relation which is unnecessary.
* Queue Size is limited ,so that if stream is much bigger than due to fixed size ,waiting Time increases
* Space inefficiency. It requires much space to process large data in using Queue and hash tables and result sets.
* In Queue, there is only one way back,from end of queue, so if the data is inserted wrong or some mistake is done you have to remove all the data till to reach that particular set.

### Learnings

* How to identify a warehouse’s components from the database data
* How to design the structure of a data warehouse (the star schema)
* connect java code to MySQL database .
* How to perform ETL operations on database .
* How to use an ETL technique and how usually ETL techniques perform
* How to use a data warehouse for data analysis through OLAP queries