

# **Sales Insights Data analysis Project**

Report is created for fulfilment of Data Analysis project  
Using SQL and PowerBI

By

Name: Vicky Kumar Singh

## **Table of content-**

1. Problem Statement
2. Data Discovery
3. Data Analysis using SQL
  - I. Importing Data to MySQL workbench
  - II. Simple analysis of data by looking into different tables and reflecting garbage values
  - III. Primary analysis of data base by running different SQL statements
4. Data Cleaning and ETL (extract transform load)
  - I. Connect MySQL with the PowerBI desktop
  - II. Loading the data into the PowerBI desktop
  - III. Transforming data with the help of Power Query
5. Building a Dashboard or a Report
6. References

## **1. Problem Statement-**

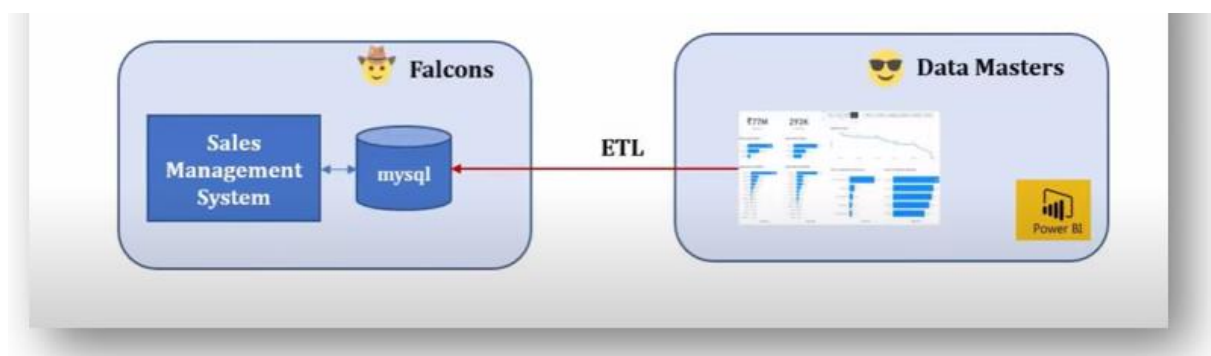
AtliQ hardware is a company in India which supplies computer hardware and peripheral devices across India only. They have many stores across India such as surge stores, Nomad stores etc. The head office of the company is situated in Delhi.

Scenario —The sales manager of the company is facing many challenges. He is facing issues in tracking sales in a dynamically growing market. He is having issues with the insights of his business. In order to this he has some of the regional managers in North, south and central India working for the company. So, he calls them and asks about the insights he wants to know. They tell him about the sales in last quarter and the growth in that quarter. So, the problem is that the conversations that are happening are verbal. Hence, the regional managers are sugar coating the facts and the manager of the company does not get the clear picture of the facts. Even after knowing that the sales are declining, he cannot do anything because he does not have the clear picture of the sales. Asking for the records the regional manager provides him with excel files. But by this he cannot figure out small things. All what the manager wants is a view of the weakest area the company needs to focus to increase the sales and improvise the declination. He is interested in simple, understandable and digestible insight. So, he is more interested in a dashboard which he can go and look at the real data because data speaks the truth. All he wants is a simple data visualization tool which he can access on a daily basis. Hence, by using such tools and technology one can make data driven decisions which helps to increase the sales of the company. So, in this project we will help a company make its own sales related dashboard using PowerBI.

**2. Data Discovery-** There is a team of software engineers (falcons) which owns sales management system. The records of this system are stored in MySQL database.

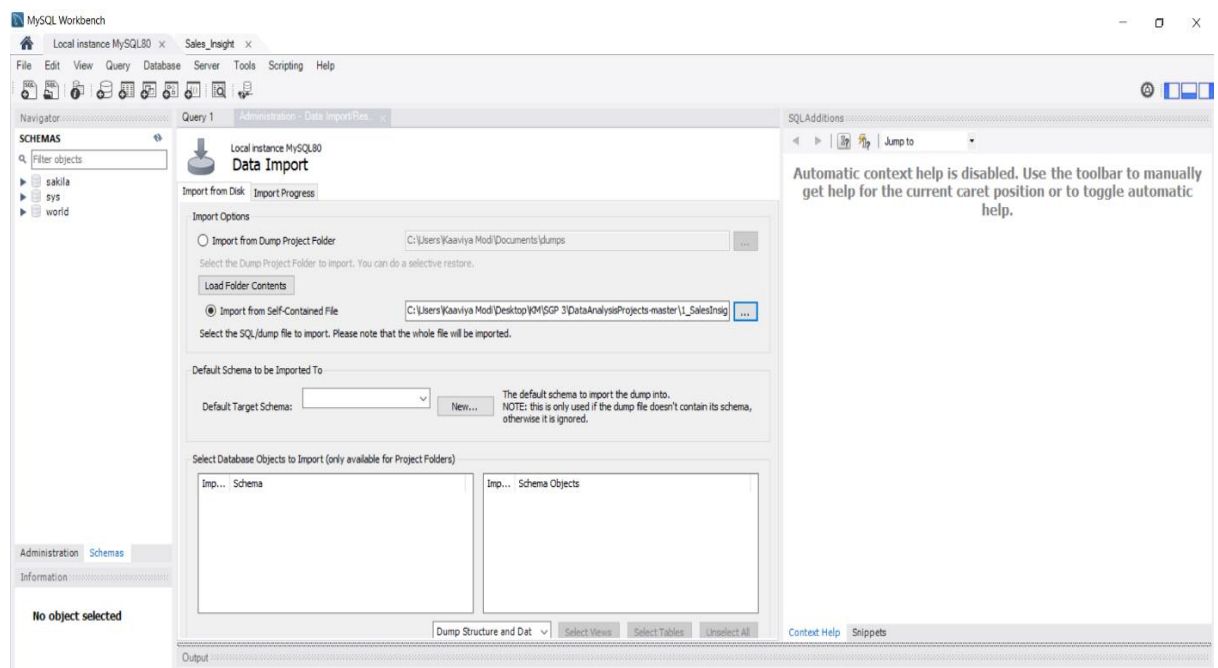
The team of Data Analyst (Data masters) reaches out to the software engineers to get an access to data base which they can use to create the dashboard in PowerBI.

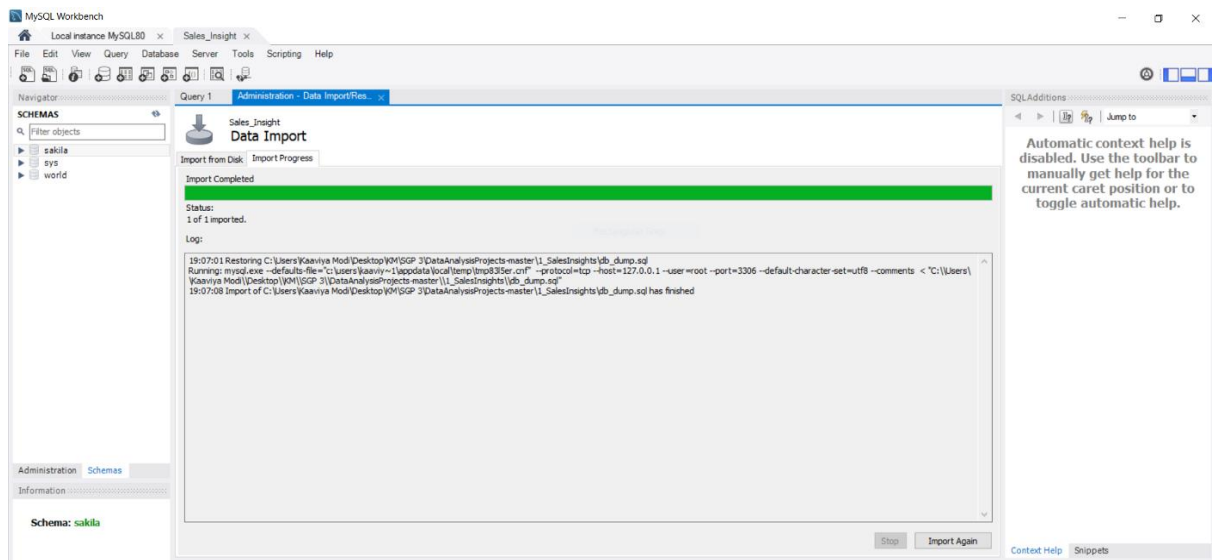
In this same manner our project is going to be executed. We are going to fetch the data from the database from company's website and then we are going to transform and load the data in the PowerBI to build the dashboard.



### 3. Data Analysis using SQL-

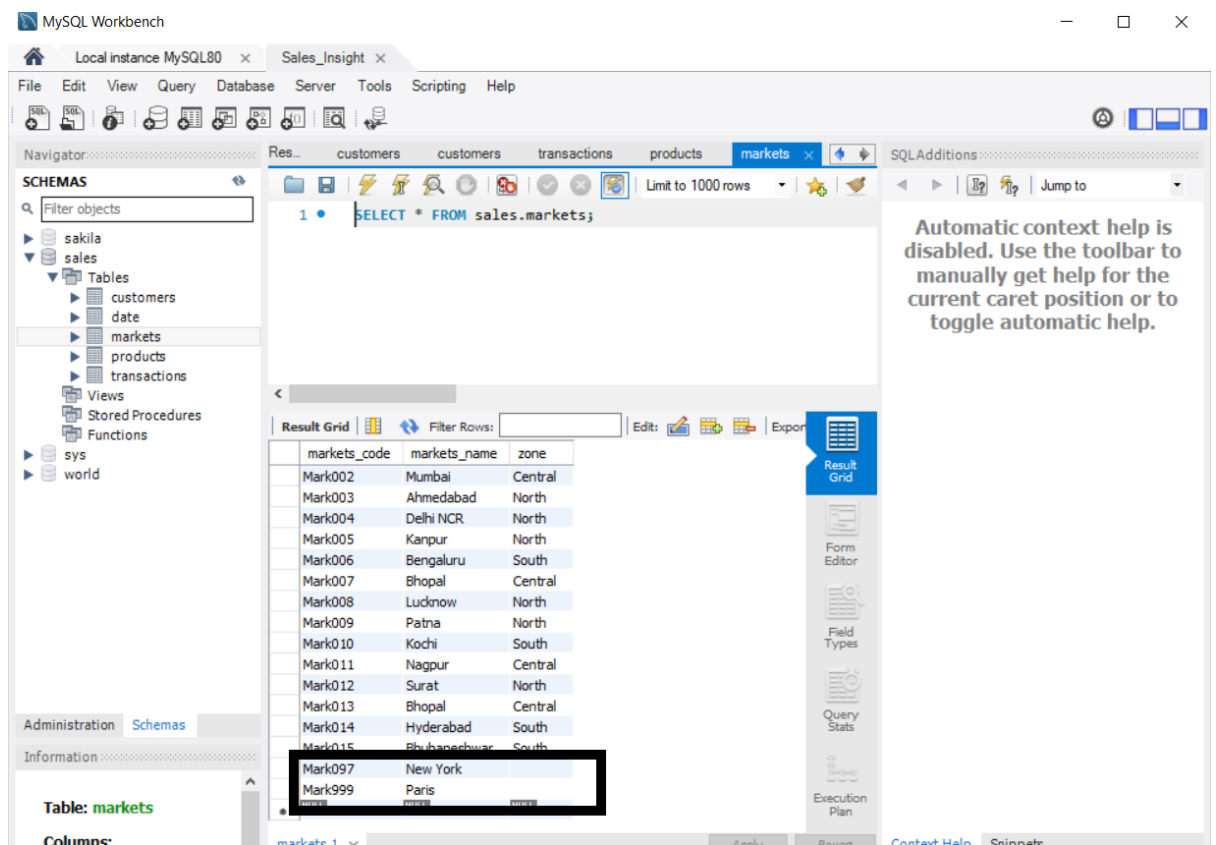
#### I. Importing Data to MySQL workbench-





The import of data is done from an already existing MySQL file. This file has to be loaded into MySQL workbench for further data analysis. The following images show that the import is a success.

## II. Simple analysis of data by looking into different tables and reflecting garbage values-



Here, we can see that the table market contains certain values which are

incorrect. AtliQ hardware company works only in India but there are some records of different non-existing cities in India.

MySQL Workbench interface showing a query result for `sales.transactions`. The result grid displays columns: `product_code`, `customer_code`, `market_code`, `order_date`, `sales_qty`, `sales_amount`, and `currency`. The data includes various transactions, with one notable entry showing a negative sales amount.

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
Prod001	Cus001	Mark001	2017-10-10	100	41241	INR
Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
Prod002	Cus003	Mark003	2018-04-06	1	875	INR
Prod002	Cus003	Mark003	2018-04-11	1	583	INR
Prod002	Cus004	Mark003	2018-06-18	6	7176	INR
Prod003	Cus005	Mark004	2017-11-20	59	500	USD
Prod003	Cus005	Mark004	2017-11-22	36	250	USD
Prod003	Cus005	Mark004	2017-11-23	39	21412	INR
Prod003	Cus005	Mark004	2017-11-27	35	19213	INR
Prod003	Cus005	Mark004	2017-11-28	310	170185	INR
Prod003	Cus005	Mark004	2017-11-29	184	101194	INR
Prod003	Cus005	Mark004	2017-11-30	35	19213	INR
Prod004	Cus005	Mark004	2017-11-29	17	9426	INR
Prod004	Cus005	Mark004	2017-12-19	1	218	INR
Prod005	Cus005	Mark004	2018-08-07	5	3093	INR
Prod003	Cus006	Mark004	2017-12-04	58	30306	INR
Prod005	Cus006	Mark004	2018-06-29	38	52319	INR

Here, we can see that table transactions contain certain negative value in amount which is not possible.

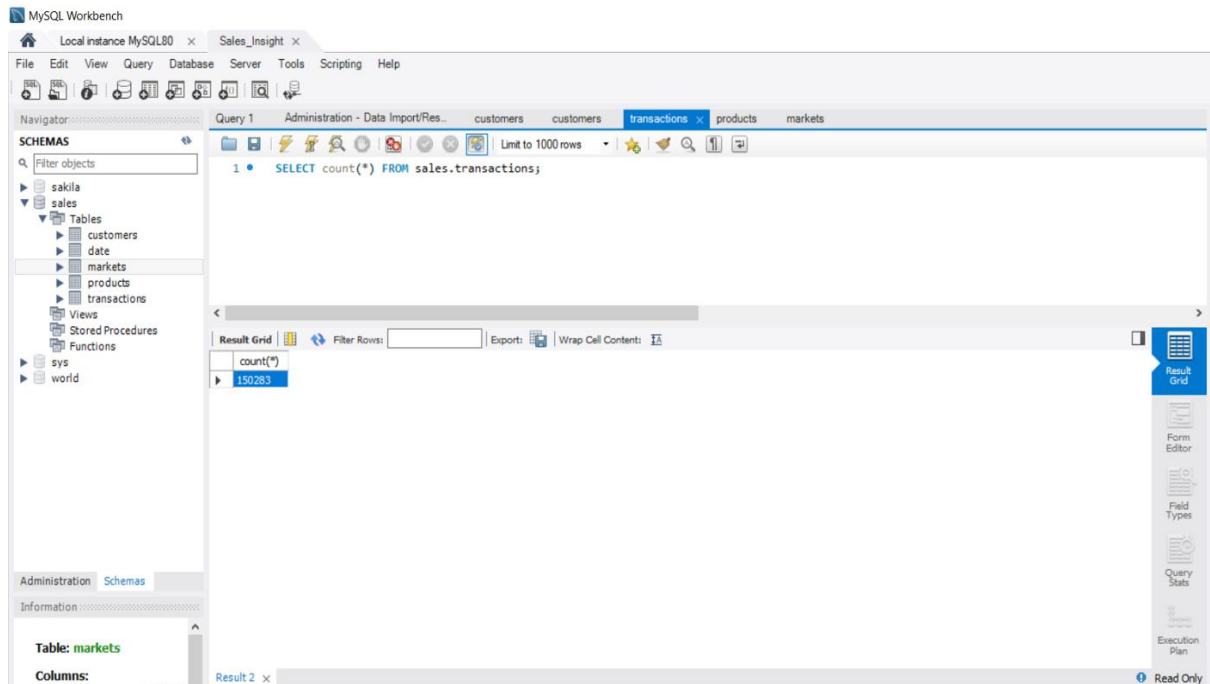
MySQL Workbench interface showing a query result for `sales.transactions`. The result grid displays columns: `product_code`, `customer_code`, `market_code`, `order_date`, `sales_qty`, `sales_amount`, and `currency`. The data includes various transactions, with one notable entry showing a sales amount in USD.

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
Prod001	Cus001	Mark001	2017-10-10	100	41241	INR
Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
Prod002	Cus003	Mark003	2018-04-06	1	875	INR
Prod002	Cus003	Mark003	2018-04-11	1	583	INR
Prod002	Cus004	Mark003	2018-06-18	6	7176	INR
Prod003	Cus005	Mark004	2017-11-20	59	500	USD
Prod003	Cus005	Mark004	2017-11-22	36	250	USD
Prod003	Cus005	Mark004	2017-11-23	39	21412	INR
Prod003	Cus005	Mark004	2017-11-27	35	19213	INR
Prod003	Cus005	Mark004	2017-11-28	310	170185	INR
Prod003	Cus005	Mark004	2017-11-29	184	101194	INR
Prod003	Cus005	Mark004	2017-11-30	35	19213	INR
Prod004	Cus005	Mark004	2017-11-29	17	9426	INR
Prod004	Cus005	Mark004	2017-12-19	1	218	INR
Prod005	Cus005	Mark004	2018-08-07	5	3093	INR
Prod003	Cus006	Mark004	2017-12-04	58	30306	INR
Prod005	Cus006	Mark004	2018-06-29	38	52319	INR

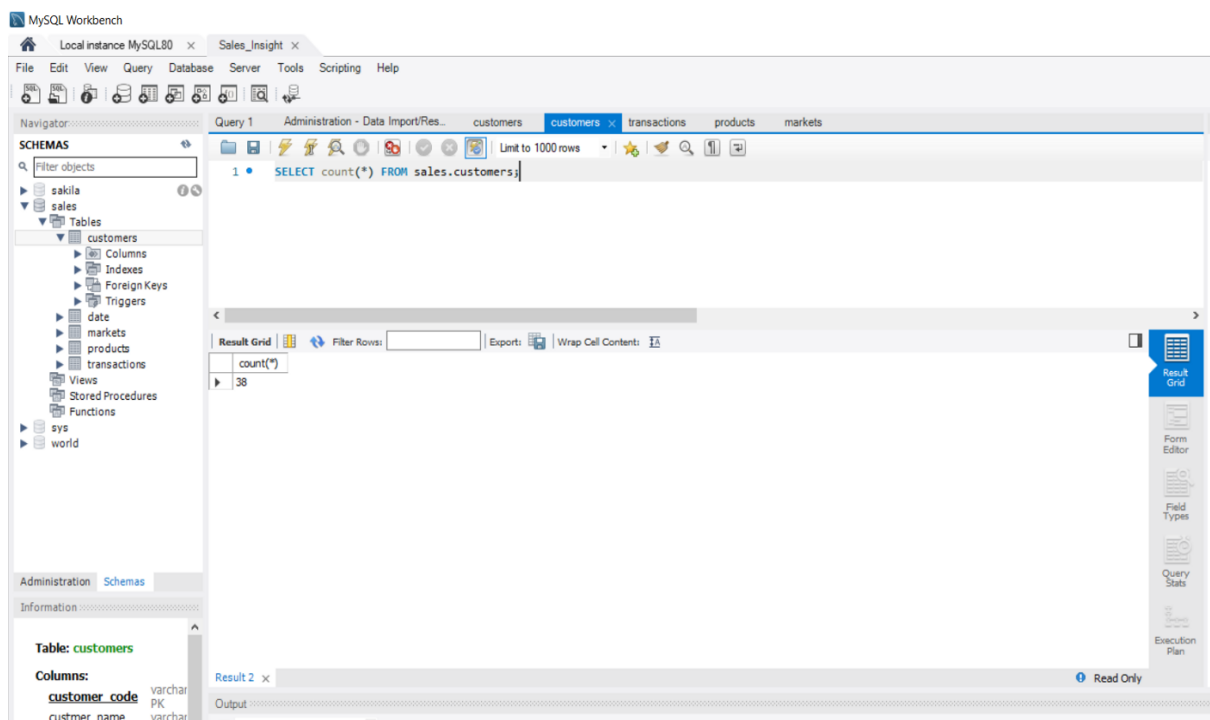
Here, you can see that certain transactions are in USD. Hence, filtration of that is also needed by converting into INR.

### III. Primary analysis of data base by running different SQL statements-

To find out how many total records are there in transaction table.



To find number of records in customer table.



To find out the records from transaction table with a specific market code.

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query: `SELECT * FROM sales.transactions where market_code = "Mark001";`. The 'Result Grid' shows the results of the query, listing columns: product\_code, customer\_code, market\_code, order\_date, sales\_qty, sales\_amount, and currency. The results are filtered to show only records where market\_code is 'Mark001'.

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
Prod001	Cus001	Mark001	2017-10-10	100	41241	INR
Prod013	Cus001	Mark001	2017-10-10	240	143560	INR
Prod001	Cus001	Mark001	2017-10-10	100	41241	INR
Prod013	Cus001	Mark001	2017-10-10	240	143560	INR
Prod016	Cus001	Mark001	2018-06-28	187	115481	INR
Prod020	Cus001	Mark001	2017-10-10	140	153019	INR
Prod020	Cus001	Mark001	2017-10-25	47	51005	INR
Prod020	Cus001	Mark001	2017-11-14	47	51005	INR
Prod040	Cus001	Mark001	2018-01-05	240	168245	INR
Prod040	Cus001	Mark001	2018-01-12	167	104648	INR
Prod040	Cus001	Mark001	2018-01-15	240	168245	INR
Prod040	Cus001	Mark001	2018-01-30	193	135532	INR
Prod040	Cus001	Mark001	2018-02-12	40	28042	INR

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query: `SELECT count(*) FROM sales.transactions where market_code = "Mark001";`. The 'Result Grid' shows the results of the query, listing columns: count(\*). The result is 1035.

count(*)
1035

To find out transaction of a particular year which is joint by the date table.



MySQL Workbench interface showing a query result for sales transactions. The query is: `SELECT sales.transactions.*, sales.date.* FROM sales.transactions INNER JOIN sales.date ON sales.transactions.order_date= sales.date.date`. The result grid displays columns: product\_code, customer\_code, market\_code, order\_date, sales\_qty, sales\_amount, currency, date, cy\_date, year, month\_name, date\_yy\_mmm. The data shows transactions from 2017-10-10 to 2018-08-07.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

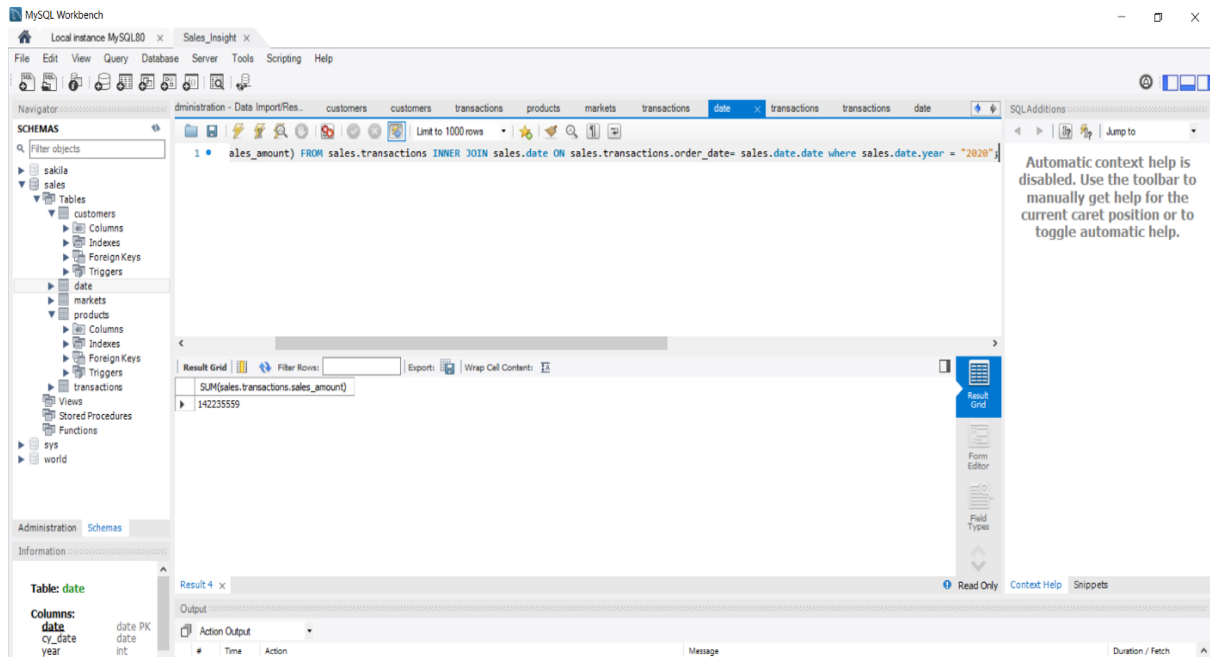
MySQL Workbench interface showing a query result for sales transactions for the year 2020. The query is: `SELECT sales.date.* FROM sales.transactions INNER JOIN sales.date ON sales.transactions.order_date= sales.date.date where sales.date.year = "2020"`. The result grid displays columns: product\_code, customer\_code, market\_code, order\_date, sales\_qty, sales\_amount, currency, date, cy\_date, year, month\_name, date\_yy\_mmm. The data shows transactions from 2020-01-09 to 2020-05-15.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

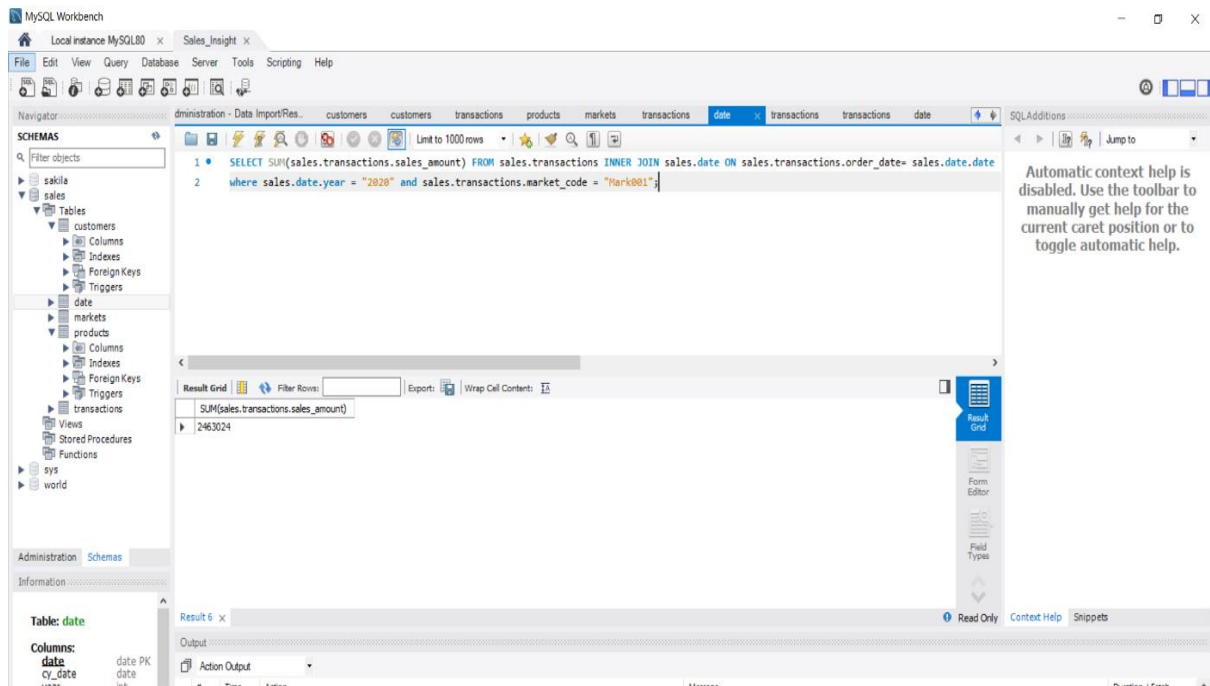
Here, we are performing inner join by joining the date and year together which shows record only of the year 2020.

To find out the total revenue of a particular year.





To find out the business you did in a particular city (eg. Chennai).

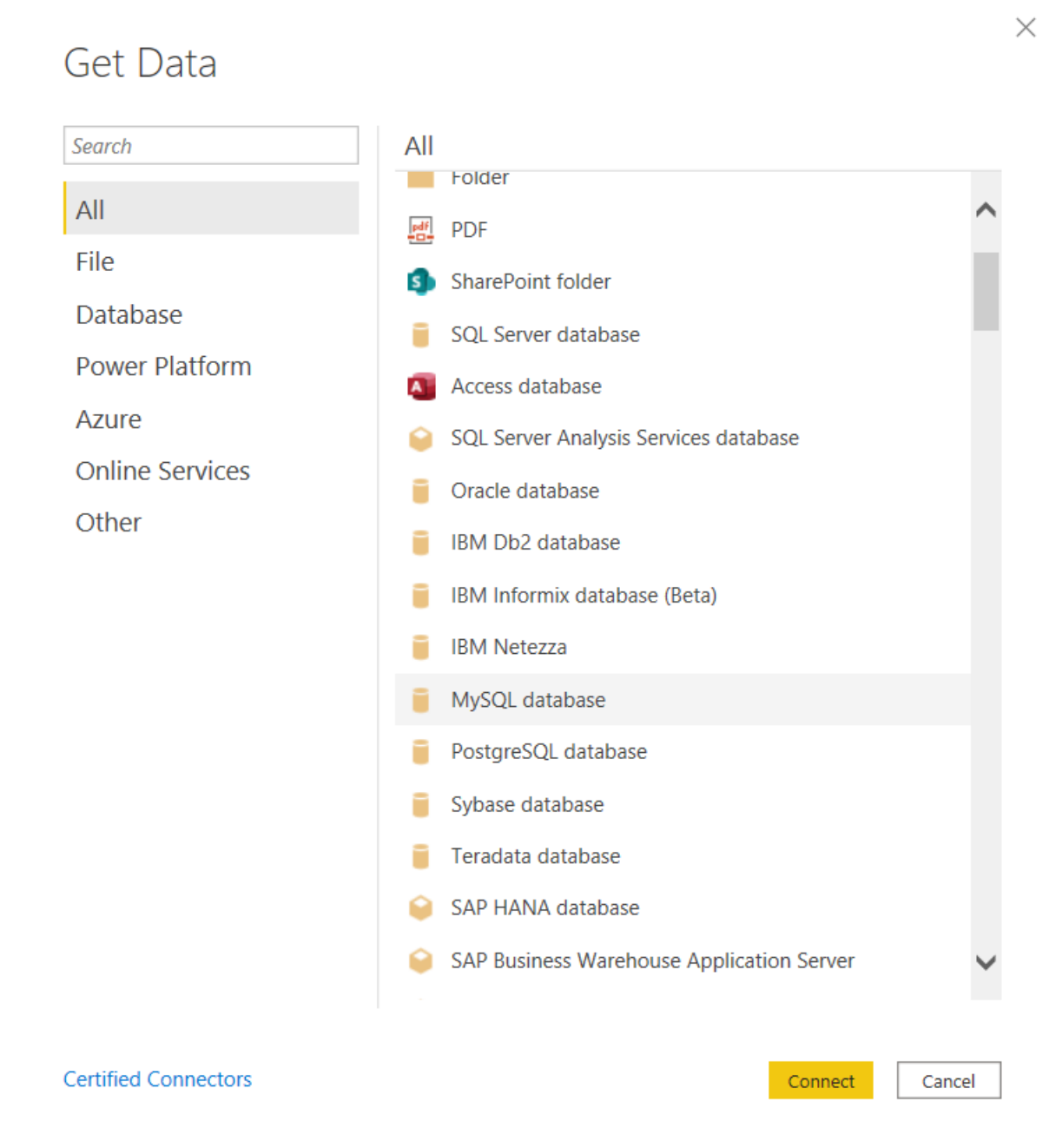


Here the market code of Chennai is used to get all the records. Similarly, if we want different of any other particular city the market code of that city is used.

#### 4. Data Cleaning and ETL (extract transform load)-

## I. Connect MySQL with the PowerBI desktop-

Here, we are using MySQL database to connect with the desktop.



## II. Loading the data into the PowerBI desktop-

Here, we are going to load all the tables we have created in the data base. This load option will connect with the SQL and pull all the records into power BI environment.

## Navigator

Display Options ▾

- localhost: sales [5]
  - ☒ sales.customers
  - ☒ sales.date
  - ☒ sales.markets
  - ☒ sales.products
  - ☒ sales.transactions

sales.products

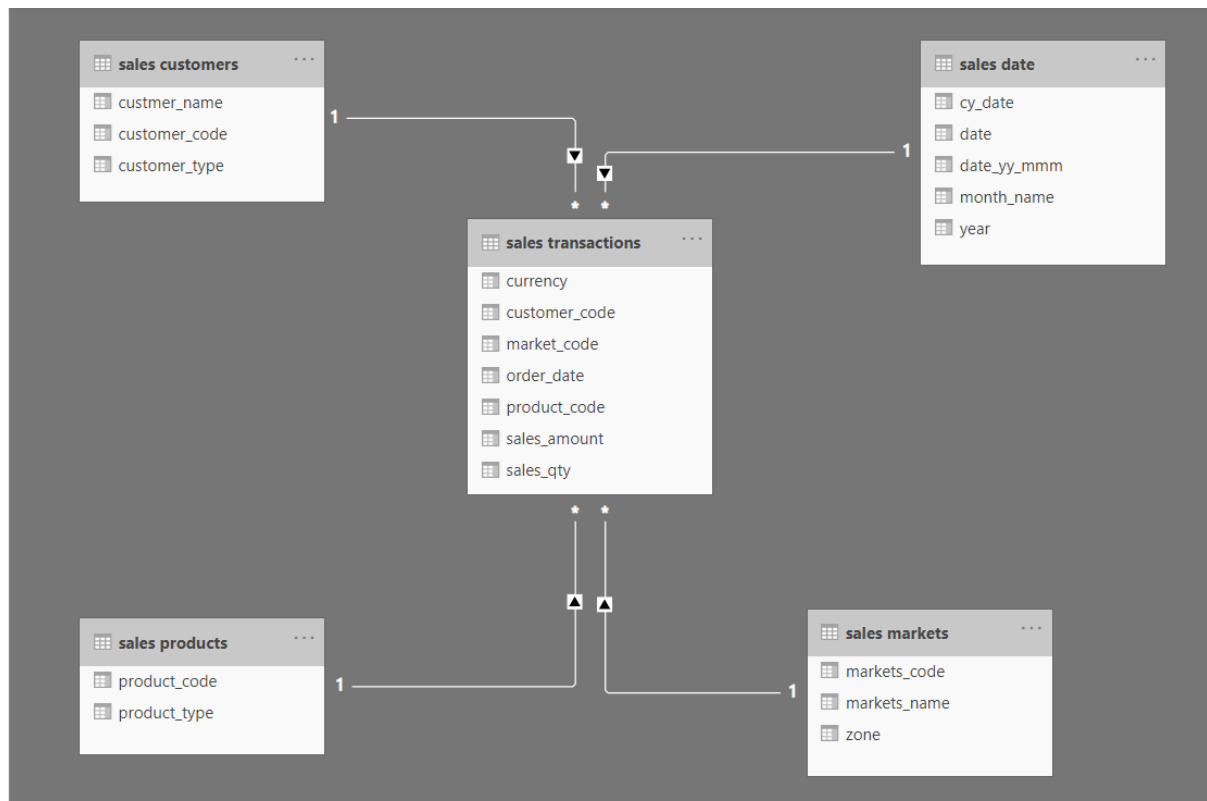
product_code	product_type
Prod001	Own Brand
Prod002	Own Brand
Prod003	Own Brand
Prod004	Own Brand
Prod005	Own Brand
Prod006	Own Brand
Prod007	Own Brand
Prod008	Own Brand
Prod009	Own Brand
Prod010	Own Brand
Prod011	Own Brand
Prod012	Own Brand
Prod013	Own Brand
Prod014	Own Brand
Prod015	Own Brand
Prod016	Own Brand
Prod017	Own Brand
Prod018	Own Brand
Prod019	Own Brand
Prod020	Own Brand
Prod021	Own Brand
Prod022	Distribution
Prod023	Distribution
Prod024	Distribution

Select Related Tables

Load

Transform Data

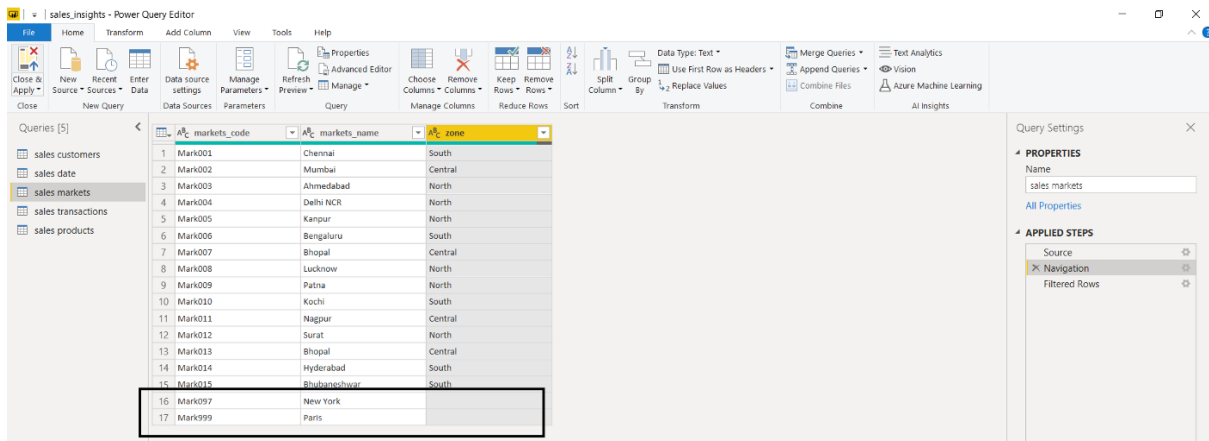
Cancel



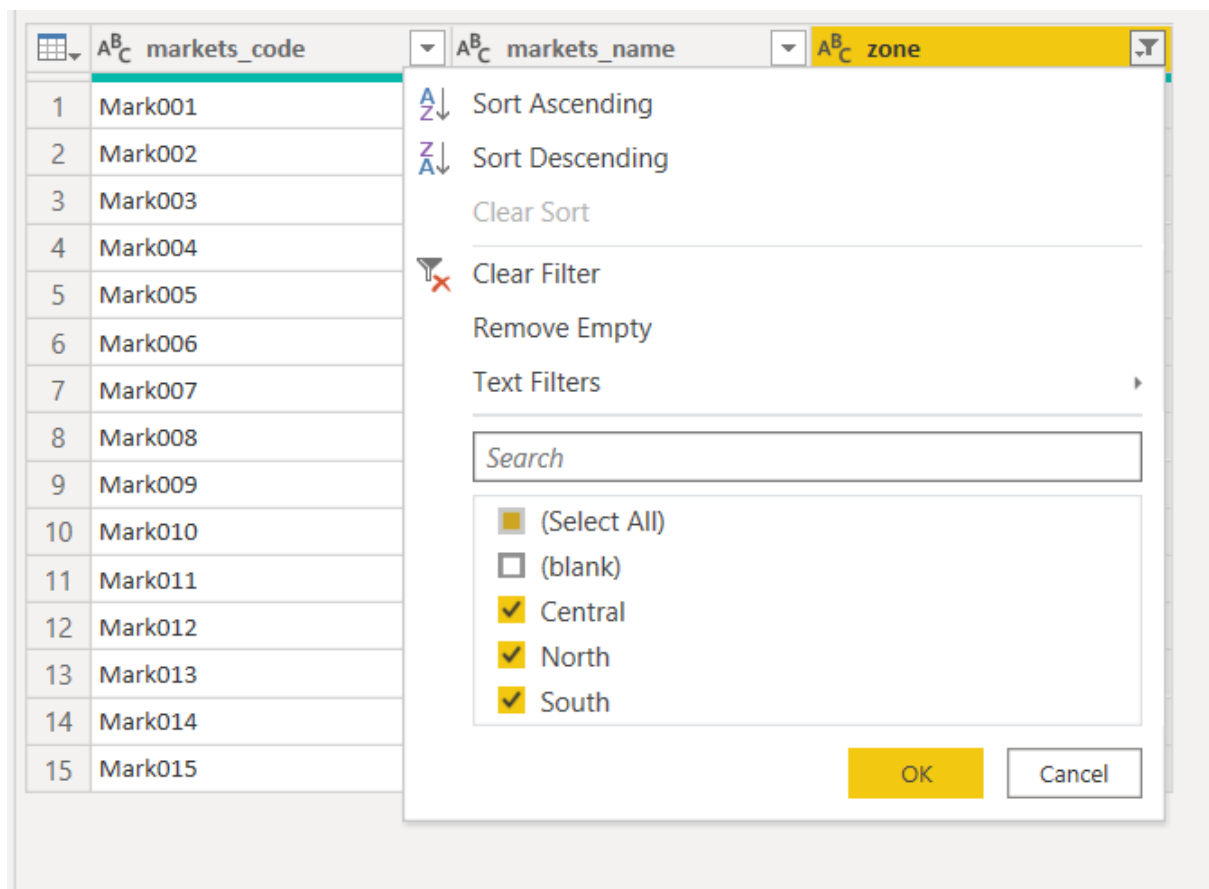
### III. Transforming data with the help of Power Query-

Performing filtration in market's table —

When we click on the transform data option, we are directed to Power query editor. Power query editor is where we perform out ETL. Here we can perform data transformation i.e., Data Cleaning/ Data Wrangling/ Data Munging.



Here, we need to filter the rows where the values are null and are inappropriate.



So, we are filtering the data and deselecting the blank option.

Power Query Editor - sales\_insights - Power Query Editor

Queries [5]

- sales customers
- sales date
- sales markets
- sales transactions
- sales products

markets_code	markets_name	zone
Mark001	Chennai	South
Mark002	Mumbai	Central
Mark003	Almedabad	North
Mark004	Delhi NCR	North
Mark005	Kanpur	North
Mark006	Bengaluru	South
Mark007	Bhopal	Central
Mark008	Lucknow	North
Mark009	Patna	North
Mark010	Kochi	South
Mark011	Nagpur	Central
Mark012	Surat	North
Mark013	Bhopal	Central
Mark014	Hyderabad	South
Mark015	Bhubaneswar	South

Query Settings

PROPERTIES

Name: sales markets

APPLIED STEPS

- Source
- Navigation
- Filtered Rows

This is the later output we get after filtration.

Performing filtration in Transaction's Table —

MySQL Workbench - Sales\_Insight

Navigator

SCHEMAS

- Filter objects
- sakila
- sales
  - Tables
    - customers
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - date
  - markets
  - products
  - transactions
- Views
- Stored Procedures
- Functions
- sys
- world

Administration Schemas

Information

products markets transactions date transactions x date customers

Limit to 1000 rows

1 • SELECT \* FROM sales.transactions where sales\_amount <= 0;

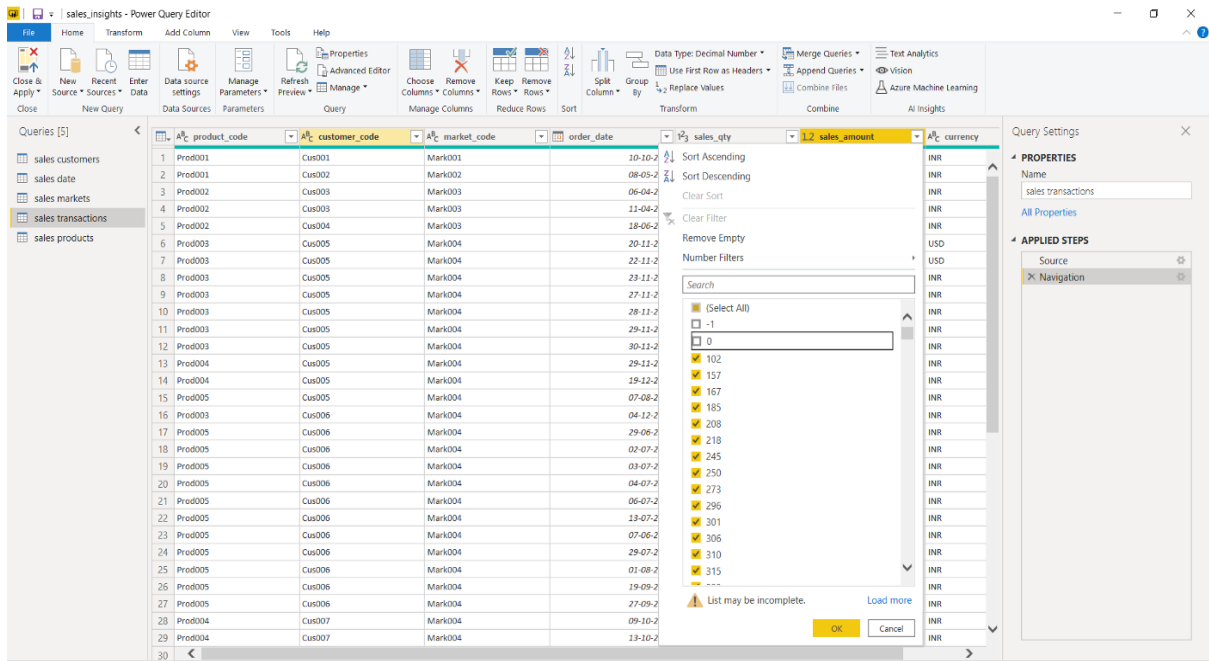
2

Result Grid

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
Prod010	Cus015	Mark006	2018-05-26	1	0	INR
Prod010	Cus003	Mark003	2019-04-30	1	0	INR
Prod011	Cus018	Mark002	2018-12-28	1	0	INR
Prod001	Cus002	Mark002	2018-05-08	3	-1	INR
Prod010	Cus015	Mark006	2018-05-26	1	0	INR
Prod010	Cus003	Mark003	2019-04-30	1	0	INR
Prod011	Cus018	Mark002	2018-12-28	1	0	INR
Prod016	Cus005	Mark006	2018-01-10	5	0	INR
Prod021	Cus003	Mark006	2017-10-10	1	0	INR
Prod021	Cus027	Mark006	2017-10-16	1	0	INR
Prod021	Cus003	Mark006	2017-10-18	1	0	INR
Prod021	Cus003	Mark006	2017-10-19	1	0	INR

transactions 4 x

When we run the query in the MySQL to filter some negative values and also zero values that appears in the table, the desired output is received. Now we will perform the similar filtration in PowerBI.



Deselecting the values, we don't want in the table.

The result after filtration. Here, the zero values represent some garbage values which is not possible so we need to clean that data

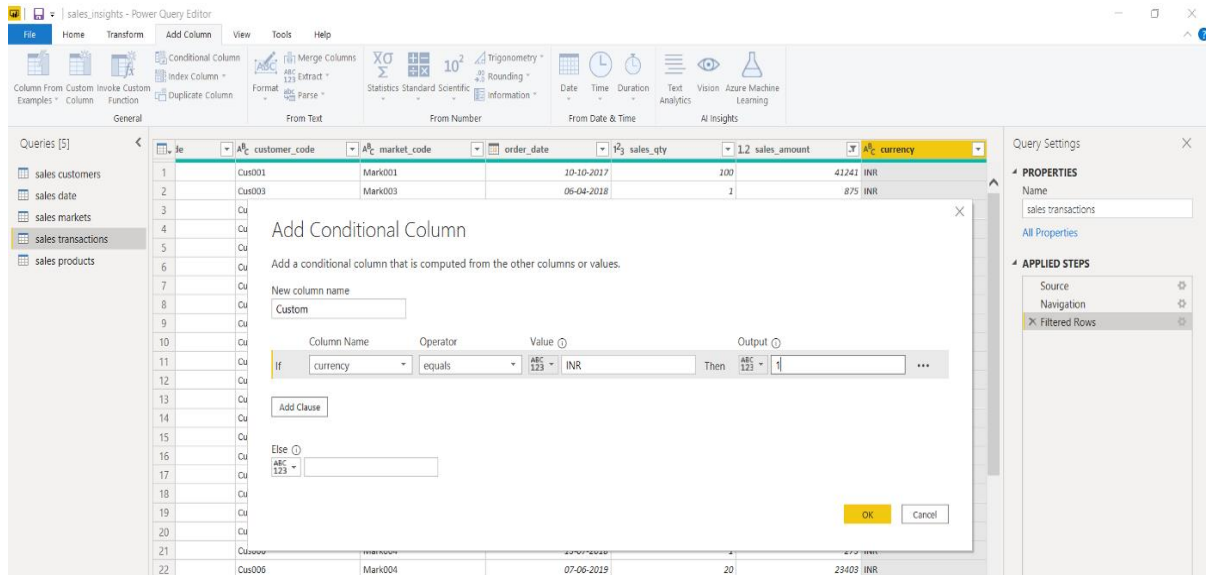
The screenshot shows the Power Query Editor interface after filtration. The table now only contains rows where sales\_qty is not zero. The 'currency' column shows values like INR and USD. The 'APPLIED STEPS' pane on the right shows 'Filtered Rows' as the current step.

product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
Prod001	Cus001	Mark001	10-10-2017	100	41241	INR
Prod002	Cus003	Mark003	06-04-2018	1	875	INR
Prod002	Cus003	Mark003	11-04-2018	1	583	INR
Prod002	Cus004	Mark003	18-06-2018	6	7176	INR
Prod003	Cus005	Mark004	20-11-2017	59	500	USD
Prod003	Cus005	Mark004	22-11-2017	36	250	USD
Prod003	Cus005	Mark004	23-11-2017	39	21412	INR
Prod003	Cus005	Mark004	27-11-2017	35	19213	INR
Prod003	Cus005	Mark004	28-11-2017	310	170185	INR
Prod003	Cus005	Mark004	29-11-2017	184	101194	INR
Prod003	Cus005	Mark004	30-11-2017	35	19213	INR
Prod004	Cus005	Mark004	29-11-2017	17	9426	INR
Prod004	Cus005	Mark004	19-12-2017	1	218	INR
Prod005	Cus005	Mark004	07-08-2018	5	3093	INR
Prod003	Cus006	Mark004	04-12-2017	58	30306	INR
Prod005	Cus006	Mark004	29-06-2018	38	52319	INR
Prod005	Cus006	Mark004	02-07-2018	93	126296	INR
Prod005	Cus006	Mark004	03-07-2018	79	107500	INR
Prod005	Cus006	Mark004	04-07-2018	1	273	INR
Prod005	Cus006	Mark004	06-07-2018	3	3574	INR
Prod005	Cus006	Mark004	13-07-2018	1	273	INR
Prod005	Cus006	Mark004	07-06-2019	20	23403	INR
Prod005	Cus006	Mark004	29-07-2019	81	76329	INR
Prod005	Cus006	Mark004	01-08-2019	5	4542	INR
Prod005	Cus006	Mark004	19-09-2019	18	16579	INR
Prod005	Cus006	Mark004	27-09-2019	90	105301	INR
Prod004	Cus007	Mark004	09-10-2017	1	185	INR
Prod004	Cus007	Mark004	13-10-2017	1	556	INR
Prod004	Cus007	Mark004	27-10-2017	1	551	INR

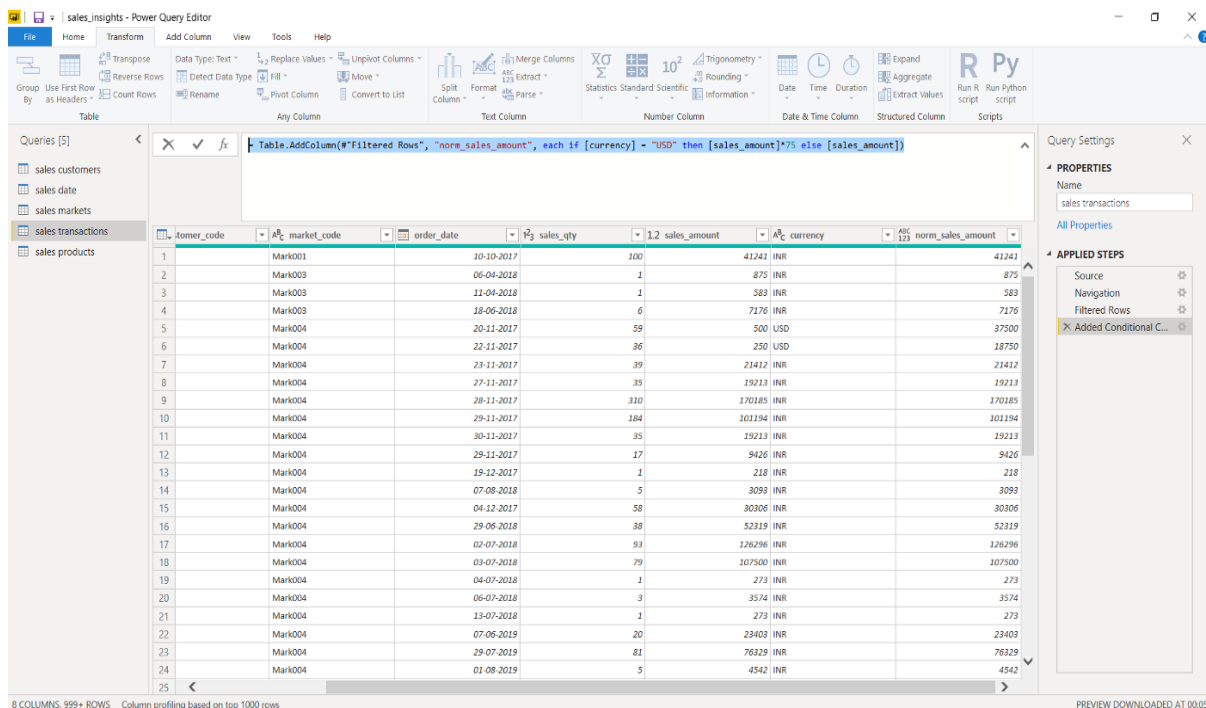
Converting the currency from USD to INR in the transaction's table —



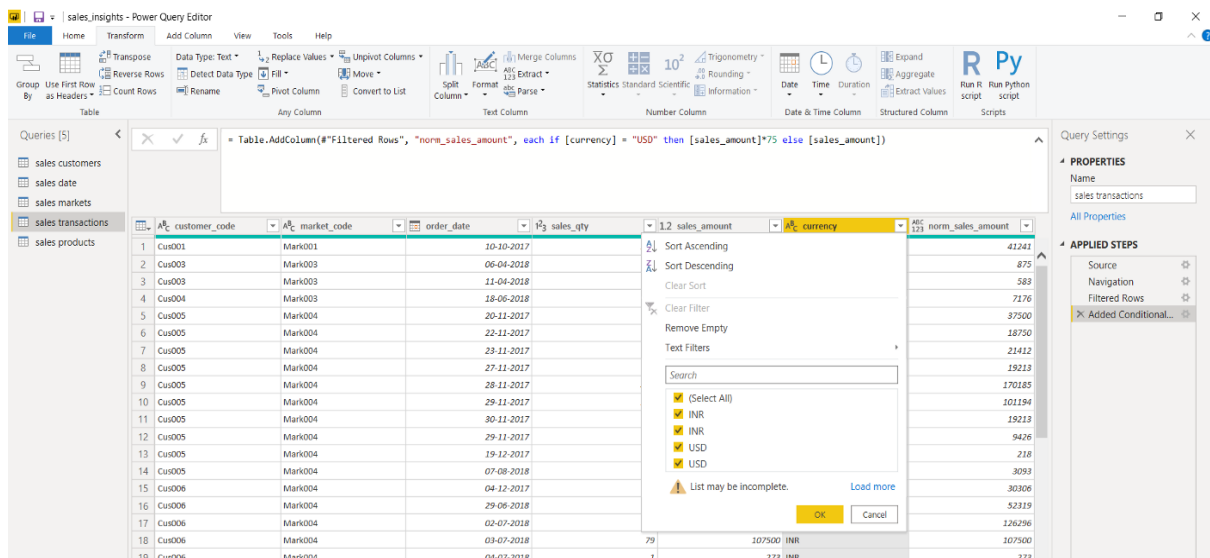
Here the company only works in India so the USD values are not possible. So, we need to convert those USD values into INR by using some formulas. Here we are going to form a separate column and display the converted currency value into it.



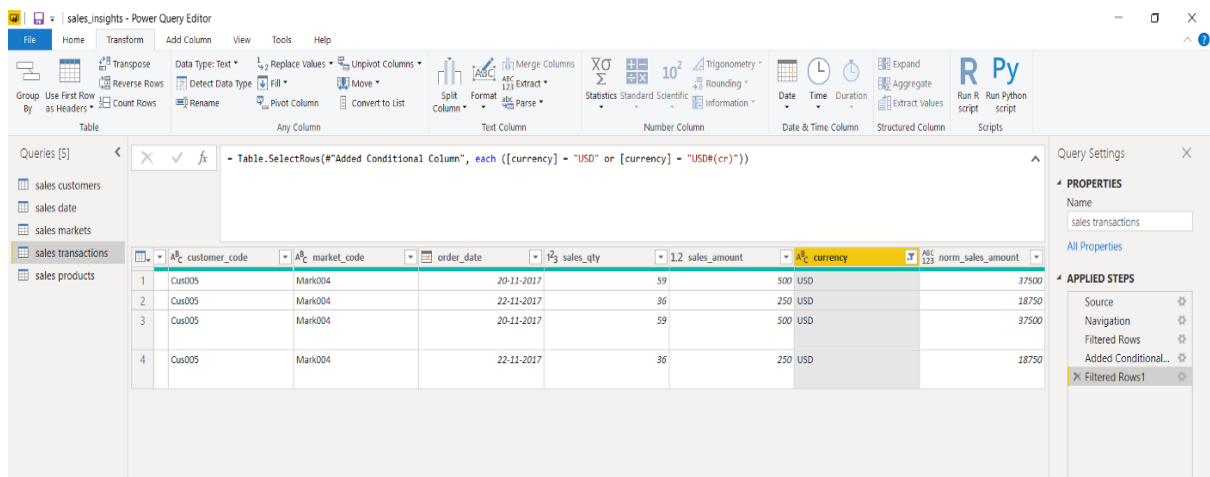
Finding out the total values having USD as currency.



By using the correct formula of the conversion, we have converted the USD currency into INR.



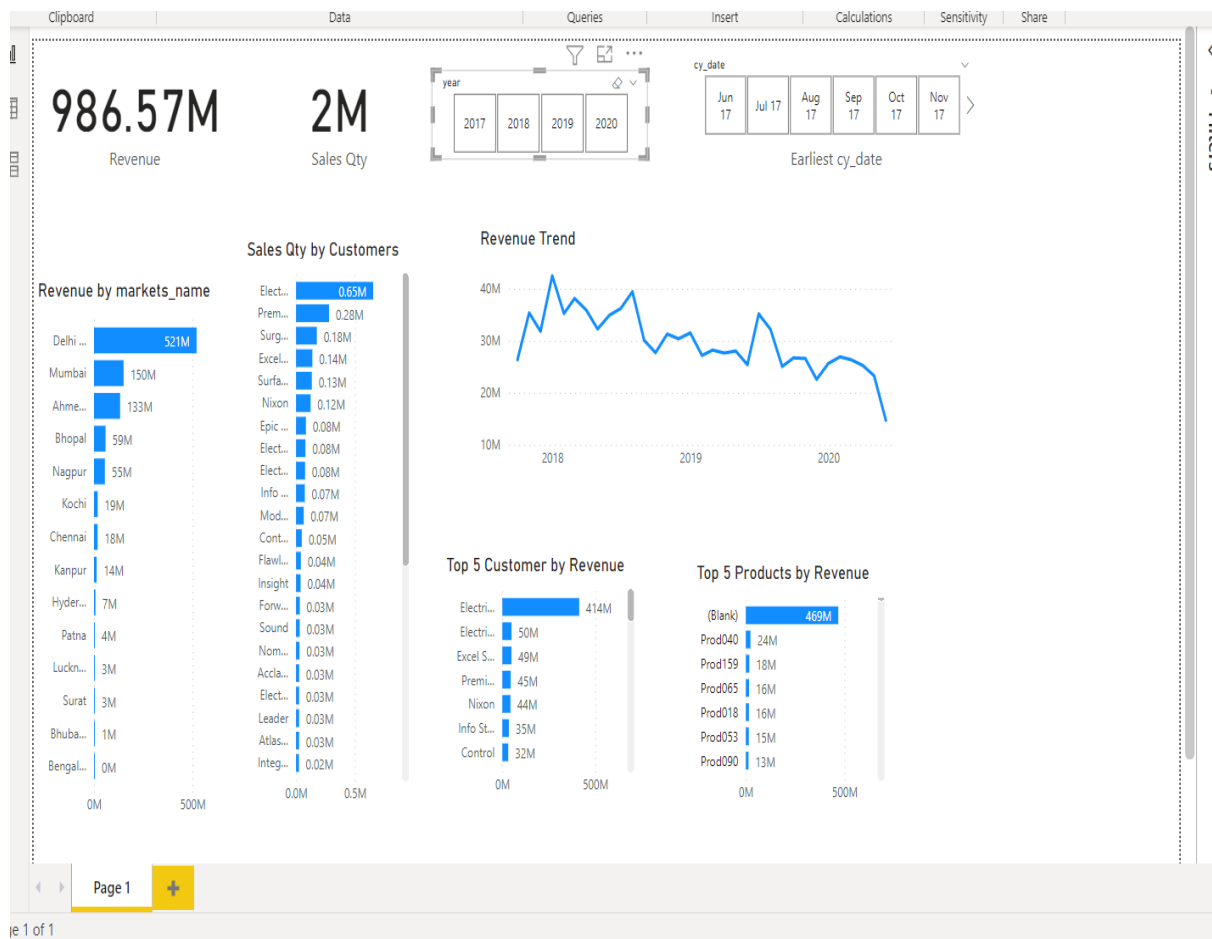
Now when we checked the filtration, we found something. Due to some reason the USD is appearing twice. So now we will filter those values also.



Hence, all the USD values are converted into INR now.

## 5. Building a Dashboard or a Report-

Dashboards/reports are created according to the requirement. What actually the company wants to look for and what is more important for the company is taken into consideration and then after the dashboard is created. There can be n number of variations to create a dashboard. Generally, the dashboard should look understandable and an ease to access.



This is how the dashboard looks after formatting it completely.

## 6. References-

- Codebasics-

<https://www.youtube.com/watch?v=hhZ62IIxYs&list=PLeo1K3hjS3uva8pk1FI3iK9kCOKQdz119>

**\*\*\*Thank You\*\*\***