

Introduction:

This project concerns ABC Purchaser Electronics Ltd, an electronics retail organization with 6 retailers around London. It has heaps of products which can be categorized into 10 and cover more than two hundred manufacturers. The goal of this coursework is to design a facts version specially tailored for its want to put in force effective reporting force effective reporting, analysis of inventory, and the performance of the providers to aid knowledgeable choice-making and strategic making plans.

This venture seeks to implement a smooth glide of records integration and retrieval by implementing a records model specifically Star Schema. The dimensional statistics version will facilitate the corporation's capability to research key performance indicators, and music income developments, control inventory ranges and examine provider effectiveness. By the end of this coursework, a comprehensive design might be offered, demonstrating how the proposed model meets the reporting and analytical wishes of the commercial enterprise, in the end riding higher strategic decisions and operational efficiencies.

Analysis Business Activity:-

1. Purchase Order Sent, which gives us information about sending orders to suppliers when there is a minimum inventory level for certain products. This table gives all the relevant information to the suppliers.
2. Receiving Purchased Orders gives us information about the stocks and the location in which they should be stored. This table has all the information about the products ordered and the details of which supplier sent the products.
3. Inventory management, gives us information about the current stocks in the inventory, and new products and also helps in maintaining the existing stocks

Identifying Fact Tables:- The fact tables in a data warehouse are the central table that stores all the significant numbers of the business activities. The fact table for ABC Consumer Electronics Outlet Ltd. Are:

1. Fact Table for Purchase Order Sent
2. Purchase Order Received
3. Stock Management

Dimensional Data Model Designing:-

Grain: The first step in building a dimensional data model is by finding out the grain of each table. The grain of a fact table is the basic level or most atomic level of detail of each row in a table. We have three main business activities to consider for the company and according to it we should create 3 fact tables. The three fact tables are

- Fact Table for Purchase Orders Sent:

1. Grain: Each record represents a unique purchase order sent to a supplier.
2. Details: The grain table is about each row of the purchase order sent to the supplier which includes the foreign keys Product ID, supplier ID, Outlet ID, Sent Date, and surrogate key sent order ID along with the other quantitative columns

Product ID	Supplier ID	Sent Order ID	Destination Outlet ID	Ordered QTY	Supplier Name	Sent Date	Product SKU
------------	-------------	---------------	-----------------------	-------------	---------------	-----------	-------------

- Fact Table for Purchase Orders Receive:

1. Grain: Each record represents a unique purchase order that has been received.
2. Details: The grain table is the visual representation of each row of the purchase order received, which includes the Surrogate Key Product Order ID and some foreign keys like Product ID, Supplier ID, Destination Outlet ID, Date, Product SKU. There are some quantitative columns as well which are received quantity, ordered quantity.

Purchase Order ID	Product ID	Supplier ID	Destination Outlet ID	Sent Date	Product SKU	Received Quantity	Received Date	Ordered quantity	Supplier Name
-------------------	------------	-------------	-----------------------	-----------	-------------	-------------------	---------------	------------------	---------------

- Fact Table for Inventory Management

1. Grain: Each record represents the current stock level of a product at a specific point in time.
2. Details: The grain table is the visual representation of each row of the Inventory management. Which includes the surrogate key Inventory ID. Certain foreign keys include product ID, Current Stock Level, Date, Active, Cost, and Retail Price.

Inventory ID	Product ID	Current Stock Level	Cost Price	Retail Price	Date Created AT	IS Active
--------------	------------	---------------------	------------	--------------	-----------------	-----------

Identifying Dimensions: The dimensions in a fact table are the descriptive context of the information stored in the particular fact table. While writing a query, to derive insights into 'who', 'where', 'when', and 'why' we turn to dimensions.

- Fact Table for Purchase Orders Sent: The dimensions of this table contain all the information that the supplier should have, like:
 1. Product Dimension: Gives details about the product for orders sent
 2. Supplier Dimension: Gives details about the supplier to whom the orders had been sent.
 3. Date Dimension: Give details about the date and the time at which the order was sent
- Fact Table for Purchase Orders Received: The dimensions of this table contain all the information that the purchase order that has been received, like:
 1. Product Dimension: Gives details about the product purchase received

2. Supplier Dimension: Gives details about from which supplier the order has been received.
3. Date Dimension: Gives details about the date and time on which the order was received
- Fact Table for Inventory Management: The dimensions for this table contain all the information necessary to manage and maintain the inventory, like:
 1. Product Dimension: Gives details about the new product, and existing product.
 2. Supplier Dimension: Gives details about the supplier of the products.
 3. Date Dimension: Gives details about the date and time to record every product movement.
 4. Outlet Dimension: Gives details about the location through which the stock moves.

Identifying the Measures and Attributes: - The attributes in a fact table that link to the dimensional tables, which provide a more detailed description of the dimensions. These attributes in these fact tables give us more information about its dimensions table, such as:

1. Product Dimension Table: The attributes for this table are Product ID, Product Name, Product Type, Brand, Product SKU, Product Description, Condition, Supplier Name.
2. Supplier Dimension Table: This table has the attributes Supplier ID, Supplier Name, Supplier Address, Phone, Email, Fax, Postcode, City, State, CountryID
3. Date Dimension Table: The attributes for this table are Date, Week, Month, Quarter, Year.
4. Outlet Dimension Table: The attributes for this table are Outlet ID, Outlet Name, and Outlet Address, City, State, Country, Postcode.

Measures are the quantitative values in the fact table that can be used for analysis. These are always numbers that can be used for various calculations over time. The measures for the fact tables are:

1. Purchase Order Sent: Order Quantity.
2. Purchase Order Received: Ordered Quantity, Received Quantity.
3. Inventory: Current Stock Level. **Star Schema**

Supplier Dimension	
Supplier ID (PK)	
Supplier Name	
Supplier Address	
Supplier Phone	
Email	
Fax	
Address	
City	
State	
Country	

Date Dimension	
Date ID (PK)	
Day	
Week Day	
Month Name	
Month No.	
QuarterNo	
Year	

Fact Table Sent Purchase Order	
Purchase Order Code(PK)	
SKU (FK)	
Supplier ID(FK)	
Order QTY	
Date ID(FK)	
Sent Date	
Product ID (FK)	
Destination Outlet ID	
Supplier Name	

Fact Table 2	
Purchase Order Code (PK)	
SKU (FK)	
Supplier ID (FK)	
Order QTY	
Date ID (FK)	
Destination Outlets ID (FK)	
Received QTY	
Received Date	
Product ID(FK)	
Sent Date	
Supplier Name	

Fact Table 3	
Current Stock Level	
Product ID(FK)	
Date ID (FK)	
Outlets ID(FK)	
Date Created At	
Cost Price	
Retail Price	
IS Active	
Inventory ID(PK)	

Product Dimension	
SKU	
ProductName	
Product Type	
Brand	
Product Type	
Product ID(PK)	
Description	
Condition	
Supplier Name	

Outlet Dimension	
Outlet ID (PK)	
Outlet Name	
Outlet Address	
City	
State	
Country	
Postcode	

Building a Data Warehouse:

After downloading the SSMS the first step is to create a database by writing the SQL command. The database is created to store all the information and to find all the business activities that the company needs. The syntax that is used to create a database is “CREATE DATABASE database_Name”. After starting the SSMs we have to click on the new query

which will open a window to write and execute the queries. So the command used to create the database for ABC Ltd is

CREATE DATABASE ABCdatabase:

After writing the query to execute the command we have to click on execute key on the top right hand side of the SSMS window. After executing if the command is run successfully the console will show that the command was run successfully and create a database to work on. The database will not show on the object explorer side until refreshed. After refreshed the new database that has been created can be seen under the database section on the left hand side as shown in the FIGURE 1.1

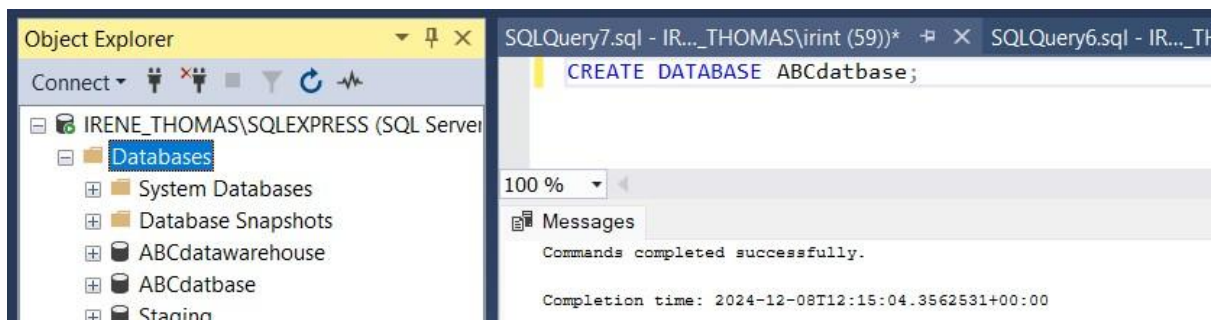


FIGURE 1.1

After creating database we can start creating dimension tables and fact tables that will be needed for ABC Ltd. The query to create a table is "CREATE TABLE TABLE_NAME(

COLUMN1,COLUMN2,...);" **Creation of Dimension Tables:**

Product Dimension Table: To create a table we have to find the Primary key and the other attributes and measures that need to be created to carry on the business activity. Here the Product dimension is being created with ProductID as the primary key and the rest are the other columns that will be needed to store the information about a particular product. The query used to create the Product Dimension table for ABC is

```
Create table dime_TableProduct(  
ProductID INT PRIMARY KEY IDENTITY(1,1),  
SKU VARCHAR(50),  
ProductName VARCHAR(100),  
ProductDescription VARCHAR(150),  
Condition VARCHAR(50),  
ProductType VARCHAR(50),  
Brand VARCHAR(50),  
SupplierName VARCHAR(100),  
);
```

After executing this query the output will look like the FIGURE1.2

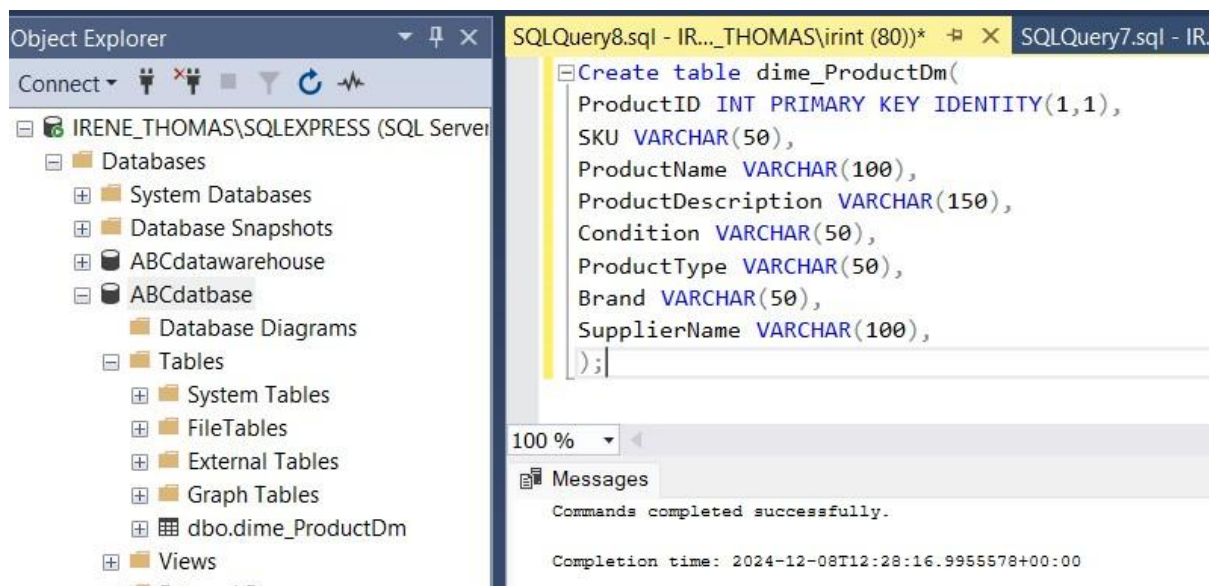


FIGURE 1.2

Supplier Dimension: The next table is the Supplier Dimension table which is used to store all the information about the Supplier who provides all the stocks necessary. The primary key in this table is Supplier ID which is a surrogate key created to make it easier to track each supplier .The query used to create a Supplier table is

```
CREATE TABLE dime_SupplierDm(  
  
SupplierID INT PRIMARY KEY IDENTITY(1,1),  
  
SupplierName VARCHAR(100),  
  
Phone INT,  
  
Email VARCHAR(50),  
  
FAX VARCHAR(20),  
  
Address VARCHAR (250),  
  
PostCode VARCHAR(20),  
  
City varchar(20),  
  
State Varchar (20),  
  
CountryID VARCHAR(20),  
  
);
```

After executing the given code the output will look like the FIGURE1.3

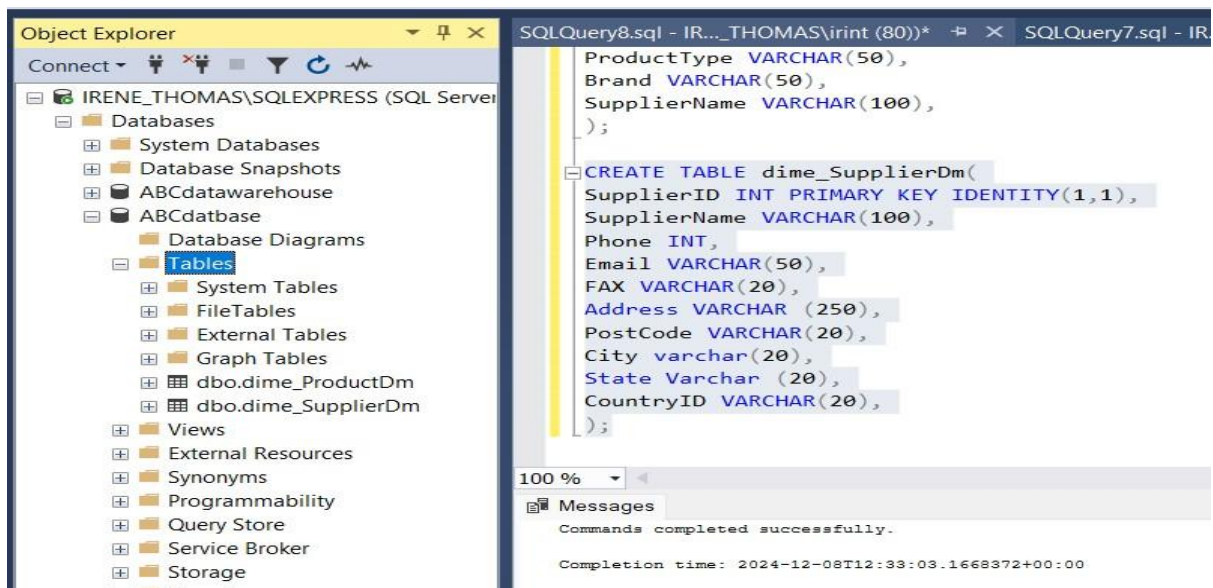


FIGURE 1.3

Date Dimension: The next table that is needed to create is the Date table which stores all the information about the date for ABC Ltd. The primary key in the table is Date ID which is used to store the date. It has all the information needed about the dates. The query used to create a Date dimension is

```
CREATE TABLE dime_DateDm(
DateID Date PRIMARY KEY,
Day INT,
WeekDay varchar(10),
Month varchar(10),
MonthofYear INT,
YearNo INT,
QuarterNo INT,
);
```

After executing the query the output will look like FIGURE 1.4

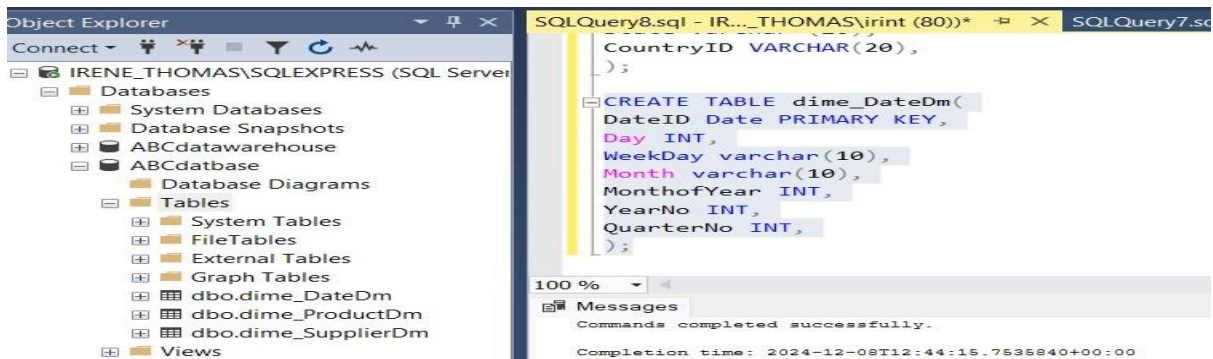


FIGURE1.4

Outlet Dimension: The next table is Outlet table which is used to create the information about the outlet from where the product is sold and received for ABC Ltd. The primary key is OutletID which is numeric value to track each outlet. The query used to create the table is :

```
CREATE TABLE dime_Outlet(
OutletID INT PRIMARY KEY IDENTITY(1,1),
OutletName VARCHAR(100),
OutletAddress VARCHAR(250),
OutletCity VARCHAR(50),
OutletState VARCHAR(50),
OutletCountry VARCHAR(50),
OutletPostcode VARCHAR(20)
);
```

After executing the code the output will look like FIGURE 1.5

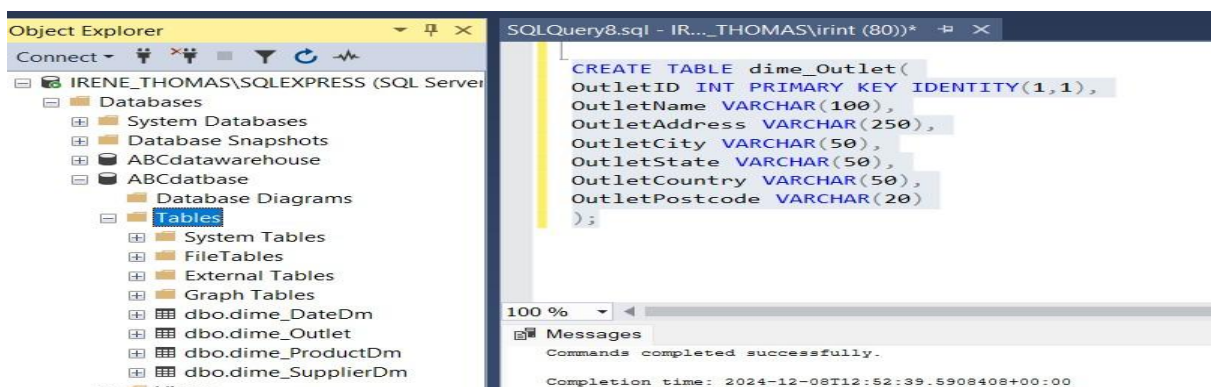


FIGURE 1.5

CREATION OF FACT TABLES: Purchase Order Sent Fact Table: This fact table was created to store all the information about the order that has been sent to the supplier for ordering stock. The

primary key in the table is PurchaseOrderID and the foreign keys are DateID, SupplierID, ProductID, .There are other quantitative values as well. The query used to create the fact table are:

```
CREATE TABLE Fact_SentPurchaseOrders(

SentOrderID varchar(50) PRIMARY KEY,

ProductID INT,

SKU varchar(50),

SupplierID INT,

SupplierName varchar(100),

OutletName varchar(100),

SentDate DATE,

OrderedQTY INT,

FOREIGN KEY (ProductID) REFERENCES [dbo].[dime_ProductDm]([ProductID]),

FOREIGN KEY (SupplierID) REFERENCES [dbo].[dime_SupplierDm](SupplierID),

FOREIGN KEY (SentDate) REFERENCES [dbo].[dime_DateDm](DateID),

);
```

When the above SQL code is executed the output will look like FIGURE1.6

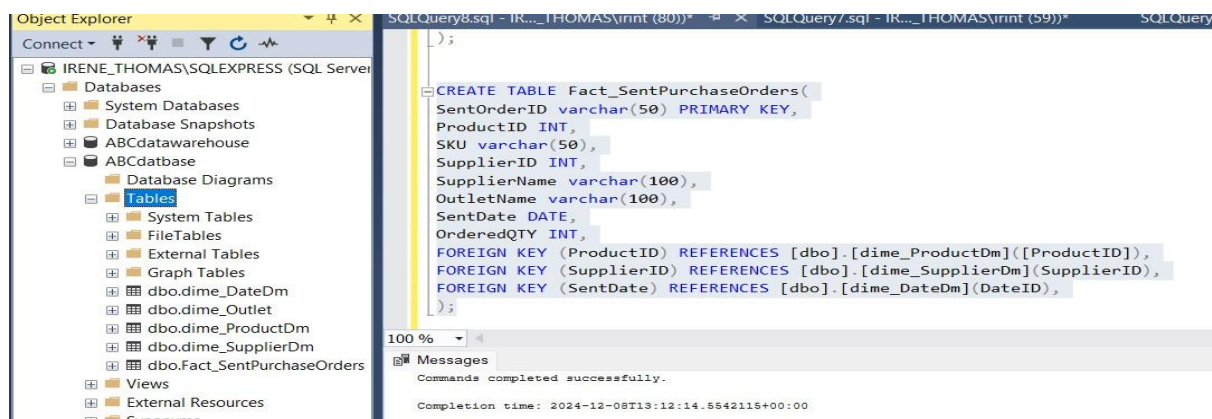


FIGURE 1.6

Purchase Order Received: This fact table has all the information regarding the stock that has been sent by the supplier and received by the company. There are 3 foreign keys which are DateID, ProductID, SupplierID, OutletID along with other attributes relevant to store the received purchase orders. The query to create a received order purchase is

```
CREATE TABLE Fact_RecievedPurchaseOrders(

PurchaseOrderID INT PRIMARY KEY,

ProductID INT,
```

```

SupplierID INT,

DateID Date,

OutletName varchar(200) ,

SentDate DATE,

RecievedDate DATE,

RecievedQTY INT,

OrderedQTY INT,

SupplierName Varchar(200),

SKU varchar(50),

FOREIGN KEY (ProductID) REFERENCES [dbo].[dime_ProductDm](ProductID),

FOREIGN KEY (SupplierID) REFERENCES [dbo].[dime_SupplierDm](SupplierID), FOREIGN KEY

(DateID) REFERENCES [dbo].[dime_DateDm](DateID),

);

```

After executing the command the output will look like FIGURE 1.7

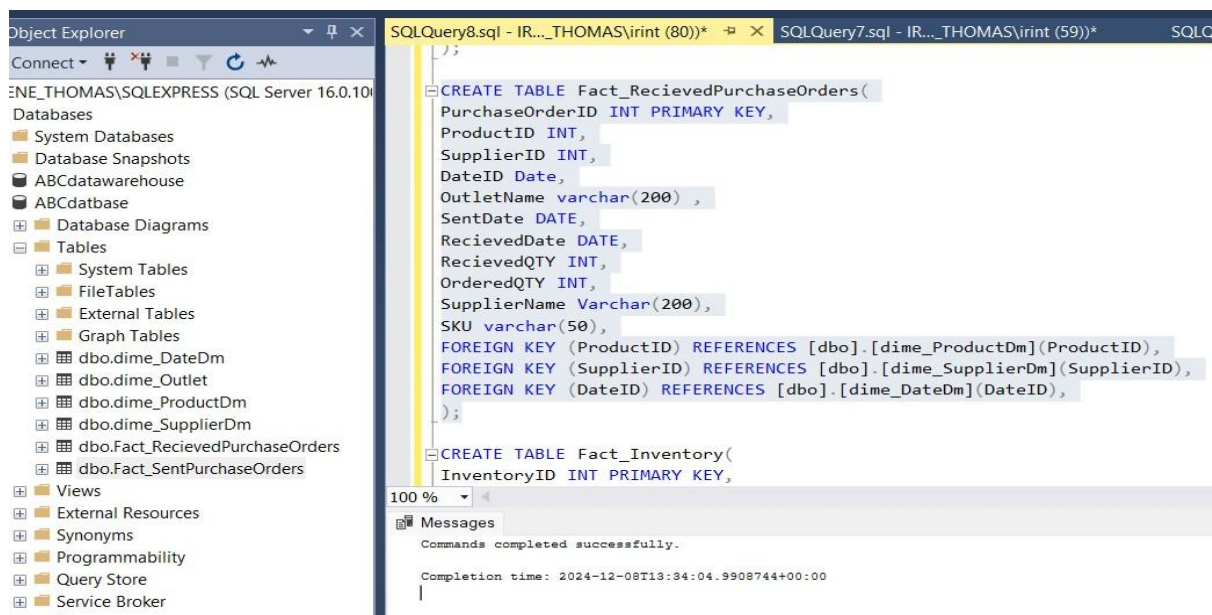


FIGURE 1.7

Inventory : This fact table has all the information about the stock that is present and keeps track on the stock movements. The primary key in the table is Inventory Id and the foreign keys are ProductID, DateID, OutletID along with others columns to store information about the stocks. The query to create a inventory fact table is

```

CREATE TABLE Fact_Inventory(

InventoryID INT PRIMARY KEY,

```

```

ProductID INT,

DateID Date,

OutletID INT,

CurrentStockLevel INT,

CostPrice DECIMAL(10, 2),

RetailPrice DECIMAL(10, 2),

DateCreatedAt Date,

IsActive BIT,

FOREIGN KEY (ProductID) REFERENCES [dbo].[dime_ProductDm](ProductID),

FOREIGN KEY (DateID) REFERENCES [dbo].[dime_DateDm](DateID),

FOREIGN KEY(OutletID) REFERENCES [dbo].[dime_Outlet],

);

```

After execution the output looks like FIGURE 1.8

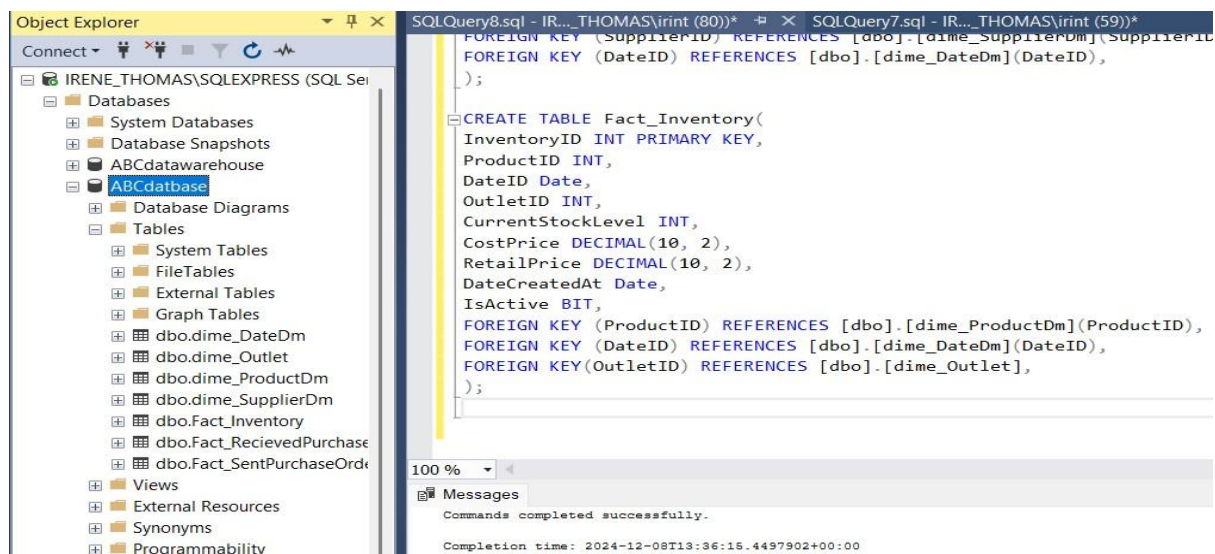


FIGURE 1.8

Population:-

After creating the database now we can test it by loading some data into it. We have to create a staging area to populate all the data into the tables. We have to populate the data by importing the sample data that has been given with the coursework. We use SQL wizard import to import all the data.

- **Creating Staging Area:** The staging area has to be created to import the data from sample. To create a staging area another database has to be created to import all

the data. The query to create a database remains the same that we used to create our original database. After creation, the output looks like FIGURE 1.9

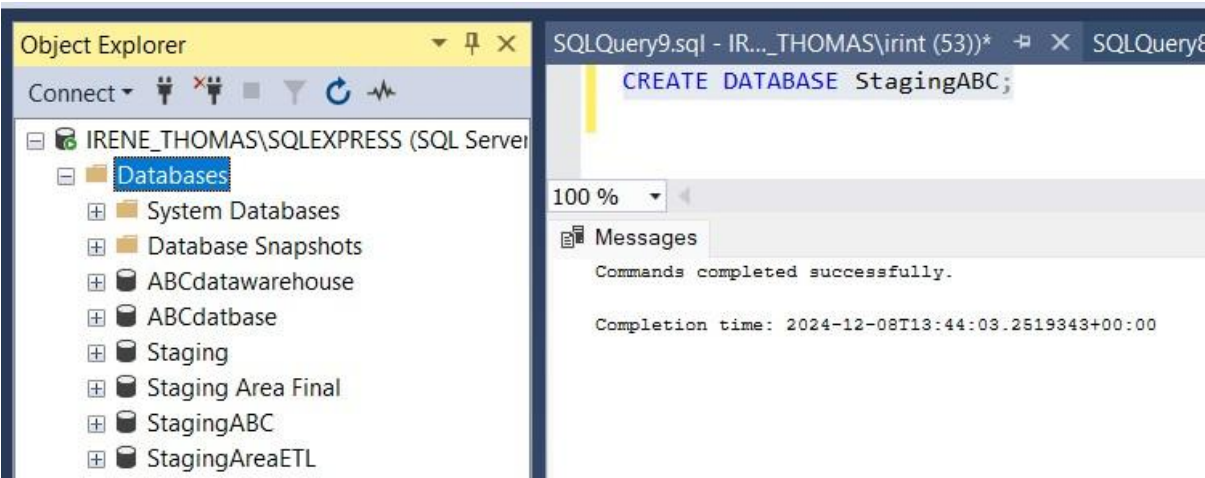


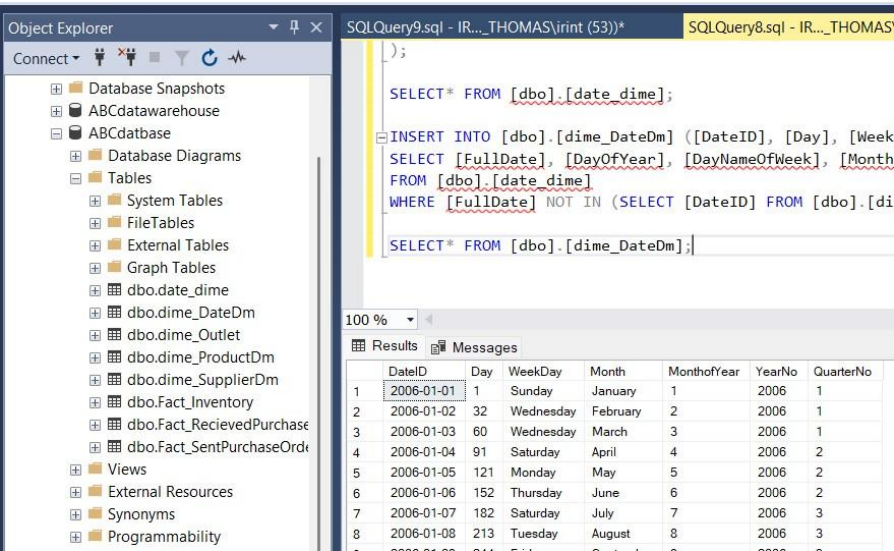
FIGURE 2.0

Then we have to populate each dimension table one by one

- **Date Dimension:** The Date dimension txt file was already provided in the coursework which was then used to populate the date dimension in the ABCdatabse . There were certain columns in the txt file that were not needed so we decided to select those needed by passing the SQL code.

```
INSERT INTO [dbo].[dime_DateDm] ([DateID], [Day], [WeekDay], [Month],
[MonthofYear], [YearNo], [QuarterNo])
SELECT [FullDate], [DayOfYear], [DayNameOfWeek], [MonthName], [MonthOfYear],
[CalendarYear], [CalendarQuarter]
FROM [dbo].[date_dime]
WHERE [FullDate] NOT IN (SELECT [DateID] FROM [dbo].[dime_DateDm]);
SELECT* FROM [dbo].[dime_DateDm];
```

This query displays the table after population which looked like FIGURE 2.1



- **Product Dimension:** We have 3 sample data which store the data from the product table which is combined into one for easy processing. After combining all three data files it has to be populated using the SQL Wizard. After populating the data a new table will be created at the destination “product table” which will then be used to insert and fetch the columns needed to be stored in the dimensions. The query used to insert the tables is:

```

INSERT INTO [dbo].[dime_ProductDm]([SKU], [ProductName],
[ProductDescription], [Condition],[ProductType], [Brand], [SupplierName])

SELECT [SKU], [ProductName], [Description], [Condition], [ProductType], [Brand],
[SupplierName]

FROM [dbo].[product table]

WHERE [SKU] NOT IN (SELECT [SKU] FROM [dbo].[dime_ProductDm]);

SELECT* FROM [dbo].[dime_ProductDm];

```

This query displays the table after population which looked like FIGURE 2.2

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure, including tables like `dbo.date_time`, `dbo.dime_DateDm`, `dbo.dime_Outlet`, `dbo.dime_ProductDm`, `dbo.dime_SupplierDm`, and `dbo.Fact_Inventory`. The central query window shows the following SQL script:

```

FROM [dbo].[date_time]
WHERE [FullDate] NOT IN (SELECT [DateID] FROM [dbo].[dime_DateDm]);

SELECT* FROM [dbo].[dime_DateDm];
SELECT* FROM [dbo].[product table];

INSERT INTO [dbo].[dime_ProductDm]([SKU], [ProductName], [ProductDescription], [Condition],[ProductType], [Brand]
SELECT [SKU], [ProductName], [Description], [Condition], [ProductType], [Brand], [SupplierName]
FROM [dbo].[product table]
WHERE [SKU] NOT IN (SELECT [SKU] FROM [dbo].[dime_ProductDm]);

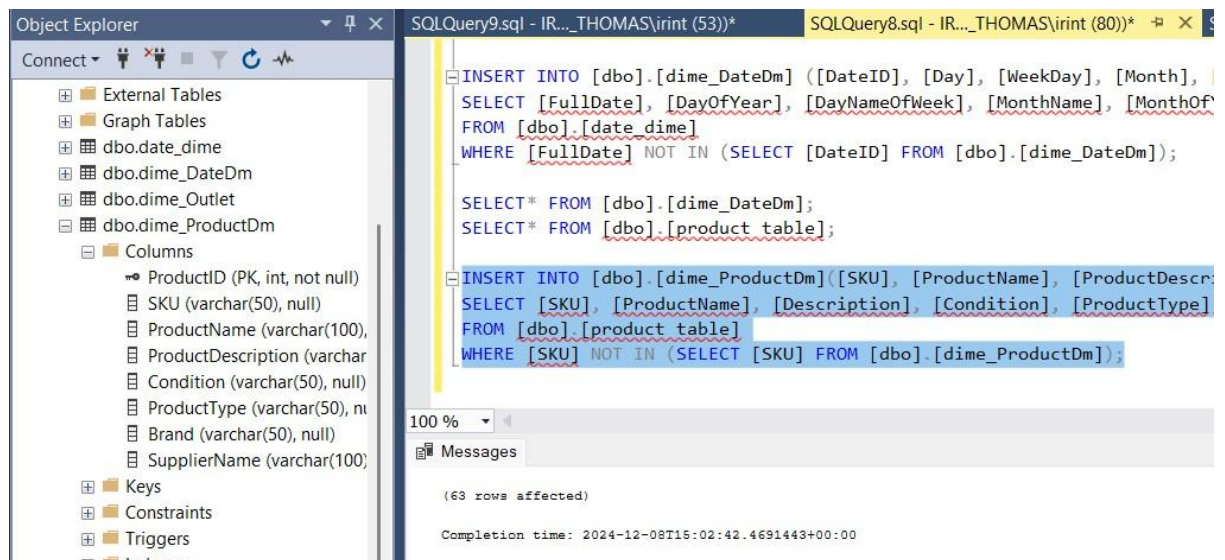
SELECT* FROM [dbo].[dime_ProductDm];

```

At the bottom, the Results pane shows a grid with 7 rows of data:

ProductID	SKU	ProductName	ProductDescription	Condition	ProductType	Brand	SupplierName
1	SEN23322	Manfrotto MN1004BAC Master Light Stand	Master Light Stand	Display	ACCESSORY	SENNHEISER	SENNHEISER
2	SO6677	Manfrotto MT057C3 Carbon Fibre 3 Section Geared	Carbon Fibre 3 Section Geared	Display	ACCESSORY	SONY	SONY
3	JV2222	Rycote 37705 Portable Recorder Suspension	Portable Recorder Suspension	New	CAMCORDER	JVC	JVC
4	SEN222	Hoya 375-HOY 37MM SKYLIGHT FILTER Hoya	37MM SKYLIGHT FILTER Hoya	New	IMAGING	SENNHEISER	SENNHEISER
5	TOW222	HOYA 40.5mm CP Filter - Slim	HOYA 40.5mm CP Filter - Slim	New	IMAGING	TOSHIBA	TOSHIBA
6	MS7771	Rycote 41118 Portable Recorder Suspension	Portable Recorder Suspension	New	CAMCORDER	MSCS	MSCS
7	CO8211	Rycote 41126 Invision Video Hotshoe	Invision Video Hotshoe	New	CAMCORDER	TOSHIBA	TOSHIBA

FIGURE 2.2



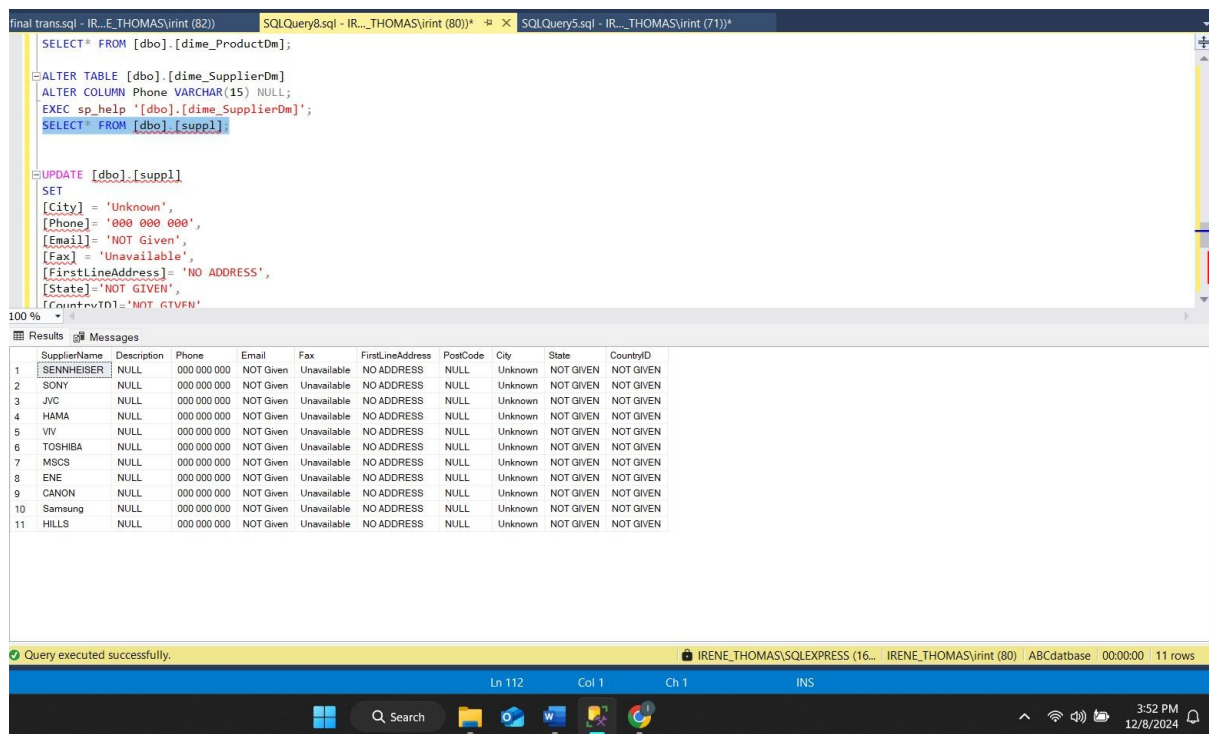
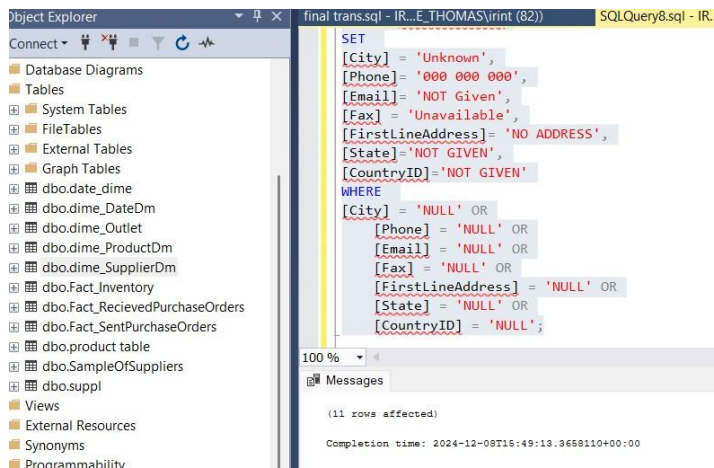
- **Supplier Dimension:** The data for the supplier dimension has to be populated by the SQL Wizard with the txt file that has been provided to us in the coursework. Once populated it will create a temporary table in the database ABCdatabase , which will be used to populate the original dimension table by fetching the specific values needed for the business activity. The supplier table had a lot of null values which need to be replaced with default values so a sql code was made to update the rows in the temporary table :

```

UPDATE [dbo].[suppl]
SET
[City] = 'Unknown',
[Phone]= '000 000 000',
[Email]= 'NOT Given',
[Fax] = 'Unavailable',
[FirstLineAddress]= 'NO ADDRESS',
[State]='NOT GIVEN',
[CountryID]='NOT GIVEN'
WHERE
[City] = 'NULL' OR
[Phone] = 'NULL' OR
[Email] = 'NULL' OR
[Fax] = 'NULL' OR
[FirstLineAddress] = 'NULL' OR
[State] = 'NULL' OR
[CountryID] = 'NULL';

```

This will create a table like FIGURE 2.3



Then it is inserted into the original dimesion table by using the code:

```
INSERT INTO [dbo].[dime_SupplierDm] ([SupplierName], [Phone], [Email], [FAX], [Address], [PostCode], [City], [State], [CountryID])
```

```
SELECT [SupplierName], [Phone], [Email], [Fax], [FirstLineAddress], [PostCode], [City], [State], [CountryID]
FROM [dbo].[suppl]
```

```
WHERE [SupplierName] NOT IN (SELECT [SupplierName] FROM [dbo].[dime_SupplierDm]);
```

This code will execute and give an output in FIGURE 2.3

The screenshot shows a SQL query window with the following code:

```

[State] = 'NULL' OR
[CountryID] = 'NULL';

INSERT INTO [dbo].[dime_SupplierDm] ([SupplierName], [Phone], [Email], [FAX], [Address], [PostCode], [
SELECT [SupplierName], [Phone], [Email], [Fax], [FirstLineAddress], [PostCode], [City], [State], [Coun
FROM [dbo].[suppl]
WHERE [SupplierName] NOT IN (SELECT [SupplierName] FROM [dbo].[dime_SupplierDm]);

SELECT* FROM [dbo].[dime_SupplierDm];

```

The results grid displays the following data:

	SupplierID	SupplierName	Phone	Email	FAX	Address	PostCode	City	State	CountryID
1	1	SENNHEISER	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
2	2	SONY	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
3	3	JVC	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
4	4	HAMA	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
5	5	VIV	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
6	6	TOSHIBA	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
7	7	MSCS	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
8	8	ENE	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN
9	9	CANON	000 000 000	NOT Given	Unavailable	NO ADDRESS	NULL	Unknown	NOT GIVEN	NOT GIVEN

FIGURE 2.3

- **Outlet Dimension:** The outlet table stores all the information about the outlet from which the stock movement goes on . So in order to populate the data we have to insert data of on of the fact tables into a temporary table because we don't have a separate file with outlet information. And as there is only one outlet information given in all the fact table files we can use any on of the. After populating the temporary table we fetch the outlet name and insert it into out original outlet dimension table by using a SQL code :

```

INSERT INTO [dbo].[dime_Outlet]([OutletName])

SELECT DISTINCT [DestinationOutletID]

FROM [dbo].[Sampleforoutlet]

WHERE [DestinationOutletID] NOT IN (SELECT [OutletName] FROM [dbo].[dime_Outlet]);

Select * FROM [dbo].[dime_Outlet];

```

This code will give us an output as FIGURE 2.4

The screenshot shows a SQL query window with the following code:

```

INSERT INTO [dbo].[dime_Outlet]([OutletName])
SELECT DISTINCT [DestinationOutletID]
FROM [dbo].[Sampleforoutlet]
WHERE [DestinationOutletID] NOT IN (SELECT [OutletName] FROM [dbo].[dime_Outlet]);

Select * FROM [dbo].[dime_Outlet];

```

The results grid displays the following data:

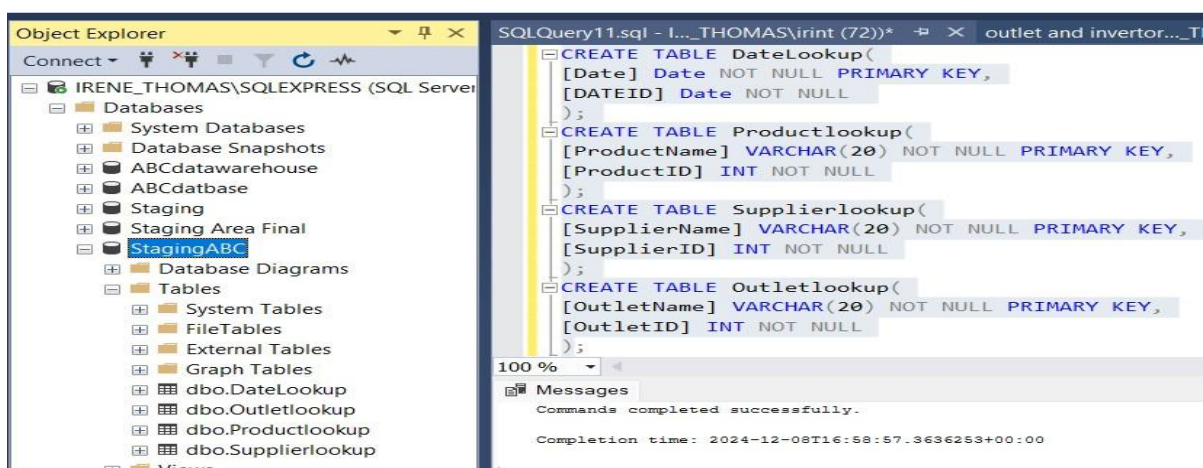
	OutletID	OutletName	OutletAddress	OutletCity	OutletState	OutletCountry	OutletPostcode
1	13	ABC Warehouse	NULL	NULL	NULL	NULL	NULL

FACT TABLE POLPULATION:

After populating all the dimension tables we can now populate all the fact tables using the staging area. After that we have to create tables for each dimension table on the staging area with the code :

```
CREATE TABLE DateLookup(  
  
[Date] Date NOT NULL PRIMARY KEY,  
  
[DATEID] Date NOT NULL  
  
);  
  
CREATE TABLE Productlookup(  
  
[ProductName] VARCHAR(20) NOT NULL PRIMARY KEY,  
  
[ProductID] INT NOT NULL  
  
);  
  
CREATE TABLE Supplierlookup(  
  
[SupplierName] VARCHAR(20) NOT NULL PRIMARY KEY,  
  
[SupplierID] INT NOT NULL  
  
);  
  
CREATE TABLE Outletlookup(  
  
[OutletName] VARCHAR(20) NOT NULL PRIMARY KEY,  
  
[OutletID] INT NOT NULL  
  
);
```

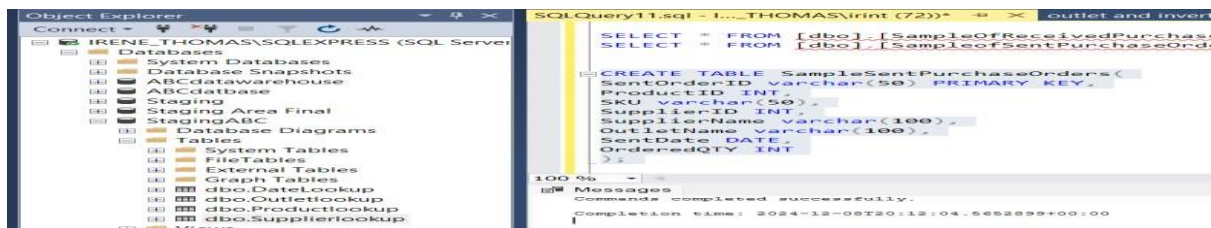
This code will create a table in staging area for date with DATE



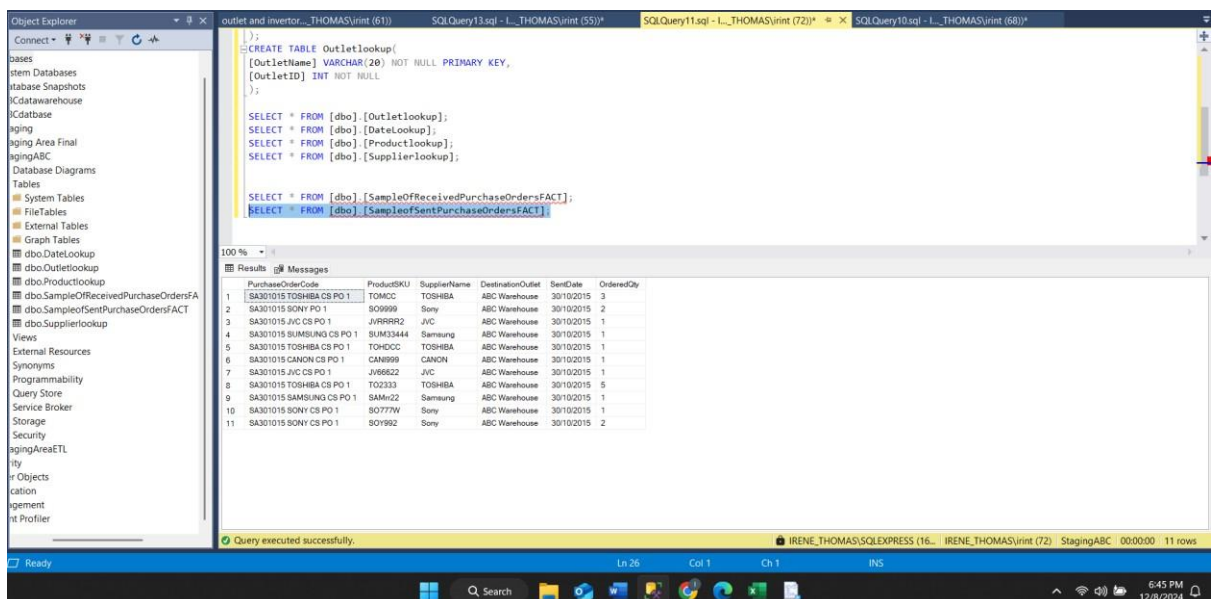
FACT table Purchase Order Sent: Created a sample fact table on the staging area using the sql code

```
CREATE TABLE SampleSentPurchaseOrders(  
  
SentOrderID varchar(50) PRIMARY KEY,
```

ProductID INT,
SKU varchar(50),
SupplierID INT,
SupplierName varchar(100),
OutletName varchar(100),
SentDate DATE,
OrderedQTY INT
);



The fact tables are populated using the flat files provided to us using SQL Wizard.



FACT table Purchase Order Recieved: Created a sample fact table on the staging area using the sql code

```
CREATE TABLE SampleRecievedPurchaseOrders(
PurchaseOrderID INT PRIMARY KEY,
ProductID INT,
SupplierID INT,
```

```

DateID Date,

OutletName varchar(200) ,

SentDate DATE,

RecievedDate DATE,

RecievedQTY INT,

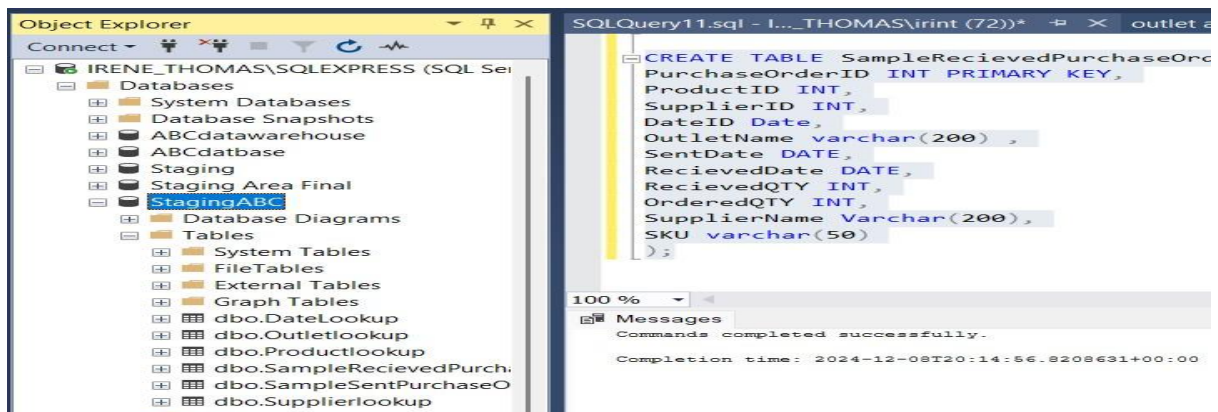
OrderedQTY INT,

SupplierName Varchar(200),

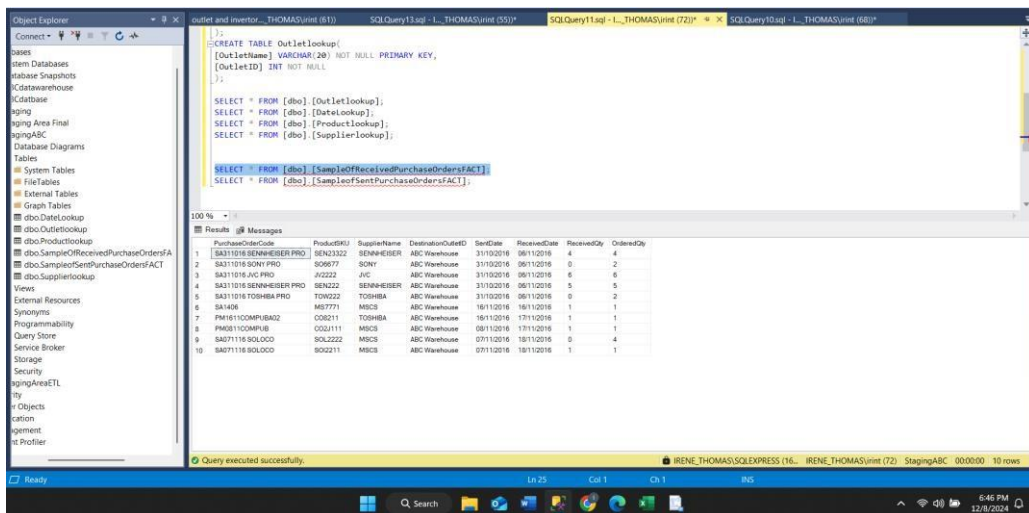
SKU varchar(50)

);

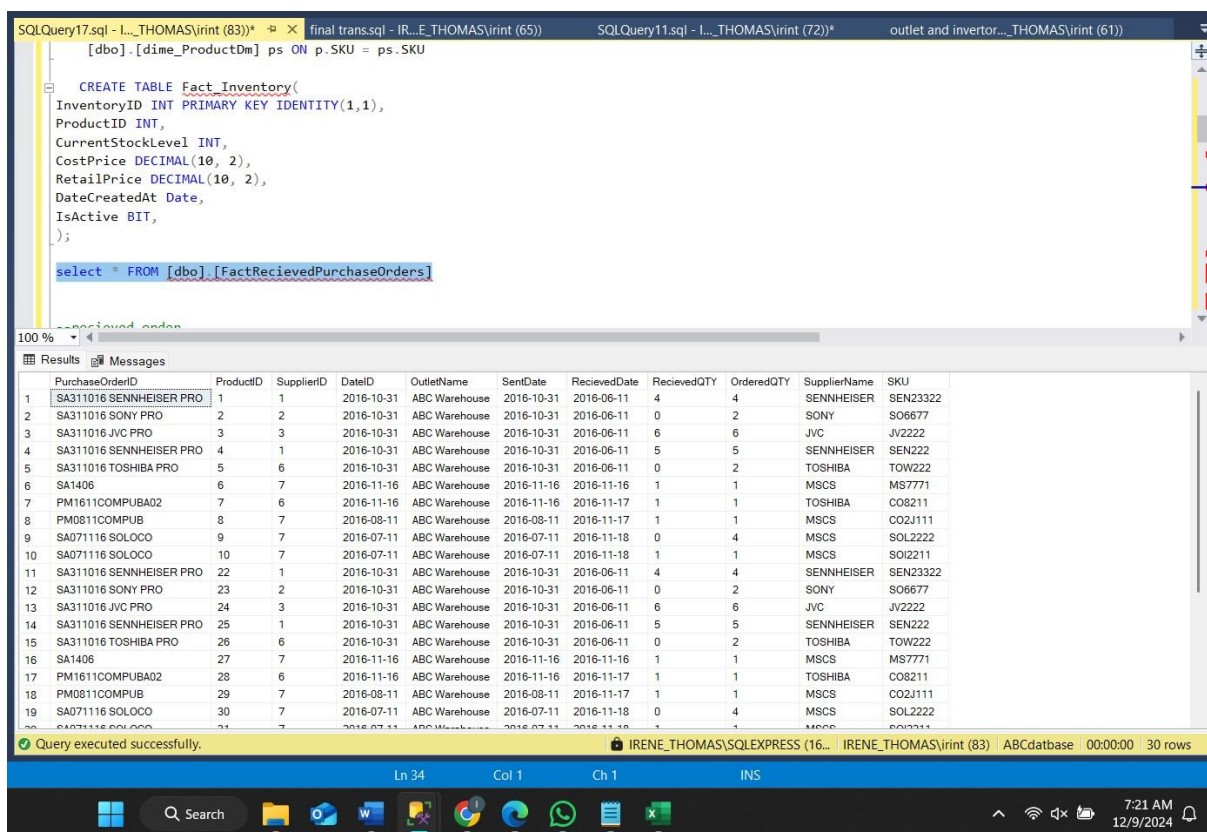
```



The fact tables are populated using the flat files provided to us using SQL Wizard.



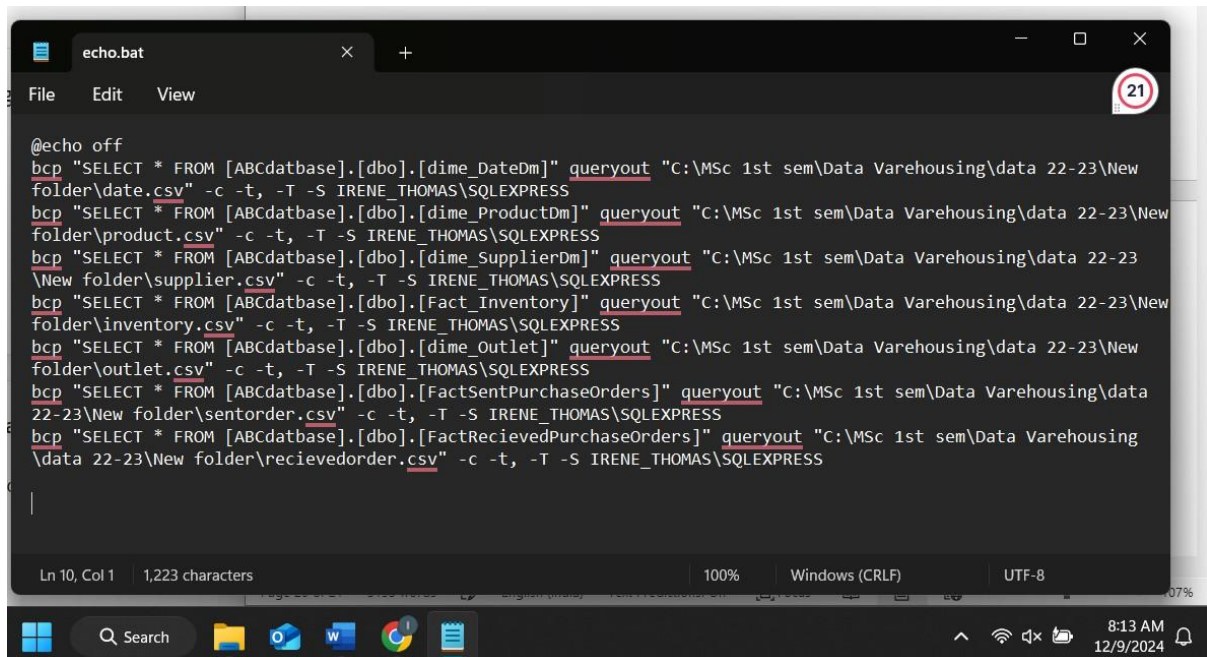
Fact Inventory: The inventory table was populated from fetching the data from the product dimension attributes which gave the output



Extracting data from the tables and dimensions into a .csv files:

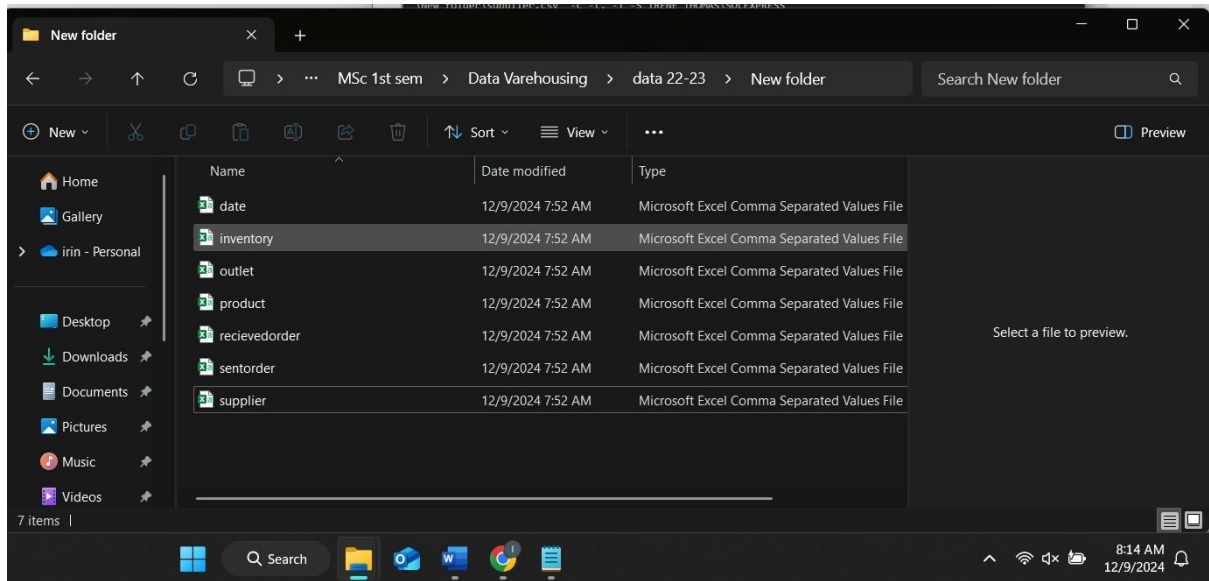
A batch file (ExportTables.bat) was created to automate the export of data from the database tables to .csv files. The script used the Bulk Copy Program (BCP) command to export data from each table. The BCP commands in the batch file specify the source table, the destination file, and the database server. For instance:

```
bcp "SELECT * FROM [ABCdatabase].[dbo].[FactRecievedPurchaseOrders]" queryout "C:\MSc 1st sem\Data  
Varehousing\data 22-23\New folder\recievedorder.csv" -c -t, -T -S  
IRENE_THOMAS\SQLEXPRESS
```



```
@echo off
bcp "SELECT * FROM [ABCdatabase].[dbo].[dime_DateDm]" queryout "C:\MSc 1st sem\Data Varehousing\data 22-23\New
folder\date.csv" -c -t, -T -S IRENE_THOMAS\SQLEXPRESS
bcp "SELECT * FROM [ABCdatabase].[dbo].[dime_ProductDm]" queryout "C:\MSc 1st sem\Data Varehousing\data 22-23\New
folder\product.csv" -c -t, -T -S IRENE_THOMAS\SQLEXPRESS
bcp "SELECT * FROM [ABCdatabase].[dbo].[dime_SupplierDm]" queryout "C:\MSc 1st sem\Data Varehousing\data 22-23
\New folder\supplier.csv" -c -t, -T -S IRENE_THOMAS\SQLEXPRESS
bcp "SELECT * FROM [ABCdatabase].[dbo].[Fact_Inventory]" queryout "C:\MSc 1st sem\Data Varehousing\data 22-23\New
folder\inventory.csv" -c -t, -T -S IRENE_THOMAS\SQLEXPRESS
bcp "SELECT * FROM [ABCdatabase].[dbo].[dime_Outlet]" queryout "C:\MSc 1st sem\Data Varehousing\data 22-23\New
folder\outlet.csv" -c -t, -T -S IRENE_THOMAS\SQLEXPRESS
bcp "SELECT * FROM [ABCdatabase].[dbo].[FactSentPurchaseOrders]" queryout "C:\MSc 1st sem\Data Varehousing\data
22-23\New folder\sentorder.csv" -c -t, -T -S IRENE_THOMAS\SQLEXPRESS
bcp "SELECT * FROM [ABCdatabase].[dbo].[FactRecievedPurchaseOrders]" queryout "C:\MSc 1st sem\Data Varehousing
\data 22-23\New folder\recievedorder.csv" -c -t, -T -S IRENE_THOMAS\SQLEXPRESS
```

Upon running the batch file in the command prompt, logs indicated successful export operations for each table. Messages like "1 rows copied" and "Clock Time (ms): 1" confirm the transfer of data rows from each table to their respective .csv files.



Excel spreadsheet showing a date breakdown table:

	A	B	C	D	E	F	G	H	I	J
1	Date	Day	WeekDay	Month	MonthNo	Quarter	Year			
2	1/1/2006	1	Sunday	January	1	2006	1			
3	1/2/2006	32	Wednesda	February	2	2006	1			
4	1/3/2006	60	Wednesda	March	3	2006	1			
5	1/4/2006	91	Saturday	April	4	2006	2			
6	1/5/2006	121	Monday	May	5	2006	2			
7	1/6/2006	152	Thursday	June	6	2006	2			
8	1/7/2006	182	Saturday	July	7	2006	3			
9	1/8/2006	213	Tuesday	August	8	2006	3			
10	1/9/2006	244	Friday	September	9	2006	3			

AutoSave Off inventory

File Home Insert Page Formu Data Review View Auton Help

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles

A1 InventoryID

	A	B	C	D	E	F	G	H
1	InventoryID	ProductID	CurrentStock	CostPrice	RetailPrice	DateCreated	IsActive	
2	1	1	2	57	114	#####	1	
3	2	22	2	57	114	#####	1	
4	3	43	2	57	114	#####	1	
5	4	2	5	298	584	#####	1	
6	5	23	5	298	584	#####	1	
7	6	44	5	298	584	#####	1	
8	7	3	6	0	59	#####	1	

AutoSave Off outlet

File Home Insert Page L Formu Data Review View Autom Help




Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles

A1 OutletID


	A	B	C	D	E	F	G	H
1	OutletID	OutletName	Address	City	State	Country	Postcode	
2	13	ABC Warehouse						
3								


AutoSave Off prod... Saved to t... IT										
File Home Insert Page Layout Formulas Data Review View Automate Help										
Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles Cells Editing Add-ins Add-ins										
A1 ProductID										
	A	B	C	D	E	F	G	H	I	J
1	ProductID	ProductSK	ProductNa	ProductDe	Condition	ProductTy	Brand	SupplierName		
2	1	SEN23322	Manfrotto I	Master Lig	Display	ACCESSOF	SENNHEIS	SENNHEISER		
3	2	SO6677	Manfrotto I	Carbon Fil	Display	ACCESSOF	SONY	SONY		
4	3	JV2222	Rycote 377	Portable R	New	CAMCORD	JVC	JVC		
5	4	SEN222	Hoya 37S-I	37MM SKY	New	IMAGING	SENNHEIS	SENNHEISER		
6	5	TOW222	HOYA 40.5	HOYA 40.5	New	IMAGING	TOSHIBA	TOSHIBA		
7	6	MS7771	Rycote 411	Portable R	New	CAMCORD	MSCS	MSCS		
8	7	CO8211	Rycote 411	Invision Vic	New	CAMCORD	TOSHIBA	TOSHIBA		
9	8	CO2J111	Rycote 411	Invision Vic	New	CAMCORD	MSCS	MSCS		
10	9	SOL2222	Rycote 411	Invision Vic	New	CAMCORD	MSCS	MSCS		


AutoSave Off





recievedorder • Saved to this PC















FileHomeInsertPage LayoutFormulasDataReviewViewAutomateHelp


Comments




Paste
Clipboard



Font



Alignment


Number

Conditional Formatting
Format as Table
Cell Styles


Styles


Cells





Editing

Add-ins

Add-ins


Analyze Data

A1



PurchaseOrderID

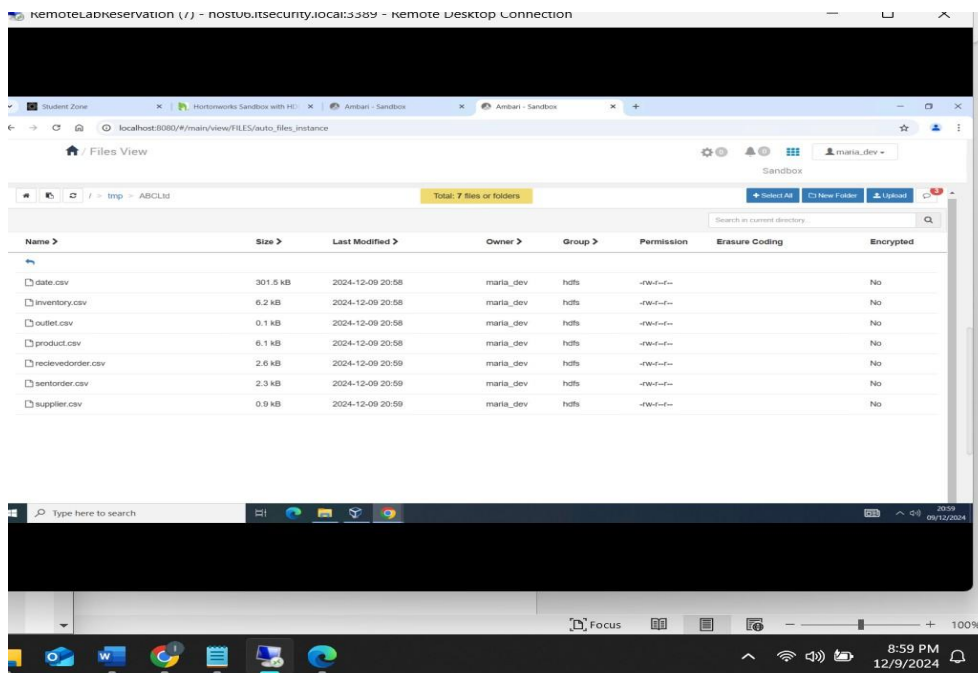
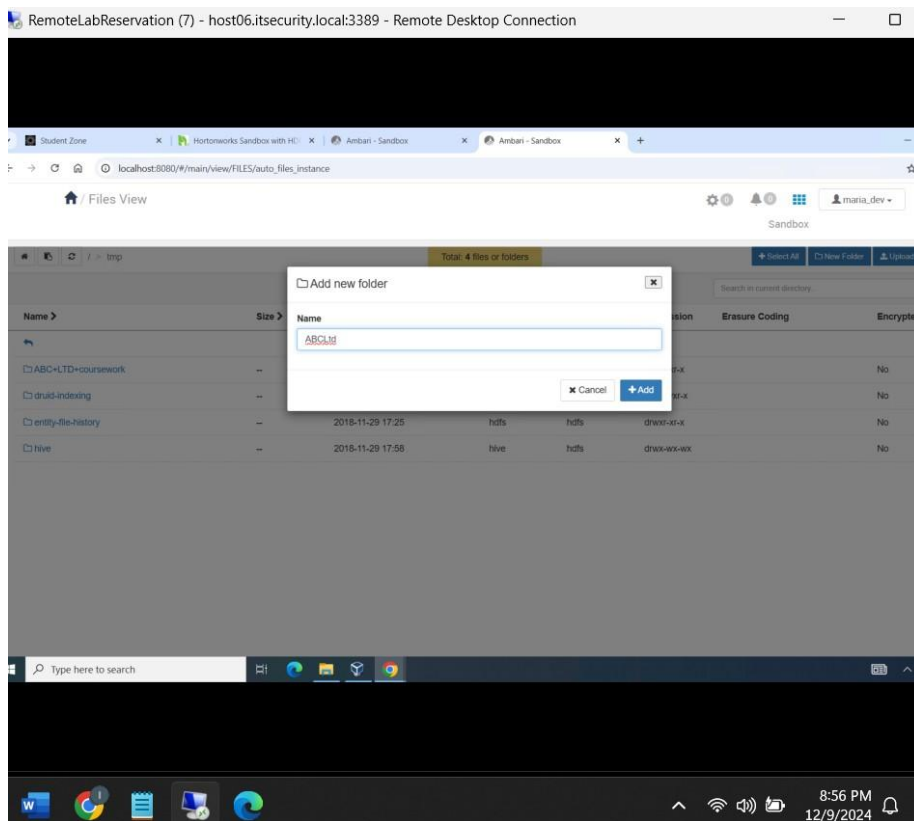
	A	B	C	D	E	F	G	H	I	J	K	L
1	PurchaseC	ProductID	SupplierID	DateID	OutletNam	SentDate	RecievedD	RecievedQ	Ordered Q	SupplierN	SKU	
2	SA311016	1	1	#####	ABC Ware	#####	#####	4	4	SENNHEIS	SEN23322	
3	SA311016	2	2	#####	ABC Ware	#####	#####	0	2	SONY	SO6677	
4	SA311016	3	3	#####	ABC Ware	#####	#####	6	6	JVC	JV2222	
5	SA311016	4	1	#####	ABC Ware	#####	#####	5	5	SENNHEIS	SEN222	
6	SA311016	5	6	#####	ABC Ware	#####	#####	0	2	TOSHIBA	TOW222	
7	SA1406	6	7	#####	ABC Ware	#####	#####	1	1	MSCS	MS7771	

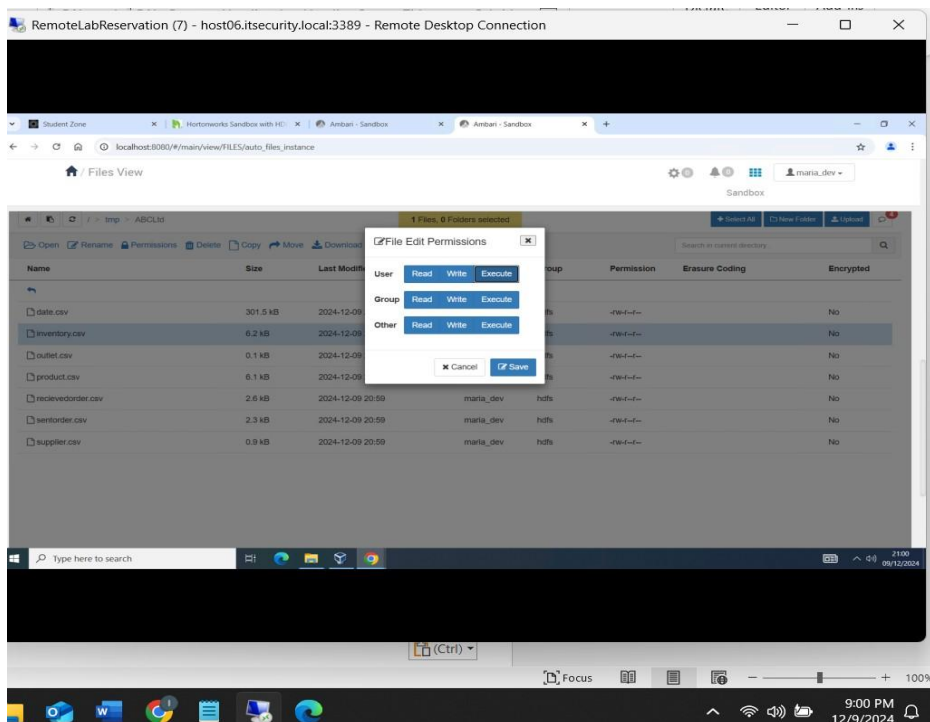
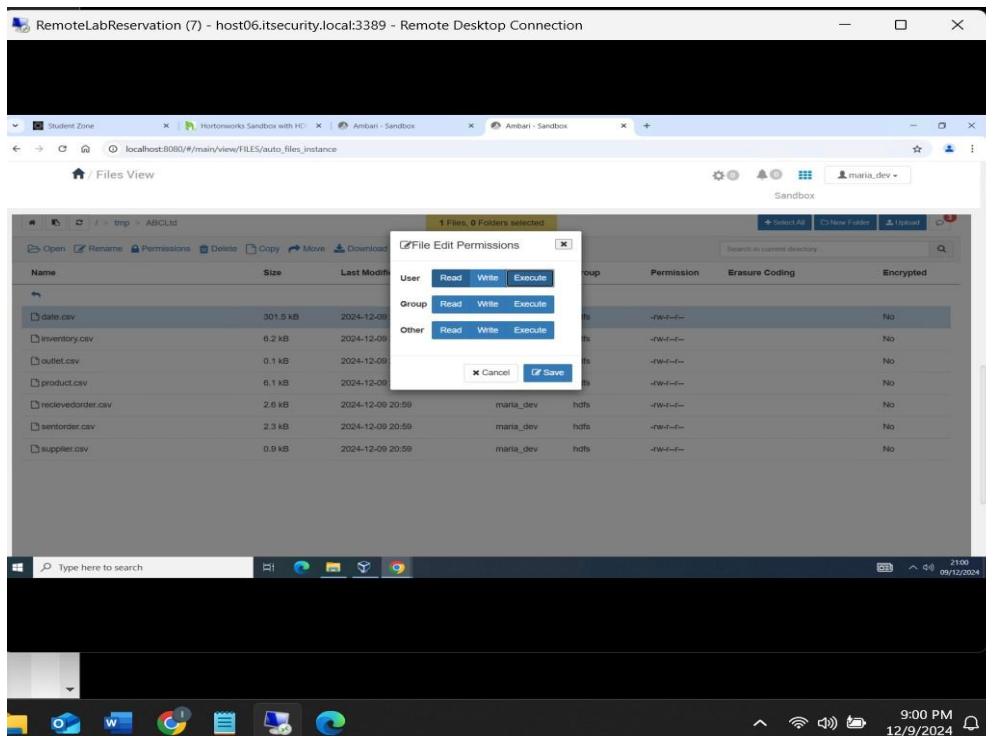
	PurchaseOrderID	ProductID	ProductSKU	SupplierName	OutletName	SentDate	Ordered QTY
1	SA301015	11	TOMCC	6 TOSHIBA	ABC Ware	10/30/2015	3
2	SA301015	12	SO9999	2 SONY	ABC Ware	10/30/2015	2
3	SA301015	13	JVRRRR2	3 JVC	ABC Ware	10/30/2015	1
4	SA301015	14	SUM33444	10 Samsung	ABC Ware	10/30/2015	1
5	SA301015	15	TOHDCC	6 TOSHIBA	ABC Ware	10/30/2015	1
6	SA301015	16	CANI999	9 CANON	ABC Ware	10/30/2015	1
7	SA301015	17	JV66622	3 JVC	ABC Ware	10/30/2015	1
8	SA301015	18	TO2333	6 TOSHIBA	ABC Ware	10/30/2015	5
9	SA301015	19	SAMrr22	10 Samsung	ABC Ware	10/30/2015	1

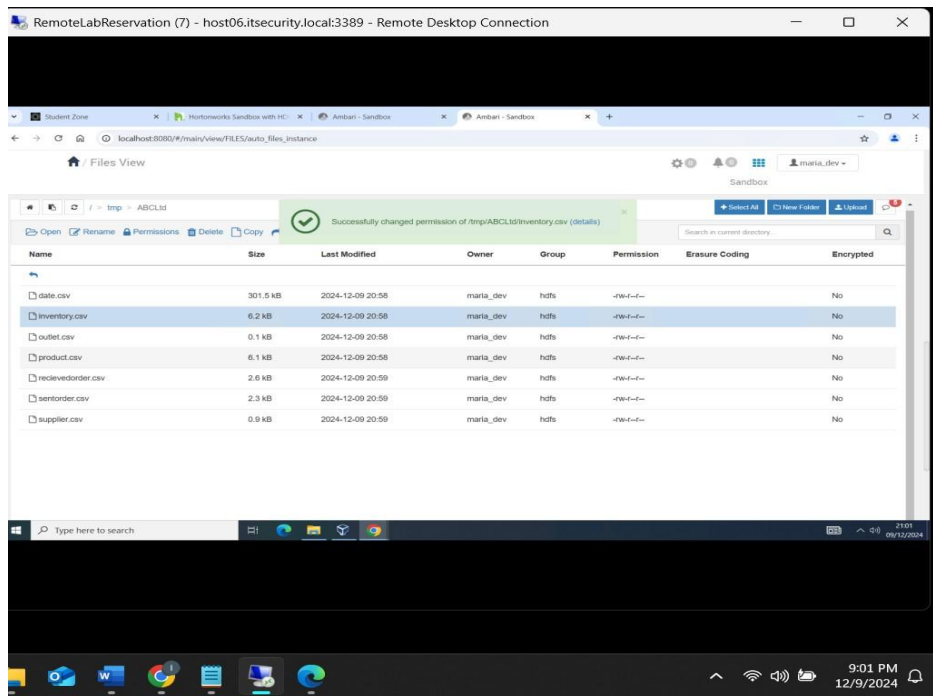
	SupplierID	SupplierName	Phone	Email	FAX	Address	Postcode	City	State	CountryID
1	1	SENNHEIS	000 000 00	NOT Given	Unavailabl	NO ADDRE	NULL	Unknown	NOT GIVEN	NOT GIVEN
2	2	SONY	000 000 00	NOT Given	Unavailabl	NO ADDRE	NULL	Unknown	NOT GIVEN	NOT GIVEN
3	3	JVC	000 000 00	NOT Given	Unavailabl	NO ADDRE	NULL	Unknown	NOT GIVEN	NOT GIVEN
4	4	HAMA	000 000 00	NOT Given	Unavailabl	NO ADDRE	NULL	Unknown	NOT GIVEN	NOT GIVEN
5	5	VIV	000 000 00	NOT Given	Unavailabl	NO ADDRE	NULL	Unknown	NOT GIVEN	NOT GIVEN

Migrating the data files from the file system to Apache HDFS.

All the extracted csv files were loaded into the hadoop by creating a new folder ABCItd and the all the permissions were selected.



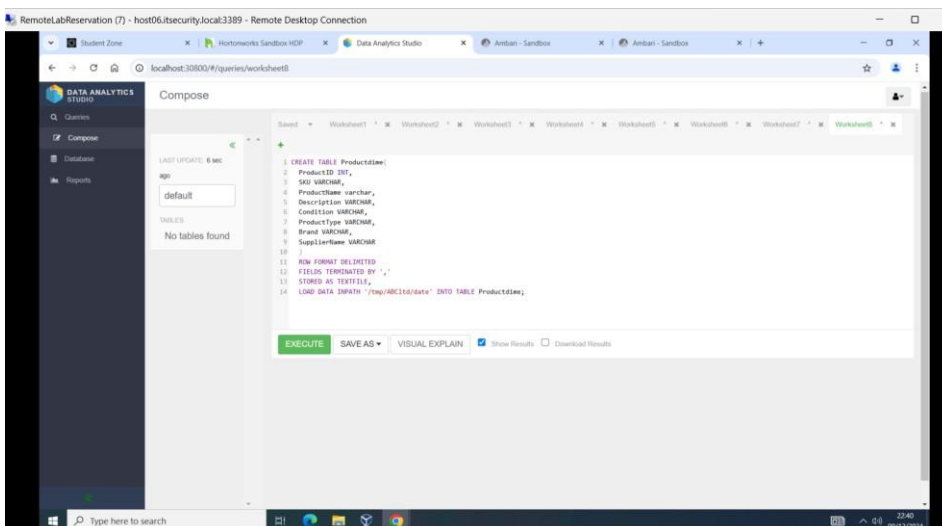




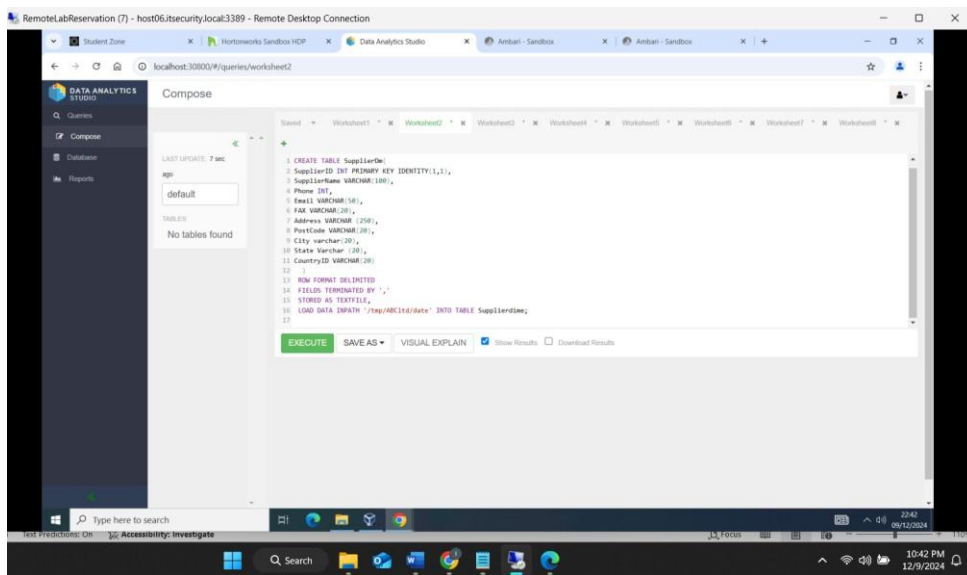
Creating data structure in Hadoop and loading data:

After loading data into the hadoop the next step was to create data structures to load the data into. External tables were created for each dataset using the HiveQL syntax to load the data files into Hive. Below are the details of the tables:

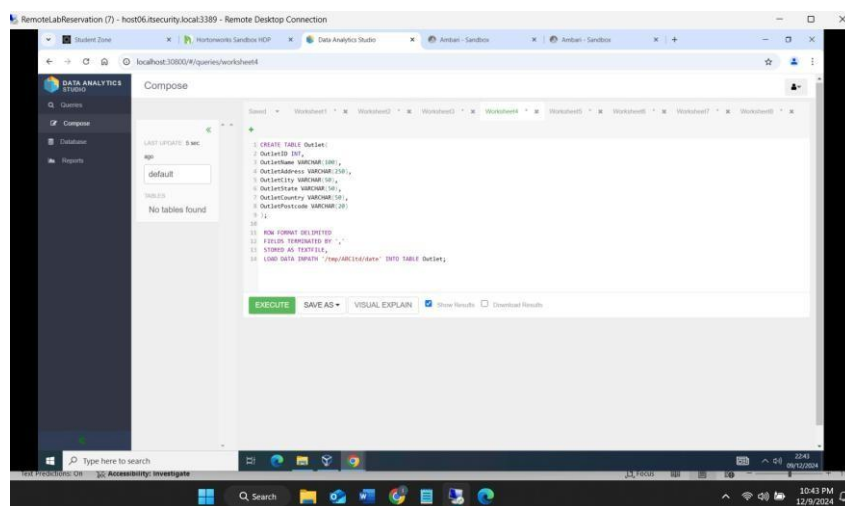
Product table: A product table was created which had all the similar columns as the data that we will be loading into the tables from product.csv file



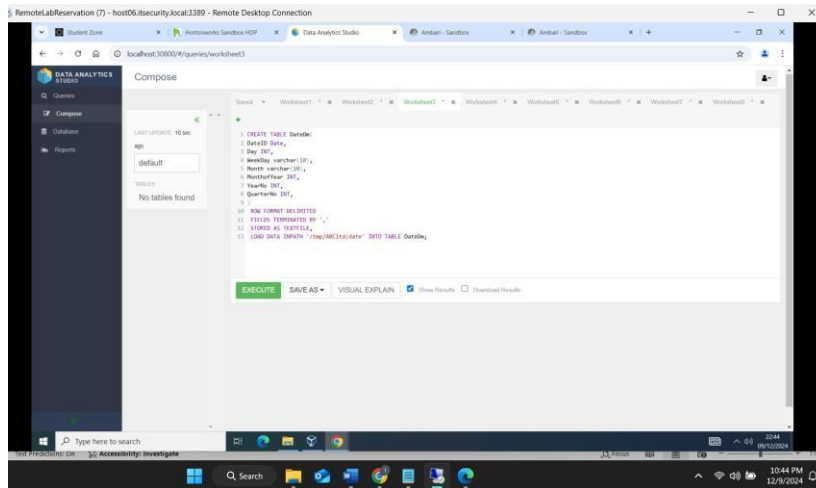
Supplier table: A supplier table was created which had all the similar columns as the data that we will be loading into the tables from the supplier.csv file



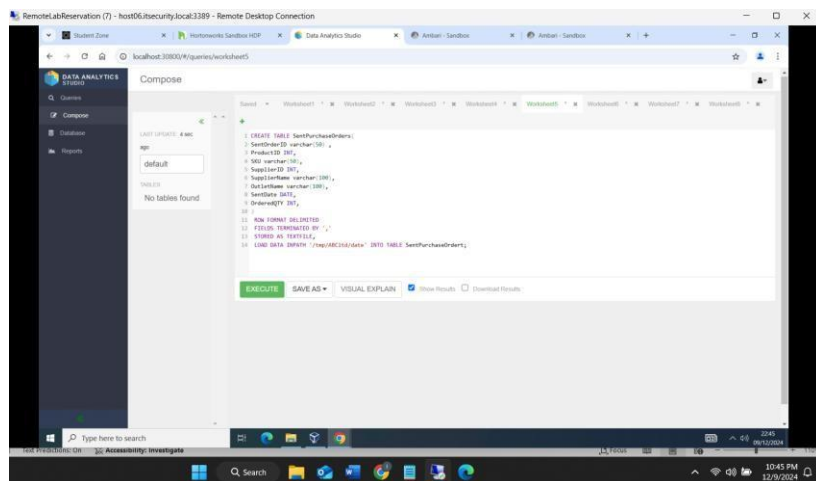
Outlet table: A outlet table was created which had all the similar columns as the data that we will be loading into the tables from the outlet.csv file.



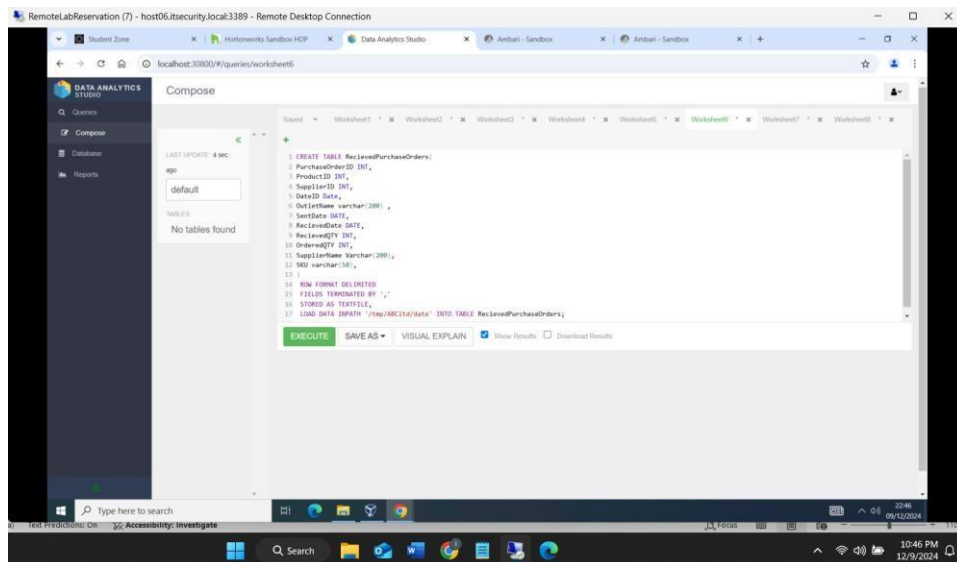
Date table: A date table was created which had all the similar columns as the data that we will be loading into the tables from the date.csv file.



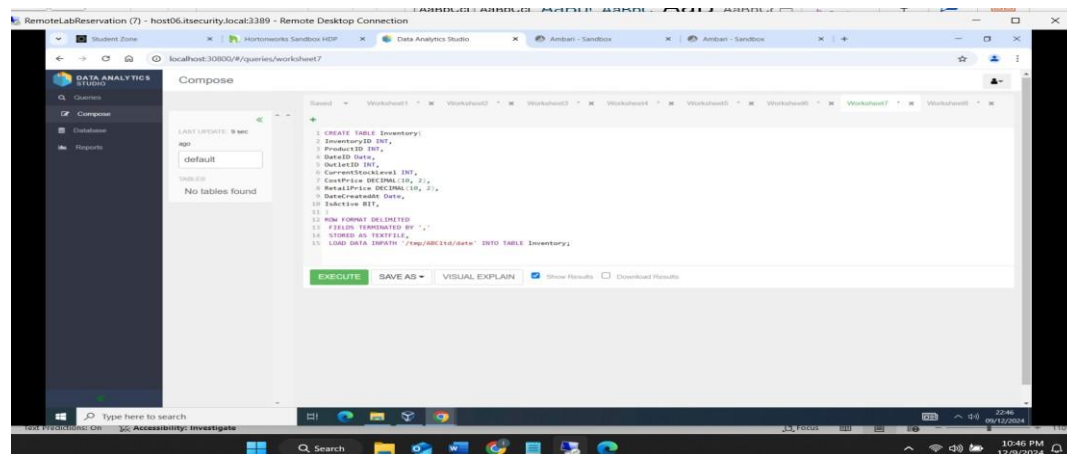
PurchaseOrderSent table: A table was created which had all the similar columns as the data that we will be loading into the tables from the sentorder.csv file.



PurchaseOrderRecieved table: A table was created which had all the similar columns as the data that we will be loading into the tables from the recieveorder.csv file.



Inventory table: A table was created which had all the similar columns as the data that we will be loading into the tables from the inventory.csv file.



Conclusion: The completion of this assignment has been a transformative journey, deepening my knowledge of facts integration and control within the realms of facts warehousing and relational databases. The hands-on experience of moving data from staging environments to the original database has enriched my technical skills and analytical skills.

Through this process, I received insights into the importance of defining primary key constraints and dealing with facts integrity. by using effectively handling demanding situations including NULL values and type conversion errors, I developed a eager awareness of how meticulous facts validation can prevent important problems throughout facts insertion using SQL Server for database control and enforcing strong transaction managing techniques allowed me to realize the importance of keeping data consistency and reliability.

Furthermore, this coursework highlighted the importance of facts integrity, suitable granularity, and effective schema layout in the building dependable data warehouse. My realistic experience with data translation, querying, and manipulation has significantly improved my capability to manipulate large datasets and extract significant insights through systematic methodologies. This combination of academic knowledge and hands-on skills has deepened my information of the essential role data warehousing and huge data generation play in addressing actual-world enterprise challenges, ultimately enhancing my problemsolving skills, attention to detail, and readiness to address complex facts tasks with confidence.