COURSEWORK - REPORT

Academic year 2022/2023 Autumn Term
CS7079 Data Warehousing and Big Data
PART A – Group Work

Abstract

The aim of this coursework is to design, implement, and test a data warehouse based on a given business case scenario, to export data from it and ingest it for further processing on a Big Data platform.

	ADEBAYO TAOFIK ADEKOYA (21053457)	PANKAJ KASHYAP (22012469)	KUNIK SUJIT NAYYAR (22018515)
Prepared By: Group 23	IBRAHIM ADEMOLA OYENEYE (22024597)	SAMRAT RAI (21025221)	SHAMSHER SINGH (22014018)
	SHRUTI ASHOKBHAI PATEL (22021294)	PRUTHVIKKUMAR ATULBHAI VAGHASIYA (ST_ID)	JEFERSON JEFERSON CARGUA BASTIDAS (ST_ID)

Background information

ABC consumer electronics outlet Ltd. (a multi-channel retailer of consumer electronics) has operations in six physical outlets with an online store. The company sells more than ten thousand products within ten categories and 200 brands. The company has already invested in information and communication technologies to run smooth operations.

Currently, the company uses a cloud-based programme Vend to support several of its retail activities. It also uses Linnworks to automate its internet operations and Xerox to help its financial department. Every day, these software programmes generate a massive amount of transactional data. Each software programme stores and manages the generated datasets independently.

Managers at ABC Consumer Electronics Retail Company are struggling to prepare reports and conduct studies that will allow them to deliver information for decision-making.

As a result, a business analyst advised the organisation to build and install a data warehouse that would suit all reporting and analytical needs.

Introduction

The data warehouse solution will exclusively focus on the inventory management business process. As reported by managers, the company is facing difficulties in controlling over-stocks and under-stocks of product items. The three main business activities which are considered under inventory management are:

1. Send purchase orders to suppliers when product items have low inventory levels. The ordered amount sent date, product and supplier data are all logged for each sent purchase order.
2. Receiving purchase orders and storing inventory in proper locations. The ordered amount, received quantity, received date, product and supplier data are all recorded for each received purchase order.
3. Stock control and maintenance entail introducing new items and modifying stock levels of current products. Each product's data, including the current stock level, are recorded.

Where:

Inventory managers are under pressure to satisfy the following reporting and analytical requirements:

- A daily stock level of all products for the last month.
- A weekly report of all products with minimum stock levels.
- Analysing stock levels by brand or product type or supplier
- Daily and weekly sent and received stock orders for the last four weeks.
- Analysing received stock orders by the supplier and by month.

Description of Data Sources

Sample data from the software application “Vend” is provided to understand the overview of suppliers, products, and stock movements. The data from the source includes supplier details, product details and details about stock movements from suppliers to warehouse locations.

1) Analysis & Design of Dimensional Data Model

1.1. To represent the three main activities of the inventory management business area, there are three different central fact tables are considered which described as follows:

1) SENT PURCHASED ORDER

The fact table will record the ordered quantity, sent date, product and supplier details when products are at their minimum level of remaining stock. This fact table seeks to streamline activities related to sending purchase orders. The lowest grain of this fact table is identified as “one sent purchased order number per row” which will store and retrieve all required information from the connected dimensions.

SENT PURCHASED ORDER
Sent purchased order number
Sent purchased order date
Sent purchased order quantity

2) RECEIVED PURCHASED ORDER

The fact table will record the destination of storage, ordered quantity, received purchased order quantity, received date, product and supplier details when products are at their minimum level of remaining stock. The objective of this fact table is to make the process of submitting purchase orders more efficient. This fact table's lowest granularity level is identified as "one received purchased order number per row," which will read and write all necessary data from the linked dimensions.

RECEIVED PURCHASED ORDER
Received purchased order number
Received purchased order sent date
Received purchased order received date
Received purchased order received quantity
Received purchased order ordered quantity

3) STOCK

This fact table will control and maintain stocks which include adding products or adjusting stock levels, as well as generating reports of stocks on demand. The focus of this fact table is to capture the movement of every unit of inventory right from when they are supplied to when they leave the different stores. The grain for this fact table, therefore, is “one product stock level per row” as this will afford stock analysis on a more granular level.

STOCK
Add new product
Add stock to existing product
Current stock level

1.2. To meet the reporting and analysis requirements mentioned by the company, the following dimensions are identified to support three fact tables.

1) Supplier: this dimension will contain and store all information about the supplier and it will connect with all three central fact tables using the primary key “supplier_key”

2) Date: this dimension will contain and store all date-related information and it will connect with all three central fact tables using the primary key “date_key”

3) Order: this dimension will contain and store order details such as purchase order type, order date and destination outlet. Sent purchased orders and received purchased orders will be filtered by purchase order type. It will join two fact tables “sent purchased order” and received “purchased order” by using the primary key “order_key”

4) Product: this dimension will store and manage all information about product type, SKU number, name, brand, stock level, price etc. It will join with all three central fact tables using the primary key “product_key”

1.3. Based on given business requirements, and determining dimensions of three central fact tables, different attributes are considered to complete the dimensional data model.

Supplier
Supplier name
Supplier phone number
Supplier description
Supplier Email
Supplier address
Supplier City
Supplier postcode
Supplier country
🗝 supplier_key

Date
Full date
Day name of the week
Week of month
Week of year
Month name
Month of year
Quarter
year
🗝 date_key

Order
Purchase order Type
Purchase order number
Ordered quantity
Ordered unit price
Ordered date
Destination outlet
Product name
🗝 order_key

Product
Product Type
SKU Number
Product name
Product Description
Product condition
Product brand
Supplier name
Product tag
Current stock level
Product date added
Retail Price
🗝 product_key

1.4. Graphical representation of structure: Simple star schema is used to design the Dimensional Data Model.

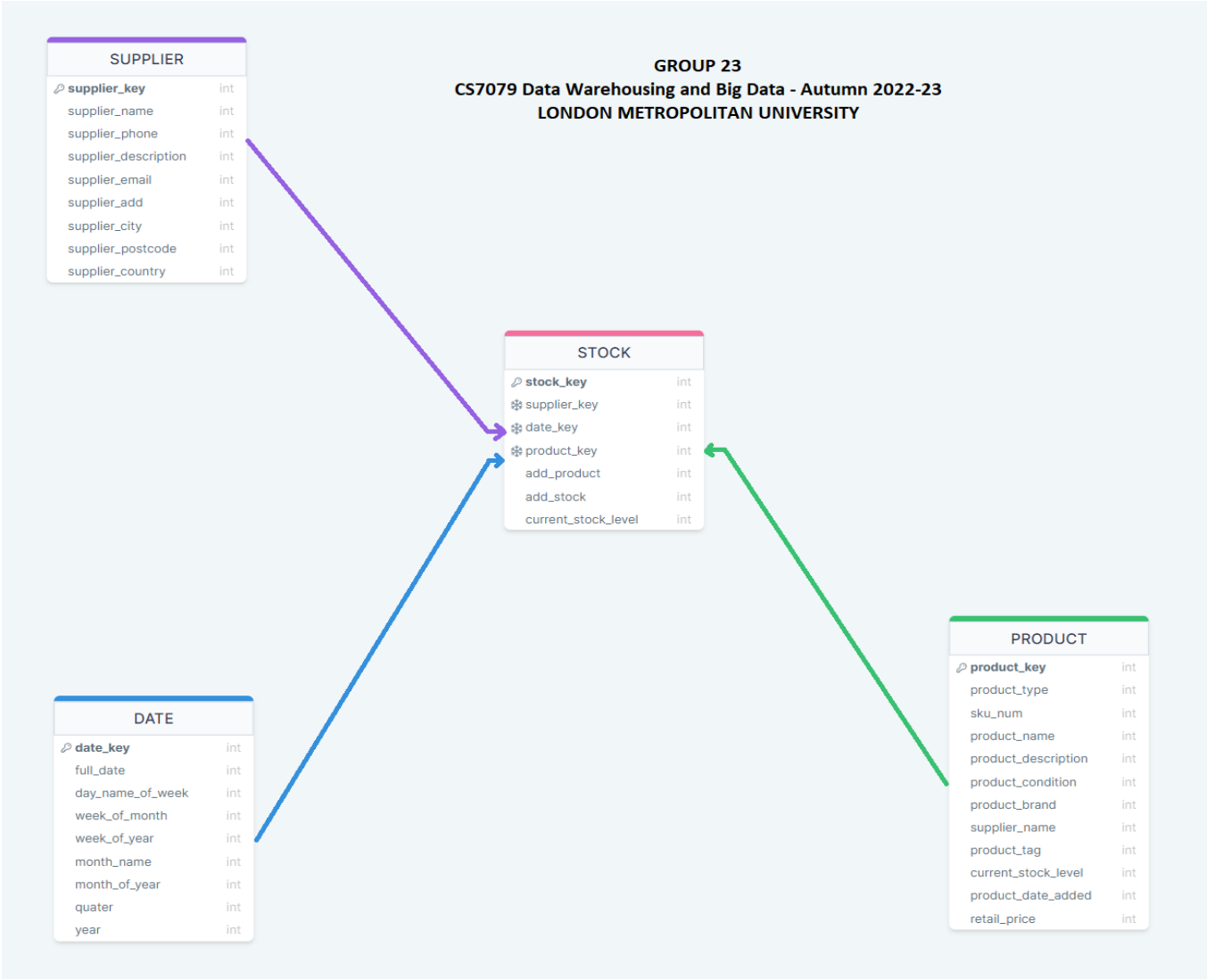


Figure 1: Supplier, Date and Product dimensions are directly connected to the "Stock" fact table.

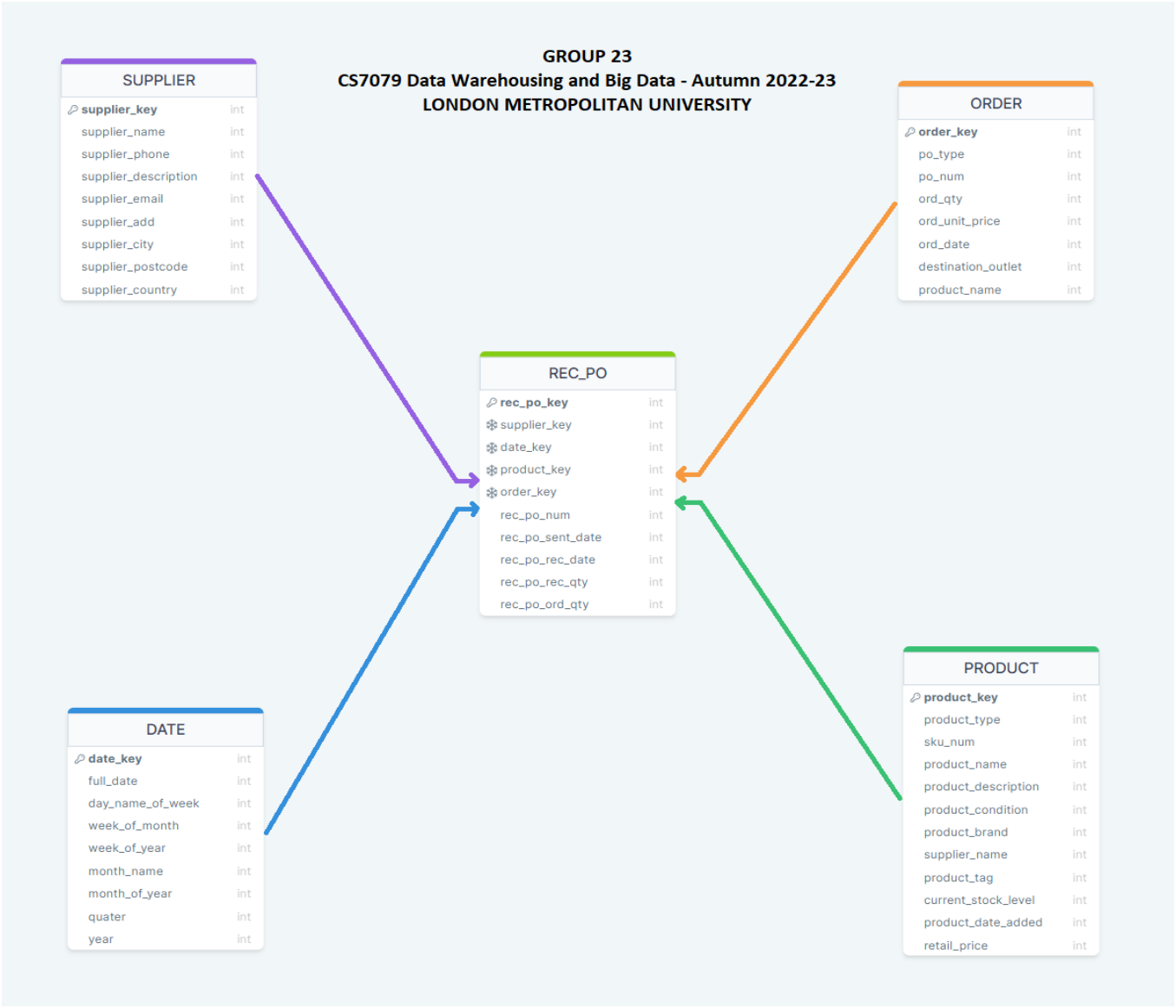


Figure 2: Supplier, Date, Product and order dimensions are directly connected to the "Received Purchased Order" fact table.

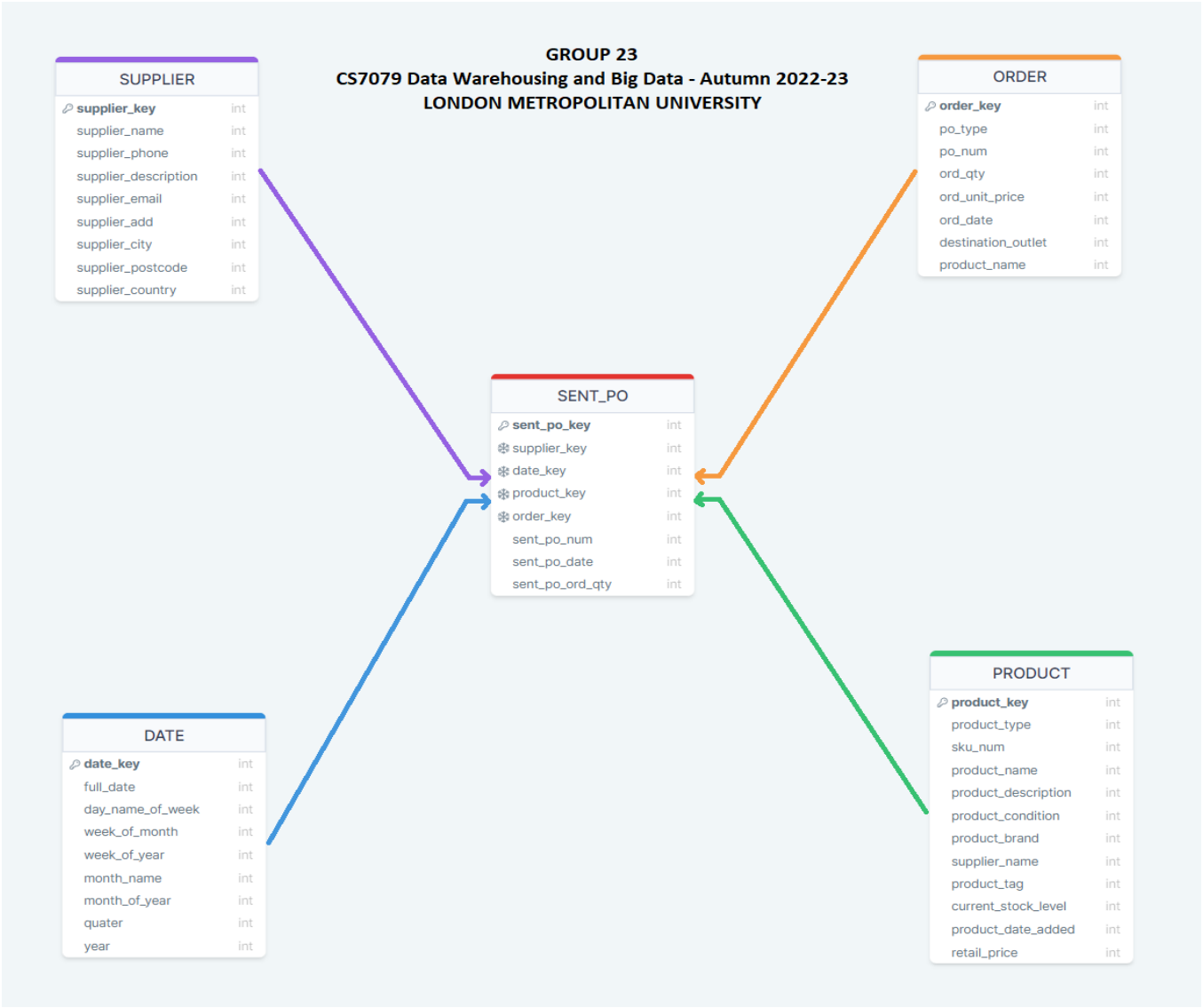


Figure 3: Supplier, Date, Product and order dimensions are directly connected to the "Sent Purchased Order" fact table

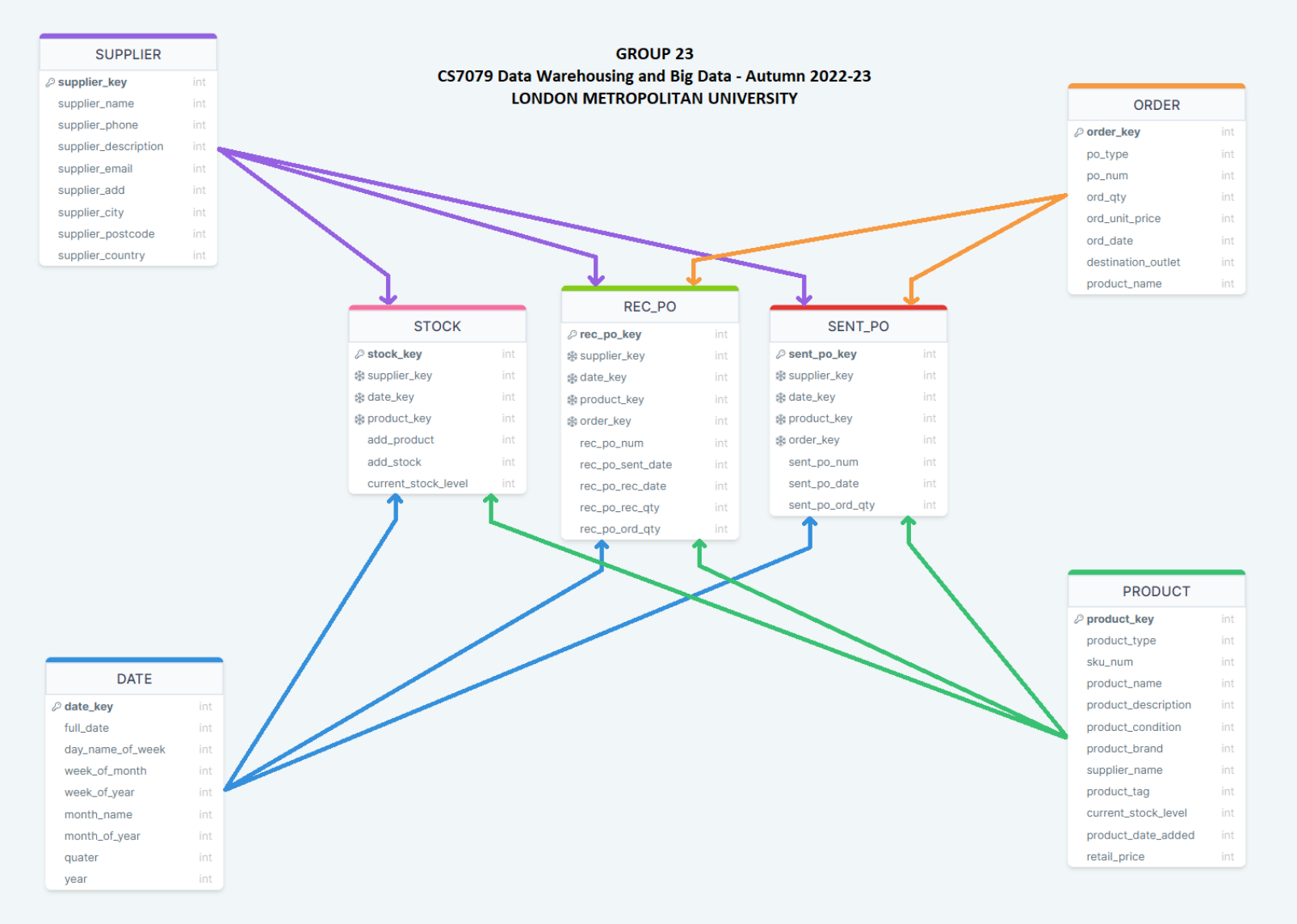


Figure 4: Supplier, Date, Product and order dimensions are connected to central fact tables.

CS7079 Data Warehousing and Big Data - Autumn 2022-23. PART B.

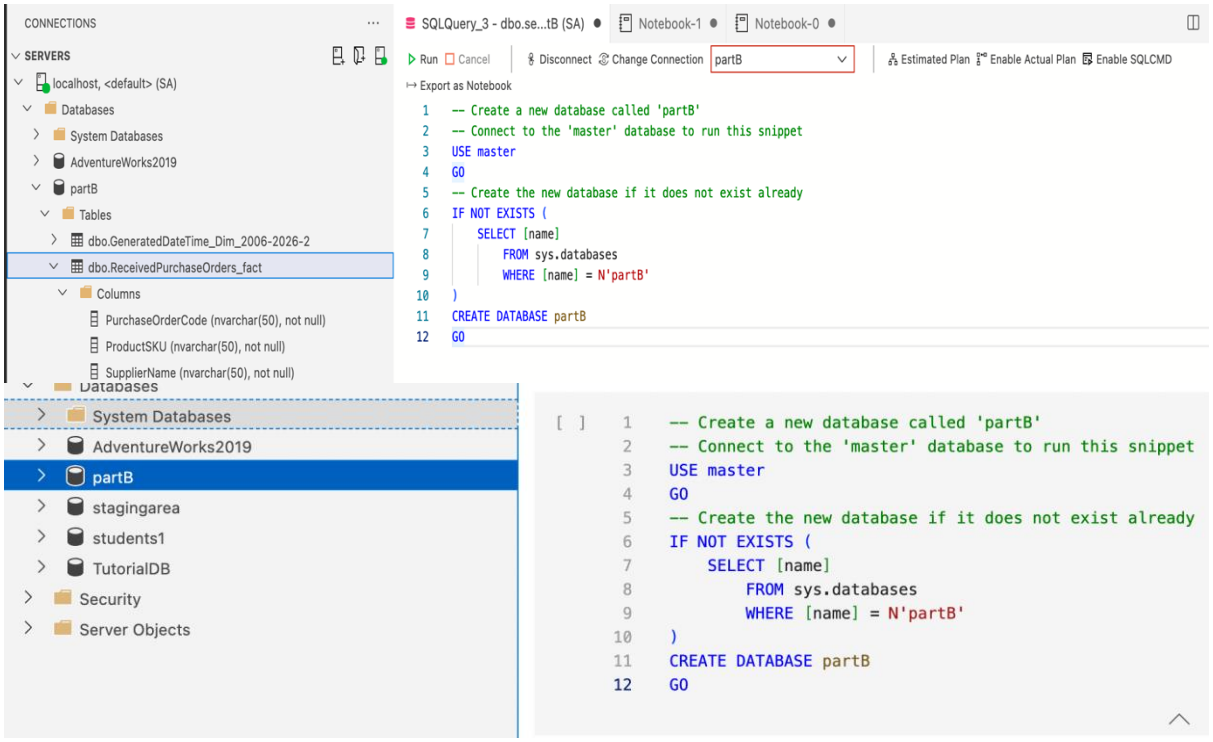
21025221

JANUARY 12, 2023

SAMRAT RAI
21025221

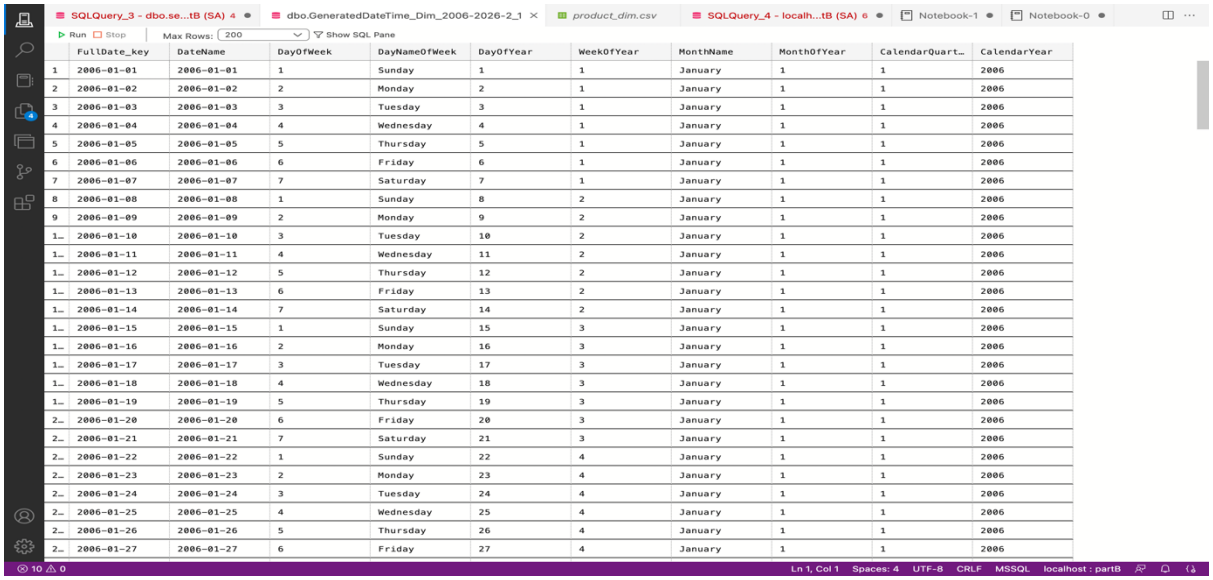
2. Create a Relational Database to store dimensions and fact tables using Microsoft SQL Server Management Studio.

2.1 create a database:



Created a database called ‘partB’ in azure data studio as SSMS is not available in MAC laptop. As you can see in the right-hand side of the picture above is where I used MsSql to create database and left-hand side where the database ‘partB’.

2.2 create dimensional tables:



	FullDate_key	DateName	DayOfWeek	DayNameOfWeek	DayOfYear	WeekOfYear	MonthName	MonthOfYear	CalendarQuart…	CalendarYear
1	2006-01-01	2006-01-01	1	Sunday	1	1	January	1	1	2006
2	2006-01-02	2006-01-02	2	Monday	2	1	January	1	1	2006
3	2006-01-03	2006-01-03	3	Tuesday	3	1	January	1	1	2006
4	2006-01-04	2006-01-04	4	Wednesday	4	1	January	1	1	2006
5	2006-01-05	2006-01-05	5	Thursday	5	1	January	1	1	2006
6	2006-01-06	2006-01-06	6	Friday	6	1	January	1	1	2006
7	2006-01-07	2006-01-07	7	Saturday	7	1	January	1	1	2006
8	2006-01-08	2006-01-08	1	Sunday	8	2	January	1	1	2006
9	2006-01-09	2006-01-09	2	Monday	9	2	January	1	1	2006
1…	2006-01-10	2006-01-10	3	Tuesday	10	2	January	1	1	2006
1…	2006-01-11	2006-01-11	4	Wednesday	11	2	January	1	1	2006
1…	2006-01-12	2006-01-12	5	Thursday	12	2	January	1	1	2006
1…	2006-01-13	2006-01-13	6	Friday	13	2	January	1	1	2006
1…	2006-01-14	2006-01-14	7	Saturday	14	2	January	1	1	2006
1…	2006-01-15	2006-01-15	1	Sunday	15	3	January	1	1	2006
1…	2006-01-16	2006-01-16	2	Monday	16	3	January	1	1	2006
1…	2006-01-17	2006-01-17	3	Tuesday	17	3	January	1	1	2006
1…	2006-01-18	2006-01-18	4	Wednesday	18	3	January	1	1	2006
1…	2006-01-19	2006-01-19	5	Thursday	19	3	January	1	1	2006
2…	2006-01-20	2006-01-20	6	Friday	20	3	January	1	1	2006
2…	2006-01-21	2006-01-21	7	Saturday	21	3	January	1	1	2006
2…	2006-01-22	2006-01-22	1	Sunday	22	4	January	1	1	2006
2…	2006-01-23	2006-01-23	2	Monday	23	4	January	1	1	2006
2…	2006-01-24	2006-01-24	3	Tuesday	24	4	January	1	1	2006
2…	2006-01-25	2006-01-25	4	Wednesday	25	4	January	1	1	2006
2…	2006-01-26	2006-01-26	5	Thursday	26	4	January	1	1	2006
2…	2006-01-27	2006-01-27	6	Friday	27	4	January	1	1	2006

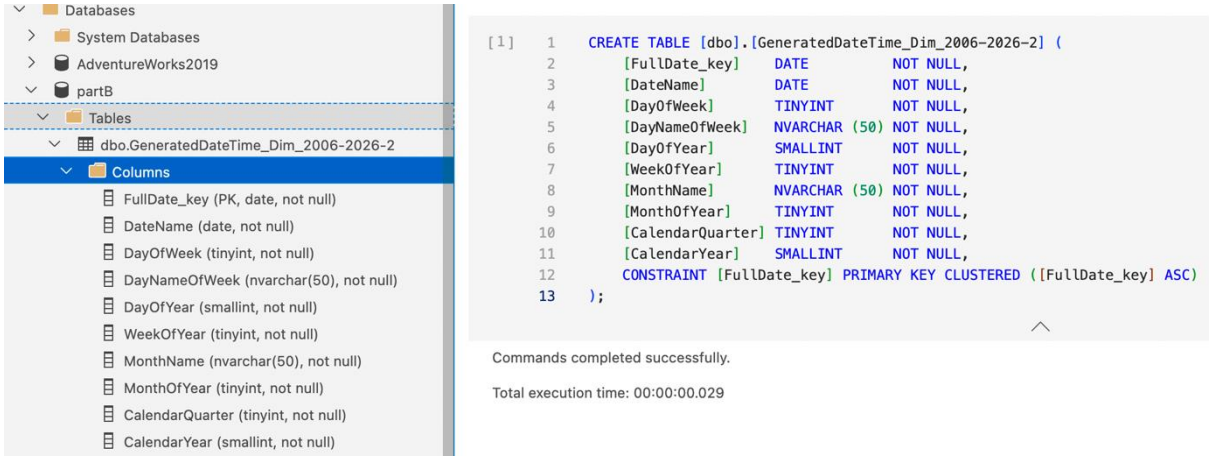


Fig 1. date dimension. here I created a date dimension table through azure data studio, where the first figure is the dimensional table and the second figure is the code and where the dimension is saved which named dbo.generateddatetime_dim_2006-2026-2 under the tables folder.

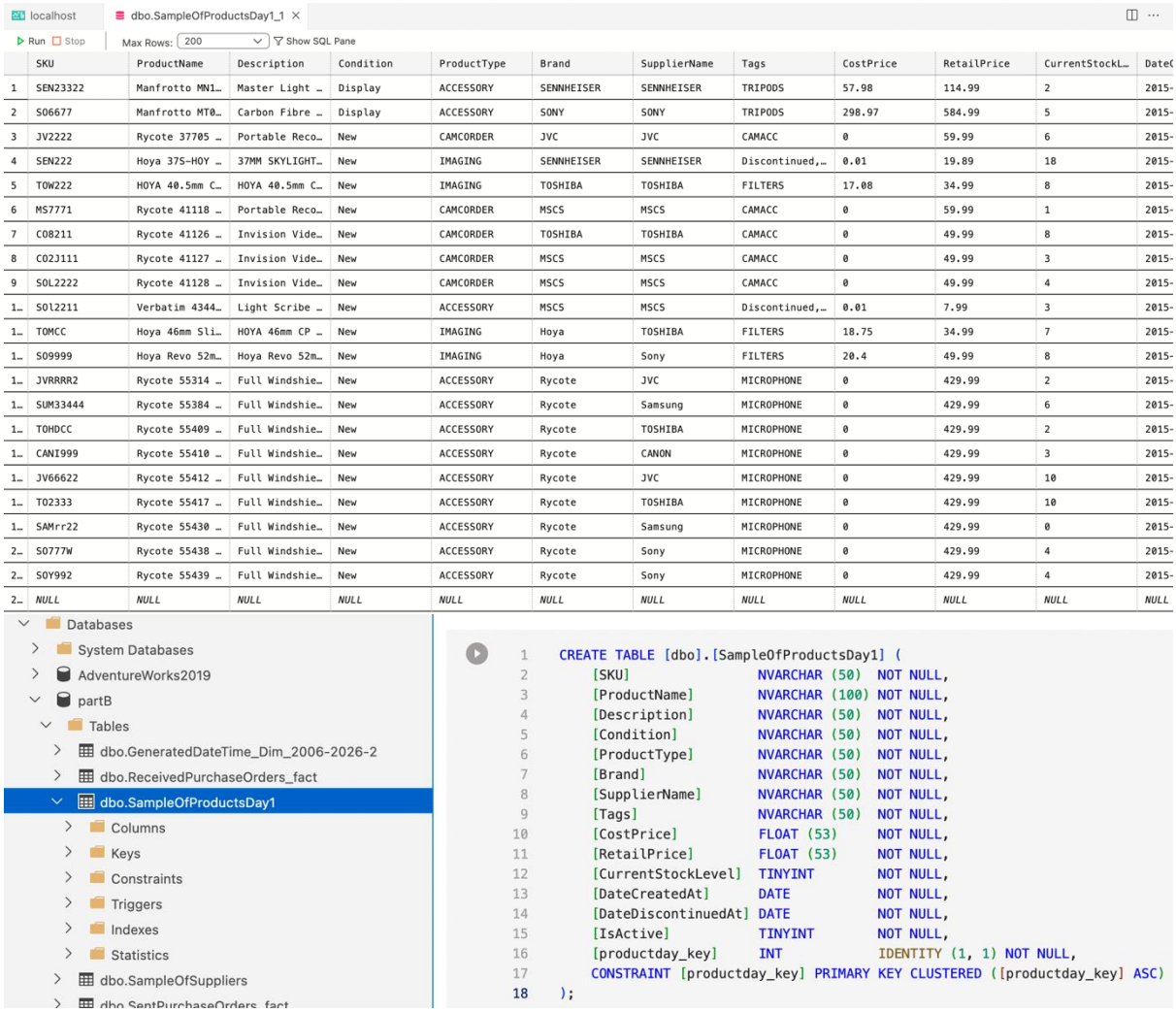


Fig 2. Product dim. here I created a product dimension table through azure data studio, where the first figure is the dimensional table, and the second figure is the code and where the dimension is saved which named dboSampleOfProductDay1, under the tables folder.

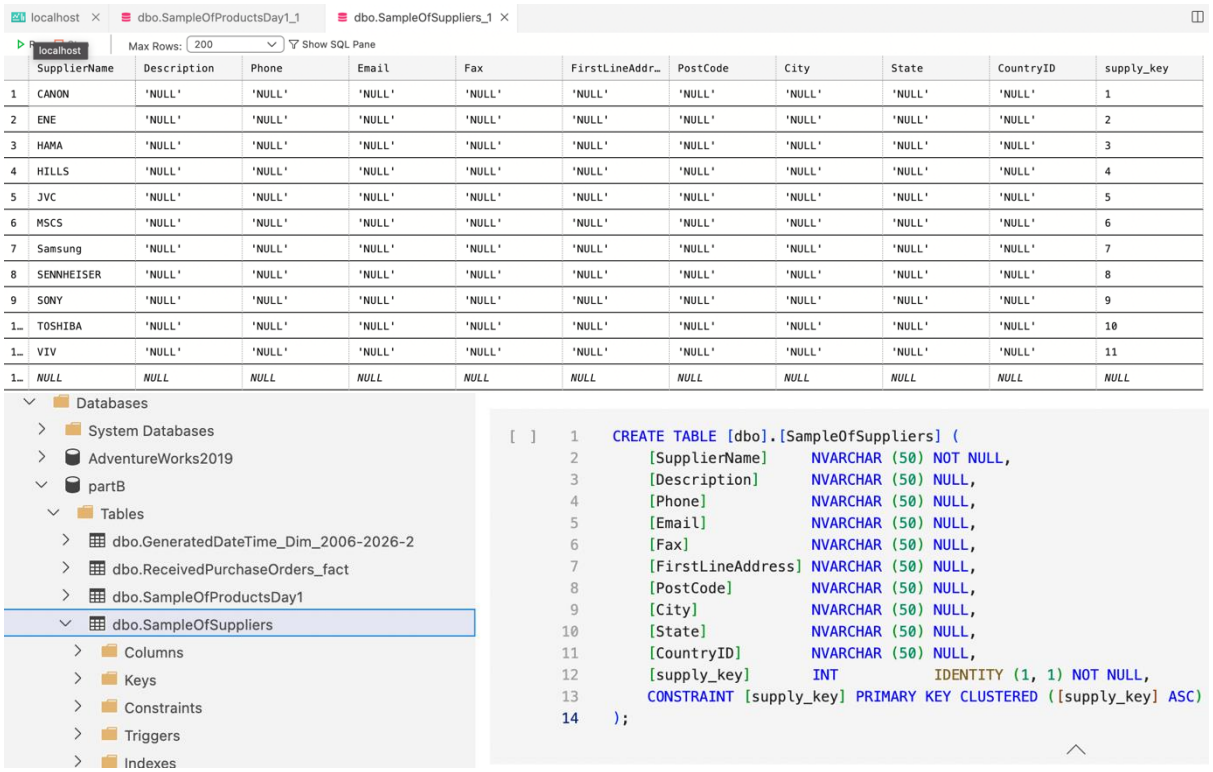


Fig 3. Supplier dim. here I created a supplier dimension table through azure data studio, where the first figure is the dimensional table, and the second figure is the code and where the dimension is saved which named dboSampleOfSuppliers, under the tables folder.

2.3 Create fact tables.

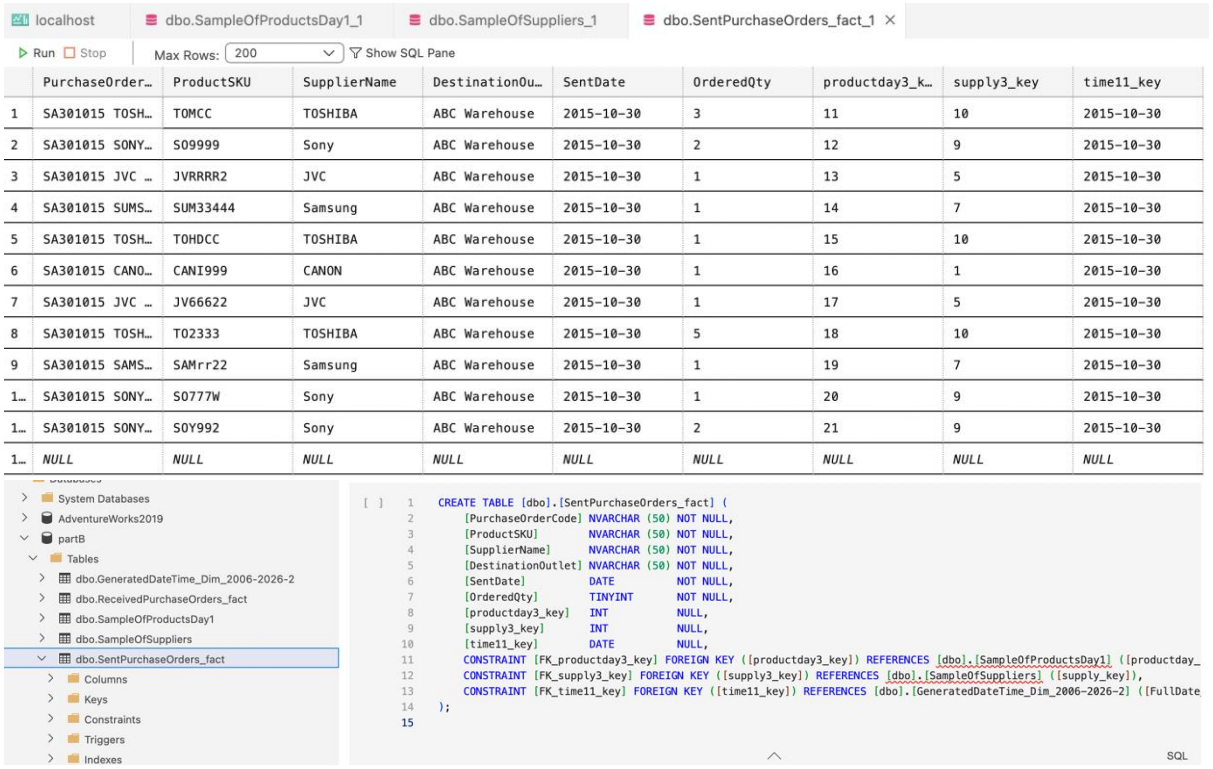
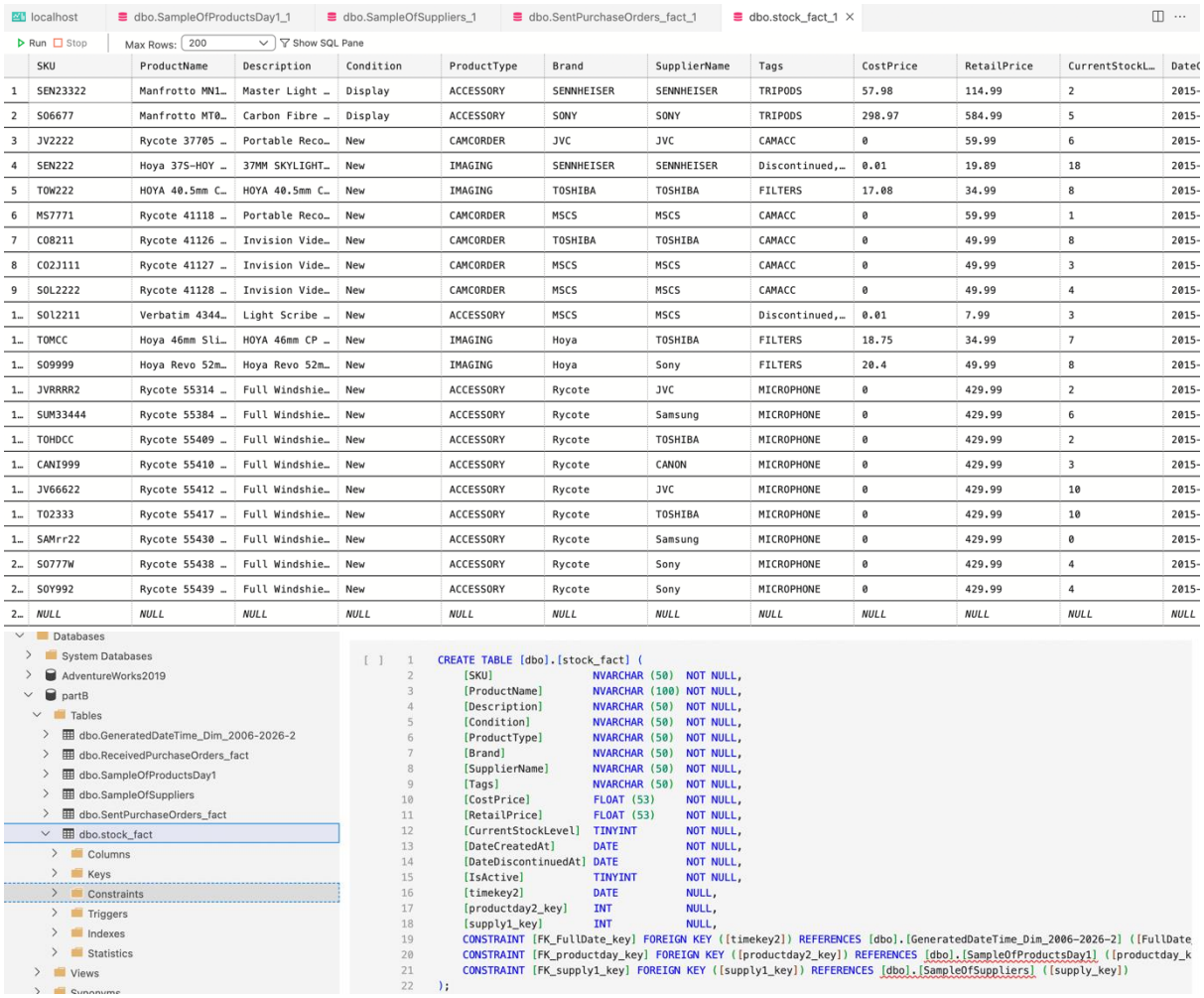


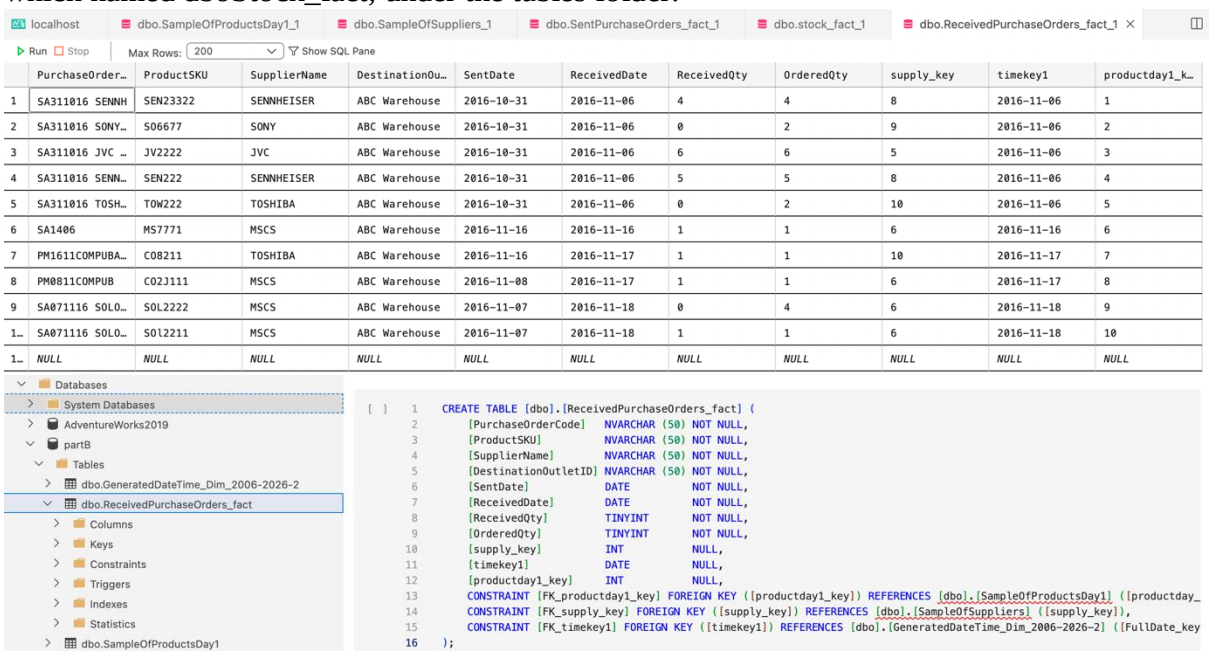
Fig1.3 sent dim. here I created a SentPurchaseOrders fact table through azure data studio, where the first figure is the fact table, and the second figure is the code and where the fact table is saved which named dboSentPurchaseOrders_fact, under the tables folder.



	SKU	ProductName	Description	Condition	ProductType	Brand	SupplierName	Tags	CostPrice	RetailPrice	CurrentStockL…	Date(
1	SEN23322	Manfrotto MN1…	Master Light …	Display	ACCESSORY	SENNHEISER	SENNHEISER	TRIPODS	57.98	114.99	2	2015-
2	SO6677	Manfrotto MT0…	Carbon Fibre …	Display	ACCESSORY	SONY	SONY	TRIPODS	298.97	584.99	5	2015-
3	JV2222	Rycote 37705 …	Portable Reco…	New	CAMCORDER	JVC	JVC	CAMACC	0	59.99	6	2015-
4	SEN222	Hoya 37S-HOY …	37MM SKYLIGHT…	New	IMAGING	SENNHEISER	SENNHEISER	Discontinued,…	0.01	19.89	18	2015-
5	TOW222	HOYA 40.5mm C…	HOYA 40.5mm C…	New	IMAGING	TOSHIBA	TOSHIBA	FILTERS	17.08	34.99	8	2015-
6	MS7771	Rycote 41118 …	Portable Reco…	New	CAMCORDER	MSCS	MSCS	CAMACC	0	59.99	1	2015-
7	CO8211	Rycote 41126 …	Invision Vide…	New	CAMCORDER	TOSHIBA	TOSHIBA	CAMACC	0	49.99	8	2015-
8	CO2J111	Rycote 41127 …	Invision Vide…	New	CAMCORDER	MSCS	MSCS	CAMACC	0	49.99	3	2015-
9	SOL2222	Rycote 41128 …	Invision Vide…	New	CAMCORDER	MSCS	MSCS	CAMACC	0	49.99	4	2015-
1…	SOl2211	Verbatim 4344…	Light Scribe …	New	ACCESSORY	MSCS	MSCS	Discontinued,…	0.01	7.99	3	2015-
1…	TOMCC	Hoya 46mm Sli…	HOYA 46mm CP …	New	IMAGING	Hoya	TOSHIBA	FILTERS	18.75	34.99	7	2015-
1…	SO9999	Hoya Revo 52m…	Hoya Revo 52m…	New	IMAGING	Hoya	Sony	FILTERS	20.4	49.99	8	2015-
1…	JVRRRR2	Rycote 55314 …	Full Windshie…	New	ACCESSORY	Rycote	JVC	MICROPHONE	0	429.99	2	2015-
1…	SUM33444	Rycote 55384 …	Full Windshie…	New	ACCESSORY	Rycote	Samsung	MICROPHONE	0	429.99	6	2015-
1…	TOHDCC	Rycote 55409 …	Full Windshie…	New	ACCESSORY	Rycote	TOSHIBA	MICROPHONE	0	429.99	2	2015-
1…	CANI999	Rycote 55410 …	Full Windshie…	New	ACCESSORY	Rycote	CANON	MICROPHONE	0	429.99	3	2015-
1…	JV66622	Rycote 55412 …	Full Windshie…	New	ACCESSORY	Rycote	JVC	MICROPHONE	0	429.99	10	2015-
1…	TO2333	Rycote 55417 …	Full Windshie…	New	ACCESSORY	Rycote	TOSHIBA	MICROPHONE	0	429.99	10	2015-
1…	SAMrr22	Rycote 55430 …	Full Windshie…	New	ACCESSORY	Rycote	Samsung	MICROPHONE	0	429.99	0	2015-
2…	SO777W	Rycote 55438 …	Full Windshie…	New	ACCESSORY	Rycote	Sony	MICROPHONE	0	429.99	4	2015-
2…	SOY992	Rycote 55439 …	Full Windshie…	New	ACCESSORY	Rycote	Sony	MICROPHONE	0	429.99	4	2015-
2…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE [dbo].[stock_fact] (
    [SKU]              NVARCHAR (50)  NOT NULL,
    [ProductName]      NVARCHAR (100) NOT NULL,
    [Description]      NVARCHAR (50)  NOT NULL,
    [Condition]        NVARCHAR (50)  NOT NULL,
    [ProductType]      NVARCHAR (50)  NOT NULL,
    [Brand]            NVARCHAR (50)  NOT NULL,
    [SupplierName]     NVARCHAR (50)  NOT NULL,
    [Tags]             NVARCHAR (50)  NOT NULL,
    [CostPrice]        FLOAT (53)     NOT NULL,
    [RetailPrice]      FLOAT (53)     NOT NULL,
    [CurrentStockLevel] TINYINT       NOT NULL,
    [DateCreatedAt]    DATE           NOT NULL,
    [DateDiscontinuedAt] DATE         NOT NULL,
    [IsActive]         TINYINT        NOT NULL,
    [timekey2]         DATE           NULL,
    [productday2_key]  INT            NULL,
    [supply1_key]      INT            NULL,
    CONSTRAINT [FK_FullDate_key] FOREIGN KEY ([timekey2]) REFERENCES [dbo].[GeneratedDateTime_Dim_2006-2026-2] ([FullDate
    CONSTRAINT [FK_productday_key] FOREIGN KEY ([productday2_key]) REFERENCES [dbo].[SampleOfProductsDay1] ([productday_k
    CONSTRAINT [FK_supply1_key] FOREIGN KEY ([supply1_key]) REFERENCES [dbo].[SampleOfSuppliers] ([supply_key])
);
```

Fig 1.4 stock fact. here I created a stock fact table through azure data studio, where the first figure is the fact table, and the second figure is the code and where the fact table is saved which named dboStock_fact, under the tables folder.



	PurchaseOrder…	ProductSKU	SupplierName	DestinationOu…	SentDate	ReceivedDate	ReceivedQty	OrderedQty	supply_key	timekey1	productday1_k…
1	SA311016 SENNH	SEN23322	SENNHEISER	ABC Warehouse	2016-10-31	2016-11-06	4	4	8	2016-11-06	1
2	SA311016 SONY…	SO6677	SONY	ABC Warehouse	2016-10-31	2016-11-06	0	2	9	2016-11-06	2
3	SA311016 JVC …	JV2222	JVC	ABC Warehouse	2016-10-31	2016-11-06	6	6	5	2016-11-06	3
4	SA311016 SENN…	SEN222	SENNHEISER	ABC Warehouse	2016-10-31	2016-11-06	5	5	8	2016-11-06	4
5	SA311016 TOSH…	TOW222	TOSHIBA	ABC Warehouse	2016-10-31	2016-11-06	0	2	10	2016-11-06	5
6	SA1406	MS7771	MSCS	ABC Warehouse	2016-11-16	2016-11-16	1	1	6	2016-11-16	6
7	PM1611COMPUBA…	CO8211	TOSHIBA	ABC Warehouse	2016-11-16	2016-11-17	1	1	10	2016-11-17	7
8	PM0811COMPUB	CO2J111	MSCS	ABC Warehouse	2016-11-08	2016-11-17	1	1	6	2016-11-17	8
9	SA071116 SOLO…	SOL2222	MSCS	ABC Warehouse	2016-11-07	2016-11-18	0	4	6	2016-11-18	9
1…	SA071116 SOLO…	SOl2211	MSCS	ABC Warehouse	2016-11-07	2016-11-18	1	1	6	2016-11-18	10
1…	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE [dbo].[ReceivedPurchaseOrders_fact] (
    [PurchaseOrderCode]   NVARCHAR (50) NOT NULL,
    [ProductSKU]          NVARCHAR (50) NOT NULL,
    [SupplierName]        NVARCHAR (50) NOT NULL,
    [DestinationOutletID] NVARCHAR (50) NOT NULL,
    [SentDate]            DATE          NOT NULL,
    [ReceivedDate]        DATE          NOT NULL,
    [ReceivedQty]         TINYINT       NOT NULL,
    [OrderedQty]          TINYINT       NOT NULL,
    [supply_key]          INT           NULL,
    [timekey1]            DATE          NULL,
    [productday1_key]     INT           NULL,
    CONSTRAINT [FK_productday1_key] FOREIGN KEY ([productday1_key]) REFERENCES [dbo].[SampleOfProductsDay1] ([productday_
    CONSTRAINT [FK_supply_key] FOREIGN KEY ([supply_key]) REFERENCES [dbo].[SampleOfSuppliers] ([supply_key]),
    CONSTRAINT [FK_timekey1] FOREIGN KEY ([timekey1]) REFERENCES [dbo].[GeneratedDateTime_Dim_2006-2026-2] ([FullDate_key
);
```

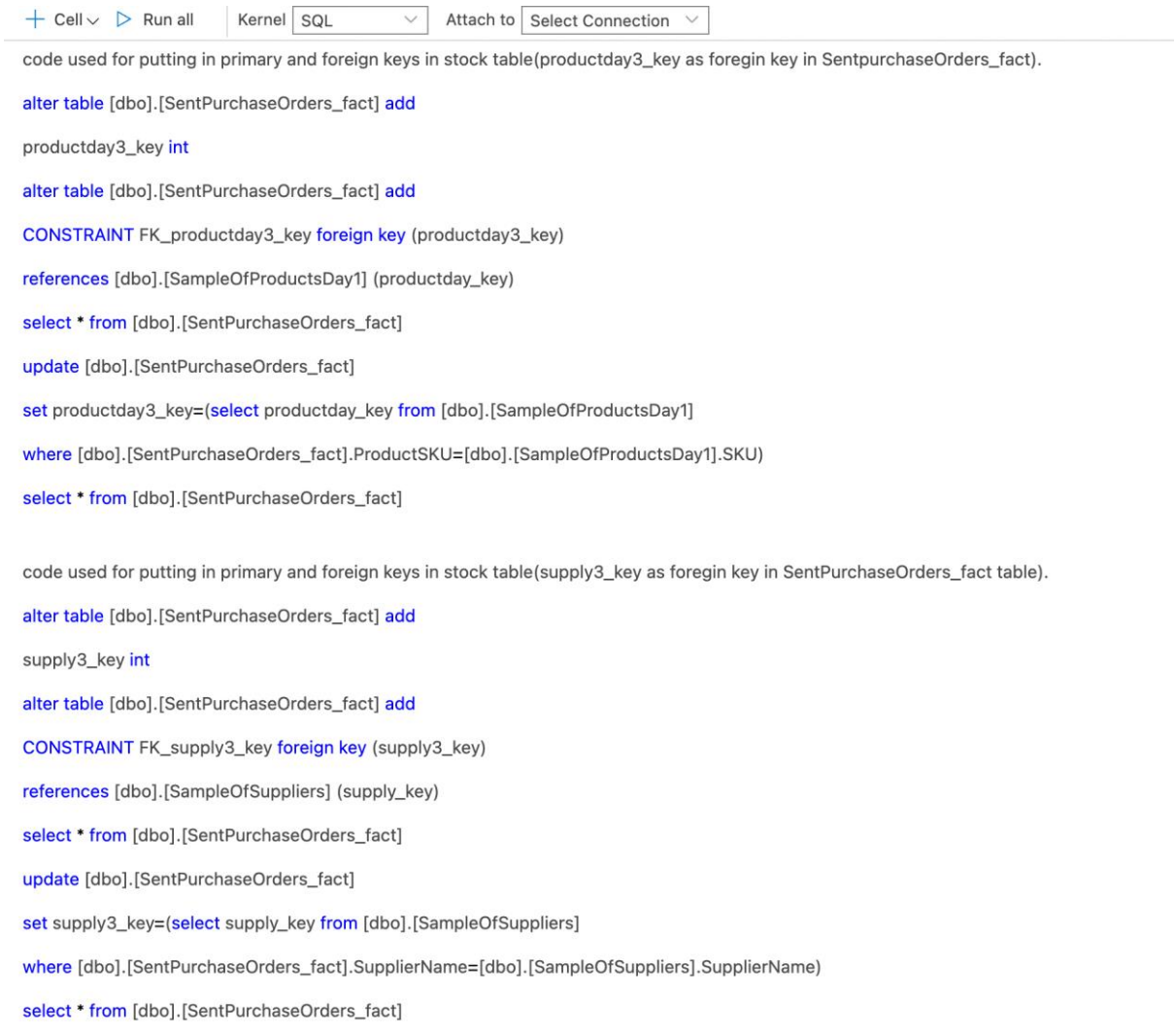
Fig 1.5 receivedpurchaseorder fact. here I created a receivedpurchaseorder fact table through azure data studio, where the first figure is the fact table, and the second figure is the code and

where the fact table is saved which named dboReceivedPurchaseOrders_fact, under the tables folder.

2.4 Add appropriate primary keys and foreign keys constraints.

code used to alter table

```
alter table [dbo].[SampleOfReceivedPurchaseOrders]
add received_orderkey int IDENTITY(1,1) PRIMARY KEY;
```



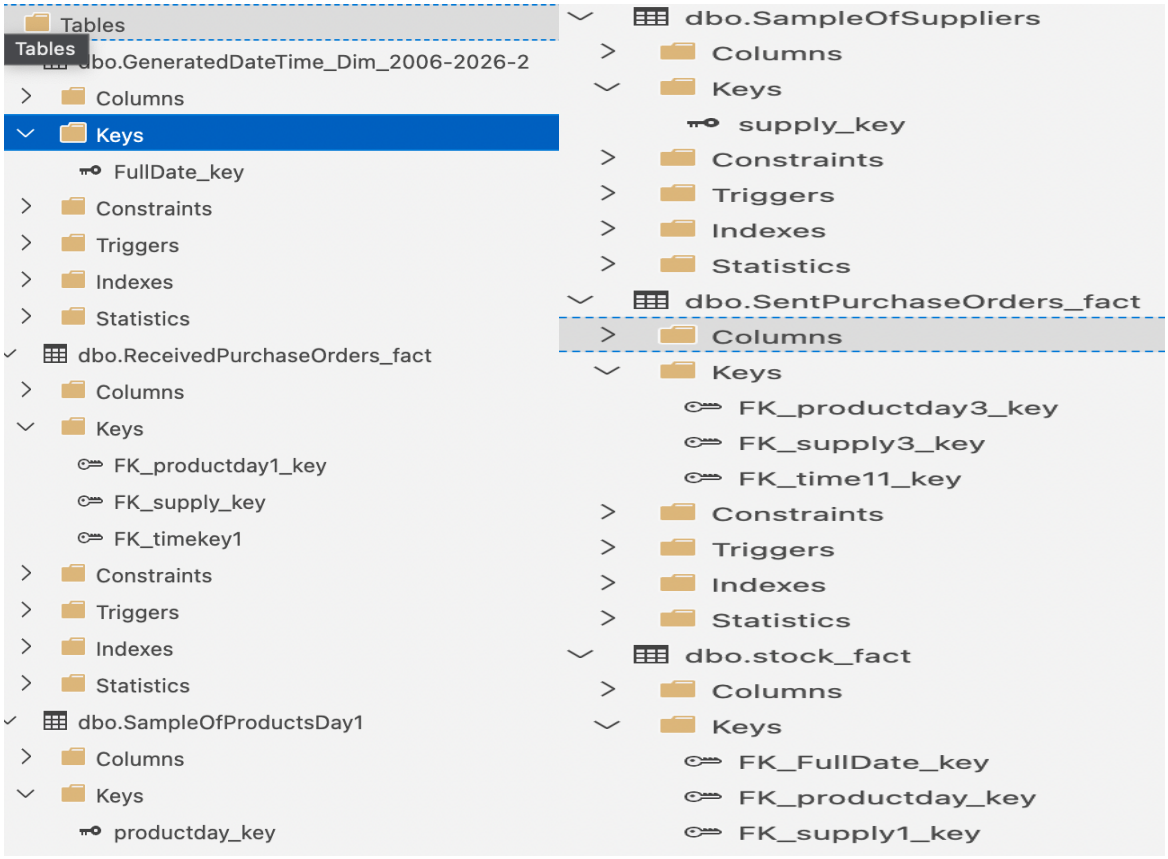
```
code used for putting in primary and foreign keys in stock table(productday2 as foregin key in stock_fact table).
alter table [dbo].[stock_fact] add
productday2_key int
alter table [dbo].[stock_fact] add
CONSTRAINT FK_productday_key foreign key (productday2_key)
references [dbo].[SampleOfProductsDay1] (productday_key)
select * from [dbo].[stock_fact]
update [dbo].[stock_fact]
set productday2_key=(select productday_key from [dbo].[SampleOfProductsDay1]
where [dbo].[stock_fact].SKU=[dbo].[SampleOfProductsDay1].SKU)
select * from [dbo].[stock_fact]

code used to plug in primary and foregin keys to fact table

alter table [dbo].[ReceivedPurchaseOrders_fact] ADD
supply_key INT
alter table [dbo].[ReceviedPurchaseOrders_fact] add
CONSTRAINT FK_supply_key foreign key (supply_key)
references [dbo].[SampleOfSuppliers](supply_key)
select * from [dbo].[ReceivedPurchaseOrders_fact]
update [dbo].[ReceivedPurchaseOrders_fact]
set supply_key=(select supply_key from [dbo].[SampleOfSuppliers]
where dbo.ReceivedPurchaseOrders_fact.SupplierName=dbo.SampleOfSuppliers.SupplierName)
```

```
code used for putting in primary and foreign keys in stock table(time11_key as foregin key in SentPurchaseOrders_fact table).
alter table [dbo].[SentPurchaseOrders_fact] add
time11_key date
alter table [dbo].[SentPurchaseOrders_fact] add
CONSTRAINT FK_time11_key foreign key (time11_key)
references [dbo].[GeneratedDateTime_Dim_2006-2026-2] (FullDate_key)
select * from [dbo].[SentPurchaseOrders_fact]
update [dbo].[SentPurchaseOrders_fact]
set time11_key=(select FullDate_key from [dbo].[GeneratedDateTime_Dim_2006-2026-2]
where [dbo].[SentPurchaseOrders_fact].SentDate=[dbo].[GeneratedDateTime_Dim_2006-2026-2].[DateName])
select * from [dbo].[SentPurchaseOrders_fact]
```

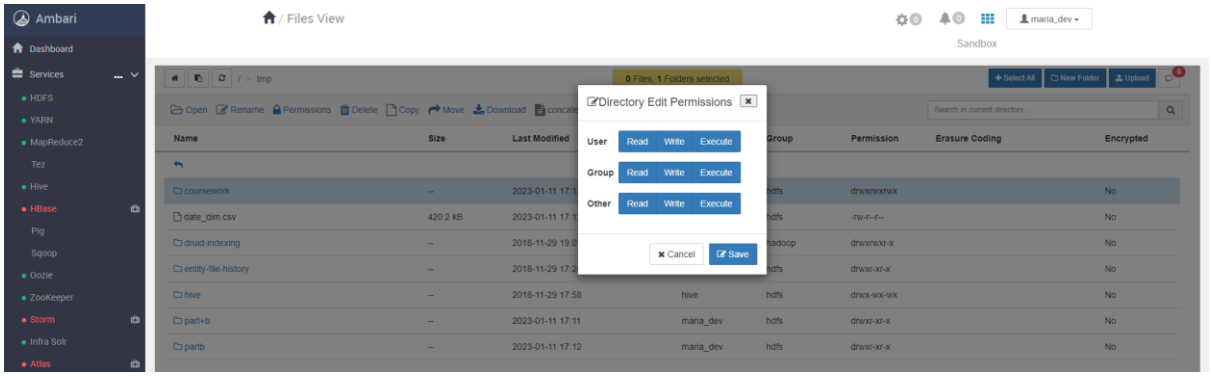
The above figures show the codes I have used to add primary keys and foreign keys constraint.



In the above figure, I have created one primary key for each dimensional table for example sampleofsuppliers has a primary key called supply key and stock fact table has 3 foreign keys called full date key, supply3 key and time11 key. I have done this for all dimensional tables and fact tables as you can see in the above figure.

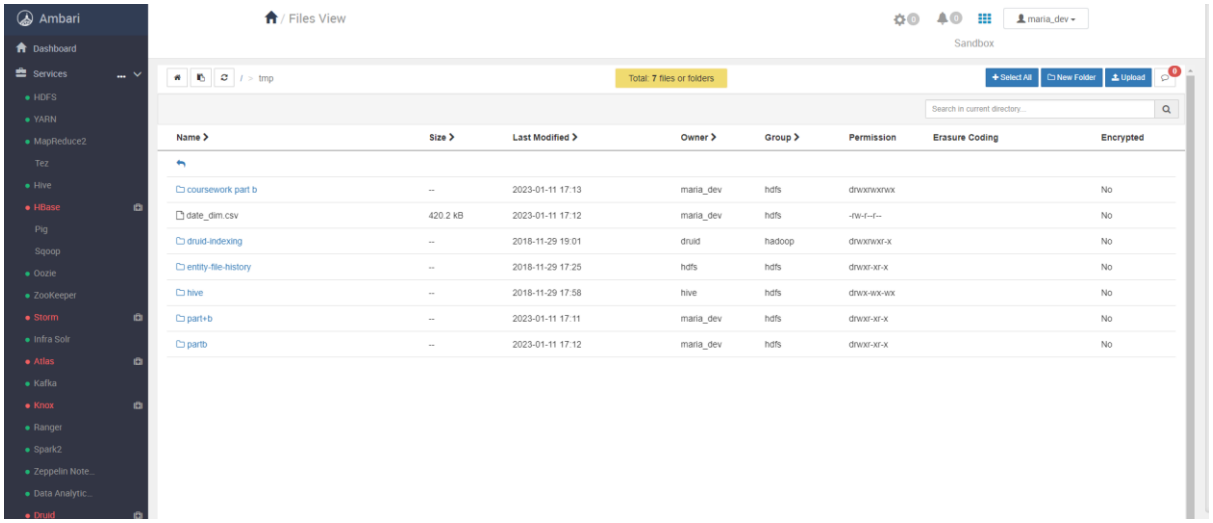
3. Migrate test data from the data warehouse to an Apache Hadoop platform for further analysis of Big Data using Hortonworks Data Platform (HDP)

3.1 Populate the data warehouse database with some test data.



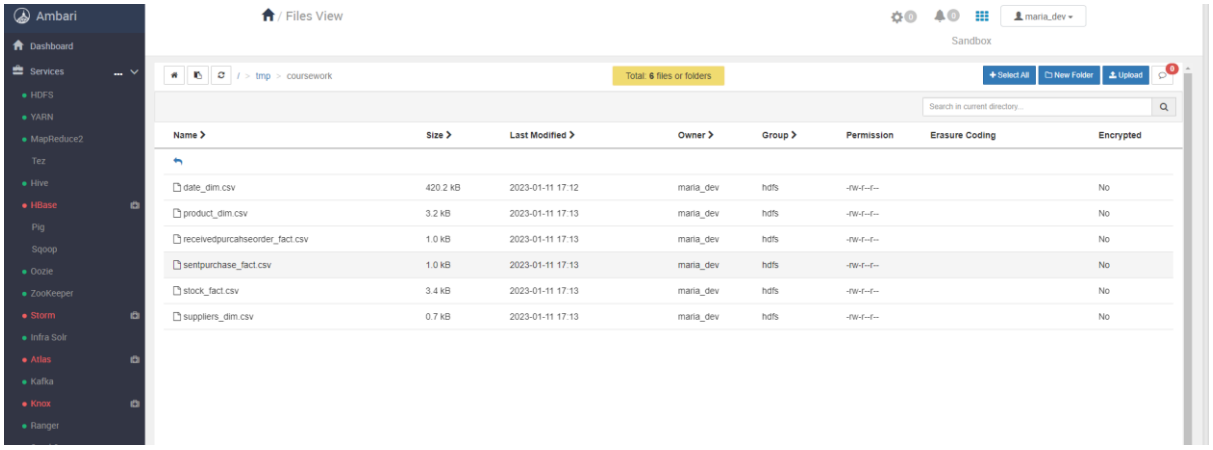
Here I am populating warehouse with some data used in azure data studio and editing Ambari's data warehouse to enable me to read, write and execute code.

3.2 Export the data warehouse database data into an external data file.



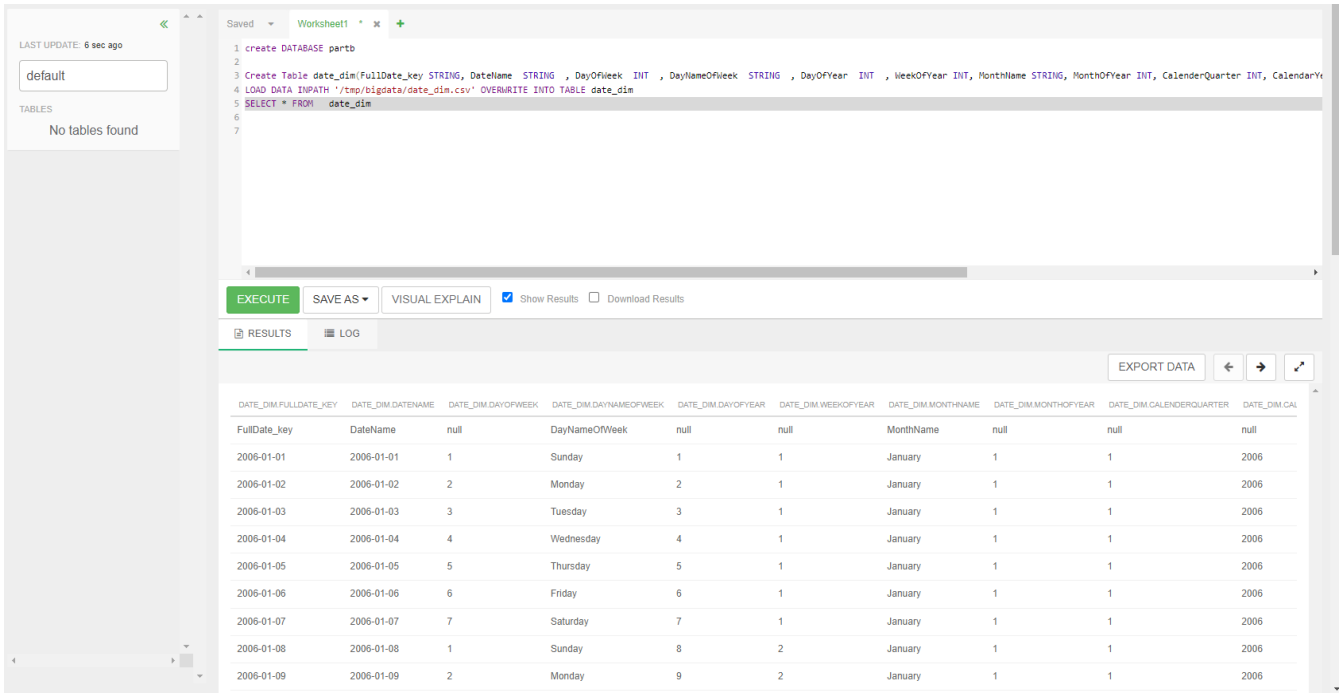
Here I have exported my database data that I used in azure data studio into an external data file called coursework part b in Ambari/Apache HDFS.

3.3 Migrate the data file from the file system to Apache HDFS.

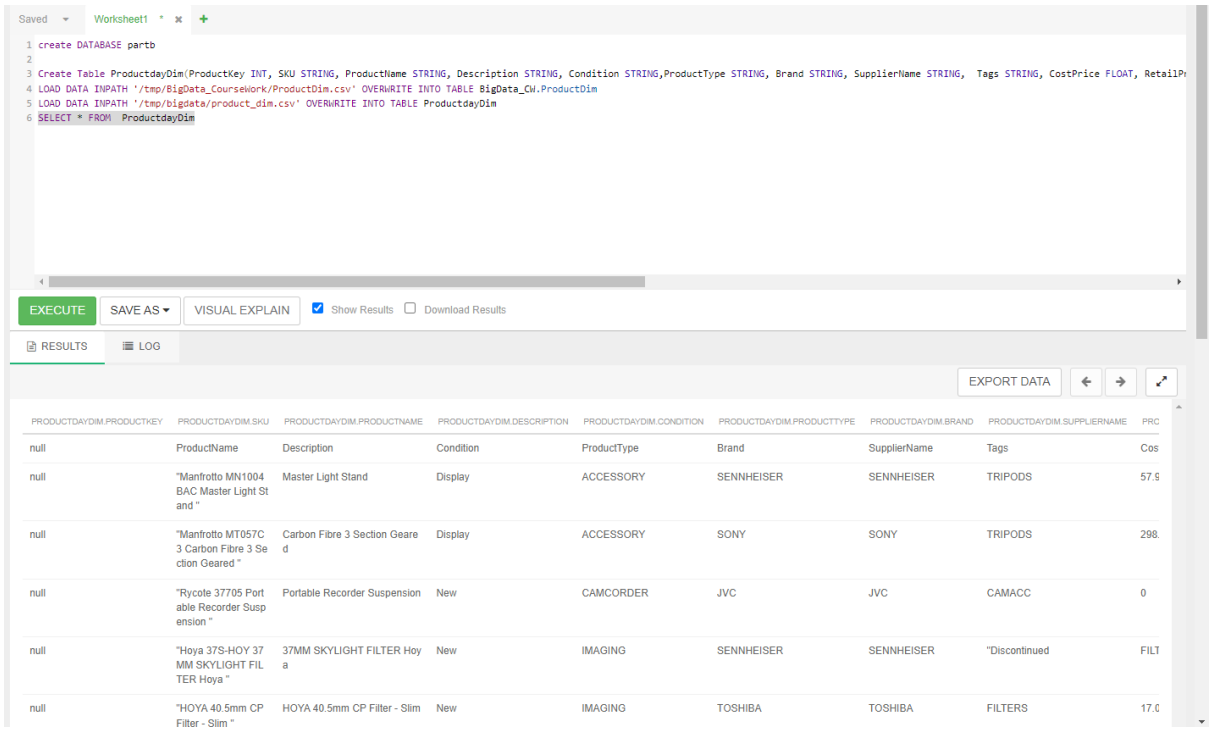


Here I have migrated all the csv files in Ambari/Apache HDFS which were used to create dimensions and fact tables in azure data studio

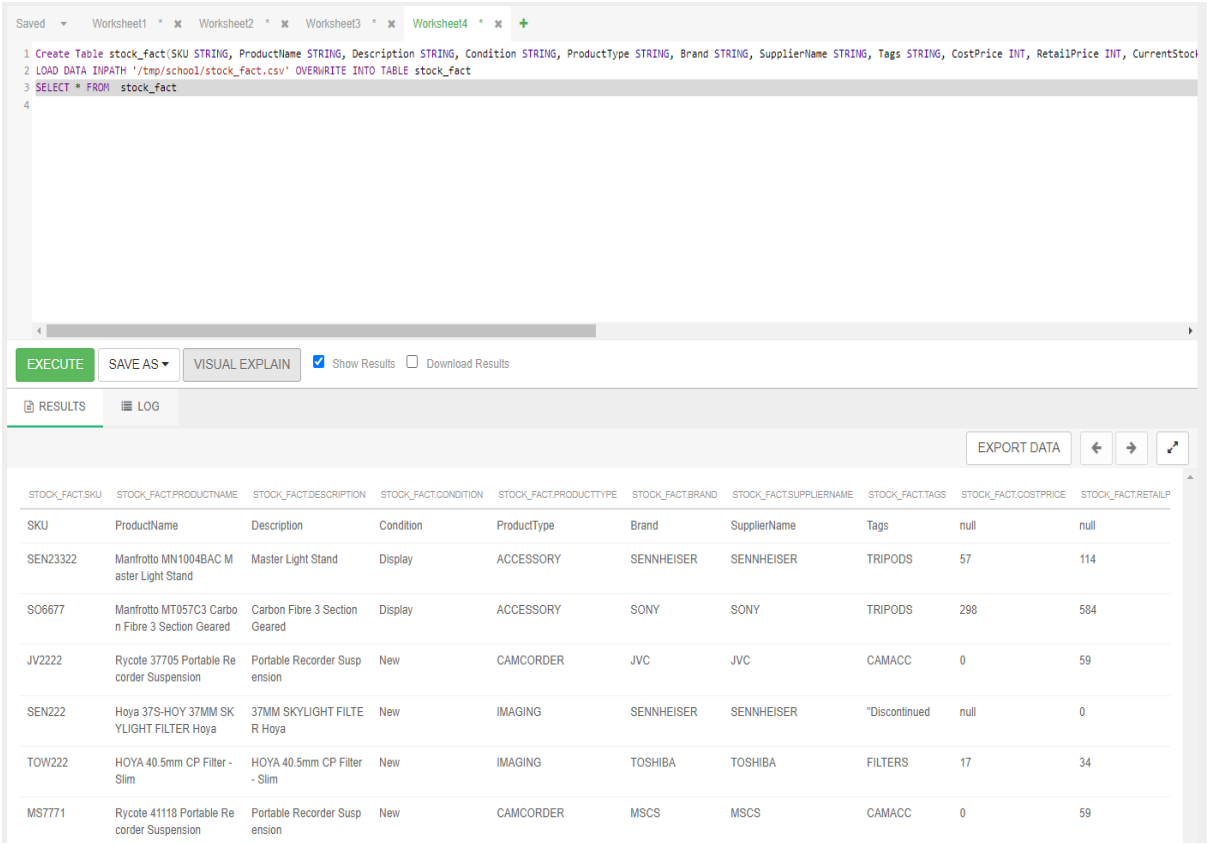
3.4 Create a suitable data structure for loading the data file into HIVE



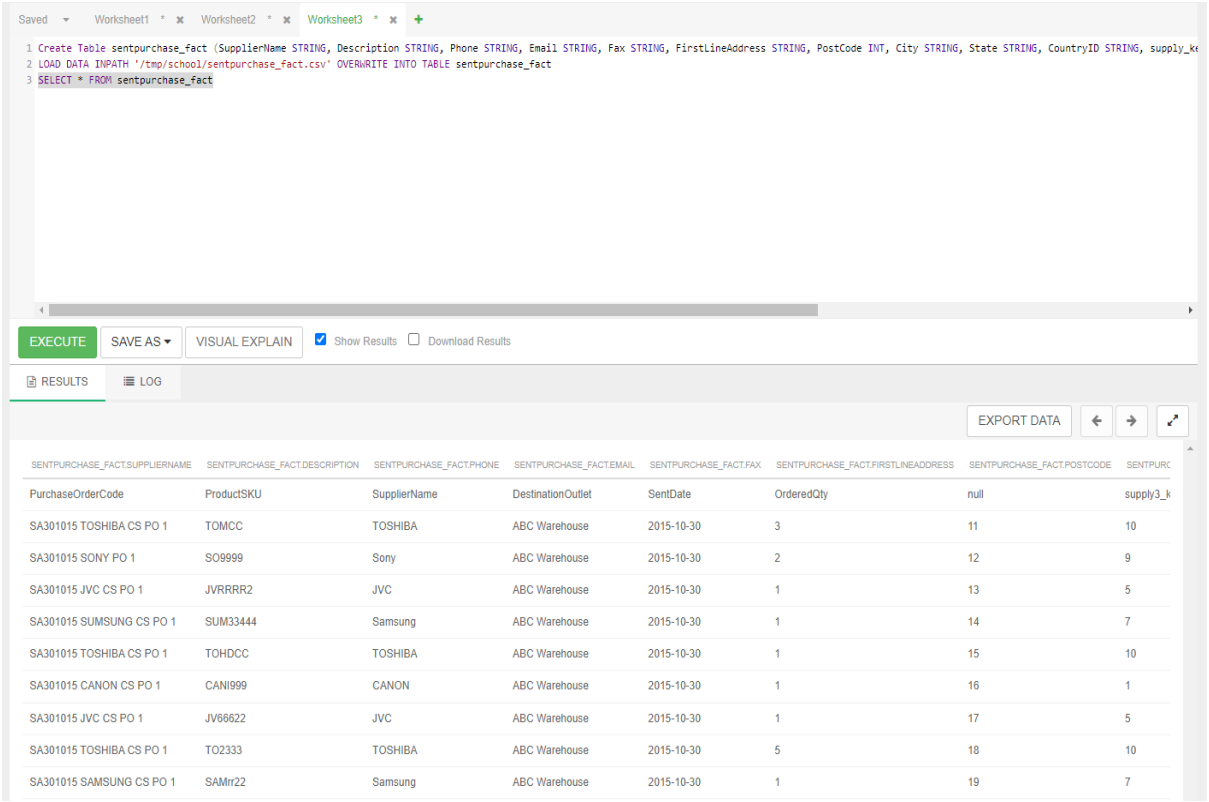
The above figure shows the code used to create date dimensional table. here I used Ambari dashboard and loaded the data file which was used in azure data studio into hive to recreate dimensional table model. MsSql was used to create dimension and table.



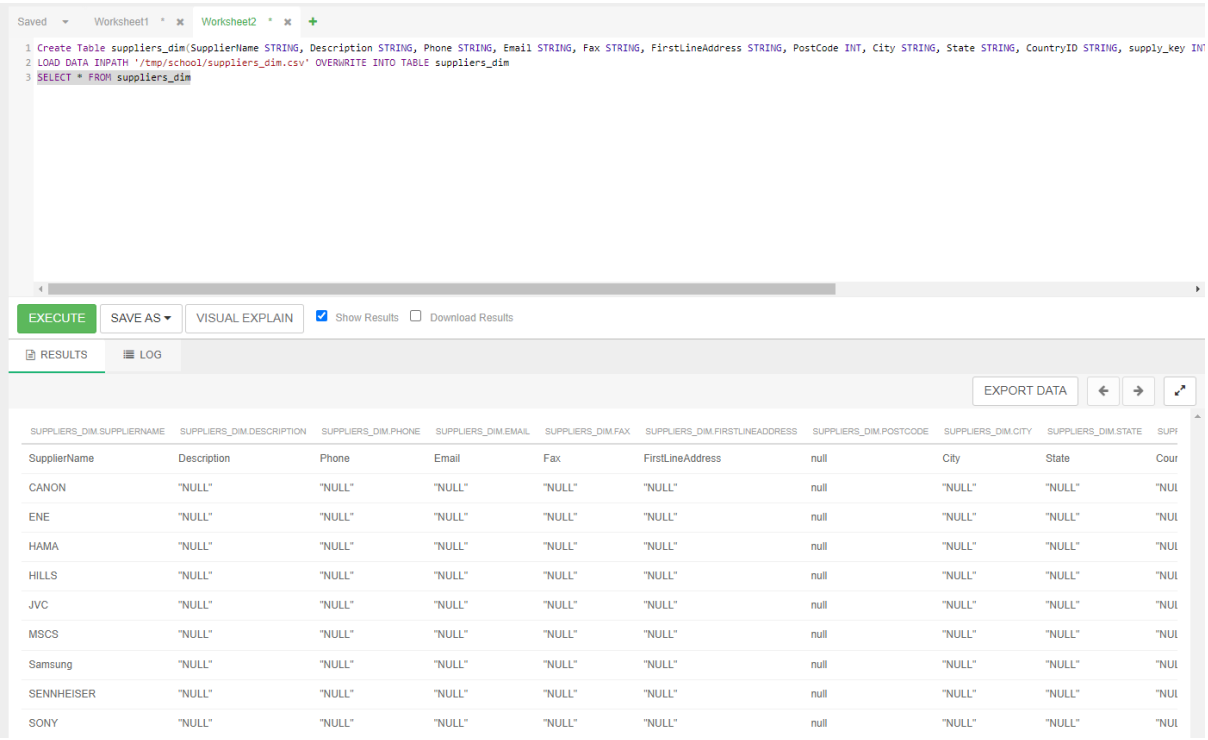
The above figure shows the code used to create product dimensional table. here I used Ambari dashboard and loaded the data file which was used in azure data studio into hive to recreate dimensional table model. MsSql was used to create dimension and table in hive.



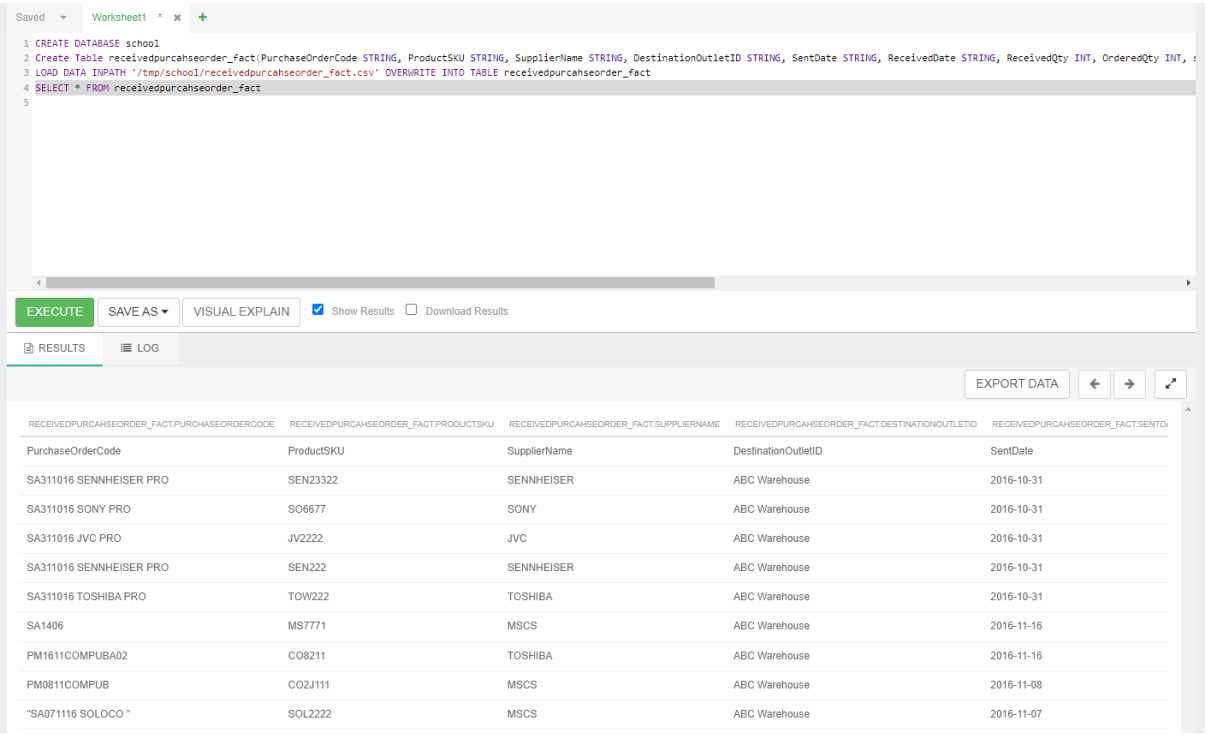
The above figure shows the code used to create stock fact table. here I used Ambari dashboard and loaded the data file which was used in azure data studio into hive to recreate fact table model. MsSql was used to create dimension and table in hive.



The above figure shows the code used to create sent purchase order fact table. here I used Ambari dashboard and loaded the data file which was used in azure data studio into hive to recreate fact table model. MsSql was used to create dimension and table in hive.



The above figure shows the code used to create supplier's dimensional table. here I used Ambari dashboard and loaded the data file which was used in azure data studio into hive to recreate dimensional table model. MsSql was used to create dimension and table in hive.



The above figure shows the code used to create received purchase order fact table. here I used Ambari dashboard and loaded the data file which was used in azure data studio into hive to recreate fact table model. MsSql was used to create dimension and table in hive.

3.5 Demonstrate the use of Apache Pig for manipulating the loaded data.

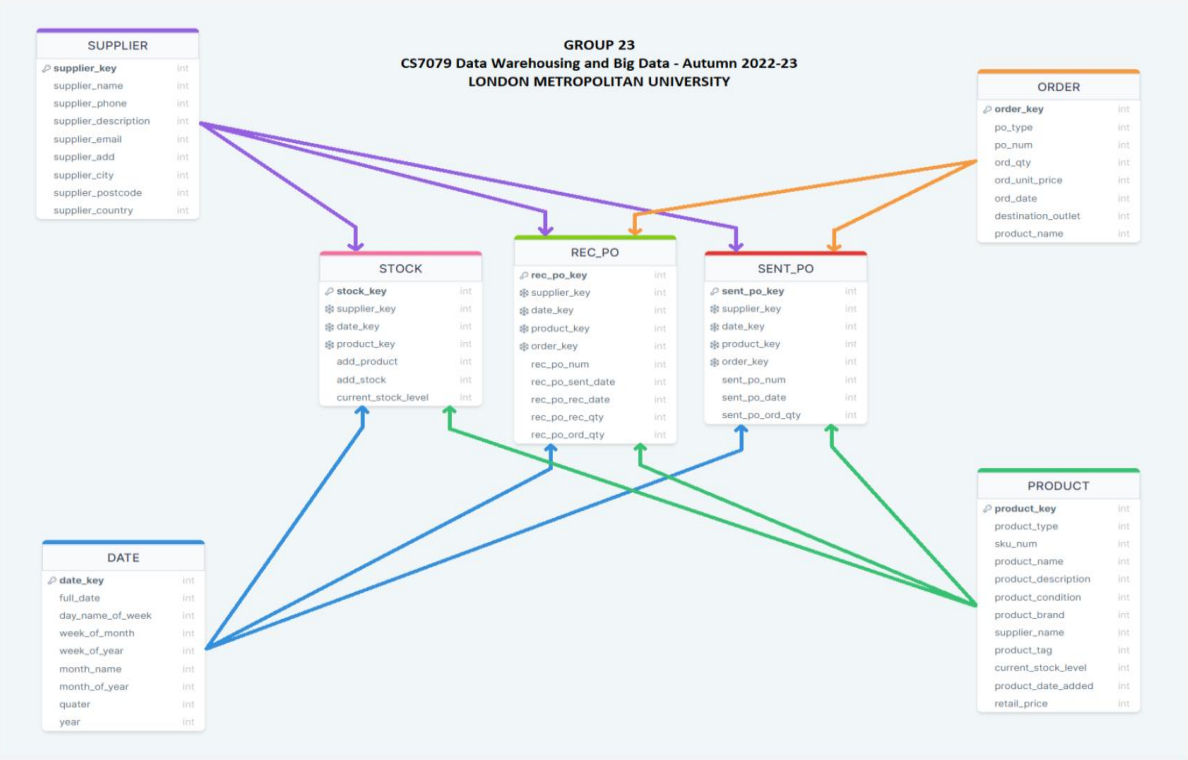
4. Written Report

4.1 An introduction section that summarises the objectives of the course work and business case scenario.

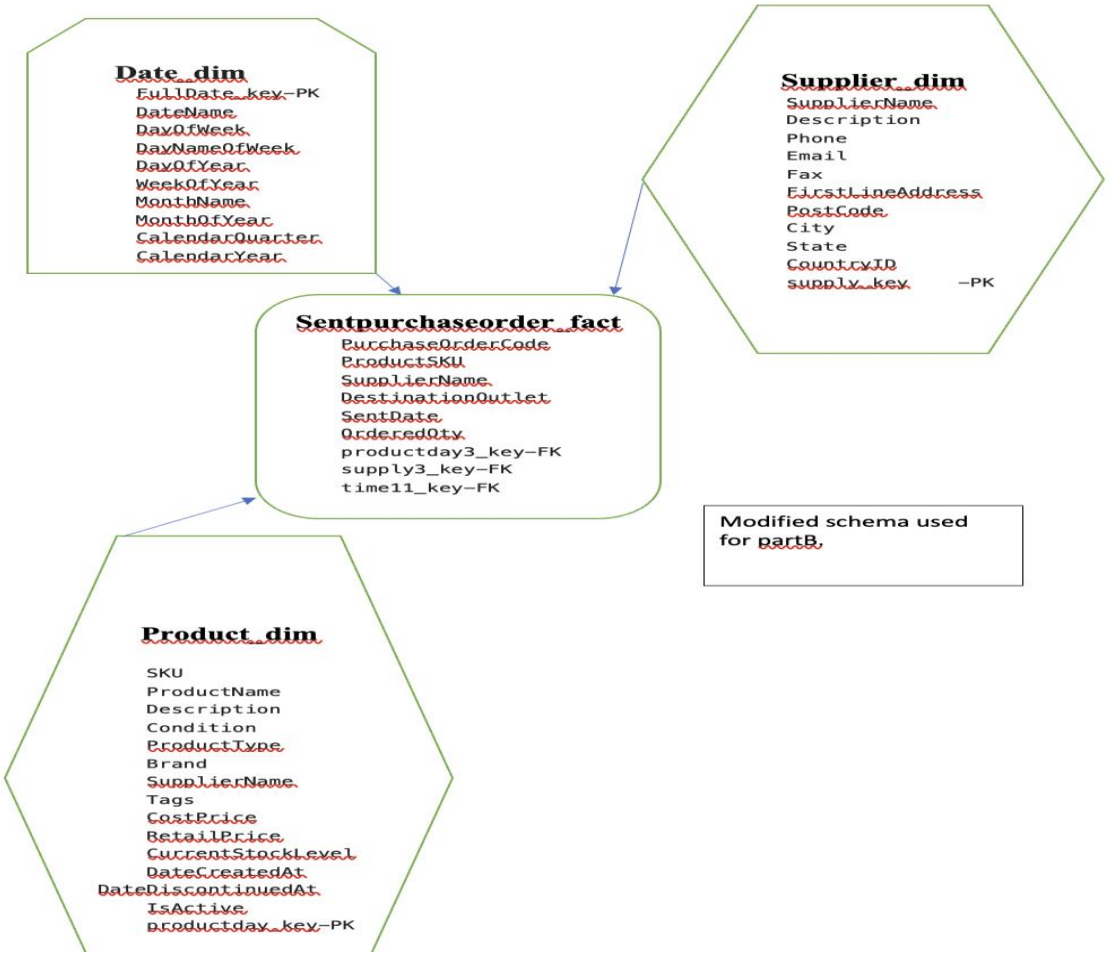
The business case scenario given in this assignment is of a consumer retail company called ABC CONSUMER ELECTRONICS outlet which is a huge electronics retail company which has multiple stores in London and performs online business in the UK and Europe with over 200 brands and 10 categories. Electronic retail business is a competitive industry which makes it hard for companies like ABC electronics to keep up and currently they are having issues in their decision-making activities as they lack to produce reports and are having difficulty in performing analysis. The company is need of a business analyst to solve its problems where their business analyst team has created business document which serves as a guide to help the company. Inventory management being the most important case were controlling and reducing overstocks and understocks are the main activities which needs to be worked on.

Objectives of coursework:
One of the main objectives of the coursework is inventory management which is broken down into 3 steps where purchase orders sent is managed well by checking ordered quantity, sent date, product and supplier detail for every purchase order sent. 2nd step is storing the received purchased orders appropriately and 3rd is maintaining and controlling stocks. Coursework was divided into two parts, in part A we had to design and analyse dimensional data model where we had to create dimensional tables, fact tables, identify grain level to create central fact table and create graphical visualization of a star schema which links dimensional tables to their fact tables. In part b we must create a relational database in SSMS in Microsoft or azure data studio in case of mac where we had to create and store dimensional tables, fact tables, primary and foreign keys. Then we had to migrate the data into Horton works sandbox for further analysis we used hive and pig.

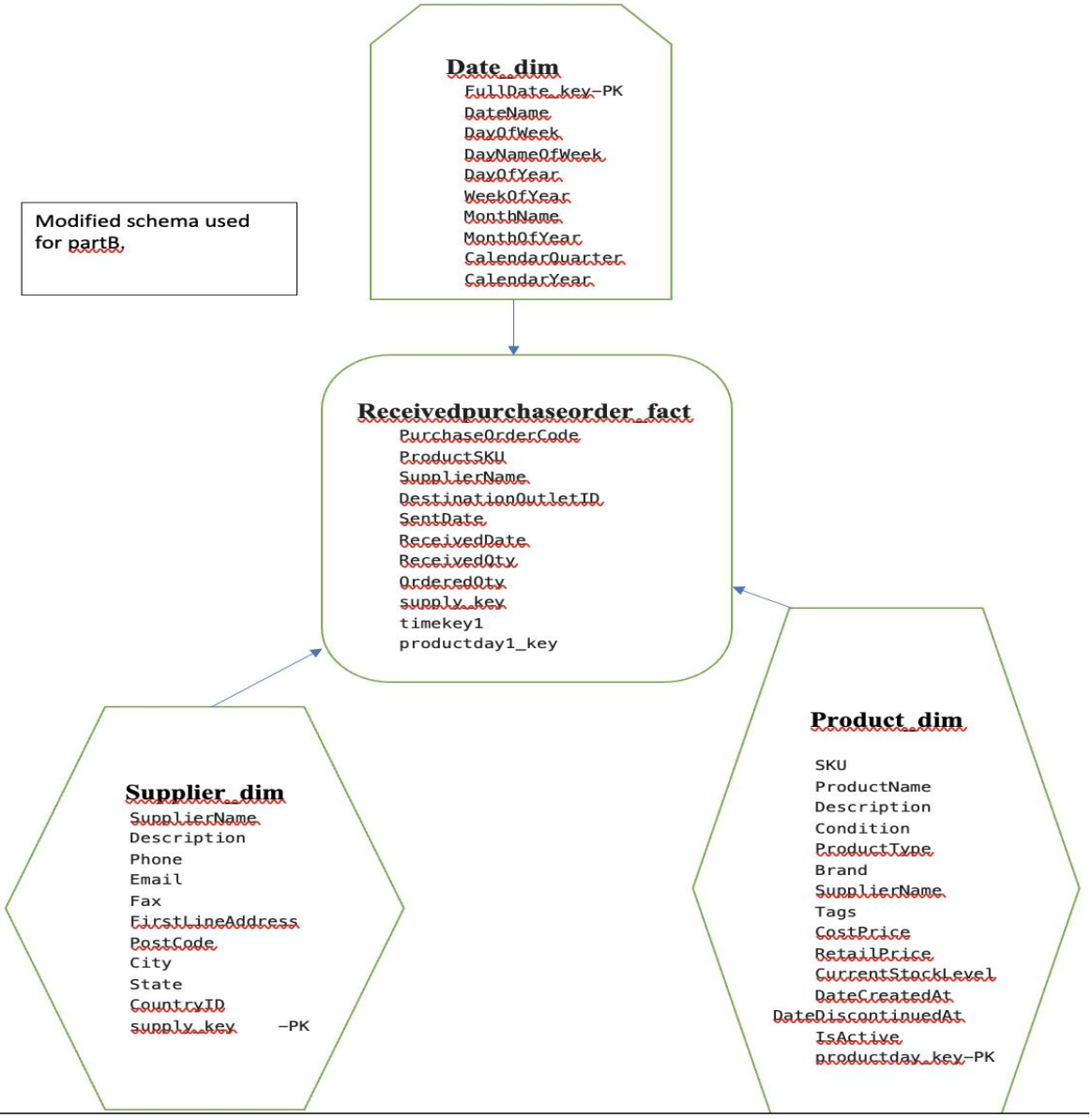
4.2 Analysis and design of dimensional data model.
Design and analysis of dimensional model were done through the raw data provided in txt and csv files. We had to extract the raw data's and identify and create them into dimensional tables and central fact tables where three central fact tables were needed for this coursework. In part a as a group, we came up with 4-dimension tables called supplier, date, order, and product. The fact tables were stock, received purchase order and sent purchase order. Primary keys were made in each dimensional table which were identified as foreign keys in fact tables. In part b, I made a little change in the dimensional table where I removed received and sent dimension table and made supplier, date, and product dimension table as we are allowed to modify some changes in our schema when doing our individual work for part B. the fact tables remained the same.



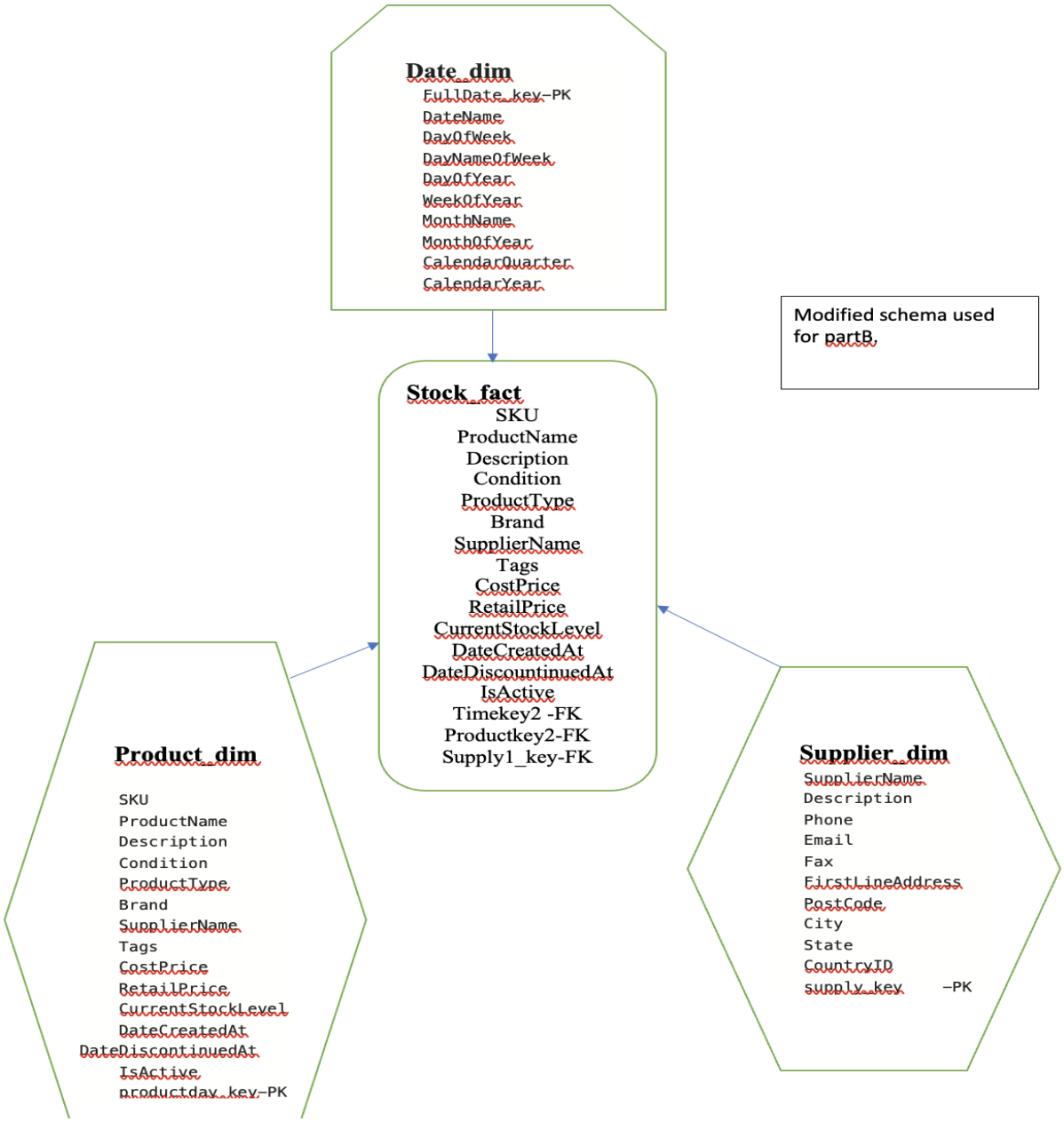
The above figure is the previous schema used for part A.



New schema used for part B. sent purchase order fact schema linked with product, supplier, and date dimensions.

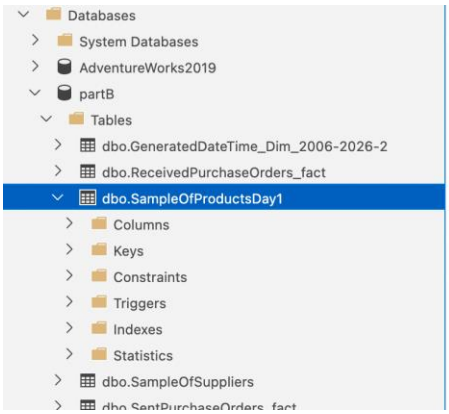


New schema used for part B. received purchase order fact schema linked with product, supplier, and date dimensions.



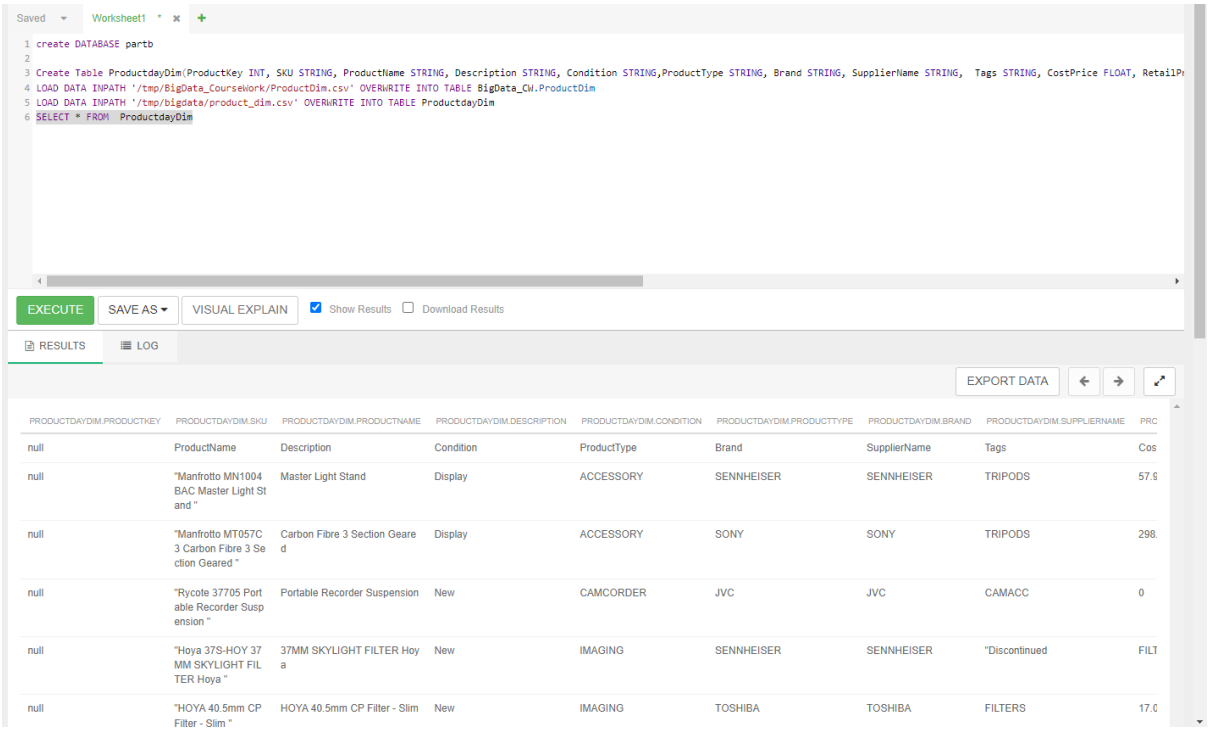
New schema used for part B. stock fact schema linked with product, supplier, and date dimensions.

4.3 Implementation and testing of data warehouse.




```
CREATE TABLE [dbo].[SampleOfProductsDay1] (
    [SKU]               NVARCHAR (50)  NOT NULL,
    [ProductName]       NVARCHAR (100) NOT NULL,
    [Description]       NVARCHAR (50)  NOT NULL,
    [Condition]         NVARCHAR (50)  NOT NULL,
    [ProductType]       NVARCHAR (50)  NOT NULL,
    [Brand]             NVARCHAR (50)  NOT NULL,
    [SupplierName]      NVARCHAR (50)  NOT NULL,
    [Tags]              NVARCHAR (50)  NOT NULL,
    [CostPrice]         FLOAT (53)     NOT NULL,
    [RetailPrice]       FLOAT (53)     NOT NULL,
    [CurrentStockLevel] TINYINT        NOT NULL,
    [DateCreatedAt]     DATE           NOT NULL,
    [DateDiscontinuedAt] DATE          NOT NULL,
    [IsActive]          TINYINT        NOT NULL,
    [productday_key]    INT            IDENTITY (1, 1) NOT NULL,
    CONSTRAINT [productday_key] PRIMARY KEY CLUSTERED ([productday_key] ASC)
);
```

The figure above shows implementation and testing of creating table in data warehouse named azure data studio.



The figure above shows implementation and testing of creating table in data warehouse named Apache hive.

4.4 Implementation and testing of bigdata storage on HDFS.

4.5 Conclusion.

In this coursework I had to learn SQL from scratch as I had no previous knowledge, and it has now become one of my favourite languages. As a mac user I struggled with getting an alternate version of SSMS as it isn't available in mac but found a software called azure data studio in which I have learned many commands and have now become familiar with it interface and can use it with ease. I have learned to use Ambari dashboard which has multiple analytics software like hive, data analytics tool, etc. which was very useful. I learnt about the fundamentals of schemas, different types of schemas like star, snowflake etc., I learnt how to identify and define grain, learned about dimensional tables, fact tables, primary keys, foreign keys, surrogate keys, natural keys. Overall, this course was very useful and sparked an interest to code and learn more often. Although I struggled as I had an mac laptop and everything in this coursework was Microsoft related, after completing this coursework it made me realize coding in mac is more fun.