

# 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: <u>Group 23</u>	IBRAHIM ADEMOLA OYENEYE (22024597)	SAMRAT RAI (21025221)
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.

<b>SENT PURCHASED ORDER</b>
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.

<b>RECEIVED PURCHASED ORDER</b>
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.

<b>STOCK</b>
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	Date	Order	Product
Supplier name	Full date	Purchase order Type	Product Type
Supplier phone number	Day name of the week	Purchase order number	SKU Number
Supplier description	Week of month	Ordered quantity	Product name
Supplier Email	Week of year	Ordered unit price	Product Description
Supplier address	Month name	Ordered date	Product condition
Supplier City	Month of year	Destination outlet	Product brand
Supplier postcode	Quarter	Product name	Supplier name
Supplier country	year	→ order_key	Product tag
→ supplier_key	→ date_key		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.

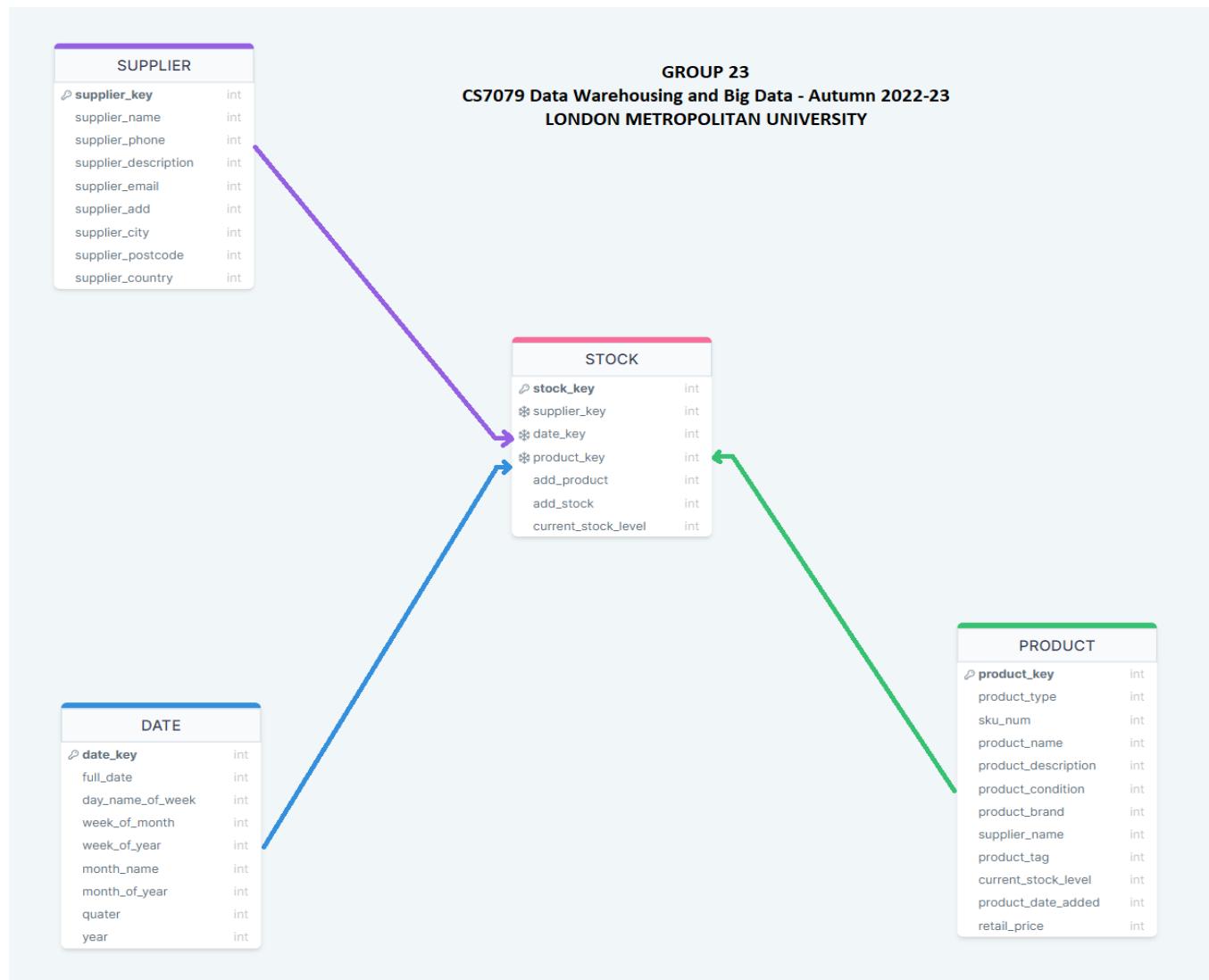


Figure 1: Supplier, Date and Product dimensions are directly connected to the “Stock” fact table.

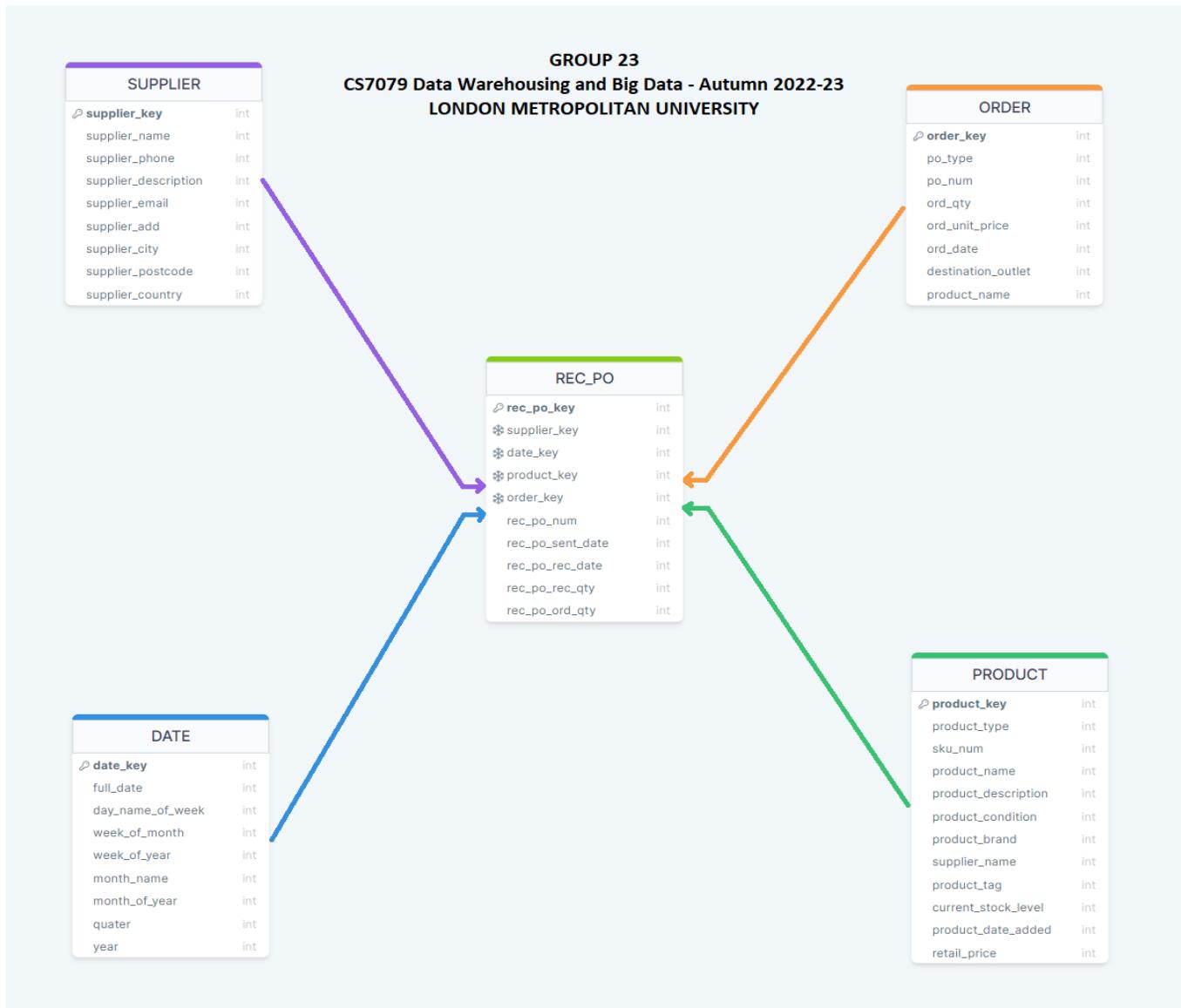


Figure 2: Supplier, Date, Product and order dimensions are directly connected to the "Received Purchased Order" fact table.

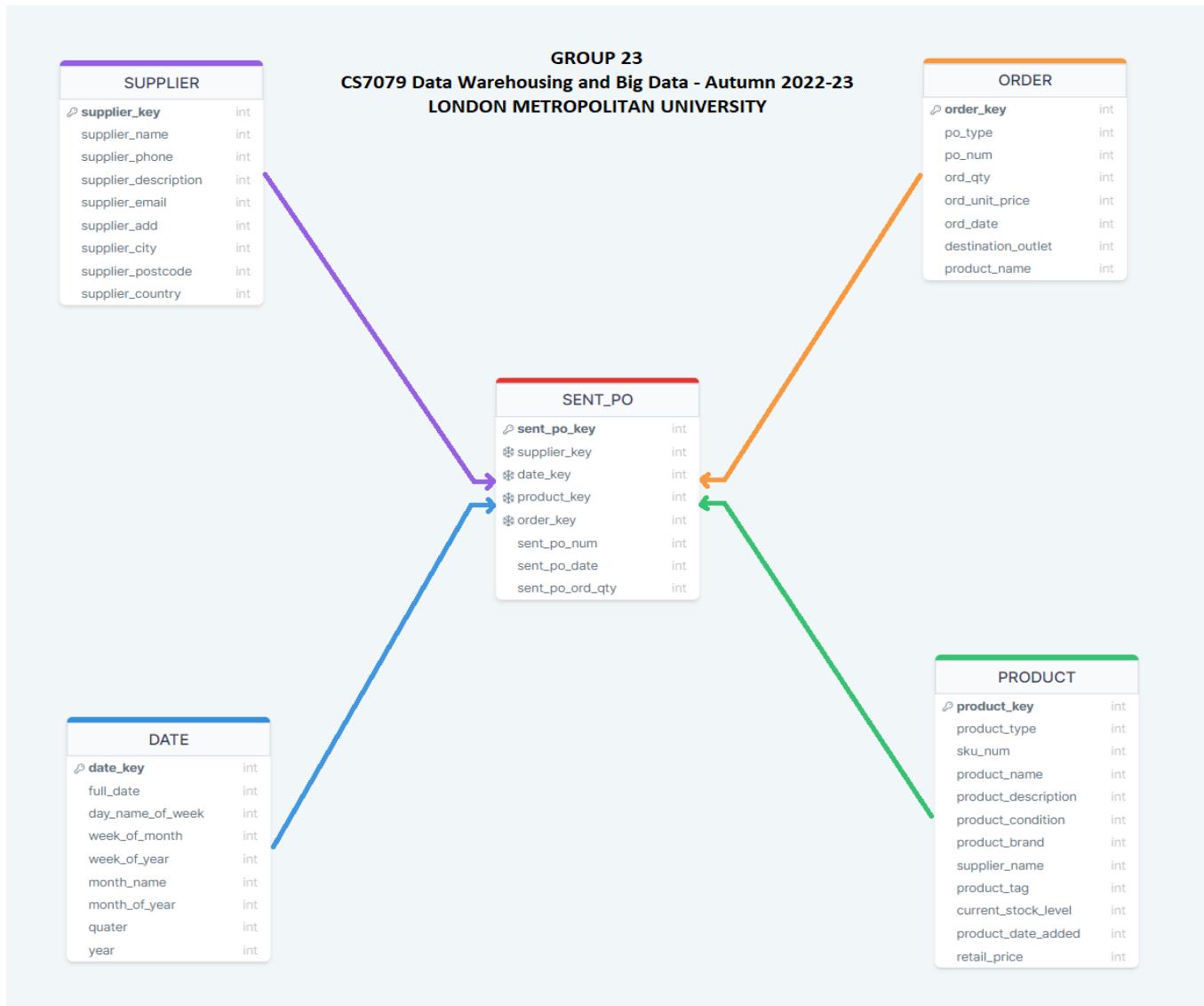


Figure 3: Supplier, Date, Product and order dimensions are directly connected to the "Sent Purchased Order" fact table

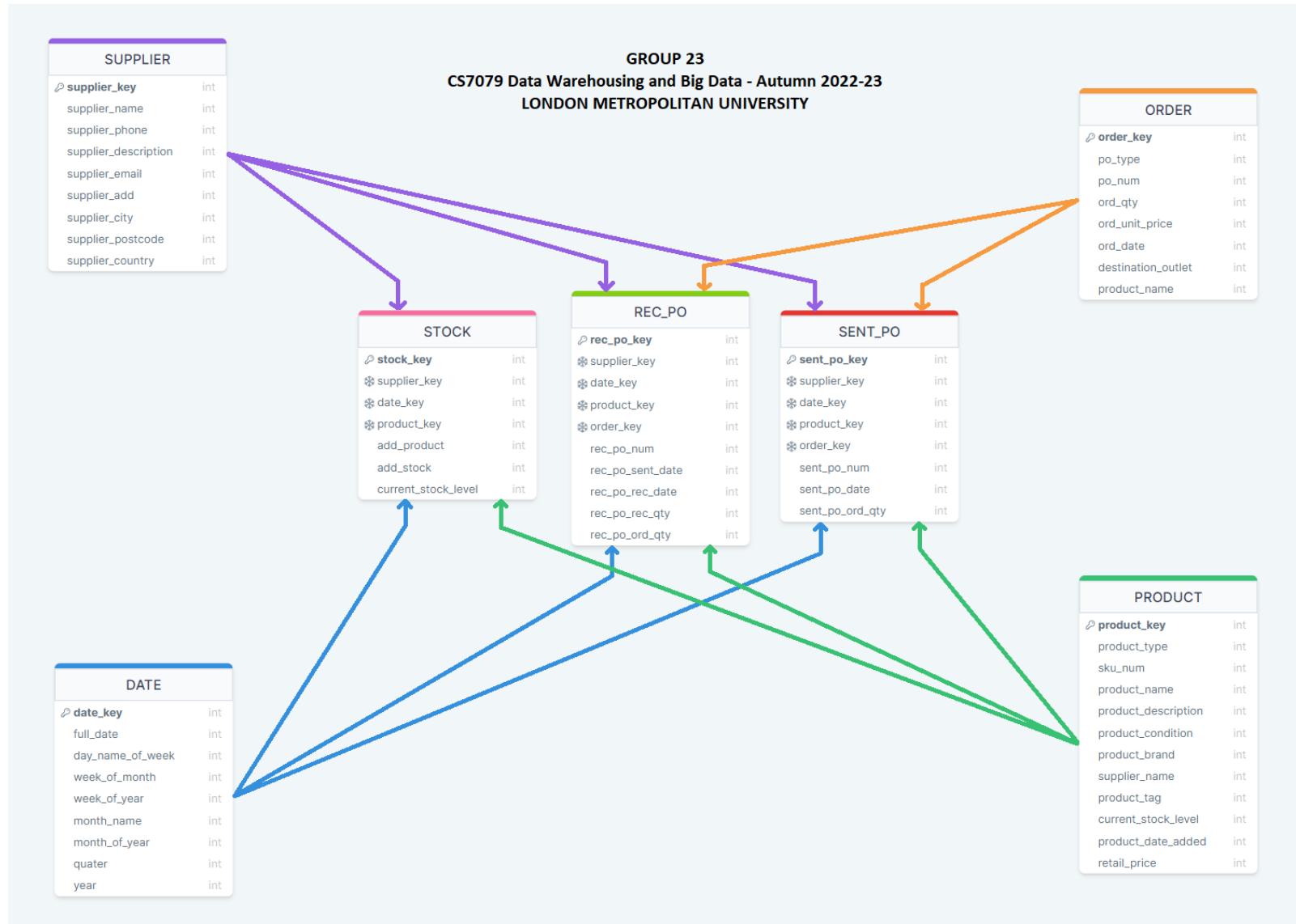
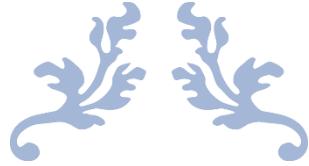


Figure 4: Supplier, Date, Product and order dimensions are connected to central fact tables.



---

## **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:

The screenshot shows the Azure Data Studio interface. On the left, the 'Servers' tree view shows 'localhost, <default> (SA)' with 'Databases' expanded, showing 'AdventureWorks2019', 'partB' (which is selected), and 'ReceivedPurchaseOrders\_fact'. On the right, the main pane displays T-SQL code for creating the database:

```

1 -- Create a new database called 'partB'
2 -- Connect to the 'master' database to run this snippet
3 USE master
4 GO
5 -- Create the new database if it does not exist already
6 IF NOT EXISTS (
7     SELECT [name]
8     FROM sys.databases
9     WHERE [name] = N'partB'
10 )
11 CREATE DATABASE partB
12 GO

```

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:

The screenshot shows the Azure Data Studio interface with multiple tabs open. The 'product\_dim.csv' tab is active, displaying a table of data:

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

The screenshot shows the Azure Data Studio interface. On the left, the object browser displays a tree structure with 'Databases', 'System Databases', 'AdventureWorks2019', 'partB', 'Tables', and 'dbo.GeneratedDateTime\_Dim\_2006-2026-2'. Under this table, 'Columns' is expanded, listing various date-related columns like 'FullDate\_key', 'DateName', 'DayOfWeek', etc. On the right, the results pane shows the T-SQL code for creating the table:

```
[1] 1 CREATE TABLE [dbo].[GeneratedDateTime_Dim_2006-2026-2] (
2     [FullDate_key] DATE NOT NULL,
3     [DateName] DATE NOT NULL,
4     [DayOfWeek] TINYINT NOT NULL,
5     [DayNameOfWeek] NVARCHAR (50) NOT NULL,
6     [DayOfYear] SMALLINT NOT NULL,
7     [WeekOfYear] TINYINT NOT NULL,
8     [MonthName] NVARCHAR (50) NOT NULL,
9     [MonthOfYear] TINYINT NOT NULL,
10    [CalendarQuarter] TINYINT NOT NULL,
11    [CalendarYear] SMALLINT NOT NULL,
12    CONSTRAINT [FullDate_key] PRIMARY KEY CLUSTERED ([FullDate_key] ASC)
13 );
```

Below the code, it says 'Commands completed successfully.' and 'Total execution time: 00:00:00.029'.

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.

The screenshot shows the Azure Data Studio interface. On the left, the object browser displays a tree structure with 'Databases', 'System Databases', 'AdventureWorks2019', 'partB', 'Tables', and 'dbo.SampleOfProductsDay1'. Under this table, 'Columns' is expanded, listing columns like 'SKU', 'ProductName', 'Description', 'Condition', 'ProductType', 'Brand', 'SupplierName', 'Tags', 'CostPrice', 'RetailPrice', 'CurrentStockLevel', and 'DateCreated'. On the right, the results pane shows the T-SQL code for creating the table:

```
[1] 1 CREATE TABLE [dbo].[SampleOfProductsDay1] (
2     [SKU] NVARCHAR (50) NOT NULL,
3     [ProductName] NVARCHAR (100) NOT NULL,
4     [Description] NVARCHAR (50) NOT NULL,
5     [Condition] NVARCHAR (50) NOT NULL,
6     [ProductType] NVARCHAR (50) NOT NULL,
7     [Brand] NVARCHAR (50) NOT NULL,
8     [SupplierName] NVARCHAR (50) NOT NULL,
9     [Tags] NVARCHAR (50) NOT NULL,
10    [CostPrice] FLOAT (53) NOT NULL,
11    [RetailPrice] FLOAT (53) NOT NULL,
12    [CurrentStockLevel] TINYINT NOT NULL,
13    [DateCreatedAt] DATE NOT NULL,
14    [DateDiscontinuedAt] DATE NOT NULL,
15    [IsActive] TINYINT NOT NULL,
16    [productday_key] INT IDENTITY (1, 1) NOT NULL,
17    CONSTRAINT [productday_key] PRIMARY KEY CLUSTERED ([productday_key] ASC)
18 );
```

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.

The screenshot shows the Azure Data Studio interface. On the left, there's a tree view of database objects under 'partB'. A table named 'dbo.SampleOfSuppliers' is selected. On the right, the SQL pane displays the CREATE TABLE statement for this table:

```

[ ] 1 CREATE TABLE [dbo].[SampleOfSuppliers] (
2     [SupplierName] NVARCHAR (50) NOT NULL,
3     [Description] NVARCHAR (50) NULL,
4     [Phone] NVARCHAR (50) NULL,
5     [Email] NVARCHAR (50) NULL,
6     [Fax] NVARCHAR (50) NULL,
7     [FirstLineAddress] NVARCHAR (50) NULL,
8     [PostCode] NVARCHAR (50) NULL,
9     [City] NVARCHAR (50) NULL,
10    [State] NVARCHAR (50) NULL,
11    [CountryID] NVARCHAR (50) NULL,
12    [supply_key] INT IDENTITY (1, 1) NOT NULL,
13    CONSTRAINT [supply_key] PRIMARY KEY CLUSTERED ([supply_key] ASC)
14 );

```

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.

The screenshot shows the Azure Data Studio interface. On the left, there's a tree view of database objects under 'partB'. A table named 'dbo.SentPurchaseOrders\_fact' is selected. On the right, the SQL pane displays the CREATE TABLE statement for this table:

```

[ ] 1 CREATE TABLE [dbo].[SentPurchaseOrders_fact] (
2     [PurchaseOrderCode] NVARCHAR (50) NOT NULL,
3     [ProductSKU] NVARCHAR (50) NOT NULL,
4     [SupplierName] NVARCHAR (50) NOT NULL,
5     [DestinationOutlet] NVARCHAR (50) NOT NULL,
6     [SentDate] DATE NOT NULL,
7     [OrderedQty] TINYINT NOT NULL,
8     [productday3_key] INT NULL,
9     [supply3_key] INT NULL,
10    [time11_key] DATE NULL,
11    CONSTRAINT [FK_productday3_key] FOREIGN KEY ([productday3_key]) REFERENCES [dbo].[SampleOfProductsDay1] ([productday3_key]),
12    CONSTRAINT [FK_supply3_key] FOREIGN KEY ([supply3_key]) REFERENCES [dbo].[SampleOfSuppliers] ([supply_key]),
13    CONSTRAINT [FK_time11_key] FOREIGN KEY ([time11_key]) REFERENCES [dbo].[GeneratedDateTime_Dim_2006-2026-2] ([FullDate])
14 );
15

```

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.

The screenshot shows the Azure Data Studio interface. At the top, there are five tabs: 'localhost' (selected), 'dbo.SampleOfProductsDay1\_1', 'dbo.SampleOfSuppliers\_1', 'dbo.SentPurchaseOrders\_fact\_1', and 'dbo.stock\_fact\_1'. Below the tabs, there is a toolbar with 'Run' and 'Stop' buttons, and a dropdown for 'Max Rows' set to 200. A 'Show SQL Pane' button is also present.

The main area displays a table with 20 rows of data. The columns are: SKU, ProductName, Description, Condition, ProductType, Brand, SupplierName, Tags, CostPrice, RetailPrice, CurrentStockLevel, and DateCreated. The data includes various camera and lens models from brands like Manfrotto, JVC, SONY, TOSHIBA, and HOYA.

On the left side, there is a tree view of the database structure under 'partB'. It shows 'Tables' containing 'GeneratedDateTime\_Dim\_2006-2026-2', 'ReceivedPurchaseOrders\_fact', 'SampleOfProductsDay1', 'SampleOfSuppliers', and 'SentPurchaseOrders\_fact'. Below 'Tables' are 'Columns', 'Keys', 'Constraints' (which is highlighted with a blue selection box), 'Triggers', 'Indexes', 'Statistics', 'Views', and 'Synonyms'.

On the right side, the 'SQL' tab contains the CREATE TABLE script for 'dbo.stock\_fact\_1':

```

1 CREATE TABLE [dbo].[stock_fact] (
2     [SKU] NVARCHAR (50) NOT NULL,
3     [ProductName] NVARCHAR (100) NOT NULL,
4     [Description] NVARCHAR (50) NOT NULL,
5     [Condition] NVARCHAR (50) NOT NULL,
6     [ProductType] NVARCHAR (50) NOT NULL,
7     [Brand] NVARCHAR (50) NOT NULL,
8     [SupplierName] NVARCHAR (50) NOT NULL,
9     [Tags] NVARCHAR (50) NOT NULL,
10    [CostPrice] FLOAT (53) NOT NULL,
11    [RetailPrice] FLOAT (53) NOT NULL,
12    [CurrentStockLevel] TINYINT NOT NULL,
13    [DateCreated] DATE NOT NULL,
14    [DateDiscontinued] DATE NOT NULL,
15    [IsActive] TINYINT NOT NULL,
16    [timekey2] DATE NULL,
17    [productday2_key] INT NULL,
18    [supply1_key] INT NULL,
19    CONSTRAINT [FK_Fulldate_key] FOREIGN KEY ([timekey2]) REFERENCES [dbo].[GeneratedDateTime_Dim_2006-2026-2] ([FullDate])
20    CONSTRAINT [FK_productday2_key] FOREIGN KEY ([productday2_key]) REFERENCES [dbo].[SampleOfProductsDay1] ([productday_k])
21    CONSTRAINT [FK_supply1_key] FOREIGN KEY ([supply1_key]) REFERENCES [dbo].[SampleOfSuppliers] ([supply_key])
22 );

```

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.

The screenshot shows the Azure Data Studio interface. At the top, there are five tabs: 'localhost' (selected), 'dbo.SampleOfProductsDay1\_1', 'dbo.SampleOfSuppliers\_1', 'dbo.SentPurchaseOrders\_fact\_1', and 'dbo.stock\_fact\_1'. Below the tabs, there is a toolbar with 'Run' and 'Stop' buttons, and a dropdown for 'Max Rows' set to 200. A 'Show SQL Pane' button is also present.

The main area displays a table with 16 rows of data. The columns are: PurchaseOrder, ProductSKU, SupplierName, DestinationOutletID, SentDate, ReceivedDate, ReceivedQty, OrderedQty, supply\_key, timekey1, and productday1\_k\_. The data includes various purchase orders from suppliers like SENNHEISER, SONY, JVC, and TOSHIBA.

On the left side, there is a tree view of the database structure under 'partB'. It shows 'Tables' containing 'GeneratedDateTime\_Dim\_2006-2026-2', 'ReceivedPurchaseOrders\_fact' (highlighted with a blue selection box), and 'SampleOfProductsDay1'. Below 'Tables' are 'Columns', 'Keys', 'Constraints', 'Triggers', 'Indexes', 'Statistics', and 'Views'.

On the right side, the 'SQL' tab contains the CREATE TABLE script for 'dbo.ReceivedPurchaseOrders\_fact\_1':

```

1 CREATE TABLE [dbo].[ReceivedPurchaseOrders_fact] (
2     [PurchaseOrderCode] NVARCHAR (50) NOT NULL,
3     [ProductSKU] NVARCHAR (50) NOT NULL,
4     [SupplierName] NVARCHAR (50) NOT NULL,
5     [DestinationOutletID] NVARCHAR (50) NOT NULL,
6     [SentDate] DATE NOT NULL,
7     [ReceivedDate] DATE NOT NULL,
8     [ReceivedQty] TINYINT NOT NULL,
9     [OrderedQty] TINYINT NOT NULL,
10    [supply_key] INT NULL,
11    [timekey1] DATE NULL,
12    [productday1_key] INT NULL,
13    CONSTRAINT [FK_productday1_key] FOREIGN KEY ([productday1_key]) REFERENCES [dbo].[SampleOfProductsDay1] ([productday_k])
14    CONSTRAINT [FK_supply1_key] FOREIGN KEY ([supply_key]) REFERENCES [dbo].[SampleOfSuppliers] ([supply_key]),
15    CONSTRAINT [FK_timekey1] FOREIGN KEY ([timekey1]) REFERENCES [dbo].[GeneratedDateTime_Dim_2006-2026-2] ([FullDate])
16 );

```

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(productday3\_key as foregin key in SentpurchaseOrders\_fact).

```
alter table [dbo].[SentPurchaseOrders_fact] add
productday3_key int
alter table [dbo].[SentPurchaseOrders_fact] add
CONSTRAINT FK_productday3_key foreign key (productday3_key)
references [dbo].[SampleOfProductsDay1] (productday_key)
select * from [dbo].[SentPurchaseOrders_fact]
update [dbo].[SentPurchaseOrders_fact]
set productday3_key=(select productday_key from [dbo].[SampleOfProductsDay1])
where [dbo].[SentPurchaseOrders_fact].ProductSKU=[dbo].[SampleOfProductsDay1].SKU
select * from [dbo].[SentPurchaseOrders_fact]
```

code used for putting in primary and foreign keys in stock table(supply3\_key as foregin key in SentPurchaseOrders\_fact table).

```
alter table [dbo].[SentPurchaseOrders_fact] add
supply3_key int
alter table [dbo].[SentPurchaseOrders_fact] add
CONSTRAINT FK_supply3_key foreign key (supply3_key)
references [dbo].[SampleOfSuppliers] (supply_key)
select * from [dbo].[SentPurchaseOrders_fact]
update [dbo].[SentPurchaseOrders_fact]
set supply3_key=(select supply_key from [dbo].[SampleOfSuppliers])
where [dbo].[SentPurchaseOrders_fact].SupplierName=[dbo].[SampleOfSuppliers].SupplierName
select * from [dbo].[SentPurchaseOrders_fact]
```

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)
```

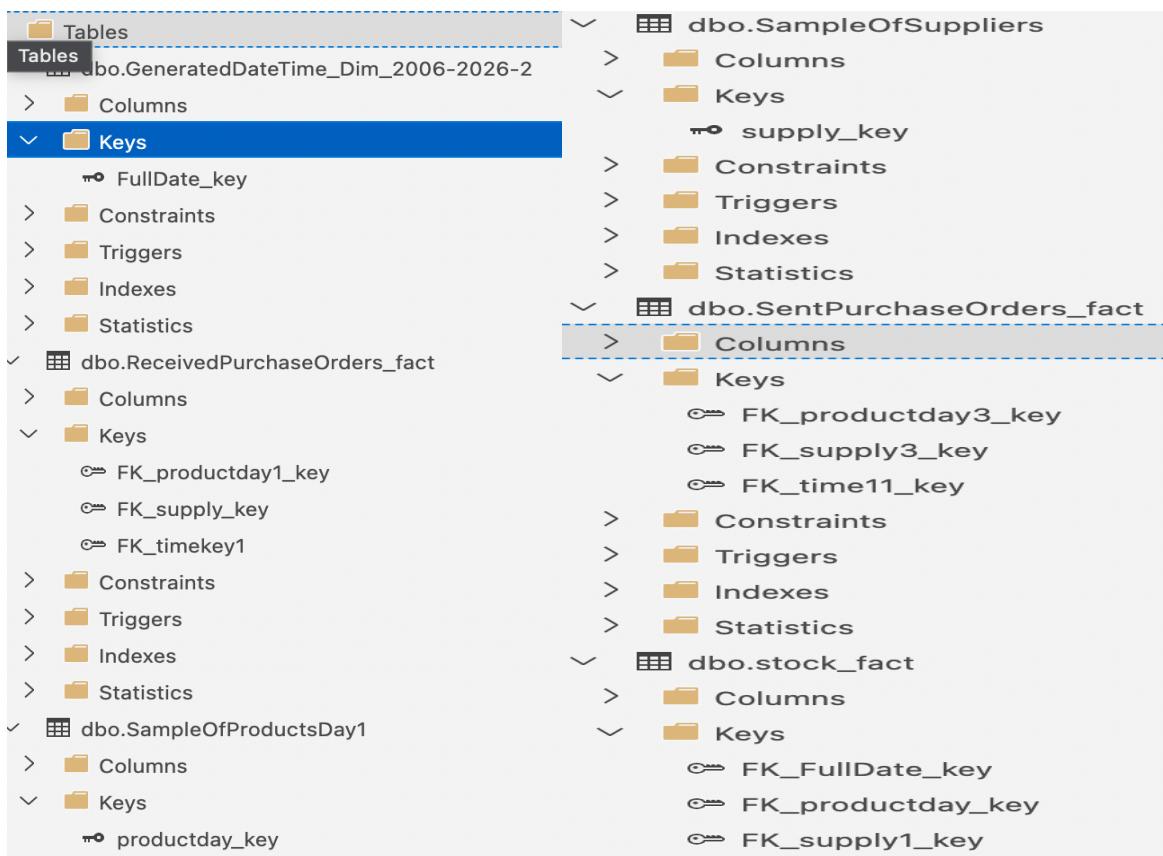
code used for putting in primary and foreign keys in stock table(time11\_key as foregin key in SentPurchaseOrders\_fact table).

```
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].[DirectoryName])


---


```

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.

User	Group	Permission	Erasure Coding	Encrypted
hdfs	hadoop	drwxrwxrwx	No	
hdfs	hadoop	-rw-r--r--	No	
hive	hdfs	drwxr-x-x	No	
part+b	maria_dev	drwxr-x-x	No	
parts	maria_dev	drwxr-x-x	No	

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.

Name	Size	Last Modified	Owner	Group	Permission	Erasure Coding	Encrypted
coursework.part.b	--	2023-01-11 17:13	maria_dev	hdfs	drwxrwxrwx		No
date_dim.csv	420.2 kB	2023-01-11 17:12	maria_dev	hdfs	-rw-r--r--		No
druid-indexing	--	2018-11-29 19:01	druid	hadoop	drwxrwxr-x		No
entity-file-history	--	2018-11-29 17:25	hdfs	hdfs	drwxr-xr-x		No
hive	--	2018-11-29 17:58	hive	hdfs	drwxrwx-wx		No
part+b	--	2023-01-11 17:11	maria_dev	hdfs	drwxr-xr-x		No
partb	--	2023-01-11 17:12	maria_dev	hdfs	drwxr-xr-x		No

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.

Name	Size	Last Modified	Owner	Group	Permission	Erasure Coding	Encrypted
date_dim.csv	420.2 kB	2023-01-11 17:12	maria_dev	hdfs	-rw-r--r--		No
product_dim.csv	3.2 kB	2023-01-11 17:13	maria_dev	hdfs	-rw-r--r--		No
receivedpurchaseorder_fact.csv	1.0 kB	2023-01-11 17:13	maria_dev	hdfs	-rw-r--r--		No
sentpurchase_fact.csv	1.0 kB	2023-01-11 17:13	maria_dev	hdfs	-rw-r--r--		No
stock_fact.csv	3.4 kB	2023-01-11 17:13	maria_dev	hdfs	-rw-r--r--		No
suppliers_dim.csv	0.7 kB	2023-01-11 17:13	maria_dev	hdfs	-rw-r--r--		No

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 screenshot shows the Azure Data Studio interface. In the top left, there's a sidebar with 'LAST UPDATE 6 sec ago' and a 'default' database selected. Below it, under 'TABLES', it says 'No tables found'. The main area is a code editor with the following SQL script:

```

1 create DATABASE partb
2
3 Create Table date_dim(FullDate_key STRING, Datename STRING , DayOfWeek INT , DaynameOfWeek STRING , DayOfYear INT , WeekOfYear INT, MonthName STRING, MonthOfYear INT, CalendarQuarter INT, CalendarYear INT)
4 LOAD DATA INPATH '/tmp/bigdata/date_dim.csv' OVERWRITE INTO TABLE date_dim
5 SELECT * FROM date_dim
6
7

```

Below the code editor are buttons for 'EXECUTE', 'SAVE AS', 'VISUAL EXPLAIN', 'Show Results' (which is checked), and 'Download Results'. The results tab is active, showing a table with data from January 1, 2006, to January 9, 2006. The columns are: DATE\_DIM.FULLDATE\_KEY, DATE\_DIM.DATENAME, DATE\_DIM.DAYOFWEEK, DATE\_DIM.DAYNAMEOFWEEK, DATE\_DIM.DAYOFYEAR, DATE\_DIM.WEEKOFYEAR, DATE\_DIM.MONTHNAME, DATE\_DIM.MONTHOFYEAR, DATE\_DIM.CALENDARQUARTER, and DATE\_DIM.CAL. The data looks like this:

DATE_DIM.FULLDATE_KEY	DATE_DIM.DATENAME	DATE_DIM.DAYOFWEEK	DATE_DIM.DAYNAMEOFWEEK	DATE_DIM.DAYOFYEAR	DATE_DIM.WEEKOFYEAR	DATE_DIM.MONTHNAME	DATE_DIM.MONTHOFYEAR	DATE_DIM.CALENDARQUARTER	DATE_DIM.CAL
2006-01-01	2006-01-01	1	Sunday	1	1	January	1	1	2006
2006-01-02	2006-01-02	2	Monday	2	1	January	1	1	2006
2006-01-03	2006-01-03	3	Tuesday	3	1	January	1	1	2006
2006-01-04	2006-01-04	4	Wednesday	4	1	January	1	1	2006
2006-01-05	2006-01-05	5	Thursday	5	1	January	1	1	2006
2006-01-06	2006-01-06	6	Friday	6	1	January	1	1	2006
2006-01-07	2006-01-07	7	Saturday	7	1	January	1	1	2006
2006-01-08	2006-01-08	1	Sunday	8	2	January	1	1	2006
2006-01-09	2006-01-09	2	Monday	9	2	January	1	1	2006

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 screenshot shows the Azure Data Studio interface. In the top left, there's a sidebar with 'LAST UPDATE 6 sec ago' and a 'default' database selected. The main area is a code editor with the following SQL script:

```

1 create DATABASE partb
2
3 Create Table ProductdayDim(ProductKey INT, SKU STRING, ProductName STRING, Description STRING, Condition STRING, ProductType STRING, Brand STRING, SupplierName STRING, Tags STRING, CostPrice FLOAT, RetailPrice FLOAT)
4 LOAD DATA INPATH '/tmp/BigData_Coursework/Productdim.csv' OVERWRITE INTO TABLE BigData_CW.ProductDim
5 LOAD DATA INPATH '/tmp/bigdata/product_dim.csv' OVERWRITE INTO TABLE ProductdayDim
6 SELECT * FROM ProductdayDim

```

Below the code editor are buttons for 'EXECUTE', 'SAVE AS', 'VISUAL EXPLAIN', 'Show Results' (which is checked), and 'Download Results'. The results tab is active, showing a table with product data. The columns are: PRODUCTDAYDIM.PRODUCTKEY, PRODUCTDAYDIM.SKU, PRODUCTDAYDIM.PRODUCTNAME, PRODUCTDAYDIM.DESCRIPTION, PRODUCTDAYDIM.CONDITION, PRODUCTDAYDIM.PRODUCTTYPE, PRODUCTDAYDIM.BRAND, PRODUCTDAYDIM.SUPPLIERNAME, PRC, and COS. The data looks like this:

PRODUCTDAYDIM.PRODUCTKEY	PRODUCTDAYDIM.SKU	PRODUCTDAYDIM.PRODUCTNAME	PRODUCTDAYDIM.DESCRIPTION	PRODUCTDAYDIM.CONDITION	PRODUCTDAYDIM.PRODUCTTYPE	PRODUCTDAYDIM.BRAND	PRODUCTDAYDIM.SUPPLIERNAME	PRC	COS
null	ProductName	Description	Condition	ProductType	Brand	SupplierName	Tags	CostPrice	RetailPrice
null	"Manfrotto MN1004 BAC Master Light Stand and"	Master Light Stand	Display	ACCESSORY	SENNHEISER	SENNHEISER	TRIPODS	57.9	69.9
null	"Manfrotto MT057C 3 Carbon Fibre 3 Section Geared "	Carbon Fibre 3 Section Geared	Display	ACCESSORY	SONY	SONY	TRIPODS	298.0	350.0
null	"Rycote 37705 Portable Recorder Suspensionension."	Portable Recorder Suspension	New	CAMCORDER	JVC	JVC	CAMACC	0	100.0
null	"Hoya 37S-HOY 37 MM SKYLIGHT FILTER HOYA"	37MM SKYLIGHT FILTER HOYA	New	IMAGING	SENNHEISER	SENNHEISER	"Discontinued"	FILTERS	17.0
null	"HOYA 40.5mm CP Filter - Slim"	HOYA 40.5mm CP Filter - Slim	New	IMAGING	TOSHIBA	TOSHIBA	FILTERS	17.0	20.0

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 screenshot shows the Apache Ambari Dashboard interface. At the top, there are tabs for 'Saved' and several worksheets. Below the tabs, a code editor displays the following SQL script:

```

1 Create Table stock_fact(SKU STRING, ProductName STRING, Description STRING, Condition STRING, ProductType STRING, Brand STRING, SupplierName STRING, Tags STRING, CostPrice INT, RetailPrice INT, CurrentStock INT)
2 LOAD DATA INPATH '/tmp/school/stock_fact.csv' OVERWRITE INTO TABLE stock_fact
3 SELECT * FROM stock_fact
4

```

Below the code editor are buttons for 'EXECUTE', 'SAVE AS', 'VISUAL EXPLAIN', 'Show Results' (which is checked), and 'Download Results'. The results section is titled 'RESULTS' and contains a table with the following data:

STOCK_FACT.SKU	STOCK_FACT.PRODUCTNAME	STOCK_FACT.DESCRIPTION	STOCK_FACT.CONDITION	STOCK_FACT.PRODUCTTYPE	STOCK_FACT.BRAND	STOCK_FACT.SUPPLIERNAME	STOCK_FACT.TAGS	STOCK_FACT.COSTPRICE	STOCK_FACT.RETAILPRICE
SKU	ProductName	Description	Condition	ProductType	Brand	SupplierName	Tags	null	null
SEN23322	Manfrotto MN1004BAC Master Light Stand	Master Light Stand	Display	ACCESSORY	SENNHEISER	SENNHEISER	TRIPODS	57	114
S06677	Manfrotto MT057C3 Carbon Fibre 3 Section Geared	Carbon Fibre 3 Section Geared	Display	ACCESSORY	SONY	SONY	TRIPODS	298	584
JV2222	Rycote 37705 Portable Recorder Suspension	Portable Recorder Suspension	New	CAMCORDER	JVC	JVC	CAMACC	0	59
SEN222	Hoya 37SHOY 37MM SKYLIGHT FILTER Hoya	37MM SKYLIGHT FILTER Hoya	New	IMAGING	SENNHEISER	SENNHEISER	"Discontinued"	null	0
TOW222	HOYA 40.5mm CP Filter - Slim	HOYA 40.5mm CP Filter - Slim	New	IMAGING	TOSHIBA	TOSHIBA	FILTERS	17	34
MS7771	Rycote 41118 Portable Recorder Suspension	Portable Recorder Suspension	New	CAMCORDER	MSCS	MSCS	CAMACC	0	59

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 screenshot shows the Apache Ambari Dashboard interface. At the top, there are tabs for 'Saved' and several worksheets. Below the tabs, a code editor displays the following SQL script:

```

1 Create Table sentpurchase_fact (SupplierName STRING, Description STRING, Phone STRING, Email STRING, Fax STRING, FirstLineAddress STRING, PostCode INT, City STRING, State STRING, CountryID STRING, supply_k)
2 LOAD DATA INPATH '/tmp/school/sentpurchase_fact.csv' OVERWRITE INTO TABLE sentpurchase_fact
3 SELECT * FROM sentpurchase_fact
4

```

Below the code editor are buttons for 'EXECUTE', 'SAVE AS', 'VISUAL EXPLAIN', 'Show Results' (which is checked), and 'Download Results'. The results section is titled 'RESULTS' and contains a table with the following data:

SENTPURCHASE_FACT.SUPPLIERNAME	SENTPURCHASE_FACT.DESCRIPTION	SENTPURCHASE_FACT.PHONE	SENTPURCHASE_FACT.EMAIL	SENTPURCHASE_FACT.FAX	SENTPURCHASE_FACT.FIRSTLINEADDRESS	SENTPURCHASE_FACT.POSTCODE	SENTPURCHASE_FACT.COUNTRYID
PurchaseOrderCode	ProductSKU	SupplierName	DestinationOutlet	SentDate	OrderedQty	null	supply3_k
SA301015 TOSHIBA CS PO 1	TOMCC	TOSHIBA	ABC Warehouse	2015-10-30	3	11	10
SA301015 SONY PO 1	SO9999	Sony	ABC Warehouse	2015-10-30	2	12	9
SA301015 JVC CS PO 1	JVRRR2	JVC	ABC Warehouse	2015-10-30	1	13	5
SA301015 SUMSUNG CS PO 1	SUM3444	Samsung	ABC Warehouse	2015-10-30	1	14	7
SA301015 TOSHIBA CS PO 1	TOHDCC	TOSHIBA	ABC Warehouse	2015-10-30	1	15	10
SA301015 CANON CS PO 1	CANI999	CANON	ABC Warehouse	2015-10-30	1	16	1
SA301015 JVC CS PO 1	JV66622	JVC	ABC Warehouse	2015-10-30	1	17	5
SA301015 TOSHIBA CS PO 1	TO2333	TOSHIBA	ABC Warehouse	2015-10-30	5	18	10
SA301015 SAMSUNG CS PO 1	SAMrr22	Samsung	ABC Warehouse	2015-10-30	1	19	7

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.

Saved    Worksheet1 \* x   Worksheet2 \* x +

```

1 Create Table suppliers_dim(SupplierName STRING, Description STRING, Phone STRING, Email STRING, Fax STRING, FirstLineAddress STRING, PostCode INT, City STRING, State STRING, CountryID STRING, supply_key INT
2 LOAD DATA INPATH '/tmp/school/suppliers_dim.csv' OVERWRITE INTO TABLE suppliers_dim
3 SELECT * FROM suppliers_dim

```

**EXECUTE** **SAVE AS** **VISUAL EXPLAIN**  Show Results  Download Results

**RESULTS** **LOG**

**EXPORT DATA** **←** **→** **↶** **↷**

SUPPLIERS_DIM.SUPPLIERNAME	SUPPLIERS_DIM.DESCRIPTION	SUPPLIERS_DIM.PHONE	SUPPLIERS_DIM.EMAIL	SUPPLIERS_DIM.FAX	SUPPLIERS_DIM.FIRSTLINEADDRESS	SUPPLIERS_DIM.POSTCODE	SUPPLIERS_DIM.CITY	SUPPLIERS_DIM.STATE	SUPPLIERS_DIM.COUNTRYID
CANON	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
ENE	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
HAMA	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
HILLS	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
JVC	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
MSCS	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
Samsung	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
SENNHEISER	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"
SONY	"NULL"	"NULL"	"NULL"	"NULL"	"NULL"	null	"NULL"	"NULL"	"NUI"

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.

Saved    Worksheet1 \* x +

```

1 CREATE DATABASE school
2 Create Table receivedpurchaseorder_fact(PurchaseOrderCode STRING, ProductSKU STRING, SupplierName STRING, DestinationOutletID STRING, SentDate STRING, ReceivedDate STRING, ReceivedQty INT, OrderedQty INT, :_
3 LOAD DATA INPATH '/tmp/school/receivedpurchaseorder_fact.csv' OVERWRITE INTO TABLE receivedpurchaseorder_fact
4 SELECT * FROM receivedpurchaseorder_fact
5

```

**EXECUTE** **SAVE AS** **VISUAL EXPLAIN**  Show Results  Download Results

**RESULTS** **LOG**

**EXPORT DATA** **←** **→** **↶** **↷**

RECEIVEDPURCHASEORDER_FACT.PURCHASEORDERCODE	RECEIVEDPURCHASEORDER_FACT.PRODUCTSKU	RECEIVEDPURCHASEORDER_FACT.SUPPLIERNAME	RECEIVEDPURCHASEORDER_FACT.DESTINATIONOUTLETID	RECEIVEDPURCHASEORDER_FACT.SENTDATE
SA311016 SENNHEISER PRO	SEN23322	SENNHEISER	ABC Warehouse	2016-10-31
SA311016 SONY PRO	SO6677	SONY	ABC Warehouse	2016-10-31
SA311016 JVC PRO	JV2222	JVC	ABC Warehouse	2016-10-31
SA311016 SENNHEISER PRO	SEN222	SENNHEISER	ABC Warehouse	2016-10-31
SA311016 TOSHIBA PRO	TOW222	TOSHIBA	ABC Warehouse	2016-10-31
SA1406	MS7771	MSCS	ABC Warehouse	2016-11-16
PM1611COMPUBA02	CO8211	TOSHIBA	ABC Warehouse	2016-11-16
PM0811COMPUB	CO2J111	MSCS	ABC Warehouse	2016-11-08
"SA071116 SOLOCO "	SOL2222	MSCS	ABC Warehouse	2016-11-07

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.**

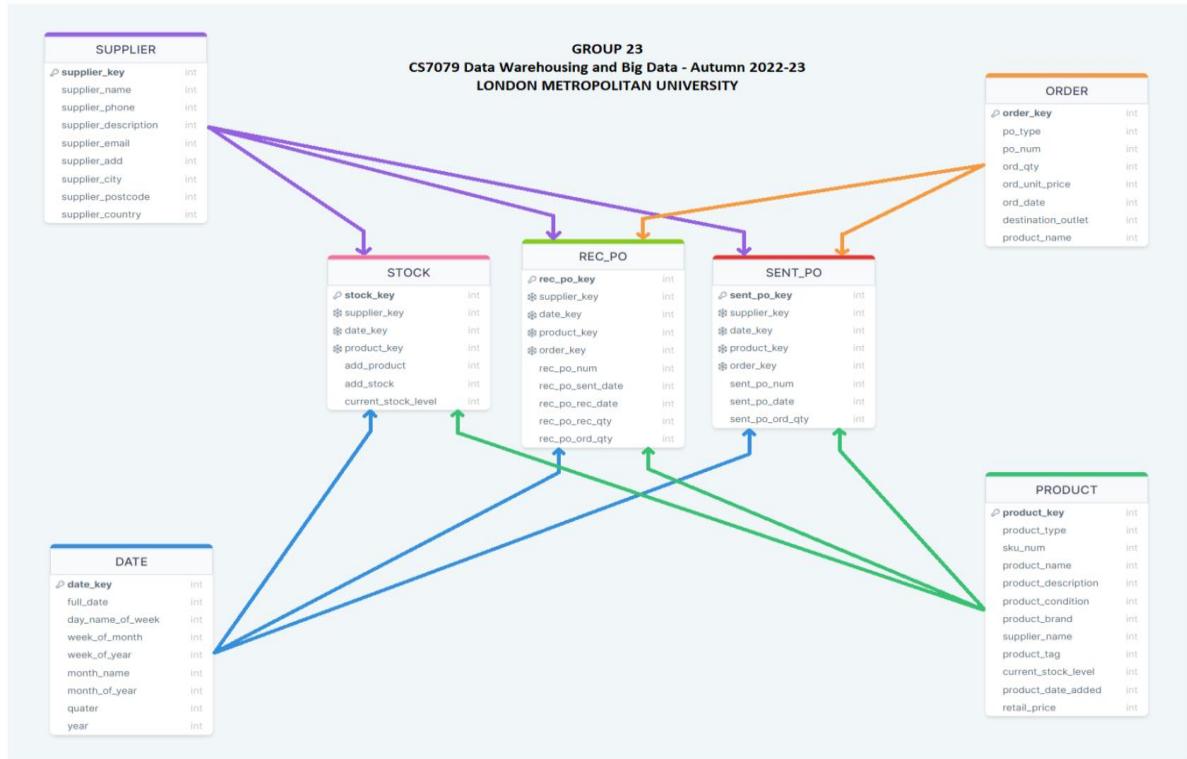
The business case scenario given in this assignment is of a consumer retail company called ABC CONSUMER ELECTRONICS outlet which is a huge electronics retail company which has multiple stores in London and performs online business in the UK and Europe with over 200 brands and 10 categories. Electronic 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:**

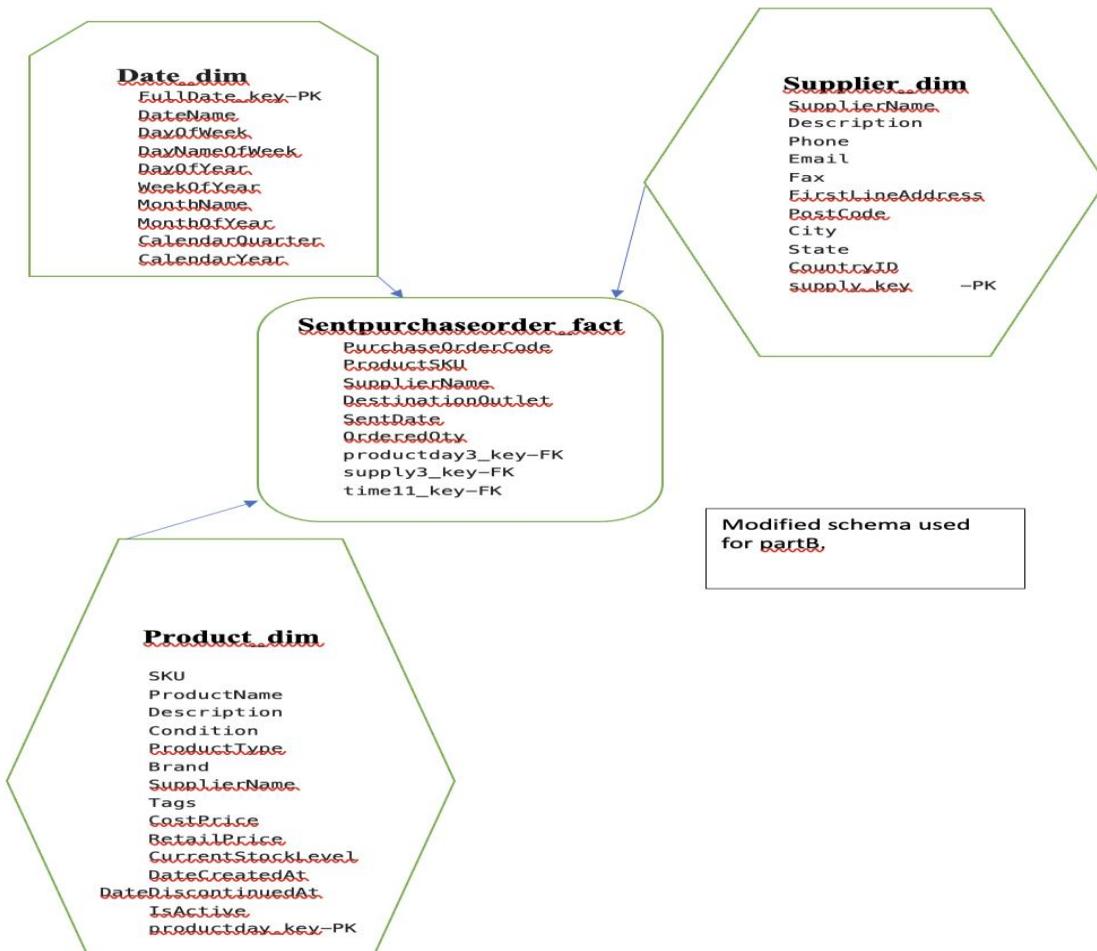
One of the main objectives of the coursework is inventory management which is broken down into 3 steps where purchase orders sent is managed well by checking ordered quantity, sent date, product and supplier detail for every purchase order sent. 2<sup>nd</sup> step is storing the received purchased orders appropriately and 3<sup>rd</sup> 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.



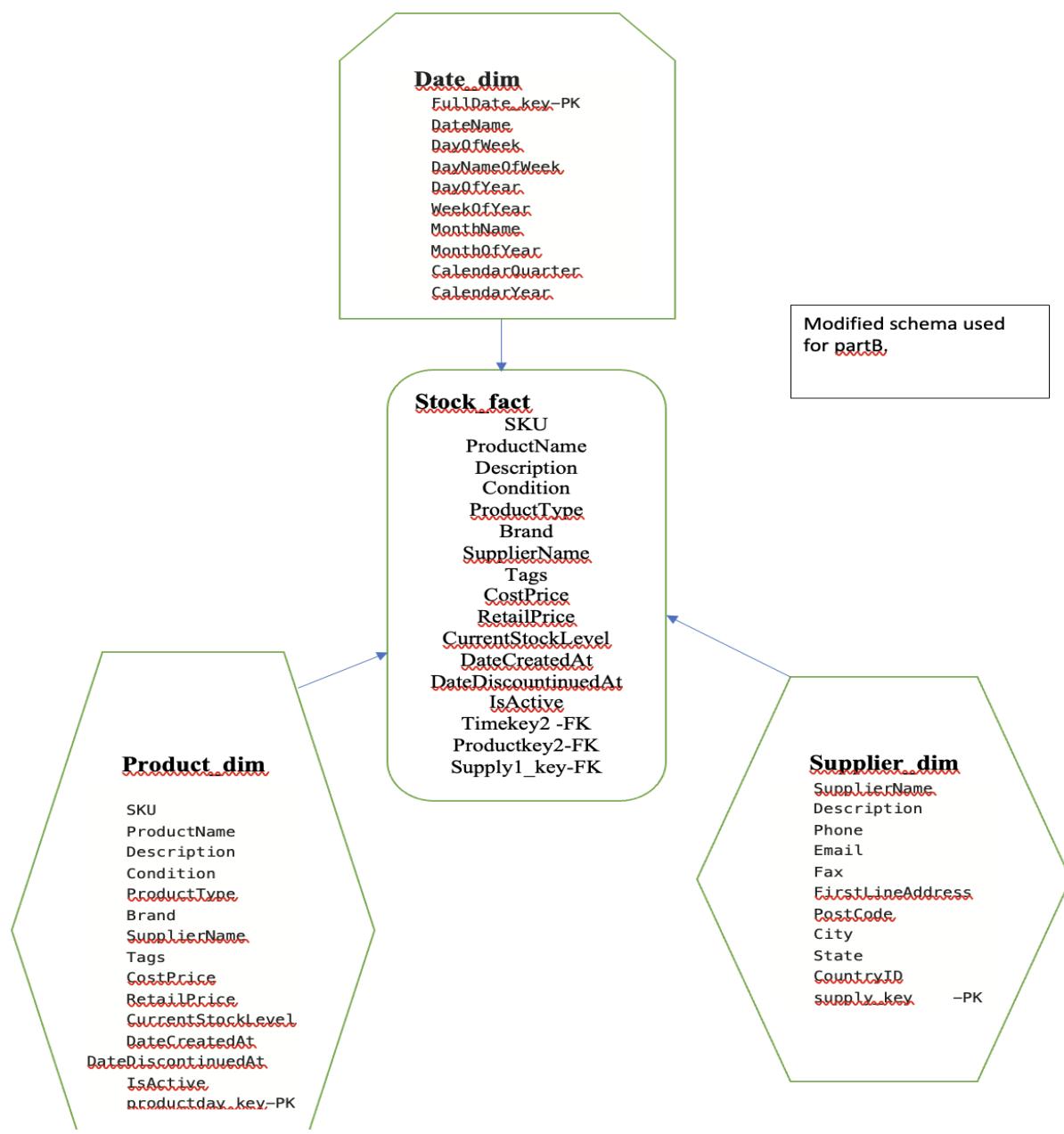
The above figure is the previous schema used for part A.



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.

The screenshot shows the database structure in SSMS. The 'partB' database is selected, and the 'dbo.SampleOfProductsDay1' table is highlighted. To the right, the script for creating the table is displayed.

```

1  CREATE TABLE [dbo].[SampleOfProductsDay1] (
2      [SKU]          NVARCHAR (50) NOT NULL,
3      [ProductName]  NVARCHAR (100) NOT NULL,
4      [Description] NVARCHAR (50) NOT NULL,
5      [Condition]   NVARCHAR (50) NOT NULL,
6      [ProductType] NVARCHAR (50) NOT NULL,
7      [Brand]        NVARCHAR (50) NOT NULL,
8      [SupplierName] NVARCHAR (50) NOT NULL,
9      [Tags]         NVARCHAR (50) NOT NULL,
10     [CostPrice]    FLOAT (53)  NOT NULL,
11     [RetailPrice]  FLOAT (53)  NOT NULL,
12     [CurrentStockLevel] TINYINT NOT NULL,
13     [DateCreatedAt] DATE    NOT NULL,
14     [DateDiscontinuedAt] DATE   NOT NULL,
15     [IsActive]     TINYINT NOT NULL,
16     [productday_key] INT     IDENTITY (1, 1) NOT NULL,
17     CONSTRAINT [productday_key] PRIMARY KEY CLUSTERED ([productday_key] ASC)
18  );

```

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

The screenshot shows the Azure Data Studio interface. At the top, there is a code editor window with the following SQL script:

```

1 create DATABASE partb
2
3 Create Table ProductdayDim(ProductKey INT, SKU STRING, ProductName STRING, Description STRING, Condition STRING, ProductType STRING, Brand STRING, SupplierName STRING, Tags STRING, CostPrice FLOAT, RetailPrice FLOAT)
4 LOAD DATA INPATH '/tmp/BigData_Coursework/ProductDim.csv' OVERWRITE INTO TABLE BigData_CW.ProductDim
5 LOAD DATA INPATH '/tmp/bigdata/product_dim.csv' OVERWRITE INTO TABLE ProductdayDim
6 SELECT * FROM ProductdayDim

```

Below the code editor are several buttons: EXECUTE, SAVE AS, VISUAL EXPLAIN, Show Results (which is checked), and Download Results. The results tab is selected, showing a table of data with the following columns: PRODUCTDAYDIM.PRODUCTKEY, PRODUCTDAYDIM.SKU, PRODUCTDAYDIM.PRODUCTNAME, PRODUCTDAYDIM.DESCRIPTION, PRODUCTDAYDIM.CONDITION, PRODUCTDAYDIM.PRODUCTTYPE, PRODUCTDAYDIM.BRAND, PRODUCTDAYDIM.SUPPLIERNAME, and PRC. The data includes entries for Manfrotto products like MN1004 and MT057C, a JVC portable recorder, and various Hoya filters.

PRODUCTDAYDIM.PRODUCTKEY	PRODUCTDAYDIM.SKU	PRODUCTDAYDIM.PRODUCTNAME	PRODUCTDAYDIM.DESCRIPTION	PRODUCTDAYDIM.CONDITION	PRODUCTDAYDIM.PRODUCTTYPE	PRODUCTDAYDIM.BRAND	PRODUCTDAYDIM.SUPPLIERNAME	PRC
null	ProductName	Description	Condition	ProductType	Brand	SupplierName	Tags	Cos
null	"Manfrotto MN1004 BAC Master Light St and"	Master Light Stand	Display	ACCESSORY	SENNHEISER	SENNHEISER	TRIPODS	57.9
null	"Manfrotto MT057C 3 Carbon Fibre 3 Section Geared "	Carbon Fibre 3 Section Geare d	Display	ACCESSORY	SONY	SONY	TRIPODS	298.0
null	"Rycote 37705 Portable Recorder Suspension "	Portable Recorder Suspension	New	CAMCORDER	JVC	JVC	CAMACC	0
null	"Hoya 37S-HOY 37 MM SKYLIGHT FILTER Hoya"	37MM SKYLIGHT FILTER Hoy a	New	IMAGING	SENNHEISER	SENNHEISER	"Discontinued	FILT
null	"HOYA 40.5mm CP Filter - Slim "	HOYA 40.5mm CP Filter - Slim	New	IMAGING	TOSHIBA	TOSHIBA	FILTERS	17.0

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 its 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 a mac laptop and everything in this coursework was Microsoft related, after completing this coursework it made me realize coding in mac is more fun.