

Database Development

using ssms and tableau

Table of Contents

[1 Introduction 2](#_Toc90378563)

[2 Importing Data 2](#_Toc90378564)

[2.1 de-normalized Data 3](#_Toc90378565)

[3 ETL Processing 4](#_Toc90378566)

[3.1 Adding missing Attributes 4](#_Toc90378567)

[3.2 Data merging and Scrubbing 4](#_Toc90378568)

[3.3 FInal\_Merged table: 6](#_Toc90378569)

[3.4 defining dImensions And fact table 6](#_Toc90378570)

[4 Star Schema 9](#_Toc90378571)

[5 Critical Analysis 10](#_Toc90378572)

[6 Query Section 10](#_Toc90378573)

[6.1 Part A 10](#_Toc90378574)

[6.1.1 Part a visualization in Tableau 13](#_Toc90378575)

[6.2 Part b 13](#_Toc90378576)

[6.2.1 Part b visualization 15](#_Toc90378577)

[6.3 Part c 15](#_Toc90378578)

[6.3.1 Part c Visualization 17](#_Toc90378579)

[6.4 part D 18](#_Toc90378580)

[6.4.1 Visualization of part d 19](#_Toc90378581)

[7 References 20](#_Toc90378582)

# Introduction

The intention of this report is to perform the database development using *dimensional modeling* also known as Kimball modeling technique to design a prototype star schema. The data which is provided belongs to the business, innovation, and skills department from April 2015 till January 2016. Here the tools I am using for the development of my database prototype are SQL Server Management Studio 18 (SSMS) and for designing the dashboard I will use tableau.

To begin with, section 1, includes the ETL *Extract, Transform and Load* of the data into SSMS and the star schema stating the fact and dimension tables for the database will be designed. Section 2 includes the transformation of the design into query implementations. Section 3 will be based on the database dashboard designing of the database using tableau.

# Importing Data

The database name is *data management*.

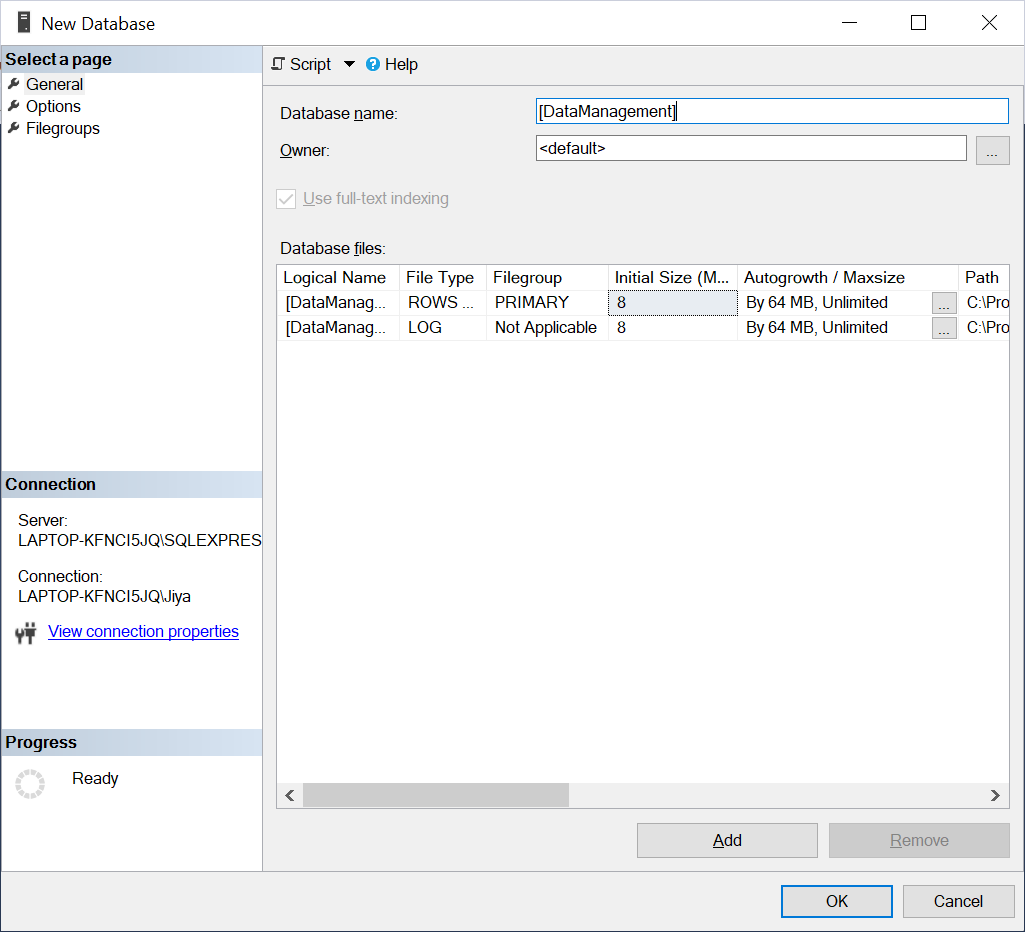


Figure 1: Creating Database

Here the next step is to import all 12 data files as a flat-file .csv and after importing data extract transform and load process will be done on it by changing the datatype and allowing null for all attributes as the data file contains empty rows. All files are imported in the same way shown below in *figure 2*.

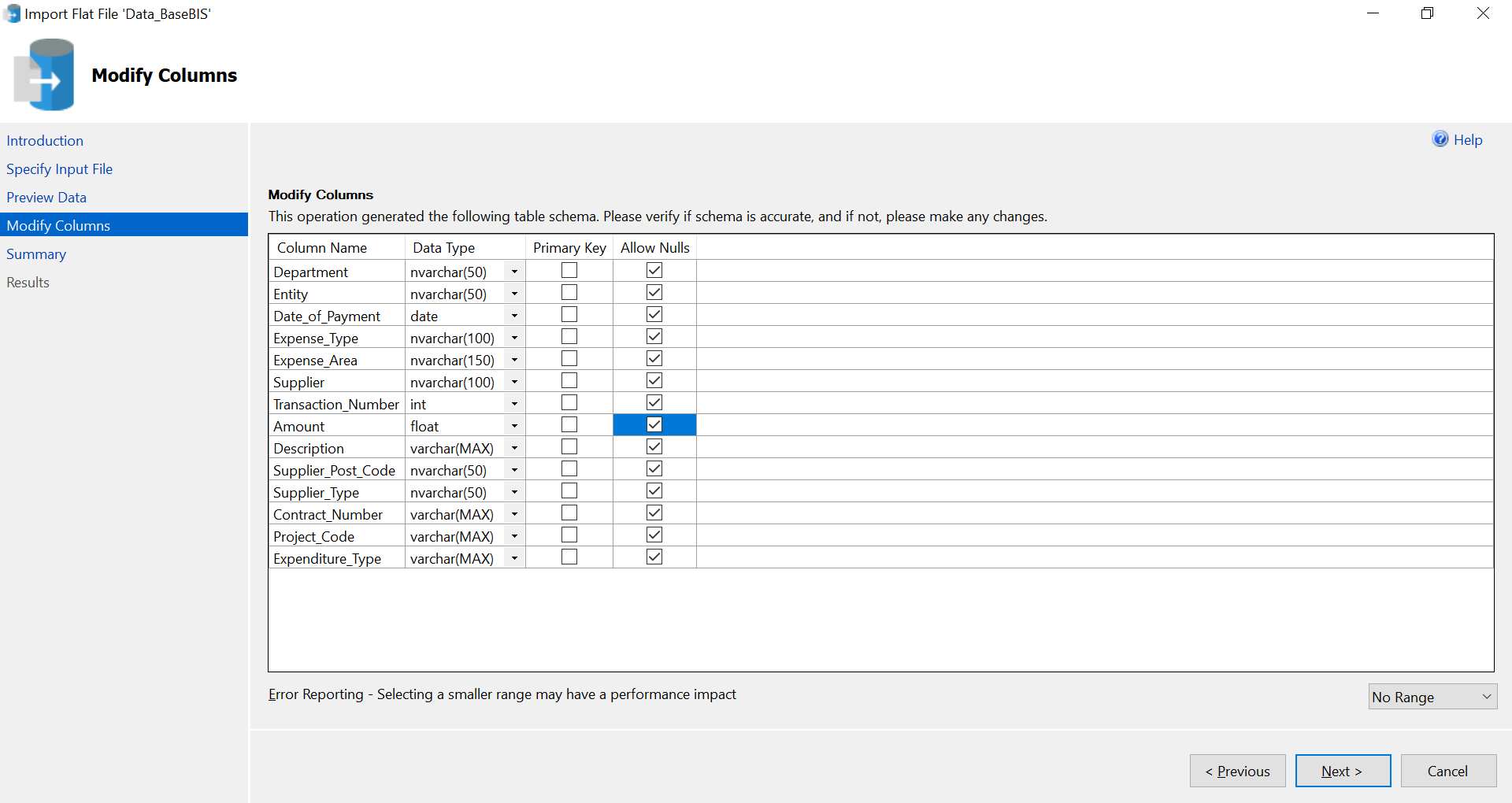


Figure 2: Importing flat files .csv

## de-normalized Data

Here you can see that the table is in de-normalized form and needs to be arranged understandably.

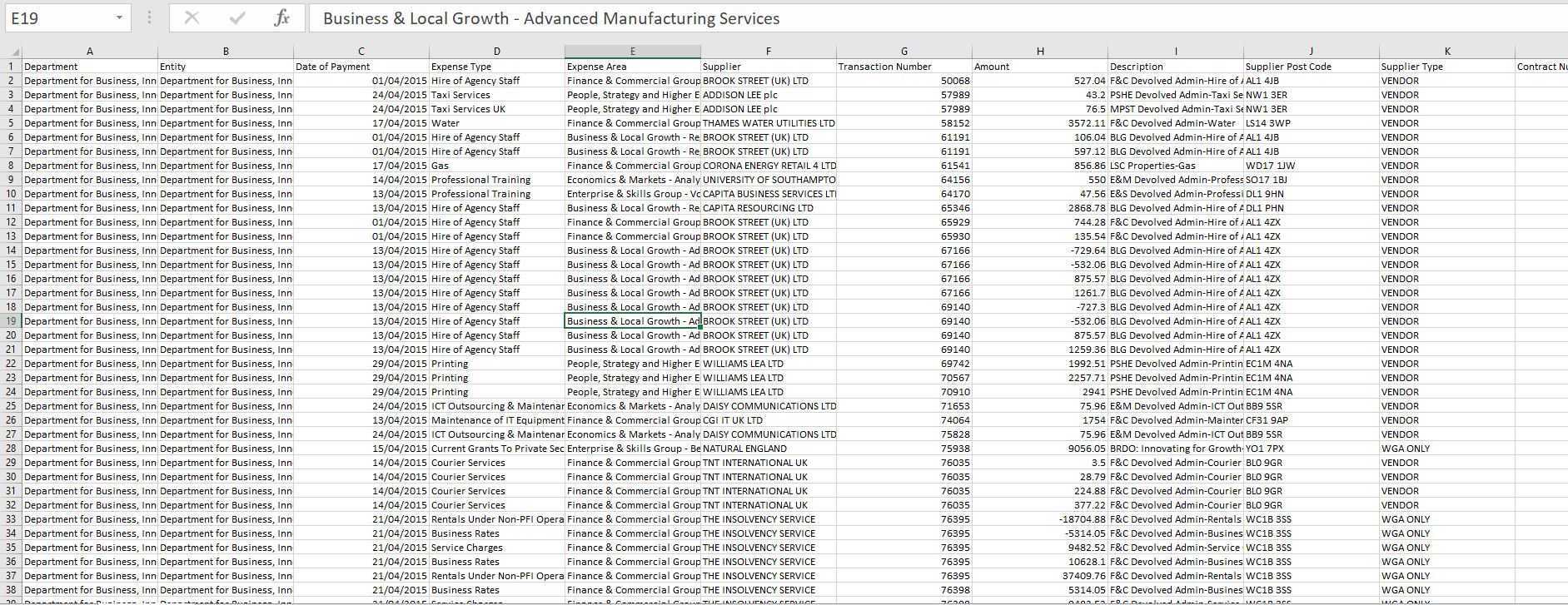


Figure 3: CSV file data

# ETL Processing

*Extraction* is the initial process where we insert data into the data warehouse, extracting means understanding the data and copying the data in ETL. After the data has been extracted now the various *transformation* needs to be done like cleaning nulls, changing data types, and de-duplication of the data if it is needed. The last step involves loading data into a dimensional model [2].

## Adding missing Attributes

In the file of august 2015 and March 2016 some attributes which I have added by using ALTER TABLE query.

ALTER TABLE August2015 ADD [Project\_Code] varchar(MAX), [Expenditure\_Type] varchar(MAX);

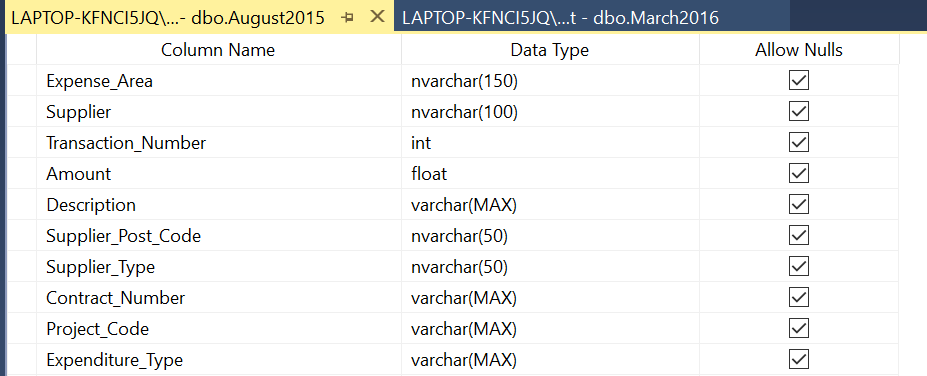


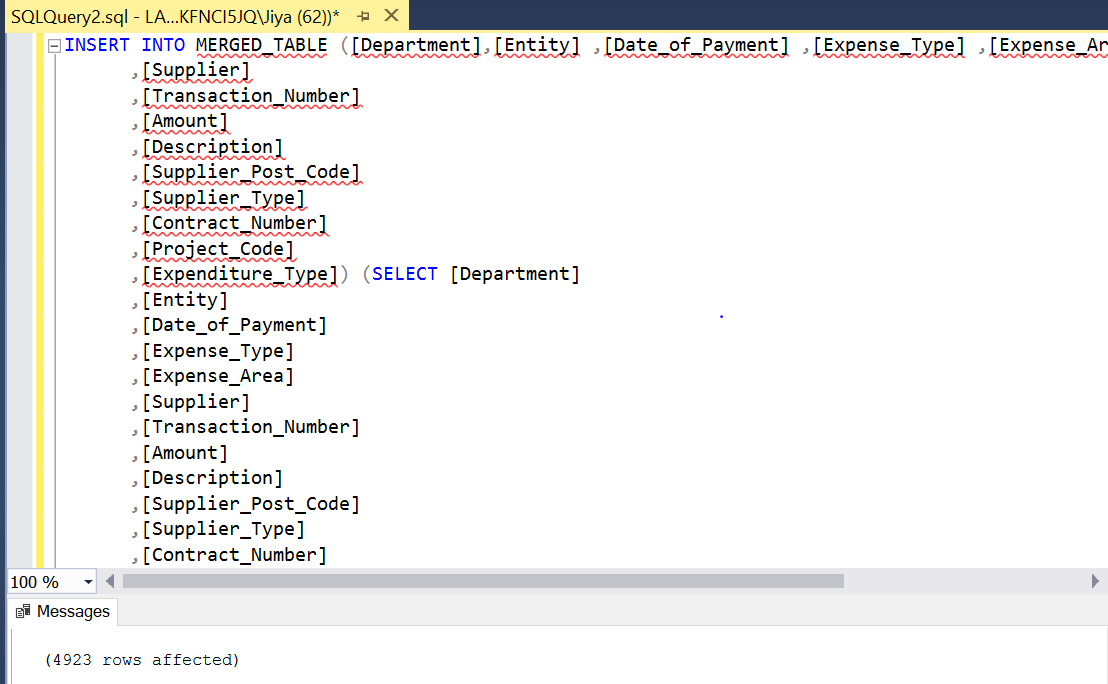
Figure : adding missing attributes

## Data merging and Scrubbing

Merging all 12 files in a single table and filtering the data to develop a dimensional model. Query for merging table I used is:

INSERT INTO MERGED\_TABLE ([Department],[Entity] ,[Date\_of\_Payment] ,[Expense\_Type] ,[Expense\_Area] ,[Supplier] ,[Transaction\_Number] ,[Amount] ,[Description] ,[Supplier\_Post\_Code] ,[Supplier\_Type] ,[Contract\_Number] ,[Project\_Code] ,[Expenditure\_Type])

(SELECT [Department] ,[Entity] ,[Date\_of\_Payment],[Expense\_Type] ,[Expense\_Area] ,[Supplier] ,[Transaction\_Number] ,[Amount] ,[Description] ,[Supplier\_Post\_Code] ,[Supplier\_Type] ,[Contract\_Number] ,[Project\_Code] ,[Expenditure\_Type] FROM November2015)



The next, the step is to clean and filter the data as per-required.

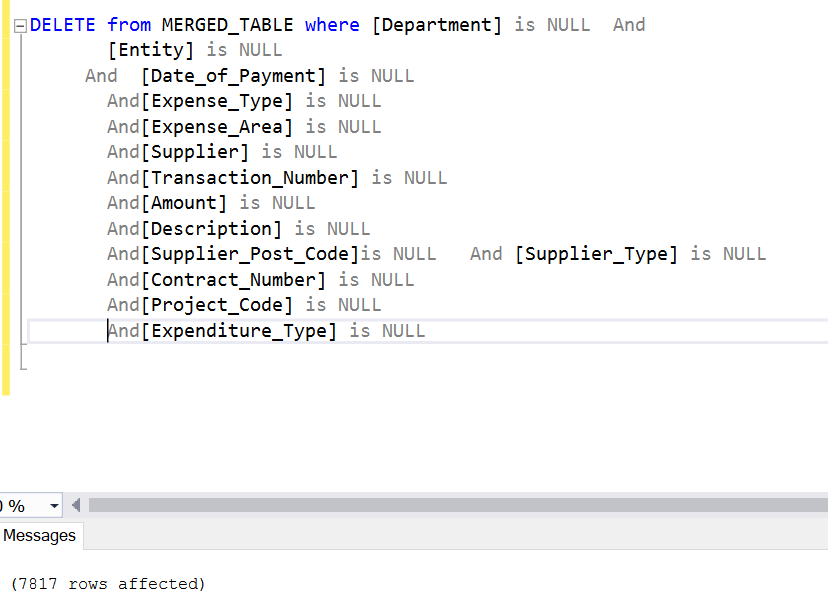


Figure : rows cleaned

I have cleaned the rows which have entirely null values created by the database it-self. As we can see in *figure5* I haven’t deleted any record I have just removed the rows which are Null throughout the attributes.

## FInal\_Merged table:

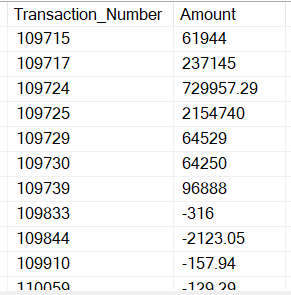
The final table as required is using the data on invoices for spending over 25000£ including the refunds that are lesser than -50 £ so I have selected that data and by using the Union all statement I have merged my filter table.

SELECT \* into Final\_Merge FROM (

SELECT \* FROM MERGED\_TABLE where Amount >25000 UNION all

SELECT \* FROM MERGED\_TABLE where Amount <= -50)

as FINALRESULT;



## defining dImensions And fact table

For the star schema I have decided on the following dimension tables:

1. Supplier details
2. Transaction details
3. Null attribute / Contract information

We will create the dimension table one by one. For all the dimension I will be creating auto generating key by keeping the identity (1,1) which means auto increment true.

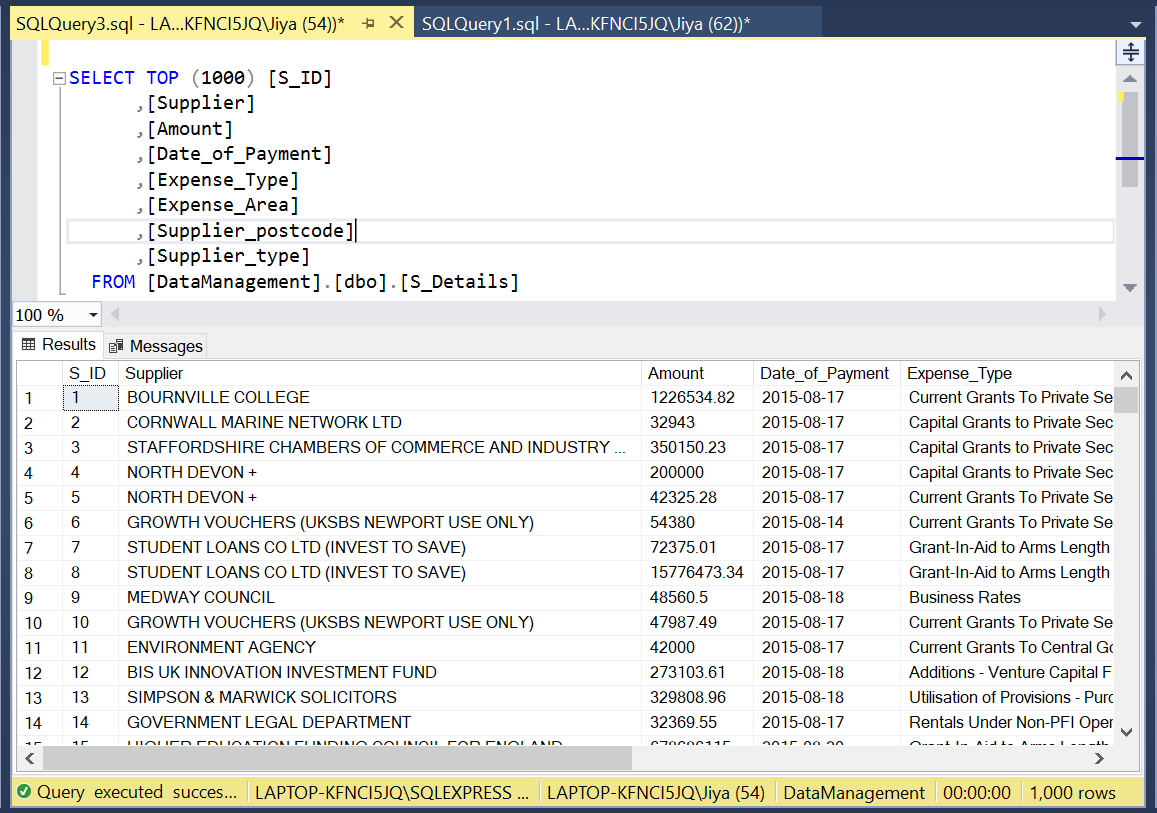


Figure : dimension table 1: Suppliers details

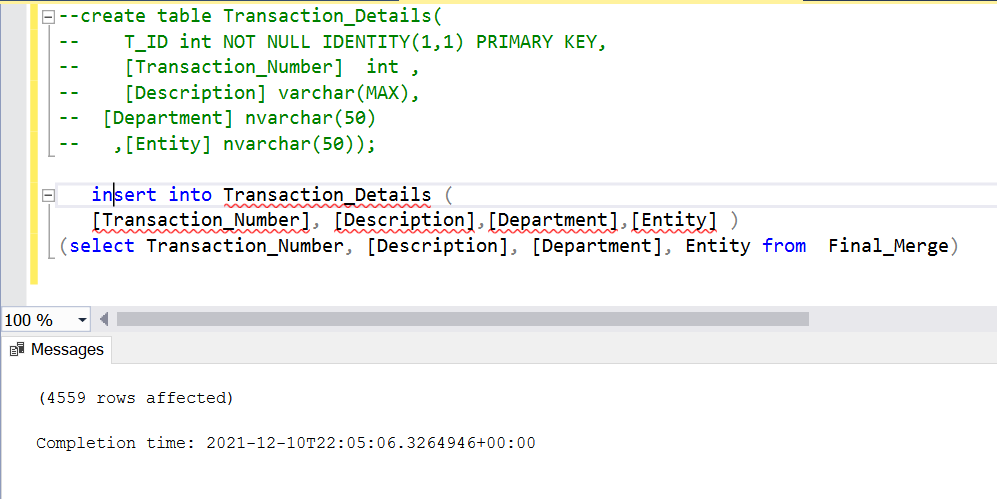


Figure 7: dimension 2: Transactions and departmental details

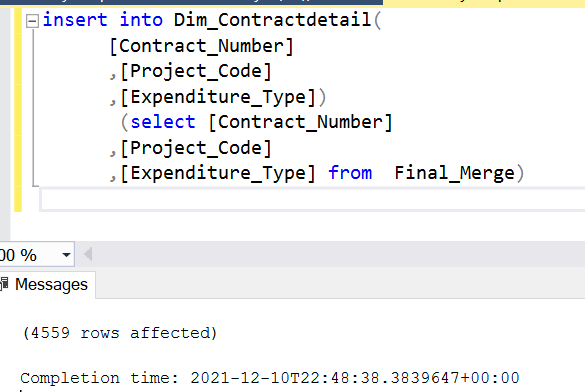
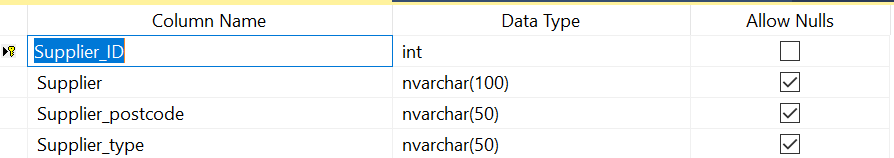


Figure 8: Dimension 3 : Null attributes for future use



For all dimensions primary key is given which is allotted as not null.

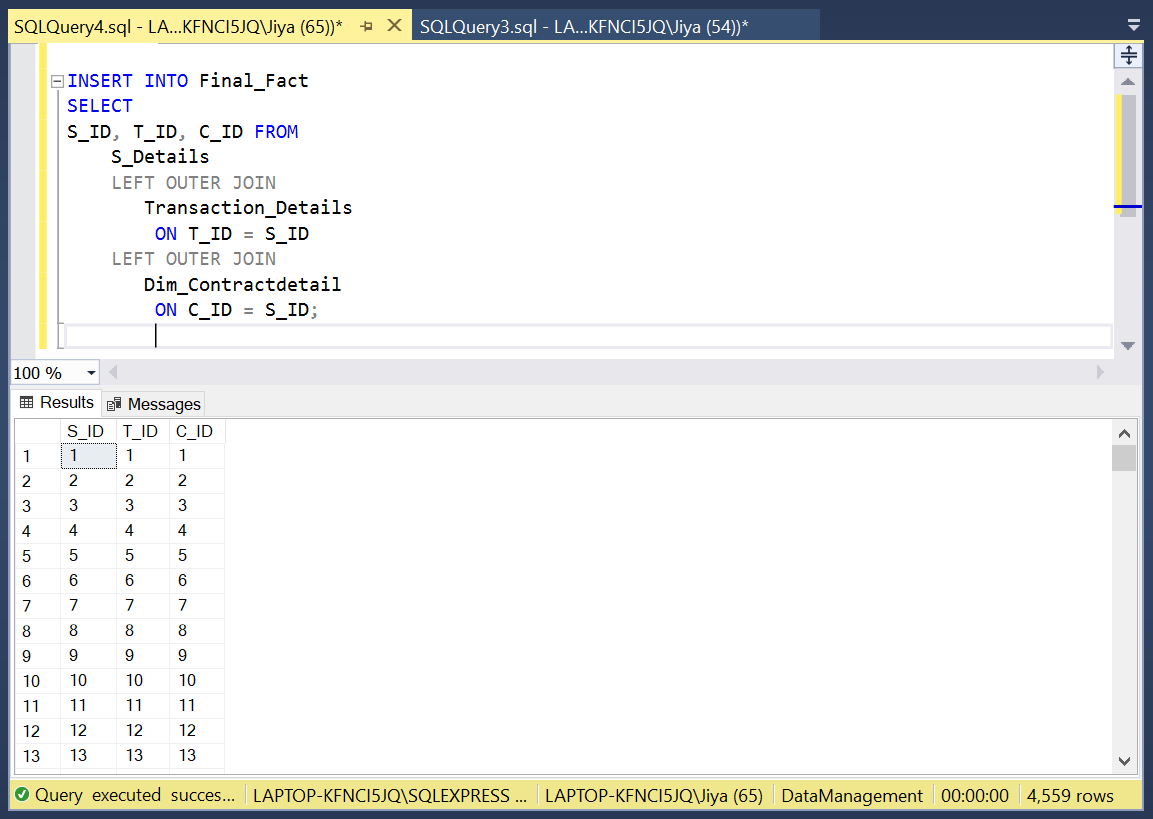
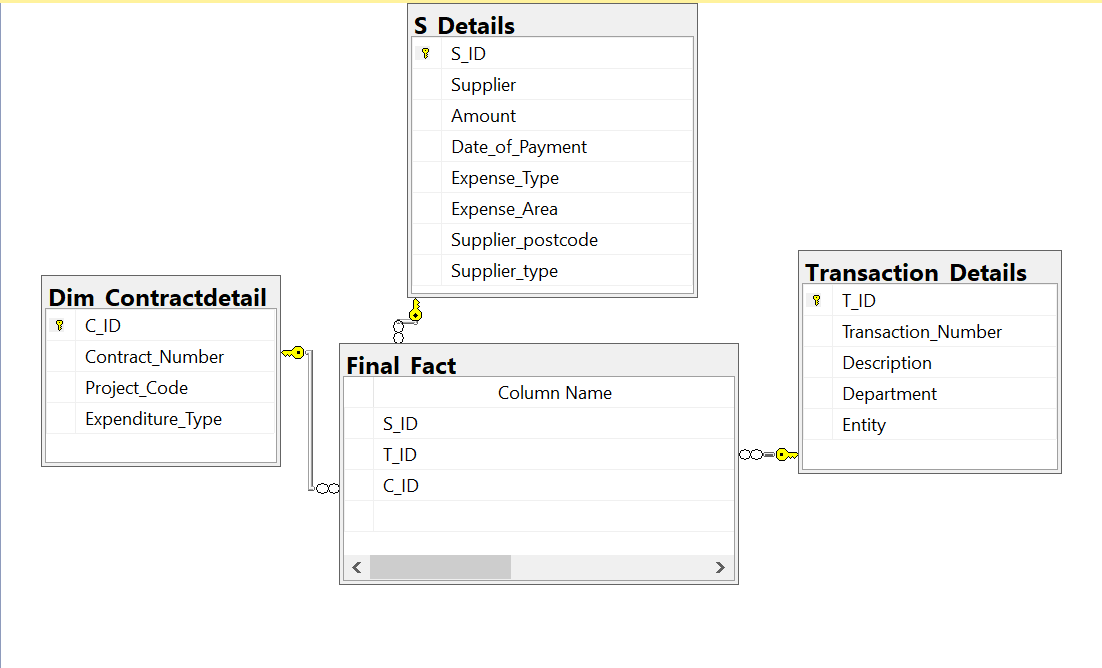


Figure : Fact table loading data

# Star Schema

I have designed my star schema in SSMS using the database diagram menu. I have dimensions connected to one fact table. Each entry in fact table is the foreign key for the dimensions it is connected to.



Primary Keys

Figure : Star Schema

# Critical Analysis

* There were many problem arises while importing the data so I have to change some of trouble some attributes to varchar (MAX).
* After I made my schema, I am unable to open that diagram again in SQL server as it kept closing the server. This issue is still unresolved.
* Some times when I run the Queries it didn’t show me the result while when A restart the SQL the same queries runs perfectly.
* For my previous star schema that was really great but I was unable to apply queries on it so I reject this schema and go with the simple one.

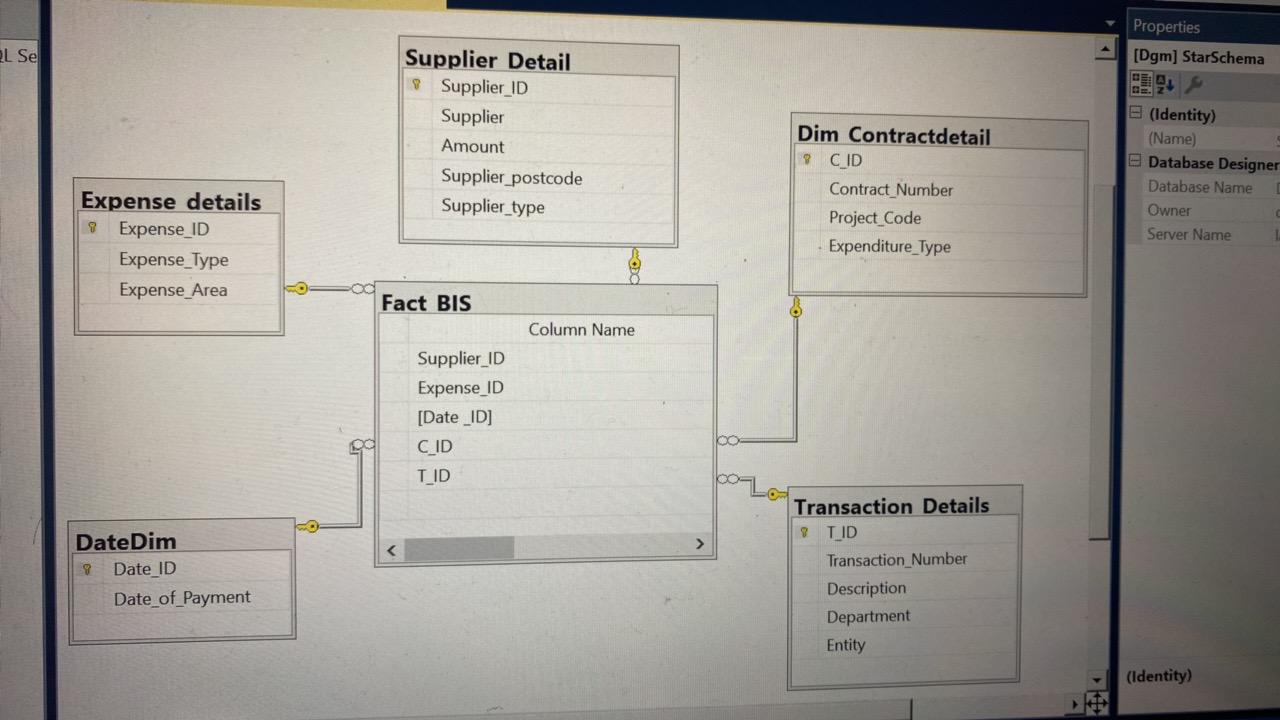


Figure : rejected Schema

# Query Section

In this section we will try to develop the four queries in SSMS and test it and build the dash board visualization in tableau.

## Part A

So, in the part a of this section we are asked to provide the top 3 supplier and the amount spent on those suppliers on yearly bases and on every quarter of the year that are April - June , July- September , October – December and January – March.

Graphical user interface, application, Word

Description automatically generated

Figure : temporary tables

*Query for part a:*

* For year

select top 3 Supplier, sum(Amount) AS finalsum, 'Year' AS 'Year' into yearly

from [DataManagement].[dbo].[S\_Details]

group by Supplier order by finalsum desc;

--For quarter 1 which is April to June

select top 3 Supplier, sum(Amount) AS Firstquarter, 'quarter1' AS 'Year' into quarter1

from [DataManagement].[dbo].[S\_Details]

where Date\_of\_Payment between '2015-04-01' and '2015-06-30'

group by Supplier order by Firstquarter desc;

--for Quarter 2 july -sep

select top 3 Supplier, sum(Amount) AS Secondquarter, 'quarter2' AS 'Year' into quarter2

from [DataManagement].[dbo].[S\_Details]

where Date\_of\_Payment between '2015-07-01' and '2015-09-30'

group by Supplier order by Secondquarter desc;

--for Quarter 3 Oct - dec

select top 3 Supplier, sum(Amount) AS ThirdQuarter, 'quarter3' AS 'Year' into quarter3

from [DataManagement].[dbo].[S\_Details]

where Date\_of\_Payment between '2015-10-01' and '2015-12-31'

group by Supplier order by ThirdQuarter desc;

--for Quarter 4 Jan - March

select top 3 Supplier, sum(Amount) AS fourthQuarter, 'quarter4' AS 'Year' into quarter4

from [DataManagement].[dbo].[S\_Details]

where Date\_of\_Payment between '2016-01-01' and '2016-03-31'

group by Supplier order by fourthQuarter desc;

--Converting separate queries into 1 using Union all

Select \* from yearly union all

Select \* from quarter1 union all

Select \* from quarter2 union all

Select \* from quarter3 union all

Select \* from quarter4

In the above section of query I have used the group by statement to list out the top 3 suppliers on the bases of sum of their amount spent from the dimension S\_details. Between clause is used to specify the year and each quarter in descending order I have stored the results in temporary tables i.e. yearly , quarter1, quarter2 , quarter3 , quarter4 to join the queries.

Graphical user interface, text, application

Description automatically generated with medium confidence

Jan-March

Oct-Dec

July-Sep

April-June

2015-2016

Figure : 1st query results

### Part a visualization in Tableau

Waterfall chart

Description automatically generated with medium confidence

Figure : Query1 visualization

In the above figure the heatmap is showing the final sum of top 3 suppliers for year and for each quarter. Like in quarter 1 the top3 suppliers are higher education , skill funds and student loan company similarly the different top suppliers for other quarters and year are shown.

## Part b

Display the result for expense type who have greater expenses greater than the average amount for both half of the year from April to September and from October to March.

*Query for part b:*

--calculating the expense type from April to September first half

select Expense\_Type, sum(Amount) AS half\_year1 ,'April-sep' AS 'time' into t1

from S\_Details

where Date\_of\_Payment between '2015-04-01' and '2015-09-30'

group by Expense\_Type;

select \* from t1

where half\_year1 > ( select AVG(amount) from s\_details where Date\_of\_Payment between '2015-04-01' and '2015-09-30')

--calculating the expense type October to March second half

select Expense\_Type, sum(Amount) AS half\_year2 ,'Oct-Mar' AS 'time' into t2

from S\_Details

where Date\_of\_Payment between '2015-10-01' and '2016-03-31'

group by Expense\_Type;

select \* from t2

where half\_year2 > (select AVG(amount) from s\_details where Date\_of\_Payment between '2015-10-01' and '2016-03-31' )

--Joining the queries

Select \* from t1

Union all Select \* from t2

In the above section of the query we have to display those expense area which have the 6 month amount greater than the average amount of those 6 months same is the case for next 6 months. Storing the results in temporary tables t1 and t2.

Graphical user interface, text, application

Description automatically generated

1st half-year

Little bit is shown in SS

)

2nd half-year

Little bit is shown in SS

)

Figure : 2nd Query Results

### Part b visualization

Graphical user interface, application

Description automatically generated

Figure : Sum of Half-year

Here in the above graph the sum of both half year which is from April2015 to September 2015 and October 2015 to March 2016 has been shown on the bases of expense type. The next graph shows the top 10 expense area with the details of sum of their half-year.

Table

Description automatically generated with low confidence

Figure : detail graph

## Part c

Displaying the monthly top 10 ranking for spend on Expense Areas. For each Expense area indicate the number of places it has moved (up/down the rankings) since the previous month. The ranking must be in ascending order.

I managed to get the part 1 of this query right but I tried a lot but didn’t find the solution to show the up or down ranking of the previous month.

*Half Query for part c*

WITH CTE AS (

select SUM(Amount) AS sum\_amount, Expense\_Area, DATENAME(month,Date\_of\_Payment) AS monthsname,

YEAR(Date\_of\_Payment) as yearsname,

RANK() OVER (

PARTITION BY YEAR(Date\_of\_Payment), DATENAME(month,Date\_of\_Payment)

ORDER BY YEAR(Date\_of\_Payment),

DATENAME(month,Date\_of\_Payment), SUM(Amount) DESC) order\_rank

from

[dbo].[S\_Details]

WHERE Expense\_Area IS NOT NULL

group by DATENAME(month,Date\_of\_Payment), Year(Date\_of\_Payment), Expense\_Area

)

--temporary table generated top expense area

select \* into Top\_Expensearea

from CTE

where order\_rank <= 10

select \* from Top\_Expensearea order by monthsname asc

with CTE is basically a recursive type of statement in the SQl serve it is used to reduce the amount of sub queries written in it here we are using it to display the top ranked order by their month extracting or partitioning the date\_of\_payment with data type date into months and date respectively. Where CTE is (command table Expression) for temporary storing the results in references with other statements unlike the union and joins statement. CTE always returns a result set. It provides a easy way to deal with complex Queries , subqueries, or multiple joins [1]. This was the most difficult part of the course work for me.

Graphical user interface, text, application

Description automatically generated

Top 10 expense area for each month arranged in ascending order

Figure : 3rd query result : TOP 10 expense areas for each MONTH

### Part c Visualization

Chart

Description automatically generated with medium confidence

Figure : Expense Area top 10 ranking with labels

Graphical user interface, chart

Description automatically generated

Figure : without labels

The above graphs show the expense area for each month on the bases of top ten rankings and the sum or amount they have spend each month. Like here April 2015 1st rank is above 5 billion.

## part D

The last section we were asked to develop a complex query totally based on time hierarchy on our own. So,I decided to target the supplier type and calculate the top 3 suppliers for each year that is 2015 and 2016.

*Query for part d:*

Select top 3 Supplier\_type , sum(Amount) as TopSuppliers, '2015' AS 'Year' into year2015

from [DataManagement].[dbo].[S\_Details]

where Date\_of\_Payment between '2015-04-01' and '2015-12-31'

group by Supplier\_type order by TopSuppliers desc;

Select top 3 Supplier\_type , sum(Amount) as TopSuppliers, '2016' AS 'Year' into year2016

from [DataManagement].[dbo].[S\_Details]

where Date\_of\_Payment between '2016-01-01' and '2016-03-31'

group by Supplier\_type order by TopSuppliers desc;

Select \* from year2015 union all

Select\* from year2016;

Graphical user interface, text, application

Description automatically generated

Figure : 4th query and result

### Visualization of part d

Graphical user interface, application

Description automatically generated

The column shows the top 3 supplier type for each year which are grant, vendor and WGA only for both years. WGA only have the greater amount through both years.s

# References

Anon., n.d. *w3resource.* [Online]   
Available at: https://www.w3resource.com/mysql/mysql-union.php

Anon., n.d. *w3schools.* [Online]   
Available at: https://www.w3schools.com/sql/sql\_groupby.asp

raturi, p., n.d. *SQL skull.* [Online]   
Available at: https://sqlskull.com/2020/11/21/how-to-get-top-n-records-for-each-category/  
[Accessed tuesday December 2021].

Ross, R. K. M., 2013. *The data warehouse toolkit.* 3rd ed. United states of America: Wiley.