

Data Management

Report On Coursework 2



Submitted By

Student ID: 4126927

Data Import – The appropriate star schema was designed using Ralph Kimball's Dimension Modelling Technique as specified in the coursework. The purpose of this model is to improve the database performance and efficiency by reducing query processing time and improving results. In order to import the data, I am using a CSV file provided in the module and following the specific steps for importing it into SSIS Tool in Visual Studio.

The Figure below shows how I merged the 12 csv files into one database

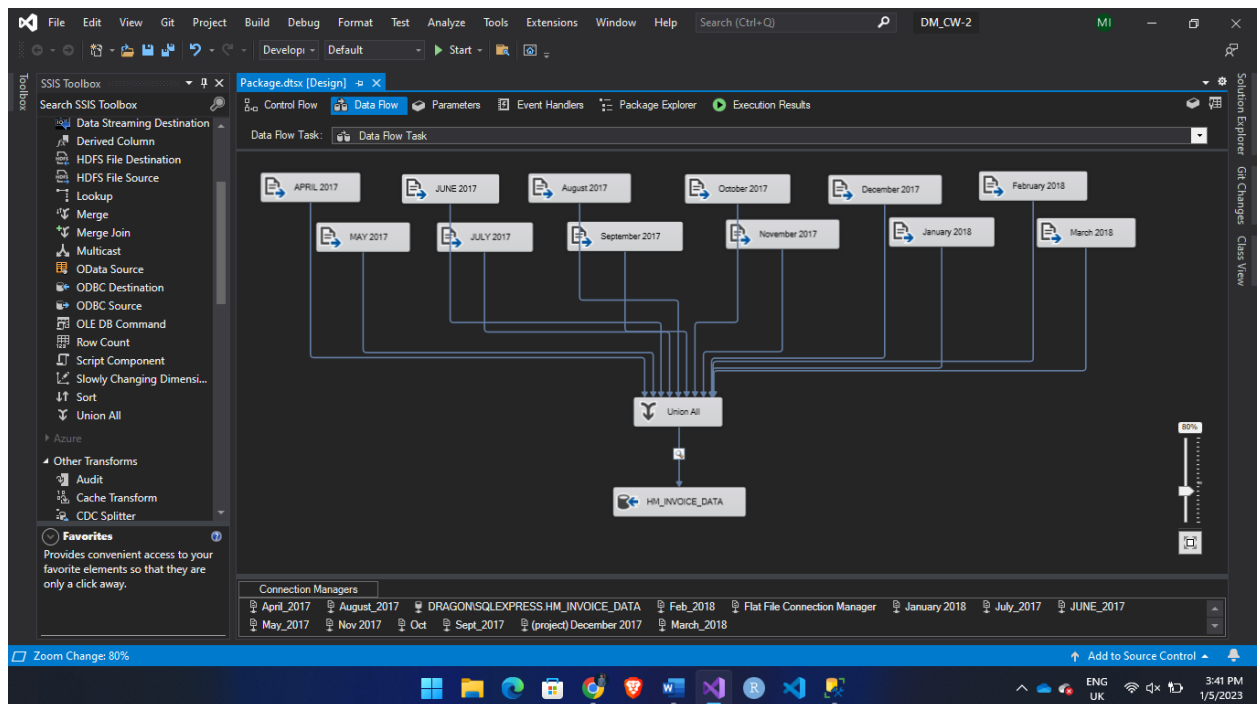


Figure :1 CSV File Import

1. First, I used Flat File Source to connect csv file
2. Then used union all to merge 12 csv file into one data file
3. Then stored the in a database using OLE DB Destination

```

SQLQuery1.sql - not connected  Ansewer_4_b.sql -...D

CREATE TABLE [HM_INVOICE_TABLE] (
    [Department] varchar(500),
    [Entity] varchar(500),
    [Date] DATETIME,
    [ExpenseType] varchar(500),
    [ExpenseArea] varchar(500),
    [SupplierName] varchar(500),
    [TransactionNumber] FLOAT,
    [Spend] FLOAT
)

```

Figure 2: SQL Query in SSMS Tool

Figure 2 contains the query used for creating Table to store out data from csv file in OLE DB Destination.

SSMS Tools was used to perform queries in this project.

SQLQuery1.sql - D...RAGON\nahid (52)* Ansewer_4_b.sql -...DRAGON\nahid (59)) DM_CW_2_QUERY.sql - not connected

```
SELECT *FROM HM_INVOICE_TABLE
```

121 %

Results Messages

	Department	Entity	Date	ExpenseType	ExpenseArea	SupplierName
1	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Grant	DBS - Disclosure & Barring Service	MAYORS OFFICE FOR POLICING AND CRIME
2	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Legal Services & Fees	DBS - Disclosure & Barring Service	BRISTOWS LLP
3	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Other Government Department	DBS - Disclosure & Barring Service	LIVERPOOL CITY COUNCIL
4	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Other Government Department	DBS - Disclosure & Barring Service	LIVERPOOL CITY COUNCIL
5	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Business Process Outsourcing	DBS - Disclosure & Barring Service	TATA CONSULTANCY SERVICES LTD
6	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Business Process Outsourcing	DBS - Disclosure & Barring Service	TATA CONSULTANCY SERVICES LTD
7	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Business Process Outsourcing	DBS - Disclosure & Barring Service	TATA CONSULTANCY SERVICES LTD
8	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Business Process Outsourcing	DBS - Disclosure & Barring Service	TATA CONSULTANCY SERVICES LTD
9	Home Office	DBS - Disclosure & Barring Service	2017-02-05 00:00:00.000	Business Process Outsourcing	DBS - Disclosure & Barring Service	TATA CONSULTANCY SERVICES LTD
10	Home Office	DBS - Disclosure & Barring Service	2017-03-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
11	Home Office	DBS - Disclosure & Barring Service	2017-05-05 00:00:00.000	Other Government Department	DBS - Disclosure & Barring Service	CROWN COMMERCIAL SERVICE
12	Home Office	DBS - Disclosure & Barring Service	2017-09-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
13	Home Office	DBS - Disclosure & Barring Service	2017-09-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
14	Home Office	DBS - Disclosure & Barring Service	2017-09-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
15	Home Office	DBS - Disclosure & Barring Service	2017-09-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
16	Home Office	DBS - Disclosure & Barring Service	2017-11-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
17	Home Office	DBS - Disclosure & Barring Service	2017-11-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
18	Home Office	DBS - Disclosure & Barring Service	2017-12-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE
19	Home Office	DBS - Disclosure & Barring Service	2017-12-05 00:00:00.000	Accounting Officer	DBS - Disclosure & Barring Service	HOME OFFICE

Query executed successfully. DRAGON\SQLEXPRESS (15.0 RTM) DRAGON\nahid (52) HM_INVOICE_DATA 00:00:00 14,407 rows

Figure 3: Merged Data Table From CSV (HM_INVOICE_TABLE)

In Figure 3 above we can see our data using SQL Query.

Data Cleaning

For cleaning the data I used SSIS Tool to perform ETL on our data. Below Figure shows how ETL was used to clean the null values from each column.

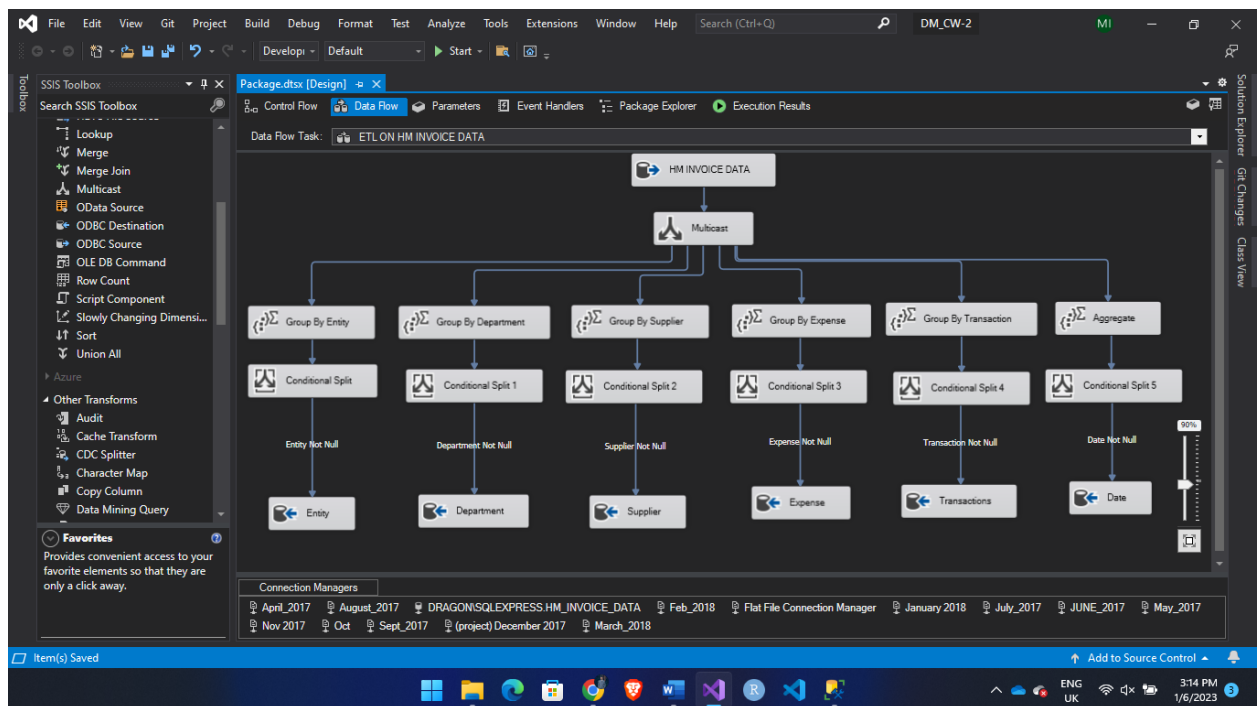


Figure 4: Data Cleaning in SSIS

In the VS SSIS Tool I used multicast to separate each table and used aggregate transformation to group each column. Figure 5 bellow displays some of the process.

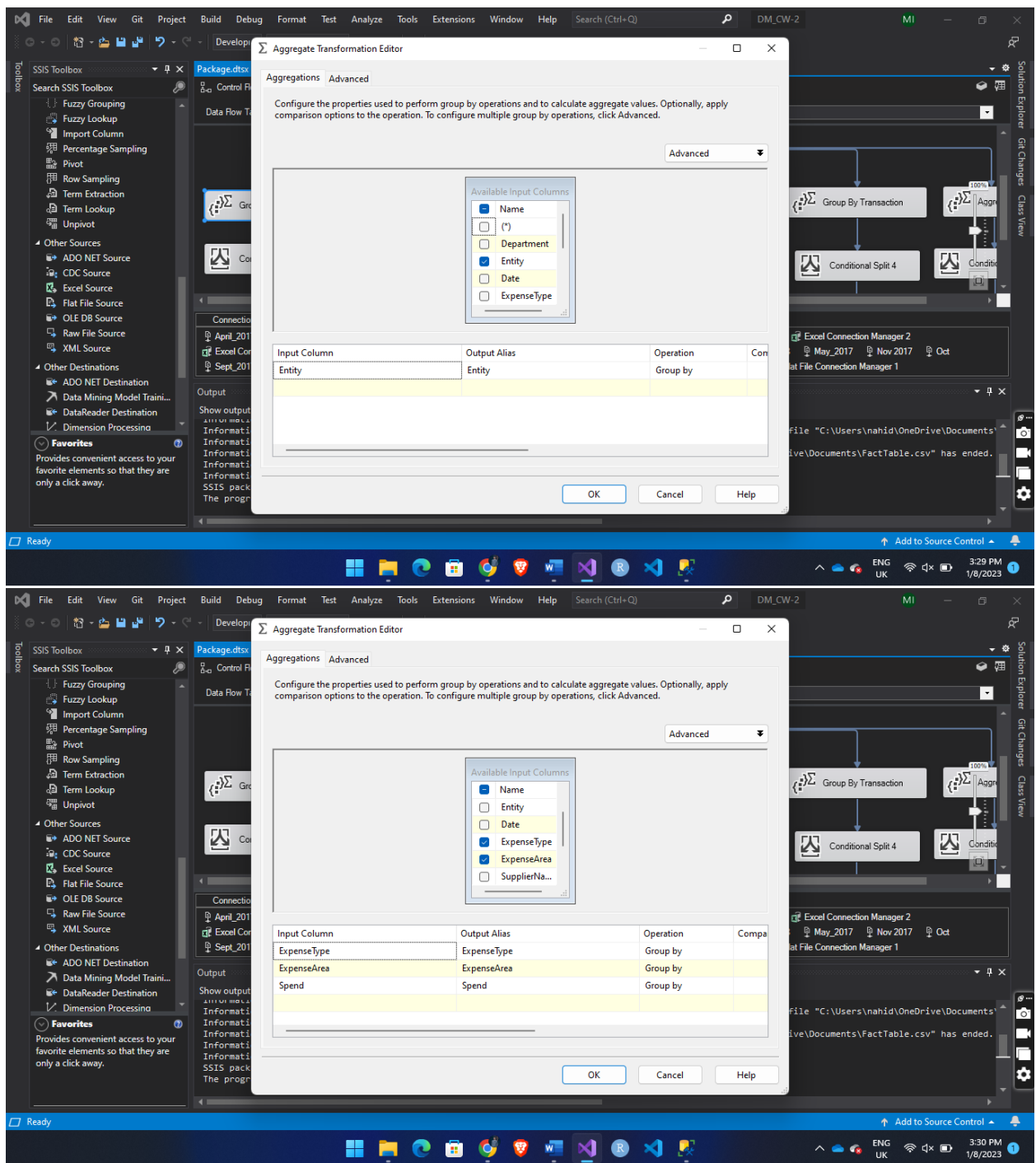


Figure 5: Grouping Each Table

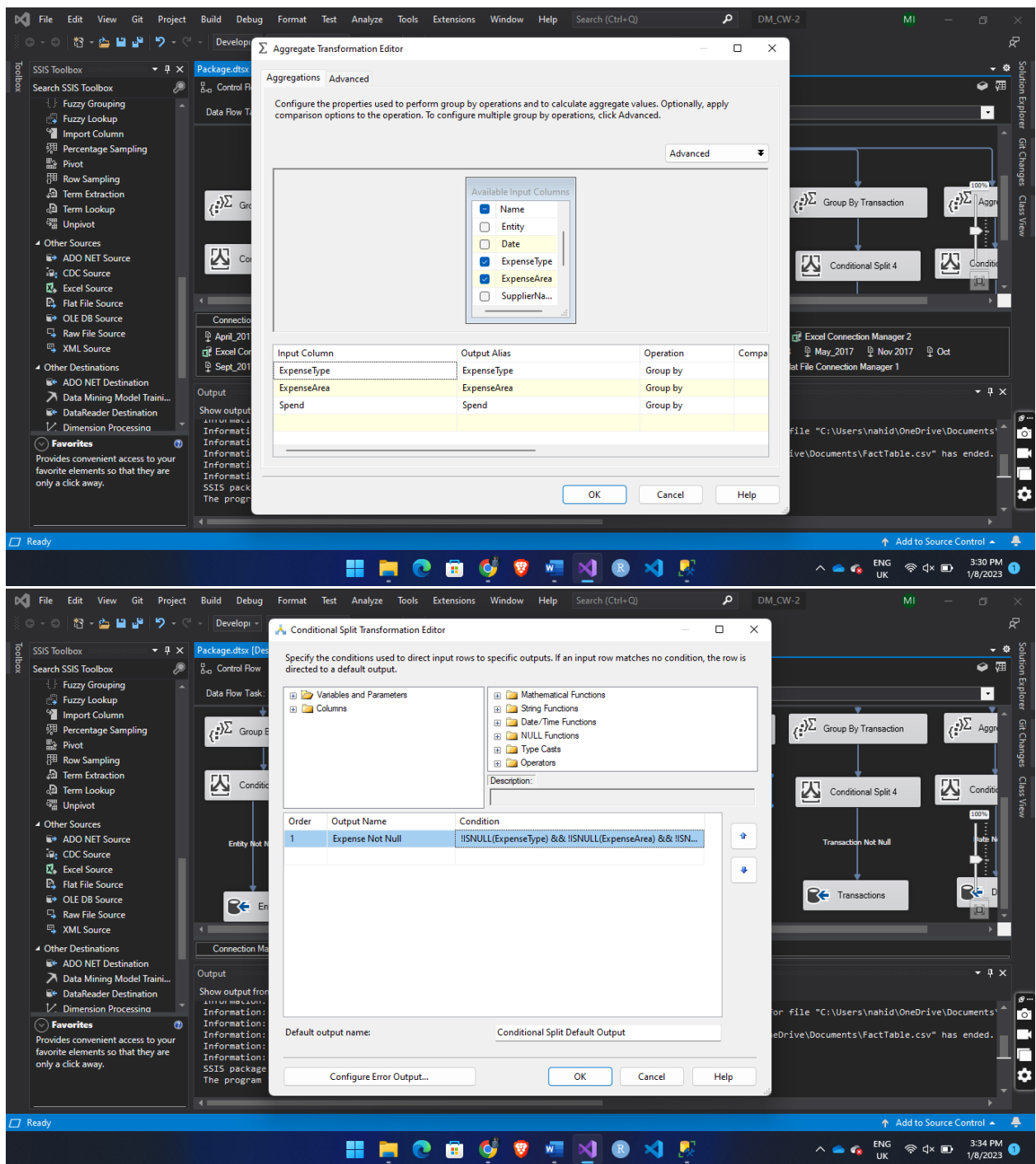


Figure 6: Cleaning Null values in SSIS

In figure 6 I used conditional split to remove null values from our data. The Following expression was used for cleaning null values

!ISNULL(Entity)

!ISNULL(Department)

!ISNULL(Supplier)

!ISNULL(Date)

!ISNULL(Transacion)

!ISNULL(ExpenseType) && !ISNULL(ExpenseArea) && !ISNULL(Spend)

Then the cleaned data was stored in 6 separate table which was created using SQL query

Below figures contains the SQL for creating **EntityTable** , **DepartmentTable** , **SupplierTable** , **DateTable** , **TransactionTable** , **ExpenseTable**

```
SQLQuery1.sql - D:\...RAGON\nahid (52))  DRAGON\SQLEXPRES...TA - StartSchema  Ansewer_4_b.sql - ...DRAGON\nahid (59))

CREATE TABLE [DateTable] (
    [Date_ID] int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Dates] DATETIME
)

CREATE TABLE [DepartmentTable] (
    [Department_ID] int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [DepartmentName] nvarchar(500)
)

CREATE TABLE [EntityTable] (
    [Entity_ID] int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Entity] nvarchar(500)
)

CREATE TABLE [ExpenseTable] (
    [Expense_ID] int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [ExpenseType] nvarchar(500),
    [ExpenseArea] nvarchar(500),
    [Spend] float
)
```



```

SQLQuery1.sql - D:\RAGON\nahid (52)  DRAGON\SQLXPRES...TA - StartSchema  Answer_4_b.sql - ...DRAGON\nahid (59)

CREATE TABLE [SupplierTable] (
    [Supplier_ID] int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [SupplierName] nvarchar(500)
)

CREATE TABLE [TransactionTable] (
    [Transaction_ID] int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [TransactionNumber] float
)

CREATE TABLE FactTable (
    Department_ID int FOREIGN KEY (Department_ID) REFERENCES DepartmentTable(Department_ID) ON UPDATE CASCADE,
    Expense_ID int FOREIGN KEY (Expense_ID) REFERENCES ExpenseTable(Expense_ID) ON UPDATE CASCADE,
    Transaction_ID int FOREIGN KEY (Transaction_ID) REFERENCES TransactionTable(Transaction_ID) ON UPDATE CASCADE,
    Supplier_ID int FOREIGN KEY (Supplier_ID) REFERENCES SupplierTable(Supplier_ID) ON UPDATE CASCADE,
    Entity_ID int FOREIGN KEY (Entity_ID) REFERENCES EntityTable(Entity_ID) ON UPDATE CASCADE,
    Date_ID int FOREIGN KEY (Date_ID) REFERENCES DataTable(Date_ID) ON UPDATE CASCADE,
);

```

Figure 7: SQL Query to create Tables

After creating the tables along with the **Foreign Key** relationship on **Fact Table** a start schema was populated. Below picture shows the start schema.

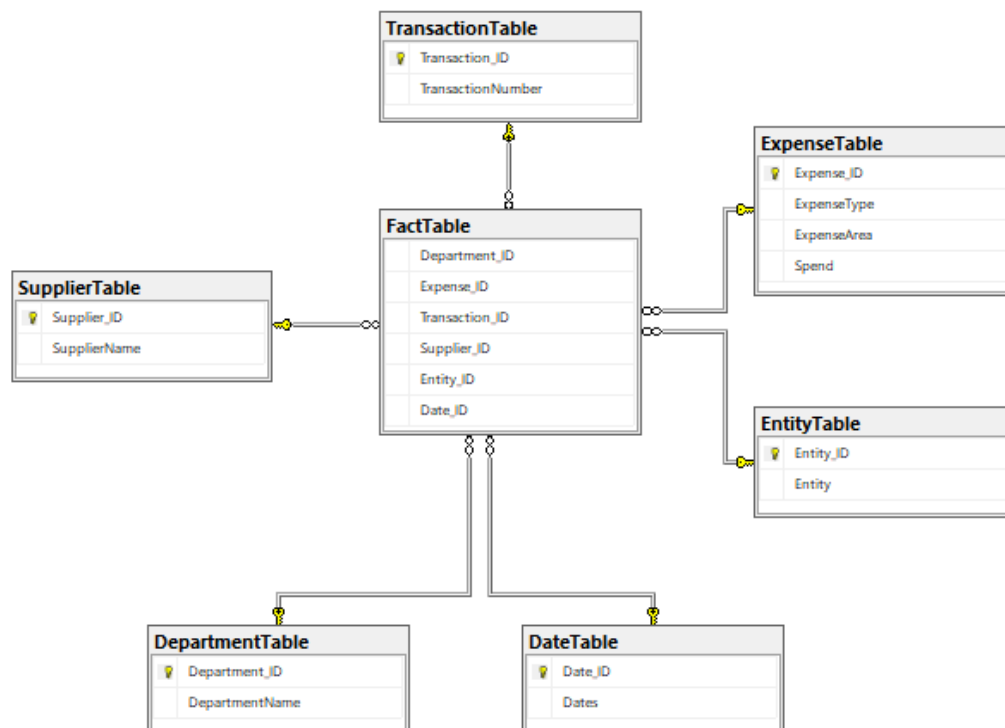


Figure 8: Star schema

After creating all these tables I inserted the cleaned data inside the table from SSIS using OLE DB Destination. Figure below displays the ETL where the cleaned data was inserted to our tables.

From above Figure we created 7 separate table.

Department Table, Supplier Table, Expense Table, Entity Table, Date Table, Transaction Table are connected to Fact Table in one-to-one relation.

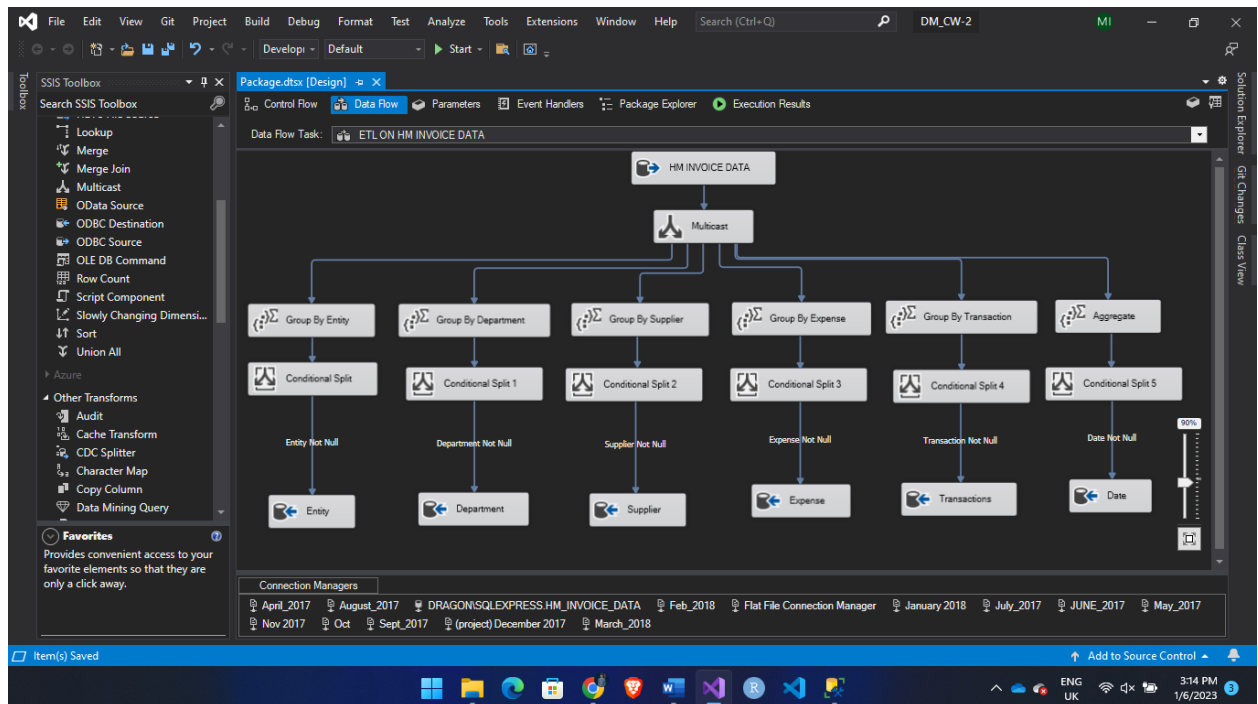


Figure 9: ETL

Critical Analysis – Transforming design to implementation

Merging 12 csv file inside the ETL was a long process and sometimes the values got mixed up with the other tables, however the issue was resolved later where I merged all the 12 files into single excel file and imported the data to the database using SSIS. Below figure demonstrate that.

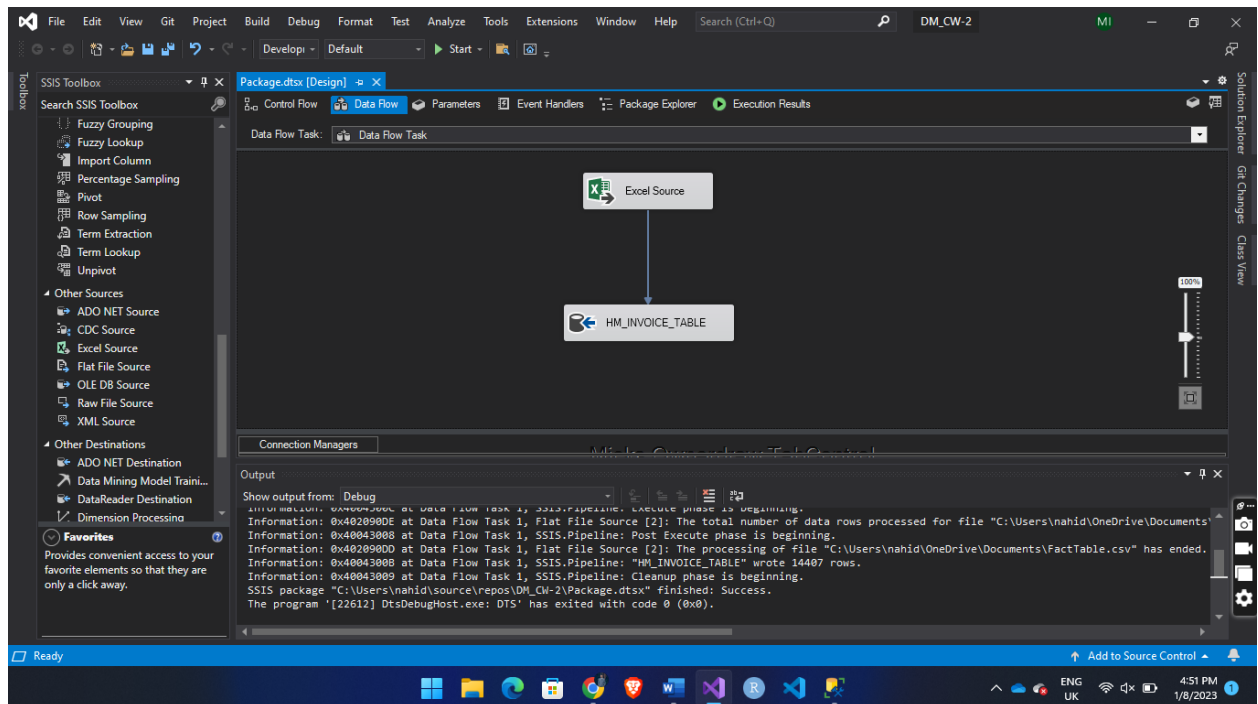


Figure 10: Importing data from excel source to OLE DB Destination

Testing and Problem-Solving Skills

- a. Calculate the top three payment types and the total amount spent on these payment types and in each of the four quarters April-June, July-September, October-December, and January-March.

```
SQLQuery1.sql - not connected  DRAGON\SQLEXPRES...TA - StartSchema  Answer_4_b.sql - not connected  DM_CW_2_QUERY.sql - not connected

/* a. Calculate the top three payment types and the total amount spent
on these payment types and in each of the four quarters April-June, July-September, October-December, and January-March. */

SELECT TOP 3 ExpenseType , Spend AS TotalSpend , 'APRIL - JUNE' AS 'Duration'
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID WHERE dates between '2017-04-01' and '2017-06-30'
UNION
SELECT TOP 3 ExpenseType , Spend AS TotalSpend , 'JULY - SEPTEMBER' AS 'Duration'
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID WHERE dates between '2017-07-01' and '2017-09-30'
UNION
SELECT TOP 3 ExpenseType , Spend AS TotalSpend , 'OCTOBER - DECEMBER' AS 'Duration'
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID WHERE dates between '2017-10-01' and '2017-12-31'
UNION
SELECT TOP 3 ExpenseType , Spend AS TotalSpend , 'JANUARY - MARCH' AS 'Duration'
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID WHERE dates between '2018-01-01' and '2018-03-31'
ORDER BY TotalSpend DESC
```

Output

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the SQL code from the previous block. The Results pane shows the output of the query, which is a table with three columns: Expense Type, TotalSpend, and Duration. The results are sorted by TotalSpend in descending order.

Expense Type	TotalSpend	Duration
IT, Telecoms - Components	202710	OCTOBER - DECEMBER
Passport Application, Production & Delivery	184717.46	JANUARY - MARCH
IT, Telecomms	173340	JULY - SEPTEMBER
Passport Application, Production & Delivery	111783.54	JANUARY - MARCH
IT, Telecoms - Components	92370	OCTOBER - DECEMBER
Passport Application, Production & Delivery	91396.2	JANUARY - MARCH
Grant	81328.12	APRIL - JUNE
Grant	55087.69	APRIL - JUNE
IT, Telecoms - Components	48060	OCTOBER - DECEMBER
IT, Telecomms	45712.8	JULY - SEPTEMBER
Grant	25115.05	APRIL - JUNE
IT, Telecomms	-70746	JULY - SEPTEMBER

- b. Calculate the top four expense areas and the totals amount spent on these areas for the year and for each of the two half-years April - Sept and Oct – Mar

```
SQLQuery1.sql - not connected  DRAGON\SQLEXPRESS...TA - StartSchema  Answer_4_b.sql -...DRAGON\nahid (58))  DM_CW_2_QUERY.sql - not connected

/* b. Calculate the top four expense areas and the totals amount spent on
these areas for the year and for each of the two half-years April - Sept and Oct - Mar */

SELECT TOP 3 ExpenseArea , Spend AS TotalSpend , 'YEAR' AS 'Duration'
FROM FactTable AS fact
left join DataTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
UNION
SELECT TOP 3 ExpenseArea , Spend AS TotalSpend , 'APRIL - SEPTEMBER' AS 'Duration'
FROM FactTable AS fact
left join DataTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID WHERE dates between '2017-04-01' and '2017-09-30'
UNION
SELECT TOP 3 ExpenseArea , Spend AS TotalSpend , 'OCTOBER - MARCH' AS 'Duration'
FROM FactTable AS fact
left join DataTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID WHERE dates between '2017-10-01' and '2018-03-31'

ORDER BY TotalSpend DESC
```

Output

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the SQL code for calculating the top four expense areas and their total spend for the year and two half-years. The results window shows the output of the query, which is a table with three columns: ExpenseArea, TotalSpend, and Duration. The results are ordered by TotalSpend in descending order.

ExpenseArea	TotalSpend	Duration
CPFG - Crime Policing & Fire Group	151836.72	APRIL - SEPTEMBER
HMPO - Her Majesty's Passport Office	111783.54	OCTOBER - MARCH
CPFG - Crime Policing & Fire Group	90558.38	APRIL - SEPTEMBER
CPFG - Crime Policing & Fire Group	79514.02	YEAR
CPFG - Crime Policing & Fire Group	66820.48	YEAR
UKBF - Border Force	61282.03	YEAR
CPFG - Crime Policing & Fire Group	52567.2	APRIL - SEPTEMBER
HMPO - Her Majesty's Passport Office	46018.94	OCTOBER - MARCH
HMPO - Her Majesty's Passport Office	33811.43	OCTOBER - MARCH

- a. For each quarter of the year rank the top 10 Suppliers by total net spend made to them by the home office. Clearly indicate the change in rank for each quarter. The rankings must be in ascending order.

```
SQLQuery1.sql - not connected  DRAGON\SQLEXPRES...TA - StartSchema  Answer_4_b.sql -...DRAGON\nahid (58))*  D

/* c. For each quarter of the year rank the top 10 Suppliers by total net spend made to
them by the home office. Clearly indicate the change in rank for each quarter. The rankings must be in ascending order */
SELECT TOP 10 SupplierName, SUM(Spend) AS Spending,
RANK() OVER(ORDER BY SUM(Spend) DESC) Q, 0 AS Q2, 0 AS Q3, 0 AS Q4
FROM FactTable AS fact
INNER JOIN DataTable AS dates ON fact.Date_ID = dates.Date_ID
INNER JOIN ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
INNER JOIN SupplierTable AS supplier ON fact.Supplier_ID = supplier.Supplier_ID
WHERE dates BETWEEN '2017-04-01' AND '2017-06-30' GROUP BY SupplierName
UNION
SELECT TOP 10 SupplierName, SUM(Spend) AS Spending, 0 AS Q1,
RANK() OVER(ORDER BY SUM(Spend) DESC) Q2, 0 AS Q3, 0 AS Q4
FROM FactTable AS fact
INNER JOIN DataTable AS dates ON fact.Date_ID = dates.Date_ID
INNER JOIN ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
INNER JOIN SupplierTable AS supplier ON fact.Supplier_ID = supplier.Supplier_ID
WHERE dates BETWEEN '2017-07-01' AND '2017-09-30' GROUP BY SupplierName
UNION
SELECT TOP 10 SupplierName, SUM(Spend) AS Spending, 0 AS Q1, 0 AS Q2,
RANK() OVER(ORDER BY SUM(Spend) DESC) Q3, 0 AS Q4
FROM FactTable AS fact
INNER JOIN DataTable AS dates ON fact.Date_ID = dates.Date_ID
INNER JOIN ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
INNER JOIN SupplierTable AS supplier ON fact.Supplier_ID = supplier.Supplier_ID
WHERE dates BETWEEN '2017-10-01' AND '2017-12-31' GROUP BY SupplierName
UNION
SELECT TOP 10 SupplierName, SUM(Spend) AS Spending, 0 AS Q1, 0 AS Q2, 0 AS Q3,
RANK() OVER(ORDER BY SUM(Spend) DESC) Q4
FROM FactTable AS fact
INNER JOIN DataTable AS dates ON fact.Date_ID = dates.Date_ID
INNER JOIN ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
INNER JOIN SupplierTable AS supplier ON fact.Supplier_ID = supplier.Supplier_ID
WHERE dates BETWEEN '2018-01-01' AND '2018-03-31' GROUP BY SupplierName ORDER BY Spending ASC
```

Output

Query executed successfully. DRAGON\SQLEXPRESS (15.0 RTM) | DRAGON\nahid (58) | HM_INVOICE_DATA | 00:00:00 | 40 rows

	SupplierName	Spending	Q	Q2	Q3	Q4
1	MITIE CARE & CUSTODY LTD	14073010.06	0	0	10	0
2	POLICE & CRIME COMMISSI...	14535128.41	0	0	9	0
3	METHODS PROFESSIONAL ...	16330828.35	0	0	8	0
4	PUBLIC HEALTH ENGLAND(...	16508667.24	0	0	0	10
5	POST OFFICE LTD	17855964.31	0	0	0	9
6	MORTON FRASER	20802965.7	0	0	7	0
7	SUFFOLK COUNTY COUNCIL...	21577488.63	0	0	0	8
8	OFFICE OF THE POLICE & C...	21693130.79	0	0	6	0
9	POLICE & CRIME COMMISSI...	26854646.87	0	0	0	7
10	SOPRA STERIA LTD	29267581.5	0	0	0	6
11	TASCOR E D SERVICES LIM...	31689059.05	0	0	0	5
12	OFFICE OF THE KENT POLI...	31749759.64	0	0	5	0
13	POLICE & CRIME COMMISSI...	31819548.14	0	0	0	4
14	OFFICE OF THE POLICE & C...	34415226.54	0	0	4	0
15	POLICE & CRIME COMMISSI...	34876730.5	0	0	0	3
16	POLICE & CRIME COMMISSI...	35761734.34	0	0	0	2
17	NORTHAMPTONSHIRE OFF...	35996406.57	0	0	3	0
18	OFFICE OF THE DURHAM P...	36382279.08	0	0	2	0
19	POLICE & CRIME COMMISSI...	38735817.6	0	0	0	1
20	OFFICE OF THE POLICE & C...	45992349.81	0	0	1	0
21	LONDON BOROUGH OF BR...	76535710	0	10	0	0
22	LONDON BOROUGH OF BA...	88792097	0	9	0	0
23	HM REVENUE & CUSTOMS	91074391	0	8	0	0
24	GOVERNMENT LEGAL DEP...	105361602...	0	7	0	0

- c. Create a fourth complex query utilizing a time related hierarchy of your own design that demonstrates the full complexity of your understanding in terms of analytic query writing.

```
SQLQuery1.sql - not connected  DRAGON\SQLEXPRESS...TA - StartSchema  Ansewer_4_b.sql -...DRAGON\nahid (58))  DM_CW_2_QUERY.sql - not connected

/** d. Create a fourth complex query utilizing a time related hierarchy of
your own design that demonstrates the full complexity of your
understanding in terms of analytic query writing.*/

SELECT TOP 3 Entity, AVG(Spend) AS AverageSpend , TransactionNumber, 'APRIL - SEPTEMBER' AS 'Duration'
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
left join EntityTable as entity ON fact.[Entity_ID] = entity.[Entity_ID]
left join TransactionTable AS T ON fact.Transaction_ID = T.Transaction_ID WHERE dates between '2017-04-01' and '2017-09-30'
GROUP BY Entity,TransactionNumber
UNION
SELECT TOP 3 Entity, AVG(Spend) AS AverageSpend , TransactionNumber, 'OCTOBER - MARCH' AS 'Duration'
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
left join EntityTable as entity ON fact.[Entity_ID] = entity.[Entity_ID]
left join TransactionTable AS T ON fact.Transaction_ID = T.Transaction_ID WHERE dates between '2017-10-01' and '2018-03-31'
GROUP BY Entity,TransactionNumber
ORDER BY AverageSpend DESC
```

Output:

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'HM_INVOICE_DATA' database selected. The right pane shows the 'SQL Query' window with the following query:

```
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
left join EntityTable as entity ON fact.[Entity_ID] = entity.[Entity_ID]
left join TransactionTable AS T ON fact.Transaction_ID = T.Transaction_ID WHERE dates between '2017-04-01' and '2017-09-30'
GROUP BY Entity,TransactionNumber
UNION
SELECT TOP 3 Entity, AVG(Spend) AS AverageSpend , TransactionNumber, 'OCTOBER - MARCH' AS 'Duration'
FROM FactTable AS fact
left join DateTable AS dates ON fact.Date_ID = dates.Date_ID
left join ExpenseTable AS expense ON fact.Expense_ID = expense.Expense_ID
left join EntityTable as entity ON fact.[Entity_ID] = entity.[Entity_ID]
left join TransactionTable AS T ON fact.Transaction_ID = T.Transaction_ID WHERE dates between '2017-10-01' and '2018-03-31'
GROUP BY Entity,TransactionNumber
```

The 'Results' pane shows the output of the query, which is a table with 6 rows and 4 columns: Entity, AverageSpend, TransactionNumber, and Duration. The data is as follows:

Entity	AverageSpend	TransactionNumber	Duration
Home Office	407278.31	1002008820	APRIL - SEPTEMBER
Home Office	372126.865	1002008818	APRIL - SEPTEMBER
Home Office	107756.135	1002008819	APRIL - SEPTEMBER
Home Office	43498.6	1002083763	OCTOBER - MAR...
Home Office	40315.2	1002083792	OCTOBER - MAR...
Home Office	2484	1002083783	OCTOBER - MAR...

The status bar at the bottom indicates that the query was executed successfully, returning 6 rows in 00:00:00. The taskbar at the bottom shows the system clock as 5:09 PM on 1/8/2023.

Practical Competence – Dashboard

Connection Tableau with MS SQL SERVER to get the data

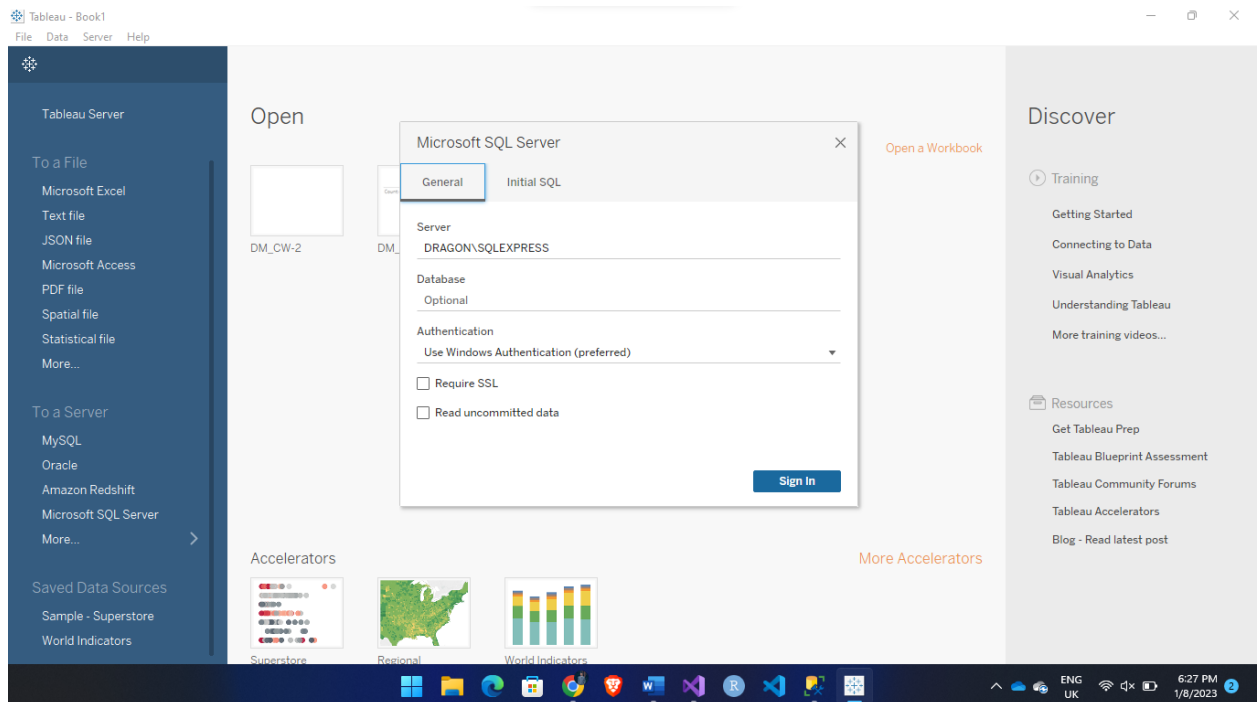
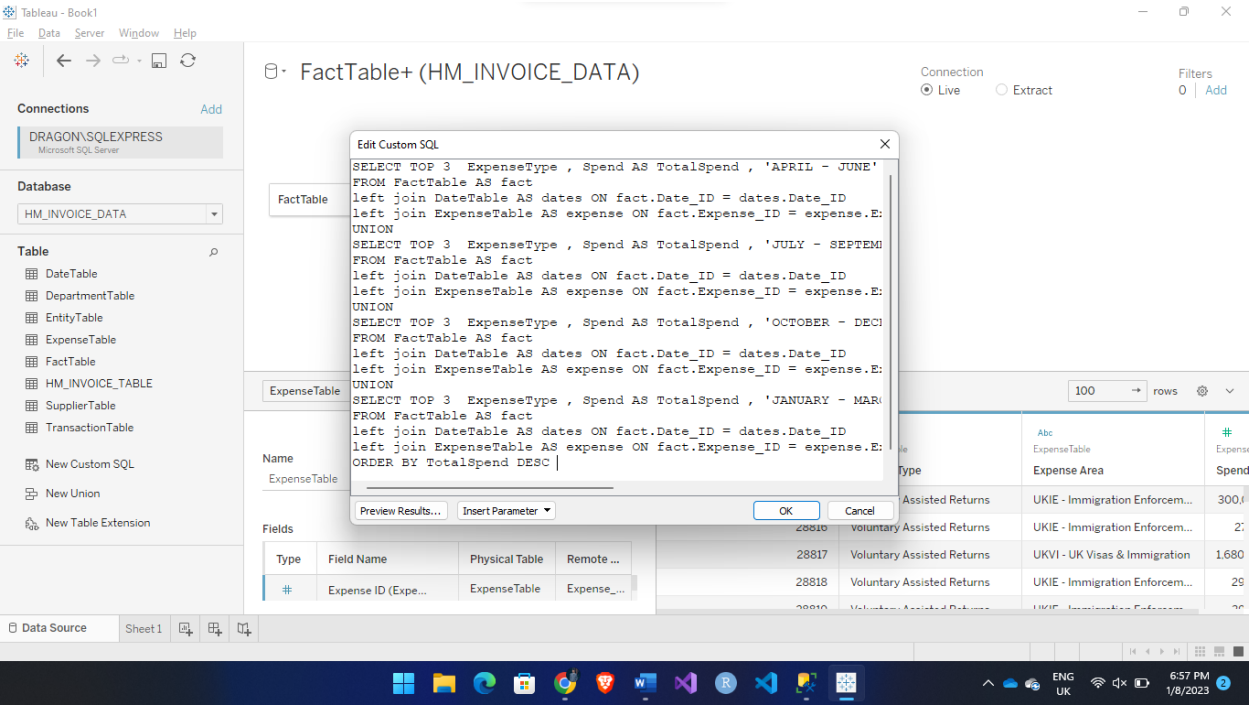
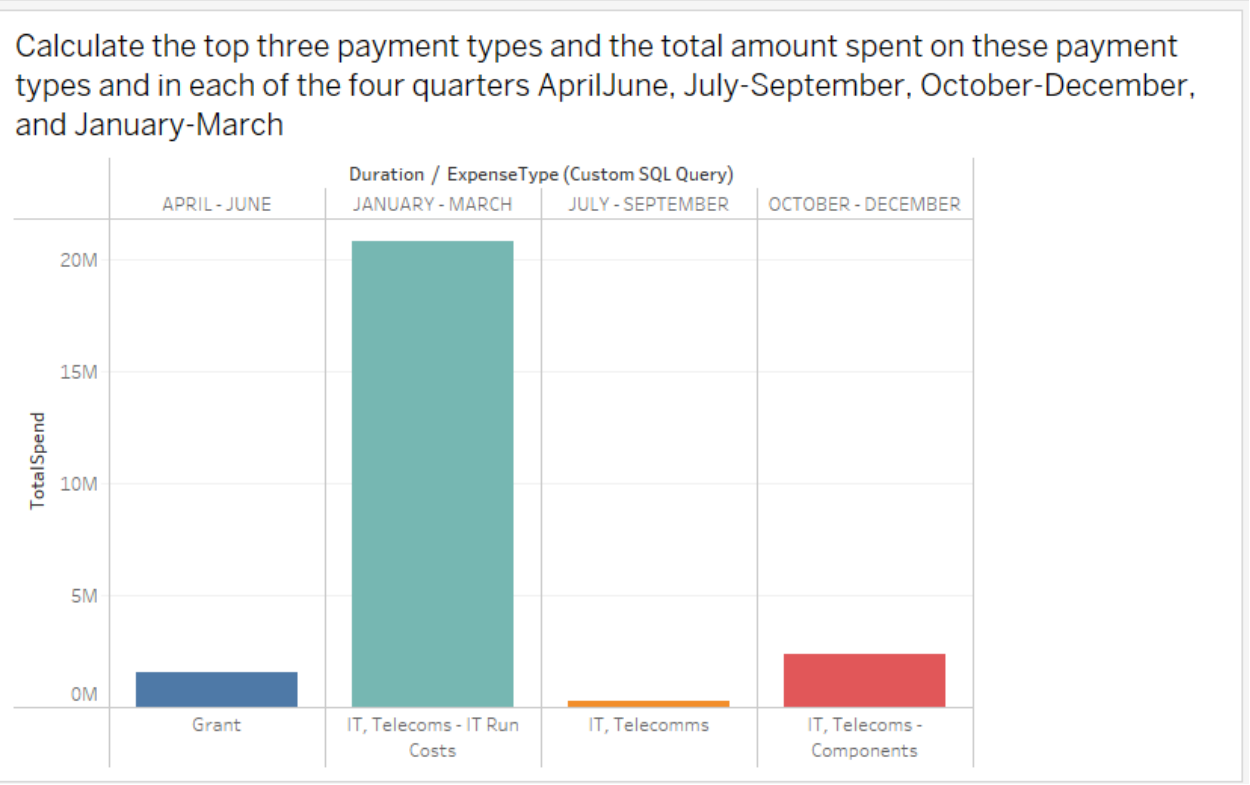


Figure Below shows how to create a custom query in Tableau

At the bottom of the left sidebar, the New Custom SQL button opens up a popup where we put our custom query to get the data and display in Tableau.

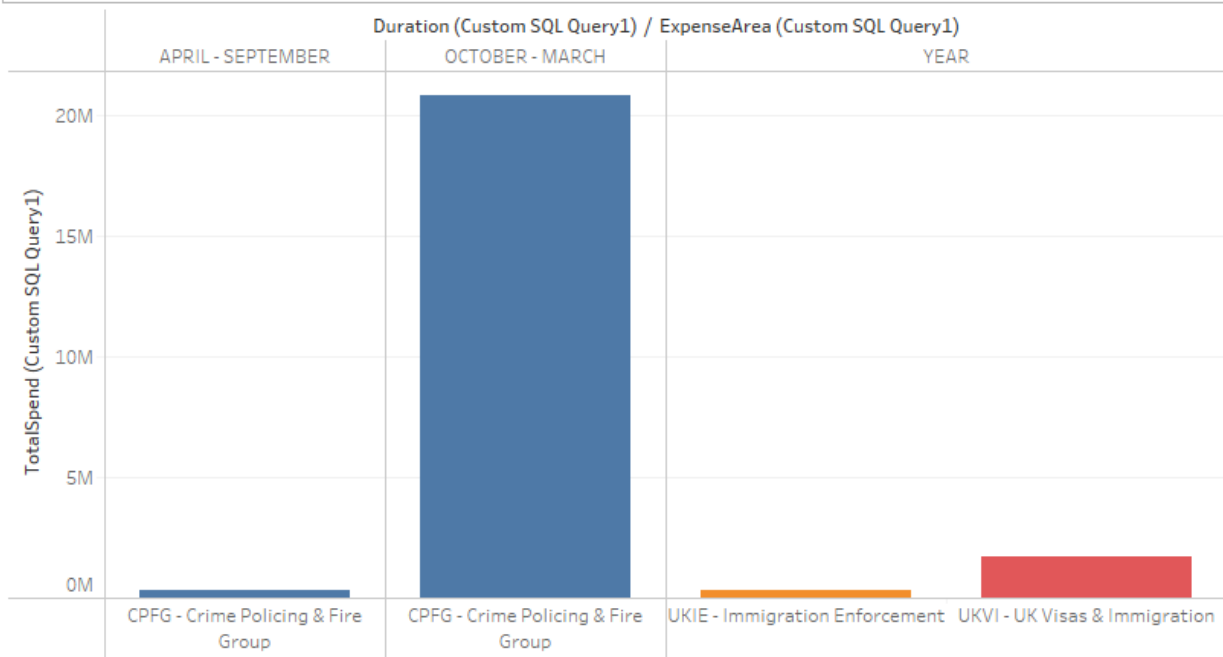


Visualizing First Query



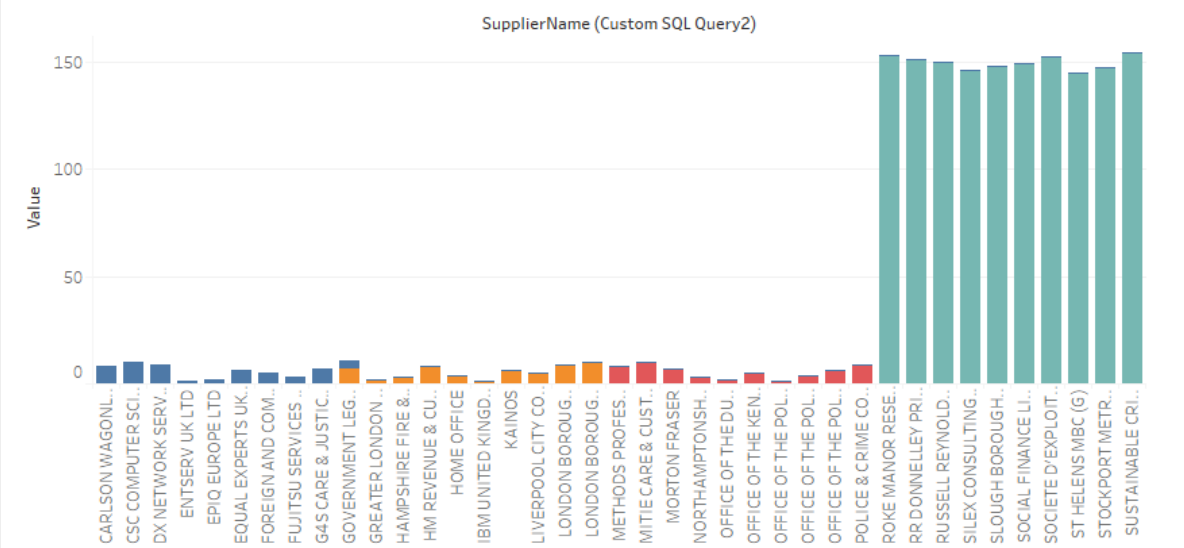
Visualizing Second Query

b. Calculate the top four expense areas and the totals amount spent on these areas for the year and for each of the two half-years April - Sept and Oct - Mar



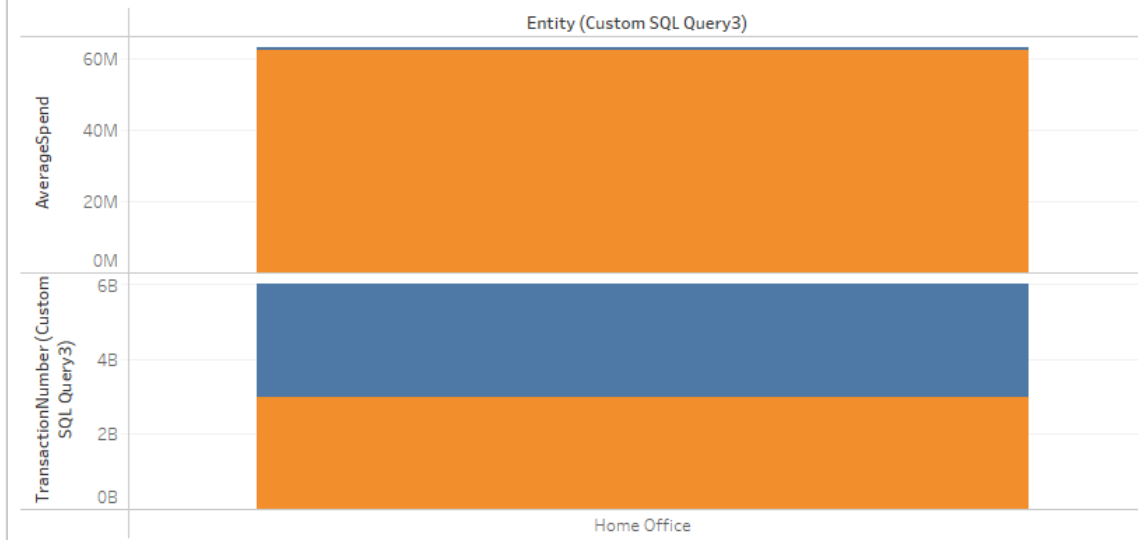
Visualizing Third Query

c. For each quarter of the year rank the top 10 Suppliers by total net spend made to them by the home office. Clearly indicate the change in rank for each quarter The rankings must be in ascending order



Visualizing Fourth Query

d. Create a fourth complex query utilizing a time related hierarchy of your own design that demonstrates the full complexity of your understanding in terms of analytic query writing



[VIDEO DEMO](#)