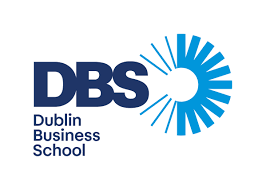
Design and Development of data storage solutions for analysis



By Tim Browning & Clinton Ngan

Contents

[Q1) (a) Develop a proof of concept data warehouse/data mart (using dimensional modelling) capturing data from an existing data source? 3](#_Toc38205711)

[Q1) (b) Document your reasons for selecting the subject area? 4](#_Toc38205712)

[Q1) (C) Identify key stakeholders? 4](#_Toc38205713)

[Q1) (D) Explain the insights that a company may derive from the given data? 4](#_Toc38205714)

[Q2) (D) Develop and present a suitable schema for the data warehouse / data mart. Discuss your reasons for the design? 5](#_Toc38205715)

[Q3) Using Microsoft SQL Server, implement your tables and extract, transform and load data from the operational source(s) into the data warehouse. Any suitable tool and / or native SQL statements may be used 7](#_Toc38205716)

[Q4) Produce four reports in support of the requirements outlined in section 1 using SSRS. Also produce four visualisations using Tableau. Discuss the visualisations and incorporate into a Tableau dashboard. 13](#_Toc38205717)

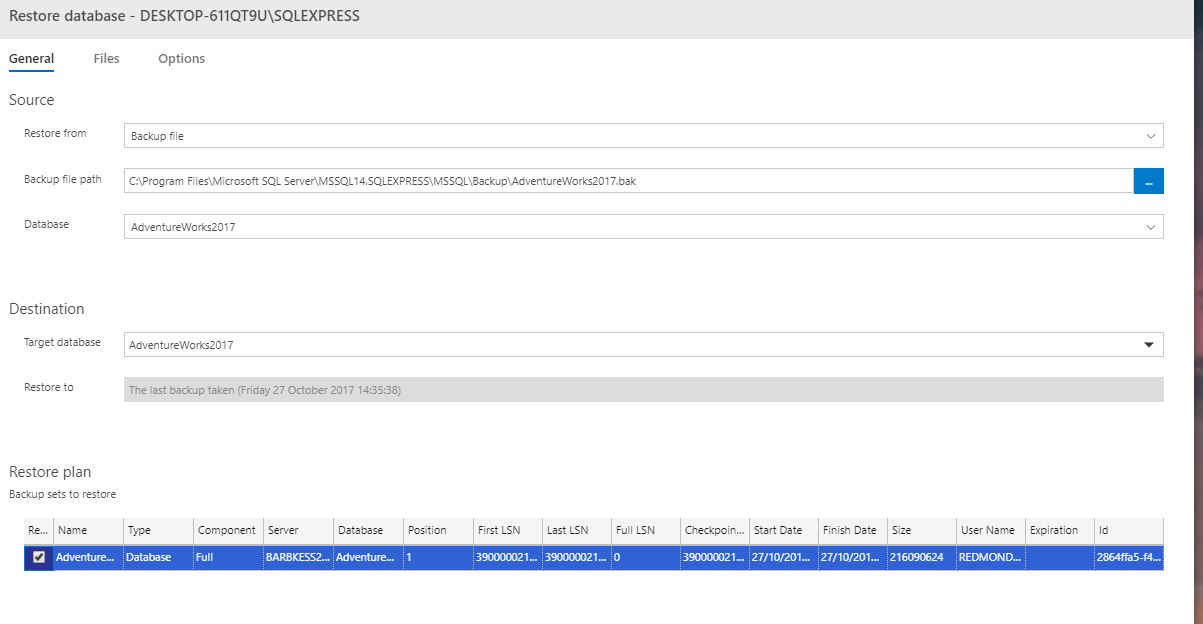
[Q5) Multiple studies have compared the performance of graph and relational database management systems. Use a relational E-R data model of an OLTP database such as Northwind and implement the OLTP database as a graph data model using the Neo4j graph database management system. Identify seven suitable queries and implement the queries in SQL and CYPHER against the relational and graph data models respectively to demonstrate the differences in storing/retrieving data from relational and graph databases. 17](#_Toc38205718)

[Q7) Contribution – Tim Browning 22](#_Toc38205719)

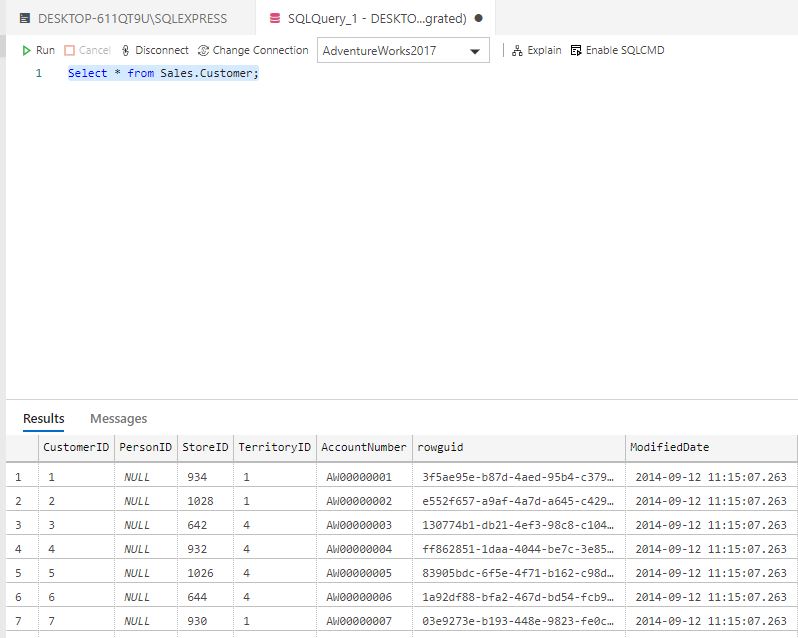
[Bibliography 23](#_Toc38205720)

## Q1) (a) Develop a proof of concept data warehouse/data mart (using dimensional modelling) capturing data from an existing data source?

1. Download AdventureWorks2017 from <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15> (MashaMSFT, n.d.)
2. Restore AdventureWorks2017.bak



1. Access/View tables from data warehouse



## Q1) (b) Document your reasons for selecting the subject area?

We have chosen “sales” as the subject area for the following reasons:

* Adventure Works is a sales-driven manufacturing company
* Retrieved data insights from the data warehouse can be used to increase profits
* Target Audiences/ Markets can be identified

## Q1) (C) Identify key stakeholders?

Data Producers, Data Custodians, Data Consumers and Data Managers

## Q1) (D) Explain the insights that a company may derive from the given data?

Please see reports at Q4 and graphs on Q5

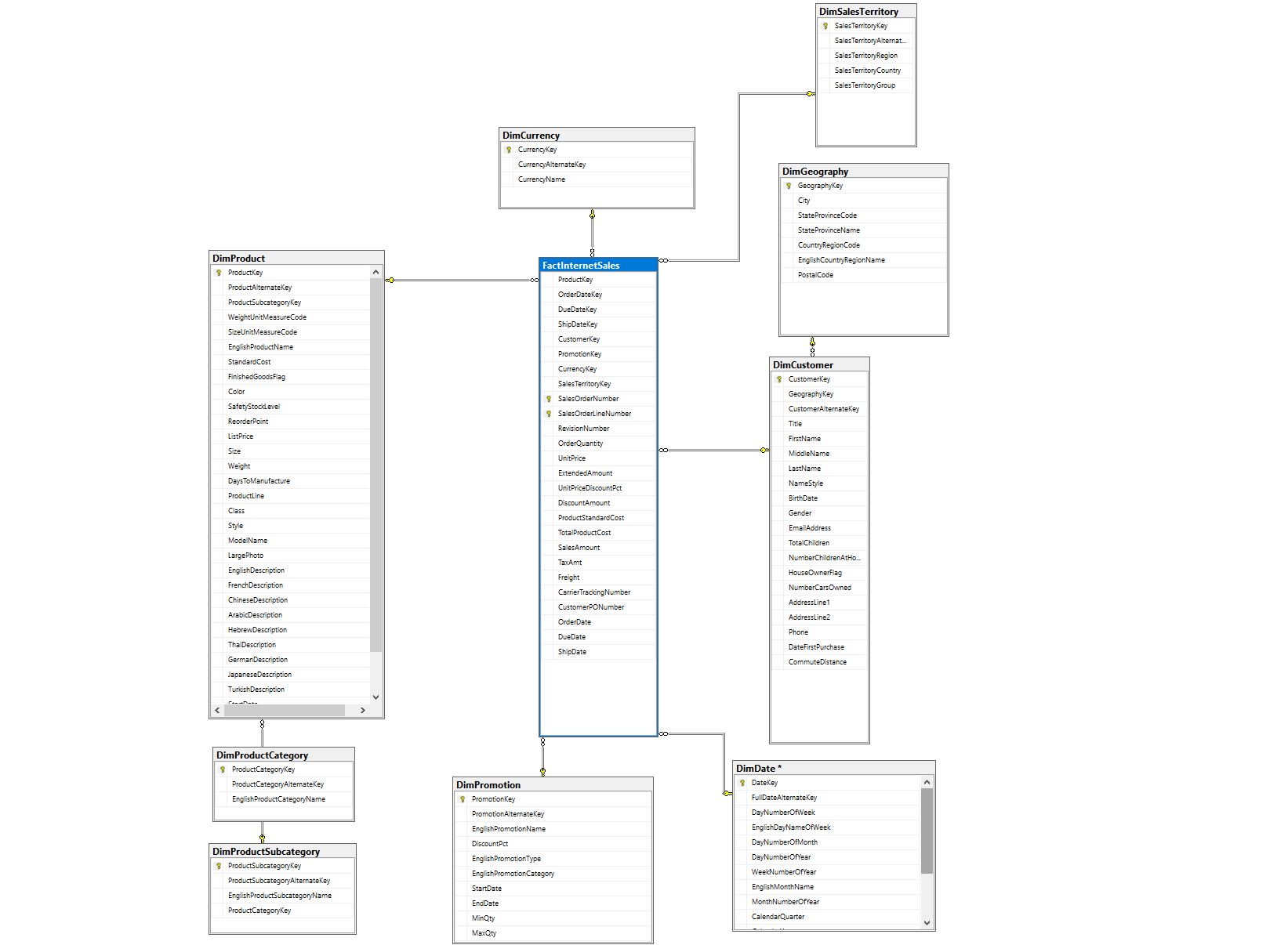
## Q2) (D) Develop and present a suitable schema for the data warehouse/data mart. Discuss your reasons for the design?

I have decided on a star schema for my database design. This star schema captures the sales aspect of the business. This star schema was created in SSMS.

* This star schema was designed to quickly browse a single dimension table. You can browse a single dimension table to determine the constraints and row headers to use when you query the fact table.
* In this star schema, the dimensions are linked only through the central fact table. This improves Query performance
* A time dimension was selected called “DimDate”. This can make it possible to analyse historic data.
* This schema features only one fact table, FactInternetSales. This fact table is a collection of keys and measures. The keys relate each row in the fact table to an associated row in a dimension table.

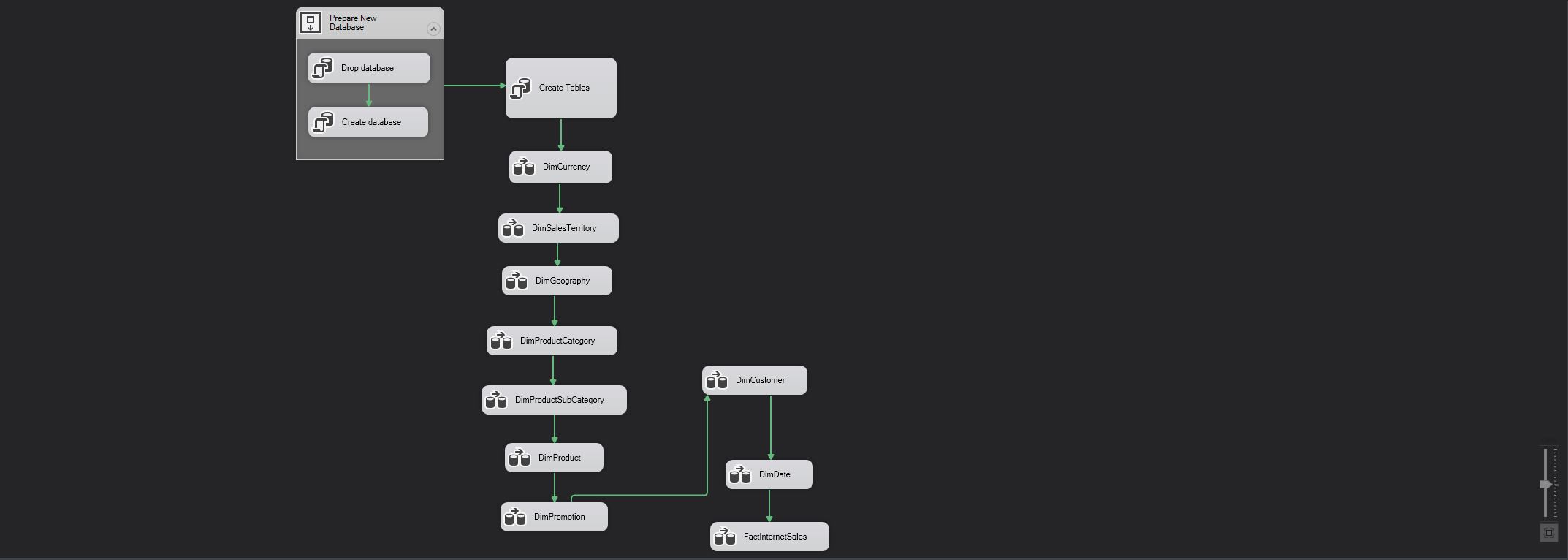
Tables Used:

* DimProduct
* DimProductSubcategory
* DimProductCategory
* DimCurrency
* FactInternetSales
* DimPromotion
* DimSalesTerritory
* DimGeography
* DimCustomer
* DimDate – This table was sourced from (Ryan, 2014)



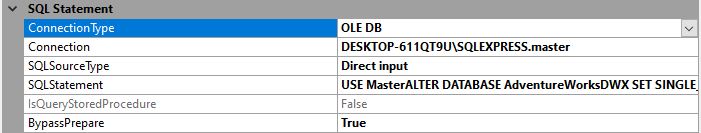
## Q3) Using Microsoft SQL Server, implement your tables and extract, transform and load data from the operational source(s) into the data warehouse. Any suitable tool and / or native SQL statements may be used

SSIS package created in visual studio 2019

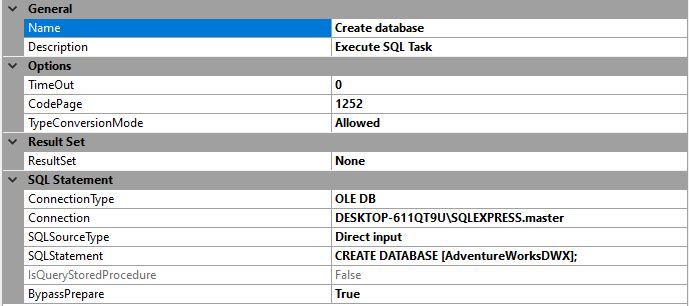


SQL statements are found @ <https://github.com/timbrowning/DataStorage> - inserts.sql & Tables.sql

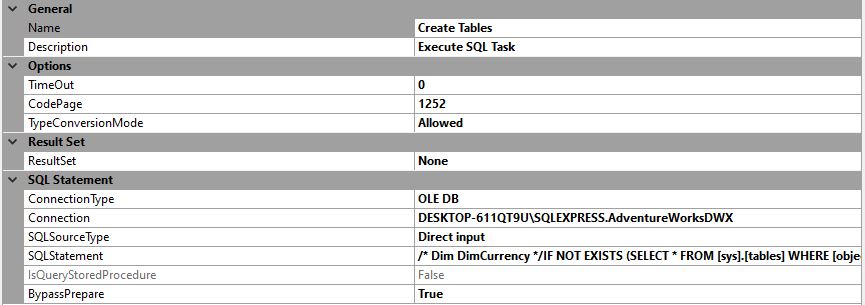
Drop Database: Removes database if database exist



Create Database: Create Database AdventureWorksDWX

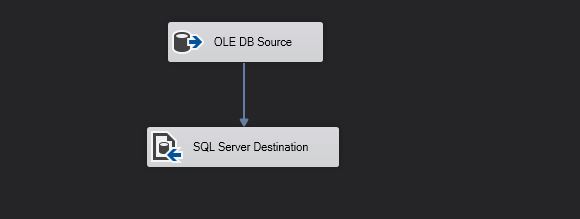


Create Tables: Create Tables for AdventureWorksDWX

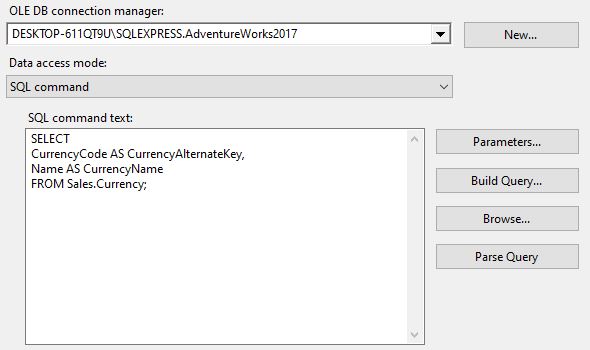
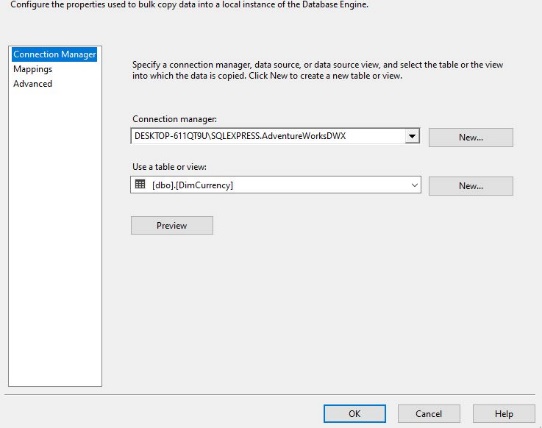


Inserting the data into the data warehouse AdventureWorksDWX –

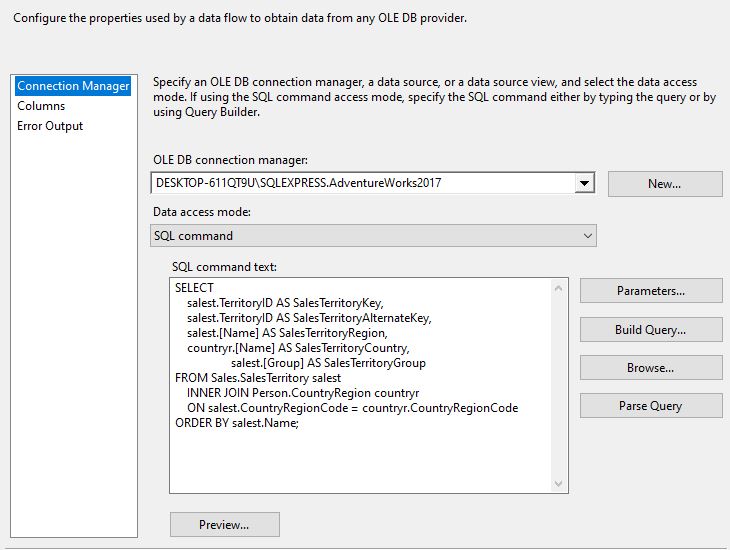
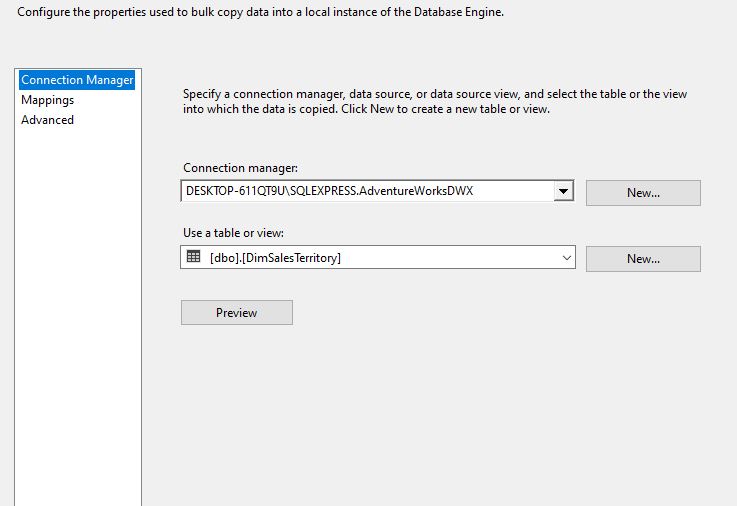
The OLE DB source extracts data from the OLE DB relational database (AdventureWorks2017). In this case we have extracted the view using a SQL command

The SQL server destination provides a connection to the local SQL server database and bulk loads the data into the SQL server tables. In this case the server destination is the AdventureWorksDWX data warehouse. 

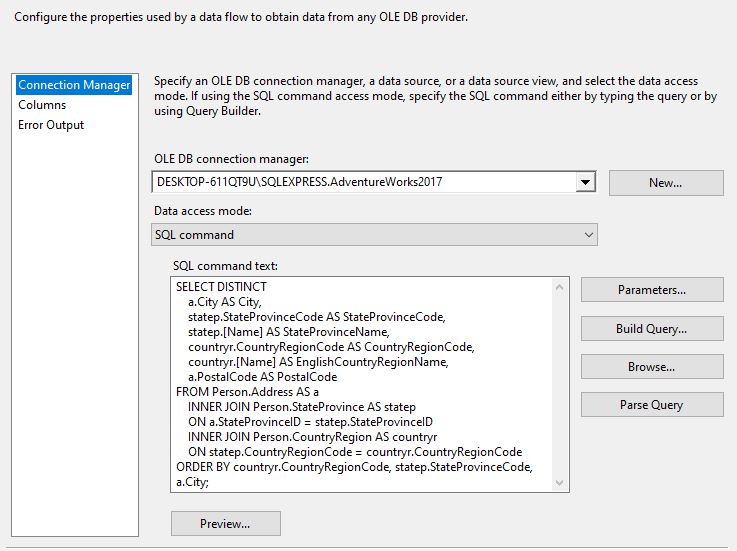
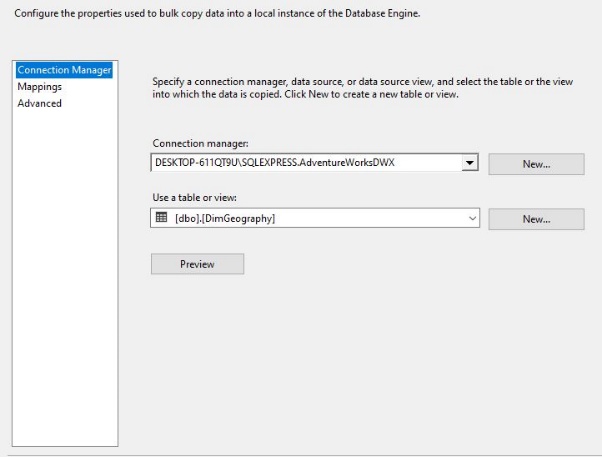
DimCurrency

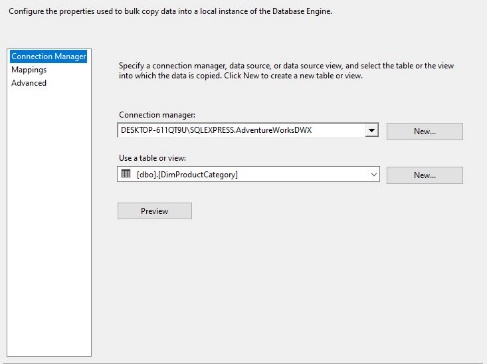
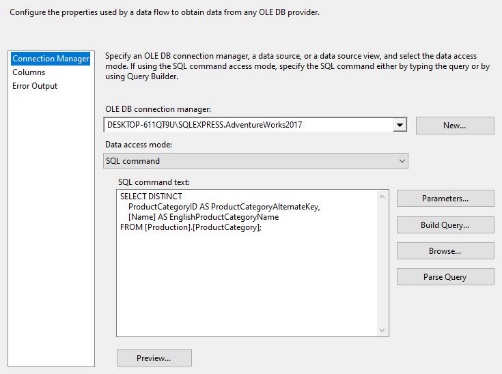
DimSalesTerritoy

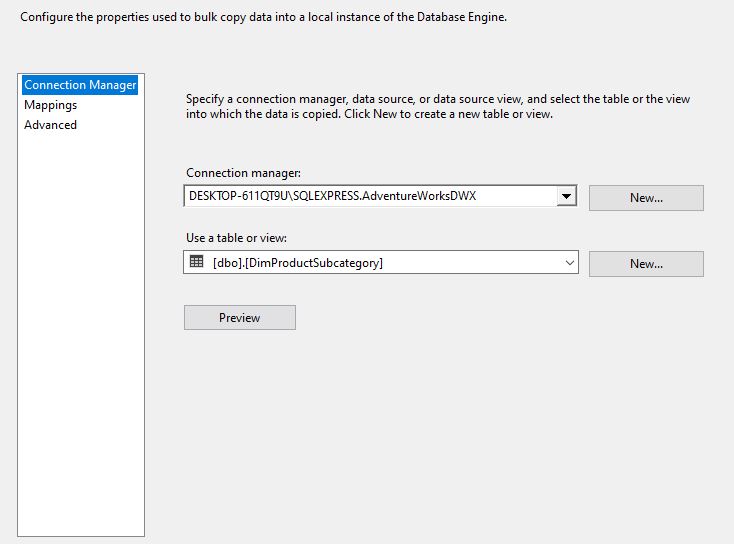
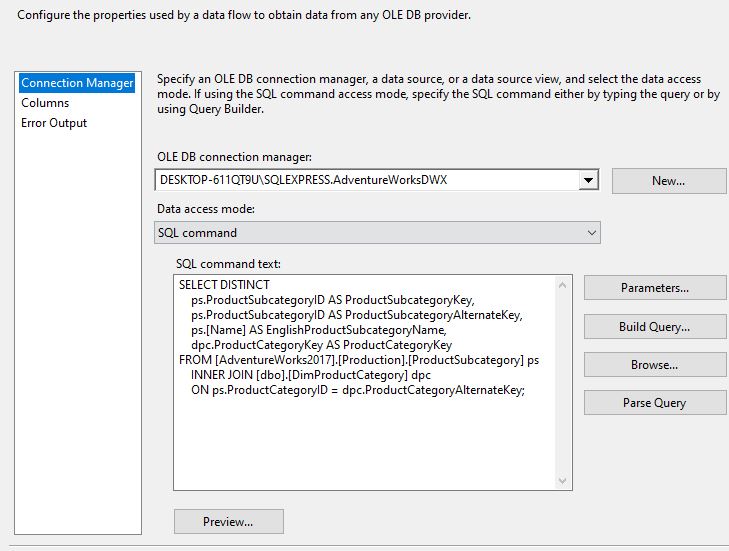
DimGeography

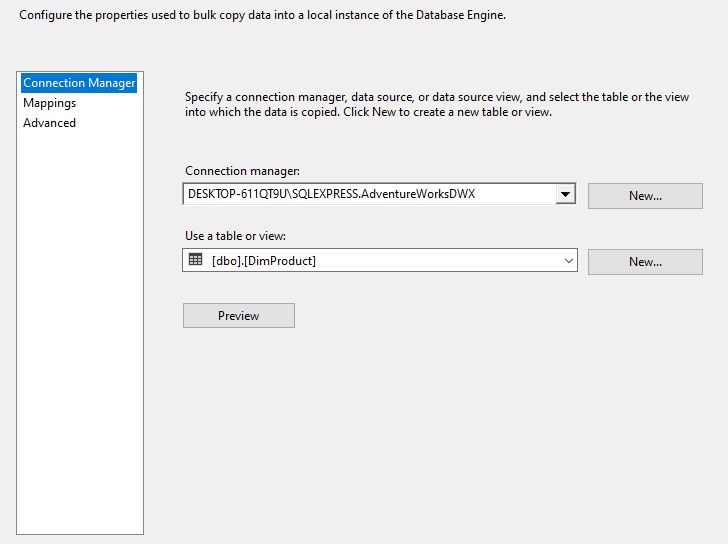
DimProductCateegory



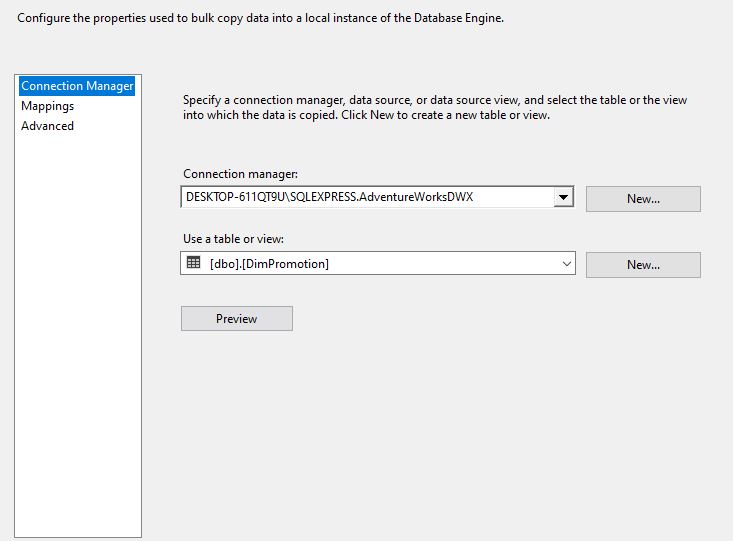
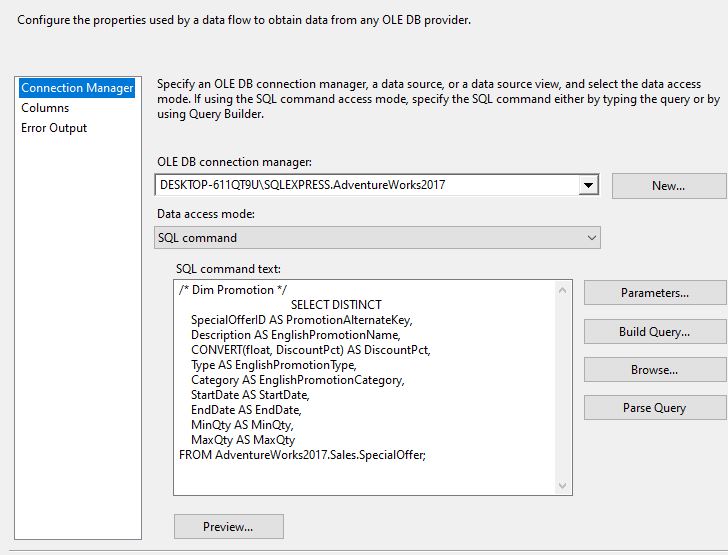
DimProductSubCategory



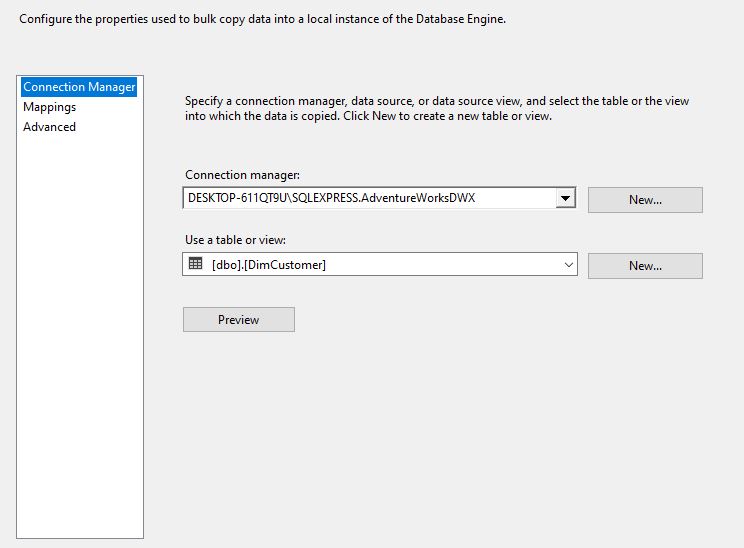
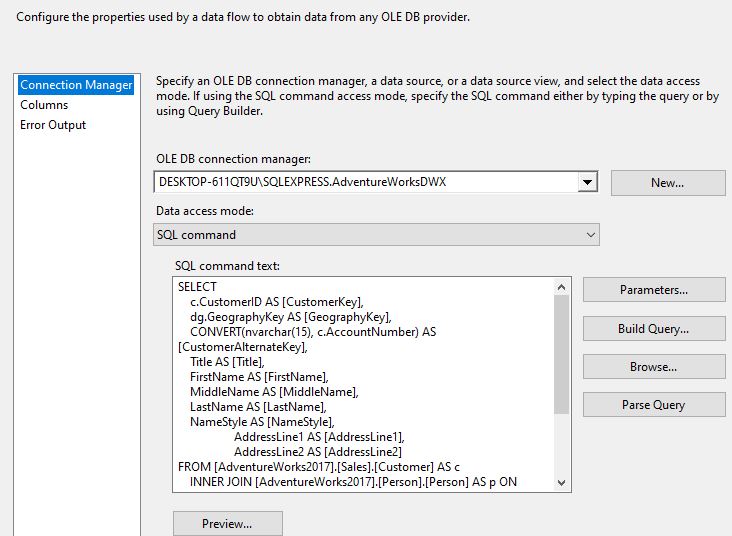
DimProduct



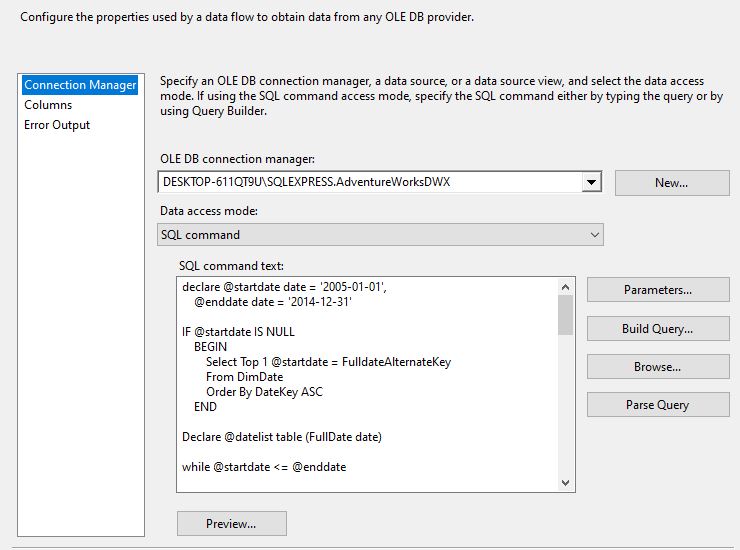
DimPromotion



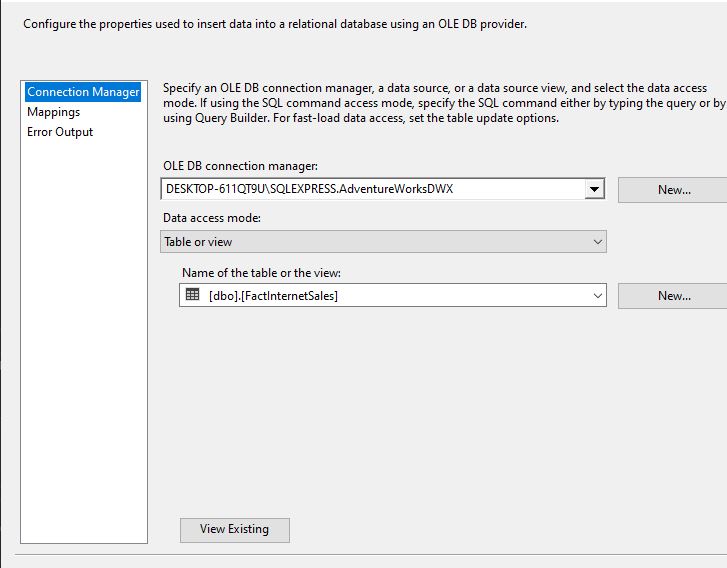
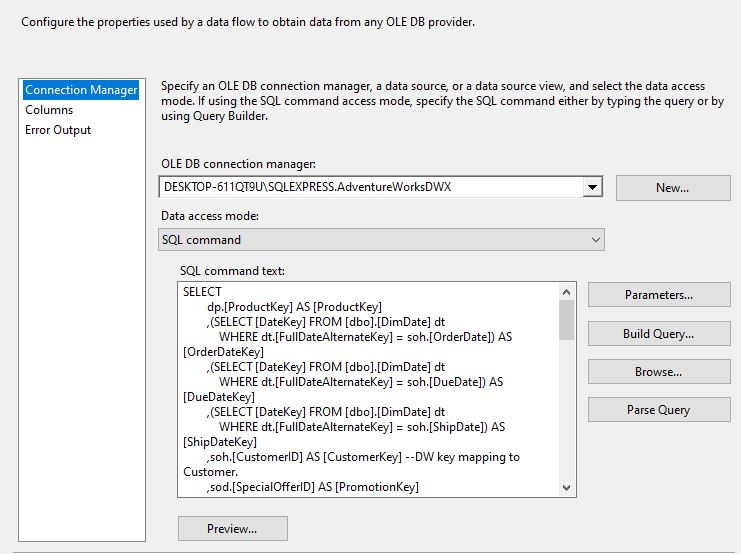
DimCustomer

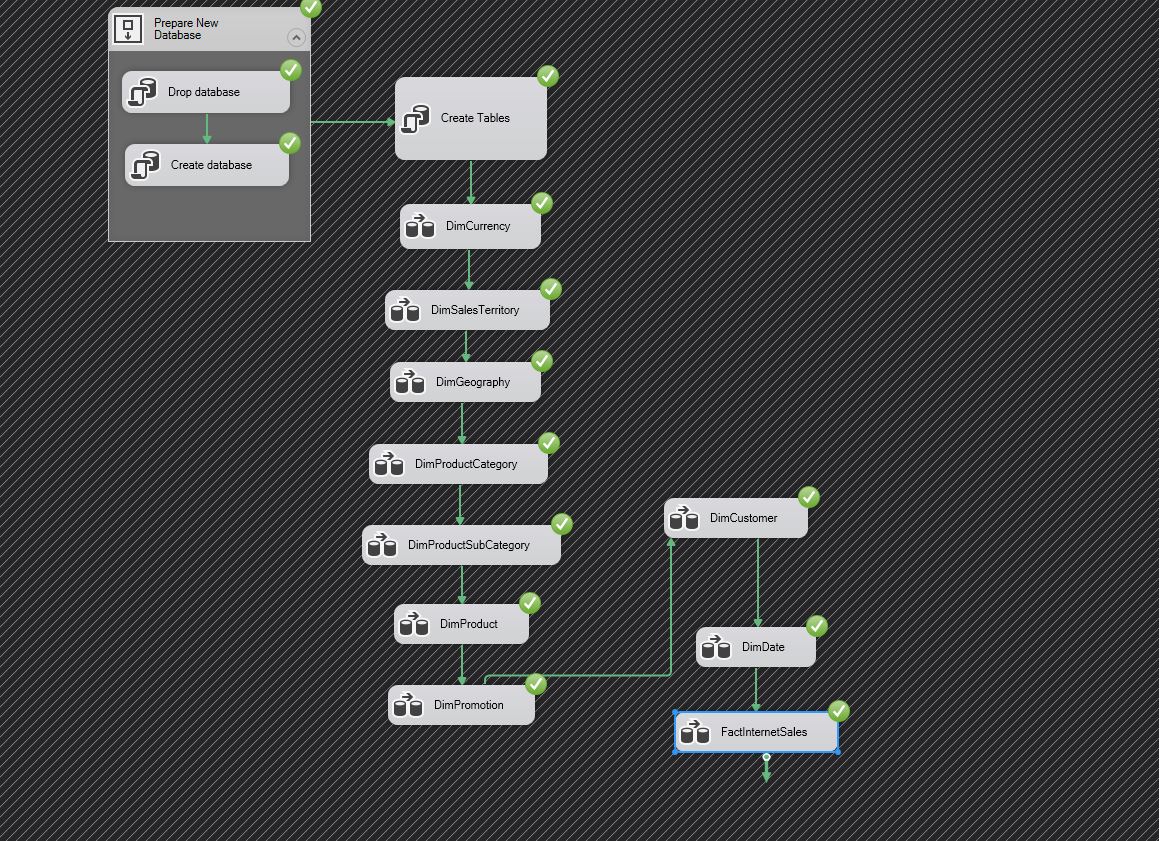


DimDate

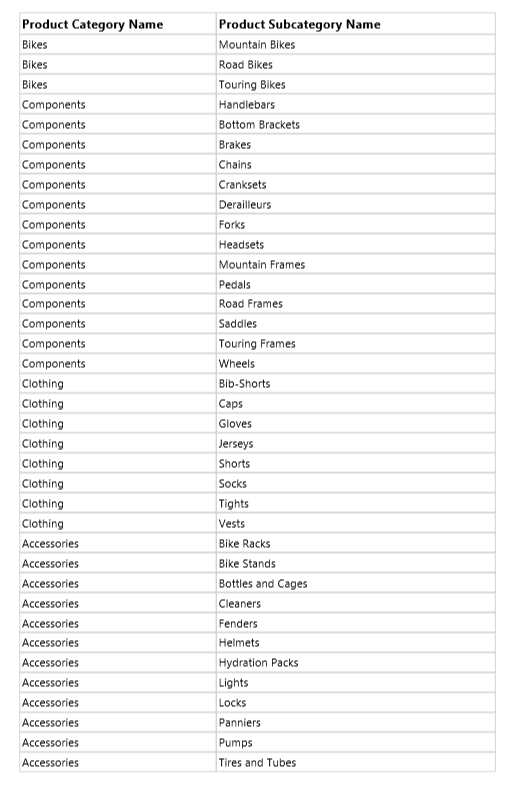


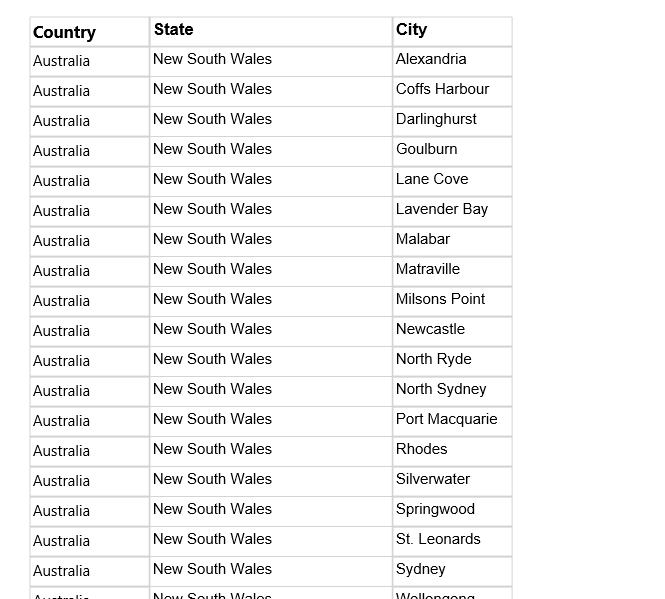
FactInternetSales



Once the start button is pressed the All tables have been created and loaded

## Q4) Produce four reports in support of the requirements outlined in section 1 using SSRS. Also produce four visualisations using Tableau. Discuss the visualisations and incorporate into a Tableau dashboard.

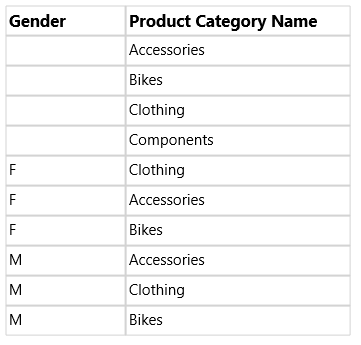
* Report 1 (Products Category and SubCategory)
* Report 2 (Country,State,City)



* Report 3 (Category and existing discounts)

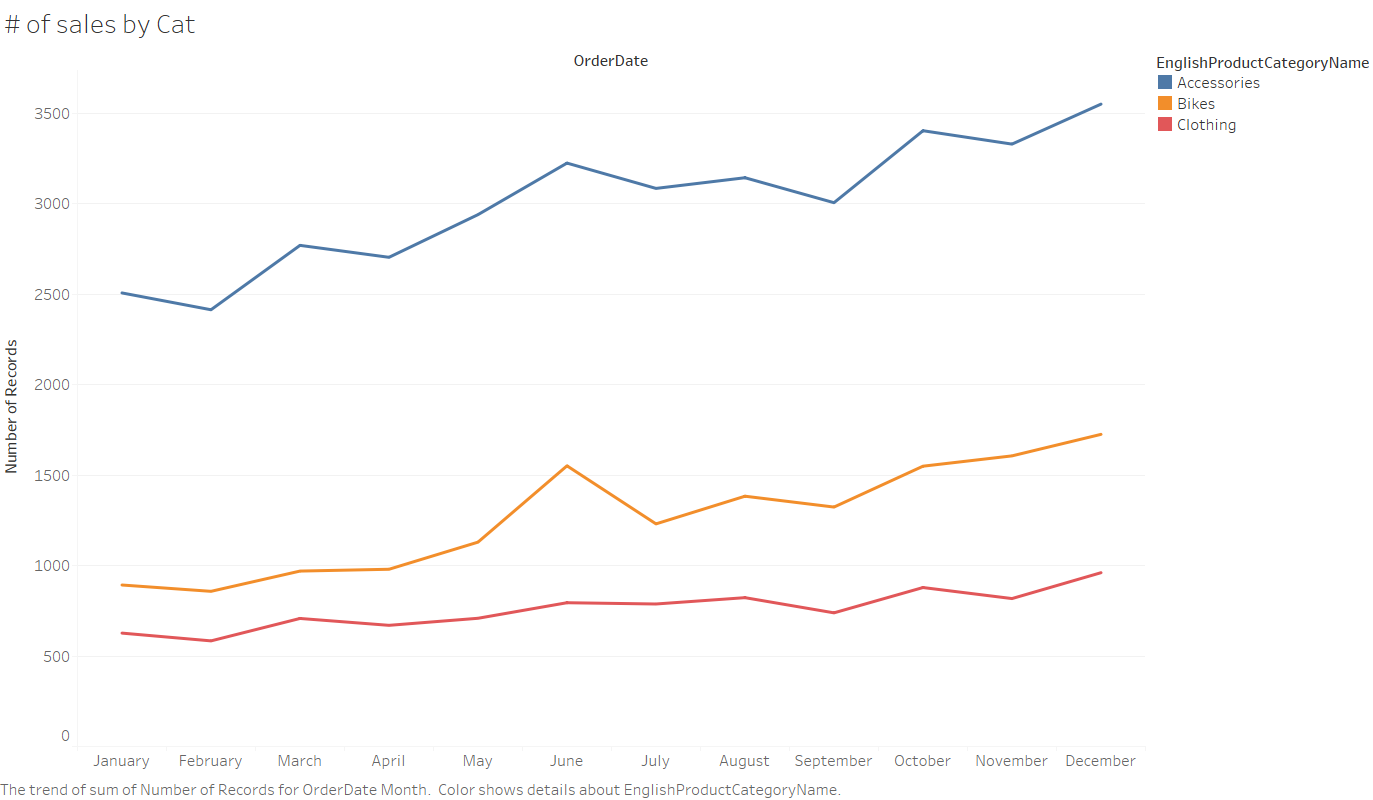


* Report 4 (Gender vs product category purchases)



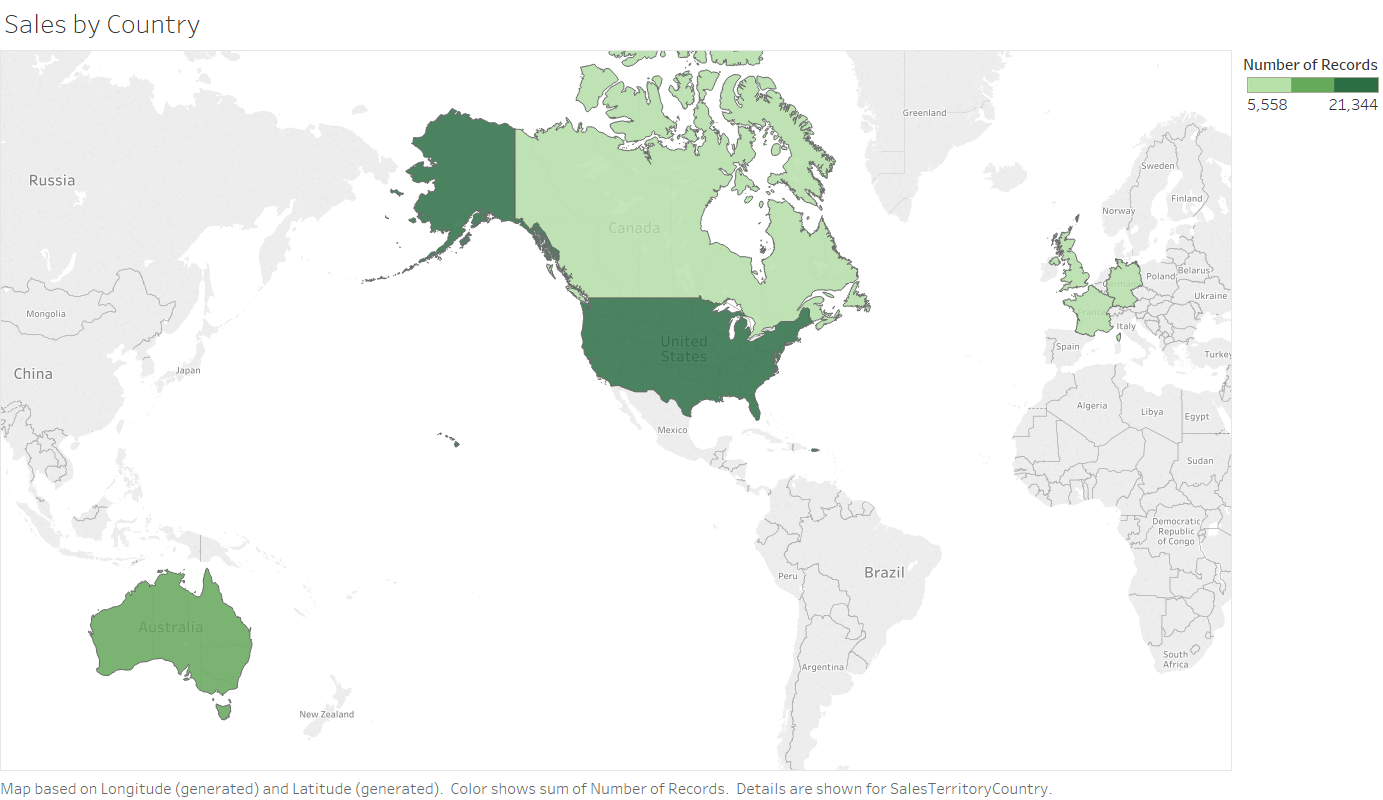
Graph 1)

We can see that most orders are for “Accessories” and all three categories seem to have an uptake of orders around summer time in June. Overall number of orders also seem to increase towards end of year



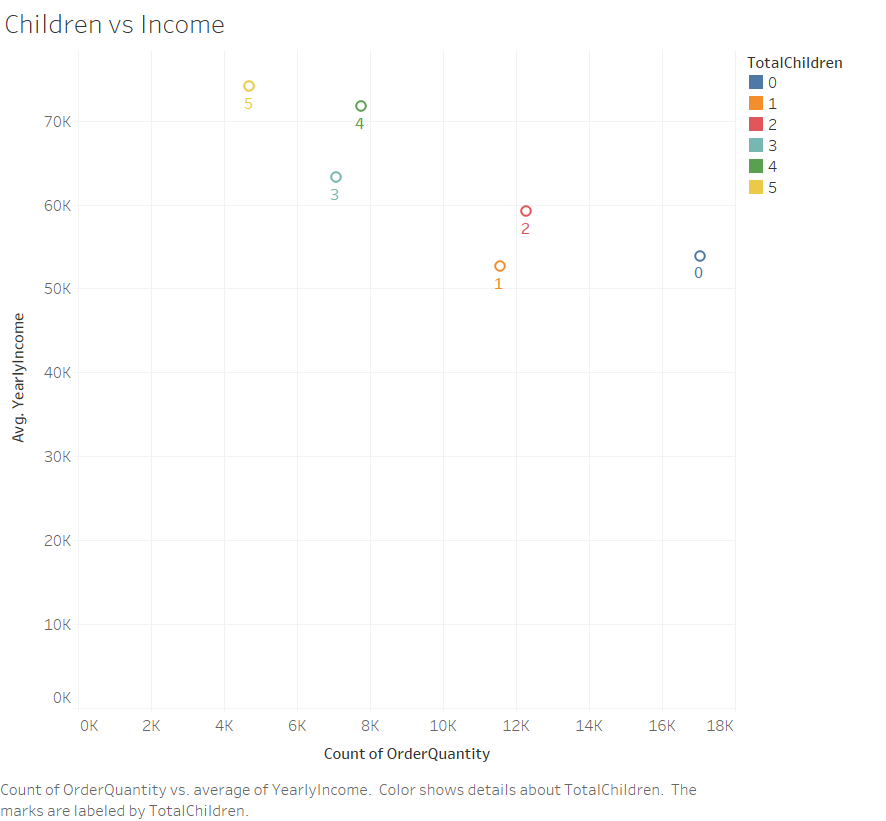
Graph 2)

Highest # of sales in America, second in Australia then Canada and Europe



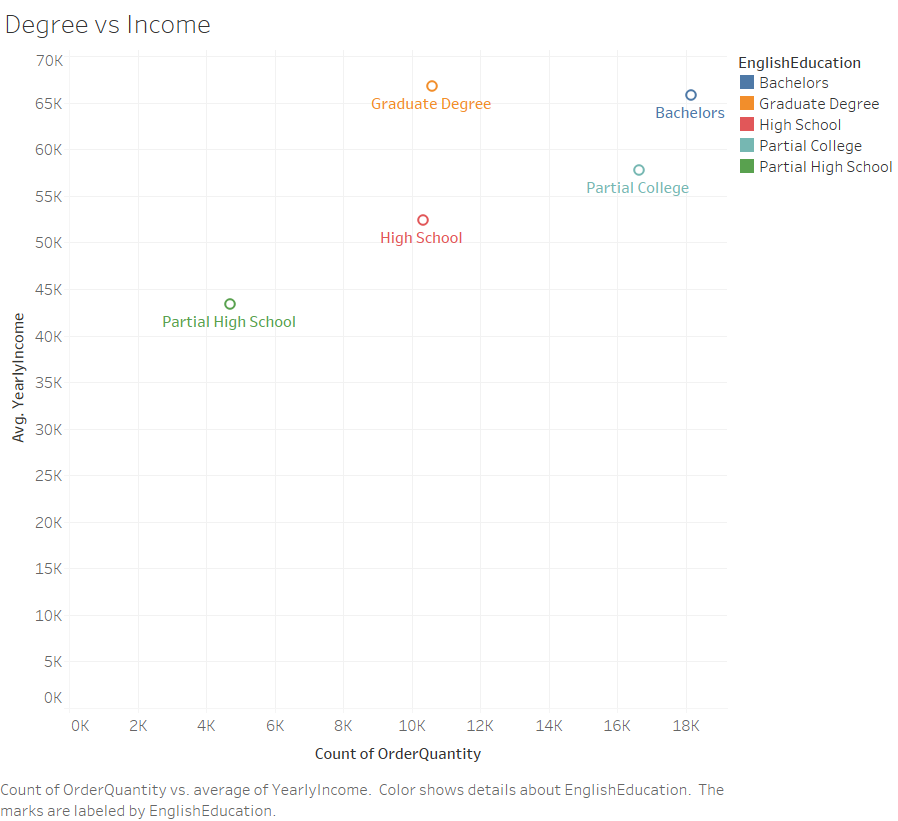
Graph 3)

The more kids a family has, the less they buy even though the higher income they earn on average. On the other hand, people with 2 or less kids earns less but are ordering more



Graph 4)

People with the lowest education background (Partial HS) earn the least and are buying the least. Interestingly, Graduate and Bachelors earn around the same amount on average but Bachelors buy the most



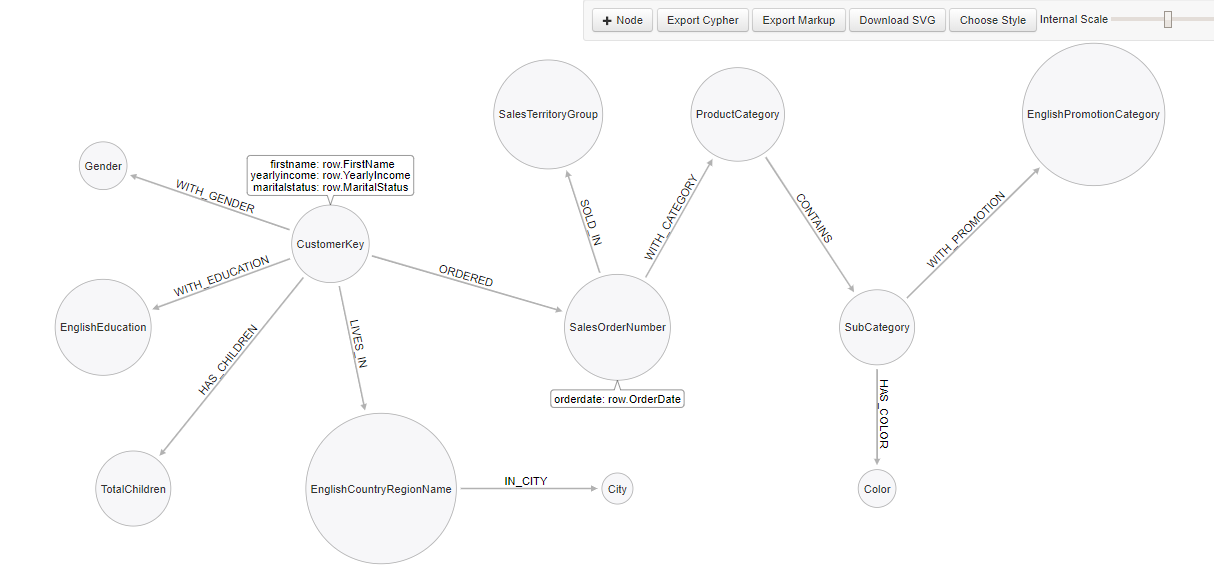
**Columns used:**

1. [OrderDate]
2. [EnglishProductCategoryName]
3. [SalesTerritoryCountry]
4. [OrderQuantity]
5. [YearlyIncome]
6. [TotalChildren]
7. [EnglishEducation]

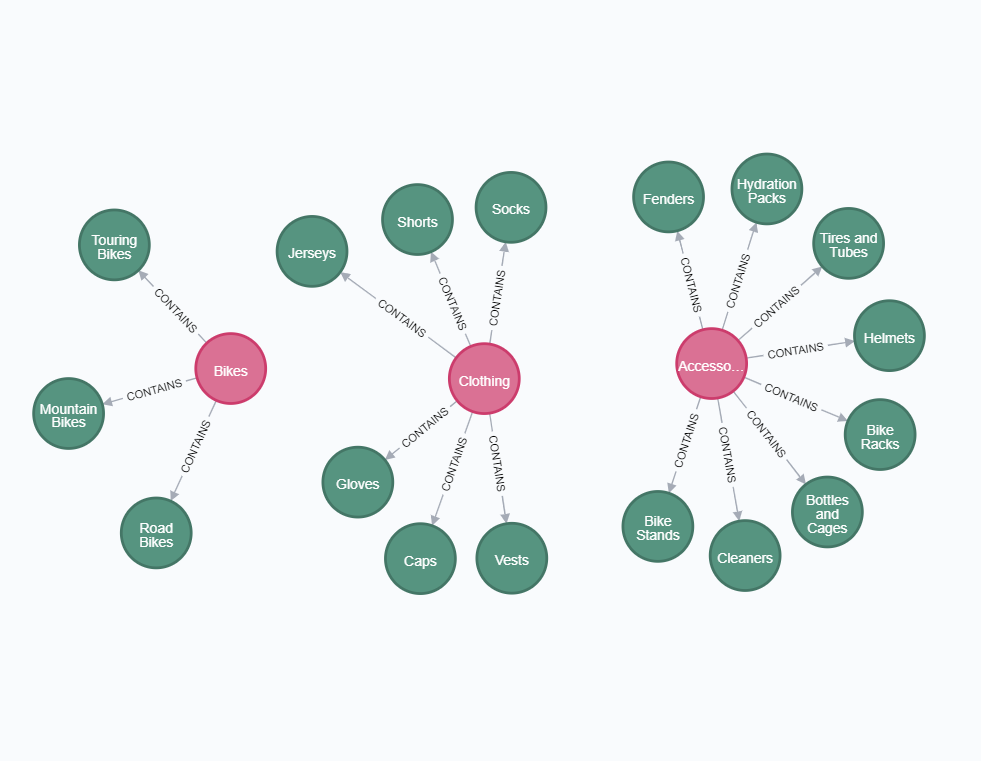
## Q5) Multiple studies have compared the performance of graph and relational database management systems. Use a relational E-R data model of an OLTP database such as Northwind and implement the OLTP database as a graph data model using the Neo4j graph database management system. Identify seven suitable queries and implement the queries in SQL and CYPHER against the relational and graph data models respectively to demonstrate the differences in storing/retrieving data from relational and graph databases.

1. Created relationships for whole data base using:

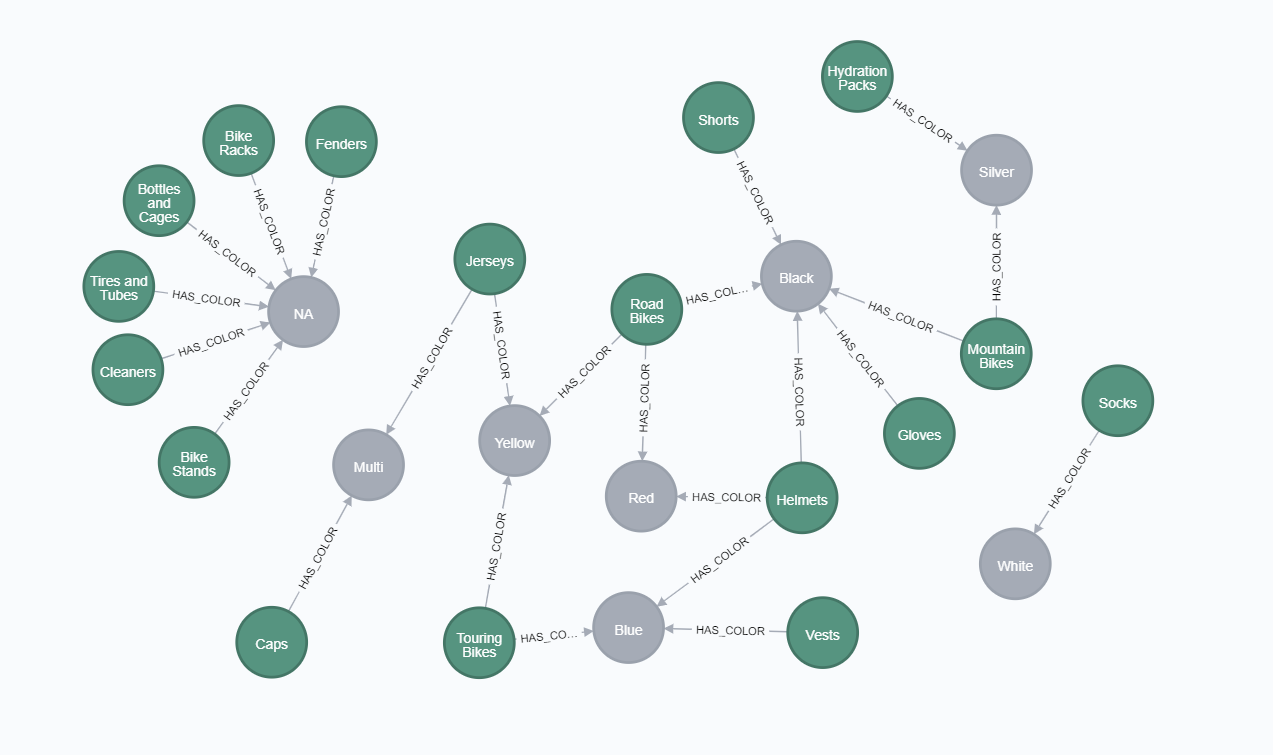
CREATE CONSTRAINT ON (son:SalesOrderNumber) ASSERT son.salesordernumber IS UNIQUE;  
CREATE CONSTRAINT ON (cus:CustomerKey) ASSERT cus.customerkey IS UNIQUE;  
:auto USING PERIODIC COMMIT LOAD CSV WITH HEADERS FROM 'file:///AW2017.csv' AS row  
MERGE (son:SalesOrderNumber {salesordernumber: row.SalesOrderNumber,orderdate:row.OrderDate})  
MERGE (pc:ProductCategory {productcategory: row.EnglishProductCategoryName})  
MERGE (psc:ProductSubCategory {productsubcategory: row.EnglishProductSubcategoryName})  
MERGE (co:Color {color: row.Color})  
MERGE (pr:Promotion {promotion: row.EnglishPromotionCategory})  
MERGE (stg:SalesTerritoryGroup {salesterritorygroup: row.SalesTerritoryGroup})  
MERGE (cus:CustomerKey {customerkey: row.CustomerKey, firstname:row.FirstName, yearlyincome:row.YearlyIncome,maritalstatus: row.MaritalStatus})  
MERGE (crn:CountryRegionName {countryregionname: row.EnglishCountryRegionName})  
MERGE (ci:City {city: row.City})  
MERGE (ge: Gender {gender: row.Gender})  
MERGE (ee: Education {education: row.EnglishEducation})  
MERGE (tc: TotalChildren {totalchildren: row.TotalChildren})  
MERGE (son)-[:WITH\_CATEGORY]->(pc)  
MERGE (son)-[:SOLD\_IN]->(stg)  
MERGE (pc)-[:CONTAINS]->(psc)  
MERGE (psc)-[:HAS\_COLOR]->(co)  
MERGE (psc)-[:WITH\_PROMOTION]->(pr)  
MERGE (cus)-[:ORDERED]->(son)  
MERGE (cus)-[:WITH\_GENDER]->(ge)  
MERGE (cus)-[:WITH\_EDUCATION]->(ee)  
MERGE (cus)-[:HAS\_CHILDREN]->(tc)  
MERGE (cus)-[:LIVES\_IN]->(crn)  
MERGE (crn)-[:HAS\_CITY]->(ci)  
RETURN \*



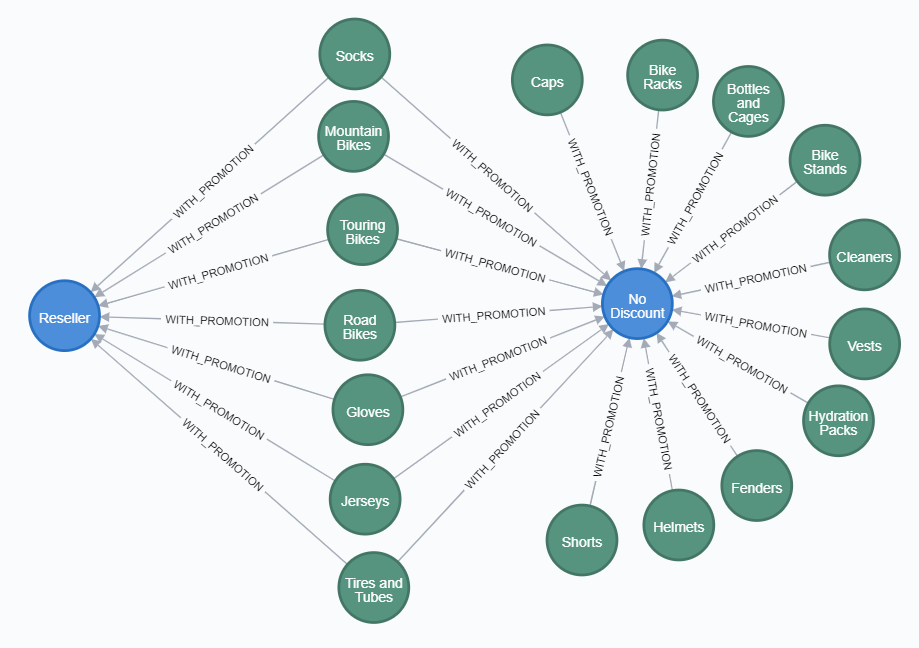
1. Query Product Categories using below code to get relationship between Product category and Product Sub-category:  
   **MATCH (pc:ProductCategory)-[:CONTAINS]->(psc:ProductSubCategory)  
   RETURN pc,psc**I use the following SQL query to get similar results showing a list of Product categories against Sub-Categories:  
   **SELECT DISTINCT EnglishProductSubcategoryName,  
   EnglishProductCategoryName  
   FROM DimProductSubcategory psc  
   LEFT JOIN DimProductCategory ON psc.ProductCategoryKey = DimProductCategory.ProductCategoryKey ORDER BY EnglishProductCategoryName**



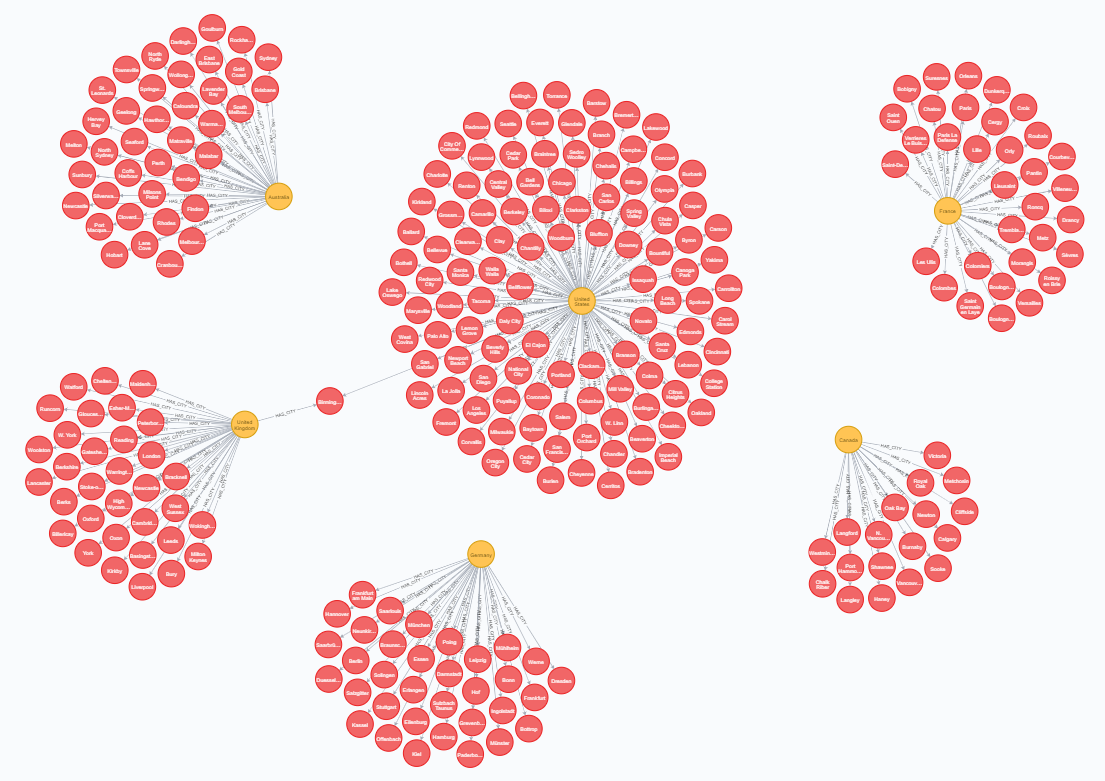
1. Query SubProduct Categories to see what colors exists for them using:  
   **MATCH (psc:ProductSubCategory)-[:HAS\_COLOR]->(co:Color)  
   RETURN psc,co**Similarly in SQL, we can use the below to obtain similar results:  
   **SELECT DISTINCT** **EnglishProductSubcategoryName,  
   Color  
   FROM DimProductSubcategory psc  
   LEFT JOIN DimProduct dp ON psc.ProductSubcategoryKey = dp.ProductSubcategoryKey**



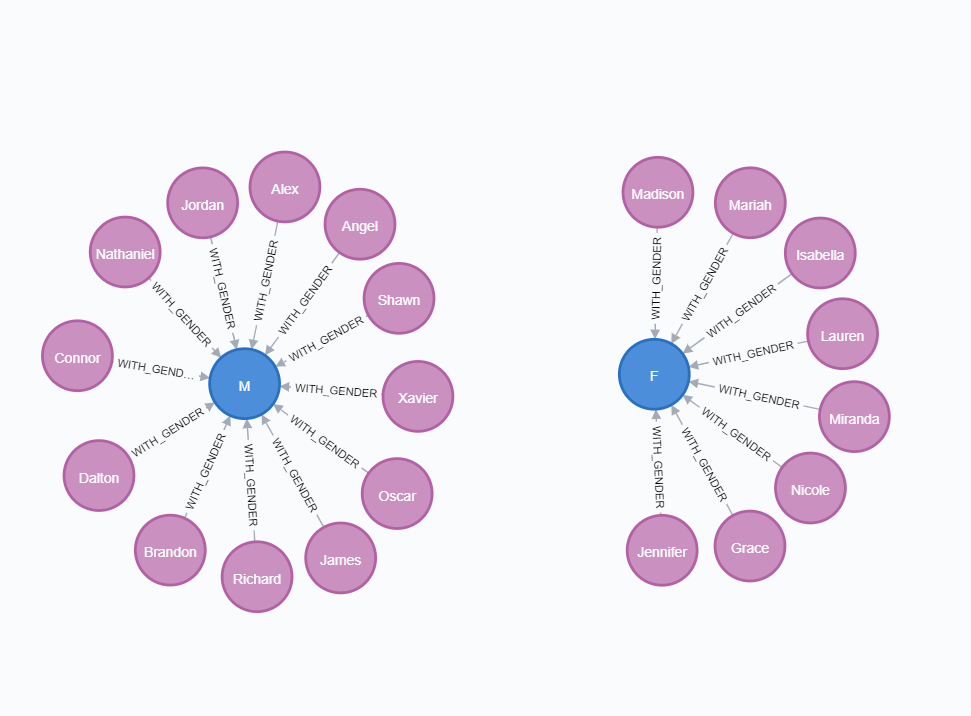
1. Query SubProduct Categories and see if they have promotions:  
   **MATCH (psc:ProductSubCategory)-[:WITH\_PROMOTION]->(pr:Promotion)  
   RETURN psc,pr**Similarly in SQL we can use:   
   **SELECT DISTINCT EnglishProductSubcategoryName,  
   EnglishPromotionCategory  
   FROM DimProductSubcategory psc  
   LEFT JOIN DimProduct dp ON psc.ProductSubcategoryKey = dp.ProductSubcategoryKey  
   LEFT JOIN FactInternetSales fis ON dp.ProductKey = fis.ProductKey  
   LEFT JOIN DimPromotion dpr ON fis.PromotionKey = dpr.PromotionKey ORDER BY EnglishPromotionCategory**

****

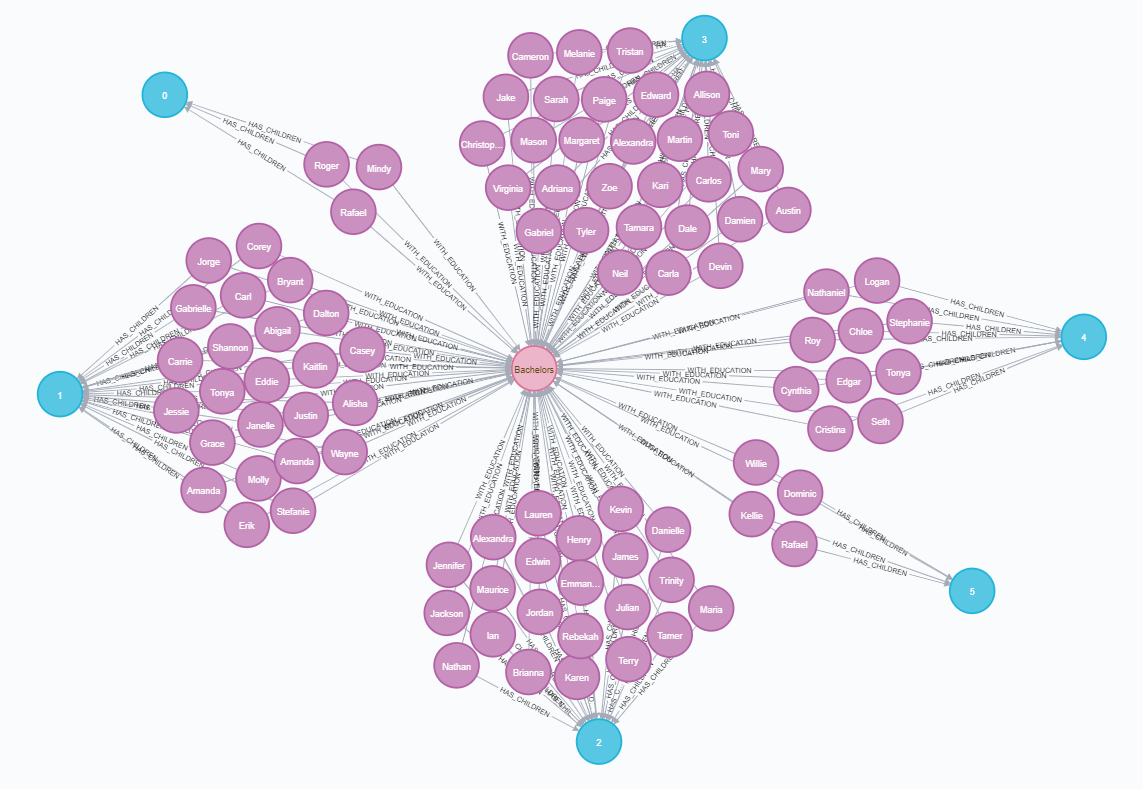
1. Query Countries and see what cities they have:  
   **MATCH (crn:CountryRegionName)-[:HAS\_CITY]->(ci:City)  
   RETURN crn,ci**SQL Equivalent:  
   **SELECT DISTINCT EnglishCountryRegionName,  
   City  
   FROM DimGeography ORDER BY EnglishCountryRegionName**

****

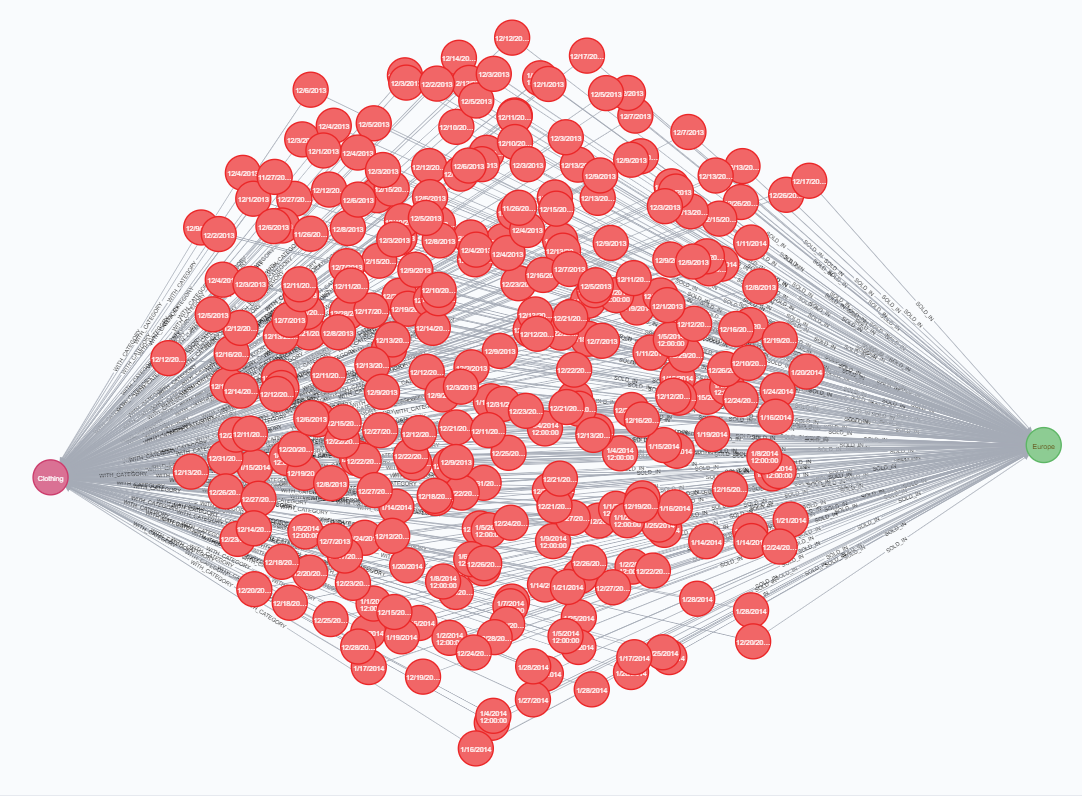
1. Query customers’ genders who live in Canada, Single and have a yearly Income over 80,000 using:  
   **MATCH (crn:CountryRegionName)<-[:LIVES\_IN]-(cus:CustomerKey)-[:WITH\_GENDER]->(ge:Gender)  
   WHERE crn.countryregionname = "Canada"  
   AND cus.maritalstatus = "S"  
   AND cus.yearlyincome > "80000"  
   RETURN cus,ge**SQL Equivalent:  
   **SELECT DISTINCT Gender,  
   FirstName  
   FROM DimCustomer dc  
   LEFT JOIN DimGeography dg ON dc.GeographyKey = dc.GeographyKey  
   WHERE EnglishCountryRegionName = 'Canada'  
   AND MaritalStatus = 'S'  
   AND YearlyIncome > '80000'**

****

1. Query Customers who have a Bachelor's degree, Married and have a yearly income of 100k to see how many children they have by using:  
   **MATCH (ee:Education)<-[:WITH\_EDUCATION]-(cus:CustomerKey)-[:HAS\_CHILDREN]->(tc:TotalChildren)  
   WHERE ee.education = "Bachelors"  
   AND cus.maritalstatus = "M"  
   AND cus.yearlyincome = "100000"  
   RETURN cus,ee,tc**SQL Equivalent:  
   **SELECT DISTINCT TotalChildren,  
   FirstName,  
   EnglishEducation  
   FROM DimCustomer dc  
   WHERE EnglishEducation = 'Bachelors'  
   AND MaritalStatus = 'S'  
   AND YearlyIncome > '100000'**



1. Query Clothing that were sold in Pacific and see individual orders using:  
   **MATCH (pc:ProductCategory)<-[:WITH\_CATEGORY]-(son:SalesOrderNumber)-[:SOLD\_IN]->(stg:SalesTerritoryGroup)  
   WHERE pc.productcategory = "Clothing"  
   AND stg.salesterritorygroup = "Pacific"  
   RETURN pc,son,stg**SQL Equivalent is:  
   **SELECT DISTINCT SalesOrderNumber,  
   EnglishProductCategoryName,  
   SalesTerritoryGroup  
   FROM FactInternetSales fis  
   LEFT JOIN DimSalesTerritory dst ON dst.SalesTerritoryKey = fis.SalesTerritoryKey  
   LEFT JOIN DimProduct dp ON dp.ProductKey = fis.ProductKey  
   LEFT JOIN DimProductSubcategory psc ON psc.ProductSubcategoryKey = dp.ProductSubcategoryKey  
   LEFT JOIN DimProductCategory pc ON pc.ProductCategoryKey = psc.ProductCategoryKey  
   WHERE EnglishProductCategoryName = 'Clothing'  
   AND SalesTerritoryGroup = 'Pacific'**



## Q7) Contribution – Tim Browning

To create a data warehouse, I first had to download the full AdventureWorks database. This file was called AdventureWorks2017.bak (MashaMSFT, n.d.). To load this database into a server I used a program called Azure Data studio. This program allowed me to restore the database from a .bak file. The steps to restore from backup are

1) Select Restore from a .BAK file

2) Choose path to backup

3) Select target database and hit restore

Once the azure data studio has finished restoring the database, I could view all the tables. To develop a suitable schema we first had to chose the subject area. The subject area we chose was “Sales”. The Schema was created in SSMS after creating the data warehouse(Q3). Here are the steps I took to create the schema:

1) Right-click on database design and select “New Database Design”

2) Highlight all tables and press add

3) Here you will see all the tables, On each table, set the primary keys. This can be done by right-clicking on the primary key and selecting “set primary key”

4) The factInternetSales table does not have a primary key as it is the fact table. To link the fact table to the dimension tables drag the primary key from the dimension table to the fact table.

5) A dialog box will appear named “Tables and Columns”. Select the same primary key on the dimension table and the foreign key on the fact table. Press Ok

6) This will create a connection between the two tables.

7) Repeat this for all the other tables ( not all the dimension tables connect to the fact table – some dimension tables connect to other dimension tables e.g DimGeography & DimProductCategory

This schema is presented in Q2, from this question I learned how to create clear star schemas to represent a data warehouse

To create the data warehouse I firstly had to decide on the dimension tables as well as the fact table. I chose DimProduct, DimProductSubcategory, DimProductCategory, DimCurrency, DimDate, DimPromotion, DimSalesTerritory, DimGeography, DimCustomer and FactInternetSales as these represented the subject area “sales”.

I used the tool visual studio to extract the data from the AdventureWorks2017 database and load it into the data warehouse AdventureWorksDWX. This was completed by creating an SSIS package. Here are the steps I took to create the package

1) Create the database AdventureWorks2017

2) Create the tables - DimProduct, DimProductSubcategory, DimProductCategory, DimCurrency, DimDate, DimPromotion, DimSalesTerritory, DimGeography, DimCustomer and FactInternetSales

3) Insert data into the tables using SQL commands. These commands were tested first in SSMS.

4) Run the SSIS package. This will automatically Extract, Transform and Load the data into the data warehouse

From question 3 I learnt how to create a data warehouse using visual studio. The data warehouse can be used to help visualise customer patterns as well as assist with making important business decisions.

The aspect I found most difficult was creating the SQL statements. Some of the tables required multiple joins which meant writing complex SQL code. I overcame this by studying simplified tutorials on W3schools (“SQL Tutorial,” n.d.)

Visual Studio would not execute any statement when I first ran the SSIS package. This was fixed by running the program as an administrator

One error I consistency got was - Build error: “The process cannot access the file because it is being used by another process”. This was fixed by opening the task manager and deleting the visual studio debug process.

Another error I received was a permissions error. This was eliminated by using this SQL command – “ALTER DATABASE AdventureWorksDWX SET SINGLE\_USER WITH ROLLBACK IMMEDIATE”

Throughout this project, I collaborated with Clinton. We both had excellent chemistry and were able to solve each others issues. All in all, this was an extremely useful project to help me become a data analyst and I am very content with my result.

## Clinton Individual Contribution:

For this assignment, Tim and I decided to divide and conquer.  I was responsible for Questions 4 and 5 while Tim 1,2 and 3. We then came together and exchanged lessons learned to ensure we’re both aware of each other’s works.  For question 4, I utilized Visual Studio to produce 4 SSRS reports by writing SQL queries in SSMS:

1) 1st report shows me a list of Product Category Names and what Product Subcategory Names exists within each category

2) 2nd report shows me a list of Country with State and City fields, thus providing the entire geography data blueprint

3) 3rd report shows me a list of product categories alongside any Promotion Types that exists for them

4) 4th report shows me a list of customer’s genders and Product Categories they have purchased

For the second part of question 4, I’ve created 4 graphs in Tableau to draw potential insights from our database by writing a SQL query that joins all useful columns into a single spreadsheet:

1) 1st graph shows me the amount of orders split by Product Category and then aggregated in a monthly manner.

2) 2nd graph shows the amount of orders by countries.

3) 3rd graph shows a scatter plot with axis “amount of sales” against “average yearly income”.  Then I scattered number of children customers have.

4) 4th graph is another scatter plot with the same axis, but this time scattered with customer’s education degrees.

For question 5, I decided to use the same data I wrote in SQL for Tableau and imported that into Neo4j to ensure data consistency.  After importing, I utilized the Arrow tool to draw my nodes & relationships blueprint. Finally, I wrote a CYPHER query with all the nodes and relationships declared in Neo4j as my backbone for the following 7 graphs.  I’ve also included the SQL equivalent in the report document to show the differences:

1) 1st graph shows me the nodes of product category connected to subcategory

2) 2nd graph shows me all the colors that exist for each product subcategory and where overlaps are

3) 3rd graph shows me the types of promotions each subcategory has

4) 4th graph shows me all the countries nodes and city nodes to see what cities exist in each country.  One of the cities exists in both UK and US (same spelling) which is valuable information when assessing data integrity

5) 5th graph shows me the gender of my target customers based on their location, marital status and income

6) 6th graph shows me the total children my target customers have based on education degree, marital status and income

7) 7th graph shows me each individual sale in the Clothing category and sold in the Pacific region

From this exercise, I have learned how to correctly select the right schema and how complex it is to create a database from scratch.  On the visual side of things, I’ve learned how important it is to think through a business problem and visualize them in ways that could prove or disprove hypothesis.  I’ve also learned a lot on the differences between SQL and CYPHER has whether it’s syntax or logic. SQL and CYPHER has a fundamentally different approach to data and the way they made me think about my data.  When dealing with SQL and Tableau I was looking to find insights from my data but when using Neo4j I was looking to find insights about my data. This was a valuable exercise and I’m very happy with the work Tim and I did.

## Bibliography

MashaMSFT, n.d. Install and configure AdventureWorks sample database - SQL Server [WWW Document]. URL https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure (accessed 4.18.20).

Ryan, D., 2014. Script To Populate AdventureWorksDW DimDate. Data Anal. Dustin Ryan. URL https://sqldusty.wordpress.com/2014/07/17/script-to-populate-adventureworksdw-dimdate/ (accessed 4.19.20).

SQL Tutorial [WWW Document], n.d. URL https://www.w3schools.com/sql/ (accessed 4.19.20).