CAB430 - Data and Information Integration

Assignment 1 - Data Warehousing Design

Group Report

Team Members: Michael Lee, Mitch Sugden

Student Numbers: n8433518, n9136321

Weight: 30%

Due Date: Sunday, 24th April 2016

Table of Contents

1. Question 1

1.1 Study of the Car Rental Data Source

The both data sets of the CarRentalDataSource were analyzed using all 7 profiling options of the Data Profiling Task..

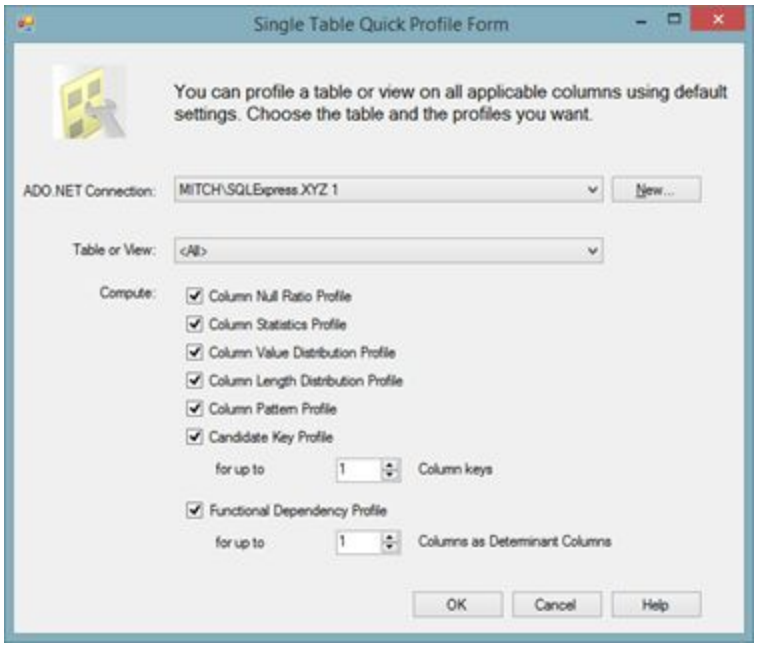


Figure 1: Data Profile Task settings prior to being executed.

Data in Central Database

Inventory of Data Stores

The data from DataInCentralDatabase has been extracted from a central server. We can assume that this data was an accumulation of manual entries by employees (i.e. store, car and order details) and manual input from customers (i.e. customer information). This may have been paperwork or direct SQL queries from a web portal.
Inventory of Data Elements
We have found the dataset to contain the following formats; NVARCHAR(255), float and DATETIME.

Column Name	Number of Distinct Values	Range of Values	Data Type	Domain
Car_BodyType	18	N/A	NVARCHAR(255)	4D SEDAN, 5H HATCHBACK, 2D CABRIOLET, 4D WAGON, 2D CONVERTIBLE, 2D COUPE, 2D HARDTOP, 3D HATCHBACK, 4D VAN, 2D ROADSTER, 4D HATCHBACK, 5D SEDAN, 3D HARDBACK, VAN, 5D WAGON, 2D SEDAN, 5D VAN
Car_Drive	5	N/A	NVARCHAR(255)	RWD, FWD, 4WD, AWD
Car_EngineSize	27	1.3L - 6.1L	NVARCHAR(255)	1.3L, 1.4L, 1.5L, 1.6L. 1.7L, 1.8L, 1.9L, 2.0L, 2.1L, 2.2L, 2.3L, 2.4L, 2.5L, 2.7L, 2.8L, 2.9L, 3.0L, 3.2L, 3.3L, 3.4L, 3.8L, 4.0L, 4.3L, 5.0L, 5.4L, 6.1L
Car_FuelSystem	16	N/A	NVARCHAR(255)	N/A
Car_MakeName	19	N/A	NVARCHAR(255)	N/A
Car_Model	91	N/A	NVARCHAR(255)	N/A
Car_Power	81	37KW - 350KW	NVARCHAR(255)	37KW - 350KW
Car_Series	209	N/A	NVARCHAR(255)	N/A

Car_StandardTransmission	30	N/A	NVARCHAR(255)	N/A
Car_TankCapacity	41	38L - 100L	NVARCHAR(255)	38L - 100L
Car_Wheelbase	91	2230mm - 3200mm	NVARCHAR(255)	2230mm - 3200mm
Car_SeatingCapacity	12	0 - 12	FLOAT	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
Car_ID	N/A	14806 - 15400	FLOAT	14806 - 15400
Car_SeriesYear	N/A	1964 – 2008	FLOAT	1964 – 2008
Car_PriceNew	N/A	2530 – 225600	FLOAT	2530 – 225600
Customer_Addresss	107	N/A	NVARCHAR(255)	N/A
Customer_Brithday	106	08-Jul-30 – 20-Aug-84	DATETIME	08-Jul-30 – 20-Aug-84
Customer_Gender	2	N/A	NVARCHAR(255)	M, F
Customer_Name	104	N/A	NVARCHAR(255)	N/A
Customer_Occupation	5	N/A	NVARCHAR(255)	N/A
Customer_Phone	60	N/A	NVARCHAR(255)	N/A
Customer_ID	N/A	11010 - 11559	FLOAT	11010 - 11559
Pickup_Store_Address	41	N/A	NVARCHAR(255)	N/A
Pickup_Store_City	41	N/A	NVARCHAR(255)	N/A

Pickup_Store_Name	41	N/A	NVARCHAR(255)	N/A
Pickup_Store_Phone	33	N/A	NVARCHAR(255)	N/A
Pickup_Store_State_Name	6	N/A	NVARCHAR(255)	Queensland, New South Wales, Victoria, South Australia, Tasmania
Return_Store_Address	41	N/A	NVARCHAR(255)	N/A
Return_Store_City	41	N/A	NVARCHAR(255)	N/A
Return_Store_Name	41	N/A	NVARCHAR(255)	N/A
Return_Store_Phone	33	N/A	NVARCHAR(255)	N/A
Return_Store_State	6	N/A	NVARCHAR(255)	Queensland, New South Wales, Victoria, South Australia, Tasmania
Order_ID	N/A	3 – 600	FLOAT	3 – 600
Order_CreateDate	N/A	03/07/2005 – 20/02/2007	FLOAT	03/07/2005 – 20/02/2007
Order_PickupDate	N/A	03/07/2005 –21/02/2007	FLOAT	03/07/2005 –21/02/2007
Order_ReturnDate	N/A	09/07/2005 – 17/03/2007	FLOAT	09/07/2005 – 17/03/2007
Order_ReturnStore	N/A	1 – 50	FLOAT	1 – 50

Order_Return_Store_Name	N/A	N/A	NVARCHAR(255)	N/A
Order_Return_Store_Address	N/A	N/A	FLOAT	N/A
Order_PickupStore	N/A	1 - 50	FLOAT	1 – 50

Core Data Elements

Customer ID
Order_ID
Car_ID
Pickup_StoreName
Return_StoreName

Grouped Data Elements

There are 5 distinct sets of elements in this data set.

- Car details
- Order details
- Customer details
- Pickup details
- Return details

Column Statics Profiles

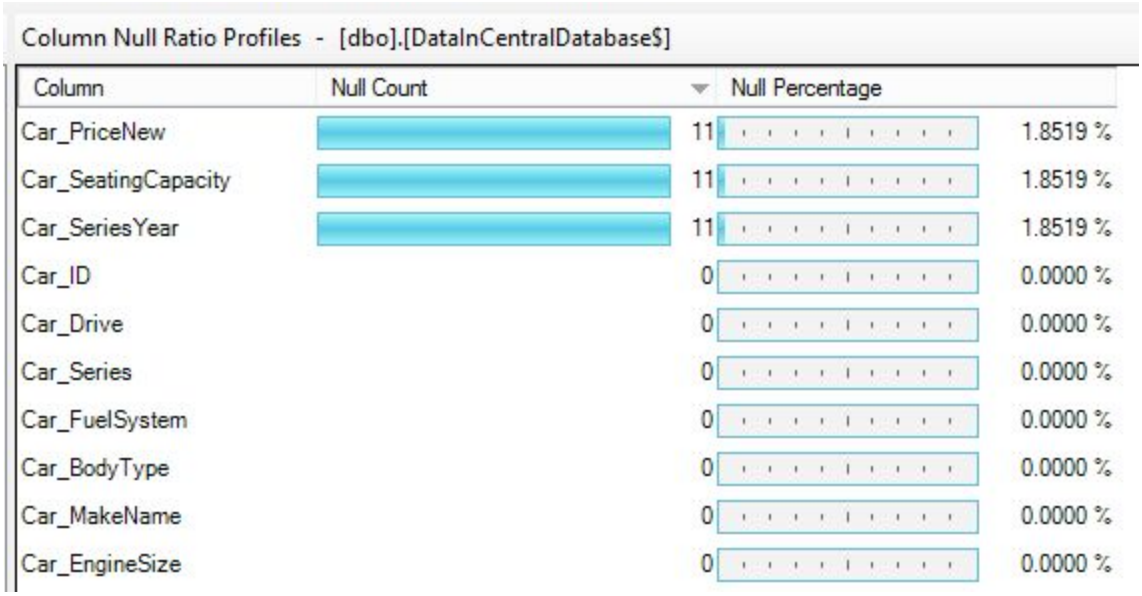
Column Statistics Profiles - [dbo].[DataInCentralDatabase$]

Column	Minimum	Maximum	Mean	Standard Deviation
Car_ID	14806	15400	15092.0824915...	167.853436138281
Car_PriceNew	2530	225600	49547.4974271...	26789.3055399624
Car_SeatingCapacity	0	12	4.89536878216...	0.810117726141688
Car_SeriesYear	1964	2008	1999.91595197...	6.24745322336242
Customer_Brithday	08-Jul-30 12:00:00 AM	20-Aug-84 12:00:00 AM		
Customer_ID	11010	11559	11283.7171717...	167.107097279031
Order_CreateDate	20050703	20070220	20058528.0858...	5646.93283572405
Order_ID	3	600	301.868686868...	172.727377872258
Order_PickupDate	20050703	20070221	20058528.6329...	5646.93510625685
Order_PickupStore	1	50	19.8468013468...	9.76621150062884
Order_ReturnDate	20050709	20070317	20058951.3888...	5851.18505479484
Order_ReturnStore	1	50	21.2424242424...	9.44549229859298

Figure 2: Column statistics of the central database, showing the minimum and maximum values.

Null values

There are 11 null values for the following Columns; Car_PriceNew, Car_SeatingCapacity, Car_StandardTransmission. This is not a significant number of null values compared to the total size of the dataset, therefore, we will ignore them.



Column Null Ratio Profiles - [dbo].[DataInCentralDatabase$]

Column	Null Count	Null Percentage
Car_PriceNew	11	1.8519 %
Car_SeatingCapacity	11	1.8519 %
Car_SeriesYear	11	1.8519 %
Car_ID	0	0.0000 %
Car_Drive	0	0.0000 %
Car_Series	0	0.0000 %
Car_FuelSystem	0	0.0000 %
Car_BodyType	0	0.0000 %
Car_MakeName	0	0.0000 %
Car_EngineSize	0	0.0000 %

Figure 3: Example of Null Ratio Profiles from the Data Profiling Tool. This is used to identify NULL values in large data sets.

Data in Store

Column Name	Number of Distinct Values	Range of Values	Data Type	Domain
Store_ID	41	1 - 50	FLOAT	1 - 50
Store_Name	41	N/A	NVARCHAR(255)	
Store_Address	41	N/A	NVARCHAR(255)	
Store_Phone	33	N/A	NVARCHAR(255)	

Store_City	41	N/A	NVARCHAR(255)	
Store_State_Name	6	N/A	NVARCHAR(255)	Queensland, New South Wales, Victoria, South Australia, Tasmania
Order_ID	N/A	3 – 600	FLOAT	3 - 600
Order_CreateDate	N/A	03/07/2005 – 20/02/2007	FLOAT	03/07/2005 – 20/02/2007
Pickup_Or_Return	2	N/A	NVARCHAR(255)	Return, Pickup
Pickup_Or_Return_Date	N/A	N/A	FLOAT	
Customer_ID	N/A	11010 - 11559	FLOAT	11010 - 11559
Customer_Name	104	N/A	NVARCHAR(255)	N/A
Customer_Phone	60	N/A	NVARCHAR(255)	N/A
Customer_Address	107	N/A	NVARCHAR(255)	N/A
Customer_Birthday	106	08-Jul-30 – 20-Aug-84	DATETIME	08-Jul-30 – 20-Aug-84
Customer_Occupation	5	N/A	NVARCHAR(255)	N/A
Customer_Gender	2	N/A	NVARCHAR(255)	M, F
Car_ID	N/A	14806 - 15400	FLOAT	14806 - 15400
Car_MakeName	19	N/A	NVARCHAR(255)	N/A

Car_Model	91	N/A	NVARCHAR(255)	N/A
Car_Series	209	N/A	NVARCHAR(255)	N/A
Car_SeriesYear	N/A	1964 - 2008	FLOAT	1964 - 2008
Car_PriceNew	N/A	2530 - 225600	FLOAT	2530 - 225600
Car_EngineSize	27	1.3L - 6.1L	NVARCHAR(255)	1.3L, 1.4L, 1.5L, 1.6L. 1.7L, 1.8L, 1.9L, 2.0L, 2.1L, 2.2L, 2.3L, 2.4L, 2.5L, 2.7L, 2.8L, 2.9L, 3.0L, 3.2L, 3.3L, 3.4L, 3.8L, 4.0L, 4.3L, 5.0L, 5.4L, 6.1L
Car_FuelSystem	16	N/A	NVARCHAR(255)	N/A
Car_TankCapacity	41	38L - 100L	NVARCHAR(255)	38L - 100L
Car_Power	81	37KW - 350KW	NVARCHAR(255)	37KW - 350KW
Car_SeatingCapacity	12	0 – 12	FLOAT	0 - 12
Car_StandardTransmission	30	N/A	NVARCHAR(255)	N/A
Car_BodyType	18	N/A	NVARCHAR(255)	4D SEDAN, 5H HATCHBACK, 2D CABRIOLET, 4D WAGON, 2D CONVERTIBLE, 2D COUPE, 2D HARDTOP, 3D HATCHBACK, 4D VAN, 2D ROADSTER, 4D HATCHBACK, 5D SEDAN,

				3D HARDBACK, VAN, 5D WAGON, 2D SEDAN, 5D VAN
Car_Drive	5	N/A	NVARCHAR(255)	N/A
Car_Wheelbase	91	2230mm - 3200mm	NVARCHAR(255)	2230mm - 3200mm

Core Data Elements

Customer ID
Order_ID
Car_ID
Pickup_Or_Return
Pickup_Or_Return_Date
Store_ID

Grouped Data Elements

There are 5 distinct sets of elements in this data set.

- Car details
- Order details
- Customer details
- Pickup / Return Details
- Store details

1.2 Description of Chosen Data

For our project we will be using the following data elements;

Candidate Key	**Hierarchy Elements**
Order_ID	Pickup_Or_Return Pickup_Or_ReturnDate Order_CreateDate

Customer_ID	Customer_Name Customer_Phone Customer_Address Customer_Birthday Customer_Gender
Car_ID	Car_Make Car_Model Car_series Car_SeriesYear Car_PriceNew Car_EngineSize Car_FuelSystem Car_TankCapacity Car_Power Car_SeatingCapacity Car_StandardTransmission Car_BodyType
Store_ID	Store_Name Store_Address Store_City Store_State

Data Profile for our Data Warehouse

Column Name	Number of Distinct Values	Range of Values	Data Type	Domain
Order_ID	N/A	3 - 600	INT	>= 1
Customer_ID	N/A	11010 - 11559	INT	>= 1
Car_ID	N/A	14806 - 15400	INT	>= 1
Store_ID	41	1 - 50	INT	>= 1
Pickup_Or_Return	2	Pickup / Return	VARCHAR(10)	Pickup, Return

Pickup_Or_ReturnDate	N/A	03/07/2005 - 17/03/2007	DATETIME	>= 03/05/2005
Order_CreateDate	N/A	03/05/2005 - 20/02/200	DATETIME	>= 03/05/2005
Customer_Name	104	N/A	VARCHAR(255)	N/A
Customer_Phone	60	1 (11) 500 555-011* - 994-555-015*	VARCHAR(25)	9999999 - 9999999999999
Customer_Address	107	N/A	VARCHAR(255)	N/A
Customer_Birthday	106	08-Jul-30 – 20-Aug-84	DATE	
Customer_Gender	2	M / F	VARCHAR(10)	M, F
Car_Make	19	N/A	VARCHAR(50)	N/A
Car_Model	91	N/A	VARCHAR(50)	N/A
Car_series	209	N/A	VARCHAR(100)	N/A
Car_SeriesYear	N/A	1964 - 2016	INT	>= 1900
Car_PriceNew	N/A	2530 - 225600	INT	>= 0
Car_EngineSize	27	1.3L - 6.1L	VARCHAR(10)	N/A
Car_FuelSystem	16	N/A	VARCHAR(50)	N/A
Car_TankCapacity	41	38L - 100L	VARCHAR(10)	>= 0L
Car_Power	81	37KW - 350KW	VARCHAR(10)	>= 0KW
Car_SeatingCapacity	12	0 - 12	INT	0 - 20
Car_StandardTransmission	30	N/A	VARCHAR(10)	N/A

Car_BodyType	18	N/A	VARCHAR(50)	N/A
Car_Drive	5	N/A	VARCHAR(10)	N/A
Store_Address	41	N/A	VARCHAR(255)	N/A
Store_City	41	N/A	VARCHAR(255)	N/A
Store_State	6	N/A	VARCHAR(255)	Queensland, New South Wales, Victoria, South Australia, Tasmania

1.3 Fact Table Design

Our fact table is made up of the foreign keys of the respective dimension tables to ensure referential integrity. These elements also form the concatenated key.

Fact Table Elements;

Fact Table Element	**Dimension Elements**
Order_ID (Foreign Key)	**Order_ID (Primary Key)** Pickup_Or_Return Pickup_Or_ReturnDate Order_CreateDate
Customer_ID (Foreign Key)	**Customer_ID (Primary Key)** Customer_Name Customer_Phone Customer_Address Customer_Birthday Customer_Gender
Car_ID (Foreign Key)	**Car_ID (Primary Key)** Car_Make Car_Model Car_series

	Car_SeriesYear Car_PriceNew Car_EngineSize Car_FuelSystem Car_TankCapacity Car_Power Car_SeatingCapacity Car_StandardTransmission Car_BodyType
Store_ID (Foreign Key)	**Store_ID (Primary Key)** Store_Address Store_City Store_State
Pickup_Or_Return	
Pickup_Or_Returndate	

1.4 Proposed Data Warehouse Schema

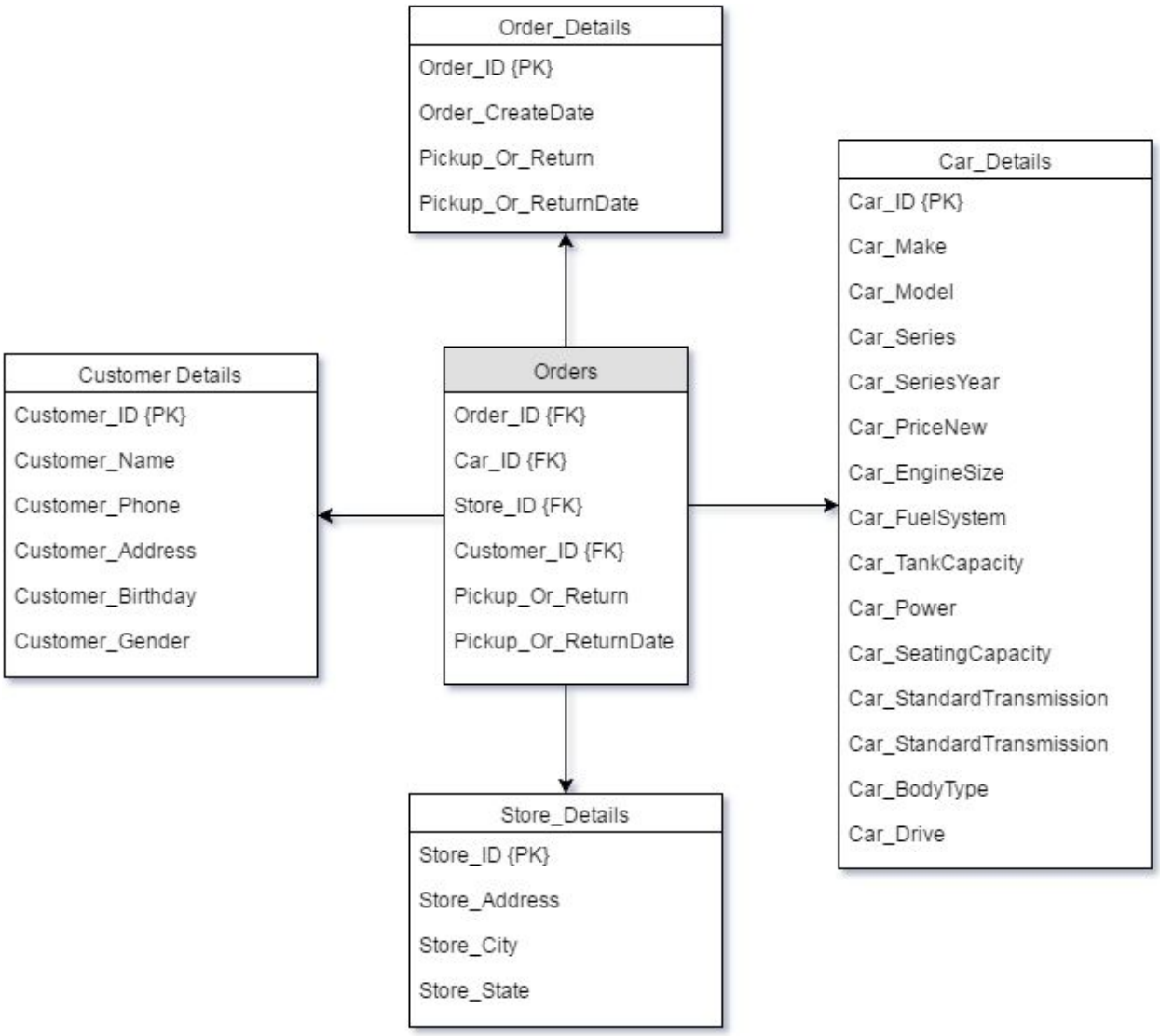


Figure 4: ER Diagram of our proposed data warehouse.

We have chosen to use the *Star Schema* as we have a small number of dimensions. This means that we were able to have minimal redundancy, and an easy to query schema.

2. Creating the Database

2.1 Scripts for Table Creation

USE "XYZ Database"

```
CREATE TABLE Lee_Order_Details (
	Order_ID INT NOT NULL PRIMARY KEY,
	Order_CreateDate DATE NOT NULL,
	Pickup_Or_Return VARCHAR(10) NOT NULL,
	Pickup_Or_ReturnDate DATE NOT NULL
);
CREATE TABLE Lee_Store_Details (
	Store_ID INT NOT NULL PRIMARY KEY,
	Store_Address VARCHAR(255) NOT NULL,
	Store_City VARCHAR(255) NOT NULL,
	Store_State VARCHAR(255) NOT NULL
);
CREATE TABLE Lee_Car_Details (
	Car_ID INT NOT NULL PRIMARY KEY,
	Car_Make VARCHAR(50) NOT NULL,
	Car_Model VARCHAR(50) NOT NULL,
	Car_Series VARCHAR(100) NOT NULL,
	Car_SeriesYear INT NOT NULL,
	Car_PriceNew INT NOT NULL,
	Car_EngineSize VARCHAR(10) NOT NULL,
	Car_FuelSystem VARCHAR(50) NOT NULL,
	Car_TankCapacity VARCHAR(10) NOT NULL,
	Car_Power VARCHAR(10) NOT NULL,
	Car_SeatingCapacity INT NOT NULL,
	Car_StandardTransmission VARCHAR(10) NOT NULL,
	Car_BodyType VARCHAR(50) NOT NULL,
	Car_Drive VARCHAR(10) NOT NULL
);
CREATE TABLE Lee_Customer_Details (
	Customer_ID int NOT NULL PRIMARY KEY,
	Customer_Name VARCHAR(255) NOT NULL,
	Customer_Phone  VARCHAR(25) NOT NULL,
	Customer_Address VARCHAR(255) NOT NULL,
	Customer_Birthday DATETIME NOT NULL,
	Customer_Gender VARCHAR(10) NOT NULL
);
GO
CREATE TABLE Lee_Fact_Orders (
	Order_ID int NOT NULL,
	Car_ID int NOT NULL,
	Store_ID int NOT NULL,
	Customer_ID int NOT NULL,
	Pickup_Or_Return VARCHAR(10) NOT NULL,
```

```
        Pickup_Or_Returndate DATETIME NOT NULL,
        FOREIGN KEY (Order_ID) REFERENCES Lee_Order_Details(Order_ID),
        FOREIGN KEY (Car_ID) REFERENCES Lee_Car_Details(Car_ID),
        FOREIGN KEY (Store_ID) REFERENCES Lee_Store_Details(Store_ID),
        FOREIGN KEY (Customer_ID) REFERENCES Lee_Customer_details(Customer_ID)
);
```

2.2 ER Diagram

This is the ER diagram created by the SQL statements in section 2.1. This shows a clear *Star Schema* with a single fact table in the centre that connects to single tiered dimension tables.

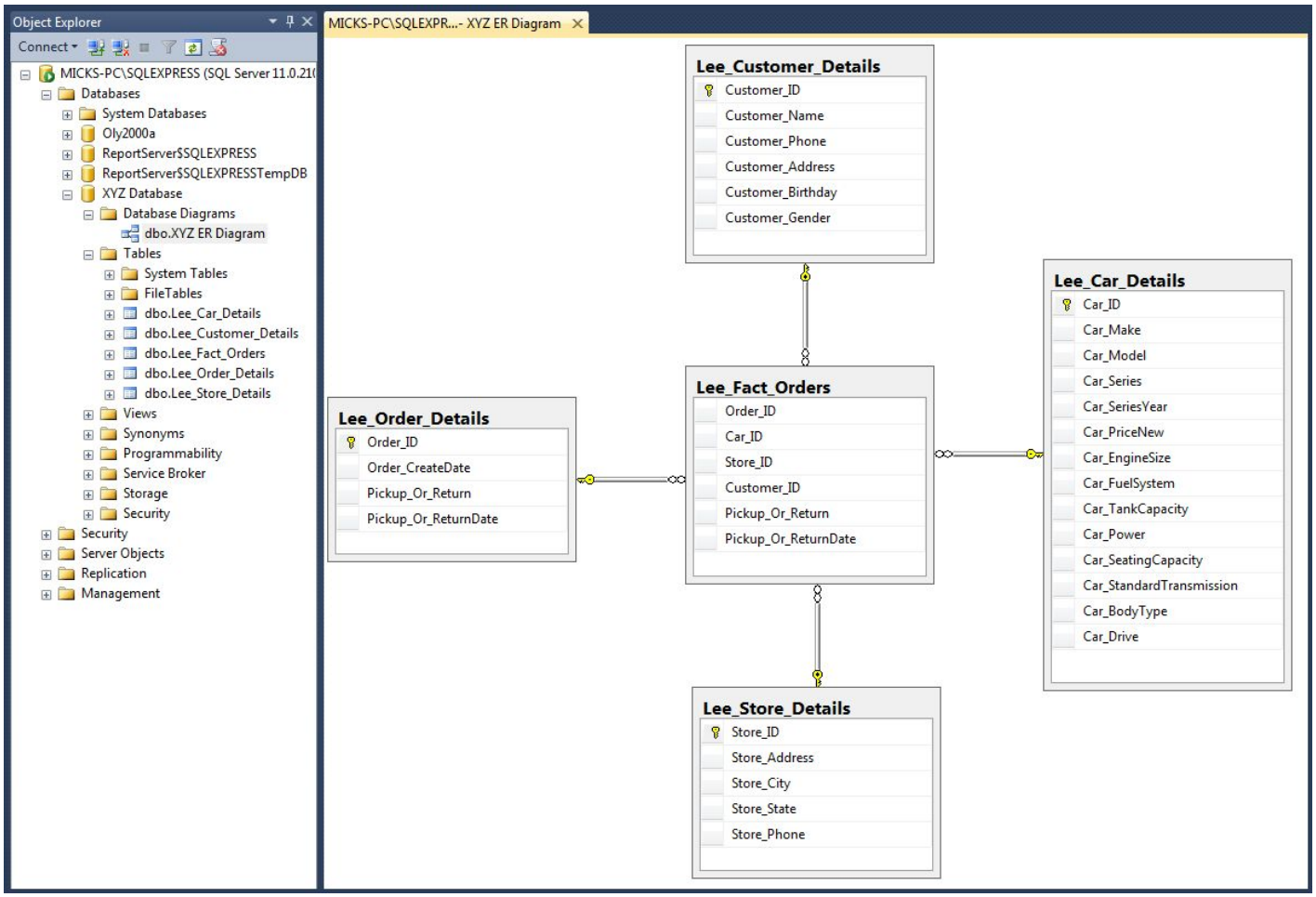


Figure 5: Screen capture of the completed ER diagram of our star schema. This was produced in SQL Server Management Studio 2012.

3. ETL Design Process

3.1 ETL Overview

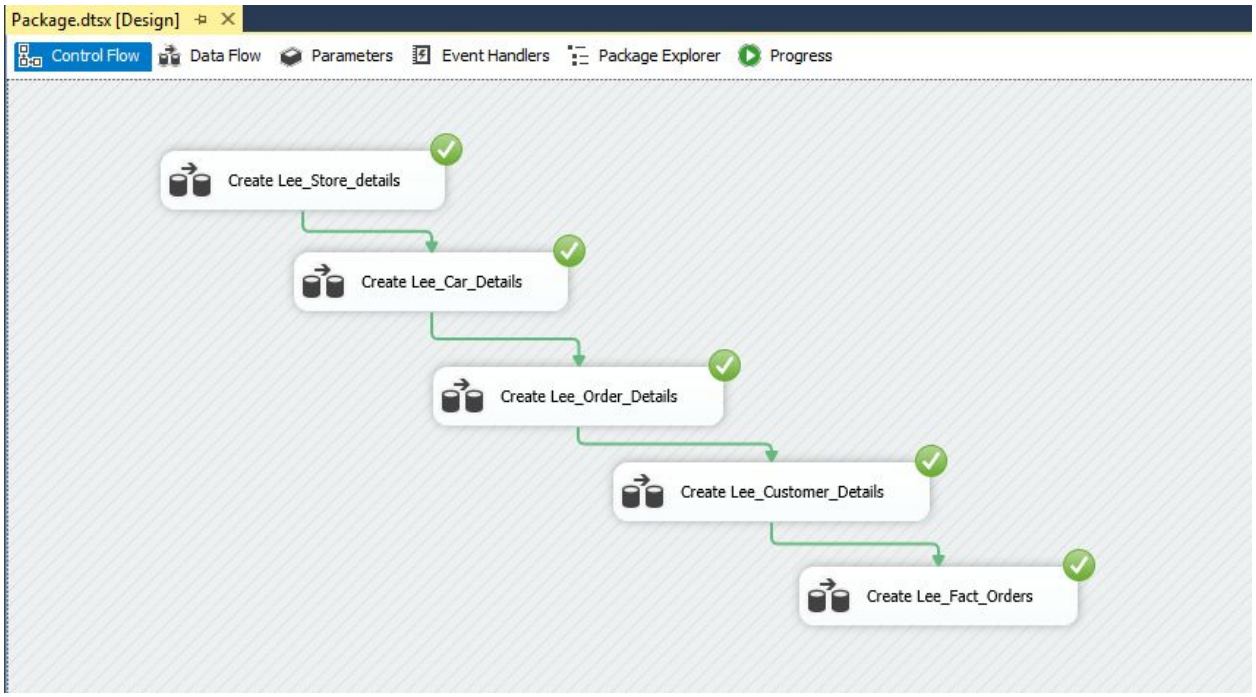


Figure 6: Screen capture of our whole ETL process. This was created using the Control Flow tab within a SQL Server Data Tools 2015 project.

For our ETL process imports data into the tables of the ER diagram shown in section 2. To do this we have created 5 separate Data Flow Tasks, one for each Table of our schema.

Create Lee_Store_Details

This task takes information about the store's from the store database and inserts them into our Lee_Store_Details table. The source data set contains multiple tuples with duplicate store information data, so we had to handle this in our Data Flow. The data set also had some *NULL* values. As our schema does not take *NULL* for any element we had to remove these values before loading the data into SQL.

Create Lee_Car_Details

For our car details dimension we only want to store the unique cars that XYZ have for renting. This task retrieves car information from the store data set and removes any duplicate value i.e. where the same car has been rented to more than one customer. It then removes some *NULL* values before loading the data into SQL.

Create Lee_Order_Details

As the store data set contains order dates as an integer, we needed to convert them to a Date format before loading into SQL. The main task in this Data Flow is converting the date columns in the data set to an SQL Date format. Once this is done, the data is checked for duplicates, sorted and loaded into SQL.

Create Lee_Customer_Details

Our customer task is quite simple. We convert the data types from the raw source to a non-unicode format, ensure that there are no duplicate Customer_ID's (Primary Key) and load the data into SQL.

Create Lee_Fact Orders

This task creates our fact table. Our fact data is loaded last as it has foreign key dependencies. The fact table also uses a date value that is converted to Date. Fact table allows duplicates of each attribute so we are able to import data from each tuple of the store data set.

3.2 ETL Dimension Processing

Dimension 1: Lee_Order_Details

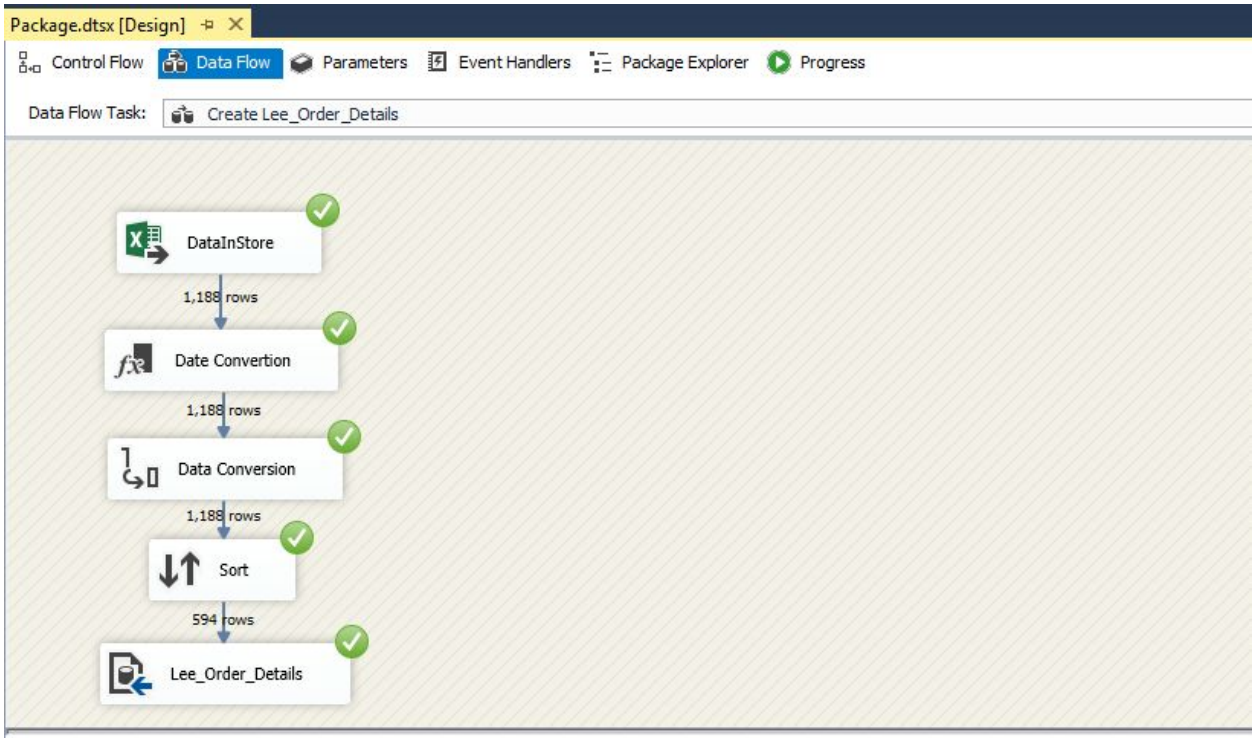


Figure 7: Completed tasks from the Create Lee_Order_Details Data Flow task.

In this Control Flow task we have used the *Derived Column* transformation,

renamed to "*Date Conversion*" to convert the integer date to an SQL Date. The *Derived Column* can be used to create new column values by applying SQL expressions. This was necessary as the source data stored some date values as float data types.

To perform this transformation successfully, we had to complete the following steps.

1. First we changed the Output Column DataType for both *Order_CreateDate* and *Pickup_Or_Return_Date* from a float a **string[DT_STR]**, this will allow us to break the string up into year, month and day.

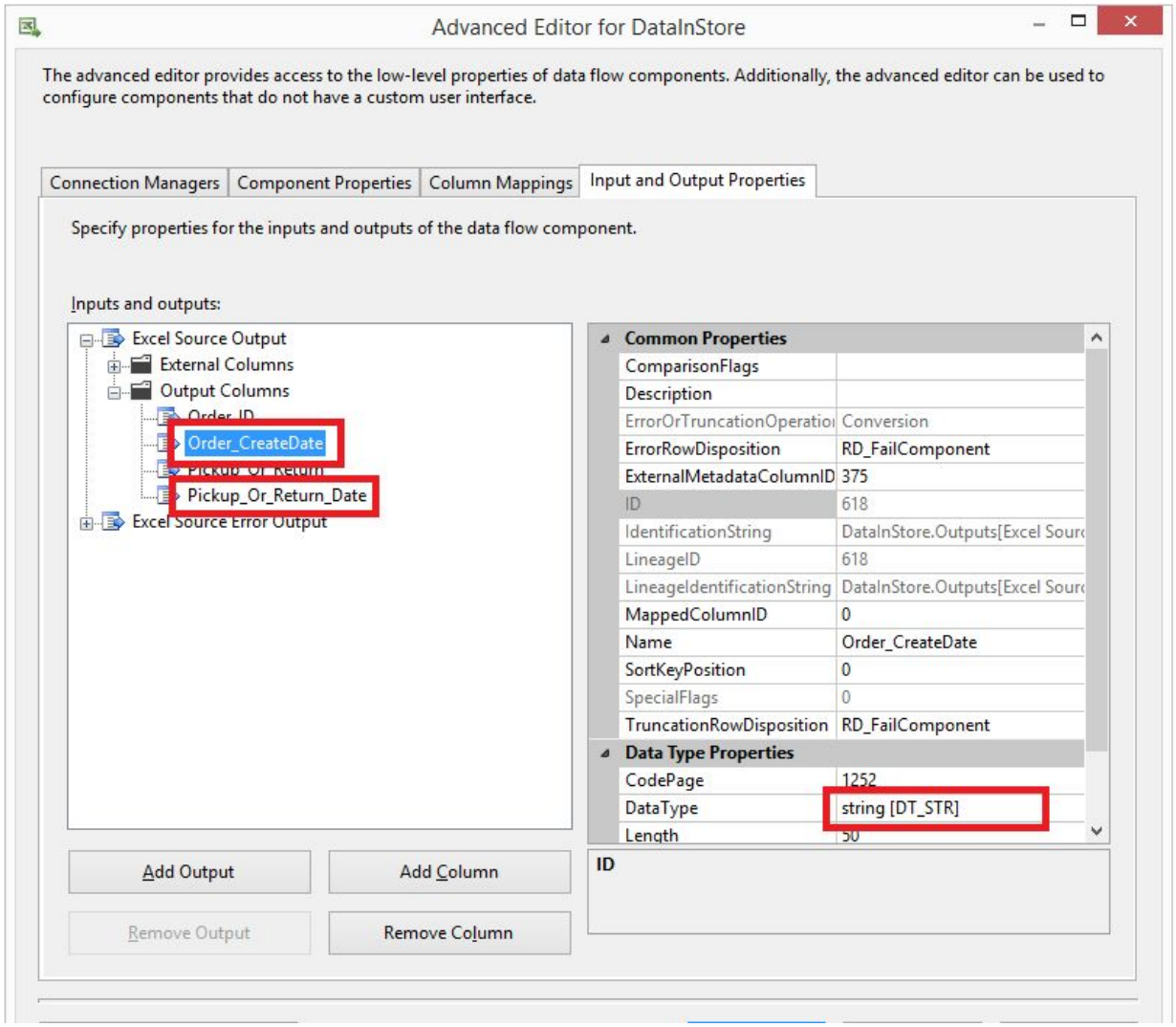


Figure 8: This figure shows the **Show Advanced Editor…** window of the DataInStore excel source.

2. Now that we have a **string[DT_STR]** as input for our *Derived Column* transformation we can add a rule for each attribute, *Order_CreateDate* and *Pickup_Or_Return_Date*, to convert it to a string value that is readable by the SQL Date data type.

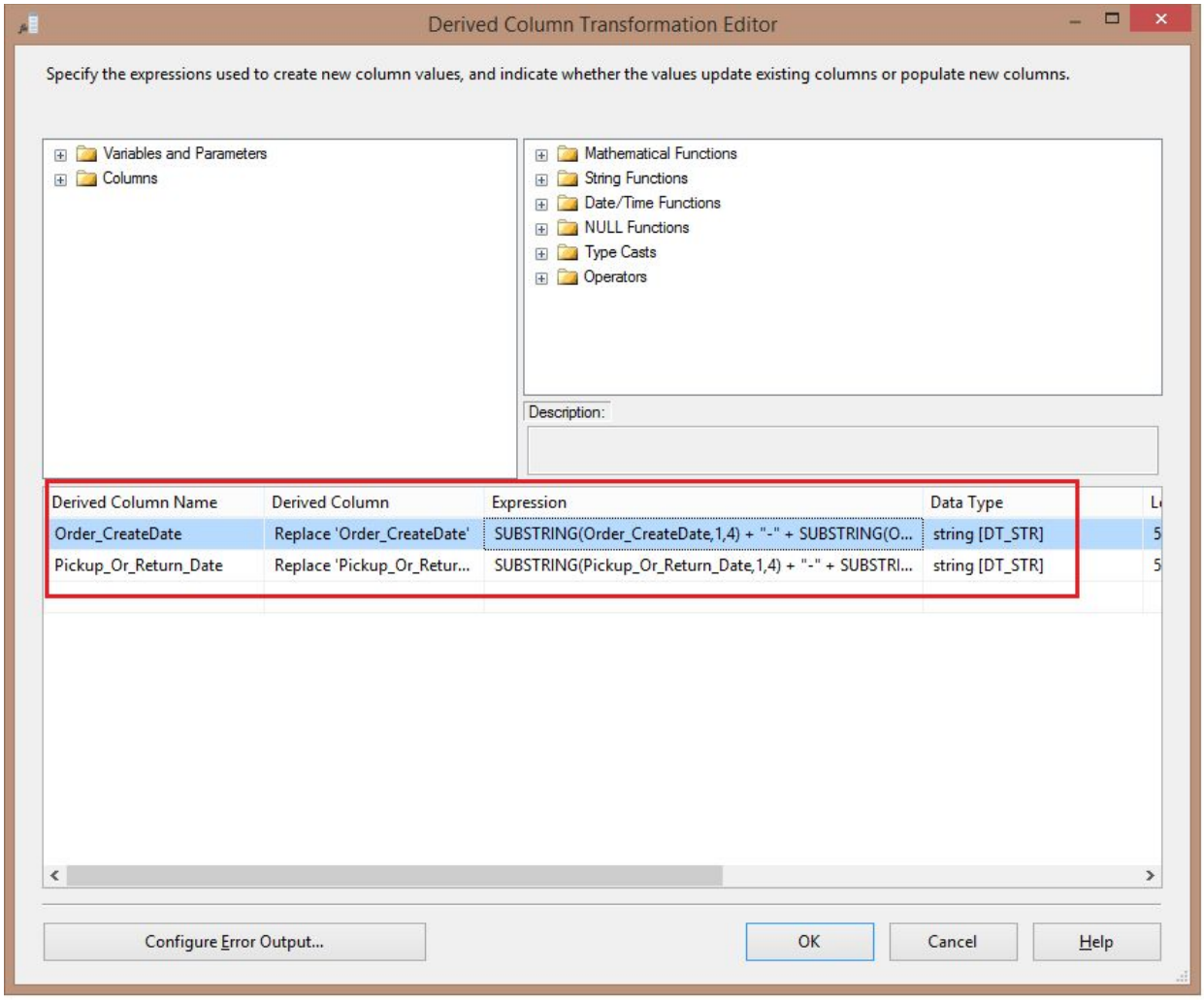


Figure 9: This screen capture shows the **Edit** screen of the *Derived Column* node. Highlighted are the expressions used for the corresponding column names.

The expression used it:
SUBSTRING(Order_CreateDate,1,4) + "-" + SUBSTRING(Order_CreateDate,5,2) + "-" + SUBSTRING(Order_CreateDate,7,2)

This converts a string value from “20161102” to “2016-11-02”.

3. The data now flows through the *Data Conversion* node which converts the data to **database date[DT_DBDate]** which can be imported into SQL.

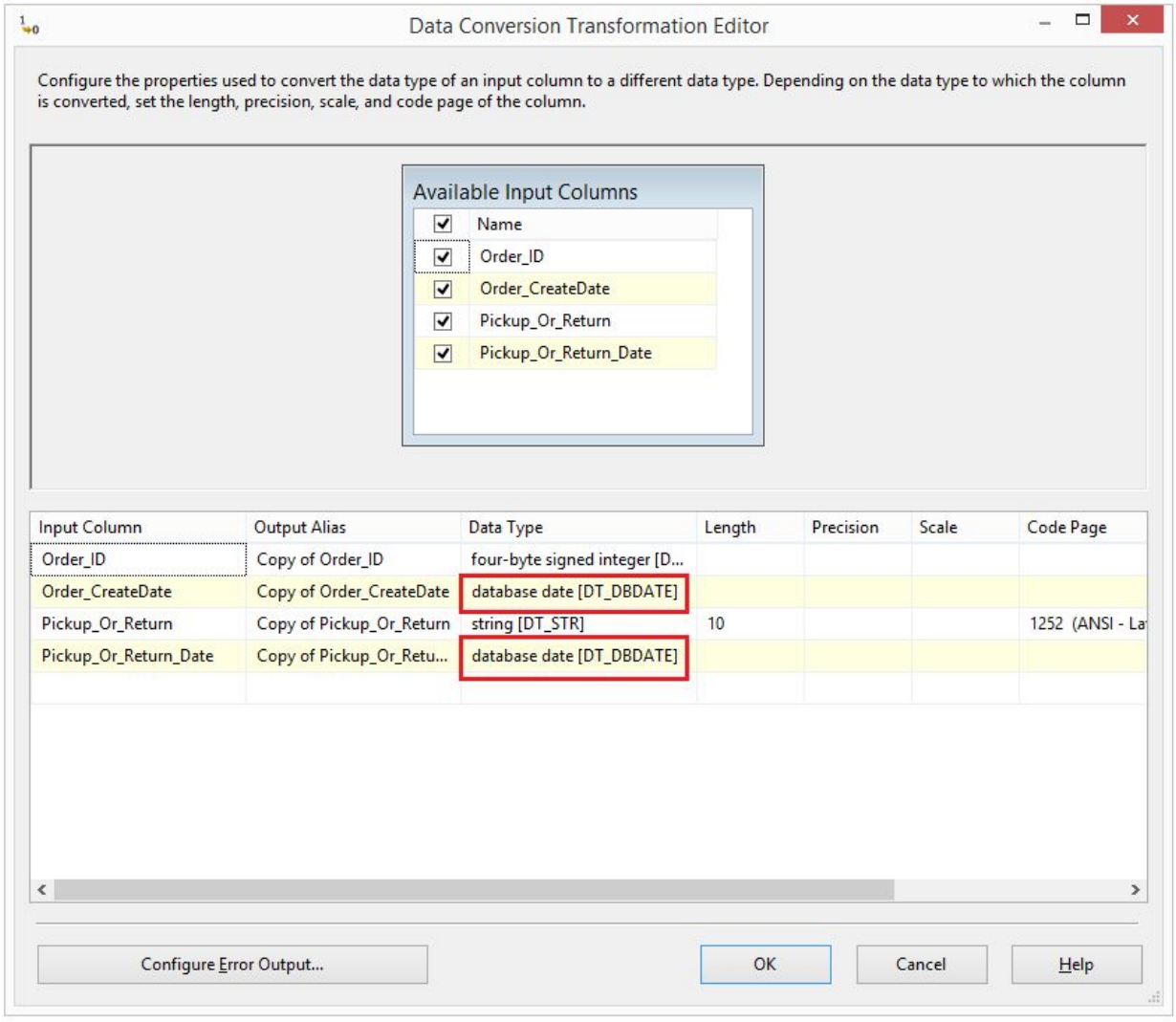


Figure 10: This screen capture shows the **Edit** screen of the *Data Conversion* node. Highlighted are the values used to represent the new data type of the corresponding Input Columns.

Dimension 2: Lee_Car_Details

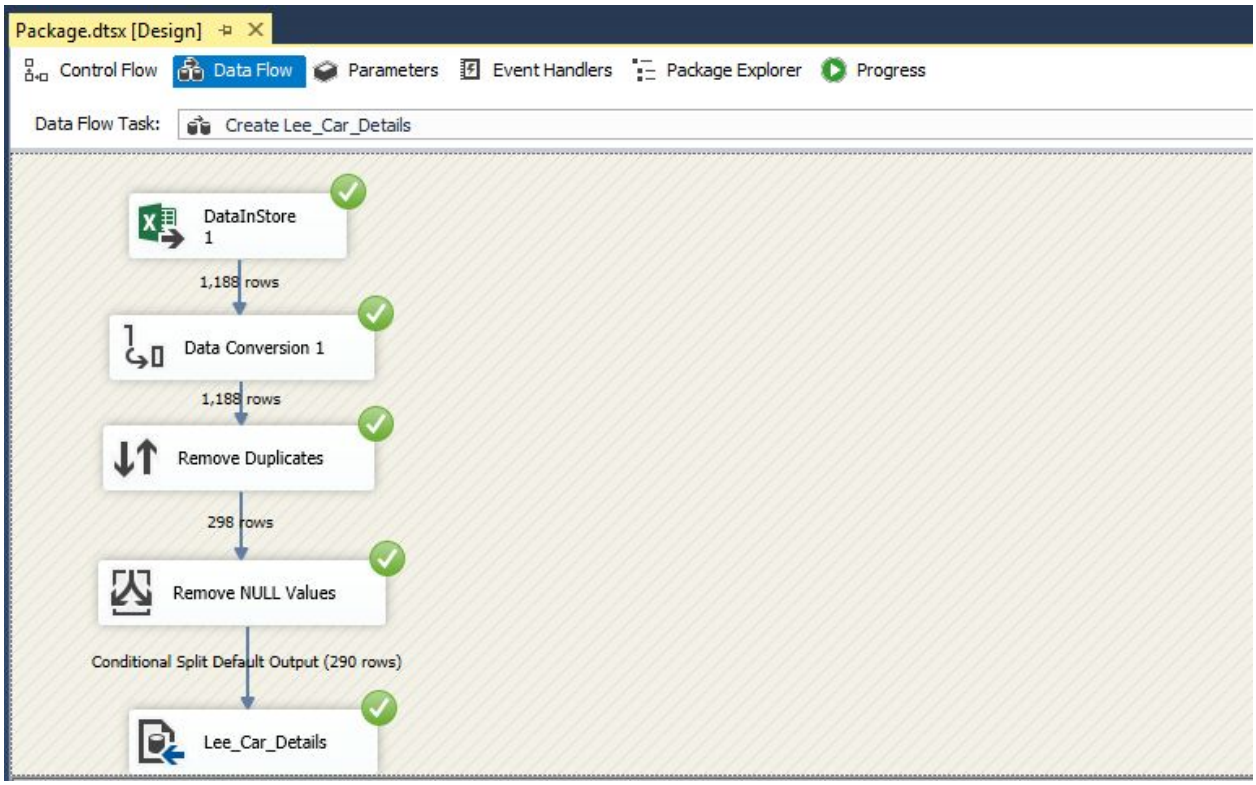


Figure 11: This screen capture shows the completed tasks of our *Create Lee_Car_Details* data flow task.

For this dimension it was necessary to use a *Conditional Split* 

. It was used to split the *NULL* values from our data set before loading into SQL. This node was used just prior to the SQL transfer to ensure that any *NULL* values during the data flow did not pass through, although in this project we would not expect any *NULL* values to be created. From examining the data profile in section 1 we found that the *NULL* values occur for a number of columns on each row, indicating that no car information was recorded for a particular order. The following screen capture shows how we are able to exclude values by using specific conditions on columns. With the *Conditional Split* node, any tuples that meet the condition are direction to a separate output. This could be sent to log or another table depending on the database manager's objectives.

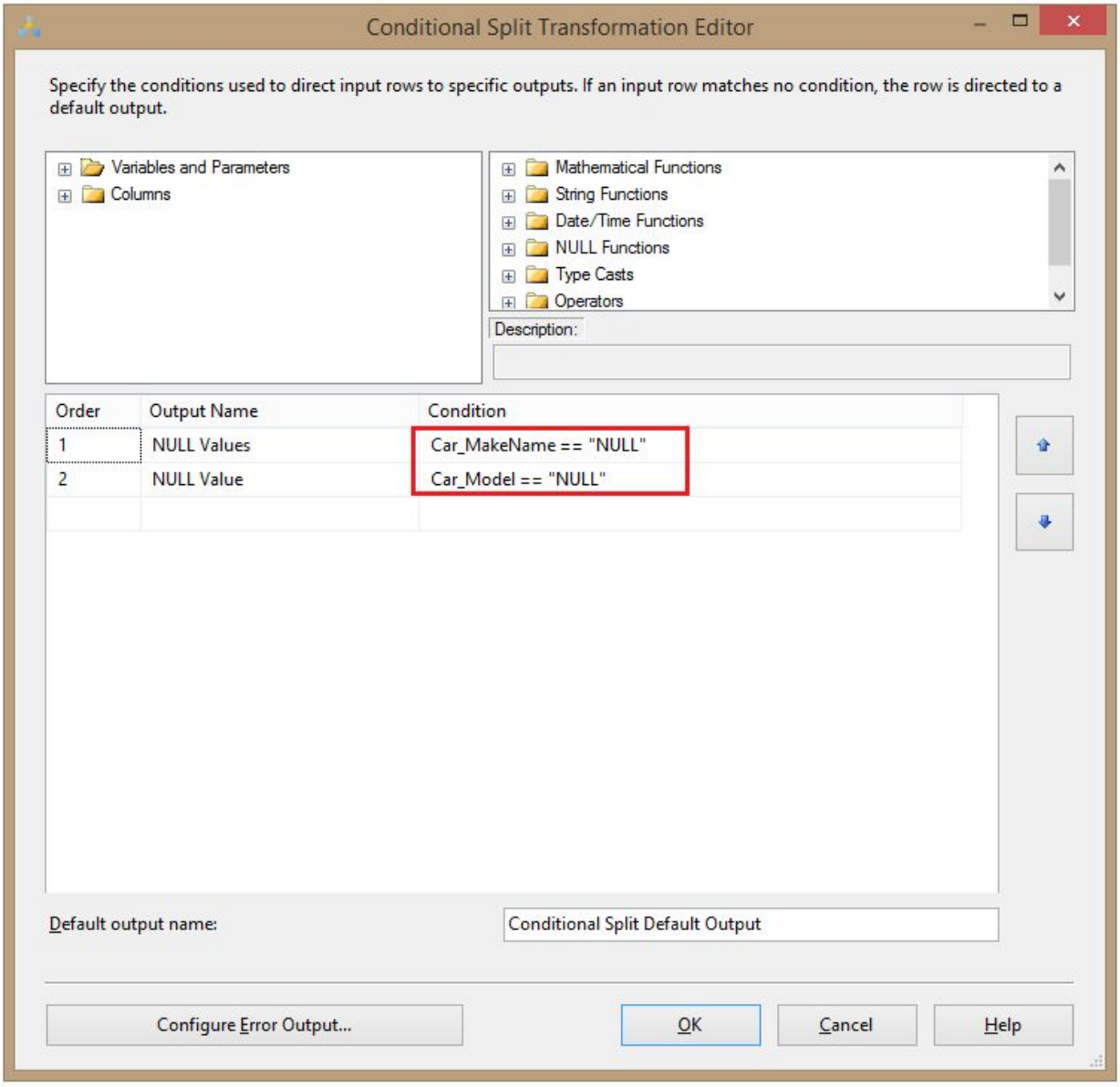


Figure 12: This is the **Edit** screen of the *Conditional Split* node. Highlighted are the conditions used to exclude *NULL* values.

3.3 ETC Process for the Fact Table

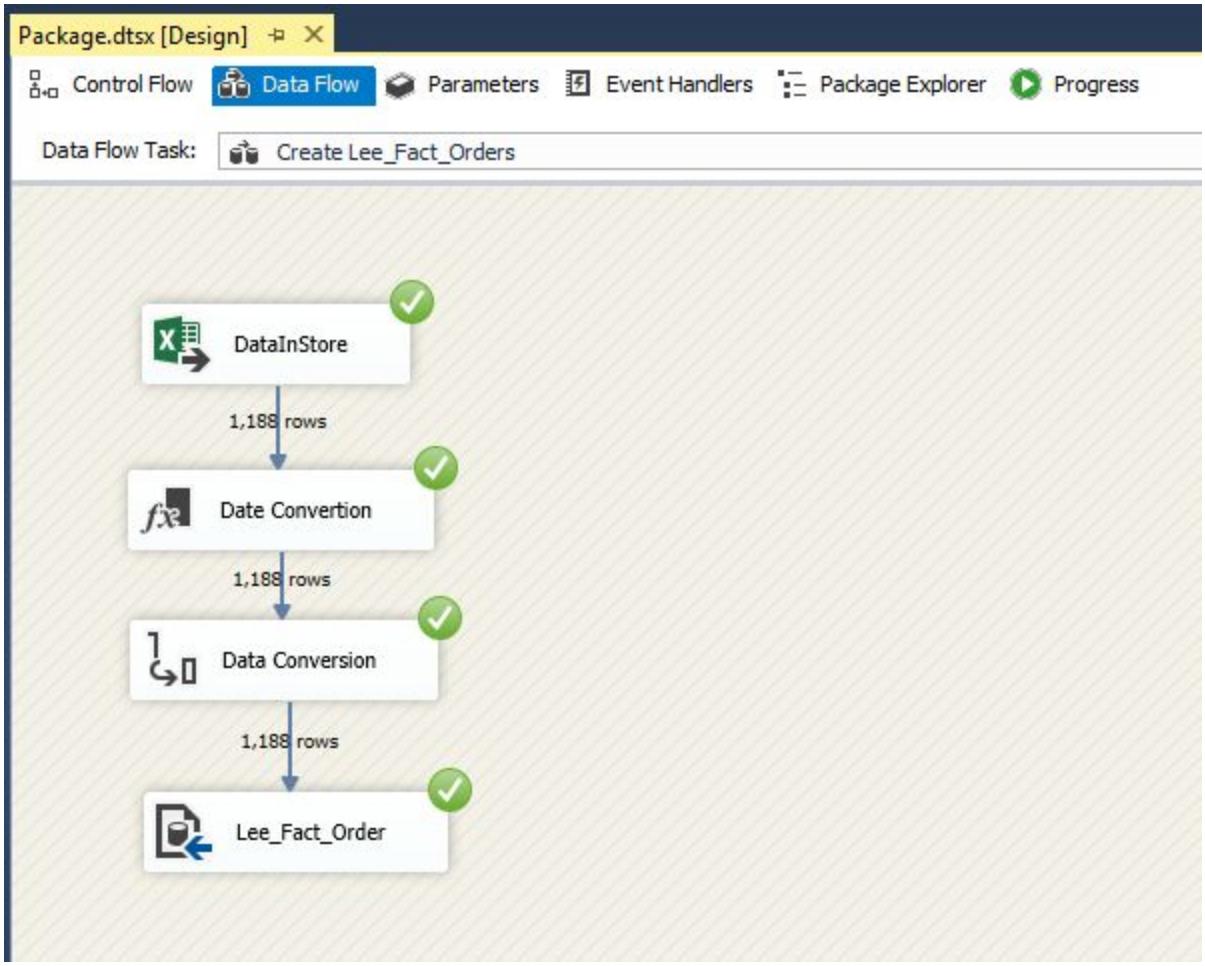


Figure 13: Completed tasks for our Fact Table data flow task.

Transformation Used

As part of our fact table processing, we used a *Data Conversion* node. This node is used to convert the data type in an input column and creates a new output column with the new data type. We needed to use this transformation as the input data was of the type *NVARCHAR* which accepts unicode values. For our data warehouse, we are using the *VARCHAR* data type, therefore, we need to convert the data to non-unicode before loading it into SQL.

How Does the Data Conversion Work?

We have implemented the *Data Conversion* node before our SQL import to ensure values are the correct type before going into our database.

To successfully convert the data we simply changed the Data Type value in the *Data Conversion* node. The following data type changes had to be made to match our database:

Unicode strings have been converted to: **string[DT_STR]**
Eight-byte unsigned float have been converted to: **four-byte signed integer (INT)**
string[DT_STR] that represent a date value have been converted to: **database timestamp[DT_DBTIMESTAMP]**

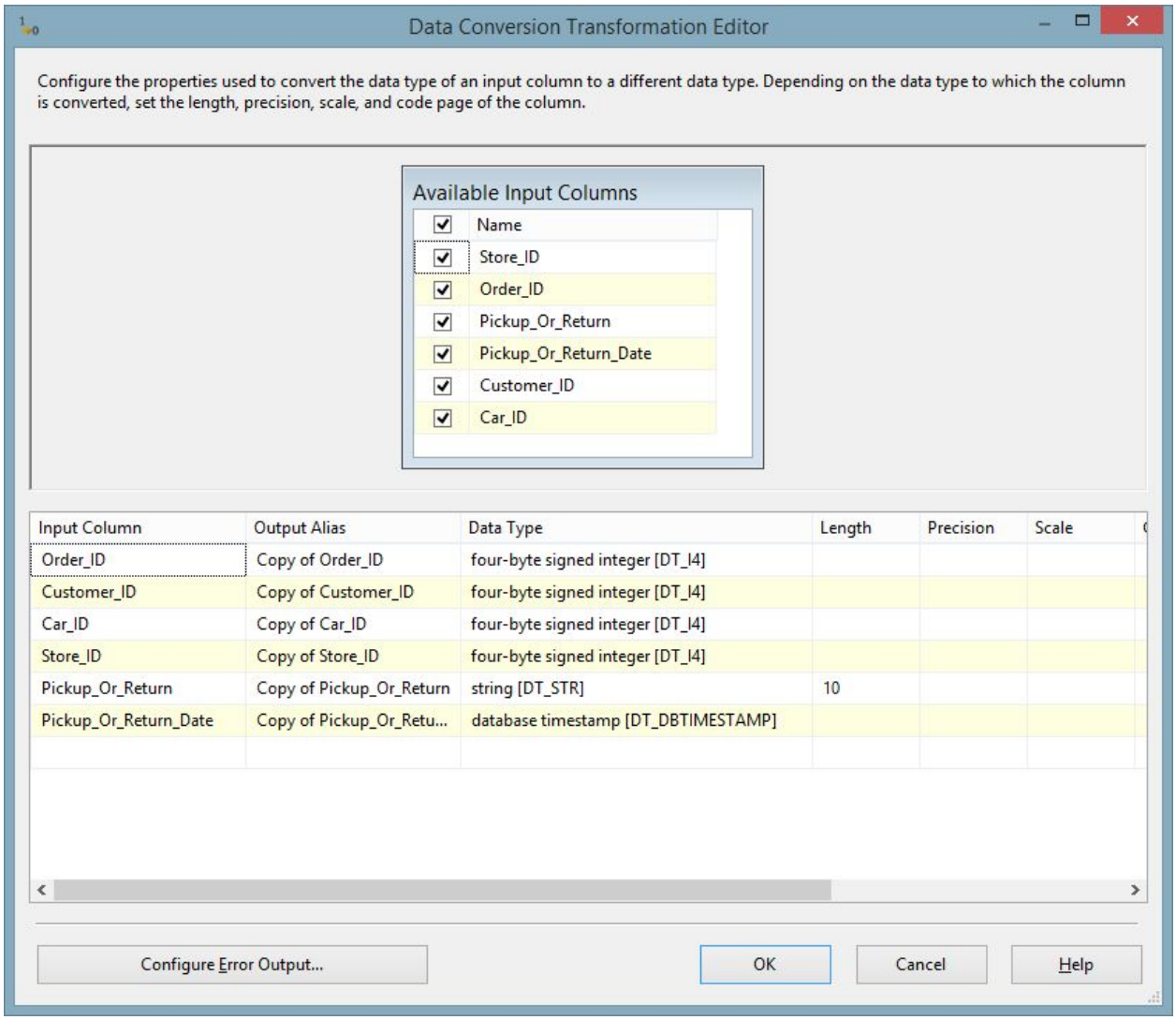


Figure 14: Final settings of the *Data Conversion* transformation node used in our Fact Table data flow task.

4. Data Warehouse Analysis

Due to an SQL server error we were unable to *Process* our data cube. Look to *Figure 21* for more information.

4.1 Car Rental Quantities

To create monthly reports showing the number of cars picked up or returned to stores in particular locations, we needed to build a cube using our fact table and the Store_Details dimension. This would allow us to query a date range, filter by Store_City and count the number of orders.

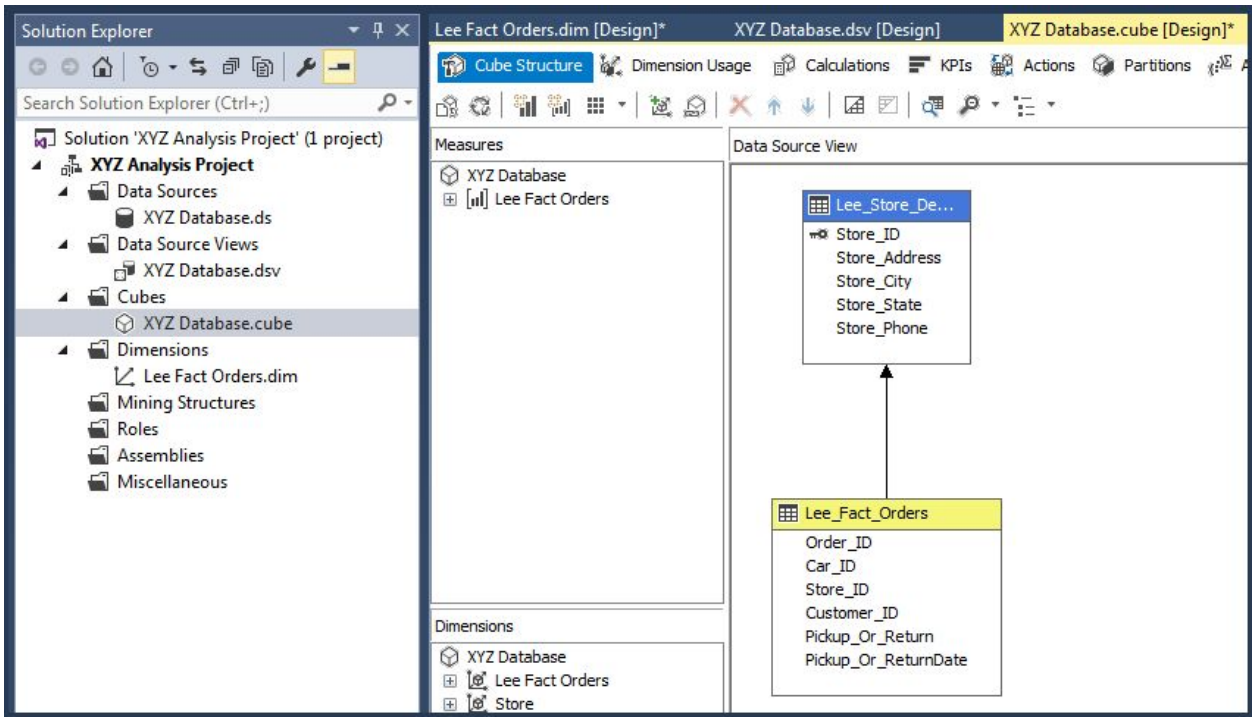


Figure 15: Car Rental Quantities Data Cube Design.

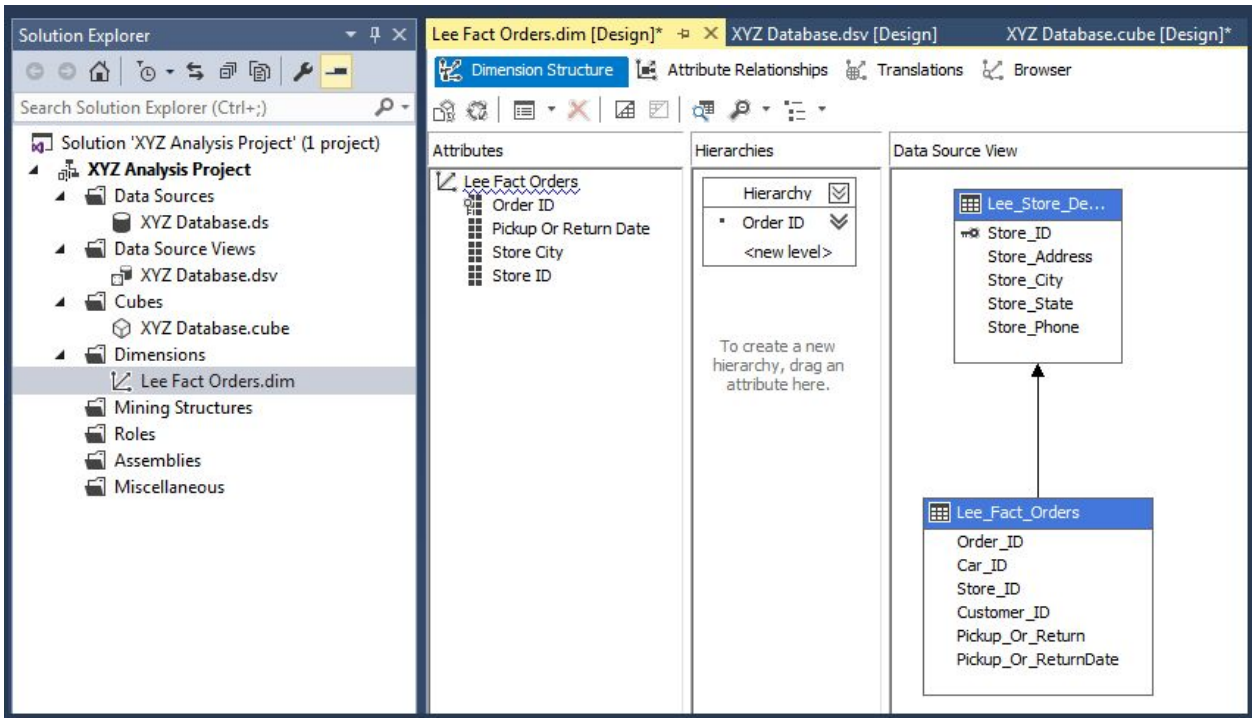


Figure 16: Car Rental Quantities Data Cube Dimensions.

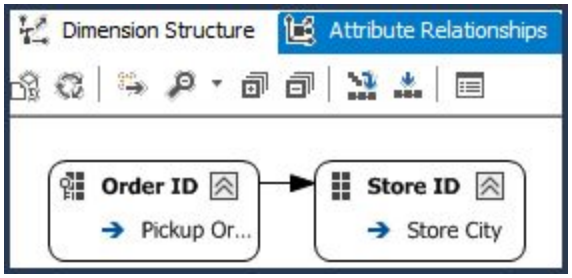


Figure 17: Car Rental Quantities Data Cube Dimension Attributes Relationships.

4.2 Car Popularity

For us to be able to provide information on popular cars to customers in a particular area, we needed to build a cube using our fact table, Store_Details, Customer_Details and Car_Details dimensions. This would allow us to find the customer's location, the time (Month), car model information, and order numbers at each store.

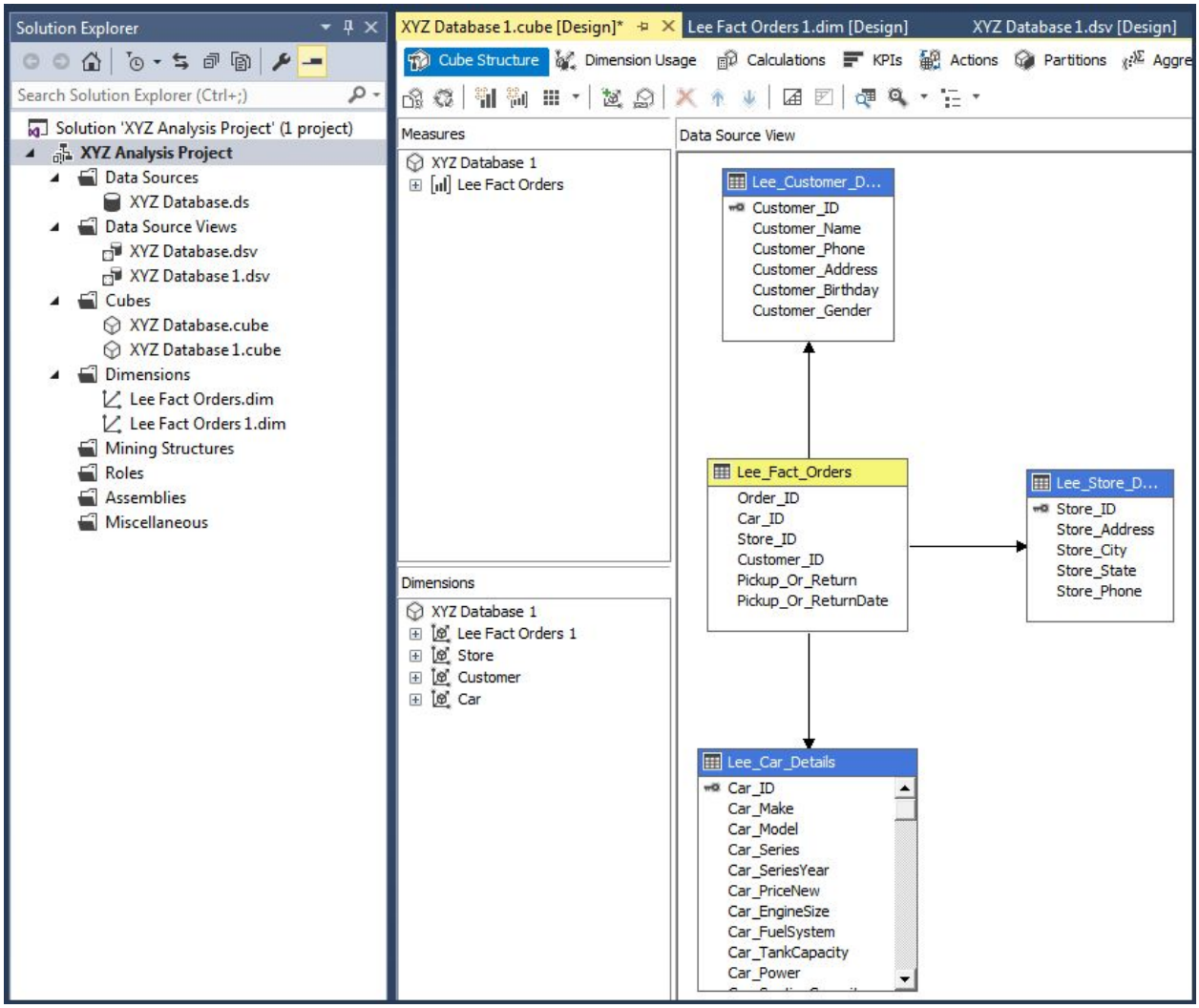


Figure 18: Car Popularity Data Cube Design.

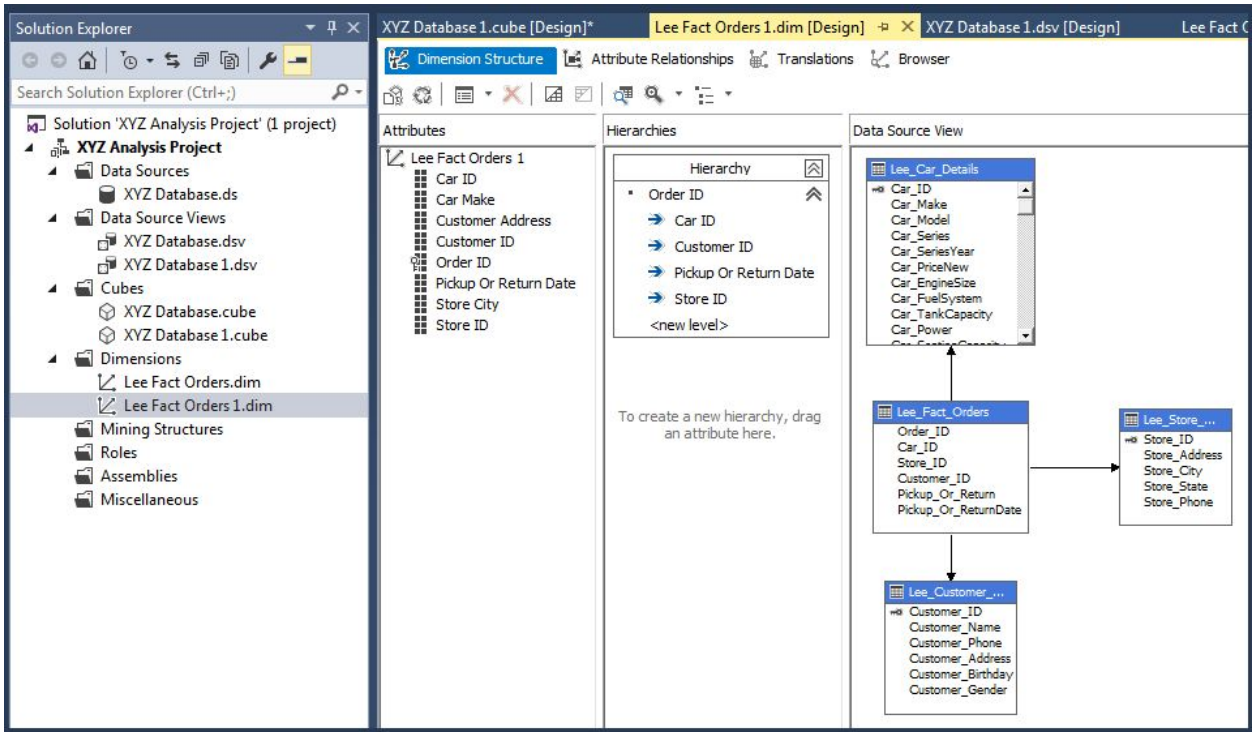


Figure 19: Car Popularity Data Cube Dimensions.

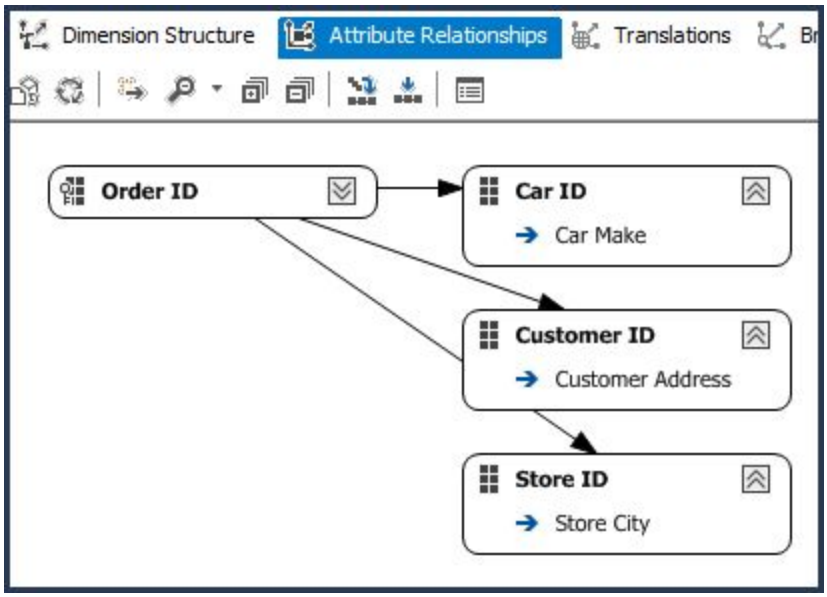


Figure 20: Car Popularity Data Cube Dimension Attribute Relationships.

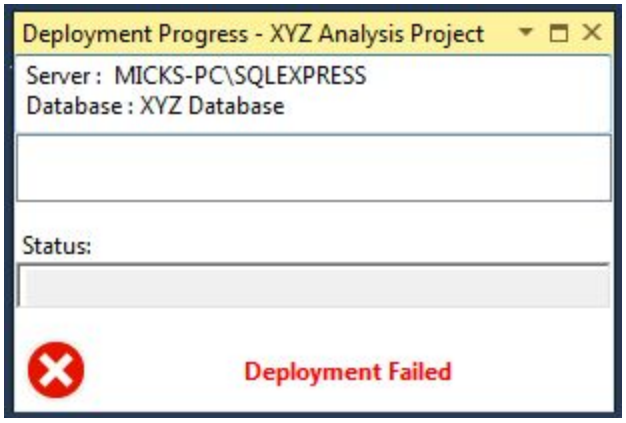


Figure 21: Deployment Error. The following error was displayed along with this screen.

Severity Code Description Project File Line
Error The project could not be deployed to the 'MICKS-PC\SQLEXPRESS' server because of the following connectivity problems :
A connection cannot be made to redirector. Ensure that 'SQL Browser' service is running.
To verify or update the name of the target server, right-click on the project in Solution Explorer, select Project Properties, click on the Deployment tab, and then enter the name of the server.

Due to this error we were unable to deploy our data cubes for question 4 into our databases. After trying everything to fix the connection issue we ran out of time.