|  |
| --- |
| Queensland University of Technology |
| CAB430 Assignment 1 |
| Data and information integration |

|  |
| --- |
| Patric Marchant  4-15-2020 |

Contents

[Question 1 2](#_Toc39321831)

[Question 2 5](#_Toc39321832)

[Question 3 7](#_Toc39321833)

[Question 4 10](#_Toc39321834)

[Statement of completeness 12](#_Toc39321835)

# Question 1

1. To complete a data profile on the full provided dataset, Excel Power Query was used to get an overview of each column. To represent the findings, a table was created listing important information about each column in the dataset, however, many of these columns contain either “NULL” values or invalid input for the datatype, below the table is a summary of the problems in each column.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Element Name | Datatype | Format/possible vals | Minimum | Maximum | Distinct/unique vals | Average | Data issues? |  |  |  |  |  |  |
| Store\_ID | Numerical | 43 | 1 | 50 | 41/0 | 20.41 | No |  |  |  |  |  |  |
| Store\_Name | Text | Brisbane\_store | N/A  (String) | N/A  (String) | 41/0 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Store\_Address | Text | 654 street name | N/A  (String) | N/A  (String) | 41/0 | N/A  (String) | Yes (Null Values) |  |  |  |  |  |  |
| Store\_Phone | Text | # (##) ### ###-#### | 1 (11) 500 555-0110 | NULL | 33/0 | N/A | Yes (Null Values, different data formats) |  |  |  |  |  |  |
| Store\_City | Text | Brisbane | N/A  (String) | N/A  (String) | 41/0 | N/A  (String) | Yes (Null Values) |  |  |  |  |  |  |
| Store\_State\_Name | Categorical | Queensland, Tasmania, New South Wales, Victoria, South Australia | N/A  (String) | N/A  (String) | 6/0 | N/A  (String) | Yes (Null Values) |  |  |  |  |  |  |
| Order\_ID | Numerical | 123 | 3 | 519 | 594/1 | 255.23 | No |  |  |  |  |  |  |
| Order\_Or\_Pickup | Categorical | Order, Pickup | N/A  (String) | N/A  (String) | 2/0 | N/A  (String) | No |  |  |  |  |  |  |
| Order\_Or\_Pickup\_Date | Date | yyyymmdd | 20050704 | 20091018 | 548/210 | 20060962.66 | No |  |  |  |  |  |  |
| Customer\_ID | Numerical | 12345 | 11010 | 11558 | 107/0 | 11238.15 | No |  |  |  |  |  |  |
| Customer\_Name | Text | John H | N/A  (String) | N/A  (String) | 104/0 | N/A  (String) | No |  |  |  |  |  |  |
| Customer\_Phone | Text | # (##) ### ###-### \* | 1 (11) 500 555-011\* | 974-555-017\* | 60/0 | N/A | Yes (mixed formats) |  |  |  |  |  |  |
| Customer\_Addresss | Text | 654 street name | N/A  (String) | N/A  (String) | 106/0 | N/A  (String) | Yes (broken/misentered data) |  |  |  |  |  |  |
| Customer\_Brithday | Date | dd/mm/yyyy | 7/06/1936 | 20/08/1984 | 106/0 | N/A | No |  |  |  |  |  |  |
| Customer\_Occupation | Categorical | Labour, Manager, Nurse, Retiree, Researcher | N/A  (String) | N/A  (String) | 5/0 | N/A  (String) | No |  |  |  |  |  |  |
| Customer\_Gender | Categorical | M, F | N/A  (String) | N/A  (String) | 2/0 | N/A  (String) | No |  |  |  |  |  |  |
| Car\_ID | Numerical | 12345 | 14806 | 15655 | 590/1 | 15272.8 | No |  |  |  |  |  |  |
| Car\_MakeName | Text | Mazda | N/A  (String) | N/A  (String) | 19/0 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_Model | Text | S40 | N/A  (String) | N/A  (String) |  | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_Series | Text | T4 | N/A  (String) | N/A  (String) | 209/1 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_SeriesYear | Date | 2004 | 1964 | 2008 | 22/34 | 1999 | Yes (null values) |  |  |  |  |  |  |
| Car\_PriceNew | Money | 33750 | 2530 | 225600 |  | 49518.2 | Yes (NULL values) |  |  |  |  |  |  |
| Car\_EngineSize | Text | 2.5L | N/A  (String) | N/A  (String) | 27/0 | N/A | Yes (NULL values) |  |  |  |  |  |  |
| Car\_FuelSystem | Text | TURBO MPFI | N/A  (String) | N/A  (String) | 16/0 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_TankCapacity | Text | 60L | N/A  (String) | N/A  (String) | 41/0 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_Power | Text | 110kW | N/A  (String) | N/A  (String) | 81/0 | N/A | Yes (NULL values) |  |  |  |  |  |  |
| Car\_SeatingCapacity | Numerical | 0, 2, 4, 5, 7, 8, 12 | 0 | 12 |  | 4.8 | Yes (NULL values) |  |  |  |  |  |  |
| Car\_StandartTransmission | Text | 4A | N/A  (String) | N/A  (String) | 30/0 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_BodyType | Text | 4D Sedan | N/A  (String) | N/A  (String) | 18/0 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_Drive | Categorical | FWD | N/A  (String) | N/A  (String) | 5/0 | N/A  (String) | Yes (NULL values) |  |  |  |  |  |  |
| Car\_Wheelbase | Numerical | 2700ml | 2230mm | NULL | 91/1 | N/A | Yes (NULL values) |  |  |  |  |  |  |

As shown in the data profile table, most of the columns in the source data file are valid, but just contain some NULL values. The first problem found was the Customer\_address field which has a misspelt column name (Customer\_Addresss), and contains some broken/misentered data, “customer\_brithday” is also another misspelt column heading.

While there are also technically no “data issues” in the order\_ID field, there is infact an order\_id which has an order, but no corresponding pickup, since all of the other order\_ID’s have a order and corresponding pickup, we will need to delete the distinct order\_ID, as our orders will need both an order and pickup date of which neither can be null.

We can also see that there are fields such as “customer\_brithday” and “order\_or\_pickup\_date” which are both keeping track of essentially the same format of information, because “order\_or\_pickup\_date” also includes time, we may strip this from the field, as the client never asks for any analytics to do with time, only date.

Finally, formatting of most of these fields will need to be set as to make sure only the correct data is logged from this point forward.

1. A)

|  |
| --- |
| **Fact\_Orders** |
| Order\_ID |
| Car\_ID |
| Customer\_ID |
| Store\_ID |
| Time\_ID |

|  |
| --- |
| **Vehicles** |
| Car\_ID |
| Car\_MakeName |
| Car\_Model |
| Car\_PriceNew |

|  |
| --- |
| **Customers** |
| Customer\_ID |
| Customer\_Occupation |
| Customer\_Gender |

|  |
| --- |
| **Stores** |
| Store\_ID |
| Store\_City |
| Store\_State\_Name |

Stores: Store\_City - > Store\_State\_Name

|  |
| --- |
| **Times** |
| Time\_ID |
| Time\_Day |
| Time\_Month |
| Time\_Year |

Times: Time\_Day -> Time\_Month -> Time\_Year

B)

The “Star” model was chosen for this data, with a diagram detailed below

A screenshot of a cell phone

Description automatically generated

# Question 2

CREATE TABLE Vehicles (

Car\_ID INT NOT NULL PRIMARY KEY,

Car\_MakeName VarChar(25) NOT NULL,

Car\_Model VarChar(20) NOT NULL,

Car\_PriceNew money NOT NULL,

);

CREATE TABLE Customers (

Customer\_ID INT NOT NULL PRIMARY KEY,

Customer\_Occupation VarChar(25) NOT NULL,

Customer\_Gender Bit NOT NULL,

);

CREATE TABLE Store (

Store\_ID Smallint NOT NULL PRIMARY KEY,

Store\_City VarChar(30) NOT NULL,

Store\_State\_Name VarChar(20) NOT NULL,

);

CREATE TABLE Times (

Time\_ID INT NOT NULL PRIMARY KEY,

Time\_Day smallint NOT NULL,

Time\_Month smallint NOT NULL,

Time\_Year smallint NOT NULL,

);

CREATE TABLE Fact\_Orders (

Order\_ID INT NOT NULL PRIMARY KEY,

Car\_ID INT NOT NULL FOREIGN KEY REFERENCES Vehicles(Car\_ID),

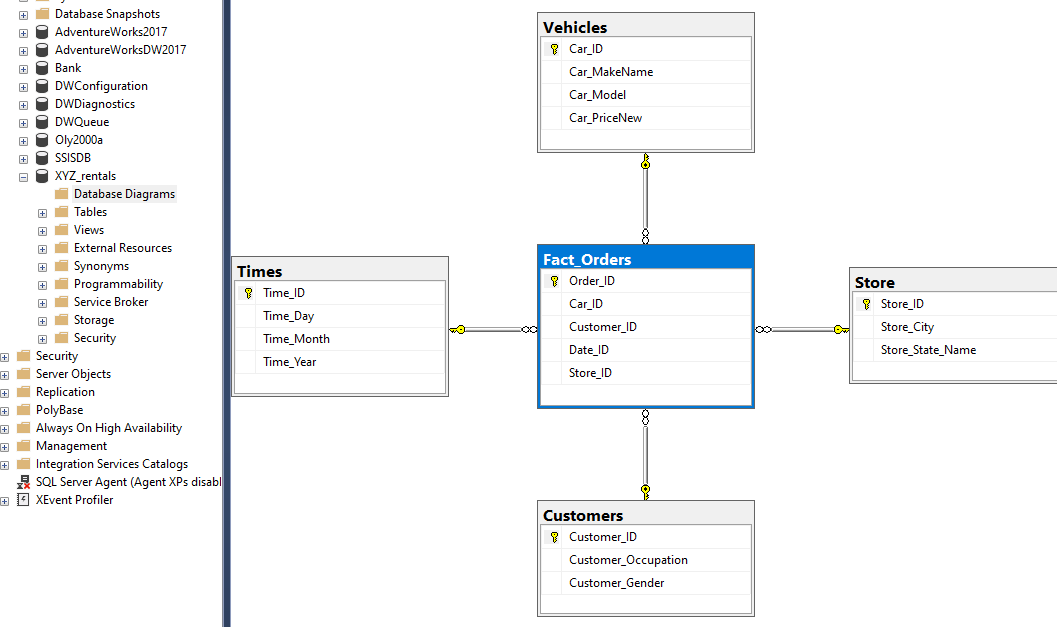
Customer\_ID INT NOT NULL FOREIGN KEY REFERENCES Customers(Customer\_ID),

Time\_ID INT NOT NULL FOREIGN KEY REFERENCES Times(Time\_ID),

Store\_ID Smallint NOT NULL FOREIGN KEY REFERENCES Store(Store\_ID),

);

2)



# Question 3

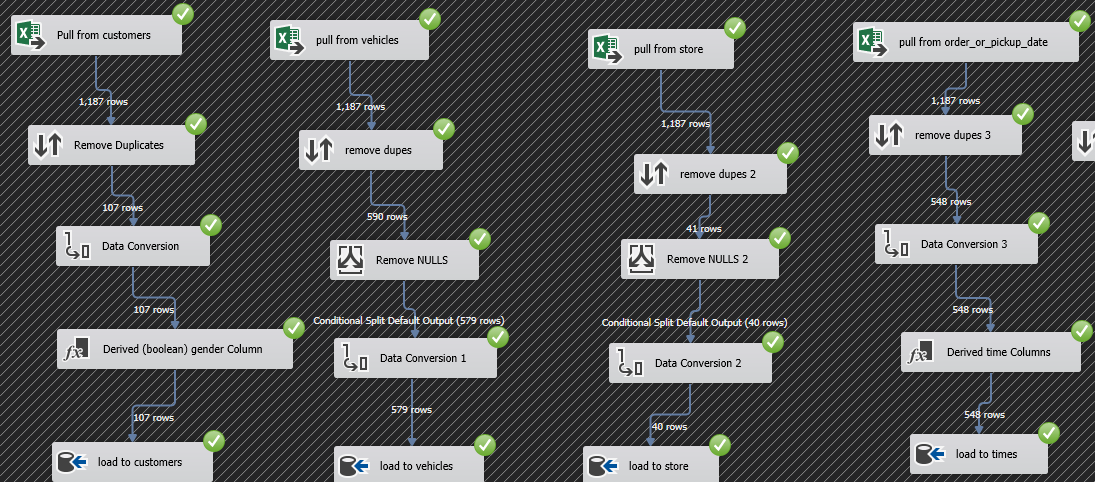
1)

For the ETL part of the task, I decided to try and put all of the data flow logic inside one data flow task, which worked in the end, however, if I had the chance to do it again, I would separate each dimension and fact tables ETL process as they often got in the way of each other running successfully.

A screenshot of a cell phone

Description automatically generated

The data flow for this task is quite straightforward, with a task being created for each dimension (and the facts table, which is discussed later), before jumping in, it is important to note each task will not be covered extensively, as each dimension undergoes essentially the same process, albeit using different transformations at different times. First, Each dimension pulls its relevant data from the source Excel file, removes duplicate primary key values and null values (in parts where null values exist). From here, the “customer” and “time” dimensions use derived columns to create new values to fill database (customer creating bool gender, time creating time\_day, time\_month, time\_year), the data then undergoes a type conversion to convert each columns values to the SQL DB’s format, and are placed in the DB.



Filling the Fact table however, is slightly different, as we first pull each order\_ID (used as primary key), remove duplicates and convert the order\_ID to the relevant format. From here, we lookup each foreign key’s for each dimension, which are then loaded to the DB and collected under the relevant order\_ID.

A screenshot of a cell phone

Description automatically generated

2)

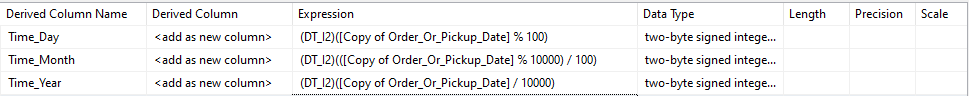
**Dimension tables**

The first transformation we use when executing each dimensions ETL task is the “Sort” transformation. While this transformation does indeed sort the values, we use it to remove any duplicate “ID” values, which can be fatal when filling the tables due to each “ID” column having a primary key constraint.

Another important transformation which is used in the ETL process is the “Derived Column” transformation. This does what it sounds like, and allows us to derive new column(s) from existing columns values. This is used in the customers ETL flow to properly cast the “customer\_gender” string to a Boolean using code (did not work, code is shown below), and the time ETL flow, where it creates new values based on order\_or\_pickup\_date (time\_day, time\_month, time\_year).

**(DT\_BOOL)([Copy of Customer\_Gender] == "M" ? 1 : 0)**

^ check if Copy\_of\_customer\_gender is equal to “M” (male), if so, set to 1 (true), else set to 0 (false)



**Fact Table**

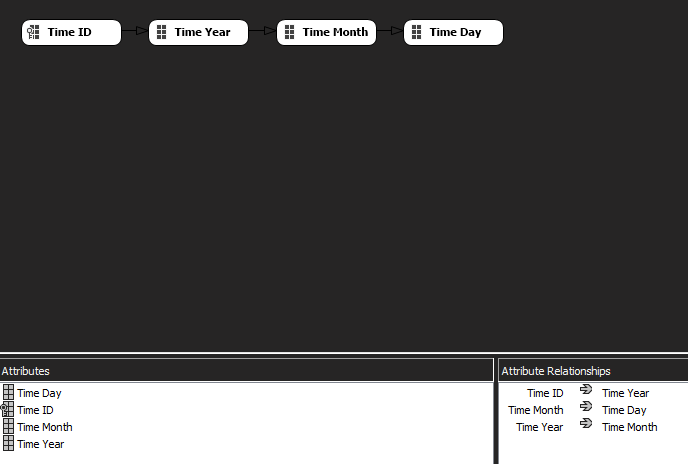
The most important transformation for building the fact table is without a doubt the “lookup” transformation, which allows us to join data with common columns in a reference dataset. This is important as the fact table relies on Foreign keys to link the data to a common “Order\_ID”, otherwise we would have four dimension tables without any relation. Using this transformation proved to be very helpful, as simply converting each “ID” column and removing duplicates was not being accepted by the DB.

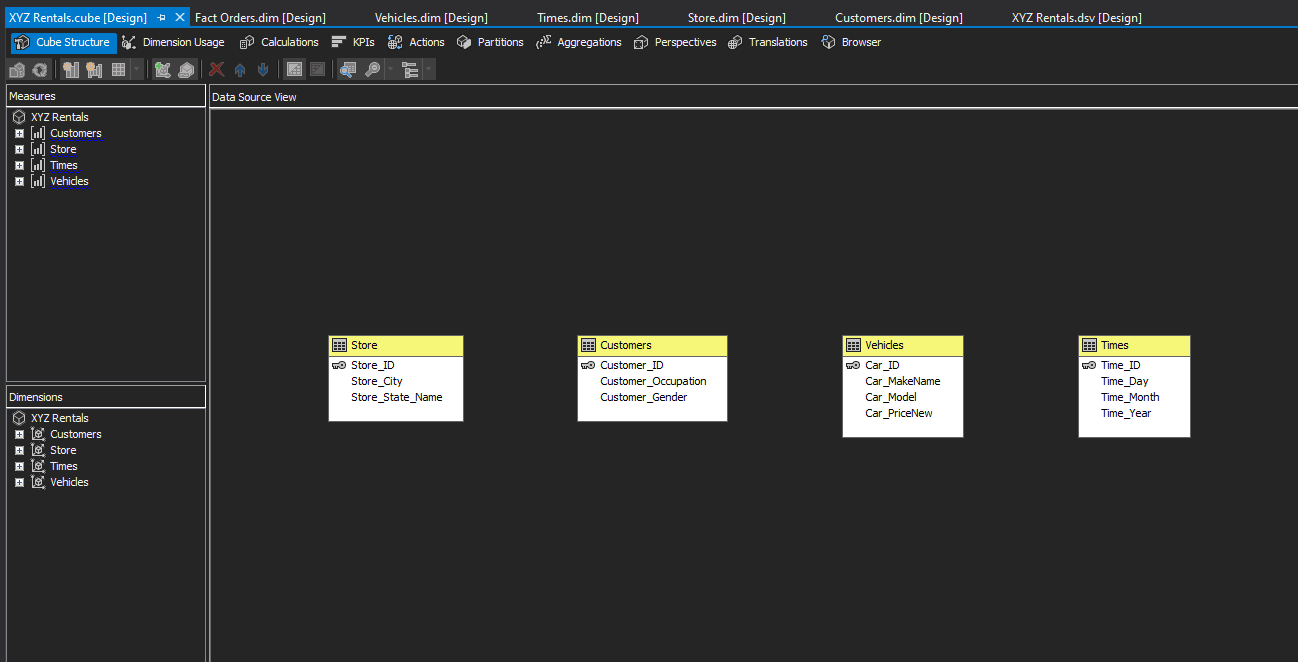
# Question 4

While the data cube was created successfully, and all dimensions and hierarchies were added, due to many deployment issues, queries were unable to run, and I was out of time to work on the assignment further.

A screenshot of a cell phone

Description automatically generated





# Statement of completeness

|  |  |
| --- | --- |
| Q1 | **YES** |
| 1. Clearly describe all information in the source data. | YES |
| 1. The detailed design of a fact table and its dimensions that include the data chosen for the proposed data warehouse. | YES |
| 1. Clearly describe the schema that you have chosen for your data warehouse. | YES |
| Q2 | **YES** |
| 1. Correct SQL statements to create a complete database including all the tables (standard data types, including all the chosen data items) | YES |
| 1. Correct ER diagram of your database | YES |
| Q3 | **YES** |
| 1. An overview of your ETL process including all the tasks that you have used in your ETL application. (Verify with your submitted IS project) | YES |
| 1. Explain three transformations used in your ETL application. (Verify with your submitted IS project) | YES |
| Q4 | **PARTIAL** |
| 1. Discussion about the data cube(s) | PARTIAL |
| 1. Result screenshots displaying the required information with a brief explanation that the results satisfy the company’s expectation. | NO |