dat

|  |  |
| --- | --- |
|  | |
| Working With Data–Assessment 2  TU060 : Data Importing | |
|  | |
| Ciaran Finnegan – Part Time – First Year 2021/2022  MSc in Computer Science (Data Science)  Student No : D21124026  4/1/2022 |  |
|  |  |

Table of Contents

1 Description of Data Elements 3

1.1 Data Files 3

2 Data Import Process 4

2.1 General Approach 4

2.2 Loading the CSVs 5

3 Next Steps? 19

3.1 The Relational Data Model for The Telco 19

3.2 The Data Warehouse 19

# Description of Data Elements

## Data Files

The following are the list of CSV files that form the basis of the initial database set up in this assignment.

These data files contain the information to allow customer analysis within an imaginary telecommunications company.

Table

Description automatically generated with medium confidence

# Data Import Process

## General Approach

Using the built in ‘Import Data…’ option within Oracle SQL Developer was the most obvious approach to load the CSV files.

The first line of each CSV file contains a row with the text descriptions of the column names.

The following rows contain the actual customer/call/company information data.

Each file was loaded separately, and SQL Developer auto generated the SQL Scripts that would execute the TABLE CREATE Statements and then INSERT the values stored in the CSV files.

This document will provide a detailed analysis, in Section 2.2.1, of the steps taken to load ***customers.csv*** into the assignment database.

**Each CSV file will go through a similar process but the subsequent sub sections in Section 2 will only provide a general overview and call out the most significant data import issues.**

## Loading the CSVs

### Customer Data

Sample view of ***customers.csv*** file (in Notepad++).

A screenshot of a computer

Description automatically generated

Open the ‘Import Data…’ option within Oracle SQL Developer.

Graphical user interface, text, application, email

Description automatically generated

This launches the **Data Import Wizard**. Navigating to the local folder storing the data calls up the ***customers.csv*** file, and File Contents preview window is available.

Graphical user interface, table

Description automatically generated

The first row of the CSV file is correctly interpreted as the column names, which is the default setting in the ***File Format*** section.

The Import Method is set on the next Wizard screen.

Graphical user interface, table

Description automatically generated

The name of the table to be created in out assignment database is set to ***tblCustomers***.

The ‘*tbl*’ prefix is a convention adopted during my time as a SQL developer with a former employer. The purpose was to label each database object according to type/purpose. Hence ‘tbl’ refers to a database table object.

The Import Method is set to ‘***Insert Script’*** because this output provides a SQL Script that contains both the CREATE TABLE SQL code, and the individual INSERT commands for each row of data. This is an option step but one which I chose because I felt it would provide more flexibility if I needed to re-create the assignment database

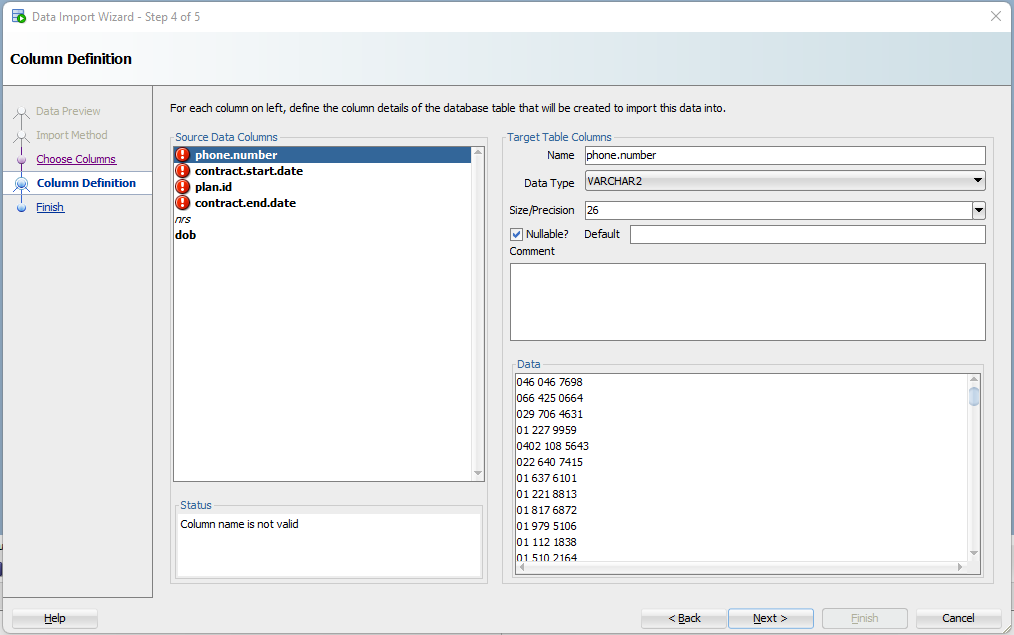
Column selection and ordering options are available on the **Choose Columns** (third) screen of the wizard.

Graphical user interface, application

Description automatically generated

All columns are selected, and the ordering of attributes appears to be acceptable. No changes are required.

The **Column Definition** (fourth) screen of the Wizard objects to the format of the column names and flags four of them as Invalid in the list in source Data Columns.



The ‘.’ character is not valid for column names. (A similar issue can be seen with data file uploads in SAS Studio).

To circumvent this issue the ‘.’ characters will be replaced with underscores (‘\_’). The updates are made in the Target Table Columns section (see below).

***Phone\_number***

Graphical user interface, text, application

Description automatically generated

This removes the error from the ‘phone\_number’ column but there are some additional considerations:

* The value is almost certainly going to be key value on the table and so we should not allow any ‘null’ entries. It will be essential for the integrity of the database that every row in this table has a unique phone number to identify the customer.
* Although this a phone number, the assumption is that we should leave the Data Type’ as VARCHAR2. This attribute is not really an integer, but an identifier in numeric format. For example, at a later stage it could be necessary to carry out some sub-string analysis on number prefixes.
* The default precision size also seems satisfactory, or at least it is not necessary to change the default value.

***Contract\_start\_date***

Graphical user interface, application

Description automatically generated

Underscore characters removed the invalid warning for the contract start date, but the following changes were also required;

* The Data Type of ‘DATE’ appears to be correct, but the ‘Format’ was changed to DD/MM/YYY. This is the format we will replicate across the other DATE columns.
* The value should ideally not be ‘null’, but we will not add a constraint.

***Plan\_id***

Graphical user interface, application

Description automatically generated

Again, underscore characters removed the invalid warning for the contract start date, but the following changes were also required;

* The column should not allow ‘null’ values as each customer must be associated with a valid contract plan.

***Contract\_end\_date***

Graphical user interface, application

Description automatically generated

Underscore characters removed the invalid warning for the contract end date, but the following changes were also required;

* The default Data Type of VARCHAR2 needs to be altered as this is a date attribute. The Data Type was changed to DATE and the ‘Format’ was set to DD/MM/YYY. This is the format we will replicate across the other DATE columns.

***Dob*** (date of birth)

Graphical user interface, application, Word

Description automatically generated

Underscore characters removed the invalid warning for the date of birth (dob) attribute, and the following observations applied;

* The Data Type of ‘DATE’ appears to be correct, and the ‘Format’ was DD/MM/YYY. This is the format we will replicate across the other DATE columns.
* A ‘null’ value is acceptable.

Finish the process.

Graphical user interface, text, application, email

Description automatically generated

The data import and SQL Script generation process is not instantaneous, but when it completes the SQL is visible in SQL Developer.

Graphical user interface, text, application, email

Description automatically generated

Selecting the ‘Run Script (F5)’ option will execute the SQL to CREATE the *tblCustomers* table and INSERT all 5000 data rows.

Graphical user interface, text, application, email

Description automatically generated

### Call Data

Sample view of ***calls.csv*** file (in Notepad++).

A screenshot of a computer

Description automatically generated

Notes:

* Connection\_id to the head of the table – not null.
* Phone\_number – not null.
* The Call\_Time is imported as a text value (VARCHAR2(26). This will require conversion routines during the data model process, but it was not possible to convert through the GUI at this point.

### Contract Plans Data

Sample view of ***contract\_plans.csv*** file (in Notepad++).

**Graphical user interface, text, application

Description automatically generated**

Notes:

* Id – not null.
* Name – not null.

### Call Rates Data

Sample view of ***call\_rates.csv*** file (in Notepad++).

**Table

Description automatically generated**

Notes:

* Call\_type\_Id – not null.
* Plan\_id – not null.

### Customer Service Data

Sample view of ***customer\_service.csv*** file (in Notepad++).

**Text

Description automatically generated**

Notes:

* connection\_id – not null (key) and moved to top.
* call\_time – stays as VARCHAR2 – difficult to convert to timestamp. This will be addressed within the Data Warehouse population process.
* call\_type\_id – move to second place.

### Rate Type Data

Sample view of ***rate\_types.csv*** file (in Notepad++).

**Graphical user interface, text, application

Description automatically generated**

Notes:

* id – not null (key).
* name – not null.

### Social Grade Data

Sample view of ***social\_grade.csv*** file (in Notepad++).

**Text

Description automatically generated**

Notes:

* Grade – not null (key), also need to remove padded blanks.
* Social class – not null.

### Voicemails Data

Sample view of ***voicemails.csv*** file (in Notepad++).

Text

Description automatically generated

Notes:

* Connection\_id – not null (key), moved to top.
* Phone number – not null.
* Call\_time – same issue with converting to timestamp.

# Next Steps?

## The Relational Data Model for The Telco

The ***CSV*** files allow the import of a fully loaded database for our Telecommunications company.

The data import process in ORACLE SQL Developer also allows a certain amount of data manipulation to convert certain attributes into the appropriate data element type.

However, this assignment is **not** about building an operational Telco database to support day-to-day operations.

Hence, although the tables loaded into our schema for the Telco have been set up with Primary Keys, and data types have been converted appropriately (as listed in Section 2), this information is not the basis for a Telco Customer Service GUI.

The information loaded provides a snapshot for a data warehouse, and subsequent data analysis for key stakeholders.

Thus, no additional work was conducted after data import to check for Foreign Keys or other data integrity constraints.

## The Data Warehouse

As described in Section 3.1, the data now loaded into our database environment is the start point for the creation of a data warehouse.

In the main report, which accompanies this assignment submission, it is described how call revenue defines the granular level of the Fact table, and how the supporting Dimension tables are created.

The source for all these data warehouse tables will come from the Telco data imported during the process described in this document.

None of the source data imported through the ***CSV*** files will be updated or deleted. All data manipulation will take place within the data warehouse star schema itself.

The supporting SQL Reports and ORACLE Machine Learning processes will also only work directly upon information from the data warehouse schema.