|  |  |
| --- | --- |
|  | |
| Working With Data–Assessment 2  TU060 : Data Warehouse Modelling / Data Analysis / Machine Learning using SQL | |
|  | |
| Ciaran Finnegan – Part Time – First Year  Student No : D21124026  4/1/2022 |  |
|  |  |

Table of Contents

1 Project Overview 3

1.1 High Level Description 3

1.2 Environment Assumptions 3

1.3 Project Execution Instructions 3

2 Section A: Business Drivers for Assignment 4

2.1 Background and Goals 4

2.2 Subject Area for Analysis 4

2.3 Key Stakeholders 4

2.4 KPIs 5

3 Section A: Data Modelling 6

3.1 Reasons for Design 6

3.2 Data Warehouse Schema 14

4 Section A: Implementing the Data Warehouse 15

4.1 Implementation using SQL Scripts 15

5 Section B: Data Analysis Using SQL 16

5.1 Data Analysis Objectives 16

5.2 Key SQL Reports + KPIs 17

5.3 Data Analysis – SQL Report Outputs from Data Warehouse 17

6 Section C: Machine Learning Using SQL 19

6.1 Overview of ML Process 19

6.2 Creating and Populating Fact Table for ML Modelling 19

6.3 Preparing Training and Test Data Sets 19

6.4 Create VIEW for Predicted Values 19

6.5 Evaluating Models 19

7 Appendices 20

7.1 Appendix 1 – SQL Scripts to build the Data Warehouse 20

7.2 Appendix 2 – SQL Scripts to Populate Data Warehouse Dimensions 20

7.3 Appendix 3 – SQL Scripts To Populate Data Warehouse Fact Table 20

7.4 Appendix 4 – SQL Scripts For All SQL Queries 20

7.5 Appendix 5 – SQL Scripts For ML Process 20

8 References 21

8.1 Data Warehouse Design 21

# Project Overview

## High Level Description

This document covers the design, implementation and observations on all parts of the January 2022 CA(2) for the working With Data module in the TU060 Part Time/first Year MSc in Science (Data Science) course.

## Environment Assumptions

The project was developerd in ORACLE SQL Developer.

All SQL scripts used in the assignment have been embedded in the document but also submitted separately as ***\*.SQL*** files.

These SQL files are grouped in ***\*.ZIP*** files that are numbers to indicate the sequence in which they should be executed in a test ORACLE database.

## Project Execution Instructions

The ***\*.ZIP*** files should be unpacked in the order of the file names.

That is, *<filename\_****1****>.zip* is unpacked first and then excuted to update the target ORACLE database, before any action is taken with *<filename\_2>.zip,* and so on.

The CREATE TABLE SQL Scripts folder contains the files creates the source data table with specific table names that are then referenced by later SQL Scritps. This is the first set of SQL scripts to be executed.

The DATAWAREHOUSE SQL Scripts folder contains the files that create and populdate the data warehouse tables from the telecoms database. This is the second set of SQL scripts to be executed.

The SQL QUERIES SQL Scripts folder contains the files that generate the data analysis output on the data warehouse. The is the third set of SQL scripts to be executed.

# Section A: Business Drivers for Assignment

## Background and Goals

A new data warehouse model is being designed and built to improve customer profile data for a this telecommunications company.

This report will explain to the key stakeholder group;

* The primary objectives driving the structure of the data warehouse.
* The range of reporting data that will be available from the warehouse.
* The predictive analysis that will be available to stakeholders in terms of possible customer churn.

## Subject Area for Analysis

In this assignment I have chosen to focus on an analysis of **Revenue performance.**

Therefore the objective will be to ask which customers generate the most financial value for company based on their activity, profile, and call plans?

## Key Stakeholders

The KPI reports produced in this project are Revenue performance data based on quartetly views of the data over an annual timeframe.

This information is thus not expected to be updated daily and is more strategic in value.

The key stakeholder for the type of reporting produced in this project would therefore be;

* **Senior Management.** Revenue performance data that could alter contract plans or rate types. These are the types of decisions that can only be actioned by those within our telecomunications company who control major resourcing and policy decisions.
* **Customer Services agents.** Those in the salesforce in our telecomunications company. The SQL queries in this assignment can be the basis for dashboards that provide a means for Customer Service to focus their attention on customers generating higher revenues.
* **Company owners/potenail investors.** How much revenue is our telecommunications company generating?

## KPIs

Section 5.2 details the specific report objective for this assignment.

Section 5.3 displays the data analysis results from the reports built in SQL for this assignment.

# Section A: Data Modelling

## Reasons for Design

The Telecomunications Data Warehouse in this project is built following the design principles as described in Kimball’s four step process.

1. Identify the Business Process. Do not re-model the Business Department / Area.
2. Identify the Grain.
3. Choose the Dimensions.
4. Choose the Facts

These steps will be applied to the creation of a new Data Warehouse for this project, but this process could also be applied to the enhancement of an existing Data Warehouse to include a new business process reporting objective.

The objectives in creating the star schema model for the Data Warehouse are;

* Be simple.
* Be easy to use.
* Any process loading into these tables should be as simple as possible.
* Queries should perform well with SQL, or other Business Intelligence tools (which are not part of this assignment task).

### Identify the Business Process

This is the first step in designing the Data Warehouse.

A ‘Business Process’ can be defined as a natural operational activity performed in the organisation, in this case for our telecomunications company, that is supported by some form of data collection.

The following should be considered when identifying the process on which we wish to focus;

1. **Look at the business process not the business department.** This allows for data to be collated and reported on in a more consistent manner across the organisation. It helps in avoiding duplication of data, which might occur if we replicate the structure of business units in the Data Warehouse. In this project we are looking at the Revenue process by measuring Revenue performance but we are not looking to build a Customer Service Department report. This assignment will look at Customer revenue generating activity across the company.
2. **Assess impact and risk in reporting on the chosen business process**. Impact is generating reports that the business actually want on a regular basis. The assignment will focus on identifying those types of customer who generate the most revenue, and being proactive to keep them in the business.
3. **In a real world scenario, the business users would provide guidance on a data warehouse procees.** Business users can also help decipher complex business processes. In this assignment we already have conduced an engagemenet with key stakeholder, which will feed into the structure of tables and the SQL queries/reports that will be executed.

For this assignment, the business process is to capture revenue generating activity by the customers of the company.

Building the data warehouse will allow the company to have a better analytical view of the revenue streams per customer, which will then help information subsequent decision making processes.

### Identify the Grain

This is the second step in designing the Data Warehouse.

This is the most important phase of the design process. Redesigning a Data Warehouse at a later date to increase the level of granularity could be an expensive and time consuming process.

The resultant Fact table will be at the centre of our star schema. This table contains all of the measurable facts about the captured business process. We will use the Fact table to extract information the key Revenue data points for customers in this telecommunications company.

I have followed three particular guidelines in my project to identify the correct level of granularity when considering the design of this Data Warehouse.

1. **What is represented by one ‘Fact’ row?** What level of granularity is captured?
2. **Choose the most atomic level of information.** The data cannot be meaningfully subdivided any further. It also allows for easy and effective aggregations.
3. **Allow scope for future reporting requirements**. It is hard to predict future user requirements so the granularity is important to allow further, possibly ad-hoc, reporting requirements.

In my Fact table the focus is on measuring revenue from customer calls. Thus in my dimensional model one Fact row represents *one call event to/from a specific customer at a specific time.*

This is a lower grain than call events by a customer in a given day, as the customer may make multiple calls or voicemails on one day in different time periods (peak or off-peak).

The term ‘call event’ is significant because the telecommunications company distinguishes, in separate database tables, between;

* A voice call made ***by a*** customer.
* A voicemail left ***by a*** customer.
* A call from a Customer Service agent ***to a*** customer.

We will not subdivide out the call ***to*** and ***from*** a customer. It will be assumed that calls made by a customer are a charge to them and revenue to the company. To maintain this level of granularity, calls from a Customer Services agent are not a charge to the telecommunications company itself and will be represented as a *zero* revenue item. This is consistent with the *call\_rate* value of ‘0’ in the source ***Call\_Rates*** database table.

### Choose the Dimensions

This is the third step in designing the Data Warehouse.

Guidelines for this process can be summarised as follows;

1. Who, what, where, when?
2. Best attributes are descriptive.
3. De-normalizes design focuses on high performance reads.
4. Use smallest data types possible.

To capture the attributes of the Revenue performance process for our telecomunications company, I need to have the information on **who** (Customer) was involved in **what** call-event and **when**.

This question dictated the choice of the dimension tables I selected for my Data Warehouse schema, and the attributes in these tables;

* Time (*dw\_dimtblDateTime*)
  + Calendar Date – text description from the Customer Support, Voicemail, and Calls tables.
  + Call Event Date – ORACLE DATE variable converted from ‘Calendar Date’ text.
  + Cal Timestamp - ORACLE Timestamp variable converted from ‘Calendar Date’ text.
  + Day of Week – Number representing day of week, Monday = 1, and so on.
* Customer (*dw\_dimtblCustomer*)
  + Phone Number – The Fact table stores the Connection ID, therefore this text variable can be stored in the Customer DIM table.

* + Plan Name – text description of Customer Plan.
  + Plan Id – Numeric Identifier for the Customer Plan. Added in the dimension table to aid the update of values for call charge in the FACT table.
* Call Event (*dw\_dimtblCallEvent*)
  + Connection Id – unique identifier for the call event.
  + Call event type – text description of call type – peak, roamining, voicemail, etc.
  + Call Event Type Id - – Numeric Identifier for the Call Type. It requires some data conversion to distinguish between ***Peak*** and ***Off-Peak*** for actual calls. Added in the dimension table to aid the update of values for call charge in the FACT table.

The TimeDate dimension table (*dw\_dimtbltimeDate*) is built to provide additional date granularity and a conversion of the date into an integer format to improve reporting performance.

A new ‘surrogate key’ has been created for each of the dimension tables. It is a simple numeric value that I have set to increment in the SQL scripts used in the CREATE TABLE routines.

The surrogate key is necessary to uniquely identify each row in the dimension table and to avoid any confusion with the source Primary Keys from the ‘operational’ database tables of the telecommunications company. This is particularly useful if the key structure in the telecomunications company operational database changed in the future. Such changes will not then have a knock on impact on the Data Warehouse and reporting applications should still be valid.

The surrogate keys of each dimension table are usually simple integer values and are also added to the Fact table. This is done to minimise the number of joins needed to fetch data, which improves the response time of queries (as does the use of simple integer key values).

### Choose the Facts

This is the fourth step in designing the Data Warehouse.

The Fact table exists at the centre of the star schema, as can be seen in Section 3.2.

Defining the measures for the Fact table should follow guidelines such as these;

1. **How does the business measure success?** For Revenue performance we are looking at the charge totals for call events, and which customers are generating the most revenue?
2. **The best measures are fully additive**. It should be possible to roll up the measures and easily perform aggregations. In the SQL Scripts in Section 5 I <***give example***> (as an example).
3. **Data access tools, such as Tableau, PowerBI, (or even SQL Scipts) are suitable for non-additive measures.** ***Year To Date averages***? are calculated in the output of one of my SQL Scripts, but would not be a meaningful unit of data in the Fact table.

The facts are numeric values that correspond to the grain of the table, as defined in Section 3.1.2.

The Fact table ***dw\_facttblCallRevenue*** will be created in out Telecommunications company data warehouse.

The colums for this Fact table can be identified as follows;

1. Date Time Foreign Key – link to DateTime Dimension table.
2. Customer Foreign Key - link to Customer Dimension table.
3. Call Event Foreign Key - link to Call Event Dimension table.
4. Cost Per Minute for Call Event – rate for this particular Call Event. A zero value indicates a call to the customer from Customer Service.
5. Duration of Call Event – recorded in seconds.
6. Charge Generated for Call Event

The Charge Generated (Revenue) amount per call event is a relatively simple metric to report on as it is stored in the Fact table, and allows for more straightforward, and performant, SQL queries.

## Data Warehouse Schema

The Data Warehouse for this project will be implemented with a Star Schema design.

This involves one central Fact table surrounded by a number of Dimension tables.

<Star Schema Diagram>

The operation database from which this Data Warehouse is build has a normalised relational structure.

To optimise queries on the Data Warehouse the tables are effectively ‘de-normalised’.

The Dimension tables contain descriptive information. The Fact table contains keys to all the dimension table Primary keys, and all the measurable attributes required to meet the reporting purpose of this Data Warehouse.

# Section A: Implementing the Data Warehouse

## Implementation using SQL Scripts

Section 1.3 of this document explains the sequence in which the assignment SQL scripts must be unpacked and executed.

ZIP files containing the SQL script folders accompany this report file. These files are also embedded in Section 7 of this report.

The SQL Developer GUI was used to lead the source assignment csv files into an ORACLE database. Those CREATE TABLE scripts were autogenerated by SQL Developer. This process is described in more detail in the supplementary report document, ***Working With Data - CA2 - Data Imports - Student Ciaran Finnegan d21124026 v1-2 281221.docx***, which accompanies this main report.

The work to implement the data warehouse, developer and excute SQL query analysis, and perform Machine Learning predictions, was all developed directly in SQL through SQL developer.

# Section B: Data Analysis Using SQL

## Data Analysis Objectives

The assignment lists the following features for this data warehouse application;

* Identify how valuable a customer is to the company relative to other customers
* Build up a picture of their customers’ profiles
* Determine whether a customer’s behaviour patterns have changed recently
* Identify the call plans which bring in the most revenue

In this section of the report we look at the analysis from the data warehouse, and the underlying SQL queries that extracted the data.

The focus was on the following questions;

* **Who** are the customers generating most revenue?
* **What** types of customer activity is generating the most revenue?
* **How** is revenue performance changing over time?

The information in the output from theses SQL queries is very high level but it intended as an ‘at a glance’ overview for Senior Management.

The message is clear and focuses on the value of customers generated higher amount so revenue.

## Key SQL Reports + KPIs

**SQL Report 1: Top 100 Customers – in last 30 days (by Revenue)**

* A snapshot view of the customer accounts,identified by phone number, which have generated the most revenue in the last quarter. The timeframe is based on the last three months tracked in the data warehouse.

**SQL Report: Revenue Per Plan Per Month**

* A snapshot view of revenue trends broken down by customer call plans. Data is tracked over a single quarter on a monthly basis (this is the range of the data in the data warehouse).

**SQL Report 3: Top 100 Customers – in last 30 days (by Activity)**

* A snapshot view of the customer accounts,identified by phone number, which have been most active, as measured by call event duration.

**SQL Report 4: Top 20 Customers – Revenue Patterns (by Month)**

* Taking the Top 20 customers from the last month, display the revenue trends for the preceeding three months.

## Data Analysis – SQL Report Outputs from Data Warehouse

**SQL Report 1: Top 100 Customers – Last Quarter (by Revenue)**

* A ..

**SQL Report: Revenue Per Plan Per Quarter**

* A ..

**SQL Report 3: Top 100 Customers – Last Quarter (by Activity)**

* A ..

**SQL Report 4: Top 20 Customers – Revenue Patterns (by Quarter)**

* A …

# Section C: Machine Learning Using SQL

## Overview of ML Process

In the..

## Creating and Populating Fact Table for ML Modelling

In the..

## Preparing Training and Test Data Sets

In the..

## Create VIEW for Predicted Values

In the..

## Evaluating Models

In the..

# Appendices

## Appendix 1 – SQL Scripts to build the Data Warehouse

In the..

## Appendix 2 – SQL Scripts to Populate Data Warehouse Dimensions

The Dimension tables in the Data Warehouse are populated..

## Appendix 3 – SQL Scripts To Populate Data Warehouse Fact Table

The Fact table in the Data Warehouse is populated in the by executing the SQL commands in the attached \*.SQL file.

## Appendix 4 – SQL Scripts For All SQL Queries

The

## Appendix 5 – SQL Scripts For ML Process

The

# References

## Data Warehouse Design

In additional to the class Module notes I followed the data warehouse design principles that were discussed in these two YouTube training videos;

***Designing Your Data Warehouse from the Ground Up* -** <https://youtu.be/patBYUGwsHE>

***Implementing a Data Warehouse with SQL Server, 01, Design and Implement Dimensions and Fact Tables* -** <https://youtu.be/StoWu2A8Ufs>