**The British College**

**KATHMANDU**

**Coursework Submission Coversheet**(individual coursework only)

**Faculty of Arts, Environment and Technology LBU Student Id:**

77178367

**For checking by the student:**

Please ensure all information is complete and correct and attach this form securely to the front of your work before posting it in a coursework collection box.

Award name: BSc (Hons) Computing

Module code: 23269 – 201516

Module name: Advanced Database A

Module run: 2016

Coursework title: 1

Due Date: 18 September (2016)

Module leader: (In LBU) Jackie Campbell, Sanela Lazarevski

Module tutor: (In TBC) Dibya Tara Shakya

**TURNITIN** Checked: YES NO ***(please circle)***

Submission date& time: Date: 18 September (2016) Time: 10 PM

**Total Word Count: Total Number of Pages (including this front sheet):**

**In submitting this form with your assignment, you make the following declaration:**  
I declare, that the coursework submitted is my own work and has not (either in whole or part) been submitted towards the award of any other qualification either at LBU or elsewhere. I have fully attributed/referenced all sources of information used during the completion of my assignment, and I am aware that failure to do so constitutes an assessment offence.

Signed: Date:

**You are strongly advised to retain a second copy of your work in case of any query about the assignment.**

5885

36

**For completion by the faculty:**

**This mark is provisional and subject to moderation and approval by the relevant examining board**

**Teacher's Feedback**

**Teacher's Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**This mark is provisional and subject to moderation and approval by the relevant examining board**

**Teacher's Feedback**

**This mark is provisional and subject to moderation and approval by the relevant examining board**

**Teacher's Feedback**

# Table of Contents

[Table of Contents 1](#_Toc463032080)

[Task 1 3](#_Toc463032081)

[Introduction 3](#_Toc463032082)

[Rationale 3](#_Toc463032083)

[Possible Report Requirements 3](#_Toc463032084)

[Data warehouse 3](#_Toc463032085)

[Data Warehouse Architecture 4](#_Toc463032086)

[Data Mart 5](#_Toc463032087)

[Business Intelligence (BI) 5](#_Toc463032088)

[OLTP versus OLAP 5](#_Toc463032089)

[Approach 6](#_Toc463032090)

[Top Down Vs Bottom Up Approach 7](#_Toc463032091)

[Top Down 7](#_Toc463032092)

[Bottom Up 7](#_Toc463032093)

[Appropriate approach 7](#_Toc463032094)

[Methodology 8](#_Toc463032095)

[Star Schema 8](#_Toc463032096)

[Star Schema architecture 8](#_Toc463032097)

[Key Characteristics of Star Schema 8](#_Toc463032098)

[Why Star Schema 8](#_Toc463032099)

[Star Schema Design Model 9](#_Toc463032100)

[Proposed data dictionary 10](#_Toc463032101)

[Dimension Tables 10](#_Toc463032102)

[Fact Tables 11](#_Toc463032103)

[Decision on granularity of data 12](#_Toc463032104)

[Proposed Data maintenance technique 12](#_Toc463032105)

[Proposed slowly changing dimension technique 13](#_Toc463032106)

[Before applying SCD approach 13](#_Toc463032107)

[After applying SCD approach 14](#_Toc463032108)

[ELT Script Design 14](#_Toc463032109)

[Addressing Data Quality 14](#_Toc463032110)

[Measure Calculation 14](#_Toc463032111)

[ETL Scripts 15](#_Toc463032112)

[DW Interface Suggestions 16](#_Toc463032113)

[Summary 16](#_Toc463032114)

[Task 2 A 17](#_Toc463032115)

[Proposed OLAP model 17](#_Toc463032116)

[QSEE generated scripts 17](#_Toc463032117)

[Summary 27](#_Toc463032118)

[Task 2 B 28](#_Toc463032119)

[Insertion of data from CSV file 28](#_Toc463032120)

[OLTP data dictionary for Staging server 28](#_Toc463032121)

[CSV File 29](#_Toc463032122)

[Importing File using APEX 29](#_Toc463032123)

[Creating New user 29](#_Toc463032124)

[Login into Apex 29](#_Toc463032125)

[Creating Workspace and user 30](#_Toc463032126)

[Login into Workspace 30](#_Toc463032127)

[Importing CSV data 30](#_Toc463032128)

[Summary 34](#_Toc463032129)

[Bibliography 35](#_Toc463032130)

# Task 1

# Introduction

Leeds city council are in charge of arrangement of transportation in the Leeds region. They have to ensure the excellency of services, roads and lighting. Now to work on their new “healthy and environmentally friendly” initiative they have planned on data warehouse project. City has two different type of transportation related records available with them. First data details, number of vehicle in the road while next data provides detailed number of all road based incidence. These existing data would be utilized to advancement the state of transportation. This document would analyze the reports requirement of “Transport Department” and discuss need of warehousing project while proposing warehouse solution.

# Rationale

To generate reports based on transport department requirements, DW analyst must study available data and then design data warehouse solution. DW to be developed will enable transport department to analyze and query historical data. Stake holder would utilize generated reports to manage as well as improve the state of transportation service of the region.

# Possible Report Requirements

1. Generate report on number of vehicles on particular road in particular year.
2. Generate report on type of road mostly utilized on particular year.
3. Generate report on comparison between number of different vehicle on particular road.
4. Generate report on preferred vehicle type based on link length.
5. Generate report on most crowded road type on certain year.
6. Generate report on least crowded road in certain year.
7. Generate report on number of certain vehicle type in certain year inside Yorkshire region.
8. Generate report on maximum number of vehicle under Leeds local authority.

# Data warehouse

A data warehouse is analysis oriented database architecture design. Data warehouse is designed to store business data from different sources such as operational database, transactional database and other external sources. This enables DW to store data in swift and collective system. This system then is utilized by business to analyze and generate reports according to their needs. In simpler form data warehouse is unified, issue oriented, non-volatile and time varying gathering of data to support business decision process. The requirement of DW is eminent for larger business where running complicated queries or perform analysis job on transactional database would hamper the performance of the system. To avoid this analytic jobs are performed on warehouse system and transactional databases are left to concenter on transaction job.

## Data Warehouse Architecture

Data warehouse can have different architecture based on the requirements. One data warehouse may have small data sources while other have larger number of data source. Figure 1 below demonstrates the basic architecture of a data warehouse solution.

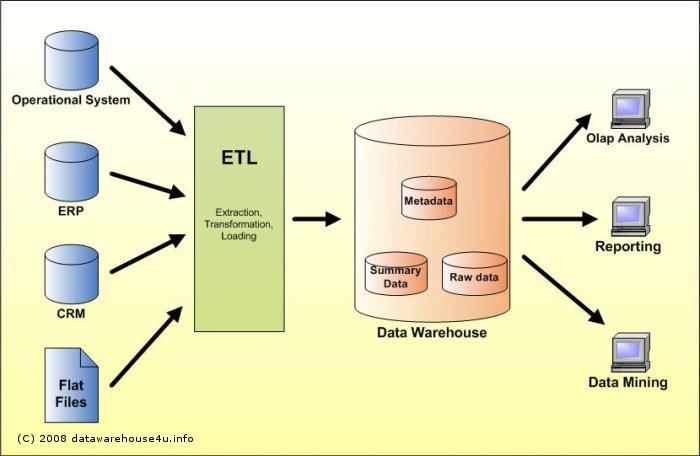


Figure Architectural Framework of a Data Warehouse (Source: http://datawarehouse4u.info/)

Operational source systems are used for everyday businesses of an organization. Data integrity are well maintained and transactions runs smoothly and efficiently in such sources. Activities such as inserting new data or update existing data are very fast. Leeds city council already have this types of sources such as incidence reporting system IRS. Next phase of data warehousing is ETL. This is process of data extraction, transformation and then loading into warehouse. Extract phase basically extract data from various sources for transformation phase. Transformation phase then applies business rules on extracted data to provide standard format. And finally, loading phase loads cleaned and transformed data into data warehouse. The final phase of data warehousing is presentation area. Here different data marts retrieve data from main data warehouse. Data marts can be OLAP analysis, reporting service or data mining etc.

## Data Mart

It is often misunderstood that data mart are synonym to data warehouse. But that is not correct, it is rather subset of DW that serves demands of specific department. Whereas, DW is central repository of all data. Data warehouse project for Leeds city council will be designed and developed to address requirements of transport department. Hence no data mart for transportation department is required to be developed as DW itself can serve the demand of stakeholder. However, if needed additional data marts such as cycle data mart or health and safety data mart can be developed later to retrieve necessary data from warehouse.

## Business Intelligence (BI)

Business Intelligence is data warehouse based technology that allows to get maximum amount of required information from available information. This is done for the purpose of improving various business developments. Data available with Leeds city council would be gathered then cleaned, integrated then analyzed according to stakeholder needs. Then it would be shared to generate reports. There are various kinds of BI solutions are available such as Management Information System (MIS), Customer Relation Management (CRM) or Online Analytical Processing and Multidimensional Analysis (OLAP) etc. This document would propose OLAP solution that would enable its stakeholder to analyze the available data to improve the state of transportation services.

## OLTP versus OLAP

In general terms, On-line transactional processing has main characteristics of having larger number of transactions such as insert/update or delete. Most of the software used on daily basis are OLTP based system. On the other hand, On-line Analytical processing (OLAP) has smaller number of transaction but has more complex queries. Figure 2 below is simpler demonstration of OLTP and OLAP role in businesses. While OLTP concentrate on quick query processing, OLAP concentrates on data analysis.

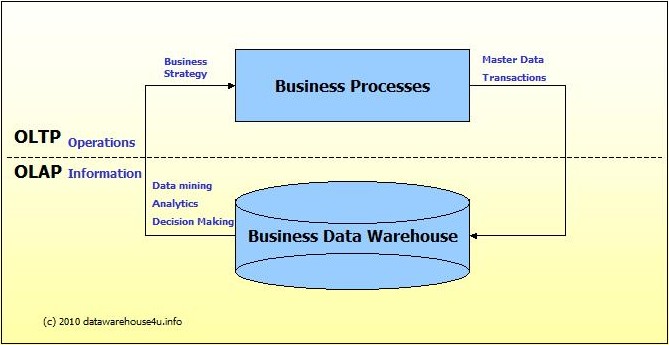


Figure Basic relation between OLTP and OLAP Source: [www.datawarehouse4u.info](http://www.datawarehouse4u.info)

Key difference between OLAP and OLTP systems are described through table below.

|  |  |  |
| --- | --- | --- |
| Difference Between OLAP and OLTP | | |
| Criteria | **OLAP** | **OLTP** |
| Data Source | Old data | Operative data |
| Query Complexity | Completive/Complex | Simple |
| Size | Large | Smaller |
| Focus | Analysis and reporting | Updating and data integrity |
| Speed | Slow | Fast |
| Schema | DE normalized- Snow flex, star | Highly Normalized |
| Tools | Management system, Decision systems | CRM, ERP |
| Model | Multi-dimensional | Single |

Leeds city council have OLTP data available that are required to develop an OLAP system. This OLAP solution would generate reports as per stake holder’s requirements. To design and develop an OLAP solution there are basically two major approaches available. These available approaches and suitable approach for this project is discussed below.

# Approach

When designing warehouses, the most popular and discussed approaches are Inmon’s (also known as Top Down) approach and Kimball’s (also known as Bottom Up) approach. Both approaches have their merits and demerits and are used based on scenarios (requirements). To develop warehouse project for transport department of Leeds city council, suitable approach must be selected. Key differences between these two approaches are shown in table 1 below.

## Top Down Vs Bottom Up Approach

|  |  |
| --- | --- |
| Top Down | Bottom Up |
| 3NF normalized warehouse is designed first. | Warehouses are generally in de-normalized form. |
| Data marts are created from DW. | Data marts are created first. |
| Data warehouse is centralized repository. |  |
| Once data warehouse is fully designed then only different Data Marts for different departments are created. | Data marts are combined together later to create data warehouse. |
| DW works as context for business intelligence | Also known as Data warehouse bus. |
| DW are time variant, nonvolatile, integrated and subject oriented. | Provides high level performance on data warehouse. |
| Uses ER tools to design warehouse. | Uses methods such as star/snow flex to design dimension models. |

Table 1Top down vs Bottom Up Approach

# Appropriate approach

Table 2 below would allow to understand and find the appropriate approach for the current project.

|  |  |  |
| --- | --- | --- |
| Criteria | Top Down | Bottom Up |
| Perseverance source data | Source data is fluctuating at high rate. | Source is stable |
| Skill requirements | Specialists team and bigger number of member is required | Small team is required |
| Cost | Higher startup cost | Lower startup cost |
| Time schedule | Smaller time for first warehouse | Requires larger time scale |

Table 2Choosing appropriate approach

Table 2 above shows for a project that has limited time frame, budget, skill requirements and less chance of data changes, Kimball’s Bottom up approach is appropriate. Warehouse project for Leeds city council will adopt this approach to design and develop data ware house system.

# Methodology

# Star Schema

While designing a warehouse based on Kimball’s bottom up approach star schema is simplest form of schema. Star name is given as dimension tables and fact table connected though lines and resemble star. Star Schema consists of de-normalized dimension tables. Due to its simples form it is most used schema.

## Star Schema architecture

Figure 3 below demonstrates basic architecture of a star schema based warehouse for a sales store. Here dimension tables categorize data and represent tables while fact table consists of numbering facts and foreign key of dimension tables. Star schema is relational schema that embodies multi-dimensional data. There can be number of fact tables as well as dimension tables.

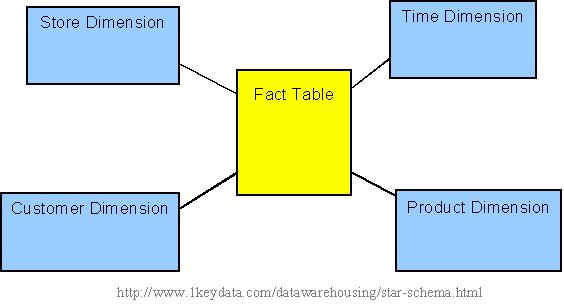


Figure Basic Architecture of star schema Source: [www.1keydata.com](http://www.1keydata.com)

## Key Characteristics of Star Schema

Some of the key characteristic of star schema are as follows:

* Each dimension tables have key to identify them i.e. primary key
* Fact table will contain foreign key for the each of its dimension tables
* Dimension tables must not contain foreign key of another dimension/parent table

# Why Star Schema

To design and develop warehouse project for Transport department of Leeds city council, Star schema methodology has been selected as it has simple architecture and requires less effort compared to Top down model and snowflake schema. Here, Snowflake schema is similar to star schema but dimension tables contains foreign key of their parent tables. This allows to manage data efficiently but loses performance and creates complicity with queries. Here are some of the benefits of choosing star schema for warehousing.

* Its simple architecture allows design and development period of warehouse to be shorter.
* It is less complicated to execute queries compared to snowflake schema and transactional tables.
* Query performance is relatively better than OLTP solutions as it has lesser number of tables. And it has less hierarchical architecture.
* Design/development and load of large data requires lesser time. Impact of load is minimal as tables are separated into fact and dimension tables.
* This schema model is relational system where fact tables and dimension tables are in relation. This allows to maintain referential integrity.

# Star Schema Design Model

Star schema design model for the data ware house project is designed using QSEE software and proposed in figure 4 below.

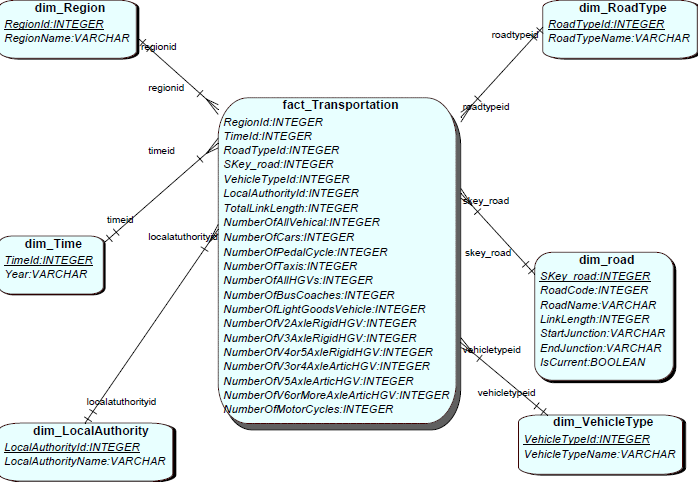


Figure Star schema model for warehouse project

# Proposed data dictionary

Now the suitable methodology and approach has been selected required data dictionary for the data warehousing needs to be planned and proposed. Based on Kimball’s bottom up approach, first dimension tables are planned using star schema method.

# Dimension Tables

To describe objects of fact table, dimension tables are developed that would store dimensions or attributes. Required dimension tables for the warehousing project are listed in data dictionary below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim\_Region | | | To store region dimension | | | |
| Attribute Name | **Type** | **Null able?** | | **Key?** | **Size** | **Remarks** |
| RegionId | Integer | False | | True |  |  |
| RegionName | Varchar | False | | False | 50 |  |

Table 3 Road dimension table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim\_Time | | | To store time dimension | | | |
| Attribute Name | **Type** | **Null able?** | | **Key?** | **Size** | **Remarks** |
| TimeId | Integer | False | | True |  |  |
| Year | Varchar | False | | False | 5 |  |

Table 4 Time dimension table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim\_VehicleType | | | To store vehicle type dimension | | | |
| Attribute Name | **Type** | **Null able?** | | **Key?** | **Size** | **Remarks** |
| VehicleTypeId | Integer | False | | True |  |  |
| VehicleTypeName | Varchar | False | | False | 20 |  |

Table 5 Vehicle Type dimension table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim\_LocalAuthority | | | To store local authority dimension | | | |
| Attribute Name | **Type** | **Null able?** | | **Key?** | **Size** | **Remarks** |
| LocalAuthorityId | Integer | False | | True |  |  |
| LocalAuthorityName | Varchar | False | | False | 20 |  |

Table 6Local Authority dimension table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim\_RoadCategory | | | To store road category dimension | | | |
| Attribute Name | **Type** | **Null able?** | | **Key?** | **Size** | **Remarks** |
| RoadTypeId | Integer | False | | True |  |  |
| RoadTypeName | Varchar | False | | False | 20 |  |

Table 7 Road Category dimension Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dim\_Road | | | To store road dimension | | | |
| Attribute Name | **Type** | **Null able?** | | **Key?** | **Size** | **Remarks** |
| Skey\_road | Interger | False | | True |  |  |
| RoadId | Integer | False | | False |  |  |
| RoadName | Varchar | False | | False | 20 |  |
| LinkLength | Integer | True | | False |  |  |
| StartJunction | Varchar | True | | False | 20 |  |
| EndJunction | Varchar | True | | False | 20 |  |

Table 8 Road dimension table

# Fact Tables

The de-normalized and central table of ware house is fact table that stores summative information and keys of each dimension tables. Required data dictionary of fact table for transport department data warehouse project is proposed in table 9 below.

|  |  |  |
| --- | --- | --- |
| Fact\_Transportation | | |
| Field | **Type** | **Description** |
| RegionId | Integer | Links fact to Region dimension |
| TimeId | Integer | Links fact to Time dimension |
| RoadTypeId | Integer | Links fact to Road Type dimension |
| Skey\_road | Integer | Links fact to Road dimension |
| VehicleTypeId | Integer | Links fact to Vehicle Type dimension |
| LocalAuthorityId | Integer | Links fact to local authority dimension |
| TotalLinkLength | Integer | Gets total road length |
| NumberOfAllVehicles | Integer | Gets number of All Vehicles |
| NumberOfCars | Integer | Gets number of Cars |
| NumberOfPedalCycle | Integer | Gets number of Pedal Cycle |
| NumberOfTaxis | Integer | Gets number of Taxis |
| NumberOfAllHGVs | Integer | Gets number of All HGVs |
| NumberOfBusCoaches | Integer | Gets number of Bus Coaches |
| NumberOfLightGoodsVehicle | Integer | Gets number of LightGoodsVehicle |
| NumberOfV2AxleRigidHGV | Integer | Gets number of V2AxleRigidHGV |
| NumberOfV3AxleRigidHGV | Integer | Gets number of V3AxleRigidHGV |
| NumberOfV4or5AxleRigidHGV | Integer | Gets number of V4or5AxleRigidHGV |
| NumberOfV3or4AxleArticHGV | Integer | Gets number of V3or4AxleArticHGV |
| NumberOfV5AxleArticHGV | Integer | Gets number of V5AxleArticHGV |
| NumberOfV6orMoreAxleArticHGV | Integer | Gets number of V6orMoreAxleArticHGV |

Table 9 Fact table for Warehouse

# Decision on granularity of data

The frequency or granularity of data is regulated by the level of granularity of dimension tables of the system. Fact table may store monthly or yearly total of vehicles. If data are stored on monthly basis, more records will be stored in fact table. This will allow user to get reports on more depth. In simple word granularity describes the level of data detail in warehouse project.

Based on the data available with Leeds city council, data granularity of warehousing project is set to store records on yearly basis. This will allow transport department to generate report on yearly basis, which is detailed enough to analyze the necessary facts and improve the state of transportation of region. Summary of numbers of all vehicles as well as different types of vehicles are stored in different fields of fact table.

# Proposed Data maintenance technique

Maintenance of data and data warehouse itself is a big challenge. Before maintaining the data, maintenance of warehouse should also be considered. Schema of warehouse can be stored in a dump file that can be restored in case of some system issues during development of warehouse.

For the data maintenance, various other procedures can be suggested. Logs can be maintained of each actions performed in DW to track the faults. Another key aspect of data maintenance is proper system testing before implementation. Data warehouse for Leeds city council would be thoroughly tested before deployed. Finally, to maintain data properly, different users will be created on different roles. User would be able to perform task on warehouse based on their role. This would help to strengthen the safety of data.

Another maintenance strategy can be having proper backup solution for warehouse. Proper business logics should be written to maintain backup and recovery of data in the scenario of data disaster.

And finally even though current data warehouse utilizes single disk drive, proposal can be made to use multiple disks. As the size of warehouse increases, more number of disks can be utilized. Tables and indexes can also be distributed intro separated to different disks. This would offer better data organization and better I/O balancing.

# Proposed slowly changing dimension technique

Over the time dimensions slowly changes its characteristics or attributes rather on regular interval. These changes are required to be tracked properly to avoid complications while analyzing and reporting historic data. Three of the most popular approaches are discussed here. In first approach (**Type One**) historical data is simple overwritten. The old information gets lost and no track of it is kept. Instead of inserting new data it updates old ones. In second approach (**Type Two**) old data are kept and new data are inserted. This preserves the dimensional data and enables to keep track of changes in attribute data. Third type of approach (**Type Three**) stores previous dimension data and new dimension data. This process is done by adding new column in dimension table. As the third, fourth and so on changes occurs size of dimension table also increases significantly.

Warehouse project for Leeds city council requires suitable SDC approach to maintain slowly changing track. After studying the available data and requirement of the system Combination of type **1,2 and 3** approaches has been selected to address slowly changing dimension. In this approach old data is preserved, current dimension data is overwritten and historical data are stored and dedicated column.

Information about road may change over time and must be tracked carefully. Name of road can change or link length may change. To address such issue, slowly changing dimension approach is applied on road dimension table. For example, name of the road M1 has now been changed to M900. Now to make change in database and keep track of such changes SCD approach is implemented. Such implementation is demonstrated in tables 10 and 11 blow.

## Before applying SCD approach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Road Id | Road Name | Link Length | Start Junction | EndJunction |
| 1 | M1 | 2.36 | M 18 Spur | 33 |

Table 10Old data

## After applying SCD approach

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Skey  \_road | Road  Id | Road  Name | Link  Length | Start  Junction | End  Junction | Effective  From | Effective  To | Current |
| 1 | 1 | M1 | 2.36 | M 18 Spur | 33 | 15-JAN-2000 | 15-JAN-20015 | N |
| 2 | 1 | M900 | 2.36 | M 18 Spur | 33 | 15-FEB-2015 |  | Y |

Table 11Newdata and old data after applying SCD approach

In table 4 above, both old and new information of road is stored in database. This allows to keep track of changes in information. To identify different rows, surrogate key is utilized.

# ELT Script Design

## Addressing Data Quality

Data quality is a very important aspect that needs to be considered while designing data warehouse system. Data in the system should be sufficiently good enough to support decision making and analytical purposes. Data warehouse to be developed would generate reports for transport department on yearly basis. For this, vehicle reports on yearly range are need. To address this, dimension table of time is needed with time key and year name. Vehicle number record would be updated on yearly basis. Another aspect of data quality is relevance. Only necessary information from existing reports would be imported into a staging table. From there, information would be extracted, transformed using necessary rules and then loaded into dimension tables.

During the ETL process, various business rules would be implemented to ensure data quality. This also includes proper use of foreign/primary keys. Furthermore, proper slowly changing dimension approach is applied to ensure quality of data after future changed. Ensuring good data quality means reports generated from warehouse would be reliable for the development of transportation structure of region under Leeds city council.

## Measure Calculation

Fact table in warehouse system consist of various fact details attributes. They store the summative fact records. To calculate and store such measurements various aggregated/analytical functions will be used during ETL process. For example, Total number of vehicle would be stored in fact table. Sum aggregated function would return total number of vehicles that will be stored in fact table. Similarly, all calculations would be done during ETL scripting process to store necessary summaries.

# ETL Scripts

Required ETL scripts (DDL scripts) for Leeds city council warehouse project is propose in table 12 below.

|  |  |
| --- | --- |
| Table | ETL Script |
| dim\_Region | CREATE TABLE dim\_Region(  RegionId INTEGER NOT NULL,  RegionName VARCHAR(50) NOT NULL,  CONSTRAINT pk\_dim\_Region PRIMARY KEY (RegionId)); |
| dim\_RoadType | CREATE TABLE dim\_RoadType(  RoadTypeId INTEGER NOT NULL,  RoadTypeName VARCHAR(20) NOT NULL,  CONSTRAINT pk\_dim\_RoadType PRIMARY KEY (RoadTypeId)); |
| dim\_road | CREATE TABLE dim\_road(  SKey\_road INTEGER NOT NULL,  RoadCode INTEGER NOT NULL,  RoadName VARCHAR(20) NOT NULL,  LinkLength INTEGER,  StartJunction VARCHAR(20),  EndJunction VARCHAR(20),  IsCurrent NUMBER(1) NOT NULL,  CONSTRAINT pk\_dim\_road PRIMARY KEY (SKey\_road)  ); |
| dim\_Time | CREATE TABLE dim\_Time(  TimeId INTEGER NOT NULL,  Year VARCHAR(8) NOT NULL,  CONSTRAINT pk\_dim\_Time PRIMARY KEY (TimeId)); |
| dim\_VehicleType | CREATE TABLE dim\_VehicleType(  VehicleTypeId INTEGER NOT NULL,  VehicleTypeName VARCHAR(20) NOT NULL,  CONSTRAINT pk\_dim\_VehicleType PRIMARY KEY (VehicleTypeId)); |
| dim\_LocalAuthority | CREATE TABLE dim\_LocalAuthority(  LocalAuthorityId INTEGER NOT NULL,  LocalAuthorityName VARCHAR(20) NOT NULL,  CONSTRAINT pk\_dim\_LocalAuthority PRIMARY KEY (LocalAuthorityId)); |
| fact\_Transportation | CREATE TABLE fact\_Transportation(  RegionId INTEGER,  TimeId INTEGER,  Skey\_road INTEGER,  RoadId INTEGER,  VehicleTypeId INTEGER,  LocalAuthorityId INTEGER,  TotalLinkLength INTEGER,  NumberOfAllVehical INTEGER,  NumberOfCars INTEGER,  NumberOfPedalCycle INTEGER,  NumberOfTaxis INTEGER,  NumberOfAllHGVs INTEGER,  NumberOfBusCoaches INTEGER,  NumberOfLightGoodsVehicle INTEGER,  NumberOfV2AxleRigidHGV INTEGER,  NumberOfV3AxleRigidHGV INTEGER,  NumberOfV4or5AxleRigidHGV INTEGER,  NumberOfV3or4AxleArticHGV INTEGER,  NumberOfV5AxleArticHGV INTEGER,  NumberOfV6orMoreAxleArticHGV INTEGER,  NumberOfMotorCycles INTEGER  ); |

Table 12DDL scripts for ETL

# DW Interface Suggestions

Apex Reporting tools will be utilized to design the reporting interfaces for data warehouse project. These user friendly interfaces would display the friendly reports to transport department that would enable them to analyze and understand the necessary developments and upgrade requirements.

DW interface should include various reports such as number of vehicles on particular road or statistics of road use on particular year. DW should be able to generate different report interface for different reports for the ease of user.

# Summary

To support business decision of Leeds city council, proposed data warehouse should generate various reports and enable transport department to carryout analysis on historical data. This report has identified potential reports requirement of stakeholders. Various methodologies or approaches were taken into consideration before choosing appropriate solution for warehousing. Furthermore, this document proposed data dictionary and star schema model for ware house. This model will be utilized to develop warehouse during development phase of warehouse development.

# Task 2 A

# Proposed OLAP model

Designed ER diagram during star schema model proposal is shown in figure 5 below.

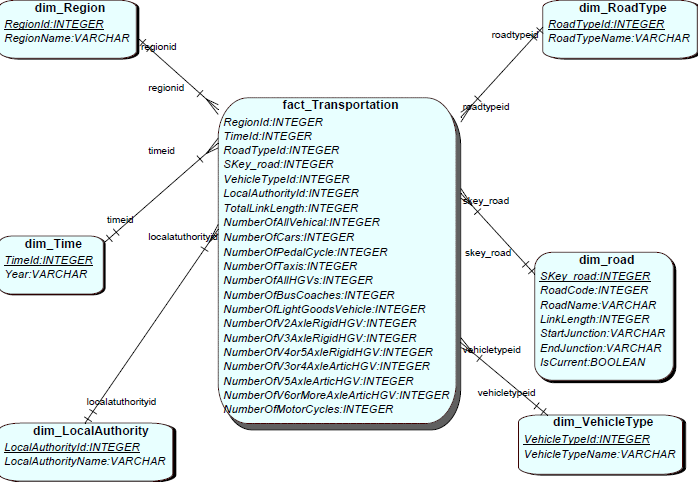


Figure Proposed Star schema model for warehouse

## QSEE generated scripts

Data Definition Language (DDL) scripts for developing dimension and fact table can be created using QSEE SQL script generator. QSEE generated scripts and their execution reports are attached here.

--------------------------------------------------------------

-- Database creation Script

-- Auto-Generated by QSEE-SuperLite (c) 2001-2004 QSEE-Technologies Ltd.

-- Verbose generation: ON

-- note: spaces within table/column names have been replaced by underscores (\_)

-- Target DB: SQL2

-- Entity Model :Entity Relationship Diagram

-- To drop the tables generated by this script run -

-- 'C:\Users\Atut\Desktop\transport\_drop.sql'

--------------------------------------------------------------

--------------------------------------------------------------

-- Table Creation --

-- Each entity on the model is represented by a table that needs to be created within the Database.

-- Within SQL new tables are created using the CREATE TABLE command.

-- When a table is created its name and its attributes are defined.

-- The values of which are derived from those specified on the model.

-- Certain constraints are sometimes also specified, such as identification of primary keys.

-- Create a Database table to represent the "dim\_Region" entity.

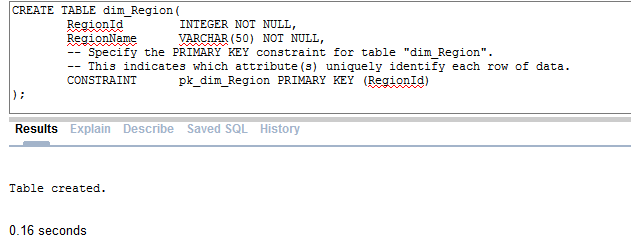


Figure Creating Region Dimension

-- Create a Database table to represent the "dim\_RoadType" entity.

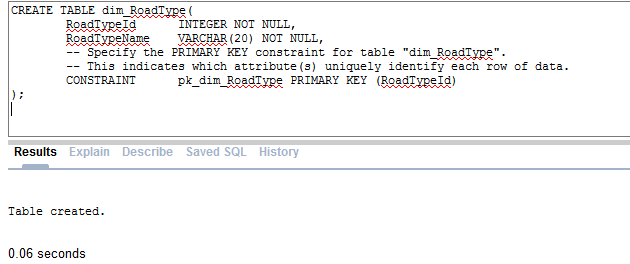


Figure Creating Roadtype Dimension

-- Create a Database table to represent the "dim\_road" entity.

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| CREATE TABLE dim\_road(  SKey\_road INTEGER NOT NULL,  RoadCode INTEGER NOT NULL,  RoadName VARCHAR(20) NOT NULL,  LinkLength INTEGER,  StartJunction VARCHAR(20),  EndJunction VARCHAR(8),  IsCurrent BOOLEAN NOT NULL,  -- Specify the PRIMARY KEY constraint for table "dim\_road".  -- This indicates which attribute(s) uniquely identify each row of data.  CONSTRAINT pk\_dim\_road PRIMARY KEY (SKey\_road)  ); | CREATE TABLE dim\_road(  SKey\_road INTEGER NOT NULL,  RoadCode INTEGER NOT NULL,  RoadName VARCHAR(20) NOT NULL,  LinkLength INTEGER,  StartJunction VARCHAR(20),  EndJunction VARCHAR(8),  IsCurrent Number(1) NOT NULL,    CONSTRAINT pk\_dim\_road PRIMARY KEY (SKey\_road)  ); | As oracle does not support Boolean data type, for defining current flag, 1 or 0 number is stored in database. Hence number (1) type is given to current column. |
| Figure Creating road dimension | | |

Table 13Modifying Generated script to create road dimension

-- Create a Database table to represent the "dim\_Time" entity.

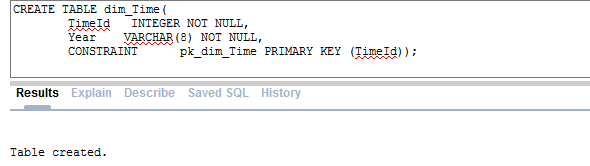


Figure Creating Time dimension

-- Create a Database table to represent the "dim\_VehicleType" entity.

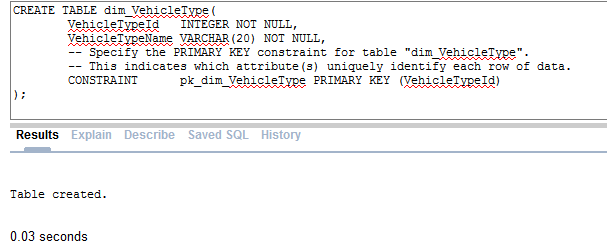


Figure Creating vehicle type dimension

-- Create a Database table to represent the "dim\_LocalAuthority" entity.

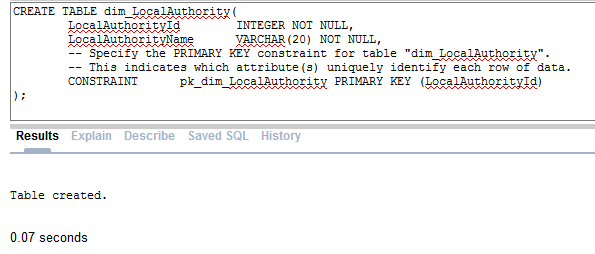


Figure Creating local authority dimension

-- Create a Database table to represent the "fact\_Transportation" entity.

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| CREATE TABLE fact\_Transportation(  RegionId INTEGER,  TimeId INTEGER,  RoadTypeId INTEGER,  Skey\_road INTEGER,  VehicleTypeId INTEGER,  LocalAuthorityId INTEGER,  TotalLinkLength INTEGER,  NumberOfAllVehical INTEGER,  NumberOfCars INTEGER,  NumberOfPedalCycle INTEGER,  NumberOfTaxis INTEGER,  NumberOfAllHGVs INTEGER,  NumberOfBusCoaches INTEGER,  NumberOfLightGoodsVehicle INTEGER,  NumberOfV2AxleRigidHGV INTEGER,  NumberOfV3AxleRigidHGV INTEGER,  NumberOfV4or5AxleRigidHGV INTEGER,  NumberOfV3or4AxleArticHGV INTEGER,  NumberOfV5AxleArticHGV INTEGER,  NumberOfV6orMoreAxleArticHGV INTEGER,  NumberOfMotorCycles INTEGER,  fk1\_RoadTypeId INTEGER NOT NULL,  fk2\_LocalAuthorityId INTEGER NOT NULL,  fk3\_TimeId INTEGER NOT NULL,  fk4\_RegionId INTEGER NOT NULL,  fk5\_Skey\_road INTEGER NOT NULL,  fk6\_VehicleTypeId INTEGER NOT NULL  ); | CREATE TABLE fact\_Transportation(  RegionId INTEGER,  TimeId INTEGER,  RoadTypeId INTEGER,  Skey\_road INTEGER,  VehicleTypeId INTEGER,  LocalAuthorityId INTEGER,  TotalLinkLength INTEGER,  NumberOfAllVehical INTEGER,  NumberOfCars INTEGER,  NumberOfPedalCycle INTEGER,  NumberOfTaxis INTEGER,  NumberOfAllHGVs INTEGER,  NumberOfBusCoaches INTEGER,  NumberOfLightGoodsVehicle INTEGER,  NumberOfV2AxleRigidHGV INTEGER,  NumberOfV3AxleRigidHGV INTEGER,  NumberOfV4or5AxleRigidHGV INTEGER,  NumberOfV3or4AxleArticHGV INTEGER,  NumberOfV5AxleArticHGV INTEGER,  NumberOfV6orMoreAxleArticHGV INTEGER,  NumberOfMotorCycles INTEGER  ); | Unneccessary attributes generated by QSEE |
|  | | |

Table 14Modifying QSEE generated script to create fact table

--------------------------------------------------------------

-- Alter Tables to add fk constraints --

-- Now all the tables have been created the ALTER TABLE command is used to define some additional

-- constraints. These typically constrain values of foreign keys to be associated in some way

-- with the primary keys of related tables. Foreign key constraints can actually be specified

-- when each table is created, but doing so can lead to dependency problems within the script

-- i.e. tables may be referenced before they have been created. This method is therefore safer.

-- Alter table to add new constraints required to implement the "fact\_Transportation\_dim\_RoadType" relationship

-- This constraint ensures that the foreign key of table "fact\_Transportation"

-- correctly references the primary key of table "dim\_RoadType"

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| ALTER TABLE fact\_Transportation ADD CONSTRAINT fk1\_fact\_Transportation\_to\_dim\_RoadType FOREIGN KEY(fk1\_RoadTypeId) REFERENCES dim\_RoadType(RoadTypeId) ON DELETE RESTRICT ON UPDATE RESTRICT; | ALTER TABLE fact\_Transportation ADD CONSTRAINT fk\_fact\_Trans\_RoadType FOREIGN KEY(RoadTypeId)  REFERENCES dim\_RoadType(RoadTypeId); | Foreign key identifier is too long and name of the attribute is not correct. |
| Figure Creating relation between fact and road type | | |

Table 15Modifying QSEE generated script to create relation between fact and road type dimension

-- Alter table to add new constraints required to implement the "fact\_Transportation\_dim\_LocalAuthority" relationship

-- This constraint ensures that the foreign key of table "fact\_Transportation"

-- correctly references the primary key of table "dim\_LocalAuthority"

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| ALTER TABLE fact\_Transportation ADD CONSTRAINT fk2\_fact\_Transportation\_to\_dim\_LocalAuthority FOREIGN KEY(fk2\_LocalAuthorityId) REFERENCES dim\_LocalAuthority(LocalAuthorityId) ON DELETE RESTRICT ON UPDATE RESTRICT; | ALTER TABLE fact\_Transportation ADD CONSTRAINT fK\_TransLAuthority FOREIGN KEY(LocalAuthorityId)  REFERENCES dim\_LocalAuthority(LocalAuthorityId); | Foreign key identifier is too long and name of the attribute is not correct. |
|  | | |

Table 16 Modifying QSEE generated script to create relation between fact and local Authority dimension

ALTER TABLE fact\_Transportation ADD CONSTRAINT fk2\_fact\_Transportation\_to\_dim\_LocalAuthority FOREIGN KEY(fk2\_LocalAuthorityId) REFERENCES dim\_LocalAuthority(LocalAuthorityId) ON DELETE RESTRICT ON UPDATE RESTRICT;

-- Alter table to add new constraints required to implement the "fact\_Transportation\_dim\_Time" relationship

-- This constraint ensures that the foreign key of table "fact\_Transportation"

-- correctly references the primary key of table "dim\_Time"

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| ALTER TABLE fact\_Transportation ADD CONSTRAINT fk3\_fact\_Transportation\_to\_dim\_Time FOREIGN KEY(fk3\_TimeId) REFERENCES dim\_Time(TimeId) ON DELETE RESTRICT ON UPDATE RESTRICT; | ALTER TABLE fact\_Transportation ADD CONSTRAINT fkTransTime FOREIGN KEY(TimeId) REFERENCES dim\_Time(TimeId); | Foreign key identifier is too long and name of the attribute is not correct. |
|  | | |

Table 17Modifying QSEE generated script to create relation between fact and Time dimension

-- Alter table to add new constraints required to implement the "fact\_Transportation\_dim\_Region" relationship

-- This constraint ensures that the foreign key of table "fact\_Transportation"

-- correctly references the primary key of table "dim\_Region"

ALTER TABLE fact\_Transportation ADD CONSTRAINT fk4\_fact\_Transportation\_to\_dim\_Region FOREIGN KEY(fk4\_RegionId) REFERENCES dim\_Region(RegionId) ON DELETE RESTRICT ON UPDATE RESTRICT;

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| ALTER TABLE fact\_Transportation ADD CONSTRAINT fk4\_fact\_Transportation\_to\_dim\_Region FOREIGN KEY(fk4\_RegionId) REFERENCES dim\_Region(RegionId) ON DELETE RESTRICT ON UPDATE RESTRICT; | ALTER TABLE fact\_Transportation ADD CONSTRAINT fkTranRegionId  FOREIGN KEY(RegionId) REFERENCES dim\_Region(RegionId); | Foreign key identifier is too long and name of the attribute is not correct. |
|  | | |

Table 18Modifying QSEE generated script to create relation between fact and region dimension

-- Alter table to add new constraints required to implement the "fact\_Transportation\_dim\_road" relationship

-- This constraint ensures that the foreign key of table "fact\_Transportation"

-- correctly references the primary key of table "dim\_road"

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| ALTER TABLE fact\_Transportation ADD CONSTRAINT fk5\_fact\_Transportation\_to\_dim\_road FOREIGN KEY(fk5\_RoadId) REFERENCES dim\_road(RoadId) ON DELETE RESTRICT ON UPDATE RESTRICT; | ALTER TABLE fact\_Transportation ADD CONSTRAINT fkFactTo\_road  FOREIGN KEY(Skey\_road) REFERENCES dim\_road(Skey\_road); | Foreign key identifier is too long and name of the attribute is not correct. |
|  | | |

Table 19Modifying QSEE generated script to create relation between fact and road dimension

-- Alter table to add new constraints required to implement the "fact\_Transportation\_dim\_VehicleType" relationship

-- This constraint ensures that the foreign key of table "fact\_Transportation"

-- correctly references the primary key of table "dim\_VehicleType"

|  |  |  |
| --- | --- | --- |
| Generated Script | Modified Script | Reason for Modification |
| ALTER TABLE fact\_Transportation ADD CONSTRAINT fk6\_fact\_Transportation\_to\_dim\_VehicleType FOREIGN KEY(fk6\_VehicleTypeId) REFERENCES dim\_VehicleType(VehicleTypeId) ON DELETE RESTRICT ON UPDATE RESTRICT; | ALTER TABLE fact\_Transportation ADD CONSTRAINT fkfact\_to\_dim\_VehicleType  FOREIGN KEY(VehicleTypeId) REFERENCES dim\_VehicleType(VehicleTypeId); | Foreign key identifier is too long and name of the attribute is not correct. |
|  | | |

Table 20Modifying QSEE generated script to create relation between fact and vehicle type dimension

--------------------------------------------------------------

-- End of DDL file auto-generation

--------------------------------------------------------------

# Summary

QSEE tool is utilized to generate DDL scripts that can create required tables for warehouse. This paper reports log of the table creation and various issues faced during scripts execution. Scripts generated from QSEE are not completely compatible with oracle, hence some changes were necessary. For example, in road dimension Number (1) data type is assigned to define true or false instead of Boolean data type which is originally generated by QSEE. Finally, all the necessary tables were created and foundation of ware house is developed.

# Task 2 B

# Insertion of data from CSV file

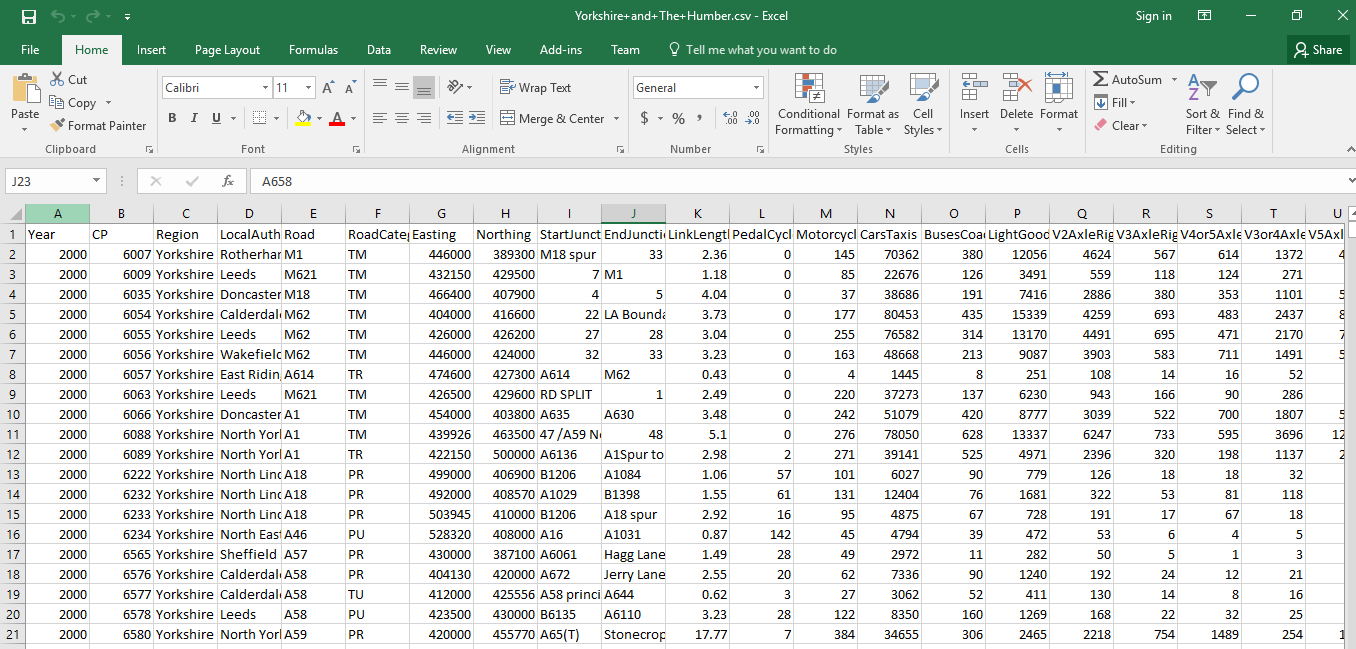


Figure 13 Screenshot of CSV file to be imported.

Leeds city has large amount of information stored in CSV file format as shown in figure 13. This information is required for development of data warehouse system. Hence, CSV file needs to be imported into a table that acts as a staging server. From this staging server, required data are extracted, transformed and loaded (ETL) into dimension tables. Data dictionary for required staging table is proposed below in table 21.

## OLTP data dictionary for Staging server

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Type | Null able? | Key? |
| ID | Number | False | Yes |
| Year | VARCHAR2(30) | True | No |
| CP | VARCHAR2(30) | True | No |
| Region | VARCHAR2(30) | True | No |
| LocalAuthority | VARCHAR2(30) | True | No |
| Road | VARCHAR2(30) | True | No |
| RoadCategory | VARCHAR2(30) | True | No |
| Easting | VARCHAR2(30) | True | No |
| Northing | VARCHAR2(30) | True | No |
| StartJunction | |  |  | | --- | --- | | VARCHAR2(255) |  | | True | No |
| EndJunction | |  |  | | --- | --- | | VARCHAR2(255) |  | | True | No |
| LinkLength\_miles | |  |  | | --- | --- | | VARCHAR2(30) |  | | True | No |
| PedalCycles | VARCHAR2(30) | True | No |
| MotorCycles | VARCHAR2(30) | True | No |
| CarsTaxis | VARCHAR2(30) | True | No |
| BusesCoaches | VARCHAR2(30) | True | No |
| LightGoodsVehicles | VARCHAR2(30) | True | No |
| V2AxleRigidHGV | VARCHAR2(30) | True | No |
| V3AxleRigidHGV | VARCHAR2(30) | True | No |
| V4or5AxleRigidHGV | VARCHAR2(30) | True | No |
| V3or4AxleArticHGV | VARCHAR2(30) | True | No |
| V5AxleArticHGV | VARCHAR2(30) | True | No |
| V6orMoreAxleArticHGV | VARCHAR2(30) | True | No |
| AllHGVs | VARCHAR2(30) | True | No |
| AllMotorVehicles | VARCHAR2(30) | True | No |

Table 21 Data dictionary for Stage table

# CSV File

# Importing File using APEX

## Creating New user

1. Create new user with DBA role for the warehouse project. [Figure 14]

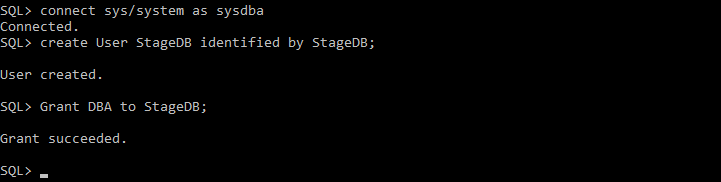


Figure Creating User StageDB

## Login into Apex

1. Start APEX application (Oracle Database XE) and login using created DBA user. [Figure 15]

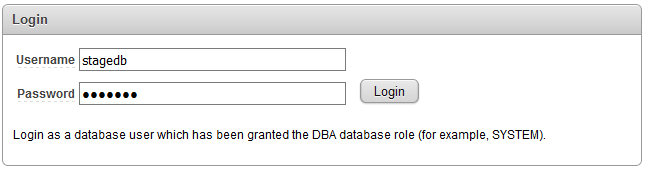


Figure Login into APEX

## Creating Workspace and user

1. In Oracle Database XE, browse to Application Express tab and create workspace [Figure 16].

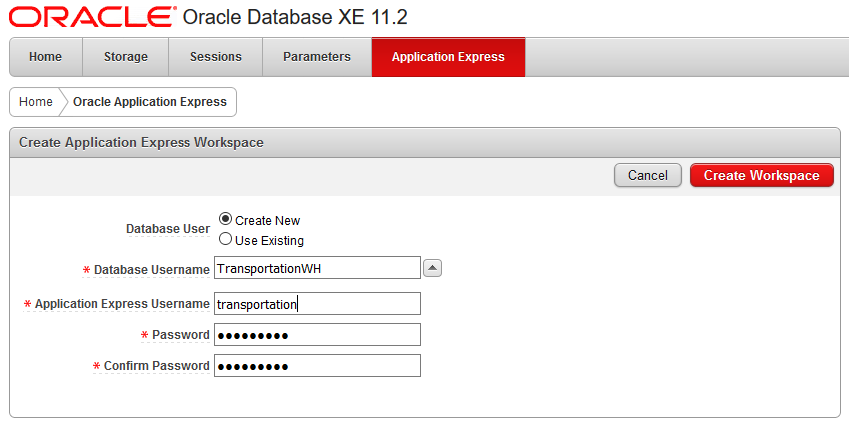


Figure Workspace Creation

## Login into Workspace

1. Login into workspace using created username [Figure 17]

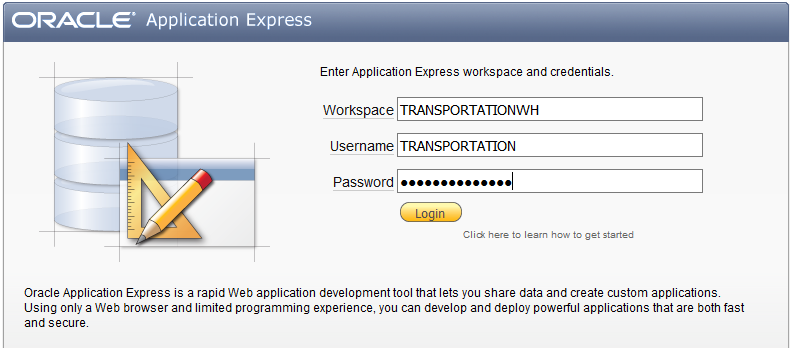


Figure Login into workspace

## Importing CSV data

1. Click on SQL workshop table 🡪 Utilities 🡪 [Figure 18]

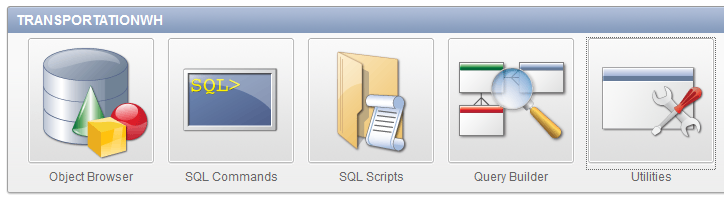


Figure Utilities Menu

1. Click on Data workshop [Figure 19]

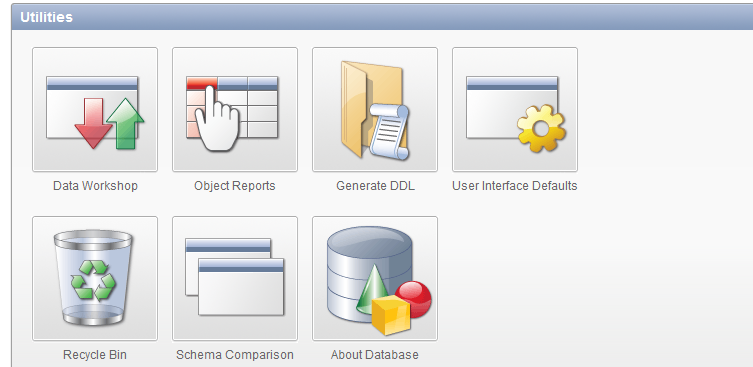


Figure Data Workshop Menu

1. Select spreadsheet Data from the Data Load section [Figure 20]

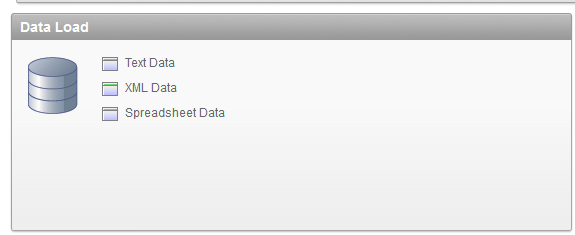


Figure Data Load Menu

1. On the Load data section🡪 Choose New table and Upload File🡪 Click Next [Figure 21]

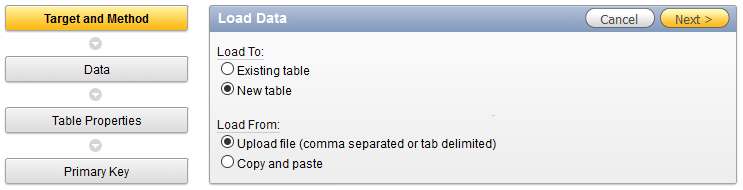


Figure Load Data Menu

1. Browse the CSV file and click Next [Figure 22]

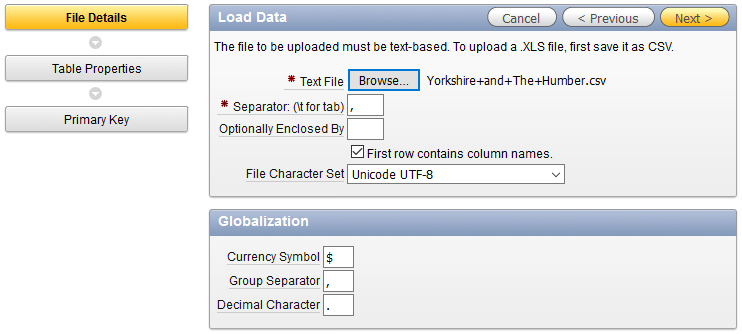


Figure File Browse Menu

1. Make any required changes to staging table🡪 Click Next [Figure 23]

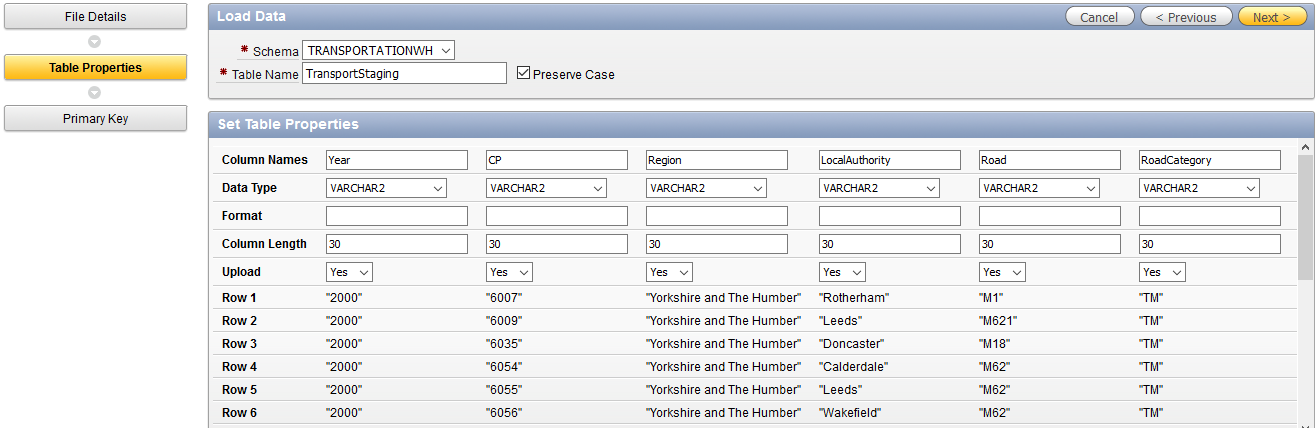


Figure Load Data staging table Matching Menu

1. Choose Primary key from 🡪 Create New Column🡪 Type ID 🡪 Give constrain name
2. Choose Generated from new Sequence 🡪 Give sequence Name 🡪 Click Load [Figure 24]

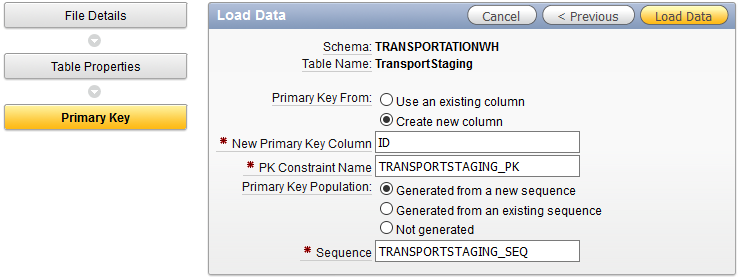


Figure Load Data Primary Key Setting Menu

1. Summary of File loading is display in summary table [Figure 25].

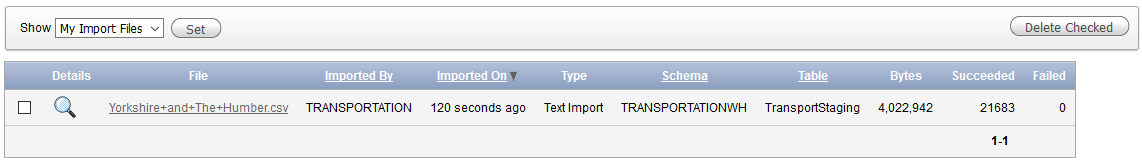


Figure Summary of File Loading

1. Confirm data loading by viewing object browser [Figure 26]
2. Run select \* from TransportStaging query to view loaded information [Figure 27]

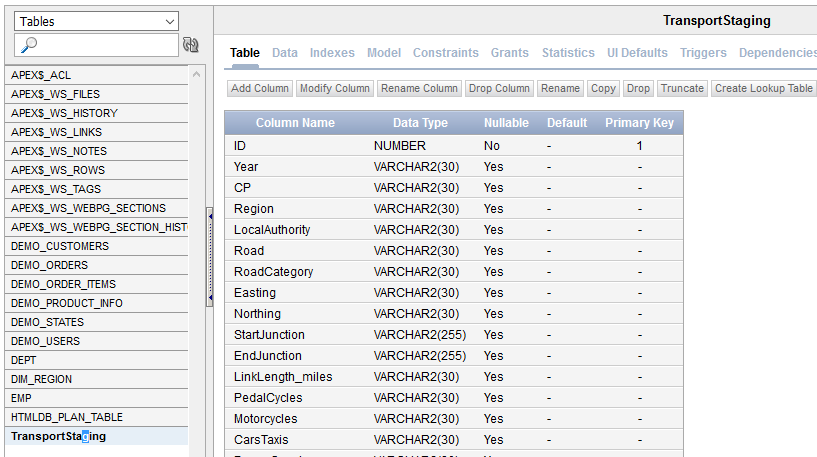


Figure TrasportStaging table in object browser

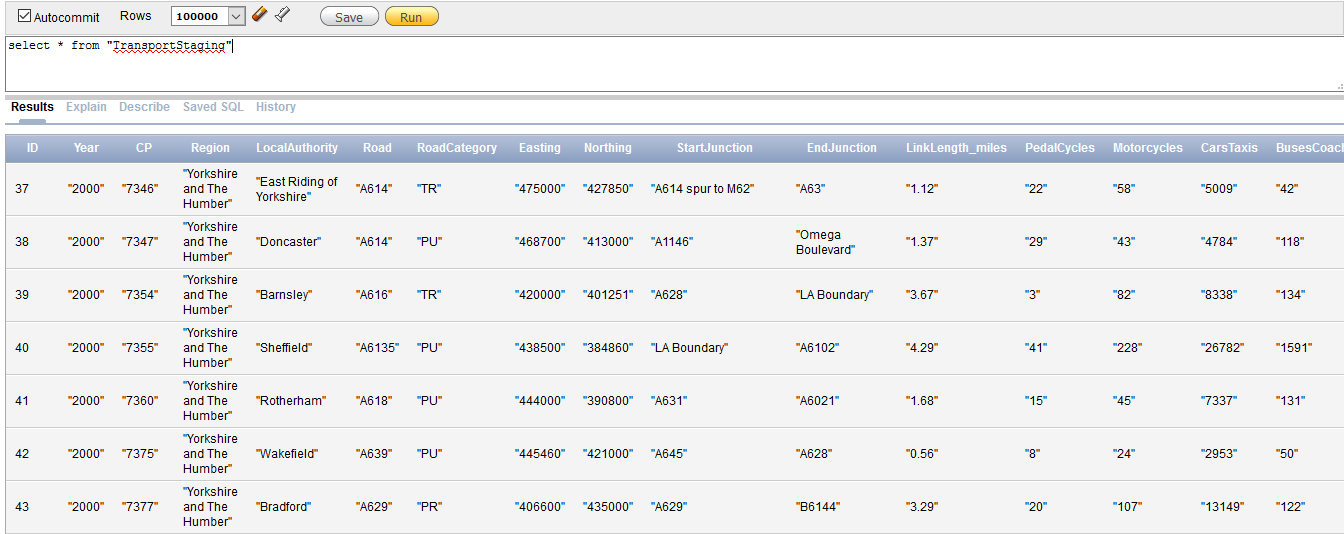


Figure Running Select query on TransportStaging.

# Summary

This paper finally documented the CSV file importing phase. Oracle offers feature that allows to import data from excel/CSV files. Each procedure followed to import data are documented in this paper. Now as all the required information is imported into stage table, ETL scripts can be developed to load information into actual warehouse.

# Bibliography

* Damson, C. 2006. *Mastering Data Warehouse Aggregates: Solutions for Star Schema Performance*. Wiley.
* Inmon, W. 2005. *Building the data warehouse*. Indianapolis, IN: Wiley Pub.
* Kimball, R. and Ross, M. 2002. *The data warehouse toolkit*. New York: Wiley.
* Wrembel, R. and Koncilia, C. 2007. *Data warehouses and OLAP*. Hershey: IRM Press.