



University of Technology, Sydney

Faculty of Engineering and Information Technology

Subject Database Programming (31253)

Assignment 1 Autumn 2012

Due date - 12th April 2012(Hand in during lecture)

INSTRUCTIONS

1. This is an individual assignment and is worth 20% of your final grade. The Assignment will require a commitment of about 16 hours and addresses objectives 1, 2 and 3
2. For ease of marking, I would like you to document your approach to the solution of the problem. Please keep your description short, relevant and to the point. It is strongly recommended that you utilize diagrams to convey your ideas. Your description will help me understand your code better. Please indicate any features that you are particularly proud of or that you want to be noted during the marking process.
3. You are expected to submit a professional presentation, prepared using a suitable Word Processor. A hand written submission is not acceptable and will not be marked.
4. I will be looking at your code and data in the Database so there is no need to include the code in your submission. Also your object creation scripts do not need to be included. So that I can access your schema please include your ORACLE userid on the front page of your submission. **DO NOT INCLUDE YOUR PASSWORD**
5. There may be errors and ambiguities in the assignment specification. If so, corrections/clarifications will be posted to the subject web sites. You are expected to check and incorporate these changes into your submission.
6. Please take note of the due date, and work to that date, as extensions will be granted only under extenuating circumstances.
7. Adequate feedback on the marking process will be provided for your submission
8. I expect to return the marked Assignments within 14 days after the submission date.

Because I will be running my marking script against your program it will be greatly appreciated if you would standardize the name of your entry procedure. The procedure name should be **clean_helens_data**



Overview

A friend of mine, whose name is Helen Fisher, works in the Business unit of a very well known charity organization. The charity organization maintains a list of individuals and organizations that it canvases during its regular charity drives. This clients list is currently being maintained in an Excel spreadsheet.

The spreadsheet has been built up by many different people over many years, so the consistency of the data is not good. Presently there are about 10,000 clients in the list and the list is growing.

My friend Helen is finding that the spreadsheet and the format of the data are limiting her ability to efficiently process the various requirements placed on her by her management. In particular, Helen would like to sort the clients into street order within suburbs, unfortunately because the address is not *atomic*; this simple task is proving to be very difficult for her.

Also, the address details do not contain a postcode which makes it difficult to comply with the post office requirement of, a postcode on each address, when she has to organize a mail drive.

In an attempt to improve her ability to work with the client list, Helen would like to '*migrate*' the data into an Oracle database which has been made available to her on the organizations computer system. Helen has no idea of the technicalities of Oracle, so I volunteered the 31253 class to help her out.

During the migration from Excel to Oracle, Helen would like to 'clean' the data and to add a postcode attribute for each one of the records.

('Clean' in this context means that the address field is to be separated into atomic values.)

The rules for 'cleaning' are specified at the rear of this document.

Helen has provided a postcode file which she has obtained from Australia Post web site. She wants this Post Office data to be used when determining the postcode for each client suburb. In order to simplify your task, I have loaded the postcode file into the database table that I called DBP_POSTCODES.

Helens original data file contains about 10000 records. This number is too large to allow each member of the class to work with efficiently. I have therefore extracted a small subset of records that are a good sample of the real data, and loaded them into a table called HELENS_DATA. You are to use the data in that table during your development. The spreadsheet that Helen gave me will be posted on UTSONline so that you have the opportunity to view the data in its entirety

Note:

You have only **SELECT** privileges to the two tables, **DBP_POSTCODE** and **HELENS_DATA**



Your task

You are to write a PL*Sql program that will *clean* Helens data.

For each record, you are to

- Split the address1 field into atomic values.
- Populate the postcode attribute for each suburb.

Helens data is to be *migrated* into a table called DBP_CLIENTS. Each correctly processed record is to be inserted into this table.

You are also required to create an exceptions table called DBP_CLIENT_EXCEPTIONS. This table is to have the same structure as the DBP_CLIENTS table but with an additional column called **COMMENTS**.

Any record that you cannot process is to be inserted into the exceptions table. The **COMMENTS** column is to be populated with the reason for the exception.

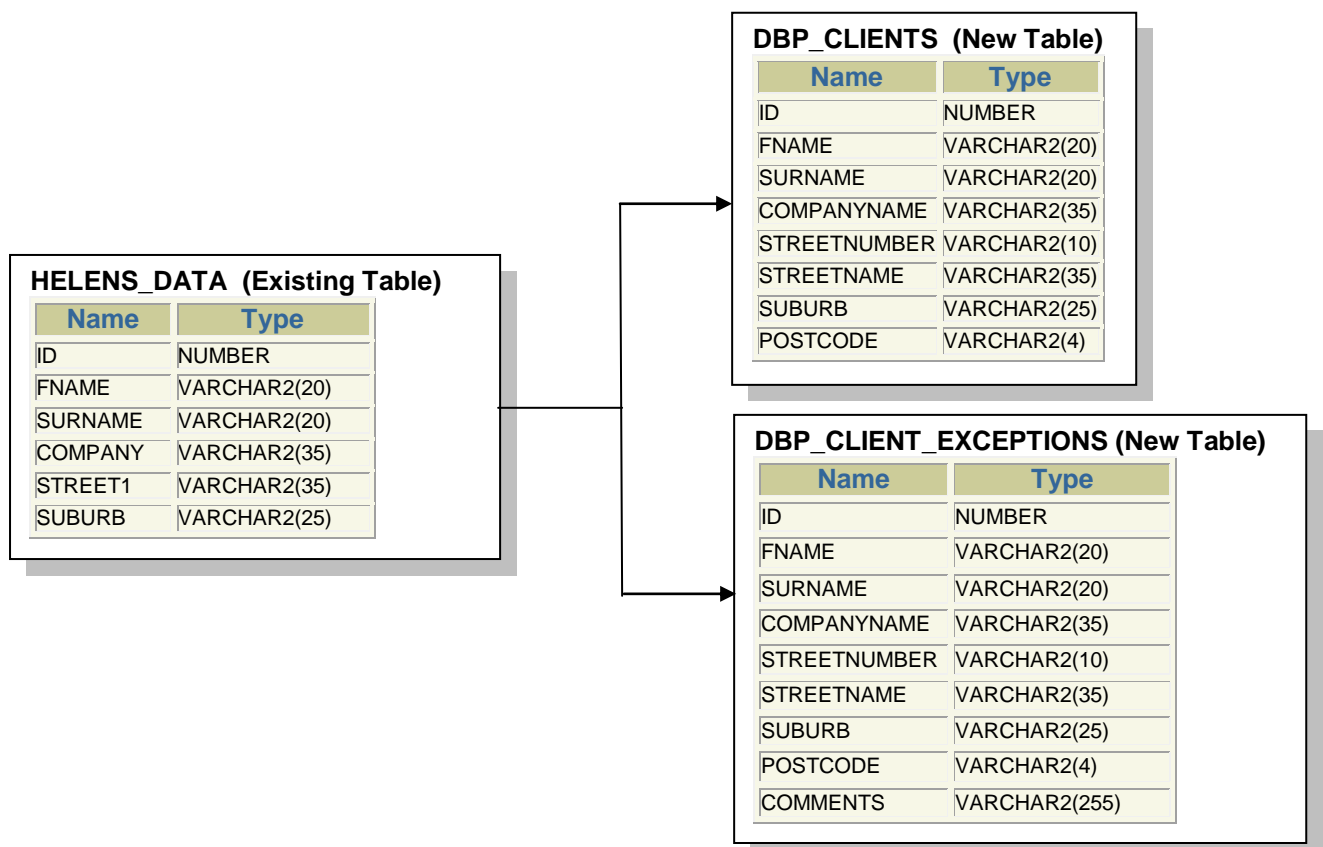
Likewise, if you are unable to determine the postcode for a record, then that record will also be placed into the exception table.

At a latter date, the records in the exceptions table will be manually examined and corrected by a consultant that will be hired for the task. The aim is to minimize the exceptions, thereby reducing the cost of the consultant.

The tables DBP_CLIENTS and DBP_CLIENT_EXCEPTIONS are to be mutually exclusive

i.e. A record must exist in either one table or the other, but not in both.

The following diagram describes the structure of the tables to be used.





Deliverables

In your submission you are expected to deliver

- Instructions to the user on how to effectively run your program
- Description of the Task
- Design of the Solution (Technical Specification Document)
- Code, both a hard copy of the code and a soft copy stored in the Database in your schema. If the code comprises of a number of Procedures and Functions, then I will expect a document that describes the structure and relationship of the code, preferably in a diagram format.

Note: I will do the bulk of the marking on line, by running your code in the Database. To make the task seamless, you should name your entry module **clean_helens_data**.

Marking Scheme

Total weight of Assignment 20%

Compiles Correctly	5
Design of the Problem Solution	15
Code Modularisation	10
Appropriate commenting and documentation	5
Variable Naming (self documenting)	10
Appropriate exception handling	10
Is the code tight and concise	20
Is the code easily maintainable	5
Is it easy to run	5
Run against the full data, how did the code perform	10
Professional Presentation	5



Conversion Rules

Following is a table listing the common format of the existing data, together with the structure of the ideal conversion.

Records that can not be converted successfully are to be placed into the exceptions table. Wherever possible, the **COMMENTS** column is to be populated with the reason for the exception.

Original Data	Conversion
STREET1	STREETNUMBER STREETNAME
1 Ada Street	1 Ada Street
1 A Kalinda Street	1A Kalinda Street
1/13 Burrawang Street	1/13 Burrawang Street
1/21-23 Chelmsford Road	1/21-23 Chelmsford Road
10/1a Frederick Street	10/1a Frederick Street
161? Lang Street	161 Lang Street
368 Cnr Barker & Anzac Pde	Barker & Anzac Pde
5/70-74/ Wilson Street	5/70-74 Wilson Street
Shop 2 4-14 Claudia Road	Shop 2 4-14 Claudia Road
PO Box 558	PO Box 558
Lot 4 Mungerie Road	Lot 4 Mungerie Road
Level 4/65 Albert Avenue	Level 4/65 Albert Avenue
Cnr Warf Rd & Hollywood Rd	Warf Rd & Hollywood Rd
Unit 85 51 Little Willandra Rd	Unit 85 51 Little Willandra Rd
Suite 2 Ground Floor	Rejected
Street Avenue Road Lane etc	Rejected