Exploring a Relational Database using Oracle SQL*Plus

Activity 1 - Review

YOUR LOGIN DETAILS

Before you start you should have the following:
Write them down so you don't forget them !

Student Login: ____________________
CourseWork Login: ____________________
Password: ____________________
Host String: STUD____________________

HOW TO USE THIS BOOKLET

READ THIS !

In order to get the most from these exercises you should attempt to:

- ✓ Read everything; don't just skip to the questions!
- ✓ Complete all parts in the order presented
- ✓ Think about the nature of the application itself
- ✓ Keep referring to
  - ✓ The ER diagram
  - ✓ The table definitions

What to do in this book:

- ✓ Read the explanations and questions
- ✓ Fill in the results or answers

- ✓ At the SQL> prompt enter a single SQL or SQL*Plus command
- ✓ You should use an editor (eg Notepad) to type and edit your command
- ✓ Save the command with the extension .SQL or txt (ie example3.SQL)
  - ✓ Do not use a word processor as the formatting interferes with the commands

- ✓ To get SQL*Plus to run the command file
  ie if it was saved in subdirectory temp on drive C called example3, *type*:

  @f:\databases\example3

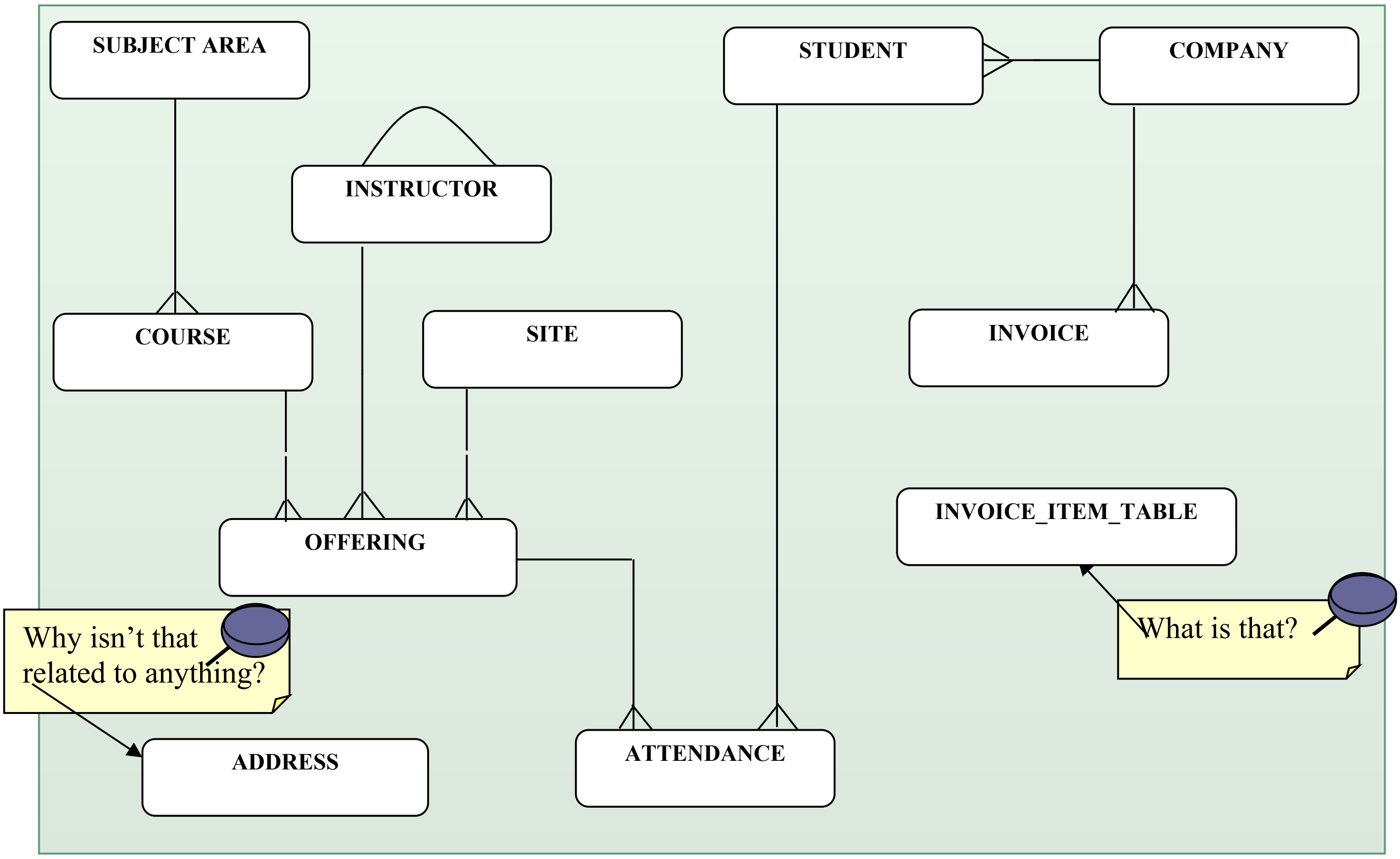
THE DATABASE APPLICATION USED IN THIS GUIDE

- The Entity relationship model is of a contract training department
- It is to be stored in an Oracle *Object* Relational Database
    - There are ten normalised tables which make up the database
    - These correspond to the entities shown in the Entity Relationship Diagram below:

Entity Relationship Diagram of the application

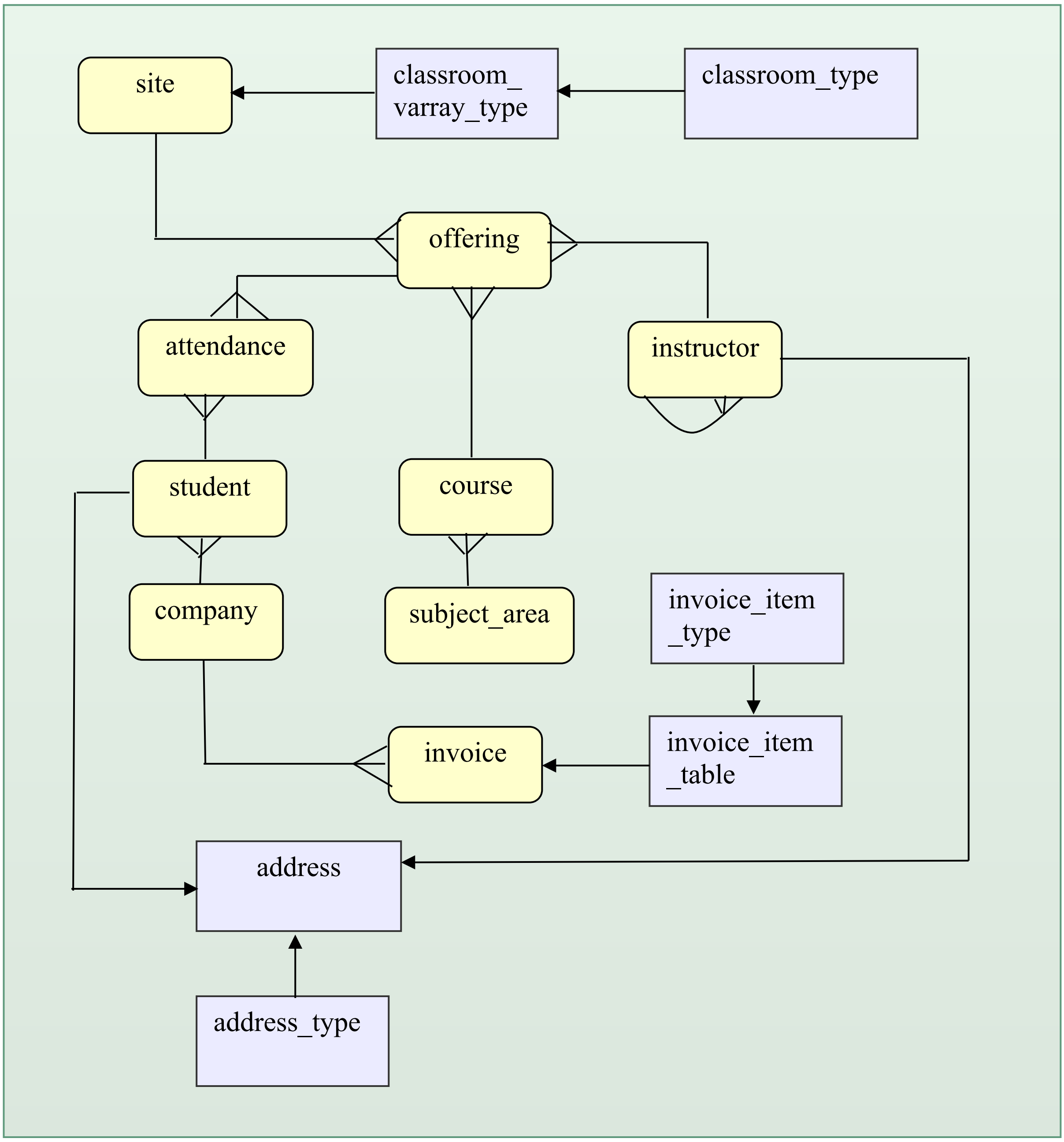
The naming conventions used in the database are:

- Table names are plural versions of entity names
- Use is made of underscore;
    - ie the entity subject area corresponds to a table called SUBJECT_AREAS
- Primary and foreign keys are all called something_id, for example course_id
- Column names are often comprised of several words separated by underscores ie: trainer_name, course_price



- We are working with an Object Relational Database (ORDB) so there will be some key differences. Keep an eye out for them and highlight them if you need to.

SCHEMA ENTITY RELATIONSHIP DIAGRAM



- Crows foot links denote entities (yellow),
- Arrow head links denote objects (lilac)
-

Database Skeleton Table

- The tables will hold data representing the following information:

COURSE DATABASE

TABLES	ATTRIBUTE	KEY	DATATYPE	CONSTRAINTS \ DEFAULTS
ADDRESSES	address_type:			
	street	New!	VARCHAR2(25)	
	city		VARCHAR2(25)	
	country		VARCHAR2(20)	
SUBJECT_AREAS	subject_id	*PK*	NUMBER(6)	
	subject_name		VARCHAR2(30)	
COURSES	course_id	*PK*	NUMBER(6)	
	title		VARCHAR2(30)	
	subject_id	*FK*	NUMBER(6)	
	duration		NUMBER(6)	
	cost		NUMBER(10,2)	
	brochure		Binary file LOB	New!
INSTRUCTORS	instructor_id	*PK*	NUMBER(6)	
	instructor_name		VARCHAR2(15)	NOT NULL
	address		REF OF address_type	
	mentor_id		NUMBER(3)	
	salary		NUMBER (10,2)	
	commission		NUMBER (10,2)	
	date_hired		DATE	
	profile		CLOB New!	
OFFERINGS	offering_id	*PK*	NUMBER(6)	
	course_id	*FK*	NUMBER(6)	NOT NULL
	site_id	*FK*	NUMBER(6)	NOT NULL
	instructor_id	*FK*	NUMBER(6)	
	start_date		DATE	
	max_no_students		NUMBER(6)	
ATTENDANCE	offering_id	*PK FK*	NUMBER(6)	
	student_id	*PK FK*	NUMBER(6)	NOT NULL
	Start_date	*PK*	DATE	
	evaluation		NUMBER(6,2)	
	amount_paid		NUMBER(10,2)	
SITES	site_id	*PK*	NUMBER(6)	
	address		address_type	
	classroom		classroom_varray_type	New!
	location		VARCHAR2(12)	

TABLES	ATTRIBUTE	KEY	DATATYPE	CONSTRAINTS \ DEFAULTS
COMPANIES	company_id	*PK*	NUMBER(6)	
	company_name		VARCHAR2(15)	
	city		VARCHAR2(12)	
	industry		VARCHAR2(15)	
	no_employees		NUMBER(5)	
	revenue		NUMBER(10,2)	
	training_budget		NUMBER(10,2)	
STUDENTS	student_id	*PK*	NUMBER(6)	
	student_firstname		VARCHAR2(20)	
	student_surname		VARCHAR2(20)	
	address		ref of address_type	New!
	phone_number		VARCHAR2(15)	
	fax_ number		NUMBER(10)	
	gender		CHAR	
	date_of_birth		DATE	
	description		VARCHAR2(250)	
	preferred_language		VARCHAR2(2)	
	company_id	*FK*	NUMBER(6)	
INVOICE_ITEM_ TABLE_TYPE	invoice_item_type:			
	student_surname		VARCHAR2(15)	
	course_title		VARCHAR2(30)	
	start_date		DATE	
	end_date		DATE	
	amount		NUMBER(10,2)	
INVOICES	invoice_number	*PK*	NUMBER(6)	
	company_id	*FK*	NUMBER(6)	
	billing_date		DATE	
	due_date		DATE	
	invoice_item		invoice_item_table_type	New!

There are some new tables and datatypes in an Object Relation Database. Keep an eye on them there will be more on these as we go along.

New!

BEFORE STARTING

- ✓ Create a folder on your flash or personal area with the module name eg CSY2038 **no spaces**
- ➢ Files saved in this folder will have the path
  e:\CSY2038
- ➢ For example a script file called CreateTest.txt will have the path
  e:\CSY2038\CreateTest.txt

NOTE: files, folders and paths names can *not* have spaces in them!

LOGGING ON TO ORACLE

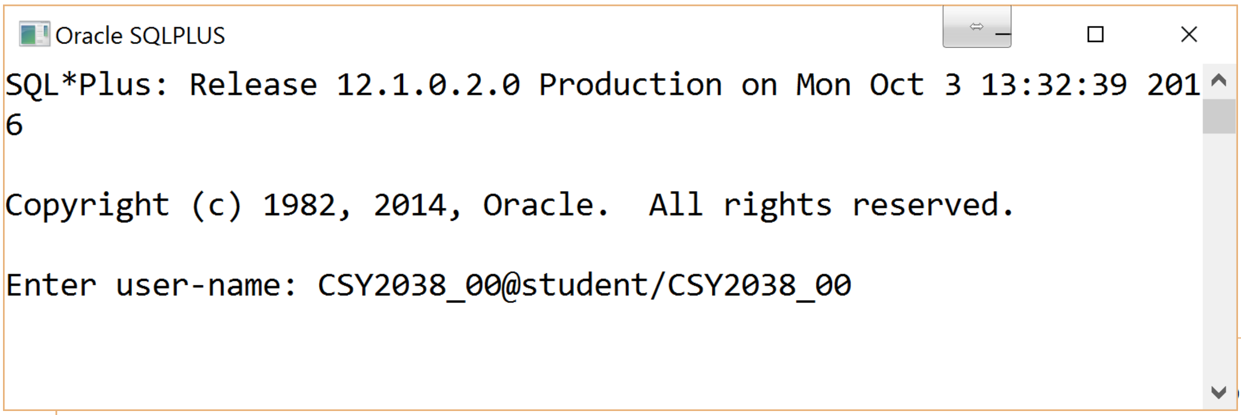
- To log on to the Oracle server
- You will need to open SQL* Plus

- On your client computer, select Oracle SQL*Plus.

The login window will appear. You need to connect your client-based SQL*Plus program to the Oracle server by **completing** your coonection details

You noted them down in the front of this book!

- At the prompt enter your username@student/password

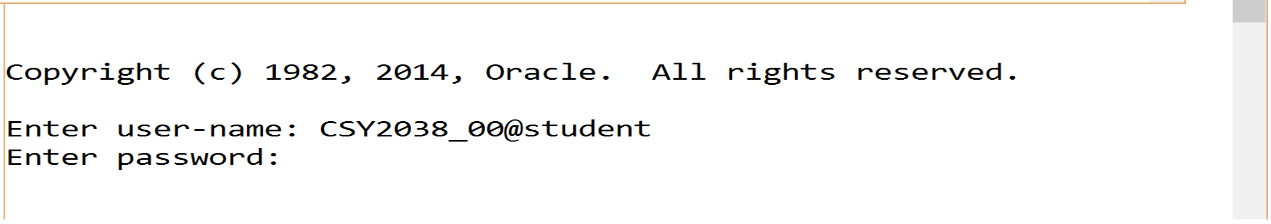


Oracle SQLPLUS

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Oct 3 13:32:39 201
6

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Enter user-name: CSY2038_00@student/CSY2038_00
```



```
16

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Enter user-name: CSY2038_00@student
Enter password:
```

- At the SQL> prompt enter a command to get SQL*Plus to interpret the SQL program

- Type **EXIT** on the command line if you want to close SQL*Plus
- Or click the cross in the corner

CHANGING YOUR PASSWORDS

- The **ALTER** command is used in the data definition language to change table or user structure
- To change passwords the following structure is used

  ALTER USER *your_user_id* **IDENTIFIED BY** *your_new_password***;**

- To enter ***a number*** as a password it should be in double quotes ie "200"
  This changes it to a text value!
- Passwords are **case sensitive**

✓ At the SQL prompt alter ***both*** your *oracle* passwords to your student number

✓ You need to log in to each account separately. You only have privileges to change your own password

✓ Log off using the **EXIT** command

✓ Test logging in to both with your new passwords

✓ If everything works log out

NOTE: Passwords **cannot** start with a number. Use double quotes or letters

- If you try to connect with an incorrect username or password you will get the following error message:

  ERROR
  ORA-01017: invalid username/password: logon denied

ERROR MESSAGES: These can be helpful, always read them and try to correct the problem before hand waving!

CREATING YOUR COPY OF THE ORACLE RELATIONAL DATABASE

➢ To create your own version of the relational database on the Oracle server
➢ You will need to execute a copy of the script files **dropCourse.txt, createCourse.txt**, **alterCourse.txt and insertCourse.txt**

  ➢ These contain SQL DDL **drop**, **create, alter and insert** commands

✓ **Download** all 4 files from the blackboard, into your CSY2038 folder
✓ Rename each file to include your login number ie **createCourse_00**
✓ Open the script files in notepad++ and look at them
✓ Consider what each script file does, do they match the ERM?
✓ Identify the new tables and datatypes?

HELP: There is an additional crib sheet file on the web. You should keep this and add to it during the course.

Q *What are the differences in this Object Relational Database from a normal Relation Database?*

✓ Log onto Oracle using your two digit login

➢ To create some tables in our relational database on the Oracle server using the create table commands in the **createCourse.txt** script file, we could just copy and paste the files into SQL *but* the buffer may not be large enough to hold all the data. Only part of the data would be copied and it would therefore be wrong. Instead we can point SQL to our file and it will read from it!

➢ The @ sign is used to point SQL in the right direction, followed by the file path and name ie: if your script file is on drive C: in the temp directory, the command would be:

@C:\temp\CSY2038\createCourse_00.txt

Q *What would the command be if the file is on your flash in your module folder?*

✓ At the SQL> prompt enter a command to get SQL*Plus to run your script files

➢ Instructions are passed to the Oracle RDBMS to create the various database tables
➢ Wait until the SQL program has created all the tables
➢ Oracle RDBMS stores the names of your tables in one of its tables called **TAB** *(tablespace)*

➢ **TAB** is a table already on the database which will store all your tables

✓ When the run command is correct, copy it to the top of the script file and comment it out eg
-- @C:\temp\CSY2038\createCourse_00.txt

✓ At the SQL> prompt type **SELECT * FROM TAB;** on a single line

Q *What is the result?*

✓ Now type each word on a separate line ie:

NOTE: SQL does not like spaces between lines!

```
SQL> SELECT
   2 *
   3 FROM
   4 TAB
   5 ;
```

Q *What result does this give you? What does it tell you about the syntax?*

Q *Do you have all the tables in the skeleton list provide on pages 4 and 5?*

➢ The **DESCRIBE** command (DESC for short) is used to describe the table structures

➢ The **DESC** command does *not* show the data stored in the tables

✓ Use the **DESC** command to view the tables stored in your database

Q *What attributes does the* ***courses*** *table have?*

CREATING MORE TABLES

- ➢ To add additional tables to the database use the CREATE TABLE command and save it in an editor
- ➢ To **create tables** the following structure is used

```
CREATE TABLE table_name (
    column_name1  datatype,
    column_name2  datatype);
```

NOTE: We have done all this before

- ➢ The command may be written directly into SQL over several lines
- ➢ SQL*Plus outputs a line number (2) on subsequent command lines
- ➢ SQL*Plus looks for a semicolon (;) which terminates a SQL command
- ➢ When entering commands in an editor the prompt and line numbers are not used

- ✓ In the *createCourse_yourlogin.txt* file add the command to create the following table
- ✓ Give the table the name 'companies'

ColumnName	Datatype	Constraints
company_id	number(6)	
company_name	varchar2(15)	
city	varchar2(12)	
industry	varchar2(15)	
no_employees	number(5)	
revenue	number(10,2)	
training_budget	number(10,2)	

NOTE!: Don't forget that the tables and drops should go in the right order

- ✓ Save the script file
- ✓ At the SQL> prompt use the @ command to run the script file
- ➢ What are all the error messages about?

HINT!:
DROP TABLE
table_name ;

- ✓ At the SQL> prompt write a command to view all your tables
- ✓ If the companies table is there, write a command to describe your new table
- ➢ Does this look right?
- ✓ In the *dropCourse_yourlogin.txt* file, in the **correct place** add the command to drop companies

CREATING PKS AND FKS

- ✓ In the *alterCourse_yourlogin.txt* file, write the command to create a PK for the companies table
- ✓ Use the ERD to find out which tables should have company_id as a FK
- ✓ In the *alterCourse_yourlogin.txt* file, write the command(s) to create the appropriate FKs
- ✓ Download and run the ***CompaniesInserts.txt***, not copy and paste
- ✓ In the *insertCourse_yourlogin.txt* file add the company insert commands
- ➢ *Think about **where they should go** in the script file, include comments*

- ✓ Tidy up your script files

  Include a header comment with your name, number, module, activity and date

  Read through the script file and make clear notes throughout to help you understand

ACTIVITY TASKS

✓ At the SQL prompt> enter the command to run your ***dropCourse_00, createCourse_00, alterCourse_00, insertCourse_00*** files

✓ Practice the following commands in the TRAINING database

NOTE: You may need to refer to your lecture notes and crib sheet to help you

1) Enter the command **SELECT * FROM TAB**;
2) Describe the sites and students tables

Q *What does this tell you? Any unusual data types?*

3) Describe the **user_tables and user_types**

Q *What does this tell you?*

4) In a new script file called *your_login_prac1.txt* write the command to:
Create a listing of student names gender and telephone number
Concatenate first and last names

NOTE: Two pipes || are used to concatenated columns

HINT: **SELECT** *column_name1* ||' ,'|| *column_name2, column_name3*, *column_name4*

Q *Why is the heading to Gender shortened to G? (look at skeleton tables for a clue)*

5) Format the output showing
"Last Name", and "First Name" on two lines
Gender as "F/M" on one line
Change the format of gender to three characters so that the entire heading is shown

NOTE: One pipe '|' Puts column names on two lines

HINT:

```
COLUMN column_name1 HEADING 'Title1 |Title2'
COLUMN column_name2 HEADING 'Title1 |Title2'
COLUMN column_name3 HEADING 'Title1 | Title2'
COLUMN column_name1 FORMAT A10;
```

6) Change the SQL to retrieve only the female students

HINT: **WHERE** column_name1 = criteria

NOTE: Formatting only last f the current session. It is overridden by formatting in select statements. ie the ||','||

7) Change the SQL to list the answers in alphabetical order of the surnames

HINT: **ORDER BY** column_name1

✓ Save your script file

✓ Log out of SQL ☺

NOTES

EXTRAS - USING THE BUFFER

- ➢ The buffer is a work area that holds/stores the latest SQL statement
- ➢ SQL statements can be written in a single line or free flow as in an editor. A semicolon indicates to the compiler that the SQL statement is at an end and should be complied
- ➢ To view the content of the buffer the **EDIT** command is used
- ➢ The built in editor is used to *edit* the current statement after it has been created to correct errors and change the SQL

✓ At the SQL> prompt enter a command to call the buffer
✓ Return to SQL>prompt

NOTE: The buffer takes up a large amount of system resources so generally try not to use it

- ➢ The buffer only hold the last set of instructions
- ➢ Changes can be made to the content of the buffer and it can be re run with the / sign

✓ Use the / to run the buffer
You may get some errors?

- ➢ There are several standard commands that can be used with the Oracle 8I buffer

COMMAND	DESCRIPTION
L	Lists the current content of the buffer
Ln	Displays text of line number ‘n’ and makes it the current line
I	Inserts a new line after the current line
A	Appends text to the current line*
DEL	Deletes the current line
C/<old>/<new>	Replaces first occurrence of the <old> text with <new> text on the current line
/	Resends content of buffer
EDIT	Opens notepad and displays the buffer contents

NOTE: All SQL commands are operated at the local machine and do not update the server until the COMMIT; command is executed

* *type two spaces after the **A** to leave one space in the SQL command eg:*
A ,student_address - *two spaces after A will generate one space before student_address*

USING AN EDITOR

- Commands written in SQL are not stored, only the subsequent changes are stored in Oracle. Therefore typing commands into a file and saving them means that we have a backup of our code if anything goes wrong and a useful reference.

NOTE: Always store you commands in a script file!

- To view the content of an editor the **EDIT** *path* command is used
- We need to point SQL at the correct file
  ie: if a script file is on drive C: in the temp directory, the command would be

EDIT C:\temp\wk1.txt

NOTE: Files can have the extension ***.sql*** or ***.txt***

Q *What would the command be if the file is on your floppy in your module folder?*

- ✓ At the SQL> prompt enter the command to call the editor
- ✓ Call the script file *your_login_prac1.txt ie CSY2038_00prac1.txt and* save in your folder
- ✓ In the editor replace the code with a command to view your tables

- ➢ Changes in the editor must be saved before it is run
- ✓ Return to the SQL> prompt
- ✓ At the SQL> prompt use the @ command to run the new script file
- ✓ Copy the run command into the top of the script file and comment it out ie
  -- @drive:\folder_1\folder_2\file_name.txt

REMEMBER!:
Oracle will not accept spaces in file, folder or path names