SIT103 Database and Information Retrieval

Practical 1: Accessing the Oracle Database

Objectives:
- To learn how to access your Deakin Oracle database
- To get familiar with basic SQL and SQL*Plus commands

1. Introduction

The Relational Database Management System (RDBMS) is responsible for accurately storing data and efficiently retrieving that data in response to user queries. The Oracle RDBMS is an industry leading database system.

The Oracle RDBMS supplies interface tools to access data stored in an Oracle database. One of these tools is known as SQL*Plus, which is a command line interface. Our practicals will use the SQL*Plus tool to learn the database language SQL and salient features of the Oracle DBMS.

If you are an enrolled Deakin University student, you have already been allocated a database which is managed by the Oracle DBMS at Deakin University. In this practical, we introduce how to access and use your Oracle database at Deakin.

2. Access the Oracle Database

To access and use the Oracle database, you need to

- Connect to a Deakin server
- Access the student database (SSID) using your Deakin username and password

2.1. Connect to a Deakin Server

Tera Term, which is a SSH (Secure Shell) client software, allows you to securely log in and use the command interface on a remote Unix-based host.

> **NOTE:**
>
> For *Mac users*, Mac OS X includes SSH built-in, so you do NOT need to install Tera Term or PuTTY. Open Terminal (found in your Utilities folder) and type "*ssh yourDeakinUsername@interactive.deakin.edu.au*" for connection.
>
> Tera Term is installed in all campus computers.
>
> Alternatively another SSH client software, **PuTTY**, can also be used to connect your computer to a Deakin server. The process is similar to that of using Tera Term which is described below.
>
> PuTTY is free to download from
> http://software.deakin.edu.au/2017/03/24/putty-ssh/

- If **Tera Term** is not installed in your own computer, you can either install **Tera Term** from Deakin Learning Toolkit or go to the following website

 http://download.cnet.com/Tera-Term/3000-2094_4-75766675.html

  to download and install it in your own computer.

- Launch **Tera Term** from the START → All Programs → Tera Term → Tera Term. A "New connection" screen will display as follow, prompting you to type in the host name.

 **Note:** You must ensure that "SSH" is selected for connecting to the Deakin host.

- Type in "**interactive.deakin.edu.au**" in the "Host:" area as above, and then click "OK" button. A new "SSH Authentication" window will display as follow, asking for your Deakin Username and Password. Type in your correct Deakin Username and Password in the "User name:" and "Passphrase:" areas respectively, and then click "OK".

- If your Deakin Username and Password are typed in correctly, you will see the following window with the Pdmenu displayed, which means you are now connected to a Deakin server, i.e., *interactive.deakin.edu.au*.
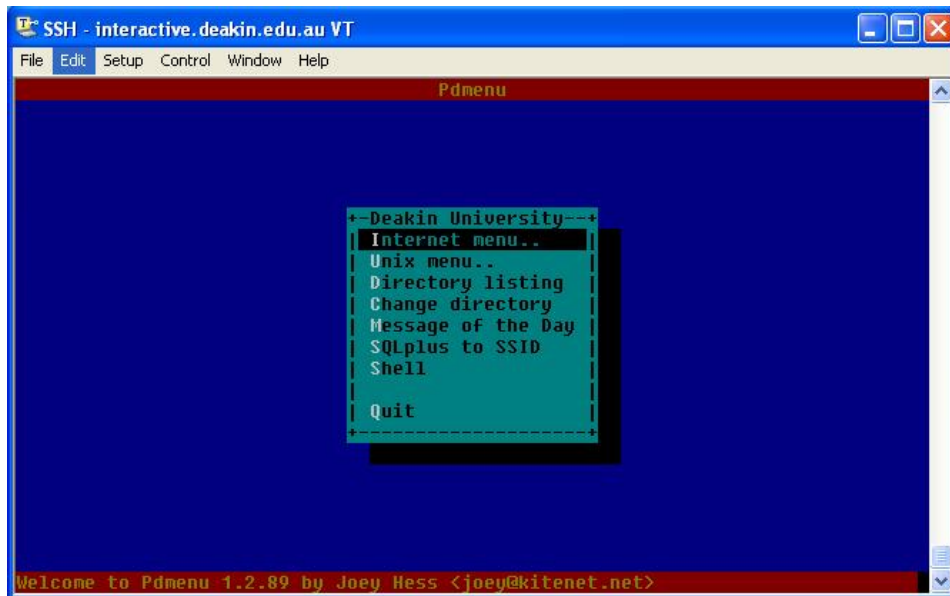
2.2. Access the Oracle database (SSID)

- Press up or down key on your keyboard to move the cursor to the option "SQLPlus to SSID" in the menu, then hit Enter key. You will see the following screen, prompting you to enter your Deakin Oracle database password.
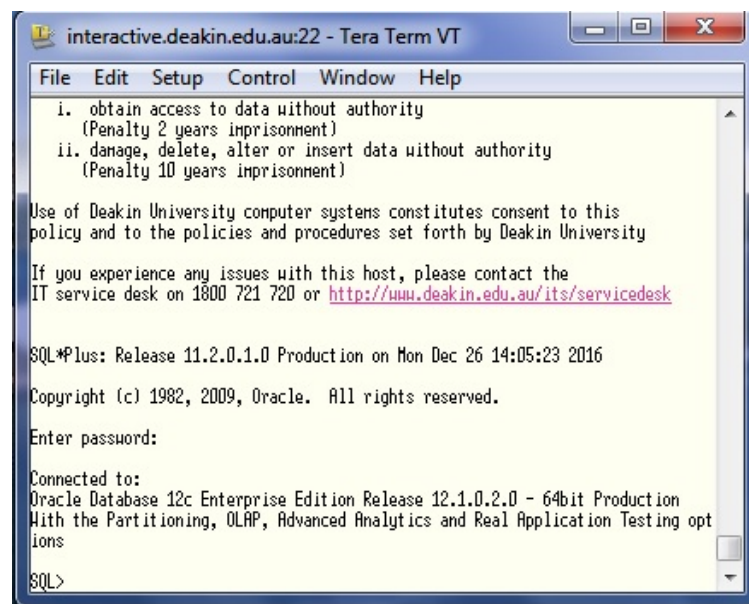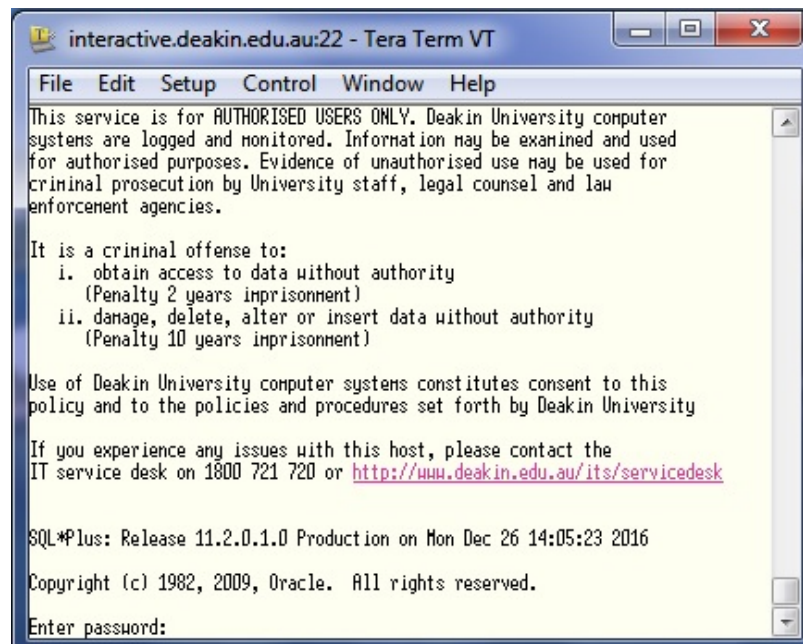
> **NOTE:**
>
>> Your **Oracle database password is your Deakin password**
>> (by default).
>>
>> When you are typing in your Deakin password within this window, **NO masking characters will be displayed**. Don't worry about it as this is the system setting. You only need to make sure the typing is correct.

- After the "Enter password:" prompt, type in your Deakin password correctly and then hit the "Enter" key, the following window will display with a SQL> prompt. This means that you are successfully connected to your own database in Deakin Oracle database management system (DBMS). Now you can type and run SQL or SQL*Plus commands at the SQL> prompt.

> **NOTE:**
>
>> If you typed in your Deakin password correctly but cannot see the above screen, the most possible reason is that your password contains some special characters, such as *. Go to Deakin website, change your Deakin password, and try again.
>>
>> If you want to quit Deakin Oracle DBMS, simply type in "quit" or "exit" under the "SQL>" prompt and hit Enter key.

2.3. Create tables within your database by SQL

A database may contain many tables. A table is used to store a kind of data, such as the student or course data. Tables in a database are created and populated with data by using SQL commands.

SQL commands are delimited by the semicolon (;) at the end, or by typing a forward slash (/) on a new line. Otherwise, the SQL command will not be executed by the system.

To create a table, you need to firstly define the table structure (relational scheme), then use the SQL command "CREATE TABLE" to create the table structure in the database. After the table is created, you can use the SQL command "INSERT INTO" to insert data into the table.

We are going to create four tables with sample data for exercises. These four tables are as follows:

Employee Table

empName	street	city
Jones	Main	Harrison
Smith	North	Rye
Hayes	Main	Harrison
Curry	North	Rye
Lindsay	Park	Pittsfield
Turner	Putnam	Stamford
Williams	Nassau	Princeton
Adams	Spring	Pittsfield

Company Table

CompName	City
Waltons	Harrison
Meyer	Rye
Waltons	Rye
Woolworths	Pittsfield
Tweeties	Harrison
Firebrand	Woodside

Works Table

EmpName	CompName	Salary
Jones	Tweeties	21000
Smith	Waltons	22000
Hayes	Woolworths	19000
Curry	Meyer	25000
Lindsay	Meyer	9000
Turner	Firebrand	20000
Williams	Tweeties	18000
Adams	Meyer	22000

Manages Table

Empname	ManagerName
Jones	Collins
Smith	Collins
Hayes	Wills
Curry	Wills
Lindsay	Mulhare
Turner	Mulhare
Williams	Bond
Adamas	Bond

At the SQL> prompt, type and run (by hitting the Enter key) the following four SQL commands **individually** to create the table structures for the tables *employee*, *works*, *manages* and *company*.

```
CREATE TABLE employee
(employeeName CHAR(15) NOT NULL,
 street       CHAR(15),
 city         CHAR(10),
 PRIMARY KEY (employeeName));


CREATE TABLE works
(employeeName CHAR(15) NOT NULL,
 companyName  CHAR(15) NOT NULL,
 salary       NUMBER(7),
 PRIMARY KEY (employeeName, companyName),
 CHECK (salary >=0));


CREATE TABLE manages
(employeeName CHAR(15) NOT NULL,
 managerName  CHAR(15),
 PRIMARY KEY (employeeName));
```

```
CREATE TABLE company
(companyName CHAR(15) NOT NULL,
 city        CHAR(10) NOT NULL,
 PRIMARY KEY (companyName, city));
```

2.4. Insert data into tables

After the table structures are created in the database, insert the sample data (rows / tuples) shown in the above tables into the database tables individually:

Example syntax for inserting *rows* / *tuples* into a table is as follows:

```
INSERT INTO company
      VALUES('Meyer', 'Rye');

INSERT INTO employee
      VALUES ('Jones', 'Main', 'Harrison');

INSERT INTO works
      VALUES ('Jones', 'Tweeties', 15000);

INSERT INTO manages
      VALUES ('Jones', 'Collins');
```

Note: The data type (e.g., CHAR or NUMBER) of an inserted data must match the data type of the corresponding column in the table structure. For example, in table *company*, the data type of the column "companyName" is character (CHAR), its corresponding inserted data must be characters and must be within single quotes in the INSERT command, like 'Meyer'. For a number data type (e.g. NUMBER for the *salary* in the table *works*), however, the inserted value (e.g. 15000) does not need quotes.

Things To Do: Use the same way above to enter all sample data into the four tables.

2.5. Some SQL and SQL*Plus commands

Now, your database has four tables (i.e., *employee*, *works*, *manages* and *company*) with sample data.

To find out all tables owned by you, type the following SQL command at the SQL> prompt:

```
SQL> SELECT table_name FROM user_tables;
```

If you want to know whether a certain table is in your database, use the command:

```
SQL > SELECT table_name FROM user_tables
    2 WHERE table_name = '<tableName>';
```

<tableName> means you must type in a real table name at this position. For example, you can check whether the table *employee* exists in the database by running

```
SQL > SELECT table_name FROM user_tables
   2 WHERE table_name = 'EMPLOYEE';
```

To view a table structure, you can use a SQL*Plus command. The command is DESCRIBE (or DESC), which does not need a semicolon at the end because it is not a SQL command. The command format is

```
DESCRIBE <tableName>
```

For example, to view the structure of table *employee*, type the following SQL*Plus command at the SQL> prompt:

```
SQL> DESCRIBE employee
```

Things To Do: Use the same way to view structures of other tables.

To view the data in a table, you can use the SQL command SELECT. The command format is

```
SELECT * FROM <tableName>;
```

For example, to view the data in table *employee*, type the following SQL command at the SQL> prompt:

```
SQL> SELECT * FROM employee;
```

Things To Do: Use the same way to view the data of other tables.

2.6. Exit the Oracle database

After finishing your work, you can exit the SQL command environment and Oracle database by typing the SQL*Plus command EXIT or QUIT at the SQL> prompt, and pressing the "Enter" or "Return" key:

```
SQL> exit    or      SQL> quit
```


3. Brief Summary

- Use **Tera Term** to connect to the Oracle DBMS.
- Create tables by using the SQL command **CREATE TABLE**.
- Insert data into tables by using the SQL command **INSERT INTO**.
- Find out all tables in the database by using the SQL command **SELECT**.
- View the structure of a table by using the SQL*Plus command **DESCRIBE**.
- View all data in a table by using the SQL command **SELECT**.
- Exit the Oracle database by using the SQL*Plus command **EXIT** or **QUIT**.
- SQL and SQL*Plus commands are NOT case sensitive.

Please make sure to finish the above exercises on time.