

## COSC 344 Lab for Week 8

### Overview

The purpose of this lab and the next lab is to become familiar with Java & SQL, C&SQL, and PHP&SQL. Three versions are provided based on different host languages, and you should complete at least one of them.

### Assessment: 10 marks

You need to first choose your preferred host language, and then do the lab work according to the corresponding version. Part 2 and Part 3 will be assessed during the labs in Week 9.

## Java Language

### Part 1

There is no actual programming in this part since you will be given the program source files. This part is designed to take you through the process of compiling a Java program with JDBC. You will be using the program as a basis for the remaining parts of this lab and the next lab.

Create a file called *pass.dat*. Put your Oracle username and password in the file on separate lines. Do not have any spaces after the data. Save the file. At the UNIX prompt, type the following to set the protections on the file so that only you can read and write it:

```
chmod 600 pass.dat
```

First let's check *pass.dat*. Copy the files, *TestUserPass.java* and *UserPass.java*, from `/coursework/344/pickup/oracle-java`. Compile them by typing `javac TestUserPass.java`. Run the program by typing `java TestUserPass`. The program should print your Oracle username and password on separate lines with an *x* immediately before and after the data. If not, fix *pass.dat*.

Now let's use a Java program to connect to Oracle. Copy the file, *TestLogin.java*, from `/coursework/344/pickup/oracle-java`. Compile and run it. It reads your *pass.dat* and attempts to connect to Oracle. The program should respond that it connected to Oracle. Then the program terminates.

### Part 2

Use *TestLogin.java* as the basis for this part. Copy it to a new file with a different name.

Write a program to select one row of data from the *department* table based on the *mgrssn* field. Print the result.

### Part 3

Again use *TestLogin.java* as a basis to make the program for this part. Write a program to give each employee a 10% pay raise.

After your program executes, check the results in Oracle interactively. You can use the load script (*company.sql*) to recreate the tables if you need.

# C Language

## Part 1

There is no actual programming in this part since you will be given a program source file. This part is designed to take you through the process of compiling and linking a program using embedded SQL. You will be using the program as a basis for the remaining parts of this lab and the next lab.

Create a file called *pass.dat*. Put your Oracle username and password in the file on separate lines. Do not have any spaces after the data. Save the file. At the UNIX prompt, type the following to set the protections on the file so only you can read and write it:

```
chmod 600 pass.dat
```

First let's check *pass.dat*. Copy the files, *Makefile*, *getresponse.c* and *tut1.c*, from */coursework/344/pickup/oracle-C*. Compile and link the file by typing *make tut1 exe=tut1*. Run the program by typing *./tut1*. The program should print your Oracle username and password on separate lines with an *X* immediately before and after the data. If not, fix *pass.dat*.

Now let's use a C program to connect to Oracle. Copy the files, *Makefile* and *pt1.pc*, from */coursework/344/pickup/oracle-C*. Compile and run it. It reads your Oracle username and password from *pass.dat* and attempts to connect to Oracle. The program should respond that it connected to Oracle. Then the program terminates.

## Part 2

Use *pt1.pc* as the basis for this part. Copy it to a new file with a different name.

Write a program to select one row of data from the *department* table based on the *mgrssn* field. Print the result. Make sure your SQL SELECT statement only retrieves one tuple. Test it in Oracle interactively first.

## Part 3

Again use *pt1.pc* as a basis to make the program for this part.

Write a program is to give each employee a 10% pay raise. After your program executes, check the results in Oracle interactively. You can use the loading script to recreate the tables if you need.

# PHP Language

## Part 1: Basic connection

There is no actual programming in this part since you will be given program source files.

This part is designed to take you through the process of including a file containing sensitive information into a PHP script, and editing that script to contain the details needed to connect to an Oracle database and run an sql query. You will be using the program as a basis for the remaining parts of this lab and the next lab.

In this section where the instructions show <username> replace this with your login name.

For example, if your login name were `jobloggs` then

```
/devel/<username>/projects
```

becomes

```
/devel/jobloggs/projects
```

Use

```
$ ssh titanium.otago.ac.nz
```

to log into the titanium server using your departmental username and the initial password is set to your student ID number. The first time you log into titanium, use the command `passwd` to change your password to something appropriate -- do not use the same password as your normal departmental password, nor the same one as you use for your Oracle account.

Once you are logged on, there is an important configuration file to be set up – this file only needs to be set up once, but it must be set up correctly. It also gives us a chance to revise some command line instructions: type

```
$ umask
```

to check the current permissions that will be set for new files you create. If it is not “0027”, change it using:

```
$ umask 0027
```

which will make it so you have full access (the '0'), your 'group' can use but not alter things (the '2'), and other people have no access (the '7'). These are the opposite of the values you would use when changing permissions using the command `chmod`

Change into the `/home/includes/<username>` directory

```
$ cd /home/includes/<username>
```

and copy the configuration file

```
/coursework/344/pickup/oracle-php/connenv
```

into this directory.

You need to make sure that all the settings for this file have been correctly set before (and after) editing it – so

```
$ ls -l
```

should list the directory, showing the file looking like

```
-rw-r-----. 1 jobloggs svr212 1402 Jun 26 11:37 connenv
```

The important things to check are the permissions which were set by the `umask` instruction:

```
-rw-r-----.
```

if these permissions are incorrect, then you can use `chmod` to fix them; and you need to check the group that has access to the file – it must be the group `svr212`

You cannot change the GID yourself if it is wrong - you must send a message to cshelp so a sysadmin can do it - that is why you must set the correct umask before copying the file. This setting allows the web-server process to read your file, but nobody else who can log into the server can read it. Placing the file in a directory outside the web-server directory tree (i.e. /home/includes) is done for added security.

Edit the line in the configuration file `connenv` which specifies the connection details and replace them with your oracle login name and password

i.e. `$conn = oci_connect("ORACLE-USERNAME", "ORACLE-PASSWORD");`

When you are logged into titanium you cannot use many of the familiar text editors and you cannot open new windows from titanium, so you have to edit the file inside the terminal window – if you are familiar with vim, or nano you can use those, otherwise we'd suggest using emacs

```
$ emacs connenv
```

Notes:

- Do not include an “&” after the filename. It will not work.
- Don't try using the mouse when editing ... move it to the side and forget it.
- To save your work use `<ctrl> x <ctrl> s`
- To quit from emacs use `<ctrl> x <ctrl> c`

Now that the configuration file has been set up, you can use it for making any database connections from a php file – the first php project will be to ensure that we can successfully connect to the database.

When you are working on titanium you need to change your working directory from your home directory to `/devel/<username>/projects` (replace `<username>` with your username)

```
$ cd /devel/<username>/projects
```

**Note:** Your home directory on titanium has the same system path / name as that on the lab computers, but it is NOT the same directory. You will not have (direct) access to your usual files.

Copy the php file `connect.php` from `/coursework/344/pickup/oracle-php` into your `/devel/<username>/projects` directory and edit it so the path for the include file is correct  
i.e. `include("/home/includes/<USERNAME>/connenv");`

Make sure the variable which contains the connection details is named correctly (in this case `$conn` )

First, check that the connection details are correct by typing

```
$ php ./connect.php
```

on the command line "Connected to Oracle!" should appear and the program will terminate.

If you are using a computer in the lab, you can access `connect.php` in your web browser with either of these 2 links (replacing `<username>` with your username first, of course).

<https://titanium.otago.ac.nz:8443/<username>/projects/connect.php>  
<https://titanium.otago.ac.nz:8443/~<username>/connect.php>

Due to security reasons, it is not allowed for an off-campus computer to connect to titanium directly. To allow your computer to access a web page in titanium, you can run the following command in a terminal of your computer to set up an SSH tunnel between your computer and the hextrime server first.

ssh -L 9999:titanium.otago.ac.nz:8443 <username>@hextreme.otago.ac.nz

where <username> should be replaced with the username you used to log into the hextreme server. Now start a browser on your computer, and access connect.php with the following link (replacing <username> with your username first, of course) :-

<https://localhost:9999/~<username>/connect.php>

The web-browser should respond that it connected to Oracle.

*For the next parts you need to make sure you have preloaded the data being queried into the database, using the company.sql file from the /coursework/344/pickup/oracle-sql directory.*

## **Part 2: Simple query**

Copy query.php to your /devel/<username>/projects directory and follow the instructions you used for editing the include line in connect.php

Access query.php with a browser, as you did for connect.php

Edit query.php to select one row of data from the department table based on the mgrssn field.

Print the result. Make sure your SQL SELECT statement only retrieves one tuple.

Test your selection query in Oracle interactively first.

## **Part 3: Simple update**

Use a new copy of query.php as the basis for this part.

Copy the file to your /devel/<username>/projects directory, giving it a different name (such as query07\_3.php), and follow the instructions you used for editing connect.php

Access query07\_3.php with a browser, as you did for connect.php

Edit query07\_3.php to give each employee a 10% pay raise.

After your program executes, check the results in Oracle interactively.

You can reload company.sql in Oracle to restore the original data in the tables if you need.

If you have time – modify the php file so it displays the employees' first name, last name, and salary both before and after the update.