COMP40725
Lab Book 8

➔ Create a folder called ***Lab8*** in your ***COMP40725*** folder. You may wish to have this inside a folder that is being synced by Google Drive.
➔ Create a new workspace called ***Lab8*** using the webinterface and connect to this using ***SQL Command Line*** .
➔ Load Customer.sql from Lab Book 5 into the database.
➔ Use Notepad++ or TextWrangler to write SQL queries. Test these queries in your ***Oracle 11g XE Database*** system. You should use ***SQL Command Line (sqlplus tool).***

➔ To enable output to screen on ***sqlplus*** run the following command when you start your session. It must be set for each user and only lasts for the session: ***SET SERVEROUTPUT ON;***
➔Use the command ***show errors;*** to illustrate errors
➔Ensure the entire ***SQL*** file can be copied and pasted into ***SQL Command Line*** and will work without errors.
➔The answer to all questions should include proof that they are correct by outputting to screen e.g. Q4,5,6 will need and output or a select statement to do this.
➔ Use comments in your sql queries: -- is for a single line, and /*... */ for multiline. Putting comments on new line avoid unexpected errors.
➔Mark will be lost for poor commenting. Example of good commenting

```
/* Q3 Output all the non prime numbers from 1 to 30 inclusive */
<code>
--flag to store if the current number is a prime
<code>
-- check all number from 1 to 30
<code>
-- assume the current number is a prime
<code>
-- for loop to generate the devisors
<code>
-- if the number divides evenly by the divisor
<code>
-- then the number is not a prime
<code>
-- exit inside loop
<code>
-- if flag as been set to indicate that current number is not a prime
<code>
-- output the non prime number
<code>
/
```

Submit the SQL statements in a ".sql" that you used to test all of the above queries and name it "Lab8_StudentNumber_FirstName_LastName.sql".

Hint: This lab requires you to apply what was covered in **Lecture 13 & 14**

1. Demonstrate the use of a FOR loop, which counts from 1 to 100. DBMS_OUTPUT.PUT_LINE each number.

2. Demonstrate the use of a nested for loop (i.e. a for loop inside a for loop) with 10 iterations per loop. DBMS_OUTPUT values in the nested loop - Example output should be (1.1, 1.2, 1.3,..)

3. Demonstrate the use of a for loop which counts from 1 to 30. In each iteration, DBMS_OUTPUT.PUT_LINE all values which are NOT prime numbers. Hint: Search online for algorithms/code examples from C/Java and then translate into SQL.

4. Multi part
    a. Create a BEFORE INSERT trigger for the Employee table (from lab 3). This trigger should automatically calculate the idEmployee value from a sequence.
    b. Show the Employee table.
    c. Create a PL/SQL block that demonstrates the trigger working, where, when you INSERT, supply the list of column names without the primary key.
    d. Show the Employee table.

5. Multi part
    a. Add a new salary column to the Employee table.
    b. Update the table to populate this column with a range of values which are less than, equal to, and greater than 100,000.
    c. Create a trigger which checks if an employee's salary is set to be more than 100,000. Output the salary value and the employee's id.
    d. Test by updating and employee to a new salary of 50,000 and then updating the same employee to 110,000.

6. Multipart
    a. Show the Employee table.
    b. Create a FOR loop which inserts 1000 employees into the employee table - it's ok if they have the same name but other values should not all be the same, this should use the trigger from part 5 to generate the primary keys.
    c. Show the Employee table.

7. Demonstrate the use of a cursor, where the cursor should access all employees. If any employee has a salary of less than 20000 raise an application error (not exception). If any employee has a salary greater than 90,000 output a message and then re-raise the exception. NB you must output to screen to show it is working.

8. Use a SELECT INTO statement to (intentionally) return 0 rows. This will cause a NO_DATA_FOUND exception. Catch the exception and output an error message.

This is not expected to list ALL employees. If any employee has a salary outside this range, the error/exception should happen.