Testing your Oracle installation (The Oracle edition that is installed in the Computing Center in the Dept, is Oracle 11G. Examples shown in class are tested on this version).

Assuming you have successfully installed Oracle Express Edition on your system, you may want to test it with a few commonly used SQL commands.

Issuing SQL statements from the command line.

Do not just directly copy the code from this file to your SQL prompt, or you may encounter errors such as the quotation mark error.

Creating a Table (or two)

At SQL> prompt, type the following lines to create a table called trial, with two attributes, tnum (an integer) and str (a string of 10 characters).

```
CREATE TABLE trial (
        tnum INTEGER,
        str VARCHAR2(30)
);
```

Note: SQL is NOT case-sensitive. You may enter text on one line or on several lines. If your command runs over several lines, you will be prompted with line numbers until you type the semicolon that ends any command.

Do you see the message, Table created, at SQL prompt?

Inserting Tuples

Now add some data tuples into table, trial.

Type the following SQL statement:

```
INSERT INTO trial VALUES (1,’A test’);
```

You should see the message, “1 row created”.

Insert at least 2 more tuples with values of your choice**.**

Fetching data from the table

At the SQL prompt, **type the following statement, to retrieve all the rows in the table, trial.**

```
SELECT * FROM trial;
```

Do you see all the tuples you have entered in the table?

Try to enter a duplicate tuple (1,’A test’) into trial, with the statement**:**

```
INSERT INTO trial VALUES (1,’A test’);
```

Did you succeed in adding a duplicate tuple? You were able to add a duplicate tuple because we did not define a primary key on the table.

Getting Meta Information about your Database

The system keeps information about your database (the tables and constraints etc, that you have created) in certain system tables. You can retrieve the names of the tables that you have created from the system table, USER_TABLES.

Type the following query:

```
SELECT TABLE_NAME
FROM USER_TABLES;
```

Do you see the table(s) that you have created so far?

Executing SQL commands from a script file.

Step 1: Create a text file called **data.sql** in a folder of your choice**.** This file will contain the SQL statements that you want to execute.

You have a couple of options to create the text file:

Using a text editor, create the file and include the SQL statements.

Type the following statements to insert values into the table test.

```
INSERT INTO trial VALUES (10,' ten');
INSERT INTO trial VALUES (11,' eleven');
```

You can type statements to insert values of your choice.

To execute the commands from the script file, data.sql as follows:

SQL> start <*Path/filename*> or SQL> @<*Path/filename*>

For example,

SQL> start /*yourpathname*/data.sql

or

SQL> @/ *yourpathname*/data.sql

Check if the data you have entered has been loaded into the table successfully.

Capturing and Recording your Session

One way to capture your SQL session is to use the spool command that SQLPLUS provides to save query results to a file.

At the SQL> prompt, type **spool <*filename*>.** For example,

SQL> spool foo;

This will create a **foo.lst** file in the current directory and will capture all input and output during your session at SQLPLUS (until you exit SQL).

You can terminate the capture with the command, **spool off;**

You can print the foo.lst file for submission (for assignments).

Issuing operating system commands from SQL

You can issue host operating system commands from SQL as follows:

Assuming you are accessing Oracle on Unix systems, you can list the contents of the current directory as

SQL> host ls

To see your current directory,

SQL> host pwd

Oracle SQL Datatypes, Reference:

http://www.ss64.com/orasyntax/datatypes.html