CA2 Lab Report

**Student Name:** Daniel Gallagher

**Course:** BSC Computing

**Module:** SQL Programming

**Lecturer:** Angela Sweeney

**Submission Date:** 31/03/2022

**Part A**

**Aims/Description**

The purpose of this lab is to create a connection to the Oracle XE database using SQL developer. The various ways that I’m going to demonstrate on how to make a connection to an Oracle database is:

1. Creating a connection to system on a VM using SQL developer
2. Creating a username schema called student.
3. Connecting to the student schema and running the create student SQL script to create the tables and populate with the provided inserts.
4. Writing a query to describe the structure of one of the tables.
5. Writing a query that joins data from 3 or more tables.
6. Writing a query that uses 2 or more restricting data operators or keywords.
7. Writing a query that uses an aggregate function.
8. Writing a query that alias all columns appropriately in a table.

**Methods**

The equipment/resources that I used during the practical CA to complete this was:

1. Eircom connect which is provided by LYIT and allowed me to have access to VMs which let me use SQL developer on my own machine.

**1.** Creating a connection to system on a VM using SQL developerGraphical user interface, application

Description automatically generated

**Aim 1:** To complete this aim, I opened SQL developer and went to the top left corner and clicked on the plus sign under connections. This brings up a window for creating new connections. I provided a connection name which can be anything you like if its user friendly to the SQL developer. Also, I provided the username and password which will be system and system for the password. I selected the option to save password which will not prompt you every time for the password every time I tried and connect to the connection which is system in this example shown in the screen shot above. I also provided a service name called xepdb1.

**2.** Creating a username schema called student.

Graphical user interface, text, application

Description automatically generated with medium confidence

**Aim 2:** I am given a SQL script called create Student user that creates the student user and assigns the password as Mustbe12bytes and as well grants privileges for the user. Also, at the top right on SQL developer we want to make sure that we are connected to the system connection as seen in the screen shot above.

Graphical user interface, application

Description automatically generated**3.** Connecting to the student schema and running the create Student SQL script to create the tables and populate with the provided inserts.

**Aim 3:** For this aim, I opened the window for creating new connections at the top left for SQL developer under connections. I named the database connection student\_conn\_xepdb1 and fill in the username with student and the password being Mustbe12bytes. I also provided a service name called xepdb1.

Graphical user interface, text, application

Description automatically generated

**Aim 3:** To complete this aim, I opened the create Student SQL script on the SQL developer and run the script to create the tables and populate with the provided inserts. Also, at the top right of the SQL developer, I made sure that I was connected to the student schema, so that the tables are created in the student schema.

**4****.** Writing a query to describe the structure of one of the tables. This Query shown on the image below describes the course table.



**Aim 4:** For this Aim, I used the describe command on the course table which shows you the structure of the table which includes name of the column, datatype of column and the nullability which means, that the column can contain null values or not.

**5.** Writing a query that joins data from 3 or more tables. This Query shown on the image below joins the section, course, and the instructor tables.

Graphical user interface, text, application

Description automatically generated

**Aim 5:** To do this Aim, I first joined the section table to the course table using inner joins which combines rows from two or more tables. Then I joined the section course number with course number in the number table. Then for the third join I done this again to join another table, but we join section to instructor using the instructor id in section and the instructor id in section.

**6.** Writing a query that uses 2 or more restricting data operators or keywords. This Query shown on the image below selects all columns from the course table and selects all columns where the cost is not null, and the course description starts off with an ‘I’.

Graphical user interface, text, application

Description automatically generated**Aim 6:** To complete this aim, I used the select command and selected every column in the table from the course table. Then I used a where to select from the cost column where the cost is not null and, that the description column starts off with a ‘I’.

**7.** Writing a query that uses an aggregate function. This query shown on the image below counts all the student IDs and distinct student IDs from the student table.

Graphical user interface, text

Description automatically generated with medium confidence

**Aim 7:** For this aim, I used the count aggregate function which returns the number of rows that matches a specified criterion from the student table, and I also used the Distinct with the count function which counts the non-duplicate values of student ids.

**8.** Writing a query that alias all columns appropriately in a table. This Query shown in the image below alias all the columns in the grade type table.

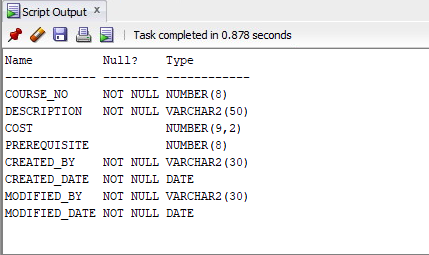
Text

Description automatically generated

**Aim 8:** To complete this aim, I used the alias which are used to give a table, or a column in a table, a temporary name. Aliases are mostly used to make a column name more readable. An alias is created with the as keyword.

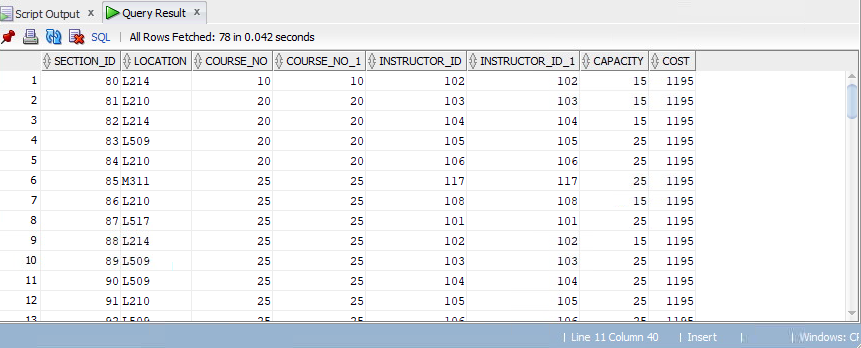
**Results/Discussions and Conclusions**

**4.** Writing a query to describe the structure of one of the tables. This query shown on the image below describes the course table.



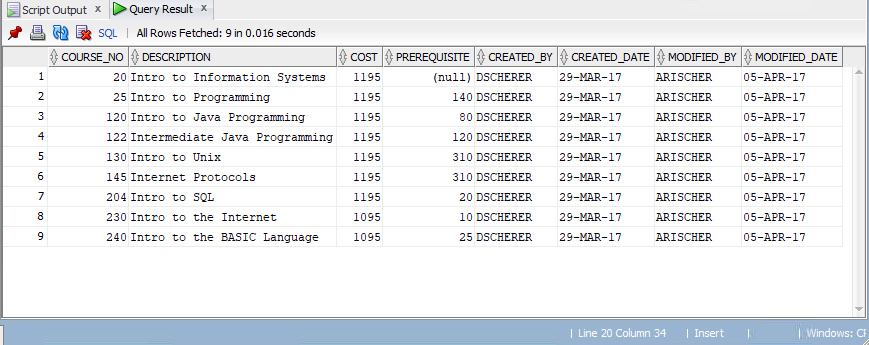
**Aim 4:** For the result of this aim, it shows the name of each column, if its null or not and what type of column it is like number, varchar, date etc from the course table.

**5.** Writing a query that joins data from 3 or more tables. This query shown on the image below joins the section, course, and instructor tables using “course no” and “instructor ID”.



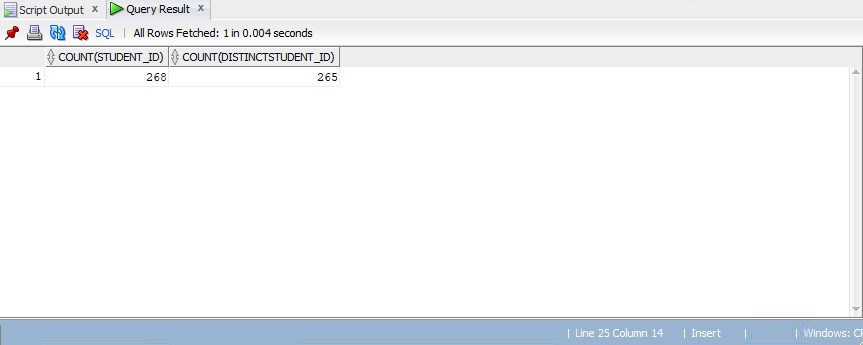
**Aim 5:** The result of this aim shows that the course no in the section table is equal to the course no in the course table. And that the instructor id in the section table is equal to the instructor id in the instructor table. It shows us that our table of 3 joins are working.

**6.** Writing a query that uses 2 or more restricting data operators or keywords. This query selects all from the course table and selects where the cost is not null, and the course description starts with an “I”.



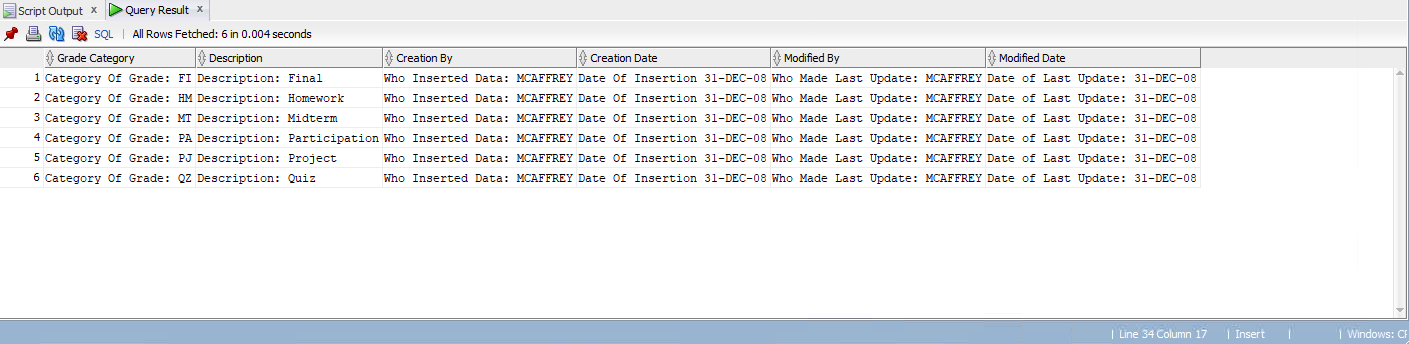
**Aim 6:** The result of this aim shows us everything from the table course and uses where function to see if the cost column is not null and that description starts with an I from the course table.

**7.** Writing a query that uses an aggregate function. This query counts all the student IDs and distinct student IDs from the student table.



**Aim 7:** For the result of this aim, it counts the number of student ids, and it also counts the number of non-duplicate student ids using the Distinct clause.

**8.** Writing a query that alias all columns appropriately in a table. This Query shown in the image below alias all the columns in the grade type table.



**Aim 8:** The result of this aim alias all columns appropriately from the grade type table. I first added a string of which is going to be added to each value in the column and then gave the column a temporary name like grade category for example from the grade type column.

**Part B**

**Aims/Description**

The purpose of this lab is to create a connection to the Cloud Database using SQL developer. The various ways that I’m going to demonstrate on how to make a connection to an Oracle database is:

1. Creating an oracle cloud account (using the email I received from oracle).
2. Creating a database instance.
3. Obtaining an instance wallet.
4. Connecting to the cloud database using SQL developer using LYIT VM.
5. Creating the same user schema in part a called student on the cloud database.
6. Connecting to student and running the create student SQL script to create the tables and populate with provided inserts.
7. Writing a query that describes the structure of one of the tables.
8. Writing a query that demonstrates the following

* Joins data from 3 or more tables.
* Uses 2 or more restricting data operators or keywords.
* Selects a date column datatype.
* Uses an aggregate function.
* Alias all columns appropriately.

**Methods**

The equipment/resources that I used during the practical CA to complete this was:

1. Eircom connect which was provided by LYIT and allowed me to have access to VMs which let me use SQL developer on my own machine.
2. Oracle cloud free tier which was provided LYIT and allowed me to have access to cloud database.

**1.** Creating an oracle cloud account (using the email I received from oracle).

Graphical user interface

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Aim 1:** To complete this aim shown on the image above, I went on the oracle cloud website, signed up filled in all the necessary information and then oracle sent me an email to verify my account and when I verified my account it took a short while before my account was created and I was able to sign into it.

**2.** Creating a database instance.

Graphical user interface, text, application, email

Description automatically generated

**Aim 2:** To do this aim shown on the image above, I signed into to my oracle account using my credentials. When I log in, I am shown all the services that are available to me on the cloud services dashboard. I selected the create an ATP database service from the dashboard and when u select that option you are shown the create autonomous database window, where you are going to specify the configuration of the instance. I called the name of the autonomous database DMDB and the database name DMDB as well. I selected the workload type to be transaction processing as I’m going to be creating transactional type database schemas. For the deployment type I selected the shared infrastructure and configured the database as always free. For database version, I selected the 21c as it’s the most recent version. For creating the administrator credentials, I filled in a password which is going to be used to access the autonomous database service console. For choosing the network access I selected the secure access from everywhere and selected license included as I’m not going to be bringing my own. And finally, all that I have left to do is click on the create autonomous database button at the bottom left. Once I done this my instance began provisioning for a short while where once available, the colour will change from orange to green.

**3.** Obtaining an instance wallet.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Aim 3:** For the completion of this aimshown on the image above,I signed into my oracle account using my credentials, and clicked on the navigation tab at the top left of the page and navigated to the autonomous transaction processing database that I created on aim 2. When I’m on page that shows me all my info about the database I created, I select the option DB connection which shows me the option to download my instance wallet. When I click on download wallet, it does prompt me for a password. So, I filled in my password and confirmed my password as well and then clicked download. The instance wallet is a zip folder which contains the parameters and credentials that are required from me to connect from SQL developer to the cloud schemas.

**4.** Connecting to the cloud database using SQL developer using LYIT VM.

Graphical user interface, text, application

Description automatically generated

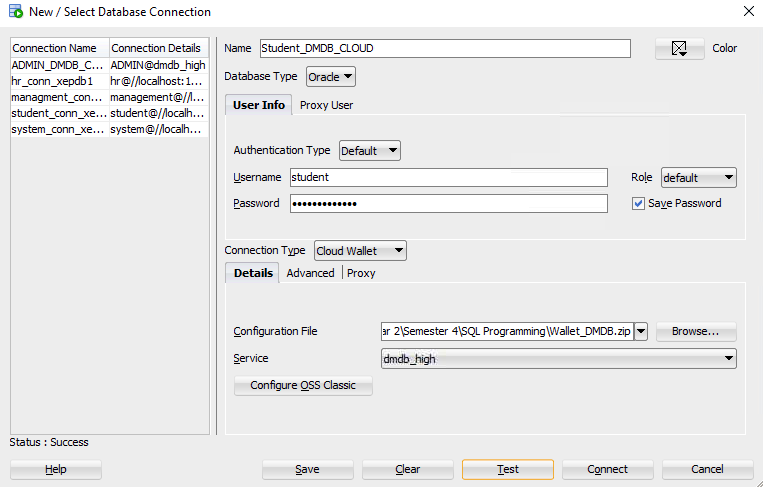
**Aim 4:** To complete this aim shown on the image above, I open SQL developer and click on create a database connection at the top left and then I’m shown the window to create my database connection. First off, I gave my database a suitable name called ADMIN\_DMDB\_CLOUD cause that’s the name of the database in the cloud. Then I filled in my username which is ADMIN and the password which is the password of when I created my account. Then for connection type, I select cloud wallet and I give the location of where I saved my wallet that I download on aim 3. I test it to see if I can connect to it an if I can I get a message at the bottom left saying that the status was a success, and I can connect to admin in the cloud.

Graphical user interface, text, application

Description automatically generated**5.** Creating the same user schema in part a called student on the cloud database.

**Aim 5:** For the completion of this aimshown on the image above, I open SQL developer, connect to the ADMIN\_DMDB\_CLOUD and run the schema that creates the user Student using the create Student SQL script while being connected to ADMIN\_DMDB\_CLOUD as shown at the top right of the SQL developer. This script creates the user called student and creates the password which is Mustbe12bytes and grants privileges to the student user.

**6.** Connecting to student and running the create student SQL script to create the tables and populate with provided inserts.

**Aim 6:** I opened SQL developer and clicked on the create database connection, which then I entered the name of the database connection and then the username which is student and the password which is Mustbe12bytes. I also added to where the location of my wallet is.

Graphical user interface, text, application, email

Description automatically generated

**Aim 6:** To complete this aim, I connect to the student connection I created called Student DMDB\_CLOUD and then I open the create Student SQL script that creates the tables and populates with provided inserts. At the top right I make sure I was connected to the student schema, so that the tables are created in the student schema.

**7.** Writing a query that describes the structure of one of the tables. This query describes the table called grade.



**Aim 7:** To complete this aim, I used the describe command on the grade table which shows you the structure of the table which includes name of the column, datatype of column and the nullability which means, that the column can contain null values or not.

**8.** Writing a query that demonstrates the following

* Joins data from 3 or more tables.
* Uses 2 or more restricting data operators or keywords.
* Selects a date column datatype.
* Uses an aggregate function.
* Alias all columns appropriately.

**8.1** A query that joins data from 3 or more tables. This query joins 3 tables together using inner joins.

Text

Description automatically generated

**Aim 8.1:** For the completion of this aimshown on the image above, I firstly joined the section table with the course using the course no in both tables. Then to join the 3rd table, I joined section and instructor using the instructor id in both tables.

**8.2** A query thatuses 2 or more restricting data operators or keywords. This Query shown on the image below selects all columns from the course table and selects all columns where the cost is not null, and the course description starts off with an ‘I’.

Text

Description automatically generated

**Aim 8.2:** To complete this aim, I used the select command and selected every column in the table from the course table. Then I used a where to select from the cost column where the cost is not null and, that the description column starts off with a ‘I’.

**8.3** A query selects a date column datatype. This query selects everything from course and selects all columns where the created date is greater than or equal to 29-Mar-17.

Text

Description automatically generated

**Aim 8.3:** For this aim, I selected everything from the course table and used a where command and select only where created date was greater than or equal to the 29 March 2017.

**8.4** A query that uses and aggregate function.



**Aim 8.4:** For the completion of this aim, I used a select max function on student id from the student table. The max function returns the largest value of the selected column which is student id.

**8.5** A query that alias all columns appropriately**.** This query shown in the image below alias all the columns in the grade table.

Text

Description automatically generated

**Aim 8.5:** To complete this aim, I used the alias which are used to give a table, or a column in a table, a temporary name. Aliases are mostly used to make a column name more readable. An alias is created with the as keyword.

**Results/Discussions and Conclusions**

**Aim 7:** A query that describes the structure of one of the tables. This query describes the grade table.

Graphical user interface, text, application

Description automatically generated

**Aim 7:** For the result of this aim, it shows the name of each column, if its null or not and what type of column it is like number, varchar, date etc from the course table.

**Aim 8.1:** A query that joins data from 3 or more tables. This query joins 3 tables together using inner joins.

Table

Description automatically generated

**Aim 8.1:** The result of this aim shows that the course no in the section table is equal to the course no in the course table. And that the instructor id in the section table is equal to the instructor id in the instructor table. It shows us that our table of 3 joins are working.

**Aim 8.2:** A query thatuses 2 or more restricting data operators or keywords.

Graphical user interface, table

Description automatically generated

**Aim 8.2:** The result of this aim shows us everything from the table course and uses where function to see if the cost column is not null and that description starts with an I from the course table.

**Aim 8.3:** A query selects a date column datatype.

Graphical user interface

Description automatically generated with medium confidence

**Aim 8.3:** The result of this aim selects everything from the course where the created date was greater than or equal to 29/MAR/17.

**Aim 8.4:** A query that uses and aggregate function.



**Aim 8.4:** Theresult of this aim selects the max (the largest value) of student’s id from the student table.

**Aim 8.5:** A query that alias all columns appropriately**.**

Graphical user interface, application, table, Excel

Description automatically generated

**Aim 8.5:** The result of this aim alias all columns appropriately from the grade table. I first added a string of which is going to be added to each value in the column and then gave the column a temporary name like Student IDs for example from the grade type column.

**References**

* <https://www.w3schools.com/sql/default.asp>
* <https://docs.oracle.com/cd/E11882_01/server.112/e41085/sqlqr01001.htm#SQLQR110>