**Part B**

**Student Name:** Eryk Gloginski **(L00157413)**

**Course:** BSc Computing

**Module:** SQL Programming

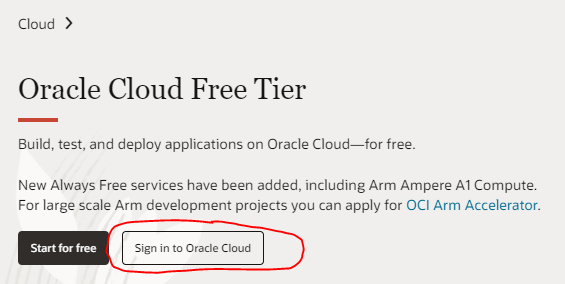
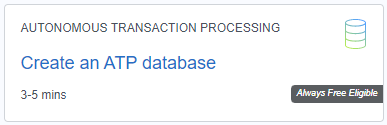
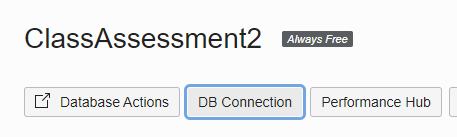
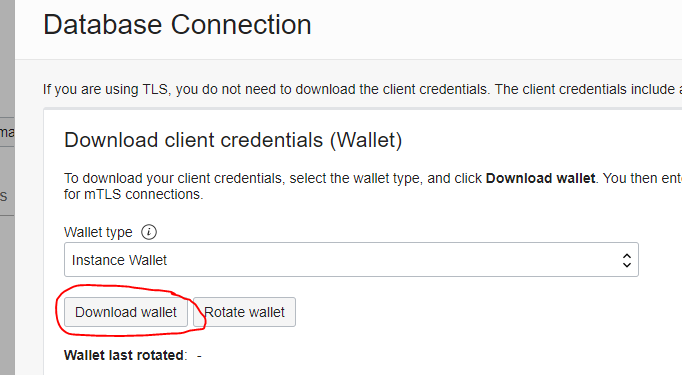
**Lecturer:** Angela Sweeney

**Submission Date:** 3/29/2022

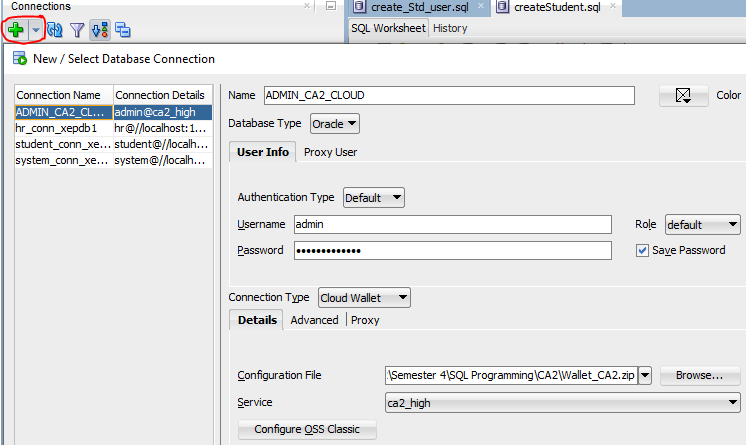
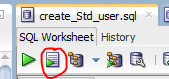
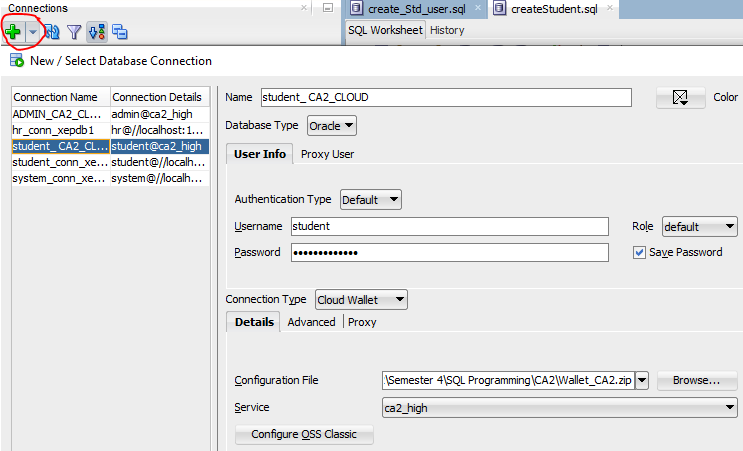
**Aims or description:** Connection to your Cloud Database using SQL developer installed on own PC, College or VM.

**Methods:**

**Section 1: Creating an Account**

I have started working on this part by creating an **Oracle Cloud** account. The *Figure 1* below highlights the **sign in** option and since I already have an account, I just sign into it on. Website: <https://cloud.oracle.com/>  
*Figure 1*  
I created an Autonomous Database Instance in *Figure 2* and I have obtained an **instance wallet** from the “**DB Connection**” in *Figure 3* and **download** it from *Figure 4*.  
*Figure 2*  
*Figure 3*  
*Figure 4*

**Section 2: Setting up the Connections**

I create a “**ADMIN\_CA2\_CLOUD**” connection using the instance wallet that I have obtained. It is shown in *Figure 5* below.  
  
*Figure 5*  
I have then run the provided SQL script called “**create\_Std\_user.sql**” to create a “**student**” schema using the button (**F5**). The *Figure 6* below shows how you run the script on the connection.  
*Figure 6*  
As the script gives the “**student**” schema the basic necessary rights to connect and create tables, I create another connection called “**student\_ CA2\_CLOUD**”. This can be seen in *Figure 7* below.  
*Figure 7*  
With this connection, the same way I have run the “**create\_Std\_user.sql**” script in *Figure 6*, I use the “**createStudent.sql**” script to create and populate the tables in the “**student**” schema which takes a lot longer than on the local database used in the previous part of this question due to it being a remote.

**Section 3: The Query**

I then describe the structure of the table “**ENROLLMENT**” which returns me the basic info on each column, for example, what type of value the ”**STUDENT\_ID**” takes and if it can be ”**NULL**” or not. The below image shows the code used and the output generated. The *Figure 8* below shows the code used and the output generated.  
Graphical user interface, text, application, email

Description automatically generated*Figure 8*

Write a query to display only the Java course numbers and the amount of people taking them.

Therefore, I display the “**COURSE\_NO**” for the Java courses and count the “**STUDENT\_ID**” from the tables “**COURSE**”, “**ENROLLMENT**”, “**SECTION**” and “**STUDENT**”. I link the “**COURSE\_NO**“ in the “**COURSE**“ table with the one in “**SECTION**”, I link the “**SECTION\_ID**“ in the “**SECTION**“ table with the one in “**ENROLLMENT**“ and I link the “**STUDENT\_ID**” in the “**ENROLLMENT**” table with the one in ”**STUDENT**” using an implicit join. I display the rows where “**COST**” in the “**COURSE**” table is not “**NULL**” and the “**DESCRIPTION**” in the “**COURSE**” table contains the word Java. I lastly group the query by the “**COURSE\_NO**” and I order it by the string of the counted “**STUDENT\_ID**” in a descending order. The *Figure 9* below shows the code used and the output generated.  
Graphical user interface, text, application

Description automatically generated *Figure 9*