**Part A**

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

**Course:** BSc Computing

**Module:** SQL Programming

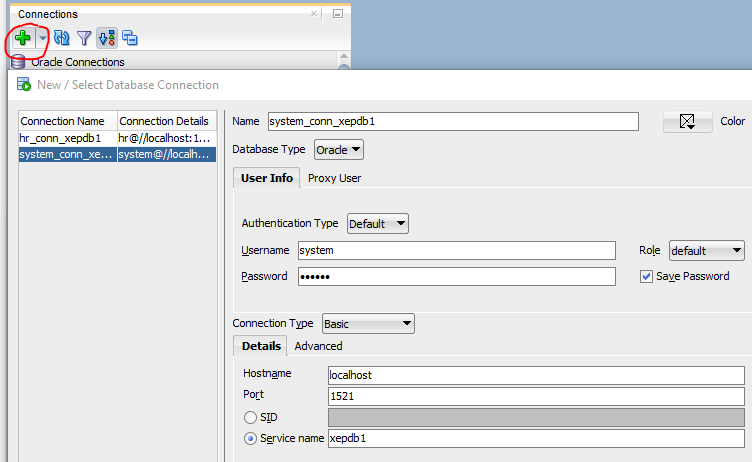
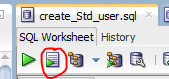
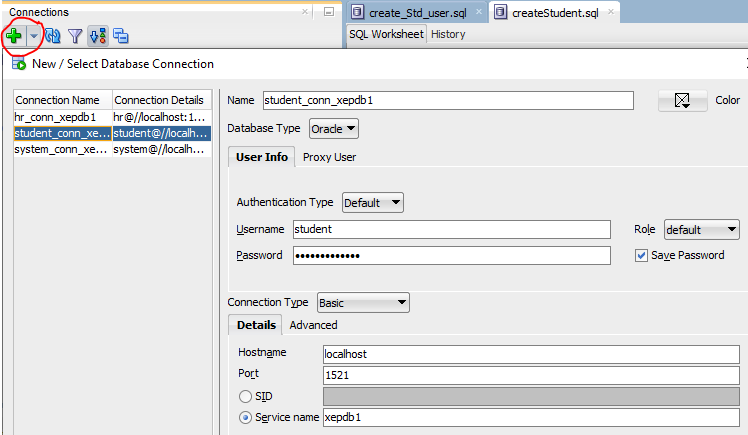
**Lecturer:** Angela Sweeney

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

**Aims or description:** Connection to a local Oracle XE database using SQL developer on your own PC, college or VM.

**Methods:**

**Section 1: Setting up the Connections**

Using “**SQL Developer 18c**”, I have started working on this part by creating the “**system\_conn\_xepdb1**” connection. The *Figure 1* below shows what how you create the connection.  
 *Figure 1*  
I have then run the provided SQL script called “**create\_Std\_user.sql**” to create a “**student**” schema using the button (**F5**). The *Figure 2* below shows how you run the script on the connection.  
 *Figure 2*  
As the script gives the “**student**” schema the basic necessary rights to connect and create tables, I create another connection called “**student\_conn\_ xepdb1**”. The *Figure 3* below shows what how you create the connection.  
 *Figure 3*  
With this connection, the same way I have run the “**create\_Std\_user.sql**” script in *Figure 2*, I run the “**createStudent.sql**” script to create and populate the tables in the “**student**” schema.

**Section 2: The Query**

I then describe the structure of the table “**STUDENT**” 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 *Figure 4* below shows the code used and the output generated.  
  
Graphical user interface, text, application

Description automatically generated *Figure 4*

Write a query to display the students who got a final grade higher than 70 in a descending order.

Therefore, I display the “**STUDENT\_ID**“ and the rounded average “**NUMERIC\_GRADE**“ from the tables “**ENROLLMENT**“, “**GRADE**“ and “**STUDENT**“. I link the “**STUDENT\_ID**“ in the “**STUDENT**“ table with the one in “**ENROLLMENT**“ and I link the “**SECTION\_ID**“ in the “**ENROLLMENT**“ table with the one in “**GRADE**“ using an implicit join. I display the rows where “**NUMERIC\_GRADE**“ is between 70 and 100 and the “**GRADE\_TYPE\_CODE**“ is “**FI**”. I lastly group the query by the “**STUDENT\_ID**” and I order it by the string of the average “**NUMERIC\_GRADE**” in a descending order. The *Figure 5* below shows the code used and the output generated.  
Table

Description automatically generated with medium confidence *Figure 5*