CST2355 – Database Systems Lab Assignment 8

Student Name: \_\_Yanzhang Wu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID: \_\_\_ 41056465 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

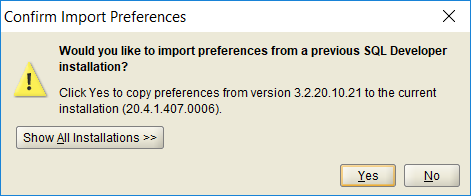
Student email: wu000298@algonquinlive.com\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Hand-in:

1. The lab assignment will be graded out of a maximum 4 points.
2. This template should be used to submit your lab assignment.
3. Make sure you have enough screenshots to completely document that you have completed all the steps.

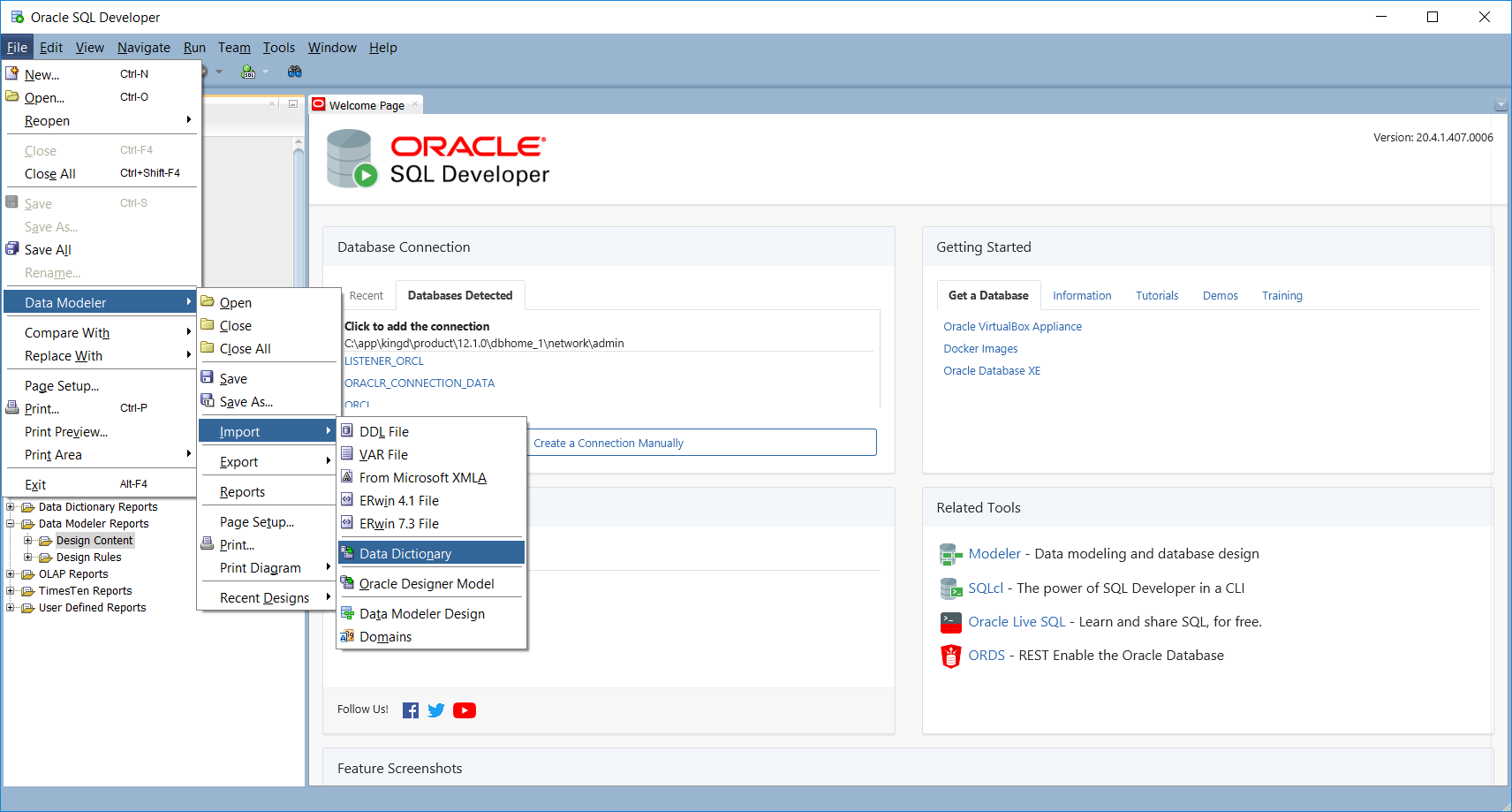
# Activities (Steps):

1. First thing we need to do is update SQL Developer to the latest version:
   1. Go to: <https://www.oracle.com/tools/downloads/sqldev-downloads.html> and download SQL Developer version 20.4 (or later). (If it is newer than 20.4, your installation/configuration steps may be slightly different.)
   2. When it is downloaded to your PC, extract all the contents of the .zip file.
   3. Then navigate to find ‘sqldeveloper.exe’ and using a right-click menu create a short-cut on your desktop by selecting the ‘Send To’ desktop option.
   4. Run the application.
   5. You will likely be prompted to import your preferences from the previously-installed SQL developer version:

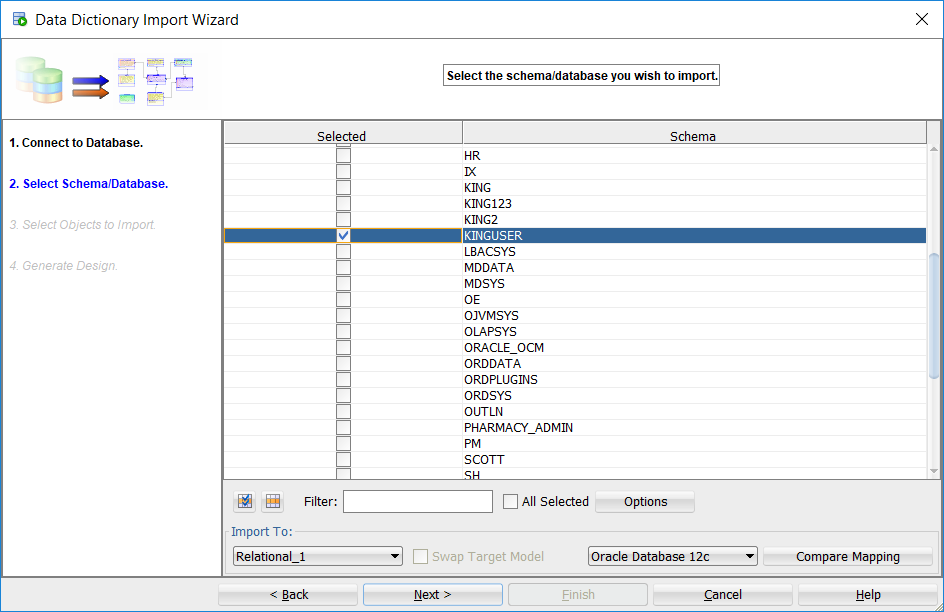


* 1. If so, import the preferences.

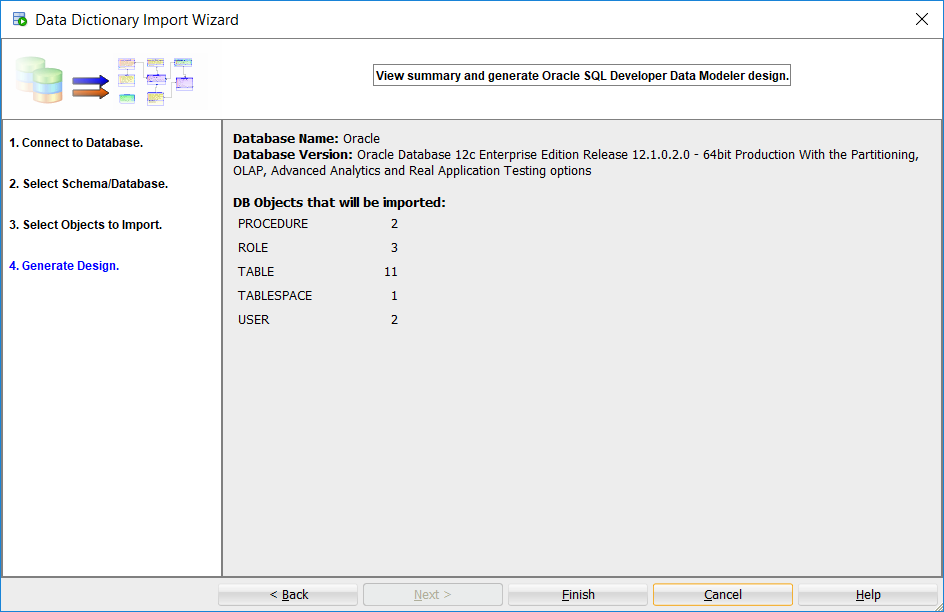
1. **Select the ‘*yourname*UserConnection*’* from the list of connections**, and then from the File menu, select: Data Modeler, Import, Data Dictionary (see below):



* 1. And click on Next. You will be prompted for your password, and then given a list of schemae from which to select.



* 1. Select the *yourname*User schema and then click Next.
     1. You will be prompted for what items you would like to import.
        1. Select all of the 11 Tables
        2. Select *yourname*User and MDSYS from Users
        3. Select all of the 4 Roles from the original database
        4. Select CST2355 from Tablespaces
        5. Select both of the 2 Stored Procedures.
  2. Then on the confirmation screen, select ‘Finish’



* 1. When completed, you will see the E-R diagram. Here is mine:

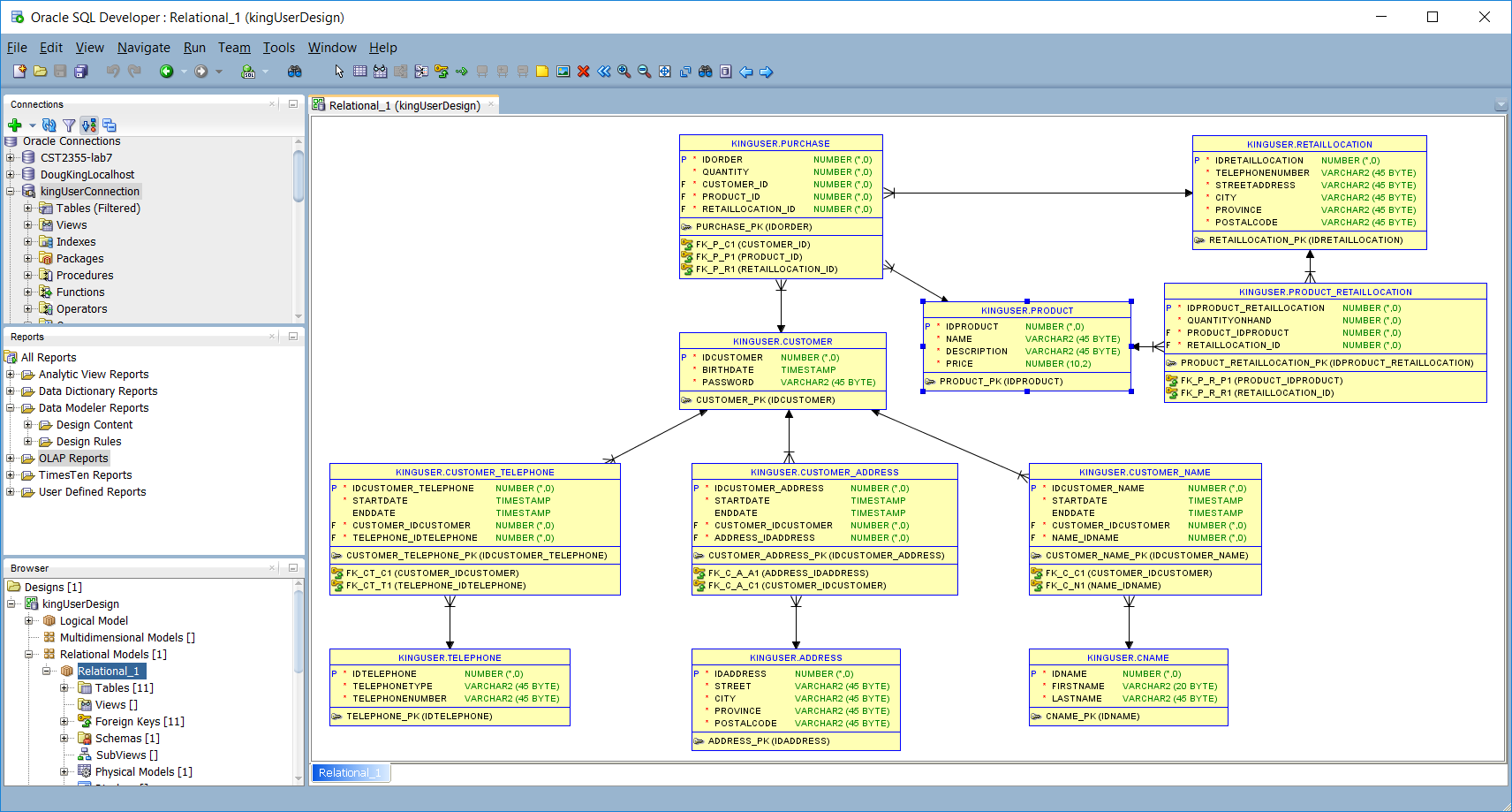


* 1. In SQL Developer, use the menu bar to select the View > Data Modeler > Browser. You will then see a list of designs. Select your new ‘Untitled’ design and save it as *yourname*UserDesign.
     1. Provide a screen shot showing the Data Modeler browser and your newly saved design.

Diagram

Description automatically generated

* + 1. Select your relational model in the browser and use the right-click menu to ‘Show” it. Here is mine:



* + 1. Fix the layout of your reverse-engineered diagram so it will all fit on one page with no crossing lines. Provide a screenshot of your relational model here:

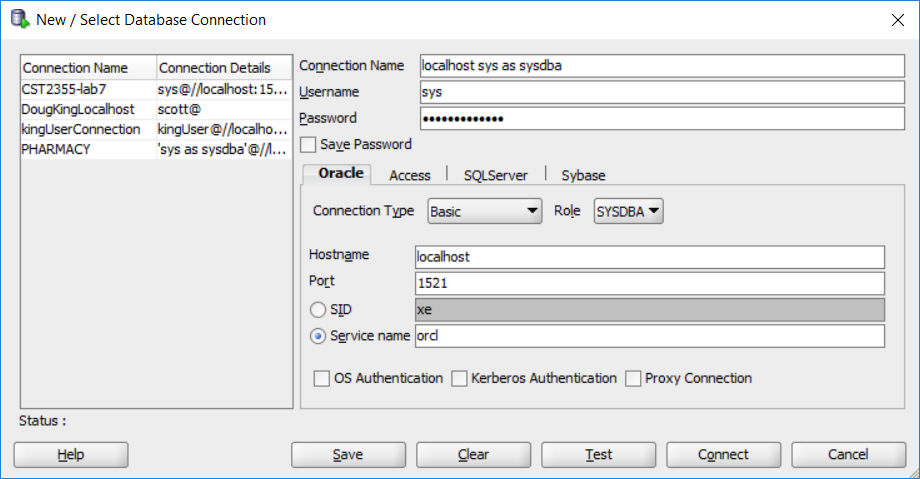
Diagram

Description automatically generated

* + 1. Select your relational model and then using the right-click menu, select ‘Engineer to Logical Model’. Provide a screenshot of your logical model here

Diagram

Description automatically generated

1. **Save your work.**
2. **Use SQL Developer** to make the following changes to the database.
   1. Create a new connection “localhost sys as sysdba” to the database for administrative tasks 
   2. Connect using the connection with your **sys** password that you set in lab 7 - '*yourFirstName*sOracle123' (e.g., dougOracle123)
   3. Once connected, create a new user using SQL developer under the “Other Users” folder. The user should:
      1. Be named testUser2 with password testUser2Password
      2. Have the applicationUser role
      3. Have the default tablespace ‘CST2355’.
      4. Show your work below:

Table

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

* 1. Provide a screenshot showing your SQL statements to GRANT the required privileges to the applicationUser role so that **testUser** (and testUser2…) will be able to insert and update records in the CUSTOMER, CUSTOMER\_NAME, and CNAME tables.

Graphical user interface, text, application

Description automatically generated

1. Create CHECK constraints for **all** of the tables where timestamps are used to indicate the beginning and ending times for the data relationship (e.g., association between customer and their name.
   * 1. Use row-level check constraints to check that if the enddate is not NULL, that the value of the enddate timestamp is after the startdate.
     2. Show your work below for **one** of the CHECK constraints.

Graphical user interface, text, application, email

Description automatically generated

* + 1. Show the results of executing an insert in SQL Developer that does not conform to the check constraint:

Text, application

Description automatically generated

1. Connect as *yourname*User and create a table called ‘UNDER18\_CUSTOMERS’ with the following fields: (choose appropriate field types)

IDUNDER18: number for use as a surrogate key

IDCUSTOMER: to identify the customer record just inserted or updated,

OPERATION: a string indicating the type of update (‘Insert’, ‘Update’),

TS: timestamp of insert/update (sysdate),

UPDATED\_BY: the database username for the connection making the update.

Provide a screenshot showing the table structure:

Text

Description automatically generatedTable

Description automatically generated

* 1. Create the sequence “UNDER18\_SEQ” starting at 1 and incrementing by 1 for use in populating the UNDER18\_CUSTOMERS table. Show your work below.

Graphical user interface, text

Description automatically generated

* 1. Using the connection as ‘sys as sysdba’, modify the permissions on testUser to GRANT inserts into the UNDER18\_CUSTOMERS table. Show your work beow.

A picture containing shape

Description automatically generated

1. Now, you will create a trigger to run after INSERTS and UPDATES on the customers table to log customers with birthdates showing that they are not yet 18 years of age. Firstly, review the “Tutorial: Creating a Trigger that Logs Table Changes” example at  
   [https://docs.oracle.com/database/121/TDDDG/tdddg\_triggers.htm#TDDDG50000](https://docs.oracle.com/database/121/TDDDG/tdddg_triggers.htm)
   1. Here are the details for your trigger:
      1. Use the construct “trunc(months\_between(sysdate, birthdate)/12)” to determine the age in years. If they are currently under 18, put an entry in the “UNDER18\_CUSTOMERS” table.
      2. Use your “UNDER18\_SEQ” sequence to populate the IDUNDER18 field.
      3. Use the customers IDCUSTOMER to populate the IDCUSTOMER field.
      4. Use the current sysdate to populate the TS field.
      5. Use the currently connected “USER” (e.g., ‘kingUser’) to populate the UPDATED\_BY field.
   2. Provide a screenshot below showing your trigger code:

Graphical user interface, text, application, email

Description automatically generated

* 1. Provide the following screenshots showing successfully connecting as **testUser** and performing inserts of a customer. (Just insert into the customer table… not any other tables too.)
     1. Provide a screenshot below for a customer over 18

Graphical user interface

Description automatically generated with medium confidence Graphical user interface, text, application, email

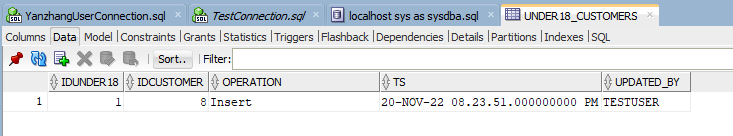
Description automatically generated Graphical user interface, text, application, Word

Description automatically generated

* + 1. Provide screenshots below for a customer under 18

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text

Description automatically generated 

1. **Now you will use the following steps** to create a function that inserts a new customer record including the associated name. The function will need to update three tables; the CUSTOMER, CNAME, and CUSTOMER\_NAME tables. The function then returns the numeric value of the IDCUSTOMER for the new customer table entry:

fn\_insertCustomer (birthdate, password, firstname, lastname)

* 1. Create three sequences: CUSTOMER\_SEQ, and CNAME\_SEQ, and CN\_SEQ that increment by 1. (Have each start 1 past the current maximum in the existing data in the table – hard code the starting value)

Text

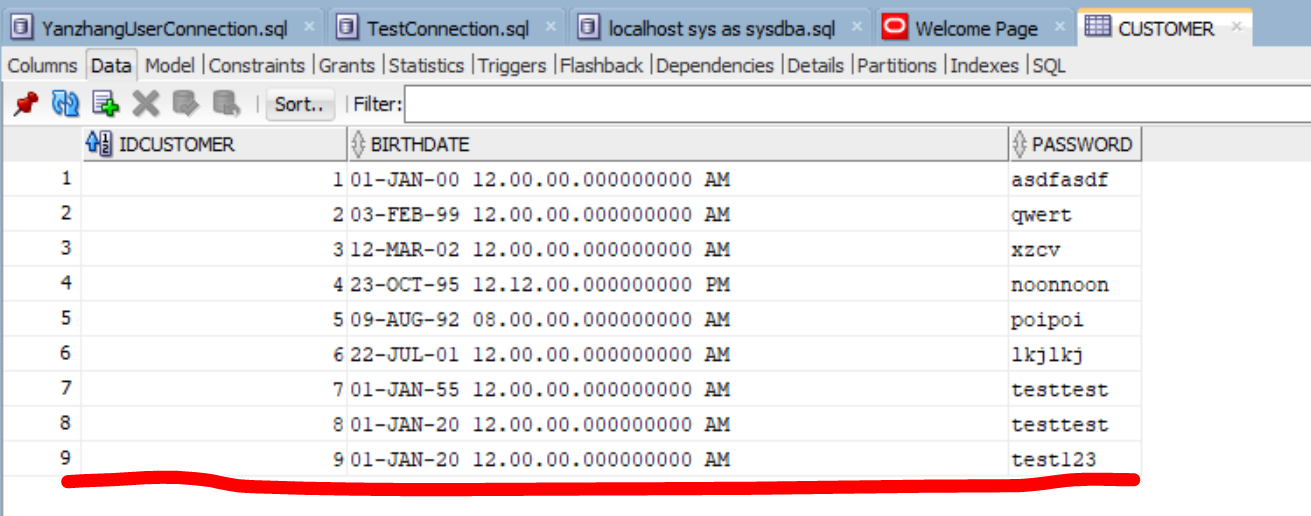
Description automatically generated with medium confidence

* + 1. The function should use the CUSTOMER\_SEQ to generate the IDCUSTOMER value.
    2. The function should use the CNAME\_SEQ to generate the key value for the CNAME table (even if both the firstname and lastname values are NULL).
    3. The function should use the CN\_SEQ to generate the key value for the CUSTOMER\_NAME table.
  1. The value of the STARTDATE field in the CUSTOMER\_NAME table should be the current SYSDATE.
  2. The value of the ENDDATE field in the CUSTOMER\_NAME table should be NULL to indicate that this is the currently valid name.
  3. Provide screenshots showing the successful running of your function.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated  Graphical user interface, application, Word

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

1. Use SQL Developer to create a VIEW that contains the IDCUSTOMER, birthdate, password, firstname, and current lastname.
   1. Name the view ‘CUSTOMER\_VIEW’.
   2. Identify the current name by looking for the entry where the ENDDATE is NULL.

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application

Description automatically generated

1. Once the view has been created, create an INSTEAD OF trigger on the CUSTOMER\_VIEW to allow updates to the customer’s name, while maintaining the history of previous names; by following these criteria:
   * 1. The trigger should be invoked for INSERTs or UPDATEs.
     2. The trigger should update the CUSTOMER\_NAME table entry that matches the IDCUSTOMER and has the ENDDATE as NULL, to have the current SYSDATE as the ENDDATE.
     3. The trigger should use the CNAME\_SEQ sequence and insert a new record into the CNAME table.
     4. The trigger should use the CN\_SEQ sequence to insert a new record into the CUSTOMER\_NAME table. The new record should link the new CNAME record to the existing CUSTOMER record; using the current SYSDATE as the STARTDATE, and also setting ENDATE to the value NULL to indicate that the new name is the currently valid name.
     5. Provide screenshot(s) to show the successful “insert” of a new name using the VIEW.

Text

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Graphical user interface, application, Word

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

1. Once you have embedded all of your screenshots, submit the file in Brightspace and you’re done!