

## CST2355 – Database Systems      Oracle SQL Project-Step 1

Student Name:	Changhong Li
Student ID:	041071022
Student email:	li000811@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. In this lab you will be using the Oracle (Version 19c) database management system, and Oracle SQL Developer (which gets installed as part of the 19c enterprise installation). The first step is to get the software installed.
  - 1.1. Download Oracle 19c Enterprise Edition.
    - 1.1.1. Please note that you should download the .zip file from Oracle at:  
<https://www.oracle.com/ca-en/database/technologies/oracle19c-windows-downloads.html>This is the current stable long-term release for Oracle Enterprise.
  - 1.1.2. Create an Oracle installation folder, using a folder name that is short and does not contain blanks. (e.g., C:\oracle19cinstall) Then, unzip the compressed file **into the Oracle installation folder**.
- 1.2. Open a “cmd” window as administrator (i.e., **Select “Run as administrator”**). Then, navigate in the cmd window to the folder (under your installation folder in step 2) where the setup.exe file has been unzipped. Then, run the following command.
  - 1.2.1.1. On Windows 10: **make sure you are running the command as administrator** and use the following command line. You need to have changed the working directory to the directory where the setup.exe program was unzipped. (Don’t just click on it in Windows Explorer....)

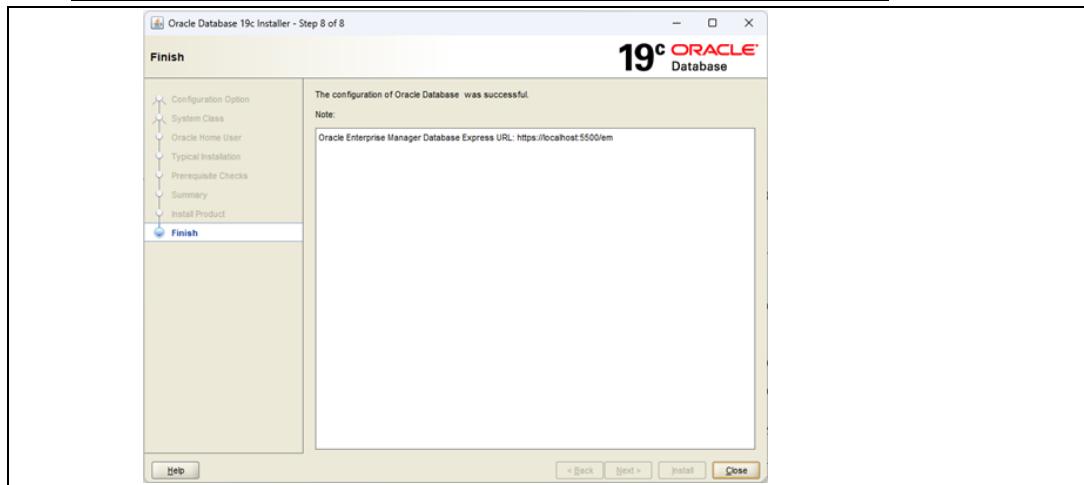
```
setup -ignorePrereq -J"-Doracle.install.db.validate.supportedOSCheck=false"
```

- 1.2.2. The installation program is a Java executable and is notoriously tricky. Sometimes it will pop-up a window underneath the main window; leaving you waiting forever because you did not realize that the program is waiting for a response.
  - 1.2.3. Respond appropriately to the installation customization queries. The critical steps are highlighted – they are mandatory for this course.
- 1.3. Step 1: Configure Security Updates:
- 1.3.1. Email option – leave blank, does not apply.
  - 1.3.2. **Deselect the update notification option.** You do NOT want to receive security updates.
  - 1.3.3. Press 'Next'
  - 1.3.4. Ignore the warning, press 'Yes'.
- 1.4. Step 2: Installation Option:
- 1.4.1. Select 'Create and configure a database', then press 'Next'!
- 1.5. Step 3: System Class:
- 1.5.1. **Select 'Desktop Class', then 'Next'!**
- 1.6. Step4: Home User Selection
- 1.6.1. Select **'Use Built-in Windows Account**, then press 'Next'!
  - 1.6.2. Ignore the security warning and proceed.
- 1.7. Step 5: Typical Installation Configuration:
- 1.7.1. Accept defaults EXCEPT for the following:
    - 1.7.1.1. Global DB name: change to '**orcl**'
    - 1.7.1.2. **CAREFUL: Make sure to uncheck the 'create as container database' checkbox**
  - 1.7.1.3. Admin (SYS) password: '*yourFirstNamesOracle123*' (e.g., *dougOracle123*)
    - 1.7.1.3.1. If you receive a warning('INS-30011') that your password strength is poor, ignore it
    - 1.7.1.3.2. 'Next' / 'Yes' to continue
- 1.8. Step 6: Prerequisite Check is performed. This may take some time. Sometimes the installer will pop-up a window underneath a visible window (so that you cannot see the pop-up) so some manual intervention may be required.
- 1.9. Step 7: Summary of response is displayed. 'Save Response File' and 'Install'.
- 1.10. Step 8: Install Product message is displayed. If you receive a Windows firewall message, allow access for Java
- 1.10.1. Several components are installed.
  - 1.10.2. In the Password Management screen, change the password for user SYSTEM to : '*yourFirstNamesystem*' (e.g., *dougsystem*). You may unlock and change passwords for some other accounts at the same time (or just leave them defaulted for now. The website [© 2022 Algonquin College of Applied Arts and Technology](https://www.orafaq.com/wiki>List_of_default_database_users</a></li></ol></div><div data-bbox=)

has a list of default Oracle usernames and default passwords.

- 1.10.3. 'OK'
- 1.10.4. Ignore any additional password standards warning message. 'Continue? Yes' / 'OK'
- 1.10.5. Windows 10: If Services for Microsoft Transaction Server cannot start – that is OK. Click Next and Yes when asked are you sure you want to continue.
- 1.11. Step 9 Installation was Successful message is displayed. **Take your screen shot before closing the window and paste it below** Then, select 'Close'

**1.11.1. Paste your screen shot showing successful installation below.**



- 1.12. Confirm you are able to **connect** to the database. (**Open a cmd window using ‘run as administrator’ and run the ‘sqlplus / as sysdba’ command line.**) Once connected, **run the command ‘show con\_name;’ which will show the connection name for your default instance. Here is mine:**

```

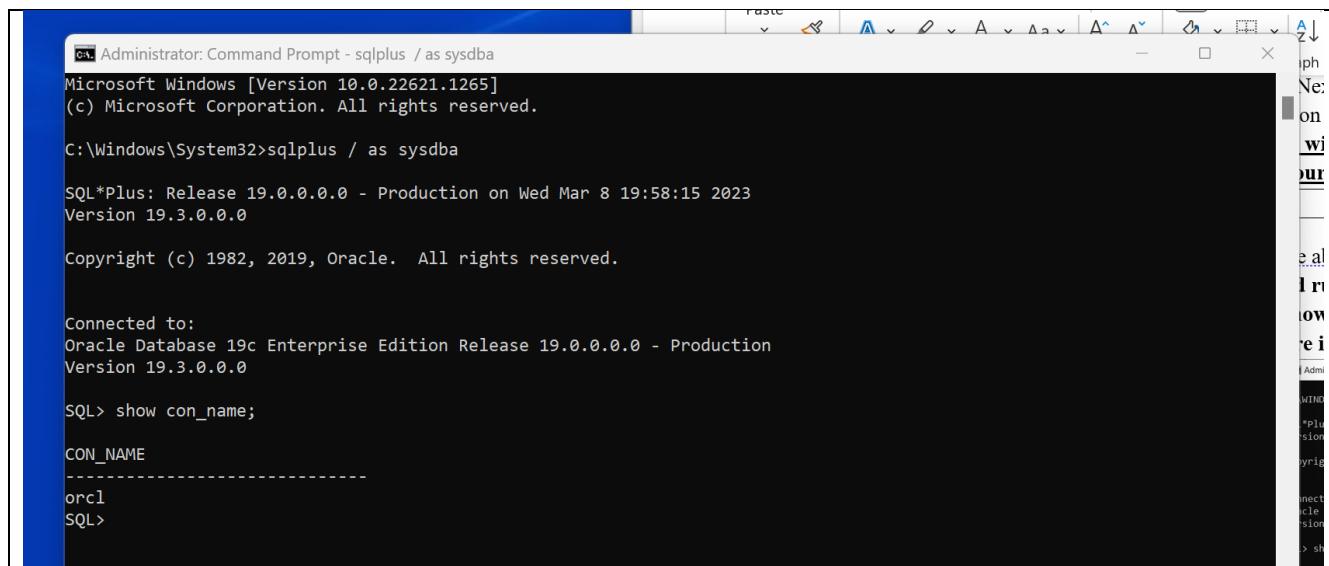
Administrator: Command Prompt - sqlplus / as sysdba
C:\WINDOWS\system32>sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0 - Production on Mon Nov 7 10:13:49 2022
Version 19.3.0.0.0
copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> show con_name;

CON_NAME
-----
orcl18276
SQL>
    
```

- 1.13. **Paste your session showing a successful connection below.**



```
Administrator: Command Prompt - sqlplus / as sysdba
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Mar 8 19:58:15 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> show con_name;

CON_NAME
-----
orcl
SQL>
```

2. Now that Oracle is installed, we will use **sqlplus** to create the database and populate some test data using the scripts supplied in Brightspace for this lab.

- 2.1. Copy the **lab7-create.sql** and **lab7-cleanup.sql** scripts to a convenient folder on your PC.
- 2.2. Open a cmd window ('run as administrator') and navigate to the folder where your sql scripts are stored. (Using the cd command....)
- 2.3. Inside the command window, startup sql plus as follows: 'sqlplus / as sysdba'
- 2.4. This will again connect you as sysadmin without needing a password.
- 2.5. Once connected, issue the following command inside sqlplus

`@lab7-create.sql`

2.5.1. This should execute the script from the current folder, and create the database. **Please note: you should expect some error messages during the script; there are some data insertion errors that you will need to clean-up**, and some other changes to be made later in the lab. For now, just ensure that the script is found and that all is OK except for the errors caused by the failed inserts.

2.5.1.1. Provide a screenshot of the initial run of the script.

```
Administrator: Command Prompt - sqlplus / as sysdba
INSERT INTO purchase VALUES (5,3,4,4,1)
*
ERROR at line 1:
ORA-02291: integrity constraint (KINGUSER.FK_P_R1) violated - parent key not
found

INSERT INTO purchase VALUES (6,2,2,3,3)
*
ERROR at line 1:
ORA-00001: unique constraint (KINGUSER.SYS_C007516) violated

INSERT INTO purchase VALUES (7,5,3,2,2)
*
ERROR at line 1:
ORA-02291: integrity constraint (KINGUSER.FK_P_R1) violated - parent key not
found

INSERT INTO purchase VALUES (8,6,2,1,4)
*
ERROR at line 1:
ORA-00001: unique constraint (KINGUSER.SYS_C007516) violated

INSERT INTO purchase VALUES (9,4,4,2,3)
*
ERROR at line 1:
ORA-00001: unique constraint (KINGUSER.SYS_C007516) violated

INSERT INTO purchase VALUES (10,3,2,3,2)
*
ERROR at line 1:
ORA-02291: integrity constraint (KINGUSER.FK_P_R1) violated - parent key not
found

INSERT INTO purchase VALUES (11,7,1,2,1)
*
ERROR at line 1:
ORA-02291: integrity constraint (KINGUSER.FK_P_R1) violated - parent key not
found

Commit complete.

SQL>
```

2.5.2. Use sqlplus to run the script **lab7-cleanup.sql** using sqlplus. This will undo the effects of the first script.

```
Administrator: Command Prompt - sqlplus / as sysdba
found

INSERT INTO purchase VALUES (11,7,1,2,1)
*
ERROR at line 1:
ORA-02291: integrity constraint (KINGUSER.FK_P_R1) violated - parent key not
found

Commit complete.

SQL> @lab7-cleanup.sql

Procedure dropped.

Procedure dropped.

DROP USER kingUser CASCADE
*
ERROR at line 1:
ORA-01031: insufficient privileges

DROP USER testUser
*
ERROR at line 1:
ORA-01031: insufficient privileges

DROP ROLE applicationAdmin
*
ERROR at line 1:
ORA-01031: insufficient privileges

DROP ROLE applicationUser
*
ERROR at line 1:
ORA-01924: role 'APPLICATIONUSER' not granted or does not exist

DROP TABLESPACE CST2355 INCLUDING CONTENTS AND DATAFILES
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
```

3. Now you will fix the creation script. Open the lab7-create.sql script using a text editor and CAREFULLY make the following changes:
  - 3.1. Change the username for the user that is assigned the application administrator role to 'yournameUser' and change the password for that user.
  - 3.2. Modify the insert statements to fix the invalid telephone numbers and postal codes.

- 3.3. Update the trigger on the retaillocation table to include both a check on postalcode and a check on the telephone number. **Provide a screenshot of your updated trigger below:**

```
CREATE OR REPLACE TRIGGER retaillocation_check
BEFORE INSERT OR UPDATE
ON retaillocation
FOR EACH ROW
DECLARE
    var_pc      varchar(45) := :NEW.postalcode;
    var_tel     varchar(45) := :NEW.telephonenumber;
BEGIN
    sp_checkInvalidPostalCode (var_pc);
    sp_checkInvalidTelephone (var_tel);
END;
/
```

4. Use a texteditor to edit the **lab7-cleanup.sql** script to be able to cleanup based on your new *yournameUser* username. Provide a screenshot showing the updated contents of the script below

```
-- Cleanup for lab7

DROP PROCEDURE liUser.sp_checkInvalidPostalCode;
DROP PROCEDURE liUser.sp_checkInvalidTelephone;
DROP USER liUser CASCADE;
DROP USER testUser;
DROP ROLE applicationAdmin;
DROP ROLE applicationUser;
DROP TABLESPACE CST2355 INCLUDING CONTENTS AND DATAFILES;

-- End of File
```

5. Create a screenshot showing you running your version of the lab7-create.sql script and including (some of) the output. You now have the storage, user, roles, tables, triggers, and stored procedures all in place. Provide the screenshot below:

```
Administrator: Command Prompt - sqlplus
1 row created.

1 row created.

1 row created.

1 row created.

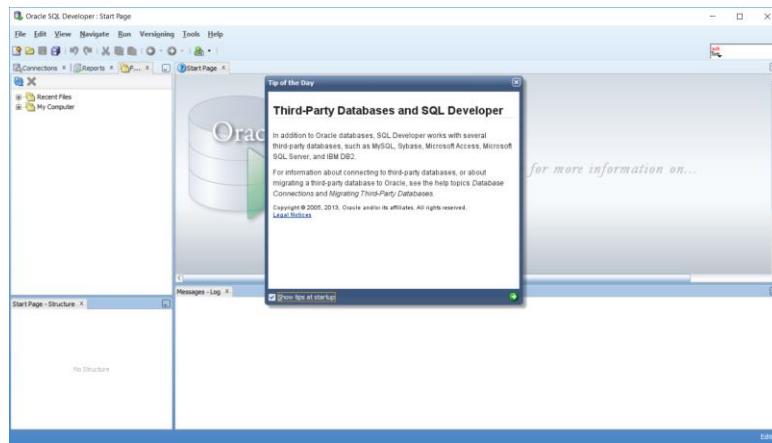
Table created.

1 row created.

Commit complete.

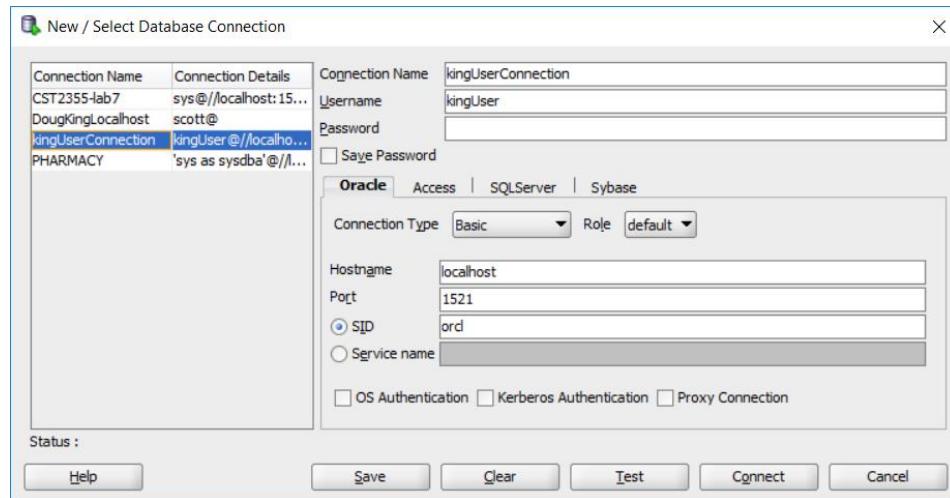
SQL>
```

6. Download and install the Oracle SQL Developer (Including JDK) from the Oracle downloads site at:  
[Oracle SQL Developer Downloads](#)  
and follow the installation instructions. Then, use the Windows Start menu to open the ‘Oracle SQL Developer’ tool in the Oracle tools folder. You should see something like this:

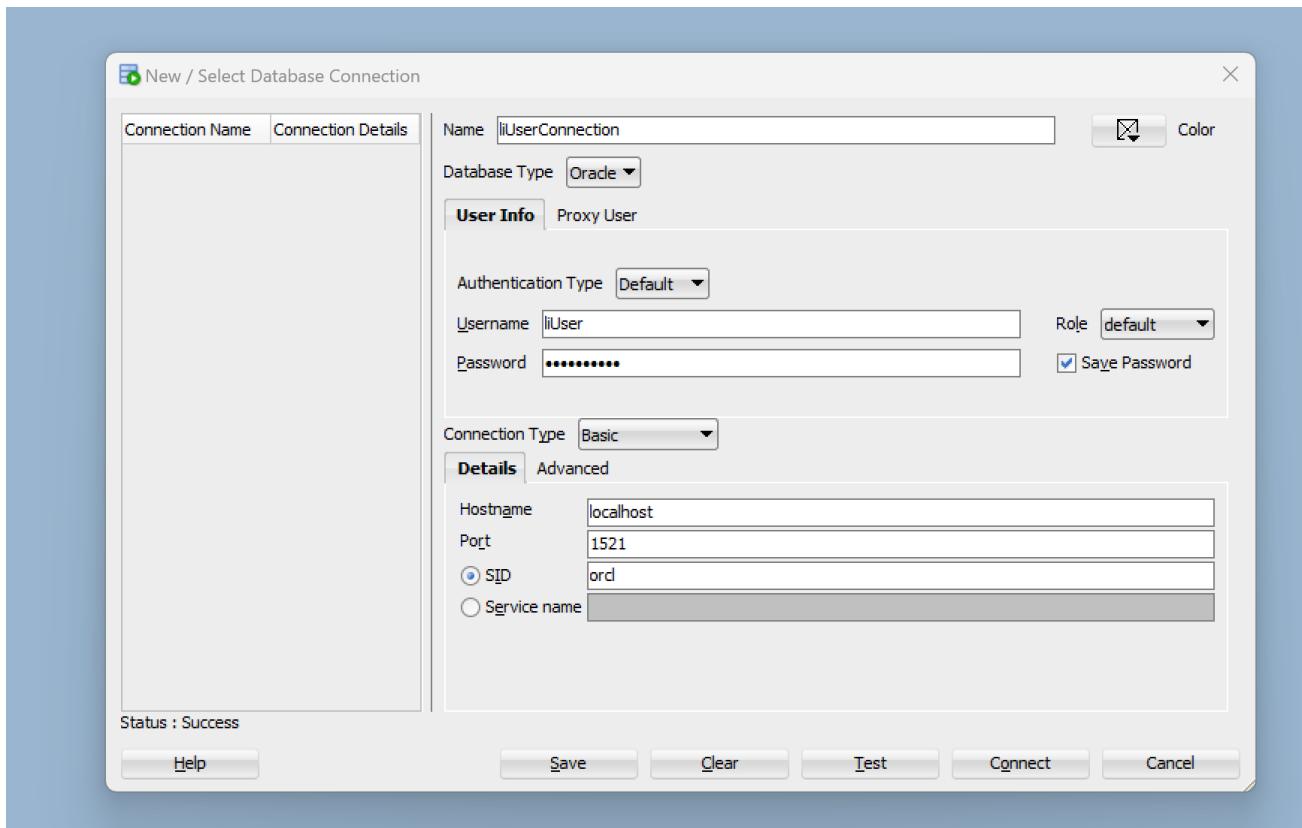


6.1.1. You need to create a new connection called *yournameUserConnection* that connects to the local Oracle database instance as *yournameUser* using the password you used in the creation script.

6.1.1.1. Click on the new connection button. You should see something like this:



6.1.1.2. You likely have less database type options, but you should fill in the form: select ‘Oracle’ and change the SID to the instance name that you had captured earlier (‘show con\_name’ using sqlplus). Mine is ‘orcl’. You should test your connection and provide the confirmation screen below.



6.1.2. Once the connection is in place, open it (providing the password for your new user) to view the contents. Provide a screenshot showing the set of tables.

Oracle SQL Developer : Table LIUSER.ADDRESS@liUserConnection

Connections

- liUserConnection
  - Tables (Filtered)
    - ADDRESS
    - CNAME
    - CUSTOMER
    - CUSTOMER\_ADDRESS
    - CUSTOMER\_NAME
    - CUSTOMER\_TELEPHONE
    - PRODUCT
    - PRODUCT\_RETALLOCATION
    - PURCHASE
    - RETALLOCATION
    - TELEPHONE
  - Views
  - Indexes
  - Packages
  - Procedures
  - Functions
  - Operators
  - Queues
  - Queues Tables
  - Triggers
  - Types
  - Sequences

Reports

- All Reports
  - Analytic View Reports
  - Data Dictionary Reports
  - Data Modeler Reports
  - OLAP Reports
  - TimesTen Reports
  - User Defined Reports

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDADDRESS	NUMBER(38,0)	No	(null)	1 (null)	
2 STREET	VARCHAR2(45 BYTE)	No	(null)	2 (null)	
3 CITY	VARCHAR2(45 BYTE)	No	(null)	3 (null)	
4 PROVINCE	VARCHAR2(45 BYTE)	No	(null)	4 (null)	
5 POSTALCODE	VARCHAR2(45 BYTE)	No	(null)	5 (null)	

6.1.3. Use the query editor in SQL Developer (or sqlplus) to provide examples of trying to insert an invalid address, an invalid telephone, and a retaillocation with both problems. Provide your screenshots below.

#### 6.1.3.1. Invalid address (postalcode):

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query is being run:

```
INSERT INTO address VALUES (3,'1 Lewis Street','Ottawa','Ontario','K2s1E6');
```

In the bottom right corner of the workspace, there is an error message:

```
Error starting at line : 1 in command -
INSERT INTO address VALUES (3,'1 Lewis Street','Ottawa','Ontario','K2s1E6')
Error report -
ORA-20000: The postal code must be 6 characters in length of the form: "A9A9A9"!
ORA-06512: at "LIUSER.SP_CHECKINVALIDPOSTALCODE", line 6
ORA-06512: at "LIUSER.ADDRESS_CHECK", line 4
ORA-04088: error during execution of trigger 'LIUSER.ADDRESS_CHECK'
```

### 6.1.3.2. Invalid telephone:

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar shows a connection named 'liUserConnection'. The 'Worksheet' pane contains the following SQL code:

```

INSERT INTO address VALUES (3,'1 Lewis Street','Ottawa','Ontario','K2S1E6');

INSERT INTO telephone VALUES (1,'home','45 987-6543');
    
```

The second insert statement is highlighted in yellow. The 'Script Output' pane at the bottom displays the error message:

```

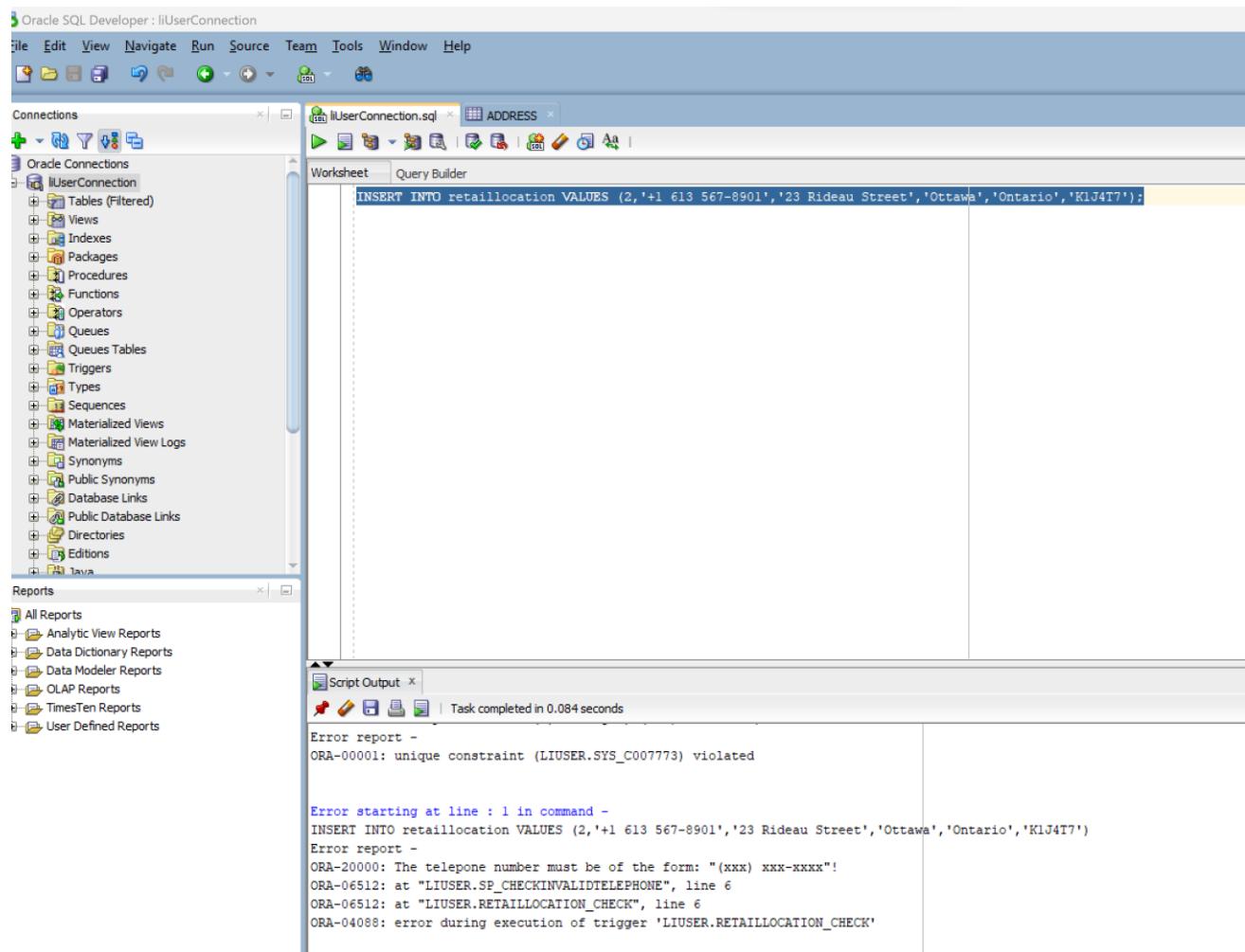
Task completed in 0.034 seconds

Error report -
ORA-00001: unique constraint (LIUSER.SYS_C007773) violated

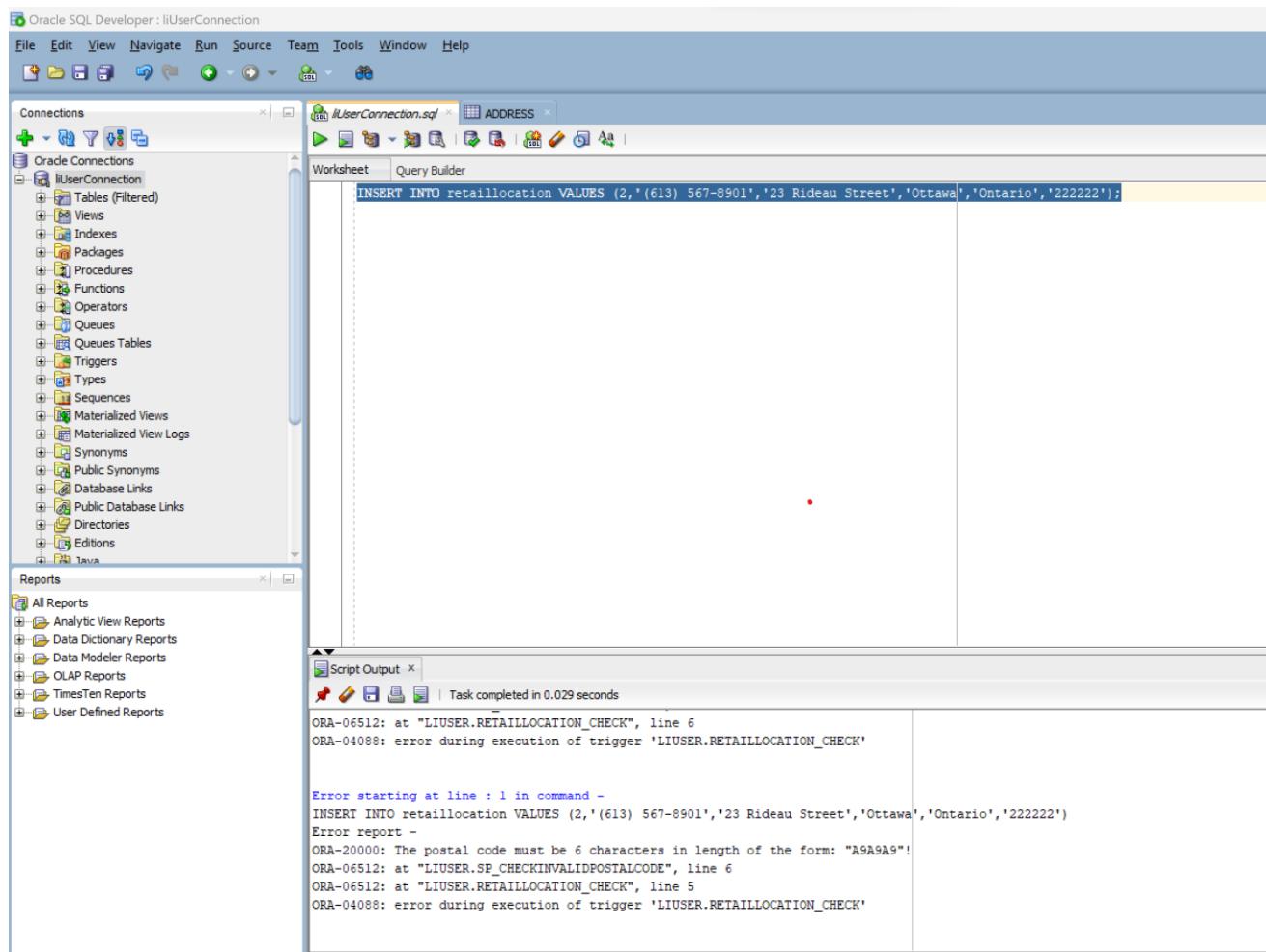
Error starting at line : 3 in command -
INSERT INTO telephone VALUES (1,'home','45 987-6543')
Error report -
ORA-20000: The telephone number must be of the form: "(xxx) xxx-xxxx"!
ORA-06512: at "LIUSER.SP_CHECKINVALIDTELEPHONE", line 6
ORA-06512: at "LIUSER.TELEPHONE_CHECK", line 4
ORA-04088: error during execution of trigger 'LIUSER.TELEPHONE_CHECK'
    
```

### 6.1.3.3. Invalid retaillocation: (3 possibilities)

#### 6.1.3.3.1. Invalid telephone field:



#### 6.1.3.3.2. Invalid postalcode field:



**6.1.3.3.3.** Both postalcode and telephone are invalid (don't try to fix the trigger... it is ok if only 1 error is shown):

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the following SQL command:

```
INSERT INTO retaillocation VALUES (2, '613567', '23 Rideau Street', 'Ottawa', 'Ontario', '222222');
```

The 'Script Output' tab displays the execution results and errors:

```

Task completed in 0.033 seconds

ORA-06512: at "LIUSER.RETAILLOCATION_CHECK", line 5
ORA-04088: error during execution of trigger 'LIUSER.RETAILLOCATION_CHECK'

Error starting at line : 1 in command -
INSERT INTO retaillocation VALUES (2, '613567', '23 Rideau Street', 'Ottawa', 'Ontario', '222222')
Error report -
ORA-20000: The postal code must be 6 characters in length of the form: "A9A9A9"!
ORA-06512: at "LIUSER.SP_CHECKINVALIDPOSTALCODE", line 6
ORA-06512: at "LIUSER.RETAILLOCATION_CHECK", line 5
ORA-04088: error during execution of trigger 'LIUSER.RETAILLOCATION_CHECK'
```

(Note: that is true, only one error is reported)

- Once you have embedded all of your screenshots, submit the file in Brightspace and you're done!

## CST2355 – Database Systems      Oracle SQL Project-Step 2

Student Name: Changhong Li  
Student ID: 041071022  
Student email: li000811@algonquinlive.com

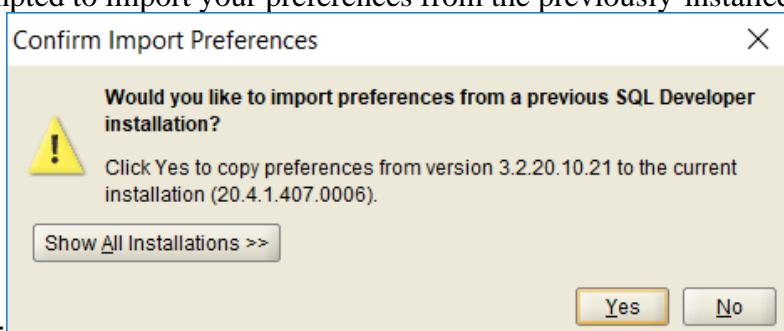
## Hand-in:

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.

**Part A – Data Modeling using SQL Developer**

## Activities (Steps):

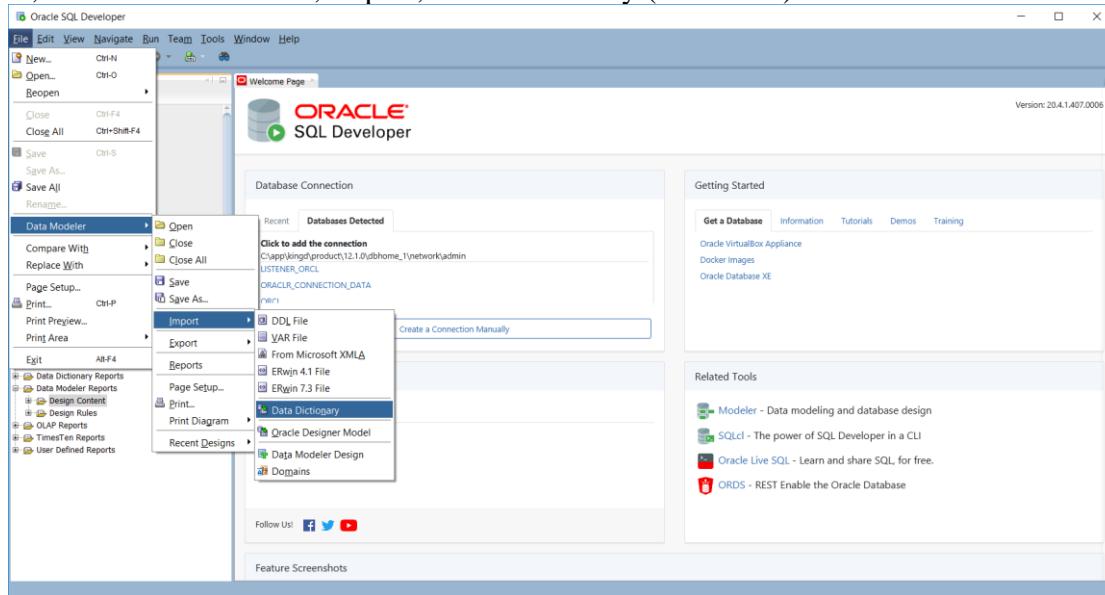
1. First thing we need to do is update SQL Developer to the latest version:
  - 1.1. Go to: <https://www.oracle.com/tools/downloads/sqldev-downloads.html> and download SQL Developer version 20.4 (or later). (If it is newer than 20.4, your installation/configuration steps may be slightly different.)
    - 1.1. When it is downloaded to your PC, extract all the contents of the .zip file.
    - 1.2. 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.
    - 1.3. Run the application.
  - 1.4. You will likely be prompted to import your preferences from the previously-installed SQL



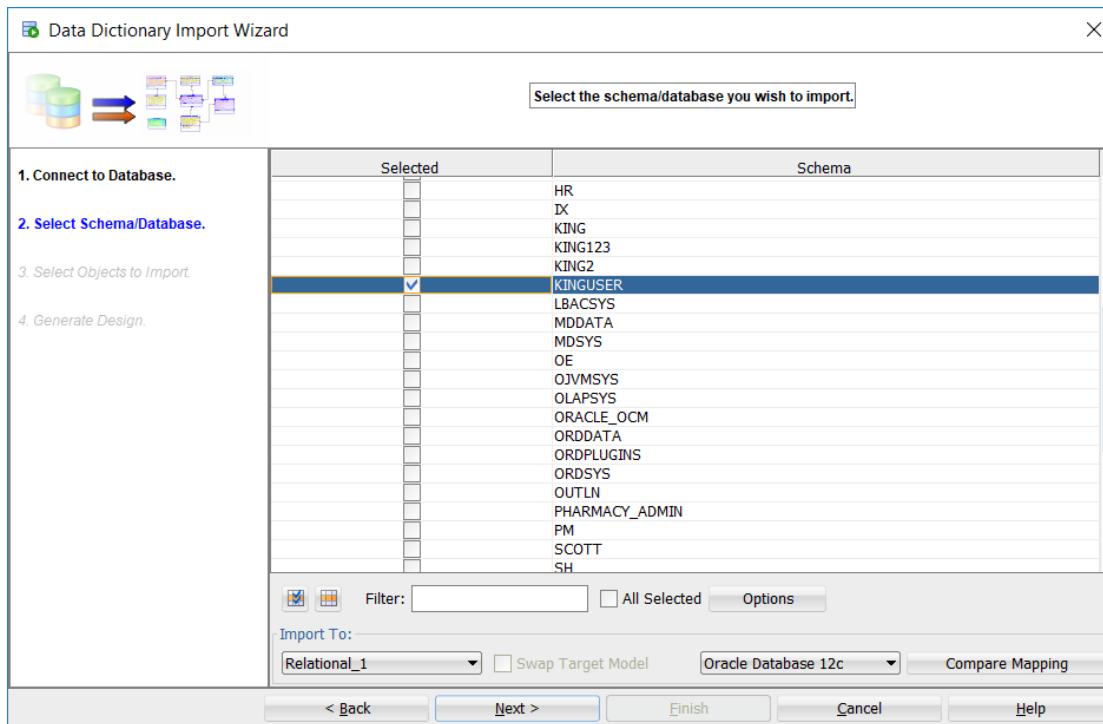
- 1.5. developer version:

- 1.6. If so, import the preferences.

2. Select the '*yournameUserConnection*' from the list of connections, and then from the File menu, select: Data Modeler, Import, Data Dictionary (see below):



- 2.1. And click on Next. You will be prompted for your password, and then given a list of schematae from which to select.

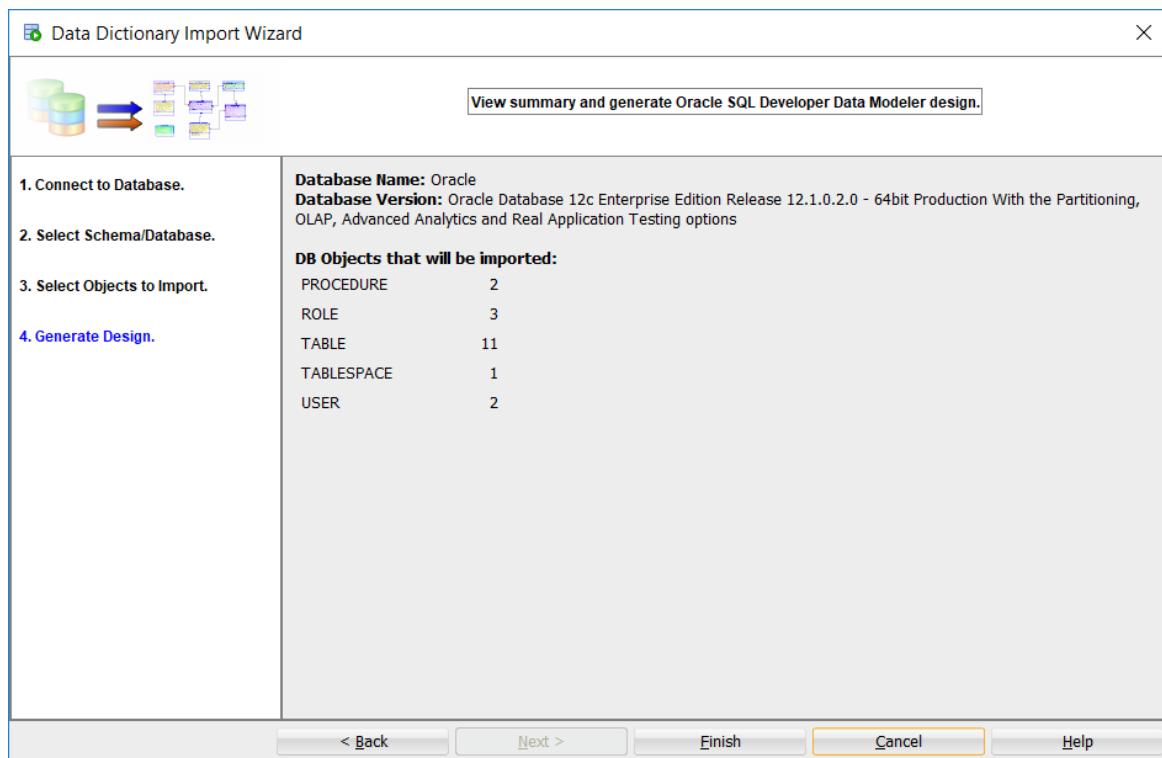


- 2.2. Select the *yournameUser* schema and then click Next.

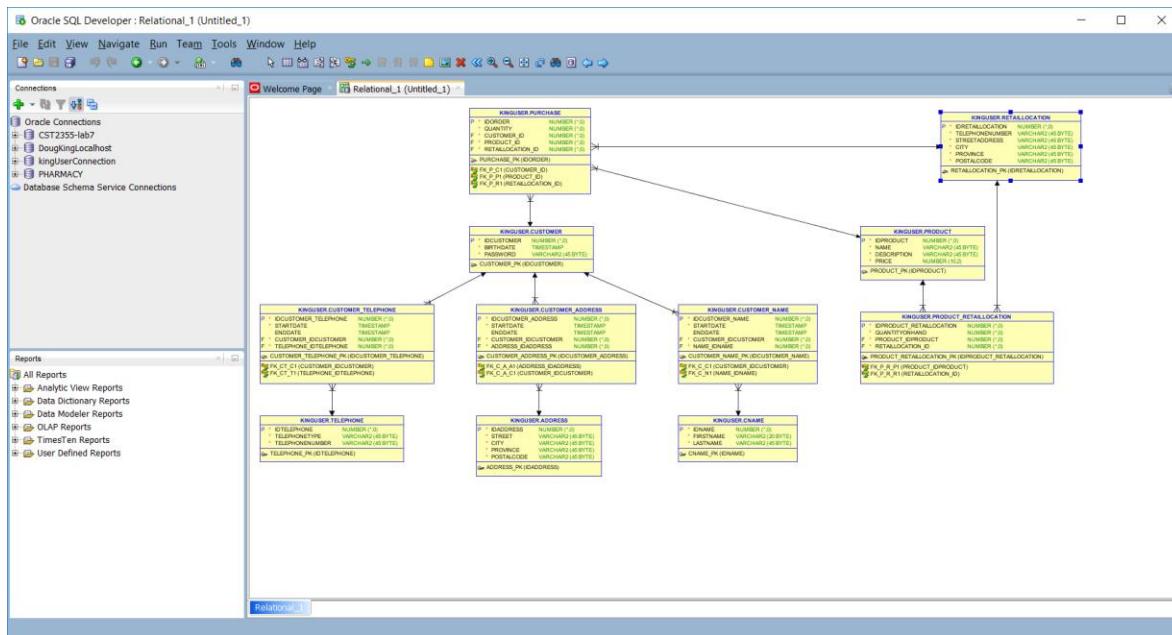
2.2.1. You will be prompted for what items you would like to import.

- 2.2.1.1. Select all of the 11 Tables
- 2.2.1.2. Select *yournameUser* and MDSYS from Users
- 2.2.1.3. Select all of the 4 Roles from the original database
- 2.2.1.4. Select CST2355 from Tablespaces
- 2.2.1.5. Select both of the 2 Stored Procedures.

2.3. Then on the confirmation screen, select ‘Finish’

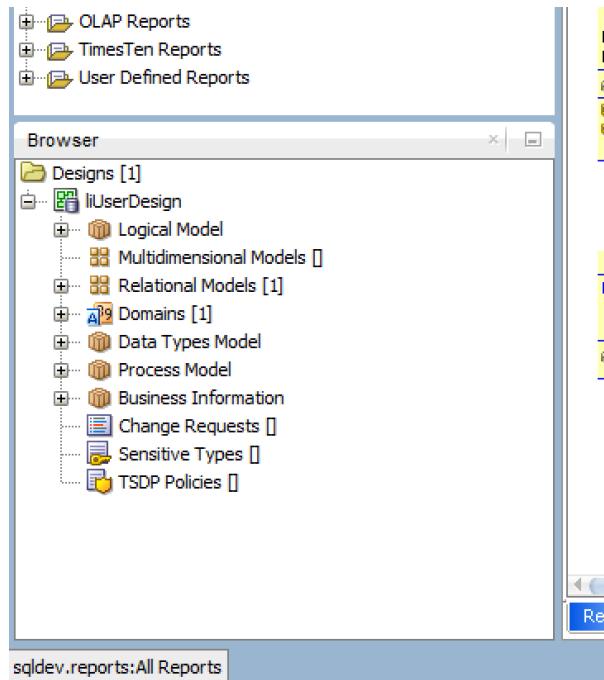


2.4. When completed, you will see the E-R diagram. Here is mine:

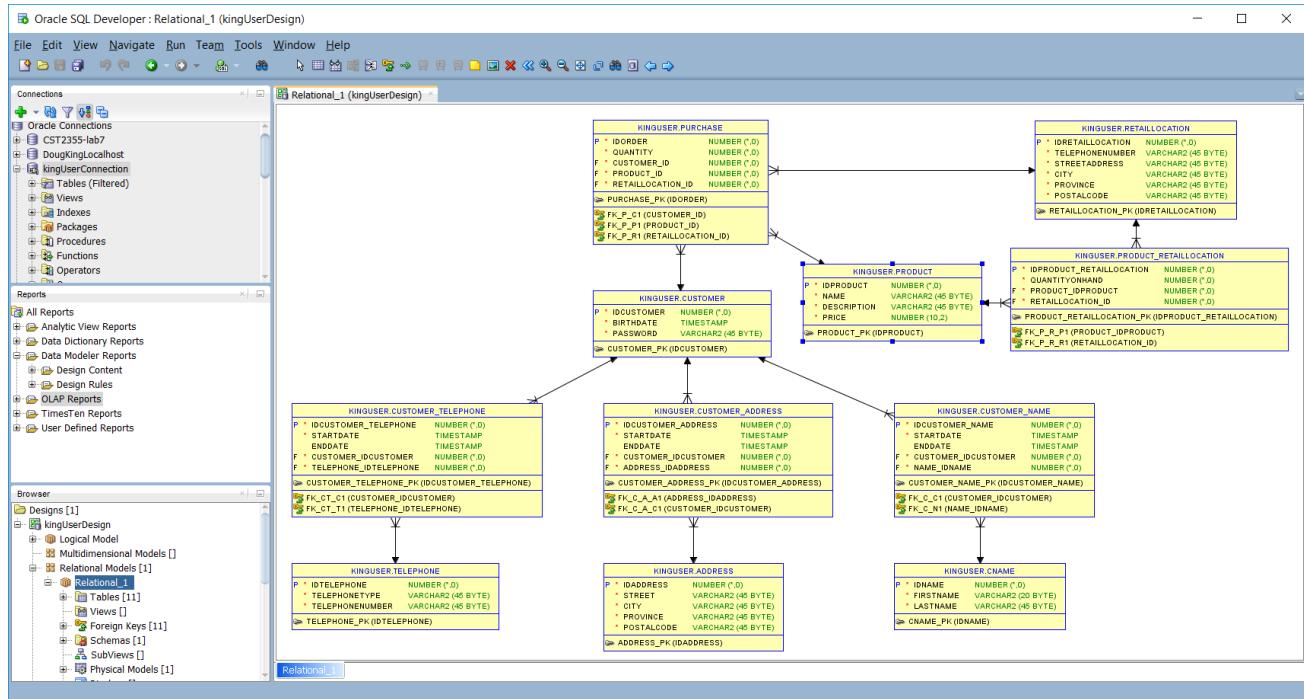


2.5. In SQL Developer, use the menu bar to select the View > Data Modeler > Browser. You will then see a list of designs. Select your new ‘Untitled’ design and save it as *yournameUserDesign*.

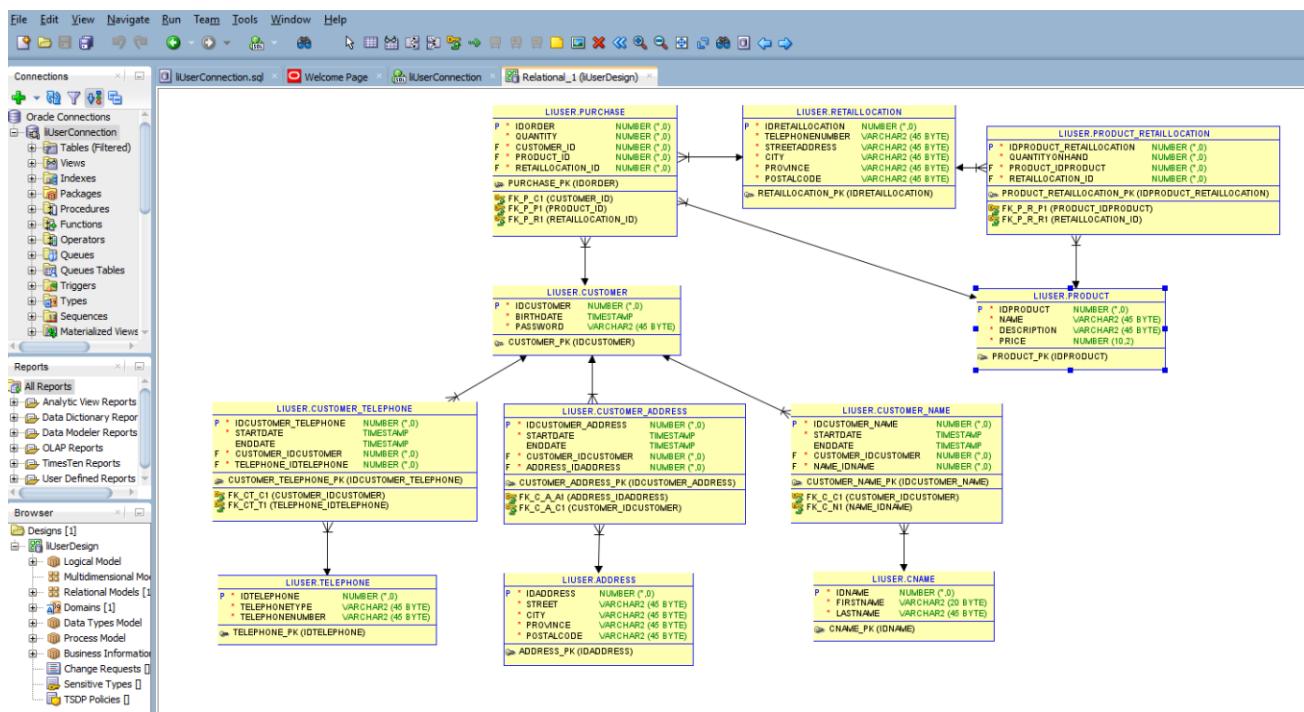
2.5.1. Provide a screen shot showing the Data Modeler browser and your newly saved design.



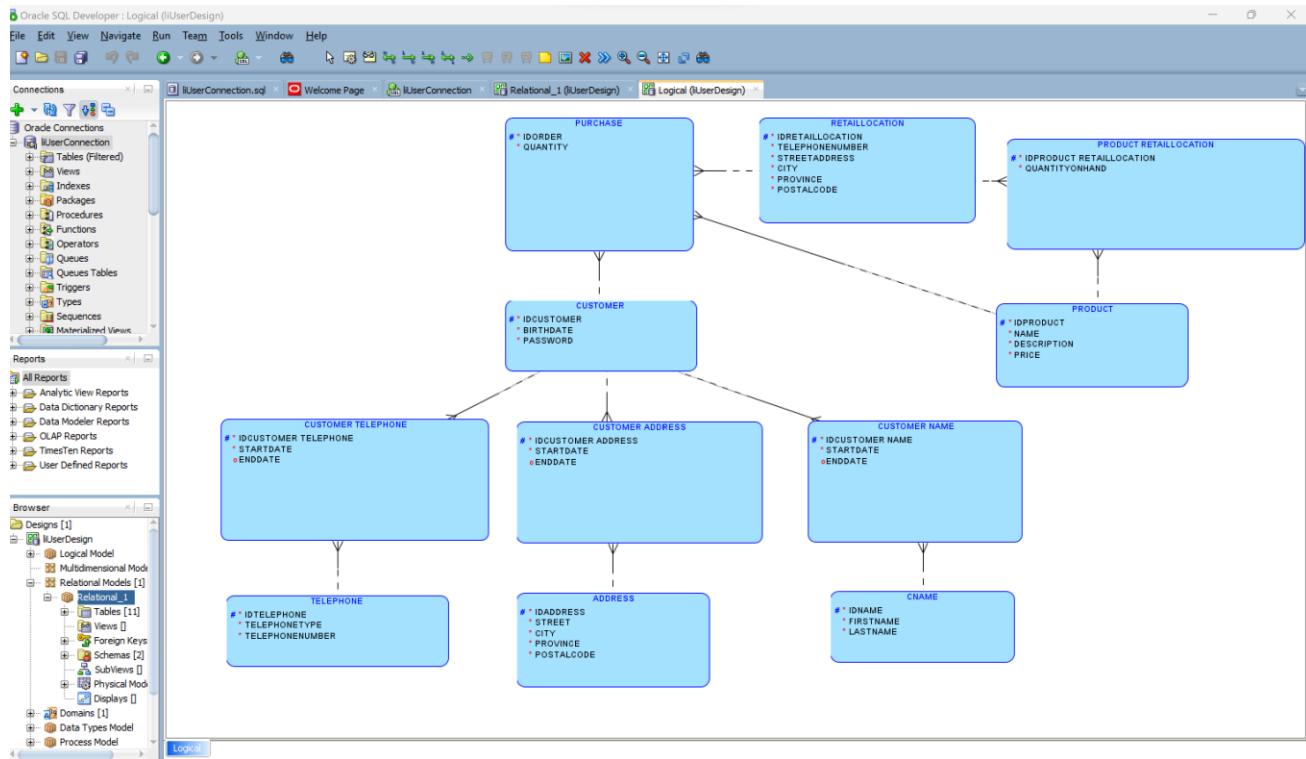
### 2.5.2. Select your relational model in the browser and use the right-click menu to ‘Show’ it. Here is mine:



### 2.5.3. 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:



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



### 3. Save your work.

## Part B – Database Trigger

**Background:** In this part of the lab, we will create a trigger on **CUSTOMER** table (created in lab7 through lab7-create.sql) script. The objective of the trigger is to check the age of every new customer record being inserted or updated into customer table and if the customer’s age is below 18, then a record will also be inserted into an additional **UNDER18\_CUSTOMERS** table.

4. Connect as *yournameUser* (created in lab7) 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:

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows an Oracle connection named 'liUserConnection' with several tables listed under 'Tables (Filtered)'. The 'liUserConnectionLab8.sql' script in the 'SQL Worksheet' tab contains the following DDL:

```

DROP TABLE UNDER18_CUSTOMERS;

--Create the 'UNDER18_CUSTOMERS' table
CREATE TABLE UNDER18_CUSTOMERS (
    IDUNDER18 NUMBER NOT NULL,
    IDCUSTOMER NUMBER NOT NULL,
    OPERATION VARCHAR2(50) NOT NULL,
    TS TIMESTAMP DEFAULT SYSDATE NOT NULL,
    UPDATED_BY VARCHAR2(50) NOT NULL
);

ALTER TABLE liUser.UNDER18_CUSTOMERS
ADD CONSTRAINT under18_customers_pk PRIMARY KEY(IDUNDER18);

ALTER TABLE liUser.UNDER18_CUSTOMERS
ADD CONSTRAINT fk_customer_comp FOREIGN KEY(IDCUSTOMER)
REFERENCES liUser.CUSTOMER(IDCUSTOMER);

DESCRIBE UNDER18_CUSTOMERS;
    
```

The 'Script Output' tab shows the results of running the script:

```

Table LIUSER.UNDER18_CUSTOMERS altered.

Table LIUSER.UNDER18_CUSTOMERS altered.

Name Null? Type
-----
IDUNDER18 NOT NULL NUMBER
IDCUSTOMER NOT NULL NUMBER
OPERATION NOT NULL VARCHAR2(50)
TS NOT NULL TIMESTAMP(6)
UPDATED_BY NOT NULL VARCHAR2(50)
    
```

The screenshot shows the Oracle SQL Developer interface. The 'liUserConnectionLab8.sql' script in the 'SQL Worksheet' tab has been modified to include the 'Columns' tab for the UNDER18\_CUSTOMERS table.

**Columns Tab Data:**

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IDUNDER18	NUMBER	No	(null)	1	(null)
2 IDCUSTOMER	NUMBER	No	(null)	2	(null)
3 OPERATION	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4 TS	TIMESTAMP(6)	No	SYSDATE	4	(null)
5 UPDATED_BY	VARCHAR2(50 BYTE)	No	(null)	5	(null)

**Edit Table Dialog:**

Schema: LIUSER  
 Name: UNDER18\_CUSTOMERS  
 Table Type: Normal

**Columns Tab Data (Detailed View):**

PK	Name	Data Type	Size	Not Null	Default	Comment
1	IDUNDER18	NUMBER		✓		
	IDCUSTOMER	NUMBER		✓		
	OPERATION	VARCHAR2	50	✓		
	TS	TIMESTAMP		✓	SYSDATE	
	UPDATED_BY	VARCHAR2	50	✓		

- 4.1. Create the sequence “UNDER18\_SEQ” starting at 1 and incrementing by 1 for use in populating the UNDER18\_CUSTOMERS → IDUNDER18 column. Show your work below.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection named "liUserConnection". In the center, the "SEQUENCES" node under "liUserConnection" is expanded, showing three sequences: "ISEQ\$\$\_75305", "UNDER18\_CUSTOMERS\_SEQ", and "UNDER18\_SEQ". The "UNDER18\_SEQ" sequence is selected. On the right, the "Details" tab of the sequence editor is open, displaying the following properties:

Name	Value
1 CREATED	23-03-30
2 LAST_DDL_TIME	23-03-30
3 SEQUENCE_OWNER	LIUSER
4 SEQUENCE_NAME	UNDER18_SEQ
5 MIN_VALUE	1
6 MAX_VALUE	1000000
7 INCREMENT_BY	1
8 CYCLE_FLAG	N
9 ORDER_FLAG	Y
10 CACHE_SIZE	0
11 LAST_NUMBER	1
12 SCALE_FLAG	N
13 EXTEND_FLAG	N
14 SHARDED_FLAG	N
15 SESSION_FLAG	N
16 KEEP_VALUE	N
17 DUPLICATED	N
18 SHARDED	N

Below the sequence editor, the SQL tab displays the DDL code for creating the sequence:

```
CREATE SEQUENCE "LIUSER"."UNDER18_SEQ" MINVALUE 1 MAXVALUE 1000000 INCREMENT BY 1 START WITH 1 NOCACHE ORDER NOCYCLE NOKEEP NOSCALE GLOBAL ;
```

5. 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#TDDDG50000)

5.1. Here are the details for your trigger:

- 5.1.1. Use the function “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.
- 5.1.2. Use your “UNDER18\_SEQ” sequence to populate the IDUNDER18 field.
- 5.1.3. Use the customer’s IDCUSTOMER to populate the IDCUSTOMER field.
- 5.1.4. Use the current sysdate to populate the TS field.
- 5.1.5. Use the currently connected “USER” (e.g., ‘kingUser’) to populate the UPDATED\_BY field.

5.2. Provide a screenshot below showing your trigger code:

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema structure under the 'Connections' tab, specifically for the 'RETAILLOCATION' schema. It lists various objects like Views, Indexes, Procedures, Functions, Operators, Queues, Queues Tables, and Triggers. The 'Triggers' section contains four triggers: ADDRESS\_CHECK, RETAILLOCATION\_CHECK, TELEPHONE\_CHECK, and TRIGGER\_CHECK\_UNDER18. The right pane shows the 'Code' tab with the PL/SQL code for the TRIGGER\_CHECK\_UNDER18 trigger. The code uses an AFTER INSERT OR UPDATE trigger on the CUSTOMER table. It declares variables for age and action, and uses the trunc function to calculate age from sysdate and birthdate. It then checks if it's an insert or update and performs the appropriate action (inserting into UNDER18\_CUSTOMERS). Finally, it inserts a record into UNDER18\_CUSTOMERS with values from UNDER18\_SEQ.NEXTVAL, IDCUSTOMER, ACTION, SYSDATE, and USER.

```

create or replace TRIGGER TRIGGER_CHECK_UNDER18
AFTER INSERT OR UPDATE ON CUSTOMER
FOR EACH ROW
DECLARE
    age NUMBER;
    action UNDER18_CUSTOMERS.OPERATION%TYPE;
BEGIN
    age := trunc(months_between(sysdate, :New.birthdate)/12);

    IF INSERTING THEN
        action := 'insert';
    ELSE
        action := 'update';
    END IF;

    IF age<18 THEN
        INSERT INTO UNDER18_CUSTOMERS (IDUNDER18, IDCUSTOMER, OPERATION, TS, UPDATED_BY)
        VALUES (UNDER18_SEQ.NEXTVAL, :NEW.IDCUSTOMER, action, sysdate, user);
    END IF;
END;
/

```

5.3. Testing the Trigger by inserting and updating records in customer table.

- 5.3.1. Now, insert a new record in the customer table. Ensure that this new customer’s age is below 18. Review the insert statement in ‘Lab7-create’ for sample of insert statement on customer.
- 5.3.2. Update the password of the new customer record inserted above.
- 5.3.3. These Insert & Update on customer should have executed the trigger in the background and populated the under18\_customers tables. Provide a screenshot of records in the

under18\_customers table. (There should be at least two records as a result of above insert and update).

lper : liUserConnection

Navigate Run Source Team Tools Window Help

Worksheet Query Builder

```

INSERT INTO customer (IDCUSTOMER, BIRTHDATE, PASSWORD)
VALUES (9,'2015-09-09','aaaaaa');

INSERT INTO customer (IDCUSTOMER, BIRTHDATE, PASSWORD)
VALUES (10,'2000-10-10','123456');

UPDATE customer SET password = 'password' WHERE IDCUSTOMER = 9;

```

Script Output Task completed in 0.061 seconds

1 row inserted.

1 row inserted.

1 row updated.

Messages - Log

NDER18\_CUSTOMERS@liUserConnection

File Tools Window Help

UNDER18\_CUSTOMERS

IDUNDER18	IDCUSTO...	OPERATION	TS	UPDATED...
1	13	9 insert	23-04-01 21:14:17.000000000	LIUSER
2	14	9 update	23-04-01 21:14:17.000000000	LIUSER

## CST2355 – Database Systems      Oracle SQL Project-Step 3

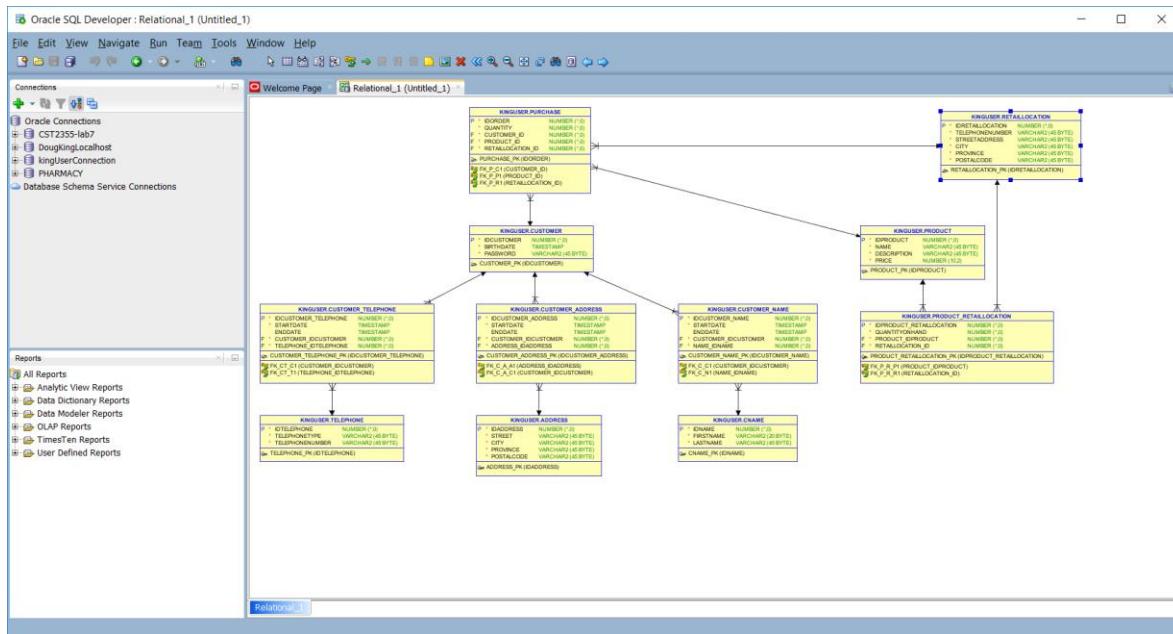
Student Name: Changhong Li  
 Student ID: 041071022  
 Student email: li000811@algonquinlive.com

### 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. We are going to create a package that shares items across a set of stored procedures and functions, based on the model that was used in lab 8: (see below)



- 1.1. First look at the tutorial at: [https://www.tutorialspoint.com/plsql/plsql\\_packages.htm](https://www.tutorialspoint.com/plsql/plsql_packages.htm) It contains an implementation of a package to manage customer records.

1.2. Prepare an sql script called “lab9-sequences.sql” that contains the CREATE SEQUENCE statements to create sequences **that will be used in this lab when inserting new entries in each of the tables in your schema.** Choose appropriate starting values so that the existing data is not in conflict with the new numbers. (e.g., start them all at 100?) Run the script to create the sequences.

1.2.1. Provide a screenshot showing the contents of your script.

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database structure under 'Connections' with 'Sequences' expanded, showing various sequence names like ADDRESS\_SEQ, CNAME\_SEQ, etc. The main workspace is a 'Worksheet' tab containing the following SQL code:

```

Drop sequence address_seq; cname_seq; customer_seq; customer_address_seq;
customer_name_seq; customer_telephone_seq; product_seq;
product_retaillocation_seq; purchase_seq; retaillocation_seq;
telephone_seq; under18_customers_seq;

--Lab09/1.2.1 create sequence statement named lab9-sequences
create sequence address_seq start with 100 increment by 1 nocache nocycle;

create sequence cname_seq start with 100 increment by 1 nocache nocycle;

create sequence customer_seq start with 100 increment by 1 nocache nocycle;

create sequence customer_address_seq start with 100 increment by 1 nocache nocycle;

create sequence customer_name_seq start with 100 increment by 1 nocache nocycle;

create sequence customer_telephone_seq start with 100 increment by 1 nocache nocycle;

create sequence product_seq start with 100 increment by 1 nocache nocycle;

create sequence product_retaillocation_seq start with 100 increment by 1 nocache nocycle;

create sequence purchase_seq start with 100 increment by 1 nocache nocycle;

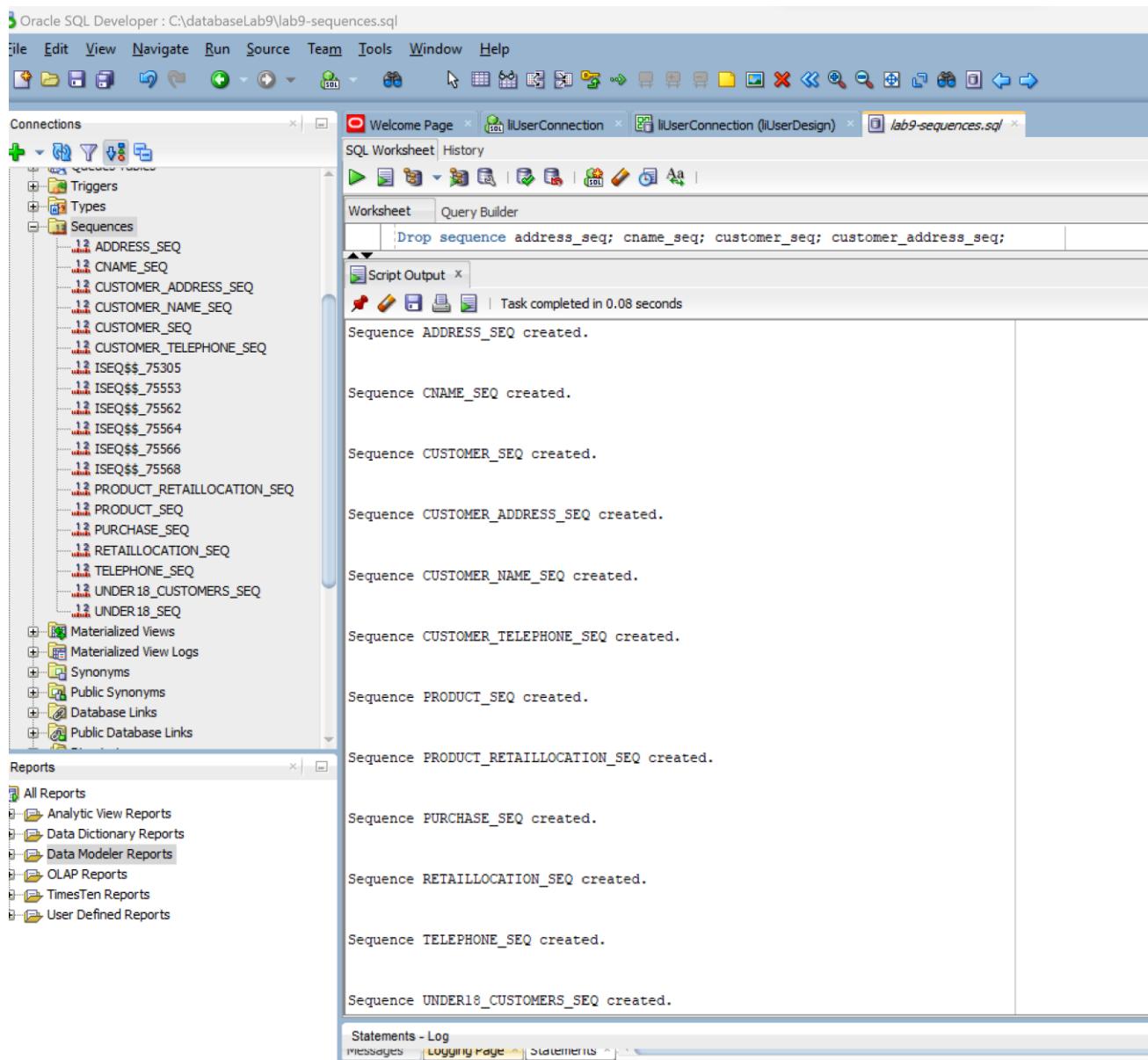
create sequence retaillocation_seq start with 100 increment by 1 nocache nocycle;

create sequence telephone_seq start with 100 increment by 1 nocache nocycle;

create sequence under18_customers_seq start with 100 increment by 1 nocache nocycle;
    
```

The 'Script Output' tab at the bottom shows the message: "Sequence UNDER18\_CUSTOMERS\_SEQ created." The 'Statements - Log' tab at the bottom is empty.

1.2.2. Provide a screenshot showing the successful running of the sequences script.



2. Create a package in the *yourNameUser* schema called *customer\_pkg* with the following specification:
  - 2.1. The package should contain three stored procedures that each use an IDCUSTOMER IN parameter along with the appropriate IN parameters to update the underlying tables.
    - 2.1.1. change\_telephone()
    - 2.1.2. change\_address()
    - 2.1.3. change\_name()
  - 2.2. The package should also contain a function called new\_customer() that returns an INTEGER containing the IDCUSTOMER for a new customer record. The new\_customer() function should have mandatory fields for birthdate and password, and optional fields for each of the data fields in the telephone, address, and name tables.

- 2.3. The package should also contain a function called `get_age()` that returns an INTEGER containing the age in years (rounded down to the nearest integer value), for a given IDCUSTOMER.
- 2.4. Provide a screenshot or screenshots showing the package **specification** (just the package specification – not the entire package body) below:

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Shows various icons for file operations, database connections, and code navigation.
- Tab Bar:** Displays tabs for "Welcome Page", "liUserConnection", "liUserConnection (liUserDesign)", "lab9-sequences.sql" (which is the active tab), and "CNAME".
- Worksheet Tab:** Set to "Worksheet".
- Code Area:** Displays the package specification for "customer\_pkg". The code includes:
  - A `CREATE OR REPLACE PACKAGE` statement for `customer_pkg`.
  - Two procedures: `change_telephone` and `change_address`, both taking `idcustomer` as an IN parameter and three VARCHAR2 parameters (`telephonetype`, `telephonenumber`, `a_street`, `a_city`, `a_province`, `a_postalcode`).
  - A procedure `change_name` taking `idcustomer` and two VARCHAR2 parameters (`firstname`, `lastname`).
  - Two functions: `new_customer` (returning an INTEGER) and `get_age` (returning an INTEGER).
  - The `new_customer` function takes nine parameters: `birthdate`, `password`, `telephonetype`, `telephonenumber`, `a_street`, `a_city`, `a_province`, `a_postalcode`, `firstname`, and `lastname`.
  - The `get_age` function takes `idcustomer` as an IN parameter.
  - The package ends with `END customer_pkg;`
- Script Output:** Shows the message "Task completed in 0.033 seconds".
- Message Bar:** Shows the message "Package CUSTOMER\_PKG compiled".

3. Provide the package body for `customer_pkg` using the following criteria.
- Each procedure should insert a new telephone/address/name as appropriate
  - Each procedure should update the entry in the relationship association table to have the sysdate timestamp as the enddate for the current entry (that is the one with NULL)

enddate gets updated to have enddate as sysdate). If there is no previous related item (i.e., no current entry with a NULL enddate) then this step should get skipped.

3.1.3. Each procedure should insert a new record in the relationship association table that has the startdate as sysdate and a NULL enddate.

3.1.4. The new\_customer() function should create the customer record along with the telephone, address, and name records as required. If all the non-key fields in a telephone, address or name record would be null, then the associated record should not be created.

**[NOTE: The new\_customer function should use the three stored procedures in the package to create the related records.]**

3.1.5. The get\_age() function should return an INTEGER containing the age in years (rounded down to the nearest integer value), for a given IDCUSTOMER.

3.1.6. Provide screenshots for the three stored procedures below:

```

b9-sequences.sql
Team Tools Window Help
File Edit View Insert Object Navigator Database SQL Worksheet History
Welcome Page iUserConnection iUserConnection (iUserDesign) lab9-sequences.sql
Worksheet Query Builder
CREATE OR REPLACE PACKAGE BODY customer_pkg AS
    PROCEDURE change_telephone(idcustomer IN INTEGER, telephonetype IN VARCHAR2, telephonenumber IN VARCHAR2)
    IS v_idtelephone NUMBER;
    BEGIN
        INSERT INTO liuser.telephone (idtelephone, telephonenumber)
        VALUES (telephone_seq.nextval, telephonenumber)
        RETURNING idtelephone INTO v_idtelephone;

        UPDATE liuser.customer_telephone
        SET enddate = sysdate
        WHERE customer_idcustomer = idcustomer AND enddate IS NULL;

        INSERT INTO liuser.customer_telephone (idcustomer_telephone, startdate, enddate, customer_idcustomer, telephone_idtelephone)
        VALUES (customer_telephone_seq.nextval, sysdate, NULL, idcustomer, v_idtelephone);
    END change_telephone;

    PROCEDURE change_address(idcustomer IN INTEGER, a_street IN VARCHAR2, a_city IN VARCHAR2, a_province IN VARCHAR2, a_postalcode IN VARCHAR2)
    IS v_idaddress NUMBER;
    BEGIN
        INSERT INTO liuser.address (idaddress, street, city, province, postalcode)
        VALUES (address_seq.nextval, a_street, a_city, a_province, a_postalcode)
        RETURNING idaddress INTO v_idaddress;

        UPDATE liuser.customer_address
        SET enddate = sysdate
        WHERE customer_idcustomer = idcustomer AND enddate IS NULL;

        INSERT INTO liuser.customer_address (idcustomer_address, startdate, enddate, customer_idcustomer, address_idaddress)
        VALUES (customer_address_seq.nextval, sysdate, NULL, idcustomer, v_idaddress);
    END change_address;

    PROCEDURE change_name(idcustomer IN INTEGER, firstname IN VARCHAR2, lastname IN VARCHAR2)
    IS v_idname NUMBER;
    BEGIN
        INSERT INTO liuser.cname (idname, firstname, lastname)
        VALUES (cname_seq.nextval, firstname, lastname)
        RETURNING idname INTO v_idname;

        UPDATE liuser.customer_name
        SET enddate = sysdate
        WHERE customer_idcustomer = idcustomer
        AND enddate IS NULL;

        INSERT INTO liuser.customer_name (idcustomer_name, startdate, enddate, customer_idcustomer, name_idname)
        VALUES (customer_name_seq.nextval, sysdate, NULL, idcustomer, v_idname);
    END change_name;

```

Script Output: Task completed in 0.041 seconds

Package Body CUSTOMER\_PKG compiled

3.1.7. Provide a screenshot showing the new\_customer function below:

```

source Team Tools Window Help
Welcome Page liUserConnection liUserConnection (liUserDesign) lab9-sequences.sql
Worksheet Query Builder
VALUES (customer_name_seq.nextval, sysdate, NULL, idcustomer, v_lusername);
END change_name;

FUNCTION new_customer(
    birthdate IN DATE,
    password IN VARCHAR2,
    phonetype IN VARCHAR2,
    telephonenumber IN VARCHAR2,
    a_street IN VARCHAR2,
    a_city IN VARCHAR2,
    a_province IN VARCHAR2,
    a_postalcode IN VARCHAR2,
    firstname IN VARCHAR2,
    lastname IN VARCHAR2)
RETURN INTEGER IS
    v_idcustomer NUMBER;
BEGIN
    INSERT INTO liuser.customer (idcustomer, birthdate, password)
    VALUES (customer_seq.nextval, birthdate, password)
    RETURNING idcustomer INTO v_idcustomer;

    change_telephone(v_idcustomer, phonetype, telephonenumber);
    change_address(v_idcustomer, a_street, a_city, a_province, a_postalcode);
    change_name(v_idcustomer, firstname, lastname);

    RETURN v_idcustomer;
END new_customer;

FUNCTION get_age(idcustomer IN INTEGER) RETURN INTEGER IS
    v_age INTEGER;
BEGIN
    SELECT TRUNC(MONTHS_BETWEEN(sysdate, birthdate) / 12) INTO v_age

```

Script Output | Task completed in 0.041 seconds

Package Body CUSTOMER\_PKG compiled

3.1.8. Provide a screenshot showing the get\_age function below:

```

FUNCTION get_age(idcustomer IN INTEGER) RETURN INTEGER IS
    v_age INTEGER;
BEGIN
    SELECT TRUNC(MONTHS_BETWEEN(sysdate, birthdate) / 12) INTO v_age
    FROM liuser.customer
    WHERE idcustomer = idcustomer;

    RETURN v_age;
END get_age;

END customer_pkg;
/

```

Script Output | Task completed in 0.041 seconds

Package Body CUSTOMER\_PKG compiled

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