CST2355 – Database Systems Lab Assignment 9

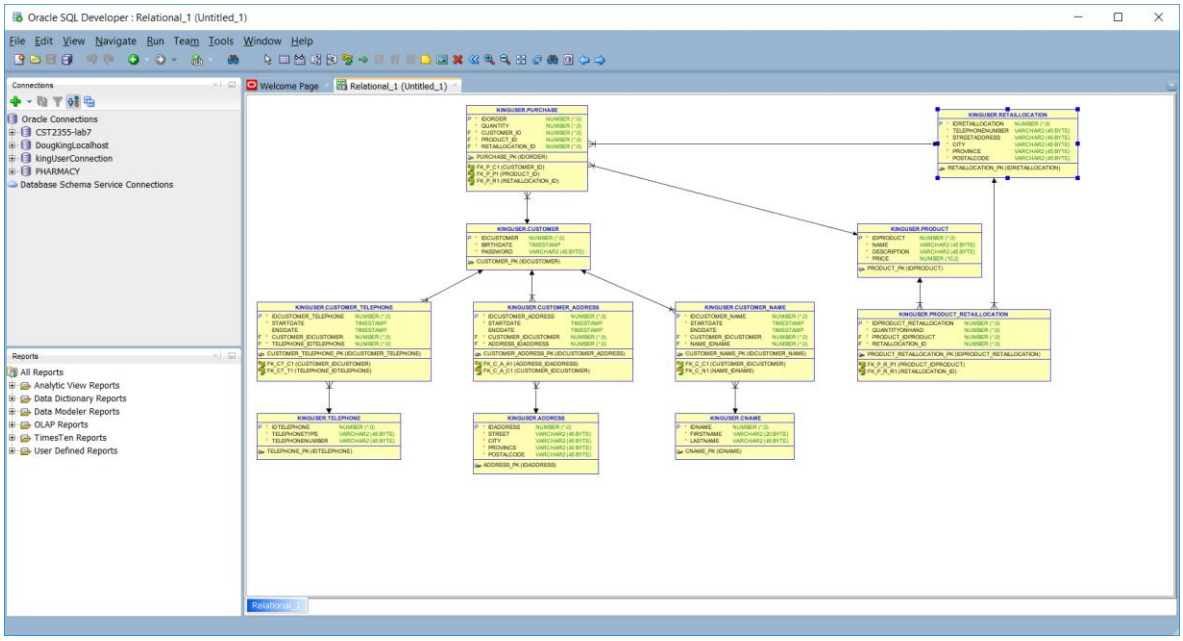
Student Name: Fei Lan
Student ID: 041055048
Student email: lan00012@algonquinlive.com

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)

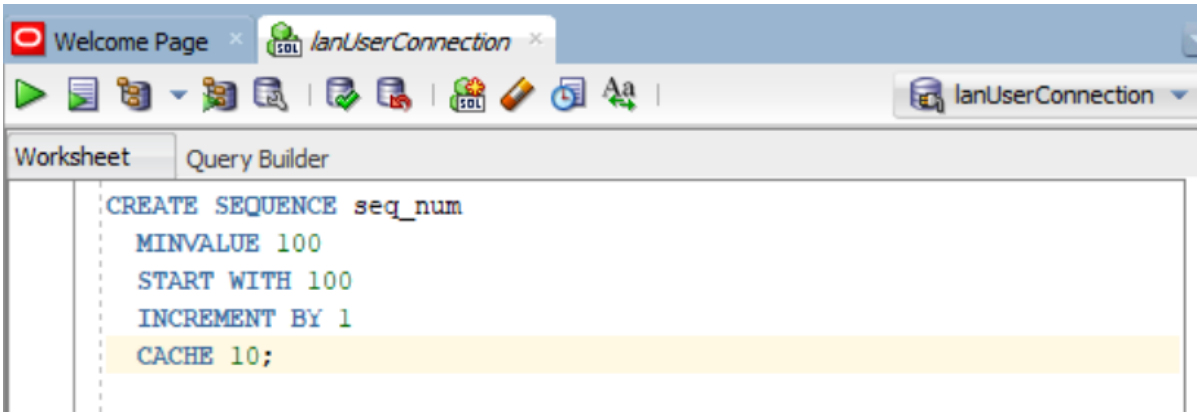


1.1. First look at the tutorial at: https://www.tutorialspoint.com/plsql/plsql_packages.htm It contains an implementation of a package to manage customer records.

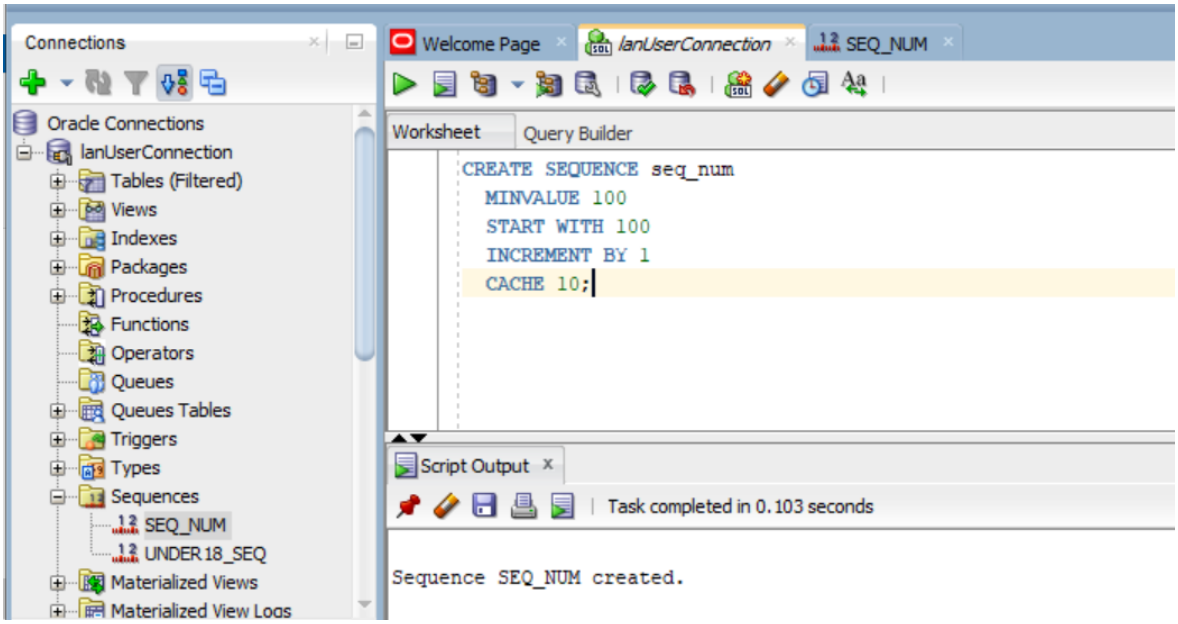
© 2021 Algonquin College of Applied Arts and Technology

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.

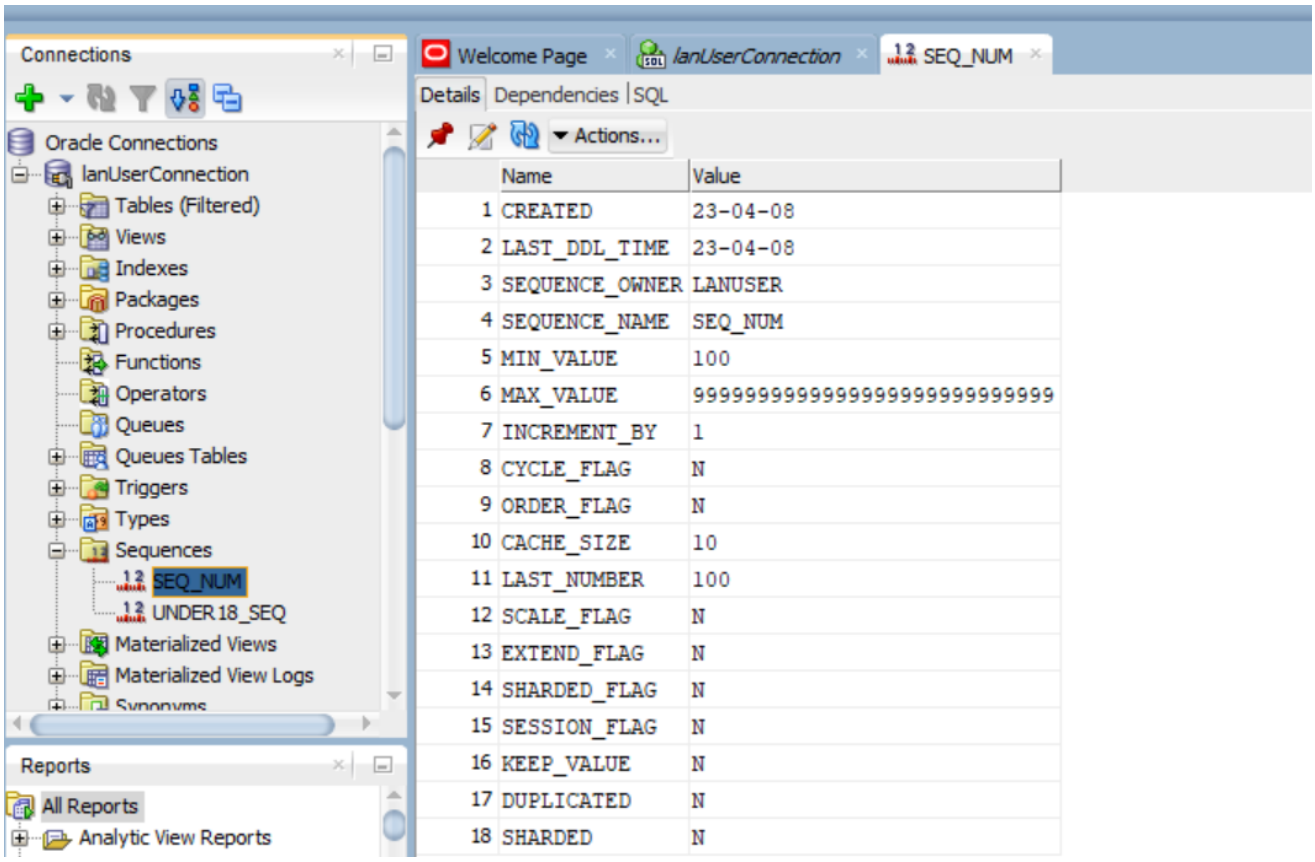
1.2.1. Provide a screenshot showing the contents of your script.



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



© 2021 Algonquin College of Applied Arts and Technology



2. Create a package in the *yourName*User 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()

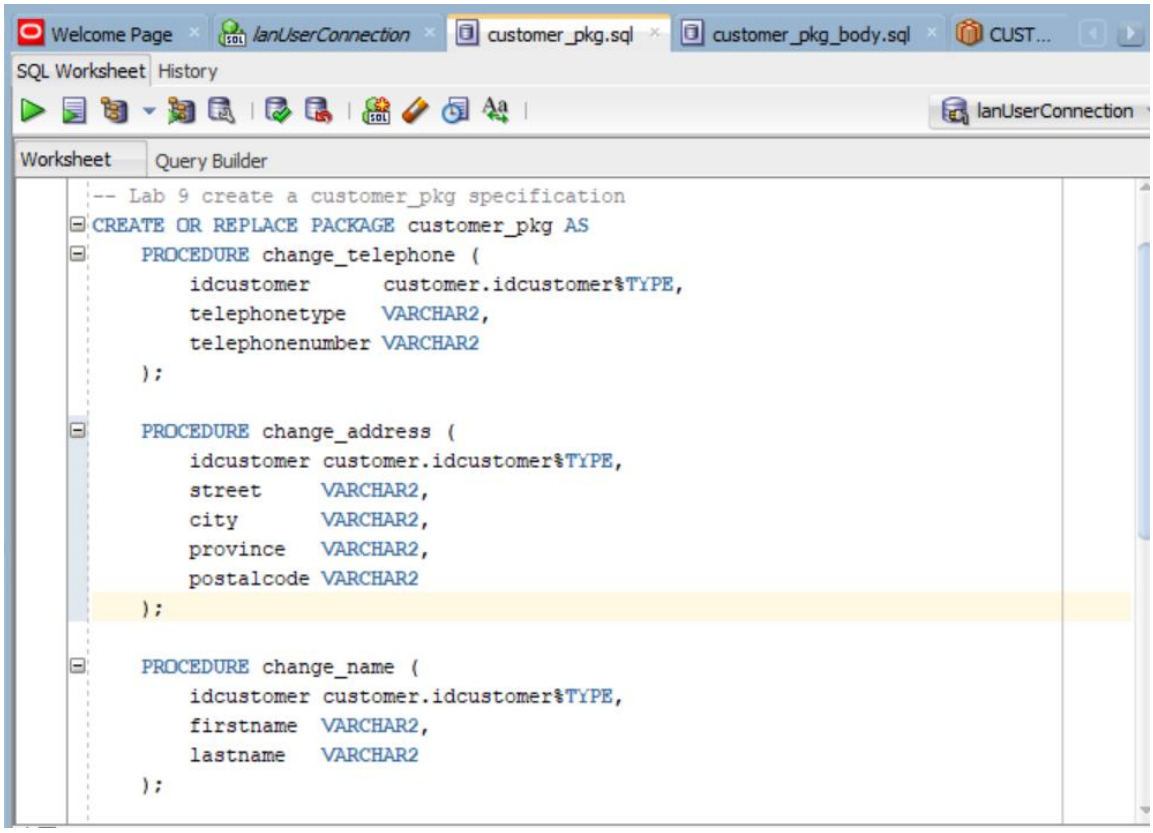
      2.1.3. change_name()

   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.

© 2021 Algonquin College of Applied Arts and Technology

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:




```oracle_sql
-- Lab 9 create a customer_pkg specification
CREATE OR REPLACE PACKAGE customer_pkg AS
    PROCEDURE change_telephone (
        idcustomer      customer.idcustomer%TYPE,
        telephonetype   VARCHAR2,
        telephonenumber VARCHAR2
    );

    PROCEDURE change_address (
        idcustomer customer.idcustomer%TYPE,
        street     VARCHAR2,
        city       VARCHAR2,
        province   VARCHAR2,
        postalcode VARCHAR2
    );

    PROCEDURE change_name (
        idcustomer customer.idcustomer%TYPE,
        firstname  VARCHAR2,
        lastname   VARCHAR2
    );
```

© 2021 Algonquin College of Applied Arts and Technology

```
    FUNCTION new_customer (
        birthdate       TIMESTAMP,
        password        VARCHAR2,
        telephonetype   VARCHAR2 DEFAULT NULL,
        telephonenumber VARCHAR2 DEFAULT NULL,
        street          VARCHAR2 DEFAULT NULL,
        city            VARCHAR2 DEFAULT NULL,
        province        VARCHAR2 DEFAULT NULL,
        postalcode      VARCHAR2 DEFAULT NULL,
        firstname       VARCHAR2 DEFAULT NULL,
        lastname        VARCHAR2 DEFAULT NULL
    ) RETURN INTEGER;

    FUNCTION get_age (
        idcustomer customer.idcustomer%TYPE
    ) RETURN INTEGER;

END customer_pkg;
/
```

3. Provide the package body for customer_pkg using the following criteria.

   3.1.1. Each procedure should <u>insert</u> a new telephone/address/name as appropriate

   3.1.2. Each procedure should <u>update</u> 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 <u>insert</u> 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. [**<u>NOTE: The new_customer function should use the three stored procedures in the package to create the related records.]</u>**

   3.1.5. The get_age() function should returns 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:

   Procedure change_telephone

Welcome Page | lanUserConnection | customer_pkg.sql | customer_pkg_body.sql

SQL Worksheet | History

0.078 seconds

Worksheet | Query Builder

```
-- Lab 9 create a customer_pkg body
CREATE OR REPLACE PACKAGE BODY customer_pkg AS

    PROCEDURE change_telephone (
        idcustomer      IN customer.idcustomer%TYPE,
        telephonetype   IN VARCHAR2,
        telephonenumber IN VARCHAR2
    ) IS
        new_idcustomer_telephone NUMBER(38, 0);
        new_idtelephone          NUMBER(38, 0);
        row_count            NUMBER;
    BEGIN
        SELECT
            COUNT(*)
        INTO row_count
        FROM
            telephone
            RIGHT OUTER JOIN customer_telephone
            ON telephone.idtelephone = customer_telephone.telephone_idtelephone
        WHERE
                customer_telephone.customer_idcustomer = idcustomer
            AND customer_telephone.enddate IS NULL;

        IF row_count != 0 THEN
            UPDATE customer_telephone
```

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

        END IF;

        new_idtelephone := seq_num.nextval;
        INSERT INTO telephone VALUES (
            new_idtelephone,
            telephonetype,
            telephonenumber
        );

        new_idcustomer_telephone := seq_num.nextval;
        INSERT INTO customer_telephone VALUES (
            new_idcustomer_telephone,
            sysdate,
            NULL,
            idcustomer,
            new_idtelephone
        );

    END change_telephone;
```

© 2021 Algonquin College of Applied Arts and Technology

Procedure change_address

```
PROCEDURE change_address (
    idcustomer customer.idcustomer%TYPE,
    street     VARCHAR2,
    city       VARCHAR2,
    province   VARCHAR2,
    postalcode VARCHAR2
) IS
    new_idcustomer_address NUMBER(38, 0);
    new_idaddress          NUMBER(38, 0);
    row_count          NUMBER;
BEGIN
    SELECT
        COUNT(*)
    INTO row_count
    FROM
        address
        RIGHT OUTER JOIN customer_address
        ON address.idaddress = customer_address.address_idaddress
    WHERE
            customer_address.customer_idcustomer = idcustomer
        AND customer_address.enddate IS NULL;

    IF row_count != 0 THEN
        UPDATE customer_address
        SET
            enddate = sysdate
```

© 2021 Algonquin College of Applied Arts and Technology

```
            WHERE
                    customer_idcustomer = idcustomer
                AND enddate IS NULL;

        END IF;

        new_idaddress := seq_num.nextval;
        INSERT INTO address VALUES (
            new_idaddress,
            street,
            city,
            province,
            postalcode
        );

        new_idcustomer_address := seq_num.nextval;
        INSERT INTO customer_address VALUES (
            new_idcustomer_address,
            sysdate,
            NULL,
            idcustomer,
            new_idaddress
        );

    END change_address;
```

© 2021 Algonquin College of Applied Arts and Technology

Procedure change_name

```
PROCEDURE change_name (
    idcustomer customer.idcustomer%TYPE,
    firstname  VARCHAR2,
    lastname   VARCHAR2
) IS
    new_idcustomer_name NUMBER(38, 0);
    new_idname          NUMBER(38, 0);
    row_count           NUMBER;
BEGIN
    SELECT
        COUNT(*)
    INTO row_count
    FROM
        cname
        RIGHT OUTER JOIN customer_name
        ON cname.idname = customer_name.name_idname
    WHERE
            customer_name.customer_idcustomer = idcustomer
        AND customer_name.enddate IS NULL;

    IF row_count != 0 THEN
        UPDATE customer_name
        SET
            enddate = sysdate
```

© 2021 Algonquin College of Applied Arts and Technology

```
            WHERE
                    customer_idcustomer = idcustomer
                AND enddate IS NULL;

        END IF;

        new_idname := seq_num.nextval;
        INSERT INTO cname VALUES (
            new_idname,
            firstname,
            lastname
        );

        new_idcustomer_name := seq_num.nextval;
        INSERT INTO customer_name VALUES (
            new_idcustomer_name,
            sysdate,
            NULL,
            idcustomer,
            new_idcustomer_name
        );

    END change_name;
```

3.1.7. Provide a screenshot showing the new_customer function below:

```
FUNCTION new_customer (
    birthdate       TIMESTAMP,
    password        VARCHAR2,
    telephonetype   VARCHAR2 DEFAULT NULL,
    telephonenumber VARCHAR2 DEFAULT NULL,
    street          VARCHAR2 DEFAULT NULL,
    city            VARCHAR2 DEFAULT NULL,
    province        VARCHAR2 DEFAULT NULL,
    postalcode      VARCHAR2 DEFAULT NULL,
    firstname       VARCHAR2 DEFAULT NULL,
    lastname        VARCHAR2 DEFAULT NULL
) RETURN INTEGER IS

    new_idcustomer           NUMBER(38, 0);
    new_idname               NUMBER(38, 0);
    new_idcustomer_name      NUMBER(38, 0);
    new_idaddress            NUMBER(38, 0);
    new_idcustomer_address   NUMBER(38, 0);
    new_idtelephone          NUMBER(38, 0);
    new_idcustomer_telephone NUMBER(38, 0);
BEGIN
    IF
        telephonetype IS NULL
        AND telephonenumber IS NULL
        AND street IS NULL
```

```
        AND city IS NULL
        AND province IS NULL
        AND postalcode IS NULL
        AND firstname IS NULL
        AND lastname IS NULL
    THEN
        RETURN -1;
    ELSE
        new_idcustomer := seq_num.nextval;
        INSERT INTO customer VALUES (
            new_idcustomer,
            birthdate,
            password
        );

        IF firstname IS NOT NULL OR lastname IS NOT NULL THEN
            change_name(new_idcustomer, firstname, lastname);
        END IF;

        IF telephonetype IS NOT NULL OR telephonenumber IS NOT NULL THEN
            change_telephone(new_idcustomer, telephonetype, telephonenumber);
        END IF;
```

```
            IF street IS NOT NULL OR city IS NOT NULL OR province IS NOT NULL
            OR postalcode IS NOT NULL THEN
                change_address(new_idcustomer, street, city, province, postalcode);
            END IF;

            RETURN new_idcustomer;
        END IF;
    END new_customer;
```

3.1.8. Provide a screenshot showing the get_age function below:

```
    FUNCTION get_age (
        idcustomer customer.idcustomer%TYPE
    ) RETURN INTEGER IS
        c_birthdate TIMESTAMP;
        age         NUMBER;
    BEGIN
        SELECT
            birthdate
        INTO c_birthdate
        FROM
            customer
        WHERE
            idcustomer = idcustomer;

        age := trunc(months_between(sysdate, c_birthdate) / 12);
        RETURN age;
    END get_age;

END customer_pkg;
/
```

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

© 2021 Algonquin College of Applied Arts and Technology