Elizabeth Waghalter **2cb58aa8-c124-4a2a-8ab8-7ca4f87804a8**

ITSE 1345 PL/SQL

Lab 2

1. CREATE SEQUENCE list#;

CREATE TABLE theater

( row\_id NUMBER GENERATED AS IDENTITY

, text VARCHAR2(10));

DECLARE

C\_XYZ CONSTANT NUMBER := 10;

V\_Counter1 BINARY\_INTEGER;

V\_Row\_number NUMBER;

StringA VARCHAR2(50);

BEGIN

NULL;

END;

/

SEQUENCE list# created.

TABLE theater created.

PL/SQL procedure successfully completed.

2. Create an anonymous block.

CREATE TABLE Student\_L2

( Stu\_ID NUMBER(5) NOT NULL

, Lname VARCHAR2(30) NOT NULL

, Fname VARCHAR2(20) NOT NULL

, MI CHAR(2)

, Major VARCHAR2(10)

, Home\_State CHAR(2));

CREATE TABLE Course\_L2

( Course\_ID VARCHAR2(10)

, Section# NUMBER(3)

, C\_Name VARCHAR2(20)

, C\_Description VARCHAR(40));

CREATE TABLE Student\_Course\_L2

( Stu\_ID NUMBER(5)

, Course\_ID VARCHAR2(10)

, Section# NUMBER(3));

BEGIN

NULL;

END;

/

Table STUDENT\_L2 created.

Table COURSE\_L2 created.

Table STUDENT\_COURSE\_L2 created.

PL/SQL procedure successfully completed.

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State) VALUES (10001, 'Smith', 'Sam', 'M', 'M', 'Math', 'Tx');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10002, 'Jones', 'Peter', 'A', 'M', 'English', 'Tx');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10003, 'Peters', 'Anne', 'A', 'F', 'English', 'Me');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10004, 'Johnson', 'John', 'J', 'M', 'CompSci', 'Ca');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10005, 'Penders', 'Alton', 'P', 'F', 'Math', 'Ga');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10006, 'Allen', 'Diane', 'J', 'F', 'Geography', 'Mn');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10007, 'Gill', 'Jennifer', ' ', 'F', 'CompSci', 'Tx');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10008, 'Johns', 'Roberta', ' ', 'F', 'CompSci', 'Tx');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10009, 'Wier', 'Paul', ' ', 'M', 'Math', 'Al');

INSERT INTO Student\_L2 (Stu\_ID, Lname, Fname, MI, Sex, Major, Home\_State)

VALUES (10010, 'Evans', 'Richard', ‘ ‘, 'M', 'English', 'Tx');

BEGIN

NULL;

END;

/

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

PL/SQL procedure successfully completed.

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('COSC\_1300', 001, 'Intro to Comp.', 'First Computer Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('ITSE\_2309', 001, 'Intro to DB', 'First Database Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('GEOG\_1791', 002, 'World Geography', 'Second Geography Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('COSC\_1315', 001, 'Intro to Prog.', 'Second Computer Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('ITSE\_1345', 001, 'Intro to DB Prog.', 'Second Database Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('ENGL\_2617', 002, 'English Literature', 'Second English Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('MATH\_1101', 001, 'Calculus I', 'Second Math Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('ENGL\_1001', 001, 'American Literature', 'First English Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('MATH\_1011', 001, 'Trig. and\_Algebra', 'First Math Course');

INSERT INTO Course\_L2 (Course\_ID, Section#, C\_Name, C\_Description)

VALUES ('GEOG\_1010', 001, 'Texas Geography', 'First Geography Course');

BEGIN

NULL;

END;

/

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

PL/SQL procedure successfully completed.

INSERT INTO STUDENT\_COURSE\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10001', 'COSC\_1300', 001);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10002', 'ITSE\_2309', 001);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10003', 'GEOG\_1791', 002);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10004', 'COSC\_1315', 001);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10005', 'ITSE\_1345', 001);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10006', 'ENGL\_2617', 002);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10007', 'MATH\_1101', 001);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10008', 'ENGL\_1001', 001);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10009', 'MATH\_1011', 001);

INSERT INTO Student\_Course\_L2 (Stu\_ID, Course\_ID, Section#)

VALUES ('10010', 'GEOG\_1010', 001);

BEGIN

NULL;

END;

/

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

PL/SQL procedure successfully completed.

SELECT and DESC all tables.

SELECT \* from Student\_L2;

STU\_ID LNAME FNAME MI SE MAJOR HO

---------- ------------------------------ -------------------- -- -- ---------- --

10001 Smith Sam M M Math Tx

10002 Jones Peter A M English Tx

10003 Peters Anne A F English Me

10004 Johnson John J M CompSci Ca

10005 Penders Alton P F Math Ga

10006 Allen Diane J F Geography Mn

10007 Gill Jennifer F CompSci Tx

10008 Johns Roberta F CompSci Tx

10009 Wier Paul M Math Al

10010 Evans Richard M English Tx

SELECT \* from Course\_L2;

COURSE\_ID SECTION# C\_NAME C\_DESCRIPTION

---------- ---------- -------------------- ----------------------------------------

COSC\_1300 1 Intro to Comp. First Computer Course

ITSE\_2309 1 Intro to DB First Database Course

GEOG\_1791 2 World Geography Second Geography Course

COSC\_1315 1 Intro to Prog. Second Computer Course

ITSE\_1345 1 Intro to DB Prog. Second Database Course

ENGL\_2617 2 English Literature Second English Course

MATH\_1101 1 Calculus I Second Math Course

ENGL\_1001 1 American Literature First English Course

MATH\_1011 1 Trig. and\_Algebra First Math Course

GEOG\_1010 1 Texas Geography First Geography Course

10 rows selected

SELECT \* from Student\_Course\_L2;

STU\_ID COURSE\_ID SECTION#

---------- ---------- ----------

10001 COSC\_1300 1

10002 ITSE\_2309 1

10003 GEOG\_1791 2

10004 COSC\_1315 1

10005 ITSE\_1345 1

10006 ENGL\_2617 2

10007 MATH\_1101 1

10008 ENGL\_1001 1

10009 MATH\_1011 1

10010 GEOG\_1010 1

10 rows selected

DESC Student\_Course\_L2;

Name Null Type

--------- ---- ------------

STU\_ID NUMBER(5)

COURSE\_ID VARCHAR2(10)

SECTION# NUMBER(3)

DESC Course\_L2;

Name Null Type

------------- ---- ------------

COURSE\_ID VARCHAR2(10)

SECTION# NUMBER(3)

C\_NAME VARCHAR2(20)

C\_DESCRIPTION VARCHAR2(40)

DESC Student\_L2;

Name Null Type

---------- -------- ------------

STU\_ID NOT NULL NUMBER(5)

LNAME NOT NULL VARCHAR2(30)

FNAME NOT NULL VARCHAR2(20)

MI CHAR(2)

SEX CHAR(2)

MAJOR VARCHAR2(10)

HOME\_STATE CHAR(2)

3. Create an anonymous block.

Count the number of courses per student in the Student\_Course\_L2 composite table

Use the tables just created above

BEGIN

FOR person IN (SELECT Stu\_ID AS student,

COUNT(Course\_ID) AS class\_enroll

FROM Student\_Course\_L2

group by Stu\_ID, Course\_ID

order by Stu\_ID) LOOP

dbms\_output.put\_line (person.student || ' is taking ' || person.class\_enroll);

END LOOP;

END;

/

PL/SQL procedure successfully completed.

10001 is taking 1

10002 is taking 1

10003 is taking 1

10004 is taking 1

10005 is taking 1

10006 is taking 1

10007 is taking 1

10008 is taking 1

10009 is taking 1

10010 is taking 1

4. Create an anonymous block.

Count the number of students enrolled in the Student table.

Use dbms\_output.put\_line with message: “Total number of students enrolled: \_\_\_\_”

DECLARE

stu\_enroll\_total NUMBER;

BEGIN

COUNT(UNIQUE Stu\_ID) AS stu\_count

INTO stu\_enroll\_total

FROM Student\_L2;

dbms\_output.put\_line('Total number of students enrolled: [' || stu\_enroll\_total || ']');

END;

/

PL/SQL procedure successfully completed.

Total number of students enrolled: [10]

5. Create an anonymous block.

Determine number of courses in the Student\_Course\_L2 table.

Use an IF statement: if # > 10, put\_line (“More than 10 courses have been established.”)

ELSIF # < 10, put\_line(“Less than 10 courses have been established.”

DECLARE

course\_count\_total NUMBER;

BEGIN

SELECT COUNT(UNIQUE Course\_ID) AS course\_count

INTO course\_count\_total

FROM Student\_Course\_L2;

IF course\_count\_total > 10 THEN

dbms\_output.put\_line('More than 10 courses have been established.');

ELSE

dbms\_output.put\_line('10 or less courses have been established.');

END IF;

END;

/

PL/SQL procedure successfully completed.

10 or less courses have been established.

6. Create an anonymous block.

Test each student from Student\_L2 table for instate status:

Use dbms\_output.put\_line to state whether each student is instate or out-of-state.

Provide a count of each.

DECLARE

in\_state CHAR(2) := 'Tx';

in\_state\_status NUMBER(2) := 0;

out\_state\_status NUMBER(2) := 0;

CURSOR state\_status\_cursor IS SELECT Stu\_ID, Home\_State FROM Student\_L2;

BEGIN

FOR student IN state\_status\_cursor LOOP

IF student.Home\_State = in\_state THEN

in\_state\_status := in\_state\_status + 1;

dbms\_output.put\_line('This student is in\_state.');

ELSE

out\_state\_status := out\_state\_status + 1;

dbms\_output.put\_line('This student is out\_of\_state.');

END IF;

END LOOP;

dbms\_output.put\_line('The total number of in\_state students: [' || in\_state\_status || ']');

dbms\_output.put\_line('The total number of out\_of\_state students: [' || out\_state\_status || ']');

END;

/

PL/SQL procedure successfully completed.

This student is in\_state.

This student is in\_state.

This student is out\_of\_state.

This student is out\_of\_state.

This student is out\_of\_state.

This student is out\_of\_state.

This student is in\_state.

This student is in\_state.

This student is out\_of\_state.

This student is in\_state.

The total number of in\_state students: [5]

The total number of out\_of\_state students: [5]

7. Create an anonymous block.

Use Named Parameters, &Variable\_Name to check student enrollment in specified courses.

Use put\_line with indicator message.

accept L2\_Stu\_var NUMBER prompt 'What is the Student ID?'

accept L2\_Cours\_var CHAR prompt 'What is the Course ID?'

DECLARE

L2\_Stu NUMBER(5);

L2\_Cours varchar2(10);

BEGIN

L2\_Stu := &L2\_Stu\_var;

L2\_Cours := '&L2\_Cours\_var';

IF L2\_Stu >= 10001 AND L2\_Stu <= 10010 THEN

IF L2\_Cours = 'cosc 1300'

or L2\_Cours = 'itse 2309'

or L2\_Cours = 'geog 1791'

or L2\_Cours = 'cosc 1315'

or L2\_Cours = 'itse 1345'

or L2\_Cours = 'eng 2617'

or L2\_Cours = 'math 1101'

or L2\_Cours = 'engl 1001'

or L2\_Cours = 'math 1011'

or L2\_Cours = 'geog 1010' THEN

dbms\_output.put\_line('Student is taking the course.');

ELSE

dbms\_output.put\_line('Student is not taking the course.');

END IF;

ELSE

dbms\_output.put\_line('Student does not exist.');

END IF;

END;

/

PL/SQL procedure successfully completed.

Student is taking the course.

8. Create an anonymous block

Use IF to test for Male or Female

Use CASE to determine the major

DECLARE

lv\_sex STUDENT\_L2.SEX%TYPE;

lv\_major STUDENT\_L2.MAJOR%TYPE;

CURSOR sex\_major IS

SELECT sex, major FROM STUDENT\_L2;

BEGIN

OPEN sex\_major;

LOOP

FETCH sex\_major

INTO lv\_sex,

lv\_major;

EXIT WHEN sex\_major%NOTFOUND;

IF lv\_sex = 'F' THEN

dbms\_output.put\_line('This student is female, ');

ELSE

dbms\_output.put\_line('This student is male, ');

END IF;

CASE

WHEN(lv\_major = 'Math') THEN

dbms\_output.put\_line('majoring in Math.');

WHEN(lv\_major = 'English') THEN

dbms\_output.put\_line('majoring in English.');

WHEN(lv\_major = 'CompSci') THEN

dbms\_output.put\_line('majoring in Computer Science.');

WHEN(lv\_major = 'Geography') THEN

dbms\_output.put\_line('majoring in Geography.');

ELSE

dbms\_output.put\_line('Please enter an appropriate major.');

END CASE;

END LOOP;

END;

/

PL/SQL procedure successfully completed.

This student is male,

majoring in Math.

This student is male,

majoring in English.

This student is female,

majoring in English.

This student is male,

majoring in Computer Science.

This student is female,

majoring in Math.

This student is female,

majoring in Geography.

This student is female,

majoring in Computer Science.

This student is female,

majoring in Computer Science.

This student is male,

majoring in Math.

This student is male,

majoring in English.

9. Create an anonymous block.

same as above except use IF construct instead of CASE construct.

DECLARE

lv\_sex STUDENT\_L2.SEX%TYPE;

lv\_major STUDENT\_L2.MAJOR%TYPE;

CURSOR sex\_major IS

SELECT sex, major FROM STUDENT\_L2;

BEGIN

OPEN sex\_major;

LOOP

FETCH sex\_major

INTO lv\_sex,

lv\_major;

EXIT WHEN sex\_major%NOTFOUND;

IF lv\_sex = 'F' THEN

dbms\_output.put\_line('This student is female, ');

ELSE

dbms\_output.put\_line('This student is male, ');

END IF;

IF(lv\_major = 'Math') THEN

dbms\_output.put\_line('majoring in Math.');

ELSIF(lv\_major = 'English') THEN

dbms\_output.put\_line('majoring in English.');

ELSIF(lv\_major = 'CompSci') THEN

dbms\_output.put\_line('majoring in Computer Science.');

ELSIF(lv\_major = 'Geography') THEN

dbms\_output.put\_line('majoring in Geography.');

ELSE

dbms\_output.put\_line('Please enter an appropriate major.');

END IF;

END LOOP;

END;

/

PL/SQL procedure successfully completed.

This student is male,

majoring in Math.

This student is male,

majoring in English.

This student is female,

majoring in English.

This student is male,

majoring in Computer Science.

This student is female,

majoring in Math.

This student is female,

majoring in Geography.

This student is female,

majoring in Computer Science.

This student is female,

majoring in Computer Science.

This student is male,

majoring in Math.

This student is male,

majoring in English.

10. Create an anonymous block

SELECT

INTO

from student\_table, temp\_table

COUNTER (student\_table%ROWCOUNT) LOOP

create temp\_table first

CREATE GLOBAL TEMPORARY TABLE temp\_table

(Stu\_ID NUMBER(5),

LName VARCHAR2(30),

FName VARCHAR2(20),

MI CHAR(2),

SEX CHAR(2),

Major VARCHAR2(10),

Home\_State CHAR(2));

Global temporary TABLE created.

DECLARE

counter NUMBER := 1;

c\_var NUMBER;

BEGIN

SELECT COUNT(Stu\_ID) into c\_var from STUDENT\_L2;

WHILE (counter < c\_var) LOOP

INSERT INTO temp\_table

SELECT \*

FROM Student\_L2;

counter := counter + 1;

END LOOP;

END;

/

PL/SQL procedure successfully completed.

DESC TEMP\_TABLE;

Name Null Type

---------- ---- ------------

STU\_ID NUMBER(5)

LNAME VARCHAR2(30)

FNAME VARCHAR2(20)

MI CHAR(2)

SEX CHAR(2)

MAJOR VARCHAR2(10)

HOME\_STATE CHAR(2)