Local V\_MINE is here: 700

Outer V\_MINE is here: 500

Outer V\_MINE is here: 1400

SET SERVEROUTPUT ON

SET VERIFY OFF

DECLARE

v\_des VARCHAR2(30);

v\_number NUMBER(8,2);

v\_constant CONSTANT VARCHAR2(10) := '704B';

v\_boolean BOOLEAN;

v\_date DATE := TRUNC(SYSDATE) + 7;

BEGIN

DBMS\_OUTPUT.PUT\_LINE ('The constant is: '||v\_constant||'.');

DBMS\_OUTPUT.PUT\_LINE ('The date is: '||v\_date||'.');

-- D

-- v\_des := 'C++ advanced';

IF v\_des LIKE '%SQL%' THEN

DBMS\_OUTPUT.PUT\_LINE(v\_des);

ELSE

IF v\_constant LIKE '%704B%' THEN

IF v\_des IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Course name: ' || v\_des || 'Room name: ' || v\_constant );

ELSE

DBMS\_OUTPUT.PUT\_LINE('Course is unknown' || ' Room name: ' || v\_constant );

END IF;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Course and location could not be determined');

END IF;

END IF;

END;

/

A-C Result:

The constant is: 704B.

The date is: 27-SEP-17.

Course is unknown Room name: 704B

D Result:

The constant is: 704B.

The date is: 27-SEP-17.

Course name: C++ advanced Room name: 704B

1. --a. Declares two variables to hold values for columns of table Lab1\_tab

/\*

CREATE TABLE lab1\_tab (

ID NUMBER,

lname VARCHAR2(20)

);

--b. sequence called Lab1\_seq, increments 5, starts with 1.

CREATE SEQUENCE lab1\_seq

START WITH 1 INCREMENT BY 5;

\*/

-- c. to g.

SET serveroutput ON

SET verify OFF

DECLARE

v\_lname student.last\_name%TYPE;

BEGIN

--b. The block then inserts into the table the last name of the student that

-- is enrolled in the most classes and his/her last name contains less than 9

-- characters. Here use a sequence for the Id

BEGIN

SELECT s.last\_name

INTO v\_lname

FROM enrollment e, student s

WHERE e.student\_id = s.student\_id

AND LENGTH(s.last\_name) < 9

GROUP BY s.last\_name

HAVING COUNT(\*) = (

SELECT MAX(count(student\_id))

FROM enrollment

GROUP BY student\_id);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

v\_lname := 'Multiple Names';

END;

INSERT INTO lab1\_tab

VALUES (LAB1\_SEQ.NEXTVAL, v\_lname);

--c. the student with the least enrollments is inserted in the table,

-- use sequence as well.

BEGIN

SELECT s.last\_name

INTO v\_lname

FROM enrollment e, student s

WHERE e.student\_id = s.student\_id

AND LENGTH(s.last\_name) < 9

GROUP BY s.last\_name

having count(\*) = (

SELECT MIN(count(student\_id))

FROM enrollment

GROUP BY student\_id);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

v\_lname := 'Multiple Names';

END;

INSERT INTO lab1\_tab

VALUES (LAB1\_SEQ.NEXTVAL, v\_lname);

--d. Insert the instructor’s last name teaching the least amount of courses

-- if his/her last name does NOT end on “s”. Here do not use the sequence to generate the ID; instead use your first variable.

BEGIN

SELECT i.last\_name

INTO v\_lname

FROM instructor i, section s

WHERE s.instructor\_id = i.instructor\_id

AND i.last\_name NOT LIKE '%s'

GROUP BY i.last\_name

having count(\*) = (

SELECT MIN(count(\*))

FROM section

GROUP BY instructor\_id

);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

v\_lname := 'Multiple Names';

END;

INSERT INTO lab1\_tab

VALUES (1, v\_lname);

--e.Now insert the instructor teaching the most number of courses and use

-- the sequence to populate his/her Id

BEGIN

SELECT i.last\_name

INTO v\_lname

FROM instructor i, section s

WHERE s.instructor\_id = i.instructor\_id

AND i.last\_name NOT LIKE '%s'

GROUP BY i.last\_name

having count(\*) = (

SELECT MAX(count(\*))

FROM section

GROUP BY instructor\_id

);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

v\_lname := 'Multiple Names';

END;

INSERT INTO lab1\_tab

VALUES (LAB1\_SEQ.NEXTVAL, v\_lname);

END;

/

Result:

select \* from lab1\_tab;

ID LNAME

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

1 Multiple Names

6 Multiple Names

1 Lowry

11 Multiple Names