**1)**

SET SERVEROUTPUT ON

DECLARE

v\_a VARCHAR2(30); /\*Introduction to Oracle PL/SQL -> has length 29 \*/

v\_b NUMBER(8,2); /\* total number of digits = 8 and rounding occurs at the second \*/

v\_c CONSTANT VARCHAR2(4) := '603D'; /\*->auto\_init to 603D\*/

v\_d BOOLEAN;

v\_e DATE := SYSDATE + 7; /\*->auto\_init = one week from today\*/

BEGIN

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

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

/\* v\_b is an EMPTY numeric variable \*/

END;

**2)**

SET SERVEROUTPUT ON

DECLARE

v\_a VARCHAR2(30) := 'Data Bases 2'; /\*Introduction to Oracle PL/SQL -> has length 29 \*/

v\_b NUMBER(8,2); /\* total number of digits = 8 and rounding occurs at the second \*/

v\_c CONSTANT VARCHAR2(4) := '603D'; /\*->auto\_init to 603D\*/

v\_d BOOLEAN;

v\_e DATE := SYSDATE + 7; /\*->auto\_init = one week from today\*/

BEGIN

IF v\_a LIKE 'Introduction to Underwater Basketweaving' THEN

DBMS\_OUTPUT.PUT\_LINE('Was a match: '||v\_a);

ELSE IF v\_c LIKE '603D' THEN

DBMS\_OUTPUT.PUT\_LINE('The course name is: '||v\_a||' and the room number is: '||v\_c);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Unidentified course and room number!');

END IF;

END IF;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occured!');

END;

**3)**

CREATE TABLE Ex3(

ID NUMBER(10,0),

NAME VARCHAR2(20)

);

CREATE SEQUENCE Ex3\_SEQ INCREMENT BY 5;

**4)**

Error report -

ORA-01422: exact fetch returns more than requested number of rows

ORA-06512: at line 5

01422. 00000 - "exact fetch returns more than requested number of rows"

\*Cause: The number specified in exact fetch is less than the rows returned.

\*Action: Rewrite the query or change number of rows requested

I added : EXCEPTION WHEN OTHERS THEN, because of the error above, since the result for “most classes” can generate more than 1 student entry!

1. **& b)**

SET SERVEROUTPUT ON

DECLARE

v\_name STUDENT.last\_name%TYPE;

v\_id STUDENT.student\_id%TYPE;

BEGIN

SELECT st.last\_name INTO v\_name FROM STUDENT st, ENROLLMENT et

WHERE st.student\_id = et.student\_id

HAVING COUNT(\*) = (SELECT MAX(COUNT(\*)) FROM STUDENT stt, ENROLLMENT ett WHERE stt.student\_id = ett.student\_id GROUP BY ett.student\_id)

GROUP BY st.last\_name;

EXCEPTION

WHEN OTHERS THEN

v\_name := 'More than one entry!';

INSERT INTO Ex3(id, name) VALUES (Ex3\_SEQ.NEXTVAL, v\_name);

DBMS\_OUTPUT.PUT\_LINE('Student name and id: '||v\_name);

END;

SAVEPOINT A;

**c)**

SET SERVEROUTPUT ON

DECLARE

v\_name STUDENT.last\_name%TYPE;

v\_id STUDENT.student\_id%TYPE;

BEGIN

SELECT st.last\_name INTO v\_name FROM STUDENT st, ENROLLMENT et

WHERE st.student\_id = et.student\_id

HAVING COUNT(\*) = (SELECT MIN(COUNT(\*)) FROM STUDENT stt, ENROLLMENT ett WHERE stt.student\_id = ett.student\_id GROUP BY ett.student\_id)

GROUP BY st.last\_name;

EXCEPTION

WHEN OTHERS THEN

v\_name := 'More than one entry!';

INSERT INTO Ex3(id, name) VALUES (Ex3\_SEQ.NEXTVAL, v\_name);

DBMS\_OUTPUT.PUT\_LINE('Student name and id: '||v\_name);

END;

SAVEPOINT B;

**d)**

SET SERVEROUTPUT ON

DECLARE

v\_name STUDENT.last\_name%TYPE;

v\_id STUDENT.student\_id%TYPE;

BEGIN

SELECT it.last\_name INTO v\_name FROM INSTRUCTOR it, SECTION st

WHERE it.instructor\_id = st.section\_id

HAVING COUNT(\*) = (SELECT MAX(COUNT(\*)) FROM INSTRUCTOR itt, SECTION stt WHERE stt.section\_id = itt.instructor\_id GROUP BY itt.instructor\_id)

GROUP BY it.last\_name;

EXCEPTION

WHEN OTHERS THEN

v\_name := 'More than one entry!';

INSERT INTO Ex3(id, name) VALUES (Ex3\_SEQ.NEXTVAL, v\_name);

DBMS\_OUTPUT.PUT\_LINE('Student name and id: '||v\_name);

END;

SAVEPOINT C;

**e)**

BEGIN

SELECT instructor\_id INTO v\_id FROM INSTRUCTOR WHERE last\_name = v\_name;

EXCEPTION

WHEN OTHER THEN

v\_id:=0;

END;

**f)**

ROLLBACK [TO\_B] to SAVEPOINT B;

**g)**

SET SERVEROUTPUT ON

DECLARE

v\_name STUDENT.last\_name%TYPE;

v\_id STUDENT.student\_id%TYPE;

BEGIN

SELECT it.last\_name INTO v\_name FROM INSTRUCTOR it, SECTION st

WHERE it.instructor\_id= st.instructor\_id

HAVING COUNT(\*) = (SELECT MIN(COUNT(\*)) FROM INSTRUCTOR itt, SECTION stt WHERE stt.instructor\_id = itt.instructor\_id GROUP BY itt.instructor\_id)

GROUP BY it.last\_name;

EXCEPTION

WHEN OTHERS THEN

v\_name := 'More than one entry!';

INSERT INTO Ex3(id, name) VALUES (v\_id, v\_name);

DBMS\_OUTPUT.PUT\_LINE('Instructor name and id: '||v\_name);

END;

**h)**

SET SERVEROUTPUT ON

DECLARE

v\_name STUDENT.last\_name%TYPE;

v\_id STUDENT.student\_id%TYPE;

BEGIN

SELECT it.last\_name INTO v\_name FROM INSTRUCTOR it, SECTION st

WHERE it.instructor\_id= st.instructor\_id

HAVING COUNT(\*) = (SELECT MIN(COUNT(\*)) FROM INSTRUCTOR itt, SECTION stt WHERE stt.instructor\_id = itt.instructor\_id GROUP BY itt.instructor\_id)

GROUP BY it.last\_name;

EXCEPTION

WHEN OTHERS THEN

v\_name := 'More than one entry!';

INSERT INTO Ex3(id, name) VALUES (Ex3\_SEQ.NEXTVAL, v\_name);

DBMS\_OUTPUT.PUT\_LINE('Instructor name and id: '||v\_name);

END;