**DBS501 ASSIGNMENT ONE GROUP 8**

**Q1:**

SET SERVEROUTPUT ON;

SET VERIFY OFF;

ACCEPT city PROMPT 'Please provide the valid city without department: '

DECLARE

v\_department departments%ROWTYPE;

v\_one\_department BOOLEAN;

v\_many\_department BOOLEAN;

v\_no\_city BOOLEAN;

v\_location departments.location\_id%TYPE;

v\_dept\_count NUMBER(3);

BEGIN

SELECT location\_id INTO v\_location

FROM locations

WHERE city = '&city';

SELECT COUNT(\*) INTO v\_dept\_count

FROM departments

WHERE location\_id = v\_location;

IF v\_dept\_count = 1 THEN

DBMS\_OUTPUT.PUT\_LINE('This city already contains department: ' || '&city');

ELSIF v\_dept\_count > 1 THEN

DBMS\_OUTPUT.PUT\_LINE('This city has MORE THAN ONE department: ' || '&city');

ELSE

SELECT MAX(department\_id) + 50 INTO v\_department.department\_id

FROM departments;

SELECT manager\_id INTO v\_department.manager\_id

FROM employees

GROUP BY manager\_id

HAVING count(manager\_id) = ( SELECT MAX(COUNT(manager\_id))

FROM employees

GROUP BY manager\_id);

v\_department.department\_name := 'Testing';

v\_department.location\_id := v\_location;

INSERT INTO departments VALUES v\_department;

END IF;

EXCEPTION

WHEN no\_data\_found THEN

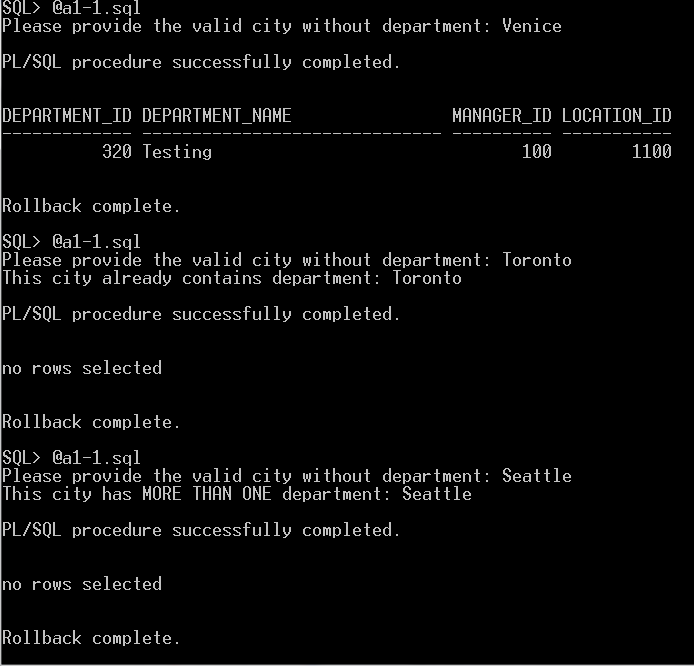
DBMS\_OUTPUT.PUT\_LINE('This city is NOT listed: ' || '&city');

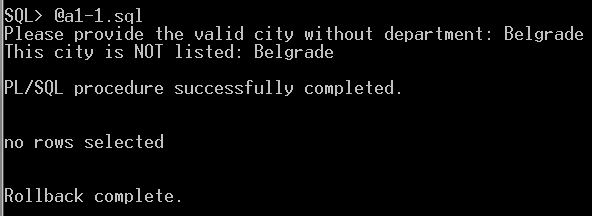
END;

/

SELECT \* FROM departments WHERE department\_name = 'Testing';

ROLLBACK;





**Q2:**

ACCEPT course\_desc PROMPT 'Enter the beginning of the Course Description in UPPER case: '

DECLARE

v\_description varchar(50):='&course\_desc';

cursor course\_c1 is

select course\_no from course

where upper(description) like v\_description||'%';

cursor section\_c1 is

select section\_id from section

where course\_no in (select course\_no from course

where prerequisite in (select course\_no from course

where upper(description) like v\_description||'%'))

order by section\_id;

v\_count number;

v\_section section\_c1%rowtype;

v\_course course\_c1%rowtype;

v\_flag char(1):='y';

e\_manyStudent EXCEPTION;

begin

open course\_c1;

fetch course\_c1 into v\_course;

if course\_c1%NOTFOUND then

DBMS\_OUTPUT.PUT\_LINE('There is NO VALID course that starts on: '||v\_description ||'. Try again.');

v\_flag :='n';

close course\_c1;

end if;

if v\_flag='y' then

open section\_c1;

fetch section\_c1 into v\_section;

if section\_c1%NOTFOUND then

DBMS\_OUTPUT.PUT\_LINE('There is NO PREREQUISITE course that starts on:'|| v\_description||'. Try again.');

close section\_c1;

else

close section\_c1;

for section\_ind in section\_c1 loop

begin

select count(\*) into v\_count

from enrollment

where section\_id=section\_ind.section\_id;

if v\_count>=7 then

raise e\_manyStudent;

else

DBMS\_OUTPUT.PUT\_LINE('There are '||v\_count ||' students for section ID '||section\_ind.section\_id);

end if;

EXCEPTION

when e\_manyStudent then

DBMS\_OUTPUT.PUT\_LINE('There are too many students for section '||section\_ind.section\_id);

DBMS\_OUTPUT.PUT\_LINE('^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^');

end;

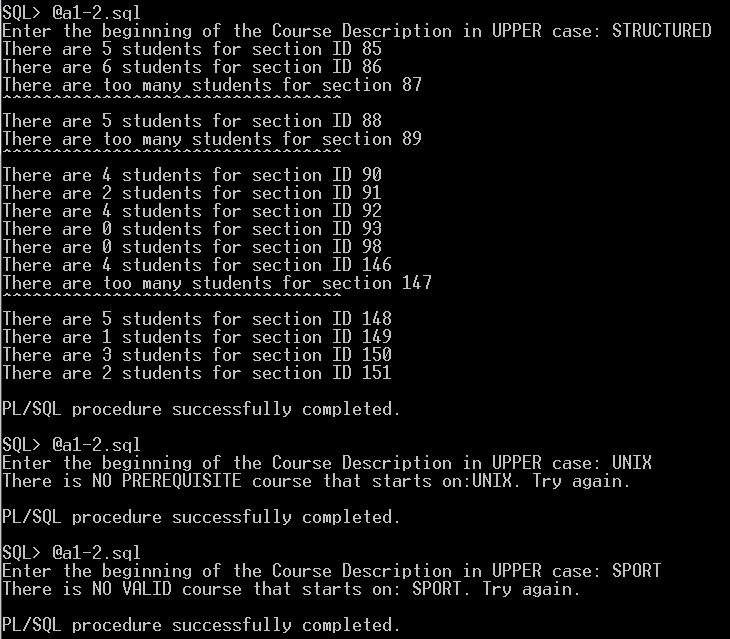
end loop;

end if;

end if;

end ;

/



**Q3:**

SET SERVEROUTPUT ON;

SET VERIFY OFF;

accept desc prompt 'Enter the beginning of the Course description in UPPER case: '

declare

desc\_input varchar(50) :='&desc';

cursor c1 is

select course\_no, description, cost, prerequisite

from course

where upper(description) like upper(desc\_input)||'%';

type pre\_rec\_type is record(

pre\_course\_no course.course\_no%type,

pre\_description course.description%type,

pre\_cost course.cost%type);

type course\_rec\_type is record(

course\_no course.course\_no%type,

description course.description%type,

cost course.cost%type,

pre\_rec pre\_rec\_type);

pre\_rec pre\_rec\_type;

course\_rec course\_rec\_type;

Begin

open c1;

fetch c1 into course\_rec.course\_no, course\_rec.description, course\_rec.cost, course\_rec.pre\_rec.pre\_course\_no;

if c1%NOTFOUND then

dbms\_output.put\_line('There is NO VALID course that starts on: '||desc\_input||'. Try again.');

close c1;

else

close c1;

for ind in c1 loop

course\_rec.course\_no:=ind.course\_no;

course\_rec.description:=ind.description;

course\_rec.cost:=ind.cost;

course\_rec.pre\_rec.pre\_course\_no:=ind.prerequisite;

begin

select description, cost into course\_rec.pre\_rec.pre\_description, course\_rec.pre\_rec.pre\_cost

from course

where course\_no =course\_rec.pre\_rec.pre\_course\_no;

dbms\_output.put\_line('Course: '||course\_rec.course\_no||' - '||course\_rec.description);

dbms\_output.put\_line('Cost: '||course\_rec.cost);

dbms\_output.put\_line('Prerequisite: '||course\_rec.pre\_rec.pre\_course\_no||' - '||course\_rec.pre\_rec.pre\_description);

dbms\_output.put\_line('Prerequisite Cost: '||course\_rec.pre\_rec.pre\_cost);

dbms\_output.put\_line('=========================================================');

exception

when NO\_DATA\_FOUND then

dbms\_output.put\_line('There is NO prerequisite course for any course that starts on '||upper(desc\_input)||'. Try again. ');

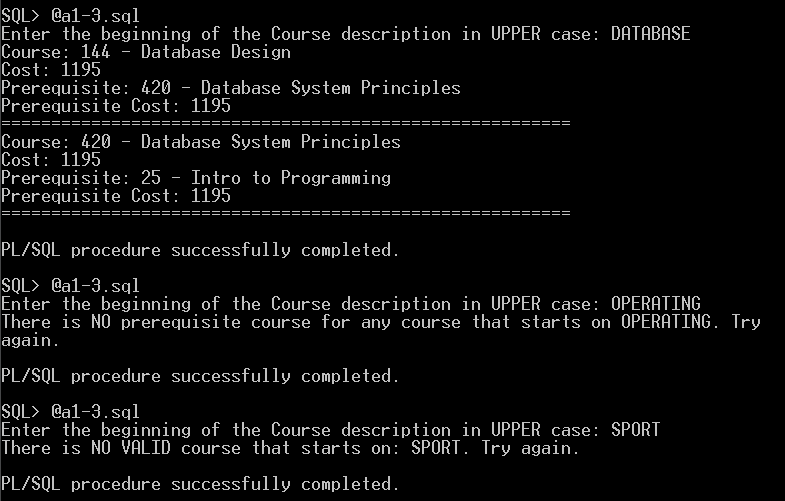
end;

end loop;

end if;

end;

/



**Q4:**

SET SERVEROUTPUT ON;

SET VERIFY OFF;

ACCEPT firstEnSearch PROMPT 'Enter your first program key word '

ACCEPT secondEnSearch PROMPT 'Enter your second program key word '

DECLARE

CURSOR course\_cl IS SELECT course\_no, description from course

WHERE UPPER(description) LIKE UPPER('%' || '&firstEnSearch' ||'%') AND UPPER (description) LIKE UPPER('%' || '&secondEnSearch' || '%');

CURSOR section\_cl(p\_course NUMBER) IS

SELECT section\_id ,section\_no FROM section WHERE course\_no = p\_course;

CURSOR studEnr(p\_section NUMBER) IS

SELECT COUNT(\*) cont FROM enrollment WHERE section\_id = p\_section;

v\_course course\_cl%ROWTYPE;

checker course\_cl%ROWTYPE;

v\_section section\_cl%ROWTYPE;

studentsEnrolled NUMBER(5);

BEGIN

OPEN course\_cl;

FETCH course\_cl into checker;

IF course\_cl%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE('There is NO course containing these 2 words. Try again.');

CLOSE course\_cl;

ELSE

CLOSE course\_cl;

for course\_ind in course\_cl LOOP

v\_course.course\_no :=course\_ind.course\_no;

v\_course.description := course\_ind.description;

DBMS\_OUTPUT.PUT\_LINE (v\_course.course\_no || ' ' || v\_course.description || CHR(10)||

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for section\_ind in section\_cl(v\_course.course\_no) LOOP

for student\_ind in studEnr(section\_ind.section\_id) LOOP

studentsEnrolled:=student\_ind.cont;

if studentsEnrolled>0 then

DBMS\_OUTPUT.PUT\_LINE ('Section: ' || section\_ind.section\_no || ' has an enrollment of: ' || studentsEnrolled);

end if;

end loop;

end loop;

end loop;

END IF;

END;

/

