Assignment – 7

Name – Abhishek Pratap Singh **Roll No – 197103**

Sec – A

Table:

```
create table student
(
  id NUMBER PRIMARY KEY,
  fname varchar2(20),
  lname varchar2(20),
  street varchar2(20),
  city varchar2(20),
  state_ varchar2(20),
  zipcode NUMBER(6)
);

create table instructor
(
   id NUMBER,
   section varchar2(20),
   PRIMARY KEY(id, section)
);

create table classes
(
classid NUMBER,
studentId NUMBER,
Foreign key (studentId) REFERENCES student (id)
);
```

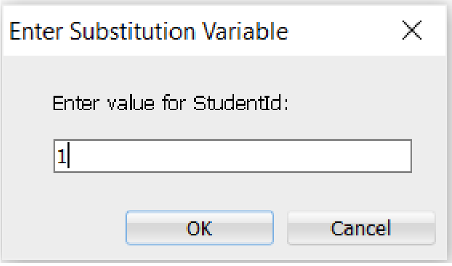
Inserting Values:

```
insert into student values (1, 'Abhishek', 'Singh', 'Kshatriya Bhawan',
'Rajgir', 'Bihar', 803116);

insert into instructor values( 1, 'A');
insert into instructor values( 1, 'B');
insert into instructor values( 1, 'C');
insert into instructor values( 1, 'D');
insert into instructor values( 2, 'A');
insert into instructor values( 2, 'B');
insert into classes values (1, 1);
```

Queries:

1	Create a package specification and the body for a package named ***School_api()***. The package contains the procedure ***Get_name_addressi()*** and the function ***Instructor_status().*** ***Get_name_addressi*****():** The procedure should accept two parameters to hold a table name and an ID and should return six parameters with first name, last name, street, city, state, and zip code information ***Instroctor_status()***: For a given instructor, determine how many sections he or she is teaching. If the number is greater than or equal to 3, return a message saying that the instructor needs a vacation. Otherwise, return a message saying how many sections this instructor is teaching.

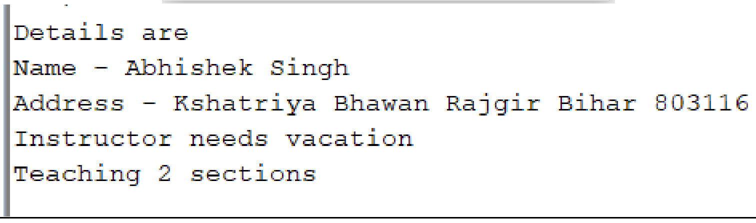
```
create or replace package School_api is
  PROCEDURE Get_name_addressi
    (p_id IN number,
    p_fname OUT varchar2,
    p_lname OUT varchar2,
    p_street OUT varchar2,
    p_city OUT varchar2,
    p_state_ OUT varchar2,
    p_zipcode OUT NUMBER);

    PROCEDURE instructor_status(
    p_id IN NUMBER,
    p_msg OUT varchar2
    );
END School_api;


create or replace package body School_api is
  PROCEDURE Get_name_addressi
    (p_id IN number,
    p_fname OUT varchar2,
    p_lname OUT varchar2,
    p_street OUT varchar2,
    p_city OUT varchar2,
    p_state_ OUT varchar2,
    p_zipcode OUT NUMBER) is
  BEGIN
    select fname, lname, street, city, state_, zipcode into
    p_fname, p_lname, p_street, p_city, p_state_, p_zipcode from student
    where id = p_id;
  END Get_name_addressi;

  PROCEDURE instructor_status
  (p_id IN NUMBER,
   p_msg OUT varchar2) is
   cnt NUMBER;
   BEGIN
   select count(*) into cnt from instructor where id = p_id;
   IF cnt >= 3
   THEN p_msg := 'Instructor needs vacation';
   ELSE p_msg := 'Teaching ' || cnt || ' sections';
   END IF;
   END instructor_status;
END;



DECLARE
    p_id number := &StudentId;
    p_fname varchar2(20);
    p_lname varchar2(20);
    p_street varchar2(20);
    p_city varchar2(20);
    p_state_ varchar2(20);
    p_zipcode NUMBER;
    p_msg varchar2(50);
    p_id1 NUMBER := 1;
    p_id2 NUMBER := 2;
BEGIN
    school_api.get_name_addressi
    (p_id, p_fname, p_lname, p_street, p_city, p_state_, p_zipcode);
    dbms_output.put_line('Details are ');
    dbms_output.put_line('Name - ' || p_fname || ' ' || p_lname);
    dbms_output.put_line('Address - ' || p_street || ' ' || p_city ||
    ' ' || p_state_ || ' ' || p_zipcode);

    school_api.instructor_status(p_id1, p_msg);
    dbms_output.put_line(p_msg);
    school_api.instructor_status(p_id2, p_msg);
    dbms_output.put_line(p_msg);
END;
```



Enter Substitution Variable

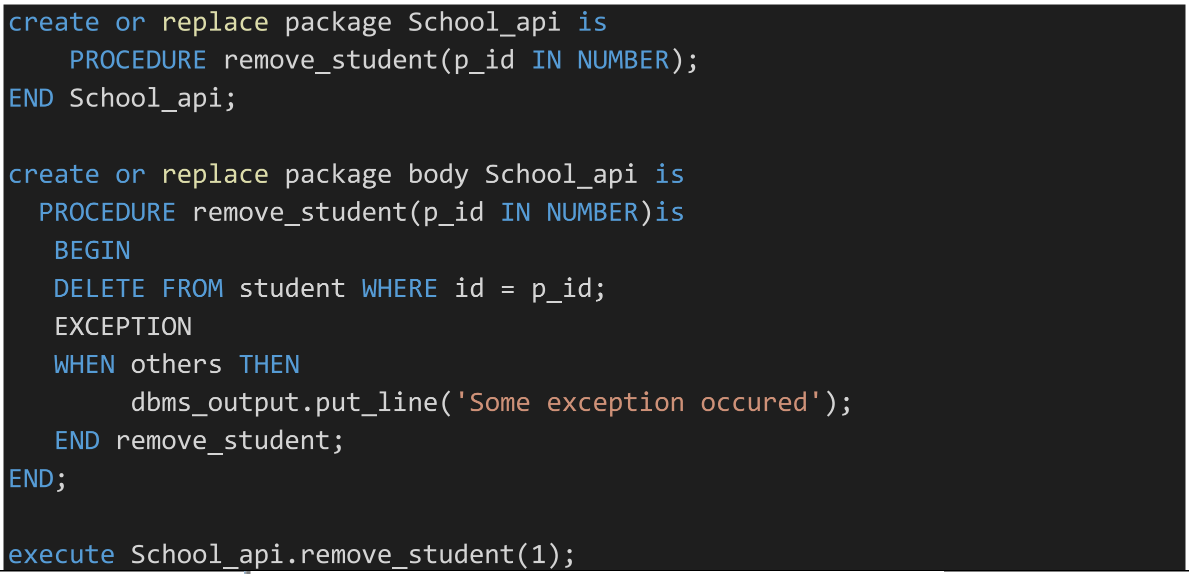
Enter value for StudentId:

1

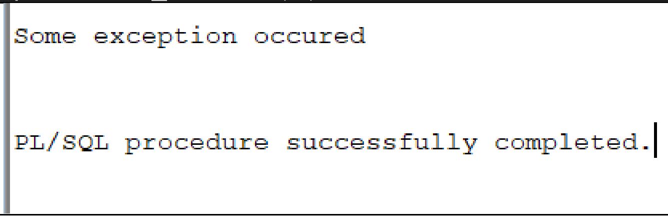


```
Details are
Name - Abhishek Singh
Address - Kshatriya Bhawan Rajgir Bihar 803116
Instructor needs vacation
Teaching 2 sections
```

2	Add a procedure to the ***school_api*** package called ***remove_student***. This procedure accepts a student_id and returns nothing. Based on the student ID passed in, it removes the student from the database. If the student does not exist or if a problem occurs while removing the student (such as a foreign key constraint violation), let the calling program handle it.



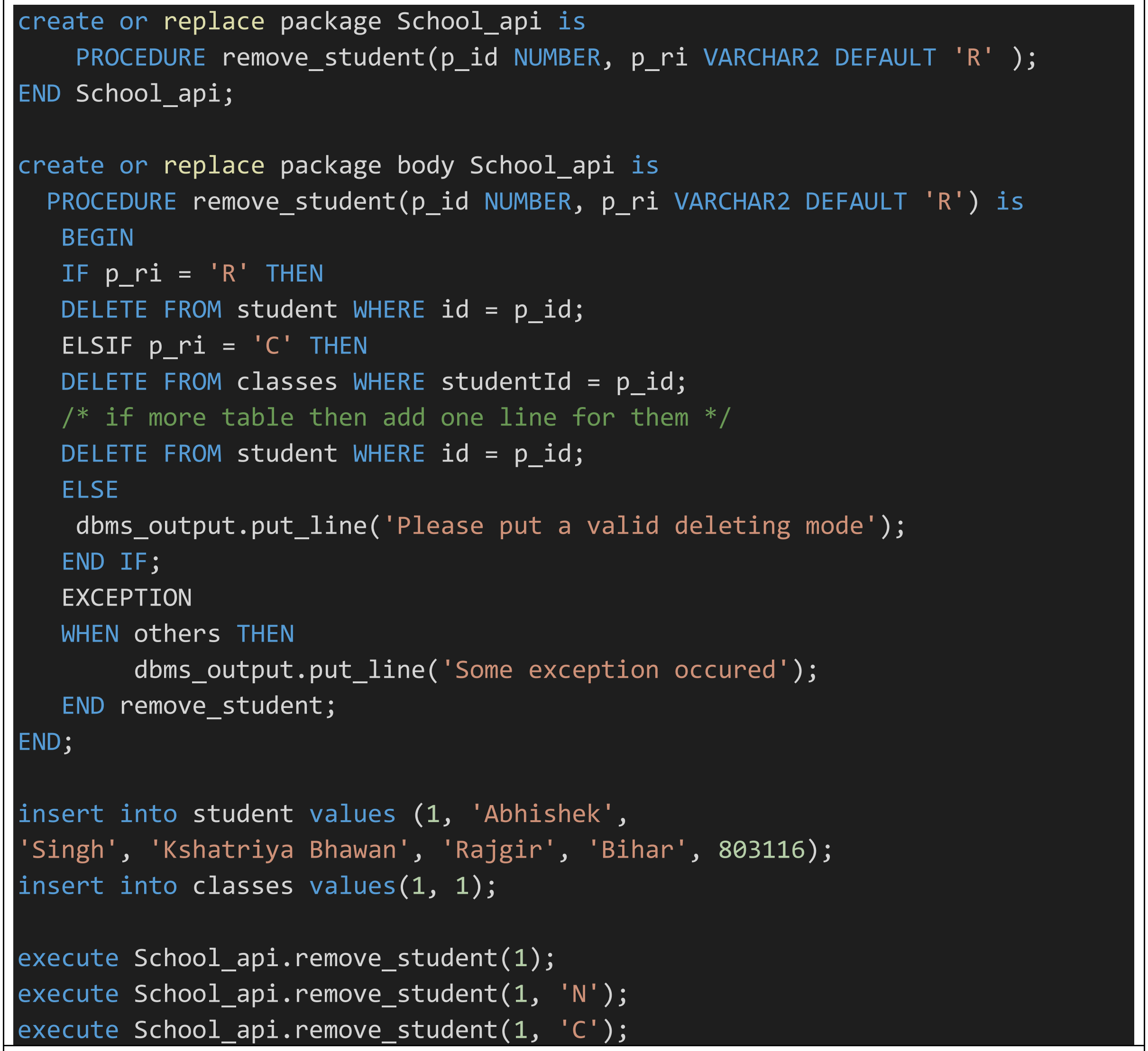
```
create or replace package School_api is
    PROCEDURE remove_student(p_id IN NUMBER);
END School_api;

create or replace package body School_api is
  PROCEDURE remove_student(p_id IN NUMBER)is
   BEGIN
   DELETE FROM student WHERE id = p_id;
   EXCEPTION
   WHEN others THEN
        dbms_output.put_line('Some exception occured');
   END remove_student;
END;

execute School_api.remove_student(1);
```



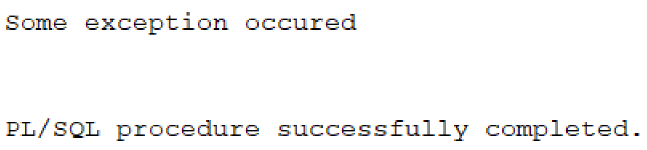
```
Some exception occured


PL/SQL procedure successfully completed.
```

3	Alter ***remove_student*** in the ***school_api*** package body to accept an additional parameter. This new parameter should be a VARCHAR2 and should be called p_ri. Make p_ri default to R. The new parameter may contain a value of R or C. If R is received, it represents DELETE RESTRICT, and the procedure acts as it does now. If

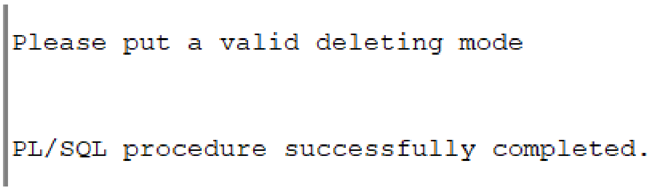
	there are enrollments for the student, the delete is disallowed. If a C is received, it represents DELETE CASCADE. This functionally means that the remove_student procedure locates all records for the student in all the Student Database tables. It removes them from the database before attempting to remove the student from the student table. Decide how to handle the situation when the user passes in a code other than C or R.



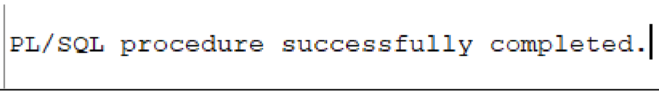
```
create or replace package School_api is
    PROCEDURE remove_student(p_id NUMBER, p_ri VARCHAR2 DEFAULT 'R' );
END School_api;

create or replace package body School_api is
  PROCEDURE remove_student(p_id NUMBER, p_ri VARCHAR2 DEFAULT 'R') is
   BEGIN
   IF p_ri = 'R' THEN
   DELETE FROM student WHERE id = p_id;
   ELSIF p_ri = 'C' THEN
   DELETE FROM classes WHERE studentId = p_id;
   /* if more table then add one line for them */
   DELETE FROM student WHERE id = p_id;
   ELSE
    dbms_output.put_line('Please put a valid deleting mode');
   END IF;
   EXCEPTION
   WHEN others THEN
        dbms_output.put_line('Some exception occured');
   END remove_student;
END;

insert into student values (1, 'Abhishek',
'Singh', 'Kshatriya Bhawan', 'Rajgir', 'Bihar', 803116);
insert into classes values(1, 1);

execute School_api.remove_student(1);
execute School_api.remove_student(1, 'N');
execute School_api.remove_student(1, 'C');
```



```
Some exception occured


PL/SQL procedure successfully completed.
```



```
Please put a valid deleting mode


PL/SQL procedure successfully completed.
```



```
PL/SQL procedure successfully completed.
```

4	Go through the lab manual LOB_lab_Manual.pdf and practice the examples give in the manual.
[redacted]	

5	Go through the lab manual PLSQL_Transactions_Manual.pdf (Only the highlighted chapter- Processing Transactions: pages from 5-40 to 5-50) then practice examples given in the manual.
[redacted]	