IS 420 Assignment 4

Due 4/20

Suppose you have created tables in assignment 1. **Please write PL/SQL program for the following problems. Please include both code and screenshots.**

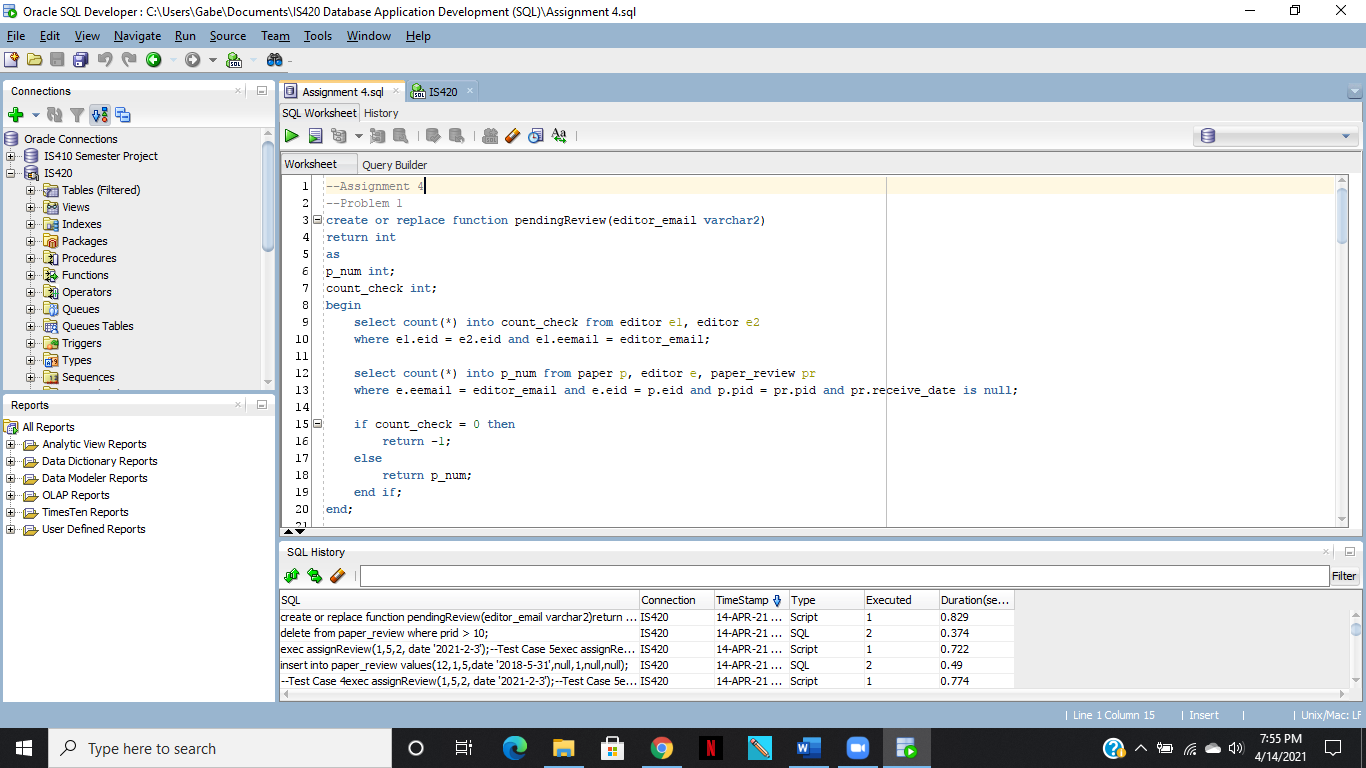
Problem 1: [40 points]

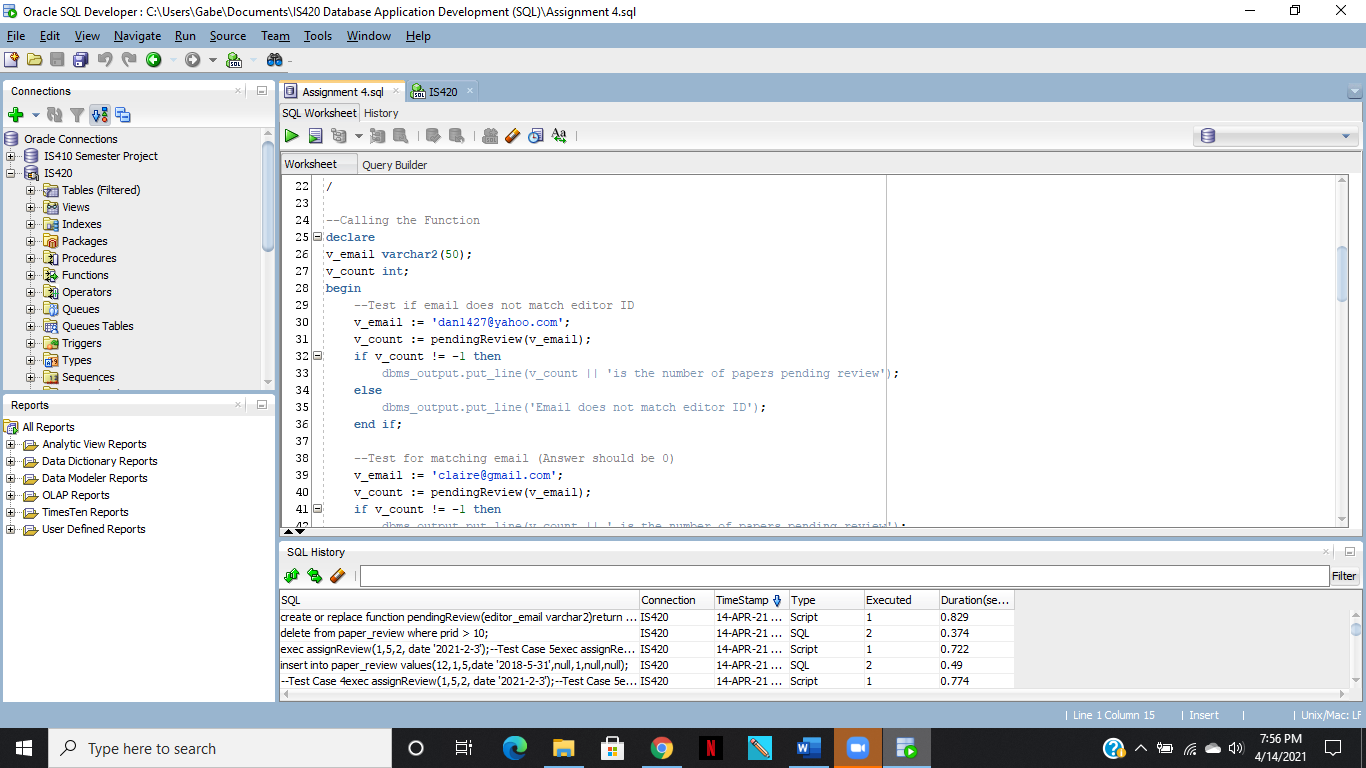
Write a PL/SQL **function** that given email of an editor, compute the number of papers assigned to the editor with pending reviews. A paper with pending review means that the paper has some reviews with NULL receive\_date.

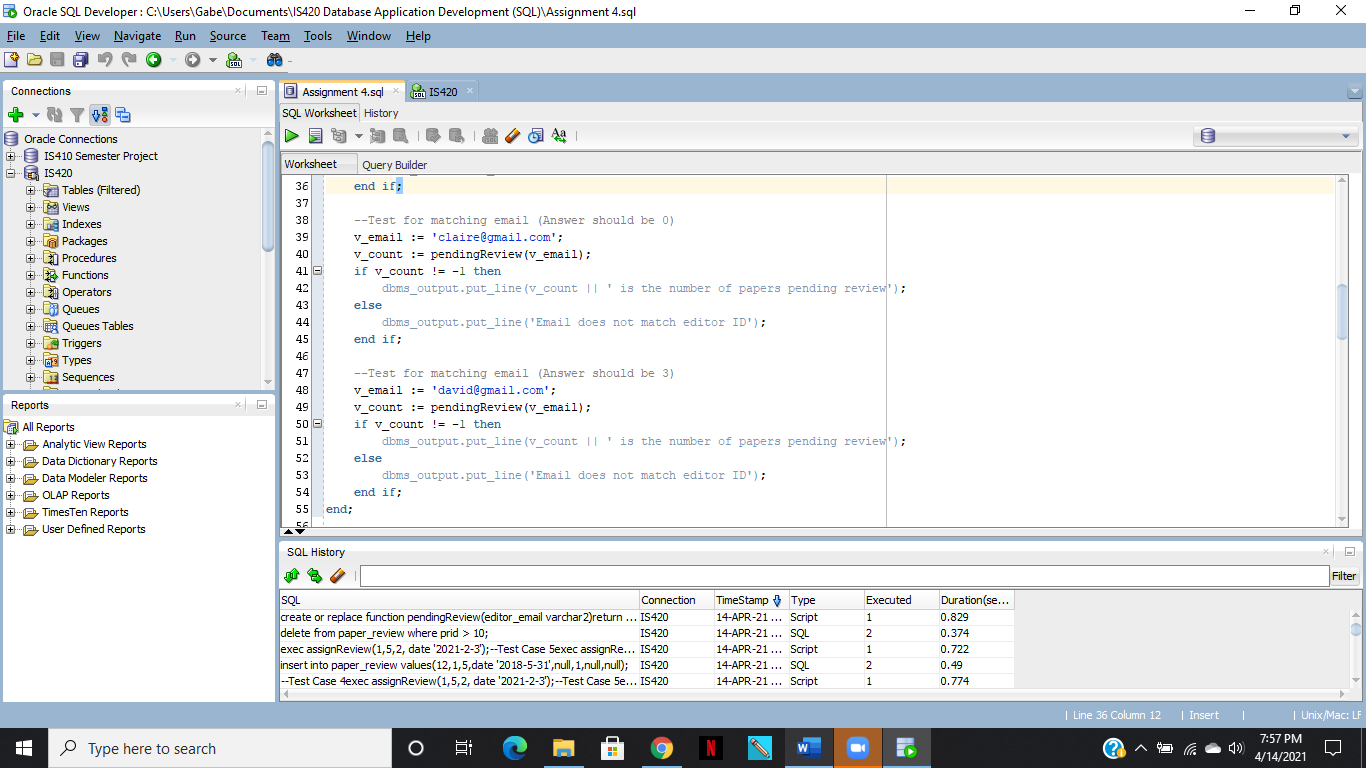
This function also checks a special case when the editor email does not match any existing editors, and returns -1 in this case.

Please also write an anonymous PL/SQL program to call this function with a few different editor emails to test both the regular and special case.

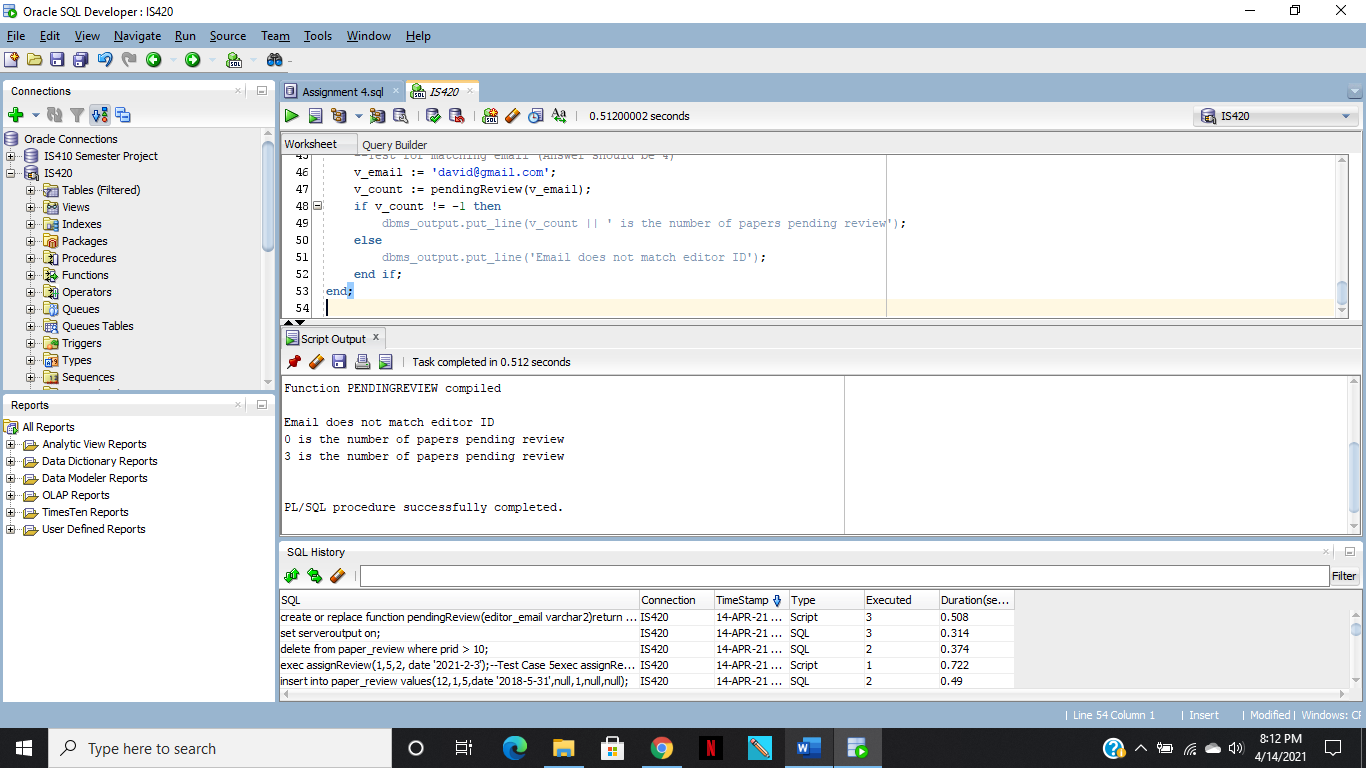
**Screenshots:**







**Results:**



**Code:**

create or replace function pendingReview(editor\_email varchar2)

return int

as

p\_num int;

count\_check int;

begin

select count(\*) into count\_check from editor e1, editor e2

where e1.eid = e2.eid and e1.eemail = editor\_email;

select count(\*) into p\_num from paper p, editor e, paper\_review pr

where e.eemail = editor\_email and e.eid = p.eid and p.pid = pr.pid and pr.receive\_date is null;

if count\_check = 0 then

return -1;

else

return p\_num;

end if;

end;

/

--Calling the Function

declare

v\_email varchar2(50);

v\_count int;

begin

--Test if email does not match editor ID

v\_email := 'dan1427@yahoo.com';

v\_count := pendingReview(v\_email);

if v\_count != -1 then

dbms\_output.put\_line(v\_count || 'is the number of papers pending review');

else

dbms\_output.put\_line('Email does not match editor ID');

end if;

--Test for matching email (Answer should be 0)

v\_email := 'claire@gmail.com';

v\_count := pendingReview(v\_email);

if v\_count != -1 then

dbms\_output.put\_line(v\_count || ' is the number of papers pending review');

else

dbms\_output.put\_line('Email does not match editor ID');

end if;

--Test for matching email (Answer should be 3)

v\_email := 'david@gmail.com';

v\_count := pendingReview(v\_email);

if v\_count != -1 then

dbms\_output.put\_line(v\_count || ' is the number of papers pending review');

else

dbms\_output.put\_line('Email does not match editor ID');

end if;

end;

Problem 2: [60 points] Write a PL/SQL procedure that assigns a reviewer to a paper in a certain round of review. The input includes a paper ID, reviewer ID, round, and due date.

The procedure does ALL of the following:

1) First check whether there is a paper with the given paper ID, if not, print an error message 'Wrong paper ID' and stop;

2) Next check whether there is a reviewer with the input reviewer ID. If not, print an error message 'wrong reviewer ID' and stop.

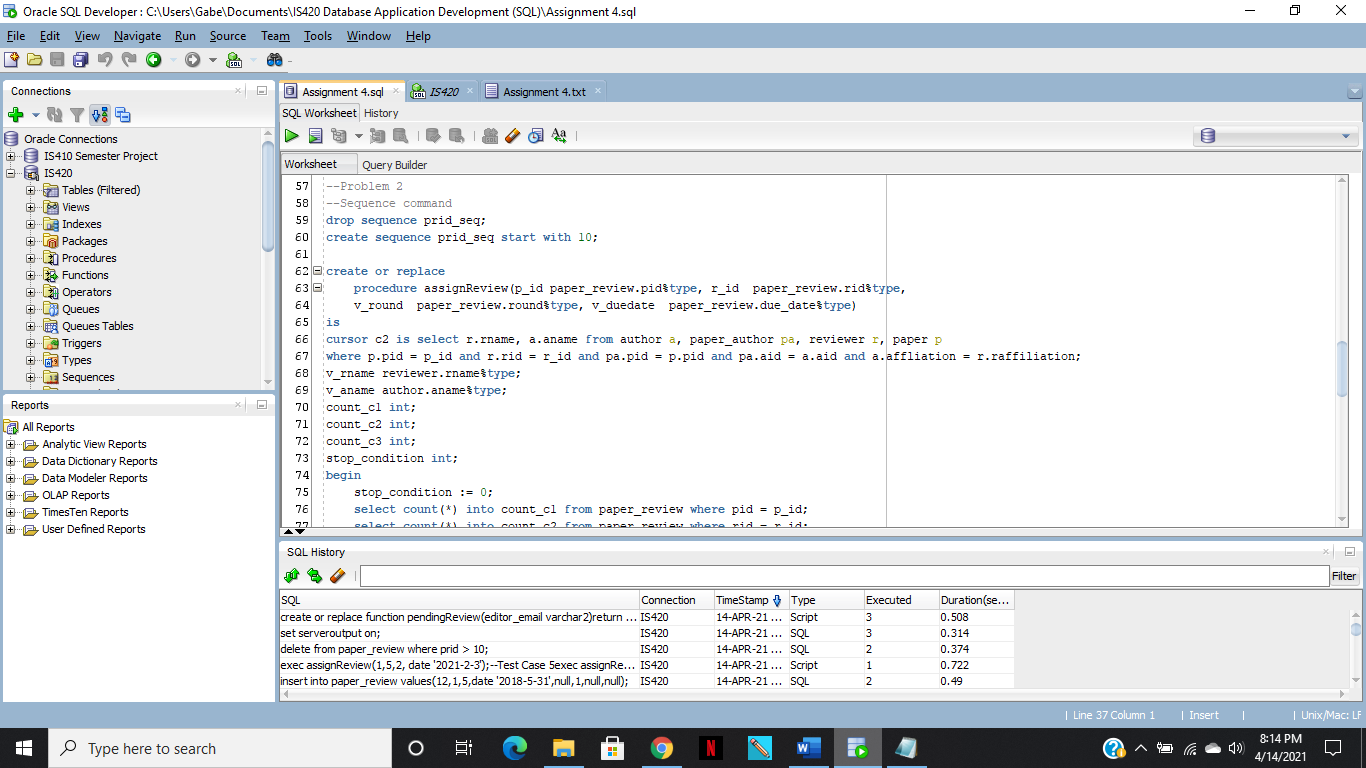
3) Next check whether there is already a row in paper\_review with given paper ID, reviewer ID, and round. If so, print a message 'Already assigned' and stop.

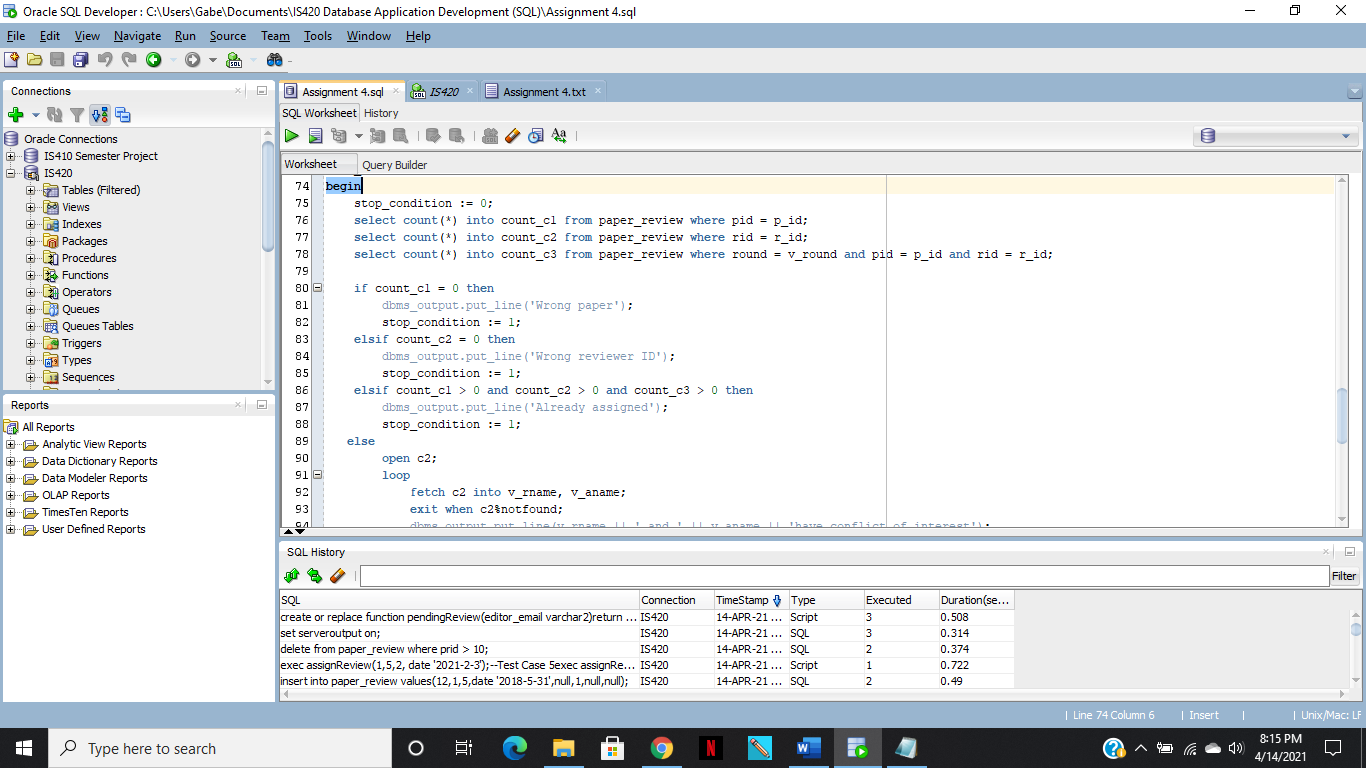
4) Now check whether there is a conflict of interest, i.e., whether any author of the paper and the input reviewer has the same affiliation. If a conflict exists, print a message 'X and Y have conflict of interest' where X and Y are the names of author and reviewer with the same affiliation and stop. Please note that a paper may have multiple authors with the same affiliation with the reviewer.

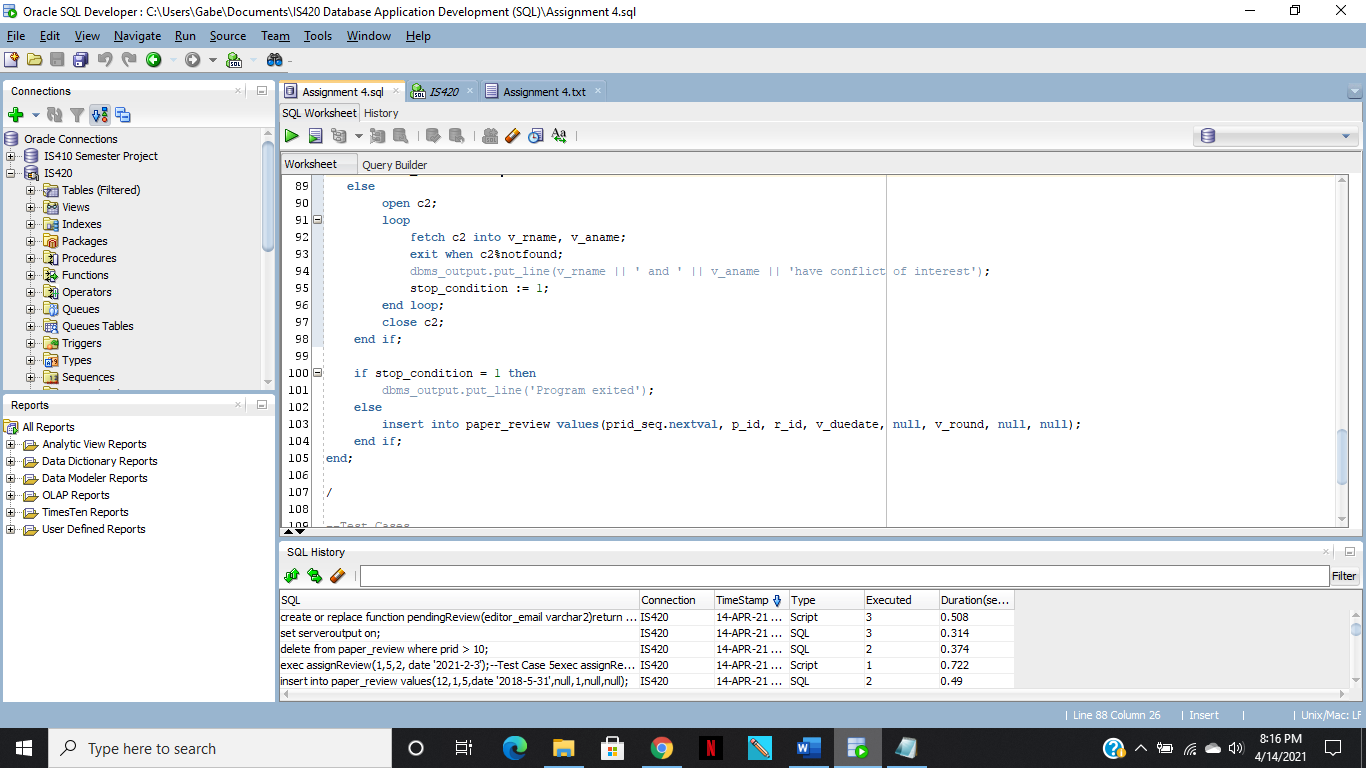
5) If none of the above happens, insert a row to paper\_review with given paper ID, reviewer ID, round, due date, and give null values to receive\_date, decision and comment. Please use a sequence to generate new prid and print out the new prid. Please include the drop sequence and create sequence command before the procedure creation code such that your procedure can be created correctly.

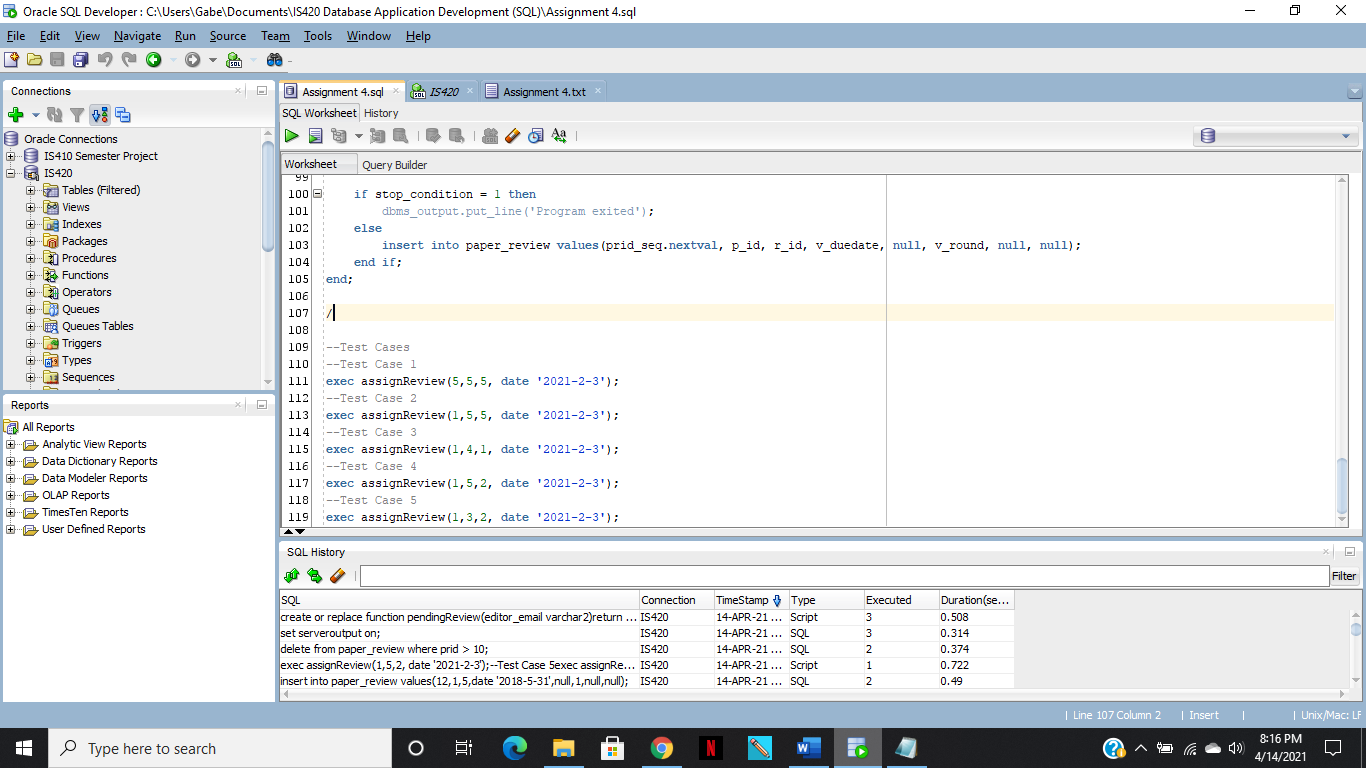
Please also include exec procedure code to test all 5 cases (5 test case needed). You will lose points if not all test cases are included.

**Screenshots:**









**Code:**

--Sequence command

drop sequence prid\_seq;

create sequence prid\_seq start with 10;

create or replace

procedure assignReview(p\_id paper\_review.pid%type, r\_id paper\_review.rid%type,

v\_round paper\_review.round%type, v\_duedate paper\_review.due\_date%type)

is

cursor c2 is select r.rname, a.aname from author a, paper\_author pa, reviewer r, paper p

where p.pid = p\_id and r.rid = r\_id and pa.pid = p.pid and pa.aid = a.aid and a.affliation = r.raffiliation;

v\_rname reviewer.rname%type;

v\_aname author.aname%type;

count\_c1 int;

count\_c2 int;

count\_c3 int;

stop\_condition int;

begin

stop\_condition := 0;

select count(\*) into count\_c1 from paper\_review where pid = p\_id;

select count(\*) into count\_c2 from paper\_review where rid = r\_id;

select count(\*) into count\_c3 from paper\_review where round = v\_round and pid = p\_id and rid = r\_id;

if count\_c1 = 0 then

dbms\_output.put\_line('Wrong paper');

stop\_condition := 1;

elsif count\_c2 = 0 then

dbms\_output.put\_line('Wrong reviewer ID');

stop\_condition := 1;

elsif count\_c1 > 0 and count\_c2 > 0 and count\_c3 > 0 then

dbms\_output.put\_line('Already assigned');

stop\_condition := 1;

else

open c2;

loop

fetch c2 into v\_rname, v\_aname;

exit when c2%notfound;

dbms\_output.put\_line(v\_rname || ' and ' || v\_aname || 'have conflict of interest');

stop\_condition := 1;

end loop;

close c2;

end if;

if stop\_condition = 1 then

dbms\_output.put\_line('Program exited');

else

insert into paper\_review values(prid\_seq.nextval, p\_id, r\_id, v\_duedate, null, v\_round, null, null);

end if;

end;

/

--Test Cases

--Test Case 1

exec assignReview(5,5,5, date '2021-2-3');

--Test Case 2

exec assignReview(1,5,5, date '2021-2-3');

--Test Case 3

exec assignReview(1,4,1, date '2021-2-3');

--Test Case 4

exec assignReview(1,5,2, date '2021-2-3');

--Test Case 5

exec assignReview(1,3,2, date '2021-2-3');