DEPARTMENT OF INFORMATION TECHNOLOGY

**Academic Year: 2023 - 24**

**COURSE CODE: DJS22ITL303**

**COURSE NAME: Database Management Systems**

**NAME: Anish Sharma**

EXPERIMENT NO:7

To implement Triggers

Q1

i)

CREATE TABLE employee

(

empno integer,

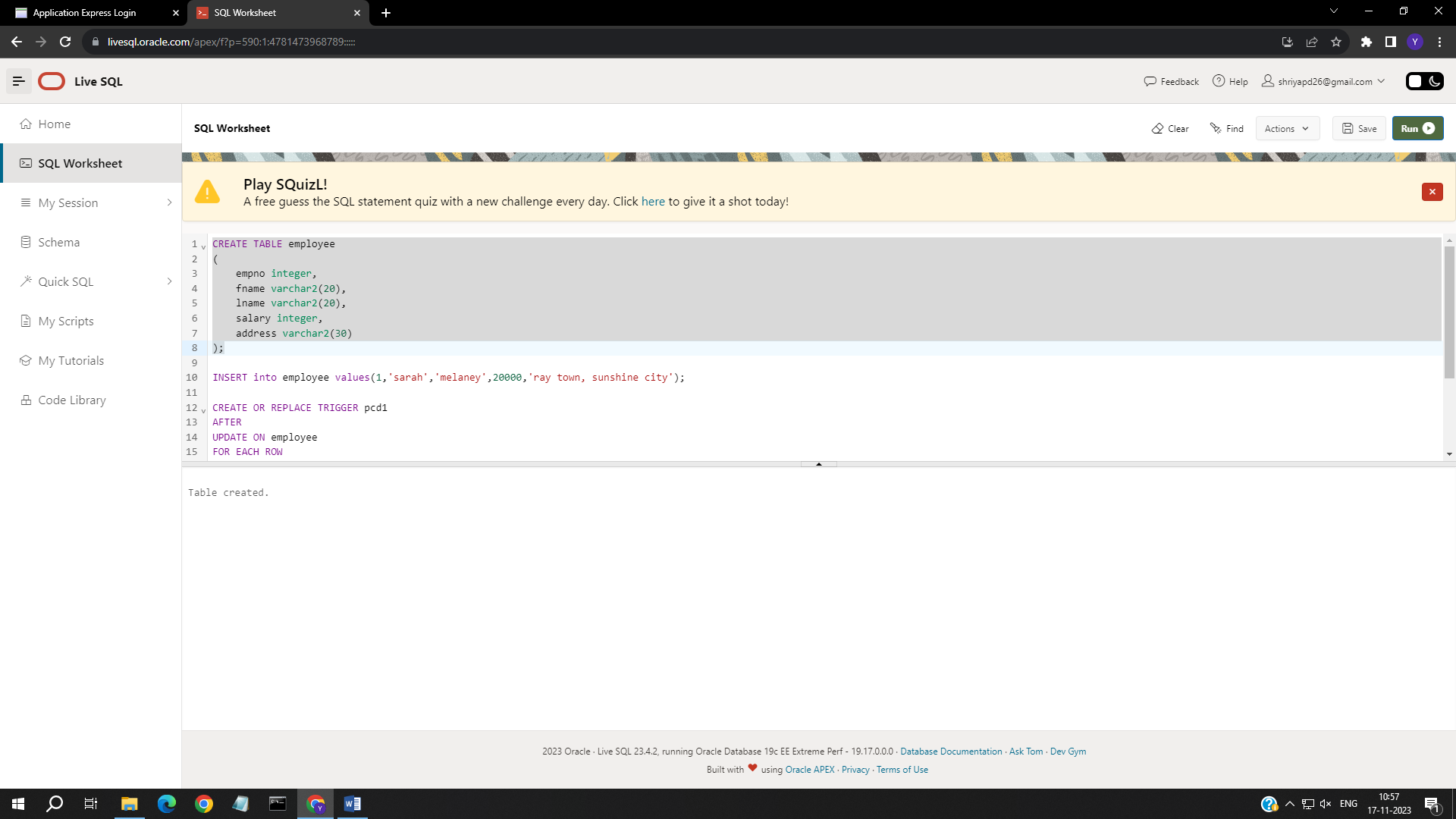
fname varchar2(20),

lname varchar2(20),

salary integer,

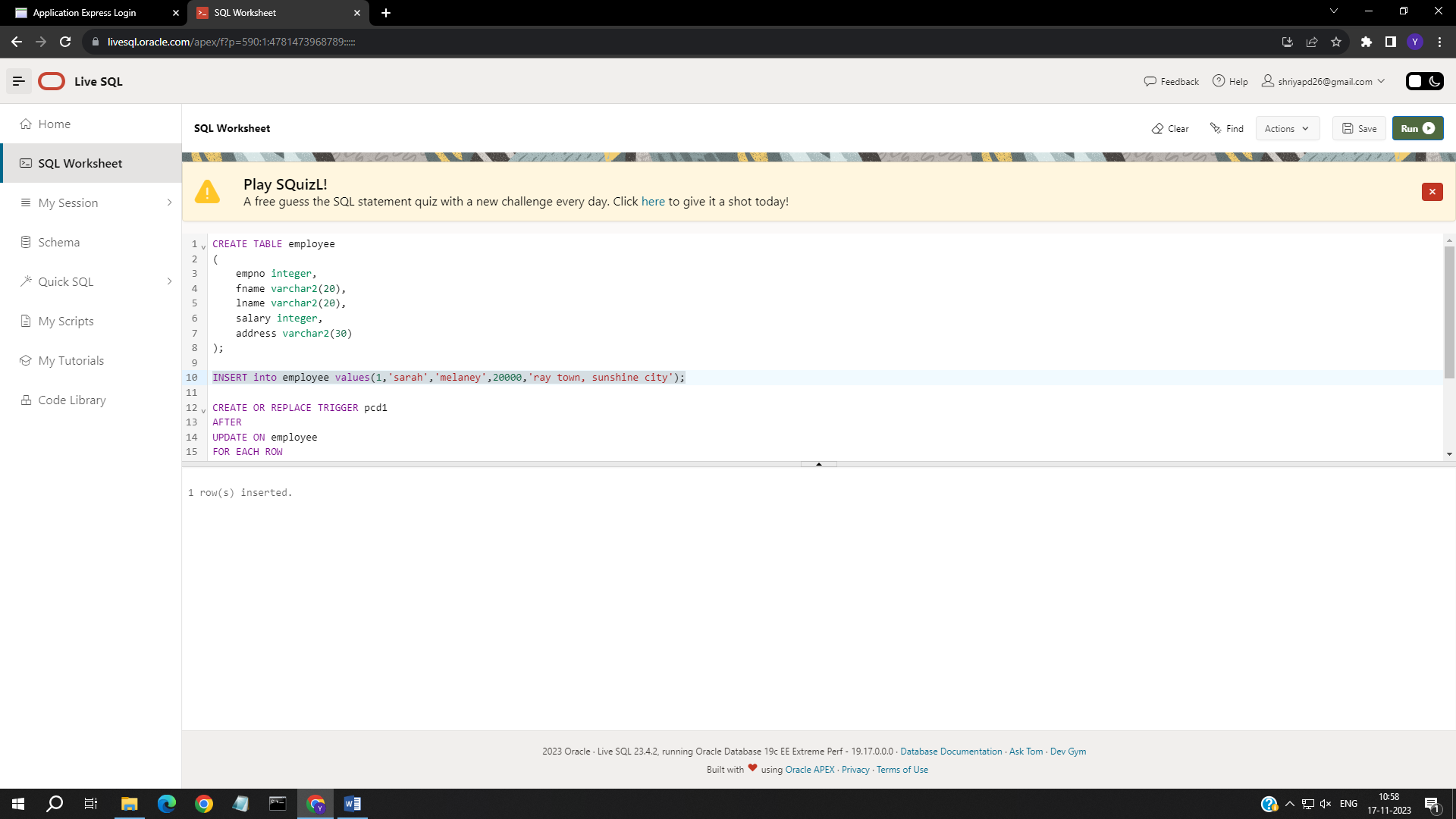
address varchar2(30)

);



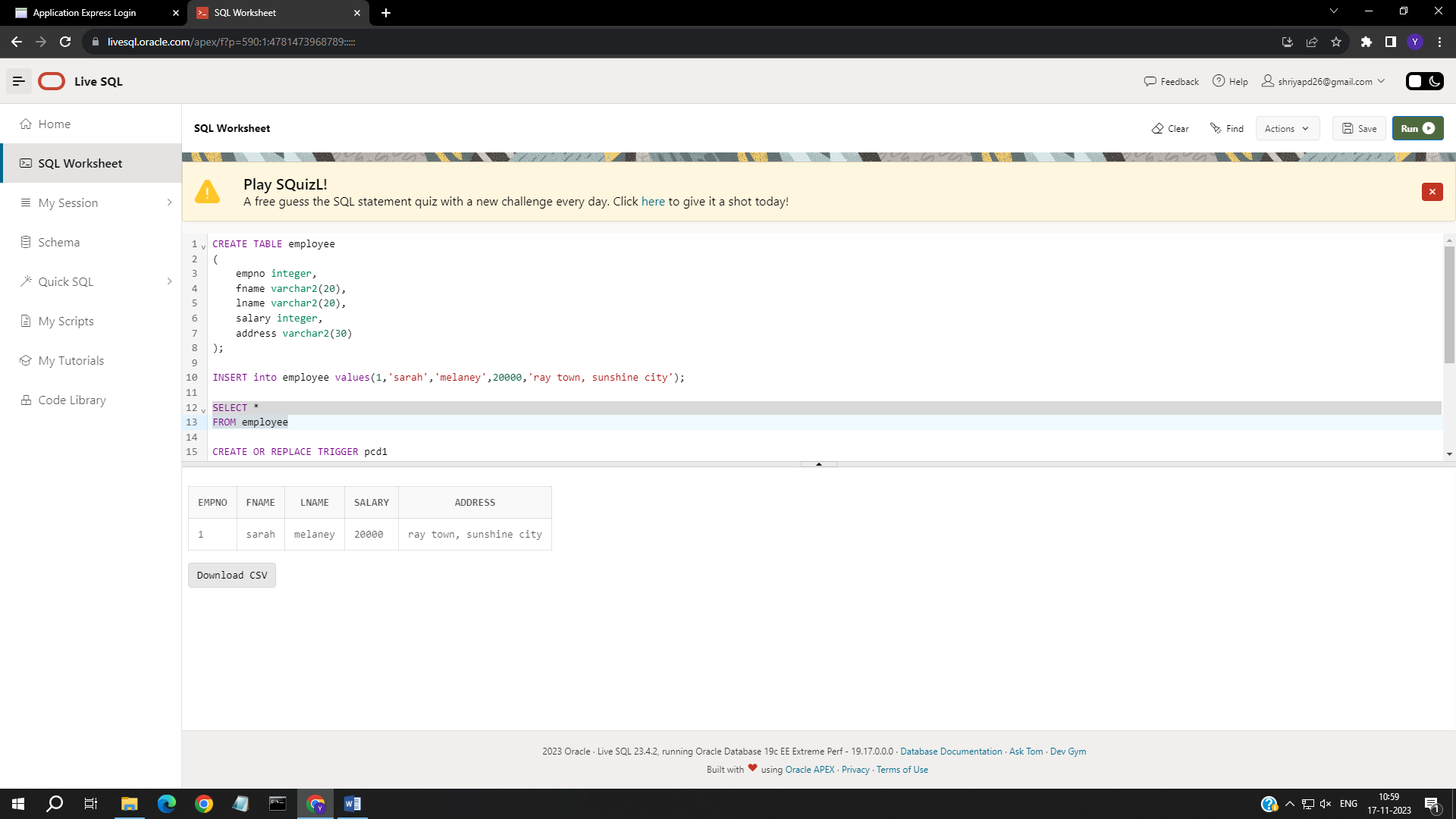
ii)

INSERT into employee values(1,'sarah','melaney',20000,'ray town, sunshine city');



SELECT \*

FROM employee



**CLASS: S. Y. B. Tech. Sem III (I1)**

**SAP ID:60003220045**

**DATE: 3rd December 2023**

iii)

CREATE OR REPLACE TRIGGER pcd1

AFTER

UPDATE ON employee

FOR EACH ROW

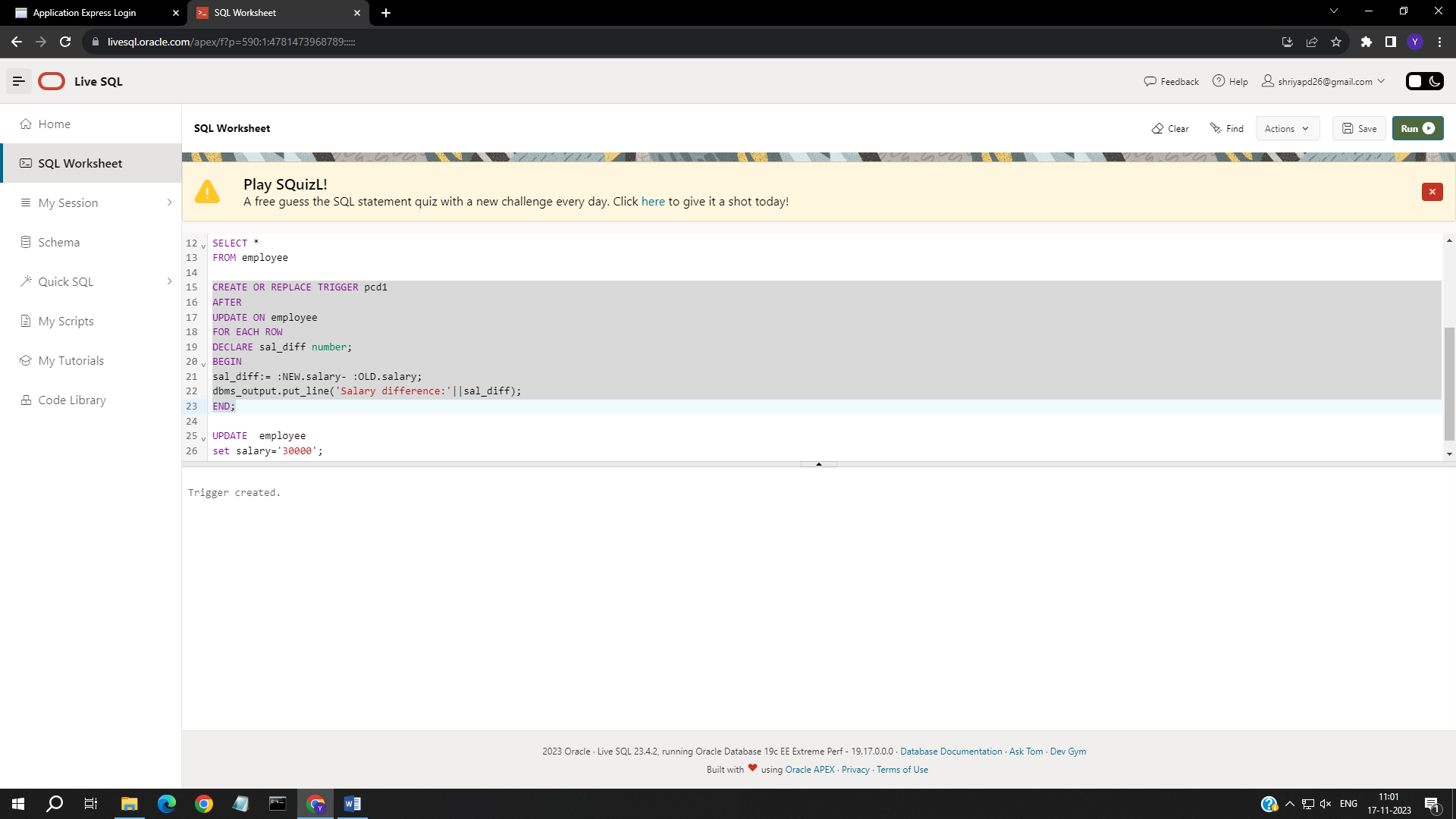
DECLARE sal\_diff number;

BEGIN

sal\_diff:= :NEW.salary- :OLD.salary;

dbms\_output.put\_line('Salary difference:'||sal\_diff);

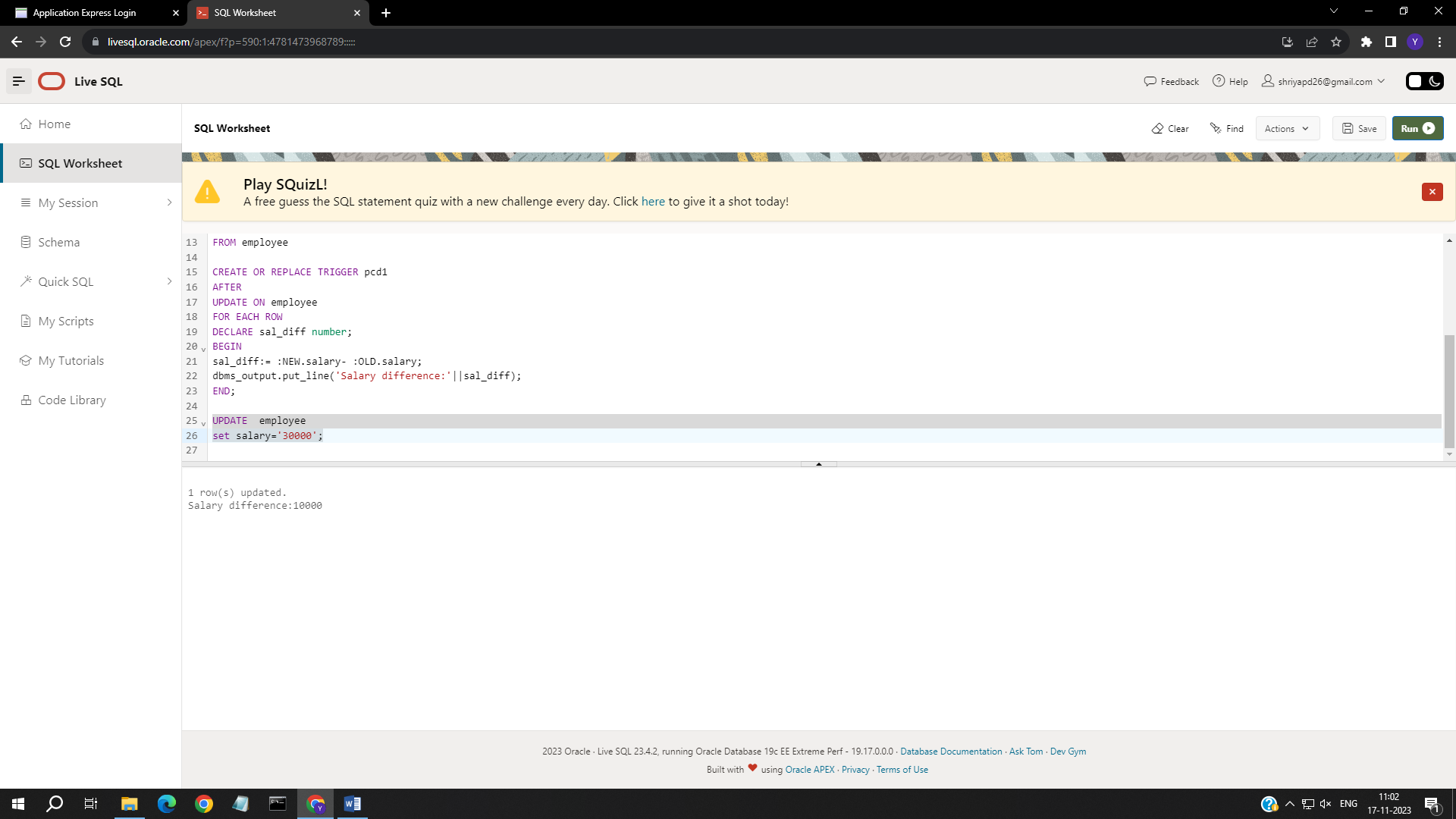
END;



iv) && v)

UPDATE employee

set salary='30000';



Q2

CREATE TABLE GRADES

(

SENO number,

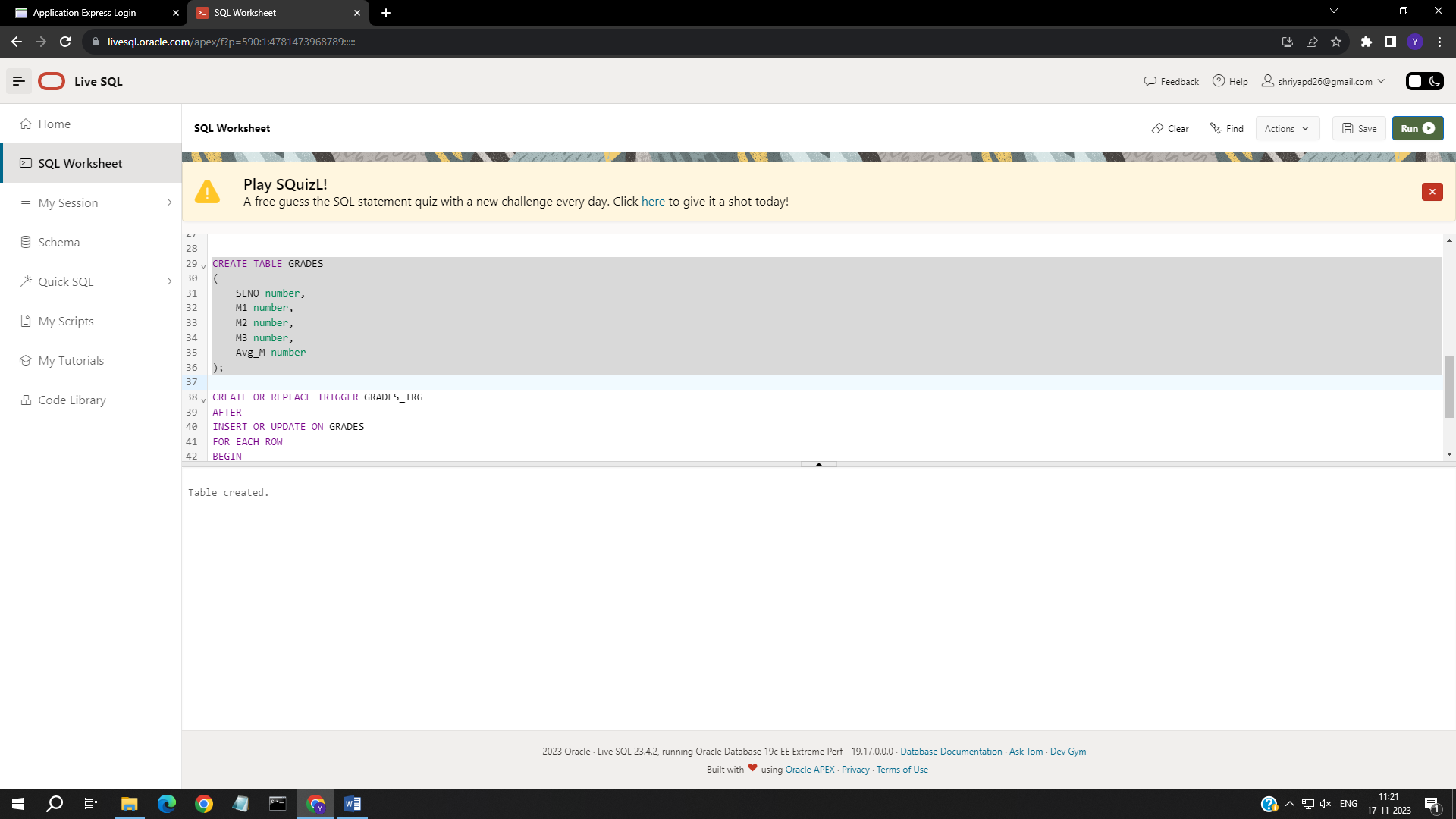
M1 number,

M2 number,

M3 number,

Avg\_M number

);



a)

CREATE OR REPLACE TRIGGER GRADES\_TRG

AFTER

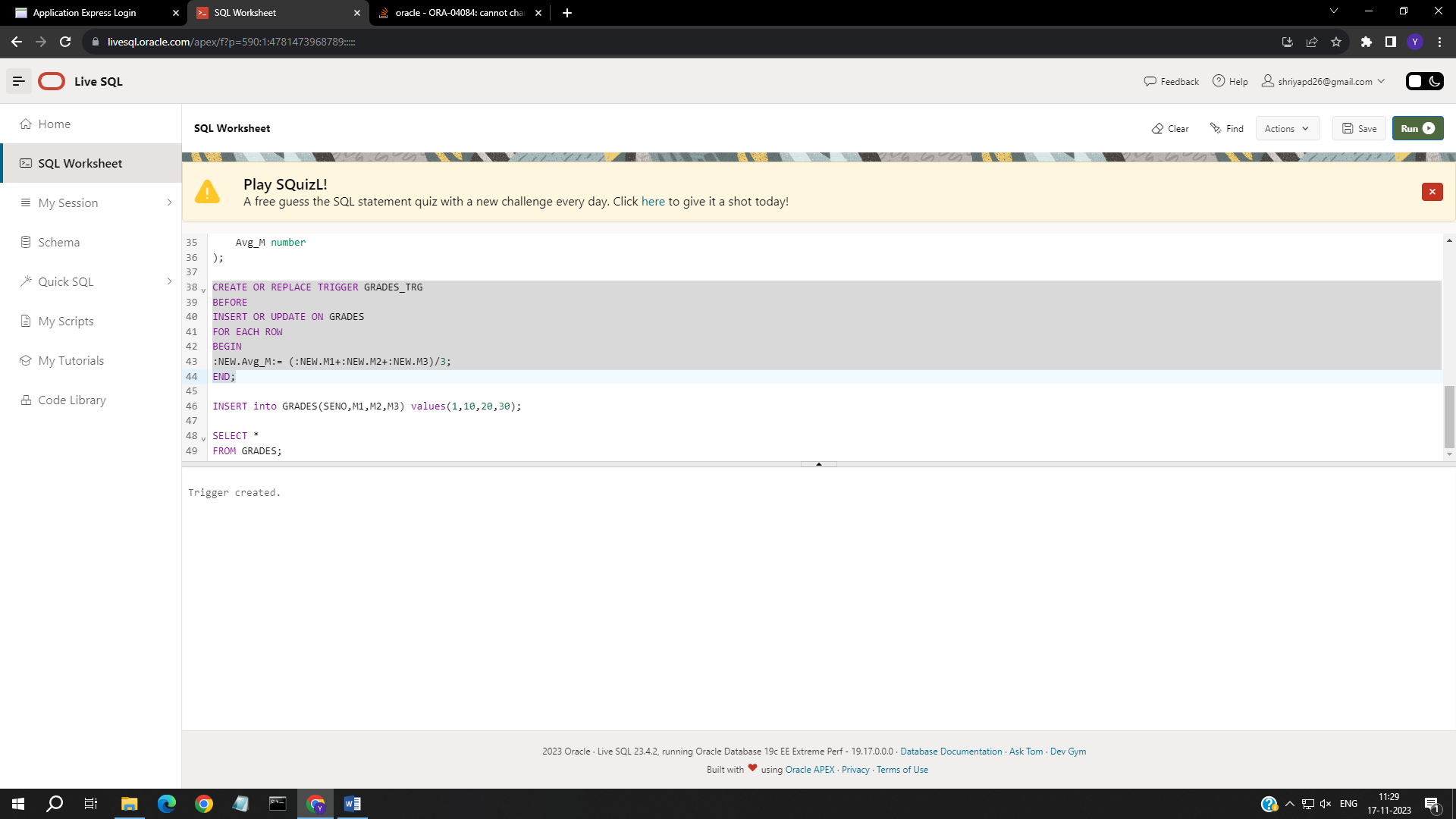
INSERT OR UPDATE ON GRADES

FOR EACH ROW

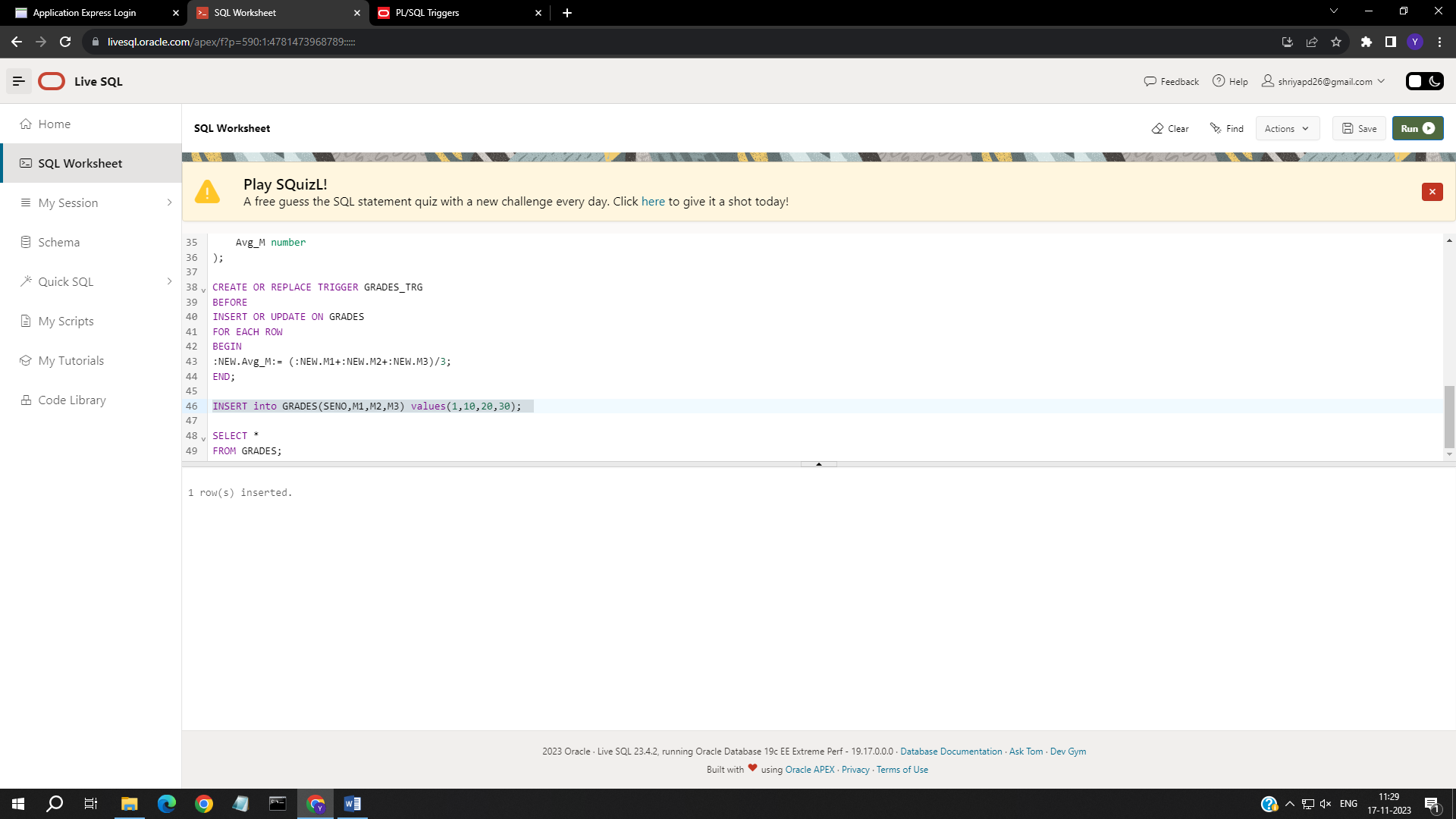
BEGIN

:NEW.Avg\_M:= (:NEW.M1+:NEW.M2+:NEW.M3)/3;

END;

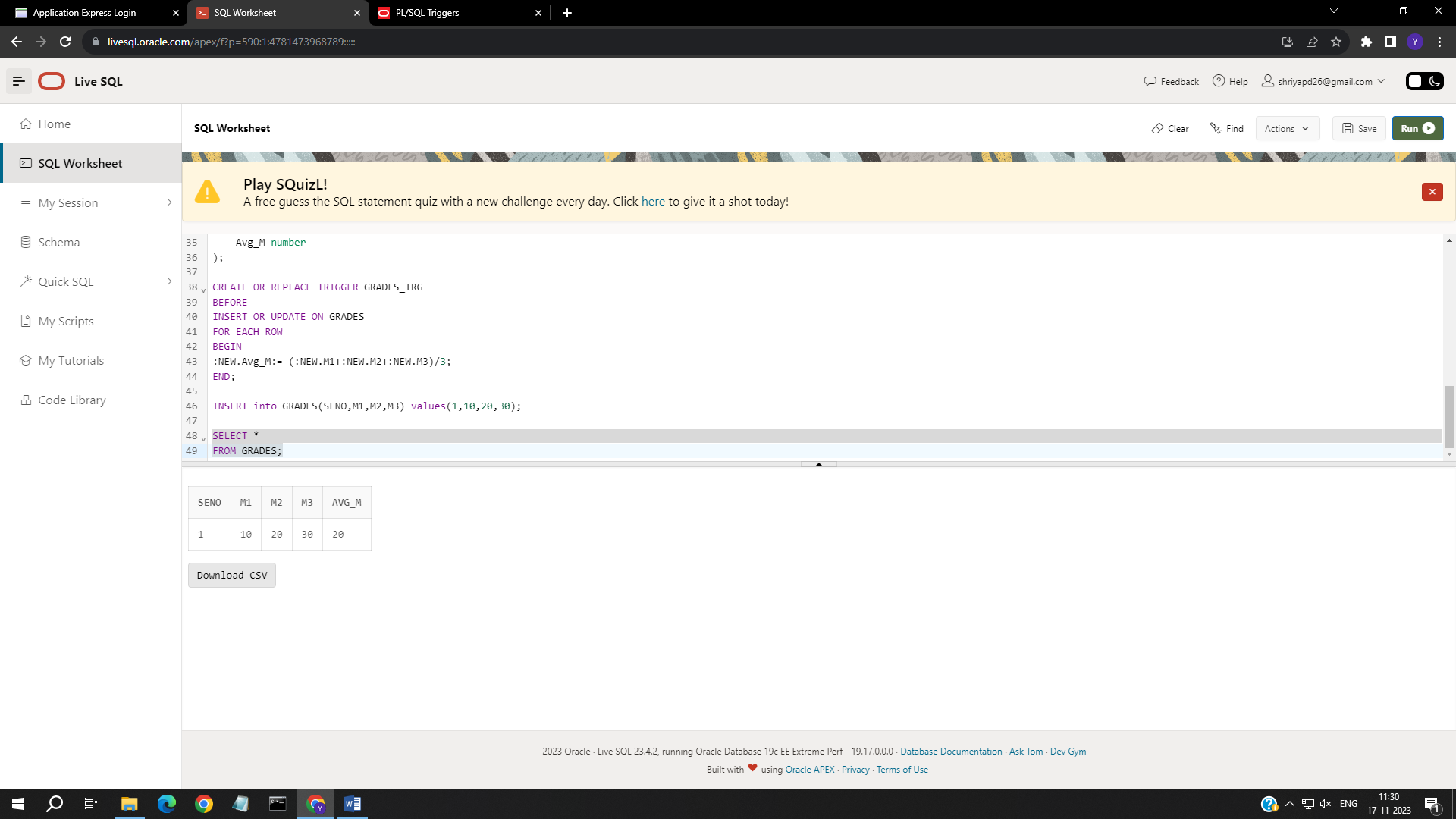


INSERT into GRADES(SENO,M1,M2,M3) values(1,10,20,30);



SELECT \*

FROM GRADES;



b)

CREATE OR REPLACE TRIGGER GRADES\_TRG\_RES

BEFORE

INSERT OR UPDATE ON GRADES

FOR EACH ROW

DECLARE duplicate\_count number;

BEGIN

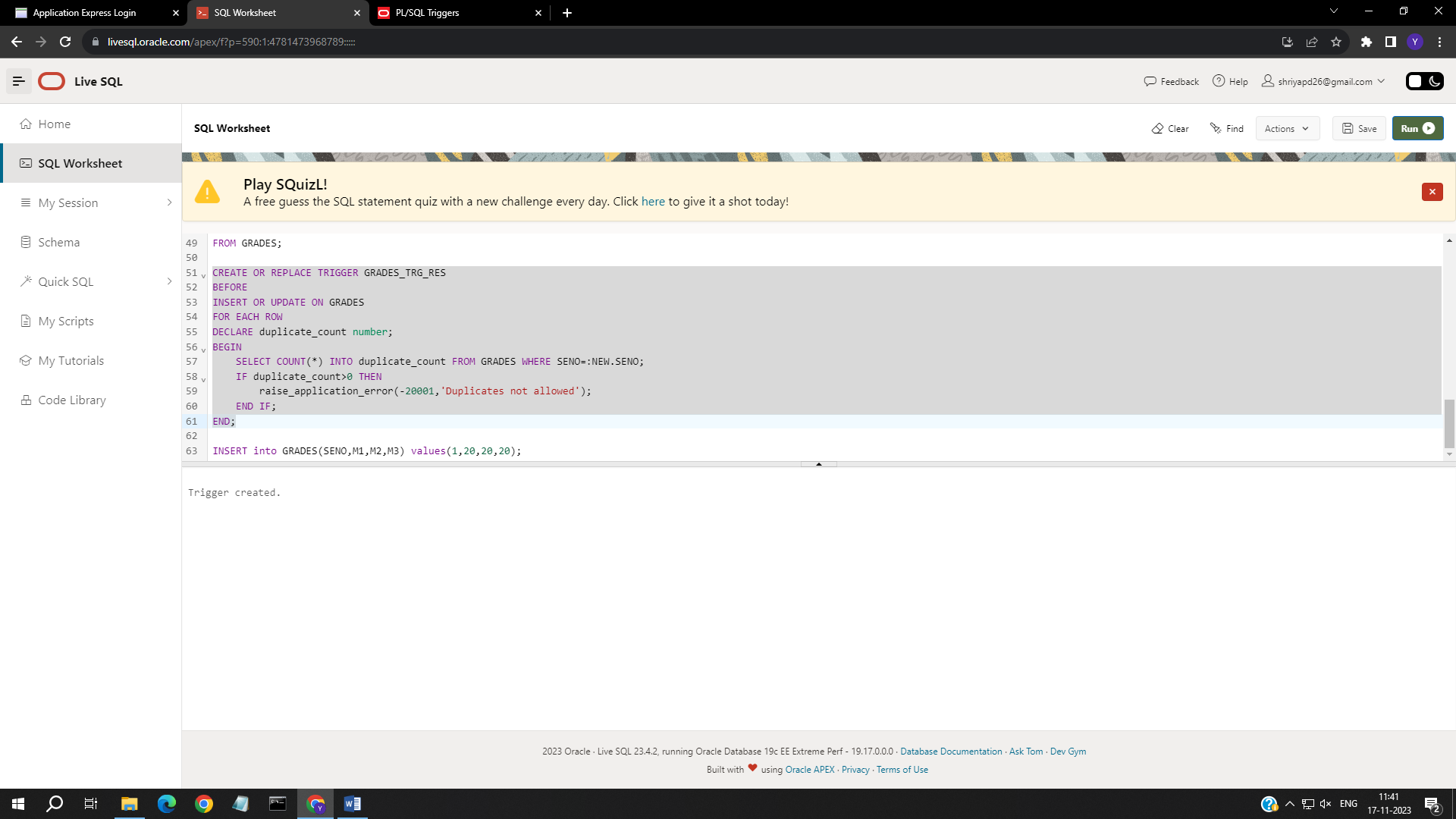
SELECT COUNT(\*) INTO duplicate\_count FROM GRADES WHERE SENO=:NEW.SENO;

IF duplicate\_count>0 THEN

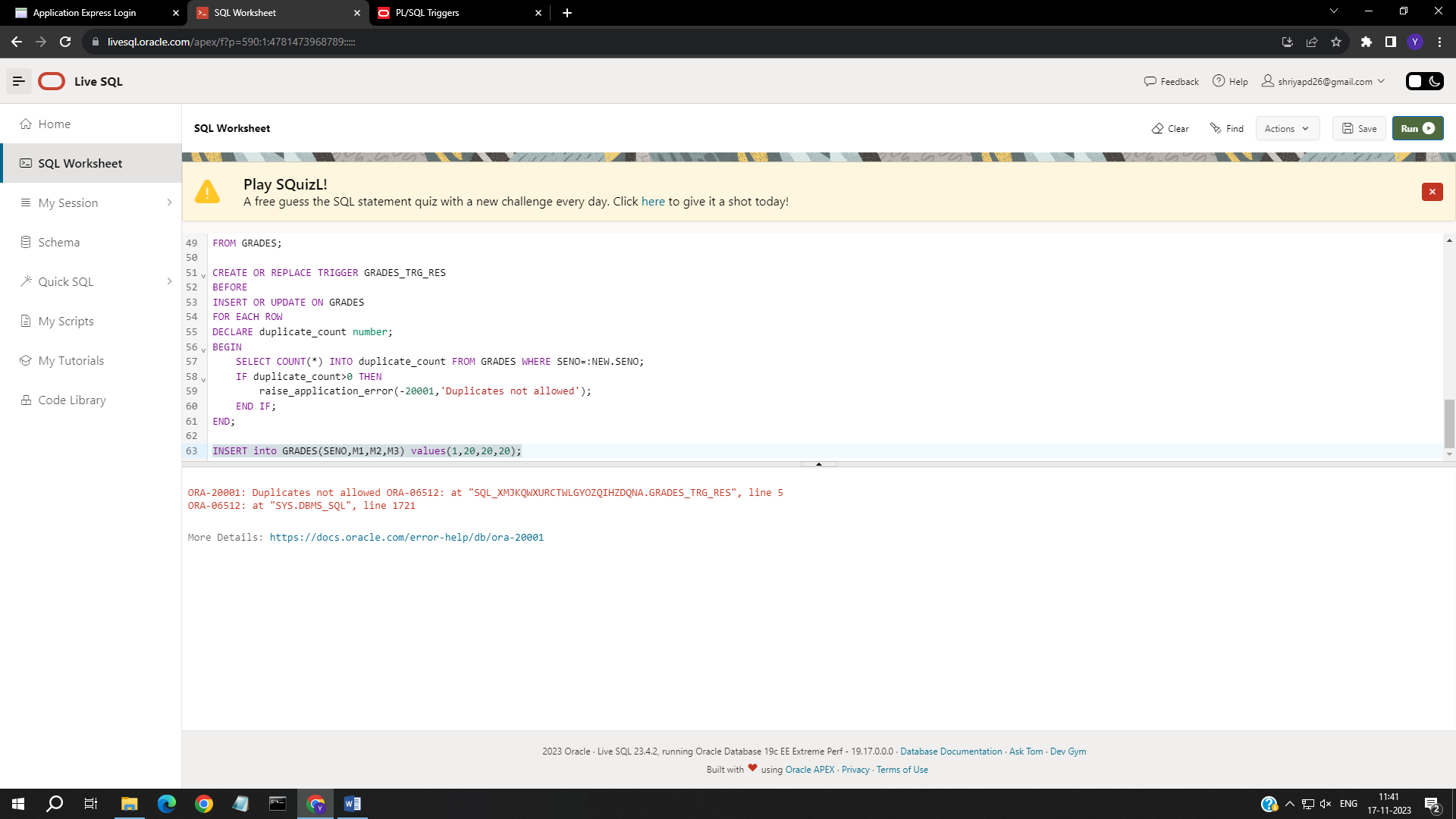
raise\_application\_error(-20001,'Duplicates not allowed');

END IF;

END;



INSERT into GRADES(SENO,M1,M2,M3) values(1,20,20,20);



# EXPERIMENT NO. 8

# To Study and Implement TCL Commands

Implement the SQL statements for the following questions 1)

create table Employeee (

first\_name varchar(20), last\_name varchar(20), salary integer,

dept varchar(20), dob Date,

city varchar(20)

);



2)

insert into Employeee values('Adam', 'Smith', 40000, 'ACCOUNTS', 'jul-23-1990' ,'Chicago'); insert into Employeee values('Banda', 'Correl', 45000, 'HR', 'dec-05-2093' ,'New York');

insert into Employeee values('Carol', 'Holmes', 35000, 'ADMIN', 'jan-10-1989' ,'Los Angeles');

insert into Employeee values('Greene', 'Breckneur', 50000, 'TECHNICAL', 'may-19-1995' ,'Amsterdam'); insert into Employeee values('Tom', 'Johnsohn', 55000, 'ACCOUNTS', 'jan-20-1998' ,'Florida');



3)

Begin savepoint A;

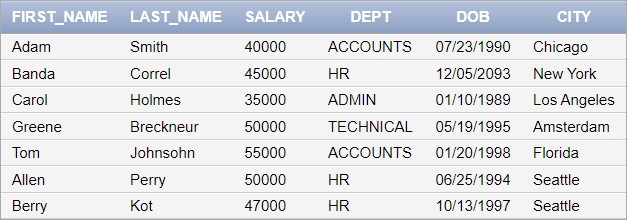
insert into Employeee values('Allen', 'Perry', 50000, 'HR', 'june-25-1994' ,'Seattle'); End;

Begin savepoint B;

insert into Employeee values('Berry', 'Kot', 47000, 'HR', 'oct-13-1997' ,'Seattle'); End;

select \*

from Employeee;

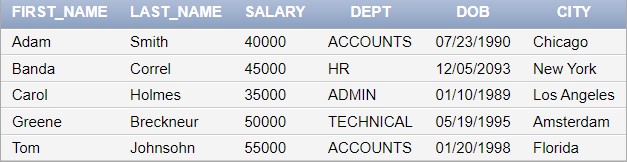


4)

Begin Rollback to A; End;

select \*

from Employeee;



5)

commit;



6)

Begin savepoint P;

insert into Employeee values('John', 'Half', 39000, 'HR', 'nov-27-1992' ,'New York'); End;



7)

Begin savepoint C;

update Employeee set last\_name= 'Paul'

where first\_name = 'Banda'; End;

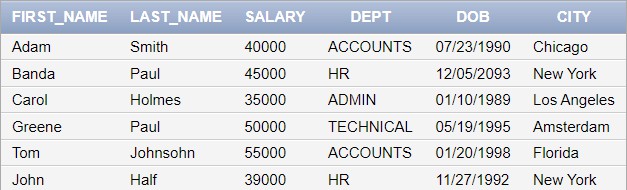
Begin savepoint D;

update Employeee set last\_name= 'Paul'

where first\_name = 'Greene'; End;

select \*

from Employeee;



8)

Begin

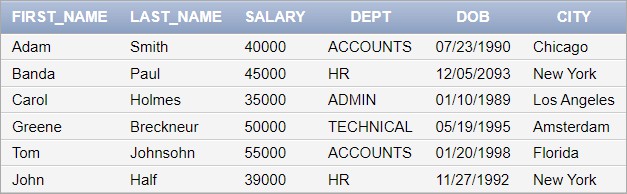
Rollback to D;

End;



select \*

from Employeee;



9)

Begin

savepoint E;

update Employeee

set salary=salary\*1.05;

End;

Begin savepoint F;

update Employeee

set salary=salary\*1.05;

End;

Begin savepoint G;

update Employeee

set salary=salary\*1.05;

End;

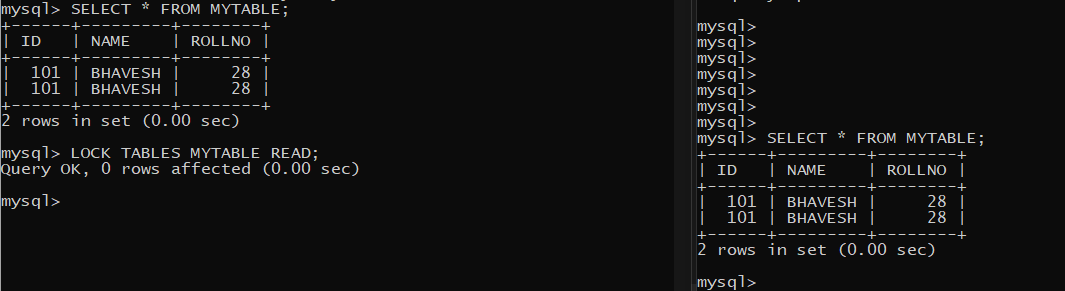
Select SUM(salary) from Employeee;



# EXPERIMENT NO. 9

1.

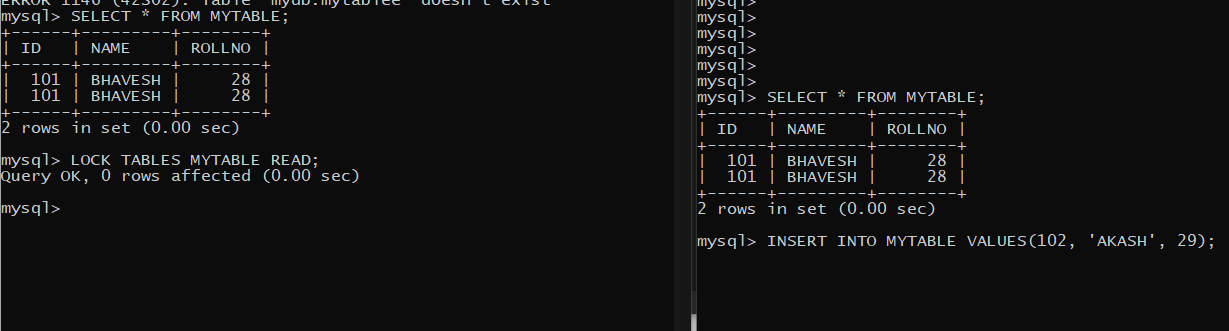
When user 1 locks in read mode, user 2 can read without any conflict but cannot write as read – write is a conflict.

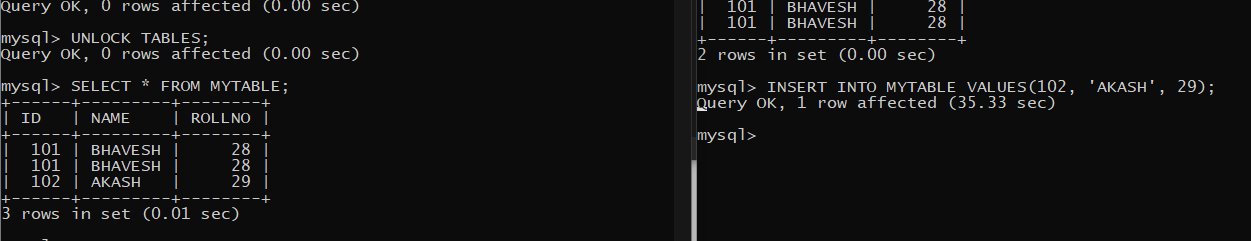
User 2 can read safely.

2.

Read – write conflict

When user 1 is locked in read mode, user 2 cannot perform write operations.



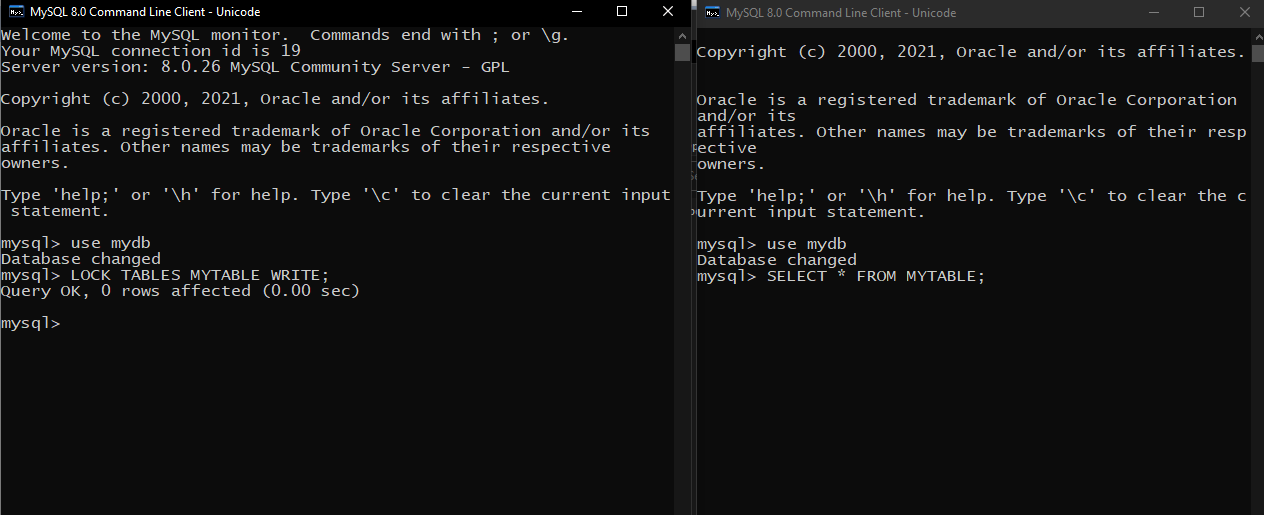


User 2 waits until the lock is removed by user 1.

Then user 2 successfully performs write operation.

Similarly, if user 1 is locked in write mode, user 2 cannot perform read operations.

# User 1 in write mode.

user 2 trying to write.

But it needs to wait for lock to be removed.

3.

Write – Write conflict

Both users cannot write on the same resource at the same time as it is a conflict.

