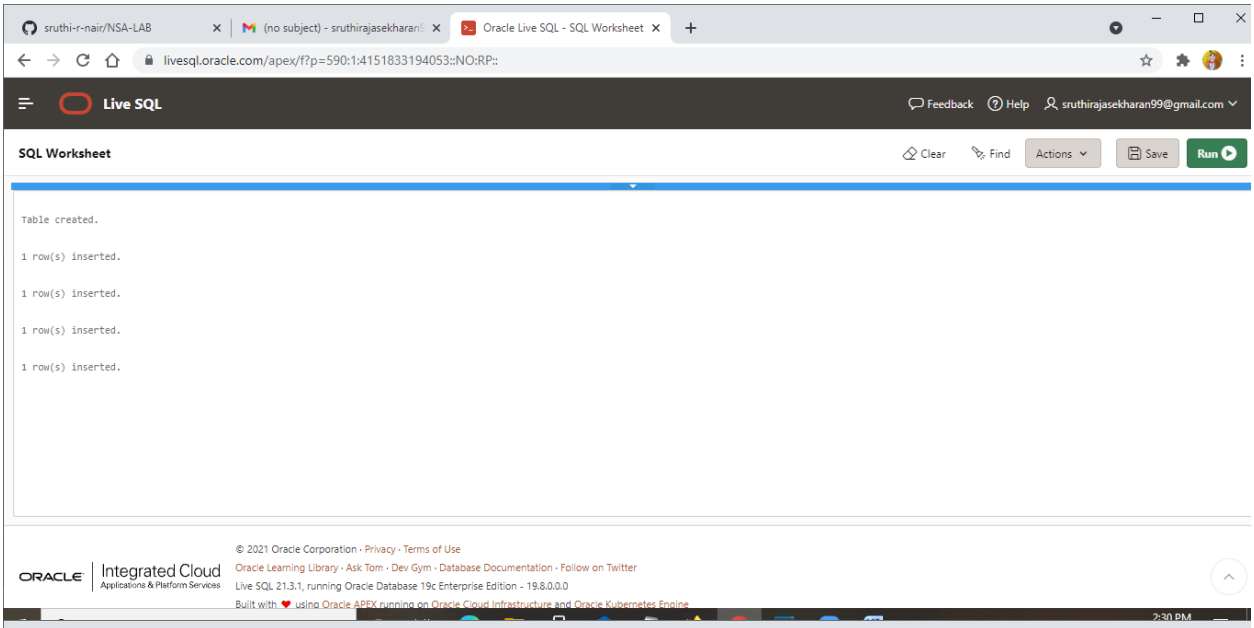
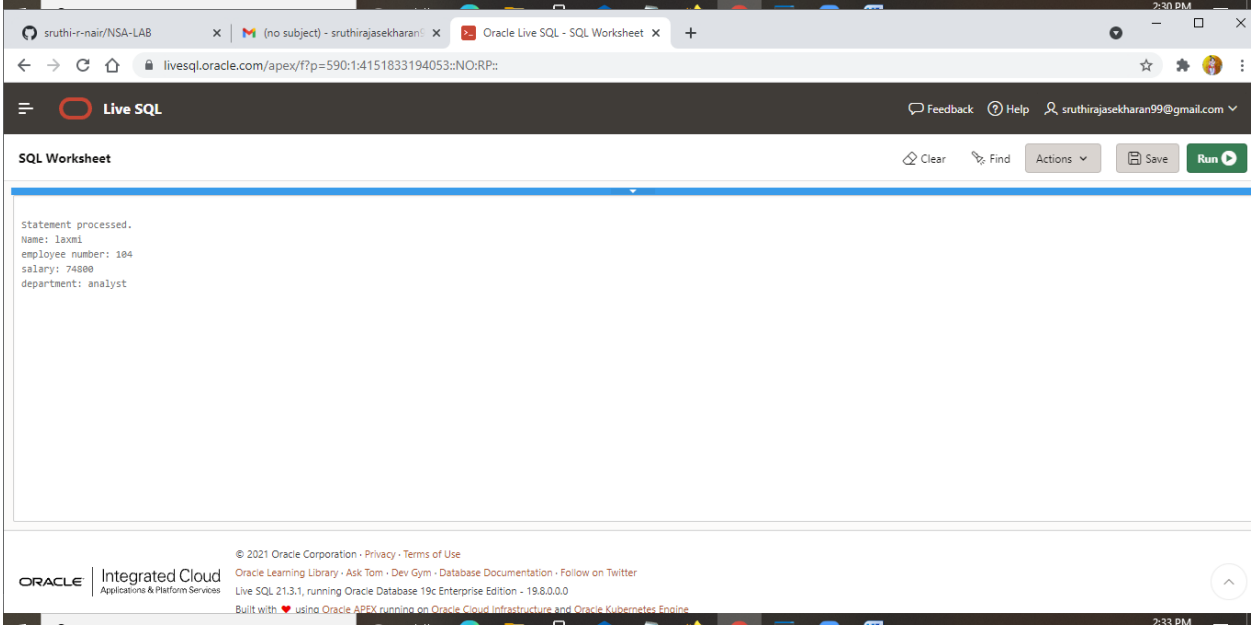
6)create table emp(emp_no int,emp_name varchar(20),salary int,emp_dpt varchar(20));
insert into emp values(101,'ammu',50000,'salesman');
insert into emp values(103,'appu',70000,'manager');
insert into emp values(102,'anju',64000,'clerk');
insert into emp values(104,'laxmi',68000,'analyst');

```
CREATE OR REPLACE PROCEDURE increSalary
IS
emp1 emp%rowtype;
sal emp.salary%type;
dpt emp.emp_dpt%type;
BEGIN
SELECT salary,emp_dpt INTO sal,dpt FROM emp WHERE emp_no = 104;
   IF dpt ='clerk' THEN
     UPDATE emp SET salary = salary+salary* 5/100 ;
   ELSIF dpt = 'salesman' THEN
     UPDATE emp SET salary = salary+salary* 7/100  ;
   ELSIF dpt = 'analyst' THEN
     UPDATE emp SET salary = salary+salary* 10/100  ;
  ELSIF dpt = 'manager' THEN
     UPDATE emp SET salary = salary+salary* 20/100  ;
  ELSE
     DBMS_OUTPUT.PUT_LINE ('NO INCREMENT');


  END IF;
   SELECT * into emp1 FROM emp WHERE emp_no = 104;
   DBMS_OUTPUT.PUT_LINE ('Name: '||emp1.emp_name);
   DBMS_OUTPUT.PUT_LINE ('employee number: '||emp1.emp_no);
   DBMS_OUTPUT.PUT_LINE ('salary: '|| emp1.salary);
   DBMS_OUTPUT.PUT_LINE ('department: '|| emp1.emp_dpt);
END;

DECLARE
BEGIN
  increSalary();
END;
```

Output:



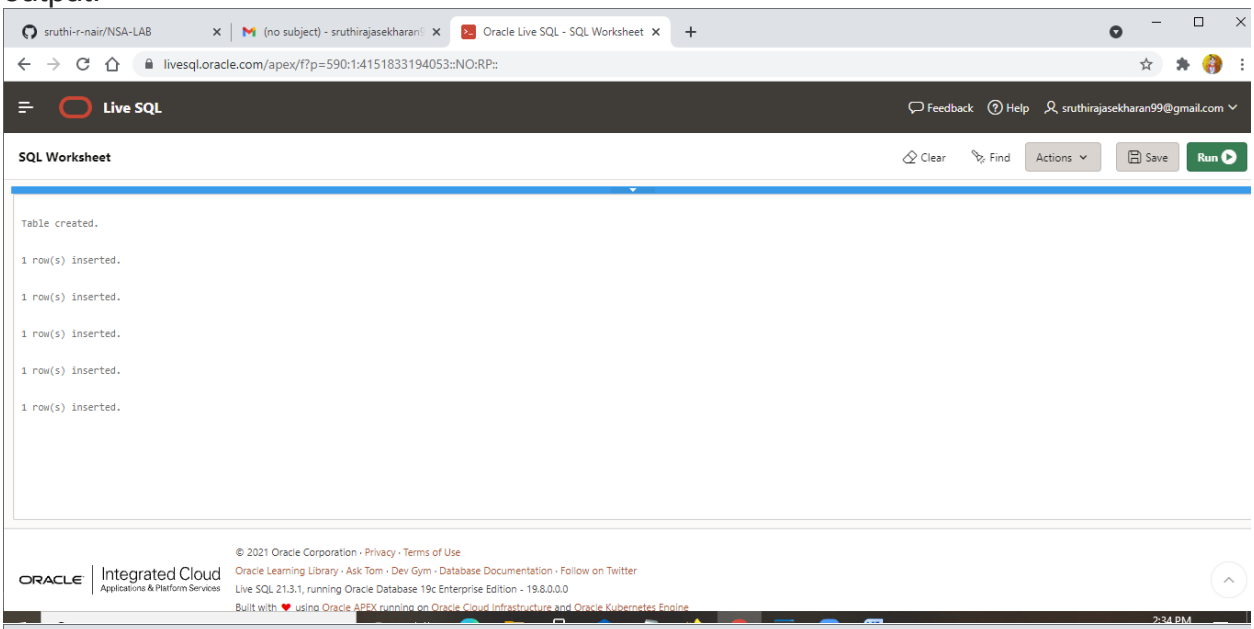


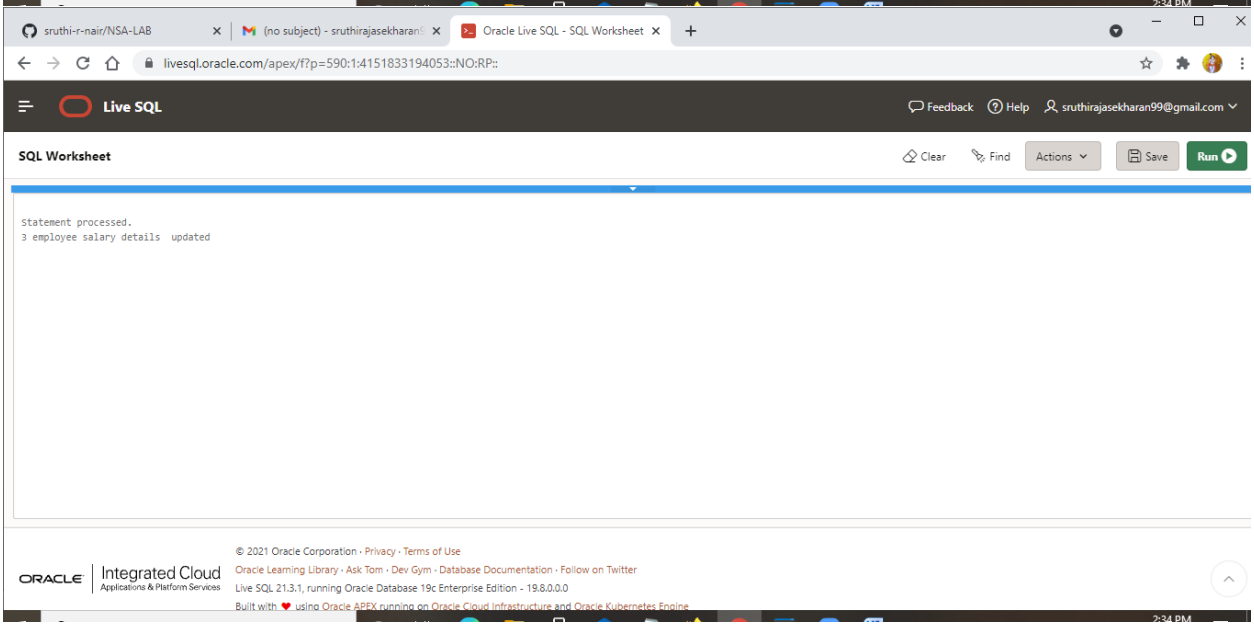
7)create table empy(emp_no int,emp_name varchar(20),salary int,emp_dpt varchar(20),dsgt varchar(20));
insert into empy values(201,'Jack',50000,'sales','president');
insert into empy values(202,'Eva',6500,'Ac','president');
insert into empy values(203,'Oliva',7500,'HR','manager');
insert into empy values(204,'Giri',7500,'Ac','snr grade');
insert into empy values(205,'Rena',7500,'HR','president');

```
DECLARE
    total_rows number(2);
```

```
   emp1 EMPY%rowtype;
BEGIN
 UPDATE empy SET salary = salary + salary * 50/100 where dsgt = 'president';
 IF sql%notfound THEN
     dbms_output.put_line('no employee salary updated');
 ELSIF sql%found THEN
    total_rows := sql%rowcount;
    dbms_output.put_line( total_rows || ' employee salary details  updated');
 end if;
end;
```

output:

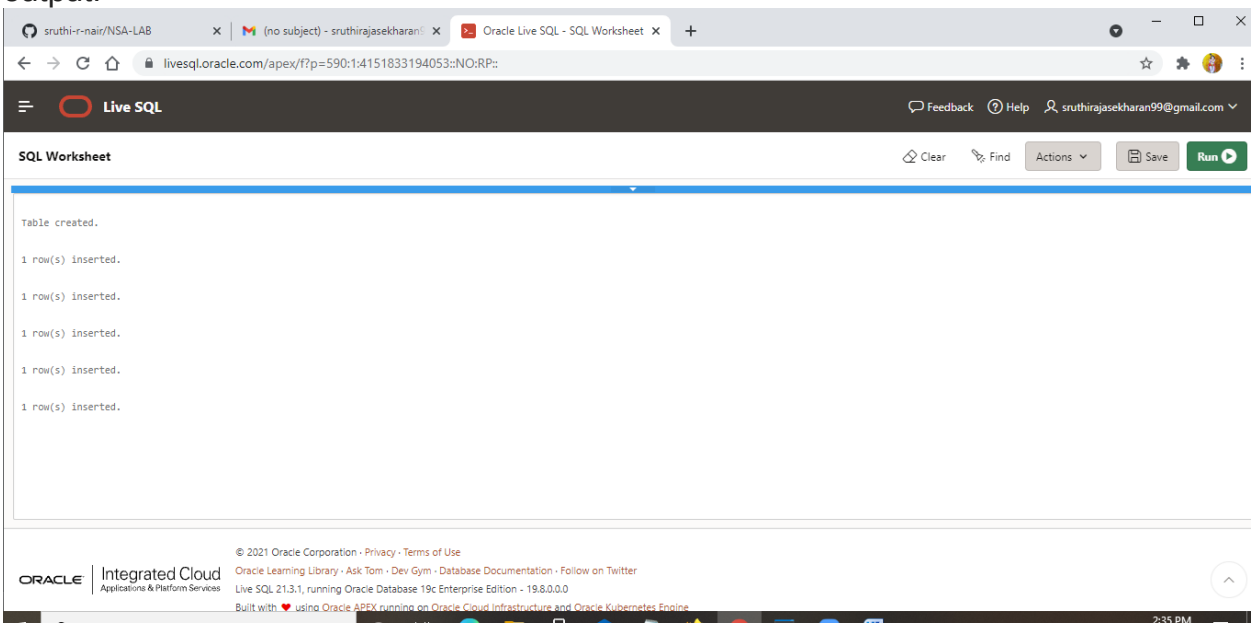


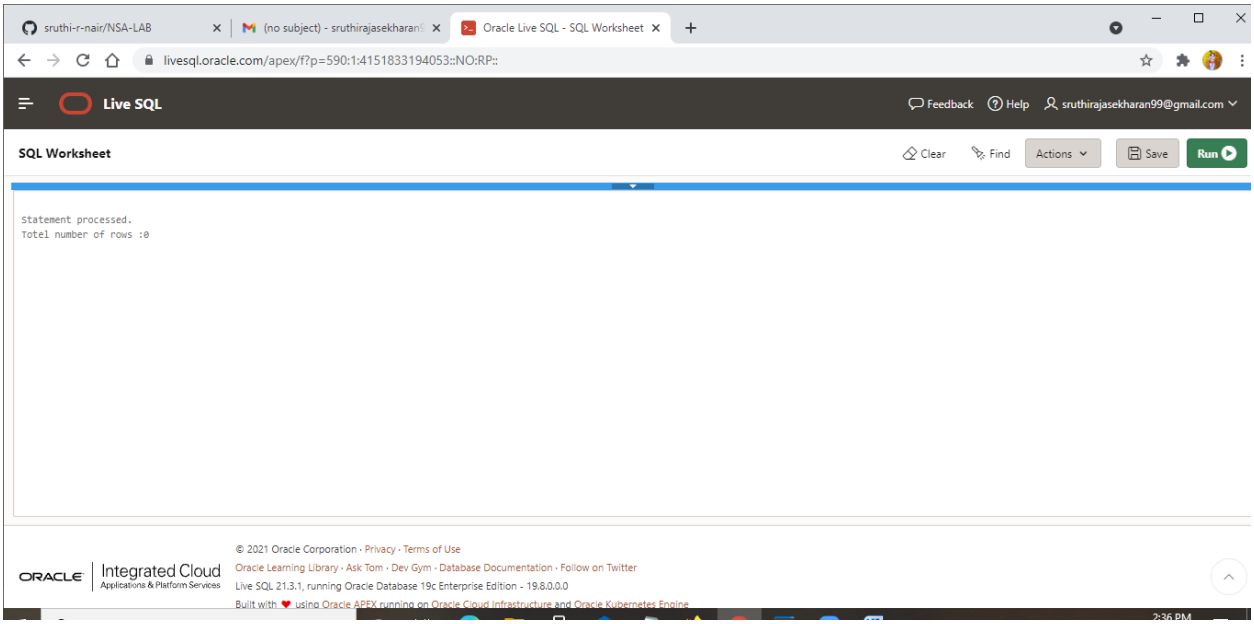


8)create table emplo(emp_no varchar(20),emp_name varchar(20),salary int,emp_dpt varchar(20),gender varchar(10));
insert into emplo values('301','Sree',50000,'sales','male');
insert into emplo values('302','Achu',6500,'Ac','female');
insert into emplo values('303','Tiffi',7500,'HR','female');
insert into emplo values('304','Seena',7500,'Ac','male');
insert into emplo values('305','Aliya',7500,'HR','female');

```
DECLARE
 CURSOR emp1 is SELECT * FROM emplo WHERE emp_name like ('s%');
 emp2 emp1%rowtype;
BEGIN
 open emp1;
 loop
  fetch emp1 into emp2;
  exit when emp1%notfound;
  dbms_output.put_line('employee information: '||' '||emp2.emp_no || ' ' ||
emp2.emp_name || ' ' || emp2.salary|| ' '||emp2.emp_dpt||' '||emp2.gender);
 end loop;
 dbms_output.put_line('Totel number of rows :'||emp1%rowcount);
close emp1;
end;
```

output:





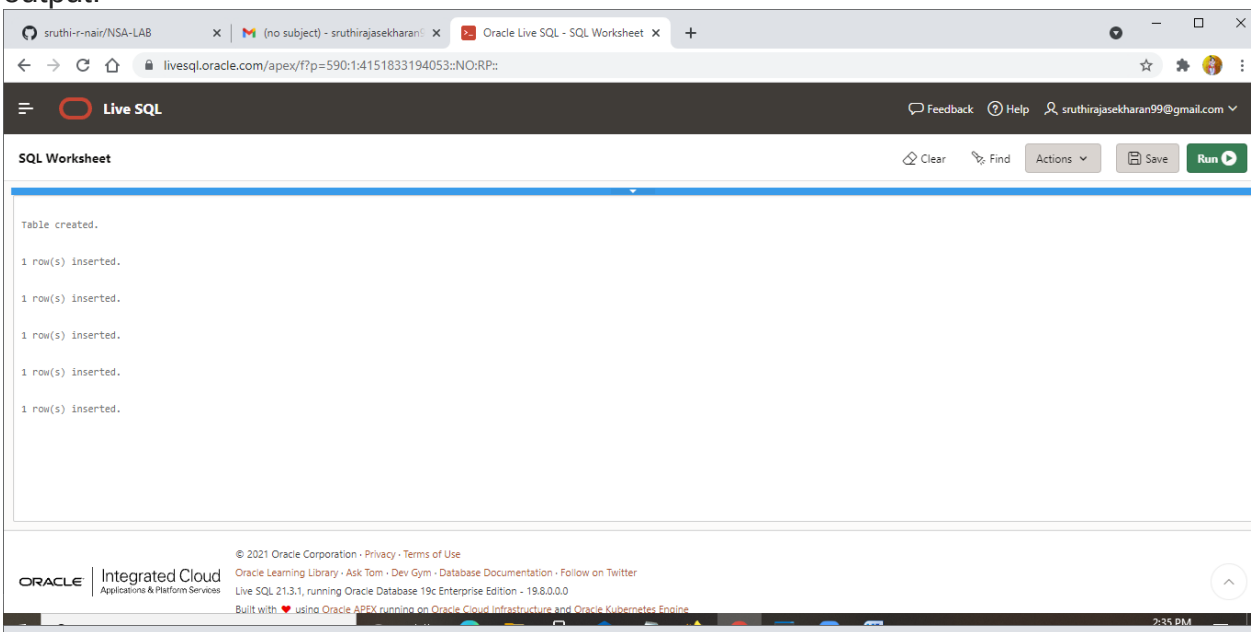
9) create table book(accession_no int , title varchar(20), publisher varchar(20), publishedDate date, author varchar(20), status varchar(30));

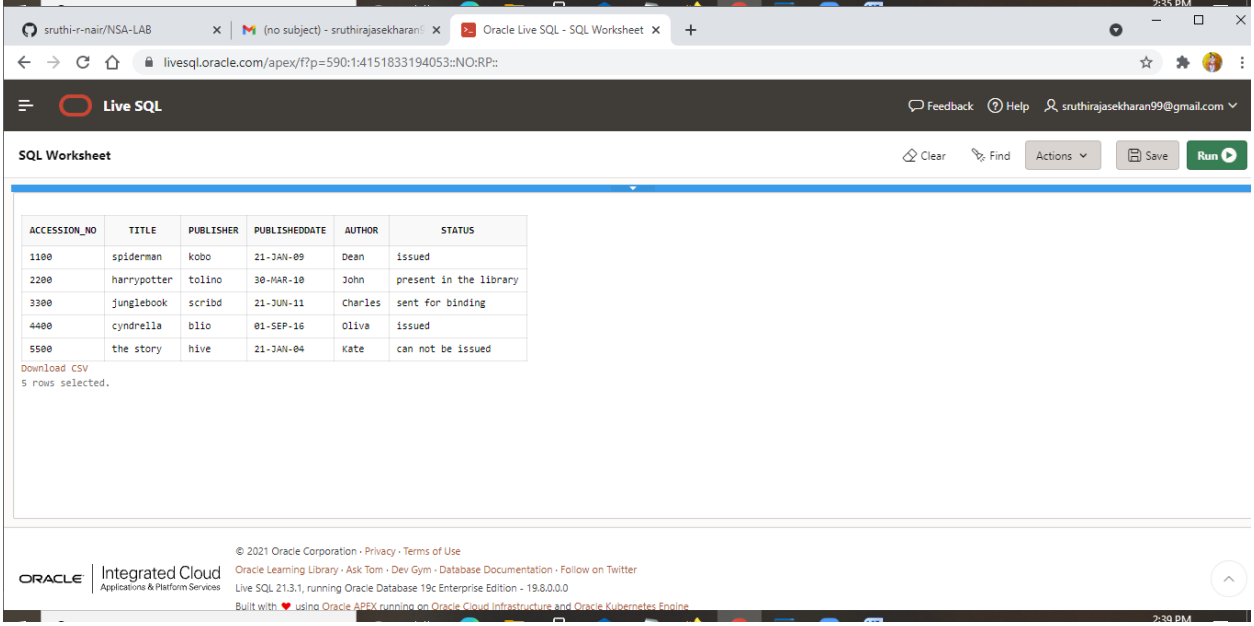
insert into book values(1100,'spiderman','kobo','21-jan-2009','Dean','issued');
insert into book values(2200,'harrypotter','tolino','30-mar-2010','John','present in the library');
insert into book values(3300,'junglebook','scribd','21-june-2011','Charles','sent for binding');
insert into book values(4400,'cyndrella','blio','01-sep-2016','Oliva','issued');
insert into book values(5500,'the story','hive','21-jan-2004','Kate','can not be issued');
insert into book values(6600,magicone','fnac','21-jan-2006','Harry',' issued');

```
CREATE OR REPLACE TRIGGER search1
 before insert  ON book
 FOR EACH ROW
 declare
 temp date;
BEGIN
 select sysdate into temp from dual;
 if inserting  then
  if :new.publishedDate < add_months(temp, -180) then
       :new.status:='cannot be issued' ;
  end if;
 end if;
end;
```

select * from book;

output:



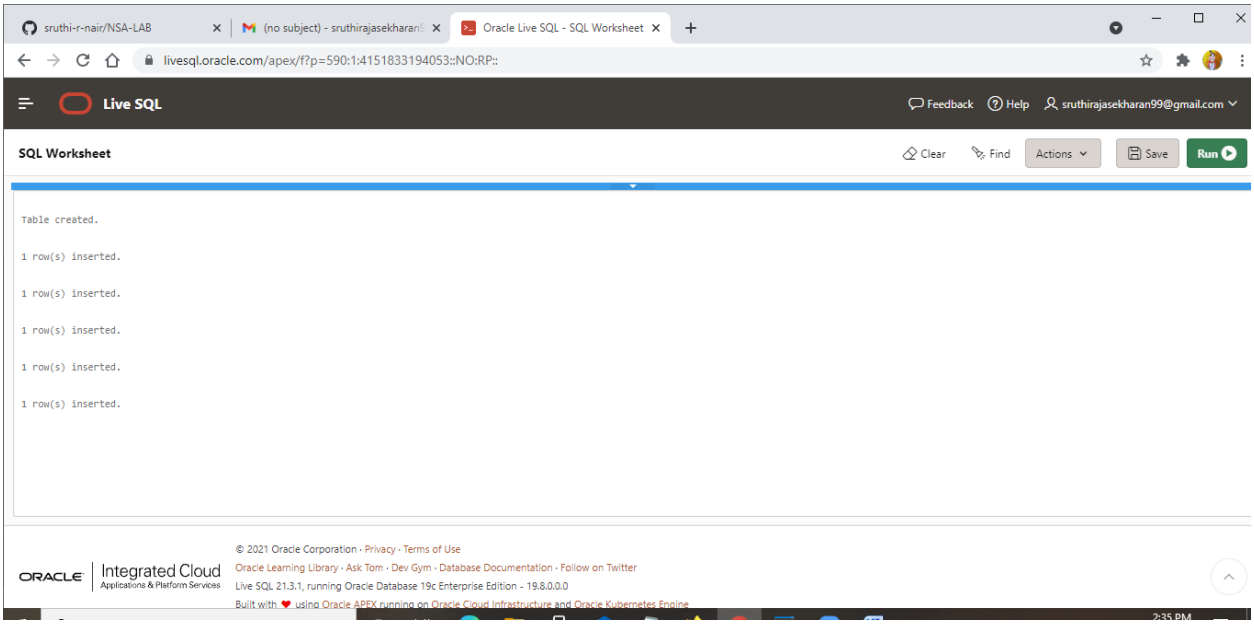


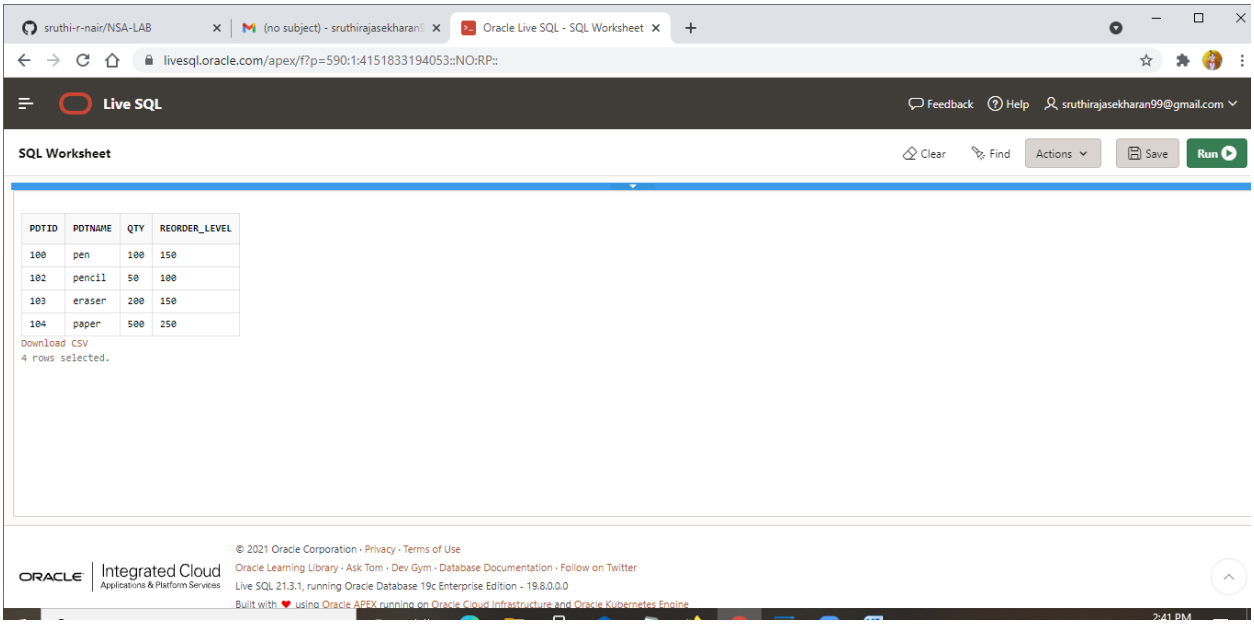
10)create table inventory(pdtid number primary key, pdtname varchar(10), qty int,reorder_level number);

```
insert into inventory values(100,'pen',100,150);
insert into inventory values(102,'pencil',50,100);
insert into inventory values(103,'eraser',200,150);
insert into inventory values(104,'paper',500,250);

CREATE OR REPLACE TRIGGER checking
 before insert  ON inventory
 FOR EACH ROW
declare
BEGIN
```

```
 if inserting  then
  if :new.qty > :new.reorder_level then
       :new.reorder_level:=0;
  end if;
 end if;
end;

select * from inventory;
```

output:




sruthi-r-nair/NSA-LAB
(no subject) - sruthirajasekharan
Oracle Live SQL - SQL Worksheet
livesql.oracle.com/apex/f?p=590:1:4151833194053::NO:RP::
Live SQL
Feedback
Help
sruthirajasekharan99@gmail.com
SQL Worksheet
Clear
Find
Actions
Save
Run
PDTID PDTNAME QTY REORDER_LEVEL
100 pen 100 150
102 pencil 50 100
103 eraser 200 150
104 paper 500 250
Download CSV
4 rows selected.
ORACLE
Integrated Cloud
Applications & Platform Services
© 2021 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter
Live SQL 21.3.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine
2:41 PM