**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Assignment** | **Page No.** |
| 1. | INTRODUCTION | 3 |
| 2. | ER DIAGRAM DESCRIPTION | 4 |
| 3 | ER DIAGRAM | 5 |
| 4. | ER TO TABLE | 6 |
| 5. | NORMALIZATION | 7 |
| 6. | CREATE TABLES | 7 |
| 7. | INSERTION COMMANDS | 10 |
| 8. | SQL QUERIES | 12 |
| 9. | PL/SQL QUERIES | 27 |

**INTRODUCTION**

In this modern era of online shopping, no seller wants to be left behind, moreover due to its simplicity the shift from an offline selling model to an online selling model is witnessing rampant growth.

Therefore, as an engineer, our job is to ease the path of this transition for the seller. Amongst many things that an online site requires the most important is a database system. Hence in this project, we are planning to design a database where small clothing sellers can sell their products online.

The basic function of our database is to digitally store information that can be captured, retrieved, and distributed easily at a later time.

This also gives businesses the ability to analyze and track information about the products, sales, and customers that have been input into the database.

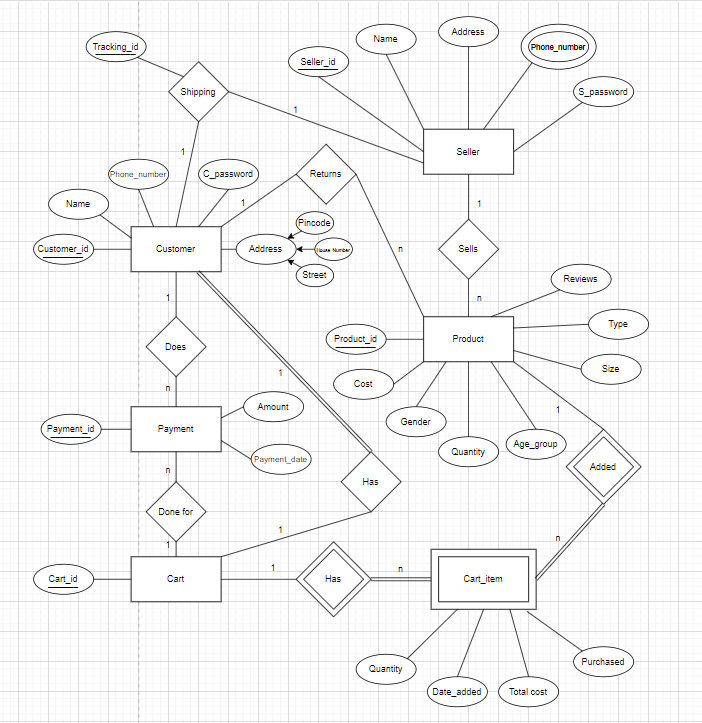
One of the biggest benefits of using a database for e-commerce is the addition of structure to vast amounts of shop data. No matter how big or small your online store is, an infrastructure is needed for all the gathered information to make sense and provide useful insights. When the data is structured, it can be accessed more efficiently by the e-commerce application.

Myntra offers a well-curated comprehensive selection of clothes from different brands. It is a one-stop shop for all the fashion and lifestyle needs. Myntra aims at providing a hassle-free and enjoyable shopping experience to shoppers across the country with the widest range of brands and products on its portal. It serves consumers through its retail websites with a focus on selection, price, and convenience. In this project, we are making a conscious effort to bring the power of fashion to shoppers with an array of the latest and trendiest products available in the country.

**ER DIAGRAM DESCRIPTION**

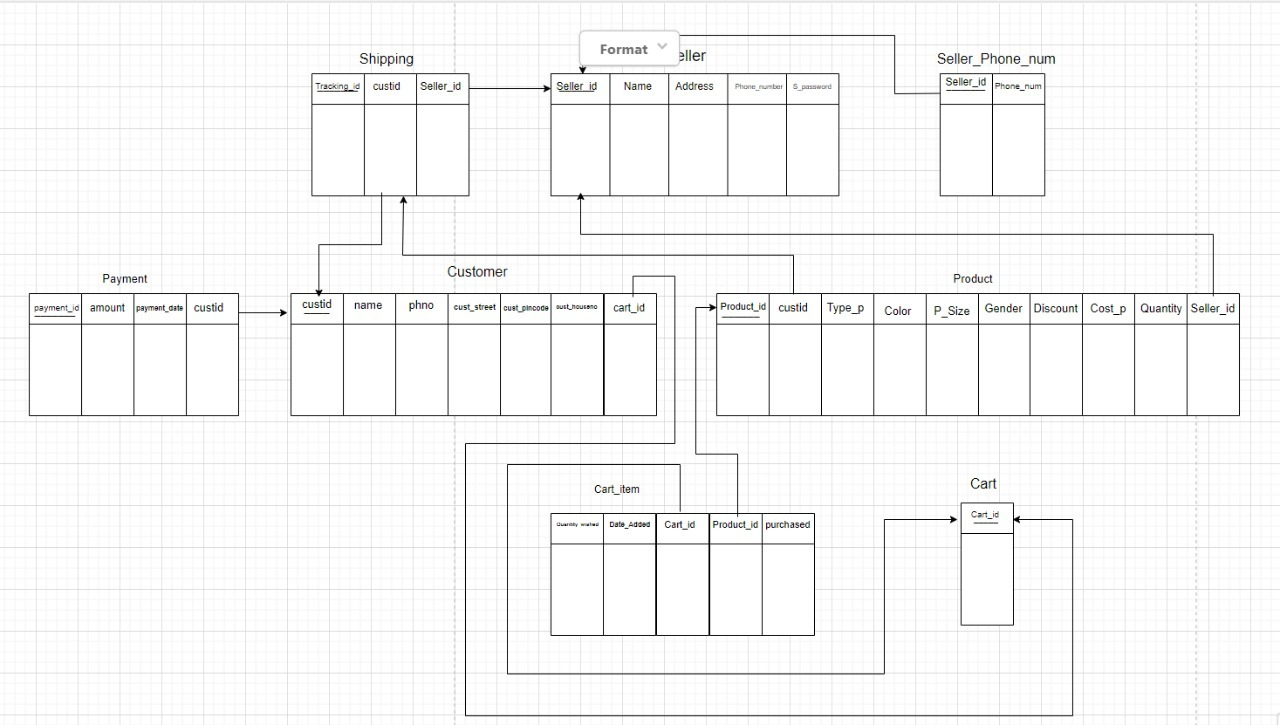
These shopping websites enable many sellers to sell their products to customers at various locations. Each seller has the following attribute: seller\_id,s\_pass,Name\_s and address. Each product has the following attributes: Product\_id, Type\_p, color,p\_size, gender, discount,cost\_p and quantity. The customer makes the payment for the cart items added to the cart. Each customer has a cart with its unique cart\_id. Cart items have the following attributes: Quantity\_wished, purchased,date\_added.

After the product is bought the sellers ship product to customers. The customer has the following attributes: name,phone\_no.,custid and address. If the customer doesn’t like the product he may request a return of the product.



**ER diagram**

**ER to Table**

****

**Normalization**

**1NF:** Since the tables do not contain any composite attribute or multi-valued attributes or their combinations, so they are normalized according to 1NF.

**2NF:** Since there is no non-primary key attribute in the tables that is partially dependent on the primary key, they are normalized according to 2NF.

**3NF:** Since no non-key attribute is transitively dependent on the primary key in the tables, they are normalized according to 3NF.

**BCNF:** Since every determinant is the primary key in the tables, they are normalized according to BCNF.

**4NF:** Since the tables have no multivalued dependencies, so they are normalized according to 4NF

**5NF:** Since the tables don’t have a lossless decomposition into any number of smaller tables (relations), so they are normalized according to 5NF.

**CREATE TABLE COMMANDS:**

CREATE TABLE Seller

(

Seller\_id VARCHAR(6) ,

s\_pass VARCHAR(10) ,

Name\_s VARCHAR(20) ,

Shop\_number NUMBER,

Street VARCHAR(30) ,

Pincode NUMBER,

PRIMARY KEY (Seller\_id)

);

CREATE TABLE Seller\_Phone\_num

(

Phone\_num NUMBER(10),

Seller\_id VARCHAR(6)REFERENCES Seller(Seller\_id) ON DELETE CASCADE,

PRIMARY KEY (Phone\_num, Seller\_id)

);

CREATE TABLE Cart

(

Cart\_id VARCHAR(7) NOT NULL,

CONSTRAINT pk\_cart PRIMARY KEY(Cart\_id)

);

CREATE TABLE customer(  
name VARCHAR(20),

phno NUMBER(10),

custid VARCHAR(12) PRIMARY KEY,

cust\_houseno NUMBER(10),

cust\_street VARCHAR(40),

cust\_pincode NUMBER(10),

cart\_id VARCHAR(7) REFERENCES cart(cart\_id));

CREATE TABLE SHIPPING

(

Tracking\_id VARCHAR (13) PRIMARY KEY,

custid VARCHAR(12) REFERENCES Customer(custid),

Seller\_id VARCHAR(6)REFERENCES Seller(Seller\_id) ON DELETE SET NULL

);

CREATE TABLE payment(  
payment\_id VARCHAR(20) PRIMARY KEY,

amount NUMBER(10),

payment\_date DATE,

custid VARCHAR(12) REFERENCES customer(custid));

CREATE TABLE Product

(

Product\_id VARCHAR(7) NOT NULL,

custid VARCHAR(12),

Type\_p VARCHAR(7) NOT NULL,

Color VARCHAR(15) NOT NULL,

P\_Size VARCHAR(2) NOT NULL,

Gender CHAR(1) NOT NULL,

Discount NUMBER(2) NOT NULL,

Cost\_p NUMBER(5) NOT NULL,

Quantity NUMBER(2) NOT NULL,

Seller\_id VARCHAR(6),

PRIMARY KEY (Product\_id),

FOREIGN KEY (Seller\_id) REFERENCES Seller(Seller\_id) ON DELETE SET NULL,

FOREIGN KEY (custid) REFERENCES customer(custid)

ON DELETE SET NULL

);

CREATE TABLE Cart\_item

(

Quantity\_wished NUMBER(1) NOT NULL,

Date\_Added DATE NOT NULL,

Cart\_id VARCHAR(7) NOT NULL,

Product\_id VARCHAR(7) NOT NULL,

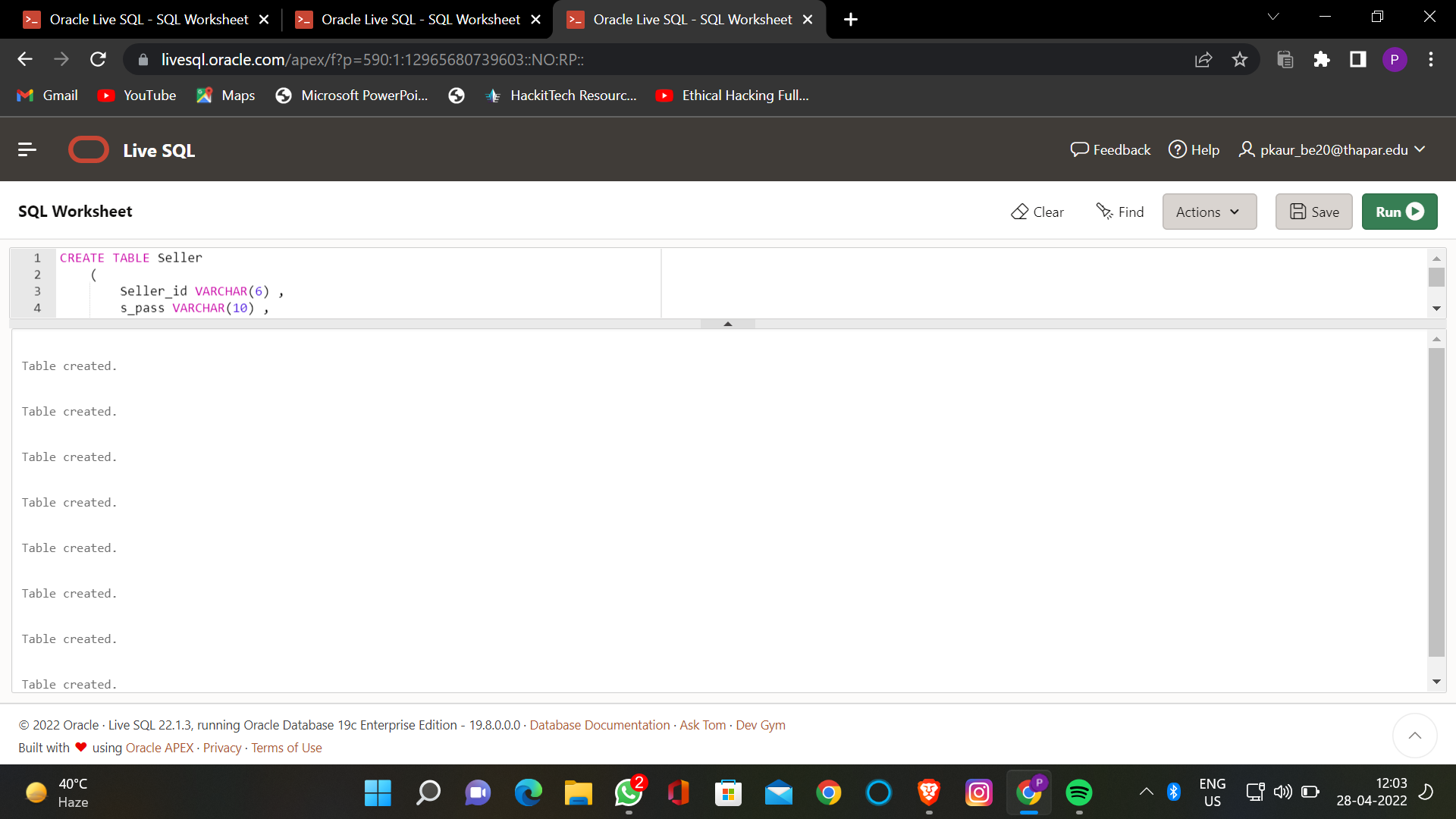
purchased VARCHAR(5),

CONSTRAINT fk\_cart FOREIGN KEY (Cart\_id) REFERENCES Cart(Cart\_id),

CONSTRAINT fk\_product FOREIGN KEY (Product\_id) REFERENCES Product(Product\_id),

CONSTRAINT pk\_cart\_item Primary key(Cart\_id,Product\_id)

);



**INSERTION COMMANDS:**

insert into Seller values('1ABQ36','#123','Ajitpal',1470,'Rajamata Street',110003);

insert into Seller values('1CFQ36','#456','Raman',2134,'Ram Laal Chowk',110005);

insert into Seller values('26KL74','#ABCD','Amrit',73,'Munshi Chowk',400004);

insert into Seller values('17KLP7','#R$23','Gopal',677,'Sahibzada Nagar',140307);

insert into Seller values('23EFD5','a345d','Sumita',523,'Nawa kot Nagar',132103);

insert into Seller\_Phone\_num values(9873654321,'26KL74');

insert into Seller\_Phone\_num values(9813732561,'23EFD5');

insert into Seller\_Phone\_num values(9765310987,'26KL74');

insert into Seller\_Phone\_num values(9349845100,'1ABQ36');

insert into Seller\_Phone\_num values(8453877652,'26KL74');

insert into Seller\_Phone\_num values(8873654321,'1CFQ36');

insert into Seller\_Phone\_num values(7765654321,'17KLP7');

insert into Cart values('1ADF34');

insert into Cart values('2FDF54');

insert into Cart values('1QRT89');

insert into Cart values('5MPL24');

insert into Cart values('1HDF98');

insert into customer values('Mansi',8862473245,'765FTYGUHUHB',1432,'Gandhi Nagar Street',147001,'2FDF54');

insert into customer values('Ajay',9834567754,'765DBHBABDBB',234,'Sabzi Mandi Street',110864,'1ADF34');

insert into customer values('Jitender',9865467356,'565FTYHYHUHB',75,'Ranjit Nagar',110086,'1QRT89');

insert into customer values('Dhruv',7865473240,'765FTYGUHUUI',101,'Virat Nagar',147345,'5MPL24');

insert into customer values('Arsh',9855327255,'865FTYGRHIHB',3465,'Model Town',132103,'1HDF98');

insert into SHIPPING values('7YIUGF','765FTYGUHUHB','26KL74');

insert into SHIPPING values('87HHKL','765DBHBABDBB','1ABQ36');

insert into SHIPPING values('34VHJF','565FTYHYHUHB','23EFD5');

insert into SHIPPING values('45GVCW','765FTYGUHUUI','1CFQ36');

insert into SHIPPING values('87JFKJ','865FTYGRHIHB','17KLP7');

insert into payment values('TEJBKD524VVR',5000,'23-DEC-2022','765FTYGUHUHB');

insert into payment values('TYRFWD524VVR',500,'25-DEC-2021','765DBHBABDBB');

insert into payment values('YSJBKD524VVR',2730,'12-SEP-2020','565FTYHYHUHB');

insert into payment values('RFGBKD524VVR',420,'02-AUG-2018','765FTYGUHUUI');

insert into payment values('TEYPOC524VVR',1190,'06-JUN-2019','865FTYGRHIHB');

insert into payment values('TEJBKD545VVR',750,'25-OCT-2015','565FTYHYHUHB');

insert into Productvalues('1267543','765FTYGUHUHB','Sandal','Blue','6','F',20,420,1,'26KL74');

insert into Product values('2345676','765DBHBABDBB','Coat','Black','32','M',30,2730,1,'23EFD5');

insert into Product values('3647833','565FTYHYHUHB','Shoes','Green','9','M',25,5000,2,'1ABQ36');

insert into Product values('7384903','765FTYGUHUUI','Shirt','White','12','F',50,500,1,'1CFQ36');

insert into Product values('4567839','865FTYGRHIHB','Jeans','Blue','36','F',0,1190,1,'17KLP7');

insert into Cart\_item values(1,'03-JUN-2022','1ADF34','1267543','yes');

insert into Cart\_item values(1,'23-AUG-2021','2FDF54','2345676','no');

insert into Cart\_item values(1,'06-JAN-2020','1QRT89','3647833','yes');

insert into Cart\_item values(1,'27-MAY-2022','5MPL24','7384903','no');

insert into Cart\_item values(1,'15-DEC-2021','1HDF98','4567839','yes');

insert into Cart\_item values(1,'17-DEC-2021','1HDF98','2345676','no');

**QUERIES:**

1. **Alter name\_s as Firstname, length of firstname as 15 and add one more field of Last\_name**

Alter table Seller

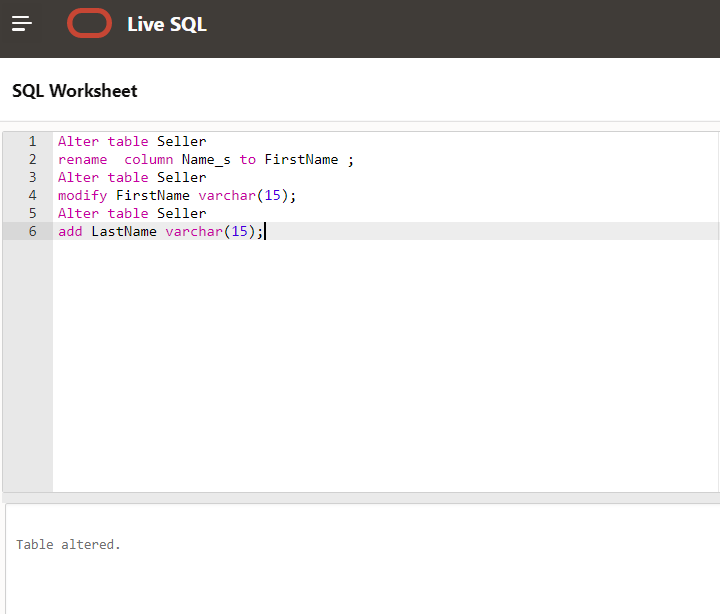
rename column Name\_s to FirstName ;

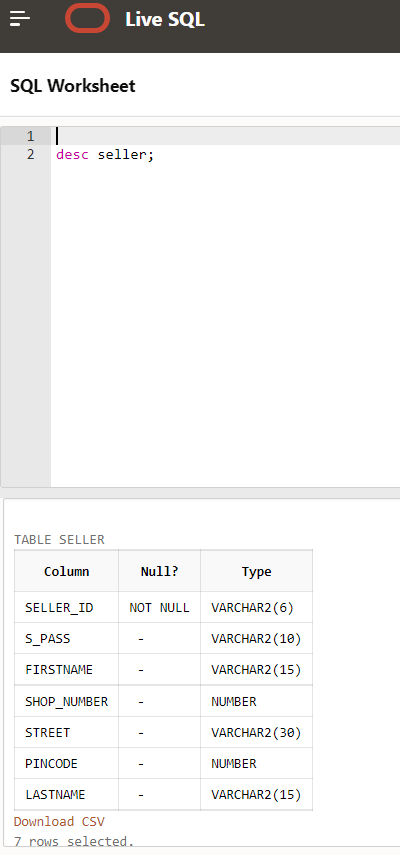
Alter table Seller

modify FirstName varchar(15);

Alter table Seller

add LastName varchar(15);

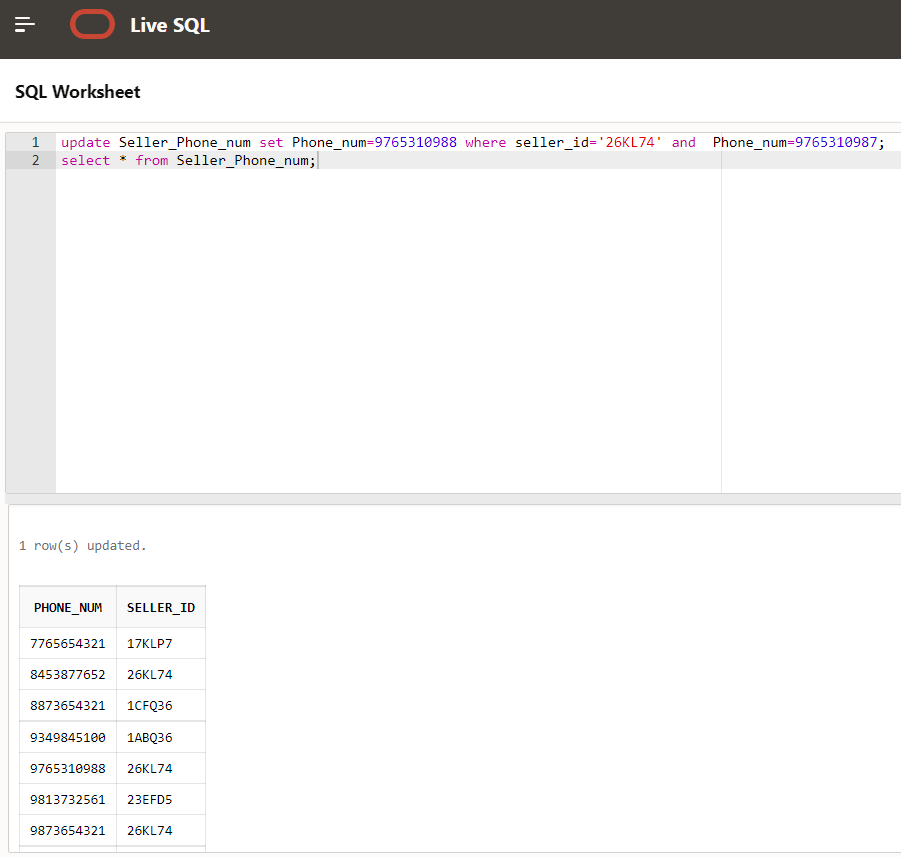


****

1. **If a seller wants update his phone number**

update Seller\_Phone\_num set Phone\_num=9765310988 where

seller\_id='26KL74' and Phone\_num=9765310987;



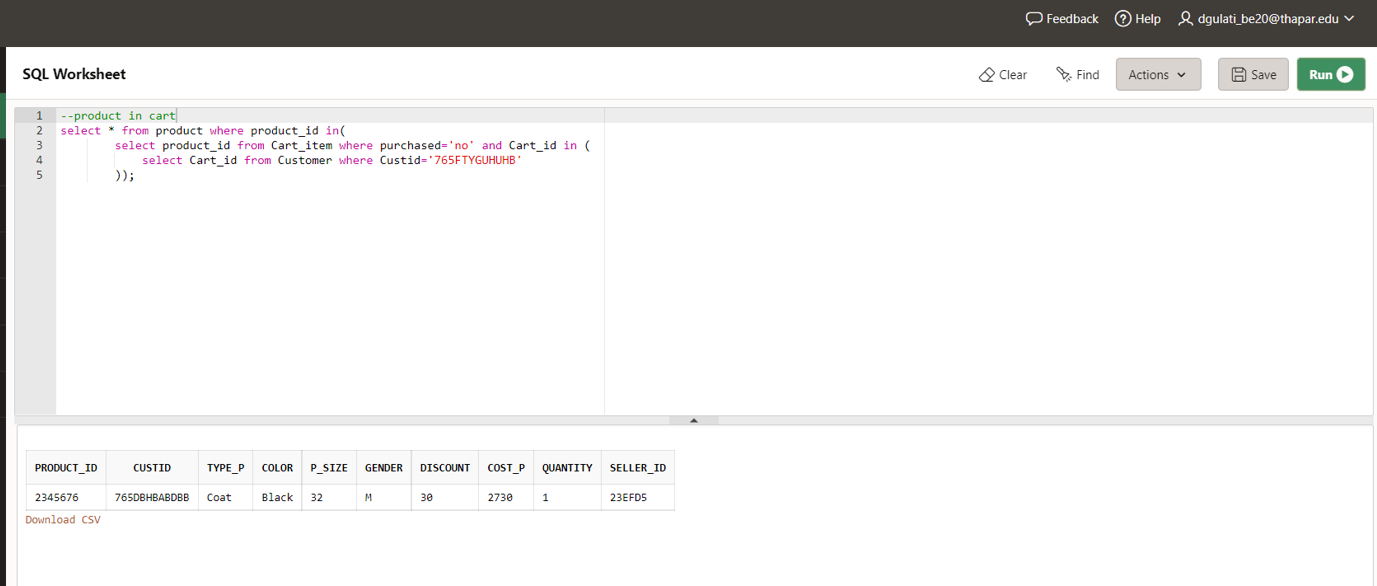
1. **If the customer wants to see details of the product present in the cart**

select \* from product where product\_id in(

select product\_id from Cart\_item where purchased=’no’ and Cart\_id in(

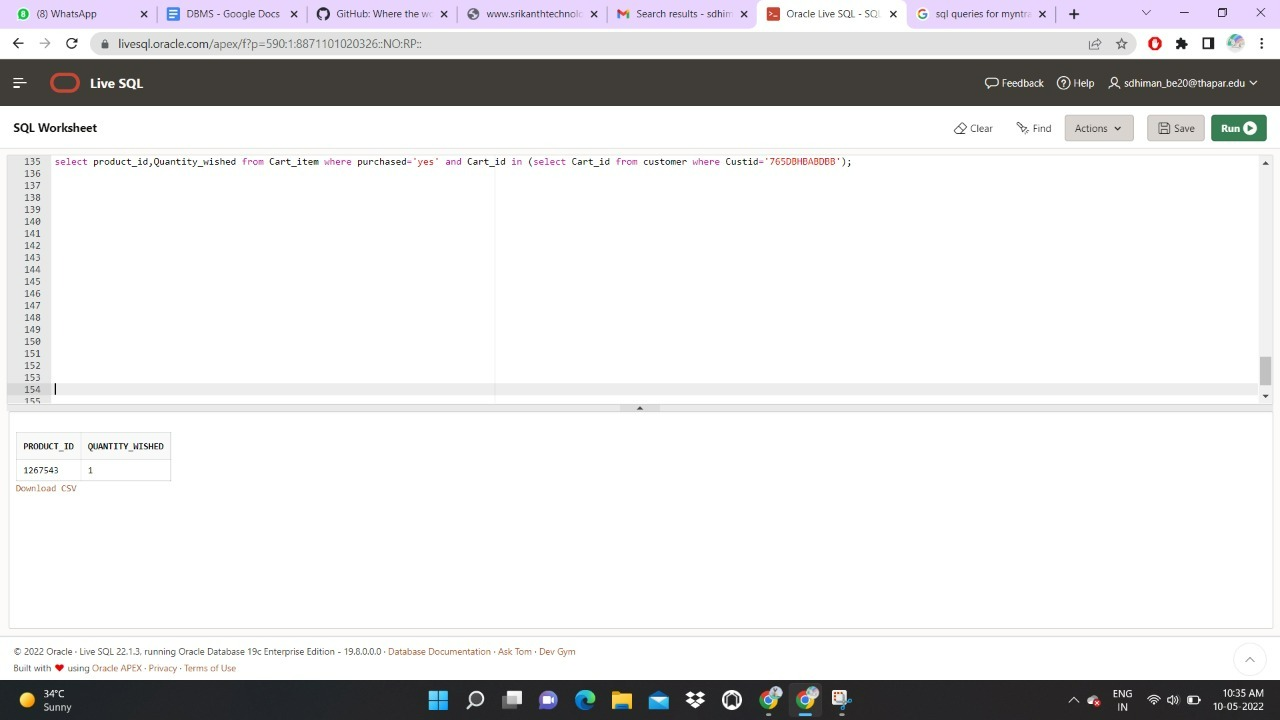
Select Cart\_id from Customer where custid=’765FTYGUHUHB’

));



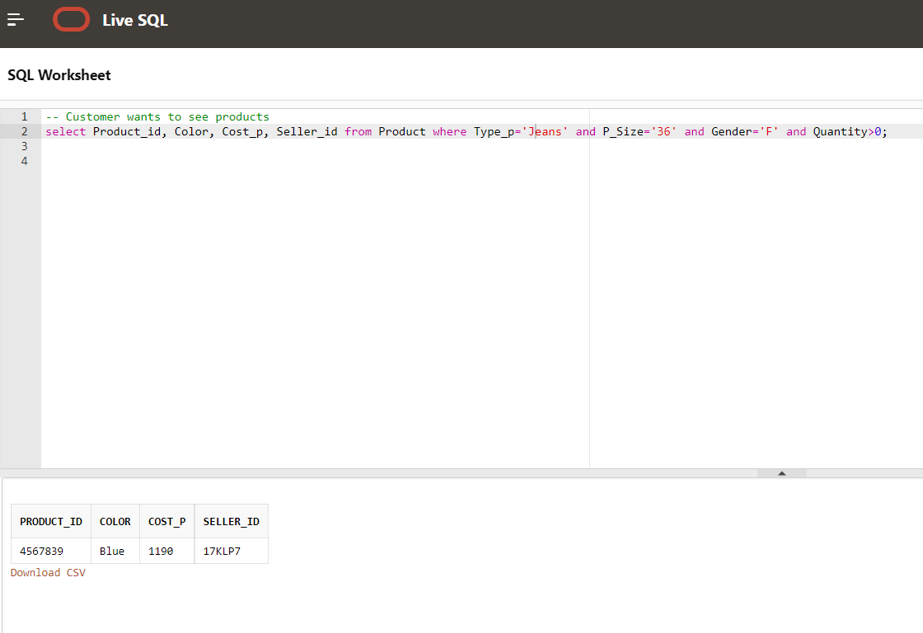
1. **If a customer wants to see order history**

select product\_id,Quantity\_wished from Cart\_item where purchased='yes' and Cart\_id in (select Cart\_id from customer where Custid='765DBHBABDBB');



1. **Customer wants to see filtered products on the basis of size,gender,type**

select product\_id, Color, Cost\_p, Seller\_id from product where Type\_p='jeans' and P\_size='36' and Gender='F' and Quantity>0);

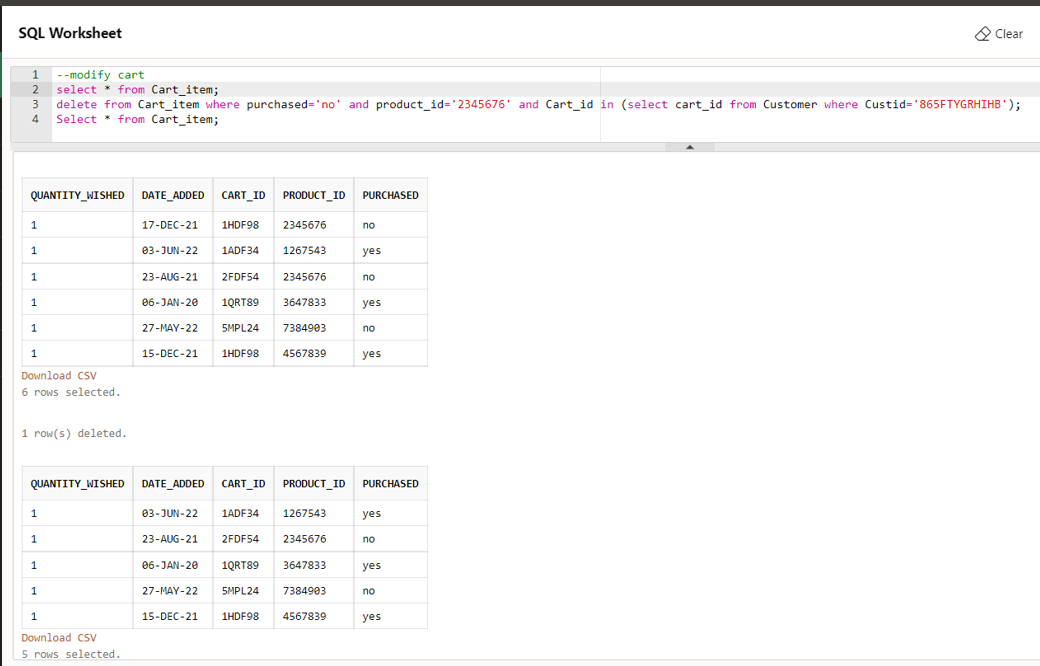


1. **If customer wants to modify the cart**

Select \* from Cart\_item;

delete from cart\_item where purchased=’no’and product\_id='2345676' and Cart\_id in (select cart\_id from Customer where Custid='865FTYGRHIHB'

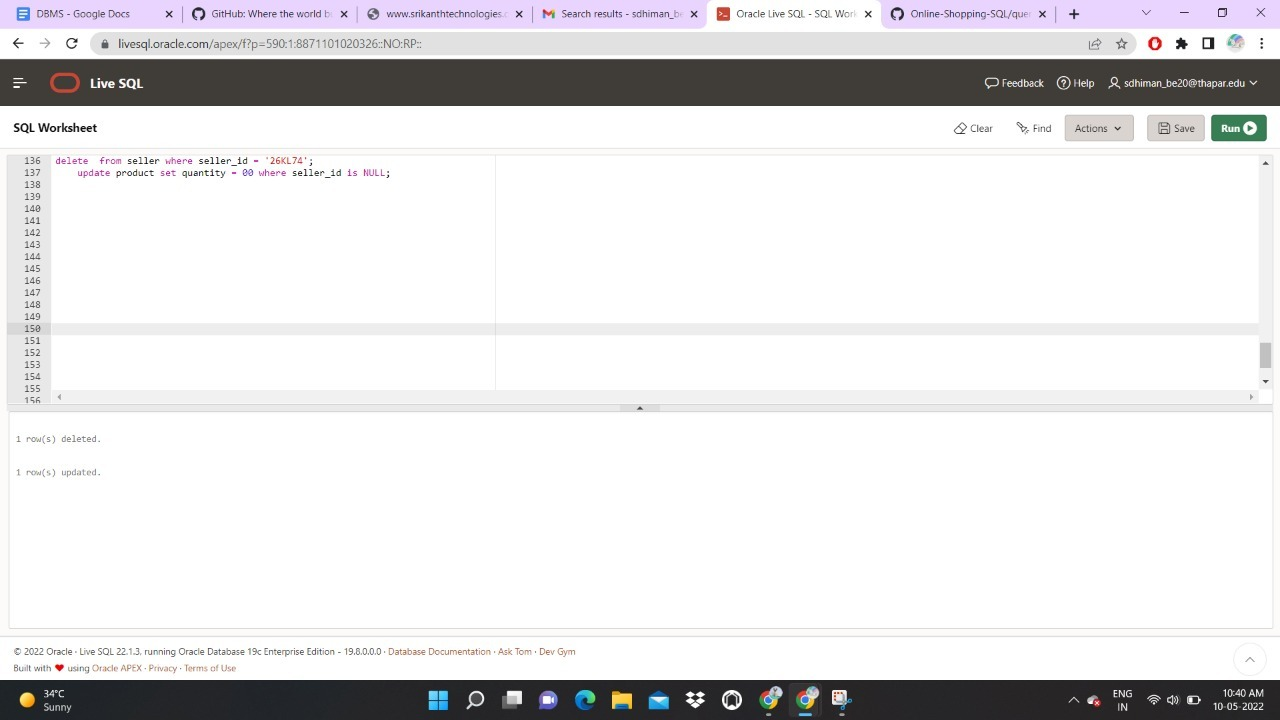
Select \* from Cart\_item

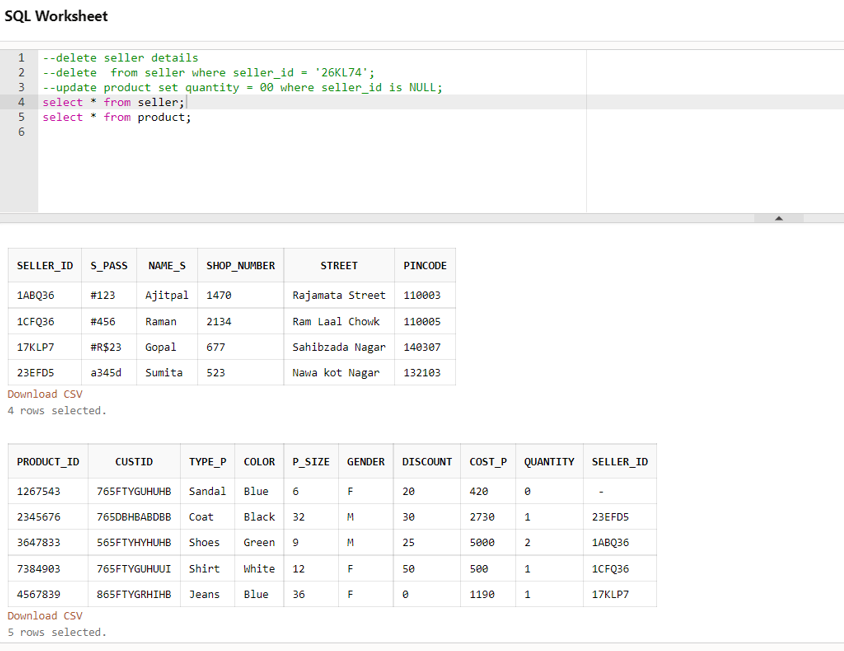
);

1. **If a seller stops selling his product**

delete from seller where seller\_id = '26KL74';

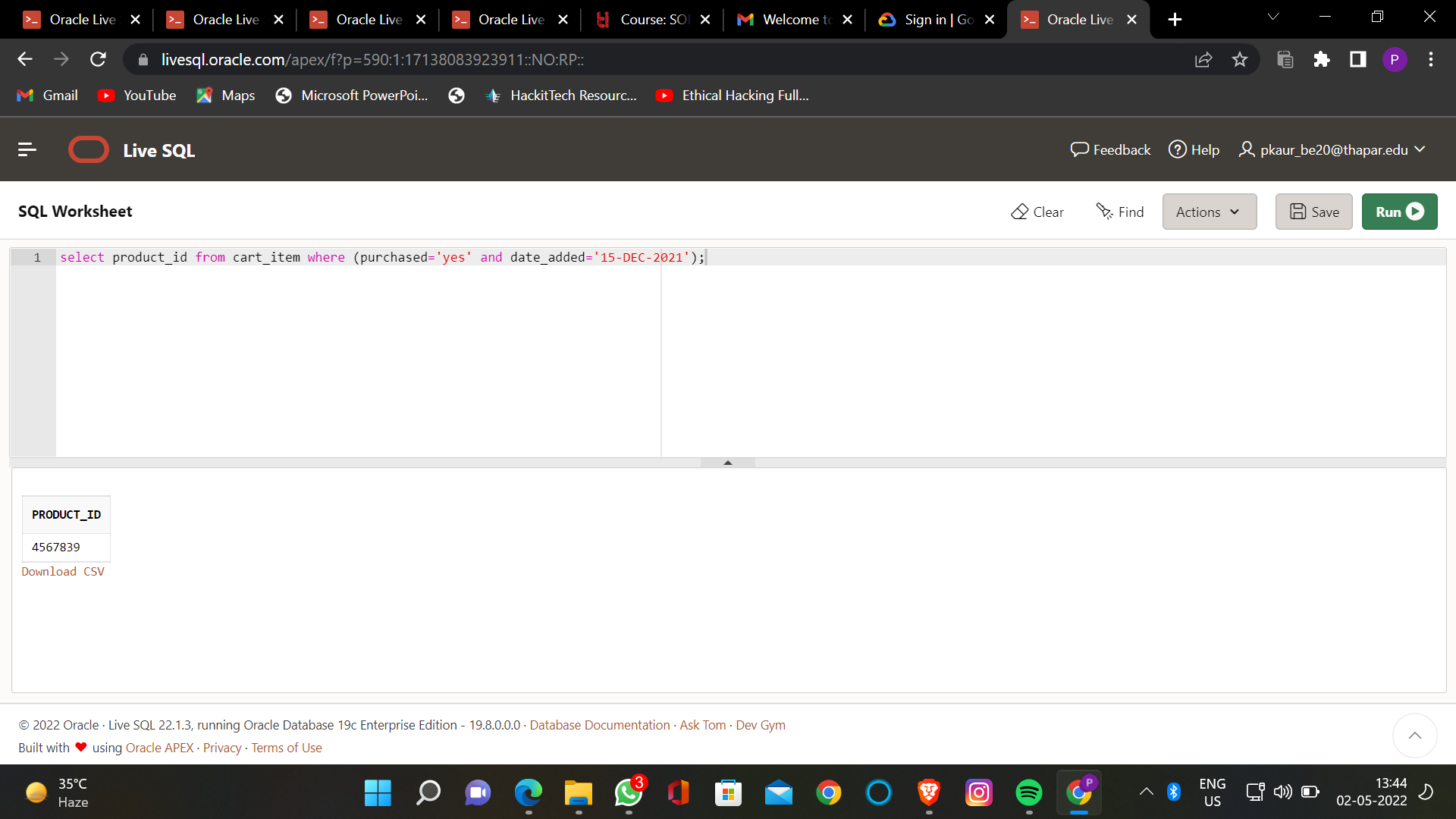
update product set quantity = 00 where seller\_id is NULL;





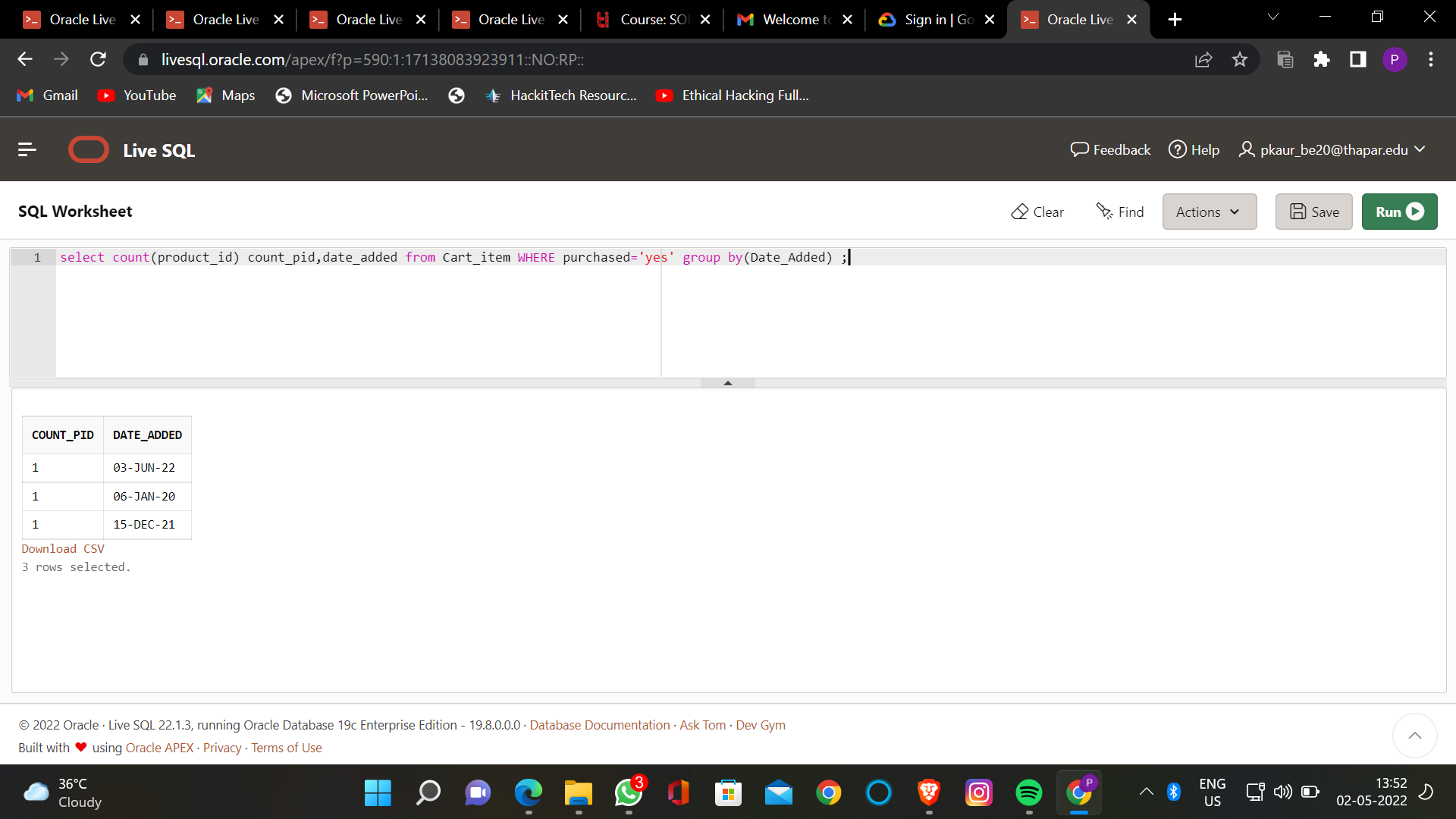
1. **If the admin wants to see what are the product purchased on the particular date**

select product\_id from cart\_item where (purchased='yes' and date\_added='15-DEC-2021');



1. **How much product sold on the particular date**

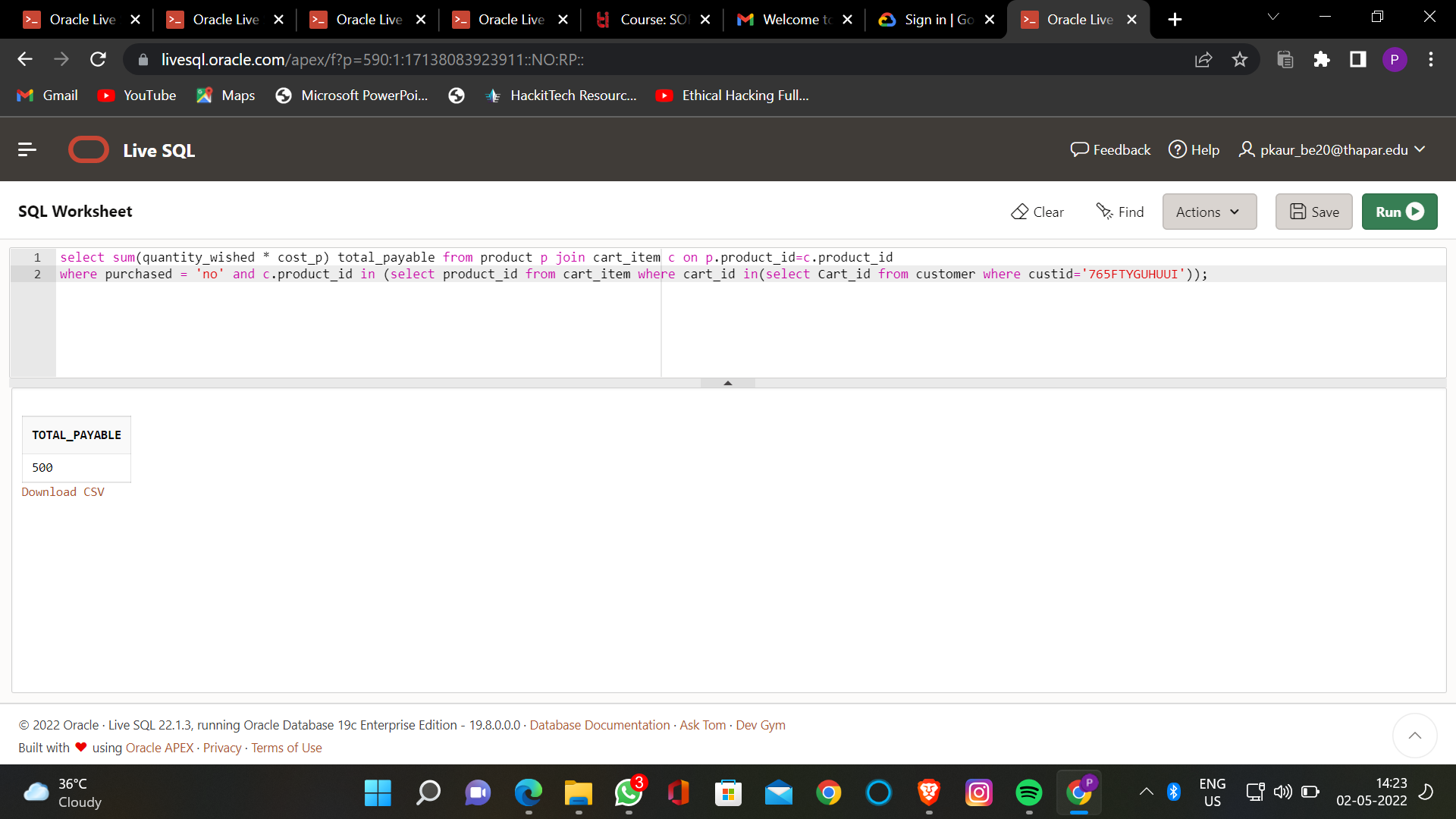
select count(product\_id) count\_pid,date\_added from Cart\_item WHERE purchased='yes' group by(Date\_Added) ;



1. **If a customer want to know the total price present in the cart**

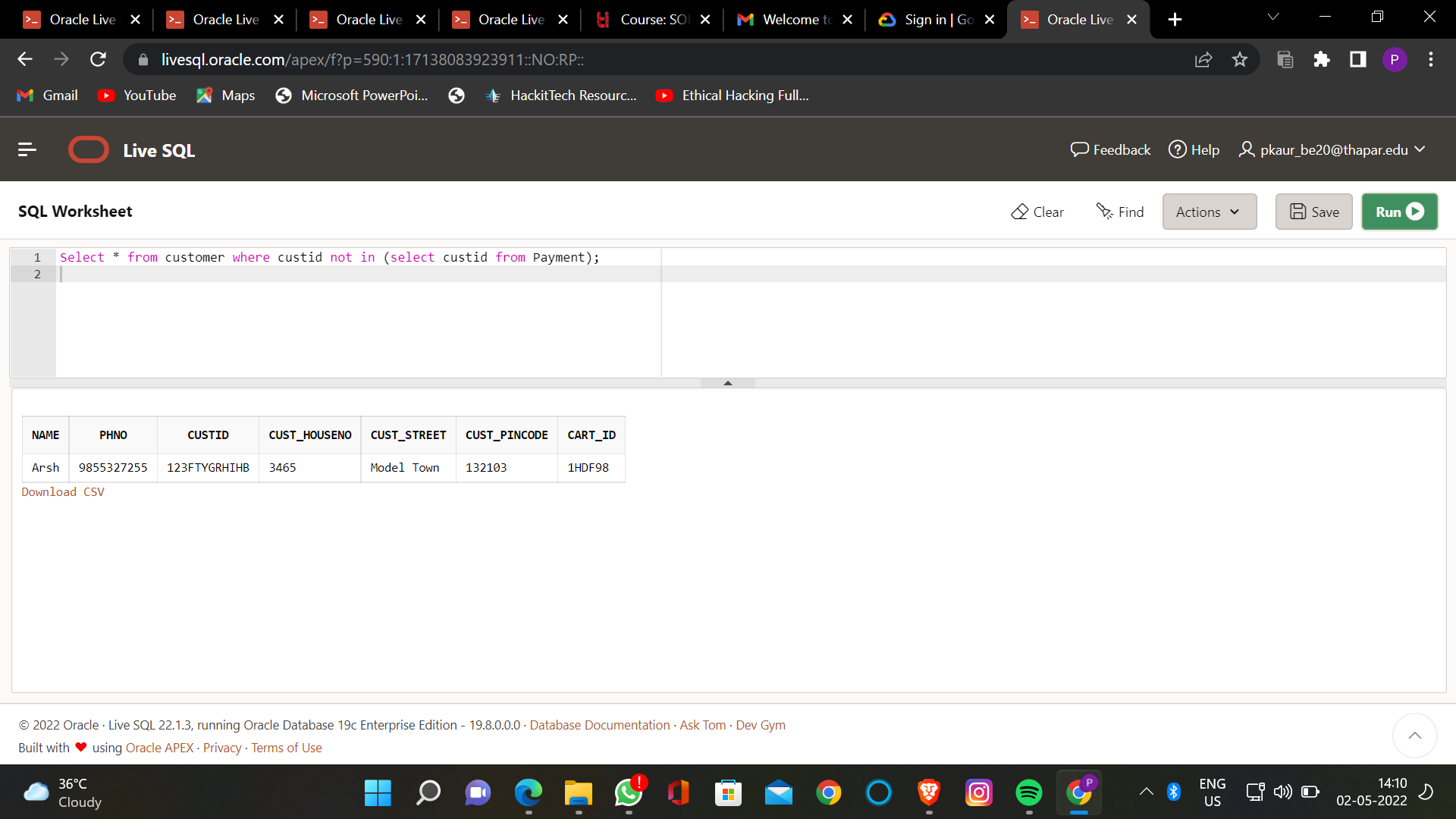
select sum(quantity\_wished \* cost\_p) total\_payable from product p join cart\_item c on p.product\_id=c.product\_id

where purchased = 'no' and c.product\_id in (select product\_id from cart\_item where cart\_id in(select Cart\_id from customer where custid='765FTYGUHUUI'));



1. **Show the details of the customer who has not purchased any thing**

Select \* from customer where custid not in (select custid from Payment);



1. **Find total sales of a seller..**

select sum(quantity\_wished \* cost\_p ) total\_profit,seller\_id from product p join cart\_item c on p.product\_id=c.product\_id WHERE purchased='yes' group by seller\_id;

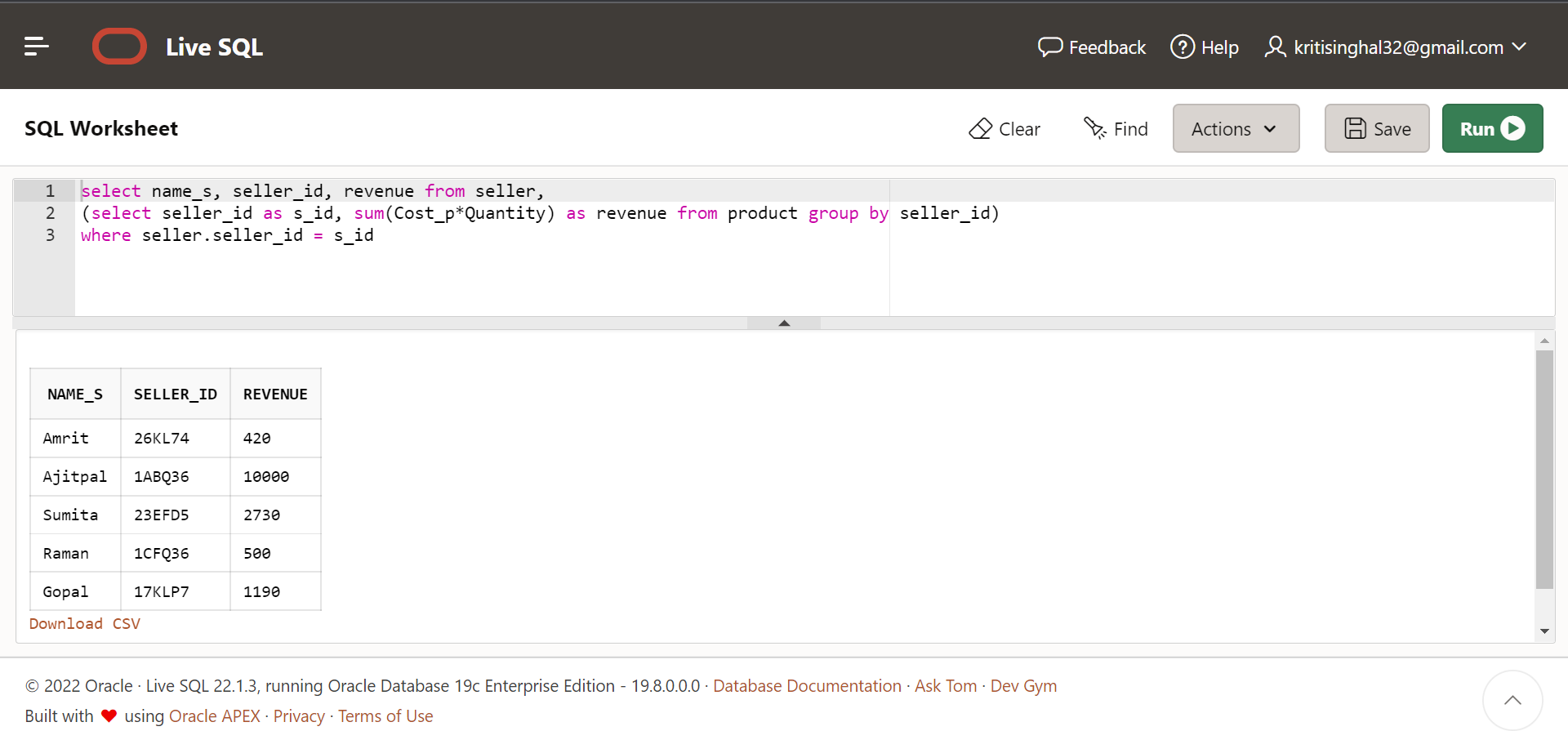


1. **Revenue generated by each seller on the particular date**

select name\_s, seller\_id, revenue from seller,

(select seller\_id as s\_id, sum(Cost\_p\*Quantity) as revenue from product group by seller\_id)

where seller.seller\_id = s\_id



1. **Create a view to display the payment details for a particular date**

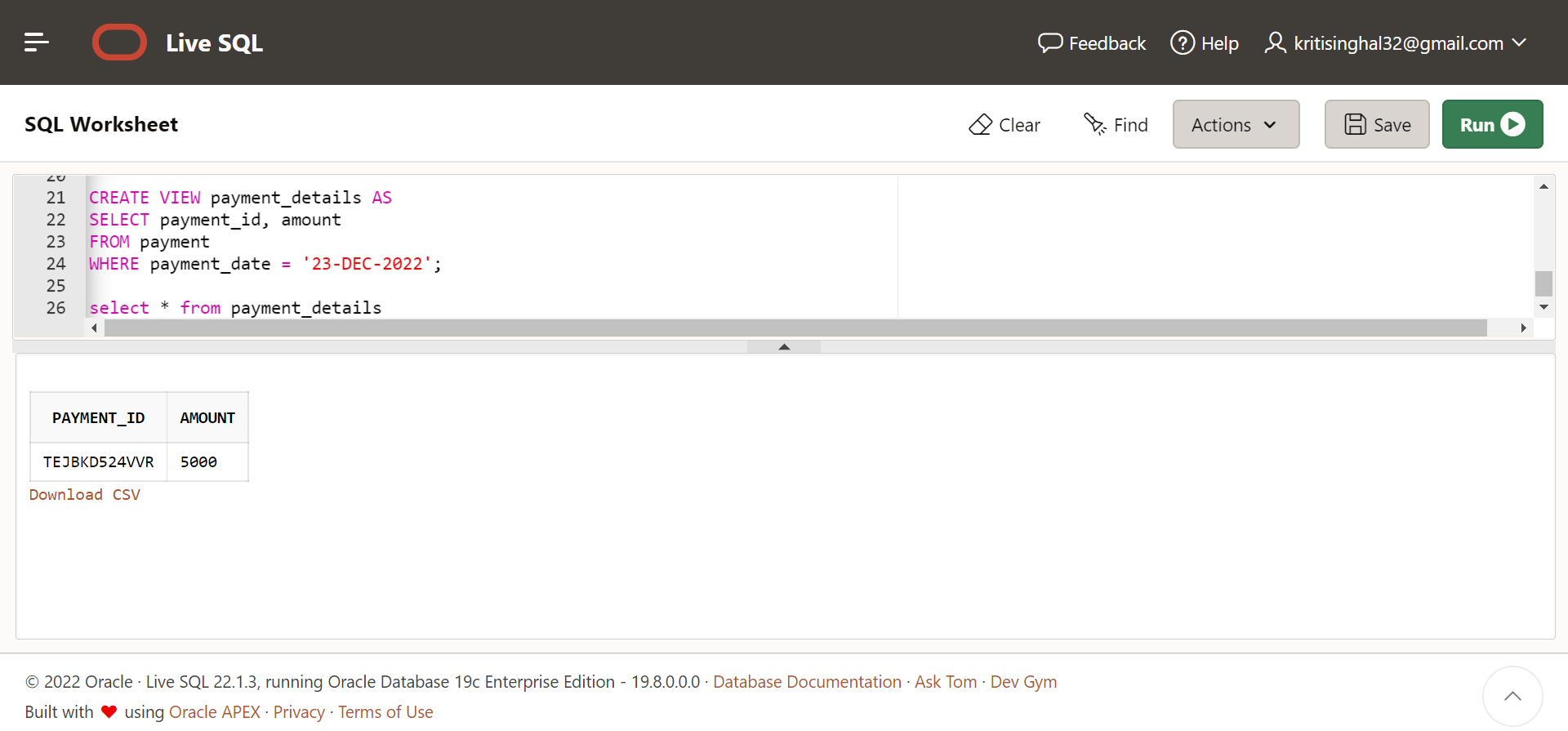
CREATE VIEW payment\_details AS

SELECT payment\_id, amount

FROM payment

WHERE payment\_date = '23-DEC-2022';

select \* from payment\_details

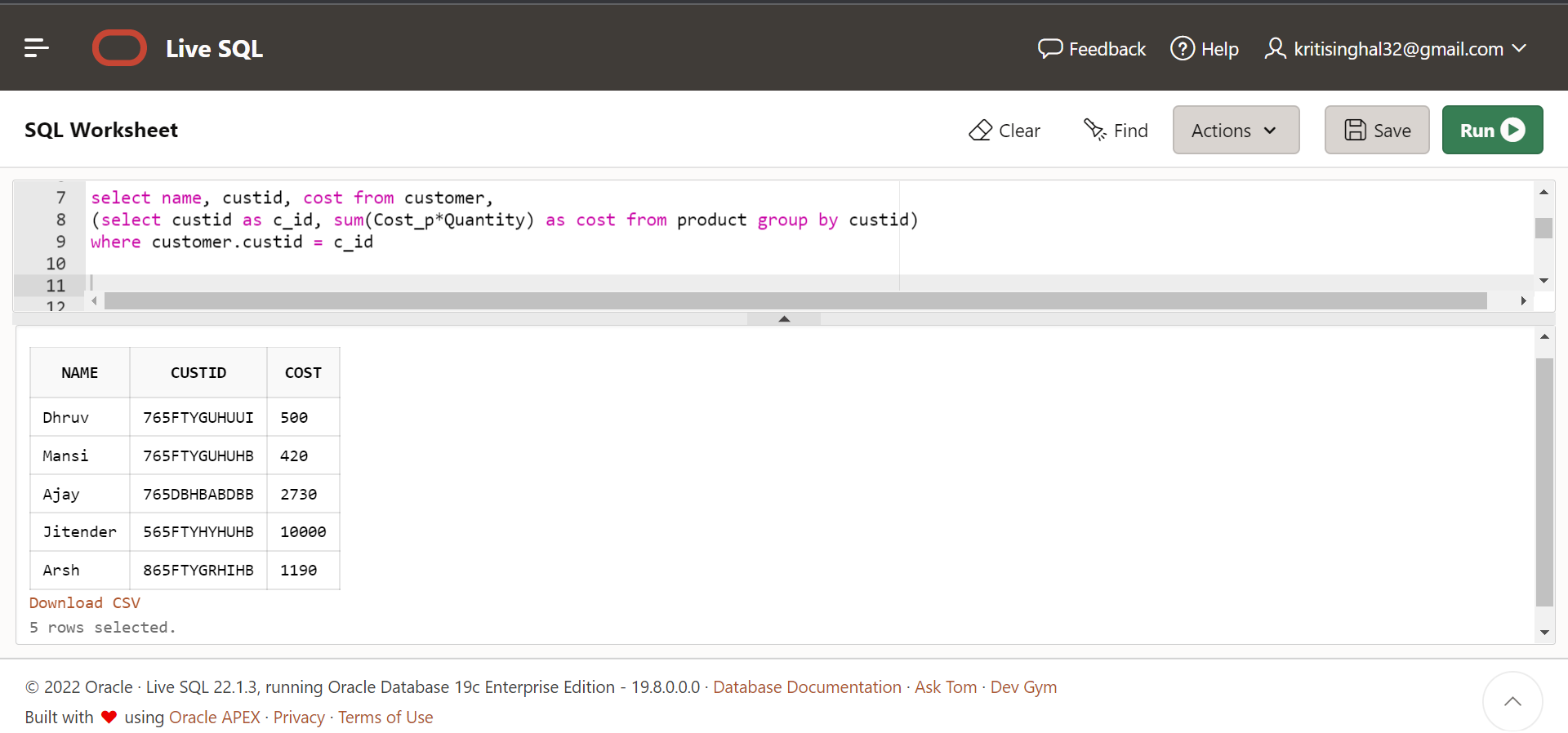


1. **Display the total purchase by each customer**

select name, custid, cost from customer,

(select custid as c\_id, sum(Cost\_p\*Quantity) as cost from product group by custid)

where customer.custid = c\_id

****

1. **Create a view to display the payment details for today**

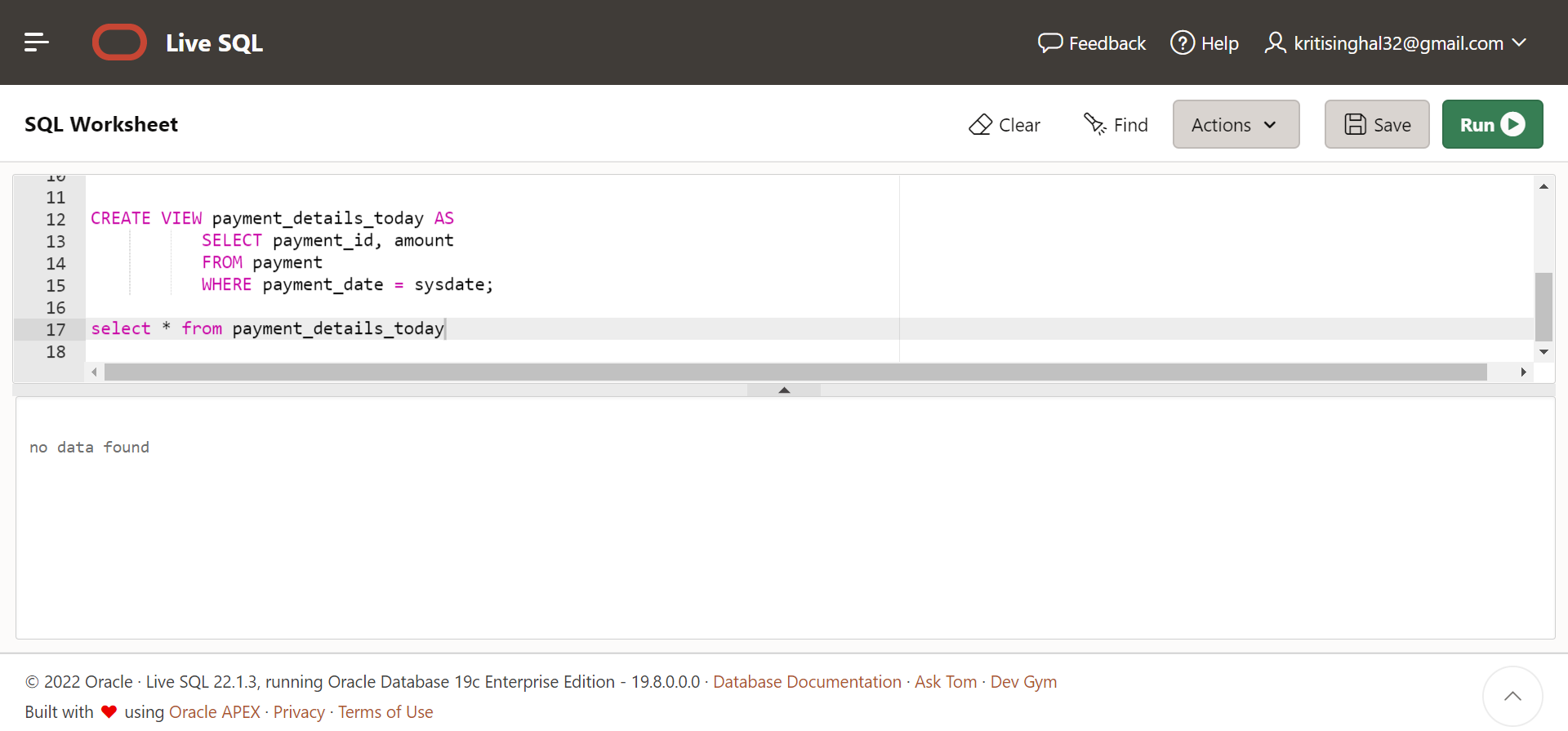
CREATE VIEW payment\_details\_today AS

SELECT payment\_id, amount

FROM payment

WHERE payment\_date = sysdate;

select \* from payment\_details\_today

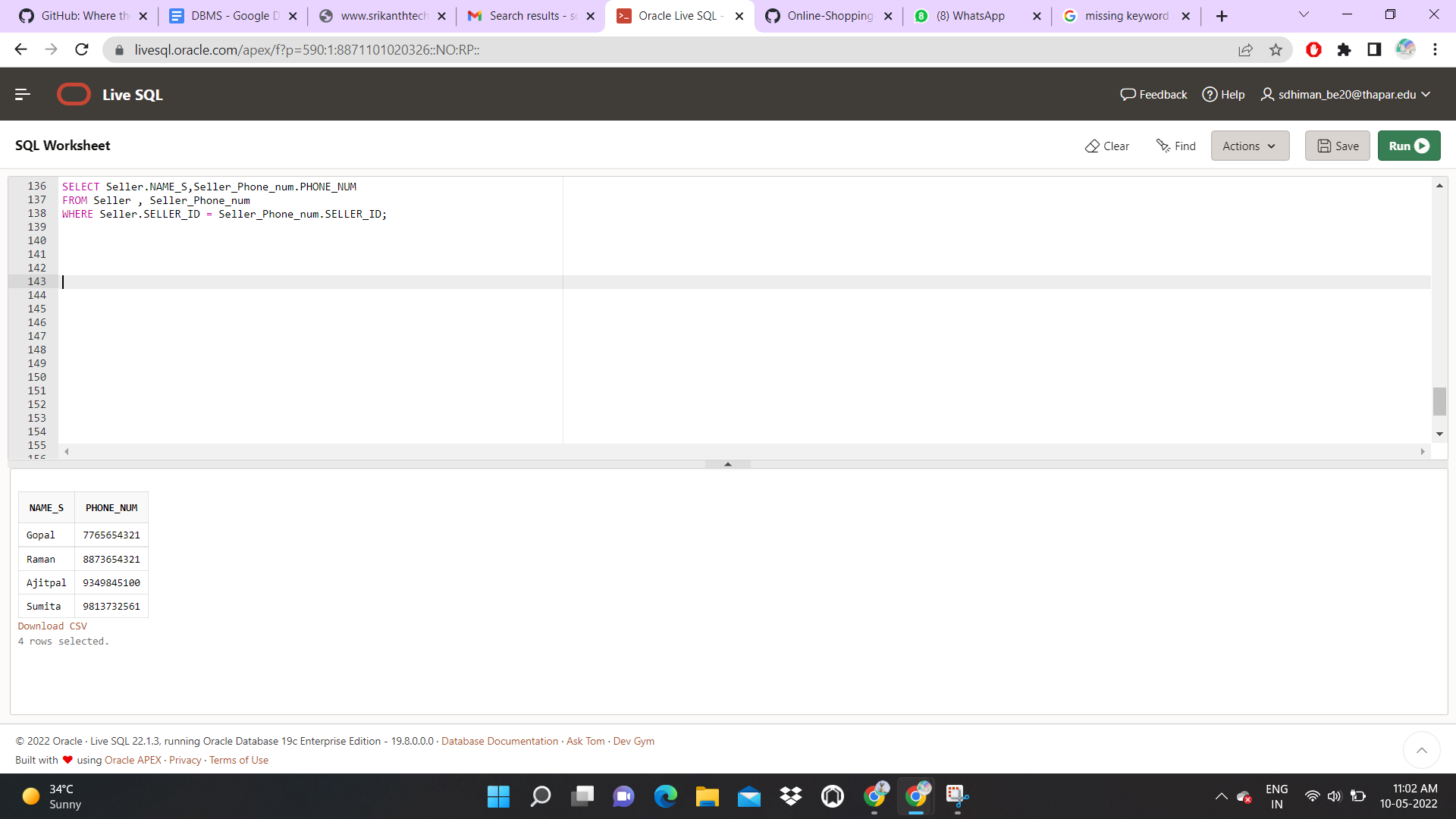


1. **Displaying all contact details for each seller**

SELECT Seller.NAME\_S,Seller\_Phone\_num.PHONE\_NUM

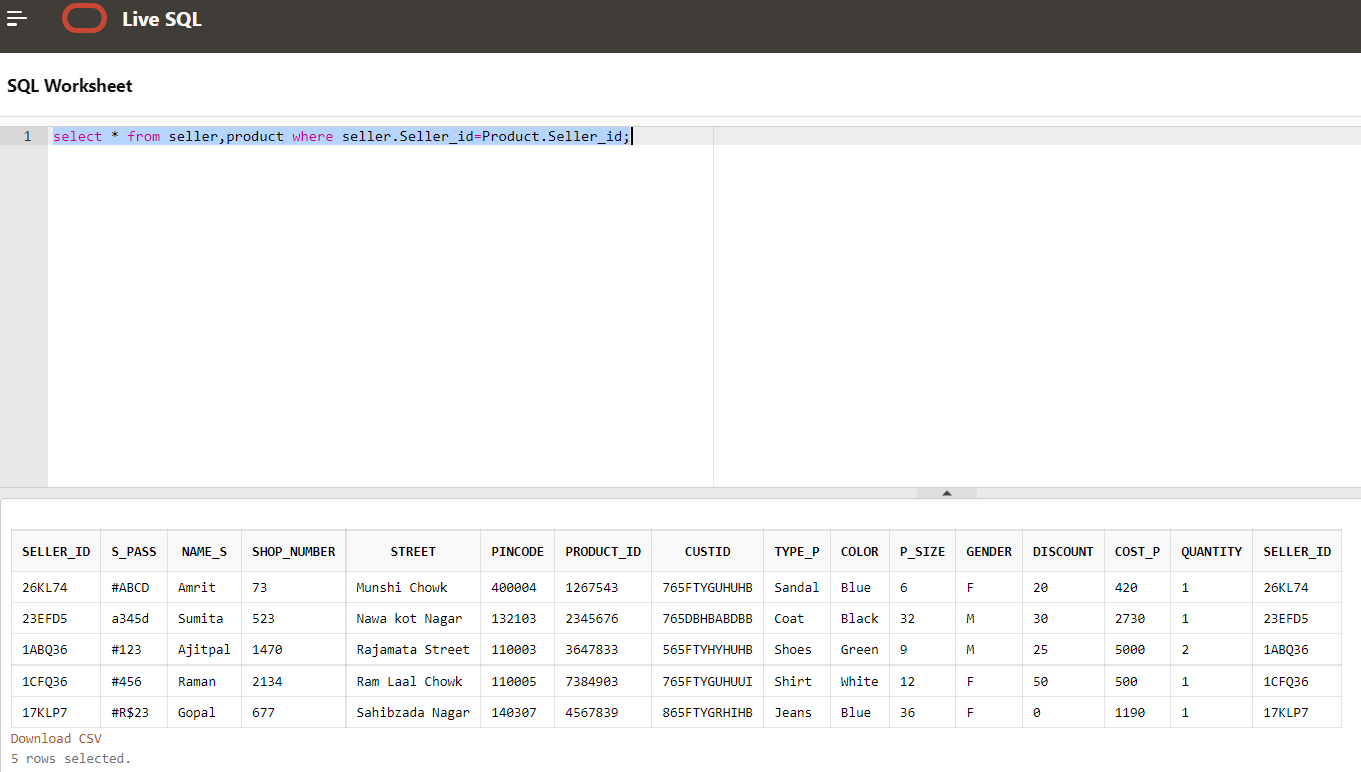
FROM Seller , Seller\_Phone\_num

WHERE Seller.SELLER\_ID = Seller\_Phone\_num.SELLER\_ID;

****

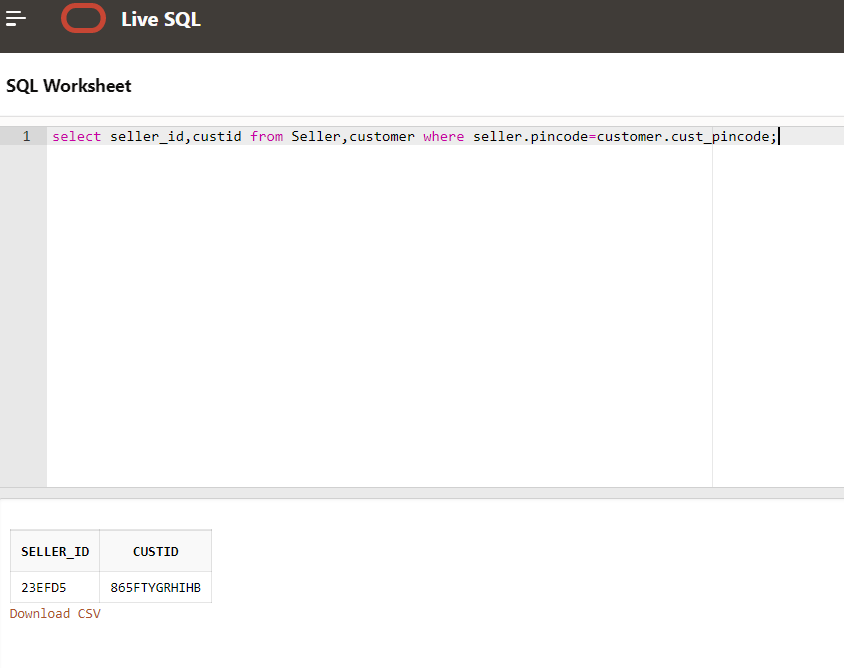
1. **Display details of seller and products sold by him**

select \* from seller,product where seller.Seller\_id=Product.Seller\_id;



1. **Select ids of seller and customer who belong to same place**

select seller\_id,custid from Seller,customer where seller.pincode=customer.cust\_pincode;



**PL/SQL:**

1. **Displaying a detailed view of each product of a particular product**

Declare

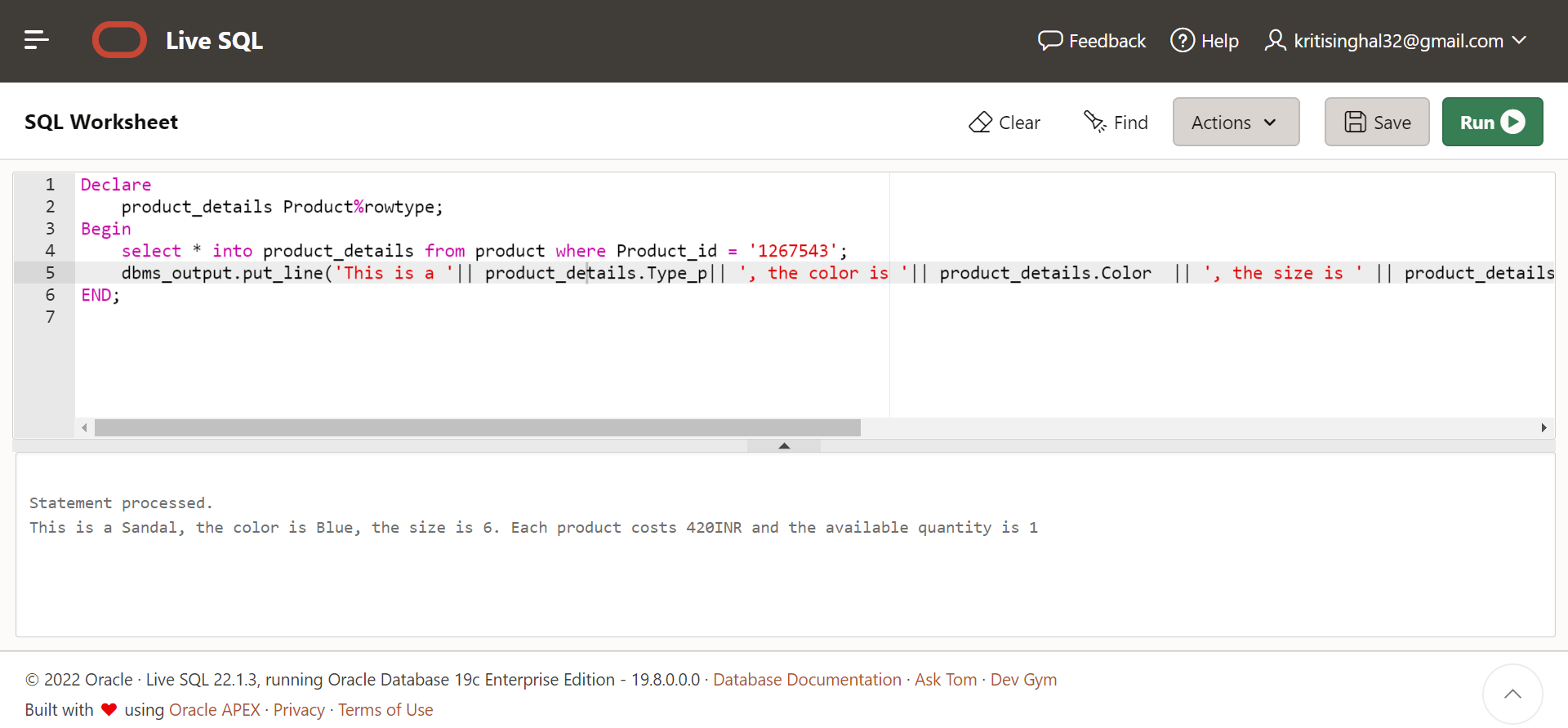
product\_details Product%rowtype;

Begin

select \* into product\_details from product where Product\_id = '1267543';

dbms\_output.put\_line('This is a '|| product\_details.Type\_p|| ', the color is '|| product\_details.Color || ', the size is ' || product\_details.P\_Size || '. Each product costs '|| product\_details.Cost\_p || 'INR and the available quantity is '|| product\_details.Quantity);

END;



1. **Error Messages displayed when too many rows or no data is selected (Inbuilt Exception Handling)**

DECLARE

product\_size VARCHAR(2);

BEGIN

select P\_Size from product into product\_size where Type\_p = 'Shoes';

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('TOO MANY ROWS');

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('NO DATA FOUND');

END;

DECLARE

product\_size VARCHAR(2);

BEGIN

select P\_Size into product\_size from product where Type\_p = 'Pant';

EXCEPTION

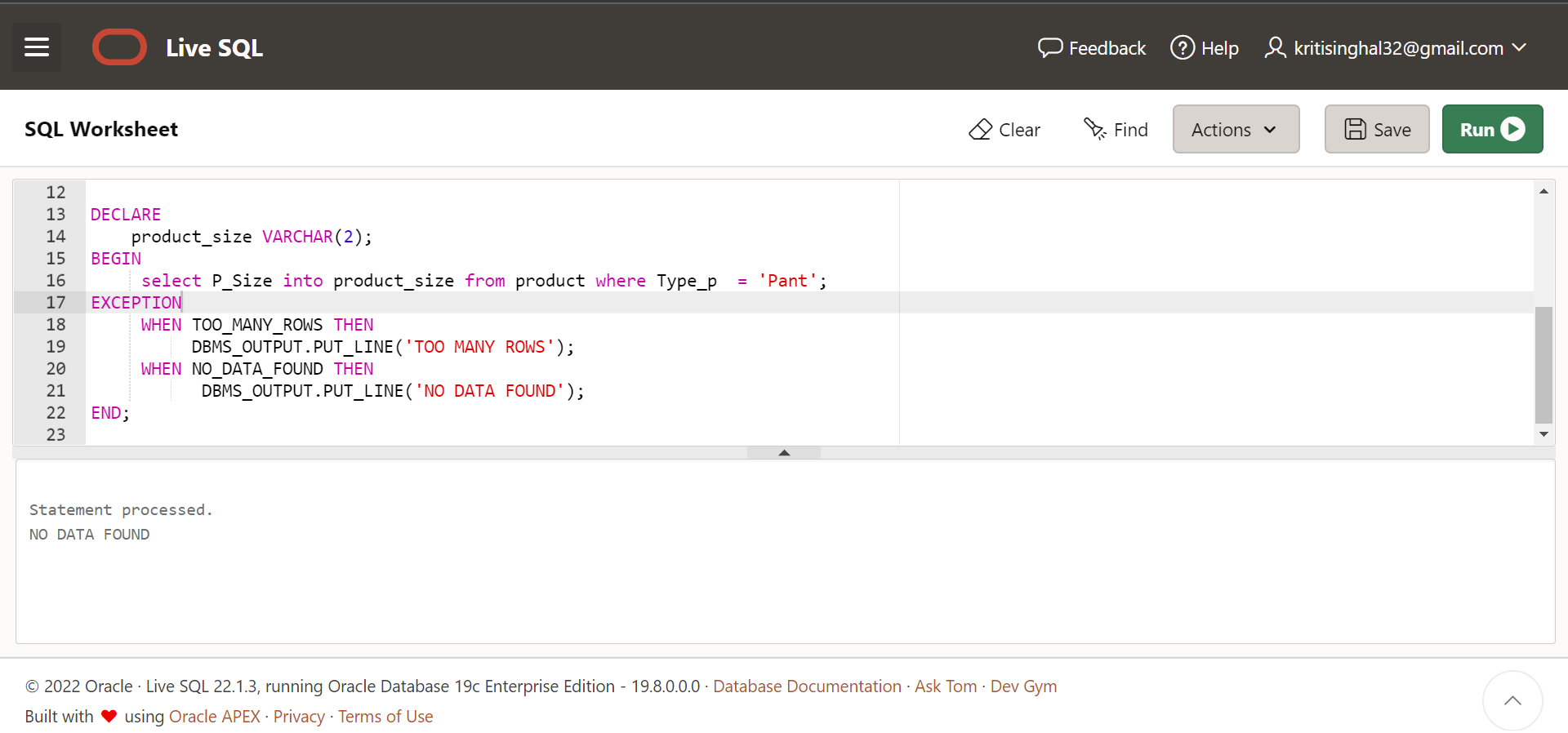
WHEN TOO\_MANY\_ROWS THEN

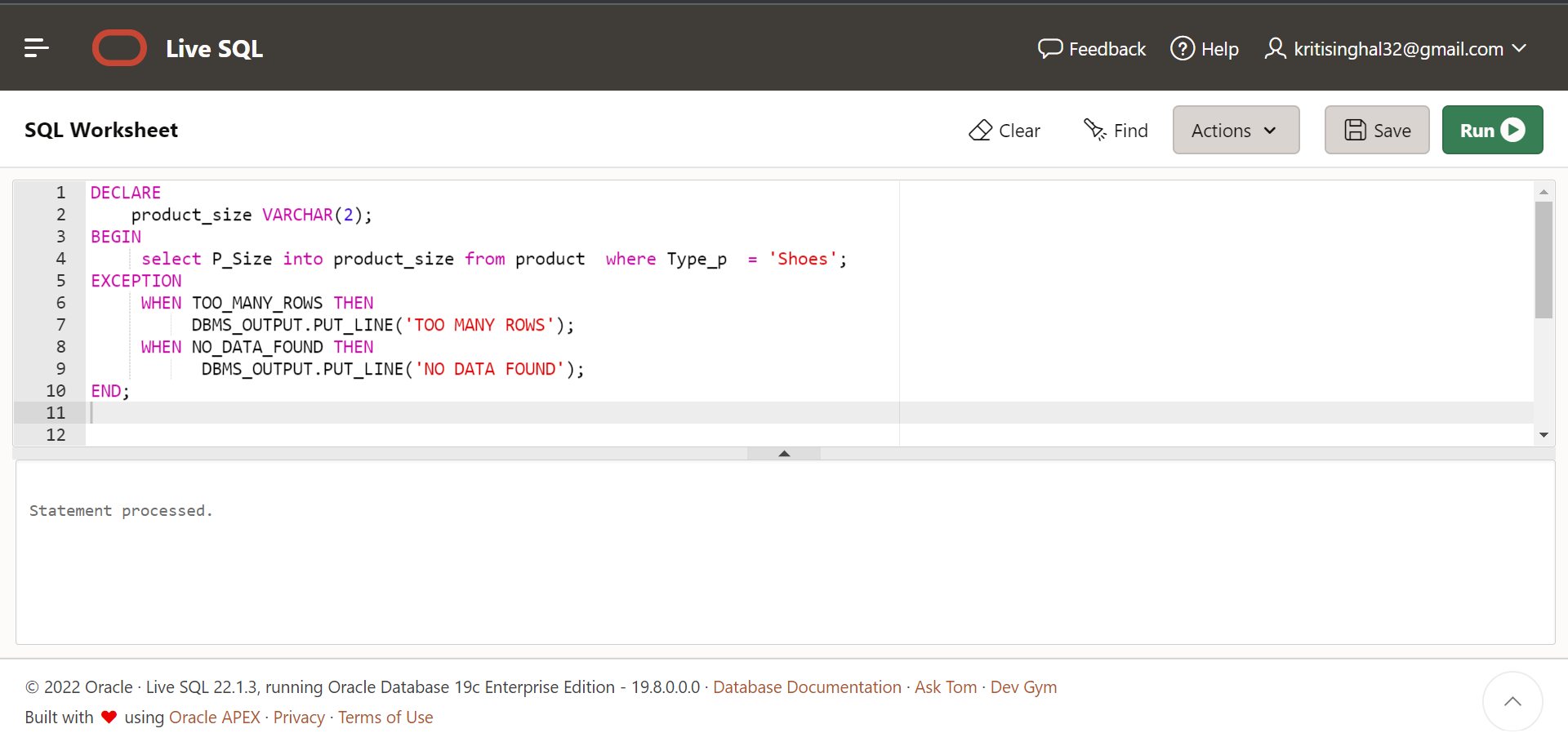
DBMS\_OUTPUT.PUT\_LINE('TOO MANY ROWS');

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('NO DATA FOUND');

END;





1. **Error Messages displayed when a wrong product id with too many characters is inserted (Non- predefined Exception Handling)**

DECLARE

Incorrect\_id EXCEPTION;

PRAGMA EXCEPTION\_INIT(Incorrect\_id ,-12899);

BEGIN

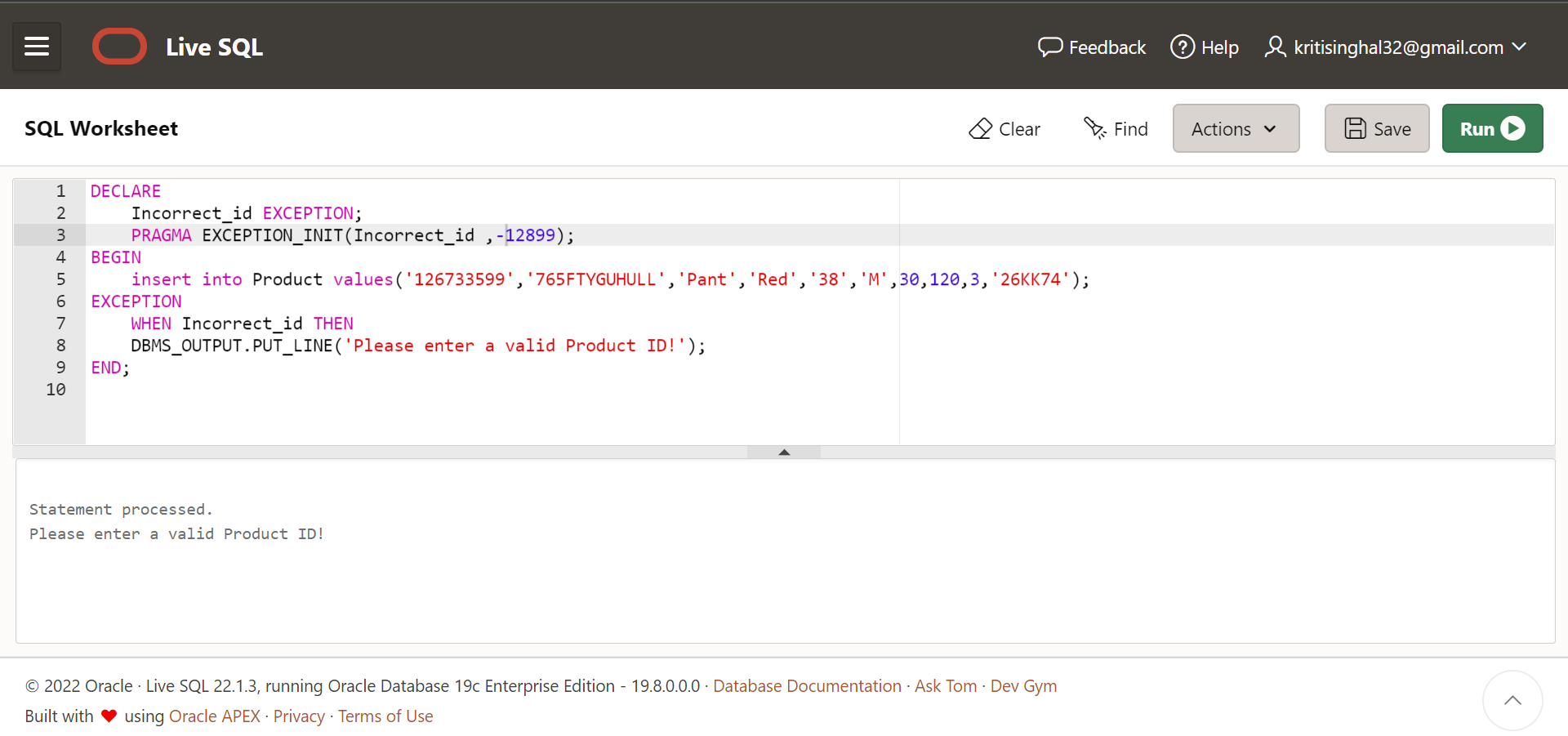
insert into Product values('126733599','765FTYGUHULL','Pant','Red','38','M',30,120,3,'26KK74');

EXCEPTION

WHEN Incorrect\_id THEN

DBMS\_OUTPUT.PUT\_LINE('Please enter a valid Product ID!');

END;



1. **Display all the products that are available in the shop**

DECLARE

cs product.Cost\_p%type;

ty product.Type\_p%type;

id product.product\_id%type;

cursor cf is

select product\_id,Cost\_p,Type\_p from product;

BEGIN

open cf;

loop

fetch cf into id,cs,ty;

exit when cf%notfound;

dbms\_output.put\_line('Product' || id || 'has cost ' || cs || ' and the type is ' || ty);

end loop;

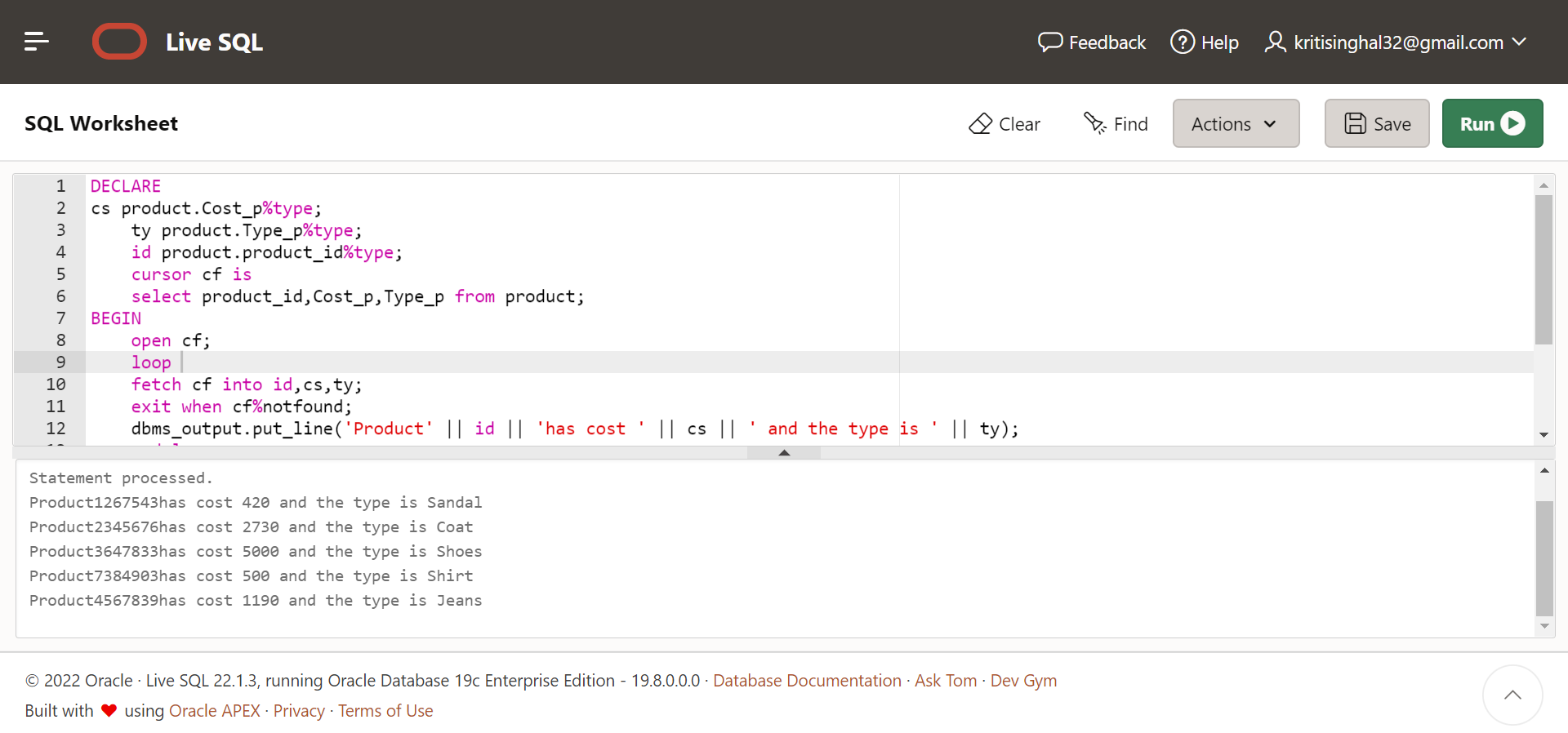
close cf;

EXCEPTION

when no\_data\_found then

dbms\_output.put\_line('No products available');

END;

****

1. **Raise error if product being ordered has a quantity of 0 (User defined Exception Handling)**

Declare

product\_details Product%rowtype;

out\_of\_stock exception;

Begin

select \* into product\_details from product where Product\_id = '1267543';

if product\_details.Quantity <= 0 THEN

RAISE out\_of\_stock;

end if;

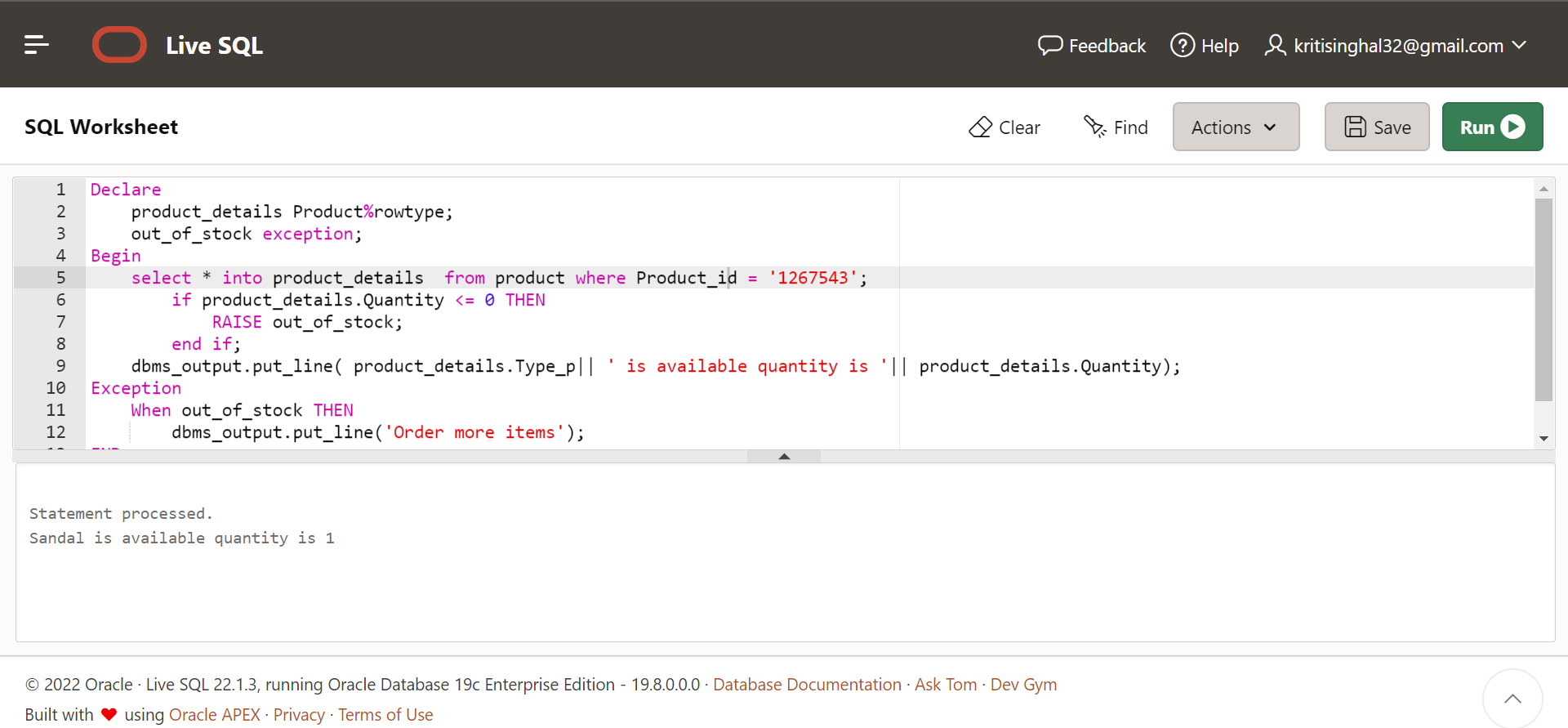
dbms\_output.put\_line( product\_details.Type\_p|| ' is available quantity is '|| product\_details.Quantity);

Exception

When out\_of\_stock THEN

dbms\_output.put\_line('Order more items');

END;



1. **Cursor for loop for finding details of a seller with given seller id**

declare

cursor c(s varchar) is

select \* from seller where Seller\_id=s;

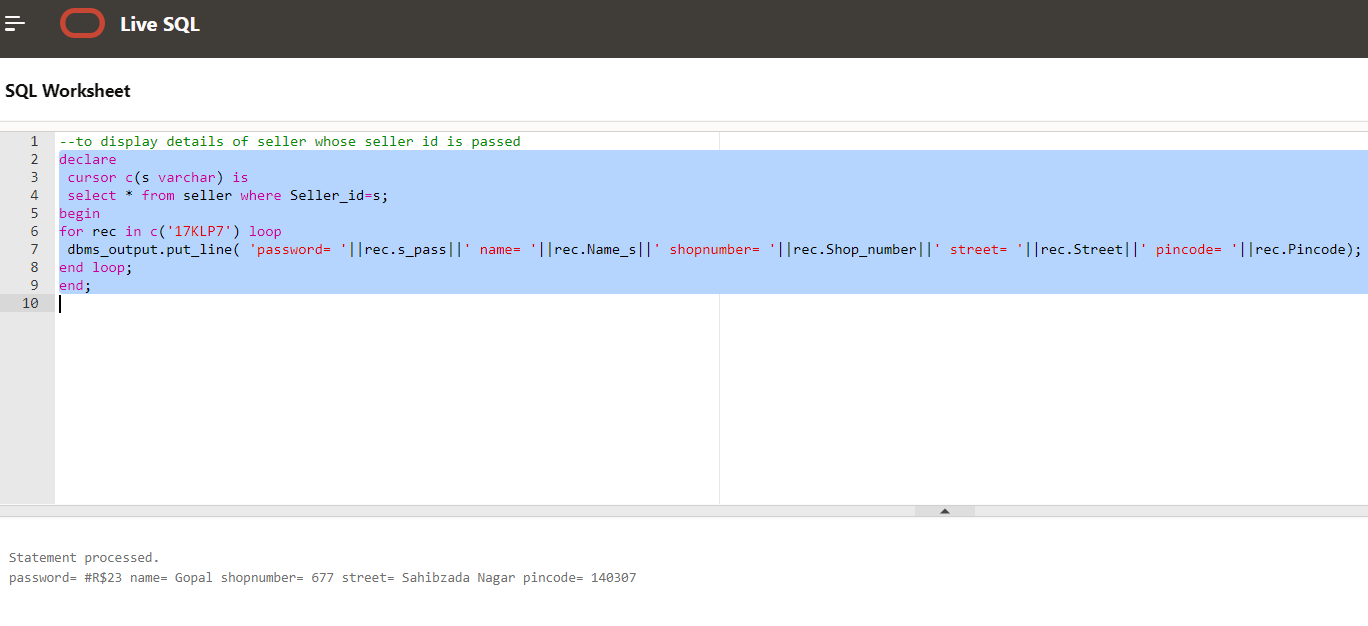
begin

for rec in c('17KLP7') loop

dbms\_output.put\_line( 'password= '||rec.s\_pass||' name= '||rec.Name\_s||' shopnumber= '||rec.Shop\_number||' street= '||rec.Street||' pincode= '||rec.Pincode);

end loop;

End;



1. **Apply 10% discount on all items using cursor**

declare

cursor c is select \* from Product;

d number

p number;

begin

for rec in c loop

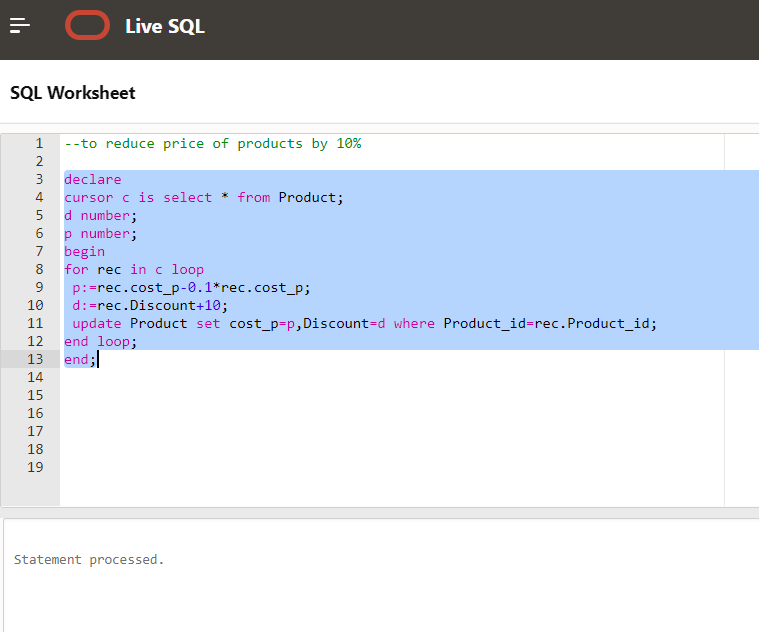
p:=rec.cost\_p-0.1\*rec.cost\_p;

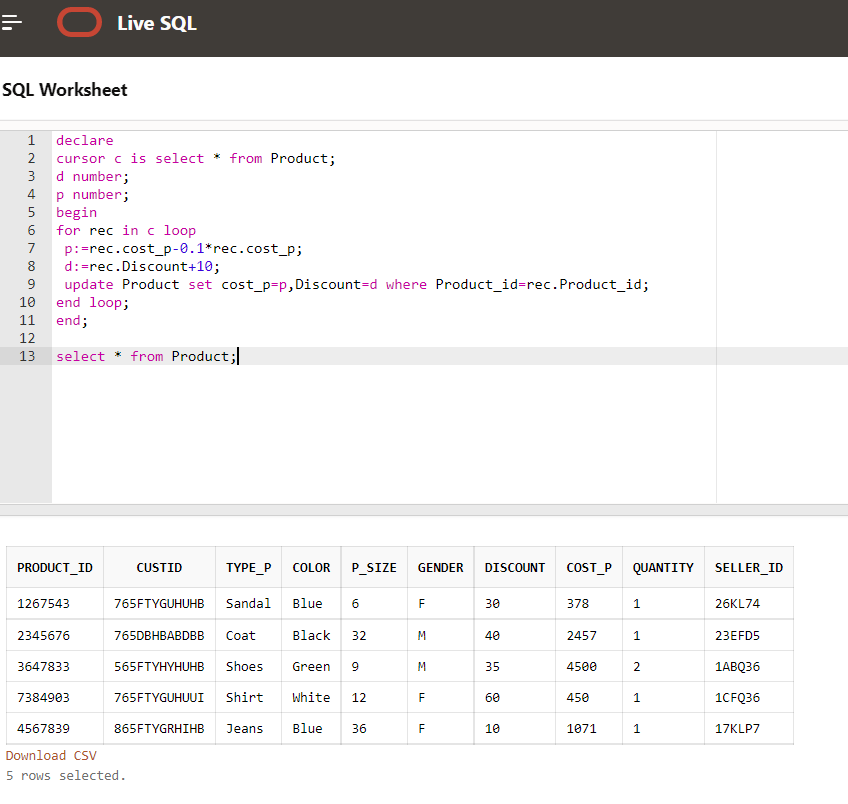
d:=rec.Discount+10;

update Product set cost\_p=p,Discount=d where Product\_id=rec.Product\_id;

end loop;

end;

****

****

1. **Procedure which returns the type of product with the cost less than the given cost**

create or replace procedure cost\_filter(c in number,t in varchar)

is

cs product.Cost\_p%type;

ty product.Type\_p%type;

id product.product\_id%type;

cursor cf is

select product\_id,Cost\_p,Type\_p from product where Cost\_p<c and Type\_p=t;

begin

open cf;

loop

fetch cf into id,cs,ty;

exit when cf%notfound;

dbms\_output.put\_line('Product' || id || 'has cost ' || cs || ' and the type is' || ty);

end loop;

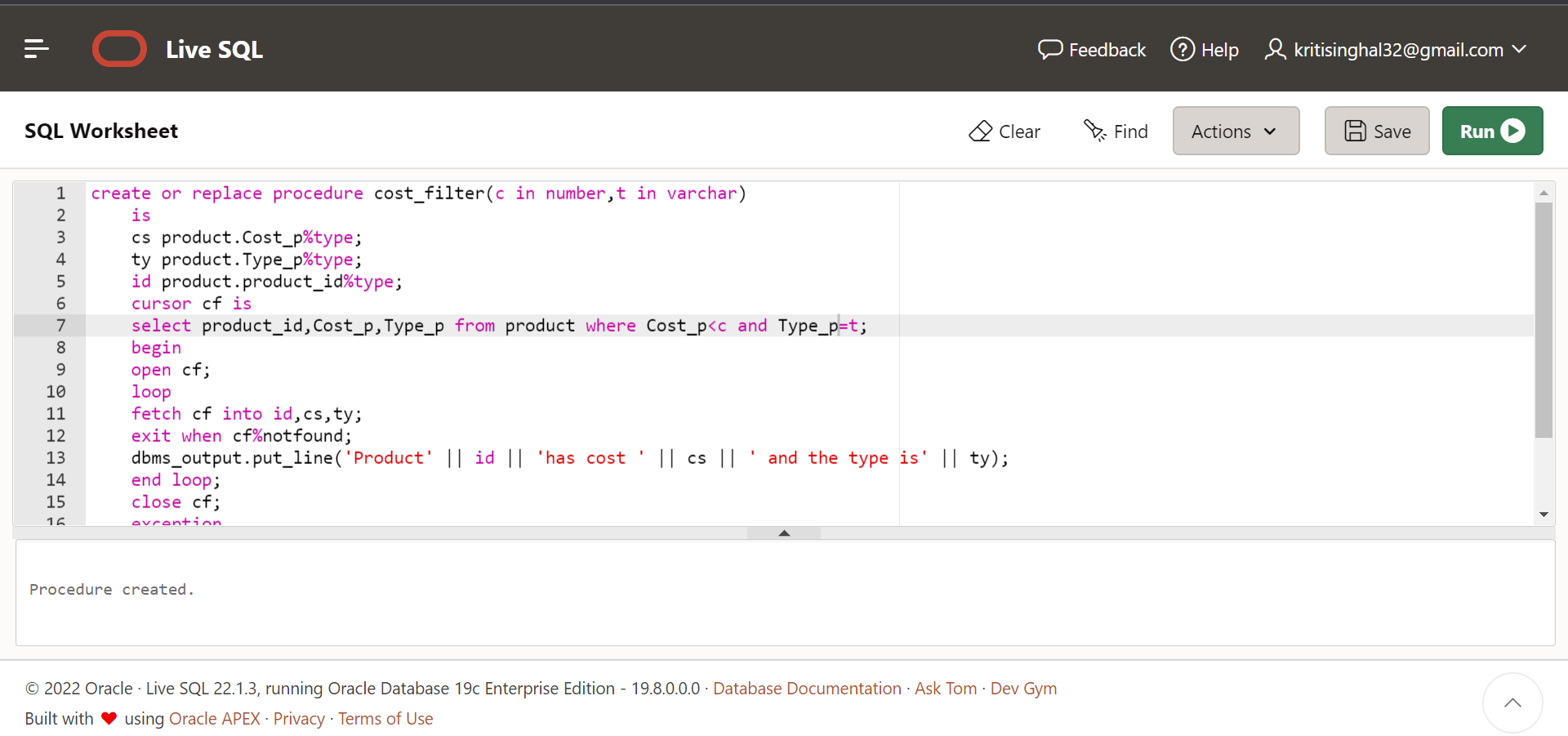
close cf;

exception

when no\_data\_found then

dbms\_output.put\_line('Sorry no such products exist');

end;



1. **Function which returns total number of products which a particular seller sells**

create or replace function totalProducts(sId in varchar)

return number

is

total number(2):=0;

begin

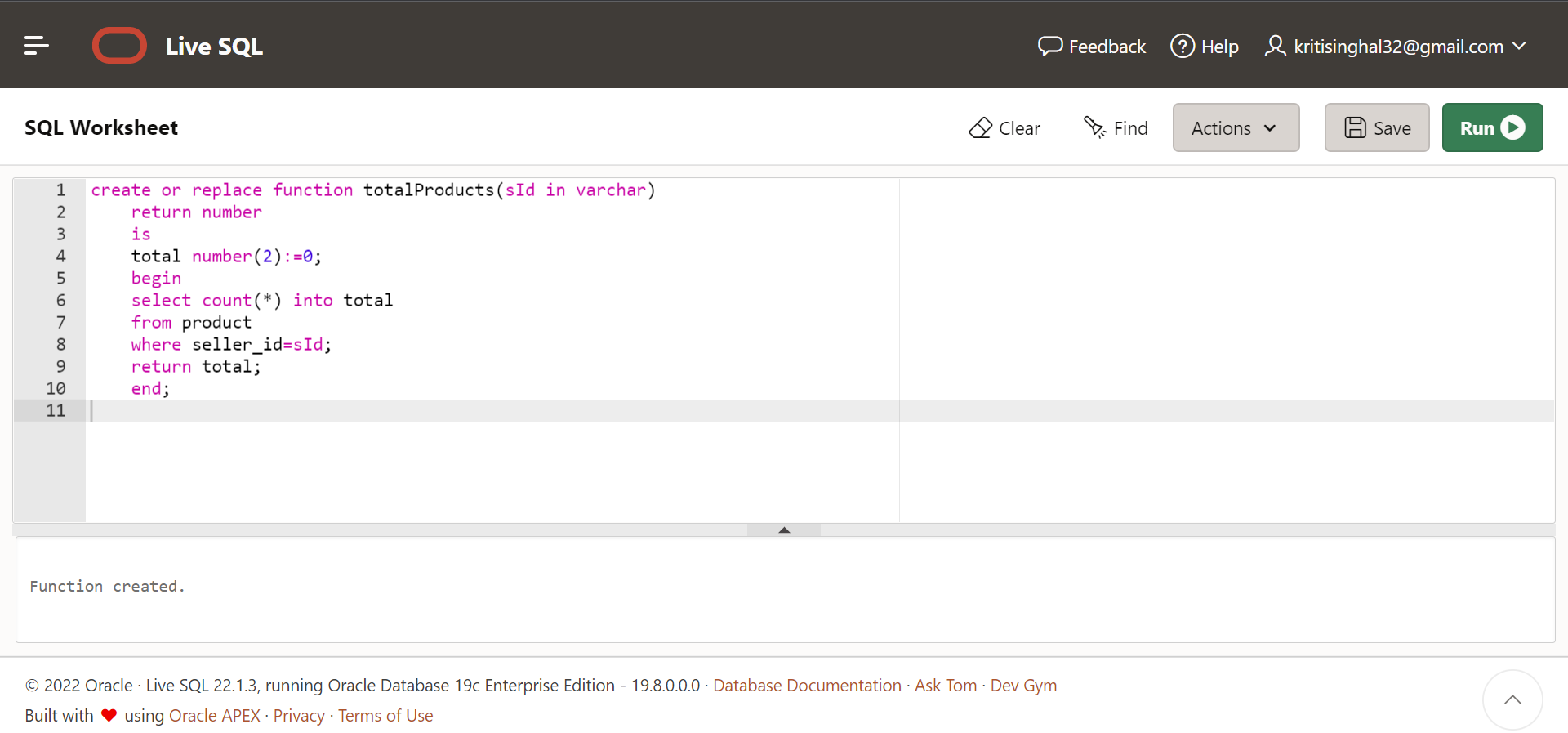
select count(\*) into total

from product

where seller\_id=sId;

return total;

end;



**Function execution:**

declare

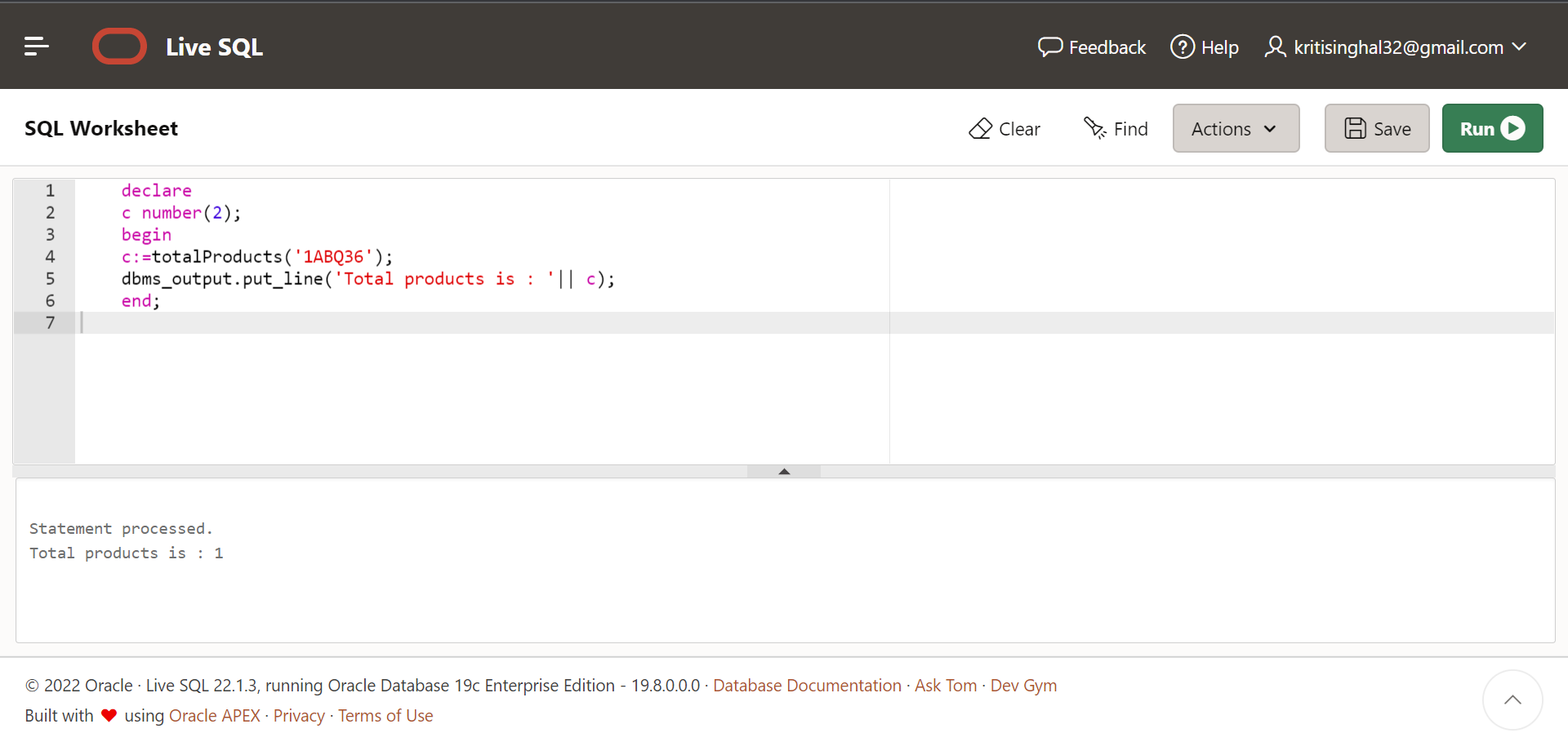
c number(2);

begin

c:=totalProducts('1ABQ36');

dbms\_output.put\_line('Total products is : '|| c);

end;



1. **Procedure which returns the total quantity of product with the given ID (Procedure with exception handling)**

create or replace procedure prod\_details(p\_id in varchar)

is

quan number(2);

begin

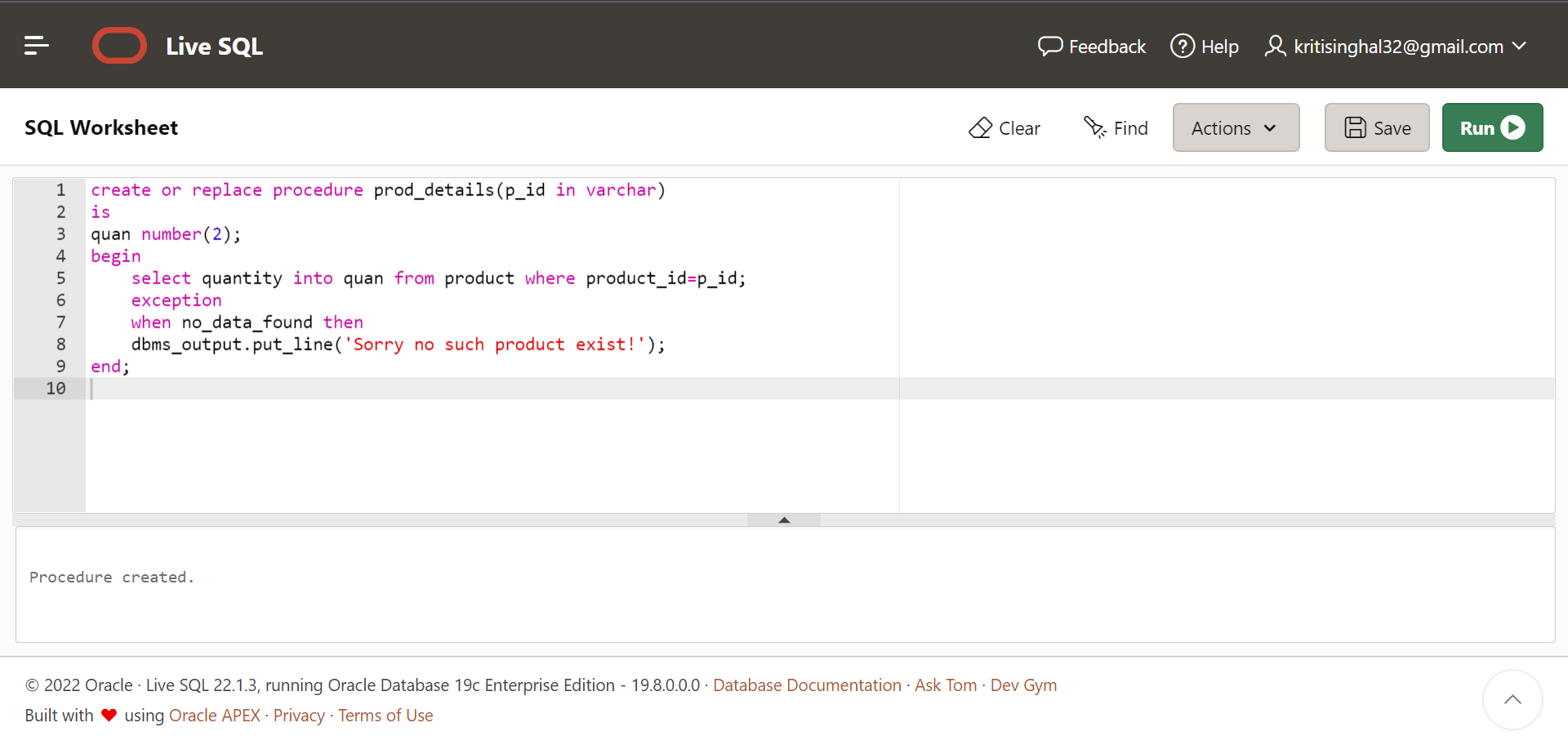
select quantity into quan from product where product\_id=p\_id;

exception

when no\_data\_found then

dbms\_output.put\_line('Sorry no such product exist !!');

end;

****

1. **Procedure to delete seller info based on sellerid**

create or replace procedure deleteseller(sellerid varchar)as

begin

delete from Seller where Seller\_id=sellerid;

end;

declare

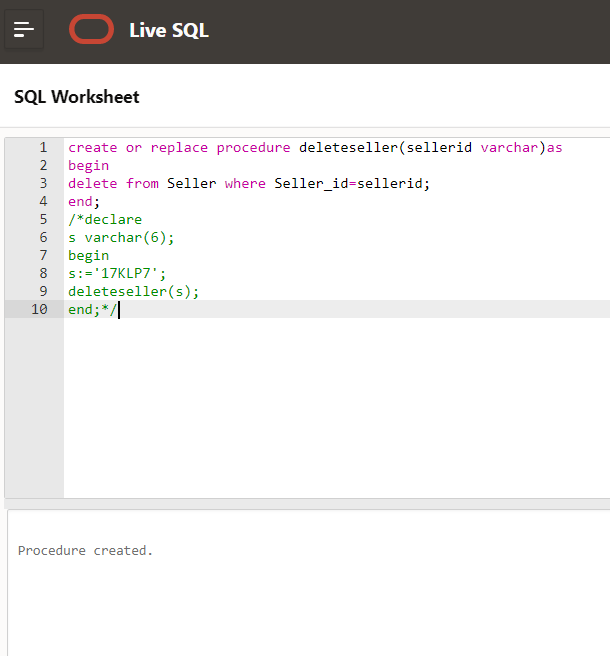
s varchar(6);

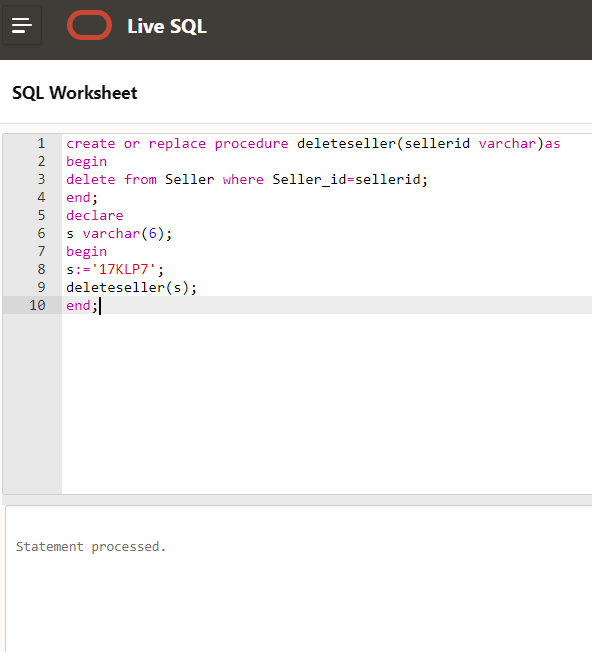
begin

s:='17KLP7';

deleteseller(s);

end;

****

****

1. **Triggers:**

**Trigger that will execute before inserting new customer to database and inserting a new cartId to the cart\_items table (Function to count number of cart items)**

create or replace function numCartId(cd in varchar)

return number

is

total number(2):=0;

begin

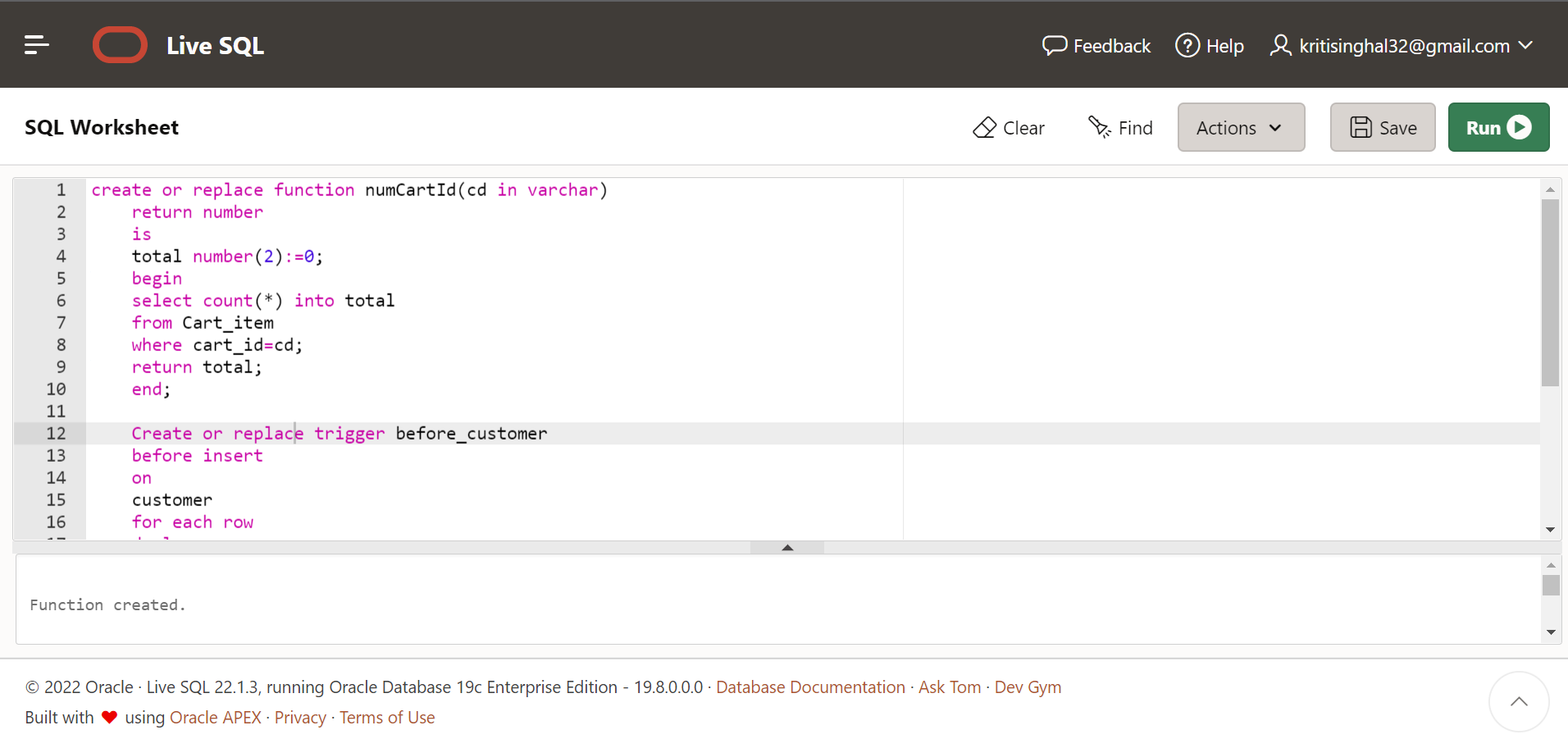
select count(\*) into total

from cart\_item

where cart\_id=cd;

return total;

end;



Create or replace trigger before\_customer

before insert

on

customer

for each row

declare

c varchar(10);

n number(2);

begin

c:= :new.cart\_id;

n:=numCartId(c);

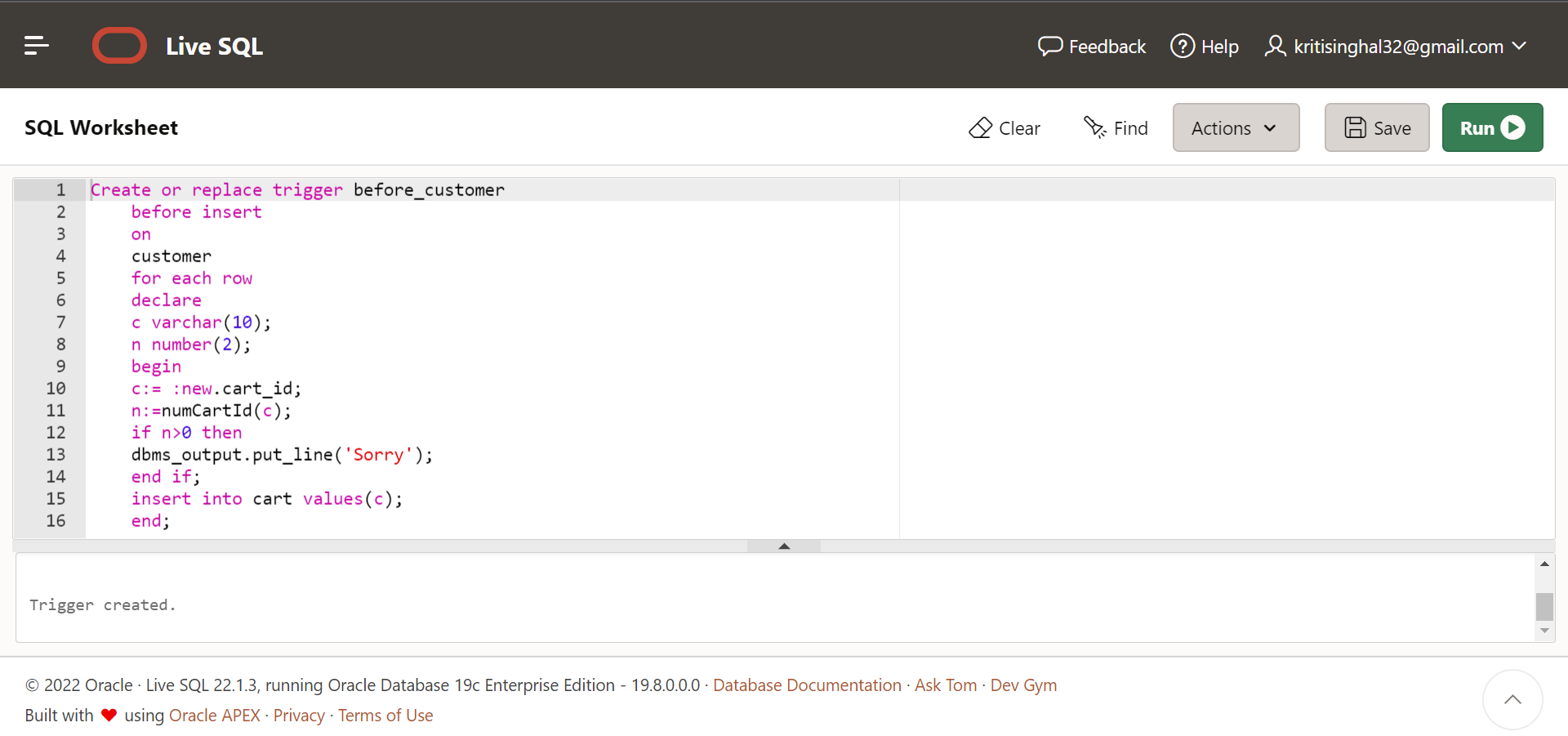
if n>0 then

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

end if;

insert into cart values(c);

end;



1. **Trigger to update the total amount of user everytime he adds something to payment table**

create or replace function total\_cost(cId in varchar)

return number

is

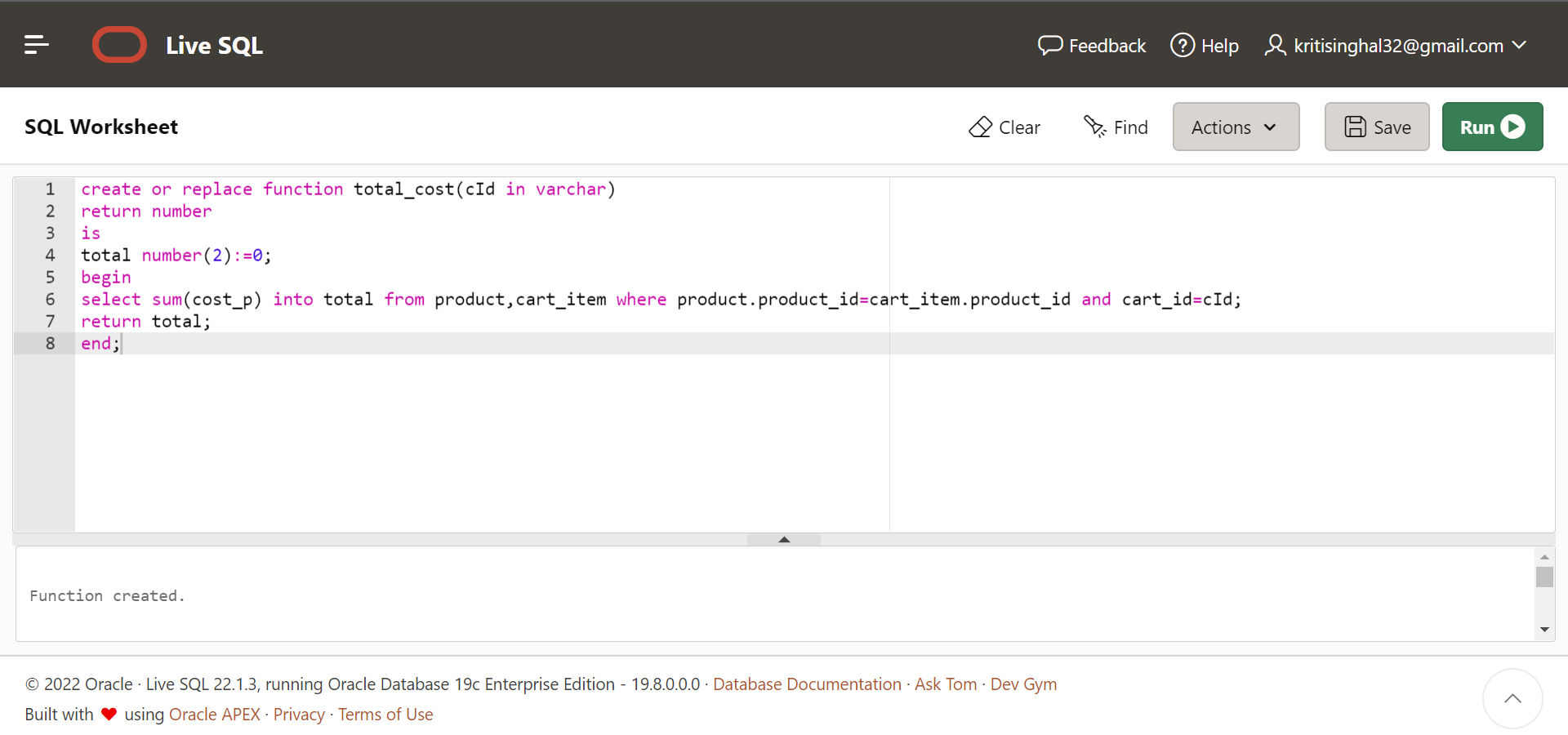
total number(2):=0;

begin

select sum(cost\_p) into total from product,cart\_item where product.product\_id=cart\_item.product\_id and cart\_id=cId;

return total;

end;



create or replace trigger before\_pay\_up

before insert

on

payment

for each row

declare

total number(3);

begin

total :=total\_cost(:new.custid);

insert into payment values(:new.payment\_id,total,:new.payment\_date,:new.custid);

end;

