ENDTERM

210103366 Bazarbayeva Gaukhar
210103210 Kuanova Aidana

BOOKTOPIA

We have chosen an online bookshop.

What is the essence of our project? A customer enters the store and makes an order. After making an order, he proceeds to payment. After payment, it is sent to the bank. The bank verifies transactions. If the payment is accepted, it is sent to the administrator. The administrator, after receiving accepted, sends the order to the customer's address using a courier.

TABLES

Admin (admin_id, name, lastname, role)

Online_shop (shop_name, url_website)

Product_category (category_id, c_name)

Product (product_id, category_id, p_name, price)

Customer (customer_number, customer_name, customer_lname, customer_contact, customer_address)

Order (order_number, customer_number, amount, order_date, product_id)

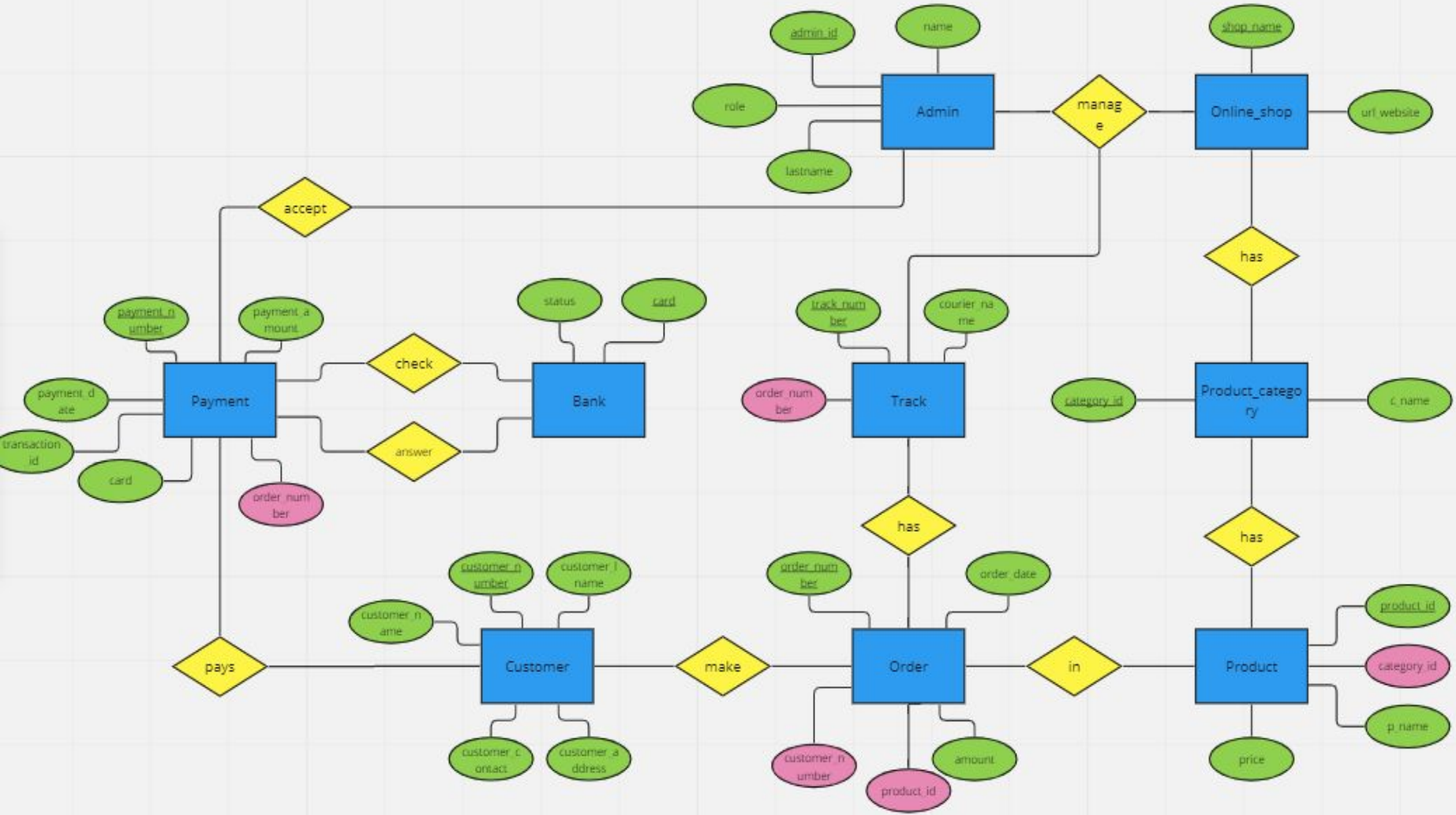
Payment (payment_number, order_number, payment_amount, payment_date, transaction_id, card)

Bank (card, status)

Track (track_number, order_number, courier_name)

ERD DIAGRAM

https://miro.com/app/board/uXjVMQSx8iQ=/


admin_id
name
role
Admin
lastname
manag
e
shop_name
Online_shop
url website
accept
has
payment_n
umber
payment_a
mount
status
card
check
payment_d
ate
Payment
Bank
transaction
id
answer
card
order_num
ber
track_num
ber
courier_na
me
order_num
ber
Track
category_id
Product_catego
ry
c_name
has
has
customer_n
umber
customer_l
name
order_num
ber
order_date
customer_n
ame
pays
Customer
make
Order
in
Product
product_id
category_id
p_name
customer_c
ontact
customer_a
ddress
customer_n
umber
amount
product_id
price

RELATION BETWEEN TABLES

1.Relationship between Admin and Online_shop 1:1

2. Relationship between Online_shop and Product_category 1:M

3. Relationship between Product_category and Product 1:M

4. Relationship between Product and Order M:1

5. Relationship between Customer and Order 1:M

6. Relationship between Customer and Payment 1:M

7. Relationship between Payment and Bank M:M

8. Relationship between Order and Track 1:1

9. Relationship between Admin and Payment 1:M

10 Relationship between Admin and Track 1:M

NORMALIZATION

1)

admin_id	name	lastname	role
1	Gaukhar	Bazarbayeva	manage
2	Aidana	Kuanova	manage

admin_id ⟶ name

admin_id ⟶ lastname

admin_id ⟶ role

admin_id ⟶ admin_id, name, lastname, role

2)

shop_name	url_website
booktopia	http://booktopia.com

shop_name ⟶ url_website

3)

category_id	c_name
1101	fantasy
1102	drama
1103	horror
1104	comedy
1105	detective

category_id ⟶ c_name

4)

category_id	product_id	p_name	price
1101	110167	Six of Crows	4500
1101	110124	Stardust	7600
1101	110118	The Last Unicorn	3800
1101	110143	The Fifth Season	5500
1101	110139	Kindred	8700

category_id	product_id
1101	110167
1101	110124
1101	110118
1101	110143
1101	110139

product_id	p_name	price
110167	Six of Crows	4500
110124	Stardust	7600
110118	The Last Unicorn	3800
110143	The Fifth Season	5500
110139	Kindred	8700

category_id ⟶ product_id

product_id ⟶ p_name

product_id ⟶ price

product_id ⟶ p_name, price

5)

customer_number	customer_name	customer_lname	customer_contact	customer_address
23456	Amelia	Carter	87013492034	California 31
23345	Vincent	Watting	87786423190	Texas 20
23445	Frazer	Masselin	87012319857	Florida 97
44533	Denise	Avon	87752973081	New York 12
23521	Shelly	Minard	87772083429	Pennsylvania 67

customer_number ➡ customer_name

customer_number ➡ customer_lname

customer_number ➡ customer_contact

customer_number ➡ customer_address

customer_number ➡ customer_name, customer_lname, customer_contact, customer_address

6)

order_number	customer_number	amount	order_date	product_id
21003	23456	4500	11 april 12:20	110167
21004	23345	7600	11 april 15:00	110124
21005	23445	3800	12 april 10:30	110118
21006	44533	5500	12 april 13:45	110143
21007	23521	8700	13 april 16:05	110139

order_number ➡ customer_number

order_number ➡ amount

order_number ➡ order_date

order_number ➡ product_id

order_number ➡ customer_number, amount, order_date, product_id

payment_numb er	order_numbe r	payment_amou nt	payment_dat e	transaction_i d	card
421	21003	4500	11 april 12:20	308	4 400 430 108 837 880
422	21004	7600	11 april 15:00	619	4 400 430 114 680 750
423	21005	3800	12 april 10:30	372	4 400 430 169 363 060

payment_number	order_number	payment_amount	payment_date	transaction_id
421	21003	4500	11 april 12:20	308
422	21004	7600	11 april 15:00	619
423	21005	3800	12 april 10:30	372

transaction_id	card
308	4 400 430 108 837 880
619	4 400 430 114 680 750
372	4 400 430 169 363 060

payment_number ➡ order_number

payment_number ➡ payment_amount

payment_number ➡ payment_date

payment_number ➡ transaction_id

payment_number ➡ order_number, payment_amount, payment_date, transaction_id

8)

card	status
4 400 430 108 837 880	accept
4 400 430 114 680 750	accept
4 400 430 169 363 060	accept
4 400 430 179 104 960	accept

Card ⟶ status

9)

track_number	order_number	courier_name
4200	21003	Tillie Yateman
4201	21004	Marnie Castells
4203	21005	Ellyn Cove
4204	21006	Nathaniel Friday
4205	21007	Heriberto Constantine

track_number ⟶ order_number

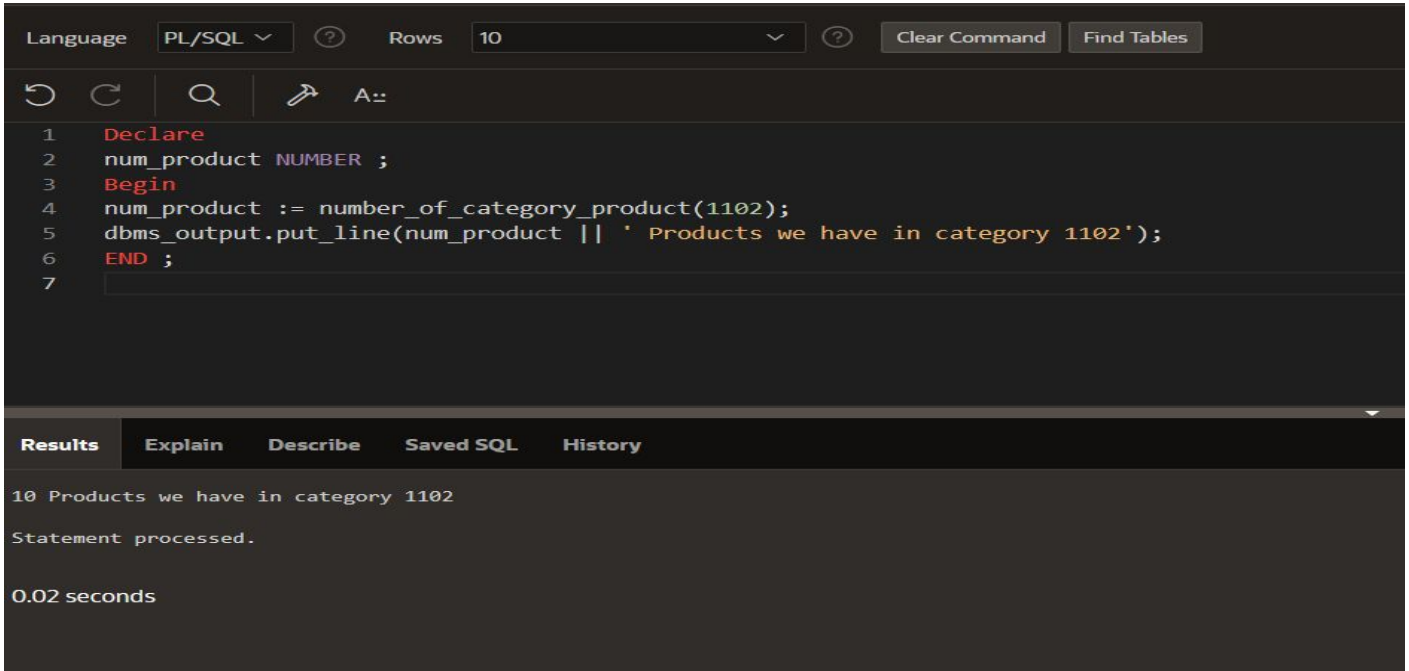
track_number ⟶ courier_name

track_number ⟶ order_number, courier_name

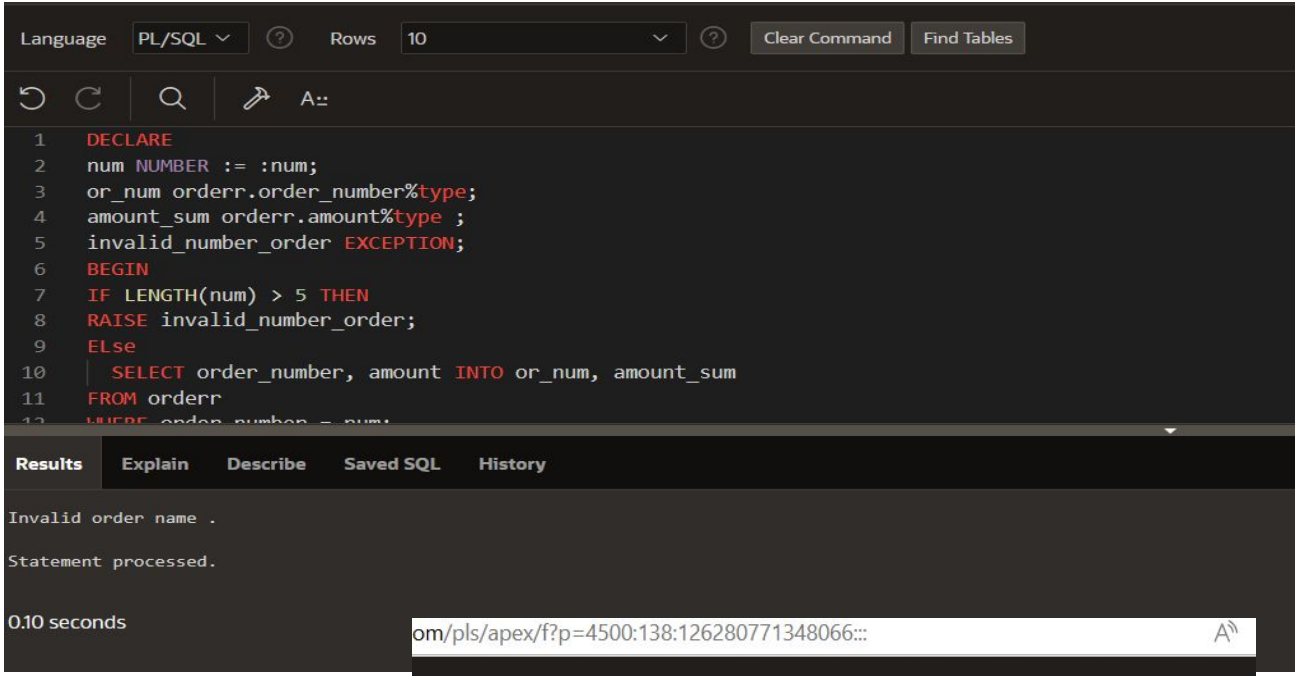
Function which counts the number of records :

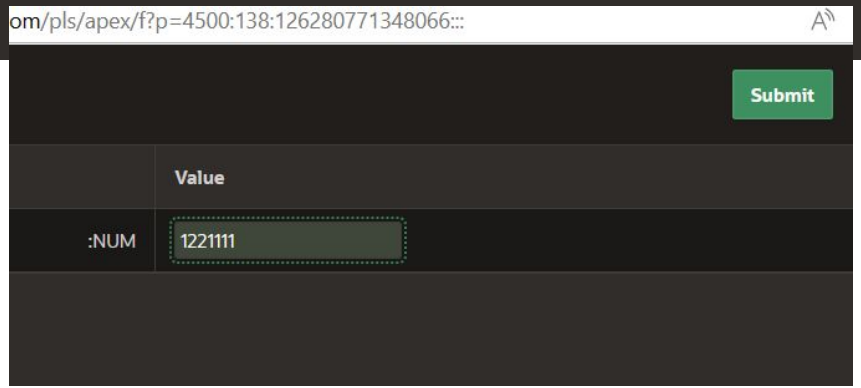
```
CREATE OR REPLACE FUNCTION number_of_category_product(c_id IN NUMBER)
RETURN NUMBER
IS num_product NUMBER;
BEGIN
 SELECT COUNT(*) INTO num_product FROM PRODUCT WHERE category_ = c_id;
Return num_product ;
END;
```

```
Declare
num_product NUMBER ;
Begin
num_product := number_of_category_product(1102);
dbms_output.put_line(num_product || ' Products we have in category 1102');
END ;
```



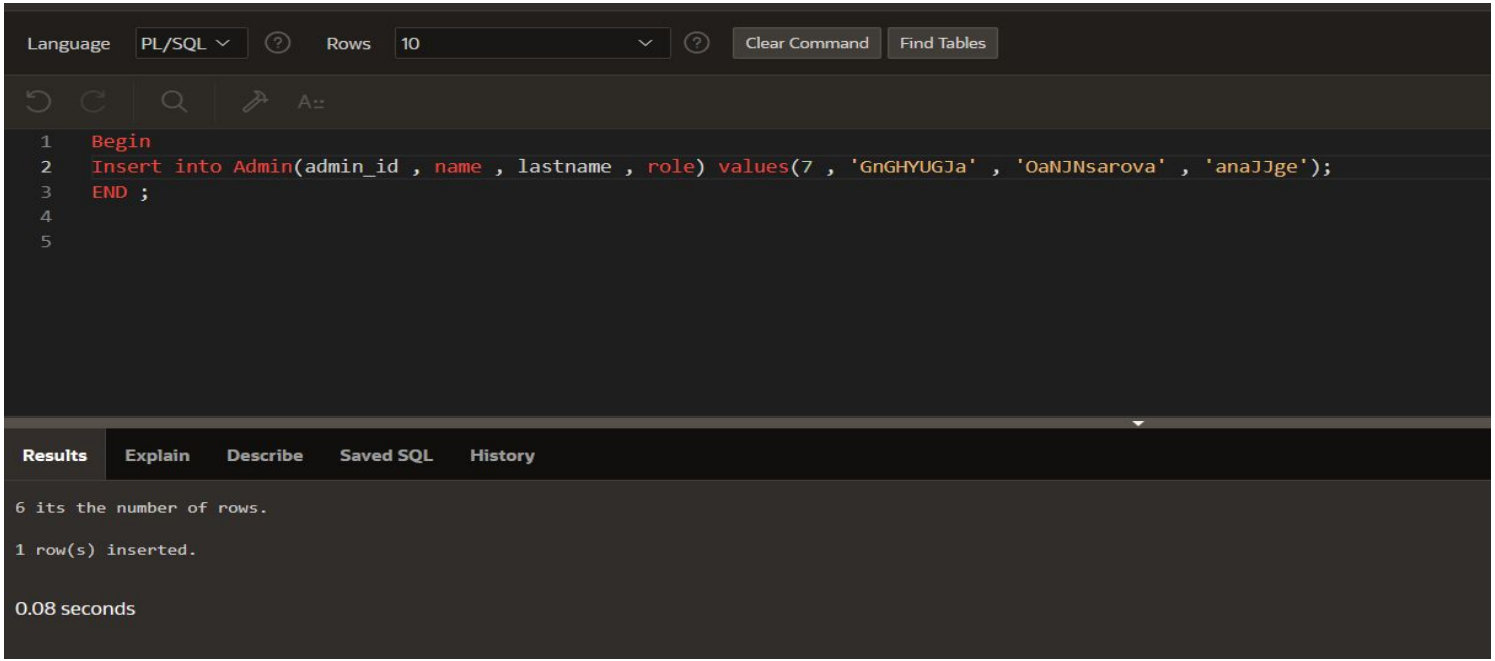
```
DECLARE
num NUMBER := :num;
or_num orderr.order_number%type;
amount_sum orderr.amount%type ;
invalid_number_order EXCEPTION;
BEGIN
IF LENGTH(num) > 5 THEN
RAISE invalid_number_order;
ELse
  SELECT order_number, amount
INTO or_num, amount_sum
FROM orderr
WHERE order_number = num;
  DBMS_OUTPUT.PUT_LINE ('Order
number is' or_num ' amount is ' ||
amount_sum);
END IF;
EXCEPTION
WHEN invalid_number_order THEN
DBMS_OUTPUT.PUT_LINE('Invalid
order name .');
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Error: ' ||
SQLERRM);
END;
```



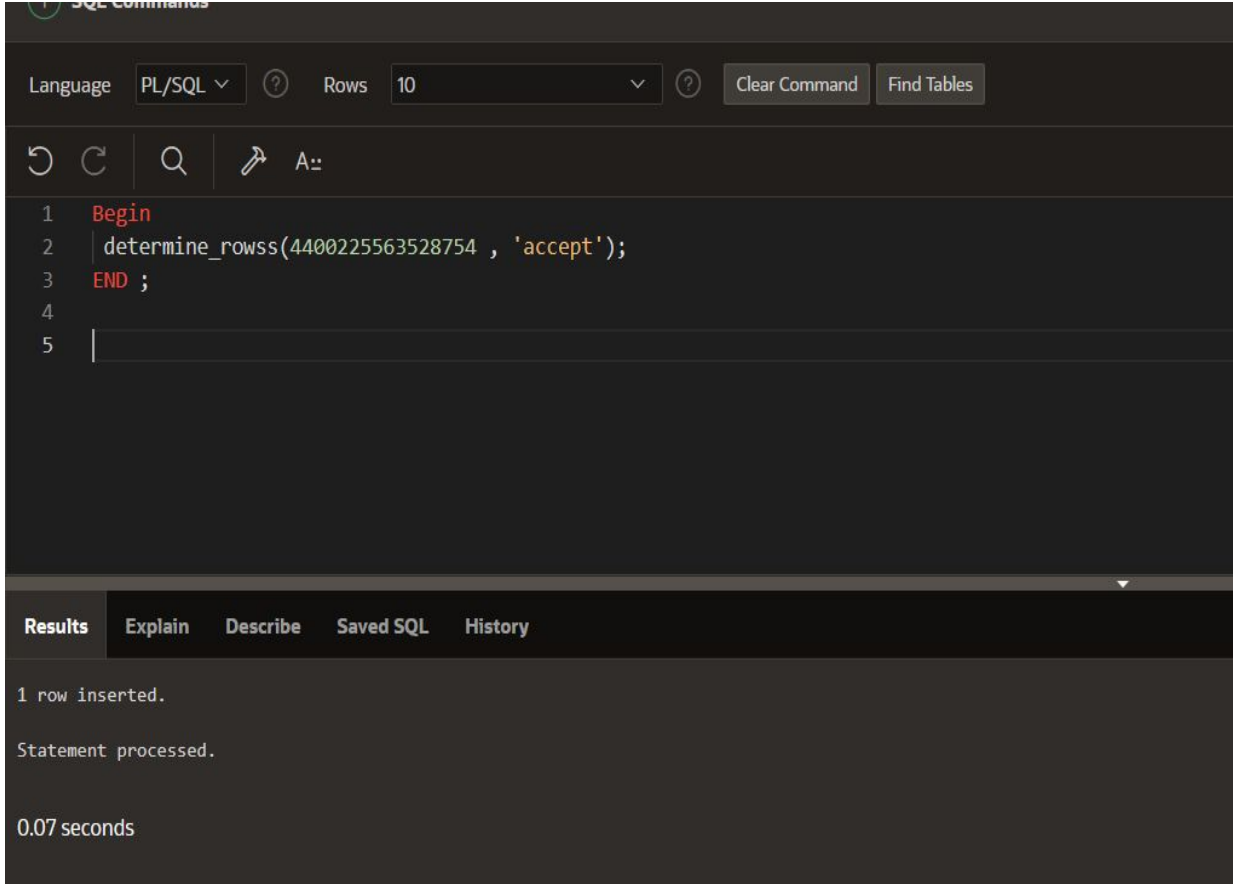


```
CREATE OR REPLACE TRIGGER currrent_rowse
BEFORE INSERT ON any_table
FOR EACH ROW
declare
row_insert NUMBER ;
BEGIN
Select Count(*) into row_insert From Admin;
DBMS_OUTPUT.PUT_LINE(row_insert || ' its the number of rows.');
END;
```

```
Begin
Insert into Admin(admin_id , name ,
lastname , role) values(6 , 'Gna' ,
'Oasarova' , 'anage');
END ;
```

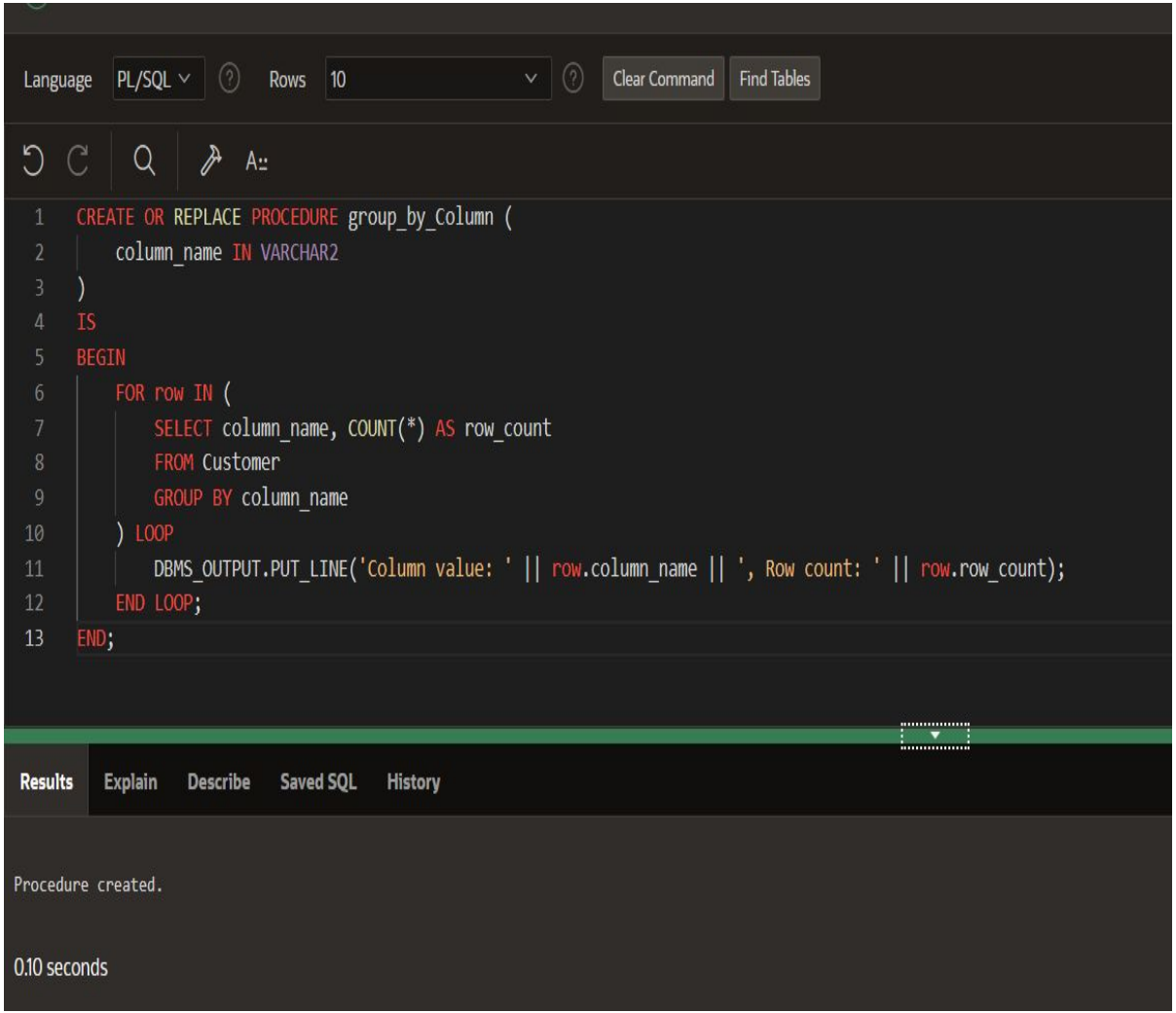


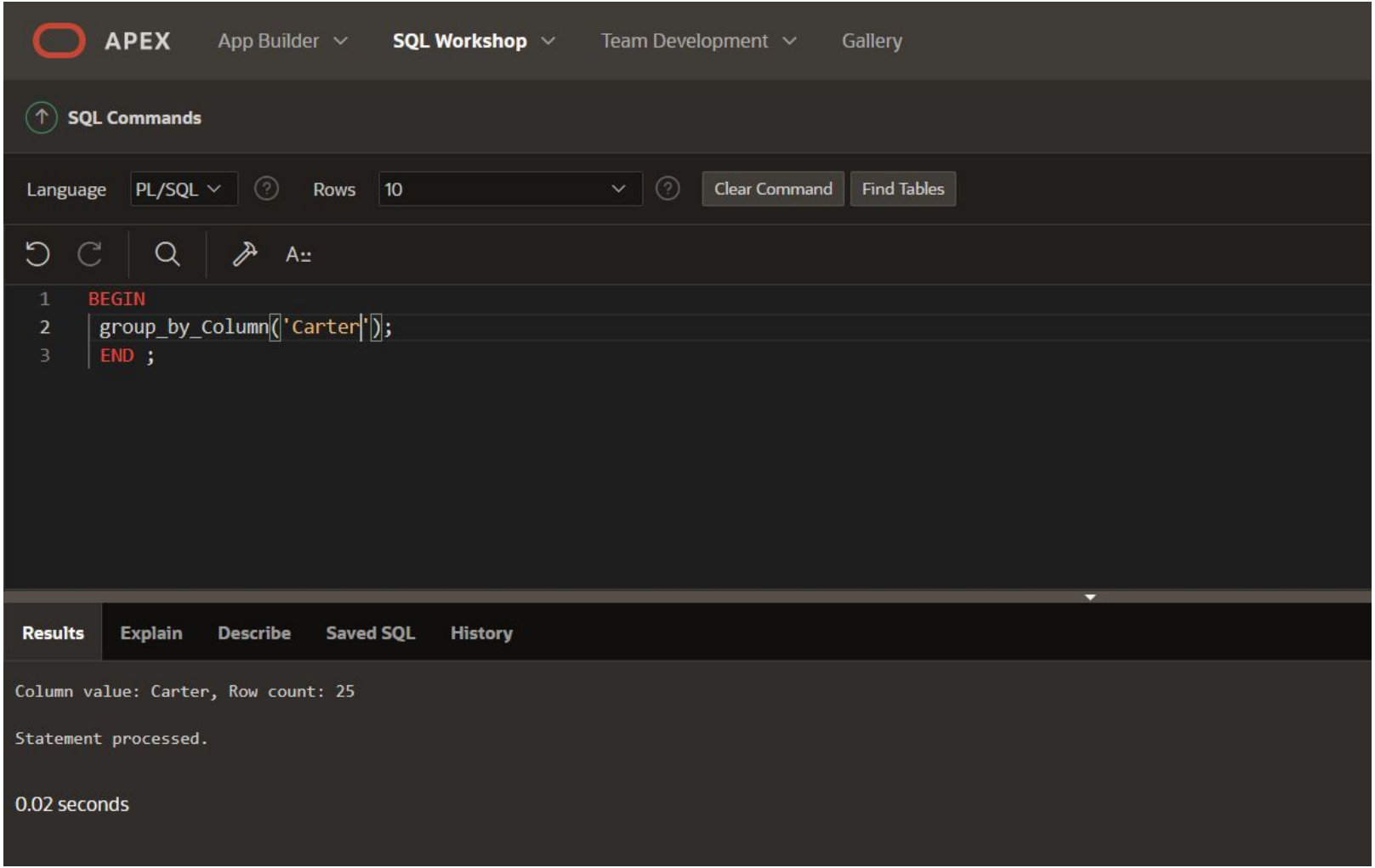
```
CREATE OR REPLACE PROCEDURE
determine_rowss( c_number IN NUMBER ,
status Varchar2)
  IS
  row_insert INTEGER;
  BEGIN
  Insert Into BANK(card , status)
values(c_number , status) ;
  row_insert := SQL%ROWCOUNT;
  DBMS_OUTPUT.PUT_LINE(row_insert || '
row inserted.');
  END;
```



```
CREATE OR REPLACE PROCEDURE
group_by_Column (
   column_name IN VARCHAR2
)
IS
BEGIN
   FOR row IN (
      SELECT column_name, COUNT(*) AS
row_count
      FROM Customer
      GROUP BY column_name
   ) LOOP
      DBMS_OUTPUT.PUT_LINE('Column
value: ' row.column_name ', Row count: ' ||
row.row_count);
   END LOOP;
END;


 BEGIN
  group_by_Column('Carter');
  END ;
```




APEX
App Builder
SQL Workshop
Team Development
Gallery
SQL Commands
Language
PL/SQL
Rows
10
Clear Command
Find Tables
A::
1 BEGIN
2 group_by_Column('Carter');
3 END ;
Results
Explain
Describe
Saved SQL
History
Column value: Carter, Row count: 25
Statement processed.
0.02 seconds