U19CS076 DBMS ASSIGNMENT 8

NAME-KRITHIKHA BALAMURUGAN

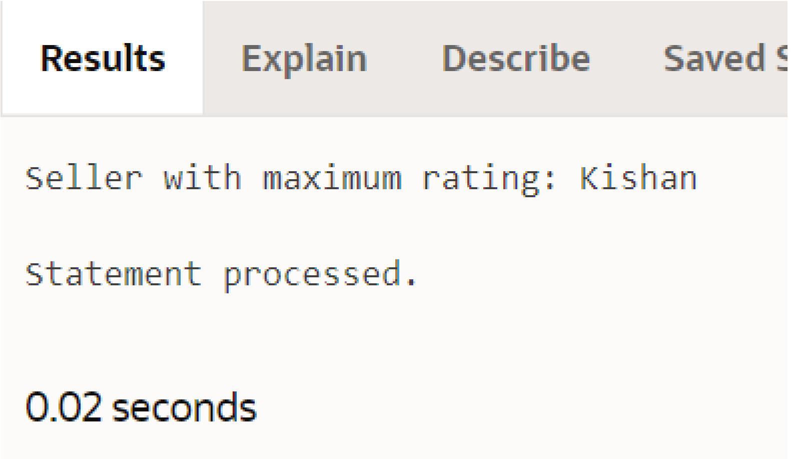
1. Create a Function which returns the seller's name with the highest rating

```
CREATE OR REPLACE FUNCTION seller_max_rating
RETURN SELLER.SELLER_NAME%TYPE
IS
ans SELLER.SELLER_NAME%TYPE;
BEGIN
 SELECT SELLER_NAME INTO ans FROM SELLER WHERE RATING=(SELECT MAX(RATI
NG) FROM SELLER);
RETURN ans;
END;
```

SQL COMMANDS FOR EXECUTION:

```
DECLARE
 ans1 SELLER.SELLER_NAME%TYPE;
BEGIN
 ans1 := seller_max_rating();
 dbms_output.put_line('Seller with maximum rating: ' || ans1);
END;
```

OUTPUT



```
Seller with maximum rating: Kishan

Statement processed.
```

0.02 seconds

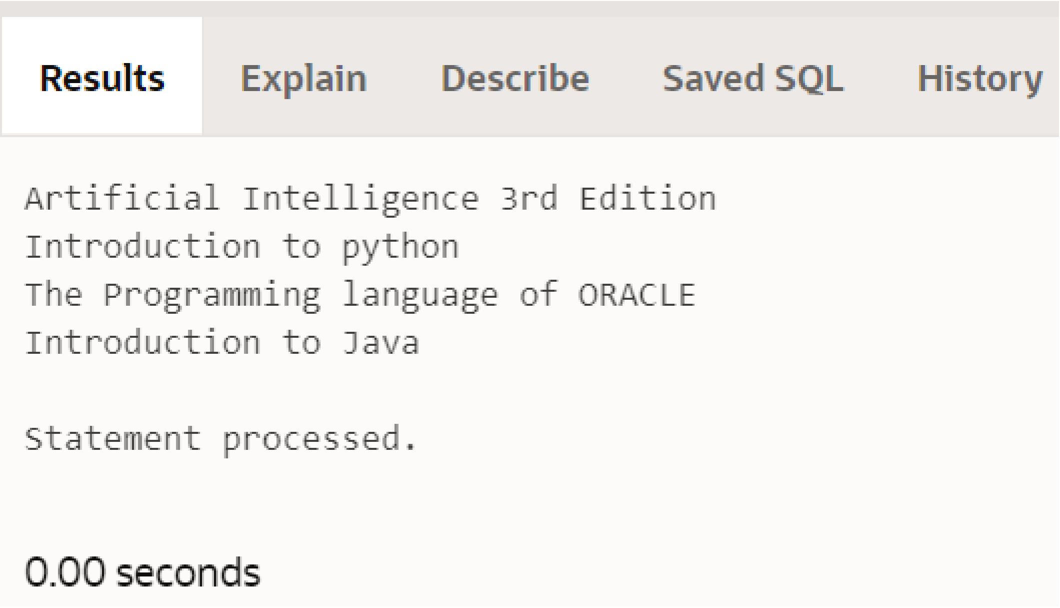
2. Create Stored procedure which takes as an input ‘category’ and outputs all the products of that category.

```
CREATE OR REPLACE PROCEDURE all_in_category (user_cat IN CATEGORY.CATE
GORY%type)
IS
c_prod product.product%type;
CURSOR c_product IS
SELECT PRODUCT FROM PRODUCT WHERE CATEGORY_ID=(SELECT CATEGORY_ID FROM
 CATEGORY WHERE CATEGORY = user_cat);
BEGIN
OPEN c_product;
LOOP
FETCH c_product INTO c_prod;
EXIT WHEN c_product%notfound;
dbms_output.put_line(c_prod);
END LOOP;
CLOSE c_product;
END;
```

SQL COMMANDS FOR EXECUTION:

```
BEGIN
 all_in_category('Books');
END;
```

OUTPUT



```
Artificial Intelligence 3rd Edition
Introduction to python
The Programming language of ORACLE
Introduction to Java

Statement processed.
```

0.00 seconds

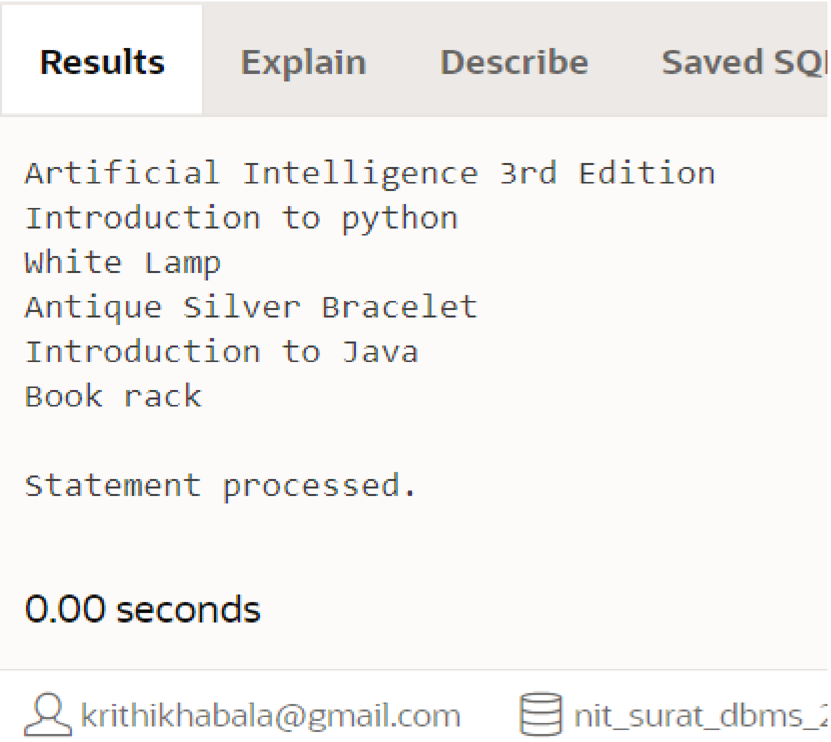
Q3. Create Stored procedure to take a range of prices as input and output all the products in the provided range

```
CREATE OR REPLACE PROCEDURE range_(ll IN PRODUCT.AMOUNT%type, ul IN PR
ODUCT.AMOUNT%type)
IS
 c_prod product.product%type;
 CURSOR c_product IS
 SELECT PRODUCT FROM PRODUCT WHERE AMOUNT BETWEEN ll AND ul;
BEGIN
 OPEN c_product;
 LOOP
 FETCH c_product INTO c_prod;
 EXIT WHEN c_product%notfound;
 dbms_output.put_line(c_prod);
 END LOOP;
 CLOSE c_product;
END;
```

SQL COMMANDS FOR EXECUTION:

```
BEGIN
 range_(500,1000);
END;
```

OUTPUT



```
Artificial Intelligence 3rd Edition
Introduction to python
White Lamp
Antique Silver Bracelet
Introduction to Java
Book rack

Statement processed.
```

0.00 seconds

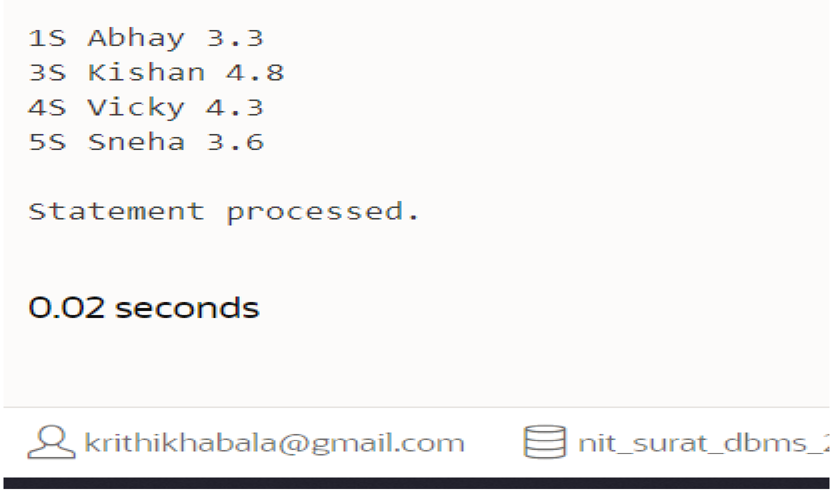
Q4. Create function to display all the seller details with rating more than 3.

```
CREATE OR REPLACE FUNCTION seller_with_greater3rating
RETURN SYS_REFCURSOR
IS
 s_details SYS_REFCURSOR;
BEGIN
 OPEN s_details FOR
 SELECT DISTINCT SELLER_ID, SELLER_NAME, RATING FROM SELLER WHERE RATI
NG>3;
 RETURN s_details;
END;
```

SQL COMMANDS FOR EXECUTION:

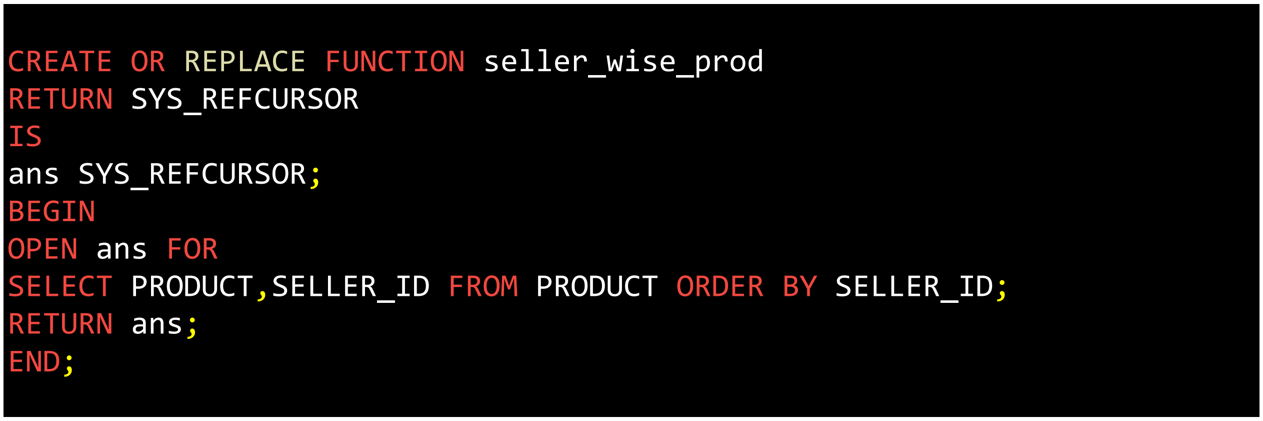
```
DECLARE
 s_details SYS_REFCURSOR; s_id SELLER.SELLER_ID%type;
 s_name SELLER.SELLER_name%type;
 s_rating SELLER.rating%type;
BEGIN
 s_details:=seller_with_greater3rating;
 LOOP
 FETCH s_details INTO s_id, s_name, s_rating;
 EXIT WHEN s_details%NOTFOUND;
 dbms_output.put_line(s_id || ' ' || s_name || ' ' || s_rating);
 END LOOP;
END;
```

Output



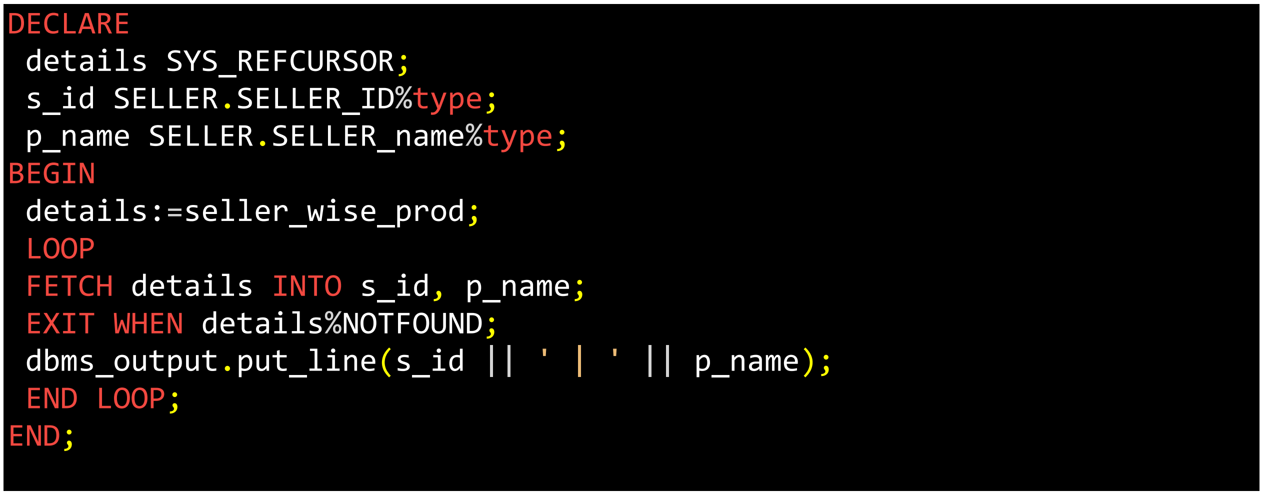
```
1S Abhay 3.3
3S Kishan 4.8
4S Vicky 4.3
5S Sneha 3.6

Statement processed.


0.02 seconds
```

Q5. Create a function to display all the products, seller wise.



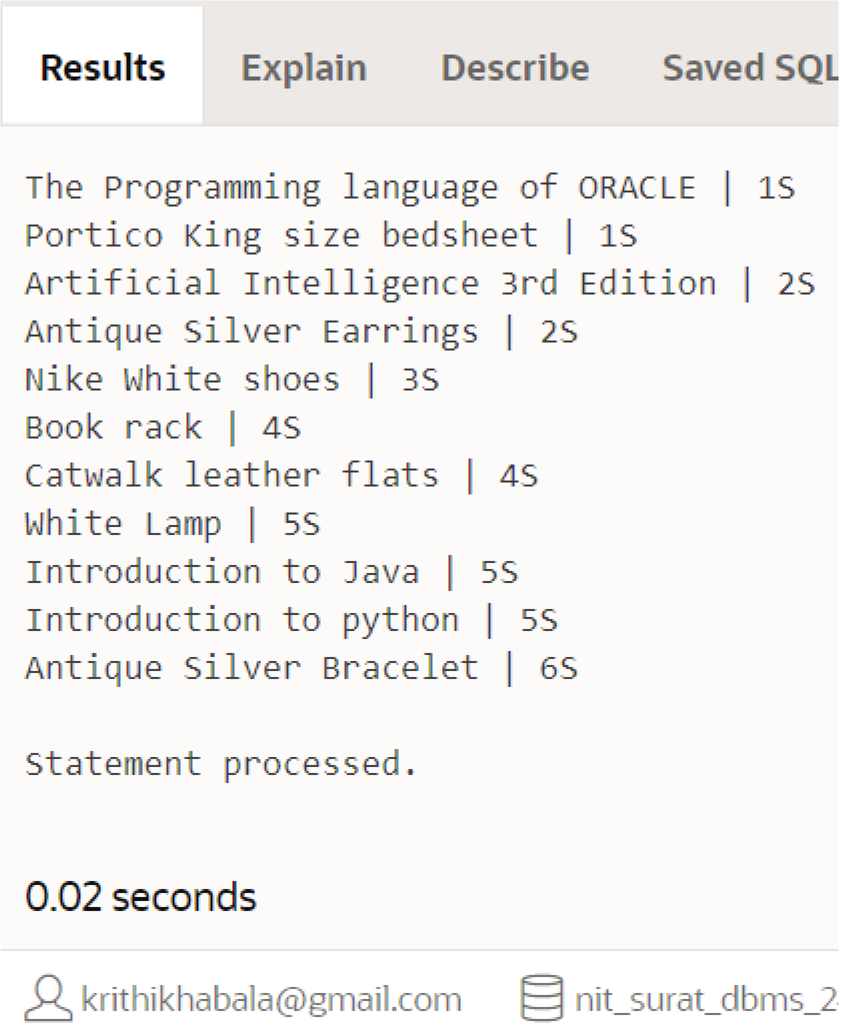
```
CREATE OR REPLACE FUNCTION seller_wise_prod
RETURN SYS_REFCURSOR
IS
ans SYS_REFCURSOR;
BEGIN
OPEN ans FOR
SELECT PRODUCT,SELLER_ID FROM PRODUCT ORDER BY SELLER_ID;
RETURN ans;
END;
```

SQL COMMANDS FOR EXECUTION:



```
DECLARE
 details SYS_REFCURSOR;
 s_id SELLER.SELLER_ID%type;
 p_name SELLER.SELLER_name%type;
BEGIN
 details:=seller_wise_prod;
 LOOP
 FETCH details INTO s_id, p_name;
 EXIT WHEN details%NOTFOUND;
 dbms_output.put_line(s_id || ' | ' || p_name);
 END LOOP;
END;
```

Output



```
The Programming language of ORACLE | 1S
Portico King size bedsheet | 1S
Artificial Intelligence 3rd Edition | 2S
Antique Silver Earrings | 2S
Nike White shoes | 3S
Book rack | 4S
Catwalk leather flats | 4S
White Lamp | 5S
Introduction to Java | 5S
Introduction to python | 5S
Antique Silver Bracelet | 6S

Statement processed.
```

0.02 seconds

krithikhabala@gmail.com nit_surat_dbms_2

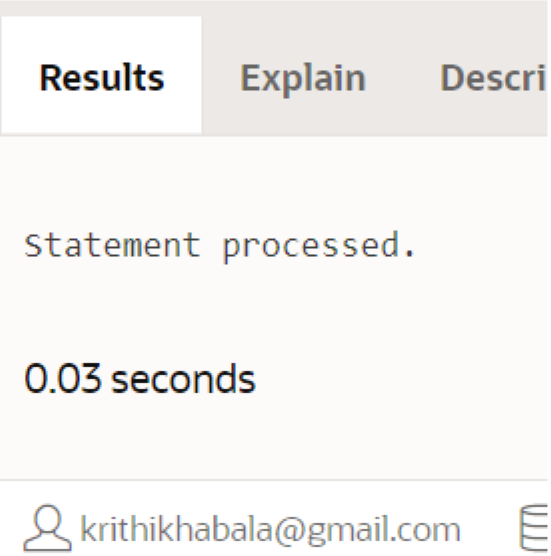
Q6. Create a Stored procedure which checks all the entries in Order_Products table and update seller and product table accordingly.

```
CREATE OR REPLACE PROCEDURE update_product_seller
AS
BEGIN
 UPDATE product p SET p.rating = (SELECT AVG(prod_rating) FROM order_p
roduct GROUP BY product_id HAVING product_id = p.product_id);
 UPDATE seller s SET s.rating = (SELECT AVG(prod_rating) FROM order_pr
oduct GROUP BY seller_id HAVING seller_id = s.seller_id);
END;
```

SQL COMMANDS FOR EXECUTION:

```
BEGIN
update_product_seller;
END;
```

Output



```
Statement processed.
```

0.03 seconds

krithikhabala@gmail.com

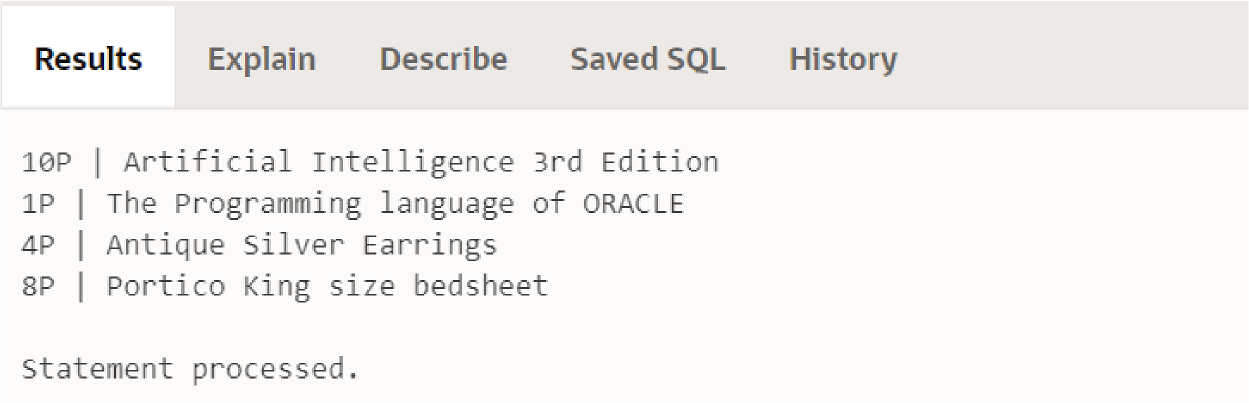
Q7. Create Stored procedure which takes as input different filters such as price range, category, product rating, seller rating, out of stock and displays the list of products with all the details after applying filters.

```
create or replace procedure filter_input(filter_type in number, val in varchar2)
is
prod_details sys_refcursor;
prod_id product.product_id%type;
prod_name product.product%type;
begin
case filter_type
when 1 then open prod_details for select PRODUCT_ID, PRODUCT  from product where amount < to_number(val);
when 2 then open prod_details for select p.PRODUCT_ID, p.PRODUCT from product p, category cat where p.category_id = cat.category_id and cat.category = val;
when 3 then open prod_details for select PRODUCT_ID, PRODUCT from product where rating > to_number(val);
when 4 then open prod_details for select p.PRODUCT_ID, p.PRODUCT  from product p,seller s where s.seller_id = p.seller_id and s.rating > to_number(val);
when 5 then open prod_details for select PRODUCT_ID, PRODUCT  from product where Quantity_Rem is not NULL;
end case;
loop
fetch prod_details into prod_id, prod_name;
exit when prod_details%notfound;
dbms_output.put_line( prod_id || ' | ' || prod_name );

end loop;
end;
```

SQL COMMANDS FOR EXECUTION:

```
begin
filter_input(4,'2.6');
end;
```

Output



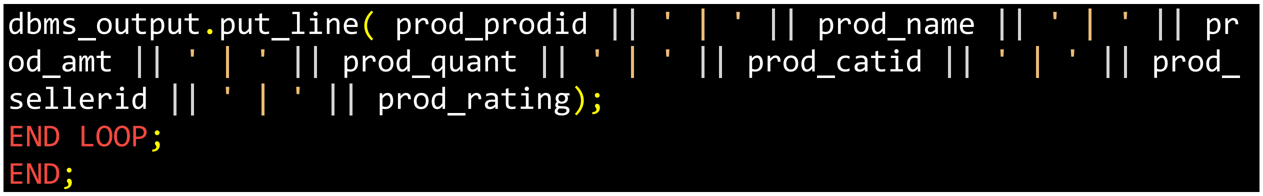
```
10P | Artificial Intelligence 3rd Edition
1P | The Programming language of ORACLE
4P | Antique Silver Earrings
8P | Portico King size bedsheet

Statement processed.
```

Q8. Create a function which takes as input sorting criteria like popularity or lowest price or highest price and display the product list accordingly.

```
CREATE OR REPLACE FUNCTION order_by_criteria(opt IN number)
RETURN SYS_REFCURSOR
IS
prod_details SYS_REFCURSOR;
BEGIN
CASE opt
WHEN 1 THEN OPEN prod_details FOR SELECT PRODUCT_ID, PRODUCT, AMOUNT,
QUANTITY_REM, CATEGORY_ID, SELLER_ID, RATING FROM PRODUCT ORDER BY AMOUNT;
WHEN 2 THEN OPEN prod_details FOR SELECT PRODUCT_ID, PRODUCT, AMOUNT,
QUANTITY_REM, CATEGORY_ID, SELLER_ID, RATING FROM PRODUCT ORDER BY AMOUNT DESC;
WHEN 3 THEN OPEN prod_details FOR SELECT PRODUCT_ID, PRODUCT, AMOUNT,
QUANTITY_REM, CATEGORY_ID, SELLER_ID, RATING FROM PRODUCT ORDER BY RATING DESC;
END CASE;
RETURN prod_details;
END;
```

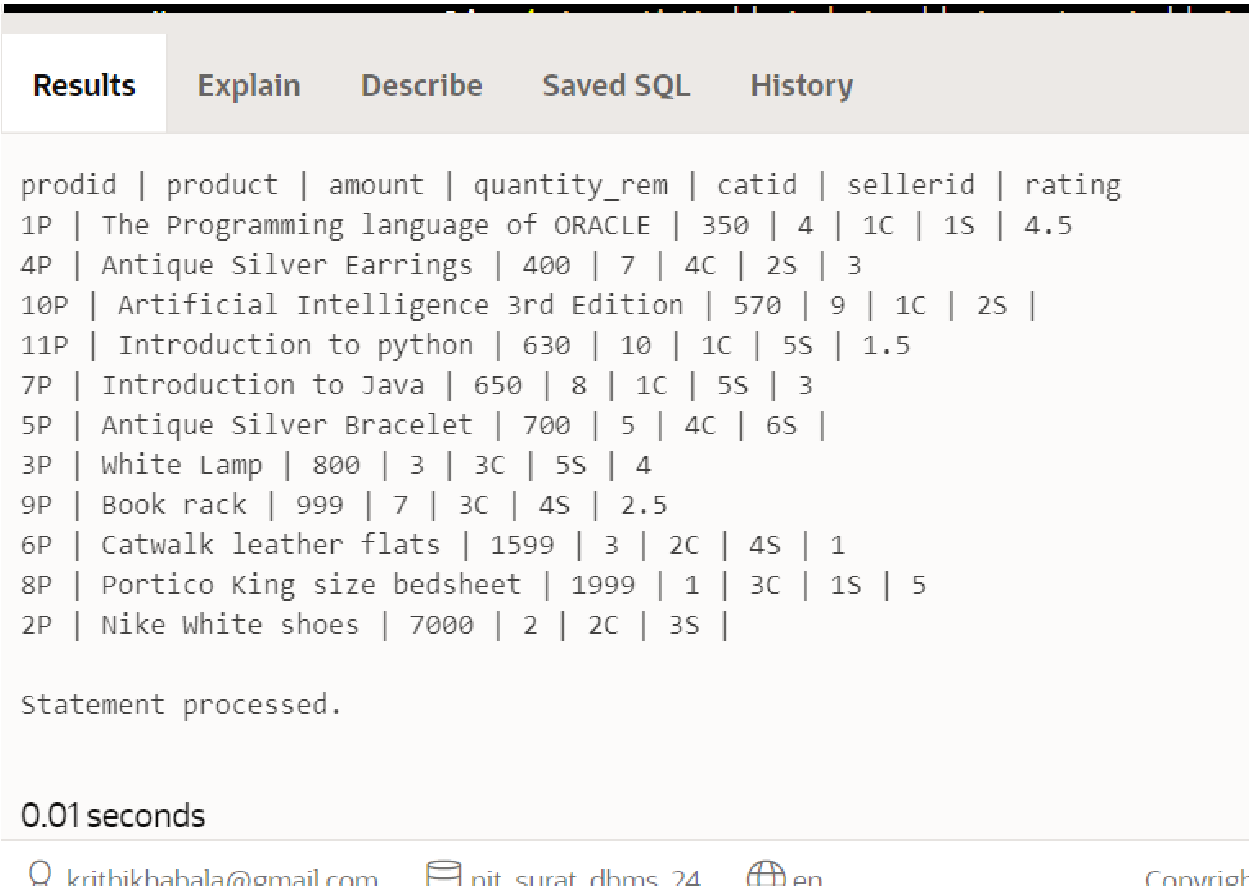
SQL COMMANDS FOR EXECUTION:

```
DECLARE
prod_details SYS_REFCURSOR;
prod_prodid PRODUCT.PRODUCT_ID%type;
prod_name PRODUCT.PRODUCT%type;
prod_amt PRODUCT.AMOUNT%type;
prod_quant PRODUCT.QUANTITY_REM%type;
prod_catid PRODUCT.CATEGORY_ID%type;
prod_sellerid PRODUCT.SELLER_ID%type;
prod_rating PRODUCT.RATING%type;
BEGIN
dbms_output.put_line( 'prodid' || ' | '  || 'product' || ' | '  || 'amount' || ' | '  || 'quantity_rem' || ' | '  || 'catid' || ' | '  || 'sellerid' || ' | '  || 'rating');
--
criteria for sorting: 1 for amount ascending 2 for amount descending,
3 for rating wise
prod_details:=order_by_criteria(1);
LOOP
FETCH prod_details INTO prod_prodid, prod_name, prod_amt, prod_quant,prod_catid, prod_sellerid, prod_rating;
EXIT WHEN prod_details%NOTFOUND;
```



```
dbms_output.put_line( prod_prodid || ' | ' || prod_name || ' | ' || pr
od_amt || ' | ' || prod_quant || ' | ' || prod_catid || ' | ' || prod_
sellerid || ' | ' || prod_rating);
END LOOP;
END;
```

Output



```
prodid | product | amount | quantity_rem | catid | sellerid | rating
1P | The Programming language of ORACLE | 350 | 4 | 1C | 1S | 4.5
4P | Antique Silver Earrings | 400 | 7 | 4C | 2S | 3
10P | Artificial Intelligence 3rd Edition | 570 | 9 | 1C | 2S |
11P | Introduction to python | 630 | 10 | 1C | 5S | 1.5
7P | Introduction to Java | 650 | 8 | 1C | 5S | 3
5P | Antique Silver Bracelet | 700 | 5 | 4C | 6S |
3P | White Lamp | 800 | 3 | 3C | 5S | 4
9P | Book rack | 999 | 7 | 3C | 4S | 2.5
6P | Catwalk leather flats | 1599 | 3 | 2C | 4S | 1
8P | Portico King size bedsheet | 1999 | 1 | 3C | 1S | 5
2P | Nike White shoes | 7000 | 2 | 2C | 3S |

Statement processed.
```

0.01 seconds