***FINAL EXAM***

***Student Name: Basavraj Nagappa Jaliminche***

***Student Id: 8800149***

***PROGRAM: Reporting Systems and Database Development (1517)***

***Course Name: Relational Databases PROG 8590***

**Q 1**

1. **Write an anonymous block using PL/SQL (program) to determine the order status of a customer. The program will search using a customer's ID and determine how many orders the customer has made. - If the customer has 3 or more orders, create a message telling the user what this customer is a ‘Preferred customer’. - If the customer has less than 3 orders, inform the user that this is a ‘Normal customer’. - If the customer has no orders yet, inform the user that the status is ‘No orders’.**
2. **Run the program 3 times hardcoding in a search for customer id 145, 111 and 155. You only need to submit one version of your code, but include 3 outputs.**

**Answer:**

--Anonymous block

DECLARE -- declaration

v\_cid number:=145; -- declarinng the value with custmarid=145

v\_ordStatus number;

BEGIN --execution section

SELECT COUNT(\*) INTO v\_ordStatus

FROM OEHR\_ORDERS

WHERE CUSTOMER\_ID = v\_cid;

IF(v\_ordStatus >= 3)

THEN --IF condition

DBMS\_OUTPUT.put\_line('Preferred customer');

ELSIF(v\_ordStatus BETWEEN 1 and 3) THEN -- condition

DBMS\_OUTPUT.put\_line('Normal customer');

ELSE --if all other condition fails

DBMS\_OUTPUT.put\_line('No orders');

END IF; --end if statement

EXCEPTION --exception handling section section

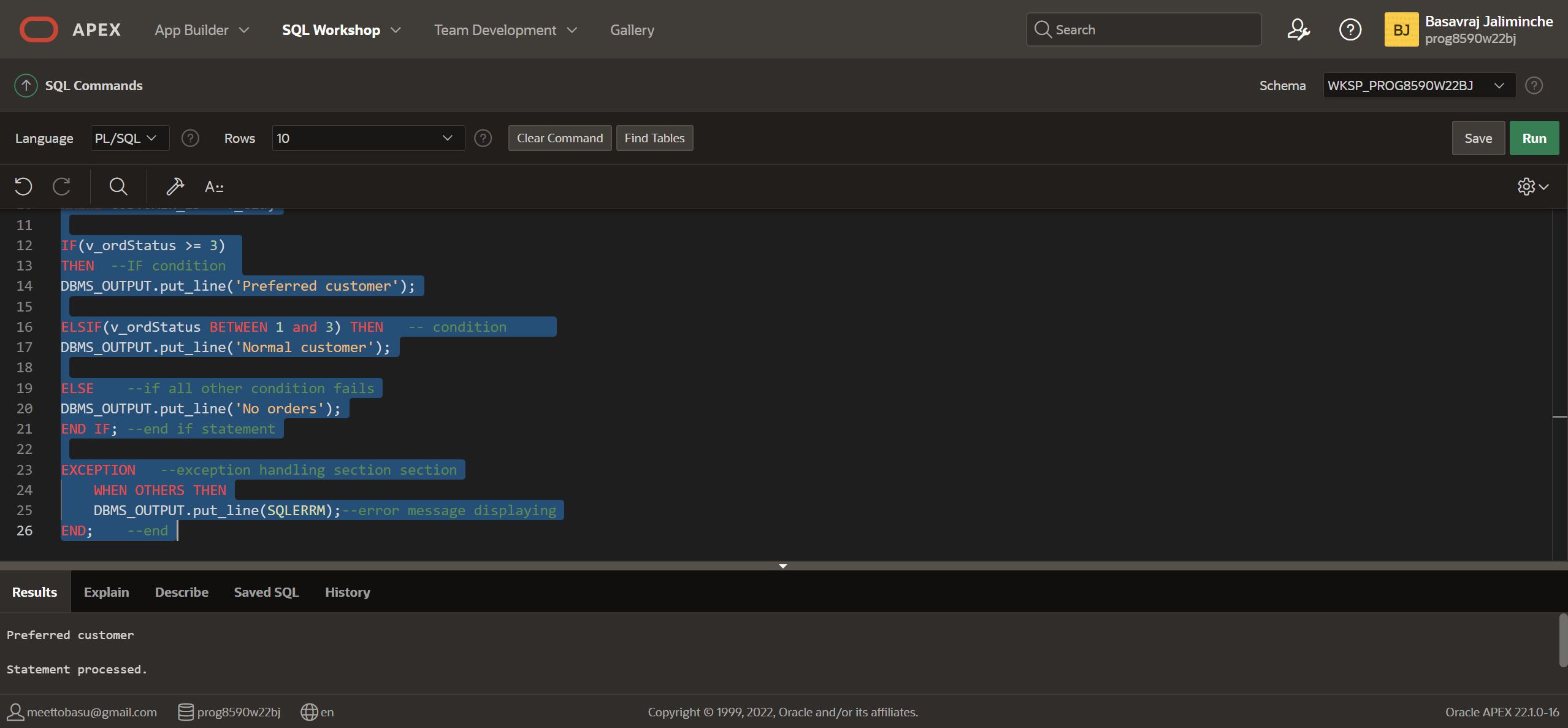
WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SQLERRM);--error message displaying

END; --end

Screenshots:

When Custamar id=145



**When Custamar id=111**

--Anonymous block

DECLARE -- declaration

v\_cid number:=111; -- declarinng the value with custmarid=145

v\_ordStatus number;

BEGIN --execution section

SELECT COUNT(\*) INTO v\_ordStatus

FROM OEHR\_ORDERS

WHERE CUSTOMER\_ID = v\_cid;

IF(v\_ordStatus >= 3)

THEN --IF condition

DBMS\_OUTPUT.put\_line('Preferred customer');

ELSIF(v\_ordStatus BETWEEN 1 and 3) THEN -- condition

DBMS\_OUTPUT.put\_line('Normal customer');

ELSE --if all other condition fails

DBMS\_OUTPUT.put\_line('No orders');

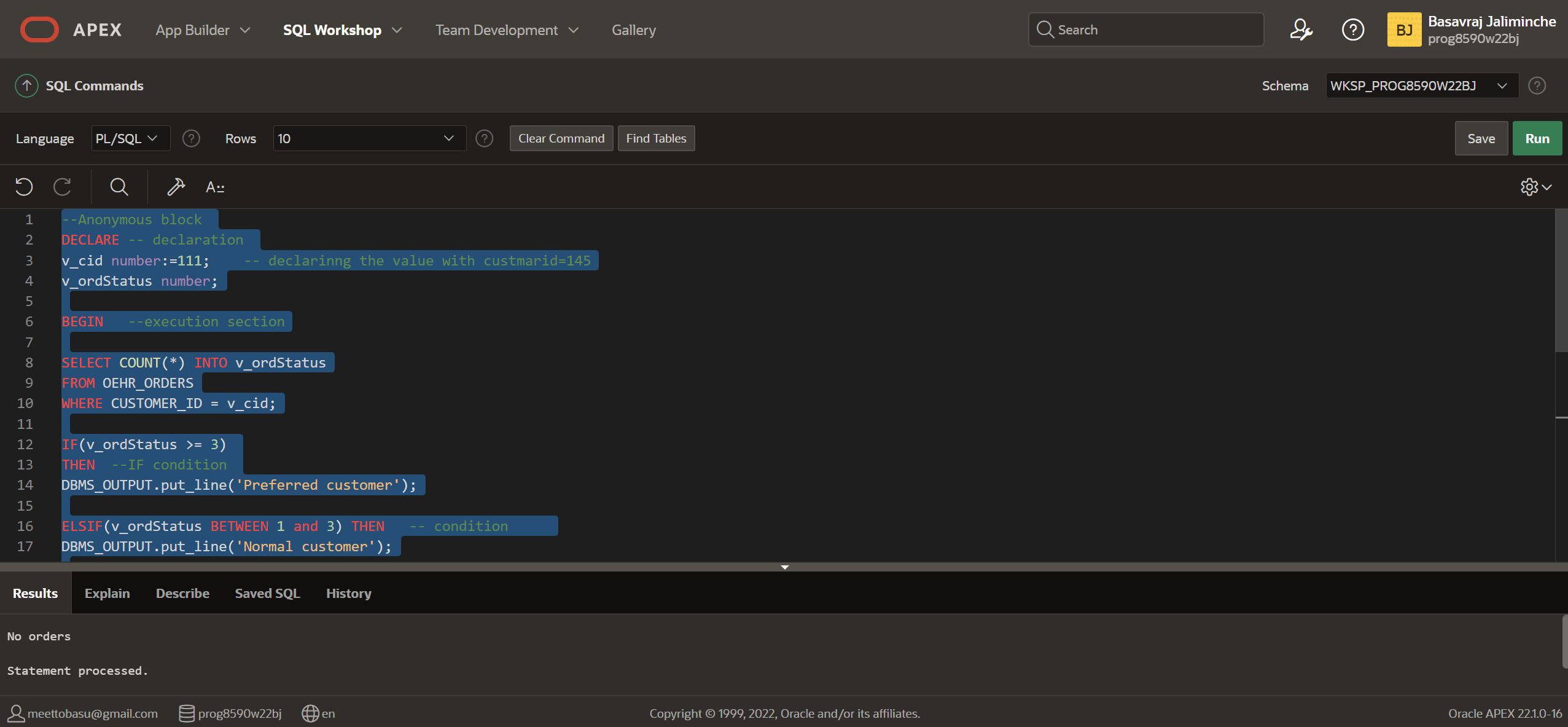
END IF; --end if statement

EXCEPTION --exception handling section section

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SQLERRM);--error message displaying

END; --end

****

**When custamar id=155**

--Anonymous block

DECLARE -- declaration

v\_cid number:=155; -- declarinng the value with custmarid=145

v\_ordStatus number;

BEGIN --execution section

SELECT COUNT(\*) INTO v\_ordStatus

FROM OEHR\_ORDERS

WHERE CUSTOMER\_ID = v\_cid;

IF(v\_ordStatus >= 3)

THEN --IF condition

DBMS\_OUTPUT.put\_line('Preferred customer');

ELSIF(v\_ordStatus BETWEEN 1 and 3) THEN -- condition

DBMS\_OUTPUT.put\_line('Normal customer');

ELSE --if all other condition fails

DBMS\_OUTPUT.put\_line('No orders');

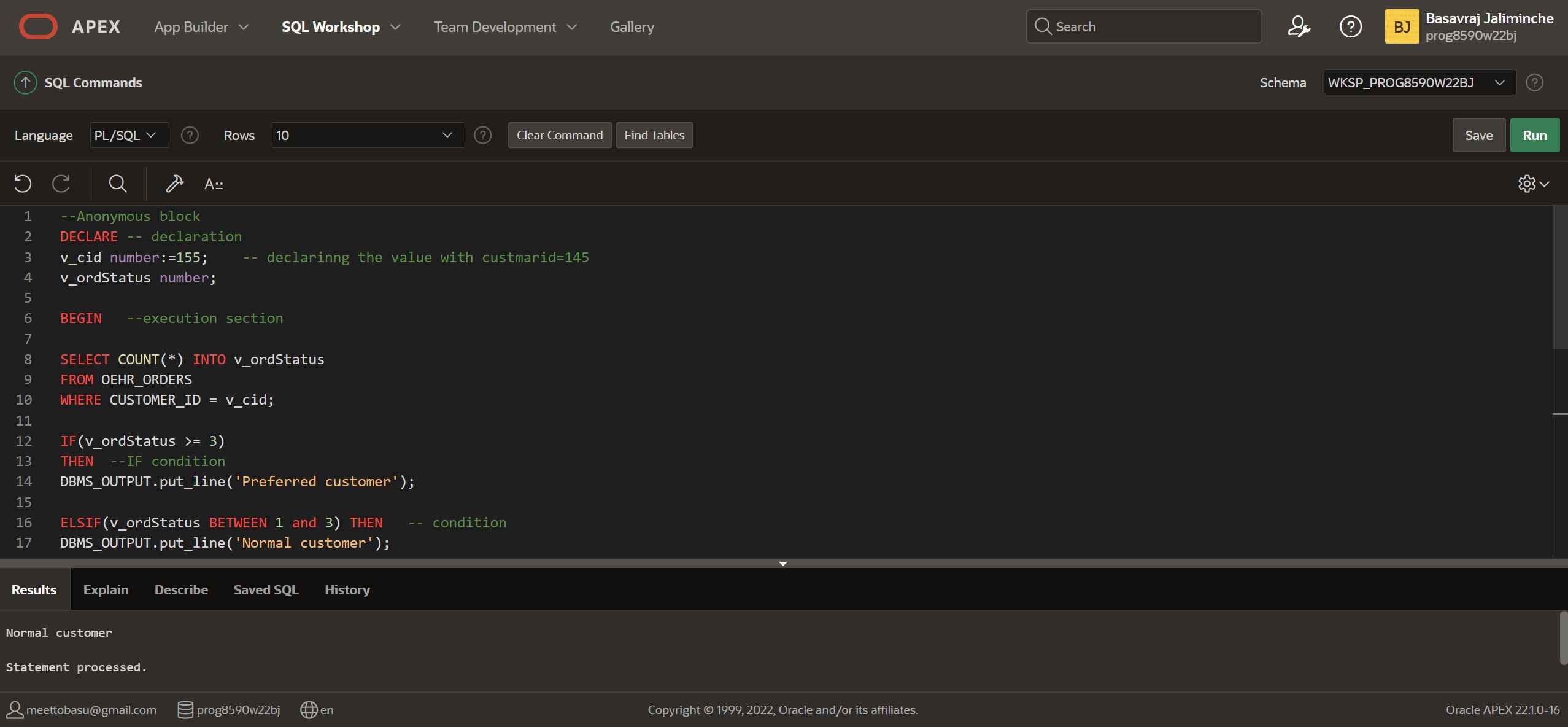
END IF; --end if statement

EXCEPTION --exception handling section section

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SQLERRM);--error message displaying

END; --end



**Question 2: [7 Marks] 1- Modify the written program in Question1 to determine the status of a customer by defining a function that takes into the customer Id as Input parameter and return the customer status as Output parameter. 2- Run the program hardcoding in a search for the customer id 155.**

**Answer:**

**-- Function Body**

**CREATE OR REPLACE FUNCTION customerStatus(cid NUMBER)**

**RETURN NUMBER**

**IS --function return datatype**

**--variable**

**v\_num NUMBER;**

**BEGIN**

**-- Query for checking count of employee in dept**

**SELECT COUNT(\*) INTO v\_num**

**FROM OEHR\_ORDERS**

**WHERE CUSTOMER\_ID = cid;**

**RETURN v\_num;**

**EXCEPTION**

**WHEN NO\_DATA\_FOUND THEN**

**DBMS\_OUTPUT.PUT\_LINE('Customer ID not exist');**

**RETURN v\_num;**

**END;**

--Anonymous Block

DECLARE

v\_orderStatus number;

cust\_id number := 155;

BEGIN

v\_orderStatus := customerStatus(cust\_id); -- calling function and assigning return value to variable

--check condition for status

IF(v\_orderStatus >= 3) THEN --IF condition for order status

DBMS\_OUTPUT.put\_line('Preferred customer');

ELSIF(v\_orderStatus BETWEEN 1 AND 2) THEN --ELSE-IF condition for order status

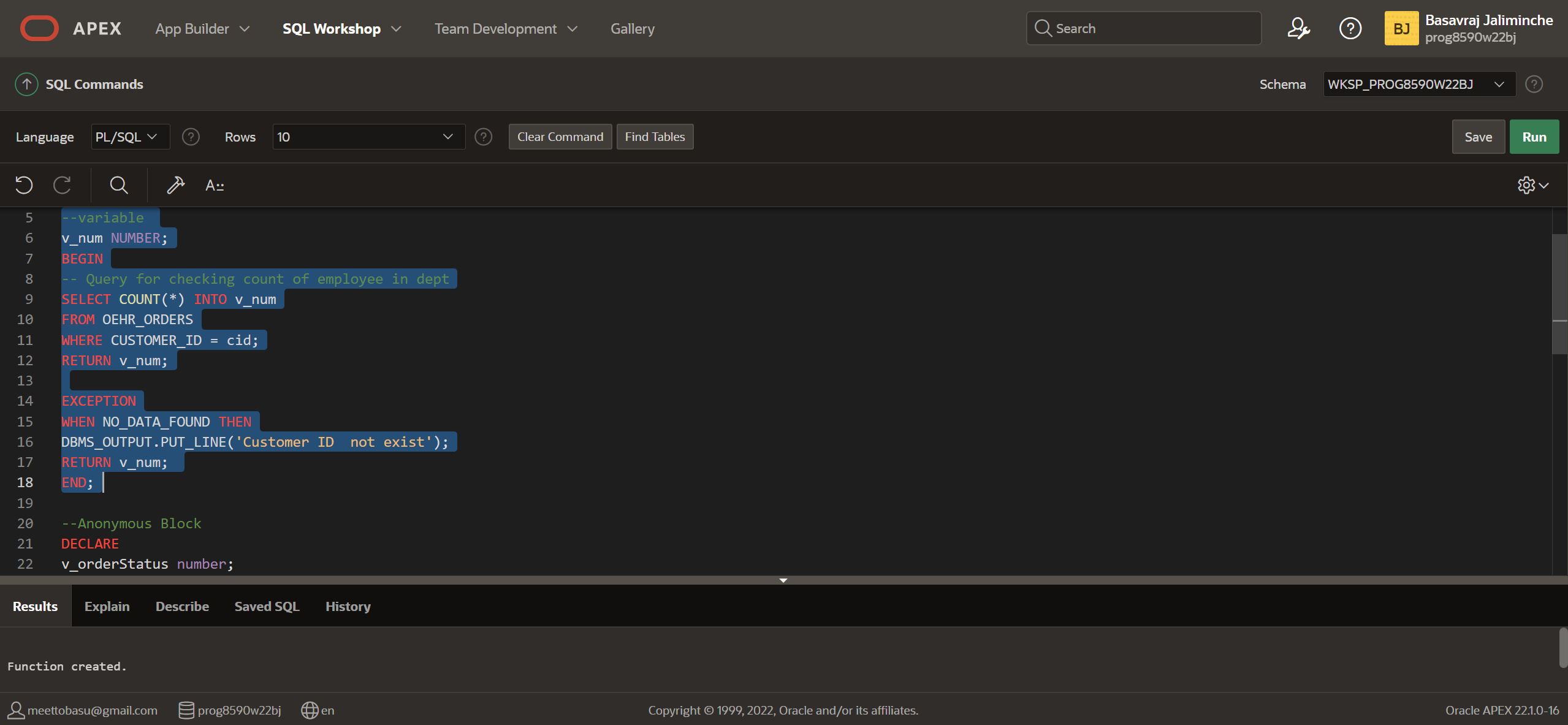
DBMS\_OUTPUT.put\_line('Normal customer');

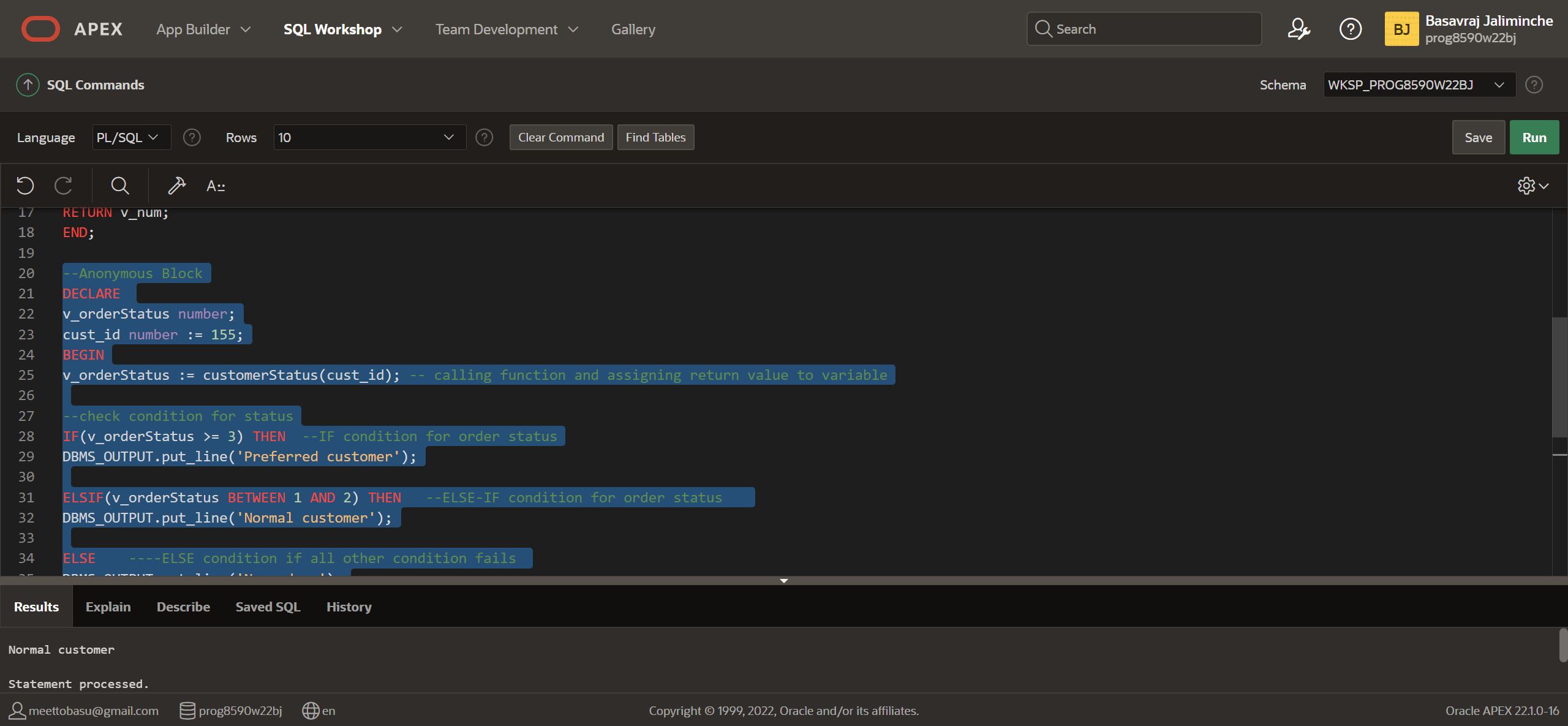
ELSE ----ELSE condition if all other condition fails

DBMS\_OUTPUT.put\_line('No orders');

END IF; --end if statement

END; -- end Anonymous Block





**Q.3 Write an anonymous block using PL/SQL (program) to output the number of employees in each department group by city from OEHR\_LOCATIONS and save it in a Varray of 30 elements. To do this, please follow these steps: 1- Define a record that holds the department name, the number of employees and the city. [5 pts] 2- Define the Varray type and the varray variable [3 pts] 3- Define the cursor to fetch the data [5 pts] 4- Load data into the varray [5 pts] and display the result [2 pts] Expected result:**

London: Department [department\_name] has 99 employee(s)

Seattle: Department [department\_name] has 99 employee(s)

Department [department\_name] has 99 employee(s)

**Answer:**

**DECLARE**

**--record that holds the department name ,number of employess...**

**TYPE r\_department\_detail IS RECORD (**

**dept\_name OEHR\_DEPARTMENTS.DEPARTMENT\_NAME%TYPE,**

**dept\_emp number,**

**dept\_city OEHR\_LOCATIONS.CITY%TYPE**

**);**

**-- VARRAY TYPE**

**TYPE t\_dept\_varray IS VARRAY(30) OF r\_department\_detail;**

**-- VARRAY VARIABLE**

**v\_dept t\_department\_varray:=t\_department\_varray();-- empty varray**

**-- CURSOR TO FETCH DATA FROM OEHR\_EMPLOYEES**

**CURSOR c\_department IS**

**SELECT COUNT(\*) EMPLOYEE\_COUNT, DEPARTMENT\_NAME, CITY**

**FROM**

**OEHR\_EMPLOYEES OE**

**LEFT JOIN**

**OEHR\_DEPARTMENTS OD**

**ON OE.DEPARTMENT\_ID = OD.DEPARTMENT\_ID**

**LEFT JOIN**

**OEHR\_LOCATIONS OL**

**ON OD.LOCATION\_ID = OL.LOCATION\_ID**

**GROUP BY DEPARTMENT\_NAME, CITY;**

**BEGIN**

**-- Loading data**

**FOR my\_data IN c\_department LOOP**

**v\_department.EXTEND;**

**v\_department(v\_department.LAST).dept\_emp:=my\_data.EMPLOYEE\_COUNT;**

**v\_department(v\_department.LAST).dept\_name:=my\_data.DEPARTMENT\_NAME;**

**v\_department(v\_department.LAST).dept\_city:=my\_data.CITY;**

**END LOOP;**

**DBMS\_OUTPUT.put\_line(v\_department.COUNT);**

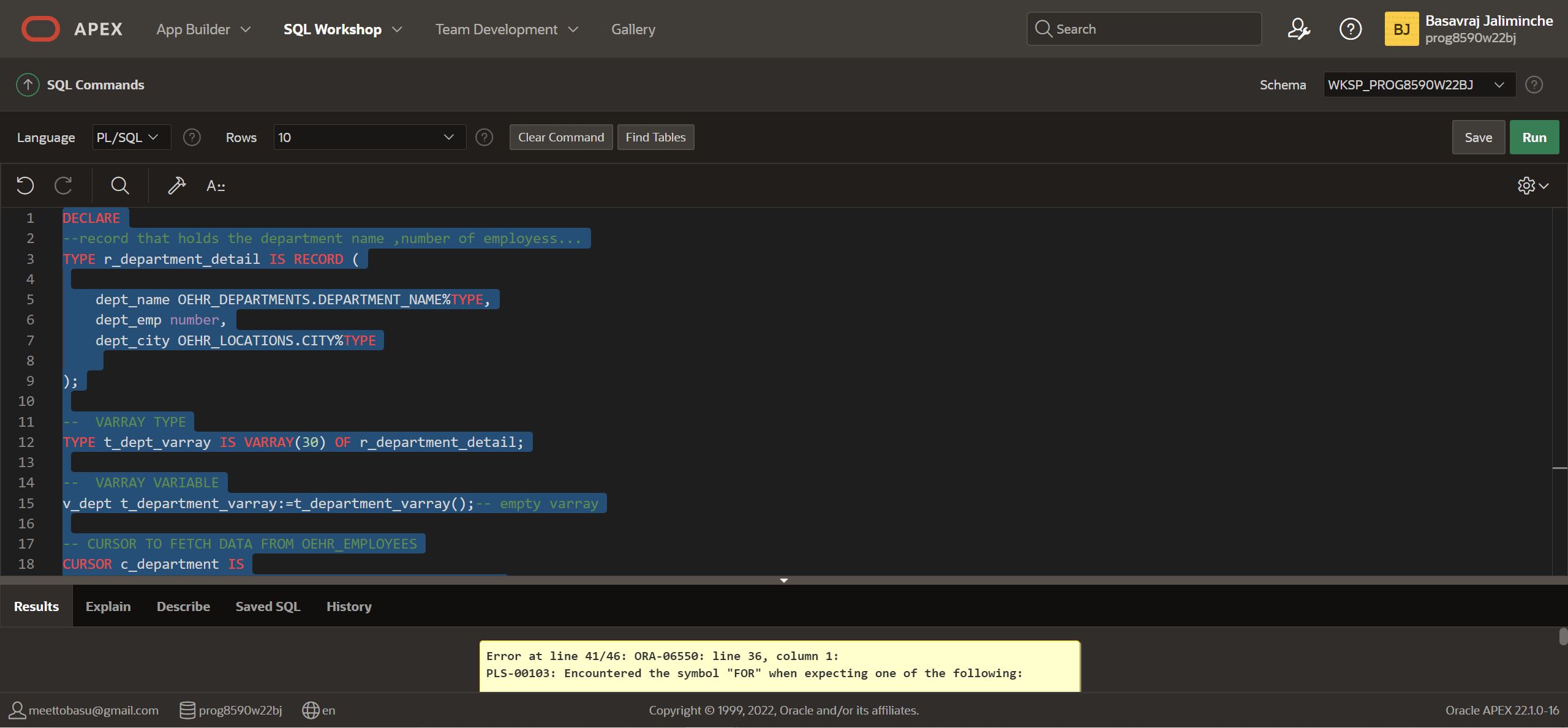
**FOR my\_index IN v\_department.FIRST..v\_department.L**

**FOR my\_index IN v\_department.FIRST..v\_department.LAST LOOP**

**DBMS\_OUTPUT.put\_line(v\_department(my\_index).dept\_city||' : '|| v\_department(my\_index).dept\_name||' has '|| v\_department(my\_index).dept\_emp||' employee(s)');**

**END LOOP;**

**END;**

****

**Q-4**

1. **Create a table named INVENTORY which has five columns: INVENTORY\_ID NUMBER(6) [Primary key], ITEM\_NAME VARCHAR2(255), QUANTITY NUMBER, PRICE NUMBER, ITEM\_SIZE VARCHAR(255), INVENTORY\_VALUE NUMBER. [2 pts]**
2. **Create a trigger that only allows an insert to complete if the SIZE value is one of the following (and any case variations): S, M, L, XL, small, medium, large, extra-large. If an invalid input is attempted to be inserted, reject the insert and raise an application error that states 'Please enter for SIZE only: S, M, L, XL, small, medium, large or extra-large'. [8 pts]**
3. **Insert into the INVENTORY table (1, 'Web Shooter, 2, 19.00, 's', 38.00) Insert into the INVENTORY table (2, 'Fantasticar', 4, 3000.00, 'very big', 12000.00) Insert into the INVENTORY table (3, 'Mjolnir', 1, 100.00, 'medium', 100.00) Run a SELECT \* statement on the INVENTORY table. [3 pts]**
4. **DELETE all rows created in Question 4-3 from the INVENTORY table. [2 pts]**
5. **Create an insert command to insert the following into the table INVENTORY: (42, 'LMD', 2, 199.99, 'M'). Run a SELECT \* statement on the INVENTORY table. You will notice the VALUE column for the item LMD is missing. [2 pts]**
6. **Create a trigger which fires on inserts or updates and fills in the INVENTORY\_VALUE column as Quantity multiplied by Price. [6 pts]**

**Answer:**