[**PROG8590-24F- Sec1   
  
Relational**](https://conestoga.desire2learn.com/d2l/home/1255613) **Databases**

**NAME: VETRI VELAN ANANDAN**

**STUDENT ID: 8979936**

**Semester Exam**

**1.**

*Question 1: [7 Marks]*

*1- Write an anonymous block using PL/SQL to determine the order status of a customer. The*

*program will search using customer ID and determine how many orders the customer has*

*made.*

*- If the customer has 3 or more orders, display the message ‘Preferred customer’.*

*- If the customer has less than 3 orders, display ‘Normal customer’.*

*- If the customer has no orders yet, display ‘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.*

**CODE:**

**DECLARE**

**v\_CUSTOMER\_ID NUMBER := 145;**

**v\_ORDER\_ID NUMBER;**

**BEGIN**

**SELECT COUNT(\*)**

**INTO v\_ORDER\_ID**

**FROM OEHR\_ORDERS**

**WHERE CUSTOMER\_ID = v\_CUSTOMER\_ID;**

**IF v\_ORDER\_ID >= 3 THEN**

**DBMS\_OUTPUT.PUT\_LINE('Preferred customer');**

**ELSIF v\_ORDER\_ID > 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE('Normal customer');**

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE('No orders');**

**END IF;**

**END;**

**Output:**

**Id:145**

****

**Id:111**

**A screenshot of a computer

Description automatically generated**

**Id: 155**

**A white rectangular object with a black border

Description automatically generated**

*2.*

*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.*

**Code:**

**CREATE OR REPLACE FUNCTION get\_ORDER\_STATUS(p\_CUSTOMER\_ID IN NUMBER)**

**RETURN VARCHAR2**

**IS**

**v\_ORDER\_ID NUMBER;**

**v\_status VARCHAR2(20);**

**BEGIN**

**SELECT COUNT(\*)**

**INTO v\_ORDER\_ID**

**FROM OEHR\_ORDERS**

**WHERE CUSTOMER\_ID = p\_CUSTOMER\_ID;**

**IF v\_ORDER\_ID >= 3 THEN**

**v\_status := 'Preferred customer';**

**ELSIF v\_ORDER\_ID > 0 THEN**

**v\_status := 'Normal customer';**

**ELSE**

**v\_status := 'No orders';**

**END IF;**

**RETURN v\_status;**

**END;  
  
output:  
A close-up of a computer screen

Description automatically generated**

**Code:**

**DECLARE**

**v\_ORDER\_STATUS VARCHAR2(20);**

**BEGIN**

**v\_ORDER\_STATUS := get\_ORDER\_STATUS(155);**

**DBMS\_OUTPUT.PUT\_LINE('Customer Status: ' || v\_ORDER\_STATUS);**

**END;**

**Output:**

**A white rectangular object with a black border

Description automatically generated**

*Question 3: [10 Marks]*

*1- Create a procedure with no parameters unordered\_products that displays the details of*

*products that have not been ordered: product\_id, product\_name and list\_price. Use*

*Explicit cursor. [9 pts]*

*2- Run the procedure [1 pt]*

**CODE:**

**CREATE OR REPLACE PROCEDURE unordered\_products**

**IS**

**CURSOR c\_unordered\_products IS**

**SELECT PRODUCT\_ID, PRODUCT\_NAME, LIST\_PRICE**

**FROM OEHR\_PRODUCT\_INFORMATION**

**WHERE PRODUCT\_ID NOT IN (SELECT PRODUCT\_ID FROM OEHR\_ORDER\_ITEMS);**

**v\_product c\_unordered\_products%ROWTYPE;**

**BEGIN**

**OPEN c\_unordered\_products;**

**LOOP**

**FETCH c\_unordered\_products INTO v\_product;**

**EXIT WHEN c\_unordered\_products%NOTFOUND;**

**DBMS\_OUTPUT.PUT\_LINE('Product ID: ' || v\_product.product\_id ||**

**', Name: ' || v\_product.product\_name ||**

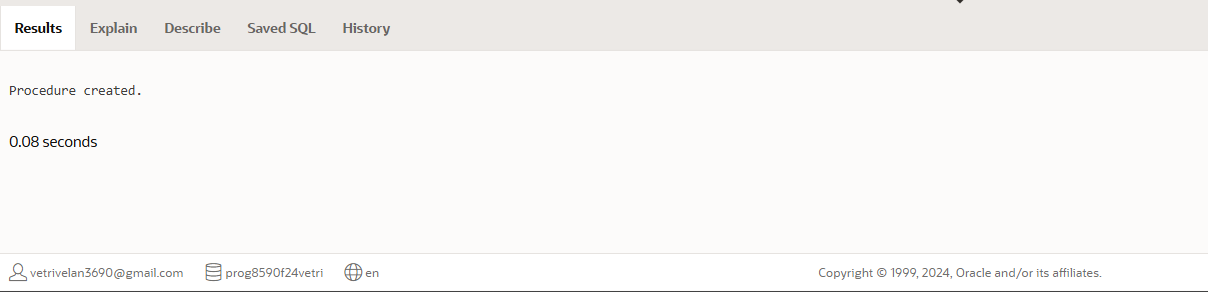
**', Price: ' || v\_product.list\_price);**

**END LOOP;**

**CLOSE c\_unordered\_products;**

**END;**

**OUTPUT:**

****

**CODE:**

**BEGIN**

**unordered\_products;**

**END;**

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**Question 4: [20 Marks]**

*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 and display the result [7 pts]*

*Expected result:*

*London:*

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

*Seattle:*

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

*Department [department\_name] has 99 emp*

**Code:**

**DECLARE**

**TYPE dept\_record IS RECORD (**

**DEPARTMENT\_NAME VARCHAR2(50),**

**EMPLOYEE\_ID NUMBER,**

**CITY VARCHAR2(50)**

**);**

**TYPE dept\_varray IS VARRAY(30) OF dept\_record;**

**v\_dept\_varray dept\_varray := dept\_varray();**

**CURSOR c\_dept\_data IS**

**SELECT d.DEPARTMENT\_NAME, COUNT(e.EMPLOYEE\_ID) AS EMPLOYEE\_ID, l.CITY**

**FROM OEHR\_DEPARTMENTS d**

**JOIN OEHR\_EMPLOYEES e ON d.DEPARTMENT\_ID = e.DEPARTMENT\_ID**

**JOIN OEHR\_LOCATIONS l ON d.LOCATION\_ID = l.LOCATION\_ID**

**GROUP BY d.DEPARTMENT\_NAME, l.CITY;**

**v\_index NUMBER := 0;**

**v\_row c\_dept\_data%ROWTYPE;**

**BEGIN**

**OPEN c\_dept\_data;**

**LOOP**

**FETCH c\_dept\_data INTO v\_row;**

**EXIT WHEN c\_dept\_data%NOTFOUND;**

**v\_index := v\_index + 1;**

**v\_dept\_varray.EXTEND;**

**v\_dept\_varray(v\_index).DEPARTMENT\_NAME := v\_row.DEPARTMENT\_NAME;**

**v\_dept\_varray(v\_index).EMPLOYEE\_ID := v\_row.EMPLOYEE\_ID;**

**v\_dept\_varray(v\_index).city := v\_row.CITY;**

**END LOOP;**

**CLOSE c\_dept\_data;**

**FOR i IN 1..v\_dept\_varray.COUNT LOOP**

**DBMS\_OUTPUT.PUT\_LINE(v\_dept\_varray(i).CITY || ': Department ' ||**

**v\_dept\_varray(i).DEPARTMENT\_NAME ||**

**' has ' || v\_dept\_varray(i).EMPLOYEE\_ID || ' employee(s)');**

**END LOOP;**

**END;**

**Output:**

**A screenshot of a computer

Description automatically generated**

*Question 5: [20 Marks]*

*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. [5 pts]*

**Code:**

**CREATE TABLE INVENTORY (**

**INVENTORY\_ID NUMBER(6) PRIMARY KEY,**

**ITEM\_NAME VARCHAR2(255),**

**QUANTITY NUMBER,**

**PRICE NUMBER,**

**ITEM\_SIZE VARCHAR2(255),**

**INVENTORY\_VALUE NUMBER**

**);**

**Output:**

**A screenshot of a computer

Description automatically generated**

*2- Create a trigger that only allows an insert to complete if the SIZE value is one of the*

*following: 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'. [10 pts]*

**code:**

**CREATE OR REPLACE TRIGGER validate\_item\_size**

**BEFORE INSERT OR UPDATE ON INVENTORY**

**FOR EACH ROW**

**BEGIN**

**IF NOT (:NEW.ITEM\_SIZE IN ('S', 'M', 'L', 'XL', 'small', 'medium', 'large', 'extra-large')) THEN**

**RAISE\_APPLICATION\_ERROR(**

**-20001,**

**'Please enter for SIZE only: S, M, L, XL, small, medium, large, or extra-large'**

**);**

**END IF;**

**END;**

**output:**

**A white screen with a black border

Description automatically generated with medium confidence**

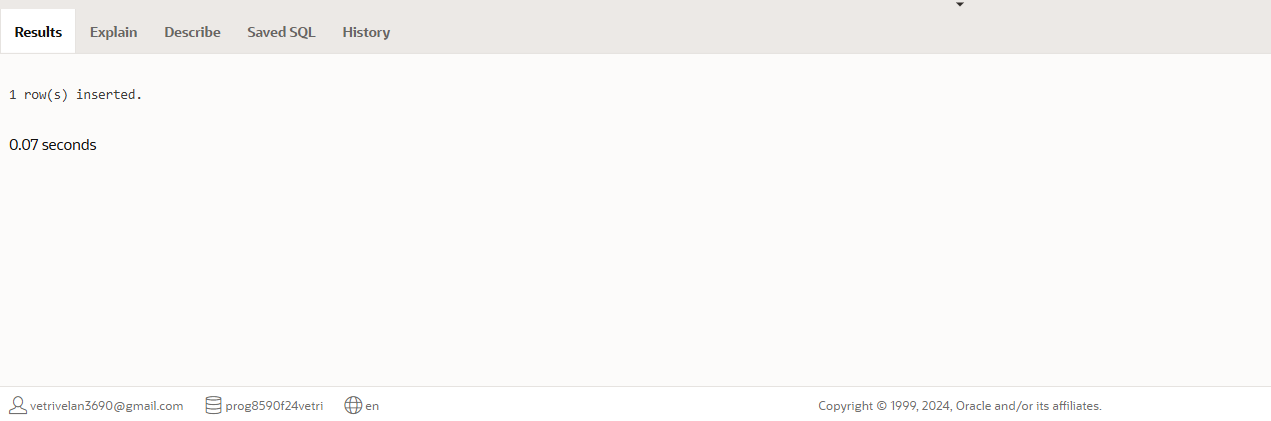
*3- Insert into the INVENTORY table (1, 'Web Shooter, 2, 19.00, 's', 38.00)*

**Code:**

**INSERT INTO INVENTORY (INVENTORY\_ID, ITEM\_NAME, QUANTITY, PRICE, ITEM\_SIZE, INVENTORY\_VALUE)**

**VALUES (1, 'Web Shooter', 2, 19.00, 'S', 38.00);**

**Output:**

****

*Insert into the INVENTORY table (2, 'Fantasticar', 4, 3000.00, 'very big', 12000.00)*

**Code:**

**INSERT INTO INVENTORY (INVENTORY\_ID, ITEM\_NAME, QUANTITY, PRICE, ITEM\_SIZE, INVENTORY\_VALUE)**

**VALUES (2, 'Fantasticar', 4, 3000.00, 'very big', 12000.00);**

**Output:**

**A screenshot of a computer

Description automatically generated**

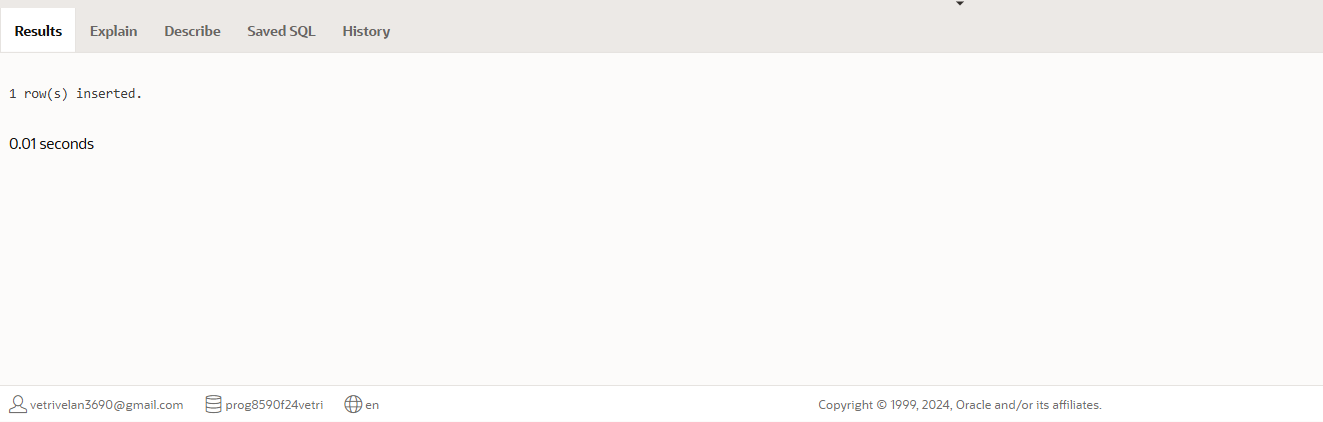
*Insert into the INVENTORY table (3, 'Mjolnir', 1, 100.00, 'medium', 100.00)*

**Code:**

**INSERT INTO INVENTORY (INVENTORY\_ID, ITEM\_NAME, QUANTITY, PRICE, ITEM\_SIZE, INVENTORY\_VALUE)**

**VALUES (3, 'Mjolnir', 1, 100.00, 'medium', 100.00);**

**Output:**

****

**Run a SELECT \* statement on the INVENTORY table. [5 pts]**

**Code:**

**SELECT \***

**FROM INVENTORY;**

**Output:**

**A screenshot of a computer

Description automatically generated**