University of Colombo

Faculty of Technology

Department of ICT

IC 2201 – Database Management Systems II (21/22)

Assignment PL/SQL

Y.E.N.PREMATHILAKA

REG. NO : 2022T01564

SUB. DATE : 28/12/2024

Contents

FACULITY OF TECHNOLOGY UNIVERSITY OF COLOMBO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

IC 2201 – DBMS II (21/22)

Assignment PL/SQL

Name: Eranda Nimsara **Index Numbers: 2022t01564**

1) Create Tables

```
-- Product table
CREATE TABLE Product (
   Product_id VARCHAR2(10) PRIMARY KEY,
   Product_name VARCHAR2(50),
   Warranty_period NUMBER(4, 1),
   Supplier_code VARCHAR2(10),
   List_price NUMBER(10, 2)
);


-- Warehouse table
CREATE TABLE Warehouse (
   Warehouse_id VARCHAR2(10) PRIMARY KEY,
   Warehouse_name VARCHAR2(50),
   Location VARCHAR2(50)
);
```

-- Inventory table

```
CREATE TABLE Inventory (
   Product_id VARCHAR2(10),
   Warehouse_id VARCHAR2(10),
   Qty_on_hand NUMBER(10),
   PRIMARY KEY (Product_id, Warehouse_id),
   FOREIGN KEY (Product_id) REFERENCES Product(Product_id),
   FOREIGN KEY (Warehouse_id) REFERENCES Warehouse(Warehouse_id)
);
```

2) Insert Data into the Tables

-- insert data into product table

```
INSERT INTO Product VALUES ('PRD01', 'Air cooler', 5, 'SW_00101', 25990.00);
INSERT INTO Product VALUES ('PRD02', 'Ceiling fan', 2, 'IN_20034', 6690.00);
INSERT INTO Product VALUES ('PRD03', 'Dry iron', 0.5, 'IN_20034', 2750.00);
INSERT INTO Product VALUES ('PRD04', 'Floor polisher', 1, NULL, 15690.00);
INSERT INTO Product VALUES ('PRD05', 'Stand fan', 0.5, 'SG_34023', 18590.00);
INSERT INTO Product VALUES ('PRD06', 'Steam iron', 0.5, NULL, 2190.00);
INSERT INTO Product VALUES ('PRD07', 'Vacuum cleaner', 1.5, 'SG_34023', 9990.00);
INSERT INTO Product VALUES ('PRD08', 'Water heater', 2, 'TW_90846', 18890.00);
INSERT INTO Product VALUES ('PRD09', 'Water purifier', 2, 'US_56798', 11850.00);
```

-- insert data into warehouse table

```
INSERT INTO Warehouse VALUES ('ST001', 'Shop Warehouse', 'Colombo');
INSERT INTO Warehouse VALUES ('ST002', 'Large Zone', 'Rathmalana');
INSERT INTO Warehouse VALUES ('ST003', 'Retail Zone', 'Kiribathgoda');
INSERT INTO Warehouse VALUES ('ST004', 'Whole Supply', 'Colombo');
```

```
-- insert data into inventory table

INSERT INTO Inventory VALUES ('PRD01', 'ST001', 30);
INSERT INTO Inventory VALUES ('PRD02', 'ST001', 45);
INSERT INTO Inventory VALUES ('PRD02', 'ST002', 20);
INSERT INTO Inventory VALUES ('PRD02', 'ST003', 10);
INSERT INTO Inventory VALUES ('PRD03', 'ST002', 50);
INSERT INTO Inventory VALUES ('PRD03', 'ST004', 50);
INSERT INTO Inventory VALUES ('PRD06', 'ST002', 75);
INSERT INTO Inventory VALUES ('PRD07', 'ST001', 15);
INSERT INTO Inventory VALUES ('PRD07', 'ST003', 10);
```

3) Create a PL/SQL Package

```
CREATE OR REPLACE PACKAGE Inventory_Pkg IS
   -- Public variables
   G_total_profit_no_discount NUMBER := 0;
   G_total_profit_with_discount NUMBER := 0;
   G_discount_percentage NUMBER := 0;

   -- Public procedures and functions
   PROCEDURE Compute_Total_Profit_No_Discount;
   PROCEDURE Compute_Total_Profit_With_Discount(p_discount_percentage NUMBER);
   FUNCTION Get_Discounted_Price(p_product_id VARCHAR2) RETURN NUMBER;
   FUNCTION Get_Discounted_Price(p_product_id VARCHAR2, p_discount_percentage NUMBER) RETURN NUMBER;

   -- Declare Get_Warehouse_Name in the package specification
   PROCEDURE Get_Warehouse_Name(p_warehouse_id VARCHAR2, p_name OUT VARCHAR2);
END Inventory_Pkg;
/
```

a) Warehouse Name (A) package Body

```
CREATE OR REPLACE PACKAGE BODY Inventory_Pkg IS

   -- Private procedure to get warehouse name
   PROCEDURE Get_Warehouse_Name(p_warehouse_id VARCHAR2, p_name OUT VARCHAR2) IS
   BEGIN
      SELECT UPPER(Warehouse_name) INTO p_name
      FROM Warehouse
      WHERE Warehouse_id = p_warehouse_id;
   EXCEPTION
      WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Warehouse ID not found.');
      WHEN OTHERS THEN
         RAISE;
   END Get_Warehouse_Name;
```

--warehouse name

```
SET SERVEROUTPUT ON;
DECLARE
   v_warehouse_name VARCHAR2(100);
BEGIN
   -- Call the procedure to fetch the warehouse name in uppercase
   Inventory_Pkg.Get_Warehouse_Name('ST001', v_warehouse_name);

   -- Display the warehouse name
   DBMS_OUTPUT.PUT_LINE('Warehouse Name in Uppercase: ' || v_warehouse_name);
END;
/
```

b)Standard Discount Functions (B)

```
-- Function to get discounted price (standard discounts)

FUNCTION Get_Discounted_Price(p_product_id VARCHAR2) RETURN NUMBER IS
  v_list_price Product.List_price%TYPE;
  v_discounted_price NUMBER;
BEGIN
  SELECT List_price INTO v_list_price
  FROM Product
  WHERE Product_id = p_product_id;
```

Apply Discounts

```
-- Apply standard discount based on price range

  v_discounted_price := CASE
    WHEN v_list_price < 6000 THEN v_list_price * 0.88
    WHEN v_list_price BETWEEN 6000 AND 12000 THEN v_list_price * 0.84
    ELSE v_list_price * 0.76
  END;

  RETURN v_discounted_price;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Product not found.');
    RETURN NULL;
  WHEN OTHERS THEN
    RAISE;
END Get_Discounted_Price;
```

c) total profit without discounts (C)

```
-- Procedure to compute total profit without discounts

PROCEDURE Compute_Total_Profit_No_Discount IS
   CURSOR cur_inventory IS
      SELECT i.Product_id, i.Qty_on_hand, p.List_price
      FROM Inventory i JOIN Product p ON i.Product_id = p.Product_id;
BEGIN
   G_total_profit_no_discount := 0;
   FOR rec IN cur_inventory LOOP
      G_total_profit_no_discount := G_total_profit_no_discount + (rec.List_price * rec.Qty_on_hand * 0.1);
   END LOOP;
END Compute_Total_Profit_No_Discount;
```

d)Get Discounted Price (D)

```
  -- Overloaded function to get discounted price (custom discount)

  FUNCTION Get_Discounted_Price(p_product_id VARCHAR2, p_discount_percentage NUMBER) RETURN NUMBER IS
    v_list_price Product.List_price%TYPE;
  BEGIN
    SELECT List_price INTO v_list_price
    FROM Product
    WHERE Product_id = p_product_id;

    RETURN v_list_price * (1 - p_discount_percentage);

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Product not found.');
      RETURN NULL;
    WHEN OTHERS THEN
      RAISE;
  END Get_Discounted_Price;
```

e) total profit with discounts (E)

```
  -- Procedure to compute total profit with discounts

   PROCEDURE Compute_Total_Profit_With_Discount(p_discount_percentage NUMBER)
IS
      CURSOR cur_inventory IS
         SELECT i.Product_id, i.Qty_on_hand, p.List_price
         FROM Inventory i JOIN Product p ON i.Product_id = p.Product_id;
   BEGIN
      G_total_profit_with_discount := 0;
      FOR rec IN cur_inventory LOOP
         G_total_profit_with_discount := G_total_profit_with_discount + (rec.List_price * (1 -
p_discount_percentage) * rec.Qty_on_hand * 0.1);
      END LOOP;
   END Compute_Total_Profit_With_Discount;


END Inventory_Pkg;
/




SET SERVEROUTPUT ON;


DECLARE
   v_discounted_price NUMBER;
BEGIN
```

f) Calling Package Methods (F)

```
-- i. Print the total profit of the current inventory with no discounts.
    Inventory_Pkg.Compute_Total_Profit_No_Discount;

   DBMS_OUTPUT.PUT_LINE('Total Profit With No Discount: ' ||
Inventory_Pkg.G_total_profit_no_discount);


   -- ii. Print the total profit of the current inventory with 10% discount for all
products.
    Inventory_Pkg.Compute_Total_Profit_With_Discount(0.10);

   DBMS_OUTPUT.PUT_LINE('Total Profit With 10% Discount: ' ||
Inventory_Pkg.G_total_profit_with_discount);


 -- iii. Print the total profit of the current inventory with 15% discount for all
products.
   Inventory_Pkg.Compute_Total_Profit_With_Discount(0.15);

   DBMS_OUTPUT.PUT_LINE('Total Profit With 15% Discount: ' ||
Inventory_Pkg.G_total_profit_with_discount);


-- iv. Print the discounted price of the product PRD01 when standard discount
percentages are applied.
   v_discounted_price := Inventory_Pkg.Get_Discounted_Price('PRD01');

   DBMS_OUTPUT.PUT_LINE('Discounted Price of PRD01 (Standard Discount): ' ||
v_discounted_price);


    -- v. Print the discounted price of the product PRD01 when a 20% discount is
applied.
   v_discounted_price := Inventory_Pkg.Get_Discounted_Price('PRD01', 0.20);

   DBMS_OUTPUT.PUT_LINE('Discounted Price of PRD01 (20% Discount): ' ||
v_discounted_price);

END;

/
```

Outputs

```
Total Profit With No Discount: 197045
Total Profit With 10% Discount: 177340.5
Total Profit With 15% Discount: 167488.25
Discounted Price of PRD01 (Standard Discount): 19752.4
Discounted Price of PRD01 (20% Discount): 20792
```