-- 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)

);

-- 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);

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;

END Inventory\_Pkg;

/

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;

-- 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 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 11999 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;

-- 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;

-- 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;

-- 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

-- 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;

/