Homework #3: Cursors

Version 1

In this assignment, you’ll:

- Answer two questions to demonstrate your understanding of PL/SQL Cursors.

- Brainstorm a real-world problem where PL/SQL Cursors could provide a solution.

**Total Points: 40**

# Task 1: Understanding PL/SQL Cursors (8 pts)

**1a. (4pts)** **Rank code order.**

Here are four steps you need when you develop an EXPLICIT cursor. Please rank them in the correct order.

[3] FETCH the cursor for retrieving data (row by row)

[2] OPEN the cursor for allocating memory

[1] DECLARE the cursor for initializing in the memory

[3] CLOSE the cursor to release allocated memory

**1b. (4pts) Complete the given query.**

If I want to print the customer full names, what should I put between the parentheses in the code below?

begin  
     FOR mycustomer IN (select CUSTFIRSTNAME || ' ' || CUSTLASTNAME as c\_name from customer where state='IN') LOOP  
 dbms\_output.put\_line(mycustomer.c\_name);  
     END LOOP;   
end;

# Task 2: Developing a PL/SQL Exercise (16 pts)

In this task, you’ll create a PL/SQL exercise based on a real-world scenario. Your exercise should be meaningful and relevant, helping others practice and apply PL/SQL skills.

**2a**.**(2pts)** **Identify a Real-World Problem**

Reflect on a situation in your personal experience or an area of interest where PL/SQL Cursors could be used to solve a problem. Briefly describe the scenario. (Please come up with a different scenario from the given example.

*A warehouse refurbishes and sells returned products. The parent company sends the warehouse packaging material so that products that are refurbished can be resold. There often is a shortage of the necessary packaging material to completely refurbish and send out products which leads to products sitting and gathering dust. The system should track packaging material by product brand and model and last shipped date. PL/SQL cursors can be used to loop through the packaging inventory and identify packaging material that has a low stock count based on a certain time duration as well as products that have a high number of returns and request the parent company to ship more packaging material over.*

**2b**.**(4 pts) Formulate the Problem Statement**

Provide more details about the example data that would be available within this problem. Think about the specific information that would be stored and the questions the cursors will help answer. Define a meaningful question that needs to be answered to address the real-world problem you described in 2a.

*The warehouse has a Products table that stores the productid, brand, model, and returnCount. There is also a PackagingMaterial table that stores the quantity of packaging material based on the productID. The PackagingMaterial table also has a lastshippedate column to show when the packaging material was last shipped*

**2c. Formulate an Exercise**

Design an exercise that includes at least two example tables to support your problem. This exercise should require at least three steps to solve, allowing others to practice PL/SQL skills in a realistic scenario.

**2c. (i) Define Your Tables (4 pts)**  
Create at least two tables, specifying column names and data types.

*Table 1: Product*

* *ProductID (INTEGER) – Unique identifier for each product*
* *ProductBrand (VARCHAR2(30)) – Brand name of each product*
* *ProductModel (VARCHAR2(20)) – Model name of each specific product model*
* *ReturnCount (INTEGER) – Number of items returned*

*Table 2: PackagingMaterial*

* *ProductBrand (VARCHAR2(30)) – Brand name of each product*
* *ProductModel (VARCHAR2(20)) – Model name of each specific product model*
* *Quantity (INTEGER) – Quantity of packaging materials remaining based on time since last shipped and ReturnCount*
* *LastShipDate (DATE) – Date of last shipment of packaging material*

**2c. (ii) Prepare a SQL File (6 pts)**  
Create and upload an SQL file that includes the SQL statements to define your tables and insert sample data. This file will enable others to load and test the data necessary for the exercise.

*CREATE TABLE product (*

*Productid NOT NULL INTEGER PRIMARY KEY,*

*productbrand VARCHAR2(30),*

*productmodel VARCHAR2(20),*

*returncount INTEGER*

*);*

*CREATE TABLE packagingmaterial (*

*productbrand VARCHAR2(30),*

*productmodel VARCHAR2(20),*

*quantity INTEGER,*

*lastshipdate DATE,*

*productid INTEGER,*

*CONSTRAINT product\_fk FOREIGN KEY (productid)*

*REFERENCES product (productid)*

*);*

# *INSERT INTO product (productid, productbrand, productmodel, returncount) VALUES (1, 'Onkore', 'OK100', 10);*

# *INSERT INTO product (productid, productbrand, productmodel, returncount) VALUES (2, 'TechStar', 'TS200', 5);*

# *INSERT INTO product (productid, productbrand, productmodel, returncount) VALUES (3, 'DogWorks', 'DW300', 15);*

# *INSERT INTO product (productid, productbrand, productmodel, returncount) VALUES (4, 'InnovaTech', 'IT400', 8);*

# *INSERT INTO product (productid, productbrand, productmodel, returncount) VALUES (5, 'BizStar', 'BS500', 12);*

# *INSERT INTO packagingmaterial (productbrand, productmodel, quantity, lastshipdate, productid) VALUES ('Onkore', 'OK100', 50, '2024-10-01', 1);*

# *INSERT INTO packagingmaterial (productbrand, productmodel, quantity, lastshipdate, productid) VALUES ('TechStar', 'TS200', 20, '2024-09-15', 2);*

# *INSERT INTO packagingmaterial (productbrand, productmodel, quantity, lastshipdate, productid) VALUES ('DogWorks', 'DW300', 5, '2024-11-01', 3);*

# *INSERT INTO packagingmaterial (productbrand, productmodel, quantity, lastshipdate, productid) VALUES ('InnovaTech', 'IT400', 30, '2024-08-20', 4);*

*INSERT INTO packagingmaterial (productbrand, productmodel, quantity, lastshipdate, productid) VALUES ('BizStar', 'BS500', 25, '2024-07-10', 5);*

# Task 3: Step-by-Step Solution (12 pts)

Provide a step-by-step solution to solve the exercise you created in Task 2.

(Step description: 2 pts / step, step solution: 2pts / step; at least 3 steps are needed to get full points for this task)

***Step 1: Declare and Open a Cursor to Select Products with High Return Counts***

*First, we need to identify products with high return counts. Let’s assume a high return count is more than 10 returns.*

*DECLARE*

*CURSOR high\_return\_cursor IS*

*SELECT productid, productbrand, productmodel, returncount*

*FROM product*

*WHERE returncount > 10;*

*BEGIN*

*OPEN high\_return\_cursor;*

*END;*

***Step 2: Fetch Data from the Cursor and Check Packaging Material Stock***

*Next, we fetch data from the cursor and check the packaging material stock for each product with a high return count. We will identify packaging materials with quantities below a certain threshold (e.g., 10 units).*

*DECLARE*

*CURSOR high\_return\_cursor IS*

*SELECT productid, productbrand, productmodel, returncount*

*FROM product*

*WHERE returncount > 10;*

*v\_productid product.productid%TYPE;*

*v\_productbrand product.productbrand%TYPE;*

*v\_productmodel product.productmodel%TYPE;*

*v\_returncount product.returncount%TYPE;*

*BEGIN*

*OPEN high\_return\_cursor;*

*LOOP*

*FETCH high\_return\_cursor INTO v\_productid, v\_productbrand, v\_productmodel, v\_returncount;*

*EXIT WHEN high\_return\_cursor%NOTFOUND;*

*-- Check packaging material stock for the product*

*DECLARE*

*CURSOR packaging\_cursor IS*

*SELECT quantity, lastshipdate*

*FROM packagingmaterial*

*WHERE productid = v\_productid AND quantity < 10;*

*v\_quantity packagingmaterial.quantity%TYPE;*

*v\_lastshipdate packagingmaterial.lastshipdate%TYPE;*

*BEGIN*

*OPEN packaging\_cursor;*

*LOOP*

*FETCH packaging\_cursor INTO v\_quantity, v\_lastshipdate;*

*EXIT WHEN packaging\_cursor%NOTFOUND;*

*DBMS\_OUTPUT.PUT\_LINE('Product: ' || v\_productbrand || ' ' || v\_productmodel || ' has low packaging stock: ' || v\_quantity || ' units. Last shipped: ' || v\_lastshipdate);*

*END LOOP;*

*CLOSE packaging\_cursor;*

*END;*

*END LOOP;*

*CLOSE high\_return\_cursor;*

*END;*

***Step 3: Close the Cursor After Processing All Data***

*Finally, ensure that all cursors are closed after processing the data*

*DECLARE*

*CURSOR high\_return\_cursor IS*

*SELECT productid, productbrand, productmodel, returncount*

*FROM product*

*WHERE returncount > 10;*

*v\_productid product.productid%TYPE;*

*v\_productbrand product.productbrand%TYPE;*

*v\_productmodel product.productmodel%TYPE;*

*v\_returncount product.returncount%TYPE;*

*BEGIN*

*OPEN high\_return\_cursor;*

*LOOP*

*FETCH high\_return\_cursor INTO v\_productid, v\_productbrand, v\_productmodel, v\_returncount;*

*EXIT WHEN high\_return\_cursor%NOTFOUND;*

*-- Check packaging material stock for the product*

*DECLARE*

*CURSOR packaging\_cursor IS*

*SELECT quantity, lastshipdate*

*FROM packagingmaterial*

*WHERE productid = v\_productid AND quantity < 10;*

*v\_quantity packagingmaterial.quantity%TYPE;*

*v\_lastshipdate packagingmaterial.lastshipdate%TYPE;*

*BEGIN*

*OPEN packaging\_cursor;*

*LOOP*

*FETCH packaging\_cursor INTO v\_quantity, v\_lastshipdate;*

*EXIT WHEN packaging\_cursor%NOTFOUND;*

*DBMS\_OUTPUT.PUT\_LINE('Product: ' || v\_productbrand || ' ' || v\_productmodel || ' has low packaging stock: ' || v\_quantity || ' units. Last shipped: ' || v\_lastshipdate);*

*END LOOP;*

*CLOSE packaging\_cursor;*

*END;*

*END LOOP;*

*CLOSE high\_return\_cursor;*

*END;*

# Task 4: Reflection (4 pts)

4a. (2pts) In a few sentences, describe any challenges you faced or additional support that could have helped create this exercise.

*The main challenges I faced were figuring out how to connect the cursor with the data in the tables and to check for low packing material versus high return count. Then, it was closing the cursors properly to ensure that everything ended properly and that there were no open ends that could lead to errors*

4b. (2pts) Briefly explain why you have chosen to work on this version of HW3.

I chose this version of Homework 3 because it allowed me to be a little creative in the problem. The problem itself is a legitimate problem of where I worked before although the information in the tables itself is dummy data. I think that what I did has potential to help the warehouse out when figuring out how to deal with shortages.