**CS 276: Lab 6 Name: Blake Bryant**

This exercise is intended to provide practice with collections, procedures and packages. Use the same database you used for the midterm exam. It is suggested you read the entire assignment before you begin.

1. Create a procedure that has an IN parameter of a pledge id and returns in an OUT parameter a collection that contains all the payments that have been made for that pledge id. You may use any of the three types of collections discussed in class this week. It is your choice. One approach is to use a cursor to get all the payments for that pledge id and use the cursor to place all the payments into the collection. This procedure should handle the exception of No Data Found for a pledge id that does not exist by outputting an appropriate message.

CREATE OR REPLACE PACKAGE PAYMENT\_LAB\_PCK IS

TYPE LIST\_OF\_PAYMENTS IS TABLE OF PAYMENT%ROWTYPE

INDEX BY PLS\_INTEGER;

END PAYMENT\_LAB\_PCK;

CREATE OR REPLACE PROCEDURE CREATE\_PAYMENT\_COLLECTION

(PLEDGE\_ID\_ARG IN NUMBER,

PAYMENT\_LIST OUT PAYMENT\_LAB\_PCK.LIST\_OF\_PAYMENTS)

IS

-- GET PAYMENT DATA

CURSOR PAYMENT\_CUR IS

SELECT \*

FROM PAYMENT

WHERE PAYMENT.IDPLEDGE = PLEDGE\_ID\_ARG;

-- SETUP ROW INDEX

ROW\_INDEX PLS\_INTEGER := 0;

BEGIN

-- FILL ARRAY

FOR PAYMENT\_REC IN PAYMENT\_CUR

LOOP

-- PUT RECORD DATA INTO ARRAY

PAYMENT\_LIST (ROW\_INDEX) := PAYMENT\_REC;

-- INCREMENT INDEX

ROW\_INDEX := ROW\_INDEX + 1;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('A NO DATA ERROR HAS OCCURED IN THE AGENT BONUS. MOST LIKELY BECAUSE PLEDGE ID DOES NOT EXIST IN DATABASE.');

END;

-- TEST PAYMENT COLLECTION

SELECT \* FROM PAYMENT WHERE IDPLEDGE = 109;

DESCRIBE PAYMENT;

DECLARE

-- PAYMENT COLLECTION

PAYMENT\_LIST PAYMENT\_LAB\_PCK.LIST\_OF\_PAYMENTS;

BEGIN

-- GET PAYMENT DATA

CREATE\_PAYMENT\_COLLECTION(109, PAYMENT\_LIST);

-- PRINT ARRAY

FOR INDX IN PAYMENT\_LIST.FIRST .. PAYMENT\_LIST.LAST

LOOP

DBMS\_OUTPUT.PUT\_LINE('------------------------------------------');

DBMS\_OUTPUT.PUT\_LINE('Payment ID is: ' || PAYMENT\_LIST(INDX).IDPAY);

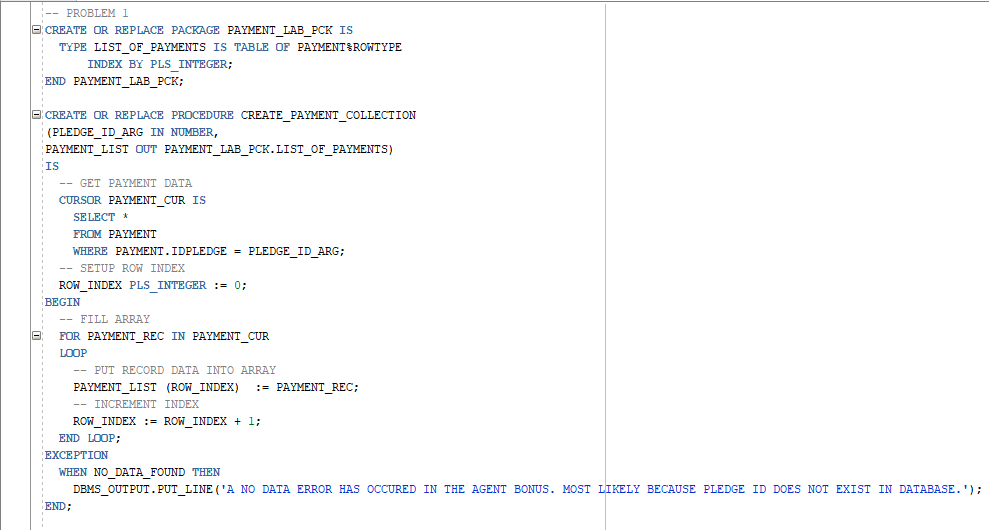
DBMS\_OUTPUT.PUT\_LINE('Payment amount is: ' || PAYMENT\_LIST(INDX).PAYAMT);

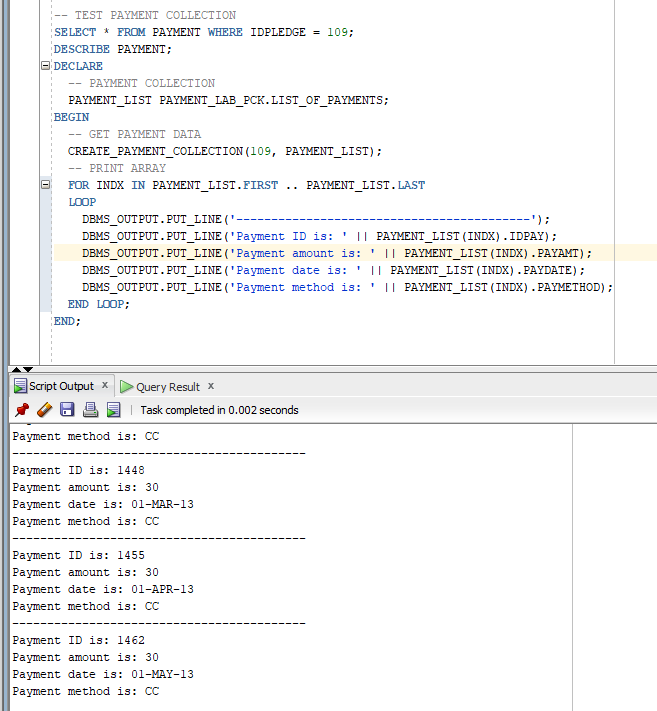
DBMS\_OUTPUT.PUT\_LINE('Payment date is: ' || PAYMENT\_LIST(INDX).PAYDATE);

DBMS\_OUTPUT.PUT\_LINE('Payment method is: ' || PAYMENT\_LIST(INDX).PAYMETHOD);

END LOOP;

END;





1. Create a second procedure that has an IN parameter of the pledge id and returns in an OUT parameter the sum of all the payments for that pledge id and returns in another OUT parameter the average of all the payments for that pledge id. This procedure must call the first procedure using the pledge id it was given. It should use the collection it gets back from the first procedure to calculate the sum and the average of the payments.

-- PROBLEM 2

CREATE OR REPLACE PROCEDURE AVERAGE\_PAYMENTS

(PLEDGE\_ID\_ARG IN NUMBER,

PAYMENT\_AMT\_SUM OUT NUMBER,

PAYMENT\_AMT\_AVERAGE OUT NUMBER)

IS

-- PAYMENT COLLECTION

PAYMENT\_LIST PAYMENT\_LAB\_PCK.LIST\_OF\_PAYMENTS;

PAYMENT\_SUM\_COUNTER NUMBER := 0;

BEGIN

-- GET PAYMENT DATA

CREATE\_PAYMENT\_COLLECTION(109, PAYMENT\_LIST);

-- CALCULATE SUM AND AVERAGE

FOR INDX IN PAYMENT\_LIST.FIRST .. PAYMENT\_LIST.LAST

LOOP

PAYMENT\_SUM\_COUNTER := PAYMENT\_LIST(INDX).PAYAMT + PAYMENT\_SUM\_COUNTER;

END LOOP;

PAYMENT\_AMT\_SUM := PAYMENT\_SUM\_COUNTER;

PAYMENT\_AMT\_AVERAGE := (PAYMENT\_SUM\_COUNTER / PAYMENT\_LIST.COUNT);

END;

-- TEST PROBLEM 2

DECLARE

PLEDGE\_ID NUMBER := 109;

PAYMENT\_SUM NUMBER;

PAYMENT\_AVERAGE NUMBER;

BEGIN

-- CALCULATE PAYMENT DATA

AVERAGE\_PAYMENTS(PLEDGE\_ID, PAYMENT\_SUM, PAYMENT\_AVERAGE);

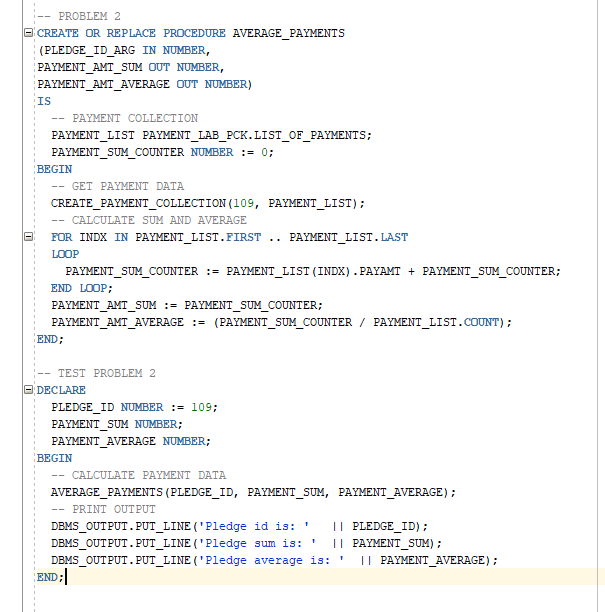
-- PRINT OUTPUT

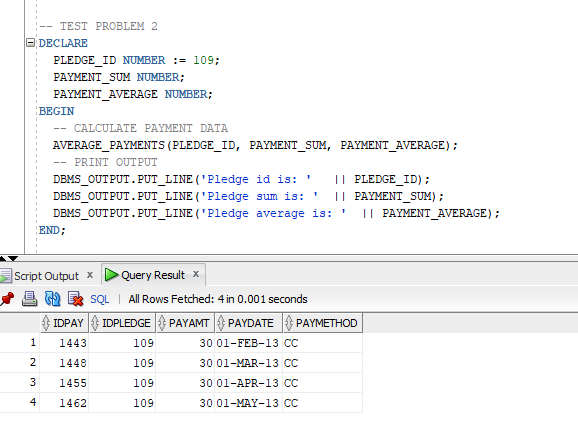
DBMS\_OUTPUT.PUT\_LINE('Pledge id is: ' || PLEDGE\_ID);

DBMS\_OUTPUT.PUT\_LINE('Pledge sum is: ' || PAYMENT\_SUM);

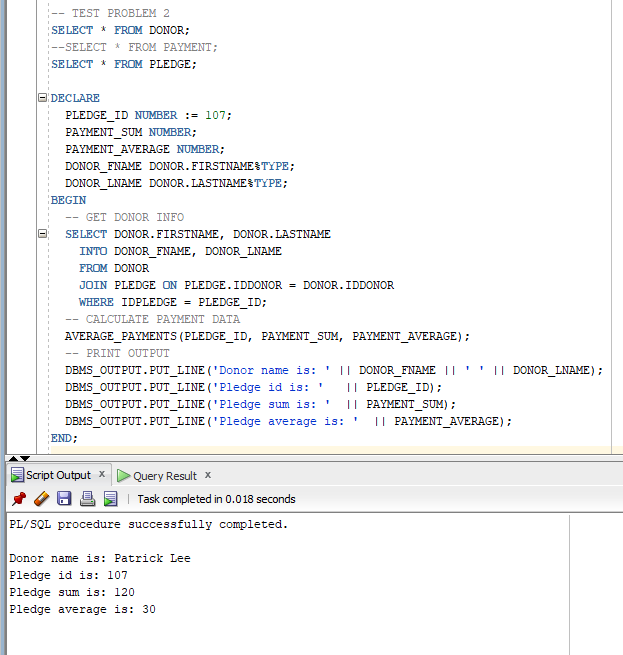
DBMS\_OUTPUT.PUT\_LINE('Pledge average is: ' || PAYMENT\_AVERAGE);

END;





1. Create an anonymous block that calls the second procedure passing a pledge id of 107. Your anonymous block should output the pledge id, the donor first name and last name that made that pledge, the total of the payments, and the average of the payments.



1. Place the two procedures in a package where they are both able to be called from modules outside the package. Place the anonymous block outside the package.

**See screenshots below**

1. Paste below screen shots of all your code in the entire package and the anonymous block, and paste a screen shot of the results of running your anonymous block with a pledge id of 107.



