**Student: Bruno Cantanhede Morgado**

**Student # 301-154-898**

**Assignment 2 COMP214\_M22**

Upload your ascii text file to the Dropbox with .txt extension.

Your files must be in sql script format so that they can be run in SQL Developer and have DBMS\_OUTPUT.PUT\_LINE for presenting results.

Update database with script file(s) from `Student Data` of Oracle 11g: PL/SQL Programming -- Chapter 4.

Answer Questions Assignment 4-9, 4-11, 4-12, 4-13 from  Chapter 4  of Oracle 11g: PL/SQL Programming **using cursors**.

/\* **Assignment 4-9**: Using an Explicit Cursor

Create a block to retrieve and display pledge and payment information for a specific donor. For

each pledge payment from the donor, display the pledge ID, pledge amount, number of monthly

payments, payment date, and payment amount. The list should be sorted by pledge ID and then

by payment date. For the first payment made for each pledge, display “first payment” on that

output row.\*/

**DECLARE**

**CURSOR pledged\_payment (donor\_id dd\_donor.iddonor%TYPE) IS**

**SELECT (don.firstname || ' ' || don.lastname) donor, pld.idpledge, pld.pledgeamt, pld.paymonths, pay.paydate**

**FROM dd\_donor don**

**JOIN dd\_pledge pld ON pld.iddonor = don.iddonor**

**JOIN dd\_payment pay ON pld.idpledge = pay.idpledge**

**WHERE don.iddonor = donor\_id**

**ORDER BY pld.idpledge, pay.paydate;**

**lv\_don1\_num dd\_donor.iddonor%TYPE := 301;**

**lv\_don2\_num dd\_donor.iddonor%TYPE := 302;**

**lv\_curr\_pledge NUMBER(4) := 0;**

**BEGIN**

**FOR cur\_pledge IN pledged\_payment(lv\_don1\_num) LOOP**

**IF lv\_curr\_pledge = 0 OR lv\_curr\_pledge != cur\_pledge.idpledge THEN**

**DBMS\_OUTPUT.PUT\_LINE('\* FIRST PAYMENT \*');**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE('Donor: ' || cur\_pledge.donor);**

**DBMS\_OUTPUT.PUT\_LINE('Pledge ID: ' || cur\_pledge.idpledge);**

**DBMS\_OUTPUT.PUT\_LINE( 'Amount: ' || cur\_pledge.pledgeamt || ' --- ' || ' Payment Months: ' || cur\_pledge.paymonths || ' --- ' || ' Pay Date: ' || cur\_pledge.paydate);**

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

**lv\_curr\_pledge := cur\_pledge.idpledge;**

**END LOOP;**

**END;**

**/**

/\*

**Assignment 4-11**: Adding Cursor Flexibility

An administration page in the DoGood Donor application allows employees to enter multiple

combinations of donor type and pledge amount to determine data to retrieve. Create a block

with a single cursor that allows retrieving data and handling multiple combinations of donor type

and pledge amount as input. The donor name and pledge amount should be retrieved and

displayed for each pledge that matches the donor type and is greater than the pledge amount

indicated. Use a collection to provide the input data. Test the block using the following input

data. Keep in mind that these inputs should be processed with one execution of the block. The

donor type code I represents Individual, and B represents Business.

\*/

**DECLARE**

**TYPE search\_donor IS RECORD**

**(**

**donor\_type dd\_donor.typecode%TYPE,**

**amount dd\_pledge.pledgeamt%TYPE**

**);**

**TYPE table\_of\_donors IS TABLE OF search\_donor;**

**first\_search search\_donor;**

**second\_search search\_donor;**

**donors table\_of\_donors := table\_of\_donors();**

**CURSOR donor\_pledged (donors table\_of\_donors, curr\_index NUMBER) IS**

**SELECT (don.firstname || ' ' || don.lastname) donor, pld.pledgeamt**

**FROM dd\_donor don**

**JOIN dd\_pledge pld USING(iddonor)**

**WHERE don.typecode IN donors(curr\_index).donor\_type**

**AND pld.pledgeamt > donors(curr\_index).amount;**

**BEGIN**

**first\_search.donor\_type := 'I';**

**first\_search.amount := 250;**

**second\_search.donor\_type := 'B';**

**second\_search.amount := 500;**

**donors.EXTEND;**

**donors(1) := first\_search;**

**donors.EXTEND;**

**donors(2) := second\_search;**

**FOR i IN 1 .. donors.COUNT LOOP**

**FOR cur\_pledge IN donor\_pledged (donors, i) LOOP**

**DBMS\_OUTPUT.PUT\_LINE('Donor: ' || cur\_pledge.donor);**

**DBMS\_OUTPUT.PUT\_LINE('Pledge Amount: ' || cur\_pledge.pledgeamt);**

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

**END LOOP;**

**END LOOP;**

**END;**

**/**

/\*

**Assignment 4-12**: Using a Cursor Variable

Create a block with a single cursor that can perform a different query of pledge payment data

based on user input. Input provided to the block includes a donor ID and an indicator value of

D or S. The D represents details and indicates that each payment on all pledges the donor has

made should be displayed. The S indicates displaying summary data of the pledge payment

total for each pledge the donor has made.

\*/

**DECLARE**

**cv\_pledge SYS\_REFCURSOR;**

**TYPE payment\_detailed IS RECORD**

**(**

**donor\_id dd\_donor.iddonor%TYPE,**

**pledge\_id dd\_pledge.idpledge%TYPE,**

**payment dd\_payment.payamt%TYPE**

**);**

**TYPE payment\_summary IS RECORD**

**(**

**donor\_id dd\_donor.iddonor%TYPE,**

**total\_payment dd\_payment.payamt%TYPE**

**);**

**details payment\_detailed;**

**summary payment\_summary;**

**donor\_id dd\_donor.iddonor%TYPE;**

**indicator\_value CHAR(1);**

**BEGIN**

**donor\_id := 302;**

**indicator\_value := 'S';**

**IF indicator\_value = 'D' THEN**

**OPEN cv\_pledge FOR SELECT**

**iddonor, idpledge, payamt**

**FROM dd\_donor**

**JOIN dd\_pledge USING(iddonor)**

**JOIN dd\_payment USING(idpledge)**

**WHERE iddonor = donor\_id;**

**LOOP**

**FETCH cv\_pledge INTO details;**

**EXIT WHEN cv\_pledge%NOTFOUND;**

**DBMS\_OUTPUT.PUT\_LINE('Donor ID: ' || details.donor\_id);**

**DBMS\_OUTPUT.PUT\_LINE('Pledge ID: ' || details.pledge\_id);**

**DBMS\_OUTPUT.PUT\_LINE('Payment: ' || details.payment);**

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

**END LOOP;**

**ELSIF indicator\_value = 'S' THEN**

**OPEN cv\_pledge FOR SELECT**

**iddonor, SUM(payamt)**

**FROM dd\_donor**

**JOIN dd\_pledge USING(iddonor)**

**JOIN dd\_payment USING(idpledge)**

**GROUP BY iddonor**

**HAVING iddonor = donor\_id;**

**LOOP**

**FETCH cv\_pledge INTO summary;**

**EXIT WHEN cv\_pledge%NOTFOUND;**

**DBMS\_OUTPUT.PUT\_LINE('Donor ID: ' || summary.donor\_id);**

**DBMS\_OUTPUT.PUT\_LINE('Total Payment: ' || summary.total\_payment);**

**END LOOP;**

**END IF;**

**END;**

**/**

/\*

**Assignment 4-13**: Exception Handling

The DoGood Donor application contains a page that allows administrators to change the ID

assigned to a donor in the DD\_DONOR table. Create a PL/SQL block to handle this task.

Include exception-handling code to address an error raised by attempting to enter a duplicate

donor ID. If this error occurs, display the message “This ID is already assigned.” Test the code

by changing donor ID 305. (Don’t include a COMMIT statement; roll back any DML actions used.)

\*/

**DECLARE**

**ex\_duplicate\_id EXCEPTION;**

**PRAGMA exception\_init(ex\_duplicate\_id, -20001);**

**lv\_donor\_id dd\_donor.iddonor%TYPE := 305;**

**lv\_new\_id dd\_donor.iddonor%TYPE := 301;**

**lv\_exist NUMBER;**

**BEGIN**

**UPDATE dd\_donor**

**SET iddonor = lv\_new\_id**

**WHERE iddonor = lv\_donor\_id;**

**SELECT COUNT(\*)**

**INTO lv\_exist**

**FROM dd\_donor**

**WHERE iddonor = lv\_new\_id;**

**IF lv\_exist = 1 THEN**

**RAISE ex\_duplicate\_id;**

**END IF;**

**EXCEPTION**

**WHEN ex\_duplicate\_id THEN**

**ROLLBACK;**

**DBMS\_OUTPUT.PUT\_LINE('\* This ID is already assigned. \*');**

**WHEN DUP\_VAL\_ON\_INDEX THEN**

**ROLLBACK;**

**DBMS\_OUTPUT.PUT\_LINE('Duplicate Value on Index: '||sqlerrm);**

**WHEN OTHERS THEN**

**ROLLBACK;**

**DBMS\_OUTPUT.PUT\_LINE('\* Please verify the imput information. A problem has occurred. \*');**

**END;**

**/**