Assignment #2 –

**Due Date:** As per dropbox

**Purpose:** The purpose of this assignment is to help you:

1. Become familiar with Cursors, Exceptions, Procedures , Functions and Triggers

**Instructions**: Be sure to read the following general instructions carefully:

This assignment should be completed individually by all the students. Submit your solution **through the dropbox.** Your submission should include PL/SQL code and the screenshot of code execution result, the submission must be named according to the following rule: **studentID(yourlastname)\_Assignment#number.doc**. e.g., 300123456(**smith)\_Assignment#2**.doc

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 3.

Assignment 3-9: Retrieving Pledge Totals Create a PL/SQL block that retrieves and displays information for a specific project based on Project ID. Display the following on a single row of output: project ID, project name, number of pledges made, total dollars pledged, and the average pledge amount.

Assignment 3-10: Adding a Project Create a PL/SQL block to handle adding a new project. Create and use a sequence named DD\_PROJID\_SEQ to handle generating and populating the project ID. The first number issued by this sequence should be 530, and no caching should be used. Use a record variable to handle the data to be added. Data for the new row should be the following: project name = HK Animal Shelter Extension, start = 1/1/2013, end = 5/31/2013, and fundraising goal = $65,000. Any columns not addressed in the data list are currently unknown.

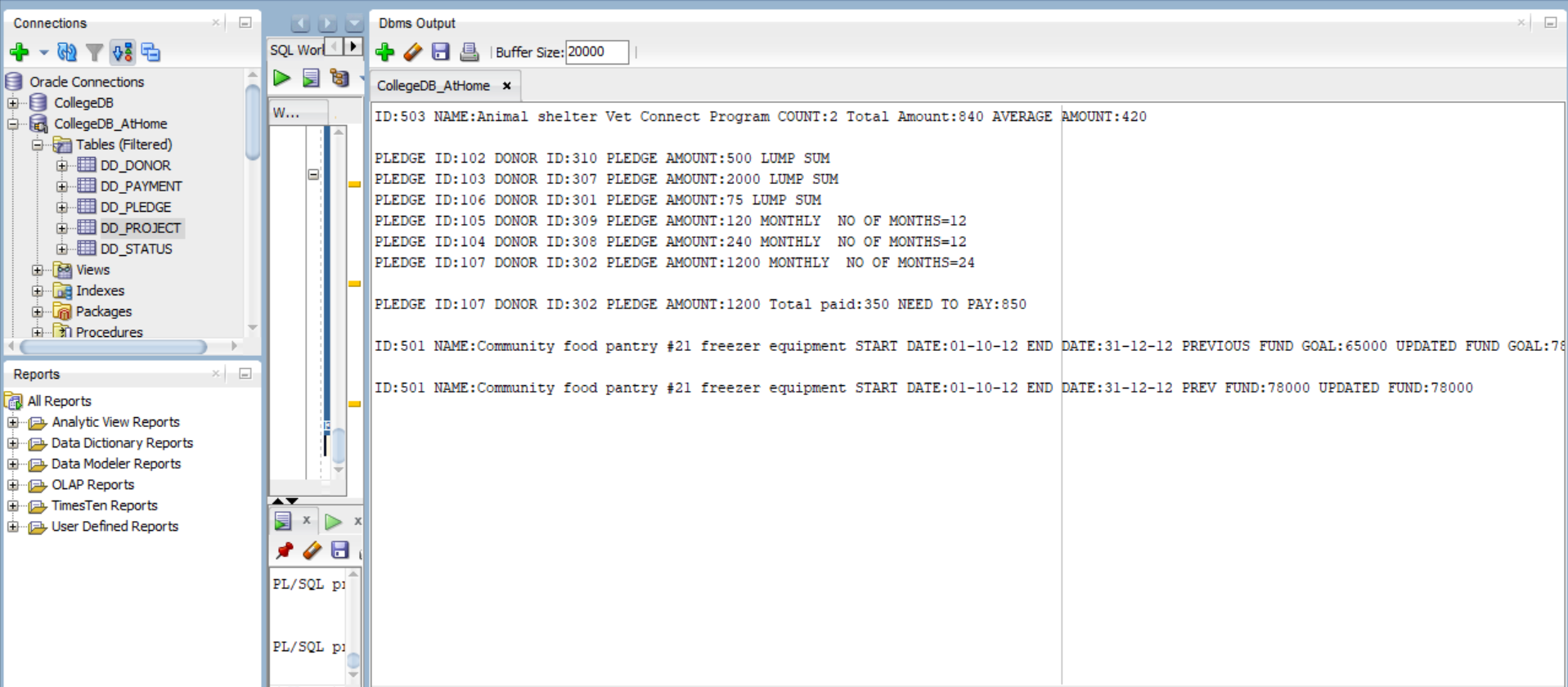
Assignment 3-11: Retrieving and Displaying Pledge Data Create a PL/SQL block to retrieve and display data for all pledges made in a specified month. One row of output should be displayed for each pledge. Include the following in each row of output:

• Pledge ID, donor ID, and pledge amount • If the pledge is being paid in a lump sum, display “Lump Sum.” • If the pledge is being paid in monthly payments, display “Monthly - #” (with the # representing the number of months for payment). • The list should be sorted to display all lump sum pledges first.

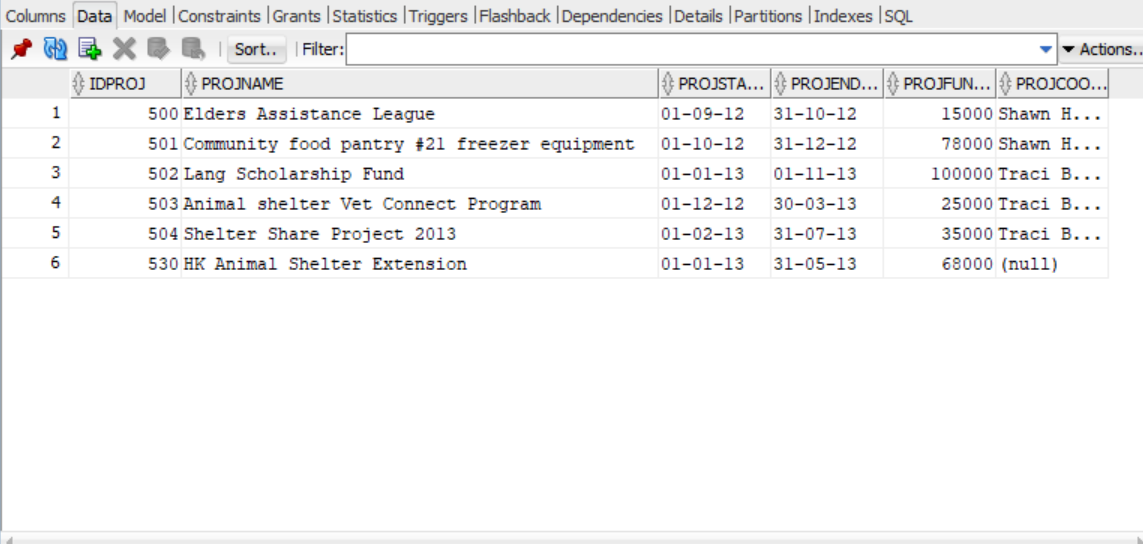
Assignment 3-12: Retrieving a Specific Pledge Create a PL/SQL block to retrieve and display information for a specific pledge. Display the pledge ID, donor ID, pledge amount, total paid so far, and the difference between the pledged amount and total paid amount.

Assignment 3-13: Modifying Data Create a PL/SQL block to modify the fundraising goal amount for a specific project. In addition, display the following information for the project being modified: project name, start date, previous fundraising goal amount, and new fundraising goal amount.

**Output for 3.9, 3.11, 3.12, 3.13:**



Changes in the table DD\_PROJECT for 3.10, 3.13.



3.9

DECLARE

TYPE pledge\_totals\_type IS RECORD(

project\_id dd\_project.idproj%TYPE,

project\_name dd\_project.projname%TYPE,

no\_of\_pledges NUMBER(3),

total\_amount NUMBER(5),

average\_amount NUMBER(5));

rec\_pledges pledge\_totals\_type;

lv\_project\_id NUMBER(3):= 503;

BEGIN

SELECT d1.idproj, d1.projname, d2.pledges, d2.total, d2.average

INTO rec\_pledges

FROM dd\_project d1 INNER JOIN (

SELECT idproj, COUNT(idpledge) pledges, SUM(pledgeamt) total, AVG(pledgeamt) average

FROM dd\_pledge

GROUP BY idproj) d2

ON d1.idproj = d2.idproj

WHERE d1.idproj = lv\_project\_id;

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE('NO DATA FOUND');

END IF;

DBMS\_OUTPUT.PUT\_LINE('ID:'|| rec\_pledges.project\_id ||' NAME:'|| rec\_pledges.project\_name

||' COUNT:'|| rec\_pledges.no\_of\_pledges ||' Total Amount:'|| rec\_pledges.total\_amount

||' AVERAGE AMOUNT:'|| rec\_pledges.average\_amount);

END;

3.10

CREATE SEQUENCE DD\_PROJID\_SEQ

INCREMENT BY 1

START WITH 530;

DECLARE

lv\_project\_name VARCHAR(40);

lv\_start DATE;

lv\_end DATE;

lv\_fund\_goal NUMBER(6);

BEGIN

lv\_project\_name := 'HK Animal Shelter Extension';

lv\_start := '01-01-13';

lv\_end :='31-05-13';

lv\_fund\_goal := 68000;

INSERT INTO dd\_project (idproj, projname, projstartdate, projenddate, projfundgoal)

VALUES (dd\_projid\_seq.nextval, lv\_project\_name, lv\_start, lv\_end, lv\_fund\_goal);

END;

3.11

DECLARE

CURSOR cur\_pledge(p\_month NUMBER) IS

SELECT idpledge, iddonor, pledgeamt, paymonths

FROM dd\_pledge

WHERE Extract(Month from pledgedate) = p\_month

ORDER BY paymonths;

TYPE type\_pledge IS RECORD(

pledge\_id dd\_pledge.idpledge%TYPE,

donor\_id dd\_pledge.iddonor%TYPE,

pledge\_amount dd\_pledge.pledgeamt%TYPE,

pay\_months dd\_pledge.paymonths%TYPE);

rec\_pledges type\_pledge;

lv\_month VARCHAR(2);

BEGIN

lv\_month:= '10';

OPEN cur\_pledge(lv\_month);

LOOP

FETCH cur\_pledge INTO rec\_pledges;

EXIT WHEN cur\_pledge%NOTFOUND;

IF rec\_pledges.pay\_months = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('PLEDGE ID:'|| rec\_pledges.pledge\_id

||' DONOR ID:'||rec\_pledges.donor\_id||' PLEDGE AMOUNT:'||rec\_pledges.pledge\_amount||' LUMP SUM');

ELSE

DBMS\_OUTPUT.PUT\_LINE('PLEDGE ID:'|| rec\_pledges.pledge\_id

||' DONOR ID:'||rec\_pledges.donor\_id||' PLEDGE AMOUNT:'||rec\_pledges.pledge\_amount

||' MONTHLY #'|| rec\_pledges.pay\_months);

END IF;

END LOOP;

CLOSE cur\_pledge;

END;

3.12

DECLARE

TYPE type\_pledge IS RECORD(

pledge\_id dd\_pledge.idpledge%TYPE,

donor\_id dd\_pledge.iddonor%TYPE,

pledge\_amount dd\_pledge.pledgeamt%TYPE,

total\_paid NUMBER(5),

need\_to\_pay NUMBER(5));

rec\_pledges type\_pledge;

lv\_pledge\_id dd\_pledge.idpledge%TYPE;

BEGIN

lv\_pledge\_id:=107;

SELECT d1.idpledge, d1.iddonor, d1.pledgeamt, total\_paid, (pledgeamt-total\_paid) need\_pay

INTO rec\_pledges

FROM dd\_pledge d1 INNER JOIN (SELECT SUM(payamt) total\_paid, idpledge

FROM dd\_payment

GROUP BY idpledge) d2

ON d1.idpledge = d2.idpledge

WHERE d1.idpledge = lv\_pledge\_id;

DBMS\_OUTPUT.PUT\_LINE('PLEDGE ID:'|| rec\_pledges.pledge\_id

||' DONOR ID:'||rec\_pledges.donor\_id||' PLEDGE AMOUNT:'||rec\_pledges.pledge\_amount

||' Total paid:'||rec\_pledges.total\_paid||' NEED TO PAY:'|| rec\_pledges.need\_to\_pay );

END;

3.13

DECLARE

lv\_project\_id NUMBER(3);

lv\_project\_name VARCHAR(70);

lv\_start DATE;

lv\_end DATE;

lv\_prev\_fund\_goal NUMBER(6);

lv\_updated\_fund\_goal NUMBER(6);

BEGIN

lv\_project\_id:= 501;

lv\_updated\_fund\_goal :=78000;

SELECT projname, projstartdate, projenddate, projfundgoal

INTO lv\_project\_name, lv\_start, lv\_end, lv\_prev\_fund\_goal

FROM dd\_project

WHERE idproj = lv\_project\_id;

UPDATE dd\_project

SET projfundgoal = lv\_updated\_fund\_goal

WHERE idproj = lv\_project\_id;

DBMS\_OUTPUT.PUT\_LINE('ID:'|| lv\_project\_id ||' NAME:'|| lv\_project\_name

||' START DATE:'|| lv\_start ||' END DATE:'|| lv\_end

||' PREV FUND:'|| lv\_prev\_fund\_goal ||' UPDATED FUND:'||lv\_updated\_fund\_goal);

END;