hands on 3

Before you start working on each chapter's exercises, you need to run a script to rebuild the database(s). This step ensures that the database(s) is in the correct state. Use the script above to rebuild the DoGood Donor database (DD_create). Use the DoGood Donor database for this assignment.

Instructions: Write appropriate PL/SQL programs to complete Hands-On Assignments 3.9-3.11 (Part II) for Chapter 3. For each program:

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 pldeges mads, total dollars pledged, and the average pledge amount.

```
declare
    lv_projid_num dd_project.idproj%TYPE := 501;
    lv_pledgesum_num dd_pledge.pledgeamt%TYPE;
    lv_pledgeavg_num dd_pledge.pledgeamt%TYPE;
    lv_pledgecount_num NUMBER(8);
    lv_projname_txt dd_project.projname%TYPE;

begin

select count(*), sum(pledgeamt), avg(pledgeamt)
    into lv_pledgecount_num, lv_pledgesum_num, lv_pledgeavg_num
    from dd_pledge join dd_project using (idproj)
    where idproj = lv_projid_num
    group by idproj;

select projname
    into lv_projname_txt
    from  dd_project
    where idproj = lv_projid_num
    ;

dbms_output.put_line( 'project name: ' || lv_projname_txt );
dbms_output.put_line( 'pledge count: ' || lv_pledgecount_num );
dbms_output.put_line( 'pledge average: ' || lv_pledgeavg_num );
dbms_output.put_line( 'pledge total for project: ' || lv_pledgesum_num );
end;
/
```

```
SQL> declare
    lv_projid_num dd_project.idproj%TYPE := 501;
    lv_pledgesum_num dd_pledge.pledgeamt%TYPE;
    lv_pledgeavg_num dd_pledge.pledgeamt%TYPE;
    lv_pledgecount_num NUMBER(8);
    lv_projname_txt dd_project.projname%TYPE;

begin

select count(*), sum(pledgeamt), avg(pledgeamt)
    into lv_pledgecount_num, lv_pledgesum_num, lv_pledgeavg_num
    from dd_pledge join dd_project using (idproj)
    where idproj = lv_projid_num
    group by idproj;

select projname
    into lv_projname_txt
    from  dd_project
    where idproj = lv_projid_num
    ;

dbms_output.put_line( 'project name: ' || lv_projname_txt );
dbms_output.put_line( 'pledge count: ' || lv_pledgecount_num );
dbms_output.put_line( 'pledge average: ' || lv_pledgeavg_num );
dbms_output.put_line( 'pledge total for project: ' || lv_pledgesum_num );
end;
/  2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
  19   20   21   22   23   24   25   26   27
project name: Community food pantry #21 freezer equipment
pledge count: 5
pledge average: 812
pledge total for project: 4060

PL/SQL procedure successfully completed.

SQL> michael lundquist 00737340
```

3.10

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 unnown

```
--DOESN'T WORK IN THE BLOCK
CREATE SEQUENCE DD_PROJID_SEQ
MINVALUE 530
START WITH 530
INCREMENT BY 1;

declare
    first_proj_var dd_project%ROWTYPE;
begin

    select DD_PROJID_SEQ.NEXTVAL
    into first_proj_var.idproj
    FROM DUAL;

    first_proj_var.projname := 'HK Animal Shelter Extension';
```

```
    first_proj_var.PROJSTARTDATE := '1-JAN-2013';
    first_proj_var.PROJENDDATE := '31-MAY-2013';
    first_proj_var.projfundgoal := 65000;

    INSERT INTO dd_project
    VALUES first_proj_var;
end;
/
```

```
SQL> declare
    first_proj_var dd_project%ROWTYPE;
begin

    select DD_PROJID_SEQ.NEXTVAL
    into first_proj_var.idproj
    FROM DUAL;

    --first_proj_var.lv_projid_num := 530;

    first_proj_var.projname := 'HK Animal Shelter Extension';
    first_proj_var.PROJSTARTDATE := '1-JAN-2013';
    first_proj_var.PROJENDDATE := '31-MAY-2013';
    first_proj_var.projfundgoal := 65000;

    INSERT INTO dd_project
    VALUES first_proj_var;


end;
/  2     3     4     5     6     7     8     9    10    11    12    13
  19    20    21

PL/SQL procedure successfully completed.

SQL> michael lundquist 00737340
```

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 - #"
- The list should be sorted to display all lump sum pledges first.

```
declare
    --using a table of records
    TYPE type_pledge IS TABLE OF dd_pledge%ROWTYPE
    index by binary_integer;
    lv_pledges_var type_pledge;
    lv_month_date number(2) := 10; -- a month to select pledges from
    lv_paymentmonths_txt varchar2(30) := 'one lump Sum.';
begin
```

```
    --getting info
    select * BULK COLLECT
    into lv_pledges_var
    from dd_pledge
    where extract(month from pledgedate) = lv_month_date;

    --displaying it
    FOR i in lv_pledges_var.first..lv_pledges_var.last loop
        if lv_pledges_var(i).paymonths > 0
            then lv_paymentmonths_txt := 'Monthly - ' ||
lv_pledges_var(i).paymonths;
        end if;
        dbms_output.put_line( 'pledge ID: ' || lv_pledges_var(i).IDPLEDGE || '
donor id ' || lv_pledges_var(i).IDDONOR || ' pledge amount ' ||
lv_pledges_var(i).pledgeamt || ' paid in ' || lv_paymentmonths_txt);
    end loop;
end;
/
```

```
SQL> declare
    --using a table of records
    TYPE type_pledge IS TABLE OF dd_pledge%ROWTYPE
    index by binary_integer;
    lv_pledges_var type_pledge;
    lv_month_date number(2) := 10; -- a month to select pledges from
    lv_paymentmonths_txt varchar2(30) := 'one lump Sum.';
begin
    --getting info
    select * BULK COLLECT
    into lv_pledges_var
    from dd_pledge
    where extract(month from pledgedate) = lv_month_date;

    --displaying it
    FOR i in lv_pledges_var.first..lv_pledges_var.last loop
        if lv_pledges_var(i).paymonths > 0
            then lv_paymentmonths_txt := 'Monthly - ' || lv_pledges_var(i).paymonths;
        end if;
        dbms_output.put_line( 'pledge ID: ' || lv_pledges_var(i).IDPLEDGE || ' donor
id ' || lv_pledges_var(i).IDDONOR || ' pledge amount ' || lv_pledges_var(i).pledgeamt
 || ' paid in ' || lv_paymentmonths_txt);
    end loop;
end;
/  2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
  19   20   21   22   23
pledge ID: 102 donor id 310 pledge amount 500 paid in one lump Sum.
pledge ID: 103 donor id 307 pledge amount 2000 paid in one lump Sum.
pledge ID: 104 donor id 308 pledge amount 240 paid in Monthly - 12
pledge ID: 105 donor id 309 pledge amount 120 paid in Monthly - 12
pledge ID: 106 donor id 301 pledge amount 75 paid in Monthly - 12
pledge ID: 107 donor id 302 pledge amount 1200 paid in Monthly - 24

PL/SQL procedure successfully completed.

SQL> michael lundquist 00737340
```