

The screenshot shows a Windows desktop environment. In the background, a Firefox browser window is open to a page titled "Notes for CIS50" at the URL [www.pgrocer.net/Cis50/cis50notes.html](http://www.pgrocer.net/Cis50/cis50notes.html). The page content includes a table of contents with links to "Introduction to views", "Introduction to reports", "Indexes in Oracle - An", "More on Primary and F", "Notes on key questions", and "More on views".

In the foreground, a Notepad window titled "CIS150\_ASSIGN\_10 - Notepad" is open, displaying the following PL/SQL code:

```

SET SERVEROUTPUT ON

DECLARE
    v_dept          inven_dept.dept%TYPE;
    v_deptname      inven_dept.deptname%TYPE;
    v_onhand        inven.onhand%TYPE;
    v_tot_onhand    onhand_info.tot_onhand%TYPE;
CURSOR inven_dept_cursor IS
    SELECT dept, deptname
    FROM inven_dept
    ORDER BY dept;
CURSOR inven_cursor IS
    SELECT onhand
    FROM inven
    WHERE v_dept = dept
    ORDER BY dept;
BEGIN
    OPEN inven_dept_cursor;
    FETCH inven_dept_cursor INTO v_dept, v_deptname;
    WHILE inven_dept_cursor%FOUND LOOP
        IF inven_cursor%ISOPEN THEN
            CLOSE inven_cursor;
        END IF;
        OPEN inven_cursor;
        v_tot_onhand := 0;
        FETCH inven_cursor INTO v_onhand;
        WHILE inven_cursor%FOUND LOOP
            v_tot_onhand := v_tot_onhand + v_onhand;
            dbms_output.put_line('The current amount for department ' ||
v_deptname || ' is ' || v_tot_onhand );
            FETCH inven_cursor INTO v_onhand;
        END LOOP;
        INSERT INTO onhand_info
            VALUES(v_dept, v_deptname, v_tot_onhand);
        CLOSE inven_cursor;
        FETCH inven_dept_cursor INTO v_dept, v_deptname;
    END LOOP;
    CLOSE inven_dept_cursor;
END;
```

A blue annotation is written in the Notepad window:

Somehow managed to not find my version on my disk so Joe let me use his. I will also include mine.

The Windows taskbar at the bottom shows the system tray with the date and time: 9:57 AM, 11/19/2019. The taskbar also includes icons for the Start menu, search, and several application windows.

```

SET SERVEROUTPUT ON
DECLARE
  v_dept          department.dept%TYPE;
  v_deptname      department.deptname%TYPE;
  v_onhand         inven.onhand%TYPE;
  v_tot_onhand    partinven.onhandtot%TYPE;
  CURSOR department_cursor IS
    SELECT dept, deptname FROM department
    ORDER BY dept;
  CURSOR inven_cursor IS
    SELECT onhand FROM inven
    WHERE v_dept = dept
    ORDER BY dept;
BEGIN
  OPEN department_cursor;
  FETCH department_cursor INTO v_dept, v_deptname;
  WHILE department_cursor%FOUND LOOP
    IF inven_cursor%ISOPEN THEN
      CLOSE inven_cursor;
    END IF;
    OPEN inven_sursor;
    v_tot_onhand := 0;
    FETCH inven_cursor INTO v_onhand;
    WHILE inven_cursor%FOUND LOOP
      v_tot_onhand := v_tot_onhand + v_onhand;
      dbms_output.put_line('The total onhand is: '||v_tot_onhand);
      FETCH inven_cursor INTO v_onhand;
    END LOOP;
    INSERT into partinven
      VALUES (v_dept, v_deptname, v_tot_onhand);
    CLOSE inven_cursor;
    FETCH department_cursor into v_dept, v_deptname;
  END LOOP;
  CLOSE department_cursor;
END;
/
SET SERVEROUTPUT OFF

```

This is a version I wrote. On the next slide, there is another example with LOOP.

```

SET SERVEROUTPUT ON
DECLARE
  v_dept          department.dept%TYPE;
  v_deptname      department.deptname%TYPE;
  v_onhand        inven.onhand%TYPE;
  v_tot_onhand    partinven.onhandtot%TYPE;
  CURSOR department_cursor IS
    SELECT dept, deptname FROM department
    ORDER BY dept;
  CURSOR inven_cursor IS
    SELECT onhand FROM inven
    WHERE v_dept = dept
    ORDER BY dept;
BEGIN
  OPEN department_cursor;
  LOOP
    FETCH department_cursor INTO v_dept, v_deptname;
    EXIT WHEN department_cursor%NOTFOUND;
    IF inven_cursor%ISOPEN THEN
      CLOSE inven_cursor;
    END IF;
    OPEN inven_cursor;
    v_tot_onhand := 0;
    LOOP
      FETCH inven_cursor INTO v_onhand;
      EXIT WHEN inven_cursor%NOTFOUND;
      v_tot_onhand := v_tot_onhand + v_onhand;
      dbms_output.put_line('The total onhand is: '||v_tot_onhand);
    END LOOP;
    INSERT into partinven
      VALUES (v_dept, v_deptname, v_tot_onhand);
    CLOSE inven_cursor;
  END LOOP;
  CLOSE department_cursor;
END;
/
SET SERVEROUTPUT OFF

```

Better if altered to EXIT at the bottom - I modeled this on one of the original cursor examples.

SMART Ink

Continuation of first cursor exampl X +

www.pgrocer.net/Cis50/curfscnt.html

### SQL CODE:

```
SET SERVEROUTPUT ON
DECLARE
  v_drive_no      drive.driveno%TYPE;
  v_drive_name    drive.drivename%TYPE;
  v_contamt       donation.contamt%TYPE;
  v_tot_contamt   cont_info.contamt%TYPE;
  CURSOR drive_cursor IS
    SELECT driveno, drivename FROM drive
    ORDER BY driveno;
  CURSOR donation_cursor IS
    SELECT contamt FROM donation
    WHERE v_drive_no = driveno
    ORDER BY driveno;
BEGIN
  OPEN drive_cursor;
  FETCH drive_cursor INTO v_drive_no, v_drive_name;
  WHILE drive_cursor%FOUND LOOP
    IF donation_cursor%ISOPEN THEN
      CLOSE donation_cursor;
    END IF;
    OPEN donation_cursor;
    v_tot_contamt := 0;
    FETCH donation_cursor INTO v_contamt;
    WHILE donation_cursor%FOUND LOOP
      v_tot_contamt := v_tot_contamt + v_contamt;
      dbms_output.put_line('The current amount is: '||v_tot_contamt);
      FETCH donation_cursor INTO v_contamt;
    END LOOP;
    INSERT into cont_info
      VALUES(v_drive_no, v_drive_name, v_tot_contamt);
    CLOSE donation_cursor;
    FETCH drive_cursor INTO v_drive_no, v_drive_name;
  END LOOP;
  CLOSE drive_cursor;
END;
/
SET SERVEROUTPUT OFF
```

This is the model that I used to write the example we just looked at.

### SQL CODE:

```
SQL> @ cursor7b
The current amount is: 25
The current amount is: 45
The current amount is: 55
The current amount is: 105
The current amount is: 40
The current amount is: 75
The current amount is: 10
The current amount is: 20

PL/SQL procedure successfully completed.

Input truncated to 20 characters
SQL> SELECT * FROM cont_info;
```

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox... X

Type here to search

10:07 AM  
11/19/2019

File Edit View History Bookmarks Tools Help SMART Ink

Introduction to Functions x +

www.pgrocer.net/Cis50/functions.html

## Introduction to Functions:

A function like a procedure receives arguments from the calling program. The difference is that a function is part of an expression and returns a single value to the calling program for its use.

In the example below, I have a function that takes an idno from the calling program and then does a select of that record from the table new\_donation\_one (notice that new\_donation\_one is a modified version of the donation file that only has 1 record for each idno). The select puts the contamt from the table into the variable v\_contamt and then checks to see if it is > 10. Depending on the answer there is a different calculation to determine the variable v\_return\_donation. At the end of the IF and the ELSE, the RETURN verb is used to return v\_return\_donation. I could also have removed the RETURN and done it once after the END IF. When the function is complete, v\_return\_donation goes back to the PL/SQL block that called the function.

Now look at the calling block calc\_func\_calc. It calls the function with the assignment statement that assigns the results of the function to the variable v\_new\_goal. The function call passes the identification for use in the function. The function returns a value that is essentially inserted on the right side of the assignment sign and thereby assigned to v\_new\_goal. All I did was display v\_new\_goal, but I could have used it to update the table, create a new record in another table etc. I executed the PL/SQL calc\_func\_calc three times and the output is shown below. Note: As with procedures, I need to execute the function code first and have the function created before I execute the code that calls the function.

**SQL CODE:**

```
SQL> edit func_calcl
```

**PL/SQL CODE:**

```
CREATE OR REPLACE FUNCTION func_calcl
(v_idno          new_donation_one.idno%TYPE)
RETURN NUMBER IS
  v_contamt      new_donation_one.contamt%TYPE;
  v_return_donation new_donation_one.contamt%TYPE;
BEGIN
  SELECT contamt INTO v_contamt
  FROM new_donation_one
  WHERE idno = v_idno;
  IF v_contamt > 10 THEN
    v_return_donation := v_contamt * 2;
    RETURN v_return_donation;
  ELSE
    v_return_donation := v_contamt * 1.5;
    RETURN v_return_donation;
  END IF;
END func_calcl;
/
```

The function I wrote receives the idno from the main program and uses that to select information from the file and process it. The RETURN returns the result of a calculation to the main pprogram.

**SQL CODE:**

```
SQL> edit calc_func_calc
```

**PL/SQL CODE:**

```
SET SERVEROUTPUT ON
DECLARE
  v_idno          new_donation_one.idno%TYPE := '&input_idno';
  v_new_goal      new_donation_one.contamt%TYPE;
BEGIN
```

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back! Refresh Firefox...

Type here to search

10:08 AM 11/19/2019

File Edit View History Bookmarks Tools Help SMART Ink

Introduction to Functions x +

www.pgrocer.net/Cis50/functions.html

SQL> edit func\_calcl

**PL/SQL CODE:**

```
CREATE OR REPLACE FUNCTION func_calcl
(v_idno          new_donation_one.idno%TYPE)
RETURN NUMBER IS
  v_contamt      new_donation_one.contamt%TYPE;
  v_return_donation new_donation_one.contamt%TYPE;
BEGIN
  SELECT contamt INTO v_contamt
  FROM new_donation_one
  WHERE idno = v_idno;
  IF v_contamt > 10 THEN
    v_return_donation := v_contamt * 2;
    RETURN v_return_donation;
  ELSE
    v_return_donation := v_contamt * 1.5;
    RETURN v_return_donation;
  END IF;
END func_calcl;
/
```

**SQL CODE:**

SQL> edit calc\_func\_calc

**PL/SQL CODE:**

```
SET SERVEROUTPUT ON
DECLARE
  v_idno          new_donation_one.idno%TYPE := '&input_idno';
  v_new_goal      new_donation_one.contamt%TYPE;
BEGIN
  v_new_goal := func_calcl(v_idno);
  dbms_output.put_line('New amount: ' || TO_CHAR(v_new_goal));
END;
/
SET SERVEROUTPUT OFF
```

**SQL CODE:**

```
SQL> select * from new_donation_one order by idno;
```

IDNO	DRI	CONTDAT	CONTAMT
11111	100	07-JAN-99	25
12121	300	10-JUN-99	75
22222	100	14-MAR-99	10
23456	300	14-JUN-99	10
33333	300	10-MAR-99	10

SQL> @ func\_calcl

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox... X

Type here to search

10:14 AM 11/19/2019

Here I call the function and send v\_idno. It is received and used to get a particular record. When the function is complete the information it returned is stored in v\_new\_goal.

Note the structure that I used with the call on the right and the field to receive the return on the left.

File Edit View History Bookmarks Tools Help

SMART Ink

Introduction to Functions

www.pgrocer.net/Cis50/functions.html

This is the example where the RETURN was moved to after the IF so that it only appeared once in the code. Note that I had to make the change to call a different function in calc\_func\_calc. Note also that I have to run the function to create it before I can run the code that calls the function.

**SQL CODE:**

```
SQL> edit func_calc
```

**PL/SQL CODE:**

```
CREATE OR REPLACE FUNCTION func_calc
(v_idno          new_donation_one.idno%TYPE)
RETURN NUMBER IS
  v_contamt      new_donation_one.contamt%TYPE;
  v_return_donation new_donation_one.contamt%TYPE;
BEGIN
  SELECT contamt INTO v_contamt
  FROM new_donation_one
  WHERE idno = v_idno;
  IF v_contamt > 10 THEN
    v_return_donation := v_contamt * 2;
  ELSE
    v_return_donation := v_contamt * 1.5;
  END IF;
  RETURN v_return_donation;
END func_calc;
/
```

**SQL CODE:**

```
SQL> edit calc_func_calc
```

**PL/SQL CODE:**

```
SET SERVEROUTPUT ON
DECLARE
  v_idno          new_donation_one.idno%TYPE := '&input_idno';
  v_new_goal      new_donation_one.contamt%TYPE;
BEGIN
  v_new_goal := func_calc(v_idno);
  dbms_output.put_line('New amount: '||TO_CHAR(v_new_goal));
END;
/
SET SERVEROUTPUT OFF
```

**SQL CODE:**

```
SQL> @ func_calc
Input truncated to 1 characters

Function created.

SQL> @ calc_func_calc
```

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox...

Type here to search

10:20 AM  
11/19/2019

Introduction to Functions x +

www.pgrocer.net/Cis50/functions.html

SET SERVEROUTPUT OFF

**SQL CODE:**

```
SQL> select * from new_donation_one order by idno;
```

IDNO	DRI	CONTDATE	CONTAMT
11111	100	07-JAN-99	25
12121	300	10-JUN-99	75
22222	100	14-MAR-99	10
23456	300	14-JUN-99	10
33333	300	10-MAR-99	10

Notice that new\_donation\_one has only one donation her idno. If multiple can be returned, I would need to set up a cursor to deal with them.

```
SQL> @ func_calci
Input truncated to 1 characters

Function created.

SQL> @ calc_func_calc
Enter value for input_idno: 12121
old 2:  v_idno          new_donation_one.idno%TYPE := '&input_idno';
new 2:  v_idno          new_donation_one.idno%TYPE := '12121';
New amount: 150

PL/SQL procedure successfully completed.

SQL> @ calc_func_calc
Enter value for input_idno: 22222
old 2:  v_idno          new_donation_one.idno%TYPE := '&input_idno';
new 2:  v_idno          new_donation_one.idno%TYPE := '22222';
New amount: 15

PL/SQL procedure successfully completed.

SQL> edit calc_func_calc

SQL> @ calc_func_calc
Enter value for input_idno: 11111
old 2:  v_idno          new_donation_one.idno%TYPE := '&input_idno';
new 2:  v_idno          new_donation_one.idno%TYPE := '11111';
New amount: 50
```

This is the example where the RETURN was moved to after the IF so that it only appeared once in the code. Note that I had to make the change to call a different function in calc\_func\_calc. Note also that I have to run the function to create it before I can run the code that calls the function.

**SQL CODE:**

```
SQL> edit func_calc
```

**PL/SQL CODE:**

```
CREATE OR REPLACE FUNCTION func_calc
```

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Refresh Firefox... X

Type here to search

10:23 AM 11/19/2019

File Edit View History Bookmarks Tools Help SMART Ink

More on procedures x +

www.pgrocer.net/Cis50/moreproc.html

In the next example, I decided to take the input contribution and up in by 10% and add that to the file - in other words I guess I "doctored/cooked" the books. To do this, I did the calculation in the procedure. First I added a variable to the block (outside the procedure) called v\_newcontamt and then within the processing part of the block, I calculated v\_newcontamt by taking the result of the pass to p\_contamt and multiplying it by 1.1. I then changed the values in the output to output this new amount instead of the content of p\_contamt. The results are shown below.

**SQL CODE:**

```
SQL> edit Call_Adddonproc2
```

**PL/SQL CODE:**

```
SET VERIFY OFF
DECLARE

v_idno      new_donation.idno%TYPE := '&input_idno';
v_driveno   new_donation.driveno%TYPE := '&input_driveno';
v_contamt   new_donation.contamt%TYPE := '&input_contamt';
v_newcontamt new_donation.contamt%TYPE;

PROCEDURE AddDonProc
(p_idno      new_donation.idno%TYPE,
 p_driveno   new_donation.driveno%TYPE,
 p_contamt   new_donation.contamt%TYPE) AS
BEGIN
  v_newcontamt := p_contamt * 1.1;
  INSERT INTO new_donation(idno, driveno, contamt)
    VALUES(p_idno, p_driveno, v_newcontamt);
END AddDonProc;

BEGIN
  IF v_contamt > 20 THEN
    AddDonProc (v_idno, v_driveno, v_contamt);
  END IF;
END;
/
SET VERIFY ON
```

**SQL CODE:**

```
SQL> @ Call_Adddonproc2
Enter value for input_idno: 11111
Enter value for input_driveno: 100
Enter value for input_contamt: 120

PL/SQL procedure successfully completed.
SQL> SELECT * FROM new_donation;
```

IDNO	DRI	CONTDATE	CONTAMT
11111	100	07-JAN-99	25
12121	200	23-FEB-99	40
23456	100	03-MAR-99	20
33333	300	10-MAR-99	10
22222	100	14-MAR-99	10
12121	100	04-JUN-99	50
11111	200	12-JUN-99	35
23456	300	14-JUN-99	10

Now I am back to Procedures and I am embedding the procedure in the main program.

Type here to search

10:37 AM 11/19/2019

File Edit View History Bookmarks Tools Help SMART Ink

More on procedures x +

www.pgrocet.net/Cis50/moreprochtml

```

SET VERIFY OFF
DECLARE

  v_idno      new_donation.idno%TYPE := '&input_idno';
  v_driveno   new_donation.driveno%TYPE := '&input_driveno';
  v_contamt   new_donation.contamt%TYPE := '&input_contamt';
  v_newcontamt new_donation.contamt%TYPE;

PROCEDURE AddDonProc
  (p_idno      new_donation.idno%TYPE,
   p_driveno   new_donation.driveno%TYPE,
   p_contamt   new_donation.contamt%TYPE) AS
BEGIN
  INSERT INTO new_donation(idno, driveno, contamt)
    VALUES (p_idno, p_driveno, v_newcontamt);
END AddDonProc;

BEGIN
  IF v_contamt > 20 THEN
    v_newcontamt := v_contamt * 1.1;
    AddDonProc (v_idno, v_driveno, v_newcontamt);
  END IF;
END;
/
SET VERIFY ON

```

**SQL CODE:**

```

SQL> @ Call_Adddonproc3
Enter value for input_idno: 33333
Enter value for input_driveno: 300
Enter value for input_contamt: 300

PL/SQL procedure successfully completed.

Input truncated to 13 characters
SQL> SELECT * FROM new_donation;

IDNO  DRI  CONTDATE  CONTAMT
-----
11111 100 07-JAN-99      25
12121 200 23-FEB-99      40
23456 100 03-MAR-99      20
33333 300 10-MAR-99      10
22222 100 14-MAR-99      10
12121 100 04-JUN-99      50
11111 200 12-JUN-99      35
23456 300 14-JUN-99      10
12121 300 10-JUN-99      75
12121 100                500
11111 100                7777
12121 100                100
11111 100                132
33333 300                330

14 rows selected.

SQL> edit Call_Adddonproc4

```

Windows taskbar: Type here to search, 10:39 AM 11/19/2019

File Edit View History Bookmarks Tools Help SMART Ink

More on procedures x +

www.pgrocet.net/Cis50/moreprochtml

12121	200	23-FEB-99	40
23456	100	03-MAR-99	20
33333	300	10-MAR-99	10
22222	100	14-MAR-99	10
12121	100	04-JUN-99	50
11111	200	12-JUN-99	35
23456	300	14-JUN-99	10
12121	300	10-JUN-99	75
12121	100		500
11111	100		7777
12121	100		100
11111	100		132
33333	300		330
12121	200		247.5

15 rows selected.

SQL> edit Call\_Adddonproc5

**PL/SQL CODE:**

```
SET VERIFY OFF
DECLARE
  v_idno      new_donation.idno%TYPE := '&input_idno';
  v_driveno   new_donation.driveno%TYPE := '&input_driveno';
  v_contamt   new_donation.contamt%TYPE := '&input_contamt';
  v_newcontamt new_donation.contamt%TYPE;

PROCEDURE AddDonProc
  (p_idno      new_donation.idno%TYPE,
  p_driveno    new_donation.driveno%TYPE) AS
BEGIN
  INSERT INTO new_donation(idno, driveno, contamt)
  VALUES (p_idno, p_driveno, v_newcontamt);
END AddDonProc;

BEGIN
  IF v_contamt > 20 THEN
    v_newcontamt := v_contamt * 1.1;
    AddDonProc (v_idno, v_driveno);
  END IF;
END;
/
SET VERIFY ON
```

**SQL CODE:**

```
SQL> @ Call_Adddonproc5
Enter value for input_idno: 11111
Enter value for input_driveno: 200
Enter value for input_contamt: 500

PL/SQL procedure successfully completed.

Input truncated to 13 characters
SQL> SELECT * FROM new_donation;
```

IDNO DRV CONTDATE CONTAMT

Type here to search

10:40 AM 11/19/2019