

## PL/SQL –Procedures and Functions

Apex Instructions for this tutorial. Type in all code in the SQL Commands environment. Save each task separately.

### Task 1

First create tables book and book\_copy, the code is specified below:

- Books – holds a list of all the different books in the library
- Books\_copies – a list of all the physical books (so there may be three copies of a specific book).

```
drop table books cascade constraints;
drop table book_copies cascade constraints;

create table books(
isbn varchar2(13) primary key,
title varchar2(200),
summary varchar2(2000),
author varchar2(40),
date_published date,
page_count number(10));

create table book_copies(
barcode_id varchar2(100) primary key,
isbn varchar2(13) not null,
constraint book_copies_isbn_fk foreign key (isbn) references books (isbn));
```

Currently you would need to insert data into a table using INSERT statement. Type in this statement.

Note: The first printing of this book was written with the following formatting of this INSERT statement:

```
INSERT INTO books (isbn, title, author) VALUES ('0-596-00180-0', 'Learning Oracle PL/SQL', 'Bill Pribyl with Steven Feuerstein');
```

Now, select data from the table:

```
SELECT author FROM books WHERE isbn = '0-596-00180-0';
```

Let's delete the book to prevent any errors from attempting to insert a duplicate.

```
/* Let's delete the book to prevent any errors from attempting to insert a duplicate. */  
  
DELETE books WHERE isbn = '0-596-00180-0';  
  
/* ...and re-execute the statement as follows: */  
  
INSERT INTO books (isbn, title, author)  
VALUES ('0-596-00180-0', 'Learning Oracle PL/SQL',  
      'Bill Pribyl with Steven Feuerstein');  
  
/* Now examine the data: */  
  
SELECT author FROM books WHERE isbn = '0-596-00180-0';  
  
/* That's better! */
```

## PL/SQL –Procedures and Testing

### Task 2 – create a procedure to insert the book (add\_book)

Note this procedure doesn't do more than just INSERT a book into the table 'books'. It also logs the individual copies of the book. Save the file as *add\_book\_proc2*.

```
CREATE OR REPLACE PROCEDURE add_book (isbn_in IN VARCHAR2,
    barcode_id_in IN VARCHAR2, title_in IN VARCHAR2, author_in IN VARCHAR2,
    page_count_in IN NUMBER, summary_in IN VARCHAR2 DEFAULT NULL,
    date_published_in IN DATE DEFAULT NULL)
AS
BEGIN
    /* check for reasonable inputs */

    IF isbn_in IS NULL
    THEN
        RAISE VALUE_ERROR;
    END IF;

    /* put a record in the "books" table */

    INSERT INTO books (isbn, title, summary, author, date_published, page_count)
    VALUES (isbn_in, title_in, summary_in, author_in, date_published_in,
        page_count_in);

    /* put a record in the "book_copies" table */

    IF barcode_id_in IS NOT NULL
    THEN
        INSERT INTO book_copies (isbn, barcode_id)
        VALUES (isbn_in, barcode_id_in);
    END IF;
END add_book;
/
```

### Task 3

To test the procedure we'll run an anonymous PL/SQL block. However, first we'll write code to DELETE data in the table, just in case it is already there:

- Save this file as *add\_one\_rec3*

```

/* Note that first time you run this code, no records will be deleted as we do not have this book */
/* First delete this book so this script won't result in an error*/

DELETE book_copies WHERE isbn = '1-56592-335-9';
DELETE books WHERE isbn = '1-56592-335-9';

/* Call add_book procedure with literal values*/
BEGIN
    add_book('1-56592-335-9',
        '100000001',
        'Oracle PL/SQL Programming',
        'Feuerstein, Steven, with Bill Pribyl',
        987,
        'Reference for PL/SQL developers, '
        || 'including examples and best practice recommendations.',
        TO_DATE('01-SEP-1997','DD-MON-YYYY'));
END;
/

/* let's view book table */

SELECT * FROM books;
SELECT * FROM book_copies;

```

The output should be:

SELECT \* FROM books

ISBN	TITLE	SUMMARY	AUTHOR	DATE_PUBLISHED	PAGE_COUNT
1-56592-335-9	Oracle PL/SQL Programming	Reference for PL/SQL developers, including examples and best practice recommendations.	Feuerstein, Steven, with Bill Pribyl	09/01/1997	987
0-596-00180-0	Learning Oracle PL/SQL	-	Bill Pribyl with Steven Feuerstein	-	-

2 rows selected. 0.01 seconds

## Task 4

Note: add\_book procedure (code we created) could be two procedures, i.e. *add\_book* and *add\_book\_copy* (example below). You will need *add\_book\_copy* procedure for next week.

Save this code as a file *add\_book\_copy3*.

```

/* Support utility used by the test_book_copy_qty script; adds record to
book_copies table */

CREATE OR REPLACE PROCEDURE add_book_copy(isbn_in IN VARCHAR2,
    barcode_id_in IN VARCHAR2)
IS
BEGIN
    IF isbn_in IS NOT NULL AND barcode_id_in IS NOT NULL
    THEN
        INSERT INTO book_copies (isbn, barcode_id)
        VALUES (isbn_in, barcode_id_in);
    END IF;
EXCEPTION
    WHEN DUP_VAL_ON_INDEX
    THEN
        NULL;
END;
/

```

In the book (Learning Oracle PL/SQL Oracle Development Languages by Steven Feuerstein, Bill Pribyl), there is extra practice to check whether a book already exists, add an extra copy of a book and more.

#### Task 5: Now your turn, using scott tables, emp and dept, complete the following...

Create and test a procedure that will:

- Add a new employee
- Delete an employee
- Update an Employee

Extra: consider the 'checks' required for each procedure. E.g. check employee doesn't already exist ...

E.g. the procedure add\_new\_employee could

Adds the employee

Add the employee to the 'annual\_leave' table with an allocated holiday allowance.

## PL/SQL – Functions and testing

### Task 6

A function 'return's something, in the SQL statement below we are using build in function **months\_between**.

E.g.

```
SELECT ename, hiredate, floor(months_between(sysdate, hiredate)/12) no_of_years
FROM emp
WHERE months_between(sysdate, hiredate) > 12 * 5;
```

You could write and define your own PL/SQL Function. This is an example of a function that calculated years' person has been hired. Save this code as a file *Function\_testing\_emp5*.

```
CREATE OR REPLACE FUNCTION Integer_years
    (date_start IN DATE,
     date_finish IN DATE)
return number
is
begin
return floor(months_between(date_start,date_finish)/12);
end;
/
```

/\* To test the function, you would write a code like this:\*/

```
SELECT  ename, hiredate, integer_years(sysdate, hiredate)
FROM emp
WHERE integer_years(sysdate, hiredate) >= 5;
```

#### Task 7

Research and list five function that could be used with emp table:

- 1.
- 2.
- 3.
- 4.
- 5.

## PL/SQL Functions

### Task 8 - Let's go back to our book tables and examples.

- Type in code below to create a function to count number of copies of a particular book. Save code as a filename *book\_copy\_qty\_func6*.

```
/* Create function to count number of copies of a particular book */  
  
CREATE OR REPLACE FUNCTION book_copy_qty(isbn_in IN VARCHAR2)  
RETURN NUMBER  
AS  
    number_o_copies NUMBER := 0;  
    CURSOR bc_cur IS  
        SELECT COUNT(*)  
        FROM book_copies  
        WHERE isbn = isbn_in;  
BEGIN  
    IF isbn_in IS NOT NULL  
    THEN  
        OPEN bc_cur;  
        FETCH bc_cur INTO number_o_copies;  
        CLOSE bc_cur;  
    END IF;  
    RETURN number_o_copies;  
END;  
/
```

### Test Function above

- Type in the code bellow: the code will execute your function to return value. Save this code as *call\_book\_copy\_qty7*.

```
/* Example of calling book_copy_qty function */  
  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('Number of copies of 1-56592-335-9: '  
        || book_copy_qty('1-56592-335-9'));  
END;  
/
```

- The output would be:

Number of copies of 1-56592-335-9: 1

### Task 10 - Now your turn:

Using your scott tables, write a function to:

- Return the department for an employee
- Return the number of employees in a department
- Check whether an employee already exists on the employee table (based on ename?)