# More PLSQL Exercises

1. Make a report using an anonymous code block.
   1. Have the report print out each publisher name
   2. For each publisher:
      1. print out the name of the books published, the books’ id, and their price
      2. print out the total price for all the books combined for the publisher.

**Example** output for ONE publisher (you need to show all publishers)

Publisher : New Age Books

----------------------------------

Book : PS7777 Emotional Security: A New Algorithm

Book : PS3333 Prolonged Data Deprivation: Four Case Studies

Book : BU2075 You Can Combat Computer Stress!

Book : PS2091 Is Anger the Enemy?

Book : PS2106 Life Without Fear

Book : PS1372 Computer Phobic and Non-Phobic Individuals: Behavior Variations

TOTAL PRICE FOR ALL BOOKS: 141.51

*DECLARE*

*CURSOR c\_publisher IS*

*SELECT \* FROM publisher ORDER BY pubname;*

*rec\_publisher publisher%rowtype;*

*-- Declaration of department cursor and record variable.*

*CURSOR c\_title (param\_publisher publisher.pubid%TYPE) IS*

*SELECT \* FROM title WHERE pubid = param\_publisher;*

*rec\_title title%ROWTYPE;*

*v\_price title.price%type;*

*-- Declaration of employees cursor and record variable.*

*BEGIN*

*OPEN c\_publisher;*

*LOOP*

*FETCH c\_publisher INTO rec\_publisher;*

*EXIT WHEN c\_publisher%NOTFOUND;*

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

*DBMS\_OUTPUT.PUT\_LINE('Publisher : '||rec\_publisher.pubname);*

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

*OPEN c\_title(rec\_publisher.pubid);*

*LOOP*

*FETCH c\_title INTO rec\_title;*

*EXIT WHEN c\_title%NOTFOUND;*

*DBMS\_OUTPUT.PUT\_LINE('Book : '||rec\_title.titleid*

*||' '||rec\_title.title);*

*END LOOP;*

*CLOSE c\_title;*

*select sum(price) into v\_price from title where pubid=rec\_publisher.pubid;*

*DBMS\_OUTPUT.PUT\_LINE('TOTAL PRICE FOR ALL BOOKS: ' || v\_price );*

*END LOOP;*

*CLOSE c\_publisher;*

*END;*

1. Create a procedure
   1. pass in a first name and a last name of an editor
   2. If the editor already exists print the editor’s id
   3. Otherwise insert the editor into the database.

*/\*NOTE this shows you how to use %notfound. Realize that it is possible that the query used for the cursor could return more than one record and as a result will only ever use the first record returned. Don’t use a cursor for queries that will only return one row at most\*/*

*create or replace*

*Procedure printtEditor(fname\_in IN varchar2, lname\_in IN varchar2)*

*is*

*enumber number;*

*cursor c1 is*

*select edid from editor where edlname = lname\_in and edfname=fname\_in;*

*rec\_1 editor.edid%type;*

*Begin*

*open c1;*

*fetch c1 into rec\_1;*

*if c1%found then*

*dbms\_output.put\_line('Editor id: ' || rec\_1);*

*else*

*enumber := 9999;*

*insert into editor (edid, edlname, edfname) values (enumber, lname\_in, fname\_in);*

*end if;*

*close c1;*

*end;*

***TO USE the procedure(note: you can put this in another function, procedure or trigger too)***

*begin*

*UpdateEditor('James', 'Smith');*

*End;*

1. Create a function
   1. You should pass to the function the first and last names of an editor
   2. If the editor is found, return the editor’s id, otherwise return 8888

*/\*NOTE this shows you how to use %notfound. Realize that it is possible that the query used for the cursor could return more than one record and as a result will only ever use the first record returned. Don’t use a cursor for queries that will only return one row at most\*/*

*CREATE OR REPLACE Function FindEditor*

*( fname\_in IN varchar2, lname\_in IN varchar2 )*

*RETURN number*

*IS*

*enumber number;*

*cursor c1 is*

*select edid*

*from editor*

*edlname = lname\_in and edfname=fname\_in;*

*BEGIN*

*open c1;*

*fetch c1 into enumber;*

*if c1%notfound then*

*enumber := 8888;*

*end if;*

*close c1;*

*RETURN enumber;*

*EXCEPTION*

*WHEN OTHERS THEN*

*raise\_application\_error(-20001,'An error was encountered - '||SQLCODE||' -ERROR- '||SQLERRM);*

*END;*

*/*

*Declare*

*enum number;*

*begin*

*enum := FindEditor('sasdfasf', 'adfsaf');*

*dbms\_output.put\_line('id number: ' || enum);*

*End;*

***OR***

*Declare*

*enum number;*

*begin*

*enum := FindEditor('James', 'Smith');*

*dbms\_output.put\_line('id number: ' || enum);*

*End;*

1. Create a trigger that fires on the title table:
   1. it should fire when:
      1. There is an insert
      2. The price is updated
      3. The title is updated
      4. A row is deleted
   2. Print to the screen what has happened (ie: the price was updated)

*CREATE OR REPLACE TRIGGER TitleTrig*

*BEFORE*

*INSERT OR*

*UPDATE OF price, title OR*

*DELETE*

*ON Title*

*BEGIN*

*CASE*

*WHEN INSERTING THEN*

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

*WHEN UPDATING('price') THEN*

*DBMS\_OUTPUT.PUT\_LINE('Updating price');*

*WHEN UPDATING('title’) THEN*

*DBMS\_OUTPUT.PUT\_LINE('Updating title');*

*WHEN DELETING THEN*

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

*END CASE;*

*END;*

***TEST:***

*insert into title values('TL9999','My book', 'science', '0736',300, 40000,30000,1,'some notes', sysdate);*

*update title set price=250 where titleid='TL9999';*

*update title set title='My second book' where titleid='TL9999';*

*delete from title where titleid='TL9999';*