CIS 229

Professor: Bob Desilets

Date 11/14/2018

Cedric L Mulumba

Chapter 7

**Hands-On Assignments Part I**

|  |
| --- |
| **Assignment 7-1: Creating a Package** |
| CREATE OR REPLACE PACKAGE order\_info\_pkg IS  FUNCTION ship\_name\_pf  (p\_basket IN NUMBER)  RETURN VARCHAR2;  PROCEDURE basket\_info\_pp  (p\_basket IN NUMBER,  p\_shop OUT NUMBER,  p\_date OUT DATE);  END; |
|  |
| CREATE OR REPLACE PACKAGE BODY order\_info\_pkg IS  FUNCTION ship\_name\_pf  (p\_basket IN NUMBER)  RETURN VARCHAR2  IS  lv\_name\_txt VARCHAR2(25);  BEGIN  SELECT shipfirstname||' '||shiplastname  INTO lv\_name\_txt  FROM bb\_basket  WHERE idBasket = p\_basket;  RETURN lv\_name\_txt;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Invalid basket id');  END ship\_name\_pf;  PROCEDURE basket\_info\_pp  (p\_basket IN NUMBER,  p\_shop OUT NUMBER,  p\_date OUT DATE)  IS  BEGIN  SELECT idshopper, dtordered  INTO p\_shop, p\_date  FROM bb\_basket  WHERE idbasket = p\_basket;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Invalid basket id');  END basket\_info\_pp;  END; |
|  |
| **Assignment 7-2: Using Program Units in a Package** |
| CREATE OR REPLACE PACKAGE order\_info\_pkg IS  FUNCTION ship\_name\_pf  (p\_basket IN NUMBER)  RETURN VARCHAR2;  PROCEDURE basket\_info\_pp  (p\_basket IN NUMBER,  p\_shop OUT NUMBER,  p\_date OUT DATE);  END; |
|  |
| CREATE OR REPLACE PACKAGE BODY order\_info\_pkg IS  FUNCTION ship\_name\_pf  (p\_basket IN NUMBER)  RETURN VARCHAR2  IS  lv\_name\_txt VARCHAR2(25);  BEGIN  SELECT shipfirstname||' '||shiplastname  INTO lv\_name\_txt  FROM bb\_basket  WHERE idBasket = p\_basket;  RETURN lv\_name\_txt;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Invalid basket id');  END ship\_name\_pf;  PROCEDURE basket\_info\_pp  (p\_basket IN NUMBER,  p\_shop OUT NUMBER,  p\_date OUT DATE)  IS  BEGIN  SELECT idshopper, dtordered  INTO p\_shop, p\_date  FROM bb\_basket  WHERE idbasket = p\_basket;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Invalid basket id');  END basket\_info\_pp;  END; |
|  |
| DECLARE  lv\_id number := 12;  lv\_name varchar2(25);  lv\_shopper bb\_basket.idshopper%type;  lv\_date bb\_basket.dtcreated%type;    BEGIN  -- test function  lv\_name := order\_info\_pkg.ship\_name\_pf(lv\_id);  dbms\_output.put\_line(lv\_id||' '||lv\_name);    -- test procedure  order\_info\_pkg.basket\_info\_pp(lv\_id, lv\_shopper, lv\_date);  dbms\_output.put\_line(lv\_id||' '||lv\_shopper||' '||lv\_date);  END; |
|  |
| SELECT lpad(order\_info\_pkg.ship\_name\_pf(idbasket), 20)  "ship\_name\_pf on 12"  FROM bb\_basket  WHERE idbasket = 12; |
|  |
| **Assignment 7-3: Creating a Package with Private Program Units** |
| CREATE OR REPLACE PACKAGE order\_info\_pkg IS  PROCEDURE basket\_info\_pp  (p\_basket IN NUMBER,  p\_shop OUT NUMBER,  p\_date OUT DATE,  p\_ship out varchar);  END |
|  |
| CREATE OR REPLACE PACKAGE BODY order\_info\_pkg IS  FUNCTION ship\_name\_pf  (p\_basket IN NUMBER)  RETURN VARCHAR2  IS  lv\_name\_txt VARCHAR2(25);  BEGIN  SELECT shipfirstname||' '||shiplastname  INTO lv\_name\_txt  FROM bb\_basket  WHERE idBasket = p\_basket;  RETURN lv\_name\_txt;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Invalid basket id');  END ship\_name\_pf;  PROCEDURE basket\_info\_pp  (p\_basket IN NUMBER,  p\_shop OUT NUMBER,  p\_date OUT DATE,  p\_ship out varchar)  IS  BEGIN  SELECT idshopper, dtordered  INTO p\_shop, p\_date  FROM bb\_basket  WHERE idbasket = p\_basket;  p\_ship := ship\_name\_pf(p\_basket);  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Invalid basket id');  END basket\_info\_pp;  END; |
|  |
| DECLARE  lv\_id number := 6;  lv\_name varchar2(25);  lv\_shopper bb\_basket.idshopper%type;  lv\_date bb\_basket.dtcreated%type;    BEGIN  order\_info\_pkg.basket\_info\_pp(lv\_id, lv\_shopper,  lv\_date, lv\_name);  dbms\_output.put\_line(lv\_id||' '||lv\_shopper  ||' '||lv\_date||' '||lv\_name);  END; |
|  |
| **Assignment 7-4: Using Packaged Variable** |
| CREATE OR REPLACE FUNCTION verify\_user  (usernm in varchar2,  passwd in varchar2)  RETURN char  IS  temp\_user bb\_shopper.username%type;  confirm char(1) := 'N';  BEGIN  SELECT username  INTO temp\_user  FROM bb\_shopper  WHERE password = passwd;    confirm := 'Y';  RETURN confirm;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('logon values are invalid');  END; |
|  |
| DECLARE  g\_ck char(1);  BEGIN  g\_ck := verify\_user('gma1', 'goofy');  END; |
|  |
| CREATE OR REPLACE PACKAGE login\_pckg is  FUNCTION verify\_user  (usernm in varchar2,  passwd in varchar2)  RETURN char;  END; |
|  |
| CREATE OR REPLACE PACKAGE BODY login\_pckg IS  FUNCTION verify\_user  (usernm in varchar2,  passwd in varchar2)  return char  is  temp\_user bb\_shopper.username%type;  confirm char(1) := 'N';    BEGIN  SELECT username  INTO temp\_user  FROM bb\_shopper  WHERE password = passwd;    confirm := 'Y';  return confirm;    EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('logon values are invalid');    END verify\_user;  END; |
|  |
| DECLARE  g\_ck char(1);  BEGIN  :g\_ck := login\_pckg.verify\_user('gma1', 'goofy');  DBMS\_OUTPUT.PUT\_LINE(:g\_ck);  END; |
|  |
| **Assignment 7-5: Overloading Packaged Procedure** |
| CREATE OR REPLACE PACKAGE shop\_query\_pkg is    PROCEDURE retrieve\_shopper  (lv\_id in bb\_shopper.idshopper%type,  lv\_name out varchar,  lv\_city out bb\_shopper.city%type,  lv\_state out bb\_shopper.state%type,  lv\_phone out bb\_shopper.phone%type);      PROCEDURE retrieve\_shopper  (lv\_last in bb\_shopper.lastname%type,  lv\_name out varchar,  lv\_city out bb\_shopper.city%type,  lv\_state out bb\_shopper.state%type,  lv\_phone out bb\_shopper.phone%type);  END; |
|  |
| CREATE OR REPLACE PACKAGE BODY shop\_query\_pkg is    PROCEDURE retrieve\_shopper  (lv\_id in bb\_shopper.idshopper%type,  lv\_name out varchar,  lv\_city out bb\_shopper.city%type,  lv\_state out bb\_shopper.state%type,  lv\_phone out bb\_shopper.phone%type)  is    BEGIN  SELECT firstname||' '||lastname, city,  state, phone  INTO lv\_name, lv\_city, lv\_state, lv\_phone  FROM bb\_shopper  WHERE idshopper = lv\_id;    END retrieve\_shopper;      PROCEDURE retrieve\_shopper  (lv\_last in bb\_shopper.lastname%type,  lv\_name out varchar,  lv\_city out bb\_shopper.city%type,  lv\_state out bb\_shopper.state%type,  lv\_phone out bb\_shopper.phone%type)  is  BEGIN  SELECT firstname||' '||lastname, city,  state, phone  INTO lv\_name, lv\_city, lv\_state, lv\_phone  FROM bb\_shopper  WHERE lastname = lv\_last;    END retrieve\_shopper;  END; |
|  |
| DECLARE  lv\_id number := 23;  lv\_last bb\_shopper.lastname%type := 'Ratman';  lv\_name varchar2(25);  lv\_city bb\_shopper.city%type;  lv\_state bb\_shopper.state%type;  lv\_phone bb\_shopper.phone%type;    BEGIN    shop\_query\_pkg.retrieve\_shopper(lv\_id, lv\_name,  lv\_city, lv\_state, lv\_phone);  DBMS\_OUTPUT.PUT\_LINE(lv\_name||' '||lv\_city  ||' '||lv\_state||' '||lv\_phone);  shop\_query\_pkg.retrieve\_shopper(lv\_last, lv\_name,  lv\_city, lv\_state, lv\_phone);  DBMS\_OUTPUT.PUT\_LINE(lv\_name||' '||lv\_city  ||' '||lv\_state||' '||lv\_phone);  END; |
|  |
| **Assignment 7-6: Creating a Package with Only a Specification** |
| CREATE OR REPLACE PACKAGE tax\_rate\_pkg is  pv\_tax\_nc constant number := .035; -- all variables are constants  pv\_tax\_tx constant number := .05;  pv\_tax\_tn constant number := .02;  END; |
|  |
| BEGIN  dbms\_output.put\_line(tax\_rate\_pkg.pv\_tax\_nc);  dbms\_output.put\_line(tax\_rate\_pkg.pv\_tax\_tx);  dbms\_output.put\_line(tax\_rate\_pkg.pv\_tax\_tn);  END; |
|  |
| **Assignment 7-7: Using a Cursor in a Package** |
| CREATE OR REPLACE PACKAGE tax\_rate\_pkg is    CURSOR cur\_tax is  SELECT taxrate, state  FROM bb\_tax;    FUNCTION get\_tax  (pv\_state in bb\_tax.state%type)  RETURN bb\_tax.taxrate%type;  END; |
|  |
| CREATE OR REPLACE PACKAGE BODY tax\_rate\_pkg is    FUNCTION get\_tax  (pv\_state in bb\_tax.state%type)  RETURN bb\_tax.taxrate%type  IS    pv\_tax bb\_tax.taxrate%type := 0.00;    BEGIN  FOR rec\_tax in cur\_tax LOOP  IF rec\_tax.state = pv\_state THEN  pv\_tax := rec\_tax.taxrate;    END IF;  END LOOP;      RETURN pv\_tax;    END get\_tax;  END; |
|  |
| BEGIN  DBMS\_OUTPUT.PUT\_LINE('NC'||' '||tax\_rate\_pkg.get\_tax('NC'));  END; |
|  |
| **Assignment 7-8: Using a One-Time-Only Procedure in a Package** |
| CREATE OR REPLACE PACKAGE login\_pkg IS    pv\_login\_time timestamp;    pv\_id\_num NUMBER(3);  FUNCTION login\_ck\_pf  (p\_user IN VARCHAR2,  p\_pass IN VARCHAR2)  RETURN CHAR;    END; |
|  |
| CREATE OR REPLACE PACKAGE BODY login\_pkg IS  FUNCTION login\_ck\_pf  (p\_user IN VARCHAR2,  p\_pass IN VARCHAR2)  RETURN CHAR  IS  lv\_ck\_txt CHAR(1) := 'N';  lv\_id\_num NUMBER(5);    BEGIN  SELECT idShopper  INTO lv\_id\_num  FROM bb\_shopper  WHERE username = p\_user  AND password = p\_pass;    lv\_ck\_txt := 'Y';  pv\_id\_num := lv\_id\_num;  RETURN lv\_ck\_txt;    EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RETURN lv\_ck\_txt;    END login\_ck\_pf;      BEGIN  SELECT systimestamp  INTO pv\_login\_time  FROM dual;    END; |
|  |
| DECLARE    lv\_user bb\_shopper.username%type := 'Crackj';  lv\_passwd bb\_shopper.password%type := 'flyby';  lv\_login char := 'N';    BEGIN  lv\_login := login\_pkg.login\_ck\_pf(lv\_user, lv\_passwd);      DBMS\_OUTPUT.PUT\_LINE(lv\_login||' '||login\_pkg.pv\_login\_time);  END; |
|  |