CIS 229

Professor: Bob Desilets

Date 10/24/2018

Cedric L Mulumba

**Hands-On Assignments Part I**

|  |
| --- |
| **Assignment 5-1: Creating a Procedure** |
| CREATE OR REPLACE PROCEDURE prod\_name\_sp  (p\_prodid in bb\_product.idproduct%type,  p\_descrip in bb\_product.description%type)  IS    BEGIN  UPDATE bb\_product  SET description = p\_descrip  WHERE idproduct = p\_prodid;  COMMIT;  END; |
|  |
| SELECT \*  FROM bb\_product; |
|  |
| BEGIN  prod\_name\_sp(1, 'CapressoBar Model #388');  END; |
|  |
| SELECT \*  FROM bb\_product; |
|  |
| **Assignment 5-2: Using a Procedure with IN Parameters** |
| CREATE OR REPLACE PROCEDURE prod\_add\_sp  (p\_name in bb\_product.productname%type,  p\_descrip in bb\_product.description%type,  p\_image in bb\_product.productimage%type,  p\_price in bb\_product.price%type,  p\_active in bb\_product.active%type)  IS    BEGIN  INSERT INTO bb\_product  (idproduct, productname, description,  productimage, price, active)  VALUES (bb\_prodid\_seq.nextval, p\_name,  p\_descrip, p\_image, p\_price, p\_active);  COMMIT;    END; |
|  |
| BEGIN  prod\_add\_sp('Roasted Blend',  'Well-balanced mix of roasted beans, medium body', 'roasted.jpg', 9.50, 1);  END; |
|  |
| SELECT idproduct, productname, description, productimage, price, active  FROM bb\_product  WHERE productname = 'Roasted Blend'; |
|  |
| **Assignment 5-3: Calculating the Tax on an Order** |
| CREATE OR REPLACE PROCEDURE tax\_cost\_sp  (p\_state in bb\_tax.state%type,  p\_subtot in number,  p\_tax out number)  IS  BEGIN  SELECT (taxrate \* p\_subtot)  INTO p\_tax  FROM bb\_tax  WHERE state = p\_state;    EXCEPTION  WHEN no\_data\_found then  p\_tax := 0;  END; |
|  |
| DECLARE  lv\_g\_tax number;  BEGIN  tax\_cost\_sp('VA',100,lv\_g\_tax);  DBMS\_OUTPUT.PUT\_LINE(lv\_g\_tax);  END; |
|  |
| **Assignment 5-4: Updating Columns in a Table** |
| CREATE OR REPLACE PROCEDURE basket\_confirm\_sp  (p\_id in bb\_basket.idbasket%type,  p\_sub in bb\_basket.subtotal%type,  p\_ship in bb\_basket.shipping%type,  p\_tax in bb\_basket.tax%type,  p\_total in bb\_basket.total%type)  IS    BEGIN  UPDATE bb\_basket  SET orderplaced = 1, subtotal = p\_sub,  shipping = p\_ship, tax = p\_tax,  total = p\_total  WHERE idbasket = p\_id;  COMMIT;    END; |
|  |
| INSERT INTO bb\_basket(idbasket, quantity, idshopper, orderplaced,  subtotal, total, shipping, tax, dtcreated, promo)  VALUES (17, 2, 22, 0, 0, 0, 0, 0, '28-feb-07', 0);  INSERT INTO bb\_basketitem(idbasketitem, idproduct, price, quantity,  idbasket, option1, option2)  VALUES (44, 7, 10.8, 3, 17, 2, 3);    INSERT INTO bb\_basketitem(idbasketitem, idproduct, price, quantity,  idbasket, option1, option2)  VALUES (45, 8, 10.8, 3, 17, 2, 3);  COMMIT; |
|  |
| BEGIN  basket\_confirm\_sp(17, 64.80, 8.00, 1.94, 74.74);  END; |
|  |
| SELECT subtotal, shipping, tax, total, orderplaced  FROM bb\_basket  WHERE idbasket = 17; |
|  |
| **Assignment 5-5: Updating Order Status** |
| CREATE OR REPLACE PROCEDURE status\_ship\_sp  (p\_id in bb\_basketstatus.idbasket%type,  p\_date in bb\_basketstatus.dtstage%type,  p\_shipper in bb\_basketstatus.shipper%type,  p\_track in bb\_basketstatus.shippingnum%type)  IS  BEGIN  INSERT INTO bb\_basketstatus  (idstatus, idbasket, idstage, dtstage, shipper, shippingnum)  VALUES (bb\_status\_seq.nextval, p\_id, 3, p\_date, p\_shipper, p\_track);  COMMIT;  END; |
|  |
| BEGIN  status\_ship\_sp(3, '20-FEB-07', 'UPS', 'zw2384yxk4957');  END; |
|  |
| SELECT idstatus, idbasket,idstage,  dtstage, shipper, shippingnum  FROM bb\_basketstatus; |
|  |
| **Assignment 5-6: Returning Order Status Information** |
| CREATE OR REPLACE PROCEDURE status\_sp  (p\_id in bb\_basketstatus.idbasket%type,  p\_date out bb\_basketstatus.dtstage%type,  p\_status out bb\_basketstatus.notes%type)  IS    lv\_stage bb\_basketstatus.idstage%type;  BEGIN  SELECT idstage, dtstage  INTO lv\_stage, p\_date  FROM bb\_basketstatus  WHERE idbasket = p\_id  AND dtstage = (select max(dtstage) from bb\_basketstatus);    CASE lv\_stage  WHEN 1 then p\_status := 'Submitted and received';  WHEN 2 then p\_status := 'Confirmed, processed, sent to shipping';  WHEN 3 then p\_status := 'Shipped';  WHEN 4 then p\_status := 'Cancelled';  WHEN 5 then p\_status := 'Backordered';  ELSE p\_status := 'No status available';  END CASE;  EXCEPTION  WHEN no\_data\_found THEN  p\_status := 'No status available';  END; |
|  |
| **Assignment 5-7: Identifying Customers** |
| CREATE OR REPLACE PROCEDURE promo\_ship\_sp  (p\_date in bb\_basket.dtcreated%type,  p\_month in bb\_promolist.month%type,  p\_year in bb\_promolist.year%type)  IS  lv\_min bb\_basket.idshopper%type;  lv\_max bb\_basket.idshopper%type;  lv\_date bb\_basket.dtcreated%type;  lv\_shopper bb\_basket.idshopper%type;  lv\_dupe\_id number;    BEGIN  SELECT min(idshopper), max(idshopper)  INTO lv\_min, lv\_max  FROM bb\_basket;  FOR i IN lv\_min..lv\_max loop      SELECT max(dtcreated)  INTO lv\_date  FROM bb\_basket  WHERE idshopper = i;    IF p\_date >= lv\_date then  SELECT count(idshopper)  INTO lv\_dupe\_id  FROM bb\_promolist  WHERE i = idshopper;    IF lv\_dupe\_id = 1 then continue;  ELSE  INSERT INTO bb\_promolist  (idshopper, month, year, promo\_flag, used)  VALUES (i, p\_month, p\_year, 1, 'N');  END IF;  END IF;  END LOOP;  COMMIT;  END; |
|  |
| BEGIN  promo\_ship\_sp('15-FEB-12', 'APR', 2012);  END; |
|  |
| SELECT \*  FROM bb\_promolist; |
|  |
| **Assignment 5-8: Adding Items to a Basket** |
| CREATE OR REPLACE PROCEDURE basket\_add\_sp  (p\_id in bb\_basketitem.idbasket%type,  p\_prod in bb\_basketitem.idproduct%type,  p\_price in bb\_basketitem.price%type,  p\_qnty in bb\_basketitem.quantity%type,  p\_size in bb\_basketitem.option1%type,  p\_form in bb\_basketitem.option2%type)  IS    BEGIN  INSERT INTO bb\_basketitem (idbasketitem, idbasket,  idproduct, price, quantity, option1, option2)  values (bb\_idbasketitem\_seq.nextval, p\_id, p\_prod,  p\_price, p\_qnty, p\_size, p\_form);  COMMIT;  END; |
|  |
| BEGIN  basket\_add\_sp(14, 8, 10.80, 1, 2, 4);  END; |
|  |
| SELECT \*  FROM bb\_basketitem; |
| **Assignment 5-9: Creating a Logon Procedure** |
| CREATE OR REPLACE PROCEDURE member\_ck\_sp(p\_user IN VARCHAR2,  p\_pass IN VARCHAR2) IS  lv\_check\_txt VARCHAR2(10) := 'INVALID';  lv\_fullname\_txt VARCHAR2(25);  CURSOR MEMBER\_CUR IS  SELECT firstname, lastname, cookie, username, password  FROM bb\_shopper  WHERE username = p\_user  AND password = p\_pass;  BEGIN  FOR REC\_CUR IN MEMBER\_CUR LOOP  IF p\_user = rec\_cur.username AND p\_pass = rec\_cur.password THEN  lv\_fullname\_txt := rec\_cur.firstname || ' ' || rec\_cur.lastname;  dbms\_output.put\_line('Welcome: ' || lv\_fullname\_txt  || 'Cookie: ' || rec\_cur.cookie);  ELSE  dbms\_output.put\_line(lv\_check\_txt);  END IF;  END LOOP;  END member\_ck\_sp; |
|  |
| BEGIN  member\_ck\_sp('rat55', 'kile');  END; |
|  |
|  |