**Assignment #3 – Cursors, Procedures & Functions**

**Question :1**

**PL/SQL Code:**

CREATE OR REPLACE

FUNCTION totalSpending

(in\_shopid IN number)

RETURN NUMBER

IS

tSpending NUMBER(5,2) :=0 ;

BEGIN

Select sum(subtotal) INTO tSpending from bb\_basket where IDSHOPPER = in\_shopid;

IF tSpending IS NULL THEN

RAISE NO\_DATA\_FOUND;

END IF;

RETURN tSpending;

END;

/

DECLARE

total\_spend NUMBER(5,2) := 0;

shopper\_id NUMBER(5,2) := 22;

BEGIN

total\_spend := totalSpending(shopper\_id);

DBMS\_OUTPUT.PUT\_LINE('Total Spending of shpper id '||shopper\_id ||' is '|| total\_spend);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No Data Found');

END;

**Output Screenshots:**

**Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application, email

Description automatically generated**

**Question :2**

**PL/SQL Code:**

create or replace procedure update\_order\_status

(b\_id in bb\_basketstatus.idbasket%type,

stage\_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, b\_id, 3, stage\_date, p\_shipper, p\_track);

commit;

end;

**/**

DECLARE

BEGIN

update\_order\_status(3, '25-02-12', 'UPS', 'ZW45325567D3245J56');

END;

**Output Screenshots:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Question :3**

**PL/SQL Code:**

CREATE OR REPLACE FUNCTION insert\_product(idbasket IN NUMBER, productid IN NUMBER, unit\_price IN NUMBER, quantity IN NUMBER)

RETURN VARCHAR2

IS

presentorder bb\_basketitem.idbasketitem%TYPE;

message VARCHAR2(100);

basketitem bb\_basketitem.idbasketitem%TYPE;

BEGIN

basketitem := BB\_IDBASKETITEM\_SEQ.NEXTval;

INSERT INTO bb\_basketitem(idbasketitem, idproduct, price, quantity, idbasket)

VALUES(basketitem, productid, unit\_price, quantity, idbasket);

SELECT idbasketitem INTO presentorder

FROM bb\_basketitem

WHERE idbasketitem = basketitem;

IF(presentorder = basketitem) THEN

MESSAGE:= 'Order updated.';

ELSE

MESSAGE:= 'Order not updated';

END IF;

RETURN MESSAGE;

END;

**/**

DECLARE

output\_message varchar2(50);

BEGIN

output\_message:= insert\_product(3, 7, 21.5, 8);

DBMS\_OUTPUT.PUT\_LINE(output\_message);

END;

**Output Screenshots:**

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Table

Description automatically generated

**Question :5**

**PL/SQL Code:**

create or replace PROCEDURE add\_product

(p\_name IN BB\_PRODUCT.PRODUCTNAME%TYPE,

p\_description IN BB\_PRODUCT.DESCRIPTION%TYPE,

p\_image IN BB\_PRODUCT.PRODUCTIMAGE%TYPE,

p\_price IN BB\_PRODUCT.PRICE%TYPE,

p\_salesstart IN BB\_PRODUCT.SALESTART%TYPE,

p\_saleend IN BB\_PRODUCT.SALEEND%TYPE,

p\_saleprice IN BB\_PRODUCT.SALEPRICE%TYPE,

p\_active IN BB\_PRODUCT.ACTIVE%TYPE,

p\_featured IN BB\_PRODUCT.FEATURED%TYPE,

p\_featurestart IN BB\_PRODUCT.FEATURESTART%TYPE,

p\_featureend IN BB\_PRODUCT.FEATUREEND%TYPE,

p\_typ IN BB\_PRODUCT.TYPE%TYPE,

p\_iddepartment IN BB\_PRODUCT.IDDEPARTMENT%TYPE,

p\_stock IN BB\_PRODUCT.STOCK%TYPE,

p\_ordered IN BB\_PRODUCT.ORDERED%TYPE,

p\_reorder IN BB\_PRODUCT.REORDER%TYPE)

IS

ID\_BASKET bb\_basketstatus.idbasket%type;

BEGIN

INSERT INTO BB\_PRODUCT (IDPRODUCT, PRODUCTNAME, DESCRIPTION, PRODUCTIMAGE, PRICE, SALESTART, SALEEND, SALEPRICE, ACTIVE, FEATURED, FEATURESTART, FEATUREEND, TYPE, IDDEPARTMENT, STOCK, ORDERED, REORDER)

VALUES (BB\_PRODID\_SEQ.NEXTVAL, p\_name, p\_description, p\_image, p\_price, p\_salesstart, p\_saleend, p\_saleprice, p\_active, p\_featured, p\_featurestart, p\_featureend, p\_typ, p\_iddepartment, p\_stock, p\_ordered, p\_reorder);

COMMIT;

END;

**/**

DECLARE

BEGIN

add\_product('Green Tea', 'Made from Camellia sinensis leaves and buds.', 'greentea.jpg', 10, '1-AUG-12', '28-AUG-12', 7, 1, null, null, null, 'C', 2, 80, 40, 20);

END;

**Output Screenshots:**

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated