

--1

CREATE OR REPLACE FUNCTION totalspending (shopper_id bb_basket.idshopper%TYPE)

RETURN bb_basket.subtotal%TYPE

IS

total bb_basket.subtotal%TYPE;

shopper_not_found Exception;

BEGIN

SELECT

SUM(subtotal)

INTO total

FROM

bb_basket

WHERE

idshopper = shopper_id;

if sql%notfound then

raise shopper_not_found;

End if;

RETURN total;

Exception

when shopper_not_found then

```
dbms_output.put_line('shopper does not exist ');
```

```
END;
```

```
--testing
```

```
/*SELECT DISTINCT
```

```
idshopper      "Shoppers ID",
```

```
totalspending(idshopper) "Total Spending"
```

```
FROM
```

```
bb_basket order by idshopper asc; */
```

```
begin
```

```
for shopper in
```

```
( select distinct idshopper , totalspending(idshopper) as total from bb_basket )
```

```
loop
```

```
DBMS_output.put_line( shopper.idshopper || ' - ' || shopper.total);
```

```
end loop;
```

```
end;
```

```
--2
```

```
create or replace procedure update_order_status ( status_id IN NUMBER,  
                                                stg_id IN NUMBER,  
                                                note IN VARCHAR2,  
                                                shipper_name IN VARCHAR2,  
                                                shipping_num IN VARCHAR2)  
AS
```

```
Begin
```

```
    Update bb_basketstatus set  
        idstage = stg_id,  
        notes = note,  
        shipper = shipper_name,  
        shippingnum = shipping_num,  
        dtstage = sysdate  
        where idstatus = status_id;  
End update_order_status;
```

```
-- testing
```

```
Declare
```

```
    sts_id Number :=2;  
    stg_id number := 4;  
    notes varchar2(50) := 'example note';  
    sname varchar2(10) := 'abc';  
    snum varchar2(20) := 'A123B456C789';
```

```
begin
```

```
    update_order_status(sts_id, stg_id, notes, sname, snum);
```

end;

--3

create or replace function new_prod(prod_id in BB_BASKETITEM.IDPRODUCT%type,quantity in number)

RETURN VARCHAR2

IS

lv_meg VARCHAR2(32) ;

u_price number(10,3);

basketitem_id number (3);

BEGIN

select max(IDBASKETITEM) into basketitem_id from BB_BASKETITEM;

basketitem_id := basketitem_id + 1;

Select price into u_price from BB_PRODUCT where IDPRODUCT= prod_id;

INSERT INTO BB_BASKETITEM (IDBASKETITEM,IDPRODUCT,PRICE,QUANTITY)

VALUES (basketitem_id, prod_id,u_price,quantity);

if sql%found then

lv_meg := 'successfully inserted';

elsif sql%notfound then

lv_meg := 'fail to insert';

end if;

RETURN lv_meg;

```
END;
```

```
--testing
```

```
DECLARE
```

```
    op_meg VARCHAR2(35);
```

```
BEGIN
```

```
    op_meg := new_prod(3, 5);
```

```
    DBMS_OUTPUT.PUT_LINE('output : ' || op_meg);
```

```
END;
```

```
-- 4
```

```
--insert covid project
```

```
insert into dd_project ( idproj, projname, projstartdate, projenddate, projfundgoal, projcoord)
```

```
    values(505, 'Covid-19 relief fund', to_date('01-jan-22','dd-mon-yy'), to_date('01-dec-22','dd-mon-yy'), 5000000, 'Parth Patel');
```

```
-- inser pledge
```

```
insert into dd_pledge (idpledge, iddonor , pledgedate , pledgeamt, idproj, idstatus, paymonths, campaign, firstpledge)
```

```
values(113, 303, sysdate, 999, 505, 20, 12, 777, 'N' );  
insert into dd_pledge (idpledge, iddonor , pledgedate , pledgeamt, idproj, idstatus, paymonths,  
campaign, firstpledge)
```

```
values(114, 309, sysdate, 888, 505, 20, 24, 777, 'N' );
```

```
-- func
```

```
create or replace function total_pledge (proj_id In number)
```

```
return number
```

```
is
```

```
total number;
```

```
begin
```

```
select
```

```
sum(pledgeamt)
```

```
into
```

```
total
```

```
from
```

```
dd_pledge
```

```
where
```

```
idproj = proj_id;
```

```
if sql%notfound then
```

```
return 0;
```

```
end if;
```

```
return total;
```

exception

when others then

DBMS_output.put_line('Error !!!! for project id: ' || proj_id);

return 0;

end total_pledge;

-- testing

--select distinct idproj, projname, total_pledge(idproj) as total

--from dd_project;

begin

for plg in (

select idproj, projname, total_pledge(idproj) as total_pledge

from dd_project

)Loop

DBMS_output.put_line(plg.idproj || ' - ' || plg.projname || ' -> ' || plg.total_pledge || '\$');

End loop;

end;

--5

```
CREATE or REPLACE PROCEDURE PROD_ADD( product_name IN VARCHAR2,  
    description IN VARCHAR2,  
    image_filename IN VARCHAR2,  
    price IN NUMBER,  
    active_status IN NUMBER,  
    p_type IN VARCHAR2,  
    dept IN NUMBER,  
    stock IN NUMBER,  
    ord IN NUMBER)
```

AS

```
prod_id number(2) := bb_prodid_seq.nextval;
```

BEGIN

```
INSERT INTO BB_PRODUCT(IDPRODUCT, PRODUCTNAME, DESCRIPTION, PRODUCTIMAGE, PRICE,  
ACTIVE, TYPE, IDDEPARTMENT, STOCK, ORDERED)
```

```
VALUES(prod_id, product_name, description, image_filename, price, active_status, p_type,  
dept, stock, ord);
```

```
END PROD_ADD;
```

--testing

declare

```
product_name VARCHAR2(30) := 'Ex product 1';
```

```
description VARCHAR2(50) := 'this prduct is to check the function';
```



```
image_filename VARCHAR2(20) := 'example1.jpg ';
```

```
price NUMBER(30,2) := '99.33';
```

```
active_status NUMBER(2):= '1';
```

```
p_type VARCHAR2(2) := 'E';
```

```
dept NUMBER(2) := 1;
```

```
stock NUMBER(3) := 24;
```

```
ord NUMBER(3):= 12;
```

```
begin
```

```
    PROD_ADD ( product_name, description, image_filename, price, active_status, p_type, dept, stock,  
ord);
```

```
End;
```