|  |  |
| --- | --- |
|  | Assignment 03 |
|  |  |
|  | Parth Chandgadhiya 300986134  COMP 214 - [Sec 12]  12/7/18 |

Table of Contents

[DDPAY\_SP Procedure: 2](#_Toc531981029)

[DDPAY\_SP Output: 2](#_Toc531981030)

[Procedure: 3](#_Toc531981031)

[Executing Procedure: 4](#_Toc531981032)

[DDCKPAY\_SP Procedure: 4](#_Toc531981033)

[DDCKPAY\_SP Output: 4](#_Toc531981034)

[1. Output for ID 104 and Payment Amount $20 4](#_Toc531981035)

[2. Output for ID 104 and Payment Amount $25 5](#_Toc531981036)

[3. a pledge ID for a pledge that does not have monthly payments associated with it. 6](#_Toc531981037)

[Procedure: 6](#_Toc531981038)

[Executing Procedure: 7](#_Toc531981039)

[1.) Executing Procedure for ID 104 and Payment Amount $25 7](#_Toc531981040)

[2.) Executing Procedure for ID 104 and Payment Amount $20 7](#_Toc531981041)

[3.) Executing Procedure for a pledge ID for a pledge that does not have monthly payments associated with it. 7](#_Toc531981042)

[DDCHKBAL\_SP Procedure: 8](#_Toc531981043)

[DDCHKBAL\_SP Output: 9](#_Toc531981044)

[1) For Pledge id 107 9](#_Toc531981045)

[2) For Pledge id 104 9](#_Toc531981046)

[Procedure: 10](#_Toc531981047)

[Executing Procedure: 11](#_Toc531981048)

[1). Executing Procedure for Pledge Id 104 11](#_Toc531981049)

[2). Executing Procedure for Pledge Id 104 11](#_Toc531981050)

[SHOP\_QUERY\_PKG Package and Body: 11](#_Toc531981051)

[SHOP\_QUERY\_PKG Output: 12](#_Toc531981052)

[SHOP\_QUERY\_PKG pacakage: 13](#_Toc531981053)

[Executing SHOP\_QUERY\_PKG: 14](#_Toc531981054)

[TAX\_RATE\_PKG Package: 14](#_Toc531981055)

[TAX\_RATE\_PKG Invoked: 15](#_Toc531981056)

[Package: 15](#_Toc531981057)

[Invoking Package: 15](#_Toc531981058)

[LOGIN\_PKG Package and Body: 16](#_Toc531981059)

[LOGIN\_PKG Output: 17](#_Toc531981060)

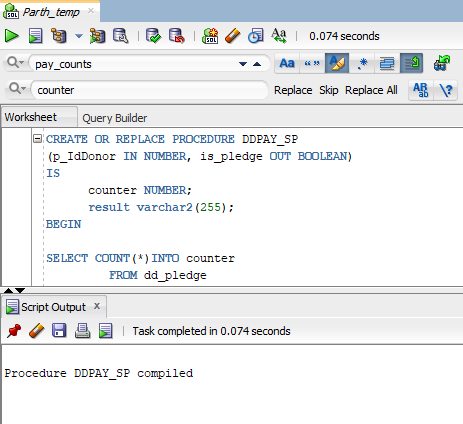
[LOGIN\_PKG Package: 17](#_Toc531981061)

[LOGIN\_PKG Invoked: 18](#_Toc531981062)

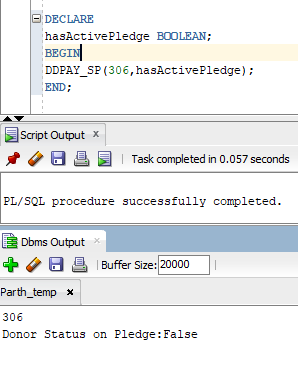
PART A

Answer 01

# DDPAY\_SP Procedure:



# DDPAY\_SP Output:



## Procedure:

CREATE OR REPLACE PROCEDURE DDPAY\_SP

(p\_IdDonor IN NUMBER, is\_pledge OUT BOOLEAN)

IS

counter NUMBER;

result varchar2(255);

BEGIN

SELECT COUNT(\*)INTO counter

FROM dd\_pledge

WHERE iddonor = counter AND paymonths > 1;

IF counter > 1 THEN

is\_pledge := TRUE;

ELSE is\_pledge := FALSE;

END IF;

DBMS\_OUTPUT.PUT\_LINE(p\_Iddonor);

DBMS\_OUTPUT.PUT\_LINE

(

'Donor Status on Pledge:' ||

CASE is\_pledge

WHEN TRUE THEN 'True'

ELSE 'False'

END

);

END DDPAY\_SP;

## Executing Procedure:

DECLARE

hasActivePledge BOOLEAN;

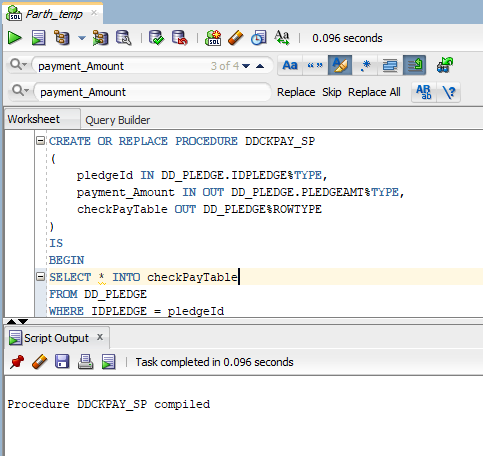
BEGIN

DDPAY\_SP(306,hasActivePledge);

END;

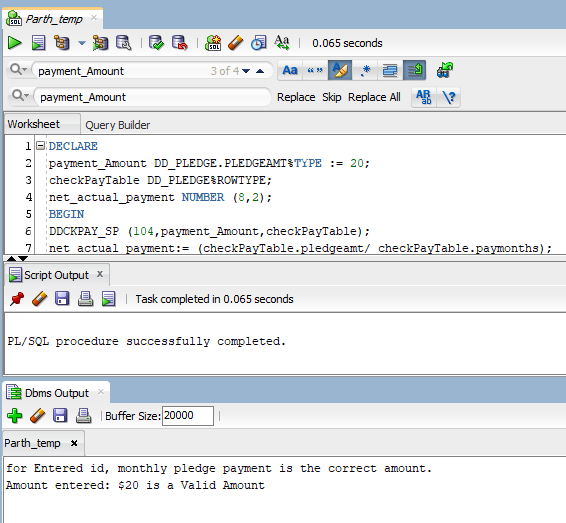
Answer 02

# DDCKPAY\_SP Procedure:

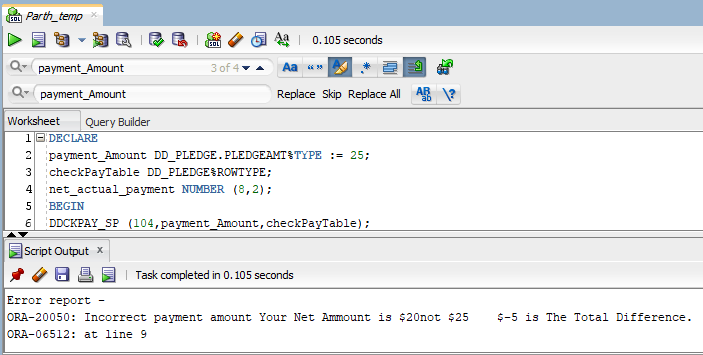


# DDCKPAY\_SP Output:

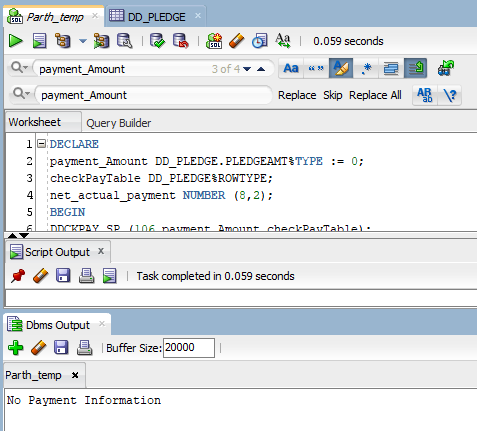
## 1. Output for ID 104 and Payment Amount $20



## 2. Output for ID 104 and Payment Amount $25



## 3. a pledge ID for a pledge that does not have monthly payments associated with it.



## Procedure:

CREATE OR REPLACE PROCEDURE DDCKPAY\_SP

(

pledgeId IN DD\_PLEDGE.IDPLEDGE%TYPE,

payment\_Amount IN OUT DD\_PLEDGE.PLEDGEAMT%TYPE,

checkPayTable OUT DD\_PLEDGE%ROWTYPE

)

IS

BEGIN

SELECT \* INTO checkPayTable

FROM DD\_PLEDGE

WHERE IDPLEDGE = pledgeId

AND PAYMONTHS >0;

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No Payment Information');

END IF;

END;

## Executing Procedure:

### 1.) Executing Procedure for ID 104 and Payment Amount $25

DECLARE

payment\_Amount DD\_PLEDGE.PLEDGEAMT%TYPE := 25;

checkPayTable DD\_PLEDGE%ROWTYPE;

net\_actual\_payment NUMBER (8,2);

BEGIN

DDCKPAY\_SP (104,payment\_Amount,checkPayTable);

net\_actual\_payment:= (checkPayTable.pledgeamt/ checkPayTable.paymonths);

IF payment\_Amount != net\_actual\_payment THEN

RAISE\_APPLICATION\_ERROR(-20050, 'Incorrect payment amount Your Net Ammount is $' ||net\_actual\_payment || 'not $' || payment\_Amount

|| ' $'||( net\_actual\_payment - payment\_Amount) ||' is The Total Difference.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('for Entered id, monthly pledge payment is the correct amount.');

END IF;

DBMS\_OUTPUT.put\_line('Amount entered: $' || net\_actual\_payment || ' is a Valid Amount');

END;

### 2.) Executing Procedure for ID 104 and Payment Amount $20

DECLARE

payment\_Amount DD\_PLEDGE.PLEDGEAMT%TYPE := 25;

checkPayTable DD\_PLEDGE%ROWTYPE;

net\_actual\_payment NUMBER (8,2);

BEGIN

DDCKPAY\_SP (104,payment\_Amount,checkPayTable);

net\_actual\_payment:= (checkPayTable.pledgeamt/ checkPayTable.paymonths);

IF payment\_Amount != net\_actual\_payment THEN

RAISE\_APPLICATION\_ERROR(-20050, 'Incorrect payment amount Your Net Ammount is $' ||net\_actual\_payment || 'not $' || payment\_Amount

|| ' $'||( net\_actual\_payment - payment\_Amount) ||' is The Total Difference.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('for Entered id, monthly pledge payment is the correct amount.');

END IF;

DBMS\_OUTPUT.put\_line('Amount entered: $' || net\_actual\_payment || ' is a Valid Amount');

END;

### 3.) Executing Procedure for a pledge ID for a pledge that does not have monthly payments associated with it.

DECLARE

payment\_Amount DD\_PLEDGE.PLEDGEAMT%TYPE := 0;

checkPayTable DD\_PLEDGE%ROWTYPE;

net\_actual\_payment NUMBER (8,2);

BEGIN

DDCKPAY\_SP (106,payment\_Amount,checkPayTable);

net\_actual\_payment:= (checkPayTable.pledgeamt/ checkPayTable.paymonths);

IF payment\_Amount != net\_actual\_payment THEN

RAISE\_APPLICATION\_ERROR(-20050, 'Incorrect payment amount Your Net Ammount is $' ||net\_actual\_payment || 'not $' || payment\_Amount

|| ' $'||( net\_actual\_payment - payment\_Amount) ||' is The Total Difference.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('for Entered id, monthly pledge payment is the correct amount.');

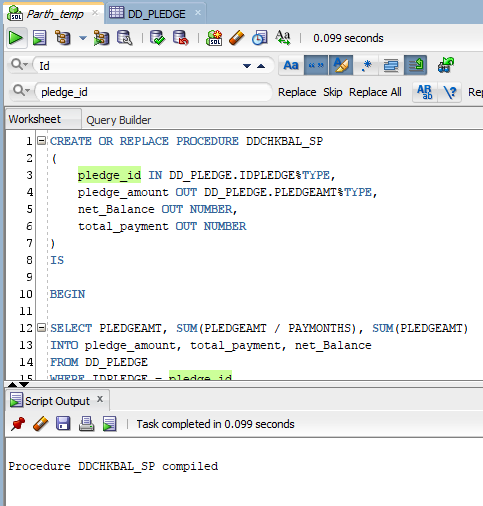
END IF;

DBMS\_OUTPUT.put\_line('Amount entered: $' || net\_actual\_payment || ' is a Valid Amount');

END;

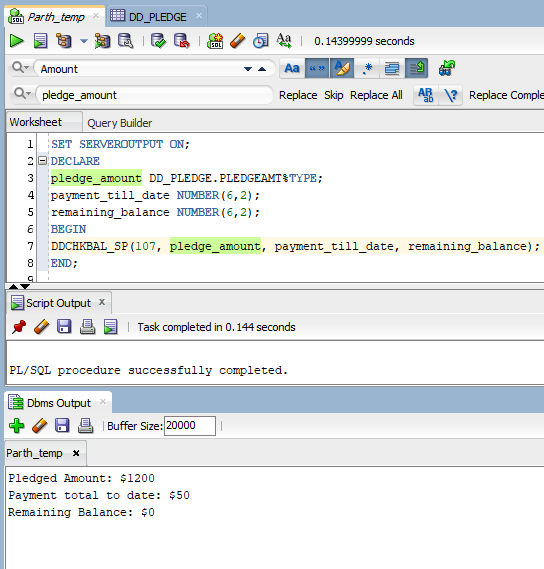
Answer 03

# DDCHKBAL\_SP Procedure:

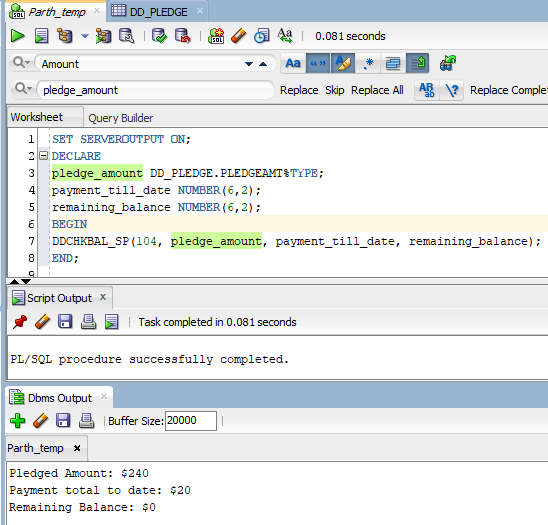


# DDCHKBAL\_SP Output:

## 1) For Pledge id 107



## 2) For Pledge id 104



## Procedure:

create or replace PROCEDURE DDCHKBAL\_SP

(

pledge\_id IN DD\_PLEDGE.IDPLEDGE%TYPE,

pledge\_amount OUT DD\_PLEDGE.PLEDGEAMT%TYPE,

net\_Balance OUT NUMBER,

total\_payment OUT NUMBER

)

IS

BEGIN

SELECT PLEDGEAMT, SUM(PLEDGEAMT / PAYMONTHS), SUM(PLEDGEAMT)

INTO pledge\_amount, total\_payment, net\_Balance

FROM DD\_PLEDGE

WHERE IDPLEDGE = pledge\_id

GROUP BY PLEDGEAMT;

net\_Balance := net\_Balance - pledge\_amount;

DBMS\_OUTPUT.PUT\_LINE('Pledged Amount: $' || pledge\_amount);

DBMS\_OUTPUT.PUT\_LINE('Payment total to date: $'|| total\_payment);

DBMS\_OUTPUT.PUT\_LINE('Remaining Balance: $' || net\_Balance);

END;

## Executing Procedure:

### 1). Executing Procedure for Pledge Id 104

SET SERVEROUTPUT ON;

DECLARE

pledge\_amount DD\_PLEDGE.PLEDGEAMT%TYPE;

payment\_till\_date NUMBER(6,2);

remaining\_balance NUMBER(6,2);

BEGIN

DDCHKBAL\_SP(107, pledge\_amount, payment\_till\_date, remaining\_balance);

END;

### 2). Executing Procedure for Pledge Id 104

SET SERVEROUTPUT ON;

DECLARE

pledge\_amount DD\_PLEDGE.PLEDGEAMT%TYPE;

payment\_till\_date NUMBER(6,2);

remaining\_balance NUMBER(6,2);

BEGIN

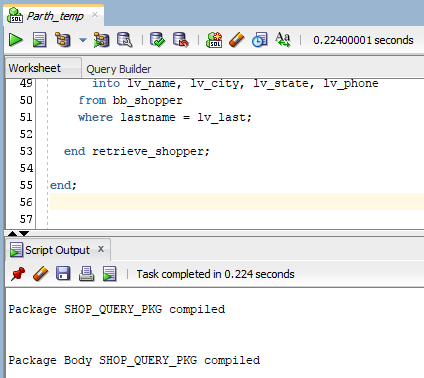
DDCHKBAL\_SP(104, pledge\_amount, payment\_till\_date, remaining\_balance);

END;

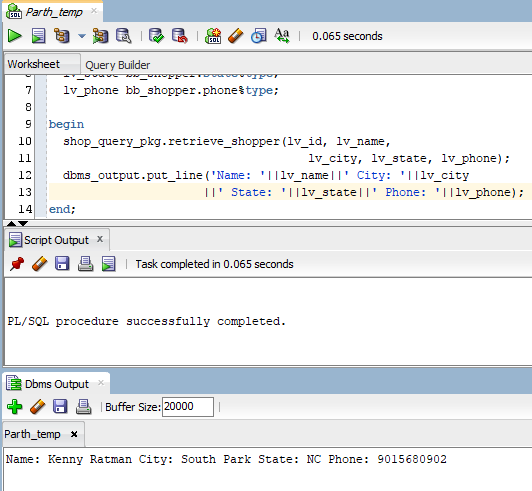
PART B

Answer 01

# SHOP\_QUERY\_PKG Package and Body:



# SHOP\_QUERY\_PKG Output:



## SHOP\_QUERY\_PKG pacakage:

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;

## Executing SHOP\_QUERY\_PKG:

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);

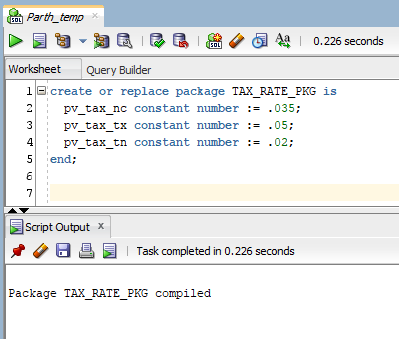
dbms\_output.put\_line('Name: '||lv\_name||' City: '||lv\_city

||' State: '||lv\_state||' Phone: '||lv\_phone);

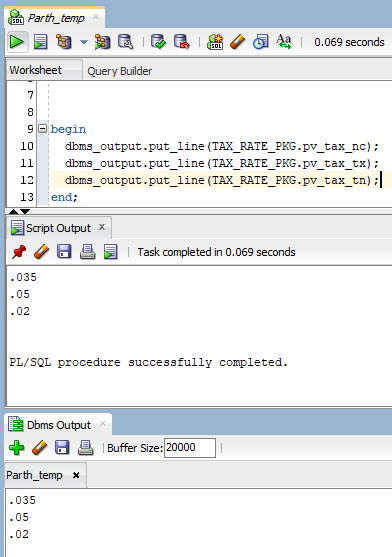
end;

Answer 02

# TAX\_RATE\_PKG Package:



# TAX\_RATE\_PKG Invoked:



## Package:

create or replace package TAX\_RATE\_PKG is

pv\_tax\_nc constant number := .035;

pv\_tax\_tx constant number := .05;

pv\_tax\_tn constant number := .02;

end;

## Invoking Package:

begin

dbms\_output.put\_line(TAX\_RATE\_PKG.pv\_tax\_nc);

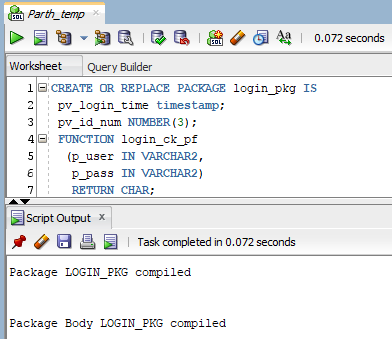
dbms\_output.put\_line(TAX\_RATE\_PKG.pv\_tax\_tx);

dbms\_output.put\_line(TAX\_RATE\_PKG.pv\_tax\_tn);

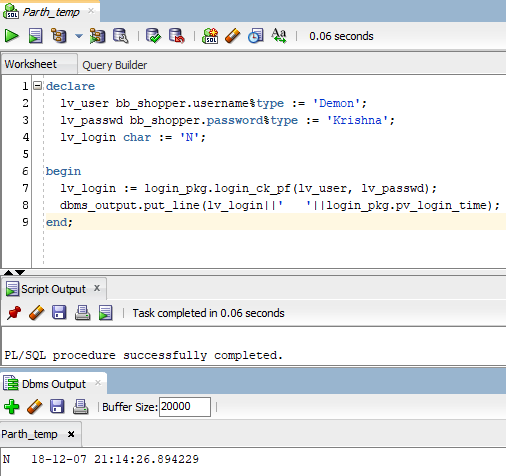
end;

Answer 03

# LOGIN\_PKG Package and Body:



# LOGIN\_PKG Output:



## LOGIN\_PKG 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;

## LOGIN\_PKG Invoked:

declare

lv\_user bb\_shopper.username%type := 'Demon';

lv\_passwd bb\_shopper.password%type := 'Krishna';

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;

**THE END**