```
[21bcs048@mepcolinux fun]$ls

fun  fun.prn

[21bcs048@mepcolinux fun]$cat fun

                                FUNCTIONS AND PROCEDURES

                                ========================


1)CREATE A PROCEDURE WHICH IS USED TO INSERT A RECORD INTO A GIFTCARD TABLE.


 CREATE OR REPLACE PROCEDURE "INSERTORD"

  2  (GID IN NUMBER,USERID IN NUMBER,AMOUNT IN NUMBER,GVALID IN DATE)

  3  IS

  4  BEGIN

  5  INSERT INTO GIFTCARD VALUES(GID,USERID,AMOUNT,GVALID);

  6  END;

  7  /


Procedure created.


SQL> SELECT * FROM GIFTCARD;


       GID     USERID     AMOUNT GVALID

---------- ---------- ---------- ---------

       570        301       5000 07-FEB-30

       522        370      40000 07-MAY-23

       590        303       8000 04-APR-30

       500        333      80000 10-MAY-90

       555        355     242020 01-MAR-70

       560        330     330000 03-APR-90

       505        302     275000 08-JAN-44
```

```
7 rows selected.
```

PROCEDUE CALLING

================

```
SQL> BEGIN
  2     INSERTORD(514,390,90000,'09-APR-2089');
  3     DBMS_OUTPUT.PUT_LINE('NEW GIFTCARD ADDED SUCCESSFULLY');
  4  END;
  5  /
NEW GIFTCARD ADDED SUCCESSFULLY

PL/SQL procedure successfully completed.
```

2)CREATE A PROCEDURE TO DISPLAY THE USER IS NOT HAVE A GIFTCARD.

```
 CREATE OR REPLACE PROCEDURE GIF
  2  IS
  3  CURSOR C1 IS SELECT * FROM USER1;
  4   TROW1 C1%ROWTYPE;
  5   CURSOR C2 IS SELECT * FROM GIFTCARD;
  6   TROW2 C2%ROWTYPE;
  7  BEGIN
  8       OPEN C1;
  9      OPEN C2;
 10      DBMS_OUTPUT.PUT_LINE('NAME');
 11    FETCH C1 INTO TROW1;
 12     FETCH C2 INTO TROW2;
 13     WHILE C1%FOUND LOOP
 14            IF TROW1.USERID!=TROW2.USERID THEN
 15              DBMS_OUTPUT.PUT_LINE(TROW1.NAME);
 16          END IF;
 17            FETCH C1 INTO TROW1;
 18             FETCH C2 INTO TROW2;
 19   END LOOP;
 20   CLOSE C1;
 21   CLOSE C2;
 22  END;
 23  /

Procedure created.
```

PROCEDURE CALLING
=================

```
SQL> BEGIN
  2  GIF;
  3  END;
  4  /
NAME
KARTHI
SRI
BALA
SURESH
VICKY
PRABU
===============================================================
```

PROCEDURE CALLING USING EXEC COMMAND
===================================

```
SQL> EXEC INSERTORD(655,320,67844,'09-SEP-2070');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM GIFTCARD;

       GID     USERID     AMOUNT GVALID
---------- ---------- ---------- ---------
       570        301       5000 07-FEB-30
       522        370      40000 07-MAY-23
       590        303       8000 04-APR-30
       655        320      67844 09-SEP-70
       500        333      80000 10-MAY-90
       555        355     242020 01-MAR-70
       560        330     330000 03-APR-90
       505        302     275000 08-JAN-44
```

PROCEDURE CALLING USING CALL METHOD

===================================

```
 CALL INSERTORD(415,311,9870,'07-MAY-2030');

Call completed.

SQL> SELECT * FROM GIFTCARD;

       GID     USERID     AMOUNT GVALID
---------- ---------- ---------- ---------
       570        301       5000 07-FEB-30
       522        370      40000 07-MAY-23
       590        303       8000 04-APR-30
       500        333      80000 10-MAY-90
       415        311       9870 07-MAY-30
       555        355     242020 01-MAR-70
       560        330     330000 03-APR-90
       505        302     275000 08-JAN-44

8 rows selected.
```

===

CREATE A PROCEDURE TO DELETE THE AMOUNT WHICH IS GREATER THAN THE AMOUNT1.

```
SQL> CREATE OR REPLACE PROCEDURE DEL_AMOUNT
  2   (AMOUNT1 IN GIFTCARD.AMOUNT%TYPE)
  3  IS
  4  BEGIN
  5     DELETE FROM GIFTCARD WHERE AMOUNT>=AMOUNT1;
  6  END;
  7  /

Procedure created.
```

PROCEDURE CALLING.
=================

```
SQL> BEGIN
  2  DEL_AMOUNT(11000);
  3  dbms_output.put_line('DELETED sucessfully!');
  4  END;
  5  /
```

```
DELETED sucessfully!

PL/SQL procedure successfully completed.
```

==

FUNCTIONS

=========

1)CREATE A USERDEFINED FUNCTION TO DISPLAY THE TOTAL NO.OF USERS.

```
SQL> CREATE OR REPLACE FUNCTION TUSER
  2  RETURN NUMBER IS
  3     TOTAL NUMBER(2) :=0;
  4  BEGIN
  5     SELECT COUNT(*) INTO TOTAL FROM USER1;
  6     RETURN TOTAL;
  7  END;
  8  /

Function created.
```

FUNCTION CALLING

================

```
SQL> DECLARE
  2     C NUMBER(2);
  3  BEGIN
  4     C:=TUSER();
  5      DBMS_OUTPUT.PUT_LINE('TOTAL USERS :'|| C);
  6  END;
  7  /
```

```
TOTAL USERS :9

PL/SQL procedure successfully completed.
```

2)CREATE A USERDEFINED FUNCTION TO DISPLAY THE AMOUNT FOR THE GID=570.

```
CREATE OR REPLACE FUNCTION FIND_AMOUNT
 RETURN NUMBER
 IS
  AMT NUMBER;
BEGIN
 SELECT AMOUNT INTO AMT FROM GIFTCARD WHERE GID=570;
 RETURN AMT;
 END FIND_AMOUNT;
/

Function created.
```

FUNCTION CALLING

================

```
SQL> DECLARE
AMOUNT NUMBER;
BEGIN
AMOUNT:=FIND_AMOUNT();
dbms_output.put_line('Total Amount: ' || AMOUNT);
END;
/
Total Amount: 5000

PL/SQL procedure successfully completed.
```

FUNCTION CALLING USING DUAL TABLE

=================================

```
(1) SQL> SELECT TUSER() AS COUNT FROM DUAL;

     COUNT
----------
         7
```

2)FUNCTION CALLING USING ORIGINAL TABLE.

```
SQL> SELECT USERID,NAME,PHONENO,TUSER() AS COUNT FROM USER1;

    USERID NAME                    PHONENO      COUNT
---------- -------------------- ---------- ----------
       301 SRI                  9087654341          7
       302 BALA                 9098754321          7
       303 SURESH               9098751231          7
       355 ABI                  9800003217          7
       370 VICKY                9990003217          7
       330 PRABU                9991103217          7

6 rows selected.
```

==

CURSOR-EXPLICIT

===============

1)WRITE A QUERY TO CREATE A CURSOR TO DISPLAY NAME,PHONENO,USERID OF USER1.

```
SQL> SET SERVEROUTPUT ON;
SQL>  DECLARE
  2    CURSOR C1 IS SELECT * FROM USER1;
  3   TROW C1%ROWTYPE;
  4    BEGIN
  5    OPEN C1;
  6   LOOP
  7   FETCH C1 INTO TROW;
  8  EXIT WHEN (C1%NOTFOUND);
  9  DBMS_OUTPUT.PUT_LINE('NAME:'||TROW.NAME||'MOBILE NO:'||TROW.PHONENO||'U
SERID:'||TROW.USERID);
 10  END LOOP;
 11  CLOSE C1;
 12  END;
 13  /
NAME:SRIMOBILE NO:9087654341USERID:301
NAME:BALAMOBILE NO:9098754321USERID:302
NAME:SURESHMOBILE NO:9098751231USERID:303
NAME:ABIMOBILE NO:9800003217USERID:355
NAME:VICKYMOBILE NO:9990003217USERID:370
NAME:PRABUMOBILE NO:9991103217USERID:330
NAME:JENIMOBILE NO:USERID:340

PL/SQL procedure successfully completed.
```

CURSOR-IMPLICIT

===============

1)WRITE A QUERY TO CREATE A CURSOR TO DECREMENT CVV FOR ALL CREDITCARDS BY 10.

```
SQL> DECLARE
  2    TOTAL_ROWS NUMBER(3);
  3  BEGIN
  4    UPDATE CREDITCARD
  5    SET CVV=CVV-10;
  6    IF SQL%NOTFOUND THEN
  7          DBMS_OUTPUT.PUT_LINE('NO CREDITCARD SELECTED');
  8    ELSIF SQL%FOUND THEN
  9          TOTAL_ROWS:=SQL%ROWCOUNT;
 10           DBMS_OUTPUT.PUT_LINE(TOTAL_ROWS||'CREDITCARD SELECTED');
 11     END IF;
 12  END;
 13  /
OLD.CVV:100001
NEW.CVV:99991
OLD.CVV:65453
NEW.CVV:65443
OLD.CVV:50000
NEW.CVV:49990
OLD.CVV:11000
NEW.CVV:10990
OLD.CVV:100001
NEW.CVV:99991
5CREDITCARD SELECTED

PL/SQL procedure successfully completed.
```

EXCEPTIONS

==========

2)WRITE A QUERY TO CREATE A CURSOR THAT THROWS AN EXECUTION ERROR WHEN NOT FOUND.

```
SQL>  DECLARE
  2    GIFD GIFTCARD.GID%TYPE :=8;
  3   GVALIDITY GIFTCARD.GVALID%TYPE;
  4   BEGIN
  5      SELECT GVALID INTO GVALIDITY
  6    FROM GIFTCARD
  7   WHERE GID=GIFD;
  8    DBMS_OUTPUT.PUT_LINE('GID:'||GIFD);
  9   EXCEPTION
 10     WHEN NO_DATA_FOUND THEN
 11           DBMS_OUTPUT.PUT_LINE('NO SUCH GIFTCARD');
 12     WHEN OTHERS THEN
 13        DBMS_OUTPUT.PUT_LINE('ERROR');
 14   END;
 15  /

NO SUCH GIFTCARD


PL/SQL procedure successfully completed.
```