==

EXPERIMENT NO. 05

==

Author : Diksha Gupta.
Roll no.: 05 [27A]
Date : 18-NOVEMBER-2022.

==

AIM : To write and execute PL/SQL blocks (with exception handling) including PL/SQL subprograms using Oracle 11g.

PROBLEM STATEMENT:

Establish the database relation EMPLOYEE and populate it with sample records. The logical schema of EMPLOYEE table is -

EMPLOYEE (EID, FNAME, LNAME, BIRTHDATE, GENDER, SSN, HIREDATE, SALARY,

DEPARTMENT, DESIGNATION)

*************************************** **QUERY-01** **
Write SQL code to create and execute an anonymous PL/SQL block that will insert 5 tuples into EXAM. Ensure to commit the populated records. Test the insertion in EXAM by displaying its contents
****

```
CREATE TABLE EXAM (
    UROLL NUMBER(4),
    COURSE CHAR(7),
    EXAMDAT DATE,
    CONSTRAINT EXAM_UROLL_PK PRIMARY KEY(UROLL)
);

Table created.
```

```
SET SERVEROUTPUT ON;
DECLARE
       V_UROLL_1 NUMBER(4) :=1001;
       V_CRS_1 CHAR(7) := 'DBMS';
       V_EXDATE_1 DATE ;

       V_UROLL_2 NUMBER(4) :=1002;
       V_CRS_2 CHAR(7) := 'CAO';
       V_EXDATE_2 DATE  ;

       V_UROLL_3 NUMBER(4) :=1003;
       V_CRS_3 CHAR(7) := 'FLAT';
       V_EXDATE_3 DATE ;

       V_UROLL_4 NUMBER(4) :=1004;
       V_CRS_4 CHAR(7) := 'FDLCA';
       V_EXDATE_4 DATE ;

       V_UROLL_5 NUMBER(4) :=1005;
       V_CRS_5 CHAR(7) := 'MATHS';
       V_EXDATE_5 DATE  ;

BEGIN

       SELECT SYSDATE  INTO V_EXDATE_1 FROM DUAL;
       INSERT
          INTO EXAM VALUES(V_UROLL_1,V_CRS_1,V_EXDATE_1);
       DBMS_OUTPUT.PUT_LINE('ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO.
         '||V_UROLL_1);



       SELECT SYSDATE + 1 INTO V_EXDATE_2 FROM DUAL;
       INSERT
         INTO EXAM VALUES(V_UROLL_2,V_CRS_2,V_EXDATE_2);
       DBMS_OUTPUT.PUT_LINE('ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO.
         '||V_UROLL_2);



       SELECT SYSDATE + 2 INTO V_EXDATE_3 FROM DUAL;
       INSERT
         INTO EXAM VALUES(V_UROLL_3,V_CRS_3,V_EXDATE_3);
       DBMS_OUTPUT.PUT_LINE('ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO.
          '||V_UROLL_3);


        SELECT SYSDATE + 3 INTO V_EXDATE_4 FROM DUAL;
        INSERT
          INTO EXAM VALUES(V_UROLL_4,V_CRS_4,V_EXDATE_4);
        DBMS_OUTPUT.PUT_LINE('ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO.
          '||V_UROLL_4);

        SELECT SYSDATE + 4 INTO V_EXDATE_5 FROM DUAL;
        INSERT
          INTO EXAM VALUES(V_UROLL_5,V_CRS_5,V_EXDATE_5);
        DBMS_OUTPUT.PUT_LINE('ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO.
         '||V_UROLL_5);

       COMMIT;
END;
/
```

OUTPUT :

```
ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO. 1001
ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO. 1002
ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO. 1003
ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO. 1004
ROW IS SUCCESSFULLY INSERTED WITH UNIVERSITY ROLL NO. 1005

PL/SQL procedure successfully completed.
```

```
  SELECT *
      FROM EXAM;

     UROLL COURSE  EXAMDAT
---------- ------- ---------
      1001 DBMS    17-NOV-22
      1002 CAO     18-NOV-22
      1003 FLAT    19-NOV-22
      1004 FDLCA   20-NOV-22
      1005 MATHS   21-NOV-22
```

```
*************************************** QUERY-02 ********************************************
Write SQL code to create and execute an anonymous PL/SQL block that will use %TYPE variables
to populate the EMPP table with corresponding tuples in EMPLOYEE table.
********************************************************************************************

CREATE TABLE EMPP
    AS (SELECT ENO AS EID,(FNAME ||' '||LNAME) AS ENAME , HIREDATE , DESIGNATION,SALARY
         FROM EMPLOYEE WHERE  1=2);

Table created.

       SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
         FROM USER_CONSTRAINTS
           WHERE TABLE_NAME='EMPP';

       CONSTRAINT_NAME      C
       -------------------- -
       SYS_C008690          C
       SYS_C008691          C
       SYS_C008692          C
       SYS_C008693          C

       4 rows selected.

       ALTER TABLE EMPP ADD CONSTRAINT EMPP_EID_PK PRIMARY KEY(EID);

       Table altered.


       ALTER TABLE EMPP
          ADD CONSTRAINT EMPP_CHK_DESIGNATION CHECK (DESIGNATION
            IN('Professor' , 'Research Asst.', 'Asso. Professor' ,'Teaching Asst.' , 'Asst.
                Professor'));

       Table altered.

       ALTER TABLE EMPP
          ADD CONSTRAINT EMPP_CHK_SALARY CHECK (SALARY >= 10000);

       Table altered.
```

```
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
     FROM USER_CONSTRAINTS
            WHERE TABLE_NAME='EMPP';

CONSTRAINT_NAME      C
-------------------- -
SYS_C008690          C
SYS_C008691          C
SYS_C008692          C
SYS_C008693          C
EMPP_CHK_DESIGNATION C
EMPP_CHK_SALARY      C
EMPP_EID_PK          P

7 rows selected.
```

```
SET SERVEROUTPUT ON;

DECLARE

       V_EID EMPP.EID%TYPE;

       V_ENAME EMPP.ENAME%TYPE;

       V_HIREDATE EMPP.HIREDATE%TYPE;

       V_DESIGNATION EMPP.DESIGNATION%TYPE;

       V_SALARY EMPP.SALARY%TYPE;

       V_TOTAL NUMBER(3);

       V_FEID NUMBER(4):=7101;


BEGIN

   SELECT COUNT(*) INTO V_TOTAL FROM EMPLOYEE;

     FOR I IN 1..V_TOTAL LOOP

        SELECT ENO , (FNAME||' '|| LNAME) , HIREDATE , DESIGNATION , SALARY

            INTO V_EID,V_ENAME,V_HIREDATE,V_DESIGNATION ,V_SALARY FROM EMPLOYEE

            WHERE ENO = V_FEID;

         INSERT INTO EMPP VALUES (V_EID , V_ENAME , V_HIREDATE ,V_DESIGNATION ,V_SALARY );

         V_FEID := V_FEID+1;

     END LOOP;

   COMMIT;
   DBMS_OUTPUT.PUT_LINE('ROWS ARE SUCCESSFULLY INSERTED ');
END;
/
```

OUTPUT :

```
ROWS ARE SUCCESSFULLY INSERTED
PL/SQL procedure successfully completed.
```

```
SELECT *
   FROM EMPP;
```

```
       EID ENAME                 HIREDATE  DESIGNATION         SALARY
---------- --------------------- --------- --------------- ---------
      7101 Eugene Sabatini       10-OCT-06 Professor        150000.0
      7102 Samantha Jones        08-NOV-06 Professor        146500.0
      7103 Alexander Lloyd       01-FEB-07 Professor        148000.0
      7104 Simon Downing         01-SEP-07 Professor        138400.0
      7105 Christina Mulboro     15-JUL-08 Asso. Professor  127400.0
      7106 Dolly Silverline      17-AUG-08 Asso. Professor  127400.0
      7107 Christov Plutnik      01-SEP-08 Asso. Professor  127400.0
      7108 Ellena Sanchez        12-NOV-09 Asso. Professor  119700.0
      7109 Martina Jacobson      15-NOV-09 Asst. Professor   91000.0
      7110 William Smithfield    23-JUN-10 Asst. Professor   86400.0
      7111 Albert Greenfield     12-JUL-16 Research Asst.    48200.0
      7112 James Washington      22-AUG-17 Research Asst.    44600.0
      7113 Julia Martin          01-DEC-18 Teaching Asst.    35600.0
      7114 Larry Gomes           18-MAY-19 Teaching Asst.    32850.0
      7115 Svetlana Sanders      15-JAN-20 Teaching Asst.    30000.0
      7116 Lovelyn Brendon       17-JUL-20 Teaching Asst.    30000.0
      7117 Hector Hercules       01-AUG-20 Teaching Asst.    32200.0

17 rows selected.
```

```
*************************************** QUERY-03 ******************************************
Write SQL code to create and execute an anonymous PL/SQL block that will use %ROWTYPE variables
to populate the MENTEE table with corresponding
tuples from Academic Schema.
*******************************************************************************************

CREATE TABLE MENTEE AS
  SELECT SID AS STAFF_NUMBER , NAME AS STAFF_NAME , FNAME||' '||LNAME AS STUDENT_NAME , ROLL
     AS ROLL_NUMBER , REG_DT
      FROM STUDENT JOIN STAFF ON SID = ADVISOR AND 1=2;

Table created.

SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
    FROM USER_CONSTRAINTS
       WHERE TABLE_NAME='MENTEE';

CONSTRAINT_NAME      C
-------------------- -
SYS_C008697          C
SYS_C008698          C
SYS_C008699          C

3 rows selected.


ALTER TABLE MENTEE ADD CONSTRAINT MENTEE_STAFF_NO_ROLL_NO_PK PRIMARY KEY( STAFF_NUMBER,
ROLL_NUMBER);

Table altered.


SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE FROM USER_CONSTRAINTS
    WHERE TABLE_NAME='MENTEE';

CONSTRAINT_NAME      C
-------------------- -
SYS_C008697          C
SYS_C008698          C
SYS_C008699          C
MENTEE_STAFF_NO_ROLL P
_NO_PK
```

```
DECLARE
       MENTEE_DATA MENTEE%ROWTYPE;
BEGIN
       FOR C IN (SELECT SID AS STAFF_NUMBER , NAME AS STAFF_NAME , FNAME||' '||LNAME AS
                 STUDENT_NAME , ROLL AS ROLL_NUMBER , REG_DT  INTO MENTEE_DATA FROM STAFF ,
                 STUDENT WHERE ADVISOR = SID )
       LOOP
             INSERT INTO MENTEE
                VALUES(C.STAFF_NUMBER,C.STAFF_NAME,C.STUDENT_NAME,C.ROLL_NUMBER,C.REG_DT);
       END LOOP;
       COMMIT;
       DBMS_OUTPUT.PUT_LINE('ROWS ARE SUCCESSFULLY INSERTED ');
END;
/
```

OUTPUT :

```
ROWS ARE SUCCESSFULLY INSERTED

PL/SQL procedure successfully completed.
```

** QUERY-04 **

Write SQL code to create and execute an anonymous PL/SQL block that will
display the contents of MENTEE table without using declared variables. You should format the output using RPAD() and/or LPADC), while including proper headers in the result.

****

```
SET SERVEROUTPUT ON;
DECLARE
  V_MEN MENTEE%ROWTYPE;

BEGIN
  DBMS_OUTPUT.PUT_LINE('--------------------------------------------------------------------
  -----------------');
  DBMS_OUTPUT.PUT_LINE(RPAD('STAFF_NAME',22) ||''|| RPAD('STAFF_NUMBER',15) ||''|| RPAD
  ('STUDENT_NAME',20) ||''|| RPAD('ROLL_NUMBER',15) ||''|| RPAD('REG_DT',15));
  DBMS_OUTPUT.PUT_LINE('--------------------------------------------------------------------
  -----------------');
  FOR C IN (SELECT * INTO V_MEN FROM MENTEE ORDER BY ROLL_NUMBER )
  LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(C.STAFF_NAME,22) ||''||RPAD(C.STAFF_NUMBER,15)
      ||''||RPAD(C.STUDENT_NAME,20)||''||RPAD(C.ROLL_NUMBER,15) ||''||RPAD(C.REG_DT,15));
  END LOOP;
```

```
END;
/
```

OUTPUT :

```
-------------------------------------------------------------------------------------
STAFF_NAME             STAFF_NUMBER   STUDENT_NAME        ROLL_NUMBER    REG_DT
-------------------------------------------------------------------------------------
Kamalkant Marathe      101            Aarya Mujumdar      1              08-JAN-21
Adishesh Vidyarthi     102            Aditi Tiwari        2              08-JAN-21
Manishi Singh          103            Anushka Shukla      3              08-JAN-21
Aasawari Deodhar       104            Aparna Jha          4              08-JAN-21
Geetika Goenka         105            Ayushi Soni         5              08-JAN-21
Deo Narayan Mishra     106            Harshada Dhakate    6              08-JAN-21
Sanjeev Banireddy      107            Hrishita Barkhade   7              11-JAN-21
Jasmine Arora          108            Neha Sah            8              11-JAN-21
Vallabh Pai            109            Oshika Roy          9              11-JAN-21
Harneet Khullar        110            Pakhee Mohabansi    10             11-JAN-21
Kamalkant Marathe      101            Prachiti Akre       11             11-JAN-21
Adishesh Vidyarthi     102            Pranjal Mundhada    12             08-JAN-21
Manishi Singh          103            Pranjali Joshi      13             08-JAN-21
Aasawari Deodhar       104            Rajshree Sharma     14             08-JAN-21
Geetika Goenka         105            Rashmi Tiwari       15             08-JAN-21
Deo Narayan Mishra     106            Reema Khandelwal    16             11-JAN-21
Sanjeev Banireddy      107            Riya Jain           17             08-JAN-21
Jasmine Arora          108            Samiksha Anasane    18             08-JAN-21
Vallabh Pai            109            Samiksha Asati      19             08-JAN-21
Harneet Khullar        110            Samruddhi Tatiwar   20             09-JAN-21
Kamalkant Marathe      101            Sanskruti Balpande  21             08-JAN-21
Adishesh Vidyarthi     102            Shradha Mahajan     22             08-JAN-21
Manishi Singh          103            Shristi Gupta       23             08-JAN-21
Aasawari Deodhar       104            Shruti Jain         24             09-JAN-21
Geetika Goenka         105            Srushti Dhakate     25             08-JAN-21
Deo Narayan Mishra     106            Varsha Valecha      26             09-JAN-21
Sanjeev Banireddy      107            Diksha Gupta        27             11-NOV-21
Jasmine Arora          108            Juhie Sayyed        28             11-NOV-21
Vallabh Pai            109            Aaron Rocque        31             08-JAN-21
Harneet Khullar        110            Adwait Warkad       32             08-JAN-21
Kamalkant Marathe      101            Ajay Kumar          33             08-JAN-21
Adishesh Vidyarthi     102            Akshat Surana       34             11-JAN-21
Manishi Singh          103            Amish Singh         35             08-JAN-21
Aasawari Deodhar       104            Arpit Thakur        36             08-JAN-21
Geetika Goenka         105            Bhushan Wanjari     37             11-JAN-21
Deo Narayan Mishra     106            Darshan Chandan     38             08-JAN-21
```

```
Sanjeev Banireddy     107         Divesh Dongare      39             08-JAN-21
Jasmine Arora         108         Gaurav Pathak       40             09-JAN-21
Vallabh Pai           109         Gunjan Nimbalkar    41             08-JAN-21
Harneet Khullar       110         Harsh Sherekar      42             09-JAN-21
Kamalkant Marathe     101         Janak Mandavgade    43             15-JAN-21
Adishesh Vidyarthi    102         Jayesh Jibhakate    44             15-JAN-21
Manishi Singh         103         Jaiwin Chaudhari    45             15-JAN-21
Aasawari Deodhar      104         Kunwar Jora         46             08-JAN-21
Geetika Goenka        105         Kush Munot          47             08-JAN-21
Deo Narayan Mishra    106         Mihir Chowdhury     48             09-JAN-21
Sanjeev Banireddy     107         Mrugal Dudhbade     49             08-JAN-21
Jasmine Arora         108         Nipun Morayya       50             08-JAN-21
Vallabh Pai           109         Piyush Nandha       51             09-JAN-21
Harneet Khullar       110         Prathamesh Gujar    52             08-JAN-21
Kamalkant Marathe     101         Prathamesh Rajbhoj  53             09-JAN-21
Adishesh Vidyarthi    102         Rajesh Thakare      54             08-JAN-21
Manishi Singh         103         Raman Khatod        55             15-JAN-21
Aasawari Deodhar      104         Rishabh Dubey       56             15-JAN-21
Geetika Goenka        105         Rohit Bhojwani      57             15-JAN-21
Deo Narayan Mishra    106         Rushikesh Malu      58             08-JAN-21
Sanjeev Banireddy     107         Sachal Hablani      59             08-JAN-21
Jasmine Arora         108         Sagar Mandal        60             08-JAN-21
Vallabh Pai           109         Sahil Chharra       61             08-JAN-21
Harneet Khullar       110         Saurabh Suchak      62             15-JAN-21
Kamalkant Marathe     101         Shivam Baghele      63             15-JAN-21
Adishesh Vidyarthi    102         Siddharth Shah      64             15-JAN-21
Manishi Singh         103         Sopan Thakre        65             11-JAN-21
Aasawari Deodhar      104         Sudhanshu Pandey    66             09-JAN-21
Geetika Goenka        105         Swyam Laira         67             11-JAN-21
Deo Narayan Mishra    106         Utkarsh Sathawane   68             11-JAN-21
Sanjeev Banireddy     107         Varunpreet Singh    69             15-JAN-21
Jasmine Arora         108         Vedant Khergade     70             15-JAN-21
Vallabh Pai           109         Vikram Kashyap      71             15-JAN-21
Harneet Khullar       110         Vinit Tiwari        72             11-JAN-21
Kamalkant Marathe     101         Yash Agrawal        73             11-JAN-21
Aasawari Deodhar      104         Yash Tekade         74             09-JAN-21
Vallabh Pai           109         Rahul Baser         75             11-NOV-21
Jasmine Arora         108         Yash Wankhedkar     77             10-NOV-21
Harneet Khullar       110         Varun Joshi         78             10-NOV-21
Vallabh Pai           109         Navin Namjoshi      81             13-NOV-21
Harneet Khullar       110         Tushar Tipnis       82             13-NOV-21
PL/SQL procedure successfully completed.
```

************************************** **QUERY-05** **

Write SQL code to create and execute an anonymous PL/SQL block that will display the system date. Use exception (exception VALUE_ERROR) to check if the variable holding the system date is large enough in size. Re-execute the block with appropriate modification to test the exception.

****

```
SET SERVEROUTPUT ON;
DECLARE
     V_DATE CHAR(9);
BEGIN
    SELECT SYSDATE INTO  V_DATE FROM DUAL;
    DBMS_OUTPUT.PUT_LINE('TODAYS DATE IS : '||V_DATE);
END;
/
```

OUTPUT :

```
TODAYS DATE IS : 17-NOV-22

PL/SQL procedure successfully completed.
```

```
SET SERVEROUTPUT ON;
DECLARE
     V_DATE CHAR(3);

BEGIN
    SELECT SYSDATE INTO  V_DATE FROM DUAL;
    DBMS_OUTPUT.PUT_LINE('TODAYS DATE IS : '||V_DATE);

EXCEPTION
  WHEN VALUE_ERROR THEN
     DBMS_OUTPUT.PUT_LINE('VALUE_ERROR OCCURE , VARIABLE IS NOT LARGE ENOUGH TO STORE
DATA');

END;
/
```

OUTPUT :

```
  VALUE_ERROR OCCURE , VARIABLE IS NOT LARGE ENOUGH TO STORE DATA

  PL/SQL procedure successfully completed.
```

*************************************** **QUERY-06** ***

Write SQL code to create and execute an anonymous PL/SQL block that will check (say, for employee number 7108) whether an employee is entitled to receive the longevity bonus. Longevity bonus is given to employees with minimum 12 year of service. Now, re-execute the block to extend longevity bonus to employees with 10 years of service.
****

```
SET SERVEROUTPUT ON;
DECLARE
     V_EMP EMPLOYEE%ROWTYPE;
     V_EMP_ENO EMPLOYEE.ENO%TYPE:=&ENO;
BEGIN
    SELECT * INTO V_EMP FROM EMPLOYEE WHERE ENO = V_EMP_ENO;
    IF((MONTHS_BETWEEN(SYSDATE , V_EMP.HIREDATE) /12) >=12) THEN
        DBMS_OUTPUT.PUT_LINE('BONUS CAN BE GIVEN TO '||V_EMP_ENO);

    ELSE
        DBMS_OUTPUT.PUT_LINE('BONUS CAN NOT BE GIVEN TO '||V_EMP_ENO);

     END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('INVALID DATA');
END;
/
```

OUTPUT :

```
Enter value for eno: 7108

BONUS CAN BE GIVEN TO 7108

PL/SQL procedure successfully completed.
```

```
SET SERVEROUTPUT ON;
DECLARE
     V_EMP EMPLOYEE%ROWTYPE;

BEGIN
    SELECT * INTO V_EMP FROM EMPLOYEE WHERE ENO = 7117;
    IF((MONTHS_BETWEEN(SYSDATE , V_EMP.HIREDATE) /12) =10) THEN
        DBMS_OUTPUT.PUT_LINE('BONUS CAN BE GIVEN TO '||V_EMP.ENO);
    ELSE
        DBMS_OUTPUT.PUT_LINE('BONUS CAN NOT BE GIVEN TO '||V_EMP.ENO);

     END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('INVALID DATA');
END;
/
```

OUTPUT :

```
BONUS CAN NOT BE GIVEN TO 7117

PL/SQL procedure successfully completed.
```

** **QUERY-07** ***
Write SQL code to create and execute an anonymous PL/SQL block that will locate the first August born employee. Re-write and execute an anonymous PL/SQL block that will locate the first August born employee, when EMPLOYEE table is searched in reversed order.
****

```
SET SERVEROUTPUT ON;
DECLARE
    V_EMP EMPLOYEE%ROWTYPE;
BEGIN
  FOR C IN (SELECT * INTO V_EMP
               FROM EMPLOYEE
                 WHERE TO_CHAR(BIRTHDATE,'DD-MON') = '01-AUG')
  LOOP
     DBMS_OUTPUT.PUT_LINE(C.ENO ||' '||C.FNAME);
  END LOOP;
END;
/
```

OUTPUT :

```
PL/SQL procedure successfully completed.
```

```
SET SERVEROUTPUT ON;
DECLARE
  V_EMP EMPLOYEE%ROWTYPE;
```

```
BEGIN
        FOR C IN (SELECT * INTO V_EMP FROM EMPLOYEE WHERE TO_CHAR(BIRTHDATE,'DD-MON') = '01-
        AUG'ORDER BY ENO DESC)
        LOOP
        DBMS_OUTPUT.PUT_LINE(C.ENO ||' '||C.FNAME);
        END LOOP;

EXCEPTION
        WHEN NO_DATA_FOUND THEN
           DBMS_OUTPUT.PUT_LINE('INVALID DATA');
END;
/
```

OUTPUT :

```
PL/SQL procedure successfully completed.
```

*************************************** **QUERY-08** **
Write SQL code to create and execute an anonymous PL/SQL block that accept staff ID from the console and will display staff details for said staff. A system exception, NO_DATA_FOUND should be cached when the mentioned staff does not exist.
****

```
SET SERVEROUTPUT ON;
DECLARE
     V_STAFF  NUMBER;
     V_STAFF_SID  NUMBER := &SID;
BEGIN
    SELECT COUNT(*) INTO V_STAFF FROM STAFF WHERE SID = V_STAFF_SID;

    IF(V_STAFF = 0) THEN
        RAISE NO_DATA_FOUND;
    ELSE
        DBMS_OUTPUT.PUT_LINE('DATA FOUND');
    END IF;

 EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('STAFF MEMBER DOES NOT EXIST');
END;
/
```

OUTPUT :

```
Enter value for sid: 103
DATA FOUND

PL/SQL procedure successfully completed.


Enter value for sid: 7011
STAFF MEMBER DOES NOT EXIST

PL/SQL procedure successfully completed.
```

*************************************** **QUERY-09** ***
Write SQL code to create and execute an anonymous PL/SQL block that defines user-defined exceptions - BELOW_PAY_RANGE and ABOVE_PAY_RANGE. Your script should accept an employee number from the console and check for the salary to fall within the payscale [minpay, maxpay]. If the salary is less than minpay, BELOW_PAY_RANGE exception is raised and when cached an appropriate message -
'<EmpNo> Receives Salary Below Scale [minpay, maxpay]'
is displayed; otherwise ABOVE_PAY_RANGE exception is raised and cached to display the appropriate message accordingly. You must appropriately catch the NO_DATA_FOUND exception also. When there are no violations, display for the employee the salary drawn. Test the above anonymous block for input employee numbers - 7101, 7104, 7106, 7109, 7111, 7114 and 7117.
****

```
SET SERVEROUTPUT ON;
DECLARE
   V_EMPOLYEE EMPLOYEE%ROWTYPE;

   V_PAYSCALE PAYSCALE%ROWTYPE;

   V_IP_EMPLOYEE  EMPLOYEE.ENO%TYPE := &EMPOLYEE_ID;

   ABOVE_PAY_RANGE  EXCEPTION ;

   BELOW_PAY_RANGE EXCEPTION;


BEGIN
   SELECT *  INTO V_EMPOLYEE
       FROM EMPLOYEE
          WHERE EMPLOYEE.ENO = V_IP_EMPLOYEE;

   SELECT * INTO V_PAYSCALE
        FROM PAYSCALE
```

```
         WHERE PAYSCALE.DESIGNATION =  V_EMPOLYEE.DESIGNATION;
   IF(V_EMPOLYEE.SALARY <  V_PAYSCALE.MINPAY) THEN
       RAISE BELOW_PAY_RANGE;

   ELSIF(V_EMPOLYEE.SALARY >  V_PAYSCALE.MAXPAY) THEN
       RAISE ABOVE_PAY_RANGE;

   ELSE
      DBMS_OUTPUT.PUT_LINE(V_IP_EMPLOYEE ||' '|| 'RECEIVED IN GIVEN PAY_RANGE [' ||'
         '||V_EMPOLYEE.SALARY||' '|| ']');
   END IF;

EXCEPTION
   WHEN ABOVE_PAY_RANGE THEN
         DBMS_OUTPUT.PUT_LINE(V_IP_EMPLOYEE ||' '|| 'RECEIVED ABOVE PAY_RANGE [' ||'
        '||V_PAYSCALE.MINPAY||','||V_PAYSCALE.MAXPAY || ' ]');

   WHEN BELOW_PAY_RANGE THEN
       DBMS_OUTPUT.PUT_LINE(V_IP_EMPLOYEE ||' '|| 'RECEIVED BELOW PAY_RANGE [' ||'
       '||V_PAYSCALE.MINPAY||','|| V_PAYSCALE.MAXPAY||' ]');

   WHEN NO_DATA_FOUND THEN
       DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND');

END;
/
```

OUTPUT :

```
Enter value for empolyee_id: 7101
7101 RECEIVED IN GIVEN PAY_RANGE [ 150000 ]
PL/SQL procedure successfully completed.

Enter value for empolyee_id: 7104
7104 RECEIVED BELOW PAY_RANGE [ 140000,200000 ]
PL/SQL procedure successfully completed.

Enter value for empolyee_id: 7106
7106 RECEIVED IN GIVEN PAY_RANGE [ 127400 ]
PL/SQL procedure successfully completed.

Enter value for empolyee_id: 7109
```

```
7109 RECEIVED ABOVE PAY_RANGE [ 50000,90000 ]
PL/SQL procedure successfully completed.

Enter value for empolyee_id: 7111
7111 RECEIVED ABOVE PAY_RANGE [ 30000,45000 ]
PL/SQL procedure successfully completed.

Enter value for empolyee_id: 7114
7114 RECEIVED ABOVE PAY_RANGE [ 20000,32500 ]
PL/SQL procedure successfully completed.

Enter value for empolyee_id: 7117
7117 RECEIVED IN GIVEN PAY_RANGE [ 32200 ]
PL/SQL procedure successfully completed.
```

```
*************************************** QUERY-10 ******************************************
 Write a SQL code to create and execute an anonymous PL/SQL block that will
 modify Query-09 to process all records of EMPLOYEE table. You need not acquire employee
 number from console. You should only report the violations.
*********************************************************************************************
```

```
SET SERVEROUTPUT ON;
DECLARE
  V_EMPOLYEE EMPLOYEE%ROWTYPE;
  V_PAYSCALE PAYSCALE%ROWTYPE;
  ABOVE_PAY_RANGE  EXCEPTION ;
  BELOW_PAY_RANGE EXCEPTION;
  C EMPLOYEE%ROWTYPE;

BEGIN

  FOR C IN (SELECT *  INTO V_EMPOLYEE FROM EMPLOYEE ) LOOP
    BEGIN
      SAVEPOINT S ;
        SELECT * INTO V_PAYSCALE FROM PAYSCALE WHERE PAYSCALE.DESIGNATION =  C.DESIGNATION;

      IF(C.SALARY <  V_PAYSCALE.MINPAY) THEN
          RAISE BELOW_PAY_RANGE;
      END IF;

      IF(C.SALARY >  V_PAYSCALE.MAXPAY) THEN
```

```
          RAISE ABOVE_PAY_RANGE;
      END IF;

EXCEPTION

  WHEN ABOVE_PAY_RANGE THEN
     DBMS_OUTPUT.PUT_LINE(C.ENO||' '|| 'RECEIVED ABOVE PAY_RANGE [' ||'
     '||V_PAYSCALE.MINPAY||','|| V_PAYSCALE.MAXPAY||' ]');
         ROLLBACK TO SAVEPOINT S;

  WHEN BELOW_PAY_RANGE THEN
    DBMS_OUTPUT.PUT_LINE(C.ENO||' '|| 'RECEIVED BELOW PAY_RANGE [' ||'
    '||V_PAYSCALE.MINPAY||','|| V_PAYSCALE.MAXPAY||' ]');
       ROLLBACK TO SAVEPOINT S;

  WHEN NO_DATA_FOUND THEN
     DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND');
   END;
 END LOOP;
END;
/
```

OUTPUT :

```
7104 RECEIVED BELOW PAY_RANGE [ 140000,200000 ]
7109 RECEIVED ABOVE PAY_RANGE [ 50000,90000 ]
7111 RECEIVED ABOVE PAY_RANGE [ 30000,45000 ]
7113 RECEIVED ABOVE PAY_RANGE [ 20000,32500 ]
7114 RECEIVED ABOVE PAY_RANGE [ 20000,32500 ]

PL/SQL procedure successfully completed.
```

==

INFERENCES OF THE EXPERIMENT

==

Hence , we have successfully write and execute PL/SQL blocks (with exception handling) including PL/SQL subprograms using Oracle 11g.