```
SQL> SET FEEDBACK ON;
SQL> SET SERVEROUTPUT ON;
```

01

```
SQL> CREATE TABLE TEST_TBL(
  2  REC_NO NUMBER(3),
  3  CURR_DT DATE,
  4  CONSTRAINT TEST_TBL_PK PRIMARY KEY (REC_NO),
  5  CONSTRAINT TEST_TBL_CHK_RANGE CHECK (REC_NO BETWEEN 101 AND 999)
  6  );

Table created.

SQL>
SQL> DECLARE
  2        BASE_CNT CONSTANT INT :=100;
  3        CNT INT;
  4  BEGIN
  5        FOR CNT IN 1 .. 10 LOOP
  6              INSERT INTO TEST_TBL(REC_NO,CURR_DT)
  7                    VALUES(BASE_CNT+CNT,SYSDATE);
  8        END LOOP;
  9    COMMIT;
 10  END;
 11  /

PL/SQL procedure successfully completed.

SQL> SELECT * FROM TEST_TBL;

    REC_NO CURR_DT
---------- ---------
       101 30-MAR-17
       102 30-MAR-17
       103 30-MAR-17
       104 30-MAR-17
       105 30-MAR-17
       106 30-MAR-17
       107 30-MAR-17
       108 30-MAR-17
       109 30-MAR-17
       110 30-MAR-17

10 rows selected.

SQL>
SQL> CREATE TABLE EMPP AS SELECT EID,FNAME||' '||LNAME AS ENAME,HIREDATE,SALARY
FROM EMPLOYEE WHERE 1=2;
CREATE TABLE EMPP AS SELECT EID,FNAME||' '||LNAME AS ENAME,HIREDATE,SALARY FROM
EMPLOYEE WHERE 1=2
             *
ERROR at line 1:
ORA-00955: name is already used by an existing object
```

```
SQL> DESC EMPP;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EID                                                NUMBER(4)
 ENAME                                              VARCHAR2(21)
 HIREDATE                                  NOT NULL DATE
 SALARY                                    NOT NULL NUMBER(7,2)

SQL> SELECT COUNT(*) FROM EMPP;

  COUNT(*)
----------
        10

1 row selected.

SQL> SELECT CONSTRAINT_NAME FROM USER_CONSTRAINTS WHERE TABLE_NAME LIKE 'EMPP';

CONSTRAINT_NAME
------------------------------
SYS_C0011231
SYS_C0011232

2 rows selected.
```

02

```
SQL>
SQL> DECLARE
  2  EMPID EMPLOYEE.EID%TYPE;
  3  ENAME VARCHAR2(40);
  4  HDATE EMPLOYEE.HIREDATE%TYPE;
  5  SAL EMPLOYEE.SALARY%TYPE;
  6  CNT INT;
  7  LCNT INT;
  8  BASE_CNT CONSTANT INT :=7100;
  9  BEGIN
 10  SELECT COUNT(*) INTO CNT FROM EMPLOYEE;
 11  FOR LCNT IN 1..CNT LOOP
 12     SELECT EID,(FNAME||' '||LNAME),HIREDATE,SALARY INTO
 13     EMPID,ENAME,HDATE,SAL FROM EMPLOYEE
 14     WHERE EID=BASE_CNT+LCNT;
 15     INSERT INTO EMPP VALUES(EMPID,ENAME,HDATE,SAL);
 16  END LOOP;
 17  END;
 18  /

PL/SQL procedure successfully completed.

SQL> SELECT * FROM EMPP;

       EID ENAME                HIREDATE      SALARY
---------- -------------------- --------- ----------
      7101 Samantha Jones       08-NOV-94      16500
      7102 Albert Greenfield    12-JUL-98      14200
      7103 Julia Martin         01-DEC-99      13320
      7104 Martina Jacobson     15-NOV-96      15550
      7105 Alexander Lloyd      01-FEB-94      17500
      7106 William Smithfield   23-JUN-96      15660
      7107 Eugene Sabatini      10-OCT-94      16500
      7108 James Washington     22-AUG-98      14000
      7109 Larry Gomes          18-MAY-99      13650
      7110 Svetlana Sanders     15-JAN-06      10000
      7101 Samantha Jones       08-NOV-94      16500

       EID ENAME                HIREDATE      SALARY
---------- -------------------- --------- ----------
      7102 Albert Greenfield    12-JUL-98      14200
      7103 Julia Martin         01-DEC-99      13320
      7104 Martina Jacobson     15-NOV-96      15550
      7105 Alexander Lloyd      01-FEB-94      17500
      7106 William Smithfield   23-JUN-96      15660
      7107 Eugene Sabatini      10-OCT-94      16500
      7108 James Washington     22-AUG-98      14000
      7109 Larry Gomes          18-MAY-99      13650
      7110 Svetlana Sanders     15-JAN-06      10000

20 rows selected.

SQL> ROLLBACK;

Rollback complete.
```

03

```
SQL> DECLARE
  2  EMP_REC EMPLOYEE%ROWTYPE;
  3  CNT INT;
  4  LCNT INT;
  5  BASE_CNT CONSTANT INT :=7100;
  6  BEGIN
  7  SELECT COUNT(*) INTO CNT FROM EMPLOYEE;
  8  FOR LCNT IN 1..CNT LOOP
  9    SELECT * INTO EMP_REC FROM EMPLOYEE WHERE EID=BASE_CNT+LCNT;
 10    INSERT INTO EMPP VALUES(EMP_REC.EID,EMP_REC.FNAME||'
'||EMP_REC.LNAME,EMP_REC.HIREDATE,EMP_REC.SALARY);
 11  END LOOP;
 12  END;
 13  /

PL/SQL procedure successfully completed.

SQL> SELECT * FROM EMPP;

       EID ENAME                HIREDATE      SALARY
---------- -------------------- --------- ----------
      7101 Samantha Jones       08-NOV-94      16500
      7102 Albert Greenfield    12-JUL-98      14200
      7103 Julia Martin         01-DEC-99      13320
      7104 Martina Jacobson     15-NOV-96      15550
      7105 Alexander Lloyd      01-FEB-94      17500
      7106 William Smithfield   23-JUN-96      15660
      7107 Eugene Sabatini      10-OCT-94      16500
      7108 James Washington     22-AUG-98      14000
      7109 Larry Gomes          18-MAY-99      13650
      7110 Svetlana Sanders     15-JAN-06      10000
      7101 Samantha Jones       08-NOV-94      16500

       EID ENAME                HIREDATE      SALARY
---------- -------------------- --------- ----------
      7102 Albert Greenfield    12-JUL-98      14200
      7103 Julia Martin         01-DEC-99      13320
      7104 Martina Jacobson     15-NOV-96      15550
      7105 Alexander Lloyd      01-FEB-94      17500
      7106 William Smithfield   23-JUN-96      15660
      7107 Eugene Sabatini      10-OCT-94      16500
      7108 James Washington     22-AUG-98      14000
      7109 Larry Gomes          18-MAY-99      13650
      7110 Svetlana Sanders     15-JAN-06      10000

20 rows selected.
```

04

```
SQL> --VAL IS BY DEFAULT ASSUMING A ROWTYPE
SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE('EID       ENAME            HIREDATE      SALARY ');
  3  FOR VAL IN(SELECT EID,ENAME,HIREDATE,SALARY FROM EMPP) LOOP
  4    DBMS_OUTPUT.PUT_LINE(RPAD(VAL.EID,8,' ')||' '||RPAD(VAL.ENAME,20,' ')||'
'||RPAD(VAL.HIREDATE,12,' ')||' '||LPAD(VAL.SALARY,6,' '));
  5  END LOOP;
  6  END;
  7  /
EID      ENAME                HIREDATE      SALARY
7101     Samantha Jones       08-NOV-94     16500
7102     Albert Greenfield    12-JUL-98     14200
7103     Julia Martin         01-DEC-99     13320
7104     Martina Jacobson     15-NOV-96     15550
7105     Alexander Lloyd      01-FEB-94     17500
7106     William Smithfield   23-JUN-96     15660
7107     Eugene Sabatini      10-OCT-94     16500
7108     James Washington     22-AUG-98     14000
7109     Larry Gomes          18-MAY-99     13650
7110     Svetlana Sanders     15-JAN-06     10000
7101     Samantha Jones       08-NOV-94     16500
7102     Albert Greenfield    12-JUL-98     14200
7103     Julia Martin         01-DEC-99     13320
7104     Martina Jacobson     15-NOV-96     15550
7105     Alexander Lloyd      01-FEB-94     17500
7106     William Smithfield   23-JUN-96     15660
7107     Eugene Sabatini      10-OCT-94     16500
7108     James Washington     22-AUG-98     14000
7109     Larry Gomes          18-MAY-99     13650
7110     Svetlana Sanders     15-JAN-06     10000
```

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

05

```
SQL> SELECT BANNER FROM V$VERSION;

BANNER
--------------------------------------------------------------------------------
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
PL/SQL Release 11.2.0.1.0 - Production
CORE   11.2.0.1.0   Production
TNS for 64-bit Windows: Version 11.2.0.1.0 - Production
NLSRTL Version 11.2.0.1.0 - Production

5 rows selected.

SQL>
SQL>
SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE(SYSTIMESTAMP);
  3  DBMS_OUTPUT.PUT_LINE(SYSTIMESTAMP- interval '3' hour);
  4  END;
  5  /
30-MAR-17 01.58.20.167000000 AM +05:30
29-MAR-17 10.58.20.167000000 PM +05:30

PL/SQL procedure successfully completed.
```

06

```
SQL> DECLARE
  2  D VARCHAR2(5);
  3  BEGIN
  4  D:=TO_CHAR(SYSDATE);
  5  DBMS_OUTPUT.PUT_LINE(D);
  6  EXCEPTION
  7  WHEN VALUE_ERROR THEN
  8  DBMS_OUTPUT.PUT_LINE('VALUE ERROR OCCURED');
  9  END;
 10  /
VALUE ERROR OCCURED

PL/SQL procedure successfully completed.

SQL> DECLARE
  2  D VARCHAR2(50);
  3  BEGIN
  4  D:=TO_CHAR(SYSDATE);
  5  DBMS_OUTPUT.PUT_LINE(D);
  6  EXCEPTION
  7  WHEN VALUE_ERROR THEN
  8  DBMS_OUTPUT.PUT_LINE(' VALUE ERROR OCCURED');
  9  END;
 10  /
30-MAR-17
```

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

07

```
SQL>
SQL> DECLARE
  2  EMPREC EMPLOYEE%ROWTYPE;
  3  BASE_CNT CONSTANT INT :=7100;
  4  CYEAR NUMBER;
  5  HYEAR NUMBER;
  6  YEARDIFF NUMBER;
  7  CNT INT;
  8  LCNT INT;
  9  YEARCNT INT :=20;
 10  BEGIN
 11  SELECT EXTRACT(YEAR FROM SYSDATE) INTO CYEAR FROM DUAL;
 12  SELECT COUNT(*) INTO CNT FROM EMPLOYEE;
 13  <<TWICELOOPER>>
 14  DBMS_OUTPUT.PUT_LINE('EMPLOYEE WORKING WITH COMPANY FOR ATLEAST '||YEARCNT||'
YEARS');
 15  FOR LCNT IN 1 .. CNT LOOP
 16     SELECT * INTO EMPREC FROM EMPLOYEE WHERE EID=BASE_CNT+LCNT;
 17     SELECT EXTRACT(YEAR FROM EMPREC.HIREDATE) INTO HYEAR FROM DUAL;
 18     YEARDIFF:=CYEAR-HYEAR;
 19     IF YEARDIFF >=YEARCNT THEN
 20        DBMS_OUTPUT.PUT_LINE(EMPREC.EID||' '||EMPREC.FNAME||' '||EMPREC.LNAME);
 21     END IF;
 22  END LOOP;
 23  IF YEARCNT=20 THEN
 24     YEARCNT:=15;
 25     GOTO TWICELOOPER;
 26  END IF;
 27  END;
 28
 29  /
EMPLOYEE WORKING WITH COMPANY FOR ATLEAST 20 YEARS
7101 Samantha Jones
7104 Martina Jacobson
7105 Alexander Lloyd
7106 William Smithfield
7107 Eugene Sabatini
EMPLOYEE WORKING WITH COMPANY FOR ATLEAST 15 YEARS
7101 Samantha Jones
7102 Albert Greenfield
7103 Julia Martin
7104 Martina Jacobson
7105 Alexander Lloyd
7106 William Smithfield
7107 Eugene Sabatini
7108 James Washington
7109 Larry Gomes

PL/SQL procedure successfully completed.
```

08

```
SQL> DECLARE
  2  EMP_REC EMPLOYEE%ROWTYPE;
  3  CNT INT;
  4  LCNT INT;
  5  MNTH INT;
  6  BASE_CNT CONSTANT INT :=7100;
  7  BEGIN
  8  SELECT COUNT(*) INTO CNT FROM EMPLOYEE;
  9  FOR LCNT IN 1..CNT LOOP
 10    SELECT * INTO EMP_REC FROM EMPLOYEE WHERE EID=BASE_CNT+LCNT;
 11    SELECT EXTRACT(MONTH FROM EMP_REC.BIRTHDATE) INTO MNTH FROM DUAL;
 12    IF MNTH=11 THEN
 13       DBMS_OUTPUT.PUT_LINE(EMP_REC.FNAME||' '||EMP_REC.LNAME|| '
'||EMP_REC.BIRTHDATE);
 14       EXIT;
 15    END IF;
 16  END LOOP;
 17  END;
 18  /
William Smithfield 01-NOV-72

PL/SQL procedure successfully completed.
```

09

```
SQL> DECLARE
  2  EMP_REC EMPLOYEE%ROWTYPE;
  3  CNT INT;
  4  LCNT INT;
  5  MNTH INT;
  6  BASE_CNT CONSTANT INT :=7100;
  7  BEGIN
  8  SELECT COUNT(*) INTO CNT FROM EMPLOYEE;
  9  FOR LCNT IN REVERSE 1..CNT LOOP
 10    SELECT * INTO EMP_REC FROM EMPLOYEE WHERE EID=BASE_CNT+LCNT;
 11    SELECT EXTRACT(MONTH FROM EMP_REC.BIRTHDATE) INTO MNTH FROM DUAL;
 12    IF MNTH=11 THEN
 13       DBMS_OUTPUT.PUT_LINE(EMP_REC.FNAME||' '||EMP_REC.LNAME|| '
'||EMP_REC.BIRTHDATE);
 14       EXIT;
 15    END IF;
 16  END LOOP;
 17  END;
 18  /
Eugene Sabatini 09-NOV-73

PL/SQL procedure successfully completed.
```

10

```
SQL> DECLARE
  2  EMPID EMPLOYEE.EID%TYPE;
  3  EREC EMPLOYEE%ROWTYPE;
  4  BEGIN
  5  DBMS_OUTPUT.PUT_LINE('ENTER EMPLOYEE ID:');
  6  EMPID:='&EMPID';
  7  SELECT * INTO EREC FROM EMPLOYEE WHERE EID=EMPID;
  8  DBMS_OUTPUT.PUT_LINE(EREC.EID||'  '||EREC.LNAME||'  '||EREC.DESIGNATION||'
'||EREC.SALARY);
  9  EXCEPTION
 10    WHEN NO_DATA_FOUND THEN
 11     DBMS_OUTPUT.PUT_LINE('NO RECORD EXIST WITH EID:='||EMPID);
 12  END;
 13  /
Enter value for empid: 7101
old   6: EMPID:='&EMPID';
new   6: EMPID:='7101';
ENTER EMPLOYEE ID:
7101  Jones  Professor 16500

PL/SQL procedure successfully completed.
```

11

```
SQL> CREATE TABLE PAYSCALE(
  2  DESIGNATION VARCHAR(15),
  3  MINPAY NUMBER(5),
  4  MAXPAY NUMBER(5),
  5  CONSTRAINT PAYSCALE_PK PRIMARY KEY (DESIGNATION),
  6  CONSTRAINT PAYSCALE_CHK_DESIGNATION CHECK (DESIGNATION IN('Professor','Sr.
Lecturer','Lecturer','Asst. Professor'))
  7  );

Table created.

SQL>
SQL> INSERT INTO PAYSCALE VALUES('Lecturer', 12000, 13500);

1 row created.

SQL> INSERT INTO PAYSCALE VALUES('Sr. Lecturer', 13000, 15000);

1 row created.

SQL> INSERT INTO PAYSCALE VALUES('Asst. Professor', 14500, 16500);

1 row created.

SQL> INSERT INTO PAYSCALE VALUES('Professor', 16000, 19000);

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
SQL> DECLARE
  2  BELOW_PAY_RANGE EXCEPTION;
  3  ABOVE_PAY_RANGE EXCEPTION;
  4  EMPID EMPLOYEE.EID%TYPE;
  5  EXPNO EMPLOYEE.EID%TYPE;
  6  EREC EMPLOYEE%ROWTYPE;
  7  PAYSCALEREC PAYSCALE%ROWTYPE;
  8  EXPMINPAY PAYSCALE.MINPAY%TYPE;
  9  EXPMAXPAY PAYSCALE.MAXPAY%TYPE;
 10  BEGIN
 11  DBMS_OUTPUT.PUT_LINE('ENTER EID OF THE EMPLOYEE:');
 12  EMPID:='&EMPID';
 13  SELECT * INTO EREC FROM EMPLOYEE WHERE EID=EMPID;
 14  SELECT * INTO PAYSCALEREC FROM PAYSCALE WHERE DESIGNATION=EREC.DESIGNATION;
 15       EXPNO:=EREC.EID;
 16       EXPMINPAY:=PAYSCALEREC.MINPAY;
 17       EXPMAXPAY:=PAYSCALEREC.MAXPAY;
 18  IF EREC.SALARY >PAYSCALEREC.MINPAY THEN
 19     IF EREC.SALARY < PAYSCALEREC.MAXPAY THEN
 20       DBMS_OUTPUT.PUT_LINE(EREC.EID||' RECEIVES SALARY IN SCALE
['||PAYSCALEREC.MINPAY||','||PAYSCALEREC.MAXPAY||']');
 21     ELSE
 22       RAISE ABOVE_PAY_RANGE;
 23     END IF;
 24  ELSE
 25     RAISE BELOW_PAY_RANGE;
 26  END IF;
 27  EXCEPTION
 28    WHEN BELOW_PAY_RANGE THEN
 29       DBMS_OUTPUT.PUT_LINE(EXPNO||' Receives Salary Below Scale
['||EXPMINPAY||','||EXPMAXPAY||']');
 30    WHEN ABOVE_PAY_RANGE THEN
 31       DBMS_OUTPUT.PUT_LINE(EXPNO||' Receives Salary Above Scale
['||EXPMINPAY||','||EXPMAXPAY||']');
 32    WHEN NO_DATA_FOUND THEN
 33       DBMS_OUTPUT.PUT_LINE('NO RECORDS FOUND WITH EID:='||EXPNO);
 34    WHEN OTHERS THEN
 35       DBMS_OUTPUT.PUT_LINE('SOMETHING NOT
CORRECT'||TO_CHAR(SQLCODE)||'::'||TO_CHAR(SQLERRM));
 36  END;
 37  /
Enter value for empid: 7110
old  12: EMPID:='&EMPID';
new  12: EMPID:='7110';
ENTER EID OF THE EMPLOYEE:
7110 Receives Salary Below Scale [12000,13500]

PL/SQL procedure successfully completed.
```

12

```
SQL> DECLARE
  2  EMPID EMPLOYEE.EID%TYPE;
  3  EXPNO EMPLOYEE.EID%TYPE;
  4  EREC EMPLOYEE%ROWTYPE;
  5  PAYSCALEREC PAYSCALE%ROWTYPE;
  6  EXPMINPAY PAYSCALE.MINPAY%TYPE;
  7  EXPMAXPAY PAYSCALE.MAXPAY%TYPE;
```

```
CNT INT;
LCNT INT;
BASE_CNT CONSTANT INT :=7100;
BEGIN
SELECT COUNT(*) INTO CNT FROM EMPLOYEE;
FOR LCNT IN 1..CNT LOOP
EMPID:=BASE_CNT+LCNT;
SELECT * INTO EREC FROM EMPLOYEE WHERE EID=EMPID;
SELECT * INTO PAYSCALEREC FROM PAYSCALE WHERE DESIGNATION=EREC.DESIGNATION;
    EXPNO:=EREC.EID;
    EXPMINPAY:=PAYSCALEREC.MINPAY;
    EXPMAXPAY:=PAYSCALEREC.MAXPAY;
IF EREC.SALARY >PAYSCALEREC.MINPAY THEN
  IF EREC.SALARY < PAYSCALEREC.MAXPAY THEN
    DBMS_OUTPUT.PUT_LINE(EREC.EID||' RECEIVES SALARY IN SCALE ['||PAYSCALEREC.MINPAY||','||PAYSCALEREC.MAXPAY||']');
  ELSE
    DBMS_OUTPUT.PUT_LINE(EXPNO||' Receives Salary Above Scale ['||EXPMINPAY||','||EXPMAXPAY||']');
  END IF;
ELSE
  DBMS_OUTPUT.PUT_LINE(EXPNO||' Receives Salary Below Scale ['||EXPMINPAY||','||EXPMAXPAY||']');
END IF;
END LOOP;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('NO RECORDS FOUND WITH EID:='||EXPNO);
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('SOMETHING NOT CORRECT'||TO_CHAR(SQLCODE)||'::'||TO_CHAR(SQLERRM));
END;
/
7101 RECEIVES SALARY IN SCALE [16000,19000]
7102 Receives Salary Below Scale [14500,16500]
7103 RECEIVES SALARY IN SCALE [12000,13500]
7104 RECEIVES SALARY IN SCALE [14500,16500]
7105 RECEIVES SALARY IN SCALE [16000,19000]
7106 RECEIVES SALARY IN SCALE [14500,16500]
7107 RECEIVES SALARY IN SCALE [16000,19000]
7108 RECEIVES SALARY IN SCALE [13000,15000]
7109 Receives Salary Above Scale [12000,13500]
7110 Receives Salary Below Scale [12000,13500]

PL/SQL procedure successfully completed.

SQL> SET FEEDBACK OFF;
SQL> SET ECHO OFF;
```