```
SQL> SET VERIFY OFF
SQL> SET AUTOTRACE ON EXPLAIN
SQL> cl scr

SQL> DECLARE
  2    V_Empno Emp.Empno%TYPE := &Empno;
  3    V_Ename Emp.Ename%TYPE;
  4    V_Sal Emp.Sal%TYPE;
  5  BEGIN
  6    SELECT Ename, Sal INTO V_Ename, V_Sal
  7    FROM Emp
  8    WHERE Empno = V_Empno;
  9    DBMS_OUTPUT.PUT_LINE('Name : '||V_Ename||' Salary : '||V_Sal);
 10  END;
 11  /
Enter value for empno: 7839

PL/SQL procedure successfully completed.

SQL> SET SERVEROUTPUT On
SQL> /
Enter value for empno: 7839
Name : KING Salary : 5000

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 7654
Name : MARTIN Salary : 1250

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1234
DECLARE
*
ERROR at line 1:
ORA-01403: no data found
ORA-06512: at line 6


SQL> DECLARE
  2    V_Empno Emp.Empno%TYPE := &Empno;
  3    V_Ename Emp.Ename%TYPE;
  4    V_Sal Emp.Sal%TYPE;
  5  BEGIN
  6    SELECT Ename, Sal INTO V_Ename, V_Sal
  7    FROM Emp
  8    WHERE Empno = V_Empno;
  9    DBMS_OUTPUT.PUT_LINE('Name : '||V_Ename||' Salary : '||V_Sal);
 10  EXCEPTION
 11  WHEN NO_DATA_FOUND THEN
 12  DBMS_OUTPUT.PUT_LINE('Sorry, Data is Not Found.');
 13  END;
 14  /
```

```
Enter value for empno: 1234
Sorry, Data is Not Found.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2    V_Emp Emp%ROWTYPE;
  3    V_Sal Emp.Sal%TYPE := &Sal;
  4  BEGIN
  5    SELECT * INTO V_Emp
  6    FROM Emp
  7    WHERE Sal = V_Sal;
  8    DBMS_OUTPUT.PUT_LINE('Name : '||V_Emp.Ename||' Salary : '||V_Emp.Sal);
  9  END ;
 10  /
Enter value for sal: 800
Name : SMITH Salary : 800

PL/SQL procedure successfully completed.

SQL> /
Enter value for sal: 5000
Name : KING Salary : 5000

PL/SQL procedure successfully completed.

SQL> /
Enter value for sal: 3000
DECLARE
*
ERROR at line 1:
ORA-01422: exact fetch returns more than requested number of rows
ORA-06512: at line 5


SQL> DECLARE
  2    V_Emp Emp%ROWTYPE;
  3    V_Sal Emp.Sal%TYPE := &Sal;
  4  BEGIN
  5    SELECT * INTO V_Emp
  6    FROM Emp
  7    WHERE Sal = V_Sal;
  8    DBMS_OUTPUT.PUT_LINE('Name : '||V_Emp.Ename||' Salary : '||V_Emp.Sal);
  9   EXCEPTION
 10    WHEN TOO_MANY_ROWS THEN
 11      DBMS_OUTPUT.PUT_LINE('More Than One Employee Having Same Salary');
 12  END ;
 13  /
Enter value for sal: 3000
More Than One Employee Having Same Salary

PL/SQL procedure successfully completed.
```

```
SQL> /
Enter value for sal: 1234
DECLARE
*
ERROR at line 1:
ORA-01403: no data found
ORA-06512: at line 5


SQL> DECLARE
  2    V_Emp Emp%ROWTYPE;
  3    V_Sal Emp.Sal%TYPE := &Sal;
  4  BEGIN
  5    SELECT * INTO V_Emp
  6    FROM Emp
  7    WHERE Sal = V_Sal;
  8    DBMS_OUTPUT.PUT_LINE('Name : '||V_Emp.Ename||' Salary : '||V_Emp.Sal);
  9   EXCEPTION
 10    WHEN TOO_MANY_ROWS THEN
 11      DBMS_OUTPUT.PUT_LINE('More Than One Employee Having Same Salary');
 12      WHEN NO_DATA_FOUND THEN
 13  DBMS_OUTPUT.PUT_LINE('Sorry, Data is Not Found.');
 14  END ;
 15  /
Enter value for sal: 3000
More Than One Employee Having Same Salary

PL/SQL procedure successfully completed.

SQL> /
Enter value for sal: 1234
Sorry, Data is Not Found.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2    V_Empno VARCHAR2(4):='&Empno';
  3    V_Ename VARCHAR2(20):='&Ename';
  4    V_Deptno VARCHAR2(2):='&Deptno';
  5  BEGIN
  6    INSERT INTO Emp(Empno,Ename,Deptno)
  7    VALUES (V_Empno,V_Ename,V_Deptno);
  8  EXCEPTION
  9    WHEN INVALID_NUMBER THEN
 10       DBMS_OUTPUT.PUT_LINE('Given Employee Number or Department Number is
Invalid');
 11  END ;
 12  /
Enter value for empno: 1234
Enter value for ename: SAMPLE
Enter value for deptno: 30

PL/SQL procedure successfully completed.
```

```
SQL> /
Enter value for empno: 123A
Enter value for ename: SAMPLE
Enter value for deptno: 30
Given Employee Number or Department Number is Invalid

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> ROLLBACK;

Rollback complete.

SQL> cl scr

SQL> DECLARE
  2    V_Num1 NUMBER;
  3  BEGIN
  4    V_Num1 := '&GiveNumber1' + '&GiveNumber2';
  5    DBMS_OUTPUT.PUT_LINE ('The Result of the Operation is: ' || V_Num1);
  6  EXCEPTION
  7    WHEN VALUE_ERROR THEN
  8      DBMS_OUTPUT.PUT_LINE ('Please Check - There is a Source of Invalid
Values in your input (OR) Operations.');
  9  END ;
 10  /
Enter value for givenumber1: 1234
Enter value for givenumber2: 2345
The Result of the Operation is: 3579

PL/SQL procedure successfully completed.

SQL> /
Enter value for givenumber1: 123A
Enter value for givenumber2: 2345
Please Check - There is a Source of Invalid Values in your input (OR)
Operations.

PL/SQL procedure successfully completed.

SQL> ED
Wrote file afiedt.buf

  1  DECLARE
  2    V_Num1 NUMBER;
  3  BEGIN
  4    V_Num1 := '&GiveNumber1' + '&GiveNumber2';
  5    DBMS_OUTPUT.PUT_LINE ('The Result of the Operation is: ' || V_Num1);
  6  EXCEPTION
  7    WHEN INVALID_NUMBER THEN
  8      DBMS_OUTPUT.PUT_LINE ('Please Check - There is a Source of Invalid
Values in your input (OR) Operations.');
  9* END ;
```

```
SQL> /
Enter value for givenumber1: 123A
Enter value for givenumber2: 2345
DECLARE
*
ERROR at line 1:
ORA-06502: PL/SQL: numeric or value error: character to number conversion error
ORA-06512: at line 4


SQL> ROLBACK;
SP2-0042: unknown command "ROLBACK" - rest of line ignored.
SQL> ROLLBACK;

Rollback complete.

SQL> cl scr

SQL> DECLARE
  2    V_Empno Emp.Empno%TYPE := &Empno;
  3    V_Ename Emp.Ename%TYPE := '&Ename';
  4    V_Deptno Emp.Deptno%TYPE := &Deptno;
  5  BEGIN
  6    INSERT INTO Emp(Empno,Ename,Deptno)
  7    VALUES (V_Empno,V_Ename,V_Deptno);
  8    DBMS_OUTPUT.PUT_LINE('Employee Successfully Inserted');
  9  END ;
 10  /
Enter value for empno: 7654
Enter value for ename: SAMPLE
Enter value for deptno: 30
DECLARE
*
ERROR at line 1:
ORA-00001: unique constraint (SCOTT.EMP_PRIMARY_KEY) violated
ORA-06512: at line 6


SQL> DECLARE
  2    V_Empno Emp.Empno%TYPE := &Empno;
  3    V_Ename Emp.Ename%TYPE := '&Ename';
  4    V_Deptno Emp.Deptno%TYPE := &Deptno;
  5  BEGIN
  6    INSERT INTO Emp(Empno,Ename,Deptno)
  7    VALUES (V_Empno,V_Ename,V_Deptno);
  8    DBMS_OUTPUT.PUT_LINE('Employee Successfully Inserted');
  9  EXCEPTION
 10    WHEN DUP_VAL_ON_INDEX THEN
 11      DBMS_OUTPUT.PUT_LINE('Employee ID Already Exists');
 12  END ;
 13  /
Enter value for empno: 1234
Enter value for ename: SAMPLE
Enter value for deptno: 30
Employee Successfully Inserted
```

```
PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1234
Enter value for ename: SAMPLE
Enter value for deptno: 30
Employee ID Already Exists

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> ROLLBACK;

Rollback complete.

SQL> cl scr

SQL> DECLARE
  2   V_Ename Emp.Ename% TYPE;
  3   V_Sal Emp.Sal%TYPE;
  4   V_RowCount PLS_INTEGER := 0;
  5   CURSOR EmpRowCount IS
  6   SELECT Ename, Sal
  7   FROM Emp
  8   ORDER BY Ename;
  9   BEGIN
 10   OPEN EMPROWCOUNT;
 11   LOOP
 12   FETCH EmpRowCount INTO V_Ename, V_Sal;
 13   EXIT WHEN EmpRowCount%NOTFOUND;
 14   V_RoWCount := EmpRoWCount%ROWCOUNT;
 15   DBMS_OUTPUT.PUT_LINE('Employee Name is,'||V_Ename||' his Salary
 16  is '||V_Sal);
 17   END LOOP;
 18   OPEN EMPROWCOUNT;
 19   DBMS_OUTPUT.PUT_LINE(V_RowCount||' Rows Processed So Far...');
 20   CLOSE EmpRowCount;
 21  EXCEPTION
 22  WHEN CURSOR_ALREADY_OPEN THEN
 23  DBMS_OUTPUT.PUT_LINE('The Requested Cursor is already Open.');
 24  END;
 25  /
Employee Name is,ADAMS his Salary
is 1100
Employee Name is,ALLEN his Salary
is 1600
Employee Name is,BLAKE his Salary
is 2850
Employee Name is,CLARK his Salary
is 2450
Employee Name is,FORD his Salary
is 3000
Employee Name is,JAMES his Salary
```

```
is 950
Employee Name is,JONES his Salary
is 2975
Employee Name is,KING his Salary
is 5000
Employee Name is,MARTIN his Salary
is 1250
Employee Name is,MILLER his Salary
is 1300
Employee Name is,SCOTT his Salary
is 3000
Employee Name is,SMITH his Salary
is 800
Employee Name is,TURNER his Salary
is 1500
Employee Name is,WARD his Salary
is 1250
The Requested Cursor is already Open.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2   V_Ename Emp.Ename% TYPE;
  3   V_Sal Emp.Sal%TYPE;
  4   V_RowCount PLS_INTEGER := 0;
  5   CURSOR EmpRowCount IS
  6   SELECT Ename, Sal
  7   FROM Emp
  8   ORDER BY Ename;
  9   BEGIN
 10   OPEN EMPROWCOUNT;
 11   LOOP
 12   FETCH EmpRowCount INTO V_Ename, V_Sal;
 13   EXIT WHEN EmpRowCount%NOTFOUND;
 14   V_RoWCount := EmpRoWCount%ROWCOUNT;
 15   DBMS_OUTPUT.PUT_LINE('Employee Name is,'||V_Ename||' his Salary
 16  is '||V_Sal);
 17   END LOOP;
 18   DBMS_OUTPUT.PUT_LINE(V_RowCount||' Rows Processed So Far...');
 19   CLOSE EmpRowCount;
 20   CLOSE EmpRowCount;
 21  EXCEPTION
 22  WHEN INVALID_CURSOR THEN
 23  DBMS_OUTPUT.PUT_LINE('The Requested Cursor is either not open or is already
closed.');
 24  END;
 25  /
Employee Name is,ADAMS his Salary
is 1100
Employee Name is,ALLEN his Salary
is 1600
Employee Name is,BLAKE his Salary
is 2850
```

```
Employee Name is,CLARK his Salary
is 2450
Employee Name is,FORD his Salary
is 3000
Employee Name is,JAMES his Salary
is 950
Employee Name is,JONES his Salary
is 2975
Employee Name is,KING his Salary
is 5000
Employee Name is,MARTIN his Salary
is 1250
Employee Name is,MILLER his Salary
is 1300
Employee Name is,SCOTT his Salary
is 3000
Employee Name is,SMITH his Salary
is 800
Employee Name is,TURNER his Salary
is 1500
Employee Name is,WARD his Salary
is 1250
14 Rows Processed So Far...
The Requested Cursor is either not open or is already closed.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2      V_Grade CHAR := UPPER('&EnterGrade');
  3  BEGIN
  4  CASE V_Grade
  5     WHEN 'A' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Excellent
Grade');
  6     WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Very Good
Grade');
  7     WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Good Grade');
  8     WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Fair Grade');
  9     WHEN 'E' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Poor Grade');
 10  END CASE;
 11  EXCEPTION
 12  WHEN CASE_NOT_FOUND THEN
 13  DBMS_OUTPUT.PUT_LINE('The Supplied Case '||V_Grade||' not found. Please
Check once again.');
 14  END;
 15  /
Enter value for entergrade: E
You are Awarded with Poor Grade

PL/SQL procedure successfully completed.

SQL> /
Enter value for entergrade: F
The Supplied Case F not found. Please Check once again.
```

```
PL/SQL procedure successfully completed.

SQL> ED
Wrote file afiedt.buf

  1  DECLARE
  2      V_Grade CHAR := UPPER('&EnterGrade');
  3  BEGIN
  4  CASE V_Grade
  5     WHEN 'A' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Excellent
Grade');
  6     WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Very Good
Grade');
  7     WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Good Grade');
  8     WHEN 'D' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Fair Grade');
  9     WHEN 'E' THEN DBMS_OUTPUT.PUT_LINE('You are Awarded with Poor Grade');
 10  END CASE;
 11* END;
SQL> /
Enter value for entergrade: F
DECLARE
*
ERROR at line 1:
ORA-06592: CASE not found while executing CASE statement
ORA-06512: at line 4


SQL> cl scr

SQL> DECLARE
  2    V_Num1 NUMBER := &GiveNumber1;
  3    V_Num2 NUMBER := &GiveNumber2;
  4    V_Result NUMBER;
  5  BEGIN
  6    V_Result := V_Num1/V_Num2;
  7    DBMS_OUTPUT.PUT_LINE ('The Result is: ' || V_Result);
  8  EXCEPTION
  9    WHEN ZERO_DIVIDE THEN
 10      DBMS_OUTPUT.PUT_LINE ('Fatal Error - Division by zero occurred');
 11  END ;
 12  /
Enter value for givenumber1: 25
Enter value for givenumber2: 5
The Result is: 5

PL/SQL procedure successfully completed.

SQL> /
Enter value for givenumber1: 25
Enter value for givenumber2: 0
Fatal Error - Division by zero occurred

PL/SQL procedure successfully completed.
```

```
SQL> ED
Wrote file afiedt.buf

  1  DECLARE
  2    V_Num1 NUMBER := &GiveNumber1;
  3    V_Num2 NUMBER := &GiveNumber2;
  4    V_Result NUMBER;
  5  BEGIN
  6    V_Result := V_Num1/V_Num2;
  7    DBMS_OUTPUT.PUT_LINE ('The Result is: ' || V_Result);
  8* END ;
SQL> /
Enter value for givenumber1: 25
Enter value for givenumber2: 0
DECLARE
*
ERROR at line 1:
ORA-01476: divisor is equal to zero
ORA-06512: at line 6


SQL> cl scr

SQL> ROLLBACk;

Rollback complete.

SQL> CREATE TABLE Messages
  2  (
  3   Results VARCHAR2(200)
  4  )
  5  /

Table created.

SQL> SELECT * FROM Messages;

no rows selected


Execution Plan
----------------------------------------------------------
   0      SELECT STATEMENT Optimizer=ALL_ROWS (Cost=2 Card=1 Bytes=102
          )

   1    0   TABLE ACCESS (FULL) OF 'MESSAGES' (TABLE) (Cost=2 Card=1 B
          ytes=102)




SQL> DECLARE
  2    V_Ename Emp.Ename%TYPE;
  3    V_Sal Emp.Sal%TYPE := &Sal;
  4    BEGIN
```

```
  5    SELECT Ename INTO V_Ename
  6    FROM Emp
  7    WHERE Sal = V_Sal;
  8    INSERT INTO Messages(Results)
  9    VALUES(USER||' Queried for Salary of '||V_Sal||' and '||V_Ename||' was
Found.');
 10  EXCEPTION
 11    WHEN NO_DATA_FOUND THEN
 12     INSERT INTO Messages(Results)
 13     VALUES(USER||' Queried For '||V_Sal||' Salary on '||TO_CHAR(SYSDATE,'DD,
Month YYYY, HH24:MI:SS')||' And The Requested Salary Was Not Found.');
 14  WHEN TOO_MANY_ROWS THEN
 15      INSERT INTO Messages(Results)
 16      VALUES(USER||' Queried For '||V_Sal||' Salary on
'||TO_CHAR(SYSDATE,'FMDD, Month YYYY, HH24:MI:SS')||' And Many Records Were
Matching Upon That Salary Hence Query Failed To Retrieve The Data.');
 17  WHEN OTHERS THEN
 18  INSERT INTO Messages(Results)
 19      VALUES(USER||' Executed For Some Data '||TO_CHAR(SYSDATE,'FMDD, Month
YYYY, HH24:MI:SS')||' And Administrators Attention is Expected.');
 20  END ;
 21  /
Enter value for sal: 800

PL/SQL procedure successfully completed.

SQL> /
Enter value for sal: 5000

PL/SQL procedure successfully completed.

SQL> /
Enter value for sal: 3000

PL/SQL procedure successfully completed.

SQL> /
Enter value for sal: 1234

PL/SQL procedure successfully completed.

SQL> /
Enter value for sal: 1250

PL/SQL procedure successfully completed.

SQL> SELECT * FROM Messages;

RESULTS
--------------------------------------------------------------------------------
SCOTT Queried for Salary of 800 and SMITH was Found.
SCOTT Queried for Salary of 5000 and KING was Found.
SCOTT Queried For 3000 Salary on 12, July 2010, 10:57:27 And Many Records Were M
atching Upon That Salary Hence Query Failed To Retrieve The Data.
```

```
SCOTT Queried For 1234 Salary on 12, July      2010, 10:57:32 And The Requested
Salary Was Not Found.

SCOTT Queried For 1250 Salary on 12, July 2010, 10:57:43 And Many Records Were M
atching Upon That Salary Hence Query Failed To Retrieve The Data.



Execution Plan
----------------------------------------------------------
   0      SELECT STATEMENT Optimizer=ALL_ROWS (Cost=3 Card=5 Bytes=510
          )

   1    0   TABLE ACCESS (FULL) OF 'MESSAGES' (TABLE) (Cost=3 Card=5 B
            ytes=510)




SQL> SPOOL OFF
SQL> cl scr

SQL> SET SERVEROUTPUT ON
SQL> SET VERIFY OFF
SQL> cl scr

SQL> DECLARE
  2  V_Ename Emp.Ename%TYPE;
  3  V_Sal Emp.Sal%TYPE;
  4  V_Job Emp.Job%TYPE;
  5  V_Deptno Emp.Deptno%TYPE;
  6  BEGIN
  7  SELECT Ename, Sal, Job, Deptno
  8  INTO
  9  V_Ename, V_Sal, V_Job, V_Deptno
 10  FROM Emp
 11  WHERE Empno = &GiveEmpNo;
 12  DBMS_OUTPUT.PUT_LINE('The Employee Name is :'||V_Ename||' Working for
Department '||V_Deptno||' Having Salary of '||V_Sal||'.');
 13   DECLARE
 14   V_Staff NUMBER(1);
 15   BEGIN
 16   SELECT COUNT(*)
 17   INTO V_Staff
 18   FROM Emp
 19   WHERE Deptno = V_Deptno;
 20   DBMS_OUTPUT.PUT_LINE('The Total Number of Employees Working in Department
'||V_Deptno||' are '||V_Staff||'.');
 21   EXCEPTION
 22   WHEN VALUE_ERROR OR INVALID_NUMBER THEN
 23   DBMS_OUTPUT.PUT_LINE('There is some Error in the Data inputs (OR) Data
outputs, Please Check, Debug the Source.');
 24   END;
 25  EXCEPTION
 26  WHEN NO_DATA_FOUND THEN
```

```
 27  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 28  END;
 29  /
Enter value for giveempno: 1234
The Given Information is Missing in the Database, Check for Proper Values.

PL/SQL procedure successfully completed.

SQL> /
Enter value for giveempno: 7839
The Employee Name is :KING Working for Department 10 Having Salary of 5000.
The Total Number of Employees Working in Department 10 are 3.

PL/SQL procedure successfully completed.

SQL> /
Enter value for giveempno: 7566
The Employee Name is :JONES Working for Department 20 Having Salary of 2975.
The Total Number of Employees Working in Department 20 are 5.

PL/SQL procedure successfully completed.

SQL> /
Enter value for giveempno: 7654
The Employee Name is :MARTIN Working for Department 30 Having Salary of 1250.
The Total Number of Employees Working in Department 30 are 6.

PL/SQL procedure successfully completed.

SQL> INSERT INTO Emp(Empno, Ename, Deptno, Job)
  2  VALUES(1234, 'SAMPLE01', 30, 'CLERK');

1 row created.

SQL> DECLARE
  2  V_Ename Emp.Ename%TYPE;
  3  V_Sal Emp.Sal%TYPE;
  4  V_Job Emp.Job%TYPE;
  5  V_Deptno Emp.Deptno%TYPE;
  6  BEGIN
  7  SELECT Ename, Sal, Job, Deptno
  8  INTO
  9  V_Ename, V_Sal, V_Job, V_Deptno
 10  FROM Emp
 11  WHERE Empno = &GiveEmpNo;
 12  DBMS_OUTPUT.PUT_LINE('The Employee Name is :'||V_Ename||' Working for
Department '||V_Deptno||' Having Salary of '||V_Sal||'.');
 13   DECLARE
 14   V_Staff NUMBER(1);
 15   BEGIN
 16   SELECT COUNT(*)
 17   INTO V_Staff
 18   FROM Emp
 19   WHERE Deptno = V_Deptno;
```

```
 20   DBMS_OUTPUT.PUT_LINE('The Total Number of Employees Working in Department
'||V_Deptno||' are '||V_Staff||'.');
 21   EXCEPTION
 22   WHEN VALUE_ERROR OR INVALID_NUMBER THEN
 23   DBMS_OUTPUT.PUT_LINE('There is some Error in the Data inputs (OR) Data
outputs, Please Check, Debug the Source.');
 24   END;
 25  EXCEPTION
 26  WHEN NO_DATA_FOUND THEN
 27  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 28  END;
 29  /
Enter value for giveempno: 7654
The Employee Name is :MARTIN Working for Department 30 Having Salary of 1250.
The Total Number of Employees Working in Department 30 are 7.

PL/SQL procedure successfully completed.

SQL> INSERT INTO Emp(Empno, Ename, Deptno, Job)
  2  VALUES(1235, 'SAMPLE02', 30, 'CLERK');

1 row created.

SQL> DECLARE
  2  V_Ename Emp.Ename%TYPE;
  3  V_Sal Emp.Sal%TYPE;
  4  V_Job Emp.Job%TYPE;
  5  V_Deptno Emp.Deptno%TYPE;
  6  BEGIN
  7  SELECT Ename, Sal, Job, Deptno
  8  INTO
  9  V_Ename, V_Sal, V_Job, V_Deptno
 10  FROM Emp
 11  WHERE Empno = &GiveEmpNo;
 12  DBMS_OUTPUT.PUT_LINE('The Employee Name is :'||V_Ename||' Working for
Department '||V_Deptno||' Having Salary of '||V_Sal||'.');
 13   DECLARE
 14   V_Staff NUMBER(1);
 15   BEGIN
 16   SELECT COUNT(*)
 17   INTO V_Staff
 18   FROM Emp
 19   WHERE Deptno = V_Deptno;
 20   DBMS_OUTPUT.PUT_LINE('The Total Number of Employees Working in Department
'||V_Deptno||' are '||V_Staff||'.');
 21   EXCEPTION
 22   WHEN VALUE_ERROR OR INVALID_NUMBER THEN
 23   DBMS_OUTPUT.PUT_LINE('There is some Error in the Data inputs (OR) Data
outputs, Please Check, Debug the Source.');
 24   END;
 25  EXCEPTION
 26  WHEN NO_DATA_FOUND THEN
 27  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
```

```
 28  END;
 29  /
Enter value for giveempno: 7654
The Employee Name is :MARTIN Working for Department 30 Having Salary of 1250.
The Total Number of Employees Working in Department 30 are 8.

PL/SQL procedure successfully completed.

SQL> INSERT INTO Emp(Empno, Ename, Deptno, Job)
  2  VALUES(1236, 'SAMPLE03', 30, 'CLERK');

1 row created.

SQL> DECLARE
  2  V_Ename Emp.Ename%TYPE;
  3  V_Sal Emp.Sal%TYPE;
  4  V_Job Emp.Job%TYPE;
  5  V_Deptno Emp.Deptno%TYPE;
  6  BEGIN
  7  SELECT Ename, Sal, Job, Deptno
  8  INTO
  9  V_Ename, V_Sal, V_Job, V_Deptno
 10  FROM Emp
 11  WHERE Empno = &GiveEmpNo;
 12  DBMS_OUTPUT.PUT_LINE('The Employee Name is :'||V_Ename||' Working for
Department '||V_Deptno||' Having Salary of '||V_Sal||'.');
 13   DECLARE
 14   V_Staff NUMBER(1);
 15   BEGIN
 16   SELECT COUNT(*)
 17   INTO V_Staff
 18   FROM Emp
 19   WHERE Deptno = V_Deptno;
 20   DBMS_OUTPUT.PUT_LINE('The Total Number of Employees Working in Department
'||V_Deptno||' are '||V_Staff||'.');
 21   EXCEPTION
 22   WHEN VALUE_ERROR OR INVALID_NUMBER THEN
 23   DBMS_OUTPUT.PUT_LINE('There is some Error in the Data inputs (OR) Data
outputs, Please Check, Debug the Source.');
 24   END;
 25  EXCEPTION
 26  WHEN NO_DATA_FOUND THEN
 27  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 28  END;
 29  /
Enter value for giveempno: 7654
The Employee Name is :MARTIN Working for Department 30 Having Salary of 1250.
The Total Number of Employees Working in Department 30 are 9.

PL/SQL procedure successfully completed.

SQL> INSERT INTO Emp(Empno, Ename, Deptno, Job)
  2  VALUES(1237, 'SAMPLE04', 30, 'CLERK');
```

```
1 row created.

SQL> DECLARE
  2  V_Ename Emp.Ename%TYPE;
  3  V_Sal Emp.Sal%TYPE;
  4  V_Job Emp.Job%TYPE;
  5  V_Deptno Emp.Deptno%TYPE;
  6  BEGIN
  7  SELECT Ename, Sal, Job, Deptno
  8  INTO
  9  V_Ename, V_Sal, V_Job, V_Deptno
 10  FROM Emp
 11  WHERE Empno = &GiveEmpNo;
 12  DBMS_OUTPUT.PUT_LINE('The Employee Name is :'||V_Ename||' Working for
Department '||V_Deptno||' Having Salary of '||V_Sal||'.');
 13   DECLARE
 14   V_Staff NUMBER(1);
 15   BEGIN
 16   SELECT COUNT(*)
 17   INTO V_Staff
 18   FROM Emp
 19   WHERE Deptno = V_Deptno;
 20   DBMS_OUTPUT.PUT_LINE('The Total Number of Employees Working in Department
'||V_Deptno||' are '||V_Staff||'.');
 21   EXCEPTION
 22   WHEN VALUE_ERROR OR INVALID_NUMBER THEN
 23   DBMS_OUTPUT.PUT_LINE('There is some Error in the Data inputs (OR) Data
outputs, Please Check, Debug the Source.');
 24   END;
 25  EXCEPTION
 26  WHEN NO_DATA_FOUND THEN
 27  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 28  END;
 29  /
Enter value for giveempno: 7654
The Employee Name is :MARTIN Working for Department 30 Having Salary of 1250.
There is some Error in the Data inputs (OR) Data outputs, Please Check, Debug
the Source.

PL/SQL procedure successfully completed.

SQL> ROLLBACK;

Rollback complete.

SQL> DECLARE
  2  V_Ename Emp.Ename%TYPE;
  3  V_Sal Emp.Sal%TYPE;
  4  V_Job Emp.Job%TYPE;
  5  V_Deptno Emp.Deptno%TYPE;
  6  BEGIN
  7  SELECT Ename, Sal, Job, Deptno
  8  INTO
  9  V_Ename, V_Sal, V_Job, V_Deptno
```

```
 10  FROM Emp
 11  WHERE Empno = &GiveEmpNo;
 12  DBMS_OUTPUT.PUT_LINE('The Employee Name is :'||V_Ename||' Working for
Department '||V_Deptno||' Having Salary of '||V_Sal||'.');
 13   DECLARE
 14   V_Staff NUMBER(1);
 15   BEGIN
 16   SELECT COUNT(*)
 17   INTO V_Staff
 18   FROM Emp
 19   WHERE Deptno = V_Deptno;
 20   DBMS_OUTPUT.PUT_LINE('The Total Number of Employees Working in Department
'||V_Deptno||' are '||V_Staff||'.');
 21   EXCEPTION
 22   WHEN VALUE_ERROR OR INVALID_NUMBER THEN
 23   DBMS_OUTPUT.PUT_LINE('There is some Error in the Data inputs (OR) Data
outputs, Please Check, Debug the Source.');
 24   END;
 25  EXCEPTION
 26  WHEN NO_DATA_FOUND THEN
 27  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 28  END;
 29  /
Enter value for giveempno: 7654
The Employee Name is :MARTIN Working for Department 30 Having Salary of 1250.
The Total Number of Employees Working in Department 30 are 6.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2  E_EmpRemaining EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(E_EmpRemaining, -2292);
  4  V_Deptno Emp.Deptno%TYPE := &GiveDeptno;
  5  BEGIN
  6  DELETE FROM Dept
  7  WHERE Deptno = V_Deptno;
  8  IF SQL%NOTFOUND THEN
  9  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 10  END IF;
 11  ROLLBACK;
 12  END;
 13  /
Enter value for givedeptno: 40

PL/SQL procedure successfully completed.

SQL> /
Enter value for givedeptno: 30
DECLARE
*
ERROR at line 1:
```

```
ORA-02292: integrity constraint (SCOTT.EMP_FOREIGN_KEY) violated - child record
found
ORA-06512: at line 6


SQL> DECLARE
  2  E_EmpRemaining EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(E_EmpRemaining, -2292);
  4  V_Deptno Emp.Deptno%TYPE := &GiveDeptno;
  5  BEGIN
  6  DELETE FROM Dept
  7  WHERE Deptno = V_Deptno;
  8  IF SQL%NOTFOUND THEN
  9  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 10  END IF;
 11  ROLLBACK;
 12  EXCEPTION
 13  WHEN E_EmpRemaining THEN
 14  DBMS_OUTPUT.PUT_LINE('Unable to Delete the Department Number '||V_Deptno||'
as the Employees are Existing. Validate Your Relations and then Try Once
Again.');
 15  WHEN NO_DATA_FOUND THEN
 16  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 17  END;
 18  /
Enter value for givedeptno: 30
Unable to Delete the Department Number 30 as the Employees are Existing.
Validate Your Relations and then Try Once Again.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2  E_EmpExists EXCEPTION;
  3  V_Count NUMBER(2);
  4  V_SalSum NUMBER(6);
  5  PRAGMA EXCEPTION_INIT(E_EmpExists, -2292);
  6  V_Empno Emp.Empno%TYPE := &GiveEmpno;
  7  BEGIN
  8  DELETE FROM Emp
  9  WHERE Empno = V_Empno;
 10  IF SQL%NOTFOUND THEN
 11  DBMS_OUTPUT.PUT_LINE('The Given Employee Number '||V_Empno||' is Missing in
the Database, Check for Proper Values.');
 12  ROLLBACK;
 13  ELSE
 14  COMMIT;
 15  END IF;
 16  EXCEPTION
 17  WHEN E_EmpExists THEN
```

```
 18  DBMS_OUTPUT.PUT_LINE('Unable to Delete the Employee Details '||V_Empno||'
as the Employees are Existing. Validate Your Relations and then Try Once
Again.');
 19  WHEN NO_DATA_FOUND THEN
 20  DBMS_OUTPUT.PUT_LINE('The Given Information is Missing in the Database,
Check for Proper Values.');
 21  END;
 22  /
Enter value for giveempno: 7654

PL/SQL procedure successfully completed.

SQL> /
Enter value for giveempno: 7566
Unable to Delete the Employee Details 7566 as the Employees are Existing.
Validate Your Relations and then Try Once Again.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2  E_NotNULLViolation EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(E_NotNULLViolation, -1400);
  4  BEGIN
  5  INSERT INTO Emp(Empno, Ename, Job, Sal, Comm, Deptno)
  6  VALUES(&Empno, 'SATISH', 'ANALYST', 25000, NULL, &Deptno);
  7  COMMIT;
  8  EXCEPTION
  9  WHEN E_NotNULLViolation THEN
 10  DBMS_OUTPUT.PUT_LINE('A Field which Cannot be NULL, is not attended, Please
Check Properly...');
 11  END;
 12  /
Enter value for empno: 1234
Enter value for deptno: 30

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: NULL
Enter value for deptno: 30
A Field which Cannot be NULL, is not attended, Please Check Properly...

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1235
Enter value for deptno: NULL
A Field which Cannot be NULL, is not attended, Please Check Properly...

PL/SQL procedure successfully completed.

SQL> cl scr
```

```
SQL> ALTER TABLE Emp
  2  ADD CONSTRAINT EmpEnameCHK CHECK(Ename = UPPER(Ename))
  3  ADD CONSTRAINT EmpJobCHK CHECK(Job = UPPER(Job))
  4  /

Table altered.

SQL> cl scr

SQL> DECLARE
  2  E_NotNULLViolation EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(E_NotNULLViolation, -1400);
  4  E_CheckViolation EXCEPTION;
  5  PRAGMA EXCEPTION_INIT(E_CheckViolation, -2290);
  6  BEGIN
  7  INSERT INTO Emp(Empno, Ename, Job, Sal, Comm, Deptno)
  8  VALUES(&Empno, '&Ename', '&Job', 25000, NULL, &Deptno);
  9  COMMIT;
 10  EXCEPTION
 11  WHEN E_CheckViolation THEN
 12  DBMS_OUTPUT.PUT_LINE('A Field with Check Constraint is not Attended
Properly, Please Check Properly.');
 13  WHEN E_NotNULLViolation THEN
 14  DBMS_OUTPUT.PUT_LINE('A Field which Cannot be NULL, is not attended, Please
Check Properly.');
 15  END;
 16  /
Enter value for empno: 1235
Enter value for ename: SAMPLE
Enter value for job: CLERK
Enter value for deptno: 30

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1236
Enter value for ename: Sample
Enter value for job: CLERK
Enter value for deptno: 30
A Field with Check Constraint is not Attended Properly, Please Check Properly.

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: NULL
Enter value for ename: SAMPLE
Enter value for job: CLERK
Enter value for deptno: 30
A Field which Cannot be NULL, is not attended, Please Check Properly.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
```

```
  2  E_NoData EXCEPTION;
  3  V_Empno Emp.Empno%TYPE := &GEmpno;
  4  V_Ename Emp.Ename%TYPE;
  5  BEGIN
  6  SELECT Ename INTO V_Ename
  7  FROM Emp
  8  WHERE Empno = V_Empno;
  9  DBMS_OUTPUT.PUT_LINE('The Employee : '||V_Ename);
 10  EXCEPTION
 11  WHEN NO_DATA_FOUND THEN
 12  DBMS_OUTPUT.PUT_LINE('Sorry! Data Not Found');
 13  END;
 14  /
Enter value for gempno: 7839
The Employee : KING

PL/SQL procedure successfully completed.

SQL> /
Enter value for gempno: 2345
Sorry! Data Not Found

PL/SQL procedure successfully completed.

SQL> ED
Wrote file afiedt.buf

  1  DECLARE
  2  E_NoData EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(E_NoData, -1403);
  4  V_Empno Emp.Empno%TYPE := &GEmpno;
  5  V_Ename Emp.Ename%TYPE;
  6  BEGIN
  7  SELECT Ename INTO V_Ename
  8  FROM Emp
  9  WHERE Empno = V_Empno;
 10  DBMS_OUTPUT.PUT_LINE('The Employee : '||V_Ename);
 11  EXCEPTION
 12  WHEN E_NoData  THEN
 13  DBMS_OUTPUT.PUT_LINE('Sorry! Data Not Found');
 14* END;
SQL> /
Enter value for gempno: 2345
E_NoData EXCEPTION;
*
ERROR at line 2:
ORA-06550: line 2, column 1:
PLS-00701: illegal ORACLE error number -1403 for PRAGMA EXCEPTION_INIT


SQL> EDED
SP2-0042: unknown command "EDED" - rest of line ignored.
SQL> ED
Wrote file afiedt.buf
```

```
  1  DECLARE
  2  E_NoData EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(E_NoData, +100);
  4  V_Empno Emp.Empno%TYPE := &GEmpno;
  5  V_Ename Emp.Ename%TYPE;
  6  BEGIN
  7  SELECT Ename INTO V_Ename
  8  FROM Emp
  9  WHERE Empno = V_Empno;
 10  DBMS_OUTPUT.PUT_LINE('The Employee : '||V_Ename);
 11  EXCEPTION
 12  WHEN E_NoData  THEN
 13  DBMS_OUTPUT.PUT_LINE('Sorry! Data Not Found');
 14* END;
SQL> /
Enter value for gempno: 2345
Sorry! Data Not Found

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2  V_ErrorCode NUMBER(6);
  3  V_ErrorMessage VARCHAR2(200);
  4  BEGIN
  5  INSERT INTO Dept
  6  VALUES(&DeptNumber, '&DeptName', '&DeptLocation');
  7  EXCEPTION
  8  WHEN OTHERS THEN
  9  V_ErrorCode := SQLCODE;
 10  V_ErrorMessage := SUBSTR(SQLERRM, 1, 200);
 11  DBMS_OUTPUT.PUT_LINE('The Error Code Traced is : '||V_ErrorCode);
 12  DBMS_OUTPUT.PUT_LINE('The Error Message Traced is : '||V_ErrorMessage);
 13  END;
 14  /
Enter value for deptnumber: 50
Enter value for deptname: SHIPPING
Enter value for deptlocation: CHENNAI

PL/SQL procedure successfully completed.

SQL> /
Enter value for deptnumber: 50
Enter value for deptname: SHIPPING
Enter value for deptlocation: CHENNAI
The Error Code Traced is : -1
The Error Message Traced is : ORA-00001: unique constraint
(SCOTT.DEPT_PRIMARY_KEY) violated

PL/SQL procedure successfully completed.

SQL> ED
Wrote file afiedt.buf
```

```
  1  DECLARE
  2  V_ErrorCode NUMBER(6);
  3  V_ErrorMessage VARCHAR2(200);
  4  BEGIN
  5  INSERT INTO Dept
  6  VALUES(&DeptNumber, '&DeptName', '&DeptLocation');
  7  EXCEPTION
  8  WHEN DUP_VAL_ON_INDEX THEN
  9  DBMS_OUTPUT.PUT_LINE('Sorry Department Number Cannot Be Duplicated');
 10  WHEN OTHERS THEN
 11  V_ErrorCode := SQLCODE;
 12  V_ErrorMessage := SUBSTR(SQLERRM, 1, 200);
 13  DBMS_OUTPUT.PUT_LINE('The Error Code Traced is : '||V_ErrorCode);
 14  DBMS_OUTPUT.PUT_LINE('The Error Message Traced is : '||V_ErrorMessage);
 15* END;
SQL> /
Enter value for deptnumber: 50
Enter value for deptname: SHIPPING
Enter value for deptlocation: CHENNAI
Sorry Department Number Cannot Be Duplicated

PL/SQL procedure successfully completed.

SQL> /
Enter value for deptnumber: NULL
Enter value for deptname: CARGO
Enter value for deptlocation: DELHI
The Error Code Traced is : -1400
The Error Message Traced is : ORA-01400: cannot insert NULL into
("SCOTT"."DEPT"."DEPTNO")

PL/SQL procedure successfully completed.

SQL> ED
Wrote file afiedt.buf

  1  DECLARE
  2  E_NullDept EXCEPTION;
  3  PRAGMA EXCEPTION_INIT(E_NullDept, -1400);
  4  V_ErrorCode NUMBER(6);
  5  V_ErrorMessage VARCHAR2(200);
  6  BEGIN
  7  INSERT INTO Dept
  8  VALUES(&DeptNumber, '&DeptName', '&DeptLocation');
  9  EXCEPTION
 10  WHEN E_NullDept  THEN
 11  DBMS_OUTPUT.PUT_LINE('Department Number Cannot Be Null');
 12  WHEN DUP_VAL_ON_INDEX THEN
 13  DBMS_OUTPUT.PUT_LINE('Sorry Department Number Cannot Be Duplicated');
 14  WHEN OTHERS THEN
 15  V_ErrorCode := SQLCODE;
 16  V_ErrorMessage := SUBSTR(SQLERRM, 1, 200);
 17  DBMS_OUTPUT.PUT_LINE('The Error Code Traced is : '||V_ErrorCode);
 18  DBMS_OUTPUT.PUT_LINE('The Error Message Traced is : '||V_ErrorMessage);
 19* END;
```

```
SQL> /
Enter value for deptnumber: NULL
Enter value for deptname: CARGO
Enter value for deptlocation: DELHI
Department Number Cannot Be Null

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> CREATE TABLE MyAudit
  2  (
  3   UserName VARCHAR2(15),
  4   ModDate DATE,
  5   Message VARCHAR2(100)
  6  )
  7  /

Table created.

SQL> CREATE TABLE TrappedMessages
  2  (
  3   ErrorNum NUMBER(6),
  4   ErrorMsg VARCHAR2(250),
  5   TrapDate DATE
  6  )
  7  /

Table created.

SQL> BEGIN
  2  INSERT INTO Dept
  3  VALUES(&DeptNumber, '&DeptName', '&DeptLocation');
  4  COMMIT;
  5  INSERT INTO MyAudit(UserName,  ModDate)
  6  VALUES(USER, SYSDATE);
  7  COMMIT;
  8  EXCEPTION
  9  WHEN OTHERS THEN
 10   DECLARE
 11   V_ErrorCode NUMBER(6);
 12   V_ErrorMessage VARCHAR2(200);
 13   BEGIN
 14   V_ErrorCode := SQLCODE;
 15   V_ErrorMessage := SUBSTR(SQLERRM, 1, 200);
 16   DBMS_OUTPUT.PUT_LINE('The Error Code Traced is : '||V_ErrorCode);
 17   DBMS_OUTPUT.PUT_LINE('The Error Message Traced is : '||V_ErrorMessage);
 18   INSERT INTO TrappedMessages(ErrorNum, ErrorMsg, TrapDate)
 19   VALUES(V_ErrorCode, V_ErrorMessage, SYSDATE);
 20   COMMIT;
 21   END;
 22  END;
 23  /
Enter value for deptnumber: 60
Enter value for deptname: CARGO
```

```
Enter value for deptlocation: DELHI

PL/SQL procedure successfully completed.

SQL> /
Enter value for deptnumber: 50
Enter value for deptname: COURIER
Enter value for deptlocation: MUMBAI
The Error Code Traced is : -1
The Error Message Traced is : ORA-00001: unique constraint
(SCOTT.DEPT_PRIMARY_KEY) violated

PL/SQL procedure successfully completed.

SQL> SELECT * FROM MyAudit;

USERNAME        MODDATE
--------------- ---------
MESSAGE
--------------------------------------------------------------------------------
SCOTT           13-JUL-10



SQL> SELECT * FROM TrappedMessages;

  ERRORNUM
----------
ERRORMSG
--------------------------------------------------------------------------------
TRAPDATE
---------
        -1
ORA-00001: unique constraint (SCOTT.DEPT_PRIMARY_KEY) violated
13-JUL-10


SQL> cl scr

SQL> DECLARE
  2  V_DeptNo Dept.Deptno%TYPE := &DeptNumber;
  3  V_DeptName Dept.Dname%TYPE := '&DeptName';
  4  V_DeptLoc Dept.Loc%TYPE := '&DeptLocation';
  5  E_InvalidDept EXCEPTION;
  6  BEGIN
  7  UPDATE Dept
  8  SET Dname = V_DeptName,
  9         Loc = V_DeptLoc
 10  WHERE Deptno = V_DeptNo;
 11  END;
 12  /
Enter value for deptnumber: 30
Enter value for deptname: SALESDIV
Enter value for deptlocation: DELHI
```

```
PL/SQL procedure successfully completed.

SQL> /
Enter value for deptnumber: 70
Enter value for deptname: SAMPLE
Enter value for deptlocation: SAMPLE

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2  V_DeptNo Dept.Deptno%TYPE := &DeptNumber;
  3  V_DeptName Dept.Dname%TYPE := '&DeptName';
  4  V_DeptLoc Dept.Loc%TYPE := '&DeptLocation';
  5  E_InvalidDept EXCEPTION;
  6  BEGIN
  7  UPDATE Dept
  8  SET Dname = V_DeptName,
  9         Loc = V_DeptLoc
 10  WHERE Deptno = V_DeptNo;
 11  IF SQL%NOTFOUND THEN
 12  DBMS_OUTPUT.PUT_LINE('The Specific Department Number '||V_DeptNo||' you
wanted to Update is not Found. Please Confirm the Data.');
 13  ENd IF;
 14  END;
 15  /
Enter value for deptnumber: 70
Enter value for deptname: SAMPLE
Enter value for deptlocation: SAMPLE
The Specific Department Number 70 you wanted to Update is not Found. Please
Confirm the Data.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2  V_DeptNo Dept.Deptno%TYPE := &DeptNumber;
  3  V_DeptName Dept.Dname%TYPE := '&DeptName';
  4  V_DeptLoc Dept.Loc%TYPE := '&DeptLocation';
  5  E_InvalidDept EXCEPTION;
  6  BEGIN
  7  UPDATE Dept
  8  SET Dname = V_DeptName,
  9         Loc = V_DeptLoc
 10  WHERE Deptno = V_DeptNo;
 11  IF SQL%NOTFOUND THEN
 12  RAISE E_InvalidDept;
 13  ENd IF;
 14  COMMIT;
 15  EXCEPTION
 16  WHEN E_InvalidDept THEN
 17  DBMS_OUTPUT.PUT_LINE('The Specific Department Number '||V_DeptNo||' you
wanted to Update is not Found. Please Confirm the Data.');
```

```
 18  INSERT INTO MyAudit(UserName, ModDate, Message)
 19  VALUES(USER, SYSDATE, 'Tried Illegal Update.');
 20  END;
 21  /
Enter value for deptnumber: 70
Enter value for deptname: SAMPLE
Enter value for deptlocation: SAMPLE
The Specific Department Number 70 you wanted to Update is not Found. Please
Confirm the Data.

PL/SQL procedure successfully completed.

SQL> SELECT * FROm MyAudit;

USERNAME        MODDATE
--------------- ---------
MESSAGE
--------------------------------------------------------------------------------
SCOTT           13-JUL-10


SCOTT           13-JUL-10
Tried Illegal Update.


SQL> SPOOL OFF
SQL> cl scr

SQL> SET SERVEROUTPUT ON
SQL> SET VERIFY OFF
SQL> cl scr

SQL> DECLARE
  2  TYPE ValidateInsertValues
  3  IS
  4  RECORD
  5  (
  6   ValidateJob             NUMBER(2),
  7   ValidateDeptno          NUMBER(2),
  8   ValidateManager         NUMBER(2),
  9   SalaryLess              NUMBER(4),
 10   SalaryMore              NUMBER(6),
 11   CurrentDate             DATE
 12   );
 13  ValidateRecordValues     ValidateInsertValues;
 14  V_Empno                        Emp.Empno%TYPE := &Empno;
 15  V_Ename                        Emp.Ename%TYPE := '&Ename';
 16  V_Job                    Emp.Job%TYPE := '&Job';
 17  V_Sal                    Emp.Sal%TYPE := &Salary;
 18  V_Comm                   Emp.Comm%TYPE := &Commission;
 19  V_Deptno                 Emp.Deptno%TYPE := &Deptno;
 20  V_MGR                    Emp.MGR%TYPE := &Manager;
 21  V_HireDate               Emp.HireDate%TYPE := '&HireDate';
 22  E_SalLess                EXCEPTION;
 23  E_SalMore                EXCEPTION;
```

```
 24  E_NoDept                EXCEPTION;
 25  E_NoJob                       EXCEPTION;
 26  E_InvalidHireDate       EXCEPTION;
 27  E_InvalidManager        EXCEPTION;
 28  E_NotNULLViolation EXCEPTION;
 29  PRAGMA EXCEPTION_INIT(E_NotNULLViolation, -1400);
 30  E_CheckViolation        EXCEPTION;
 31  PRAGMA EXCEPTION_INIT(E_CheckViolation, -2290);
 32  BEGIN
 33  ValidateRecordValues.SalaryLess := 500;
 34  ValidateRecordValues.SalaryMore := 150000;
 35  SELECT
 36  SYSDATE
 37  INTO
 38  ValidateRecordValues.CurrentDate
 39  FROM DUAL;
 40  ValidateRecordValues.CurrentDate :=
TO_DATE(TO_CHAR(ValidateRecordValues.CurrentDate, 'DD-MON-YY'), 'DD-MON-YY');
 41  SELECT
 42  COUNT(*)
 43  INTO
 44  ValidateRecordValues.ValidateJob
 45  FROM Emp
 46  WHERE Job = V_Job;
 47  SELECT
 48  COUNT(*)
 49  INTO
 50  ValidateRecordValues.ValidateDeptno
 51  FROM Dept
 52  WHERE Deptno = V_Deptno;
 53  SELECT
 54  COUNT(*)
 55  INTO
 56  ValidateRecordValues.ValidateManager
 57  FROM Emp
 58  WHERE MGR = V_MGR;
 59  IF ValidateRecordValues.ValidateJob = 0
 60  THEN
 61   RAISE E_NoJob;
 62  ELSIF ValidateRecordValues.ValidateDeptno = 0
 63  THEN
 64   RAISE E_NoDept;
 65  ELSIF ValidateRecordValues.ValidateManager = 0
 66  THEN
 67   RAISE E_InvalidManager;
 68  END IF;
 69  IF V_Sal < ValidateRecordValues.SalaryLess
 70  THEN
 71   RAISE E_SalLess;
 72  ELSIF V_Sal > ValidateRecordValues.SalaryMore
 73  THEN
 74   RAISE E_SalMore;
 75  END IF;
 76  IF ValidateRecordValues.CurrentDate <> V_HireDate
 77  THEN
```

```
 78   RAISE E_InvalidHireDate;
 79  END IF;
 80  INSERT INTO Emp(Empno, Ename, Job, Sal, Comm, Deptno, MGR, HireDate)
 81  VALUES(V_Empno, V_Ename, V_Job, V_Sal, V_Comm, V_Deptno, V_MGR,
V_HireDate);
 82  COMMIT;
 83  EXCEPTION
 84  WHEN DUP_VAL_ON_INDEX THEN
 85  DBMS_OUTPUT.PUT_LINE('Sorry, The Field With Unique Value is Getting
Duplicated.');
 86  ROLLBACK;
 87  WHEN E_CheckViolation THEN
 88  DBMS_OUTPUT.PUT_LINE('A Field with Check Constraint is not Attended
Properly, Please Check Properly.');
 89  ROLLBACK;
 90  WHEN E_NotNULLViolation THEN
 91  DBMS_OUTPUT.PUT_LINE('A Field which Cannot be NULL, is not attended, Please
Check Properly.');
 92  ROLLBACK;
 93  WHEN E_NoJob THEN
 94  DBMS_OUTPUT.PUT_LINE('Sorry, The Job To be Inserted is Not Acceptable.');
 95  ROLLBACK;
 96  WHEN E_NoDept THEN
 97  DBMS_OUTPUT.PUT_LINE('Sorry, The Department Number To be Inserted is Not
Acceptable.');
 98  ROLLBACK;
 99  WHEN E_InvalidManager THEN
100  DBMS_OUTPUT.PUT_LINE('Sorry, The Manager Number To be Inserted is Not
Acceptable.');
101  ROLLBACK;
102  WHEN E_SalLess THEN
103  DBMS_OUTPUT.PUT_LINE('Sorry, The Salary To be Inserted is Less Than The
Acceptable Value.');
104  ROLLBACK;
105  WHEN E_SalMore THEN
106  DBMS_OUTPUT.PUT_LINE('Sorry, The Salary To be Inserted is More Than The
Acceptable Value.');
107  ROLLBACK;
108  WHEN E_InvalidHireDate THEN
109  DBMS_OUTPUT.PUT_LINE('Sorry, The Hire Date To be Inserted is Not
Acceptable.');
110  ROLLBACK;
111  END;
112  /
Enter value for empno: 1234
Enter value for ename: SAMPLE
Enter value for job: CLERK
Enter value for salary: 2500
Enter value for commission: NULL
Enter value for deptno: 30
Enter value for manager: 7566
Enter value for hiredate: 14-JUL-10

PL/SQL procedure successfully completed.
```

```
SQL> /
Enter value for empno: 1234
Enter value for ename: SAMPLE
Enter value for job: CLERK
Enter value for salary: 2000
Enter value for commission: NULL
Enter value for deptno: 30
Enter value for manager: 7566
Enter value for hiredate: 14-JUL-10
Sorry, The Field With Unique Value is Getting Duplicated.

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1235
Enter value for ename: SAMPLE
Enter value for job: CLERK
Enter value for salary: 250
Enter value for commission: NULL
Enter value for deptno: 30
Enter value for manager: 7566
Enter value for hiredate: 14-JUL-10
Sorry, The Salary To be Inserted is Less Than The Acceptable Value.

PL/SQL procedure successfully completed.

SQL> /
Enter value for empno: 1235
Enter value for ename: SAMPLE
Enter value for job: CLERK
Enter value for salary: 2500
Enter value for commission: NULL
Enter value for deptno: 30
Enter value for manager: 7566
Enter value for hiredate: 11-JUl-10
Sorry, The Hire Date To be Inserted is Not Acceptable.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> SELECT MGR, COUNT(*) EmpCount
  2  FROM Emp
  3  GROUP BY MGR;

       MGR   EMPCOUNT
---------- ----------
      7566          3
      7698          5
      7782          1
      7788          1
      7839          3
      7902          1
                    1
```

```
7 rows selected.

SQL> DELETE FROM Emp
  2  WHERE Empno = 1234;

1 row deleted.

SQL> COMMIT;

Commit complete.

SQL> COLUMN Empno FORMAT 9999
SQL> COLUMN Sal FORMAT 9999
SQL> COLUMN Comm FORMAT 9999
SQL> COLUMN MGR FORMAT 99
SQL> COLUMN MGR FORMAT 9999
SQL> COLUMN Deptno FORMAT 99
SQL> cl scr

SQL> SELECT * FROM Emp;

EMPNO ENAME      JOB          MGR HIREDATE    SAL  COMM DEPTNO
----- ---------- --------- ----- --------- ----- ----- ------
 7839 KING       PRESIDENT       17-NOV-81  5000           10
 7698 BLAKE      MANAGER    7839 01-MAY-81  2850           30
 7782 CLARK      MANAGER    7839 09-JUN-81  2450           10
 7566 JONES      MANAGER    7839 02-APR-81  2975           20
 7654 MARTIN     SALESMAN   7698 28-SEP-81  1250  1400     30
 7499 ALLEN      SALESMAN   7698 20-FEB-81  1600   300     30
 7844 TURNER     SALESMAN   7698 08-SEP-81  1500     0     30
 7900 JAMES      CLERK      7698 03-DEC-81   950           30
 7521 WARD       SALESMAN   7698 22-FEB-81  1250   500     30
 7902 FORD       ANALYST    7566 03-DEC-81  3000           20
 7369 SMITH      CLERK      7902 17-DEC-80   800           20

EMPNO ENAME      JOB          MGR HIREDATE    SAL  COMM DEPTNO
----- ---------- --------- ----- --------- ----- ----- ------
 7788 SCOTT      ANALYST    7566 09-DEC-82  3000           20
 7876 ADAMS      CLERK      7788 12-JAN-83  1100           20
 7934 MILLER     CLERK      7782 23-JAN-82  1300           10

14 rows selected.

SQL> DECLARE
  2  V_Ename Emp.Ename%TYPE;
  3  V_Job Emp.Job%TYPE;
  4  E_ManyEmployees EXCEPTION;
  5  CURSOR EmpCursor IS
  6  SELECT MGR, COUNT(*) Tot_Emp
  7  FROM Emp
  8  WHERE MGR IS NOT NULL
  9  GROUP BY MGR;
 10  BEGIN
 11  FOR MgrRecord IN EmpCursor
 12  LOOP
```

```
 13  BEGIN
 14  SELECT Ename, Job INTO
 15  V_Ename, V_Job
 16  FROM Emp
 17  WHERE Empno = MgrRecord.Mgr;
 18  IF MgrRecord.Tot_Emp > 3 THEN
 19  RAISE E_ManyEmployees;
 20  ELSE
 21  DBMS_OUTPUT.PUT_LINE('Employee, '||V_Ename||' Manages
'||MgrRecord.Tot_Emp||' Employees.');
 22  END IF;
 23  EXCEPTION
 24  WHEN E_ManyEmployees THEN
 25  DBMS_OUTPUT.PUT_LINE('Employee, '||V_EName||' Manages Many Employees,
Chance of decreasing his Performance, Recommend for Extra Allowances or
Emoluments.');
 26  END;
 27  END LOOP;
 28  END;
 29  /
Employee, JONES Manages 2 Employees.
Employee, BLAKE Manages Many Employees, Chance of decreasing his Performance,
Recommend for Extra Allowances or Emoluments.
Employee, CLARK Manages 1 Employees.
Employee, SCOTT Manages 1 Employees.
Employee, KING Manages 3 Employees.
Employee, FORD Manages 1 Employees.

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> DECLARE
  2  V_Empno Emp.Empno%TYPE := &GEmpno;
  3  V_Ename Emp.Ename%TYPE;
  4  BEGIN
  5  SELECT Ename INTo V_Ename
  6  FROM Emp
  7  WHERE Empno = V_Empno;
  8  EXCEPTION
  9  WHEN NO_DATA_FOUND THEN
 10  DBMS_OUTPUT.PUT_LINE('Sorry Data Not Found');
 11  END;
 12  /
Enter value for gempno: 7839

PL/SQL procedure successfully completed.

SQL> /
Enter value for gempno: 1234
Sorry Data Not Found

PL/SQL procedure successfully completed.

SQL> ED
```

```
Wrote file afiedt.buf

  1  DECLARE
  2  V_Empno Emp.Empno%TYPE := &GEmpno;
  3  V_Ename Emp.Ename%TYPE;
  4  BEGIN
  5  SELECT Ename INTo V_Ename
  6  FROM Emp
  7  WHERE Empno = V_Empno;
  8  EXCEPTION
  9  WHEN NO_DATA_FOUND THEN
 10  RAISE_APPLICATION_ERROR(-20220, 'Sorry Data Not Found');
 11* END;
SQL> /
Enter value for gempno: 1234
DECLARE
*
ERROR at line 1:
ORA-20220: Sorry Data Not Found
ORA-06512: at line 10


SQL> cl scr

SQL> DECLARE
  2  V_Deptno Dept.Deptno%TYPE := &Deptno;
  3  V_TotEmp NUMBER;
  4  E_InvalidDept EXCEPTION;
  5  BEGIN
  6  IF V_Deptno NOT IN (10, 20, 30, 40) THEN
  7  RAISE E_InvalidDept;
  8  ELSE
  9  SELECT COUNT(*) INTO
 10  V_TotEmp FROM Emp
 11  WHERE Deptno = V_Deptno;
 12  DBMS_OUTPUT.PUT_LINE('The Total Employees in '||V_Deptno||' are
'||V_TotEmp||'.');
 13  END IF;
 14  DBMS_OUTPUT.PUT_LINE('No Exception was Raised...');
 15  EXCEPTION
 16  WHEN E_InvalidDept THEN
 17  RAISE_APPLICATION_ERROR(-20220, 'Sorry There is no Such Department...as You
requested.');
 18  END;
 19  /
Enter value for deptno: 30
The Total Employees in 30 are 6.
No Exception was Raised...

PL/SQL procedure successfully completed.

SQL> /
Enter value for deptno: 50
DECLARE
*
```

```
ERROR at line 1:
ORA-20220: Sorry There is no Such Department...as You requested.
ORA-06512: at line 17


SQL> SPOOL OFF
```