

Spool Generated For Class of Oracle By Satish K Yellanki

SQL> cl scr

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

MGR	COUNT(*)
7566	2
7698	5
7782	1
7788	1
7839	3
7902	1
	1

7 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
 22 ||MgrRecord.Tot_Emp||' Employees.');
```

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.

Spool Generated For Class of Oracle By Satish K Yellanki

Employee, KING Manages 3 Employees.
Employee, FORD Manages 1 Employees.

PL/SQL procedure successfully completed.

SQL> cl scr

```
SQL> DECLARE
  2  V_TestVariable CHAR(5) := '&String';
  3  BEGIN
  4  DBMS_OUTPUT.PUT_LINE ('This is A Test Line. ');
  5  DBMS_OUTPUT.PUT_LINE(V_TestVariable);
  6  EXCEPTION
  7  WHEN INVALID_NUMBER OR VALUE_ERROR THEN
  8  DBMS_OUTPUT.PUT_LINE('An Error Raised. ');
  9  WHEN OTHERS THEN
 10  DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
 11  END;
 12  /
```

Enter value for string: Snake

This is A Test Line.

Snake

PL/SQL procedure successfully completed.

SQL> /

Enter value for string: Sample

```
DECLARE
```

```
*
```

ERROR at line 1:

ORA-06502: PL/SQL: numeric or value error: character string buffer too small

ORA-06512: at line 2

SQL> ED

Wrote file afiedt.buf

```
 1  BEGIN
 2  DECLARE
 3  V_TestVariable CHAR(5) := '&String';
 4  BEGIN
 5  DBMS_OUTPUT.PUT_LINE ('This is A Test Line. ');
 6  DBMS_OUTPUT.PUT_LINE(V_TestVariable);
 7  EXCEPTION
 8  WHEN INVALID_NUMBER OR VALUE_ERROR THEN
 9  DBMS_OUTPUT.PUT_LINE('An Error Raised. ');
10  WHEN OTHERS THEN
11  DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
12  END;
13  EXCEPTION
14  WHEN INVALID_NUMBER OR VALUE_ERROR THEN
15  DBMS_OUTPUT.PUT_LINE('An Error Raised. ');
16  WHEN OTHERS THEN
17  DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
18* END;
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> /  
Enter value for string: Snake  
This is A Test Line.  
Snake
```

PL/SQL procedure successfully completed.

```
SQL> /  
Enter value for string: Sample  
An Error Raised.
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> ED  
Wrote file afiedt.buf
```

```
1 BEGIN  
2 DECLARE  
3 V_TestVariable CHAR(5) := '&String';  
4 BEGIN  
5 DBMS_OUTPUT.PUT_LINE ('This is A Test Line.');6 DBMS_OUTPUT.PUT_LINE(V_TestVariable);  
7 EXCEPTION  
8 WHEN INVALID_NUMBER OR VALUE_ERROR THEN  
9 DBMS_OUTPUT.PUT_LINE('An Error Raised.');10 WHEN OTHERS THEN  
11 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');12 END;  
13 EXCEPTION  
14 WHEN INVALID_NUMBER OR VALUE_ERROR THEN  
15 DBMS_OUTPUT.PUT_LINE('An Error Raised.');16 WHEN OTHERS THEN  
17 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');18* END;
```

```
SQL> /  
Enter value for string: Sample  
An Error Raised.
```

PL/SQL procedure successfully completed.

```
SQL> ED  
Wrote file afiedt.buf
```

```
1 BEGIN  
2 DECLARE  
3 V_TestVariable CHAR(5) := '&String';  
4 BEGIN  
5 DBMS_OUTPUT.PUT_LINE ('This is A Test Line.');6 DBMS_OUTPUT.PUT_LINE(V_TestVariable);  
7 EXCEPTION  
8 WHEN INVALID_NUMBER OR VALUE_ERROR THEN  
9 DBMS_OUTPUT.PUT_LINE('An Error Raised.');10 WHEN OTHERS THEN
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
11  DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
12  END;
13  EXCEPTION
14  WHEN INVALID_NUMBER OR VALUE_ERROR THEN
15  RAISE_APPLICATION_ERROR(-20001, 'The Actual Size of The Memory Allocated is
'|VSIZE(V_TestVariable)||' Bytes, You Are Trying To Occupy
'|LENGTH(V_TestVariable)||' Characters...');
16  WHEN OTHERS THEN
17  DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
18* END;
```

SQL> /

Enter value for string: Snake

```
RAISE_APPLICATION_ERROR(-20001, 'The Actual Size of The Memory Allocated is
'|VSIZE(V_TestVariable)||' Bytes, You Are Trying To Occupy
'|LENGTH(V_TestVariable)||' Characters...');
```

*

ERROR at line 15:

ORA-06550: line 15, column 86:

PLS-00201: identifier 'V_TESTVARIABLE' must be declared

ORA-06550: line 15, column 1:

PL/SQL: Statement ignored

SQL> ED

Wrote file afiedt.buf

```
1  DECLARE
2  V_INTTestVariable CHAR(5);
3  BEGIN
4  DECLARE
5  V_TestVariable CHAR(5) := '&String';
6  BEGIN
7  DBMS_OUTPUT.PUT_LINE ('This is A Test Line. ');
8  DBMS_OUTPUT.PUT_LINE(V_TestVariable);
9  V_INTTestVariable := V_TestVariable;
10 EXCEPTION
11 WHEN INVALID_NUMBER OR VALUE_ERROR THEN
12 DBMS_OUTPUT.PUT_LINE('An Error Raised. ');
13 WHEN OTHERS THEN
14 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
15 END;
16 EXCEPTION
17 WHEN INVALID_NUMBER OR VALUE_ERROR THEN
18 RAISE_APPLICATION_ERROR(-20001, 'The Actual Size of The Memory Allocated is
'|VSIZE(V_INTTestVariable)||' Bytes, You Are Trying To Occupy 6 Characters...');
19 WHEN OTHERS THEN
20 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
21* END;
```

SQL> /

Enter value for string: Snake

```
RAISE_APPLICATION_ERROR(-20001, 'The Actual Size of The Memory Allocated is
'|VSIZE(V_INTTestVariable)||' Bytes, You Are Trying To Occupy 6 Characters...');
```

*

ERROR at line 18:

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
ORA-06550: line 18, column 80:
PLS-00204: function or pseudo-column 'VSIZE' may be used inside a SQL statement
only
ORA-06550: line 18, column 1:
PL/SQL: Statement ignored
```

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

```
1 DECLARE
2 V_INTestVariable CHAR(5);
3 V_Size NUMBER;
4 V_Message VARCHAR2(200);
5 BEGIN
6 DECLARE
7 V_TestVariable CHAR(5) := '&String';
8 BEGIN
9 DBMS_OUTPUT.PUT_LINE ('This is A Test Line. ');
10 DBMS_OUTPUT.PUT_LINE(V_TestVariable);
11 V_INTestVariable := V_TestVariable;
12 EXCEPTION
13 WHEN INVALID_NUMBER OR VALUE_ERROR THEN
14 DBMS_OUTPUT.PUT_LINE('An Error Raised. ');
15 WHEN OTHERS THEN
16 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
17 END;
18 EXCEPTION
19 WHEN INVALID_NUMBER OR VALUE_ERROR THEN
20 V_Size := VSIZE(V_INTestVariable);
21 V_Message := 'The Actual Size of The Memory Allocated is '||V_Size||' Bytes,
You Are Trying To Occupy 6 Characters...';
22 RAISE_APPLICATION_ERROR(-20001, V_Size);
23 WHEN OTHERS THEN
24 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
25* END;
```

```
SQL> /
Enter value for string: Snake
V_Size := VSIZE(V_INTestVariable);
*
```

```
ERROR at line 20:
ORA-06550: line 20, column 11:
PLS-00204: function or pseudo-column 'VSIZE' may be used inside a SQL statement
only
ORA-06550: line 20, column 1:
PL/SQL: Statement ignored
```

```
SQL> ED
Wrote file afiedt.buf
```

```
1 DECLARE
2 V_INTestVariable CHAR(5);
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
3 V_Size NUMBER;
4 V_Message VARCHAR2(200);
5 BEGIN
6 DECLARE
7 V_TestVariable CHAR(5) := '&String';
8 BEGIN
9 DBMS_OUTPUT.PUT_LINE ('This is A Test Line. ');
10 DBMS_OUTPUT.PUT_LINE(V_TestVariable);
11 V_INTestVariable := V_TestVariable;
12 EXCEPTION
13 WHEN INVALID_NUMBER OR VALUE_ERROR THEN
14 DBMS_OUTPUT.PUT_LINE('An Error Raised. ');
15 WHEN OTHERS THEN
16 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
17 END;
18 EXCEPTION
19 WHEN INVALID_NUMBER OR VALUE_ERROR THEN
20 SELECT VSIZE(V_INTestVariable) INTO V_Size
21 FROM DUAL;
22 V_Message := 'The Actual Size of The Memory Allocated is '||V_Size||' Bytes,
You Are Trying To Occupy 6 Characters...';
23 RAISE_APPLICATION_ERROR(-20001, V_Size);
24 WHEN OTHERS THEN
25 DBMS_OUTPUT.PUT_LINE('Some Error, Please Check');
26* END;
```

SQL> /

Enter value for string: Snake

This is A Test Line.

Snake

PL/SQL procedure successfully completed.

SQL> /

Enter value for string: Sample

DECLARE

*

ERROR at line 1:

ORA-20001:

ORA-06512: at line 23

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;
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
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: 20
The Total Employees in 20 are 5.
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> 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_NotNULLViolation EXCEPTION;
23 PRAGMA EXCEPTION_INIT(E_NotNULLViolation, -1400);
24 E_CheckViolation EXCEPTION;
25 PRAGMA EXCEPTION_INIT(E_CheckViolation, -2290);
26 BEGIN
27 ValidateRecordValues.SalaryLess := 500;
28 ValidateRecordValues.SalaryMore := 150000;
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
29  SELECT
30  SYSDATE
31  INTO
32  ValidateRecordValues.CurrentDate
33  FROM DUAL;
34  ValidateRecordValues.CurrentDate :=
TO_DATE(TO_CHAR(ValidateRecordValues.CurrentDate, 'DD-MON-YY'), 'DD-MON-YY');
35  SELECT
36  COUNT(*)
37  INTO
38  ValidateRecordValues.ValidateJob
39  FROM Emp
40  WHERE Job = V_Job;
41  SELECT
42  COUNT(*)
43  INTO
44  ValidateRecordValues.ValidateDeptno
45  FROM Dept
46  WHERE Deptno = V_Deptno;
47  SELECT
48  COUNT(*)
49  INTO
50  ValidateRecordValues.ValidateManager
51  FROM Emp
52  WHERE MGR = V_MGR;
53  IF ValidateRecordValues.ValidateJob = 0
54  THEN
55  RAISE_APPLICATION_ERROR(-20100, 'Sorry, The Job To be Inserted is Not
Acceptable.');
```

```
56  ROLLBACK;
57  ELSIF ValidateRecordValues.ValidateDeptno = 0
58  THEN
59  RAISE_APPLICATION_ERROR(-20101, 'Sorry, The Department Number To be
Inserted is Not Acceptable.');
```

```
60  ROLLBACK;
61  ELSIF ValidateRecordValues.ValidateManager = 0
62  THEN
63  RAISE_APPLICATION_ERROR(-20102, 'Sorry, The Manager Number To be Inserted
is Not Acceptable.');
```

```
64  ROLLBACK;
65  END IF;
66  IF V_Sal < ValidateRecordValues.SalaryLess
67  THEN
68  RAISE_APPLICATION_ERROR(-20103, 'Sorry, The Salary Value is Less Than
Expected, Hence Insert is Not Acceptable.');
```

```
69  ROLLBACK;
70  ELSIF V_Sal > ValidateRecordValues.SalaryMore
71  THEN
72  RAISE_APPLICATION_ERROR(-20104, 'Sorry, The Salary Value is More Than
Expected, Hence Insert is Not Acceptable.');
```

```
73  ROLLBACK;
74  END IF;
75  IF ValidateRecordValues.CurrentDate <> V_HireDate
76  THEN
```


Spool Generated For Class of Oracle By Satish K Yellanki

```
77  RAISE_APPLICATION_ERROR(-20105, 'Sorry, The Hire Date Value is Above The
System Requirements, Hence Insert is Not Acceptable. ');
78  ROLLBACK;
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  RAISE_APPLICATION_ERROR(-20106, 'A Field with Check Constraint is not
Attended Properly, Please Check Properly. ');
89  ROLLBACK;
90  WHEN E_NotNULLViolation THEN
91  RAISE_APPLICATION_ERROR(-20107, 'A Field which Cannot be NULL, is not
attended, Please Check Properly. ');
92  ROLLBACK;
93  END;
94  /
```

```
Enter value for empno: 7839
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: 30-AUG-10
Sorry, The Field With Unique Value is Getting Duplicated.
```

PL/SQL procedure successfully completed.

```
SQL> /
Enter value for empno: NULL
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: 30-AUG-10
```

DECLARE

*

```
ERROR at line 1:
ORA-20107: A Field which Cannot be NULL, is not attended, Please Check
Properly.
ORA-06512: at line 91
```

```
SQL> /
Enter value for empno: 1234
Enter value for ename: Sample
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
Enter value for job: SRCLERK
Enter value for salary: 2000
Enter value for commission: NULL
Enter value for deptno: 30
Enter value for manager: 7566
Enter value for hiredate: 30-AUG-10
DECLARE
*
ERROR at line 1:
ORA-20100: Sorry, The Job To be Inserted is Not Acceptable.
ORA-06512: at line 55
```

```
SQL> cl scr
```

```
SQL> <<EmpSearch>>
 2  DECLARE
 3  V-Empno Emp.Empno%TYPE := &PEmpno;
 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(V_Ename);
10  EXCEPTION
11  WHEN NO_DATA_FOUND THEN
12  DBMS_OUTPUT.PUT_LINE('Sorry!');
13  END EmpSearch;
14  /
Enter value for pempno: 7839
V-Empno Emp.Empno%TYPE := 7839;
*
ERROR at line 3:
ORA-06550: line 3, column 2:
PLS-00103: Encountered the symbol "-" when expecting one of the following:
constant exception <an identifier>
<a double-quoted delimited-identifier> table LONG_ double ref
char time timestamp interval date binary national character
nchar
```

```
SQL> ED
```

```
Wrote file afiedt.buf
```

```
 1  <<EmpSearch>>
 2  DECLARE
 3  V_Empno Emp.Empno%TYPE := &PEmpno;
 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(V_Ename);
10  EXCEPTION
11  WHEN NO_DATA_FOUND THEN
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
12 DBMS_OUTPUT.PUT_LINE('Sorry!');
13* END EmpSearch;
SQL> /
Enter value for pempno: 7839
KING
```

PL/SQL procedure successfully completed.

```
SQL> ED
Wrote file afiedt.buf
```

```
1 CREATE OR REPLACE PROCEDURE
2 EmpSearch(PEmpno Emp.Empno%TYPE)
3 IS
4 V_Empno Emp.Empno%TYPE := PEmpno;
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(V_Ename);
11 EXCEPTION
12 WHEN NO_DATA_FOUND THEN
13 DBMS_OUTPUT.PUT_LINE('Sorry!');
14* END EmpSearch;
SQL> /
```

Procedure created.

```
SQL> EXEC EmpSearch(7839);
KING
```

PL/SQL procedure successfully completed.

```
SQL> SPOOL OFF
SQL> cl scr
```

```
SQL> DESC USER_OBJECTS
```

Name	Null?	Type
OBJECT_NAME		VARCHAR2(128)
SUBOBJECT_NAME		VARCHAR2(30)
OBJECT_ID		NUMBER
DATA_OBJECT_ID		NUMBER
OBJECT_TYPE		VARCHAR2(19)
CREATED		DATE
LAST_DDL_TIME		DATE
TIMESTAMP		VARCHAR2(19)
STATUS		VARCHAR2(7)
TEMPORARY		VARCHAR2(1)
GENERATED		VARCHAR2(1)
SECONDARY		VARCHAR2(1)

```
SQL> SELECT OBJECT_NAME, OBJECT_TYPE
2 FROM USER_OBJECTS;
```

Spool Generated For Class of Oracle By Satish K Yellanki

OBJECT_NAME

OBJECT_TYPE

ORDID
SEQUENCE

PRODID
SEQUENCE

CUSTID
SEQUENCE

OBJECT_NAME

OBJECT_TYPE

MESSAGES
TABLE

EMPSEARCH
PROCEDURE

EMPLOYEE
SYNONYM

OBJECT_NAME

OBJECT_TYPE

EMPENAME
VIEW

EMPLOYEEENAMES
SYNONYM

EMPENAMEJOB
VIEW

OBJECT_NAME

OBJECT_TYPE

SAMPLESEQ01
SEQUENCE

EMPNAMEJOBS
SYNONYM

EMPENAMES
VIEW

Spool Generated For Class of Oracle By Satish K Yellanki

OBJECT_NAME

OBJECT_TYPE

SAMPLESEQ02
SEQUENCE

NAMEATLEASTWITHONEA
SYNONYM

SAMPLESEQ03
SEQUENCE

OBJECT_NAME

OBJECT_TYPE

EMPENAMES01
VIEW

SAMPLESEQ04
SEQUENCE

NAMESTARTINGWITHS
SYNONYM

OBJECT_NAME

OBJECT_TYPE

SAMPLESEQ05
SEQUENCE

MYAUDIT
TABLE

SAMPLESEQ06
SEQUENCE

OBJECT_NAME

OBJECT_TYPE

TRAPPEDMESSAGES
TABLE

DEPT
TABLE

DEPT_PRIMARY_KEY

Spool Generated For Class of Oracle By Satish K Yellanki

INDEX

OBJECT_NAME

OBJECT_TYPE

EMP
TABLE

SAMPLESEQ07
SEQUENCE

EMP_PRIMARY_KEY
INDEX

OBJECT_NAME

OBJECT_TYPE

BONUS
TABLE

SALGRADE
TABLE

DUMMY
TABLE

OBJECT_NAME

OBJECT_TYPE

CUSTOMER
TABLE

CUSTOMER_PRIMARY_KEY
INDEX

ORD
TABLE

OBJECT_NAME

OBJECT_TYPE

ORD_PRIMARY_KEY
INDEX

ITEM
TABLE

Spool Generated For Class of Oracle By Satish K Yellanki

```
ITEM_PRIMARY_KEY  
INDEX
```

```
OBJECT_NAME  
-----
```

```
OBJECT_TYPE  
-----
```

```
PRODUCT  
TABLE
```

```
PRODUCT_PRIMARY_KEY  
INDEX
```

```
PRICE  
TABLE
```

```
OBJECT_NAME  
-----
```

```
OBJECT_TYPE  
-----
```

```
PRICE_INDEX  
INDEX
```

```
SALES  
VIEW
```

41 rows selected.

```
SQL> cl scr
```

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> SET VERIFY OFF
```

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PROCEDURE
```

```
 2 MyBonus  
 3 AS  
 4 CURSOR DeptCursor IS  
 5 SELECT Deptno FROM Dept;  
 6 BEGIN  
 7 FOR R_GroupBonus IN DeptCursor LOOP  
 8 UPDATE Emp  
 9 SET Sal = Sal * 0.95  
10 WHERE Deptno = R_GroupBonus.DeptNo;  
11 DBMS_OUTPUT.PUT_LINE('The Bonus Information is '||R_GroupBonus.Deptno);  
12 END LOOP;  
13 END MyBonus;  
14 /
```

Procedure created.

```
SQL> COLUMN Empno FORMAT 9999
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> COLUMN Sal FORMAT 9999
SQL> COLUMN Comm FORMAT 9999
SQL> COLUMN MGR FORMAT 9999
SQL> COLUMN Sal FORMAT 999999
SQL> SELECT Ename, Sal, Deptno
  2 FROM Emp
  3 ORDER BY Deptno;
```

ENAME	SAL	DEPTNO
KING	5000	10
CLARK	2450	10
MILLER	1300	10
JONES	2975	20
SCOTT	3000	20
ADAMS	1100	20
SMITH	800	20
FORD	3000	20
BLAKE	2850	30
MARTIN	1250	30
ALLEN	1600	30

ENAME	SAL	DEPTNO
TURNER	1500	30
JAMES	950	30
WARD	1250	30

14 rows selected.

```
SQL> cl scr
```

```
SQL> DESC USER_OBJECTS
```

Name	Null?	Type
OBJECT_NAME		VARCHAR2(128)
SUBOBJECT_NAME		VARCHAR2(30)
OBJECT_ID		NUMBER
DATA_OBJECT_ID		NUMBER
OBJECT_TYPE		VARCHAR2(19)
CREATED		DATE
LAST_DDL_TIME		DATE
TIMESTAMP		VARCHAR2(19)
STATUS		VARCHAR2(7)
TEMPORARY		VARCHAR2(1)
GENERATED		VARCHAR2(1)
SECONDARY		VARCHAR2(1)

```
SQL> COLUMN OBJECT_NAME FORMAT A15
SQL> COLUMN OBJECT_TYPE FORMAT A15
SQL> COLUMN STATUS FORMAT A10
SQL> SELECT OBJECT_NAME, OBJECT_TYPE, STATUS
  2 FROM USER_OBJECTS
  3 WHERE OBJECT_TYPE = 'PROCEDURE';
```


Spool Generated For Class of Oracle By Satish K Yellanki

OBJECT_NAME	OBJECT_TYPE	STATUS
EMPSEARCH	PROCEDURE	INVALID
MYBONUS	PROCEDURE	VALID

```
SQL> ALTER PROCEDURE EMPSEARCH COMPILE;
```

Procedure altered.

```
SQL> SELECT OBJECT_NAME, OBJECT_TYPE, STATUS
2 FROM USER_OBJECTS
3 WHERE OBJECT_TYPE = 'PROCEDURE';
```

OBJECT_NAME	OBJECT_TYPE	STATUS
EMPSEARCH	PROCEDURE	VALID
MYBONUS	PROCEDURE	VALID

```
SQL> BEGIN
2 EmpSearch(&GEmpno);
3 END;
4 /
```

Enter value for gempno: 7839
KING

PL/SQL procedure successfully completed.

```
SQL> /
Enter value for gempno: 7654
MARTIN
```

PL/SQL procedure successfully completed.

```
SQL> /
Enter value for gempno: 1234
Sorry!
```

PL/SQL procedure successfully completed.

```
SQL> EXEC EmpSearch(&GEmpno);
Enter value for gempno: 7839
KING
```

PL/SQL procedure successfully completed.

```
SQL> /
Enter value for gempno: 7654
MARTIN
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> SELECT OBJECT_NAME, OBJECT_TYPE, STATUS
2 FROM USER_OBJECTS
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
3 WHERE OBJECT_TYPE = 'PROCEDURE';
```

OBJECT_NAME	OBJECT_TYPE	STATUS
EMPSEARCH	PROCEDURE	VALID
MYBONUS	PROCEDURE	VALID

```
SQL> SELECT Ename, Sal
2 FROM Emp
3 ORDER BY Deptno;
```

ENAME	SAL
KING	5000
CLARK	2450
MILLER	1300
JONES	2975
SCOTT	3000
ADAMS	1100
SMITH	800
FORD	3000
BLAKE	2850
MARTIN	1250
ALLEN	1600

ENAME	SAL
TURNER	1500
JAMES	950
WARD	1250

14 rows selected.

```
SQL> EXEC MYBONUS
The Bonus Information is 10
The Bonus Information is 20
The Bonus Information is 30
The Bonus Information is 40
```

PL/SQL procedure successfully completed.

```
SQL> SELECT Ename, Sal
2 FROM Emp
3 ORDER BY Deptno;
```

ENAME	SAL
KING	4750
CLARK	2328
MILLER	1235
JONES	2826
SCOTT	2850
ADAMS	1045
SMITH	760
FORD	2850

Spool Generated For Class of Oracle By Satish K Yellanki

```
BLAKE      2708
MARTIN     1188
ALLEN      1520
```

```
ENAME      SAL
-----
TURNER     1425
JAMES      903
WARD       1188
```

14 rows selected.

SQL> cl scr

SQL> DESC USER_SOURCE

Name	Null?	Type
NAME		VARCHAR2(30)
TYPE		VARCHAR2(12)
LINE		NUMBER
TEXT		VARCHAR2(4000)

SQL> COLUMN TEXT FORMAT A40

```
SQL> SELECT TO_CHAR(Line, 99)|| '>', Text
2 FROM USER_SOURCE
3 WHERE NAME = 'EMPSEARCH';
```

TO_C TEXT

```
-----
1> PROCEDURE
2> EmpSearch(PEmpno Emp.Empno%TYPE)
3> IS
4> V_Empno Emp.Empno%TYPE := PEmpno;
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(V_Ename);
11> EXCEPTION
```

TO_C TEXT

```
-----
12> WHEN NO_DATA_FOUND THEN
13> DBMS_OUTPUT.PUT_LINE('Sorry!');
14> END EmpSearch;
```

14 rows selected.

SQL> ED

Wrote file afiedt.buf

```
1 SELECT TO_CHAR(Line, 99)|| '>', Text
2 FROM USER_SOURCE
3* WHERE NAME = 'MYBONUS'
```

Spool Generated For Class of Oracle By Satish K Yellanki

SQL> /

TO_C TEXT

```
-----  
1> PROCEDURE  
2> MyBonus  
3> AS  
4> CURSOR DeptCursor IS  
5> SELECT Deptno FROM Dept;  
6> BEGIN  
7> FOR R_GroupBonus IN DeptCursor LOOP  
8> UPDATE Emp  
9> SET Sal = Sal * 0.95  
10> WHERE Deptno = R_GroupBonus.DeptNo;  
11> DBMS_OUTPUT.PUT_LINE('The Bonus Informat
```

TO_C TEXT

```
-----  
ion is '||R_GroupBonus.Deptno);  
  
12> END LOOP;  
13> END MyBonus;
```

13 rows selected.

SQL> ROLLBACK;

2

SQL> ROLLBACK;

Rollback complete.

SQL> cl scr

SQL> SELECT Ename, Sal

2 FROM Emp;

ENAME	SAL
KING	5000
BLAKE	2850
CLARK	2450
JONES	2975
MARTIN	1250
ALLEN	1600
TURNER	1500
JAMES	950
WARD	1250
FORD	3000
SMITH	800

ENAME	SAL
SCOTT	3000
ADAMS	1100
MILLER	1300

Spool Generated For Class of Oracle By Satish K Yellanki

14 rows selected.

SQL> cl scr

```
SQL> CREATE OR REPLACE PROCEDURE
  2  EmpBonus
  3  AS
  4  CURSOR DeptCursor IS
  5  SELECT Deptno FROM Dept;
  6  BEGIN
  7  FOR R_GroupBonus IN DeptCursor
  8  LOOP
  9  DECLARE
 10  E_JobNotFound EXCEPTION;
 11  CURSOR EmpCursor IS
 12  SELECT *
 13  FROM Emp
 14  WHERE Deptno = R_GroupBonus.Deptno;
 15  BEGIN
 16  FOR R_EmpCursor IN EmpCursor
 17  LOOP
 18  IF R_EmpCursor.Job = 'PRESIDENT'
 19  THEN
 20  UPDATE Emp
 21  SET Sal = Sal + (Sal * 0.40)
 22  WHERE Empno = R_EmpCursor.Empno;
 23  DBMS_OUTPUT.PUT_LINE('The Job For Updation is '||R_EmpCursor.Job);
 24  DBMS_OUTPUT.PUT_LINE('Old Sal : '||R_EmpCursor.Sal);
 25  DBMS_OUTPUT.PUT_LINE('Increment Added : '||((R_EmpCursor.Sal +
(R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
 26  DBMS_OUTPUT.PUT_LINE('The Salary With Increment is : '||((R_EmpCursor.Sal +
(R_EmpCursor.Sal * 0.40)));
 27  ELSE
 28  IF R_EmpCursor.Job = 'MANAGER'
 29  THEN
 30  UPDATE Emp
 31  SET Sal = Sal + (Sal * 0.35)
 32  WHERE Empno = R_EmpCursor.Empno;
 33  DBMS_OUTPUT.PUT_LINE('The Job For Updation is '||R_EmpCursor.Job);
 34  DBMS_OUTPUT.PUT_LINE('Old Sal : '||R_EmpCursor.Sal);
 35  DBMS_OUTPUT.PUT_LINE('Increment Added : '||((R_EmpCursor.Sal +
(R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
 36  DBMS_OUTPUT.PUT_LINE('The Salary With Increment is :
' || (R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
 37  IF R_EmpCursor.Job = 'ANALYST'
 38  THEN
 39  UPDATE Emp
 40  SET Sal = Sal + (Sal * 0.30)
 41  WHERE Empno = R_EmpCursor.Empno;
 42  DBMS_OUTPUT.PUT_LINE('The Job For Updation is
' || R_EmpCursor.Job);
 43  DBMS_OUTPUT.PUT_LINE('Old Sal : '||R_EmpCursor.Sal);
 44  DBMS_OUTPUT.PUT_LINE('Increment Added : '||((R_EmpCursor.Sal
+ (R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
45          DBMS_OUTPUT.PUT_LINE('The Salary With Increment is :
'| |(R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
46          ELSE
47              IF R_EmpCursor.Job = 'SALESMAN'
48              THEN
49                  UPDATE Emp
50                  SET Sal = Sal + (Sal * 0.25)
51                  WHERE Empno = R_EmpCursor.Empno;
52                  DBMS_OUTPUT.PUT_LINE('The Job For Updation is
'| |R_EmpCursor.Job);
53                  DBMS_OUTPUT.PUT_LINE('Old Sal : '| |R_EmpCursor.Sal);
54                  DBMS_OUTPUT.PUT_LINE('Increment Added :
'| |((R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
55                  DBMS_OUTPUT.PUT_LINE('The Salary With Increment is :
'| |(R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
56          ELSE
57              IF R_EmpCursor.Job = 'CLERK'
58              THEN
59                  UPDATE Emp
60                  SET Sal = Sal + (Sal * 0.20)
61                  WHERE Empno = R_EmpCursor.Empno;
62              ELSE
63                  DBMS_OUTPUT.PUT_LINE('The Job For Updation is
'| |R_EmpCursor.Job);
64                  DBMS_OUTPUT.PUT_LINE('Old Sal :
'| |R_EmpCursor.Sal);
65                  DBMS_OUTPUT.PUT_LINE('Increment Added :
'| |((R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
66                  DBMS_OUTPUT.PUT_LINE('The Salary With Increment
is : '| |(R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
67              END IF;
68          END IF;
69      END IF;
70  END IF;
71  END IF;
72  END LOOP;
73  EXCEPTION
74  WHEN E_JobNotFound THEN
75      DBMS_OUTPUT.PUT_LINE('The Respective Job is Not Found...Please Verify!');
76  END;
77  DBMS_OUTPUT.PUT_LINE('The Bonus Information is '| |R_GroupBonus.Deptno);
78  END LOOP;
79  END EmpBonus;
80  /
```

Procedure created.

```
SQL> EXEC EmpBonus;
The Job For Updation is PRESIDENT
Old Sal : 5000
Increment Added : 2000
The Salary With Increment is : 7000
The Job For Updation is MANAGER
Old Sal : 2450
Increment Added : 980
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
The Salary With Increment is : 3430
The Job For Updation is MANAGER
Old Sal : 2450
Increment Added : 980
The Salary With Increment is : 3430
The Bonus Information is 10
The Job For Updation is MANAGER
Old Sal : 2975
Increment Added : 1190
The Salary With Increment is : 4165
The Job For Updation is MANAGER
Old Sal : 2975
Increment Added : 1190
The Salary With Increment is : 4165
The Bonus Information is 20
The Job For Updation is MANAGER
Old Sal : 2850
Increment Added : 1140
The Salary With Increment is : 3990
The Job For Updation is MANAGER
Old Sal : 2850
Increment Added : 1140
The Salary With Increment is : 3990
The Bonus Information is 30
The Bonus Information is 40
```

PL/SQL procedure successfully completed.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PROCEDURE
  2  EmpBonus
  3  AS
  4  CURSOR DeptCursor IS
  5  SELECT Deptno FROM Dept;
  6  BEGIN
  7  FOR R_GroupBonus IN DeptCursor
  8  LOOP
  9  DECLARE
 10  E_JobNotFound EXCEPTION;
 11  CURSOR EmpCursor IS
 12  SELECT *
 13  FROM Emp
 14  WHERE Deptno = R_GroupBonus.Deptno;
 15  BEGIN
 16  FOR R_EmpCursor IN EmpCursor
 17  LOOP
 18  IF R_EmpCursor.Job = 'PRESIDENT'
 19  THEN
 20  UPDATE Emp
 21  SET Sal = Sal + (Sal * 0.40)
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
22 WHERE Empno = R_EmpCursor.Empno;
23 DBMS_OUTPUT.PUT_LINE('The Job For Updation is '||R_EmpCursor.Job);
24 DBMS_OUTPUT.PUT_LINE('Old Sal : '||R_EmpCursor.Sal);
25 DBMS_OUTPUT.PUT_LINE('Increament Added : '||((R_EmpCursor.Sal +
(R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
26 DBMS_OUTPUT.PUT_LINE('The Salary With Increment is : '||((R_EmpCursor.Sal +
(R_EmpCursor.Sal * 0.40)));
27 ELSE
28     IF R_EmpCursor.Job = 'MANAGER'
29     THEN
30     UPDATE Emp
31     SET Sal = Sal + (Sal * 0.35)
32     WHERE Empno = R_EmpCursor.Empno;
33     DBMS_OUTPUT.PUT_LINE('The Job For Updation is '||R_EmpCursor.Job);
34     DBMS_OUTPUT.PUT_LINE('Old Sal : '||R_EmpCursor.Sal);
35     DBMS_OUTPUT.PUT_LINE('Increament Added : '||((R_EmpCursor.Sal +
(R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
36     DBMS_OUTPUT.PUT_LINE('The Salary With Increment is :
' ||((R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
37     IF R_EmpCursor.Job = 'ANALYST'
38     THEN
39     UPDATE Emp
40     SET Sal = Sal + (Sal * 0.30)
41     WHERE Empno = R_EmpCursor.Empno;
42     DBMS_OUTPUT.PUT_LINE('The Job For Updation is
' ||R_EmpCursor.Job);
43     DBMS_OUTPUT.PUT_LINE('Old Sal : '||R_EmpCursor.Sal);
44     DBMS_OUTPUT.PUT_LINE('Increament Added : '||((R_EmpCursor.Sal
+ (R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
45     DBMS_OUTPUT.PUT_LINE('The Salary With Increment is :
' ||((R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
46     ELSE
47     IF R_EmpCursor.Job = 'SALESMAN'
48     THEN
49     UPDATE Emp
50     SET Sal = Sal + (Sal * 0.25)
51     WHERE Empno = R_EmpCursor.Empno;
52     DBMS_OUTPUT.PUT_LINE('The Job For Updation is
' ||R_EmpCursor.Job);
53     DBMS_OUTPUT.PUT_LINE('Old Sal : '||R_EmpCursor.Sal);
54     DBMS_OUTPUT.PUT_LINE('Increament Added :
' ||((R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
55     DBMS_OUTPUT.PUT_LINE('The Salary With Increment is :
' ||((R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
56     ELSE
57     IF R_EmpCursor.Job = 'CLERK'
58     THEN
59     UPDATE Emp
60     SET Sal = Sal + (Sal * 0.20)
61     WHERE Empno = R_EmpCursor.Empno;
62     ELSE
63     DBMS_OUTPUT.PUT_LINE('The Job For Updation is
' ||R_EmpCursor.Job);
64     DBMS_OUTPUT.PUT_LINE('Old Sal :
' ||R_EmpCursor.Sal);
```


Spool Generated For Class of Oracle By Satish K Yellanki

```
65          DBMS_OUTPUT.PUT_LINE('Increment Added :
'|((R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)) - (R_EmpCursor.Sal)));
66          DBMS_OUTPUT.PUT_LINE('The Salary With Increment
is : '|(R_EmpCursor.Sal + (R_EmpCursor.Sal * 0.40)));
67          END IF;
68          END IF;
69          END IF;
70          END IF;
71      END IF;
72      END LOOP;
73      EXCEPTION
74      WHEN E_JobNotFound THEN
75      DBMS_OUTPUT.PUT_LINE('The Respective Job is Not Found...Please Verify!');
76      END;
77      DBMS_OUTPUT.PUT_LINE('The Bonus Information is '|R_GroupBonus.Deptno);
78      END LOOP;
79      END EmpBonus;
80      /
```

Procedure created.

```
SQL> EXEC EmpBonus
The Job For Updation is PRESIDENT
Old Sal : 5000
Increment Added : 2000
The Salary With Increment is : 7000
The Job For Updation is MANAGER
Old Sal : 2450
Increment Added : 980
The Salary With Increment is : 3430
The Job For Updation is MANAGER
Old Sal : 2450
Increment Added : 980
The Salary With Increment is : 3430
The Bonus Information is 10
The Job For Updation is MANAGER
Old Sal : 2975
Increment Added : 1190
The Salary With Increment is : 4165
The Job For Updation is MANAGER
Old Sal : 2975
Increment Added : 1190
The Salary With Increment is : 4165
The Bonus Information is 20
The Job For Updation is MANAGER
Old Sal : 2850
Increment Added : 1140
The Salary With Increment is : 3990
The Job For Updation is MANAGER
Old Sal : 2850
Increment Added : 1140
The Salary With Increment is : 3990
The Bonus Information is 30
The Bonus Information is 40
```

Spool Generated For Class of Oracle By Satish K Yellanki

PL/SQL procedure successfully completed.

SQL> cl scr

SQL> SELECT Ename, Sal FROM Emp;

ENAME	SAL
KING	7000
BLAKE	3848
CLARK	3308
JONES	4016
MARTIN	1250
ALLEN	1600
TURNER	1500
JAMES	950
WARD	1250
FORD	3000
SMITH	800

ENAME	SAL
SCOTT	3000
ADAMS	1100
MILLER	1300

14 rows selected.

SQL> cl scr

```
SQL> CREATE OR REPLACE PROCEDURE
  2  GetEnameSalJob00
  3  (
  4    PEmpno      Emp.Empno%TYPE
  5  )
  6  AS
  7    V_Ename      Emp.Ename%TYPE;
  8    V_Sal        Emp.Sal%TYPE;
  9    V_Job        Emp.Job%TYPE;
 10  BEGIN
 11  SELECT
 12    Ename, Sal, Job
 13  INTO
 14    V_Ename, V_Sal, V_Job
 15  FROM Emp
 16  WHERE Empno = PEmpno;
 17  DBMS_OUTPUT.PUT_LINE('The Details of Employee '||PEmpno||' Are...');
 18  DBMS_OUTPUT.PUT_LINE('The Name of The Employee is : '||V_Ename);
 19  DBMS_OUTPUT.PUT_LINE('The Salary of The Employee is : '||V_Sal);
 20  DBMS_OUTPUT.PUT_LINE('The Job of The Employee is : '||V_Job);
 21  END GetEnameSalJob00;
 22  /
```

Procedure created.

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> EXEC GetEnameSalJob00;  
BEGIN GetEnameSalJob00; END;
```

```
      *  
ERROR at line 1:  
ORA-06550: line 1, column 7:  
PLS-00306: wrong number or types of arguments in call to 'GETENAMESALJOB00'  
ORA-06550: line 1, column 7:  
PL/SQL: Statement ignored
```

```
SQL> EXEC GetEnameSalJob00(7839);  
The Details of Employee 7839 Are...  
The Name of The Employee is : KING  
The Salary of The Employee is : 7000  
The Job of The Employee is : PRESIDENT
```

PL/SQL procedure successfully completed.

```
SQL> <<GetEnameSalJob00>>  
 2  DECLARE  
 3      PEmpno          Emp.Empno%TYPE := &pEmpno;  
 4      V_Ename         Emp.Ename%TYPE;  
 5      V_Sal           Emp.Sal%TYPE;  
 6      V_Job           Emp.Job%TYPE;  
 7  BEGIN  
 8      SELECT  
 9      Ename, Sal, Job  
10 INTO  
11 V_Ename, V_Sal, V_Job  
12 FROM Emp  
13 WHERE Empno = PEmpno;  
14 DBMS_OUTPUT.PUT_LINE('The Details of Employee '||PEmpno||' Are...');  
15 DBMS_OUTPUT.PUT_LINE('The Name of The Employee is : '||V_Ename);  
16 DBMS_OUTPUT.PUT_LINE('The Salary of The Employee is : '||V_Sal);  
17 DBMS_OUTPUT.PUT_LINE('The Job of The Employee is : '||V_Job);  
18 END GetEnameSalJob00;  
19 /
```

```
Enter value for pempno: 7839  
The Details of Employee 7839 Are...  
The Name of The Employee is : KING  
The Salary of The Employee is : 7000  
The Job of The Employee is : PRESIDENT
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PROCEDURE  
 2  GetEnameSalJob01  
 3  (  
 4      PEmpno          Emp.Empno%TYPE  
 5  )  
 6  AS  
 7  V_EmpRec Emp%ROWTYPE;
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
8 BEGIN
9 SELECT
10 *
11 INTO
12 V_EmpRec
13 FROM Emp
14 WHERE Empno = PEmpno;
15 DBMS_OUTPUT.PUT_LINE('The Details of Employee '||PEmpno||' Are...');
16 DBMS_OUTPUT.PUT_LINE('The Name of The Employee is : '||V_EmpRec.Ename);
17 DBMS_OUTPUT.PUT_LINE('The Salary of The Employee is : '||V_EmpRec.Sal);
18 DBMS_OUTPUT.PUT_LINE('The Job of The Employee is : '||V_EmpRec.Job);
19 END GetEnameSalJob01;
20 /
```

Procedure created.

```
SQL> EXEC GetEnameSalJob01(7839);
The Details of Employee 7839 Are...
The Name of The Employee is : KING
The Salary of The Employee is : 7000
The Job of The Employee is : PRESIDENT
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PROCEDURE
2  EmpInsert
3  (
4  P_Empno Emp.Empno%TYPE,
5  P_Ename Emp.Ename%TYPE,
6  P_Sal Emp.Sal%TYPE,
7  P_Deptno Emp.Deptno%TYPE,
8  P_Job Emp.Job%TYPE,
9  P_Comm Emp.Comm%TYPE,
10 P_HireDate Emp.HireDate%TYPE,
11 P_MGR Emp.MGR%TYPE
12 )
13 AS
14 BEGIN
15 INSERT INTO
16 Emp (Empno, Ename, Sal, Deptno, Job, Comm, HireDate, MGR)
17 VALUES(P_Empno, UPPER(P_Ename), P_Sal, P_Deptno, UPPER(P_Job), P_Comm,
P_HireDate, P_MGR);
18 COMMIT;
19 Exception
20 WHEN DUP_VAL_ON_INDEX THEN
21 RAISE_APPLICATION_ERROR(-20001, 'Employee already
22 exists');
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
23 WHEN OTHERS THEN
24 RAISE_APPLICATION_ERROR(-20011, SQLERRM);
25 END;
26 /
```

Procedure created.

```
SQL> DECLARE
  2  TYPE EmpRecord
  3  IS RECORD
  4  (
  5  Empno  Emp.Empno%TYPE,
  6  Ename  Emp.Ename%TYPE,
  7  Sal    Emp.Sal%TYPE,
  8  Deptno      Emp.Deptno%TYPE,
  9  Job        Emp.Job%TYPE,
 10  Comm      Emp.Comm%TYPE,
 11  HireDate   Emp.HireDate%TYPE,
 12  MGR       Emp.MGR%TYPE
 13  );
 14  V_EmpRecord EmpRecord;
 15  BEGIN
 16  V_EmpRecord.Empno := &SEmpno;
 17  V_EmpRecord.Ename := '&SEname';
 18  V_EmpRecord.Sal := &SSal;
 19  V_EmpRecord.Deptno := &SDeptno;
 20  V_EmpRecord.Job := '&SJob';
 21  V_EmpRecord.Comm := &SComm;
 22  V_EmpRecord.HireDate := &SHireDate;
 23  V_EmpRecord.MGR := &SMGR;
 24  EmpInsert(
 25          V_EmpRecord.Empno,
 26          V_EmpRecord.Ename,
 27          V_EmpRecord.Sal,
 28          V_EmpRecord.Deptno,
 29          V_EmpRecord.Job,
 30          V_EmpRecord.Comm,
 31          V_EmpRecord.HireDate,
 32          V_EmpRecord.MGR
 33  );
 34  END;
 35  /
```

```
Enter value for sempno: 1234
Enter value for sename: SAMPLE
Enter value for ssal: 2500
Enter value for sdeptno: 30
Enter value for sjob: CLERK
Enter value for scomm: NULL
Enter value for shiredate: 31-AUG-10
Enter value for smgr: 7566
V_EmpRecord.HireDate := 31-AUG-10;
```

*

```
ERROR at line 22:
ORA-06550: line 22, column 28:
PLS-00201: identifier 'AUG' must be declared
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

ORA-06550: line 22, column 1:
PL/SQL: Statement ignored

```
SQL> /
Enter value for sempno: 1234
Enter value for sename: SAMPLE
Enter value for ssal: 2500
Enter value for sdeptno: 30
Enter value for sjob: CLERK
Enter value for scomm: NULL
Enter value for shiredate: '31-AUG-10'
Enter value for smgr: 7566
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PROCEDURE EmpDataDisplay
 2 AS
 3 V_RowCount NUMBER(4);
 4 TYPE GenericCursor
 5 IS REF CURSOR;
 6 V_GenericCursor GenericCursor;
 7 TYPE TablesRecordType IS RECORD
 8 (
 9     EmpRecord Emp%ROWTYPE
10 );
11 V_EmpRecordType TablesRecordType;
12 BEGIN
13 OPEN V_GenericCursor
14 FOR
15 SELECT * FROM Emp;
16 DBMS_OUTPUT.PUT_LINE(RPAD(LPAD('Employees Information', 49, '*'), 80,
17 '*'));
18 DBMS_OUTPUT.PUT_LINE(RPAD('-', 80, '-'));
19 DBMS_OUTPUT.PUT_LINE(RPAD('EmpNo', 8) || RPAD('Ename', 12) || RPAD('Job',
20 12) || RPAD('Deptno',
21 19
22 8) || RPAD('Mgr', 10) || RPAD('Hiredate', 12) || RPAD('Sal', 12) || RPAD('Comm',
23 10));
24 DBMS_OUTPUT.PUT_LINE(RPAD('-', 80, '-'));
25 LOOP
26     FETCH V_GenericCursor INTO V_EmpRecordType.EmpRecord;
27     EXIT WHEN V_GenericCursor%NOTFOUND;
28     V_RowCount := V_GenericCursor%ROWCOUNT;
29     DBMS_OUTPUT.PUT_LINE(RPAD(V_EmpRecordType.EmpRecord.Empno, 8)
```

Spool Generated For Class of Oracle By Satish K Yellanki

```

28  ||RPAD(V_EmpRecordType.EmpRecord.Ename,
12)| |RPAD(V_EmpRecordType.EmpRecord.Job,
29
30  12)| |RPAD(V_EmpRecordType.EmpRecord.Deptno,
31
32  8)| |NVL(TO_CHAR(RPAD(V_EmpRecordType.EmpRecord.MGR, 10)), 'No
33
34  Manager')| |RPAD(V_EmpRecordType.EmpRecord.Hiredate,
35
36  12)| |RPAD(V_EmpRecordType.EmpRecord.Sal,
37
38  12)| |NVL(TO_CHAR(RPAD(V_EmpRecordType.EmpRecord.Comm, 12)), '-NA-');
39  END LOOP;
40  CLOSE V_GenericCursor;
41  DBMS_OUTPUT.PUT_LINE(V_RowCount|| ' Rows Processed So Far...');
42  END EmpDataDisplay;
43  /

```

Procedure created.

```
SQL> EXEC EmpDataDisplay;
```

*****Employees Information*****

```

-----
EmpNo   Ename      Job          Deptno  Mgr      Hiredate    Sal      Comm
-----
7839    KING       PRESIDENT    10      No
Manager17-NOV-81  7000      -NA-
7698    BLAKE      MANAGER      30      7839     01-MAY-81   3847.5   -NA-
7782    CLARK      MANAGER      10      7839     09-JUN-81   3307.5   -NA-
7566    JONES      MANAGER      20      7839     02-APR-81   4016.25  -NA-
7654    MARTIN     SALESMAN     30      7698     28-SEP-81   1250     1400
7499    ALLEN      SALESMAN     30      7698     20-FEB-81   1600     300
7844    TURNER     SALESMAN     30      7698     08-SEP-81   1500     0
7900    JAMES      CLERK        30      7698     03-DEC-81   950      -NA-
7521    WARD       SALESMAN     30      7698     22-FEB-81   1250     500
7902    FORD       ANALYST      20      7566     03-DEC-81   3000     -NA-
7369    SMITH      CLERK        20      7902     17-DEC-80   800      -NA-
7788    SCOTT      ANALYST      20      7566     09-DEC-82   3000     -NA-
7876    ADAMS      CLERK        20      7788     12-JAN-83   1100     -NA-
7934    MILLER     CLERK        10      7782     23-JAN-82   1300     -NA-
1234    SAMPLE     CLERK        30      7566     31-AUG-10   2500     -NA-

```

15 Rows Processed So Far...

PL/SQL procedure successfully completed.

```
SQL> CREATE OR REPLACE PROCEDURE DeptDataDisplay
```

```

2  AS
3  V_RowCount NUMBER(4);
4  TYPE GenericCursor
5  IS REF CURSOR;
6  V_GenericCursor GenericCursor;
7  TYPE TablesRecordType IS RECORD
8  (
9  DeptRecord Dept%ROWTYPE

```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
10 );
11 V_DeptRecordType TablesRecordType;
12 BEGIN
13 OPEN V_GenericCursor
14 FOR
15 SELECT * FROM Dept;
16 DBMS_OUTPUT.PUT_LINE(RPAD(LPAD('Department Information', 29, '*'), 49,
'*'));
17 DBMS_OUTPUT.PUT_LINE(RPAD('-', 40, '-'));
18 DBMS_OUTPUT.PUT_LINE(RPAD('Deptno', 8)|| RPAD('Dname', 12)||RPAD('Loc',
12));
19 DBMS_OUTPUT.PUT_LINE(RPAD('-', 40, '-'));
20 LOOP
21 FETCH V_GenericCursor INTO V_DeptRecordType.DeptRecord;
22 EXIT WHEN V_GenericCursor%NOTFOUND;
23 V_RowCount := V_GenericCursor%ROWCOUNT;
24 DBMS_OUTPUT.PUT_LINE(RPAD(V_DeptRecordType.DeptRecord.Deptno, 8)
||RPAD(V_DeptRecordType.DeptRecord.Dname,
12)||RPAD(V_DeptRecordType.DeptRecord.Dname, 12));
25 END LOOP;
26 CLOSE V_GenericCursor;
27 DBMS_OUTPUT.PUT_LINE(V_RowCount||' Rows Processed So Far...');
28 END DeptDataDisplay;
29 /
```

Procedure created.

```
SQL> EXEC DeptDataDisplay
*****Department Information*****
-----
Deptno  Dname      Loc
-----
10      ACCOUNTING ACCOUNTING
20      RESEARCH   RESEARCH
30      SALES      SALES
40      OPERATIONS OPERATIONS
4 Rows Processed So Far...
```

PL/SQL procedure successfully completed.

```
SQL> CREATE OR REPLACE PROCEDURE SalGradeDataDisplay
2 AS
3 V_RowCount NUMBER(4);
4 TYPE GenericCursor
5 IS REF CURSOR;
6 V_GenericCursor GenericCursor;
7 TYPE TablesRecordType IS RECORD
8 (
9 SalGradeRecord SalGrade%ROWTYPE
10 );
11 V_SalGradeRecordType TablesRecordType;
12 BEGIN
13 OPEN V_GenericCursor
14 FOR
15 SELECT * FROM SalGrade;
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
16      DBMS_OUTPUT.PUT_LINE(RPAD(LPAD('Salary Grade Information', 29, '*'),
49, '*'));
17      DBMS_OUTPUT.PUT_LINE(RPAD('-', 40, '-'));
18      DBMS_OUTPUT.PUT_LINE(RPAD('Grade', 8) || RPAD('LoSal',
12) || RPAD('HiSal', 12));
19      DBMS_OUTPUT.PUT_LINE(RPAD('-', 40, '-'));
20      LOOP
21          FETCH V_GenericCursor INTO V_SalGradeRecordType.SalGradeRecord;
22          EXIT WHEN V_GenericCursor%NOTFOUND;
23          V_RowCount := V_GenericCursor%ROWCOUNT;
24          DBMS_OUTPUT.PUT_LINE(RPAD(V_SalGradeRecordType.SalGradeRecord.Grade,
8) || RPAD(V_SalGradeRecordType.SalGradeRecord.LoSal,
12) || RPAD(V_SalGradeRecordType.SalGradeRecord.HiSal, 12));
25      END LOOP;
26      CLOSE V_GenericCursor;
27      DBMS_OUTPUT.PUT_LINE(V_RowCount || ' Rows Processed So Far...');
28  END SalGradeDataDisplay;
29  /
```

Procedure created.

```
SQL> EXEC SalGradeDataDisplay;
*****Salary Grade Information*****
-----
Grade   LoSal      HiSal
-----
1       700        1200
2       1201       1400
3       1401       2000
4       2001       3000
5       3001       9999
5 Rows Processed So Far...
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> DECLARE
2  V_Table VARCHAR2(30) := UPPER('&GiveTableName');
3  E_IllegalRequest EXCEPTION;
4  BEGIN
5  IF V_Table = 'EMP' THEN
6  EmpDataDisplay;
7  ELSE
8  IF V_Table = 'DEPT' THEN
9  DeptDataDisplay;
10 ELSE
11     IF V_Table = 'SALGRADE' THEN
12     SalGradedataDisplay;
13     ELSE
14     RAISE E_IllegalRequest;
15     END IF;
16 END IF;
17 END IF;
18 EXCEPTION
```

Spool Generated For Class of Oracle By Satish K Yellanki

```

19 WHEN E_IllegalRequest THEN
20 DBMS_OUTPUT.PUT_LINE('Sorry You Either Do Not Have Permissions on The Table
OR The Requested Table Does Not Exist.');
```

21 END;
22 /
Enter value for givetablename: Emp

*****Employees Information*****

```

-----
EmpNo      Ename      Job          Deptno  Mgr      Hiredate    Sal      Comm
-----
7839      KING      PRESIDENT    10      No
Manager17-NOV-81  7000      -NA-
7698      BLAKE     MANAGER      30      7839     01-MAY-81   3847.5   -NA-
7782      CLARK     MANAGER      10      7839     09-JUN-81   3307.5   -NA-
7566      JONES     MANAGER      20      7839     02-APR-81   4016.25  -NA-
7654      MARTIN    SALESMAN     30      7698     28-SEP-81   1250     1400
7499      ALLEN     SALESMAN     30      7698     20-FEB-81   1600     300
7844      TURNER    SALESMAN     30      7698     08-SEP-81   1500     0
7900      JAMES     CLERK        30      7698     03-DEC-81   950      -NA-
7521      WARD      SALESMAN     30      7698     22-FEB-81   1250     500
7902      FORD      ANALYST      20      7566     03-DEC-81   3000     -NA-
7369      SMITH     CLERK        20      7902     17-DEC-80   800      -NA-
7788      SCOTT     ANALYST      20      7566     09-DEC-82   3000     -NA-
7876      ADAMS     CLERK        20      7788     12-JAN-83   1100     -NA-
7934      MILLER    CLERK        10      7782     23-JAN-82   1300     -NA-
1234      SAMPLE    CLERK        30      7566     31-AUG-10   2500     -NA-
-----

```

15 Rows Processed So Far...

PL/SQL procedure successfully completed.

SQL> /

Enter value for givetablename: Dept

*****Department Information*****

```

-----
Deptno  Dname      Loc
-----
10      ACCOUNTING ACCOUNTING
20      RESEARCH   RESEARCH
30      SALES      SALES
40      OPERATIONS OPERATIONS
-----

```

4 Rows Processed So Far...

PL/SQL procedure successfully completed.

SQL> /

Enter value for givetablename: SalGrade

*****Salary Grade Information*****

```

-----
Grade  LoSal      HiSal
-----
1      700        1200
2      1201       1400
3      1401       2000
4      2001       3000
-----

```

Spool Generated For Class of Oracle By Satish K Yellanki

```
5          3001          9999
5 Rows Processed So Far...
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> ED
```

Wrote file afiedt.buf

```
1 CREATE OR REPLACE PROCEDURE
2 OddNumber(Num1 NUMBER, Num2 NUMBER)
3 IS
4 MyNum NUMBER(4);
5 BEGIN
6 MyNum := Num1;
7 WHILE MyNum < Num2 LOOP
8 IF MOD(MyNum,2) != 0 THEN
9 DBMS_OUTPUT.PUT_LINE('The Odd Number : ' || MyNum);
10 End IF;
11 MyNum := MyNum +1;
12 END LOOP;
13* END;
SQL> /
```

Procedure created.

```
SQL> EXEC OddNumber(10, 25)
```

```
The Odd Number : 11
The Odd Number : 13
The Odd Number : 15
The Odd Number : 17
The Odd Number : 19
The Odd Number : 21
The Odd Number : 23
```

PL/SQL procedure successfully completed.

```
SQL> EXEC OddNumber(25, 10)
```

PL/SQL procedure successfully completed.

```
SQL> EXEC OddNumber(Num2 => 25, Num1 => 10)
```

```
The Odd Number : 11
The Odd Number : 13
The Odd Number : 15
The Odd Number : 17
The Odd Number : 19
The Odd Number : 21
The Odd Number : 23
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> CREATE OR REPLACE PROCEDURE
  2 OddNumber(Num1 NUMBER, Num2 NUMBER DEFAULT 30)
  3 IS
  4 MyNum NUMBER(4);
  5 BEGIN
  6 MyNum := Num1;
  7 WHILE MyNum < Num2 LOOP
  8 IF MOD(MyNum,2) != 0 THEN
  9 DBMS_OUTPUT.PUT_LINE('The Odd Number : '||MyNum);
 10 End IF;
 11 MyNum := MyNum +1;
 12 END LOOP;
 13 END;
 14 /
```

Procedure created.

```
SQL> EXEC OddNumber(20);
The Odd Number : 21
The Odd Number : 23
The Odd Number : 25
The Odd Number : 27
The Odd Number : 29
```

PL/SQL procedure successfully completed.

```
SQL> EXEC OddNumber(20, 40);
The Odd Number : 21
The Odd Number : 23
The Odd Number : 25
The Odd Number : 27
The Odd Number : 29
The Odd Number : 31
The Odd Number : 33
The Odd Number : 35
The Odd Number : 37
The Odd Number : 39
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PROCEDURE
  2 FindEmp
  3 (I_Empno IN NUMBER,
  4 O_Ename OUT VARCHAR2,
  5 O_Job OUT VARCHAR2)
  6 AS
  7 BEGIN
  8 SELECT Ename, Job INTO O_Ename, O_Job
  9 FROM Emp WHERE Empno = I_Empno;
 10 EXCEPTION
 11 WHEN NO_DATA_FOUND THEN
 12 DBMS_OUTPUT.PUT_LINE('Error in Finding the Details of Employee Number : '||
I_Empno);
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
13 END FindEmp;
14 /
```

Procedure created.

```
SQL> DECLARE
  2 V_Ename Emp.Ename%TYPE;
  3 V_Job Emp.Job%TYPE;
  4 BEGIN
  5 FindEmp(7839, V_Ename, V_Job);
  6 DBMS_OUTPUT.PUT_LINE('Employee 7839 is : '|| V_Ename||', '||V_Job||'.');
  7 END;
  8 /
```

Employee 7839 is : KING, PRESIDENT.

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 V_Job Emp.Job%TYPE;
  5 BEGIN
  6 FindEmp(V_Empno, V_Ename, V_Job);
  7 DBMS_OUTPUT.PUT_LINE('Employee '||V_Empno||' is : '|| V_Ename||',
' ||V_Job||'.');
  8* END;
```

```
SQL> /
```

Enter value for gempno: 7654
Employee 7654 is : MARTIN, SALESMAN.

PL/SQL procedure successfully completed.

```
SQL> /
```

Enter value for gempno: 7566
Employee 7566 is : JONES, MANAGER.

PL/SQL procedure successfully completed.

```
SQL> /
```

Enter value for gempno: 7788
Employee 7788 is : SCOTT, ANALYST.

PL/SQL procedure successfully completed.

```
SQL> /
```

Enter value for gempno: 1234
Employee 1234 is : SAMPLE, CLERK.

PL/SQL procedure successfully completed.

```
SQL> /
```

Enter value for gempno: 2234

Spool Generated For Class of Oracle By Satish K Yellanki

Error in Finding the Details of Employee Number : 2234
Employee 2234 is : , .

PL/SQL procedure successfully completed.

SQL> cl scr

```
SQL> CREATE OR REPLACE PROCEDURE EmpInfo(I_Deptno IN NUMBER)
  2 AS
  3 CURSOR EmpInfoCursor IS
  4 SELECT Ename, Job, Sal, Comm
  5 FROM Emp
  6 WHERE Deptno = I_Deptno;
  7 EmpRecord EmpInfoCursor%ROWTYPE;
  8 NEmployees NUMBER := 0;
  9 TSalary NUMBER := 0;
 10 AVGSalary NUMBER(7,2) := 0;
 11 MAXSalary NUMBER(7,2) := 0;
 12 BEGIN
 13 OPEN EmpInfoCursor;
 14 LOOP
 15 FETCH EmpInfoCursor INTO EmpRecord;
 16 EXIT WHEN EmpInfoCursor%NOTFOUND;
 17 DBMS_OUTPUT.PUT_LINE('Employee Name : '||EmpRecord.Ename);
 18 DBMS_OUTPUT.PUT_LINE('Employee Job : '||EmpRecord.Job);
 19 DBMS_OUTPUT.PUT_LINE('Employee Salary : '||EmpRecord.Sal);
 20 DBMS_OUTPUT.PUT_LINE('Employee Commission : '||EmpRecord.Comm);
 21 DBMS_OUTPUT.PUT_LINE('*****');
 22 TSalary := TSalary + EmpRecord.Sal;
 23 NEmployees := NEmployees + 1;
 24 IF EmpRecord.Sal > MAXSalary THEN
 25 MAXSalary := EmpRecord.Sal;
 26 END IF;
 27 END LOOP;
 28 AVGSalary := TSalary / NEmployees;
 29 DBMS_OUTPUT.PUT_LINE('Number of Employees : '||NEmployees);
 30 DBMS_OUTPUT.PUT_LINE('Total Salary : '||TSalary);
 31 DBMS_OUTPUT.PUT_LINE('Maximum Salary : '||MAXSalary);
 32 DBMS_OUTPUT.PUT_LINE('Average Salary : '||AVGSalary);
 33 CLOSE EmpInfoCursor;
 34 END EmpInfo;
 35 /
```

Procedure created.

```
SQL> EXEC EmpInfo(10)
Employee Name : KING
Employee Job : PRESIDENT
Employee Salary : 7000
Employee Commission :
*****
Employee Name : CLARK
Employee Job : MANAGER
Employee Salary : 3307.5
Employee Commission :
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
*****  
Employee Name : MILLER  
Employee Job : CLERK  
Employee Salary : 1300  
Employee Commission :  
*****  
Number of Employees : 3  
Total Salary : 11607.5  
Maximum Salary : 7000  
Average Salary : 3869.17
```

PL/SQL procedure successfully completed.

SQL> /

Procedure created.

```
SQL> EXEC EmpInfo(20)  
Employee Name : JONES  
Employee Job : MANAGER  
Employee Salary : 4016.25  
Employee Commission :  
*****  
Employee Name : FORD  
Employee Job : ANALYST  
Employee Salary : 3000  
Employee Commission :  
*****  
Employee Name : SMITH  
Employee Job : CLERK  
Employee Salary : 800  
Employee Commission :  
*****  
Employee Name : SCOTT  
Employee Job : ANALYST  
Employee Salary : 3000  
Employee Commission :  
*****  
Employee Name : ADAMS  
Employee Job : CLERK  
Employee Salary : 1100  
Employee Commission :  
*****  
Number of Employees : 5  
Total Salary : 11916.25  
Maximum Salary : 4016.25  
Average Salary : 2383.25
```

PL/SQL procedure successfully completed.

SQL> cl scr

```
SQL> <<EmpInfo>>  
2 DECLARE  
3 CURSOR EmpInfoCursor IS
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
4 SELECT Ename, Job, Sal, Comm
5 FROM Emp
6 WHERE Deptno = &I_Deptno;
7 EmpRecord EmpInfoCursor%ROWTYPE;
8 NEmployees NUMBER := 0;
9 TSalary NUMBER := 0;
10 AVGSalary NUMBER(7,2) := 0;
11 MAXSalary NUMBER(7,2) := 0;
12 BEGIN
13 OPEN EmpInfoCursor;
14 LOOP
15 FETCH EmpInfoCursor INTO EmpRecord;
16 EXIT WHEN EmpInfoCursor%NOTFOUND;
17 DBMS_OUTPUT.PUT_LINE('Employee Name : '||EmpRecord.Ename);
18 DBMS_OUTPUT.PUT_LINE('Employee Job : '||EmpRecord.Job);
19 DBMS_OUTPUT.PUT_LINE('Employee Salary : '||EmpRecord.Sal);
20 DBMS_OUTPUT.PUT_LINE('Employee Comission : '||EmpRecord.Comm);
21 DBMS_OUTPUT.PUT_LINE('*****');
22 TSalary := TSalary + EmpRecord.Sal;
23 NEmployees := NEmployees + 1;
24 IF EmpRecord.Sal > MAXSalary THEN
25 MAXSalary := EmpRecord.Sal;
26 END IF;
27 END LOOP;
28 AVGSalary := TSalary / NEmployees;
29 DBMS_OUTPUT.PUT_LINE('Number of Employees : '||NEmployees);
30 DBMS_OUTPUT.PUT_LINE('Total Salary : '||TSalary);
31 DBMS_OUTPUT.PUT_LINE('Maximum Salary : '||MAXSalary);
32 DBMS_OUTPUT.PUT_LINE('Average Salary : '||AVGSalary);
33 CLOSE EmpInfoCursor;
34 END EmpInfo;
35 /
```

Enter value for i_deptno: 10

```
Employee Name : KING
Employee Job : PRESIDENT
Employee Salary : 7000
Employee Comission :
*****
Employee Name : CLARK
Employee Job : MANAGER
Employee Salary : 3307.5
Employee Comission :
*****
Employee Name : MILLER
Employee Job : CLERK
Employee Salary : 1300
Employee Comission :
*****
Number of Employees : 3
Total Salary : 11607.5
Maximum Salary : 7000
Average Salary : 3869.17
```

PL/SQL procedure successfully completed.

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> /
Enter value for i_deptno: 20
Employee Name : JONES
Employee Job : MANAGER
Employee Salary : 4016.25
Employee Commission :
*****
Employee Name : FORD
Employee Job : ANALYST
Employee Salary : 3000
Employee Commission :
*****
Employee Name : SMITH
Employee Job : CLERK
Employee Salary : 800
Employee Commission :
*****
Employee Name : SCOTT
Employee Job : ANALYST
Employee Salary : 3000
Employee Commission :
*****
Employee Name : ADAMS
Employee Job : CLERK
Employee Salary : 1100
Employee Commission :
*****
Number of Employees : 5
Total Salary : 11916.25
Maximum Salary : 4016.25
Average Salary : 2383.25
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PROCEDURE
 2  EmployeeInsert(
 3      PEmpno                IN Emp.Empno%TYPE,
 4      PName                 IN Emp.Ename%TYPE,
 5      PJob                  IN Emp.Job%TYPE,
 6      PSal                  IN Emp.Sal%TYPE,
 7      PComm                 IN Emp.Comm%TYPE,
 8      PDeptno              IN Emp.Deptno%TYPE,
 9      PMGR                  IN Emp.MGR%TYPE,
10     PHireDate              IN Emp.HireDate%TYPE
11     )
12 IS
13 TYPE ValidateInsertValues
14 IS
15 RECORD
16 (
17     SalaryLess          NUMBER(4),
18     SalaryMore          NUMBER(6)
19 );
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
20 ValidateRecordValues      ValidateInsertValues;
21 TYPE CheckBits
22 IS
23 RECORD
24 (
25   CheckJobBit              BOOLEAN,
26   CheckDeptnoBit          BOOLEAN,
27   CheckMGRBit             BOOLEAN,
28   CheckHireDateBit        BOOLEAN,
29   CheckSalBit             BOOLEAN
30 );
31 CheckBitsInstance CheckBits;
32 E_ForiegnKey EXCEPTION;
33 PRAGMA EXCEPTION_INIT(E_ForiegnKey, -2292);
34 E_NotNULLViolation EXCEPTION;
35 PRAGMA EXCEPTION_INIT(E_NotNULLViolation, -1400);
36 E_CheckViolation          EXCEPTION;
37 PRAGMA EXCEPTION_INIT(E_CheckViolation, -2290);
38 BEGIN
39   CheckJob(PJob, CheckBitsInstance.CheckJobBit);
40   CheckDeptno(PDeptno, CheckBitsInstance.CheckDeptnoBit);
41   CheckMGRNO(PMGR, CheckBitsInstance.CheckMGRBit);
42   CheckHireDate(PHireDate, CheckBitsInstance.CheckHireDateBit);
43   CheckSalary(PSal, CheckBitsInstance.CheckSalBit);
44   IF CheckBitsInstance.CheckJobBit THEN
45     IF CheckBitsInstance.CheckDeptnoBit THEN
46       IF CheckBitsInstance.CheckMGRBit THEN
47         IF CheckBitsInstance.CheckHireDateBit THEN
48           IF CheckBitsInstance.CheckSalBit THEN
49             INSERT INTO Emp(Empno, Ename, Job, Sal, Comm,
Deptno, MGR, HireDate)
50             VALUES(PEmpno, PEname, PJob, PSal, PComm, PDeptno,
PMGR, PHireDate);
51             COMMIT;
52             DBMS_OUTPUT.PUT_LINE('Your Record is Inserted
Sucessfully...');
53           ELSE
54             DBMS_OUTPUT.PUT_LINE('Insert Failed Due To Problem
in Salary Values Out of Range');
55           END IF;
56         ELSE
57           DBMS_OUTPUT.PUT_LINE('Insert Failed Due To Problem in
Hire Date Not Matching With Server Standards.');
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
67 END IF;
68 EXCEPTION
69 WHEN DUP_VAL_ON_INDEX THEN
70 DBMS_OUTPUT.PUT_LINE('Sorry, The Field With Unique Value is Getting
Duplicated. Hence Insert is Cancelled.');
```

```
71 ROLLBACK;
72 WHEN E_CheckViolation THEN
73 DBMS_OUTPUT.PUT_LINE('A Field with Check Constraint is not Attended
Properly, Please Check Properly. Hence Insert is Cancelled.');
```

```
74 ROLLBACK;
75 WHEN E_NotNULLViolation THEN
76 DBMS_OUTPUT.PUT_LINE('A Field which Cannot be NULL, is not attended, Please
Check Properly. Hence Insert is Cancelled.');
```

```
77 ROLLBACK;
78 WHEN E_ForeignKey THEN
79 DBMS_OUTPUT.PUT_LINE('A Field is Violating The Parent Child Relation. Hence
Insert is Cancelled.');
```

```
80 ROLLBACK;
81 END EmployeeInsert;
82 /
```

Warning: Procedure created with compilation errors.

SQL> SHOW ERRORS

Errors for PROCEDURE EMPLOYEEINSERT:

LINE/COL ERROR

```
-----
39/1 PL/SQL: Statement ignored
39/1 PLS-00201: identifier 'CHECKJOB' must be declared
40/1 PL/SQL: Statement ignored
40/1 PLS-00201: identifier 'CHECKDEPTNO' must be declared
41/1 PL/SQL: Statement ignored
41/1 PLS-00201: identifier 'CHECKMGRNO' must be declared
42/1 PL/SQL: Statement ignored
42/1 PLS-00201: identifier 'CHECKHIREDATE' must be declared
43/1 PL/SQL: Statement ignored
43/1 PLS-00201: identifier 'CHECKSALARY' must be declared
```

SQL> CREATE OR REPLACE PROCEDURE

```
2 CheckJob(PCheckJob IN Emp.Job%TYPE, PChkBit OUT BOOLEAN)
3 IS
4 V_JobExists          NUMBER(2);
5 V_Job                Emp.Job%TYPE;
6 E_NoJob              EXCEPTION;
7 BEGIN
8 SELECT
9   COUNT(*)
10 INTO
11   V_JobExists
12 FROM Emp
13 WHERE Job = PCheckJob;
14 IF V_JobExists = 0
15 THEN
16   RAISE E_NoJob;
17 ELSE
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
18 SELECT
19   DISTINCT Job
20 INTO
21   V_Job
22 FROM Emp
23 WHERE Job = PCheckJob;
24 DBMS_OUTPUT.PUT_LINE('Success! ' || V_Job || ' is Found...');
25 PChkBit := TRUE;
26 END IF;
27 EXCEPTION
28 WHEN E_NoJob THEN
29   DBMS_OUTPUT.PUT_LINE('Sorry, The Job ' || PCheckJob || ' To be Inserted is Not
Acceptable.');
```

Procedure created.

```
SQL> ALTER PROCEDURE EMPLOYEEINSERT COMPILE;
ALTER PROCEDURE EMPLOYEEINSERT COMPILE
```

*

```
ERROR at line 1:
ORA-00922: missing or invalid option
```

```
SQL> ALTER PROCEDURE EMPLOYEEINSERT COMPILE;
```

Warning: Procedure altered with compilation errors.

```
SQL> SHOW ERRORS
Errors for PROCEDURE EMPLOYEEINSERT:
```

LINE/COL ERROR

```
-----
40/1      PL/SQL: Statement ignored
40/1      PLS-00201: identifier 'CHECKDEPTNO' must be declared
41/1      PL/SQL: Statement ignored
41/1      PLS-00201: identifier 'CHECKMGRNO' must be declared
42/1      PL/SQL: Statement ignored
42/1      PLS-00201: identifier 'CHECKHIREDATE' must be declared
43/1      PL/SQL: Statement ignored
43/1      PLS-00201: identifier 'CHECKSALARY' must be declared
SQL> CREATE OR REPLACE PROCEDURE
  2  CheckDeptno(PCheckDeptno IN Emp.Deptno%TYPE, PChkBit OUT BOOLEAN)
  3  IS
  4  V_DeptnoExists          NUMBER(2);
  5  V_Deptno                Emp.Job%TYPE;
  6  E_NoDept               EXCEPTION;
  7  BEGIN
  8  SELECT
  9    COUNT(*)
10 INTO
11   V_DeptnoExists
12 FROM Emp
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
13 WHERE Deptno = PCheckDeptno;
14 IF V_DeptnoExists = 0
15 THEN
16   RAISE E_NoDept;
17 ELSE
18   SELECT
19     DISTINCT Deptno
20   INTO
21     V_Deptno
22   FROM Dept
23   WHERE Deptno = PCheckDeptno;
24   DBMS_OUTPUT.PUT_LINE('Success! ' || V_Deptno || ' is Found...');
25   PChkBit := TRUE;
26   END IF;
27   EXCEPTION
28   WHEN E_NoDept THEN
29     DBMS_OUTPUT.PUT_LINE('Sorry, The Department Number ' || PCheckDeptno || ' To be
Inserted is Not Acceptable.');
```

Procedure created.

```
SQL> ALTER PROCEDURE EMPLOYEEINSERT COMPILE;
```

Warning: Procedure altered with compilation errors.

```
SQL> SHOW ERRORS
```

Errors for PROCEDURE EMPLOYEEINSERT:

LINE/COL ERROR

```
-----
41/1      PL/SQL: Statement ignored
41/1      PLS-00201: identifier 'CHECKMGRNO' must be declared
42/1      PL/SQL: Statement ignored
42/1      PLS-00201: identifier 'CHECKHIREDATE' must be declared
43/1      PL/SQL: Statement ignored
43/1      PLS-00201: identifier 'CHECKSALARY' must be declared
SQL> CREATE OR REPLACE PROCEDURE
  2 CheckMGRNO(PCheckMGRNO IN Emp.Deptno%TYPE, PChkBit OUT BOOLEAN)
  3 IS
  4   V_MGRNOExists          NUMBER(2);
  5   V_MgrNo                Emp.MGR%TYPE;
  6   E_InvalidManager      EXCEPTION;
  7 BEGIN
  8   SELECT
  9     COUNT(*)
10  INTO
11    V_MGRNOExists
12  FROM Emp
13  WHERE MGR = PCheckMGRNO;
14  IF V_MGRNOExists = 0
15  THEN
16    RAISE E_InvalidManager;
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
17 ELSE
18 SELECT
19   DISTINCT MGR
20 INTO
21   V_MgrNo
22 FROM Emp
23 WHERE MGR = PCheckMGRNO;
24 DBMS_OUTPUT.PUT_LINE('Success! ' || V_MgrNo || ' is Found...');
25 PChkBit := TRUE;
26 END IF;
27 EXCEPTION
28 WHEN E_InvalidManager THEN
29   DBMS_OUTPUT.PUT_LINE('Sorry, The Manager Number ' || PCheckMGRNO || ' To be
Inserted is Not Acceptable.');
```

Procedure created.

```
SQL> ALTER PROCEDURE EMPLOYEEINSERT COMPILE;
```

Warning: Procedure altered with compilation errors.

```
SQL> SHOW ERRORS
```

Errors for PROCEDURE EMPLOYEEINSERT:

LINE/COL ERROR

```
-----
42/1      PL/SQL: Statement ignored
42/1      PLS-00201: identifier 'CHECKHIREDATE' must be declared
43/1      PL/SQL: Statement ignored
43/1      PLS-00201: identifier 'CHECKSALARY' must be declared
SQL> CREATE OR REPLACE PROCEDURE
  2 CheckHireDate(PCheckHireDate IN Emp.HireDate%TYPE, PChkBit OUT BOOLEAN)
  3 IS
  4   V_CurrentDate          Emp.HireDate%TYPE;
  5   E_InvalidHireDate      EXCEPTION;
  6 BEGIN
  7   SELECT
  8   SYSDATE
  9   INTO
10   V_CurrentDate
11   FROM DUAL;
12   V_CurrentDate := TO_DATE(TO_CHAR(V_CurrentDate, 'DD-MON-YY'), 'DD-MON-YY');
13   IF V_CurrentDate <> PCheckHireDate
14   THEN
15     RAISE E_InvalidHireDate;
16   ELSE
17     DBMS_OUTPUT.PUT_LINE('Success! ' || V_CurrentDate || ' Matches The Server
Clock.');
```

```
18   PChkBit := TRUE;
```

```
19   END IF;
```

```
20   EXCEPTION
```

```
21   WHEN E_InvalidHireDate THEN
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
22 DBMS_OUTPUT.PUT_LINE('Sorry, The Hire Date '||PCheckHireDate||' To be
Inserted is Not Acceptable.');
```

```
23 ROLLBACK;
24 END CheckHireDate;
25 /
```

Procedure created.

```
SQL> ALTER PROCEDURE EMPLOYEEINSERT COMPILE;
```

Warning: Procedure altered with compilation errors.

```
SQL> SHOW ERRORS
```

Errors for PROCEDURE EMPLOYEEINSERT:

LINE/COL ERROR

```
-----
43/1      PL/SQL: Statement ignored
43/1      PLS-00201: identifier 'CHECKSALARY' must be declared
SQL> CREATE OR REPLACE PROCEDURE
  2 CheckSalary(PCheckSalary IN Emp.Sal%TYPE, PChkBit OUT BOOLEAN)
  3 IS
  4 V_SalaryLess          NUMBER(6);
  5 V_SalaryMore          NUMBER(6);
  6 E_SalLess             EXCEPTION;
  7 E_SalMore             EXCEPTION;
  8 BEGIN
  9 V_SalaryLess := 500;
 10 V_SalaryMore := 150000;
 11 IF PCheckSalary < V_SalaryLess
 12 THEN
 13 RAISE E_SalLess;
 14 ELSIF PCheckSalary > V_SalaryMore
 15 THEN
 16 RAISE E_SalMore;
 17 END IF;
 18 DBMS_OUTPUT.PUT_LINE('Success! The Salary '||PCheckSalary||' is
Acceptable.');
```

```
19 PChkBit := TRUE;
20 EXCEPTION
21 WHEN E_SalLess THEN
22 DBMS_OUTPUT.PUT_LINE('Sorry, The Salary '||PCheckSalary||' To be Inserted
is Less Than The Acceptable Value.');
```

```
23 ROLLBACK;
24 WHEN E_SalMore THEN
25 DBMS_OUTPUT.PUT_LINE('Sorry, The Salary '||PCheckSalary||' To be Inserted
is More Than The Acceptable Value.');
```

```
26 ROLLBACK;
27 END CheckSalary;
28 /
```

Procedure created.

```
SQL> ALTER PROCEDURE EMPLOYEEINSERT COMPILE;
```

Spool Generated For Class of Oracle By Satish K Yellanki

Procedure altered.

```
SQL> DECLARE
  2  V_Empno          Emp.Empno%TYPE := &Empno;
  3  V_Ename          Emp.Ename%TYPE := '&Ename';
  4  V_Job            Emp.Job%TYPE  := '&Job';
  5  V_Sal            Emp.Sal%TYPE  := &Sal;
  6  V_Comm           Emp.Comm%TYPE := &Commission;
  7  V_Deptno        Emp.Deptno%TYPE := &Deptno;
  8  V_MGR           Emp.MGR%TYPE  := &MGR;
  9  V_HireDate       Emp.HireDate%TYPE := '&HireDate';
 10 BEGIN
 11  EmployeeInsert(V_Empno, V_Ename, V_Job, V_Sal, V_Comm, V_Deptno, V_MGR,
V_HireDate);
 12 END;
 13 /
```

```
Enter value for empno: 1235
Enter value for ename: SAMPLE
Enter value for job: CLERK
Enter value for sal: 2500
Enter value for commission: NULL
Enter value for deptno: 30
Enter value for mgr: 7566
Enter value for hiredate: 31-AUG-10
Success! CLERK is Found...
Success! 30 is Found...
Success! 7566 is Found...
Success! 31-AUG-10 Matches The Server Clock.
Success! The Salary 2500 is Acceptable.
Your Record is Inserted Successfully...
```

PL/SQL procedure successfully completed.

```
SQL> /
Enter value for empno: 1236
Enter value for ename: SAMPLE
Enter value for job: SRCLERK
Enter value for sal: 2000
Enter value for commission: NULL
Enter value for deptno: 80
Enter value for mgr: 7654
Enter value for hiredate: 28-AUG-10
Sorry, The Job SRCLERK To be Inserted is Not Acceptable.
Sorry, The Department Number 80 To be Inserted is Not Acceptable.
Sorry, The Manager Number 7654 To be Inserted is Not Acceptable.
Sorry, The Hire Date 28-AUG-10 To be Inserted is Not Acceptable.
Success! The Salary 2000 is Acceptable.
Insert Failed Due To Problem in Designation Not Suitable To The Organization.
```

PL/SQL procedure successfully completed.

```
SQL> SPOOL OFF
SQL> cl scr
```

```
SQL> CREATE OR REPLACE FUNCTION
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
2 Factorial(Num NUMBER)
3 RETURN NUMBER
4 IS
5 Fact NUMBER(4) := 1;
6 BEGIN
7 FOR MyIndex IN REVERSE 1..Num
8 LOOP
9 Fact := Fact * MyIndex;
10 END LOOP;
11 RETURN Fact;
12 END;
13 /
```

Function created.

```
SQL> EXEC Factorial(5)
BEGIN Factorial(5); END;
```

```
      *
ERROR at line 1:
ORA-06550: line 1, column 7:
PLS-00221: 'FACTORIAL' is not a procedure or is undefined
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
```

```
SQL> cl scr
```

```
SQL> SELECT Factorial(5) Fact FROM DUAL;
```

```
      FACT
-----
      120
```

```
SQL> CREATE TABLE SampFun
2 (
3 SampID NUMBER(2),
4 SampFact NUMBER(4)
5 );
```

Table created.

```
SQL> INSERT INTO SampFun
2 VALUES(1, Factorial(1));
```

1 row created.

```
SQL> INSERT INTO SampFun
2 VALUES(2, Factorial(2));
```

1 row created.

```
SQL> INSERT INTO SampFun
2 VALUES(3, Factorial(3));
```

Spool Generated For Class of Oracle By Satish K Yellanki

1 row created.

```
SQL> INSERT INTO SampFun
  2  VALUES(4, Factorial(4));
```

1 row created.

```
SQL> INSERT INTO SampFun
  2  VALUES(5, Factorial(5));
```

1 row created.

```
SQL> SELECT * FROM SampFun;
```

SAMPID	SAMPFACT
1	1
2	2
3	6
4	24
5	120

```
SQL> UPDATE SampFun
  2  SET
  3  SampFact = Factorial(5)
  4  WHERE SampID = 3;
```

1 row updated.

```
SQL> SELECT * FROM SampFun;
```

SAMPID	SAMPFACT
1	1
2	2
3	120
4	24
5	120

```
SQL> UPDATE SampFun
  2  SET
  3  SampFact = Factorial(5)
  4  WHERE SampFact = Factorial(4);
```

1 row updated.

```
SQL> SELECT * FROM SampFun;
```

SAMPID	SAMPFACT
1	1
2	2
3	120
4	120
5	120

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> DELETE FROM SampFun
  2  WHERE SampID = 3;
```

1 row deleted.

```
SQL> SELECT * FROM SampFun;
```

SAMPID	SAMPFACT
1	1
2	2
4	120
5	120

```
SQL> DELETE FROM SampFun
  2  WHERE SampFact = Factorial(5);
```

2 rows deleted.

```
SQL> SELECT * FROM SampFun;
```

SAMPID	SAMPFACT
1	1
2	2

```
SQL> DROP TABLE SampFun;
```

Table dropped.

```
SQL> PURGE RECYCLEBIN;
```

Recyclebin purged.

```
SQL> cl scr
```

```
SQL> ED
```

Wrote file afiedt.buf

```
  1  CREATE TABLE SampFun
  2  (
  3  SampID NUMBER(2),
  4  SampFact NUMBER(4)
  5  CONSTRAINT SampFun_SampFact_CHK
  6  CHECK(SampFact > Factorial(5))
  7* )
```

```
SQL> /
```

```
  CHECK(SampFact > Factorial(5))
```

*

ERROR at line 6:

ORA-00904: "FACTORIAL": invalid identifier

```
SQL> cl scr
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> DECLARE
  2  V_Factorial NUMBER(4) := 0;
  3  BEGIN
  4  V_Factorial := Factorial(5);
  5  DBMS_OUTPUT.PUT_LINE('The Factorial is : '||V_Factorial);
  6  END;
  7  /
```

The Factorial is : 120

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE FUNCTION
  2  Combination(Num1 NUMBER, Num2 NUMBER)
  3  RETURN NUMBER
  4  IS
  5  Combi NUMBER(4,2) := 1;
  6  BEGIN
  7  Combi := (Factorial(Num1) / (Factorial(Num1 - Num2) * Factorial(Num2)));
  8  RETURN Combi;
  9  END;
 10  /
```

Function created.

```
SQL> SELECT Combination(6, 2) Combi FROM DUAL;
```

```
      COMBI
-----
          15
```

```
SQL> DROP FUNCTION Factorial;
```

Function dropped.

```
SQL> SELECT Combination(6, 2) Combi FROM DUAL;
SELECT Combination(6, 2) Combi FROM DUAL
      *
```

ERROR at line 1:

ORA-06575: Package or function COMBINATION is in an invalid state

```
SQL> CREATE OR REPLACE FUNCTION
  2  Factorial(Num NUMBER)
  3  RETURN NUMBER
  4  IS
  5  Fact NUMBER(4) := 1;
  6  BEGIN
  7  FOR MyIndex IN REVERSE 1..Num
  8  LOOP
  9  Fact := Fact * MyIndex;
 10  END LOOP;
 11  RETURN Fact;
```

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

```
12 END;  
13 /
```

Function created.

```
SQL> SELECT Combination(6, 2) Combi FROM DUAL;
```

```
      COMBI  
-----  
          15
```

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE FUNCTION  
 2  EmpExp(V_Empno NUMBER)  
 3  RETURN NUMBER  
 4  IS  
 5  V_HireDate Emp.HireDate%TYPE;  
 6  V_Exp NUMBER(4,2) := 1;  
 7  BEGIN  
 8  SELECT HireDate INTO V_HireDate  
 9  FROM Emp  
10  WHERE Empno = V_Empno;  
11  V_Exp := MONTHS_BETWEEN(SYSDATE,V_HireDate) / 12;  
12  RETURN V_Exp;  
13  END;  
14  /
```

Function created.

```
SQL> SELECT Ename, HireDate, MONTHS_BETWEEN(SYSDATE, HireDate) / 12 EmpExp  
 2  FROM Emp  
 3  WHERE Empno = 7839;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.7912641

```
SQL> SELECT Ename, HireDate, EmpExp(Empno) EmpExp  
 2  FROM Emp  
 3  WHERE Empno = 7839;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.79

```
SQL> SELECT Ename, HireDate, EmpExp(Empno) EmpExp  
 2  FROM Emp;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.79
BLAKE	01-MAY-81	29.33
CLARK	09-JUN-81	29.23
JONES	02-APR-81	29.41

Spool Generated For Class of Oracle By Satish K Yellanki

```

MARTIN      28-SEP-81      28.93
ALLEN       20-FEB-81      29.53
TURNER      08-SEP-81      28.98
JAMES       03-DEC-81      28.75
WARD        22-FEB-81      29.53
FORD        03-DEC-81      28.75
SMITH       17-DEC-80      29.71
    
```

```

ENAME      HIREDATE      EMPEXP
-----
SCOTT       09-DEC-82      27.73
ADAMS       12-JAN-83      27.64
MILLER      23-JAN-82      28.61
    
```

14 rows selected.

```

SQL> CREATE OR REPLACE FUNCTION
  2  EmpTotSal(V_Empno NUMBER)
  3  RETURN NUMBER
  4  IS
  5  V_Sal Emp.Sal%TYPE;
  6  V_Comm Emp.Comm%TYPE;
  7  V_TotSal NUMBER(9, 2) := 1;
  8  BEGIN
  9  SELECT Sal, Comm INTO V_Sal, V_Comm
 10  FROM Emp
 11  WHERE Empno = V_Empno;
 12  V_TotSal := V_Sal + NVL(V_Comm, 0);
 13  RETURN V_TotSal;
 14  END;
 15  /
    
```

Function created.

```

SQL> SELECT Ename, Sal, Comm, EmpTotSal(Empno) TotSal, HireDate, EmpExp(Empno)
EmpExp
  2  FROM Emp;
    
```

```

ENAME      SAL      COMM      TOTSAL HIREDATE      EMPEXP
-----
KING        5000          5000 17-NOV-81      28.79
BLAKE       2850          2850 01-MAY-81      29.33
CLARK       2450          2450 09-JUN-81      29.23
JONES       2975          2975 02-APR-81      29.41
MARTIN      1250      1400      2650 28-SEP-81      28.93
ALLEN       1600      300      1900 20-FEB-81      29.53
TURNER      1500      0      1500 08-SEP-81      28.98
JAMES       950          950 03-DEC-81      28.75
WARD        1250      500      1750 22-FEB-81      29.53
FORD        3000          3000 03-DEC-81      28.75
SMITH       800          800 17-DEC-80      29.71

ENAME      SAL      COMM      TOTSAL HIREDATE      EMPEXP
-----
SCOTT       3000          3000 09-DEC-82      27.73
    
```

Spool Generated For Class of Oracle By Satish K Yellanki

ADAMS	1100	1100	12-JAN-83	27.64
MILLER	1300	1300	23-JAN-82	28.61

14 rows selected.

SQL> cl scr

```
SQL> CREATE OR REPLACE PROCEDURE
 2 EmpInfo(I_Deptno IN NUMBER)
 3 AS
 4 CURSOR EmpInfoCursor IS
 5 SELECT Empno, Ename, Job, Sal, Comm
 6 FROM Emp
 7 WHERE Deptno = I_Deptno;
 8 EmpRecord EmpInfoCursor%ROWTYPE;
 9 NEmployees NUMBER := 0;
10 TSalary NUMBER := 0;
11 AVGSalary NUMBER(7,2) := 0;
12 MAXSalary NUMBER(7,2) := 0;
13 V_Exp NUMBER(4,2) := 1;
14 BEGIN
15 OPEN EmpInfoCursor;
16 LOOP
17 FETCH EmpInfoCursor INTO EmpRecord;
18 EXIT WHEN EmpInfoCursor%NOTFOUND;
19 DBMS_OUTPUT.PUT_LINE('Employee Name : '||EmpRecord.Ename);
20 DBMS_OUTPUT.PUT_LINE('Employee Job : '||EmpRecord.Job);
21 DBMS_OUTPUT.PUT_LINE('Employee Salary : '||EmpRecord.Sal);
22 DBMS_OUTPUT.PUT_LINE('Employee Comission : '||EmpRecord.Comm);
23 V_Exp := EmpExp(EmpRecord.Empno);
24 DBMS_OUTPUT.PUT_LINE('Employee's Experiance : '||V_Exp);
25 DBMS_OUTPUT.PUT_LINE('*****');
26 TSalary := TSalary + EmpRecord.Sal;
27 NEmployees := NEmployees + 1;
28 IF EmpRecord.Sal > MAXSalary THEN
29 MAXSalary := EmpRecord.Sal;
30 END IF;
31 END LOOP;
32 AVGSalary := TSalary / NEmployees;
33 DBMS_OUTPUT.PUT_LINE('Number of Employees : '||NEmployees);
34 DBMS_OUTPUT.PUT_LINE('Total Salary : '||TSalary);
35 DBMS_OUTPUT.PUT_LINE('Maximum Salary : '||MAXSalary);
36 DBMS_OUTPUT.PUT_LINE('Average Salary : '||AVGSalary);
37 CLOSE EmpInfoCursor;
38 END EmpInfo;
39 /
```

Procedure created.

```
SQL> EXEC EmpInfo(10)
Employee Name : KING
Employee Job : PRESIDENT
Employee Salary : 5000
Employee Comission :
Employee's Experiance : 28.79
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
*****
Employee Name : CLARK
Employee Job : MANAGER
Employee Salary : 2450
Employee Commission :
Employee's Experience : 29.23
*****
Employee Name : MILLER
Employee Job : CLERK
Employee Salary : 1300
Employee Commission :
Employee's Experience : 28.61
*****
Number of Employees : 3
Total Salary : 8750
Maximum Salary : 5000
Average Salary : 2916.67
```

PL/SQL procedure successfully completed.

SQL> cl scr

```
SQL> CREATE OR REPLACE FUNCTION
  2 GetDeptDetails(P_Deptno NUMBER)
  3 RETURN Dept%ROWTYPE
  4 IS
  5 V_DeptRec Dept%ROWTYPE;
  6 V_Deptno Dept.Deptno%TYPE := P_Deptno;
  7 BEGIN
  8 SELECT * INTO V_DeptRec
  9 FROM Dept
 10 WHERE Deptno = V_Deptno;
 11 RETURN V_DeptRec;
 12 END;
 13 /
```

Function created.

```
SQL> SELECT GetDeptDetails(10) Dept10Det FROM DUAL;
SELECT GetDeptDetails(10) Dept10Det FROM DUAL
```

```
*
ERROR at line 1:
ORA-00902: invalid datatype
```

```
SQL> DECLARE
  2 V_DeptRec Dept%ROWTYPE;
  3 BEGIN
  4 V_DeptRec.Deptno := &GDeptno;
  5 V_DeptRec := GetDeptDetails(V_DeptRec.Deptno);
  6 DBMS_OUTPUT.PUT_LINE(V_DeptRec.Deptno || ' ' || V_DeptRec.DName || '
' || V_DeptRec.Loc);
  7 END;
  8 /
```

Enter value for gdeptno: 10

**Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090**

Spool Generated For Class of Oracle By Satish K Yellanki

10 ACCOUNTING NEW YORK

PL/SQL procedure successfully completed.

SQL> cl scr

```
SQL> CREATE OR REPLACE PROCEDURE
  2  EmpInfo(I_Deptno IN NUMBER)
  3  AS
  4  V_DeptRec Dept%ROWTYPE;
  5  V_Exp NUMBER(4,2) := 1;
  6  CURSOR EmpInfoCursor IS
  7  SELECT Empno, Ename, Job, Sal, Comm
  8  FROM Emp
  9  WHERE Deptno = I_Deptno;
10  EmpRecord EmpInfoCursor%ROWTYPE;
11  BEGIN
12  OPEN EmpInfoCursor;
13  V_DeptRec := GetDeptDetails(I_Deptno);
14  DBMS_OUTPUT.PUT_LINE('Displaying The Report of...');
15  DBMS_OUTPUT.PUT_LINE('Department Number : '||V_DeptRec.Deptno||', Name :
'||V_DeptRec.DName||', Location : '||V_DeptRec.Loc);
16  LOOP
17  FETCH EmpInfoCursor INTO EmpRecord;
18  EXIT WHEN EmpInfoCursor%NOTFOUND;
19  DBMS_OUTPUT.PUT_LINE('Employee Name : '||EmpRecord.Ename);
20  DBMS_OUTPUT.PUT_LINE('Employee Job : '||EmpRecord.Job);
21  DBMS_OUTPUT.PUT_LINE('Employee Salary : '||EmpRecord.Sal);
22  DBMS_OUTPUT.PUT_LINE('Employee Comission : '||NVL(TO_CHAR(EmpRecord.Comm),
'NO COMMISSION'));
23  V_Exp := EmpExp(EmpRecord.Empno);
24  DBMS_OUTPUT.PUT_LINE('Employee's Experiance : '||V_Exp);
25  DBMS_OUTPUT.PUT_LINE('*****');
26  END LOOP;
27  CLOSE EmpInfoCursor;
28  END EmpInfo;
29  /
```

Procedure created.

```
SQL> EXEC EmpInfo(10);
Displaying The Report of...
Department Number : 10, Name : ACCOUNTING, Location : NEW YORK
Employee Name : KING
Employee Job : PRESIDENT
Employee Salary : 5000
Employee Comission : NO COMMISSION
Employee's Experiance : 28.79
*****
Employee Name : CLARK
Employee Job : MANAGER
Employee Salary : 2450
Employee Comission : NO COMMISSION
Employee's Experiance : 29.23
*****
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

Employee Name : MILLER
Employee Job : CLERK
Employee Salary : 1300
Employee Commission : NO COMMISSION
Employee's Experience : 28.61

PL/SQL procedure successfully completed.

SQL> cl scr

```
SQL> CREATE OR REPLACE PACKAGE MathsBody
 2 IS
 3 FUNCTION Factorial(Num NUMBER)
 4 RETURN NUMBER;
 5 FUNCTION Combination(Num1 NUMBER, Num2 NUMBER)
 6 RETURN NUMBER;
 7 PROCEDURE ProdSeries(StartRange NUMBER);
 8 PROCEDURE PrintEvenOdd(Num1 NUMBER, Num2 NUMBER);
 9 PROCEDURE SquareArea(Side IN NUMBER);
10 PROCEDURE CubeVolume(Radius IN NUMBER);
11 END MathsBody;
12 /
```

Package created.

```
SQL> CREATE OR REPLACE PACKAGE BODY MathsBody
 2 AS
 3 FUNCTION Factorial(Num NUMBER)
 4 RETURN NUMBER
 5 IS
 6 Fact NUMBER(4) := 1;
 7 BEGIN
 8 FOR MyIndex IN REVERSE 1..Num
 9 LOOP
10 Fact := Fact * MyIndex;
11 END LOOP;
12 RETURN Fact;
13 END Factorial;
14 FUNCTION Combination(Num1 NUMBER, Num2 NUMBER)
15 RETURN NUMBER
16 IS
17 Combi NUMBER(4,2) := 1;
18 BEGIN
19 Combi := (Factorial(Num1) / ( Factorial(Num1-Num2) * Factorial(Num2)));
20 RETURN Combi;
21 END Combination;
22 PROCEDURE ProdSeries(StartRange NUMBER)
23 IS
24 Result NUMBER;
25 BEGIN
26 FOR MyIndex IN 1..5 LOOP
27 Result := StartRange * MyIndex;
28 DBMS_OUTPUT.PUT_LINE(StartRange||' X '||MyIndex||' = '|| Result);
29 END LOOP;
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
30 END ProdSeries;
31 PROCEDURE PrintEvenOdd(Num1 NUMBER, Num2 NUMBER)
32 IS
33 V_Num1 NUMBER;
34 EvenValue VARCHAR2(1000);
35 OddValue VARCHAR2(1000);
36 BEGIN
37 V_Num1 := Num1;
38 WHILE V_Num1 < Num1
39 LOOP
40 IF MOD(V_Num1,2) != 0 THEN
41 OddValue := OddValue||' '||V_Num1;
42 ELSE
43 EvenValue := EvenValue||' '||V_Num1;
44 END IF;
45 V_Num1 := V_Num1 + 1;
46 END LOOP;
47 DBMS_OUTPUT.PUT_LINE('The Odd Numbers in The Series are : '||OddValue);
48 DBMS_OUTPUT.PUT_LINE('The Even Numbers in The Series are : '||EvenValue);
49 END PrintEvenOdd;
50 PROCEDURE SquareArea(Side IN NUMBER)
51 IS
52 BEGIN
53 DBMS_OUTPUT.PUT_LINE('Area of The Square = ' || (Side * Side) );
54 END;
55 PROCEDURE CubeVolume(Radius IN NUMBER)
56 IS
57 BEGIN
58 DBMS_OUTPUT.PUT_LINE('Volume of The Cube = ' || (Radius * Radius *
Radius));
59 END;
60 END MathsBody;
61 /
```

Package body created.

```
SQL> SELECT Factorial(5) FROM DUAL;
```

```
FACTORIAL(5)
-----
          120
```

```
SQL> SELECT MathsBody.Factorial(5) FROM DUAL;
```

```
MATHSBODY.FACTORIAL(5)
-----
                   120
```

```
SQL> DROP FUNCTION Factorial;
```

Function dropped.

```
SQL> SELECT Factorial(5) FROM DUAL;
SELECT Factorial(5) FROM DUAL
```

*

Spool Generated For Class of Oracle By Satish K Yellanki

ERROR at line 1:
ORA-00904: "FACTORIAL": invalid identifier

```
SQL> SELECT MathsBody.Factorial(5) FROM DUAL;
```

```
MATHSBODY.FACTORIAL(5)
-----
                        120
```

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PACKAGE EmpPackage
 2 IS
 3 PROCEDURE MyBonus;
 4 PROCEDURE FindEmp
 5 (I_Empno IN NUMBER,
 6  O_Ename OUT VARCHAR2,
 7  O_Job OUT VARCHAR2);
 8 PROCEDURE EmpInfo(I_Deptno IN NUMBER);
 9 FUNCTION EmpExp(V_Empno NUMBER)
10 RETURN NUMBER;
11 FUNCTION EmpGrade(I_Grade NUMBER)
12 RETURN VARCHAR2;
13 END EmpPackage;
14 /
```

Package created.

```
SQL> CREATE OR REPLACE PACKAGE BODY EmpPackage
 2 IS
 3 PROCEDURE MyBonus
 4 AS
 5 CURSOR DeptCursor IS
 6 SELECT Deptno FROM Dept;
 7 BEGIN
 8 FOR R_GroupBonus IN DeptCursor LOOP
 9 UPDATE Emp
10 SET Sal = Sal * 0.95
11 WHERE Deptno = R_GroupBonus.DeptNo;
12 DBMS_OUTPUT.PUT_LINE('The Bonus Information is '||R_GroupBonus.Deptno);
13 END LOOP;
14 END MyBonus;
15 PROCEDURE FindEmp
16 (I_Empno IN NUMBER,
17  O_Ename OUT VARCHAR2,
18  O_Job OUT VARCHAR2)
19 AS
20 BEGIN
21 SELECT Ename, Job INTO O_Ename, O_Job
22 FROM Emp WHERE Empno = I_Empno;
23 EXCEPTION
24 WHEN NO_DATA_FOUND THEN
25 DBMS_OUTPUT.PUT_LINE('Error in Finding the Details of Employee Number : '||
I_Empno);
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
26 END FindEmp;
27 PROCEDURE EmpInfo(I_Deptno IN NUMBER)
28 AS
29 CURSOR EmpInfoCursor IS
30 SELECT Ename, Job, Sal, Comm
31 FROM Emp
32 WHERE Deptno = I_Deptno;
33 EmpRecord EmpInfoCursor%ROWTYPE;
34 NEmployees NUMBER := 0;
35 TSalary NUMBER := 0;
36 AVGSalary NUMBER(7,2) := 0;
37 MAXSalary NUMBER(7,2) := 0;
38 BEGIN
39 OPEN EmpInfoCursor;
40 LOOP
41 FETCH EmpInfoCursor INTO EmpRecord;
42 EXIT WHEN EmpInfoCursor%NOTFOUND;
43 DBMS_OUTPUT.PUT_LINE('Employee Name : ' || EmpRecord.Ename);
44 DBMS_OUTPUT.PUT_LINE('Employee Job : ' || EmpRecord.Job);
45 DBMS_OUTPUT.PUT_LINE('Employee Salary : ' || EmpRecord.Sal);
46 DBMS_OUTPUT.PUT_LINE('Employee Comission : ' || EmpRecord.Comm);
47 DBMS_OUTPUT.PUT_LINE('*****');
48 TSalary := TSalary + EmpRecord.Sal;
49 NEmployees := NEmployees + 1;
50 IF EmpRecord.Sal > MAXSalary THEN
51 MAXSalary := EmpRecord.Sal;
52 END IF;
53 END LOOP;
54 AVGSalary := TSalary / NEmployees;
55 DBMS_OUTPUT.PUT_LINE('Number of Employees : ' || NEmployees);
56 DBMS_OUTPUT.PUT_LINE('Total Salary : ' || TSalary);
57 DBMS_OUTPUT.PUT_LINE('Maximum Salary : ' || MAXSalary);
58 DBMS_OUTPUT.PUT_LINE('Average Salary : ' || AVGSalary);
59 CLOSE EmpInfoCursor;
60 END EmpInfo;
61 FUNCTION EmpExp(V_Empno NUMBER)
62 RETURN NUMBER
63 IS
64 V_HireDate Emp.HireDate%TYPE;
65 V_Exp NUMBER(4,2) := 1;
66 BEGIN
67 SELECT HireDate INTO V_HireDate
68 FROM Emp
69 WHERE Empno = V_Empno;
70 V_Exp := MONTHS_BETWEEN(SYSDATE,V_HireDate) / 12;
71 RETURN V_Exp;
72 END EmpExp;
73 FUNCTION EmpGrade(I_Grade NUMBER)
74 RETURN VARCHAR2
75 IS
76 V_Num NUMBER(4);
77 BEGIN
78 SELECT COUNT(*) INTO V_Num
79 FROM Emp, SalGrade
80 WHERE Sal BETWEEN LoSal AND HiSal AND
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
81           Grade = I_Grade;
82 RETURN 'The Total Employees For The Grade Given By You Are : '||V_Num;
83 END EmpGrade;
84 END EmpPackage;
85 /
```

Package body created.

```
SQL> SELECT Ename, HireDate, EmpExp(Empno) EmpExp
2 FROM Emp;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.79
BLAKE	01-MAY-81	29.33
CLARK	09-JUN-81	29.23
JONES	02-APR-81	29.41
MARTIN	28-SEP-81	28.93
ALLEN	20-FEB-81	29.53
TURNER	08-SEP-81	28.98
JAMES	03-DEC-81	28.75
WARD	22-FEB-81	29.53
FORD	03-DEC-81	28.75
SMITH	17-DEC-80	29.71

ENAME	HIREDATE	EMPEXP
SCOTT	09-DEC-82	27.73
ADAMS	12-JAN-83	27.64
MILLER	23-JAN-82	28.61

14 rows selected.

```
SQL> SELECT Ename, HireDate, EmpPackage.EmpExp(Empno) EmpExp
2 FROM Emp;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.79
BLAKE	01-MAY-81	29.33
CLARK	09-JUN-81	29.23
JONES	02-APR-81	29.41
MARTIN	28-SEP-81	28.93
ALLEN	20-FEB-81	29.53
TURNER	08-SEP-81	28.98
JAMES	03-DEC-81	28.75
WARD	22-FEB-81	29.53
FORD	03-DEC-81	28.75
SMITH	17-DEC-80	29.71

ENAME	HIREDATE	EMPEXP
SCOTT	09-DEC-82	27.73
ADAMS	12-JAN-83	27.64
MILLER	23-JAN-82	28.61

Spool Generated For Class of Oracle By Satish K Yellanki

14 rows selected.

```
SQL> DROP FUNCTION EMPEXP;
```

Function dropped.

```
SQL> SELECT Ename, HireDate, EmpExp(Empno) EmpExp
  2 FROM Emp;
SELECT Ename, HireDate, EmpExp(Empno) EmpExp
      *
```

ERROR at line 1:

ORA-00904: "EMPEXP": invalid identifier

```
SQL> SELECT Ename, HireDate, EmpPackage.EmpExp(Empno) EmpExp
  2 FROM Emp;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.79
BLAKE	01-MAY-81	29.33
CLARK	09-JUN-81	29.23
JONES	02-APR-81	29.41
MARTIN	28-SEP-81	28.93
ALLEN	20-FEB-81	29.53
TURNER	08-SEP-81	28.98
JAMES	03-DEC-81	28.75
WARD	22-FEB-81	29.53
FORD	03-DEC-81	28.75
SMITH	17-DEC-80	29.71

ENAME	HIREDATE	EMPEXP
SCOTT	09-DEC-82	27.73
ADAMS	12-JAN-83	27.64
MILLER	23-JAN-82	28.61

14 rows selected.

```
SQL> SPOOL OFF
```

```
SQL> cl scr
```

```
SQL> SET VERIFY OFF
```

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> cl scr
```

```
SQL> SELECT MathsBody.Factorial(5) Fact FROM DUAL;
```

```
FACT
-----
120
```

```
SQL> SELECT MathsBody.Combination(6, 2) Combi FROM DUAL;
```

Spool Generated For Class of Oracle By Satish K Yellanki

COMBI

15

```
SQL> CREATE OR REPLACE PACKAGE MathsBody
  2 IS
  3 FUNCTION Combination(Num1 NUMBER, Num2 NUMBER)
  4 RETURN NUMBER;
  5 PROCEDURE ProdSeries(StartRange NUMBER);
  6 PROCEDURE PrintEvenOdd(Num1 NUMBER, Num2 NUMBER);
  7 PROCEDURE SquareArea(Side IN NUMBER);
  8 PROCEDURE CubeVolume(Radius IN NUMBER);
  9 END MathsBody;
10 /
```

Package created.

```
SQL> CREATE OR REPLACE PACKAGE BODY MathsBody
  2 AS
  3 FUNCTION Factorial(Num NUMBER)
  4 RETURN NUMBER
  5 IS
  6 Fact NUMBER(4) := 1;
  7 BEGIN
  8 FOR MyIndex IN REVERSE 1..Num
  9 LOOP
10 Fact := Fact * MyIndex;
11 END LOOP;
12 RETURN Fact;
13 END Factorial;
14 FUNCTION Combination(Num1 NUMBER, Num2 NUMBER)
15 RETURN NUMBER
16 IS
17 Combi NUMBER(4,2) := 1;
18 BEGIN
19 Combi := (Factorial(Num1) / ( Factorial(Num1-Num2) * Factorial(Num2)));
20 RETURN Combi;
21 END Combination;
22 PROCEDURE ProdSeries(StartRange NUMBER)
23 IS
24 Result NUMBER;
25 BEGIN
26 FOR MyIndex IN 1..5 LOOP
27 Result := StartRange * MyIndex;
28 DBMS_OUTPUT.PUT_LINE(StartRange||' X '||MyIndex||' = '|| Result);
29 END LOOP;
30 END ProdSeries;
31 PROCEDURE PrintEvenOdd(Num1 NUMBER, Num2 NUMBER)
32 IS
33 V_Num1 NUMBER;
34 EvenValue VARCHAR2(1000);
35 OddValue VARCHAR2(1000);
36 BEGIN
37 V_Num1 := Num1;
38 WHILE V_Num1 < Num1
```


Spool Generated For Class of Oracle By Satish K Yellanki

```
39 LOOP
40 IF MOD(V_Num1,2) != 0 THEN
41 OddValue := OddValue||' '|V_Num1;
42 ELSE
43 EvenValue := EvenValue||' '|V_Num1;
44 END IF;
45 V_Num1 := V_Num1 + 1;
46 END LOOP;
47 DBMS_OUTPUT.PUT_LINE('The Odd Numbers in The Series are : '|OddValue);
48 DBMS_OUTPUT.PUT_LINE('The Even Numbers in The Series are : '|EvenValue);
49 END PrintEvenOdd;
50 PROCEDURE SquareArea(Side IN NUMBER)
51 IS
52 BEGIN
53 DBMS_OUTPUT.PUT_LINE('Area of The Square = ' || (Side * Side) );
54 END;
55 PROCEDURE CubeVolume(Radius IN NUMBER)
56 IS
57 BEGIN
58 DBMS_OUTPUT.PUT_LINE('Volume of The Cube = ' || (Radius * Radius *
Radius));
59 END;
60 END MathsBody;
61 /
```

Package body created.

```
SQL> SELECT MathsBody.Factorial(5) Fact FROM DUAL;
SELECT MathsBody.Factorial(5) Fact FROM DUAL
      *
ERROR at line 1:
ORA-00904: "MATHSBODY"."FACTORIAL": invalid identifier
```

```
SQL> SELECT MathsBody.Combination(6, 2) Combi FROM DUAL;
```

```
      COMBI
-----
      15
```

```
SQL> cl scr
```

```
SQL> EXEC EmpPackage.EmpInfo(10);
Employee Name : KING
Employee Job : PRESIDENT
Employee Salary : 5000
Employee Comission :
*****
Employee Name : CLARK
Employee Job : MANAGER
Employee Salary : 2450
Employee Comission :
*****
Employee Name : MILLER
Employee Job : CLERK
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

Employee Salary : 1300
Employee Commission :

Number of Employees : 3
Total Salary : 8750
Maximum Salary : 5000
Average Salary : 2916.67

PL/SQL procedure successfully completed.

```
SQL> SELECT Ename, HireDate, EmpPackage.EmpExp(Empno) EmpExp
2 FROM Emp;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.79
BLAKE	01-MAY-81	29.34
CLARK	09-JUN-81	29.23
JONES	02-APR-81	29.42
MARTIN	28-SEP-81	28.93
ALLEN	20-FEB-81	29.54
TURNER	08-SEP-81	28.98
JAMES	03-DEC-81	28.75
WARD	22-FEB-81	29.53
FORD	03-DEC-81	28.75
SMITH	17-DEC-80	29.71

ENAME	HIREDATE	EMPEXP
SCOTT	09-DEC-82	27.73
ADAMS	12-JAN-83	27.64
MILLER	23-JAN-82	28.61

14 rows selected.

```
SQL> CREATE OR REPLACE PACKAGE BODY EmpPackage
2 IS
3 PROCEDURE MyBonus
4 AS
5 CURSOR DeptCursor IS
6 SELECT Deptno FROM Dept;
7 BEGIN
8 FOR R_GroupBonus IN DeptCursor LOOP
9 UPDATE Emp
10 SET Sal = Sal * 0.95
11 WHERE Deptno = R_GroupBonus.DeptNo;
12 DBMS_OUTPUT.PUT_LINE('The Bonus Information is '||R_GroupBonus.Deptno);
13 END LOOP;
14 END MyBonus;
15 PROCEDURE FindEmp
16 (I_Empno IN NUMBER,
17 O_Ename OUT VARCHAR2,
18 O_Job OUT VARCHAR2)
19 AS
20 BEGIN
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
21 SELECT Ename, Job INTO O_Ename, O_Job
22 FROM Emp WHERE Empno = I_Empno;
23 EXCEPTION
24 WHEN NO_DATA_FOUND THEN
25 DBMS_OUTPUT.PUT_LINE('Error in Finding the Details of Employee Number : '||
I_Empno);
26 END FindEmp;
27 PROCEDURE EmpInfo(I_Deptno IN NUMBER)
28 AS
29 CURSOR EmpInfoCursor IS
30 SELECT Empno, Ename, Job, Sal, Comm
31 FROM Emp
32 WHERE Deptno = I_Deptno;
33 EmpRecord EmpInfoCursor%ROWTYPE;
34 NEmployees NUMBER := 0;
35 TSalary NUMBER := 0;
36 AVGSalary NUMBER(7,2) := 0;
37 MAXSalary NUMBER(7,2) := 0;
38 V_EmpExp NUMBER(4, 2);
39 BEGIN
40 OPEN EmpInfoCursor;
41 LOOP
42 FETCH EmpInfoCursor INTO EmpRecord;
43 EXIT WHEN EmpInfoCursor%NOTFOUND;
44 DBMS_OUTPUT.PUT_LINE('Employee Name : '||EmpRecord.Ename);
45 DBMS_OUTPUT.PUT_LINE('Employee Job : '||EmpRecord.Job);
46 DBMS_OUTPUT.PUT_LINE('Employee Salary : '||EmpRecord.Sal);
47 DBMS_OUTPUT.PUT_LINE('Employee Comission : '||EmpRecord.Comm);
48 V_EmpExp := EmpExp(EmpRecord.Empno);
49 DBMS_OUTPUT.PUT_LINE('Employee Experience : '||V_EmpExp);
50 DBMS_OUTPUT.PUT_LINE('*****');
51 TSalary := TSalary + EmpRecord.Sal;
52 NEmployees := NEmployees + 1;
53 IF EmpRecord.Sal > MAXSalary THEN
54 MAXSalary := EmpRecord.Sal;
55 END IF;
56 END LOOP;
57 AVGSalary := TSalary / NEmployees;
58 DBMS_OUTPUT.PUT_LINE('Number of Employees : '||NEmployees);
59 DBMS_OUTPUT.PUT_LINE('Total Salary : '||TSalary);
60 DBMS_OUTPUT.PUT_LINE('Maximum Salary : '||MAXSalary);
61 DBMS_OUTPUT.PUT_LINE('Average Salary : '||AVGSalary);
62 CLOSE EmpInfoCursor;
63 END EmpInfo;
64 FUNCTION EmpExp(V_Empno NUMBER)
65 RETURN NUMBER
66 IS
67 V_HireDate Emp.HireDate%TYPE;
68 V_Exp NUMBER(4,2) := 1;
69 BEGIN
70 SELECT HireDate INTO V_HireDate
71 FROM Emp
72 WHERE Empno = V_Empno;
73 V_Exp := MONTHS_BETWEEN(SYSDATE,V_HireDate) / 12;
74 RETURN V_Exp;
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
75 END EmpExp;
76 FUNCTION EmpGrade(I_Grade NUMBER)
77 RETURN VARCHAR2
78 IS
79 V_Num NUMBER(4);
80 BEGIN
81 SELECT COUNT(*) INTO V_Num
82 FROM Emp, SalGrade
83 WHERE Sal BETWEEN LoSal AND HiSal AND
84         Grade = I_Grade;
85 RETURN 'The Total Employees For The Grade Given By You Are : ' || V_Num;
86 END EmpGrade;
87 END EmpPackage;
88 /
```

Package body created.

```
SQL> SELECT Ename, HireDate, EmpPackage.EmpExp(Empno) EmpExp
2 FROM Emp;
```

ENAME	HIREDATE	EMPEXP
KING	17-NOV-81	28.79
BLAKE	01-MAY-81	29.34
CLARK	09-JUN-81	29.23
JONES	02-APR-81	29.42
MARTIN	28-SEP-81	28.93
ALLEN	20-FEB-81	29.54
TURNER	08-SEP-81	28.98
JAMES	03-DEC-81	28.75
WARD	22-FEB-81	29.53
FORD	03-DEC-81	28.75
SMITH	17-DEC-80	29.71

ENAME	HIREDATE	EMPEXP
SCOTT	09-DEC-82	27.73
ADAMS	12-JAN-83	27.64
MILLER	23-JAN-82	28.61

14 rows selected.

```
SQL> EXEC EmpPackage.EmpInfo(10);
```

Employee Name : KING

Employee Job : PRESIDENT

Employee Salary : 5000

Employee Comission :

Employee Experience : 28.79

Employee Name : CLARK

Employee Job : MANAGER

Employee Salary : 2450

Employee Comission :

Employee Experience : 29.23

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

Employee Name : MILLER
Employee Job : CLERK
Employee Salary : 1300
Employee Commission :
Employee Experience : 28.61

Number of Employees : 3
Total Salary : 8750
Maximum Salary : 5000
Average Salary : 2916.67

PL/SQL procedure successfully completed.

SQL> cl scr

```
SQL> CREATE OR REPLACE PACKAGE EmpPackage
 2 IS
 3 PROCEDURE MyBonus;
 4 PROCEDURE FindEmp
 5 (I_Empno IN NUMBER,
 6 O_Ename OUT VARCHAR2,
 7 O_Job OUT VARCHAR2);
 8 PROCEDURE EmpInfo(I_Deptno IN NUMBER);
 9 FUNCTION EmpGrade(I_Grade NUMBER)
10 RETURN VARCHAR2;
11 END EmpPackage;
12 /
```

Package created.

```
SQL> CREATE OR REPLACE PACKAGE BODY EmpPackage
 2 IS
 3 PROCEDURE MyBonus
 4 AS
 5 CURSOR DeptCursor IS
 6 SELECT Deptno FROM Dept;
 7 BEGIN
 8 FOR R_GroupBonus IN DeptCursor LOOP
 9 UPDATE Emp
10 SET Sal = Sal * 0.95
11 WHERE Deptno = R_GroupBonus.DeptNo;
12 DBMS_OUTPUT.PUT_LINE('The Bonus Information is '||R_GroupBonus.Deptno);
13 END LOOP;
14 END MyBonus;
15 PROCEDURE FindEmp
16 (I_Empno IN NUMBER,
17 O_Ename OUT VARCHAR2,
18 O_Job OUT VARCHAR2)
19 AS
20 BEGIN
21 SELECT Ename, Job INTO O_Ename, O_Job
22 FROM Emp WHERE Empno = I_Empno;
23 EXCEPTION
24 WHEN NO_DATA_FOUND THEN
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
25 DBMS_OUTPUT.PUT_LINE('Error in Finding the Details of Employee Number : '||
I_Empno);
26 END FindEmp;
27 PROCEDURE EmpInfo(I_Deptno IN NUMBER)
28 AS
29 CURSOR EmpInfoCursor IS
30 SELECT Empno, Ename, Job, Sal, Comm
31 FROM Emp
32 WHERE Deptno = I_Deptno;
33 EmpRecord EmpInfoCursor%ROWTYPE;
34 NEmployees NUMBER := 0;
35 TSalary NUMBER := 0;
36 AVGSalary NUMBER(7,2) := 0;
37 MAXSalary NUMBER(7,2) := 0;
38 V_EmpExp NUMBER(4, 2);
39 BEGIN
40 OPEN EmpInfoCursor;
41 LOOP
42 FETCH EmpInfoCursor INTO EmpRecord;
43 EXIT WHEN EmpInfoCursor%NOTFOUND;
44 DBMS_OUTPUT.PUT_LINE('Employee Name : '||EmpRecord.Ename);
45 DBMS_OUTPUT.PUT_LINE('Employee Job : '||EmpRecord.Job);
46 DBMS_OUTPUT.PUT_LINE('Employee Salary : '||EmpRecord.Sal);
47 DBMS_OUTPUT.PUT_LINE('Employee Comission : '||EmpRecord.Comm);
48 V_EmpExp := EmpExp(EmpRecord.Empno);
49 DBMS_OUTPUT.PUT_LINE('Employee Experience : '||V_EmpExp);
50 DBMS_OUTPUT.PUT_LINE('*****');
51 TSalary := TSalary + EmpRecord.Sal;
52 NEmployees := NEmployees + 1;
53 IF EmpRecord.Sal > MAXSalary THEN
54 MAXSalary := EmpRecord.Sal;
55 END IF;
56 END LOOP;
57 AVGSalary := TSalary / NEmployees;
58 DBMS_OUTPUT.PUT_LINE('Number of Employees : '||NEmployees);
59 DBMS_OUTPUT.PUT_LINE('Total Salary : '||TSalary);
60 DBMS_OUTPUT.PUT_LINE('Maximum Salary : '||MAXSalary);
61 DBMS_OUTPUT.PUT_LINE('Average Salary : '||AVGSalary);
62 CLOSE EmpInfoCursor;
63 END EmpInfo;
64 FUNCTION EmpExp(V_Empno NUMBER)
65 RETURN NUMBER
66 IS
67 V_HireDate Emp.HireDate%TYPE;
68 V_Exp NUMBER(4,2) := 1;
69 BEGIN
70 SELECT HireDate INTO V_HireDate
71 FROM Emp
72 WHERE Empno = V_Empno;
73 V_Exp := MONTHS_BETWEEN(SYSDATE,V_HireDate) / 12;
74 RETURN V_Exp;
75 END EmpExp;
76 FUNCTION EmpGrade(I_Grade NUMBER)
77 RETURN VARCHAR2
78 IS
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
79 V_Num NUMBER(4);
80 BEGIN
81 SELECT COUNT(*) INTO V_Num
82 FROM Emp, SalGrade
83 WHERE Sal BETWEEN LoSal AND HiSal AND
84         Grade = I_Grade;
85 RETURN 'The Total Employees For The Grade Given By You Are : '||V_Num;
86 END EmpGrade;
87 END EmpPackage;
88 /
```

Warning: Package Body created with compilation errors.

SQL> SHOW ERRORS

Errors for PACKAGE BODY EMPPACKAGE:

LINE/COL ERROR

```
-----
48/1      PL/SQL: Statement ignored
48/13     PLS-00313: 'EMPEXP' not declared in this scope
```

SQL> ED

Wrote file afiedt.buf

```
 1 CREATE OR REPLACE PACKAGE BODY EmpPackage
 2 IS
 3 FUNCTION EmpExp(V_Empno NUMBER)
 4 RETURN NUMBER
 5 IS
 6 V_HireDate Emp.HireDate%TYPE;
 7 V_Exp NUMBER(4,2) := 1;
 8 BEGIN
 9 SELECT HireDate INTO V_HireDate
10 FROM Emp
11 WHERE Empno = V_Empno;
12 V_Exp := MONTHS_BETWEEN(SYSDATE,V_HireDate) / 12;
13 RETURN V_Exp;
14 END EmpExp;
15 PROCEDURE MyBonus
16 AS
17 CURSOR DeptCursor IS
18 SELECT Deptno FROM Dept;
19 BEGIN
20 FOR R_GroupBonus IN DeptCursor LOOP
21 UPDATE Emp
22 SET Sal = Sal * 0.95
23 WHERE Deptno = R_GroupBonus.DeptNo;
24 DBMS_OUTPUT.PUT_LINE('The Bonus Information is '||R_GroupBonus.Deptno);
25 END LOOP;
26 END MyBonus;
27 PROCEDURE FindEmp
28 (I_Empno IN NUMBER,
29  O_Ename OUT VARCHAR2,
30  O_Job OUT VARCHAR2)
31 AS
32 BEGIN
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
33 SELECT Ename, Job INTO O_Ename, O_Job
34 FROM Emp WHERE Empno = I_Empno;
35 EXCEPTION
36 WHEN NO_DATA_FOUND THEN
37 DBMS_OUTPUT.PUT_LINE('Error in Finding the Details of Employee Number : '||
I_Empno);
38 END FindEmp;
39 PROCEDURE EmpInfo(I_Deptno IN NUMBER)
40 AS
41 CURSOR EmpInfoCursor IS
42 SELECT Empno, Ename, Job, Sal, Comm
43 FROM Emp
44 WHERE Deptno = I_Deptno;
45 EmpRecord EmpInfoCursor%ROWTYPE;
46 NEmployees NUMBER := 0;
47 TSalary NUMBER := 0;
48 AVGSalary NUMBER(7,2) := 0;
49 MAXSalary NUMBER(7,2) := 0;
50 V_EmpExp NUMBER(4, 2);
51 BEGIN
52 OPEN EmpInfoCursor;
53 LOOP
54 FETCH EmpInfoCursor INTO EmpRecord;
55 EXIT WHEN EmpInfoCursor%NOTFOUND;
56 DBMS_OUTPUT.PUT_LINE('Employee Name : '||EmpRecord.Ename);
57 DBMS_OUTPUT.PUT_LINE('Employee Job : '||EmpRecord.Job);
58 DBMS_OUTPUT.PUT_LINE('Employee Salary : '||EmpRecord.Sal);
59 DBMS_OUTPUT.PUT_LINE('Employee Comission : '||EmpRecord.Comm);
60 V_EmpExp := EmpExp(EmpRecord.Empno);
61 DBMS_OUTPUT.PUT_LINE('Employee Experience : '||V_EmpExp);
62 DBMS_OUTPUT.PUT_LINE('*****');
63 TSalary := TSalary + EmpRecord.Sal;
64 NEmployees := NEmployees + 1;
65 IF EmpRecord.Sal > MAXSalary THEN
66 MAXSalary := EmpRecord.Sal;
67 END IF;
68 END LOOP;
69 AVGSalary := TSalary / NEmployees;
70 DBMS_OUTPUT.PUT_LINE('Number of Employees : '||NEmployees);
71 DBMS_OUTPUT.PUT_LINE('Total Salary : '||TSalary);
72 DBMS_OUTPUT.PUT_LINE('Maximum Salary : '||MAXSalary);
73 DBMS_OUTPUT.PUT_LINE('Average Salary : '||AVGSalary);
74 CLOSE EmpInfoCursor;
75 END EmpInfo;
76 FUNCTION EmpGrade(I_Grade NUMBER)
77 RETURN VARCHAR2
78 IS
79 V_Num NUMBER(4);
80 BEGIN
81 SELECT COUNT(*) INTO V_Num
82 FROM Emp, SalGrade
83 WHERE Sal BETWEEN LoSal AND HiSal AND
84 Grade = I_Grade;
85 RETURN 'The Total Employees For The Grade Given By You Are : '||V_Num;
86 END EmpGrade;
```


Spool Generated For Class of Oracle By Satish K Yellanki

```
87* END EmpPackage;  
SQL> /
```

Package body created.

```
SQL> SELECT Ename, HireDate, EmpPackage.EmpExp(Empno) EmpExp  
2 FROM Emp;  
SELECT Ename, HireDate, EmpPackage.EmpExp(Empno) EmpExp  
*
```

ERROR at line 1:

ORA-00904: "EMPPACKAGE"."EMPEXP": invalid identifier

```
SQL> EXEC EMPPACKAGE.EmpInfo(10)  
Employee Name : KING  
Employee Job : PRESIDENT  
Employee Salary : 5000  
Employee Commission :  
Employee Experience : 28.79  
*****  
Employee Name : CLARK  
Employee Job : MANAGER  
Employee Salary : 2450  
Employee Commission :  
Employee Experience : 29.23  
*****  
Employee Name : MILLER  
Employee Job : CLERK  
Employee Salary : 1300  
Employee Commission :  
Employee Experience : 28.61  
*****  
Number of Employees : 3  
Total Salary : 8750  
Maximum Salary : 5000  
Average Salary : 2916.67
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE PACKAGE EmpResultSet  
2 AS  
3 TYPE EmpRefCursor IS REF CURSOR;  
4 END EmpResultSet;  
5 /
```

Package created.

```
SQL> CREATE OR REPLACE FUNCTION  
2 FunctionEmpResultSet  
3 RETURN EmpResultSet.EmpRefCursor  
4 IS  
5 V_EmpResultSet EmpResultSet.EmpRefCursor;  
6 Emp_SQLSMTMT VARCHAR2(100);
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
7 BEGIN
8 Emp_SQLSMTMT := 'SELECT * FROM Emp';
9 OPEN V_EmpResultSet FOR Emp_SQLSMTMT;
10 RETURN V_EmpResultSet;
11 END FunctionEmpResultSet;
12 /
```

Function created.

```
SQL> COLUMN empno FORMAT 9999
SQL> COLUMN sal FORMAT 9999
SQL> COLUMN comm FORMAT 9999
SQL> SELECT FunctionEmpResultSet FROM DUAL;
```

FUNCTIONEMPRESULTSET

CURSOR STATEMENT : 1

CURSOR STATEMENT : 1

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> ED
```

Wrote file afiedt.buf

```
1 CREATE OR REPLACE FUNCTION
2 FunctionEmpResultSet(P_Query VARCHAR2)
3 RETURN EmpResultSet.EmpRefCursor
4 IS
5 V_EmpResultSet EmpResultSet.EmpRefCursor;
6 Emp_SQLSMTMT VARCHAR2(400) := P_Query;
7 BEGIN
8 OPEN V_EmpResultSet FOR Emp_SQLSMTMT;
9 RETURN V_EmpResultSet;
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
10* END FunctionEmpResultSet;  
SQL> /
```

Function created.

```
SQL> SELECT FunctionEmpResultSet('&GiveQuery') FROM DUAL;  
Enter value for givequery: SELECT * FROM Emp
```

FUNCTIONEMPRESULTSET

CURSOR STATEMENT : 1

CURSOR STATEMENT : 1

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> /  
Enter value for givequery: SELECT * FROM Dept
```

FUNCTIONEMPRESULTSET

CURSOR STATEMENT : 1

CURSOR STATEMENT : 1

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> /
```

Spool Generated For Class of Oracle By Satish K Yellanki

Enter value for givequery: SELECT * FROM SalGrade

FUNCTIONEMPRESULTSET

CURSOR STATEMENT : 1

CURSOR STATEMENT : 1

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

SQL> /

Enter value for givequery: SELECT Ename, Dname, Sal, Grade FROM Emp, Dept, SalGrade WHERE Emp.Deptno = Dept.Deptno AND Emp.Sal BETWEEN LoSal AND HiSal

FUNCTIONEMPRESULTSET

CURSOR STATEMENT : 1

CURSOR STATEMENT : 1

ENAME	DNAME	SAL	GRADE
SMITH	RESEARCH	800	1
JAMES	SALES	950	1
ADAMS	RESEARCH	1100	1
MARTIN	SALES	1250	2
WARD	SALES	1250	2
MILLER	ACCOUNTING	1300	2
TURNER	SALES	1500	3
ALLEN	SALES	1600	3
CLARK	ACCOUNTING	2450	4
BLAKE	SALES	2850	4
JONES	RESEARCH	2975	4

ENAME	DNAME	SAL	GRADE
FORD	RESEARCH	3000	4
SCOTT	RESEARCH	3000	4
KING	ACCOUNTING	5000	5

14 rows selected.

SQL> cl scr

```
SQL> CREATE OR REPLACE PACKAGE Employee_RefCur_PKG
2 AS
3 TYPE EmpCursor IS REF CURSOR;
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
4 PROCEDURE
5 EmployeeSearch
6 (
7   IN_EName      IN VARCHAR2,
8   OUT_EmpCursor OUT EmpCursor);
9 END Employee_RefCur_PKG;
10 /
```

Package created.

```
SQL> CREATE OR REPLACE PACKAGE BODY Employee_RefCur_PKG
2 AS
3 PROCEDURE
4 EmployeeSearch
5 (
6   IN_EName      IN VARCHAR2,
7   OUT_EmpCursor OUT EmpCursor
8 )
9 IS
10 BEGIN
11 OPEN OUT_EmpCursor
12 FOR
13 SELECT Emp.Empno, Emp.Ename, Emp.Job,
14        Emp.Sal, Dept.Dname, Dept.Loc
15 FROM Emp, Dept
16 WHERE Ename LIKE UPPER('%' || IN_EName || '%')
17        AND
18        Emp.Deptno = Dept.Deptno
19 ORDER BY UPPER(Emp.Ename);
20 END EmployeeSearch;
21 END Employee_RefCur_PKG;
22 /
```

Package body created.

```
SQL> DECLARE
2 O_EmpCursor Employee_RefCur_PKG.EmpCursor;
3 TYPE RefData IS RECORD
4 (
5   Empno Emp.Empno%TYPE,
6   Ename Emp.Ename%TYPE,
7   Job Emp.Job%TYPE,
8   Sal Emp.Sal%TYPE,
9   Dname Dept.Dname%TYPE,
10  Loc Dept.Loc%TYPE
11 );
12 MyRefData RefData;
13 IN_EName VARCHAR2(10) := '&GName';
14 TYPE MyArray IS
15 TABLE OF
16 MyRefData%TYPE;
17 V_MyArray MyArray;
18 BEGIN
19 Employee_RefCur_PKG.EmployeeSearch(IN_EName, O_EmpCursor);
20 FETCH O_EmpCursor BULK COLLECT INTO V_MyArray;
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
21 CLOSE O_EmpCursor;
22 FOR LoopIndex IN 1 .. V_MyArray.Count
23 LOOP
24 DBMS_OUTPUT.PUT_LINE(V_MyArray(LoopIndex).Empno||',
'|V_MyArray(LoopIndex).Ename||', '| V_MyArray(LoopIndex).Job||',
'|V_MyArray(LoopIndex).Sal||', '|V_MyArray(LoopIndex).Dname||', '|
V_MyArray(LoopIndex).Loc);
25 END LOOP;
26 END;
27 /
```

Enter value for gname: Smith
7369, SMITH, CLERK, 800, RESEARCH, DALLAS

PL/SQL procedure successfully completed.

```
SQL> /
Enter value for gname: S
7876, ADAMS, CLERK, 1100, RESEARCH, DALLAS
7900, JAMES, CLERK, 950, SALES, CHICAGO
7566, JONES, MANAGER, 2975, RESEARCH, DALLAS
7788, SCOTT, ANALYST, 3000, RESEARCH, DALLAS
7369, SMITH, CLERK, 800, RESEARCH, DALLAS
```

PL/SQL procedure successfully completed.

```
SQL> /
Enter value for gname: A
7876, ADAMS, CLERK, 1100, RESEARCH, DALLAS
7499, ALLEN, SALESMAN, 1600, SALES, CHICAGO
7698, BLAKE, MANAGER, 2850, SALES, CHICAGO
7782, CLARK, MANAGER, 2450, ACCOUNTING, NEW YORK
7900, JAMES, CLERK, 950, SALES, CHICAGO
7654, MARTIN, SALESMAN, 1250, SALES, CHICAGO
7521, WARD, SALESMAN, 1250, SALES, CHICAGO
```

PL/SQL procedure successfully completed.

```
SQL> cl scr
```

```
SQL> DELETE FROM Emp;
```

14 rows deleted.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER WeekEndCheck
2 AFTER INSERT OR UPDATE OR DELETE
3 ON Emp
4 DECLARE
5 V_WeekDay VARCHAR2(100);
6 BEGIN
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
7 V_WeekDay := TO_CHAR(SYSDATE,'DY');
8 IF V_WeekDay = 'SAT' OR V_WeekDay = 'SUN' THEN
9 RAISE_APPLICATION_ERROR(-20010,'An Illegal Intrusion into the System was
Detected.');
```

Trigger created.

```
SQL> DELETE FROM Emp;
```

14 rows deleted.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> DELETE FROM Emp;
```

```
DELETE FROM Emp
```

*

ERROR at line 1:

ORA-20010: An Illegal Intrusion into the System was Detected.

ORA-06512: at "SCOTT.WEEKENDCHECK", line 6

ORA-04088: error during execution of trigger 'SCOTT.WEEKENDCHECK'

```
SQL> cl scr
```

```
SQL> DROP TRIGGER WeekEndCheck;
```

Trigger dropped.

```
SQL> cl scr
```

```
SQL> DELETE FROM Emp;
```

14 rows deleted.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> CREATE OR REPLACE TRIGGER illegalTime
```

```
2 BEFORE INSERT OR UPDATE OR DELETE
```

```
3 ON Emp
```

```
4 DECLARE
```

```
5 V_Time NUMBER;
```

```
6 BEGIN
```

```
7 V_Time := TO_CHAR(SYSDATE, 'HH24');
```

```
8 IF V_Time NOT BETWEEN 10 AND 17 THEN
```

```
9 RAISE_APPLICATION_ERROR(-20011,' Illegal Intrusion, Not Business Hours.');
```

```
10 END IF;
```

```
11 END;
```

```
12 /
```

Spool Generated For Class of Oracle By Satish K Yellanki

Trigger created.

```
SQL> DELETE FROM Emp;
DELETE FROM Emp
```

*

ERROR at line 1:

```
ORA-20011: Illegal Intrusion, Not Business Hours.
ORA-06512: at "SCOTT.ILLEGALTIME", line 6
ORA-04088: error during execution of trigger 'SCOTT.ILLEGALTIME'
```

```
SQL> CREATE OR REPLACE TRIGGER WeekEndCheck
 2 AFTER INSERT OR UPDATE OR DELETE
 3 ON Emp
 4 DECLARE
 5 V_WeekDay VARCHAR2(100);
 6 BEGIN
 7 V_WeekDay := TO_CHAR(SYSDATE,'DY');
 8 IF V_WeekDay = 'SAT' OR V_WeekDay = 'SUN' THEN
 9 RAISE_APPLICATION_ERROR(-20010,'An Illegal Intrusion into the System was
Detected.');
```

```
10 END IF;
11 END;
12 /
```

Trigger created.

```
SQL> DELETE FROM Emp;
DELETE FROM Emp
```

*

ERROR at line 1:

```
ORA-20011: Illegal Intrusion, Not Business Hours.
ORA-06512: at "SCOTT.ILLEGALTIME", line 6
ORA-04088: error during execution of trigger 'SCOTT.ILLEGALTIME'
```

```
SQL> SPOOL OFF
```

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER WeekEndCheck
 2 AFTER INSERT OR UPDATE OR DELETE
 3 ON Emp
 4 DECLARE
 5 V_WeekDay VARCHAR2(100);
 6 BEGIN
 7 V_WeekDay := TO_CHAR(SYSDATE,'DY');
 8 IF V_WeekDay = 'SAT' OR V_WeekDay = 'SUN' THEN
 9 RAISE_APPLICATION_ERROR(-20010,'An Illegal Intrusion into the System was
Detected.');
```

```
10 END IF;
11 END;
12 /
```

Trigger created.

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> DELETE FROM Emp
  2  WHERE Deptno = 30;
DELETE FROM Emp
      *
ERROR at line 1:
ORA-20010: An Illegal Intrusion into the System was Detected.
ORA-06512: at "SCOTT.WEEKENDCHECK", line 6
ORA-04088: error during execution of trigger 'SCOTT.WEEKENDCHECK'
```

```
SQL> CREATE OR REPLACE TRIGGER WeekEndCheck
  2  AFTER INSERT OR UPDATE OR DELETE
  3  ON Emp
  4  FOR EACH ROW
  5  DECLARE
  6  V_WeekDay VARCHAR2(100);
  7  BEGIN
  8  V_WeekDay := TO_CHAR(SYSDATE,'DY');
  9  IF V_WeekDay = 'SAT' OR V_WeekDay = 'SUN' THEN
 10  RAISE_APPLICATION_ERROR(-20010,'An Illegal Intrusion into the System was
Detected.');
```

Trigger created.

```
SQL> DELETE FROM Emp
  2  WHERE Deptno = 30;
DELETE FROM Emp
      *
ERROR at line 1:
ORA-20010: An Illegal Intrusion into the System was Detected.
ORA-06512: at "SCOTT.WEEKENDCHECK", line 6
ORA-04088: error during execution of trigger 'SCOTT.WEEKENDCHECK'
```

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER WeekEndCheck
  2  AFTER INSERT OR UPDATE OR DELETE
  3  ON Emp
  4  DECLARE
  5  V_WeekDay VARCHAR2(100);
  6  BEGIN
  7  V_WeekDay := TO_CHAR(SYSDATE,'DY');
  8  IF V_WeekDay = 'SAT' OR V_WeekDay = 'SUN' THEN
  9  DBMS_OUTPUT.PUT_LINE('An Illegal Intrusion into the System was Detected.');
```

Trigger created.

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> DELETE FROM Emp
      2 WHERE Deptno = 30;
```

An Illegal Intrusion into the System was Detected.

6 rows deleted.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER WeekEndCheck
      2 AFTER INSERT OR UPDATE OR DELETE
      3 ON Emp
      4 FOR EACH ROW
      5 DECLARE
      6 V_WeekDay VARCHAR2(100);
      7 BEGIN
      8 V_WeekDay := TO_CHAR(SYSDATE,'DY');
      9 IF V_WeekDay = 'SAT' OR V_WeekDay = 'SUN' THEN
     10 DBMS_OUTPUT.PUT_LINE('An Illegal Intrusion into the System was Detected.');
```

Trigger created.

```
SQL> DELETE FROM EMP
      2 WHERE Empno = 7654;
```

An Illegal Intrusion into the System was Detected.

1 row deleted.

```
SQL> DELETE FROM EMP
      2 WHERE Deptno = 30;
```

An Illegal Intrusion into the System was Detected.

An Illegal Intrusion into the System was Detected.

An Illegal Intrusion into the System was Detected.

An Illegal Intrusion into the System was Detected.

An Illegal Intrusion into the System was Detected.

5 rows deleted.

```
SQL> cl scr
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE VIEW DeptView
      2 AS
      3 SELECT Deptno, Dname, Loc
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
4 FROM Dept
5 /
```

View created.

```
SQL> DESC DeptView
```

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

```
SQL> SELECT * FROM DeptView;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> INSERT INTO DeptView
2 VALUES(50, 'SHIPPING', 'CHENNAI');
```

1 row created.

```
SQL>
```

```
SQL> SELECT * FROM DeptView;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	SHIPPING	CHENNAI

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> SELECT * FROM DeptView;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> CREATE OR REPLACE TRIGGER DeptDel
2 INSTEAD OF DELETE ON DeptView
3 FOR EACH ROW
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
4 BEGIN
5 DELETE FROM Dept
6 WHERE Deptno = :OLD.Deptno;
7 DBMS_OUTPUT.PUT_LINE('Trigger Fired, Data Deleted');
8 END;
9 /
```

Trigger created.

```
SQL> DELETE FROM DeptView
2 WHERE Deptno = 40;
Trigger Fired, Data Deleted
```

1 row deleted.

```
SQL> cl scr
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE VIEW Employees_Dept_Info
2 AS
3 SELECT Dname, Ename
4 FROM Emp, Dept
5 WHERE Dept.DeptNo = Emp.DeptNo;
```

View created.

```
SQL> CREATE OR REPLACE TRIGGER Insert_Dept_Info
2 INSTEAD OF INSERT ON Employees_Dept_Info
3 DECLARE
4 V_DeptNo Dept.Deptno%TYPE;
5 Duplicate_Record EXCEPTION;
6 PRAGMA EXCEPTION_INIT(Duplicate_Record, -00001);
7 BEGIN
8 SELECT MAX(Deptno) INTO V_Deptno
9 FROM Dept;
10 V_Deptno := V_deptno + 10;
11 INSERT INTO Dept
12 VALUES(V_Deptno,:NEW.Dname, 'Not Confirmed');
13 EXCEPTION
14 WHEN Duplicate_Record THEN
15 RAISE_APPLICATION_ERROR(num=> -20107, msg=> 'Duplicated Department
Number!');
16 END Insert_Dept_Info;
17 /
```

Trigger created.

```
SQL> INSERT INTO Employees_Dept_Info
2 VALUES('SHIPPING', NULL);
```

Spool Generated For Class of Oracle By Satish K Yellanki

1 row created.

```
SQL> SELECT * FROM Dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	SHIPPING	Not Confirmed

```
SQL> cl scr
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> DELETE FROM Emp
      2 WHERE Deptno = 30;
```

6 rows deleted.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER Tab_ReadOnly
      2 BEFORE DELETE OR INSERT OR UPDATE
      3 ON Emp
      4 FOR EACH ROW
      5 BEGIN
      6   RAISE_APPLICATION_ERROR(-20201, 'Table Status: READ ONLY.');
```

Trigger created.

```
SQL> DELETE FROM Emp
      2 WHERE Deptno = 30;
```

```
DELETE FROM Emp
      *
```

ERROR at line 1:

ORA-20201: Table Status: READ ONLY.

ORA-06512: at "SCOTT.TAB_READONLY", line 2

ORA-04088: error during execution of trigger 'SCOTT.TAB_READONLY'

```
SQL> SELECT Ename FROM Emp;
```

ENAME

Spool Generated For Class of Oracle By Satish K Yellanki

```
-----  
KING  
BLAKE  
CLARK  
JONES  
MARTIN  
ALLEN  
TURNER  
JAMES  
WARD  
FORD  
SMITH  
  
ENAME  
-----  
SCOTT  
ADAMS  
MILLER  
  
14 rows selected.  
  
SQL> cl scr  
  
SQL> DROP TRIGGER Tab_ReadOnly;  
  
Trigger dropped.  
  
SQL> cl scr  
  
SQL> CREATE TABLE EmpAudit  
2  (  
3  UserName VARCHAR2(20),  
4  TranDate DATE,  
5  TranTime DATE,  
6  TranType VARCHAR2(10)  
7  );  
  
Table created.  
  
SQL> COLUMN UserName FORMAT A10  
SQL> cl sscr  
SP2-0158: unknown CLEAR option "sscr"  
SQL> cl scr  
  
SQL> SELECT * FROM EmpAudit;  
  
no rows selected  
  
SQL> CREATE OR REPLACE TRIGGER EmpAuditTrg  
2  AFTER INSERT OR DELETE OR UPDATE  
3  ON Emp  
4  DECLARE  
5  V_TranType EmpAudit.TranType%TYPE;  
6  BEGIN  
7  IF INSERTING THEN
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
8 V_TranType := 'Insert';
9 ELSIF UPDATING THEN
10 V_TranType := 'Update';
11 ELSE
12 V_TranType := 'Delete';
13 END IF;
14 INSERT INTO EmpAudit
15 VALUES(USER, TO_DATE(TO_CHAR(SYSDATE, 'DD-MON-YYYY'), 'DD-MON-YYYY'),
TO_DATE(TO_CHAR(SYSDATE, 'HH24:MI:SS'), 'HH24:MI:SS'), V_TranType);
16 END;
17 /
```

Trigger created.

```
SQL> DELETE FROM Emp
2 WHERE Empno = 7654;
```

1 row deleted.

```
SQL> SELECT * FROM EmpAudit;
```

USERNAME	TRANDATE	TRANTIME	TRANTYPE
SCOTT	03-SEP-10	01-SEP-10	Delete

```
SQL> DELETE FROM Emp
2 WHERE Deptno = 30;
```

5 rows deleted.

```
SQL> SELECT * FROM EmpAudit;
```

USERNAME	TRANDATE	TRANTIME	TRANTYPE
SCOTT	03-SEP-10	01-SEP-10	Delete
SCOTT	03-SEP-10	01-SEP-10	Delete

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER EmpAuditTrg
2 AFTER INSERT OR DELETE OR UPDATE
3 ON Emp
4 FOR EACH ROW
5 DECLARE
6 V_TranType EmpAudit.TranType%TYPE;
7 BEGIN
8 IF INSERTING THEN
9 V_TranType := 'Insert';
10 ELSIF UPDATING THEN
11 V_TranType := 'Update';
12 ELSE
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
13 V_TranType := 'Delete';
14 END IF;
15 INSERT INTO EmpAudit
16 VALUES(USER, TO_DATE(TO_CHAR(SYSDATE, 'DD-MON-YYYY'), 'DD-MON-YYYY'),
TO_DATE(TO_CHAR(SYSDATE, 'HH24:MI:SS'), 'HH24:MI:SS'), V_TranType);
17 END;
18 /
```

Trigger created.

```
SQL> DELETE FROM Emp
      2 WHERE Empno = 7654;
```

1 row deleted.

```
SQL> SELECT * FROM EmpAudit;
```

USERNAME	TRANDATE	TRANTIME	TRANTYPE
SCOTT	03-SEP-10	01-SEP-10	Delete

```
SQL> DELETE FROM Emp
      2 WHERE Deptno = 30;
```

5 rows deleted.

```
SQL> SELECT * FROM EmpAudit;
```

USERNAME	TRANDATE	TRANTIME	TRANTYPE
SCOTT	03-SEP-10	01-SEP-10	Delete
SCOTT	03-SEP-10	01-SEP-10	Delete
SCOTT	03-SEP-10	01-SEP-10	Delete
SCOTT	03-SEP-10	01-SEP-10	Delete
SCOTT	03-SEP-10	01-SEP-10	Delete
SCOTT	03-SEP-10	01-SEP-10	Delete

6 rows selected.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> cl scr
```

```
SQL> CREATE TABLE Emp_Audit_Table
      2 (
      3   AuditSeq          NUMBER,
      4   UserName          VARCHAR2(10),
      5   TimeTaran        DATE,
      6   UserTerm         VARCHAR2(10),
      7   TranType        VARCHAR2(10),
      8   ProcDone        VARCHAR2(10),
```


Spool Generated For Class of Oracle By Satish K Yellanki

```
9      Enum          NUMBER
10 );
```

Table created.

```
SQL> COLUMN AuditSeq FORMAT 999
SQL> COLUMN UserName FORMAT A6
SQL> COLUMN UserTerm FORMAT A6
SQL> COLUMN TranType FORMAT A7
SQL> COLUMN ProcDone FORMAT A7
SQL> SELECT * FROM Emp_Audit_Table;
```

no rows selected

```
SQL> CREATE SEQUENCE Audit_Seq;
```

Sequence created.

```
SQL> CREATE TABLE Audit_Table_Values
2  (
3      Audit_Seq          NUMBER,
4      ColumnName        VARCHAR2(10),
5      OldValue          NUMBER,
6      NewValue          NUMBER
7  );
```

Table created.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER Audit_On_Emp_Table
2  AFTER INSERT OR UPDATE OR DELETE
3  ON Emp
4  FOR EACH ROW
5  DECLARE
6  Tran_Time          DATE;
7  User_Terminal     VARCHAR2(10);
8  BEGIN
9  Tran_Time := SYSDATE;
10 User_Terminal := USERENV('TERMINAL');
11 IF INSERTING THEN
12   INSERT INTO Emp_Audit_Table
13   VALUES(
14           Audit_Seq.NEXTVAL,
15           USER,
16           Tran_Time,
17           User_Terminal,
18           'Emp',
19           'INSERT',
20           :NEW.Empno
21   );
22 ELSIF DELETING THEN
23   INSERT INTO Emp_Audit_Table
24   VALUES(
25           Audit_Seq.NEXTVAL,
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
26         USER,
27         Tran_Time,
28         User_Terminal,
29         'Emp',
30         'DELETE',
31         :OLD.Empno
32     );
33 ELSE
34     INSERT INTO Emp_Audit_Table
35     VALUES (
36         Audit_Seq.NEXTVAL,
37         USER,
38         Tran_Time,
39         User_Terminal,
40         'Emp',
41         'UPDATE',
42         :OLD.Empno
43     );
44 IF UPDATING('SAL') THEN
45     INSERT INTO Audit_Table_Values
46     VALUES(
47         Audit_Seq.CURRVAL,
48         'SAL',
49         :OLD.Sal,
50         :NEW.Sal
51     );
52 ELSIF UPDATING ('DEPTNO') THEN
53     INSERT INTO Audit_Table_Values
54     VALUES(
55         Audit_Seq.CURRVAL,
56         'DEPTNO',
57         :OLD.Deptno,
58         :NEW.DEPTNO
59     );
60 END IF;
61 END IF;
62 END;
63 /
```

Trigger created.

```
SQL> INSERT INTO Emp(Empno, Ename, Deptno, Sal)
  2 VALUES(1234, 'SAMPLE01', 30, 2000);
```

1 row created.

```
SQL> SELECT * FROM EmpAudit;
```

```
USERNA  TRANDATE  TRANTIME  TRANTYP
-----  -
SCOTT   03-SEP-10  01-SEP-10  Insert
```

```
SQL> SELECT * FROM Emp_Audit_Table;
```

```
AUDITSEQ  USERNA  TIMETARAN  USERTE  TRANTYP  PROCDON  ENUM
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
-----  
1 SCOTT 03-SEP-10 NIT Emp INSERT 1234
```

```
SQL> DELETE FROM Emp  
2 WHERE Empno = 1234;
```

1 row deleted.

```
SQL> SELECT * FROM EmpAudit;
```

```
USERNA TRANDATE TRANTIME TRANTYP  
-----  
SCOTT 03-SEP-10 01-SEP-10 Insert  
SCOTT 03-SEP-10 01-SEP-10 Delete
```

```
SQL> SELECT * FROM Emp_Audit_Table;
```

```
AUDITSEQ USERNA TIMETARAN USERTE TRANTYP PROCDON ENUM  
-----  
1 SCOTT 03-SEP-10 NIT Emp INSERT 1234  
2 SCOTT 03-SEP-10 NIT Emp DELETE 1234
```

```
SQL> UPDATE Emp  
2 SET Sal = Sal + 1000  
3 WHERE Empno = 7654;
```

1 row updated.

```
SQL> SELECT * FROM EmpAudit;
```

```
USERNA TRANDATE TRANTIME TRANTYP  
-----  
SCOTT 03-SEP-10 01-SEP-10 Insert  
SCOTT 03-SEP-10 01-SEP-10 Delete  
SCOTT 03-SEP-10 01-SEP-10 Update
```

```
SQL> SELECT * FROM Emp_Audit_Table;
```

```
AUDITSEQ USERNA TIMETARAN USERTE TRANTYP PROCDON ENUM  
-----  
1 SCOTT 03-SEP-10 NIT Emp INSERT 1234  
2 SCOTT 03-SEP-10 NIT Emp DELETE 1234  
3 SCOTT 03-SEP-10 NIT Emp UPDATE 7654
```

```
SQL> SELECT * FROM Audit_Table_Values  
2 /
```

```
AUDIT_SEQ COLUMNNAME OLDVALUE NEWVALUE  
-----  
3 SAL 1250 2250
```

```
SQL> UPDATE Emp  
2 SET  
3 Deptno = 20  
4 WHERE Empno = 7654;
```

Spool Generated For Class of Oracle By Satish K Yellanki

1 row updated.

```
SQL> SELECT * FROM Audit_Table_Values
2 /
```

AUDIT_SEQ	COLUMNNAME	OLDVALUE	NEWVALUE
3	SAL	1250	2250
4	DEPTNO	30	20

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE TABLE RecycleBin
2 (
3   Empno NUMBER(6),
4   Ename VARCHAR2(20),
5   Job VARCHAR(20),
6   MGR NUMBER(6),
7   HireDate DATE,
8   Sal NUMBER(7,2),
9   Comm NUMBER(7,2),
10  Deptno NUMBER(2)
11 )
12 /
```

Table created.

```
SQL> COLUMN Empno FORMAT 9999
SQL> COLUMN Sal FORMAT 9999
SQL> COLUMN Comm FORMAT 9999
SQL> COLUMN Ename FORMAT A10
SQL> COLUMN Job FORMAT A14
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER EmpRBin
2 BEFORE DELETE
3 ON Emp
4 FOR EACH ROW
5 BEGIN
6   INSERT INTO RecycleBin
7
```

```
VALUES (:OLD.Empno, :OLD.Ename, :OLD.Job, :OLD.MGR, :OLD.HireDate, :OLD.Sal, :OLD
.Comm, :OLD.Deptno);
8 END;
9 /
```

Trigger created.

```
SQL> DELETE FROM Emp
2 WHERE Empno = 7654;
```

Spool Generated For Class of Oracle By Satish K Yellanki

1 row deleted.

```
SQL> SELECT Ename FROM Emp WHERE Empno = 7654;
```

no rows selected

```
SQL> SELECT * FROM RecycleBin;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30

```
SQL> DELETE FROM Emp
      2 WHERE Deptno = 30;
```

5 rows deleted.

```
SQL> SELECT * FROM RecycleBin;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		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

6 rows selected.

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER PresidentNo
      2 BEFORE DELETE ON Emp
      3 FOR EACH ROW
      4 WHEN (OLD.Job = 'PRESIDENT')
      5 BEGIN
      6 RAISE_APPLICATION_ERROR(-20555, 'Sorry President Cannot be Deleted...');
      7 END PresidentNo;
      8 /
```

Trigger created.

```
SQL> DELETE FROM Emp WHERE Empno = 7654;
```

1 row deleted.

```
SQL> ROLLBACK;
```

Rollback complete.

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> DELETE FROM Emp WHERE Ename = 'KING';
DELETE FROM Emp WHERE Ename = 'KING'
      *
ERROR at line 1:
ORA-20555: Sorry President Cannot be Deleted...
ORA-06512: at "SCOTT.PRESIDENTNO", line 2
ORA-04088: error during execution of trigger 'SCOTT.PRESIDENTNO'
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> SELECT Ename, Sal FROM Emp;
```

ENAME	SAL
KING	5000
BLAKE	2850
CLARK	2450
JONES	2975
MARTIN	1250
ALLEN	1600
TURNER	1500
JAMES	950
WARD	1250
FORD	3000
SMITH	800

ENAME	SAL
SCOTT	3000
ADAMS	1100
MILLER	1300

14 rows selected.

```
SQL> CREATE OR REPLACE TRIGGER ChkSalary
  2 BEFORE UPDATE OF Sal ON Emp
  3 FOR EACH ROW
  4 WHEN (NEW.Sal < OLD.Sal)
  5 BEGIN
  6 RAISE_APPLICATION_ERROR(-20500, 'Salary Cannot be Decreased...');
  7 END;
  8 /
```

Trigger created.

```
SQL> UPDATE Emp
  2 SET Sal = 1000
  3 WHERE Ename = 'ADAMS';
UPDATE Emp
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
*
ERROR at line 1:
ORA-20500: Salary Cannot be Decreased...
ORA-06512: at "SCOTT.CHK$ALARY", line 2
ORA-04088: error during execution of trigger 'SCOTT.CHK$ALARY'
```

```
SQL> ROLLBACK;
```

```
Rollback complete.
```

```
SQL> cl scr
```

```
SQL> CREATE OR REPLACE TRIGGER GenEmpno
 2 BEFORE INSERT ON Emp
 3 FOR EACH ROW
 4 DECLARE
 5 V_Empno Emp.Empno%TYPE;
 6 BEGIN
 7 SELECT MAX(Empno) INTO V_Empno
 8 FROM Emp;
 9 :NEW.Empno := V_Empno + 1;
10 End GenEmpno;
11 /
```

```
Trigger created.
```

```
SQL> SELECT Empno, Ename FROM Emp;
```

```
EMPNO ENAME
-----
7839 KING
7698 BLAKE
7782 CLARK
7566 JONES
7654 MARTIN
7499 ALLEN
7844 TURNER
7900 JAMES
7521 WARD
7902 FORD
7369 SMITH
```

```
EMPNO ENAME
-----
7788 SCOTT
7876 ADAMS
7934 MILLER
```

```
14 rows selected.
```

```
SQL> INSERT INTO Emp(Ename, Deptno, Sal, Job)
 2 VALUES('SAMPLE', 30, 2000, 'CLERK');
```

```
1 row created.
```

Spool Generated For Class of Oracle By Satish K Yellanki

```
SQL> SELECT Empno, Ename FROM Emp;
```

```
EMPNO ENAME
-----
7839 KING
7698 BLAKE
7782 CLARK
7566 JONES
7654 MARTIN
7499 ALLEN
7844 TURNER
7900 JAMES
7521 WARD
7902 FORD
7369 SMITH
```

```
EMPNO ENAME
-----
7788 SCOTT
7876 ADAMS
7934 MILLER
7935 SAMPLE
```

15 rows selected.

```
SQL> cl scr
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE SEQUENCE Sample_ID_Key;
```

Sequence created.

```
SQL> CREATE TABLE Sample_Tab
2  (
3  SampID VARCHAR2(6),
4  SampName VARCHAR2(10),
5  SampDate DATE
6  );
```

Table created.

```
SQL> CREATE OR REPLACE TRIGGER
2  Sample_ID_Key_Gen
3  BEFORE INSERT ON Sample_Tab
4  FOR EACH ROW
5  DECLARE
6  V_SampID NUMBER;
7  BEGIN
8  SELECT Sample_ID_Key.NEXTVAL INTO V_SampID
```

Document Generated By SkyEss Techno Solutions Pvt. Ltd.
For Queries And Live Project Experience in Any Domain
Mail at: info@skyesmail.com (OR) rajesh.b@skyesmail.com
Mobile : 9030750090

Spool Generated For Class of Oracle By Satish K Yellanki

```
9 FROM DUAL;
10 :NEW.SampID:= 'SAM' ||LPAD(V_SampID, 3, 0);
11 END;
12 /
```

Trigger created.

```
SQL> INSERT INTO Sample_Tab(SampName, SampDate)
2 VALUES('SAMPLE01', SYSDATE);
```

1 row created.

```
SQL> INSERT INTO Sample_Tab(SampName, SampDate)
2 VALUES('SAMPLE01', SYSDATE);
```

1 row created.

```
SQL> SELECT * FROM Sample_Tab;
```

SAMPID	SAMPNAME	SAMPDATE
SAM001	SAMPLE01	03-SEP-10
SAM002	SAMPLE01	03-SEP-10

```
SQL> cl scr
```

```
SQL> CREATE TABLE UserLog
2 (UserId VARCHAR2(30),
3 LogDate DATE,
4 Action VARCHAR2(50)
5 )
6
```

```
SQL>
```

```
SQL> ROLLBACK;
```

Rollback complete.

```
SQL> cl scr
```

```
SQL> CREATE TABLE UserLog
2 (UserId VARCHAR2(30),
3 LogDate DATE,
4 Action VARCHAR2(50)
5 )
6 /
```

Table created.

```
SQL> CREATE OR REPLACE TRIGGER LogOnTrigger
2 AFTER LOGON ON SCHEMA
3 BEGIN
4 INSERT INTO UserLog(UserID, LogDate, Action)
5 VALUES(USER, SYSDATE, 'Logged On...');
6 END LogOnTrigger;
7 /
```

Spool Generated For Class of Oracle By Satish K Yellanki

Trigger created.

```
SQL> CREATE OR REPLACE TRIGGER LogOffTrigger
  2 BEFORE LOGOFF ON SCHEMA
  3 BEGIN
  4 INSERT INTO UserLog(UserID, LogDate, Action)
  5 VALUES(USER, SYSDATE, 'Logged Off...');
  6 END LogOffTrigger;
  7 /
```

Trigger created.

```
SQL> SELECT * FROM UserLog;
```

no rows selected

```
SQL> SPOOL OFF
```