STORED FUNCTIONS:

A FUNCTION IS BLOCK OF CODE TO PERFORM SOME TASK AND MUST RETURN A VALUE.THESE FUNCTIONS ARE CREATED BY USER EXPLICITELY.SO THAT WE CAN ALSO CALLED AS "USER DEFINED FUNCTION"

SYNTAX:

```
CREATE OR REPLACE FUNCTION <FUNCTION_NAME>
[(ARUGMENT DATATYPE,
                              ARGUMENT DATATYPE,)]
                RETURN <DATATYPE>
                    IS
 BEGIN
  <EXEC-STATEMENTS>;
  RETURN (VALUE);
 END <FUNCTION_NAME>;
 /
```

HOW TO CALL A STORED FUNCTION:

```
SELECT <FNAME>(VALUES) FROM DUAL;
```

EX: CREATE A SF TO ACCEPT EMPLOYEE NUMBER AND RETURN THAT EMPLOYEE NAME FROM EMP TABLE?

```
CREATE OR REPLACE FUNCTION SF1(P_EMPNO NUMBER)
RETURN VARCHAR2
AS
V_ENAME VARCHAR2(10);
BEGIN
SELECT ENAME INTO V_ENAME FROM EMP WHERE
EMPNO=P_EMPNO;
RETURN V_ENAME;
END;
/
```

FUNCTION CREATED.

SQL> SELECT SF1(7566) FROM DUAL;

EX: CREATE A SF TO INPUT DEPARTMENT NAME AND RETURN SUM OF SALARY OF DEPARTMENT?

```
FUNCTION SF1(P_DNAME VARCHAR2)
RETURN NUMBER
AS
V_TOTSAL NUMBER (10);
BEGIN
SELECT SUM(SAL) INTO V_TOTSAL FROM EMP E,DEPT D
WHERE E. DEPTNO=D.DEPTNO AND DNAME=P_DNAME;
RETURN V_TOTSAL;
END;
/
```

SAL> SELECT SF1('SALES') FROM DUAL;

EX: CREATE A SF TO RETURN NO. OF EMPLOYEE IN BETWEEN GIVEN DATES?

```
FUNCTION SF2(SD DATE, ED DATE)
RETURN NUMBER
AS
V_COUNT NUMBER (10);
BEGIN
SELECT COUNT (*) INTO V_COUNT FROM EMP
WHERE HIREDATE BETWEEN SD AND ED;
RETURN V_COUNT;
END;
/
```

SQL> SELECT SF2('01-JAN-81','31-DEC-81') FROM DUAL;

EX: CREATE A SF TO INPUT EMPLOYEE NUMBER AND RETURN THAT EMPLOYEE GROSS SALARY AS PER GIVEN CONDITIONS ARE

I) HRA -------- 10%

II) DA -------- 20%

III) PF --------10%.

```
FUNCTION SF3(P_EMPNO NUMBER)
RETURN NUMBER
AS
V_BSAL NUMBER (10);
V_HRA NUMBER (10);
V_DA NUMBER (10);
V_PF NUMBER (10);
V_GROSS NUMBER (10);
BEGIN
SELECT SAL INTO V_BSAL FROM EMP WHERE EMPNO=P_EMPNO;
V_HRA: =V_BSAL*0.1;
V_DA: =V_BSAL*0.2;
V_PF: =V_BSAL*0.1;
V_GROSS: =V_BSAL+V_HRA+V_DA+V_PF;
RETURN V_GROSS;
END;
/
```

SQL> SELECT SF3(7788) FROM DUAL;

EX: WRITE A FUNCTION TO FIND SIMPLE INTEREST.

```
CREATE OR REPLACE FUNCTION SI (P NUMBER, T NUMBER, R NUMBER)
                RETURN NUMBER
                    IS
SIMPLE_INT NUMBER;
```

```
BEGIN
 SIMPLE_INT: =(P*T*R)/100;
 RETURN (SIMPLE_INT);
END SI;
/
```

> GENERALLY, FUNCTIONS ARE EXECUTED BY USING 'SELECT' STATEMENT.

```
SQL> SELECT SI (1000,2,10) FROM DUAL;
```

EX: CREATE A SF TO FIND EXPERIENCE OF GIVEN EMPLOYEE?

```
 CREATE OR REPLACE FUNCTION EMP_EXP (TEMPNO
EMP.EMPNO%TYPE)
                RETURN VARCHAR2
                    IS
 TDATE EMP.HIREDATE%TYPE;
 TEXP NUMBER;
 BEGIN
  SELECT HIREDATE INTO TDATE FROM EMP
                     WHERE EMPNO=TEMPNO;
  TEXP: =ROUND((SYSDATE-TDATE)/365);
  RETURN (TEMPNO||' EMPLOYEE EXPERIENCE IS '||TEXP||'
YEARS.');
 EXCEPTION
  WHEN NO_DATA_FOUND THEN
   RETURN ('GIVEN EMPLOYEE RECORD NOT FOUND.');
 END EMP_EXP;
```

```
SQL> SELECT EMP_EXP (7788) FROM DUAL;
SQL> SELECT EMP_EXP(EMPNO) FROM EMP;
```

FUNCTION FOR TO CALCULATE EMPLOYEE EXPERIENCE:

```
CREATE OR REPLACE FUNCTION EMP_EXPE (TEMPNO
EMP.EMPNO%TYPE)
                    RETURN NUMBER
                        IS
 TEXP NUMBER;
 BEGIN
  SELECT ROUND((SYSDATE-HIREDATE)/365) INTO TEXP FROM
EMP
                                    WHERE EMPNO=TEMPNO;
  RETURN(TEXP);
 END EMP_EXPE;
```

NOTE:

ALL FUNCTIONS ARE STORED IN USER_OBJECTS.

ALL FUNCTIONS BODIES ARE STORED IN 'USER_SOURCE' SYSTEM TABLE.

> TO SEE THE FUNCTION BODY.

EX:

```
SQL> SELECT TEXT FROM USER_SOURCE WHERE
NAME='EMP_EXPE';
```

DROPPING FUNCTIONS:

SYNTAX:

```
SQL> DROP FUNCTION <FUNCTION_NAME>;
```

EX:

```
SQL> DROP FUNCTION EMP_EXPE;
```