Ex: No: 01 **Creation of a database and writing SQL queries to**

Date:20.02.24 **retrieve information from the database.**

AIM:

To create a database and write SQL queries to retrieve information from the database.

PROCEDURE:

SQL>Create table EMP (empno number, ename varchar, job varchar, mgr varchar, hiredate date, sal number, comm number deptno number);

Table created.

SQL>Insert into EMP values(7369,'BEN','CLERK' , 7002,17-DEC-80 , 1000 ,20);
SQL>Insert into EMP values(7499 'ADAMS' ,'SALESMAN', 7600, 20-FEB-81, 1200, 300, 30);
SQL>Insert into EMP values(7521, 'WILLIAMS', 'SALESMAN', 7500, 22-FEB-81, 900, 500, 30);
SQL>Insert into EMP values(7566, 'JOHN', 'MANAGER' , 7800, 02-APR-81, 2000, NULL, 20);
SQL>Insert into EMP values(7654, 'MARK', 'SALESMAN' , 7402, 28-SEP-81, 1800, 1400, 30);
SQL>Insert into EMP values(7698, 'KALIX' , 'MANAGER', 7819, 01-MAY-81, 2200,NULL , 30);
SQL>Insert into EMP values(7782, 'CELINE', 'MANAGER', 7222, 09-JUN-81, 2420,NULL , 10);
SQL>Insert into EMP values(7788,'ANAND', 'ANALYST', 7566, 19-APR-87, 3000, NULL, 20);
SQL>Insert into EMP values(7839, 'KING', 'PRESIDENT', 17-NOV-81, 5200,NULL, 10);
SQL>Insert into EMP values(7844, 'HULK', 'SALESMAN', 7698, 08-SEP-81, 1650, NULL, 30);

SQL> select * from EMP;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	BEN	CLERK	7200	17-DEC-80	1000		20
7499	ADAMS	SALESMAN	7698	20-FEB-81	1200	300	30
7521	WILLIAMS	SALESMAN	7698	22-FEB-81	900	500	30

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JOHN	MANAGER	7839	02-APR-81	2000		20
7654	MARK	SALESMAN	7402	28-SEP-81	1250	1800	30
7698	KALIX	MANAGER	7819	01-MAY-81	2000		30

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	CELINE	MANAGER	7222	09-JUN-81	2420		10
7788	ANAND	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5200		10

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM

```
   DEPTNO
----------
      7844 HULK     SALESMAN        7698 08-SEP-81       1650
        30


10 rows selected.
```

RESULT:

Thus to create a database and writing SQL queries to retrieve information from the database is verified successful

Ex: No: 02 **Performing Insertion, Deletion, Modifying,**
Date:20.02.24 **Altering, Updating and Viewing records based on condition**

AIM:

To performing Insertion, Deletion, Modifying, Altering, Updating and Viewing records based on conditions.

PROCEDURE

SQL>Create table EMP (empno number, ename varchar, job varchar, mgr varchar, hiredate date, sal number, comm number deptno number);

Table created.

SQL>Insert into EMP values(7369,'HARI','CLERK' , 7902 ,17-DEC-80 , 800 ,20);
SQL>Insert into EMP values(7499 'JOVIN' ,'SALESMAN', 7698, 20-FEB-81, 1600, 300, 30);
SQL>Insert into EMP values(7521, 'MEGHA', 'SALESMAN', 7698, 22-FEB-81, 1250, 500, 30);
SQL>Insert into EMP values(7566, 'VISA', 'MANAGER' , 7839, 02-APR-81, 2975, NULL, 20);
SQL>Insert into EMP values(7654, 'MARK', 'SALESMAN' , 7698, 28-SEP-81, 1250, 1400, 30);
SQL>Insert into EMP values(7698, 'PRIYA' , 'MANAGER', 7839, 01-MAY-81, 2850,NULL , 30);
SQL>Insert into EMP values(7782, 'BLESS', 'MANAGER', 7839, 09-JUN-81, 2450,NULL , 10);
SQL>Insert into EMP values(7788,'SCOTT', 'ANALYST', 7566, 19-APR-87, 3000, NULL, 20);
SQL>Insert into EMP values(7839, 'KINDEL', 'PRESIDENT', 17-NOV-81, 5000,NULL, 10);
SQL>Insert into EMP values(7844, 'JIMMY', 'SALESMAN', 7698, 08-SEP-81, 1500, NULL, 30);

Select * from emp;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	HARI	CLERK	7902	17-DEC-80	800		20
7499	JOVIN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	MEGHA	SALESMAN	7698	22-FEB-81	1250	500	30

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	VISA	MANAGER	7839	02-APR-81	2975		20
7654	MARK	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	PRIYA	MANAGER	7839	01-MAY-81	2850		30

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	BLESS	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KINDEL	PRESIDENT		17-NOV-81	5000		10

```
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7844 JIMMY     SALESMAN        7698 08-SEP-81       1500          0
        30

 10 rows selected.

SQL> UPDATE EMP
  2  SET SAL = SAL + 200;

11 rows updated.

SQL> UPDATE EMP
  2  SET SAL = 3000
  3  WHERE EMPNO = 7566;

1 row updated.

SQL> DELETE FROM EMP
  2  WHERE ENAME = 'Jimmy';

0 rows deleted.

SQL> SELECT * FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7369 HARI      CLERK           7902 17-DEC-80       1000
        20

      7499 JOVIN     SALESMAN        7698 20-FEB-81       1800        300
        30

      7521 MEGHA     SALESMAN        7698 22-FEB-81       1450        500
        30
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	VISA	MANAGER	7839	02-APR-81	3000		20
7654	MARK	SALESMAN	7698	28-SEP-81	1450	1400	30
7698	PRIYA	MANAGER	7839	01-MAY-81	3050		30

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	BLESSY	MANAGER	7839	09-JUN-81	2650		10
7788	SCOTT	ANALYST	7566	19-APR-87	3200		20
7839	KINDEL	PRESIDENT		17-NOV-81	5200		10

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7844	JIMMY	SALESMAN	7698	08-SEP-81	1700	0	30
7934	ABI	CLERK	7782	23-JAN-82	1500		10

11 rows selected.

```
SQL> SELECT EMPNO AS ENO, ENAME
  2  FROM EMP
  3  WHERE SAL > 3000;

       ENO ENAME
---------- ----------
      7654 MARK
      7698 PRIYA
      7788 SCOTT
      7839 KINDEL

SQL> SELECT *
  2  FROM EMP
  3  WHERE SAL > 2000 AND SAL < 3000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7782 BLESS      MANAGER         7839 09-JUN-81       2650
        10


SQL> SELECT *
  2  FROM EMP
  3  WHERE JOB <> 'MANAGER';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
----------
      7369 HARI       CLERK           7902 17-DEC-80       1000
        20

      7499 JOVIN      SALESMAN        7698 20-FEB-81       1800        300
        30

      7521 MEGHA      SALESMAN        7698 22-FEB-81       1450        500
        30


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
    DEPTNO
```

```
----------
      7654 MARK     SALESMAN        7698 28-SEP-81       1450       1400
        30

      7788 SCOTT      ANALYST         7566 19-APR-87       3200
        20

      7839 KINDEL       PRESIDENT           17-NOV-81       5200
        10


    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
      7844 JIMMY     SALESMAN        7698 08-SEP-81       1700          0
        30

      7934 ABI     CLERK           7782 23-JAN-82       1500
        10


8 rows selected.

SQL> SELECT *
  2  FROM EMP
  3  WHERE ENAME LIKE 'J%';

    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
      7499 JOVIN      SALESMAN        7698 20-FEB-81       1800        300
        30

      7844 JIMMY     SALESMAN        7698 08-SEP-81       1700          0
        30


SQL> SELECT *
  2  FROM EMP
  3  WHERE LENGTH(ENAME) = 4;

    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
```

---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO

      7369 HARI   CLERK   7902 17-DEC-80   1000
       20

      7654 MARK   SALESMAN   7698 28-SEP-81   1450   1400
      30

SQL> SELECT EMPNO AS ENO, ENAME, SAL
  2 FROM EMP
  3 WHERE DEPTNO IN (10, 20);

      ENO ENAME   SAL
---------- ---------- ----------
      7369 HARI   1000
      7566 VISA   3000
      7782 BLESS   2650
      7788 SCOTT   3200
      7839 KINDEL   5200
      7934 ABI   1500

6 rows selected.

SQL> SELECT *
  2 FROM EMP
  3 WHERE SUBSTR(ENAME, 3, 1) = 'r';

no rows selected

SQL> SELECT *
  2 FROM EMP
  3 WHERE COMM IS NULL;

    EMPNO ENAME   JOB   MGR HIREDATE   SAL   COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO

      7369 HARI   CLERK   7902 17-DEC-80   1000
       20

```
      7566 VISA      MANAGER         7839 02-APR-81       3000
        20

      7698 PRIYA     MANAGER         7839 01-MAY-81       3050
        30


    EMPNO ENAME      JOB             MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
      7782 BLESS     MANAGER         7839 09-JUN-81       2650
        10

      7788 SCOTT     ANALYST         7566 19-APR-87       3200
        20

      7839 KINDEL      PRESIDENT           17-NOV-81       5200
        10



    EMPNO ENAME      JOB             MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
      7934 ABI     CLERK           7782 23-JAN-82       1500
        10


10 rows selected.

SQL> SELECT MGR AS ENO
  2  FROM EMP
  3  WHERE JOB = 'MANAGER';

       ENO
----------
      7566
      7698
      7782
```

```
SQL> CREATE TABLE new_emp AS
  2  SELECT * FROM EMP;

Table created.

SQL> COMMIT;

Commit complete.
```

RESULT:

Thus the program to perform Insertion, Deletion, Modifying, Altering, updating and Viewing records has been successfully executed and verified.

Ex: No: 03 Creating an Employee database to set various constraints
Date:27.02.24 and Creation of Views Indexes, Save point.

AIM:

To Create an Employee database to set various constraints and Creation of Views Indexes, Save point..

PROCEDURE:

```
SQL> CREATE TABLE Parent_Student (
  2     PARENT_ID INT,
  3     STUDENT_ID INT,
  4     PARENT_NAME VARCHAR(50),
  5     MOBILE VARCHAR(15)
  6  );
```

Table created.

```
SQL> INSERT INTO Parent_Student VALUES (1011, 1430, 'RM', '8931222345');
```

1 row created.

```
SQL> INSERT INTO Parent_Student VALUES (1012, 1431, 'JIN', '8931223456');
```

1 row created.

```
SQL> INSERT INTO Parent_Student VALUES (1013, 1432, 'TATA', '8931223298');
```

1 row created.

```
SQL> INSERT INTO Parent_Student VALUES (1014, 1433, 'JHOPE', '8931223666');
```

1 row created.

```
SQL> INSERT INTO Parent_Student VALUES (1015, 1434, 'JIMIN', '8931223777');
```

1 row created.

```
SQL> INSERT INTO Parent_Student VALUES  (1016, 1435, 'V', '8931223888');
```

1 row created.

```
SQL> INSERT INTO Parent_Student VALUES (1017, 1436, JUNGKOOK, '8931223999');
```

1 row created.

```
SQL> INSERT INTO Parent_Student VALUES (1018, 1437, 'SUGA', '8931223000');

1 row created.

SQL> SELECT *FROM Parent_Student;

 PARENT_ID STUDENT_ID PARENT_NAME
---------- ---------- --------------------------------------------------
MOBILE
--------------
      1011       1430 RM
8931222345

      1012       1431 JIN
8931223456

      1013       1432 TATA
8931223298


 PARENT_ID STUDENT_ID PARENT_NAME
---------- ---------- --------------------------------------------------
MOBILE
--------------
      1014       1433 JHOPE
8931223666

      1015       1434 JIMIN
8931223777


 PARENT_ID STUDENT_ID PARENT_NAME
---------- ---------- --------------------------------------------------
MOBILE
--------------
      1016       1435 V
8931223888

      1017       1436 JUNGKOOK
8931223999

      1018       1437 SUGA
8931223000
```

8 rows selected.

```
SQL> CREATE TABLE Student_Marks (
  2     STUDENT_ID INT,
  3     NAME VARCHAR(50),
  4     MARK1 INT,
  5     MARK2 INT,
  6     MARK3 INT
  7  );
```

Table created.

```
SQL> INSERT INTO Student_Marks VALUES (1430, 'JO', 89, 87, 90);
```

1 row created.

```
SQL> INSERT INTO Student_Marks VALUES (1431, 'BALA', 43, 77, 60)
  2  ;
```

1 row created.

```
SQL> INSERT INTO Student_Marks VALUES (1432, 'HARSHAN', 49, 97, 80);
```

1 row created.

```
SQL> INSERT INTO Student_Marks VALUES (1433, 'PRIYA', 75, 82, 88);
```

1 row created.

```
SQL> INSERT INTO Student_Marks VALUES  (1434, 'SIDDHU', 65, 79, 70);
```

1 row created.

```
SQL> INSERT INTO Student_Marks VALUES  (1435, 'NAMJOON', 80, 88, 92);
```

1 row created.

```
SQL> INSERT INTO Student_Marks VALUES (1436, 'ANIL', 55, 63, 67);
```

1 row created.

```
SQL> SELECT *FROM Student_Marks;
```

STUDENT_ID NAME MARK1
---------- ---------- ----------
MARK2 MARK3
---------- ----------
1430 JO 89
87 90

1431 BALA 43
77 60

1432 HARSHAN 49
97 80

STUDENT_ID NAME MARK1
---------- ---------- ----------
MARK2 MARK3
---------- ----------
1433 PRIYA 75
82 88

1434 SIDDHU 65
79 70

1435 NAMJOON 80
88 92

STUDENT_ID NAME MARK1
---------- ---------- ----------
MARK2 MARK3
---------- ----------
1436 ANIL 55
63 67

7 rows selected.

SQL> ALTER TABLE Parent_Student ADD CONSTRAINT PK_Parent_Student

```
PRIMARY KEY(PARENT_ID);

Table altered.

SQL> ALTER TABLE Student_Marks ADD CONSTRAINT PK_Student_Marks
PRIMARY KEY(STUDENT_ID);

Table altered.

SQL> ALTER TABLE Parent_Student
  2  DROP PRIMARY KEY;

Table altered.

SQL> ALTER TABLE Parent_Student ADD CONSTRAINT PK_Parent_Student
PRIMARY KEY(STUDENT_ID);

Table altered.

SQL> ALTER TABLE Student_Marks
  2  ADD CONSTRAINT FK_Student_Marks_Parent
  3  FOREIGN KEY (STUDENT_ID)
  4  REFERENCES Parent_Student(STUDENT_ID);

Table altered.

SQL> CREATE TABLE ORDER_PROCESSING (
  2      Order_ID NUMBER(3),
  3      Product_ID VARCHAR2(10),
  4      Quantity NUMBER(3,2),
  5      Price NUMBER(4,2)
  6  );

Table created.

SQL>
SQL> -- Insert values
SQL> INSERT INTO ORDER_PROCESSING VALUES (101, 'NOODLES-22', 6.5, 30.50);

1 row created.

SQL> INSERT INTO ORDER_PROCESSING VALUES (102, 'OIL', 2.0, 90.50);

1 row created.

SQL> INSERT INTO ORDER_PROCESSING VALUES (103, 'SHOE', 2, 95);
```

```
1 row created.

SQL> INSERT INTO ORDER_PROCESSING VALUES (104, 'WATER BOTS', 2, 20);

1 row created.

SQL>
SQL> -- Savepoint
SQL> SAVEPOINT A;

Savepoint created.

SQL>
SQL> -- Insert more values
SQL> INSERT INTO ORDER_PROCESSING VALUES (105, 'EGG PLANT', 8, 40.50);

1 row created.

SQL> INSERT INTO ORDER_PROCESSING VALUES (106, 'SOAP', 1, 75.50);

1 row created.
SQL> -- Savepoint B
SQL> SAVEPOINT B;

Savepoint created.

SQL>
SQL> -- Insert values
SQL> INSERT INTO ORDER_PROCESSING VALUES (107, 'FACE WASH', 1, 45.50);

1 row created.

SQL> INSERT INTO ORDER_PROCESSING VALUES (108, 'TONER', 1, 75.50);

1 row created.

SQL>
SQL> -- Savepoint C
SQL> SAVEPOINT C;

Savepoint created.

SQL>
SQL> -- Insert values
```

```
SQL> INSERT INTO ORDER_PROCESSING VALUES (109, 'SUGAR', 2.0, 60.50);

1 row created.
```

RESULT:

Thus the SQL commands has been verified and executed successfully for creating an Employee database to set various constraints and Creation of Views Indexes, Save point

Ex: No: 04 Joins and Nested Queries.

Date:05.02.24

AIM:

To execute and verify the SQL commands for various join operations.

PROCEDURE:

```
SQL> CREATE TABLE SALGRADE (
   GRADE NUMBER(1),
   LOSAL NUMBER(4),
   HISAL NUMBER(4));

SQL> INSERT INTO SALGRADE (GRADE, LOSAL, HISAL)
VALUES
(1, 700, 1200),
(2, 1201, 1400),
(3, 1401, 2000),
(4, 2001, 3000),
(5, 3001, 9999);

SQL> CREATE TABLE EMP (
   EMPNO NUMBER(4),
   ENAME VARCHAR2(10),
   JOB VARCHAR2(9),
   MGR NUMBER(4),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2)
);

SQL> INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM,
DEPTNO)
VALUES
(7369, 'DHONI', 'CLERK', 7902, TO_DATE('17-DEC-80', 'DD-MON-RR'), 800, NULL,
20),
(7499, 'VIRAT', 'SALESMAN', 7698, TO_DATE('20-FEB-81', 'DD-MON-RR'), 1600, 300,
30),
(7521, 'ROHIT', 'SALESMAN', 7698, TO_DATE('22-FEB-81', 'DD-MON-RR'), 1250, 500,
30),
(7566, 'SAM CURRAN', 'MANAGER', 7839, TO_DATE('02-APR-81', 'DD-MON-RR'),
2975, NULL, 20),
(7654, 'MART', 'SALESMAN', 7698, TO_DATE('28-SEP-81', 'DD-MON-RR'), 1250, 1400,
30),
```

```
(7698, 'MATHEESHA', 'MANAGER', 7839, TO_DATE('01-MAY-81', 'DD-MON-RR'),
2850, NULL, 30),
(7782, 'IMRAN', 'MANAGER', 7839, TO_DATE('09-JUN-81', 'DD-MON-RR'), 2450,
NULL, 10),
(7788, 'BRAVO', 'ANALYST', 7566, TO_DATE('19-APR-87', 'DD-MON-RR'), 3000,
NULL, 20),
(7839, 'KINGSLEY', 'PRESIDENT', NULL, TO_DATE('17-NOV-81', 'DD-MON-RR'),
5000, NULL, 10),
(7844, 'DUBE', 'SALESMAN', 7698, TO_DATE('08-SEP-81', 'DD-MON-RR'), 1500, 0, 30),


SQL> SELECT *FROM SALGRADE;

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

SQL> SELECT *FROM EMP;

    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
     7369 DHONI     CLERK           7902 17-DEC-80        800
       20

     7499 VIRAT     SALESMAN        7698 20-FEB-81       1600        300
       30

     7521 ROHIT       SALESMAN        7698 22-FEB-81       1250        500
       30


    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
     7566 SAM CURRAN     MANAGER         7839 02-APR-81       2975
       20

     7654 MART     SALESMAN        7698 28-SEP-81       1250       1400
       30
```

```
      7698 MATHEESHA      MANAGER         7839 01-MAY-81       2850
        30


    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
      7782 IMRAN      MANAGER         7839 09-JUN-81       2450
        10

      7788 BRAVO      ANALYST         7566 19-APR-87       3000
        20

      7839 KINGSLEY       PRESIDENT           17-NOV-81       5000
        10


    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
---------- ---------- --------- ---------- --------- ---------- ----------
   DEPTNO
----------
      7844 DUBE     SALESMAN        7698 08-SEP-81       1500          0
        30


10 rows selected.

SQL> SELECT e.EMPNO, e.ENAME, e.JOB, e.SAL, s.GRADE, s.LOSAL, s.HISAL
  2  FROM EMP e
  3  JOIN SALGRADE s ON e.DEPTNO = s.GRADE;

no rows selected

SQL> SELECT e.EMPNO, e.ENAME, e.JOB, e.SAL, s.GRADE, s.LOSAL, s.HISAL
  2  FROM EMP e
  3  JOIN SALGRADE s ON e.SAL BETWEEN s.LOSAL AND s.HISAL;

    EMPNO ENAME      JOB              SAL      GRADE      LOSAL      HISAL
---------- ---------- --------- ---------- ---------- ---------- ----------
      7369 DHONI      CLERK            800          1        700       1200
      7876 ROHIT      CLERK           1100          1        700       1200
      7654 MART     SALESMAN         1250          2       1201       1400
      7844 DUBE     SALESMAN         1500          3       1401       2000
```

```
      7499 VIRAT      SALESMAN        1600          3       1401       2000
      7782 IMRAN      MANAGER         2450          4       2001       3000
      7698 MATHEESHA      MANAGER         2850          4       2001       3000
      7566 SAM CURRAN     MANAGER         2975          4       2001       3000

    EMPNO ENAME      JOB              SAL      GRADE      LOSAL      HISAL
---------- ---------- --------- ---------- ---------- ---------- ----------
      7788 BRAVO      ANALYST         3000          4       2001       3000
      7839 KINGSLEY       PRESIDENT       5000          5       3001       9999

10 rows selected.


SQL> SELECT e1.ENAME AS Employee_Name, e1.SAL AS Employee_Salary,
e2.GRADE, e2.LOSAL, e2.HISAL
  2  FROM EMP e1
  3  JOIN SALGRADE e2 ON e1.SAL BETWEEN e2.LOSAL AND e2.HISAL;

EMPLOYEE_N EMPLOYEE_SALARY      GRADE      LOSAL      HISAL
---------- --------------- ---------- ---------- ----------
DHONI                800          1        700       1200
ROHIT               1250          2       1201       1400
MART                1250          2       1201       1400
DUBE                1500          3       1401       2000
VIRAT               1600          3       1401       2000
IMRAN               2450          4       2001       3000
MATHEESHA           2850          4       2001       3000
SAM CURRAN 2975          4       2001       3000

EMPLOYEE_N EMPLOYEE_SALARY      GRADE      LOSAL      HISAL
---------- --------------- ---------- ---------- ----------
BRAVO               3000          4       2001       3000
KINGSLEY            5000          5       3001       9999

10 rows selected.

SQL> SELECT e.EMPNO, e.ENAME, e.JOB, e.SAL, s.GRADE, s.LOSAL, s.HISAL
  2  FROM EMP e
  3  LEFT OUTER JOIN SALGRADE s ON e.DEPTNO = s.GRADE;

    EMPNO ENAME      JOB              SAL      GRADE      LOSAL      HISAL
---------- ---------- --------- ---------- ---------- ---------- ----------
      7839 KINGSLEY       PRESIDENT       5000
      7782 IMRAN      MANAGER         2450
```

7844 DUBE SALESMAN 1500
7698 MATEESHA MANAGER 2850
7654 MART SALESMAN 1250
7521 ROHIT SALESMAN 1250
7499 VIRAT SALESMAN 1600

EMPNO	ENAME	JOB	SAL	GRADE	LOSAL	HISAL
7788	BRAVO	ANALYST	3000			
7566	SAM CURRAN	MANAGER	2975			
7369	DHONI	CLERK	800			

10 rows selected.

RESULT:

Thus the SQL commands has been verified and executed successfully for various join operations.

Ex: No: 05 Study of PL/SQL block.
Date:12.03.24

AIM:

To write a PL/SQL block using different control (if, if else, for loop, while loop,…) statements.

PROCEDURE:

Program to find factorial of a number:

```
SQL> DECLARE
  2    n NUMBER := 11; -- Change this to the number whose factorial you want to calculate
  3    factorial NUMBER := 1;
  4  BEGIN
  5    FOR i IN 1..n LOOP
  6      factorial := factorial * i;
  7    END LOOP;
  8
  9    DBMS_OUTPUT.PUT_LINE('Factorial of ' || n || ' is: ' || factorial);
 10  END;
 11  /
Factorial of 11 is: 39916800

PL/SQL procedure successfully completed.
```

Program to reverse a number:

```
SQL> DECLARE
  2    num NUMBER := 12345; -- Change this to the number you want to reverse
  3    reversed_num NUMBER;
  4  BEGIN
  5    SELECT TO_NUMBER(REVERSE(TO_CHAR(num))) INTO reversed_num FROM
dual;
  6
  7    DBMS_OUTPUT.PUT_LINE('Reversed number: ' || reversed_num);
  8  END;
  9  /
Reversed number: 54321

PL/SQL procedure successfully completed.
```

Program to generate Fibonacci series:

```
SQL> DECLARE
  2    n NUMBER := 10; -- Change this to the number of Fibonacci terms you want to
generate
  3    first_term NUMBER := 0;
  4    second_term NUMBER := 1;
  5    next_term NUMBER;
  6  BEGIN
  7    DBMS_OUTPUT.PUT_LINE('Fibonacci Series:');
  8    DBMS_OUTPUT.PUT_LINE(first_term); -- Print the first term
  9    DBMS_OUTPUT.PUT_LINE(second_term); -- Print the second term
 10
 11    FOR i IN 3..n LOOP
 12      next_term := first_term + second_term;
 13      DBMS_OUTPUT.PUT_LINE(next_term); -- Print the next term
 14      first_term := second_term;
 15      second_term := next_term;
 16    END LOOP;
 17  END;
 18  /
Fibonacci Series:
0
1
1
2
3
5
8
13
21
34

PL/SQL procedure successfully completed.
```

RESULT:

Thus the Study of PL/SQL block has been implemented by various control structures and are verified and executed successfully.

Ex: No: 06 Write a PL/SQL block to satisfy some conditions
Date:19.03.24 by accepting input from the user

AIM:

To implement the PL/SQL block to satisfy some conditions by accepting input from the user.

PROCEDURE:

PL/SQL block to calculate sum of two numbers and display the output

```
SQL> DECLARE
  2    A NUMBER(2); -- Declare A as a number with a precision of 2 (range
-99 to 99)
  3    B NUMBER(2); -- Declare B as a number with a precision of 2 (range
-99 to 99)
  4    C NUMBER(3); -- Declare C as a number with a precision of 3 (range
-999 to 999)
  5
  6  BEGIN
  7    A := 10; -- Assign value 10 to A
  8    B := 20; -- Assign value 20 to B
  9    C := A + B; -- Calculate the sum of A and B and assign it to C
 10
 11    DBMS_OUTPUT.PUT_LINE('C: ' || C); -- Output the value of C
 12    DBMS_OUTPUT.PUT_LINE('Sum of two numbers: ' || C); -- Output
the sum with a message
 13  END;
 14  /
C: 30
Sum of two numbers: 30

PL/SQL procedure successfully completed.
```

PL/SQL block TO accepts employee number and increment is salary by 1000

```
SQL> DECLARE
  2      v_empno NUMBER; -- Employee number (you can replace
this with an actual value)
  3  BEGIN
  4      -- Replace v_empno with the desired employee number
```

```
  5      v_empno := &Empno; -- Input parameter (prompted for user
input)
  6
  7      -- Update the salary for the specified employee
  8      UPDATE emp
  9      SET sal = sal + 1000
 10       WHERE empno = v_empno;
 11
 12       -- Display a message indicating the update
 13       DBMS_OUTPUT.PUT_LINE('Salary updated for employee '
|| v_empno);
 14  END;
 15  /
Enter value for empno: 12
old   5:     v_empno := &Empno; -- Input parameter (prompted for
user input)
new   5:     v_empno := 12; -- Input parameter (prompted for user
input)
Salary updated for employee 12
```

PL/SQL block to accept empno and delete that row from the emp table

```
SQL> CREATE OR REPLACE PROCEDURE DeleteEmployee(empno IN NUMBER) IS
  2  BEGIN
  3      DELETE FROM emp WHERE Empno = empno;
  4      IF SQL%ROWCOUNT > 0 THEN
  5          DBMS_OUTPUT.PUT_LINE('Deleted ' || SQL%ROWCOUNT || ' record(s) from
emp.');
  6      ELSE
  7          DBMS_OUTPUT.PUT_LINE('No records found for empno ' || empno);
  8      END IF;
  9  END;
 10  /

Procedure created.
```

RESULT:

Thus the PL/SQL block to satisfy some conditions by accepting input from the user has been verified and executed successfully.