EX NO : 01

DATE : 15.02.2024

DATA DEFINITION LANGUAGE, DATA MANIPULATION LANGUAGE COMMANDS

AIM

To define and manage the structure of the database objects using Data Definition language commands and Data Manipulation commands in Oracle database software using SQLPlus tool.

SYNTAX

i. CREATE TABLE table_name (column1 datatype(size) , column2 datatype(size) … column datatype(size));

ii. INSERT INTO table_name VALUES (values1, values2,…valuesn);

iii. ALTER TABLE tablename

   a) MODIFY column_name datatype(size);
   b) RENAME COLUMN old_column_name TO new_column_name;
   c) ADD column_name datatype(size);
   d) DROP column_name;

iv) SELECT [*][column(s)] FROM table_name;

v) DESC table_name;

vi) UPDATE table_name SET column_name = values;

vii) RENAME TABLE old_table_name TO new_table_name;

viii) TRUNCATE TABLE table_name;

OUTPUT

1.

```
SQL> CREATE TABLE PROGRAMMER_2022503305 (
  2  EMPNO NUMBER(5) PRIMARY KEY,
  3  PROJID VARCHAR2(5),
  4  LASTNAME VARCHAR2(30) NOT NULL,
  5  FIRSTNAME VARCHAR2(30) NOT NULL,
  6  HIREDATE DATE DEFAULT CURRENT_DATE,
  7  LANGUAGE VARCHAR2(15) CHECK (LANGUAGE IN ('VB','JAVA','C++','PYTHON','RUBY')),
  8  TASKNO NUMBER(2) CHECK (TASKNO >=11 AND TASKNO <= 99),
  9  PRIVILEGE VARCHAR2(25));

Table created.

SQL> DESC PROGRAMMER_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(5)
 PROJID                                             VARCHAR2(5)
 LASTNAME                                  NOT NULL VARCHAR2(30)
 FIRSTNAME                                 NOT NULL VARCHAR2(30)
 HIREDATE                                           DATE
 LANGUAGE                                           VARCHAR2(15)
 TASKNO                                             NUMBER(2)
 PRIVILEGE                                          VARCHAR2(25)
```

2.

```
SQL> INSERT INTO PROGRAMMER_2022503305 VALUES(
  2  &EMPNO,
  3  '&LASTNAME',
  4  '&FIRSTNAME',
  5  TO_DATE('&HIREDATE','dd/mm/yyyy'),
  6  '&PROJID',
  7  '&LANGUAGE',
  8  &TASKNO,
  9  '&PRIVILEGE');
Enter value for empno: 201
old   2: &EMPNO,
new   2: 201,
Enter value for lastname: GUPTA
old   3: '&LASTNAME',
new   3: 'GUPTA',
Enter value for firstname: SAURAV
old   4: '&FIRSTNAME',
new   4: 'SAURAV',
Enter value for hiredate: 01/01/1995
old   5: TO_DATE('&HIREDATE','dd/mm/yyyy'),
new   5: TO_DATE('01/01/1995','dd/mm/yyyy'),
Enter value for projid: NPR
old   6: '&PROJID',
new   6: 'NPR',
Enter value for language: VB
old   7: '&LANGUAGE',
new   7: 'VB',
Enter value for taskno: 52
old   8: &TASKNO,
new   8: 52,
Enter value for privilege: SECRET
old   9: '&PRIVILEGE')
new   9: 'SECRET')

1 row created.
```

QUESTIONS

1. Create the table with the constraint specified
2. Insert the values using static and dynamic method.
3. Insert an employee with

   EmpNo896,LastName=’Dilip’FirstName=’Kumar’HireDate=’08-JAN-1997’

   projid=’Rnc’

   Language=Python

   TaskNo=12

   Privilige=secret.

   Check the integrity constraint violation and report the same.
4. Add the column salary. Max size 10 and 4 places of decimal
5. Modify the column salary. Max size is 12 and 4 places of decimal.
6. Update the column salary based on privilege
7. Modify the column EmpNo. Change the maximum size to 10. Report if any error occurs.
8. Modify the column Privilege. Set the constraint as NOT NULL. Report if any error occurs.
9. Rename the column Language to Prog_language
10. Update the language for an employee with EmpNo 896to "Python"
11. Increase the salary of all employees by 10%
12. Set the hire date to the current date for all employees hired after 08/30/98
13. Update thePrivilegetoTopSecretfor employees hired between1/1/98 to 1/1/2000
14. Update the Language for employees whose last names containsvowelsto "Go":
15. Update theLanguagecolumn to "Swift" for rows where theLastNamecolumn has a length of exactly 5 characters
16. Drop the column EmpNo. Check the integrity constraint violation and report the same.
17. The columnsalaryin the PROGRAMMER Table is no longer needed. Delete the column.

```
SQL> /
Enter value for empno: 390
old   2: &EMPNO,
new   2: 390,
Enter value for lastname: GHOSH
old   3: '&LASTNAME',
new   3: 'GHOSH',
Enter value for firstname: PINKY
old   4: '&FIRSTNAME',
new   4: 'PINKY',
Enter value for hiredate: 01/05/1993
old   5: TO_DATE('&HIREDATE','dd/mm/yyyy'),
new   5: TO_DATE('01/05/1993','dd/mm/yyyy'),
Enter value for projid: KCW
old   6: '&PROJID',
new   6: 'KCW',
Enter value for language: JAVA
old   7: '&LANGUAGE',
new   7: 'JAVA',
Enter value for taskno: 11
old   8: &TASKNO,
new   8: 11,
Enter value for privilege: TOP SECRET
old   9: '&PRIVILEGE')
new   9: 'TOP SECRET')

1 row created.
```

` 3.

```
SQL> INSERT INTO PROGRAMMER_2022503305 VALUES(
  2  &EMPNO,
  3  '&LASTNAME',
  4  '&FIRSTNAME',
  5  TO_DATE('&HIREDATE','dd/mm/yyyy'),
  6  '&PROJID',
  7  '&LANGUAGE',
  8  &TASKNO,
  9  '&PRIVILEGE');
Enter value for empno: 896
old   2: &EMPNO,
new   2: 896,
Enter value for lastname: JHA
old   3: '&LASTNAME',
new   3: 'JHA',
Enter value for firstname: RANJIT
old   4: '&FIRSTNAME',
new   4: 'RANJIT',
Enter value for hiredate: 15/06/1997
old   5: TO_DATE('&HIREDATE','dd/mm/yyyy'),
new   5: TO_DATE('15/06/1997','dd/mm/yyyy'),
Enter value for projid: KCW
old   6: '&PROJID',
new   6: 'KCW',
Enter value for language: JAVA
old   7: '&LANGUAGE',
new   7: 'JAVA',
Enter value for taskno: 11
old   8: &TASKNO,
new   8: 11,
Enter value for privilege: TOP SECRET
old   9: '&PRIVILEGE')
new   9: 'TOP SECRET')

1 row created.
```

18. Create a table Department with Dno(charactertype), dname (character of 25) , location (character of 25)
19. Insert three rows with dno D101, D102, D103 and include other field values as your choice.
20. Add a column deptid in the programmer table as foreign key.
21. Insert the deptid value as D105 in Programmer table and validate the same. Report if any error occurs.
22. Create a duplicate table with the same structure and dataHint:create table new_tableasselect * from existing_table;
23. Delete all the entries in the table but retain the structure of the table.
24. Drop the table Department as it not needed. Report if any constraint violation occurs.
25. Drop the table programmeras it not needed. Report if any constraint violation occurs.

```
SQL> SELECT * FROM PROGRAMMER_2022503305;

    EMPNO LASTNAME                    FIRSTNAME                   HIREDATE  PROJI LANGUAGE          TASKNO PRIVILEGE
--------- ------------------------------ ------------------------------ --------- ----- --------------- ---------- -------------------------
      201 GUPTA                     SAURAV                      01-JAN-95 NPR   VB                52 SECRET
      390 GHOSH                     PINKY                       01-MAY-93 KCW   JAVA              11 TOP SECRET
      789 AGARWAL                   PRAVEEN                     31-AUG-98 Rnc   VB                11 SECRET
      134 CHAUDHURY                 SUPRIYO                     15-JUL-95 TIPPS C++               52 SECRET
      896 JHA                       RANJIT                      15-JUN-97 KCW   JAVA              11 TOP SECRET
      345 JOHN                      PETER                       15-NOV-99 TIPPS JAVA              52
      563 ANDERSON                  ANDY                        15-AUG-94 NITTS C++               89 CONFIDENTIAL

7 rows selected.
```

4.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 ADD SALARY NUMBER(10,4);

Table altered.
```

```
SQL> DESC PROGRAMMER_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(5)
 LASTNAME                                  NOT NULL VARCHAR2(30)
 FIRSTNAME                                 NOT NULL VARCHAR2(30)
 HIREDATE                                           DATE
 PROJID                                             VARCHAR2(5)
 LANGUAGE                                           VARCHAR2(15)
 TASKNO                                             NUMBER(2)
 PRIVILEGE                                          VARCHAR2(25)
 SALARY                                             NUMBER(10,4)
```

5.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 MODIFY SALARY NUMBER(12,4);

Table altered.
```

```
SQL> DESC PROGRAMMER_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(5)
 LASTNAME                                  NOT NULL VARCHAR2(30)
 FIRSTNAME                                 NOT NULL VARCHAR2(30)
 HIREDATE                                           DATE
 PROJID                                             VARCHAR2(5)
 LANGUAGE                                           VARCHAR2(15)
 TASKNO                                             NUMBER(2)
 PRIVILEGE                                          VARCHAR2(25)
 SALARY                                             NUMBER(12,4)
```

6.

```
SQL> SELECT * FROM PROGRAMMER_2022503305;

     EMPNO LASTNAME                       FIRSTNAME                    HIREDATE  PROJI PROG_LANGUAGE       TASKNO PRIVILEGE                   SALARY
---------- ------------------------------ ---------------------------- --------- ----- --------------- ---------- ------------------------- ----------
       201 GUPTA                          SAURAV                       01-JAN-95 NPR   VB                      52 SECRET                    10000.89
       390 GHOSH                          PINKY                        01-MAY-93 KCW   JAVA                    11 TOP SECRET                50000.89
       789 AGARWAL                        PRAVEEN                      31-AUG-98 Rnc   VB                      11 SECRET                    10000.89
       134 CHAUDHURY                      SUPRIYO                      15-JUL-95 TIPPS C++                     52 SECRET                    10000.89
       896 JHA                            RANJIT                       15-JUN-97 KCW   PYTHON                  11 TOP SECRET                50000.89
       345 JOHN                           PETER                        15-NOV-99 TIPPS JAVA                    52
       563 ANDERSON                       ANDY                         15-AUG-94 NITTS C++                     89 CONFIDENTIAL            90000.999
```

7.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 MODIFY EMPNO NUMBER(10);

Table altered.

SQL> DESC PROGRAMMER_2022503305;
 Name                                                                                                                                                  Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------------- -------- ----------------
---
 EMPNO                                                                                                                                                 NOT NULL NUMBER(10)
 LASTNAME                                                                                                                                              NOT NULL VARCHAR2(30)
 FIRSTNAME                                                                                                                                             NOT NULL VARCHAR2(30)
 HIREDATE                                                                                                                                                       DATE
 PROJID                                                                                                                                                         VARCHAR2(5)
 LANGUAGE                                                                                                                                                       VARCHAR2(15)
 TASKNO                                                                                                                                                         NUMBER(2)
 PRIVILEGE                                                                                                                                                      VARCHAR2(25)
 SALARY                                                                                                                                                         NUMBER(12,4)
```

8.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 MODIFY PRIVILEGE VARCHAR2(25) NOT NULL;
ALTER TABLE PROGRAMMER_2022503305 MODIFY PRIVILEGE VARCHAR2(25) NOT NULL
*
ERROR at line 1:
ORA-02296: cannot enable (CT2022503305.) - null values found
```

9.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 RENAME COLUMN LANGUAGE TO PROG_LANGUAGE;

Table altered.

SQL> DESC PROGRAMMER_2022503305;
 Name                                                                                                                                                  Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------------- -------- ----------------
---
 EMPNO                                                                                                                                                 NOT NULL NUMBER(10)
 LASTNAME                                                                                                                                              NOT NULL VARCHAR2(30)
 FIRSTNAME                                                                                                                                             NOT NULL VARCHAR2(30)
 HIREDATE                                                                                                                                                       DATE
 PROJID                                                                                                                                                         VARCHAR2(5)
 PROG_LANGUAGE                                                                                                                                                  VARCHAR2(15)
 TASKNO                                                                                                                                                         NUMBER(2)
 PRIVILEGE                                                                                                                                                      VARCHAR2(25)
 SALARY                                                                                                                                                         NUMBER(12,4)
```

10.

```
SQL> UPDATE PROGRAMMER_2022503305 SET PROG_LANGUAGE = 'PYTHON' WHERE EMPNO = 896;

1 row updated.

SQL> SELECT * FROM PROGRAMMER_2022503305;

     EMPNO LASTNAME                    FIRSTNAME                 HIREDATE  PROJI PROG_LANGUAGE       TASKNO PRIVILEGE                  SALARY
---------- ------------------------------ ------------------------------ --------- ----- --------------- ---------- ------------------------- ----------
       201 GUPTA                     SAURAV                    01-JAN-95 NPR   VB                52 SECRET                    10000.89
       390 GHOSH                     PINKY                     01-MAY-93 KCW   JAVA              11 TOP SECRET
       789 AGARWAL                   PRAVEEN                   31-AUG-98 Rnc   VB                11 SECRET                    10000.89
       134 CHAUDHURY                 SUPRIYO                   15-JUL-95 TIPPS C++               52 SECRET                    10000.89
       896 JHA                       RANJIT                    15-JUN-97 KCW   PYTHON            11 TOP SECRET
       345 JOHN                      PETER                     15-NOV-99 TIPPS JAVA              52
       563 ANDERSON                  ANDY                      15-AUG-94 NITTS C++               89 CONFIDENTIAL

7 rows selected.
```

11.

```
SQL> UPDATE PROGRAMMER_2022503305 SET SALARY = SALARY * 10/100;

7 rows updated.

SQL> SELECT * FROM PROGRAMMER_2022503305;

     EMPNO LASTNAME                    FIRSTNAME                 HIREDATE  PROJI PROG_LANGUAGE       TASKNO PRIVILEGE                  SALARY
---------- ------------------------------ ------------------------------ --------- ----- --------------- ---------- ------------------------- ----------
       201 GUPTA                     SAURAV                    01-JAN-95 NPR   VB                52 SECRET                    1000.089
       390 GHOSH                     PINKY                     01-MAY-93 KCW   JAVA              11 TOP SECRET                5000.089
       789 AGARWAL                   PRAVEEN                   31-AUG-98 Rnc   VB                11 SECRET                    1000.089
       134 CHAUDHURY                 SUPRIYO                   15-JUL-95 TIPPS C++               52 SECRET                    1000.089
       896 JHA                       RANJIT                    15-JUN-97 KCW   PYTHON            11 TOP SECRET                5000.089
       345 JOHN                      PETER                     15-NOV-99 TIPPS JAVA              52
       563 ANDERSON                  ANDY                      15-AUG-94 NITTS C++               89 CONFIDENTIAL              9000.0999

7 rows selected.
```

12.

```
SQL> UPDATE PROGRAMMER_2022503305 SET HIREDATE = CURRENT_DATE WHERE HIREDATE > DATE '1998-08-30';

2 rows updated.

SQL> SELECT * FROM PROGRAMMER_2022503305;

     EMPNO LASTNAME                    FIRSTNAME                 HIREDATE  PROJI PROG_LANGUAGE       TASKNO PRIVILEGE                  SALARY
---------- ------------------------------ ------------------------------ --------- ----- --------------- ---------- ------------------------- ----------
       201 GUPTA                     SAURAV                    01-JAN-95 NPR   VB                52 SECRET                    1000.089
       390 GHOSH                     PINKY                     01-MAY-93 KCW   JAVA              11 TOP SECRET                5000.089
       789 AGARWAL                   PRAVEEN                   13-FEB-24 Rnc   VB                11 SECRET                    1000.089
       134 CHAUDHURY                 SUPRIYO                   15-JUL-95 TIPPS C++               52 SECRET                    1000.089
       896 JHA                       RANJIT                    15-JUN-97 KCW   PYTHON            11 TOP SECRET                5000.089
       345 JOHN                      PETER                     13-FEB-24 TIPPS JAVA              52
       563 ANDERSON                  ANDY                      15-AUG-94 NITTS C++               89 CONFIDENTIAL              9000.0999

7 rows selected.
```

13.

```
SQL> INSERT INTO PROGRAMMER_2022503305 VALUES (733,'NEVAL','ELSA','29-SEP-1999','HPR','C++',54,'SECRET',1000.089);

1 row created.

SQL> UPDATE PROGRAMMER_2022503305 SET PRIVILEGE = 'TOP SECRET' WHERE HIREDATE BETWEEN DATE '1998-01-01' AND DATE '2000-01-01';

1 row updated.

SQL> SELECT * FROM PROGRAMMER_2022503305;

     EMPNO LASTNAME                       FIRSTNAME                      HIREDATE  PROJI PROG_LANGUAGE       TASKNO PRIVILEGE                    SALARY
---------- ------------------------------ ------------------------------ --------- ----- --------------- ---------- ------------------------- ----------
       201 GUPTA                          SAURAV                         01-JAN-95 NPR   VB                        52 SECRET                    1000.089
       390 GHOSH                          PINKY                          01-MAY-93 KCW   JAVA                      11 TOP SECRET                5000.089
       789 AGARWAL                        PRAVEEN                        13-FEB-24 Rnc   VB                        11 SECRET                    1000.089
       134 CHAUDHURY                      SUPRIYO                        15-JUL-95 TIPPS C++                       52 SECRET                    1000.089
       896 JHA                            RANJIT                         15-JUN-97 KCW   PYTHON                    11 TOP SECRET                5000.089
       345 JOHN                           PETER                          13-FEB-24 TIPPS JAVA                      52
       563 ANDERSON                       ANDY                           15-AUG-94 NITTS C++                       89 CONFIDENTIAL              9000.0999
       733 NEVAL                          ELSA                           29-SEP-99 HPR   C++                       54 TOP SECRET                1000.089

8 rows selected.
```

14.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 ADD LANGUAGE VARCHAR2(15) CHECK(LANGUAGE IN('VB','JAVA','C++','PYTHON','RUBY','SWIFT','GO'));

Table altered.
```

```
SQL> UPDATE PROGRAMMER_2022503305 SET LANGUAGE = 'GO' WHERE LASTNAME LIKE '%A%' OR LASTNAME LIKE '%E%' OR LASTNAME LIKE '%I%' OR LASTNAME LIKE '%O%' OR LASTNAME LIKE '%U%';

8 rows updated.

SQL> SELECT * FROM PROGRAMMER_2022503305;

     EMPNO LASTNAME                       FIRSTNAME                      HIREDATE  PROJI     TASKNO PRIVILEGE                    SALARY LANGUAGE
---------- ------------------------------ ------------------------------ --------- ----- ---------- ------------------------- ---------- ---------------
       201 GUPTA                          SAURAV                         01-JAN-95 NPR           52 SECRET                    1000.089 GO
       390 GHOSH                          PINKY                          01-MAY-93 KCW           11 TOP SECRET                5000.089 GO
       789 AGARWAL                        PRAVEEN                        13-FEB-24 Rnc           11 SECRET                    1000.089 GO
       134 CHAUDHURY                      SUPRIYO                        15-JUL-95 TIPPS         52 SECRET                    1000.089 GO
       896 JHA                            RANJIT                         15-JUN-97 KCW           11 TOP SECRET                5000.089 GO
       345 JOHN                           PETER                          13-FEB-24 TIPPS         52                                    GO
       563 ANDERSON                       ANDY                           15-AUG-94 NITTS         89 CONFIDENTIAL              9000.0999 GO
       733 NEVAL                          ELSA                           29-SEP-99 HPR           54 TOP SECRET                1000.089 GO

8 rows selected.
```

15.

```
SQL> UPDATE PROGRAMMER_2022503305 SET LANGUAGE = 'SWIFT' WHERE LASTNAME LIKE '____';

3 rows updated.

SQL> SELECT * FROM PROGRAMMER_2022503305;

     EMPNO LASTNAME                       FIRSTNAME                      HIREDATE  PROJI     TASKNO PRIVILEGE                    SALARY LANGUAGE
---------- ------------------------------ ------------------------------ --------- ----- ---------- ------------------------- ---------- ---------------
       201 GUPTA                          SAURAV                         01-JAN-95 NPR           52 SECRET                    1000.089 SWIFT
       390 GHOSH                          PINKY                          01-MAY-93 KCW           11 TOP SECRET                5000.089 SWIFT
       789 AGARWAL                        PRAVEEN                        13-FEB-24 Rnc           11 SECRET                    1000.089 GO
       134 CHAUDHURY                      SUPRIYO                        15-JUL-95 TIPPS         52 SECRET                    1000.089 GO
       896 JHA                            RANJIT                         15-JUN-97 KCW           11 TOP SECRET                5000.089 GO
       345 JOHN                           PETER                          13-FEB-24 TIPPS         52                                    GO
       563 ANDERSON                       ANDY                           15-AUG-94 NITTS         89 CONFIDENTIAL              9000.0999 GO
       733 NEVAL                          ELSA                           29-SEP-99 HPR           54 TOP SECRET                1000.089 SWIFT

8 rows selected.
```

16.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 DROP COLUMN EMPNO;

Table altered.
```

```
SQL> SELECT * FROM PROGRAMMER_2022503305;

LASTNAME                       FIRSTNAME                      HIREDATE  PROJI     TASKNO PRIVILEGE                    SALARY LANGUAGE
------------------------------ ------------------------------ --------- ----- ---------- ------------------------ ---------- ---------------
GUPTA                          SAURAV                         01-JAN-95 NPR           52 SECRET                    1000.089 SWIFT
GHOSH                          PINKY                          01-MAY-93 KCW           11 TOP SECRET                5000.089 SWIFT
AGARWAL                        PRAVEEN                        13-FEB-24 Rnc           11 SECRET                    1000.089 GO
CHAUDHURY                      SUPRIYO                        15-JUL-95 TIPPS         52 SECRET                    1000.089 GO
JHA                            RANJIT                         15-JUN-97 KCW           11 TOP SECRET                5000.089 GO
JOHN                           PETER                          13-FEB-24 TIPPS         52                                    GO
ANDERSON                       ANDY                           15-AUG-94 NITTS         89 CONFIDENTIAL              9000.0999 GO
NEVAL                          ELSA                           29-SEP-99 HPR           54 TOP SECRET                1000.089 SWIFT

8 rows selected.
```

17.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 DROP COLUMN SALARY;

Table altered.

SQL> SELECT * FROM PROGRAMMER_2022503305;

LASTNAME                       FIRSTNAME                      HIREDATE  PROJI     TASKNO PRIVILEGE                LANGUAGE
------------------------------ ------------------------------ --------- ----- ---------- ------------------------ ---------------
GUPTA                          SAURAV                         01-JAN-95 NPR           52 SECRET                  SWIFT
GHOSH                          PINKY                          01-MAY-93 KCW           11 TOP SECRET              SWIFT
AGARWAL                        PRAVEEN                        13-FEB-24 Rnc           11 SECRET                  GO
CHAUDHURY                      SUPRIYO                        15-JUL-95 TIPPS         52 SECRET                  GO
JHA                            RANJIT                         15-JUN-97 KCW           11 TOP SECRET              GO
JOHN                           PETER                          13-FEB-24 TIPPS         52                         GO
ANDERSON                       ANDY                           15-AUG-94 NITTS         89 CONFIDENTIAL            GO
NEVAL                          ELSA                           29-SEP-99 HPR           54 TOP SECRET              SWIFT

8 rows selected.
```

18.

```
SQL> CREATE TABLE DEPARTMENT_2022503305 (
  2  DNO VARCHAR2(4),
  3  DNAME VARCHAR2(25),
  4  LOCATION VARCHAR2(25));

Table created.

SQL> DESC DEPARTMENT_2022503305;
 Name                                                                                                                                                  Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------------- -------- ------------------------------------------------------------------------------------
---
 DNO                                                                                                                                                            VARCHAR2(4)
 DNAME                                                                                                                                                          VARCHAR2(25)
 LOCATION                                                                                                                                                       VARCHAR2(25)
```

19

```
SQL> INSERT INTO DEPARTMENT_2022503305 VALUES(
  2  '&DNO',
  3  '&DNAME',
  4  '&LOCATION');
Enter value for dno: D101
old   2: '&DNO',
new   2: 'D101',
Enter value for dname: TESTING
old   3: '&DNAME',
new   3: 'TESTING',
Enter value for location: HANGAR
old   4: '&LOCATION')
new   4: 'HANGAR')

1 row created.

SQL> /
Enter value for dno: D102
old   2: '&DNO',
new   2: 'D102',
Enter value for dname: DEBUGGING
old   3: '&DNAME',
new   3: 'DEBUGGING',
Enter value for location: PARK
old   4: '&LOCATION')
new   4: 'PARK')

1 row created.
```

```
SQL> /
Enter value for dno: D103
old   2: '&DNO',
new   2: 'D103',
Enter value for dname: PROGRAMMING
old   3: '&DNAME',
new   3: 'PROGRAMMING',
Enter value for location: HUT
old   4: '&LOCATION')
new   4: 'HUT')

1 row created.

SQL> SELECT * FROM DEPARTMENT_2022503305;

DNO  DNAME                     LOCATION
---- ------------------------- -------------------------
D101 TESTING                   HANGAR
D102 DEBUGGING                 PARK
D103 PROGRAMMING               HUT
```

20.

```
SQL> ALTER TABLE PROGRAMMER_2022503305 ADD DEPTID VARCHAR2(4) REFERENCES DEPARTMENT_2022503305(DNO);

Table altered.

SQL> DESC PROGRAMMER_2022503305;
 Name                                                                                                                                                  Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------------- -------- ----------------------
---
 LASTNAME                                                                                                                                              NOT NULL VARCHAR2(30)
 FIRSTNAME                                                                                                                                             NOT NULL VARCHAR2(30)
 HIREDATE                                                                                                                                                       DATE
 PROJID                                                                                                                                                         VARCHAR2(5)
 TASKNO                                                                                                                                                         NUMBER(2)
 PRIVILEGE                                                                                                                                                      VARCHAR2(25)
 LANGUAGE                                                                                                                                                       VARCHAR2(15)
 DEPTID                                                                                                                                                         VARCHAR2(4)
```

```
SQL> DESC PROGRAMMER_2022503305;
 Name                                                                                                                                                  Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------------- -------- ----------------------
---
 LASTNAME                                                                                                                                              NOT NULL VARCHAR2(30)
 FIRSTNAME                                                                                                                                             NOT NULL VARCHAR2(30)
 HIREDATE                                                                                                                                                       DATE
 PROJID                                                                                                                                                         VARCHAR2(5)
 TASKNO                                                                                                                                                         NUMBER(2)
 PRIVILEGE                                                                                                                                                      VARCHAR2(25)
 LANGUAGE                                                                                                                                                       VARCHAR2(15)
 DEPTID                                                                                                                                                         VARCHAR2(4)
```

21.

```
SQL> INSERT INTO PROGRAMMER_2022503305 VALUES(
  2  'ANIL'
  3  ,'SHARUKH',
  4  '25-JUN-1998',
  5  'YRC',
  6  52,
  7  'SECRET'
  8  ,'PYTHON',
  9  'D105');
INSERT INTO PROGRAMMER_2022503305 VALUES(
*
ERROR at line 1:
ORA-02291: integrity constraint (CT2022503305.SYS_C008421) violated - parent key not found
```

22.

```
SQL> CREATE TABLE DUPLICATE_2022503305 AS SELECT * FROM PROGRAMMER_2022503305
  2  ;

Table created.

SQL> SELECT * FROM DUPLICATE_2022503305;

LASTNAME                       FIRSTNAME                      HIREDATE  PROJI     TASKNO PRIVILEGE                 LANGUAGE        DEPT
------------------------------ ------------------------------ --------- ----- ---------- ------------------------- --------------- ----
GUPTA                          SAURAV                         01-JAN-95 NPR           52 SECRET                    SWIFT
GHOSH                          PINKY                          01-MAY-93 KCW           11 TOP SECRET                SWIFT
AGARWAL                        PRAVEEN                        13-FEB-24 Rnc           11 SECRET                    GO
CHAUDHURY                      SUPRIYO                        15-JUL-95 TIPPS         52 SECRET                    GO
JHA                            RANJIT                         15-JUN-97 KCW           11 TOP SECRET                GO
JOHN                           PETER                          13-FEB-24 TIPPS         52                           GO
ANDERSON                       ANDY                           15-AUG-94 NITTS         89 CONFIDENTIAL              GO
NEVAL                          ELSA                           29-SEP-99 HPR           54 TOP SECRET                SWIFT

8 rows selected.
```

23.

```
SQL> TRUNCATE TABLE DUPLICATE_2022503305;

Table truncated.

SQL> DESC DUPLICATE_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
---
 LASTNAME                                  NOT NULL VARCHAR2(30)
 FIRSTNAME                                 NOT NULL VARCHAR2(30)
 HIREDATE                                           DATE
 PROJID                                             VARCHAR2(5)
 TASKNO                                             NUMBER(2)
 PRIVILEGE                                          VARCHAR2(25)
 LANGUAGE                                           VARCHAR2(15)
 DEPTID                                             VARCHAR2(4)

SQL>
```

24.

```
SQL> DROP TABLE DEPARTMENT_2022503305;
DROP TABLE DEPARTMENT_2022503305
           *
ERROR at line 1:
ORA-02449: unique/primary keys in table referenced by foreign keys
```

25.

```
SQL> DROP TABLE PROGRAMMER_2022503305;

Table dropped.

SQL> DESC PROGRAMMER_2022503305;
ERROR:
ORA-04043: object PROGRAMMER_2022503305 does not exist
```

RESULT

Thus, managing the structure of the database objects using Data Definition language commands has been executed successfully.

EX NO : 02

DATE : 19.02.2024

ER DIAGRAM

AIM

To draw the ER diagram for the given scenarios.

PROCEDURE:

STEP 1: Go to draw.io website in you search engine.

STEP 2: Place the necessary entity relationship shapes such as rectangle, oval etc.

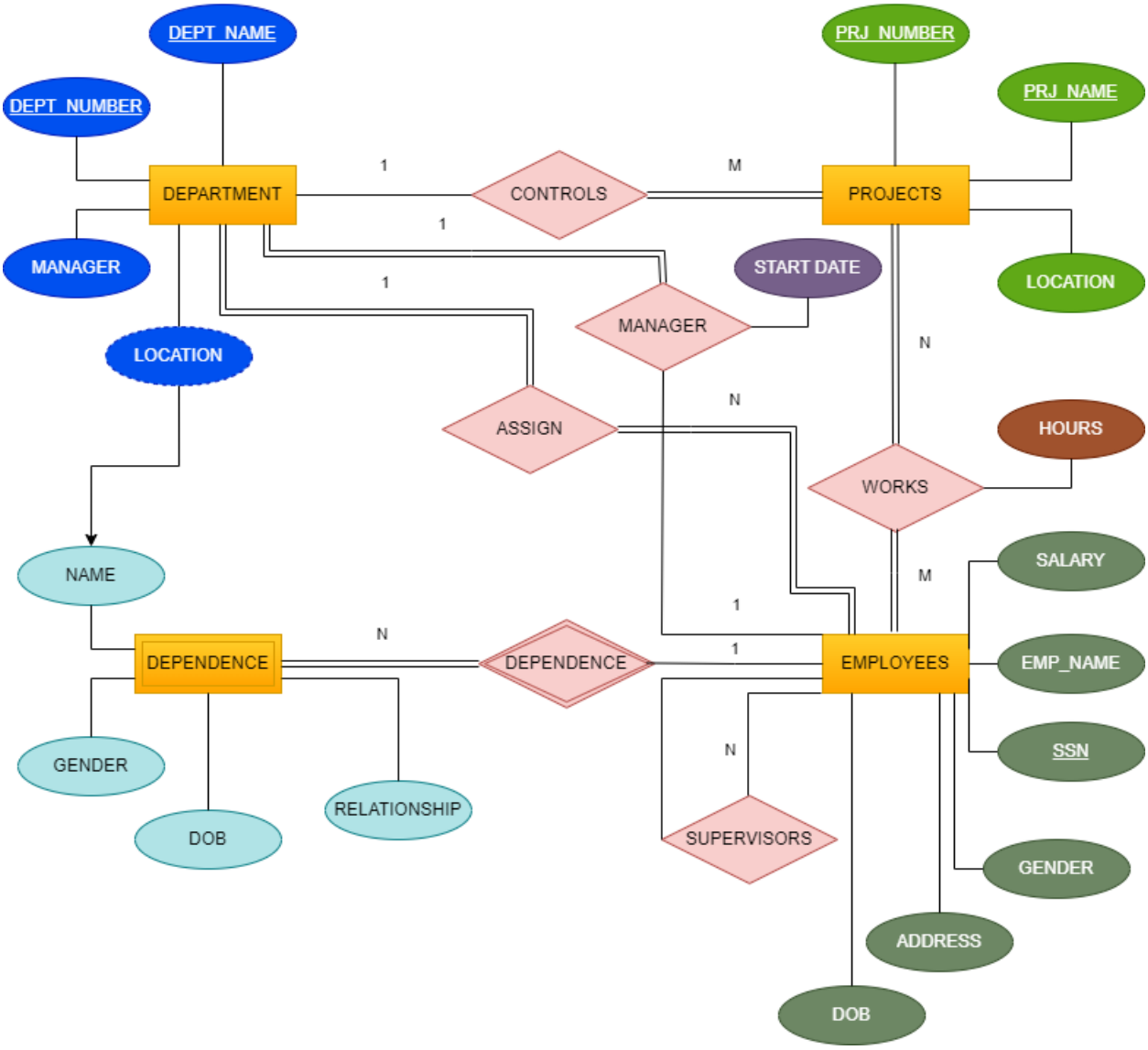
STEP 3: Name the symbols.

STEP 4: Connect the symbols using the connections in the menu bar.

STEP 5: Add some styles to the entity relationship shapes and the add some text styles.

STEP 6: Screenshot the output

2022503305

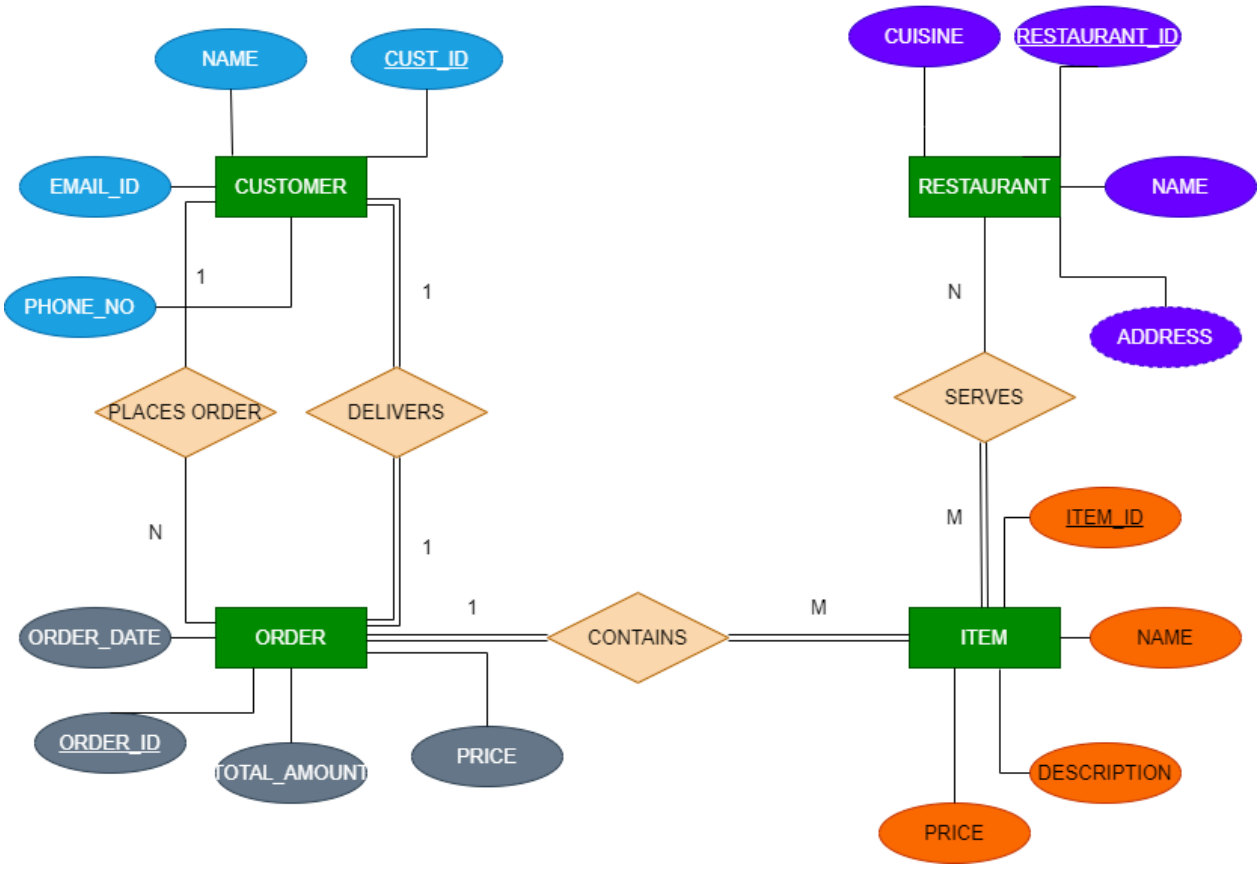

DEPT_NAME
DEPT_NUMBER
DEPARTMENT
MANAGER
LOCATION
NAME
DEPENDENCE
GENDER
DOB
RELATIONSHIP
1
CONTROLS
M
1
1
MANAGER
START DATE
ASSIGN
N
N
DEPENDENCE
1
SUPERVISORS
PRJ_NUMBER
PRJ_NAME
PROJECTS
LOCATION
N
HOURS
WORKS
M
SALARY
1
EMPLOYEES
EMP_NAME
SSN
GENDER
ADDRESS
DOB

QUESTIONS

1. The company is organized into departments. Each departmenthas a name, an unique number, and a particular employee whomanages the department. We keep track of the start date whentheemployeebeganmanagingthedept.Adeptmayhaveseveral locations. A dept controls a number of projects, each ofwhichhasaname,uniquenumber, andsinglelocation.Westoreeachemployee'sname,SSN,address,salary,sex,andDOB.

   Anemployee is assigned to one dept but may work onseveralprojects,whicharenotnecessarycontrolledbythesamedept.Wekeeptrackofthenumb erofhoursperweekthat an employee works for each project. We also keep track ofthe direct supervisor of each employee. We want to keep trackof the dependents of each employee for insurance purpose. Wekeep each dependent's first name sex, DOB, and relationship totheemployee.

2022503305


NAME
CUST_ID
EMAIL_ID
CUSTOMER
PHONE_NO
1
1
PLACES ORDER
DELIVERS
N
1
ORDER_DATE
ORDER
1
CONTAINS
M
ORDER_ID
TOTAL_AMOUNT
PRICE
CUISINE
RESTAURANT_ID
RESTAURANT
NAME
N
ADDRESS
SERVES
M
ITEM_ID
ITEM
NAME
DESCRIPTION
PRICE

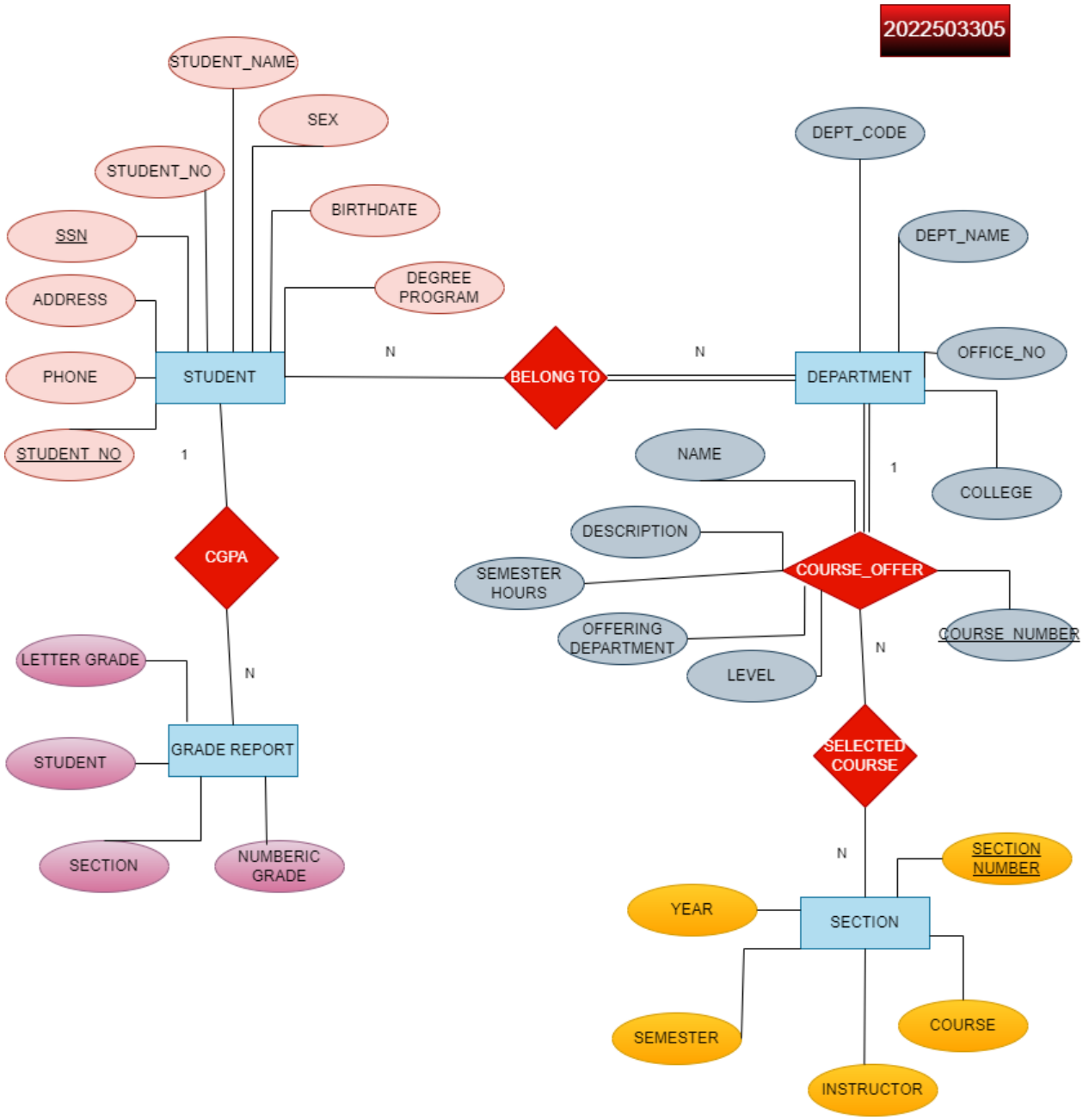
2.

a) Customerrepresents individual users who place orders on Swiggy. Each customer is defined byCustomerID,Name, Email, and Phone Number. Each customer can place multiple orders, but each order is placed by exactly one customer.

b) Restaurantrepresents the restaurants partnered with Swiggy. Each restaurant is described by theRestaurantID,Name, Address, Cuisine.Each restaurant can serve multiple food items, but each food item is served by exactly one restaurant.

c) Orderrepresents individual orders placed by customers. Each order is identified by a unique OrderID and has attributes such as OrderDateandTotalAmount.Each order can have one delivery,and each delivery corresponds to exactly one order.

d) Itemrepresents individual food items available for order. Each item is described by theItemID and has attributes such as Name, DescriptionandPrice.Each order can contain multiple items, and each item can be part of multipleorders.


2022503305
STUDENT_NAME
SEX
STUDENT_NO
BIRTHDATE
SSN
DEGREE PROGRAM
ADDRESS
PHONE
STUDENT
N
BELONG TO
N
DEPARTMENT
OFFICE_NO
STUDENT_NO
1
NAME
1
COLLEGE
DESCRIPTION
CGPA
SEMESTER HOURS
COURSE_OFFER
OFFERING DEPARTMENT
COURSE_NUMBER
LETTER GRADE
N
N
LEVEL
SELECTED COURSE
STUDENT
GRADE REPORT
N
SECTION NUMBER
YEAR
SECTION
SECTION
NUMBERIC GRADE
SEMESTER
COURSE
INSTRUCTOR

3. ConsiderthefollowingsetofrequirementsforaUNIVERSITYdatabasethatisused tokeeptrackofstudents'transcripts.
   a) The university keeps track of each student's name, studentnumber, social security number, current address and phone,permanent address and phone, birthdate, sex, class (freshman,sophomore,...,graduate),majordepartment,minordepartment (if any), and degree program (B.A., B.S., ..., Ph.D.).Some user applications need to refer to the city, state, and zipof the student's permanent address, and to the student's lastname. Both social security number and student number haveuniquevalues for each student.
   b) Each department is described by a name, department code,office number, office phone, and college. Both name and codehaveunique valuesforeach department.
   c) Each course has a course name, description, course number,number of semester hours, level, and offering department. Thevalueofcoursenumberis unique foreachcourse.
   d) Each section has an instructor, semester, year, course, andsectionnumber.Thesectionnumberdistinguishesdifferentsections of the same course that are taught during the samesemester/year; its values are 1, 2, 3, ..., up to the number ofsectionstaughtduring eachsemester.
   e) Agradereporthasastudent,section,lettergrade,andnumericgrade (0,1,2,3,4 forF,D,C,B,A,respectively).

RESULT

Thus, the ER-diagram has been drawn and the necessary styles has been given successfully.

EX NO : 03

DATE : 20.02.2024

DATA QUERY LANGUAGE

AIM

To retrieve the records in the tables from the database using the Data Query Language.

SYNTAX

i. SELECT * FROM table_name;

ii. SELECT DISTINCT column_names FROM table_name;

iii. SELECT columnname,COUNT(column_name) FROM table_name GROUP BY column_name;

iv. SELECT column_names FROM table_name;

v. SELECT column_name AS another_column_name FROM table_name;

vi. SELECT column_name FROM table_name WHERE column_name = value;

QUESTIONS

I. Write a query to retrieve all columns from the 'Programmer' table.

II. Write a query to retrieve the unique ProjId present in the 'Programmer' table

III. Write a query to retrieve total number of employees in the 'Programmer' table."

IV. Write a query to retrieve the 'LastName' and 'FirstName' columns.

V. Write a query tocalculate the annual salary for each employee in the 'Programmer'table by multiplying the 'salary' column by 12.

VI. Write a query toprovide an alias 'full_name' for the concatenation of 'FirstName' and 'LastName' columns in the 'Programmer' table."

VII. Write a query toconcatenate the 'FirstName' and 'LastName' columns with a space in between, and provide the result as 'Employee Name' from the 'Programmer' table."

VIII. Write a query to retrieve all rows from the 'Programmer' table where the 'Privilege' is 'Top Secret'."

IX. Write a query toretrieve all rows from the 'Programmer' table where the 'HireDate' falls between '1995-01-01' and '1998-12-31'.

I.

```
SQL> SELECT * FROM PROGRAMMER_2022503305;

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE               SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       201 GUPTA               SAURAV              01-JAN-95 NPR   VB                  8 SECRET                132325
       390 GHOSH               PINKY               01-JUN-93 KCW   JAVA                8 TOP SECRET           1675325
       789 AGARWAL             PRAVEEN             31-AUG-98 RNC   VB                  3 SECRET                132325
       134 CHAUDRY             SUPRIYO             15-JUL-95 TIPPS C++                 7 SECRET                132325
       896 JHA                 RANJITH             15-MAY-97 KCW   JAVA                5 TOP SECRET           1675325
       563 ANDERSON            ANDY                15-AUG-94 NITTS C++                 3 CONFIDENTIAL         1823452

6 rows selected.
```

II.

```
SQL> SELECT DISTINCT PROJ_ID FROM PROGRAMMER_2022503305;

PROJ_
-----
TIPPS
NITTS
KCW
RNC
NPR
```

III.

```
SQL> SELECT EMP_NO,COUNT(EMP_NO) FROM PROGRAMMER_2022503305 GROUP BY EMP_NO;

    EMP_NO COUNT(EMP_NO)
---------- -------------
       134             1
       201             1
       390             1
       563             1
       789             1
       896             1

6 rows selected.
```

IV.

```
SQL> SELECT FIRST_NAME, LAST_NAME FROM PROGRAMMER_2022503305;

FIRST_NAME          LAST_NAME
------------------- -------------------
SAURAV              GUPTA
PINKY               GHOSH
PRAVEEN             AGARWAL
SUPRIYO             CHAUDRY
RANJITH             JHA
ANDY                ANDERSON

6 rows selected.
```

X. Write a query to retrieve all rows from the 'Programmer' table where the 'HireDate' falls in the month of ‘August'.

XI. Write a query to retrieve all rows from the 'Programmer' table where the 'HireDate' falls in the month of ‘August', ’November’.

XII. Write a query to retrieve all rows from the 'Programmer' table where the 'HireDate' falls in the month according to the user input.

XIII. Write a query to retrieve all rows from the 'Programmer' table where the 'salary' is greater than 50000.

XIV. Write a query to retrieve all rows from the 'Programmer' table where the 'salary' is greater than 50000 and the 'ProjId' is not ‘TIPPS’.

XV. Write a query to retrieve all rows from the 'Programmer' table where the 'Language' is either 'Java' or 'C++'.

XVI. Write a query to retrieve all rows from the 'Programmer' table where the 'Privilege' is NULL."

XVII. Write a query to calculate the count of distinct entries of 'ProjId' column in the 'Programmer' table.

XVIII. Write a query to retrieve details of employees from the 'Programmer' table who joined in either January, February, or March.

XIX. Retrieve the details of employees from the 'Programmer' table whose last name starts with 'S'.

V.

```
SQL> SELECT SALARY,12* SALARY AS ANNUAL_SALARY FROM PROGRAMMER_2022503305;

    SALARY ANNUAL_SALARY
---------- -------------
    132325       1587900
   1675325      20103900
    132325       1587900
    132325       1587900
   1675325      20103900
   1823452      21881424

6 rows selected.
```

VI.

```
SQL> SELECT FIRST_NAME || LAST_NAME AS FULL_NAME FROM PROGRAMMER_2022503305;

FULL_NAME
----------------------------------------
SAURAVGUPTA
PINKYGHOSH
PRAVEENAGARWAL
SUPRIYOCHAUDRY
RANJITHJHA
ANDYANDERSON

6 rows selected.
```

VII.

```
SQL> SELECT FIRST_NAME || ' ' || LAST_NAME AS EMPLOYEE_NAME FROM PROGRAMMER_2022503305;

EMPLOYEE_NAME
----------------------------------------
SAURAV GUPTA
PINKY GHOSH
PRAVEEN AGARWAL
SUPRIYO CHAUDRY
RANJITH JHA
ANDY ANDERSON

6 rows selected.
```

VIII.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE PRIVILEGE IN 'TOP SECRET';

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE              SALARY
---------- -------------------- -------------------- --------- ----- ---------- ---------- -------------------- ----------
       390 GHOSH                PINKY               01-JUN-93 KCW   JAVA                8 TOP SECRET             1675325
       896 JHA                  RANJITH             15-MAY-97 KCW   JAVA                5 TOP SECRET             1675325
```

IX.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE HIRE_DATE BETWEEN TO_DATE('01-JAN-1995','DD-MM-YYYY') AND TO_DATE('31-DEC-1998','DD-MM-YYYY');

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE               SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       201 GUPTA               SAURAV              01-JAN-95 NPR   VB                  8 SECRET                132325
       789 AGARWAL             PRAVEEN             31-AUG-98 RNC   VB                  3 SECRET                132325
       134 CHAUDRY             SUPRIYO             15-JUL-95 TIPPS C++                 7 SECRET                132325
       896 JHA                 RANJITH             15-MAY-97 KCW   JAVA                5 TOP SECRET           1675325
```

X.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'MON') = 'AUG';

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE               SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       789 AGARWAL             PRAVEEN             31-AUG-98 RNC   VB                  3 SECRET                132325
       563 ANDERSON            ANDY                15-AUG-94 NITTS C++                 3 CONFIDENTIAL         1823452
```

XI.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'MON') = 'AUG' OR TO_CHAR(HIRE_DATE,'MON') = 'NOV';

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE               SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       789 AGARWAL             PRAVEEN             31-AUG-98 RNC   VB                  3 SECRET                132325
       563 ANDERSON            ANDY                15-AUG-94 NITTS C++                 3 CONFIDENTIAL         1823452
```

XII

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'MON') = '&MM';
Enter value for mm: JAN
old   1: SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'MON') = '&MM'
new   1: SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'MON') = 'JAN'

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE               SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       201 GUPTA               SAURAV              01-JAN-95 NPR   VB                  8 SECRET                132325
```

XIII

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE SALARY >= 50000;

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE               SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       201 GUPTA               SAURAV              01-JAN-95 NPR   VB                  8 SECRET                132325
       390 GHOSH               PINKY               01-JUN-93 KCW   JAVA                8 TOP SECRET           1675325
       789 AGARWAL             PRAVEEN             31-AUG-98 RNC   VB                  3 SECRET                132325
       134 CHAUDRY             SUPRIYO             15-JUL-95 TIPPS C++                 7 SECRET                132325
       896 JHA                 RANJITH             15-MAY-97 KCW   JAVA                5 TOP SECRET           1675325
       563 ANDERSON            ANDY                15-AUG-94 NITTS C++                 3 CONFIDENTIAL         1823452

6 rows selected.
```

XIV.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE SALARY >= 50000 AND PROJ_ID NOT LIKE 'TIPPS';

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE              SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       201 GUPTA               SAURAV              01-JAN-95 NPR   VB                  8 SECRET                132325
       390 GHOSH               PINKY               01-JUN-93 KCW   JAVA                8 TOP SECRET           1675325
       789 AGARWAL             PRAVEEN             31-AUG-98 RNC   VB                  3 SECRET                132325
       896 JHA                 RANJITH             15-MAY-97 KCW   JAVA                5 TOP SECRET           1675325
       563 ANDERSON            ANDY                15-AUG-94 NITTS C++                 3 CONFIDENTIAL          1823452
```

XV.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE LANGUAGE IN ('JAVA','C++');

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE              SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       390 GHOSH               PINKY               01-JUN-93 KCW   JAVA                8 TOP SECRET           1675325
       134 CHAUDRY             SUPRIYO             15-JUL-95 TIPPS C++                 7 SECRET                132325
       896 JHA                 RANJITH             15-MAY-97 KCW   JAVA                5 TOP SECRET           1675325
       563 ANDERSON            ANDY                15-AUG-94 NITTS C++                 3 CONFIDENTIAL          1823452
```

XVI - I

```
SQL> INSERT INTO PROGRAMMER_2022503305(EMP_NO,LAST_NAME, FIRST_NAME, HIRE_DATE,PROJ_ID,LANGUAGE,TASK_NO, SALARY) VALUES
  2  (
  3  876,
  4  'HARYA',
  5  'GRANDE',
  6  '12-MAY-2003',
  7  'KCW',
  8  'PYTHON',
  9  5,
 10  145631);

1 row created.
```

XVI - II.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE PRIVILEGE IS NULL;

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE              SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       876 HARYA               GRANDE              12-MAY-03 KCW   PYTHON              5                       145631
```

XVII.

```
SQL> SELECT DISTINCT PROJ_ID,COUNT(*) FROM PROGRAMMER_2022503305 GROUP BY PROJ_ID;

PROJ_   COUNT(*)
----- ----------
TIPPS          1
NITTS          1
KCW            3
RNC            1
NPR            1
```

XVIII.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'MON') = 'JAN' OR TO_CHAR(HIRE_DATE,'MON') = 'FEB' OR TO_CHAR(HIRE_DATE,'MON') = 'MAR';

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE              SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       201 GUPTA               SAURAV              01-JAN-95 NPR   VB                  8 SECRET                132325
```

XIX.

```
SQL> INSERT INTO PROGRAMMER_2022503305 VALUES (126,'SAN','SAIYER','15-APR-2000','TCS','RUBY',1,'TOP SECRET',1675325);

1 row created.

SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE LAST_NAME LIKE 'S%';

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE      TASK_NO PRIVILEGE                SALARY
---------- ------------------- ------------------- --------- ----- ---------- ---------- -------------------- ----------
       126 SAN                 SAIYER              15-APR-00 TCS   RUBY                1 TOP SECRET             1675325
```

RESULT

Thus, retrieving the records in the tables from the database using the Data Query Language has been executed successfully.

EX NO : 04

DATE : 28.02.2024

ER - MAPPING

AIM

To map the ER – Relation for the given ER – Diagram.

PROCEDURE

STEP 1: Explore the given ER- Relational diagram.

STEP 2: Find the tables in the diagram.

STEP 3: Create the tables with required constraints.

STEP 4: Use keys such as UNIQUE KEY, FOREIGN KEY, PRIMARY KEY.

STEP 5: Screenshot the outputs.

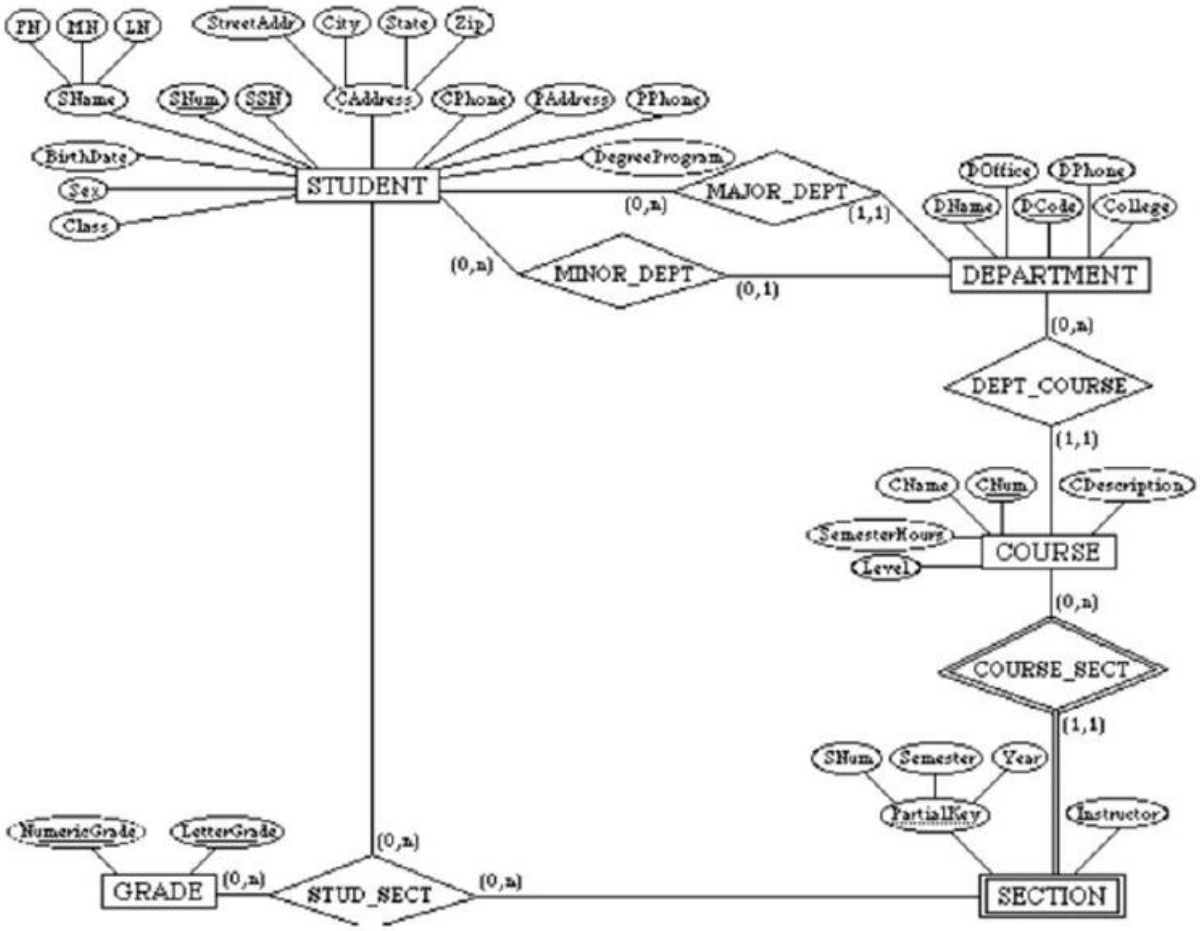
1.

CREATING STUDENT TABLE

```
SQL> CREATE TABLE STUDENT_2022503305(
  2  S_NO NUMBER(5) PRIMARY KEY,
  3  F_NAME VARCHAR2(20) NOT NULL,
  4  M_NAME VARCHAR2(20) NOT NULL,
  5  L_NAME VARCHAR2(20) NOT NULL,
  6  SSN NUMBER(5) NOT NULL UNIQUE,
  7  CLASS VARCHAR2(10) NOT NULL,
  8  SEX VARCHAR2(1) NOT NULL CHECK (SEX IN('F', 'M', 'O')),
  9  DOB DATE,
 10  STREET_ADD VARCHAR2(10),
 11  CITY VARCHAR2(20),
 12  STATE VARCHAR2(20),
 13  ZIP NUMBER(15),
 14  P_ADDRESS VARCHAR2(50) NOT NULL,
 15  C_PHONE NUMBER(10) NOT NULL,
 16  P_PHONE NUMBER(10) NOT NULL,
 17  DEGREE_PROGRAM VARCHAR2(20)
 18  );

Table created.

SQL> DESC STUDENT_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 S_NO                                      NOT NULL NUMBER(5)
 F_NAME                                    NOT NULL VARCHAR2(20)
 M_NAME                                    NOT NULL VARCHAR2(20)
 L_NAME                                    NOT NULL VARCHAR2(20)
 SSN                                       NOT NULL NUMBER(5)
 CLASS                                     NOT NULL VARCHAR2(10)
 SEX                                       NOT NULL VARCHAR2(1)
 DOB                                                DATE
 STREET_ADD                                         VARCHAR2(10)
 CITY                                               VARCHAR2(20)
 STATE                                              VARCHAR2(20)
 ZIP                                                NUMBER(15)
 P_ADDRESS                                 NOT NULL VARCHAR2(50)
 C_PHONE                                   NOT NULL NUMBER(10)
 P_PHONE                                   NOT NULL NUMBER(10)
 DEGREE_PROGRAM                                     VARCHAR2(20)
```

QUESTIONS

1.


FN
MN
LN
StreetAddr
City
State
Zip
SName
SNum
SSN
CAddress
CPhone
PAddress
PPhone
BirthDate
Sex
Class
STUDENT
DegreeProgram
MAJOR_DEPT
(0,n)
(1,1)
DOffice
DPhone
DName
DCode
College
(0,n)
MINOR_DEPT
(0,1)
DEPARTMENT
(0,n)
DEPT_COURSE
(1,1)
CName
CNum
CDescription
SemesterHours
Level
COURSE
(0,n)
COURSE_SECT
(1,1)
SNum
Semester
Year
PartialKey
Instructor
NumericGrade
LetterGrade
(0,n)
GRADE
(0,n)
STUD_SECT
(0,n)
SECTION

CREATING DEPARTMENT TABLE

```
SQL> CREATE TABLE DEPARTMENT_2022503305(
  2  D_CODE NUMBER(5) PRIMARY KEY,
  3  D_NAME VARCHAR2(15) UNIQUE NOT NULL,
  4  D_OFFICE VARCHAR2(20) NOT NULL,
  5
SQL> CREATE TABLE DEPARTMENT_2022503305(
  2  D_CODE NUMBER(5) PRIMARY KEY,
  3  D_NAME VARCHAR2(15) UNIQUE NOT NULL,
  4  D_OFFICE VARCHAR2(20) NOT NULL,
  5  D_PHONE NUMBER(10) NOT NULL,
  6  COLLEGE VARCHAR2(20) NOT NULL
  7  );

Table created.
```

```
SQL> DESC DEPARTMENT_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 D_CODE                                    NOT NULL NUMBER(5)
 D_NAME                                    NOT NULL VARCHAR2(15)
 D_OFFICE                                  NOT NULL VARCHAR2(20)
 D_PHONE                                   NOT NULL NUMBER(10)
 COLLEGE                                   NOT NULL VARCHAR2(20)
```

CREATING COURSE TABLE

```
SQL> CREATE TABLE COURSE_2022503305(
  2  C_NUM NUMBER(5) PRIMARY KEY,
  3  C_NAME VARCHAR2(20) NOT NULL,
  4  SEMESTER_HOURS NUMBER(2),
  5  LVL NUMBER(3),
  6  C_DESCRIPTION VARCHAR2(20)
  7  );

Table created.

SQL> DESC COURSE_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 C_NUM                                     NOT NULL NUMBER(5)
 C_NAME                                    NOT NULL VARCHAR2(20)
 SEMESTER_HOURS                                     NUMBER(2)
 LVL                                                NUMBER(3)
 C_DESCRIPTION                                      VARCHAR2(20)
```

CREATING SECTION TABLE HAVING FOREIGN KEY

```
SQL> CREATE TABLE SECTION_2022503305(
  2  S_NO NUMBER(5) REFERENCES STUDENT_2022503305(S_NO),
  3  SEMESTER NUMBER(1) CHECK(SEMESTER IN(1,2,3,4,5,6,7,8)),
  4  YEAR NUMBER(1) CHECK(YEAR IN(1,2,3,4)),
  5  INSTRUCTOR VARCHAR2(20)
  6  );

Table created.

SQL> DESC SECTION_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 S_NO                                               NUMBER(5)
 SEMESTER                                           NUMBER(1)
 YEAR                                               NUMBER(1)
 INSTRUCTOR                                         VARCHAR2(20)
```

CREATING STUDENT SECTION TABLE

```
SQL> CREATE TABLE STUDENT_SECT_2022503305(
  2  NUMBERIC GRADE NUMBER(2) NOT NULL,
  3  LETTER GR
  4
SQL> CREATE TABLE STUDENT_SECT_2022503305(
  2  NUMBERIC_GRADE NUMBER(2) NOT NULL,
  3  LETTER_GRADE VARCHAR2(5) NOT NULL
  4  );

Table created.

SQL> DESC STUDENT_SECT_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NUMBERIC_GRADE                            NOT NULL NUMBER(2)
 LETTER_GRADE                              NOT NULL VARCHAR2(5)
```

AFTER INSERTING VALUES IN THE STUDENT TABLE

```
SQL> SELECT * FROM STUDENT_2022503305;

     S_NO F_NAME               M_NAME               L_NAME                    SSN CLASS      S DOB       STREET_ADD CITY                STATE                     ZIP P_ADDRESS
          C_PHONE    P_PHONE DEGREE_PROGRAM
---------- -------------------- -------------------- -------------------- ---------- ---------- - --------- ---------- -------------------- -------------------- ---------- --------------------
--------------------------- ---------- ---------- --------------------
     1001 HARVA                N                    CHAUNDAR                  101 A          M 25-JAN-02 THILLAI ST TIRUCHIRAPALLI       TAMIL NADU              620020 THIRUCHIRAPALLI
                               987654987  987654987 B.E CSE
     1002 JOHN                 N                    PICHAI                    102 A          M 19-MAR-02 RAJA ST    COIMBATORE           TAMIL NADU              678439 COIMBATORE
                              9873467234 8376234923 B.E CSE
     2001 GNEVA                K                    KESAV                     201 B          F 09-OCT-01 THOMAS ST  KARAIKUDI            TAMIL NADU              675421 KARAIKUDI
                              8744393421 9837352384 B.E ECE
     2002 THIYA                K                    SADAV                     202 B          F 17-SEP-04 KONAR ST   TANJORE              TAMIL NADU              620031 TANJORE
                              7362836234 8372385321 B.E ECE
     3001 KALYAN               K                    THEE                      301 C          M 22-APR-02 UMAR ST    CHENNAI              TAMIL NADU             6753623 CHENNAI
                              8473625423 9872364738 B.TECH IT
     3002 YUVASHREE            M                    SANKAR                    302 C          F 29-MAY-05 QATAR ST   MADURAI]             TAMIL NADU             6473523 MADURAI
                              8436483452 9862746532 B.TECH IT
```

```
SQL> SELECT S_NO,F_NAME || ' ' || M_NAME || ' '|| L_NAME AS S_NAME, STREET_ADD || ', ' || CITY || ', '|| STATE || '- ' || ZIP AS C_ADDRESS FROM STUDENT_2022503305;

     S_NO S_NAME                                                         C_ADDRESS
---------- ---------------------------------------------------------------- ----------------------------------------------------------------------------------------------------
     1001 HARVA N CHAUNDAR                                                 THILLAI ST, TIRUCHIRAPALLI, TAMIL NADU- 620020
     1002 JOHN N  PICHAI                                                   RAJA ST, COIMBATORE, TAMIL NADU- 678439
     2001 GNEVA K KESAV                                                    THOMAS ST, KARAIKUDI, TAMIL NADU- 675421
     2002 THIYA K SADAV                                                    KONAR ST, TANJORE, TAMIL NADU- 620031
     3001 KALYAN K THEE                                                    UMAR ST, CHENNAI, TAMIL NADU- 6753623
     3002 YUVASHREE M SANKAR                                               QATAR ST, MADURAI], TAMIL NADU- 6473523
```

AFTER INSERTING VALUES IN THE DEPARTMENT TABLE

```
SQL> SELECT * FROM DEPARTMENT_2022503305;

    D_CODE D_NAME          D_OFFICE                D_PHONE COLLEGE
---------- --------------- -------------------- ---------- --------------------
      1101 CT              ADMIN BLOCK           144321762 MIT
      1102 ECE             HANGAR                144765765 MIT
      1103 IT              RAJAM                 144098987 MIT
```

AFTER INSERTING VALUES IN THE COURSE TABLE

```
SQL> SELECT * FROM COURSE_2022503305;

     C_NUM C_NAME               SEMESTER_HOURS        LVL C_DESCRIPTION
---------- -------------------- -------------- ---------- --------------------
     10001 DATA STRUCTURES                  19          3 ALGORITHMS
     10002 PYTHON                           17          3 PROGRAMMING
     10003 CIRCUIT THEORY                   20          5 ABOUT CIRCUITS
     10004 SEMICONDUCTOR                    15          3 SEMICONDUCTORS
     10005 SOFTWARE ENGG                    15          3 INFO TECH
     10006 DATABASE                         18          4 ORACLE, SQL

6 rows selected.
```

AFTER INSERTING VALUES IN THE SECTION TABLE

```
SQL> SELECT * FROM SECTION_2022503305;

      S_NO   SEMESTER       YEAR INSTRUCTOR
---------- ---------- ---------- --------------------
      1001          4          2 THIYA
      1002          4          2 RAJA
      2001          4          2 AMAR
      2002          4          2 OVIYA
      3001          4          2 IRAN
      3002          4          2 LILLY

6 rows selected.
```

AFTER INSERTING VALUES IN THE STUDENT_SECT TABLE

```
SQL> SELECT * FROM STUDENT_SECT_2022503305;

NUMBERIC_GRADE LETTE
-------------- -----
             1 D
             2 C
             3 B
             4 A
             5 O
```

VIOLATING THE INTERGRITY CONSTRAINT BY ADDING THE UNKNOWN VALUE TO THE CHILD TABLE WHICH IS NOT AVAILABLE IN THE PARENT TABLE

```
SQL> /
Enter value for s_no: 1003
old   2: &S_NO,
new   2: 1003,
Enter value for semester: 4
old   3: &SEMESTER,
new   3: 4,
Enter value for year: 2
old   4: &YEAR,
new   4: 2,
Enter value for instructor: AMAR
old   5: '&INSTRUCTOR')
new   5: 'AMAR')
INSERT INTO SECTION_2022503305 VALUES(
*
ERROR at line 1:
ORA-02291: integrity constraint (CT2022503305.SYS_C0011230) violated - parent key not found
```

2.

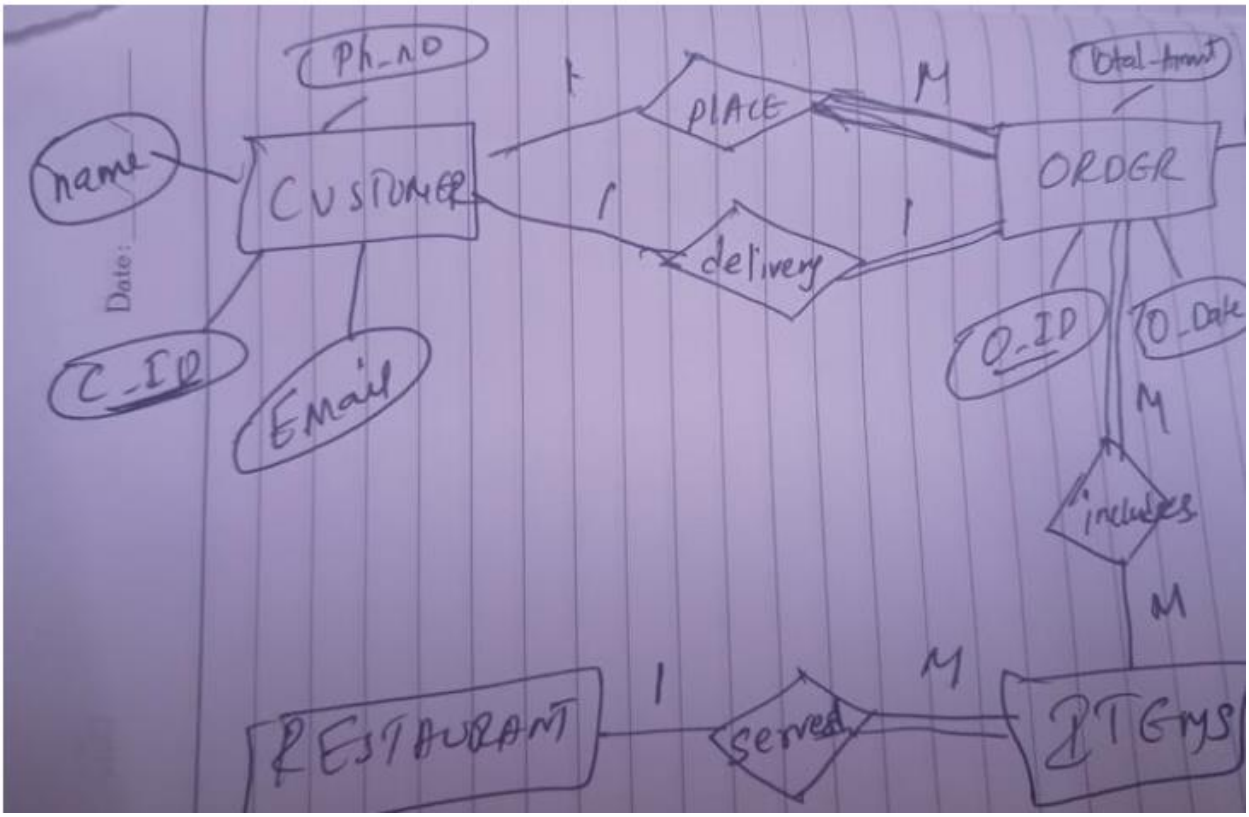
CREATING THE CUSTOMER TABLE

```
SQL> CREATE TABLE CUSTOMER_2022503305(
  2  C_ID NUMBER(5) PRIMARY KEY,
  3  NAME VARCHAR2(20) NOT NULL,
  4  EMAIL VARCHAR2(50) NOT NULL,
  5  PH_NO NUMBER(10) NOT NULL);

Table created.

SQL> DESC CUSTOMER_2022503305;
 Name                                                                                                                                                  Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------------- -------- ----------------------------
---
 C_ID                                                                                                                                                  NOT NULL NUMBER(5)
 NAME                                                                                                                                                  NOT NULL VARCHAR2(20)
 EMAIL                                                                                                                                                 NOT NULL VARCHAR2(50)
 PH_NO                                                                                                                                                 NOT NULL NUMBER(10)
```

CREATING THE ORDER TABLE

```
SQL> CREATE TABLE ORDER_2022503305(
  2  O_ID NUMBER(5) PRIMARY KEY,
  3  O_DESC VARCHAR2(30),
  4  O_DATE DATE,
  5  TOT_AMT NUMBER(10),
  6  I_ID NUMBER(5) REFERENCES ITEM_2022503305(I_ID),
  7  DP_ID NUMBER(5) REFERENCES DELIVERY_PERSON(DP_ID),
  8  C_ID NUMBER(5) REFERENCES CUSTOMER_2022503305(C_ID),
  9  R_ID NUMBER(5) REFERENCES RESTAURANT_2022503305(R_ID));

Table created.

SQL> DESC ORDER_2022503305;
 Name                                                                                                                                          Null?    Type
 ----------------------------------------------------------------------------------------------------------------------------------------------- -------- ----------------------------
---
 O_ID                                                                                                                                          NOT NULL NUMBER(5)
 O_DESC                                                                                                                                                 VARCHAR2(30)
 O_DATE                                                                                                                                                 DATE
 TOT_AMT                                                                                                                                                NUMBER(10)
 I_ID                                                                                                                                                   NUMBER(5)
 DP_ID                                                                                                                                                  NUMBER(5)
 C_ID                                                                                                                                                   NUMBER(5)
 R_ID                                                                                                                                                   NUMBER(5)
```

2.


Ph_NO
name
CUSTOMER
Date:
C_ID
Email
1
M
PLACE
Total-Amnt
ORDER
1
delivery
O_ID
O_Date
M
includes
M
RESTAURANT
1
served
M
RTGMS

CREATING THE ITEM TABLE

```
SQL> CREATE TABLE ITEM_2022503305(
  2  I_ID NUMBER(5) PRIMARY KEY,
  3  I_NAME VARCHAR2(20) NOT NULL,
  4  I_DESC VARCHAR2(50),
  5  I_RATINGS NUMBER(2));

Table created.

SQL> DESC ITEM_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
---
 I_ID                                      NOT NULL NUMBER(5)
 I_NAME                                    NOT NULL VARCHAR2(20)
 I_DESC                                             VARCHAR2(50)
 I_RATINGS                                          NUMBER(2)
```

CREATING THE DELIVERY PERSONS TABLE

```
SQL> CREATE TABLE DELIVERY_PERSON_2022503305(
  2  DP_ID NUMBER(5) PRIMARY KEY,
  3  DP_NAME VARCHAR2(20) NOT NULL,
  4  DP_PH NUMBER(10) NOT NULL UNIQUE,
  5  DP_RATINGS NUMBER(2));

Table created.

SQL> DESC DELIVERY_PERSON_2022503305;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
---
 DP_ID                                     NOT NULL NUMBER(5)
 DP_NAME                                   NOT NULL VARCHAR2(20)
 DP_PH                                     NOT NULL NUMBER(10)
 DP_RATINGS                                         NUMBER(2)
```

CREATING THE RESTAURANT TABLE

```
SQL> CREATE TABLE RESTAURANT_2022503305(
  2  R_ID NUMBER(5) PRIMARY KEY,
  3  R_NAME VARCHAR2(20) NOT NULL,
  4  R_PLACE VARCHAR2(20) NOT NULL,
  5  R_RATINGS NUMBER(2),
  6  R_ITEMS NUMBER(5) REFERENCES ITEM_2022503305(I_ID),
  7  R_PH NUMBER(10) NOT NULL);

SQL> CREATE TABLE RESTAURANT_2022503305(
  2  R_ID NUMBER(5) PRIMARY KEY,
  3  R_NAME VARCHAR2(20) NOT NULL,
  4  R_PLACE VARCHAR2(20) NOT NULL,
  5  R_RATINGS NUMBER(2),
  6  R_ITEMS NUMBER(5) REFERENCES ITEM_2022503305(I_ID),
  7  R_PH NUMBER(10) NOT NULL);
```

RESULT

Thus, ER-relational mapping for the given ER-Diagram has been done successfully.

EX NO : 05

DATE : 05.02.2024

DATA QUERY FUNCTIONS

AIM

To retrieve the records in the tables from the database using Data Query Functions.

SYNTAX

i. SELECT column_name1 || column_name2 AS name FROM table_name WHERE condition;

ii. SELECT * FROM table_name TO_CHAR(column_name, YYYY/MM/DD) = value

iii. SELECT DISTINCT column_name FROM table_name;

iv. SELECT * FROM table_name;

v. SELECT column_name FROM table_name where LENGTH(column_name) condtion;

vi. SELECT column_name FROM table_name WHERE condition;

vii. SELECT column_name ORDER BY column_name ASC/DES FETCH FIRST ROW ONLY;

viii. SELECT column_name FROM table_name WHERE column_name = TO_DATE('date') value;

ix. SELECT * FROM table_nameWHERESUBSTR(column_name,st_range,end_range) value;

x. SELECT * FROM table_name WHERE LOWER(column_name) LIKE '%value';

xi. SELECT * FROM table_name WHERE LOWER(column_name) LIKE 'value%';

xii. SELECT * FROM table_name WHERE LOWER(column_name) LIKE '_value_';

xiii. SELECT * FROM table_name WHERE LOWER(column_name) LIKE '_value';

xiv. SELECT * FROM table_name WHERE LOWER(column_name) LIKE ‘value_’;

1.

```
SQL> SELECT FIRST_NAME || ' ' || LAST_NAME AS FULL_NAME FROM PROGRAMMER_2022503305 WHERE TASK_NO > 50;

FULL_NAME
----------------------------------------
SAURAV GUPTA
PINKY GHOSH
RANJITH JHA
```

2.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'YYYY') = 1999;

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE   PRIVILEGE              SALARY    TASK_NO
---------- ------------------- ------------------- --------- ----- ---------- -------------------- ---------- ----------
       201 GUPTA               SAURAV              01-JAN-99 NPR   VB         SECRET                 132325         52
```

3.

```
SQL> SELECT DISTINCT PROJ_ID FROM PROGRAMMER_2022503305;

PROJ_
-----
TIPPS
NITTS
KCW
RNC
NPR
```

4.

```
SQL> SELECT * FROM PROGRAMMER_2022503305;

    EMP_NO LAST_NAME           FIRST_NAME          HIRE_DATE PROJ_ LANGUAGE   PRIVILEGE              SALARY    TASK_NO
---------- ------------------- ------------------- --------- ----- ---------- -------------------- ---------- ----------
       201 GUPTA               SAURAV              01-JAN-99 NPR   VB         SECRET                 132325         52
       390 GHOSH               PINKY               01-JUN-93 KCW   JAVA       TOP SECRET            1675325         65
       789 AGARWAL             PRAVEEN             31-AUG-98 RNC   VB         SECRET                 132325         45
       134 CHAUDRY             SUPRIYO             15-JUL-95 TIPPS C++        SECRET                 132325
       896 JHA                 RANJITH             15-MAY-97 KCW   JAVA       TOP SECRET            1675325         62
       563 ANDERSON            ANDY                15-AUG-94 NITTS C++        CONFIDENTIAL          1823452         45

6 rows selected.
```

5.

```
SQL> SELECT EMP_NO, FIRST_NAME, LAST_NAME, HIRE_DATE, PROJ_ID, LANGUAGE, PRIVILEGE, SALARY, TASK_NO FROM PROGRAMMER_2022503305;

    EMP_NO FIRST_NAME          LAST_NAME           HIRE_DATE PROJ_ LANGUAGE   PRIVILEGE              SALARY    TASK_NO
---------- ------------------- ------------------- --------- ----- ---------- -------------------- ---------- ----------
       201 SAURAV              GUPTA               01-JAN-99 NPR   VB         SECRET                 132325         52
       390 PINKY               GHOSH               01-JUN-93 KCW   JAVA       TOP SECRET            1675325         65
       789 PRAVEEN             AGARWAL             31-AUG-98 RNC   VB         SECRET                 132325         45
       134 SUPRIYO             CHAUDRY             15-JUL-95 TIPPS C++        SECRET                 132325
       896 RANJITH             JHA                 15-MAY-97 KCW   JAVA       TOP SECRET            1675325         62
       563 ANDY                ANDERSON            15-AUG-94 NITTS C++        CONFIDENTIAL          1823452         45

6 rows selected.
```

QUESTIONS

1. Report the employee full name who has the Task no greater than 50.

2.Report the employee details who hadhired in the year 1999.

3.Report the employee details whose ProjId is TIPPS and Programming language is Java

4.Report on unique ProjIdvalues from the PROGRAMMER table.

5.Use different way to return all columns and rows of data from the PROGRAMMER table6.Report the employee details whose length of full name is greater than 10 letters.

7.Report the employee ProjId and Task no Whose Privilege is Secret but starts at position 5.

8.Report the employee who has hired earlier than all others

9.List the employee project details whose name is "JOHN PETER" and "GHOSH PINKY".

10.Report the project details of an employee by reading their empno at run time.

11.Obtain the information about the employee who has recruited 30 days before 13/06/95.

12.Find the details of the employee whose first letter of full name and first letter of last name starts with same character.

13.Find the details of the employee whose first letter of full name and first letter of last name not starts with same character.

14.Find the details of the employee with FIRST_NAME values that do NOT begin with the letter "A" or those that do NOT comply with a COMMISSION greater than 2% percent.

15.Find the details of the employee whose lastname contains "u" letter and had thesalary greater than 5000 bearing taskno 89 or 52 or 10 whose commission calculated should not be null.

16.Report the details of the employee whose hire date is in the month that starts with 'J'(i.e January or june)

6.

```
SQL> SELECT FIRST_NAME || ' ' || LAST_NAME AS FULL_NAME FROM PROGRAMMER_2022503305 WHERE LENGTH(FIRST_NAME || ' ' || LAST_NAME) > 10;

FULL_NAME
----------------------------------------
SAURAV GUPTA
PINKY GHOSH
PRAVEEN AGARWAL
SUPRIYO CHAUDRY
RANJITH JHA
ANDY ANDERSON

6 rows selected.
```

7.

```
SQL> SELECT PROJ_ID, TASK_NO FROM PROGRAMMER_2022503305 WHERE PRIVILEGE = 'SECRET';

PROJ_ID       TASK_NO
---------- ----------
VB                 52
RNC                45
TIPPS              45
TIPPS              52
```

8.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 ORDER BY HIRE_DATE ASC FETCH FIRST 1 ROW ONLY;

    EMP_NO LAST_NAME            FIRST_NAME           HIRE_DATE PROJ_ LANGUAGE   PRIVILEGE                SALARY    TASK_NO
---------- -------------------- -------------------- --------- ----- ---------- -------------------- ---------- ----------
       390 GHOSH                PINKY                01-JUN-93 KCW   JAVA       TOP SECRET              1675325         65
```

9.

```
SQL> SELECT EMP_NO, FIRST_NAME, LAST_NAME, PROJ_ID, LANGUAGE, TASK_NO FROM PROGRAMMER_2022503305 WHERE LAST_NAME || ' ' || FIRST_NAME = 'GHOSH PINKY' OR LAST_NAME || ' ' || FIRST_NAME = 'JOHN PETER
';

   EMP_NO FIRST_NAME          LAST_NAME           PROJ_ LANGUAGE      TASK_NO
--------- ------------------- ------------------- ----- ---------- ----------
      390 PINKY               GHOSH               KCW   JAVA               65
      345 PETER               JOHN                TIPPS JAVA               52
```

10.

```
SQL> SELECT EMP_NO, FIRST_NAME, LAST_NAME, PROJ_ID, LANGUAGE, TASK_NO FROM PROGRAMMER_2022503305 WHERE EMP_NO = &EMP_NO;
Enter value for emp_no: 345
old   1: SELECT EMP_NO, FIRST_NAME, LAST_NAME, PROJ_ID, LANGUAGE, TASK_NO FROM PROGRAMMER_2022503305 WHERE EMP_NO = &EMP_NO
new   1: SELECT EMP_NO, FIRST_NAME, LAST_NAME, PROJ_ID, LANGUAGE, TASK_NO FROM PROGRAMMER_2022503305 WHERE EMP_NO = 345

    EMP_NO FIRST_NAME          LAST_NAME           PROJ_ LANGUAGE      TASK_NO
---------- ------------------- ------------------- ----- ---------- ----------
       345 PETER               JOHN                TIPPS JAVA               52
```

11.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE HIRE_DATE = TO_DATE('01-JUL-93') -30;

    EMP_NO LAST_NAME            FIRST_NAME           HIRE_DATE PROJ_ LANGUAGE   PRIVILEGE                SALARY    TASK_NO
---------- -------------------- -------------------- --------- ----- ---------- -------------------- ---------- ----------
       390 GHOSH                PINKY                01-JUN-93 KCW   JAVA       TOP SECRET              1675325         65
```

12.

```
SQL> SELECT FIRST_NAME || ' ' || LAST_NAME AS FULL_NAME FROM PROGRAMMER_2022503305 WHERE SUBSTR(LAST_NAME,1,1) = SUBSTR(FIRST_NAME,1,1);

FULL_NAME
----------------------------------------
ANDY ANDERSON
```

17. Query the ProjId from Programmer table.i.e ., The Project Idfor the <FirstName><LastName> ‘s projectis: <ProjId >, where the ProjId being concatenated to the statement so as to create the single expression aliased as ProjectDescription. The literal "'s projectis:" is then concatenated with this text, and the ProjIdfield is concatenated after that.

    Example:Project Description“The projectId for the Saurav Gupta’s Projectis: NPR.”

18. Find the the number of days for which staff were employed in a job.Create an alias for the expression column in your query using Days Employed.
19. Write an SQL query to determine the number of years each staff member has been employed in a job. Display the employee number (EmpNo), project ID (ProjId), hire date (HireDate), and create an alias for the expression column representing the number of years employed using the name 'Years Employed'. Consider that there are 365 days in a year.
20. Display the salary of each employee in the following format: '$9.999.000'.
21. Create an SQL query to show the hire date of employees, with the month expressed in full name format.
22. Write an SQL query to display employee details, showing the hire date in the 'yyyy' format.

13.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE SUBSTR(FIRST_NAME,1,1) != SUBSTR(LAST_NAME,1,1);

    EMP_NO LAST_NAME            FIRST_NAME           HIRE_DATE PROJ_ID    LANGUAGE      TASK_NO PRIVILEGE                    SALARY
---------- -------------------- -------------------- --------- ---------- ---------- ---------- ------------------------- ----------
       201 GUPTA                SAURAV               01-JAN-99 VB         VB                 52 SECRET                    132325
       390 GHOSH                PINKY                01-JUN-93 KCW        JAVA               65 TOP SECRET               1675325
       789 AGARWAL              PRAVEEN              31-AUG-98 RNC        VB                 45 SECRET                    132325
       134 CHAUDRY              SUPRIYO              15-JUN-95 TIPPS      C++                45 SECRET                    132325
       896 JHA                  RANJITH              15-MAY-97 KCW        JAVA               62 TOP SECRET               1675325
       345 JOHN                 PETER                15-NOV-98 TIPPS      JAVA               52 SECRET                   1525285

6 rows selected.
```

14.

```
SQL> SELECT * FROM PROGRAMMER_2022503305;

    EMP_NO LAST_NAME            FIRST_NAME           HIRE_DATE PROJ_ID    LANGUAGE      TASK_NO PRIVILEGE                    SALARY  COMISSION
---------- -------------------- -------------------- --------- ---------- ---------- ---------- ------------------------- ---------- ----------
       201 GUPTA                SAURAV               01-JAN-99 VB         VB                 52 SECRET                    132325          2
       390 GHOSH                PINKY                01-JUN-93 KCW        JAVA               65 TOP SECRET               1675325          6
       789 AGARWAL              PRAVEEN              31-AUG-98 RNC        VB                 45 SECRET                    132325          2
       134 CHAUDRY              SUPRIYO              15-JUN-95 TIPPS      C++                45 SECRET                    132325          2
       896 JHA                  RANJITH              15-MAY-97 KCW        JAVA               62 TOP SECRET               1675325          6
       563 ANDERSON             ANDY                 15-AUG-94 NITTS      C++                45 CONFINDENTIAL            1823452
       345 JOHN                 PETER                15-NOV-98 TIPPS      JAVA               52 SECRET                   1525285          2

7 rows selected.
```

15.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE LOWER(LAST_NAME) LIKE '%u%' AND SALARY>5000 AND (TASK_NO = 89 OR TASK_NO = 52 OR TASK_NO = 10) AND COMISSION IS NOT NULL;

    EMP_NO LAST_NAME            FIRST_NAME           HIRE_DATE PROJ_ID    LANGUAGE      TASK_NO PRIVILEGE                    SALARY  COMISSION
---------- -------------------- -------------------- --------- ---------- ---------- ---------- ------------------------- ---------- ----------
       201 GUPTA                SAURAV               01-JAN-99 VB         VB                 52 SECRET                    132325          2
```

16.

```
SQL> SELECT * FROM PROGRAMMER_2022503305 WHERE TO_CHAR(HIRE_DATE,'MONTH') LIKE 'J%';

    EMP_NO LAST_NAME            FIRST_NAME           HIRE_DATE PROJ_ID    LANGUAGE      TASK_NO PRIVILEGE                    SALARY  COMISSION
---------- -------------------- -------------------- --------- ---------- ---------- ---------- ------------------------- ---------- ----------
       201 GUPTA                SAURAV               01-JAN-99 VB         VB                 52 SECRET                    132325          2
       390 GHOSH                PINKY                01-JUN-93 KCW        JAVA               65 TOP SECRET               1675325          6
       134 CHAUDRY              SUPRIYO              15-JUN-95 TIPPS      C++                45 SECRET                    132325          2
```

17.

```
SQL> SELECT 'THE PROJECT ID FOR' || FIRST_NAME || ' ' || LAST_NAME || ' PROJECT IS' FROM PROGRAMMER_2022503305;

'THEPROJECTIDFOR'||FIRST_NAME||''||LAST_NAME||'PROJECTIS'
--------------------------------------------------------------------------------
THE PROJECT ID FORSAURAV GUPTA PROJECT IS
THE PROJECT ID FORPINKY GHOSH PROJECT IS
THE PROJECT ID FORPRAVEEN AGARWAL PROJECT IS
THE PROJECT ID FORSUPRIYO CHAUDRY PROJECT IS
THE PROJECT ID FORRANJITH JHA PROJECT IS
THE PROJECT ID FORANDY ANDERSON PROJECT IS
THE PROJECT ID FORPETER JOHN PROJECT IS
```

18.

```
SQL> SELECT EMP_NO, LAST_NAME, FIRST_NAME, TRUNC(SYSDATE) - TO_DATE(HIRE_DATE,'DD/MM/YY') AS "DAYS EMPLOYED" FROM PROGRAMMER_2022503305;

    EMP_NO LAST_NAME           FIRST_NAME          DAYS EMPLOYED
---------- ------------------- ------------------- -------------
       201 GUPTA               SAURAV                      -27329
       390 GHOSH               PINKY                       -25289
       789 AGARWAL             PRAVEEN                     -27206
       134 CHAUDRY             SUPRIYO                     -26033
       896 JHA                 RANJITH                     -26733
       563 ANDERSON            ANDY                        -25729
       345 JOHN                PETER                       -27282

7 rows selected.
```

19.

```
SQL> SELECT EMP_NO, PROJ_ID, HIRE_DATE, TO_CHAR(SYSDATE, 'yyyy') - (TO_CHAR(HIRE_DATE,'yyyy')) AS " YEARS EMPLOYEED" FROM PROGRAMMER_2022503305;

    EMP_NO PROJ_ID    HIRE_DATE  YEARS EMPLOYEED
---------- ---------- --------- ---------------
       201 VB         01-JAN-99              25
       390 KCW        01-JUN-93              31
       789 RNC        31-AUG-98              26
       134 TIPPS      15-JUN-95              29
       896 KCW        15-MAY-97              27
       563 NITTS      15-AUG-94              30
       345 TIPPS      15-NOV-98              26

7 rows selected.
```

20.

```
SQL> SELECT EMP_NO, FIRST_NAME, LAST_NAME, TO_CHAR(SALARY,'FM$9,999,0000') SALARY FROM PROGRAMMER_2022503305;

    EMP_NO FIRST_NAME          LAST_NAME           SALARY
---------- ------------------- ------------------- -------------
       201 SAURAV              GUPTA               $13,2325
       390 PINKY               GHOSH               $167,5325
       789 PRAVEEN             AGARWAL             $13,2325
       134 SUPRIYO             CHAUDRY             $13,2325
       896 RANJITH             JHA                 $167,5325
       563 ANDY                ANDERSON            $182,3452
       345 PETER               JOHN                $152,5285

7 rows selected.
```

21.

```
SQL>    SELECT EMP_NO, FIRST_NAME, LAST_NAME, TO_CHAR(HIRE_DATE,'FMDD-MONTH-YY') HIRE_DATE FROM PROGRAMMER_2022503305;

    EMP_NO FIRST_NAME          LAST_NAME           HIRE_DATE
---------- ------------------- ------------------- ----------------------------------------
       201 SAURAV              GUPTA               1-JANUARY-99
       390 PINKY               GHOSH               1-JUNE-93
       789 PRAVEEN             AGARWAL             31-AUGUST-98
       134 SUPRIYO             CHAUDRY             15-JUNE-95
       896 RANJITH             JHA                 15-MAY-97
       563 ANDY                ANDERSON            15-AUGUST-94
       345 PETER               JOHN                15-NOVEMBER-98

7 rows selected.
```

22.

```
SQL> SELECT EMP_NO, FIRST_NAME, LAST_NAME, TO_CHAR(HIRE_DATE,'FMDD-MONTH-YYYY') HIRE_DATE FROM PROGRAMMER_202250330

    EMP_NO FIRST_NAME          LAST_NAME           HIRE_DATE
---------- ------------------- ------------------- ----------------------------------------
       201 SAURAV              GUPTA               1-JANUARY-1999
       390 PINKY               GHOSH               1-JUNE-1993
       789 PRAVEEN             AGARWAL             31-AUGUST-1998
       134 SUPRIYO             CHAUDRY             15-JUNE-1995
       896 RANJITH             JHA                 15-MAY-1997
       563 ANDY                ANDERSON            15-AUGUST-1994
       345 PETER               JOHN                15-NOVEMBER-1998

7 rows selected.
```

RESULT

Thus, retrieving the records in the tables from the database using Data Query Functions has been done successfully.

EX NO : 06

DATE : 13.03.2024

VIEWS

AIM

To simplify the complex queries, enhance the data and provide data abstraction using the logical representation by Views.

SYNTAX

i. CREATE VIEW view_name AS SELECT column_name FROM table_name;

ii. INSERT INTO view_name VALUES(value1, value2, value3… valuen);

iii. UPDATE view_name SET column_name = value WHERE column_name condition;

iv. SELECT * FROM view_name;

v. DROP VIEW view_name;

vi. DESC view_name;

1.

```
SQL> INSERT INTO EMP_VIEW VALUES('JOYCLE','WALLACE',999887775);

1 row created.

SQL> SELECT * FROM EMP_VIEW;

F_NAME               L_NAME                         SSN
-------------------- -------------------- -----------
JOHN                 SMITH                    123456789
FRANKLIN             WONG                     333445555
ALICIA               ZELAYA                   999887777
JENNIFER             WALLACE                    9876543
JAMES                BORG                     888665555
JOYCLE               WALLACE                  999887775
```

2.

```
SQL> SELECT * FROM EMPLOYEE_2022503305;

F_NAME          MINIT           L_NAME                   SSN B_DATE    ADDRESS            SE  SUPER_SSN       D_NO     SALARY
--------------- --------------- --------------- ---------- --------- ------------------- -- ---------- ---------- ----------
JOHN            B               SMITH            123456789 09-JAN-65 731, FONDREN        M   333445555          5      30000
FRANKLIN        T               WONG             333445555 08-DEC-55 638 VOUS            M       40000          5      40000
ALICIA          J               ZELAYA           999887777 19-JAN-68 3321 CASTEL         F   888665555          4      43000
JENNIFER        S               WALLACE            9876543 20-JUN-62 975 FIRE OAK        F   333445555          5      38000
JAMES           E               BORG             888665555 10-NOV-37 450 STONE           M   888665555          1     650000
JOYCLE                          WALLACE          999887775

6 rows selected.
```

3.

```
SQL> UPDATE EMP_VIEW SET L_NAME = 'GREW' WHERE SSN = 999887775;

1 row updated.
```

i. CHANGES IN VIEW TABLE

```
SQL> SELECT * FROM EMP_VIEW;

F_NAME               L_NAME                         SSN
-------------------- -------------------- -----------
JOHN                 SMITH                    123456789
FRANKLIN             WONG                     333445555
ALICIA               ZELAYA                   999887777
JENNIFER             WALLACE                    9876543
JAMES                BORG                     888665555
JOYCLE               GREW                     999887775

6 rows selected.
```

QUESTION

1. 1.Insert , update, delete on simple views that reflects views and base table.
2. Insert, delete, update on simple views that violate the constraint such as primary key. Note: primary key of a table is a foreign key on other table.
3. Insert, update, delete on complex views that reflects views and base table
4. Drop view and perform query on base table and vice versa.
5. Drop column in base table after creation of view with the respective column and perform select query on view.
6. Rename column in base table after creation of view with the respective column and perform select query on view.
7. Rename table in base table after creation of view and perform select query on view.
8. Modify data type of a column in base table after creation of view with the respective column and perform select query on view.
9. Create a view from existing view.

ii. CHANGES IN BASE TABLE.

```
SQL> SELECT * FROM EMPLOYEE_2022503305;

F_NAME          MINIT           L_NAME                SSN B_DATE    ADDRESS            SE  SUPER_SSN       D_NO     SALARY
--------------- --------------- --------------- ---------- --------- ------------------ -- ---------- ---------- ----------
JOHN            B               SMITH           123456789 09-JAN-65 731, FONDREN       M   333445555          5      30000
FRANKLIN        T               WONG            333445555 08-DEC-55 638 VOUS           M       40000          5      40000
ALICIA          J               ZELAYA          999887777 19-JAN-68 3321 CASTEL        F   888665555          4      43000
JENNIFER        S               WALLACE           9876543 20-JUN-62 975 FIRE OAK       F   333445555          5      38000
JAMES           E               BORG            888665555 10-NOV-37 450 STONE          M   888665555          1     650000
JOYCLE                          GREW            999887775

6 rows selected.
```

```
SQL> DELETE FROM EMPLOYEE_2022503305 WHERE SSN = 9876543;

1 row deleted.
```

i. CHANGES IN VIEW TABLE

```
SQL> SELECT * FROM EMP_VIEW;

F_NAME               L_NAME                        SSN
-------------------- -------------------- -----------
JOHN                 SMITH                  123456789
FRANKLIN             WONG                   333445555
ALICIA               ZELAYA                 999887777
JAMES                BORG                   888665555
JOYCLE               GREW                   999887775
```

ii. CHANGES IN BASE TABLE

```
SQL> SELECT * FROM EMPLOYEE_2022503305;

F_NAME          MINIT           L_NAME                SSN B_DATE    ADDRESS            SE  SUPER_SSN       D_NO     SALARY
--------------- --------------- --------------- ---------- --------- ------------------ -- ---------- ---------- ----------
JOHN            B               SMITH           123456789 09-JAN-65 731, FONDREN       M   333445555          5      30000
FRANKLIN        T               WONG            333445555 08-DEC-55 638 VOUS           M       40000          5      40000
ALICIA          J               ZELAYA          999887777 19-JAN-68 3321 CASTEL        F   888665555          4      43000
JAMES           E               BORG            888665555 10-NOV-37 450 STONE          M   888665555          1     650000
JOYCLE                          GREW            999887775
```

2.

SETTING FOREIGN KEY

```
SQL> ALTER TABLE EMPLOYEE_2022503305 MODIFY D_NO REFERENCES DEPARTMENT_3305(D_NUMBER);

Table altered.
```

CREATING VIEW FOR BOTH THE TABLES AS EMP_VIEW

```
SQL> CREATE VIEW EMP_DEPT AS SELECT EMPLOYEE_2022503305.F_NAME, EMPLOYEE_2022503305.L_NAME, EMP
LOYEE_2022503305.SSN, DEPARTMENT_3305.D_NAME, DEPARTMENT_3305.D_NUMBER FROM EMPLOYEE_2022503305
, DEPARTMENT_3305 WHERE EMPLOYEE_2022503305.D_NO = DEPARTMENT_3305.D_NUMBER;
```

DESCRIBING THE VIEW

```
SQL> DESC EMP_DEPT;
 Name                                                                     Null?    Type
 ------------------------------------------------------------------------ -------- ------------
---
 F_NAME                                                                            VARCHAR2(20)
 L_NAME                                                                            VARCHAR2(20)
 SSN                                                                      NOT NULL NUMBER(10)
 D_NAME                                                                            VARCHAR2(20)
 D_NUMBER                                                                 NOT NULL NUMBER(2)
```

RETRIVING DATA FROM THE VIEW

```
SQL> SELECT * FROM EMP_DEPT;

F_NAME               L_NAME                      SSN D_NAME                 D_NUMBER
-------------------- -------------------- ---------- -------------------- ----------
JOHN                 SMITH                 123456789 RESEARCH                      5
FRANKLIN             WONG                  333445555 RESEARCH                      5
ALICIA               ZELAYA                999887777 ADMINISTRATION                4
JAMES                BORG                  888665555 HEADQUARTERS                  1
```

INSERTING IN EMP_DEPT TABLE

```
SQL> INSERT INTO EMP_DEPT(F_NAME,L_NAME,SSN) VALUES ('HALARY','FANE',123456789);
INSERT INTO EMP_DEPT(F_NAME,L_NAME,SSN) VALUES ('HALARY','FANE',123456789)
*
ERROR at line 1:
ORA-00001: unique constraint (CT2022503305.SYS_C0013977) violated
```

UPDATING IN EMP_DEPT TABLE

```
SQL> UPDATE EMP_DEPT SET SSN = 123456789 WHERE SSN = 888665555;
UPDATE EMP_DEPT SET SSN = 123456789 WHERE SSN = 888665555
*
ERROR at line 1:
ORA-00001: unique constraint (CT2022503305.SYS_C0013977) violated
```

DELETING IN EMP_DEPT TABLE

```
SQL> DELETE FROM EMP_DEPT WHERE SSN = 123456789
  2  ;

1 row deleted.
```

3.

CREATING A COMPLEX VIEW

```
SQL> CREATE VIEW EMPLOYEE_DEPT AS SELECT EMPLOYEE_2022503305.F_NAME, EMPLOYEE_2022503305.L_NAME,
EMPLOYEE_2022503305.SSN, DEPARTMENT_3305.D_NUMBER, DEPARTMENT_3305.D_NAME FROM EMPLOYEE_202250330
5, DEPARTMENT_3305 WHERE DEPARTMENT_3305.D_NUMBER =  EMPLOYEE_2022503305.D_NO;

View created.
```

RETRIVING DATA FROM THE VIEW

```
SQL> SELECT * FROM EMPLOYEE_DEPT;

F_NAME               L_NAME                    SSN   D_NUMBER D_NAME
-------------------- -------------------- ---------- ---------- --------------------
JOHN                 SMITH                 123456789          5 RESEARCH
FRANKLIN             WONG                  333445555          5 RESEARCH
ALICIA               ZELAYA                999887777          4 ADMINISTRATION
JENNIFER             WALLACE                98765321          4 ADMINISTRATION
RAMESH               NARAYANAN             666884444          5 RESEARCH
JAMES                BORG                  888665555          1 HEADQUARTERS

6 rows selected.
```

INSERTING IN COMPLEX VIEW TABLE

```
SQL> INSERT INTO EMP_DEPT VALUES ('HALARY','FANE',123456789,'RESEARCH',5);
INSERT INTO EMP_DEPT VALUES ('HALARY','FANE',123456789,'RESEARCH',5)
*
ERROR at line 1:
ORA-01776: cannot modify more than one base table through a join view
```

UPDATING IN COMPLEX VIEW TABLE

```
SQL> UPDATE EMP_DEPT SET D_NAME = 'OFFICE' WHERE D_NUMBER = 4;
UPDATE EMP_DEPT SET D_NAME = 'OFFICE' WHERE D_NUMBER = 4
                *
ERROR at line 1:
ORA-01779: cannot modify a column which maps to a non key-preserved table
```

DELETING IN COMPLEX VIEW TABLE

```
SQL> DELETE FROM EMP_DEPT WHERE SSN = 123456789
  2  ;

1 row deleted.

SQL> COMMIT;
```

4.

DROPING THE VIEW

```
SQL> DROP VIEW EMP_VIEW;

View dropped.
```

INSERTING INTO THE BASE TABLE

```
SQL>  INSERT INTO EMPLOYEE_2022503305 VALUES ('AHMAD','V','JABBAR',123123123,'18-JUL-1972','352 HOSTEL', 'M', 4531468,5,40000);

1 row created.

SQL> SELECT * FROM EMPLOYEE_2022503305;

F_NAME          MINIT           L_NAME                SSN B_DATE    ADDRESS             SE  SUPER_SSN       D_NO     SALARY
--------------- --------------- --------------- ---------- --------- ------------------- -- ---------- ---------- ----------
FRANKLIN        T               WONG            333445555 08-DEC-55 638 VOUS            M      40000          5      40000
ALICIA          J               ZELAYA          999887777 19-JAN-68 3321 CASTEL         F  888665555          4      43000
JAMES           E               BORG            888665555 10-NOV-37 450 STONE           M  888665555          1     650000
JOYCLE                          GREW            999887775
AHMAD           V               JABBAR          123123123 18-JUL-72 352 HOSTEL          M    4531468          5      40000
```

UPDATING THE BASE TABLE

```
SQL> UPDATE EMPLOYEE_2022503305 SET B_DATE = '19-AUG-1969' WHERE SSN = 999887775;

1 row updated.

SQL> SELECT * FROM EMPLOYEE_2022503305;

F_NAME          MINIT           L_NAME                SSN B_DATE    ADDRESS             SE  SUPER_SSN       D_NO     SALARY
--------------- --------------- --------------- ---------- --------- ------------------- -- ---------- ---------- ----------
FRANKLIN        T               WONG            333445555 08-DEC-55 638 VOUS            M      40000          5      40000
ALICIA          J               ZELAYA          999887777 19-JAN-68 3321 CASTEL         F  888665555          4      43000
JAMES           E               BORG            888665555 10-NOV-37 450 STONE           M  888665555          1     650000
JOYCLE                          GREW            999887775 19-AUG-69
AHMAD           V               JABBAR          123123123 18-JUL-72 352 HOSTEL          M    4531468          5      40000
```

DELETING A ROW IN THE BASE TABLE

```
SQL> DELETE FROM EMPLOYEE_2022503305 WHERE SSN =333445555;

1 row deleted.

SQL> SELECT * FROM EMPLOYEE_2022503305;

F_NAME              MINIT               L_NAME                    SSN B_DATE    ADDRESS             SE  SUPER_SSN       D_NO     SALARY
------------------- ------------------- ------------------- ---------- --------- ------------------- -- ---------- ---------- ----------
ALICIA              J                   ZELAYA               999887777 19-JAN-68 3321 CASTEL         F   888665555          4      43000
JAMES               E                   BORG                 888665555 10-NOV-37 450 STONE           M   888665555          1     650000
JOYCLE                                  GREW                 999887775 19-AUG-69
AHMAD               V                   JABBAR               123123123 18-JUL-72 352 HOSTEL          M     4531468          5      40000
```

5.

DROPING THE RESPECTIVE COLUMN WHICH IS AVAILABLE IN BOTH TABLES

```
SQL> ALTER TABLE EMPLOYEE_2022503305 DROP COLUMN B_DATE;

Table altered.
```

PERFORMING SELECT QUERY AFTER DROPING THE RESPECTIVE COLUMN

```
SQL> SELECT * FROM EMP_VIEW;
SELECT * FROM EMP_VIEW
              *
ERROR at line 1:
ORA-04063: view "CT2022503305.EMP_VIEW" has errors
```

6.

CREATING THE VIEW

```
SQL> CREATE VIEW EMP_VIEW AS SELECT F_NAME, L_NAME, SSN, SUPER_SSN FROM EMPLOYEE_2022503305;

View created.

SQL> SELECT * FROM EMP_VIEW;

F_NAME              L_NAME                    SSN  SUPER_SSN
------------------- ------------------- ---------- ----------
ALICIA              ZELAYA               999887777  888665555
JAMES               BORG                 888665555  888665555
JOYCLE              GREW                 999887775
AHMAD               JABBAR               123123123    4531468
```

CHANGING THE COLUMN NAME OF THE BASE TABLE

```
SQL> ALTER TABLE EMPLOYEE_2022503305 RENAME COLUMN F_NAME TO FIRST_NAME;

Table altered.

SQL> SELECT * FROM EMP_VIEW;
SELECT * FROM EMP_VIEW
              *
ERROR at line 1:
ORA-04063: view "CT2022503305.EMP_VIEW" has errors
```

7.

CREATING THE VIEW

```
SQL> CREATE VIEW EMP_VIEW AS SELECT FIRST_NAME, L_NAME, SSN, SUPER_SSN FROM EMPLOYEE_2022503305;

View created.

SQL> SELECT * FROM EMP_VIEW;

FIRST_NAME           L_NAME                    SSN  SUPER_SSN
-------------------- -------------------- ---------- ----------
ALICIA               ZELAYA               999887777  888665555
JAMES                BORG                 888665555  888665555
JOYCLE               GREW                 999887775
AHMAD                JABBAR               123123123    4531468
```

RENAMING THE BASE TABLE AND PERFORMING THE SELECT QUERY

```
SQL> RENAME EMPLOYEE_2022503305 TO EMPLOYEE_3305;

Table renamed.

SQL> SELECT * FROM EMP_VIEW;
SELECT * FROM EMP_VIEW
              *
ERROR at line 1:
ORA-04063: view "CT2022503305.EMP_VIEW" has errors
```

8.

CREATING THE VIEW

```
SQL> CREATE VIEW DEPT_VIEW AS SELECT D_NAME, D_NUMBER, MGR_SSN, LOCATION FROM DEPARTMENT_3305;

View created.

SQL> SELECT * FROM DEPT_VIEW;

D_NAME                D_NUMBER    MGR_SSN   LOCATION
-------------------- ---------- ---------- ----------
RESEARCH                     5  333445555
ADMINISTRATION               4  987654321
HEADQUARTERS                 1  888665555
```

MODIFYING THE DATATYPE OF THE BASE TABLE AND PERFORMING THE SELECT QUERY

```
SQL> ALTER TABLE DEPARTMENT_3305 MODIFY LOCATION VARCHAR2(15);

Table altered.

SQL> SELECT * FROM DEPT_VIEW;

D_NAME                D_NUMBER    MGR_SSN LOCATION
-------------------- ---------- ---------- ---------------
RESEARCH                     5  333445555
ADMINISTRATION               4  987654321
HEADQUARTERS                 1  888665555
```

9.

```
CREATE VIEW view_2022503305 AS
SELECT
    DeptID,
    DeptName,
    Location
FROM
    dept_view_2022503305
WHERE
    Location = 'New York';
```

RESULT

Thus, simplification the complex queries, enhancing the data and providing data abstraction using the logical representation by Views has been done successfully.

EX NO : 07

DATE : 02.04.2024

JOINS

AIM

To practice the queries on joins such as natural join, cross join, non equii join, self join.

SYNTAX

i. SELECT * FROM table_name1 NATURAL JOIN table_name2;

ii. SELECT * FROM table_name1 JOIN table_name2 ON condition;

iii. SELECT * FROM table_name1 JOIN table_name2 ON condtion WHERECondition;

iv. SELECT * FROM table_name1 INNER JOIN table_name2 ON condition;

v. SELECT * FROM table_name1 LEFT JOIN table_name2 ON condition;

vi. SELECT * FROM table_name1 RIGHT JOIN table_name2 ON condtion;

1.

```
SQL> SELECT D_NAME, D_NUMBER, D_LOCATIONS, MGR_SSN FROM DEPARTMENT_3305 NATURAL JOIN DEPT_LOCATION_3305;


D_NAME                D_NUMBER D_LOCATIONS            MGR_SSN
-------------------- ---------- -------------------- ----------
HEADQUARTERS                 1 HOUSTON              888665555
ADMINISTRATION               4 STAFFORD             987654321
RESEARCH                     5 SUGARLAND            333445555
RESEARCH                     5 HOUSTON              333445555
RESEARCH                     5 BELLAIRE             333445555
```

2.

```
SQL> SELECT FIRST_NAME, L_NAME, MINIT,D_NUMBER, D_NAME FROM EMPLOYEE_3305 NATURAL JOIN DEPARTMENT_3305;

FIRST_NAME           L_NAME               MINIT                D_NUMBER D_NAME
-------------------- -------------------- -------------------- ---------- --------------------
JOHN                 SMITH                B                             5 RESEARCH
FRANKLIN             WONG                 T                             5 RESEARCH
JENNIFER             WALLACE              S                             5 RESEARCH
RAMESH               NARAYAN              K                             5 RESEARCH
ALICIA               ZELAYA               J                             5 RESEARCH
JAMES                BORG                 E                             5 RESEARCH
JOYCLE               GREW                 A                             5 RESEARCH
AHMAD                JABBAR               V                             5 RESEARCH
JOHN                 SMITH                B                             4 ADMINISTRATION
FRANKLIN             WONG                 T                             4 ADMINISTRATION
JENNIFER             WALLACE              S                             4 ADMINISTRATION

FIRST_NAME           L_NAME               MINIT                D_NUMBER D_NAME
-------------------- -------------------- -------------------- ---------- --------------------
RAMESH               NARAYAN              K                             4 ADMINISTRATION
ALICIA               ZELAYA               J                             4 ADMINISTRATION
JAMES                BORG                 E                             4 ADMINISTRATION
JOYCLE               GREW                 A                             4 ADMINISTRATION
AHMAD                JABBAR               V                             4 ADMINISTRATION
JOHN                 SMITH                B                             1 HEADQUARTERS
FRANKLIN             WONG                 T                             1 HEADQUARTERS
JENNIFER             WALLACE              S                             1 HEADQUARTERS
RAMESH               NARAYAN              K                             1 HEADQUARTERS
ALICIA               ZELAYA               J                             1 HEADQUARTERS
JAMES                BORG                 E                             1 HEADQUARTERS

FIRST_NAME           L_NAME               MINIT                D_NUMBER D_NAME
-------------------- -------------------- -------------------- ---------- --------------------
JOYCLE               GREW                 A                             1 HEADQUARTERS
AHMAD                JABBAR               V                             1 HEADQUARTERS

24 rows selected.
```

QUESTIONS

1. Find the department detailssuch as department name, department number, location and manager
2. Find the employeeand his/herdepartment details
3. Find the employee and department details for the department "Research"
4. Find the employee and department details for the manager id 98754321.
5. Find the employee and department details for the department location 'Houston'
    i. Add a new employee without assigning his dept details.
    ii. Add a new department dnumber=3and dlocation='NEW YORK'
6. Find the employee and dept details who wereassigned to any of the department.
7. Find thedepartment detailsthat has employee assigned.
8. Find the employee and manager nameof each departmentOuter join(LEFT /RIGHT/FULL , IS NULL , IS NOT NULL)
9. Find the employee who has not assigned to any department.
10. .Find the DEPARTMENT that has no employee.
11. Find the employee who has not assigned to any department.
12. Find all employeeswho areeither assigned or not assigned to any department.
13. Find departmentdetails that has either employees or no employees

3.

```
SQL> SELECT FIRST_NAME, MINIT, L_NAME, SSN, B_DATE, SEX, SUPER_SSN, SALARY, ADDRESS, D_NO, D_NUMBER, D_NAME, MGR_SSN FROM EMPLOYEE_3305 JOIN DEPARTMENT_3305 ON D_NUMBER = 5;

FIRST_NAME          MINIT               L_NAME                    SSN B_DATE    SE  SUPER_SSN     SALARY ADDRESS                 D_NO   D_NUMBER D_NAME               MGR_SSN
------------------- ------------------- ------------------- ---------- --------- -- ---------- ---------- -------------------- ---------- ---------- -------------------- ----------
JOHN                B                   SMITH                123456789 09-JAN-65 M  3334445555      30000 721 HOUSTON                 5          5 RESEARCH             333445555
FRANKLIN            T                   WONG                3334445555 08-DEC-55 M  8886665555      40000 683 HOUSTON                 5          5 RESEARCH             333445555
JENNIFER            S                   WALLACE              987654321 20-AUG-41 F  8886665555      43000 291 HOUSTON                 4          5 RESEARCH             333445555
RAMESH              K                   NARAYAN             6668884444 15-SEP-52 M  3334445555      38000 975 HOUSTON                 5          5 RESEARCH             333445555
ALICIA              J                   ZELAYA               999887777 19-JAN-88 F   888665555      43000 3321 CASTEL                 4          5 RESEARCH             333445555
JAMES               E                   BORG                 888665555 10-NOV-37 M   888665555     650000 450 STONE                   1          5 RESEARCH             333445555
JOYCLE              A                   GREW                 999887775 31-JUL-72 M   333445555      25000 5631 HOUSTON                5          5 RESEARCH             333445555
AHMAD               V                   JABBAR               123123123 29-MAR-69 M     4531468      40000 352 HOSTEL                  5          5 RESEARCH             333445555

8 rows selected.
```

4.

```
SQL> SELECT FIRST_NAME, MINIT, L_NAME, SSN, B_DATE, SEX, SUPER_SSN, SALARY, ADDRESS, D_NO, D_NUMBER, D_NAME, MGR_SSN FROM EMPLOYEE_3305 JOIN DEPARTMENT_3305 ON SUPER_SSN = MGR_SSN WHERE SUPER_SSN =
 987654321;

FIRST_NAME          MINIT               L_NAME                    SSN B_DATE    SE  SUPER_SSN     SALARY ADDRESS                 D_NO   D_NUMBER D_NAME               MGR_SSN
------------------- ------------------- ------------------- ---------- --------- -- ---------- ---------- -------------------- ---------- ---------- -------------------- ----------
JENNIFER            S                   WALLACE              987654321 20-AUG-41 F  987654321      43000 291 HOUSTON                 4          4 ADMINISTRATION       987654321
ALICIA              J                   ZELAYA               999887777 19-JAN-88 F  987654321      43000 3321 CASTEL                 4          4 ADMINISTRATION       987654321
```

5.

```
SQL> SELECT * FROM EMPLOYEE_3305 INNER JOIN DEPT_LOCATION_3305 ON D_NO = D_NUMBER AND D_LOCATIONS = 'HOUSTON';

FIRST_NAME          MINIT               L_NAME                    SSN ADDRESS              SE  SUPER_SSN       D_NO     SALARY B_DATE      D_NUMBER D_LOCATIONS
------------------- ------------------- ------------------- ---------- -------------------- -- ---------- ---------- ---------- --------- ---------- ------------
JOHN                B                   SMITH                123456789 721 HOUSTON          M  3334445555          5      30000 09-JAN-65          5 HOUSTON
FRANKLIN            T                   WONG                3334445555 683 HOUSTON          M  8886665555          5      40000 08-DEC-55          5 HOUSTON
RAMESH              K                   NARAYAN             6668884444 975 HOUSTON          M  3334445555          5      38000 15-SEP-52          5 HOUSTON
JAMES               E                   BORG                 888665555 450 STONE            M   888665555          1     650000 10-NOV-37          1 HOUSTON
JOYCLE              A                   GREW                 999887775 5631 HOUSTON         M   333445555          5      25000 31-JUL-72          5 HOUSTON
AHMAD               V                   JABBAR               123123123 352 HOSTEL           M     4531468          5      40000 29-MAR-69          5 HOUSTON

6 rows selected.
```

i. Add a new employee without assigning his dept details.

```
SQL> INSERT INTO EMPLOYEE_3305(FIRST_NAME, MINIT, L_NAME, SSN, ADDRESS, SEX, SUPER_SSN, SALARY, B_DATE) VALUES(
  2  'GAYATHIRI',
  3  'V',
  4  'RAMAKRISHNAN',
  5  987987987,
  6  'SUGARLAND',
  7  'F',
  8  987987456,
  9  435000,
 10  '04-DEC-1989');

1 row created.
```

```
FIRST_NAME          MINIT               L_NAME                    SSN ADDRESS              SE  SUPER_SSN       D_NO     SALARY B_DATE
------------------- ------------------- ------------------- ---------- -------------------- -- ---------- ---------- ---------- ---------
JOHN                B                   SMITH                123456789 721 HOUSTON          M  3334445555          5      30000 09-JAN-65
FRANKLIN            T                   WONG                3334445555 683 HOUSTON          M  8886665555          5      40000 08-DEC-55
JENNIFER            S                   WALLACE              987654321 291 HOUSTON          F   987654321          4      43000 20-AUG-41
RAMESH              K                   NARAYAN             6668884444 975 HOUSTON          M  3334445555          5      38000 15-SEP-52
GAYATHIRI           V                   RAMAKRISHNAN         987987987 SUGARLAND            F   987987456                435000 04-DEC-89
ALICIA              J                   ZELAYA               999887777 3321 CASTEL          F   987654321          4      43000 19-JAN-88
JAMES               E                   BORG                 888665555 450 STONE            M   888665555          1     650000 10-NOV-37
JOYCLE              A                   GREW                 999887775 5631 HOUSTON         M   333445555          5      25000 31-JUL-72
AHMAD               V                   JABBAR               123123123 352 HOSTEL           M     4531468          5      40000 29-MAR-69
```

ii. Add a new department dnumber = 3and dlocation='NEW YORK'

```
SQL> SELECT * FROM DEPT_LOCATION_3305;

  D_NUMBER D_LOCATIONS
---------- --------------------
         1 HOUSTON
         5 BELLAIRE
         5 SUGARLAND
         5 HOUSTON
         3 NEWYORK
```

6.

```
SQL> SELECT * FROM EMPLOYEE_3305 INNER JOIN DEPARTMENT_3305 ON D_NO = D_NUMBER;

FIRST_NAME      MINIT           L_NAME               SSN ADDRESS             SE  SUPER_SSN       D_NO     SALARY B_DATE    D_NAME               D_NUMBER    MGR_SSN MGR_START
--------------- --------------- --------------- --------- -------------------- -- ---------- ---------- ---------- --------- -------------------- ---------- ---------- ---------
JAMES           E               BORG             888665555 450 STONE            M   888665555          1     650000 10-NOV-37 HEADQUARTERS                 1  888665555 19-JUN-81
JENNIFER        S               WALLACE          987654321 291 HOUSTON          F   987654321          4      43000 20-AUG-41 ADMINISTRATION               4  987654321 01-JAN-68
ALICIA          J               ZELAYA           999887777 3321 CASTEL          F   987654321          4      43000 19-JAN-88 ADMINISTRATION               4  987654321 01-JAN-68
JOHN            B               SMITH            123456789 721 HOUSTON          M  3334445555          5      30000 09-JAN-65 RESEARCH                     5  333445555 22-MAY-88
FRANKLIN        T               WONG            3334445555 683 HOUSTON          M  8886665555          5      40000 08-DEC-55 RESEARCH                     5  333445555 22-MAY-88
RAMESH          K               NARAYAN         6668884444 975 HOUSTON          M  3334445555          5      38000 15-SEP-52 RESEARCH                     5  333445555 22-MAY-88
AHMAD           V               JABBAR           123123123 352 HOSTEL           M     4531468          5      40000 29-MAR-69 RESEARCH                     5  333445555 22-MAY-88
JOYCLE          A               GREW             999887775 5631 HOUSTON         M   333445555          5      25000 31-JUL-72 RESEARCH                     5  333445555 22-MAY-88

8 rows selected.
```

7.

```
SQL> SELECT D.D_NUMBER, D.D_NAME, D.MGR_SSN, E.SSN FROM DEPARTMENT_3305 D INNER JOIN EMPLOYEE_3305 E ON E.D_NO = D.D_NUMBER INNER JOIN DEPT_LOCATION_3305 L ON L.D_NUMBER = E.D_NO;

  D_NUMBER D_NAME                MGR_SSN        SSN
---------- -------------------- --------- ----------
         5 RESEARCH             333445555  123456789
         5 RESEARCH             333445555  123456789
         5 RESEARCH             333445555  123456789
         5 RESEARCH             333445555 3334445555
         5 RESEARCH             333445555 3334445555
         5 RESEARCH             333445555 3334445555
         5 RESEARCH             333445555 6668884444
         5 RESEARCH             333445555 6668884444
         5 RESEARCH             333445555 6668884444
         5 RESEARCH             333445555  999887775
         5 RESEARCH             333445555  999887775

  D_NUMBER D_NAME                MGR_SSN        SSN
---------- -------------------- --------- ----------
         5 RESEARCH             333445555  999887775
         5 RESEARCH             333445555  123123123
         5 RESEARCH             333445555  123123123
         5 RESEARCH             333445555  123123123
         1 HEADQUARTERS         888665555  888665555

16 rows selected.
```

8.

```
SQL> SELECT E1.SSN, E1.FIRST_NAME, E1.SUPER_SSN, E2.FIRST_NAME, E2.L_NAME FROM EMPLOYEE_3305 E1, EMPLOYEE_3305 E2 WHERE E1.SUPER_SSN = E2.SSN;

       SSN FIRST_NAME           SUPER_SSN FIRST_NAME          L_NAME
---------- -------------------- ---------- -------------------- --------------------
 123456789 JOHN                3334445555 FRANKLIN             WONG
6668884444 RAMESH              3334445555 FRANKLIN             WONG
 987654321 JENNIFER             987654321 JENNIFER             WALLACE
 999887777 ALICIA               987654321 JENNIFER             WALLACE
 888665555 JAMES                888665555 JAMES                BORG
```

9.

```
SQL> SELECT FIRST_NAME, L_NAME, SSN, D_NUMBER FROM EMPLOYEE_3305 LEFT JOIN DEPARTMENT_3305 ON D_NUMBER = D_NO WHERE D_NUMBER IS NULL;

FIRST_NAME           L_NAME                    SSN   D_NUMBER
-------------------- -------------------- ---------- ----------
GAYATHIRI            RAMAKRISHNAN          987987987
```

10.

```
SQL> SELECT D_NO, D_NAME, MGR_SSN, FIRST_NAME FROM EMPLOYEE_3305 RIGHT JOIN DEPARTMENT_3305 ON D_NUMBER = D_NO WHERE D_NUMBER IS NULL;

no rows selected
```

11.

```
SQL> SELECT FIRST_NAME, L_NAME, D_NO FROM EMPLOYEE_3305 LEFT JOIN DEPARTMENT_3305 ON D_NO = D_NUMBER WHERE D_NUMBER IS NULL;

FIRST_NAME           L_NAME                    D_NO
-------------------- -------------------- ----------
GAYATHIRI            RAMAKRISHNAN
```

12.

```
SQL> SELECT FIRST_NAME, MINIT, L_NAME, SSN, D_NO, D_NAME FROM EMPLOYEE_3305 LEFT JOIN DEPARTMENT_3305 ON D_NO = D_NUMBER WHERE D_NUMBER IS NULL OR D_NUMBER IS NOT NULL;

FIRST_NAME           MINIT                L_NAME                    SSN       D_NO D_NAME
-------------------- -------------------- -------------------- ---------- ---------- --------------------
JOHN                 B                    SMITH                 123456789          5 RESEARCH
FRANKLIN             T                    WONG                 3334445555          5 RESEARCH
RAMESH               K                    NARAYAN              6668884444          5 RESEARCH
JOYCLE               A                    GREW                  999887775          5 RESEARCH
AHMAD                V                    JABBAR                123123123          5 RESEARCH
JENNIFER             S                    WALLACE               987654321          4 ADMINISTRATION
ALICIA               J                    ZELAYA                999887777          4 ADMINISTRATION
JAMES                E                    BORG                  888665555          1 HEADQUARTERS
GAYATHIRI            V                    RAMAKRISHNAN          987987987

9 rows selected.
```

13.

```
SQL> SELECT FIRST_NAME, MINIT, L_NAME, SSN, D_NO, D_NAME FROM EMPLOYEE_3305 RIGHT JOIN DEPARTMENT_3305 ON D_NO = D_NUMBER WHERE D_NUMBER IS NOT NULL;

FIRST_NAME           MINIT                L_NAME                    SSN       D_NO D_NAME
-------------------- -------------------- -------------------- ---------- ---------- --------------------
JAMES                E                    BORG                  888665555          1 HEADQUARTERS

JENNIFER             S                    WALLACE               987654321          4 ADMINISTRATION
ALICIA               J                    ZELAYA                999887777          4 ADMINISTRATION
JOHN                 B                    SMITH                 123456789          5 RESEARCH
FRANKLIN             T                    WONG                 3334445555          5 RESEARCH
RAMESH               K                    NARAYAN              6668884444          5 RESEARCH
AHMAD                V                    JABBAR                123123123          5 RESEARCH
JOYCLE               A                    GREW                  999887775          5 RESEARCH

9 rows selected.
```

RESULT

Thus, practicing the queries on joins such as natural join, cross join, non equii join, self join has been done successfully.

EX NO : 08

DATE : 02.04.2024

AGGREGATE FUNCTIONS – GROUP BY, ORDER BY

AIM

To operate the aggregate functions like Group by and order by clause in queries.

SYNTAX

i. SELECT column_name FROM table_name GROUP BY column_name;

ii. SELECT column_name FROM table_name HAVING condition;

iii. SELECT column_name FROM table_name ORDER BY column_name DESC/ASC;

iv. SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name;

1.

```
SQL> SELECT DNO,COUNT(*) FROM EMPLOYEE_2022503305 GROUP BY DNO;

       DNO   COUNT(*)
---------- ----------
         1          1
         5          4
         4          3
```

2.

```
SQL> SELECT DNO, AVG(SALARY) FROM EMPLOYEE_2022503305 GROUP BY DNO;

       DNO AVG(SALARY)
---------- -----------
         1       55000
         5       31250
         4       31000
```

3.

```
SQL> SELECT DNO,COUNT(*) FROM EMPLOYEE_2022503305 GROUP BY DNO
  2  HAVING COUNT(*) > 10;

no rows selected
```

4.

```
SQL> SELECT DNO, SUM(SALARY), COUNT(*) FROM EMPLOYEE_2022503305
  2  GROUP BY DNO;

       DNO SUM(SALARY)   COUNT(*)
---------- ----------- ----------
         1       55000          1
         5      125000          4
         4       93000          3
```

QUESTIONS

1. How many employees are working in each department?
2. What is the average salary for employees in each department?
3. How many departments have more than 10 employees?
4. What is the total salary expense for each department?
5. How many employees have a salary greater than $50,000?
6. What is the highest salary in the company?
7. Findthe number of employees directly managed by each employee.
8. How many employees have the same salary?
9. Which department has the highest average salary?
10. How many employees were born in each year?
11. What is the total salary expense for male and female employees separately?
12. How many employees have the same last name?
13. Which department has the most employees?
14. What is the average age of employees in each department?
15. How many employees have a salary within a specific range?
16. What is the total salary expense for employees supervised by each supervisor?
17. How many employees have the same birthdate?
18. Which department has the highest total salary expense?
19. What is the average salary for employees hired in each year?
20. How many employees live in each city?

5.

```
SQL> SELECT DNO, SALARY, COUNT(*) FROM EMPLOYEE_2022503305
  2  GROUP BY DNO, SALARY HAVING SALARY>50000;

       DNO     SALARY   COUNT(*)
---------- ---------- ----------
         1      55000          1
```

6.

```
SQL> SELECT * FROM (SELECT MAX(SALARY) FROM EMPLOYEE_2022503305
  2  ORDER BY SALARY DESC)
  3  WHERE ROWNUM=1;

MAX(SALARY)
-----------
      55000
```

7.

```
SQL> SELECT E.SUPER_SSN, COUNT(*) FROM EMPLOYEE_2022503305 E JOIN
  2  EMPLOYEE_2022503305 M ON E.SUPER_SSN = M.SSN GROUP BY E.SUPER_SSN;

 SUPER_SSN   COUNT(*)
---------- ----------
 987987987          1
 987654321          2
 333445555          1
 888665555          2
```

8.

```
SQL> SELECT SALARY, COUNT(*) FROM EMPLOYEE_2022503305 GROUP BY SALARY;

    SALARY   COUNT(*)
---------- ----------
     55000          1
     30000          2
     43000          1
     40000          1
     25000          3
```

9.

```
SQL> SELECT * FROM (SELECT AVG(SALARY) FROM EMPLOYEE_2022503305
  2  ORDER BY DNO, SALARY DESC) WHERE ROWNUM=1;

AVG(SALARY)
-----------
      34125
```

10.

```
SQL> SELECT DNO, B_DATE, COUNT(*) FROM EMPLOYEE_2022503305
  2  GROUP BY DNO, B_DATE;

       DNO B_DATE      COUNT(*)
---------- --------- ----------
         5 15-SEP-62          1
         1 10-NOV-37          1
         4 20-JUN-41          1
         5 31-JUL-72          1
         5 09-JAN-65          1
         4 29-MAR-69          1
         5 08-DEC-55          1
         4 19-JAN-68          1

8 rows selected.
```

11.

```
SQL> SELECT SEX, SUM(SALARY), COUNT(*) FROM EMPLOYEE_2022503305
  2  GROUP BY SEX;

SE SUM(SALARY)   COUNT(*)
-- ----------- ----------
M       180000          5
F        93000          3
```

12.

```
SQL> SELECT DNO, LNAME, COUNT(*) FROM EMPLOYEE_2022503305
  2  GROUP BY DNO, LNAME HAVING COUNT(*) > 1;

no rows selected
```

13.

```
SQL> SELECT * FROM (SELECT E.DNO, D.DNAME, COUNT(*) FROM EMPLOYEE_2022503305 E JOIN
  2  DEPARTMENT_2022503305 D ON E.DNO = D.DNUMBER GROUP BY E.DNO, D.DNAME ORDER BY
  3  COUNT(*) DESC) WHERE ROWNUM = 1;

       DNO DNAME              COUNT(*)
---------- --------------- ----------
         5 RESEARCH                 4
```

14.

```
SQL> SELECT DNO, DNAME, AVG(EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM B_DATE))
  2  FROM EMPLOYEE_2022503305 JOIN DEPARTMENT_2022503305 ON DNUMBER = DNO
  3  GROUP BY DNO, DNAME;

       DNO DNAME          AVG(EXTRACT(YEARFROMCURRENT_DATE)-EXTRACT(YEARFROMB_DATE))
---------- --------------- ------------------------------------------------------------
         1 HEADQUARTERS                                                            87
         4 ADMINISTRATION                                                    64.6666667
         5 RESEARCH                                                            60.5
```

15.

```
SQL> SELECT SALARY, COUNT(*) FROM EMPLOYEE_2022503305
  2  GROUP BY SALARY HAVING SALARY BETWEEN 25000 AND 40000;

    SALARY   COUNT(*)
---------- ----------
     30000          2
     40000          1
     25000          3
```

16.

```
SQL> SELECT DNO, DNAME, SUM_SALARY FROM(
  2  SELECT DNO, DNAME, SUM(SALARY) AS SUM_SALARY FROM EMPLOYEE_2022503305 JOIN
  3  DEPARTMENT_2022503305 ON DNUMBER = DNO
  4  GROUP BY DNO, DNAME
  5  ORDER BY SUM_SALARY DESC) WHERE ROWNUM <= 1;

       DNO DNAME          SUM_SALARY
---------- -------------- ----------
         5 RESEARCH           125000
```

17.

```
SQL> SELECT B_DATE, COUNT(*) FROM EMPLOYEE_2022503305
  2  GROUP BY B_DATE HAVING COUNT(*) > 1;

no rows selected
```

18.

```
SQL> SELECT DNO, DNAME, SUM(SALARY) AS SUM_SALARY FROM
  2  EMPLOYEE_2022503305 JOIN DEPARTMENT_2022503305 ON DNUMBER = DNO
  3  GROUP BY DNO, DNAME
  4  ORDER BY SUM_SALARY DESC;

       DNO DNAME           SUM_SALARY
---------- --------------- ----------
         5 RESEARCH            125000
         4 ADMINISTRATION       93000
         1 HEADQUARTERS         55000
```

19.

```
SQL> SELECT EXTRACT(YEAR FROM B_DATE) AS YEAR, AVG(SALARY)
  2  FROM EMPLOYEE_2022503305 GROUP BY EXTRACT(YEAR FROM B_DATE);

      YEAR AVG(SALARY)
---------- -----------
      1962       30000
      1968       25000
      1955       40000
      1941       43000
      1972       25000
      1969       25000
      1965       30000
      1937       55000

8 rows selected.
```

20.

```
SQL> SELECT SUBSTR(REGEXP_SUBSTR(ADDRESS,',[^,]+'),2) AS CITY, COUNT(*) FROM
  2  EMPLOYEE_2022503305 GROUP BY SUBSTR(REGEXP_SUBSTR(ADDRESS,',[^,]+'),2);

CITY                                                                                                          COUNT(*)
------------------------------------------------------------------------------------------------------------- ----------
 HOUSTON                                                                                                             2
 RICE                                                                                                                1
 FIRE OK                                                                                                             1
 STONE                                                                                                               1
 SPRING                                                                                                              1
 VOSS                                                                                                                1
 BALLAIRE                                                                                                            1

7 rows selected.
```

RESULT

Thus, operating the aggregate functions like Group by and order by clause in queries has been done successfully.

EX NO : 09

DATE : 16.04.2024

SET OPERATIONS

AIM

To do the set operations like union, intersect, minus in queries.

SYNTAX

i. SELECT column_name FROM table_name1 UNION SELECT column_name FROM table_name2;

ii. SELECT column_name FROM table_name1 INTERSECT SELECT column_name FROM table_name2;

iii. SELECT column_name FROM table_name1 MINUS SELECT column_name FROM table_name2;

iv. SELECT column_name FROM table_name1 WHERE condition UNION SELECT column_name FROM table_name2 WHERE condition;

1.

```
SQL> SELECT * FROM EMPLOYEE_3305;

FNAME           M LNAME               SSN B_DATE    ADDRESS                        SE     SALARY  SUPER_SSN        DNO
--------------- - --------------- --------- --------- ------------------------------ -- ---------- ---------- ----------
JOHN            B SMITH           123456789 09-JAN-65 731 FONDREN, HOUSTON, TX       M       30000  333445555          5
FRANKLIN        T WONG            333445555 08-DEC-55 3321 CASTLE, SPRING, TX        F       25000  987654321          4
ALICIA          J ZELAYA          999887777 19-JAN-68 3321 VOSS, HOUSTON, TX         F       25000  888665555          4
JENNIFER        S WALLACE         987654321 20-JUN-41 291 BERRY, BELLAIRE, TX        F       43000  888665555          4
RAMESH          K NARAYAN         666884444 15-SEP-62 973 FIRE OAL, HUMBLE, TX       M       38000  333445555          5
```

```
SQL> SELECT * FROM MGR_2022503305;

FNAME           M LNAME             MGR_SSN B_DATE    ADDRESS            SE     SALARY MGR_SUPER_SSN        DNO
--------------- - --------------- --------- --------- ------------------ -- ---------- ------------- ----------
John            D Doe           123456789 15-JAN-90 123 Main St        M       50000     987654321          1
Jane            M Smith         987654321 20-MAY-92 456 Elm St         F       60000                        2
Alice           K Johnson       246813579 10-OCT-88 789 Oak St         F       55000     987654321          1
Bob             T Williams      135792468 25-AUG-95 101 Pine St        M       52000     987654321          1
Emily           R Brown         369258147 05-DEC-93 222 Maple St       F       58000                        2
```

2.

```
SQL> select dno from EMPLOYEE_3305  union  select dno from MGR_2022503305;

       DNO
----------
         1
         2
         4
         5
```

3.

```
SQL> select dno from EMPLOYEE_3305  intersect  select dno from MGR_2022503305;

no rows selected
```

QUESTIONS

1. 1.Create a Emp_Table and Mgr_Table details using Employee table.
2. List all unique department IDs from both Employees and Managers
3. Find common department IDs between Employees and Managers
4. Find employees and managers who are in the same department
5. List department IDs that are in Employees but not in Managers
6. Find all employees and managers.
7. List employees who are not managers
8. Identify departments with employees but no managers
9. Find the employees who works in department as Manager=33344555 works in.
10. Find the employees and managers who work in department 5 but not department 4.
11. List employees and managers working in departments with IDs greater than 55555555

4.

```
SQL> select fname,lname,dno from EMPLOYEE_3305  union  select fname,lname,dno from MGR_2022503305  order by 3;

FNAME                LNAME                       DNO
-------------------- -------------------- ----------
Alice                Johnson                       1
Bob                  Williams                      1
John                 Doe                           1
Emily                Brown                         2
Jane                 Smith                         2
ALICIA               ZELAYA                        4
FRANKLIN             WONG                          4
JENNIFER             WALLACE                       4
JOHN                 SMITH                         5
RAMESH               NARAYAN                       5

10 rows selected.
```

5.

```
SQL> select dno from EMPLOYEE_3305  minus  select dno from MGR_2022503305;

       DNO
----------
         4
         5
```

6.

```
SQL> select fname,lname from EMPLOYEE_3305  union  select fname,lname from MGR_2022503305;

FNAME                LNAME
-------------------- --------------------
ALICIA               ZELAYA
Alice                Johnson
Bob                  Williams
Emily                Brown
FRANKLIN             WONG
JENNIFER             WALLACE
JOHN                 SMITH
Jane                 Smith
John                 Doe
RAMESH               NARAYAN

10 rows selected.
```

7.

```
SQL> select fname from EMPLOYEE_3305  minus  select fname from MGR_2022503305;

FNAME
--------------------
ALICIA
FRANKLIN
JENNIFER
JOHN
RAMESH
```

8.

```
SQL> select dno from EMPLOYEE_3305  minus  select dno from MGR_2022503305;

       DNO
----------
         4
         5
```

9.

```
SQL> select fname,lname from EMPLOYEE_3305  where dno = (select dno from EMPLOYEE_3305
intersect select dno from MGR_2022503305 where ssn=333445555);

no rows selected
```

10.

```
SQL> select fname,lname,dno from EMPLOYEE_3305 where dno=5  union  select fname,lname,
dno from MGR_2022503305 where dno=5  minus  select fname,lname,dno from EMPLOYEE_3305
where dno=4  minus  select fname,lname,dno from MGR_2022503305 where dno=4;

FNAME                LNAME                       DNO
-------------------- -------------------- ----------
JOHN                 SMITH                         5
RAMESH               NARAYAN                       5
```

11.

```
SQL> select fname,lname,ssn from EMPLOYEE_3305 where ssn > 55555555  union  select
fname,lname,mgr_ssn from MGR_2022503305 where mgr_ssn > 55555555;

FNAME                LNAME                     SSN
-------------------- -------------------- ----------
ALICIA               ZELAYA                999887777
Alice                Johnson               246813579
Bob                  Williams              135792468
Emily                Brown                 369258147
FRANKLIN             WONG                  333445555
JENNIFER             WALLACE               987654321
JOHN                 SMITH                 123456789
Jane                 Smith                 987654321
John                 Doe                   123456789
RAMESH               NARAYAN               666884444

10 rows selected.
```

RESULT

Thus the set operations has been done successfully and output was verified

EX NO : 10

DATE : 23.04.2024

NORMALISATION

AIM

To implement normalization for the given table and create tables.

PROCEDURE

STEP 1: DETERMINE THE GIVEN RELATION

STEP 2: CHECK AND VERIFY IT SATISFIES 1NF

STEP 3: IF IT SATISFIES 1NF THEN CHECK WHETHER IT SATISFIES 2NF.

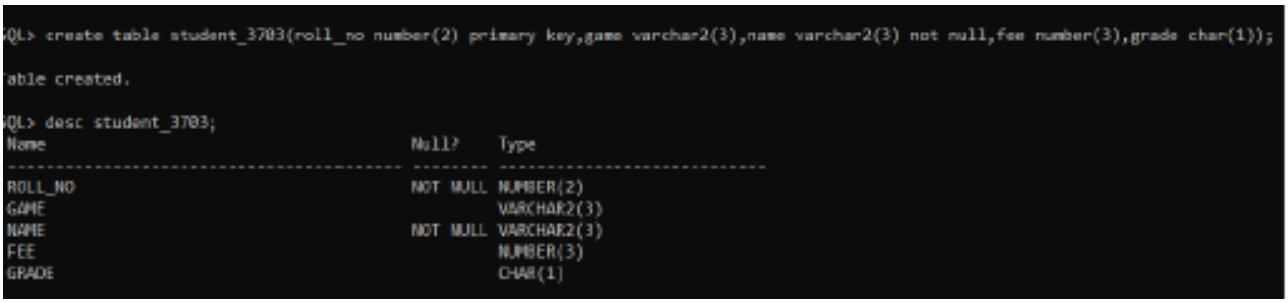
STEP 4: IF IT SATISFIES 2NF THEN CHECK WHETHER IT SATISFIES 3NF.

STEP 5: IF YES THEN 4NF.

STEP 6: IF YES THEN 5NF

OUTPUT

- 1NF is satisfied.
- 2NF is not satisfied as the proper subset of the candidate key determines the non-prime attribute.
- As 2NF is not satisfied then all other higher normal forms cannot be satisfied.
- So, the highest normal form satisfied by the table is 1NF.



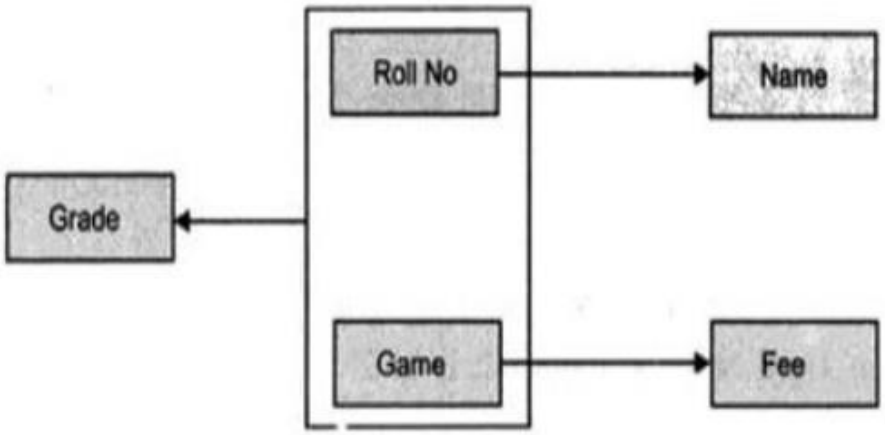
QUESTIONS

1. DETERMINE THE NORMAL FORM FOR THE FOLLOWING

Student

RollNo.	Game	Name	Fee	Grade
1	Cricket	Amit	200	A
2	Badminton	Dheeraj	150	B
3	Cricket	Lalit	200	A
4	Badminton	Parul	150	C
5	Hockey	Jack	100	A
6	Cricket	John	200	C

(a)



Dependency Chart

OUTPUT

- 1NF is satisfied as there are no multivalued tuples.
- 2NF is satisfied as the candidate key is single.
- 3NF is not satisfied as there is a transitive dependency. (rollno ->semester; semester->hostel)

So here the table is broke into two tables:

1.Roll no, name, semester (primary key->roll no, foreign key)

2.Semester, hostel (primary key->semester)

```
SQL> create table student2_3703(semester number(1) primary key,hostel varchar2(2));

Table created.

SQL> desc student2_3703;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SEMESTER                                  NOT NULL NUMBER(1)
 HOSTEL                                             VARCHAR2(2)
```

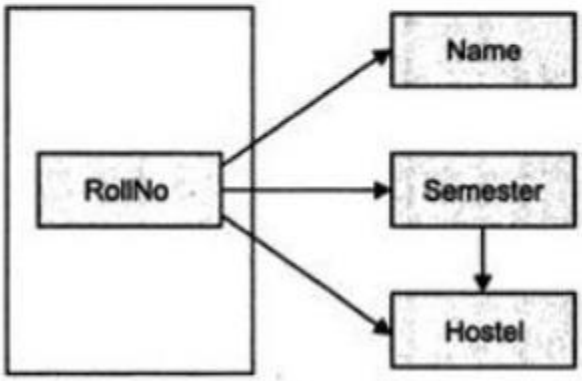
```
SQL> create table student2a_3703(roll_no number(1) primary key,name varchar2(30), semester number(1),constraint fk1 foreign key(semester) references student2_3703(semester));

Table created.

SQL> desc student2a_3703;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLL_NO                                   NOT NULL NUMBER(1)
 NAME                                               VARCHAR2(30)
 SEMESTER                                           NUMBER(1)
```

2. DETERMINE THE NORMAL FORM FOR THE FOLLOWING

Student

RollNo.	Name	Semester	Hostel
1	Lalit	1	H1
2	Gaurav	2	H2
3	Vishal	1	H1
4	Neha	4	H4
5	John	3	H3

(a)



(b)

OUTPUT

- Till 5NF all the normal forms are satisfied and so the table is all fine without dividing it furthermore.

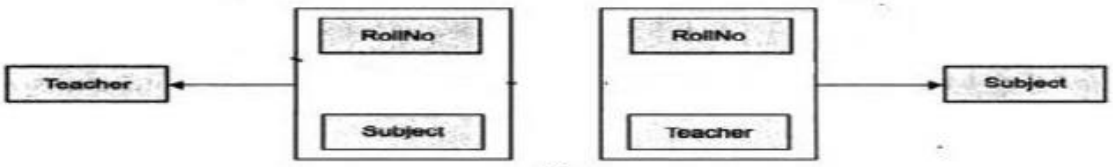
```
SQL> create table student3_3703(roll_no number(1), subject varchar2(10),teacher varchar2(2));

Table created.

SQL> desc student3_3703;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLL_NO                                            NUMBER(1)
 SUBJECT                                            VARCHAR2(10)
 TEACHER                                            VARCHAR2(2)
```

3. DETERMINE THE NORMAL FORM FOR THE FOLLOWING

Student

RollNo.	Subject	Teacher
1	C	T1
2	C++	T2
3	C	T1
4	Java	T3
5	Java	T3
1	Oracle	T5
6	Oracle	T5
3	C++	T2
7	VB	T4
8	Oracle	T6

(a)



(b)

RESULT

Thus, the implementation of normalization for the given table and create tables has been done successfully.

EX NO : 11

DATE : 14.05.2024

SUB QUERIES

AIM

To implement subquery type queries for the given problems using oracle.

SYNTAX

i. SELECT column_name1 || column_name2 FROM table_name WHERE column_name IN (SELECT stmt);

ii. SELECT column_names FROM table_name WHERE column_name = (SELECT stmt);

iii. SELECT column_name FROM table_name WHERE column_name > (SELECT stmt);

iv. SELECT column_name FROM table_name WHERE (SELECT stmt);

v. SELECT column_name FROM table_name WHERE column_name < (SELECT stmt);

1.

```
SQL> SELECT FNAME || LNAME FROM EMPLOYEE_3305 WHERE DNO IN
  2  (SELECT DNUMBER FROM DEPT_LOCATIONS_3305 WHERE DLOCATION = 'NEW YORK');
```

2.

```
SQL> SELECT SSN FROM EMPLOYEE_3305 WHERE DNO IN(SELECT DNO FROM EMPLOYEE_3305 WHERE
FNAME = 'JOHN' AND LNAME = 'SMITH');

       SSN
----------
 123456789
 666884444
 123456783
```

3.

```
SQL> SELECT FNAME || LNAME FROM EMPLOYEE_3305 WHERE SALARY > (SELECT AVG(SALARY) FROM
EMPLOYEE_3305);

FNAME||LNAME
----------------------------------------
JENNIFERWALLACE
RAMESHNARAYAN
```

4.

```
SQL> SELECT SSN FROM EMPLOYEE_3305 WHERE DNO = (SELECT DNUMBER FROM DEPARTMENT_3305
WHERE DNUMBER = 5);

       SSN
----------
 123456789
 666884444
 123456783
```

QUESTIONS

1. List the names of employees who work in departments located in 'New York'.
2. Retrieve the SSNs of employees who work in departments managed by 'John Doe'.
3. Find the names of employees who earn a salary greater than the average salary of their department.
4. Retrieve the SSNs of employees who work on projects located in the department with Dnumber '5'.
5. Find the names of employees who work on more than one project.
6. List the SSNs of employees who do not work on any project.
7. Retrieve the SSNs of employees who have the same manager as employee with SSN '987654321'.
8. List the names of employees who work on projects with 'ProjectX' in the project name.
9. Find the SSNs of employees who don't work on any project.
10. Retrieve the names of employees who have the same salary as employee 'Jane Smith'.
11. Retrieve the names of employees who earn a salary greater than the average salary of all employees.
12. Find the names of employees who work on projects managed by 'John Doe'.

5.

```
SQL> SELECT FNAME, LNAME FROM EMPLOYEE_3305 WHERE (SELECT COUNT(*) FROM WORKS_ON_3305
WHERE WORKS_ON_3305.ESSN = EMPLOYEE_3305.SSN) > 1;

no rows selected
```

6.

```
SQL> SELECT FNAME || LNAME FROM EMPLOYEE_3305 WHERE SSN IN (SELECT ESSN FROM WORKS_ON_3305);

no rows selected
```

7.

```
SQL> SELECT SSN, FNAME FROM EMPLOYEE_3305 WHERE SUPER_SSN IN (SELECT SUPER_SSN FROM EMPLOYEE_
3305 WHERE SSN = 987654321) AND SSN != 987654321;

       SSN FNAME
---------- --------------------
 999887777 ALICIA
```

8.

```
SQL> SELECT FNAME, LNAME FROM EMPLOYEE_3305 WHERE (SELECT COUNT(*) FROM WORKS_ON_3305 WHERE W
ORKS_ON_3305.ESSN = EMPLOYEE_3305.SSN) = 0;

FNAME                LNAME
-------------------- --------------------
JOHN                 SMITH
FRANKLIN             WONG
ALICIA               ZELAYA
JENNIFER             WALLACE
RAMESH               NARAYAN
JOHN                 SMITH

6 rows selected.
```

9.

```
SQL> SELECT FNAME || LNAME FROM EMPLOYEE_3305 WHERE SSN IN (SELECT ESSN FROM WORKS_ON_3305);

no rows selected
```

10.

```
SQL> SELECT FNAME || ' ' || LNAME AS "NAME" FROM EMPLOYEE_3305 WHERE SALARY = (SELECT SALARY
FROM EMPLOYEE_3305 WHERE FNAME = 'JOHN' AND LNAME = 'SMITH');

NAME
---------------------------------------
JOHN SMITH
```

11.

```
SQL> SELECT FNAME || LNAME FROM EMPLOYEE_3305 WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYE
E_3305);

FNAME||LNAME
---------------------------------------
JENNIFERWALLACE
RAMESHNARAYAN
```

12.

```
SQL> SELECT FNAME || LNAME FROM EMPLOYEE_3305 WHERE SSN IN (SELECT SUPER_SSN FROM EMPLOYEE_33
05 WHERE FNAME = 'JOHN' AND LNAME = 'SMITH');

FNAME||LNAME
---------------------------------------
FRANKLINWONG
```

RESULT

Thus, the implementation of subqueries has been done successfully.

EX NO : 12

DATE : 14.05.2024

PLSQL BASICS

AIM

To perform Select, retrieving, insert, delete and update operations using PLSQL basics.

PROCEDURE

STEP 1: Open the notepad.

STEP 2: Type the PLSQL code in the notepad

STEP 3: Declare the necessary variables you want.

STEP 4: Save the file as filname.sql

STEP 5: Go the SQLPlus and Type “GET source_address”

STEP 6: Then Type “SET SERVEROUTPUT ON”

STEP 7: Click enter and comment “/”.

SYNTAX

i. [DECLARE]
BEGIN
--statements

[EXCEPTION]
END;

ii. BEGIN
SELECT statement;
UPDATE statement;
DELETE statement;
DBMS_OUTPUT.PUT_LINE (‘value’);

1. SELECT STATEMENT IN PLSQL

```
SQL> get D:/DBMS/2_PLSQL.SQL
  1  DECLARE
  2   V_FNAME VARCHAR2(30);
  3   V_LNAME VARCHAR2(30);
  4  BEGIN
  5   SELECT FNAME, LNAME INTO V_FNAME, V_LNAME FROM EMPLOYEE_2022503305 WHERE FNAME = 'JAMES';
  6   DBMS_OUTPUT.PUT_LINE ('FIRST NAME: '||V_FNAME ||'  LAST NAME: '||V_LNAME);
  7* END;
SQL> SET SERVEROUTPUT ON
SQL> /
FIRST NAME: JAMES  LAST NAME: BORG

PL/SQL procedure successfully completed.
```

2. RETRIVING DATA IN PLSQL

```
SQL> get D:/DBMS/1_PLSQL.SQL
  1  DECLARE
  2   V_D_NAME DEPARTMENT_3305.D_NAME%TYPE;
  3   V_D_NUMBER DEPARTMENT_3305.D_NUMBER%TYPE;
  4   BEGIN
  5    SELECT D_NAME, D_NUMBER INTO V_D_NAME, V_D_NUMBER FROM DEPARTMENT_3305 WHERE D_NUMBER = &D_NUMBER_INPUT;
  6    DBMS_OUTPUT.PUT_LINE('Department Name: '||V_D_NAME ||'   Department Number: '||V_D_NUMBER);
  7*  END;
SQL> SET SERVEROUTPUT ON
SQL> /
Enter value for d_number_input: 5
old   5:   SELECT D_NAME, D_NUMBER INTO V_D_NAME, V_D_NUMBER FROM DEPARTMENT_3305 WHERE D_NUMBER = &D_NUMBER_INPUT;
new   5:   SELECT D_NAME, D_NUMBER INTO V_D_NAME, V_D_NUMBER FROM DEPARTMENT_3305 WHERE D_NUMBER = 5;
Department Name: RESEARCH   Department Number: 5

PL/SQL procedure successfully completed.
```

3. INSERTING DATA IN PLSQL

```
SQL> get D:/DBMS/3_PLSQL.SQL
SP2-0160: unable to open "D:/DBMS/3_PLSQL.SQL"
SQL> get D:/DBMS/3_PLSQL.SQL
  1  BEGIN
  2   INSERT INTO COURSE_2022503305 VALUES (10007, 'Linear Algbra', 20, 4, 'Algebra');
  3* END;
SQL> SET SERVEROUTPUT ON
SQL> /

PL/SQL procedure successfully completed.
```

4. DELETING DATA IN PLSQL

```
SQL> DECLARE
  2  BEGIN
  3  DELETE FROM EMPLOYEE_3305
  4  WHERE FNAME = 'JOYCE';
  5  END;
  6  /

PL/SQL procedure successfully completed.
```

5. IF CONDITION

```
  1  DECLARE
  2      V_day VARCHAR2(15);
  3      v_time VARCHAR(8);
  4  BEGIN
  5      v_day := TO_CHAR(SYSDATE, 'fmDAY');
  6      v_time := TO_CHAR(SYSDATE, 'HH24:MI');
  7      DBMS_OUTPUT.put_line('Day:' || V_day);
  8      DBMS_OUTPUT.put_line('Time:' || v_time); -- corrected colon to semicolon
  9      IF V_day IN ('Saturday', 'Sunday') THEN
 10          DBMS_OUTPUT.put_line(V_day || ',' || v_time);
 11          IF v_time BETWEEN '12:01' AND '23:59' THEN
 12              DBMS_OUTPUT.put_line('It''s afternoon'); -- corrected quotation mark
 13          ELSE
 14              DBMS_OUTPUT.put_line('It''s morning'); -- corrected quotation mark
 15          END IF;
 16      END IF;
 17* END;
 18  /
Day:TUESDAY
Time:21:22

PL/SQL procedure successfully completed.

SQL>
```

6. FUNCTION

```
SQL> create or replace function totalemp
  2  return number is total number(2) := 0;
  3  begin
  4  select count(*) into total from e_3049;
  5  return total;
  6  end;
  7  /

Function created.
```

```
SQL> declare
  2  num number(2);
  3  begin
  4  num := totalemp();
  5  dbms_output.put_line('total no.of.employee:'|| num);
  6  end;
  7  /
total no.of.employee:4

PL/SQL procedure successfully completed.

SQL>
```

RESULT

Thus, performing Select, retrieving, insert, delete and update operations using PLSQL basics has been done successfully.

EX NO : 13

DATE : 15.05.2024

CURSORS AND LOOPS

AIM

To implement cursors and loops using oracle.

PROCEDURE

STEP 1: Open the notepad.

STEP 2: Type the PLSQL code in the notepad

STEP 3: Declare the necessary variables you want.

STEP 4: Save the file as filname.sql

STEP 5: Go the SQLPlus and Type “GET source_address”

STEP 6: Then Type “SET SERVEROUTPUT ON”

STEP 7: Click enter and comment “/”.

SYNTAX

1. **CASE** Syntax

```
DECLARE
Stmt;
   BEGIN
    Stmt:
    CASE
     Stmt;
   END;
```

CASE

```
  1  --SET VERIFY OFF;
  2  DECLARE
  3  GRADE CHAR(1) := UPPER('&GRADE');
  4  appraisal varchar(20);
  5  BEGIN
  6  appraisal :=
  7  CASE
  8  when GRADE = 'A' THEN 'Excellent'
  9  When GRADE IN ('B', 'C') THEN 'GOOD'
 10  ELSE 'NO SUCH GRADE'
 11  END;
 12  DBMS_OUTPUT.PUT_LINE('GRADE '||GRADE||':: Appraisal '||appraisal);
 13* END;
SQL> /
Enter value for grade: A
old   3: GRADE CHAR(1) := UPPER('&GRADE');
new   3: GRADE CHAR(1) := UPPER('A');

PL/SQL procedure successfully completed.
```

LOOP

```
SQL> get C:\Users\PRIYADHARSHINI\OneDrive\Desktop\eg_loop.sql;
  1  DECLARE
  2    newcity VARCHAR(20) := 'Montreal';
  3    counting NUMBER := 0;
  4    dnum dept_loc_3501.dnumber%TYPE;
  5    dloc dept_loc_3501.dlocation%TYPE;
  6  BEGIN
  7    SELECT MAX(dnumber) INTO dnum FROM dept_loc_3501;
  8    LOOP
  9      counting := counting + 1;
 10      dloc := 'Montreal';
 11      INSERT INTO dept_loc_3501(dnumber, dlocation) VALUES (counting, dloc);
 12      DBMS_OUTPUT.PUT_LINE(counting);
 13      EXIT WHEN counting >= 3;
 14    END LOOP;
 15* END;
 16  /

PL/SQL procedure successfully completed.
```

1. **LOOP** SYNTAX

```
DECLARE
Stmt;
BEGIN
  Stmt
  LOOP
   Stmt
  END LOOP;
END;
```

2. **INSERT BEFORE**

```
CREATE TRIGGER trigger_name BEFORE INSERT ON table_name FOR
EACH ROW
BEGIN
  Stmt;
END;
```

3. **INSERT AFTER**

```
CREATE TRIGGER trigger_name AFTER INSERT ON table_name FOR
EACH ROW
BEGIN
  Stmt;
END;
```

4. **UPDATE BEFORE**

```
CREATE TRIGGER trigger_name BEFORE UPDATE ON table_name FOR
EACH ROW
BEGIN
  Stmt;
END;
```

INSERT BEFORE TRIGGER

```
  1  CREATE OR REPLACE TRIGGER INS_2022503305
  2  BEFORE INSERT ON EMPLOYEE_2022503305
  3  FOR EACH ROW
  4  BEGIN
  5  DBMS_OUTPUT.PUT_LINE('TRIGGER BEFORE!');
  6  EXCEPTION
  7  WHEN OTHERS THEN
  8  DBMS_OUTPUT.PUT_LINE('ERROR OCCURRED: '|| SQLERRM);
  9  END;
SQL> /

Trigger created.
```

```
SQL> INSERT INTO EMPLOYEE_2022503305 (FNAME, LNAME) VALUES ('RON'
    ,'WILSLEY');
```

5. UPDATE AFTER

```
CREATE TRIGGER trigger_name AFTER UPDATE ON table_name FOR
EACH ROW
BEGIN
  Stmt;
END;
```

6. CURSORS

```
DELCARE
  CURSOR cursor_name IS
  Stmt;
  BEGIN
  Stmt;
END;
```

INSERT AFTER TRIGGER

```
 1  CREATE OR REPLACE TRIGGER INS_2022503305
 2  BEFORE INSERT ON EMPLOYEE_2022503305
 3  FOR EACH ROW
 4  BEGIN
 5  DBMS_OUTPUT.PUT_LINE('TRIGGER AFTER ');
 6  EXCEPTION
 7  WHEN OTHERS THEN
 8  DBMS_OUTPUT.PUT_LINE('ERROR OCCURRED: '    SQLERRM);
 9  END;
SQL> /

    Trigger created.
```

```
1  INSERT INTO EMPLOYEE_2022503305 (FNAME, LNAME) VALUES ('REVI', 'SHAREN');
```

UPDATE BEFORE TRIGGER

```
 1  CREATE OR REPLACE TRIGGER INS_2022503305
 2  BEFORE UPDATE ON EMPLOYEE_2022503305
 3  FOR EACH ROW
 4  BEGIN
 5  DBMS_OUTPUT.PUT_LINE('TRIGGER BEFORE ');
 6  EXCEPTION
 7  WHEN OTHERS THEN
 8  DBMS_OUTPUT.PUT_LINE('ERROR OCCURRED: '    SQLERRM);
 9  END;
SQL

    Trigger created.
```

UPDATE AFTER TRIGGER

```
1  CREATE OR REPLACE TRIGGER INS_2022503305
2  AFTER UPDATE ON EMPLOYEE_2022503305
3  FOR EACH ROW
4  BEGIN
5  DBMS_OUTPUT.PUT_LINE('TRIGGER BEFORE ');
6  EXCEPTION
7  WHEN OTHERS THEN
8  DBMS_OUTPUT.PUT_LINE('ERROR OCCURRED: '   SQLERRM);
9  END;
SQL

    Trigger created.
```

USING OLD AND NEW TRIGGER KEYWORD

```
1  CREATE OR REPLACE TRIGGER INS_2022503305
2  AFTER UPDATE ON EMPLOYEE_2022503305
3  FOR EACH ROW
4  BEGIN
5  DBMS_OUTPUT.PUT_LINE('TRIGGER AFTER ');
6  EXCEPTION
7  WHEN OTHERS THEN
8  DBMS_OUTPUT.PUT_LINE('OLD SALARY: '    :OLD.salary);
9  DBMS_OUTPUT.PUT_LINE('NEW SALARY: '    :NEW.salary);
10 EXCEPTION
11 WHEN OTHERS THEN
12 DBMS_OUTPUT.PUT_LINE('ERROR OCCURED: '    SQLERRM);
13 END;
SQL

    Trigger created.
```

CURSOR

```
 1  DECLARE
 2     CURSOR EMP_CURSOR IS
 3     SELECT SSN, LNAME FROM EMPLOYEE_2022503305 WHERE DNO    ;
 4     SSN EMPLOYEE_202250335.SSN TYPE;
 5     LAST_NAME EMPLOYEE_2022503305.LNAME TYPE;
 6     BEGIN
 7     OPEN EMP_CURSOR;
 8     FETCH EMP_CURSOR INTO SSN, LAST_NAME;
 9     DBMS_OUTPUT.PUT_LINE(SSN         LAST_NAME);
10     END;
SQL
PL SQL procedure successfully completed
```

RESULT

Thus, implementing cursors and loops using oracle has been done successfully.