**DATA DEFINITION LANGUAGE (DDL) COMMANDS IN RDBMS**

**AIM:**

To execute and verify the Data Definition Language commands and constraints

**DDL** (DATA DEFINITION LANGUAGE)

* CREATE
* ALTER
* DROP
* TRUNCATE
* COMMENT
* RENAME

**PROCEDURE**

STEP 1: Start

STEP 2: Create the table with its essential attributes.

STEP 3: Execute different Commands and extract information from the table.

STEP 4: Stop

**SQL COMMANDS**

1. COMMAND NAME: **CREATE**

COMMAND DESCRIPTION: **CREATE** command is used to create objects in the database.

2. COMMAND NAME: **DROP**

COMMAND DESCRIPTION: **DROP** command is used to delete the object from the database.

3. COMMAND NAME: **TRUNCATE**

COMMAND DESCRIPTION: **TRUNCATE** command is used to remove all the records from the table.

4. COMMAND NAME: **ALTER**

COMMAND DESCRIPTION: **ALTER** command is used to alter the structure of database.

5. COMMAND NAME: **RENAME**

COMMAND DESCRIPTION: **RENAME** command is used to rename the objects.

**QUERY: 01**

Q1. Write a query to create a table employee with empno, ename, designation, and salary.

**Syntax for creating a table:**

**SQL: CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1**

**<DATATYPE> (SIZE), COLUMN NAME.1 <DATATYPE> (SIZE)**

**……………………………);**

**QUERY: 01**

**SQL>CREATE TABLE EMP (EMPNO NUMBER (4),**

**ENAME VARCHAR2 (10),**

**DESIGNATIN VARCHAR2 (10),**

**SALARY NUMBER (8,2));**

**Table created.**

**QUERY: 02**

Q2. Write a query to display the column name and datatype of the table employee.

**Syntax for describe the table:**

**SQL: DESC <TABLE NAME>;**

**SQL> DESC EMP;**

**Name Null? Type**

**-------------------------------------- -------- -------------**

**EMPNO NUMBER(4)**

**ENAME VARCHAR2(10)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**QUERY: 03**

Q3. Write a query for create a from an existing table with all the fields

**Syntax For Create A from An Existing Table With All Fields**

**SQL> CREATE TABLE <TRAGET TABLE NAME> SELECT \* FROM**

**<SOURCE TABLE NAME>;**

**QUERY: 03**

**SQL> CREATE TABLE EMP1 AS SELECT \* FROM EMP;**

**Table created.**

**SQL> DESC EMP1**

**Name Null? Type**

**------------------------------------- -------- ------------------**

**EMPNO NUMBER(4)**

**ENAME VARCHAR2(10)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**QUERY: 04**

Q4. Write a query for create a from an existing table with selected fields

**Syntax For Create A from An Existing Table With Selected Fields**

**SQL> CREATE TABLE <TRAGET TABLE NAME> SELECT EMPNO, ENAME**

**FROM <SOURCE TABLE NAME>;**

**QUERY: 04**

**SQL> CREATE TABLE EMP2 AS SELECT EMPNO, ENAME FROM EMP;**

**Table created.**

**SQL> DESC EMP2**

**Name Null? Type**

**------------------------------------- -------- ----------------------**

**EMPNO NUMBER (4)**

**ENAME VARCHAR2 (10)**

**QUERY: 05**

Q5. Write a query for create a new table from an existing table without any record:

**Syntax for create a new table from an existing table without any record:**

**SQL> CREATE TABLE <TRAGET TABLE NAME> AS SELECT \* FROM**

**<SOURCE TABLE NAME> WHERE <FALSE CONDITION>;**

**QUERY: 05**

**SQL> CREATE TABLE EMP3 AS SELECT \* FROM EMP WHERE 1>2;**

**Table created.**

**SQL> DESC EMP3;**

**Name Null? Type**

**----------------------------------------- -------- ------------------------**

**EMPNO NUMBER(4)**

**ENAME VARCHAR2(10)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2);**

**ALTER & MODIFICATION ON TABLE**

**QUERY: 06**

Q6. Write a Query to Alter the column EMPNO NUMBER (4) TO EMPNO NUMBER (6).

**Syntax for Alter & Modify on a Single Column:**

**SQL > ALTER <TABLE NAME> MODIFY <COLUMN NAME> <DATATYPE> (SIZE);**

**QUERY: 06**

**SQL>ALTER TABLE EMP MODIFY EMPNO NUMBER (6);**

**Table altered.**

**SQL> DESC EMP;**

**Name Null? Type**

**----------------------------------------- -------- ----------------------------**

**EMPNO NUMBER(6)**

**ENAME VARCHAR2(10)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**QUERY: 07**

Q7. Write a Query to Alter the table employee with multiple columns (EMPNO, ENAME.)

**Syntax for alter table with multiple column:**

**SQL > ALTER <TABLE NAME> MODIFY <COLUMN NAME1> <DATATYPE>**

**(SIZE), MODIFY <COLUMN NAME2> <DATATYPE> (SIZE)**

**………………………………………….;**

**QUERY: 07**

**SQL>ALTER TABLE EMP MODIFY (EMPNO NUMBER (7), ENAME VARCHAR2(12));**

**Table altered.**

**SQL> DESC EMP;**

**Name Null? Type**

**----------------------------------------- -------- ----------------------------**

**EMPNO NUMBER(7)**

**ENAME VARCHAR2(12)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2);**

**QUERY: 08**

Q8. Write a query to add a new column in to employee

**Syntax for add a new column:**

**SQL> ALTER TABLE <TABLE NAME> ADD (<COLUMN NAME> <DATA**

**TYPE> <SIZE>);**

**QUERY: 08**

**SQL> ALTER TABLE EMP ADD QUALIFICATION VARCHAR2(6);**

**Table altered.**

**SQL> DESC EMP;**

**Name Null? Type**

**----------------------------------------- -------- ----------------------------**

**EMPNO NUMBER(7)**

**ENAME VARCHAR2(12)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**QUALIFICATION VARCHAR2(6)**

**QUERY: 09**

Q9. Write a query to add multiple columns in to employee

**Syntax for add a new column:**

**SQL> ALTER TABLE <TABLE NAME> ADD (<COLUMN NAME1> <DATA**

**TYPE> <SIZE>,(<COLUMN NAME2> <DATA TYPE> <SIZE>,**

**………………………………………………………………);**

**QUERY: 09**

**SQL>ALTER TABLE EMP ADD (DOB DATE, DOJ DATE);**

**Table altered.**

**SQL> DESC EMP;**

**Name Null? Type**

**----------------------------------------- -------- ----------------------------**

**EMPNO NUMBER(7)**

**ENAME VARCHAR2(12)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**QUALIFICATION VARCHAR2(6)**

**DOB DATE**

**DOJ DATE**

**REMOVE / DROP**

**QUERY: 10**

Q10. Write a query to drop a column from an existing table employee

**Syntax for add a new column:**

**SQL> ALTER TABLE <TABLE NAME> DROP COLUMN <COLUMN NAME>;**

**QUERY: 10**

**SQL> ALTER TABLE EMP DROP COLUMN DOJ;**

**Table altered.**

**SQL> DESC EMP;**

**Name Null? Type**

**----------------------------------------- -------- -------------**

**EMPNO NUMBER(7)**

**ENAME VARCHAR2(12)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**QUALIFICATION VARCHAR2(6)**

**DOB DATE**

**QUERY: 11**

Q10. Write a query to drop multiple columns from employee

**Syntax for add a new column:**

**SQL> ALTER TABLE <TABLE NAME> DROP <COLUMN**

**NAME1>,<COLUMN NAME2>,…………………………….. ;**

**QUERY: 11**

**SQL> ALTER TABLE EMP DROP (DOB, QUALIFICATION);**

**Table altered.**

**SQL> DESC EMP;**

**Name Null? Type**

**----------------------------------------- -------- --------------**

**EMPNO NUMBER(7)**

**ENAME VARCHAR2(12)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**REMOVE**

**QUERY: 12**

Q10. Write a query to rename table emp to employee

**Syntax for add a new column:**

**SQL> ALTER TABLE RENAME <OLD NAME> TO <NEW NAME>**

**QUERY: 12**

**SQL> ALTER TABLE EMP RENAME EMP TO EMPLOYEE;**

**SQL> DESC EMPLOYEE;**

**Name Null? Type**

**----------------------------------------- -------- --------------**

**EMPNO NUMBER(7)**

**ENAME VARCHAR2(12)**

**DESIGNATIN VARCHAR2(10)**

**SALARY NUMBER(8,2)**

**CONSTRAINTS**

Constraints are part of the table definition that limits and restriction on the value entered into its columns.

**TYPES OF CONSTRAINTS:**

1) Primary key

2) Foreign key/references

3) Check

4) Unique

5) Not null

6) Null

7) Default

**CONSTRAINTS CAN BE CREATED IN THREE WAYS:**

1) Column level constraints

2) Table level constraints

3) Using DDL statements-alter table command

**OPERATION ON CONSTRAINT**:

i) ENABLE

ii) DISABLE

iii) DROP

**Column level constraints Using Primary key**

Q13. Write a query to create primary constraints with column level

**Primary key**

**Syntax for Column level constraints Using Primary key:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE)<TYPE OF CONSTRAINTS> , COLUMN NAME.1 <DATATYPE> (SIZE)

……………………………);

**QUERY:13**

SQL>CREATE TABLE EMPLOYEE(EMPNO NUMBER(4) **PRIMARY**

**KEY,**

ENAME VARCHAR2(10),

JOB VARCHAR2(6),

SAL NUMBER(5),

DEPTNO NUMBER(7));

**Column level constraints Using Primary key with naming convention**

Q14. Write a query to create primary constraints with column level with naming convention

**Syntax for Column level constraints Using Primary key:**

SQL: >CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE)CONSTRAINTS <NAME OF THE CONSTRAINTS> <TYPE OF THE

CONSTRAINTS> , COLUMN NAME.1 <DATATYPE> (SIZE)

……………………………);

**QUERY:14**

SQL>CREATE TABLE EMPLOYEE(EMPNO NUMBER(4)

**CONSTRAINT EMP\_EMPNO\_PK PRIMARY KEY,**

ENAME VARCHAR2(10),

JOB VARCHAR2(6),

SAL NUMBER(5),

DEPTNO NUMBER(7));

**Table Level Primary Key Constraints**

Q15. Write a query to create primary constraints with table level with naming convention

**Syntax for Table level constraints Using Primary key:**

SQL: >CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE) , COLUMN NAME.1 <DATATYPE> (SIZE), CONSTRAINTS <NAME OF

THE CONSTRAINTS> <TYPE OF THE CONSTRAINTS>);

**QUERY: 15**

SQL>CREATE TABLE EMPLOYEE (EMPNO NUMBER(6),

ENAME VARCHAR2(20),

JOB VARCHAR2(6),

SAL NUMBER(7),

DEPTNO NUMBER(5),

**CONSTRAINT EMP\_EMPNO\_PK PRIMARY**

**KEY(EMPNO));**

**Table level constraint with alter command (primary key):**

Q16. Write a query to create primary constraints with alter command

**Syntax for Column level constraints Using Primary key:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE), COLUMN NAME.1 <DATATYPE> (SIZE) );

SQL> ALTER TABLE <TABLE NAME> ADD CONSTRAINTS <NAME OF THE

CONSTRAINTS> <TYPE OF THE CONSTRAINTS> <COLUMN NAME>);

**QUERY: 16**

SQL>CREATE TABLE EMPLOYEE(EMPNO NUMBER(5),

ENAME VARCHAR2(6),

JOB VARCHAR2(6),

SAL NUMBER(6),

DEPTNO NUMBER(6));

SQL>ALTER TABLE EMP3 ADD CONSTRAINT **EMP3\_EMPNO\_PK PRIMARY**

**KEY (EMPNO);**

**Reference /foreign key constraint**

**Column level foreign key constraint:**

Q.17. Write a query to create foreign key constraints with column level

**Parent Table:**

**Syntax for Column level constraints Using Primary key:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE)<TYPE OF CONSTRAINTS> , COLUMN NAME.1 <DATATYPE> (SIZE)

……………………………);

**Child Table:**

**Syntax for Column level constraints Using foreign key:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE), COLUMN NAME2 <DATATYPE> (SIZE) REFERENCES <TABLE NAME>

(COLUMN NAME> ……………………………);

**QUERY: 17**

SQL>CREATE TABLE DEPT(DEPTNO NUMBER(2) PRIMARY

KEY,

DNAME VARCHAR2(20),

LOCATION VARCHAR2(15));

SQL>CREATE TABLE EMP4

(EMPNO NUMBER(3),

DEPTNO NUMBER(2) **REFERENCES DEPT(DEPTNO),**

DESIGN VARCHAR2(10));

**Column level foreign key constraint with naming conversions:**

**Parent Table:**

**Syntax for Column level constraints Using Primary key:**

Q.18. Write a query to create foreign key constraints with column level

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE)<TYPE OF CONSTRAINTS> , COLUMN NAME.1 <DATATYPE> (SIZE)

……………………………);

**Child Table:**

**Syntax for Column level constraints using foreign key:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE) , COLUMN NAME2 <DATATYPE> (SIZE) **CONSTRAINT <CONST.**

**NAME>** REFERENCES <TABLE NAME> (COLUMN NAME>

……………………………);

**QUERY:18**

SQL>CREATE TABLE DEPT(DEPTNO NUMBER(2) PRIMARY KEY,

DNAME VARCHAR2(20),

LOCATION VARCHAR2(15));

SQL>CREATE TABLE EMP4A

(EMPNO NUMBER(3),

DEPTNO NUMBER(2)**CONSTRAINT EMP4A\_DEPTNO\_FK**

**REFERENCES DEPT(DEPTNO),**

DESIGN VARCHAR2(10));

**Table Level Foreign Key Constraints**

Q.19. Write a query to create foreign key constraints with Table level

**Parent Table:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE)<TYPE OF CONSTRAINTS> , COLUMN NAME.1 <DATATYPE> (SIZE)

……………………………);

**Child Table:**

**Syntax for Table level constraints using foreign key:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE), COLUMN NAME2 <DATATYPE> (SIZE), **CONSTRAINT <CONST.**

**NAME>** REFERENCES <TABLE NAME> (COLUMN NAME> );

**QUERY: 19**

SQL>CREATE TABLE DEPT

(DEPTNO NUMBER(2) PRIMARY KEY,

DNAME VARCHAR2(20),

LOCATION VARCHAR2(15));

SQL>CREATE TABLE EMP5

(EMPNO NUMBER(3),

DEPTNO NUMBER(2),

DESIGN VARCHAR2(10)**CONSTRAINT ENP2\_DEPTNO\_FK FOREIGN**

**KEY(DEPT NO)REFERENCESDEPT(DEPTNO));**

**Table Level Foreign Key Constraints with Alter command**

Q.20. Write a query to create foreign key constraints with Table level with alter

command.

**Parent Table:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE)<TYPE OF CONSTRAINTS> , COLUMN NAME.1 <DATATYPE> (SIZE)

……………………………);

**Child Table:**

**Syntax for Table level constraints using foreign key:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE) , COLUMN NAME2 <DATATYPE> (SIZE));

SQL> ALTER TABLE <TABLE NAME> ADD CONSTRAINT <CONST. NAME>

REFERENCES <TABLE NAME> (COLUMN NAME>);

**QUERY:20**

SQL>CREATE TABLE DEPT

(DEPTNO NUMBER(2) PRIMARY KEY,

DNAME VARCHAR2(20),

LOCATION VARCHAR2(15));

SQL>CREATE TABLE EMP5

(EMPNO NUMBER(3),

DEPTNO NUMBER(2),

DESIGN VARCHAR2(10));

SQL>ALTER TABLE EMP6 ADD CONSTRAINT EMP6\_DEPTNO\_FK FOREIGN

KEY(DEPTNO)REFERENCES DEPT(DEPTNO);

**Check constraint**

**Column Level Check Constraint**

Q.21. Write a query to create Check constraints with column level

**Syntax for column level constraints using Check:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE) CONSTRAINT <CONSTRAINTS NAME> <TYPE OF CONSTRAINTS>

(CONSTRAITNS CRITERIA) , COLUMN NAME2 <DATATYPE> (SIZE));

**QUERY:21**

SQL>CREATE TABLE EMP7(EMPNO NUMBER(3),

ENAME VARCHAR2(20),

DESIGN VARCHAR2(15),

SAL NUMBER(5)CONSTRAINT EMP7\_SAL\_CK CHECK(SAL>500 AND

SAL<10001),

DEPTNO NUMBER(2));

**Table Level Check Constraint:**

Q.22. Write a query to create Check constraints with table level

**Syntax for Table level constraints using Check:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE), (COLUMN NAME2 <DATATYPE> (SIZE), CONSTRAINT

<CONSTRAINTS NAME> <TYPE OF CONSTRAINTS> (CONSTRAITNS

CRITERIA)) ;

**QUERY:22**

SQL>CREATE TABLE EMP8(EMPNO NUMBER(3),

ENAME VARCHAR2(20),

DESIGN VARCHAR2(15),

SAL NUMBER(5),DEPTNO NUMBER(2),

CONSTRAINTS EMP8\_SAL\_CK CHECK(SAL>500 AND

SAL<10001));

**Check Constraint with Alter Command**

Q.23. Write a query to create Check constraints with table level using alter command.

**Syntax for Table level constraints using Check:**

SQL:>CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1 <DATATYPE>

(SIZE), (COLUMN NAME2 <DATATYPE> (SIZE), CONSTRAINT

<CONSTRAINTS NAME> <TYPE OF CONSTRAINTS> (CONSTRAITNS

CRITERIA)) ;

**QUERY:23**

SQL>CREATE TABLE EMP9(EMPNO NUMBER,

ENAME VARCHAR2(20),

DESIGN VARCHAR2(15),

SAL NUMBER(5));

SQL>ALTER TABLE EMP9 ADD CONSTRAINTS EMP9\_SAL\_CK

CHECK(SAL>500 AND SAL<10001);

**Unique Constraint**

**Column Level Constraint**

Q.24. Write a query to create unique constraints with column level

**Syntax for Column level constraints with Unique:**

SQL :> CREATE <OBJ.TYPE> <OBJ.NAME> (<COLUMN NAME.1>

<DATATYPE> (SIZE) CONSTRAINT <NAME OF CONSTRAINTS>

<CONSTRAINT TYPE>, (COLUMN NAME2 <DATATYPE> (SIZE)) ;

**QUERY:24**

SQL>CREATE TABLE EMP10(EMPNO NUMBER(3),

ENAME VARCHAR2(20),

DESGIN VARCHAR2(15)CONSTRAINT EMP10\_DESIGN\_UK UNIQUE,

SAL NUMBER(5));

**Table Level Constraint**

Q.25. Write a query to create unique constraints with table level

**Syntax for Table level constraints with Unique:**

SQL :> CREATE <OBJ.TYPE> <OBJ.NAME> (<COLUMN NAME.1>

<DATATYPE> (SIZE), (COLUMN NAME2 <DATATYPE> (SIZE), CONSTRAINT

<NAME OF CONSTRAINTS> <CONSTRAINT TYPE>(COLUMN NAME);) ;

**QUERY:25**

SQL>CREATE TABLE EMP11(EMPNO NUMBER(3),

ENAME VARCHAR2(20),

DESIGN VARCHAR2(15),

SAL NUMBER(5),CONSTRAINT EMP11\_DESIGN\_UK UNIGUE(DESIGN));

**Table Level Constraint Alter Command**

Q.26. Write a query to create unique constraints with table level

**Syntax for Table level constraints with Check Using Alter**

SQL :> CREATE <OBJ.TYPE> <OBJ.NAME> (<COLUMN NAME.1>

<DATATYPE> (SIZE), (COLUMN NAME2 <DATATYPE> (SIZE)) ;

SQL> ALTER TABLE ADD <CONSTRAINTS> <CONSTRAINTS NAME>

<CONSTRAINTS TYPE>(COLUMN NAME);

**QUERY:26**

SQL>CREATE TABLE EMP12

(EMPNO NUMBER(3),

ENAME VARCHAR2(20),

DESIGN VARCHAR2(15),

SAL NUMBER(5));

SQL>ALTER TABLE EMP12 ADD CONSTRAINT EMP12\_DESIGN\_UK

UNIQUE(DESING);

**Not Null**

**Column Level Constraint**

Q.27. Write a query to create Not Null constraints with column level

**Syntax for Column level constraints with Not Null:**

SQL :> CREATE <OBJ.TYPE> <OBJ.NAME> (<COLUMN NAME.1>

<DATATYPE> (SIZE) CONSTRAINT <NAME OF CONSTRAINTS>

<CONSTRAINT TYPE>, (COLUMN NAME2 <DATATYPE> (SIZE)) ;

**QUERY: 27**

SQL>CREATE TABLE EMP13

(EMPNO NUMBER(4),

ENAME VARCHAR2(20) CONSTRAINT EMP13\_ENAME\_NN NOT NULL,

DESIGN VARCHAR2(20),

SAL NUMBER(3));

**Null**

**Column Level Constraint**

Q.28. Write a query to create Null constraints with column level

**Syntax for Column level constraints with Null:**

SQL :> CREATE <OBJ.TYPE> <OBJ.NAME> (<COLUMN NAME.1>

<DATATYPE> (SIZE) CONSTRAINT <NAME OF CONSTRAINTS>

<CONSTRAINT TYPE>, (COLUMN NAME2 <DATATYPE> (SIZE)) ;

**QUERY:28**

SQL>CREATE TABLE EMP13

(EMPNO NUMBER(4),

ENAME VARCHAR2(20) CONSTRAINT EMP13\_ENAME\_NN NULL,

DESIGN VARCHAR2(20),

SAL NUMBER(3));

**Constraint Disable \ Enable**

**Constraint Disable**

Q.29. Write a query to disable the constraints

**Syntax for disabling a single constraint in a table:**

SQL>ALTER TABLE <TABLE-NAME> DISABLE CONSTRAINT <CONSTRAINTNAME>

**QUERY:29**

SQL>ALTER TABLE EMP13 DISABLE CONSTRAINT EMP13\_ENAME\_NN

NULL;

**Constraint Enable**

Q.30. Write a query to enable the constraints

**Syntax for disabling a single constraint in a table:**

SQL>ALTER TABLE <TABLE-NAME> DISABLE CONSTRAINT <CONSTRAINTNAME>

**QUERY:30**

SQL>ALTER TABLE EMP13 ENABLE CONSTRAINT EMP13\_ENAME\_NN

NULL;