

## **Lab Assignment -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> (CONSTRITNS  
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> (CONSTRITNS 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> (CONSTRAINTS 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),  
DESIGN 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 <CONSTRAINT-NAME>

**Constraint Enable**

**QUERY:29**

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

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 <CONSTRAINT-NAME>

**QUERY:30**

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