

# **Lab Session 03**

## ***(Data Definition Language (DDL))***

## **Data Definition Language (DDL):**

Data Definition Language (DDL) statements are used to define the database structure or schema.

- **CREATE** - to create objects in the database
- **ALTER** - alters the structure of the database
- **DROP** - delete objects from the database
- **TRUNCATE** - remove all records from a table, including all spaces allocated for the records are removed
- **RENAME** - rename an object

### **Creating a Table:**

#### **Syntax**

```
CREATE TABLE TABLE_NAME  
(Column_name1    data_type [DEFAULT value],  
Column_name2    data_type [DEFAULT value] ,  
Column_name3    data_type [DEFAULT value], .....)
```

#### **Example**

```
CREATE TABLE customer  
(Cust_id NUMBER (2),  
LastName VARCHAR2 (14),  
FirstName VARCHAR2 (14),  
Address VARCHAR2 (20),  
Telno NUMBER (20));
```

### **Create Table from Another Table:**

#### **Syntax:**

```
CREATE TABLE table_name[colname1,colname2] AS [subquery];
```

#### **Example:**

The following example creates a table, DEPT30, that contains details of all employees working in department 30

```
CREATE TABLE dept30 AS SELECT empno, ename, sal * 12 ANNSAL, hiredate FROM emp  
WHERE deptno = 30;
```

## Constraints in DDL:

Constraints are certain restrictions over different tables/columns in a Database. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

1. Not Null
2. Unique
3. Primary Key
4. Foreign Key
5. Check
6. Default

### 1. Not Null:

The NOT NULL constraint enforces a column to NOT accept NULL values. This means that NOT NULL constraint enforces a field to always contain a value.

#### Example:

```
CREATE TABLE customer (Cust_id NUMBER(2) NOT NULL,  
LastName VARCHAR2(14),  
FirstName VARCHAR2(14) NOT NULL,  
Address VARCHAR2(20),  
Telno NUMBER(20));
```

### 2.Unique:

The UNIQUE constraint uniquely identifies each record in a database table. The **UNIQUE** and **PRIMARY KEY** constraints both provide a guarantee for uniqueness for a column or set of columns.

#### Example:

```
CREATE TABLE customer  
(Cust_id NUMBER(2) UNIQUE,  
LastName VARCHAR2(14),  
FirstName VARCHAR2(14) NOT NULL,  
Address VARCHAR2(20),  
Telno NUMBER(20) UNIQUE);
```

### 3. Primary Key Constraint

The PRIMARY KEY constraint uniquely identifies each record in a database table.

**Example:**

```
CREATE TABLE customer
(Cust_id NUMBER(2) PRIMARY key ,
LastName VARCHAR2(14),
FirstName VARCHAR2(14) NOT NULL,
Address VARCHAR2(20),
Telno NUMBER(20));
```

#### **4. Foreign Key Constraint**

A FOREIGN KEY in one table points to a PRIMARY KEY in another table. The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. The FOREIGN KEY constraint also prevents that invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

**Example:**

```
CREATE TABLE EMP3
(
EMPNO NUMBER(4) PRIMARY KEY,
DEPTNO NUMBER(7,2) NOT NULL,
ENAME VARCHAR2(9) NOT NULL,
CONSTRAINT FK_EMP_DEPTNO FOREIGN KEY (DEPTNO)
REFERENCES DEPT(DEPTNO)
);
```

#### **5. Check Constraint**

Specifies a condition that must be true.

**Example:**

```
CREATE TABLE EMP3
( EMPNO NUMBER (4) NOT NULL CHECK(EMPNO>0) PRIMARY KEY,
DEPTNO NUMBER(7,2) NOT NULL,
ENAME VARCHAR2(9) NOT NULL,
CONSTRAINT FK_EMP_DEPTNO FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO));
```

#### **6.Default Constraint**

The DEFAULT constraint is used to insert a default value into a column. The default value will be added to all new records, if no other value is specified.

**Example:**

```
CREATE TABLE EMP3  
(EMPNO NUMBER(4) NOT NULL,  
DEPTNO NUMBER(7,2) NOT NULL,  
ENAME VARCHAR2(9) NOT NULL,  
HIRE_DATE DATE DEFAULT SYSDATE);
```

**Naming a Constraint:**

```
(Cust_id NUMBER(2) NOT NULL ,  
LastName VARCHAR2(14),  
FirstName VARCHAR2(14) NOT NULL,  
Address VARCHAR2(20),  
Telno NUMBER(20) ,  
CONSTRAINT uc_CidTelnum UNIQUE (Cust_id,telno));
```

**Adding Constraint in an Existing Table:**

Example:

1. ALTER TABLE Customer  
ADD UNIQUE (FirstName);
2. ALTER TABLE EMP3  
MODIFY hire\_date DEFAULT (sysdate)

**Dropping Constraint:**

Example:

1. ALTER TABLE Customer  
DROP CONSTRAINT uc\_CidTelnum;
2. ALTER TABLE EMP3  
ALTER COLUMN HIRE\_DATE DROP DEFAULT

**Detailed Example**

```
CREATE TABLE DEPT (  
DEPTNO      NUMBER(2) constraint DEPT_DEPTNO_PK PRIMARY KEY,  
DNAME       VARCHAR2(14),  
LOC         VARCHAR2(13),  
CONSTRAINT DEPT_DNAME_UK  UNIQUE(DNAME));
```

```
CREATE TABLE EMP3 (  
EMPNO NUMBER(4) CONSTRAINT EMP_EMPNO_PK PRIMARY KEY,  
ENAME VARCHAR2(10) NOT NULL,  
JOB VARCHAR2(9),  
MGR NUMBER(4),  
HIREDATE DATE DEFAULT SYSDATE,  
SAL NUMBER(7, 2),  
COMM NUMBER(7, 2),  
DEPTNO NUMBER(7, 2) NOT NULL,  
CONSTRAINT EMP_DEPTNO_CK CHECK (DEPTNO BETWEEN 1 AND 50),  
CONSTRAINT EMP_DEPTNO_FK FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO));
```

## **Drop Statement**

- The DROP TABLE statement is used to delete a table.

### **Syntax:**

```
DROP TABLE table_name;
```

- The DROP DATABASE statement is used to delete a database.

### **Syntax:**

```
DROP DATABASE database_name;
```

## **Truncate Table Statement**

If we just want to delete the data and not the table itself we use Truncate statement

### **Syntax**

```
TRUNCATE TABLE table_name
```

## **Alter Table Statement**

- **TO ADD A COLUMN IN A TABLE**
  - ALTER TABLE table\_name  
ADD column\_name datatype
- **TO DELETE A COLUMN IN A TABLE**
  - ALTER TABLE table\_name  
DROP COLUMN column\_name
- **TO CHANGE THE DATA TYPE OF A COLUMN IN A TABLE**

- ALTER TABLE table\_name  
MODIFY column\_name datatype

## **Referential Integrity:**

Referential integrity is a property of data which, when satisfied, requires every value of one attribute (column) of a relation (table) to exist as a value of another attribute in a different (or the same) relation (table). It is imposed through Primary and Foreign Key Constraints. It is violated through Delete and Update statements.

### **Deferred Constraint Checking:**

Sometimes it is necessary to defer the checking of certain constraints, most commonly in the "chicken-and-egg" problem. Suppose we want to say:

```
CREATE TABLE chicken (cID INT PRIMARY KEY, eID INT REFERENCES egg(eID));
```

```
CREATE TABLE egg(eID INT PRIMARY KEY, cID INT REFERENCES chicken(cID));
```

But if we simply type the above statements into Oracle, we'll get an error. The reason is that the CREATE TABLE statement for chicken refers to table egg, which hasn't been created yet! Creating egg won't help either, because egg refers to chicken.

### **Getting Rid of the Problem:**

To work around this problem, we need SQL schema modification commands. First, create chicken and egg without foreign key declarations:

```
CREATE TABLE chicken(cID INT PRIMARY KEY, eID INT);
```

```
CREATE TABLE egg(eID INT PRIMARY KEY, cID INT);
```

Then, we add foreign key constraints:

```
ALTER TABLE chicken ADD CONSTRAINT chickenREFegg FOREIGN KEY (eID) REFERENCES  
egg(eID) INITIALLY DEFERRED DEFERRABLE;
```

```
ALTER TABLE egg ADD CONSTRAINT eggREFchicken FOREIGN KEY (cID) REFERENCES  
chicken(cID)
```

```
INITIALLY DEFERRED DEFERRABLE;
```

### **Dropping the Tables**

Finally, to get rid of the tables, we have to drop the constraints first, because Oracle won't allow us to drop a table that's referenced by another table.

```
ALTER TABLE egg DROP CONSTRAINT eggREFchicken;
```

```
ALTER TABLE chicken DROP CONSTRAINT chickenREFegg;
```

```
DROP TABLE egg;
```

```
DROP TABLE chicken;
```

## Exercise – Scenario 1

### Step: 01

Create a table Faculty based on the following chart

Column	Data type	Constraints
Faculty_Id	Number (6)	Primary Key => faculty_pk
Last_Name	Varchar2(15)	Not NULL
First_Name	Varchar2(15)	Not NULL
Dept	Char(3)	

### Step: 02

Create a table Department based on the following chart

Column	Data type	Constraints
Dept_Code	Char (3)	Primary Key => dept_pk
Dept_Name	Varchar2(20)	Not NULL



**Step: 03**

Add a new column Location to Dept table which has data type Char(7). Save the SQL statement as ex3.sql. Confirm and validate the modification of the table.

**Step: 04**

Add a new FOREIGN KEY constraint (named faculty\_dept\_fk) to column Dept on Faculty table that refers to column Dept\_Code on Dept table.

**Step: 05**

Increase Last\_Name column to 25 characters long.

\*\*\*\*\*

**Scenario 2**

Create the following library database using SQL

