



# **Specifying Constraints in SQL**

Dr. Narjis Mezaal Shati

# CONSTRAINTS



In attribute-level  
Or column-level

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

In table-level  
Or relation-level

- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

Figure1: constraints in SQL

# □ NOT NULL

Because SQL allows Nulls as attribute values, a constraint NOT NULL specified if Null is not permitted for a particular attribute.

This is implicitly specified for the attributes that are part of the primary key, but it can be specified for any other attributes whose values are required not to be Null.

It is also possible to define a default value for an attribute definition.

# □ NOT NULL

## EXAMPLES:

DEPT

DeptNo	Dname	Loc
--------	-------	-----

1. Create table DEPT with NOT NULL constraint.

```
CREATE TABLE DEPT
```

```
(DeptNo INT(2),
```

```
Dname VARCHAR (14) NOT NULL,
```

```
Loc VARCHAR (14));
```

# □ NOT NULL

## EXAMPLES:

2. Create table DEPT with NOT NULL constraint with default value.

```
CREATE TABLE DEPT
```

```
(DeptNo INT(2),
```

```
Dname VARCHAR (14) NOT NULL DEFAULT 'Ali',
```

```
Loc VARCHAR (13));
```

# □ CHECK

Use to restrict attribute or domain values. For example, suppose that department numbers are restricted to integer numbers between 1 and 20; so we can change the attribute declaration of DeptNo to the following:

```
DeptNo INT CHECK (DeptNo >0 and  
DeptNo < 21)
```

# □ CHECK

EXAMPLE: DEPT

DeptNo	Dname	Loc
--------	-------	-----

Create table DEPT with CHECK constraint on DeptNo in table-level.

```
CREATE TABLE DEPT
  (DeptNo INT(2),
   Dname VARCHAR (14),
   Loc VARCHAR (14),
   CONSTRAINT DeptDeptNoCK CHECK(
    DeptNo BETWEEN 10 AND 99));
```

# □ UNIQUE

Specifies alternate (secondary) keys, for example:

Dname VARCHAR (15) UNIQUE

## EXAMPLE:

DEPT

DeptNo	Dname	Loc
--------	-------	-----

Create table DEPT with **UNIQUE** constraint in table-level.



# □ UNIQUE

```
CREATE TABLE DEPT
```

```
(DeptNo INT(2),
```

```
Dname VARCHAR (14),
```

```
Loc VARCHAR (13),
```

```
CONSTRAINT DeptDNameUQ UNIQUE(DName));
```

# □ PRIMARY KEY

Specifies one or more attributes that make up the primary key of a relation.

**EXAMPLES:** DEPT

<u>DeptNo</u>	Dname	Loc
---------------	-------	-----

1. Create table DEPT with PRIMARY KEY constraint in column-level.

```
CREATE TABLE DEPT  
(DeptNo INT(2) PRIMARY KEY,  
Dname VARCHAR (14),  
Loc VARCHAR (13));
```

# □ PRIMARY KEY

2. Create table DEPT with PRIMARY KEY constraint in table-level.

```
CREATE TABLE DEPT  
(DeptNo INT(2),  
Dname VARCHAR (14),  
Loc VARCHAR (13),  
CONSTRAINT Dept_DeptNo_PK  
PRIMARY KEY(DeptNo));
```

# □ FOREIGN KEY

Use to Specify referential integrity.

## EXAMPLES:

EMP

<u>EmpNo</u>	EName	Job	Mgr	Hiredate	Sal	Comm	DeptNo
--------------	-------	-----	-----	----------	-----	------	--------

DEPT

<u>DeptNo</u>	Dname	Loc
---------------	-------	-----

1. Create table EMP with FOREIGN KEY constraint on DeptNo column in column-level.

# □ FOREIGN KEY

CREATE TABLE EMP

(EmpNo INT (4),

Ename VARCHAR(10),

Job VARCHAR(9),

Mgr INT(4),

Hiredate DATE,

Sal DEC(7,2),

Comm DEC (7,2),

DeptNo INT(2) REFERENCES Dept(DeptNo));

# □ FOREIGN KEY

2. Create table EMP with FOREIGN KEY constraint on DeptNo column in table-level.

```
CREATE TABLE EMP  
(EmpNo INT (4),  
  Ename VARCHAR(10),  
  Job VARCHAR(9),  
  Mgr INT(4),  
  Hiredate DATE,  
  Sal DEC(7,2),  
  Comm DEC (7,2),  
  DeptNo INT(2),  
  CONSTRAINT Emp_DeptNo_FK FOREIGN KEY  
  (DeptNo) REFERENCES Dept(DeptNo));
```