Lecture 8

Chapter 7: Constraints (Examples)

John Connor

April 25, 2018

Not-Null Constraint

```
CREATE TABLE Musician (
    ID INT PRIMARY KEY,
    LastName VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255),
    Age int
);
```

Which of the following will execute without problems?

```
INSERT INTO Musician VALUES (1, 'Coltrane', 'John', 40);
INSERT INTO Musician VALUES (2, 'Davis', 40);
INSERT INTO Musician (ID, LastName, Age)
    VALUES (3, 'Davis', 40);
INSERT INTO Musician (ID, FirstName, Age)
    VALUES (4, 'Thelonious', 64);
```

Not-Null Constraint

Will this delete the records with a `NULL` first name?

```
DELETE FROM Musician WHERE FirstName = NULL;
```

How about this?

```
DELETE FROM Musician WHERE FirstName IS NULL;
```

More Null Weirdness

`NULL` is probably weirder than you think!

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;` `NULL`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;` `NULL`
What is the value of `SELECT 1 <> NULL;`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;` `NULL`
What is the value of `SELECT 1 <> NULL;` `NULL`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;` `NULL`
What is the value of `SELECT 1 <> NULL;` `NULL`
What is the value of `SELECT NULL = NULL;`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;` `NULL`
What is the value of `SELECT 1 <> NULL;` `NULL`
What is the value of `SELECT NULL = NULL;` `NULL`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;` `NULL`
What is the value of `SELECT 1 <> NULL;` `NULL`
What is the value of `SELECT NULL = NULL;` `NULL`
What is the value of `SELECT NULL <> NULL;`

More Null Weirdness

`NULL` is probably weirder than you think!

What is the value of `SELECT 1 = 1;` `TRUE`
What is the value of `SELECT 1 = 0;` `FALSE`
What is the value of `SELECT 1 <> 0;` `TRUE`
What is the value of `SELECT 1 = NULL;` `NULL`
What is the value of `SELECT 1 <> NULL;` `NULL`
What is the value of `SELECT NULL = NULL;` `NULL`
What is the value of `SELECT NULL <> NULL;` `NULL`

Keys and Foreign Keys

What is wrong with the following?

Keys and Foreign Keys

What is wrong with the following? For example:

```
CREATE TABLE Instrument (
  Name VARCHAR(255)
);

CREATE TABLE Plays (
  MusicianID INT REFERENCES Musician(ID),
  Instrument VARCHAR(255) REFERENCES Instrument(Name)
);
```

Keys and Foreign Keys

Assume that the following statements have been executed along with the correct statements from the previous slides.

```
CREATE TABLE Instrument (
   Name VARCHAR(255) PRIMARY KEY
);

CREATE TABLE Plays (
   MusicianID INT REFERENCES Musician(ID),
   Instrument VARCHAR(255) REFERENCES Instrument(Name)
);

INSERT INTO Instrument VALUES ('Sax'), ('Piano'), ('Drur
```

Keys and Foreign Keys

Assume that the following statements have been executed along with the correct statements from the previous slides.

```
CREATE TABLE Instrument (
   Name VARCHAR(255) PRIMARY KEY
);

CREATE TABLE Plays (
   MusicianID INT REFERENCES Musician(ID),
   Instrument VARCHAR(255) REFERENCES Instrument(Name)
);

INSERT INTO Instrument VALUES ('Sax'), ('Piano'), ('Drur
```

Which of the following will execute without problems?

```
INSERT INTO Plays VALUES (1, 'Sax');
INSERT INTO Plays VALUES (3, 'Trumpet');
INSERT INTO Plays VALUES (14, 'Piano');
```

Policies

Assume the default policy. Will the following queries work?

```
DELETE FROM Instrument WHERE Name = 'Sax';
DELETE FROM Musician WHERE ID = 1;
```

Policies

Assume the default policy. Will the following queries work?

```
DELETE FROM Instrument WHERE Name = 'Sax';
DELETE FROM Musician WHERE ID = 1;
```

What will happen if the above runs under the cascade policy?

Policies

Assume the default policy. Will the following queries work?

```
DELETE FROM Instrument WHERE Name = 'Sax';
DELETE FROM Musician WHERE ID = 1;
```

What will happen if the above runs under the cascade policy?

Under the set-null policy?

Cyclic References

What is wrong with the following, and how can it be fixed?

```
CREATE TABLE Foo (
   ID INT PRIMARY KEY REFERENCES Bar(ID);
);
CREATE TABLE Bar (
   ID INT PRIMARY KEY REFERENCES Foo(ID);
);
INSERT INTO Foo VALUES (1);
```

Cyclic References

What is wrong with the following, and how can it be fixed?

```
CREATE TABLE Foo (
  ID INT PRIMARY KEY REFERENCES Bar(ID);
);
CREATE TABLE Bar (
  ID INT PRIMARY KEY REFERENCES Foo(ID);
);
INSERT INTO Foo VALUES (1);
```

```
CREATE TABLE Foo (
  ID INT PRIMARY KEY REFERENCES Bar(ID)
    DEFERRABLE INITIALLY DEFERRED;
);
CREATE TABLE Bar (
  ID INT PRIMARY KEY REFERENCES Foo(ID)
    DEFERRABLE INITIALLY DEFERRED;
);
INSERT INTO Foo VALUES (1);
```

Unique Columns

What is wrong with the following?

Unique Columns

What is wrong with the following?

```
CREATE TABLE States (
   Name VARCHAR(255) UNIQUE,
   Abbr CHAR(2) UNIQUE
);
INSERT INTO States VALUES
   ('Louisianna', 'LA'), ('Lewisianna', 'LA');
```

Unique Columns

What is wrong with the following?

```
CREATE TABLE States (
   Name VARCHAR(255) UNIQUE,
   Abbr CHAR(2) UNIQUE
);
INSERT INTO States VALUES
   ('Louisianna', 'LA'), ('Lewisianna', 'LA');
```

What happens in the following queries?

```
INSERT INTO States VALUES (NULL, NULL);
INSERT INTO States VALUES ('Florida', NULL);
INSERT INTO States VALUES ('New York', NULL);
INSERT INTO States VALUES ('Louisianna', NULL);
SELECT *
  FROM States A JOIN States B ON A.Abbr = B.Abbr;
```

Check Constraints (pg. 321)

Is this

```
CREATE TABLE Foo (
  ID INT PRIMARY KEY
);
CREATE TABLE Bar (
  ID INT PRIMARY KEY REFERENCES Foo(ID)
);
```

equivalent to

```
CREATE TABLE Foo (
  ID INT PRIMARY KEY
);
CREATE TABLE Bar (
  ID INT PRIMARY KEY
    CHECK (ID IN (SELECT ID FROM Foo))
);
```

Check Constraints

```
CREATE TABLE Foo (
  ID INT PRIMARY KEY
);
CREATE TABLE Bar (
  ID INT PRIMARY KEY
    CHECK (ID IN (SELECT ID FROM Foo))
);
INSERT INTO Foo VALUES (1), (2), (3);
```

Which of the following will succeed?

```
INSERT INTO Bar VALUES (1);
INSERT INTO Bar VALUES (4);
DELETE FROM Foo WHERE ID = 1;
```

More Constraints

```
CREATE TABLE People (
    ID INT PRIMARY KEY,
    Name VARCHAR(255),
    BirthYear INT,
    MotherID INT REFERENCES People(ID),
    FatherID INT REFERENCES People(ID)
);
```

How can we ensure that a person has a birth year that is after the birth year of their mother and father?

More Constraints

```
CREATE TABLE People (
   ID INT PRIMARY KEY,
   Name VARCHAR(255),
   BirthYear INT
     CHECK (BirthYear > (
        SELECT BirthYear FROM People
         WHERE ID = MotherID)
        AND BirthYear > (
        SELECT BirthYear FROM People
         WHERE ID = FatherID)),
   MotherID INT REFERENCES People(ID),
   FatherID INT REFERENCES People(ID)
);
```

Assertions

```
CREATE ASSERTION NoTimeTravel (NOT EXISTS (
  (SELECT BirthYear FROM People
    WHERE BirthYear > (SELECT BirthYear FROM People
                         WHERE ID = MotherID)
       OR BirthYear > (SELECT BirthYear FROM People
                         WHERE ID = FatherID)
  )));
```