4.9 Column constraints

NOT NULL constraint

All table columns, except primary keys, may contain NULL values by default. The ***NOT NULL*** constraint is used in a CREATE TABLE statement to prevent a column from having a NULL value.

PARTICIPATION ACTIVITY 4.9.1: NOT NULL constraint on the Employee table.

1 2 2x speed

```
CREATE TABLE Employee (
   ID          SMALLINT UNSIGNED,
   Name        VARCHAR(60) NOT NULL,
   BirthDate   DATE,
   Salary      DECIMAL(7,2),
   PRIMARY KEY (ID)
);
```

Employee

ID	Name	BirthDate	Salary
6381	Maria Rodriguez	NULL	92300
~~2538~~	~~NULL~~	~~1990-12-03~~	~~423.00~~

The NOT NULL constraint prevents Name from being NULL when inserting a new row into Employee.

Captions

1. The BirthDate column allows NULL values by default.
2. The NOT NULL constraint prevents Name from being NULL when inserting a new row into Employee.

Feedback?

PARTICIPATION ACTIVITY 4.9.2: NOT NULL constraint.

Refer to the statement below.

```
CREATE TABLE Department (
   Code      TINYINT UNSIGNED,
   Name      VARCHAR(20),
   ManagerID SMALLINT,
   PRIMARY KEY (Code)
);
```

1) Which columns may contain NULL values?

- Code
- Code and Name
- Name and ManagerID

Correct

All columns by default, except the primary key Code, can contain NULL values.

2) Which alteration to the CREATE TABLE statement prevents ManagerID from being NULL?

- `ManagerID NOT NULL SMALLINT,`
- `ManagerID NOT NULL,`
- `ManagerID SMALLINT NOT NULL,`

Correct

The NOT NULL constraint must be listed after the column name and data type.

Feedback?

DEFAULT constraint

When a row is inserted into a table, an unspecified value is assigned NULL by default. The ***DEFAULT*** constraint is used in a CREATE TABLE statement to specify a column's default value when no value is provided.

PARTICIPATION ACTIVITY 4.9.3: DEFAULT constraints on the Employee table.

1 2 3 2x speed

```
CREATE TABLE Employee (
   ID          SMALLINT UNSIGNED,
   Name        VARCHAR(60),
   BirthDate   DATE DEFAULT '2000-01-01',
   Salary      DECIMAL(7,2) DEFAULT 0.00,
   PRIMARY KEY (ID)
);
```

Employee

ID	Name	BirthDate	Salary
6381	NULL	1970-12-01	92300
2538	Lisa Ellison	2000-01-01	423.00
5384	Sam Snead	2002-10-31	0.00

The Salary column has a DEFAULT constraint, so salary is 0.00 if no salary is specified when inserting a new row.

Captions

1. The Name column does not have a DEFAULT constraint. When a row is inserted into Employee with no specified name, Name is NULL by default.
2. The BirthDate column has a DEFAULT constraint, so January 1, 2000 is used if a row is inserted with no specified birth date.
3. The Salary column has a DEFAULT constraint, so salary is 0.00 if no salary is specified when inserting a new row.

Feedback?

PARTICIPATION ACTIVITY 4.9.4: DEFAULT constraint.

Refer to the statement below.

```
CREATE TABLE Department (
   Code       TINYINT UNSIGNED,
   Name       VARCHAR(20),
   ManagerID SMALLINT,
   PRIMARY KEY (Code)
);
```

1) What column values are NULL when inserting a row with Code 999 but no specified name or manager?

- ○ Name only
- ○ ManagerID only
- ◉ Name and ManagerID

Correct

Both columns are missing DEFAULT constraints, so both column values are assigned NULL by default.

2) Which alteration to the CREATE TABLE statement sets the default Name to Accounting?

- ○ `Name VARCHAR(20) DEFAULT Accounting,`
- ○ `Name VARCHAR(20) DEFAULT 'Accounting',`
- ○ `Name VARCHAR(20) 'Accounting',`

Feedback?

UNIQUE constraint

A table's primary key always has unique values, but values in other columns may contain duplicates. The ***UNIQUE*** constraint ensures that all column values are unique.

The UNIQUE constraint may be applied to a single column or to multiple columns. A constraint that is applied to a single column is called a ***column-level constraint***. A constraint that is applied to multiple columns is called a ***table-level constraint***.

MySQL creates an index for each UNIQUE constraint, which can improve query retrieval performance.

PARTICIPATION ACTIVITY 4.9.5: UNIQUE constraints on the Employee table.

1 2 3 4 5 2x speed

```
CREATE TABLE Employee (
   ID          SMALLINT UNSIGNED,
   Name        VARCHAR(60),
   Extension   CHAR(4),
   Username    VARCHAR(50) UNIQUE,
   UNIQUE (Name, Extension),
   PRIMARY KEY (ID)
);
```

Employee

ID	Name	Extension	Username
6381	Maria Rodriguez	5050	mrodriguez
~~2538~~	~~Marty Rodriguez~~	~~5102~~	~~mrodriguez~~
5384	Maria Rodriguez	4888	mrod2
~~9001~~	~~Maria Rodriguez~~	~~5050~~	~~mariarod~~

Attempting to insert Maria Rodriquez with extension 5050 fails because (Maria Rodriguez, 5050) already exists.

Captions

1. The Username column has a UNIQUE column-level constraint, so each Username must be different.
2. Attempting to insert a second row with username 'mrodriguez' fails because 'mrodriguez' already exists.
3. The UNIQUE table-level constraint requires each combination of Name and Extension be different.
4. Inserting Maria Rodriquez with extension 4888 does not violate the UNIQUE constraint because (Maria Rodriguez, 5050) and (Maria Rodriguez, 4888) are different.

5. Attempting to insert Maria Rodriquez with extension 5050 fails because (Maria Rodriguez, 5050) already exists.

Feedback?

PARTICIPATION ACTIVITY 4.9.6: UNIQUE constraint.

Indicate whether the rows violate the Department table's UNIQUE constraints or not.

```
CREATE TABLE Department (
   Code         TINYINT UNSIGNED,
   Name         VARCHAR(20) UNIQUE,
   ManagerID    SMALLINT,
   Appointment DATE,
   PRIMARY KEY (Code),
   UNIQUE (ManagerID, Appointment)
);
```

1) `(44, 'Engineering', 2538, '2020-01-15')`
`(82, 'Sales', 6381, '2019-08-01')`

- (●) OK
- ( ) Violates

Correct

The Codes 44 and 82 are different. The Names 'Engineering' and 'Sales' are different. The ManagerID and Appointment pairs (2538, '2020-01-15') and (6381, '2019-08-01') are different.

2) `(12, 'Marketing', 2538, '2019-11-22')`
`(99, 'Marketing', NULL, NULL)`

- ( ) OK
- (●) Violates

Correct

The Name column has a UNIQUE constraint, but the two rows have the same department name 'Marketing'.

3)

Correct

```
(44, 'Engineering', 2538,
'2020-01-15')
(82, 'Sales', 2538, '2020-
01-15')
```

The two rows have different Codes and Names, but the UNIQUE constraint on (Manager, Appointment) is violated; (2538, '2020-01-15') is the same in both rows.

- ○ OK
- ◉ Violates

4)

```
(12, 'Marketing', 5384,
'2019-11-22')
(99, 'Technical support',
5384, NULL)
```

Correct

The two rows have different Codes and Names, and (5384, '2019-11-22') and (5384, NULL) are different.

- ◉ OK
- ○ Violates

Feedback?

CHECK constraint

The ***CHECK*** constraint specifies an expression that limits the range of a column's values. Ex: `CHECK (Salary > 20000)` ensures the Salary is greater than 20,000. If the CHECK expression does not evaluate to TRUE or UNKNOWN (for NULL values), the constraint is violated.

A CHECK constraint can be a column-level or table-level constraint.

PARTICIPATION ACTIVITY 4.9.7: CHECK constraints on the Employee table.

1 2 3 4 5 2x speed

```
CREATE TABLE Employee (
   ID        SMALLINT UNSIGNED,
   Name      VARCHAR(60),
   BirthDate DATE,
   HireDate  DATE CHECK (HireDate >= '2000-01-01' AND HireDate <= '2019-12-31'),
   CHECK (BirthDate < HireDate),
```

```
    PRIMARY KEY (ID)
);
```

Employee

ID	Name	BirthDate	HireDate
6381	Maria Rodriguez	1970-12-01	2000-01-31
~~2538~~	~~Lisa Ellison~~	~~2000-01-01~~	~~2020-03-15~~
~~5384~~	~~Sam Snead~~	~~2003-11-29~~	~~2003-11-01~~
8312	Jiho Chen	NULL	2013-06-03

The NULL BirthDate makes the table-level CHECK constraint evaluate to UNKNOWN, which does not violate the constraint.

Captions

1. The CHECK column-level constraint ensures all rows have a HireDate between Jan 1, 2000 and Dec 31, 2019.
2. Maria's HireDate is between Jan 1, 2000 and Dec 31, 2019. But inserting Lisa fails because Mar 15, 2020 is after Dec 31, 2019.
3. The CHECK table-level constraint requires BirthDate to come before HireDate.
4. Inserting Sam fails because the BirthDate Nov 29, 2003 is after HireDate Nov 1, 2003.
5. The NULL BirthDate makes the table-level CHECK constraint evaluate to UNKNOWN, which does not violate the constraint.

Feedback?

BETWEEN operator

*Determining if a value is between two other values, like the animation above does with HireDate, is a common SQL operation. The **BETWEEN** operator provides an alternative way to determine if a value is between two other values. Ex:*

```
-- Same as: HireDate >= '2000-01-01' AND HireDate <= '2020-01-01'
HireDate DATE CHECK (HireDate BETWEEN '2000-01-01' AND '2020-01-01'),
```

PARTICIPATION ACTIVITY 4.9.8: CHECK constraint.

The CREATE TABLE statement below specifies two CHECK constraints. The column-level CHECK constraint limits the Size column to one of the values listed after the IN operator.

```
CREATE TABLE Department (
   Code          TINYINT UNSIGNED,
   Name          VARCHAR(20),
   ManagerID     SMALLINT,
   AdminAssistID SMALLINT,
   Size          VARCHAR(6) CHECK (Size IN ('small', 'medium', 'large')),
   PRIMARY KEY (Code),
   CHECK (ManagerID >= 1000 AND ManagerID <> AdminAssistID)
);
```

Indicate whether each row violates the Department table's CHECK constraints or not.

1) `(44, 'Engineering', 2538, 1024, 'tiny')`

   - ○ OK
   - ◉ Violates

Correct

The column-level CHECK constraint limits the Size column to 'small', 'medium', or 'large'. The size 'tiny' violates the constraint.

2) `(82, 'Sales', 999, 1024, 'medium')`

   - ○ OK
   - ◉ Violates

Correct

The table-level CHECK constraint requires ManagerID to be >= 1000, but 999 < 1000.

3)

```
(99, 'Technical support',
5384, 5384, 'large')
```

- ○ OK
- ◉ Violates

Correct

The ManagerID and AdminAssistID are both 5384, but the table-level CHECK constraint requires ManagerID and AdminAssistID to be different.

4) `(99, 'Technical support', 5384, 2399, 'large')`

- ◉ OK
- ○ Violates

Correct

Both CHECK constraints are met. The ManagerID 5384 is >= 1000 and <> 2399, and 'large' is a valid Size.

5) `(50, 'Maintenance', NULL, 4380, NULL)`

- ◉ OK
- ○ Violates

Correct

Both CHECK constraints are met. The NULL ManagerID causes the table-level CHECK constraint to evaluate to UNKNOWN. The NULL Size causes the column-level CHECK constraint to evaluate to UNKNOWN. Constraints that evaluate to UNKNOWN are not violated.

Feedback?

CONSTRAINT keyword

MySQL gives constraints a default name if no name is specified. A constraint can be given a name using the ***CONSTRAINT*** keyword, followed by the constraint name and declaration. Constraint names can help a database administrator or programmer identify which constraint is being violated in a database error message.

Constraints may be added to existing tables using the ALTER TABLE statement along with the CONSTRAINT keyword. Adding a constraint to an existing table fails if the table contains data that violates the constraint.

Most databases support dropping constraints using an ALTER TABLE statement with `DROP CONSTRAINT ConstraintName`. However, MySQL requires `DROP INDEX ConstraintName` for a UNIQUE constraint and `DROP CHECK ConstraintName` for a CHECK constraint.

PARTICIPATION 4.9.9: Naming constraints with CONSTRAINT.

ACTIVITY

1 2 3 4 2x speed

```
CREATE TABLE Employee (
   ID         SMALLINT UNSIGNED,
   Name       VARCHAR(60),
   HireDate   DATE CONSTRAINT HireCheck CHECK (HireDate >= '2000-01-01'), Name: HireChec
   CONSTRAINT UniqueNameHiredate UNIQUE (Name, HireDate),     Name: UniqueNameHiredate
   PRIMARY KEY (ID)
);
```

Add a new row: (305, 'Fred Yu', '1999-12-30') X

Error message: Check on constraint 'HireCheck' is violated.

```
ALTER TABLE Employee
DROP CHECK HireCheck;
```

Add a new row: (305, 'Fred Yu', '1999-12-30') ✓

```
ALTER TABLE Employee
ADD CONSTRAINT HireCheck CHECK (HireDate < '2000-02-14');
```

Add a new row: (610, 'Emma Jackson', '2014-12-30') X

Error message: Check on constraint 'HireCheck' is violated.

The ALTER TABLE statement adds a new constraint that ensures the HireDate is before Feb 14, 2000. Emma's HireDate violates the new constraint.

Captions

1. The CONSTRAINT keyword gives the CHECK and UNIQUE constraints user-defined names.
2. The database uses constraint names in error messages. Fred's HireDate is not >= 2000-01-01.
3. The ALTER TABLE statement drops the HireCheck constraint. Any HireDate may now be added to Employee.

4. The ALTER TABLE statement adds a new constraint that ensures the HireDate is before Feb 14, 2000. Emma's HireDate violates the new constraint.

Feedback?

PARTICIPATION ACTIVITY 4.9.10: Naming constraints.

1) Add a CHECK constraint called OrgCheck that limits OrgCode to 'ACH', 'PLT', or 'WAN'.

```
CREATE TABLE Department (
   Code TINYINT UNSIGNED,
   Name VARCHAR(20),
   OrgCode CHAR(3)
                        (OrgCode IN ('ACH',
'PLT', 'WAN')),
   PRIMARY KEY (Code)
);
```

Check Show answer

Answer

CONSTRAINT OrgCheck CHECK

The constraint name follows CONSTRAINT, then the CHECK constraint.

2) Add a UNIQUE constraint called UniqueNameMgr that ensures the Name and ManagerID combination are unique.

```
CREATE TABLE Department (
   Code TINYINT UNSIGNED,
   Name VARCHAR(20),
   ManagerID SMALLINT,
   OrgCode CHAR(3),

,
   PRIMARY KEY (Code)
);
```

Check Show answer

Answer

CONSTRAINT UniqueNameMgr UNIQUE (Name, ManagerID)

The table-level constraint appears on a line by itself because the constraint applies to more than one column.

Check Show answer

3) Add a constraint called MgrIdCheck to the existing Department table to ensure ManagerID is 2000 or above.

```
ALTER TABLE Department
                                    
(ManagerID >= 2000);
```

Check Show answer

Answer

ADD CONSTRAINT MgrIdCheck CHECK

The ADD CONSTRAINT keywords add the new CHECK constraint to Department.

4) Drop the UNIQUE constraint called UniqueNameMgr from the existing Department table.

```
ALTER TABLE Department
                                    
;
```

Check Show answer

Answer

DROP INDEX UniqueNameMgr

The DROP INDEX keywords drop a UNIQUE constraint.

Feedback?

CHALLENGE ACTIVITY 4.9.1: Column constraints.

379958.2369558.qx3zqy7

Start

1 2 3

```
CREATE TABLE Country (
   IndepDate DATE,
```

```
    Under14PopPct FLOAT,
    15to64PopPct FLOAT NOT NULL,
    Continent VARCHAR(8) NOT NULL,
    Over65PopPct FLOAT,
    ISOCode2 CHAR(2),
    PRIMARY KEY (IndepDate)
);
```

Which columns can contain NULL values?

- [ ] 15to64PopPct
- [ ] Continent
- [ ] ISOCode2
- [ ] IndepDate
- [ ] Under14PopPct
- [ ] Over65PopPct

4 ✓ 5

1 2 3 4 5

Check Next

Feedback?

Exploring further:

- How MySQL Deals with Constraints from MySQL.com
- CHECK Constraints from MySQL.com

How was this section? Provide feedback