**MySQL Constraints**

SQL constraints are used to specify rules for data in a table.

## Create Constraints

Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

### Syntax

CREATE TABLE table\_name (  
    column1 datatype *constraint*,  
    column2 datatype *constraint*,  
    column3 datatype *constraint*,  
    ....  
);

## MySQL Constraints

SQL constraints are used to specify rules for the data in a table.

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.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

* [NOT NULL](https://www.w3schools.com/MySQL/mysql_notnull.asp) - Ensures that a column cannot have a NULL value
* [UNIQUE](https://www.w3schools.com/MySQL/mysql_unique.asp) - Ensures that all values in a column are different
* [PRIMARY KEY](https://www.w3schools.com/MySQL/mysql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* [FOREIGN KEY](https://www.w3schools.com/MySQL/mysql_foreignkey.asp) - Prevents actions that would destroy links between tables
* [CHECK](https://www.w3schools.com/MySQL/mysql_check.asp) - Ensures that the values in a column satisfies a specific condition
* [DEFAULT](https://www.w3schools.com/MySQL/mysql_default.asp) - Sets a default value for a column if no value is specified
* [CREATE INDEX](https://www.w3schools.com/MySQL/mysql_create_index.asp) - Used to create and retrieve data from the database very quickly

# MySQL NOT NULL Constraint

## MySQL NOT NULL Constraint

By default, a column can hold NULL values.

The NOT NULL constraint enforces a column to NOT accept NULL values.

This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

## NOT NULL on CREATE TABLE

The following SQL ensures that the "ID", "LastName", and "FirstName" columns will NOT accept NULL values when the "Persons" table is created:

### Example

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);

CREATE table users (id int AUTO\_INCREMENT PRIMARY KEY, fname varchar(20) not null, lname varchar(20) not null, city varchar(20))

INSERT into users (fname, lname, city) values ('bhavin', 'thakar', 'rajkot')

INSERT into users (fname, lname, city) values ('bhavya', 'dava', null);

## NOT NULL on ALTER TABLE

To create a NOT NULL constraint on the "Age" column when the "Persons" table is already created, use the following SQL:

### Example

ALTER TABLE Persons  
MODIFY Age int NOT NULL;

ALTER TABLE users MODIFY COLUMN city varchar(20) not null

INSERT into users (fname, lname, city) values ('bhavya', 'dava', null);

#1048 - Column 'city' cannot be null

# MySQL UNIQUE Constraint

## MySQL UNIQUE Constraint

The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

DROP TABLE users

## UNIQUE Constraint on CREATE TABLE

The following SQL creates a UNIQUE constraint on the "ID" column when the "Persons" table is created:

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    UNIQUE (ID)  
);

CREATE TABLE users (id int AUTO\_INCREMENT PRIMARY key, fname varchar(20) not null, lname varchar(20) not null, city varchar(20) not null, email varchar(64) UNIQUE)

INSERT into users (fname, lname, city, email) VALUES ('het', 'manani', 'rajkot', 'het@gmail.com')

INSERT into users (fname, lname, city, email) VALUES ('het', 'manani', 'rajkot', 'het@gmail.com')

INSERT into users (fname, lname, city, email) VALUES ('het', 'manani', 'rajkot', null)

INSERT into users (fname, lname, city, email) VALUES ('het', 'manani', 'rajkot', null);

INSERT into users (fname, lname, city, email) VALUES ('het', 'manani', 'rajkot', null);

## UNIQUE Constraint on ALTER TABLE

To create a UNIQUE constraint on the "ID" column when the table is already created, use the following SQL:

ALTER TABLE Persons  
ADD UNIQUE (ID);

To name a UNIQUE constraint, and to define a UNIQUE constraint on multiple columns, use the following SQL syntax:

ALTER TABLE Persons  
ADD CONSTRAINT UC\_Person UNIQUE (ID,LastName);

drop TABLE users

CREATE TABLE users (id int AUTO\_INCREMENT PRIMARY key, fname varchar(20) not null, lname varchar(20) not null, city varchar(20) not null, email varchar(64) not null)

ALTER TABLE users MODIFY COLUMN email varchar(64) not null UNIQUE

alter TABLE users add CONSTRAINT unq\_email UNIQUE(email)

INSERT into users (fname, lname, city, email) VALUES ('het', 'manani', 'rajkot', 'het@gmail.com')

INSERT into users (fname, lname, city, email) VALUES ('het', 'manani', 'rajkot', 'het@gmail.com')

#1062 - Duplicate entry 'het@gmail.com' for key 'unq\_email'

## DROP a UNIQUE Constraint

To drop a UNIQUE constraint, use the following SQL:

ALTER TABLE Persons  
DROP INDEX UC\_Person;

ALTER TABLE users drop CONSTRAINT unq\_email

To name a UNIQUE constraint, and to define a UNIQUE constraint on multiple columns, use the following SQL syntax:

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CONSTRAINT UC\_Person UNIQUE (ID,LastName)  
);

create table listofvillage (villageid int AUTO\_INCREMENT PRIMARY key, villagename varchar(20) not null, cityname varchar(20) not null, taluka varchar(20) not null, district varchar(20) not null, state varchar(20) not null, CONSTRAINT unq\_villagename UNIQUE(villagename, cityname, taluka))

INSERT into listofvillage (villagename, cityname, taluka, district, state) VALUES('Navagam', 'Rajkot', 'Rajkot', 'Rajkot', 'Gujarat')

INSERT into listofvillage (villagename, cityname, taluka, district, state) VALUES('Navagam', 'Rajkot', 'Rajkot', 'Rajkot', 'Gujarat');

INSERT into listofvillage (villagename, cityname, taluka, district, state) VALUES('Navagam', 'Gondal', 'Rajkot', 'Rajkot', 'Gujarat');

# MySQL PRIMARY KEY Constraint

## MySQL PRIMARY KEY Constraint

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

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

CREATE TABLE students (id int PRIMARY key, fname varchar(20) not null, lname varchar(20) not null, email varchar(64) not null)

INSERT into students (id, fname, lname, email) values (1, 'keyur', 'ramani', 'keyur@rajkot.com')

INSERT into students (id, fname, lname, email) values (1, 'keyur', 'ramani', 'keyur@rajkot.com')

#1062 - Duplicate entry '1' for key 'PRIMARY'

## DROP a PRIMARY KEY Constraint

To drop a PRIMARY KEY constraint, use the following SQL:

ALTER TABLE Persons  
DROP PRIMARY KEY;

alter TABLE students drop PRIMARY key

## PRIMARY KEY on ALTER TABLE

To create a PRIMARY KEY constraint on the "ID" column when the table is already created, use the following SQL:

ALTER TABLE Persons  
ADD PRIMARY KEY (ID);

alter TABLE students add PRIMARY key (id)

TRUNCATE TABLE students

alter table students add CONSTRAINT pri\_key\_id PRIMARY key (id)

INSERT into students (id, fname, lname, email) values (1, 'keyur', 'ramani', 'keyur@rajkot.com');

INSERT into students (id, fname, lname, email) values (1, 'keyur', 'ramani', 'keyur@rajkot.com')

#1062 - Duplicate entry '1' for key 'PRIMARY'

# MySQL FOREIGN KEY Constraint

## MySQL FOREIGN KEY Constraint

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the [PRIMARY KEY](https://www.w3schools.com/MySQL/mysql_primarykey.asp) in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

## FOREIGN KEY on CREATE TABLE

The following SQL creates a FOREIGN KEY on the "PersonID" column when the "Orders" table is created:

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

CREATE TABLE students (id int AUTO\_INCREMENT PRIMARY key, fname varchar(20) not null, lname varchar(20) not null, city varchar(20) not null)

INSERT into students (fname, lname, city) VALUES ('het', 'Manani', 'rajkot'),('bhavin', 'thakar', 'rajkot'),('bhavya', 'dava', 'rajkot'),('rajiv', 'shekh', 'rajkot'),('yash', 'singla', 'rajkot'),('yash', 'chavda', 'rajkot'),('keyur', 'ramani', 'rajkot'),('prince', 'gajirapa', 'rajkot'),('kalpes', 'chauahn', 'rajkot'),('het', 'Manani', 'rajkot');

CREATE TABLE attendance (attendance\_id int AUTO\_INCREMENT PRIMARY key, students\_id int, absents int, presents int, FOREIGN key (students\_id) REFERENCES students(id))

drop TABLE attendance

CREATE TABLE attendance (attendance\_id int AUTO\_INCREMENT PRIMARY key, students\_id int, absents int, presents int, CONSTRAINT fk\_students\_attendance\_id FOREIGN key (students\_id) REFERENCES students(id))

INSERT into attendance (students\_id, absents, presents) VALUES (1, 111, 112)

INSERT into attendance (students\_id, absents, presents) VALUES (111, 111, 112);

#1452 - Cannot add or update a child row: a foreign key constraint fails (`224sample`.`attendance`, CONSTRAINT `fk\_students\_attendance\_id` FOREIGN KEY (`students\_id`) REFERENCES `students` (`id`))

ALTER TABLE attendance drop CONSTRAINT fk\_students\_attendance\_id

ALTER TABLE attendance add CONSTRAINT fk\_students\_attendance\_id FOREIGN key (students\_id) REFERENCES students (id);