

| | |
|-------------------------------|---|
| SQL Introduction | > |
| SQL SELECT (I) | > |
| SQL SELECT (II) | > |
| SQL JOIN | > |
| SQL DATABASE & TABLE | > |
| SQL Insert, Update and Delete | > |
| SQL Constraints | ^ |
| SQL Constraints | |
| SQL Not Null Constraint | |
| SQL Unique Constraints | |
| SQL Primary Key | |
| SQL Foreign Key | |
| SQL Check | |
| SQL Default | |
| SQL Create Index | |
| SQL Additional Topics | > |

| |
|---------------------------|
| Related Topics |
| SQL DEFAULT Constraint |
| SQL UNIQUE Constraint |
| SQL PRIMARY KEY |
| SQL ALTER TABLE Statement |
| SQL Constraints |
| SQL CREATE INDEX |

SQL NOT NULL Constraint

In this tutorial, we'll learn to use the NOT NULL constraint with the help of examples.

The `NOT NULL` constraint in a column means that the column cannot store `NULL` values. For example,

```
CREATE TABLE Colleges (  
  college_id INT NOT NULL,  
  college_code VARCHAR(20),  
  college_name VARCHAR(50)  
);
```

Run Code >>

Here, the `college_id` and the `college_code` columns of the `Colleges` table won't allow `NULL` values.

Note: The `NOT NULL` constraint is used to add a constraint table whereas `IS NULL` and `NOT NULL` are used with the `WHERE` clause to select rows from the table.

Remove NOT NULL Constraint

We can also remove the NOT NULL constraint if that is no longer needed. For example,

SQL Server

```
ALTER TABLE Customers  
ALTER COLUMN age INT;
```

Oracle

```
ALTER TABLE Customers  
MODIFY (age NULL);
```

MySQL

```
ALTER TABLE Customers  
MODIFY age INT;
```

PostgreSQL

```
ALTER TABLE Customer  
ALTER COLUMN age DROP NOT NULL;
```

NOT NULL Constraint With Alter Table

We can also add the `NOT NULL` constraint to a column in an existing table using the `ALTER TABLE` command. For example,

SQL Server

```
ALTER TABLE Customers  
ALTER COLUMN age INT NOT NULL;
```

Oracle

```
ALTER TABLE Customers  
MODIFY age INT NOT NULL;
```

MySQL

```
ALTER TABLE Customers  
MODIFY COLUMN age INT NOT NULL;
```

PostgreSQL

```
ALTER TABLE Customers  
ALTER COLUMN age SET NOT NULL;
```

Here, the SQL command adds the `NOT NULL` constraint to the column `college_id` in an existing table.

Now when we try to insert records in a Colleges table without value for `college_id`, SQL will give us an error. For example,

```
INSERT INTO Colleges(college_code, college_name)  
VALUES ('NYC', 'US');
```

Run Code >>

Here, the SQL command gives us an error because we cannot skip the `college_id` field in a table because of the `NOT NULL` constraint.

Recommended Readings

- SQL Constraints
- SQL INSERT INTO
- SQL PRIMARY KEY
- SQL FOREIGN KEY

Previous Tutorial:
SQL Constraints

Next Tutorial:
SQL Unique Constraints

Did you find this article helpful?

Related Tutorials

Programming

SQL DEFAULT Constraint

Programming

SQL UNIQUE Constraint

Programming

SQL PRIMARY KEY

Programming

SQL ALTER TABLE Statement