SQL Server: Check Constraints

techonthenet.com/sql_server/check.php

This SQL Server tutorial explains how to use the **check constraints** in SQL Server (Transact-SQL) with syntax and examples.

What is a check constraint in SQL Server?

A check constraint in SQL Server (Transact-SQL) allows you to specify a condition on each row in a table.

Note

- A check constraint can NOT be defined on a SQL View.
- The check constraint defined on a table must refer to only columns in that table. It can not refer to columns in other tables.
- A check constraint can NOT include a Subquery.
- A check constraint can be defined in either a CREATE TABLE statement or a ALTER TABLE statement.

Using a CREATE TABLE statement

The syntax for creating a check constraint using a CREATE TABLE statement in SQL Server (Transact-SQL) is:

```
CREATE TABLE table_name
(
  column1 datatype [ NULL | NOT NULL ],
  column2 datatype [ NULL | NOT NULL ],

  ...

  CONSTRAINT constraint_name
    CHECK [ NOT FOR REPLICATION ] (column_name condition)

);
```

table_name
The name of the table that you wish to create with a check constraint.

constraint_name
The name to assign to the check constraint.

column_name
The column in the table that the check constraint applies to.

condition
The condition that must be met for the check constraint to succeed.

Example

Let's look at an example of how to use the CREATE TABLE statement in SQL Server to create a check constraint.

For example:

```
CREATE TABLE employees
( employee_id INT NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  first_name VARCHAR(50),
  salary MONEY,
  CONSTRAINT check_employee_id
    CHECK (employee_id BETWEEN 1 and 10000)
);
```

In this first example, we've created a check constraint on the *employees* table called *check_employee_id*. This constraint ensures that the *employee_id* field contains values between 1 and 10000.

Let's take a look at another example.

```
CREATE TABLE employees
( employee_id INT NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  first_name VARCHAR(50),
  salary MONEY,
  CONSTRAINT check_salary
    CHECK (salary > 0)
);
```

In this second example, we've created a check constraint on the *employees* table called *check_salary*. This constraint ensures that the salary value is greater than 0.

Using an ALTER TABLE statement

The syntax for creating a check constraint in an ALTER TABLE statement in SQL Server (Transact-SQL) is:

```
ALTER TABLE table_name
ADD CONSTRAINT constraint_name
  CHECK (column_name condition);
```

table_name
The name of the table that you wish to modify by adding a check constraint.

constraint_name
The name to assign to the check constraint.

column_name
The column in the table that the check constraint applies to.

condition
The condition that must be met for the check constraint to succeed.

Example

Let's look at an example of how to use the ALTER TABLE statement to create a check constraint in SQL Server.

For example:

```
ALTER TABLE employees
ADD CONSTRAINT check_last_name
  CHECK (last_name IN ('Smith', 'Anderson', 'Jones'));
```

In this example, we've created a check constraint on the existing *employees* table called *check_last_name*. It ensures that the *last_name* field only contains the following values: Smith, Anderson, or Jones.

Drop a Check Constraint

The syntax for dropping a check constraint in SQL Server (Transact-SQL) is:

```
ALTER TABLE table_name
DROP CONSTRAINT constraint_name;
```

table_name
The name of the table that you wish to drop the check constraint.

constraint_name
The name of the check constraint to remove.

Example

Let's look at an example of how to drop a check constraint in SQL Server.

For example:

```
ALTER TABLE employees
DROP CONSTRAINT check_last_name;
```

In this SQL Server example, we are dropping a check constraint on the *employees* table called *check_last_name*.

Enable a Check Constraint

The syntax for enabling a check constraint in SQL Server (Transact-SQL) is:

```
ALTER TABLE table_name
WITH CHECK CHECK CONSTRAINT constraint_name;
```

table_name
The name of the table that you wish to enable the check constraint.

constraint_name
The name of the check constraint to enable.

Example

Let's look at an example of how to enable a check constraint in SQL Server.

For example:

```
ALTER TABLE employees
WITH CHECK CHECK CONSTRAINT check_salary;
```

In this example, we are enabling a check constraint on the *employees* table called *check_salary*.

Disable a Check Constraint

The syntax for disabling a check constraint in SQL Server (Transact-SQL) is:

```
ALTER TABLE table_name
NOCHECK CONSTRAINT constraint_name;
```

table_name
The name of the table that you wish to disable the check constraint.

constraint_name
The name of the check constraint to disable.

Example

Let's look at an example of how to disable a check constraint in SQL Server.

For example:

```
ALTER TABLE employees
NOCHECK CONSTRAINT check_salary;
```

In this SQL Server example, we are disabling a check constraint on the *employees* table called *check_salary*.