SQL CHECK Constraint

The CHECK constraint in SQL is used to limit the values that can be inserted into a column. It allows you to define a condition that must be met for the data to be valid in the specified column.

OR

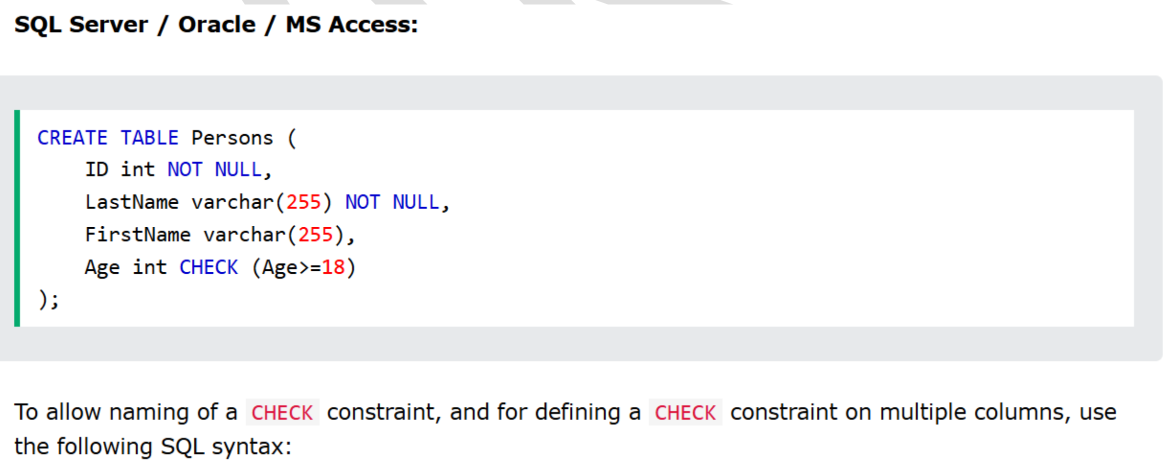
The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a column it will allow only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

SQL CHECK on CREATE TABLE

The following SQL creates a CHECK constraint on the "Age" column when the "Persons" table is created. The CHECK constraint ensures that the age of a person must be 18, or older:



SQL Server / Oracle / MS Access:

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int CHECK (Age>=18)
);
```

To allow naming of a `CHECK` constraint, and for defining a `CHECK` constraint on multiple columns, use the following SQL syntax:

MySQL / SQL Server / Oracle / MS Access:

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    City varchar(255),
    CONSTRAINT CHK_Person CHECK (Age>=18 AND City='Sandnes')
);
```

SQL CHECK on ALTER TABLE

To create a `CHECK` constraint on the "Age" column when the table is already created, use the following SQL:

MySQL / SQL Server / Oracle / MS Access:

```
ALTER TABLE Persons
ADD CHECK (Age>=18);
```

To allow naming of a `CHECK` constraint, and for defining a `CHECK` constraint on multiple columns, use the following SQL syntax:

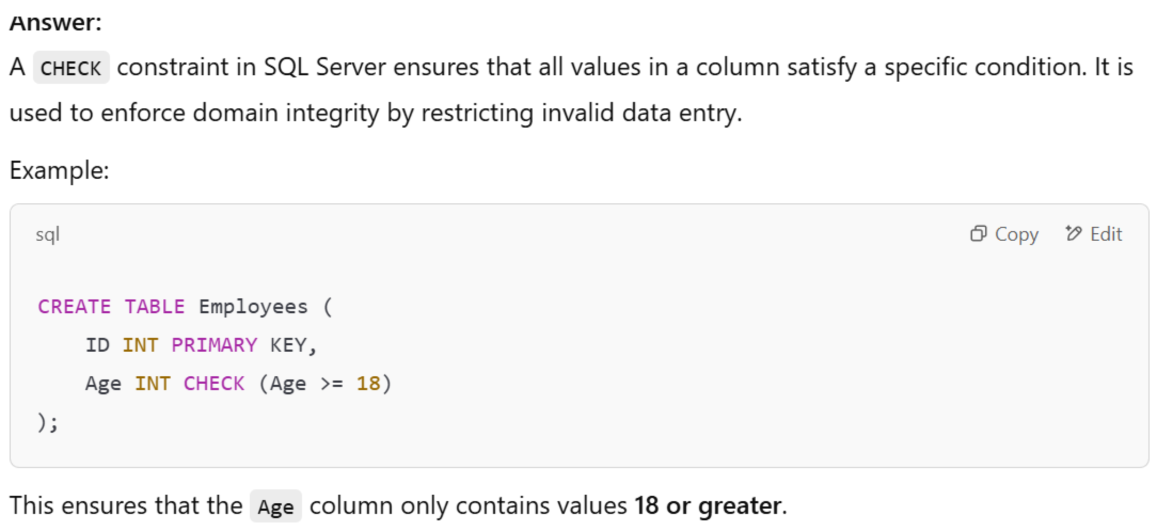
DROP a CHECK Constraint

To drop a `CHECK` constraint, use the following SQL:

SQL Server / Oracle / MS Access:

```
ALTER TABLE Persons
DROP CONSTRAINT CHK_PersonAge;
```

1. **What is a CHECK constraint in SQL Server?**



Answer:

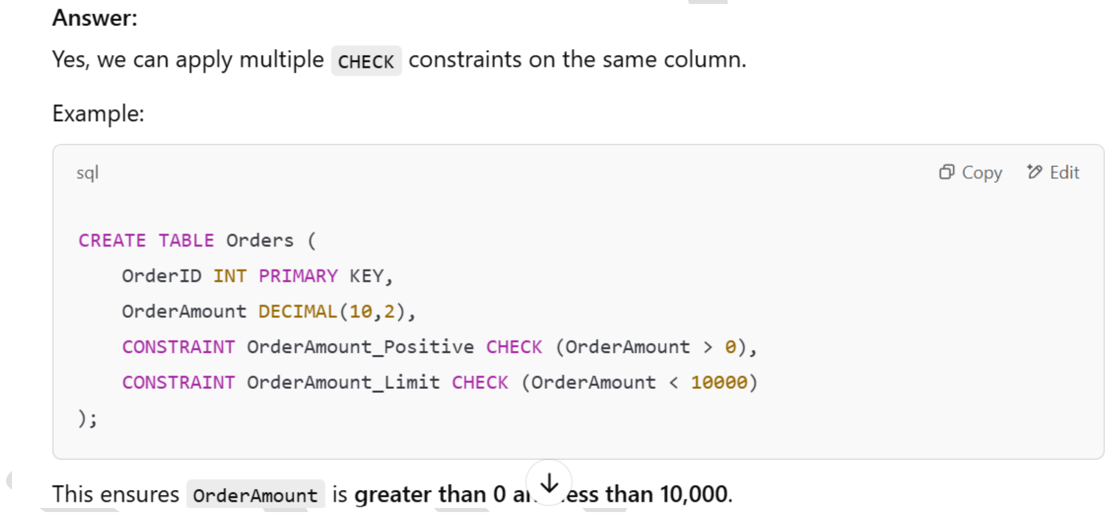
A `CHECK` constraint in SQL Server ensures that all values in a column satisfy a specific condition. It is used to enforce domain integrity by restricting invalid data entry.

Example:

```sql
CREATE TABLE Employees (
    ID INT PRIMARY KEY,
    Age INT CHECK (Age >= 18)
);
```

This ensures that the `Age` column only contains values **18 or greater**.

2. **Can we apply multiple CHECK constraints on a single column?**



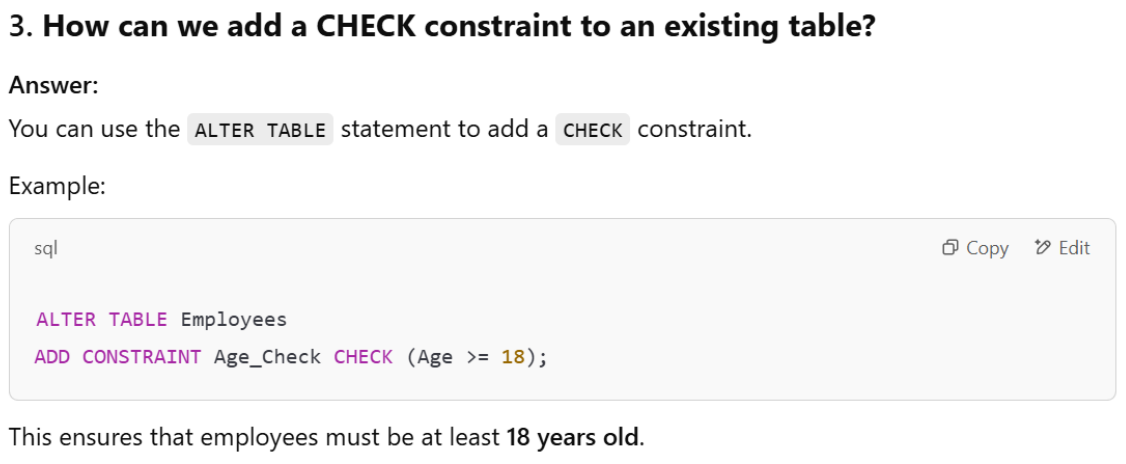
Answer:

Yes, we can apply multiple `CHECK` constraints on the same column.

Example:

```sql
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    OrderAmount DECIMAL(10,2),
    CONSTRAINT OrderAmount_Positive CHECK (OrderAmount > 0),
    CONSTRAINT OrderAmount_Limit CHECK (OrderAmount < 10000)
);
```

This ensures `OrderAmount` is **greater than 0 a** **ess than 10,000**.



3. How can we add a CHECK constraint to an existing table?

Answer:

You can use the `ALTER TABLE` statement to add a `CHECK` constraint.

Example:

```sql
ALTER TABLE Employees
ADD CONSTRAINT Age_Check CHECK (Age >= 18);
```

This ensures that employees must be at least **18 years old**.

4. How can we remove a CHECK constraint from a table?

Answer:

Use the `ALTER TABLE` statement with `DROP CONSTRAINT`.

Example:

sql Copy Edit

```sql
ALTER TABLE Employees
DROP CONSTRAINT Age_Check;
```

This removes the `Age_Check` constraint from the **Employees** table.

3. **Can we use multiple columns in a CHECK constraint?**

   Yes, we can use multiple columns in a single `CHECK` constraint.

   Example:

   sql Copy Edit

```sql
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    Price DECIMAL(10,2),
    Discount DECIMAL(10,2),
    CONSTRAINT Price_Discount_Check CHECK (Price > Discount)
);
```

   This ensures that the **Price** is always **greater than the Discount**.

4. What happens if we try to insert a value that violates a CHECK constraint?

   Answer:

   If an `INSERT` or `UPDATE` statement tries to insert a value that does not meet the `CHECK` constraint condition, SQL Server throws an error.

   Example:

   sql Copy Edit

```sql
INSERT INTO Employees (ID, Age) VALUES (1, 15);
```

   If `Age` has a `CHECK (Age >= 18)`, this will **fail** with an error message.

7. Can we disable a CHECK constraint temporarily?

Answer:

Yes, you can disable a `CHECK` constraint using:

sql

Copy Edit

```sql
ALTER TABLE Employees NOCHECK CONSTRAINT Age_Check;
```

To enable it back:

sql

Copy Edit

```sql
ALTER TABLE Employees CHECK CONSTRAINT Age_Check;
```

This is useful when performing bulk inserts.