CHECK CONSTRAINT

SQL Constraints Check Constraint is used to specify a predicate that every tuple must satisfy in a given relation. It limits the values that a column can hold in a relation.

- The predicate in check constraint can hold a sub query.
- Check constraint defined on an attribute restricts the range of values for that attribute.
- If the value being added to an attribute of a tuple violates the check constraint, the check constraint evaluates to false and the corresponding update is aborted.
- Check constraint is generally specified with the CREATE TABLE command in SQL.

Syntax

```
CREATE TABLE pets (
    ID INT NOT NULL,
    Name VARCHAR(30) NOT NULL,
    Breed VARCHAR(20) NOT NULL,
    Age INT,
    GENDER VARCHAR(9),
    PRIMARY KEY(ID),
    CHECK(GENDER IN ('Male', 'Female', 'Unknown'))
);
```

Note: The check constraint in the above SQL command restricts the GENDER to belong to only the categories specified. If a new tuple is added or an existing tuple in the relation is updated with a GENDER that doesn't belong to any of the three categories mentioned, then the corresponding database update is aborted.

Query

Constraint: Only students with age >= 17 are can enroll themselves in a university. **Schema for student database in university**.

```
CREATE TABLE student (
    StudentID INT NOT NULL,
    Name VARCHAR(30) NOT NULL,
    Age INT NOT NULL,
    GENDER VARCHAR(9),
    PRIMARY KEY(StudentID),
    CHECK(Age >= 17)
);
```

Student relation

StudentID	Name	Age	Gender
1001	Ron	18	Male

1002	Sam	17	Male
1003	Georgia	17	Female
1004	Erik	19	Unknown
1005	Christine	17	Female

Explanation: In the above relation, the age of all students is greater than equal to 17 years, according to the constraint mentioned in the check statement in the schema of the relation. If, however following SQL statement is executed:

```
INSERT INTO student (StudentID, Name, Age, GENDER)
VALUES (1006, 'Emma', 16, 'Female');
```

There won't be any database update and as the age < 17 years. Different options to use **Check constraint**:

With alter: Check constraint can also be added to an already created relation using the syntax:

```
ALTER TABLE TABLE_NAME
MODIFY COLUMN_NAME
CHECK (Predicate);
```

Giving variable name to check constraint:Check constraints can be given a variable name using the syntax:

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT CHECK_CONST
CHECK (Predicate);
```

Remove check constraint: Check constraint can be removed from the relation in the database from SQL server using the syntax:

```
ALTER TABLE TABLE_NAME
DROP CONSTRAINT CHECK_CONSTRAINT_NAME;
```

Drop check constraint: Check constraint can be dropped from the relation in the database in MySQL using the syntax:

```
ALTER TABLE TABLE_NAME
DROP CHECK CHECK_CONSTRAINT_NAME;
```

View existing constraints on a particular table

> If you want to check if a constraint or any constraint exists within the table in mysql then you can use the following command. This command will show a tabular output of all the constraint-related data for the table name you’ve passed in the statement, in our case we’ll use the employee table.
>
> ```
> SELECT * FROM information_schema.table_constraints
> WHERE table_schema = schema() AND table_name = 'employee';
> ```