SQL Constraints

In this lesson, we will take a look at the different constraints enforced on columns and tables.

SQL constraints

Constraints are the rules enforced on data columns in a table. These 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 database.

Constraints can either be column level or table level. Column level constraints are applied only to one column, whereas table level constraints are applied to the entire table.

The following are some of the most commonly used constraints available in SQL:

The NOT NULL constraint

Ensures that a column cannot have a `NULL` value. By default, a column can hold `NULL` values. If you do not want a column to have a `NULL` value, then you need to define such a constraint on this column specifying that `NULL` is now not allowed for that column. A `NULL` is not the same as no data, rather, it represents unknown data.

The DEFAULT constraint

Provides a default value for a column when none is specified.

The UNIQUE constraint

Ensures that all the values in a column are different. The UNIQUE constraint prevents two records from having identical values in a column. In the CUSTOMER table, for example, you might want to prevent two or more people from having the same ID.

The PRIMARY key constraint

Uniquely identifies each row/record in a database table. A primary key is a field that uniquely identifies each row/record in a database table. A primary key must contain unique values and cannot have `NULL` values. A table can have only one primary key, which may consist of single or multiple fields. When multiple fields are used as a primary key, they are called a composite key.

The FOREIGN Key constraint

Uniquely identifies a row/record in any other database table. A foreign key is a key used to link two tables together. This is sometimes also called a referencing key. It can be a column or a combination of columns whose values match a primary key in a different table.

The CHECK constraint

Ensures that all values in a column satisfy certain conditions. It enables a condition to check the value being entered into a record. If the condition evaluates to false, the record violates the constraint and isn't entered into the table. For example, in the CUSTOMER table, we can check if the customer is over 18 years old by applying the CHECK constraint on the `Age` attribute (column).

Get excited as we are going to write our first query in the next lesson.