**What are SQL constraints?**

SQL constraints are a set of rules implemented on tables in relational databases to dictate what data can be inserted, updated or deleted in its tables. This is done to ensure the accuracy and the reliability of information stored in the table. Constraints enforce limits to the data or type of data that can be inserted/updated/deleted from a table. The purpose of constraints is to maintain the data integrity during an update/delete/insert into a table.  Once the constraint is placed, if any operation in the database does not follow the rules specified by the constraint, the particular operation is aborted. In this article, we will go through what SQL constraints are, what are the different kinds of SQL constraints are commonly used and how to implement and get rid of them. First, however, we will take a brief look into why they are needed.

## ****Types of SQL Constraints****

SQL constraints can be at a column or a table level. Column level constraints apply to specific columns in a table and do not specify a column name except the check constraints. They refer to the column that they follow. The names are specified by the Table-level constraints of the columns to which they apply.

Following is a list of the most commonly used column and table level SQL constraints:

Column Level Constraints include:

* NOT NULL Constraint
* UNIQUE Constraint
* DEFAULT Constraint
* CHECK Constraint
* PRIMARY KEY Constraint
* FOREIGN KEY Constraint

Column Level Constraints include:

* UNIQUE Constraint
* CHECK Constraint
* PRIMARY KEY Constraint
* FOREIGN KEY Constraint
* INDEX Constraint

Let us now dive into the world of SQL constraints! We will browse in detail what each constraint is, why we use them and how to apply and remove them.

### The NOT NULL Constraint

A NOT NULL constraint specifies that no cell value for any row in this column can be blank. Generally, this rule is applied to columns that capture information that is absolutely vital to identify and extract data from a table. Continuing the Sales table example, Sale\_Id and the Sales\_Amount would be potential columns for applying the NOT NULL constraint.

Applying the NOT NULL constraint:

The NOT NULL constraint can be defined either during the creation of the table or can be put in place later via an alter statement.

Declaring a NOT NULL Constraint during the Creation of a Table:

### The DEFAULT Constraint

The DEFAULT constraint is used to specify a default value that is to be entered in any record in a particular column is left blank. The default value will be added to all new records if no other value is specified.

**Applying the DEFAULT Constraint:**

The DEFAULT constraint can be defined either during the creation of the table or can be put in place later via an alter statement.

**Declaring a DEFAULT Constraint during the Creation of a Table:**

The following SQL sets a DEFAULT value for the ‘Vendor\_Name’ column when the ‘Sales’ table is created: