Databases and SQLᵝ → Basics of SQL → The NULL value

Theory: The NULL value

🕒 12 minutes 0 / 5 problems solved

Skip this topic Start practicing

2811 users solved this topic. Latest completion was about 5 hours ago.

Imagine you've created a table to store data about Nobel prizes. You defined the following columns: year, field, winner's name, and year of winner's birth to evaluate the age of winners. However, for some winners, the year of birth is undefined, for instance, for the European Union citizens that received the Nobel Peace Prize in 2012. You may put some special value like 0 in these cells, but it may potentially lead to mistakes. For such cases, SQL has a special value: **NULL**. In this topic we'll discuss how to use it.

§1. NULL value

NULL is used in SQL to indicate that some data value is unknown or undefined. It should not be confused with a value of 0 or ''. Arithmetic or string expressions with NULL among the operands are evaluated as NULL, for example, $2 + 2 * NULL$ equals NULL.

A NULL value can be stored in a column of any type. However, a software engineer may use a **NOT NULL constraint** in ***create table*** statement to specify that a column should not store NULL values. For example, according to this code, only the column "winner_birth_year" may contain NULLs:

```
CREATE TABLE winners (
    year INTEGER NOT NULL,
    field VARCHAR(20) NOT NULL,
    winner_name VARCHAR(100) NOT NULL,
    winner_birth_year INTEGER);
```

§2. Comparisons with NULL

NULL value basically means “could be anything”. Because of this, comparisons with NULL can never result in either TRUE or FALSE, but always in a third logical result, **UNKNOWN.** It is, actually, the NULL value of boolean type and some data management systems do not distinguish a special UNKNOWN value. For example, the result of each of the following comparisons is UNKNOWN:

- NULL = 1
- NULL <> 1
- NULL > 1
- NULL = '1'
- NULL = NULL

Nothing equals NULL; not even NULL equals NULL!

How can we check whether an expression or a value is null or not? SQL supports special predicates for that: **IS NULL** and **IS NOT NULL**. For example, queries below both return TRUE as a result:

```
SELECT 0+NULL IS NULL;
SELECT '' IS NOT NULL;
```

The following queries are evaluated as FALSE:

```
SELECT NULL IS NOT NULL;
SELECT 1-1 IS NULL;
```

§3. TRUE, FALSE, and UNKNOWN

In logic expressions with UNKNOWN operands, the result is UNKNOWN if it depends on an operand that is UNKNOWN. Thus, in contrast to comparisons, the result of a logic expression can be something other than UNKNOWN

Current topic:

The NULL value ···

Topic depends on:

✕ Expressions ···

Topic is required for:

Special SQL functions - CASE, COALESCE ···

Table of contents:

even though one operand is UNKNOWN. Let's consider the following examples:

- (NULL = 1) AND TRUE evaluates to UNKNOWN (the result would be TRUE only if both operands were TRUE),
- (NULL = 1) OR TRUE equals to TRUE (the result is TRUE because at least one operand is TRUE).

Now you know how to work with data when there is actually no data and use NULL values where they are needed.

Report a typo

231 users liked this theory. 0 didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Start practicing

Comments (1) Hints (0) Useful links (0) Show discussion