Theory: Literals

10 minutes 0 / 5 problems solved

Skip this topic Start practicing

§1. Introduction

In almost any program or data analysis script, you have to operate with constant values called **literals**. For example, if you're analyzing census data and need to extract census rows according to specific criteria (individuals named Jessie born between 1920 and 2000), you often have to use literals (Jessie, 1920, and 2000).

In this topic, we'll discuss three basic types of literals: numeric, string, and boolean. To apply this newly acquired knowledge right away, we'll write a "Hello world!" program.

§2. String literals

A string constant in SQL is a sequence of characters surrounded by single (') or double (") for example, 'Hyperskill', 'Hello world!', and "SQL (Structured Query Language)". To include a single-quote character within a string literal wrapped in single quotes, type two adjacent single quotes, for example, 'Jessie''s birthday'. Alternatively, wrap the literal in double quotes so that single quote will be treated as a regular symbol. For example, "Sophie's choice".

§3. Numeric literals

Numeric literals can be either positive or negative numbers specified as *integer* (for example, 1, +9000, -256), *decimal* (for example, 2.3, +876.234, -1024.0), or real values in *exponential* notation (for example, 0.4e3, +7.192834e+10, -4.0e-5). If you do not specify the sign, then a positive number is assumed by default.

Numeric literals may be INTEGER, REAL, and DECIMAL; the data management system automatically defines its type based on the context. For example, if you specify numeric value without decimal point that fits INTEGER range of values, the system will treat it as INTEGER, and otherwise as DECIMAL. Numeric values specified in exponential notation are treated as REAL data.

You can directly specify the type of a literal using the **CAST(*value* AS *type*) function**. Instead of placeholders ***value*** and ***type,*** you can use your literal and type, for example, CAST(1 AS DECIMAL(20,3)) interprets numeric value 1 as DECIMAL(20,3). Of course, you can specify only meaningful type casts: for instance, CAST(123.4 AS INTEGER) would cause an error.

§4. Boolean literals

Boolean literals are boolean logic truth values: TRUE (true) and FALSE (false). No matter how you specify the value (TRUE or true), these values are identical boolean literals (same applies to FALSE values).

§5. Hello, World

Now we are ready to write a traditional "Hello, World!" program. This SQL code (actually, a single query) implements it:

```
1 SELECT 'Hello, World!';
```

The query evaluation result is the following (its actual representation may be different depending on the environment you run it):

```
Hello, World!
```

Current topic:

Literals

Topic depends on:

× Basic data types

Topic is required for:

Expressions

Basic DELETE statement

Table of contents:

↑ Literals

§1. Introduction

§2. String literals

§3. Numeric literals

§4. Boolean literals

§5. Hello, World

§6. Conclusion

Feedback & Comments

Actually, the query declaratively states that we want to select this string as a result. The statement consists of 3 parts: the **keyword SELECT** (case insensitive), the literal that we want to receive, and a semicolon that defines the end of the query.

Summarizing, a simple SQL query that extracts any literal, whether it's a string, numeric, or boolean, looks as follows (you may replace `literal` with any correctly specified constant you want):

```
1 SELECT literal;
```

§6. Conclusion

Now you know how to define numeric, string, and boolean constant values and say "Hello, Data!" – and even more – in SQL.

Report a typo

317 users liked this theory. **10** didn't like it. **What about you?**

Start practicing

Comments (5) Hints (0) Useful links (2) Show discussion