

# Theory: Basic data types

🕒 15 minutes   0 / 4 problems solved

Skip this topic

Start practicing

4506 users solved this topic. Latest completion was about 1 hour ago.

## §1. Introduction

As you already know, SQL is a language used for working with different types of data organized into a table.

Usually, data values from the same column in a table have the same meaning and type. For example, a table *Car* may look like this:

Table “Car”					
manufacture year	color	horsepower	price	electricity	...
2018	red	283	34.990	TRUE	...
2019	black	313	50.00	FALSE	...
...	...	...	...	...	...

We see that values in *manufacture year* column are integer numbers, values in *price* are decimal, and values in *electricity* are boolean. SQL databases usually require that each column in a database table has a name and a **data type**. The column *data type* restricts the set of values that can be stored in the column and defines the set of possible operations on them.

ANSI standard defines a pretty complex set of data types. Besides, database vendors usually add their own non-standard data options. In this topic, we will consider a very basic subset of data types: **INTEGER**, **REAL**, **DECIMAL**, **VARCHAR**, and **BOOLEAN**.

## §2. Integer

**INTEGER** is a numeric data type that represents some range of mathematical integers. Usually, database management systems support a range from -2147483648 to +2147483647, which is enough for the majority of business tasks.

**INTEGER** type is good for counters, numeric identifiers, and any integer business value you can imagine that fits the scale range. For example, in a database for an advertising company you may have a table that stores a banner identifier, a number of clicks on it, and a number of banner impressions: all in **INTEGER** columns.

## §3. Real

Numeric data, of course, goes beyond integer values: for example, consider the distance from the Earth to Mars (225.0e+06 km), the white cell count in blood (from 1.2e+09 to 3.0e+09 per liter), or electron’s mass (9.10938356e-31 kg). For such real values, SQL has a special data type with an intuitive name **REAL**. This data type is often called **FLOAT** reflecting the way the numbers are stored internally: exponential form is utilized and thus decimal point may “float”, that is, it can be placed anywhere relative to the significant digits of the number.

This numeric type is usually called **inexact** because some values are stored as approximations (for example, 0.1), so storing and retrieving a value might not always work as you’d expect. Thus, if you require exact storage and calculations (for example, for billing database), **REAL** numeric data is not your choice. However, **REAL** data is often found in systems that operate very small and very large real numbers and require fast processing time. For example, the distance between galaxies or the diameter of an atomic nucleus can be stored in one column of **REAL** type. In most systems, the **REAL** type has a scale range from 1e-37 to 1e+37 with a precision of at least 6 decimal digits; absolute accuracy for small numbers is higher than for larger ones.

Current topic:

[Basic data types](#) ...

Topic depends on:

✗

[What is SQL](#) ...

Topic is required for:

[Literals](#) ...

[Basic CREATE statement](#) ...

Table of contents:

1

[Basic data types](#)

§1.

[Introduction](#)

§2.

[Integer](#)

§3.

[Real](#)

§4.

[Decimal](#)

§5.

[Text](#)

§6.

[Boolean](#)

§7.

[Who defines types, and how?](#)

§8.

[Conclusion](#)

[Feedback & Comments](#)

## §4. Decimal

In everyday life, we usually face decimal numbers quite a lot: for example, when measuring body temperature (36.6 degrees Celsius) or weight (52.7 kg), estimating the height of the famous Tower of Pisa (55.86 m), and, of course, counting our precious finances (\$103050.79). SQL supports a special data type for such values – **DECIMAL (precision, scale)**.

As you see, this type has two parameters: precision and scale. **Scale** defines the count of digits in the fractional part, to the right of the decimal point. **Precision** is the total count of digits in the number, that is, on both sides of the decimal point. Scale, as you may have concluded, cannot exceed precision.

The DECIMAL type is usually referenced as an accurate numeric type with unlimited precision and scale; however, there is a limit – up to 131072 digits before the decimal point and up to 16383 digits after the decimal point. Come to think of it, that should be enough!

One of the most important differences between DECIMAL and REAL types for applications is *rounding rules*. DECIMAL values round intuitively and predictably, for instance, rounding -0.5 will result in -1. Rounding rules for REAL data type may be less intuitive, and here rounding -0.5 will give us -0. However, arithmetic operations on REAL types are faster and values themselves occupy less storage space.

## §5. Text

Of course, one may want to process something other than numeric data, and SQL supports a family of data types designed to represent text data. Let's consider one of them, quite a universal and basic one – **VARCHAR(n)**.

This type represents a string of symbols of varying length not longer than *n*. For example, one can insert the strings "apple", "plum", and "peach" into a column with the type VARCHAR(5). The strings "orange" and "banana" will exceed the length restriction and the system will either truncate them or generate an error if one tries to insert such long values.

## §6. Boolean

The **BOOLEAN** type represents boolean logic (truth) values: either TRUE or FALSE. This simple data type can be utilized for any attributes with flag semantics, for example, whether a client has visited a competitor's site.

## §7. Who defines types, and how?

As a database user, you should just know the types of table columns you utilize to be able to process them correctly. However, as a software engineer, you should know how to create a table and define the column types.

Let's consider an example of an SQL query that defines a table "census" with 5 columns: "id" of type INTEGER, "name" of type VARCHAR(20), "birth\_place\_latitude" of type REAL, "year\_income" of type DECIMAL(20,3), and "is\_parent" of type BOOLEAN.

```
1 CREATE TABLE census (  
2     id INTEGER,  
3     name VARCHAR(20),  
4     birth_place_latitude REAL,  
5     year_income DECIMAL(20,3),  
6     is_parent BOOLEAN  
7 );
```

One may see the following pattern:

```
1 CREATE TABLE table_name (  
2     column_name_1 column_type_1,  
3     ...,  
4     column_name_n column_type_n  
5 );
```

## §8. Conclusion

Data may be very diverse, and SQL supports an extensive set of data types to represent this diversity. We have discussed a basic subset of data types just to start with, yet there is more to the topic: type casting, compound types, special types for numeric data, text, and timestamps, and so on.

 Report a typo

**397** users liked this theory. **4** didn't like it. What about you?



Start practicing

[Comments \(10\)](#)

[Hints \(0\)](#)

[Useful links \(2\)](#)

[Show discussion](#)