

PostgreSQL – Data Types

 [geeksforgeeks.org/postgresql-data-types](https://www.geeksforgeeks.org/postgresql-data-types)

In this article, we will look into the various data types available in PostgreSQL. The following data types are supported by PostgreSQL:

- **Boolean**
- **Character** Types [such as *char*, *varchar*, and *text*]
- **Numeric** Types [such as *integer* and *floating-point number*]
- **Temporal** Types [such as *date*, *time*, *timestamp*, and *interval*]
- **UUID** [for storing *UUID (Universally Unique Identifiers)*]
- **Array** [for storing *array strings, numbers, etc.*]
- **JSON** [stores *JSON data*]
- **hstore** [stores *key-value pair*]
- **Special** Types [such as *network address and geometric data*]

Now let's get an overview of the above-mentioned data types.

Boolean:

In PostgreSQL, the “bool” or “boolean” keyword is used to initialize a Boolean data type. These data types can hold *true*, *false*, and *null* values. A boolean data type is stored in the database according to the following:

- 1, yes, y, t, true values are converted to true
- 0, no, false, f values are converted to false

When queried for these boolean data types are converted and returned according to the following:

- t to true
- f to false
- space to null

Characters :

PostgreSQL has three character data types namely, **CHAR(n)**, **VARCHAR(n)**, and **TEXT**.

- *CHAR(n)* is used for data(string) with a fixed-length of characters with padded spaces. In case the length of the string is smaller than the value of “n”, then the rest of the remaining spaces are automatically padded. Similarly for a string with a length greater than the value of “n”, PostgreSQL throws an error.
- *VARCHAR(n)* is the variable-length character string. Similar to *CHAR(n)*, it can store “n” length data. But unlike *CHAR(n)* no padding is done in case the data length is smaller than the value of “n”.

- *TEXT* is the variable-length character string. It can store data with unlimited length.

Numeric:

PostgreSQL has 2 types of numbers namely, integers and floating-point numbers.

1. Integer:

- *Small integer* (SMALLINT) has a range -32, 768 to 32, 767 and has a size of 2-byte.
- *Integer* (INT) has a range -2, 147, 483, 648 to 2, 147, 483, 647 and has a size of 4-byte.
- **Serial** (SERIAL) works similar to the integers except these are automatically generated in the columns by PostgreSQL.

2. Floating-point number:

- *float(n)* is used for floating-point numbers with n precision and can have a maximum of 8-bytes.
- *float8* or *real* is used to represent 4-byte floating-point numbers.
- A real number N(d,p) meaning with d number of digits and p number of decimal points after, are part of *numeric* or *numeric(d, p)*. These are generally very precise.

Temporal data type:

This data type is used to store date-time data. PostgreSQL has 5 temporal data type:

- DATE is used to store the dates only.
- TIME is used to stores the time of day values.
- TIMESTAMP is used to stores both date and time values.
- TIMESTAMPTZ is used to store a timezone-aware timestamp data type.
- INTERVAL is used to store periods of time.

Arrays:

In PostgreSQL, an array column can be used to store an array of strings or an array of integers etc. It can be handy when storing data likes storing days of months, a year, or even a week, etc.

JSON:

PostgreSQL supports 2 types of JSON types namely JSON and JSONB(Binary JSON). The JSON data type is used to store plain JSON data that get parsed every time it's called by a query. Whereas the JSONB data type is used to store JSON data in a binary format. It is one hand makes querying data faster whereas slows down the data insertion process as it supports indexing of table data.

UUID:

The UUID data type allows you to store Universal Unique Identifiers defined by RFC 4122. The UUID values guarantee a better uniqueness than SERIAL and can be used to hide sensitive data exposed to the public such as values of id in URL.

The UUID stands for Unique Universal Identifiers. These are used to give a unique ID to a data that is unique throughout the database. The UUID data type are used to store UUID of the data defined by RFC 4122. These are generally used to protect sensitive data like credit card informations and is better compared to SERIAL data type in the context of uniqueness.

Special data types:

In addition to the primitive data types, PostgreSQL also supports some special data types that are related to network or geometric. These special data types are listed below:

- box: It is used to store rectangular box.
- point: It is used to store geometric pair of numbers.
- lseg: It is used to store line segment.
- point: It is used to store geometric pair of numbers.
- polygon: It is used to store closed geometric.
- inet: It is used to store an IP4 address.
- macaddr: It is used to store a MAC address.