



Search Documentation:  Search

[Home](#) → [Documentation](#) → [Manuals](#) → [PostgreSQL 9.5](#)

This page in other versions: [9.3](#) / [9.4](#) / **9.5** / [9.6](#) / [current](#) (10) | Development versions: [devel](#) / [11](#) | Unsupported versions: [7.1](#) / [7.2](#) / [7.3](#) / [7.4](#) / [8.0](#) / [8.1](#) / [8.2](#) / [8.3](#) / [8.4](#) / [9.0](#) / [9.1](#) / [9.2](#)

**[PostgreSQL 9.5.14 Documentation](#)**

[Prev](#)      [Up](#)      [Next](#)

# Chapter 8. Data Types

## Table of Contents

### 8.1. [Numeric Types](#)

- 8.1.1. [Integer Types](#)
- 8.1.2. [Arbitrary Precision Numbers](#)
- 8.1.3. [Floating-Point Types](#)
- 8.1.4. [Serial Types](#)

### 8.2. [Monetary Types](#)

### 8.3. [Character Types](#)

### 8.4. [Binary Data Types](#)

- 8.4.1. [bytea Hex Format](#)
- 8.4.2. [bytea Escape Format](#)

### 8.5. [Date/Time Types](#)

- 8.5.1. [Date/Time Input](#)
- 8.5.2. [Date/Time Output](#)
- 8.5.3. [Time Zones](#)
- 8.5.4. [Interval Input](#)
- 8.5.5. [Interval Output](#)

### 8.6. [Boolean Type](#)

### 8.7. [Enumerated Types](#)

- 8.7.1. [Declaration of Enumerated Types](#)
- 8.7.2. [Ordering](#)
- 8.7.3. [Type Safety](#)
- 8.7.4. [Implementation Details](#)

### 8.8. [Geometric Types](#)

- 8.8.1. [Points](#)
- 8.8.2. [Lines](#)
- 8.8.3. [Line Segments](#)
- 8.8.4. [Boxes](#)
- 8.8.5. [Paths](#)
- 8.8.6. [Polygons](#)

### 8.8.7. [Circles](#)

## 8.9. [Network Address Types](#)

### 8.9.1. [inet](#)

### 8.9.2. [cidr](#)

### 8.9.3. [inet vs. cidr](#)

### 8.9.4. [macaddr](#)

## 8.10. [Bit String Types](#)

## 8.11. [Text Search Types](#)

### 8.11.1. [tsvector](#)

### 8.11.2. [tsquery](#)

## 8.12. [UUID Type](#)

## 8.13. [XML Type](#)

### 8.13.1. [Creating XML Values](#)

### 8.13.2. [Encoding Handling](#)

### 8.13.3. [Accessing XML Values](#)

## 8.14. [JSON Types](#)

### 8.14.1. [JSON Input and Output Syntax](#)

### 8.14.2. [Designing JSON documents effectively](#)

### 8.14.3. [jsonb Containment and Existence](#)

### 8.14.4. [jsonb Indexing](#)

## 8.15. [Arrays](#)

### 8.15.1. [Declaration of Array Types](#)

### 8.15.2. [Array Value Input](#)

### 8.15.3. [Accessing Arrays](#)

### 8.15.4. [Modifying Arrays](#)

### 8.15.5. [Searching in Arrays](#)

### 8.15.6. [Array Input and Output Syntax](#)

## 8.16. [Composite Types](#)

### 8.16.1. [Declaration of Composite Types](#)

### 8.16.2. [Constructing Composite Values](#)

### 8.16.3. [Accessing Composite Types](#)

### 8.16.4. [Modifying Composite Types](#)

### 8.16.5. [Using Composite Types in Queries](#)

### 8.16.6. [Composite Type Input and Output Syntax](#)

## 8.17. [Range Types](#)

### 8.17.1. [Built-in Range Types](#)

### 8.17.2. [Examples](#)

### 8.17.3. [Inclusive and Exclusive Bounds](#)

### 8.17.4. [Infinite \(Unbounded\) Ranges](#)

### 8.17.5. [Range Input/Output](#)

### 8.17.6. [Constructing Ranges](#)

### 8.17.7. [Discrete Range Types](#)

8.17.8. [Defining New Range Types](#)

8.17.9. [Indexing](#)

8.17.10. [Constraints on Ranges](#)

8.18. [Object Identifier Types](#)

8.19. [pg\\_lsn Type](#)

8.20. [Pseudo-Types](#)

PostgreSQL has a rich set of native data types available to users. Users can add new types to PostgreSQL using the [CREATE TYPE](#) command.

[Table 8-1](#) shows all the built-in general-purpose data types. Most of the alternative names listed in the "Aliases" column are the names used internally by PostgreSQL for historical reasons. In addition, some internally used or deprecated types are available, but are not listed here.

**Table 8-1. Data Types**

Name	Aliases	Description
bigint	int8	signed eight-byte integer
bigserial	serial8	autoincrementing eight-byte integer
bit [ (n) ]		fixed-length bit string
bit varying [ (n) ]	varbit [ (n) ]	variable-length bit string
boolean	bool	logical Boolean (true/false)
box		rectangular box on a plane
bytea		binary data ("byte array")
character [ (n) ]	char [ (n) ]	fixed-length character string
character varying [ (n) ]	varchar [ (n) ]	variable-length character string
cidr		IPv4 or IPv6 network address
circle		circle on a plane
date		calendar date (year, month, day)
double precision	float8	double precision floating-point number (8 bytes)
inet		IPv4 or IPv6 host address
integer	int, int4	signed four-byte integer
interval [ fields ] [ (p) ]		time span
json		textual JSON data
jsonb		binary JSON data, decomposed
line		infinite line on a plane
lseg		line segment on a plane
macaddr		MAC (Media Access Control) address
money		currency amount
numeric [ (p, s) ]	decimal [ (p, s) ]	exact numeric of selectable precision
path		geometric path on a plane
pg_lsn		PostgreSQL Log Sequence Number

Name	Aliases	Description
point		geometric point on a plane
polygon		closed geometric path on a plane
real	float4	single precision floating-point number (4 bytes)
smallint	int2	signed two-byte integer
smallserial	serial2	autoincrementing two-byte integer
serial	serial4	autoincrementing four-byte integer
text		variable-length character string
time [ (p) ] [ without time zone ]		time of day (no time zone)
time [ (p) ] with time zone	timetz	time of day, including time zone
timestamp [ (p) ] [ without time zone ]		date and time (no time zone)
timestamp [ (p) ] with time zone	timestamptz	date and time, including time zone
tsquery		text search query
tsvector		text search document
txid_snapshot		user-level transaction ID snapshot
uuid		universally unique identifier
xml		XML data

**Compatibility:** The following types (or spellings thereof) are specified by SQL: bigint, bit, bit varying, boolean, char, character varying, character, varchar, date, double precision, integer, interval, numeric, decimal, real, smallint, time (with or without time zone), timestamp (with or without time zone), xml.

Each data type has an external representation determined by its input and output functions. Many of the built-in types have obvious external formats. However, several types are either unique to PostgreSQL, such as geometric paths, or have several possible formats, such as the date and time types. Some of the input and output functions are not invertible, i.e., the result of an output function might lose accuracy when compared to the original input.

---

[Prev](#)
[Home](#)
[Next](#)  
 WITH Queries (Common Table Expressions)
 [Up](#)
 Numeric Types

## Submit correction

If you see anything in the documentation that is not correct, does not match your experience with the particular feature or requires further clarification, please use [this form](#) to report a documentation issue.

[Privacy Policy](#) | [About PostgreSQL](#)

Copyright © 1996-2018 The PostgreSQL Global Development Group