Data Types in PostgreSQL

PDP ACADEMY

Boolean data type

BOOLEAN that can have three values: true, false and NULL.

True	False
true	false
‘t’	‘f ‘
‘true’	‘false’
‘y’	‘n’
‘yes’	‘no’
‘1’	‘0’

Character data types

PostgreSQL provides **three primary character types.**

- CHARACTER(n) or CHAR(n)
- CHARACTER VARYINGING(n) or VARCHAR(n)
- TEXT

Character Types	Description
CHARACTER VARYING(n), VARCHAR(n)	variable-length with length limit
CHARACTER(n), CHAR(n)	fixed-length, blank padded
TEXT	variable unlimited length

Numeric data type

Numeric types consist of two-, four-, and eight-byte integers, four- and eight-byte floating-point numbers, and selectable-precision decimals.

documentation

DATE data type

PostgreSQL uses 4 bytes to store a date value. The lowest and highest values of the DATE data type are **4713 BC** and **5874897 AD**.

When storing a date value, PostgreSQL uses the **yyyy-mm-dd** format (2000-12-31). It also uses this format for inserting data into a date column.

PostgreSQL DATE functions

- Get the current date
- Output a PostgreSQL date value in a specific format(`TO_CHAR`)
- Get the interval between two dates(`INTERVAL`)
- Calculate ages in years, months, and days(`AGE`)
- Extract year, quarter, month, week, day from a date value(`EXTRACT`)

Interval data type

The **interval** data type allows you to store and manipulate a period of time in years, months, days, hours, minutes, seconds, etc.

Intervalstyle => [sql_standard, postgres, postgres_verbose, iso_8601]

documentation

timestamp data types

PostgreSQL provides you with two temporal data types for handling timestamp:

- timestamp: a timestamp without timezone one.
- timestamptz: timestamp with a timezone.

The timestamp data type allows you to store both date and time. However, it does not have any time zone data. It means that when you change the timezone of your database server, the timestamp value stored in the database will not change automatically.

The timestamptz datatype is the timestamp with the time zone. The timestamptz datatype is a time zone-aware date and time data type. Both timestamp and timestamptz uses 8 bytes for storing the timestamp

- `SELECT NOW();`
- `SELECT CURRENT_TIMESTAMP;`
- `SHOW timezone`
- `SET timezone;`
- `pg_timezone_names;`
- `SELECT TIMEOFDAY();`
- `SELECT timezone(tz, timezone);`

wikipedia

PDP ACADEMY

TIME data type

TIME data type allows to store the time of day values.

- `HH:MI:SS(00:00:00)`
- `SELECT CURRENT_TIME(precision);`
    - `0 <= precision <= 6`
- `SELECT LOCALTIME(precision);`
    - `0 <= precision <= 6`
- `SELECT LOCALTIME AT TIME ZONE 'UTC-5';`
- `column TIME with time zone;`

UUID data type

A UUID value is 128-bit quantity generated by an algorithm that make it unique in the known universe using the same algorithm. The following shows some examples of the UUID values:

- `select gen_random_uuid();`
- `select * from pg_extension;`
- `CREATE EXTENSION "uuid-ossp";`
- `select uuid_generate_v1();`
- `select uuid_generate_v4();`

ARRAY data type

Array plays an important role in PostgreSQL. Every data type has its own companion array type e.g., integer has an integer[] array type, character has character[] array type, etc.

PostgreSQL allows you to define a column to be an array of any valid data type including built-in type, user-defined type or enumerated type.

- `unnest()`

HSTORE data type

The hstore module implements the hstore data type for storing key-value pairs in a single value. The hstore data type is very useful in many cases, such as semi-structured data or rows with many attributes that are rarely queried.

- `CREATE EXTENSION hstore;`
- `select column ->'key' … ,`
- `… SET column = column || '"key"=>"value"' :: hstore;`
- `… SET column = delete(column, 'key');`
- `… WHERE column ? 'key';`
- `… WHERE column @> '”key”=>”value”';`
- `… WHERE ?& ARRAY [ 'key1', 'key2' ];`
- `… WHERE ?| ARRAY [ 'key1', 'key2' ];`
- `akeys(column)`
- `skeys(column)`
- `avals (attr)`
- `svals (attr)`
- `hstore_to_json (column)`

PDP ACADEMY

JSON data type

JSON stands for JavaScript Object Notation. JSON is an open standard format that consists of key-value pairs.

The main usage of JSON is to transport data between a server and a web application. Unlike other formats, JSON is human-readable text. PostgreSQL supports native JSON data type since version **9.2**. It provides many functions and operators for manipulating JSON data.

- `operators (->, ->>)`

Custom data types

It is possible to create custom data type using two ways using **CREATE DOMAIN** and **CREATE TYPE** statement.

- **CREATE DOMAIN** creates a user-defined data type with constraints such as **NOT NULL**, **CHECK**, **PATTERN MATCHING** etc.
- **CREATE TYPE** creates a composite type used in stored procedures as the data types of returned values.

PDP ACADEMY

Interview Questions