



Universidad Nacional Autónoma de México



Núñez Badillo Armando Adair

Bases de datos

Grupo 1

Tarea 9

Tipos de Datos en PostgreSQL

Semestre 2026-1

Profesor: Ing. Fernando Arreola

Fecha de entrega: 17/10/25

Numéricos

Se usan para almacenar números, ya sean enteros o decimales.

integer (o int)

Es el tipo de dato estándar para números enteros.

- **Ejemplo:** 43, 2304, -45000, 2147483647
- **Capacidad:** 4 bytes
- **Formato de representación:** Un número entero dentro del rango de -2,147,483,648\$ a \$+2,147,483,647\$.

numeric (o decimal)

Se usa para números que requieren una precisión exacta. Es **ideal para dinero** y cálculos financieros, ya que evita los problemas de redondeo de los tipos flotantes.

- **Ejemplo:** Se define con precisión y escala: numeric(10, 2). Esto significa 10 dígitos en total, con 2 después del punto decimal. Almacenaría 12345678.99.
- **Capacidad:** Variable. Depende de la precisión que se defina.
- **Formato de representación:** Número decimal exacto.

double precision (o float)

Para números de punto flotante (decimales inexactos). Útil para cálculos científicos donde la precisión extrema no es tan crítica como el rango.

- **Ejemplo:** 3.1415926535, 1.2345E+10
- **Capacidad:** 8 bytes
- **Formato de representación:** Precisión flotante de 15 dígitos decimales.

Carácter

Se usan para almacenar texto.

varchar(n)

Es el tipo de texto más común. (n) es el **límite máximo** de caracteres. Solo usa el espacio que necesita.

- **Ejemplo:** varchar(100) podría almacenar 'Hola Mundo mi nombre es Armando Nunez'.
- **Capacidad:** Variable. El tamaño del texto real más 1 o 4 bytes de sobrecarga (overhead).
- **Formato de representación:** Cadena de texto de longitud variable, hasta el límite n especificado.

text

Para cadenas de texto de longitud "ilimitada" (el límite es de 1 GB). Ideal por si no se sabe qué tan largo será un texto (como un comentario de usuario o un artículo de blog).

- **Ejemplo:** 'Érase una vez un texto blablablá...'
- **Capacidad:** Variable. El tamaño del texto real más 4 bytes de sobrecarga.
- **Formato de representación:** Cadena de texto de longitud variable sin un límite predefinido.

char(n)

Menos común. Es una cadena de texto de **longitud fija**. Si guardas un texto más corto que n, PostgreSQL lo rellenará con espacios hasta completar n.

- **Ejemplo:** char(2) para un código de país. Si guardas 'MX', se almacena 'MX'. Si guardaras 'U', se almacenaría 'U ' (con un espacio).
- **Capacidad:** Exactamente n bytes (para codificación estándar).
- **Formato de representación:** Cadena de texto de longitud fija, rellena con espacios si es necesario.

Fecha

Para almacenar fechas, horas o ambos.

timestamp (o timestamptz)

Almacena fecha y hora. La versión timestamptz (con zona horaria) es la más recomendada, ya que guarda la marca de tiempo en UTC y la convierte automáticamente a la zona horaria del cliente.

- **Ejemplo:** '2025-10-16 20:30:00'
- **Capacidad:** 8 bytes
- **Formato de representación:** Almacena fecha completa (año, mes, día) y hora (horas, minutos, segundos, y opcionalmente microsegundos).

date

Almacena únicamente la fecha (sin hora).

- **Ejemplo:** '2025-10-16'
- **Capacidad:** 4 bytes
- **Formato de representación:** Almacena año, mes y día.

interval

Un tipo especial para almacenar un "período" o "duración" de tiempo.

- **Ejemplo:** '1 year 2 months 3 days', '5 hours 30 minutes'
- **Capacidad:** 16 bytes
- **Formato de representación:** Almacena una duración en meses, días y segundos. Muy útil para cálculos como "sumar 30 días a esta fecha".

Y Otros 3 que me llamen la atención ejemplos, capacidad y formato de representación

boolean

El tipo de dato lógico.

- **Ejemplo:** true, false, o NULL (desconocido).

- **Capacidad:** 1 byte. Es muy eficiente.
- **Formato de representación:** Representa un estado binario: verdadero (t, true, yes, 1) o falso (f, false, no, 0).

jsonb

Para almacenar datos en formato JSON (JavaScript Object Notation), pero en una versión binaria optimizada. Es **extremadamente potente**.

- **Ejemplo:** '{ "cliente_id": 123, "tags": ["premium", "web"], "activo": true }'
- **Capacidad:** Variable (depende del tamaño del documento JSON).
- **Formato de representación:** Almacena el documento JSON en un formato binario que permite indexar (buscar muy rápido) *dentro* del JSON.

uuid

Un Identificador Único Universal (Universally Unique Identifier). Es un número de 128 bits que se genera de forma que es (prácticamente) único en todo el mundo.

- **Ejemplo:** 'a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11'
- **Capacidad:** 16 bytes.
- **Formato de representación:** Una cadena hexadecimal de 32 dígitos separada por guiones. Es la mejor opción para claves primarias (IDs) en sistemas distribuidos, ya que evita conflictos que podrías tener con un integer secuencial.

Referencias:

[1] PostgreSQL Global Development Group, "Chapter 8. Data Types," *PostgreSQL Documentation*. [Online]. Available: <https://www.postgresql.org/docs/current/datatype.html>. (accessed: Oct. 16, 2025).

[2] PostgreSQL Global Development Group, "8.1. Numeric Types," *PostgreSQL Documentation*. [Online]. Available: <https://www.postgresql.org/docs/current/datatype-numeric.html>. (accessed: Oct. 16, 2025).

[3] PostgreSQL Global Development Group, "8.4. Character Types," *PostgreSQL Documentation*. [Online]. Available: <https://www.postgresql.org/docs/current/datatype-character.html>. (accessed: Oct. 16, 2025).

[4] PostgreSQL Global Development Group, "8.5. Date/Time Types," *PostgreSQL Documentation*. [Online]. Available: <https://www.postgresql.org/docs/current/datatype-datetime.html>. (accessed: Oct. 16, 2025).

[5] PostgreSQL Global Development Group, "8.6. Boolean Type," *PostgreSQL Documentation*. [Online]. Available: <https://www.postgresql.org/docs/current/datatype-boolean.html>. (accessed: Oct. 16, 2025).

[6] PostgreSQL Global Development Group, "8.14. JSON Types," *PostgreSQL Documentation*. [Online]. Available: <https://www.postgresql.org/docs/current/datatype-json.html>. (accessed: Oct. 16, 2025).

[7] PostgreSQL Global Development Group, "8.12. UUID Type," *PostgreSQL Documentation*. [Online]. Available: <https://www.postgresql.org/docs/current/datatype-uuid.html>. (accessed: Oct. 16, 2025).