

TAREA 9 - Tipos de Datos en PostgreSQL

Vences Santillán Carlos Eduardo

17 de octubre de 2025

Tipos de Datos en PostgreSQL

PostgreSQL es uno de los sistemas de gestión de bases de datos más potentes y flexibles. Permite almacenar y manipular una amplia variedad de tipos de datos, desde números y texto hasta fechas, booleanos y estructuras más complejas. A continuación, se explican los principales grupos de tipos de datos con ejemplos.

1. Tipos Numéricos (NUMBER)

Los tipos numéricos se utilizan para representar valores enteros, decimales o de punto flotante. Algunos de los más comunes son:

- **SMALLINT:** Entero pequeño (de -32,768 a 32,767).
- **INTEGER o INT:** Entero estándar (de -2,147,483,648 a 2,147,483,647).
- **BIGINT:** Entero grande.
- **NUMERIC(p,s) o DECIMAL(p,s):** Números con precisión y escala definidas.
- **REAL / DOUBLE PRECISION:** Números de punto flotante (no exactos, más rápidos).

Ejemplo:

```
CREATE TABLE productos (  
    id SERIAL PRIMARY KEY,  
    precio NUMERIC(8,2),  
    stock INT  
);
```

2. Tipos de Caracteres (CHARACTER)

Se usan para almacenar texto, nombres o cadenas de caracteres. Los tipos más comunes son:

- **CHAR(n)**: Cadena de longitud fija. Si la cadena es más corta, se completa con espacios.
- **VARCHAR(n)**: Cadena de longitud variable con un límite máximo de caracteres.
- **TEXT**: Cadena de longitud ilimitada, ideal para descripciones largas.

Ejemplo:

```
CREATE TABLE clientes (  
    nombre VARCHAR(50),  
    correo TEXT  
);
```

3. Tipos de Fecha y Hora

Estos tipos permiten almacenar información temporal, como fechas y horas exactas.

- **DATE**: Guarda solo la fecha (año, mes, día).
- **TIME**: Guarda solo la hora.
- **TIMESTAMP**: Guarda fecha y hora combinadas.
- **TIMESTAMPTZ**: Guarda fecha y hora con zona horaria.
- **INTERVAL**: Representa periodos de tiempo (días, meses, años).

Ejemplo:

```
CREATE TABLE pedidos (  
    id SERIAL PRIMARY KEY,  
    fecha_pedido DATE,  
    fecha_entrega TIMESTAMP  
);
```

4. Tipos Booleanos (BOOLEAN)

Representan valores lógicos, como verdadero o falso. PostgreSQL acepta también valores equivalentes como t/f, yes/no, o 1/0.

Ejemplo:

```
CREATE TABLE usuarios (  
    id SERIAL PRIMARY KEY,  
    activo BOOLEAN DEFAULT TRUE  
);
```

5. Tipos Binarios (BYTEA)

Se usan para almacenar datos binarios, como imágenes, PDF o archivos.

Ejemplo:

```
CREATE TABLE documentos (  
    id SERIAL PRIMARY KEY,  
    archivo BYTEA  
);
```

6. Tipos JSON (JSON y JSONB)

Permiten almacenar información estructurada en formato JSON. La diferencia principal es que **JSONB** guarda los datos en formato binario, lo que mejora el rendimiento en búsquedas.

Ejemplo:

```
CREATE TABLE configuraciones (  
    id SERIAL PRIMARY KEY,  
    datos JSONB  
);  
  
INSERT INTO configuraciones (datos)  
VALUES ('{"tema": "oscuro", "idioma": "es"}');
```

7. Tipos Geométricos

PostgreSQL permite trabajar con datos espaciales simples, como coordenadas y figuras geométricas básicas.

- **POINT:** Par de coordenadas (x, y).
- **LINE:** Línea infinita definida por dos puntos.
- **CIRCLE:** Círculo definido por un punto central y un radio.

Ejemplo:

```
CREATE TABLE ubicaciones (  
    id SERIAL PRIMARY KEY,  
    posicion POINT  
);
```

Referencias

- PostgreSQL Global Development Group. (2024). *PostgreSQL 16 Documentation*. Recuperado de <https://www.postgresql.org/docs/current/datatype.html>
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill.
- Date, C. J. (2003). *An Introduction to Database Systems* (8th ed.). Addison-Wesley.