Universidad Nacional Autónoma de México.

Bases de datos.

Tipos de datos de Postgres.

Profesor: Fernando Arreola Franco.

Alumno: Oswaldo Flores Herrera.

Fecha: 04 de Abril 2025.

Flores Herrera Oswaldo.
Universidad Nacional Autónoma de México.

Tipos de Datos de Postgres.

Numéricos.

Respecto a nuestro primer tipo de dato numérico encontraremos que almacenar valores enteros, decimales y de punto flotante, cada uno con diferentes rangos y precisiones. Esta información es esencial para elegir el tipo de dato adecuado según los requisitos de almacenamiento y cálculo.

1. Tipos de Enteros (Integer Types)

Estos tipos almacenan números sin decimales y varían en tamaño y rango:

- SMALLINT (2 bytes): Rango: -32,768 a +32,767
- Ideal para valores pequeños con ahorro de espacio.

INTEGER o INT (4 bytes):

- Rango: -2,147,483,648 a +2,147,483,647
- El tipo más común para enteros en PostgreSQL.

BIGINT (8 bytes):

- Rango: -9,223,372,036,854,775,808 a +9,223,372,036,854,775,807
- Usado cuando se necesitan números muy grandes.

Tipos Autoincrementales (para claves primarias):

- SERIAL (basado en INTEGER):
    - Rango: 1 a 2,147,483,647
- BIGSERIAL (basado en BIGINT):
    - Rango: 1 a 9,223,372,036,854,775,807

2. Tipos de Punto Flotante (Floating-Point Types)

Útiles para números con decimales, pero con posibles errores de redondeo:

- REAL (4 bytes):
    - Precisión: ~6 dígitos decimales
    - Más rápido, pero menos preciso.
- DOUBLE PRECISION (8 bytes):
    - Precisión: ~15 dígitos decimales
    - Recomendado para cálculos científicos o estadísticos.

Flores Herrera Oswaldo.
Universidad Nacional Autónoma de México.

3. Tipos de Precisión Exacta (Exact Numeric Types)

Ideales para cálculos que requieren precisión absoluta (ej. operaciones financieras):

- DECIMAL / NUMERIC (tamaño variable):
    - Precisión configurable (ej: NUMERIC(10,2) para 10 dígitos totales, 2 decimales).
    - Sin límite de rango, pero depende de la precisión definida.

Carácter.

PostgreSQL dispone de tres alternativas principales para el almacenamiento de cadenas de texto:

Variantes disponibles

1. CHAR(n) - Texto de tamaño fijo
    - Siempre ocupa el espacio definido (completa con espacios si es necesario)
    - Por defecto equivale a CHAR(1) cuando no se indica tamaño
2. VARCHAR(n) - Texto de longitud variable
    - Permite establecer un límite máximo de caracteres
    - Útil cuando se requiere control dimensional
3. TEXT - Texto sin restricciones
    - Capacidad ilimitada para contenido extenso
    - Máxima versatilidad sin límites predefinidos

Funcionamiento destacable

- El sistema rechaza automáticamente textos que superen el tamaño establecido
- Caso especial: los espacios adicionales se eliminan sin generar error
- En conversiones explícitas, recorta el texto al tamaño máximo permitido
- VARCHAR sin especif icar tamaño funciona idénticamente a TEXT
- Rendimiento equivalente en todas las opciones (particularidad de PostgreSQL)

Flores Herrera Oswaldo.
Universidad Nacional Autónoma de México.

Fecha.

PostgreSQL ofrece un completo soporte para los tipos de datos de fecha y hora según el estándar SQL, permitiendo almacenar tanto momentos temporales específicos como intervalos de duración. La mayoría de estos tipos (excepto el tipo básico 'date') admiten una precisión de hasta microsegundos, y pueden configurarse con un parámetro opcional (como 'timetz(p)') para especificar el número de dígitos fraccionarios en los segundos. El tipo 'date' se destaca por almacenar únicamente valores calendarios con precisión diaria, sin componentes horarios. Además de los formatos tradicionales, PostgreSQL acepta valores especiales intuitivos como 'now' (que registra la hora exacta del sistema), 'yesterday', 'today' y 'tomorrow' (que se interpretan automáticamente como las 00:00 UTC de la fecha correspondiente). Esta combinación de precisión técnica y expresividad en los valores hace que los tipos temporales de PostgreSQL sean especialmente potentes para aplicaciones que requieren manejo detallado de fechas, horarios y cálculos de intervalos, manteniendo al mismo tiempo una sintaxis clara y accesible para los desarrolladores.

Tipos de datos de direcciones de redes.

PostgreSQL incluye tipos especializados para el manejo de direcciones de red, ofreciendo ventajas significativas sobre el almacenamiento en formato de texto convencional.

Principales Tipos de Red

1. Tipo INET
    - Almacena direcciones IPv4/IPv6 individuales o rangos
    - Formato flexible: acepta dirección con/sin máscara de red
    - Ejemplo: 192.168.1.1 o 192.168.1.0/24
2. Tipo CIDR
    - Especializado en rangos de red (no hosts individuales)
    - Aplica reglas estrictas de validación CIDR
    - Completa automáticamente direcciones parciales
3. Tipo MACADDR
    - Para direcciones físicas de hardware

- Acepta múltiples formatos de entrada
- Estandariza la salida a formato canónico

Beneficios Clave

- Validación automática de direcciones
- Operaciones especializadas para redes
- Eficiencia de almacenamiento frente a texto
- Ordenación consistente (IPv4 antes que IPv6)

Tipo de datos booleanos.

PostgreSQL implementa el tipo de dato lógico estándar de SQL, diseñado para representar valores de verdad binarios con una opción para estados indeterminados.

Representación de Valores

Valores positivos aceptados:

- Forma canónica: TRUE (opción preferida)
- Alternativas equivalentes: 't', 'true', 'y', 'yes', '1'

Valores negativos aceptados:

- Forma canónica: FALSE (opción preferida)
- Alternativas equivalentes: 'f', 'false', 'n', 'no', '0'

Estado especial:

- NULL representa incertidumbre o valor desconocido

Buenas Prácticas

1. Priorizar el uso de TRUE/FALSE para:
    - Mayor claridad de código
    - Cumplimiento con estándares SQL
    - Portabilidad entre sistemas

2. Casos de uso típicos:
    - Bandera de estado (activo/inactivo)
    - Indicadores binarios (sí/no)
    - Condiciones lógicas en consultas
3. Ventajas del tipo nativo:
    - Validación automática de valores
    - Eficiencia en almacenamiento
    - Operadores lógicos integrados

Flores Herrera Oswaldo.
Universidad Nacional Autónoma de México.

Bibliografías.

- “Data Types”. PostgreSQL Documentation. Accedido el 6 de abril de 2025. [En línea].Disponible: https://www.postgresql.org/docs/8.1/datatype.html#DATATYPE-NUMERIC
- Neon. “PostgreSQL Character Types: CHAR, VARCHAR, And TEXT”. Neon. Accedido el 6 de abril de 2025. [En línea]. Disponible: https://neon.tech/postgresql/postgresql-tutorial/postgresql-char-varchar-text