

Tareas BD

TAREA 6 - Tipos de datos

1. Datos Numéricos

| Tipo de Dato | Descripción | Capacidad de Representación | Forma de Representación | Limitaciones o Desventajas |
|---|--|--|-----------------------------------|---|
| <code>smallint</code> | Entero pequeño | -32,768 a 32,767 | 2 bytes | Rango limitado; adecuado solo para valores pequeños. |
| <code>integer</code> o <code>int</code> | Entero estándar | -2,147,483,648 a 2,147,483,647 | 4 bytes | Puede ser insuficiente para valores muy grandes. |
| <code>bigint</code> | Entero grande | -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 | 8 bytes | Mayor uso de almacenamiento; operaciones más lentas debido al tamaño. |
| <code>decimal(p,s)</code> o <code>numeric(p,s)</code> | Número exacto con precisión y escala definidas | Hasta 131,072 dígitos antes del punto decimal y hasta 16,383 después | Variable, depende de la precisión | Operaciones más lentas debido a la precisión exacta. |
| <code>real</code> | Coma flotante, precisión simple | ~6 dígitos decimales de precisión | 4 bytes | Inexactitud en cálculos debido a la representación en coma flotante. |
| <code>double precision</code> | Coma flotante, precisión doble | ~15 dígitos decimales de precisión | 8 bytes | Similar a <code>real</code> , pero con mayor precisión y uso de almacenamiento. |

Nota sobre `decimal(p,s)` o `numeric(p,s)` :

- **p (precisión):** Número total de dígitos.

- **s (escala):** Número de dígitos a la derecha del punto decimal.

Por ejemplo, `numeric(4,2)` puede almacenar valores desde -99.99 hasta 99.99.

2. Caracteres

| Tipo de Dato | Descripción | Capacidad de Representación | Forma de Representación | Limitaciones o Desventajas |
|---|---|-----------------------------|---|---|
| <code>char(n)</code> o <code>character(n)</code> | Cadena de longitud fija de <code>n</code> caracteres | Hasta 1 GB | Almacena exactamente <code>n</code> caracteres, rellenando con espacios si es necesario | Espacio desperdiciado si las cadenas son más cortas que <code>n</code> |
| <code>varchar(n)</code> o <code>character varying(n)</code> | Cadena de longitud variable hasta <code>n</code> caracteres | Hasta 1 GB | Almacena solo los caracteres proporcionados sin relleno | Rendimiento ligeramente menor en comparación con <code>text</code> debido a la verificación de longitud |
| <code>text</code> | Cadena de longitud ilimitada | Hasta 1 GB | Almacena solo los caracteres proporcionados | No permite restricciones directas de longitud |

3. Fecha

| Tipo de Dato | Descripción | Capacidad de Representación | Forma de Representación | Limitaciones o Desventajas |
|--|-------------------------------|-----------------------------|-------------------------|---|
| <code>date</code> | Fecha (año, mes, día) | 4713 AC a 5874897 DC | 4 bytes | No incluye información de hora |
| <code>time [(p)] [without time zone]</code> | Hora del día (sin fecha) | 00:00:00 a 24:00:00 | 8 bytes | No incluye información de fecha ni zona horaria |
| <code>time [(p)] with time zone</code> | Hora del día con zona horaria | 00:00:00+13 a 24:00:00-12 | 12 bytes | Puede ser confuso manejar zonas horarias |
| <code>timestamp [(p)] [without time zone]</code> | Fecha y hora sin zona horaria | 4713 AC a 294276 DC | 8 bytes | No considera diferencias de zona horaria |

| | | | | |
|---|-------------------------------|---------------------|---------|--|
| <code>timestamp [(p)] with time zone</code> | Fecha y hora con zona horaria | 4713 AC a 294276 DC | 8 bytes | La conversión de zonas horarias puede afectar el rendimiento |
|---|-------------------------------|---------------------|---------|--|

Nota sobre `p` (precisión): Define el número de dígitos en la parte fraccionaria de los segundos. Por ejemplo, `timestamp(3)` incluye milisegundos.

4. Tipos de Datos Especiales en PostgreSQL

PostgreSQL ofrece tipos de datos especializados para necesidades particulares:

| Tipo de Dato | Descripción | Capacidad de Representación | Forma de Representación | Limitaciones o Desventajas |
|----------------------|--|-----------------------------|---|--|
| <code>boolean</code> | Valores lógicos <code>TRUE</code> , <code>FALSE</code> o <code>NULL</code> | 1 byte | Almacena <code>true</code> o <code>false</code> | Puede no ser eficiente en términos de espacio si se almacenan muchos valores booleanos |
| <code>bytea</code> | Almacena datos binarios | Hasta 1 GB | Binario | Manejo complejo y puede requerir codificación/decodificación |