

Criando Bancos e Tabelas no PostgreSQL

16.5

Aula 9

Introdução a linguagem SQL

Tipos de Dados

FONTE: <https://www.postgresql.org/docs/current/>

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipos de dados – Tabela 1/4

Nome	Aliases	Descrição
bigint	int8	inteiro de oito bytes com sinal
bigserial	serial8	inteiro de oito bytes com autoincremento
bit [(n)]		cadeia de bits de comprimento fixo
bit varying [(n)]	varbit [(n)]	cadeia de bits de comprimento variável
boolean	bool	booleano lógico (verdade/falso)
box		caixa retangular no plano
bytea		dados binários (“matriz de bytes”)
character [(n)]	char [(n)]	cadeia de caracteres de comprimento fixo
character varying [(n)]	varchar [(n)]	cadeia de caracteres de comprimento variável
cidr		endereço de rede IPv4 ou IPv6
circle		círculo no plano
date		data de calendário (ano, mês, dia)

Tipos de dados – Tabela 2/4

Nome	Aliases	Descrição
double precision	float8	número de ponto flutuante de precisão dupla (8 bytes)
inet		endereço de hospedeiro IPv4 ou IPv6
integer	int, int4	inteiro de quatro bytes com sinal
interval [campos] [(p)]		intervalo de tempo
json		dados textuais JSON
jsonb		dados binários JSON, decomposto
line		linha infinita no plano
lseg		segmento de linha no plano
macaddr		endereço MAC (Media Access Control)
macaddr8		endereço MAC (Media Access Control) (formato EUI-64)
money		quantia monetária
numeric [(p, s)]	decimal [(p, s)]	numérico exato com precisão selecionável
path		caminho geométrico no plano

Tipos de dados – Tabela 2/4

Nome	Aliases	Descrição
pg_lsn		Número de sequência do log (Log Sequence Number) do PostgreSQL
pg_snapshot		instantâneo (snapshot) do ID da transação no nível do usuário
point		ponto geométrico no plano
polygon		caminho geométrico fechado no plano
real	float4	número de ponto flutuante de precisão simples (4 bytes)
smallint	int2	inteiro de dois bytes com sinal
smallserial	serial2	inteiro de dois bytes com autoincremento
serial	serial4	inteiro de quatro bytes com autoincremento
text		cadeia de caracteres de comprimento variável
time [(p)] [sem zona horária]		hora do dia (sem zona horária)
time [(p)] com zona horária	timetz	hora do dia, incluindo a zona horária

Tipos de dados – Tabela 4/4

Nome	Aliases	Descrição
timestamp [(p)] [sem zona horária]		data e hora (sem zona horária)
timestamp [(p)] com zona horária	timestampz	data e hora, incluindo a zona horária
tsquery		consulta de procura de texto completo
tsvector		documento de procura de texto completo
txid_snapshot		instantâneo (snapshot) do ID da transação no nível do usuário (obsoleto; veja pg_snapshot)
uuid		identificador universalmente único
xml		dados XML

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipos de dados numéricos

Os tipos de dados numéricos consistem em inteiros de dois, quatro e oito bytes, números de ponto flutuante de quatro e oito bytes, e decimais de precisão selecionável.

A a seguir lista os tipos de dados disponíveis.

Tipos de dados numéricos

Nome	Tamanho de armazenamento	Descrição	Intervalo dos valores
smallint	2 bytes	inteiro de pequeno intervalo	-32768 a +32767
integer	4 bytes	escolha usual para inteiro	-2147483648 a +2147483647
bigint	8 bytes	inteiro de grande intervalo	-9223372036854775808 a +9223372036854775807
decimal	variável	precisão especificada pelo usuário, exato	até 131072 dígitos antes do ponto decimal; até 16383 dígitos após o ponto decimal
numeric	variável	precisão especificada pelo usuário, exato	até 131072 dígitos antes do ponto decimal; até 16383 dígitos após o ponto decimal
real	4 bytes	precisão variável, inexato	precisão de 6 dígitos decimais
double precision	8 bytes	precisão variável, inexato	precisão de 15 dígitos decimais
smallserial	2 bytes	inteiro pequeno com autoincremento	1 a 32767
serial	4 bytes	inteiro com autoincremento	1 a 2147483647
bigserial	8 bytes	inteiro grande com autoincremento	1 a 9223372036854775807

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipos de dados de data e hora

O PostgreSQL aceita o conjunto completo de tipos de dados de data e hora do padrão SQL, mostrados na Tabela a seguir.

As operações disponíveis nesses tipos de dados são descritas em Funções e operadores de data e hora.

As datas são contadas segundo o calendário gregoriano, mesmo em anos anteriores à introdução desse calendário

Tipos de dados de data e hora

Nome	Tamanho de armazenamento	Descrição	Menor valor	Maior valor	Resolução
timestamp [(p)] [without time zone]	8 bytes	tanto data quanto hora (sem zona horária)	4713 A.C.	294276 D.C.	1 microssegundo
timestamp [(p)] with time zone	8 bytes	tanto data quanto hora, com zona horária	4713 A.C.	294276 D.C.	1 microssegundo
date	4 bytes	data (sem hora do dia)	4713 A.C.	5874897 D.C.	1 dia
time [(p)] [without time zone]	8 bytes	hora do dia (sem data)	00:00:00	24:00:00	1 microssegundo
time [(p)] with time zone	12 bytes	hora do dia (sem data), com zona horária	00:00:00+1559	24:00:00-1559	1 microssegundo
interval [campos] [(p)]	16 bytes	intervalo de tempo	-178000000 anos	178000000 anos	1 microssegundo

Interval

O tipo de dados interval possui uma opção adicional, que é a de restringir o conjunto de campos armazenados escrevendo uma destas frases:

- YEAR
- MONTH
- DAY
- HOUR
- MINUTE
- SECOND
- YEAR TO MONTH
- DAY TO HOUR
- DAY TO MINUTE
- DAY TO SECOND
- HOUR TO MINUTE
- HOUR TO SECOND
- MINUTE TO SECOND

Tipos de dados de data e hora

Ver
documentação
original para

- Entrada de data e hora
- Saída de data e hora
- Zonas horárias
- Entrada de intervalo
- Saída de intervalo

Tipo de dados monetário

O tipo de dados `money` armazena uma quantia monetária com precisão fracionária fixa; veja a tabela a seguir.

A precisão fracionária é determinada pela configuração do parâmetro `lc_monetary` no servidor de banco de dados.

O intervalo mostrado na tabela assume serem dois dígitos fracionários.

A entrada é aceita em vários formatos, incluindo literais inteiros e de ponto flutuante, assim como a formatação monetária típica, como '\$1.000,00'.

A saída geralmente está nessa última forma, mas depende da localidade.

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipos de dados monetário

Nome	Tamanho de armazenamento	Descrição	Intervalo dos valores
money	8 bytes	quantia monetária	-92233720368547758.08 a +92233720368547758.07

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipos de dados geométricos

Nome	Tamanho de armazenamento	Descrição	Representação
point	16 bytes	Ponto no plano	(x,y)
line	32 bytes	Linha infinita	$\{A,B,C\}$
lseg	32 bytes	Segmento de linha finito	$((x_1,y_1),(x_2,y_2))$
box	32 bytes	Caixa retangular	$((x_1,y_1),(x_2,y_2))$
path	$16+16n$ bytes	Caminho fechado (semelhante ao polígono)	$((x_1,y_1),\dots)$
path	$16+16n$ bytes	Caminho aberto	$[(x_1,y_1),\dots]$
polygon	$40+16n$ bytes	Polígono (semelhante ao caminho fechado)	$((x_1,y_1),\dots)$
circle	24 bytes	Círculo	$\langle(x,y),r\rangle$ (ponto central e raio)

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipo booleano

Nome	Tamanho de armazenamento	Descrição
boolean	1 byte	estado de verdade ou falso

O tipo de dados boolean pode ter vários estados: “true”, “false”, e um terceiro estado, “desconhecido”, representado pelo valor NULL no padrão SQL.

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipos de dados de endereço de rede

Nome	Tamanho de armazenamento	Descrição
cidr	7 ou 19 bytes	redes IPv4 e IPv6
inet	7 ou 19 bytes	redes e hospedeiros IPv4 e IPv6
macaddr	6 bytes	endereços MAC
macaddr8	8 bytes	endereços MAC (formato EUI-64)

O PostgreSQL oferece tipos de dados para armazenar endereços IPv4, IPv6 e MAC, conforme mostrado na acima.

É melhor usar esses tipos de dados em vez de usar tipos de dados de texto simples para armazenar endereços de rede, porque esses tipos de dados oferecem verificação de erros de entrada, além de operadores e funções especializados.

Tipos de dados

Tipos de dados numéricos

Tipo de dados monetário

Tipos de dados de cadeia de caracteres

Tipo de dados binário

Tipos de dados de data e hora

Tipo de dados booleano

Tipos de dados enum

Tipos de dados geométricos

Tipos de dados de endereço de rede

Tipos de dados de cadeia de bits

Tipos de dados de procura de texto completo

Tipo de dados UUID

Tipo de dados XML

Tipos de dados JSON

Tipos de dados matriz

Tipos de dados compostos

Tipos de dados de intervalo

Tipos de dados de domínio

Tipos de dados de identificador de objeto

Tipo de dados pg_lsn

Pseudotipos

Tipo booleano

Nome	Descrição
character varying(n), varchar(n)	comprimento variável, com limite
character(n), char(n)	comprimento fixo, completado com espaços
text	comprimento variável, sem limite

Dados binários

BLOBs, CLOBs. BYTEAs, OIDs

Tipos de binários

Nome	Tamanho de armazenamento	Descrição
bytea	1 ou 4 bytes mais a string binária real	variable-length binary string

Dados binários

Uma string binária é uma sequência de octetos (ou bytes).

As cadeias binárias distinguem-se das cadeias de caracteres de duas maneiras.

- Primeiro, as strings binárias permitem especificamente armazenar octetos de valor zero e outros octetos “não imprimíveis” (geralmente, octetos fora do intervalo decimal 32 a 126).

Dados binários

As cadeias de caracteres não permitem zero octetos e também proíbem quaisquer outros valores de octeto e sequências de valores de octeto que sejam inválidos de acordo com a codificação do conjunto de caracteres selecionado no banco de dados.

Dados binários

Segundo,

- as operações em cadeias binárias processam os bytes reais, enquanto o processamento de cadeias de caracteres depende das configurações de localidade.

Resumindo, as strings binárias são apropriadas para armazenar dados que o programador considera “bytes brutos”, enquanto as strings de caracteres são apropriadas para armazenar texto.

BLOB & CLOB

Armazenar grandes objetos binários (*Binary Large Objects* - BLOBs), tais como vídeos, imagens e arquivos PDF, e grandes objetos textuais (*Character Large Objects* - CLOBs) em um banco de dados é uma atividade que pode apresentar dificuldade maior que a vivenciada em se lidar com tipos de dados tradicionais.

Cada SGBD apresenta sua padronização e o PostgreSQL apresenta seus métodos de armazenamento, exclusão e recuperação de BLOBs e CLOBs. A portabilidade entre SGBDs não costuma ser fácil neste caso.

Dados binários

O tipo `bytea` suporta dois formatos de entrada e saída: formato “hex” e formato histórico “escape” do PostgreSQL.

Ambos são sempre aceitos na entrada.

O formato de saída depende do parâmetro de configuração `bytea_output`; o padrão é hexadecimal.

Dados binários

O padrão SQL define um tipo de string binária diferente, chamado BLOB ou BINARY LARGE OBJECT.

O formato de entrada é diferente do bytea, mas as funções e operadores fornecidos são basicamente os mesmos.

Operações com BLOBs eCLOBs

Quatro funções permitem operações sobre large objects, viabilizando a sua criação, recuperação, consulta e exclusão:

**\lo_import ARQUIVO
[COMENTÁRIO]**

- Importa um arquivo do filesystem (levando em conta a search_path) para o postgresql, atribuindo ao mesmo um identificador de objeto (OID) e um comentário, quando fornecido este parâmetro

\lo_list -

- Apresenta uma listagem dos objetos binários importados

**\lo_export OIDLOB
ARQUIVO**

- Exporta um BLOB ou CLOB do postgres para o filesystem sem removê-lo do postgresql, referenciando-o pelo seu OID

\lo_unlink OIDLOB

- Elimina um large object do postgresql com base no seu OID.

Armazenamento de dados binários

O PostgreSQL provê duas formas distintas para armazenar dados binários.

Os dados binários podem ser armazenados em uma tabela utilizando o tipo de dado *bytea*, ou através da funcionalidade de objeto grande que armazena os dados binários em uma tabela a parte em um formato especial, e faz referência a esta tabela armazenando um valor do tipo *oid* na tabela do usuário.

Armazenamento de dados binários

Para ser possível determinar qual é o método apropriado, é necessário compreender as limitações de cada método.

O tipo de dado *bytea* não é muito adequado para armazenar quantidades muito grande de dados binários.

Embora uma coluna do tipo *bytea* possa conter até 1 GB de dados binários, é necessária uma grande quantidade de memória para processar um valor tão grande.

Armazenamento de dados binários

O método objeto grande para armazenar dados binários é mais adequado para armazenar valores muito grandes, mas possui suas próprias limitações.

Especificamente, a exclusão da linha que contém a referência ao objeto grande não exclui o objeto grande.

A exclusão do objeto grande é uma operação que precisa ser realizada em separado.

Os objetos grandes também possuem alguns problemas de segurança, uma vez que qualquer um que se conecte ao banco de dados pode ver e/ou modificar qualquer objeto grande, mesmo que não possua permissão para ver/atualizar a linha que faz referência ao objeto grande.

Tipos de dados de identificador de objeto

Os identificadores de objeto (OIDs) são usados internamente pelo PostgreSQL como chaves primárias para várias tabelas do sistema.

O tipo de dados oid representa um identificador de objeto.

Existem também vários tipos de dados alias para oid, cada um chamado `regalguma_coisa`.

A a seguir dá uma visão geral.

Tipos de binários

Nome	Referência	Descrição	Exemplo de valor
oid	todos	identificador numérico de objeto	564182
regclass	pg_class	nome da relação	pg_type
regcollation	pg_collation	nome da ordenação	"POSIX"
regconfig	pg_ts_config	configuração da procura de texto	english
regdictionary	pg_ts_dict	dicionário de procura de texto	simple
regnamespace	pg_namespace	nome do espaço de nomes	pg_catalog
regoper	pg_operator	nome do operador	+
regoperator	pg_operator	operador com os tipos de dados dos argumentos	*(integer,integer) ou -(NONE,integer)
regproc	pg_proc	nome da função	sum
regprocedure	pg_proc	função com os tipos de dados dos argumentos	sum(int4)
regrole	pg_authid	nome do papel (role)	smithee
regtype	pg_type	nome do tipo de dados	integer

Tipos de dados de identificador de objeto

No momento, o tipo de dados oid é implementado como inteiro sem sinal de quatro bytes.

Portanto, não é grande o suficiente para fornecer unicidade em todo o banco de dados em grandes bancos de dados, ou mesmo em grandes tabelas individualmente.

O próprio tipo de dados oid tem poucas operações além da comparação.

No entanto, pode ser convertido em inteiro e depois tratado usando os operadores padrão para inteiros. (Cuidado com a possível confusão entre “com sinal” versus “sem sinal”, se isso for feito.)

Tipos de dados de identificador de objeto

Os tipos de dados alias de OID não têm operações próprias, exceto rotinas de entrada e saída especializadas.

Essas rotinas conseguem aceitar e exibir nomes simbólicos para objetos do sistema, em vez do valor numérico bruto que o tipo de dados oid usaria.

Os tipos de dados alias permitem a procura simplificada de valores de OID para objetos.

