



Banco de dados com PostgreSQL

Aula 1



Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.





Um banco de dados relacional é um tipo de banco de dados que armazena e fornece acesso a pontos de dados relacionados entre si. Bancos de dados relacionais são baseados no modelo relacional, uma maneira intuitiva e direta de representar dados em tabelas. Em um banco de dados relacional, cada linha na tabela é um registro com uma ID exclusiva chamada chave. As colunas da tabela contêm atributos dos dados e cada registro geralmente tem um valor para cada atributo, facilitando o estabelecimento das relações entre os pontos de dados.





SQL é a sigla para *Structured Query Language*, ou Linguagem de Consulta Estruturada, em português. É uma linguagem de programação utilizada para trabalhar com banco de dados relacionais.

Com o SQL, é possível realizar consultas à base de dados informando diversos tipos de parâmetros (seguindo algumas regras e normas da linguagem), podendo obter grandes quantidades de informação de diferentes fontes (no caso, tabelas) com poucos ou até mesmo apenas uma instrução.





Um banco de dados não relacional é um banco de dados que não usa o esquema de tabela de linhas e colunas encontrado na maioria dos sistemas de banco de dados tradicionais. Em vez disso, os bancos de dados não relacionais usam um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados que está sendo armazenado. Por exemplo, os dados podem ser armazenados como pares chave/valor simples, como documentos JSON ou como um gráfico que consiste em bordas e vértices.

O que esses armazenamentos de dados têm em comum é que eles não usam um modelo relacional. Além disso, eles tendem a ser mais específicos no tipo de dados ao qual dão suporte e no modo como os dados podem ser consultados.





O termo NoSQL refere-se aos armazenamentos de dados que não usam o SQL para consultas. Em vez disso, os armazenamentos de dados usam outras linguagens de programação e constructos para consultar os dados. Na prática, "NoSQL" significa "banco de dados não relacionais", mesmo que muitos desses bancos de dados deem suporte a consultas compatíveis com SQL. No entanto, a estratégia de execução de consulta subjacente é geralmente muito diferente da maneira como um RDBMS tradicional executa a mesma consulta SQL.

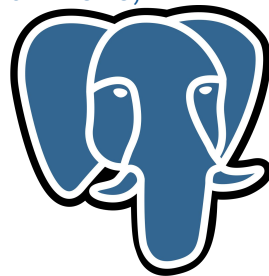




PostgreSQL é um sistema gerenciador de banco de dados objeto relacional (SGBD), desenvolvido como projeto de código aberto.

- Gratuito e OpenSource
- Multiplataforma: Windows, Linux e Mac

O PostgreSQL é um sistema que lida bem com altos volumes de solicitações e com cargas de trabalho grandes, ou seja, funciona muito bem para sites com intensidade de acesso. E-commerces famosos, por exemplo, é um ótimo exemplo de estrutura que precisa desse sistema para ter um desempenho otimizado, devido ao alto número de acessos simultâneos recebidos.

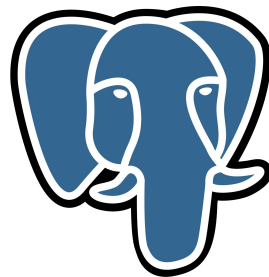




Há vários tipos de dados incorporados disponíveis no PostgreSQL. Além disso, também é possível criar (definir) tipos de dados personalizados caso seja necessário, e as próprias tabelas são consideradas como sendo tipos em si, podendo inclusive serem usadas dentro de uma coluna em uma tabela diferente.

Separamos os tipos de dados suportados pelo PostgreSQL dentro das seguintes categorias:

- Tipos numéricos
- Tipos de String
- Tipos de Data e Hora
- Outros tipos



Tipos numéricos

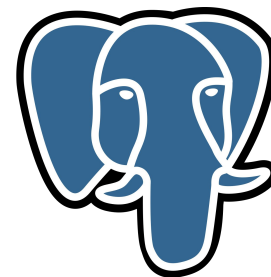


smallint	Tipo inteiro com 2 bytes, na faixa de -32768 até +32767
int	Tipo inteiros com 4 bytes, na faixa de -2147483648 até +2147483647
bigint / int8	Inteiro com 8 bytes com sinal, na faixa de -9223372036854775808 até 9223372036854775807
smallserial / serial2	Tipo inteiro sequencial, usados em campos de auto-incremento. Intervalo de 1 até 32767.
serial / serial4	Serial com 4 bytes. Intervalo de 1 até 2147483647.
bigserial / serial8	Serial com 8 bytes. Intervalo de 1 até 9223372036854775807.
numeric(p,s)	Números decimais, com escala s e precisão p. Até 131072 dígitos antes do ponto decimal e 16383 dígitos após o ponto decimal.
double precision	Números de ponto flutuante. precisão de 15 dígitos decimais.

Tipos de String



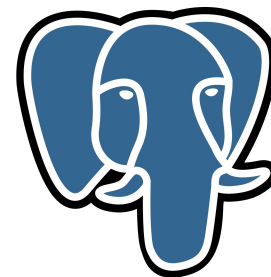
<code>varchar(n)</code> ou <code>character varying(n)</code>	String com no máximo n caracteres, sem espaços no final
<code>char(n)</code>	String com exatamente n caracteres
<code>text</code>	Texto de tamanho ilimitado



Tipos de Data e Hora



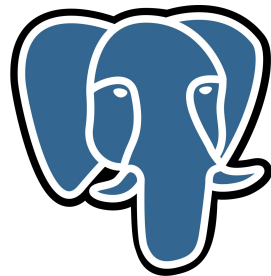
Date	Representa datas sem a hora. A representação padrão é a ISO 8601, no formato AAAA-MM-DD. Intervalo de datas possível: de 4713 A.C até 5874897 AD.
Timestamp	Data e hora, independente de fuso horário (com ou sem). Intervalo de datas possível: de 4713 A.C até 294276 AD.
Time	Somente a hora, sem a data, no formato hh:mm:ss.
Interval	Unidade de tempo que é utilizada para adicionar ou subtrair tempo de um timestamp.



Outros tipos de dados



money	moneyRepresentação de moedas, no intervalo de de -92233720368547758,08 até +92233720368547758,07
boolean	Valores true ou false (verdadeiro ou falso)

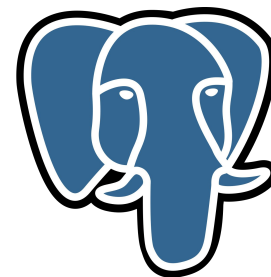




Os comandos DDL (Data Definition Language) são os comandos SQL utilizados para definir e alterar a estrutura do banco de dados, tabelas, views, etc.

Os comandos DDL são:

- CREATE
- ALTER
- TRUNCATE
- DROP

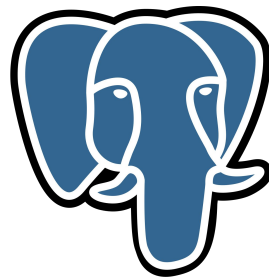




CREATE

O comando CREATE é utilizado para criação de objetos como database, table, index, views, store procedure, function e triggers.

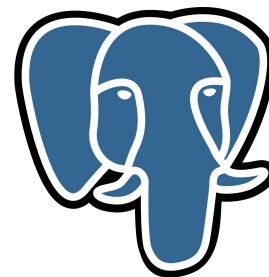
```
CREATE TABLE <table_name>
(column_name_1 datatype,
column_name_2 datatype,
.
.
column_name_n datatype
);
```





Um exemplo de criação de tabela

```
CREATE TABLE tb_alunos (  
    id int,  
    nome varchar(50),  
    endereco varchar(100),  
    nascimento date,  
    primary key(id)  
)
```

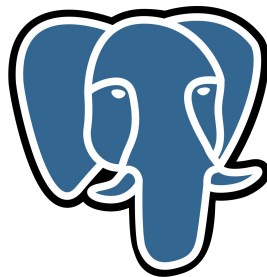





ALTER

O comando ALTER permite a alteração da estrutura de um objeto dentro do banco de dados. Dentro de uma tabela, por exemplo, podemos realizar as seguintes operações:

- Adicionar uma coluna em uma tabela
- Excluir uma coluna existente
- Alterar o tipo de dado de uma coluna



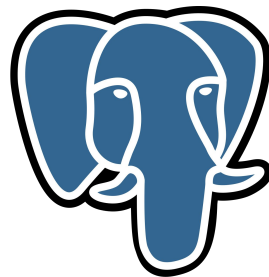


```
ALTER TABLE tb_cobranca
    ADD dataPagamento DATE

ALTER TABLE tb_cobranca
    ALTER COLUMN valorPago TYPE MONEY

ALTER TABLE tb_cobranca
    ALTER COLUMN valor SET NOT NULL

ALTER TABLE tb_cobranca
    DROP COLUMN chavePix
```

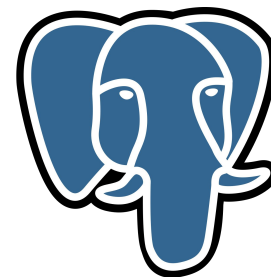




TRUNCATE

Utilizada para remover todas as linhas de uma tabela, além do espaço alocado para elas.

```
TRUNCATE TABLE tb_cachorro
```



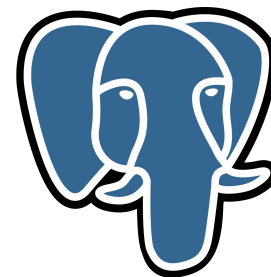


DROP

Utilizamos o comando DROP para remover objetos, como tabelas e bancos de dados.

```
DROP TABLE tb_cachorro
```

```
DROP DATABASE petshop
```



Exercício 1



Vamos criar um banco de dados de controle de um petshop.

Nesse exercício, crie as seguintes estruturas:

- Um schema, chamado petshop
- A tabela tb_animal, com as colunas id, nome, dono e raça
- A coluna nome é um varchar, e indica o nome do animal
- A coluna dono é o nome do dono, e também é um varchar
- A coluna raça indica a raça do animal, e também é um varchar





O próximo passo será criar uma tabela para controlar a raça dos animais.

- A tabela se chamará `tb_raca`, com uma coluna de `id`
- Deve ter uma coluna com o nome da raça
- Uma coluna indicando o animal a qual essa raça pertence

Altere também a tabela `tb_animal`, excluindo o campo `raça`, e criando um novo, do tipo `int`



O próximo passo será criar uma tabela para controlar os donos cadastrados no sistema

- A tabela se chamara tb_pessoa, com uma coluna de id
- Criaremos uma coluna com o nome do dono e seu contato (telefone e email)

Altere também a tabela tb_animal, excluindo o campo dono, e criando um novo, do tipo int

