

Tecnologias de Banco de Dados

Msc. Eliezio Soares
eliezio.soares@ifrn.edu.br

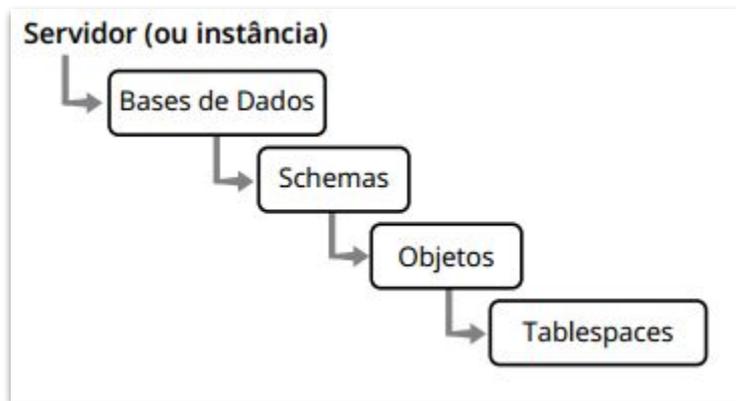


INSTITUTO FEDERAL
Rio Grande do Norte

Campus
Carrais Novos

Organização Geral

- Uma instância PostgreSQL pode conter diversos bancos de dados, que por sua vez contém Schemas que armazenam objetos (tabelas, índices, funções etc).



Fonte: Caiut, Fábio. 2015.

Bases de Dados

- Uma base de dados é uma coleção de objetos, tais como:
 - Tabelas
 - Visões
 - Funções
- Um servidor pode ter várias bases de dados;
- Cada conexão é feita em uma base de dados;
- Não se pode acessar objetos de outra base de dados senão a que se está conectado;
- Bases de dados são fisicamente separadas;
- Toda base possui um proprietário (*owner*);

Bases de Dados

Quando uma instância é iniciada (comando initdb), são criadas três bases de dados:

1. **Postgres:** Base criada para a conexão padrão. Não é importante para o funcionamento do servidor.
2. **Template0:** Base utilizada como modelo para novas bases pelo próprio Postgres. Template0 não pode ser alterada.
3. **Template1:** Base utilizada como modelo para outras a serem criadas. Essa base pode ser alterada, logo o que for acrescentado será padrão para as bases copiadas de Template1.

O comando CREATE DATABASE, funciona copiando uma base de dados.

Base de Dados - Criação

- Ao executar o comando CREATE DATABASE, é possível utilizar os parâmetros:
 - **OWNER:** Usuário dono da base de dados, tendo privilégios para criar schemas, objetos e permissões;
 - **TEMPLATE:** Especifica a base modelo a partir da qual a nova base será copiada;
 - **ENCODING:** Define o conjunto de caracteres a ser utilizado.
 - **TABLESPACE:** Define o local onde serão criados os objetos na base.

```
postgres=# CREATE DATABASE rh OWNER postgres ENCODING 'UTF-8' TABLESPACE = tbs_disco2;
```

Bases de Dados

- Exemplo cópia do Template1:

- Conectar ao banco

```
postgres=# \c template1
```

```
template1=# CREATE TABLE templateexemplo(id serial, data timestamp);
```

```
template1=# \d
```

```
template1=# CREATE DATABASE bdExemploTemplate1 TEMPLATE template1;
```

```
template1=# \c bdExemploTemplate1
```

```
bdExemploTemplate1=# \d
```

Bases de Dados

- Exemplo cópia do Template0:

- Conectar ao banco

```
template1=# CREATE DATABASE bdExemploTemplate0 TEMPLATE template0;
```

```
template1=# \c bdExemploTemplate0
```

```
bdExemploTemplate1=# \d
```

```
bdExemploTemplate1=# \l
```

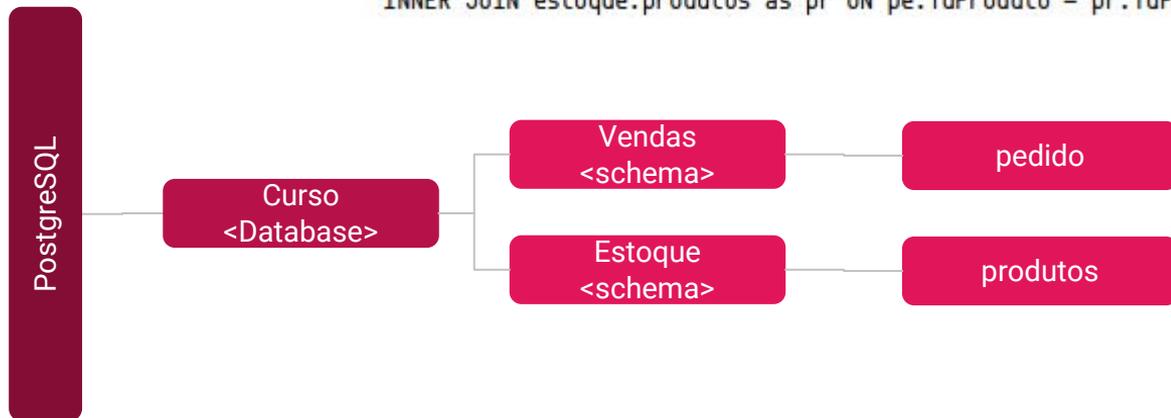
```
bdExemploTemplate1=# \+
```

E se não utilizamos o parâmetro TEMPLATE???

Schemas

- As bases de dados podem ser organizadas em schemas:
 - Schemas são, portanto, uma partição lógica para os objetos da base de dados.
 - É possível acessar objetos de mais de um schema em uma mesma consulta.

```
curso=> SELECT *  
        FROM vendas.pedido as pe  
        INNER JOIN estoque.produtos as pr ON pe.idProduto = pr.idProduto;
```



Schemas

- É possível visualizar os schemas é possível consultar a tabela `pg_namespace` no catálogo, ou utilizar o comando `\dn+` no psql.
- A criação de um schema se dá através do comando:

```
dbexemplo=# CREATE SCHEMA nomeesquema;
```

- Somente usuários com permissão CREATE e superusuários podem criar schemas.
- Para se atribuir a propriedade de um schema a outro usuário, utiliza-se a propriedade AUTHORIZATION:

```
dbexemplo=# CREATE SCHEMA nomeesquema AUTHORIZATION  
nomeusuario;
```

Schemas

- Visualisar os schemas:

```
dbexemplo=# \dn+
```

- Para visualizar os objetos de um schema específico:

```
dbexemplo=# \dt nomeschema.*
```

Schemas - DROP

- Remover um schema:

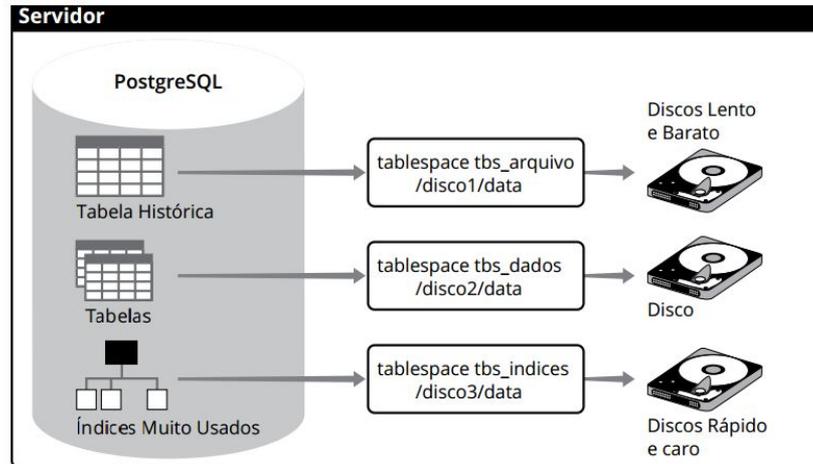
```
dbexemplo=# DROP SCHEMA nomeschema;
```

- Não é possível remover um schema não vazio.
- Uma alternativa é utilizar a cláusula CASCADE:

```
dbexemplo=# DROP SCHEMA nomeschema CASCADE;
```

Tablespaces

- Os Tablespaces são locais no sistema de arquivos para armazenar dado.
 - São diretórios no sistema de arquivos local;
 - É possível utilizar outros discos para expansão da capacidade ou melhoria de desempenho.



Tablespaces

- Para listar os tablespaces existentes no servidor, pode se consultar a tabela do catálogo `pg_tablespace` ou:

```
dbxemplo=# \db
```

- Por padrão são definidos dois tablespaces:
 - **Pg_default:** Aponta para PGDATA/base
 - **Pg_global:** Aponta para o diretório PGDATA/global, que contém objetos compartilhados entre todas as bases, como `pg_tablespace`, por exemplo.

Tablespaces - Criação e uso

- Criar um novo tablespace:

```
dbexemplo=# CREATE TABLESPACE tbs_arquivo LOCATION  
'/media/db/data2';
```

- O diretório precisa já existir e estar vazio.
- A propriedade do diretório deve ser do usuário postgres.

- Inserir objetos em um tablespace:

```
dbexemplo=# CREATE TABLE registro (login varchar(20), datahora timestamp)  
TABLESPACE tbs_arquivo;
```

- Criar um banco de dados em um tablespace específico:

```
dbexemplo=# CREATE DATABASE vendas TABLESPACE tbs_dados;
```

Tablespaces - Criação e uso

- Alterar o tablespace de uma tabela ou índice:

```
dbexemplo=# ALTER DATABASE dbexemplo SET TABLESPACE tbs_dados;
```

- Todos os objetos terão o tablespace movidos para o novo local.

- Exclusão de tablespaces:

```
dbexemplo=# DROP TABLESPACE tbs_arquivo;
```

Leitura complementar

- Ler sessão “**Catálogo de Sistema do PostgreSQL**” - páginas 49 - 53.
 - CAIUT, Fábio. Administração de banco de dados. 1ª Edição. Rio de Janeiro. RNP/ESR, 2015.

Bibliografia Utilizada

CAIUT, Fábio. Administração de banco de dados. 1ª Edição. Rio de Janeiro. RNP/ESR, 2015.

PostgreSQL 10.5

Documentation. Disponível em:
<<https://www.postgresql.org/docs/10/static/index.html>>. Acesso em 09 Ago. 2018.

