




Gerenciamento de Banco de Dados

Prof. Esp. Tiago segato



Envolvidos no Gerenciamento do DB

- Quem são os envolvidos no Gerenciamento de Banco de Dados?
- Há 4 papéis envolvidos
 - Administradores de Dados
 - Data Base Administradores
 - Desenvolvedores Programadores
 - Usuário Final
- Tarefas distintas ou comuns
- Qual a vantagem de cada um desempenhar o seu papel?

Administrador de Dados

- Responsável por extrair do processo ou negócio em questão quais os campos (informações) serão armazenados no BD.
- Precisa conhecer Banco de Dados e o processo / negócio em questão.
 - Não descartar informações importantes
 - Não utilizar Informações irrelevantes
- Não necessita saber como implementar as decisões no Banco de Dados (DBA).
- Define o tipo de Acesso a Dados que cada envolvido no Projeto terá.

Data Base Administrator

- Implementa as decisões e solicitações providas do AD.
- Profissionais de TI (profundo conhecimento no Banco de Dados em questão).
- Quem conhece as restrições tecnológicas
- Acompanha o ciclo de vida do BD e é encarregado em observar o comportamento do BD, bem como identificar problemas (performance, eficiência etc).

Programadores/Desenvolvedores

- Criam a Interface (Sistemas) por meio de Linguagens de Programação.
- Não só visual, mas também tem a responsabilidade em consistir os dados antes de serem enviados aos BD.
- Garantir que cada acesso ao BD seja fornecido somente a seu solicitante e existir as informações necessárias de cada vez.
- Trabalham em conjunto com os Analistas.
 - Identificando as melhores maneiras de interagir com os dados,
 - Fazendo com que o sistema em questão seja o mais simples, intuitivo e confiável o possível.

Usuário Final

- Causa a necessidade dos outros papéis e inclusive dos BDs.
- Além de utilizar o sistema é um dos principais envolvidos no projeto do Banco de Dados.
 - Por que?
- Quem fornece as Informações necessárias do negócio.
- Deve confiar no profissional de TI (não esconder detalhes das Informações)

Gerenciando um Banco de Dados

- Dependendo da complexidade do banco ou das ferramenta utilizadas, pode ser fácil ou extremamente complexo.
- Quais são essas ferramentas?
- PostgreSQL:
 - PgAdmin III
 - phpPgAdmin
 - PSQL
- Gerenciar:
 - Tablespace, Banco de Dados, Esquemas, Tabelas, Índices, Visões e Domínios

PSQL

- PSQL é um programa-cliente que acompanha o PostgreSQL em sua instalação padrão.
- Servem para fazer a ponte entre o DBA e o PostgreSQL.
- Supre as necessidades que surgirem através da ferramenta gráfica.
- Permite de maneira simples e rápida realizar os primeiros acessos ao servidor.
- Opera em modo texto (desvantagem ligado à praticidade)
- Modo mais abrangente, pois permite que todo e qualquer comando SQL seja completamente manipulado e alterável pelo usuário.

Pesquisa...

- ▀ Quais são as ferramentas de serviço (programas-cliente) utilizadas pelos respectivos SGBDs citados abaixo?
 - ▀ Oracle, IBM DB2, SQL Server, MySQL e FireBird
- ▀ Descreva suas principais características, vantagens e desvantagens comparando com as ferramentas do PostgreSQL.

Gerenciando...

- Tablespaces;
- Databases;
- Eschemas;
- Tables;
- Índex;
- Domains.

Comandos PSQL

- Os comandos devem ser executados através do Shell do Postgres
- **Iniciar / Programas / PostgreSQL 9.1 / SQL Shell (psql)**

Gerenciamento de TABLESPACE

- São definições de locais para armazenamento lógico das informações do servidor.
- Trata-se de diretórios existentes em um S.O.
- Torna possível armazenar informações do servidor em locais distintos
 - Políticas de backups, utilização de mais de um disco, organização, etc.
- É utilizada em conjunto com BD e Tabelas.

Criando uma TABLESPACE

- Sintaxe:
 - `CREATE TABLESPACE nome LOCATION 'localização' ;`
- Onde localização é o endereço lógico (diretório) de armazenamento.
- OBS: É necessário que o diretório esteja vazio e o usuário tenha permissões de gravação no mesmo (S.O.).

Visualizando uma TABLESPACE

- Sintaxe:
 - `\db`
- OBS: Todos os tablespaces serão visualizados e os que iniciam por pg_ são internos do PostgreSQL porém não devem ser excluídos.

Alterando uma TABLESPACE

- Somente podem ser alterados o **nome da tablespace** e o **nome do seu proprietário**.
- Por questões de segurança o Diretório não pode ser alterado.
- Sintaxe:
 - ALTER TABLESPACE **nome** RENAME TO **novو_nome**;
 - ALTER TABLESPACE **nome** OWNER TO **novو_proprietário**;

Excluindo uma TABLESPACE

- Sintaxe:
 - DROP TABLESPACE **nome**;
- OBS:
 - Não é possível excluir a Tablespace caso não esteja vazia, nem ao mesmo se utilizar o CASCADE.
 - Para isto é necessário esvaziar todo o diretório e em seguida executar o comando referente a exclusão.

Gerenciamento de BD

- Objeto mais básico em um SGBD, pois qualquer outro objeto que venha a ser criado após este (tabelas, visões, procedures, etc) estarão de alguma forma interligados.
- Deve ser criado de maneira otimizada:
 - Economizar recursos de Hardware;
 - Rapidez nas consultas;
 - Uso eficiente;
 - Segurança em seus dados e acessos;
 - Não falhar.

Criando um BD

- Sintaxe:
 - CREATE DATABASE nome [argumentos];
- Além do nome (obrigatório), permite alguns argumentos opcionais:
 - OWNER *usuário*: Quem será seu responsável;
 - TEMPLATE *nome*: Para seguir modelos de bds;
 - ENCODING *valor*: Indica o conjunto de caracteres a ser utilizado;
 - TABLESPACE *nome*: qual tablespace vai pertencer;
 - CONNECTION LIMIT *valor*: limita conexões simultâneas (-1 = ilimitado)
- CREATE DATABASE **escola** ENCODING 'UTF8'
TABLESPACE **pg_default** CONNECTION LIMIT **4**;

Visualizando um DB

- Sintaxe:

- \I

- OBS: Esta sintaxe retorna os bancos criados no SGBD acompanhado de mais algumas informações como vistas anteriormente.

Ativando um BD

- Sintaxe:

- \c nome_banco;

- É necessário “mostrar” qual Banco de Dados se deseja manipular.
- Sem essa Ativação/Conexão se for utilizado algum comando como o SELECT não se saberá em qual banco existente o mesmo será aplicado.

Alterando um BD

■ Sintaxes:

- ALTER DATABASE **nome** RENAMR TO **novو_nome**; (alterando o nome do banco)
- ALTER DATABASE **nome** OWNER TO **novو_proprietário**; (alterando o nome do responsável do banco)
- ALTER DATABASE **nome** CONNECTION LIMIT **novو_valor**; (alterando valor máximo de conexões simultâneas)

Excluindo um BD

■ Sintaxe:

- DROP DATABASE **nome**;

■ OBS:

- Para efetuar a exclusão é necessário ser o responsável pelo mesmo.
- Uma vez que o banco de dados é excluído, toda a sua estrutura bem como os dados contidos nela serão excluídos.

Gerenciamento de Esquemas

- São fichários de organização dos objetos criados em um BD.
 - Objetos: Tabelas, Visões, índices etc.
- Portanto utilizando um Esquema é possível criar uma organização onde é vinculado cada objeto a um Esquema diferente.
- Ex: DB EMPRESA – é possível criar os esquemas de cada setor da mesma separadamente, como RH, TI, Financeiro, etc.
 - Dentro de cada esquema as tabelas referente a cada área.
- Não oferece vantagem quanto a Otimização e sim a Organização e manipulação futura.
 - Importante para o Controle de Acesso.

Criando um Esquema

- Sintaxe:
 - `CREATE SCHEMA nome [AUTHORIZATION proprietário];`
- OBS: O campo entre [] é opcional, caso não seja preenchido o proprietário será o usuário padrão do postgresQL.

Visualizando um Esquema

- Sintaxe:

- \dn

- Para visualizar as estruturas pertencentes ao Esquema a sintaxe é:

- \dtvs nome_esquema

Alterando um Esquema

- Sintaxe:

- ALTER SCHEMA nome RENAME TO novo_nome;

- Para alterar o proprietário do Esquema a sintaxe é:

- ALTER SCHEMA nome OWNER TO novo_proprietário;

Excluindo um Esquema

- Sintaxe:
 - DROP SCHEMA nome [CASCADE];
- OBS: A opção [CASCADE] indica que ao remover o Esquema todos os objetos pertencentes ao mesmo também serão removidos.
- Caso a opção não esteja marcada será emitida uma mensagem informando que o mesmo possui objetos e a operação será abortada.

Gerenciamento de Tabelas

- O gerenciamento de tabelas é um dos conjuntos de comandos mais completos e customizáveis.
- Tendo em vista uma grande quantidade de parâmetros que podem ser utilizados.
- Além reproduzir tudo que contém na ferramenta gráfica, permite mais flexibilidade nas opções de configurações das tabelas.

Criando uma Tabela

■ Sintaxe:

```
CREATE [ TEMPORARY | TEMP] TABLE
[nome_êsquema.]nome (
  Coluna tipo [restrição],
  Coluna tipo [restrição],
  ) [opções avançadas];
```

■ **Temporary e Temp:** não criam a tabela definitiva no BD, mas sim enquanto a sessão do usuário estiver ativa.

■ Sintaxe das Colunas:

```
■ nome_coluna tipo_dado [DEFAULT valor] [NULL |
  NOT NULL] [PRIMARY KEY | UNIQUE];
```

Visualizando uma Tabela

■ Sintaxe:

```
■ \d nome_tabela;
```

■ Utilizando o psql crie e visualize a seguinte tabela:

```
■ ALUNO = {ra, nome, idade, curso}
```

■ Onde:

```
■ Ra: chave-primária e not null;
```

```
■ Nome: not null
```


Alterando uma Tabela

Permite: Renomear a Tabela, suas Colunas, Inserir novos Campos, Remover Colunas, dentre outros.

- Sintaxe: Renomear a Tabela
 - `ALTER TABLE nome_tabela RENAME TO novo_nome;`
- Sintaxe: Renomear a Coluna
 - `ALTER TABLE nome_tabela RENAME nome_coluna TO novo_nome_coluna;`
- Sintaxe: Adicionar uma Coluna
 - `ALTER TABLE nome_tabela ADD sintaxe_inserção_coluna;`

Alterando uma Tabela

- Sintaxe: Removendo uma Coluna
 - `ALTER TABLE nome_tabela DROP nome_coluna [CASCADE];`
- Sintaxe: Alterando o tipo de dado da Coluna
 - `ALTER TABLE nome_tabela ALTER nome_coluna TYPE novo_tipo_dado expressão_SQL;`

Excluindo uma Tabela

- Sintaxe:
 - DROP TABLE nome_tabela [CASCADE];
- Pode se também excluir apenas os dados de uma Tabela, mantendo-se a sua estrutura:
 - TRUNCATE TABLE nome_tabela;

Bloqueando uma Tabela

- Em algumas situações, como em Transações, Alterações de grande volume de dados ou até mesmo um Backup, será necessário manter o Banco de Dados consistente.
- LOCK TABLE
- Bloqueia a tabela em questão para que outros usuários ou solicitação não altere suas informações.
- Sintaxe:
 - LOCK TABLE nome_tabela;
- Desbloqueio: feito automaticamente após a transação ter sido concluída.

Gerenciamento de Índices

- Recurso diretamente ligado a Otimização de um SGBD. Prepara as estruturas de organização de registros para tornar a sua busca e consulta mais rápida, precisa e eficiente.
- Ex: Gerar Relatórios dos Funcionários por ordem Alfabética e de Nascimento.
- O Índice é a seleção de uma ou mais colunas para já armazenar os seus dados ordenados no banco, de forma transparente para os usuários.
- Características:
 - Ocupa muito espaço em disco e deve ser utilizado em situações frequentes.

Criando um Índice

- Sintaxe:
 - `CREATE INDEX nome ON tabela [USING algoritmo] (coluna);`
- Onde: USING é opcional, ele fornece algoritmos do próprio PostgreSQL para realizar as buscas dentro do Índice criado.
- Os Algoritmos disponíveis são: BTREE, HASH, GIST e CIN.
- Ex:
`CREATE INDEX func_data_nasc ON funcionarios USING BTREE (data_nascimento);`

Exemplo

- Sendo assim, quando realizamos uma pesquisa pelo filtro *login='TIAGO'*, o banco saberá exatamente em que local do disco (hd) encontra-se o registro que atenda a esse critério.
- **CREATE INDEX** *idx_usuario_login_senha* **ON** *usuario* **USING** *btree* (*login*,*senha*);
- No caso acima, foi criado um índice *idx_usuario_login_senha* na tabela *usuario* para os campos *login* e *senha*. Então, a consulta será otimizada quando esses 2 campos forem utilizados em uma consulta.
- **SELECT * FROM** *usuario* **WHERE** *login* = 'TIAGO' **AND** *senha* = '123456';

Visualizando um Índice

- Sintaxe:
 - \di

Renomeando um Índice

- Sintaxe:
 - **ALTER INDEX** *nome* **RENAME TO** *novo_nome*;

Removendo um Índice

- Sintaxe:
 - DROP INDEX nome [tipo_remoção];
- Onde: [tipo_remoção]: pode ser CASCADE (exclui outros objetos dependentes deste índice) ou RESTRICT (aborta a operação, caso haja outros objetos dependentes deste índice).
- A estrutura do índice é removida, porém os dados originais da tabela continuarão intactos.

Gerenciamento de Domínios

- São comparados à máscaras de dados devido a sua funcionalidade:
 - Verificar se o valor de um campo (coluna) está de acordo com o padrão preestabelecido pelo usuário.
- Validação: CEP, Telefone, CPF etc.
 - Porém não valida se os dados existem ou realmente são de quem o digitou, mas sim se os mesmos estão de acordo com os padrões do devido campo (letras, números, caracteres especiais etc.).

Criando um Domínio

■ Sintaxe:

■ `CREATE DOMAIN nome_tipo_dado [DEFAULT valor]
[CONSTRAINT nome_regra CHECK (fórmula)];`

■ Onde:

- **nome:** nome do domínio, que deverá ser referenciado na criação das tabelas;
- **tipo_dado:** indica o tipo de dado sem validação;
- **valor:** caso tenha-se um valor padrão para quando o mesmo não for preenchido;
- **nome_regra:** poderá haver mais de uma regra por domínio;
- **fórmula:** onde será definido a fórmula de validação do valor do domínio.

Exemplo Criação de Domínio

- Criação de Domínio para o campo sexo. Onde só serão aceitos os Caracteres M ou F.

```
CREATE DOMAIN coluna_sexo char(1) DEFAULT 'M'
CONSTRAINT validaM_F CHECK ( VALUE IN ('M', 'F'));
```

- Utilizando o Domínio:

```
CREATE TABLE aluno (
  ra int NOT NULL,
  nome varchar(30),
  sexo coluna_sexo
);
```

Visualizando um Domínio

- Permite visualizar as relações que contenham um domínio e suas fórmulas de validação.
- Sintaxe:
 - \dD

Alterando um Domínio

- **Adicionando uma regra de validação:**
- Sintaxe:
 - ALTER DOMAIN nome ADD CONSTRAINT nome_regra CHECK (fórmula);
 - Deve-se informar o nome do Domínio que receberá a nova regra;
- **Excluindo uma regra de validação:**
- Sintaxe:
 - ALTER DOMAIN nome DROP CONSTRAINT nome_regra;
 - Exclui-se somente a regra e não o Domínio;

Excluindo um Domínio

- Sintaxe:
 - DROP DOMAIN nome [CASCADE];
- Obs: Caso a opção CASCADE for escolhida além do domínio, será apagado também o campo (coluna) e registros na qual o domínio foi utilizado.

► Ex:

► sexo **coluna_sexo**

RA	Nome	Sexo
123	João	M
456	Maria	F

Logs

- São Arquivos que registram todas as ações executadas em um SGBD.
- O PostgreSQL (e a maioria dos SGBDs) pode e, por padrão, realiza o gerenciamento de suas ações dentro de uma ou mais arquivos de log no sistema.
- Este tipo de procedimento é vital para possibilitar futuras análises do BD, ou servidores, na resolução de problemas ou outras situações que venham a ocorrer.
- A localização destes arquivos pode ser obtida no arquivo de configuração do PostgreSQL (postgresql.conf)