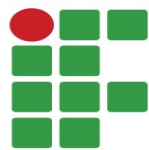


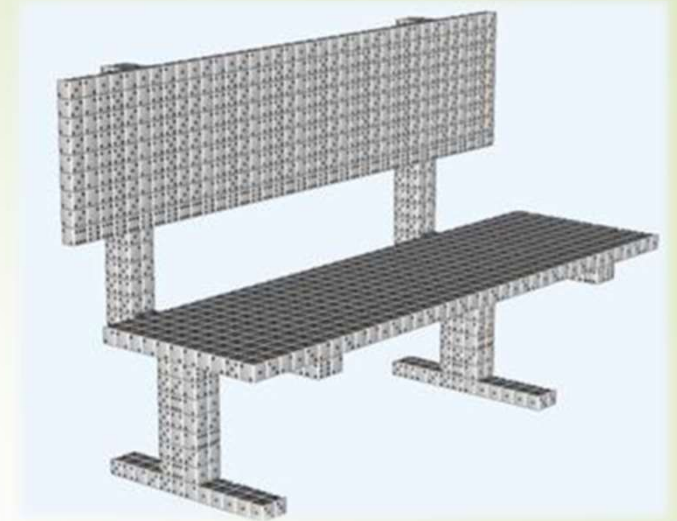
Banco de Dados II

- Introdução à Administração de BD

Prof. Angelo Augusto Frozza, Dr.
<http://about.me/TilFrozza>



INSTITUTO FEDERAL
Catarinense
Campus Camboriú



Roteiro



- Tarefas de Administração de BD
- Conhecimentos necessários para um DBA
- Administração de BD no *PostgreSQL*
 - *Usuários*
 - *Permissões*
 - *Backup e Restore*
 - *Vacumm*
- Dicas

Tarefas de Administração de BD

- São funções do **DBA** (*Database Administrator*):
 - Modelagem de dados
 - Implementação de BD
 - Instalação e suporte a SGBDs
 - Gestão de segurança e integridade
 - Gestão de performance/desempenho
 - Monitoramento, manutenção, *backup/recovery* de BD
 - Suporte à usuários

Conhecimentos necessários para um DBA

- Arquitetura de computadores
- Sistemas operacionais
- Redes
- Projeto conceitual e lógico de BD
- Arquitetura de SGBDS
- BD para DW
- Gerenciamento de esquemas
- Ferramentas de administração de BD

Administração de BD no *PostgreSQL*

- Comando **CREATE DATABASE** e *templates*
 - O comando CREATE DATABASE copia um BD já existente.
- Exemplos:
 - *template1* – BD padrão do sistema
 - *template0* – BD *standard* (sem nenhuma configuração)
- *Templates* deveriam ser configurados como *read-only*.

Administração de BD no *PostgreSQL*

➤ *Schemas*

- Representam instâncias diferentes de um mesmo BD.
- São usados para:
 - Permitir que usuários diferentes usem o mesmo BD sem interferir uns com os outros;
 - Organizar objetos de um BD em grupos lógicos de forma à torna-los mais organizados;
 - Aplicações desenvolvidas por terceiros podem ser direcionadas para *schemas* diferentes, prevenindo o conflito com nomes de objetos pré-existentes no BD.

Administração de BD no *PostgreSQL*

- Gerenciamento de usuários e permissões
 - É similar ao gerenciamento de usuários de um SO.
 - Comando **CREATE USER *nome***
 - Comando **DROP USER *nome***

Administração de BD no *PostgreSQL*

- Gerenciamento de usuários e permissões
 - Atributos de usuário
 - Comando **CREATE USER *nome* CREATEUSER;**
 - Cria um super usuário (pode criar outros usuários)
 - **ALTER USER *nome* NOCREATEUSER;** – revoga o atributo de super usuário
 - Comando **CREATE USER *nome* CREATEDB; [NOCREATEDB]**
 - Comando **CREATE USER *nome* PASSWORD 'senha';**
 - Comando **ALTER USER *nome* PASSWORD 'nova_senha';**

Administração de BD no *PostgreSQL*

- Gerenciamento de usuários e permissões
 - Grupos de usuários
 - **CREATE GROUP** *nome*;
 - **ALTER GROUP** *nome* **ADD USER** *nomeu_suario_1*, *nome_usuario_2*, ...;
 - **ALTER GROUP** *nome* **DROP USER** *nomeu_suario_1*, *nome_usuario_2*, ...;

Administração de BD no *PostgreSQL*

➤ Gerenciamento de usuários e permissões

➤ Permissões

- Todo objeto do BD tem um *owner* (proprietário), que é o usuário que o criou.
- Para que outros usuários tenham acesso a um objeto, é necessário definir *permissões*:
 - **SELECT, INSERT, UPDATE, DELETE, RULE, REFERENCES (fk), TRIGGER**
- Para atribuir privilégios usa-se o comando **GRANT**:
 - **GRANT UPDATE ON *tabela* TO *nome_usuario*;**
 - **GRANT SELECT ON *tabela* TO GROUP *nome_do_grupo*;**
 - **REVOKE ALL ON *tabela* FROM PUBLIC;**

Administração de BD no *PostgreSQL*

- Gerenciamento de usuários e permissões

- Permissões de *Schemas*

- **GRANT USAGE ON SCHEMA *nome_schema* TO *nome_usuario*;**

- **GRANT CREATE ON SCHEMA *nome_schema* TO *nome_usuario*;**

Administração de BD no *PostgreSQL*

➤ *Backup e Restore*

- Deve-se realizar regularmente o *backup* de BDs.
- No *PostgreSQL* existem dois tipos fundamentais de *backup*:
 - SQL *dump*
 - *Backup* a nível de Sistema Operacional

Administração de BD no *PostgreSQL*

➤ *Backup e Restore*

➤ *SQL dump* (backup lógico)

- Gera-se um arquivo texto com comandos SQL que permitem recriar o BD no mesmo estado em que ele estava no momento do *dump*.
- `pg_dump nome_bd -U nome_usuario -h host -p port > arquivo.bkp`
- Deve ser executado por um super usuário.
- O *pg_dump* não bloqueia as operações rodando no BD durante sua execução.

Administração de BD no *PostgreSQL*

➤ *Backup e Restore*

➤ Restaurando um *dump*

➤ Cria-se um novo BD:

createdb -T *template0* *nome_bd*

➤ Executa-se o comando:

psql *nome_bd* -U *nome_usuario* -h *host* -p *port* < *arquivo.bkp*

➤ O *psql* irá recuperar todas as configurações de usuários, caso estas estiverem definidas.

Administração de BD no *PostgreSQL*

➤ *Backup e Restore*

➤ Restaurando um *dump*

➤ Gerando um *dump* diretamente em outro servidor:

```
pg_dump -U nome_usuario -h host_1 -p port nome_bd |  
psql -U nome_usuario -h host_2 -p port nome_bd
```

Administração de BD no *PostgreSQL*

➤ *Backup e Restore*

➤ Usando o *pg_dumpall*

➤ Gera um *backup* de todo o *cluster*, incluindo dados de usuários:

```
pg_dumpall -U nome_usuario -h host_1 -p port |  
psql -U nome_usuario -h host_2 -p port
```


Administração de BD no *PostgreSQL*

➤ *Backup e Restore*

- **Grandes BDs** podem gerar arquivos de *dump* maior do que o permitido pelo SO.
- Nesses casos, pode-se usar alternativas como:

```
pg_dump -U nome_usuario -h host -p port nome_bd |  
gzip > arquivo_backup_zipado.gz
```

e

```
createdb -U nome_usuario -h host -p port nome_bd
```

```
gunzip -c arquivo_backup_zipado.gz | psql nome_bd
```

Administração de BD no *PostgreSQL*

➤ *Backup e Restore*

➤ **Limitações**, alguns exemplos:

- O *pg_dump* pode gerar alguns problemas em função da ordem que é feito o *backup* de objetos;
- Problemas com funções usadas como valor *default* para colunas;
- Dependências circulares;
- Grandes objetos.

Administração de BD no *PostgreSQL*

➤ **Backup** a nível de SO (*backup* físico)

- Copia-se diretamente os arquivos que o PostgreSQL usa para guardar dados em um BD.

➤ Exemplo:

```
tar -cf backup.tar /usr/local/pgsql/data
```

Administração de BD no *PostgreSQL*

➤ *Backup* a nível de SO

➤ Limitações:

- O servidor DEVE estar em SHUT DOWN (ninguém usando).
- Deve-se copiar os arquivos de dados e arquivos de *log de commit* (*pg_clog/**).
- Na restauração, tem de restaurar todos os arquivos.
- Provavelmente o *backup* será maior que o realizado com *pg_dump*.
- É mais inseguro, pois pode dar problema com sistema de arquivos corrompido.

- Uma **ESTRATÉGIA DE BACKUP** deve combinar *backup* lógico e físico.

Dicas



➡ Curso *on-line* gratuito

➡ **Administração de *PostgreSQL***

<https://www.ginead.com.br/curso/curso-de-administracao-de-postgresql>

Contato



➡ Prof. Angelo Augusto Frozza, Dr.



angelo.frozza@ifc.edu.br

<http://www.ifc-camboriu.edu.br/~frozza>



@TilFrozza

<http://www.twitter.com/TilFrozza>

<http://about.me/TilFrozza>