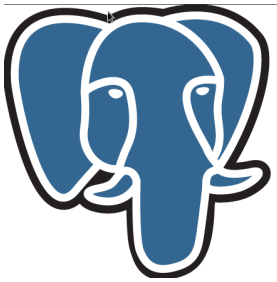

PostgreSQL

the world's most advanced open source database

Curso de PostgreSQL

Agnaldo Neto Marinho

PostgreSQL

Sumário

Capítulo 1

Introdução ao curso

Seja bem-vindo ao **Curso de PostgreSQL**. Este curso esta sendo fomentado pela *Centro de Tecnologia da Informação e Comunicação - CTIC* da *Universidade Federal do Pará* e ministrado por *Agnaldo Neto Marinho*. Realiza(ou)-se de *03/09/2015 a 09/09/2015*.

Os procedimentos descritos neste material foram validados sob a distribuição Debian GNU/Linux Jessie, todavia a base teórica ministrada é o conhecimento fundamental para a aplicabilidade dos procedimentos técnicos sob qualquer sistema operacional.

Apesar das peculiaridades de cada sistema operacional o conteúdo será abordado de forma isenta, para que o participante tenha a possibilidade de utilizar o conhecimento adquirido no ambiente que lhe for mais adequado.

Neste capítulo, serão abordados os seguintes temas: licenciamento deste material, origem do *software* utilizado (executável ou do código fonte) e configurações essenciais da distribuição Debian GNU/Linux.

1.1 Licença do material

1.2 Compilação de *software* X *software* da distribuição/sistema operacional

O acesso ao código fonte do *software* e sua compilação, é uma das liberdades propiciadas pelo *software livre*. Entretanto, o *software* também pode ser obtido em forma executável (compilada), e de forma integrada ao sistema operacional (empacotado), já estando pronto para utilização. Cada uma destas opções possui vantagens e desvantagens que serão enumeradas a seguir:

Características do *software* obtido na forma de executável (previamente compilado):

- **V:** Instalação rápida que requer menos espaço em disco; evita a compilação do *software*, assim como, a instalação de *software* de compilação (make, gcc, etc) e cabeçalhos de bibliotecas (libc6-dev, etc)
- **V:** Instalação automatizada de *software* e de bibliotecas necessárias (dependências) para o funcionamento do *software* principal.
- **V:** Versão testada pelo distribuidor do *software* (em geral o distribuidor do sistema operacional), e possivelmente livre de erros.
- **V:** Possibilita atualizações e correções de falhas de segurança de forma automática, e fornecida pelo distribuidor do sistema operacional.
- **V:** Facilita suporte externo devido ao método de instalação padronizado e utilização de versões invariantes do *software*.
- **V/D:** Pode não ser a versão mais nova do *software*, e não possuir funcionalidades mais recentes. Todavia, a utilização de versões maduras, tende a fornecer maior estabilidade.

Características do *software* obtido a partir do código fonte:

- **D:** Instalação mais complexa e demorada, demanda instalação manual de bibliotecas externas.
- **D:** Atualizações e correções são manuais, exigindo atenção diária às atualizações necessárias para correções de falhas de segurança.
- **D:** Dificulta suporte externo pois não é um método de instalação padronizado.
- **V/D:** Permite utilizar a última versão do *software*, com os novos recursos, mas trata-se de código menos testado podendo possuir falhas não detectadas.
- **V:** Pode permitir um ganho de performace com a compilação com otimizações do processador, e também com o desligamento de recursos não utilizados do *software*.

Após o levantamento destas características, é notável que em ambientes corporativos a utilização de *software* fornecido por um distribuidor é essencial para continuidade da disponibilidade dos sistemas.

Diminui-se o esforço empregado para manter o parque tecnológico atualizado e livre de falhas. Dessa forma, o treinamento utilizará os pacotes fornecidos pelo distribuidor do sistema operacional escolhido.

1.3 Debian GNU/Linux

Os sistemas operacionais baseados em tecnologias livres tendem a fornecer *software* que realizam instalações automatizadas. O Debian fornece os utilitários **apt-get** e **aptitude** para esta funcionalidade, sendo o segundo, sucessor e atualmente de uso recomendado.

O Debian fornece repositórios web que contém os *software* disponíveis para instalação, e estes são distribuídos em forma de pacotes: arquivos compactados com rotinas de pré/pós instalação e remoção, e informações sobre dependências, recomendações e sugestões de software adicionais.

O comportamento padrão do utilitário *aptitude* ao instalar um *software* é realizar a instalação das **dependências**, e também daqueles especificados como **recomendações**. Entretanto, este comportamento induz a instalação de *software* não requeridos, e demanda a utilização de espaço em disco adicional.

A instalação automática de *software* recomendado pode ser desabilitada através da adição da configuração abaixo ao arquivo **/etc/apt/apt.conf**:

- Debian Jessie (aptitude 0.4.11):

```
Apt::Install-Recommends "false";
```

O utilitário *aptitude* também requer a configuração da fonte dos *software* a serem instalados, e isto é realizado no arquivo **/etc/apt/sources.list**, conforme indicado no quadro abaixo.

```
deb http://ftp.br.debian.org/debian jessie main contrib non-free
deb http://security.debian.org/ jessie/updates main
```

Caso a conectividade seja fornecida por um proxy via http, a seguinte configuração deve ser adicionada ao arquivo **/etc/apt/apt.conf**, com a devida adequação ao endereço IP do proxy:

```
Acquire::http::Proxy "http://172.16.0.1:3128/";
```

Após a definição das fontes, é necessário o *download* da lista de *software* disponíveis, que é formada por informações como versão e descrição de cada *software*. Esse *download* deve ser realizado através do comando:

```
  # aptitude update
```

A lista de *software* disponíveis pode ser consultada, como indicado no exemplo abaixo:

- Pesquisar pelo nome do *software*:

```
  # aptitude search postgresql
```

- Pesquisar nas descrições do *software*, equivalente ao *apt-cache search openldap*:

```
  # aptitude search ~d'postgresql'
```

Maiores informações sobre um determinado *software* podem ser obtidas como indicado a seguir:

```
  # aptitude show postgresql
```

Capítulo 2

Entendendo um Banco de Dados

2.1 Bancos de Dados Relacionais

Um banco de dados é uma aplicação que lhe permite armazenar e obter de volta dados com eficiência. O que o torna *relacional* é a maniera como os dados são armazenados e organizados no banco de dados.

Quando falamos em banco de dados, aqui, nos referimos a um banco de dados relacional - RDBMS *Relational Database Management System.*

Em um banco de dados relacional, todos os dados são guardados em tabelas. Estas têm uma estrutura que se repete a cada linha, como você pode observar um uma planilha. São os relacionamento entre as tabelas que as tornam "relacionais"

- O modelo relacional surgiu devido às seguintes necessidades: - Aumentar a independência de dados nos sistemas gerenciadores de bancos de dados;

  - Prover um conjunto de funções apoiadas em álgebra relacional para armazenamento e recuperação de dados;

- A estrutura fundamental do modelo relacional é a relação. Uma relação é constituída por um ou mais atributos (campos), que traduzem o tipo de dados armazenados. Cada instância do esquema (linha), designa-se por tupla (registro). - O modelo implementa estruturas de dados organizados em relações (tabelas).

2.2 Banco de Dados Objeto-Relacional

- O PostgreSQL é normalmente considerado um sistema gerenciador de banco de dados relacional (SGBD-R, ou RDBMS, em inglês.) Entretanto, o PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional (SGBD-OR).

- Por ser objeto-relacional, o PostgreSQL suporta recursos inexistentes a um banco de dados puramente ralacional, tais como: herança entre tabelas, arrays em colunas e sobrecarga de funções.

:

Capítulo 3

Introdução ao Postgresql

O PostgreSQL é um SGBD (Sistema Gerenciador de Banco de Dados) objeto relacional de código aberto, com mais de 15 anos de desenvolvimento. é extremamente robusto e confiável, além de ser extremante flexivel e rico em recursos. Ele é considerado objeto relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados.

3.1 O que é o PostgreSQL?

- O PostgreSQL é um dos bancos de dados abertos mais utilizados atualmente, possui recursos avançados e compete igualmente com muitos bancos de dados comerciais.
- O banco de dados PostgreSQL nasceu na Universidade de Berkeley, em 1986, como um projeto acadêmico e se encontra hoje na versão 9.1, sendo um projeto mantido pela comunidade de *Software Livre*.
- A coordernação de desenvolvimento do PostgreSQL é executado pelo *PostgreSQL Global Development Group* que conta com um grande número de desenvolvimento ao redor do mundo.
- Ele é um SGBD muito adequado para o estudo universitário do modelo relacional, além de ser uma ótima opção para empresas implatarem soluções de alta confiabilidade sem altos custos de licenciamento.
- É um programa distribuido sob a licença BSD, o que torna o seu código fonte disponível e o seu uso livre para aplicações comercias ou não.
- O PostgreSQL foi implementado em diversos ambientes de produção no mundo, entre eles, um bom exemplo do seu pontencial é o banco de dados que armazena os registro de domínio .org, mantido pela empresa Afilias.

3.2 Principais Funcionalidades

- Banco de dados objeto-relacional
  - Herança entre as tabelas
  - Sobrecarga de funções
  - Colunas do tipo array
- Suporte a transações (padrão ACID)

- Lock por registro (row level locking)
- Integridade referencial
- Sub-consultas.
- Controle de concorrência multi-versão (MVCC);
- Funções armazenadas (Stored Procedures), que podem ser escritas em várias linguagens de programação (PL/PgSQL, Perl, Python, Ruby, e outras);
- Gatilhos (Triggers);
- Tipos definidos pelo usuário;
- Esquemas (Schemas);
- Conexões SSL.
- Áreas de armazenamento (Tablespaces)
- Pontos de salvamento (Savepoints)
- Commit em duas fases
- Arquivamento e restauração do banco a partir de logs de transação
- Diversas ferramentas de replicação
- Extensões para dados geoespaciais, indexação de textos, xml e várias outras.
- Acesso via drivers ODBC e JDBC, além do suporte nativo em várias linguagens
- Suporte ao armazenamento de BLOBs (binary large objects)
- Sub-queries e queries na cláusula FROM
- Sofisticado mecanismo de tuning
- Suporte a conexão de banco de dados seguras (criptografia)
- Modelo de segurança para o acesso aos objetos do banco de dados por roles
- Triggers views e functions (PL/pgSQL, Perl, Python e Tcl
- Mecanismos próprio de logs

3.3 Plataforma Suportadas

- IBM AIX
- FreeBSD< OpenBSD, NetBSD
- HP-UX
- Irix
- Linux

- MarcOS X
- Microsoft Windows (suporte nativo desde a versão 8.0)
- SCO Open Server
- Sun Solaris
- Tru64 Unix
- Unix Ware

Capítulo 4

Intefaces de Acesso ao PostgreSQL

No jargão de banco de dados, o PostgreSQL utiliza o modelo cliente-servidor. Uma sessão do PostgreSQL consiste nos seguintes processos (programas) cooperando entre si:

4.1 Interfaces de Acesso ao Banco de Dados

- O PostgreSQL pode ser acessado a partir de várias linguagens, entre elas estão:
  - C, C++
  - Java (JDBC)
  - PHP, JSP, ColdFusion
  - TCL/Tk
  - Perl
  - Python
  - ODBC (ASP, Delphi ou qualquer linguagem que suporte ODBC)

4.2 Conexão JDBC

- Abaixo um exeomplo de conexção utilizando drive JDBC:

```
1        public Connection connect() {
2            driver = "org.postgresql.Driver";
3            url = "jdbc:postgresql://172.16.128.13:5432/teste?user=
4                postgres";
5            try{
6               Class.forName(drive).newInstance();
7               con = DriverManager.getConnection(url);
8            }
9            catch (Exception e){
10               System.out.println("Error");
11               e.printStackTrace();
12           }
13           return con;
14       }
```

- Para o URL de conexão temos as opções:
  - **User**: usuário para conexão.
  - **Password**: senha para a conexão.
  - **Database**: nome do banco de dados a se acessado.
  - **Port**: porta de conexão.
  - **IP**: endereço IP do servidor.

4.3 Configuração ao PostgreSQL

Depois de baixar e instalar é hora de configurar. O usuário root do nosso banco de dados é o **postgres**. No processo de instalação foi criado um usuário chamdo postgres também no sistema. Então, nos logaremos com este usuário.

```
# Su postgres
```

Por padrão, **o usuário de banco de dados 'postgres'** não tem senha, então agora nos logaremos no shell do PostgreSQL para alterar a senha do usuário postgres.

Primeiro, logando no shell...

```
$ psql
```

Agora, já no shell do PostgreSQL, vamos alterar a senha do usuário postgres

```
postgres=# ALTER USER postgres WITH PASSWORD 'qualquersenha';
```

Esse cara que a gente acabou de configurar ai é o root do banco de dados... A gente não vai ficar usando esse usuário nas nossas aplicações, né? Então! vamos criar um novo usuário.

```
postgres=# CREATE USER usuario NOCREATEDB NOSUPERUSER NOCREATEROLE PASSWORD
        'senha';
```

Agora, vamos criar uma tabela também

```
postgres=# CREATE DATABASE minhabase;
```

4.4 Introdução ao psql

- O psql é o modo interativo do PostgreSQL para acesso e manipulação dos bancod de dados.

  ```
  psql [-h hostname -p port -U user -W] [database]
  ```

- Onde:
  - **hostname**: nome ou IP do servidor (padrão é localhost)
  - **port**: porta de conexão (padrão ŕ 5432)
  - **user**: usuário postgresql (padrão é o usuário de sistema operacional)
  - **database**: nome do banco de dados.
- A opção -w força a entrada da senha do usuário.

o

Capítulo 5

Liguagem SQL

5.1 A Linguagem SQL

- SQL (Structured Query Languagem) é uma linguagem declarativa de acesso à banco de dados.
- Por ser uma linguagem padronizada, a migração para o PostgreSQL é fácilitada para aqueles que conhecem SQL.
- O PostgreSQL está em conformidade com a maior parte das espeficições SQL92 e SQL99.
- A linguagem SQL não considera a caixa dos comandos (*xase insensitive*). Entretanto, a caixa faz diferença para os leitores entre *aspas*.

5.2 Introdução

Este capítulo fornece uma visão geral sobre como utilizar a linguagem SQL para realizar operações simples. O propósito deste tutorial é apenas fazer uma introdução e, de forma alguma, ser um tutorial completo sobre a linguagem SQL. É preciso estar ciente que algumas funcionalidades da linguagem SQL do PostgreSQL são extensões ao padrão.

Conforme criando o usuario e banco de dados, conforme descrito no capítulo anterior, e que o psql esteja ativo.

5.3 Criação de Tabelas

Pode-se criar uma tabela especificando o seu nome juntamente com os nomes das colunas e seus tipos de dado:

```
CREATE TABLE clima {
    ciade           char(80),
    tem_min         int,            -- temperatura mínima
    temp_max        int,            -- temperatura máxima
    prcp            real,           -- precipitação
    data            date
}
```

Apêndice A

Licença