

Treinamento PostgreSQL - Aula 01

Eduardo Ferreira dos Santos

SparkGroup
Treinamento e Capacitação em Tecnologia
eduardo.edusantos@gmail.com
eduardosan.com

27 de Maio de 2013

Sumário

1 Apresentação

2 Introdução

- Conceito de banco de dados
- Histórico do PostgreSQL
- Comunidade
- Desenho Conceitual
- Representação de dados
- Armazenamento e Indexação

3 Teoria Relacional

- Modelo de dados e definições
- Gerenciamento de tabelas
- Restrições, integridade referencial e herança

4 Referências

Ementa - Administração de Dados (Parte 1)

Introdução Conceito de banco de dados, Histórico do PostgreSQL, Comunidade, Desenho Conceitual, Representação de dados, Armazenamento, Indexação;

Teoria Relacional Modelo de dados e definições, Gerenciamento de tabelas, Restrições e integridade referencial, Herança entre tabelas;

Conjuntos Álgebra relacional e operações de conjuntos

Linguagem SQL Sintaxe, Definição de dados, Manipulação de dados, Consultas, Tipos de dado, Funções e operadores, Conversão de tipos, Índices, Busca textual (*Full Text Search*), Controle de concorrência, Análise de performance;

Administração do Banco de Dados

PostgreSQL em GNU/Linux Instalação e Configuração;

Manipulação da estrutura do banco de dados Manipulação de *tablespaces* e *schemas*;

Administração do Servidor Segurança lógica e física, Monitoramento, Ferramentas administrativas e Backup;

PTR *PITR - Point-in-Time Recovery*;

Migração Desenhando um projeto de migração de dados para o PostgreSQL.

Alta disponibilidade

Cluster Virtualização, Arquitetura e definições;

Escalabilidade horizontal Soluções de replicação de base de dados
Master-Slave, Multimaster;

Escalabilidade vertical Banco de dados distribuídos, Replicação de discos;

Escalabilidade nativa *Streaming Replication/Hot Standby;*

Administração do cluster Balançamento de carga, Alta disponibilidade com
Heartbeat.

Performance Tuning

- PostgreSQL por dentro Estrutura do PostgreSQL, Sistema Operacional, Disco e Regras gerais de performance;
- Otimização de SQL Análise de consultas e Plano de execução;
- Otimizando os discos Configurações de I/O, Tabela de partições e Tipos de disco;
- Configuração do PostgreSQL *shared_buffers* e referência do arquivo [postgresql.conf](#);
- Otimizando o SO Otimização de Kernel para GNU/Linux, Memória e Disco;

Cronograma

Semana 1: 27-31 de Maio Administração de Dados

Semana 2: 3-7 de Junho Administração de Banco de Dados

Semana 3: 10-14 de Junho Alta disponibilidade

Semana 4: 17-21 de Junho Performance Tuning

Para começar

- Banco de dados **não é para amadores**.
- O Sistema Operacional pode ser o melhor amigo (ou inimigo) do DBA.
- Especificar corretamente o hardware **DEVE** ser trabalho do DBA, pois é **extremamente importante**.
- Os riscos dos erros do DBA são sempre maiores. Aprenda a conviver com o **conservadorismo**.
- Estudar é importante, mas para um DBA **é muito mais importante!**

Vale lembrar

- Nenhum software vai administrar o banco pra você. Ajudam se forem bem operados;
- Se você não sabe o que é partição, tipo de sistema de arquivos, paginação, estratégias de *buffering*, volte algumas casas.
- Está assustado?

Vale lembrar

- Nenhum software vai administrar o banco pra você. Ajudam se forem bem operados;
- Se você não sabe o que é partição, tipo de sistema de arquivos, paginação, estratégias de *buffering*, volte algumas casas.
- Está assustado?
- É possível se tornar um grande DBA sem sair de casa. Acesse www.postgresql.org/doc e seja feliz!
- Infelizmente, não é possível evoluir muito em nossa profissão sem o Inglês. Se quiser ser realmente bom, domine a língua primeiro.
- Finalmente, aprenda fazendo. As empresas valorizam muito o estudo, mas para administrar dados tem que ter experiência.

1 Apresentação

2 Introdução

- **Conceito de banco de dados**
- Histórico do PostgreSQL
- Comunidade
- Desenho Conceitual
- Representação de dados
- Armazenamento e Indexação

3 Teoria Relacional

- Modelo de dados e definições
- Gerenciamento de tabelas
- Restrições, integridade referencial e herança

4 Referências

Bancos de Dados

- O que é um banco de dados?
- O que é um dado?
- Sistema Gerenciados de Banco de Dados (SGBD)[ETEC, 2013]:
 - Gerenciar grande quantidade de informação;
 - Evitar redundância de dados e inconsistência;
 - Facilitar o acesso;
 - Segurança no acesso aos dados;
 - Garantir a integridade;
 - Facilitar a migração se necessário.

Abstração de dados

- Esconde os detalhes de como os dados são armazenados ou mantidos[ETEC, 2013]:
 - Abstração física Como os dados são armazenados no disco;
 - Abstração conceitual Quais dados são armazenados e como estão relacionados;
 - Abstração de visão Expõe somente a parte do Banco de Dados que o usuário necessita. Um mesmo SGBD pode ter **diversas visões** do mesmo dado.

Tipos de Bancos de Dados

- O dado a ser armazenado e o tipo de abstração afetam o Tipo de Banco de Dados a ser utilizado

Tipos de Bancos de Dados

- O dado a ser armazenado e o tipo de abstração afetam o Tipo de Banco de Dados a ser utilizado
- Bancos de Dados Relacionais

Tipos de Bancos de Dados

- O dado a ser armazenado e o tipo de abstração afetam o Tipo de Banco de Dados a ser utilizado
- Bancos de Dados Relacionais
- Bancos de Dados Orientados a Objetos

Tipos de Bancos de Dados

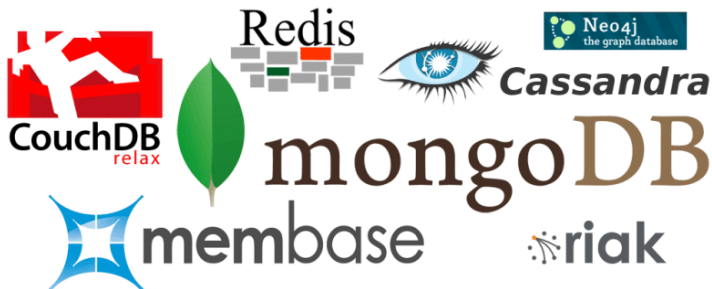
- O dado a ser armazenado e o tipo de abstração afetam o Tipo de Banco de Dados a ser utilizado
- Bancos de Dados Relacionais
- Bancos de Dados Orientados a Objetos
- Bancos de Dados Documentais

Bancos de Dados Relacionais



Bancos de Dados Relacionais

Bancos de Dados Orientados a Objetos (NoSQL)



Bancos de Dados NoSQL

Bancos de Dados Documentais



Lightbase

1 Apresentação

2 Introdução

- Conceito de banco de dados
- **Histórico do PostgreSQL**
- Comunidade
- Desenho Conceitual
- Representação de dados
- Armazenamento e Indexação

3 Teoria Relacional

- Modelo de dados e definições
- Gerenciamento de tabelas
- Restrições, integridade referencial e herança

4 Referências

Origens

- 1985 Primeiro desenho de um possível sistema de armazenamento (ainda projeto Postgres) na universidade de Berkeley, Califórnia [Stonebreaker and Rowe, 1985]
- 1987 Primeira modelagem [Stonebreaker and Rowe, 1987]
- 1989 Definição do primeiro sistema de regras [Stonebreaker et al., 1989]
Fim da fase acadêmica

Origens

- 1985 Primeiro desenho de um possível sistema de armazenamento (ainda projeto Postgres) na universidade de Berkeley, Califórnia [Stonebreaker and Rowe, 1985]
- 1987 Primeira modelagem [Stonebreaker and Rowe, 1987]
- 1989 Definição do primeiro sistema de regras [Stonebreaker et al., 1989]
Fim da fase acadêmica
- 1993 Empresa Illustra transforma parte do código em um produto comercial. Mais tarde a empresa se tornaria Informix, até ser comprada pela IBM
- 1993 Desenvolvimento de uma versão paralela pela comunidade (principalmente universidades) e novo nome: PostgreSQL

1 Apresentação

2 Introdução

- Conceito de banco de dados
- Histórico do PostgreSQL
- **Comunidade**
- Desenho Conceitual
- Representação de dados
- Armazenamento e Indexação

3 Teoria Relacional

- Modelo de dados e definições
- Gerenciamento de tabelas
- Restrições, integridade referencial e herança

4 Referências

No Brasil

- Crescimento através da comunidade
 - 1 Registro do domínio postgresql.br;
 - 2 Criação da lista de discussão PostgreSQL BR (<http://listas.postgresql.org.br/cgi-bin/mailman/listinfo/pgbr-geral>);
 - 3 Crescimento da utilização no nicho dos DBA's voltado para migração do Oracle;
 - 4 Lançamento da versão Windows;
 - 5 Realização da primeira Conferência Brasileira de PostgreSQL¹.

¹PGConBr, realizada em São Paulo em 2007:

<http://pgbr.postgresql.org.br/2007/>

A Comunidade PostgreSQL

Não se sinta sozinho: seja bem-vindo à nossa comunidade:
www.postgresql.org.br / listas.postgresql.org.br

- Lá você encontra:
 - Listas de discussão;
 - Documentação;
 - Amigos :)



1 Apresentação

2 Introdução

- Conceito de banco de dados
- Histórico do PostgreSQL
- Comunidade
- **Desenho Conceitual**
- Representação de dados
- Armazenamento e Indexação

3 Teoria Relacional

- Modelo de dados e definições
- Gerenciamento de tabelas
- Restrições, integridade referencial e herança

4 Referências

O Desenho do Postgres³

- Considerado o sucessor do INGRES
- Principais escolhas de *design*:
 - 1 Suportar objetos complexos;
 - 2 Fornecer ao usuário **extensibilidade** para os tipos de dado, operadores e métodos de acesso;
 - 3 Prover funcionalidades para manter a inferência cruzada de relacionamentos consistente. Ex.: Alerts e Triggers;
 - 4 Simplificar o código para recuperação em caso de desastres do DBMS²;
 - 5 Utilizar-se do que há de melhor em termos de hardware para obter melhor performance dos componentes;
 - 6 Manter o mais inalterado possível o **modelo relacional**.

²DBMS ou *Database Management System* é o Sistema Gerenciador de Banco de Dados

³Do artigo original em inglês, *The Design of Postgres* [Stonebreaker and Rowe, 1985]

1 Apresentação

2 Introdução

- Conceito de banco de dados
- Histórico do PostgreSQL
- Comunidade
- Desenho Conceitual
- **Representação de dados**
- Armazenamento e Indexação

3 Teoria Relacional

- Modelo de dados e definições
- Gerenciamento de tabelas
- Restrições, integridade referencial e herança

4 Referências

Representando dados reais

- Objetos complexos: definição e representação;
- Trabalho com **tuplas**⁴ de dados:
 - Quádrupla ou tupla de quatro elementos: (1, 2, 3, 4)
 - Conjuntos próximos do mundo real: (*gato*, *cachorro*, *papagaio*)
 - Extrapolação para o modelo relacional:
(1, *gato*), (2, *cachorro*), (3, *papagaio*)
 - Tupla de três elementos (**tripla**) cada um uma tupla de dois elementos (**dupla**)
 - Definição de **tabela** para **animais de estimação**

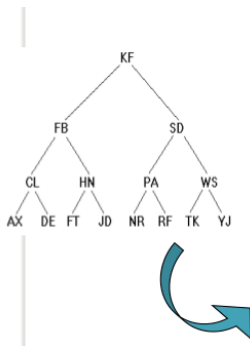
⁴Tupla pode ser definida como uma sequência ordenada de elementos acessíveis através de um índice inteiro

Representando dados reais

- Extensibilidade a outros domínios de aplicação:
 - Tipos de dado padrão: *inteiro*, *float*, *char*
 - Tipos de dado extendidos: *date*, *timestamp*, *money*
 - Objetos complexos? *polygon*, *point* (dados geométricos)

Representando dados reais

- Utilização de algoritmos de armazenamento e indexação: BTree, B+Tree, etc;



	chave	esq	dir		chave	esq	dir
1	KF	2	3	9	DE	-	-
2	FB	4	5	10	FT	-	-
3	SD	6	7	11	JD	-	-
4	CL	8	9	12	NR	-	-
5	HN	10	11	13	RF	-	-
6	PA	12	13	14	TK	-	-
7	WS	14	15	15	YJ	-	-
8	AX	-	-				

Representação Encadeada.

BTree

1 Apresentação

2 Introdução

- Conceito de banco de dados
- Histórico do PostgreSQL
- Comunidade
- Desenho Conceitual
- Representação de dados
- **Armazenamento e Indexação**

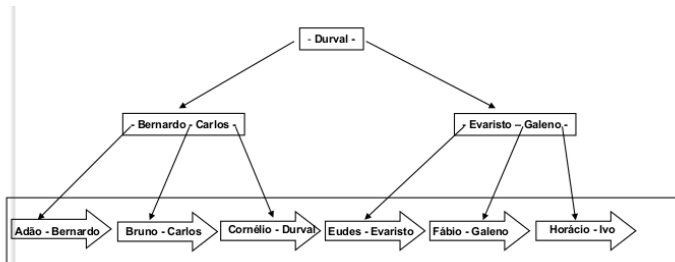
3 Teoria Relacional

- Modelo de dados e definições
- Gerenciamento de tabelas
- Restrições, integridade referencial e herança

4 Referências

Representando dados reais

- Utilização de algoritmos de armazenamento e indexação: BTree, B+Tree, etc;



B+Tree

Conceitos principais

- Foco em **teorias** para manipulação e armazenamento de dados;
- Armazenamento complexo e extensibilidade a outras aplicações;
- O principal objetivo é (e sempre foi) fazer o **melhor banco de dados**;
- Vamos chegar onde nenhum outro elefante jamais esteve: *new features*, melhor performance e mais confiabilidade.

- 1 Apresentação
- 2 Introdução
 - Conceito de banco de dados
 - Histórico do PostgreSQL
 - Comunidade
 - Desenho Conceitual
 - Representação de dados
 - Armazenamento e Indexação
- 3 Teoria Relacional
 - Modelo de dados e definições
 - Gerenciamento de tabelas
 - Restrições, integridade referencial e herança
- 4 Referências

O modelo de dados

- Modelo relacional estendido através de tipos de dado abstratos:
 - Operadores definidos pelos usuários;
 - Construção de procedimentos internos;
 - Procedimentos globais.
- **Tabela (definição)**: Coleção de relações que contém tuplas representando entidades do mundo real.
 - Coleção fixa de atributos;
 - **Chave primária (definição)**: Sequência de atributos da relação que identificam unicamente cada tupla.

O modelo de dados

- **Conceito:** dados organizados em tabelas ou relações;
- Organização em linhas (tuplas) e campos (atributos).

Emp		CódigoEmp	Nome	CodigoDepto	CategFuncional
E5	Souza	D1	C5		
E3	Santos	D2	C5		
E2	Silva	D1	C2		
E1	Soares	D1	—		

coluna (atributo)

nome do campo
(nome do atributo)

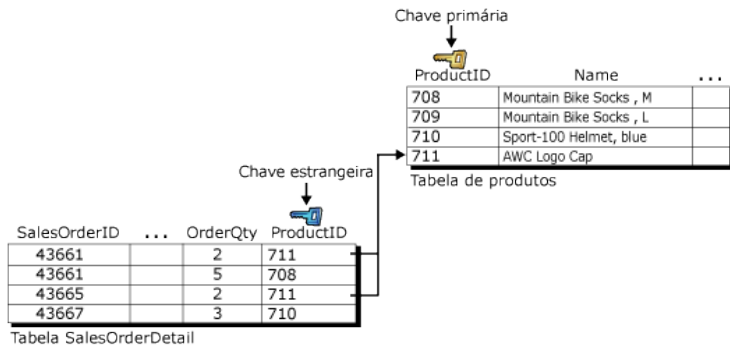
linha (tupla)

valor de campo
(valor de atributo)

- 1 Apresentação
- 2 Introdução
 - Conceito de banco de dados
 - Histórico do PostgreSQL
 - Comunidade
 - Desenho Conceitual
 - Representação de dados
 - Armazenamento e Indexação
- 3 Teoria Relacional
 - Modelo de dados e definições
 - **Gerenciamento de tabelas**
 - Restrições, integridade referencial e herança
- 4 Referências

Utilizando os dados

- O dado na tupla é acessível através de um índice numérico;
- Para os índices numéricos fixos definimos **chaves**;
- Chaves primárias e chaves estrangeiras;
- Domínio da coluna ou domínio do campo;



Fonte: [MSDN, 2013]

- 1 Apresentação
- 2 Introdução
 - Conceito de banco de dados
 - Histórico do PostgreSQL
 - Comunidade
 - Desenho Conceitual
 - Representação de dados
 - Armazenamento e Indexação
- 3 Teoria Relacional
 - Modelo de dados e definições
 - Gerenciamento de tabelas
 - Restrições, integridade referencial e herança
- 4 Referências

Restrições

- **Restrição (definição)**: conjunto de regras ou valores permitidos aos campos;
- Objetivo principal de todo SGBD é manter a **integridade dos dados**;
- Os dados armazenados são consistentes entre si;
- **Restrição de integridade (definição)**: regra de consistência dos dados garantida pelo próprio SGBD;

Restrições

- Restrições de integridade podem ser classificadas como:
 - Integridade de domínio** Campo deve obedecer o conjunto de valores permitidos para a coluna;
 - Integridade de vazio** Indica se os campos de uma coluna podem ou não ser vazios;
 - Integridade de chave** Define que as chaves devem ser únicas;
 - Integridade referencial** Os campos definidos na chave estrangeira devem aparecer na chave primária da tabela referenciada.

Criando um banco de dados

- Segundo [Heuser, 1998], um banco de dados relacional deve conter no mínimo as seguintes definições:
 - 1 Tabelas que formam o banco de dados;
 - 2 Colunas que as tabelas possuem;
 - 3 Restrições de integridade.

Resumo dos conceitos

- O modelo relacional utiliza informações organizadas em **tuplas**;
- **Tabelas** são coleções de relações que contém tuplas representando entidades do mundo real;
- Cada **tupla** deve ter um **identificador** (chave) e um conjunto de **campos** (atributos);
- As **restrições de integridade** representam o conjunto de valores permitidos ao campo;
- Um **banco de dados relacional** deve ter no mínimo tabelas, colunas e restrições de integridade.



ETEC (2013).

Apostila completa de banco de dados.

http://www.etefernandoprestes.com.br/arquivos_geral/downloads/BD.pdf

Acessado em 22/05/2013.



Heuser, C. A. (1998).

Projeto de Banco de Dados.

Bookman.



MSDN (2013).

Modelos de chave.

<http://msdn.microsoft.com/pt-br/library/ms184276v=sql.105.aspx>

Acessado em 22/05/2013.



Stonebreaker, M. and Rowe, L. A. (1985).

The design of postgres.

<http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M85-95.pdf>

Acessado em 22/05/2013.



Stonebreaker, M. and Rowe, L. A. (1987).

The postgres datamodel.

<http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M87-13.pdf>

Acessado em 22/05/2013.



Stonebreaker, M., Rowe, L. A., and Potamianos, S. (1989).

Postgres rule system.

<http://www.postgresql.org/docs/9.2/interactive/biblio.html#STON87A>

Acessado em 22/05/2013.

Contato

Eduardo Ferreira dos Santos
Sparkgroup
Lightbase Consultoria em Software Público

eduardo.santos@lightbase.com.br
eduardo.edusantos@gmail.com

www.postgresql.org.br
www.eduardosan.com

+55 61 3347-1949