



UNIVERSIDADE
FEDERAL DO CEARÁ
Campus Russas


Disciplina: Fundamentos de Banco de Dados

11. SQL (Parte I)

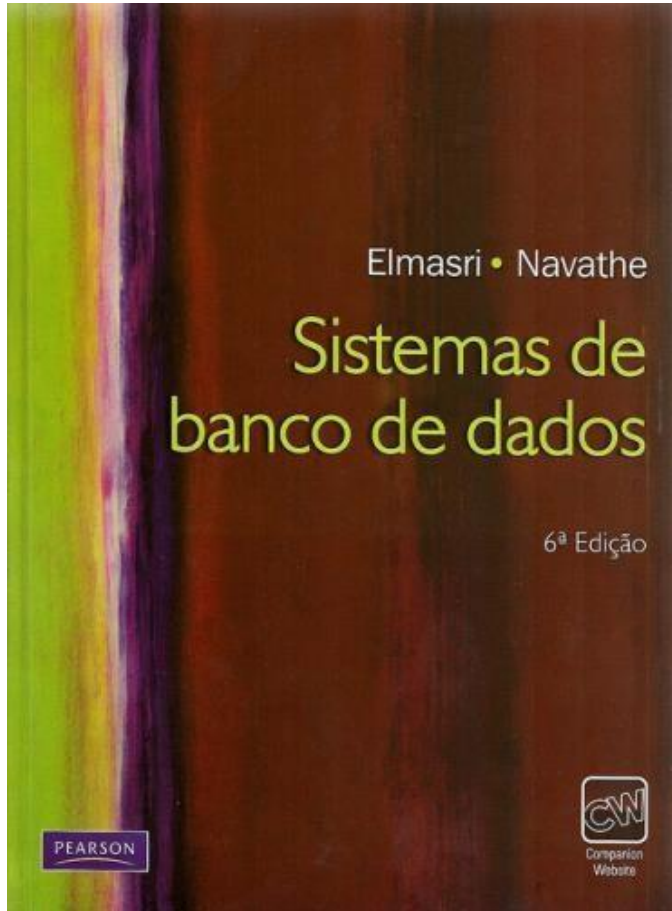
Professora: Marília S. Mendes

E-mail: marilia.mendes@ufc.br

Organização da disciplina

- ☑ Conceitos Gerais
 - ☑ SGBD e modelo de dados
 - ☑ Modelo ER
 - ☑ Modelo Relacional
 - ☑ Álgebra Relacional
 - ☑ Mapeamento ER-Relacional
 - ☑ Normalização
 - ☐ SQL ← 
-

Onde encontrar a matéria?



► Capítulo 4

SQL – Introdução

- *Structured Query Language* (SQL), ou Linguagem de Consulta Estruturada, é uma linguagem de pesquisa declarativa para banco de dados relacionais.
 - A linguagem SQL é um grande padrão de banco de dados, é simples e fácil de usar!
 - Uma consulta SQL especifica **a forma do resultado** e não o caminho para chegar a ele.
 - Introduzida como linguagem de consulta para o sistema R. Trata conjuntos de tuplas.
-

SQL – Tipos de comandos

- ***Data Definition Language (DDL)*** – Linguagem de Definição de Dados, que permite determinar o esquema do banco de dados, bem como alterá-lo e excluí-lo e trabalha com os metadados;
 - ▶ Create, Drop, Alter
 - ***Data Manipulation Language (DML)*** – Linguagem de Manipulação de Dados. Permite a manipulação dos dados, ou seja, inclusão, alteração e exclusão dos dados;
 - ▶ Insert, Update, Delete
-

SQL – Tipos de comandos

- **Data Control Language (DCL)** – Linguagem de Controle de Dados. Permite controlar a licença e a autorização dos usuários para com os dados.
 - ▶ Grant, Revoke
 - ▶ **Data Transaction Language (DTL)** – Linguagem de Transação de Dados, que oferece comandos para se trabalhar com as transações
 - ▶ Commit, Rollback
 - ▶ **Data Query Language (DQL)** – Linguagem de Consulta de Dados, que proporciona a consulta de dados
 - ▶ Select
-

Principais banco de dados

	MySQL	PostgreSQL	Firebird	Oracle	SQL Server
SGBD	Sim	Sim	Sim	Sim	Sim
ACID	Sim	Sim	Sim	Sim	Sim
Licença Comercial	Não	Sim	Sim	Não	Não
Licença Estudante	Sim	Sim	Sim	Oracle Express	SQL Server Express

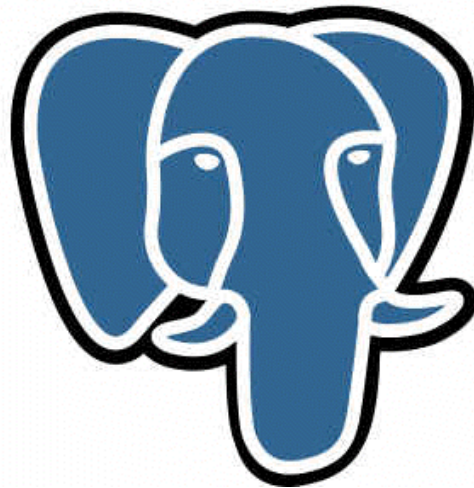
PostgreSQL

- Desenvolvido no Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley, onde o professor Michael Stonebraker liderou o projeto.
 - A implementação do SGBD POSTGRES começou em 1986.
 - A primeira versão de demonstração do sistema se tornou operacional em 1987, e foi exibida em 1988 na Conferência ACM-SIGMOD.
 - Em 1994, Andrew Yu e Jolly Chen adicionaram um interpretador da linguagem SQL ao POSTGRES.
 - Postgre95 \Rightarrow PostgreSQL
-

PostgreSQL

- Um software Open Source
- Versão atual: PostgreSQL 9.3.5.1

PostgreSQL



Principais Tipos de Dados

- **Char:** um caractere
- **Varchar(n):** cadeia de caracteres de tamanho n
- **Int:** números inteiros
- **Float (n):** numero real com precisão de n dígitos
- **Double:** numero real
- **Numeric (p,d):** números formatados.
Ex.: numeric (6,2)
- **Date:** Data no formato ano-mês-dia.
Ex: 2004-12-04
- **Time:** Hora no formato horas:minutos:segundos. Ex.: 09:12:47

Os tipos de dados
dependem do
SGBD utilizado!



Aula de hoje

SQL - DDL

- ▶ CREATE TABLE
- ▶ ALTER TABLE
- ▶ DROP TABLE
- ▶ CREATE INDEX
- ▶ ALTER INDEX
- ▶ DROP INDEX

SQL - DML

- ▶ INSERT
- ▶ DELETE
- ▶ UPDATE



Definição de dados, restrições e mudanças no esquema

- ▶ A DDL permite a criação (CREATE), remoção (DROP) e modificação (ALTER) das descrições das tabelas (relações) de uma base de dados.

SQL - DDL

- ▶ Data Definition Language (DDL): permite a criação dos componentes do banco de dados, como tabelas, índices etc.
 - ▶ Principais comandos DDL:
 - ▶ CREATE TABLE
 - ▶ ALTER TABLE
 - ▶ DROP TABLE
 - ▶ CREATE INDEX
 - ▶ ALTER INDEX
 - ▶ DROP INDEX
-

CREATE TABLE

- ➡ Esse comando SQL especifica uma nova relação da base de dados através da definição do seu nome, e da especificação de cada um de seus atributos com seus tipos de dados (INTEGER, FLOAT, DECIMAL(i,j), CHAR(n), VARCHAR(n))
 - ➡ A restrição NOT NULL pode ser especificada sobre um atributo
-

Exemplo – CREATE TABLE

```
CREATE TABLE DEPARTAMENTO  
( DNOME          VARCHAR(10)    NOT NULL,  
  DNUMERO        INTEGER        NOT NULL,  
  GERSSN         CHAR(9),  
  GERDATAINICIO CHAR(9)  
);
```

Exemplo – CREATE TABLE

- ➡ Em SQL2, o comando CREATE TABLE pode ser usado para especificar os atributos da chave primária, chave secundária, e as restrições de integridade referencial (chave estrangeira).
 - ➡ Atributos chave podem ser especificados através dos termos: PRIMARY KEY e UNIQUE
-

100

1000

DROP TABLE

- ➡ Faz a remoção de uma relação (tabela da base) *com a sua definição*
 - ➡ A relação não poderá mais ser usada em consultas, modificações, ou qualquer outro comando já que a sua descrição não existe mais
-

DROP TABLE – Exemplo

DROP TABLE DEPENDENTE;

Recursos Adicionais da SQL2 e SQL-99

- ▶ **CREATE SCHEMA**
 - ▶ Opções para definir restrições de integridade referencial
REFERENTIAL INTEGRITY
-

CREATE SCHEMA

- ▶ Especifica o esquema de uma nova base de dados através de um nome.

Restrições (Constraints)

- ▶ **Constraints** são regras agregadas a colunas ou tabelas

- Restrições básicas
 - Restrições de chave e de integridade referencial.
 - Restrições sobre domínios de atributos e NULLs.
 - Restrições sobre tuplas individuais de uma relação.
-

Restrições (Constraints)

- PRIMARY KEY (PK)
 - Especifica um ou mais atributos para fazer parte da chave primária da relação.
 - Dnumero INT **PRIMARY KEY**;

PRIMARY KEY (DNUMERO)

Opções para Integridade Referencial

- ▶ Pode-se especificar as seguintes cláusulas sobre restrições de integridade referencial (chaves estrangeiras)
 - ▶ RESTRICT,
 - ▶ CASCADE,
 - ▶ SET NULL ou
 - ▶ SET DEFAULT

**O que fazer se uma
regra for violada?**

Restrições (Constraints)

► Chave estrangeira

FOREIGN KEY nome-chave-estrangeira (lista-de-colunas)

REFERENCES nome-tabela (lista-de-colunas)

ON UPDATE ação – determina qual ação o banco de dados deve tomar quando for excluída ou alterada uma linha da tabela que contém referencia a esta chave.

ON DELETE ação

- **SET NULL** (altera o conteúdo da coluna para nulo, perdendo a referência, sem deixar valores inconsistentes),
- **SET DEFAULT** (altera o conteúdo da coluna para o valor especificado na cláusula DEFAULT, se houver),
- **CASCADE** (exclui ou altera todos os registros que se relacionam a eles),
- **NO ACTION** (em caso de alteração, não modifica os valores que se relacionam a eles) ou
- **RESTRICT** (não permite a exclusão da chave primária)

Restrições (Constraints)

- FOREIGN KEY (FK)

- Implementa o conceito de chave estrangeira e garante a integridade referencial.
- Deve referenciar um campo que possua chave primária ou uma restrição UNIQUE.
- Operação padrão: rejeitar a atualização na violação.
- Conecta uma cláusula de ação de disparo referencial.
 - Opções incluem SET NULL, CASCADE e SET DEFAULT
 - Ação tomada pelo SGBD para SET NULL ou SET DEFAULT é a mesma para ON DELETE e ON UPDATE

Restrições (Constraints)

- **UNIQUE**
 - Especifica chaves alternativas (secundárias).
 - Garante que o atributo não terá valores repetidos na tabela.
 - Dnome VARCHAR(15) **UNIQUE**;
-

Restrições (Constraints)

- **NOT NULL**
 - O atributo deve ser obrigatoriamente preenchido.
 - NULL não é permitido para um determinado atributo.
 - **Valor padrão (DEFAULT)**
 - Atribui um valor padrão ao atributo, caso não seja especificado um valor.
 - DEFAULT <value>
 - **CHECK**
 - Verifica se o valor inserido é permitido para o atributo.
 - Dnumero INT NOT NULL CHECK (Dnumero > 0 AND Dnumero < 21);
-

Especificando restrições usando check

Cláusulas CHECK no final de uma instrução CREATE TABLE.

- Aplicada a cada tupla individualmente.
- `CHECK (Dep_data_criacao <= Data_inicio_gerente);`

Imagine um campo de sexo. Esse campo só poderá aceitar M para masculino ou F para feminino. O domínio dessa coluna é {M, F} e seria criado desta forma:

Constraint sexo `check` (sexo IN ('M','F'))

Dando nome a restrições

- Palavra-chave **CONSTRAINT**
 - Nomeia uma restrição.
 - Útil para posterior alteração.

Exemplo – Integridade Referencial

```
CREATE TABLE DEPARTAMENTO
( DNOME                VARCHAR(10)          NOT NULL,
  DNUMERO              INTEGER              NOT NULL,
  GERSSN               CHAR(9),
  GERDATAINICIO        CHAR(9),

  PRIMARY KEY (DNUMERO),
  UNIQUE (DNOME),
  FOREIGN KEY (GERSSN) REFERENCES EMPREGADO

  ON DELETE SET DEFAULT
  ON UPDATE CASCADE
);
```

Exemplo – Integridade Referencial

CREATE TABLE EMPREGADO

**(ENOME VARCHAR(30) NOT NULL,
 ESSN CHAR(9),
 DATANASC DATE,
 DNO INTEGER DEFAULT 1,
 SUPERSSN CHAR(9),**

PRIMARY KEY (ESSN),

**FOREIGN KEY (DNO) REFERENCES DEPARTAMENTO
ON DELETE SET DEFAULT
ON UPDATE CASCADE,**

**FOREIGN KEY (SUPERSSN) REFERENCES EMPREGADO
ON DELETE SET NULL
ON UPDATE CASCADE,**

);

ALTER TABLE

- ➡ Usado para acrescentar um atributo a uma das relações da base
 - ➡ O novo atributo terá o valor NULL em todas as tuplas da relação logo após a execução do comando; assim, a restrição NOT NULL *não pode ser* especificada para tal atributo
-

ALTER TABLE - Exemplo

```
ALTER TABLE EMPREGADO  
ADD CARGO VARCHAR(12);
```

➡ Os usuários da base de dados precisam, então, entrar com valores para esse novo atributo CARGO para cada tupla de EMPREGADO. Isso pode ser feito através do comando UPDATE.

ALTER TABLE

- ▶ Acrescentar novas colunas

ALTER TABLE *tabela*

ADD *nome-coluna tipo-de-dado constraints [,
nome-coluna tipo-de dado constraints, ...]*

- ▶ Exemplo

ALTER TABLE cliente

ADD email varchar(80) **UNIQUE**

ALTER TABLE

- ▶ Acrescentar novas *constraints*

ALTER TABLE *tabela*

ADD (*constraint*)

- ▶ Exemplo

ALTER TABLE cliente

ADD PRIMARY KEY (CDCLIENTE)

ALTER TABLE

- ▶ Modificar columnas:

ALTER TABLE *tabela*

MODIFY (*nome-coluna tipo-de-dado constraints*)

- ▶ Exemplo:

ALTER TABLE cliente

MODIFY email varchar(100) **NOT
NULL**

ALTER TABLE

- ▶ Excluindo elementos

ALTER TABLE tabela

DELETE elemento

- ▶ Exemplo 1 – exclusão de coluna:

ALTER TABLE cliente

DELETE email

- ▶ Exemplo 2 – exclusão de constraint de tabela:

ALTER TABLE cliente

DELETE primary key

- ▶ Exemplo 3 – Exclusão de constraint de tabela:

ALTER TABLE cliente

DELETE FOREIGN KEY pedido_cliente_fk

ALTER TABLE

- ▶ Trocar nome de elementos

- ▶ Alteração de nome de tabela

- ```
ALTER TABLE tabela
```

- ```
RENAME nome-tabela
```

- ▶ Exemplo:

- ```
ALTER TABLE cliente
```

- ```
RENAME cli
```

- ▶ Alteração de nome de coluna

- ```
ALTER TABLE tabela
```

- ```
RENAME coluna-velha TO coluna-nova
```

- ▶ Exemplo:

- ```
ALTER TABLE cliente
```

- ```
RENAME nome_cliente TO nmcliente
```

Criação de índice

- ▶ O índice serve para prover um acesso rápido a linhas das tabelas. Por meio deles é possível unir uma ou mais colunas por onde o acesso é mais frequente.
- ▶ Por exemplo: quando temos uma tabela de pessoas, normalmente queremos fazer buscas em ordem alfabética. Como o nome não é uma boa chave primária para a tabela*, então criamos um índice para o nome da pessoa.

*Por que o nome não é uma boa chave primária para tabelas?

CREATE INDEX

**CREATE [UNIQUE] INDEX nome ON tabela
(coluna [,coluna, ...] [ASC | DESC])**

- ▶ Onde
 - ▶ UNIQUE: identifica que esse índice não permite repetição de conteúdo na chave (lista de colunas). Se o índice não for especificado, admitirá repetição.
 - ▶ Nome – nome do objeto que será criado.
 - ▶ Tabela – nome da tabela que contém as colunas
 - ▶ Coluna – lista de colunas que compõe a chave de indexação
-

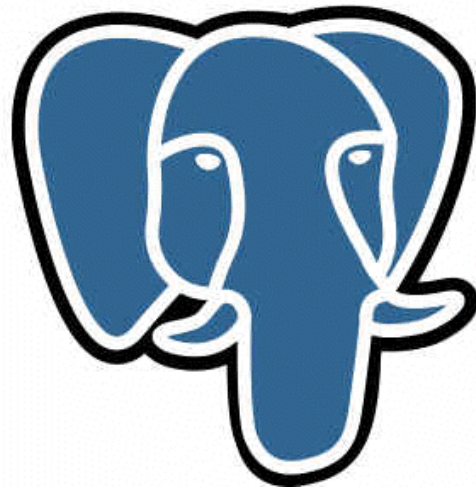
CREATE INDEX

- ▶ ASC – determina que a ordem de indexação é ascendente (opção-padrão, caso não seja definido DESC)
 - ▶ DESC – determina que a ordem de indexação é descendente
 - ▶ Exemplos:
 - ▶ CREATE INDEX xAutor ON Autor (NMAUTOR)
 - ▶ CREATE INDEX xPedido ON Pedido (CDPEDIDO DESC)
 - ▶ CREATE INDEX xCidade ON cidade (SGESTADO, CDCIDADE)
 - ▶ CREATE UNIQUE INDEX xcliente ON Cliente (CDCLIENTE)
-

DROP INDEX

- ▶ Para excluir um índice, utilize o comando:
 - ▶ DROP INDEX nome
 - ▶ Exemplo:
 - ▶ DROP INDEX xAutor
-

PostgreSQL



► Práticas

Práticas

- ▶ Prompt do linux
 - ▶ Interface desktop (PgAdmin)
-

Prompt do linux

- ▶ Para entrar:

- ▶ ~\$ sudo -u postgres psql postgres

- ▶ Para sair:

- ▶ \q



SGBD – Cliente PostgreSQL – Resumo

- Criar um banco de dados
 - `create database nome_bd;`
 - Usar (**c**onectar) um banco de dados como padrão
 - `\c nome_bd;`
 - Apagar um banco de dados
 - `drop database nome_bd;`
-

SGBD – Cliente PostgreSQL – Resumo

- Obtendo informações sobre o BD...
 - Mostrar todos os bancos de dados
 - \l
 - Mostrar todas as tabelas de um BD
 - \d;
 - Mostrar detalhes de uma tabela
 - \d nome_tabela
 - Mostrar esquemas
 - \dn
 - Ajuda
 - \?
 - Ajuda sobre comandos do cliente (terminal)
 - \h
 - Ajuda sobre comandos SQL
 - Sair (quit) do cliente (terminal)
 - \q
-

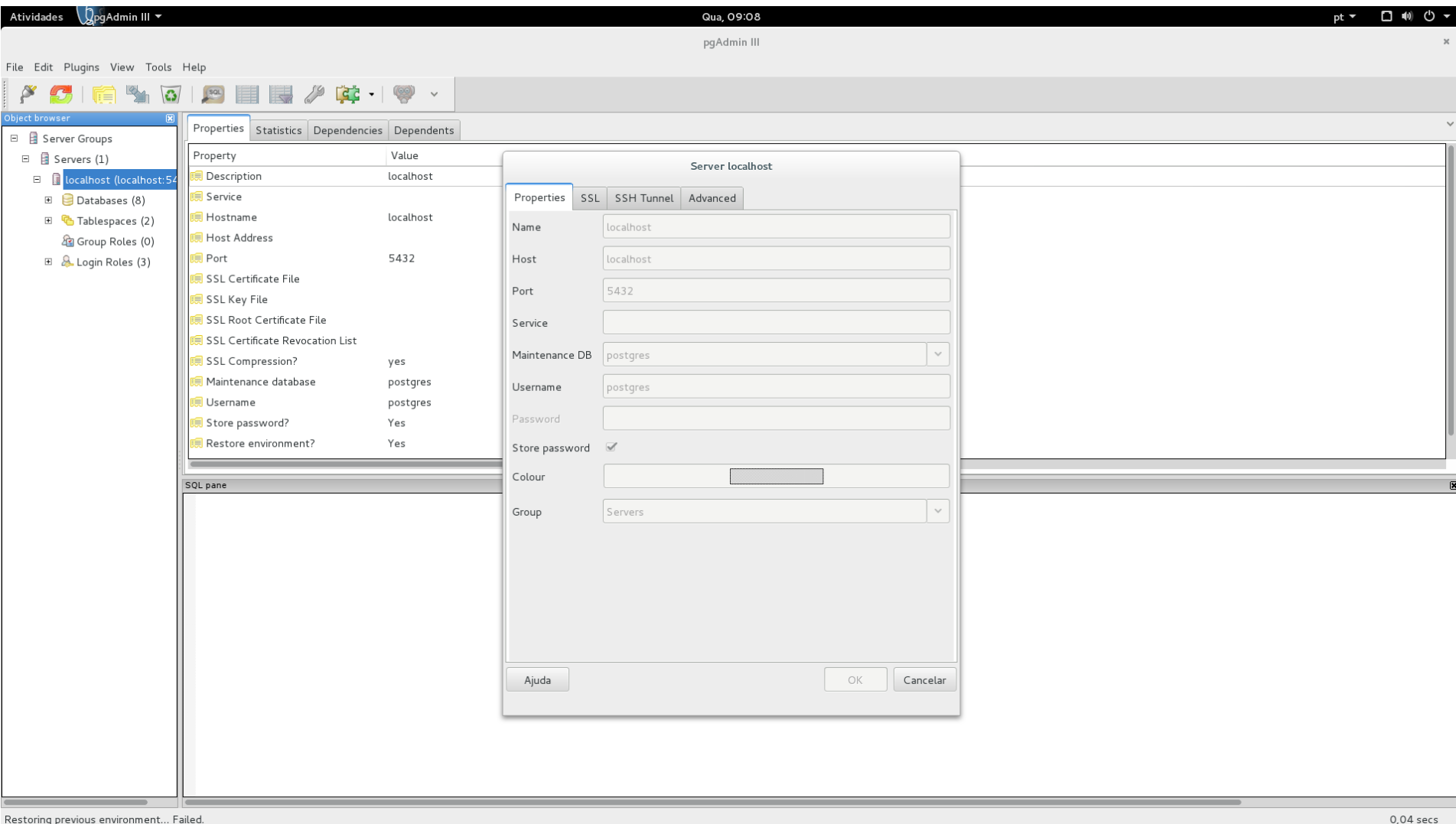
Usando o Pgadmin no linux

- ▶ **Para instalar:**

- ▶ `# apt-get install postgresql pgadmin3`



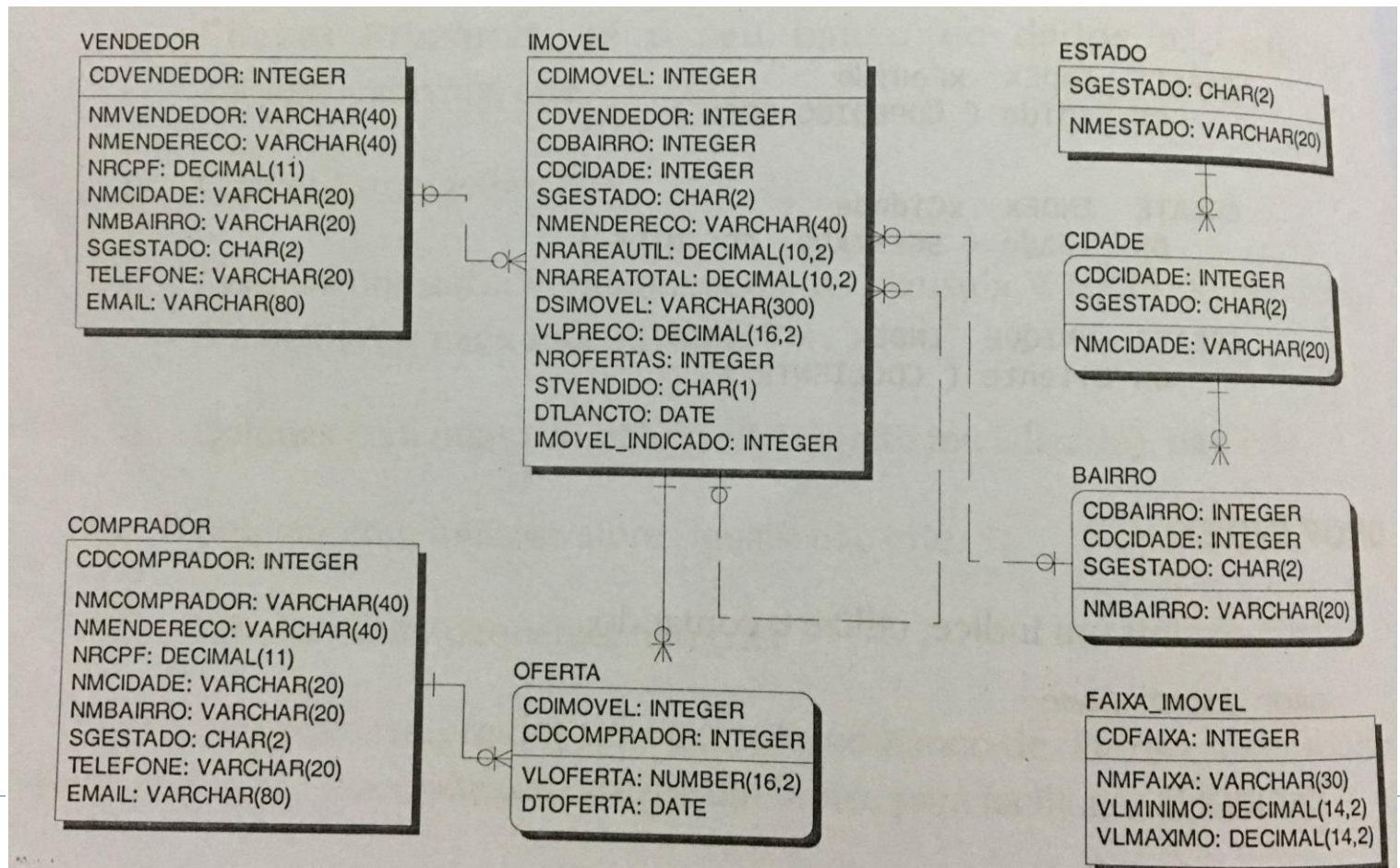
Usando o Pgadmin no linux



Exercícios

Exercício 1 – Database RedeImoveis

- ▶ Crie as tabelas para o modelo de dados abaixo, incluindo os relacionamentos e toda a integridade referencial:



Exercício 2 – Database ImovelNet

- ▶ Crie índices para VENDEDOR.NMVENDEDOR e COMPRADOR.NMCOMPRADOR.

Exercício 3 – Database ImovelNet

- ▶ Crie índices para OFERTA.CDIMOVEL (ascendente) e OFERTA.VLOFERTA (descendente).

SQL - DML

- ▶ Data Manipulation Language (DML): permite a manipulação dos dados armazenados no banco de dados.
 - ▶ Comandos DML:
 - ▶ INSERT
 - ▶ DELETE
 - ▶ UPDATE
-

O comando INSERT

- ▶ Especificar o nome da relação e uma lista de valores para a tupla:
 - ▶ Insert into professores values (default, '111', 'Marilia', '2014--07--31', 'Russas', 'CE');
 - ▶ Insert into professores (cpf, nome, data_cadastro, cidade) values ('122', 'Marcos', '2015--02--12', 'Russas');
 - ▶ Insert into professores (uf, nome, cidade) values ('CE', 'Markos', 'Russas'), ('CE', 'Livia', 'Quixada'), ('CE', 'Ronan', 'Fortaleza');
-

O comando DELETE

- ▶ Remove tuplas de uma relação
- ▶ Inclui uma cláusula WHERE para selecionar as tuplas a serem excluídas

U4A:	DELETE FROM	FUNCIONARIO
	WHERE	Unome='Braga';
U4B:	DELETE FROM	FUNCIONARIO
	WHERE	Cpf='12345678966';
U4C:	DELETE FROM	FUNCIONARIO
	WHERE	Dnr=5;
U4D:	DELETE FROM	FUNCIONARIO;

O comando UPDATE

- ▶ Modifica valores de atributo de uma ou mais tuplas selecionadas
- ▶ Cláusula **SET** adicional no comando UPDATE
 - ▶ Especifica os atributos a serem modificados e seus novos valores

```
U5: UPDATE   PROJETO
      SET     Projlocal = 'Santo André', Dnum
              = 5
      WHERE   Projnumero=10;
```

Exercícios

Exercício 4 – Database ImovelNet

- Inclua linhas na tabela ESTADO:

	sgestado character(2)	nmestado character varying(20)
1	SP	São Paulo
2	RJ	Rio de Janeiro
3	CE	Fortaleza

Exercício 5 – Database ImovelNet

- Inclua linhas na tabela CIDADE:

	<code>cdcidade</code> <code>integer</code>	<code>nmcidade</code> <code>character varying(20)</code>	<code>sgestado</code> <code>character(2)</code>
1	1	São Paulo	SP
2	2	Rio de Janeiro	RJ
3	3	Fortaleza	CE

Exercício 6 – Database ImovelNet

- Inclua linhas na tabela BAIRRO:

Output pane					
Data Output		Explain	Messages	History	
	cdbairro	nmbairro	cdcidade	sgestado	
	integer	character varying(20)	integer	character(2)	
1	1	Jardins	1	SP	
2	2	Morumbi	1	SP	
3	3	Aeroporto	1	SP	

Exercício 7 – Database ImovelNet

► Inclua linhas na tabela VENDEDOR:

Data Output

Explain

Messages

History

	cdvendedor integer	nmvendedor character varying(40)	nmendereco character varying(40)	nrcpf numeric(11,0)	nmcidade character varying(20)	nmbairro character varying(20)	sgestado character(2)	telefone character varying(20)	email character varying(80)
1	1	Maria da Silva	Rua do Grito,45						msilva@novatec.com.br
2	2	Marcos Andrade	Av. da Saudade, 325						mandrade@novatec.com.br
3	3	André Cardoso	Av. Brasil,401						acardoso@novatec.com.br
4	4	Tatiana Souza	Rua do Imperador,778						tsouza@novatec.com.br
5	5	Luiz Gonzaga	Rua da Paz,145						gonzagao@novatec.com.br
6	6	Renato Russo	Rua do Sono,15						rrusso@novatec.com.br
7	7	Jorge Benjor	Rua da música,1025						jben@novatec.com.br
8	8	Gal Costa	Rua do Itapoã,12						gal@novatec.com.br

OK.



Exercício 8 – Database ImovelNet

- Inclua linhas na tabela IMOVE:

	cdimovel integer	cdvendedor integer	cdbairro integer	cdcidade integer	sgestado character(2)	nmendereco character varying(40)	nrareutil numeric(10,2)	nrareatotal numeric(10,2)	dsimovel character varying(300)	vlpreco numeric(16,2)	nroferas integer	stvendido character(1)	dtlancto date	imovelIndicado integer
1	1	1	1	1	SP	Al Tiete,3304 ap101	250.00	400.00		180000.00			2001-11-05	

Exercício 9 – Database ImovelNet

► Inclua linhas na tabela COMPRADOR:

Data Output	Explain	Messages	History						
	cdcomprador integer	nmcomprador character varying(40)	nmendereco character varying(40)	nrcpf numeric(11,0)	nmcidade character varying(20)	nmbairro character varying(20)	sgestado character(2)	telefone character varying(20)	email character varying(80)
1	1	Emmanuel Antunes	R Saraiva, 452						eantunes@ufc.br
2	2	Joana Pereira	Av. Portugal, 52						jpereira@novatec.br
3	3	Ronaldo Campelo	R Saraiva, 790						rcampelo@novatec.br
4	4	Manfred Augusto	R Saraiva, 351						manfred@ufc.br
5	5	Zé Pereira	R das Floras, 152						zezim@ufc.br

Exercício 10 – Database ImovelNet

- Inclua linhas na tabela OFERTA:

	cdcomprador integer	cdimovel integer	vloferta numeric(16,2)	dtoferta date
1	1	1	170000.00	2002-01-10

Exercício 11 – Database ImovelNet

- ▶ Aumente o preço de venda dos imóveis em 10%.

areatotal	dsimovel	vlpreco	nrofert	stvendido	dtlancto	imovelid
numeric(10,2)	character varying(300)	numeric(16,2)	integer	character(1)	date	integer
400.00		180000.00			2001-11-05	

total	dsimovel	vlpreco	nrofert	stvendido	dtlancto	imovelid
ic(10,2)	character varying(300)	numeric(16,2)	integer	character(1)	date	integ
400.00		198000.00			2001-11-05	

Exercício 12 – Database ImovelNet

- ▶ Abaixar o preço de venda dos imóveis do vendedor 1 em 5%.



Exercício 13 – Database ImovelNet

- ▶ Aumente em 5% o valor das ofertas do comprador 2.



Exercício 14 – Database ImovelNet

- ▶ Altere o endereço do Comprador 3 para R.ANANÁS, 45 e o estado para RJ.



Exercício 15 – Database ImovelNet

- ▶ Altere a oferta do comprador 2 no imóvel 4 para 101.000.



Exercício 16 – Database ImovelNet

- ▶ Exclua a oferta do comprador 3 no imóvel 1.



Exercício 17 – Database ImovelNet

- ▶ Exclua a cidade 3 do Estado SP.



Bibliografia Utilizada nesta aula

- ▶ OLIVEIRA, C.H. SQL: Curso prático. Novatec, 2002. ISBN: 9788575220245 .
- ▶ ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6 ed. Pearson/Addison-Wesley, 2011. ISBN: 9788579360855