



## • Módulo Um

- Criação de bancos de dados
- Criação de esquemas, tabelas, visões, índices e clusters.
- Integridade referencial e integridade de domínios.
- Funções de agregação, matemáticas, data e hora, manipulação de strings, formatação e endereçamento de rede.
- Operações especiais entre tabelas - união, intersecção e diferença.
- Visões, funções armazenadas com e sem retorno e triggers.
- Conexões nativas, ODBC e JDBC.



### Modelo Relacional:

- Consolidado no mercado
  - SGBDs Open Source - PostgreSQL, MySQL, Firebird ...
  - SGBDs proprietários - Oracle, Sybase, DB2, MS SQLServer, Interbase ...
    - Relações
    - Integridade referencial
    - Integridade de domínio



### Modelo Relacional:

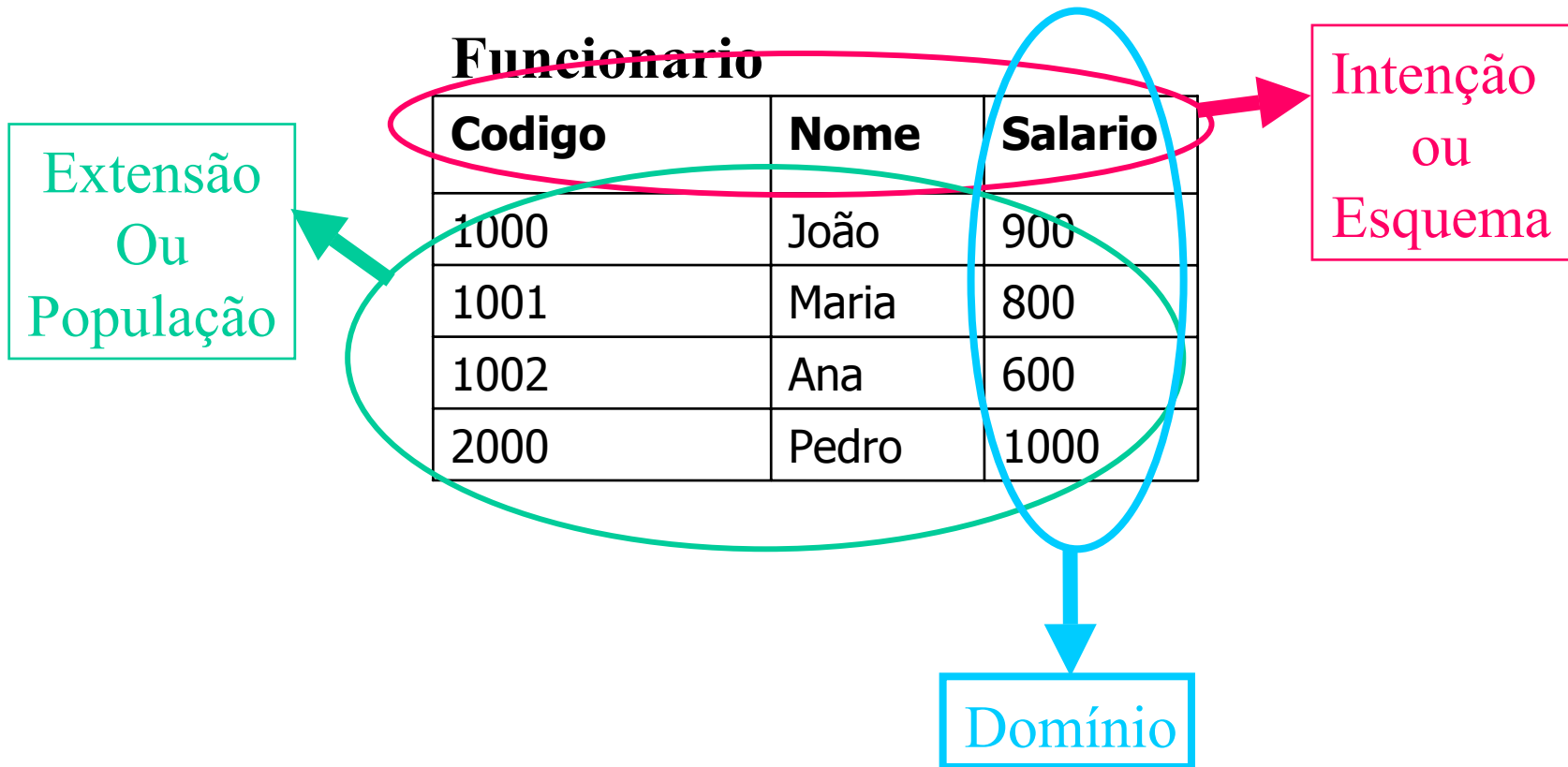
- Relação  $\Leftrightarrow$  Tabela
- Tabela - armazena em suas colunas os valores relativos aos atributos das entidades modeladas.
- Cada domínio é formado por um conjunto de valores possíveis e um tipo de dado para determinado atributo (coluna).
  - Salário
    - Entre 350,00 e 10.000,00 unidades de moeda
    - Tipo - numeric(8,2) ou double



## PostgreSQL - Funcionalidades Básicas

### Modelo Relacional:

- Relação  $\Leftrightarrow$  Tabela





# PostgreSQL - Funcionalidades Básicas

## Modelo Relacional:

- Integridade referencial
  - implementa os relacionamentos entre entidades (tabelas)

### Fornecimento

cod_fornec varchar	cod_peca varchar	cod_proj varchar	quantidade numeric
F1	P1	J1	200
F1	P1	J4	700
F2	P3	J1	400
F2	P3	J2	200
F2	P3	J3	200
F2	P3	J4	500
F2	P3	J5	600
F2	P3	J6	400
F2	P3	J7	800
F2	P5	J2	100
F3	P3	J1	200
F3	P4	J2	500
F4	P1	J4	100

### Peça

cod_peca [PK] varchar	nome varchar	cor varchar	peso numeric	cidade varchar
P1	NULT	VERMELHO	12.00	LONDRES
P2	BOLT	VERDE	17.00	PARIS
P3	SCREW	AZUL	17.00	ROMA
P4	SCREW	VERMELHO	14.00	LONDRES
P5	CAM	AZUL	12.00	PARIS
P6	COG	VERMELHO	19.00	LONDRES

Chave primária

Somente são permitidas as entradas para códigos de peças em Fornecimento se os valores existirem em Peça

Chave estrangeira



## PostgreSQL - Funcionalidades Básicas

### Modelo Relacional:

- Integridade de domínio

#### Fornecimento

cod_fornec varchar	cod_peca varchar	cod_proj varchar	quantidade numeric
F1	P1	J1	200
F1	P1	J4	700
F2	P3	J1	400
F2	P3	J2	200
F2	P3	J3	200
F2	P3	J4	500
F2	P3	J5	600
F2	P3	J6	400
F2	P3	J7	800
F2	P5	J2	100
F3	P3	J1	200
F3	P4	J2	500
F4	P1	J4	100

Não são admitidos  
valores nulos ou  
valores negativos  
Tipo - numeric



## PostgreSQL - Funcionalidades Básicas

- Psql – interface de linha de comando, terminal interativo

psql -U postgres

psql dbmapas -U postgres

psql -h localhost -d dbmapas -U postgres

psql -h localhost -d dbmapas -U postgres -p 5432

-U – usuário

-u – obriga a solicitação de usuário e senha

-h host

-d database

-p port



## PostgreSQL - Funcionalidades Básicas

- **Psql**

```
CREATE TABLESPACE dataspace OWNER postgres  
LOCATION 'C:/Arquivos deProgramas/PostgreSQL/8.1/data/dados';
```

```
CREATE DATABASE banco OWNER postgres  
TEMPLATE template1 TABLESPACE dataspace;
```

- **SO**

```
createdb -D dataspace -O postgres -T template1 banco -U postgres  
psql banco -U postgres
```





## PostgreSQL - Funcionalidades Básicas

- Psql

```
Create table cliente (  
codcli numeric(4) primary key not null,  
nome varchar(40),  
endereco varchar(100),  
idade date);
```

\q

```
drop database banco;
```



## PostgreSQL - Funcionalidades Básicas

- **SO**

```
pg_dump -f dbmapas.sql -C -d -U postgres banco
pg_dump -help
```

parâmetros para o pg\_dump:

- f : arquivo de saída
- C : comandos para a criação do banco
- d : copia dados utilizando comandos INSERT
- U : usuário
- a : copia somente os dados, sem o esquema
- ...



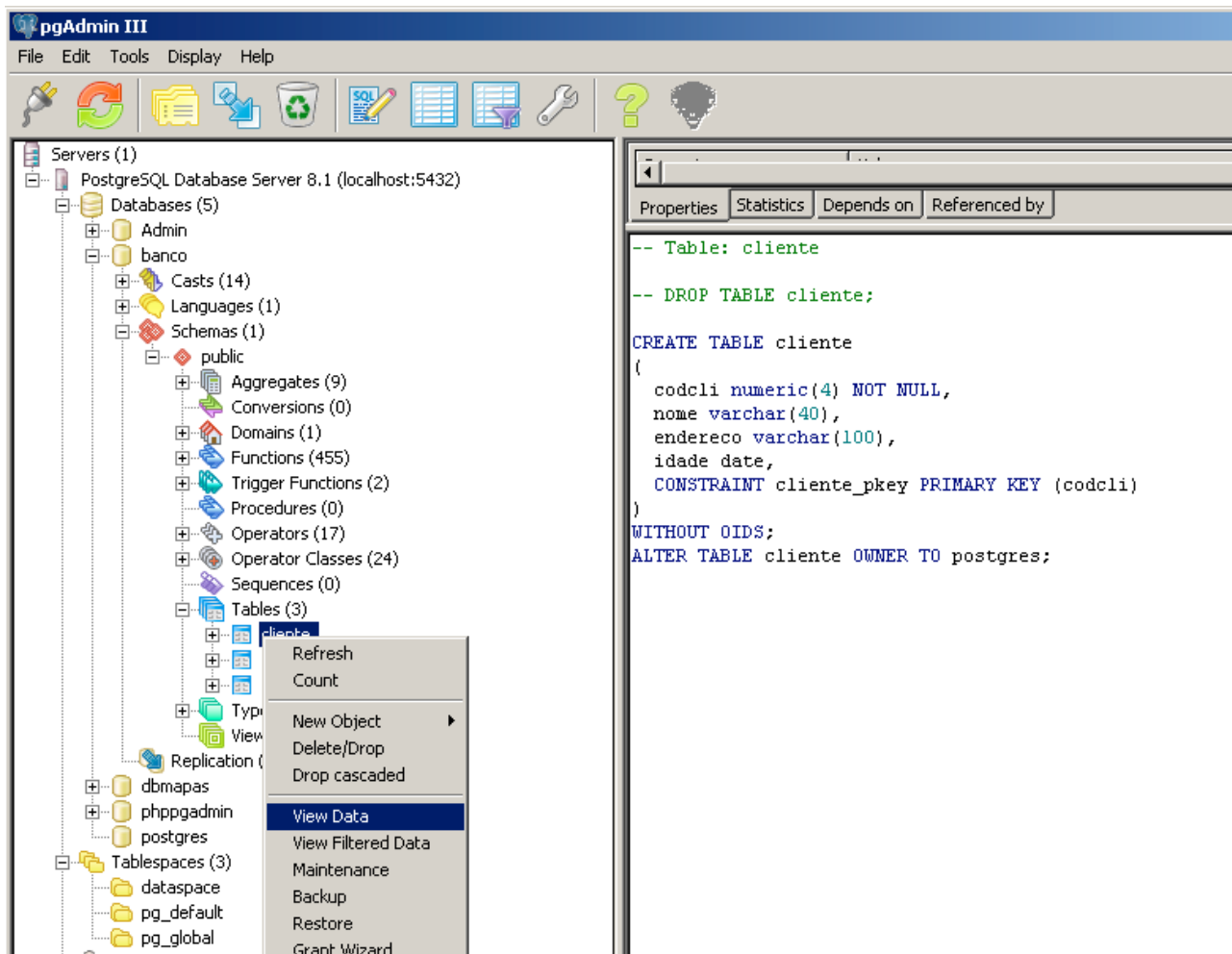
## PostgreSQL - Funcionalidades Básicas

- PgAdmin III
- O PgAdmin III é uma interface gráfica do PostgreSQL, através da qual podemos criar os mais diversos tipos de objetos no banco de dados, como tablespaces, bancos de dados, esquemas, tabelas, visões, roles ...
- Além da facilidade para a criação dos objetos, é possível usar ferramentas para edição e execução de comandos SQL, visualização dos dados e propriedades dos objetos.



# PostgreSQL - Funcionalidades Básicas

- PgAdmin III





- **PgAdmin III**
  - Criação de objetos
    - Databases
    - Schemas
    - Tabelas
    - Índices
      - Integridade referencial
    - Tabelas de sistema



## PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**
  - Roda no browser
  - É possível criar os seguintes objetos:
    - Databases
    - Schemas
    - Tabelas
    - Visões
    - Seqüências
    - Funções
    - Domínios



# PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**

phpPgGIS :: Administration Tool for PostGIS :: - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:86/phpPgGIS-4.0.1-3/

Últimas notícias Inglês -> Português Dúvidas Firefox

**phpPg GIS**

PostgreSQL 8.1.3 rodando em localhost:5432 -- Você está logado como usuário "postgres", 16th Nov, 2006 10:38PM

phpPgGIS :: Administration Tool for PostGIS :: PostgreSQL?

Banco de dados?	Usuários?	Grupos?	Tablespaces?	Exportar
<b>Banco de dados</b>				

Banco de dados	Proprietário	Codificação	Tablespace	Ações			Comentário
dbmapas	postgres	SQL_ASCII	pg_default	Deletar	Privilégios	Alterar	
phppgadmin	postgres	SQL_ASCII	pg_default	Deletar	Privilégios	Alterar	
postgres	postgres	SQL_ASCII	pg_default	Deletar	Privilégios	Alterar	
teste	monica	SQL_ASCII	dataspace	Deletar	Privilégios	Alterar	

**Criar banco de dados**

**Servers**

- PostgreSQL
  - dbmapas
    - Esquemas
      - public
        - Tabelas
        - Visualizações
        - Sequências
        - Funções
        - Domains
      - sqlj
    - phppgadmin
    - postgres
      - Esquemas
    - teste



## PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**
  - Roles
  - Usuários
  - Tablespaces
  - Backup de dados e/ou esquema





# PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**

Visualização  
de mapas



# PostgreSQL - Funcionalidades Básicas

- **Psql**

Criando um banco de dados:

```
create database banco owner postgres template template1  
tablespace dataspace;
```

- **Parâmetros:**

Owner: proprietário do banco de dados

Template: modelo de banco de dados

Encoding: método de codificação multibyte

Tablespace: área para criação do banco de dados

Connecton limit: número limite de conexões



## PostgreSQL - Funcionalidades Básicas

- **Psql**

- **Criando um banco de dados:**

`create database banco owner postgres template template1  
tablespace dataspace;`

- **Parâmetros:**

**Owner:** proprietário do banco de dados

**Template:** modelo de banco de dados

**Encoding:** método de codificação multibyte

**Tablespace:** área para criação do banco de dados

**Connecton limit:** número limite de conexões

`psql -d banco -U postgres`



- **Psql**

- Eliminando um banco de dados:

`drop database banco;`

- Criação de esquemas:

`create schema papelaria;`

Um esquema é formado por um conjunto de objetos de um banco de dados. O usual é atribuir esquemas a usuários específicos. Por exemplo, é possível ter dois usuários com atribuições diferentes sobre os objetos do mesmo banco de dados, cada qual com o seu esquema próprio.

`create schema authorization usuario;`

Cria um esquema com o nome *usuario* e autorização para o usuário – o usuário já deve ter sido criado.



Caso o esquema não seja informado a tabela é criada no esquema "public"

- **Psql**

- Criação de tabelas:

```
create table usuario.aluno (  
  cpf numeric (11) not null primary key,  
  rg varchar(20),  
  nome varchar(60),  
  endereco varchar(80),  
  cidade varchar(20),  
  telefone numeric(10),  
  datanasc date,  
  peso numeric(4,2),  
  altura numeric(3,2));
```

Definição de  
constraint -  
chave primária



Caso o esquema não seja informado a tabela é criada no esquema "public"

- **Psql**

- Criação de tabelas:

```
create table usuario.modalidade (  
codmod numeric(4) not null primary key,  
descmod varchar(20));
```

```
create table usuario.turma (  
codturma numeric(4) not null primary key,  
codmod numeric(4),  
sala varchar(20),  
constraint fk_modalidade foreign key (codmod)  
references usuario.modalidade);
```

Definição de  
constraint -  
chave  
estrangeira



## PostgreSQL - Funcionalidades Básicas

- **Psql**

- Criação de índices:

Índices são essenciais  
aumento de performance  
no banco de dados

```
create index indsala on usuario.turma(sala);
```

```
create index indlocal on usuario.local(pais,estado,cidade);
```

É possível indexar uma ou mais colunas de uma tabela. Os índices devem ser determinados a partir de aplicativos de usuários, com base nas consultas mais freqüentes realizadas sobre a tabela.



# PostgreSQL - Funcionalidades Básicas

- **Psql**

- Tabelas de sistema ⇔ Catálogo
- pg\_tables
- pg\_views
- pg\_user
- pg\_roles
- pg\_index
- pg\_type
- pg\_tablespace ... entre outras.

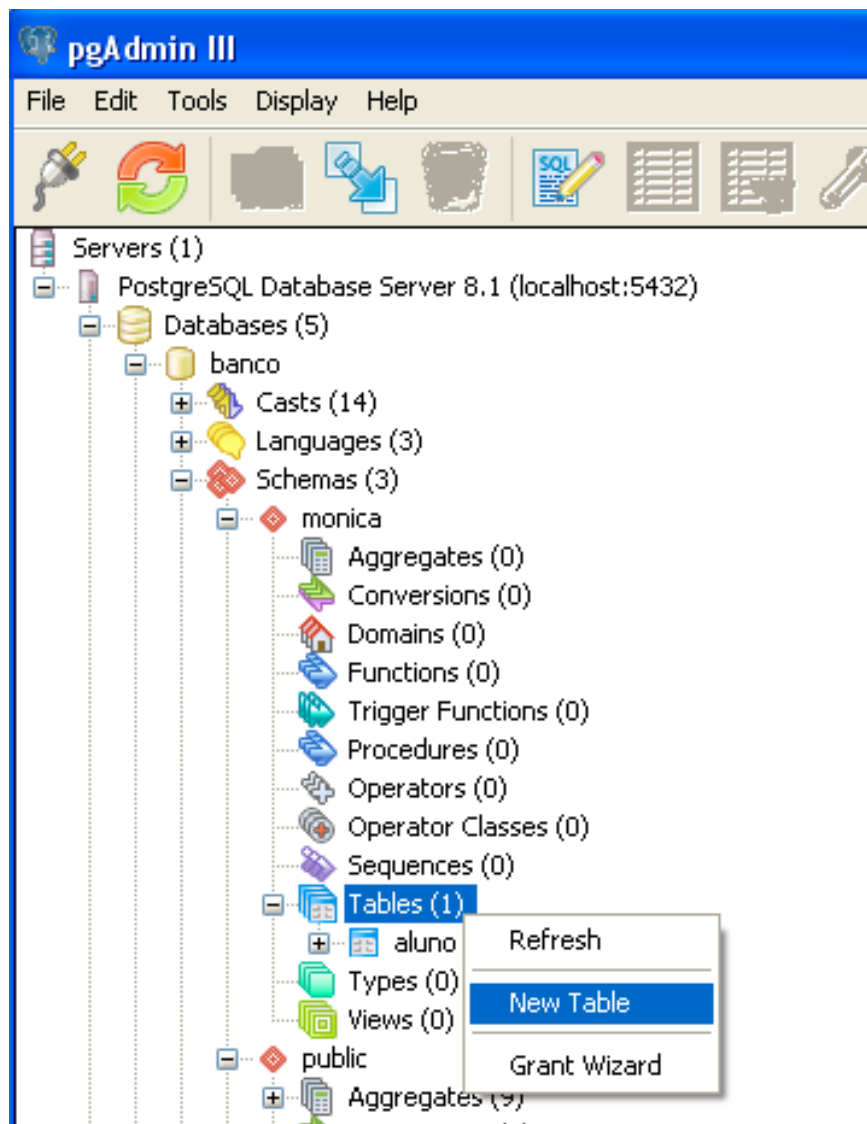
`select * from pg_tables;`





# PostgreSQL - Funcionalidades Básicas

- **pgAdminIII**  
Criação de tabelas





# PostgreSQL - Funcionalidades Básicas

- **pgAdminIII**  
Criação de tabelas

**New Table**

Properties Columns Constraints Privileges SQL

Name: modalidade

OID:

Owner: monica

Tablespace: dataspace

Has OIDs: ☐

Inherits from tables

Remove

Add: monica.modalidade

Comment:

Use replication:

Help OK Cancel



## PostgreSQL - Funcionalidades Básicas

### pgAdminIII

Criação de tabelas

Definição de colunas

The screenshot shows the 'New Column' dialog box in pgAdmin III. The 'Properties' tab is active, displaying the following fields and values:

- Name: codmod
- Data type: numeric (dropdown menu)
- Length: 4
- Precision: 2
- Default value: (empty text box)
- Sequence: (empty dropdown menu)
- Not NULL: ☒
- Statistics: (empty text box)
- Comment: (empty text area)
- Use replication: (empty dropdown menu)

At the bottom of the dialog are three buttons: Help, OK, and Cancel.

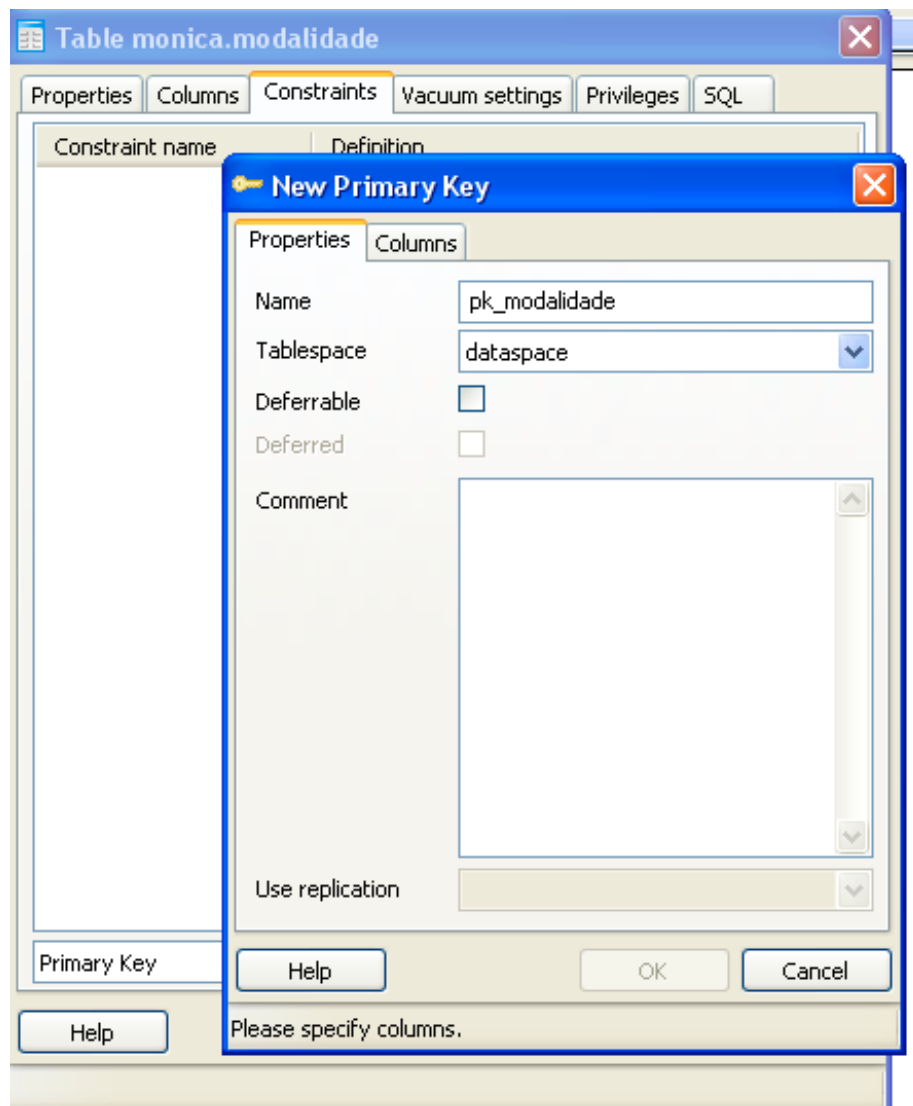


## PostgreSQL - Funcionalidades Básicas

### pgAdminIII

Criação de tabelas

Definição de chaves

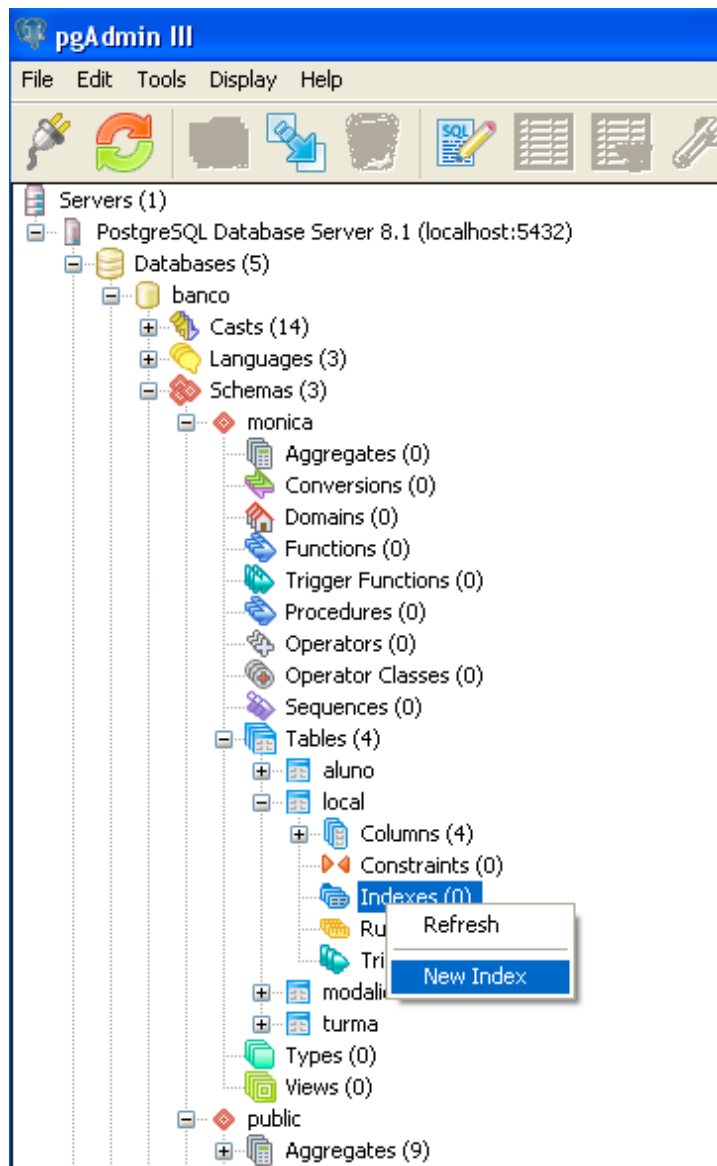




# PostgreSQL - Funcionalidades Básicas

## pgAdminIII

### Criação de índices





## pgAdminIII

### Criação de índices

The image shows the 'New Index' dialog box in pgAdmin III. The dialog has three tabs: 'Properties', 'Columns', and 'SQL'. The 'Properties' tab is selected. It contains the following fields and options:

- Name:** indlocal
- Tablespace:** dataspace (dropdown menu)
- Access method:** btree (dropdown menu)
- Unique:** ☐
- Clustered:** ☐
- Constraint:** (empty text area with scrollbars)
- Comment:** (empty text area with scrollbars)
- Use replication:** (dropdown menu)

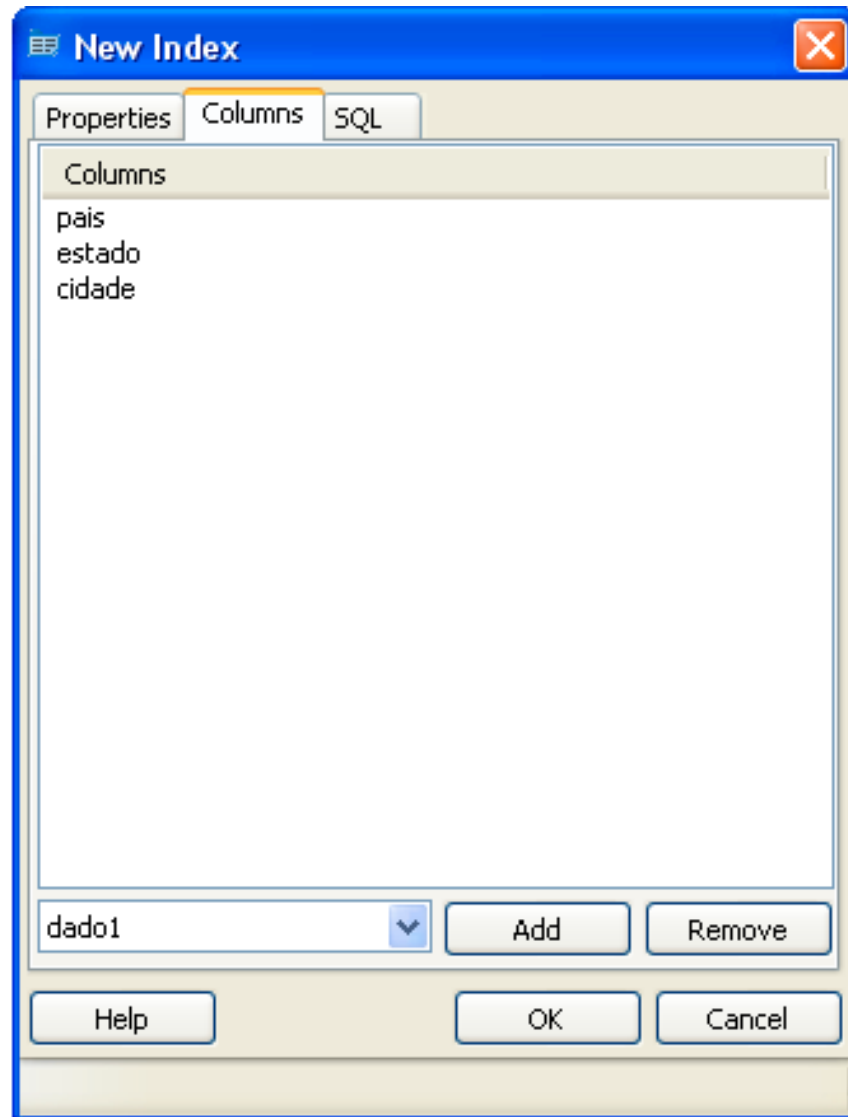
At the bottom of the dialog are three buttons: 'Help', 'OK', and 'Cancel'. Below the dialog box, a status bar displays the message: 'Please specify columns.'



## PostgreSQL - Funcionalidades Básicas

### pgAdminIII

#### Criação de índices



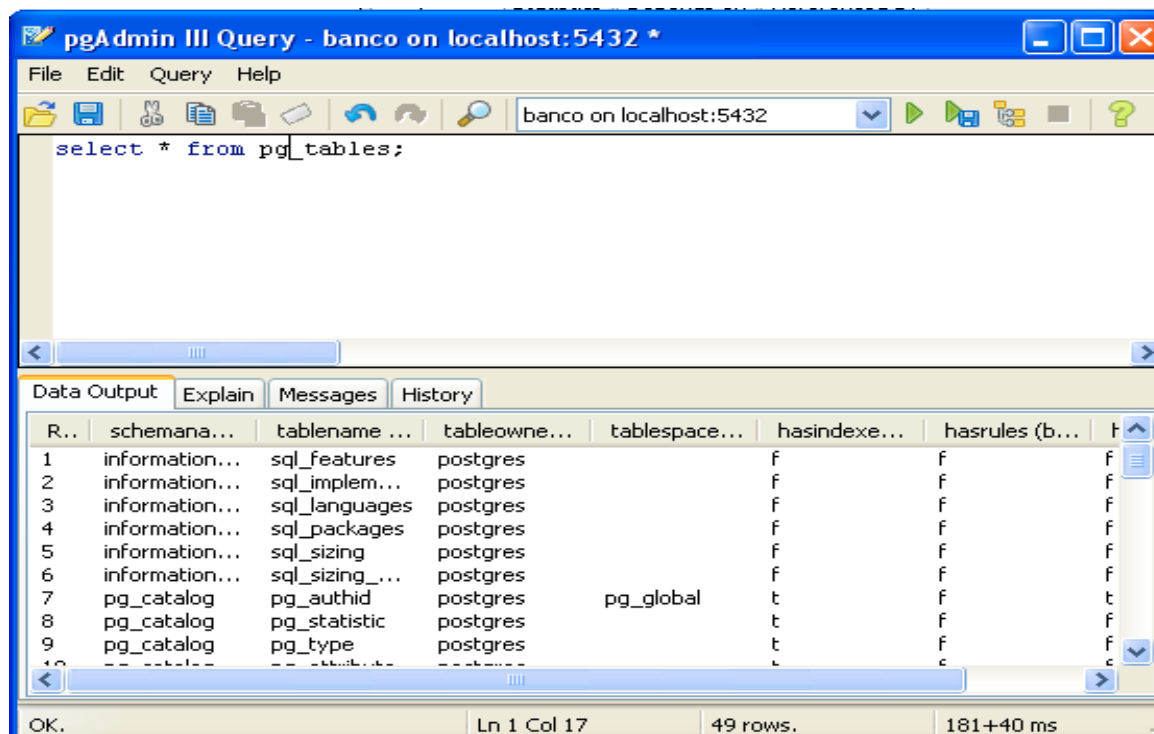


# PostgreSQL - Funcionalidades Básicas

- **pgAdminIII**

- **Tabelas de sistemas**

não é possível acessar o catálogo do PostgreSQL, diretamente pelo pgAdminIII.







# PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**

## Criação de tabelas

phpPgGIS : Administration Tool for PostGIS : - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:86/phpPgGIS-4.0.1-3/

Últimas notícias Inglês -> Português Dúvidas Firefox

**phpPg GIS**

PostgreSQL 8.1.3 rodando em localhost:5432 -- Você está logado como usuário "postgres", 18th Nov, 2006 9:36PM

phpPgGIS : Administration Tool for PostGIS : PostgreSQL: banco: public:

**Tabelas** Visualizações Sequências Funções Domains

Tabela	Proprietário	Tablespace	Estimated row count	Ações				
aluno	postgres		0	Navegar	Selecionar	Inserir	Vazio	Deletar
funcionario	postgres		0	Navegar	Selecionar	Inserir	Vazio	Deletar
geometry_columns	postgres		0	Navegar	Selecionar	Inserir	Vazio	Deletar
spatial_ref_sys	postgres		2671	Navegar	Selecionar	Inserir	Vazio	Deletar

**Criar tabela**



# PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**

## Criação de tabelas

phpPgGIS :: Administration Tool for PostGIS :: - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:86/phpPgGIS-4.0.1-3/

Últimas notícias Inglês -> Português Dúvidas Firefox

**phpPg GIS**

Servers

- PostgreSQL
  - banco
  - dbmapas
  - phppgadmin
  - postgres
  - teste

PostgreSQL 8.1.3 rodando em localhost:5432 -- Você está logado como usuário "postgres" 9:38PM

phpPgGIS :: Administration Tool for PostGIS :: PostgreSQL?: banco?: public?:

**Criar tabela?**

Nome: funcionario

Number of columns: 3

Options: ☒ WITHOUT OIDS

Tablespace: dataspace

Comentário: fno - primary  
cod\_cargo - foreign

Próximo Cancelar

Informe as propriedades da tabela



# PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**

## Criação de tabelas

SERIAL- auto incremento

Character varying - varchar

phpPgGIS :: Administration Tool for PostGIS :: - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:86/phpPgGIS-4.0.1-3/

Últimas notícias Inglês -> Português Dúvidas Firefox

PostgreSQL 8.1.3 rodando em localhost:5432 -- Você está logado como usuário "postgres", 18th Nov, 2006 9:42PM

phpPgGIS :: Administration Tool for PostGIS :: PostgreSQL banco?: public?

**Criar tabela?**

Column	Tipo	Extensão	Não Nulo	Chave única	Chave primária	Padrão
1. fno	SERIAL		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2. fnome	character varying	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3. cod_cargo	numeric	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	



# PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**

## Criação de índices

phpPgGIS :: Administration Tool for PostGIS :: - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:86/phpPgGIS-4.0.1-3/

Últimas notícias Inglês -> Português Dúvidas Firefox

**phpPg GIS**

Servers

- PostgreSQL
  - banco
    - Esquemas
      - monica
      - public
        - Tabelas
          - aluno
          - funcionario
          - geometry\_columns
          - spatial\_ref\_sys
        - Visualizações
        - {a\_n} Sequências
        - f\_x Funções
        - Domínios

PostgreSQL 8.1.3 rodando em localhost:5432 -- Você está logado como usuário "postgres", 18th Nov, 2006 9:52PM

phpPgGIS :: Administration Tool for PostGIS :: PostgreSQL?: banco?: public?: funcionario?:

Colunas **Índices?** Restrição? Gatilhos? Regras? Info Privilegios?

Nome	Definição	Restrição	Clustered?
pf_func	CREATE UNIQUE INDEX pf_func ON funcionario USING btree (fno)	Chave primária	Não

**Criar índice**



# PostgreSQL - Funcionalidades Básicas

- **phpPgGIS**

## Criação de índices

Defina as propriedades do índice

phpPgGIS : Administration Tool for PostGIS : Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:86/phpPgGIS-4.0.1-3/

Últimas notícias Inglês -> Português Dúvidas Firefox

phpPgGIS

PostgreSQL 8.1.3 rodando em localhost:5432 -- Você está logado como usuário "postgres", 18th Nov, 2006 9:58PM

SQL | Find | Deslogar

phpPgGIS : Administration Tool for PostGIS : PostgreSQL ?; banco ?; public ?; funcionario ?

Criar índice?

Nome do índice

Colunas na tabela	Colunas no índice
fno	cod_cargo
fnome	

<< >>

Tipo de índice BTREE

Único ☐

Onde { }

Tablespace dataspace

Criar Cancelar



# PostgreSQL - Funcionalidades Básicas

phpPgGIS

Tabelas de sistemas

The screenshot shows the phpPgGIS SQL editor interface. The 'Servidor' dropdown is set to 'PostgreSQL (localhost:5432)' and the 'Banco de dados' dropdown is set to 'banco'. The 'Schema search path' is set to 'public'. The SQL query entered is 'select \* from pg\_tables;'. A purple oval highlights the query and the text 'É PRECISO CONHECER OS NOMES DAS TABELAS DE SISTEMA!!!!|' below it. At the bottom, there are buttons for 'Rodar', 'Explain', 'Explain Analyze', and 'Reiniciar', and a status bar indicating 'Concluído'.

Não é possível acessar o catálogo de sistema diretamente pelo phpPgGIS, assim como no PgAdmin III

The screenshot shows the phpPgGIS search results page. The left sidebar displays a tree view of the database structure, including 'PostgreSQL', 'banco', 'Esquemas', 'monica', 'public', 'Tabelas', 'aluno', 'funcionario', 'geometry\_columns', 'spatial\_ref\_sys', 'Visualizações', 'Sequências', 'Funções', and 'Domains'. The main content area shows the search results for the query 'select \* from pg\_tables;'. The results are displayed in a table with columns: 'username', 'usesysid', 'usecreatedb', 'usesuper', 'usecatupd', 'passwd', 'valuntil', and 'useconfig'. The results show two rows: 'postgres' and 'monica'. Below the table, it indicates '2 Linha(s)' and 'Total runtime: 7.318 ms'. The status bar shows 'SQL executado.' and links for 'Edit SQL' and 'Criar reporte'.

username	usesysid	usecreatedb	usesuper	usecatupd	passwd	valuntil	useconfig
postgres	10	TRUE	TRUE	TRUE	*****	NULL	NULL
monica	50389	FALSE	FALSE	FALSE	*****	NULL	NULL



## PostgreSQL - Funcionalidades Básicas

### Exercícios:

- Crie um novo banco de dados no PostgreSQL, usando como template o "template1".
- A partir do modelo conceitual desenvolvido no exercício da página 27, crie as tabelas, chaves primárias e estrangeiras necessárias. As tabelas podem ser criadas no esquema "public". Não é necessária a criação de um novo esquema.



- **Módulo Dois**
- **Introdução a SQL**
  - DDL, DML e DCL
  - Consultas
  - Operações especiais - união, intersecção e diferença
  - Funções de agregação, matemáticas, data e hora, string, formatação, endereçamento de rede.
  - Visões
  - Funções e procedimentos armazenados
  - Triggers
  - Exercícios





- **Introdução a SQL**
  - SQL - Structure Query Language - Linguagem de consulta estruturada.
  - Desenvolvida para definir, manipular e controlar o acesso a dados de bancos de dados relacionais.
  - Subdivide-se em DDL, DML e DCL.

Documentação disponível em:

<http://www.postgresql.org/docs/>



- **Introdução a SQL**
  - **DDL - Data Definition Language**
    - Através deste conjunto de instruções são criados, alterados e eliminados todos os objetos do banco de dados, como tabelas, visões, índices ...

```
create table cliente (  
  codcli numeric(10) primary key,  
  nomecli varchar(40),  
  email varchar(100));
```

```
create view v_cliente as (  
  select nomecli, email  
  from cliente);
```

```
alter table cliente  
add column cpf numeric(11);
```

```
drop table cliente cascade;
```



- **Introdução a SQL**
  - **DML - Data Manipulation Language**
    - Conjunto de instruções que permitem acessar, inserir, alterar e excluir os dados do banco de dados.

```
Select * from v_cliente;
```

```
Insert into cliente values (
```

```
1, 'Machado de Assis', 'assis@gmail.com');
```

```
Delete from cliente;
```

```
Delete from cliente where  
codcli = 100;
```

```
update cliente set email =  
'machado@gmail.com' where  
codcli = 1;
```



- **Introdução a SQL**
  - **DCL - Data Control Language**
    - Conjunto de instruções usado para controlar o acesso aos dados. Através delas é possível criar usuários, roles, e definir atribuições de sistema ou sobre objetos do banco de dados.

```
create user fulano;
```

```
create role teste;
```

```
grant insert, delete, update, select on cliente to teste;
```

```
grant teste to fulano;
```



- **Introdução a SQL**
    - **Scripts SQL**
    - Os scripts são formados por uma sequência de instruções DDL, DML, DCL.
    - São chamados a partir de linha de comando do sistema operacional
- ou
- São executados através de um editor SQL.



## PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**
  - **Exemplo de script sql:**

-- início script

```
CREATE TABLESPACE dataspace OWNER postgres LOCATION 'C:/Arquivos de  
Programas/PostgreSQL/8.1/data/dados';
```

```
CREATE DATABASE banco OWNER postgres TEMPLATE template1 TABLESPACE dataspace;
```

```
\c banco postgres; (\connect dbname user)
```

```
create schema projetos;
```

```
set search_path to projetos;
```

...



- **Introdução a SQL**
  - **Exemplo de script sql:**

-- continuação script

```
drop table projetos.fornecimento;
```

```
drop table projetos. peca;
```

```
drop table projetos.projeto;
```

```
drop table projetos.fornecedor;
```

```
create table projetos.pecas
```

```
(Cod_Peca varchar(4) not null,
```

```
PNome varchar(15),
```

```
Cor varchar(15),
```

```
Peso numeric(6,2),
```

```
Cidade varchar(20),
```

```
constraint pk_pecas primary key (Cod_Peca));
```



- **Introdução a SQL**
  - **Exemplo de script sql:**

-- continuação script

```
INSERT INTO projetos.projeto VALUES ('J4','CONSOLE','ATENAS');
INSERT INTO projetos.projeto VALUES ('J5','COLLATOR','LONDRES');
INSERT INTO projetos.projeto VALUES ('J6','TERMINAL','OSLO');
INSERT INTO projetos.projeto VALUES ('J7','TAPE','LONDRES');
...
INSERT INTO projetos.fornecimento VALUES ('F1','P1','J1',200);
INSERT INTO projetos.fornecimento VALUES ('F1','P1','J4',700);
INSERT INTO projetos.fornecimento VALUES ('F2','P3','J1',400);
INSERT INTO projetos.fornecimento VALUES ('F2','P3','J2',200);
INSERT INTO projetos.fornecimento VALUES ('F2','P5','J2',100);
```





- **Introdução a SQL**

- **Executando um script:**

- O script dado como exemplo está disponível no material de apoio. Faça o download.
    - Para executar o script pelo psql digite no prompt do SO:

- `psql -f criatab_pos.sql -U postgres`

ou

- abra o editor SQL pelo PgAdminIII e abra o arquivo de script. Execute-o.
      - com o banco já criado, vamos a próxima etapa ...



- **Introdução a SQL**
  - Consultas: retornam dados a partir de uma ou mais tabelas do banco de dados.

```
select * from peca;
```

```
select cod_pecas, pnome from peca;
```

```
select cod_pecas, quantidade from fornecimento;
```

```
select peca.cod_pecas, pnome, quantidade
```

```
from peca, fornecimento
```

```
where peca.cod_pecas = fornecimento.cod_pecas;
```



- **Introdução a SQL**

- **Junção natural: verifica igualdade de chaves primária e estrangeira entre duas tabelas.**

A coluna `cod_peca` existe na tabela `Peca` e também na tabela `Fornecimento`, por isso, é preciso indicar de qual tabela se deseja recuperar o campo.

```
select peca.cod_peca, pnome, quantidade  
from peca, fornecimento  
where  
peca.cod_peca = fornecimento.cod_peca;
```



### • Introdução a SQL

- Sub-consultas: são instruções SQL embutidas dentro de outras instruções. Sua finalidade é recuperar valores de tabelas ou listas de valores, para fins de comparação.

A consulta mais externa compara as cidades da tabela peca com a lista criada a partir da consulta mais interna

```
select pnome, cidade from peca
```

```
where cidade in (
```

```
select distinct cidade from fornecedor);
```

A consulta mais interna retorna uma lista com o nome de todas as cidades de fornecedores.



- **Introdução a SQL**
  - **Comando ALTER:** altera a estrutura dos objetos criados através do comando create ...

`set search_path to projetos;`

`alter table fornecedor add column email varchar(80);`

`alter table peca rename to equipamento;`

`alter table fornecedor drop column email;`

`alter role teste rename to roleproj;`

`alter schema projetos owner to roleproj;`

`alter index indlocal set tablespace pg_default;`



- **Introdução a SQL**
  - **Comando DROP: elimina qualquer objeto criado através do comando create ...**

```
set search_path to projetos;
```

```
drop table fornecimento;
```

```
drop table fornecedor cascade;
```

```
drop index indlocal;
```

```
drop view v_cliente;
```

```
drop schema projetos cascade;
```

```
drop database banco;
```

```
drop tablespace dataspace;
```



- **Introdução a SQL**
  - **Comandos:**
    - **Insert - popula as tabelas**

```
INSERT INTO PECA VALUES ('P2','BOLT','VERDE',17,'PARIS');
```

```
INSERT INTO FORNECEDOR VALUES ('F5','ADAMS',30,'ATENAS');
```

```
INSERT INTO PROJETO VALUES ('J1','SORTER','PARIS');
```

```
INSERT INTO FORNECIMENTO VALUES ('F2','P5','J2',100);
```



- **Introdução a SQL**
  - **Comandos:**
    - **Update** - atualiza os dados das tabelas

```
UPDATE PECA SET PESO = 20 WHERE COD_PECA = 'P1';
```

```
UPDATE FORNECIMENTO SET QUANTIDADE = QUANTIDADE +  
1000;
```

```
UPDATE FORNECEDOR SET STATUS = 30 WHERE FNOME =  
'FULANO';
```





- **Introdução a SQL**

- **Comandos:**

- **Delete** - elimina uma linha ou linhas a partir de um critério estabelecido pelo usuário.

DELETE FROM FORNECIMENTO WHERE DATA < 01/01/2006;

DELETE FROM PROJETO WHERE CIDADE = 'ROMA';

DELETE FROM FORNECEDOR;



- **Introdução a SQL**
  - Operações relacionais:
    - União: retorna todas os valores indicados para as relações envolvidas na operação.

SELECT CIDADE FROM FORNECEDOR

**UNION**

SELECT CIDADE FROM PECA

**UNION**

SELECT CIDADE FROM PROJETO;



- **Introdução a SQL**

- Operações relacionais:

- Intersecção: retorna somente os valores que são comuns as duas relações para os atributos solicitados.

SELECT CIDADE FROM FORNECEDOR

**INTERSECT**

SELECT CIDADE FROM PECA;



- **Introdução a SQL**
  - Operações relacionais:
    - **Diferença:** retorna os valores presentes na primeira relação que não estão na segunda relação. A operação de diferença é a única em que a ordem das relações interfere no resultado.

**SELECT CIDADE FROM FORNECEDOR**

**EXCEPT**

**SELECT CIDADE FROM PECA;**



## PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**
  - Operadores:

Operator/Element	Associativity	Description
.	left	table/column name separator
::	left	PostgreSQL-style typecast
[ ]	left	array element selection
-	right	unary minus
^	left	exponentiation
* / %	left	multiplication, division, modulo
+ -	left	addition, subtraction
IS		IS TRUE, IS FALSE, IS UNKNOWN, IS NULL
ISNULL		test for null
NOTNULL		test for not null
(any other)	left	all other native and user-defined operators
IN		set membership
BETWEEN		range containment
OVERLAPS		time interval overlap
LIKE ILIKE SIMILAR		string pattern matching
< >		less than, greater than
=	right	equality, assignment
NOT	right	logical negation
AND	left	logical conjunction
OR	left	logical disjunction

Fonte: [www.postgresql.org](http://www.postgresql.org)



## PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**

- **Operadores:**

- `select quantidade from fornecimento where quantidade < 300;`
    - `select quantidade from fornecimento where quantidade <> 300;`
    - `select quantidade from fornecimento where quantidade = 300;`
    - `select cidade from peca where cidade = any (select cidade from projeto);`
    - `select cidade from peca where cidade = some (select cidade from projeto);`
    - `select cidade from peca where cidade in (select cidade from projeto);`
    - `select cod_pecas, quantidade, (quantidade*0.1) as porcentual from fornecimento where cod_pecas = 'P1';`
    - `select quantidade from fornecimento where quantidade between 300 and 500;`



# PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**

- **Operadores:**

- `select * from peca where pnome like 'B%';`
  - `select * from peca where pnome ilike 'b%';`
  - `select * from projeto where cidade = 'ATENAS' or cidade = 'OSLO';`
  - `select * from projeto where cidade ilike 'atenas' or cidade ilike 'oslo';`
  - `select cod_pecas from peca where peso notnull;`
  - `select cod_pecas from peca where peso isnull;`
  - `select (numero::integer)*2 from teste;`

O campo numero foi definido como char e transformado em inteiro para fazer a multiplicação



## PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**
  - **Operadores - exercício**
- `create table teste ( id numeric(20),numero char(10));`
- `insert into teste values (1,'1000');`
- `insert into teste values (2,'11/21/2006');`
- `select * from teste;`
- `select (numero::integer)/2 from teste where id =1;`
- `select (numero::date)+30 from teste where id =2;`
- `select (numero::timestamp) from teste where id =2;`





- **Introdução a SQL**
  - **Funções de agregação:**

`select count(*) from fornecimento;` -> conta linhas

`select sum(peso) from peca;` -> soma dos valores do campo

`select avg(peso) from peca;` -> média dos valores do campo

`select min(peso) from peca;` -> mínimo dos valores do campo



## PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**
  - **Funções matemáticas:**
    - `select trunc(78.9);` -> inteiro
    - `select sqrt(121);` -> raiz quadrada
    - `select power(2,4);` -> exponencial
    - `select abs(-23.76);` -> absoluto
    - `select trunc(peso) from peca;`



- **Introdução a SQL**

- **Funções com strings:**

- `select fnome||sobrenome from fornecedor;` -> concatenação
    - `select bit_length(fnome) from fornecedor;` -> número de bits
    - `select char_length('fulano');` -> número de caracteres
    - `select upper('fulano');` -> maiúsculas
    - `select lower('FULANO');` -> minúsculas
    - `select substring('Olivia Palito' from 8);` -> string a partir da posição indicada



Execute as três primeiras instruções para fazer os testes!!!

- **Introdução a SQL**

- Funções de manipulação e formatação de data e hora:

`alter table fornecimento add column data date;`

`update fornecimento set data = current_timestamp;`

`select * from fornecimento;`

`select to_char(data, 'Day, DD HH12:MI:SS') from fornecimento;` -> dia da semana, dia do mês e hora

`select to_char(data, 'Month') from fornecimento;` -> mês

`select to_char(data, 'DD') from fornecimento;` -> dia do mês

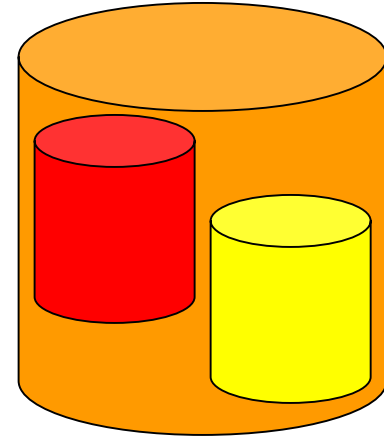
`select to_char(data, 'DDD') from fornecimento;` -> dia do ano



- **Introdução a SQL**
  - Funções de formatação de data e hora:
  - `select to_char(data, 'HH12:MI:SS') from fornecimento;`
  - `select to_date('12/05/2006', 'DD/MM/YYYY');`
  - `insert into fornecimento values ('F3','P2','J4',900,to_date('01/01/2006','dd/mm/yyyy'));`
  - `select to_char(data, 'HH24:MI:SS') from fornecimento;`
  - `select to_char(data, 'HH24:MI:SS') from fornecimento;`



- **Introdução a SQL**
  - Funções de endereçamento de rede:
  - `select netmask('192.168.4.1');` -> retorna a máscara de rede;
  - `select network('200.135.240.29');` -> extrai a parte da rede do endereço
  - `select cidr('10.1.2');` -> retorna o número de bits de uma máscara de rede sem identificação de classe
  - `select family('200.135.240.29');` -> retorna a família da rede, se ipv4 ou ipv6
  - `select family('3ffe:6a88:85a3::0370:7344');`

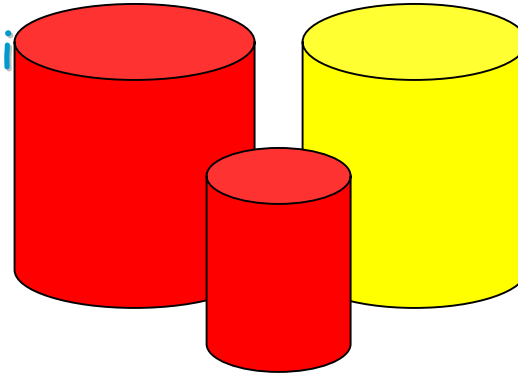


- **Introdução a SQL**
  - Operações especiais:
  - **UNIÃO**
  - Retorna todos os valores de um ou mais atributos de tipos compatíveis das duas relações.

```
SELECT FNAME, CIDADE FROM PROJETOS.FORNECEDOR
```

```
UNION
```

```
SELECT CNAME, CIDADE FROM PROJETOS.CLIENTE;
```



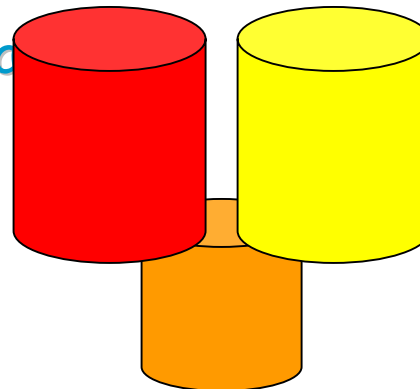
- **Introdução a SQL**
  - Operações especiais:
  - DIFERENÇA
  - Retorna os valores de um ou mais atributos de tipos compatíveis que pertencem a primeira relação mas não a segunda.

```
SELECT CIDADE FROM PROJETOS.FORNECEDOR
```

```
EXCEPT
```

```
SELECT CIDADE FROM PROJETOS.CLIENTE;
```





- **Introdução a SQL**
  - Operações especiais:
  - **INTERSECÇÃO**
  - Retorna os valores de um ou mais atributos de tipos compatíveis que são comuns as duas relações.

```
SELECT CIDADE FROM PROJETOS.FORNECEDOR
```

```
INTERSECT
```

```
SELECT CIDADE FROM PROJETOS.CLIENTE;
```



- **Introdução a SQL**

- **Visões:** são objetos do banco de dados que tem a finalidade de filtrar dados para o usuário, a partir de uma ou mais tabelas, de forma a garantir a segurança e controle de acesso a estes dados.

```
create view v_cliproj as (  
select Cnome, Jnome from projetos.cliente, projetos.cliproj,  
projetos.projeto where  
projetos.cliente.cod_cli = projetos.cliproj.cod_cli and  
projetos.projeto.cod_proj = projetos.cliproj.cod_proj);  
select * from v_cliproj;
```



- **Introdução a SQL**

- **Funções:** são utilizadas para obter resultados a partir de um conjunto de dados do banco de dados. Quando armazenadas no banco de dados, aumentam a performance de processamento e diminuem a sobre carga no tráfego de rede, assim como os procedimentos armazenados. Estes por sua vez, são responsáveis pelas transações no banco de dados.
- Uma transação se caracteriza pela modificação do banco de dados, através de comandos de inserção, exclusão ou alteração de dados.



# PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**

- **Funções:**

```
CREATE OR REPLACE FUNCTION idade(char)
```

```
RETURNS integer AS '
```

```
DECLARE
```

```
    idade integer;
```

```
BEGIN
```

```
    select (current_date - datanasc)/365 into idade from aluno where nome  
        = $1;
```

```
    RETURN idade;
```

```
END;
```

```
' LANGUAGE plpgsql;
```



# PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**

- **Funções:**

```
CREATE OR REPLACE FUNCTION media(char)
```

```
RETURNS numeric AS '
```

```
DECLARE
```

```
    med numeric;
```

```
BEGIN
```

```
    select avg(quantidade) into med from fornecimento where cod_fornec =  
        $1;
```

```
    RETURN med;
```

```
END;
```

```
' LANGUAGE plpgsql;
```



- **Introdução a SQL**
  - **Procedimentos: “funções sem retorno”**

```
CREATE OR REPLACE FUNCTION projetos.aumenta(numeric) returns numeric  
AS $BODY$
```

```
BEGIN
```

```
    update projetos.fornecimento set quantidade = quantidade + $1;
```

```
    return null;
```

```
END; $BODY$
```

```
LANGUAGE 'plpgsql';
```

```
SELECT projetos.aumenta(2000);
```



## PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**
  - **Triggers: Gatilhos** – são disparados a partir de um evento no banco de dados, como exclusão, inserção ou atualização. Podem acontecer antes ou depois destes eventos.
  - No PostgreSQL os gatilhos executam procedimentos ou “funções sem retorno”.



# PostgreSQL - Funcionalidades Básicas

- **Introdução a SQL**

```
CREATE OR REPLACE FUNCTION backupfor() RETURNS trigger AS  
$$
```

```
BEGIN
```

```
insert into bakfornec (SELECT * FROM FORNECEDOR);
```

```
RETURN NULL;
```

```
END;
```

```
$$
```

```
LANGUAGE plpgsql;
```

```
CREATE TRIGGER tbackup
```

```
AFTER DELETE ON fornecedor
```

```
FOR EACH ROW EXECUTE PROCEDURE backupfor();
```

O gatilho indicado é responsável por executar a função que faz uma cópia das linhas da tabela fornecedor para uma tabela auxiliar, toda vez que uma exclusão é executada.





- **Introdução a SQL - Exercícios**

- 1) Adicione uma coluna a tabela Peça, nomeando-a como "valor".
- 2) Atualize esta tabela adicionando valores a nova coluna.
- 3) Crie um procedimento que aumente em 3% os valores das peças.
- 4) Altere o procedimento anterior passando o percentual como parâmetro.
- 5) Crie uma função que retorne o somatório dos valores das peças.
- 6) Crie uma visão que retorne os nomes dos projetos e o nome das peças que foram fornecidas para estes projetos.
- 7) Crie uma visão que retorne os nomes das peças e suas quantidades, somente das peças que tem quantidade inferior a quantidade média fornecida.



### - Módulo Três

- Estrutura de tabelas:

- Uma tabela é definida a partir de uma classe do PostgreSQL. Assim, para verificar a estrutura de uma tabela, é preciso acessar três tabelas do catálogo do sistema. São elas - `pg_class`, `pg_attribute`, `pg_type`.
- `pg_class` armazena as propriedades de classes como tabelas, visões, índices, etc ...
- `pg_attribute` guarda as propriedades dos atributos/colunas das tabelas e,
- `pg_type` descreve os tipos de dados disponíveis no sistema



### - Módulo Três

- Estrutura de tabelas:

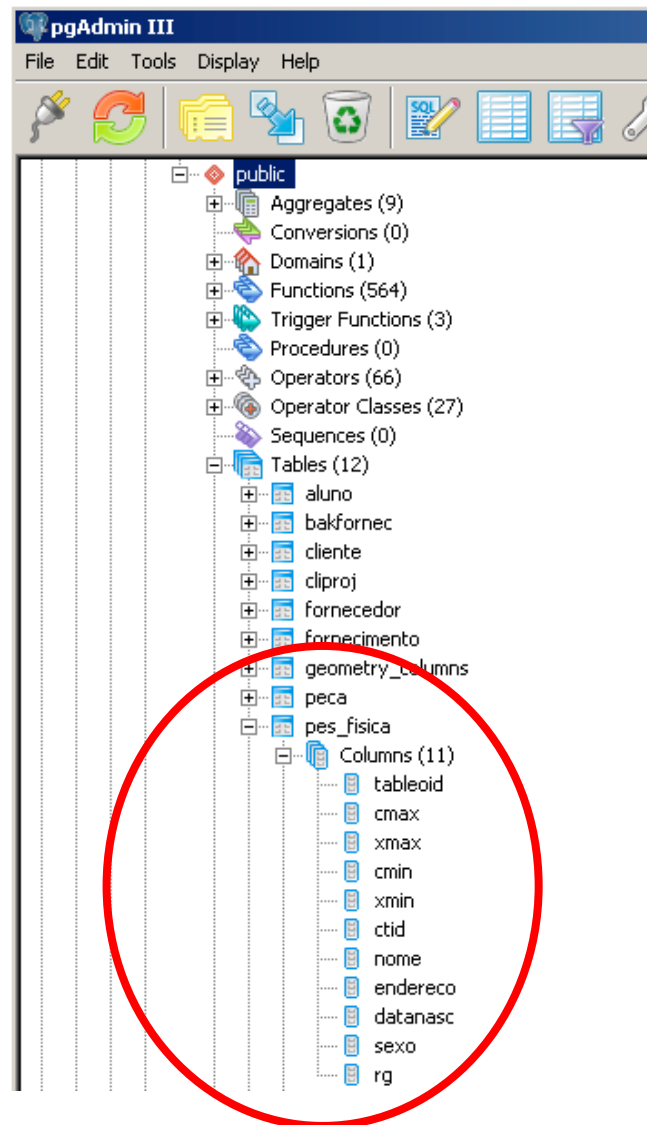
- `select relname, attname, typename from pg_class, pg_attribute, pg_type where pg_class.oid=attrelid and pg_type.oid = atttypid and relname = 'pessoa';`



# PostgreSQL - Funcionalidades Básicas

## - Módulo Três

- Estrutura de tabelas:
- É claro que você pode utilizar uma interface para visualizar as estruturas das tabelas ou outros objetos.





### - Módulo Três

- Agrupamento de tabelas:
  - o agrupamento pode ser entendido de duas formas - através de operações de junção - join, já abordadas no módulo II ou, como tabelas clusterizadas, visando ganho de performance no banco de dados.



## PostgreSQL - Funcionalidades Básicas

- **Módulo Três**
- **Agrupamento de tabelas:**
  - **Clusters** - um cluster pode ser considerado um agrupamento de tabelas ou ainda uma agrupamento de linhas em uma tabela. Deve ser utilizado quando duas tabelas são frequentemente utilizadas em consultas ou visões, no primeiro caso e, para agrupar linhas de uma mesma tabela de possuam o mesmo valor para um índice.



## PostgreSQL - Funcionalidades Básicas

- **Módulo Três - Agrupamento de tabelas:**
  - O PostgreSQL faz agrupamento de linhas de uma tabela e não agrupamento de tabelas.
  - Neste caso é necessário que o índice já tenha sido criado na tabela. A partir do índice, o comando "CLUSTER" informa ao gerenciador que as linhas de uma tabela devem ser agrupadas por aquele índice. Quando é feita a consulta sobre a tabela, tendo como parâmetro de busca os valores do índice, a busca obviamente torna-se mais rápida, pois as linhas já estão ordenadas pelo valor do índice.
  - É bom lembrar que o cluster é estático. Isto é, as novas inserções na tabela não são consideradas até que o comando seja executado novamente.



## PostgreSQL - Funcionalidades Básicas

- **Módulo Três**
- **Agrupamento de tabelas:**
  - Considerando que se queira consultar a tabela "fornecimentos" com mais frequência passando como parâmetros os valores dos projetos, é importante indexá-la e criar um cluster a partir do índice.

pgAdmin III Edit Data - PostgreSQL Database Server 8.1 (localhost:5

	cod_fornec varchar	cod_peca varchar	cod_proj varchar	quantidade numeric
1	F1	P1	J1	7200
2	F1	P1	J4	7700
3	F2	P3	J1	7400
4	F2	P3	J2	7200
5	F2	P3	J3	7200
6	F2	P3	J4	7500
7	F2	P3	J5	7600
8	F2	P3	J6	7400
9	F2	P3	J7	7800
10	F2	P5	J2	7100
11	F3	P3	J1	7200
12	F3	P4	J2	7500
13	F4	P1	J4	7100
14	F4	P6	J3	7300
15	F4	P6	J7	7300
16	F5	P2	J2	7200
17	F5	P2	J4	7100
18	F5	P3	J4	7200
19	F5	P4	J4	7800
20	F5	P5	J4	7400
21	F5	P5	J5	7500
22	F5	P5	J7	7100
23	F5	P6	J2	7200
24	F5	P6	J4	7500
25	F9	P9	J11	200
26	F9	P9	J20	700
27	F9	P3	J11	400
28	F9	P3	J20	200
29	F9	P3	J30	200
30	F9	P3	J20	500





## PostgreSQL - Funcionalidades Básicas

- **Módulo Três**
- **Agrupamento de tabelas:**
- `create index indproj on  
fornecimentos(cod_proj);`
- `cluster indproj on  
fornecimentos;`

pgAdmin III Edit Data - PostgreSQL Database Server 8.1 (localhost:5

	cod_fornec varchar	cod_pec varchar	cod_proj varchar	quantidade numeric
1	F1	P1	J1	7200
2	F2	P3	J1	7400
3	F3	P3	J1	7200
4	F9	P9	J11	200
5	F9	P3	J11	400
6	F10	P3	J11	200
7	F2	P3	J2	7200
8	F2	P5	J2	7100
9	F3	P4	J2	7500
10	F5	P2	J2	7200
11	F5	P6	J2	7200
12	F9	P9	J20	700
13	F9	P3	J20	200
14	F9	P3	J20	500
15	F9	P87	J20	100
16	F10	P4	J20	500
17	F2	P9	J20	100
18	F20	P9	J20	200
19	F20	P9	J20	100
20	F20	P3	J20	200
21	F20	P4	J20	800
22	F20	P87	J20	400
23	F20	P6	J20	200
24	F20	P6	J20	500
25	F2	P3	J3	7200
26	F4	P6	J3	7300
27	F9	P3	J30	200
28	F2	P6	J30	300
29	F1	P1	J4	7700
30	F2	P3	J4	7500



- **Módulo III**

- **Conexões nativas**

- Para acessar o banco de dados, são necessárias interfaces implementadas através de linguagens de programação. Para isso existem algumas formas de conectar a aplicação ao SGBD.
    - Conexão nativa
    - ODBC
    - JDBC
    - ponte ODBC-JDBC



- **Módulo III**

- **Conexões nativas**
- **Conexão nativa:** este tipo de conexão é feito através de funcionalidades próprias das linguagens e não tem a intermediação de um “driver” para acesso ao banco de dados;
- **Exemplo PHP:**
- `$conexao = pg_connect('host=localhost port=5432 dbname=postgres user=postgres password=postgres');`



- **Módulo III**

- **Conexões nativas**

- **Exemplo Perl:**

```
$dbh = DBI->connect($db, $db_user, $db_pass,  
                    { RaiseError => 1, AutoCommit => 0 }  
    ) || die "Erro ao conectar ao banco: $DBI::errstr\n";
```

- **Exemplo Python:**

```
con = PgSQL.connect(host='localhost', database='banco',  
                    user='postgres', password='postgres')
```



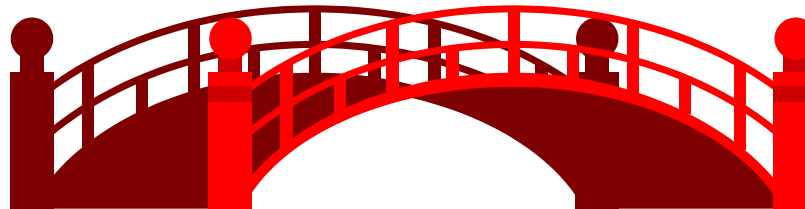
## • Módulo III

### - ODBC

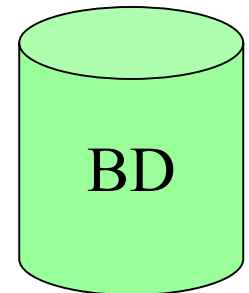
- ODBC: é uma tecnologia Microsoft para acessar a diversos bancos de dados. O driver ODBC é fornecido pelo fabricante do SGBD.



Interface



ODBC



Base de Dados



- **Módulo III**

- ODBC: é através do driver ODBC que a estrutura interna do banco de dados e os tipos de dados são reconhecidos, bem como a conversão dos dados é feita, caso seja necessário. É uma especificação para acesso a bancos de dados, com funções para conexão e execução de comandos e transações.



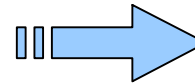
# PostgreSQL - Funcionalidades Básicas



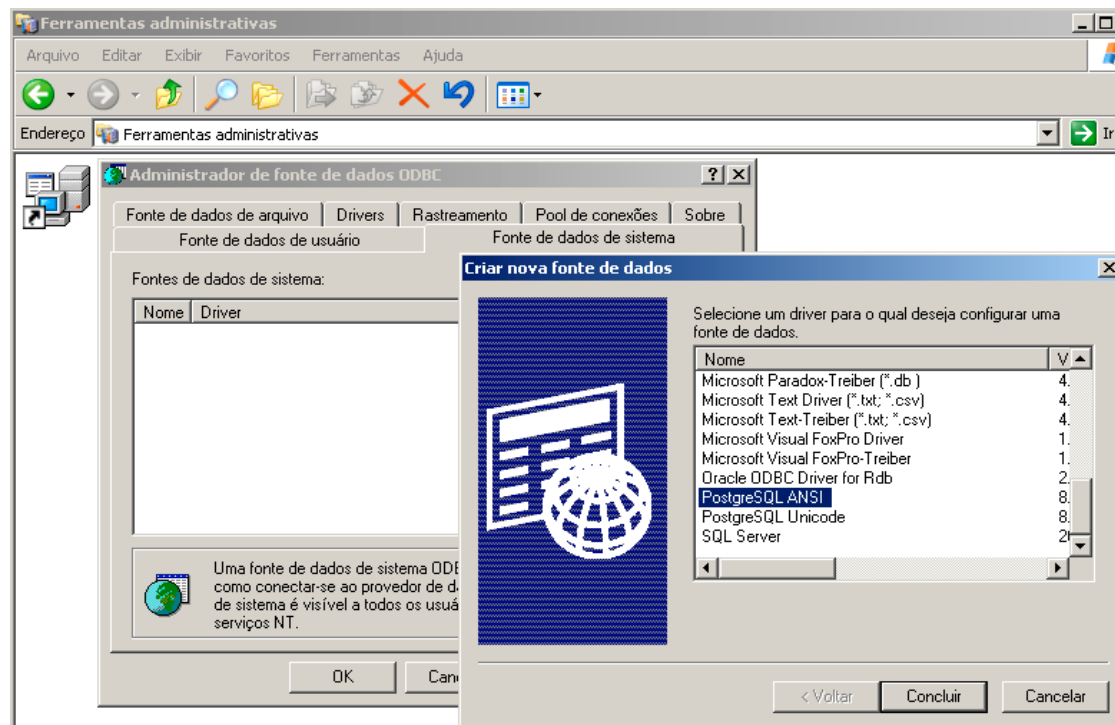
Microsoft

- **Módulo III**
- **ODBC**

- Instalação do Driver
- Painel de Controle
- Configuração

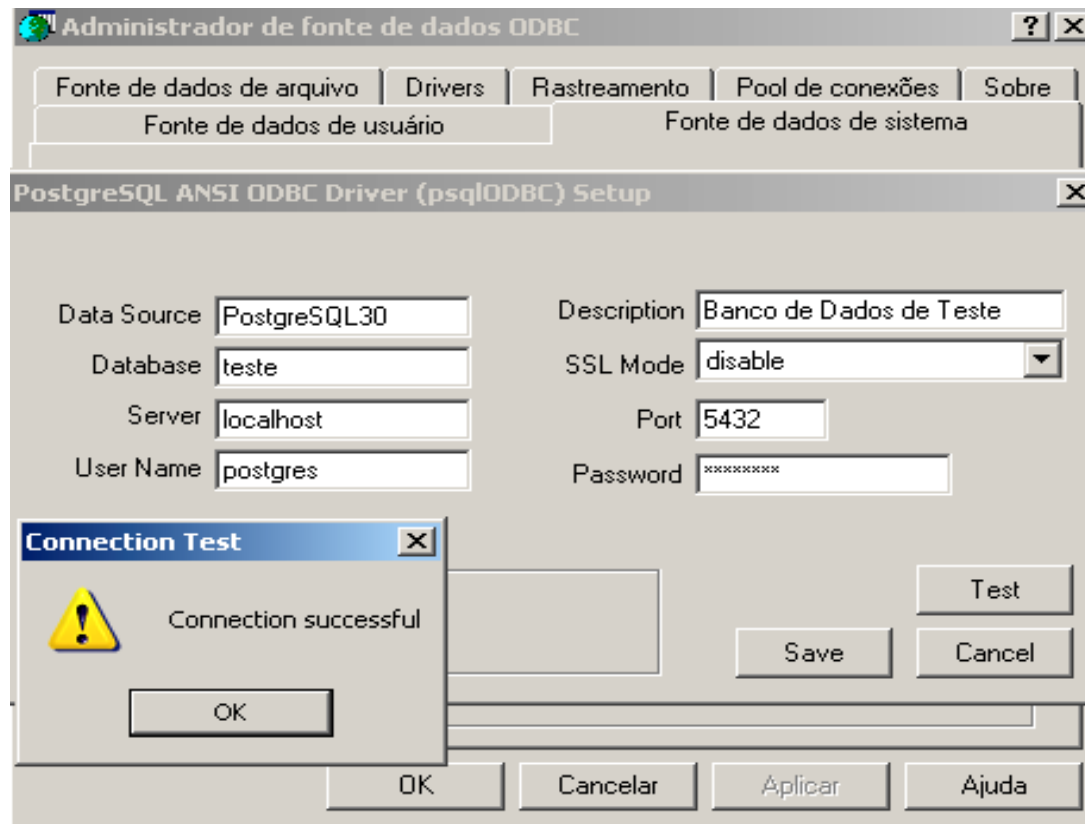


Fontes de dados (ODBC)  
Atalho  
2 KB





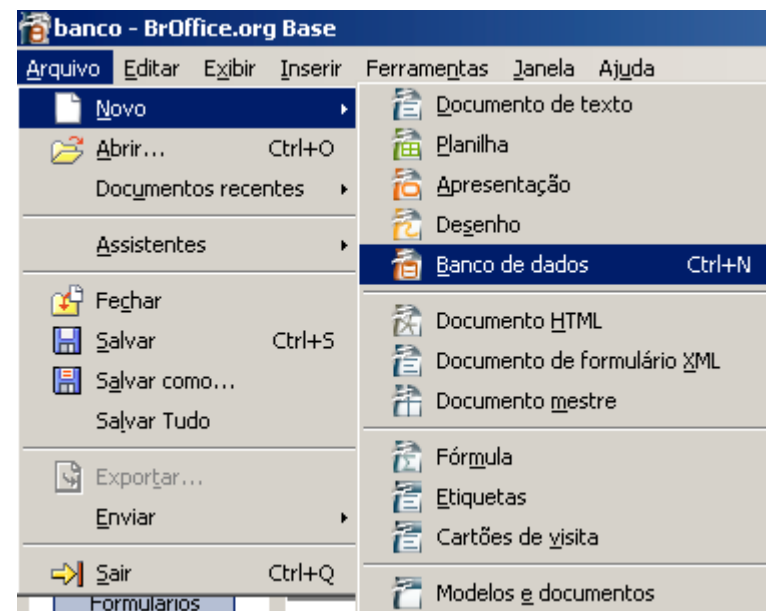
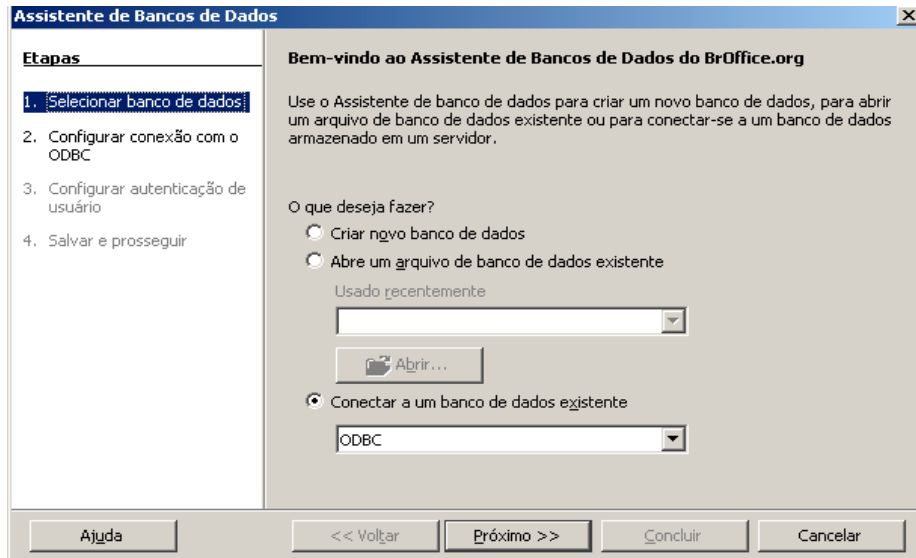
- **Módulo III**
- **ODBC**
  - **Configuração**





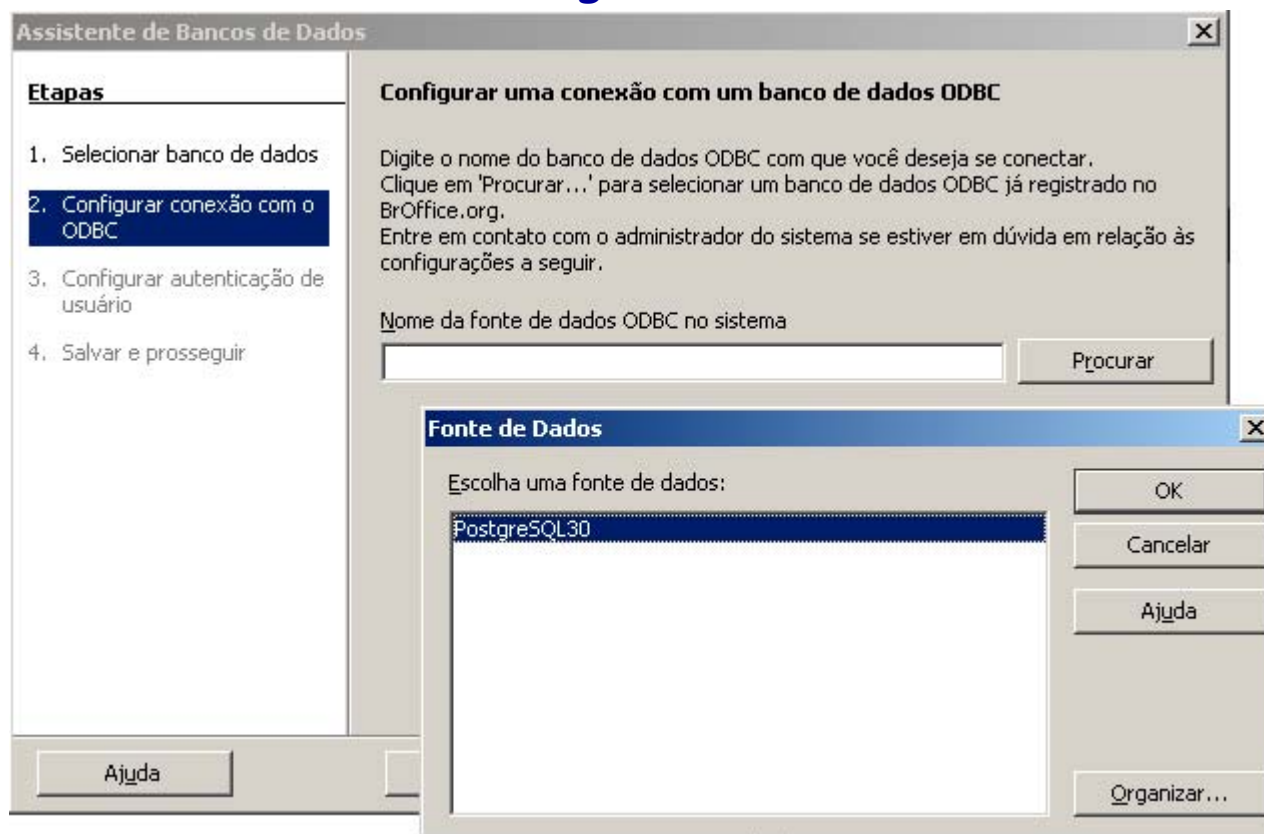


- **Módulo III**
- **ODBC**
  - Teste de conexão ODBC:
    - Acesse o BrOffice.org





- **Módulo III**
- **ODBC**
  - **Teste de conexão ODBC:**
    - **Acesse o BrOffice.org**

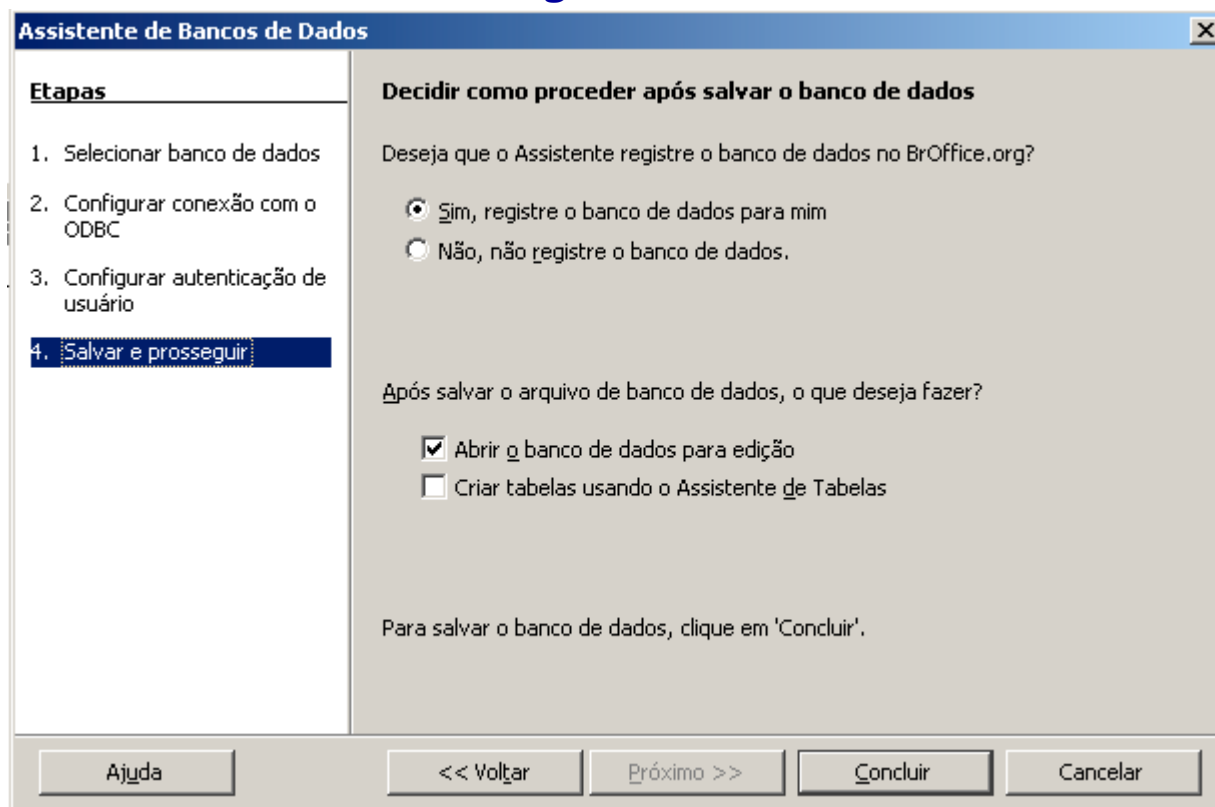




- **Módulo III**
- **ODBC**
  - **Teste de conexão ODBC:**
    - **Acesse o BrOffice.org**



- **Módulo III**
- **ODBC**
  - **Teste de conexão ODBC:**
    - **Acesse o BrOffice.org**

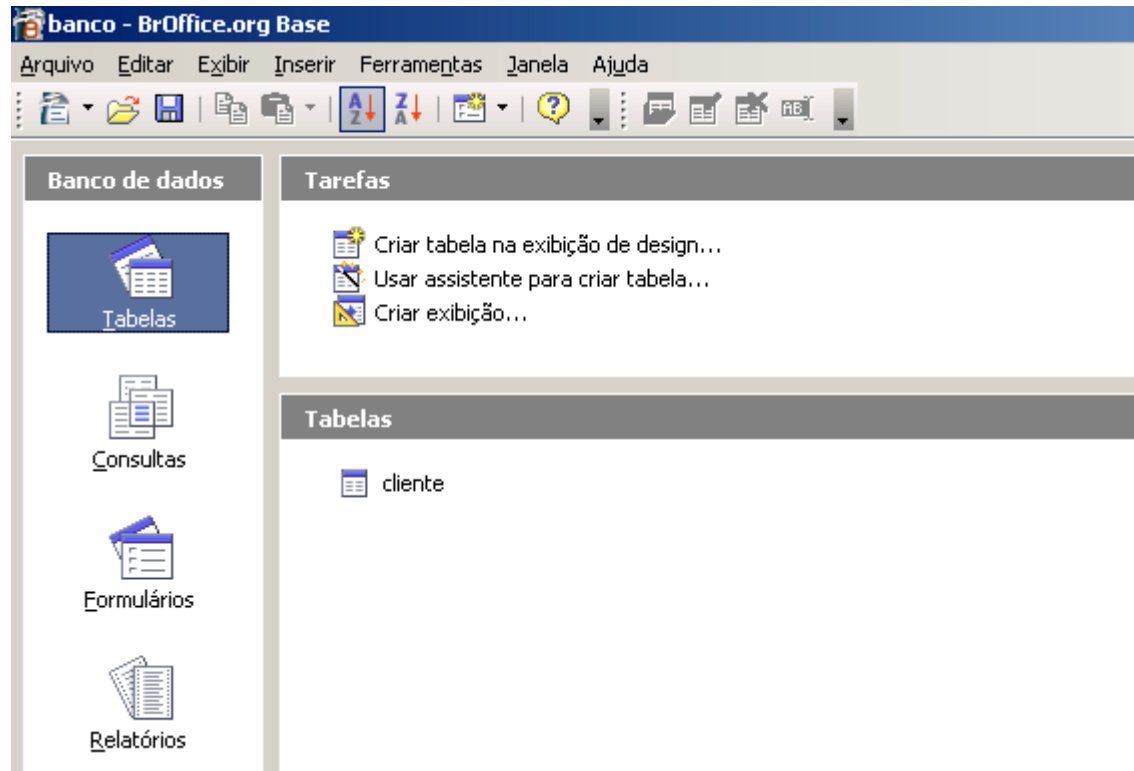




# PostgreSQL - Funcionalidades Básicas



- **Módulo III**
- **ODBC**
  - Teste de conexão ODBC:
    - Acesse o [BrOffice.org](http://BrOffice.org)





## • Módulo III

### - ODBC - Exemplo Php:

Digite para a conexão:

- datasource
- usuario
- senha

```
<?
```

```
$conexao = odbc_connect('PostgreSQL30', 'postgres', 'postgres');
```

```
$instrucao = "select cod_cli, cnome from cliente";
```

```
$resultado = odbc_exec($conexao,$instrucao);
```

```
while ($linha = odbc_fetch_array($resultado)) {
```

```
print "<pre>";
```

```
echo $linha[cod_cli],"----", $linha[cnome];
```

```
print "<pre>";};
```

```
?>
```



## • Módulo III

- JDBC
- Um driver JDBC é uma interface, uma API como ODBC, para acessar bancos de dados usando a linguagem de programação Java.
- Tem as mesmas funcionalidades de um driver ODBC, como funções de conexão, recuperação de dados, controle de transações, etc ...
- Não precisa ser instalado - o driver é um "arquivo jar". Só é necessário mapear a localização do arquivo no disco.



- **Módulo III**
  - **JDBC - Exemplo**

```
import java.sql.*;
import java.io.*;
public class post
{
    public static void main (String args [ ] ) throws SQLException, IOException
    {
        System.out.println ("Carregando o driver Jdbc ... " );
        try { Class.forName("org.postgresql.Driver");}
        catch(ClassNotFoundException e)
        {
            System.out.println("O drive nao carrega ....");
            e.printStackTrace();
            return; }
    }
```





- **Módulo III**
  - **JDBC - Exemplo**

```
System.out.println("Conectando ....");  
// Cria a conexão com o banco de dados  
String url =  
    "jdbc:postgresql://localhost/teste?user=postgres&password=postgres";  
Connection conn = DriverManager.getConnection(url);  
// Cria uma instrução SQL para consulta dos dados  
Statement stmt = conn.createStatement();  
// Passa como parâmetro a instrução que retorna um conjunto de registros  
ResultSet rs = stmt.executeQuery("select cod_cli, cnome from cliente");
```



- **Módulo III**
  - **JDBC - Exemplo**

```
// lê o conjunto de registros enquanto não for fim de cursor
while (rs.next()) {
    // variáveis que recebem os valores lidos da tabela
    double c = rs.getDouble("cod_cli");
    String n = rs.getString("cnome");
    // retorna na tela os valores armazenados nas variáveis
    System.out.println(c + " " + n + " ");
}
stmt.close(); // fecha o cursor
conn.close(); // fecha a conexão
}
```



- **Módulo III**
  - **JDBC - Exemplo**

Não esqueça de  
incluir  
o driver JDBC  
no CLASSPATH.

- Painel de controle
- Sistema
- Guia Avançado
- Variáveis de Ambiente

**CLASSPATH=C:\Arquivos de programas\PostgreSQL\8.1\jdbc\postgresql-8.1-404.jdbc3.jar**

**CLASSPATH=/usr/local/lib/myapp.jar:/usr/local/pgsql/share/java/postgresql.jar**



- **Módulo III**
  - JDBC - Exemplo

Eclipse + Plugin + JDBC + PostgreSQL  
Netbeans + JDBC + PostgreSQL

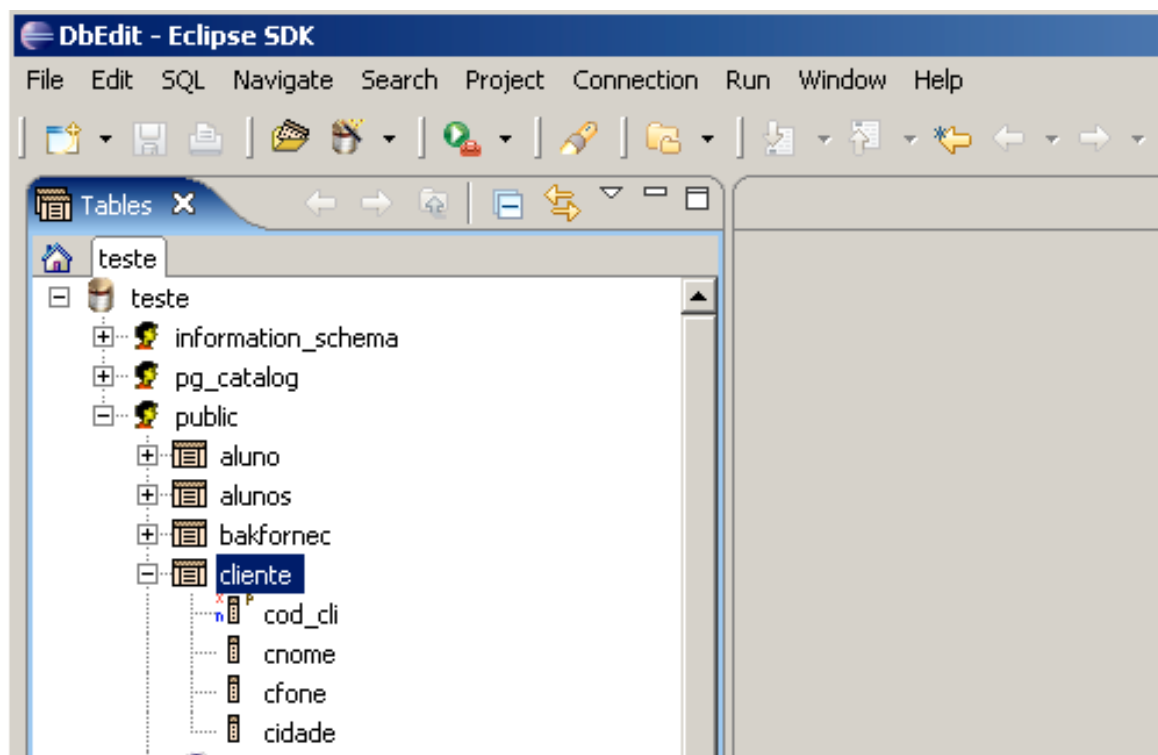
Para este exemplo vamos usar:  
Eclipse + DBEdit + JDBC + PostgreSQL



## • Módulo III

### - JDBC - Exemplo

- Adicione o plugin do DBEdit ao Eclipse
- Altere o ambiente da perspectiva Java para a perspectiva do DBEdit

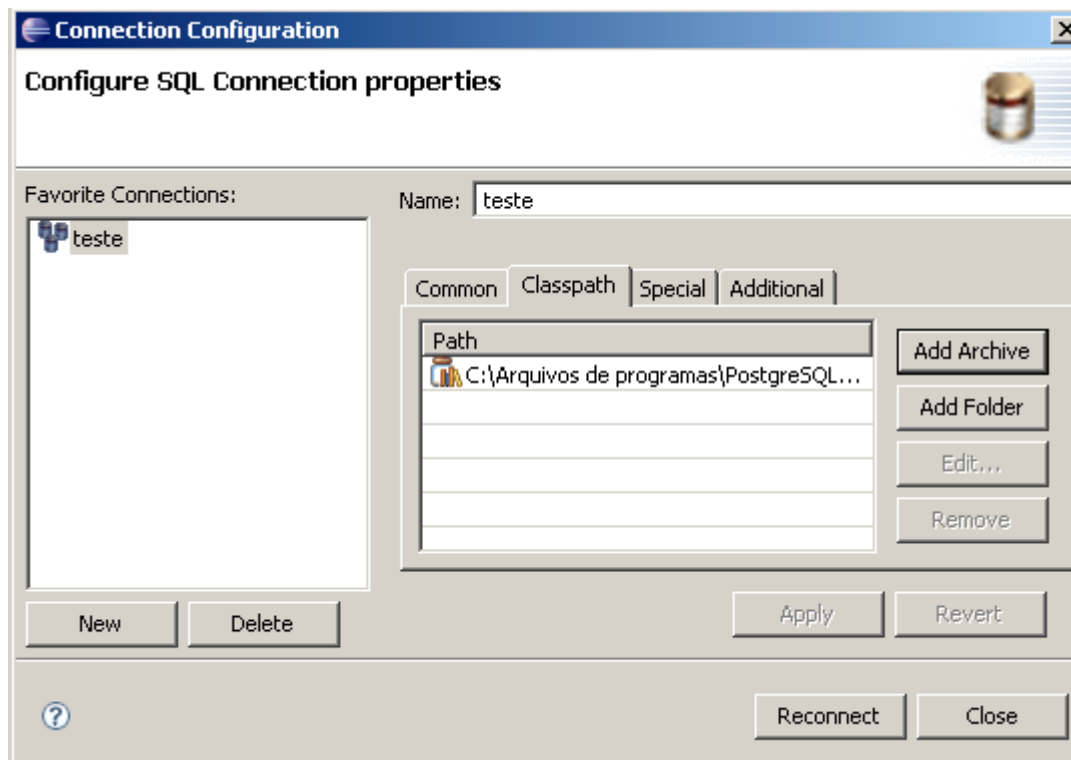




## • Módulo III

### - JDBC - Exemplo

- Crie uma conexão com o banco usando o driver JDBC
- Configure a conexão com o banco de dados

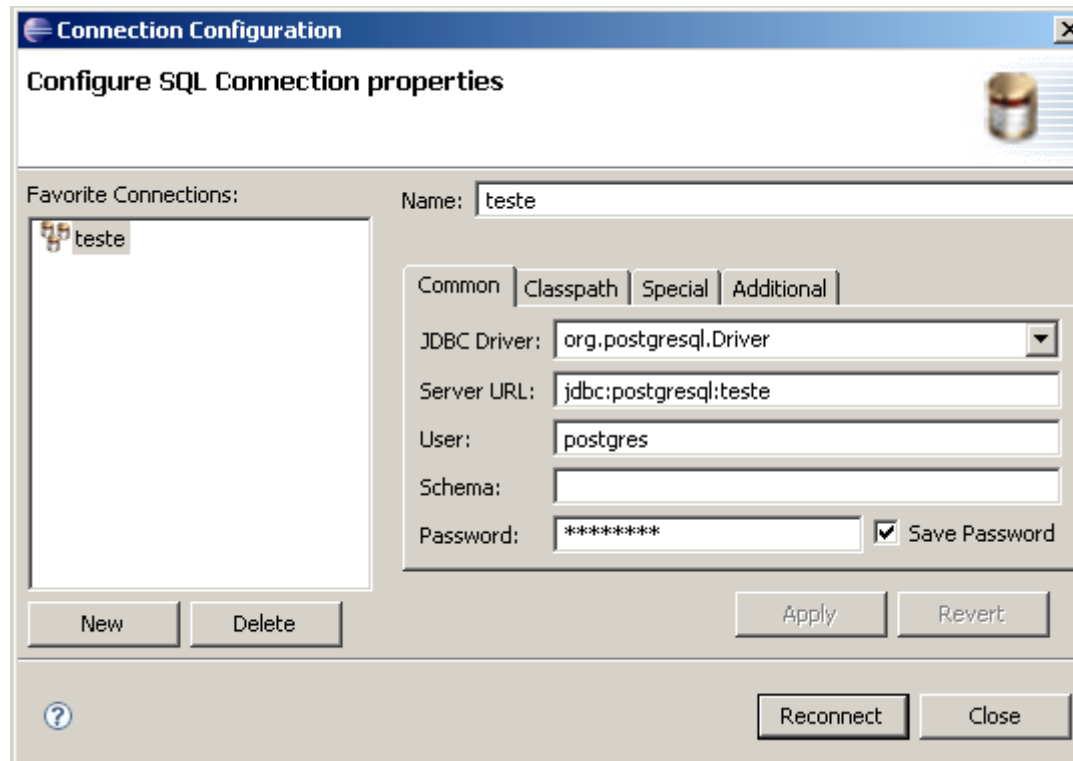




## • Módulo III

### - JDBC - Exemplo

- Crie uma conexão com o banco usando o driver JDBC
- Configure a conexão com o banco de dados

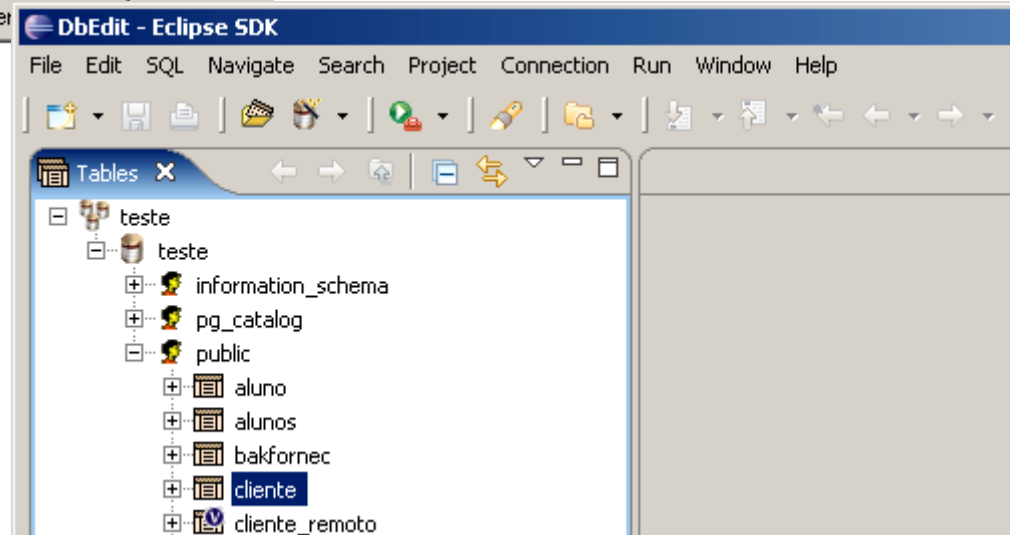
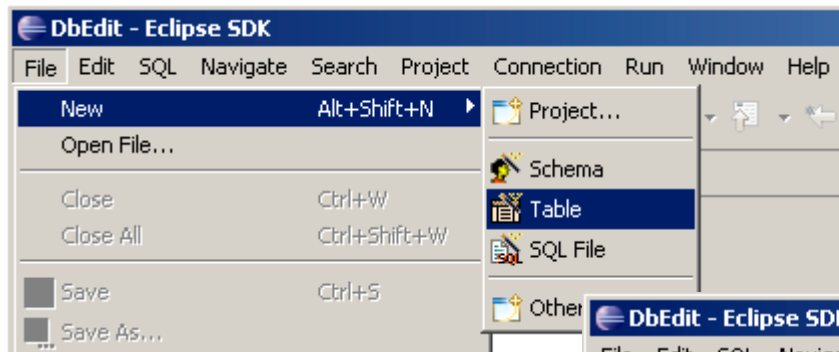




## • Módulo III

### - JDBC - Exemplo

- Adicione uma tabela ao projeto de banco de dados



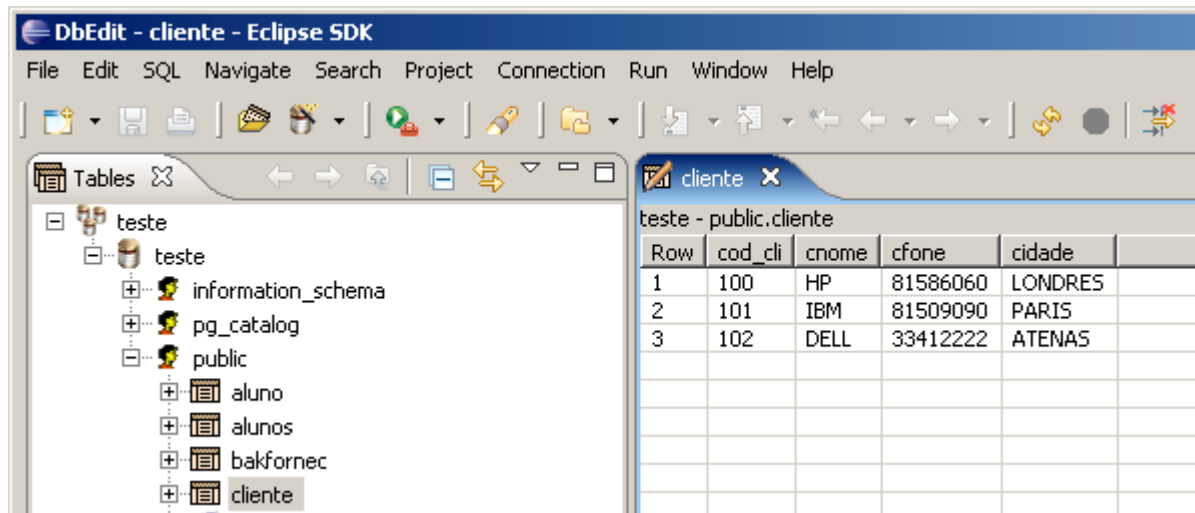




## • Módulo III

### - JDBC - Exemplo

- Adicione uma tabela ao projeto de banco de dados





- **Módulo III**
  - **JDBC - Exemplo**
    - **BrOffice.org + JDBC + PostgreSQL**

**Assistente de Bancos de Dados**

**Etapas**

1. Selecionar banco de dados
2. **Configurar conexão com o JDBC**
3. Configurar autenticação de usuário
4. Salvar e prosseguir

**Configurar uma conexão com um banco de dados JDBC**

Forneça as informações necessárias à conexão com um banco de dados JDBC. Entre em contato com o administrador do sistema se estiver em dúvida em relação às configurações a seguir.

URL da fonte de dados

Classe do driver JDBC



# PostgreSQL - Funcionalidades Básicas



- **Módulo III**
  - **JDBC - Exemplo**

**BrOffice.org +  
JDBC +  
PostgreSQL**

