
Banco de Dados II

Dr^a. Alana Moraes

O SGBD PostgreSQL

BD e esquemas

Revisão

Por onde nos comunicaremos?

<https://sites.google.com/site/disciplinasalana/ifpb/banco-de-dados-ii>

O que já foi visto?



O que já foi visto?

- Apresentação da Disciplina
- Contextualização Inicial



O que faremos hoje?

■ Revisão sobre Banco de Dados I

- DDL e DML: revisão e complementos

■ Vamos utilizar o Postgresql

- Como instalar?
- Como rodá-lo?

PostgreSQL

- Objeto-relacional

- Veremos a parte **relacional**

- Open-source

- Arquitetura cliente-servidor

- Podem estar em máquinas diferentes

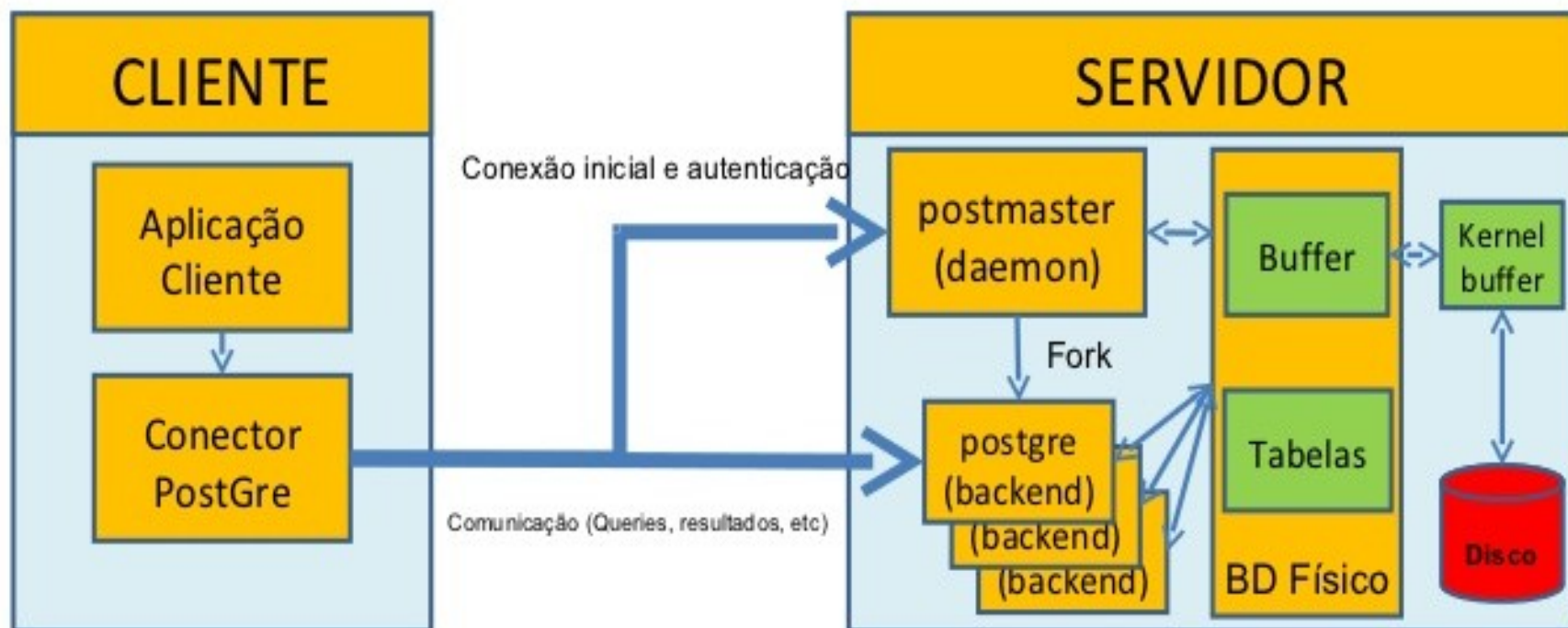
- Iniciado pelo prof. Stonebraker

- 16 anos no mercado

<https://>

www.postgresql.org/docs/9.5/static/tutorial-install.html

Arquitetura Cliente-Servidor



PostgreSQL

- SGBD “maduro”
- Flexível
- Robusto
- Confiável
- Integridade transacional
- Sintaxe:
<https://www.postgresql.org/docs/9.2/sql-syntax.html>

Recursos do PostgreSQL

- Controle de concorrência multiversão (MVCC)
- Controle SSL
- ACID
 - Atomicidade, Consistência, Isolamento e Durabilidade.
- Triggers
- Sub-consultas, etc.

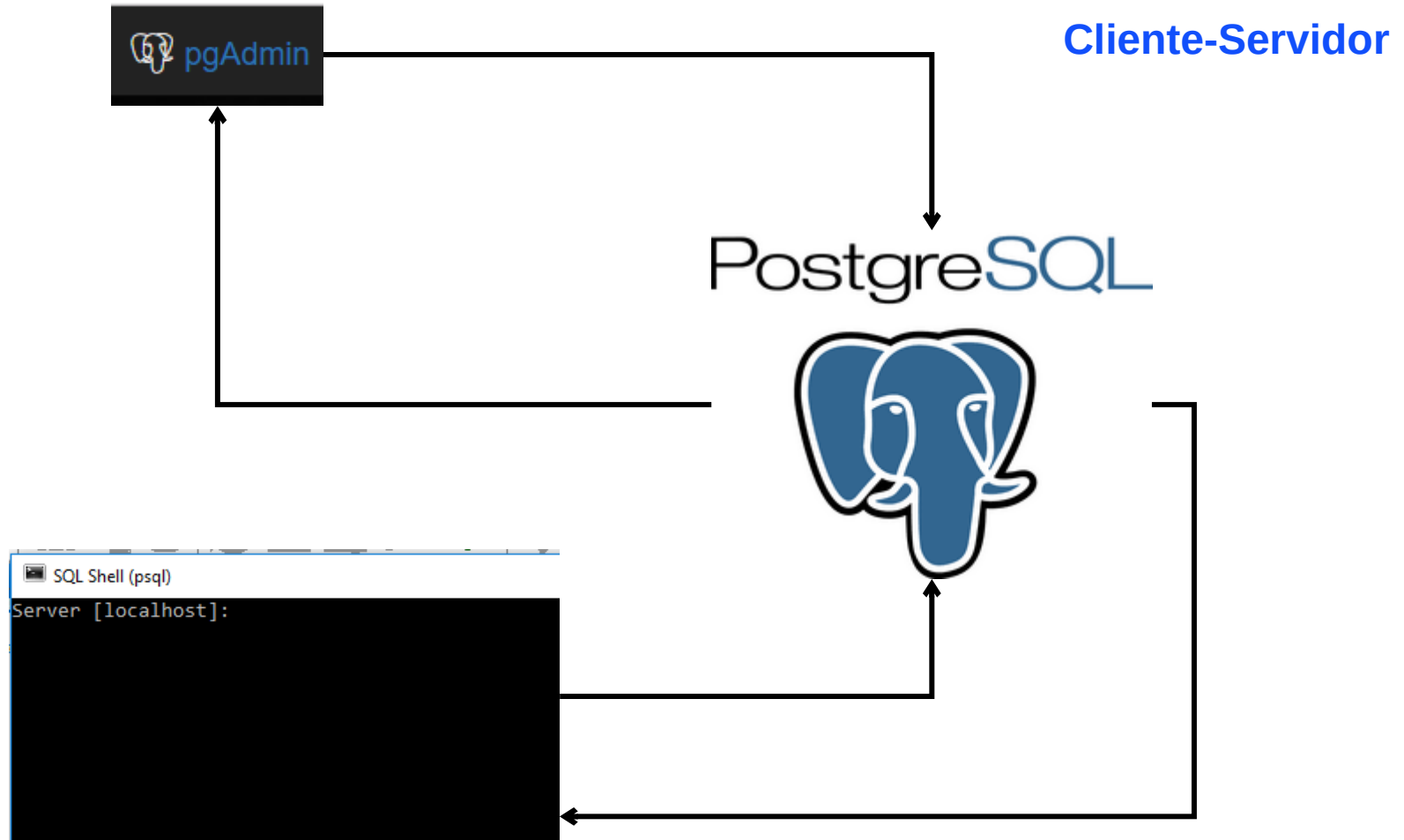
Como Instalar o Postgresql?

- Passo a passo no site:
 - Há um passo a passo no GIT da disciplina para Linux e Windows.

Como interagir com o PostgreSQL?

- Terminal (psql)
- Interface Gráfica
 - PgAdmin (<https://www.pgadmin.org/>)
 - PgModeler (<https://pgmodeler.io/>)

PgAdmin e Postgres



PostgreSQL- PSQL

- O psql é um arquivo executável, um cliente em modo texto que pode ser encontrado no diretório de instalação do Postgres
 - (no caso do Windows encontra-se em C:\Program Files\PostgreSQL\9.4\bin)
- <https://www.postgresql.org/docs/9.0/app-psql.html>

PostgreSQL- PSQL

■ Mostra versão do PostgreSQL

- `psql --version`

■ Lista os bancos de dados

- `psql -l`

■ (debug) Mostra internamente como cada consulta é realizada

- `psql banco -E`

■ Conecta à console psql no banco de dados

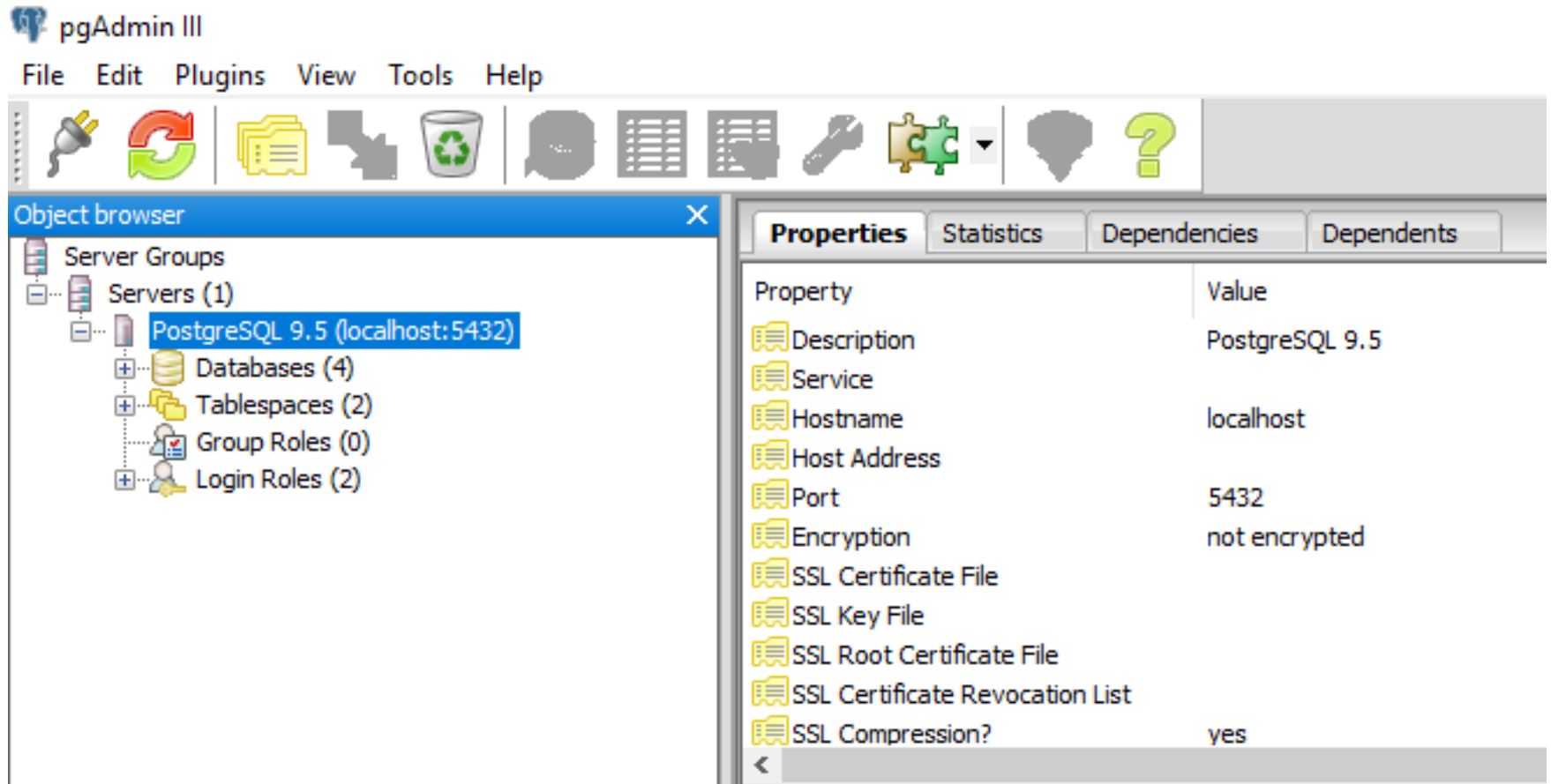
- `psql -U nomeusuario nomebanco`

PostgreSQL- PSQL

■ Se entrar no psql:

- =# - este prompt indica um superusuário
- => - este indica um usuário comum
- -# - indica comando não finalizado. Aguardando o ponto e vírgula
- (# - aguardando o fecha parênteses)
- '# - aguardando um fecha apóstrofo '

Conexão ao Banco



PostgreSQL e bancos

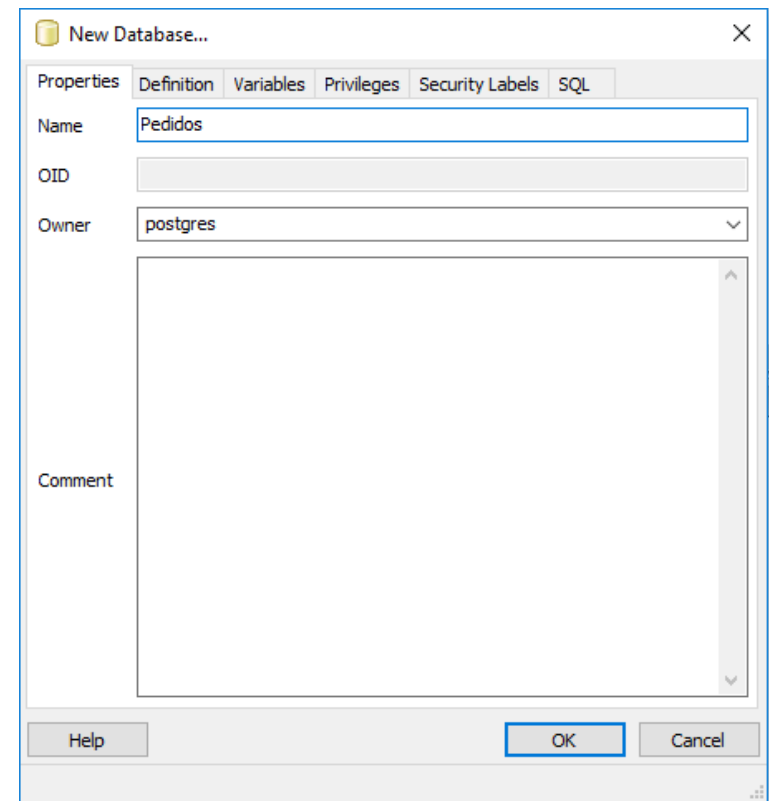
- O SGBD provê o gerenciamento de diferentes **bancos de dados**
- Cada **BD** está associado a um **projeto** (escopo) ou **aplicação**

➤ Create database

(New Database)

➤ Pedidos

- Create database Pedidos;
** Super usuário ou usuário
que tenha privilégio createdb



Create database

```
CREATE DATABASE "Pedidos"  
WITH OWNER = postgres  
ENCODING = 'UTF8'  
TABLESPACE = pg_default  
LC_COLLATE = 'Portuguese_Brazil.1252'  
LC_CTYPE = 'Portuguese_Brazil.1252'  
CONNECTION LIMIT = -1;
```



Sem
limite

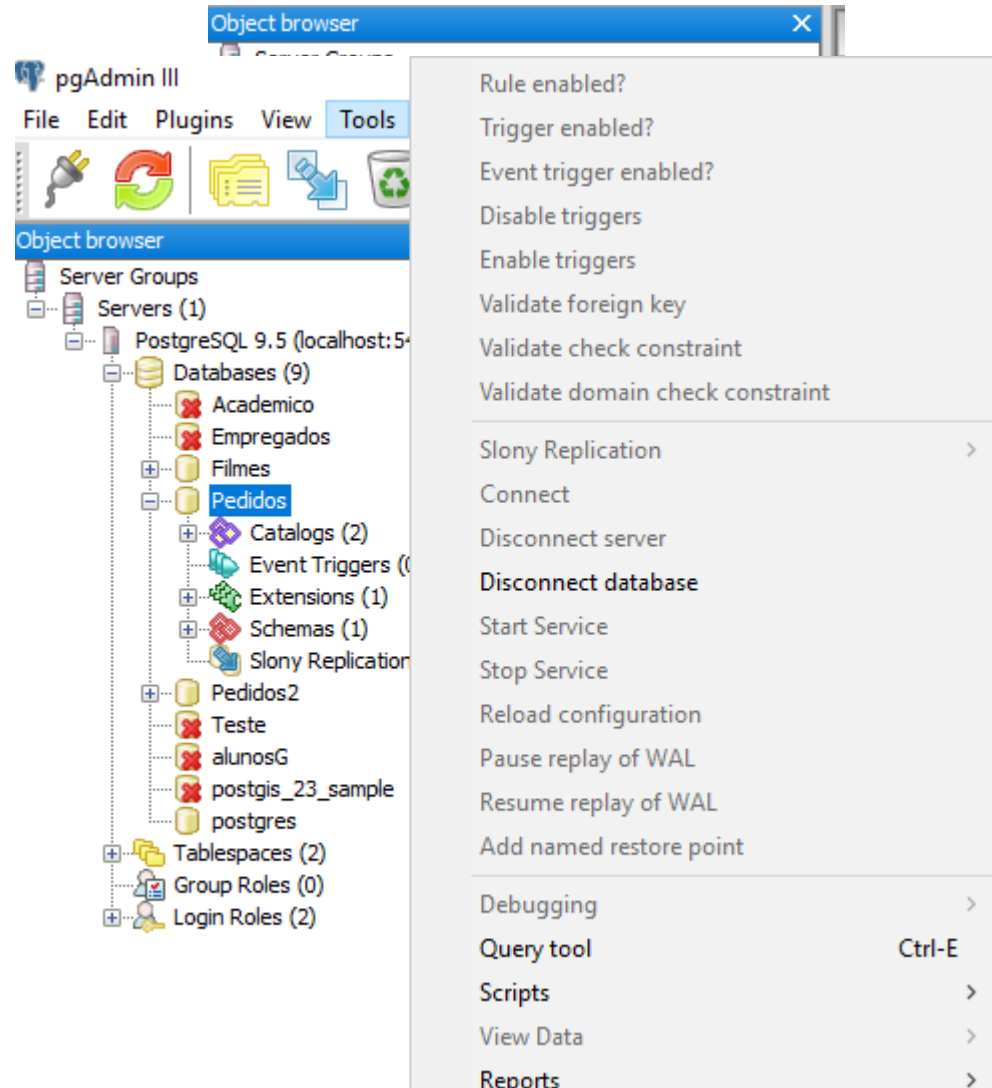
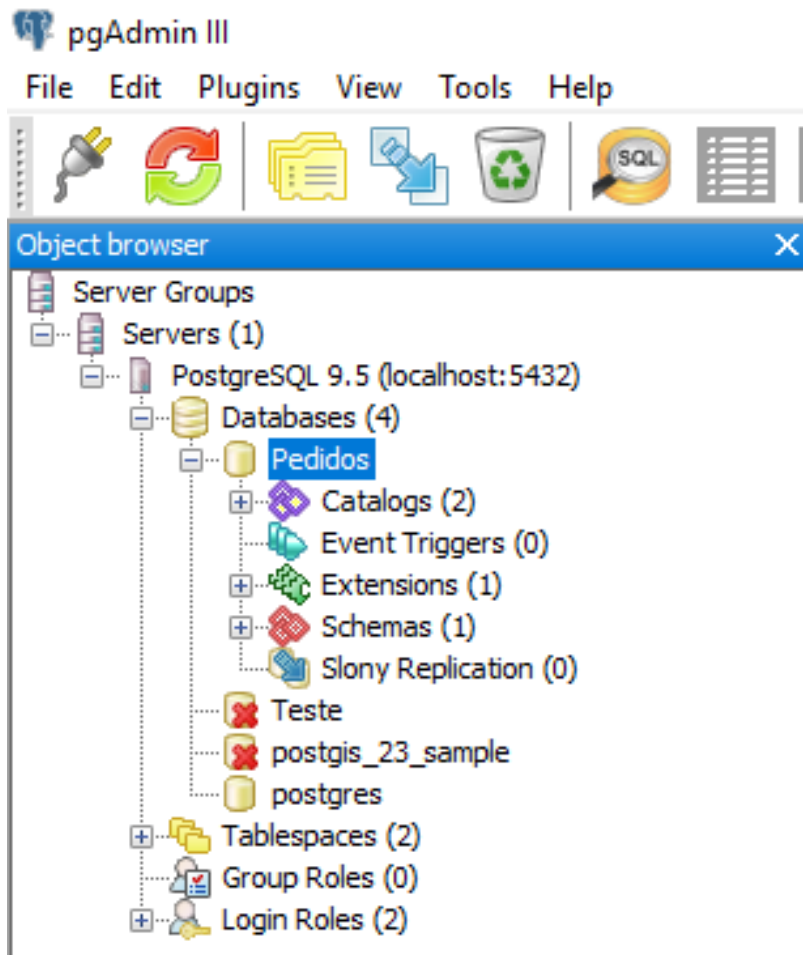
Ou `createdb Pedidos;`

Para remover: `Drop Database Pedidos;`

Para alterar: `Alter Database...`

PostgreSQL e SQL

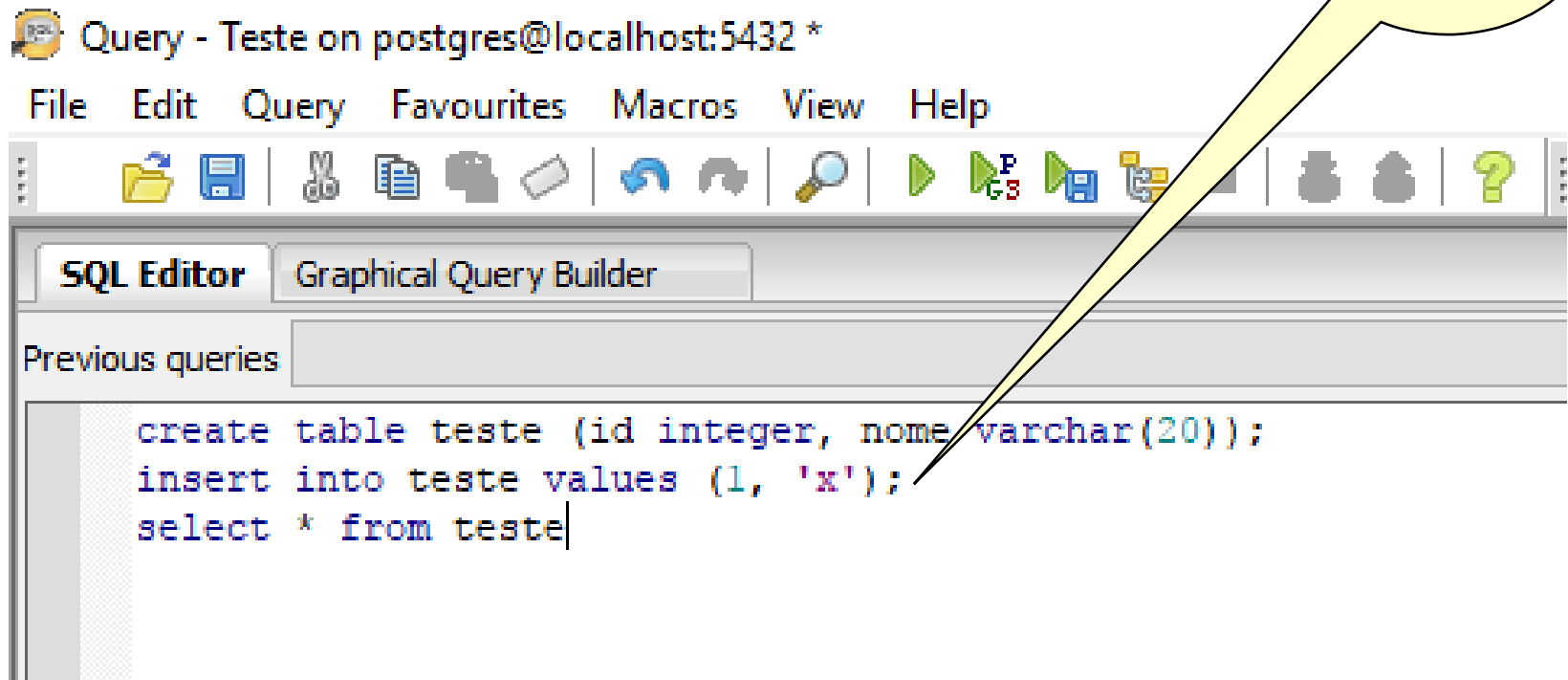
➤ Pedidos



Exemplo

1. Criar tabela
2. Popular Tabela
3. Consultar tabela

Insira 4
vezes



Exemplo

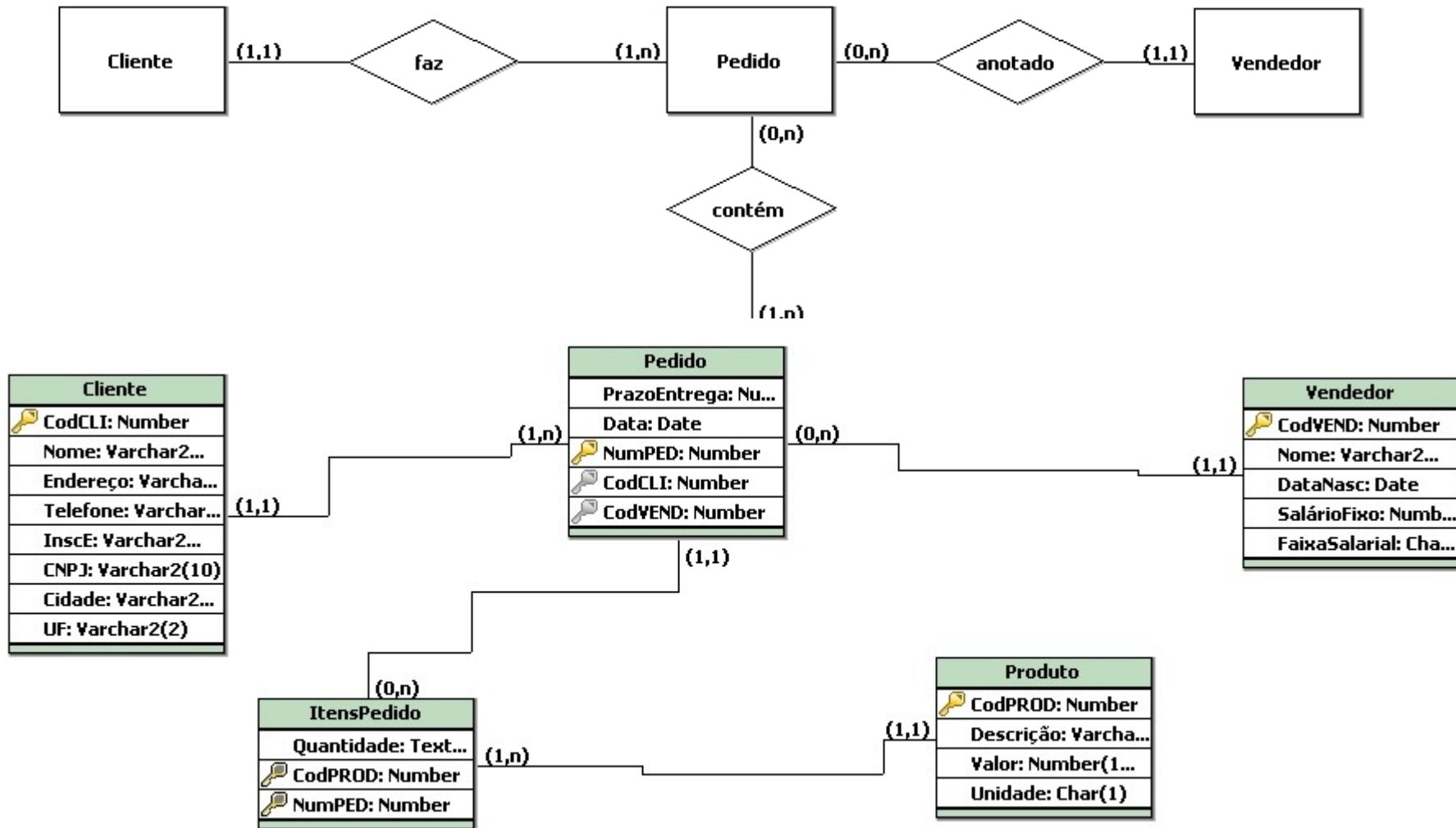
Data Output			Explain	Messages	His
	id integer	nome character varying(20)			
1	1	X			
2	1	X			
3	1	X			
4	1	X			

????

Scripts SQL

- Um **script SQL** é um conjunto de comandos SQL salvos como um arquivo SQL
- Pode-se usar Scripts SQL para criar, editar, visualizar, executar e excluir registros
- Vamos criar um banco para uma aplicação
exemplo: Pedidos
 - Requisitos:
 - Cadastrar Pedido
 - Cadastrar Cliente
 - Cadastrar Vendedor
 - Cadastrar Produto

Aplicação Exemplo 1



Principais Tipos de Dados no PostgreSQL

Tipo	Descrição	Alcance
smallint	[numérico] número inteiro pequeno alcance	-32.768 a 32.767
integer	[numérico] escolha típica para inteiro	-2147483648 a 2147483647
bigint	[numérico] número inteiro grande gama-	-9223372036854775808 para 9223372036854775807
numeric	[numérico] precisão especificada pelo usuário, exato	até 131072 dígitos antes do ponto decimal; até 16383 dígitos depois do ponto decimal
real	[numérico] precisão variável, inexata	precisão 6 dígitos decimais
serial	[numérico] integer autoincrementável	1-2147483647
Money	[monetário] valor em moeda, precisão fixa	-92233720368547758,08 para +92233720368547758,07

Principais Tipos de Dados no PostgreSQL

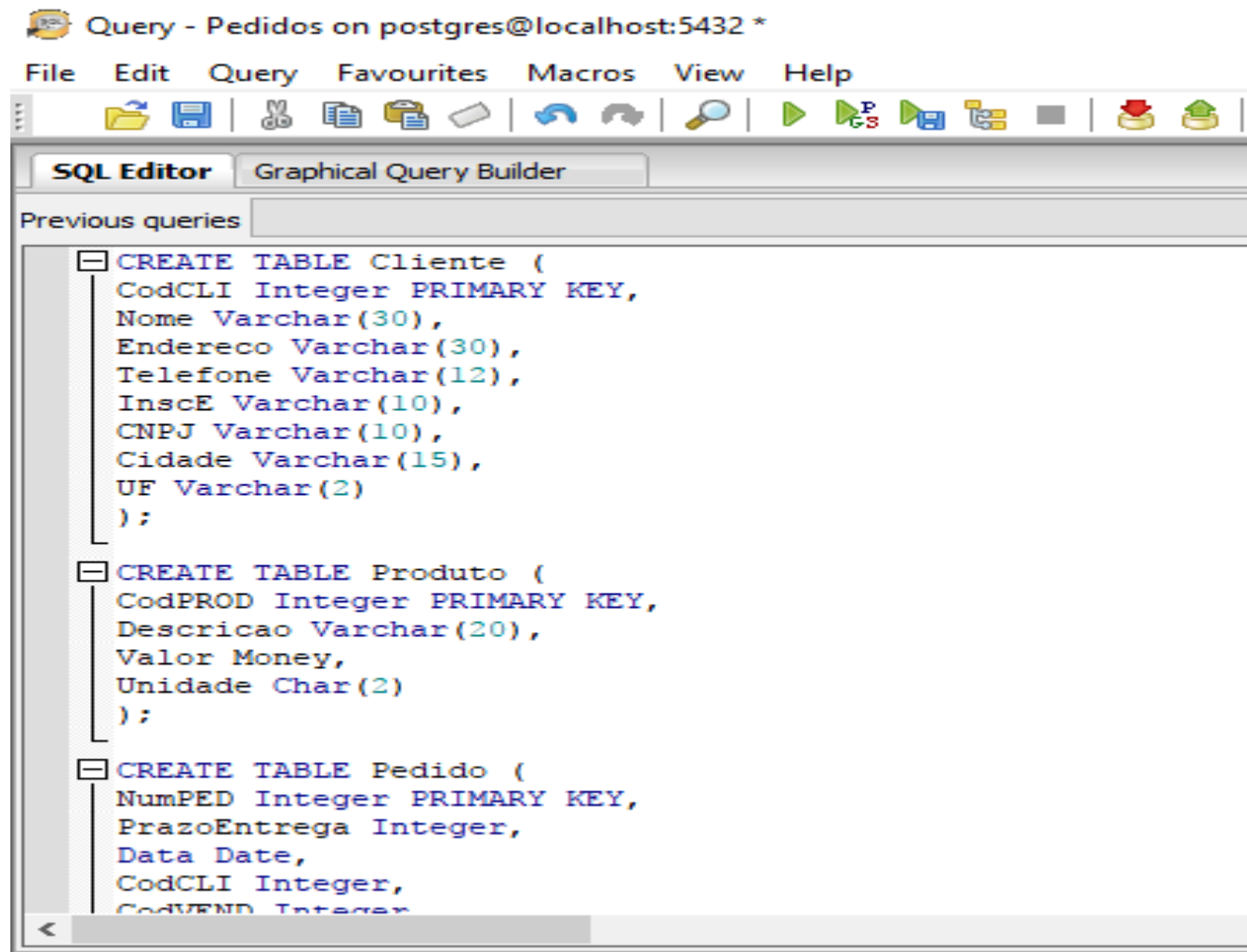
Tipo	Descrição
Varchar(n)	[caracter] comprimento variável com limite
Char(n)	[caracter] comprimento fixo
text	[caracter] comprimento ilimitado variável
date	[data] somente data
timestamp [(p)] [sem fuso horário]	[data] tanto a data e hora
timestamp [(p)] com fuso horário	[data] tanto a data e a hora, com fuso horário
booleano	estado de verdadeiro ou falso
point	[geométrico] (x, y)

Scripts de Criação e População


1. CriarPedidos.sql
2. Inscliente.sql
3. Insvendedor.sql
4. Insproduto.sql
5. Inspedido.sql
6. Insitenspedido.sql

**** Observar ordem para criação e população!!**

Query Tool



Schema

Properties	Statistics	Dependencies	Dependents
Schema	Owner	Comment	
 public	postgres	standard public schema	

**Um esquema é um espaço de nomes:
contém objetos com nome (tabelas, tipos de dado,
funções, views...)**

Quais tabelas existem no esquema criado??

Em SQL

■ CREATE SCHEMA nome_do_esquema
[AUTHORIZATION nome_do_usuario]
[elemento_do_esquema [...]] CREATE SCHEMA
AUTHORIZATION nome_do_usuario
[elemento_do_esquema [...]]

Exemplo

■ CREATE SCHEMA hollywood

CREATE TABLE films (title text, release date, awards text[])

CREATE VIEW winners AS SELECT title, release FROM
films WHERE awards IS NOT NULL;

Ou

CREATE SCHEMA hollywood;

CREATE TABLE hollywood.films (title text, release date,
awards text[]);

CREATE VIEW hollywood.winners AS SELECT title, release
FROM hollywood.films WHERE awards IS NOT NULL;

Voltando à Aplicação Pedidos, faça:

1. Insira três vendedores, três clientes e três pedidos
2. Atualize um dos produtos quanto a seu valor
3. Atualize o salário de todos os vendedores
4. Apague um dos vendedores cadastrados

** Use a ferramenta de consultas e o editor SQL do PostgreSQL!

Verifique

```
Select numped, codprod, quantidade  
  From itenspedido  
  Where quantidade > 20;
```

```
Select descricao  
  From produto  
  Where unidade = 'KG' and valor > 4.0
```

```
Select codprod, descricao  
  From produto  
  Where valor between 1.00 and 5.50;
```


Verifique

Select nome

From vendedor

Where faixacomissao IN ('A','B');

Select nome

From cliente

Where InscE is null;

Select nome, (salariofixo *1.15) + 120 as "Novo
Salário"

From vendedor

Where faixacomissao = 'C'

Order by nome;

Verifique

Select *
From Cliente
Where nome like '_____A';

Select *
From Cliente
Where nome like '_____a'

Verifique

```
select count(distinct(faixacomissao))  
from vendedor;
```

```
select faixacomissao, count(*)  
from vendedor  
group by faixacomissao;
```

```
Select numped, count(*) as "Total de Produtos"  
from itenspedido  
group by numped;
```

Verifique

```
select faixacomissao, min(Salariofixo),  
       max(Salariofixo)  
from vendedor  
where nome like 'J%' or nome like 'A%'  
group by faixacomissao  
having count(*) > 1;
```

Verifique

```
Select nome  
  from vendedor  
UNION  
Select nome  
  from cliente;
```

Atividade

```
CREATE TABLE moneyExemplo (  
    moneyCash money,  
    numericCash numeric(10,2));
```

```
INSERT INTO moneyExemplo VALUES (12.24, 12.24);
```

```
SELECT * FROM moneyExemplo;
```

```
SELECT '12.34'::numeric::money;
```

```
drop table moneyExemplo;
```

	moneycash money	numericcash numeric(10,2)
1	R\$12, 24	12.24

Dúvidas?

Alanamm.prof@gmail.com