

# Curso prático de instalação e utilização do PostgreSQL em máquinas locais

Este documento visa apresentar, de uma forma sumária, como instalar a base de dados **PostgreSQL** em computadores pessoais e não dispensa a leitura atenta dos conteúdos leccionados nas aulas teóricas.

Para ser possível desenvolver no Lazarus IDE aplicações que acedam a base de dados relacionais como o PostgreSQL, é necessário instalar um conjunto de bibliotecas (as famosas DLL em Windows), que vão permitir às aplicações Lazarus comunicar com a base de dados.

Neste documento é apresentado o processo de instalação e configuração de um servidor de base de dados PostgreSQL em computadores pessoais, o qual pode ser utilizado pelos alunos em alternativa ao serviços de base de dados disponibilizado pela FEUP em db.fe.up.pt.

Docentes: Andry Maykol Pinto, José Faria e Paulo Portugal

## Conteúdos

Parte 1 – Instalar o Tutorial de PostgreSQL .....	2
Como instalar o PostgreSQL na máquina .....	3
Atualizar as variáveis do sistema: .....	5
Parte 2 – Usar o PostgreSQL .....	8
Configurar e utilizar um servidor de base de dados PostgreSQL na máquina local.....	8
Criar um novo servidor para a base de dados.....	9
Criar as tabelas para a base de dados .....	10
Povoar as tabelas da base de dados .....	15
Pesquisar nas tabelas da base de dados .....	17
Consultar e manipular as base de dados .....	18
Exercicio 1 - Teste e corrija se necessário os seguintes comandos SQL: .....	18
Exercicio 2- Consultas às bases de dados (crie os comandos SQL para obter os seguintes resultados).....	19
Exercicio 3 – Editar as bases de dados (crie os comandos SQL para obter os seguintes resultados).....	21

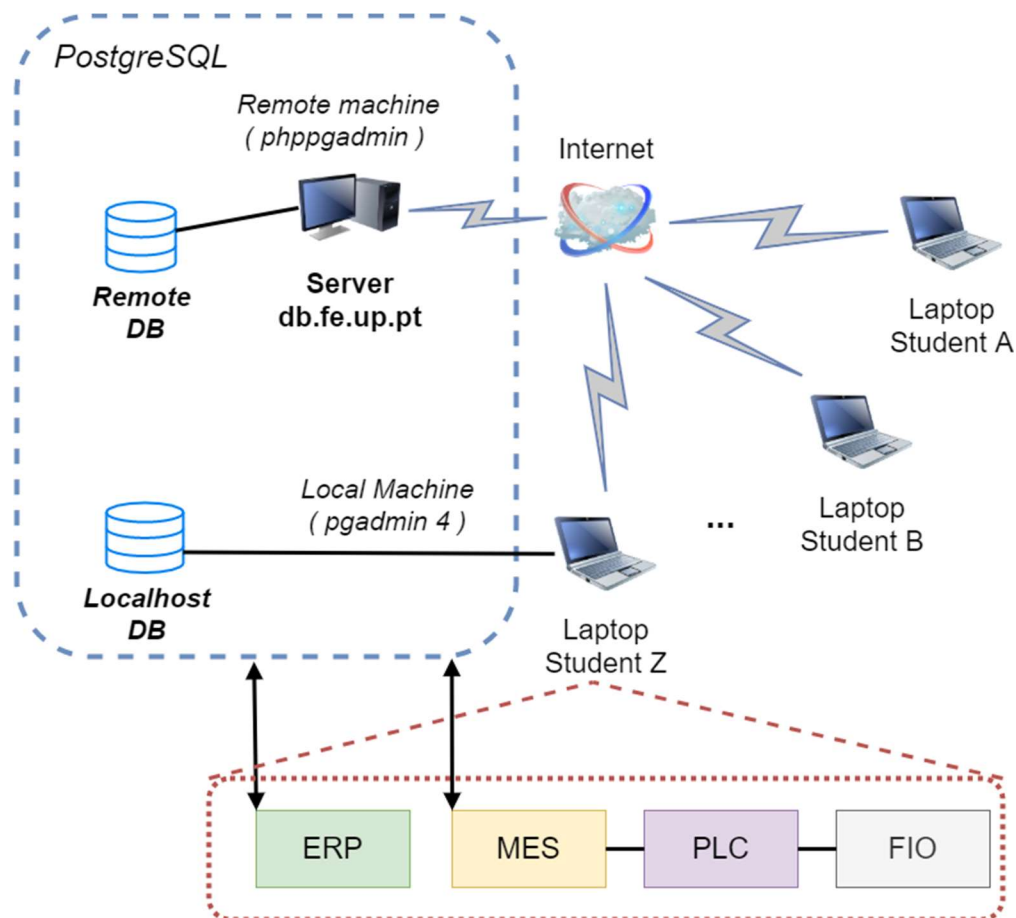
## Parte 1 – Instalar o Tutorial de PostgreSQL

Computadores com o sistema operativo Windows10 não possuem o PostgreSQL instalado nativamente e, portanto, torna-se necessário proceder à sua instalação quando se pretende desenvolver aplicações no Lazarus IDE que utilizam base de dados.

No trabalho prático, os estudantes vão poder utilizar uma base de dados PostgreSQL instalada localmente no seu próprio computador (tópico central deste tutorial) ou a base de dados remota que é disponibilizada pela FEUP e está acessível em: db.up.pt. A vantagem da utilização de um servidor local face ao servidor da FEUP reside no facto de não necessitar de ligação à rede da FEUP por VPN o que, nesta fase, é uma condição a evitar sempre que possível.

De notar que versões local e remota do PostgreSQL têm funcionalidades idênticas e recorrem a ferramentas de gestão semelhantes entre si, que possibilitam a interação e a configuração interna de tabelas, esquemas, acessos, etc: phpPgAdmin (no caso remoto) e phAdmin 4 (na instalação local).

Durante a elaboração do seu trabalho prático de INFI, vai poder alterar facilmente as configurações dos componentes que permitem o acesso ao PostgreSQL para, em qualquer momento, poder comutar entre a utilização da base de dados local ou remota.



Como as arquitecturas dos PCs são hoje em dia, na sua maioria de 64 bits, torna-se necessário instalar as bibliotecas do PostgreSQL x64. Após essa instalação, será necessário atualizar as variáveis de ambiente do Windows.

Este tutorial irá considerar o sistema operativo de Windows 10 e a versão do instalador PostgreSQL (*postgresql-12.2-1-windows-x64*).

### Como instalar o PostgreSQL na máquina

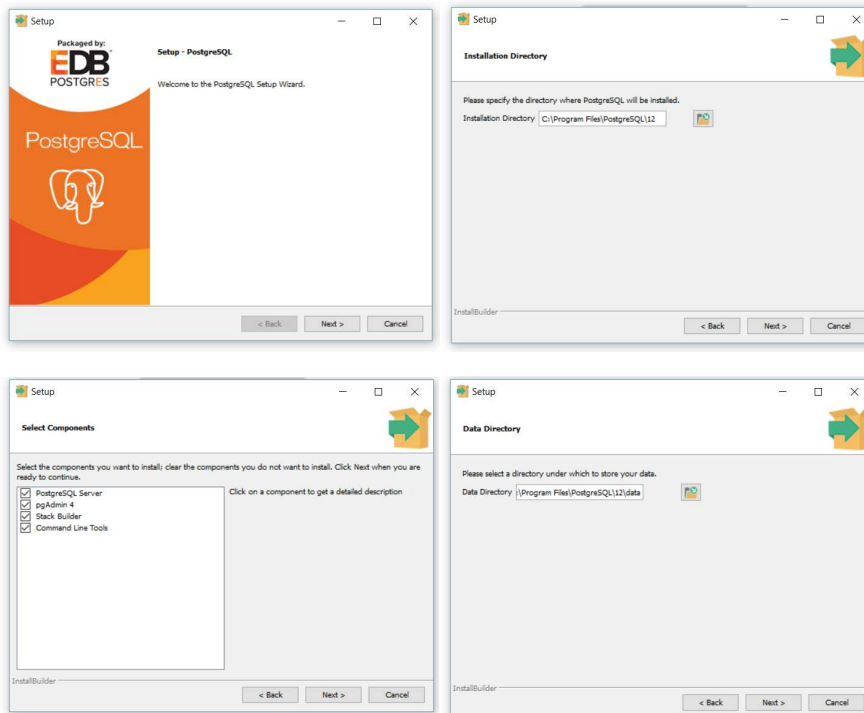
- 1) Fazer o download do instalador do PostgreSQL em:

Devem usar o instalador disponível no moodle ([link direto](#)).

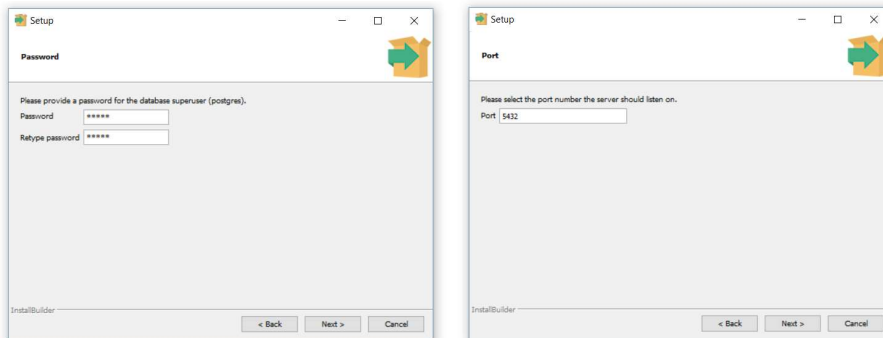
- 2) Executar normalmente o instalador do PostgreSQL. Não é necessário realizar muitas configurações especiais, bastando seguir o botão (“Next”) na maioria das vezes.

O diretório onde o PostgreSQL (versão 12) será instalado é o seguinte:

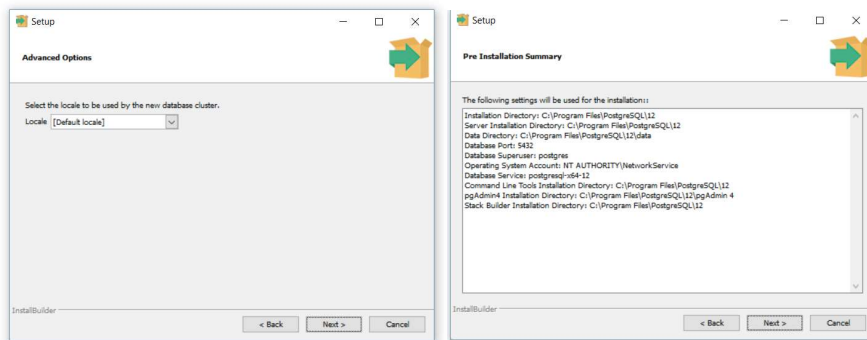
**C:\Program Files\PostgreSQL\12**



Introduza a palavra passe: “**admin**” e confirme que o “Port number” é o número **5432**

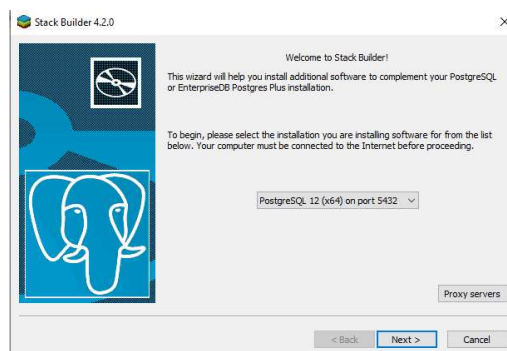


Basta confirmar agora a instalação:

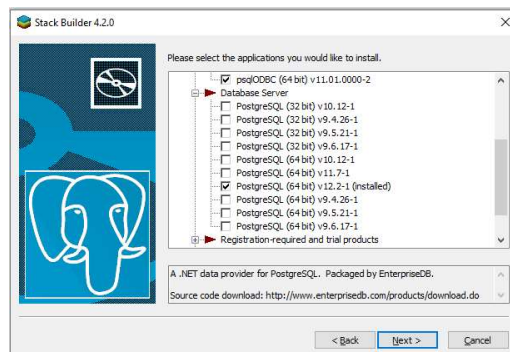
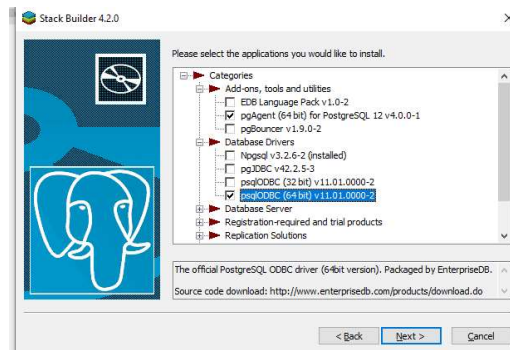


(*opcional!*) Após a instalação, pode executar o “**Stack Builder**” para instalar alguns extras:

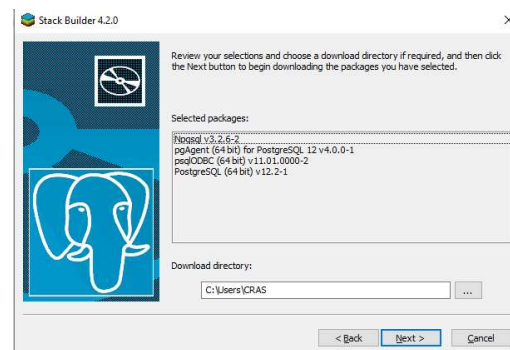
Selecione a versão do PostgreSQL que foi instalado.



Selecione os extras, *drivers* e *database servers* que deseja instalar.



Continue para fazer o download e para instalar:

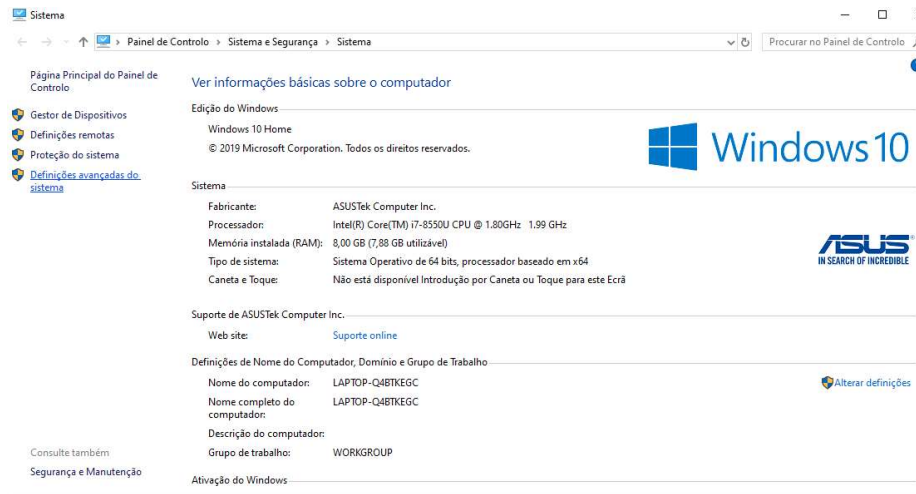


Atualizar as variáveis do sistema:

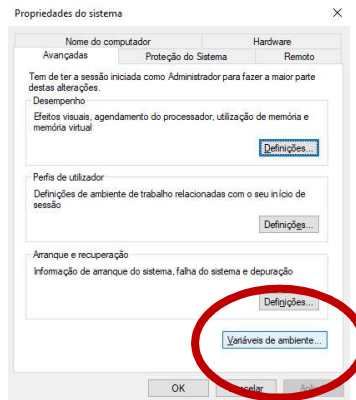
Esta é uma etapa necessária para a programação em Lazarus IDE, caso contrário as consultas à base de dados não irão funcionar (de fato, o programa irá dar um erro sempre que tenta fazer uma conexão à base de dados).

No “Meu computador” deve clicar com o botão lado direito e abrir as “Propriedades”.

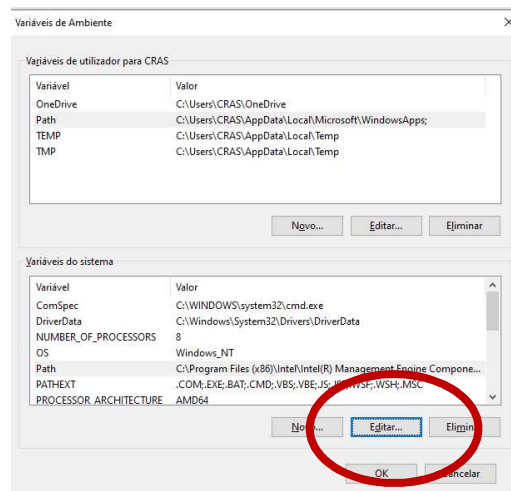
Em seguida, deve clicar em “definições avançadas”:



Clicar em “variáveis de ambiente”:



Nas “variáveis do sistema”, deve “Editar” o “Path”:



Acrescente dois campos:

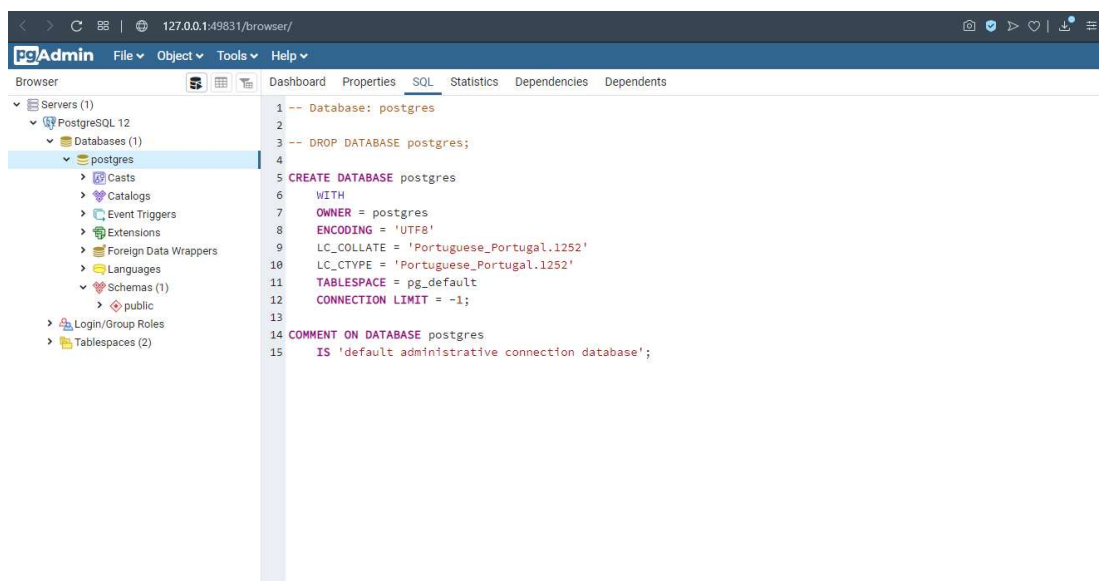


## Parte 2 – Usar o PostgreSQL

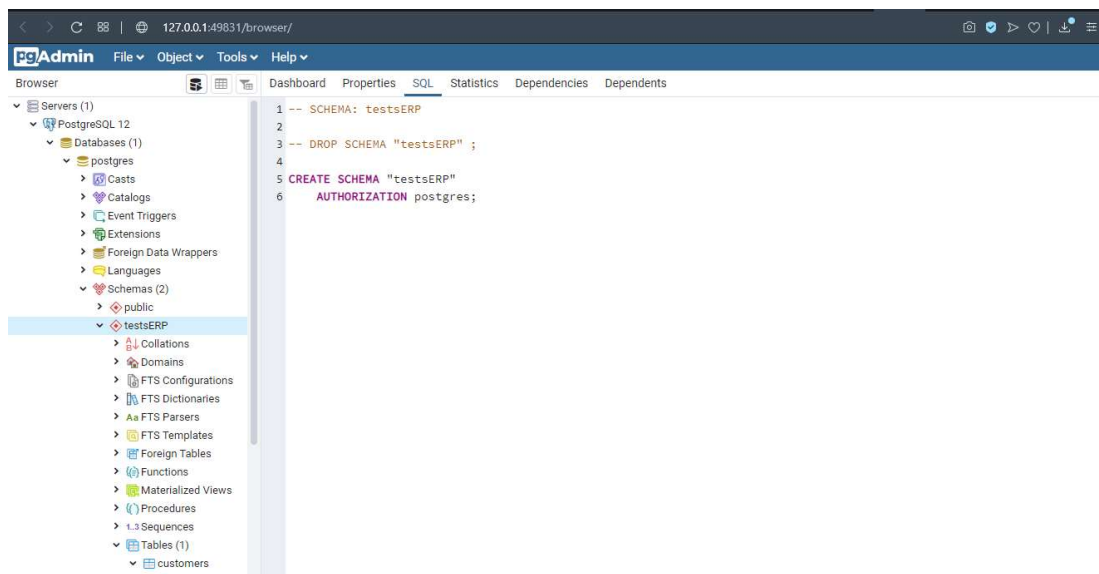
### Configurar e utilizar um servidor de base de dados PostgreSQL na máquina local

Após esta instalação, pode já executar o programa “**pgAdmin 4**” pois já está disponível no seu PC). Este programa permite configurar servidores de base de dados do tipo PostgreSQL. Em cada servidor que configure pode adicionar as Tabelas (consulte as aulas teóricas de SQL), definir alguns parâmetros de configuração e os acessos a utilizadores da base de dados (*login* e palavras chave).

Exemplo de um servidor “PostgreSQL 12” com uma base de dados “postgres”:



Ao expandir a árvore dos esquemas, verifica-se que existe um esquema “testsERP” que contém as tabelas desejadas.

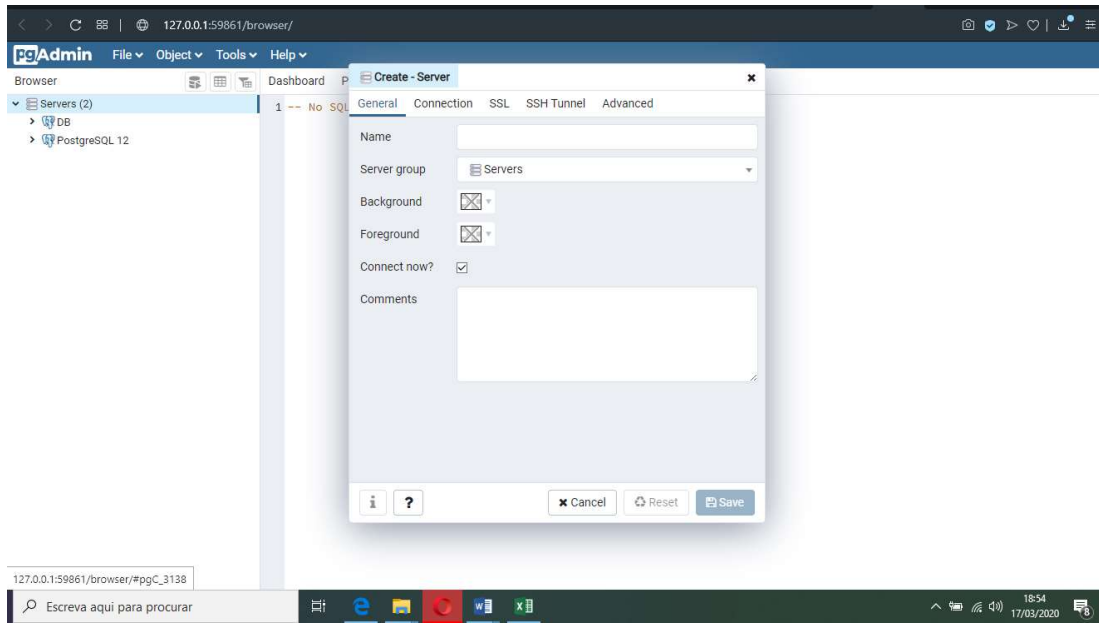




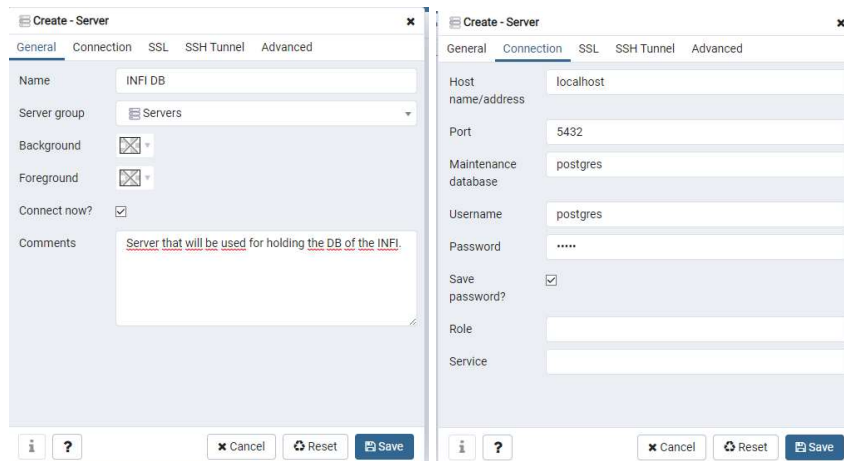
Criar um novo servidor para a base de dados

Claro que, na primeira vez que usa o pgAdmin, ele surgirá com um outro aspeto pois necessita de configurar o servidor de acordo com as suas necessidades e/ou especificidades do projeto.

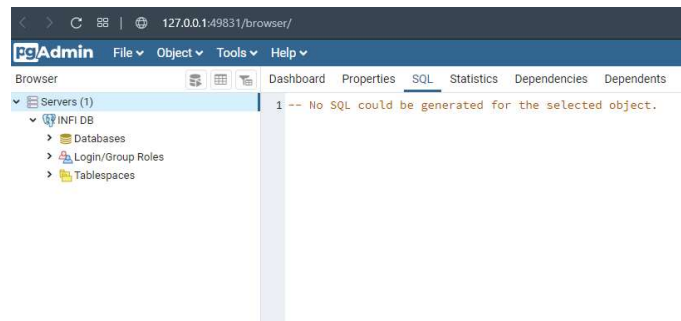
Muito sumariamente, deve clicar com o botão direito do rato em “Servers” e fazer “Create”.



Percorra as TABs (*General* e *Connection*) e preencher os campos: *Name*, *Comments*, *Host name/address*, *Port*, *Maintenance database*, *Username* e *Password*.

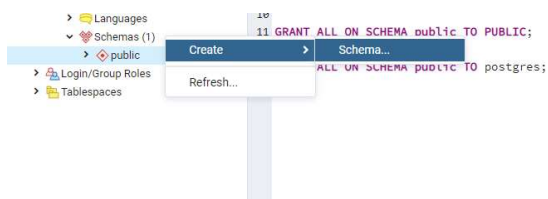


Para criar o servidor especificado basta fazer “Save”, e em seguida pode consultar o seu estado (clique com o botão do lado direito sobre o servidor criado).



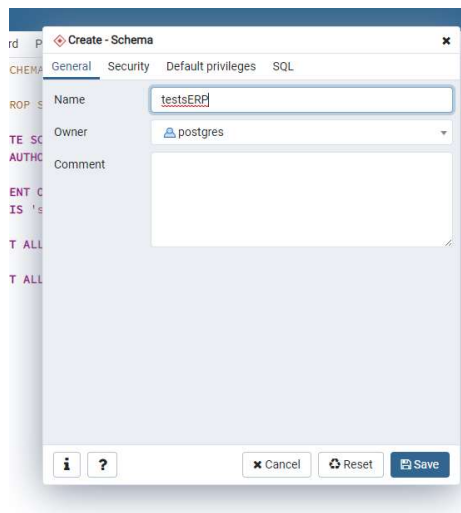
Criar as tabelas para a base de dados

Crie agora um novo esquema, clicando com o botão do lado direito do rato em “Schemes”:

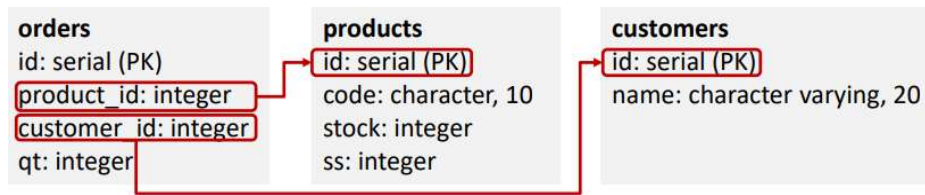


Na janela de configuração, preencha o nome do esquema: “testsERP”.

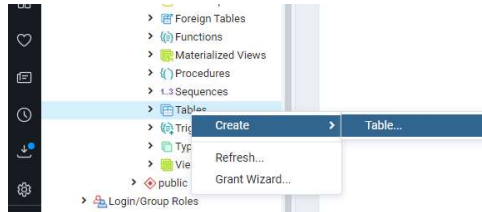
**Cuidado:** a utilização de letras maiúsculas irá fazer com que seja necessário utilizar aspas, por exemplo, no `SELECT * FROM "testsERP".customers`



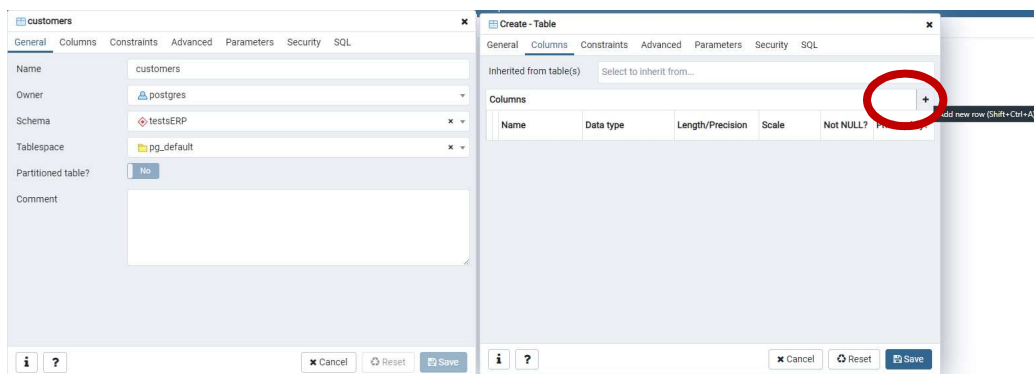
Vamos agora criar algumas tabelas exemplo:



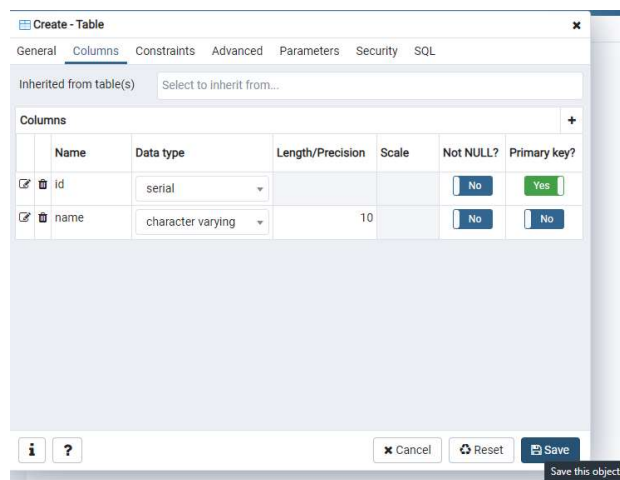
Agora, é só criar as tabelas da base de dados que necessita, para isso:



Na TAB *General*, pode definir o nome da tabela no campo do **Name** e na TAB *Columns*, pode definir as colunas clicando no botão “+”.



Criar a Tabela “customers”:



Criar a Tabela “products”:

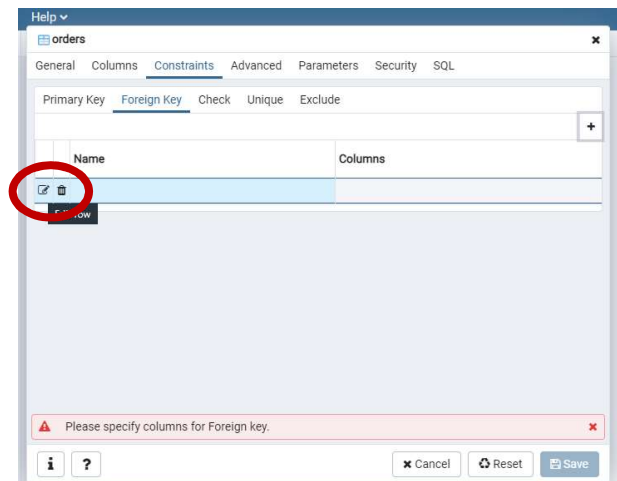
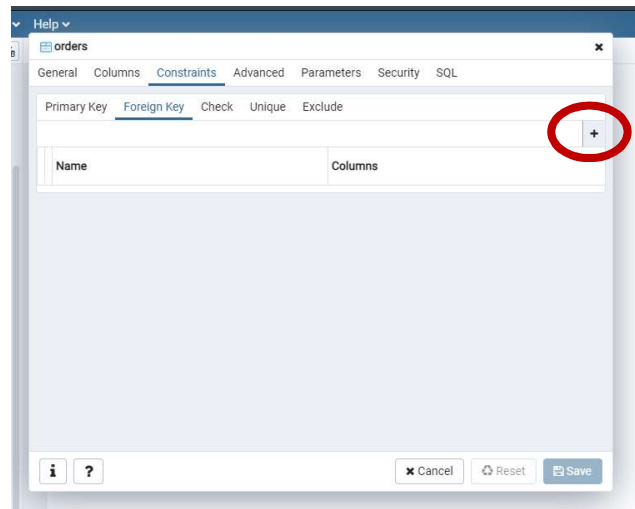
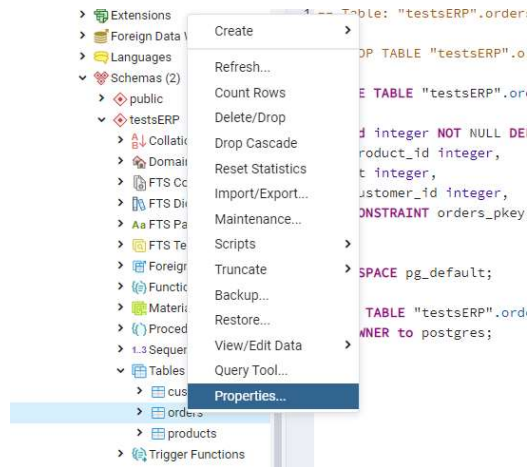
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id	serial			No	Yes
<input checked="" type="checkbox"/>	code	character	2		No	No
<input checked="" type="checkbox"/>	stock	integer			No	No
<input checked="" type="checkbox"/>	ss	integer			No	No

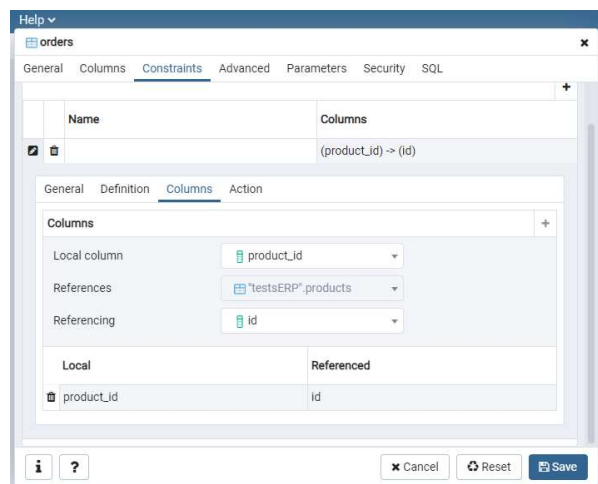
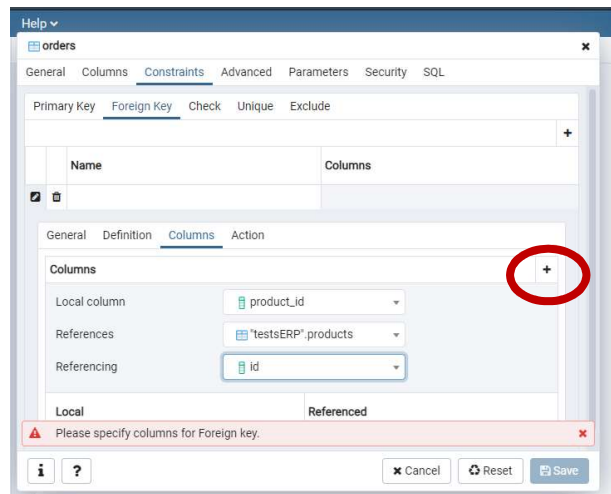
Criar a Tabela “orders”:

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id	serial			No	Yes
<input checked="" type="checkbox"/>	product	integer			No	No
<input checked="" type="checkbox"/>	customer	integer			No	No
<input checked="" type="checkbox"/>	qt	integer			No	No

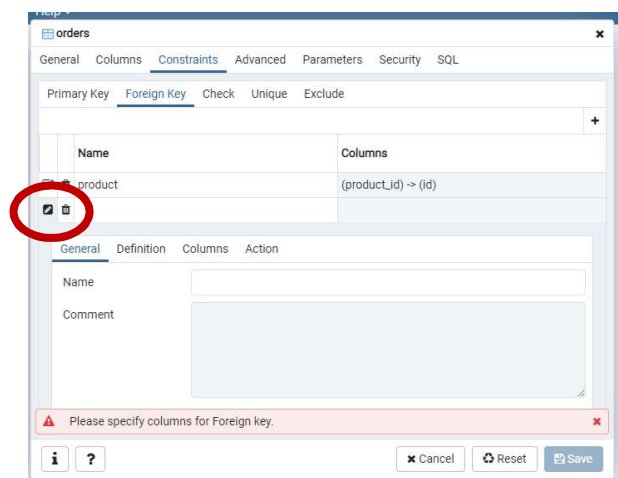
Para definir as “Chaves Estrangeiras” basta ir à TAB *Constraints* e definir o seguinte (através do clique no “+” seguido do “edit row”).

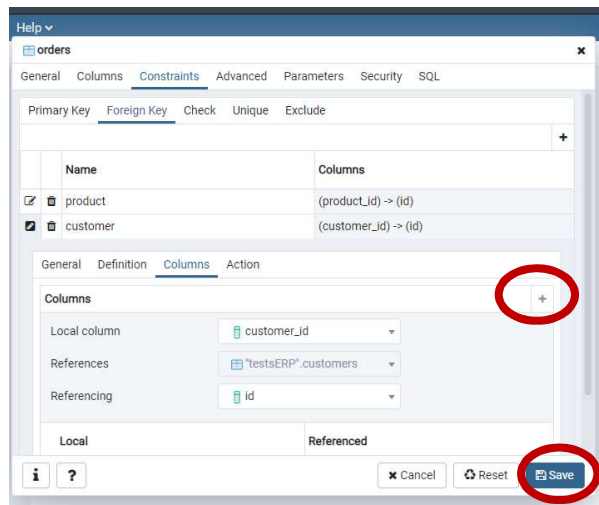
Vamos definir a chave estrangeira “product\_id”.





A mesma coisa para a chave estrangeira "customer\_id".





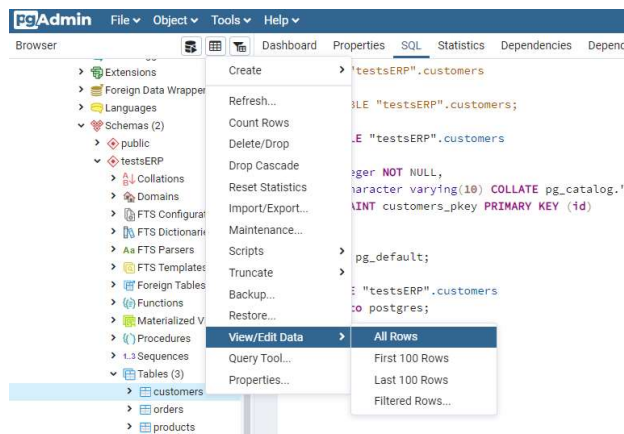
Em SQL para o PgAdmin, a tabela “orders” fica:

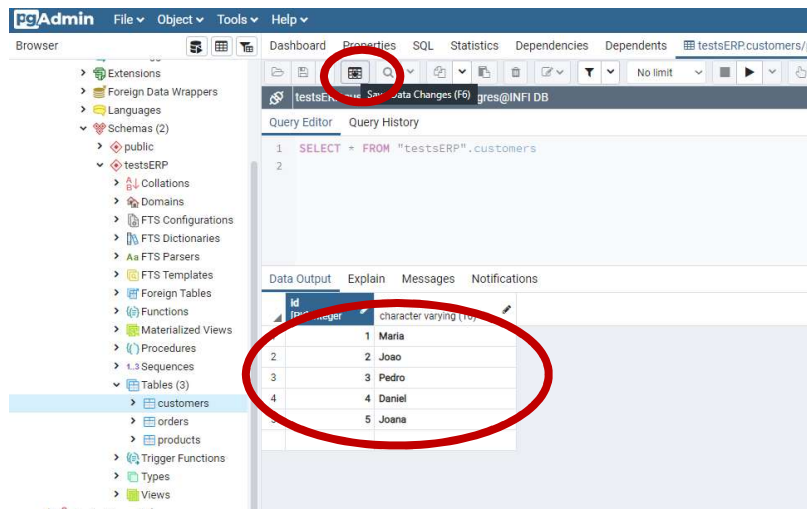
```

2
3 -- DROP TABLE "testsERP".orders;
4
5 CREATE TABLE "testsERP".orders
6 (
7   id integer NOT NULL DEFAULT nextval('"testsERP".orders_id_seq'::regclass),
8   product_id integer,
9   qt integer,
10  customer_id integer,
11  CONSTRAINT orders_pkey PRIMARY KEY (id),
12  CONSTRAINT customer FOREIGN KEY (customer_id)
13    REFERENCES "testsERP".customers (id) MATCH SIMPLE
14    ON UPDATE NO ACTION
15    ON DELETE NO ACTION
16    NOT VALID,
17  CONSTRAINT product FOREIGN KEY (product_id)
18    REFERENCES "testsERP".products (id) MATCH SIMPLE
19    ON UPDATE NO ACTION
20    ON DELETE NO ACTION
21    NOT VALID
22 )
23
  
```

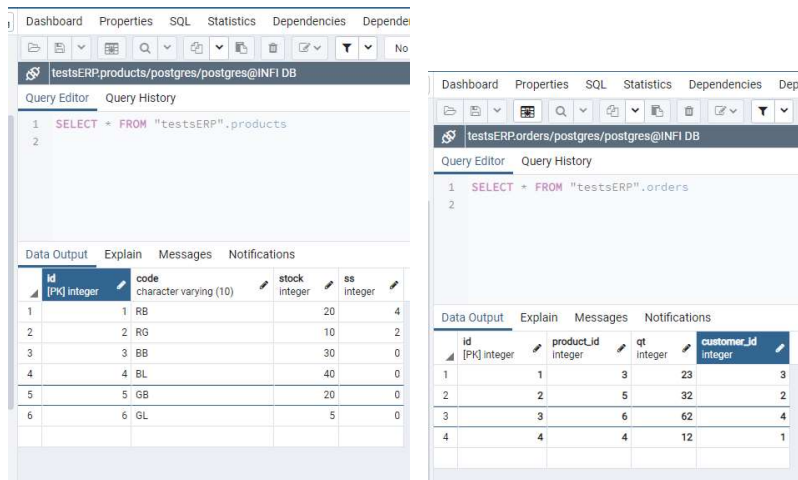
Povoar as tabelas da base de dados

Para inserir os elementos manualmente, deve ir a “View/Edit Data” na tabela pretendida e preencher adequadamente a matriz que aparece. No final deve guardar os dados (pressionando o F6 do teclado ou no botão assinalado).





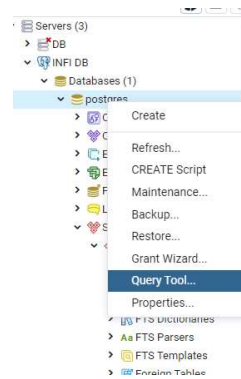
Replicar o processo para as tabelas "orders" e "products":





Pesquisar nas tabelas da base de dados

Experimente agora realizar as seguintes consultas à base de dados (clique com o botão direito do rato conforme indicado na imagem e seleccione o “*Query Tool...*”):



Consultar e manipular as base de dados

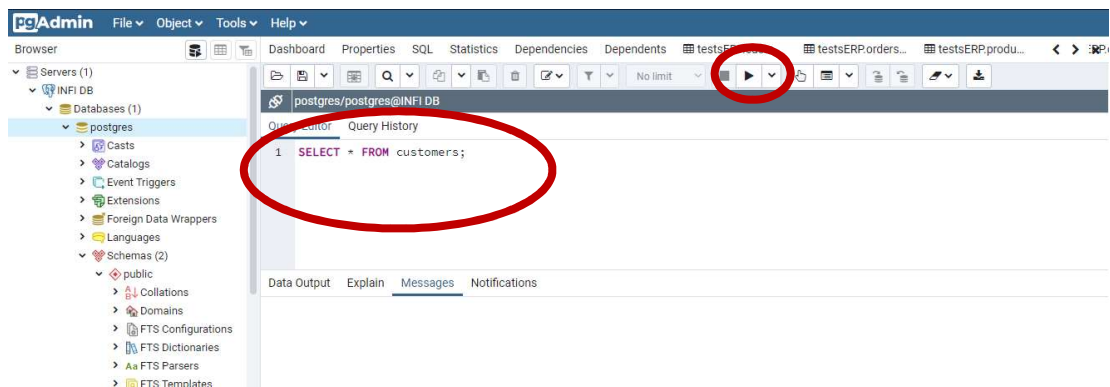
Exercicio 1 - Teste e corrija se necessário os seguintes comandos SQL:

*SELECT \* FROM customers;*

*SELECT \* FROM products;*

*SELECT \* FROM orders;*

*Nota: no final da aula, experimente no servidor da FEUP e justifique os resultados obtidos.*



Exercicio 2- Consultas às bases de dados (crie os comandos SQL para obter os seguintes resultados).

From "testsERP".orders:

	id integer	product_id integer	qt integer	customer_id integer	id integer	code character varying (10)	stock integer	ss integer
1	1	3	23	3	3	BB	30	0
2	2	5	32	2	5	GB	20	0
3	3	6	62	4	6	GL	5	0
4	4	4	12	1	4	BL	40	0

Successfully run. Total query runtime: 63 msec. 4 rows affected.

From "testsERP".orders:

	id integer	product_id integer	qt integer	customer_id integer	id integer	name character varying (10)
1	1	3	23	3	3	Pedro
2	2	5	32	2	2	Joao
3	3	6	62	4	4	Daniel
4	4	4	12	1	1	Maria

Successfully run. Total query runtime: 54 msec. 4 rows affected.

From "testsERP".orders:

	id [PK] integer	qt integer
1	1	23
2	2	32
3	3	62
4	4	12

Successfully run. Total query runtime: 88 msec. 4 rows affected.

From "testsERP".orders:

Data Output Explain Messages Notifications

	Product Type character varying (10)	Available Stock integer	Name of client character varying (10)	Quantity for Client integer
1	BB	30	Pedro	23
2	GB	20	Joao	32
3	GL	5	Daniel	62
4	BL	40	Maria	12

✓ Successfully run. Total query runtime: 51 msec. 4 rows affected.

Where 'Pedro':

Data Output Explain Messages Notifications

	Product Type character varying (10)	Available Stock integer	Name of client character varying (10)	Quantity for Client integer
1	BB	30	Pedro	23

✓ Successfully run. Total query runtime: 60 msec. 1 rows affected.

Exercicio 3 – Editar as bases de dados (crie os comandos SQL para obter os seguintes resultados).

Adicionar um novo produto 'QA' com stock de 55 e sem stock de segurança.

Data Output						Explain	Messages	Notifications
id [PK] integer	code character varying (10)	stock integer	ss integer					
1	RB	20	4					
2	RG	10	2					
3	BB	30	0					
4	BL	40	0					
5	GB	20	0					
6	GL	5	0					
7	QA	55	0					

Adicionar um novo cliente “Leandro”

Data Output		Explain	Messages	Notificat
id [PK] integer	name character varying (10)			
1	Maria			
2	Joao			
3	Pedro			
4	Daniel			
5	Joana			
6	Leandro			

Atualizar a ordem do cliente 1 para 10 peças do tipo 6.

Data Output					Explain	Messages	Notifications
id [PK] integer	product_id integer	qt integer	customer_id integer				
1	1	3	23	3			
2	2	5	32	2			
3	3	6	62	4			
4	4	4	12	1			

Apagar o cliente Leandro.

Data Output		Explain	Messages	Notifications
id	[PK] Integer	name	character varying (10)	
1		1	Maria	
2		2	Joao	
3		3	Pedro	
4		4	Daniel	
5		5	Joana	

✓ Successfully run. Total query runtime: 56 msec. 5 rows affected.

Apagar o cliente Maria e discuta o resultado.

FIM.