

**Q4) Crie um banco de dados para o sistema de varejo apresentado na aula anterior (você pode utilizar os cluster e tablespace padrão do PostgreSQL ou pode criar específicos para o sistema).**

Pede-se para criar um banco de dados para o ecommerce criado na aula 02. A questão diz que podemos usar o padrão ou criar um outro cluster para nosso ecommerce.

Será criado um cluster novo para essa atividade.

Pelo terminal da minha máquina local, usei o seguinte comando para criar um cluster novo e inicializá-lo:

```
sudo pg_createcluster 14 clusterempresa --start
```

```
letonio@letonio-Inspiron-15-3567:~$ sudo pg_createcluster 14 clusterempresa
--start
[sudo] senha para letonio:
Creating new PostgreSQL cluster 14/clusterempresa ...
/usr/lib/postgresql/14/bin/initdb -D /var/lib/postgresql/14/clusterempresa -
-auth-local-peer --auth-host scram-sha-256 --no-instructions
Running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok
Ver Cluster      Port Status Owner      Data directory
Log file
14 clusterempresa 5433 online postgres /var/lib/postgresql/14/clusterempres
a /var/log/postgresql/postgresql-14-clusterempresa.log
letonio@letonio-Inspiron-15-3567:~$
```

Lembrando que 14 está relacionado com a versão do psql, “clusterempresa” é o nome do cluster e a flag “--start” inicia o cluster após ser criado.

Vamos confirmar que o cluster está ativo realmente:

```
systemctl status postgresql@14-clusterempresa
```

```
letonio@letonio-Inspiron-15-3567:~$ systemctl status postgresql@14-clusterem
presa
● postgresql@14-clusterempresa.service - PostgreSQL Cluster 14-clusterempres
Loaded: loaded (/lib/systemd/system/postgresql@.service; enabled-runtime; ve
Active: active (running) since Tue 2023-02-07 16:23:30 -03; 2min 56s ago
Process: 24532 ExecStart=/usr/bin/pg_ctlcluster --skip-systemctl-redirect=
Main PID: 24538 (postgres)
Tasks: 7 (limit: 9295)
Memory: 18.2M
CPU: 317ms
```

A imagem acima confirma o que era esperado, o cluster “clusterempresa” foi criado e está ativo.

Um cluster pode conter um ou vários banco de dados (database). Vamos criar dentro do “clusterempresa” nosso primeiro database, que sera chamado “dbempresa”.

O comando a seguir cria o banco de dados que queremos:

```
sudo -u postgres psql -c "CREATE DATABASE dbempresa;"
```

```
letonio@letonio-Inspiron-15-3567:~$ sudo -u postgres psql -c "CREATE DATABASE dbempresa;"
could not change directory to "/home/letonio": Permissão negada
CREATE DATABASE
letonio@letonio-Inspiron-15-3567:~$
```

O database foi criado.

A critério de informação, nosso cluster fica em “/etc/postgresql/14”

```
letonio@letonio-Inspiron-15-3567:~$ cd /etc/postgresql/14/
letonio@letonio-Inspiron-15-3567:/etc/postgresql/14$ ls
clusterempresa  main
letonio@letonio-Inspiron-15-3567:/etc/postgresql/14$
```

Vamos acessar dois arquivos de configuração, para permitir que nossa máquina local acesse o postgresql local via pgAdmin. A vantagem é que o pgAdmin possui interface gráfica que vai facilitar na hora de escrevermos o código das nossas entidades.

Para habilitar essa conexão, algumas mudanças são necessárias.

Acessando o arquivo presente em “/etc/postgresql/postgresql.conf/14/clusterempresa”, por meio do comando: “`sudo nano /etc/postgresql/14/clusterempresa/postgresql.conf`”

```
# - Connection Settings -

#listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                           # comma-separated list of addresses;
#                                           # defaults to 'localhost'; use '*' for all
#                                           # (change requires restart)
port = 5433                              # (change requires restart)
max_connections = 100                    # (change requires restart)
```

A linha #listen\_addresses precisa ser descomentada. Uma observação importante é que a porta está definida como port = 5433, isso acontece porque a porta padrão (5432) já está sendo ocupada pelo cluster main, que já vem como padrão quando instalamos o PostgreSQL.

O arquivo modificado fica assim:

```
# - Connection Settings -

listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                           # comma-separated list of addresses;
#                                           # defaults to 'localhost'; use '*' for all
#                                           # (change requires restart)
port = 5433                              # (change requires restart)
max_connections = 100                    # (change requires restart)
```

O próximo arquivo que precisa ser observado é:

`sudo nano /etc/postgresql/14/clusterempresa/pg_hba.conf`

Para que as alterações sejam aplicadas, faz-se necessário reiniciar o cluster, usando o comando:

“`sudo systemctl restart postgresql@14-clusterempresa`”

```
letonio@letonio-Inspiron-15-3567:/etc/postgresql/14/main$ sudo systemctl restart postgresql@14-clus
terempresa
letonio@letonio-Inspiron-15-3567:/etc/postgresql/14/main$ ^C
```

Para verificar se está ativo, usa-se:

“systemctl status postgresql@14-clusterempresa”

```
letonio@letonio-Inspiron-15-3567:/etc/postgresql/14/main$ systemctl status postgresql@14-clusteremp
resa
● postgresql@14-clusterempresa.service - PostgreSQL Cluster 14-clusterempresa
   Loaded: loaded (/lib/systemd/system/postgresql@.service; enabled-runtime; vendor preset: enab
   Active: active (running) since Tue 2023-02-07 17:23:27 -03; 2min 29s ago
```

Está ativo, agora vamos tentar acessar.

Dependendo se for máquina local ou remota, pode ser necessário adicionar uma linha dando permissão para o ip de uma máquina que está em outro lugar, no acesso local ipv4:

```
# TYPE      DATABASE    USER        ADDRESS          METHOD
# "local" is for Unix domain socket connections only
local      all         all          peer
# IPv4 local connections:
host       all         all          127.0.0.1/32     scram-sha-256
# IPv6 local connections:
host       all         all          ::1/128          scram-sha-256
# Allow replication connections from localhost, by a user with the
```

A próxima etapa é entrar no pgAdmin, criar um novo server, informar o host (localhost) e a porta (5433) e fazer o acesso, que nesse caso, será feito usando o usuario “postgres” que vem como padrão.

The image shows the 'Register - Server' dialog in pgAdmin. It has two tabs: 'General' and 'Connection'. In the 'General' tab, the 'Name' is 'serverLocal\_empresa' and the 'Server group' is 'Servers'. In the 'Connection' tab, the 'Host name/addresses' is 'localhost', the 'Port' is '5433', the 'Maintenance database' is 'postgres', and the 'Username' is 'postgres'.

Field	Value
Name	serverLocal_empresa
Server group	Servers
Host name/addresses	localhost
Port	5433
Maintenance database	postgres
Username	postgres

Meu postgres não tinha senha, então acessei e adicionei uma senha:

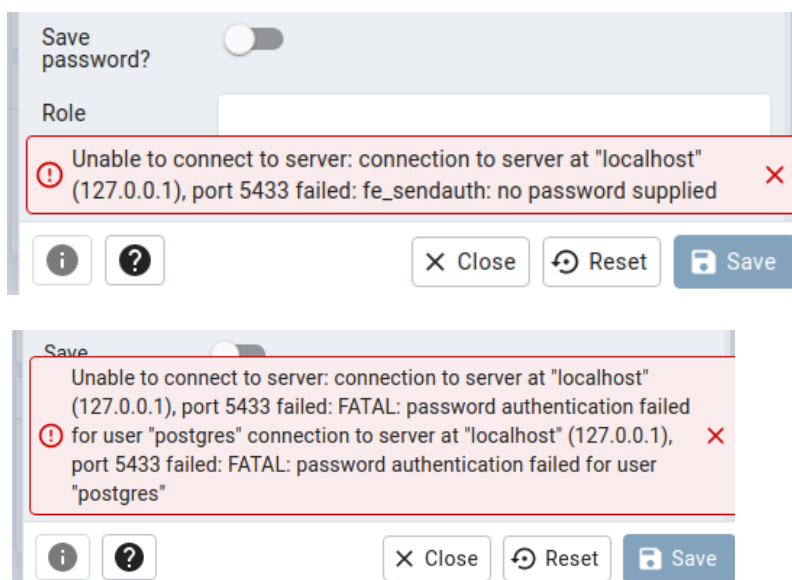
```

letonio@letonio-Inspiron-15-3567:/etc/postgresql/14/main$ sudo -i -u postgres
postgres@letonio-Inspiron-15-3567:~$ psql
psql (14.6 (Ubuntu 14.6-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# ALTER USER postgres WITH PASSWORD '54321'
postgres=#

```

Mesmo adicionando e tentando, apareceu erro:



Resolvi criar um outro usuário para tentar fazer o acesso:

Acessei o psql:

`sudo -u postgres psql`

Criei um novo usuário:

`CREATE USER letspg WITH PASSWORD 'letspg';`

Mudei o dono do database “dbempresa”, para ser de “letspg”:

`ALTER DATABASE dbempresa OWNER TO letspg;`

```

letonio@letonio-Inspiron-15-3567:/etc/postgresql/14/main$ sudo -u postgres psql
psql (14.6 (Ubuntu 14.6-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE USER letspg WITH PASSWORD 'letspg';
CREATE ROLE
postgres=# ALTER DATABASE dbempresa OWNER TO letspg;
ALTER DATABASE
postgres=#

```

Agora, vou tentar fazer a conexão com o banco de dados “dbempresa”, usando o usuário “letspg”, cuja senha é “letspg”

Não consegui fazer o acesso pela porta 5433, que foi a definida num arquivo de configuração.

Precisei usar a 5432, a imagem a seguir ilustra as coisas que adicionei:

General	Connection	SSL	SSH Tunnel	Advanced
<div> <div>Name</div> <div>server_local_Empresa</div> </div>				

### Register - Server

General	Connection	SSL	SSH
Host name/addresses			
localhost			
Port			
5432			
Maintenance database			
dbempresa			
Username			
letspg			
Kerberos authentication?			
<input type="checkbox"/>			
Password			
.....			

Finalmente, foi possível fazer a conexão:

File Object Tools Help

Browser

Servers (2)

serverLocal

server\_local\_Empresa

Databases (3)

aula03db

dbempresa

postgres

Login/Group Roles

Tablespaces

Dashboard Properties SQL Statistics Dependencies

Server sessions

Total

Active

6

5

4

3

2

1

0

Tuples in

Inserts

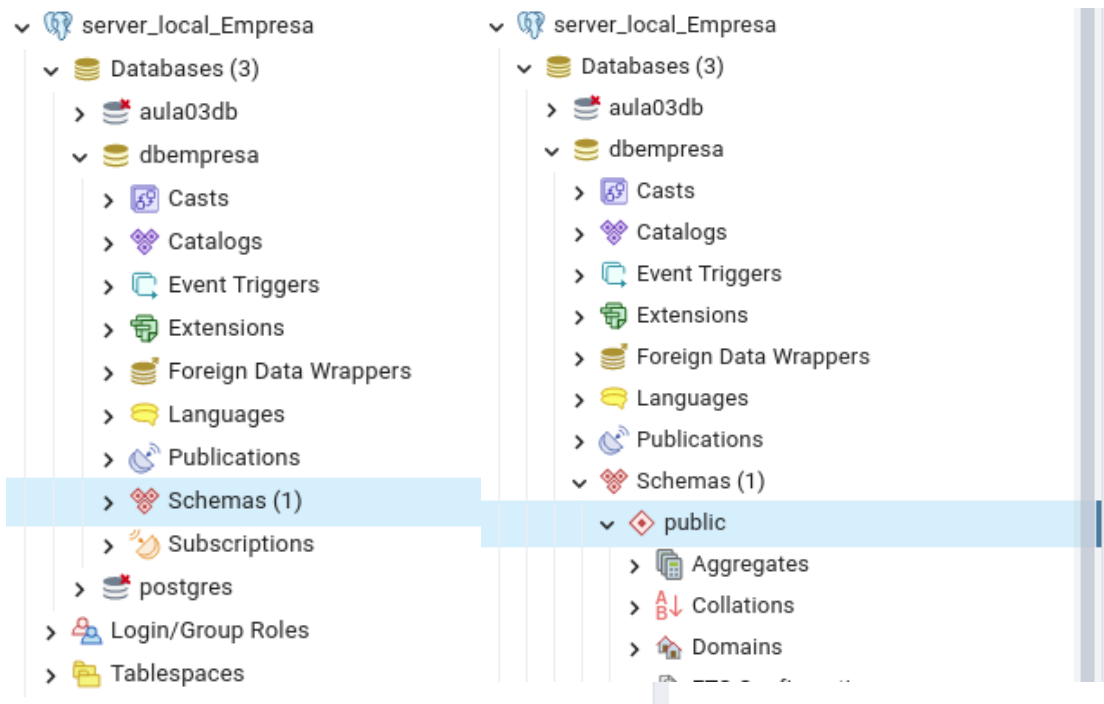
Updates

Deletes

1

1,400

Decidi por usar o esquema padrão, portanto, Vamos acessar dbempresa->Schemas->Public



Vamos clicar em public e adicionar os códigos na próxima questão (botão direito em public → Query Tool)!