

Objetivo:

- Copiar registros.

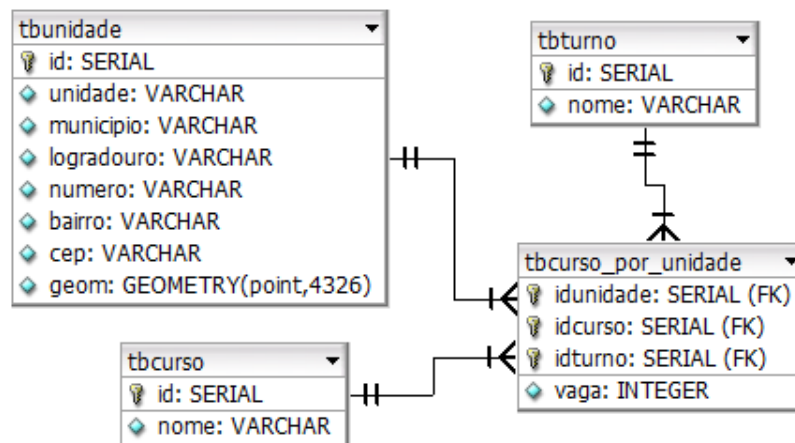
Gravação das explicações:

Aula 6 - Exercícios

<https://web.microsoftstream.com/video/a6b6489c-9edc-4d7f-ba59-6fb66ab31d1f>

Requisitos:

- Usar o pgAdmin 4 e QGIS;
- Ter um BD de nome bdaula no PostgreSQL com a extensão espacial PostGIS;
- Carregar o shapefile de municípios do país na tabela tbmunicipio;
- Use as cláusulas do arquivo Aula 5 - Clausulas SQL.txt para criar e preencher as tabelas a seguir.



Copiar registros:

A saída de uma cláusula select é projetada no dispositivo de saída. Até aqui, esse dispositivo foi a tela do computador. Porém, podemos inserir numa tabela os registros retornados pela cláusula select. Esse recurso possibilita salvar o resultado, uma vez que as tabelas do SGBD não são apagadas ao desligar o computador.

A cláusula insert possui a seguinte estrutura:

```

insert into nomeDaTabela (nomeDaColuna1, nomeDaColuna2, nomeDaColuna3)
values
(valorDaColuna1, valorDaColuna2, valorDaColuna3),
(valorDaColuna1, valorDaColuna2, valorDaColuna3);
  
```

Então a saída de uma cláusula select com a mesma quantidade de colunas e tipos de dados pode ser encadeada.

Observe que nesse caso não se usa o termo values:

```

insert into nomeDaTabela (nomeDaColuna1, nomeDaColuna2, nomeDaColuna3)
select nomeDaColuna1, nomeDaColuna2, nomeDaColuna3
from nomeDaTabela;
  
```

Exemplo 1 – Copiar os endereços das Fatecs para a tbendereco.

```

drop table if exists tbendereco;
create table if not exists tbendereco(
  
```

```
id serial not null primary key,  
logradouro varchar,  
numero varchar,  
cep varchar,  
municipio varchar  
);
```

A ordem das colunas no select precisa ser a mesma do insert:

```
insert into tbendereco(id, logradouro, numero, cep, municipio)  
select id, logradouro, numero, cep, municipio  
from tbunidade;
```

A cláusula a seguir também está correta, pois a ordem das colunas do select é a mesma do insert:

```
insert into tbendereco(logradouro, numero, cep, municipio, id)  
select logradouro, numero, cep, municipio, id  
from tbunidade;
```

Exemplo 2 – Copiar as localizações da tbunidade para a tblocal. A geometria da tblocal possui o SRS (Sistema de Referência Espacial) SIRGAS 2000 UTM zona 23S SRID 31983.

```
drop table if exists tblocal;  
create table if not exists tblocal(  
id serial not null primary key,  
nome varchar,  
geom geometry(point, 31983)  
);
```

Para inserir os registros na tblocal teremos de converter de WGS 84 para o SRID de destino.

```
insert into tblocal(id, nome, geom)  
select id, unidade, st_transform(geom, 31983)  
from tbunidade;
```

Exercícios:

Acesse <https://postgis.net/docs/reference.html> para obter detalhes sobre as funções do PostGIS.

1 – Fazer uma consulta para listar as unidades que possuem o curso de Gestão Financeira. Evitar registros repetidos e carregar o resultado na tabela tbgestao.

A tbgestao terá 3 registros.

```
drop table if exists tbgestao;  
create table if not exists tbgestao(  
id serial not null primary key,  
unidade varchar,  
geom geometry(point, 4326)  
);
```

Dica: use o termo distinct para evitar repetições.

2 – Alterar a consulta do Exercício 1 para carregar o resultado na tbgestaometro.

A tbgestaometro terá 3 registros.

```
drop table if exists tbgestaometro;
create table if not exists tbgestaometro(
    id serial not null primary key,
    unidade varchar,
    geom geometry(point,31983)
);
```

Dica: use a função st_transform.

3 – Fazer uma consulta para listar os municípios que fazem interseção com alguma Fatec. Carregar o resultado na tbmunicipiofatec.

A tbmunicipiofatec terá 75 registros.

```
drop table if exists tbmunicipiofatec;
create table if not exists tbmunicipiofatec(
    like tbmunicipio
);
```

O termo `like tbmunicipio` é usado para criar uma tabela com as mesmas colunas e chave primária da tbmunicipio.

Dica: use a função st_intersects no termo where para fazer a ligação entre as tabelas.

4 – Alterar o Exercício 3 para obter o centroide dos municípios e carregar o resultado na tbcentroide.

A tbcentroide terá 75 registros.

```
drop table if exists tbcentroide;
create table if not exists tbcentroide(
    gid serial not null primary key,
    municipio varchar,
    geom geometry(point,4326)
);
```

Dica: use a função st_centroid.

5 – Fazer uma consulta para obter a distância, em metros, entre cada Fatec e a Fatec de Jacareí. Carregar o resultado na tbdistancia.

A tbdistancia terá 75 registros.

```
drop table if exists tbdistancia;
create table if not exists tbdistancia(
    id serial not null primary key,
    unidade varchar,
    distancia float,
    geom geometry(point,31983)
);
```

Dicas:

- Use a tbunidade duas vezes no termo from;
- Use a função st_transform para converter as geometrias de WGS 84 para SIRGAS 2000 UTM zona 23S, SRID 31983.

6 – Fazer uma consulta para obter os municípios da mesorregião do VALE DO PARAIBA PAULISTA. Carregar o resultado na tbvale.

A tbvale terá 79 registros.

```
drop table if exists tbvale;  
create table if not exists tbvale(  
    like tbmunicipio  
);
```

7 – Fazer uma consulta para obter as Fatecs que estão na mesorregião do VALE DO PARAIBA PAULISTA. Carregar o resultado na tbfatecvale.

A tbfatecvale terá 7 registros.

```
drop table if exists tbfatecvale;  
create table if not exists tbfatecvale(  
    like tbunidade  
);
```

Dica: use a função st_intersects.

8 – Alterar a consulta do Exercício 7 para obter um buffer de 15 km no entorno de cada ponto. Carregar o resultado na tbentorno.

A tbentorno terá 7 registros.

```
drop table if exists tbentorno;  
create table if not exists tbentorno(  
    id serial not null primary key,  
    unidade varchar,  
    geom geometry(polygon,31983)  
);
```

Dica: use a função st_buffer e a função st_transform.

9 – Fazer uma consulta para obter as Fatecs que possuem curso EaD. Carregar o resultado na tbead.

A tbead terá 47 registros.

```
drop table if exists tbead;  
create table if not exists tbead(  
    id serial not null primary key,  
    unidade varchar,  
    geom geometry(point,4326)  
);
```

10 – Fazer uma consulta para obter o nome da unidade, curso e turno de todos os cursos. Carregar o resultado na tbtdo.

A tbtdo terá 453 registros.

```
drop table if exists tbtdo;  
create table if not exists tbtdo(  
    id serial not null primary key,  
    unidade varchar,  
    curso varchar,  
    turno varchar  
);
```