

Minicurso:

# Introdução à Análise de Dados Espaciais

---

Introdução à Extensão Espacial PostGIS



23 de Outubro de 2019  
São José dos Campos - SP

Adriano P. Almeida  
Felipe Carvalho  
Felipe Menino  
Helmécio Neto

# INTRODUÇÃO

## Armazenamento de dados



# INTRODUÇÃO

## Banco de dados

ID	MUNÍCIOPIO	ESTADO	POPULAÇÃO	PIB PER CAPITA (R\$)
1	São José dos Campos	SP	629.921	53.615,25
2	Boa Vista	RR	284.313	24.852,52
3	Belém	PA	1.492.745	20.352,00
4	São Paulo	SP	12.252.023	57.071,43
5	Itaituba	PA	97.493	16.261,21

Fonte: IBGE, 2010

# INTRODUÇÃO

## Banco de dados

ID	MUNICÍPIO	ESTADO	POPULAÇÃO	PIB PER CAPITA (R\$)
1	São José dos Campos	SP	629.921	53.615,25
2	Boa Vista	RR	284.313	24.852,52
3	Belém	PA	1.492.745	20.352,00
4	São Paulo	SP	12.252.023	57.071,43
5	Itaituba	PA	97.493	16.261,21

Fonte: IBGE, 2010

1. Qual a população média dos municípios dos estado do Pará?
2. Qual a ordenação decrescente (por estados) da população brasileira?
3. Quantos municípios do estado de São Paulo possui menos de 300 mil pessoas?
4. Quais municípios do estado de Roraima possui PIB per capita menor que a média do estado, e população superior a 180 mil habitantes ?

# INTRODUÇÃO

## Banco de dados

ID	MUNÍCIPIO	ESTADO	POPULAÇÃO	PIB PER CAPITA (R\$)
1	São José dos Campos	SP	629.921	53.615,25
2	Boa Vista	RR	284.313	24.852,52
3	Belém	PA	1.492.745	20.352,00
4	São Paulo	SP	12.252.023	57.071,43
5	Itaituba	PA	97.493	16.261,21

Fonte: IBGE, 2010

1. Qual a área do município mais populoso de Roraima?
2. Qual a distância entre os municípios com maior e menor PIB percapita do estado do Pará?
3. Quantos municípios fazem fronteira com o município mais populoso de São Paulo?
4. Qual a média do PIB percapita dos municípios que fazem fronteira com o município menor PIB percapita Pará?

**O QUE DIFERE UM  
BANCO DE DADOS  
DE UM BANCO  
TRADICIONAL DE  
DADOS  
GEOGRÁFICO?**



# INTRODUÇÃO

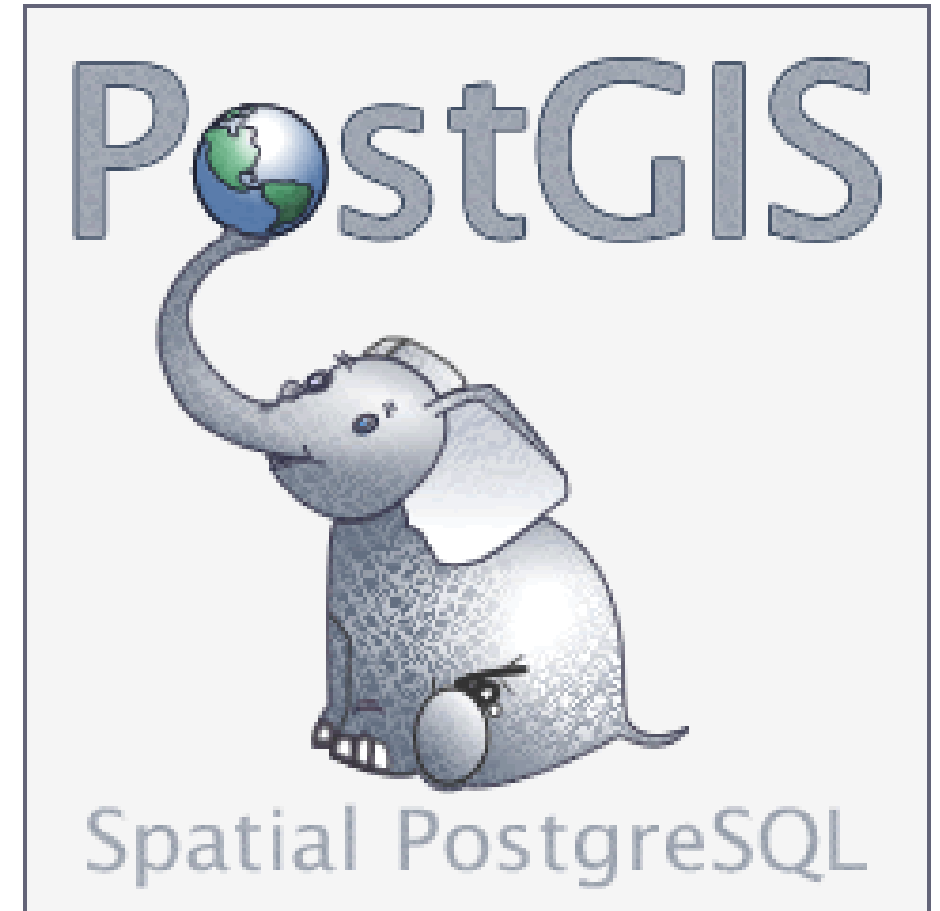
Banco de dados  
geográficos



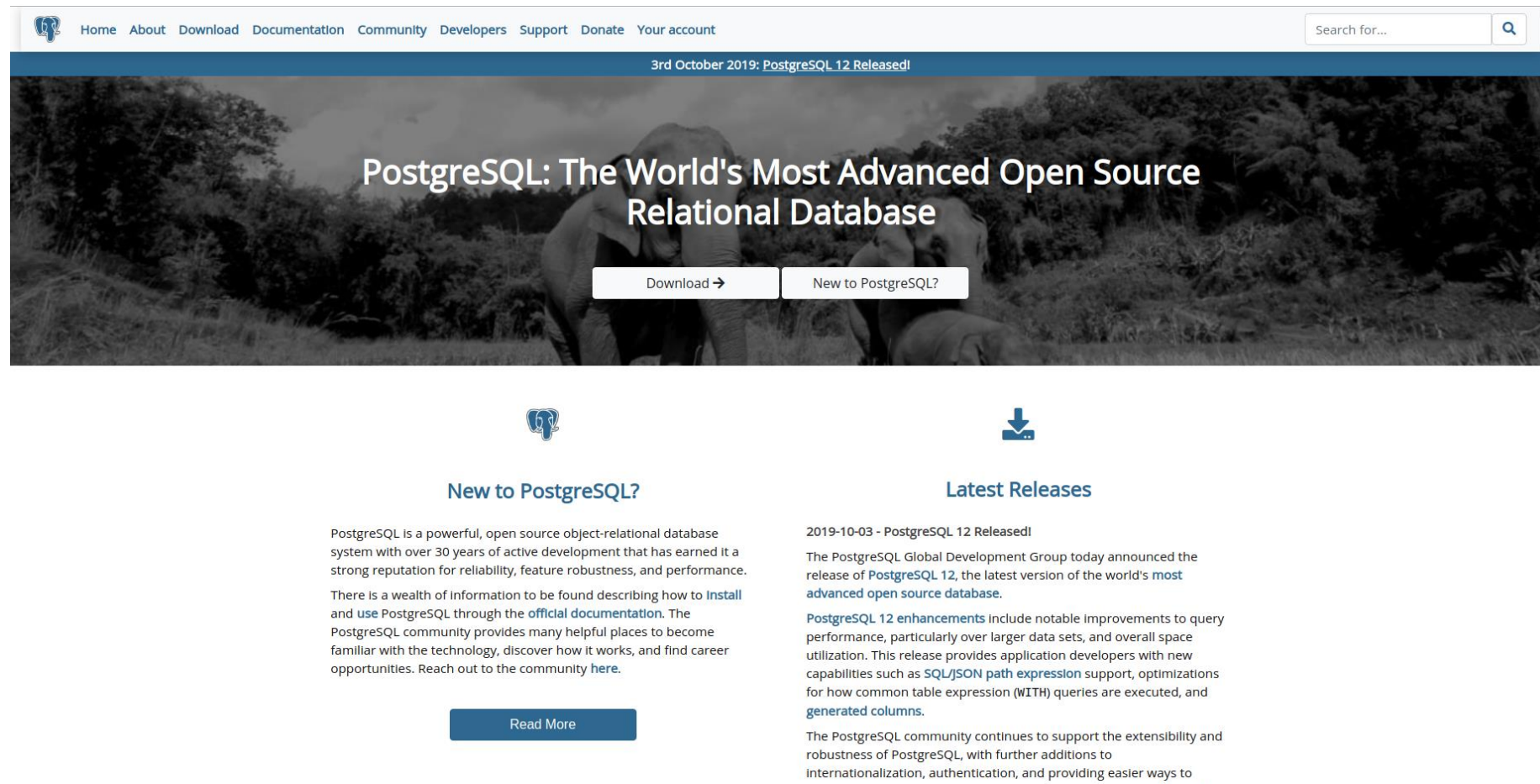
# INTRODUÇÃO

## PostGIS

- Fácil instalação
- Fácil integração com SIG
- Segue o padrão da OGC
- Com suport à GiST e R-Tree
- Material online de fácil acesso
- Comunidade ativa
- Grátis e código fonte aberto







The screenshot shows the PostgreSQL website homepage. At the top, there is a navigation bar with links: Home, About, Download, Documentation, Community, Developers, Support, Donate, and Your account. A search bar is located on the right. Below the navigation bar, a blue banner features the text "3rd October 2019: PostgreSQL 12 Released!". The main heading reads "PostgreSQL: The World's Most Advanced Open Source Relational Database". Below this heading are two buttons: "Download →" and "New to PostgreSQL?". The page is divided into two columns. The left column, titled "New to PostgreSQL?", contains text about PostgreSQL's history and a "Read More" button. The right column, titled "Latest Releases", features a download icon and text about the PostgreSQL 12 release, including details about its enhancements and availability.

Home About Download Documentation Community Developers Support Donate Your account

Search for...

3rd October 2019: [PostgreSQL 12 Released!](#)

## PostgreSQL: The World's Most Advanced Open Source Relational Database

[Download →](#) [New to PostgreSQL?](#)

### New to PostgreSQL?

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

There is a wealth of information to be found describing how to [install](#) and [use](#) PostgreSQL through the [official documentation](#). The PostgreSQL community provides many helpful places to become familiar with the technology, discover how it works, and find career opportunities. Reach out to the community [here](#).

[Read More](#)

### Latest Releases

2019-10-03 - PostgreSQL 12 Released!

The PostgreSQL Global Development Group today announced the release of **PostgreSQL 12**, the latest version of the world's **most advanced open source database**.

**PostgreSQL 12 enhancements** include notable improvements to query performance, particularly over larger data sets, and overall space utilization. This release provides application developers with new capabilities such as **SQL/JSON path expression** support, optimizations for how common table expression (WITH) queries are executed, and **generated columns**.

The PostgreSQL community continues to support the extensibility and robustness of PostgreSQL, with further additions to internationalization, authentication, and providing easier ways to

# POSTGIS

## Instalação

### MacOS:

```
brew install postgres
```

```
brew install postgis
```

### Linux:

```
sudo apt install postgresql postgresql-contrib
```

```
sudo apt-get install postgis
```

# POSTGIS

## Primeiros passos

### Criando banco de dados:

```
CREATE DATABASE <database>;
```

### Habilitando extensão:

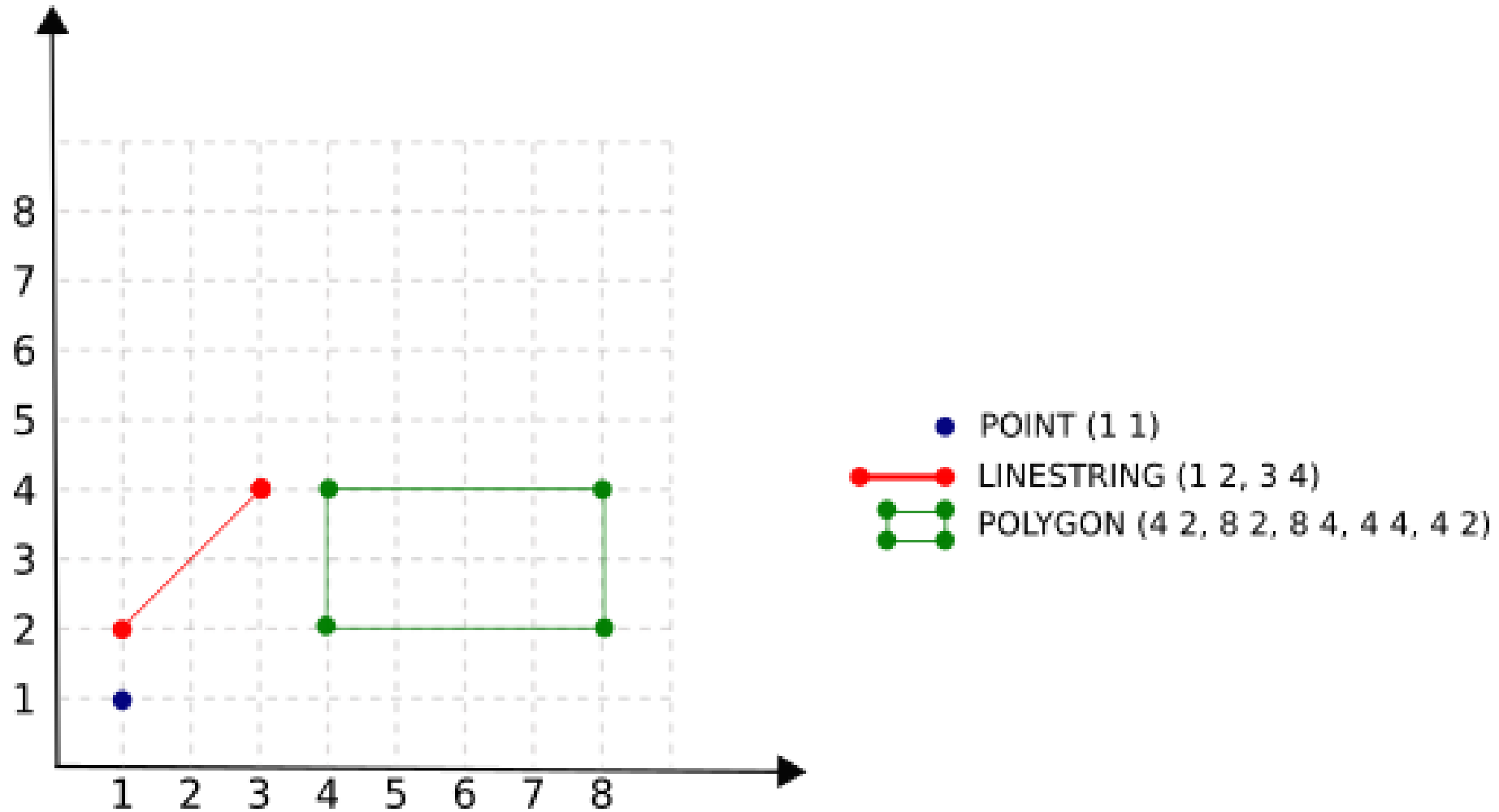
```
CREATE EXTENSION postgis;
```

# POSTGIS

## Criando tabela

```
CREATE TABLE <nome-tabela>(
  id serial PRIMARY KEY,
  <coluna1> <tipo>,
  <coluna2> <tipo>,
  .
  .
  .
  <colunan> <tipo>,
  <geometria> GEOMETRY
);
```

# WELL-KNOW TEXT (WKT)



# WELL-KNOW TEXT (WKT)

Multi-Ponto		<pre>MULTIPOINT (10 40, 40 30, 20 20, 30 10)</pre>
Multi-Segmento		<pre>MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))</pre>
Muli-Polígono		<pre>MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))</pre>
Coleção		<pre>GEOMETRYCOLLECTION (POINT (40 10), LINESTRING (10 10, 20 20, 10 40), POLYGON ((40 40, 20 45, 45 30, 40 40)))</pre>

# WELL-KNOW TEXT (WKT)

## SQL

Uma das formas de interagir com objetos geométricos utilizando SQL é através da função **ST\_GeomFromText()**, que recebe como argumento o WKT da geometria.

```
SELECT ST_GeomFromText('POINT(1 1)');
```

```
SELECT ST_GeomFromText('LINESTRING(1 2, 3 4)');
```

```
SELECT ST_GeomFromText('POLYGON(4 2, 8 2, 8 4, 4 4, 4 2)');
```

# WELL-KNOW TEXT (WKT)

## SQL

Ao ser executada a função para criação da geometria, o retorno padrão é o formato **WKB**, que é o formato binário do WKT.

```
SELECT ST_GeomFromText('POINT(1 1)');
```

Saída:

```
st_geomfromtext | 010100000000000000000000F03F000000000000F03F
```



# WELL-KNOW TEXT (WKT)

## SQL

Para ser exibido a formatação da geometria em WKT, é necessário utilizar a função **ST\_AsText()**.

```
SELECT ST_AsText(ST_GeomFromText('POINT(1 1)'));
```

Saída:

```
st_astext | POINT(1 1)
```

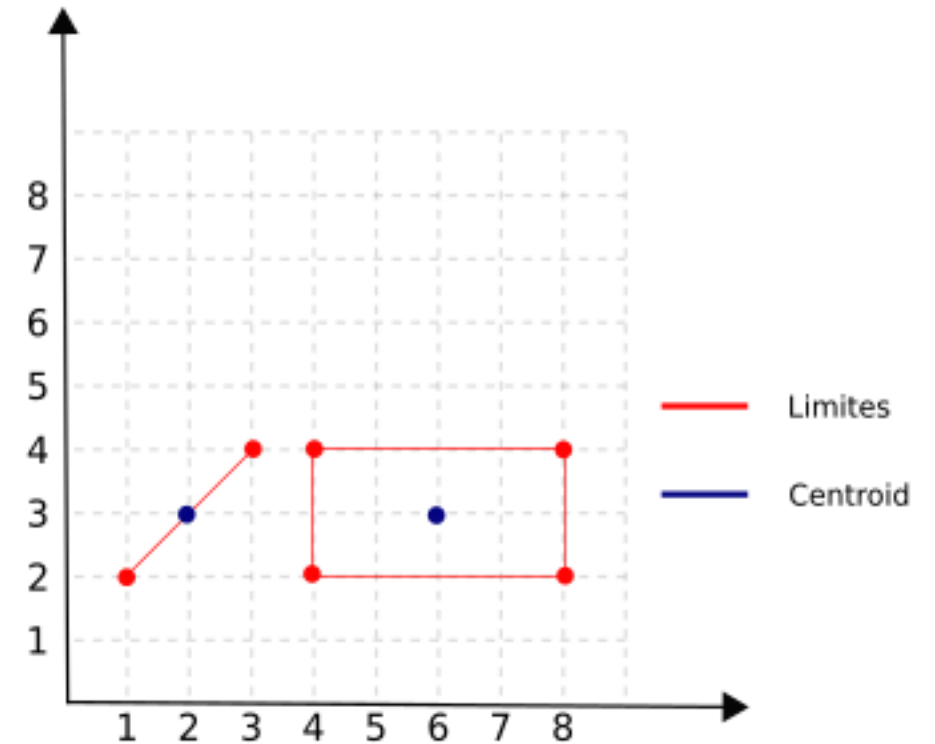
# OPERAÇÕES GEOMÉTRICAS

## Obter ponto central

Para obter o ponto central das figuras geométricas, é utilizada a função **ST\_Centroid()**, passando como argumento a geometria.

```
SELECT  
ST_AsText(ST_Centroid(ST_GeomFromText('LINESTRING(1  
2, 3 4)')));
```

saída:  
st\_astext | POINT(2 3)



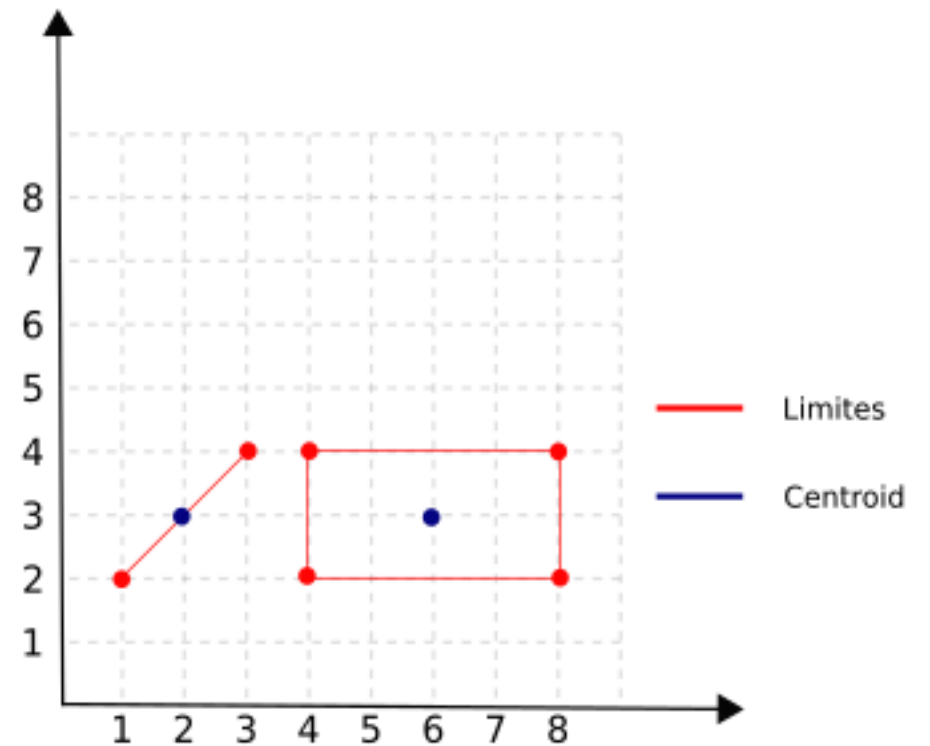
# OPERAÇÕES GEOMÉTRICAS

## Obter ponto central

Para obter o ponto central das figuras geométricas, é utilizada a função **ST\_Centroid()**, passando como argumento a geometria.

```
SELECT  
ST_AsText(ST_Centroid(ST_GeomFromText('POLYGON((4  
2, 8 2, 8 4, 4 4, 4 2 ))')));
```

saída:  
st\_astext | POINT(6 3)



# OPERAÇÕES GEOMÉTRICAS

## Calculando a área

Para calcular a área de alguma figura geométrica 2D, basta utilizar a função **ST\_Area()**.

```
SELECT ST_Area(ST_GeomFromText('POLYGON((4 2, 8 2, 8 4, 4 4, 4 2 ))'));
```

Saída:

```
st_area | 8
```

# OPERAÇÕES GEOMÉTRICAS

## Calculando a área

Para calcular o comprimento de alguma figura geométrica, basta utilizar a função **ST\_Length()**.

```
SELECT ST_Length(ST_GeomFromText('LINESTRING(1 2, 3 4)'));
```

Saída:

```
st_LENGTH | 2.82842712474619
```

# RELACIONAMENTOS ESPACIAIS

## ST\_Relate

Os relacionamentos espaciais são definidos através do resultado da matriz de 9-intersecção. O PostGIS provê uma função para obter uma string representando a matriz de 9-intersecções.

```
SELECT ST_Relate('LINESTRING(1 2,3 4)', 'POLYGON((4 2, 8 2, 8 4, 4 4, 4 2 ))');
```

Saída:

```
st_relate | FF1FF0212
```

# RELACIONAMENTOS ESPACIAIS

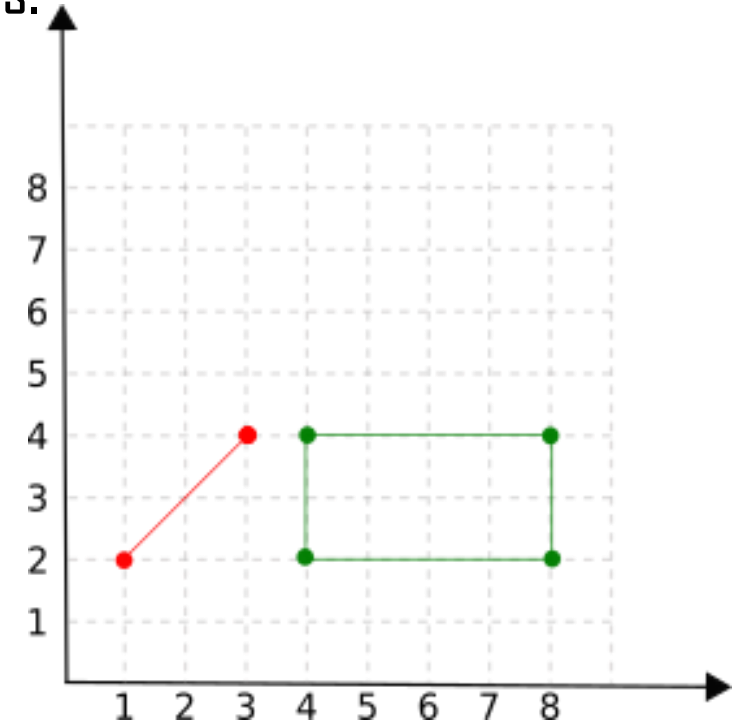
## Distância

Para calcular a distância entre duas geometrias no PostGIS, é disponibilizada a função **ST\_Distance()**, onde são passadas como argumento as duas geometrias.

```
SELECT ST_Distance('LINESTRING(1 2,3 4)', 'POLYGON((4 2, 8 2, 8 4, 4 4, 4 2 ))');
```

Saída:

st_distance	
1	



# RELACIONAMENTOS ESPACIAIS

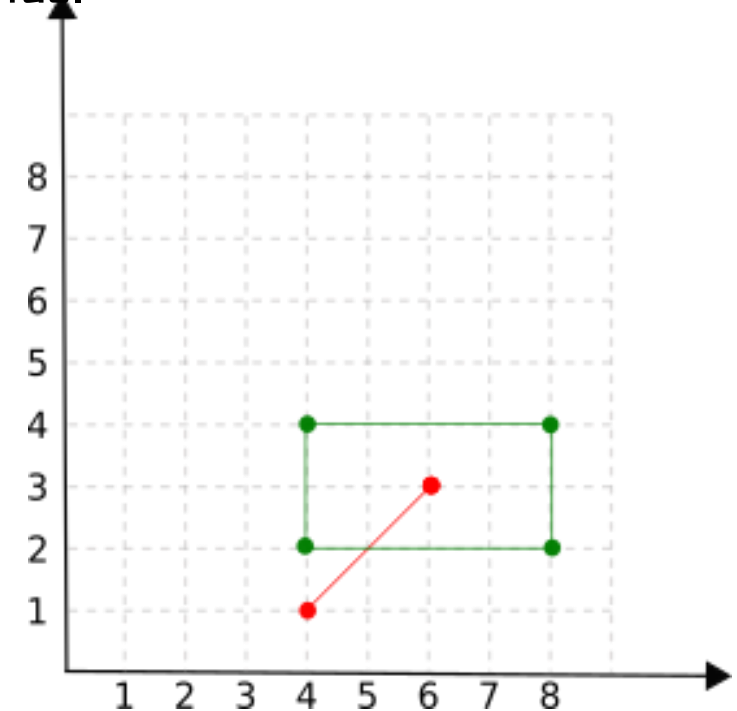
## Intersecção

Para verificar se há intersecção entre duas geometrias no PostGIS, é disponibilizada a função **ST\_Intersects()**, onde são passadas como argumento as duas geometrias.

```
SELECT ST_Intersects('LINESTRING(4 1,6 3)', 'POLYGON((4 2, 8  
2, 8 4, 4 4, 4 2 ))');
```

Saída:

```
st_intersects | t
```





# RELACIONAMENTOS ESPACIAIS

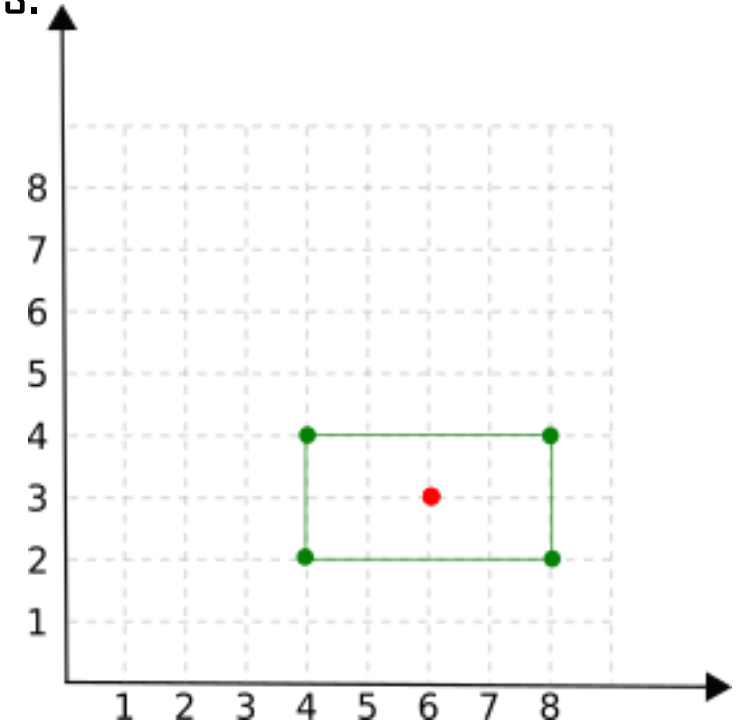
## Contém

Para verificar uma geometria contém uma outra dentro no PostGIS, é disponibilizada a função **ST\_Contains()**, onde são passadas como argumento as duas geometrias.

```
SELECT ST_Contains('POLYGON((4 2, 8 2, 8 4, 4 4, 4 2 ))',  
'POINT(6 3)');
```

Saída:

```
st_intersects | t
```



# RELACIONAMENTOS ESPACIAIS

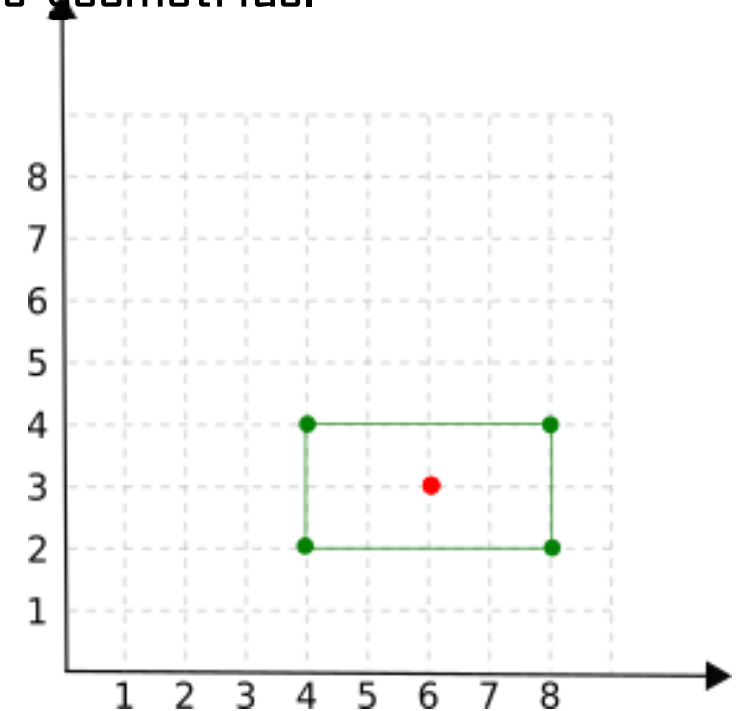
## Dentro

De forma análoga ao **ST\_Contains()**, a função **ST\_Within()** pode ser usada para verificar se uma geometria esta dentro da outra, bastando somente inverter a ordem das geometrias.

```
SELECT ST_Within('POINT(6 3)', 'POLYGON((4 2, 8 2, 8 4, 4 4,  
4 2 )))');
```

Saída:

st_within	t
-----------	---



# POPULANDO TABELAS

## Inserindo geometrias no banco de dados

```
INSERT INTO <nome-tabela> (<col1>, ..., <coln>, <geom>) VALUES(<val1>, ..., <valn>,  
'POINT(1 1)');
```

```
INSERT INTO <nome-tabela> (<col1>, ..., <coln>, <geom>) VALUES(<val1>, ..., <valn>,  
'LINESTRING(1 2, 3 4)');
```

```
INSERT INTO <nome-tabela>  
(<col1>, ..., <coln>, <geom>)  
VALUES(<val1>, ..., <valn>, 'POLYGON(4 2, 8 2, 8 4, 4 4, 4 2)');
```

# RECUPERANDO REGISTROS

Recuperando registros da tabela

```
SELECT <col1>, <coln>, ST_AsText(<geom>) FROM <nome-tabela>;
```

# IMPORTANDO ARQUIVOS

## Importando arquivos shapefile

```
2pgsql -s <SRID> -W <encoding> <shapefile> <banco-de-dados> | psql -d <banco-de-dados> -U <usuario>
```

# O QUE HÁ ALÉM DO TÚNEL?



Info@opengeospatial.org

OGC®  
Making location count.

About Standards Innovation News & Events Membership Resources

Simple Feature Access - Part 1: Common Architecture

1) Downloads

2) Related News

1) Downloads

Version	Document Title (click to download)	Document #	Type
1.2.1	<a href="#">OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture</a>	06-103r4	IS
1.2.0	<a href="#">OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture</a>	06-103r3	D-IS
1.1.0	<a href="#">OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture</a>	05-126	D-IS

Submit a Change Request, Requirement, or Comment for this OGC standard.

2) Related News

No Results Found for

[Go To OGC Press Page](#)

OGC® Standards

- 3D Tiles
- 3dP
- ARML2.0
- Cat: ebRIM App Profile: Earth Observation Products
- Catalogue Service
- CDB
- CityGML
- Coordinate Transformation
- Filter Encoding
- GML in JPEG 2000
- GeoAPI
- GeoPackage
- GeoSciML
- GeoSPARQL
- Geography Markup Language
- GeoRSS
- Geospatial eXtensible Access Control Markup Language (GeoXACML)
- Geospatial User Feedback (GUF)
- GeoTiff
- GroundwaterML
- I3s
- IndoorGML
- KML
- LandInfra/InfraGML
- LAS
- Location Services (OpenLS)
- Moving Features
- NetCDF
- Observations and Measurements
- OGC API - Features
- Open GeoSMS
- OpenMI

Open Geospatial Consortium Inc.

Date: 2011-05-28

Reference number of this document: OGC 06-103r4

Version: 1.2.1

Status: Corrigendum

Category: OpenGIS ® Implementation Standard

Editor: John R. Herring

OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture

Copyright © 2010 Open Geospatial Consortium, Inc.

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>

Document type: OpenGIS Implementation Standard

Document subtype: (none)

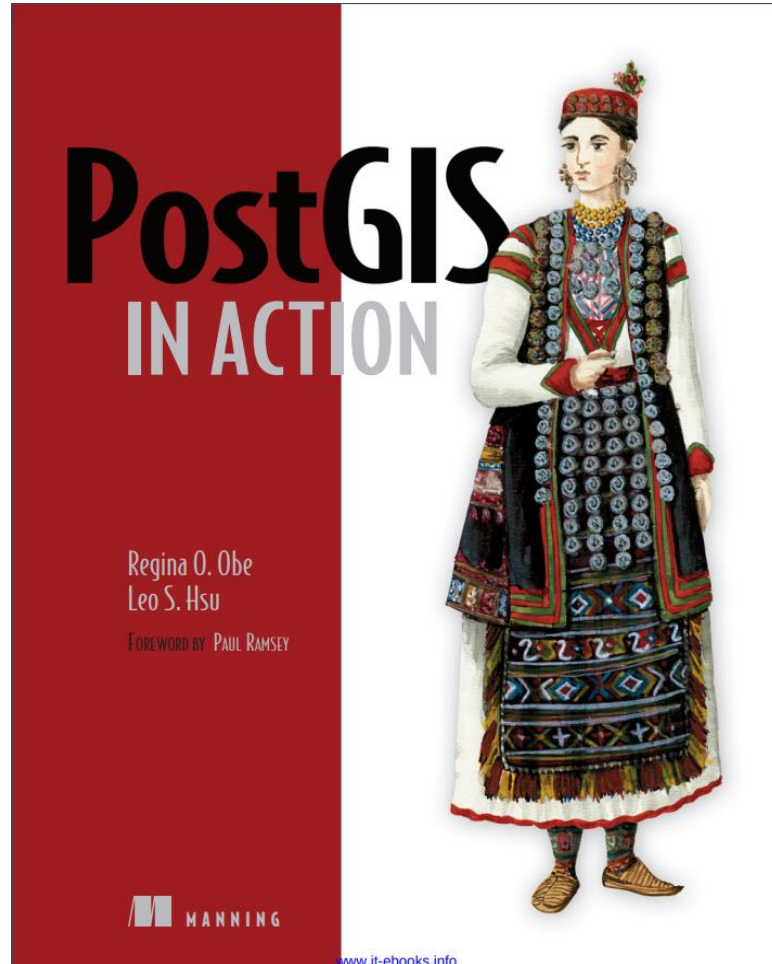
Document stage: Corrigendum

Document language: English

31/34


VI CimaTech- Congress of Industrial Management  
Aeronautical Technology  
Introdução à Análise de Dados Espaciais

# REFERÊNCIAS



## *PostGIS in Action*

REGINA O. OBE  
LEO S. HSU

  
MANNING  
Greenwich  
(74" w. long.)

[www.it-ebooks.info](http://www.it-ebooks.info)



1 Introdução

1.1 Representações de dados espaciais

1.2 Sistema de referência espacial

1.3 Projeções cartográficas

1.4 Sistema de Coordenadas

1.5 Open Geospatial Consortium (OGC)

1.6 Relacionamentos espaciais nominais

1.7 Matriz de 9-Interseções Estendidas

2 R

2.1 Tibble

2.2 SF

3 Python

3.1 Pandas

3.2 GeoPandas

4 Introdução ao postgis

4.1 Instalação

4.2 Um pouco de SQL

4.3 Funções geométricas

4.4 Relacionamentos espaciais

5 Para saber mais

5.1 IMPACTOS DA RESOLUÇÃO E ESCALA

5.2 CRIAÇÃO E AVALIAÇÃO DE MODELOS

5.3 CONECTIVIDADE DEMOGRÁFICA

5.4 PARAMETRIZAÇÃO DE ALGORITMOS

5.5 QUANTIFICAÇÃO REMOTA DADOS

6 References

1 Introdução

A aplicação de dados espaciais no entendimento de um fenômeno geográfico data-se muito antes da criação dos computadores como os conhecemos. Um famoso exemplo no estudo de "análises espaciais de dados" foi pioneiro em relacionar variáveis estudadas a sua localização geográfica. Em meados do século XIX, durante uma das epidemias de cólera na cidade de Londres, John Snow foi bem-sucedido em determinar o epicentro da epidemia. Cruzando a localização das casas em que ocorreram óbitos por cólera com pontos de abastecimentos de água para consumo humano da cidade, John Snow foi capaz de determinar a forma de transmissão da doença e identificar a bomba de água localizada em "broad-street" como crítica (Figura 1.1).



Figure 1.1: Mapa de Londres com óbitos por cólera (pontos) e poços de água (cruzetas)

Como abordado anteriormente, o uso de dados espaciais abrangem diversas aplicações, tais como: estudo de concentração espacial na distribuição de roubos, análise de regiões para fins de zoneamento agrícola e estimativa da extensão de um depósito mineral em uma região a partir de amostras (Monteiro et al. 2004). Desta forma, sempre que **onde** aparece no problema a ser estudado, usa-se recursos computacionais e ferramentas voltadas para análise de dados espaciais. A Figura 1.2 apresenta alguns equipamentos de coleta de dados espaciais, como satélite, radar e VANT (Veículo aéreo não tripulado).

1 Introdução

1.1 Representações de dados espaciais

1.2 Sistema de referência espacial

1.3 Projeções cartográficas

1.4 Sistema de Coordenadas

1.5 Open Geospatial Consortium (OGC)

1.6 Relacionamentos espaciais nominais

1.7 Matriz de 9-Interseções Estendidas

2 R

2.1 Tibble

2.2 SF

3 Python

3.1 Pandas

3.2 GeoPandas

4 Introdução ao postgis

4.1 Instalação

4.2 Um pouco de SQL

4.3 Funções geométricas

4.4 Relacionamentos espaciais

5 Para saber mais

5.1 IMPACTOS DA RESOLUÇÃO E ESCALA

5.2 CRIAÇÃO E AVALIAÇÃO DE MODELOS

5.3 CONECTIVIDADE DEMOGRÁFICA

5.4 PARAMETRIZAÇÃO DE ALGORITMOS

5.5 QUANTIFICAÇÃO REMOTA DADOS

6 References

4 Introdução ao postgis

Uma das principais características que diferenciam os bancos de dados geográficos dos demais bancos de dados convencionais, é a presença de uma coluna com a finalidade de armazenar uma propriedade geográfica do registro. Além disso, os bancos de dados geográficos também devem oferecer de forma otimizada suporte à operações espaciais. Diferente de um passado não muito distante, atualmente existem muitas alternativas de bancos de dados geográficos, como: [SQL Server Spatial](#), [ESRI ArcSDE](#), [Oracle Spatial](#), [GeoMesa](#), [PostGIS](#), etc. Cada uma destas opções tem o objetivo em comum trabalhar com dados geográficos e todas estão sendo muito utilizadas em diversos segmentos nos dias de hoje, sendo o PostGIS a mais popular de todas.

O PostGIS é uma extensão geográfica *open source* para o SGDB (Sistema Gerenciador de Banco de Dados) [PostgreSQL](#), lançada em 2001 inicialmente por uma empresa canadense chamada Refrations Research. O PostGIS segue a padronização estabelecida pelo OGC (*Open GIS Consortium*), que provê suporte para todos os objetos e funções da especificação [SFS \(Simple Features for SQL\)](#). De forma concisa, a especificação SFS foi criada pelo OGC (consórcio formado por empresas, universidades, etc) e trata das questões de representação da componente espacial e vetorial dos dados geográficos, garantindo assim, a interoperabilidade entre os sistemas os utilizam.

Este capítulo tem como objetivo introduzir ao leitor à extensão espacial PostGIS. Ao final, o leitor deverá estar apto a configurar corretamente o ambiente para a utilização da ferramenta, carregar arquivos vetoriais e realizar operações espaciais.

4.1 Instalação

Como o PostGIS é uma extensão do PostgreSQL, primeiramente deverá ser feita a instalação do SGBD.

Importante:

Não esqueça o nome de usuário e senha que você definir durante a instalação, pois essas informações serão de suma importância, anote se necessário.

A versão do PostgreSQL utilizada aqui é a 11.5. A instalação do PostgreSQL e PostGIS no sistema operacional windows, pode ser feita através do instalador disponível [nesse link](#), para isso fique atento durante a instalação e na segunda tela do assistente, no item [Spatial Extensions](#) marque a opção PostGIS.

A instalação do PostgreSQL e PostGIS no sistema Operacional MacOS pode ser feita através do



Minicurso:

# Introdução à Análise de Dados Espaciais

---

Introdução à Extensão Espacial PostGIS



23 de Outubro de 2019  
São José dos Campos - SP



**OBRIGADO!**

Adriano P. Almeida  
Felipe Carvalho  
Felipe Menino  
Helmécio Neto