

# INTRODUÇÃO AO BANCO DE DADOS GEOGRÁFICO

Prof. Sérgio Souza Costa

# Sobre mim

---

- Home: <https://profsergiocosta.github.io>
- Email: prof.sergio.costa@gmail.com

# Introdução

---

- Arquiteturas de sistemas de informação geográfica.
- Modelagem de dados
- Linguagem de consulta

# Arquiteturas de sistemas de informação geográfica

- ❑ Arquitetura baseada em arquivos proprietários e de intercâmbio.
  - ▣ Primeira geração
- ❑ Arquitetura dual e banco de dados relacionais.
  - ▣ Segunda geração
- ❑ Arquitetura integrada e banco de dados objeto relacionais
  - ▣ Terceira geração

**Primeira  
geração**



API  
Proprietária

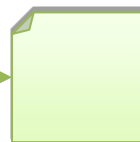


Geometria, índices e atributos armazenados em sistemas de arquivos proprietários.

**Segunda  
geração**



API  
Proprietária



Geometria e índices armazenados em sistemas de arquivos proprietários.

SQL



Atributos armazenados em sistemas de gerenciamento de banco de dados

**Terceira  
geração**



SQL



Geometria, índices, e atributos armazenados em sistemas de gerenciamento de banco de dados

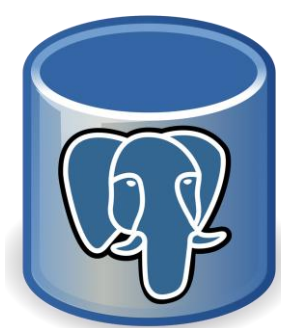
# Sistema de gerenciamento de banco de dados

---

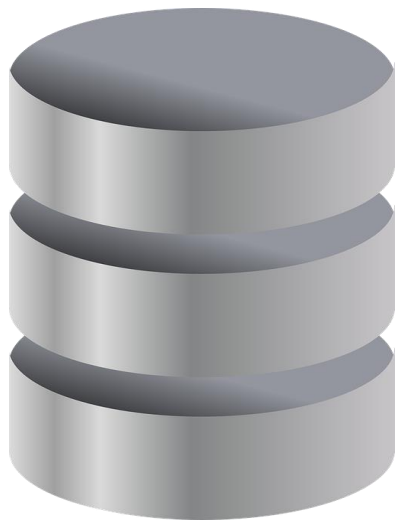
- ❑ O que é um sistema de gerenciamento de banco de dados ?
- ❑ Por que usar sistema de gerenciamento de banco de dados ?

# Sistema de gerenciamento de banco de dados

- ❑ Um banco de dados é uma coleção de dados relacionados.
- ❑ Um sistema gerenciador de banco e dados é uma coleção de programas que permite aos usuários criar e manter um banco de dados.



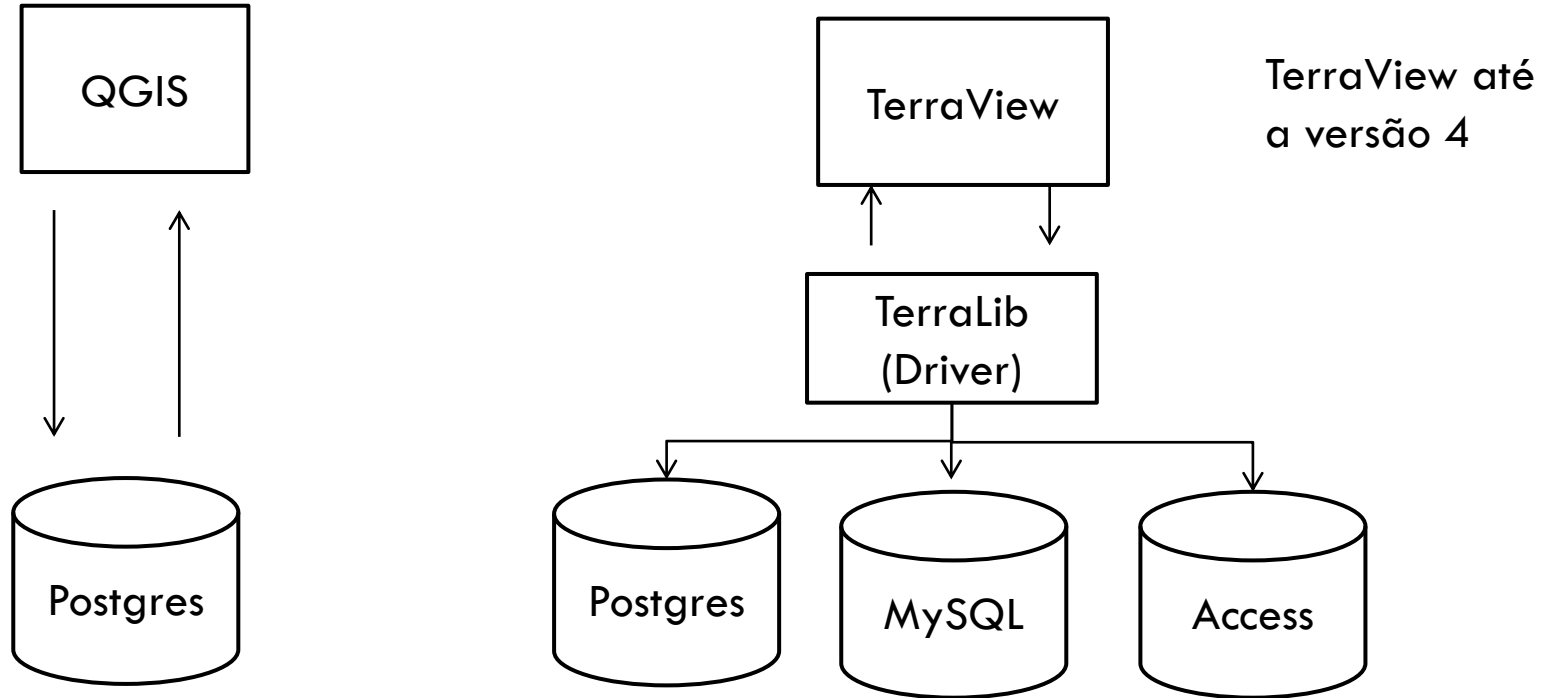
# Sistema de gerenciamento de banco de dados



- ❌ Restringir acesso não autorizado.
- ❌ Tornar as consultas mais eficientes.
- ❌ Backup e recuperação.
- ❌ Controlar a redundância.
- ❌ Representar relacionamentos complexos.
- ❌ Impor restrições de integridade.



# Exemplos de SIGs “open source”



# Modelagem de dados



A modelagem de dados tem como objetivo criar modelos que descrevam como os dados estão organizados e como se relacionam entre si.

Nesta aula, vou discutir separadamente:

- Modelagem de dados tradicionais
- Modelagem de dados espaciais

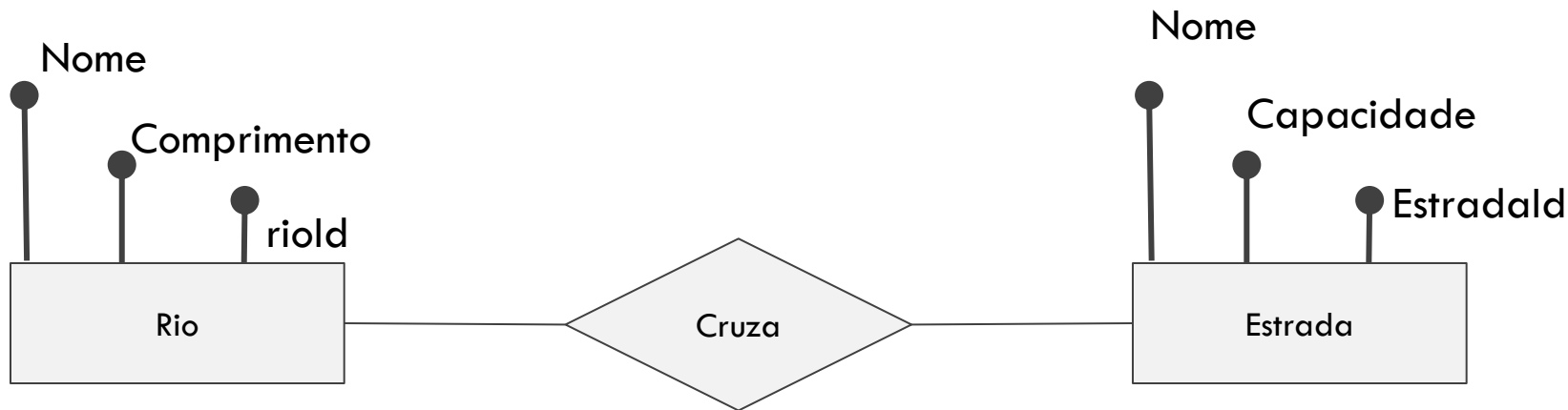
# Modelagem de dados tradicionais



- Modelo Entidade-Relacionamento
- Modelo Relacional
- Modelo Orientado a Objetos

# Modelo de entidade e relacionamento

Entidades, relacionamentos e atributos:



# Modelo relacional

Entidades e relacionamentos são mapeados para tabelas, atributos são nominados como campos. Além disso todo campo possui um dado domínio.



# Modelo orientado a objetos



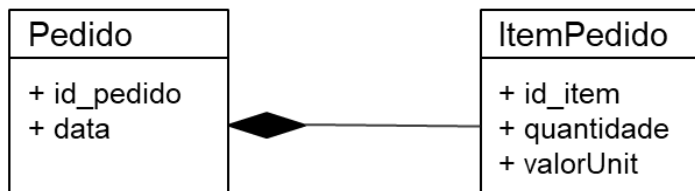
Os bancos de dados relacionais são dominantes. Contudo, existem hoje alguns bancos de dados de objetos e mais comumente os bancos de dados objeto-relacionais.

Os objeto-relacionais integram alguns recursos dos banco de dados de objetos aos bancos de dados relacionais já existentes.

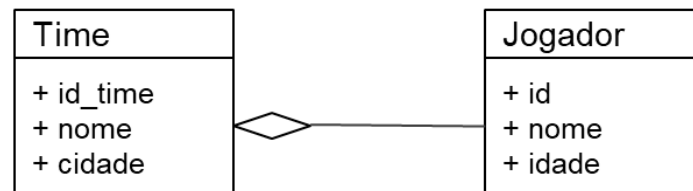
Tanto os banco de dados de objetos, quanto os objeto-relacionais têm usado os diagramas de classes que já são conhecidos pelos programadores.

# Modelo orientado a objetos

Nesse modelo, as entidades são mapeadas como classes. As relações são nomeadas como associação e podem ser classificadas como composição (as partes não existem de modo independente) ou agregação (as partes existem de modo independente).



Composição



Agregação

# Modelagem de dados espaciais



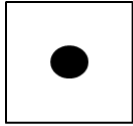
Extensões através de pictogramas

Modelo OMT-G

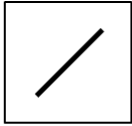


# Extensões através de pictogramas

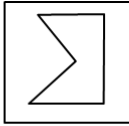
Em SHEKHAR (2005) o autor apresenta alguns pictogramas que podem agregar informações aos modelos de entidade e relacionamento ou diagrama de classes:



Ponto

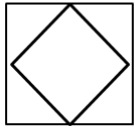


Linha

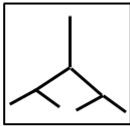


Polígono

Formas  
básicas



Part-de  
(partição)

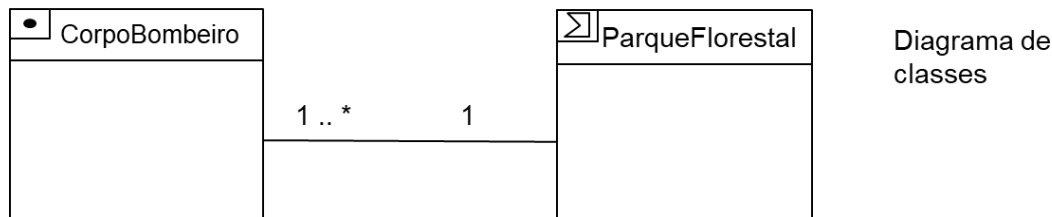
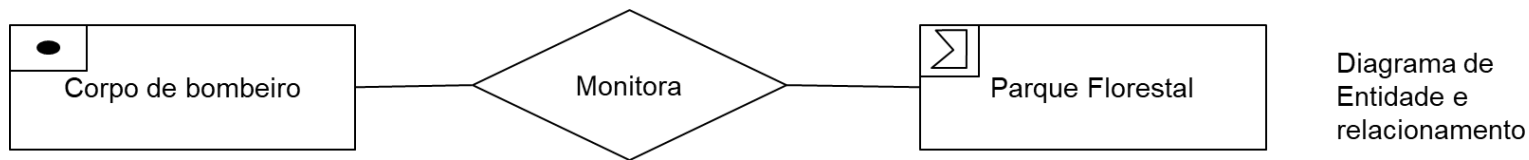


Part-de  
(rede)

Relações  
espaciais

# Extensões através de pictogramas

Em SHEKHAR (2005) o autor apresenta alguns pictogramas que podem agregar informações aos modelos de entidade e relacionamento ou diagrama de classes:



# Modelo OMT-G



Borges et al. (2001), desenvolveu uma extensão para modelagem orientada a objetos, provendo primitivas para modelar a geometria, topologia, múltiplas representações e relacionamentos espaciais. Essa extensão foi então denominada de OMT-G.

# Modelo OMT-G

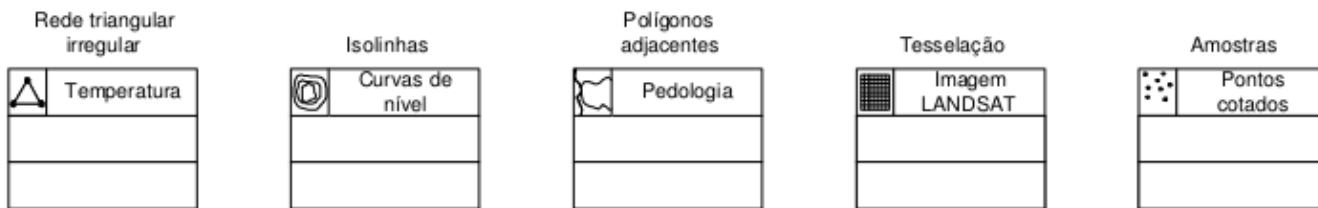


Borges et al. (2001), desenvolveu uma extensão para modelagem orientada a objetos, provendo primitivas para modelar a geometria, topologia, múltiplas representações e relacionamentos espaciais. Essa extensão foi então denominada de OMT-G.

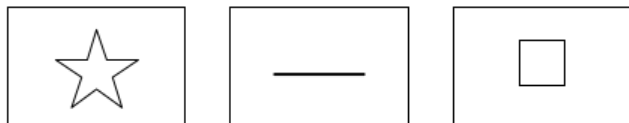
Essa extensão, utiliza pictogramas similares aos apresentados em SHEKHAR (2005), porém muito mais rica.

# Modelo OMT-G

O modelo OMT-G dá suporte as duas diferentes visões de dados geográficos: os geo-campos e geo-objetos.



Geo-campos



Ponto

Linha

Polígono

Geo-objetos

Veja mais em:

<https://pt.slideshare.net/skosta/modelagem-de-dados-geograficos>

# Linguagem de consulta

- Linguagem de consulta (SQL)
- Extensões espaciais
  - ▣ Tipos de dados espaciais
  - ▣ Operadores espaciais

# Linguagem de consulta

Linguagem padrão para consulta e manipulação de dados

**Structured Query Language**

Desenvolvida pela IBM na década de 70, inicialmente chamada de SEQUEL.

Tem evoluído, e vários padrões já foram desenvolvidos:

ANSI SQL, SQL92 (SQL2), SQL99 (SQL3), SQL2003 ...

# Linguagem de consulta (SQL)

**Produto**

Nome da tabela

Campos (ou atributos)

Nome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

Linhas (ou tuplas)



# Linguagem de consulta (SQL)

**Produto**

Nome da tabela

Campos (ou atributos)

Nome	Preço	Categoria	Marca
Galaxy		Celular	Samsung
S04 Mo		Eletrodoméstico	Três Corações
Bella A		Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

Uma tupla ou linha é  
usualmente referida como  
registro, enquanto a  
tabela é uma coleção de  
registros.

Linhas (ou tuplas)

# Linguagem de consulta (SQL)

- ❑ Todo campo tem um domínio, ou seja, ele precisa ter um tipo de dados definido:
- ❑ Os tipos básicos são:
  - ▣ Caracteres: CHAR(20), VARCHAR(50)
  - ▣ Números: INT, BIGINT, SMALLINT, FLOAT
  - ▣ Outros: MONEY, DATETIME, ...

- Uma tabela pode ser descrita textualmente pelo seu nome, e dos seus campos:
  - ▣ Produto (Nome, Preço, Categoria, Marca)
  
- Um dos campos pode ser escolhido como chave primária, um campo que pode ser usado para identificar. Nesse caso ele é usualmente sublinhado:
  - ▣ Produto (Nome, Preço, Categoria, Marca)

- Definição de dados: criação das estruturas
  - ▣ Data Definition Language (DDL)
  - ▣ create, alter, drop
- Manipulação de dados: atualização e consultas
  - ▣ Data Manipulation Language (DML)
  - ▣ select, insert, update, delete ...
- Controle de acesso
  - ▣ Data Control Language (DCL)
  - ▣ grant Revoke, etc

# Linguagem de consulta (SQL) - Select

## □ Forma básica

```
SELECT      lista de campos  
FROM lista de tabelas  
WHERE      condição.
```

Nome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

Nome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

```
SELECT *  
FROM produto  
WHERE categoria =  
'Eletrodoméstico'
```

Nome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

Nome	Preço	Categoria	Marca
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial

```
SELECT *
FROM produto
WHERE categoria =
'Eletrodoméstico'
```



Nome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

Nome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

***SELECT*** Nome, Marca  
***FROM*** produto  
***WHERE*** preco > 1000

Nome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

Nome	Marca
Galaxy A9 SM-A910	Samsung
Smart TV HD Série 4 LED 32 polegadas	Samsung

***SELECT*** Nome, Marca  
***FROM*** produto  
***WHERE*** preco > 1000

# Linguagem de consulta (SQL) - Select (chaves)

Chave estrangeira

Chave primária

PNome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

ENome	País
Samsung	Coréia do Sul
Três Corações	Brasil
Mondial	Brasil

# Linguagem de consulta (SQL) - Select (chaves)

PNome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

ENome	País
Samsung	Coréia do Sul
Três Corações	Brasil
Mondial	Brasil

PNome	Preço
S04 Modo Expresso	299.99
Bella Arome II C-09	149.99

```
SELECT          PNome, Preço
FROM            Produto, Empresa
WHERE           Marca=ENome
                  AND Pais='Brasil'
```

# Linguagem de consulta (SQL) - Select (agregação)

- ❑ SQL suporta várias operações de agregação para os dados básicos:
  - ▣ sum, count, min, max, avg

```
SELECT      avg (Preco)
FROM        Produto
```

```
SELECT      count (*)
FROM        Produto
where marca = 'samsung'
```

# Linguagem de consulta (SQL) - Select (agregação)

PNome	Preço	Categoria	Marca
Galaxy A9 SM-A910	1198	Celular	Samsung
S04 Modo Expresso	299.99	Eletrrodoméstico	Três Corações
Bella Arome II C-09	149.99	Eletrrodoméstico	Mondial
Smart TV HD Série 4 LED 32 polegadas	1250	TV	Samsung

ENome	País
Samsung	Coréia do Sul
Três Corações	Brasil
Mondial	Brasil

```
SELECT      sum(Preço)
FROM        Produto, Empresa
WHERE       Marca=ENome
            AND Pais='Brasil'
```



$149.99 + 299.99 = 449.98$

# Extensões espaciais





# Extensões espaciais

- Alguns gerenciadores de banco de dados dão suporte a manipulação de dados espaciais, sendo referido como banco de dados espaciais (ou geográficos).
  - ▣ Banco de dados espaciais armazenam e manipulam objetos espaciais como qualquer outro objeto.
- Onde um banco de dados espacial é:

*“Spatial database system is a database system that offers spatial data types in its data model and query language and supports spatial data types in its implementation, providing at least spatial indexing and spatial join methods”  
(Guting, 1994).*

# Extensões espaciais

Essas extensões espaciais tem que dar suporte a três aspectos principais dos gerenciadores de banco de dados: dados espaciais, funções espaciais e índices.

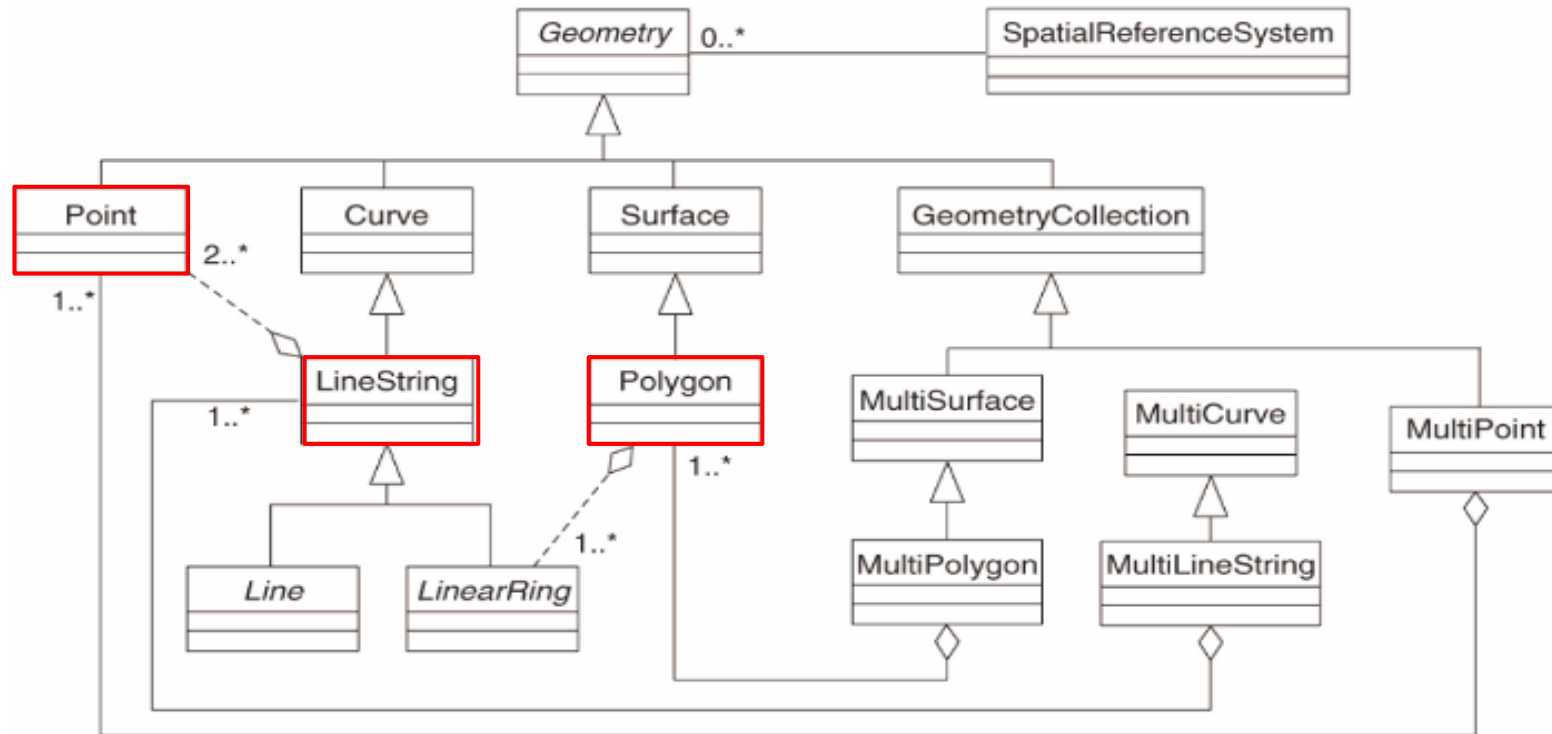
## Convencionais

- ❑ Tipos
  - ▣ char, integer, real, date
- ❑ Índices
  - ▣ b-tree, hash
- ❑ Funções
  - ▣ strlen(char)
  - ▣ pow(real, real),
  - ▣ now()



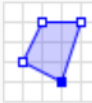
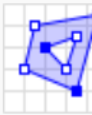
## PostGIS

- ❑ Tipos
  - ▣ geometry, geography
- ❑ Índices
  - ▣ r-tree, quad-tree, kd-tree
- ❑ Funções
  - ▣ ST\_Length(geometry),
  - ▣ ST\_X(geometry)

# Extensões espaciais – tipos de dados

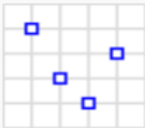
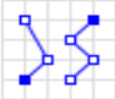
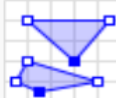
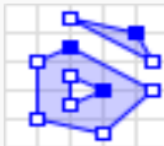


# Extensões espaciais – tipos de dados

Geometry primitives (2D)		
Type	Examples	
<u>Point</u>	POINT (30 10)	
<u>LineString</u>	LINESTRING (30 10, 10 30, 40 40)	
<u>Polygon</u>	POLYGON ((30 10, 10 20, 20 40, 40 40, 30 10))	
	POLYGON ((35 10, 10 20, 15 40, 45 45, 35 10), (20 30, 35 35, 30 20, 20 30))	

*Os dados estão descritos no formato textual Well-known text (**WKT**).*

# Extensões espaciais – tipos de dados

Multipart geometries (2D)		
Type	Examples	
<u>MultiPoint</u>	MULTIPOINT ((10 40), (40 30), (20 20), (30 10))	
	MULTIPOINT (10 40, 40 30, 20 20, 30 10)	
<u>MultiLineString</u>	MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))	
<u>MultiPolygon</u>	MULTIPOLYGON (((30 20, 10 40, 45 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))	
	MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40)), ((20 35, 45 20, 30 5, 10 10, 10 30, 20 35), (30 20, 20 25, 20 15, 30 20)))	

*Os dados estão descritos no formato textual Well-known text (**WKT**).*

# Extensões espaciais - exemplo

Criando a extensão em um dado banco de dados:

```
create extension postgis;
```

Verificando a tabela do sistema de referência:

Data Output	Explain	Messages	History	
	srid integer	auth_name character varying(256)	auth_srid integer	srtext character varying(2048)
1	3819	EPSG	3819	GEOGCS["HD1909",DATUM["Hungarian_Datum_1909",SPHEROID[
2	3821	EPSG	3821	GEOGCS["TWD67",DATUM["Taiwan_Datum_1967",SPHEROID["GRS
3	3824	EPSG	3824	GEOGCS["TWD97",DATUM["Taiwan_Datum_1997",SPHEROID["GRS
4	3889	EPSG	3889	GEOGCS["IGRS",DATUM["Iraqi_Geospatial_Reference_System
5	3906	EPSG	3906	GEOGCS["MGI_1901",DATUM["MGI_1901",SPHEROID["Bessel 18.
6	4001	EPSG	4001	GEOGCS["Unknown datum based upon the Airy 1830 ellipso.
7	4002	EPSG	4002	GEOGCS["Unknown datum based upon the Airy Modified 184'

# Extensões espaciais - exemplo

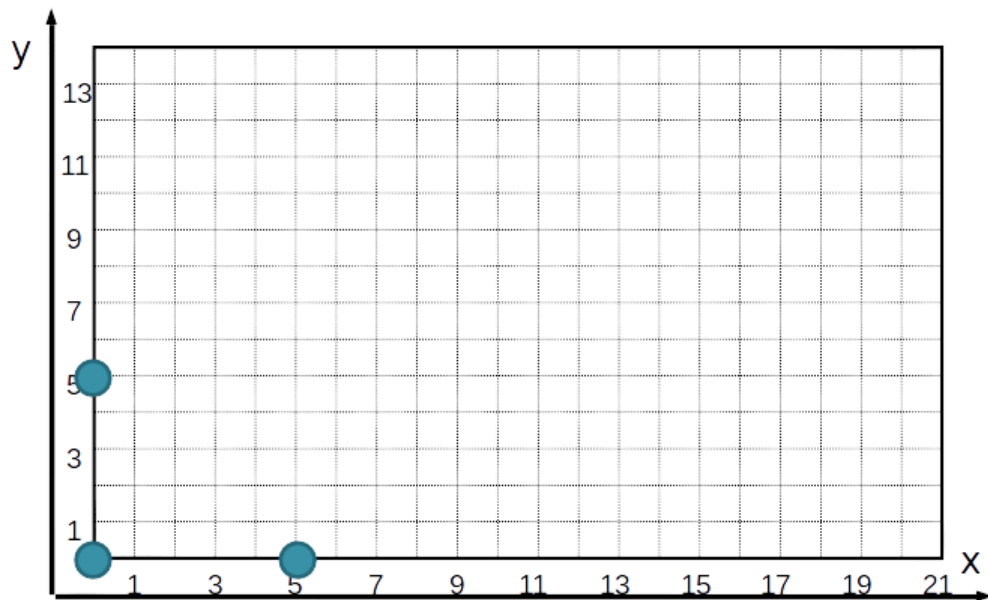


Criando uma tabela de pontos:

```
CREATE TABLE pontos (  
    geom_pt    geometry(POINT,0),  
    nome       varchar  
);
```

# Extensões espaciais - exemplo

Considerem três pontos no plano cartesiano:





## Extensões espaciais - exemplo



Inserindo os três pontos na tabela pontos:

```
INSERT INTO pontos VALUES ('POINT(0 0)', 'Origem');
```

```
INSERT INTO pontos VALUES ('POINT(5 0)', 'Eixo X');
```

```
INSERT INTO pontos VALUES ('POINT(0 5)', 'Eixo Y');
```

© 2014 Pearson Education, Inc. or its affiliate(s). All rights reserved. Pearson Education, Inc., publishing as Pearson Benjamin Cummings, 101 Philip Drive, Assinippi Park, New York, NY 10984-2135

```
SELECT nome, geom_pt
FROM pontos;
```

[illegible]

0000

# Extensões espaciais - exemplo

Consulta:

```
SELECT nome, ST_AsText(geom_pt)
FROM pontos;
```

*Função PostGIS*



*WKT (Well-Known  
Text)*



	nome character varying	st_astext text
1	Origem	POINT(0 0)
2	Eixo X	POINT(5 0)
3	Eixo Y	POINT(0 5)

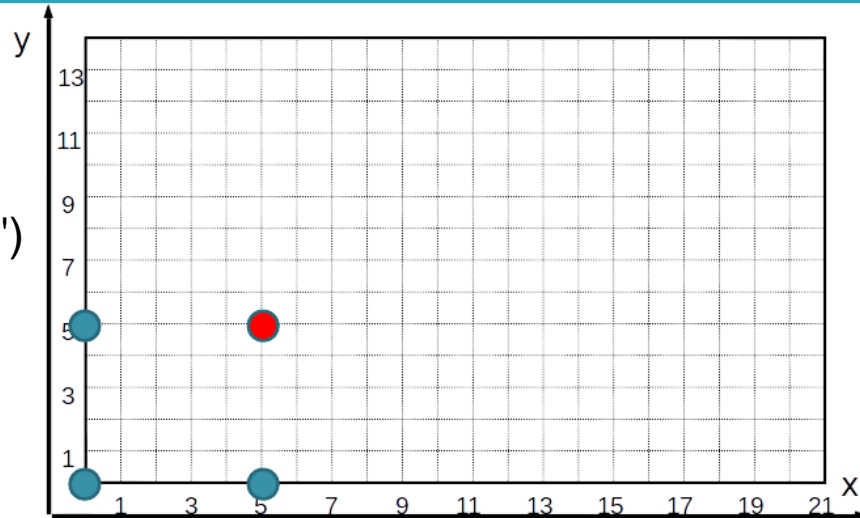
# Extensões espaciais - exemplo

Consulta:

```
SELECT nome, ST_AsText(geom_pt),  
          ST_Distance(geom_pt, 'POINT(5 5)')  
FROM pontos;
```

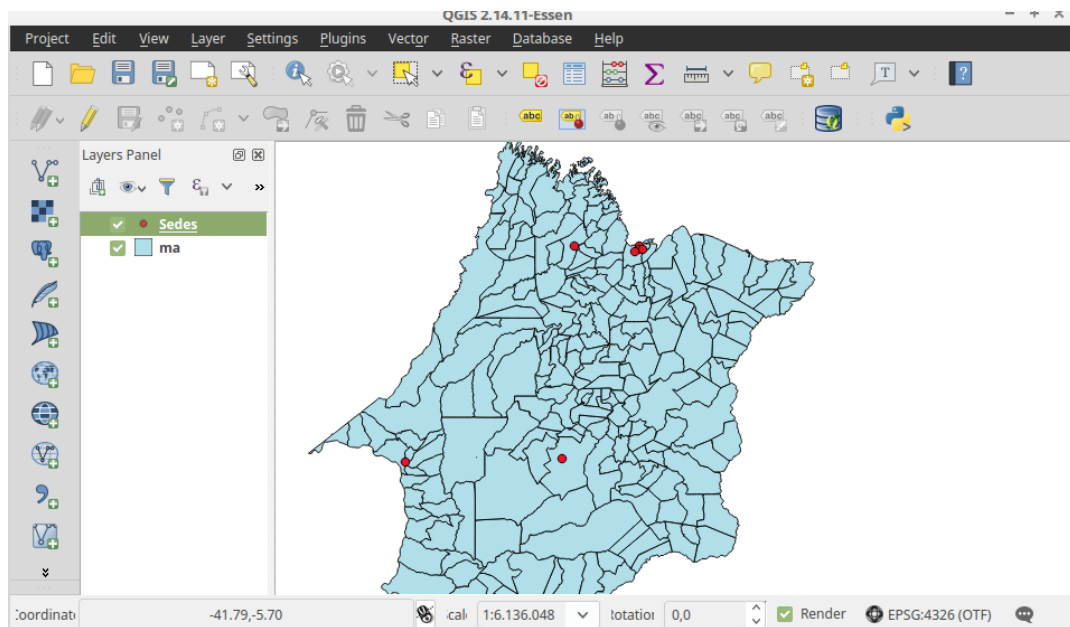
*Função  
PostGIS*

	nome character varying	st_astext text	st_distance double precision
1	Origem	POINT(0 0)	7.07106781186548
2	Eixo X	POINT(5 0)	5
3	Eixo Y	POINT(0 5)	5



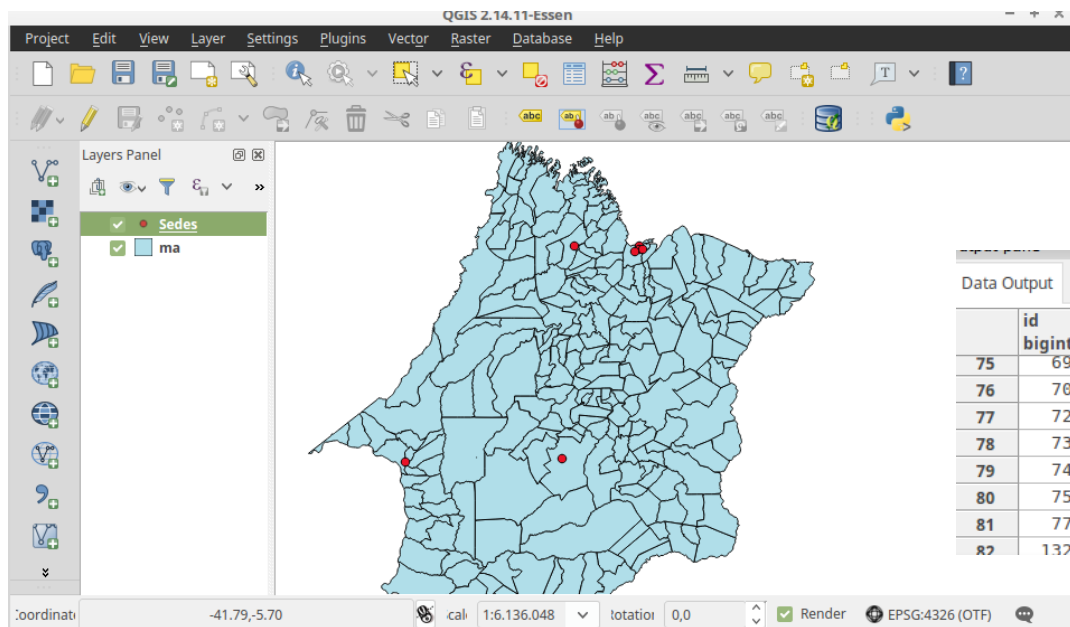
# Extensões espaciais - exemplo

O mais comum é utilizarmos um banco de dados integrado com um sistema de informação geográfica, por exemplo o QGIS:



# Extensões espaciais - exemplo

O mais comum é utilizarmos um banco de dados integrado com um sistema de informação geográfica, por exemplo o QGIS:



**select \* from ma;**



Data Output					Explain	Messages	History
	id	geom	cd_geocodm	nm_municip			
	bigint	geometry(MultiPolygon,401	character varying(20)	character varying(60)			
75	69	0106000020B30F0000010	2104057	ESTREITO			
76	70	0106000020B30F0000010	2104073	FEIRA NOVA DO MARANHÃO			
77	72	0106000020B30F0000010	2104099	FORMOSA DA SERRA NEGRA			
78	73	0106000020B30F0000010	2104107	FORTALEZA DOS NOGUEIRAS			
79	74	0106000020B30F0000010	2104206	FORTUNA			
80	75	0106000020B30F0000010	2104305	GODOFREDO VIANA			
81	77	0106000020B30F0000010	2104503	GOVERNADOR ARCHER			
82	132	0106000020B30F0000010	2107704	PARATRANO			

# Extensões espaciais - exemplo

Exemplo de uso das funções Within (dentro) e de transformações entre projeções:

Todas as sedes que estão localizadas em São Luís (186):

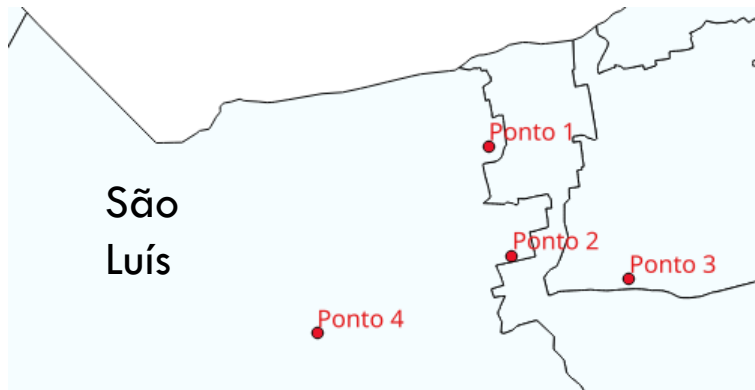
```
select distinct "Name", ma.id from sedes, ma where ma.id = 186 and  
    ST_Within(          ST_Transform(sedes.geom, 4326),  
                ST_Transform(ma.geom, 4326))
```

# Extensões espaciais - exemplo

Exemplo de uso das funções Within (dentro) e de transformações entre projeções:

Todas as sedes que estão localizadas em São Luís (186):

```
select distinct "Name", ma.id from sedes, ma where ma.id = 186 and  
    ST_Within(  
        ST_Transform(sedes.geom, 4326),  
        ST_Transform(ma.geom, 4326))
```



	Name	id
	character varying	bigint
1	Ponto 1	186
2	Ponto 2	186
3	Ponto 4	186



# Referências

GÜTING, Ralf Hartmut. An introduction to spatial database systems. **The VLDB Journal—The International Journal on Very Large Data Bases**, v. 3, n. 4, p. 357-399, 1994.

Borges, K. A. V.; Davis JR., C. A.; Laender, A. H. F., OMT-G: an object-oriented data model for geographic applications. *Geoinformatica*, v. 5, n.3, p. 221-260, 2001

SHEKHAR, Shashi; CHAWLA, Sanjay. **Spatial databases: a tour**. Upper Saddle River, NJ: prentice hall, 2005.