



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



# Sistemas de Bancos de Dados Geoespaciais

Evolução das Tecnologias de Bancos de Dados

Dr. Gilberto Ribeiro de Queiroz

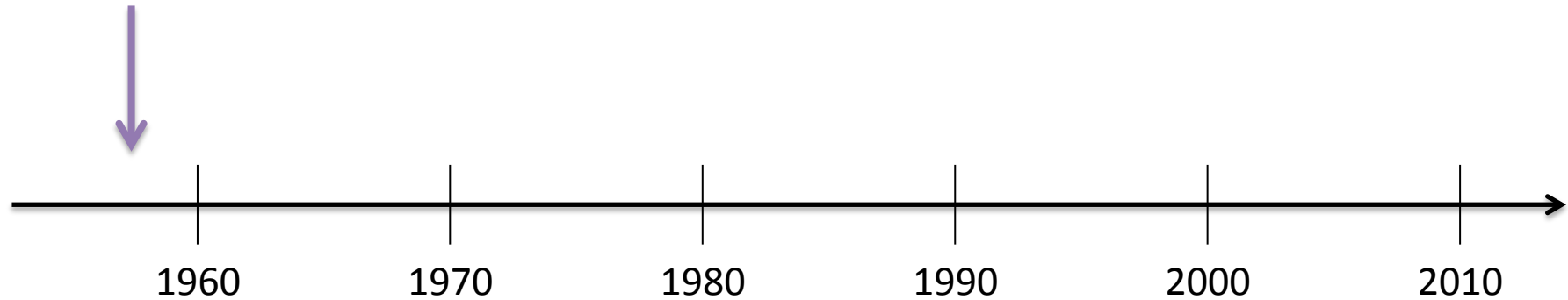
<[gribeiro@dpi.inpe.br](mailto:gribeiro@dpi.inpe.br)>

# SGBD: uma tecnologia amplamente difundida

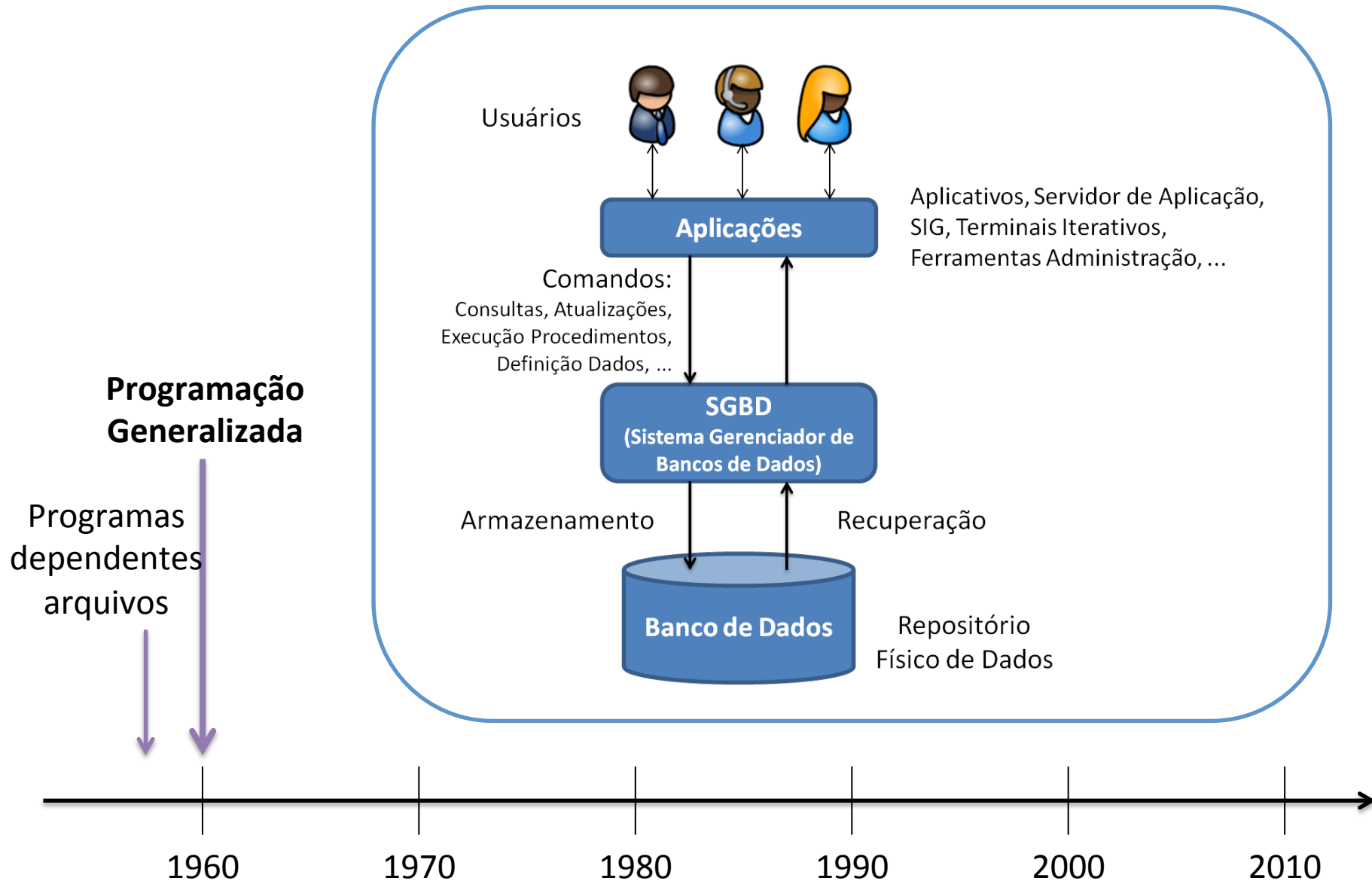
- A tecnologia de bancos de dados tem sido um componente fundamental em quase todos os tipos de aplicações:
  - Conta bancária: depósitos e saques
  - Reservas de passagens aéreas
  - Reservas em hotéis
  - Compras de livros, CDs, DVDs e outros bens (Amazon)
  - Busca por artigos em uma revista eletrônica (Transactions of GIS ou ACM digital library)
  - Sites de mapeamento: OpenStreetMap, GoogleMaps e Bing Maps

# Evolução das Tecnologias de Bancos Dados

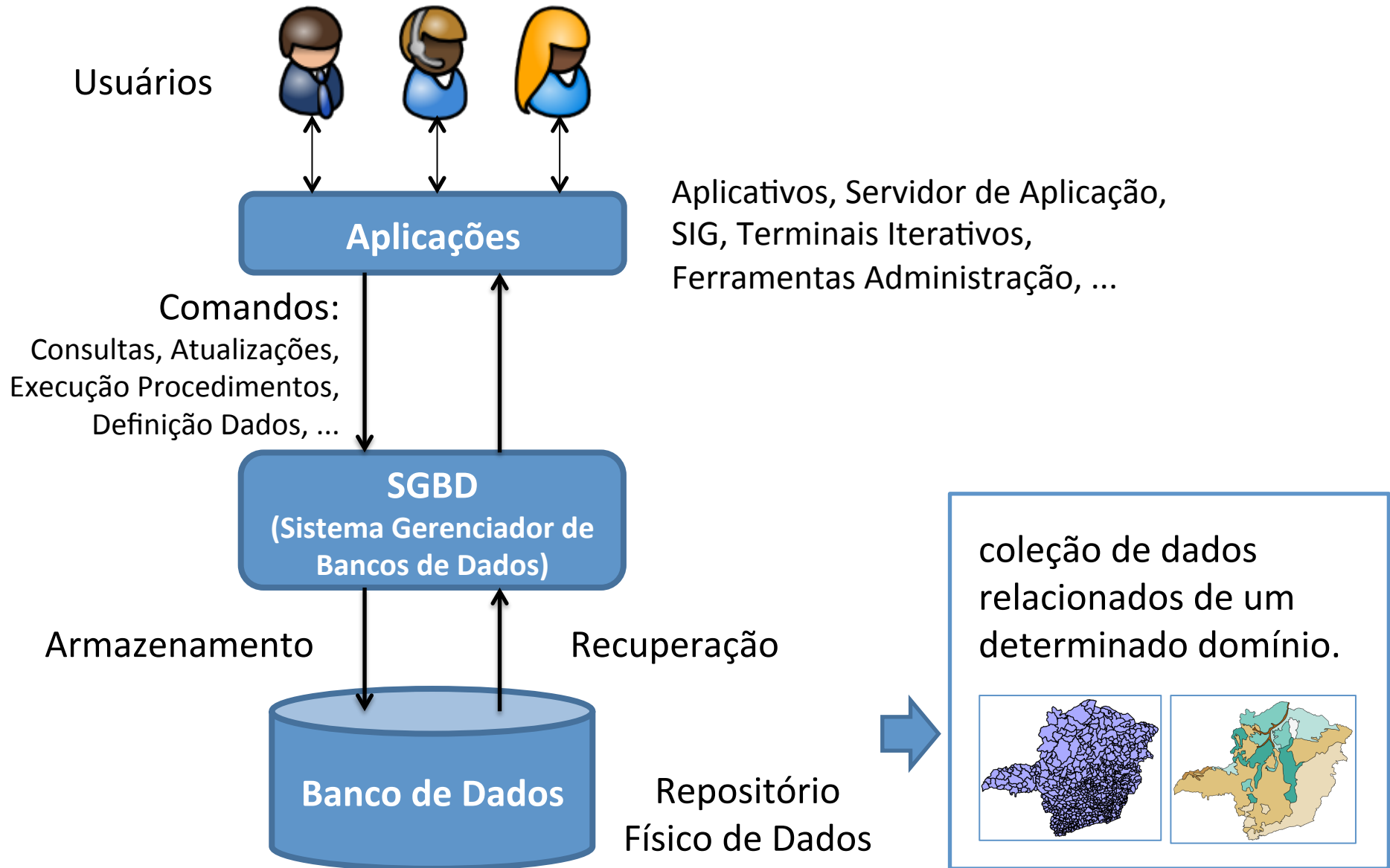
Programas  
dependentes  
arquivos



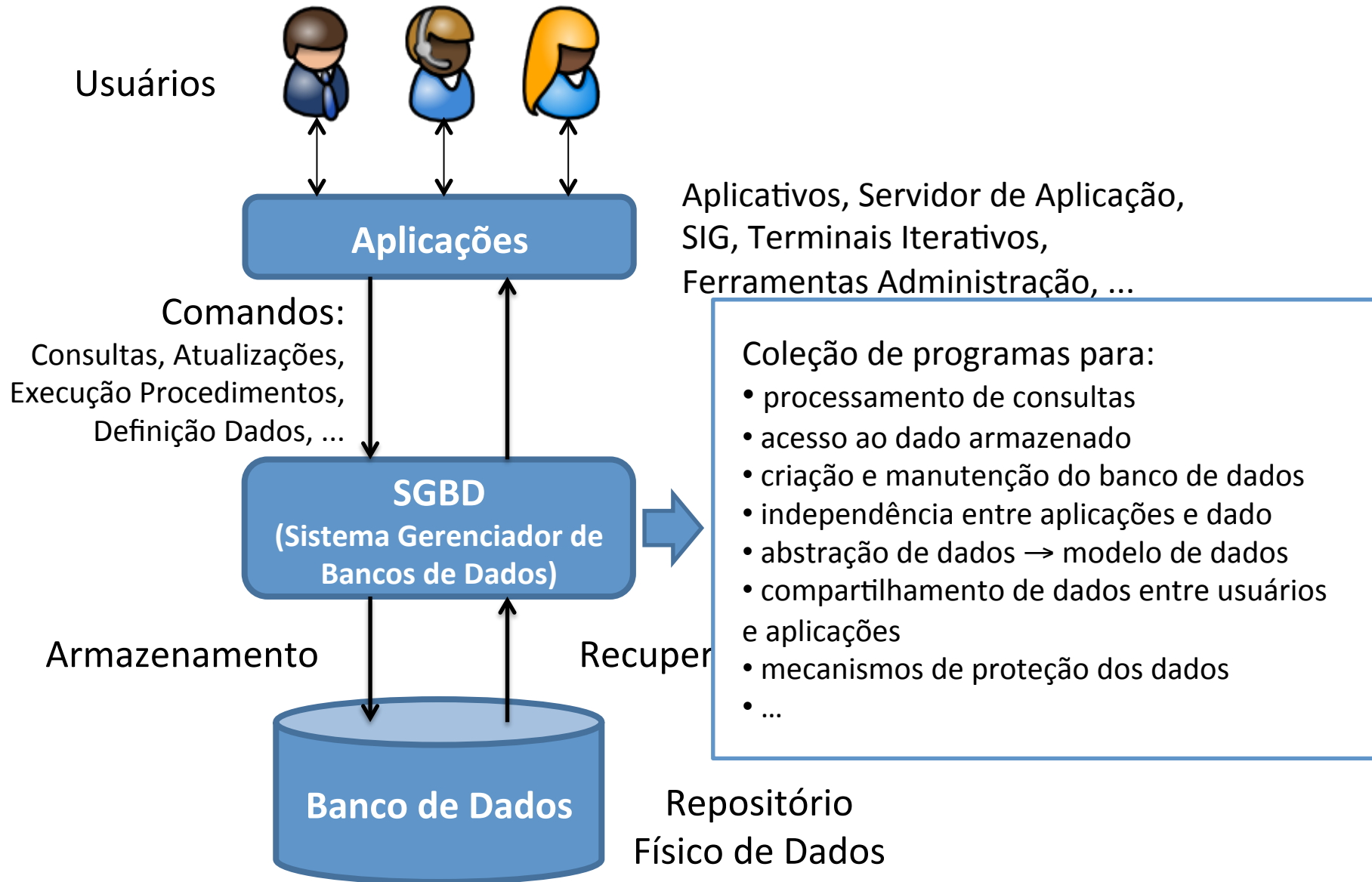
# Evolução das Tecnologias de Bancos Dados



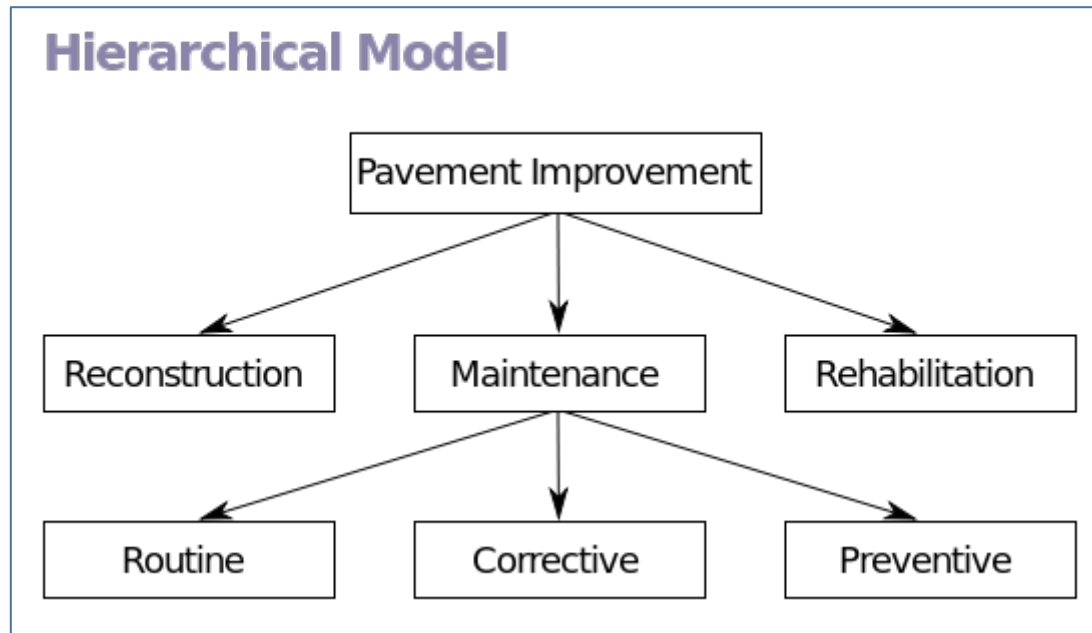
# Sistemas de Bancos de Dados



# Sistemas de Bancos de Dados



# Evolução das Tecnologias de Bancos Dados

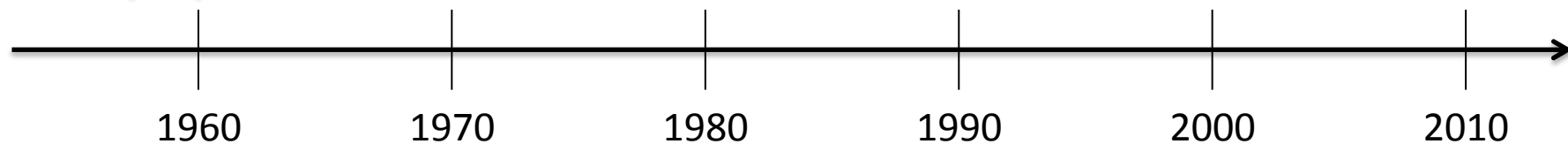


Fonte: [Wikipedia](#)

Programação Generalizada

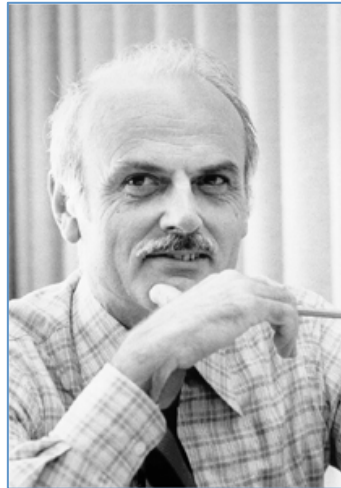
Programas dependentes arquivos

**Modelo Banco Dados Hierárquico (IBM IMS)**



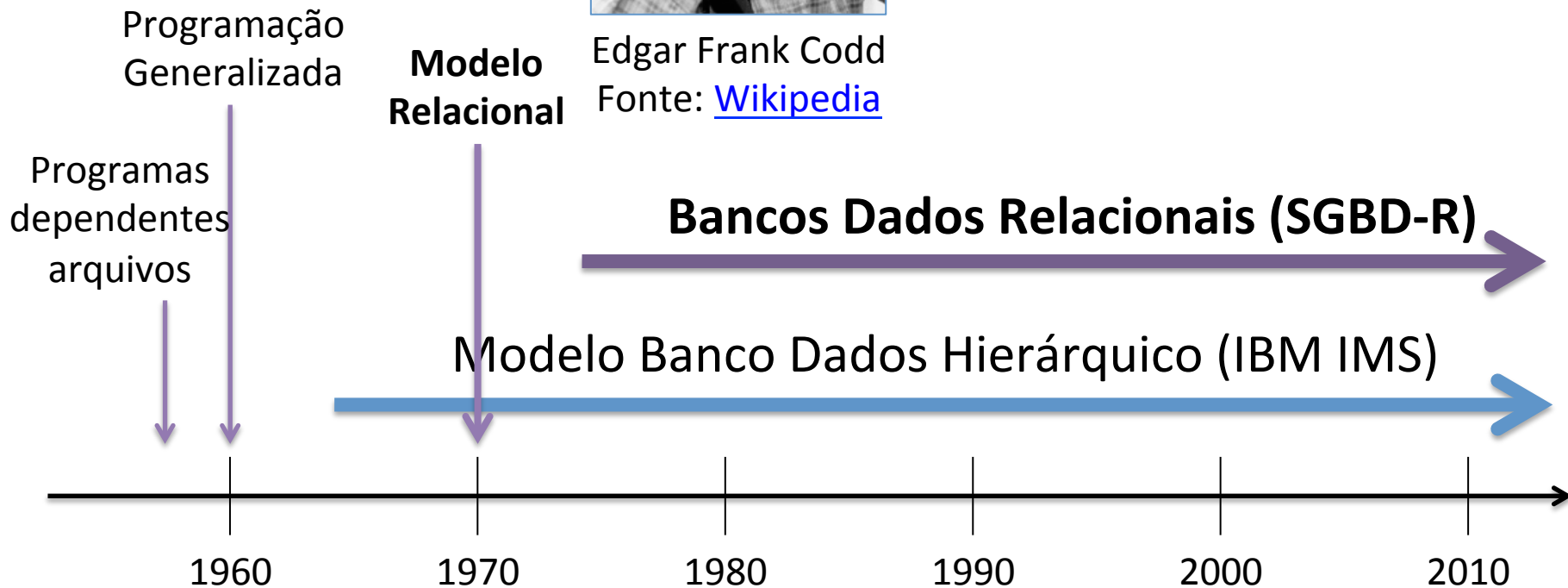
# Evolução das Tecnologias de Bancos Dados

ACM Turing Award (1981)



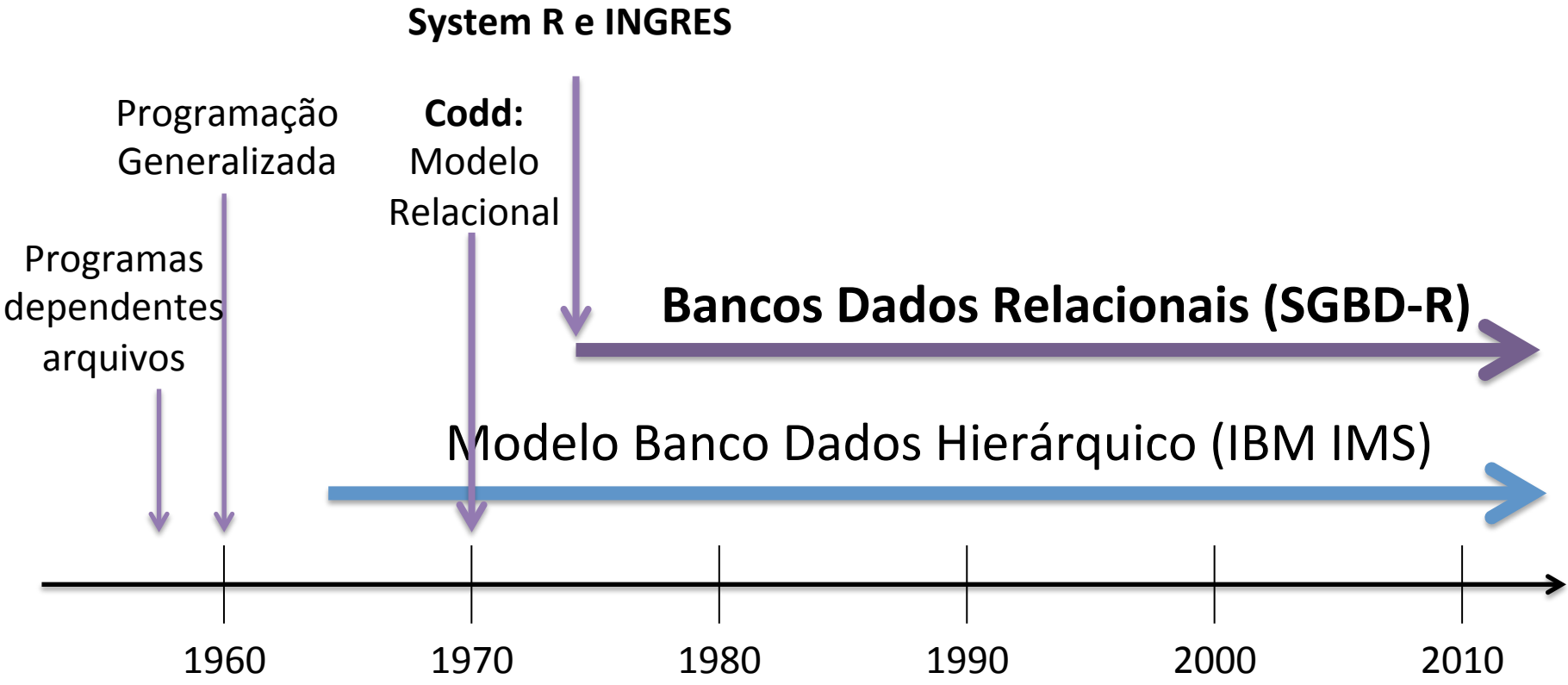
E. F. Codd. 1970. *A relational model of data for large shared data banks*. Communications of the ACM, v. 13, n. 6, June 1970, pp. 377-387.

Edgar Frank Codd  
Fonte: [Wikipedia](#)

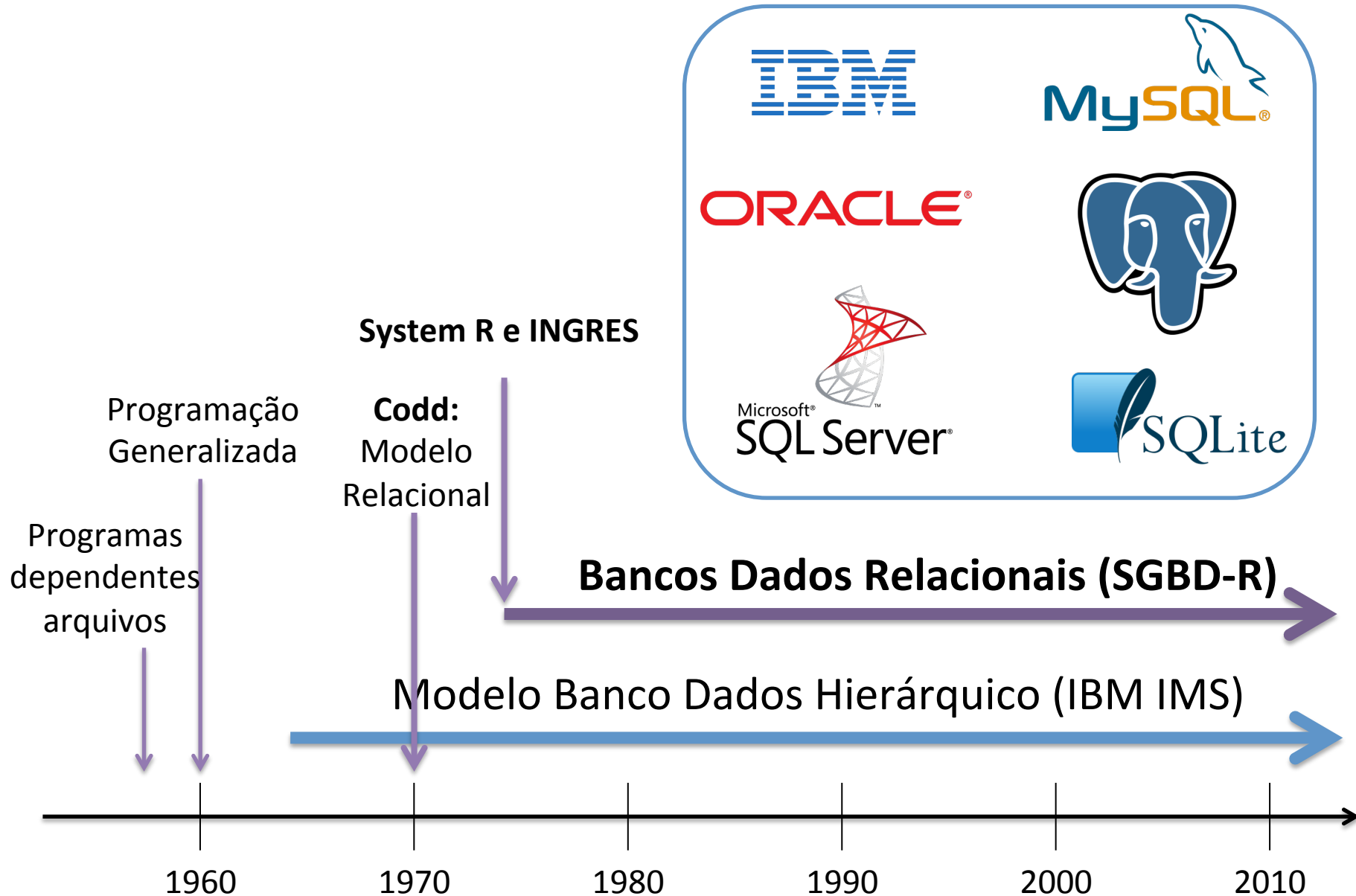




# Evolução das Tecnologias de Bancos Dados



# Evolução das Tecnologias de Bancos Dados



Quais são os principais conceitos em bancos de dados relacionais?

# Relação (ou Tabela)

- Um banco de dados relacional é organizado em uma coleção de relações (ou tabelas) possivelmente relacionadas entre si.



países			
id	nome	populacao	fronteira
1	Alemanha	82.000.000	
2	Brasil	190.000.000	
...	...	...	...

Diagrama explicativo da tabela:

- Tabela**: Indica o cabeçalho da tabela "países".
- Colunas**: Indica as colunas "id", "nome", "populacao" e "fronteira".
- Linha**: Indica uma linha de dados, como a linha com "1", "Alemanha", "82.000.000" e o mapa da Alemanha.
- Esquema Tabela**: Indica a estrutura da tabela, incluindo o cabeçalho e as colunas.
- Instância**: Indica os dados reais armazenados na tabela, como as linhas com "1", "2" e "...".

# Modelo Relacional

- Toda tabela (ou relação) possui um nome:
  - Em geral, esse nome é único dentro de um mesmo banco de dados.\*
- As colunas de uma tabela são também chamadas de:
  - campos, domínios ou atributos.
- Cada coluna possui um nome e deve ter um tipo de dado associado:
  - Numérico, Cadeia de Caracteres, Data e Hora, Geométrico.
- As linhas também são conhecidas por:
  - tuplas ou registros.

\* Conforme veremos mais adiante os SGBD-R podem relaxar esta afirmação com o uso de esquemas (ou *namespaces*)

# Relacionamentos entre tabelas

países		
id	nome	populacao
1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...

cidades			
cid	nome	populacao	pais_id
191	Correntina	...	2
181	Ouro Preto	...	2
987	Munster	...	1
192	Belmonte	...	2
...	...	...	...

países_x_cidades					
pid	p_nome	p_populacao	cid	c_nome	c_populacao
2	Brasil	190000000	191	Correntina	...
2	Brasil	190000000	181	Ouro Preto	...
1	Alemanha	82000000	987	Munster	...
2	Brasil	190000000	192	Belmonte	...
...	...	...	...	...	...

# Chave Primária (Primary Key)

- Campo ou conjunto de campos cujos valores identificam unicamente cada linha de uma tabela.

**Chave Primária** →

países		
id	nome	populacao
1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...

**Chave Primária** →

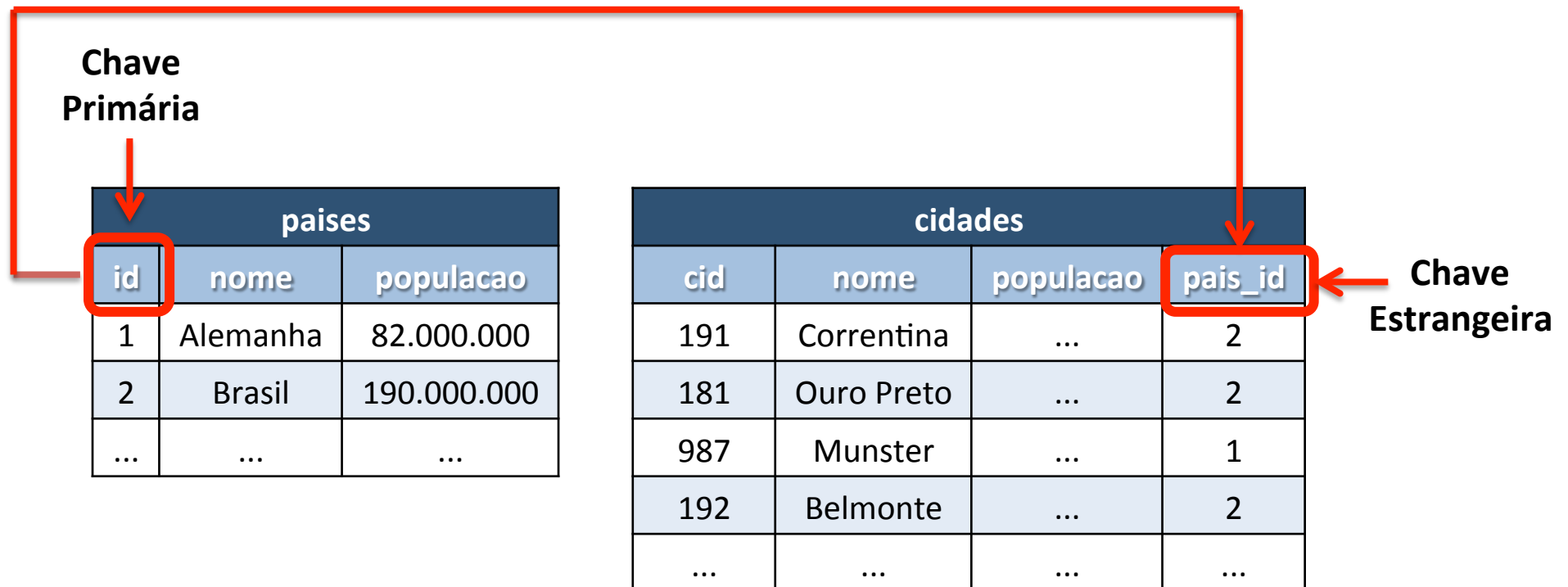
cidades			
cid	nome	populacao	pais_id
191	Correntina	...	2
181	Ouro Preto	...	2
987	Munster	...	1
192	Belmonte	...	2
...	...	...	...

**Chave Primária  
Composta**

cliente_telefone		
ncid	fone	tipo
1	555-7654	residencial
1	345-9876	comercial
2	888-7777	residencial

# Chave Estrangeira (Foreign Key)

- Coluna ou combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma outra tabela\*.



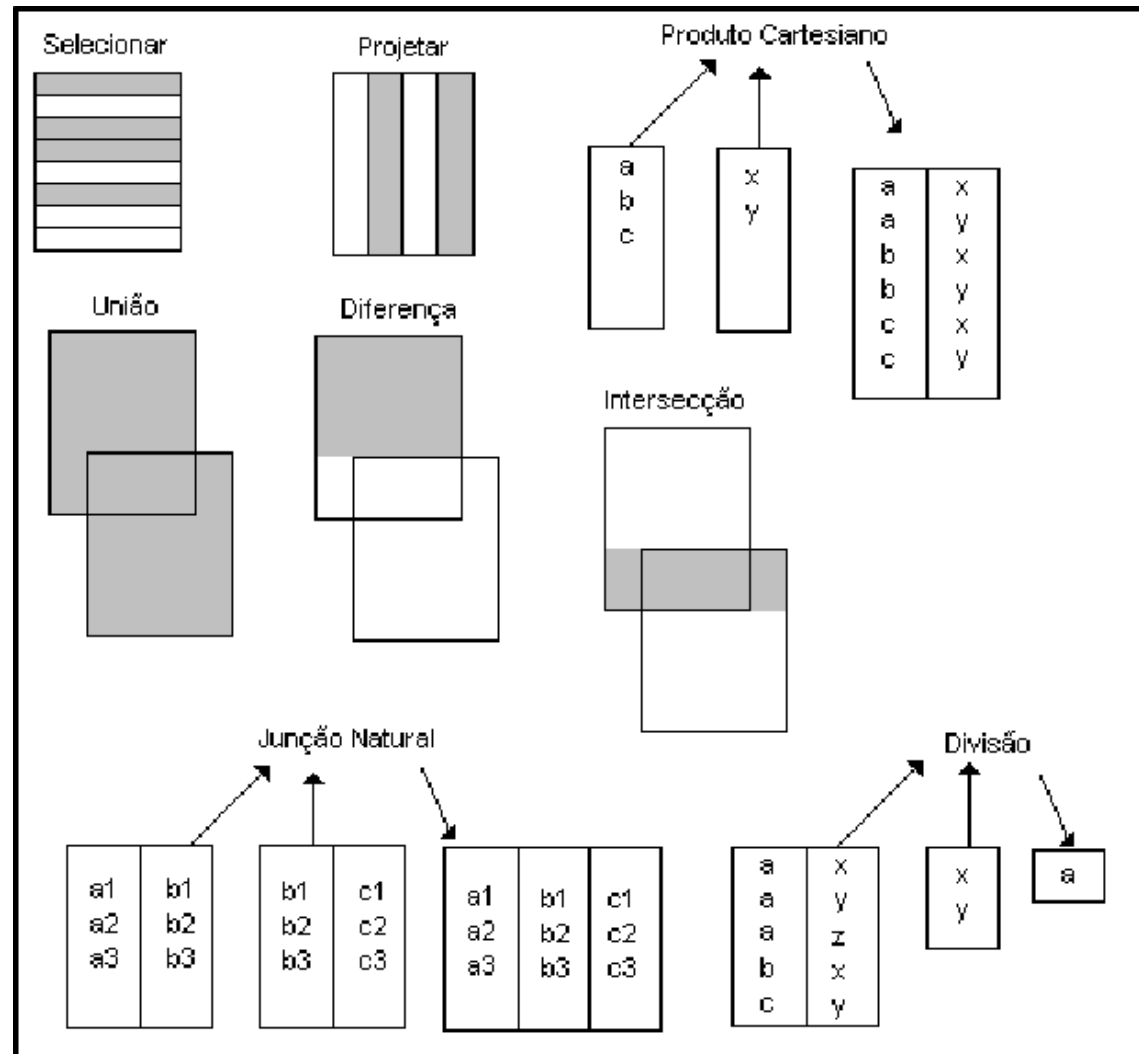
\*uma chave estrangeira não precisa ter o mesmo nome do que a chave primária correspondente na outra tabela (apenas o mesmo domínio)



# Álgebra Relacional

- Linguagem formal de consulta.
- Conjunto de operações que usam uma ou mais relações como entrada e geram uma nova relação de saída:
  - operação  $(R_1) \rightarrow R_n$
  - operação  $(R_1, R_2) \rightarrow R_n$
- Operações básicas:
  - Operações unárias: seleção, projeção.
  - Operações binárias: produto cartesiano, junção, interseção, união e diferença.
- Os operadores podem ser combinados de forma a realizar operações mais complexas.

# Álgebra Relacional: Operadores



Fonte: C. J. Date (1993)

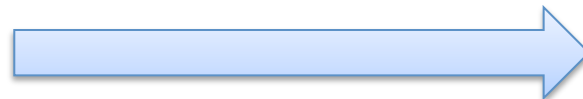
# Álgebra Relacional: Seleção

- Este operador seleciona tuplas (linhas) de uma relação que satisfazem um certo predicado ou condição.
- Exemplo: para a relação “países”, selecionar as tuplas cuja população seja maior que 100.000.000.

países		
id	nome	populacao
1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...

Tabela de Entrada

$\sigma_{populacao \geq 10^8}(países)$



nova_relacao		
id	nome	populacao
2	Brasil	190.000.000
...	...	...

Tabela de Saída

# Álgebra Relacional: Projeção

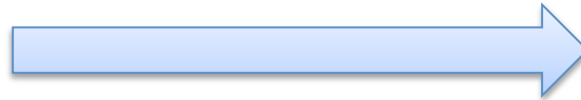
- Este operador gera uma nova relação contendo apenas as colunas desejadas de uma relação de entrada.
- Exemplo: projetar o atributo nome sobre a relação “países”.

países		
id	nome	populacao
1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...



Tabela de Entrada

$\pi_{nome}(países)$



nova_relacao
nome
Alemanha
Brasil
...



Tabela de Saída

# Álgebra Relacional: Produto Cartesiano

- Este operador gera uma nova relação formada pela combinação de todas as tuplas de duas relações de entrada.

$$(países) \times (cidades)$$

nova_relacao						
id	nome	populacao	cid	nome	populacao	pais_id
1	Alemanha	82000000	191	Correntina	...	2
1	Alemanha	82000000	181	Ouro Preto	...	2
1	Alemanha	82000000	987	Munster	...	1
1	Alemanha	82000000	192	Belmonte	...	2
2	Brasil	190.000.000	191	Correntina	...	2
2	Brasil	190.000.000	181	Ouro Preto	...	2
2	Brasil	190.000.000	987	Munster	...	1
2	Brasil	190.000.000	192	Belmonte	...	2
...	...	...	...	...	...	...

# Álgebra Relacional: Junção (Join)

- Produto cartesiano seguido de uma seleção.

$$(paises)\theta(cidades) \Leftrightarrow \sigma_{paises.id=cidades.pais\_id}(paises \times cidades)$$

nova_relacao						
id	nome	populacao	cid	nome	populacao	pais_id
1	Alemanha	82000000	987	Munster	...	1
2	Brasil	190.000.000	191	Correntina	...	2
2	Brasil	190.000.000	181	Ouro Preto	...	2
2	Brasil	190.000.000	192	Belmonte	...	2
...	...	...	...	...	...	...

# Linguagem de Consulta: SQL

- O modelo relacional (Codd, 1970) é a base para linguagens de alto nível:
  - Álgebra/Cálculo Relacional → Linguagem Declarativa → ISO/SQL (Structured Query Language)

**CREATE TABLE** pais

```
(  
id          INT4 PRIMARY KEY,  
nome       VARCHAR(50),  
populacao  INT4  
);
```

Definição Dados

pais		
id	nome	populacao

Manipulação  
Dados

pais		
id	nome	populacao
1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...

Manipulação  
Dados

```
INSERT INTO pais  
VALUES (1, 'Alemanha', 82000000)
```

```
INSERT INTO pais  
VALUES (2, 'Brasil', 190000000)
```

# Linguagem de Consulta: SQL

- O modelo relacional (Codd, 1970) é a base para linguagens de alto nível:
  - Álgebra/Cálculo Relacional → Linguagem Declarativa → ISO/SQL (Structured Query Language)

países		
id	nome	populacao
1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...

**Consulta**  
**(Não-Procedural)**

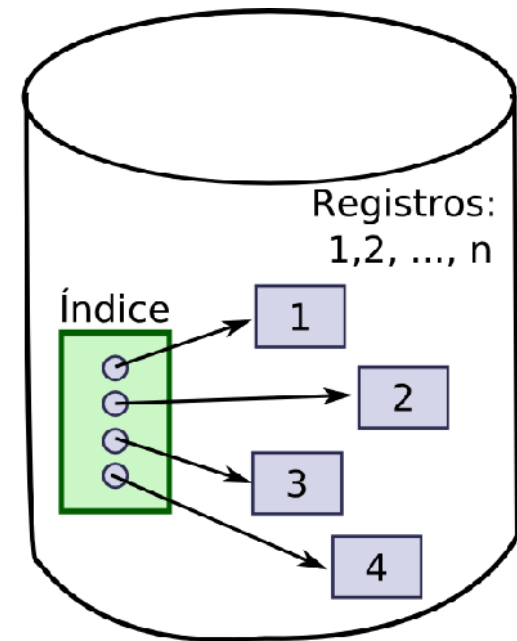
```
SELECT nome  
FROM países  
WHERE populacao > 80000000
```

**Nota:** stored procedures ou procedural languages: PL/SQL, T-SQL, PL/pgSQL



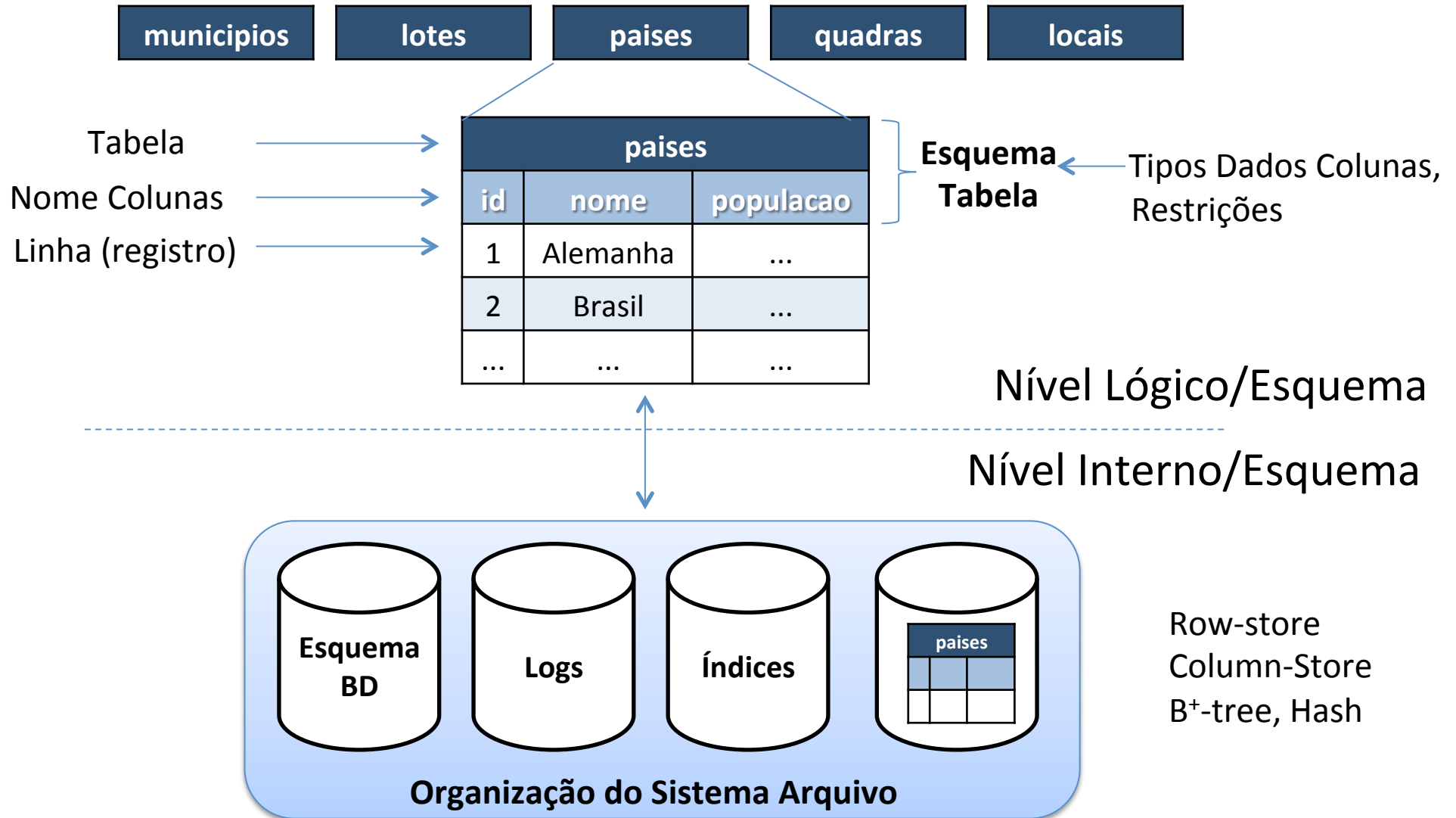
# Métodos de Acesso (Indexação)

- Problema: Como processar de forma eficiente as consultas?
  - Através do uso de estruturas de dados conhecidas como Índices ou Métodos de Acesso;
- Os índices reduzem o conjunto de objetos a serem verificados durante o processamento das consultas:
  - Normalmente, uma consulta envolve apenas uma pequena parcela do banco de dados;
  - Neste caso, percorrer todo o banco pode ser bastante ineficiente;
  - Portanto, um plano de execução eficiente para a consulta tipicamente considera a existência de índices.



Registros de um arquivo e o índice associado a este arquivo

# Independência Física dos Dados



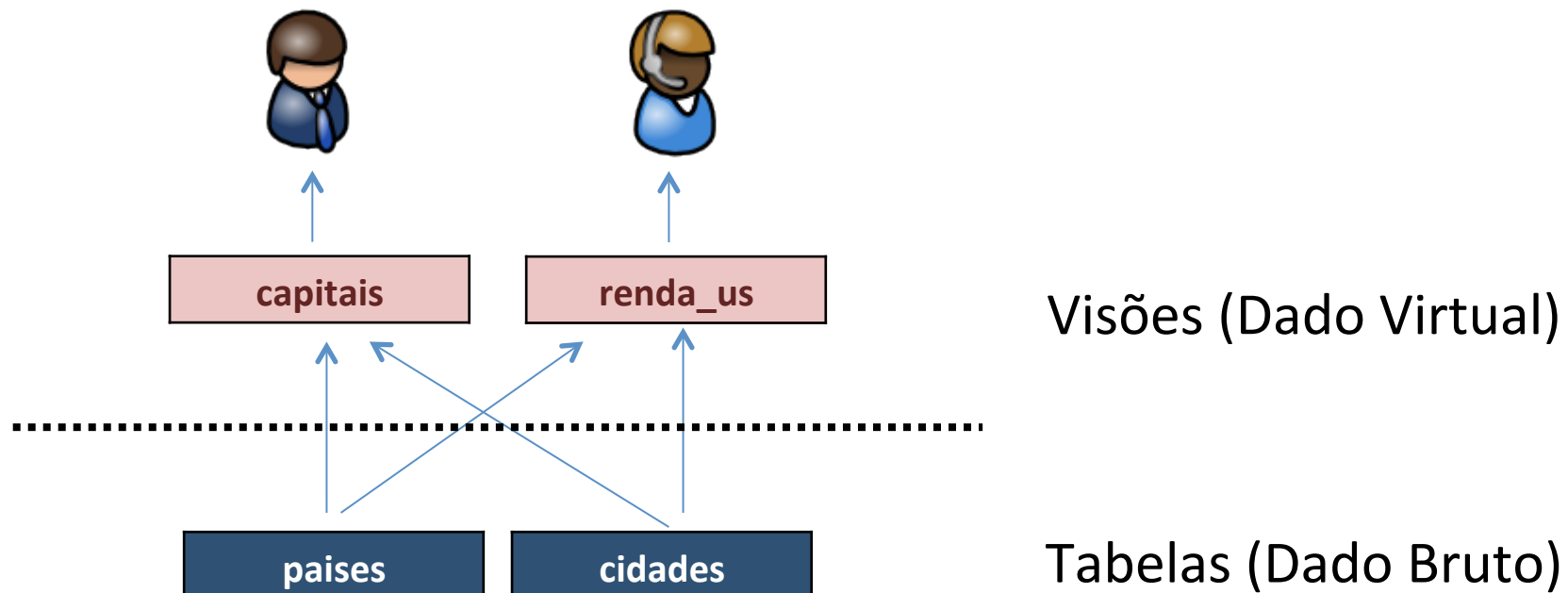
Fonte: Adaptado de Gray (1996)

# Como a independência física é alcançada?

- Esquema do Banco de Dados:
  - Uma característica fundamental de um SGBD-R é que ele não contém apenas os dados brutos sobre o domínio de interesse;
  - Todo SGBD-R mantém a definição ou descrição da estrutura do banco de dados (*self-describing*);
  - Essas informações são mantidas no catálogo do sistema (ou dicionário do sistema) e são denominadas de metadados do banco de dados.
  - Na prática os SGBD-R armazenam essas informações de definição em tabelas do próprio sistema (tabelas de metadado ou tabelas do catálogo).
- O modelo de dados relacional fornece para as aplicações uma abstração independente da representação física dos dados.

# Visões (Views)

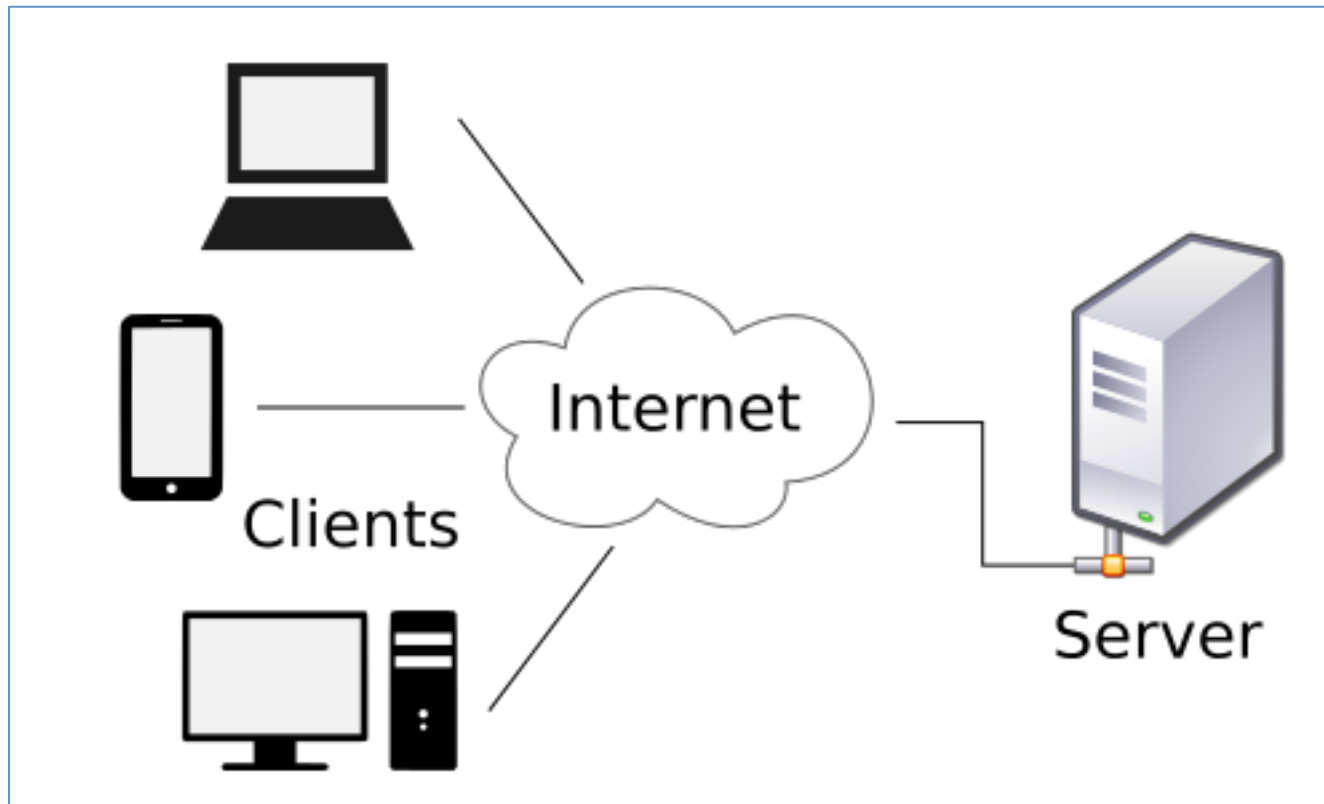
- Muitas vezes pode ser necessário fornecer diferentes perspectivas do banco de dados dependendo do usuário.  
Uma visão (ou view) pode ser:
  - um subconjunto dos dados do banco de dados
  - pode conter dados derivados do banco de dados



# Arquiteturas de SGBD-R

- Cliente/Servidor
- Embutido (ou embarcado)
- Em memória (In-memory)
- Paralelos/Distribuídos
- Armazenamento Linha x Coluna

# Arquiteturas de SGBD-R: Cliente Servidor



Fonte: [Wikipedia](https://pt.wikipedia.org/wiki/Arquitetura_cliente-servidor)

# Arquiteturas de SGBD-R: Embedded

```
#include <sqlite3.h>

int main(int argc, char** argv) {
    int rc = sqlite3_open("/opt/data/mydb.sqlite", &db);

    if(rc) {
        fprintf(stderr, "Can't open database: %s\n", sqlite3_errmsg(db));
        sqlite3_close(db);
        return EXIT_FAILURE;
    }

    rc = sqlite3_exec(db, "Select * from tabela", callback, 0, &zErrMsg);

    if( rc!=SQLITE_OK ){
        char* zErrMsg = 0;
        fprintf(stderr, "SQL error: %s\n", zErrMsg);
        sqlite3_free(zErrMsg);
    }

    sqlite3_close(db);
    return EXIT_SUCCESS;
}
```



# Arquiteturas de SGBD-R: row x column store

países		
id	nome	populacao
1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...

## Row Store

Layout em Disco

1	Alemanha	82M	2	Brasil	190M
---	----------	-----	---	--------	------

3	Argentina	...	...	...	...
---	-----------	-----	-----	-----	-----

Ex: PostgreSQL, MySQL

## Column Store

Layout em Disco

1	Alemanha	82.000.000
2	Brasil	190.000.000
...	...	...

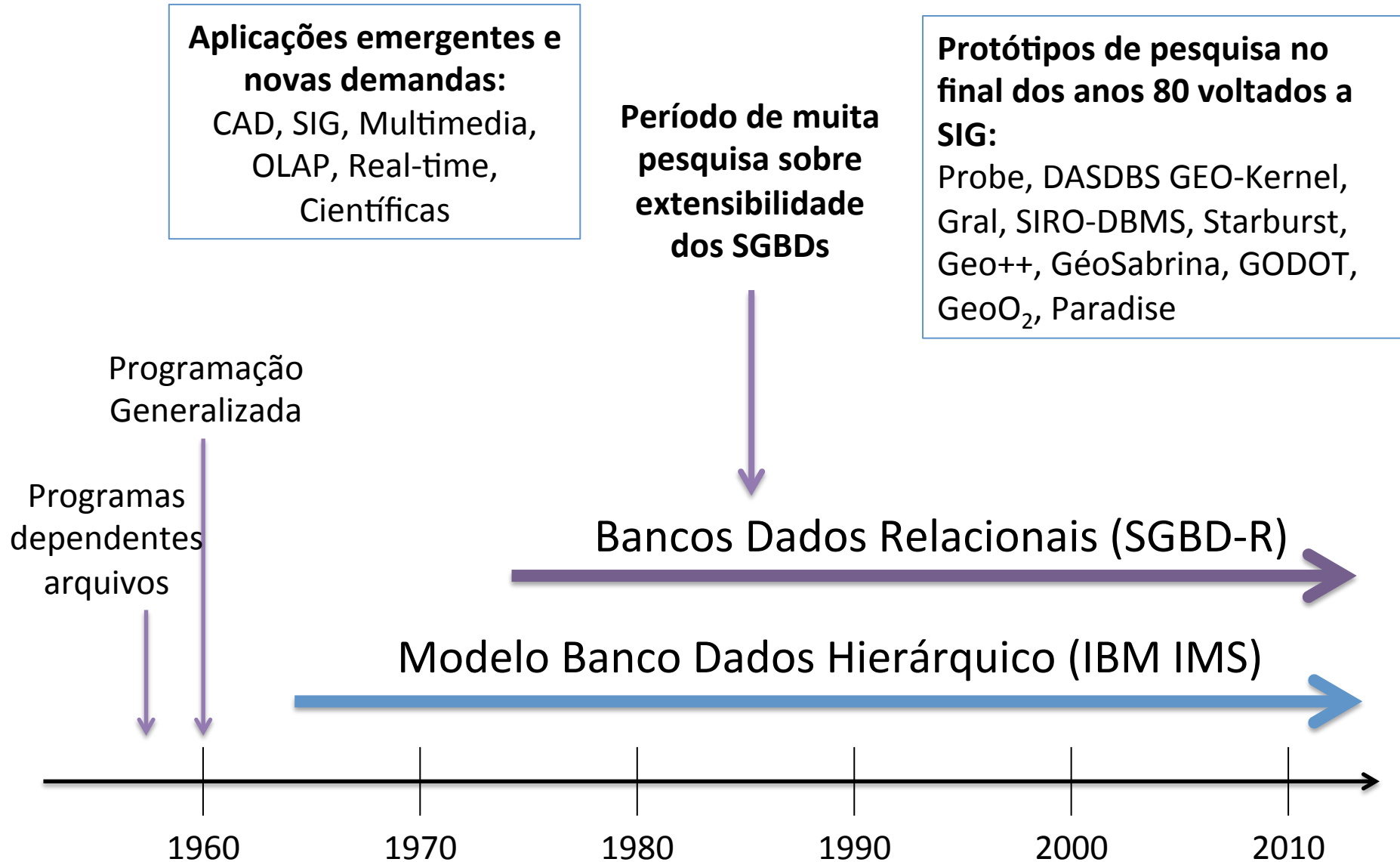
Ex: C-Store, MonetDB, Vertica



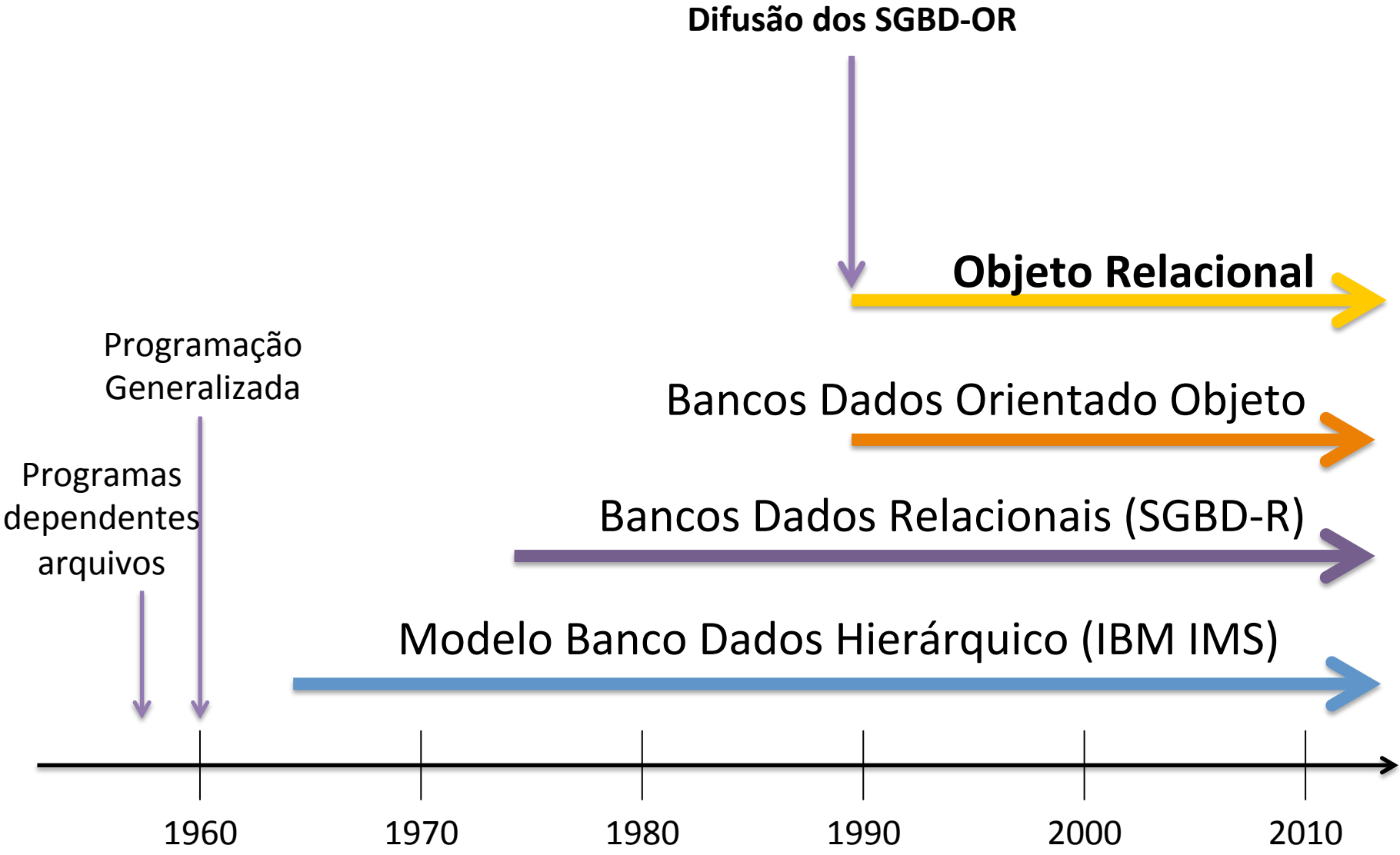
# Outros Conceitos Importantes

- Projeto de bancos de dados:
  - Modelo Entidade-Relacionamento (ER).
- Normalização:
  - Evitar anomalias com o projeto do banco de dados.
- Transações (ACID).
- Gatilhos (Trigger).
- Procedimentos Armazenados (Stored Procedure).

# Evolução das Tecnologias de Bancos Dados



# Evolução das Tecnologias de Bancos Dados



# SGBD-OR: User Defined Types (UDT)

```
CREATE TYPE geo_point AS  
(  
  x REAL,  
  y REAL,  
  srid INTEGER  
);
```

# SGBD-OR: User Defined Types (UDT)

```
CREATE TABLE sedes_municipais  
(  
  id INTEGER PRIMARY KEY,  
  location GEO_POINT  
);
```



```
INSERT INTO sedes_municipais  
VALUES (1, '(1, 2, 4326)::GEO_POINT);
```

# SGBD-OR: User Defined Functions (UDF)

- Possibilita criar ou estender a álgebra de um determinado tipo de dado.

```
CREATE OR REPLACE FUNCTION less_than(first GEO_POINT, second GEO_POINT)
RETURNS REAL
AS $$
BEGIN
    IF(first.x < second.x)
    THEN
        RETURN TRUE;
    END IF;

    IF(first.x > second.x)
    THEN
        RETURN FALSE;
    END IF;

    ...
    RETURN FALSE;
END;
$$
LANGUAGE plpgsql;
```

# SGBD-OR: User Defined Functions (UDF)

- Possibilita criar ou estender a álgebra de um determinado tipo de dado.

```
CREATE OR REPLACE FUNCTION distance(first GEO_POINT, second GEO_POINT)
RETURNS REAL
AS $$
DECLARE
    dx REAL;
    dy REAL;
BEGIN
    dx = (first.x - second.x) * (first.x - second.x);

    dy = (first.y - second.y) * (first.y - second.y);

    RETURN sqrt(dx + dy);
END;
$$
LANGUAGE plpgsql;
```

# SGBD-OR: User Defined Functions (UDF)

- UDFs passam a fazer parte da linguagem de consulta do SGBD:

```
SELECT less_than('(1, 2, 4326)::GEO_POINT, '(10, 20, 4326)::GEO_POINT);
```

```
SELECT less_than('(1, 2, 4326)::GEO_POINT, '(-1, 2, 4326)::GEO_POINT);
```

```
SELECT distance('(1, 2, 4326)::GEO_POINT, '(10, 20, 4326)::GEO_POINT);
```



# SGBD-OR: Sobrecarga de Operadores

```
CREATE OPERATOR <  
(  
  leftarg = GEO_POINT,  
  rightarg = GEO_POINT,  
  procedure = less_than,  
  commutator = >,  
  negator = >=  
);
```



```
SELECT '(1, 2, 4326)'::GEO_POINT < '(10, 2, 4326)'::GEO_POINT;
```

# SGBD-OR: User Defined Access Methods

B-tree

Operation	Strategy Number
less than	1
less than or equal	2
equal	3
greater than or equal	4
greater than	5

Hash

Operation	Strategy Number
equal	1

GiST – Rtree 2D

Operation	Strategy Number
strictly left of	1
does not extend to right of	2
overlaps	3
does not extend to left of	4
strictly right of	5
same	6
contains	7
contained by	8
does not extend above	9
strictly below	10
strictly above	11
does not extend below	12

# SGBD-OR: UDTs mais Complexos

```
CREATE TYPE Geometry
(
  internallength = variable,
  input = geometry_in,
  output = geometry_out,
  send = geometry_send,
  receive = geometry_recv,
  typmod_in = geometry_typmod_in,
  typmod_out = geometry_typmod_out,
  delimiter = ':',
  alignment = double,
  analyze = geometry_analyze,
  storage = main);
```

```
CREATE OR REPLACE FUNCTION _ST_Touches(geom1 geometry, geom2 geometry)
  RETURNS boolean
  AS '$libdir/postgis-2.1','touches'
  LANGUAGE 'c' IMMUTABLE STRICT
  COST 100;
```

...

# Evolução das Tecnologias de Bancos Dados

PostgreSQL → PostGIS

MySQL → Spatial and Geodetic Geography Types

SQLite → SpatiaLite and RasterLite

Oracle → Oracle Spatial, GeoRaster, Topology and Network Models

IBM DB2 → Spatial Extender

SQL Server (2008) → Spatial Types

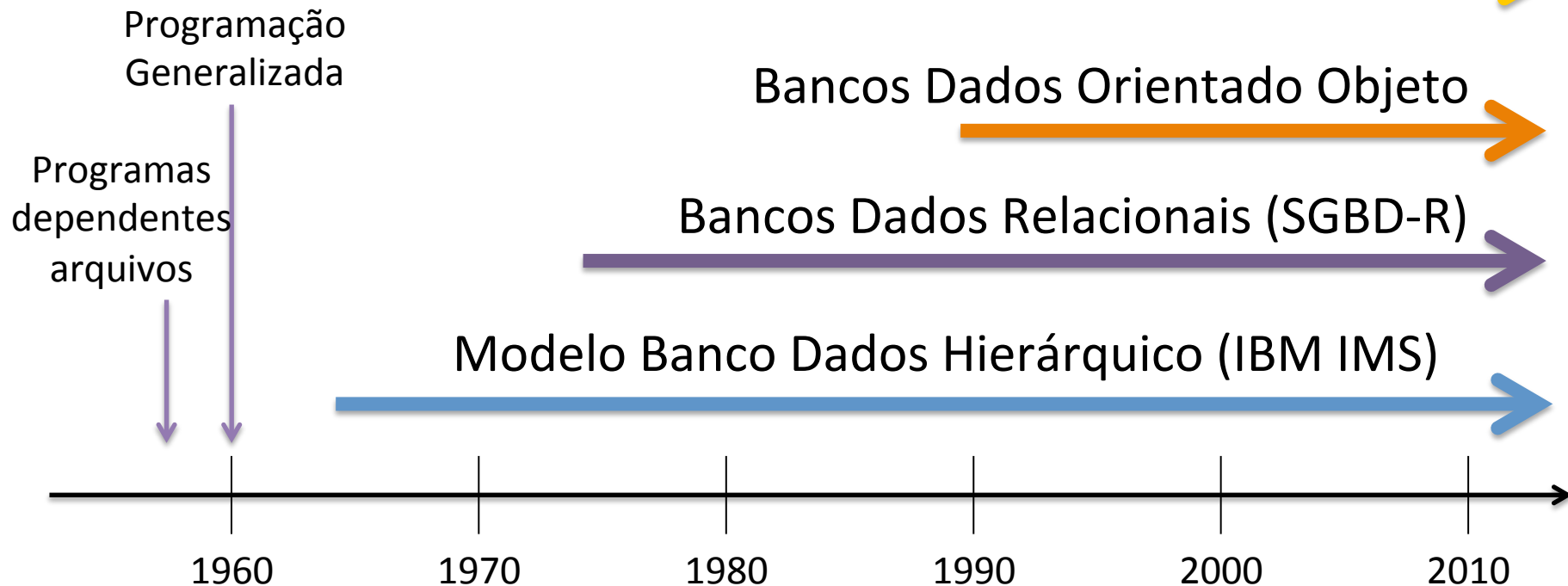
Geoespacial

Objeto Relacional

Bancos Dados Orientado Objeto

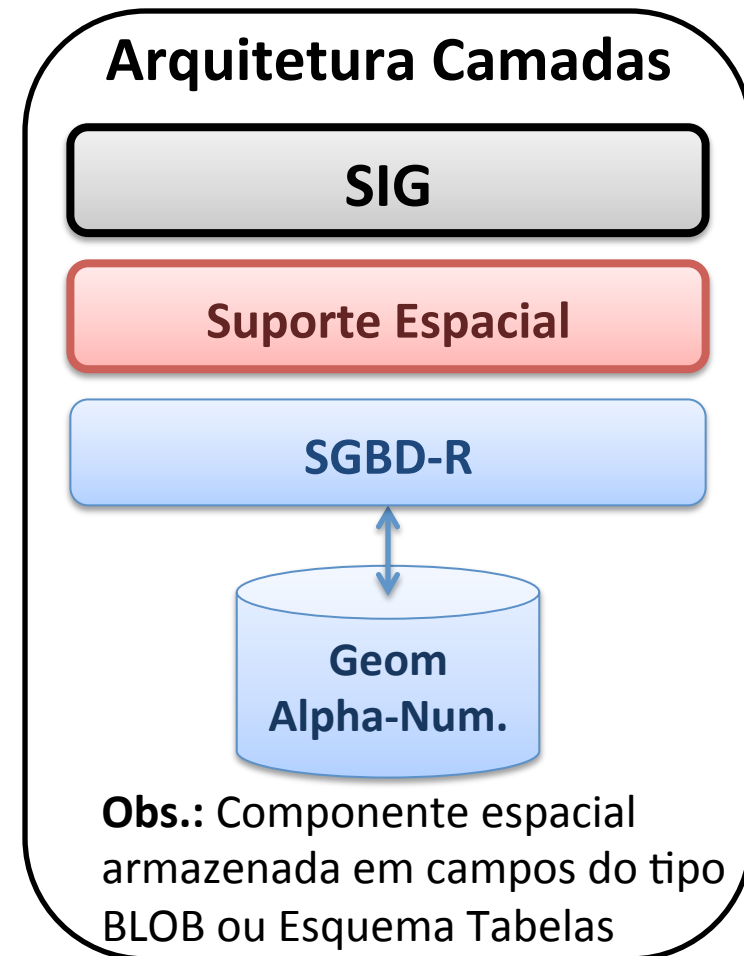
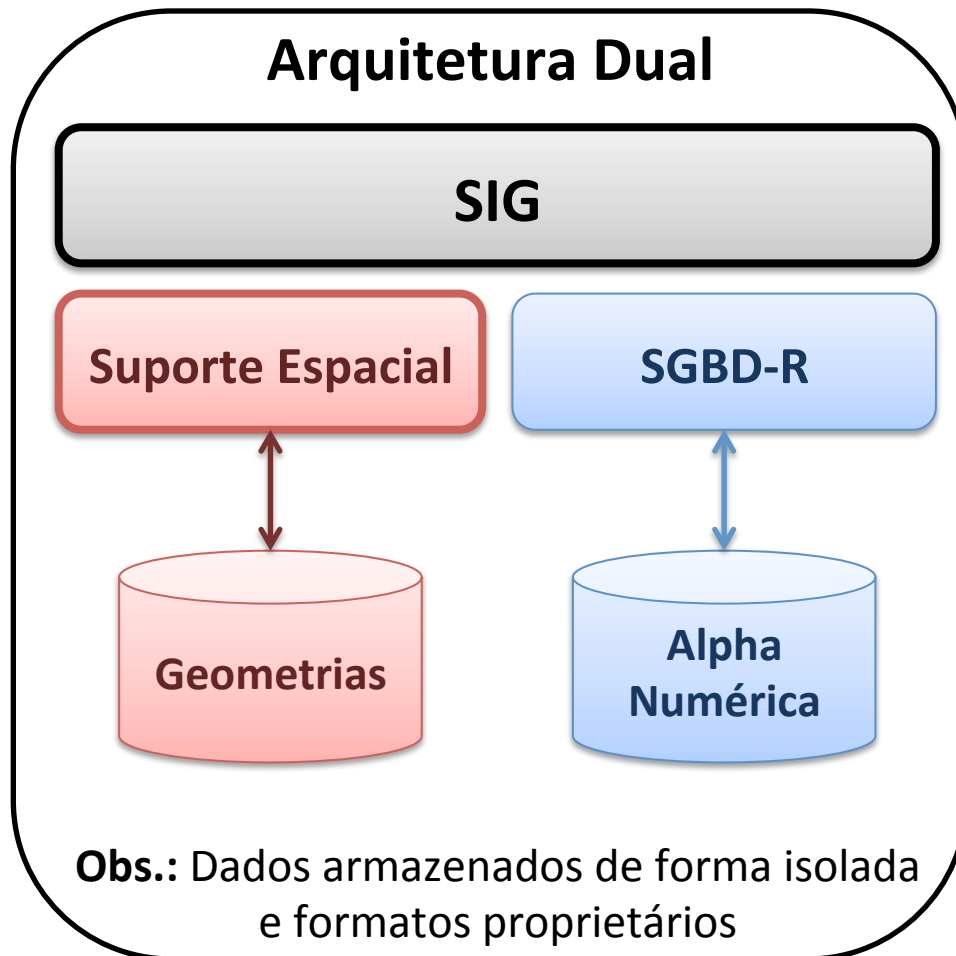
Bancos Dados Relacionais (SGBD-R)

Modelo Banco Dados Hierárquico (IBM IMS)



# SIG e SGBD-R



- Como era a integração SIG e SGBD-R antes da inclusão do suporte espacial?



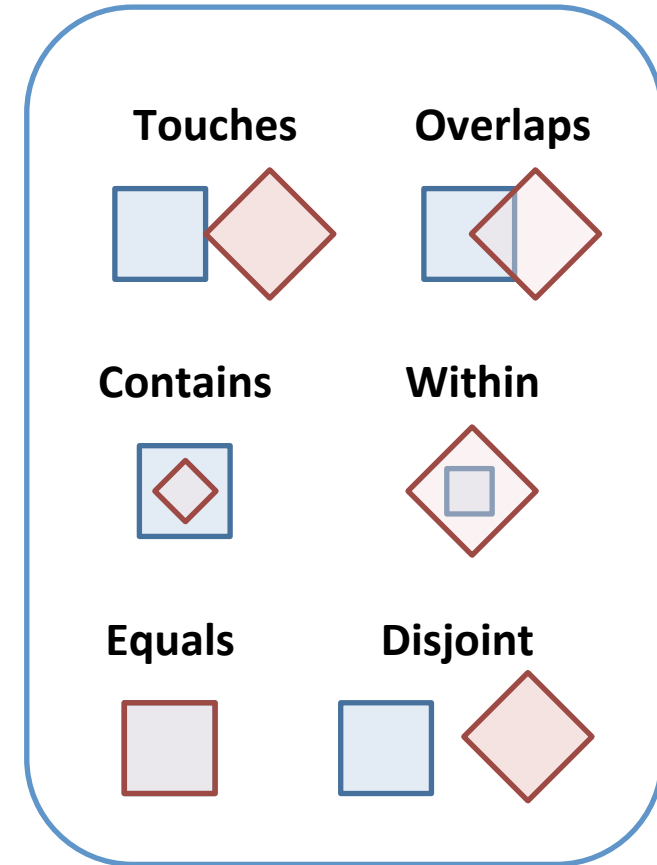
# SIG e SGBD-R: Como passou a ser esta integração?

- Arquitetura Integrada: Tipos de Dados Geoespaciais
- Padronização: OGC Simple Features e ISO/SQL-MM Spatial

Tabelas com feições: geometrias vetoriais

países			
id	nome	populacao	fronteira
1	Alemanha	82.000.000	
2	Brasil	190.000.000	
...	...	...	...

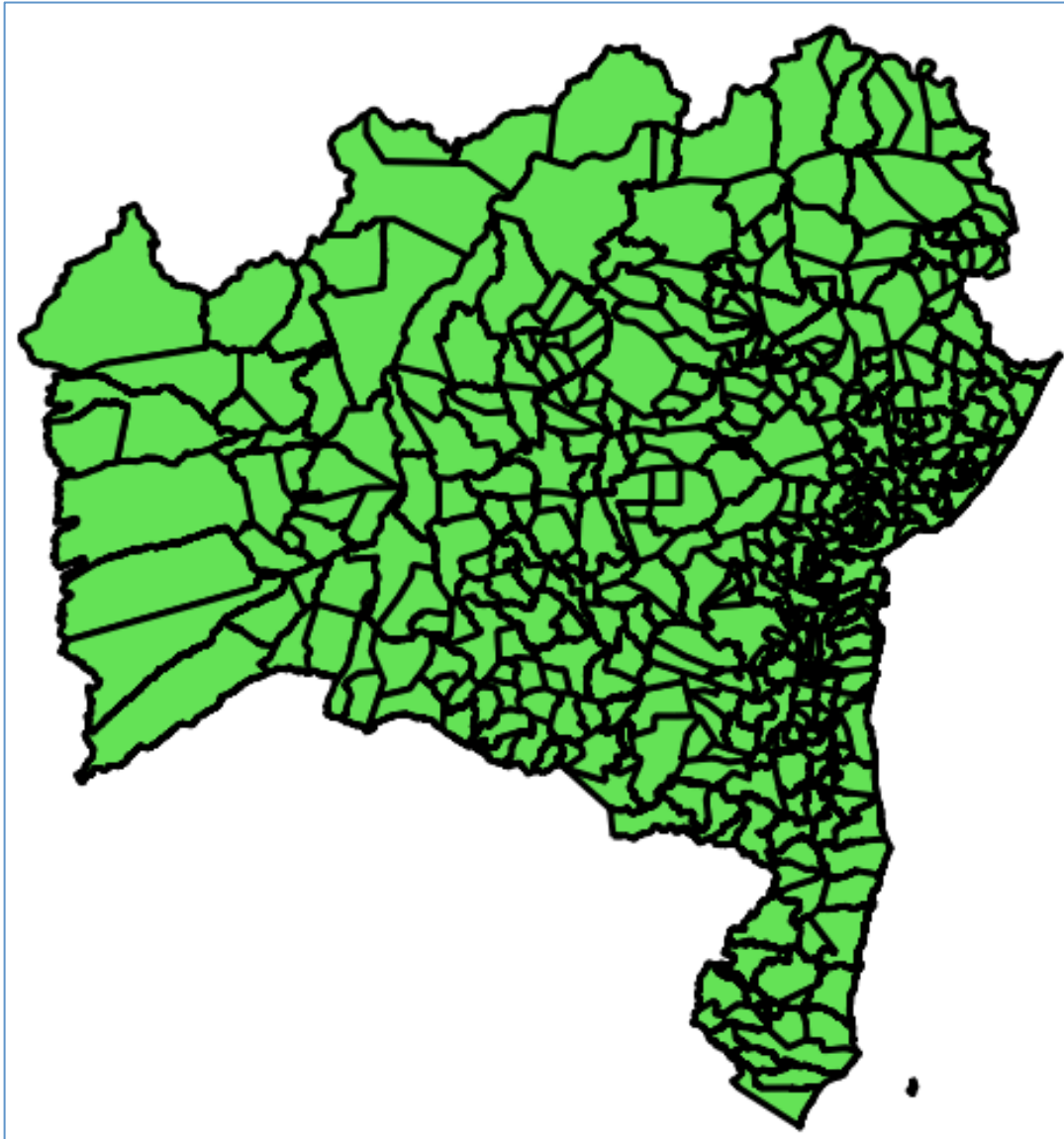
Operações espaciais



# SGBD-R com Suporte Espacial

Exemplo: PostGIS Geometry

# Tabela: ba\_municipios



**Arquivo:**

dados/shp/29mu2500gsr

**Tipo de dado:**

Polígonos (417)

**Sistema de Referência Espacial:**

4674 => Lat/Long SIRGAS 2000

**Nome da tabela a ser criada:**

ba\_municipios

**Codificação dos caracteres :**

LATIN1

**Fonte do dado:**

IBGE



# Tabela: terras\_indigenas



**Arquivo:**

dados/shp/LIM\_Terra\_Indigena\_A

**Tipo de dado:**

Polígonos (38)

**Sistema de Referência Espacial:**

4674 => Lat/Long SIRGAS 2000

**Nome da tabela a ser criada:**

terras\_indigenas

**Codificação dos caracteres :**

LATIN1

**Fonte do dado:**

IBGE

# Consulta Espacial

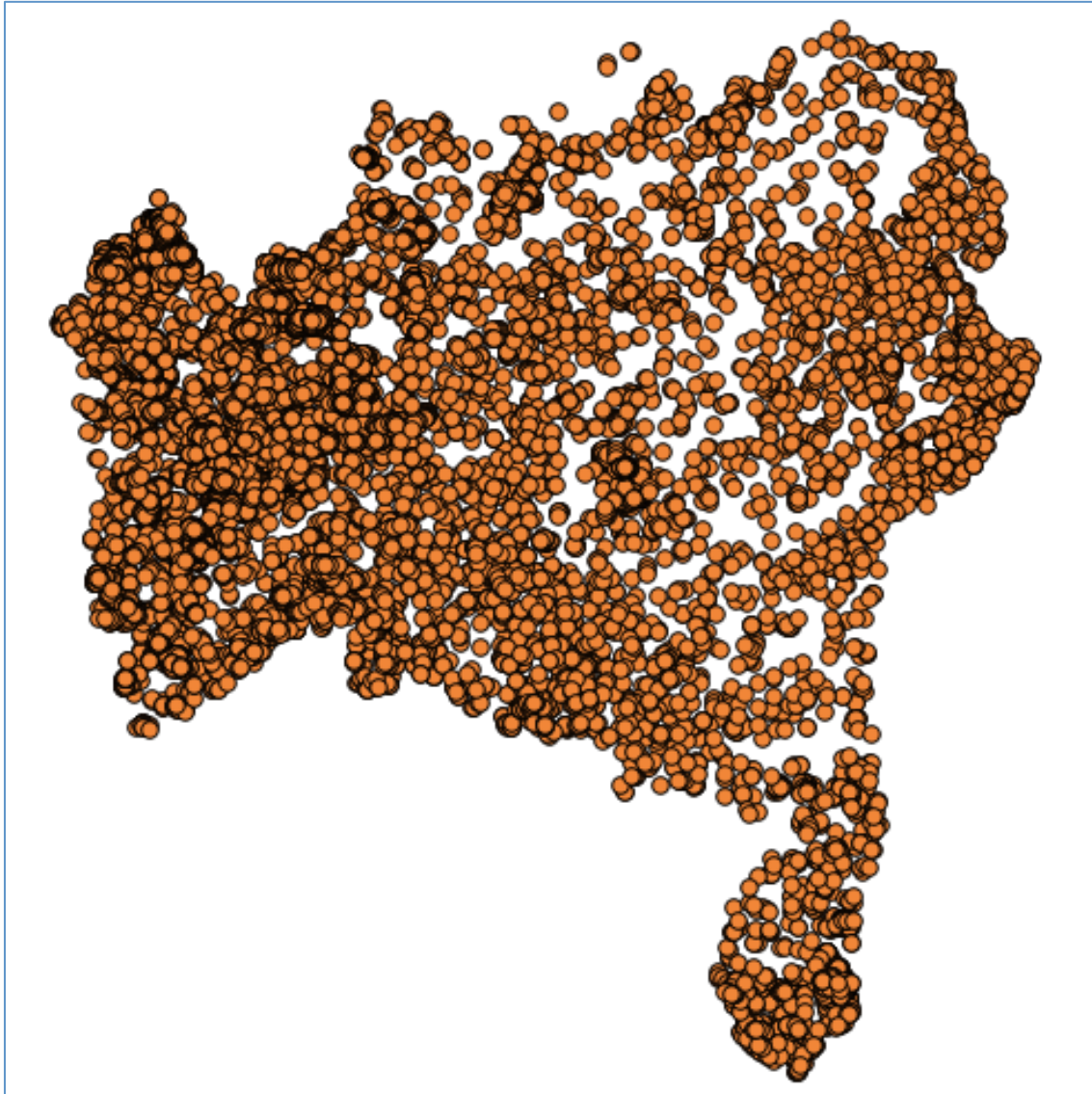
- **Pergunta:** Quais as áreas de terras indígenas na Bahia?

```
SELECT m.gid, m.nome_munic, t.nometi
FROM ba_municipios m, terras_indigenas t
WHERE ST_Intersects(m.geom, t.geom)
```

1	159	Ibotirama	Ibotirama
2	159	Ibotirama	Barra
3	269	Muquém de São Fran	Barra
4	371	Serra do Ramalho	Vargem Alegre
5	72	Camamu	Fazenda Bahiana
6	33	Banzaê	Kiriri
7	318	Quijingue	Kiriri
8	327	Ribeira do Pombal	Kiriri
9	394	Tucano	Kiriri
10	127	Euclides da Cunha	Massacara
11	70	Camacan	Caramuru/Paraguass
12	184	Itaju do Colônia	Caramuru/Paraguass
13	293	Pau Brasil	Caramuru/Paraguass
14	312	Potiraguá	Caramuru/Paraguass

15	313	Prado	Águas Belas
16	311	Porto Seguro	Barra Velha
17	311	Porto Seguro	Coroa Vermelha
18	339	Santa Cruz Cabrali	Coroa Vermelha
19	139	Glória	Pankararé
20	294	Paulo Afonso	Pankararé
21	333	Rodelas	Pankararé
22	139	Glória	Brejo do Burgo
23	294	Paulo Afonso	Brejo do Burgo
24	333	Rodelas	Brejo do Burgo
25	139	Glória	Kantaruré
26	339	Santa Cruz Cabrali	Mata Medonha
27	311	Porto Seguro	Imbiriba

# Tabela: focos



**Arquivo:**

dados/shp/focos\_incendio\_bahia

**Tipo de dado:**

Pontos (18072)

**Período:**

01-01-2013 a 22-09-2013

**Sistema de Referência Espacial:**

4618 => Lat/Long SAD/69

**Nome da tabela a ser criada:**

focos

**Codificação dos caracteres :**

LATIN1

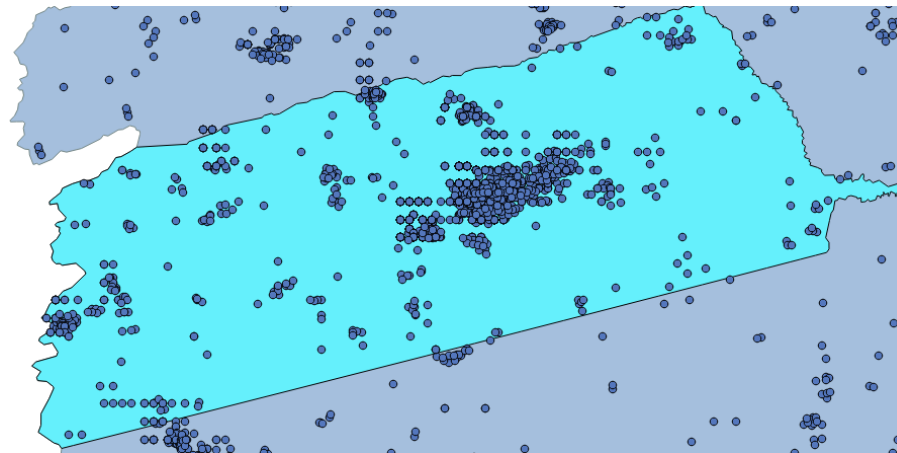
**Fonte do dado:**

INPE

# Consulta Espacial

- **Pergunta:** Em qual município da Bahia foi detectado maior número de focos?

```
SELECT m.gid, m.nome_munic, COUNT(*) AS num_focos
FROM ba_municipios m, focos f
WHERE ST_Contains(m.geom, ST_Transform(f.geom, 4674))
GROUP BY m.gid, m.nome_munic
ORDER BY num_focos DESC
LIMIT 1
```



R.

Correntina	2045
------------	------

# Consulta Espacial

- **Pergunta:** Algum foco de incêndio foi detectado em terras indígenas?

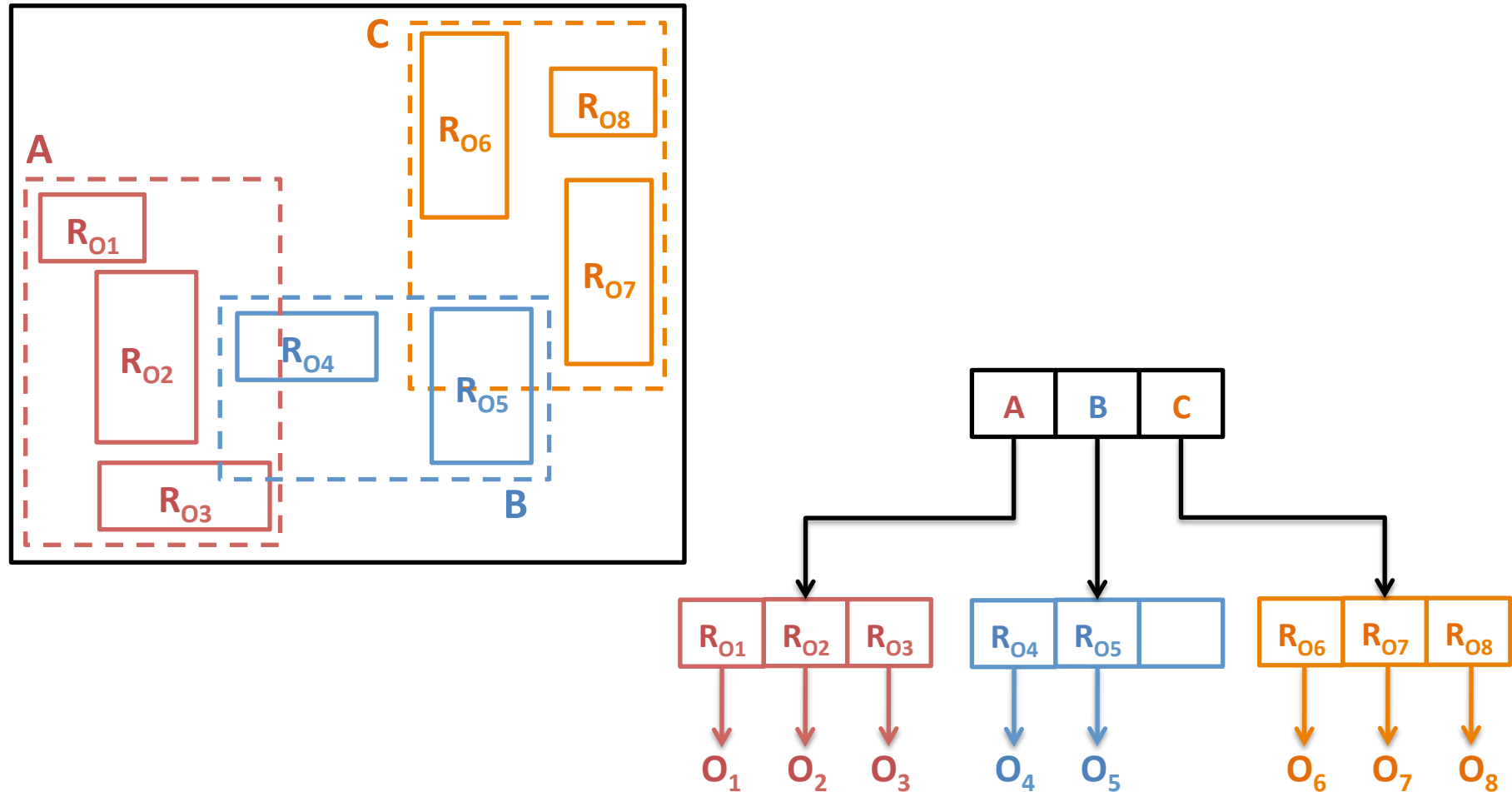
```
SELECT t.gid, MAX(nometi), COUNT(*) AS num_focos
FROM terras_indigenas t, focos f
WHERE ST_Contains(t.geom, ST_Transform(f.geom, 4674))
GROUP BY t.gid
```

	nome character varying(80)	num_focos bigint
1	Caramuru/Paraguass	2
2	Massacara	1
3	Pankararé	1
4	Kiriri	2
5	Águas Belas	7
6	Brejo do Burgo	1
7	Barra Velha	31

O que mais existe nesta integração entre  
SGBD-R e Dados Geográficos?

# Índices Espaciais (Árvores-R, Quadrees, Fixed-Grid)

Exemplo: R-Tree:



O que mais existe nesta integração entre SGBD-R e Dados Geográficos?

**Discussão:** Como Armazenar e Gerenciar Dados Matriciais?



**Solução 1:** Usando um SGBD-R com  
suporte matricial

PostGIS Raster

Oracle GeoRaster

**Solução 1:** Usando um SGBD-R com suporte matricial

PostGIS Raster



Oracle GeoRaster

# PostGIS Raster

Raster (Matricial - Células Multibandas)

Visão do espaço na forma de uma grade retangular,  
com células contendo uma ou mais valores numéricos

# PostGIS Raster

- O projeto do PostGIS Raster possibilita trabalhar com vários casos de uso com imagens:
  - Armazém de imagens (possivelmente não relacionados)
  - Tiles:
    - Regulares ou irregulares (pode ter missing tiles)
  - *in db x out db storage*
- Novas visões com metadados das imagens:
  - raster\_columns
  - raster\_overviews

# PostGIS Raster → SQL

- Table creation:

```
CREATE TABLE raster_table  
( rid SERIAL PRIMARY KEY,  
  rast RASTER  
);
```

- Spatial index creation:

```
CREATE INDEX spidx_table_col ON raster_table  
  USING gist(ST_ConvexHull(raster-col));
```

# Organizando uma imagem em blocos (Tiles)

- Uma estratégia muito comum dos SIG é particionar uma imagem em blocos durante o armazenamento.
- O PostGIS raster suporta esta estratégia de armazenamento.

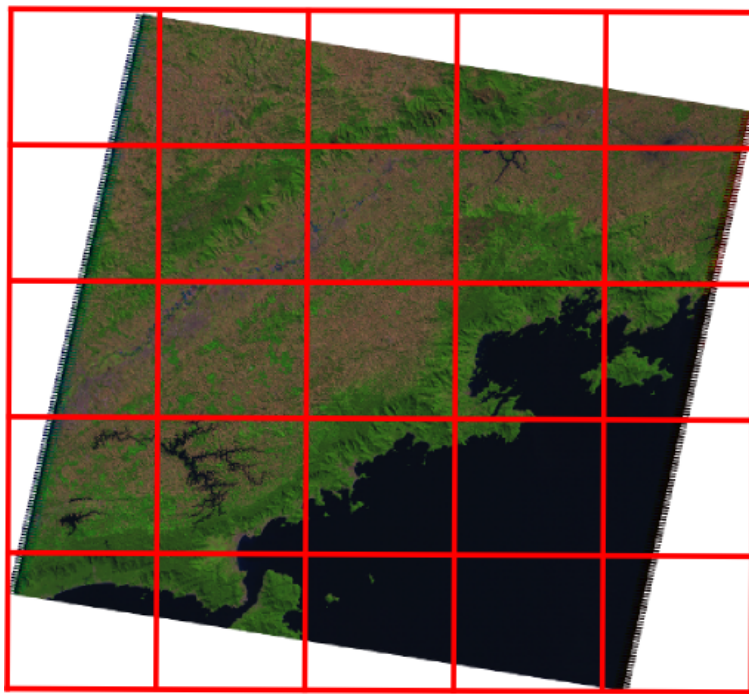




Tabela: imagem_sp	
gid	raster_tile
1	
2	
...	

Carregando um raster para o banco:

`raster2pgsql`

# raster2pgsql

- Ao executar o comando:  
`$ raster2pgsql -?`
- recebemos uma mensagem informando a versão da GDAL:  
`RELEASE: 2.1.6 GDAL_VERSION=111 (r13384)`
- A sintaxe básica deste comando é:  
`raster2pgsql [<options>] <raster>[ <raster>[ ...]]  
[[<schema>.]<table>]`
- Múltiplos rasters podem ser especificados usando: \*, ?.



# Importando a imagem bahia.tif

```
$ raster2pgsql -c -C -P -r -s 31984 -I -M  
-t 128x128 bahia.tif  
public.img_bahia > img_bahia.sql
```

- Onde:

- s 31984

- c: criar uma nova tabela e populá-la;

- C: aplicar algumas constraints;

- r: aplicar constraints de blocagem regular;

- t: tile size;

- I: criar índice espacial sobre a coluna raster;

- M: executar vacuum analyze na tabela raster.

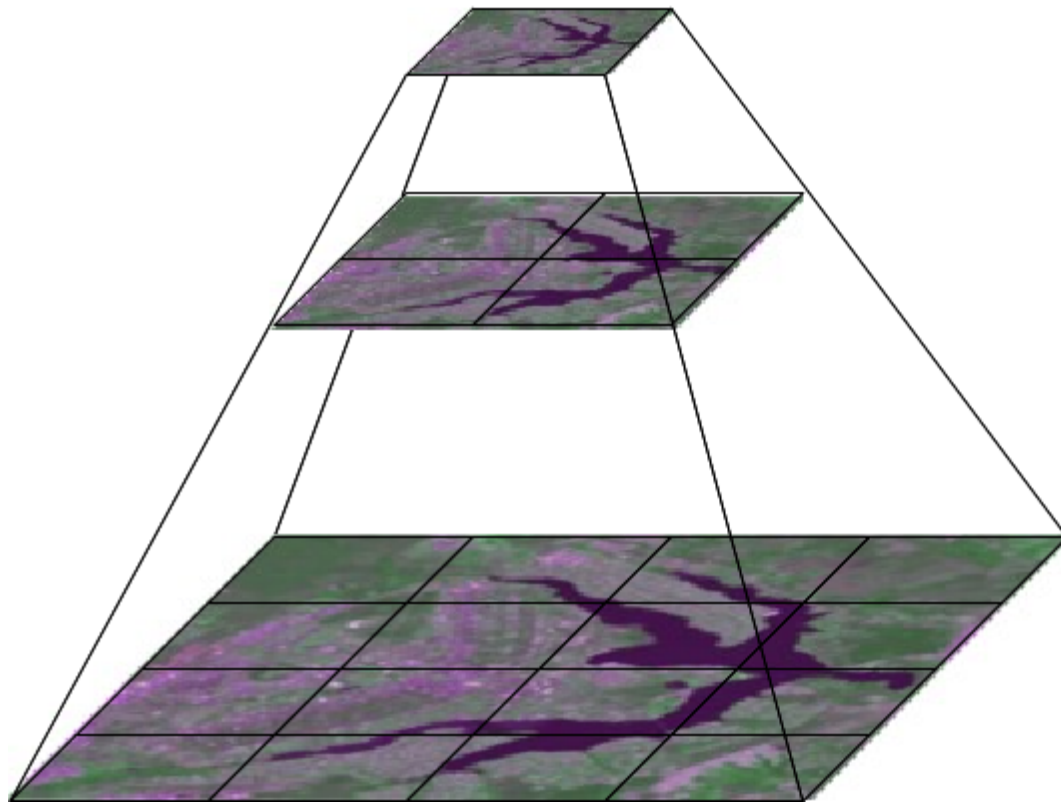
- Importar para o banco:

- \$ psql -U postgres -d bdgcurso -f img\_bahia.sql

# Visualizando tabelas raster com o Quantum GIS

# Overviews (Pirâmides/Multi-resolução)

- Parâmetro raster2pgsql: -1 2,4



$$R_2 = R \times 2^2$$

$$R_1 = R \times 2^1$$

$$R_0 = R \times 2^0$$

# Importando imagens com multiresolução

```
$ raster2pgsql -c -C -P -r -s 31984 -I -M  
-t 128x128 -l 2,4,8,16 bahia.tif  
public.img_bahia_mr > img_bahia_mr.sql
```

- Importar para o banco:

```
$ psql -U postgres -d bdgcurso -f img_bahia_mr.sql
```

# Processamento de Imagens no SGBD

```
-- Fazendo um clip da imagem CBERS
CREATE TABLE recorte_raster AS
  SELECT ST_Clip(rast,
    ST_Buffer(ST_Centroid(ST_Envelope(rast)), 100),
    false)
  FROM img_brasilia_cbars
  WHERE rid = 2100;
```

**Discussão:** Atualmente, não há um padrão bem definido para as extensões matriciais.

# O que mais existe nesta integração entre SGBD-R e Dados Geográficos?

Armazenamento baseado em modelos topológicos.

Redes espaciais: roteamento, análise de fluxo.

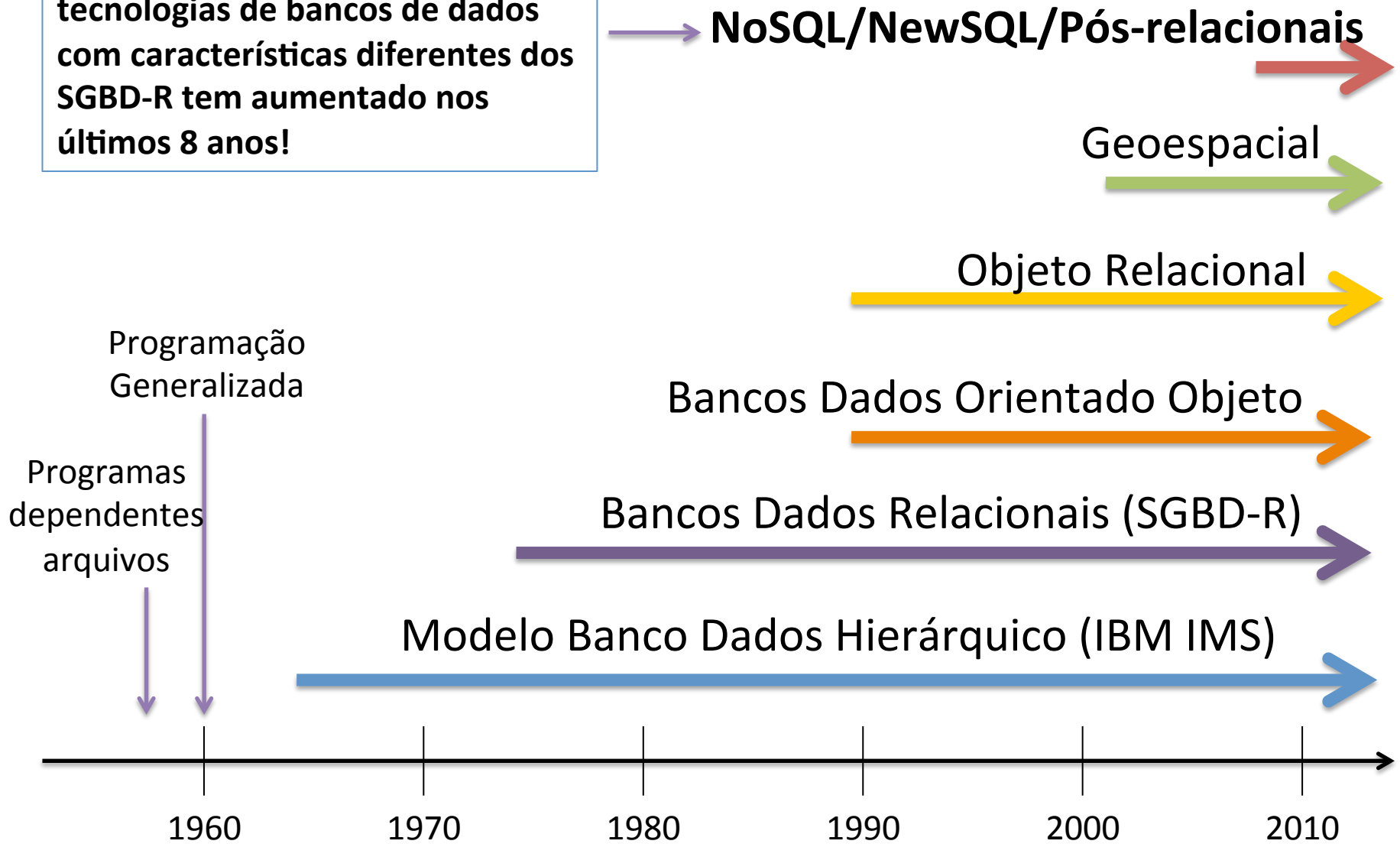
# Bancos de Dados x Information Retrieval

- Bancos de dados → Informações estruturadas
  - Esquemas
  - SQL
- IR → mais voltado para informações não estruturadas como processamento de documentos e texto livre.
  - Web Search Engines
  - SVM (Support Vector Machines)



# Evolução das Tecnologias de Bancos Dados

Interessante: o número de tecnologias de bancos de dados com características diferentes dos SGBD-R tem aumentado nos últimos 8 anos!



# O “cardápio” de opções aumentou?

- *Sistemas Não-Relacionais* ou *Not Only SQL* ou *Pós-relacionais*:
  - <http://nosql-database.org/>
  - <https://en.wikipedia.org/wiki/NoSQL>
- Diferentes modelos de dados:
  - Document Oriented: MongoDB, CouchDB;
  - Column Stores: Cassandra;
  - Graph Databases: OrientDB, Neo4J;
  - Array Databases: SciDB, Rasdaman.
- Nem todos são baseados no paradigma de transações ACID.
- Escalabilidade: Horizontal x Vertical

# Suporte Espacial em NoSQL

- MongoDB
- CouchDB
- Apache Solr
- Neo4J Spatial

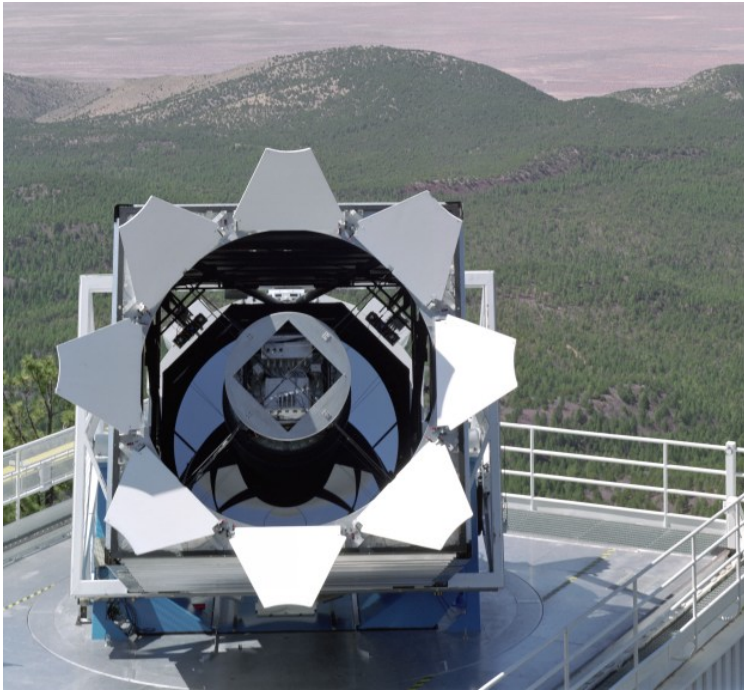
**Reflexão:** Existe algum problema com o projeto do suporte Raster dos atuais SGBD-R?

Como lidar com os requisitos de aplicações de EO que podem necessitar como entrada dados massivos?

Os dados utilizados em diversas áreas da  
Ciência encontram-se na forma de  
**Arrays**

Arrays = Matrizes

# Imagens de Telescópios Astronômicos



[The SDSS survey telescope](#)  
(Sacramento Mountains)

Fonte: [The Sloan Digital Sky Survey](#)

## [Sloan Digital Sky Survey / SkyServer](#)

- Em operação desde 2000
- SDSS I e II (2000-2008):
  - 10 TiB raw data, 120 TiB processados
- [SDSS III](#)

Fonte: Takar (2013)

## [Messier 51, The Whirlpool Galaxy](#)



# Biologia: Genoma, Máquinas de Sequenciamento

```
GATCAGAGAAATTCCAGCATATGACATCCAGG
CGCTAGCCGGTATATGAAATGAGAGGATCATC
ACACTATGTGATGACATACTAGACC GG TGATG
GGGATATCAGGAATTCCAGCATATGACATCCA
CGCGCTAGCCGGTATATGAAGGATGAGAGGGA
GCCACCACTATGTGATGACATACTAGACC GG T
ACGATGGATTACAGGAATTCCAGCATATGACA
GAGGCCACGCGCTAGCCGCTATATGAAATGAG
AGAGGGACACCACTATGTGATGACATACTAGA
CCC GG TGATGGATTACAGGAATCCAGCATA
TGACATACACGCGCTAGCCGAGTATATGAGAG
ACATGAGAGGGACACCACTATGTGATGACATA
CCCTAGACC GG TGATGGATTACAGGAATTCCC
GCATATGACACCCACGCGCTAGCACGTATAAG
CATTGAAATGAGAGAGGAATCCACTATGTGAT
GACATACTAGACC G T T T GTGATGGATTACAGG
AATCCAGCATATGACATCCACATCCCTAGGTC
CAGGTATATGAAATGAGAGGGACACCACTATG
```

**INDEX**  
**AAA** OFFSETS: 9, 49, 257, 467, 571  
**ATC** OFFSETS: 2, 60, 104, 127, 319, 480, 551  
**GGG** OFFSETS: 40, 124, 141, 194, 300, 404

Fonte: [Schatz and Langmead \(2013\)](#)

DNA -> blueprint for all living organisms

Genomics: Source of big data

Large dense matrix of floating point numbers

Analytics: Regression, Covariance, Biclustering, SVD, Statistics

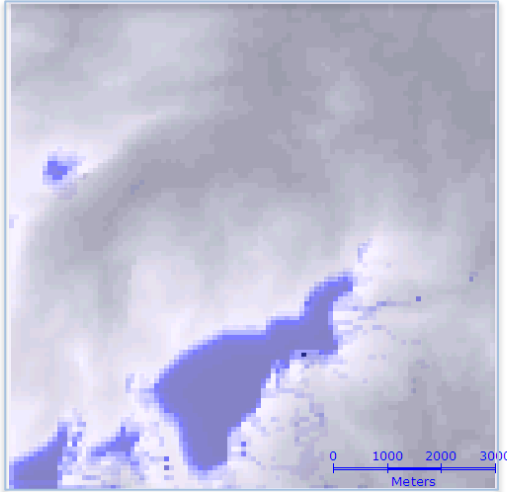
Fonte: Taft et al. (2013)

Some projects:

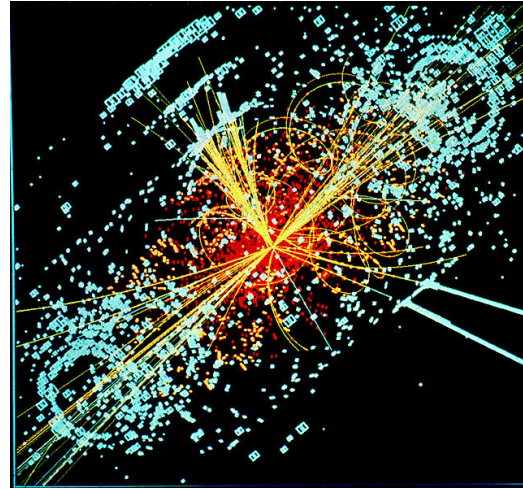
- ✓ [The Human Genome Project \(HGP\)](#)
- ✓ [1000 Genomes](#)
- ✓ [Stanford Microarray database](#)

# Simulação Computacional

Fonte: (Carneiro, 2006)

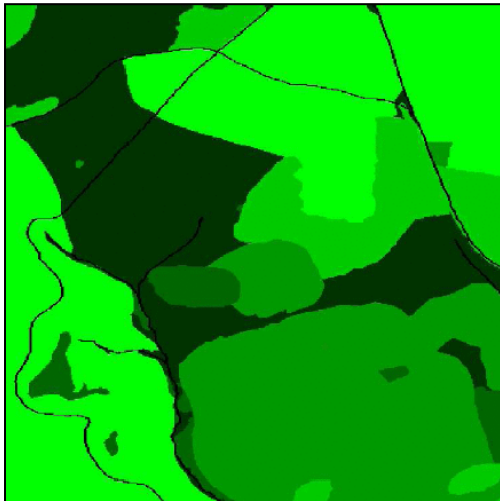


Fonte: [Wikipedia](#)

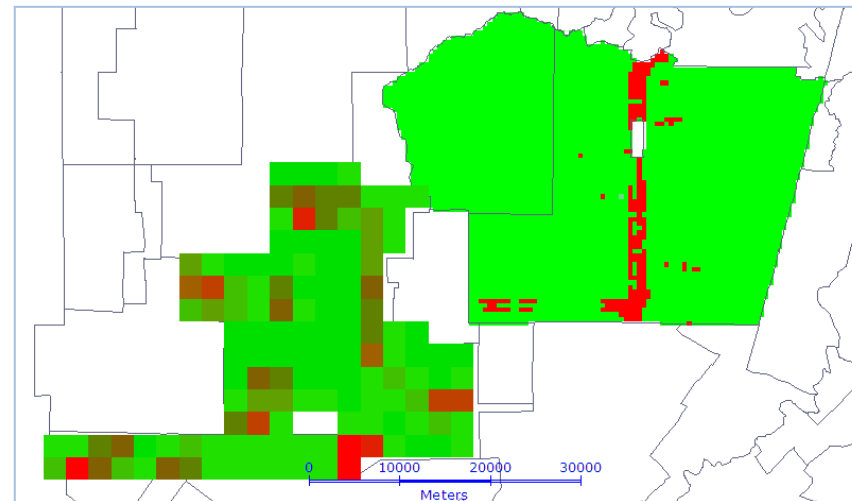


Simulated Large Hadron Collider CMS particle detector data depicting a Higgs boson produced by colliding protons decaying into hadron jets and electrons

Fonte: (Almeida et al, 2008)

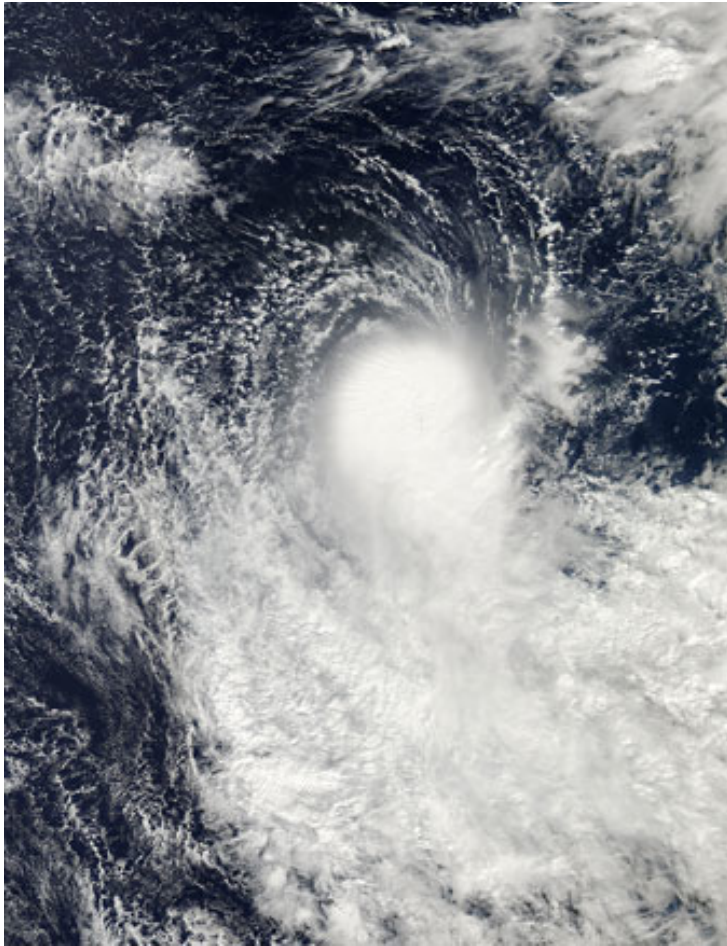


Fonte: (Carneiro, 2006)

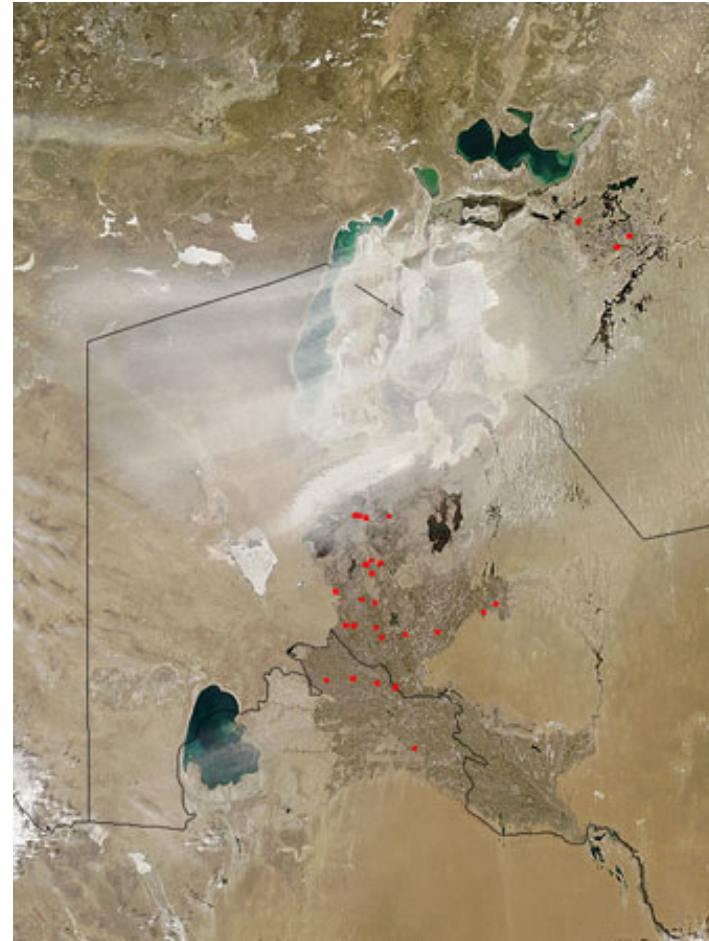




# Dados Geoespaciais: imagens de satélite



Tropical Cyclone Jack (24S) in the South Indian Ocean  
Fonte: <http://modis.gsfc.nasa.gov> (Abril, 2014)



Spreading dust storm over the Aral Sea  
Fonte: <http://modis.gsfc.nasa.gov> (Abril, 2014)

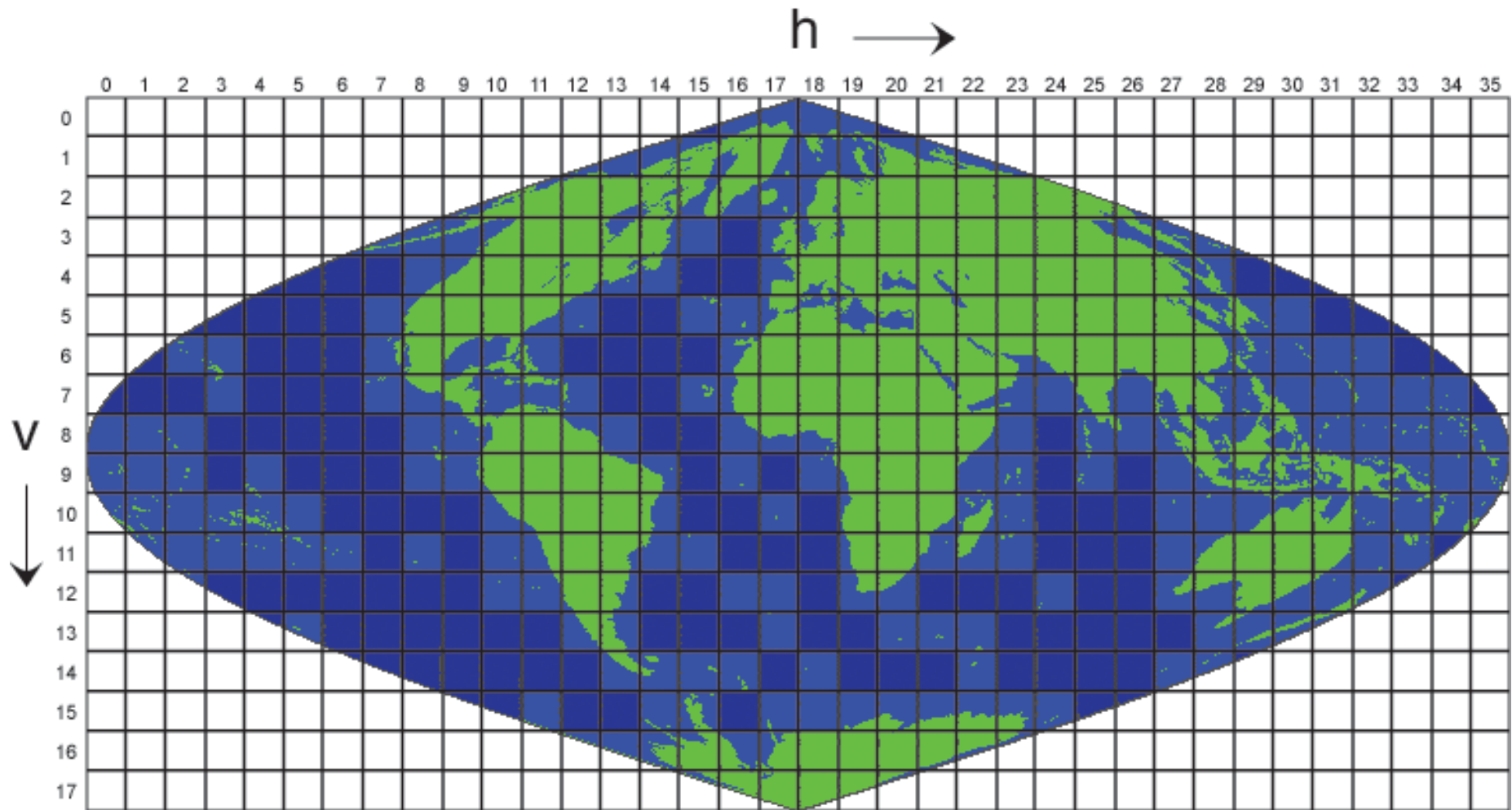
# Arrays: A.K.A.

- Raster data
- Regularly gridded data
- Multi-Dimensional Discrete Data (MDD)
- Imagens
- Dados Matriciais

# Estudo de Caso em EO

Dados do Sensor MODIS

# MODIS: Grade Sinusoidal



Fonte: [http://nsidc.org/data/modis/data\\_summaries/landgrid.html](http://nsidc.org/data/modis/data_summaries/landgrid.html)

# Arquivos Originais → HDF

- Cada arquivo contém 12 *subdatasets* (ou bandas):
  - 11 bandas de 16-bits;
  - 1 banda de 8-bits (*pixel reliability*).
- Arquivos de tamanho variável: ~5 MiB → ~230 MiB
- Cada tile (cena) possui 4.800 x 4.800 pixels.
- Internamente os HDFs são organizados em blocos de 4.800 x 208 pixels.
- CRS:  
`+proj=sinu +lon_0=0 +x_0=0 +y_0=0 +a=6371007.181 +b=6371007.181  
+units=m +no_defs`
- Resolução Temporal: 16-dias

# Brasil → 30 Tiles

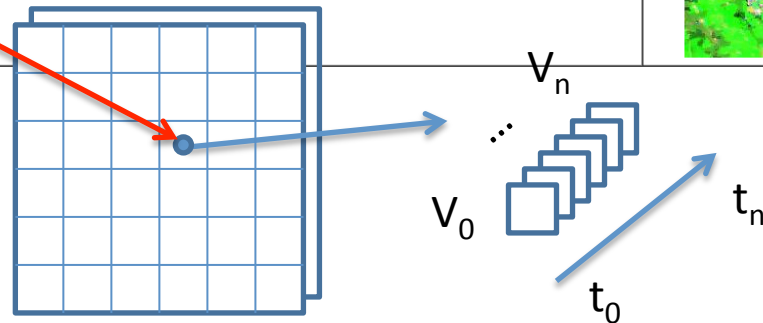
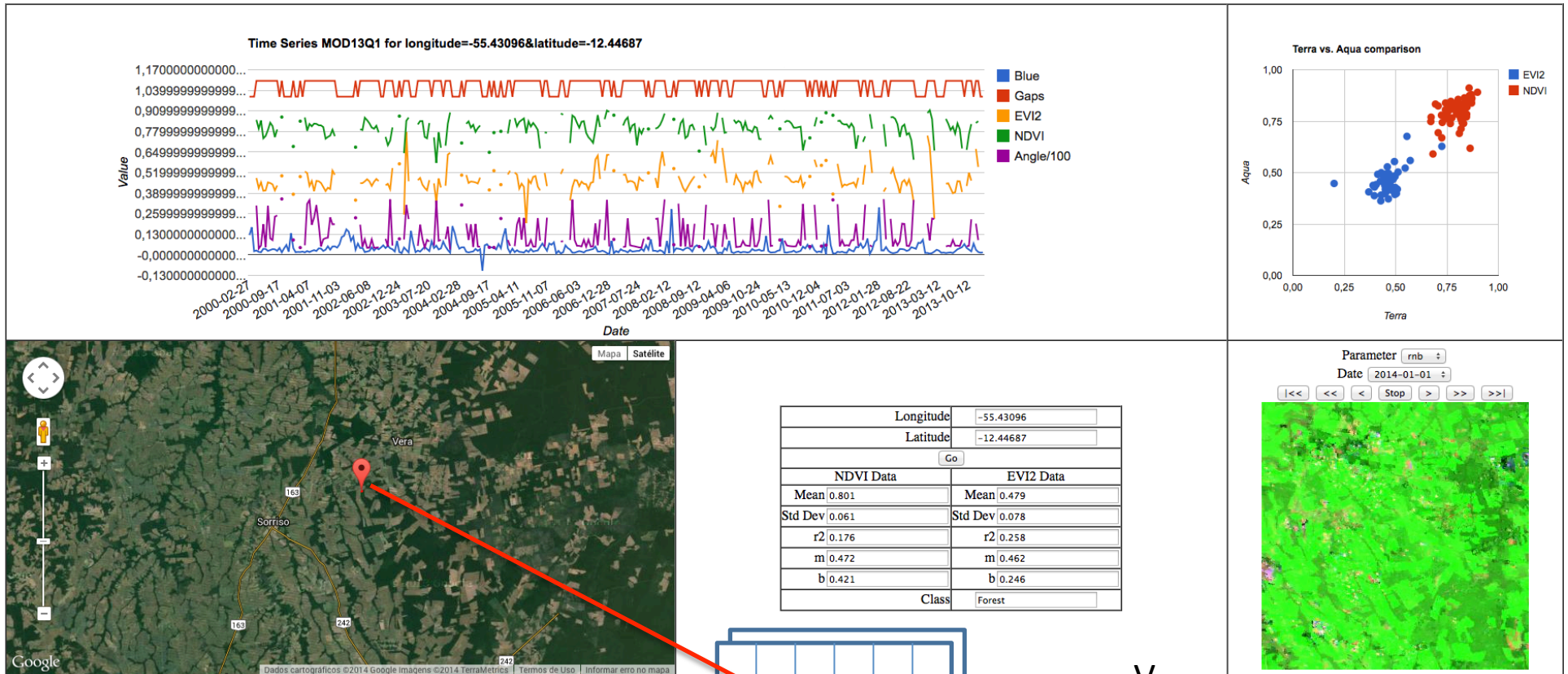
- Considerando 15 anos de observação.
- Para cada tile:  $23 * 15 \rightarrow 345$  imagens.
- 30 tiles:  $30 * 345 \rightarrow 10.350$  imagens.
- 1 imagem  $\rightarrow 4800 \times 4800 \rightarrow 23.040.000$  pixels.
- $T_0 \rightarrow 30\text{-tiles} \rightarrow 691.200.000$  pixels.
- 10.350 imagens:  $238.464.000.000$  pixels.
- Considerando apenas o atributo (banda) red-reflectance:  
 $238.464.000.000 \times 2 \text{ bytes} = 444.17 \text{ GiB}$

# Exemplo: Mapa de Vegetação Global de Fevereiro de 2000 a Fevereiro de 2014



Fonte: [NASA Earth Observatory](#) (23 de Abril , 2014)

# Exemplo: Time Series in Land Use and Land Cover Change (LUCC) Studies





# Novas Tecnologias de Bancos de Dados para Dados Matriciais

## Array Databases

“Arrays as first class citizens”

## Solução 2: Usando um Array Database

rasdaman

SciDB

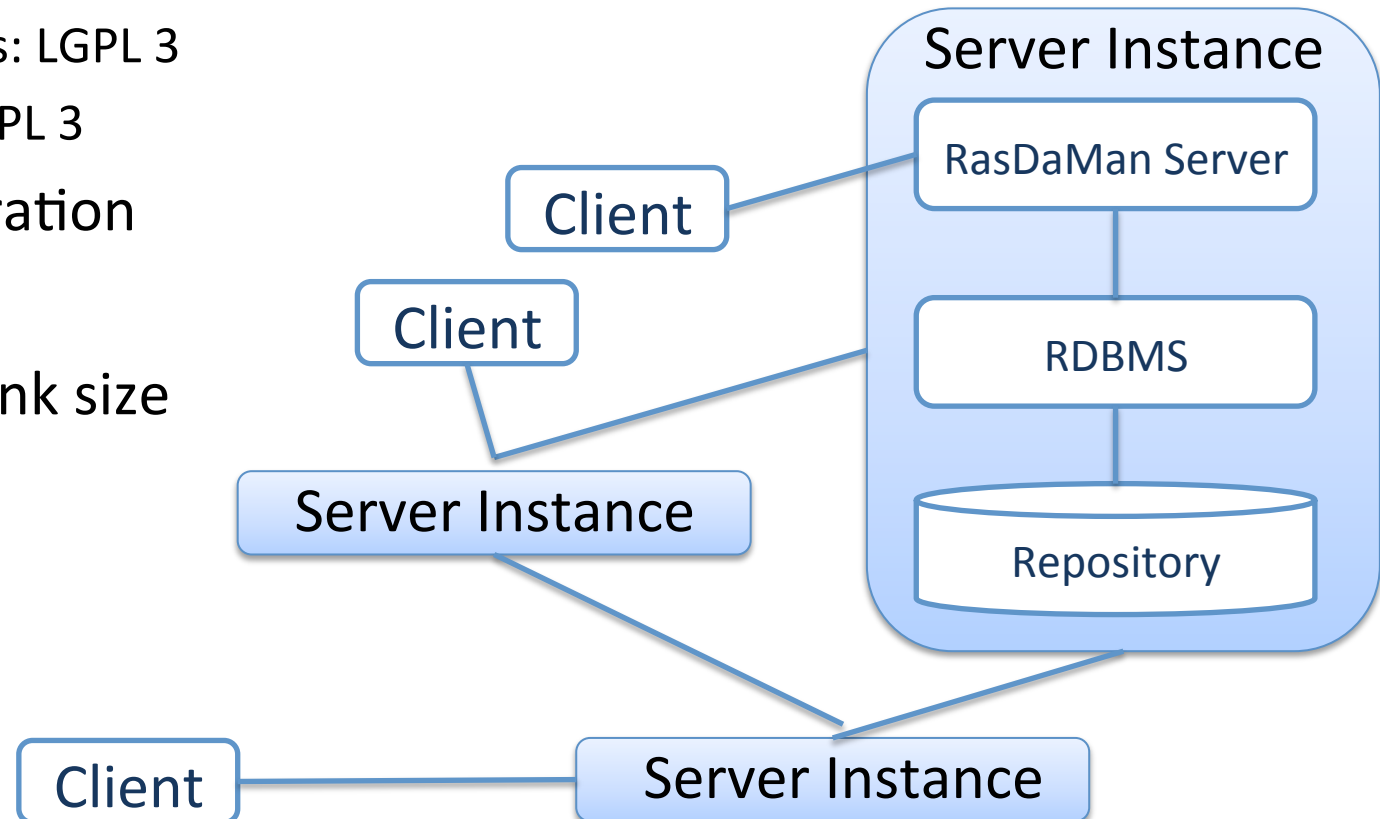
# **Raster Data Manager** (rasdaman)

Prof. Peter Baumann

Jacobs University, Computer Science, Bremen,  
Germany

# rasdaman

- Domain-neutral Array Database System.
- Site: <http://www.rasdaman.org>
- License:
  - Client libraries: LGPL 3
  - Server side: GPL 3
- Server Federation
- OGC WCPS
- Variable chunk size



# rasdaman → rasql

- Array Algebra → Embedded Query Language (SQL) → rasql
- Example:
  - \* “subtracting each MDD of collection mr2 from each MDD of collection mr where at least one difference pixel value is greater than 50”:

```
select mr - mr2
  from mr, mr2
 where some_cells( mr - mr2 > 50 )
```

## Solução 2: Usando um Array Database

rasdaman

SciDB 

# SciDB

“SciDB is an open-source analytical database oriented toward the data management needs of scientists.”

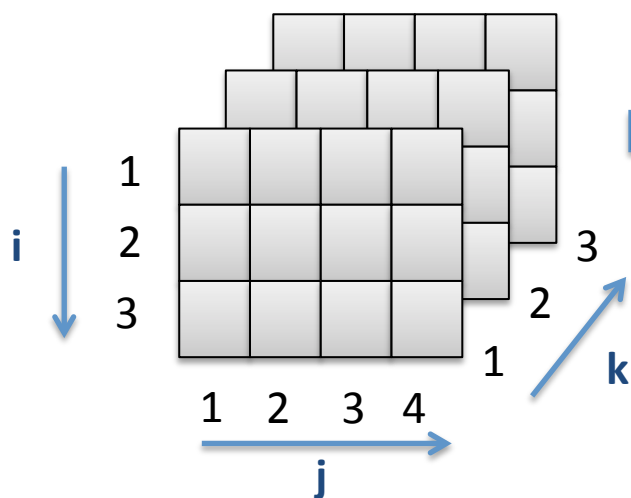
(Stonebraker et al., 2013)

# Arrays, Dimensões, Atributos e Particionamento

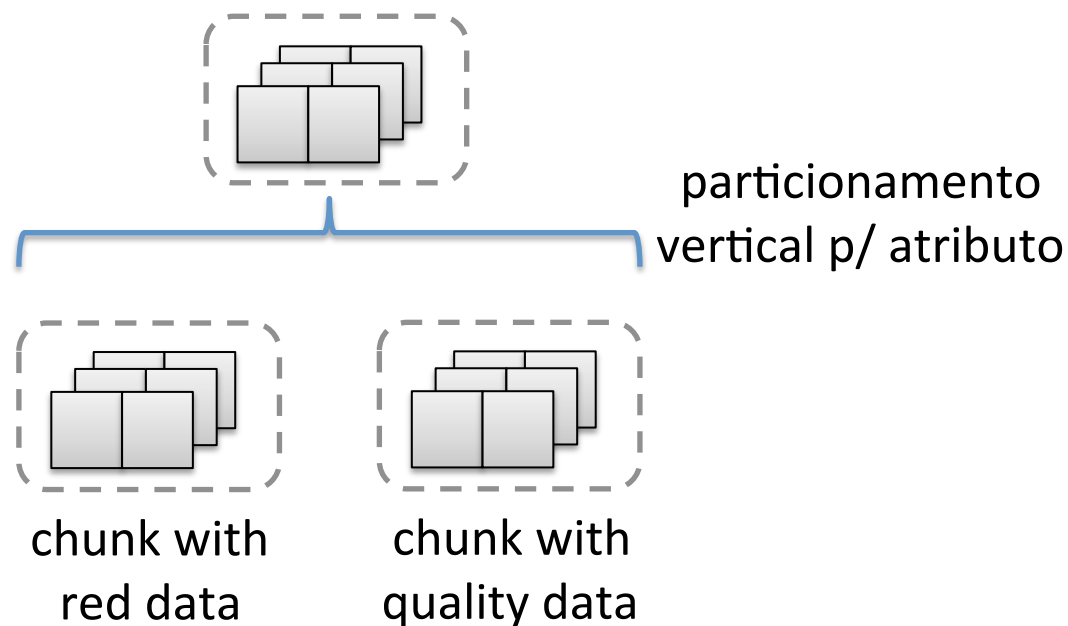
mod09q1

<attributes>

[j=1:4, 2, 1, i=1:3, 1, 1, t=1:\*, 3, 0]



Array é dividido em chunks  
chunk-size: 2 x 1 x 3



- ✓ Replicação
- ✓ Compressão
- ✓ Versionamento



# Linguagens de Consulta: AQL e AFL

```
AQL% SELECT sqrt(v1) INTO a3
      FROM a1
      WHERE j = 1;
```

```
AQL% SELECT sum(v1) FROM a5 WHERE j = 1;
```

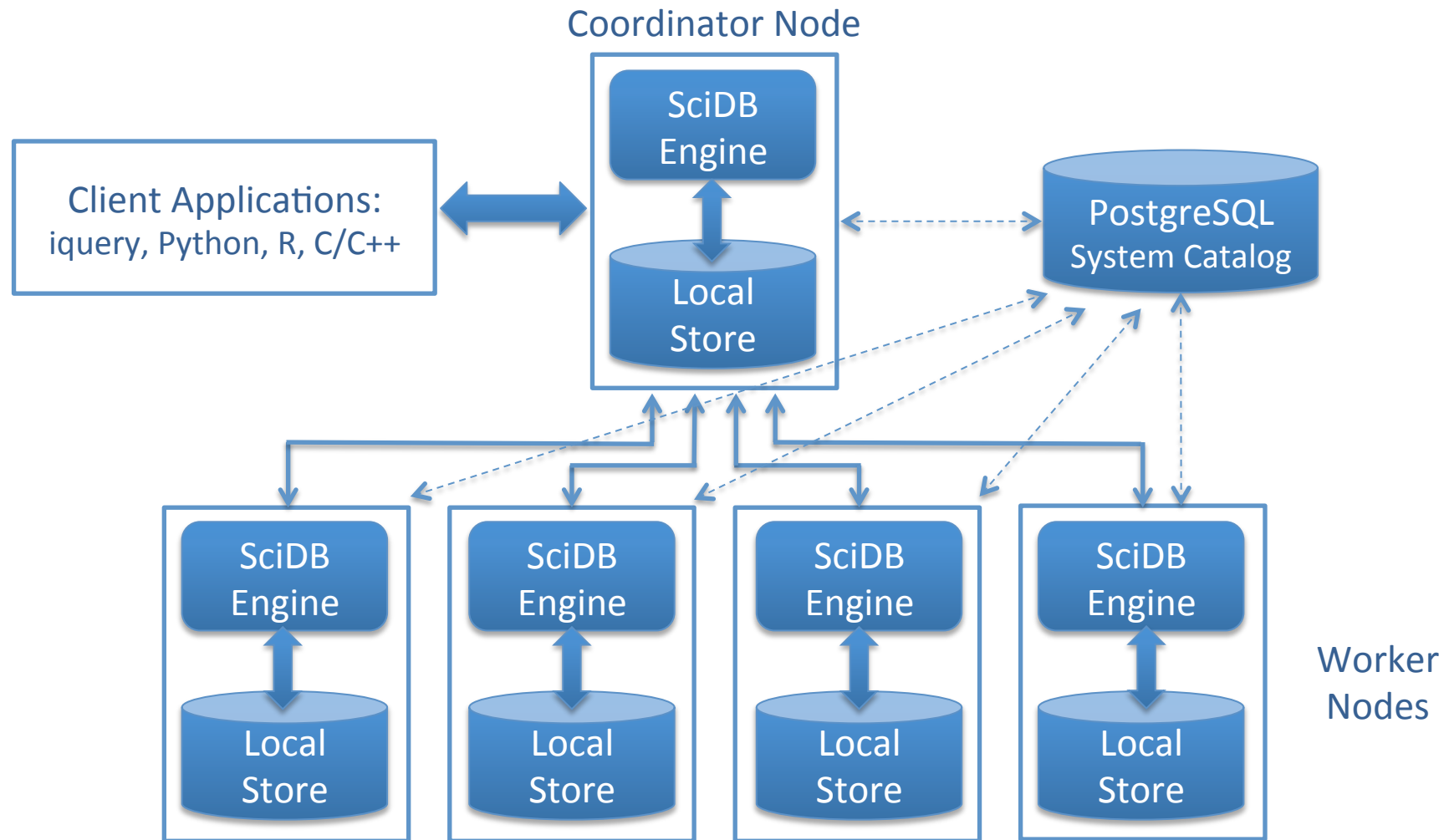
```
AFL% build(<v2:double>[j=1:4,2,1, y=1:4,2,1],
          j - i);
```

1	0	1	2	3
2	-1	0	1	2
3	-2	-1	0	1
4	-3	-2	-1	0
	1	2	3	4

# Integração com ambientes estatísticos e de álgebra linear

- R
- [ScaLAPACK](#) — Scalable Linear Algebra PACKage

# Arquitetura



Source: Adapted from PARADIGM4

# Behind SciDB

- Founders: Mike Stonebraker, Andy Palmer, David DeWitt, Kian-Tat Lim, Jacek Becla  
Science Advisory Board:
  - “... we have spent a large amount of time talking to scientists about their requirements from a data management system (see the “Use Cases” section of the scidb.org website) ...” Seering et al. (2012).
- Large Synoptic Survey Telescope (LSST) project:
  - [Extreme Large Databases](#) (XLDB)
- Intel Science and Technology Center for Big Data:  
<http://istc-bigdata.org>
- The develop team:  
Company: Paradigm4

# O que não se tem no SciDB?

- Loader de matrizes multidimensionais.
- As matrizes não tem um sistema de referencia espacial associado.
- As células não possuem informações de resolução.
- Não existem operadores espaciais como clipping c/ uma máscara vetorial.
- Não se tem um suporte a definição de metadados dos arrays. (Embora o catálogo esteja no PostgreSQL!)

# Referências

# Livros

- ELMASRI, R.; NAVATHE, S. B. *Fundamentals of database systems*. Addison Wesley, 2006. 1139p.
- DATE, C. J. *An introduction to database systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1991.

# Artigos

- E. F. Codd. 1970. ***A relational model of data for large shared data banks***. *Communications of the ACM*, v. 13, n. 6, June 1970, pp. 377-387.
- Chen, P. ***The Entity-Relationship Model-Toward a Unified View of Data***. *ACM Transactions on Database Systems*, vl. 1, n. 1. March 1976, pp. 9-36.
- GRAY, J. ***Evolution of Data Management***. *IEEE Computer* 29(10): 38-46, 1996.
- Vijlbrief, T., and P. van Oosterom. ***The GEO++ System: An Extensible GIS***. *Proc. 5th Intl. Symposium on Spatial Data Handling*, Charleston, South Carolina, 1992, 40-50.



# Artigos

- STONEBRAKER, M.; BROWN, P.; POLIAKOV, A.; RAMAN, S. ***The architecture of SciDB***. In Proceedings of the 23rd international conference on Scientific and statistical database management (SSDBM'11), Judith Bayard Cushing, James French, and Shawn Bowers (Eds.). Springer-Verlag, Berlin, Heidelberg, 2011, 1-16.
- TAFT, R.; VARTAK, M.; SATISH, N. R.; SUNDARAM, N.; MADDEN, S.; STONEBRAKER, M. ***Genbase: a complex analytics genomics benchmark***. Computer Science and Artificial Intelligence Laboratory Technical Report, MIT-CSAIL-TR-2013-028, November 19, 2013.

# Especificações e Padrões

- OGC. ***OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture***. Available at: <http://www.opengeospatial.org>. Access: October, 2012.
- OGC. ***OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option***. Available at: <http://www.opengeospatial.org>. Access: October, 2012.
- ISO. ***SQL Multimedia and Application Packages – Part 3: Spatial***.

# Slides

- NAUGHTON, J. F. ***DBMS Research: First 50 Years, Next 50 Years***. Kynote speaker' slides at ICDE 2010. Disponível em: <http://pages.cs.wisc.edu/~naughton/naughtonicde.pptx>. Acesso: Abril de 2013.

**Convite:**

Curso de Bancos de Dados Geográficos 2015