

# Bancos de Dados Geográficos

Introdução a Sistemas de Bancos de Dados

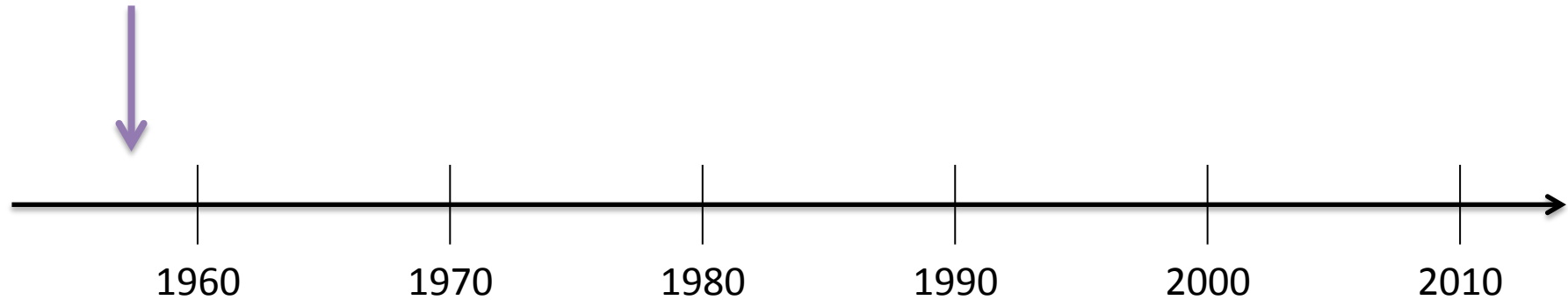
Gilberto Ribeiro de Queiroz <[gribeiro@dpi.inpe.br](mailto:gribeiro@dpi.inpe.br)>

# SGBD: uma tecnologia amplamente difundida

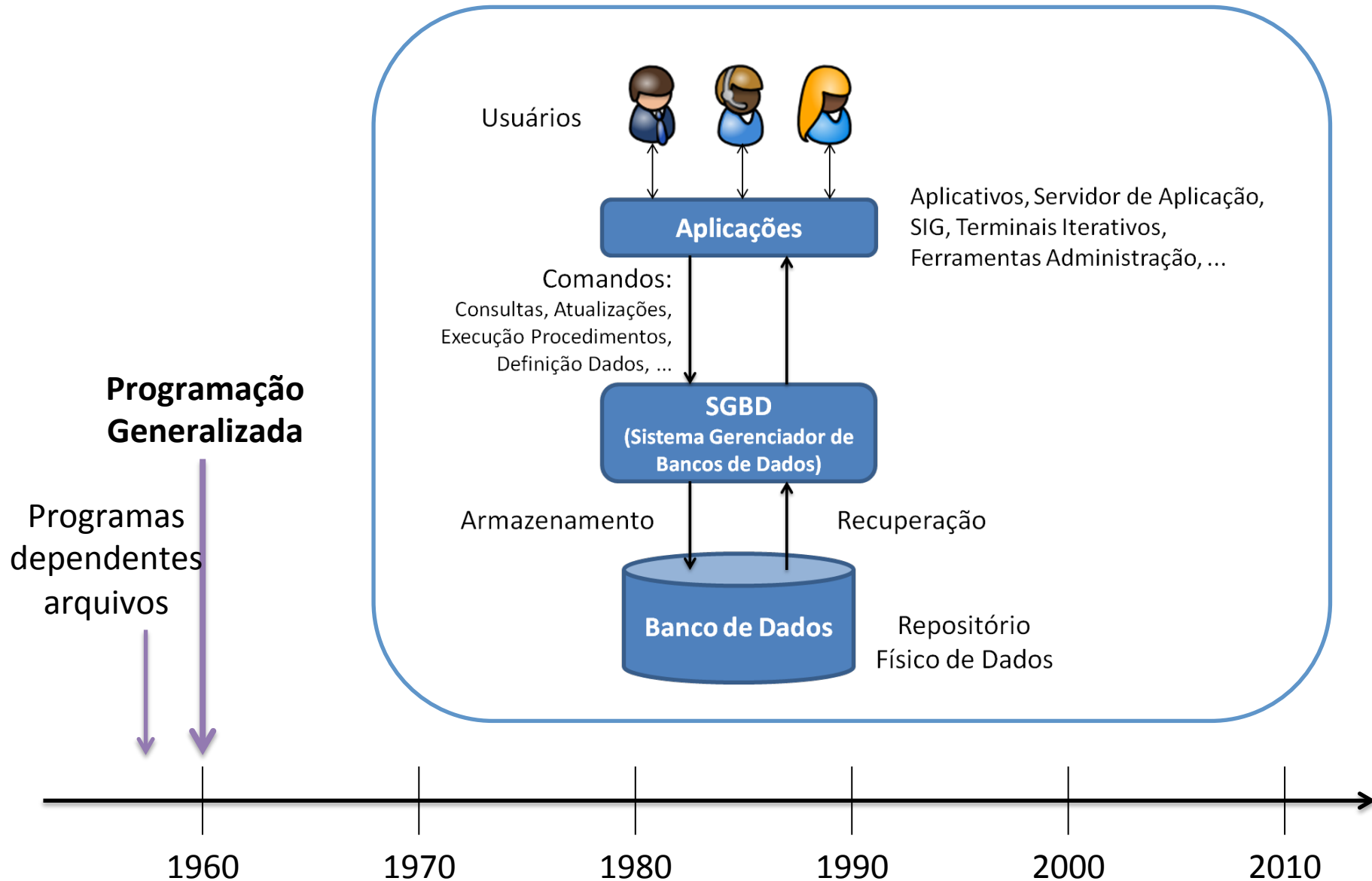
- A tecnologia de bancos de dados tem sido um componente fundamental em quase todos os tipos de aplicações:
  - Conta bancária: depósitos e saques
  - Reservas de passagens aéreas
  - Reservas em hotéis
  - Compras de livros, CDs, DVDs e outros bens (Amazon)
  - Busca por artigos em uma revista eletrônica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

# Evolução das Tecnologias de Bancos Dados

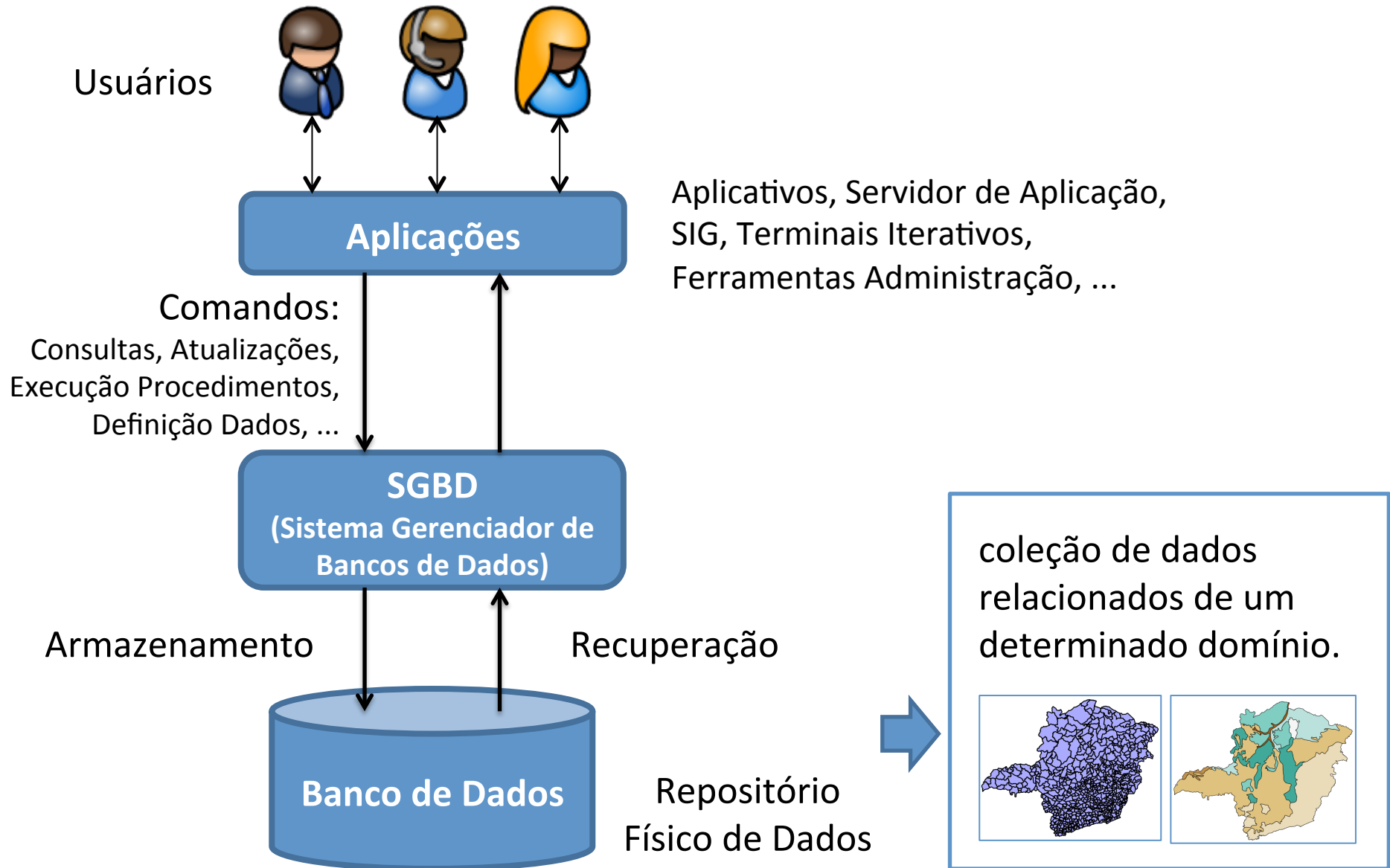
**Programas  
dependentes  
arquivos**



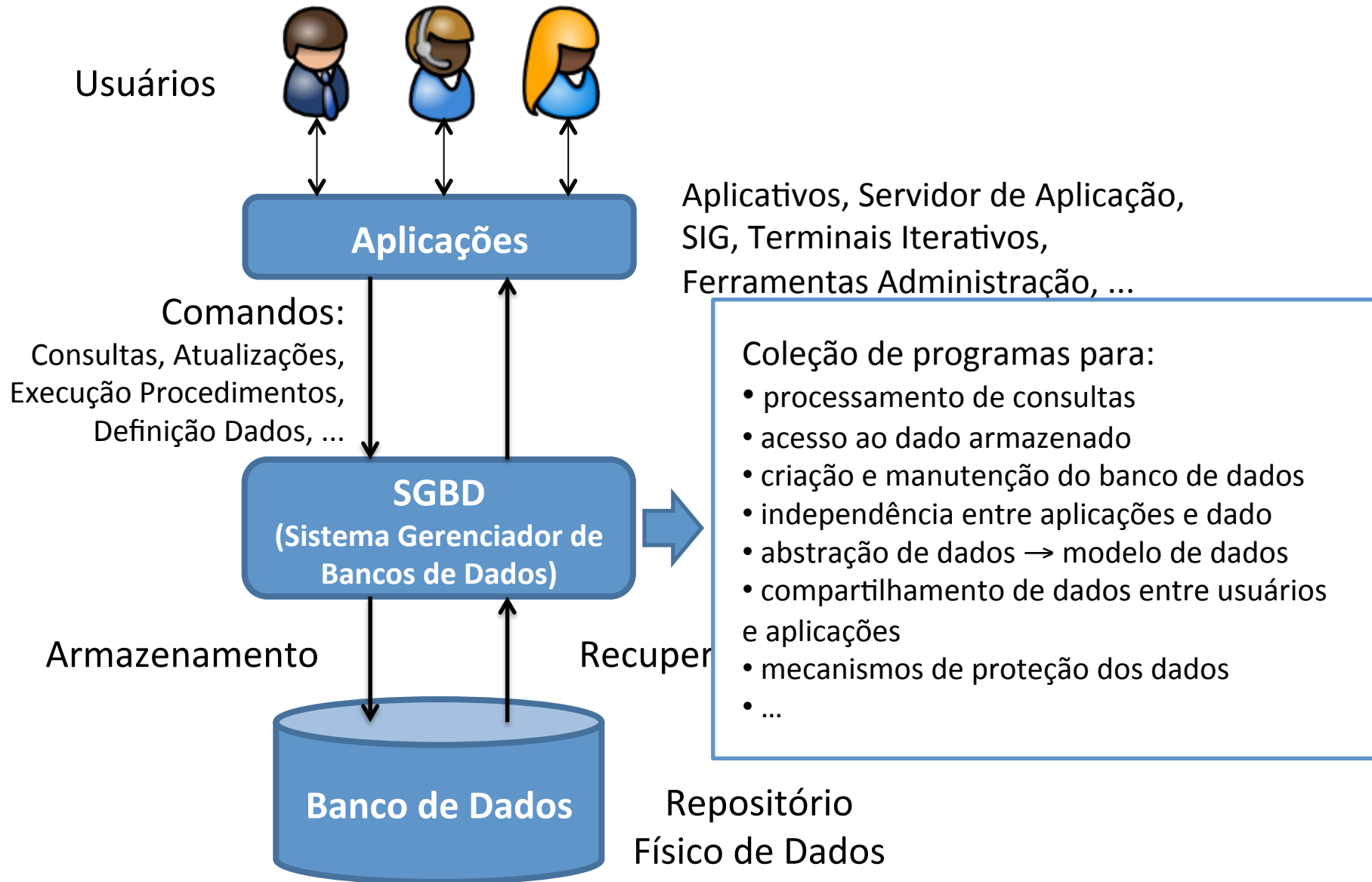
# Evolução das Tecnologias de Bancos Dados



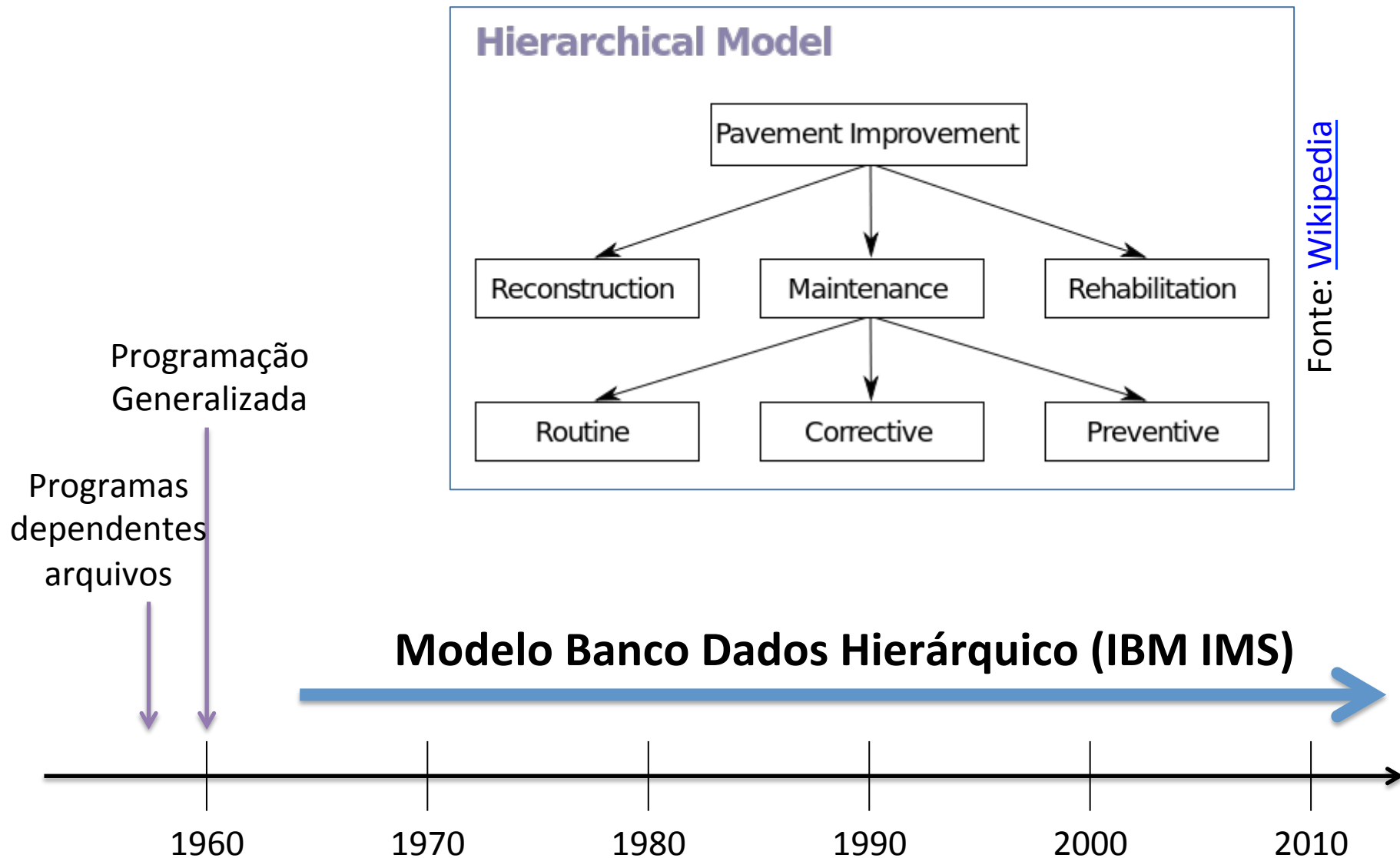
# Sistemas de Bancos de Dados



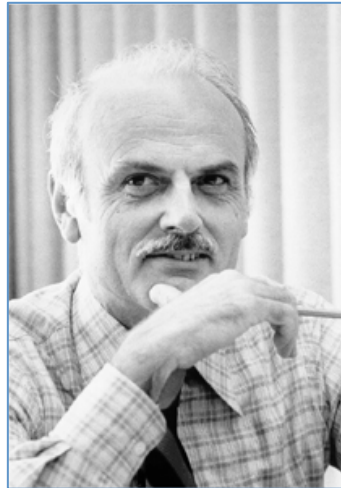
# Sistemas de Bancos de Dados



# Evolução das Tecnologias de Bancos Dados

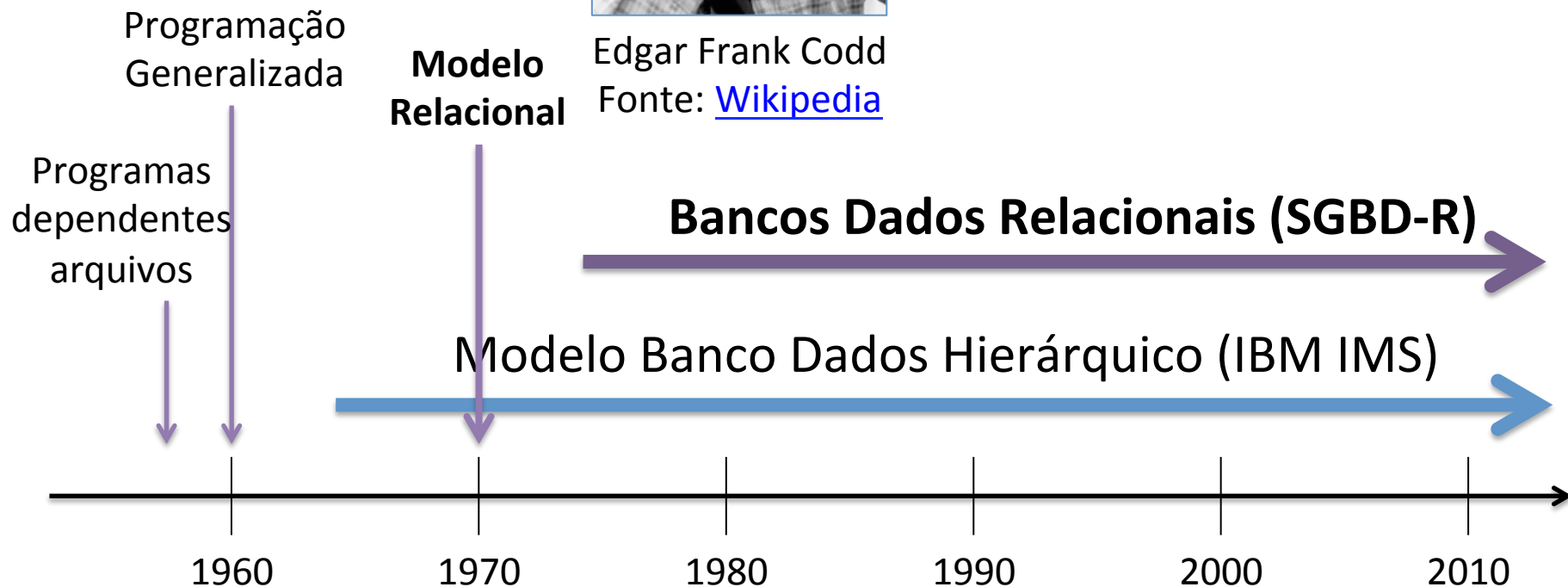


# Evolução das Tecnologias de Bancos Dados



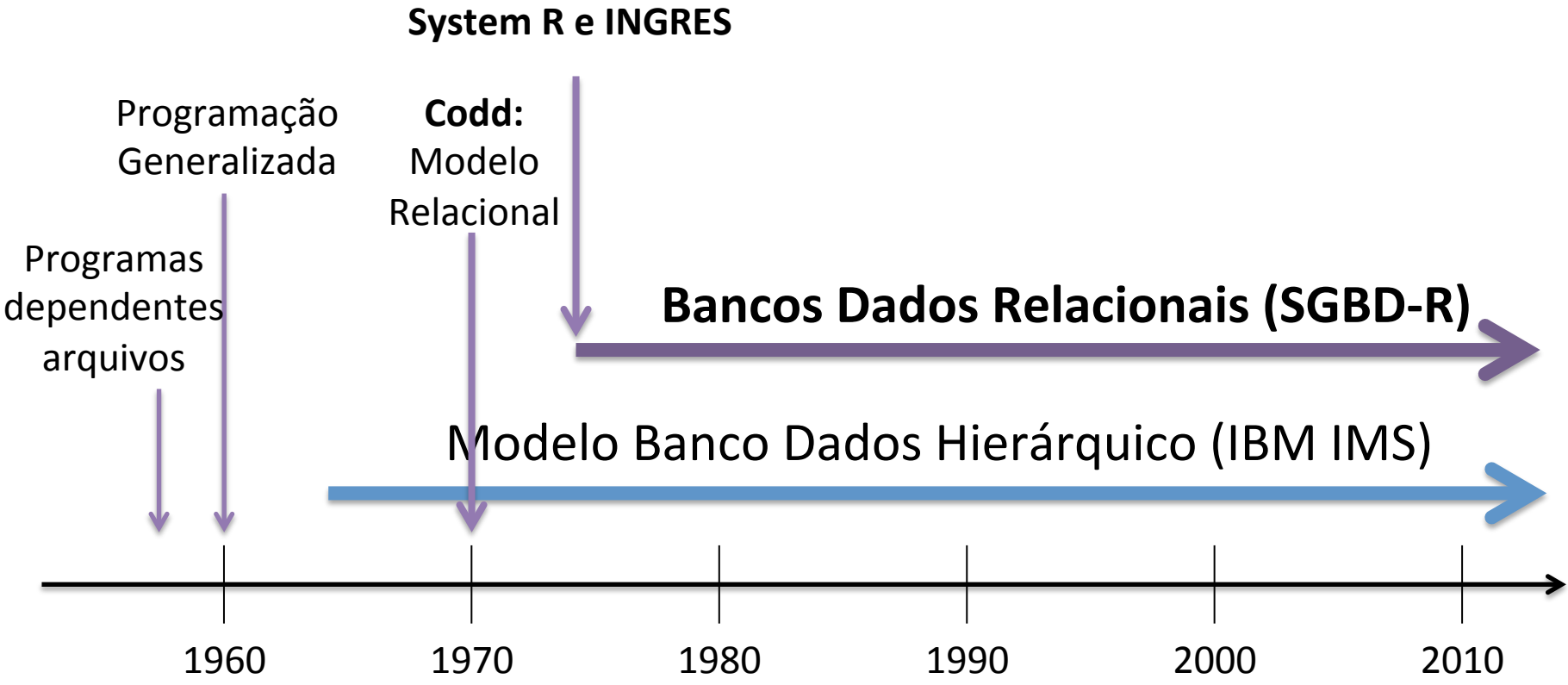
E. F. Codd. 1970. *A relational model of data for large shared data banks*. Communications of the ACM, v. 13, n. 6, June 1970, pp. 377-387.

Edgar Frank Codd  
Fonte: [Wikipedia](#)

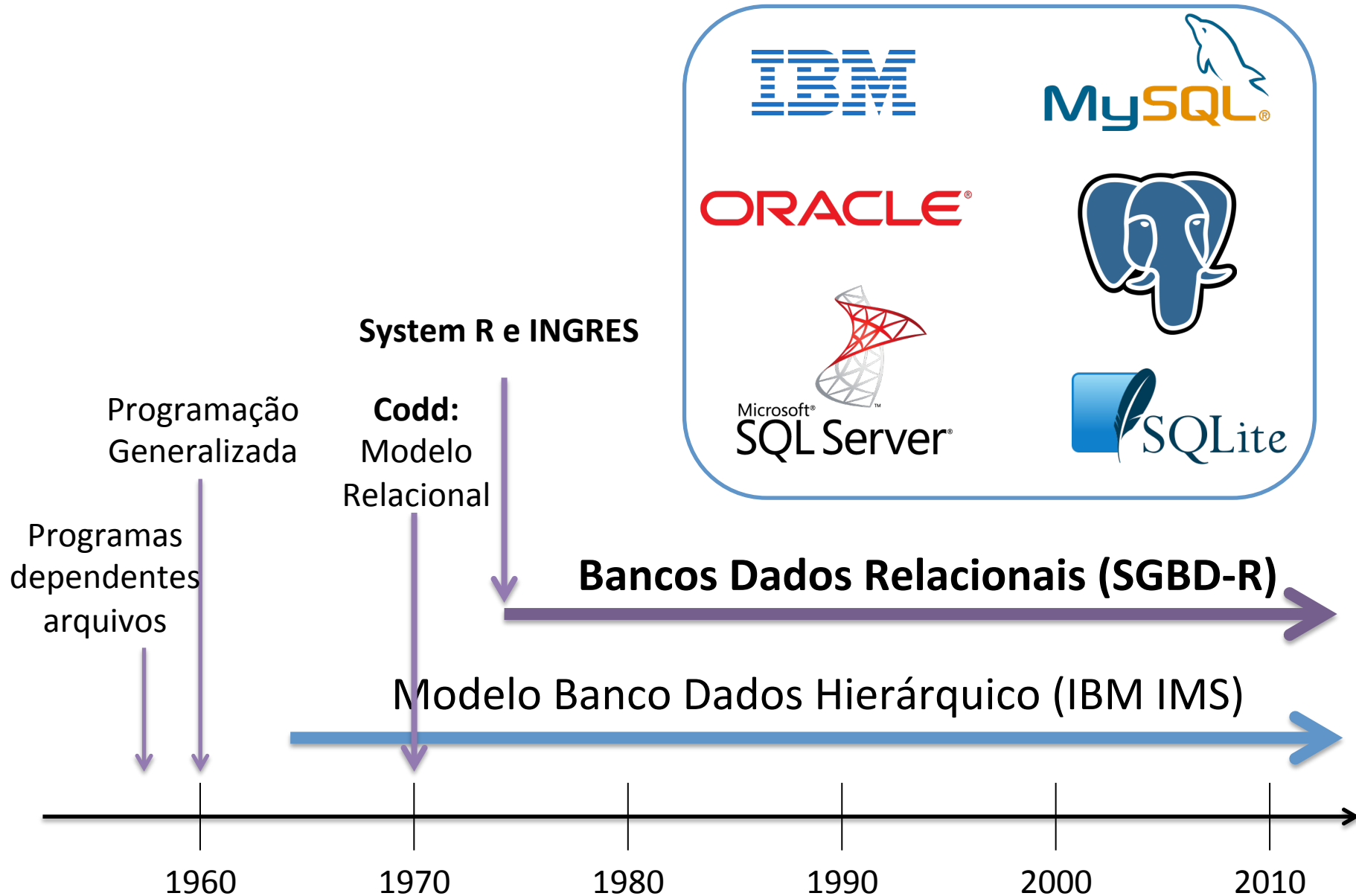




# Evolução das Tecnologias de Bancos Dados



# 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	
...	...	...	...

Diagram illustrating a table structure with labels and annotations:

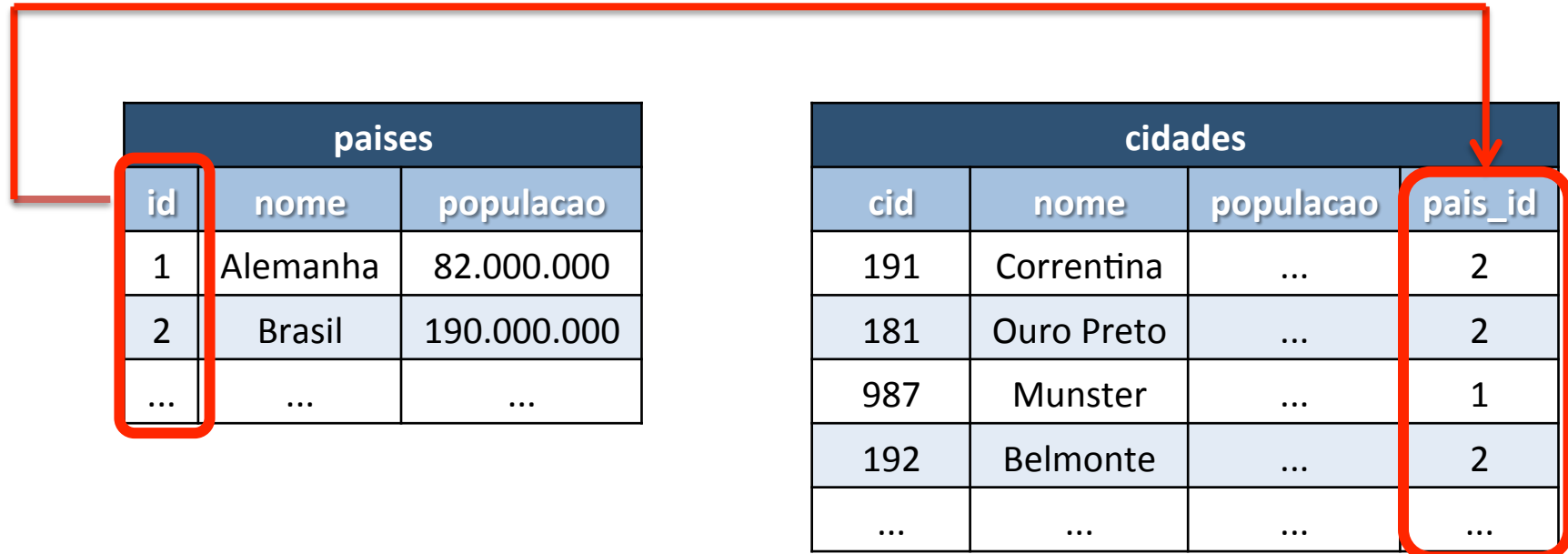
- Tabela** (Table) points to the entire table structure.
- Colunas** (Columns) points to the header row.
- Linha** (Row) points to the first data row.
- Esquema Tabela** (Table Schema) points to the header row.
- Instância** (Instance) points to the data rows.

# 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_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** →

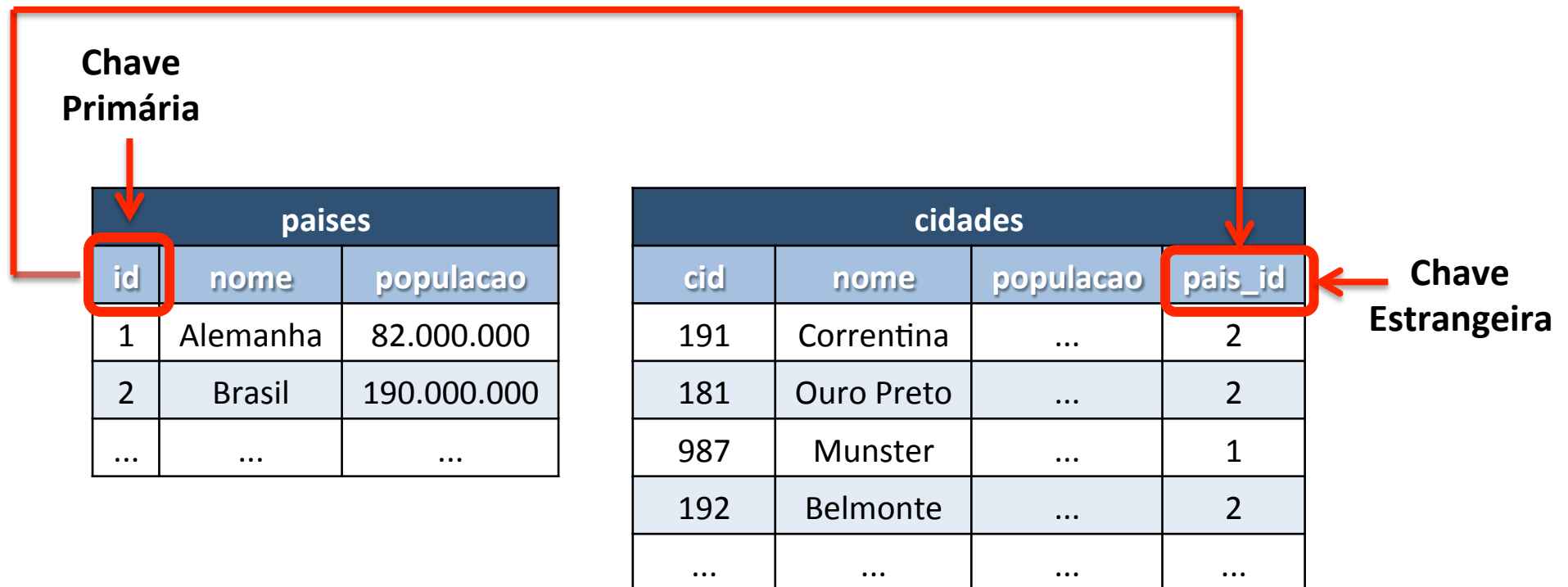
cidades			
cid	nome	populacao	pais_id
191	Correntina	...	2
181	Ouro Preto	...	2
987	Munster	...	1
192	Belmonte	...	2
...	...	...	...

**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)



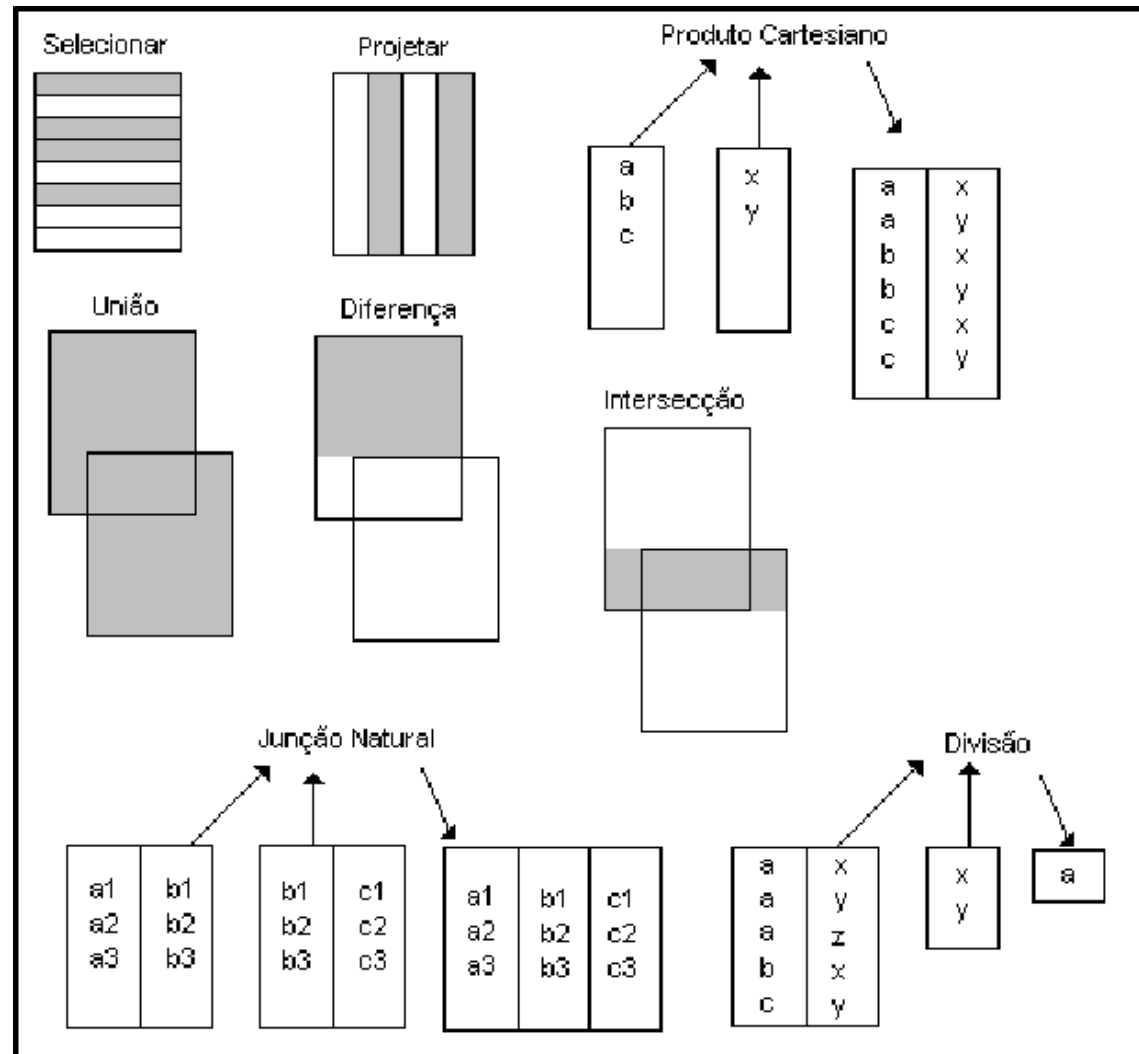
# Restrições de Integridade (Constraints)

- Permitem estabelecer critérios para manutenção da consistência dos dados no banco de dados:
  - **Restrições de domínio:**
    - Especifica os possíveis valores de uma coluna (tipo de dado de uma coluna).
  - **Chave primária.**
  - **Chave estrangeira (ou restrição de integridade referencial):**
    - Uma chave estrangeira especifica uma restrição de integridade referencial entre duas relações  $R_1$  e  $R_2$  sobre um conjunto de colunas.
  - **Valor único:**
    - Garante que o valor em um campo ou conjunto de campos sejam únicos dentro da tabela (semelhante ao conceito de chave primária).
  - **Restrições de nulidade:**
    - especifica se o valor de uma coluna pode ou não ser nulo.
  - **Restrições de valores:**
    - Possibilitam avaliar se o valor de uma ou mais colunas satisfaz uma determinada expressão (fórmula).
- Veremos ao longo do curso de forma prática o que significa cada uma dessas restrições e suas implicações.

# 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)

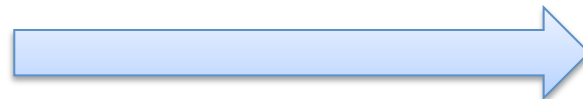
# Álgebra Relacional: Seleção

- 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

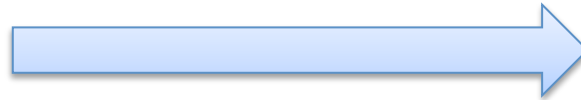
# Álgebra Relacional: Projeção

- 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 Relacional → Linguagem Declarativa → ISO/SQL (Structured Query Language)

**CREATE TABLE** países

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

Definição Dados

países		
id	nome	populacao

Manipulação  
Dados

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

Manipulação  
Dados

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

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



# Linguagem de Consulta: SQL

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

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

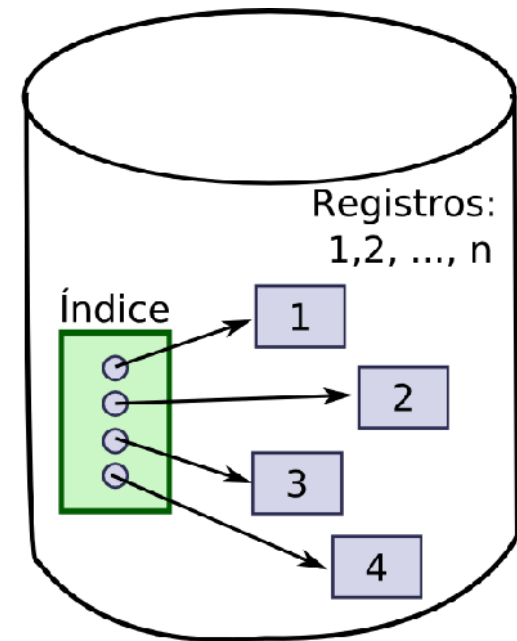
**Consulta**  
**(Não-Procedural)**

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

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

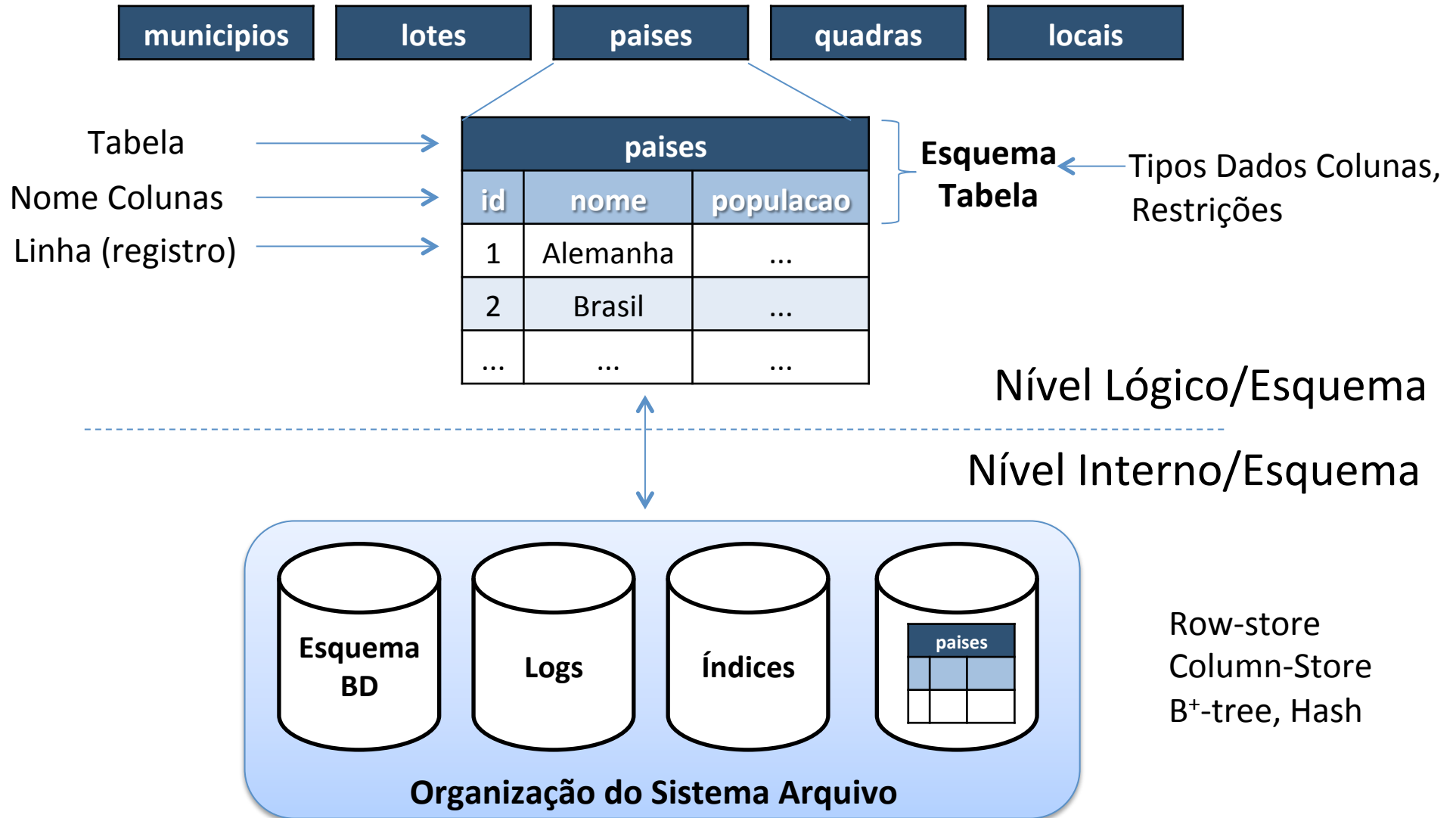
# Métodos de Acesso (Indexação)

- 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



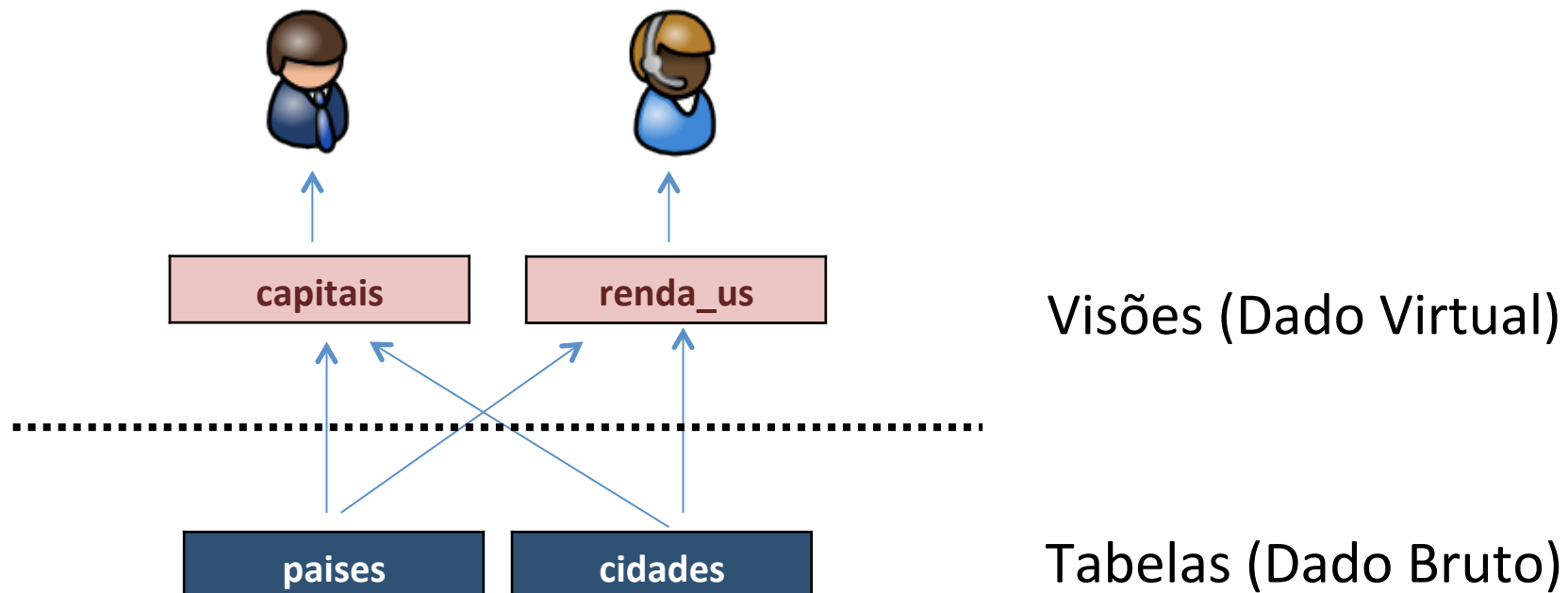
Fonte: Adaptado de Gray (1996)

# 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



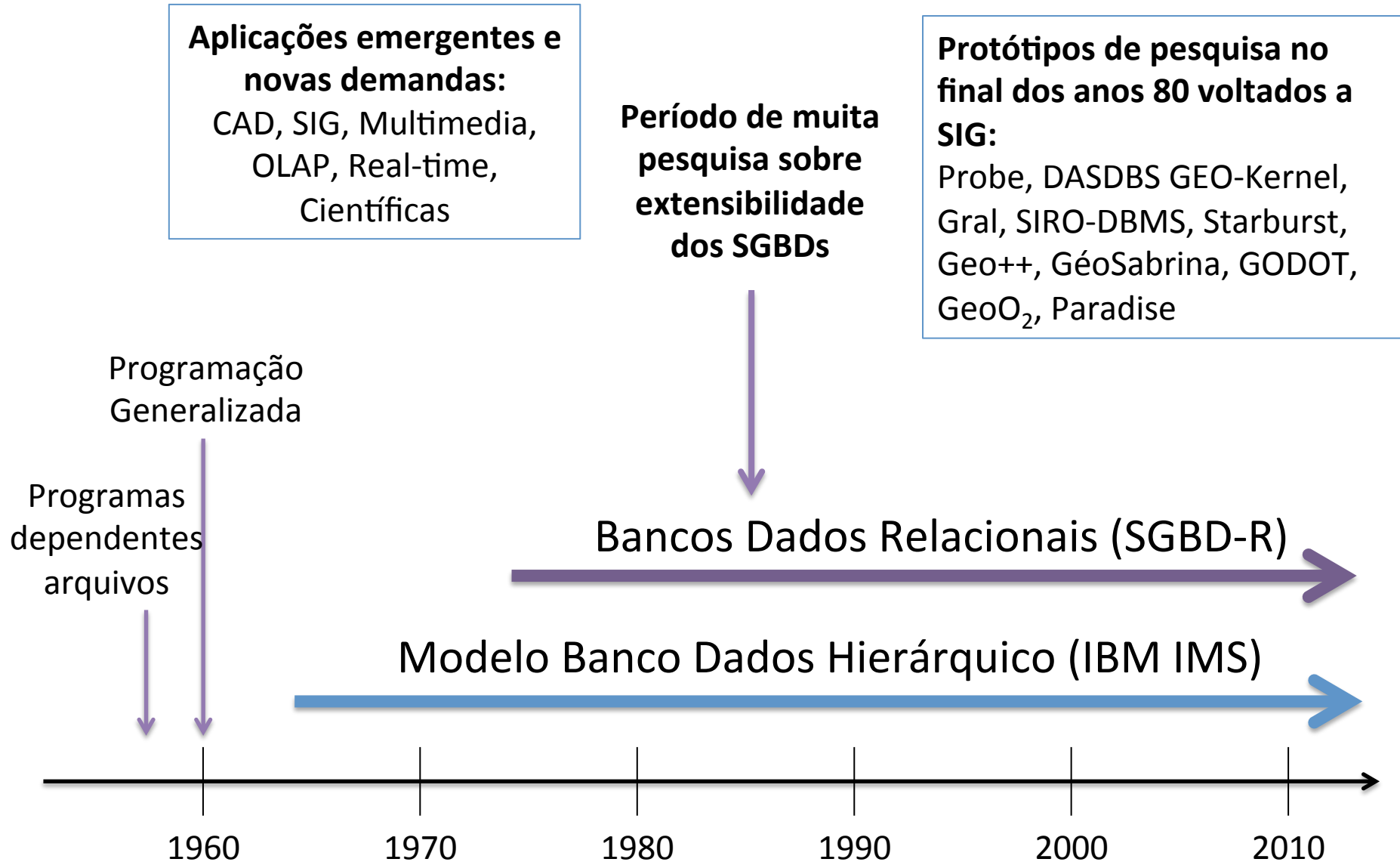
# Arquiteturas de SGBD-R

- Cliente/Servidor ou Embutido (ou embarcado)
- Em memória (In-memory)
- Paralelos ou Distribuídos
- Armazenamento Linha x Coluna:
  - linha: atributos do registro são colocados contiguamente no meio de armazenamento. (bom para aplicações OLTP)
  - coluna: todos os valores de uma coluna são armazenados contiguamente. (bom para aplicações otimizadas para leitura tais como data warehouses e OLAP)

# Outros Conceitos Importantes

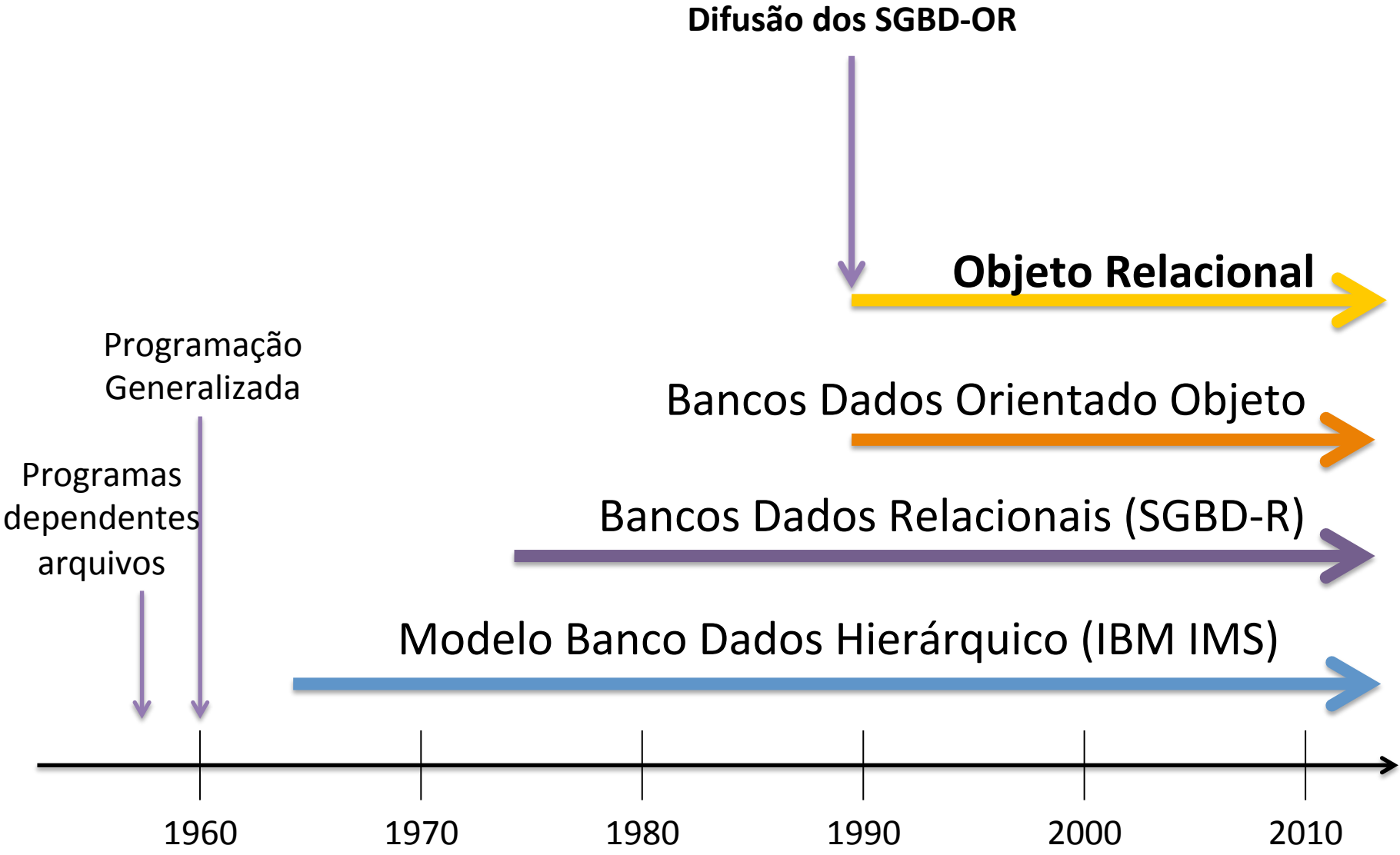
- Projeto de bancos de dados:
  - Modelo Entidade-Relacionamento (ER).
- Normalização:
  - Evitar anomalias com o projeto do banco de dados.
- Transações (ACID).
- Gatilhos (Trigger).
- Procedimentos Armazenados (Stored Procedure).

# Evolução das Tecnologias de Bancos Dados





# Evolução das Tecnologias de Bancos Dados



# SGBD Objeto-Relacionais

**CREATE TYPE** Address **AS**

```
(  
  street  VARCHAR(50),  
  city    VARCHAR(50),  
  zip     CHAR(8)  
);
```



**CREATE TABLE** Student

```
(  
  name    VARCHAR(50),  
  address Address  
);
```

**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 FUNCTION** ValidAddress **RETURNS BOOLEAN**  
**EXTERNAL NAME** '/address-module.so' **LANGUAGE 'C';**

**INSERT INTO** student **VALUES**('eduardo', ('albino sartori', 'ouro preto', '35400'));  
**SELECT** (address).city **FROM** student;

# 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

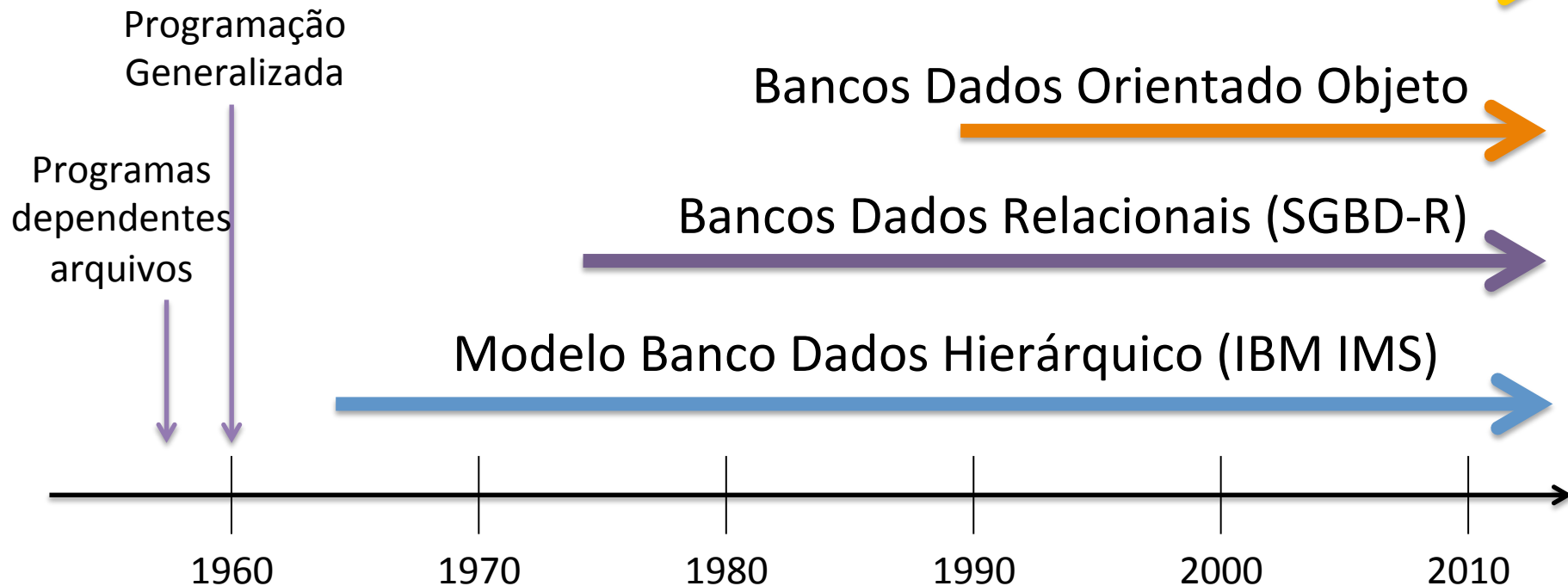
Geoespacial

Objeto Relacional

Bancos Dados Orientado Objeto

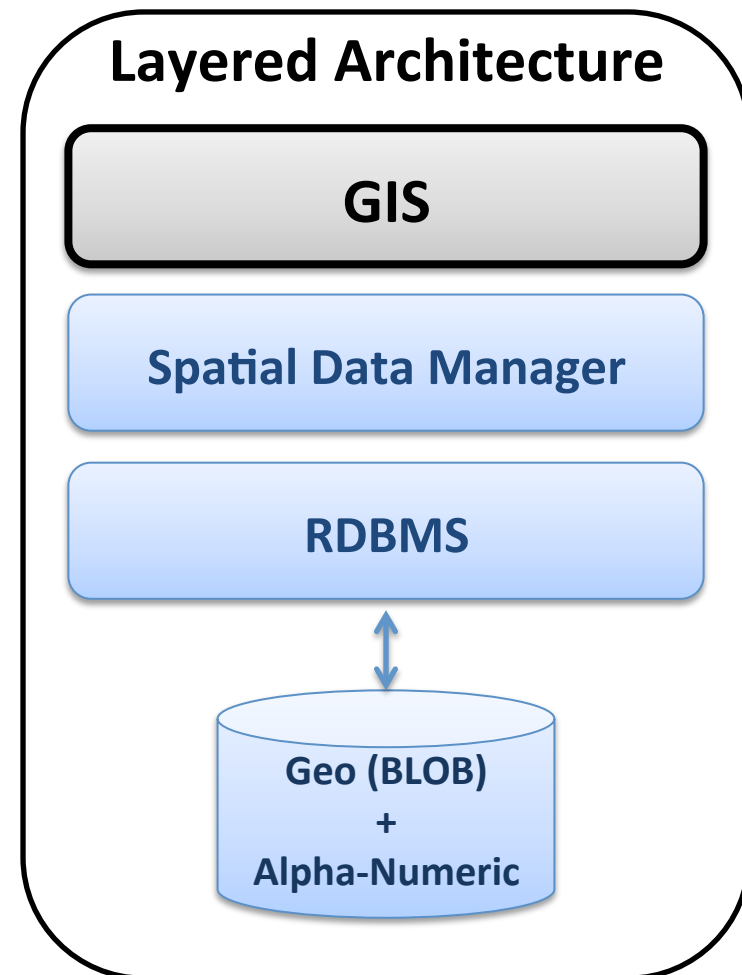
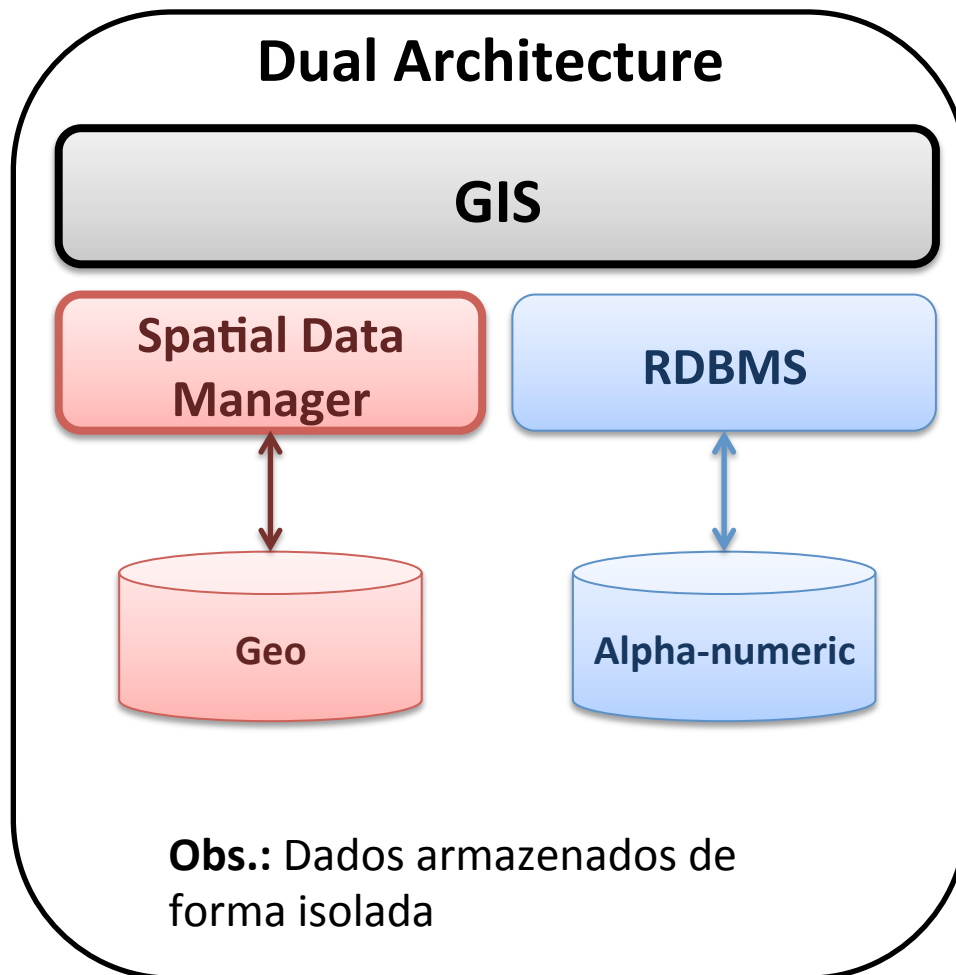
Bancos Dados Relacionais (SGBD-R)

Modelo Banco Dados Hierárquico (IBM IMS)

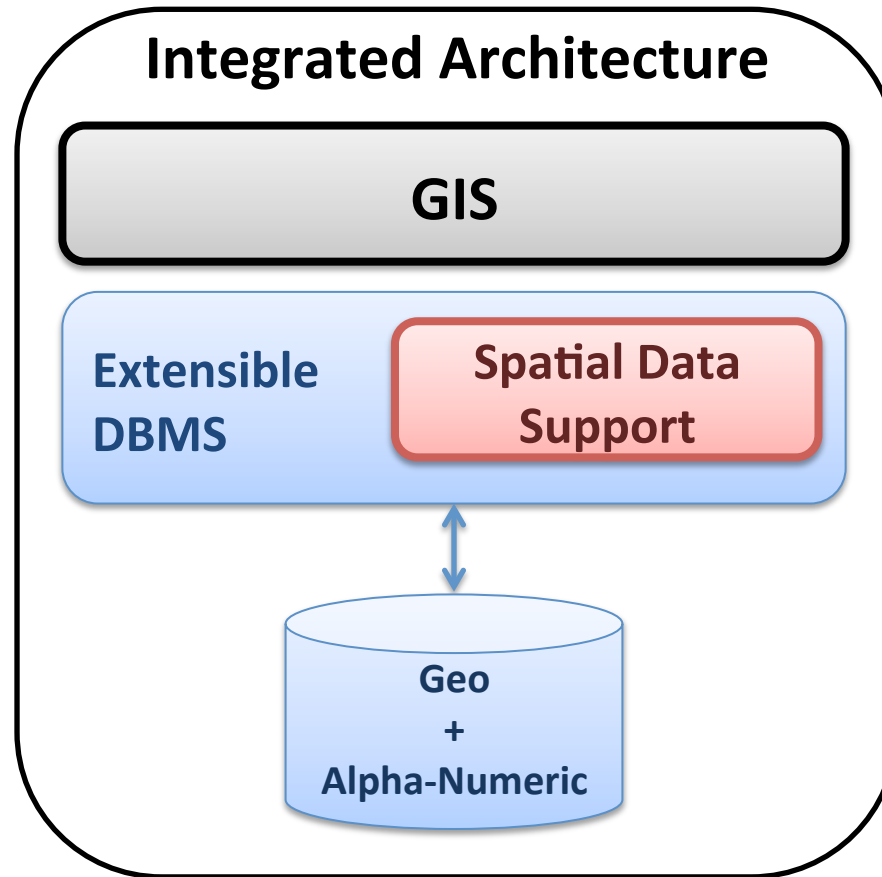


# SIG e SGBD-R

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





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



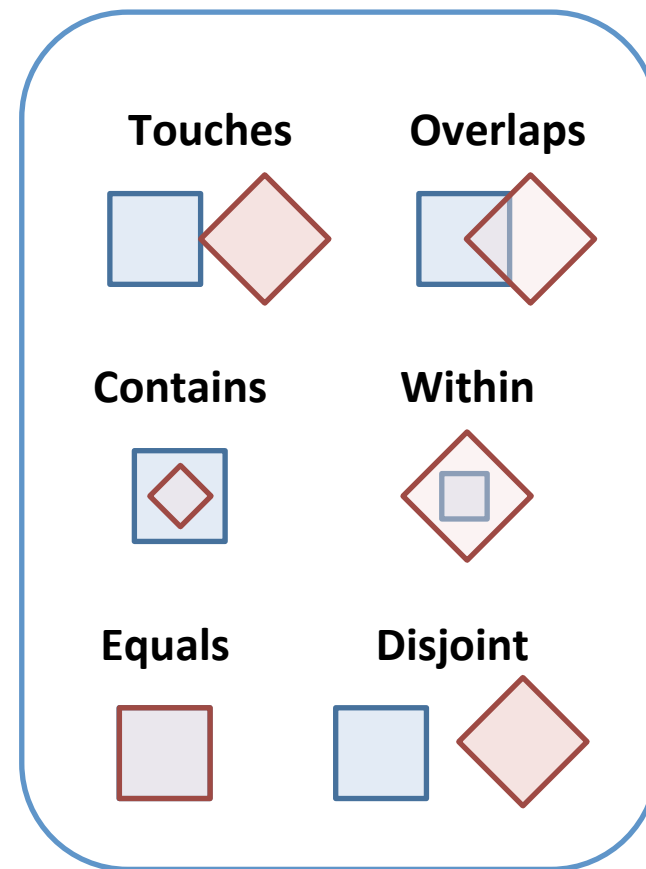
# SIG e SGBD-R: Como passou a ser esta integração?

- 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



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

- Índices Espaciais: árvores-R, Quadrees, Fixed-Grid.
- Armazenamento dados matriciais.
- Armazenamento baseado em modelos topológicos.
- Redes espaciais: roteamento, análise de fluxo.

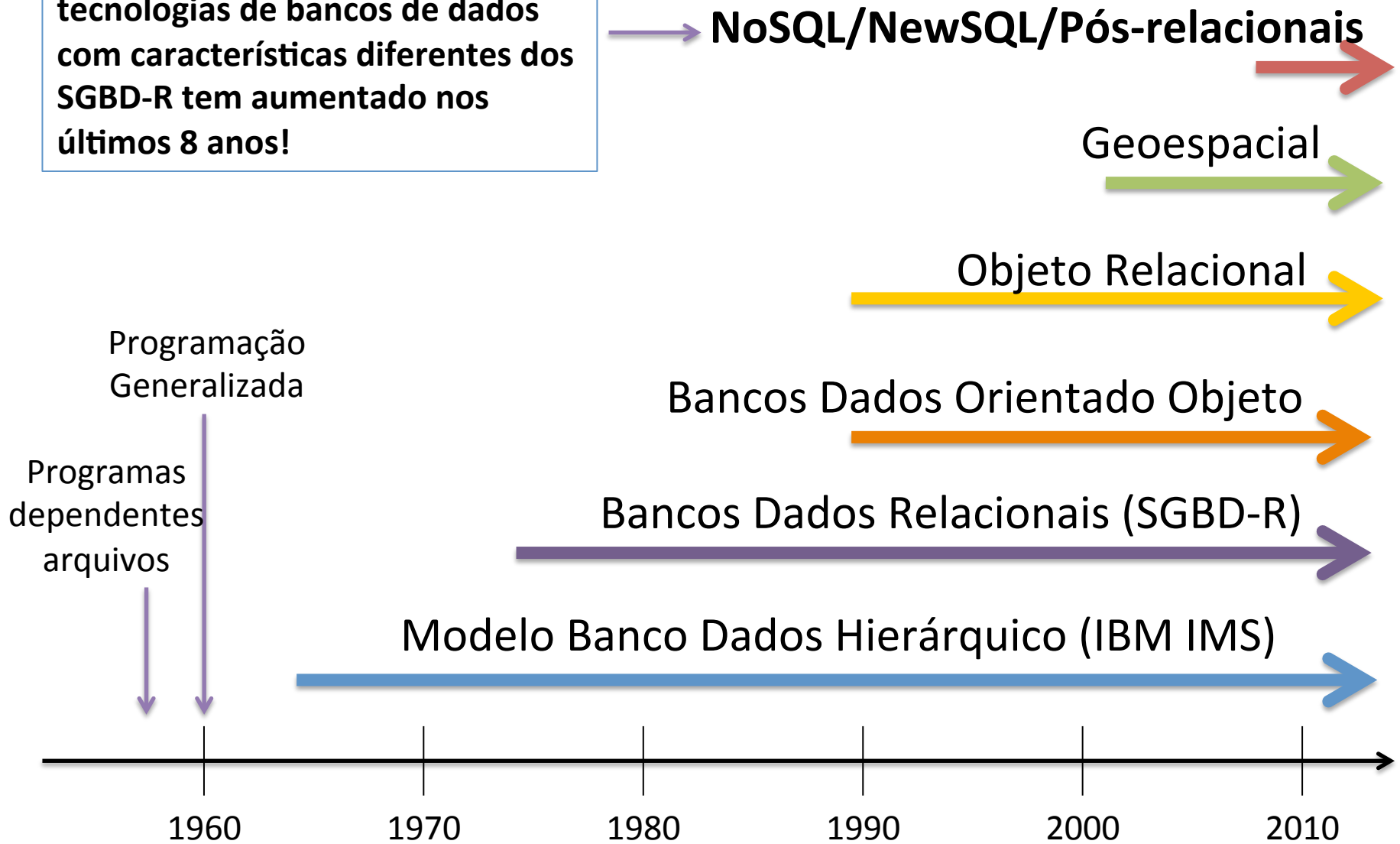
# Bancos de Dados x Information Retrieval

- Bancos de dados → Informações estruturadas
- IR → mais voltado para informações não estruturadas como processamento de documentos e texto livre.



# 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

# Referências

- ELMASRI, R.; NAVATHE, S. B. ***Fundamentals of database systems***. Addison Wesley, 2006. 1139p.
- GRAY, J. ***Evolution of Data Management***. IEEE Computer 29(10): 38-46, 1996.
- 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.

# Referências

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