



Bancos de Dados Geográficos

Curso de Verão - Geoinformática
27 de Janeiro de 2020

Evolução das Tecnologias de Bancos de Dados

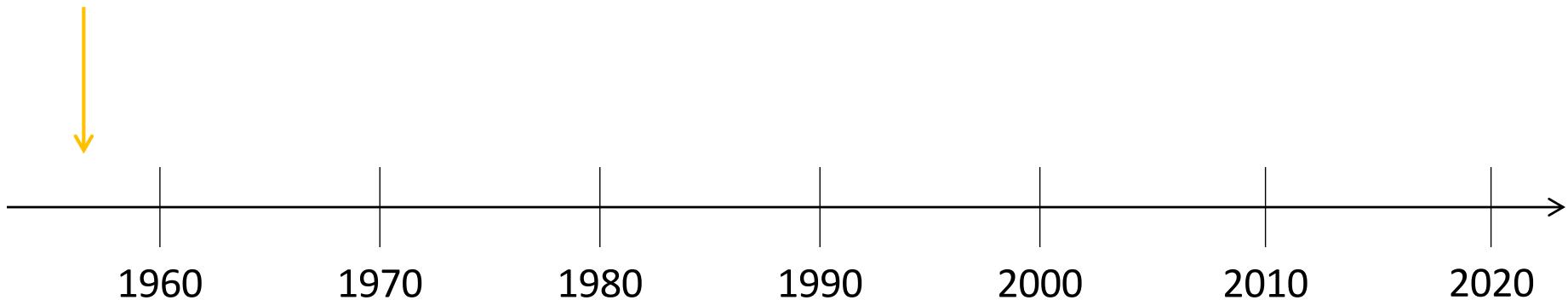
Gilberto Ribeiro de Queiroz

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 3.0 Unported” license.

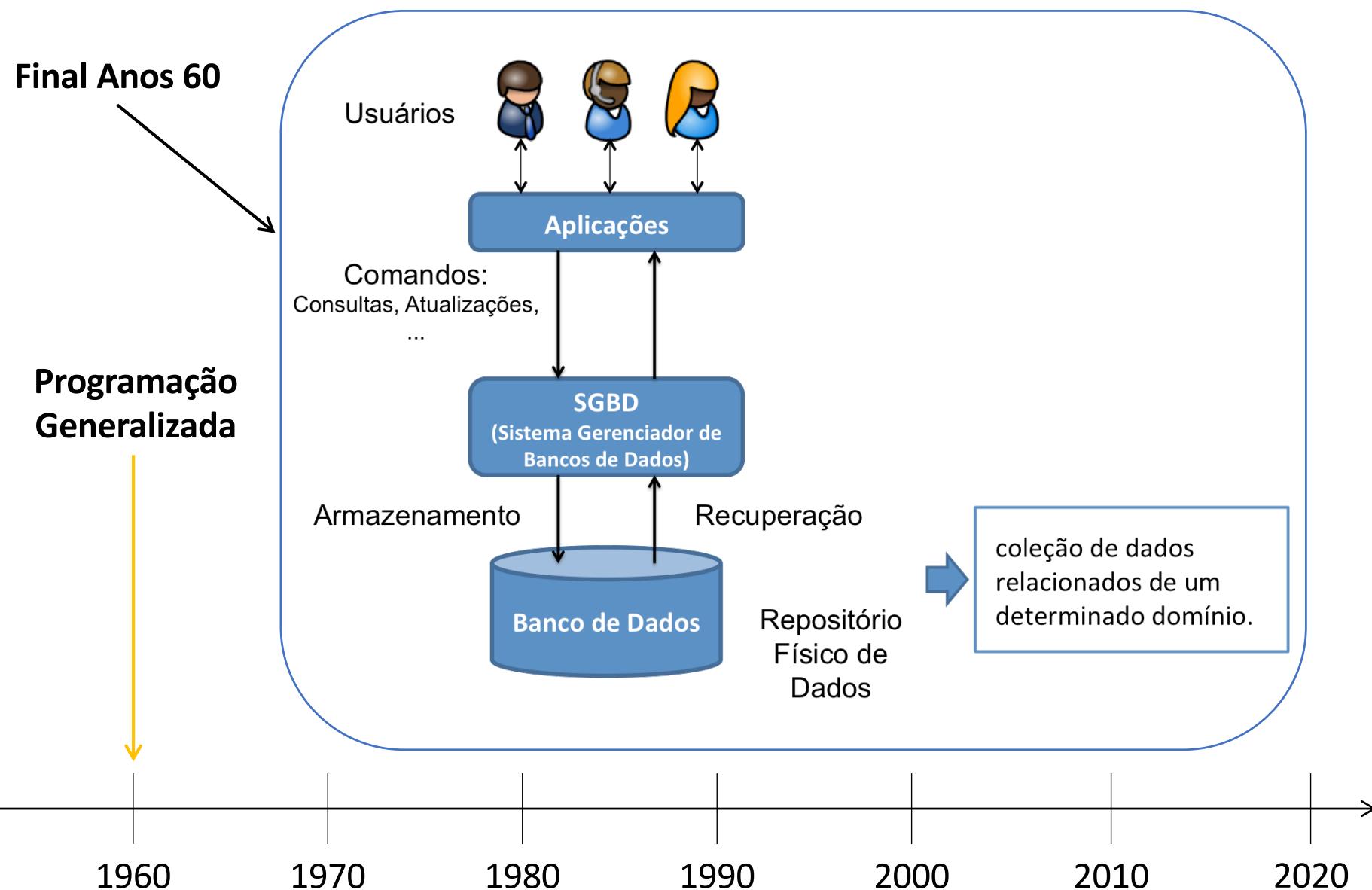


Evolução das Tecnologias de Bancos Dados

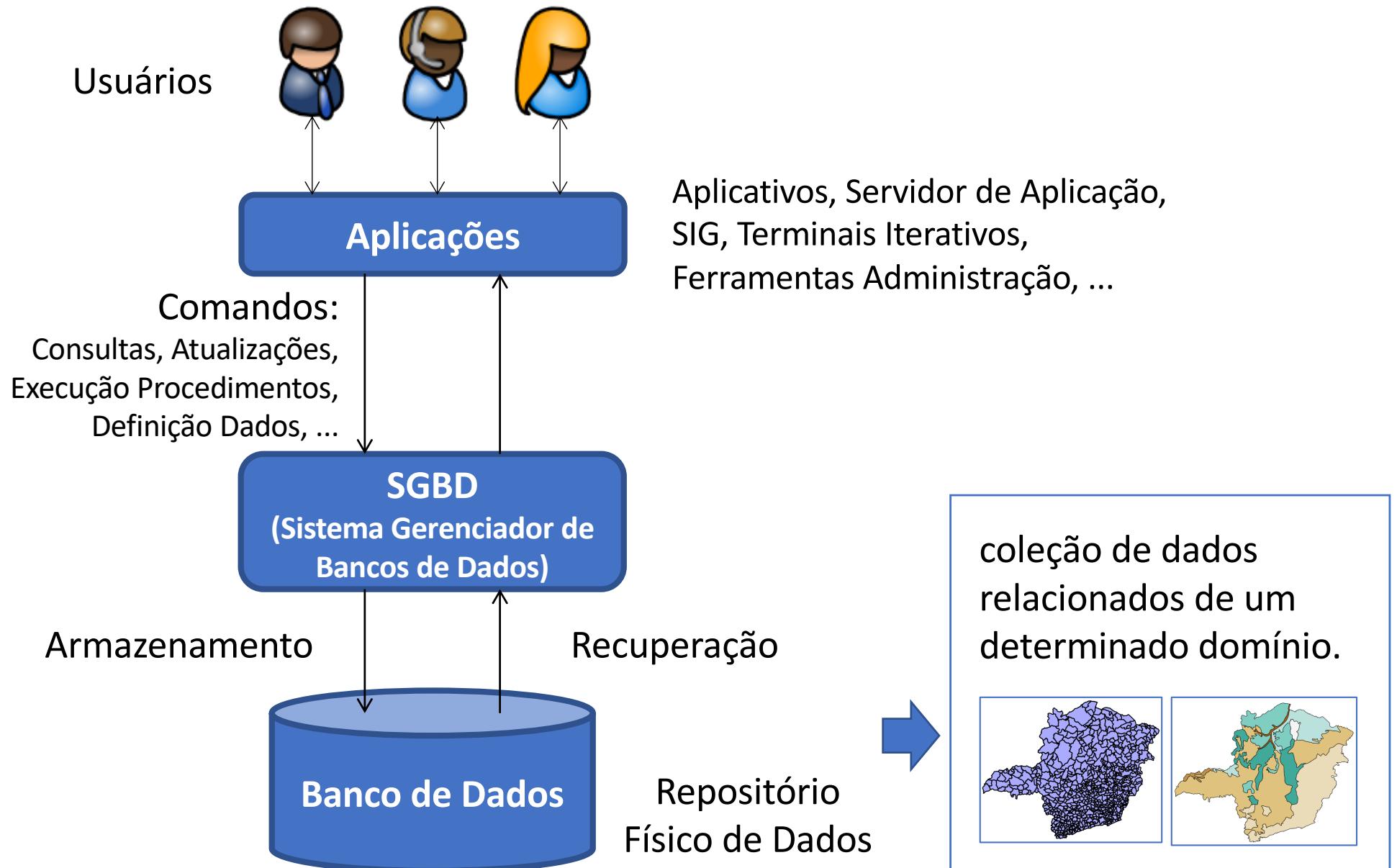
**Programas
dependentes
arquivos**



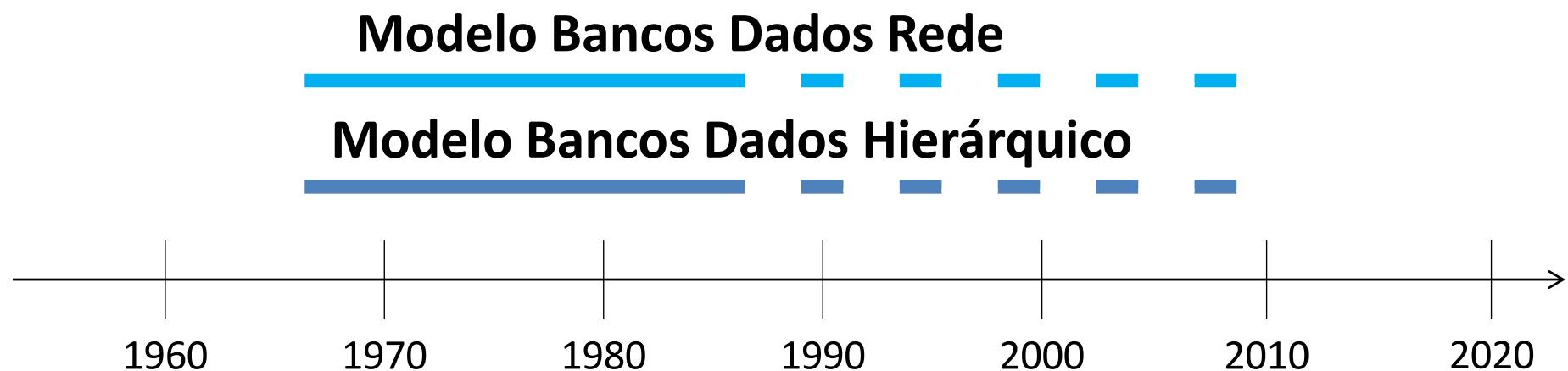
Evolução das Tecnologias de Bancos Dados



Sistemas de Bancos de Dados

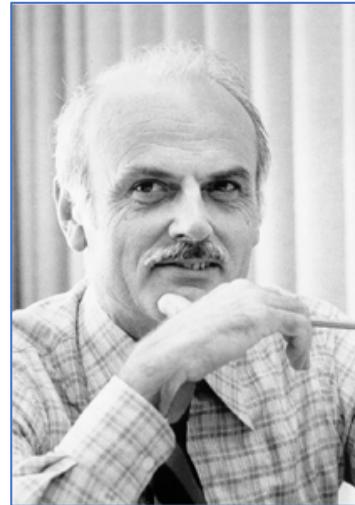


Evolução das Tecnologias de Bancos Dados



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.

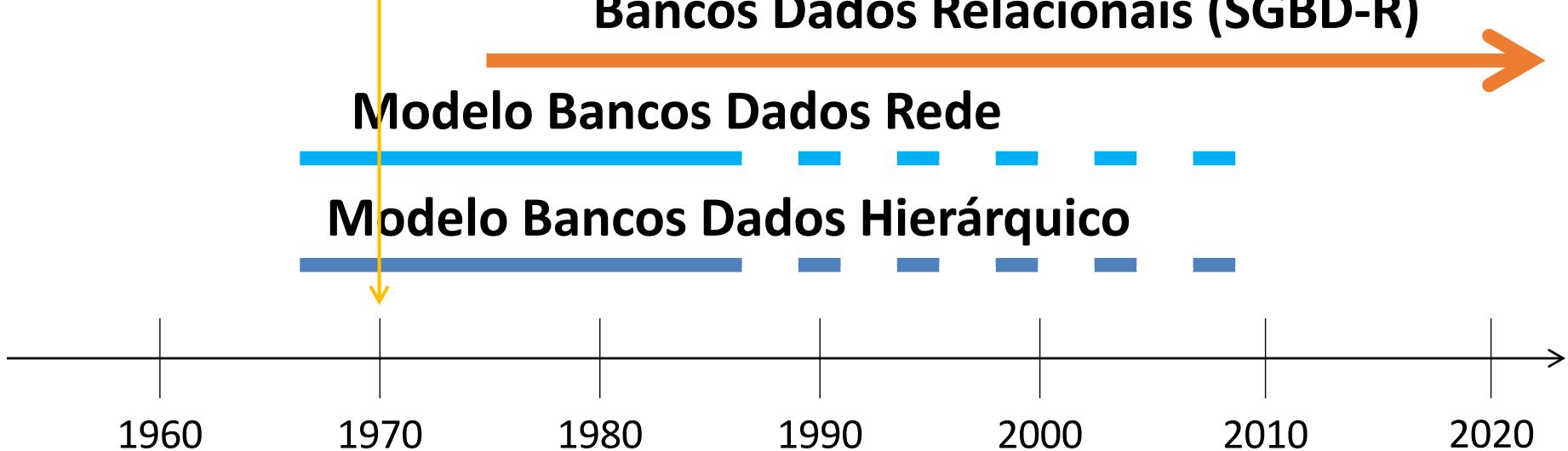
Modelo
Relacional

Edgar Frank Codd
Fonte: [Wikipedia](#)

Bancos Dados Relacionais (SGBD-R)

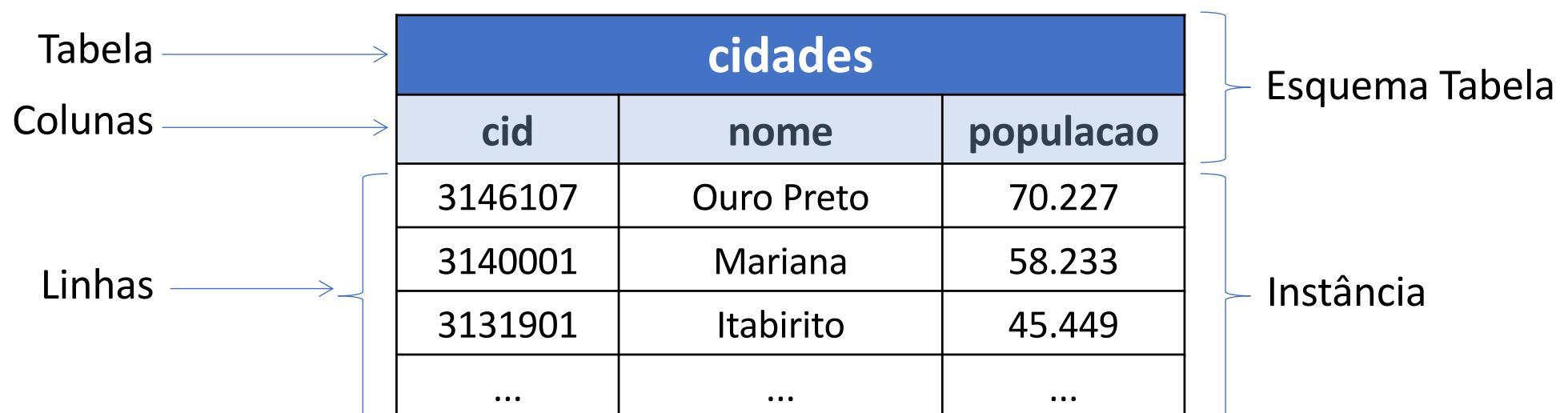
Modelo Bancos Dados Rede

Modelo Bancos Dados Hierárquico



Relação (ou Tabela)

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



Relacionamentos entre tabelas

países	
pid	nome
1	Alemanha
2	Brasil
...	...

cidades			
cid	nome	populacao	pais_id
3146107	Ouro Preto	70.227	2
3140001	Mariana	58.233	2
9879999	Munster	291.754	1
3131901	Itabirito	45.449	2
...

países_x_cidades				
pid	p_nome	cid	c_nome	c_populacao
2	Brasil	3146107	Ouro Preto	70.227
2	Brasil	3140001	Mariana	58.233
1	Alemanha	9879999	Munster	291.754
2	Brasil	3131901	Itabirito	45.449
...

Álgebra Relacional

- **Exemplo:**

- **Operador de Seleção:** seleciona tuplas de uma relação que satisfazem um certo predicado (ou condição).
- **Consulta:** selecionar as cidades com população acima de 60.000 habitantes.

cidades			
cid	nome	populacao	pais_id
...	Ouro Preto	70.227	2
...	Mariana	58.233	2
...	Munster	291.754	1
...	Itabirito	45.449	2

Tabela/Relação de Entrada

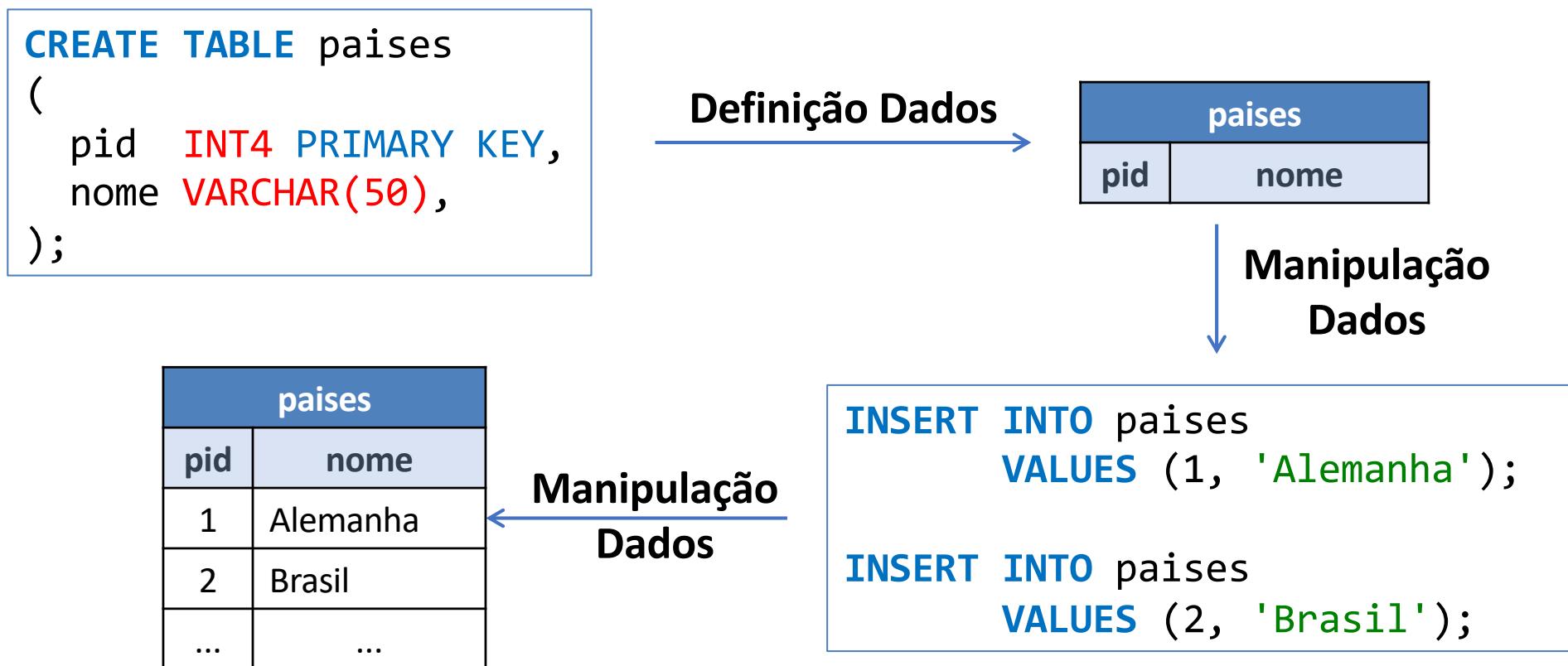
$$\sigma_{populacao > 60000}(cidades)$$


nova_relacão			
cid	nome	populacao	pais_id
...	Ouro Preto	70.227	2
...	Munster	291.754	1

Tabela/Relação de Saída

Linguagem de Consulta: SQL

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



Linguagem de Consulta: SQL

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

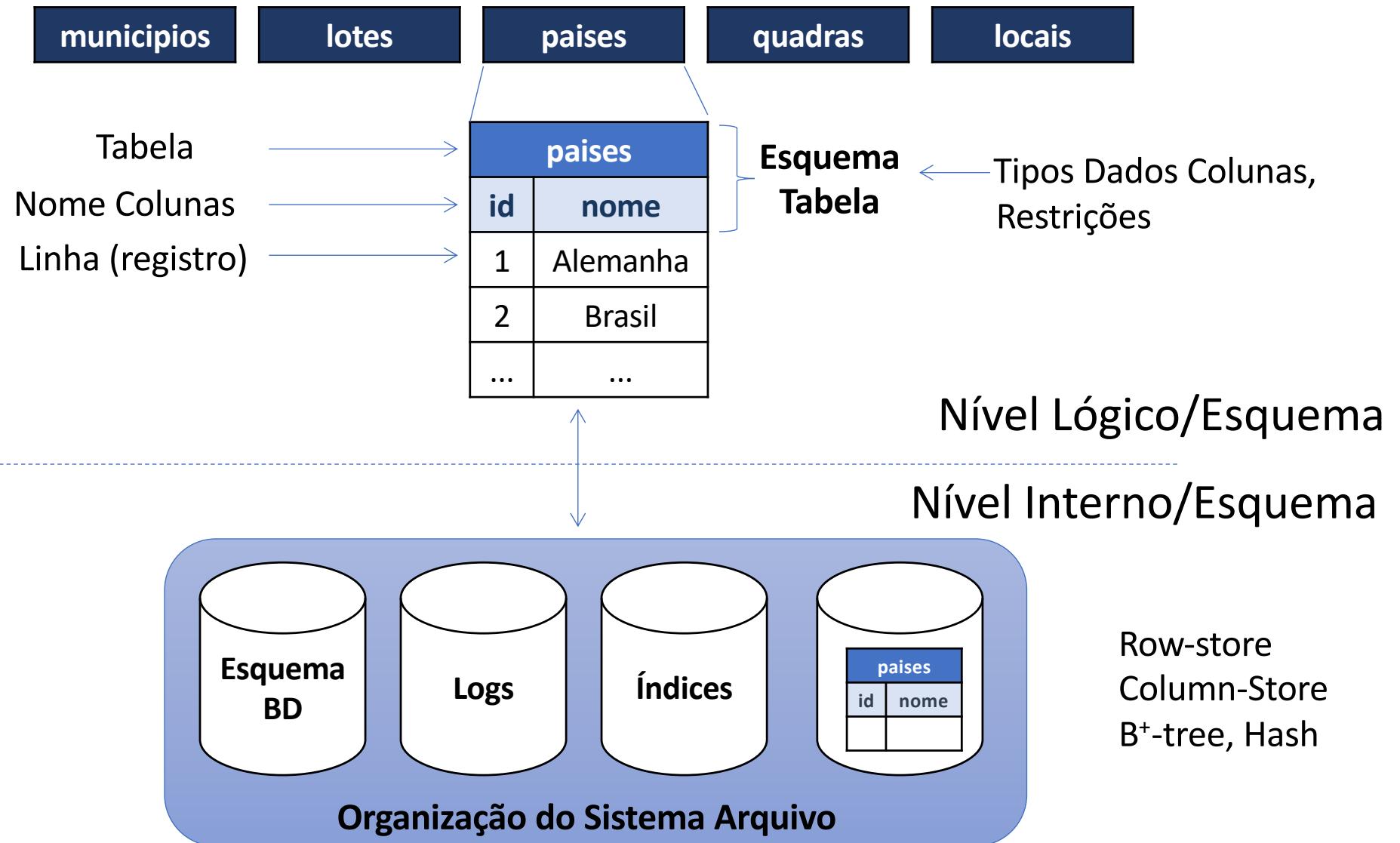
cidades			
cid	nome	populacao	pais_id
...	Ouro Preto	70.227	2
...	Mariana	58.233	2
...	Munster	291.754	1
...	Itabirito	45.449	2

```
SELECT nome  
FROM cidades  
WHERE populacao > 60000
```

Consulta (Não-Procedural)

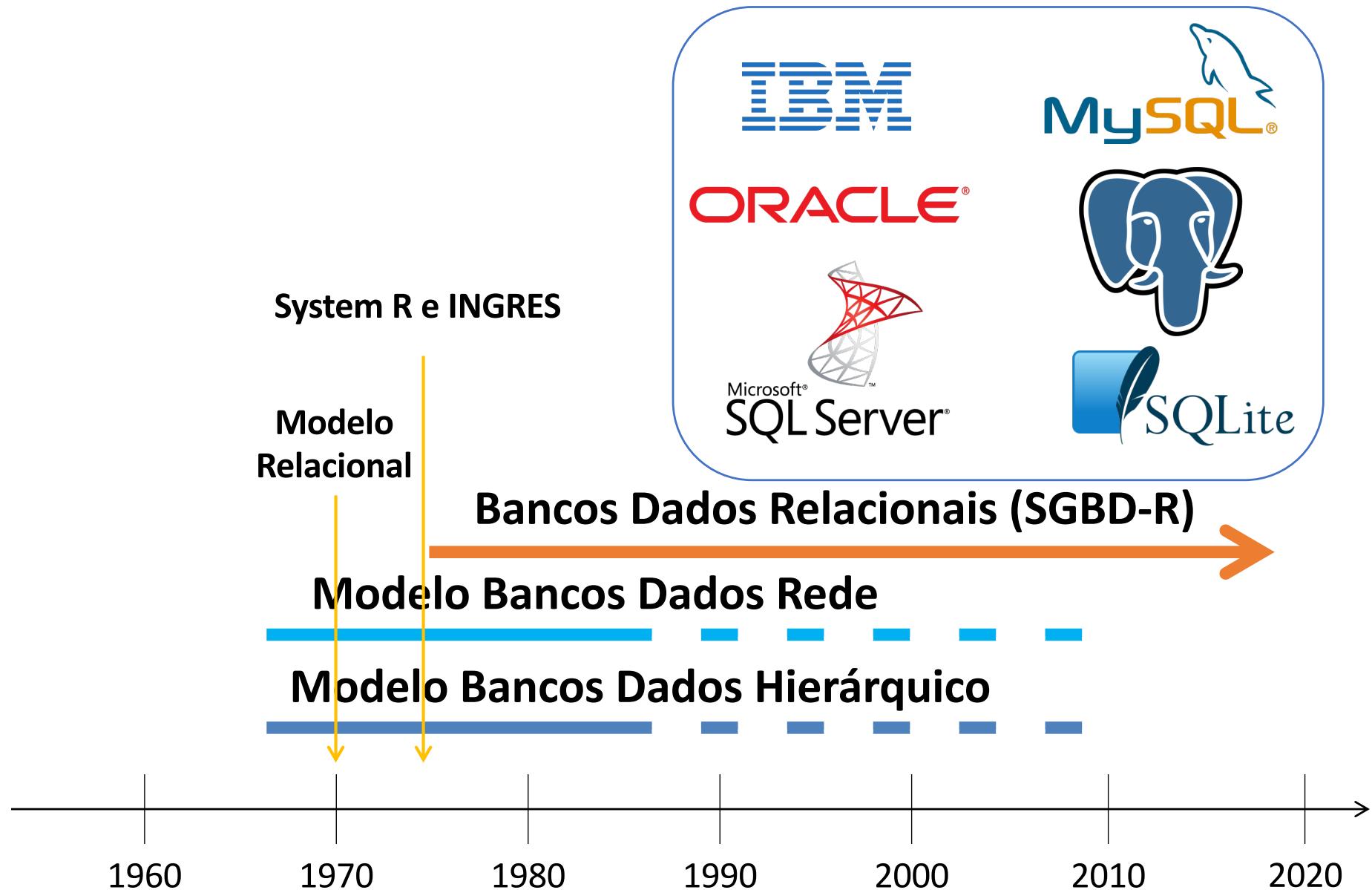
resultado
nome
Ouro Preto
Munster

Independência Física dos Dados

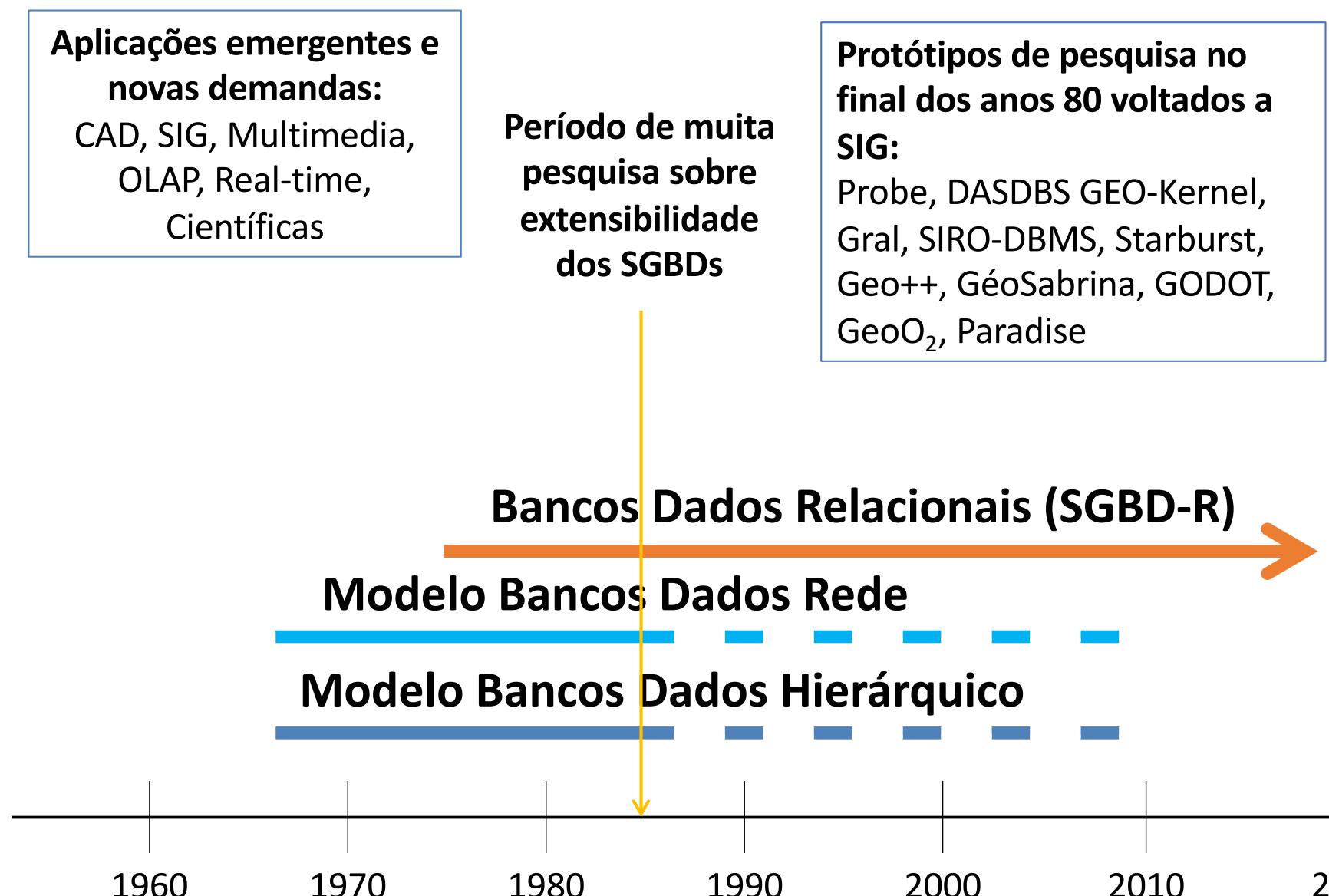


Fonte: Adaptado de Gray (1996)

Evolução das Tecnologias de Bancos Dados



Evolução das Tecnologias de Bancos Dados



Gral: An Extensible Relational Database System for Geometric Applications

Ralf Hartmut Güting

Fachbereich Informatik, Universität Dortmund
D-4600 Dortmund 50, West Germany

Abstract: We describe the architecture of a relational database system that is extensible by user-defined data types and operations, including relation operations. The central concept is to use languages based on many-sorted algebra to represent queries as well as query execution plans. This leads to a simple and clean extensible system architecture, eases the task of an application developer by providing a uniform framework, and also simplifies rule-based optimization. As a case study the extensions needed for a geometric database system are considered.

1. Introduction

Much of the database research of recent years was aimed at providing a better support for non-standard applications such as office information systems, geographic information systems, CAD databases, etc. A common need of these applications is the representation and manipulation of more complex objects than those representable by a tuple of a relation in the traditional relational model, for example, an office form, a complete map or a river, say, in a geographic information system, or the design of a VLSI circuit.

A fundamental choice for the representation of a complex

A lot of work has been done to support the modeling of visible object structures. Enhancements to the relational model have been proposed by linking together tuples to represent an object either explicitly [Co79, HaL82] or implicitly, through the use of nested relations [ScS86]. Most of the more recent data models and system proposals do also support structural object orientation, for example [MaD86, CaD88, PiT86].

The idea of allowing application-specific abstract data types as base types, or attribute domains, of a database system was perhaps first put forward in [StRG83]. Since base types need to be implemented in a programming language and because they are application-specific, a user must be able to implement such a type and to add it to a database system. This observation has led to efforts by several groups to construct *extensible* database systems. Two directions can be distinguished. One is to select a data model and to implement for this data model a system with well-defined interfaces for user extensions. This is the approach chosen by the POSTGRES [StR86] and Starburst [Schw86] projects, based on the relational model, and within the PROBE project [Daya87] for an extended functional data model. A different view is taken in the EXODUS [Care86] and GENESIS [Bato86] projects where a collection of powerful tools for

R. H. Güting. 1989. Gral: an extensible relational database system for geometric applications. In Proceedings of the 15th international conference on Very large data bases (VLDB '89). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 33-44.

THE **GEO++** SYSTEM: AN EXTENSIBLE GIS

Tom Vijlbrief

TNO Institute for Perception,

P.O. Box 23, 3769 ZG Soesterberg, The Netherlands.

Email: tom@izf.tno.nl

and

Peter van Oosterom

TNO Physics and Electronics Laboratory,

P.O. Box 96864, 2509 JG The Hague, The Netherlands.

Email: oosterom@fel.tno.nl

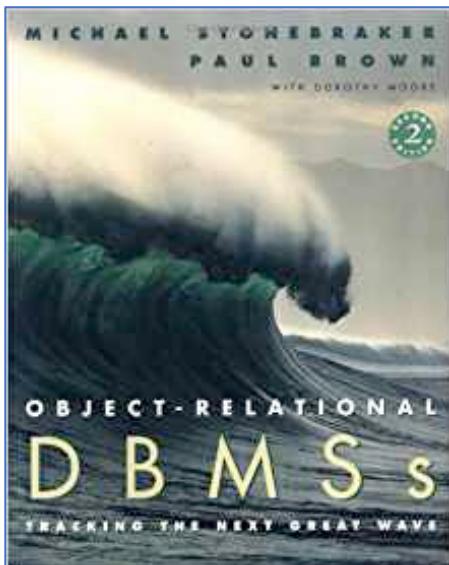
In this paper we present a classification of the architectures of Geographic Information Systems (GIS). Most commercial GISs are closed. This means that if certain functionality is not available, it is impossible for the users to extend or modify the system for their own purpose. We then present a solution based on the extensible database management system “Postgres”, in which new data types and operators may be defined. The resulting extensible GIS is called *GEO++*. We illustrate this powerful capability with two examples.

1 Introduction

Most commercial Geographic Information Systems (GISs) are based on a relational DataBase Management System (DBMS), such as Oracle or Ingres. One obvious drawback of the standard DBMSs is that they cannot manipulate geographic data. That is, there are no ge-

Rowe, & Hirohama, 1990) that we implemented. Section 4 enumerates the basic capabilities of our Postgres GIS front-end *GEO++*. The implementation of the system has been described in an earlier paper (van Oosterom & Vijlbrief, 1991). The real power of *GEO++* is demonstrated in Section 5, in which the system is extended with user-defined types.

Evolução das Tecnologias de Bancos Dados



Stonebraker et al. (1996)

Difusão dos SGBD-OR

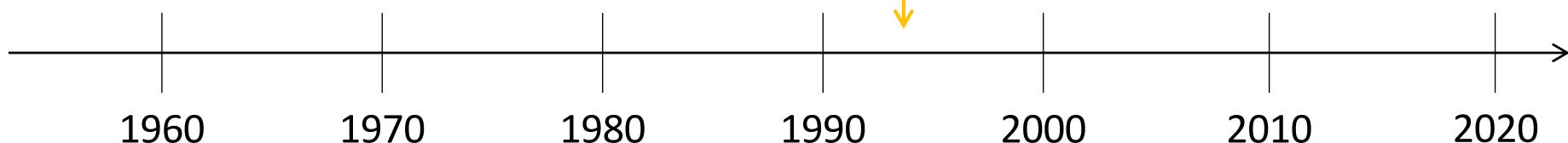
Objeto Relacional

Bancos Dados Orientado Objeto

Bancos Dados Relacionais (SGBD-R)

Modelo Bancos Dados Rede

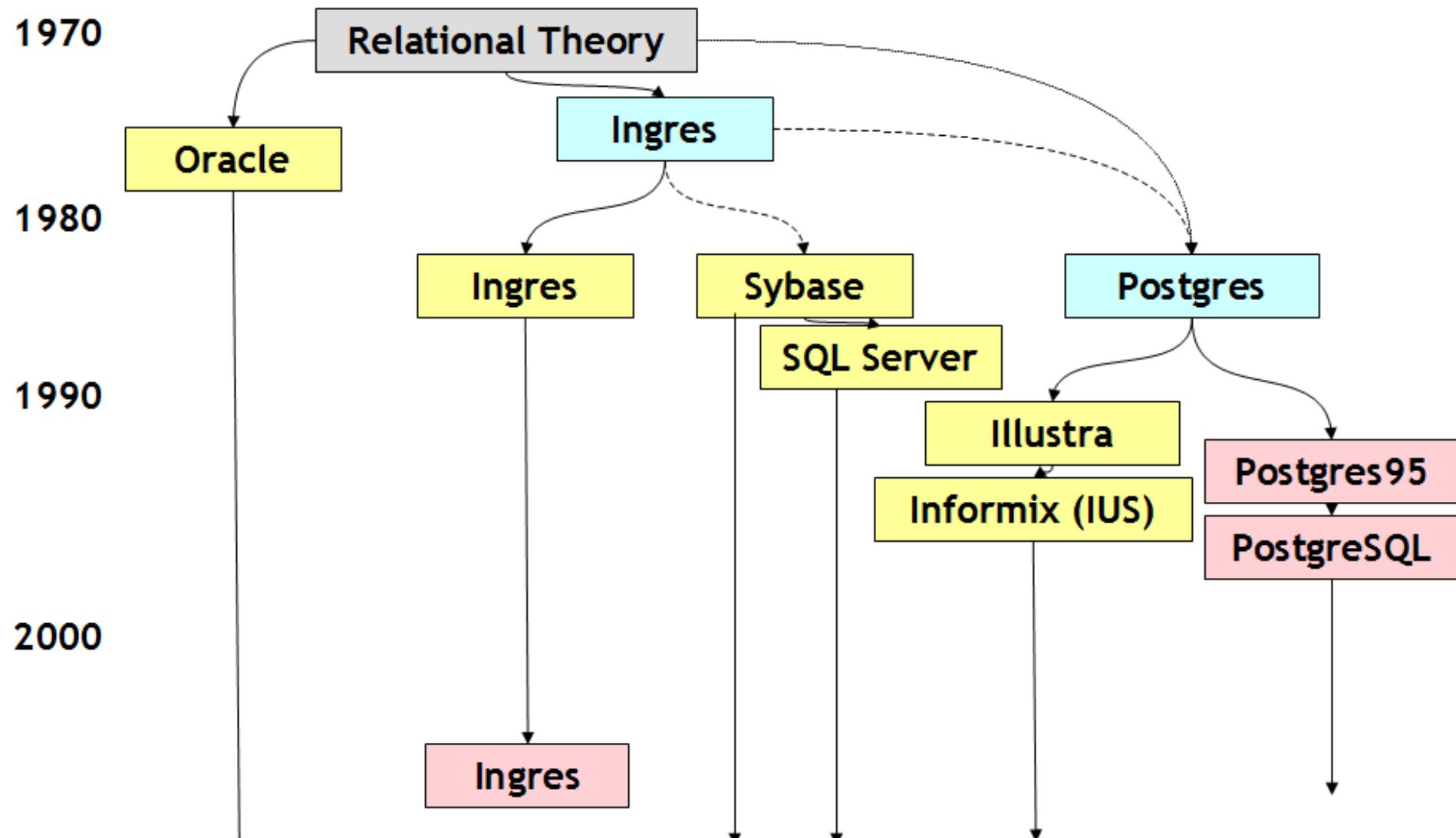
Modelo Bancos Dados Hierárquico



PostgreSQL

- É um sistema gerenciador de bancos de dados objeto-relacional open-source (licença ao estilo BSD):
 - <http://www.postgresql.org>
- O código fonte do seu núcleo encontra-s escrito na Linguagem de Programação C.
- Disponível para diversas plataformas: Linux, Mac OS X e Microsoft Windows.
- Versão que usaremos durante o curso: 11.x

História do PostgreSQL



Fonte: Ramsey (2007)

Prática

Criando e Consultando Tabelas no PostgreSQL

Sistemas Gerenciadores de Bancos de Dados Objeto-Relacional (SGBD-OR)

- Em um SGBD-OR, um **tipo de dado** (*data type*) é definido por uma **representação de armazenamento** juntamente com **operadores e funções** sobre este tipo.
- A flexibilidade do sistema de tipos dos SGBD-OR os tornam uma ferramenta poderosa para modelar aplicações mais complexas, tais como GIS.

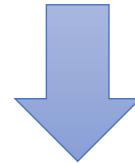
SGBD-OR: User Defined Types

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

O comando **CREATE TYPE** permite definir a **representação de armazenamento** para o tipo **geo_point**.

SGBD-OR: User Defined Types

```
CREATE TABLE escolas  
(  
    gid           INTEGER PRIMARY KEY,  
    nome          VARCHAR(100),  
    localizacao  GEO_POINT  
);
```



```
INSERT INTO escolas  
VALUES (3, 'Instituto São José',  
        '(1, 2, 4326)::GEO_POINT);
```

SGBD-OR: User Defined Functions

```
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);
    dy = (first.y - second.y);
    RETURN sqrt(dx * dx + dy * dy);
END;
$$
LANGUAGE plpgsql;
```

O comando **CREATE FUNCTION** permite **criar ou estender** a álgebra de um determinado tipo de dado, neste caso o tipo **geo_point**.

SGBD-OR: User Defined Functions

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

As funções definidas pelo usuário passam a fazer,
automaticamente, parte da **linguagem de consulta do SGBD**.

SGBD-OR: Sobrecarga de Operadores

```
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: Sobrecarga de Operadores

```
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);
```

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

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

Hash

Operation	Strategy Number
equal	1

SGBD-OR: UDTs mais Complexos

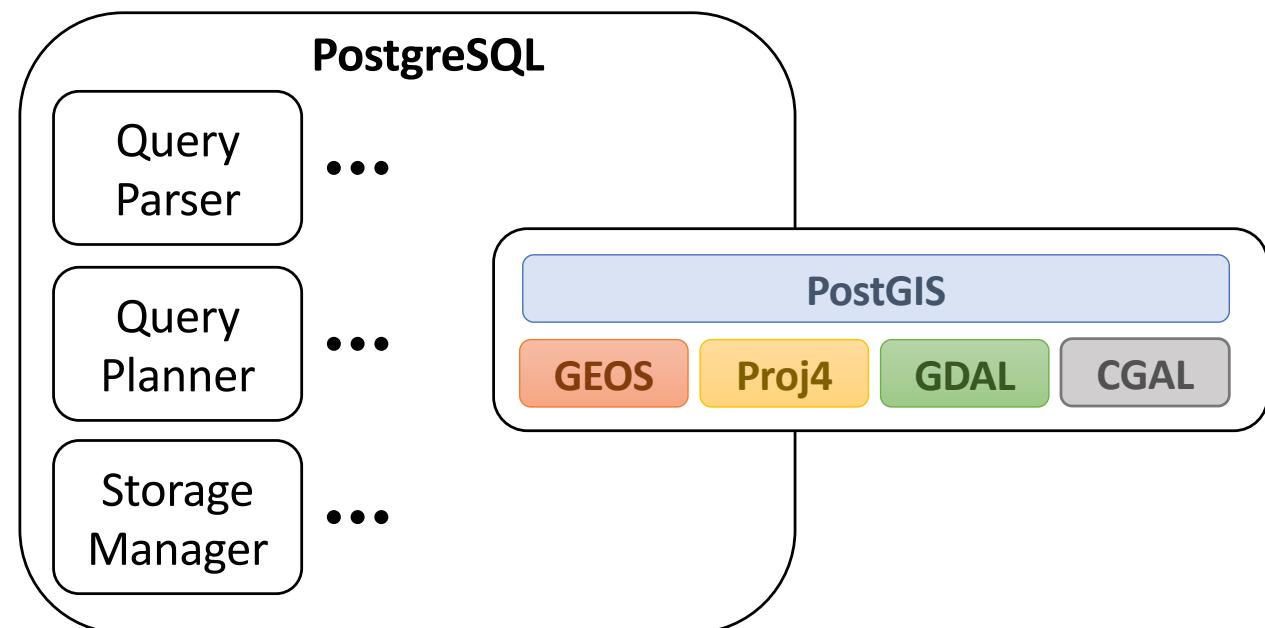
```
CREATE TYPE Geometry
(
    internallength = variable,
    input = geometry_in,
    output = geometry_out,
    send = geometry_send,
    receive = geometry_recv,
    typmod_in = geometry_typmod_in,
    typmod_out = geometry_typmod_out,
    delimiter = ':',
    alignment = double,
    analyze = geometry_analyze,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

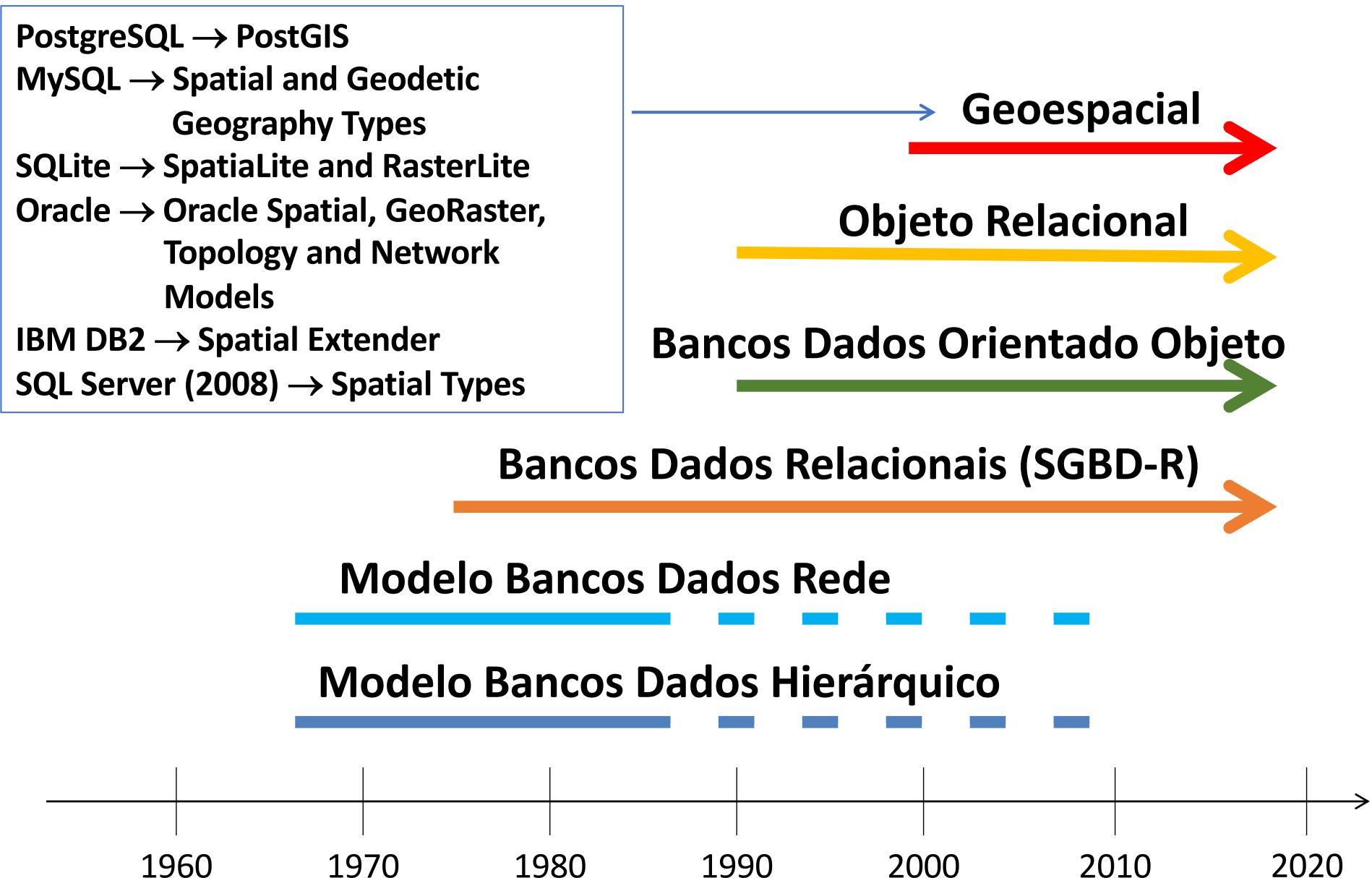
PostGIS



- Projeto de software livre (GPLv2) desenvolvido inicialmente pela empresa Canadense Refractions Research: <http://postgis.refractions.net>
- Extensão geográfica para o SGBD-OR PostgreSQL:
 - Inicialmente:
 - Tipos geométricos e operadores espaciais OGC SFS
 - Índice espacial: árvore-R sobre GiST.
 - Atualmente:
 - Tipos circulares e compostos, 3D, Tipo geográfico, Raster, Topologia, Redes, Geocodificação de endereços

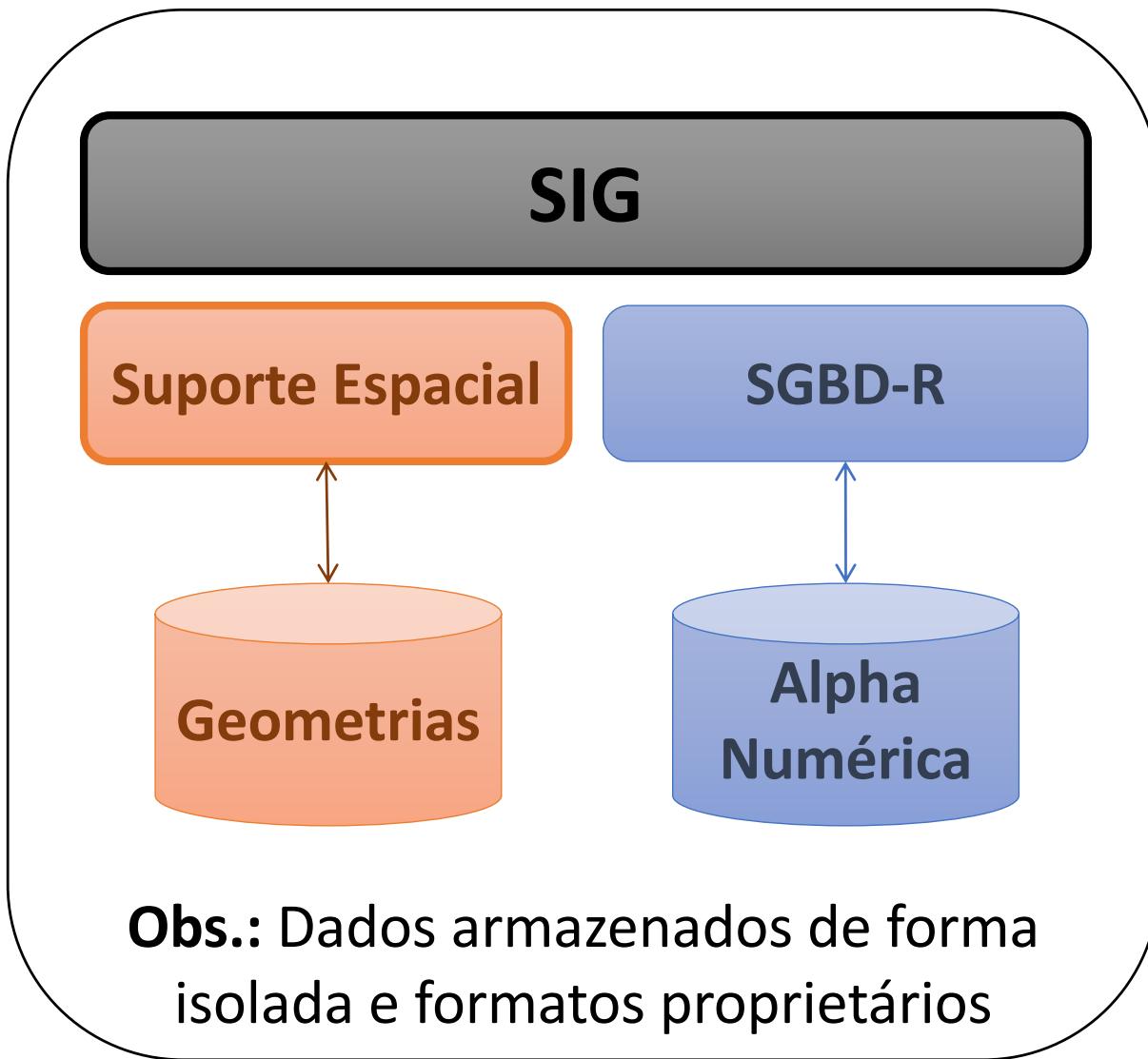


Evolução das Tecnologias de Bancos Dados

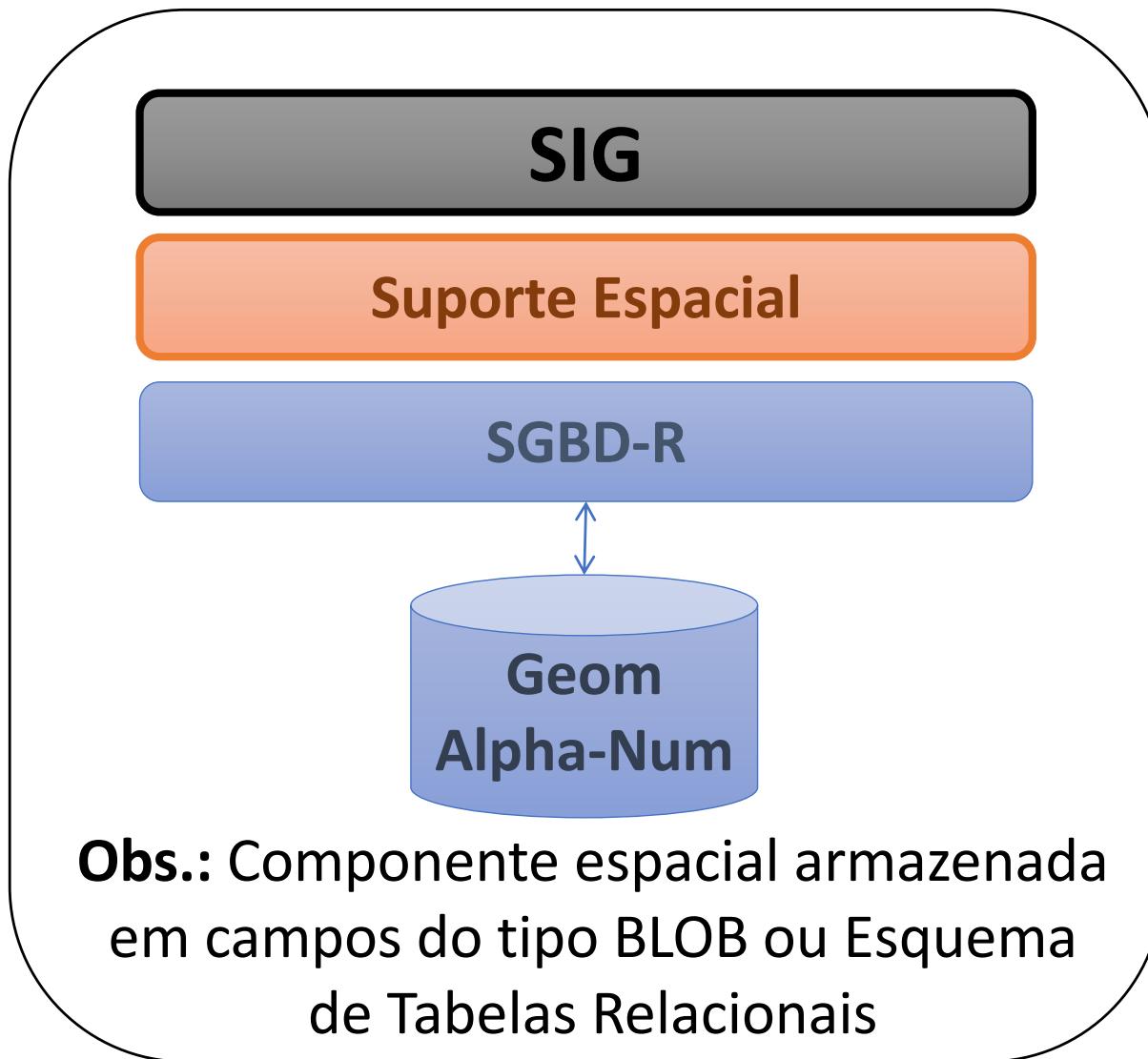


Antes dos anos 2000, como era a integração SIG e SGBD?

Arquitetura Dual



Arquitetura em Camadas



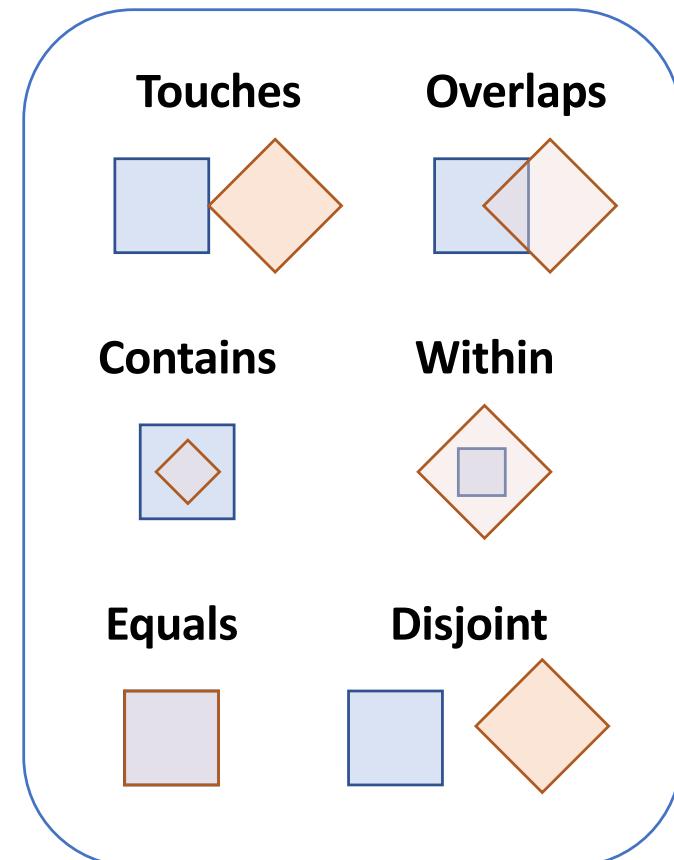
Com a integração do suporte
espacial nos SGBD-R como passou a
ser a integração SIG \leftrightarrow SGBD-R?

Arquitetura Integrada: Tipos de Dados Geoespaciais

Tabela com Feições

unidades_federativas				
ufid	nome	populacao	e_vida	fronteira
31	Minas Gerais	20.997.560	77	
35	São Paulo	44.396.484	77,8	
...

Operações Espaciais



Obs.: Padrões OGC Simple Feature e ISO/SQL-MM Spatial

Padrões SQL: OGC SFSQL e ISO SQL/MM Spatial



OGC Standards

Below is a list of OGC Implementation Standards.

Implementation Standards are different from the Abstract Specification. They are written for a more technical audience and detail the interface structure between software components. An interface specification is considered to be at the implementation level of detail if, when implemented by two different software engineers in ignorance of each other, the resulting components plug and play with each other at that interface.

Any Schemas (xsd, xslt, etc) that support an approved Implementation Standard can be found in the official [OGC Schema Repository](#).

▼ OGC® Standards

- [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](#)
- [Geospatial eXtensible Access Control](#)

♦ Document Title (click to view/download)	♦ Version	♦ Doc.#	♦ Editor	♦ Date
CF-netCDF3 Data Model Extension standard	3.1	11-165r2	Ben Domenico and Stefano Nativi	2013-01-03

Evolução das Tecnologias de Bancos Dados

Grande variedade de novas tecnologias de bancos de dados!

NoSQL/Pós-relacionais

Geoespacial

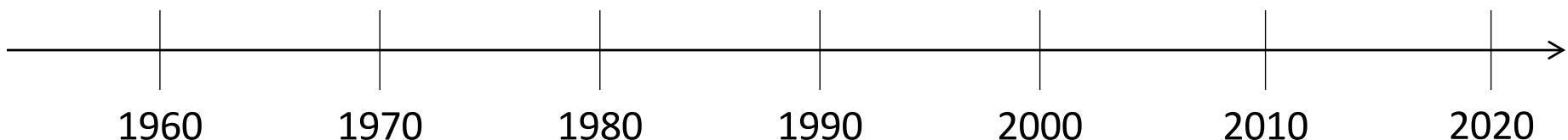
Objeto Relacional

Bancos Dados Orientado Objeto

Bancos Dados Relacionais (SGBD-R)

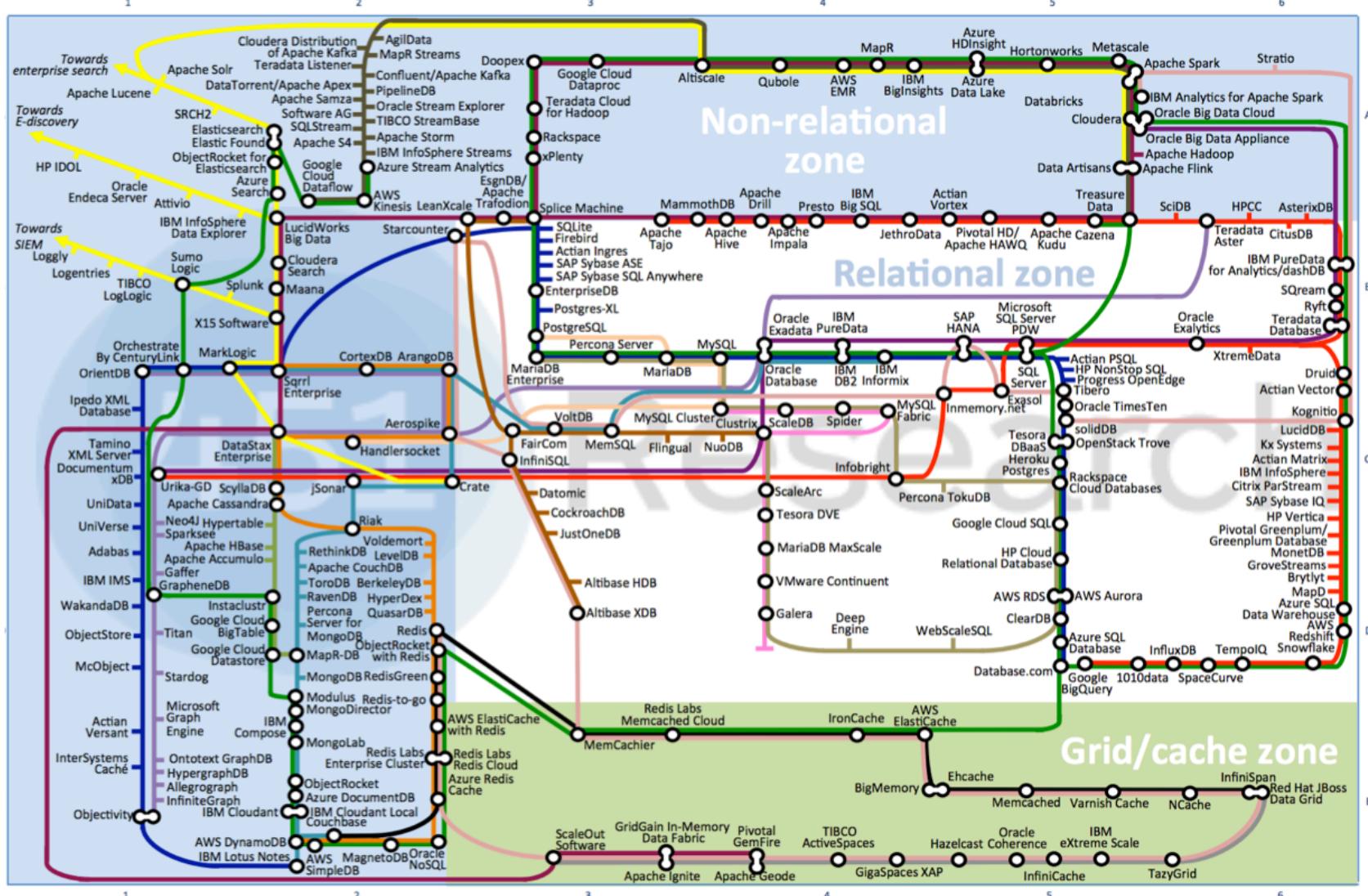
Modelo Bancos Dados Rede

Modelo Bancos Dados Hierárquico



Data Platforms Map

January 2016



[https://451research.com/
state-of-the-database-landscape](https://451research.com/state-of-the-database-landscape)

© 2016 by 451 Research LLC.
All rights reserved

Tecnologias de Bancos de Dados

Considerações Finais

Considerações Finais

- Nesta aula apresentamos um panorama geral das tecnologias de bancos dados, com foco na tecnologia relacional.
- Exercitamos consultas básicas em SQL e a criação de funções em PL/pgSQL.
- Discutimos a criação de novos tipos de dados e operações.

Referências Bibliográficas

Padrões e Especificações

- OGC. *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture*. Disponível em: <http://www.opengeospatial.org>. Acesso: Outubro de 2012.
- OGC. *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option*. Disponível em: <http://www.opengeospatial.org>. Acesso: Outubro de 2012.
- ISO. *SQL Multimedia and Application Packages – Part 3: Spatial*.

Artigos

- R. H. Güting. *Gral: an extensible relational database system for geometric applications*. In Proceedings of the 15th international conference on Very large data bases (VLDB '89). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989, pp. 33-44.
- Vijlbrief, T., and P. van Oosterom. *The GEO++ System: An Extensible GIS*. Proc. 5th Intl. Symposium on Spatial Data Handling, Charleston, South Carolina, 1992, pp. 40-50.
- Dangermond, J. *A Classification of Software Components Commonly Used in Geographic Information Systems*. In Proceedings of the U.S.-Australia Workshop on the Design and Implementation of Computer-Based Geographic Information Systems, Honolulu, HI, 1982, pp. 70–91.

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*, v. 1, n. 1, 1976, pp. 9-36.
- GRAY, J. *Evolution of Data Management*. *IEEE Computer*, v. 29, n. 10, 1996, pp. 38-46.