



Introdução aos Sistemas de Base de Dados

Base de Dados - 2024/25

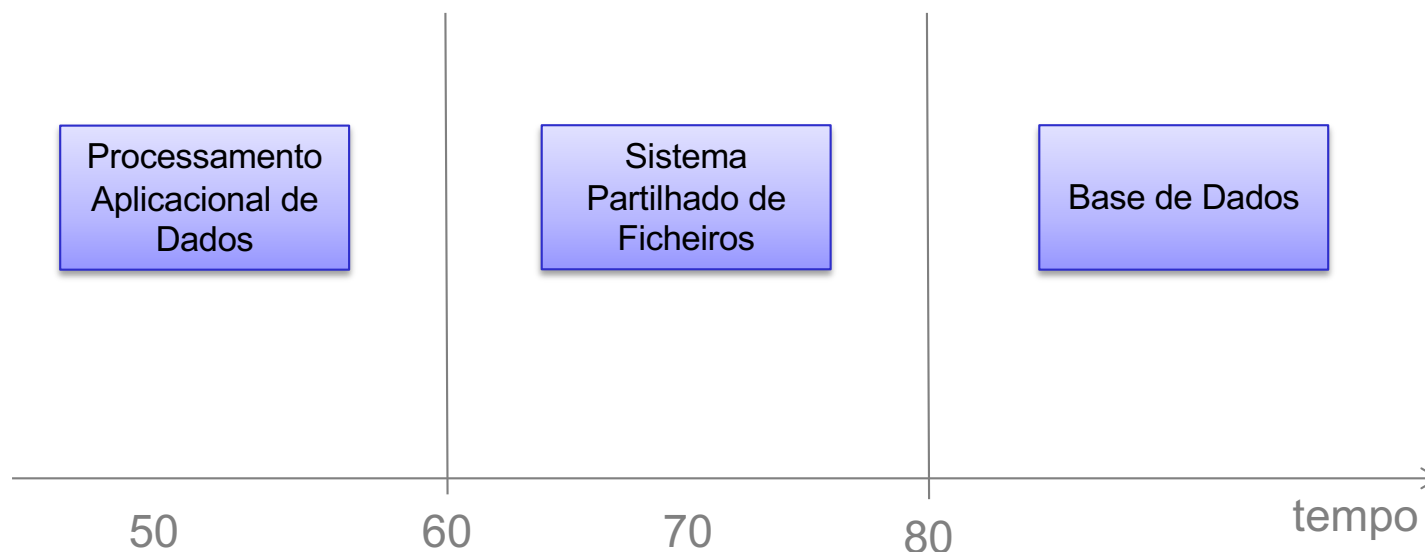
Carlos Costa

(Adapted from several DB courses and Books - see bibliography)

Base de Dados - Conceito

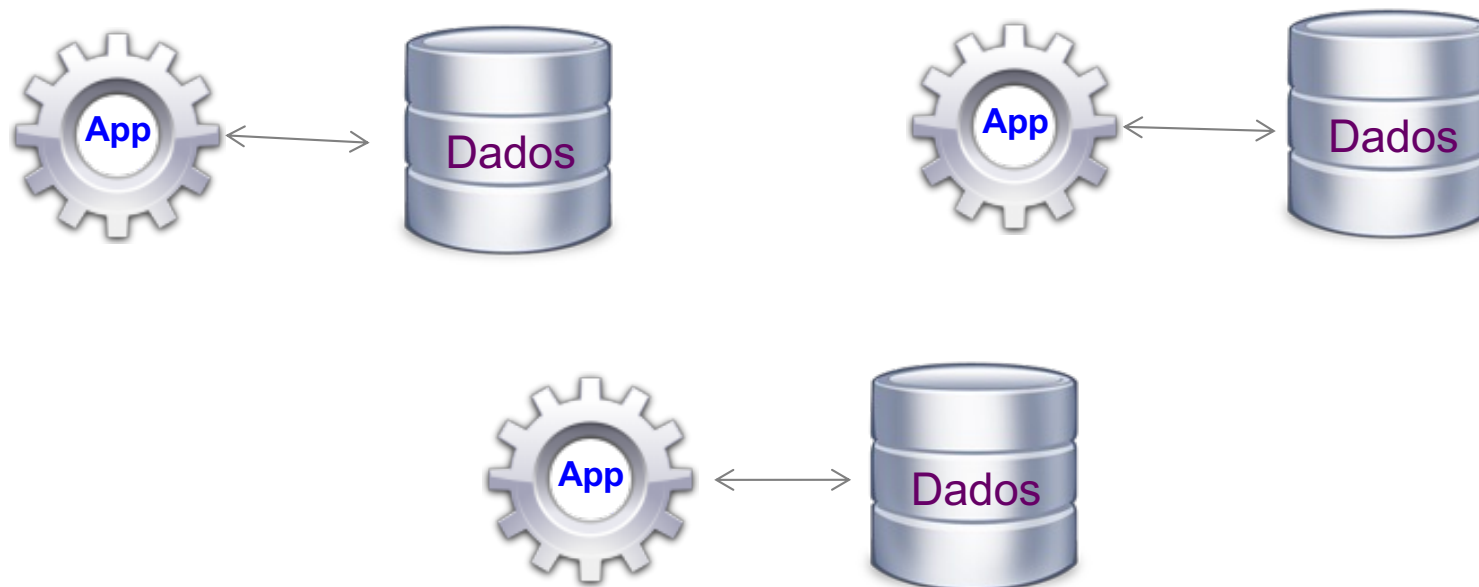
- **Base de Dados** (BD): uma coleção organizada de dados que estão relacionados e que podem ser partilhados por múltiplas aplicações.

Evolução



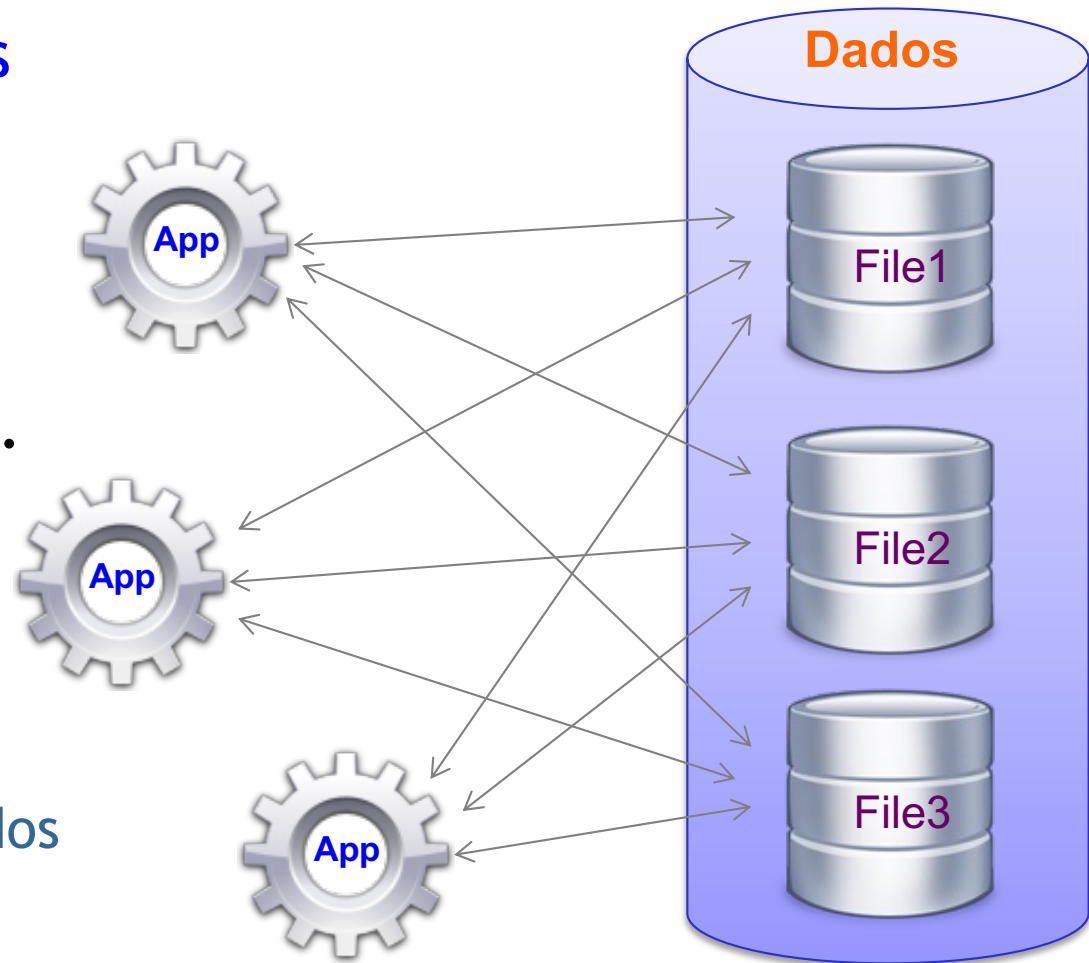
Processamento Isolado de Dados

- **Dados isolados** - cada aplicação gere os seus próprios dados.
- Os mesmos **dados** podem estar **replicados**.
- Diferentes **organizações** e **formatos** de dados.
- Problemas de “sincronismo” -> **incoerências**.

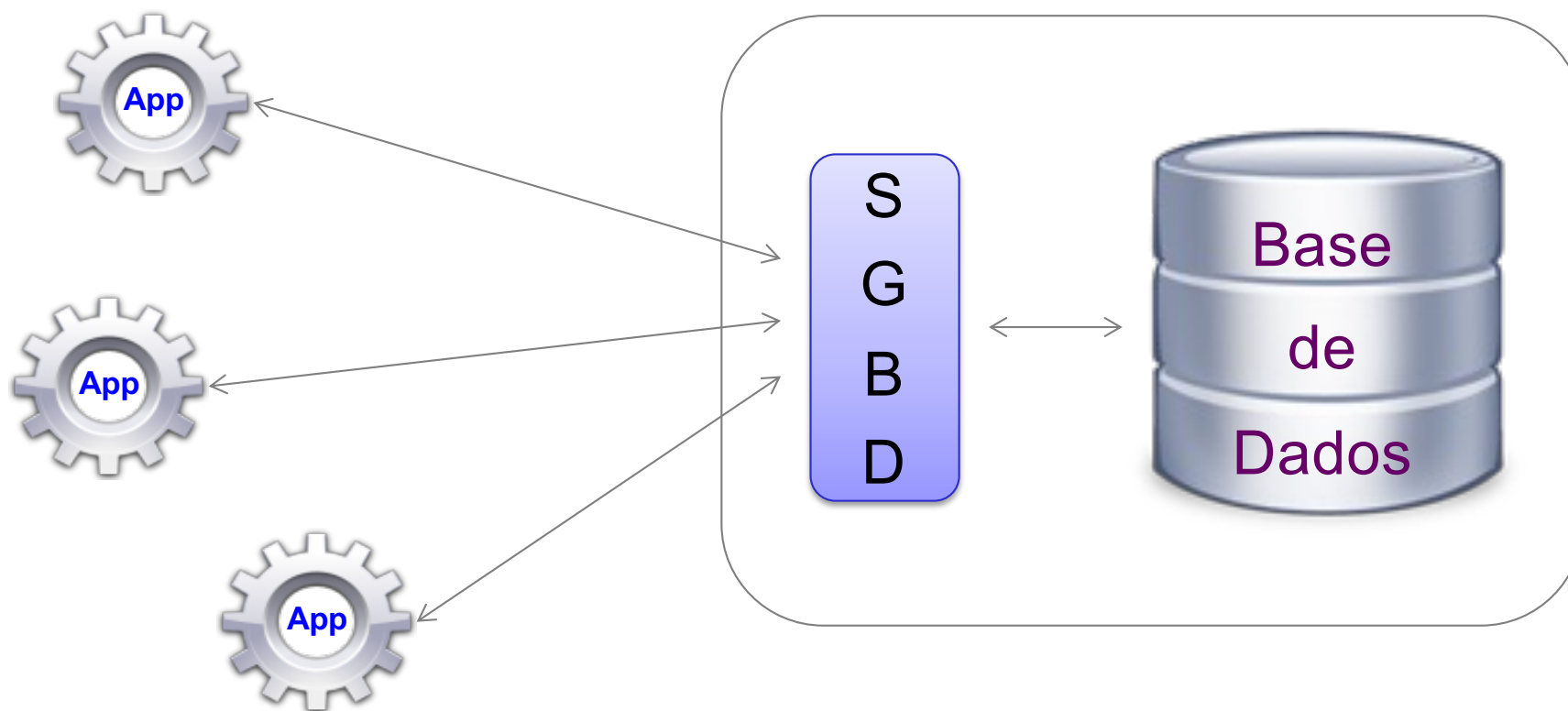


Sistema de Gestão de Ficheiros

- Dados organizados e armazenados em ficheiros **partilhados** por várias aplicações.
- Cada aplicação acede diretamente aos ficheiros.
- Cada aplicação usa uma **interface proprietária**.
- Problemas:
 - Acesso concorrente aos dados
 - Integridade
 - Segurança



Sistema de Gestão de Base de Dados (SGBD)



Database Management System (DBMS): “is a general-purpose software system that facilitates the processes of *defining*, *constructing*, *manipulating*, and *sharing* databases among various users and applications.”

Sistema de Gestão de Base de Dados (SGBD)

- Definição (*Defining*)
 - Especificação do tipo de dados, estruturas de dados e restrições
 - database catalog or dictionary
- Construção (*Constructing*)
 - Processo de armazenamento de dados
- Manipulação (*Manipulating*)
 - Envolve operações como a pesquisa e obtenção de dados
- Partilha (*Sharing*)
 - Acesso simultâneo aos dados por parte de vários utilizadores e programas

SGBD - Características Gerais

- Entidade única que opera com a BD
 - O acesso à BD é sempre mediado pelo SGDB
- Existe uma interface de acesso que esconde os detalhes de armazenamento físico dos dados
- Elevada abstração ao nível aplicativo
- Os dados estão integrados (nível lógico) numa mesma unidade de armazenamento
- Suporta uma ou mais BD
- Keyword - Data Independence

SGBD - Vantagens

- Independência entre programas e dados
- Integridade dos dados
 - Controlo de alteração de dados de acordo com as regras de integridade definidas
- Consistência dos dados
 - Nos processos de transações e mesmo em falhas de software/hardware
- Eficiência no acesso aos dados
 - Especialmente em cenários de manipulação de grandes quantidades de dados, por um ou mais utilizadores
- Isolamento utilizadores
 - Cada utilizador tem a “sensação” de ser o único

SGBD - Vantagens (cont.)

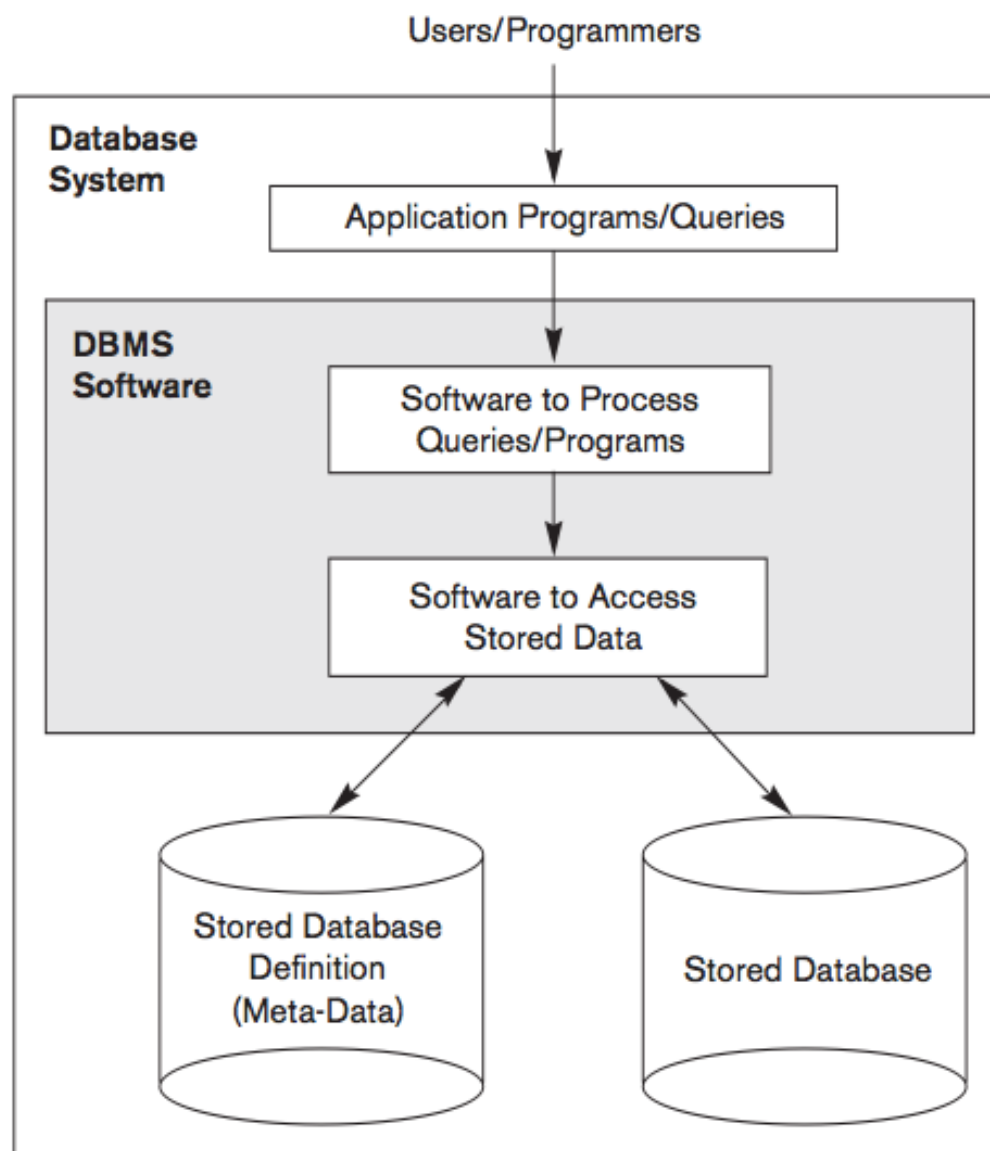
- Melhor gestão do acesso concorrencial
- Serviços de Segurança
 - Controlo de Acessos / Permissões
 - Codificação de Dados
- Mecanismos de backup e recuperação de dados
- Administração de dados
 - Disponibilidade de ferramentas desenvolvidas pelo fabricante e/ou terceiras entidades
- Linguagem de desenho e manipulação de dados

Nota: Muitas das vantagens anteriores são também requisitos funcionais de um SGBD.

SGBD - Potenciais Desvantagens

- Maiores custos e complexidade na instalação e manutenção
 - Especial em soluções empresariais
- Não respondem aos requisitos de alguns cenários aplicativos
 - Por exemplo, pesquisa de texto
 - Motivou o aparecimento de novos modelos (NoSQL, IndexEngine, etc)
- Centralização dos dados mais suscetível a problemas de tolerância a falhas (software e hardware) e de escalabilidade

SGBD - Vista Simplificada



SGBD - Utilizadores

- Utilizadores Finais
 - aqueles que usam o sistema com determinada finalidade com recurso a ferramentas disponibilizadas pelo fabricante do sistema ou aplicações de terceiras entidades.
- Programadores de Aplicações
 - Desenvolvem aplicações que permitem que os utilizadores interajam com a base de dados. Podem utilizar várias linguagem de programação.
- Administradores da Base de dados
 - Tratam dos processos de gestão e manutenção da base de dados.

SGBD - Metadados

- Metadados (dados sobre dados)
- O SGBD armazena uma descrição da própria estrutura da base de dados, restrições de integridade e condições de acesso.
 - Descritores de objetos da base de dados (tabelas, utilizadores, regras, vistas, indexes, etc)
 - Informação sobre dados em uso e por quem (locks).
 - *Schemas e mappings*

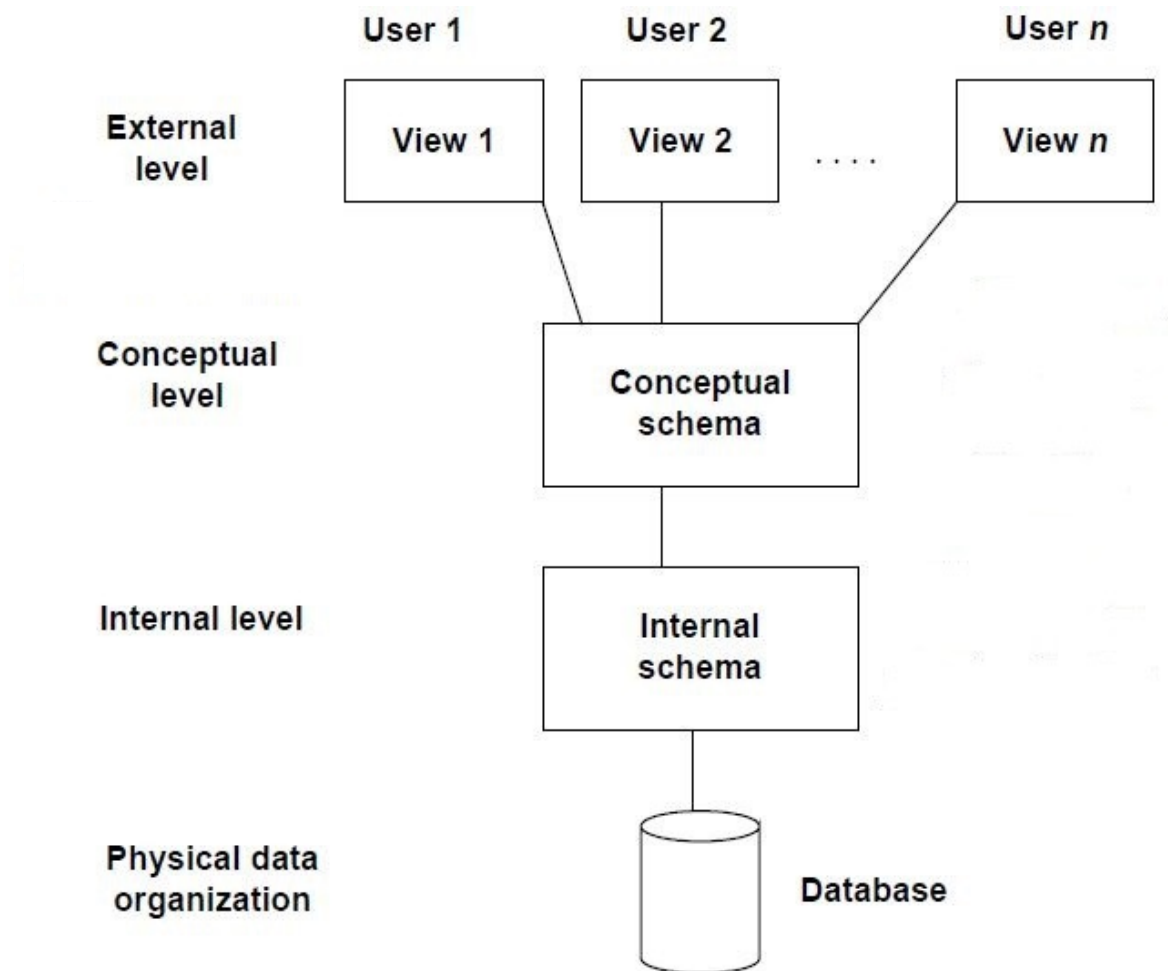
Interfaces (Aplicações)

- Web-based
- Form-based (desktop)
- GUI (Graphical User Interface)
 - Manipulação visual de esquemas de BD com recurso a diagramas. Possibilidade de construção e execução de queries.
- Natural Query Language
- DBMS Command Line
 - Criar contas de utilizadores, parametrizar o sistema, definir permissões e privilégios, definir/alterar estruturas de dados, definir tipos de dados, etc.
 - Utilizando uma linguagem própria - SQL

SGBD - Arquitetura ANSI/SPARC¹

Three-level architecture:

- External level
database users
- Conceptual level
database designers
and administrators
- Internal level
systems designers



ANSI/SPARC - Nível Interno

- Lida com a implementação física da BD
 - Estrutura dos registos em disco - files, pages, blocks
 - Indexes e ordenação dos registos
- Domínio: Programadores de sistemas de BD
- Exemplo de Esquema

```
RECORD FUNCIONARIO
  LENGTH=44
  HEADER: BYTE (5)
    OFFSET=0
  NOME: BYTE (25)
    OFFSET=5
  SALARIO: FULLWORD
    OFFSET=30
  DEPARTAMENTO: BYTE (10)
    OFFSET=34
```


ANSI/SPARC - Nível Conceptual

- Esquema Conceptual - descreve a estrutura da base de dados para os utilizadores
 - Descreve entidades, tipo de dados, relações, operações, restrições, etc
 - Utiliza (tipicamente) um modelo de dados para descrição do esquema conceptual
- Oculta detalhes de implementação física(abstração)
- Domínio: Administrador BD e prog. de aplicações
- Exemplo de esquema

```
CREATE TABLE FUNCIONARIO  
  (Nome VARCHAR(25) ,  
   Salario REAL, Dept_Nome VARCHAR(10))
```

ANSI/SPARC - Nível Externo

- Oferece vistas da base de dados adaptadas a casa utilizador
 - Apresentação dos dados pode ser trabalhada, parte dos dados pode ser ocultada, etc.
- Domínio: Utilizadores finais e prog. de aplicações
- Exemplo de Esquema

FolhaPagamentos:

char *Nome
double Salario

Funcionarios:

char *Nome
char *Departamento

ANSI/SPARC - Independência dos dados

- A alteração do esquema (*schema*) de um nível não tem impacto no esquema do nível acima.

=> Dois níveis de independência

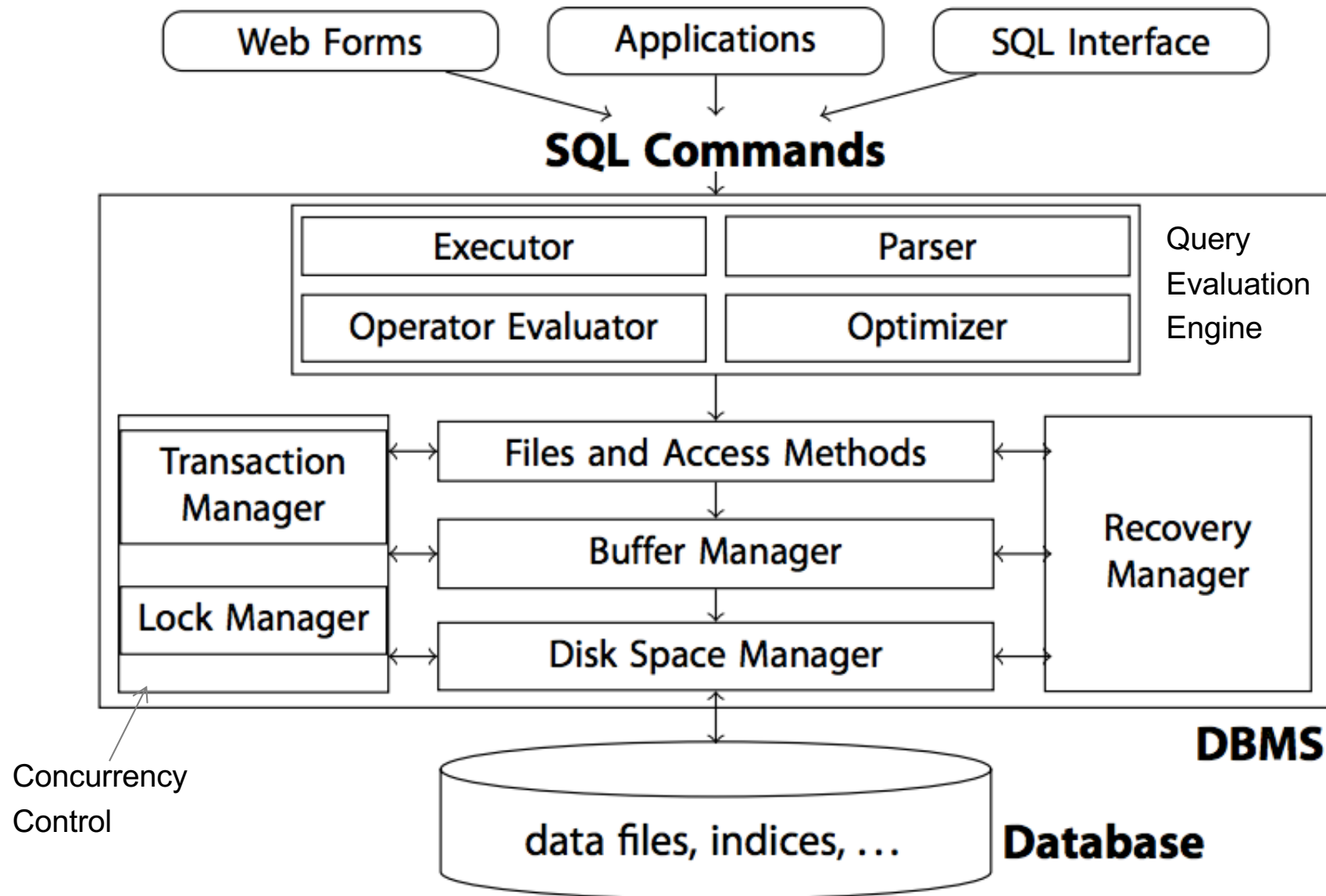
■ Nível Físico

- Alterações do nível físico não devem ter impacto no esquema conceptual.
- Por exemplo, podemos alterar a forma como armazenamos os dados no sistema de ficheiros por razões de desempenho.

■ Nível Lógico

- Alterações no esquema conceptual (modelo de dados) não devem repercutir-se nos esquemas externos ou aplicações já desenvolvidas.

SGBD - Arquitetura Típica



Modelo de Base de Dados

- Modelo de BD - coleção de conceitos para descrição lógica de dados (Modelo Lógico)
- **Esquema (Schema)**: a descrição de um conjunto particular de dados com recurso a um determinado modelo
- Um bom modelo de dados é fundamental para garantir a independência dos dados
- O **Modelo Relacional** é um dos mais utilizados nos dias de hoje.
 - Bancos, Hospitais, Finanças, Seguradoras, etc

Modelos de Base de Dados

- 1ª Geração (Pré-relacional)

- Hierárquico
- Rede

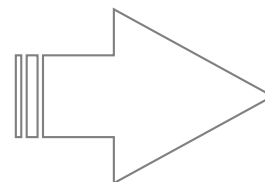
- 2ª Geração

- Relacional

- 3ª Geração (Pós-relacional)

- Object-relational
- Object-oriented
- Key-value store
- Document-oriented
- Column-oriented
- Graph database

...

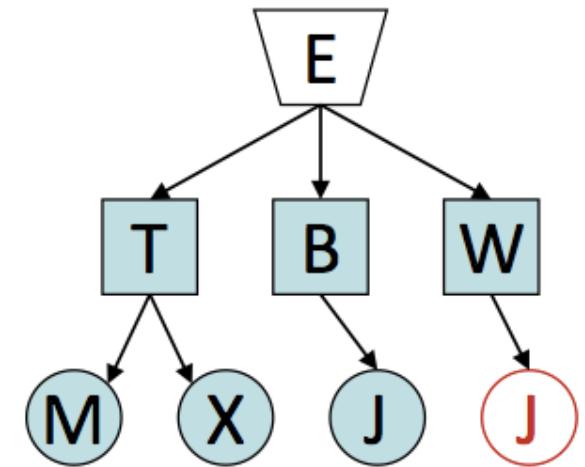


Disciplina
de
Base de Dados

Modelos NoSQL

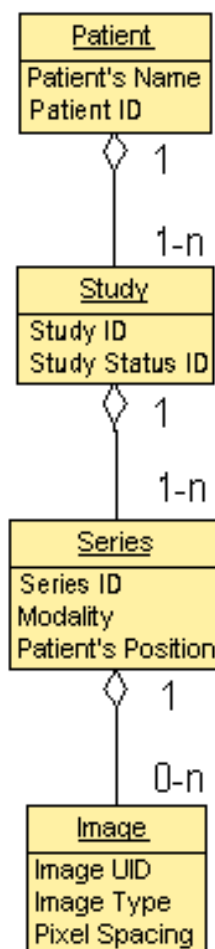
Phases	Targets		
Conceptual Design	Conceptual Data Model	ERD(Entity Relationship Diagram) UML(Unified Modeling Language) ORM(Object Role Modeling) FCO-IM(Fully Communication Oriented Information Modeling)	
Logical Design	NoSQL Data Model	Key-Value , Document, Column Family , Graph	
Physical Design	NoSQL Database	Key-Value	Riak, Redis, Memcached ,Berkeley DB ,Hamster DB, Amazon Dynamo DB ,Project Voldemort
		Document	MongoDB, Couch DB, Terrastore, Orient DB, Raven DB
		Column Family	Cassandra, HBase, Hypertable, Amazon Simple DB
		Graph	Neo4J, Infinite Graph, Orient DB, Flock DB

Modelo Hierárquico

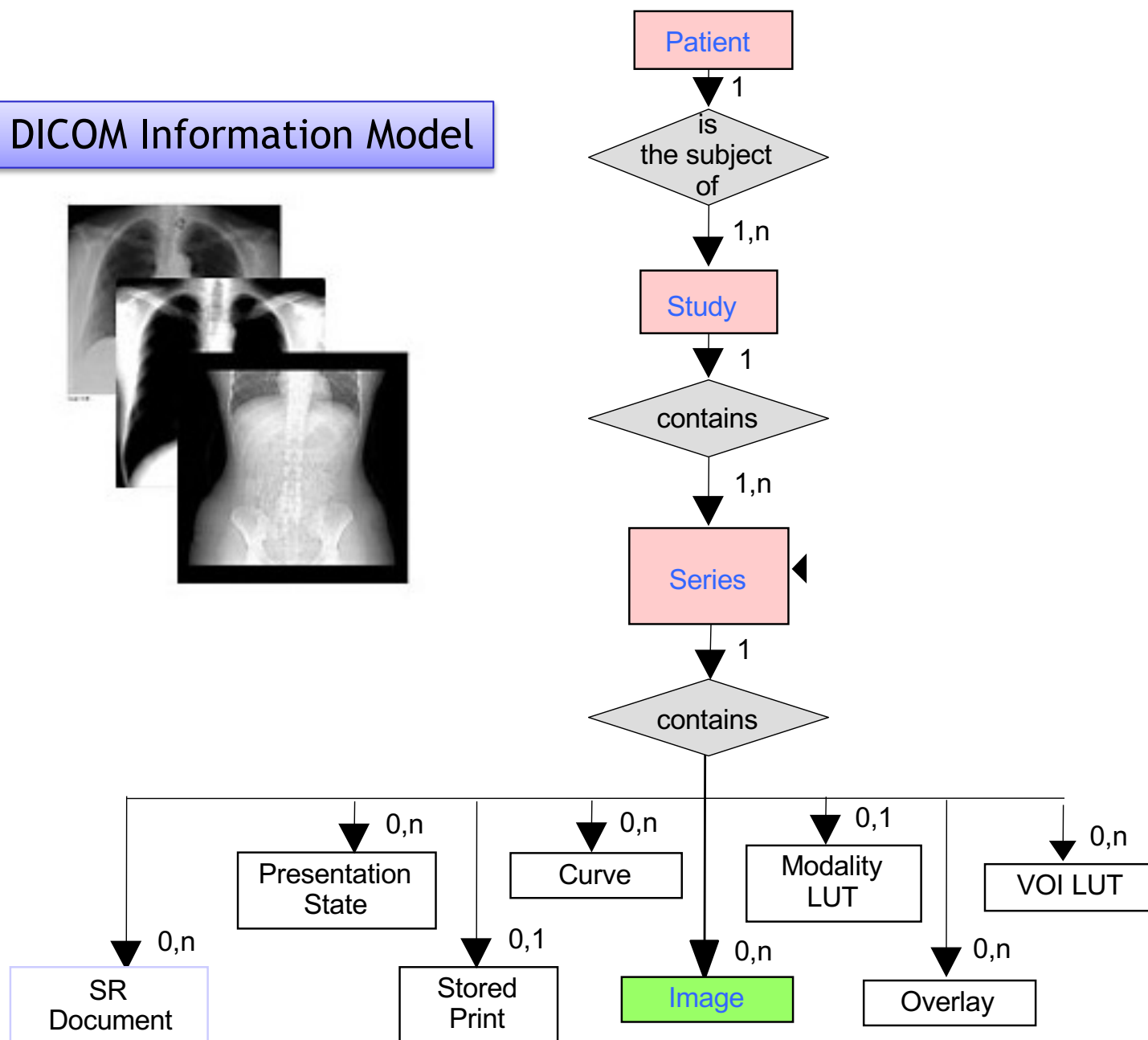


- Dados estão armazenados numa **estrutura** hierárquica (**árvore**).
- Os nós da árvore designa-se como **registos** que estão ligados por **ponteiros** (*links*).
- Um registo é composto por um conjunto de atributos.
- Um link é uma associação entre dois registos do tipo **pai-filho**.
- Um registo pai encontra-se associado a N registos filhos (**1:N**).

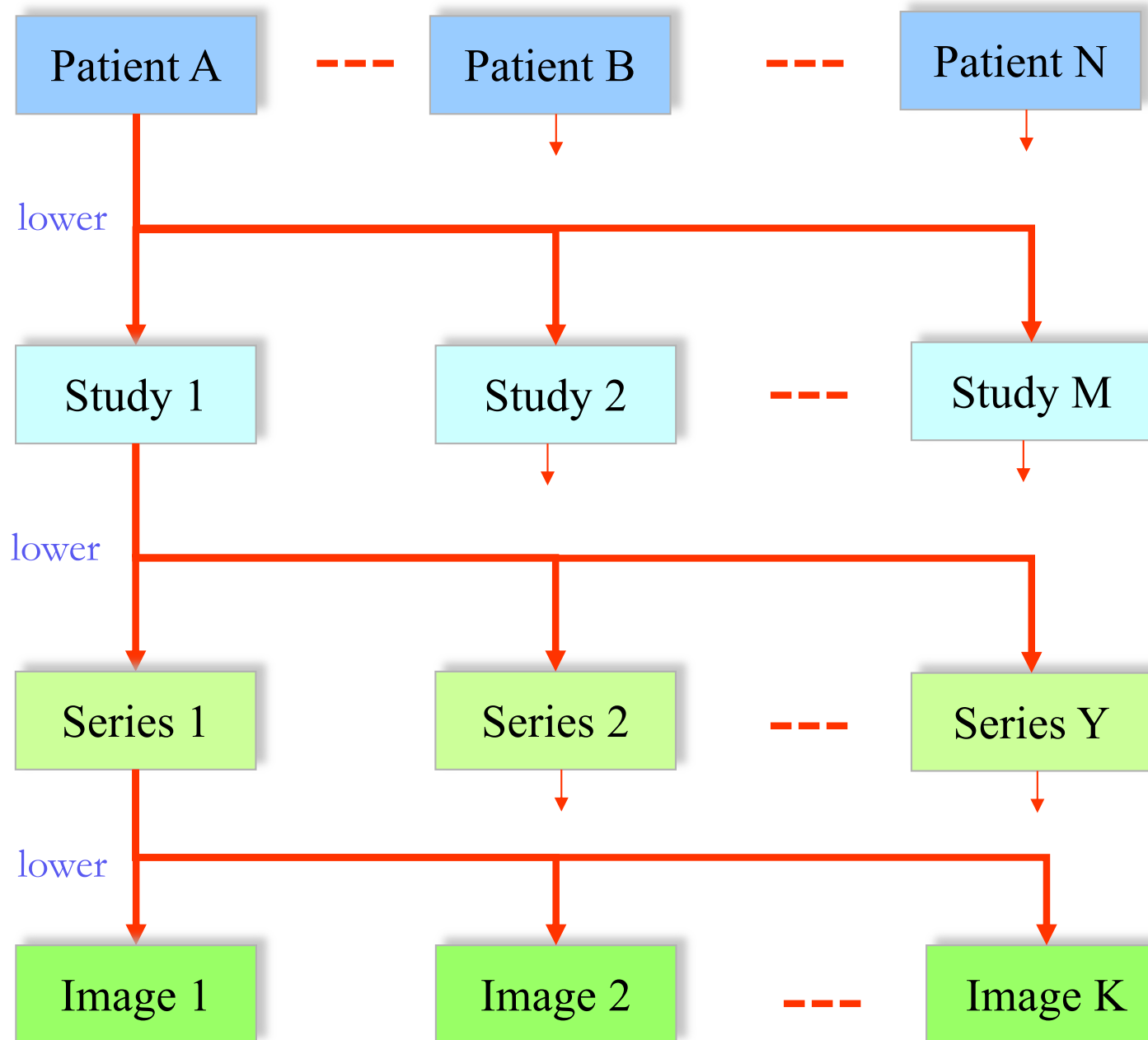
Modelo Hierárquico - Caso Real (2023)



DICOM Information Model



Modelo Hierárquico - Caso Real (2023)

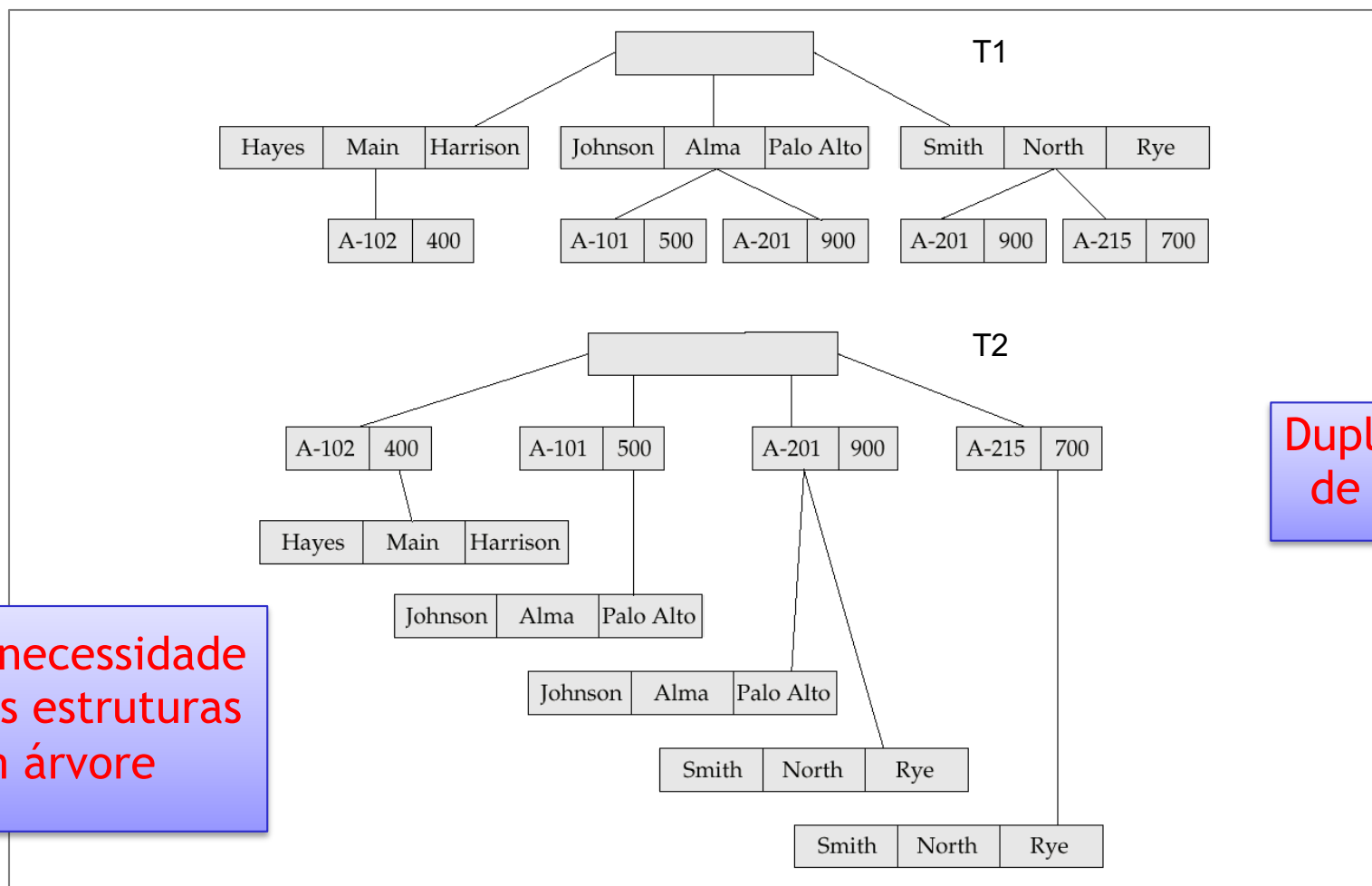
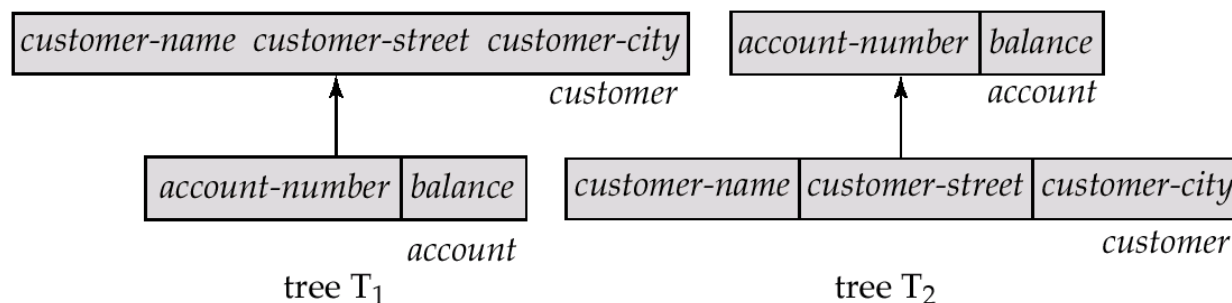


Modelo Hierárquico - (Des)vantagens

- Adaptado a cenários de acesso sequencial aos dados.
 - Qualquer acesso aos dados passa sempre pelo segmento raiz.
 - A maior parte das necessidades atuais requer acesso aleatório!
- Redundância de informação
 - Desperdício de espaço e inconsistências de dados
- Restrições de integridade, exemplo:
 - A eliminação de um segmento pai, implica a remoção de todos os segmentos filhos associados.
- Não permite estabelecer associações N:M

Modelo Hierárquico - Relação N:M

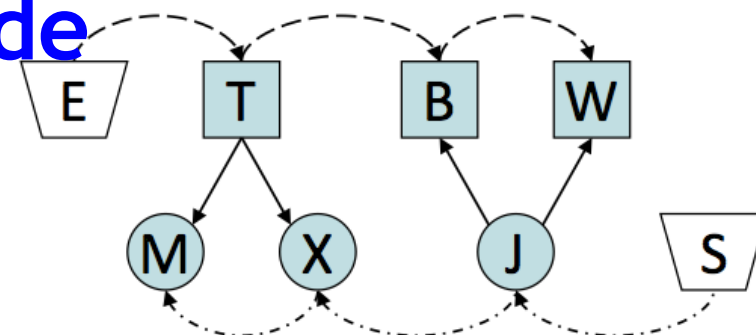
Uma conta pode ter vários titulares (clientes).
Um cliente pode ter várias contas.



Temos necessidade de duas estruturas em árvore

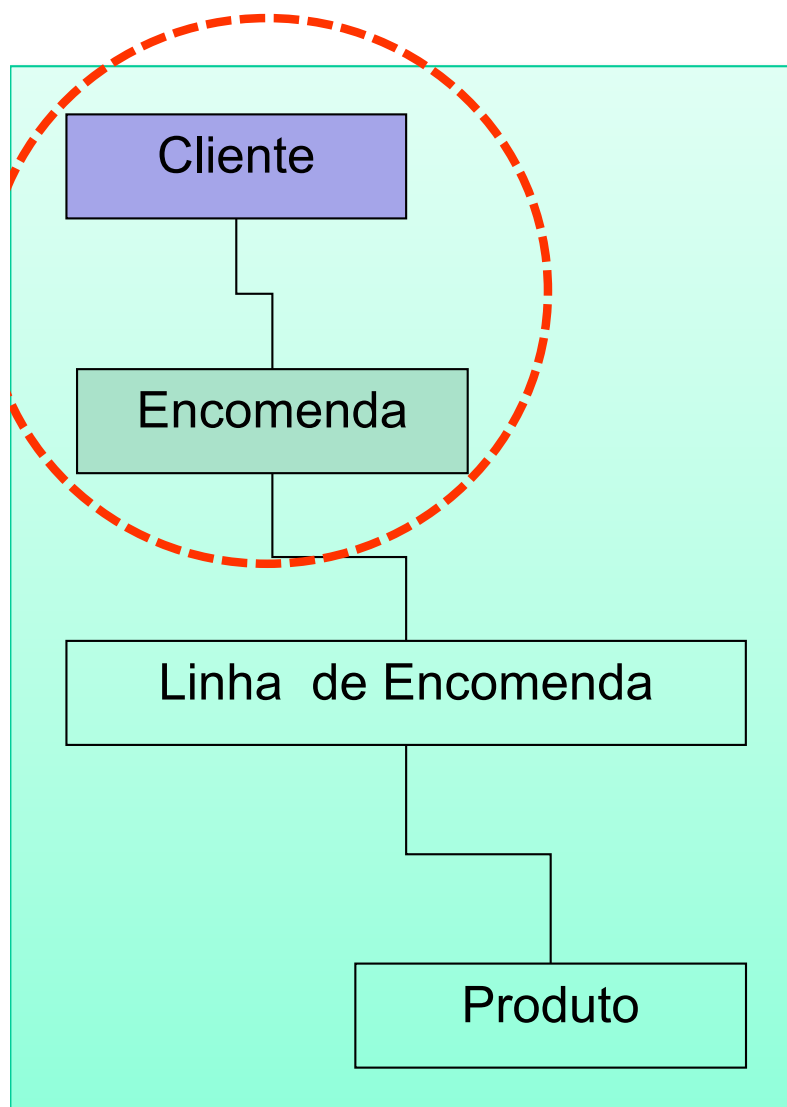
Duplicação de Dados

Modelo de Rede

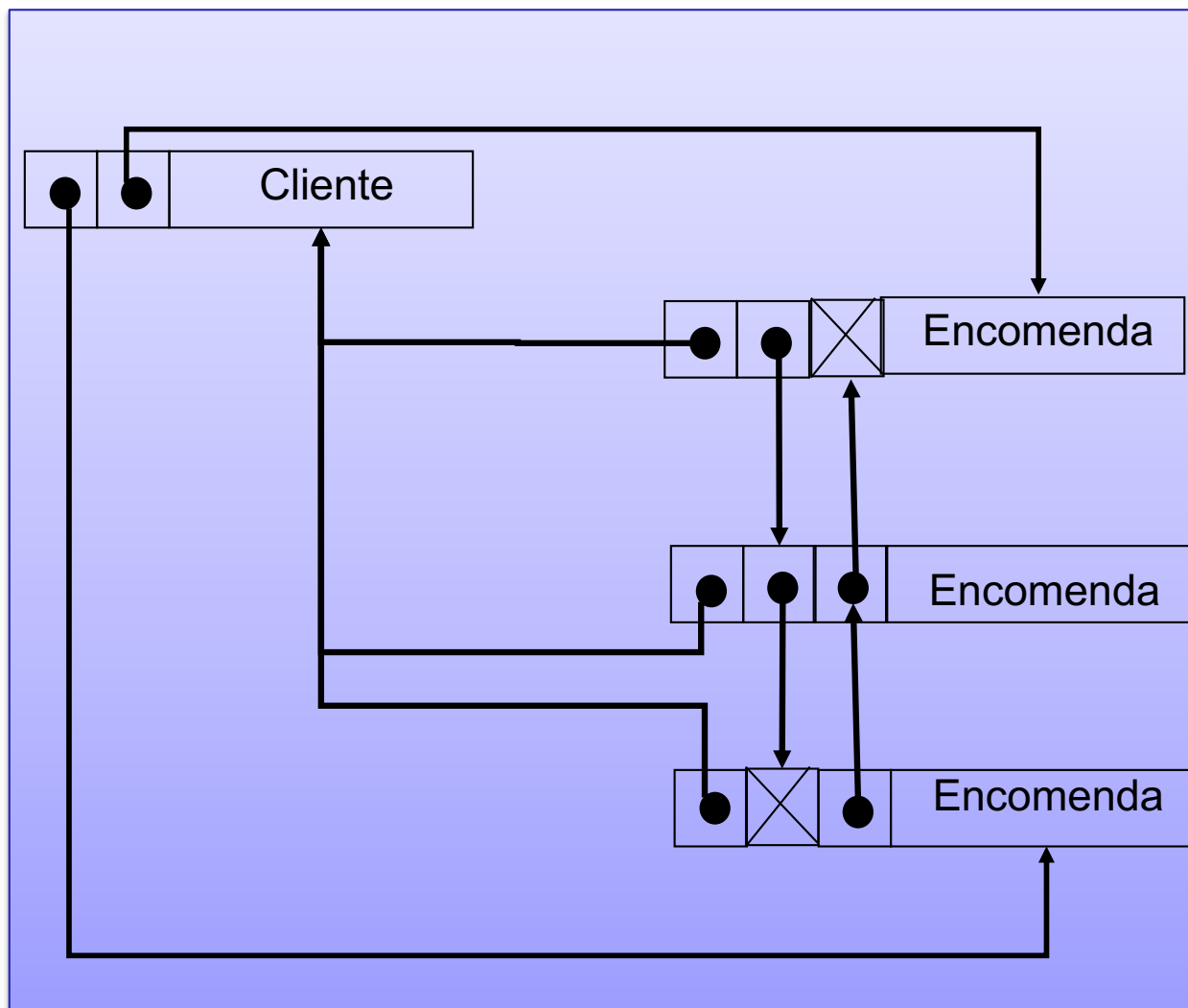


- Extensão do modelo hierárquico.
- Permite que um mesmo registo esteja envolvido em várias associações -> visão de rede.
- Melhorias na capacidade de navegação na estrutura de dados.
- Relações representadas através de grafos.
- Um conjunto (*set*) suporta associação entre registos do mesmo tipo
 - Tipicamente implementados com listas ligadas circulares
- Relacionamento 1:N entre dois tipos de registo.

Modelo Hierárquico -> Rede

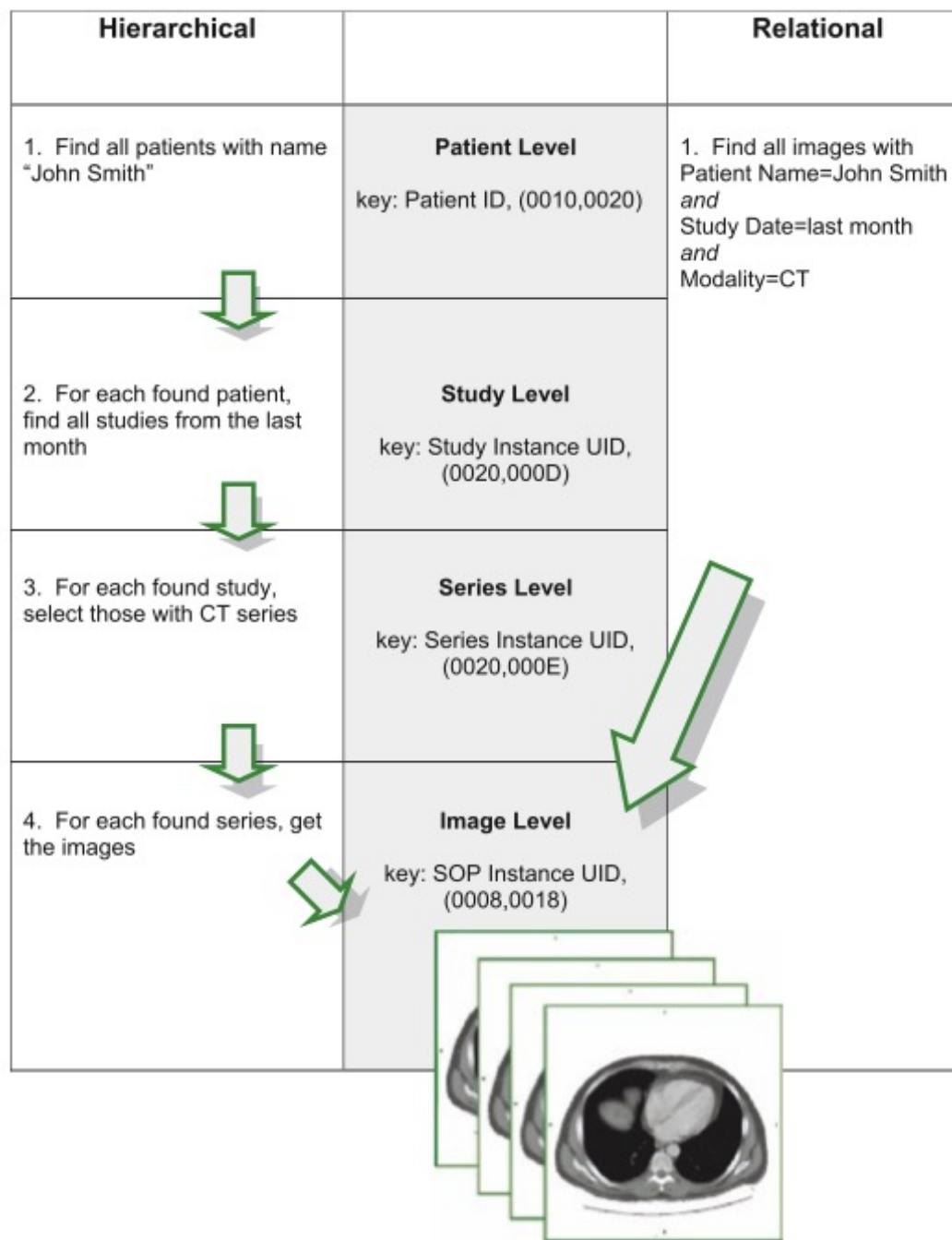


Hierárquico



Rede

Modelo Hierárquico versus Relacional

























SGBD

ALGUMAS CURIOSIDADES...

DB-Engines Ranking - Engine

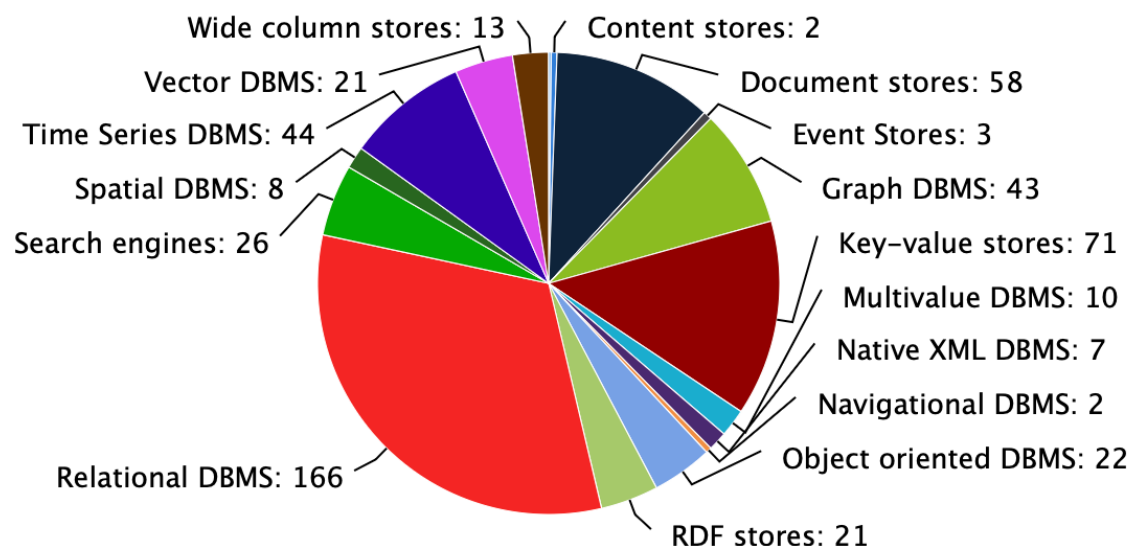
424 systems in ranking, February 2025

Rank			DBMS	Database Model	Score		
Feb 2025	Jan 2025	Feb 2024			Feb 2025	Jan 2025	Feb 2024
1.	1.	1.	Oracle	Relational, Multi-model 	1254.82	-3.93	+13.38
2.	2.	2.	MySQL 	Relational, Multi-model 	999.99	+1.84	-106.67
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model 	786.87	-11.69	-66.70
4.	4.	4.	PostgreSQL 	Relational, Multi-model 	659.62	-3.79	+30.21
5.	5.	5.	MongoDB 	Document, Multi-model 	396.63	-5.87	-23.73
6.	 7.	6.	Redis 	Key-value, Multi-model 	157.91	+4.55	-2.80
7.	 6.	 9.	Snowflake 	Relational	155.58	+1.68	+28.13
8.	8.	 7.	Elasticsearch	Multi-model 	134.63	-0.29	-1.11
9.	9.	 8.	IBM Db2	Relational, Multi-model 	125.44	+2.46	-6.79
10.	10.	10.	SQLite	Relational	113.82	+7.13	-3.47
11.	11.	 12.	Apache Cassandra 	Wide column, Multi-model 	102.58	+3.39	-6.69
12.	12.	 11.	Microsoft Access	Relational	96.54	+3.84	-16.63

Ranks database management systems according to their popularity.

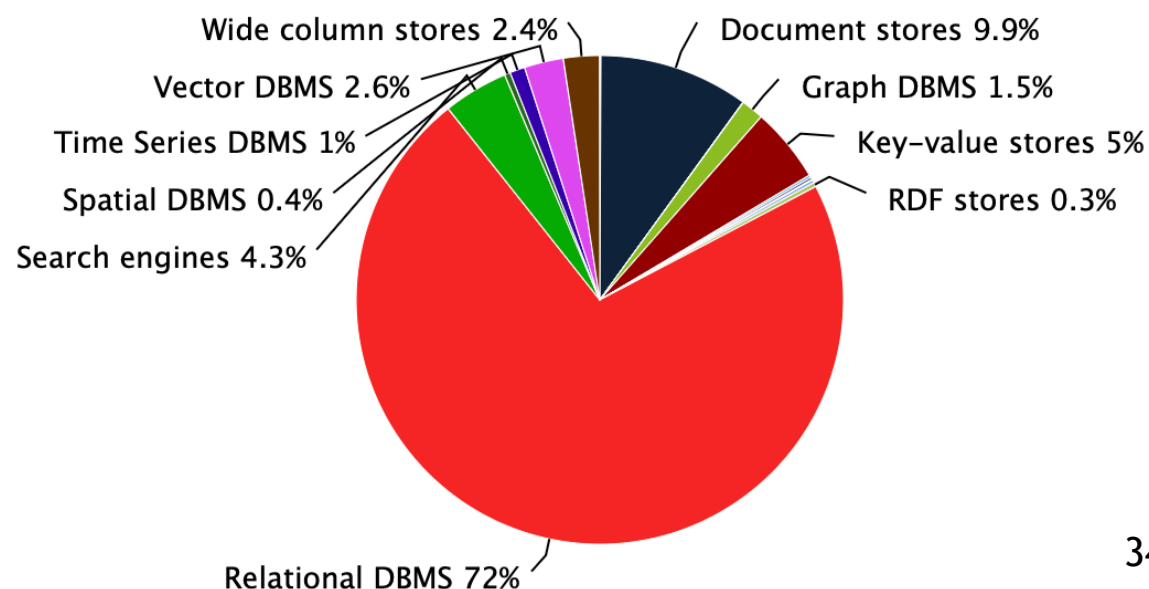
Source: DB-Engines Ranking (02/2025)

DB-Engines Ranking - Model



number systems per category

scores per category in percent



Resumo

- Introdução aos Sistemas de Base de Dados
- Sistemas Gestores de Base de Dados
- Modelos de Base de Dados