

Cognitive Data Science

PROF. DR. FERNANDO T. FERNANDES

FIAP

Agenda

- SGBD
- Modelo Relacional
- Propriedades de um SGBD
- Transações e suas propriedades

Sistemas Gerenciadores de Banco de Dados (SGBDs)

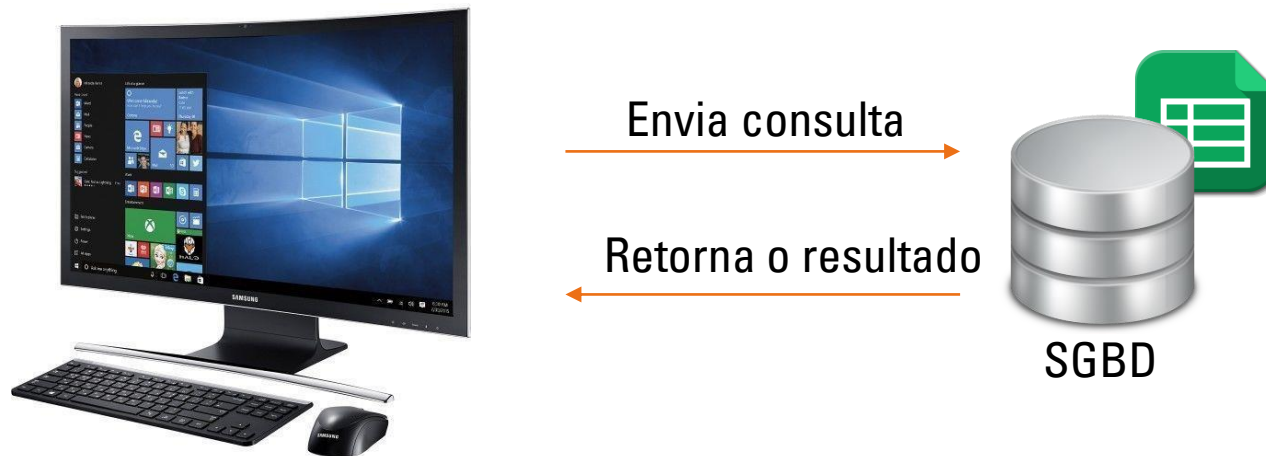
- *Database Management System* (DBMS)
- Sistema computadorizado que permite *definir, construir, manipular* e *compartilhar* bancos de dados.

Sistemas Gerenciadores de Banco de Dados (SGBDs)

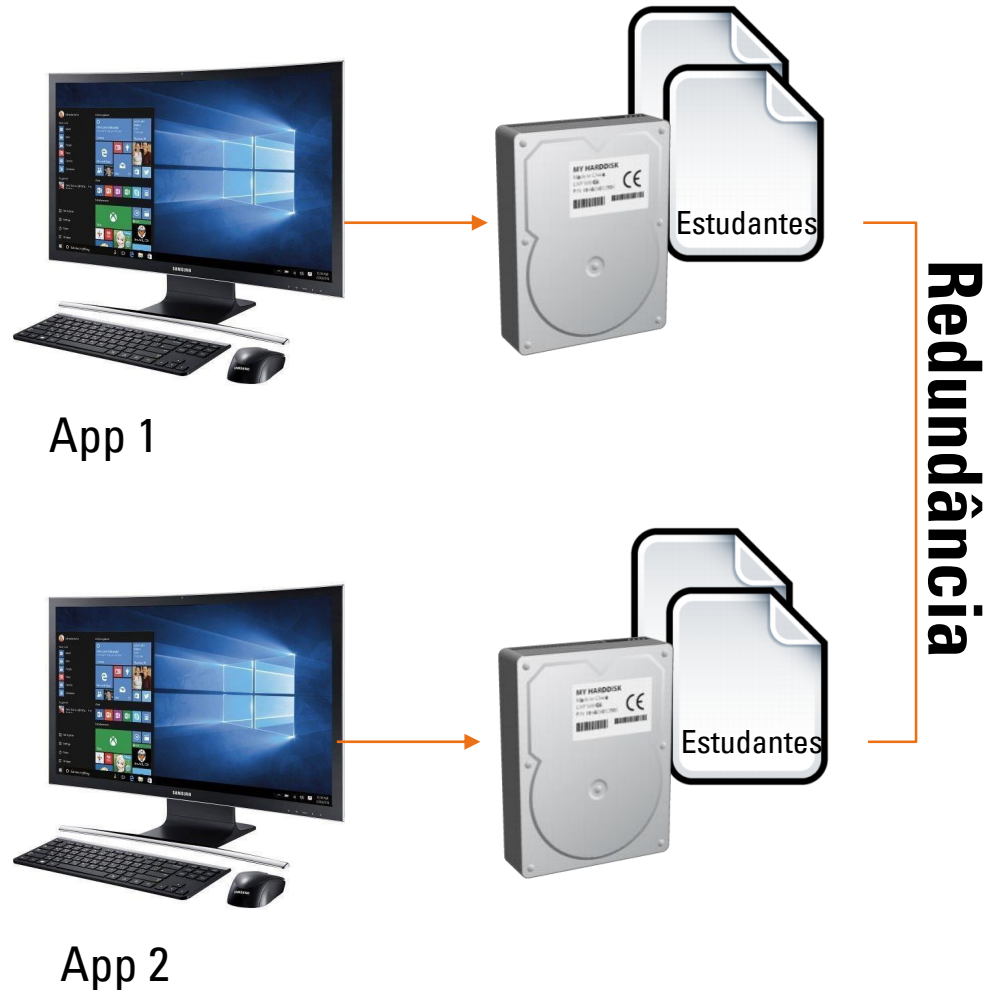
- **Definir** - Definir os tipos de dados, estrutura e restrições aos dados.
- **Construir** - Armazenar os dados em algum meio controlado pelo SGBD.
- **Manipular** - Pesquisar, incluir, excluir, atualizar e gerar relatórios.
- **Compartilhar** – Permitir acesso a múltiplos usuários e aplicações simultaneamente.

Acesso a SGBDs

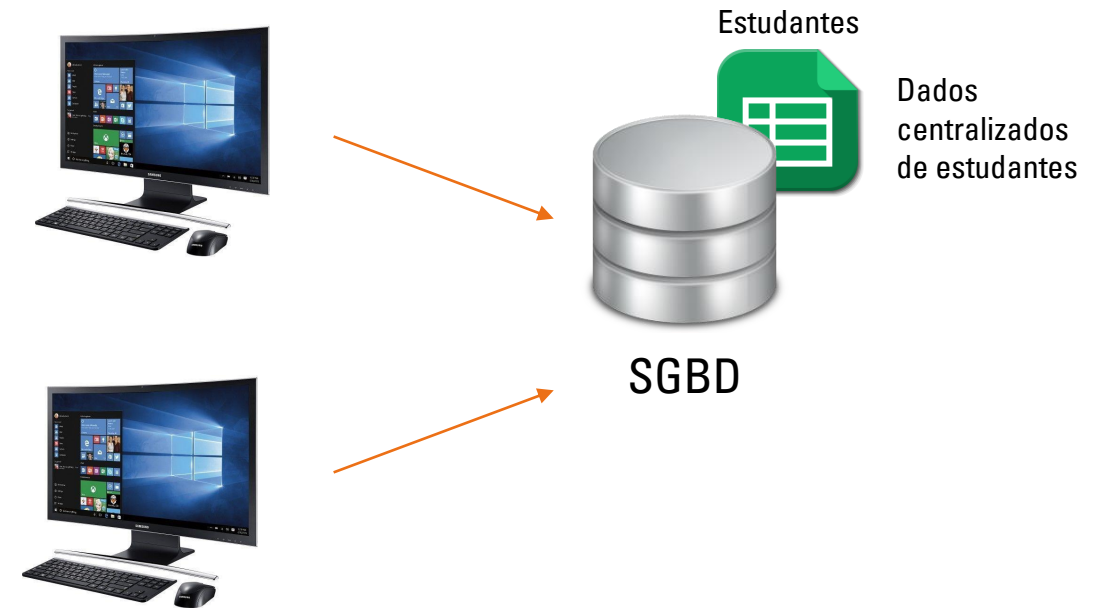
- Uma aplicação pode acessar uma base de dados, realizando consultas (*queries*) ou transações (*transactions*) que podem inserir novos dados.



Sistemas Tradicionais vs SGBDs



Diminuem a redundância, economizando armazenamento



FIAP

Sistemas Gerenciadores de Banco de Dados (SGBDs)

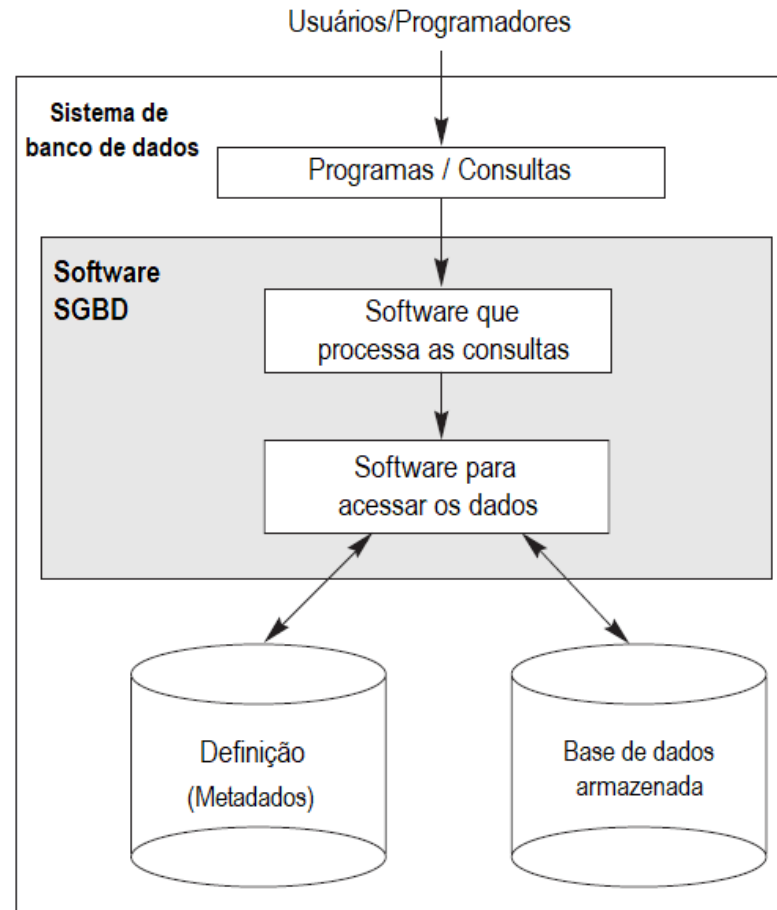


Fig.1 – Ambiente simplificado de um sistema de banco de dados

Fonte: ELMASRI e NAVATHE (2018, p.38)

Modelo Relacional – Proposto por Edgar “Ted” Codd

- Dados organizados em Tabelas
- Sem ponteiros
- Tabelas conectadas por campos relacionados (IDs)

Information Retrieval

P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, joins, retrieval language, predicate calculus, security, data integrity

CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

1. Relational Model and Normal Form

1.1. INTRODUCTION

This paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childs [1], the principal application of relations to data systems has been to deductive question-answering systems. Levin and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of *data independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of *data inconsistency* which are expected to become troublesome even in nondeductive systems.

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the “connection trap”).

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. DATA DEPENDENCIES IN PRESENT SYSTEMS

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed *without logically impacting some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependence, indexing dependence, and access path dependence. In some systems these dependencies are not clearly separable from one another.

1.2.1. Ordering Dependence. Elements of data in a data bank may be stored in a variety of ways, some involving no concern for ordering, some permitting each element to participate in one ordering only, others permitting each element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the

Volume 13 / Number 6 / June, 1970

Communications of the ACM 377

Propriedades de um SGBD

- Auto-descritivo

- Uso de **metadados** para descrever os dados armazenados
- Estrutura dos arquivos, tipos de dados, armazenamento, restrições (**catálogo**)

- Ex:

RELAÇÕES

NOME_RELACAO	N_COLUNAS
ESTUDANTE	3
CURSO	2
NOTAS	3

COLUNAS

NOME_COLUNA	TIPO_DADO	PERTENCE_A
Nome	Character (30)	ESTUDANTE
matricula	Integer (1)	ESTUDANTE
Semestre	Integer (1)	ESTUDANTE
Nome_curso	Character (10)	CURSO
sigla_curso	Character (10)	CURSO

Propriedades de um SGBD

- Múltiplas visualizações
 - Cada aplicação pode requerer informações diferentes. (ex: visão da secretaria VS inscrições)

BOLETINS

NOME	DESEMPENHO ALUNOS			
	NOME_DISCIPLINA	NOTA	SEMESTRE	ANO
JOÃO	CD1001	6	1	1
	PG1001	7	1	1
JOSÉ	CD1001	10	1	1
	PG1001	10	1	1
	CD2001	9	2	2
	PG2001	10	2	2

PRE_REQS

NOME_DISCIPLINA	SIGLA_DISCIPLINA	PRE REQ
COG. DATA SCIENCE	CD2001	CD1001
PLAT. COGNITIVAS	PG2001	PG1001

Propriedades de um SGBD

- **Isolamento entre aplicações e dados**

- Nas aplicações tradicionais baseadas em arquivos, a estrutura de arquivos de dados está embutida na aplicação. Qualquer alteração nesta estrutura, requer recodificar a aplicação.

- Ex: Adicionar um campo “Data de Nascimento” pode “quebrar” a aplicação, pois a aplicação não saberá onde ele começa e termina.

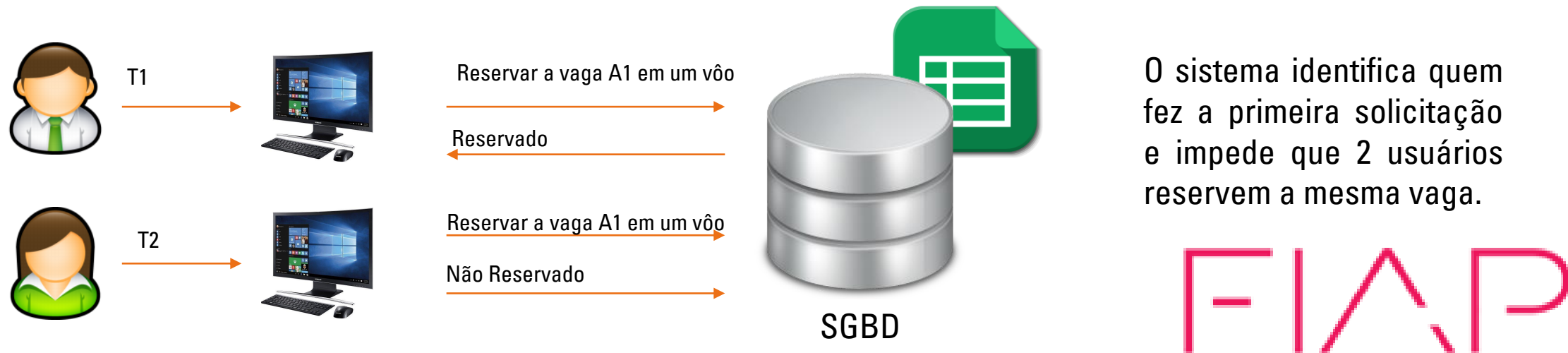
- Nos sistemas que usam SGBD, é possível adicionar campos, mudando a descrição da estrutura de uma tabela (metadados).

- Ex: Após adicionar o campo “Data de Nascimento”, o sistema lê a estrutura do catálogo e saberá como ler o novo campo.

Propriedades de um SGBD

○ Muti-usuário

- Os usuários podem acessar ao mesmo tempo o SGBD
- O SGBD deve possuir algum controle de concorrência que permite realizar *transações* no mesmo dado de forma sequencial.
- Este tipo de controle é chamado de processamento de dados Online (OLTP)



Propriedades de uma transação - ACID

- **Atomicidade** – Todas as operações em um dado em um banco de dados são executadas completamente ou não são executadas.
 - Ex: Realizo a inclusão de um registro completo (Commit) ou não insiro nada (Rollback).
- **Consistência** – A execução de uma transação deve levar de um estado consistente a um outro estado consistente.
 - Ex: Apenas dados válidos são salvos.
- **Isolamento** – Cada transação é executada isoladamente da outra, mesmo quando há concorrência. Uma transação não interfere na outra
- **Durabilidade** – Após as transações, os dados são persistentes, mesmo em caso de falhas.
 - Recuperabilidade de dados (ex: logs)

FIAP

Vantagens de um SGBD

- Evitar redundâncias — economiza espaço em disco
 - Normalização de dados
- Controle de acesso
- Persistência de dados
- Melhor performance de buscas
- Backup e recuperação de dados
- Múltiplos usuários
- Regras (*constraints*) - ex: somente 1 número de matrícula por aluno

Modelos de Dados

- **Modelo de dados relacional:** O mais famoso e mais utilizado dos modelos.
 - Representa um banco de dados como uma coleção de **tabelas**. Cada tabela pode ser armazenada em um arquivo diferente.
 - Utilizado nos SGBDs relacionais - Relational Database Management Systems (RDBMS)
- Outros modelos de dados
 - Modelo de dados Hierárquico (Usado entre 1965 e 1985)
 - Modelo de dados de Objeto
 - Modelo de dados XML
 - Modelo de em rede
 - Modelos de dados para Big Data
 - Modelo de dados de documentos (JSON), chave-valor, grafos e baseado em colunas

Principais SGBDs Relacionais

ORACLE®







Principais SGBDs Relacionais

System Properties Comparison MariaDB vs. Microsoft SQL Server vs. MySQL vs. PostgreSQL

Please select [another system](#) to include it in the comparison.

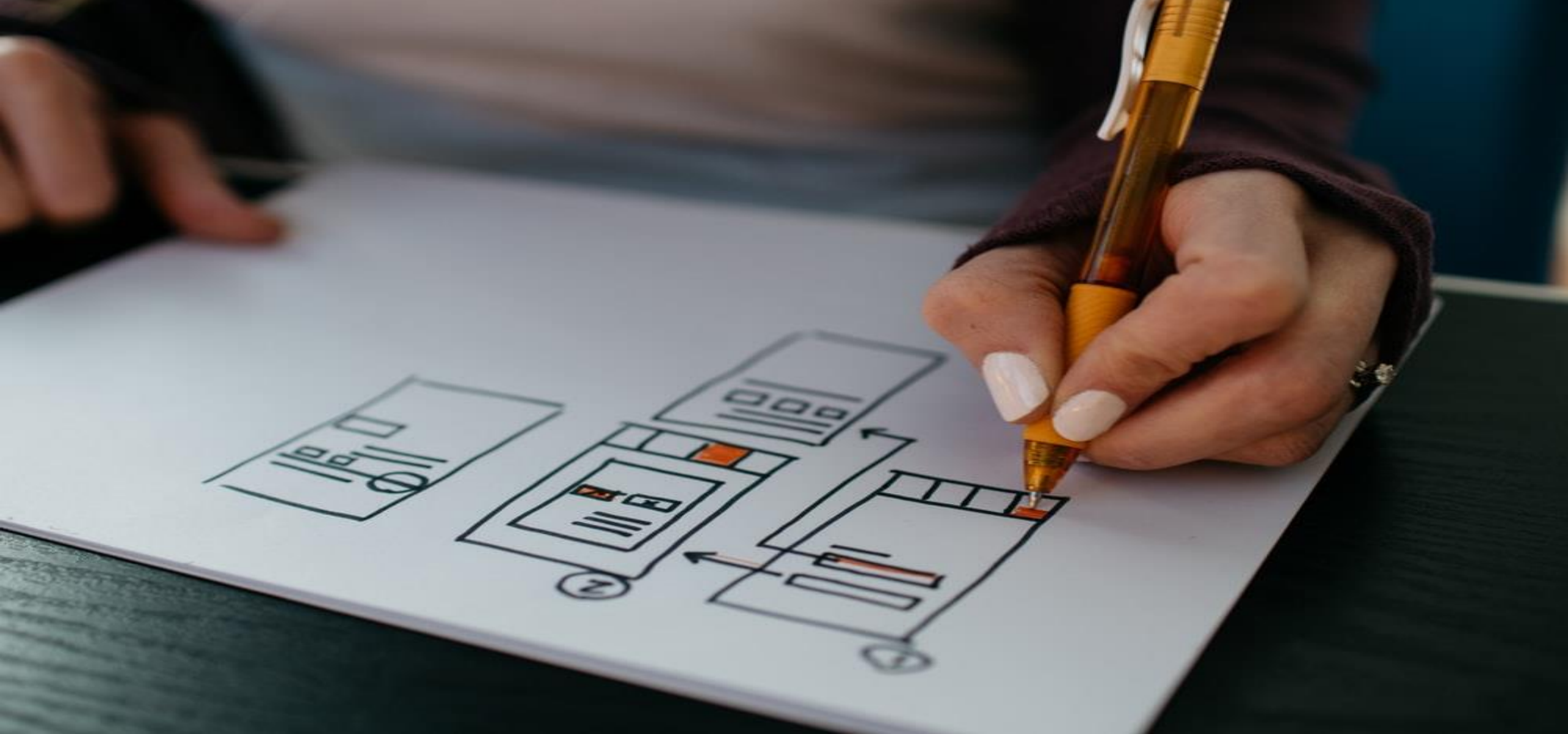
Editorial information provided by DB-Engines

Name	MariaDB X	Microsoft SQL Server X	MySQL X	PostgreSQL X
Description	MySQL application compatible open source RDBMS, enhanced with high availability, security, interoperability and performance capabilities. MariaDB ColumnStore provides a column-oriented storage engine and MariaDB Xpand supports distributed SQL.	Microsofts flagship relational DBMS	Widely used open source RDBMS	Widely used open source RDBMS 
Primary database model	Relational DBMS	Relational DBMS	Relational DBMS 	Relational DBMS 
Secondary database models	Document store Graph DBMS  Spatial DBMS	Document store Graph DBMS Spatial DBMS	Document store Spatial DBMS	Document store Spatial DBMS

<https://db-engines.com/en/system/MariaDB%3BMicrosoft+SQL+Server%3BMySQL%3BPostgreSQL>

Pipeline de Dados





Prática

Exercício 1

- Uma empresa deseja armazenar produtos, clientes e fornecedores. Quantas entidades (tabelas) são necessárias?
- Quais relacionamentos teríamos ?
- Quais regras teríamos para incluir os registros?

Exercício 2 – Base de dados que armazena estudantes e disciplinas cursadas

ESTUDANTE

nome	matricula	semestre	curso
Maria	1	1	IA
José	2	2	IA
Alfredo	3	2	EN

PRE_REQS

NOME_DISCIPLINA	SIGLA_DISCIPLINA	PRE REQ
COG. DATA SCIENCE	CD2001	CD1001
PLAT. COGNITIVAS	PG2001	PG1001

DISCIPLINA

nome_disciplina	sigla_disciplina	creditos	curso
Plataformas Cognitivas	PC1001	4	IA
Cognitive Data Science	CD1002	4	IA
Teoria dos Compiladores	TC2031	3	EN
Inteligência Artificial - Conceitos	IA2021	3	IA

NOTAS

matricula	sigla_disciplina	nota
1	PC1001	10
1	CD1002	9
1	IA2021	8
2	PC1001	8
2	CD1002	10
3	TC2031	6

FIAP

Exercício 2

- Quais relacionamentos vocês enxergam?
- Que consultas vocês fariam à base do slide anterior ?
- Se a sigla do curso IA mudasse para 1TIAR, quais colunas seriam alteradas?
- Existe alguma possível regra a ser incluída na base ?
- É possível melhorar essa base ? Tem informação redundante?

Exemplo – Base de dados que armazena estudantes e disciplinas cursadas

Nome da disciplina é redundante

PRE_REQS

Nome da disciplina	SIGLA_DISCIPLINA	PRE REQ
COG. DATA SCIENCE	CD2001	CD1001
PLATAFORMAS COGNITIVAS	PG2001	PG1001

DISCIPLINA

nome_disciplina	sigla_disciplina	creditos	curso
Plataformas Cognitivas	PC1001	4	IA
Cognitive Data Science	CD1002	4	IA
Teoria dos Compiladores	TC2031	3	EN
Inteligência Artificial - Conceitos	IA2021	3	IA

ESTUDANTE

nome	matricula	semestre	curso
Alfredo	1	1	IA
	2	2	IA
	3	2	EN

NOTAS

matricula	sigla_disciplina	nota
1	PC1001	10
1	CD1002	9
1	IA2021	8
2	PC1001	8
2	CD1002	10
3	TC2031	6

Consultas

- Listar participantes matriculados em um curso específico
- Consultar estudantes com notas inferiores a 6
- Consultar disciplinas de um curso específico
- Consultar estudantes de uma disciplina específica
- Consultar se estudante cumpriu os pre-reqs para cursar uma disciplina
- Caso um curso tenha um número de créditos mínimos, listar estudantes que já cumpriram todos os créditos.

Obrigado!

Contato: proffernando.fernandes@fiap.com.br