



BANCO DE DADOS 2

INTRODUÇÃO

Professor: Nicollas Cretton

E-mail: nicollas.cretton@professor.faminas.edu.br

INTRODUÇÃO

➤ **NoSQL** significa Not Only SQL (não apenas SQL).

➤ Nome dado a uma categoria de bancos de dados que não seguem o modelo relacional tradicional.

- **Não trabalham** com de tabelas, linhas e colunas estruturadas.

➤ O termo surgiu realmente em 2009.

PRINCIPAIS CARACTERÍSTICAS DOS BANCOS NOSQL

➤ **Flexibilidade**

- Não exigem uma estrutura de esquema fixa. Isso significa que diferentes registros podem ter diferentes conjuntos de campos. Podem ser desnormalizados.

➤ **Escalabilidade horizontal**

- São projetados para escalar adicionando mais máquinas ao invés de depender de uma única máquina poderosa. Os dados são particionados em várias máquinas.

PRINCIPAIS CARACTERÍSTICAS DOS BANCOS NOSQL

➤ Alta disponibilidade

- Os dados são replicados para garantir disponibilidade, mesmo que exista falha ou indisponibilidade de um servidor.

➤ Open Source

- A maioria dos bancos NoSQL são gratuitos.

TIPOS DE BANCOS NOSQL

➤ **Chave-Valor (Key-value)**

➤ **Colunar (Column)**

➤ **Grafos (Graph)**

➤ **Documentos (Document)**

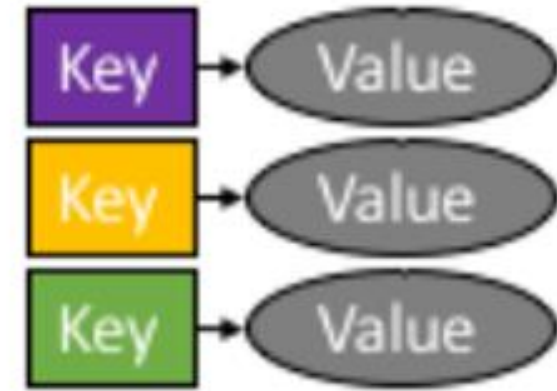
TIPOS DE BANCOS NOSQL

➤ Chave-Valor (Key-value)

- Os dados são armazenados em pares de chave-valor
 - Cada dado é identificado por uma chave única (string)
- Funciona como uma hash table
 - Os valores guardados podem ser de vários tipos:
 - BLOB (Binary Large Object)
 - JSON
 - String
 - Int

TIPOS DE BANCOS NOSQL

➤ Chave-Valor (Key-value)



Key-Value

Phone directory

Key	Value
Paul	(091) 9786453778
Greg	(091) 9686154559
Marco	(091) 9868564334

Key

Value

K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

TIPOS DE BANCOS NOSQL

➤ Chave-Valor (Key-value)



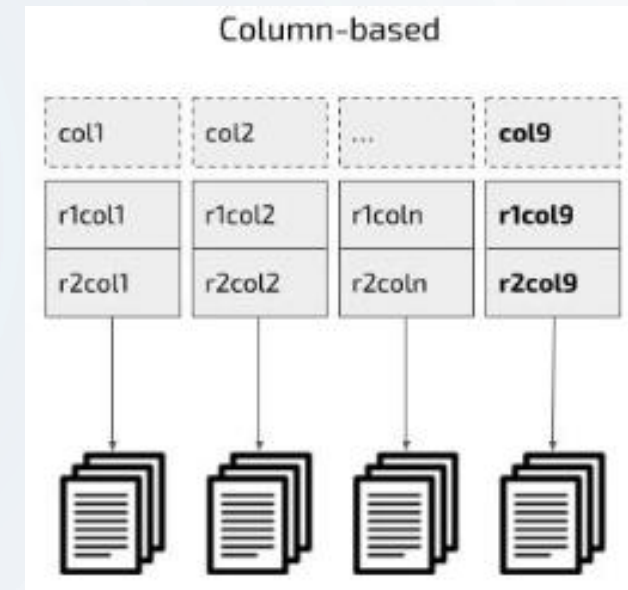
TIPOS DE BANCOS NOSQL

➤ Colunar (Column)

- Possui tabelas, linhas e colunas
 - As tabelas são na verdade uma família de colunas (column families)
 - As linhas continuam sendo os registros individuais
 - As colunas são armazenados separadamente, podendo estar em discos diferentes
- Otimiza a leitura/consulta de dados em massa, principalmente em agregação

TIPOS DE BANCOS NOSQL

➤ Colunar (Column)



Row oriented (Relational)

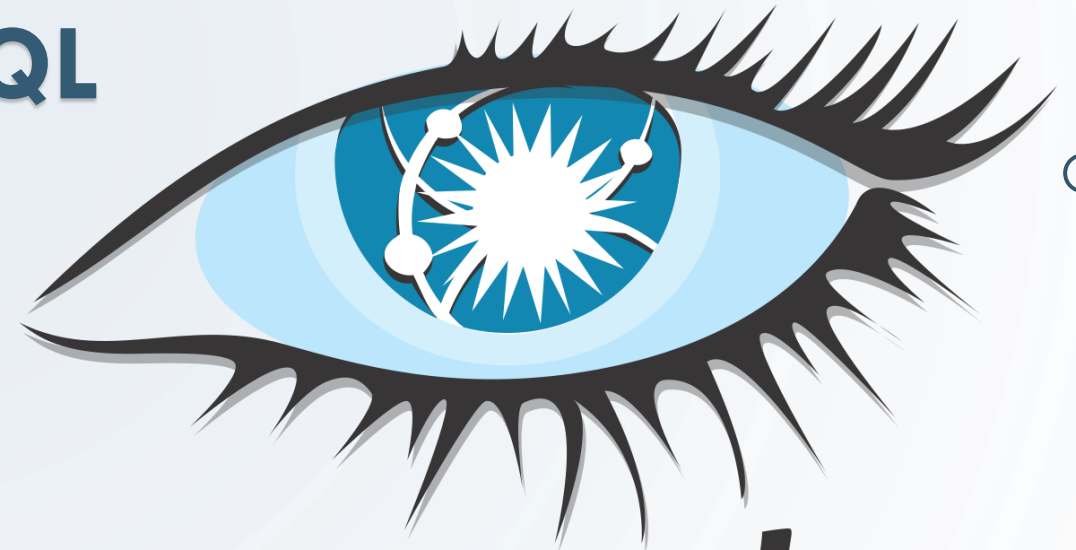
Students		
ID	First name	Last name
1	Luna	Lovegood
2	Hermione	Granger
3	Ron	Weasley

Column oriented

Students		
ID	First name	Last name
1	Luna	Lovegood
2	Hermione	Granger
3	Ron	Weasley

TIPOS DE BANCOS NOSQL

➤ Colunar (Column)



cassandra



Google
BigTable

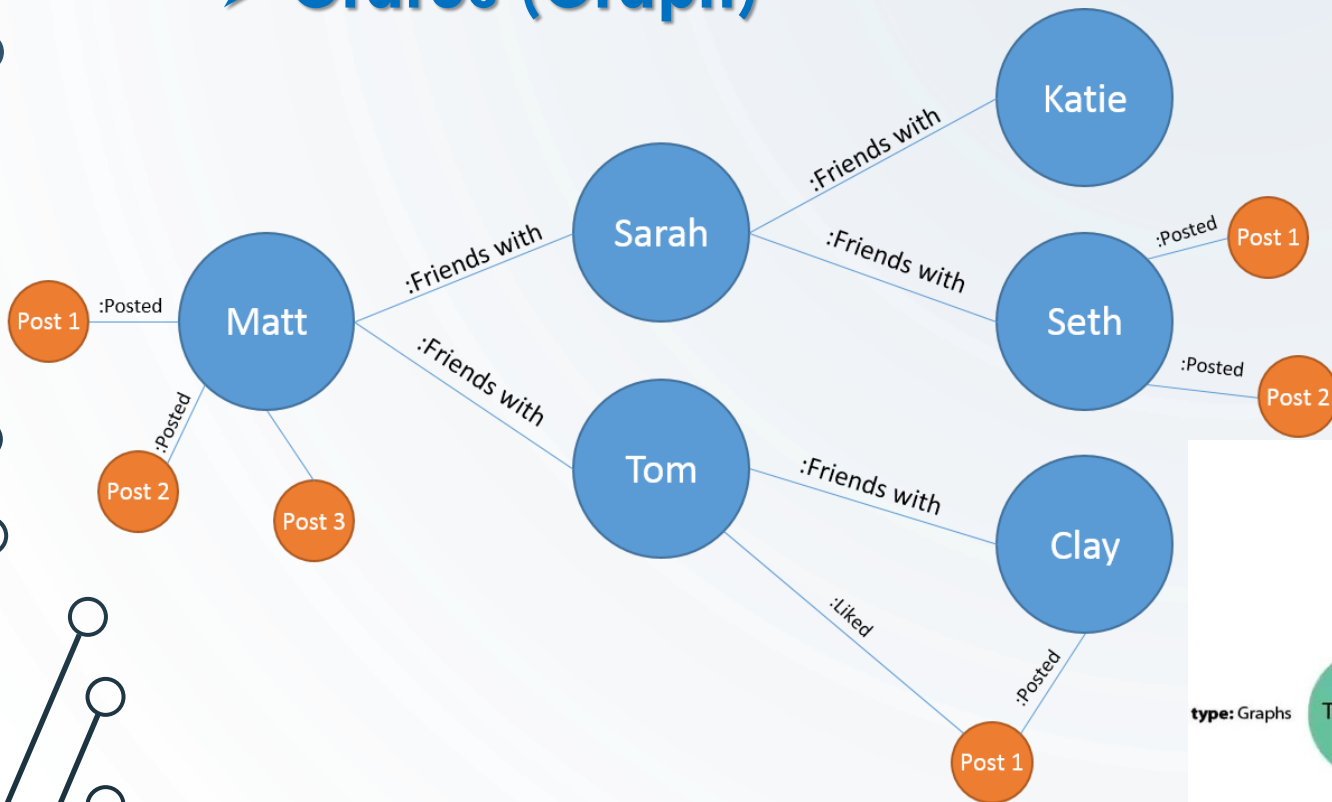
TIPOS DE BANCOS NOSQL

➤ Grafos (Graph)

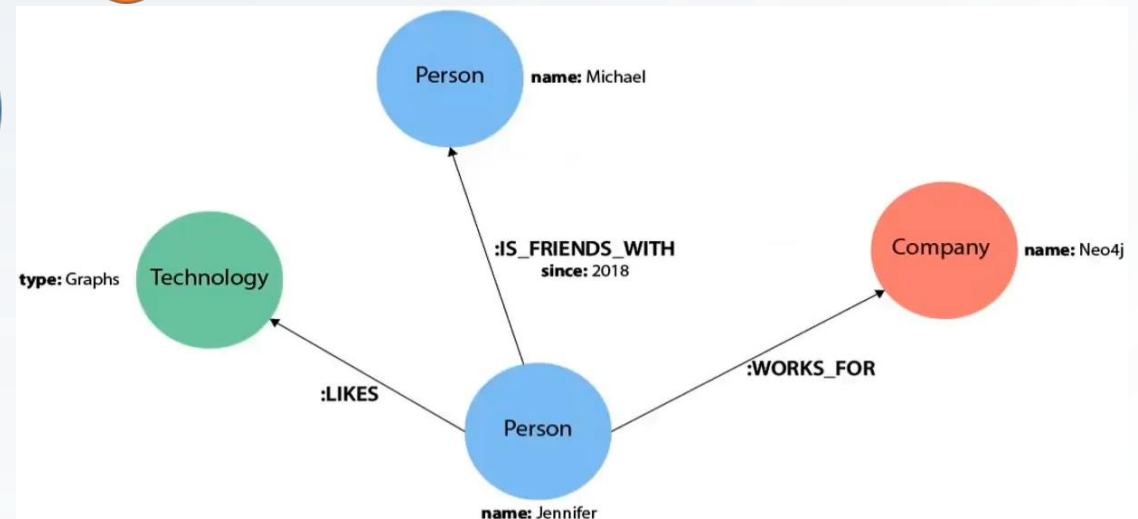
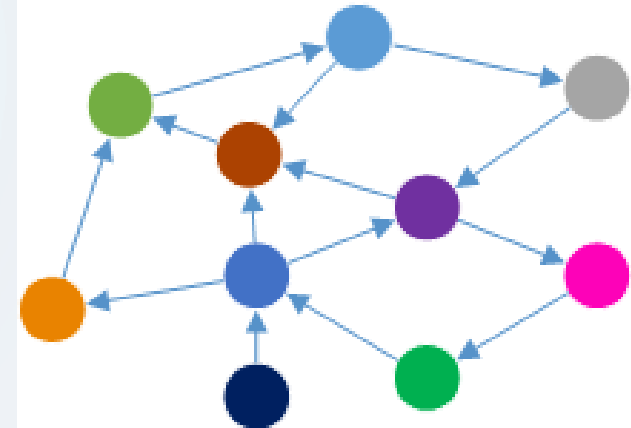
- É composto por vértices e arestas
 - Vértices (nó) guardam dados
 - Arestas guardam o relacionamento entre estes vértices
- O relacionamento é tratado como uma prioridade (junto com os dados)
 - Unidirecional: Maria segue João, mas João não necessariamente segue Maria (como no Twitter).
 - Bidirecional: Maria e João são amigos (como no Facebook).

TIPOS DE BANCOS NOSQL

➤ Grafos (Graph)



Graph Database

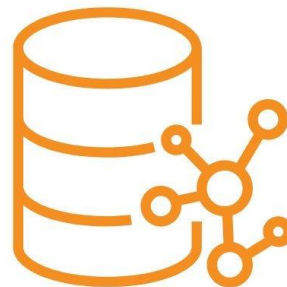


TIPOS DE BANCOS NOSQL

➤ Grafos (Graph)



ArangoDB



**Amazon
Neptune**

TIPOS DE BANCOS NOSQL

➤ Documentos (Document)

- Armazena os dados em documentos
 - JSON, BSON ou XML
- Cada documento é um registro autônomo
 - { "nome": "João", "idade": 25 }
 - { "nome": "Maria", "idade": 30, "telefone": "1 23-456-7890" }

TIPOS DE BANCOS NOSQL

➤ Documentos (Document)

- Pode ser:
 - Pares chave-valor (Key-value pairs):
 - Pares chave com valores em array (Key-arrayed value pairs)
 - Pares chave-valor embutidos (Embedded key-value pairs)

TIPOS DE BANCOS NOSQL

➤ Documentos (Document)

- Pode ser:
 - **Pares chave-valor (Key-value pairs):**
 - Estrutura básica
 - A **chave é o nome de um campo** e o valor pode ser qualquer tipo de dado
 - { "nome": "João", "idade": 25 }

TIPOS DE BANCOS NOSQL

➤ Documentos (Document)

- Pode ser:
 - **Pares chave com valores em array (Key-arrayed value pairs)**
 - Os valores associados a uma chave podem ser **arrays**
 - { "nome": "João", "telefones": ["123-456-7890", "987-654-3210"] }

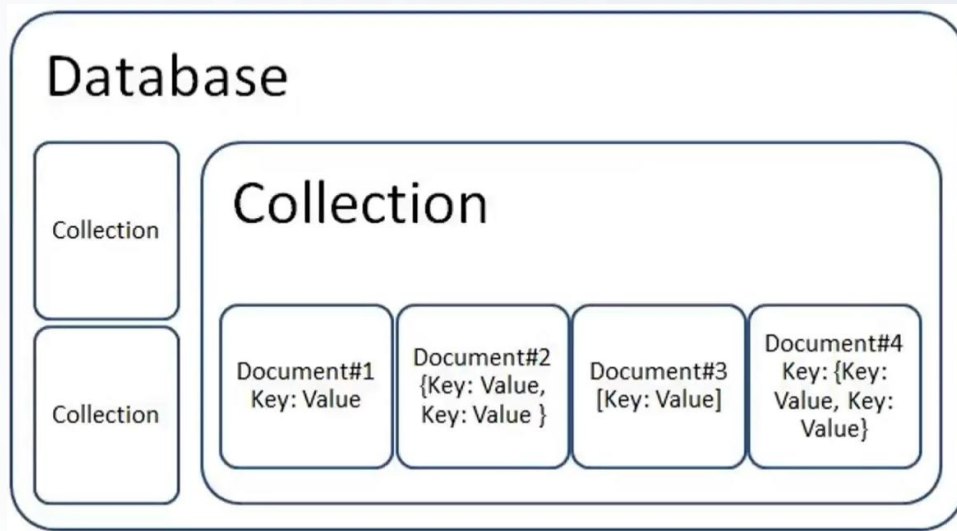
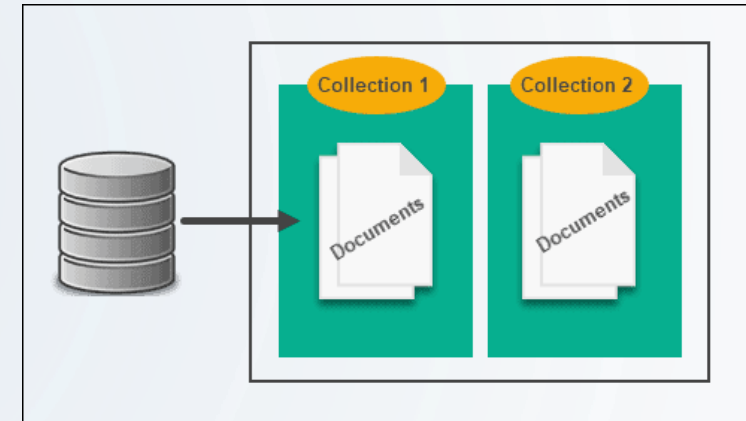
TIPOS DE BANCOS NOSQL

➤ Documentos (Document)

- Pode ser:
 - **Pares chave-valor embutidos (Embedded key-value pairs)**
 - Os documentos podem ter subdocumentos embutidos
 - {
 - "nome": "João",
 - "endereco": {
 - "rua": "Rua das Flores",
 - "cidade": "São Paulo"
 - }
 - }

TIPOS DE BANCOS NOSQL

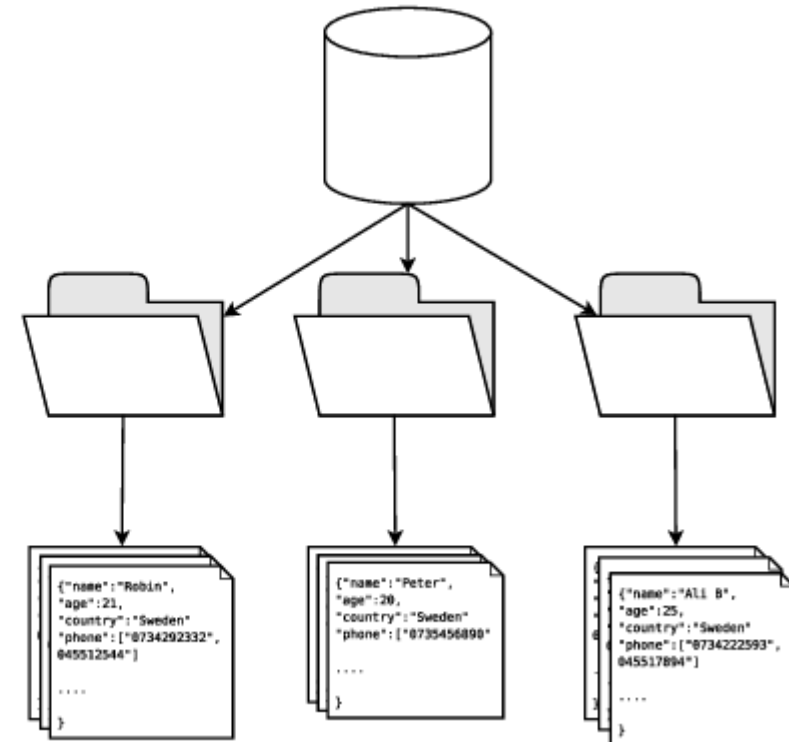
➤ Documentos (Document)



Database

Collections

Documents

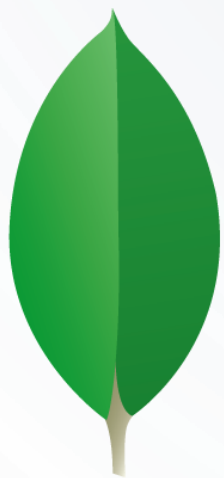


TIPOS DE BANCOS NOSQL

➤ Documentos (Document)



elasticsearch



mongoDB



CouchDB

MONGODB

- Nome vem da palavra “humongous”
 - Traduzindo grande pra C#
- Por ser **NoSQL**, não utiliza tabelas e esquemas rígidos como os bancos relacionais.
 - Os dados são armazenados em documentos
 - **JSON**: entrada e saída de dados
 - **BSON**: armazenamento interno
- Flexível. Não precisa declarar uma estrutura antes de armazenar os dados.
- Ótimo suporte para fazer consultas no banco.

MONGODB

➤ JSON (JavaScript Object Notation)

- É um formato de troca de dados leve e amplamente utilizado para armazenar e transferir informações entre sistemas.
- Ele é baseado em texto e segue uma estrutura simples e legível, com chaves e valores que podem ser de vários tipos.

➤ BSON (Binary JSON)

- Ele foi projetado para ser mais eficiente do que o JSON, suportando tipos de dados adicionais e melhor desempenho.
 - Date, Decimal128, ObjectId, etc.
- Por ser binário, ocupa menos espaço e é mais rápido do que JSON.

DIFERENÇAS ENTRE O MONGODB E SGBDR

➤ SGBDR (Sistema Gerenciador de Banco de Dados Relacional)

- RDBMS (Relational Database Management System)

RDBMS	MongoDB
É um banco de dados relacional.	É um banco de dados não relacional e orientado a documentos.
Não é adequado para armazenamento de dados hierárquicos.	Adequado para armazenamento de dados hierárquicos.
É escalável verticalmente (ou seja, aumentando a RAM).	É escalável horizontalmente (ou seja, podemos adicionar mais servidores).
Possui um esquema predefinido.	Possui um esquema dinâmico.
É bastante vulnerável a injeção de SQL.	Não é afetado por injeção de SQL.
Centraliza-se nas propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade).	Centraliza-se no teorema CDP (Consistência, Disponibilidade e Tolerância a Partições).

DIFERENÇAS ENTRE O MONGODB E SGBDR

RDBMS	MongoDB
É baseado em linhas.	É baseado em documentos.
É mais lento em comparação com o MongoDB.	É quase 100 vezes mais rápido que o RDBMS.
Suporta joins complexos.	Não suporta joins complexos.
É baseado em colunas.	É baseado em campos.
Não oferece cliente JavaScript para consultas.	Oferece um cliente JavaScript para consultas.
Suporta apenas a linguagem de consulta SQL.	Suporta a linguagem de consulta JSON juntamente com SQL.

TERMINOLOGIAS

SQL	MongoDB
Tabela	Coleção
Linha	Documento
Coluna	Campo
É baseado em colunas	É baseado em campos
Joins	Documentos Embutidos, Vinculação (Linking)

CDP

➤ **CDP** – **C**onsistência, **D**isponibilidade e Tolerância à **P**artição

- **CAP** – **C**onsistency, **A**vailability, **P**artition tolerance

- Teorema de Brewer, homenagem ao cientista da computação Eric Brewer

➤ É um sistema distribuído, podendo fornecer apenas duas das três características.

➤ Na presença de uma partição de rede, é necessário escolher entre consistência e disponibilidade.

- Se priorizar consistência, algumas solicitações podem não ser atendidas, sacrificando a disponibilidade.
- Se priorizar disponibilidade, os dados podem estar inconsistentes entre os nós.

CDP

➤ Consistência (Consistency)

- Garante que todos os nós do sistema vejam os mesmos dados ao mesmo tempo. Ou seja, após uma gravação de dados, todos os nós devem retornar o mesmo valor para uma leitura.
 - Todos os clientes veem os mesmos dados ao mesmo tempo.
 - Toda leitura recebe a gravação mais recente ou um erro.

CDP

➤ Disponibilidade (Availability)

- O sistema deve sempre responder a solicitações, mesmo que alguns nós falhem. Ou seja, o sistema continua acessível e operacional.
 - Qualquer cliente que faz uma solicitação de dados recebe uma resposta.
 - Toda solicitação recebe uma resposta (não um erro).

CDP

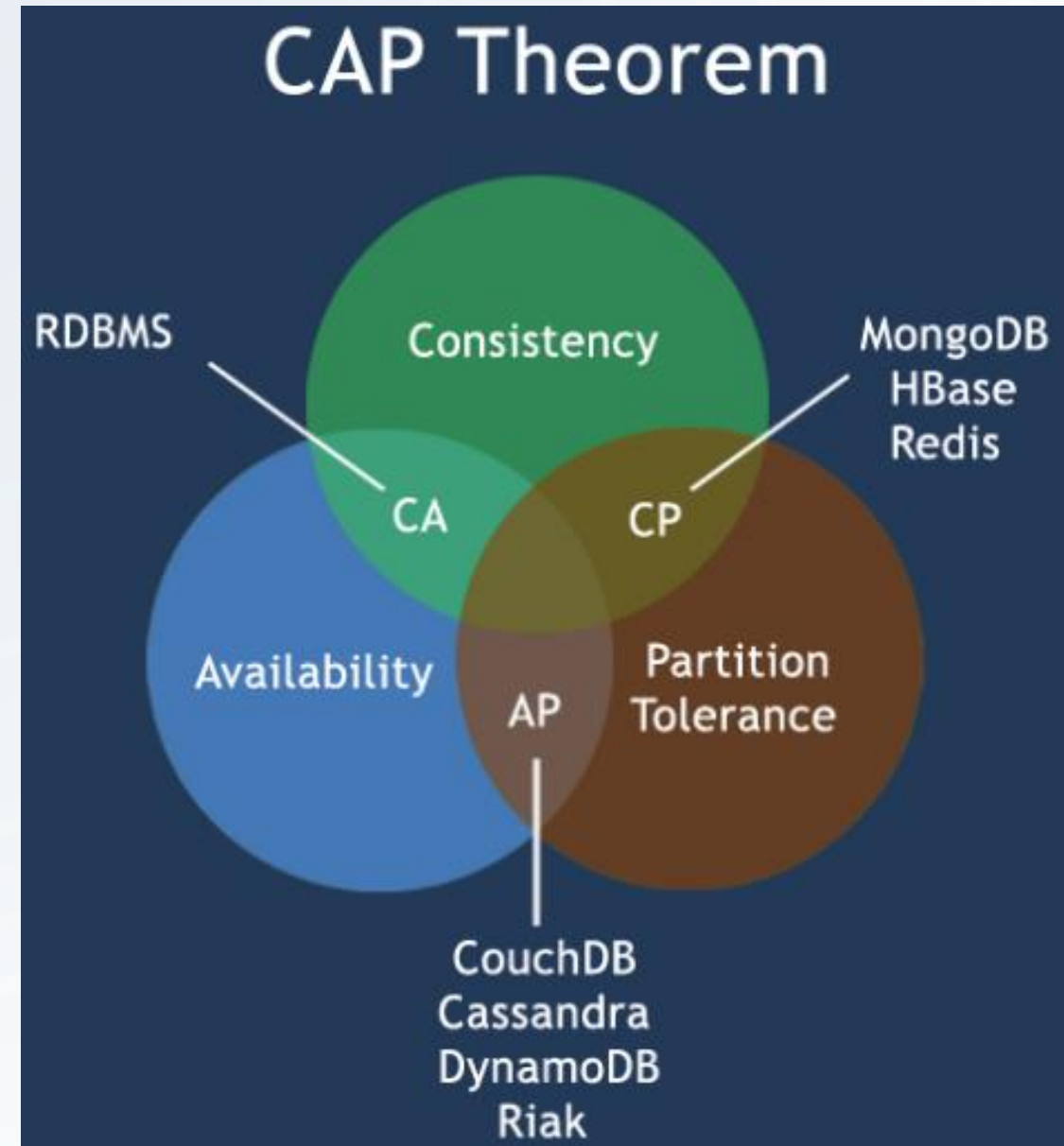
➤ Tolerância à Partição (Partition tolerance)

- O sistema continua a funcionar mesmo se houver falha ou perda de comunicação entre partes da rede. Em outras palavras, o sistema é resiliente a partições de rede.
 - Uma interrupção de comunicação dentro de um sistema distribuído.
 - O sistema continua a operar, apesar de mensagens serem descartadas (ou atrasadas).

CDP

➤ ○ **MongoDB** segue as características **CP**

- Um banco de dados **CP** oferece **consistência e tolerância à partição, à custa da disponibilidade.**
- Quando ocorre uma partição entre dois nós, o sistema precisa desligar o nó não consistente até que a partição seja resolvida.





BANCO DE DADOS 2

INTRODUÇÃO

Professor: Nicollas Cretton

E-mail: nicollas.cretton@professor.faminas.edu.br