

Curso Superior de Tecnologia em Desenvolvimento de Sistemas Multiplataforma

Disciplina: Banco de Dados Não Relacional

Onde estamos em Banco de Dados?

Prof. Diogo Branquinho Ramos

diogobranquinho@gmail.com

São José dos Campos - SP

Curso Superior de Tecnologia em Desenvolvimento de Sistemas Multiplataforma

Disciplina: Banco de Dados Não Relacional

Aula 02: Onde estamos em Banco de Dados?

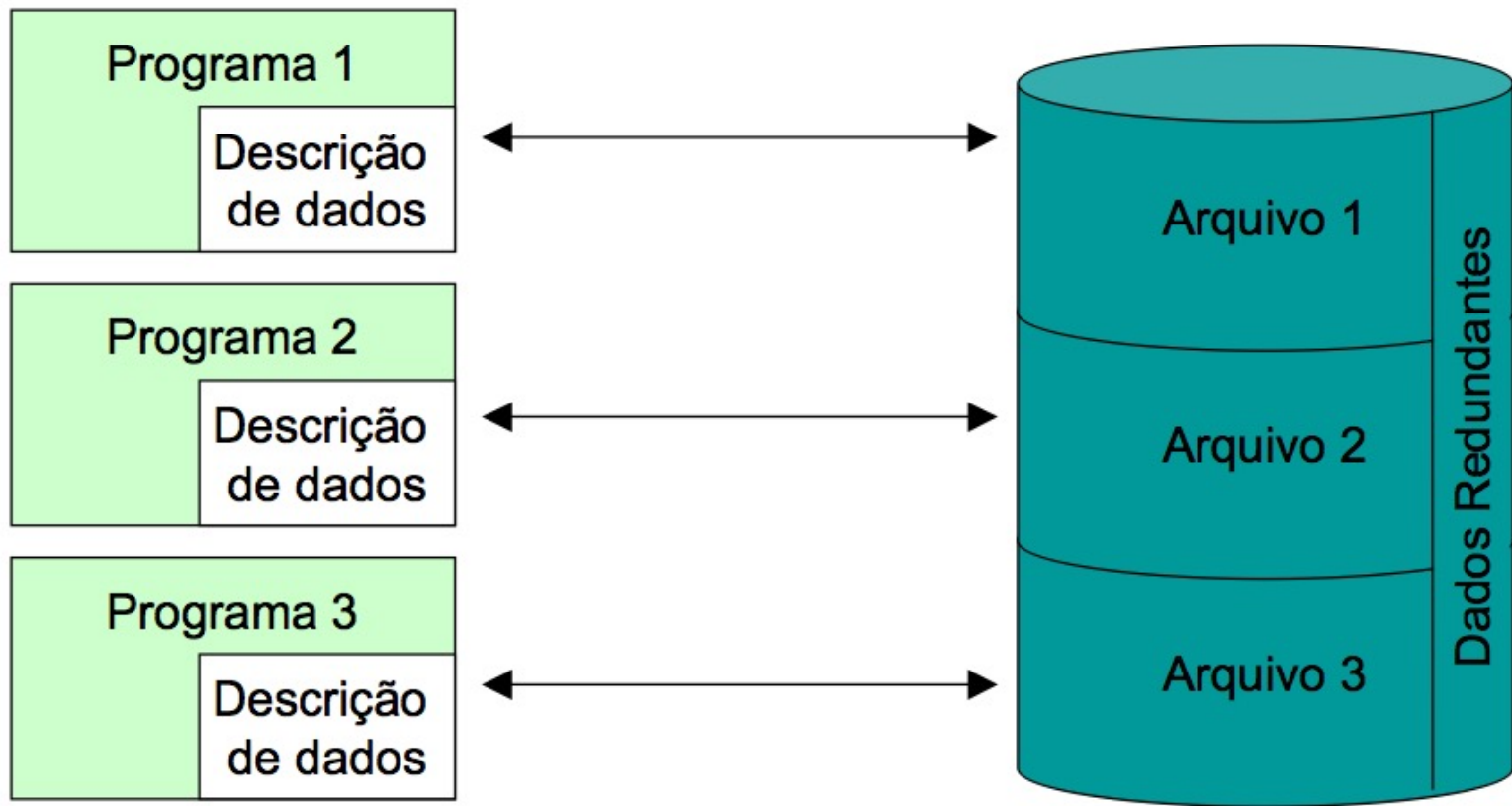
Prof. Diogo Branquinho Ramos

diogo.branquinho@fatec.sp.gov.br

São José dos Campos - SP

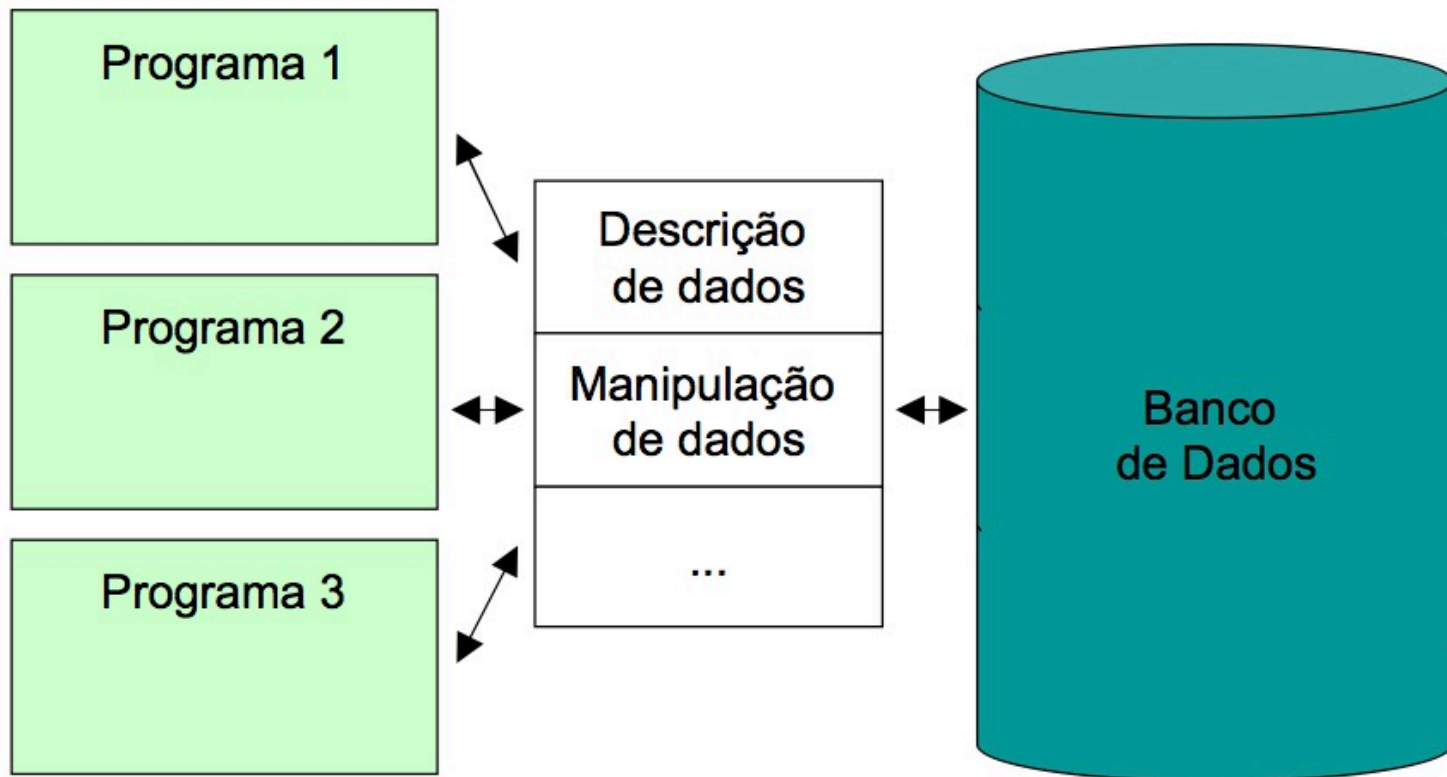
Histórico dos Sistemas de Bancos de Dados

- Antes: Processamento tradicional de arquivos



Histórico dos Sistemas de Bancos de Dados

- Processamento de Banco de Dados
 - Independência de dados



NoSQL = Not Only SQL



Você não é o Uber, Facebook...

Architecture Open Source

Why Uber Engineering Switched from Postgres to MySQL

Evan Klitzke

The Architecture of Postgres

We encountered many Postgres limitations:

- Inefficient architecture for writes
- Inefficient data replication
- Issues with table corruption
- Poor replica MVCC support
- Difficulty upgrading to newer releases

Netflix's big leap from Oracle to Cassandra

Roopa Tangirala (Netflix)
1:50pm-2:30pm Thursday, 03/31/2016
Enterprise Adoption

Location: LL20 D

Tags: media

Average rating: ★★★★★ (4.16, 19 ratings)

 Download slides (PPTX)

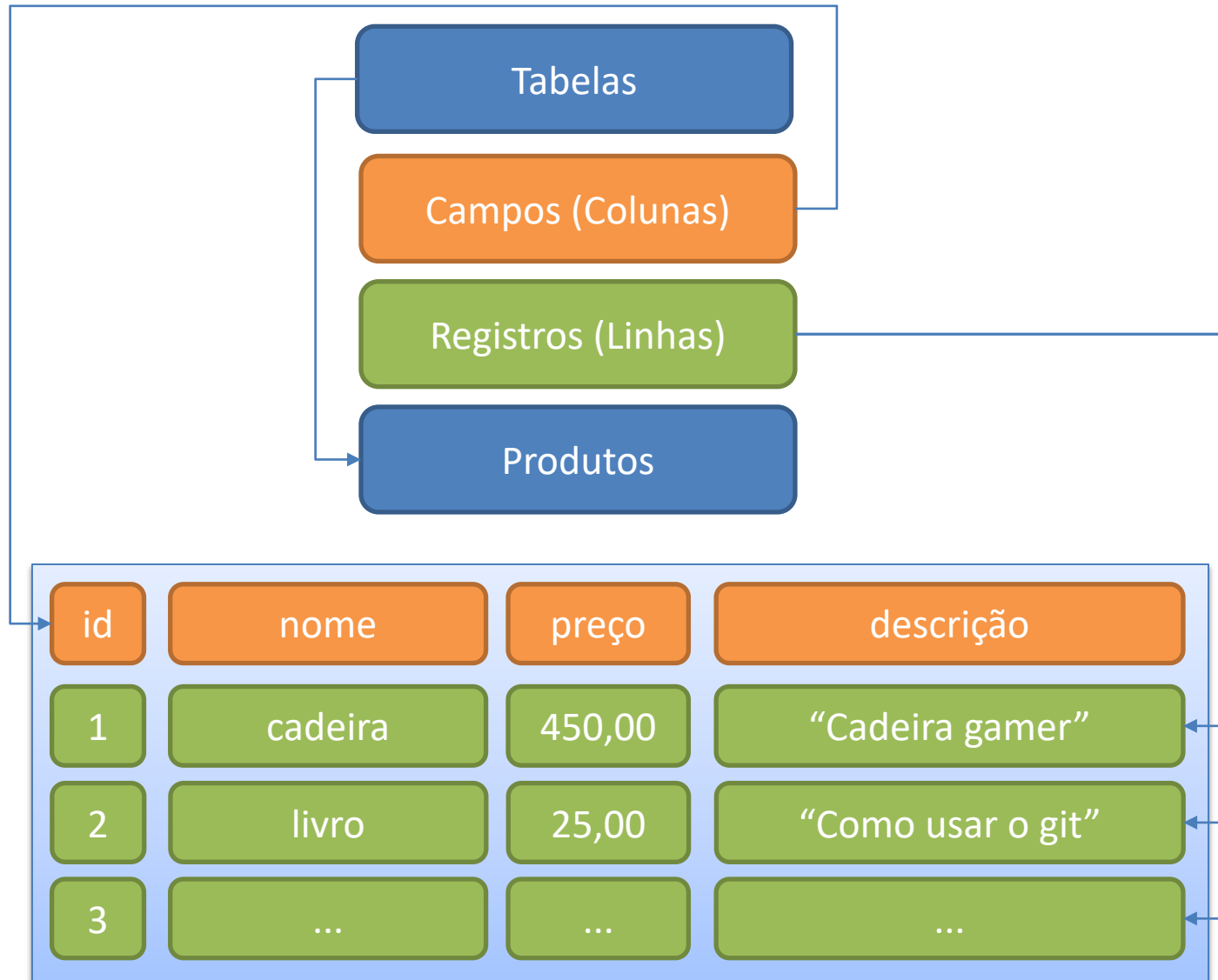
Casos extremos de performance

<https://eng.uber.com/postgres-to-mysql-migration/>
<https://conferences.oreilly.com/strata/strata-ca-2016/public/schedule/detail/46839>

Structured Query Language - SQL

- Liguagem de Consulta Estruturada
- Bancos de Dados Relacionais
- Teoria dos conjuntos
- Projeto de SQL é dos anos 70
- Estrutura de dados rígida

Schema BD Relacional



Schema & Normalização dos Dados

Produtos

| id | nome | preço | descrição |
|----|---------|--------|-------------------|
| 1 | cadeira | 450,00 | "Cadeira gamer" |
| 2 | livro | 25,00 | "Como usar o git" |
| 3 | ... | ... | ... |

Todos os registros devem seguir o schema

Relacionamentos

Um-para-Um

Um-para-Muitos

Muitos-para-Muitos

Usuários

| id | email | nome |
|----|-----------------|-------------|
| 1 | ze@teste.com | José Santos |
| 2 | ma@teste.com.br | Maria Silva |
| 3 | ... | ... |

Compras

| id | usuário_id | produto_id |
|----|------------|------------|
| 1 | 1 | 2 |
| 2 | 1 | 1 |
| 3 | 2 | 1 |

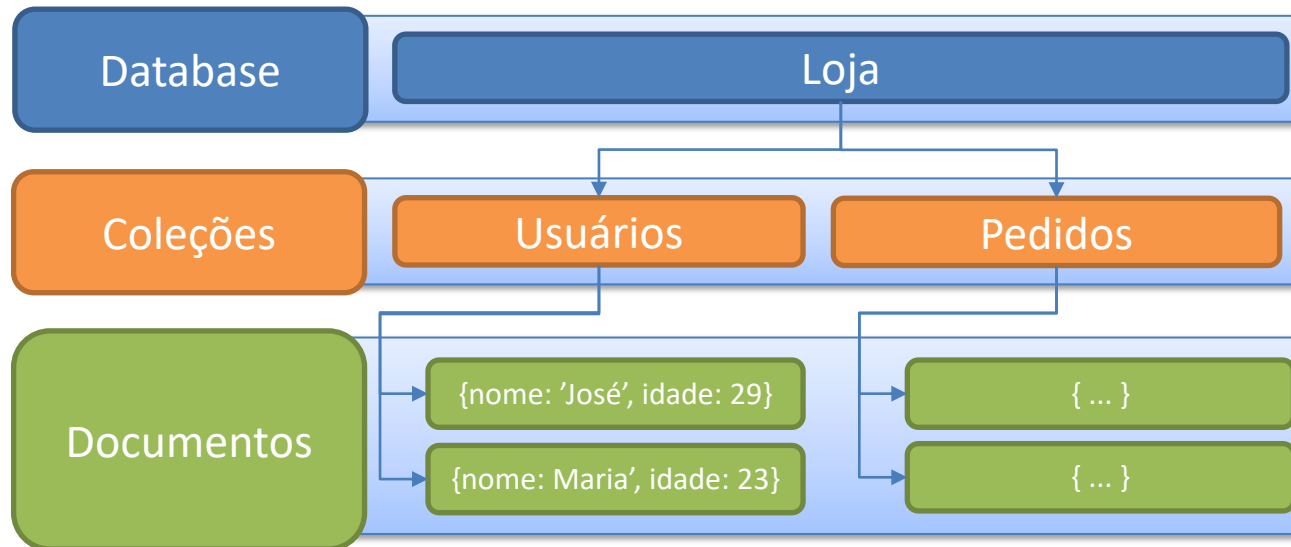
Produtos

| id | nome | preço | descrição |
|----|---------|--------|-------------------|
| 1 | cadeira | 450,00 | "Cadeira gamer" |
| 2 | livro | 25,00 | "Como usar o git" |
| 3 | ... | ... | ... |

NoSQL – Not only SQL

- Não relacional
- Open Source
- Cluster-friendly
- Sem estrutura fixa
- 2009
- Modelos de Dados
 - Documents – MongoDB, CouchDB, RavenDB
 - Column-Family – Cassandra, Apache, HBase
 - Key-value – Redis, Riak
 - Graph – Neo4j

noSQL



Estrutura do Dado

Sem schema

| | | | |
|-------|---------------|-----------|-----|
| Id: 1 | nome: 'José' | idade: 29 | ... |
| Id: 2 | nome: 'Maria' | | ... |
| Id: 3 | | idade: 31 | ... |

Relacionamentos

Sem / poucos relacionamentos

Dados relacionais precisam ser realizados manualmente

NoSQL pode ser mais eficiente em armazenar objetos e dados que não precisam ser normalizados.

Relacionamentos - NoSQL

Compras

{id: 212451, user:{id: 1, email: 'ze@teste.com'}, product: {id:2, preço: 25,00}}

{id: 215464, user:{id: 2, email: 'maria@teste.com'}, product: {id:1, preço: 450,00}}

{id: 154156, product: {id:2, preço: 25,00}}

{...}

Usuários

{id: 1, nome: 'José', email: 'ze@teste.com'}

{id: 2, nome: 'Maria', email: 'maria@teste.com'}

{...}

Produtos

{id: 1, nome: 'Cadeira', preço: 450,00}

{id: 2, nome: 'Livro', preço: 25,00}

{...}

25d54f88-adfb-11eb-8529-0242ac130003

Diogo postou:

Kkkkkk chegou minhas férias

Comentário:

Marco: showww

Marcos: agora vai

Raian: aproveite

Usuário

1 | Diogo | 37 | 1 - Harry Potter - Abr-2020 | 2 - Senhor dos Aneis Set-2021

2 | Marcos | 4 -50 tons de cinza | 5 - Kamasutra

SQL vs NoSQL

SQL

Schema para estrutura dos dados

Relações (1-1, 1-n, n-n)

Os dados são distribuídos por múltiplas tabelas

Limitações para blocos de queries de leitura e escrita por segundo

Garante ACID

MySQL

NoSQL

Sem schema

Sem/poucas relações

Os dados são agregados/aninhados em algumas coleções

Boa performance para blocos de queries de leitura e escrita por segundo

Não garante ACID

Redis

Junção das duas tecnologias

Quando usar NoSQL

- Se você sabe o que está fazendo
 - Soluções em SQL tendem a funcionar mesmo com problemas
- Se a sua aplicação não precisa de um esquema claro
- Se o seu front-end já faz todas as validações e regras de controle
- Se a sua aplicação precisa de muitas leituras
- Se a sua aplicação precisa de muitas escritas (não updates)
- Se existem funções que são muito acessadas (milhares e milhares de vezes) e você precisa garantir que as queries executem o mais rápido possível
- Se a sua aplicação permite um relaxamento de ACID

Quando o assunto é performance

- Se você executa muitas operações de Map/Reducê, voce vai acabar usando [Hadoop](#).
- Se você utilizar shading de forma pesada e só faz consultas *Key-Value*, então [Riak KV](#) pode ser uma opção interessante.
- Se você se preocupa com uma possível escalabilidade massiva [Cassandra](#) pode ser uma melhor opção.
- Se você precisa de cache, Redis e Memcached podem ser o suficiente.
- Se você precisa de queues, você eventualmente vai evoluir para [RabbitMQ](#), [ZeroMQ](#), etc.
- Se você precisa de buscas, Elasticsearch, Solr ou Sphinx serão mais poderosos.
- Quanto Hibernate no Java, SQLAlchemy no Python, ActiveRecord no Ruby, Eloquent ORM no PHP não fazem queries eficientes

MongoDB University

- Curso online gratuito de 7 semanas
- Conteúdo mais que o essencial para trabalhar com NoSQL e MongoDB
- <https://university.mongodb.com>

| | | |
|--|--|--|
| <p>INTRO COURSE</p> <p>M001</p> <p>MongoDB Basics</p> <p>Start by learning the fundamentals of MongoDB. Then, build and deploy an application leveraging any MongoDB driver.</p> <p>View Course</p> | <p>RECOMMENDED</p> <p>M121</p> <p>The MongoDB Aggregation Framework</p> <p>Learn how to use MongoDB's Aggregation Framework.</p> <p>View Course</p> | <p>RECOMMENDED</p> <p>M320</p> <p>MongoDB Data Modeling</p> <p>Explore best practices, common patterns, and techniques for effectively creating data models for MongoDB.</p> <p>View Course</p> |
|--|--|--|