

FIAP GRADUAÇÃO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MASTERING RELATIONAL AND NON-RELATIONAL DATABASES

Prof. Dr. Renê de Ávila Mendes

Renê de Ávila Mendes



- Graduado em Sistemas de Informação (2007)
- Mestre em Engenharia Elétrica (2017)
- Doutor em E.E. e Computação (2021)
- Analista de Banco de Dados no Instituto Presbiteriano Mackenzie
- Professor atualmente na FIAP, e em cursos de Pós-Graduação na Unicsul, Unianchieta, Uniútao
- Instrutor certificado de Knime Analytics Platform (2022 a 2024)

linkedin.com/in/renemendes

pf1752@fiap.com.br

1º SEMESTRE

CALENDÁRIO 2026

01 JANEIRO

- 01 Confraternização Universal
- 25 Aniversário de São Paulo
- 01 A 31 Período de férias

02 FEVEREIRO

- 09 Início das Aulas (veteranos)
- 16 E 17 Carnaval (aulas suspensas)
- 18 Quarta-feira de Cinzas (aulas suspensas)
- 23 Aula Inaugural (calouros)
- 23 A 27 FIAP First Week (calouros)

03 MARÇO

- 02 Início das Aulas (calouros)
- 09 A 13 Período para solicitação de mudança de turma e curso
- 09 A 13 Período para solicitação de dispensa de disciplina
- 16 A 20 Divulgação dos pedidos de mudança de turma e curso
- 16 A 20 Período para regulamentação das disciplinas em regime de dependência
- 30 Divulgação das dispensas das disciplinas

04 ABRIL

- 02 Quinta-feira Santa (aulas suspensas)
- 03 Paixão de Cristo (aulas suspensas)
- 05 Páscoa
- 20 Aulas suspensas
- 21 Trindades (aulas suspensas)

05 MAIO

- 01 Dia Mundial do Trabalho (aulas suspensas)
- 25 A 05/06 Período de avaliação semestral (Global Solutions)



Fevereiro - 10,24

Março - 3,(10),27,24,(31)

Abril - 7,14,28

Maio - (5),12,19

CHECKPOINT

10/03 (remoto)

31/03 (remoto)

05/05 (presencial)

Objetivos da disciplina

DISCIPLINA: Mastering Relational and Non-Relational Databases

OBJETIVOS: Durante este curso, você mergulhará na linguagem ANSI SQL e PL/SQL, desvendando sua versatilidade no contexto do banco de dados Oracle. Aprenderá sobre componentes cruciais, incluindo procedures, functions, packages, triggers, collections, PL/SQL tables, PL/SQL com Bulk processing, Autonomous transactions e Exception handling. Além disso, receberá orientações essenciais para aprimorar a otimização de consultas SQL. Navegue pelo mundo do MongoDB para aprofundar os conhecimentos em bancos de dados não relacionais. Do design sem esquema ao armazenamento flexível de documentos, você mergulhará no cerne do MongoDB e suas capacidades. Ganhe expertise em consultas usando a Linguagem de Consulta MongoDB (MQL), aprenda a realizar operações eficientes de CRUD e aproveite o poder dos pipelines de agregação para processamento avançado de dados. Explore o rico ecossistema do MongoDB e aprenda sobre índices, modelagem de dados e as melhores práticas para aplicações de alto desempenho. Domine conceitos como shard e replicação para garantir escalabilidade e disponibilidade de dados. Além disso, descubra a integração do MongoDB com linguagens de programação e estruturas populares.

| CONTEÚDO DO CURSO

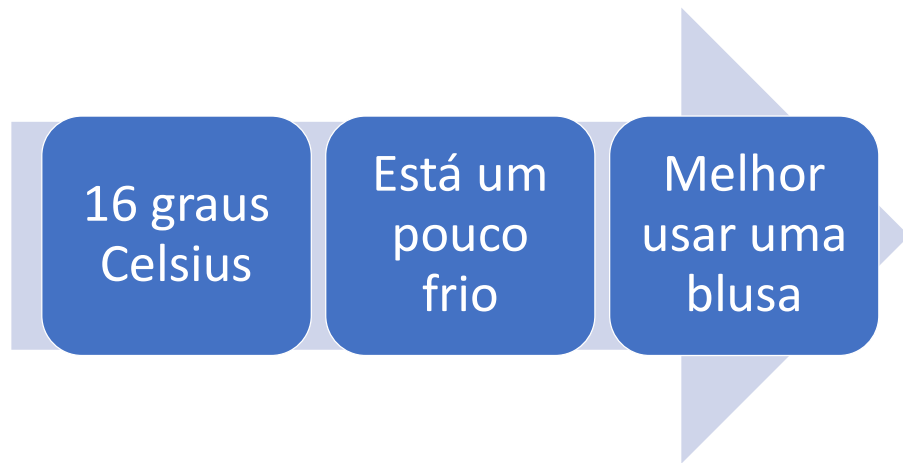
Mastering Relational and NON-Relational Database

- Conceitos de PL/SQL
- Tipo de dados
- Estruturas de Controle
- Cursores
- Storage Procedures
- Functions
- Packages
- Triggers
- Introdução a Big Data
- Introdução a MongoDB
- MongoDB - Operações CRUD
- Introdução a Neo4J
- Knowledge Graphs
- Introdução a Redis

CONTEXTUALIZAÇÃO



Dados, informação, conhecimento



“O dado é um fato, um valor documentado ou um resultado de medição. Quando um sentido semântico ou um significado é atribuído aos dados, gera-se informação. Quando estes significados se tornam familiares, ou seja, quando um agente os aprende, este se torna consciente e capaz de tomar decisões a partir deles, e surge o conhecimento.”

(DA SILVA, Leandro Augusto; PERES, Sarajane Marques; BOSCARIOLI, Clodis. **Introdução à mineração de dados: com aplicações em R**. Elsevier Brasil, 2017, p6.)

Dados – A realidade atual



{JSON}

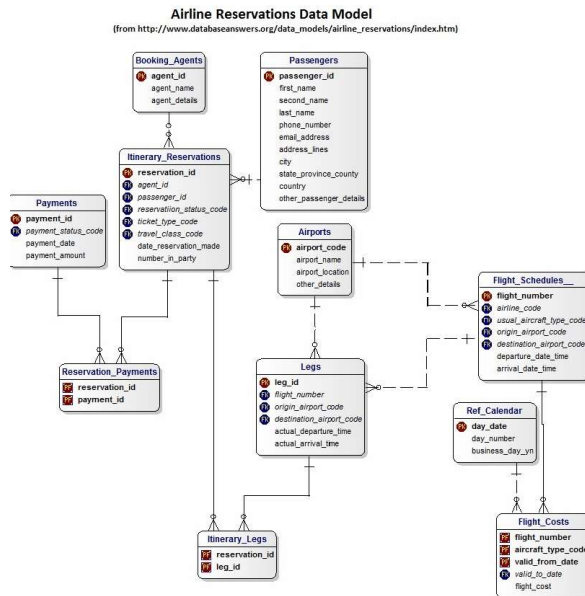
Dados estruturados



DADOS ESTRUTURADOS – dados que podem ser descritos por um **modelo de tipos de dados** e que podem ser armazenados, recuperados e processados com base neste modelo. Podem ser facilmente carregados, armazenados, recuperados e analisados. Geralmente são armazenados em bancos de dados relacionais. Exemplos: dados obtidos por formulários.

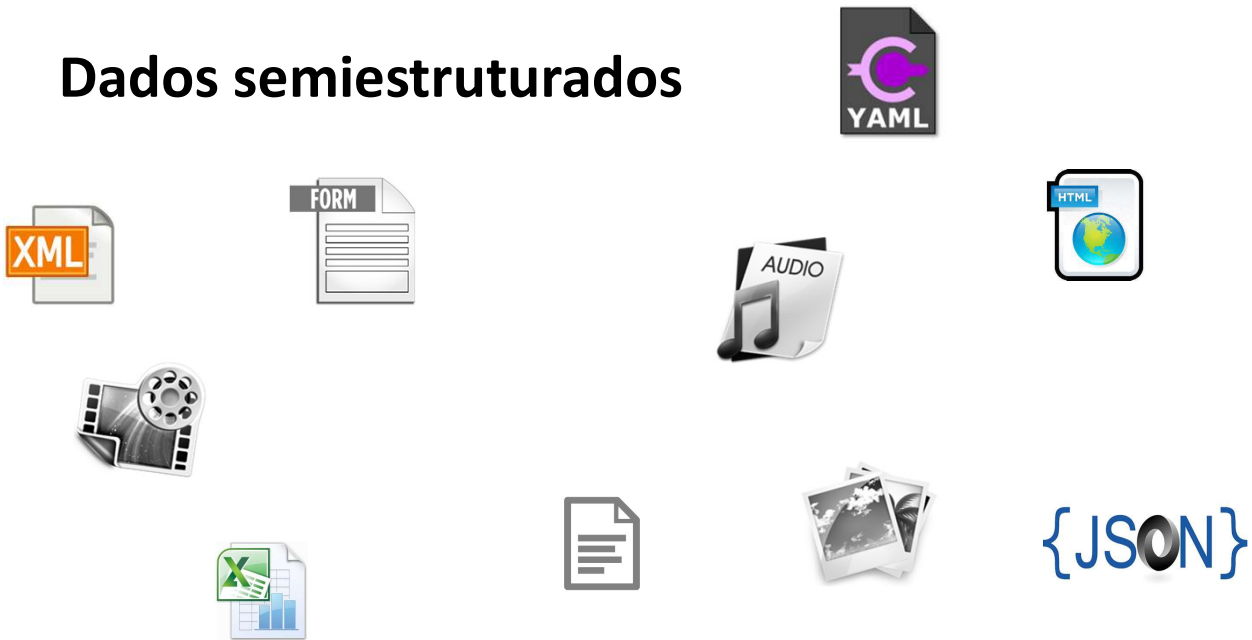
Structured - Os dados estruturados geralmente são gerenciados por SQL, uma linguagem de programação criada para gerenciar e consultar dados nos SGBDR. Os dados estruturados são fáceis de inserir, consultar, armazenar e analisar. Exemplos de dados estruturados incluem números, palavras e datas.

Dados estruturados – Exemplo



Um modelo relacional é um bom exemplo de armazenamento de dados estruturados.

Dados semiestruturados



DADOS SEMIESTRUTURADOS – uma forma de dados estruturados que não se conforma à estrutura formal dos modelos de dados utilizados em bancos relacionais, **mas que contêm tags ou outros marcadores que separam elementos semânticos e que garantem hierarquias de campos e registros dentro dos dados.** Geralmente armazenados em bancos de dados orientados a objetos. Exemplos: XML e JSON

Semi-structured - Dados semiestruturados são dados que não seguem um sistema de banco de dados convencional. Os dados semiestruturados podem estar na forma de dados estruturados que não são organizados em modelos de banco de dados relacional, como tabelas. Capturar dados semiestruturados para análise é diferente de capturar um formato de arquivo fixo. Portanto, a captura de dados semiestruturados requer o uso de regras complexas que decidem dinamicamente o próximo processo após a captura dos dados.

Dados semiestruturados – Exemplos

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

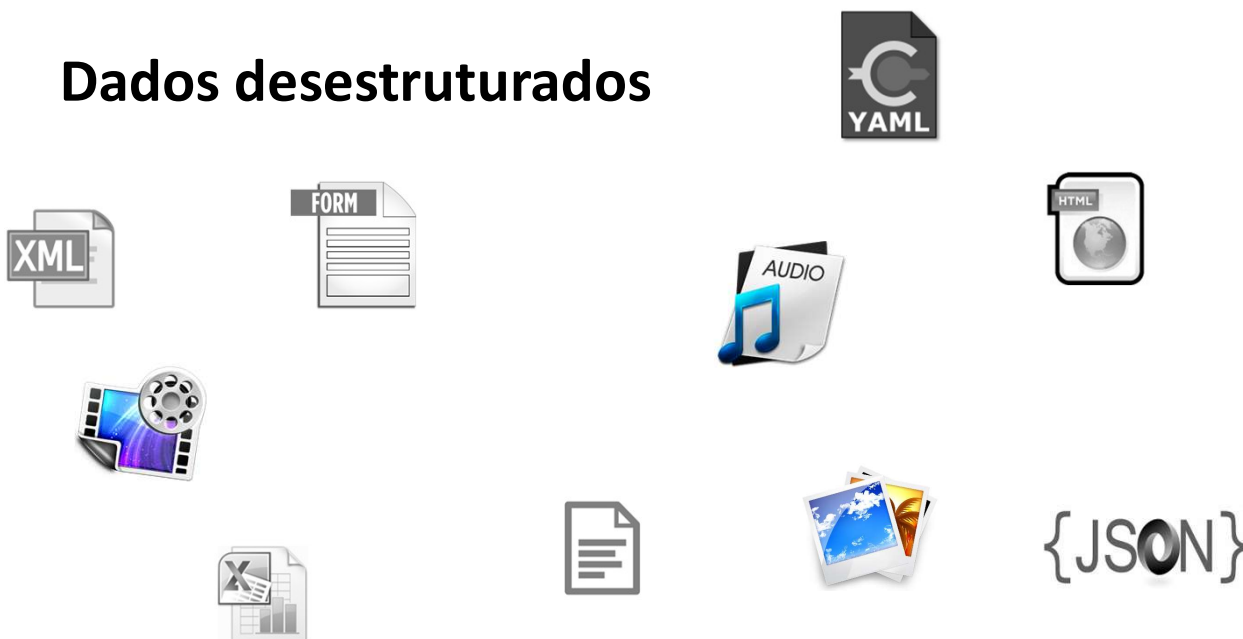
```
first name: John
last name: Smith
age: 25
address:
  street address: 21 2nd Street
  city: New York
  state: NY
  postal code: '10021'
phone numbers:
  - type: home
    number: 212 555-1234
  - type: fax
    number: 646 555-4567
sex:
  type: male
```

```
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <age>25</age>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumbers>
    <phoneNumber>
      <type>home</type>
      <number>212 555-1234</number>
    </phoneNumber>
    <phoneNumber>
      <type>fax</type>
      <number>646 555-4567</number>
    </phoneNumber>
  </phoneNumbers>
  <sex>
    <type>male</type>
  </sex>
</person>
```

Armazenamento de dados nos formatos JSON, YAML e XML.

Fonte: <https://en.wikipedia.org/wiki/JSON>

Dados desestruturados



DADOS DESESTRUTURADOS – dados que podem ser analisados mas que não se apresentam de forma estruturada. Podem ser transformados em dados estruturados por um processo que descreve os documentos desestruturados por suas características principais usando-se metadados.

Unstructured - Dados não estruturados, como mensagens de texto, informações de localização, vídeos e dados de mídia social, são dados que não seguem um formato especificado. Considerando que o tamanho desse tipo de dados continua aumentando com o uso de smartphones, a necessidade de analisar e entender esses dados se tornou um desafio.

A photograph of Eric Schmidt, CEO of Google, speaking and gesturing with his hands. The image is semi-transparent, serving as a background for the quote.

“

A cada **dois dias** nós criamos 5 exabytes de dados, isso é o **mesmo** que foi criado do início da civilização até **2003**.

-Eric Schmidt (CEO do Google)

<https://www.forbes.com/sites/ciocentral/2012/04/24/the-web-is-much-bigger-and-smaller-than-you-think>

Visualizando o volumes de dados

Byte	1 grão de arroz
Kilobyte	1 xícara de arroz
Megabyte	8 sacos de arroz
Gigabyte	1 container de arroz
Terabyte	2 navios cargueiros
Petabyte	suficiente para cobrir a cidade de Campinas.
Exabyte	suficiente para cobrir os estados de Minas Gerais, Rio de Janeiro, Espírito Santo e São Paulo.
Zettabyte	preenche o oceano Pacífico.

Para que serve um banco de dados?



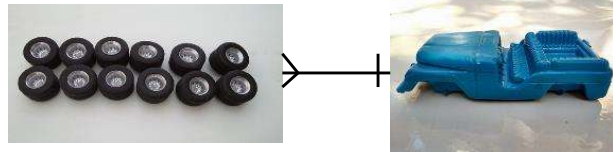
Para que serve um banco de dados?



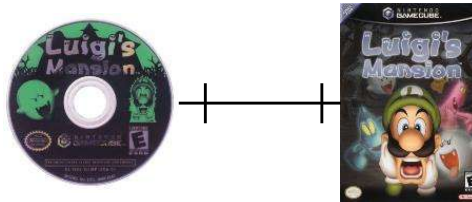
- Como faço para encontrar coisas de maneira fácil e rápida?

- Bancos de dados resolvem o mesmo problema: organizam dados para facilitar a localização posterior.

Banco de dados relacionais



- Quando organizamos brinquedos faz sentido guardarmos rodas soltas de carrinhos em uma caixa e os carrinhos sem roda em outra.
- Bancos de dados relacionais também separam dados em “caixas” diferentes.
- São chamados de relacionais porque mantêm o controle de quais rodas são de qual carrinho sem roda, ou seja, qual roda está relacionada a qual carrinho.
- Em bancos de dados relacionais, as “caixas” são chamadas de “tabelas”.



Banco de dados não relacionais



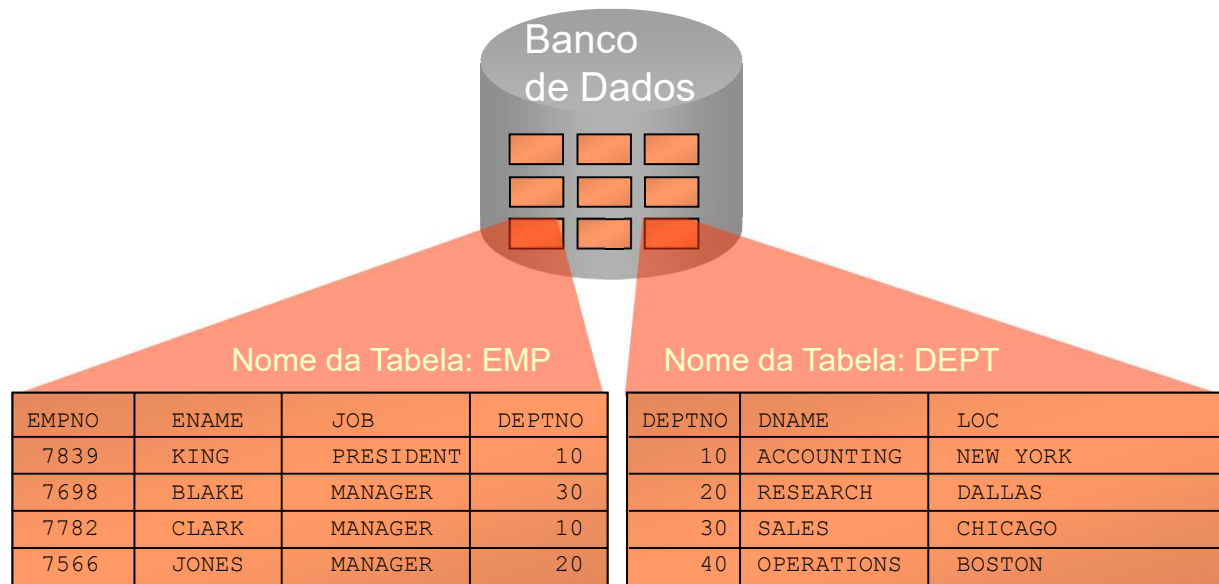
- Existem bancos de dados que permitem que guardemos um brinquedo e todas as suas partes na mesma “caixa”.
- Esse tipo de banco é chamado de banco não relacional ou NoSQL.
- Alguns modelos de dados de banco NoSQL:
 - Chave/Valor (Key/value)
 - Colunas (Columnar)
 - Documento (Document)
 - Grafos (Graph)

- **SGBD – Sistema Gerenciador de Banco de Dados**

- conjunto de programas que permite gerenciar os **bancos de dados**, dar acesso aos **dados** de maneira simples, autorizar o acesso às informações a múltiplos usuários e incluir , alterar e remover os **dados** presentes no **banco de dados**.
- Também conhecido como DBMS (*Database Management System*) ou RDBMS (*Relational Database Management System*)

Definição de Banco de Dados Relacional

Um banco de dados relacional é um conjunto de relações ou tabelas bidimensionais.



Definição de Banco de Dados Relacional

Um **banco de dados relacional** usa relações ou tabelas bidimensionais para armazenar informações.

Por exemplo, você pode armazenar informações sobre todos os funcionários de uma empresa. Em

um banco de dados relacional, você cria várias tabelas para armazenar informações diferentes sobre funcionários, como tabelas de funcionários, departamentos e salários.

Terminologia de Banco de Dados Relacional

2	3	4					
1	6	5					
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Terminologia de Banco de Dados Relacional

Um banco de dados relacional pode conter uma ou várias tabelas. Uma *tabela* é a estrutura de armazenamento básica de um RDBMS. Ela armazena todos os dados necessários sobre algo do mundo real, por exemplo, funcionários, NFFs ou clientes.

O slide mostra o conteúdo da *relação* ou *tabela* EMP. Os números indicam o seguinte:

1. Uma *linha* simples ou *tupla* que representa todos os dados necessários para um funcionário específico. Cada linha de uma tabela deve ser identificada por uma chave primária, que não permite linhas duplicadas. A ordem das linhas não é importante; especifique essa ordem quando os dados forem recuperados.
2. Uma *coluna* ou *atributo* que contém o número do funcionário, que é também a chave primária. O número do funcionário identifica um *único* funcionário na tabela EMP. Uma chave primária deve conter um valor.
3. Uma coluna que não é um valor de chave. Uma coluna representa um tipo de dados em uma tabela; no exemplo, o cargo de todos os funcionários. A ordem das colunas não é importante durante o armazenamento de dados; especifique essa ordem quando os dados forem recuperados.
4. Uma coluna que contém o número do departamento, que é também uma *chave estrangeira*. Uma chave estrangeira é uma coluna que define como as tabelas se relacionam umas com as outras. Uma chave estrangeira se refere a uma chave primária ou a uma chave exclusiva em outra tabela. No exemplo, DEPTNO

identifica com exclusividade um departamento da tabela DEPT.