



Banco de dados

Revisão



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

Banco de Dados

2

**Entidades e
Relacionamento**

3

Dados

4

DDL X DML

5

CRUD

6

**Funções e
Joins**

7

**Stored
Procedures**



1. Banco de Dados



1. O que são bancos de dados?

De maneira conceitual...

- **Banco de dados** é o agrupamento de dados que tratam do mesmo assunto, e que precisam ser armazenados para segurança ou conferência futura.
- Um banco de dados pressupõe a possibilidade armazenar **grandes quantidades de dados de forma organizada.**
- **Tipos de Bancos de Dados:** Relacionais e Não-Relacionais





1.1. Relembrando...

MySQL

MySQL é um sistema de gerenciamento de bancos de dados relacional, considerado o banco de dados de código aberto mais popular do mundo.



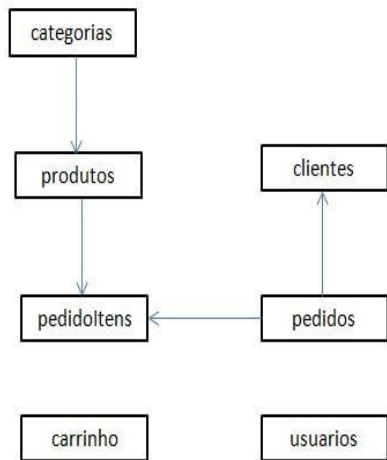
SQL

SQL (Structured Query Language) é a linguagem padrão universal para manipular bancos de dados relacionais através dos SGBDs.





1.2. Modelagem de Dados



A **modelagem de dados** é uma técnica usada para a especificação das regras de negócios e as estruturas de dados de um banco de dados. Serve como uma etapa para projetar o banco.



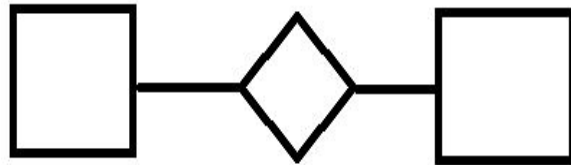


1.3. Tipos de Modelagem

Conceitual: Desenho do conceito ou idéia;

Lógico: Definição das regras de negócio;

Físico: Criação do banco de dados diretamente no SGBD.





1.4 Resumo

1. Entre os componentes de um modelo conceitual, podemos relacionar:
 - Entidades;
 - Atributos;
 - Relacionamentos;
2. Modelagem lógica serve para a coleta de informações sobre as necessidades de negócios e não envolve projetar um banco de dados.
3. Modelagem física é necessária para o projeto real do banco de dados.



2. Entidades e Relacionamentos.



2. Entidade

Entidade é um objeto existente no mundo real, com uma identificação distinta e significado próprio. São os objetos que existem no negócio, ou ainda, que descrevem o negócio em si. Se algo existe e proporciona algum interesse em manter dados sobre ele, isto se caracteriza como uma Entidade do negócio.

FILMES

ATORES

GÊNEROS

**Para o nome dessas tabelas,
devemos usar substantivos no
plural.**



2.1 Atributos

São as características que definirão cada entidade. Por exemplo, a entidade “Filmes” pode ter os seguintes atributos:

FILMES

titulo
avaliacao
data_lancament
o
premios

Se o nome do atributo for mais de uma palavra, nenhum espaço será colocado. Para isso, você pode usar, por exemplo, “-”. Também deve-se evitar acentos.





2.2 Chave Primária

Uma **chave primária** ou **Primary key** é um campo que identifica exclusivamente cada linha de uma tabela. Em outras palavras, não pode haver duas linhas em uma tabela que tenha a mesma estação.

id	título	avaliacao	data_lancamento	premios
1001	Pulp Fiction	9.8	1995-02-16	10
1002	Kill Bill Vol. 1	9.5	2003-11-27	25

Nesse caso os id dos filmes
não podem se repetir

Para identificar a chave primária em uma entidade, podemos escrever o atributo em negrito seguido pelas iniciais **PK** entre parênteses.

FILMES

id (PK)

título

avaliacao

data_lancamento

premio

3. Dados



3. Dados

- Os dados ou atributos de cada registro em uma tabela devem ser de um tipo de dados específico.
- Quando projetamos um banco de dados, temos que pensar sobre que tipo de dados é útil para nosso modelo.
- Cada tipo de dados tem um certo tamanho e quanto mais precisão colocarmos nesse tipo de dado, mais rápido e com melhor desempenho o MySQL funcionará.

012345

Tipos numéricos

2020-05-09

Tipos de data

Juan Perez

Tipos de texto





3.1. Tipos de Dados

O MySQL suporta os seguintes tipos de dados:

- Numéricos Inteiros;
- Numéricos Decimais;
- Texto (strings);
- Booleanos;
- Data;



4. DDL X DML



4. DDL X DML

Apesar do MySQL ser uma única linguagem , ela é dividida em tipos de acordo com a funcionalidade dos comandos:

DDL - Data Definition Language ou Linguagem de Definição de Dados. Apesar do conceito, não interage com os dados, mas com os objetos do banco.

Os comandos DDL são: CREATE, ALTER e DROP.

DML -, Data Manipulation Language ou Linguagem de Manipulação de Dados. Como o próprio conceito sugere, interage com os dados.

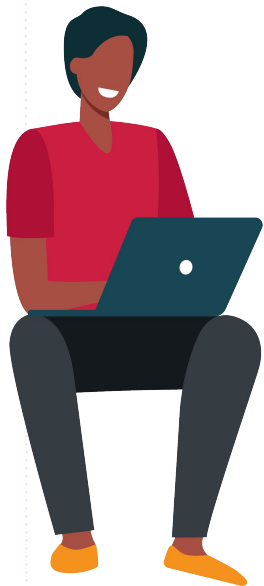
Os comandos DML são: SELECT, INSERT, UPDATE e DELETE



5. CRUD



5. CRUD



Durante o curso, aprendemos a criar, alterar e excluir base de dados e tabelas, utilizando os comandos DDL, que são: **CREATE, ALTER E DROP.**

Quando falamos de **CRUD**, estamos nos referindo às 4 operações básicas de manipulação de dados, ou seja, comandos DML, que são:

Create (Criar) - INSERT
Read (Ler) - SELECT
Uppdate (Atualizar) - UPDATE
Delete (Excluir) - DELETE

Vamos lembrar?





5.1 Relembrando...

Utilizamos o comando **SELECT** para realizar uma consulta no banco de dados:

SINTAXE: SELECT `CAMPO` FROM `TABELA`

Para inserir dados em uma tabela, utilizamos o comando **INSERT:**

SINTAXE: INSERT INTO `TABELA` (campo1, campo2,...) VALUES (valor1,valor2...)

Para atualizarmos algum campo de uma tabela, utilizamos o comando **UPDATE:**

SINTAXE: Update `tabela` SET `campo` = novo_valor WHERE `campo` = `valor_antigo`

Finalmente, para excluirmos um dado ou registro de uma tabela, utilizamos o comando **DELETE:**

SINTAXE: DELETE FROM `tabela` WHERE `campo`= valor.



6. Funções e Joins



6. Funções

Para manipularmos os dados de uma tabela de forma mais performática e assertiva, utilizamos alguns recursos que facilitam nossa consulta. São eles:

- Operadores: <, >, =, <=, >=, <>. IS NULL, BETWEEN, IN, LIKE;
- Para ordenar os dados, utilizamos o Order By;
- Funções básicas: SUM, CONCAT, COALESCE, DATEDIFF, EXTRACT, REPLACE, CASE +, -, *, /, etc...;
- Format.





6.1 JOINS

São usados em uma instrução SQL para recuperar dados de várias tabelas ao mesmo tempo.

Até aqui, trabalhamos as junções INNER JOIN, LEFT JOIN e RIGHT JOIN.

Vamos relembrar?

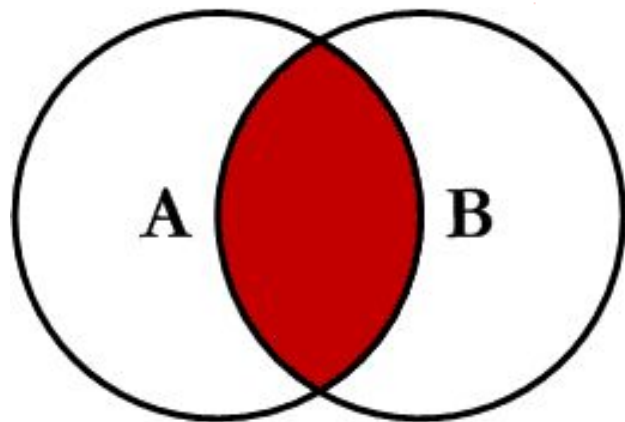




INNER JOIN

Utiliza-se INNER JOIN quando queremos recuperar os dados comuns entre duas ou mais tabelas.

Ex. Temos uma tabela **Clientes** e uma tabela **Produtos**. Queremos saber os produtos que este cliente comprou.

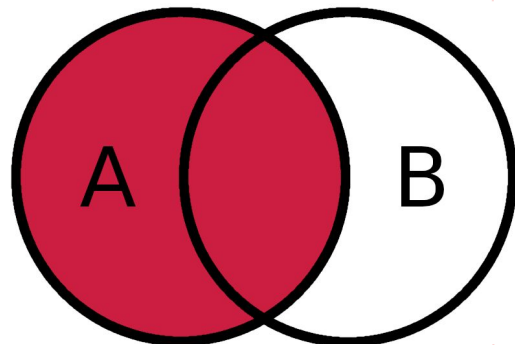




LEFT JOIN

Retorna todos os registros da primeira tabela, mesmo quando os registros não atendem à condição indicada na cláusula ON.

Ex. Temos uma tabela **Clientes** e uma tabela **Credito**. Queremos listar todos os clientes e seus créditos, mesmo que o cliente não tenha crédito.

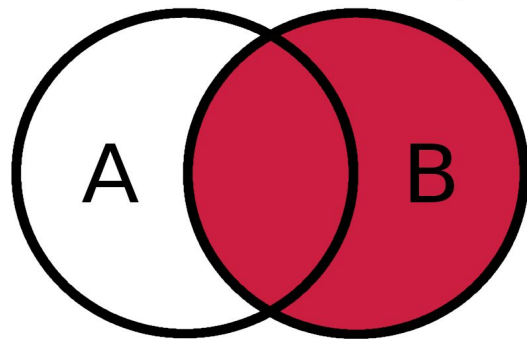




RIGHT JOIN

Retorna todos os registros da segunda tabela, mesmo quando os registros não atendem à condição indicada na cláusula ON.

Ex. Temos uma tabela **Produtos** e uma tabela **Categoria**. Queremos listar todas as categorias e os produtos relacionados, mesmo que a categoria não tenha produto relacionado.



7. Stored Procedures



7. Stored Procedures

Stored procedures são rotinas definidas no banco de dados, identificadas por um nome pelo qual podem ser invocadas. Um procedimento desses pode executar uma série de instruções, receber parâmetros e retornar valores.

As cláusulas presentes em uma estrutura básica são:

- **CREATE PROCEDURE** - Cria procedure;
- **BEGIN** - Marca o início da procedure;
- **END** - Marca o fim da procedure;
- **DELIMITER** - Delimita começo e fim de procedure;





Procedures com Parâmetros.

Nas procedures podemos fazer uso de parâmetros para a execução. Os “parâmetros” são opcionais e, caso não sejam necessários, devem permanecer apenas os parênteses vazios na declaração do procedure.

Para que um procedimento receba parâmetros, é necessário seguir a seguinte sintaxe, dentro dos parênteses: (MODO nome TIPO).

O “MODO” indica a forma como o parâmetro será tratado no procedimento, se será apenas um dado de entrada, apenas de saída ou se terá ambas as funções.

O TIPO é o tipo de dado do parâmetro (int, varchar, decimal, etc).



DigitalHouse>
Coding School