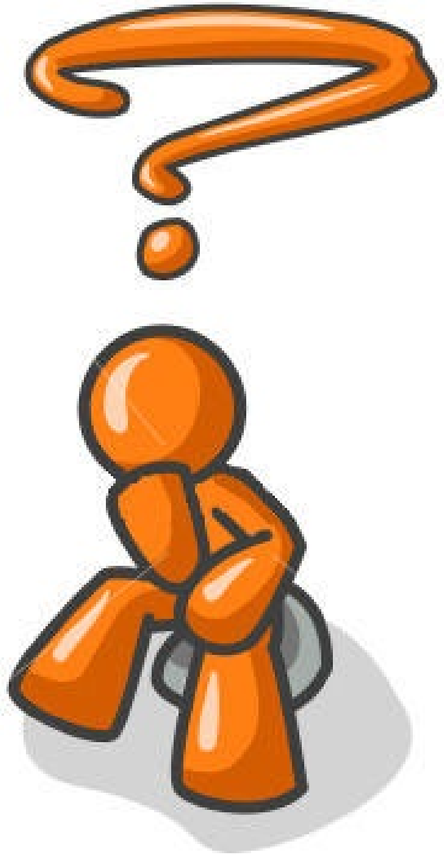
SENAI

Desenvolvimento de Sistemas

Banco de Dados

Conceitos Fundamentais e Implementação Prática

O que é Banco de Dados?



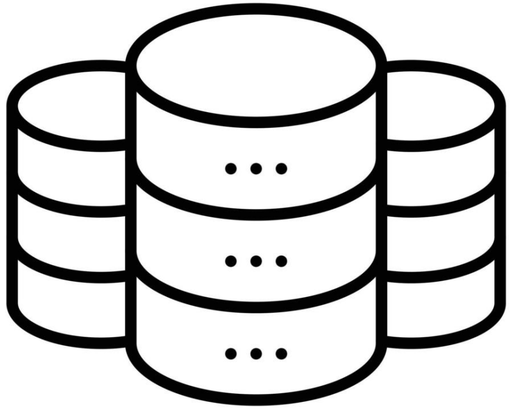
Banco de Dados

O que é um Banco de Dados?

Um banco de dados é uma coleção organizada de informações estruturadas, armazenadas eletronicamente e acessíveis de várias maneiras. Os bancos de dados permitem armazenar, gerenciar e recuperar dados de forma eficiente.

O que é um banco de dados relacional?

- Coleção de informações;
- Todos os dados de um banco de dados **relacional** são armazenados em **tabelas;**
- As **linhas** de uma tabela representam **registros;**
- As **colunas** de uma tabelas representam **campos.**



Sistemas Gerenciadores de Banco de Dados - SGDB

Estrutura Cliente/Servidor

Estrutura Cliente/Servidor

A maioria dos SGBDs opera em uma arquitetura cliente/servidor:

- **Servidor**: Armazena os dados e processa as consultas
- **Cliente**: Interface ou aplicação que envia solicitações ao servidor

Modelo Conceitual

MER – Modelo Entidade Relacionamento

(É um modelo conceitual que representa a estrutura dos dados de uma aplicação.)

Exemplos:

Para um sistema de biblioteca:

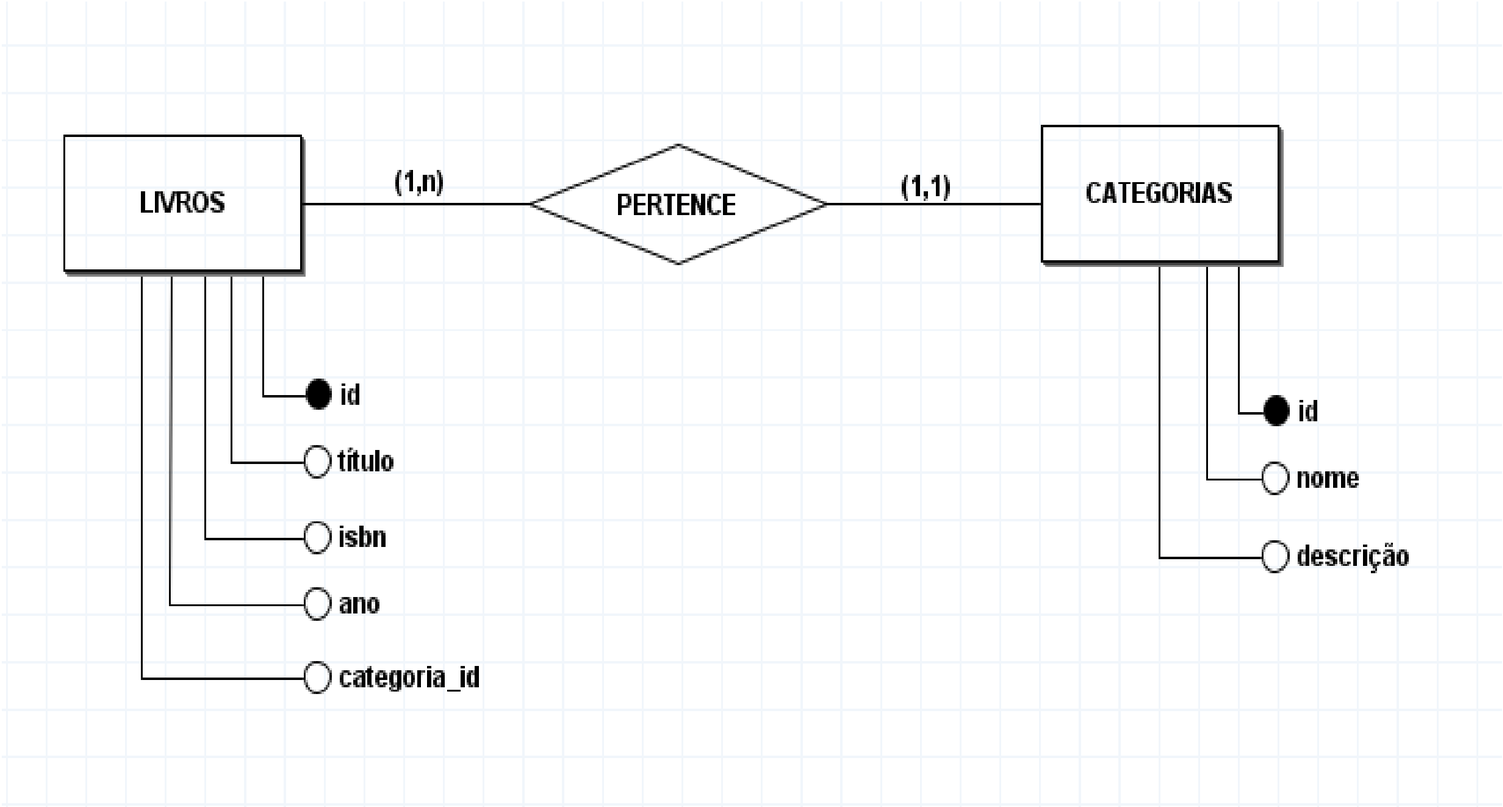
LIVRO = Id (número inteiro), Título (texto pequeno), ISBN (texto pequeno), Ano (número inteiro), Id da Categoria (número inteiro)

CATEGORIA = Id (número inteiro), Nome (texto pequeno), Descrição (texto médio)

Modelo Conceitual

DER – Diagrama Entidade Relacionamento

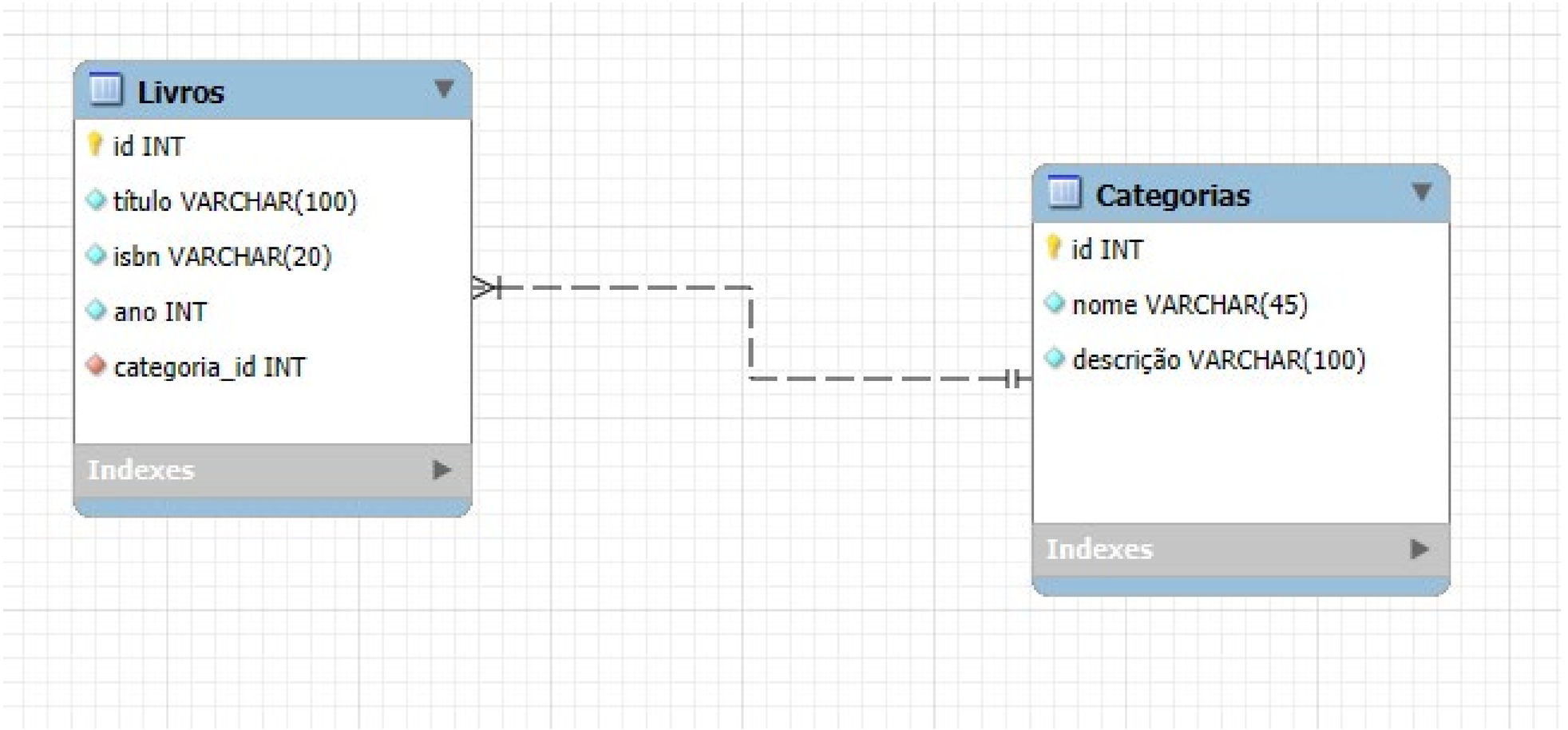
O DER é uma representação gráfica do Modelo Entidade-Relacionamento, utilizando símbolos e notações específicas.



- Um livro pertence a exatamente uma categoria e uma categoria pode ter vários livros.

Modelo Lógico

O modelo lógico representa os dados de forma mais detalhada, especificando tabelas, colunas, tipos de dados e relacionamentos.



SENAI

Desenvolvimento de Sistemas

Implementar Banco de Dados

Modelo Físico

O modelo físico é a implementação real do banco de dados, incluindo definições de tabelas, índices, chaves e constraints em SQL.

Tipo de Dados

Números Inteiros

- **TINYINT**: Número inteiro pequeno (-128 a 127)
- **SMALLINT**: Número inteiro médio (-32.768 a 32.767)
- **INT**: Número inteiro padrão
- **BIGINT**: Número inteiro grande

Números Decimais

- **DECIMAL(p,s)**: Número com precisão definida (p dígitos no total, s após a vírgula)
- **FLOAT**: Número de ponto flutuante de precisão simples
- **DOUBLE**: Número de ponto flutuante de precisão dupla

Texto

- **CHAR(n)**: String de tamanho fixo (n caracteres)
- **VARCHAR(n)**: String de tamanho variável (máximo de n caracteres)
- **TEXT**: Texto longo (até 65.535 caracteres)

Datas e Horas

- **DATE**: Data no formato YYYY-MM-DD
- **TIME**: Hora no formato HH:MM
- **DATETIME**: Data e hora no formato YYYY-MM-DD HH:MM

Outros

- **ENUM**: Lista de valores predefinidos do tipo string
- **BOOLEAN**: Valor verdadeiro/falso (1/0)

SQL – Structured Query Language

SQL é a linguagem padrão para interagir com bancos de dados relacionais. Ela permite criar, ler, atualizar e excluir dados, além de definir estruturas de banco de dados.

DDL (Data Definition Language)

Comandos para definir e modificar estruturas de banco de dados:

- **CREATE:** Cria um banco de dados, tabela, índice, etc.
- **ALTER:** Modifica uma estrutura existente
- **DROP:** Remove uma estrutura existente
- **TRUNCATE:** Remove todos os registros de uma tabela

Comandos SQL

Exemplo de criação.

- Criar banco de dados:

```
CREATE DATABASE biblioteca CHARACTER SET utf8mb4;
```

- Visualizar banco de dados:

```
SHOW DATABASES;
```

Modelo Físico (SQL)

```sql
CREATE TABLE livros (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    titulo VARCHAR(100) NOT NULL,
    isbn VARCHAR(20) NOT NULL,
    ano INT NOT NULL,
    categoria_id INT NOT NULL
);

CREATE TABLE categorias (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(45) NOT NULL,
    descricao VARCHAR(100)
);

-- Criando relação entre tabelas com chave estrangeira
ALTER TABLE livros
ADD CONSTRAINT fk_livros_categorias
FOREIGN KEY (categoria_id) REFERENCES categorias(id);
```

SQL – Structured Query Language

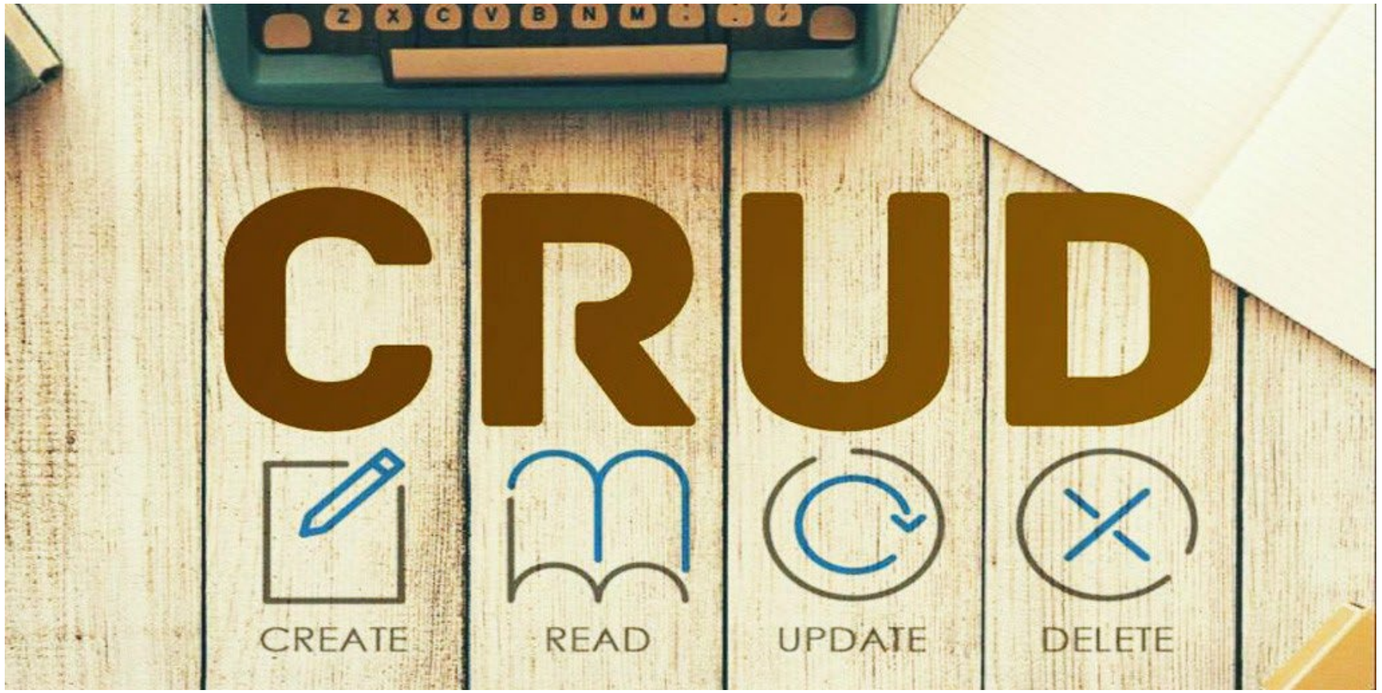
DML (Data Manipulation Language)

Comandos para manipular dados dentro das tabelas:

- **INSERT:** Insere novos registros
- **UPDATE:** Atualiza registros existentes
- **DELETE:** Remove registros
- **SELECT:** Recupera registros (às vezes classificado como DQL - Data Query Language)

O que é CRUD?

CRUD é um acrônimo para as **quatro operações básicas** que podem ser realizadas em um banco de dados:



CREATE (C) - INSERT

Comando responsável por inserir dados em uma tabela

```sql
-- Inserindo categorias
INSERT INTO categorias (nome, descricao)
VALUES ('Ficção Científica', 'Livros que exploram conceitos científicos avançados');

INSERT INTO categorias (nome, descricao)
VALUES ('Romance', 'Narrativas centradas em relações amorosas');

-- Inserindo livros
INSERT INTO livros (titulo, isbn, ano, categoria_id)
VALUES ('Fundação', '9788576572664', 1951, 1);

INSERT INTO livros (titulo, isbn, ano, categoria_id)
VALUES ('Orgulho e Preconceito', '9788544001820', 1813, 2);
```

READ (R) - SELECT

Principal comando SQL, utilizado para tarefas de consulta de dados no banco.

```sql
-- Selecionar todos os livros
SELECT * FROM livros;

-- Selecionar livros com informações de categoria
SELECT
    l.id, l.titulo, l.ano, c.nome AS categoria
FROM
    livros l
JOIN
    categorias c ON l.categoria_id = c.id
WHERE
    l.ano > 1900;
```

UPDATE (U)

Comando para alterar valores de registros existentes:

```sql
-- Atualizando o ano de um livro
UPDATE livros
SET ano = 1952
WHERE id = 1;
```

DELETE (D)

Comando responsável por excluir um ou mais registros de uma tabela.

```sql
-- Removendo um livro específico
DELETE FROM livros
WHERE id = 1;
```

Recursos Avançados em SQL

Views

Uma VIEW é uma tabela virtual baseada no resultado de uma consulta SQL. Elas são usadas para:

- Simplificar consultas complexas
- Fornecer uma camada de abstração
- Implementar segurança de dados

Exemplo:

```sql
-- Criando uma view que mostra livros com suas categorias
CREATE VIEW vw_livros_categorias AS
SELECT
    l.id, l.titulo, l.isbn, l.ano, c.nome AS categoria, c.descricao AS categoria_descrica
FROM
    livros l
JOIN
    categorias c ON l.categoria_id = c.id;

-- Usando a view
SELECT * FROM vw_livros_categorias WHERE ano > 1900;
```

Recursos Avançados em SQL

Stored Procedures

Procedimentos armazenados são conjuntos de instruções SQL que podem ser executados como uma única unidade.
Obs: Parecido com uma função em qualquer linguagem de programação.

Exemplo:

```
-- Procedure para adicionar um novo livro
DELIMITER //
CREATE PROCEDURE sp_adicionar_livro(
    IN p_titulo VARCHAR(100),
    IN p_isbn VARCHAR(20),
    IN p_ano INT,
    IN p_categoria_id INT
)
BEGIN
    INSERT INTO livros (titulo, isbn, ano, categoria_id)
    VALUES (p_titulo, p_isbn, p_ano, p_categoria_id);

    SELECT 'Livro adicionado com sucesso!' AS mensagem;
END //
DELIMITER ;

-- Executando a procedure
CALL sp_adicionar_livro('Neuromancer', '9788576570493', 1984, 1);
```

Recursos Avançados em SQL

Triggers

Triggers são blocos de código SQL que são executados automaticamente em resposta a eventos específicos (INSERT, UPDATE, DELETE) em uma tabela.

Exemplo:

```
-- Trigger para registrar alterações em livros
DELIMITER //
CREATE TRIGGER tr_log_alteracoes_livros
AFTER UPDATE ON livros
FOR EACH ROW
BEGIN
    INSERT INTO log_alteracoes (tabela, id_registro, campo_alterado, valor_antigo, valor_
    VALUES (
        'livros',
        NEW.id,
        CASE
            WHEN OLD.titulo != NEW.titulo THEN 'titulo'
            WHEN OLD.ano != NEW.ano THEN 'ano'
            ELSE 'outro'
        END,
        CASE
            WHEN OLD.titulo != NEW.titulo THEN OLD.titulo
            WHEN OLD.ano != NEW.ano THEN CAST(OLD.ano AS CHAR)
            ELSE ''
        END,
        CASE
            WHEN OLD.titulo != NEW.titulo THEN NEW.titulo
            WHEN OLD.ano != NEW.ano THEN CAST(NEW.ano AS CHAR)
            ELSE ''
        END,
        NOW()
    );
END //
DELIMITER ;
```

Exercícios:

Hora de praticar!!


SENAI
DEPARTAMENTO REGIONAL
DE SÃO PAULO
www.sp.senai.br