

FIC 2 – Módulo III

Aula 2 – Princípios de banco de dados e Sqlite

Persistência de dados com
Sequelize.
Antonio Izo Júnior

Sumário

- Princípios de banco de dados.
- Relacionamento entre tabelas.
- Banco de dados Sqlite.

Princípio de banco de dados

É importante entendermos alguns pontos na criação de um banco de dados (database).

- Um banco de dados é formado por um conjunto de tabelas.
- Cada tabela armazena dados sobre um determinado assunto, como, por exemplo, marcas ou modelos de automóveis.
- Uma tabela tem registros (linhas) e campos (colunas).



Princípio de banco de dados

- Os campos podem ter diferentes tipos de dados, como texto, números e datas.

Campos da
tabela.

Exemplo de uma tabela **Funcionário**.

MATRICULA	NOME	NASCIMENTO	TELEFONE
1234	ANA	12-04-1978	01219219
1235	IVO	01-12-2000	07280921
1236	OTO	07-02-1987	06924324

Registros da tabela

Princípio de banco de dados

- No exemplo anterior, vimos que a tabela **funcionário** possui um conjunto de atributos (campos) que são específicos dela.
- Em um banco de dados, criamos várias tabelas.
- Agora vamos criar a tabela **setor com alguns atributos**.

Princípio de banco de dados

Tabela setor.

IDSETOR	NOME	RAMAL	E-MAIL
1	FINANCEIRO	4254	FINANCEIRO@EMPRESA.COM
2	PORTARIA	4253	PORTARIA@EMPRESA.COM
3	SECRETARIA	4237	SECRETARIA@EMPRESA.COM

Repare que o primeiro campo da tabela possui um código que não se repete.

Princípio de banco de dados

Toda tabela deve possuir um campo que possua dados que não podem se repetir em nenhum registro dela.
Tais campos são chamados de **chave primária**.

IDSETOR	
1	← CHAVE PRIMÁRIA.
2	
3	

A **chave primária** evita **duplicidade de registro** nas tabelas.

Relacionamento entre tabelas

- As tabelas de um banco de dados precisam se comunicar para gerar informações de retorno.
- Vamos imaginar o seguinte cenário:
- A empresa precisa saber quais são os funcionários que trabalham em determinado setor.
- Com o exemplo que fizemos, conseguimos extrair tal informação?

Relacionamento entre tabelas

- Infelizmente não. Nas tabelas “setor” e “funcionário” não possuíamos a informação dos setores em que o funcionário trabalha.

MATRICULA	NOME	NASCIMENTO	TELEFONE
1234	ANA	12-04-1978	01219219
1235	IVO	01-12-2000	07280921
1236	OTO	07-02-1987	06924324

IDSETOR	NOME	RAMAL	E-MAIL
1	FINANCEIRO	4254	FINANCEIRO@EMPRESA.COM
2	PORTARIA	4253	PORTARIA@EMPRESA.COM
3	SECRETARIA	4237	SECRETARIA@EMPRESA.COM

Relacionamento entre tabelas

- Podemos resolver o problema de forma rápida, incluindo a chave primária de uma tabela na outra.
- Chamamos esse processo de **chave estrangeira**.
- Para sabermos qual será a tabela que receberá a chave estrangeira, precisamos identificar qual será a tabela que receberá menos duplicidade de registros.
- Vamos verificar como ficaria nos casos das tabelas “funcionário” e “setor”.

Relacionamento entre tabelas

MATRICULA	IDSETOR	NOME	NASCIMENTO	TELEFONE
1234	1	ANA	12-04-1978	01219219
1235	3	IVO	01-12-2000	07280921
1236	2	OTO	07-02-1987	06924324

IDSETOR	MATRICULA	NOME	RAMAL	E-MAIL
1	1234	FINANCEIRO	4254	FINANCEIRO@EMPRESA.COM
2	1236	PORTARIA	4253	PORTARIA@EMPRESA.COM
3	1235	SECRETARIA	4237	SECRETARIA@EMPRESA.COM



Relacionamento entre tabelas

- Repare que, nas duas tabelas, o relacionamento aparentemente está perfeito.
- Mas, e se a empresa contratar um outro funcionário para o setor financeiro? Vamos ver como ficaria?

Relacionamento entre tabelas

MATRICULA	IDSETOR	NOME	NASCIMENTO	TELEFONE
1234	1	ANA	12-04-1978	01219219
1235	3	IVO	01-12-2000	07280921
1236	2	OTO	07-02-1987	06924324
1237	1	CARINA	09-09-1990	02932176

IDSETOR	MATRICULA	NOME	RAMAL	E-MAIL
1	1234	FINANCEIRO	4254	FINANCEIRO@EMPRESA.COM
2	1236	PORTARIA	4253	PORTARIA@EMPRESA.COM
3	1235	SECRETARIA	4237	SECRETARIA@EMPRESA.COM
1	1237	FINANCEIRO	4254	FINANCEIRO@EMPRESA.COM

Duplicidade de registros que são chaves primárias.

Relacionamento entre tabelas

- No caso das tabelas “setor” e “funcionário”, vimos que a tabela “setor” não pode receber a chave estrangeira da tabela “funcionário”, pois será gerada uma duplicidade de chave primária com o campo **IDSETOR**.
- Assim, a tabela “funcionário” receberá o campo **IDSETOR** como **chave estrangeira**.

Banco de dados Sqlite

- O SQLite, ou também chamado de banco de dados **incorporado**, é uma biblioteca, feita na linguagem C e de código aberto, que armazena seus registros em um único arquivo.
- Ao contrário dos Sistemas de Gerenciamento de Banco de Dados (SGBDs) tradicionais que precisam, obrigatoriamente, de uma arquitetura cliente-servidor.
- Por sua praticidade e simplicidade, o SQLite está presente em **aparelhos de televisão, aplicações mobile, desktop e web.**



Banco de dados Sqlite

- Grandes empresas utilizam o SQLite, entre elas estão:
 - **Google:** usa o SQLite em seu sistema operacional Android e no Chrome, seu navegador Web;
 - **Microsoft:** usa o SQLite como componente principal do Windows 10 e em outros produtos;
 - **McAfee:** usa SQLite em seus programas de antivírus.

Banco de dados Sqlite

- As principais vantagens no uso do SQLite são:
 - É estável, multiplataforma e compatível com versões anteriores;
 - Seu código é de domínio público e gratuito, ou seja, todos podem utilizar;
 - Não necessita de instalação ou configuração;
 - Guarda o banco de dados em um único arquivo.

Na próxima aula, vamos praticar a criação de banco de dados.

Até lá...

Referências Bibliográficas

- ZHAO, Alice. **SQL Guia Prático: Um guia para o uso de SQL**. Editora Novatec, 2023.
- NEWMAN, Chris. **SQLite**. 1ª. Editora Sams, 2004.





**INSTITUTO
FEDERAL**
Espírito Santo