

Sequelize ORM para NodeJs

<LAB365>

SENAI

AGENDA

- Revisão de Banco de Dados Relacional
- Introdução a ORM
- Sequelize para Express

SQL : DDL & DML

<LAB365>

SENAI

Tipo de Datos

TEXTOS

- CHAR
- VARCHAR
- TEXT
- ENUM

NÚMEROS

- INT
- DECIMAL, NUMERIC
- FLOAT, REAL

DATAS

- DATE
- DATETIME
- TIMESTAMP

OUTROS

- BOOLEAN

- A linguagem SQL pode ser dividida em alguns subconjuntos.
- A linguagem de **definição de dados (Data Definition Language - DDL)** engloba os comandos declarativos para criação, modificação e exclusão das tabelas.
- Já a linguagem de **manipulação de dados (Data Manipulation Language - DML)** é composta pelos comandos declarativos para que os usuários realizem consultas, insiram, modifiquem e excluam linhas (tuplas).

```
create table departamento (  
    id bigserial primary key,  
    nome varchar(40) not null  
);  
  
create table funcionario (  
    id bigserial primary key,  
    nome varchar(30) not null,  
    login varchar(20) not null unique,  
    departamento_id bigint,  
    foreign key (departamento_id) references departamento (id)  
);
```

```
• CREATE TABLE Produto (  
  ID SERIAL PRIMARY KEY,  
  Nome VARCHAR(100),  
  Preço DECIMAL(10,2),  
  Quantidade INT,  
  Status VARCHAR(20),  
  CHECK (Preço >= 0),    -- Restrição: O preço não pode ser negativo  
  CHECK (Quantidade >= 0 AND Quantidade <= 100),  -- Restrição: A quantidade deve estar entre 0 e 100  
  CHECK (Status IN ('Ativo', 'Inativo'))  -- Restrição: O status deve ser 'Ativo' ou 'Inativo'  
);
```

Introdução a ORM

<LAB365>

SENAI

O que é ORM?

Object Relational Mapper (ORM) é uma técnica de mapeamento de objeto relacional que permite fazer uma relação dos objetos com os dados que os mesmos representam.

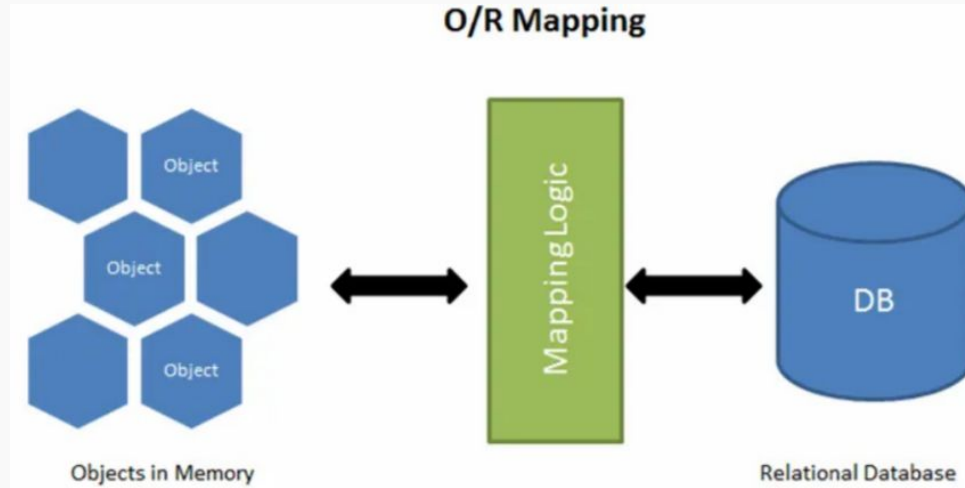
ORM é uma técnica de programação que visa simplificar o desenvolvimento de aplicativos ao mapear objetos de um sistema orientado a objetos para representações de dados em um banco de dados relacional.

Para que serve o ORM?

Os **ORMs** oferecem uma maneira de trabalhar com banco de dados relacionais usando linguagens de programação orientadas a objetos, como JavaScript, Python, Java, etc. Eles **abstraem a complexidade de escrever consultas SQL diretamente**, permitindo que os desenvolvedores interajam com o banco de dados usando objetos e métodos familiares **em suas linguagens de programação**.

“Os ORMs abstraem a complexidade de escrever consultas SQL diretamente em suas linguagens de programação.”

Para que serve o ORM?



Para que serve o ORM?

- **Abstração do Banco de Dados:** Os ORMs abstraem os detalhes específicos do banco de dados, como sintaxe SQL, tipos de dados, etc., permitindo que os desenvolvedores se concentrem mais na lógica de negócios de suas aplicações.
- **Produtividade:** Com ORMs, os desenvolvedores podem escrever menos código para realizar operações com banco de dados, economizando tempo e esforço. Isso geralmente resulta em desenvolvimento mais rápido e iterativo.

Para que serve o ORM?

- **Portabilidade:** Os ORMs oferecem uma camada de abstração entre a aplicação e o banco de dados subjacente, tornando mais fácil a migração entre diferentes sistemas de banco de dados sem alterar muito o código da aplicação.
- **Manutenção:** ORMs simplificam a manutenção do código, pois fornecem uma camada de abstração que facilita a compreensão e a modificação do código em comparação com consultas SQL embutidas diretamente no código.

Para que serve o ORM?

- **Segurança:** ORMs geralmente oferecem recursos integrados para prevenir ataques de injeção de SQL, ajudando a proteger contra vulnerabilidades de segurança comuns.

Para que serve o ORM?

Os mais famosos para o NodeJS são **Sequelize**, KNEX, TypeORM e **Prisma**. Existem diversos ORMs para outras linguagens também.

<https://sequelize.org/>

<https://knexjs.org/>

<https://typeorm.io/#/>

<https://www.prisma.io/>



Sequelize para Express

<LAB365>

SENAI

- É um **ORM** baseado em Promise para Node.js, e suporta os dialetos PostgreSQL, MySQL, MariaDB e SQLite e recursos a transação, relacionamentos, replicação de leitura e etc.



- Possui um mecanismo de **migração** muito poderoso que pode transformar um esquema existente de banco de dados em uma nova versão e também fornece mecanismos de **sincronização de banco de dados** que podem criar estrutura de banco de dados especificando a estrutura do modelo.



INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



Prática!

VScode!

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





OBRIGADO!

<LAB365>