

Exercício 1

Ao desenvolver um novo banco de dados é comum utilizar o Modelo Relacional pois esse, apesar de sua simplicidade, é bastante representativo. Defina, com suas palavras, o que é o Modelo Relacional.

Exercício 2

Defina com suas palavras, no que se refere aos SGBD, os conceitos de tabela, atributos, domínio e *tuplas*.

Exercício 3

Qual o conceito de chave primária? Qual a importância de sua utilização? Dê exemplos.

Exercício 4

Qual o conceito de chave estrangeira? Qual sua relação com as chave primária?

Exercício 5

Utilizando a ferramenta *MySQL Workbench*, identifique e crie as tabelas de um modelo relacional (*crie atributos adicionais, quando achar necessário*).

Você irá modelar as entidades necessárias ao registro de operações que envolvem as operações de utilização de Caixas Eletrônicos, considere os seguintes pontos:

- Cartões de Crédito ou Débito possuem um número único e uma operadora (bandeira) como por exemplo Visa/MasterCard/Elo;
- Para cada operação (compra, depósito ou transferência) realizada deve-se registrar a data, a hora e valor envolvido na transação;
- Cada cartão pertence a uma única conta bancária, mas cada conta pode ter vários cartões de diferentes bandeiras;
- Cada cliente tem apenas uma única conta no banco;
- Cada conta é de um único cliente e possui um saldo;

1. Introdução

Esta atividade tem como objetivo praticar os conhecimentos iniciais na tecnologia Node.js

Ao desenvolver as atividades práticas de laboratório, atente sempre em implementar código de maneira organizada, formatada e estruturada: o chamado código limpo. Mantenha seu código sempre de acordo com as *guidelines* ditas pelas boas práticas de programação. Peça orientação a seu professor mediador sempre que tiver dúvidas.

Atividade 01

Descreva como funciona o Event Loop do Node.js, quais vantagens e desvantagens desta abordagem.

Atividade 02

Crie um servidor web em Node.js puro (sem uso de nenhuma biblioteca externa) que satisfaça as seguintes exigências:

1. Ao acessar o endereço principal via GET, ex: <http://localhost:8080/>, deve ser retornado o conteúdo de um arquivo HTML (index.html) lido através da biblioteca fs do Node.js.
2. Caso seja acessado qualquer outro endereço ou qualquer outro método HTTP, deve ser retornado o conteúdo de uma página HTML de erro 404.
3. Os arquivos index.html e 404.html são de livre criação.
4. As leituras aos arquivos devem ser feitas de forma assíncrona