



Data Base Design

Aprenda a modelar banco de dados relacionais

O que é um banco de dados?

- O **banco de dados** é uma entidade utilizada para gerenciar os dados de uma aplicação;
- Nos **DBs** temos as tabelas, que são responsáveis por categorizar os dados;
- Além disso, temos outros elementos importantes, como: **queries**, **views**, **stored procedures** e mais;
- Um **DB bem modelado** representa também o sucesso da aplicação;



Principais tipos de DBs

- **Relacional:** o que iremos utilizar no curso, é o mais comum nas aplicações (MySQL);
- **NoSQL:** DBs não relacionais, baseado em documentos (MongoDB);
- **Object Database:** DB orientado a objetos, integrado nas aplicações (ObjectBox);
- **Cloud Database:** Este DB fica em servidores cloud e tem alta escalabilidade, podemos utilizar SQL ou NoSQL;



O que é uma tabela?

- As tabelas são quem recebe a maioria das **operações** de banco de dados, o famoso **CRUD**;
- Elas **categorizam os dados**, os salvando em linhas;
- Os tipos dos mesmo são separados em colunas, como uma tabela de Excel;
- **Criar corretamente as tabelas** também faz parte do DB Design;
- As tabelas possuem características especiais chamadas **constraints**;



A importância do DB Design

- Realizar um bom planejamento do banco de dados e uma modelagem de dados é crucial para o sucesso da aplicação;
- **Reduz erros** no desenvolvimento da aplicação e do próprio DB;
- Melhora a **performance** da aplicação;
- Ajuda na **comunicação** do time de desenvolvimento;
- Consegue integrar pessoas não-técnicas no projeto, pois passa por várias fases: **conceitual, lógico e físico**;



Softwares para DB Design

- Temos diversos softwares para modelar o banco de dados, online e offline;
- Um bem famoso online é o **Lucidchart**, o problema dos online é que geralmente temos limitações pois são pagos;
- O mais utilizado offline é o **MySQL Workbench**, ele faz tudo que os outros fazem e ainda conseguimos transformar a modelagem em um banco real;
- Além disso, os recursos do software o tornam uma boa opção para uso geral, como fazer **queries** em banco de dados;



Instalando o Workbench

- XAMPP ou WAMPP não precisa instalar o server, veremos isso adiante;
- O server é interessante pois podemos **conectar no servidor do banco** e testar queries nos modelos desenvolvidos;
- Vamos lá!



Diagramas do curso

- Os diagramas estão separados por seções;
- Cada um tem um print screen, facilitando a visualização;
- É o material de apoio durante as atividades e projeto do curso;





Introdução

Conclusão da seção



Modelagem de dados

Introdução da seção

O que é modelagem de dados?

- A **modelagem de dados** faz parte do desenvolvimento do diagrama do banco de dados;
- A dividimos em três etapas: **conceitual, lógica e física**;
- Vamos precisar **entender melhor o que precisamos resolver**, qual software está sendo desenvolvido;
- Levantaremos os requisitos e então começamos as etapas descritas acima;



Tipos de modelagem

- **Conceitual:** modelagem de mais alto nível, basicamente definimos as entidades que estão envolvidas no projeto;
- **Lógica:** Nesta etapa podemos definir as relação entre as entidades, e também os seus atributos;
- **Física:** Aqui implementamos recursos mais técnicos, como: constraints, índices, tipos de dados, triggers e etc;
- Realizaremos todas estas etapas no projeto;



Levantamento de requisitos

- **Levantamento de requisitos** é a etapa onde pessoas técnicas e não técnicas se reúnem para falar sobre o produto/projeto;
- Nesta etapa vamos conseguir fazer duas etapas: **conceitual** e **lógica**;
- Precisamos identificar todas as **entidades** que compõem o sistema;
- E também todos os **dados que precisamos guardar** nestas entidades;
- Geralmente o levantamento é feito por uma **reunião**, no curso teremos uma etapa que vai simular isso;



Nosso estudo de caso

- Nosso projeto consiste em um **e-commerce de livros**;
- Vamos precisar passar por **todo o processo de design de DB**;
- Ou seja, pelas fases: **conceitual, lógica e física**;
- Depois nas últimas seções implementaremos **novas funcionalidades**, como se o projeto estivesse crescendo;
- Vamos ver o que foi definido!



Coral Store

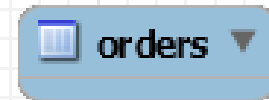
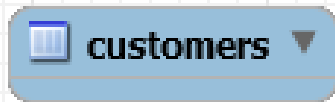
A Coral Store é uma pequena livraria de bairro, o dono nos contratou para desenvolver um e-commerce para sua empresa. A ideia central é captar os dados dos clientes, eles são: nome, email, senha, endereço e telefone. O sistema também precisa possibilitar cadastrar os produtos, como vendemos apenas livros o site a princípio não terá categorias. Os produtos precisam conter: nome, preço, descrição, número de páginas e ISBN. Precisamos também salvar as compras de cada usuário, que deve conter: o valor total e itens comprados. Este será o MVP da Coral Store, se o projeto der certo seremos contratados para expandir o mesmo.

Definindo entidades

- Agora iniciamos a **modelagem conceitual**;
- Vamos precisar ler o nosso documento novamente;
- Identificaremos as **principais entidades** que constituem o projeto que precisamos desenvolver;
- Após este passo criaremos a primeira versão do nosso **diagrama de banco de dados**;
- Vamos lá!



Definindo Entidades



Primary key

- Toda tabela precisa de uma **primary key**;
- Ela precisa ser **única**, ou seja, seu valor não pode se repetir;
- Para resolver esse problema, geralmente é criada uma coluna chamada **id**;
- Onde ela recebe duas características (**Constraints**):
- **Primary Key** – Valor único que não se repete ao longo da tabela como id, cpf, etc
- **Not Null** – Não admite registro vazio;
- Vamos aplicar isso também no nosso projeto!

Definindo atributos

- Agora que temos todas as entidades mapeadas entramos na **fase lógica**;
- Aqui **transformaremos em colunas para as tabelas todos os dados que precisamos salvar** das entidades;
- Isso já torna o nosso projeto muito mais próximo da **versão final**;
- Vamos lá!



Definindo atributos

customers
id INT
name VARCHAR(45)
email VARCHAR(45)
password VARCHAR(45)
address VARCHAR(45)
customerscol VARCHAR(45)
phone VARCHAR(45)
Indexes

products
id INT
name VARCHAR(45)
description VARCHAR(45)
price VARCHAR(45)
page_qty VARCHAR(45)
isbn VARCHAR(45)
Indexes

orders
id INT
customer VARCHAR(45)
items VARCHAR(45)
total VARCHAR(45)
Indexes

Sobre a física

- A próxima etapa seria a **fase física**, correto?
- Porém na verdade esta etapa seria já o banco **completamente desenhado e otimizado** para desenvolver a aplicação;
- Então normalmente é feito o processo de **normalização** antes, que é o vamos aprender nas próximas aulas;
- Com a normalização, teremos a fase física sendo aplicado aos poucos;
- E então poderemos desenvolver nossa aplicação!





Modelagem de dados

Conclusão da seção



Iniciando a normalização

Introdução da seção

O que é normalização?

- **Normalização** é um processo onde melhoramos a qualidade do nosso banco de dados, a fim de deixá-lo mais **otimizado**;
- Temos algumas formas normais para aplicar;
- As principais são três: **NF1**, **NF2** e **NF3**; (normal form)
- Ao longo do curso aplicaremos todas em nosso banco de dados;
- E com isso teremos também a modelagem física sendo concebida;



Forma normal 1

- Cada NF deve seguir um conjunto de requisitos para ser totalmente aplicada;
- As da NF1 são:
- Colunas devem ter apenas um valor em cada célula (atomicidade);
- Cada coluna deve salvar apenas um tipo de dado;
- O nome das colunas deve ser único;
- A ordem das colunas não importa;



Aplicando NF1 em clientes

- Agora vamos ter que **analisar todas as tabelas** e aplicar a normalização;
- É normal que ao aplicar as NFs seja necessário **criar uma nova tabela**;
- Alguns dos pontos a serem observados em customers são: **address**
e name;
- Ambas possuem a possibilidade de inserção de vários dados;
- Vamos lá!



N1 - Customers

addresses
id INT
street VARCHAR(45)
number VARCHAR(45)
neighborhood VARCHAR(45)
city VARCHAR(45)
region VARCHAR(45)
complement VARCHAR(45)
customer VARCHAR(45)
Indexes

customers
id INT
first_name VARCHAR(45)
last_name VARCHAR(45)
email VARCHAR(45)
password VARCHAR(45)
phone VARCHAR(45)
Indexes

products
id INT
name VARCHAR(45)
description VARCHAR(45)
price VARCHAR(45)
page_qty VARCHAR(45)
isbn VARCHAR(45)
Indexes

orders
id INT
customer VARCHAR(45)
items VARCHAR(45)
total VARCHAR(45)
Indexes

Meta:

1 – Dividir nome em first_name e last_name

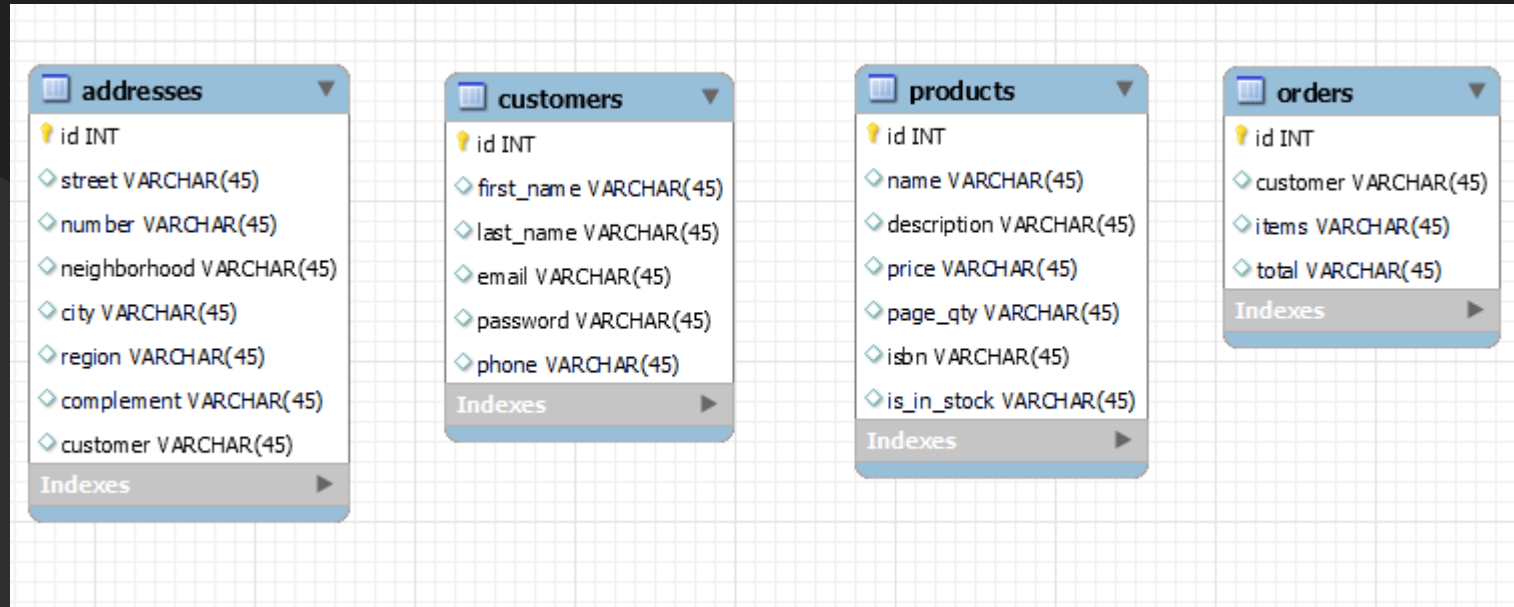
2 – Criar nova tabela addresses para os endereço

Aplicando NF1 em produtos

- Agora vamos analisar a **tabela de produtos**;
- Será que precisamos fazer alguma otimização para a NF1?
- Além disso, surgiu a necessidade de um novo campo, precisamos colocar se o produto **está ou não disponível em estoque**;
- Vamos lá!



N1 - Products



Meta:

1 – Em products criar o item is_in_stock para saber se o produto está em estoque

Aplicando NF1 em pedidos

- Agora vamos analisar a **tabela de pedidos**;
- O campo que precisamos observar é **items**, ele também nos permite adicionar mais um item;
- Vamos lá!



N1 - Orders

Meta:

- 1 – Em orders criar retirar o itens.
- 2- Criar uma nova tabela “ordem_items” para armazenar os itens do pedido, com os campos com os campos id, order, item

