

Faculdade Ruy Barbosa Wyden

Bianca Santos

Gabriel Salazar

Gabriel Souza

Jeferson Jesus

Levi Pires

Lismar Brasileiro

Lucas Rangel

Luis Henrique

Magno Tibiriça

Trabalho de Banco de Dados:

Loja de sandálias

Salvador

2023

Bianca Santos
Gabriel Salazar
Gabriel Souza
Levi Pires
Lismar Brasileiro
Lucas Rangel
Luis Henrique
Magno Tibiriça

**Trabalho de Banco de Dados:
Loja de sandálias**

Trabalho avaliativo da matéria de Banco de dados da Faculdade Ruy Barbosa Wyden como requisito parcial para a conclusão do semestre.

Docente: Heleno Cardoso

Salvador
2023

Sumário

1. Introdução.....	4.
2. Modelo conceitual.....	5.
3. Modelo Lógico.....	8.
4. Modelo Físico.....	11.
5. Conclusão.....	13.

1. Introdução

Este trabalho possui como objetivo desenvolver um banco de dados para ser utilizado por um comércio de sandálias. Possibilitando que todas as informações pertinentes a estoque, valores, sandálias e registro de vendas consigam ser armazenadas, historiadas e acessadas pelos vendedores.

No campo profissional, como desenvolvedor de DB's, a atividade foi feita com intuito de vivenciar e ganhar experiência para criar bancos de dados, saber o uso dos códigos de DDL, DCL, DML, DQL e DTL. Saber usar os modelos conceitual, lógico e físico para construir um banco de dados por etapas e de modo mais completo.

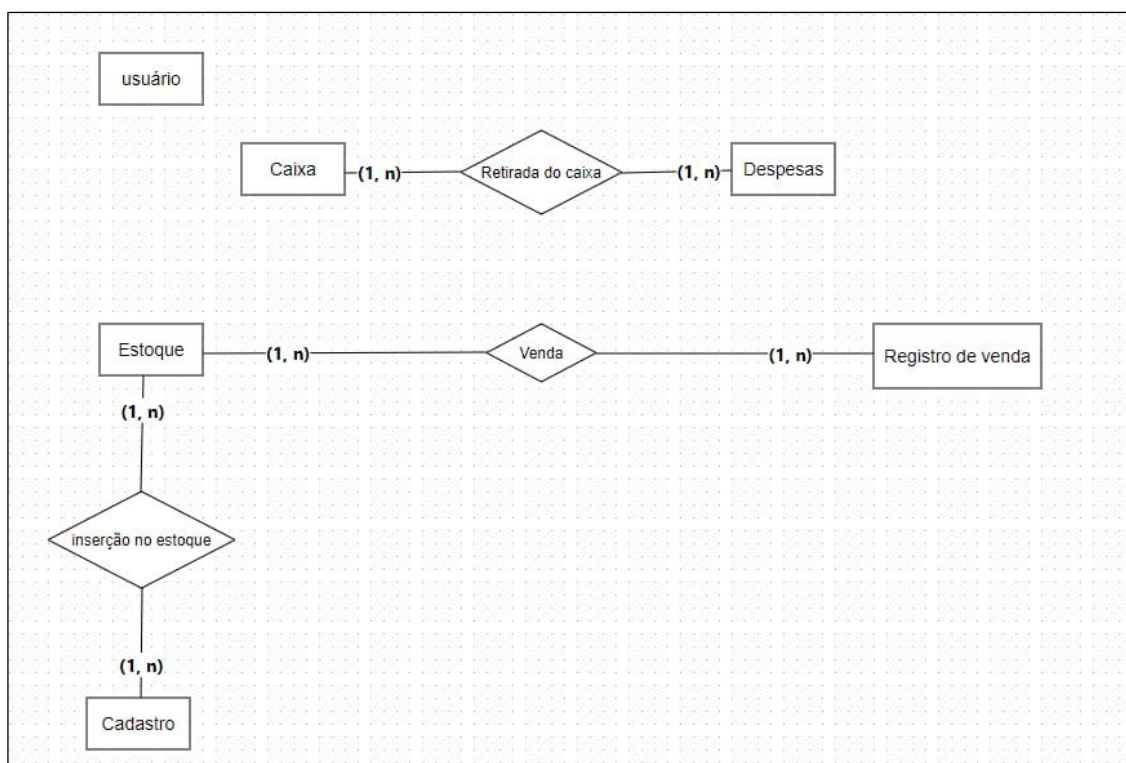
Utilizamos como metodologia iniciando pela elaboração do modelo conceitual para seguir com a etapa do modelo físico e utilizar da engenharia reversa para gerar o modelo lógico.

Nas próximas páginas iremos expor a respeito do modelo conceitual a fim de apresentar as ideologias usadas e como funcionam as cardinalidades, após o modelo conceitual será apresentado o modelo logico onde será exibido um melhor esboço do banco de dados. Após apresentado os tópicos anteriores então será o modelo físico que irá conter os códigos utilizados para a criação das tabelas, inserção dos dados e outras modificações feitas.

2. Modelo conceitual

Elaboramos o projeto de uma loja de sandálias com a finalidade do modelo conceitual de dados é capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio. Um negócio que possua segurança para acesso ao sistema. Os vendedores possuem seus usuários e executam o seu login. Após o login o usuário tem liberdade para cadastrar novas sandálias, inserir no estoque as sandálias que foram recém cadastradas, conferir estoque, dar baixa nas vendas, criar os registros de vendas, notificar valor de caixa, as despesas extras e em caso de necessidade o valor que teve que ser retirado do caixa.

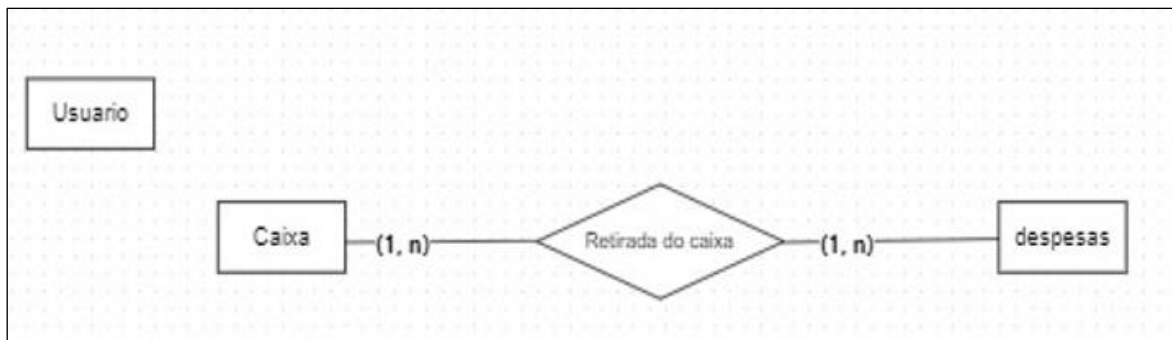
Figura 1 – Diagrama entidade-relacionamento



Fonte: Autoria própria

Essa estrutura do banco de dados representa o modelo conceitual que abrange várias entidades interligadas, cada uma desempenhando um papel crucial no gerenciamento de dados. As entidades são representadas através de um retângulo com o nome da entidade escrito em seu centro e os relacionamentos os losangos.

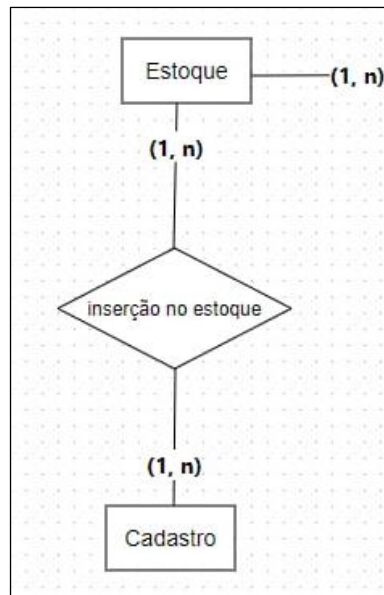
Figura 2 – Recorte da relação de retirada do caixa



Fonte: Autoria Própria

A presença importante da entidade "Caixa", responsável por manter informações sobre o capital disponível no caixa em um determinado dia. A relação entre "Caixa" e "Despesas" é caracterizada como uma retirada do caixa, pois representa o processo de retirada de capital para cobrir despesas específicas. Esta relação de um-para-muitos $(1, n)$ é crucial para rastrear todas as retiradas de caixa associadas a uma única entrada de despesa, garantindo a precisão do registro financeiro.

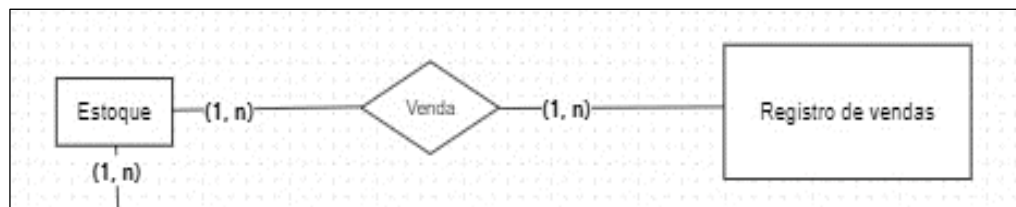
Figura 3 – Relacionamento de inserção no estoque



Fonte: Autoria própria

Destaca-se a entidade "Cadastro", a qual simboliza o momento de registro de um novo modelo de sandália. A relação um-para-muitos $(1, n)$ estabelecida entre "Cadastro" e "Estoque" reflete a inserção de múltiplos modelos de sandálias no estoque após o registro.

Figura 4 – Relacionamento de venda

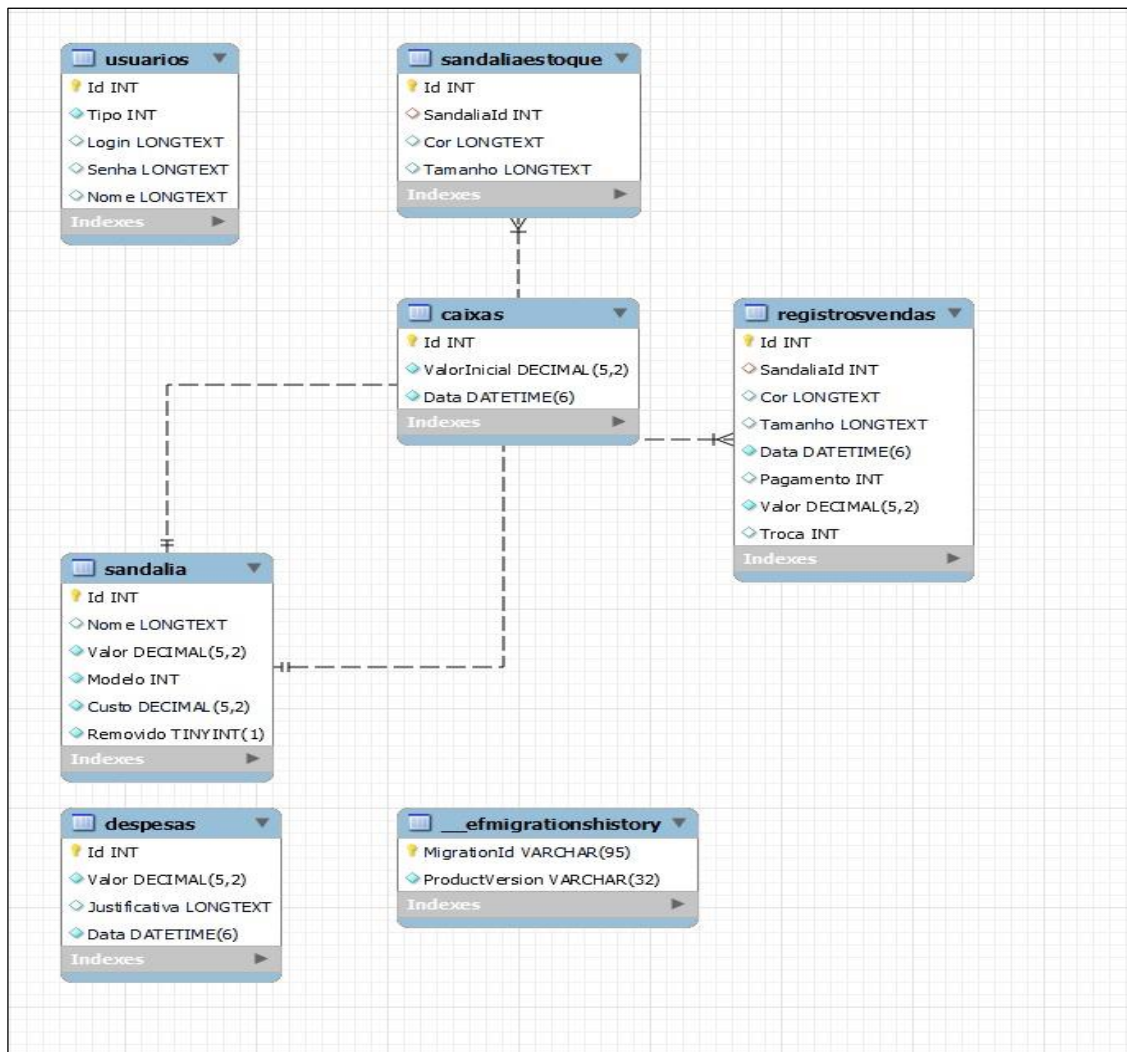


Fonte: Autoria própria

A dinâmica da entidade "Estoque" ganha destaque ao estabelecer uma conexão com a entidade "Registro de Vendas" por meio da relação de vendas. Esta relação, também de um-para-muitos, $(1, n)$. É crucial para registrar cada transação de venda em relação aos itens do estoque, possibilitando um controle preciso das vendas realizadas e dos produtos disponíveis. A relação da pessoa cadastrada com a inserção no estoque também é de um para muitos porque cada cadastro é único, porém a inserção pode acontecer várias vezes e em várias quantidades $(1, n)$.

3. Modelo lógico

Figura 5 – Modelo Lógico



Fonte: MySQL Workbench

Na figura acima é representado o diagrama do modelo lógico, como ele é apresentado no MySQL. Através dele e de suas chaves conseguimos identificar os atributos de cada entidade. Como por exemplo, no nosso modelo de banco de dados de sandálias existem as entidades: Sandália, despesas, caixas, registrosvendas, sandaliaestoque.

A entidade usuários é responsável por guardar os dados referente ao login de cada vendedor da loja, como seu nome, login, senha e o tipo de acesso que ele possui ao banco de dados.

A entidade sandália estoque tem como função reunir todas as sandálias que possuem no estoque e utiliza como foreign key o atributo sandaliasID, que é uma primary key da entidade sandália, para saber quais sandálias possuem no estoque através do id da sandália, além das sandaliasID possuem também cor e tamanho possibilitando o vendedor saber se possui as sandálias nas especificações desejadas.

A entidade caixa é para saber o valor que se inicia em cada caixa por isso o uso dos atributos Id para identificar o caixa e o atributo ValorInicial para saber com quanto se inicia. Utiliza-se também o atributo data para registrar as datas.

A entidade sandália é responsável pelos dados das sandálias como o Id de cada sandália, o nome que será a descrição que cada sandália irá receber, o modelo que possui um código de referência para o usuário do banco de dados identificar se é havaiana ou outro tipo, o custo valor pago pelo dono da loja para revenda e removido que é referente as sandálias que foram removidas da entidade.

A entidade despesas é utilizado para justificar qualquer tipo de retirada do caixa por isso o atributo “justificativa”, também utiliza a data para guardar o dia do acontecimento e valor para saber o valor que foi retirado.

A entidade registrovendas é utilizado para criar a nota fiscal da compra do cliente e utiliza o Id para identificar cada venda, novamente faz uso da sandaliasID como uma foreign key (FK), possui as atribuições como a cor, o tamanho, a data da compra, o tipo de pagamento e o valor da sandália.

Além dos atributos utilizados também foram utilizados tipos de entradas de dados em cada atributo, estes são:

- INT: Tipo de dado inteiro informa que é referente a algum número inteiro.
- DECIMAL: Tipo de dado numeral também, porém possibilita o uso de números decimais.
- LONGTEXT: Tipo de dado para texto com grande quantidade de intervalo.
- TINYINT: Tipo de dado INT, porém com uma menor quantidade intervalo
- DATETIME: Define uma data combinada com uma hora do dia que inclui frações de segundos e se baseia em um período de 24 horas.

4. Modelo Físico

Será apresentado os create tables utilizados para criar as tabelas com os atributos e os tipos de dados que serão inseridos, suas características e as primary keys' de cada tabela.

```
CREATE TABLE `__efmigrationshistory` (  
  `MigrationId` varchar(95) NOT NULL,  
  `ProductVersion` varchar(32) NOT NULL,  
  PRIMARY KEY (`MigrationId`)  
)
```

```
CREATE TABLE `caixas` (  
  `Id` int NOT NULL AUTO_INCREMENT,  
  `ValorInicial` decimal(5,2) NOT NULL,  
  `Data` datetime(6) NOT NULL,  
  PRIMARY KEY (`Id`)  
)
```

```
CREATE TABLE `despesas` (  
  `Id` int NOT NULL AUTO_INCREMENT,  
  `Valor` decimal(5,2) NOT NULL,  
  `Justificativa` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,  
  `Data` datetime(6) NOT NULL,  
  PRIMARY KEY (`Id`)  
)
```

```
CREATE TABLE `registrosvendas` (  
  `Id` int NOT NULL AUTO_INCREMENT,
```

```

`SandaliaId` int DEFAULT NULL,
`Cor` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
`Tamanho` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
`Data` datetime(6) NOT NULL,
`Pagamento` int DEFAULT '0',
`Valor` decimal(5,2) NOT NULL DEFAULT '0.00',
`Troca` int DEFAULT '0',
PRIMARY KEY (`Id`),
KEY `IX_RegistrosVendas_SandaliaId` (`SandaliaId`),
CONSTRAINT `FK_RegistrosVendas_Sandalia_SandaliaId` FOREIGN KEY
(`SandaliaId`) REFERENCES `sandalia` (`Id`) ON DELETE RESTRICT
)

```

```

CREATE TABLE `sandalia` (
`Id` int NOT NULL AUTO_INCREMENT,
`Nome` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
`Valor` decimal(5,2) NOT NULL,
`Modelo` int NOT NULL,
`Custo` decimal(5,2) NOT NULL DEFAULT '0.00',
`Removido` tinyint(1) NOT NULL DEFAULT '0',
PRIMARY KEY (`Id`)
)

```

```

CREATE TABLE `sandaliaestoque` (
`Id` int NOT NULL AUTO_INCREMENT,
`SandaliaId` int DEFAULT NULL,

```

```

`Cor` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
`Tamanho` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
PRIMARY KEY (`Id`),
KEY `IX_SandaliaEstoque_SandaliaId` (`SandaliaId`),
CONSTRAINT `FK_SandaliaEstoque_Sandalia_SandaliaId` FOREIGN KEY
(`SandaliaId`) REFERENCES `sandalia` (`Id`) ON DELETE RESTRICT
)

CREATE TABLE `usuarios` (
`Id` int NOT NULL AUTO_INCREMENT,
`Tipo` int NOT NULL,
`Login` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
`Senha` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
`Nome` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci,
PRIMARY KEY (`Id`)
)

```

5. Conclusão

O trabalho teve como intuito a criação de um banco de dados de livre escolha, optamos por escolher fazer um BD de uma loja de sandália. Conseguimos criar um modelo conceitual como um esboço para o projeto e adquirimos a habilidade de identificar as entidades, os relacionamentos e os tipos de cardinalidade. Com a criação do modelo lógico criamos os atributos responsáveis por compor as tabelas e especificar as sandálias, usuários, os registros das vendas, estoque, caixas e as despesas. Além da criação das tabelas também possibilitou usar os tipos de dados de maneira mais eficaz para cada atributo. A construção do modelo físico foi de crucial importância pois fizemos uso dos códigos (SQL-DDL) para criação de tabelas já com os atributos inseridos, as primary keys e as foreign keys. De modo geral o projeto possibilitou a gente vivenciar a criação de um banco de dados do zero.