

FACULDADE DE TECNOLOGIA DE CARAPICUÍBA

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

3º SEMESTRE - NOTURNO

DISCIPLINA: BANCO DE DADOS

PROF: SANDRA HENRIQUES

Anderson Anselmi Nunes - RA 1430481923044

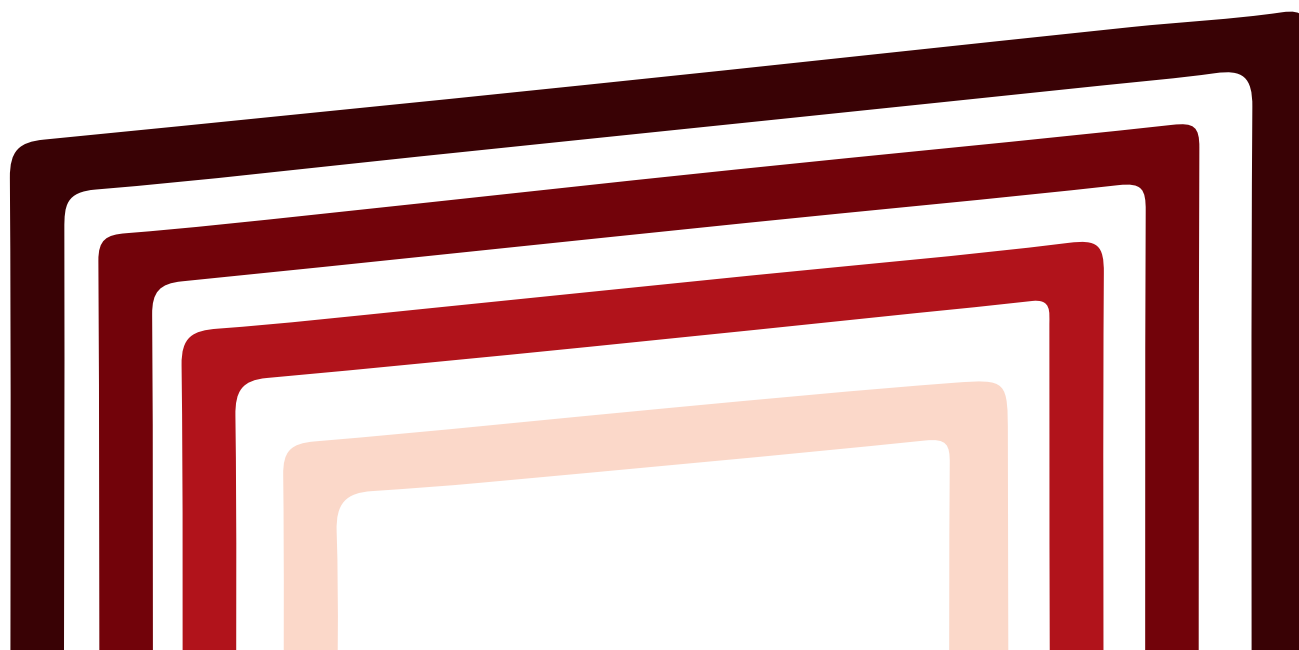
Lucas Perego de Oliveira - RA 1430481923020

Lucas Silva Pereira - RA 1430481923025

Levy José - RA 1430481923033

Fatec

Carapicuíba



Descrição do Sistema

A Drogaria **Blue Pill** é uma rede de farmácias que atua na região de SP, com 2 unidades físicas e vendas online através do site da empresa. Atualmente não utiliza um sistema para controle e gerenciamento de suas operações, motivo pelo qual estamos desenvolvendo este projeto.

Todo o controle é realizado de forma manual, com exceção do controle de estoque de medicamentos que é feito através de planilha eletrônica (Excel). Um dos principais problemas identificados é a lentidão no atendimento. O cliente chega no balcão e faz o pedido de um medicamento, o atendente precisa pesquisar em uma lista de papel, consultar o preço, fazer a soma do total dos produtos e calcular o desconto caso a pessoa seja aposentada, tenha mais de 65 anos ou possua cadastro na loja e realize o pagamento a vista. Isso gera filas e descontentamento dos clientes, o que faz com que percam vendas.

Além disso, como o troco é calculado manualmente pelo atendente, podem ocorrer erros na devolução do troco, o que pode ocasionar insatisfação dos clientes e reclamações.

Todas as vendas são registradas em um caderno, o que permite controlar o caixa e fazer as baixas no estoque no final do dia, um processo que também consome muito tempo.

O objetivo do sistema é eliminar os problemas elencados, automatizando os processos e consequentemente trazendo um ganho de produtividade e mais segurança e confiabilidade as operações. Com isso, o atendimento será melhorado evitando reclamações e aumentando a confiança e a fidelidade dos consumidores.

Facilitará também a gestão das atividades, melhorando o desempenho e otimizando o tempo gasto na realização das tarefas, fazendo com que ele possa aumentar sua competitividade, afinal uma empresa nos dias de hoje sem um sistema de informação para auxiliar na sua gestão é uma empresa em desvantagem competitiva no mercado.

Para isso são premissas do sistema:

- que ele seja online, responsivo, com uma interface amigável, intuitiva e fácil de utilizar
- cadastrar de novos produtos
- controlar o estoque de medicamentos e a venda de medicamentos controlados
- consultar o preço dos medicamentos
- gerar pedidos efetuando o cálculo da soma dos medicamentos e o valor do troco
- registrar os pedidos de vendas
- atualizar os preços online de maneira automática
- cadastrar descontos
- cadastrar clientes
- cadastrar fornecedores
- cadastrar funcionários
- cadastrar formas de pagamento
- fazer a abertura / fechamento do caixa diariamente
- emitir relatórios (limite de 3 relatórios definidos pelo cliente)
- possuir níveis de permissão para realizar as tarefas, pois algumas são exclusivas do gerente. Para isso será necessário a criação de um login e senha para cada usuário do sistema.

Atendentes podem acessar o sistema para realizar consultas, gerar pedidos, cadastrar novos medicamentos e liberar vendas de medicamentos controlados. Já os Gerentes possuem acesso a todas as funcionalidades do sistema.

Todas as vendas são realizadas através de pedido gerado pelo sistema para controle. Um pedido pode possuir um ou mais medicamentos e podem ser vendidos em uma ou mais unidades. Para medicamentos controlados é obrigatório a apresentação de receita médica e informado no pedido o número do CRM do médico que receitou.

Clientes aposentados ou com mais de 65 anos tem desconto de 10% em todos medicamentos.

Clientes cadastrados tem 5% de desconto para pagamento a vista na compra de qualquer medicamento. Pagamentos com cartão de débito ou crédito não possuem desconto.

Os descontos não são cumulativos, ou seja, se o cliente for cadastrado e aposentado por exemplo só poderá ter no máximo 10% de desconto.

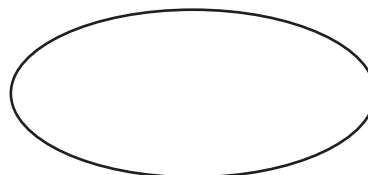
Definição Conceitual sobre o que é um DER

Um diagrama entidade relacionamento (DER) é uma representação gráfica que descreve o modelo de dados de um sistema com alto nível de abstração e, portanto, utilizado para representar o modelo conceitual de um negócio.

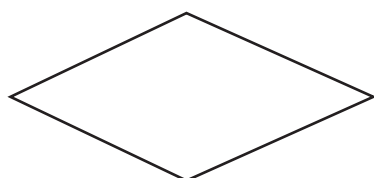
Cada representação gráfica do modelo possui um significado para facilitar o entendimento, sendo composto por símbolos geométricos. De acordo com a definição de Peter Chen podemos descrevê-los assim:



Retângulos:
Conjunto de Entidades



Elipses:
Conjunto de Atributos



Losangos:
Conjunto de Relacionamentos



Linhas:
Ligações Possíveis

Etapas de elaboração do DER

Passo 1 – Análise de requisitos

Nesta etapa é feita a leitura do briefing onde são levantados os requisitos necessários a construção do sistema e grifamos os substantivos associados a coisas do mundo real. Selecionamos estes substantivos:

Drogaria, controle, gerenciamento, medicamentos, vendas, filas, descontentamento, planilha eletrônica, atendimento, cliente, pedido, caixa, estoque, empresa, preços, descontos, fornecedores, funcionários, formas de pagamento, permissão, gerente, usuário, receita, médico, aposentados

Passo 2 – Descartar ocorrências únicas no Banco de Dados

Nesta etapa foram eliminados substantivos que como Entidades teriam apenas uma ocorrência no BD, ou seja um único objeto. Foram eliminados: gerente e drogaria

Passo 3 – Descartar substantivos

Nesta etapa eliminamos os substantivos que servem apenas para o entendimento do problema a ser resolvido. Foram eliminados: controle, gerenciamento, filas, descontentamento, planilha eletrônica, atendimento, preços, descontos, permissão, gerente, receita, médico, aposentados.

Passo 4 - Não se aplica

Passo 5 – Listar substantivos

Nesta etapa, são selecionados os substantivos que vão se tornar Entidades. Definimos as seguintes Entidades: medicamentos, cliente, pedido, estoque, empresa, fornecedores, funcionários, formas de pagamento, usuário

Passo 6 – Identificar os relacionamentos

Para cada par de Entidades, foram identificados os relacionamentos existentes conforme descritos abaixo:

Empresa x Funcionários (trabalham)

Funcionários x Usuários (possuem acesso ao sistema)

Empresa x Produtos (vendem)

Empresa x Estoque (possuem)

Fornecedores x Estoque (fornecem)

Estoque x Produtos (abastecem)

Clientes x Produtos (compram)

Produtos x Pedido (possuem)

Pedido x Formas de Pagamento (possuem)

Passo 7 – Identificar os Atributos de cada Entidade

Imaginando a maneira como teríamos a partir de dados armazenados as informações sobre aquela Entidade, teríamos algo assim:

Clientes: idCli, nomeCli, cpfCli, dtNascCli

Pedidos: idPed, qtdeItemPed, valorItemPed, descItemPed, totalItemPed

FormasPagto: idPagto, nomePagto

Produtos: idProd, nomeProd, valorProd, controladoProd,.crmMedicoProd

Fornecedores: idForn, nomeForm, telForn, emailForn

Estoques: idEst, qtdeEst

Empresas: idEmp, nomeEmp, telEmp, emailEmp

Funcionarios: idFunc, nomeFunc, cargoFunc

Usuarios: idUsuario, loginUsuario, senhaUsuario, tipoUsuario

Diagrama DER

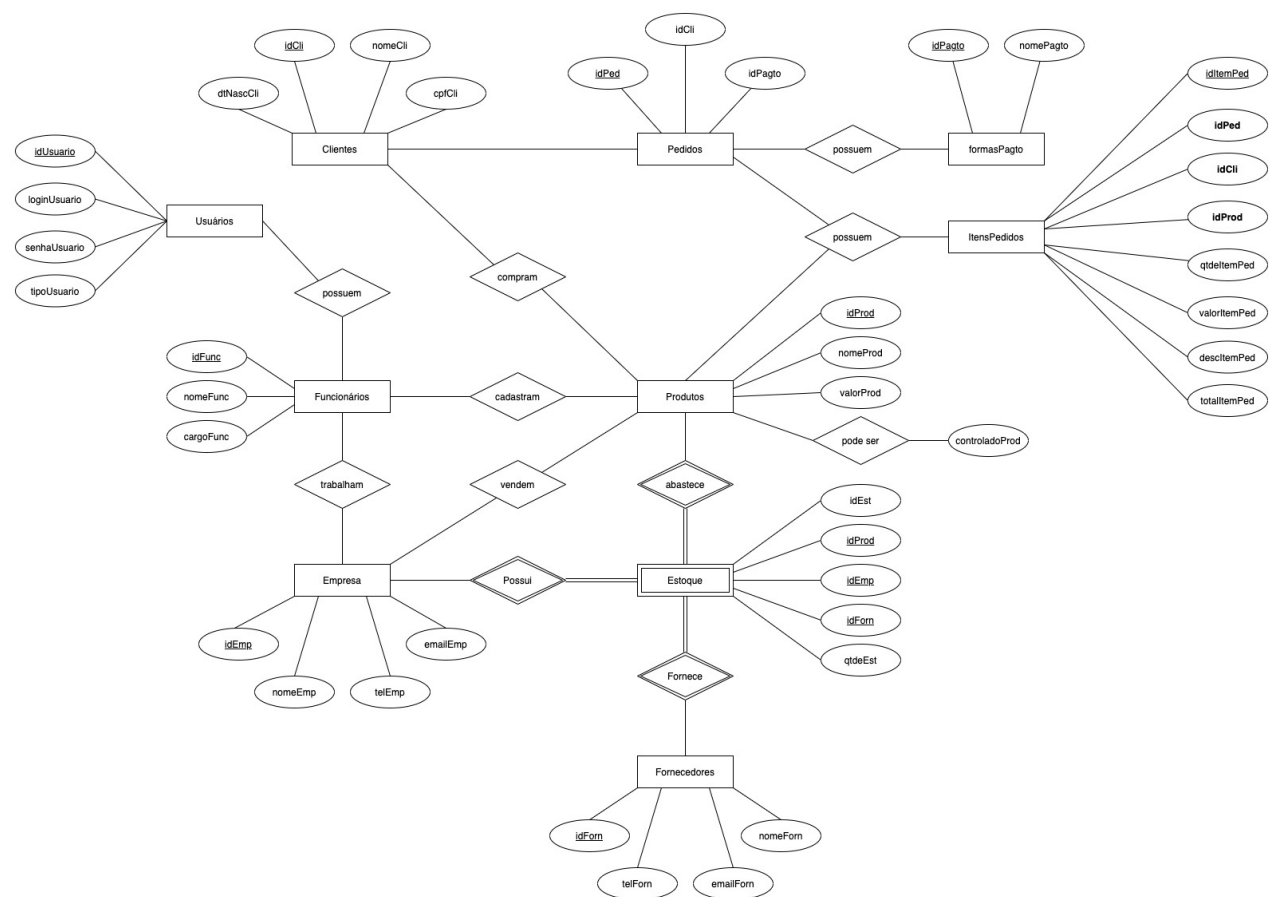
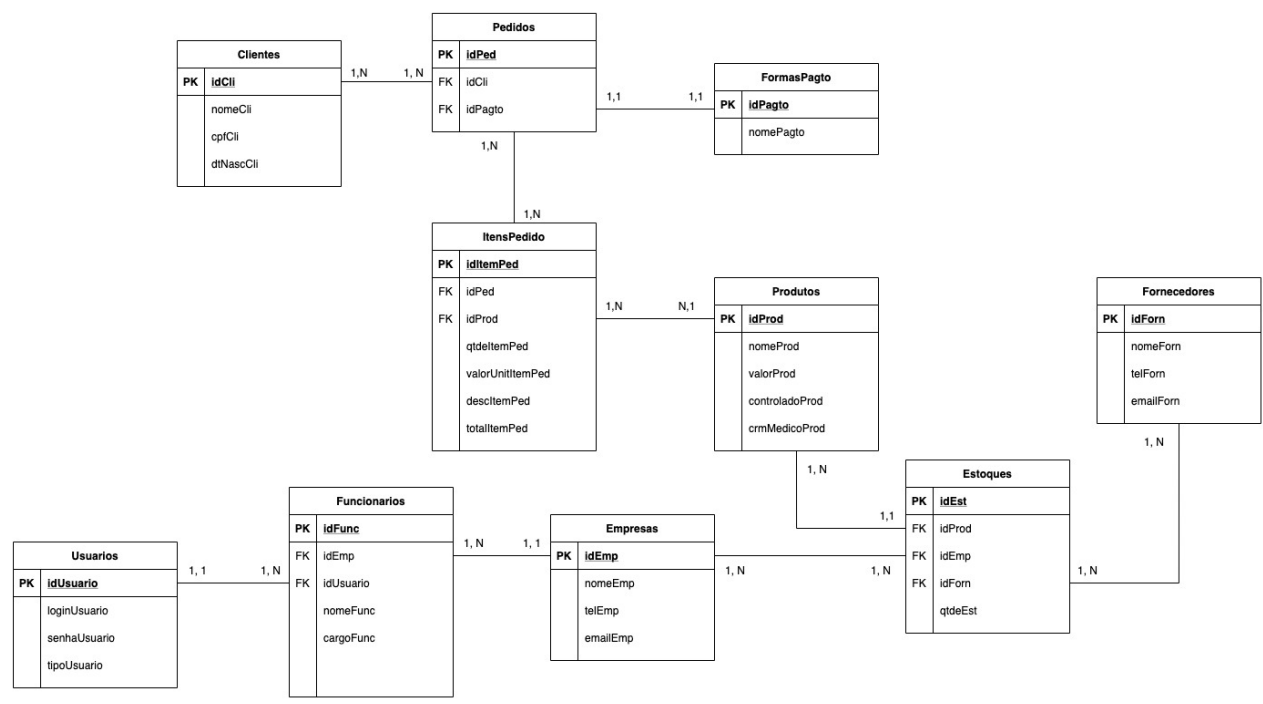


Diagrama MER



Definição Conceitual sobre o que um MER

O Modelo Entidade Relacionamento (MER) é um modelo de dados representativo de um SGBD, baseando-se no princípio de que todos os dados estão armazenados em tabelas, foi criado por Edgar Frank Codd, este modelo é baseado na lógica e na teoria dos conjuntos, ou seja, baseia-se na entidade e relação, sendo uma entidade um objeto que possui dados exibidos em tabela, sendo seus atributos relacionados em outras tabelas, de forma que uma interaja com a outra.

Etapas de elaboração do MER

Para elaboração do nosso projeto MER, utilizamos as seguintes regras durante o processo de migração DER -> MER

Regra 1 – Criação das Tabelas com base nas Entidades

Criação das tabelas com o mesmo nome da Entidade referenciada, além de seus atributos.

Clientes: idCli, nomeCli, cpfCli, dtNascCli

Pedidos: idPed, idCli, idPagto

ItensPedido: idPed, idProd, qtdItemPed, valorItemPed, descItemPed, totalItemPed

FormasPagto: idPagto, nomePagto

Produtos: idProd, nomeProd, valorProd, controladoProd,.crmMedicoProd

Fornecedores: idForn, nomeForm, telForn, emailForn

Estoques: idEst, idProd, idEmp, idForn, qtdeEst

Empresas: idEmp, nomeEmp, telEmp, emailEmp

Funcionarios: idFunc, idUsuario, nomeFunc, cargoFunc

Usuarios: idUsuario, loginUsuario, senhaUsuario, tipoUsuario

Regra 4 – Relacionamentos 1:1 entre Entidades

Escolhemos uma das relações e incluímos como chave estrangeira (identificada pela sigla FK), a chave primária (identificada pela sigla PK) da outra Entidade. Exemplo:

Pedidos: [idPed, **idCli**, **idPagto**]

FormasPagto: [idPagto, nomePagto]

Regra 5 – Relacionamentos do tipo 1:N entre Entidades

Identificamos a tabela que deve receber como chave estrangeira (identificada pela sigla FK), a chave primária (identificada pela sigla PK) da outra Entidade. Exemplo:

Clientes: [idCli, nomeCli, cpfCli, dtNascCli]

Pedidos: [idPed, **idCli**, **idPagto**]

Regra 8 – Para relacionamentos do tipo N-ário: N > 2 entre Entidades

Criamos uma nova tabela e incluímos como chave estrangeira (identificada pela sigla FK), a chave primária (identificada pela sigla PK) das outras Entidades participantes. Exemplo:

Produtos: [idProd, nomeProd, valorProd, controladoProd, crmMedicoProd]

Fornecedores: [idForn, nomeForm, telForn, emailForn]

Empresas: [idEmp, nomeEmp, telEmp, emailEmp]

Estoques: [idEst, **idProd**, **idEmp**, **idForn**, qtdeEst]

**nova tabela criada

4. Normalização

Para normalizar os campos e registros do banco de dados foram definidos os seguintes critérios:

Todos os registros serão gravados em **UPPERCASE** e sem acentuação, minimizando redundâncias e consequentemente evitando riscos de inconsistências;

Todos atributos que possam ser calculados com base em outros atributos devem ser eliminados como por exemplo idade (que pode ser calculada pela data atual menos data de nascimento);

Separando as tabelas onde atributos aparecem mais de uma vez, dividindo o banco de dados de forma que o atributo dependa exclusivamente da chave de sua tabela;

Utilizar nomes para as tabelas que remetam a seu conteúdo;

Outro ponto a destacar é que utilizando as regras de normalização você reduz e facilita as manutenções no banco de dados.

DDL - Conceito

Data Definition language (DDL), é utilizada para a definição e manutenção das estruturas de banco de dados, tais como a criação do próprio banco de dados e das tabelas que o compõem, além das relações entre as tabelas e os objetos do banco de dados.

Exemplo de comandos utilizados: CREATE, ALTER, DROP

DML - Conceito

Data Manipulation Language, é utilizada para realizar consultas, inserções, exclusões e alterações nos registros das tabelas de forma simultânea. Exemplo de comandos utilizados: INSERT, UPDATE, SELECT, DELETE

DDL e DML do Projeto

Criação das QUERIES utilizadas no projeto.

Banco de Dados utilizado: **MySQL com SGBD MySQLWorkbench**

DDL - TABELA CLIENTES:

```
CREATE TABLE `Clientes` (  
  `idCli` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `cpfCli` int(11) DEFAULT NULL,  
  `dtNascCli` date DEFAULT NULL,  
  PRIMARY KEY (`idCli`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA EMPRESAS:

```
CREATE TABLE `Empresas` (  
  `idEmp` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `nomeEmp` varchar(255) DEFAULT NULL,  
  `telEmp` varchar(11) DEFAULT NULL,  
  `emailEmp` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`idEmp`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA FORNECEDORES:

```
CREATE TABLE `Fornecedores` (  
  `idForn` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `nomeForn` varchar(255) DEFAULT NULL,  
  `telForn` varchar(11) DEFAULT NULL,  
  `emailForn` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`idForn`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA ESTOQUES:

```
CREATE TABLE `Estoques` (  
  `idEst` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `idProd` int(11) unsigned DEFAULT NULL,  
  `idEmp` int(11) unsigned DEFAULT NULL,  
  `idForn` int(11) unsigned DEFAULT NULL,  
  `qtdeEst` int(11) DEFAULT NULL,  
  PRIMARY KEY (`idEst`),  
  KEY `FK_ESTPROD` (`idProd`),  
  KEY `FK_ESTEMP` (`idEmp`),  
  KEY `FK_ESTFORN` (`idForn`),  
  CONSTRAINT `FK_ESTEMP` FOREIGN KEY (`idEmp`)  
    REFERENCES `Empresas` (`idEmp`),  
  CONSTRAINT `FK_ESTFORN` FOREIGN KEY (`idForn`)  
    REFERENCES `Fornecedores` (`idForn`),  
  CONSTRAINT `FK_ESTPROD` FOREIGN KEY (`idProd`)  
    REFERENCES `Produtos` (`idProd`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA FORMAS DE PAGAMENTO:

```
CREATE TABLE `FormasPagto` (  
  `idPagto` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `nomePagto` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`idPagto`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA FUNCIONARIOS:

```
CREATE TABLE `Funcionarios` (  
  `idFunc` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `idEmp` int(11) unsigned DEFAULT NULL,  
  `idUsuario` int(11) unsigned DEFAULT NULL,  
  `nomeFunc` varchar(255) DEFAULT NULL,  
  `cargoFunc` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`idFunc`),  
  KEY `FK_FUNCEMP` (`idEmp`),  
  KEY `FK_FUNCUSU` (`idUsuario`),  
  CONSTRAINT `FK_FUNCEMP` FOREIGN KEY (`idEmp`)  
    REFERENCES `Empresas` (`idEmp`),  
  CONSTRAINT `FK_FUNCUSU` FOREIGN KEY (`idUsuario`)  
    REFERENCES `Usuarios` (`idUsuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA ITENS DO PEDIDO:

```
CREATE TABLE `ItensPedido` (  
  `idItemPed` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `idPed` int(11) unsigned DEFAULT NULL,  
  `idProd` int(11) unsigned DEFAULT NULL,  
  `qtdItemPed` int(11) DEFAULT NULL,  
  `valorItemPed` decimal(10,2) DEFAULT NULL,  
  `descItemPed` decimal(10,2) DEFAULT NULL,  
  `totalItemPed` decimal(10,2) DEFAULT NULL,  
  PRIMARY KEY (`idItemPed`),  
  KEY `FK_ITEMPED` (`idPed`),  
  KEY `FK_ITEMPROD` (`idProd`),  
  CONSTRAINT `FK_ITEMPED` FOREIGN KEY (`idPed`)  
    REFERENCES `Pedidos` (`idPed`),  
  CONSTRAINT `FK_ITEMPROD` FOREIGN KEY (`idProd`)  
    REFERENCES `Produtos` (`idProd`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA PEDIDOS:

```
CREATE TABLE `Pedidos` (  
  `idPed` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `idCli` int(11) unsigned NOT NULL,  
  `idPagto` int(11) unsigned NOT NULL,  
  PRIMARY KEY (`idPed`),  
  KEY `FK_PEDCLI` (`idCli`),  
  KEY `FK_PEDPAG` (`idPagto`),  
  CONSTRAINT `FK_PEDCLI` FOREIGN KEY (`idCli`) REFERENCES  
  `Clientes` (`idCli`),  
  CONSTRAINT `FK_PEDPAG` FOREIGN KEY (`idPagto`)  
  REFERENCES `FormasPagto` (`idPagto`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA PRODUTOS:

```
CREATE TABLE `Produtos` (  
  `idProd` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `nomeProd` varchar(255) DEFAULT NULL,  
  `valorProd` decimal(10,2) DEFAULT NULL,  
  `controladoProd` tinyint(1) DEFAULT NULL,  
  `crmMedicoProd` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`idProd`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DDL - TABELA USUARIOS:

```
CREATE TABLE `Usuarios` (  
  `idUserario` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `loginUsuario` varchar(255) DEFAULT NULL,  
  `senhaUsuario` varchar(255) DEFAULT NULL,  
  `tipoUsuario` tinyint(1) DEFAULT NULL,  
  PRIMARY KEY (`idUserario`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

DML - TABELA CLIENTES:

Query para cadastrar clientes

INSERT INTO Clientes

(idCli, nomeCli, cpfCli, dtNascCli)

VALUES

```
(
    1
    , `ANDERSON NUNES`
    , `13453949854`
    , `1972-08-28`
),
(
    2
    , `LUCAS PEREGO DE OLIVEIRA`
    , `12345678911`
    , `1998-10-05`
),
(
    3
    , `LUCAS SILVA PEREIRA`
    , `22222222333`
    , `1999-12-02`
),
(
    4
    , `LEVI JOSE`
    , `99912345611`
    , `2000-02-24`
)
```


DML - TABELA CLIENTES:

Query para consultar clientes e ordenar em ordem alfabética

```
SELECT idCli, nomeCli, cpfCli, dtNascCli  
FROM Clientes  
ORDER BY nomeCli ASC
```

resultado:

1	ANDERSON NUNES	13453949854	1972-28-08
4	LEVI JOSE	99912345611	2000-02-24
2	LUCAS PEREGO DE OLIVEIRA	12345678911	1998-10-05
3	LUCAS SILVA PEREIRA	2222222333	1999-12-02

QUERY para buscar apenas os 2 primeiros registros

```
SELECT idCli, nomeCli, cpfCli, dtNascCli  
FROM Clientes  
ORDER BY nomeCli ASC  
LIMIT 2
```

resultado:

1	ANDERSON NUNES	13453949854	1972-28-08
4	LEVI JOSE	99912345611	2000-02-24

Query para alterar o campo CPF de um determinado cliente

```
UPDATE Clientes  
SET cpfCli = '12345678900'  
WHERE idCli = 2
```

resultado:

2	LUCAS PERGO DE OLIVEIRA	12345678900	1998-10-05
---	-------------------------	-------------	------------

Resultado final da Modelagem

Clientes: [idCli, nomeCli, cpfCli, dtNascCli]

Pedidos: [idPed, idCli, idPagto]

ItensPedido: [idItemPed, idPed, idProd, qtdItemPed, valorItemPed, descItemPed, totalItemPed]

FormasPagto: [idPagto, nomePagto]

Produtos: [idProd, nomeProd, valorProd, controladoProd,.crmMedicoProd]

Fornecedores: [idForn, nomeForm, telForn, emailForn]

Estoques: [idEst, idProd, idEmp, idForn, qtdeEst]

Empresas: [idEmp, nomeEmp, telEmp, emailEmp]

Funcionarios: [idFunc, idUsuario, nomeFunc, cargoFunc]

Usuarios: [idUsuario, loginUsuario, senhaUsuario, tipoUsuario]

Considerações finais sobre a modelagem do BD

Fazer a modelagem de dados é um processo que requer bastante atenção, pois qualquer erro pode resultar em uma modelagem sem eficiência e com dados redundantes.

O passo a passo é importante para identificarmos as Entidades e os Atributos, além dos relacionamentos entre eles. É um pouco confuso o diagrama DER, mas se for bem feito seguindo as regras para sua criação facilita bastante a montagem do MER depois.

É um conhecimento a mais que adquirimos e que com certeza nos ajudará a estruturar projetos de forma mais precisa e adequada.

Referências Bibliográficas

Material apresentado em sala de aula, disciplina

Banco de Dados: **HENRIQUES, Sandra Bianca**