**BANCO DE DADOS**

**Trabalho – Relatório**

|  |  |
| --- | --- |
| **Curso:** | CIÊNCIA DE DADOS |
| **Aluno(a):** | WANDERSON TEIXEIRA SOUSA |
| **RU:** | 2412022 |

1. **1ª Etapa – Modelagem**

**Pontuação:** 25 pontos.

Dadas as regras de negócio abaixo listadas, referentes ao estudo de caso de uma Clínica Médica, elabore o Modelo Entidade-Relacionamento (MER), isto é, o modelo conceitual.

O Modelo Entidade-Relacionamento (MER) deve contemplar os seguintes itens:

* Entidades;
* Atributos;
* Relacionamentos;
* Cardinalidades;
* Chaves primárias;
* Chaves estrangeiras.

Uma Clínica Médica necessita controlar os dados das consultas realizadas. Para isso, contratou um profissional de Banco de Dados, a fim de modelar o Banco de Dados que armazenará os dados das consultas.

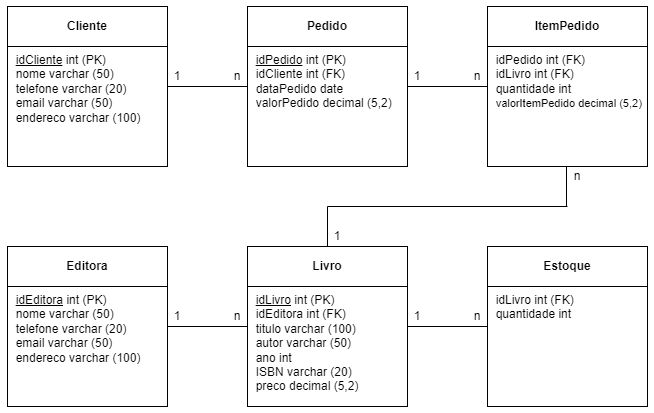
As regras de negócio são:

* Médico – Deverão ser armazenados os seguintes dados: CRM, especialidade, nome, telefone, e-mail e endereço, sendo o endereço composto por rua, número, complemento, bairro, CEP, cidade e estado;
* Consulta – Deverão ser armazenados os seguintes dados: identificação da consulta, data e horário;
* Paciente – Deverão ser armazenados os seguintes dados: CPF, nome, telefone, e-mail e endereço, sendo o endereço composto por rua, número, complemento, bairro, CEP, cidade e estado;
* Convênio – Deverão ser armazenados os seguintes dados: identificação do convênio, empresa, tipo, vencimento e percentual de coparticipação;
* Um médico pode realizar zero ou várias consultas, assim como zero ou várias consultas podem ser realizadas por um médico;
* Um médico pode atender zero ou vários convênios, assim como zero ou vários convênios podem ser atendidos por um médico;
* Um paciente pode marcar zero ou várias consultas, assim como zero ou várias consultas podem ser marcadas por um paciente;
* Um paciente pode possuir zero ou vários convênios, assim como zero ou vários convênios podem pertencer a um paciente.
* Diagrama

  Descrição gerada automaticamente

1. **2ª Etapa – Implementação**

Considere o seguinte Modelo Relacional (modelo lógico), referente ao estudo de caso de uma Livraria:



Com base no Modelo Relacional dado e utilizando a *Structured Query Language* (SQL), no MySQL Workbench, implemente o que se pede.

**Observação:** Para testar o Banco de Dados após a implementação, utilize os comandos contidos no arquivo “Trabalho – Populando o Banco de Dados” para popular as tabelas. Tal arquivo contém todos os comandos de inserção dos dados (fictícios) necessários para a realização dos testes.

**Pontuação:** 25 pontos.

1. Implemente um Banco de Dados chamado “Livraria”. Após, implemente as tabelas, conforme o Modelo Relacional dado, observando as chaves primárias e as chaves estrangeiras. Todos os campos, de todas as tabelas, não podem ser nulos (*not null*).

create database Livraria;

use Livraria;

create table Cliente (

    idCliente int not null,

    nome varchar(50) not null,

    telefone varchar(20) not null,

    email varchar(50) not null,

    endereco varchar(100) not null,

    primary key (idCliente)

);

create table Pedido (

    idPedido int not null,

    idCliente int not null,

    dataPedido date not null,

    valorPedido decimal(5,2) not null,

    primary key (idPedido),

    foreign key (idCliente) references cliente (idCliente)

);

create table Editora (

    idEditora int not null,

    nome varchar(50) not null,

    telefone varchar(20) not null,

    email varchar(50) not null,

    endereco varchar(100) not null,

    primary key (idEditora)

);

create table Livro (

    idLivro int not null,

    idEditora int not null,

    titulo varchar(100) not null,

    autor varchar(50) not null,

    ano int not null,

    ISBN varchar(20) not null,

    preco decimal(5,2) not null,

    primary key (idLivro),

    foreign key (idEditora) references editora (idEditora)

);

create table ItemPedido (

    idPedido int not null,

    idLivro int not null,

    quantidade int not null,

    valorItemPedido decimal(5,2) not null,

    foreign key (idPedido) references pedido (idPedido),

    foreign key (idLivro) references livro (idLivro)

    );

create table Estoque (

    idLivro int not null,

    quantidade int not null,

    foreign key (idLivro) references Livro (idLivro)

    );

**Pontuação:** 10 pontos.

1. Implemente uma consulta para listar o quantitativo de livros cadastrados, independentemente da editora.

SELECT count(idLivro) AS 'Quantidade de Livros'

FROM livro;



**Pontuação:** 10 pontos.

1. Implemente uma consulta para listar o nome dos clientes cadastrados. A listagem deve ser mostrada em ordem crescente.

SELECT nome

FROM cliente

ORDER BY nome ASC;

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Pontuação:** 10 pontos.

1. Implemente uma consulta para listar o nome de todas as editoras e os títulos de seus respectivos livros. A listagem deve ser mostrada em ordem decrescente pelo nome das editoras.

SELECT e.nome AS 'Nome editora', l.titulo AS 'Titulo livro'

from editora e

INNER JOIN livro l ON e.idEditora = l.idEditora

ORDER BY e.nome DESC;

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Pontuação:** 10 pontos.

1. Implemente uma consulta para listar o nome das editoras e a média de preço de seus respectivos livros. Para isso, utilize o comando *group by*.

SELECT e.nome AS 'Nome da Editora',

    ROUND(AVG(l.preco), 2) AS 'Média de preço dos Livros'

from editora e

INNER JOIN livro l ON e.idEditora = l.idEditora

GROUP BY e.nome;

Interface gráfica do usuário, Tabela

Descrição gerada automaticamente

**Pontuação:** 10 pontos.

1. Implemente uma consulta para listar o nome de todos os clientes e a quantidade de livros comprados pelos mesmos. Para isso, utilize o comando *group by*.

SELECT c.nome AS "Nome do Cliente",

       COUNT(ip.idLivro) AS "Quantidade de Livros Comprados"

FROM Cliente c

INNER JOIN Pedido p ON c.idCliente = p.idCliente

INNER JOIN ItemPedido ip ON p.idPedido = ip.idPedido  -- Juntando a tabela de livros, se aplicável

GROUP BY c.nome

ORDER BY c.nome;

Tabela

Descrição gerada automaticamente