Análise e Desenvolvimento de Sistemas

Banco de Dados I

Gustavo David Cesário Da Silva

Gestão de Hotel

Cajazeiras - PB

2023

Gustavo David Cesário Da Silva

Gestão de Hotel

Trabalho apresentado ao Curso Superior Tecnólogo de Análise e Desenvolvimento de Sistemas do IFPB – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba Campus Cajazeiras, para a disciplina de Banco de Dados 1, ministrado pelo Prof. Dr. Fábio Gomes de Andrade.

Contents

1 Introdução

Um Hotel cresceu. Com esse crescimento, também veio uma consequência: a grande quantidade de dados que são gerados todos os dias. Algo que antes era simples, hoje passa a ser complexo. Dito isso, a necessidade de que estes dados sejam armazenados e gerenciados, também que isso seja feito de maneira íntegra e consistente, faz-se necessária, tal que esse armazenamento seja capaz de responder algumas consultas muito importantes no dia a dia corriqueiro desse hotel. Contudo, esses dados precisam ser restritos a pessoas específicas.

Dessarte, um banco de dados se torna não só uma possibilidade, mas também imprescindível e inerente ao desempenho. É um ponto crítico na operação eficiente e eficaz de um hotel, permitindo uma gestão mais informada e personalizada das operações e uma melhor experiência para os hóspedes.

2 Modelo Conceitual

2.1 Levantamento de Requisitos

1. O hotel possui um conjunto de funcionários. Um funcionário pode ser Auxiliar de Serviços, Recepcionista ou Cozinheiro. Não é permitido que funcionários desempenhem mais que um papel no hotel. Para cada funcionário, é armazenado seu nome, CPF, telefone, salário e data de admissão.

2. O hotel possui duas formas para hospedagem: Reserva e Balcão. Para cada hospedagem, devem ser registrados o nome do cliente responsável e a data em que a mesma foi efetuada.

3. Para cada reserva, também são armazenados a data prevista para entrada, data prevista para saída e o status. Já para hospedagens feitas no Balcão, também deve ser registrado a data de entrada, a data prevista para saída e o Recepcionista que realizou o check-in e deve ser aplicada uma comissão a esse atendente. A Reserva por ser feita via Booking, não requer associação a um atendente.

4. No momento em que o cliente chega ao hotel, é feito um cadastro juntamente com o check-in, visando facilitar próximas hospedagens. Cada cliente terá armazenado seu nome, CPF, data de nascimento, um telefone para contato e a forma de pagamento preferencial.

5. O hotel aceita três formas de pagamento: Dinheiro, PIX e cartão. Para cada pagamento, deve ser armazenado o valor e a data do pagamento. Para pagamento por PIX, também deve ser armazenado o CPF do cliente que fez o pagamento. Já para a forma de pagamento por cartão, também deve ser gerado um cadastro para facilitar próximas estadias. Nesse cadastro são armazenados o número do cartão, o titular e a data de validade. O pagamento em dinheiro não requer nenhum armazenamento. Cada pagamento deve ser associado a uma hospedagem e essa associação requer um valor. Assim, cada hospedagem terá um valor variável de pagamento.

6. Cada hospedagem está relacionada a um tipo de quarto. Para cada tipo de quarto, deve ser registrada a capacidade (individual, duplo ou triplo) e o padrão do quarto (simples ou luxo) e o preço. Diversos quartos

estão disponíveis no hotel e estão tão, e somente, associados a um tipo específico de quarto.

7. Para cada quarto, deve ser armazenado seu número (que é único), seu andar e o Auxiliar de Serviços responsável pela limpeza.

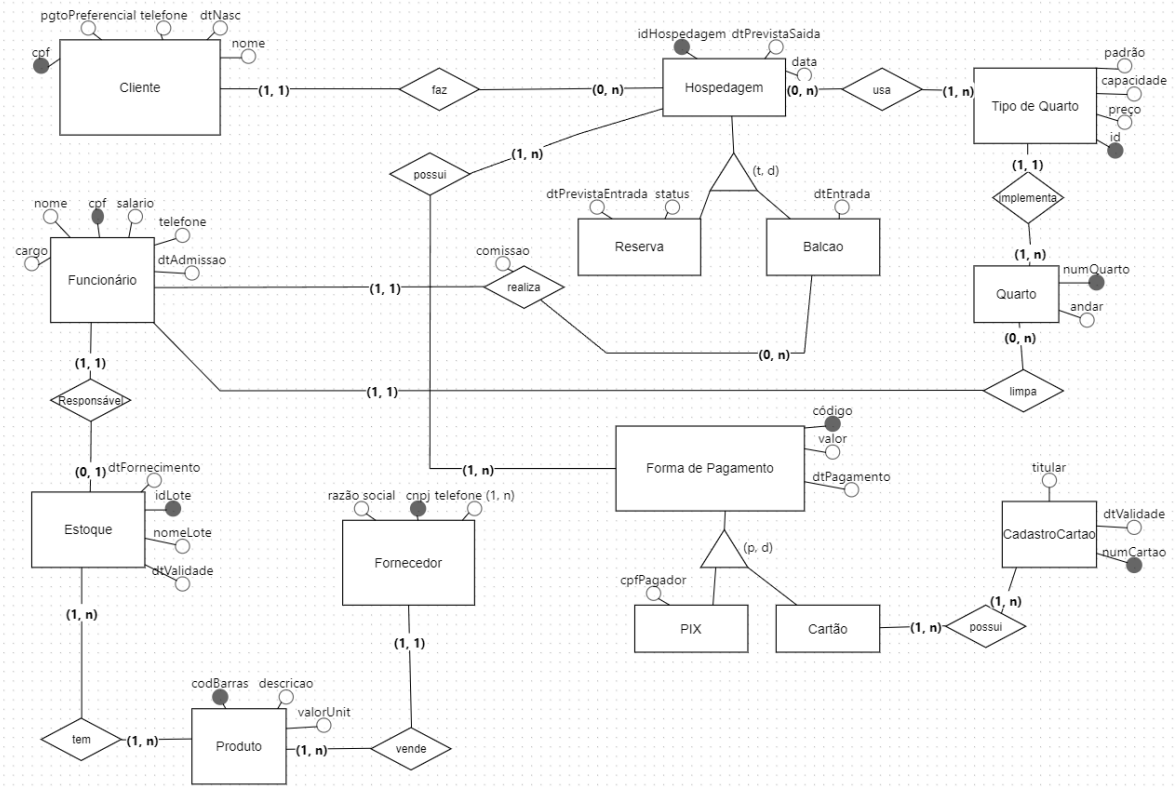
8. O hotel possui um fornecedor de alimentos. Para cada fornecedor, deve ser armazenado seu CNPJ, razão social e telefone. Cada fornecedor, vende um lote que são armazenados em um estoque. Para cada Produto, deve ser armazenado seu valor unitário, o código de barras e a descrição.

9. Por fim, o hotel possui um estoque de alimentos e para o controle dos produtos que nela estão, deve ser armazenado quem é o Cozinheiro responsável, todos os lotes de alimentos, as datas de validade e as datas de fornecimento.

2.2 Consultas

- Listar detalhes das hospedagens com informações do cliente associado;
- Listar informações sobre os quartos e os funcionários associados a eles;
- Listar funcionários que não possuem número de telefone registrado;
- Listar informações sobre os quartos que não possuem um funcionário responsável;
- Listar clientes cujos nomes contenham 'Andrade';
- Listar produtos cujas descrições comecem com a letra 'M';
- Ordenar os funcionários em ordem decrescente de salário;
- Listar produtos ordenados pelo valor unitário em ordem crescente;
- Verificar os dados do funcionário que possui o maior salário;
- Verificar qual Produto tem mais estoque;
- Consultar funcionários com salários superiores à média de salários;
- Busca o funcionário que realizou o maior número de hospedagens.

2.3 Diagrama Entidade-Relacionamento



2.4 Dicionário Conceitual de Dados

Entidade Cliente:

- Criada para armazenar os dados dos clientes do hotel.

Atributos:

- Nome: armazena o nome de um cliente;
- CPF: armazena o número de CPF de um cliente;
- Telefone: armazena um ou mais números de telefone de um cliente;
- Data de Nascimento: armazena a data de nascimento de um cliente.
- Pagamento preferencial: armazena o tipo preferencial de pagamento de um cliente.

Entidade Funcionário:

- Gerada para armazenar informações de um funcionário do hotel.

Atributos:

- Nome: armazena o nome de um funcionário;
- CPF: armazena o número de CPF de um funcionário;
- Salário: armazena o valor do salário de um funcionário;
- Telefone: armazena um telefone de um funcionário;
- Data de Admissão: armazena a data em que um funcionário foi contratado;
- Cargo: armazena o cargo de um funcionário.

Entidade Hospedagem:

- Generalização das Entidades Reserva e Balcão, utilizada para armazenar informações comuns das hospedagens.

Atributos:

- Data da Hospedagem: armazena a data de uma hospedagem;
- Id da Hospedagem: armazena o número do código da hospedagem.
- Data prevista da sáida: armazena a data prevista para a sáida do cliente

Entidade Reserva:

- Especialização da Entidade Hospedagem. Gerada para armazenar informações específicas de um reserva.

Atributos:

- Data prevista de Entrada: armazena a data prevista de entrada de um cliente;
- Status: armazena o status de uma reserva.

Entidade Balcão:

- Especialização da Entidade Hospedagem, utilizada para armazenar informações sobre hospedagens feitas no balcão.

Atributos:

- Data de Entrada: armazena a data que um cliente se hospedou;

Entidade Tipo de Quarto:

- Gerada para armazenar informações sobre um tipo de quarto.

Atributos:

- Padrão: armazena o padrão de um tipo de quarto;
- Capacidade: armazena a capacidade de um tipo de quarto;
- Preço: armazena o preço de um tipo de quarto.

Entidade Quarto:

- Gerada para armazenar informações sobre um quarto.

Atributos:

- Número do Quarto: armazena o número do quarto;
- Andar: armazena um número que representa em qual andar o quarto se localiza.

Entidade Produto:

- Gerada para armazenar informações sobre um produto que é vendido por um Fornecedor.

Atributos:

- Código de Barras: armazena o número do código de barras de um produto;
- Descrição: armazena a descrição de um produto;
- Valor Unitário: armazena o valor por unidade de um produto.

Entidade Estoque:

- Gerada para armazenar informações sobre o estoque de um produto.

Atributos:

- Id do Lote: armazena um número que representa o id do lote;
- Nome do Lote: armazena o nome do lote de um produto;
- Data Fornecimento: armazena a data que o fornecimento foi recebido;
- Data validade: armazena a data de validade dos produtos de um lote;

Entidade Fornecedor:

- Gerada para armazenar informações sobre um fornecedor.

Atributos:

- CNPJ: armazena o CNPJ de um fornecedor;
- Razão social: armazena a razão social de um fornecedor;
- Telefone: armazena o número de telefone de um fornecedor.

Entidade Forma de Pagamento:

- Generalização das Entidades PIX e Cartão, utilizada para armazenar informações comuns das formas de pagamento.

Atributos:

- Código: armazena o código de um pagamento;
- Valor: armazena o valor de um pagamento;
- Data Do Pagamento: armazena a data que um pagamento foi realizado.

Entidade PIX:

- Especialização da Entidade Forma de Pagamento, utilizada para armazenar informações sobre pagamentos feitos com PIX.

Atributos:

- CPF Pagador: armazena o número de CPF de um Cliente que fez um pagamento.

Entidade Cartão:

- Especialização da Entidade Forma de Pagamento, utilizada para armazenar informações sobre pagamentos feitos com cartão.

Atributos:

-

Entidade Cadastro Cartão:

- Gerada para armazenar informações sobre um cadastro de cartão.

Atributos:

- Número do Cartão: armazena o número de um cartão;
- Data de Validade: armazena a data de validade de um cartão;
- Titular: armazena o nome do titular do cartão.

Relacionamentos:

1. **faz**: relaciona Cliente e Hospedagem, sem atributos. Um Cliente pode hospedar-se zero ou muitas vezes e uma Hospedagem está relacionada a no mínimo um e no máximo um Cliente.
2. **usa**: relaciona Hospedagem e Tipo de Quarto, sem atributos. Uma Hospedagem tem que usar no mínimo um Tipo de Quarto e no máximo vários e um Tipo de Quarto pode ser usado por no mínimo zero e no máximo várias Hospedagens.
3. **implementa**: relaciona Tipo de Quarto e Quarto, sem atributos. Um Tipo de quarto tem que implementar no mínimo um e no máximo vários Quartos e um Quarto pode ser implementado por no mínimo um e no máximo um Tipo de Quarto.
4. **possui**: relaciona Hospedagem a Forma de Pagamento, com o atributo *valor*. Uma Hospedagem tem que possuir no mínimo uma e no máximo várias Formas de Pagamento e uma Forma de Pagamento tem que estar associada a no mínimo um e no máximo várias Hospedagens.

5. **realiza**: relaciona Funcionário e Balcão, com o atributo *comissão*. Um Funcionário pode realizar no mínimo zero e no máximo vários atendimentos no Balcão e cada atendimento no Balcão tem que estar relacionado a no mínimo um e no máximo um Funcionário.

6. **limpa**: relaciona Funcionário e Quarto, sem atributos. Um Funcionário pode limpar no mínimo zero e no máximo vários Quartos e um Quarto tem que ser limpo por no mínimo um e no máximo um Funcionário.

7. **responsável**: relaciona Funcionário e Estoque, sem atributos. Um Funcionário tem que ser responsável por no mínimo um e no máximo um Estoque e um Estoque tem que ser responsabilide de no mínimo um e no máximo um Funcionário.

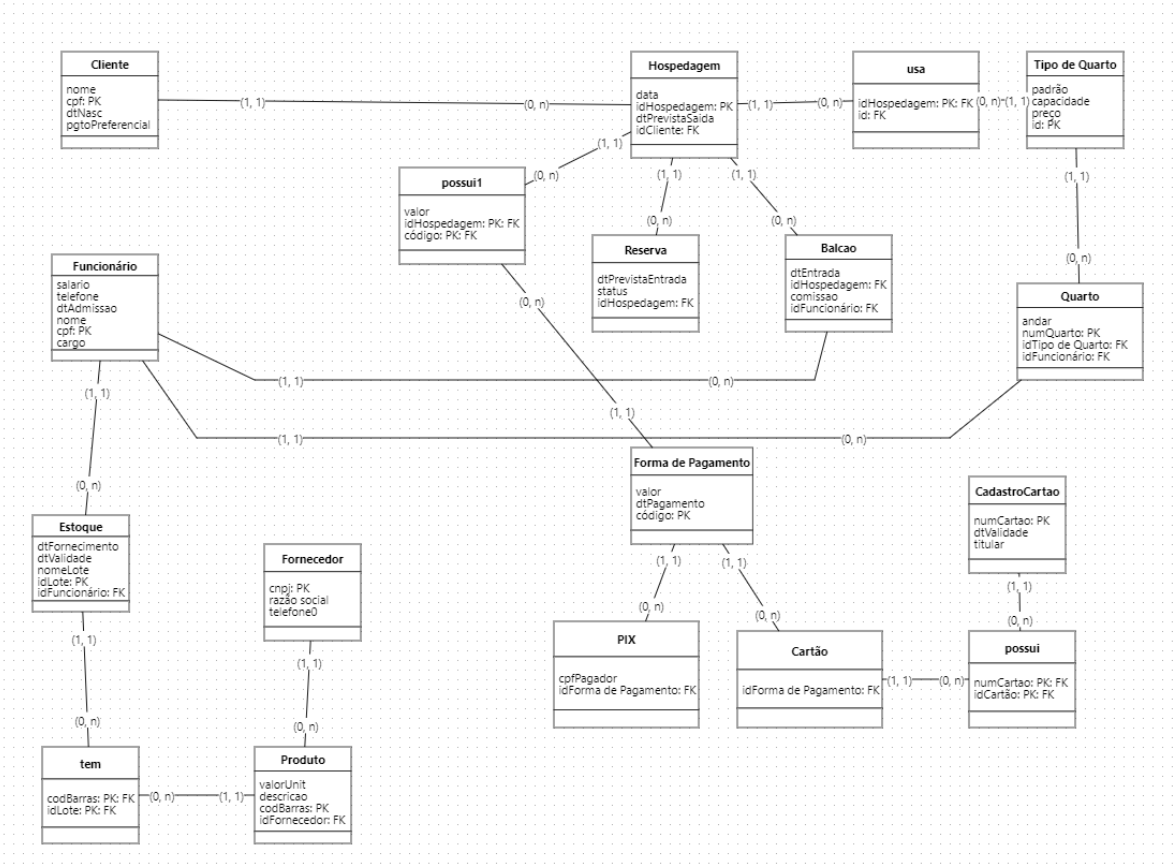
8. **tem**: relaciona Estoque e Produto, sem atributos. Um Estoque tem que ter no mínimo um e no máximo vários Produtos e um Produto tem que estar relacionado a no mínimo um e no máximo vários Estoques.

9. **vendido**: relaciona Produto e Fornecedor, sem atributos. Um Produto tem que ser vendido por no mínimo um e no máximo um Fornecedor e um Fornecedor tem que vender no mínimo um e no máximo vários Produtos.

10. **possui**: relaciona Cartão e Cadastro Cartão, sem atributos. Um Cartão tem que possuir no mínimo um e no máximo vários Cadastros e um Cadastro de Cartão tem que ter no mínimo um e no máximo vários Cartões.

3 Modelo Lógico

3.1 Modelagem Lógica



3.2 Mapeamento Entidade-Relacionamento

- **Cliente**(CPF, nome, dtNasc, pgtoPrf, telefone)
- **Hospedagem**(id, data, dtPreSaida, cpfCliente)
- **Reserva**(idHosp, dtPreEnt, status)
- **Balcão**(idHosp, dtEntrada, cpfFunc, comissao)
- **Tipo de Quarto**(id, padrão, capacidade, preço)
- **Quarto**(num, andar, idTipo, cpfFunc)
- **Funcionário**(cpf, nome, salário, telefone dtAdmissão, cargo)
- **Forma de Pagamento**(código, valor, dtPgto, cpfPagador,
- **CadastroCartao**(num, dtVal, titular)
- **Produto**(codBarras, descrição, valorU, cnpjForn)
- **Fornecedor**(cpnj, rzSocial)
- **Telefone**(cnpjForn, telefone)
- **Estoque**(id, nome, dtForne, dtVal, cpfFunc)
- **HospedagemUsaTipo**(idHosp, idTipo)
- **HospedagemPossuiPagamento**(idHosp, codPgto, valor)
- **EstoqueTemProduto**(idLote,codBarras)
- **PagamentoPossuiCadastro**(codPgto, numCartao)

3.3 Dicionário Lógico de Dados

Cliente: Relação que guarda os dados referentes aos clientes cadastrados				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
CPF	Representa o número de CPF de um Cliente	Character(11)	Character(11)	Chave Primária
Nome	Representa o nome de um Cliente.	Varchar(100)	Varchar(100)	Não nulo
dtNasc	Representa a data de nascimento de um Cliente	Date	Date	Não nulo
pgtoPrf	Representa o pagamento preferencial de um Cliente	Varchar(10)	Varchar(10)	Sem restrição
telefone	Representa o número de telefone de um Cliente	Varchar(10)	Varchar(10)	Não nulo

Hospedagem: Relação que guarda os dados comuns às hospedagens				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
id	Representa o número de identificação de uma Hospedagem	Integer	Números Inteiros Positivos	Chave Primária
Data	Representa a data em que uma Hospedagem foi registrada	Date	Date	Não nulo
DataPreSaida	Representa a data prevista para o término de uma hospedagem	Date	Date	Não nulo
cpfCliente	Representa o cpf de um Cliente que está associado àquela Hospedagem	Character(11)	Character(11)	Chave estrangeira referenciando Cpf na tabela CLIENTE

Reserva: Relação que guarda os dados específicos de uma Reserva				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
idHospedagem	Representa o número de identificação de uma Hospedagem	Integer	Números Inteiros Positivos	Chave Primária Chave estrangeira referenciando id na tabela Hospedagem
DataEntrada	Representa a data prevista para o inicio de uma hospedagem	Date	Date	Não nulo
Status	Representa o estado atual daquela Hospedagem	Varchar(10)	Varchar(10)	Não nulo

Balcão: Relação que guarda os dados de um atendimento feito no balcão				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
idHosp	Representa o número de identificação de uma Hospedagem	Integer	Números Inteiros Positivos	Chave Primária Chave estrangeira referenciando id na tabela Hospedagem
DataEntrada	Representa a data do inicio de uma hospedagem	Date	Date	Não nulo
CpfFunc	Representa o número de cpf de um funcionário	Character(11)	Character(11)	Não nulo e Chave estrangeira referenciando cpf na tabela Funcionário
Comissão	Representa o valor da comissão aplicada ao funcionário	Float(2)	Números reais positivos	Sem restrição

Tipo de Quarto: Relação que guarda os dados de um tipo de quarto				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
id	Representa o número de identificação de um tipo de quarto	Integer	Números Inteiros Positivos	Chave Primária
Padrão	Representa o padrão de um tipo de quarto	Varchar(10)	Varchar(10)	Não nulo
Capacidade	Representa o número de hóspedes suportado pelo quarto	Int	Números inteiros positivos	Não nulo
Preço	Representa o valor do tipo de quarto	Float(2)	Números reais positivos	Não nulo

Quarto: Relação que guarda os dados de um quarto				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
Num	Representa o número de um quarto	Integer	Números inteiros positivos	Chave Primária
Andar	Representa o andar que um quarto se localiza	Varchar(10)	Varchar(10)	Não nulo
idTipo	Representa o número de identificação do tipo do quarto	INTEGER	Números inteiros positivos	Chave estrangeira referenciando id na tabela Tipo de Quarto
cpfFunc	Representa o número de cpf do funcionário responsável pelo quarto	Character(11)	Character(11)	Chave estrangeira referenciando cpf na tabela funcionário

Funcionário: Relação que guarda os dados de um funcionário				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
CPF	Representa o número de cpf de um funcionário	Character(11)	Character(11)	Chave Primária
Nome	Representa o nome de um funcionário	Varchar(100)	Varchar(100)	Não nulo
Salário	Representa o valor do salário de um funcionário	Float(2)	Números reais positivos	Não nulo
Telefone	Representa o número de telefone de um funcionário	Varchar(10)	Varchar(10)	Não nulo
dtAdmissao	Representa a data de admissão de um funcionário	Date	Date	Não nulo
cargo	Representa o cargo de um funcionário	Varchar(20)	Varchar(20)	Não nulo

Forma de pagamento: Relação que guarda os dados dos pagamentos				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
Código	Representa o código do pagamento	Integer	Números Inteiros Positivos	Chave Primária
Valor	Representa o valor do pagamento	Float(2)	Números reais positivos	Não nulo
dtPgto	Representa a data que um pagamento foi registrado	Date	Date	Não nulo

Pix: Relação que guarda os dados dos pagamentos feitos no pix				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
codPagto	Representa o código do pagamento	Integer	Números Inteiros Positivos	Chave Estrangeira referenciando código na tabela Forma de Pagamento
cpfPagador	Representa o número de cpf do pagador	Character(11)	Character(11)	Não nulo

Cartão: Relação que guarda os dados dos pagamentos feitos com Cartão				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
codPagto	Representa o código do pagamento	Integer	Números Inteiros Positivos	Chave Primária e Chave Estrangeira referenciando código na tabela Forma de Pagamento

CadastroCartão: Relação que guarda os dados dos cadastros dos cartões				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
num	Representa o número do cartão	Varchar(25)	Varchar(25)	Chave Primária
dtVal	Representa a data de validade do cartão	Date	Date	Não nulo
titular	Representa o nome do titular do cartão	Varchar(100)	Varchar(100)	Não nulo

Produto: Relação que guarda os dados dos produtos				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
codBarras	Representa o código de barras de um produto	Varchar(50)	Varchar(50)	Chave Primária
descrição	Representa a descrição de um produto	Varchar(45)	Varchar(45)	Não nulo
valorU	Representa o valor de um produto	Float(2)	Float(2)	Não nulo
cnpjForn	Representa o CNPJ de um fornecedor	Varchar(45)	Varchar(45)	Chave estrangeira referenciando cnpj na tabela Fornecedor

Fornecedor: Relação que guarda os dados do Fornecedor				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
cnpj	Representa o CNPJ de um Fornecedor	Varchar(45)	Varchar(45)	Chave Primária
rzSocial	Representa o nome da razão social do Fornecedor	Varchar(100)	Varchar(100)	Não nulo

Telefone: Relação que guarda os números de telefone de um Fornecedor				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
cnpjForn	Representa o CNPJ de um Fornecedor	Varchar(45)	Varchar(45)	Chave estrangeira referenciando cnpj na tabela Fornecedor
telefone	Representa o número de telefone do Fornecedor	Varchar(10)	Varchar(10)	Não nulo

Estoque: Relação que guarda os dados de um Estoque				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
id	Representa o número de identificação de um estoque	Integer	Números Inteiros Positivos	Chave primária
nome	Representa o nome de determinado lote	Varchar(100)	Varchar(100)	Não nulo
dtForne	Representa a data de fornecimento de determinado lote	Date	Date	Não nulo
dtVal	Representa a data de validade de determinado lote	Date	Date	Não nulo
cpfFunc	Representa o número de CPF do funcionário responsável pelo estoque	Character(11)	Character(11)	Chave estrangeira referenciando cpf na tabela Funcionário

HospedagemUsaTipo: Relação que guarda os dados do relacionamento entre Hospedagem e Tipo de Quarto				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
idHosp	Representa o número de identificação da Hospedagem	Integer	Números Inteiros Positivos	Chave primária e chave estrangeira referenciando id na tabela Hospedagem
idTipo	Representa o número de identificação do Tipo de Quarto	Integer	Números Inteiros Positivos	Chave primária e chave estrangeira referenciando id na tabela Tipo de Quarto

HospedagemPossuiPagamento:Relação que guarda os dados do relacionamento entre Hospedagem e Pagamento				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
idHosp	Representa o número de identificação da Hospedagem	Integer	Números Inteiros Positivos	Chave primária e chave estrangeira referenciando id na tabela Hospedagem
codPgto	Representa o código de um pagamento	Integer	Números Inteiros Positivos	Chave primária e chave estrangeira referenciando código na tabela Forma de Pagamento

EstoqueTemProduto:Relação que guarda os dados do relacionamento entre Estoque e Produto				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
idLote	Representa o número de identificação do estoque	Integer	Números Inteiros Positivos	Chave primária e chave estrangeira referenciando id na tabela Estoque
codBarras	Representa o código de barras de um produto	Varchar(50)	Varchar(50)	Chave primária e chave estrangeira referenciando codBarras na tabela Produto

PagamentoPossuiCadastro:Relação que guarda os dados do relacionamento entre Pagamento e Cadastro Cartão				
Atributo	**Descrição**	**Tipo**	**Domínio**	**Restrição**
codPgto	Representa o código de um pagamento	Integer	Números Inteiros Positivos	Chave primária e chave estrangeira referenciando código na tabela Forma de Pagamento
numCartao	Representa o número de um cartão	Varchar(25)	Varchar(25)	Chave primária e chave estrangeira referenciando num na tabela CadastroCartão

4 Modelo Físico

4.1 Criando Relações

Relação Cliente

```
CREATE TABLE IF NOT EXISTS
Cliente
(
cpf CHARACTER(11) UNIQUE,
nome VARCHAR(100) NOT NULL,
dtNasc DATE NOT NULL,
pgtoPreferencial VARCHAR(10),
telefone VARCHAR(10) NOT NULL,
CONSTRAINT ClientePK PRIMARY
KEY(cpf)
);
```

Relação Funcionário

```
CREATE TABLE IF NOT EXISTS
Funcionário
(
cpf CHARACTER(11) UNIQUE,
nome VARCHAR(100) NOT NULL,
salario FLOAT(2),
telefone VARCHAR(10),
dtAdmissao DATE NOT NULL,
cargo VARCHAR(20) NOT NULL,
CONSTRAINT FuncionarioPK
PRIMARY KEY(cpf),
CONSTRAINT SalPositivo CHECK
(salario¿0)
);
```

Relação Hospedagem

```
CREATE TABLE IF NOT EXISTS
Hospedagem
(
idHosp INTEGER,
dataHosp DATE NOT NULL,
dtPreSaida DATE NOT NULL,
cpfCliente CHARACTER(11),
CONSTRAINT HospedagemPK
PRIMARY KEY (idHosp),
CONSTRAINT cpfClienteFK
FOREIGN KEY (cpfCliente)
REFERENCES Cliente(cpf)
);
```

Relação Reserva

```
CREATE TABLE IF NOT EXISTS
Reserva
(
idHosp INTEGER,
dtEntrada DATE NOT NULL,
status VARCHAR(10) NOT NULL,
CONSTRAINT ReservaPK PRIMARY
KEY (idHosp),
CONSTRAINT ReservaFK FOREIGN
KEY (idHosp)
REFERENCES Hospedagem(idHosp)
);
```

Relação Balcão

```
CREATE TABLE IF NOT EXISTS
Balcao
(
idHosp INTEGER,
dtEntrada DATE NOT NULL,
comissao FLOAT(2),
cpfFunc CHARACTER(11),
CONSTRAINT BalcaoPK PRIMARY
KEY(idHosp),
CONSTRAINT BalcaoFK FOREIGN
KEY(idHosp)
REFERENCES Hospedagem(idHosp),
CONSTRAINT CpfFuncFK FOREIGN
KEY(cpfFunc)
REFERENCES Funcionário(cpf)
);
```

Relação TipoQuarto

```
CREATE TABLE IF NOT EXISTS
TipoQuarto
(
idTipo INTEGER,
padrão VARCHAR(10) NOT NULL,
capacidade INT NOT NULL,
preço FLOAT(2) NOT NULL,
CONSTRAINT TipoPK PRIMARY
KEY(idTipo)
);
```

Relação Quarto

```
CREATE TABLE IF NOT EXISTS
Quarto
(
numQuarto INT,
andar VARCHAR(10) NOT NULL,
idTipo INTEGER,
cpfFunc CHARACTER(11),
CONSTRAINT QuartoPK PRIMARY
KEY(numQuarto),
CONSTRAINT TipoFK Foreign Key
(idTipo)
REFERENCES TipoQuarto(idTipo),
CONSTRAINT LimpaFK Foreign Key
(cpfFunc)
REFERENCES Funcionário(cpf)
);
```

Relação FormaPagamento

```
CREATE TABLE IF NOT EXISTS
FormaPagamento
(
código INTEGER,
valor FLOAT(2) NOT NULL,
dtPgto DATE NOT NULL,
CONSTRAINT CodigoPK PRIMARY
KEY(código)
);
```

Relação Pix

```
CREATE TABLE IF NOT EXISTS
PIX
(
cpfPagador CHARACTER(11),
codPgto INTEGER,
CONSTRAINT codPgtoPixPK
PRIMARY KEY(codPgto),
CONSTRAINT cpfPagadorFK
FOREIGN KEY(cpfPagador)
REFERENCES Cliente(cpf),
CONSTRAINT codPgtoFK FOREIGN
KEY(codPgto)
REFERENCES
FormaPagamento(código)
);
```

Relação Cartão

```
CREATE TABLE IF NOT EXISTS
Cartão
(
codPgto INTEGER,
CONSTRAINT codPgtoCartaoPK
PRIMARY KEY(codPgto),
CONSTRAINT codPgtoFK FOREIGN
KEY(codPgto)
REFERENCES
FormaPagamento(código)
);
```

Relação CadastroCartão

```
CREATE TABLE IF NOT EXISTS
CadastroCartao
(
num VARCHAR(25),
dtVal DATE NOT NULL,
titular VARCHAR(100) NOT NULL,
CONSTRAINT NumPK PRIMARY
KEY(num)
);
```

Relação Fornecedor

```
CREATE TABLE IF NOT EXISTS
Fornecedor
(
cnpj VARCHAR(45),
rzSocial VARCHAR(100) NOT NULL,
CONSTRAINT cnpjPK PRIMARY
KEY(cnpj)
);
```

Relação Telefone

```
CREATE TABLE IF NOT EXISTS
Telefone
(
cnpjForn VARCHAR(45),
telefone VARCHAR(10) NOT NULL
CONSTRAINT cnpjFornPK
PRIMARY KEY,
CONSTRAINT cnpjFornFK Foreign
Key(cnpjForn)
REFERENCES Fornecedor(cnpj)
);
```

Relação Produto

```
CREATE TABLE IF NOT EXISTS
Produto
(
codBarras VARCHAR(50),
descricao VARCHAR(45) NOT NULL,
valorU FLOAT(2) NOT NULL,
cnpjForn VARCHAR(45),
CONSTRAINT CodBarrasPK
PRIMARY KEY(codBarras),
CONSTRAINT cnpjFornFK
FOREIGN KEY(cnpjForn)
REFERENCES Fornecedor(cnpj)
);
```

Relação Estoque

```
CREATE TABLE IF NOT EXISTS
Estoque
(
idLote INTEGER,
nome VARCHAR(100) NOT NULL,
dtForne DATE NOT NULL,
dtVal DATE NOT NULL,
cpfFunc CHARACTER(11),
CONSTRAINT idLotePK PRIMARY
KEY(idLote),
CONSTRAINT cpfFuncFK FOREIGN
KEY(cpfFunc)
REFERENCES Funcionário(cpf)
);
```

Relação HospedagemUsaTipo

```
CREATE TABLE IF NOT EXISTS
HospedagemUsaTipo
(
idHosp INTEGER,
idTipo INTEGER,
CONSTRAINT idHospIdTipoPK
PRIMARY KEY(idHosp,idTipo),
CONSTRAINT idHospFK FOREIGN
KEY(idHosp)
REFERENCES Hospedagem(idHosp),
CONSTRAINT idTipoFK FOREIGN
KEY(idTipo)
REFERENCES TipoQuarto(idTipo)
);
```

Relação HospedagemPossuiPgto

```
CREATE TABLE IF NOT EXISTS
HospedagemPossuiPgto
(
idHosp INTEGER,
codPgto INTEGER,
valor FLOAT(2) NOT NULL,
CONSTRAINT idHospCodPgtoPK
PRIMARY KEY(idHosp, codPgto),
CONSTRAINT idHospFK FOREIGN
KEY(idHosp)
REFERENCES Hospedagem(idHosp),
CONSTRAINT codPgtoFK FOREIGN
KEY(codPgto)
REFERENCES
FormaPagamento(código)
);
```

Relação EstoqueTemProduto

```
CREATE TABLE IF NOT EXISTS
EstoqueTemProduto
(
idLote INTEGER,
codBarras VARCHAR(50),
valor FLOAT(2) NOT NULL,
CONSTRAINT idLoteCodBarrasPK
PRIMARY KEY(idLote,codBarras),
CONSTRAINT idLoteFK FOREIGN
KEY(idLote)
REFERENCES Estoque(idLote),
CONSTRAINT codBarrasFK
FOREIGN KEY(codBarras)
REFERENCES Produto(codBarras)
);
```

Relação PagamentoPossuiCadastro

```
CREATE TABLE IF NOT EXISTS
PagamentoPossuiCadastro
(
codPgto INTEGER,
numCartao VARCHAR(25),
CONSTRAINT
codPgtoNumCartaoPK PRIMARY
KEY(codPgto, numCartao),
CONSTRAINT codPgtoFK FOREIGN
KEY(codPgto)
REFERENCES
FormaPagamento(código),
CONSTRAINT numCartaoFK Foreign
Key(numCartao)
REFERENCES CadastroCartao(num)
);
```

4.2 Criando Visões

Uma visão que mostra quantos atendimentos cada recepcionista fez:

```
CREATE VIEW RecepcionistasAtendimentos AS
SELECT f.cpf, f.nome AS nomeFuncionario, COUNT(b.idHosp) AS
numHospedagensAtendidas
FROM Funcionario f
LEFT JOIN Balcao b ON f.cpf = b.cpfFunc
WHERE f.cargo = 'Recepcionista'
GROUP BY f.cpf, f.nome;
```

Uma visão que mostra as hospedagens e tipos de quarto utilizados:

```
CREATE VIEW DetalhesHospedagemTipoQuarto AS
SELECT h.idHosp, h.dataHosp, h.dtPreSaida, tq.padrão AS
tipoQuarto, tq.capacidade, tq.preço
FROM Hospedagem h
LEFT JOIN HospedagemUsaTipo hut ON h.idHosp = hut.idHosp
LEFT JOIN TipoQuarto tq ON hut.idTipo = tq.idTipo;
```

Uma visão que mostra os fornecedores e os produtos que eles fornecem:

```
CREATE VIEW FornecedorProdutos AS
SELECT f.cnpj, f.rzSocial AS nomeFornecedor, p.codBarras,
p.descricao AS descricaoProduto, p.valorU AS valorUnitario
FROM Fornecedor f
LEFT JOIN Produto p ON f.cnpj = p.cnpjForn;
```

Uma visão que mostra o estoque e o funcionário responsável por ele:

```
CREATE VIEW EstoqueFuncionario AS
SELECT e.idLote, e.nome AS nomeProduto, e.dtForne, e.dtVal,
f.nome AS nomeFuncionario
FROM Estoque e
LEFT JOIN Funcionario f ON e.cpfFunc = f.cpf;
```

Uma visão que mostra os telefones associados aos fornecedores:

```
CREATE VIEW EstoqueFuncionario AS
SELECT e.idLote, e.nome AS nomeProduto, e.dtForne, e.dtVal,
f.nome AS nomeFuncionario
FROM Estoque e
LEFT JOIN Funcionario f ON e.cpfFunc = f.cpf;
```

4.3 Criando Índices

Índices são utilizados para otimizar o desempenho de consultas

```
– Índice para a tabela Cliente no campo 'cpf'
CREATE INDEX idxCpf ON Cliente(cpf);

– Índice para a tabela Funcionario no campo 'cpf'
CREATE INDEX idxCpfFuncionario ON Funcionario(cpf);

– Índice para a tabela Hospedagem no campo 'idHosp'
CREATE INDEX idxIdHospHospedagem ON Hospedagem(idHosp);

– Índice para a tabela Reserva no campo 'idHosp'
CREATE INDEX idxIdHospReserva ON Reserva(idHosp);

– Índice para a tabela Balcao no campo 'idHosp'
CREATE INDEX idxIdHospBalcao ON Balcao(idHosp);

– Índice para a tabela TipoQuarto no campo 'idTipo'
CREATE INDEX idxIdTipoTipoquarto ON TipoQuarto(idTipo);

– Índice para a tabela Quarto no campo 'numQuarto'
CREATE INDEX idxNumQuartoQuarto ON Quarto(numQuarto);

– Índice para a tabela FormaPagamento no campo 'código'
CREATE INDEX idxCodigoFormapagamento ON
FormaPagamento(código);

– Índice para a tabela PIX no campo 'cpfPagador'
CREATE INDEX idxCpfPagadorPix ON PIX(cpfPagador);

– Índice para a tabela Cartão no campo 'codPgto'
CREATE INDEX idxCodPgtoCartao ON Cartão(codPgto);

– Índice para a tabela CadastroCartao no campo 'num'
CREATE INDEX idxNumCadastroCartao ON CadastroCartao(num);
```

Índices são utilizados para otimizar o desempenho de consultas

```
– Índice para a tabela Fornecedor no campo 'cnpj'
CREATE INDEX idxCnpjFornecedor ON Fornecedor(cnpj);

– Índice para a tabela Produto no campo 'codBarras'
CREATE INDEX idxCodBarrasProduto ON Produto(codBarras);

– Índice para a tabela Estoque no campo 'idLote'
CREATE INDEX idxIdLoteEstoque ON Estoque(idLote);

– Índice para a tabela HospedagemUsaTipo nos campos 'idHosp' e
'idTipo'
CREATE INDEX idxIdHospIdTipoHospedagemUsaTipo ON
HospedagemUsaTipo(idHosp, idTipo);

– Índice para a tabela HospedagemPossuiPgto nos campos 'idHosp'
e 'codPgto'
CREATE INDEX idxIdHospCodPgtoHospedagemPossuiPgto ON
HospedagemPossuiPgto(idHosp, codPgto);

– Índice para a tabela EstoqueTemProduto nos campos 'idLote' e
'codBarras'
CREATE INDEX idxIdLoteCodBarrasEstoqueTemProduto ON
EstoqueTemProduto(idLote, codBarras);

– Índice para a tabela PagamentoPossuiCadastro nos campos
'codPgto' e 'numCartao'
CREATE INDEX idxCodPgtoNumCartaoPagamentoPossuiCadastro
ON PagamentoPossuiCadastro(codPgto, numCartao);
```

4.4 Criando Procedimentos Armazenados

Atualizar Salário de um Funcionário

```
CREATE OR REPLACE PROCEDURE AtualizarSalarioFuncionario(
IN cpfAtual CHARACTER(11),
IN novoSalario FLOAT(2)
) AS
$$
BEGIN
UPDATE Funcionario SET salario = novoSalario WHERE cpf = cpfAtual;
END
$$
LANGUAGE plpgsql;
```

Atualizar a data de saída prevista de uma hospedagem

```
CREATE OR REPLACE PROCEDURE AtualizarDataPreSaidaHospedagem(
IN idHospAtual INTEGER,
IN novaDtPreSaida DATE
) AS
$$
BEGIN
UPDATE Hospedagem SET dtPreSaida = novaDtPreSaida WHERE idHosp = idHospAtual;
END
$$
LANGUAGE plpgsql;
```

Atualizar o valor de um produto no estoque

```
CREATE OR REPLACE PROCEDURE
AtualizarValorProdutoNoEstoque(
IN codBarrasAtual VARCHAR(50),
IN novoValor FLOAT(2)
) AS
$$
BEGIN
UPDATE Produto SET valorU = novoValor WHERE codBarras =
codBarrasAtual;
END
$$
LANGUAGE plpgsql;
```

Atualizar o valor de um produto no estoque

```
CREATE OR REPLACE PROCEDURE InserirTelefoneFornecedor(
IN cnpjForn VARCHAR(45),
IN telefone VARCHAR(10)
) AS
$$
BEGIN
INSERT INTO Telefone(cnpjForn, telefone)
VALUES (cnpjForn, telefone);
END
$$ LANGUAGE plpgsql;
```

4.5 Criando Gatilhos

Impede a exclusão de um cliente caso existam hospedagens associadas

```
CREATE OR REPLACE FUNCTION verificaHospedagensCliente()
RETURNS TRIGGER AS $$
BEGIN
IF EXISTS (SELECT 1 FROM Hospedagem WHERE cpfCliente =
OLD.cpf) THEN
RAISE EXCEPTION 'Não é possível excluir o cliente, pois existem
hospedagens
associadas a ele!';
END IF;
RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER restricaoExclusaoCliente
BEFORE DELETE ON Cliente
FOR EACH ROW
EXECUTE FUNCTION verificaHospedagensCliente();
```

Impede a inserção de um lote se a validade for anterior ao fornecimento

```
CREATE OR REPLACE FUNCTION verificaDatasEstoque()
RETURNS TRIGGER AS $$
BEGIN
IF NEW.dtVal < NEW.dtForne THEN
RAISE EXCEPTION 'Data de validade não pode ser anterior à data
de fornecimento!';
END IF;
RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER restricaoDataEstoque
BEFORE INSERT ON Estoque
FOR EACH ROW
EXECUTE FUNCTION verificaDatasEstoque();
```

Impede a alteração do CPF de um cliente após o cadastro inicial

```
CREATE OR REPLACE FUNCTION verificaAtualizacaoCpfCliente()
RETURNS TRIGGER AS $$
BEGIN
IF OLD.cpf <> NEW.cpf THEN
RAISE EXCEPTION 'Não é permitido alterar o CPF do cliente!';
END IF;
RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER restricaoAtualizacaoCpfCliente
BEFORE UPDATE ON Cliente
FOR EACH ROW
EXECUTE FUNCTION verificaAtualizacaoCpfCliente();
```

4.6 Povoando o Banco

Populando a tabela Cliente

```
INSERT INTO Cliente
(cpf, nome, dtNasc, pgtoPreferencial, telefone)
VALUES
('02335678301', 'Fabiano Andrade', '15-05-1998', 'Cartão',
'8334567890'),
('98765432307', 'Marilia Da Silva', '08-08-1980', 'PIX',
'9876543210'),
('45678912355', 'Theo José Da Silva', '09-05-1990', 'PIX',
'8891111111'),
('36925814706', 'João Pedro Andrade', '11-12-1997', 'Cartão',
'8362222822'),
('36125814789', 'Ross Geller', '11-12-1997', 'Cartão',
'8524227222');
```

Populando a tabela Funcionário

```
INSERT INTO Funcionario
(cpf, nome, salario, telefone, dtAdmissao, cargo)
VALUES
('11122233344', 'Maria Fernanda De Andrade Cesário', 3960.00,
'5555555555', '01-01-2023', 'Recepcionista'),
('44455566677', 'Eduarda Andrade', 2940.00, '6666666666', '01-01-
2023', 'Cozinheira'),
('77788899900', 'Phoebe Buffay', 1980.00, '7777777777', '01-01-
2023', 'Auxiliar de Limpeza'),
('12365478900', 'Rachel Green', 1980.00, '8888888888', '01-01-
2023', 'Auxiliar de Limpeza'),
('14366478910', 'Monica Eustace', 2940.00, '999999999', '01-01-
2023', 'Cozinheira');
```

Populando a tabela TipoQuarto

```
INSERT INTO TipoQuarto
(idTipo, padrão, capacidade, preço)
VALUES
(1, 'Standard', 2, 200.00),
(2, 'Master', 4, 400.00),
(3, 'Deluxe', 6, 600.00);
```

Populando a tabela Quarto

```
INSERT INTO Quarto
(numQuarto, andar, idTipo, cpfFunc)
VALUES
(101, 'Primeiro', 1, '77788899900'),
(102, 'Primeiro', 1, '77788899900'),
(201, 'Segundo', 2, '12365478900'),
(202, 'Segundo', 2, '12365478900'),
(301, 'Terceiro', 3, '77788899900'),
(302, 'Terceiro', 3, '12365478900');
```

Populando a tabela Hospedagem

```
INSERT INTO Hospedagem
(idHosp, dataHosp, dtPreSaida, cpfCliente)
VALUES
(1, '1-11-2023', '3-11-2023', '02335678301'),
(2, '2-11-2023', '4-11-2023', '98765432307'),
(3, '12-11-2023', '14-11-2023', '45678912355'),
(4, '15-11-2023', '16-11-2023', '36925814706'),
(5, '16-11-2023', '20-11-2023', '36125814789');
```

Populando a tabela Reserva

```
INSERT INTO Reserva
(idHosp, dtEntrada, status)
VALUES
(1, '2023-11-1', 'Confirmada'),
(5, '2023-11-16', 'Pendente');
```

Populando a tabela Balcão

```
INSERT INTO Balcao
(idHosp, dtEntrada, comissao, cpfFunc)
VALUES
(2, '2023-11-2', 20.00, '11122233344'),
(3, '2023-11-12', 20.00, '11122233344'),
(4, '2023-11-15', 20.00, '11122233344');
```

Populando a tabela FormaPagamento

```
INSERT INTO FormaPagamento
(código, valor, dtPgto)
VALUES
(1, 600.00, '2023-11-1'),
(2, 400.00, '2023-11-2'),
(3, 400.00, '2023-11-12'),
(4, 200.00, '2024-11-15'),
(5, 600.00, '2024-11-16');
```

Populando a tabela Pix

```
INSERT INTO PIX
(cpfPagador, codPgto)
VALUES
('02335678301', 1),
('36125814789', 5);
```

Populando a tabela Cartão

```
INSERT INTO Cartão (codPgto) VALUES (2), (3);
```

Populando a tabela CadastroCartão

```
INSERT INTO CadastroCartao
(num, dtVal, titular)
VALUES
('1111222233334444', '2028-08-11', 'Theo José Da Silva'),
('5555666677778888', '2028-06-30', 'Marilia Da Silva');
```

Populando a tabela Fornecedor

```
INSERT INTO Fornecedor
(cnpj, rzSocial)
VALUES
('12345678901234', 'Copal Distribuidora'),
('98765432109876', 'PMG Atacadista de Alimentos e Bebidas'),
('53762432309671', 'Aliança Distribuidora'),
('43755435109870', 'Vital Atacadista'),
('78765432109872', 'Triunfo Alimentícia LTDA');
```

Populando a tabela Telefone

```
INSERT INTO Telefone
(cnpjForn, telefone)
VALUES
('12345678901234', '9115161711'),
('98765432109876', '8395999798'),
('53762432309671', '8891011111'),
('43755435109870', '9594996997'),
('78765432109872', '9144996934');
```

Populando a tabela Produto

```
INSERT INTO Produto
(codBarras, descricao, valorU, cnpjForn)
VALUES
('ABCDE12345', 'Ovo', 1.00, '12345678901234'),
('FGHIC6789G', 'Café', 5.00, '98765432109876'),
('BBCDE12345', 'Macarrão', 1.00, '12345678901234'),
('FGHIJ67890', 'Pão', 4.00, '98765432109876'),
('SGQIJ67898', 'Cuscuz', 1.00, '98765432109876');
```

Populando a tabela Estoque

```
INSERT INTO Estoque
(idLote, nome, dtForne, dtVal, cpfFunc)
VALUES
(1, 'Lote de Ovos', '2023-11-01', '2023-12-01', '14366478910'),
(2, 'Lote de Café', '2023-10-15', '2024-10-15', '44455566677'),
(3, 'Lote de Macarrão', '2023-10-15', '2024-10-15', '14366478910'),
(4, 'Lote de Pão', '2023-10-15', '2023-10-25', '44455566677'),
(5, 'Lote de Cuscuz', '2023-10-15', '2023-12-25', '44455566677');
```

Populando a tabela HospedagemUsaTipo

```
INSERT INTO HospedagemUsaTipo (idHosp, idTipo)
VALUES (1, 3), (2, 2), (3, 2), (4, 1), (5, 3);
```

Populando a tabela HospedagemPossuiPgto

```
INSERT INTO HospedagemPossuiPgto (idHosp, codPgto)
VALUES (1, 1), (2, 2), (3, 3), (4, 4), (5, 5);
```

Populando a tabela EstoqueTemProduto

```
INSERT INTO EstoqueTemProduto (idLote, codBarras)
VALUES
(1, 'ABCDE12345'),
(2, 'FGHIC6789G'),
(3, 'BBCDE12345'),
(4, 'FGHIJ67890'),
(5, 'SGQIJ67898');
```

Populando a tabela PagamentoPossuiCadastro

```
INSERT INTO PagamentoPossuiCadastro (codPgto, numCartao)
VALUES
(2, '1111222233334444'),
(3, '5555666677778888');
```

4.7 Consultas

Busca detalhes das hospedagens com informações do cliente associado.
SELECT h.idHosp, h.dataHosp, h.dtPreSaida, c.nome AS nomeCliente, c.cpf AS cpfCliente FROM Hospedagem h INNER JOIN Cliente c ON h.cpfCliente = c.cpf;

Busca informações sobre os quartos e os funcionários associados a eles
SELECT tq.idTipo, tq.padrão, tq.capacidade, tq.preço, q.numQuarto, q.andar, f.nome AS funcionarioResponsavel FROM TipoQuarto tq INNER JOIN Quarto q ON tq.idTipo = q.idTipo LEFT JOIN Funcionario f ON q.cpfFunc = f.cpf;

Busca funcionários que não possuem número de telefone
SELECT * FROM Funcionario WHERE telefone IS NULL;

Busca informações sobre os quartos que não possuem um responsável
SELECT * FROM Quarto WHERE cpfFunc IS NULL;

Busca clientes cujos nomes contenham 'Andrade'
SELECT * FROM Cliente WHERE nome LIKE '%Andrade%';

Busca produtos cujas descrições comecem com a letra M
SELECT * FROM Produto WHERE descricao LIKE 'M

Busca funcionários em ordem decrescente de salário

```
SELECT * FROM Funcionario ORDER BY salario DESC;
```

Busca produtos ordenados pelo valor unitário em ordem crescente

```
SELECT * FROM Produto ORDER BY valorU ASC;
```

Busca o funcionário que possui o maior salário

```
SELECT *
FROM Funcionario
WHERE salario = (
SELECT MAX(salario)
FROM Funcionario
);
```

Busca o produto com a maior quantidade em estoque

```
SELECT p.*
FROM Produto p
WHERE codBarras IN (
SELECT codBarras
FROM EstoqueTemProduto
GROUP BY codBarras
ORDER BY COUNT(*) DESC
LIMIT 1
);
```

Busca o funcionário com salário superior à média de salários

```
SELECT *
FROM Funcionario f
WHERE salario > (
SELECT AVG(salario)
FROM Funcionario
);
```

Busca o funcionário que realizou o maior número de hospedagens

```
SELECT f.*
FROM Funcionario f
WHERE f.cpf IN (
SELECT b.cpfFunc
FROM Balcao b
GROUP BY b.cpfFunc
ORDER BY COUNT(*) DESC
LIMIT 1
);
```

Busca pagamentos que não estão associados a Pix nem cartão

```
SELECT *
FROM FormaPagamento fp
WHERE NOT EXISTS (
SELECT *
FROM PIX p
WHERE p.codPgto = fp.código
) AND NOT EXISTS (
SELECT *
FROM Cartão c
WHERE c.codPgto = fp.código
);
```

Busca os pagamentos utilizados em PIX ou Cartão nas reservas
SELECT codPgto FROM HospedagemPossuiPgto WHERE codPgto IN (SELECT codPgto FROM PIX) UNION SELECT codPgto FROM HospedagemPossuiPgto WHERE codPgto IN (SELECT codPgto FROM Cartão);

Busca osCNPJs de fornecedores sem produtos no estoque
SELECT cnpj FROM Fornecedor EXCEPT SELECT cnpjForn FROM Produto WHERE codBarras IN (SELECT codBarras FROM EstoqueTemProduto);