

Gestão de Reservas para Hotéis

1. Introdução

Este projeto tem como objetivo desenvolver um sistema de gestão de reservas para hotéis, utilizando um banco de dados relacional. O sistema permite o cadastro e gerenciamento de hotéis, quartos, clientes, reservas e pagamentos, garantindo controle eficiente sobre a ocupação e faturamento.

2. Justificativa

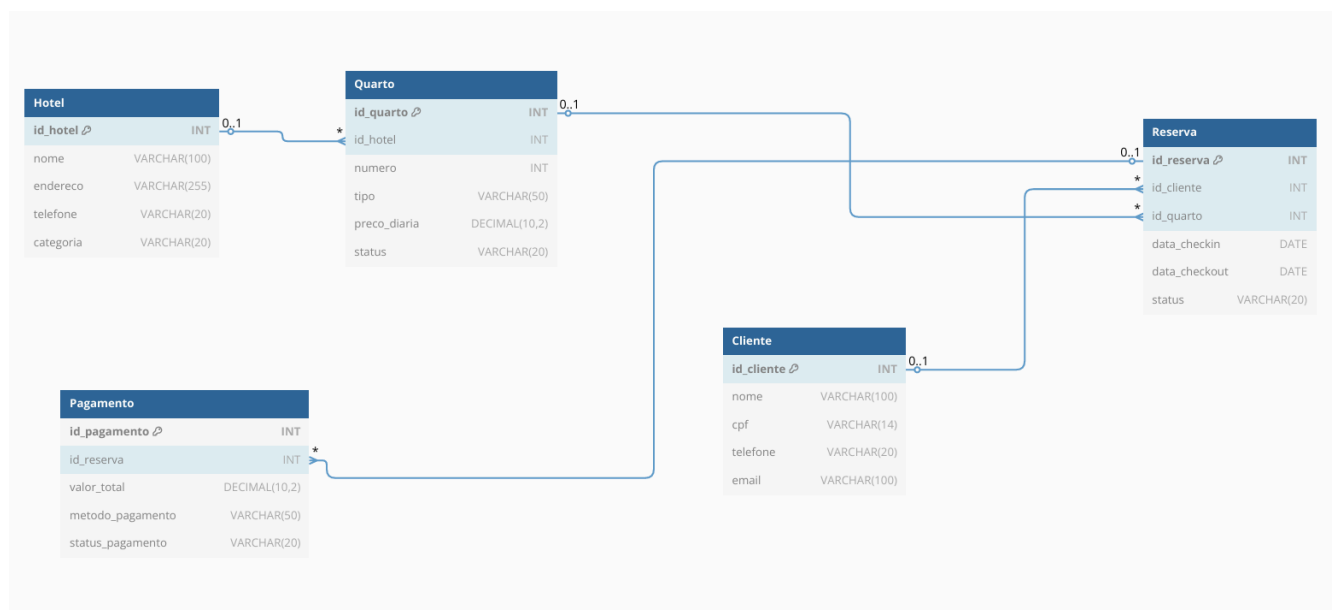
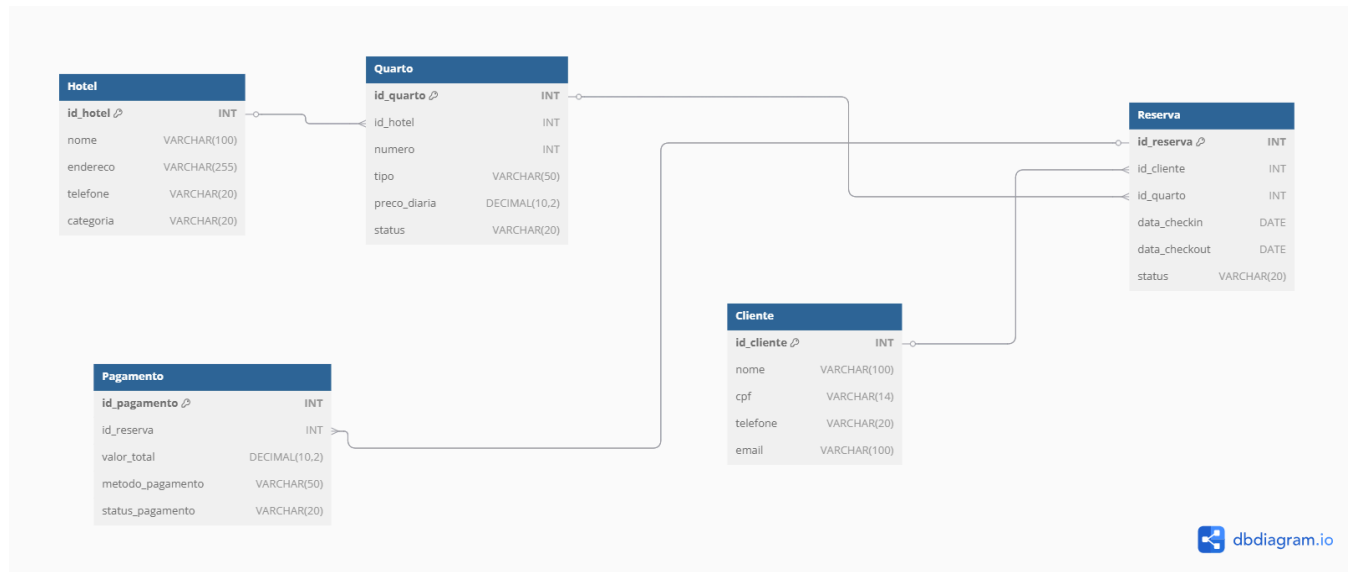
A gestão eficiente de reservas é essencial para hotéis de qualquer porte. O uso de um banco de dados facilita a organização das informações, melhora a experiência do cliente e otimiza a tomada de decisão com base em relatórios detalhados.

3. Modelagem do Banco de Dados

O banco de dados foi projetado utilizando um modelo relacional, contendo as seguintes tabelas principais:

- **Hotel:** Armazena informações sobre os hotéis cadastrados.
- **Quarto:** Contém detalhes dos quartos disponíveis nos hotéis.
- **Cliente:** Registra os clientes que realizam reservas.
- **Reserva:** Gerencia as reservas feitas pelos clientes.
- **Pagamento:** Controla os pagamentos das reservas.

3.1. Diagrama Entidade-Relacionamento (DER)



4. Funcionalidades Implementadas

O sistema permite as seguintes operações:

- Cadastro, atualização, remoção e listagem de hotéis, clientes, quartos, reservas e pagamentos.
- Consultas avançadas utilizando SQL JOIN e filtros para relatórios de faturamento, taxa de ocupação e clientes frequentes.

5. Consultas SQL Utilizadas

Abaixo estão descritas algumas consultas SQL avançadas utilizadas para geração de relatórios e análise de dados:

5.1. Listar todas as reservas com informações do cliente e do hotel

```
SELECT
    r.id_reserva, c.nome AS cliente, h.nome AS hotel, q.numero AS quarto,
    r.data_checkin, r.data_checkout, r.status
FROM reserva r
JOIN cliente c ON r.id_cliente = c.id_cliente
JOIN quarto q ON r.id_quarto = q.id_quarto
JOIN hotel h ON q.id_hotel = h.id_hotel
ORDER BY r.data_checkin;
```

♦ **Descrição:** Lista todas as reservas, trazendo detalhes do cliente, hotel e número do quarto.

5.2. Listar quartos disponíveis em um hotel específico

```
SELECT id_quarto, numero, tipo, preco_diaria
FROM quarto
WHERE id_hotel = %s AND status = 'Disponível'
ORDER BY preco_diaria;
```

♦ **Descrição:** Retorna todos os quartos disponíveis em um hotel específico, ordenando pelo preço.

5.3. Relatório de faturamento dos hotéis

```
SELECT h.nome AS hotel, SUM(p.valor_total) AS total_faturado
FROM pagamento p
JOIN reserva r ON p.id_reserva = r.id_reserva
JOIN quarto q ON r.id_quarto = q.id_quarto
JOIN hotel h ON q.id_hotel = h.id_hotel
WHERE p.status_pagamento = 'Pago'
GROUP BY h.nome
ORDER BY total_faturado DESC;
```

♦ **Descrição:** Calcula o faturamento total de cada hotel, considerando apenas pagamentos confirmados.

5.4. Listar clientes que fizeram mais de X reservas

```
SELECT c.nome, COUNT(r.id_reserva) AS total_reservas
FROM cliente c
JOIN reserva r ON c.id_cliente = r.id_cliente
GROUP BY c.nome
HAVING COUNT(r.id_reserva) >= %s
ORDER BY total_reservas DESC;
```

♦ **Descrição:** Retorna os clientes mais frequentes, ou seja, aqueles que fizeram pelo menos um número mínimo de reservas.

5.5. Relatório de ocupação de um hotel específico

```
SELECT
    h.nome AS hotel,
    COUNT(r.id_reserva) AS total_reservas,
    ROUND((COUNT(r.id_reserva) * 100.0 / COUNT(q.id_quarto)), 2) AS taxa_ocupacao
FROM hotel h
JOIN quarto q ON h.id_hotel = q.id_hotel
LEFT JOIN reserva r ON q.id_quarto = r.id_quarto
WHERE h.id_hotel = %s
GROUP BY h.nome;
```

♦ **Descrição:** Calcula a taxa de ocupação (%) de um hotel, baseando-se no número de reservas e quantidade de quartos.

6. Resultados e Análises

- Testes foram realizados para validar a integridade dos dados e a eficiência das consultas.
- Os relatórios gerados permitem uma visão clara do desempenho do hotel, ocupação e receitas.
- As funcionalidades podem ser testadas e confirmadas ao se executar o código conforme a instrução

7. Instruções de Execução do Código

7.1. Requisitos

Antes de executar o projeto, certifique-se de ter os seguintes requisitos instalados:

- **Python 3.x**
- **PostgreSQL**
- Biblioteca `psycopg2` (para conexão com o banco de dados)

Para instalar as dependências, execute:

```
pip install psycopg2
```

7.2. Configuração do Banco de Dados

1. Acesse o PostgreSQL e crie o banco de dados:

```
CREATE DATABASE hotel;
```

2. Conecte-se ao banco e execute o script SQL de criação das tabelas.
3. Configure a conexão no arquivo `db_connection.py`, alterando os parâmetros conforme seu ambiente.

7.3. Executando o Sistema

1. Execute o script para criar as tabelas:

```
python db_operations.py
```

2. Teste as operações CRUD rodando:

```
python test_crud.py
```

3. Para visualizar os relatórios, execute as funções disponíveis no `db_operations.py` ou crie um script específico para chamar as consultas avançadas.

8. Conclusão

O projeto demonstrou a importância do uso de bancos de dados na gestão de hotéis, otimizando a organização das informações e facilitando a tomada de decisões estratégicas.

