



**UNIVERSIDADE  
E D U A R D O  
MONDLANE**

**Faculdade de Engenharia**

**Departamento De Engenharia Electrotécnica**

**2º ano –Laboral ( Eng. Informática)**

**Base de Dados e POO1**

**Tema:**

**Sistema de Gestão de Dados de um Hotel**

**Discentes :**

Alfaica Keen Anastácio

Momade, Laiza Jhazila Ali

**Corpo Docente :**

Regente:

Engº. Cristiliano Maculuve

Maputo, Novembro de 2024

# Índice

1. Introdução .....	3
2. Objectivos .....	4
1.1. Geral: .....	4
1.2. Específicos: .....	4
Base de Dados I .....	5
3. Minimundo .....	5
4. Diagrama Entidade Relacionamento .....	6
5. Mapeamento .....	7
6. Normalização .....	7
7. Inserção de dados .....	9
8. Dicionário de dados .....	11
9. Inserção de dados .....	13
10. Consultas .....	14
11. Álgebra Relacional .....	16
12. Implementação de Triggers .....	17
13. Implementação de procedimentos/funções .....	18
Base de Dados II .....	20
14. Transações relevantes que podem ocorrer na Base de Dados do Hotel .....	20
15. Transações escritas em linguagem “Read” e “Write” .....	20
16. Análise de Volume .....	24
17.4. Propostas de desnormalização .....	28
18.2. Propostas índice .....	31
19. Conclusão .....	33
20. Bibliografia .....	34

## **1. Introdução**

Com o crescimento constante do turismo em Moçambique, os estabelecimentos hoteleiros enfrentam desafios crescentes em termos de eficiência operacional e gestão de clientes. A administração eficaz de reservas, a alocação de quartos, e a oferta de serviços personalizados são fundamentais para a satisfação dos hóspedes e a sustentabilidade dos negócios. Nesse contexto, um sistema de gestão hoteleira se torna imprescindível, permitindo a centralização das informações e a automação de processos.

Este projecto propõe o desenvolvimento de um sistema de gestão hoteleira que atende às necessidades específicas dos hotéis em Moçambique. O sistema abrange diversas entidades, incluindo Hotel, Cliente, Reserva, Quarto, Funcionário, Restaurante, e Pacote, cada uma com seus respectivos atributos e relacionamentos. Através da implementação de um banco de dados relacional, o sistema permitirá um gerenciamento eficiente das operações do hotel, desde a reserva até o check-out, além de possibilitar a oferta de serviços adicionais que atendam às expectativas dos clientes.

A adoção deste sistema não apenas melhora a experiência dos hóspedes, mas também oferece aos gestores uma ferramenta poderosa para a análise de dados e a tomada de decisões estratégicas. Em última análise, este projecto visa contribuir para a melhoria da qualidade dos serviços hoteleiros em Moçambique, promovendo a satisfação do cliente e a lucratividade dos estabelecimentos.

## **2. Objectivos**

### **1.1. Geral:**

- Desenvolver um sistema de gerenciamento de banco de dados para um ambiente hoteleiro, que centralize e optimize a gestão de informações relacionadas a reservas, quartos, clientes, funcionários, serviços adicionais e pacotes promocionais, promovendo eficiência operacional e uma melhor experiência para os clientes.

### **1.2. Específicos:**

- Gerenciar eficientemente recursos e estoques (implementar funcionalidades para monitorar e gerenciar eficientemente recursos como quartos disponíveis, serviços de restaurante etc).
- Centralizar informações dos hóspedes (criar um sistema centralizado para armazenar informações detalhadas sobre os hóspedes como estadias, preferências de quarto etc).
- Personalizar a experiência do hóspede.
- Garantir a melhor organização por parte administrativa e digitalizar o processo de gerenciamento de reservas dos clientes do hotel.
- Fornecer um SGBD para anotação e consulta de dados do hotel.

# Base de Dados I

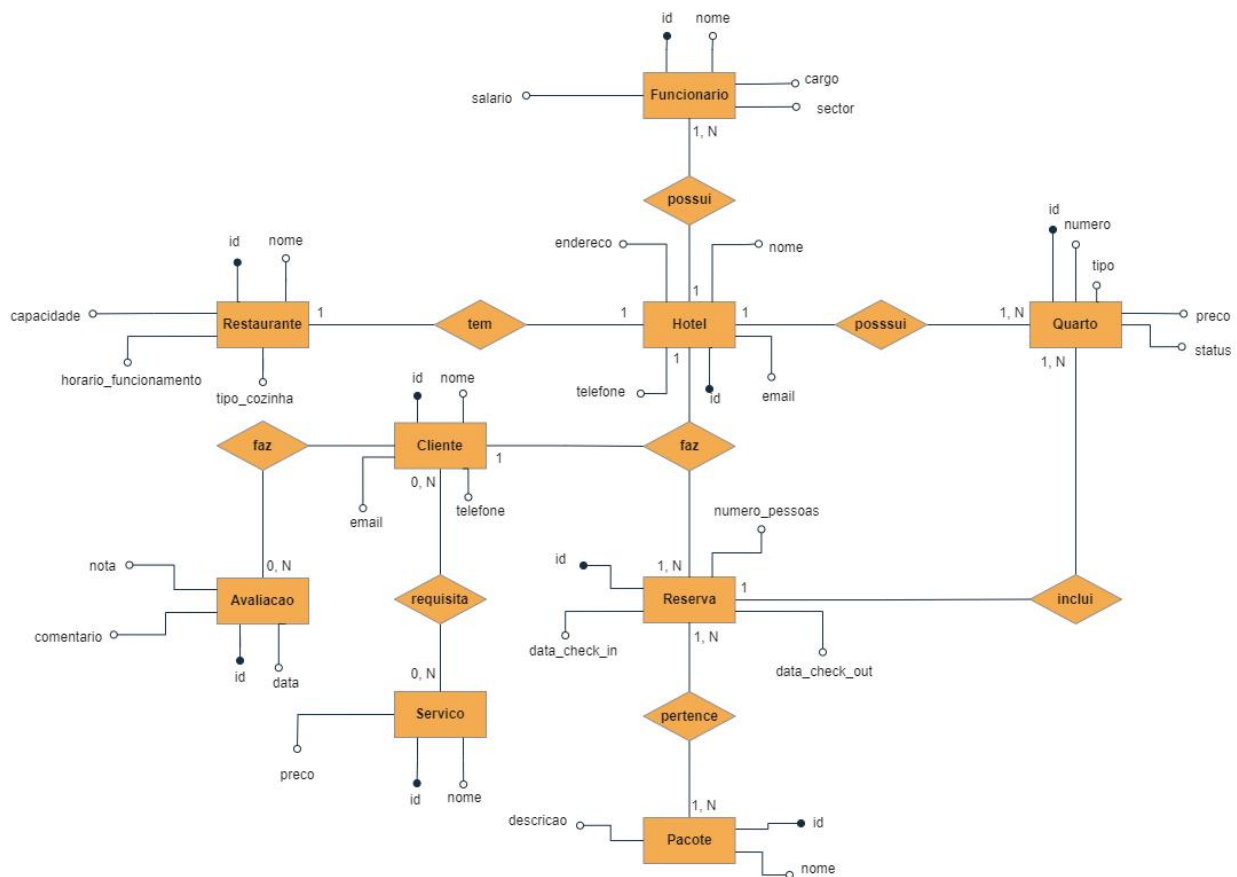
## 3. Minimundo

O Hotel representa o estabelecimento hoteleiro em si, com informações como nome, endereço, telefone e e-mail, essenciais para sua identificação e contato. Um hotel pode atender a diversos clientes, estabelecendo uma relação de um para muitos. Cliente refere-se aos indivíduos que buscam os serviços do hotel. O cliente é caracterizado por atributos como nome, sobrenome, documento de identificação, e-mail e telefone. Reserva, registra as solicitações de hospedagem feitas pelos clientes. Inclui dados como data de check-in, data de check-out, número de pessoas e tipo de quarto desejado. Funcionário engloba os colaboradores do hotel, desempenhando diversas funções administrativas e de atendimento. Seus atributos incluem nome, sobrenome, cargo, setor e salário. Quarto representa os espaços de alojamento disponíveis para os hóspedes. Cada quarto é identificado por um número e possui atributos como tipo, preço e status (ocupado ou disponível). Restaurante refere-se aos estabelecimentos de alimentação presentes no hotel. Possui informações como nome, capacidade, tipo de cozinha e horário de funcionamento.

- Cada cliente pode realizar várias reservas em um hotel, enquanto cada reserva está associada a apenas um cliente (1:N e 1:1, respectivamente);
- As reservas podem envolver um ou mais quartos, e um quarto pode ser associado a uma ou mais reservas, configurando uma relação de muitos para muitos, cada reserva pode estar associada a um pacote de hospedagem específico oferecido pelo hotel, isso permite que o hotel personalize a experiência dos hóspedes de acordo com o pacote escolhido os pacotes podem ser, pacote de lua de mel, pacote família pacote fim de semana e pacote individual ou solo.
- Os funcionários trabalham em um único hotel, mas cada hotel pode contar com vários funcionários;
- Cada quarto pertence a um único hotel, com a possibilidade de um hotel possuir vários quartos;
- Um hotel pode abrigar no máximo um restaurante, configurando uma relação de um para um

- O hotel possui serviços adicionais ,estes representam serviços extras oferecidos pelo hotel aos hóspedes, como spa, academia, transporte, serviços de evento (reservado por um hóspede / cliente envolve um espaço específico, um numero específico de pessoas entre outros, pode ser eventos sociais e festas privadas, eventos corporativos e de networking, casamentos e banquetes, organização de conferências e reuniões etc) os clientes podem solicitar e utilizar diferentes serviços adicionais durante sua estadia.
- Cada cliente pode deixar zero ou mais comentários e avaliações sobre sua experiência.

#### 4. Diagrama Entidade Relacionamento



## 5. Mapeamento

**Hotel** (id\_hotel, nome, endereço, telefone, email)

**Cliente** (id\_cliente, nome, email, telefone)

**Reserva** (id\_reserva, data\_check\_in, data\_check\_out, numero\_pessoas, tipo\_quarto, #id\_cliente)

**Quarto** (id\_quarto, numero\_quarto, tipo, preco, status, #id\_hotel)

**Funcionario** (id\_funcionario, nome, sobrenome, cargo, sector, salario, #id\_hotel)

**Restaurante** (id\_restaurante, nome, capacidade, tipo\_cozinha, horário\_funcionamento, #id\_hotel)

**Pacote** (id\_pacote, nome, descrição)

**Reserva\_Pacote** (#id\_reserva, #id\_pacote)

**Servico** (id\_servico, nome, preco)

**Cliente\_Servico** (#id\_cliente, #id\_servico)

**Avaliacao** (id\_avaliacao, data, nota, comentário, #id\_cliente)

## 6. Normalização

### 6.1. Primeira Forma Normal (1FN)

As tabelas já estão na 1FN pois não existem tabelas aninhadas

### 6.2. Segunda Forma Normal (2FN)

As tabelas já se encontram na 2FN pois não temos dependências parciais, visto que, nas tabelas com chaves primárias compostas as colunas dependem de toda chave.

### 6.3. Terceira Forma Normal (3FN)

**Hotel** (id\_hotel, nome, endereço, telefone, email)

**Cliente** (id\_cliente, nome, email, telefone)

**Reserva** (id\_reserva, data\_check\_in, data\_check\_out, numero\_pessoas, tipo\_quarto, #id\_cliente)

**Quarto** (id\_quarto, numero\_quarto, tipo, preco, status, #id\_hotel)

**Funcionario** (id\_funcionario, nome, sobrenome, cargo, sector, salario, #id\_hotel)

**Restaurante** (id\_restaurante, nome, capacidade, tipo\_cozinha, horário\_funcionamento, #id\_hotel)

**Pacote** (id\_pacote, nome, descrição)

**Reserva\_Pacote** (#id\_reserva, #id\_pacote)

**Servico** (id\_servico, nome, preco)

**Cliente\_Servico** (#id\_cliente, #id\_servico)

**Avaliacao** (id\_avaliacao, data, nota, comentário, #id\_cliente)

## 7. Criação de tabelas

```
XAMPP for Windows - mysql
MariaDB [hotel]> CREATE TABLE Hotel (
->   hotel_id INT PRIMARY KEY AUTO_INCREMENT,
->   nome VARCHAR(255) NOT NULL,
->   endereco VARCHAR(255) NOT NULL,
->   telefone VARCHAR(15) NOT NULL,
->   email VARCHAR(255) NOT NULL
-> );
Query OK, 0 rows affected (0.009 sec)

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Cliente (
->   cliente_id INT PRIMARY KEY AUTO_INCREMENT,
->   nome VARCHAR(100) NOT NULL,
->   sobrenome VARCHAR(100) NOT NULL,
->   documento_identificacao VARCHAR(50) NOT NULL,
->   email VARCHAR(255) NOT NULL,
->   telefone VARCHAR(15) NOT NULL
-> );
Query OK, 0 rows affected (0.005 sec)
```



```

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Reserva (
->     reserva_id INT PRIMARY KEY AUTO_INCREMENT,
->     data_check_in DATE NOT NULL,
->     data_check_out DATE NOT NULL,
->     numero_pessoas INT NOT NULL,
->     tipo_quarto VARCHAR(50) NOT NULL,
->     cliente_id INT,
->     FOREIGN KEY (cliente_id) REFERENCES Cliente(cliente_id)
-> );
Query OK, 0 rows affected (0.008 sec)

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Quarto (
->     quarto_id INT PRIMARY KEY AUTO_INCREMENT,
->     numero_quarto INT NOT NULL UNIQUE,
->     tipo VARCHAR(50) NOT NULL,
->     preco DECIMAL(10, 2) NOT NULL,
->     status ENUM('Ocupado', 'Disponível') NOT NULL,
->     hotel_id INT,
->     FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id)
-> );
Query OK, 0 rows affected (0.006 sec)

```

```

XAMPP for Windows - mysql
MariaDB [hotel]> CREATE TABLE Funcionario (
->     funcionario_id INT PRIMARY KEY AUTO_INCREMENT,
->     nome VARCHAR(100) NOT NULL,
->     sobrenome VARCHAR(100) NOT NULL,
->     cargo VARCHAR(50) NOT NULL,
->     setor VARCHAR(50) NOT NULL,
->     salario DECIMAL(10, 2) NOT NULL,
->     hotel_id INT,
->     FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id)
-> );
Query OK, 0 rows affected (0.008 sec)

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Restaurante (
->     restaurante_id INT PRIMARY KEY AUTO_INCREMENT,
->     nome VARCHAR(100) NOT NULL,
->     capacidade INT NOT NULL,
->     tipo_cozinha VARCHAR(100) NOT NULL,
->     horario_funcionamento VARCHAR(50) NOT NULL,
->     hotel_id INT UNIQUE,
->     FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id)
-> );
Query OK, 0 rows affected (0.006 sec)

```

```

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Pacote (
->     pacote_id INT PRIMARY KEY AUTO_INCREMENT,
->     nome VARCHAR(100) NOT NULL,
->     descricao TEXT NOT NULL
-> );
Query OK, 0 rows affected (0.006 sec)

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Reserva_Pacote (
->     reserva_id INT,
->     pacote_id INT,
->     PRIMARY KEY (reserva_id, pacote_id),
->     FOREIGN KEY (reserva_id) REFERENCES Reserva(reserva_id),
->     FOREIGN KEY (pacote_id) REFERENCES Pacote(pacote_id)
-> );
Query OK, 0 rows affected (0.005 sec)

```

```

XAMPP for Windows - mysql
MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Servico_Adicional (
->     servico_id INT PRIMARY KEY AUTO_INCREMENT,
->     nome VARCHAR(100) NOT NULL,
->     preco DECIMAL(10, 2) NOT NULL
-> );
Query OK, 0 rows affected (0.006 sec)

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Cliente_Servico (
->     cliente_id INT,
->     servico_id INT,
->     PRIMARY KEY (cliente_id, servico_id),
->     FOREIGN KEY (cliente_id) REFERENCES Cliente(cliente_id),
->     FOREIGN KEY (servico_id) REFERENCES Servico_Adicional(servico_id)
-> );
Query OK, 0 rows affected (0.005 sec)

MariaDB [hotel]>
MariaDB [hotel]> CREATE TABLE Avaliacao (
->     avaliacao_id INT PRIMARY KEY AUTO_INCREMENT,
->     data DATE NOT NULL,
->     nota INT CHECK (nota >= 1 AND nota <= 5),
->     comentario TEXT,
->     cliente_id INT,
->     FOREIGN KEY (cliente_id) REFERENCES Cliente(cliente_id)
-> );
Query OK, 0 rows affected (0.007 sec)

```

## 8. Dicionário de dados

```
XAMPP for Windows - mysql  X + v

MariaDB [hotel]> describe hotel;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| hotel_id | int(11)   | NO   | PRI | NULL    | auto_increment |
| nome     | varchar(255) | NO   |     | NULL    |              |
| endereco | varchar(255) | NO   |     | NULL    |              |
| telefone | varchar(15)  | NO   |     | NULL    |              |
| email    | varchar(255) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.003 sec)

MariaDB [hotel]> describe cliente;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| cliente_id | int(11)   | NO   | PRI | NULL    | auto_increment |
| nome       | varchar(100) | NO   |     | NULL    |              |
| sobrenome  | varchar(100) | NO   |     | NULL    |              |
| documento_identificacao | varchar(50) | NO   |     | NULL    |              |
| email      | varchar(255) | NO   |     | NULL    |              |
| telefone   | varchar(15)  | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.004 sec)

MariaDB [hotel]> describe reserva;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| reserva_id | int(11)   | NO   | PRI | NULL    | auto_increment |
| data_check_in | date      | NO   |     | NULL    |              |
| data_check_out | date      | NO   |     | NULL    |              |
| numero_pessoas | int(11)   | NO   |     | NULL    |              |
| tipo_quarto | varchar(50) | NO   |     | NULL    |              |
| cliente_id | int(11)   | YES  | MUL | NULL    |              |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.003 sec)
```

```
XAMPP for Windows - mysql  X + v

Prompt de comando: XAMPP for Windows - mysql -
MariaDB [hotel]> describe quarto;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| quarto_id | int(11)   | NO   | PRI | NULL    | auto_increment |
| numero_quarto | int(11)   | NO   | UNI | NULL    |              |
| tipo      | varchar(50) | NO   |     | NULL    |              |
| preco     | decimal(10,2) | NO   |     | NULL    |              |
| status    | enum('Ocupado', 'Dispon?vel') | NO   |     | NULL    |              |
| hotel_id | int(11)   | YES  | MUL | NULL    |              |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.003 sec)
```

MariaDB [hotel]> describe funcionario;

Field	Type	Null	Key	Default	Extra
funcionario_id	int(11)	NO	PRI	NULL	auto_increment
nome	varchar(100)	NO		NULL	
sobrenome	varchar(100)	NO		NULL	
cargo	varchar(50)	NO		NULL	
setor	varchar(50)	NO		NULL	
salario	decimal(10,2)	NO		NULL	
hotel_id	int(11)	YES	MUL	NULL	

7 rows in set (0.003 sec)

MariaDB [hotel]> describe restaurante;

Field	Type	Null	Key	Default	Extra
restaurante_id	int(11)	NO	PRI	NULL	auto_increment
nome	varchar(100)	NO		NULL	
capacidade	int(11)	NO		NULL	
tipo_cozinha	varchar(100)	NO		NULL	
horario_funcionamento	varchar(50)	NO		NULL	
hotel_id	int(11)	YES	UNI	NULL	

6 rows in set (0.004 sec)

MariaDB [hotel]> describe reserva\_pacote;

Field	Type	Null	Key	Default	Extra
reserva_id	int(11)	NO	PRI	NULL	
pacote_id	int(11)	NO	PRI	NULL	

2 rows in set (0.003 sec)

MariaDB [hotel]> describe servico\_adicional;

Field	Type	Null	Key	Default	Extra
servico_id	int(11)	NO	PRI	NULL	auto_increment
nome	varchar(100)	NO		NULL	
preco	decimal(10,2)	NO		NULL	

3 rows in set (0.003 sec)

MariaDB [hotel]> describe cliente\_servico;

Field	Type	Null	Key	Default	Extra
cliente_id	int(11)	NO	PRI	NULL	
servico_id	int(11)	NO	PRI	NULL	

2 rows in set (0.003 sec)

MariaDB [hotel]> describe avaliacao;

Field	Type	Null	Key	Default	Extra
avaliacao_id	int(11)	NO	PRI	NULL	auto_increment
data	date	NO		NULL	
nota	int(11)	YES		NULL	
comentario	text	YES		NULL	
cliente_id	int(11)	YES	MUL	NULL	

5 rows in set (0.004 sec)

## 9. Inserção de dados

```
XAMPP for Windows - mysql × + v
MariaDB [hotel]> INSERT INTO Quarto (numero_quarto, tipo, preco, status, hotel_id) VALUES
-> (101, 'Duplo', 1500.00, 'Disponível', 1),
-> (102, 'Individual', 1000.00, 'Disponível', 1),
-> (103, 'Suíte', 2500.00, 'Ocupado', 2),
-> (104, 'Duplo', 1600.00, 'Disponível', 2),
-> (201, 'Individual', 1200.00, 'Disponível', 3),
-> (202, 'Duplo', 1800.00, 'Disponível', 3),
-> (203, 'Suíte', 3000.00, 'Ocupado', 4),
-> (204, 'Lua de Mel', 3500.00, 'Disponível', 5),
-> (205, 'Família', 4000.00, 'Disponível', 4),
-> (301, 'Individual', 1100.00, 'Ocupado', 1),
-> (302, 'Duplo', 1700.00, 'Disponível', 2),
-> (303, 'Lua de Mel', 3200.00, 'Disponível', 5);
Query OK, 12 rows affected, 3 warnings (0.007 sec)
Records: 12 Duplicates: 0 Warnings: 3
```

```
MariaDB [hotel]> INSERT INTO Restaurante (nome, capacidade, tipo_cozinha, horario_funcionamento, hotel_id) VALUES
-> ('Restaurante Delícias', 50, 'Cozinha Internacional', '07:00 - 22:00', 1),
-> ('Restaurante Beira Gourmet', 80, 'Cozinha Moçambicana', '06:00 - 23:00', 2),
-> ('Restaurante Mar Azul', 100, 'Cozinha Internacional', '07:00 - 22:00', 3),
-> ('Restaurante Gourmet Tropical', 120, 'Cozinha Africana', '08:00 - 23:00', 4),
-> ('Restaurante Oásis', 80, 'Cozinha Portuguesa', '06:00 - 22:00', 5);
Query OK, 5 rows affected, 3 warnings (0.002 sec)
Records: 5 Duplicates: 0 Warnings: 3
```

```
XAMPP for Windows - mysql × + v
MariaDB [hotel]> INSERT INTO Hotel (nome, endereco, telefone, email) VALUES
-> ('Hotel Moçambique', 'Av. 24 de Julho, 123, Maputo', '+258 21 1234567', 'contato@hotelmoz.co.mz'),
-> ('Hotel Beira Mar', 'Rua dos Pescadores, 456, Beira', '+258 23 7654321', 'reservas@beiramar.co.mz'),
-> ('Hotel Polana', 'Av. Julius Nyerere, 1234, Maputo', '+258 21 9876543', 'reservas@polanahotel.co.mz'),
-> ('Hotel Pemba Paradise', 'Av. Marginal, 789, Pemba', '+258 27 8765432', 'contact@pembaparadise.co.mz'),
-> ('Hotel Tete Golden', 'Rua dos Mineradores, 555, Tete', '+258 25 6543219', 'golden@tetehotel.co.mz');
Query OK, 5 rows affected, 1 warning (0.007 sec)
Records: 5 Duplicates: 0 Warnings: 1

MariaDB [hotel]>
MariaDB [hotel]> INSERT INTO Cliente (nome, sobrenome, documento_identificacao, email, telefone) VALUES
-> ('Ana', 'Ferreira', 'BI123456', 'ana.ferreira@gmail.com', '+258 84 9876543'),
-> ('Carlos', 'Mendes', 'BI789012', 'carlos.mendes@hotmail.com', '+258 82 6543210'),
-> ('Marta', 'Sousa', 'BI345678', 'marta.sousa@yahoo.com', '+258 82 1234567'),
-> ('João', 'Lourenço', 'BI012345', 'joao.lourenco@gmail.com', '+258 84 1122334'),
-> ('Sofia', 'Nunes', 'BI678901', 'sofia.nunes@hotmail.com', '+258 82 4433221'),
-> ('Miguel', 'Pereira', 'BI112233', 'miguel.pereira@yahoo.com', '+258 84 9988776'),
-> ('Luísa', 'Campos', 'BI445566', 'luisa.campos@gmail.com', '+258 82 7766554'),
-> ('Fernando', 'Dias', 'BI778899', 'fernando.dias@outlook.com', '+258 84 6655443');
Query OK, 8 rows affected, 3 warnings (0.001 sec)
Records: 8 Duplicates: 0 Warnings: 3

MariaDB [hotel]>
MariaDB [hotel]>
MariaDB [hotel]> INSERT INTO Reserva (data_check_in, data_check_out, numero_pessoas, tipo_quarto, cliente_id) VALUES
-> ('2024-10-20', '2024-10-25', 2, 'Duplo', 1),
-> ('2024-11-05', '2024-11-10', 1, 'Individual', 2),
-> ('2024-12-15', '2024-12-20', 3, 'Suite', 3),
-> ('2024-10-10', '2024-10-12', 1, 'Individual', 4),
-> ('2024-11-01', '2024-11-07', 2, 'Duplo', 5),
-> ('2024-12-20', '2024-12-25', 4, 'Suite', 6),
-> ('2024-12-30', '2025-01-02', 2, 'Lua de Mel', 7),
-> ('2024-10-15', '2024-10-20', 3, 'Família', 8);;
Query OK, 8 rows affected, 3 warnings (0.002 sec)
Records: 8 Duplicates: 0 Warnings: 3
```

```

XAMPP for Windows - mysql
MariaDB [hotel]> INSERT INTO Pacote (nome, descricao) VALUES
-> ('Lua de Mel', 'Pacote especial para casais em lua de mel, inclui jantar romântico. '),
-> ('Família', 'Pacote família com descontos especiais e atividades para crianças. '),
-> ('Fim de Semana', 'Pacote para estadia de fim de semana com café da manhã incluído. '),
-> ('Romântico', 'Pacote para casais com jantar e decoração especial no quarto. '),
-> ('Negócios', 'Pacote voltado para clientes corporativos com serviços de reuniões e internet de alta velocidade. '),
-> ('Aventura', 'Pacote de turismo e aventura com excursões e atividades ao ar livre. ');
Query OK, 6 rows affected, 9 warnings (0.003 sec)
Records: 6 Duplicates: 0 Warnings: 9

MariaDB [hotel]>
MariaDB [hotel]>
MariaDB [hotel]> INSERT INTO Reserva_Pacote (reserva_id, pacote_id) VALUES
-> (1, 1),
-> (2, 3),
-> (3, 2),
-> (4, 1),
-> (5, 3);
Query OK, 5 rows affected (0.001 sec)
Records: 5 Duplicates: 0 Warnings: 0

MariaDB [hotel]>
MariaDB [hotel]>
MariaDB [hotel]> INSERT INTO Servico_Adicional (nome, preco) VALUES
-> ('Spa', 500.00),
-> ('Academia', 300.00),
-> ('Transporte Aeroporto', 700.00),
-> ('Piscina', 600.00),
-> ('Sauna', 700.00),
-> ('Massagem', 800.00),
-> ('Serviço de Quarto', 200.00),
-> ('Estacionamento', 100.00);
Query OK, 8 rows affected, 1 warning (0.001 sec)
Records: 8 Duplicates: 0 Warnings: 1

MariaDB [hotel]>
MariaDB [hotel]>
MariaDB [hotel]> INSERT INTO Cliente_Servico (cliente_id, servico_id) VALUES
-> (1, 1),
-> (2, 2),
-> (3, 3),
-> (2, 4),
-> (3, 5),
-> (4, 1),

```

## 10.Consultas

### 1. Listar todos os hotéis e suas informações

```

MariaDB [hotel]> SELECT * FROM Hotel;
+-----+-----+-----+-----+-----+
| hotel_id | nome           | endereco                                     | telefone | email                                     |
+-----+-----+-----+-----+-----+
| 1       | Hotel Mo?ambique | Av. 24 de Julho, 123, Maputo               | +258 21 1234567 | contato@hotelmoz.co.mz |
| 2       | Hotel Beira Mar  | Rua dos Pescadores, 456, Beira             | +258 23 7654321 | reservas@beiramar.co.mz |
| 3       | Hotel Polana     | Av. Julius Nyerere, 1234, Maputo           | +258 21 9876543 | reservas@polanahotel.co.mz |
| 4       | Hotel Pemba Paradise | Av. Marginal, 789, Pemba                 | +258 27 8765432 | contact@pembaparadise.co.mz |
| 5       | Hotel Tete Golden | Rua dos Mineradores, 555, Tete            | +258 25 6543219 | golden@tetehotel.co.mz |
+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

```

2. Listar todos os quartos disponíveis do hotel 1

```
MariaDB [hotel]> SELECT numero_quarto, tipo, preco
-> FROM Quarto
-> WHERE status = 'Disponível' AND hotel_id = 1;
+-----+-----+-----+
| numero_quarto | tipo      | preco  |
+-----+-----+-----+
| 101           | Duplo     | 1500.00 |
| 102           | Individual | 1000.00 |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

3. Consultar todas as reservas feitas pelo cliente 2

```
MariaDB [hotel]> SELECT r.reserva_id, r.data_check_in, r.data_check_out, r.numero_pessoas, r.tipo_quarto
-> FROM Reserva r
-> INNER JOIN Cliente c ON r.cliente_id = c.cliente_id
-> WHERE c.cliente_id = 2;
+-----+-----+-----+-----+-----+
| reserva_id | data_check_in | data_check_out | numero_pessoas | tipo_quarto |
+-----+-----+-----+-----+-----+
| 2          | 2024-11-05    | 2024-11-10    | 1              | Individual  |
+-----+-----+-----+-----+-----+
1 row in set (0.003 sec)
```

4. Listar todos os funcionários de um hotel 3

```
MariaDB [hotel]> SELECT f.nome, f.sobrenome, f.cargo, f.salario
-> FROM Funcionario f
-> WHERE f.hotel_id = 3;
+-----+-----+-----+-----+
| nome      | sobrenome | cargo      | salario  |
+-----+-----+-----+-----+
| Carlos    | Costa     | Recepcionista | 18000.00 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

5. Listar todos os serviços adicionais disponíveis

```
MariaDB [hotel]> SELECT * FROM Servico_Adicional;
+-----+-----+-----+
| servico_id | nome                | preco  |
+-----+-----+-----+
| 1          | Spa                 | 500.00 |
| 2          | Academia            | 300.00 |
| 3          | Transporte Aeroporto | 700.00 |
| 4          | Piscina             | 600.00 |
| 5          | Sauna               | 700.00 |
| 6          | Massagem            | 800.00 |
| 7          | Serviço de Quarto   | 200.00 |
| 8          | Estacionamento       | 100.00 |
+-----+-----+-----+
8 rows in set (0.000 sec)
```

6. Listar todas as avaliações e comentários feitos por clientes

```
MariaDB [hotel]> SELECT c.nome, c.sobrenome, a.nota, a.comentario
-> FROM Avaliacao a
-> INNER JOIN Cliente c ON a.cliente_id = c.cliente_id;
```

nome	sobrenome	nota	comentario
Ana	Ferreira	5	Excelente servi?o e atendimento!
Carlos	Mendes	4	Boa experi?ncia, mas a internet poderia ser melhor.
Marta	Sousa	5	Hotel maravilhoso, voltarei com certeza!
Jo?o	Louren?o	4	Quarto confort?vel, mas o caf? da manh? poderia ser melhor.
Sofia	Nunes	5	?timo atendimento e localiza??o!
Miguel	Pereira	3	Bom hotel, mas os pre?os s?o altos.
Lu?sa	Campos	5	Perfeito para uma lua de mel, servi?o excelente!
Fernando	Dias	4	Aventura incr?vel, adorei as atividades ao ar livre.

```
8 rows in set (0.001 sec)
```

## 11.Álgebra Relacional

1. Listar todos os hotéis e suas informações

$$\pi_{hotel\_id, nome, endereco, telefone, email} (Hotel)$$

2. Listar todos os quartos disponíveis do hotel 1

$$\pi_{numero\_quarto, tipo, preco} (\sigma_{(status = 'Disponível' \wedge hotel\_id = 1)} (Quarto))$$

3. Consultar todas as reservas feitas pelo cliente 2

$$T_1 = \sigma_{Cliente.cliente\_id = 2} (Reserva \bowtie Cliente)$$

$$Result = \pi_{reserva\_id, data\_check\_in, data\_check\_out, numero\_pessoas, tipo\_quarto} (T_1)$$

4. Listar todos os funcionários de um hotel 3

$$\pi_{nome, sobrenome, cargo, salario} (\sigma_{hotel\_id = 1} (Funcionario))$$



5. Listar todos os serviços adicionais disponíveis

$\pi_{servico\_id, nome, descricao, preco}(Servico\_Adicional)$

6. Listar todas as avaliações e comentários feitos por clientes

$\pi_{Cliente.nome, Cliente.sobrenome, Avaliacao.nota, Avaliacao.comentario}(Avaliacao \bowtie Cliente)$

## 12.Implementação de Triggers

1. Trigger para actualizar o Status do Quarto Após a Reserva

```
MariaDB [hotel]> ALTER TABLE Reserva ADD CONSTRAINT fk_quarto FOREIGN KEY (quarto_id) REFERENCES Quarto(quarto_id);
Query OK, 8 rows affected (0.033 sec)
Records: 8 Duplicates: 0 Warnings: 0

MariaDB [hotel]> DELIMITER //
MariaDB [hotel]>
MariaDB [hotel]> CREATE TRIGGER atualizar_status_quarto
-> AFTER INSERT ON Reserva
-> FOR EACH ROW
-> BEGIN
->     UPDATE Quarto
->     SET status = 'Ocupado'
->     WHERE quarto_id = NEW.quarto_id; -- Assumindo que agora há a coluna quarto_id na tabela Reserva
-> END //
Query OK, 0 rows affected (0.032 sec)
```

2. Trigger para actualizar o Status do Quarto ao Cancelar uma Reserva

```
MariaDB [hotel]> CREATE TRIGGER atualizar_status_quarto_cancelamento
-> AFTER DELETE ON Reserva
-> FOR EACH ROW
-> BEGIN
->     UPDATE Quarto
->     SET status = 'Disponível'
->     WHERE quarto_id = OLD.quarto_id;
-> END //
Query OK, 0 rows affected (0.033 sec)

MariaDB [hotel]>
MariaDB [hotel]> DELIMITER ;|
```

### 3. Trigger para actualizar o Saldo do Cliente ao Inserir um Serviço Adicional

```
MariaDB [hotel]> CREATE TRIGGER atualizar_saldo_cliente
-> AFTER INSERT ON Cliente_Servico
-> FOR EACH ROW
-> BEGIN
->   UPDATE Cliente
->   SET saldo = saldo + (SELECT preco FROM Servico_Adicional WHERE servico_id = NEW.servico_id)
->   WHERE cliente_id = NEW.cliente_id;
-> END //
Query OK, 0 rows affected (0.008 sec)
```

## 13.Implementação de procedimentos/funções

### 1. Procedimento para Inserir uma Nova Reserva

```
MariaDB [hotel]> CREATE PROCEDURE inserir_reserva(
->   IN p_data_check_in DATE,
->   IN p_data_check_out DATE,
->   IN p_numero_pessoas INT,
->   IN p_tipo_quarto VARCHAR(50),
->   IN p_cliente_id INT,
->   IN p_quarto_id INT
-> )
-> BEGIN
->   INSERT INTO Reserva (data_check_in, data_check_out, numero_pessoas, tipo_quarto, cliente_id, quarto_id)
->   VALUES (p_data_check_in, p_data_check_out, p_numero_pessoas, p_tipo_quarto, p_cliente_id, p_quarto_id);
->   UPDATE Quarto
->   SET status = 'Ocupado'
->   WHERE quarto_id = p_quarto_id;
-> END //
Query OK, 0 rows affected (0.008 sec)
```

### 2. Função para Calcular a Duração da Estadia

```
MariaDB [hotel]>
MariaDB [hotel]> DELIMITER ;
MariaDB [hotel]> DELIMITER //
MariaDB [hotel]>
MariaDB [hotel]> CREATE FUNCTION calcular_duracao_estadia(
->   p_data_check_in DATE,
->   p_data_check_out DATE
-> ) RETURNS INT
-> BEGIN
->   RETURN DATEDIFF(p_data_check_out, p_data_check_in);
-> END //
Query OK, 0 rows affected (0.005 sec)
```

### 3. Procedimento para actualizar o Saldo do Cliente

```
MariaDB [hotel]>
MariaDB [hotel]> DELIMITER ;
MariaDB [hotel]> DELIMITER //
MariaDB [hotel]>
MariaDB [hotel]> CREATE PROCEDURE atualizar_saldo_cliente(
->     IN p_cliente_id INT,
->     IN p_valor DECIMAL(10, 2)
-> )
-> BEGIN
->     UPDATE Cliente
->     SET saldo = saldo + p_valor
->     WHERE cliente_id = p_cliente_id;
-> END //
Query OK, 0 rows affected (0.005 sec)
```

### 4. Função para Obter o Saldo Total de um Cliente

```
MariaDB [hotel]>
MariaDB [hotel]> DELIMITER ;
MariaDB [hotel]> DELIMITER //
MariaDB [hotel]>
MariaDB [hotel]> CREATE FUNCTION obter_saldo_total_cliente(
->     p_cliente_id INT
-> ) RETURNS DECIMAL(10, 2)
-> BEGIN
->     DECLARE saldo_total DECIMAL(10, 2);
->
->     SELECT SUM(preco) INTO saldo_total
->     FROM Servico_Adicional sa
->     JOIN Cliente_Servico cs ON sa.servico_id = cs.servico_id
->     WHERE cs.cliente_id = p_cliente_id;
->
->     RETURN COALESCE(saldo_total, 0); -- Retorna 0 se não houver saldo
-> END //
Query OK, 0 rows affected (0.008 sec)
```

## Base de Dados II

### 14. Transações relevantes que podem ocorrer na Base de Dados do Hotel

- I. Inserir uma nova reserva para um cliente em um hotel específico, associando-o a um ou mais quartos disponíveis
- II. Registrar o check-in de um cliente, actualizando a reserva como activa e confirmando a ocupação do quarto.
- III. Registrar o check-out do cliente, liberando o quarto e encerrando a reserva.
- IV. Inserir um novo cliente no sistema, registando seus dados pessoais.
- V. Inserir uma nova avaliação após o término da estadia, contendo a nota e os comentários do cliente.
- VI. Alterar o preço de um determinado tipo de quarto em um hotel específico.
- VII. Registrar a solicitação de um ou mais serviços adicionais por parte de um cliente durante sua estadia.
- VIII. Inserir um novo pacote de hospedagem (como "Lua de Mel" ou "Pacote Família"), especificando sua descrição e nome.
- IX. Associar uma reserva existente a um pacote promocional oferecido pelo hotel.

### 15. Transações escritas em linguagem “Read” e “Write”

- i. Transação 1 (T1): Cliente A faz uma reserva para um tipo quarto.

Transação 2 (T2): Cliente B tenta reservar o mesmo tipo de quarto de A e no mesmo hotel.

Código do cliente (id): X

T1	T2
Read(X)	
X = 1	
	Read(X)
W(X)	
	X =1
	Write(X)

### Possíveis conflitos

$$R_1(X) < W_2(X)$$

$$W_1(X) < W_2(X)$$

### Resolução de conflitos (2PL)

T1	T2
Lock_x(X)	<i>bloqueado</i>
Read(X)	<i>bloqueado</i>
X = 1	<i>bloqueado</i>
W(X)	<i>bloqueado</i>
Unlock(X)	<i>bloqueado</i>
	Lock_X(X)
	Read(X)
	X =1
	Write(X)
	Unlock(X)

ii. Transação 1 (T1): Cliente A faz check-out

Transação 2 (T2): Cliente B tenta fazer check-in no mesmo quarto imediatamente após o check-out

Status do quarto: X (ocupado/livre)

T1	T2
Read(X)	
X = “livre”	
	Read(X)
	X = “Ocupado”
Write (X)	
	Write(X)

#### Possíveis conflitos

$$R_1(X) < W_2(X)$$

$$W_1(X) < W_2(X)$$

#### Resolução de conflitos

T1	T2
Lock_x(X)	
Read(X)	<i>bloqueado</i>
X = “livre”	<i>bloqueado</i>
Write (X)	<i>bloqueado</i>
Unlock(X)	<i>bloqueado</i>
	Lock_x(X)
	Read(X)
	X = “Ocupado”
	Write(X)
	Unlock(X)

iii. T1: Cliente A deixa uma avaliação

T2: Cliente A solicita um serviço adicional durante o mesmo período

T3: O gerente decide alterar a nota da avaliação do Cliente A antes de ser confirmada

Nota: X

Comentário: Y

Serviço adicional: Z

T1	T2	T3
Read(X)	Read(Z)	
X = 4	Z = “Massagem”	
	Write(Z)	Read(X)
		X = 6
Write(X)		
Read(Y)		Write(X)
Y = “Bom serviço”		
Write (Y)		

Conflitos

$R_1(X) < W_3(X)$

$W_1(X) < W_3(X)$

### Resolução de conflitos

T1	T2	T3
Lock_x(X)	Read(Z)	
X = 4	Z = “Massagem”	
Write(X)	Write(Z)	
Lock_x(Y)		
Read(Y)		
Y = “Bom serviço”		
Write (Y)		
Unlock(X)		

Unlock(Y)		Lock_x(X)
		Read(X)
		X = 6
		Write(X)
		Unlock(X)

## 16. Análise do Volume de Transações

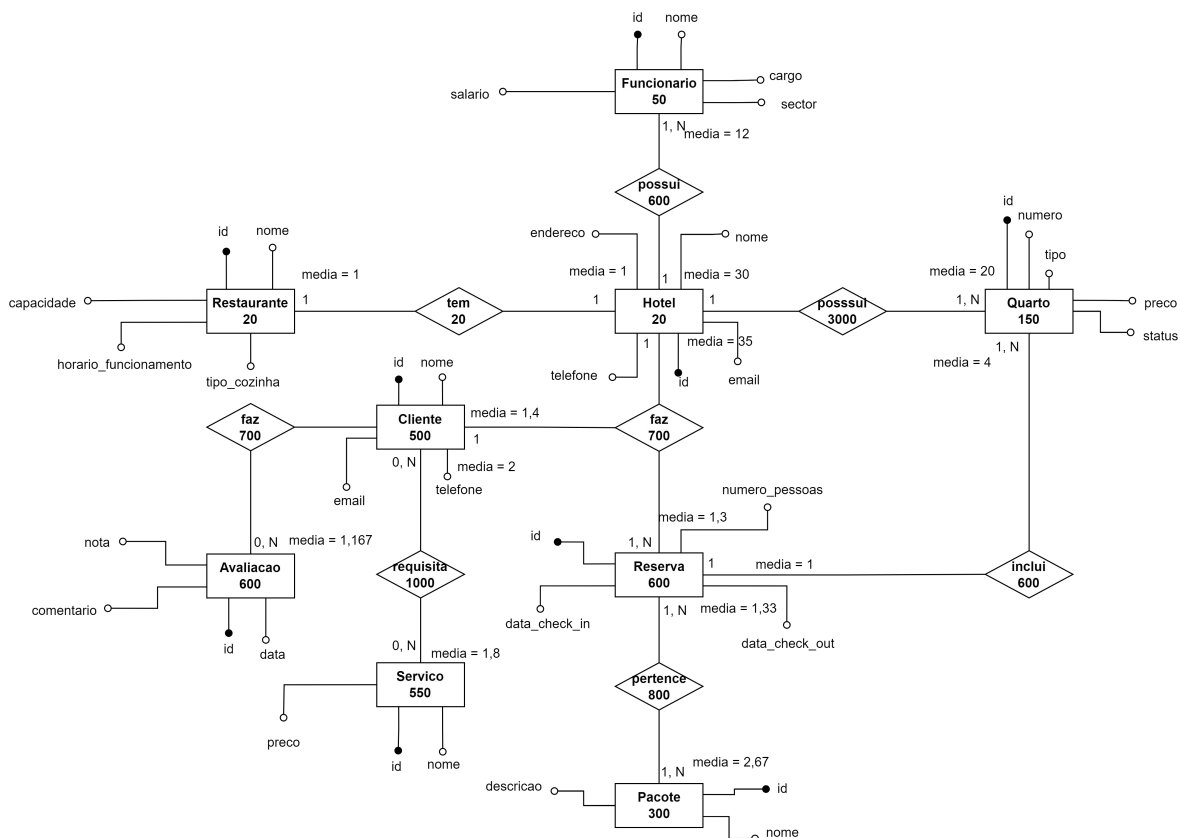
A análise do volume de transações é um processo crítico na gestão de bases de dados que avalia a quantidade e a frequência das operações realizadas sobre o banco de dados. Essa análise ajuda a identificar padrões de uso, otimizar o desempenho do sistema e garantir que a infraestrutura seja capaz de atender à carga esperada, seja em termos de leitura, escrita, ou ambas.

### 16.1. Importância da Análise do Volume de Transações

- a) **Planeamento de Capacidade:** Ajuda a estimar os recursos necessários, como armazenamento, CPU e memória, com base no número de transações.
- b) Evita gargalos de desempenho e falhas no sistema.
- c) **Otimização do Desempenho:** Permite identificar consultas e operações frequentes, possibilitando a implementação de estratégias como índices ou desnormalização para otimizar o acesso aos dados.
- d) **Identificação de Gargalos:** Avalia transações que consomem muitos recursos e identificam possíveis pontos de falha.
- e) **Modelagem de Carga:** Estima cenários futuros de carga com base nos padrões actuais de transações, importante em sistemas que esperam crescimento de usuários. Elementos da Análise de Volume de Transações



- f) **Frequência de Operações:** Mede quantas vezes as operações de leitura, escrita, actualização e exclusão são executadas.
- g) **Tipos de Transações:** Analisa transações simples (leitura ou escrita única) e complexas (que envolvem múltiplas tabelas e junções).
- h) **Tamanho dos Dados Manipulados:** Avalia o volume médio de dados lidos, inseridos ou alterados em cada transação.
- i) **Padrões de Acesso:** Identifica quais tabelas ou colunas são mais acessadas, além dos horários de maior volume de transações (picos de uso).
- j) **Tempo de Execução:** Mede o tempo necessário para completar diferentes tipos de transações e determina operações críticas.



Com base na tabela acima, podemos concluir o seguinte:

- O sistema cadastrou 20 hotéis e cada hotel tem em média 12 funcionários e 1 restaurante
- Um pacote em média pertence a 3 reservas
- O Sistema registou 600 reservas e cada reserva possui 4 quartos
- Em média cada cliente requisita 2 serviços

Aqui podemos listar algumas operações que podem ocorrer na base no sistema:

Op1: Listar as reservas por quanto, 100 vezes por dia

Op2: Reservar quarto para um cliente: 20 vezes por dia

Op3: Requisitar um serviço, 50 vezes por dia

Op4: Deixar uma avaliação, 60 vezes por dia.

Tabela de frequência das operações

Operação	Frequência	Conceito	E/R	W/R	Media de ocorrência
Op1	100	Reserva Possui Quarto	E R E	W W W	100 $100 * 1 = 100$ $100 * 4 = 400$
Op2	20	Cliente Faz Reserva	E R E	R W W	$20 * 1,4 = 28$ $20 * 1,3 = 26$ 20
Op3	50	Cliente Requisita Serviço	E R E	R W W	$50 * 2 = 100$ $50 * 1,8 = 90$ 50
Op4	60	Cliente Faz Avaliação	E R E	R W W	$60 * 1,4 = 84$ $60 * 1,167 = 70$ 60

## 17.Desnormalização

A **desnormalização** é um processo no design de bancos de dados que envolve a introdução de redundâncias controladas nas tabelas para melhorar o desempenho de leitura e reduzir o tempo de resposta das consultas, especialmente em sistemas de alto volume de transações e consultas

complexas. Ao contrário da **normalização**, que elimina redundâncias para manter a consistência e integridade dos dados, a desnormalização prioriza a eficiência, mesmo que isso aumente o risco de inconsistências.

### Quando Usar a Desnormalização?

A desnormalização é indicada principalmente em sistemas onde:

- O tempo de leitura das consultas precisa ser otimizado, como em sistemas OLAP (Online Analytical Processing) ou em relatórios frequentes.
- A complexidade das junções entre tabelas (joins) afeta negativamente o desempenho.
- As operações de escrita (inserção, actualização e exclusão) são menos frequentes do que as operações de leitura.

#### 17.1. Técnicas de Desnormalização

- a) **Duplicação de Dados:** Repetição de dados entre tabelas para evitar consultas que exijam junções complexas.  
Exemplo: Adicionar o nome de um cliente diretamente na tabela de pedidos, em vez de buscar na tabela de clientes.
- b) **Adição de Colunas Derivadas:** Criação de colunas que armazenam dados calculados ou frequentemente consultados, como o total de um pedido.
- c) **Fusão de Tabelas (Table Denormalization):** Combinação de tabelas relacionadas em uma única tabela para evitar junções.
- d) **Introdução de Tabelas de Resumo (Summary Tables):** Tabelas que agregam dados frequentemente consultados, como somas, contagens ou médias.

#### 17.2. Vantagens da Desnormalização

- **Melhoria no desempenho de leitura:** Reduz o tempo das consultas ao diminuir o número de junções necessárias.
- **Simplificação das consultas:** Consultas SQL mais simples e diretas.

- **Redução de cargas pesadas de processamento:** Ideal para sistemas que priorizam leitura em vez de escrita.

### 17.3. Desvantagens da Desnormalização

- **Aumento de espaço em disco:** Dados redundantes consomem mais espaço.
- **Risco de inconsistência:** Dados duplicados podem divergir em caso de actualizações incorretas.
- **Manutenção mais complexa:** Processos de inserção, actualização e exclusão tornam-se mais propensos a erros e exigem mais esforço.

### 17.4. Propostas de desnormalização

#### i. Incorporar informações de clientes diretamente na tabela Reserva

**Problema:** actualmente, para buscar detalhes de uma reserva e os dados do cliente, é necessário realizar um join entre as tabelas Reserva e Cliente.

**Proposta:** Adicionar as colunas nome, sobrenome e documento\_identificacao do cliente directamente na tabela Reserva.

**Vantagem:** Reduz a necessidade de joins frequentes para consultas de reservas com detalhes de clientes.

**Desvantagem:** A actualização de dados de clientes em Cliente precisa ser replicada em Reserva.

```
13 • ALTER TABLE Reserva
14     ADD COLUMN nome_cliente VARCHAR(100),
15     ADD COLUMN sobrenome_cliente VARCHAR(100),
16     ADD COLUMN documento_identificacao VARCHAR(50);
--
```

#### ii. Incorporar informações do quarto diretamente na tabela Reserva

**Problema:** Para consultar detalhes de uma reserva e o tipo/preço do quarto, é necessário um join com a tabela Quarto.

**Proposta:** Adicionar as colunas tipo e preco do quarto diretamente na tabela Reserva.

**Vantagem:** Reduz a quantidade de joins necessários para exibir os detalhes completos de uma reserva.

**Desvantagem:** Se as características de um quarto mudarem (ex.: preço), é necessário actualizar múltiplas linhas.

```
17
18 • ALTER TABLE Reserva
19   ADD COLUMN tipo_quarto_reservado VARCHAR(50),
20   ADD COLUMN preco_quarto_reservado DECIMAL(10, 2);
21
```

### iii. Adicionar resumo de reservas na tabela Hotel

**Problema:** Para calcular o número de reservas feitas em cada hotel ou a receita total gerada, é necessário realizar cálculos a partir da tabela Reserva.

**Proposta:** Adicionar colunas total\_reservas e receita\_total na tabela Hotel, actualizadas via triggers.

**Vantagem:** Acelera consultas frequentes de estatísticas por hotel.

**Desvantagem:** A consistência dos dados depende de triggers bem implementados.

```
22 • ALTER TABLE Hotel
23   ADD COLUMN total_reservas INT DEFAULT 0,
24   ADD COLUMN receita_total DECIMAL(10, 2) DEFAULT 0.00;
25
```

### iv. Criar uma tabela agregada de históricos de clientes

**Problema:** Consultas para listar reservas passadas, serviços adicionais contratados e avaliações exigem múltiplos joins entre as tabelas Reserva, Cliente\_Servico e Avaliacao.

**Proposta:** Criar uma tabela desnormalizada que consolide essas informações.

**Vantagem:** Consultas históricas tornam-se muito mais rápidas.

**Desvantagem:** Exige actualizações regulares para manter a tabela sincronizada.

```
25
26 • ○ CREATE TABLE Historico_Cliente (
27     cliente_id INT,
28     nome_cliente VARCHAR(100),
29     total_reservas INT,
30     total_gasto DECIMAL(10, 2),
31     ultima_avaliacao INT,
32     PRIMARY KEY (cliente_id)
33 );
```

## 18. Índices

Os índices são estruturas de dados auxiliares criadas em tabelas de banco de dados para acelerar o processo de recuperação de informações. Eles funcionam como um catálogo que facilita a busca por registos, minimizando a necessidade de leitura completa das tabelas (full table scan). Um índice pode ser criado em uma ou mais colunas de uma tabela e é particularmente útil em consultas frequentes com cláusulas como WHERE, JOIN, ORDER BY e GROUP BY.

### 18.1. Tipos de Índices

- a) **Índice Clusterizado (Clustered Index):** Organiza fisicamente os dados da tabela na ordem do índice. Cada tabela pode ter apenas um índice clusterizado, pois determina a disposição dos registos. É ideal para consultas que precisam acessar intervalos de valores ordenados.
- a) **Índice Não Clusterizado (Non-clustered Index):** Armazena uma cópia ordenada das colunas do índice, enquanto os dados reais permanecem em outra estrutura. Ele inclui um ponteiro para localizar os registos na tabela original.
- b) **Índices Compostos:** Criados em mais de uma coluna, úteis para consultas que filtram ou ordenam com base em múltiplos critérios.

- c) **Índices Únicos:** Garantem que os valores de uma coluna ou combinação de colunas sejam únicos, evitando duplicidades.
- d) **Índices de Texto Completo (Full-text Indexes):** Aplicáveis para pesquisas em grandes volumes de texto, permitindo buscas avançadas, como localização de palavras ou frases em colunas de texto.

### Vantagens dos Índices

- **Melhoria de desempenho:** Reduzem o tempo necessário para recuperar dados.
- **Auxiliam na ordenação e agrupamento:** Facilitam operações que envolvem ORDER BY ou GROUP BY.
- **Geração eficiente de planos de execução:** Optimizam o trabalho do otimizador de consultas (Query Optimizer).

### Desvantagens dos Índices

- **Espaço adicional:** Consomem espaço em disco devido à manutenção da estrutura do índice.
- **Impacto na escrita:** Inserções, actualizações e exclusões podem se tornar mais lentas, pois os índices precisam ser actualizados constantemente.
- **Manutenção complexa:** Exige um balanceamento cuidadoso para evitar overheads desnecessários.

## 18.2. Propostas de índice

### i. Índice para busca de reservas por cliente

Permite localizar rapidamente todas as reservas feitas por um cliente específico.

```
2  
3 • CREATE INDEX idx_reserva_cliente ON Reserva(cliente_id);  
4
```

### ii. Índice para busca de reservas por quarto

Facilita a busca de reservas relacionadas a um quarto específico.

```
4
5 • CREATE INDEX idx_reserva_quarto ON Reserva(quarto_id);
6
```

### iii. Índice para busca de quartos por hotel

Acelera a consulta de quartos disponíveis dentro de um hotel específico.

```
6
7 • CREATE INDEX idx_quarto_hotel ON Quarto(hotel_id);
8
```

### iv. Índice para disponibilidade de quartos

Acelera a consulta de quartos disponíveis ou ocupados.

```
8
9 • CREATE INDEX idx_quarto_status ON Quarto(status);
10
```

### v. Índice para busca de hotéis por localização

Melhora o desempenho de consultas por hotéis com base em suas localizações.

```
10
11 • CREATE INDEX idx_hotel_endereco ON Hotel(endereco);
12
```



## **19. Conclusão**

O desenvolvimento do sistema de gestão hoteleira proposto neste projecto proporcionou uma solução robusta e bem estruturada para atender às necessidades de uma rede de hotéis. A modelagem da base de dados seguiu princípios de normalização para garantir integridade e evitar redundâncias, mas também considerou possibilidades de desnormalização e optimizações para lidar com o alto volume de transações e consultas frequentes no contexto real.

Através das funcionalidades implementadas, como a gestão de reservas, avaliação de clientes, controle de serviços adicionais e operação de quartos, o sistema demonstra ser capaz de suportar as operações administrativas e melhorar a experiência do cliente.

A análise de volume de transações e o mapeamento de acessos destacaram as áreas de maior demanda no sistema, permitindo proposições de índices e estratégias de particionamento para melhorar a eficiência e escalabilidade.

Este projecto, além de atender ao objectivo académico, serve como base sólida para a implementação prática de um sistema de gestão hoteleira.

## 20.Bibliografia

- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th Edition). Pearson.
- Ramakrishnan, R., & Gehrke, J. (2002). *Database Management Systems* (3rd Edition). McGraw-Hill.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7th Edition). McGraw-Hill.
- Date, C. J. (2019). *An Introduction to Database Systems* (8th Edition). Pearson.