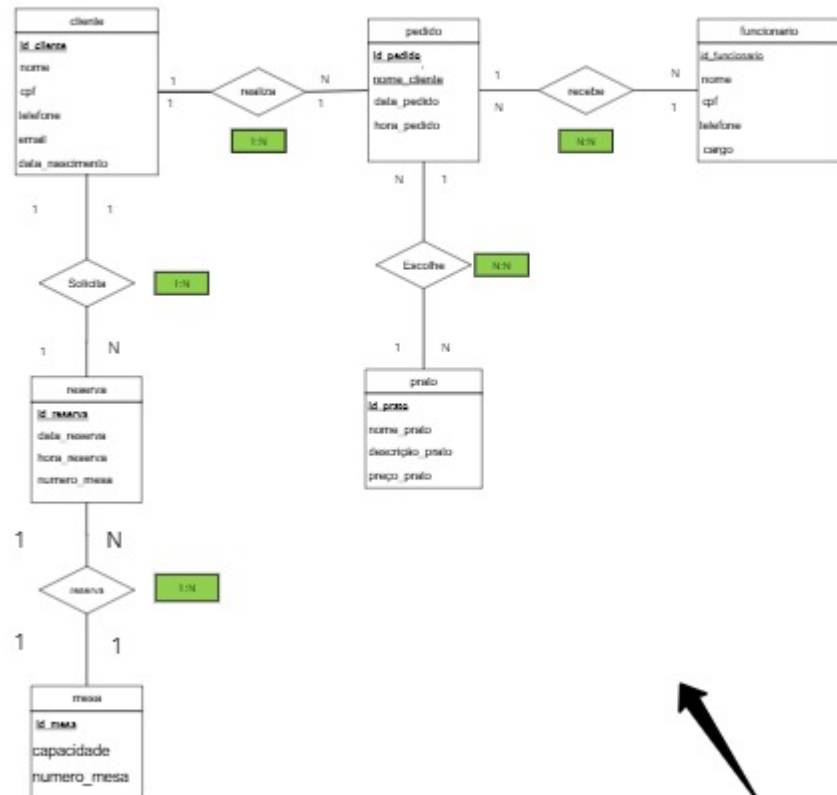
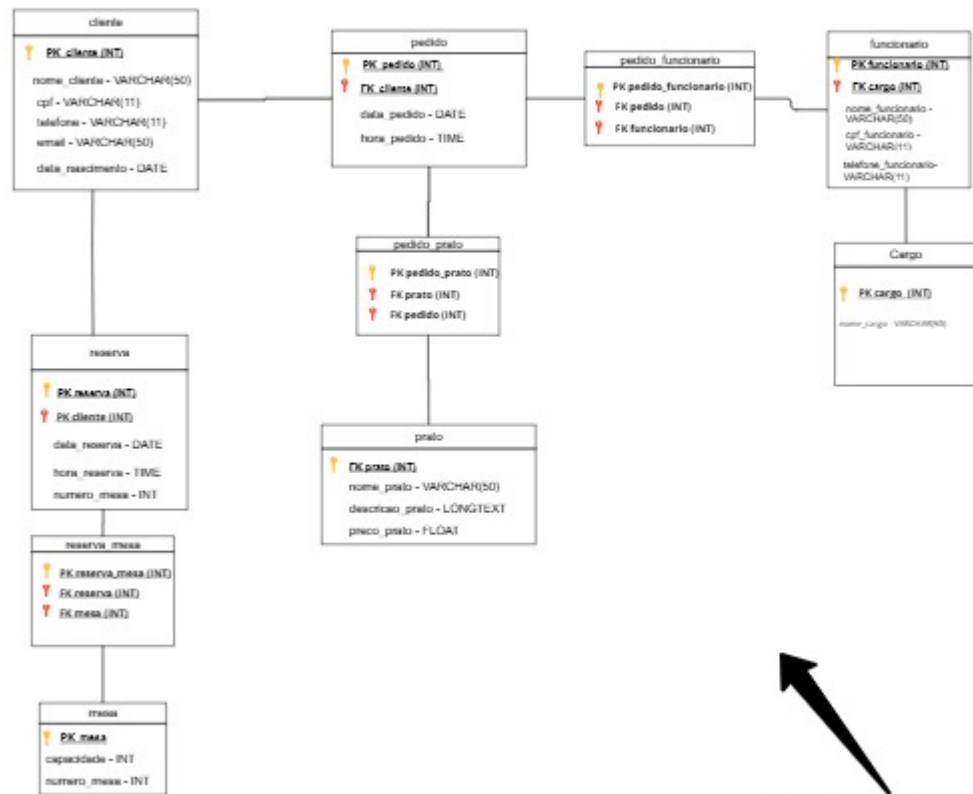



Modelo conceitual



Conferir Relacionamentos




Conferir Tabelas associativas

 Envios até **30/09**, segunda-feira, às **23h59**.


 Submissão no LMS

Todos devem submeter o trabalho, independente do trabalho ter sido em grupo!

 Formato de envio: pasta zipada contendo 3 arquivos:

1. Diagrama com o modelo relacional (parte 1)
2. Arquivo .sql com todos os comandos DDL, DML e DQL com o modelo (parte 1)
3. Arquivo .sql com as consultas que respondem a parte 2

Importante: **identificar o zip com o número do grupo**, conforme definido aqui para que eu possa identificar os integrantes (ex. grupo1.zip)

 **Parte 1**: escolher um dos projetos

1. Criar o modelo e aplicar as formas normais
2. Criar o banco de dados (DDL)
3. Inserir ao menos 5 registros em cada tabela (DML)
4. Atualizar e deletar ao menos 2 registros (DML)
5. Realizar 4 consultas (1 com agrupamento, 3 com joins) (DQL)

Parte 2: usar o banco de dados já populado para responder as perguntas por meio de consultas (DQL)

Parte 1 – Projeto 1

Sistema de Gerenciamento de Restaurante

O restaurante ada.food deseja organizar informações para melhorar sua logística e funcionamento, incluindo dados sobre seus clientes, funcionários, pratos, pedidos, mesas e reservas. Seguem os requisitos:

- Cada cliente deve ter um registro contendo nome, cpf, telefone, email e data de nascimento.
- Um cliente pode fazer reservas e pedidos.
- Cada funcionário deve ter um registro contendo nome, cpf, cargo (ex.: Garçon, Cozinheiro), e telefone.
- Funcionários podem estar envolvidos em diferentes pedidos, e cada pedido pode ser atendido por vários funcionários (ex.: garçom que atende e cozinheiro que prepara o prato).
- O restaurante oferece vários pratos, cada um com um nome, descrição, e preço.
- O sistema deve permitir o registro de pedidos, incluindo a data, hora, e o cliente que fez o pedido.
- Um pedido pode incluir vários pratos, e cada prato pode estar em vários pedidos.
- O restaurante possui várias mesas, cada uma com um número identificador e capacidade de assentos.
- Clientes podem fazer reservas de mesas, incluindo a data, hora e o número da mesa reservada.

1. Requisitos e Planejamento

- **Entendimento das necessidades:** Identificar quais dados serão armazenados, o volume esperado e como eles serão usados (consultas, relatórios, etc.).
- **Definição de objetivos:** Clarificar o propósito do banco, as funcionalidades que ele deve suportar e os requisitos de desempenho.

2. Modelagem de Dados

- **Modelo conceitual:** Criar um modelo de alto nível que descreva as entidades, atributos e relacionamentos. A modelagem entidade-relacionamento (ER) é comum nesta etapa.
- **Modelo lógico:** Converter o modelo conceitual em um formato que pode ser implementado em um sistema de banco de dados. Definir tabelas, colunas, tipos de dados e as relações entre elas.
- **Modelo físico:** Ajustar o modelo lógico para o sistema de gerenciamento de banco de dados (SGBD) específico que será utilizado, otimizando o design para desempenho.

3. Escolha do SGBD

- Decidir qual sistema de banco de dados usar, como **MySQL**, **PostgreSQL**, **SQL Server**, **MongoDB** (para bancos de dados NoSQL), dependendo dos requisitos de dados e performance.

4. Normalização e Desnormalização

- **Normalização:** Organizar os dados para minimizar a redundância e garantir a integridade. Normalizar até uma forma (normal forms) adequada ao caso de uso.
- **Desnormalização:** Às vezes, é necessário desnormalizar para otimizar o desempenho, especialmente em cenários de alta leitura.

5. Criação de Esquema e Tabelas

- Definir as estruturas de tabelas, colunas, índices e chaves (primárias, estrangeiras).
- Especificar **constraints** (restrições) como **UNIQUE**, **NOT NULL**, **CHECK** e **FOREIGN KEY** para garantir a integridade dos dados.

6. Configuração de Índices e Otimizações

- Criar índices nas colunas que serão frequentemente usadas em consultas para melhorar a performance.
- Ajustar a estrutura do banco conforme necessário para suportar melhor consulta e inserção de dados.

7. Segurança e Controle de Acesso

- Definir usuários, roles e permissões. Configurar níveis de acesso apropriados para leitura, escrita, modificação, etc.
- Implementar criptografia e backup de dados conforme necessário.

8. População de Dados

- Inserir os dados iniciais no banco. Isso pode ser feito através de **ETL** (Extração, Transformação e Carga), carregando dados de fontes externas.

9. Testes e Validação

- Verificar se o banco atende aos requisitos de desempenho e integridade de dados. Realizar testes de carga e estresse para identificar gargalos.
- Validar as consultas e relatórios para garantir que os dados retornados estão corretos.

10. Manutenção e Monitoramento