Exercício de Modelagem Conceitual e Normalização de Banco de Dados para Cadastro de Provas de Trânsito

Cenário:

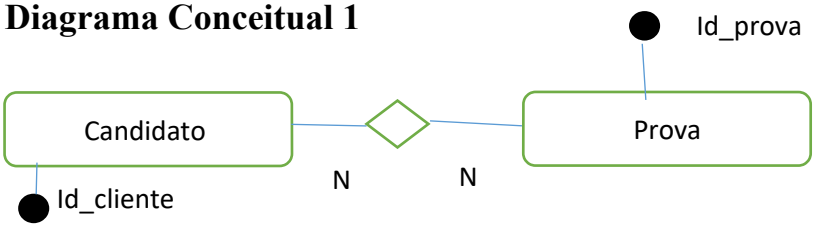
Uma empresa que administra provas de trânsito deseja organizar seus dados para facilitar o gerenciamento dos candidatos que participam das provas. O sistema deve armazenar informações sobre os candidatos, as provas realizadas e as notas obtidas. Para manter a simplicidade, o sistema será modelado com no máximo três tabelas, aplicando boas práticas de normalização.

Requisitos do sistema:

1. **Candidatos**: Cada candidato tem um identificador único (ID), nome e CPF.
2. **Provas**: Cada prova tem um identificador único (ID_Prova), uma data de realização e uma descrição do tipo de prova (por exemplo, "Prova Teórica" ou "Prova Prática").
3. **Notas**: Cada candidato pode participar de várias provas. Para cada prova realizada, o candidato recebe uma nota que é registrada junto à data da prova.
4. Banco de dados é Sqlite

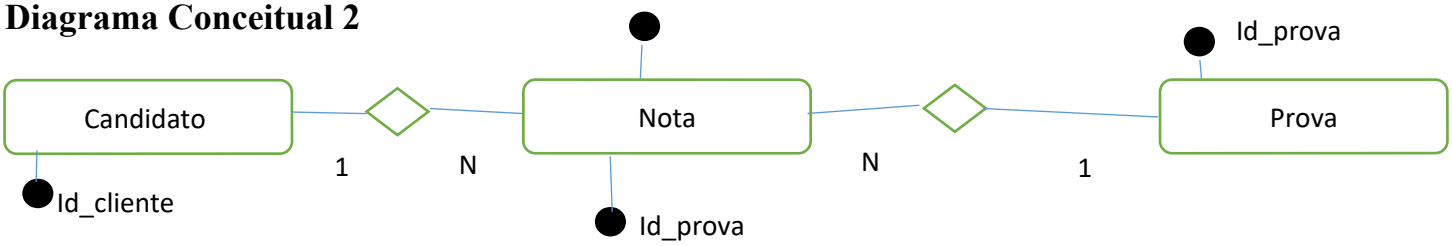
I. Monte o DER para este sistema.
II. Forneça o projeto lógico das tabelas.
III. Cuide para que as relações do projeto lógico estejam normalizadas na 1FN, 2FN e 3FN.
IV. Explique por que cada tabela do modelo lógico está normalizada.

Diagrama Conceitual 1



OU

Diagrama Conceitual 2



Neste digrama um candidato pode realizar a mesma prova uma única vez, pois id_prova não pode repetir para o mesmo candidato. A solução é o candidato realizar uma nova prova. Assim, um candidato pode realizar múltiplas provas diferentes.

Candidato (ID_Candidato, Nome, CPF)
Prova (ID_Prova, Data_Prova, Descrição)
Nota (ID_Candidato, ID_Prova, Nota) resolve Relacionamento entre Candidato e Prova id_candato

Explicação:

O sistema pode registrar múltiplas entradas na tabela **Nota**, com cada tentativa representada por um registro separado. Isso é possível porque tanto **ID_Candidato** quanto **ID_Prova** são chaves estrangeiras, o que permite múltiplas tentativas para o mesmo candidato e prova.

A tabela intermediária **Nota** resolve o relacionamento muitos-para-muitos entre candidatos e provas, garantindo que cada tentativa de prova tenha uma nota registrada de maneira única e sem redundâncias.

Projeto Lógico:

- **Tabela Candidato**
  - ID_Candidato int (PK)
  - Nome text not null
  - CPF int not null
- **Tabela Prova**
  - ID_Prova int (PK)
  - Dia_prova int
  - Mês_prova int

- Ano_prova int
- Descrição not null

🕐 **Tabela Nota**

- ID_Candidato (PK) (FK para candidato)
- ID_Prova (PK) (FK para candidato)
- Nota real not null

Explicação:

1. **Tabela Candidato**: Armazena os dados dos candidatos que participam das provas de trânsito, incluindo o **ID_Candidato**, **Nome** e **CPF**.
2. **Tabela Prova**: Armazena as informações de cada prova aplicada, incluindo o **ID_Prova**, **Data_Prova** e **Descrição** (como "Prova Teórica" ou "Prova Prática").
3. **Tabela Nota**: Relaciona cada candidato com as provas que ele realizou, armazenando a **Nota** correspondente a cada prova. Nessa tabela, os campos **ID_Candidato** e **ID_Prova** formam uma chave composta (FK), permitindo múltiplas tentativas de provas para o mesmo candidato e registrando a nota de cada tentativa.

1ª Forma Normal (1FN):

- Elimina valores repetidos ou múltiplos valores em uma única coluna. As tabelas já estão na 1ª FN, pois todos os atributos contêm valores atômicos (únicos e indivisíveis).

2ª Forma Normal (2FN):

- Uma tabela está na 2ª FN se estiver na 1ª FN e todos os atributos que não são chaves dependem inteiramente da chave primária.

3ª Forma Normal (3FN):

- Para estar na 3ª FN, a tabela deve estar na 2ª FN e não deve haver dependência transitiva (ou seja, nenhum atributo não chave deve depender de outro atributo não chave).