

(A) A informação crítica que falta nos atributos Telefone\_local e Telefone\_celular ou seja o código de área (ou DDD). O código de área é essencial para realizar chamadas para diferentes cidades ou estados. Sem ele, não é possível identificar a qual localidade o número pertence, o que impede que a ligação seja realizada corretamente.

(B) Seria mais apropriado incluir novos atributos para armazenar os códigos de área. Isso garantiria que a informação sobre cada telefone estivesse completa e que os atributos fossem autoexplicativos. Assim, teríamos algo como:

Telefone\_local

DDD\_local

Telefone\_celular

DDD\_celular

Outra opção seria incorporar o DDD diretamente no atributo

Telefone\_local, colocando-o entre parênteses, como por exemplo:

Telefone\_local (DDD).

(C) Vantagens e desvantagens de dividir o atributo Nome

Vantagens:Flexibilidade: Permite fácil manipulação de nomes em sistemas que requerem diferentes formatos (ex: apenas sobrenome para listagens).

Validação: Facilita a validação e formatação adequada, como garantir que os sobrenomes sejam sempre reconhecidos.

Pesquisa: Melhora a capacidade de pesquisa e ordenação em consultas que dependem de partes do nome.

Desvantagens:Complexidade: Aumenta a complexidade do esquema do banco de dados, necessitando de mais atributos.

Inconsistência: Pode levar a inconsistências se não for gerenciado adequadamente (ex: usuários podem não usar um nome do meio).

Desperdício de espaço: Em alguns casos, pode resultar em desperdício de espaço se muitos alunos não tiverem nomes do meio.

(D) Armazenar em um único atributo: Quando a informação é simples e frequentemente utilizada em conjunto, como um nome completo ou um endereço. Também é adequado se a informação não precisará ser manipulada separadamente.

Separar a informação: Quando as partes da informação têm utilidades distintas, como pesquisa, ordenação ou formatação. Também é adequado quando as informações podem ter variações (ex: nomes diferentes em diferentes contextos) ou se existe a necessidade de validação independente.

(E) Sugestões para armazenar entre 0 e 5 telefones

Abordagem 1: Atributos fixos

Criação de cinco atributos específicos para os telefones, como:

Telefone\_1

Telefone\_2

Telefone\_3

Telefone\_4

Telefone\_5

Com isso, é fácil armazenar até cinco números, mas poderemos ter muitos valores nulos, e assim ocupando muito espaço e memória.

Abordagem 2: Tabela separada com Chave Composta

Criação de uma tabela separada para armazenar os TELEFONES, que terá os seguintes atributos:

ID\_Aluno (chave estrangeira referenciando a ALUNO)

Telefone

DDD

Chave Composta

A tabela TELEFONES pode ter uma chave composta formada por ID\_Aluno e Telefone. Isso permite que um aluno tenha múltiplos telefones, garantindo que não haja duplicação de números para um mesmo aluno.

Estrutura da Tabela

Tabela ALUNO:

ID\_Aluno (chave primária)

Nome

CPF

Idade

Tabela TELEFONES:

ID\_Aluno (chave estrangeira referenciando ALUNO)

Telefone

DDD

Chave Primária: (ID\_Aluno, Telefone)