

Normalização de Banco de Dados

Respostas:

1

1- Normalização é o processo de organização de dados em um banco de dados. Isso inclui a criação de tabelas e o estabelecimento de relações entre essas tabelas de acordo com as regras projetadas para proteger os dados e tornar o banco de dados mais flexível, eliminando a redundância e a dependência inconsistente.

2- A normalização de banco de dados 1FN, 2FN e 3FN é um processo crucial para garantir a eficiência e a integridade dos dados em sistemas de gerenciamento de banco de dados. Essa prática ajuda a organizar as informações de forma mais lógica, reduzindo a redundância e eliminando possíveis problemas de inconsistência.

A Primeira Forma Normal (1FN) é um conceito fundamental em bancos de dados relacionais, que visa organizar os dados de forma eficiente e livre de redundância. Para atender aos requisitos da 1FN, uma tabela de banco de dados deve atender a certas regras específicas, que são essenciais para a integridade e consistência dos dados.

A Segunda Forma Normal (2FN) é uma etapa crucial no processo de normalização de bancos de dados. Ela se concentra na eliminação de dependências parciais, que ocorrem quando uma coluna em uma tabela depende apenas de uma parte da chave primária composta. Para entender a 2FN, é importante primeiro compreender a estrutura da chave primária composta e como as dependências funcionais são identificadas.

Na Terceira Forma Normal (3FN), buscamos eliminar as dependências transitivas que ainda possam existir após a aplicação da Segunda Forma Normal. Isso significa que, em uma tabela, cada campo não chave deve depender apenas da chave primária, e não de outros campos não chave.

3- A normalização de banco de dados é essencial para garantir que os dados sejam organizados de maneira eficaz e confiável. Ela oferece vantagens como melhor desempenho de consulta, simplificação de operações de manutenção e integridade de dados.

2 - FORMAS NORMAIS

1- Significa que possui chave primária; Não possui grupos repetitivos; Todos os seus atributos são atômicos, ou seja, não precisa ser decomposto.

2- Primeira Forma Normal (1NF): Elimina grupos repetitivos e garante que cada valor em uma tabela seja atômico. Segunda Forma Normal (2NF): Elimina dependências parciais e garante que todos os atributos de uma tabela dependam totalmente da chave primária.

3- Para estar na 3FN precisa estar na 2FN;
Geração de novas tabelas com DF diretas; Análise de dependências funcionais entre atributos não chave; Verificar a dependência exclusiva da chave primária; Entidades na 3FN também não podem conter atributos que sejam resultados de algum cálculo de outro atributo.

3 - APLICAÇÃO E NORMALIZAÇÃO

1- Cada tabela em 1FN tem uma chave primária única que identifica cada fila da tabela. A chave primária pode ser composta por uma ou mais colunas da tabela. Cada coluna pode ter um só valor. Cada célula da tabela deve ter um valor atômico, isto é, este não pode ser dividido.

2- Para chegar a segunda forma normal verifique se a chave primária dessa tabela é composta ou simples. Se for simples, já se encontra na segunda forma normal. Se for composta, verifique se todos os atributos da relação dependem de todos os atributos que compõem a chave primária. Por exemplo, se a chave primária é composta dos atributos A , B e o campo C em questão depende somente de B. Se sim, já está na segunda forma normal. Se não, pegue o atributo (C) que depende parcialmente da chave primária e crie uma nova tabela. Essa tabela terá como chave primária o campo da chave primária original que determinou o campo C (nesse exemplo é o B) e adicione como atributo simples da relação o C.

3- Para transformar uma tabela que não está na Terceira Forma Normal (3NF) para que ela esteja em 3NF, você precisa seguir alguns passos que envolvem normalização e decomposição da tabela. Vou descrever o processo geral:

1. Revisão das Formas Normais Anteriores

Primeiro, é importante entender as Formas Normais anteriores (1NF e 2NF):

Primeira Forma Normal (1NF): A tabela deve ter apenas valores atômicos (não pode haver conjuntos de valores ou listas dentro de uma célula).

Segunda Forma Normal (2NF): A tabela deve estar na 1NF e todos os atributos não-chave devem depender completamente da chave primária (ou seja, não deve haver dependências parciais).

2. Identificar Dependências Funcionais

Identifique todas as dependências funcionais na tabela. Uma dependência funcional é uma relação entre dois conjuntos de atributos, onde um conjunto de atributos determina o valor de outro.

3. Verificar a Terceira Forma Normal (3NF)

Para que uma tabela esteja em 3NF, ela deve atender aos seguintes critérios:

Estar na 2NF.

Não deve haver dependências transitivas, ou seja, não deve haver atributos não-chave que dependem de outros atributos não-chave. Em outras palavras, cada atributo não-chave deve depender diretamente da chave primária.

4. Decomposição da Tabela

Se a tabela não está em 3NF, você precisará decompor a tabela para eliminar dependências transitivas. Aqui está o processo de decomposição:

Identifique Dependências Transitivas: Verifique quais atributos não-chave dependem de outros atributos não-chave.

Crie Tabelas Separadas: Para cada dependência transitiva identificada, crie uma nova tabela para representar essa dependência. A tabela original deve manter apenas as dependências diretas da chave primária.

4 - DESAFIOS E CONSIDERAÇÕES

1- É aceitável a desnormalização e é recomendada quando há necessidade de reduzir o número de tabelas envolvidas em consultas que acontecem com frequência, com a consequente redução do uso de recursos pelo SGBD.

5 - EXEMPLOS PRÁTICOS

1- um banco de dados para um sistema de gerenciamento de cursos em uma universidade. Queremos armazenar informações sobre cursos, alunos e inscrições em cursos.

1. Tabelas Normalizadas

Aqui estão três tabelas normalizadas para esse banco de dados:

1. Tabela Cursos

ID_Curso	Nome_Curso	Departamento
1	Matemática	Ciências Exatas
2	História	Ciências Humanas
3	Biologia	Ciências Biológicas

2.

Tabela Alunos

ID_Aluno	Nome_Aluno	Data_Nascimento
1	João Silva	2000-01-15
2	Maria Oliveira	1999-07-22
3	Pedro Santos	2001-11-05

3.

Tabela Inscrições

ID_Inscrição	ID_Aluno	ID_Curso	Data_Inscrição
1	1	2	2024-01-10
2	1	3	2024-02-01
3	2	1	2024-01-15

Formas Normais Aplicadas

1. Primeira Forma Normal (1NF)

A 1NF exige que todos os atributos da tabela tenham valores atômicos e que não haja duplicação de registros. No exemplo:

- Cada tabela (Cursos, Alunos, e Inscrições) tem atributos atômicos. Por exemplo, a tabela Cursos tem colunas separadas para ID_Curso, Nome_Curso e Departamento, e não há listas ou conjuntos de valores em uma célula.
- Não há registros duplicados dentro de cada tabela.

2. Segunda Forma Normal (2NF)

A 2NF exige que a tabela esteja na 1NF e que todos os atributos não-chave dependam completamente da chave primária (ou seja, não deve haver dependências parciais).

- **Tabela Cursos:** A chave primária é ID_Curso, e todos os atributos (Nome_Curso e Departamento) dependem completamente de ID_Curso.
- **Tabela Alunos:** A chave primária é ID_Aluno, e todos os atributos (Nome_Aluno e Data_Nascimento) dependem completamente de ID_Aluno.
- **Tabela Inscrições:** A chave primária é ID_Inscrição, e todos os atributos (ID_Aluno, ID_Curso, e Data_Inscrição) dependem completamente de ID_Inscrição.

3. Terceira Forma Normal (3NF)

A 3NF exige que a tabela esteja na 2NF e que não haja dependências transitivas (ou seja, atributos não-chave não devem depender de outros atributos não-chave).

- **Tabela Cursos:** Está na 3NF porque não há atributos não-chave que dependam de outros atributos não-chave. Nome_Curso e Departamento dependem diretamente de ID_Curso.
- **Tabela Alunos:** Está na 3NF porque todos os atributos (Nome_Aluno e Data_Nascimento) dependem diretamente de ID_Aluno.
- **Tabela Inscrições:** Está na 3NF porque ID_Aluno e ID_Curso são chaves estrangeiras que referenciam outras tabelas e não há atributos não-chave que dependem de outros atributos não-chave dentro da tabela Inscrições.

6

1- Existem diversos sites e aplicativos bons que fazem tutoriais e cursos online.

2- Um exemplo é a plataforma [Twygo](#).

referencias bibliográficas:

<https://learn.microsoft.com/pt-br/office/troubleshoot/access/database-normalization-description>

<https://spaceprogrammer.com/bd/normalizando-um-banco-de-dados-por-meio-das-3-principais-formas/#:~:text=Primeira%20Forma%20Normal,-Podemos%20dizer%20que&text=Possui%20chave%20prim%C3%A1ria%3B,seja%20C%20n%C3%A3o%20precisa%20ser%20decomposto>.

<https://guiadehospedagem.com.br/normalizacao-banco-de-dados-1fn-2fn-3fn/#:~:text=A%20normaliza%C3%A7%C3%A3o%20de%20banco%20de,eliminand o%20poss%C3%ADveis%20problemas%20de%20inconsist%C3%Aancia>.

<https://www.dio.me/articles/normalizacao-de-banco-de-dados-por-que-organizacao-e-fundamental>

<https://www.dio.me/articles/organizando-um-banco-de-dados-usando-as-formas-normais>

https://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados

[https://ebaonline.com.br/blog/normalizacao-de-bases-de-dados#:~:text=Primeira%20forma%20normal%20\(1FN\)&text=Cada%20tabela%20em%201FN%20tem,este%20n%C3%A3o%20pode%20ser%20dividido.](https://ebaonline.com.br/blog/normalizacao-de-bases-de-dados#:~:text=Primeira%20forma%20normal%20(1FN)&text=Cada%20tabela%20em%201FN%20tem,este%20n%C3%A3o%20pode%20ser%20dividido.)

<https://spaceprogrammer.com/bd/normalizando-um-banco-de-dados-por-meio-das-3-principais-formas/>

https://twygo.com/plataforma-lms-ads/?utm_source=google&utm_medium=pesquisapaga&utm_campaign=pmax&utm_source=&utm_medium=&utm_campaign=&utm_content={conteudo}&utm_term=&hsa_acc=4047120479&hsa_cam=21247574583&hsa_grp=&hsa_ad=&hsa_src=x&hsa_tgt=&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=Cj0KCQjwtsy1BhD7ARIsAHOi4xaZZAING24xs0eZ9SDLI0O7W2eOBAXj25nYjEng6lXZic-sUxssODgaAvXuEALw_wcB