



Criação de Banco de Dados

1

Banco de Dados II



Banco de Dados II

Criação de Banco de Dados

Um banco de dados bem planejado dá aos usuários acesso a informações essenciais. Ao seguir os princípios apresentados nesta apresentação, você poderá criar um banco de dados que executa bem e se adapta às necessidades futuras. Abordaremos os princípios básicos para se criar um banco de dados, bem como maneiras de refiná-lo para obter melhores resultados.



Banco de Dados II

O processo de criação de banco de dados

Um banco de dados bem-estruturado:

- Economiza espaço em disco ao eliminar dados redundantes.
- Mantém a exatidão e a integridade dos dados.
- Oferece acesso aos dados de maneiras úteis.

Criar um banco de dados eficiente e útil é uma questão de seguir o processo adequado, incluindo as fases a seguir:



Banco de Dados II

O processo de criação de banco de dados

1. Análise de requisitos, ou identificação do objetivo do banco de dados
2. Organizando dados em tabelas
3. Especificando chaves primárias e analisando relações
4. Normalizando para padronizar as tabelas

Vamos analisar cada passo mais de perto. Observe que este guia trata do modelo de banco de dados relacional de Edgar Codd, conforme escrito em SQL (em vez de modelos de dados hierárquicos, rede ou objetos).



Banco de Dados II

Análise de requisitos: identificando o objetivo do banco de dados

Compreender a finalidade do banco de dados servirá de base para informar suas escolhas durante todo o processo de criação. Certifique-se de considerar o banco de dados de todas as perspectivas. Por exemplo, se você estiver criando um banco de dados para uma biblioteca pública, seria importante considerar as maneiras pelas quais os usuários e bibliotecários precisariam acessar os dados.



Banco de Dados II

Análise de requisitos: identificando o objetivo do banco de dados

Aqui estão algumas maneiras de coletar informações antes de criar o banco de dados:

- Entreviste as pessoas que o usarão
- Analise formulários corporativos, como faturas, quadros de horários e pesquisas
- Faça um pente fino em todos os sistemas de dados existentes (incluindo arquivos físicos e digitais)



Banco de Dados II

Análise de requisitos: identificando o objetivo do banco de dados

Comece reunindo todos os dados existentes que serão incluídos no banco de dados. Em seguida, liste os tipos de dados que você deseja armazenar e as entidades, ou pessoas, coisas, locais e eventos que esses dados descrevem, assim:

Clientes

- Nome
- Endereço
- Cidade, Estado, CEP
- Endereço de e-mail

Produtos

- Nome
- Preço
- Quantidade em estoque
- Quantidade em pedidos

Pedidos

- ID do pedido
- Representante de vendas
- Data
- Produto(s)
- QUANTIDADE
- Preço
- Total



Banco de Dados II

Análise de requisitos: identificando o objetivo do banco de dados

Essas informações mais tarde se tornarão parte do dicionário de dados, que descreve as tabelas e os campos dentro do banco de dados. Certifique-se de dividir as informações em pequenas partes úteis. Por exemplo, considere separar o endereço do país para que você possa filtrar mais tarde as pessoas pelo país de residência. Além disso, evite colocar o mesmo ponto de dados em mais de uma tabela, o que acrescenta complexidade desnecessária.

Depois de saber quais os tipos de dados que o banco de dados incluirá, de onde esses dados vêm e como eles serão usados, você estará pronto para começar a planejar o banco de dados real.



Banco de Dados II

Estrutura de banco de dados: os blocos de construção de um banco de dados

O próximo passo é estabelecer uma representação visual do seu banco de dados. Para fazer isso, você precisa entender exatamente como os bancos de dados relacionais são estruturados.

Dentro de um banco de dados, os dados relacionados são agrupados em tabelas, cada uma das quais consiste em linhas (também chamadas tuplas) e colunas, como uma planilha.



Banco de Dados II

Estrutura de banco de dados: os blocos de construção de um banco de dados

Para converter suas listas de dados em tabelas, comece criando uma tabela para cada tipo de entidade, como produtos, vendas, clientes e pedidos. Eis um exemplo:

Cada linha de uma tabela é chamada de registro. Os registros incluem dados sobre algo ou alguém, como um cliente em particular. Como comparação, as colunas (também conhecidas como campos ou atributos) contêm um único tipo de informação que aparece em cada registro, como os endereços de todos os clientes listados na tabela.



Banco de Dados II

Estrutura de banco de dados: os blocos de construção de um banco de dados

1º nome	Sobrenome	Idade	Código postal
Roger	Williams	43	34760
Jerrica	Jorgensen	32	97453
Samantha	Hopkins	56	64829



Banco de Dados II

Estrutura de banco de dados: os blocos de construção de um banco de dados

Para manter os dados consistentes de um registro para o outro, atribua o tipo de dados apropriado a cada coluna. Os tipos de dados comuns incluem:

- CHAR - um tamanho específico de texto
- VARCHAR - texto de tamanhos variáveis
- TEXT - grandes quantidades de texto
- INT - número inteiro positivo ou negativo
- FLOAT, DOUBLE - também podem armazenar números de pontos flutuantes
- BLOB - dados binários



Banco de Dados II

Estrutura de banco de dados: os blocos de construção de um banco de dados

Alguns sistemas de gestão de banco de dados também oferecem o tipo de dados Autonumeração, que gera automaticamente um número exclusivo em cada linha.

Para criar uma visão geral do banco de dados, conhecido como um diagrama entidade-relacionamento, você não incluirá as tabelas reais. Em vez disso, cada tabela se torna uma caixa no diagrama. O título de cada caixa deve indicar o que os dados nessa tabela descrevem, enquanto os atributos são listados abaixo, assim:

Student

Student ID

Birth date

Grade level



Banco de Dados II

Estrutura de banco de dados: os blocos de construção de um banco de dados

Finalmente, você deve decidir qual atributo ou quais atributos servirão como chave primária para cada tabela, se houver. Uma chave primária (PK) é um identificador exclusivo para uma determinada entidade, o que significa que você poderia escolher um cliente exato, mesmo se você só conhecesse o valor..



Banco de Dados II

Estrutura de banco de dados: os blocos de construção de um banco de dados

Os atributos escolhidos como chaves primárias devem ser únicos, imutáveis e sempre presentes (nunca NULOS ou vazios). Por esta razão, os números de pedido e nomes de usuários são boas chaves primárias, ao contrário de números de telefone ou endereços. Você também pode usar vários campos em conjunto como chave primária (isso é conhecido como uma chave composta).



Banco de Dados II

Criando relações entre entidades

Com suas tabelas de banco de dados agora convertidas em tabelas, você está pronto para analisar as relações entre essas tabelas. A cardinalidade se refere à quantidade de elementos que interagem entre duas tabelas relacionadas. Identificar a cardinalidade ajuda a garantir que você tenha dividido os dados em tabelas de forma mais eficiente.

Cada entidade pode potencialmente ter uma relação com todas as outras, mas essas relações normalmente são uma de três tipos:

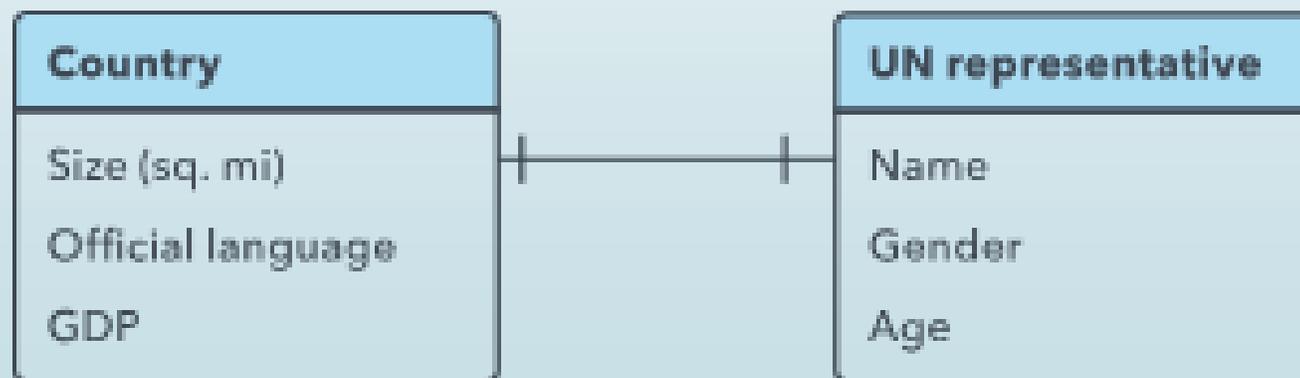


Banco de Dados II

Criando relações entre entidades

Relações uma a uma

Quando há apenas uma instância da Entidade A para cada instância da Entidade B, diz-se que elas têm uma relação uma para uma (frequentemente escrito 1:1). Você pode indicar esse tipo de relação em um diagrama ER com uma linha com um traço em cada extremidade:





Banco de Dados II

Criando relações entre entidades

Relações uma a uma

A menos que você tenha uma boa razão em contrário, uma relação 1:1 geralmente indica que seria melhor combinar os dados das duas tabelas em uma única tabela.

No entanto, convém criar tabelas com uma relação 1:1 sob um determinado conjunto de circunstâncias. Se você tiver um campo com dados opcional, como "descrição", que esteja em branco para muitos dos registros, você pode mover todas as descrições para sua própria tabela, eliminando o espaço vazio e melhorando o desempenho do banco de dados.

Para garantir que os dados correspondam corretamente, você teria então de incluir pelo menos uma coluna idêntica em cada tabela, provavelmente a chave primária.

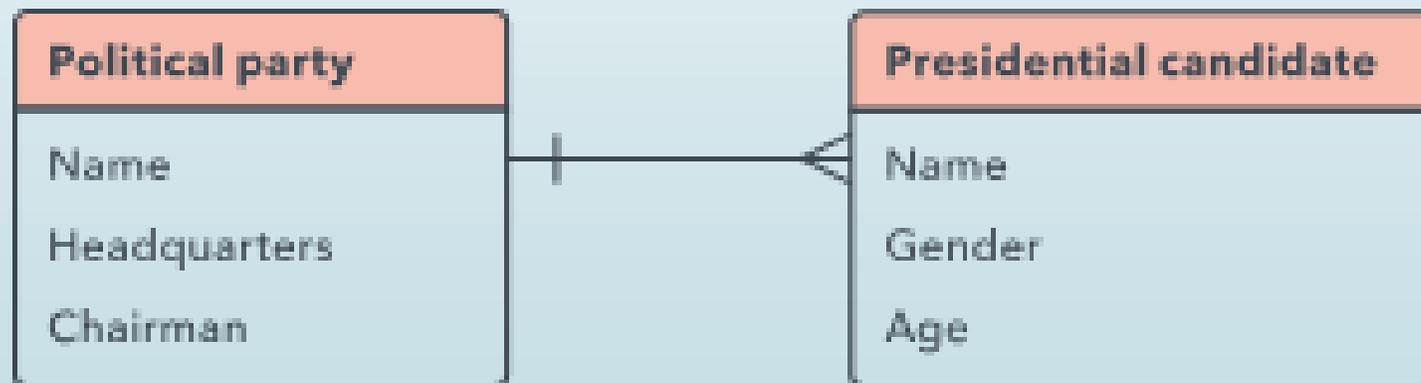


Banco de Dados II

Criando relações entre entidades

Relações uma para muitas

Essas relações ocorrem quando um registro em uma tabela é associado com várias entradas em outra tabela. Por exemplo, um único cliente pode ter feito muitos pedidos, ou um cliente pode ter vários livros retirados da biblioteca de uma só vez. As relações uma para muitas (1:M) são indicadas com o que é chamado de "notação pé de galinha", como neste exemplo:





Banco de Dados II

Criando relações entre entidades

Relações uma para muitas

Para implementar uma relação 1:M à medida que você configura um banco de dados, basta adicionar a chave primária do lado "um" da relação como um atributo na outra tabela. Quando uma chave primária é listada em outra tabela desta forma, ela é chamada de chave estrangeira. A tabela no lado "1" da relação é considerada a tabela primária em relação à tabela secundária no outro lado.



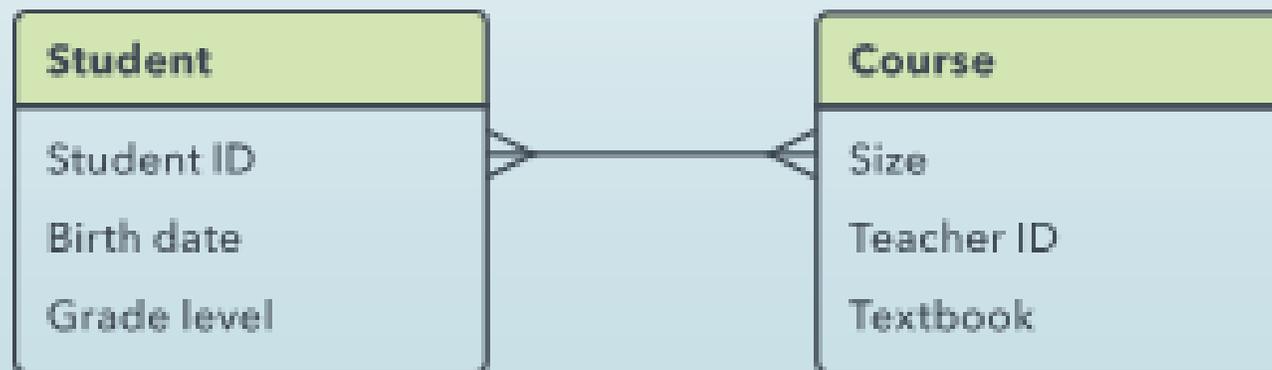
Banco de Dados II

Criando relações entre entidades

Relações muitas para muitas

Quando várias entidades de uma tabela podem ser associadas a várias entidades em outra tabela, diz-se que elas têm uma relação muitas para muitas (M:N). Isso pode acontecer no caso de alunos e aulas, uma vez que um aluno pode ter muitas aulas e uma aula pode ter muitos alunos.

Em um diagrama ER, essas relações são retratadas com estas linhas:





Banco de Dados II

Criando relações entre entidades

Relações muitas para muitas

Infelizmente, não é diretamente possível implementar esse tipo de relação em um banco de dados. Em vez disso, você tem que dividi-lo em duas relações uma para muitas.

Para fazer isso, crie uma nova entidade entre essas duas tabelas. Se a relação M:N existir entre vendas e produtos, você pode chamar essa nova entidade de "produtos_vendidos", uma vez que ela mostraria o conteúdo de cada venda. As tabelas de vendas e produtos teriam uma relação 1:M com produtos_vendidos. Esse tipo de entidade intermediária é chamada de tabela de ligação, entidade associativa ou tabela de junção em vários modelos.



Banco de Dados II

Criando relações entre entidades

Relações muitas para muitas

Cada registro na tabela de ligação corresponderia às duas entidades nas tabelas vizinhas (também pode incluir informações suplementares). Por exemplo, uma tabela de ligação entre alunos e aulas pode ter esta aparência:



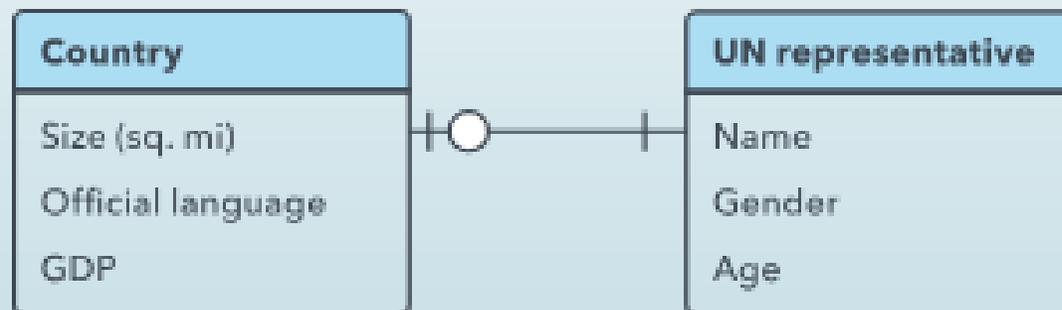


Banco de Dados II

Criando relações entre entidades

Obrigatório ou não?

Outra maneira de analisar as relações é considerar qual lado da relação tem que existir para que o outro exista. O lado não obrigatório pode ser marcado com um círculo sobre linha onde estaria um traço. Por exemplo, um país tem que existir para que ele tenha um representante nas Nações Unidas, mas o oposto não é verdade:



Duas entidades podem ser mutuamente dependentes (uma não poderia existir sem a outra).



Banco de Dados II

Criando relações entre entidades

Relações recursivas

Às vezes, uma tabela aponta para si mesma. Por exemplo, uma tabela de funcionários pode ter um atributo "gerente" que se refere a outro indivíduo na mesma tabela. Isso é chamado de relação recursiva.



Banco de Dados II

Criando relações entre entidades

Relações redundantes

Uma relação redundante é aquela expressa mais de uma vez. Normalmente, é possível remover uma das relações sem perder nenhuma informação importante. Por exemplo, se uma entidade "alunos" tem uma relação direta com outra entidade chamada "professores", mas também tem uma relação com os professores indiretamente através de "aulas", seria bom remover a relação entre "alunos" e "professores". É melhor excluir essa relação, porque a única maneira que os alunos são atribuídos aos professores é através das aulas.



Banco de Dados II

Normalização de banco de dados

Depois de ter um design preliminar para o banco de dados, você pode aplicar regras de normalização para certificar-se de que as tabelas estão estruturadas corretamente. Pense nestas regras como padrões da indústria.

Dito isto, nem todos os bancos de dados são bons candidatos para a normalização. Em geral, os bancos de dados de processamento de transações on-line (OLTP, na abreviação em inglês), nos quais os usuários se preocupam com a criação, a leitura, a atualização e a exclusão de registros, devem ser normalizados.



Banco de Dados II

Normalização de banco de dados

Bancos de dados de processamento analítico on-line (OLAP, na sigla em inglês), que favorecem a análise e a elaboração de relatórios podem ser melhorados com um certo grau de desnormalização, uma vez que a ênfase é na velocidade de cálculo. Estes incluem aplicativos de suporte à decisão onde os dados precisam ser analisados rapidamente, mas não alterados.

Cada formulário, ou nível de normalização, inclui as regras associadas com os formulários inferiores.



Banco de Dados II

Normalização de banco de dados

Primeiro formulário normal

O primeiro formulário normal (abreviado como 1NF) especifica que cada célula na tabela pode ter apenas um valor, nunca uma lista de valores, portanto, uma tabela como esta é incorreta:

ID do produto	Cor	Preço
1	marrom, amarelo	US\$ 15
2	vermelho, verde	US\$ 13
3	azul, laranja	US\$ 11



Banco de Dados II

Normalização de banco de dados

Primeiro formulário normal

Você pode ser tentado a contornar isso dividindo esses dados em colunas adicionais, mas isso também é contra as regras: uma tabela com grupos de atributos repetidos ou estreitamente relacionados não atende ao primeiro formulário normal. A tabela abaixo, por exemplo, está incorreta:

Products
Color1
Color2
Color3
Price

Em vez disso, divida os dados em várias tabelas ou registros até que cada célula contenha apenas um valor e não haja colunas extras. Nesse ponto, diz-se que os dados são atômicos, ou divididos até o menor tamanho útil. Para a tabela acima, você poderia criar uma tabela adicional chamada "Detalhes de vendas" que corresponderia a produtos específicos com vendas. "Vendas" então teria uma relação 1:M com "Detalhes de vendas".



Banco de Dados II

Normalização de banco de dados

Segundo formulário normal

Além disso, uma tabela com uma chave primária composta de vários campos viola o segundo formulário normal se um ou mais dos outros campos não dependerem de cada parte da chave.

Assim, uma tabela com esses campos não atenderia ao segundo formulário normal, porque o atributo "nome do produto" depende de ID do produto, mas não do número do pedido:

- Número do pedido (chave primária)
- ID do produto (chave primária)
- Nome do produto



Banco de Dados II

Normalização de banco de dados

Terceiro formulário normal

O terceiro formulário normal (3NF) adiciona a essas regras a exigência de que cada coluna sem chave seja independente de qualquer outra coluna. Se a alteração de um valor em uma coluna sem chave faz com que outro valor seja alterado, essa tabela não atende ao terceiro formulário normal.



Banco de Dados II

Normalização de banco de dados

Terceiro formulário normal

Isso evita que você armazene quaisquer dados derivados na tabela, como a coluna "imposto" abaixo, que depende diretamente do preço total do pedido:

Pedido	Preço	Imposto
14325	US\$ 40,99	US\$ 2,05
14326	US\$ 13,73	US\$ 0,69
14327	US\$ 24,15	US\$ 1,21



Banco de Dados II

Normalização de banco de dados

Terceiro formulário normal

Formulários adicionais de normalização foram propostos, incluindo o formulário normal de Boyce-Codd, o quarto a sexto formulários normais, e o formulário normal de domínio-chave, mas os três primeiros são os mais comuns.

Embora esses formulários expliquem as práticas recomendadas a serem seguidas em geral, o grau de normalização depende do contexto do banco de dados.



Banco de Dados II

Normalização de banco de dados

Dados multidimensionais

Alguns usuários podem querer acessar várias dimensões de um único tipo de dados, especialmente em bancos de dados OLAP. Por exemplo, eles podem querer saber as vendas por cliente, estado e mês. Nesta situação, é melhor criar uma tabela central de fatos que outras tabelas de clientes, estados e meses podem consultar, como esta:





Banco de Dados II

Regras de integridade de dados

Você também deve configurar seu banco de dados para validar os dados de acordo com as regras apropriadas. Muitos sistemas de gestão de banco de dados, como o Microsoft Access, aplicam algumas dessas regras automaticamente.

A regra de integridade da entidade diz que a chave primária nunca pode ser NULA. Se a chave é composta de várias colunas, nenhuma delas pode ser NULA. Caso contrário, ele poderia não conseguir identificar o registro de forma exclusiva.



Banco de Dados II

Regras de integridade de dados

A regra de integridade referencial requer que cada chave estrangeira listada em uma tabela seja correspondida com uma chave primária na tabela que faz referência. Se a chave primária mudar ou for excluída, essas alterações precisarão ser implementadas sempre que essa chave for referenciada em todo o banco de dados.

As regras de integridade da lógica de negócios garantem que os dados se encaixem dentro de certos parâmetros lógicos. Por exemplo, o horário de consulta teria de estar dentro



Banco de Dados II

Adicionando índices e exibições

Um índice é essencialmente uma cópia classificada de mais uma ou colunas, com os valores em ordem ascendente ou descendente. Adicionar um índice permite aos usuários localizar registros mais rapidamente. Em vez de reclassificar para cada consulta, o sistema pode acessar registros na ordem especificada pelo índice.

Embora os índices acelerem a recuperação de dados, eles podem diminuir a inserção, atualização e exclusão, uma vez que o índice precisa ser reconstruído sempre que um registro é alterado.

Uma exibição é simplesmente uma consulta salva sobre os dados. Eles podem unir dados úteis de várias tabelas ou mostrar parte de uma tabela.



Banco de Dados II

Propriedades estendidas

Depois de concluir o layout básico, é possível refinar o banco de dados com propriedades estendidas, como texto instrucional, máscaras de entrada e regras de formatação que se aplicam a um determinado esquema, exibição ou coluna. A vantagem é que, como essas regras são armazenadas no próprio banco de dados, a apresentação dos dados será consistente entre os vários programas que acessam os dados.



Banco de Dados II

SQL e UML

A UML (Unified Modeling Language) é outra forma visual de expressar sistemas complexos criados em uma linguagem orientada a objetos. Vários dos conceitos mencionados neste guia são conhecidos na UML sob diferentes nomes. Por exemplo, uma entidade é conhecida como uma classe na UML.

A UML não é usada tão frequentemente hoje como era antes. Hoje em dia, ela é frequentemente usada academicamente e em comunicações entre projetistas de software e seus clientes.



SQL e UMLDados II

Sistemas de gestão de bancos de dados

Muitas das escolhas de design que você fará dependem de qual sistema de gestão de banco de dados você usa. Alguns dos sistemas mais comuns incluem:

- Oracle DB
- MySQL
- Microsoft SQL Server
- PostgreSQL
- IBM DB2

Quando houver escolha, selecione um sistema de gestão de banco de dados apropriado baseado em custos, sistemas operacionais, recursos e muito mais.



Referências

- **Lucidchart**

www.lucidchart.com/pages/pt/tutorial-de-criacao-e-estruturacao-de-banco-de-dados#discoveryTop