



Bancos de Dados

Fundamentos de Bancos de Dados

Bancos de dados e seus elementos estão presentes em nosso dia a dia, sendo elementos indispensáveis para a sociedade atual. A todo momento estamos lidando, mesmo que sem perceber, com ações que envolvem algum tipo de banco de dados. Um saque em um caixa eletrônico, o aluguel de um carro, reserva um quarto de hotel, são apenas alguns dos exemplos de ações que nos colocam em contato com banco de dados.

Podemos definir os dados como sendo fatos que podemos representar com medidas ou escalas ou mesmo símbolos formais dentro de um modelo formal que representa o dito “mundo real”.

O motivo pelo qual utilizamos modelos é porque não podemos transportar o “mundo real” para dentro de um computador ou outro dispositivo computacional. Utilizamos modelos para que possamos refletir as coisas que consideramos importantes no mundo real e as entidades e propriedade que necessitamos manipular.

Dado Vs. Informação

Para fazermos uma diferenciação definimos informação como “aquilo que é obtido quando depuramos os dados brutos.”. Dados brutos podem não fazer muito sentido quando se precisa tomar decisões, mas quando levados a um nível de abstração mais elevado eles se tornam informações.

Imagine um agricultor que utiliza um sistema de medida, contagem, para sua plantação, armazenando quanto foi produzido a cada safra. Isso já seria muito interessante, pois ele poderia fazer levantamentos de quando um ano foi melhor ou pior que outro. Mas imagine esse mesmo agricultor armazenando dados como: “esse ano houve cheia e x% da plantação foi perdida” ou ainda: “plantamos 50% de cultura x e 50% de cultura y, e isso foi bem proveitoso”. São abstrações que vão além de uma simples contagem. Essas abstrações podem ajudar o agricultor a tomar decisões do tipo: “no período dos meses x e y, melhor evitar plantar cultura a, pois é época de cheias” e por aí vai. Esse tipo de informação “qualitativa” pode ser muito mais eficiente do que a “quantitativa”, pois permite que façamos previsões.

Entidades, Atributos, Valores e Relacionamentos

Entidades, atributos, valores e relacionamentos são os componentes de um modelo relacional. No SQL, a representação é feita através de linhas e de colunas. Diferentemente de planilhas, o Modelo Relacional é feito de linhas e as linhas são as colunas.

Entidades

Uma entidade pode ser um objeto concreto em sua realidade, como uma pessoa ou coisa, ou pode ser uma relação entre objetos em sua realidade, como um casamento, que pode ser tratado como se fosse um objeto. Não é óbvio que algumas informações devam sempre ser modeladas como uma entidade, um atributo ou um relacionamento. Mas pelo menos no SQL você terá uma tabela para cada classe de entidade, e cada linha representará uma instância dessa classe.

- **Entidades como Objetos:** De um modo geral, os objetos são passivos e sofrem ação no modelo. Seus atributos são alterados por processos fora de si mesmos. A

rigor, cada linha de uma tabela de objetos deve corresponder a uma “coisa” na realidade do banco de dados, mas nem sempre de forma única. É mais conveniente manusear uma tigela de arroz como uma única coisa em vez de fornecer um número de peça para cada grão.

- **Entidades como relacionamentos:** Um relacionamento é uma maneira de unir objetos para obter novas informações que existem separadamente dos objetos específicos. O problema é que o relacionamento é muitas vezes representado por algum tipo de símbolo na realidade.

Atributos

Os atributos pertencem às entidades e as definem. Leibniz chegou mesmo a dizer que uma entidade é a soma de todos os seus atributos. O SQL concorda com essa instrução e modela atributos como colunas nas linhas de tabelas que podem assumir valores.

Valores

Um valor pertence a um atributo. O valor específico para um atributo específico é extraído de um domínio ou tem um tipo de dados. Existem várias escolas de pensamento sobre domínios, tipos de dados e valores, mas as duas principais escolas são as seguintes:

- Tipos de dados e domínios são conjuntos de valores no banco de dados. Ambos são conjuntos finitos porque todos os modelos são finitos. O tipo de dados difere por ter operadores no hardware ou software para que o usuário do banco de dados não precise fazer todo esse trabalho. Um domínio é construído em um subconjunto de um tipo de dados, que herda alguns ou todos os seus

operadores do tipo de dados e restrições originais, mas agora o banco de dados pode ter operadores definidos pelo usuário no domínio.

- Um domínio é um conjunto finito ou infinito de valores com operadores que existe na realidade do banco de dados. Um tipo de dados é um subconjunto de um domínio suportado pelo computador no qual o banco de dados reside. O banco de dados aproxima um domínio com um subconjunto de um tipo de dados, que herda alguns ou todos os seus operadores do tipo de dados original e outras restrições e operadores dados a ele pelo designer do banco de dados.

Relacionamentos

Existem relacionamentos entre entidades. Já falamos sobre entidades como relacionamentos e como a linha não é clara quando você cria um modelo.

Modelagem ER: Em 1976, Peter Chen inventou a modelagem entidade-relacionamento (ER) como uma técnica de design de banco de dados. Os diagramas originais usavam uma caixa para uma entidade, um diamante para um relacionamento e linhas para conectá-los. A simplicidade dos diagramas usados neste método o tornaram a técnica de design de banco de dados mais popular em uso hoje. O método original era muito mínimo, então outras pessoas adicionaram outros detalhes e símbolos ao diagrama básico.

O que é Modelo Relacional?

O Modelo Relacional (RM) representa o banco de dados como uma coleção de relações. Uma relação nada mais é do que uma tabela de valores. Cada linha na

tabela representa uma coleção de valores de dados relacionados. Essas linhas na tabela denotam uma entidade ou relacionamento do mundo real.

O nome da tabela e os nomes das colunas são úteis para interpretar o significado dos valores em cada linha. Os dados são representados como um conjunto de relações. No modelo relacional, os dados são armazenados como tabelas. No entanto, o armazenamento físico dos dados é independente da forma como os dados são organizados logicamente.

Sistemas de gerenciamento de banco de dados relacionais são:

- DB2 e Informix Dynamic Server – IBM
- Oracle e RDB – Oracle
- SQL Server e Acesso – Microsoft

Conceitos de modelo relacional em DBMS

- **Atributo:** Cada coluna em uma Tabela. Atributos são as propriedades que definem uma relação, por exemplo, Student_Rollno, NAME, etc.
- **Tabelas** – No modelo Relacional as relações são salvas no formato de tabela. Ele é armazenado junto com suas entidades. Uma tabela tem duas linhas e colunas de propriedades. As linhas representam registros e as colunas representam atributos.
- **Tupla** – nada mais é do que uma única linha de uma tabela, que contém um único registro.

- **Esquema de relação:** Um esquema de relação representa o nome da relação com seus atributos.
- **Grau:** O número total de atributos que na relação é chamado de grau da relação.
- **Cardinalidade:** Número total de linhas presentes na Tabela.
- **Coluna:** A coluna representa o conjunto de valores para um atributo específico.
- **Instância de relação** – A instância de relação é um conjunto finito de tuplas no sistema RDBMS. As instâncias de relação nunca têm tuplas duplicadas.
- **Chave de relação** – Cada linha tem um, dois ou vários atributos, que é chamado de chave de relação.
- **Domínio de atributo** – Cada atributo tem algum valor e escopo pré-definidos que é conhecido como domínio de atributo

Restrições de Integridade Relacional

Restrições de Integridade Relacional em SGBD são referidas a condições que devem estar presentes para uma relação válida. Essas restrições relacionais no DBMS são derivadas das regras no minimundo que o banco de dados representa.

Existem muitos tipos de restrições de Integridade no DBMS. As restrições no sistema de gerenciamento de banco de dados relacional são divididas principalmente em três categorias principais:

- Restrições de domínio
- Principais restrições

- Restrições de integridade referencial

Restrições de domínio

As restrições de domínio podem ser violadas se um valor de atributo não estiver aparecendo no domínio correspondente ou não for do tipo de dados apropriado.

As restrições de domínio especificam isso em cada tupla e o valor de cada atributo deve ser exclusivo. Isso é especificado como tipos de dados que incluem tipos de dados padrão inteiros, números reais, caracteres, booleanos, strings de comprimento variável, etc.

Restrições de chave

Um atributo que pode identificar exclusivamente uma tupla em uma relação é chamado de chave da tabela. O valor do atributo para diferentes tuplas na relação deve ser único.

Restrições de integridade referencial

As restrições de Integridade Referencial em DBMS são baseadas no conceito de Chaves Estrangeiras. Uma chave estrangeira é um atributo importante de uma relação que deve ser referido em outros relacionamentos. O estado de restrição de integridade referencial acontece onde a relação se refere a um atributo chave de uma relação diferente ou igual. No entanto, esse elemento-chave deve existir na tabela.

Operações no Modelo Relacional

Quatro operações básicas de atualização executadas no modelo de banco de dados relacional são: Inserir, atualizar, excluir e selecionar.

- Inserir é usado para inserir dados na relação
- Delete é usado para excluir tuplas da tabela.
- Modify permite alterar os valores de alguns atributos em tuplas existentes.
- Selecionar permite que você escolha um intervalo específico de dados.

Sempre que uma dessas operações for aplicada, as restrições de integridade especificadas no esquema do banco de dados relacional nunca devem ser violadas.

Práticas recomendadas para criar um modelo relacional

- Os dados precisam ser representados como uma coleção de relações
- Cada relação deve ser descrita claramente na tabela
- As linhas devem conter dados sobre instâncias de uma entidade
- As colunas devem conter dados sobre atributos da entidade
- As células da tabela devem conter um único valor
- Cada coluna deve receber um nome exclusivo
- Duas linhas não podem ser idênticas

- Os valores de um atributo devem ser do mesmo domínio

Vantagens do modelo de banco de dados relacional

- **Simplicidade:** Um modelo de dados relacional em SGBD é mais simples que o modelo hierárquico e de rede.
- **Independência Estrutural:** O banco de dados relacional se preocupa apenas com dados e não com uma estrutura. Isso pode melhorar o desempenho do modelo.
- **Fácil de usar:** O modelo Relacional em DBMS é fácil, pois as tabelas compostas por linhas e colunas são bastante naturais e simples de entender
- **Capacidade de consulta:** Possibilita que uma linguagem de consulta de alto nível, como SQL, evite a navegação complexa do banco de dados.
- **Independência de dados:** A estrutura do banco de dados relacional pode ser alterada sem a necessidade de alterar nenhum aplicativo.
- **Escalável:** Em relação ao número de registros, ou linhas, e ao número de campos, um banco de dados deve ser ampliado para melhorar sua usabilidade.

Desvantagens do Modelo Relacional

- Poucos bancos de dados relacionais têm limites de comprimento de campo que não podem ser excedidos.

- Os bancos de dados relacionais às vezes podem se tornar complexos à medida que a quantidade de dados aumenta e as relações entre os dados se tornam mais complicadas.
- Sistemas de banco de dados relacionais complexos podem levar a bancos de dados isolados onde as informações não podem ser compartilhadas de um sistema para outro.

Modelo Orientado a Objetos

Um banco de dados orientado a objetos é uma coleção de programação orientada a objetos e banco de dados relacional. Existem vários itens que são criados usando linguagens de programação orientadas a objetos, como C++, Java, que podem ser armazenadas em bancos de dados relacionais, mas bancos de dados orientados a objetos são adequados para esses itens.

Um banco de dados orientado a objetos é organizado em torno de objetos em vez de ações e dados em vez de lógica. Por exemplo, um registro multimídia em um banco de dados relacional pode ser um objeto de dados definível, em oposição a um valor alfanumérico.

Os Bancos de Dados Orientados a Objetos armazenam dados na forma de objetos. Um objeto é algo exclusivamente identificável que modela uma entidade do mundo real e tem estado e comportamento. Em Bancos de Dados baseados em Orientação a Objetos, as capacidades do paradigma baseado em Objetos para Programação e Bancos de Dados são combinados devido a remover as limitações dos Bancos de Dados Relacionais e por demanda de algumas aplicações avançadas.

Por que os Bancos de Dados Orientados a Objetos?

Existem três razões para a necessidade de OODBMS:

- Limitação do RDBMS
- Necessidade de aplicativos avançados
- Popularidade do Paradigma Orientado a Objetos

A. Limitação do RDBMS

Essas limitações estão no modelo relacional. Devido a isso essas limitações são refletidas para todos os RDBMS.

Essas limitações são:

- Má representação de entidades do mundo real: O modelo relacional não pode representar o mundo real de maneira correta porque tem apenas uma semântica que é tabela, que pode representar a entidade do mundo real de maneira correta.
- A normalização é necessária, mas às vezes não útil: Normalização em RDBMS para manter a consistência do banco de dados, mas algumas relações não estão relacionadas com o mundo real.
- Sobrecarga da estrutura semântica: Dados Relacionais modelo tem apenas uma estrutura semântica para representar dados e relacionamento que é tabela.
- Suporte insatisfatório para integridade e restrições corporativas:

Restrições são muito necessárias para que seu banco de dados tenha os dados desejados. O RDM suporta apenas um número limitado de restrições. As restrições corporativas são aquelas definidas pelos padrões da indústria.

- Estrutura de dados homogênea: RDM requer estruturas de dados homogêneas como:
- RDM assume homogeneidade horizontal e vertical.
- A álgebra matemática relacional tem apenas um número fixo de operações devido às quais as operações do Modelo Relacional não podem ser estendidas.
- As tabelas podem armazenar apenas valor atômico/único: Sem dúvida, isso é propriedade do RDM. Mas às vezes, em muitas situações, essa propriedade se torna sua limitação.
- A normalização é fortemente recomendada: Na maioria das situações, você deve normalizar a relação para tornar os dados consistentes dentro do seu banco de dados.
- Dificuldade em lidar com consultas recursivas: Há um suporte muito ruim para lidar com consultas recursivas no RDBMS.

Para isso você deve saber:

- A profundidade da consulta recursiva deve ser conhecida.
- Você pode usar as operações de fechamento transitivo para lidar com consultas recursivas no RDBMS.

Exemplos de BDOOs:

- ORION (Microeletronics Corporation), OPENOODB (Texas Instruments), Sistema IRIS (HP), ODE (Lucent Technologies), Projeto ENCORE/ObServer (Brown University).

Exemplos de sistemas de gerência de bancos de dados orientados a objetos (SGBDOO):

- GEMSTONE/OPAL, ONTOS, Objectivity, Versant, ObjectStore (O2), ARDENT, POET, Caché.
- O Object Data Management Group (ODMG) é uma organização de fabricantes e usuários de SGBDOOs que vem criando padrões para tais sistemas.

Modelos e Ferramentas

Entidade Relacionamento – ER

Na engenharia de software esse modelo descreve os **objetos**, **entidades**, que fazem parte de um domínio de negócios, os **atributos** são suas **características** e definem como é o **relacionamento** entre essas entidades. Esse modelo é utilizado como representação para a estrutura do banco de dados da aplicação, podendo o BD possuir várias entidades como chaves e tabelas intermediárias que só tem sentido no contexto de bases relacionais.

Entidades

São os objetos e demais partes envolvidas em um domínio e podem ser classificados como físicos ou lógicos, dependendo de sua existência no mundo real.

- **Entidades Físicas:** são as tangíveis, que realmente existem no mundo real.
Exemplos: uma pessoa, uma empresa ou um produto.
- **Entidades lógicas:** existem em decorrência da interação entre ou com as entidades físicas, não sendo objetos físicos no mundo real. Exemplos: uma venda ou um cadastro.

As entidades podem também ser classificadas segundo sua existência:

- **Entidades fortes:** independem de outras entidades, possuindo por si só sentido de existência. Exemplo: a entidade produto de um sistema de vendas.
- **Entidades fracas:** dependem de outras para existir, não fazendo sentido sozinhas. Exemplo: entidade venda depende de produto.
- **Entidades associativas:** existe quando há a necessidade de associação de uma entidade com um relacionamento já existente.

Relacionamentos

Os relacionamentos são definidos de acordo com a quantidade de objetos envolvidos em cada lado do relacionamento:

- **Relacionamento 1...1 (um para um):** cada entidade faz referência a apenas uma unidade da outra. Exemplo: Ficha do funcionário e funcionário, um funcionário só pode ter uma ficha, e uma ficha só pode pertencer a um funcionário.
- **Relacionamento 1...n ou 1...* (um para muitos):** uma entidade faz referência a várias unidades da outra entidade. Exemplo: sócio do clube e seus dependentes. Um sócio pode ter vários dependentes, mas cada dependente só possui um sócio.
- **Relacionamento n...n ou ... (muitos para muitos):** cada entidade de cada um dos lados, faz referência a várias unidades uma da outra. Exemplo: um curso, várias disciplinas, uma disciplina em vários cursos.

Atributos

Características que descrevem cada entidade dentro do domínio. Costumam ser identificados na análise de requisitos.

Quanto à função podem ser:

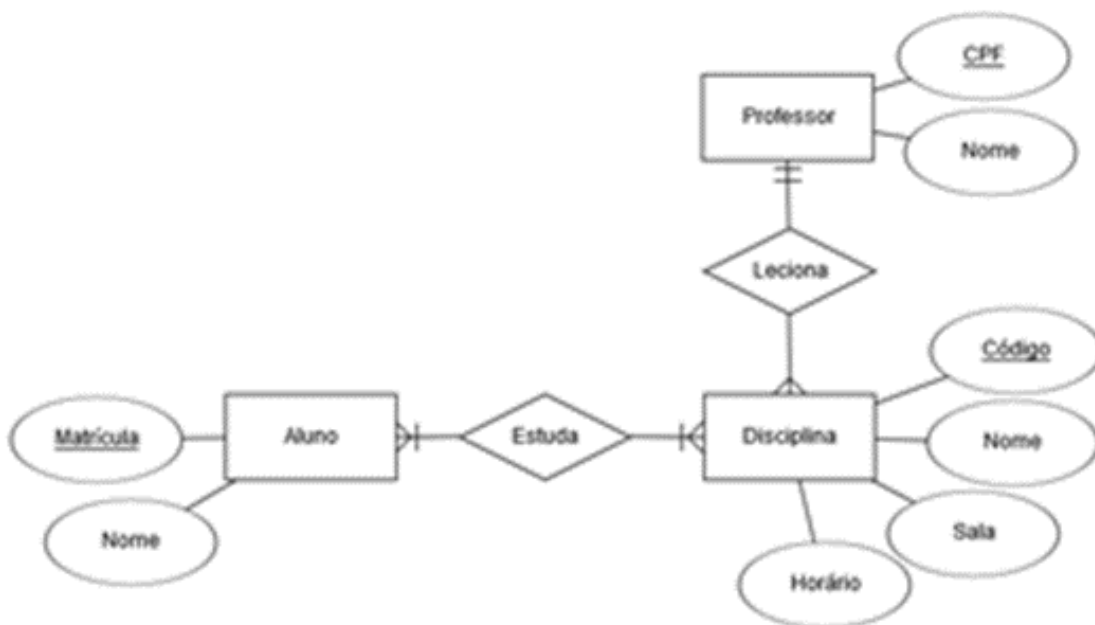
- **Descritivos:** representam características intrínsecas de uma entidade, tais como nome ou cor.
- **Nominativos:** são descritivos e definem e identificam um objeto.
- **Referenciais:** ligam uma entidade a outra em um relacionamento.

Quanto à sua estrutura:

- **Simples:** um único atributo define uma característica da entidade. Exemplos: nome, peso.
- **Compostos:** definem uma informação da entidade, são usados vários atributos. Por exemplo, o endereço pode ser composto por rua, número, bairro, etc.

Diagrama Entidade Relacionamento

Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a representação gráfica de um MER, sendo muitas vezes tomado como o próprio modelo.



Referência Bibliográfica

ELMASRI, Ramez. **Sistemas de banco de dados**. São Paulo: Pearson Addison Wesley, 2015.

Ir para exercício