

Revisão	Item	Natureza da Alteração	Data
000		Emissão – Introdução a modelagem de banco de dados	05/01/2024

Elaborador:	Brayan Silva	Data 05/01/2024
Revisor:		

1. Objetivo

Detalhar uma breve introdução em modelagem de banco de dados e a lógica de sistema de organização de um SGBD.

2. Aplicação

Esta explicação será utilizada em nosso curso, sem isso não conseguiremos partir para atividades ou colocar em prática conceitos aprendidos em aula.

3. Definições

SGBD – Sistema de Gerenciamento de Banco de Dados.

4. Equipamentos e Softwares

5. Condições Específicas

A principal motivação ao se trabalhar com sistemas de gerenciamento de bancos de dados está na diminuição de redundância de dados em relação a um processamento otimizado, uma vez que sua função é a de armazenar volumes crescentes de dados. A redundância de dados em um SGBD refere-se à presença desnecessária e repetitiva das mesmas informações em diferentes partes do banco de dados. Essa redundância pode surgir devido a várias razões, e pode ter implicações negativas na eficiência, consistência e manutenção do sistema. Aqui estão algumas considerações importantes sobre a redundância de dados:

Ineficiência de Armazenamento:

- A redundância aumenta a quantidade de armazenamento necessária para manter os dados, já que as mesmas informações são duplicadas em várias tabelas ou registros.
- Isso pode levar a um uso ineficiente do espaço em disco, especialmente em sistemas com grandes volumes de dados.

Problemas de Consistência:

- A presença de dados redundantes pode resultar em inconsistências se uma cópia dos dados for alterada, mas outra não.
- Atualizar todas as instâncias redundantes de uma informação torna-se um desafio e pode levar a dados desatualizados ou incorretos.

Dificuldade na Manutenção:

- Quando há redundância, qualquer alteração na estrutura dos dados requer atualizações em várias partes do banco de dados.
- A manutenção do sistema torna-se mais complexa e propensa a erros, aumentando o tempo e os recursos necessários para implementar mudanças.

Aumento do Tempo de Recuperação de Dados:

- Recuperar e manipular dados em um ambiente redundante pode resultar em operações mais lentas, especialmente em consultas que envolvem junções em várias tabelas.

Possibilidade de Inconsistência Referencial:

- A redundância pode levar a problemas de integridade referencial, especialmente quando não são implementadas devidas restrições para manter a consistência entre as cópias dos dados.

Segurança e Privacidade:

- A redundância pode aumentar os riscos de segurança, pois a duplicação de dados oferece mais oportunidades para violações de privacidade e acesso não autorizado.

Algumas estratégias surgem em relação ao trabalho com essas redundâncias.

Normalização:

- A normalização é uma técnica de projeto de banco de dados que visa reduzir a redundância, dividindo as tabelas em partes menores e mais coesas.
- Normalizar um banco de dados significa aplicar regras específicas para organizar os dados de maneira eficiente e reduzir a duplicação.

Uso de Chaves Estrangeiras:

- Utilizar chaves estrangeiras para estabelecer relacionamentos entre tabelas, reduzindo a necessidade de repetir informações em várias tabelas.

Desnormalização Controlada:

- Em alguns casos, é aceitável realizar desnormalização controlada para otimizar o desempenho em consultas específicas, mas isso deve ser feito com cuidado para evitar problemas de consistência.

SGBDs que Suportam Views:

- Utilizar Views para criar representações virtuais dos dados, sem a necessidade de armazenar informações redundantes.

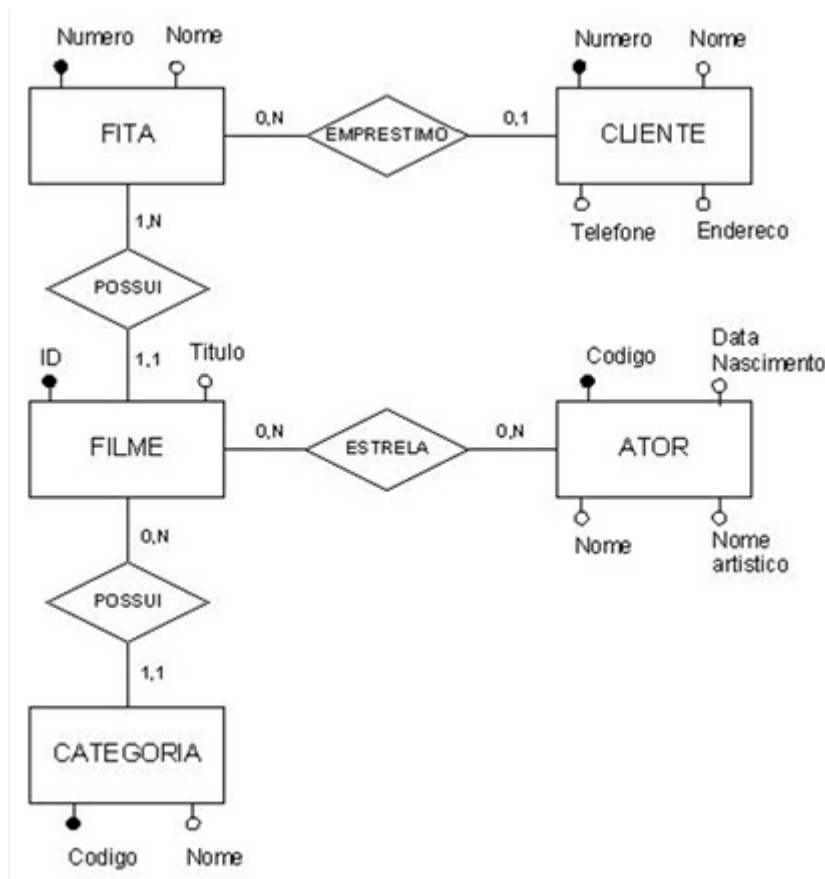
Implementação de Restrições e Regras de Integridade:

- Implementar restrições e regras de integridade referencial para garantir que as relações entre tabelas sejam mantidas consistentes.

Ou seja, minimizar a redundância de dados é crucial para manter a eficiência, a consistência e a integridade em um sistema de gerenciamento de banco de dados. A normalização adequada e o uso de **técnicas de modelagem** eficazes são fundamentais para lidar com esse desafio.

O que é a modelagem de um banco de dados e o porquê dela ser tão necessária ao se trabalhar com sistemas de gerenciamento de banco de dados?

A modelagem de um banco de dados é o processo de criar uma representação estruturada e organizada dos dados que serão armazenados e manipulados em um sistema de gerenciamento de banco de dados (SGBD). Essa representação é essencial para definir como os dados se relacionam entre si e como serão armazenados, permitindo a eficiente recuperação, atualização e manipulação das informações. De início começamos com o entendimento de **modelagem conceitual** que tem por objetivo representar as informações de forma independente de qualquer SGBD específico, focando nos conceitos e nas relações entre eles.



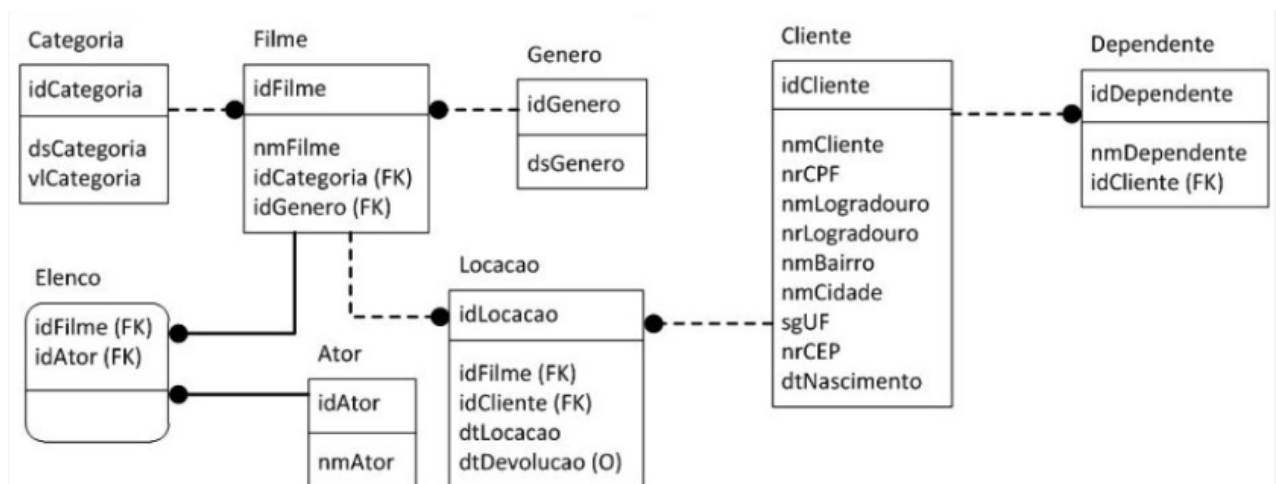
A modelagem de dados deve ser iniciada a partir do conhecimento do mundo real, é a etapa de análise e coleta de dados sobre o problema a ser resolvido . O **modelo conceitual**, que é o primeiro passo a ser desenvolvido, pois independe de tecnologia e serve como base para um projeto de desenvolvimento de software.

Entidade: Tipo de dado a ser tratado ex: (cliente, fita, filme, ator, categoria).

Relacionamento: Indica relação entre entidades ex: (empresta, possui, estrela).

Cardinalidade mínima e máxima: Indica quantos tipos de dados podem aparecer em um relacionamento.

Atributo: corresponde a alguma propriedade de interesse que ajuda a descrever uma entidade, ex: (número, nome, telefone e endereço de um funcionário).



A **modelagem lógica** de banco de dados é uma fase intermediária no processo de desenvolvimento de um sistema de banco de dados, situada entre a modelagem conceitual e a modelagem física. Nesta fase, o objetivo é traduzir o modelo conceitual, que descreve as entidades e seus relacionamentos de maneira independente do sistema de gerenciamento de banco de dados (SGBD), em um modelo lógico que possa ser implementado em um ambiente específico. Veja as principais características da modelagem lógica de banco de dados:

Tradução do Modelo Conceitual:

- A modelagem lógica representa uma tradução direta do modelo conceitual para conceitos mais próximos à implementação em um SGBD específico.
- Os conceitos abstratos, como entidades e relacionamentos, são convertidos em estruturas mais tangíveis, como tabelas, chaves primárias e estrangeiras.

Definição de Tabelas e Atributos:

- As entidades do modelo conceitual se tornam tabelas, e os atributos associados a essas entidades se tornam os campos dessas tabelas.
- As chaves primárias são identificadas para garantir unicidade nas linhas de cada tabela.

Relacionamentos e Chaves Estrangeiras:

- Os relacionamentos entre entidades são traduzidos em chaves estrangeiras nas tabelas, estabelecendo assim as relações entre elas.
- A integridade referencial é aplicada para garantir que as relações sejam mantidas consistentes.

Normalização:

- A normalização é aplicada para organizar os dados de maneira eficiente, removendo redundâncias e dependências funcionais indesejadas.
- Técnicas como a primeira, segunda e terceira formas normais são frequentemente aplicadas durante a modelagem lógica.

Índices e Restrições:

- Índices são identificados e definidos para otimizar a recuperação de dados, acelerando as consultas.
- Restrições de integridade, como regras de validação e triggers, podem ser especificadas para garantir que os dados estejam de acordo com as regras do modelo.

Tipos de Dados e Limitações:

- Os tipos de dados são atribuídos aos campos de acordo com as necessidades do sistema.
- Limitações, como restrições de tamanho e formatos específicos, são aplicadas aos campos para garantir a consistência e a validade dos dados.

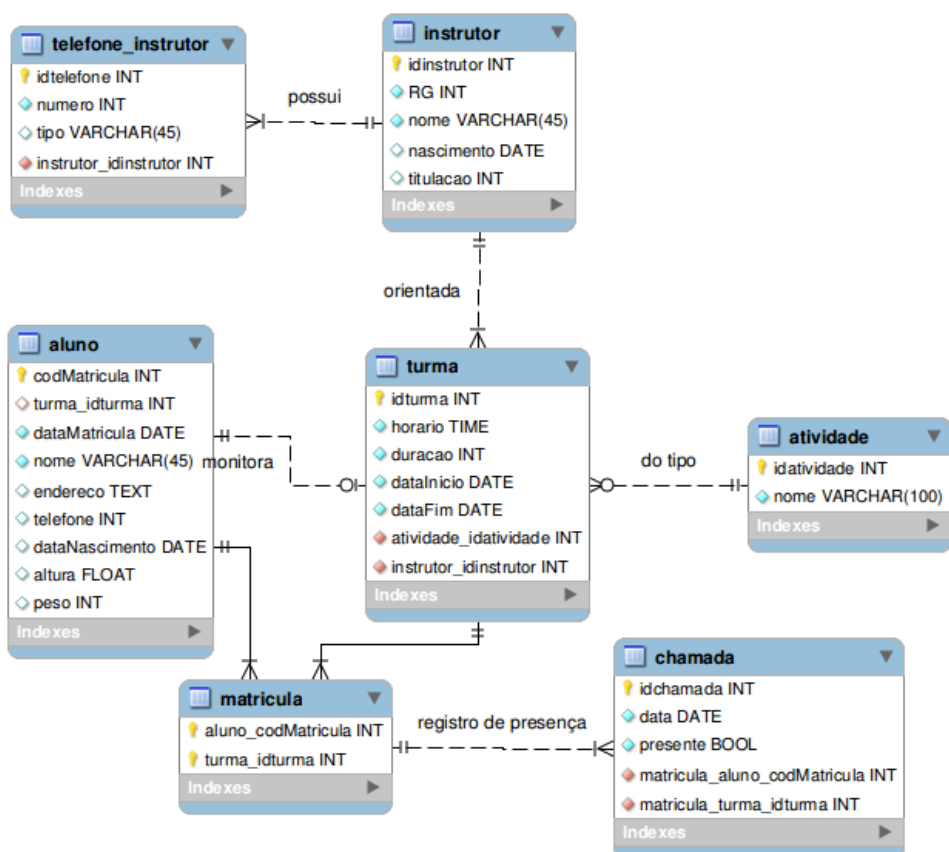
Documentação Detalhada:

- A modelagem lógica geralmente inclui uma documentação detalhada que descreve cada tabela, campo, relacionamento, índice e restrição.
- Isso facilita a compreensão do modelo por parte dos desenvolvedores e administradores de banco de dados.

Ferramentas de Modelagem:

- Ferramentas específicas de modelagem, como o MySQL Workbench, Microsoft Visio, ou ERwin, são frequentemente utilizadas para auxiliar na criação e visualização do modelo lógico.

A modelagem lógica serve como base para a implementação física do banco de dados e fornece uma visão mais detalhada e específica de como os dados serão armazenados e relacionados dentro de um SGBD específico.



A **modelagem física** de banco de dados é a fase do projeto em que o modelo de dados resultante da modelagem lógica é convertido em uma estrutura específica e otimizada para o Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido. Essa etapa envolve decisões detalhadas sobre como os dados serão armazenados fisicamente em termos de tabelas, índices, espaços de armazenamento e outras configurações específicas do SGBD. Aqui estão alguns aspectos importantes da modelagem física:

Escolha do SGBD:

- A primeira decisão crucial é escolher o SGBD que será usado para implementar o banco de dados. Cada SGBD pode ter suas próprias características, sintaxe SQL específica e otimizações.

Mapeamento de Tabelas:

- Cada tabela definida na modelagem lógica é mapeada para uma estrutura de tabela específica no SGBD, levando em consideração tipos de dados, tamanhos e restrições.

Índices e Otimização:

- São definidos índices para acelerar operações de consulta e manipulação de dados. A escolha de quais colunas indexar e que tipo de índices utilizar é uma parte crítica da otimização de desempenho.

Particionamento de Tabelas:

- Se necessário para grandes conjuntos de dados, as tabelas podem ser particionadas com base em critérios específicos, como intervalos de valores ou regiões geográficas.

Configurações de Armazenamento:

- Definição de configurações específicas do SGBD para o armazenamento físico dos dados, como a alocação de espaço em disco, tamanho de páginas, entre outros.

Considerações de Desempenho:

- Ajuste de parâmetros de configuração para otimizar o desempenho do banco de dados, como cache, memória disponível e paralelismo.

Segurança e Controle de Acesso:

- Implementação de medidas de segurança, como permissões de acesso e autenticação, para garantir a integridade e a confidencialidade dos dados.

Backup e Recuperação:

- Desenvolvimento de estratégias de backup e recuperação para garantir a segurança dos dados em caso de falhas ou perda de dados.

Fragmentação e Compactação:

- Otimização do armazenamento físico para minimizar a fragmentação e compactar dados, melhorando a eficiência do espaço em disco.

Alocação de Recursos:

- Alocação de recursos do sistema, como CPU e memória, para garantir que o banco de dados tenha o desempenho adequado.

Implementação de Procedimentos Armazenados e Gatilhos:

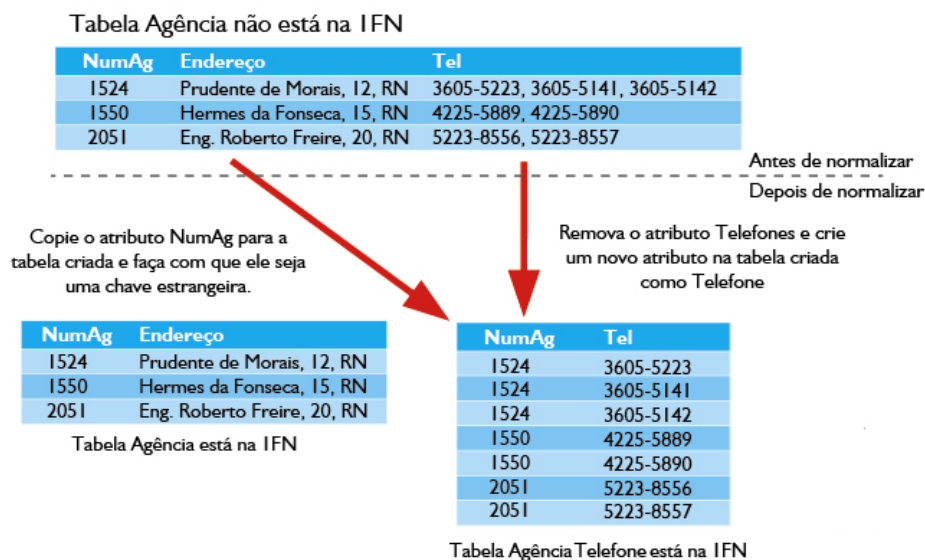
- Se já não tiver sido feito durante a modelagem lógica, implementação de procedimentos armazenados e gatilhos para automatizar certas operações e garantir a consistência dos dados.

A modelagem física é essencial para garantir que o banco de dados seja eficiente, seguro e capaz de lidar com as demandas específicas do ambiente em que será implantado. Cada SGBD pode ter suas próprias ferramentas e interfaces para auxiliar nesse processo, e as decisões tomadas nesta fase terão um impacto direto no desempenho e na escalabilidade do sistema.

Mas como a normalização pode me ajudar?

A **normalização de banco de dados** é um processo utilizado no design de sistemas de gerenciamento de banco de dados (SGBD) para organizar dados de maneira eficiente e reduzir a redundância. O objetivo principal da normalização é evitar problemas de inconsistência e redundância de dados, melhorando a integridade e a eficiência do banco de dados.

Existem diferentes formas normais (1NF, 2NF, 3NF, BCNF, etc.), cada uma delas com regras específicas para organizar os dados de maneira mais eficiente. Vamos abordar algumas das formas normais mais comuns:



Primeira Forma Normal (1NF):

- Cada coluna em uma tabela deve conter apenas valores atômicos (indivisíveis).
- Cada célula deve conter um único valor, e não uma lista de valores.

Alocacao	
NumEmp	INT
NumAg	VARCHAR(45)
AgEnd	VARCHAR(45)
Nom eEmp	VARCHAR(45)
horasSem	VARCHAR(45)
Indexes	

NumEmp	NumAg	AgEnd	NomeEmp	horasSem
12	1524	Prudente de Moraes, 12, RN	Luiz	20
12	1550	Hermes da Fonseca, 15, RN	Luiz	18
15	1524	Prudente de Moraes, 12, RN	Ricardo	10
15	1550	Hermes da Fonseca, 15, RN	Ricardo	15

Chave primária composta

Os valores de AgEnd podem ser obtidos apenas a partir do atributo Numag.

Os valores de NomeEmp podem ser obtidos apenas a partir do atributo NumEmp.

Os valores de horasSem podem ser obtidos apenas a partir do atributo NumEmp e NumAg.

Segunda Forma Normal (2NF):

- Deve estar em 1NF.
- Todos os atributos não chave devem ser totalmente dependentes da chave primária.

Funcionário não está na 3FN

FuncN	Nome	Cargo	Salário	NumAg	Endereço
25	Luiz	Caixa	2000	1632	Prudente de Moraes, 15
30	Ricardo	Gerente	5000	1668	Hermes da Fonseca, 20
31	Josita	Caixa	2000	1632	Prudente de Moraes, 15
32	Francisca	Gerente	5600	1632	Prudente de Moraes, 15
33	Andréia	Caixa	2300	1668	Hermes da Fonseca, 20

O atributo NumAg é definido como chave primária na tabela Agência

Mova o atributo Endereço para a nova tabela

NumAg	Endereço
1632	Prudente de Moraes, 15
1668	Hermes da Fonseca, 20
1632	Prudente de Moraes, 15
1632	Prudente de Moraes, 15
1668	Hermes da Fonseca, 20

Agência na 3FN

Funcionário na 3FN

FuncN	Nome	Cargo	Salário	NumAg
25	Luiz	Caixa	2000	1632
30	Ricardo	Gerente	5000	1668
31	Josita	Caixa	2000	1632
32	Francisca	Gerente	5600	1632
33	Andréia	Caixa	2300	1668

O atributo NumAg torna-se uma chave estrangeira na tabela Funcionário

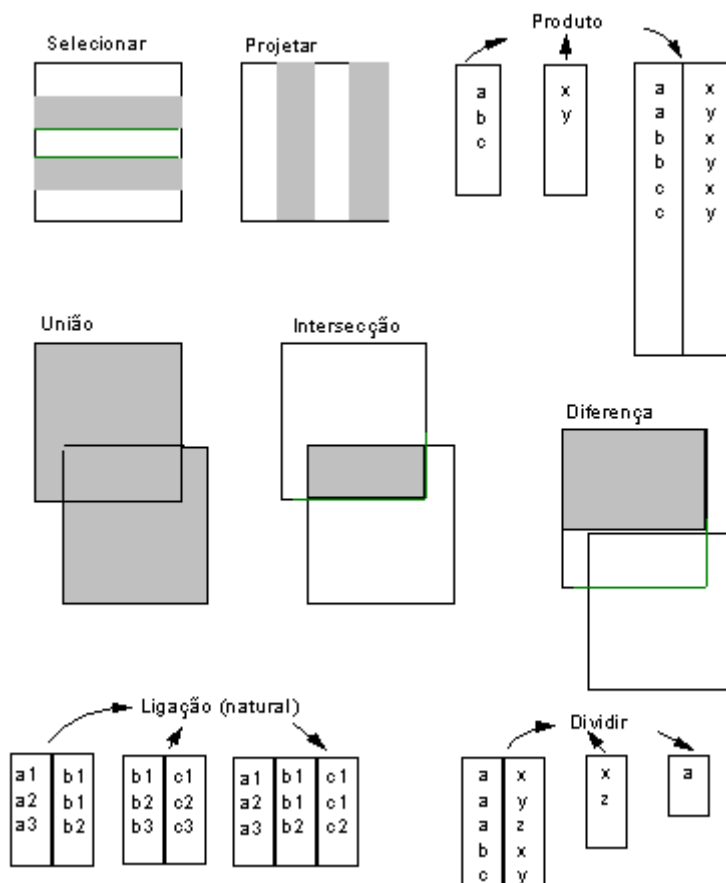
Terceira Forma Normal (3NF):

- Deve estar em 2NF.
- Todos os atributos não chave devem ser transitivamente dependentes da chave primária.

Cada nível de normalização visa eliminar diferentes tipos de dependências e redundâncias nos dados, garantindo que as atualizações, inserções e exclusões no banco de dados sejam feitas de forma consistente e evitando anomalias, como a anomalia de inserção, atualização e exclusão. A normalização, no entanto, não é uma abordagem única e universal para todos os casos. Às vezes, é necessário fazer compromissos e considerar as necessidades específicas do sistema. Em alguns casos, desnormalizar parcialmente o banco de dados pode ser necessário para otimizar o desempenho em consultas específicas.

Após saber sobre essa parte introdutória de como funciona um sistema de gerenciamento de banco de dados, como dou início em atividades práticas?

A **álgebra relacional** é um conjunto de operações matemáticas definidas sobre relações, que são estruturas de dados tabulares usadas em bancos de dados relacionais. Essas operações são fundamentais para manipular dados em sistemas de gerenciamento de banco de dados relacionais (SGBDR). A álgebra relacional fornece uma maneira formal e teórica de descrever consultas e operações em bancos de dados.



Seleção (σ): Esta operação é usada para extrair as linhas que atendem a uma condição específica de uma relação. A notação $\sigma_{\text{condição}}(R)$ representa a seleção dos tuples de R que satisfaçam a condição.

Exemplo: $\sigma_{\text{idade} > 25}(\text{Pessoas})$ - Seleciona todas as pessoas com idade superior a 25 anos.

Projeção (π): A operação de projeção é usada para selecionar colunas específicas de uma relação. A notação $\pi_{\langle \text{col1, col2, ...} \rangle}(R)$ representa a projeção de R nas colunas especificadas.

Exemplo: $\pi_{\langle \text{Nome, Idade} \rangle}(\text{Pessoas})$ - Projeta apenas as colunas Nome e Idade da relação Pessoas.

União (\cup): A união de duas relações R e S (denotada como $R \cup S$) retorna uma relação que contém todas as tuplas de R e S, sem duplicatas.

Interseção (\cap): A interseção de duas relações R e S (denotada como $R \cap S$) retorna uma relação que contém apenas as tuplas que estão presentes tanto em R quanto em S.

Diferença ($-$): A diferença entre duas relações R e S (denotada como $R - S$) retorna uma relação que contém todas as tuplas de R que não estão presentes em S.

Produto Cartesiano (\times): O produto cartesiano de duas relações R e S (denotado como $R \times S$) retorna uma relação que contém todas as combinações possíveis de tuplas de R e S.

Exemplo: $\text{Pessoas} \times \text{Departamentos}$ - Todas as combinações possíveis de pessoas e departamentos.

Junção (\bowtie): A junção de duas relações R e S (denotada como $R \bowtie S$) combina as tuplas de R e S com base em uma condição de junção específica.

Exemplo: $\text{Pessoas} \bowtie_{\langle \text{DepartamentoID} \rangle}$ - Junta pessoas e departamentos onde o ID do departamento é igual.

Finalizando essa introdução, a álgebra relacional fornece uma base teórica sólida para consultas em bancos de dados relacionais. Linguagens de consulta como SQL são inspiradas na álgebra relacional, e os SGBD's utilizam esses conceitos para otimizar e executar consultas eficientemente.