

Cálculo Relacional

Para falarmos de cálculo relacional, é necessário falar também de álgebra relacional. Álgebra relacional é o conjunto básico de operações para o modelo relacional, são operações que permitem a um usuário especificar as solicitações básicas de recuperação.

Uma expressão de cálculo especifica o que será recuperado, em vez de como recuperá-lo. Desse modo, o cálculo relacional é considerado uma linguagem não procedural. Difere da álgebra relacional, em que precisamos escrever uma sequência de operações para especificar um requisito de recuperação, assim ela pode ser considerada um caminho procedural para determinar uma consulta.

As operações de álgebra produzem, novas relações, que podem ser manipuladas, adiante, usando-se as operações da mesma álgebra. Uma sequência de operações de álgebra relacional forma uma expressão de álgebra relacional cujos resultados também serão uma relação que representa o resultado de uma consulta de banco de dados (ou solicitação de recuperação). Enquanto a álgebra define um conjunto de operações para o modelo relacional, o cálculo relacional provê uma notação declarativa de nível superior para a especificação de consultas relacionais.

A álgebra relacional freqüentemente é considerada como uma parte do modelo relacional de dados e suas operações podem ser divididas em dois grupos. Um grupo inclui um conjunto de operações da teoria de conjunto matemática — essas operações são aplicadas porque cada relação é definida como um conjunto de tuplas no modelo relacional formal

Os conjuntos de operações incluem UNIÃO (UNION), INTERSEÇÃO (INTERSECTION), DIFERENÇA DE CONJUNTO (SET DIFFERENCE) e PRODUTO CARTESIANO (CROSS PRODUCT).

O outro grupo consiste em operações desenvolvidas especificamente para os bancos de dados relacionais — estas incluem SELEÇÃO (SELECT), PROJEÇÃO (PROJECT) e JUNÇÃO (JOIN), entre outras.

Uma expressão de cálculo relacional cria uma nova relação, que é especificada em termos de variáveis que abrangem as linhas das relações armazenadas no banco de dados (em cálculos de tuplas) ou as colunas das relações armazenadas (em cálculo de domínio).

Cálculos relacional de Tuplas e Cálculo relacional de Domínio

Há dois tipos de cálculo relacional: (1) o cálculo relacional de tupla, que usa as variáveis de tupla que abrangem as tuplas (linhas) das relações e (2) o cálculo relacional de domínio, que usa as variáveis de domínio que englobam os domínios (colunas das relações).

Tupla:

O cálculo relacional de tupla está baseado na especificação de um número de variáveis de tupla. Cada variável de tupla geralmente abrange uma particular relação do banco de dados,

significando que a variável pode tomar, como seu valor, qualquer tupla individual daquela relação.

Exemplo:

Recupere a data de nascimento e o endereço do empregado (ou empregados) cujo nome seja 'John B. Smith'.

QO: {t.DATANASC, I t.ENDEREÇO I EMPREGADO(t) AND t.PNOME='John' AND t.MINIT='B' AND t.UNOME='Smith'}

No cálculo relacional de tuplas, especificamos primeiro os atributos requisitados t.DATANASC e t.ENDEREÇO para cada tupla t selecionada. Depois, estabelecemos a condição para selecionar uma tupla após a barra (I) — a saber, seja t uma tupla da relação EMPREGADO cujos valores dos atributos PNOME, MINICIO e UNOME são 'John', 'B' e 'Smith', respectivamente.

Domínio

Há outro tipo de cálculo relacional chamado cálculo relacional de domínio ou, simplesmente, cálculo de domínio. Esse se difere do cálculo de tupla pelo tipo de variáveis usadas nas fórmulas: em vez de ter variáveis abrangendo as tuplas, as variáveis abrangem os valores únicos dos domínios dos atributos. Para formar uma relação de *grau* n , o resultado de uma consulta, devemos ter n dessas variáveis de domínio — uma para cada atributo.

Exemplo:

Recupere a data de nascimento e o endereço do empregado cujo nome seja 'John B. Smith'.

QO: {uv I (3 q) (3 r) (3 s) (3 t) (3 w) (3 x) (3 y) (3 Z)
(iMPKGADo(qrstuvwxyz) AND <2='JOHN' AND r='B' AND S='SMITH')}

Precisamos de dez variáveis para a relação EMPREGADO, uma para abranger cada domínio de atributo, na ordem. Das dez variáveis q, r, s.....z, apenas uev são livres. Especificaremos, primeiro, os atributos solicitados, DATANASC e ENDEREÇO, pelas variáveis de domínio livres u para DATANASC e v para ENDEREÇO. Depois, especificaremos a condição para selecionar uma tupla, seguindo a barra (I) — assim, se a sequência de valores designados às variáveis qrstuvwxyz for uma tupla da relação EMPREGADO, os valores para q (PNOME), r (MINICIAL) e s (UNOME) seriam 'John', 'B' e 'Smith', respectivamente. Por conveniência, quantificaremos apenas aquelas variáveis que aparecem, de fato, em uma condição (essas poderiam ser q, r e s em QO), no restante de nossos exemplos. Uma notação alternativa taquigráfica, usada em QBE, para escrever essa consulta, seria designar as constantes 'John', 'B' e 'Smith' diretamente como mostrado em QOA. Aqui, todas as variáveis que não aparecem à esquerda da barra são, implicitamente, quantificadas por existência. QOA: {uv I EMPREGAD0('John', 'B', 'Smith', t, u, v, w, x, y, z)}

Importância da álgebra relacional:

1. Provê um fundamento formal para operações do modelo relacional;

2. É usada como uma base para implementar e otimizar as consultas em sistemas de gerenciadores de banco de dados relacional (SGBDRs);
3. Alguns de seus conceitos são incorporados na linguagem de consulta-padrão SQL para os SGBDRs.

Conclusão:

O presente trabalho mostra de forma superficial os conteúdos necessários para o estudo e aprofundamento no conteúdo de banco de dados, sendo eles: cálculo relacional e álgebra relacional. De forma a exemplificar o uso dos termos e suas relações, onde álgebra se dá como o processo de busca de dados, e o cálculo, o resultado da busca.

Ainda dentro do tema cálculo relacional, esse se divide em duas formas: Tuplas e Domínios. Sendo elas formas diferentes de fazer a mesma coisa: consultar informações de forma precisa.

O conteúdo é mais técnico e extenso do que o apresentado, mas são conceitos conceituais que estão por trás das consultas realizadas na linguagem/ferramenta SQL no cotidiano. No fim, conhecer essas bases ajuda não só a entender melhor como os bancos funcionam, mas também a criar consultas mais funcionais.

Referências:

Elmasri, Ramez e Navathe, Shamkant B. **Sistemas de banco de dados**. revisor técnico Luis Ricardo de Figueiredo. -São Paulo : Pearson Addison Wesley, 2005. Disponível em: <https://tonysoftwares.com.br/attachments/article/5297/Sistema_de_banco_de_dados_Navathe.pdf>. Acesso em: 16/04/2025