Terceiro Resumo - Banco de Dados I

🧒 **Aluno** - Fabio Henrique Alves Fernandes | 19.1.4128

🎓 **Professor Doutor** - Guilherme Tavares De Assis

🏠 **Instituição** - Universidade Federal de Ouro Preto | Departamento de Computação

📅 **Data De Entrega** - 2 de dezembro de 2021

Álgebra Relacional

A álgebra relacional traz uma um conjunto de operações do modelo relacional, possibilitando solicitar recuperações de dados, sendo que o resultado de cada recuperação é uma nova relação, formada por uma ou mais relações.

As operações de recuperação podem ser operações específicas de banco de dados relacionais, ou operações da teoria de conjuntos.

Operação Seleção

A operação de seleção é utilizada para selecionar um conjunto de tuplas de uma seleção.

σ<cond>(<R>), onde <cond> é uma condição de seleção e <R> é o nome da relação.

Essa operação só pode ser feita a partir de uma única relação. O numero de atributos da relação resultante é sempre o mesmo da relação original. Ela também é comutativa, e podemos combinar uma cascata de seleções em uma única operação de seleção.

Operação Projeção

A operação de projeção é usada para selecionar dentre uma relação, um conjunto de atributos.

π<atributos>(<R>), onde <atributos> é uma lista de atributos
dentre os atributos da relação R e <R> é o
nome de uma relação.

A projeção também é uma operação que é feita com uma única relação, assim como a de seleção. Caso os atributos presentes não sejam chaves da relação R, é possível a ocorrência de tuplas duplicadas. Sendo assim, a projeção remove todas as tuplas duplicadas, tornando o resultado válido, tornando o número de tuplas na relação resultante menor ou igual ao número de tuplas de R. Diferente da seleção, a projeção não é uma operação comutativa, então quando fazemos **π<lista1>(π <lista2>(<R>))** o resultado retornado será **π<lista1>(<R>)**

Sequência de Operações

Dentro da álgebra relacional é muito comum aplicarmos diversas operações em sequência, sejam como uma única expressão, ou aplicando uma de cada vez, criando relações intermediárias.

Dentro da técnica de sequencia de operações, podemos renomear os atributos das relações intermediárias e de resultado, bastando listar o nome dos novos atributos entre parênteses, juntamente do nome das novas relações.

Caso nenhuma renomeação seja aplicada em uma seleção, os nomes dos atributos na relação resultante são os mesmos da relação original. Já em uma projeção sem renomeação, a relação resultante possui os mesmos nomes dos atributos especificados na lista de projeção e aparecem na mesma ordem listada.

$$\text{Dep5\_Emps} \leftarrow \sigma_{\text{NumDepto} = 5} (\text{Empregado})$$
$$\text{Resultado (PNome, UNome, Sal)} \leftarrow \pi_{\text{PrimeiroNome, UltimoNome, Salario}} (\text{Dep5\_Emps})$$

Operação Renomeação

A operação de renomeação é usada geralmente para renomear uma relação ou até mesmo seus atributos.

ρS(b1,b2,...,bn)(<R>) ou **ρS(<R>)** ou **ρ(b1,b2,...,bn)(<R>),** onde <S> é o novo nome da relação, <b1, b2, ..., bn> são os novos nomes dos atributos e <R> é a relação original. A primeira expressão renomeia tanto a relação quanto os atributos, a segunda renomeia apenas a relação e a terceira renomeia apenas os atributos.

Operações Teóricas de Conjuntos

Dentro da álgebra relacional temos um grupo padrão de operações matemáticas sobre conjuntos. As operações só envolvem duas relações, e, para algumas operações, as relações devem possuir o mesmo número de tuplas para serem consideradas aptas.

Dizemos que duas relações (R e S) são compatíveis para união se possuírem o mesmo grau 'n' e se os domínios de cada atributos forem iguais.

As operações teóricas de conjuntos que exigem relações compatíveis para união são:

- União: R ∪ S gera uma relação que inclui todas as tuplas que estão em R ou em S ou em ambas;
- Interseção: R ∩ S gera uma relação que inclui todas as tuplas que estão tanto em R quanto em S;
- Diferença: R - S gera uma relação que inclui todas as tuplas que estão em R, mas não estão em S.

A relação resultante das operações possui os mesmos nomes de atributos da primeira relação envolvida nas operações. As operações de união e interseção são comutativas e associativas.

$$R \cup S = S \cup R \quad e \quad R \cap S = S \cap R$$
$$R \cup (S \cup T) = (R \cup S) \cup T \quad e \quad (R \cap S) \cap T = R \cap (S \cap T)$$

A operação de conjunto binária Produto Cartesiano, representada por 'X', é utilizada para combinar tuplas de suas relações de forma combinatória, onde as relações não precisam ser compatíveis para união.

O resultado de R X S é uma relação Q, onde os número atributos de Q é a soma do número de atributos de R com os de S. A relação Q possui uma tupla para cada combinação de tuplas das relações envolvidas. Então, se R possui um número n de tuplas e S possui um numero m de tuplas, então Q possuirá n * m tuplas.

Por ser uma operação que geram muitas tuplas espúrias, ela não é comumente usada, sendo prática quando é seguida de uma seleção que combina valores de atributos nas relações

envolvidas.

Operação Junção

Uma vez que o produto cartesiano seguido de uma seleção é utilizado com muita frequência, foi criada uma operação especial chamada Junção, para especificar tal sequência como uma única operação. Ela é usada para combinar tuplas relacionadas de duas relações em uma única tupla.

$$R \bowtie_{<cond>} S$$

onde R e S são relações e <cond> é uma condição de junção entre as relações.

Assim como em um produto cartesiano, uma junção entre R e S gera uma relação Q com a soma dos atributos de R com S. Q possui uma tupla para cada combinação de tuplas das relações envolvidas, sempre que a combinação satisfazer a condição imposta na junção.

A operação junção mais comum, chamada Equijunção, envolve apenas condições de junção com comparações de igualdade. Uma equijunção onde dois atributos da comparação têm o memo nome é chamada Junção Natural, sendo denotada por *. Neste caso, apenas um dos atributos da comparação aparece na relação resultante e a condição de junção não é especificada.

Operação Divisão

A operação binária de divisão, representada por ÷, é utilizada para um tipo especial de consulta que ocorre, algumas vezes, em aplicações de banco de dados. A operação Divisão R(Z) ÷ S(X) só pode ser aplicada se $X \subseteq Z$. O resultado dessa divisão é uma relação T que contém o conjunto de atributos de R que não são atributos de S, ou seja, os atributos Z-X. Uma tupla de T é formada por valores dos atributos Z-X de R, cujos valores referentes dos atributos X de R combinaram com todos os valores dos atributos X de S.

Funções de Agregação e Agrupamento

Uma solicitação que pode se expressa na álgebra relacional é a aplicação de funções matemáticas de agregação em coleção de valores do banco de dados. A mais comuns dessas funçções são as de soma (Sum), média (Average), valor máximo (Maximum), valor mínimo (Minimum) e contador de tuplas (Count).

Outra solicitação envolve o agrupamento de tuplas de uma relação por meio dos valores de alguns atributos e depois a aplicação de uma função de agregação em cada grupo.

Uma função de agrupamento é definida como <atributos de agrupamento> $\Im$ <funções de agregação> (<R>) onde <R> é uma relação, <atributos de agrupamento> é uma lista de atributos de R responsável pelo agrupamento e <funções de agregação> é uma lista de pares de (<função><atributo>): em cada um destes pares, <função> é uma das funções de agregação permitidas e <atributo> é um atributo de R cuja função de agregação será aplicada.

A relação resultante possui os atributos de agrupamento e os resultados gerados pelas funções de agregação. Haverá sempre uma tupla para cada grupo gerado pelos atributos de agrupamento.

Caso não se aplique nenhuma renomeação, os atributos da relação resultante correspondentes às funções de agregação são, cada um deles, a concatenação do nome da função e o nome do atributo (<função>_<atributo>).

Fechamento Recursivo

É um tipo de operação que é aplicada a um auto-relacionamento entre tuplas do mesmo tipo.

Operações de Junção Externa

As operações de Junção Externa são extensões da junção. Elas são utilizadas quando desejamos manter todas as tuplas da relação R, ou de S, ou de ambas, no resultado da junção, caso elas possuam ou não tuplas combinantes.

Elas se classificam como:

Junção Externa à Esquerda

Denotada por:

Essa junção mantém as tuplas da relação R. Se não houver tupla em S que combine, os atributos de S são preenchidos com valores nulos.

Junção Externa à Direita

Denotada por:

Essa junção mantém as tuplas da relação S. Se não houver tupla em R que combine, os atributos de R são preenchidos com valores nulos.

Junção Externa Completa

Denotada por:

Essa junção mantém todas as tuplas de ambas as relações, Se alguma tupla não combina, os atributos são completados com valores nulos

A união externa serve para realizar a união entre tuplas de duas relações, caso as mesmas não sejam compatíveis para união. A operação vai encontrar a união entre tuplas de duas relações que são parcialmente compatíveis, ou seja, apenas alguns atributos são compatíveis para a união. Já os atributos que não são compatíveis, são mantidos na relação resultante e, caso não possuem valores para uma determinada tupla, seus valores serão nulos.

SQL

SQL é uma linguagem de consulta que implementa as operações da álgebra relacional de forma bem amigável. Além das consultas, a SQL também permite a definição da estrutura de dados, a definição de restrições de integridade, a modificação dos dados no banco de dados, a especificação de restrições de segurança e controle de transações e a utilização em linguagens hospedeiras.

De forma geral, SQL utiliza dos termos tabela, linha e coluna para relação, tupla e atributo, respectivamente.

Esquema e Catálogo

Esquema

É uma forma de agrupar tabelas e outros componentes pertencentes a uma mesma aplicação de banco de dados. Um esquema é definido por um nome e inclui um identificador de autorização para indicar o usuário que é dono do esquema.

Catálogo

É uma coleção de esquemas num ambiente SQL. Esquemas de um mesmo catálogo podem compartilhar certos elementos como definições de domínio, por exemplo.

Comando CREATE TABLE

É o comando usado para especificar uma nova relação, fornecendo um nome e informando os seus atributos e suas restrições. Inicialmente, os atributos são especificados, informando o nome, o tipo de dado e qualquer restrição para o atributo, como, por exemplo, NOT NULL. Depois, são especificadas as restrições (CONSTRAINT) de chave (PRIMARY KEY, UNIQUE) e de integridade referencial (FOREIGN KEY). Essas restrições podem ser especificadas também no comando ALTER TABLE, que permite a alteração da definição de uma relação.

```
CREATE TABLE  Empregado
(    PrimeiroNome     VARCHAR(15)     NOT NULL,
     InicialMeio      CHAR,
     UltimoNome       VARCHAR(15)     NOT NULL,
     NumEmpregado     CHAR(9)         NOT NULL,
     DataNascimento   DATE,
     Endereco         VARCHAR(30),
     Sexo             CHAR,
     Salario          DECIMAL(10,2),
     NumSupervisor    CHAR(9),
     NumDepto         INT             NOT NULL,
   CONSTRAINT PK_Emp PRIMARY KEY (NumEmpregado),
   CONSTRAINT FK_NumSup FOREIGN KEY (NumSupervisor)
     REFERENCES  Empregado (NumEmpregado),
   CONSTRAINT FK_EmpDep FOREIGN KEY (NumDepto)
     REFERENCES  Departamento (NumDepto)
);
```

Exemplo de criação de tabela

Tipos de dados numéricos

- Inteiro: integer ou int, smallint;
- Real: float, real, double precision;

- Números formatados: decimal(i, j) ou numeric (i, j) ou dec (i, j), onde i é número de dígitos a esquerda e j é o número de dígitos a direita do ponto.

Tipos de dados alfanuméricos

- Cadeia de caracteres de tamanho fixo: char (n) ou character (n);
- Cadeia de caracteres de tamanho variável: varchar (n) ou char varying (n) ou character varying (n), onde n é a quantidade máxima de caracteres.

Tipo data

- Date (AAAA-MM-DD)

Tipo hora

- Time (HH:MM:SS)

Podemos também criar um domínio, usando o nome do domínio como um tipo de dado de algum atributo usando CREATE DOMAIN.

Como a linguagem SQL permite valores nulos, podemos especificar uma restrição de NOT NULL se o valor nulo não puder ser definido para um determinado atributo. Para definir um valor *default* para determinado atributo, usando do DEFAULT <valor> na definição do atributo. Esse valor padrão é incluído na nova tupla quando um valor explícito não for fornecido para o atributo.

Podemos especificar também qua ação será tomada se alguma restrição de integridade referencial for violada, seja pela exclusão de uma tupla ou a atualização de alguma chave primária. As opções podem ser bloqueio (*default*), propagação (CASCADE), substituição por nulo (SET NULL) e substituição pelo valor padrão (SET DEFAULT) que devem ser especificadas com a cláusula ON DELETE (em uma operação de exclusão) ou ON UPDATE (em uma operação de atualização), em cada restrição de integridade referencial.

Comando DROP SCHEMA

Para remover um esquema de um banco de dados, temos o comando DROP SCHEMA.

DROP SCHEMA <E> <opção>

Onde <E> é o nome do esquema a ser removido e <opção> pode ser RESTRICT (não elimina o esquema se houver algum elemento nele) ou CASCADE (elimina o esquema e todos os seus elementos).

Comando DROP TABLE

Para remover uma relação de um banco de dados, temos o comando DROP TABLE.

DROP TABLE <R> <opção>

Onde <R> é o nome da relação a ser removida e <opção> pode ser RESTRICT (não elimina a relação se houver alguma restrição ou visão associada a ela) ou CASCADE (elimina a relação e todas as restrições e visões associadas a ela).

Comando ALTER TABLE

O comando ALTER TABLE pode ser usado tanto para adicionar quanto para remover atributos e restrições de uma determinada relação, e também para alterar a definição de determinado atributo.

- ALTER TABLE <R> ADD <A> <D>: adiciona o atributo <A> , cujo domínio é <D>, na relação existente <R>. Se já existirem tuplas na relação <R>, o novo atributo receberá valores nulos para essas tuplas.
- ALTER TABLE <R> DROP <A> <opção> : remove o atributo <A> da relação existente <R>. A <opção> pode ser RESTRICT (não elimina o atributo se houver alguma restrição ou visão referenciando-o) ou CASCADE (elimina o atributo, as visões e as restrições que o referenciam).
- ALTER TABLE <R> ALTER <A> DROP DEFAULT : remove a cláusula de default referente ao atributo <A> da relação existente <R>.
- ALTER TABLE <R> ALTER <A> SET DEFAULT <V>: adiciona uma cláusula de default, referente ao atributo <A> da relação existente <R>, com o valor <V>.
- ALTER TABLE <R> DROP CONSTRAINT <C>: remove a restrição <C> referente à relação existente <R>.
- ALTER TABLE <R> ADD CONSTRAINT <C>: adiciona a restrição <C> na relação existente <R>.

Consultas Básicas em SQL

A estrutura básica de uma pesquisa em SQL é geralmente:

SELECT $<A_1>, <A_2>, \ldots, <A_n>$	{projeção}
FROM $<R_1>, <R_2>, \ldots, <R_m>$	{produto cartesiano}
WHERE <cond>;	{seleção}

Onde cada <Ai> representa um atributo, cada <Rj> representa uma relação e <cond> representa uma condição de seleção como uma expressão lógica. O comando WHERE pode ser omitido, caso não tenha nenhuma condição de seleção. Caso quisermos todos os atributos na consulta de uma determinada relação, usamos um * no lugar da lista de atributos no comando SELECT.

Palavras-chave DISTINCT e ALL

A linguagem SQL permite duas tuplas idênticas dentro de uma mesma relação, já que uma relação em SQL não é um conjunto de tuplas. Para remover as tuplas duplicadas em uma consulta, usamos a palavra DISTINCT no comando SELECT. Para especificar que as tuplas duplicadas não devem ser removidas, devemos usar a palavra ALL.

Operador LIKE

Também temos o operador de comparação LIKE, que permite condições de comparação em partes de uma cadeia de caracteres. O caractere '%' substitui um número arbitrário de caracteres. Já o caractere "_" substitui um único caractere.

Operador BETWEEN

A partir do operador BETWEEN, conseguimos especificar um intervalo de valores para um atributo.

Valor NULL

SQL também permite consultas em que o valor retornado é NULL. Para isso, usamos os operadores IS ou IS NOT.

Renomeando Relações

Dentro de uma consulta, de forma a facilitar, podemos também "renomear" as relações, a partir do operador AS.

Podemos também "renomear" os nomes dos atributos.

Operadores Aritméticos

Nas consultas, também podemos usar das operações básicas da aritmética ('+', '-', '*' e '/').

Cláusula ORDER BY

Também podemos ordenar as tuplas resultantes de uma consulta pelos valores de um ou mais atributos, usando o ORDER BY.

Operações de Conjunto

Pensando na álgebra relacional, em SQL também podemos implementar as operações de conjunto.

- União (UNION);
- Interseção (INTERSECT);
- Diferença (EXCEPT).

Sempre que usamos a operação de união, as tuplas duplicadas são eliminadas automaticamente. Para que possamos mantê-las, podemos usar a operação UNION ALL.

Junção entre Tabelas

Dentro da cláusula FROM, podemos inserir as cláusulas JOIN (INNER JOIN) e ON. Podemos também usar as cláusulas de junção natural (NATURAL JOIN) e junção externa (OUTER JOIN), tanto à esquerda, (LEFT OUTER JOIN), quanto à direita (RIGHT OUTER JOIN) e também a junção externa completa (FULL OUTER JOIN).

Subconsultas

Uma subconsulta é um bloco completo que existe dentro da cláusula WHERE de uma outra consulta, chamada de consulta externa. Esse bloco é literalmente um novo bloco de pesquisa, com as mesmas cláusulas básicas de consulta.

Operadores SOME e ALL

Além do IN, temos outros operadores comparadores.

- <operador lógico> SOME (ou ANY);
- <operador lógico> ALL.

SOME e ALL precisam sempre de algum comparador lógico, seja ele =, <>, >, >=, < ou ≤. Quando temos o operador '= SOME', podemos dizer que ele é equivalente ao operador IN.

Conjuntos Explícitos de Valores

Podemos também usar de um conjunto explícito de valores na cláusula WHERE ao invés de fazer uma subconsulta.

Subconsultas Correlacionadas

Sempre que uma condição na cláusula WHERE de uma subconsulta fizer uma referência a algum atributo de uma relação declarada na consulta externa, dizemos que as duas consultas são correlacionadas.

Função EXISTS

A função EXISTS é utilizada para verificar se o resultado de uma subconsulta correlacionada é vazia ou não.

Agrupamento e Funções de Agregação

Assim como na álgebra relacional, temos em SQL conceitos de agrupamento e funções de agregação, sendo alguns deles:

- COUNT - retorna o número de tuplas recuperadas em uma consulta;
- SUM - retorna a soma dos valores de um atributo em uma consulta;
- MAX - retorna o valor máximo de um atributo em uma consulta;
- MIN - retorna o valor mínimo de um atributo em uma consulta;
- AVG - retorna a média dos valores de um atributo em uma consulta.

As funções de agregação podem ser usadas em uma cláusula SELECT ou em uma cláusula HAVING. Já a cláusula GROUP BY serve para que possamos agrupar tuplas que possuem um mesmo valor para os atributos relacionados em na cláusula. A HAVING pode ser usada em conjunto com o GROUP BY, onde devemos especificar uma condição de seleção a ser aplicada em cada grupo de tuplas recuperadas na consulta. Somente grupos que satisfazem a condição serão retornados.

Comando INSERT

O comando INSERT é usado geralmente para adicionar uma única tupla a uma relação. Devemos especificar o nome da relação e uma lista de valores para a tupla. Os valores da tupla devem estar relacionados na mesma ordem em que foram definidos no CREATE TABLE.

Uma outra forma de usar o comando INSERT permite especificar nomes explícitos de atributos que correspondem aos valores fornecidos no comando.

Uma variação do comando INSERT inclui múltiplas tuplas numa relação, conjuntamente com a criação da relação e a carga da mesma com o resultado de uma consulta.

Comando DELETE

O comando DELETE é o responsável por deletar tuplas de uma determinada relação. Ele possui uma cláusula WHERE, que seleciona as tuplas a serem deletadas. Quando não especificamos quais tuplas vão ser deletadas, todas as tuplas da relação serão excluídas, deixando somente a tabela vazia no banco de dados.

Todas as tuplas são explicitamente excluídas de uma só relação, entretanto, a exclusão pode se propagar para tuplas de outras relações de acordo com as restrições de integridade referencial definidas anteriormente.

Comando UPDATE

É geralmente usado para modificar valores de atributos de uma ou mais tuplas de determinada relação. Nela podemos incluir a cláusula WHERE, selecionando as tuplas a serem modificadas e a cláusula SET, especificando os atributos a serem modificados, além de seus novos valores. Modificar uma chave primaria pode causar uma propagação entre as chaves estrangeiras de outras relações, de acordo com as restrições de integridade referencial definidas. Lembrando que podemos também modificar diversas tuplas usando um único comando UPDATE.

Visões em SQL

Visão em SQL, nada mais é que uma relação única derivada de outras relações ou de outras visões já antes definidas. Uma visão não é necessariamente física dentro do banco, tornando as operações de atualização bem limitadas.

Para criar uma visão, usamos o comando CREATE VIEW, onde especificamos o nome da visão, uma lista de nomes de atributos e uma consulta que especifica o conteúdo da visão. Se nenhum dos atributos da visão resultar da aplicação de funções ou operações aritméticas, não tem necessidade de especificar os nomes de atributos para a visão, já que seriam os mesmos nomes de atributos das relações definidoras da visão.

Com uma visão já definida, podemos realizar diversas consultas a partir dela. Uma das principais vantagens de uma visão e poder simplificar a especificação de certas consultas. Visões também são utilizadas como mecanismos de segurança e autorização. Dessa forma, podemos criar níveis de acesso para cada tipo de usuário.