João Vitor Gonçalves da Silva – 18.1.4006

BCC 321 – Banco de Dados I

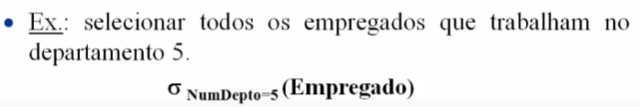
Resumo III – Álgebra Relacional e SQL

1. **Introdução a álgebra relacional**

A álgebra relacional diz respeito ao conjunto básico de operações realizadas dentro do modelo relacional para manipulação do banco de dados. As operações básicas de recuperação, feitas sobre uma ou mais relações, geram uma nova relação. Elas se dividem em:

* Operações especificas de banco de dados relacionais: seleção, projeção, junção, etc;
* Operações da teoria de conjuntos: união, interseção, diferença e produto cartesiano.
  1. **Operação de Seleção**

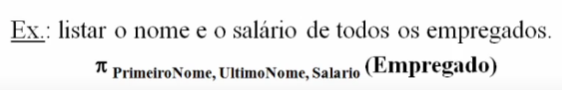
É utilizada para selecionar um conjunto de tuplas (linhas) de uma relação e é representado por: **σ <cond> (<R>),** onde “<cond>” é a condição que a tupla deve atender para ser retornada e “(<R>)” é a relação em que a busca é realizada. E uma relação unária pois é realizada apenas em uma relação por vez. O grau da relação resultante é o mesmo da relação em que a seleção foi feita. Exemplo:



Retorna todas as tuplas de Empregado no qual o numero do departamento é 5.

* 1. **Operação de Projeção**

É utilizada para selecionar um conjunto de atributos de uma relação e é representada por:  {\displaystyle \pi }**π <atributos>(<R>),** onde “<atributos>” é a lista de atributos que se deseja selecionar de uma relação e “(<R>)” é o nome da relação que a projeção é feita. Essa operação não retorna tuplas na qual os atributos selecionados são iguais. Também é uma operação unária, mas não é comutativa. Exemplo:



Retorna todos os valores presentes nos atributos PrimeiroNome, UltimoNome e Salario da relação Empregado.

* 1. **Sequência de Operações**

As operações descritas pela álgebra relacional podem ser utilizadas em conjunto para se obter buscas mais complexas e especificas. Ao utilizar operações em sequência, pode-se utilizar elas em uma única linha ou fazer cada operação em separado. Ao usar as operações separadamente, gera-se relações intermediarias na qual é atribuído um nome. Exemplo: “Listar nome e salário de todos os empregados que trabalham no departamento 5”;



Exemplo de operações em sequencia em uma única linha



Exemplo de operações em linhas separadas, onde “Dep5\_Emps” é uma relação intermediaria resultante da seleção sobre a relação Empregado. Então, é feito a projeção em cima da relação intermediária.

A sequência de operações também pode ser utilizada para renomear atributos das relações intermediárias. Para isso, basta incluir os novos nomes dos atributos entre parênteses na frente do nome da relação intermediária. Esses nomes não são alterados no banco físico. Exemplo:

Resultado (PNome, UNome, Sal) 🡨**π**PrimeiroNome,UltimoNome,Salario (Dep5\_Emps)

* 1. **Operação de Renomeação**

É utilizada para renomear atributos e relações, tendo três tipos:

* P s (b1,.., bn) (<R>): onde s é o novo nome da relação, (b1,..., bn) são os nomes dos atributos e (<R>) é a relação original. Essa expressão renomeia a relação e seus atributos.
* P s (<R>): Essa expressão renomeia apenas a relação R.
* P (b1, ..., bn) (<R>): Essa expressão renomeia apenas os atributos de R.

1. **Operação Teórica de Conjuntos**

Representa um grupo de operações matemáticas sobre conjuntos. São operações binárias pois são realizadas em duas relações. Para algumas relações, elas devem ser compatíveis para união, ou seja, possuírem mesmo grau e o domínio entre os atributos. São dividas em:

* União: R U S, que gera uma relação que inclui todas as tuplas de R, de S ou de ambas.
* Interseção: R ∩ S, que gera uma relação que inclui as tuplas que estão tanto em R quanto em S.
* Diferença: R – S, que gera uma relação que inclui as tuplas que estão em R, mas não estão em S.
* Produto Cartesiano: R(a1, ..., na) x S(b1, ..., bm), que gera uma relação com n+m atributos e possui uma tupla para cada combinação de tuplas das relações. As relações envolvidas nessa operação não precisam ser compatíveis para união. Como geram várias tuplas sem sentido, é utilizado junto de uma seleção para selecionar apenas as tuplas que fazem sentido.
* Junção: R |><|<cond> S, que realiza o mesmo processo do produto cartesiano, mas ele já seleciona as tuplas válidas por meio do condicional <cond>.

A relação resultante das operações de conjunto mantem os nomes dos atributos da primeira relação que está na operação, com exceção do produto cartesiano, já que este também inclui os atributos da segunda relação envolvida. Ambas operações de união e interseção são comutativas e associativas.

1. **SQL**

O SQL é uma linguagem de consulta que possibilita implementar o banco de dados em sua forma física a partir de um esquema relacional. Ele utiliza os termos tabela para relação, linha para tuplas e colunas para atributos.

* 1. **Comandos básicos**

Alguns comandos são:

* CREATE TABLE <nome da tabela> (); : esse comando cria uma nova relação, seus atributos ficam dentro dos parênteses, com o domínio de cada um e suas restrições. Também podem ser especificados as restrições CONSTRAINT, de chave (PRIMARY KEY, UNIQUE) e de integridade referencial (FOREIGN KEY);
* Os domínios dos atributos podem ser: inteiros (integer ou int), real (float, real, double precision), caracteres de tamanho fixo (char(n) ou character(n)) e de tamanho variável (varchar(n)) onde “n” é o tamanho máximo de caracteres. Também possui os tipos para representar data (date aaaa-mm-dd) e hora (time hh:mm:ss) e criar domínios com outros nomes com o comando CREATE DOMAIN. Ex: CREATE DOMAIN tipoNum AS CHAR(9).
* ALTER TABLE <nome da tabela>: comando que permite alterações das definições de uma relação. Pode ser utilizada para especificar restrições de chave e integridade referencial.
* NOT NULL e DEFAULT: São restrições dos atributos. Como o SQL aceita valores nulos, adiciona-se a restrição NOT NULL para impedir isso. E para declarar um valor padrão/inicial para a variável, utiliza-se a restrição DEFAULT <valor>.
* ON DELETE/ON UPDATE: São funções que especificam a ação tomada na exclusão/atualização de uma tupla que está sendo referenciada por outra relação. Possui os valores CASCADE (propagação), SET NULL (substituição por nulo) e SET DEFAULT (substituição por valor default), sendo a opção de bloqueio a opção padrão.
* DROP SCHEMA: remove um esquema em um banco de dados, sua sintaxe é DROP SCHEMA <E> <opção>, sendo <E> o nome do esquema e a <opção> RESTRICT (que não elimina caso houver elementos nesse esquema) e CASCADE (elimina o esquema e todos os elementos).
* DROP TABLE: remove uma relação do banco de dados e sua sintaxe é DROP TABLE <R> <opção>, sendo <R> o nome da tabela e a <opção> RESTRICT (não remove a tabela caso haja alguma restrição ou visão) e CASCADE (elimina a tabela e todas as visões e restrições associadas a ela).
* INSERT VALUE: adiciona uma tupla na tabela especificada.
  1. **Comandos de Consulta**

Uma consulta em um banco de dados SQL se resume basicamente nas cláusulas SELECT <ai> FROM <Ri> WHERE <cond>, sendo <ai> uma lista de atributos, <Ri> uma lista de relações e <cond> a condição que deve ser atendida para que a lista de atributos seja retornada.

Como o SQL não remove tuplas iguais, usa-se a palavra-chave DISTINCT após o SELECT para que ele apenas retorne as tuplas que sejam distintas.

Em SQL também existe o operador LIKE que permite comparações com partes de cadeias de caractere, sendo o “%” que substitui um numero indeterminado de caracteres e o “\_” que substitui apenas um caractere. O operador BETWEEN permite comparações em um determinado intervalo de valores.

O SQL também permite utilizar operações aritméticas nas pesquisas (+, -, /, \*). Ele também permite o uso de ORDER BY <clasula> que ordena as tuplas retornadas pelo SELECT de acordo com a condição especificada em <clasula>.