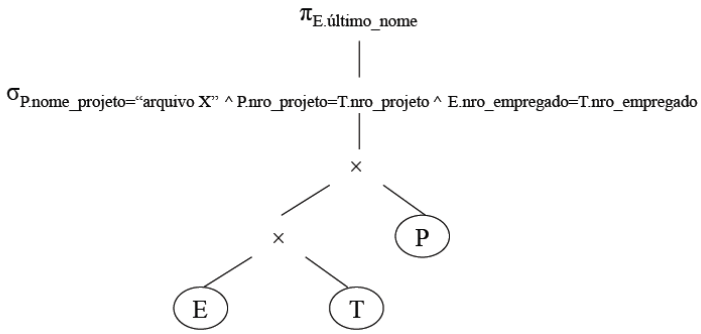
**UEM – UNIVERSIDADE ESTADUAL DE MARINGÁ**
**CTC – CENTRO DE TECNOLOGIA**
**DIN - DEPARTAMENTO DE INFORMÁTICA**
**PROFESSORA: HELOISE MANICA PARIS TEIXEIRA**
**DISCIPLINA: BANCO DE DADOS II**

RESPOSTAS DE EXERCÍCIOS SELECIONADOS - PROCESSAMENTO DE CONSULTA

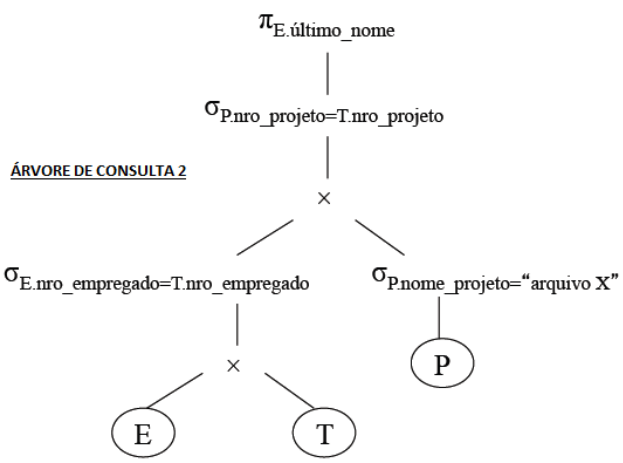
1) Dado o seguinte comando SQL e sua árvore de consulta:
   a. construa uma árvore de consulta equivalente que torne a consulta mais eficiente.
   b. indique algoritmos a serem aplicados em cada operação

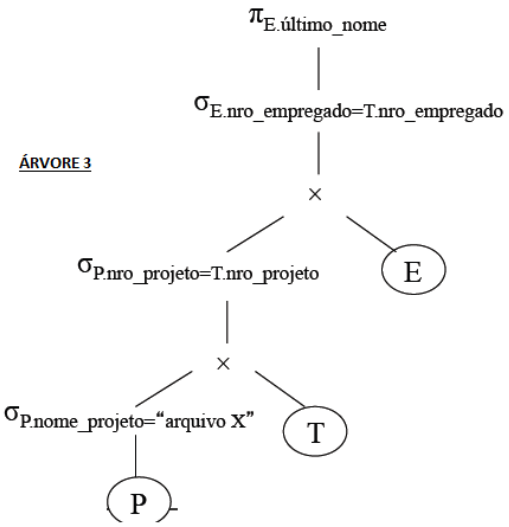



```
SELECT E.ultimo_nome
FROM empregado E, trabalha T, projeto P
WHERE P.nome_projeto = "arquivo X"
AND P.nro_projeto = T.nro_projeto
AND E.nro_empregado = T.nro_empregado
```

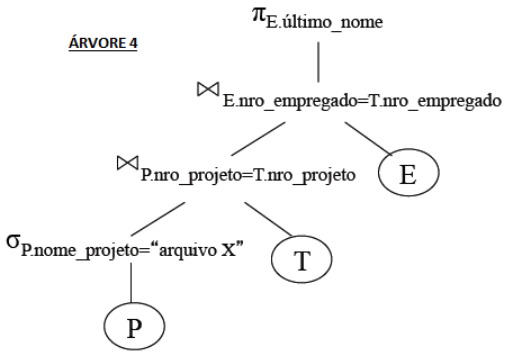
Resposta com comentários:

O produto cartesiano das relações empregado, trabalha e projeto produz uma grande relação, a qual provavelmente precisará ser armazenada em disco.
Objetivo → **reduzir** o tamanho dos resultados intermediários;
Heurística: executar as operações de seleção tão cedo quanto possível → árvore de consulta 2.
Heurística: diminuir os tamanhos das relações a serem utilizadas no produto cartesiano → árvore 3.
Heurística: substituir operações de produto cartesiano seguidas pelos respectivos critérios de seleção por operações de junção → árvore 4.
Heurística: executar as operações de projeção tão cedo quanto possível→ árvore 5.

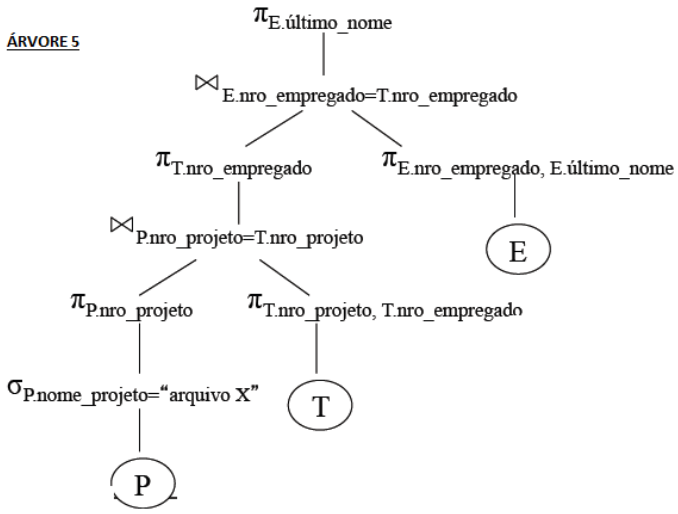


ÁRVORE 4




ÁRVORE 5



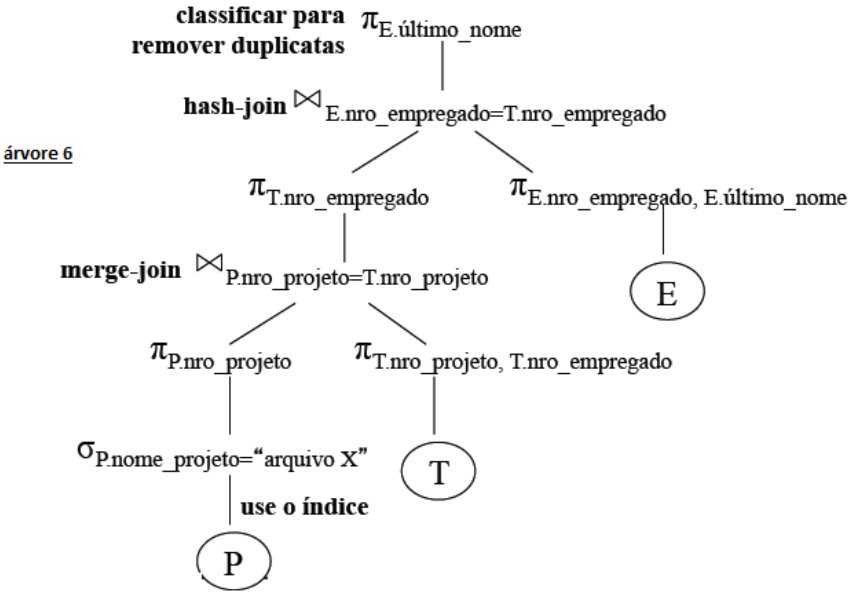

➔ A geração de expressões é apenas parte do processo de otimização de consultas. Cada operação na expressão pode ser implementada com diferentes algoritmos.

➔ Uma estratégia de consulta define exatamente que algoritmo é utilizado para cada operação e como a execução das operações é coordenada → árvore 6.

➔ Exemplo: operação de junção:

– junção de laço aninhado
– junção de laço aninhado em blocos
– junção de laço aninhado indexado
– merge-join
– hash-join

árvore 6



2) Considere que as relações r1(C, D, E) e r2(C, D, E) tenham as seguintes propriedades:
   - R1 tem 20.000 tuplas
   - R2 tem 45.000 tuplas
   - Um bloco pode conter 25 tuplas de r1 ou 30 tuplas de r2

Estime o número de transferências de bloco necessárias, usando cada uma das seguintes estratégias de junção para r1 ⋈ r2

a. Junção de loop aninhado
b. Junção de loop aninhado em bloco
c. Junção merge
d. Junção hash

Resposta:

$r_1$ precisa de 800 blocos, e $r_2$ precisa de 1500 blocos. Vamos considerar $M$ páginas de memória. Se $M >$ 800, a junção pode ser feita facilmente em 1500 + 800 acessos ao disco, usando até mesmo a junção de loop aninhado simples. Assim, consideramos apenas o caso onde $M \leq 800$ páginas.

**a.** Junção de loop aninhado: Usando $n$ como a relação mais externa, precisamos de 20000 * 1500+800 = 30.000.800 acessos ao disco, se $r_2$ for a relação mais externa, precisamos de 45000 * 800 + 1500 = 36.001.500 acessos ao disco.

**b.** Junção de loop aninhado em bloco: Se $r_1$ for a relação mais externa, precisamos de $\left\lceil \frac{800}{M-1} \right\rceil$ * 1500 + 800 acessos ao disco; se $r_2$ for a relação mais externa, precisamos de $\left\lceil \frac{1500}{M-1} \right\rceil$ * 800 + 1500 acessos ao disco.

**c.** Junção merge: Supondo que $r_1$ e $r_2$ não estão inicialmente classificados na chave de junção, o custo de classificação total inclusivo da saída é $B_s = 1500(2\lfloor log_{M-1}(1500/M) \rceil + 2) + 800(2\_log_{M-1}(800/M) \rceil +$ 2) acessos ao disco. Supondo que todas as tuplas com o mesmo valor para os atributos da junção cabem na memória, o custo total é $B_s$ + 1500 + 800 acessos ao disco.

**d.** Junção hash: Consideramos que não haja overflows. Como $r_1$ é menor, nós o usamos como relação de montagem e $r_2$ como relação de sonda. Se $M >800/M$, ou seja, não é preciso particionamento recursivo, então o custo é 3(1500 + 800) = 6900 acessos ao disco; senão, o custo é 2(1500 + 800)$\lceil log_{M-1}(800) - 1 \rceil$ + 1500 + 800 acessos ao disco.

3) Mostre que as seguintes equivalências se mantêm. Explique como você pode aplica-las para melhorar a eficiência de certas consultas.

**a.** $E_1 \bowtie_\theta (E_2 - E_3) = (E_1 \bowtie_\theta E_2 - E_1 \bowtie_\theta E_3)$

**b.** $\sigma_\theta ({}_A\mathcal{G}_F (E)) = {}_A\mathcal{G}_F (\sigma_\theta(E))$, onde $\theta$ utiliza apenas atributos de $A$.

**c.** $\sigma_\theta(E_1 \rceil\bowtie E_2) = \sigma_\theta(E_1) \rceil\bowtie E_2$, onde $\theta$ usa apenas atributos de $E_1$.

Respostas:

**a.** $E_1 \bowtie_\theta (E_2 - E_3) = (E_1 \bowtie_\theta E_2 - E_1 \bowtie_\theta E_3)$
Vamos renomear $(E_1 \bowtie_\theta (E_2\text{-}E_3))$ como $R_1$, $(E_1 \bowtie_\theta E_2)$ como $R_2$ e $(E_1 \bowtie_\theta E_3)$ como $R_3$. É claro que, se uma tupla $t$ pertence a $R_1$, ela também pertencerá a $R_2$. Se uma tupla $t$ pertence a $R_3$, $t$[atributos de $E_3$] pertencerá a $E_3$, e por isso $t$ não pode pertencer a $R_1$. Destes dois, podemos dizer que

$$\forall t,\ t \in R_1 \Rightarrow t \in (R_2 - R_3)$$

É claro que, se uma tupla $t$ pertence a $R_2 - R_3$, então $t$[atributos de $R_2$] $\in E_2$ e $t$[atributos de $R_2$] $\notin E_3$. Portanto:

$$\forall t,\ t \in (R_2 - R_3) \Rightarrow t \in R_1$$

As duas equações acima implicam na equivalência indicada.
Essa equivalência é útil porque a avaliação da junção do lado direito produzirá muitas tuplas que finalmente serão removidas do resultado. A expressão do lado esquerdo pode ser avaliada de forma mais eficiente.

**b.** $\sigma_\theta\ ({}_A\mathcal{G}_F\ (E)) = {}_A\mathcal{G}_F\ (\sigma_\theta(E))$, onde $\theta$ utiliza apenas atributos de $A$.
$\theta$ usa apenas atributos de $A$. Portanto, se alguma tupla $t$ na saída de ${}_A\mathcal{G}_F\ (E)$ for filtrada pela seleção do lado esquerdo, todas as tuplas em $E$ cujo valor em $A$ é igual a $t[A]$ são filtradas pela seleção do lado direito. Portanto:

$$\forall t,\ t \notin \sigma_\theta({}_A\mathcal{G}_F\ (E)) \Rightarrow t \notin {}_A\mathcal{G}_F\ (\sigma_\theta(E))$$

Usando um raciocínio semelhante, também podemos concluir que

$$\forall t,\ t \notin {}_A\mathcal{G}_F\ (\sigma_\theta(E)) \Rightarrow t \notin \sigma_\theta({}_A\mathcal{G}_F\ (E))$$

As duas equações acima implicam na equivalência indicada.
Essa equivalência é útil porque a avaliação do lado direito evita a realização da agregação sobre grupos que de alguma forma serão removidos do resultado. Assim, a expressão do lado direito pode ser avaliada de forma mais eficiente do que a expressão do lado esquerdo.

**c.** $\sigma_\theta(E_1 \rtimes E_2) = \sigma_\theta(E_1) \rtimes E_2$, onde $\theta$ usa apenas atributos de $E_1$.
$\theta$ usa apenas atributos de $E_1$. Portanto, se alguma tupla $t$ na saída de $(E_1 \rtimes E_2)$ for filtrada pela seleção do lado esquerdo, todas as tuplas em $E_1$ cujo valor for igual a $t[E_1]$ são filtradas pela seleção do lado direito. Portanto:

$$\forall t,\ t \notin \sigma_\theta(E_1 \rtimes E_2) \Rightarrow t \notin \sigma_\theta(E_1) \rtimes E_2$$

Usando um raciocínio semelhante, também podemos concluir que

$$\forall t,\ t \notin \sigma_\theta(E_1) \rtimes E_2 \Rightarrow t \notin \sigma_\theta(E_1 \rtimes E_2)$$

As duas equações acima implicam na equivalência indicada.
Essa equivalência é útil porque a avaliação do lado direito evita a produção de muitas tuplas de saída que de alguma forma serão removidas do resultado. Assim, a expressão do lado direito pode ser avaliada de forma mais eficiente do que a expressão do lado esquerdo.

4) Dê um exemplo de relações para mostrar que as expressões $\Pi_A(R-S)$ e $\Pi_A(R) - \Pi_A(S)$ não são equivalentes.

Resposta:

$R = \{(1, 2)\}$, $S = \{(1, 3)\}$
O resultado da expressão do lado direito é $\{(1)\}$, onde o resultado da expressão do lado direito é vazio.

5) Considere as relações $r_1$(A,B,C), $r_2$(C,D,E) e $r_3$(E,F), com chaves primárias A, C e E, respectivamente. Suponha que r1 tenha 1.000 tuplas, r2 tenha 1.500 tuplas e r3 tenha 750 tuplas. Estime o tamanho de $r_1 \bowtie r_2 \bowtie r_3$ e dê uma estratégia eficiente para calcular a junção.

Resposta:

- A relação resultante da junção de $r_1$, $r_2$ e $r_3$ será a mesma, não importa de que maneira as juntemos, devido às propriedades associativa e comutativa das junções. Assim, vamos considerar o tamanho com base na estratégia de $((r_1 \bowtie r_2) \bowtie r_3)$. A junção de $r_1$ com $r_2$ gerará uma relação de no máximo 1000 tuplas, pois *C* é a chave para $r_2$. De modo semelhante, a junção desse resultado com $r_3$ gerará uma relação de no máximo 1000 tuplas, pois *E* é uma chave para $r_3$. Portanto, a relação final terá no máximo 1000 tuplas.
- Uma estratégia eficiente para calcular essa junção seria criar um índice sobre o atributo *C* para a relação $r_2$ e sobre *E* para $r_3$. Depois, para cada tupla em $r_1$, fazemos o seguinte:
  a. Use o índice para $r_2$ para examinar no máximo uma tupla que combina com o valor *C* de $r_1$.
  b. Use o índice criado em *E* para pesquisar em $r_3$ no máximo uma tupla que combina com o valor exclusivo para *E* em $r_2$.

6) Considere as mesmas relações do exercício anterior e que não existam chaves primárias. Considere que V(C,r1)=900, V(C,r2)=1000, V(E,r2)=50, V(E,r3)=100. Suponha que r1 tem 1.000 tuplas, r2 tem 1.500 e r3 750 tuplas. Estime o tamanho de $r_1 \bowtie r_2 \bowtie r_3$ e uma estratégia eficiente para calcular a junção.

Resposta:

O tamanho estimado da relação pode ser determinado pelo cálculo do número médio de tuplas que seriam juntadas a cada tupla da segunda relação. Nesse caso, para cada tupla em $r_1$, $1500/V(C, r_2) = 15/11$ tuplas (na média) of $r_2$ se juntariam a ela. A relação intermediária teria 15000/11 tuplas. Essa relação é juntada com $r_3$ para gerar um resultado de aproximadamente 10.227 tuplas ($15000/11 \times 750/100 = 10227$). Uma boa estratégia deverá juntar $r_1$ e $r_2$ primeiro, pois a relação intermediária tem aproximadamente o mesmo tamanho de $r_1$ ou $r_2$. Depois, $r_3$ é juntada a esse relatório.