

AULA 6 – OTIMIZAÇÃO DE CONSULTA

PROFA. DRA. LEILA BERGAMASCO

CC6240 – Tópicos Avançados em Banco de Dados

AGENDA

- Otimização de consultas

RECAPITULANDO

- Otimização de Consulta
 - O usuário não escreve a consulta pensando em eficiência
 - Cabe ao SGBD elaborar diferentes estratégias para recuperar de forma eficiente os dados.
 - Aulas passadas vimos como partes específicas das consultas influenciam o resultado
 - Consultas utilizando atributos que sejam ou não chaves primárias
 - Consultas utilizando atributos que sejam ou não índices
 - Consultas utilizando =, < , >
 - Consultas que utilizam ORDER BY
 - Consultas que utilizam JOIN
 - Merge, Hash, Nested (Aninhado)
 - Hoje veremos como o ARRANJO dessas cláusulas são processadas pelo SGBD

ANTES, VAMOS RELEMBRAR ALGUNS TERMOS MATEMÁTICOS...

- Propriedade associativa

- A propriedade associativa é uma regra matemática que determina que a ordem das operações não muda o resultado (se a operação for a mesma).

- Exemplo:

- $(7 + 4) + 5 = 16 \rightarrow 4 + (5 + 7) = 16$

- Propriedade comutativa

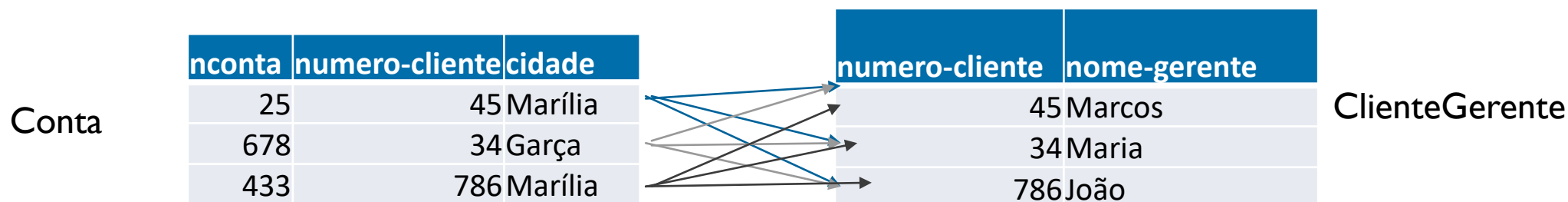
- A propriedade comutativa é uma regra matemática que determina que a ordem dos termos não muda o resultado.

- Exemplo:

- $7 + 4 = 11 \rightarrow 4 + 7 = 11$

... E ENTENDER O QUE É JUNÇÃO THETA

- No produto cartesiano, temos que $A \times B$ é uma combinação todos-para-todos: Conta X ClienteGerente



nconta	numero-cliente	cidade	numero-cliente	nome-gerente
25	45	Marília	45	Marcos
25	45	Marília	34	Maria
25	45	Marília	786	João
678	34	Garça	45	Marcos
678	34	Garça	34	Maria
678	34	Garça	786	João
433	786	Marília	45	Marcos
433	786	Marília	34	Maria
433	786	Marília	786	João

Essas condições são chamadas junções theta. Quando há condições de igualdade dizemos que é uma equi-junção

E ENTENDER O QUE É JUNÇÃO THETA

- Recuperar os números de cliente e a cidade dos clientes do gerente João

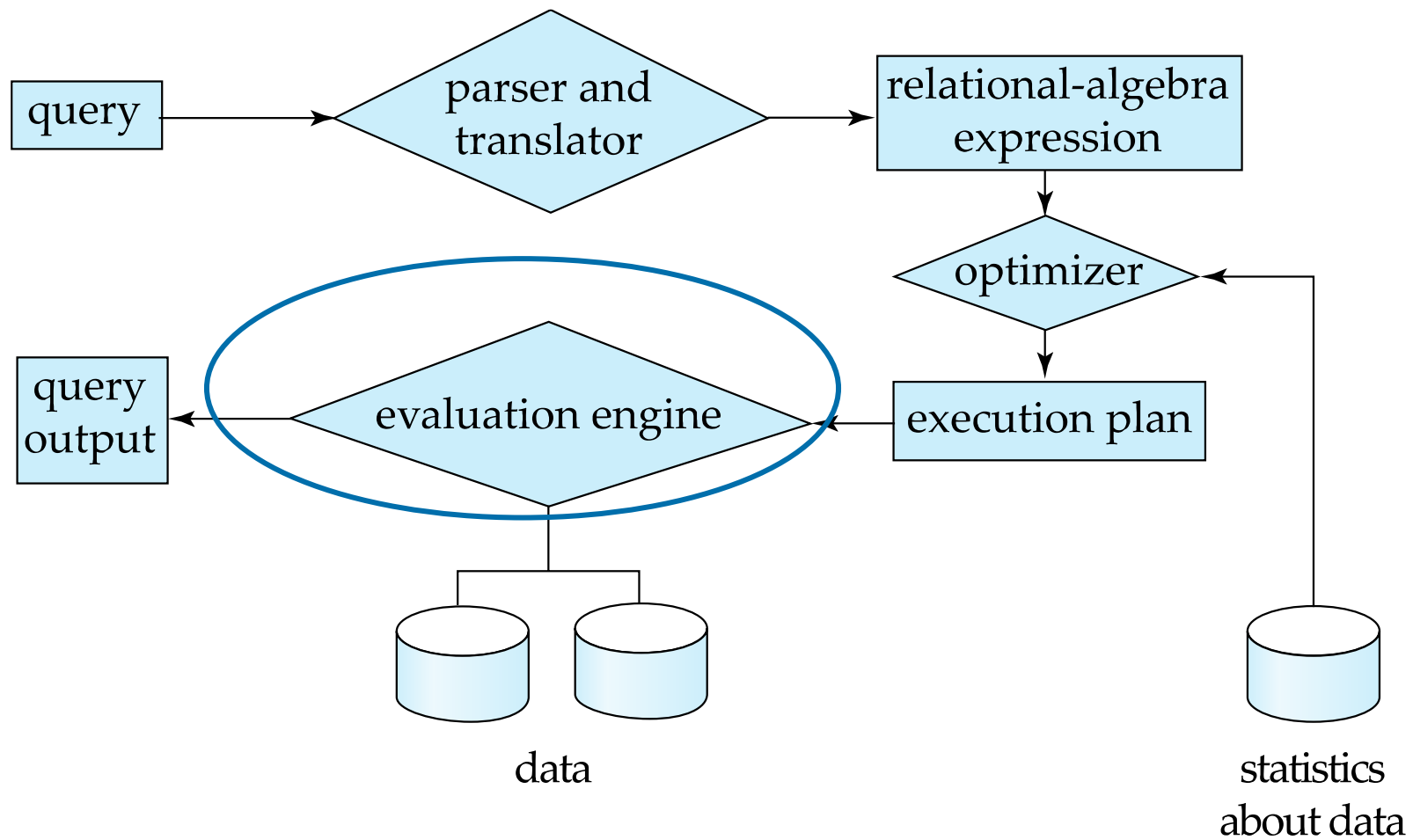
```
SELECT c.numero-cliente, cli.cidade
FROM cliente cli
INNER JOIN conta c on c.numero-cliente = cli.numero_cliente
WHERE nome-gerente ilike 'João%'
```

$$\Pi_{\text{conta.numero-cliente, cidade}} (\sigma_{\text{conta.numero-cliente=ClienteGerente.numero-cliente} \wedge \text{nome-gerente="João"}} (\text{ContaXClienteGerente}))$$

- É uma combinação dos operadores produto cartesiano e seleção. Ela cria uma relação pela combinação dos campos de uma relação com aquelas de uma outra baseada em uma comparação de valores entre colunas que não necessariamente tem o mesmo nome
- $\langle \text{tabela 1 (E1)} \rangle \times \langle \text{condição} \rangle \langle \text{tabela 2 (E2)} \rangle$

Conta \times $\Pi_{\text{conta.numero-cliente=ClienteGerente.numero-cliente} \wedge \text{nome-gerente="João"}}$ *ContaGerente*

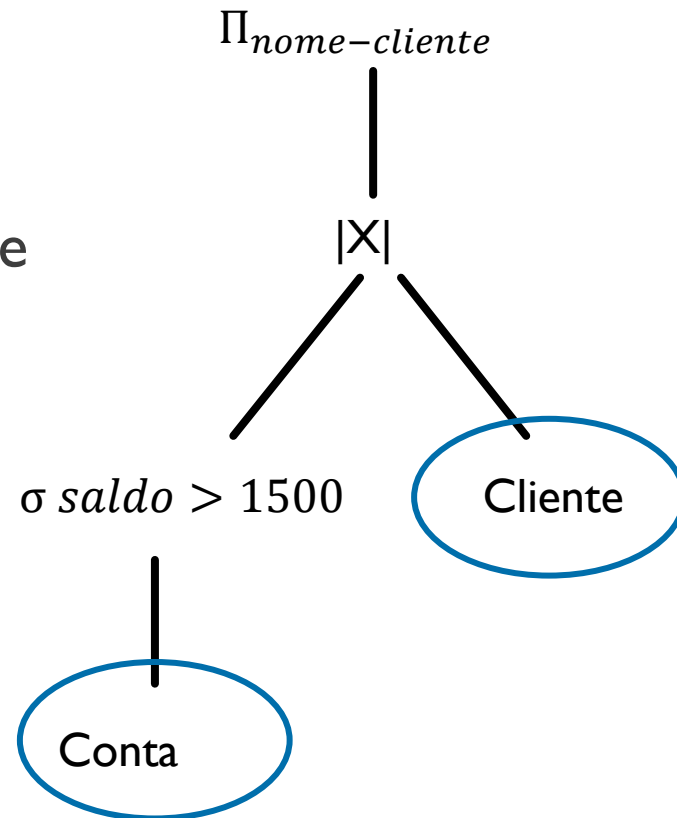
PROCESSAMENTO DE CONSULTAS



ÁRVORE CANÔNICA

$$\Pi_{nome-cliente}(\sigma_{conta.saldo > 1500}(Conta \bowtie Cliente))$$

- Recuperar os nomes de clientes que possuam saldo maior que 1500



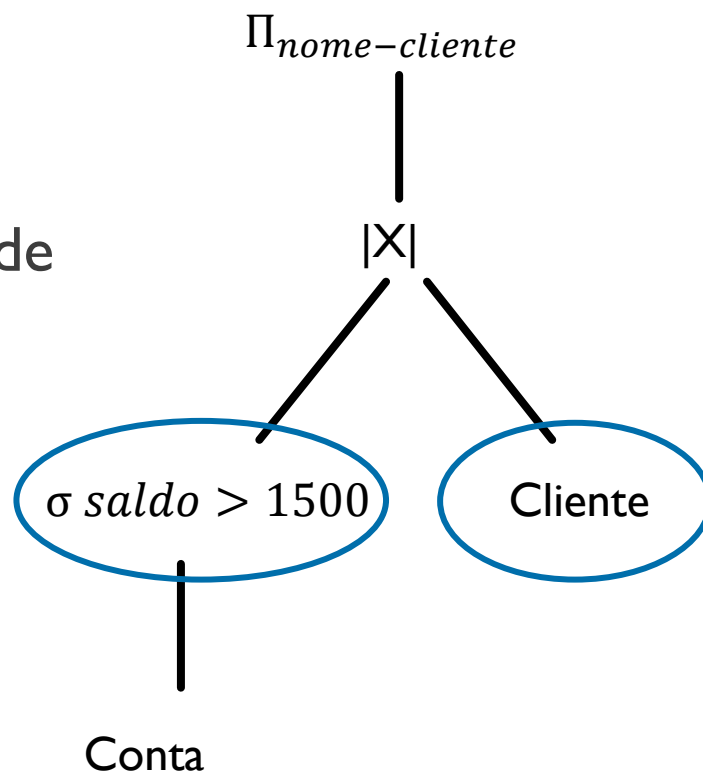
Começa com operações de nível mais baixo, que têm como entrada as relações do BD. Cada operação tem resultado armazenado em relação temporária no BD.

Deve-se considerar o custo das operações envolvidas e do armazenamento temporário.

ÁRVORE CANÔNICA

$\Pi_{nome-cliente}(\sigma_{conta.saldo > 1500}(Conta \bowtie Cliente))$

- Recuperar os nomes de clientes que possuam saldo maior que 1500



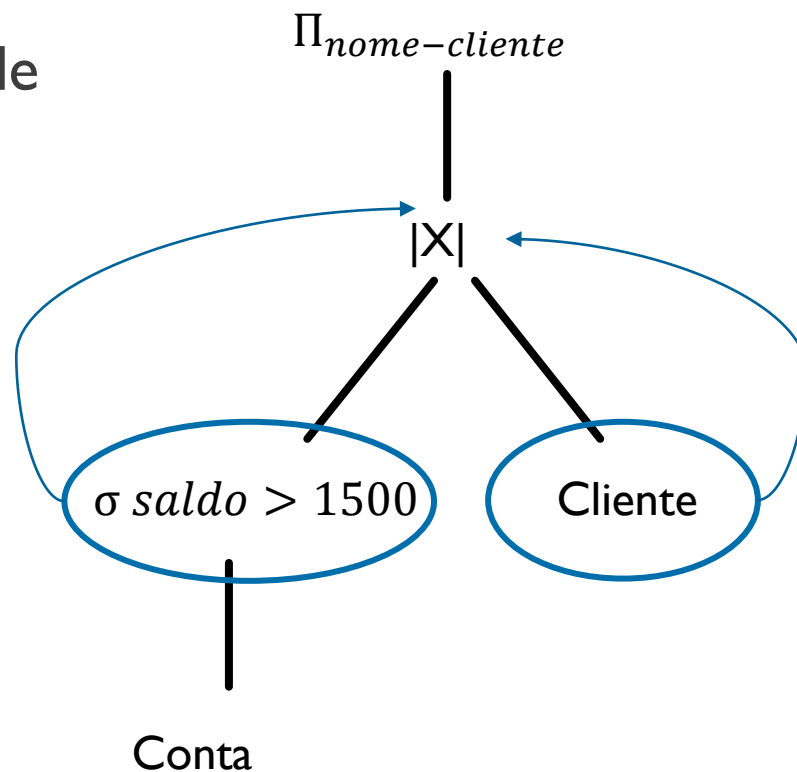
Começa com operações de nível mais baixo, que têm como entrada as relações do BD. Cada operação tem resultado armazenado em relação temporária no BD.

Deve-se considerar o custo das operações envolvidas e do armazenamento temporário.

ÁRVORE CANÔNICA

$$\Pi_{nome-cliente}(\sigma_{saldo > 1500}(\text{Conta} \bowtie \text{Cliente}))$$

- Recuperar os nomes de clientes que possuam saldo maior que 1500

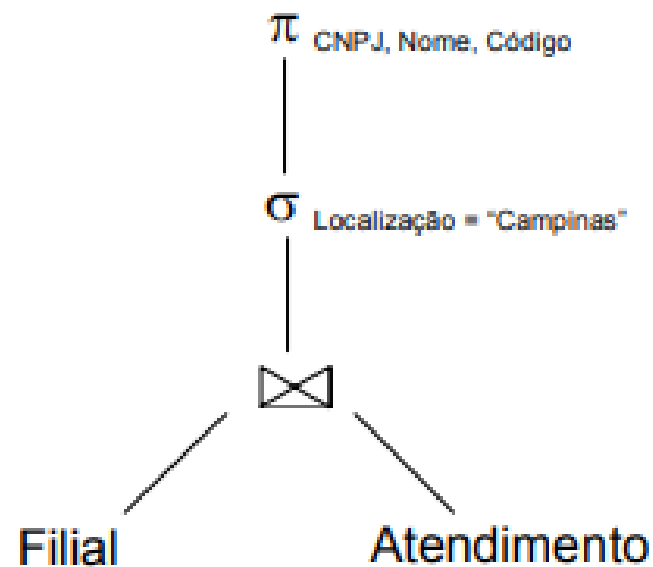


Resultado de uma operação é passado para próxima etapa, sem armazenamento temporário.

ÁRVORES DE EXECUÇÃO COM +1 TABELA

Qual a consulta correspondente dessa árvore canônica?

$\pi_{\text{CNPJ, Nome, Código}}(\sigma_{\text{Localização}=\text{"Campinas"}}(\text{Filial} \bowtie \text{Atendimento}))$



PROCESSAMENTO DE CONSULTA

- E quando há mais de uma tabela envolvida?
- Recupere os nomes de todos os clientes que possuem uma conta em qualquer agência localizada no Brooklyn.



$$\Pi_{cli.nome-cliente}(\sigma_{cidade-agencia='Brooklyn'} (Conta \mid X \mid Agencia \mid X \mid Depositante))$$

Recupere os nomes de todos os clientes que possuem uma conta em qualquer agência localizada no Brooklyn.

Como fica a consulta?

agência			cliente		
nome_agência	cidade_agência	fundos	nome_cliente	rua_cliente	cidade_cliente
Downtown	Brooklyn	900000	Jones	Main	Harrison
Redwood	Palo Alto	210000	Smith	North	Rye
Perrydige	Horseneck	170000	Hayes	Main	Harrison
Mianus	Horseneck	40000	Curry	North	Rye
Round Hill	Horseneck	8000000	Lindsay	Park	Pittfield
Pownal	Bennington	30000	Turner	Putnam	Stamford
North Town	Rye	370000	Williams	Nassau	Princeton
Brighton	Brooklyn	710000	Adams	Spring	Pittsfield
			Johnson	Alma	Palo Alto
			Glenn	Sand Hill	Woodside
			Brooks	Senator	Brooklyn
			Green	Walnut	Stamford

nome_cliente	número_conta
Johnson	A-101
Smith	A-215
Hayes	A-102
Turner	A-305
Johnson	A-201
Jones	A-217
Lindsay	A-222

nome_agência	número_conta	saldo
Downtown	A-101	500
Mianus	A-215	700
Perryridge	A-102	400
Round Hill	A-305	350
Brighton	A-201	900
Redwood	A-222	700
Brighton	A-217	750

depositante			conta		
-------------	--	--	-------	--	--

PROCESSAMENTO DE CONSULTA

- E quando há mais de uma tabela envolvida?
- Recupere os nomes de todos os clientes que possuem uma conta em qualquer agência localizada no Brooklyn.
- Para resolver esta expressão, seguindo a forma como ela está escrita, é necessário criar uma relação intermediária grande (a junção das três relações)

$$\pi_{\text{nome_cliente}} \left(\sigma_{\text{cidade_agência} = \text{"Brooklyn"}} \left(\text{agência} \bowtie \text{conta} \bowtie \text{depositante} \right) \right)$$

Junção (conta |X| depositante)

nome_agência	número_conta	saldo	nome_cliente	número_conta
Downtown	A-101	500	Johnson	A-101
Mianus	A-215	700	Smith	A-215
Perryridge	A-102	400	Hayes	A-102
Round Hill	A-305	350	Turner	A-305
Brighton	A-201	900	Johson	A-201
Redwood	A-222	700	Lindsay	A-222
Brighton	A-217	750	Jones	A-217

Junção (agência |X| (conta |X| depositante))

nome_agência	número_conta	saldo	nome_cliente	nome_agência	cidade_agência	fundos
Downtown	A-101	500	Johnson	Downtown	Brooklyn	900000
Mianus	A-215	700	Smith	Mianus	Horseneck	40000
Perryridge	A-102	400	Hayes	Perrydige	Horseneck	170000
Round Hill	A-305	350	Turner	Round Hill	Horseneck	800000
Brighton	A-201	900	Johson	Brighton	Brooklyn	710000
Redwood	A-222	700	Lindsay	Redwood	Palo Alto	210000
Brighton	A-217	750	Jones	Brighton	Brooklyn	710000

PROCESSAMENTO DE CONSULTA

- Entretanto, somente as tuplas que pertencem às agências localizadas no “Brooklyn” são interessantes.
- Reescrevendo a consulta, consegue-se eliminar a necessidade de considerar as tuplas que não têm cidade_agência = “Brooklyn”, reduzindo o tamanho do resultado intermediário:

$$\pi_{\text{nome_cliente}} \left(\sigma_{\text{cidade_agência} = \text{"Brooklyn"}} \left(\text{agência } |X| \left(\text{conta } |X| \right. \right. \right. \\ \left. \left. \left. \text{depositante} \right) \right) \right)$$
$$\pi_{\text{nome_cliente}} \left(\left(\sigma_{\text{cidade_agência} = \text{"Brooklyn"}} \left(\text{agência} \right) \right) |X| \left(\text{conta } |X| \right. \right. \\ \left. \left. \text{depositante} \right) \right)$$

Junção (conta |X| depositante)

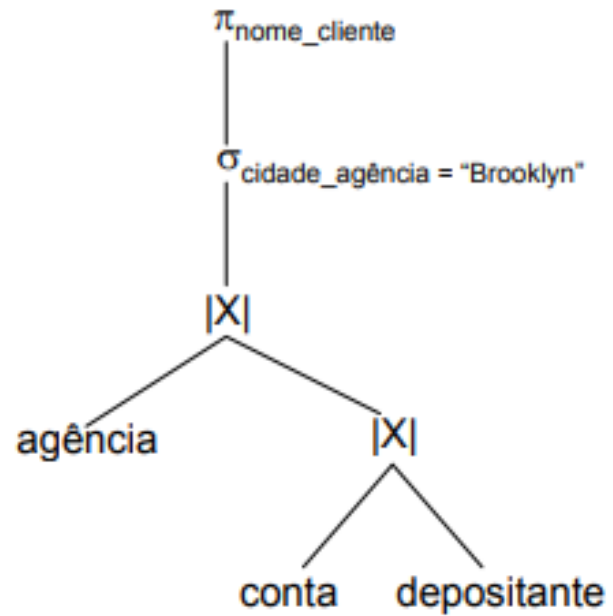
nome_agência	número_conta	saldo	nome_cliente	número_conta
Downtown	A-101	500	Johnson	A-101
Mianus	A-215	700	Smith	A-215
Perryridge	A-102	400	Hayes	A-102
Round Hill	A-305	350	Turner	A-305
Brighton	A-201	900	Johnson	A-201
Redwood	A-222	700	Lindsay	A-222
Brighton	A-217	750	Jones	A-217

$\sigma_{\text{cidade_agência} = \text{"Brooklyn"}} (\text{agência})$

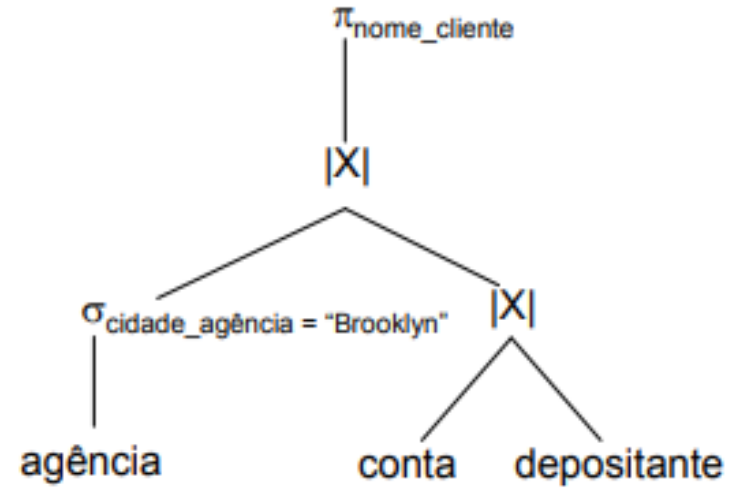
nome_agência	cidade_agência	fundos
Downtown	Brooklyn	900000
Brighton	Brooklyn	710000

$(\sigma_{\text{cidade_agência} = \text{"Brooklyn"}} (\text{agência})) |X| (\text{conta } |X| \text{ depositante})$

nome_agência	número_conta	saldo	nome_cliente	cidade_agência	fundos
Downtown	A-101	500	Johnson	Brooklyn	900000
Brighton	A-201	900	Johnson	Brooklyn	710000
Brighton	A-217	750	Jones	Brooklyn	710000



(a) Árvore da expressão inicial



(b) Árvore da expressão transformada

COMO OTIMIZAR?

- Dada uma expressão em álgebra relacional, o otimizador deve propor um plano de avaliação que gere o mesmo resultado com menor custo.
- Para isso, é necessário gerar planos alternativos.
 - Como?
 - Regras de equivalência algébrica
 - Devem ser conhecidas pelo otimizador para que possam ser geradas transformações válidas.
 - Algoritmo de otimização algébrica
 - Indica a ordem de aplicação das regras e de outros processamentos de otimização.

REGRAS DE EQUIVALÊNCIA ALGÉBRICA

- Expressões de duas formas diferentes são equivalentes
- Preservar a equivalência = relações geradas pelas duas expressões têm o mesmo conjunto de atributos e contêm o mesmo conjunto de tuplas, embora seus atributos possam estar ordenados de forma diferente.

σ saldo < 2500 (π saldo (conta))

- Notação:
 - $\theta, \theta_1, \dots, \theta_n$ são predicados
 - L_1, L_2, \dots, L_n são listas de atributos
 - E_1, E_2, \dots, E_n são relações

π saldo (σ saldo < 2500(conta))

COMO OTIMIZAR?

- 1. As operações de uma seleção conjuntiva (que utilizam AND) podem ser divididas em uma sequência de seleções individuais

$$S_{q_1 \cup q_2}(E) = S_{q_1}(S_{q_2}(E))$$

- 2. As operações de seleção são comutativas

WHERE a=10 and b=12

=

WHERE b=12 and a=10

$$S_{q_1}(S_{q_2}(E)) = S_{q_2}(S_{q_1}(E))$$

ordem dos termos não muda o resultado.

- 3. Somente operações finais são necessárias em uma sequência de operações de projeção.

$$\Pi_{L_1}(\Pi_{L_2}(\dots(\Pi_{L_n}(E))\dots)) = \Pi_{L_1}(E)$$

$Conta \mid X \mid_{nome-gerente="João"}(ContaGerente)$

COMO OTIMIZAR?

$$\Pi_{conta.numero-cliente, cidade}(\sigma_{conta.numero-cliente=ClienteGerente.numero-cliente} \wedge \sigma_{nome-gerente="João"}(Conta \times ClienteGerente))$$

- 4. Seleções podem ser combinadas com os produtos cartesianos e junções theta.

- Se todos os atributos da condição forem atributos de apenas uma relação R

$$\sigma_{\theta}(E_1 \times E_2) = E_1 \mid X \mid_{\theta} E_2$$

- 5. Operações de junção theta (e junções naturais) são comutativas

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

Porém lembre-se que o custo muda! Esse é o objetivo

COMO OTIMIZAR?

depositante

nome cliente	número conta
--------------	--------------

conta

nome_agência	número_conta	saldo
--------------	--------------	-------

agência

nome_agência	cidade_agência	fundos
--------------	----------------	--------

- 6a. Operações de junção natural são associativas:

$$E_1 \bowtie_{\theta} E_2 = E_2 \bowtie_{\theta} E_1$$

onde θ_2 contém somente atributos de E_2 e E_3 .

$$(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2} E_3 = E_1 \bowtie_{\theta_1} (E_2 \bowtie_{\theta_2} E_3)$$

- 6b. Junções theta são associativas da seguinte maneira:

$(\text{Depositante} \mid X \mid \text{dep.nome} - \text{cliente} = ' \text{Jones}' \mid \text{Conta}) \mid X \mid \text{nome} - \text{agencia} = \text{Brooklyn Agencia}$

E1

E2

E3

$(\text{Depositante} \mid \text{dep.nome} - \text{cliente} = ' \text{Jones}' \mid \text{Conta}) \mid X \mid \text{nome} - \text{agencia} = \text{Brooklyn Agencia}$

E1

E2

E3

COMO OTIMIZAR?

- 7. A operação de seleção pode ser distribuída por meio da operação de junção theta, observando as seguintes condições:
 - 7a. Quando todos os atributos de θ_0 envolvem somente os atributos de uma das expressões (E_1) que estão participando da junção.

$$\sigma_{\theta_0}(E_1 \bowtie E_2) = (\sigma_{\theta_0}(E_1)) \bowtie E_2$$

- 7b. Quando θ_1 envolve somente atributos de E_1 e θ_2 envolve somente atributos de E_2 .

$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie E_2) = (\sigma_{\theta_1}(E_1)) \bowtie (\sigma_{\theta_2}(E_2))$$

COMO OTIMIZAR?

- 7b. Quando θ_1 envolve somente atributos de E_1 e θ_2 envolve somente atributos de E_2

$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \mid X \mid E_2) = (\sigma_{\theta_1}(E_1)) \mid X \mid (\sigma_{\theta_2}(E_2))$$

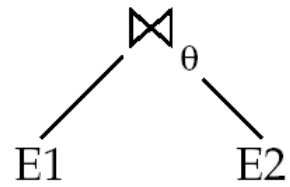
- “Recuperar os nomes dos clientes com nome de cliente \neq Jones e saldo > 500
- $\sigma_{\text{nome-cliente}='Jones' \wedge \text{saldo}>500}(\text{Depositante} \mid X \mid \text{Conta})$
- $(\sigma_{\text{nome-cliente}='Jones'}(\text{Depositante})) \mid X \mid (\sigma_{\text{saldo}>500}(\text{Conta}))$

depositante

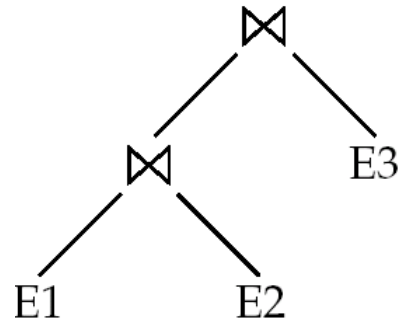
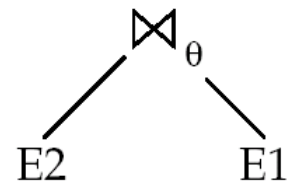
nome_cliente	número_conta
--------------	--------------

conta

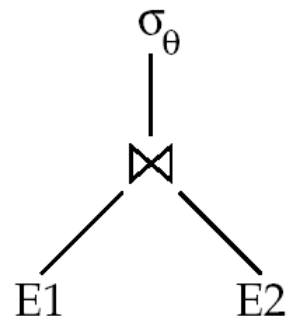
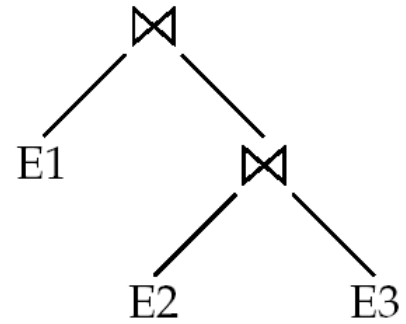
nome_agência	número_conta	saldo
--------------	--------------	-------



Rule 5

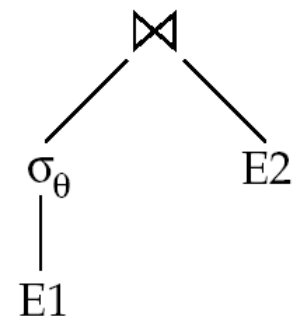


Rule 6a



Rule 7a

If θ only has
attributes from E1



COMO OTIMIZAR?

- 8. A operação de projeção é distribuída pela operação de junção theta como segue:
 - Sejam L_1 e L_2 conjuntos de atributos de E_1 e E_2 , respectivamente.
 - 8a. Se θ envolve unicamente atributos de $L_1 \cup L_2$:

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = (\Pi_{L_1}(E_1)) \bowtie_{\theta} (\Pi_{L_2}(E_2))$$

COMO OTIMIZAR?

- 9. Os conjuntos de operações união e intersecção são comutativos

$$E_1 \cup E_2 = E_2 \cup E_1$$

$$E_1 \cap E_2 = E_2 \cap E_1$$

- 10. União e intersecção são associativas

$$(E_1 \cup E_2) \cup E_3 = E_1 \cup (E_2 \cup E_3)$$

$$(E_1 \cap E_2) \cap E_3 = E_1 \cap (E_2 \cap E_3)$$

COMO OTIMIZAR?

- I 1. A operação seleção é distribuída sobre \vee , \cap e $-$.

$$\sigma_{\theta} (E_1 - E_2) = \sigma_{\theta} (E_1) - \sigma_{\theta}(E_2)$$

- I 2. A operação de projeção é distribuída por meio da operação de união

$$\Pi_L(E_1 \cup E_2) = (\Pi_L(E_1)) \cup (\Pi_L(E_2))$$

EXEMPLO DE TRANSFORMAÇÃO

- Agencia (nome_agência, cidade_agência, fundos)
- Conta (nome_agência, número_conta, saldo)
- ClienteConta(nome_cliente, número_conta)

EXEMPLO DE TRANSFORMAÇÃO

$\Pi_{\text{nome-cliente}} (\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} (\text{agência} \bowtie_X (\text{conta} \bowtie_X \text{clienteConta}))$

- Mostrar os nomes de todos os clientes que têm uma conta em alguma agência localizada em Brooklyn

Como transformar usando a regra 7?

- 7. A operação de seleção pode ser distribuída por meio da operação de junção theta, observando as seguintes condições:

7a. Quando todos os atributos de θ_0 envolvem somente os atributos de uma das expressões (E_1) que estão participando da junção.

$$\sigma_{\theta_0}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_0}(E_1)) \bowtie_{\theta} E_2$$

7b. Quando θ_1 envolve somente atributos de E_1 e θ_2 envolve somente atributos de E_2 .

$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_1}(E_1)) \bowtie_{\theta} (\sigma_{\theta_2}(E_2))$$

EXEMPLO DE TRANSFORMAÇÃO

- Mostrar os nomes de todos os clientes que têm uma conta em alguma agência localizada em Brooklyn

$$\Pi_{\text{nome-cliente}} \left(\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} \left(\text{agencia} \bowtie \left(\text{conta} \bowtie \text{clienteConta} \right) \right) \right)$$

Como transformar usando a regra 7?

$$\Pi_{\text{nome-cliente}} \left(\left(\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} \left(\text{agencia} \right) \right) \bowtie \left(\text{conta} \bowtie \text{clienteConta} \right) \right)$$

- Há vantagens ?
 - relações intermediárias menores

EXEMPLO DE TRANSFORMAÇÃO

- Mostrar os nomes de todos os clientes com uma conta numa agência de Brooklyn cujo saldo é maior de \$1000

$$\Pi_{\text{nome-cliente}} \left(\sigma_{\text{cidade-agência} = \text{"Brooklyn"} \wedge \text{saldo} > 1000} \left(\text{agencia} \bowtie_X \left(\text{conta} \bowtie_X \text{clienteConta} \right) \right) \right)$$

Como transformar usando a regra 5?

Agencia (nome_agência, cidade_agência, fundos)
 Conta (nome_agência, número_conta, saldo)
 ClienteConta(nome_cliente, número_conta)

- 5. Operações de junção theta (e junções naturais) são comutativas

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

EXEMPLO DE TRANSFORMAÇÃO

- Mostrar os nomes de todos os clientes com uma conta numa agência de Brooklyn cujo saldo é maior de \$1000

$$\Pi_{\text{nome-cliente}} \left(\sigma_{\text{cidade-agência} = \text{"Brooklyn"} \wedge \text{saldo} > 1000} (\text{agência} \mid X \mid (\text{conta} \mid X \mid \text{clienteConta})) \right)$$

Como transformar usando a regra 5?

$$\Pi_{\text{nome-cliente}} \left(\left(\sigma_{\text{cidade-agência} = \text{"Brooklyn"} \wedge \text{saldo} > 1000} (\text{agência} \mid X \mid (\text{conta})) \right) \mid X \mid \text{clienteConta} \right)$$

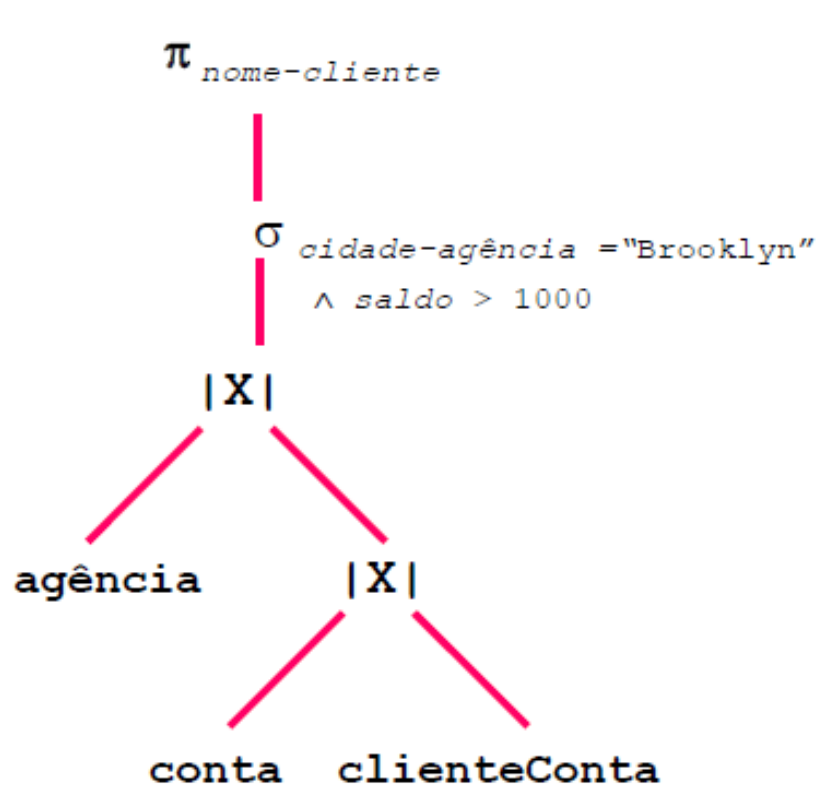
Segunda forma permite a aplicação da regra 7:

$$\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} (\text{agência}) \mid X \mid \sigma_{\text{saldo} > 1000} (\text{conta})$$

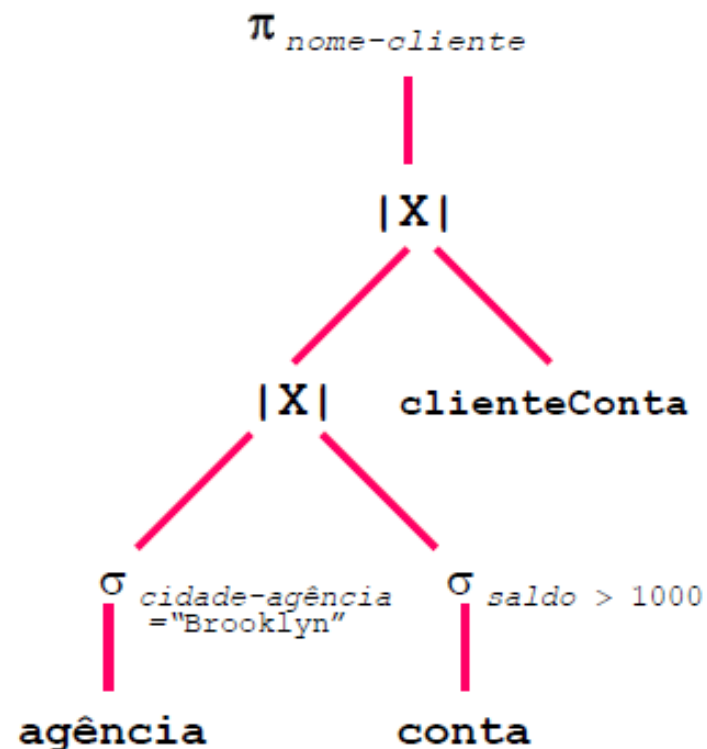
Agencia (nome_agência, cidade_agência, fundos)

Conta (nome_agência, número_conta, saldo)

ClienteConta(nome_cliente, número_conta)



$\Pi_{\text{nome-cliente}} (\sigma_{\text{cidade-agência} = \text{"Brooklyn"} \wedge \text{saldo} > 1000} (\text{agência} |X| (\text{conta} |X| \text{dono-conta})))$



$\Pi_{\text{nome-cliente}} (\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} (\text{agência}) |X| \sigma_{\text{saldo} > 1000} (\text{conta}) |X| \text{dono-conta})$

EXEMPLO DE PROJEÇÃO

$$\Pi_{\text{nome-cliente}} ((\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} (\text{agência}) \mid X \mid \text{conta}) \mid X \mid \text{clienteConta})$$

- ✓ Agência (nome_agência, cidade_agência, fundos)
- ✓ Conta (nome_agência, número_conta, saldo)
- ✓ ClienteConta (nome_cliente, número_conta)

$$(\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} (\text{agência}) \mid X \mid \text{conta})$$

- Quando calculamos...

$$(\text{nome-agência}, \text{cidade-agência}, \text{fundos}, \text{número_conta}, \text{saldo})$$

- Obtemos uma nova relação:

Fazer a projeção o mais cedo possível reduz o tamanho da relação intermediária.

- Usando a regra de equivalência 8b é possível eliminar atributos não necessários de resultados intermediários:

$$\Pi_{\text{nome-cliente}} ((\Pi_{\text{número-conta}} (\sigma_{\text{cidade-agência} = \text{"Brooklyn"}} (\text{agência}) \mid X \mid \text{conta}) \mid X \mid \text{clienteConta}))$$

DICAS - SALVA JOIN DEMORADO

- Para todas as relações $r1$, $r2$ e $r3$,
 - $(r1 \bowtie_X r2) \bowtie_X r3 = r1 \bowtie_X (r2 \bowtie_X r3)$
- Se $r2 \bowtie_X r3$ é muito grande e $r1 \bowtie_X r2$ é pequena, nós selecionamos $(r1 \bowtie_X r2) \bowtie_X r3$
- Assim é possível calcular e armazenar a menor relação temporária.

ENUMERAÇÃO DE EXPRESSÕES EQUIVALENTES

- Otimizadores de consulta usam regras de equivalência para gerar sistematicamente expressões equivalentes para uma expressão de consulta.
- Passos para gerar todas as expressões equivalentes:
- Repita
 1. Aplicar todas as regras de equivalência possíveis sobre toda expressão encontrada até o momento
 2. Adicionar expressões geradas ao conjunto de expressões equivalentes
- Até que nenhuma nova expressão equivalente possa ser gerada

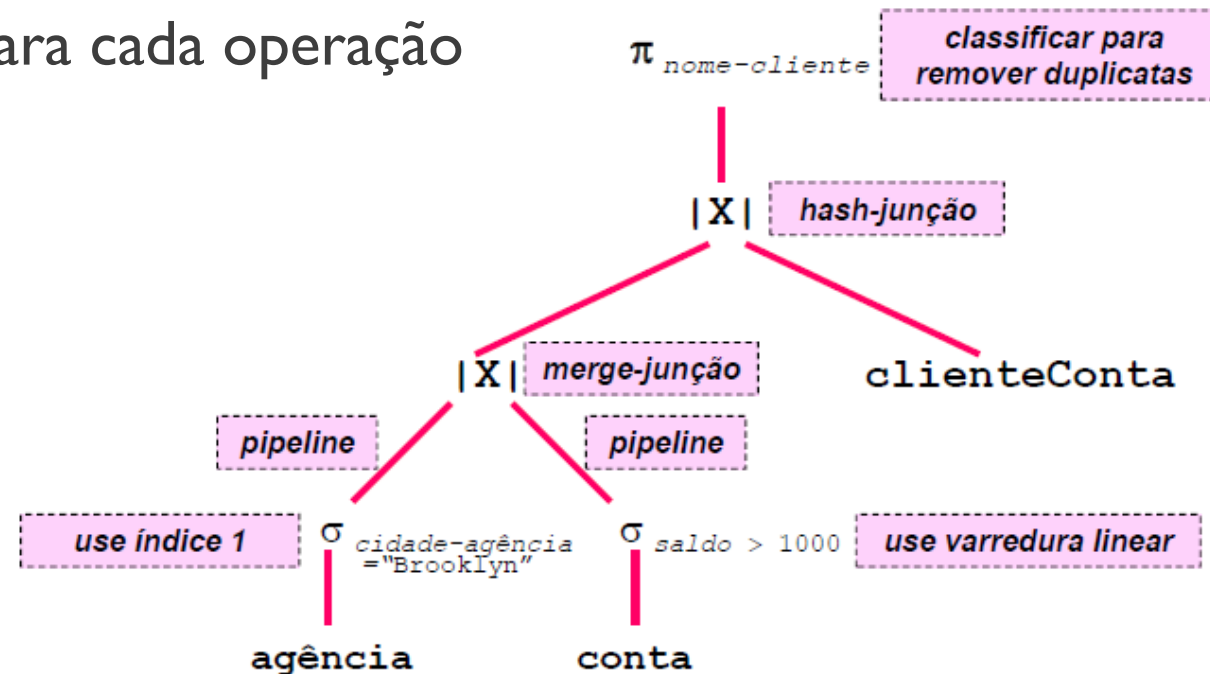
Qual o problema??

ENUMERAÇÃO DE EXPRESSÕES EQUIVALENTES

- Abordagem muito custosa em espaço e tempo
- Duas outras abordagens:
 - Geração do plano otimizado baseado nas regras de transformação
 - Abordagem especial para consultas com somente seleções, projeções e junções

ESCOLHA DO PLANO DE AVALIAÇÃO

- Como um plano de avaliação é escolhido?
 - Em função das estatísticas das relações de entrada.
 - Uma opção: algoritmo mais barato para cada operação
- Qual é o problema?
 - Micro x Macro



ESCOLHA DO PLANO DE AVALIAÇÃO

- Escolher o algoritmo mais barato para cada uma das operações pode não conduzir ao melhor plano total.
 - A junção merge pode ser mais custosa que a junção hash, mas pode fornecer uma saída ordenada que reduz o custo das operações seguintes.
 - A junção de laços aninhados pode fornecer facilidades para o pipeline
- Otimizadores de consultas práticos incorporam elementos das seguintes abordagens:
 - Busca todos os planos e escolhe o melhor usando o custo.
 - Usa heurísticas para escolher um plano.

OPÇÃO I – OTIMIZAÇÃO BASEADA EM CUSTO

- Gera faixa de planos de avaliação a partir de uma determinada consulta usando regras de equivalência
 - Escolhe o de menor custo
- Problema?
 - Quantidade de planos pode ser imensa!
 - Exemplo: achar a melhor ordenação de junções para $r_1 \mid X \mid r_2 \mid X \mid \dots \mid X \mid r_n$
 - Quantidade de ordens de junção diferentes:
 - Se $n = 5 \rightarrow 1.680$ ordens de junção de diferentes
 - Se $n = 7 \rightarrow 665.280$ ordens de junção de diferentes
 - Se $n = 10 \rightarrow$ número é maior que 17,6 bilhões!

$$\frac{(2(n-1))!}{(n-1)!}$$

OPÇÃO 2 - PROGRAMAÇÃO DINÂMICA

- Método para a construção de algoritmos para a resolução de problemas computacionais, em especial os de otimização combinatória.
- Aplicável a problemas no qual a solução ótima pode ser computada a partir da solução ótima previamente calculada e memorizada
- Evita recálculo de outros subproblemas que, sobrepostos, compõem o problema original.

PROGRAMAÇÃO DINÂMICA

- Ideia: armazenar resultados de cálculos para reutilizá-los
- Exemplo:
 - Calcular $(r1 \mid X \mid r2 \mid X \mid r3) \mid X \mid r4 \mid X \mid r5$
 - 12 ordens de junção para $(r1 \mid X \mid r2 \mid X \mid r3)$
 - Uma vez conhecida a melhor combinação, basta usá-la para o resultado com $\mid X \mid r4 \mid X \mid r5$
 - Em vez de 144 cálculos, executa-se $12 + 12$ cálculos

CUSTO DE PROGRAMAÇÃO DINÂMICA

- Com programação dinâmica a complexidade em tempo da otimização com árvores é $O(3^n)$.
- Com $n = 10$, este número ≈ 59000 no lugar de 17,6 bilhões ($O(n!)$)

OPÇÃO 3 - OTIMIZAÇÃO HEURÍSTICA

- Muitas vezes calcular o custo é muito caro
- Otimização heurística: usa regras heurísticas para transformar consultas, independentemente do custo
- Exemplos:
 - Execute operações de seleção assim que possível
 - Execute projeções antes de outras operações

OTIMIZAÇÃO HEURÍSTICA

- Decompor as seleções conjuntivas em uma sequência de operações de seleção simples (regra de equivalência 1).
- Mover as operações de seleção para a parte inferior da árvore de consultas para poder executá-las o mais cedo possível (regras de equiv. 2, 7a, 7b, 10).
- Executar primeiro as operações de seleção e junção que produzirão as relações de menor tamanho. (regra de equiv. 6 e 10).
- Trocar as operações de produto cartesiano seguidas por uma condição de seleção por operações de junção (regra de equiv. 4a).
- Decompor e mover para baixo da árvore o quanto possível as listas de atributos de projeção, criando novas projeções onde seja necessário (regras de equiv. 3, 8a, 11).

SGBDS E OTIMIZAÇÃO

- Maioria dos otimizadores dos SGBDs combina elementos de custo com elementos de heurísticas.
- Mesmo com o uso de heurísticas, a otimização baseada no custo da consulta impõe um overhead considerável.
- Este custo é mais do que compensado pela economia no tempo de execução da consulta, especificamente pela redução do número de acessos de discos lentos

BIBLIOGRAFIA

- ABRAHAM SILBERSCHATZ, HENRY F. KORTH, S. SUDARSHAN. Sistema de Banco de Dados. 6. Campus. 0. ISBN 9788535245356.
- ELMASRI, RAMEZ, SHAMKANT B. NAVATHE. Sistemas de banco de dados. Vol. 6. São Paulo: Pearson Addison Wesley, 2011.
- DATE, CHRISTHOPER J. Introdução a Sistemas de Bancos de Dados, 5ª. Edição. Campus, Rio de Janeiro (2004).

OBRIGADO E ATÉ A PRÓXIMA AULA!