

T1 - Implementação de Sistemas de Banco de Dados

Pedro E. Melha Lemos, Rafael Scotti Zanella

¹Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Faculdade de Informática -- Bacharelado em Ciência da Computação
Av. Ipiranga, 6681 – Bairro Partenon – CEP 90619-900 – Porto Alegre – RS – Brazil

{pedro.elpidio, rafael.scotti}@acad.pucrs.br

Resumo. *Este relatório apresenta comparação entre Árvores de resolução em álgebra relacional e o plano de resolução do Oracle SQL Developer sobre 3 consultas diferentes. Além disso, são feitas otimizações sobre estas consultas, o que muda o plano de resolução sobre elas.*

1. Introdução

Com o objetivo de extrair dados de um Banco de Dados, realiza-se consultas no mesmo. Essas consultas muitas vezes podem acontecer de forma demorada devido a grande quantidade de itens no banco ou um *hardware* lento ou simplesmente se as consultas não estão devidamente otimizadas. Tendo isso em vista, foram realizados alguns testes em um banco de dados utilizando o SGBD *Oracle SQL Developer*, da Oracle, onde foram feitas consultas com e sem otimização com a intenção de comparar seus resultados. Árvores de resolução em álgebra relacional também foram criadas para fazer a análise.

2. Primeira consulta

```
select NUMERO_GRUPO, NOME_GRUPO,  
       NUMERO_PESSOA, NOME_PESSOA,  
       TIPO_EMPENHO, DESCRICAO_EMPENHO,  
       NUMERO_EMPENHO, VALOR_EMPENHADO, DATA_DESPESA  
from PESSOAS1 natural join DESPESAS1  
      natural join GRUPOS  
      natural join TIPOS_EMPENHOS  
where NUMERO_GRUPO between 1 and 3  
      and NUMERO_PESSOA > 1000  
      and NUMERO_EMPENHO > 8000  
      and TIPO_EMPENHO between 5 and 8  
      and DATA_DESPESA between '01-OUT-10' and '1-OUT-11'  
order by VALOR_EMPENHADO;
```

2.1. Árvore de resolução em álgebra relacional

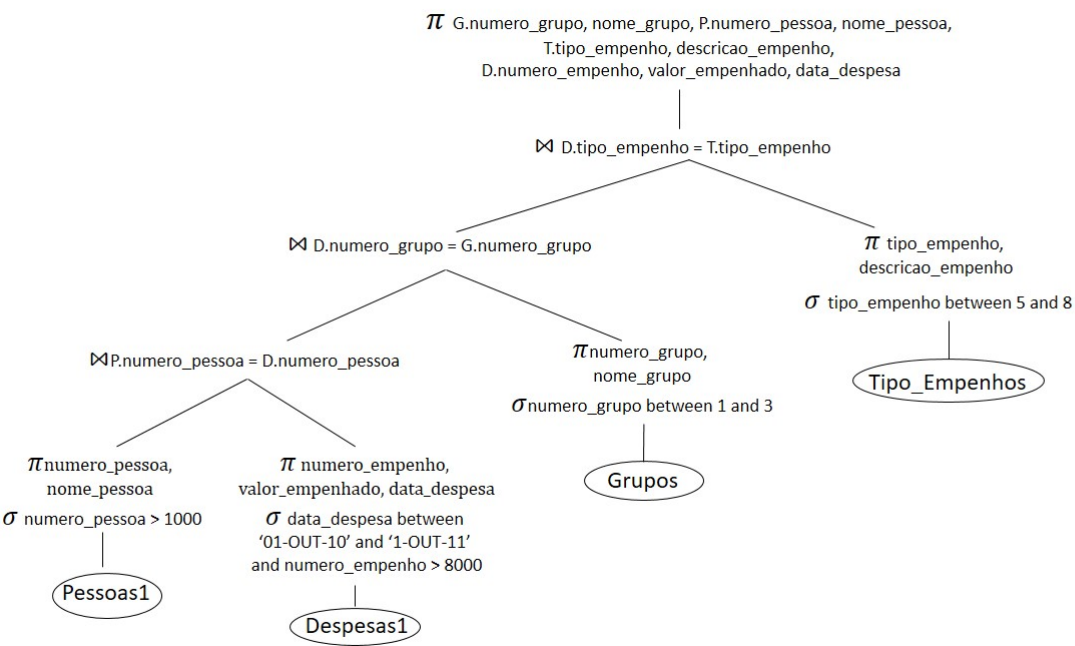


Figura 1. Árvore referente a primeira consulta

2.2. Amostra de execução da consulta sobre o banco de dados

	NUMERO_GRUPO	NUMERO_PESSOA	NOME_PESSOA	TIPO_EMPENHO	DESCRICAO_EMPENHO	NUMERO_EMPENHO
1	3	OUTROS SERVIÇOS DE TERCEIROS	JOSÉ GUSTAVO OLIVNYK	8	Coletivo c/Lista (atual)	17521
2	1	PESSOAL	IVONE MARIA DA SILVA	8	Coletivo c/Lista (atual)	17294
3	1	PESSOAL	BELINDA DA SILVEIRA	8	Coletivo c/Lista (atual)	8511
4	1	PESSOAL	BELINDA DA SILVEIRA	6	Pessoal	17256
5	1	PESSOAL	MARIA CELIA DA COSTA PEREIRA	6	Pessoal	9864
6	1	PESSOAL	JOÃO CARLOS MUNIZ SOARES	6	Pessoal	17277
7	1	PESSOAL	OLINDA MEIRELES DA SILVA	6	Pessoal	17270
8	1	PESSOAL	CARLOS MARTINS FERREIRA	6	Pessoal	17244
9	1	PESSOAL	DARCI DA SILVA ANTUNES	6	Pessoal	9784
10	3	OUTROS SERVIÇOS DE TERCEIROS	MARCELO OKABAYASHI	8	Coletivo c/Lista (atual)	19094

Figura 2. Amostra de execução da primeira consulta

2.3. Plano de resolução (Oracle SQL Developer)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				194
SORT		ORDER BY		194
FILTER				697
Filter Predicates				
TO_DATE('01-OUT-10') <= TO_DATE('01-OUT-11')				
HASH JOIN				193
Access Predicates				
PESSOAS1.NUMERO_PESSOA=DESPESAS1.NUMERO_PESSOA				
HASH JOIN				163
Access Predicates				
DESPESAS1.NUMERO_GRUPO=GRUPOS.NUMERO_GRUPO				
TABLE ACCESS	GRUPOS	FULL	4	3
Filter Predicates				
AND				
GRUPOS.NUMERO_GRUPO <= 3				
GRUPOS.NUMERO_GRUPO >= 1				
HASH JOIN				159
Access Predicates				
DESPESAS1.TIPO_EMPENHO=TIPOS_EMPENHOS.TIPO_EMPENHO				
TABLE ACCESS	TIPOS_EMPENHOS	FULL	4	3
Filter Predicates				
AND				
TIPOS_EMPENHOS.TIPO_EMPENHO <= 8				
TIPOS_EMPENHOS.TIPO_EMPENHO >= 5				
TABLE ACCESS	DESPESAS1	FULL	2343	156
Filter Predicates				
AND				
DESPESAS1.TIPO_EMPENHO <= 8				
DESPESAS1.TIPO_EMPENHO >= 5				
DESPESAS1.DATA_DESPESA <= '01-OUT-11'				
DESPESAS1.NUMERO_EMPENHO >= 8000				
DESPESAS1.NUMERO_GRUPO <= 3				
DESPESAS1.NUMERO_PESSOA >= 1000				
DESPESAS1.DATA_DESPESA >= '01-OUT-10'				
DESPESAS1.NUMERO_GRUPO >= 1				
TABLE ACCESS	PESSOAS1	FULL	10104	30
Filter Predicates				
PESSOAS1.NUMERO_PESSOA >= 1000				

Figura 3. Plano de resolução referente a primeira consulta

2.4. Comparação entre a árvore de resolução em Álgebra Relacional, com os planos de resolução do Oracle

Após realizar as consultas em SQL e implementar a árvore de resolução em álgebra relacional, percebe-se que assim como na árvore de resolução, o plano de resolução da Oracle faz primeiro uma seleção dos predicados *data_despesa*, onde a data esteja entre '01-OUT-10' e '01-OUT-11' e *numero_empenho* maior que 8000, da tabela *DESPESAS1* e, na tabela *PESSOAS1*, onde o *numero_pessoa* seja maior que 1000. Após realiza a junção dessas tabelas. A seguir, faz uma seleção onde *numero_grupo* da tabela *GRUPOS* esteja entre 1 e 3 e, após é feita a junção do resultado com a tabela *DESPESAS1*. A terceira e ultima junção é feita com as tabelas *DESPESAS1* e *TIPOS_EMPENHOS* após ter sido realizado uma seleção na tabela *TIPOS_EMPENHOS* onde utiliza o predicado *tipo_empenho* e este esteja entre 5 e 8.

2.5. Opinião sobre a otimização

Nessa primeira consulta, é feita uma criação de índice que otimiza o acesso aos registros. Como a coluna *DATA_DESPESA* é utilizada nas cláusulas *select* e *where* desta consulta, é então criado o índice *IX_DATA_DESPESA* sobre a tabela *DESPESAS1*. Com isso os acessos deveriam estar otimizados. No entanto, infelizmente no plano de resolução desta consulta com e sem a otimização não houve mudanças.

3. Segunda consulta

```
select NUMERO_PESSOA, NOME_PESSOA, VALOR_EMPENHADO
from PESSOAS1 natural join DESPESAS1
where VALOR_EMPENHADO >= (select avg(VALOR_EMPENHADO)
                           from DESPESAS1)
order by VALOR_EMPENHADO;
```

3.1. Árvore de resolução em álgebra relacional

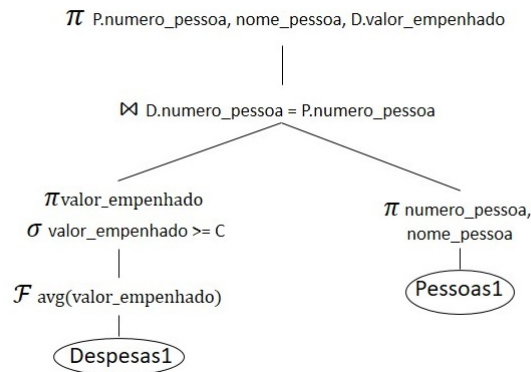


Figura 4. Árvore referente a segunda consulta

3.2. Amostra de execução da consulta sobre o banco de dados

	NUMERO_PESSOA	NOME_PESSOA	VALOR_EMPENHADO
1	4894	CBR CLINICA BEIRA RIO LTDA	45135
2	29035	TRANS SUL EMPRESA DE TRANSPORTES DE CARGAS LTDA	45143,1
3	3059	ASSOC DAS E DE T DE P DO S DE B E DA R M DE P ALEGRE (RS)	45167,6
4	3059	ASSOC DAS E DE T DE P DO S DE B E DA R M DE P ALEGRE (RS)	45186,9
5	10687	FOLHA DE PAGAMENTO	45201
6	10687	FOLHA DE PAGAMENTO	45206,2
7	28140	SULTRAUMA CLINICA DE TRAUMATO E ORTOPEDIA LTDA	45221,2
8	6123	COMPANHIA DE PROCES DE DADOS DO MUN DE P ALEGRE PROCENPA S/A	45254,5
9	6485	CRISTALIA PRODUTOS QUIMICOS FARMACEUTICOS LTDA	45300
10	4898	CCS SERVIÇOS TERCEIRIZADOS LTDA-ME	45328,7

Figura 5. Amostra de execução da segunda consulta

3.3. Plano de resolução (Oracle SQL Developer)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				343
SORT		ORDER BY		343
HASH JOIN				186
Access Predicates	PESSOAS1.NUMERO_PESSOA=DESPESAS1.NUMERO_PESSOA			
TABLE ACCESS	DESPESAS1	FULL		155
Filter Predicates	DESPESAS1.VALOR_EMPENHADO >= (SELECT AVG(VALOR_EMPENHADO) FROM DESPESAS1 DESPESAS1)			
SORT		AGGREGATE		1
TABLE ACCESS	DESPESAS1	FULL		155
TABLE ACCESS	PESSOAS1	FULL		30

Figura 6. Plano de resolução referente a segunda consulta

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				212
SORT		ORDER BY		212
HASH JOIN				184
Access Predicates	PESSOAS1.NUMERO_PESSOA=DESPESAS1.NUMERO_PESSOA			
TABLE ACCESS	DESPESAS1	BY INDEX ROWID		153
INDEX	IX_VALOR_EMPENHADO	RANGE SCAN		2
Filter Predicates	DESPESAS1.VALOR_EMPENHADO >= (SELECT AVG(VALOR_EMPENHADO) FROM DESPESAS1 DESPESAS1)			
SORT		AGGREGATE		1
INDEX	IX_VALOR_EMPENHADO	FAST FULL SCAN		27
TABLE ACCESS	PESSOAS1	FULL		30

Figura 7. Plano de resolução referente a segunda consulta utilizando índice

3.4. Comparação entre a árvore de resolução em Álgebra Relacional, com os planos de resolução do Oracle

Após analisar a árvore de resolução em álgebra relacional e comparar com o *resolution plan* do Oracle SQL Developer, percebe-se uma grande semelhança entre eles, tendo apenas um detalhe de diferença. Nota-se que no plano de resolução da Oracle não é mostrado como é feito tecnicamente a seleção da função agregada, enquanto que na árvore de álgebra relacional sabe-se que elas são divididas e processadas em duas consultas. Primeiro é feito a média (AVG) de *valor_empenhado* da tabela DESPESAS1 e, após, o resultado é passado para uma outra seleção que faz a comparação do maior, ou igual, *valor_empenhado* em relação a essa média.

A seguir é feita a junção das tabelas DESPESAS1 e PESSOAS1 e, então, o resultado (projeção das colunas *numero_pessoa*, *nome_pessoa* e *valor_empenhado*) é projetado.

3.5. Opinião sobre a otimização

Tal qual como na primeira consulta, nesta também é feita a criação de índice: IX_VALOR_EMPENHADO para otimizar o acesso aos registros devido o uso da coluna VALOR_EMPENHADO da tabela DESPESAS1 nas cláusulas *select* e *where*. Diferente do caso apresentado na consulta anterior, nesta consulta o plano de resolução apresentado pelo Oracle SQL Developer é consideravelmente diferente, já que este está considerando o índice criado para ordenação.

4. Terceira consulta

```
select NOME_RUBRICA, CODIGO_RUBRICA,  
       VALOR_EMPENHADO, VALOR_PAGO  
from RUBRICAS left join DESPESAS1  
    on RUBRICAS.NUMERO_RUBRICA = DESPESAS1.NUMERO_RUBRICA  
order by CODIGO_RUBRICA;
```

4.1. Árvore de resolução em álgebra relacional

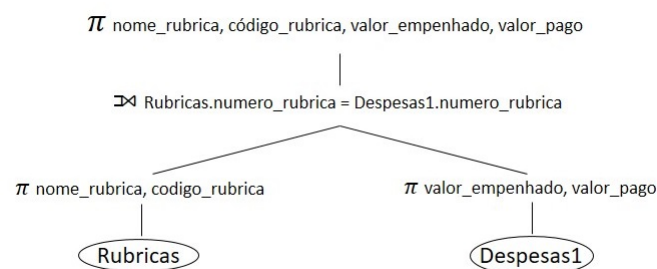


Figura 8. Árvore referente a terceira consulta

4.2. Amostra de execução da consulta sobre o banco de dados

	NOME_RUBRICA	CODIGO_RUBRICA	VALOR_EMPENHADO	VALOR_PAGO
1	PROVENTOS - PESSOAL CIVIL	319001010000	2722,4	2722,4
2	PROVENTOS - PESSOAL CIVIL	319001010000	2722,4	2722,4
3	PROVENTOS - PESSOAL CIVIL	319001010000	189,8	189,8
4	PROVENTOS - PESSOAL CIVIL	319001010000	4618662,8	4618662,8
5	PROVENTOS - PESSOAL CIVIL	319001010000	1218338,9	1218338,9
6	PROVENTOS - PESSOAL CIVIL	319001010000	1030,8	1030,8
7	PROVENTOS - PESSOAL CIVIL	319001010000	251232,2	251232,2
8	PROVENTOS - PESSOAL CIVIL	319001010000	336809,3	336809,3
9	PROVENTOS - PESSOAL CIVIL	319001010000	275950,8	275950,8
10	PROVENTOS - PESSOAL CIVIL	319001010000	13184,9	13184,9

Figura 9. Amostra de execução da terceira consulta

4.3. Plano de resolução (Oracle SQL Developer)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				44265
SORT		ORDER BY		1094
HASH JOIN		OUTER		44265
Access Predicates				
RUBRICAS.NUMERO_RUBRICA=DESPESAS1.NUMERO_RUBRICA(+)				
TABLE ACCESS	RUBRICAS	FULL		454
TABLE ACCESS	DESPESAS1	FULL		44265

Figura 10. Plano de resolução da terceira consulta

```

select NOME_RUBRICA, CODIGO_RUBRICA,
       VALOR_EMPENHADO, VALOR_PAGO
from RUBRICAS inner join DESPESAS1
    on RUBRICAS.NUMERO_RUBRICA = DESPESAS1.NUMERO_RUBRICA
union
select NOME_RUBRICA, CODIGO_RUBRICA,
       NULL, NULL
from RUBRICAS
where not exists (
    select * from DESPESAS1
    where RUBRICAS.NUMERO_RUBRICA = DESPESAS1.NUMERO_RUBRICA
)
order by CODIGO_RUBRICA;

```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				44325
UNION-ALL				2098
SORT		UNIQUE		1255
HASH JOIN				44265
Access Predicates				
RUBRICAS.NUMERO_RUBRICA=DESPESAS1.NUMERO_RUBRICA				
TABLE ACCESS	RUBRICAS	FULL		454
TABLE ACCESS	DESPESAS1	FULL		44265
HASH JOIN		ANTI		60
Access Predicates				
RUBRICAS.NUMERO_RUBRICA=DESPESAS1.NUMERO_RUBRICA				
TABLE ACCESS	RUBRICAS	FULL		454
TABLE ACCESS	DESPESAS1	FULL		44265

Figura 11. Plano de resolução referente a terceira consulta após otimização

4.4. Comparação entre a árvore de resolução em Álgebra Relacional com os planos de resolução do Oracle

Observando a árvore de resolução em álgebra relacional, percebe-se que foi realizada uma junção externa (*left join*) nas tabelas RUBRICAS E DESPESAS1. Esse tipo de junção pega os elementos da tabela RUBRICAS e faz uma união com elementos em comum com a tabela DESPESAS1.

4.5. Opinião sobre a otimização

Na terceira consulta a otimização garante tem uma abordagem diferente: ao invés de criar índices para otimizar os acessos aos registros, a otimização tem por finalidade fazer a junção externa não explicitamente. Como a operação de junção externa é consideravelmente custosa, a solução sugerida é que fosse realizada uma junção interna entre as tabelas RUBRICAS e DESPESAS1 e a respectiva união com os registros que não são em comum.

5. Conclusão

Após finalizar as análises, conclui-se que apenas adicionando índices em uma consulta ou alterando a forma que ela é feita, obtém-se resultados consideravelmente superiores. E tendo em vista banco de dados grandes, a otimização faz uma diferença muito importante.