



RENAN HENRIQUE GOMES DAMAZIO ASSUNÇÃO RA 21038114  
ALEX ARANTES GONÇALVES RA 21011214

## **RELATÓRIO 7**

Tópicos Emergentes em Bancos de Dados

## QUESTÃO 1

Afim de avaliar o impacto de materialized views em relações propostas para a questão 1, as relações RelA, RelB, RelC e RelD foram criadas de acordo com as especificações esquemáticas propostas. Após a criação das relações, foram analisados dois possíveis transações denominadas T1 e T2 para análise de desempenho quanto a transação utilizando a união das mesmas com as suas respectivas materialized views. O código para criação das relações, estrutura das transações T1 e T2, bem como as análises utilizadas neste relatório podem ser obtidas no arquivo em anexo nomeado Q1.txt.

A princípio foram criadas duas Materialized views completas das transações T1 e T2. Como era esperado, o tempo de execução caiu drasticamente conforme relacionado na tabela 1

**Tempos de execução View vs Materialized View**

Transição	ViewTempo execução (ms)	Mat.View Tempo execução (ms)
T1	1035	14
T2	13411	66

Tabela 1

Se a necessidade do uso das transações T1 e T2 é o de acesso aos dados atualizados até o dia anterior( $D - 1$ ), esta solução seria viável, no entanto se a necessidade das transições é o de acesso aos dados mais recentes, a materialized view completa não compensaria, porque a atualização requer tempo tão considerável quanto ao acesso aos dados por meio das joins das transações.

Materialized views parciais também foram consideradas a saber:

- Por conta da dependência da relação RelC com RelD, não foi possível criar uma MV das relações RelA e RelC.
- MV de natural join de RelC e RelB e posterior junção desta MV com as relações RelA e RelD foi possível, no entanto não houve considerável melhora de desempenho.
- Embora as transações se utilizam das mesmas junções, as mesmas tentativas de realizar MV parciais para transação 2 foi realizada porque a mesma apresenta uma cláusula "where" que resulta em menos tuplas numa MV, no entanto não houve melhora de desempenho.

Levando se em consideração as materialized views completas das duas transações e as expectativas de crescimento de cada relação, a expectativa do crescimento de páginas para os próximos 6 meses foi estimada conforme gráfico G1.

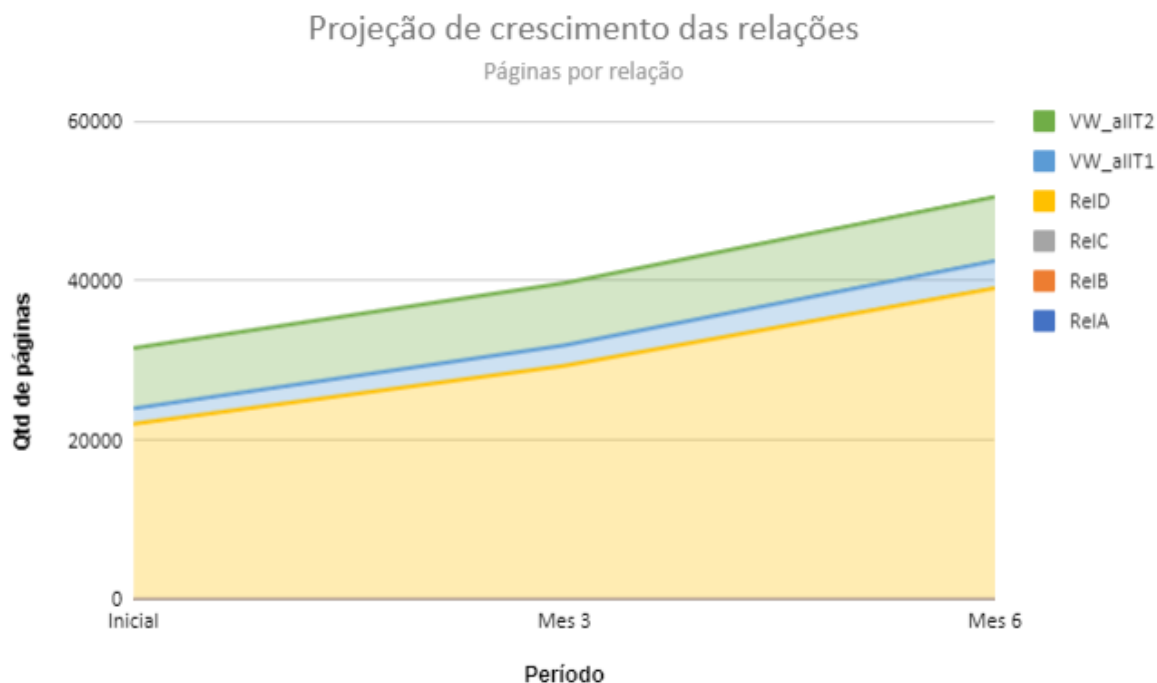


Gráfico G1 - expectativa de crescimento de páginas por relação para os próximos 6 meses

## Conclusão

- i. O uso de materialized views depende principalmente da criticidade dos dados e da frequência de atualização destes dados;
- ii. Como nenhum dado foi apresentado sobre a criticidade dos dados, para esta questão foi considerado que os dados não eram críticas e que se poderia atualizado a MV 1 vez ao dia;
- iii. Utilização de MV gera a necessidade da criação de estruturas para mante-las atualizadas, o que gera, dentre outras coisas, mais esforço de evolução;
- iv. O uso de MV para a resolução deste exercício gerou um aumento significativo no número de pages nos período de 6 meses, porém, consideramos o aumento pouco significativo quando levado em conta os benefícios para o desempenho da transação, além do fato de que atualmente espaço em disco não é um grande problema.

## QUESTÃO 2

Para análise da situação proposta na questão 2 optou-se por utilizar uma transação considerando o pior caso, onde todos os atributos da relação emp são utilizados. Dessa forma foi possível mensurar o tempo de resposta para efeitos de comparação. Todos os scripts utilizados se encontram no arquivo Q2.txt

```
SELECT emp.*, depto.nmdepto, sum(1) FROM emp NATURAL JOIN depto LEFT  
JOIN depte using(codemp) GROUP BY codemp, depto.nmdepto
```

A média do tempo de execução desta transição foi de 1044 ms.

Foi analisada a princípio a criação de uma view materializada que contivesse todos os dados desta transação, cujos scripts não foram utilizados na produção mas seguem no arquivo Q2MV.txt. Essa hipótese foi descartada devido ao alto número de atualizações necessárias, por a transação ser crítica. Esse descarte se deu principalmente devido ao alto número de atualizações da tabela emp e também pelo fato de não ser o objetivo da questão avaliar MV e sim desnormalização.

Ao se criar uma relação desnormalizada com redundância em *nmemp* nomeada *emp2* e novamente uma transição que retorna os mesmos resultados da transação padrão, houve um ganho considerável de performance. A média do tempo de transação caiu para 652 ms.

Uma nova relação desnormalizada com redundância foi criada, desta vez considerando *nmemp* além da função agregada de soma de número de dependentes por empregado nomeada como *emp3*. O ganho de performance foi ainda maior reduzindo a média do tempo da transição para 88 ms

A queda tempo de retorno da transição em questão pode ser verificado pelo gráfico G2

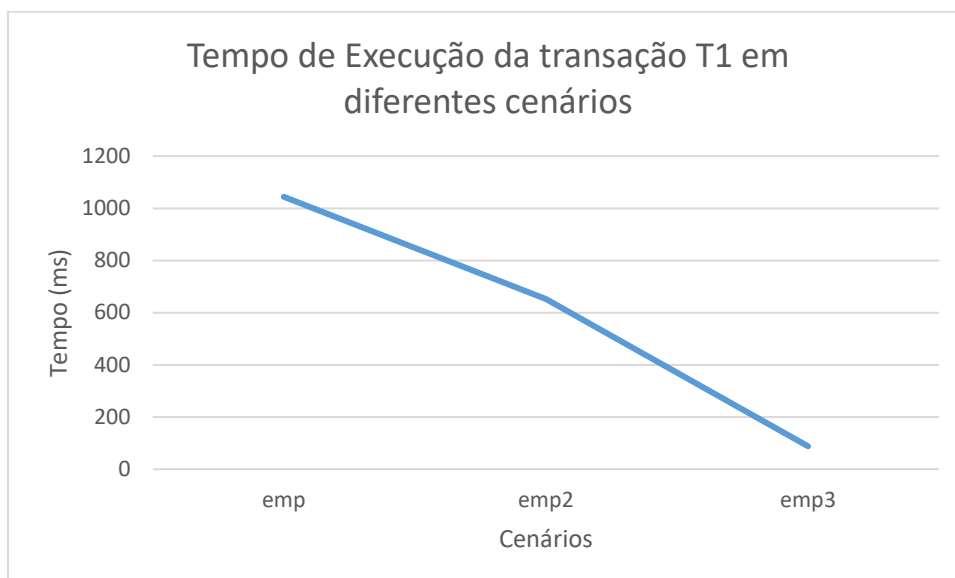


Gráfico G2 - Tempo de execução das visões desnormalizadas com redundância.

A performance do tempo de inserção de dados também foi mensurada. Foram consideradas para esta análise dois casos

- Caso A: atualização do nome do departamento, a qual acontece 1 vez por mês

Para este caso foi utilizada para análise a Transação:

```
update emp3 set nmdepto = 'novoDepto' where codemp <= 10000
```

A qual foi executada em 132ms

- Caso B: atualização do número de dependentes, segundos os dados foi calculado que acontece cerca de 98 vezes no mês, sempre um novo dependente é inserido, sendo o referido empregado atualizado

Para este caso foi utilizada para análise a Transação:

```
update emp3 set deptes = 1 where codemp = 4
```

A qual foi executada em menos de 1 ms

Desta forma obtem-se o valor de que o tempo médio de atualização da relação desnormalizada emp3 é de cerca de 230ms ao mês. Para manter a relação atualizada deve então criar 2 trigger: uma trigger em depto que atualiza o nome do departamento de todos os funcionários sempre um departamento muda de nome, conforme a

primeira query e outra na tabela depte que atualiza o número de dependentes de um empregado sempre que um dependente é adicionado.

### **Conclusão**

- i. Views materialização não da indicadas para esse caso
- ii. A desnormalização com redundância apenas de nmdepto gerou grandes melhorias no desempenho da transação e será absorvida para o projeto.
- iii. A desnormalização com redundância do sum de dependentes por funcionário também gerou grandes melhorias no projeto, onde a soma de ambas as redundância diminuiu o tempo médio de execução da transação de testes de em cerca de 12x
- iv. Com a utilização de redundância será necessário um maior esforço para manter os dados redundantes sempre atualizados, o que será feito através de triggers, conforme exposto no relatório.