

Qlik Brasil: Melhores Práticas – Concatenar tabelas sem perder a leitura otimizada

Postado por [Yuri Nicolett](#) 24/10/2015

Fala pessoal, beleza?

Acredito que muitos aqui já elaboraram um belíssimo código seguinte todos a maioria das melhores práticas em script/modelagem e precisaram concatenar (Concatenate) duas ou mais tabelas (fatos ou não), porém essa concatenação esta demorando muito tempo para ocorrer. Já passaram por isso?

Esse artigo explicará como realizar a concatenação de tabelas mantendo uma leitura otimizada. Vamos lá?

Lembram o que é uma leitura otimizada? Eu falei disso neste post: [Melhores Práticas – QVD Optimized](#).

Em resumo, um QVD mantém a leitura otimizada somente quando realizamos essas ações em sua leitura:

- Renomear campos
- DISTINCT
- Omitir Campos (Omit)
- Where Exists Simples

IMPORTANTE: A leitura otimizada é realizada apenas na leitura de um QVD, esqueça o LOAD RESIDENT.

O que é Concatenar

O conceito de concatenar é um termo utilizado para unir duas coisas. Em nosso caso "coisa" significa tabela, ou seja, unir duas tabelas. Para entender melhor como realmente o concatenar funciona, vamos a um exemplo:

Exemplo

Tenho duas tabelas: Vendas e População

ANOMES	VENDEDOR	VALOR
201501	YURI	100
201502	YURI	150
201503	YURI	200
201501	PAULA	80
201502	PAULA	100
201503	PAULA	300

ANOMES	CIDADE	POPULAÇÃO
201501	RIO CLARO	10
201502	RIO CLARO	20
201503	RIO CLARO	30
201504	RIO CLARO	40

Agora vou concatenar as duas tabelas:

ANOMES	VENDEDOR	VALOR	CIDADE	POPULAÇÃO
201501	YURI	100		
201502	YURI	150		
201503	YURI	200		
201501	PAULA	80		
201502	PAULA	100		
201503	PAULA	300		
201501			RIO CLARO	10
201502			RIO CLARO	20
201503			RIO CLARO	30
201504			RIO CLARO	40

Note que basicamente uma tabela foi colocada abaixo da outra. Esse é o resultado quando realizamos uma concatenação das tabelas.

OBSERVAÇÃO: As células em branco ficarão com o valor nulo no qlikview.

A sintaxe

Entendemos o resultado de uma concatenação de tabelas, mas como deve ser a sintaxe para fazer isso? Basicamente o comando Concatenate deve ser utilizado antes do LOAD de uma tabela. Vamos usar o exemplo das tabelas de Vendas e População para aprender:

Vendas:

```
LOAD
    ANOMES,
    VENDEDOR,
    VALOR
FROM Exemplo01.xlsx (ooxml, embedded labels, table is Exemplo);
```

CONCATENATE

População:

```
LOAD
    ANOMES,
    CIDADE,
    POPULAÇÃO
FROM Exemplo01.xlsx (ooxml, embedded labels, table is Exemplo);
```

Desta forma, o QlikView vai interpretar que a tabela População vai concatenar os seus dados com a tabela Vendas, pois o script de leitura da tabela Vendas está logo acima da tabela População. Caso a concatenação que deseja fazer não seja com a tabela acima, você pode simplesmente criar uma referência no Concatenate, dessa forma:

Vendas:

```
LOAD
    ANOMES,
    VENDEDOR,
    VALOR
FROM Exemplo01.xlsx (ooxml, embedded labels, table is Exemplo);
```

```
CONCATENATE (Vendas)
```

População:

```
LOAD
    ANOMES,
    CIDADE,
    POPULAÇÃO
FROM Exemplo01.xlsx (ooxml, embedded labels, table is Exemplo);
```

OBSERVAÇÃO: Como sabemos a concatenação resultará em apenas uma tabela, desta forma o nome da tabela resultante será a tabela que receberá os dados da outra, em nosso exemplo a tabela final ficará com o nome de Vendas. Beleza! Agora eu entendi o que é Concatenar, mas como fazê-la de forma otimizada?

Manter a leitura otimizada

As formas de Concatenar mantendo a leitura otimizada são:

- AutoConcatenate
- Mantendo os campos existentes e adicionando (escadinha)

AutoConcatenate

O AutoConcatenate é quando o QlikView identifica duas ou mais tabelas que possuem, exatamente, as mesmas colunas e faz a concatenação automática delas, ou seja, você não precisa utilizar o comando Concatenate. Veja a imagem para entender melhor

Tabela 1	A	B	C	D	E	F
Tabela 2	A	B	C	D	E	F
Tabela 3	A	B	C	D	E	F
Tabela 4	A	B	C	D	E	F
Tabela 5	A	B	C	D	E	F

Todas as tabelas possuem as mesmas colunas, são elas: A, B, C, D, E e F. Se carregarmos essas cinco tabelas sem utilizar o Concatenate, o QlikView utilizará o conceito de AutoConcatenate e concatenará as cinco tabelas de maneira otimizada. Veja que o código está sem o comando Concatenate:

```
[Tabela 1]:  
LOAD  
    A, B, C, D, E, F  
FROM Tabela1.qvd (qvd);
```

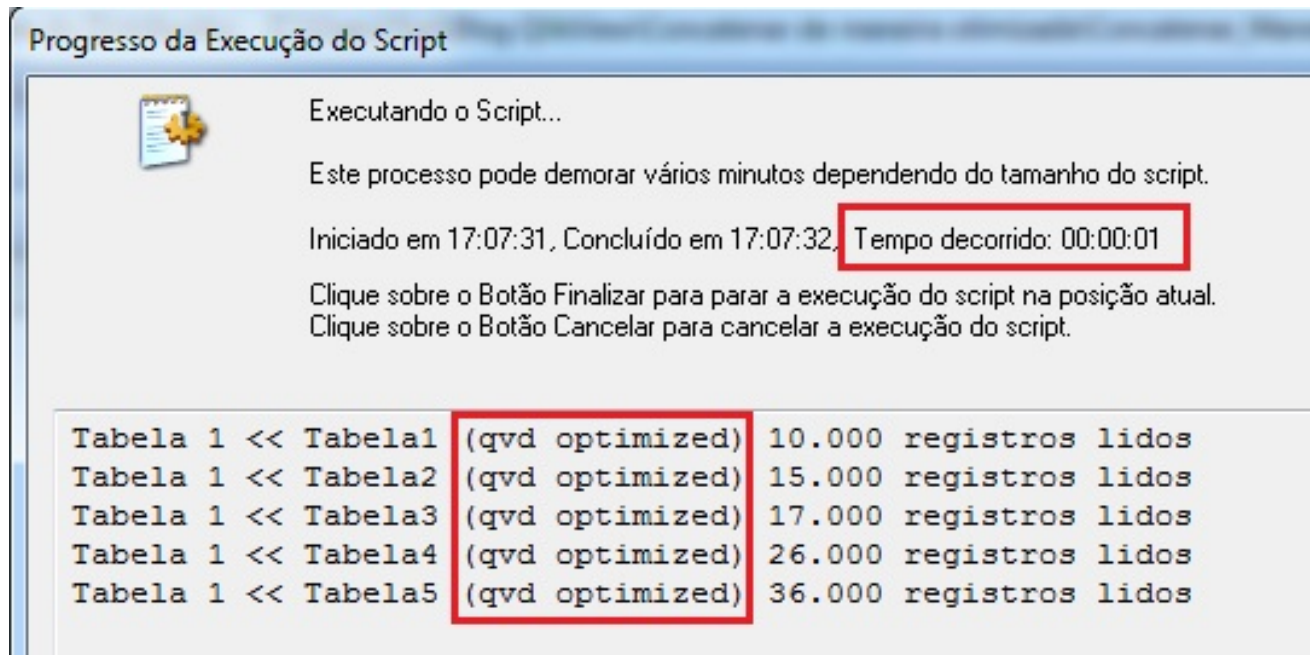
```
[Tabela 2]:  
LOAD  
    A, B, C, D, E, F  
FROM Tabela2.qvd (qvd);
```

```
[Tabela 3]:  
LOAD  
    A, B, C, D, E, F  
FROM Tabela3.qvd (qvd);
```

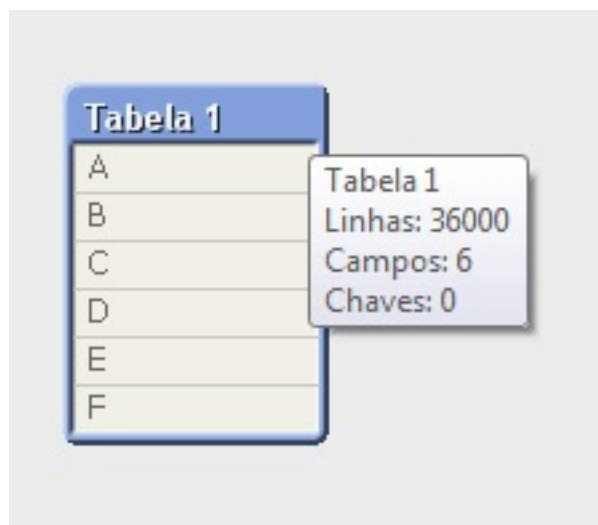
```
[Tabela 4]:  
LOAD  
    A, B, C, D, E, F  
FROM Tabela4.qvd (qvd);
```

```
[Tabela 5]:  
LOAD  
    A, B, C, D, E, F  
FROM Tabela5.qvd (qvd);
```

Veja a execução do script mantendo a leitura otimizada:



Resultado das cinco tabelas concatenadas:



PS: Lembrem-se que o nome que a tabela concatenada irá assumir é o nome da primeira tabela que foi carregada, no exemplo: Tabela 1.Yuri, essa eu já sabia! facin!!

Mantendo os campos existentes e adicionando

Manter os campos existentes e adicionando? Ahn?Esse caso é o mais interessante do comportamento do QlikView, e eu gosto de chama-lo de escadinha, pois tabelas que possuem alguns campos em comum podem sim fazer a concatenação de maneira otimizada.Vamos ao exemplo e depois explico melhor!

Tabela 1	A	B				
Tabela 2	A	B	C			
Tabela 3	A	B	C	D		
Tabela 4	A	B	C	D	E	
Tabela 5	A	B	C	D	E	F

PS: Enxergaram a escadinha? rsrs

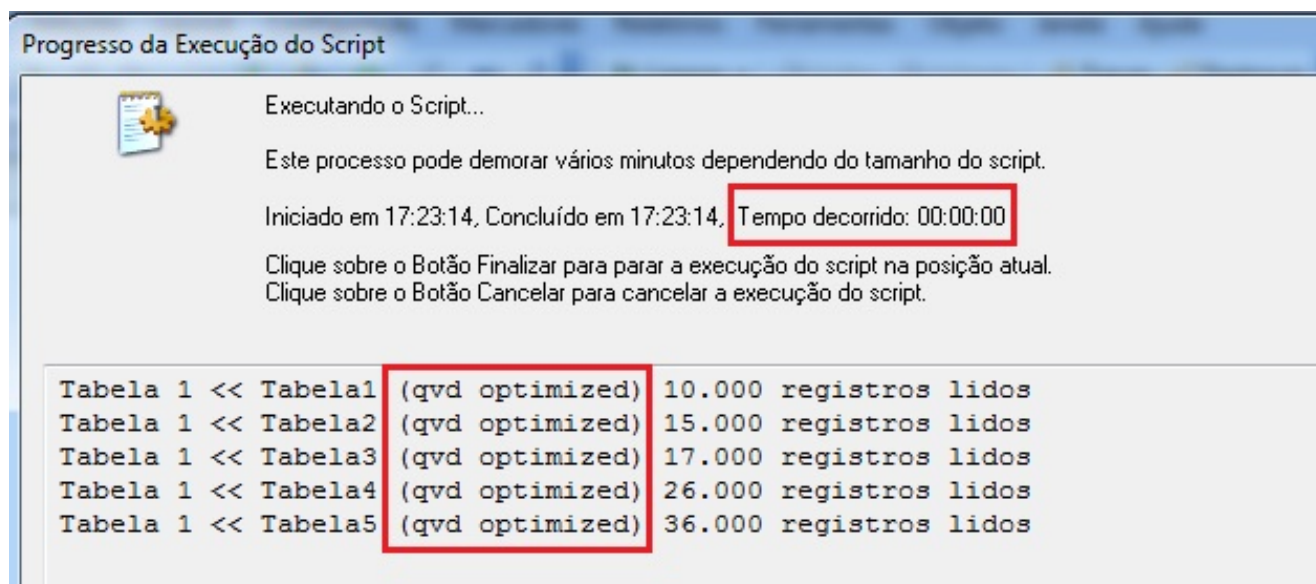
Vamos entender

- A minha primeira tabela possui as colunas A e B
- A segunda tabela precisa conter as mesmas colunas para poder adicionar outras, no caso precisa conter as colunas A e B para adicionar a coluna C.
- A terceira tabela precisa conter todos os campos que a tabela resultante (Concatenação da Tabela 1 com a Tabela 2) possui que são as colunas A, B e C (A coluna C originou na concatenação da Tabela 2) para poder adicionar a coluna D.
- A quarta tabela precisa conter todos os campos que a tabela resultante (Concatenação da Tabela 1 com a Tabela 2 e a Tabela 3) possui que são as colunas A, B, C e D (A coluna C originou na concatenação da Tabela 2 e a coluna D originou na concatenação com a Tabela 3) para poder adicionar a coluna E.
- E assim vai..... rsrs !!

Nesta caso o código precisa utilizar o comando Concatenate, veja como ficou:

```
[Tabela 1]:  
LOAD A, B FROM Tabela1.qvd (qvd);  
  
CONCATENATE  
  
[Tabela 2]:  
LOAD A, B, C FROM Tabela2.qvd (qvd);  
  
CONCATENATE  
  
[Tabela 3]:  
LOAD A, B, C, D FROM Tabela3.qvd (qvd);  
  
CONCATENATE  
  
[Tabela 4]:  
LOAD A, B, C, D, E FROM Tabela4.qvd (qvd);  
  
CONCATENATE  
  
[Tabela 5]:  
LOAD A, B, C, D, E, F FROM Tabela5.qvd (qvd);
```

Veja a execução do script mantendo a leitura otimizada:



Progresso da Execução do Script

Executando o Script...

Este processo pode demorar vários minutos dependendo do tamanho do script.

Iniciado em 17:23:14, Concluído em 17:23:14, Tempo decorrido: 00:00:00

Clique sobre o Botão Finalizar para parar a execução do script na posição atual.
Clique sobre o Botão Cancelar para cancelar a execução do script.

Tabela 1	<<	Tabela1	(qvd optimized)	10.000	registros	lidos
Tabela 1	<<	Tabela2	(qvd optimized)	15.000	registros	lidos
Tabela 1	<<	Tabela3	(qvd optimized)	17.000	registros	lidos
Tabela 1	<<	Tabela4	(qvd optimized)	26.000	registros	lidos
Tabela 1	<<	Tabela5	(qvd optimized)	36.000	registros	lidos

Resultado das cinco tabelas concatenadas:

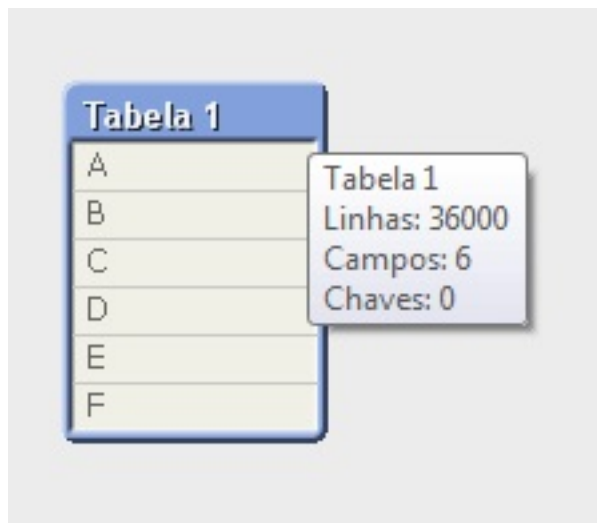


Tabela 1
A
B
C
D
E
F

Tabela 1
Linhas: 36000
Campos: 6
Chaves: 0

Pô Yuri, mas nem sempre vamos ter tabelas no modelo de escadinha. Ajuda aí!! (rsrsrs)
Vamos para um exemplo mais prático

Cenário Real

Imaginem que temos que concatenar duas tabelas, Tabela 1 e Tabela 2, sendo que:

Tabela 1 possui as colunas: A, B, C e D.

Tabela 2 possui as colunas: A, B, D, F e G

Pensando Alto: As duas tabelas possuem as colunas A, B e D, porém a tabela 2 não possui a coluna C e ainda possui as colunas F e G para adicionar... é... já era a escadinha, exemplo deu certo! rsrs

Vamos lá!

Primeiro passo

Carregar as duas tabelas

```
[Tabela 1]:  
LOAD A, B, C, D FROM Tabela1.qvd (qvd);
```

```
[Tabela 2]:  
LOAD A, B, D, F, G FROM Tabela2.qvd (qvd);
```

Segundo passo

Criar a coluna C na tabela 2.

Para criar esse registro de maneira otimizada, podemos criar uma tabela sem registro com um valor nulo sendo atribuído com o nome de C.

```
CONCATENATE ([Tabela 2])
```

```
LOAD  
    Null() as C  
AutoGenerate(0);
```

Terceiro passo

Gerar um novo QVD da Tabela 2 agora possuindo as colunas A, B, C, D, F e G.

```
STORE [Tabela 2] INTO Tabela2.qvd (qvd);  
DROP TABLE [Tabela 2];
```

Quarto passo

Carregar o novo QVD da Tabela 2 e Concatenar com a Tabela 1.

```
CONCATENATE ([Tabela 1])
```

```
LOAD A, B, C, D, F, G FROM Tabela2.qvd (qvd);
```

Entendendo

```
[Tabela 1]:  
LOAD A, B, C, D FROM Tabela1.qvd (qvd);
```

✓ **Leitura Otimizada**

```
[Tabela 2]:  
LOAD A, B, D, F, G FROM Tabela2.qvd (qvd);
```

✓ **Leitura Otimizada**

```
CONCATENATE ([Tabela 2])
```

```
LOAD  
  Null() as C  
AutoGenerate(0);
```

} **Leitura não otimizada
de um registro**

```
STORE [Tabela 2] INTO Tabela2.qvd (qvd);  
DROP TABLE [Tabela 2];
```

Gravo a tabela em QVD

```
CONCATENATE ([Tabela 1])
```

```
LOAD A, B, C, D, F, G FROM Tabela2.qvd (qvd);
```

✓ **Leitura Otimizada**

Depois de entender como é o comportamento do QlikView na concatenação de tabelas, o "pulo do gato" foi adicionar a coluna C na Tabela 2 através de um AutoGenerate.

Veja a execução do script mantendo a leitura otimizada:

Progresso da Execução do Script

Executando o Script...

Este processo pode demorar vários minutos dependendo do tamanho do script.

Iniciado em 17:47:11, Concluído em 17:47:12, Tempo decorrido: 00:00:01

Clique sobre o Botão Finalizar para parar a execução do script na posição atual.
Clique sobre o Botão Cancelar para cancelar a execução do script.

Tabela 1 << Tabela1	(qvd optimized)	10.000 registros lidos
Tabela 2 << Tabela2	(qvd optimized)	5.000 registros lidos
Tabela 2 << AUTOGENERATE(0)	5.000 registros lidos	Criando coluna C
Tabela 1 << Tabela2	(qvd optimized)	15.000 registros lidos

Exemplo para Download

A aplicação de exemplo esta disponível para download [aqui!](#)

Conclusão

Isso pode parecer bobo e talvez desnecessário, mas te garanto que em projetos grandes essas manhas vão ajudar bastante para manter a performance de tempo de recarga das aplicações.

E não poderia deixar de falar que é sempre importante aprendermos sobre melhores práticas e entendermos o comportamento do QlikView para determinadas ações.

É isso pessoal!

Até a próxima semana!!

9 Visualizações Tags: [concatenate](#), [deployment_best_practices](#), [concatenar](#), [dicas](#), [qvd optimized](#), [best practices](#), [melhores práticas](#), [leitura otimizada](#), [concatenar com leitura otimizada](#), [juntar tabelas](#), [dicas de desenvolvimento](#), [concatenate qvd optimized](#)

Não há comentários neste post