



DICA TÉCNICA 02

Colunas Calculadas

Sumário

1.	INTRODUÇÃO	3
2.	CRIANDO UMA COLUNA CALCULADA.....	3
3.	SINTAXE	5
4.	APLICAÇÕES COMUNS DE COLUNAS CALCULADAS	6
4.01	UTILIZANDO OPERADORES ARITMÉTICOS	6
4.02	CONCATENANDO TEXTO.....	6
4.03	IDENTIFICAR PARTE DO CONTEÚDO DE UM CAMPO TEXTO/ALFANUMÉRICO (EQUIVALENTE AO LIKE)	6
4.04	SEPARANDO PARTE DE UM CAMPO TEXTO/ALFANUMÉRICO	6
4.05	CONVERTENDO TIPOS DE DADOS.....	7
4.06	UTILIZANDO FUNÇÕES MATEMÁTICAS PRONTAS	7
4.07	UTILIZANDO IF-ELSE	7
4.08	COMPODO FÓRMULAS COM E / OU.....	8
4.09	INSERÇÃO DE CONSTANTES COM CASAS DECIMAIS	8
4.10	CALCULAR DIFERENÇA DE DIAS ENTRE DATAS	8
4.11	CALCULAR DIFERENÇA DE DIAS ÚTEIS ENTRE DATAS	8
4.12	DIFERENÇA DE DIAS ENTRE UMA DATA E A DATA ATUAL	8
4.13	DEIXAR CAMPO NUMÉRICO SEM CONTEÚDO NA AUSÊNCIA DE UMA DATA	8
4.14	SOMAR/SUBTRAIR N DIAS DE UMA DATA	8
4.15	IDENTIFICAR MÊS ANTERIOR.....	8
4.16	EXTRAIR DIA DA DATA ATUAL	8
4.17	EXTRAIR MÊS DA DATA ATUAL	8
4.18	CALCULAR VALOR DO MÊS ATUAL.....	8
4.19	CALCULAR VALOR DO MÊS ANTERIOR	9
4.20	TRANSFORMAR HORA, MINUTO E SEGUNDO (FORMATO HH:MM:SS) EM MINUTOS (E DECIMAL)	9
4.21	TRANSFORMAR HORA E MINUTO (FORMATO HH:MM) EM MINUTOS (E DECIMAL).....	9
4.22	TRANSFORMAR HORA E MINUTO (FORMATO HH:MM) EM HORAS (E DECIMAL)	9
5	FAQ	9
	APÊNDICE 1 – MEMBROS DA CLASSE MATH.....	10

1. Introdução

No SuperBI, as colunas calculadas são utilizadas para gerar dimensões ou métricas com base em cálculos onde as variáveis são outras dimensões ou métricas do cubo.

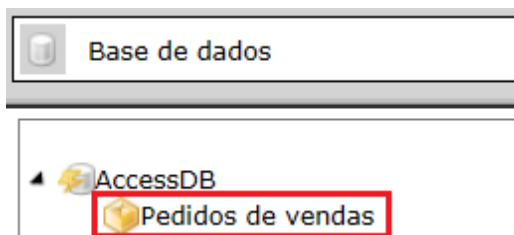
Ao ser criada, cada coluna calculada passa a fazer parte do cubo que lhe deu origem na condição de dimensão ou métrica, como qualquer outra coluna, o que permite seu uso em objetos e até mesmo como variáveis em novas colunas calculadas.

Exceções:

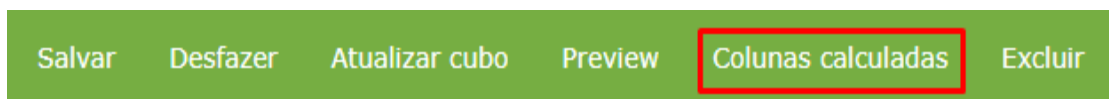
- colunas calculadas com sumarização "Formula" ou "Distinct count" não ficam disponíveis em filtros.
- colunas calculadas com sumarização "Soma", "Média" e "Nada" não podem usar outra coluna calculada dentro da fórmula.
- colunas calculadas com sumarização "Formula" não podem usar colunas calculadas com sumarização "Formula" ou "Nada" dentro da fórmula.

2. Criando uma coluna calculada

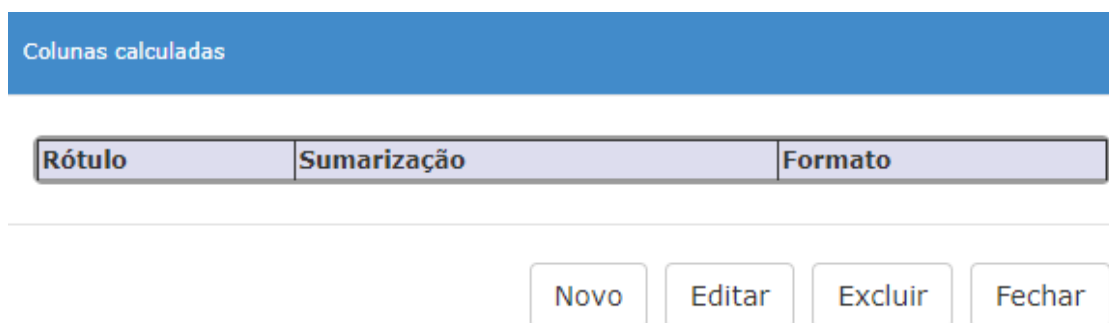
Para criar uma coluna calculada, basta selecionar um cubo no módulo de base de dados



e clicar no menu *Colunas calculadas*.



Com isso, será aberta a tela de manutenção de colunas calculadas para o cubo selecionado.



Clicando no botão "Novo", abre-se a tela "Edita coluna calculada" com as respectivas propriedades.

Rótulo:*

Descrição:

Formato:

Sumarização:

Rótulo (de preenchimento obrigatório): nome a ser atribuído à nova coluna.

Descrição: de preenchimento opcional, para documentar algum detalhe da coluna.

Formato: aplicável somente a colunas numéricas.

Sumarização: combo-box com as seguintes opções:

- . Nada - se aplica a fórmulas que envolvem colunas alfanuméricas (concatenação ou substring, p.ex.) ou soma/subtração de datas.
- . Soma - aplica-se a fórmulas de resultado numérico a ser somado nos agrupamentos
- . Média - aplica-se a fórmulas de resultado numérico, a cujos valores, quando agrupados, deve ser atribuída a média aritmética simples.
- . Distinct count - para obter contagem única de elementos da coluna a ser selecionada dentro de um novo combo-box ("Coluna distinct count") que é aberto na sequência.
- . Formula: utiliza-se quando se deseja obter uma média aritmética ponderada como resultado.

Segue tabela com exemplos e respectivos resultados obtidos:

Dados do cubo		Como fica o resultado, informando em "Sumarização":			
<i>Custo total</i>	<i>Qtd</i>	<i>Formula</i>	<i>Média</i>	<i>Soma</i>	<i>Nada</i>
1.000,00	20	50,00	50,00	50,00	50,00
2.000,00	20	100,00	100,00	100,00	100,00
<u>5.000,00</u>	<u>25</u>	<u>200,00</u>	<u>200,00</u>	<u>200,00</u>	<u>200,00</u>
8.000,00	65	123,08	116,67	350,00	
		Média aritmética ponderada (recalcula a cada nível de agrupamento/total)	Média aritmética simples (faz a média a partir de cada linha do cubo)	Não faria sentido, neste caso	Para a situação onde não se aplica exibir algo na(s) linha(s) de total

Na tela seguinte encontra-se um painel para inserir a respectiva expressão e outro com a lista das métricas/dimensões que podem ser utilizadas na expressão da coluna calculada. Para adicioná-las à expressão, basta selecionar o item escolhido e clicar no botão *Adicionar colunas* (<<). Para verificar se a sintaxe está correta, basta clicar no botão *Testar sintaxe*. O botão "Aplicar", serve para confirmar a criação da coluna nova ou respectiva alteração, conforme o caso.

Edita coluna calculada
✕

Nome:*

Fórmula:*

formula= |

Colunas disponíveis

- Categoria (texto)
- Cidade (texto)
- Cliente (texto)
- Data (data)
- NumPedido (numérico decimal)
- País (texto)
- Produto (texto)
- Qtd_fat (numérico decimal)
- Qtd_ped (numérico decimal)
- Saldo qtd (numérico decimal)
- Valor da venda (numérico decima)
- Valor do custo (numérico decimal)
- Vendedor (texto)

3. Sintaxe

Para criar uma coluna calculada a sintaxe utilizada é ***formula = expressão***; O valor da coluna calculada será o valor atribuído a variável *formula*, na expressão introduzida. A *expressão* da coluna calculada pode ser composta por operações entre métricas, dimensões e valores fixos. As operações envolvem operadores aritméticos, relacionais e lógicos.

Exemplo:

*formula = ((get('areaDePlantio') * 2) + (get('areaDePlantio') * 3) / (get('areaDePlantio') - 100));*

4. Aplicações comuns de colunas calculadas

4.01 Utilizando operadores aritméticos

Exemplifica o uso de operadores aritméticos, multiplicação, divisão, adição, subtração, aplicando precedência de operadores com o uso dos parêntesis.

$$formula = ((get('areaDePlantio') * 2) + (get('areaDePlantio') * 3) / (get('areaDePlantio') - 100));$$

Resultado:

regiao	areaDePlantio	FormulaTeste!
CARAMBEI	6.783,25	13.569,54
CENTRO OESTE-MS	2.492,65	4.988,43
GOIÁS	23.579,42	47.161,85
LDB-LONDRINA	4.728,68	9.460,42
LDB-NORTE	6.443,07	12.889,19

4.02 Concatenando texto

Exemplifica a concatenação das dimensões *anoAgricola* e *cultura*, separadas por um hífen.

$$formula = get('anoAgricola') + " - " + get('cultura');$$

Resultado:

anoAgricola	cultura	AnoAgricola+Cultura	areaDePlantio
2006/2007	ALGODAO	2006/2007 - ALGODAO	0,86
2006/2007	ARROZ	2006/2007 - ARROZ	2.389,20
2006/2007	FEIJÃO	2006/2007 - FEIJÃO	100,70
2006/2007	MILHO	2006/2007 - MILHO	4.296,70
2006/2007	SOJA	2006/2007 - SOJA	114.147,19

ATENÇÃO: Utilizar aspas como delimitador de texto fixo a ser concatenado com os valores, como no exemplo acima, onde o hífen está entre aspas. Se os valores a serem concatenados forem métricas, e utilizar apóstrofes como delimitador, o resultado da expressão será os valores somados com o valor ASCII do caracter (no caso o hífen, que tem valor 45).

4.03 Identificar parte do conteúdo de um campo texto/alfanumérico (equivalente ao LIKE)

Para isso é usada a função Contains, StartsWith ou EndsWith, assim:

formula= (get('RotuloColuna')).Contains("texto") ? "Contem" : "NaoContem"

formula= (get('RotuloColuna')).StartsWith("texto") ? "ComecaCom" : "NaoComecaCom"

formula= (get('RotuloColuna')).EndsWith("texto") ? "TerminaCom" : "NaoTerminaCom"

4.04 Separando parte de um campo texto/alfanumérico

Para isso é usada a função *SUBSTRING*.

Sintaxe: .Substring(startIndex) ou .Substring(startIndex, length):

Exemplo: formula= (get('RotuloColuna')).Substring(0, 5)

4.05 Convertendo tipos de dados

Para converter um dado numérico para texto, basta utilizar a expressão `Convert.ToString(valor)`. Para converter de texto para numérico, podemos utilizar `Convert.ToInt32(valor)`, `Convert.ToDouble(valor)` e `Convert.ToDecimal(valor)`.

```
formula = get('cultura') + Convert.ToString(get('areaDePlantio'));
```

4.06 Utilizando funções matemáticas prontas

Ao criar colunas calculadas no SuperBI, é possível utilizar constantes e funções que facilitam a realização de muitas operações matemáticas. No exemplo abaixo, é utilizada a função `Sqrt` para calcular a raiz quadrada da métrica `areaDePlantio`. Note que nessa coluna calculada é feita uma conversão entre tipos de dados. Para mais informações a respeito, veja item 5 da seção FAQs. Para ver a lista completa de constantes e funções da classe `Math`, vide item **Apêndice 1 - Membros da classe Math**.

```
formula = Math.Sqrt((double) get('areaDePlantio'));
```

Resultado:

regiao	areaDePlantio	RaizQuadradaArea
CARAMBEI	6.783,25	82,36
CENTRO OESTE-MS	2.492,65	49,93
GOIÁS	23.579,42	153,56
LDB-LONDRINA	4.728,68	68,77
LDB-NORTE	6.443,07	80,27

4.07 Utilizando if-else

Além dos operadores aritméticos relacionais e lógicos, é possível utilizar as dimensões/métricas para aplicar condições que vão influenciar no resultado, utilizando a sintaxe `if else`. No exemplo abaixo, se o valor da métrica `areaDePlantio` for menor que 10000, o resultado recebe o valor "PEQUENA", senão, recebe o valor "GRANDE".

```
if(get('areaDePlantio') < 10000) result = "PEQUENA";
else result = "GRANDE";
```

Essa coluna calculada também poderia ser escrita utilizando operador ternário (maneira compacta de expressar `if-else`), da seguinte forma:

```
result = get('areaDePlantio') < 10000 ? "PEQUENA" : "GRANDE";
```

Resultado:

regiao	areaDePlantio	tamanhoArea
CARAMBEI	6.783,25	PEQUENA
CENTRO OESTE-MS	2.492,65	PEQUENA
GOIÁS	23.579,42	GRANDE
LDB-LONDRINA	4.728,68	PEQUENA
LDB-NORTE	6.443,07	PEQUENA

Tendo uma **dimensão** como condição:

```
formula= (get('status')).ToUpper() == "OVERDUE" ? get('Invoice_amount') : 0
```

4.08 Compondo fórmulas com E / OU

"E" é representado por &&

"OU" é representado por ||

Exemplo de cálculo de um valor em atraso:

```
formula= get('Due_date') < DateTime.Today && (get('receipt_date') ==  
Convert.ToDateTime("1900-01-01")) ? get('Invoice_amount') : 0
```

```
formula= get('Due_date') > DateTime.Today || (get('receipt_date') <>  
Convert.ToDateTime("1900-01-01")) ? 0 : get('Invoice_amount')
```

4.09 Inserção de constantes com casas decimais

Exemplo, obter 10% do valor da venda :

```
result = get('Valor da venda') * (decimal)0.1;
```

4.10 Calcular diferença de dias entre datas

```
formula = (get('DataFinal') - get('DataInicial')).Days;
```

4.11 Calcular diferença de dias úteis entre datas

```
formula = GetDifWorkdays(get('DataFinal'), get('DataInicial'));
```

4.12 Diferença de dias entre uma data e a data atual

```
formula = (DateTime.Today - get('Data')).TotalDays
```

4.13 Deixar campo numérico sem conteúdo na ausência de uma data

```
if ((get('DataFinal')).Year == 1900)
```

```
formula = null;
```

```
else
```

```
formula = (get('DataFinal') - get('DataInicial')).Days;
```

4.14 Somar/subtrair n dias de uma data

```
formula = DateTime.Today.AddDays(-30);
```

```
formula = (get('DataVenda')).AddDays(30);
```

4.15 Identificar mês anterior

```
formula = ((get('DataVenda')).Year == DateTime.Today.AddMonths(-1).Year &&
```

```
(get('DataVenda')).Month == DateTime.Today.AddMonths(-1).Month) ? 'Mês anterior' :
```

```
'Outro mês';
```

4.16 Extrair dia da data atual

```
formula= DateTime.Today.Day
```

4.17 Extrair mês da data atual

```
formula= DateTime.Today.Month
```

4.18 Calcular valor do mês atual

```
formula = (get('data')).Year == DateTime.Today.Year && ((get('data')).Month ==
```

```
DateTime.Today.Month) ? get('peso_liquido') : 0
```


4.19 Calcular valor do mês anterior

formula = ((get('DataVenda')).Year == DateTime.Today.AddMonths(-1).Year && (get('DataVenda')).Month == DateTime.Today.AddMonths(-1).Month) ? get('ValorVenda') : 0;

4.20 Transformar hora, minuto e segundo (formato HH:MM:SS) em minutos (e decimal)

*formula = Convert.ToDecimal((Convert.ToDecimal((get('HHMMSS')).Split(new Char[]{' ':''})[0]) * 60) + Convert.ToDecimal((get('HHMMSS')).Split(new Char[]{' ':''})[1]) + (Convert.ToDecimal((get('HHMMSS')).Split(new Char[]{' ':''})[2]) / 60))*

4.21 Transformar hora e minuto (formato HH:MM) em minutos (e decimal)

*formula = Convert.ToDecimal((Convert.ToDecimal((get('HHMM')).Split(new Char[]{' ':''})[0]) * 60) + Convert.ToDecimal((get('HHMM')).Split(new Char[]{' ':''})[1]))*

4.22 Transformar hora e minuto (formato HH:MM) em horas (e decimal)

formula = Convert.ToDecimal((Convert.ToDecimal((get('HHMM')).Split(new Char[]{' ':''})[0])) + Convert.ToDecimal((get('HHMM')).Split(new Char[]{' ':''})[1]) / 60);

5 FAQ

Quando tento adicionar uma coluna calculada apresenta um erro do tipo "Melhor método sobrecarregado compatível com 'System.Math.Sqrt(double)' tem alguns argumentos inválidos. Argumento '1': não pode converter de 'decimal' em 'double'."

O que devo fazer?

Resposta: Esse erro acontece porque o método *Sqrt* da classe *Math* espera como parâmetro um valor numérico do tipo *double*, e o parâmetro utilizado representa um numérico do tipo *decimal*. Para executar a coluna calculada corretamente, basta fazer um *typecast* (conversão de tipo de dado), da seguinte maneira: "System.Math.Sqrt((double)parâmetro)".

Apêndice 1 – Membros da classe Math

Funções:

Nome	Descrição	Sintaxe
Abs	Retorna o valor absoluto de um número específico.	<i>Math.Abs(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Acos	Retorna o ângulo cujo cosseno é o número especificado.	<i>Math.Acos(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Asin	Retorna o ângulo cujo seno é o número especificado.	<i>Math.Asin(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Atan	Retorna o ângulo cuja tangente é o número especificado.	<i>Math.Atan(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Atan2	Retorna o ângulo cuja tangente é o quociente de dois números especificados.	<i>Math.Atan2(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> .
BigMul	Produz o produto completo de dois números de 32 bits.	<i>Math.BigMul(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>int</i> .
Ceiling	Retorna o menor inteiro maior ou igual ao número especificado.	<i>Math.Ceiling(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> ou <i>decimal</i> .
Cos	Retorna o cosseno do ângulo especificado.	<i>Math.Cos(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Cosh	Retorna o cosseno hiperbólico do ângulo especificado.	<i>Math.Cosh(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
DivRem	Calcula o quociente de dois números e também retorna o resto em um parâmetro de saída.	<i>Math.DivRem(x, y, z)</i> Onde <i>x</i> , <i>y</i> e <i>z</i> são numéricos, do tipo <i>int</i> .
Exp	Retorna <i>e</i> elevado à potência especificada.	<i>Math.Exp(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Floor	Retorna o maior inteiro menor ou igual ao número especificado.	<i>Math.Floor(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> ou <i>decimal</i> .
IEEERemainder	Retorna o número especificado do restante resultantes da divisão de um número especificado por outro.	<i>Math.IEEERemainder(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> .
Log	Retorna o logaritmo de um número especificado em uma base especificada.	<i>Math.Log(x)</i> ou <i>Math.Log(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> .
Log10	Retorna o logaritmo base 10 de um número especificado.	<i>Math.Log10(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .

Max	Retorna o maior dos dois números especificados.	<i>Math.Max (x, y)</i> Onde x e y são numéricos, do tipo <i>int</i> , <i>double</i> ou <i>decimal</i> .
Min	Retorna o menor dos dois números.	<i>Math.Min (x, y)</i> Onde x e y são numéricos, do tipo <i>int</i> , <i>double</i> ou <i>decimal</i> .
Pow	Retorna um número especificado elevado à potência especificada.	<i>Math.Pow(x, y)</i> Onde x e y são numéricos, do tipo <i>double</i> .
Round	Arredonda um valor para o inteiro mais próximo ou o número especificado de casas decimais.	<i>Math.Round (x)</i> ou <i>Math.Round (x, y)</i> Onde x e y são numéricos, x do tipo <i>double</i> ou <i>decimal</i> e y do tipo <i>int</i> .
Sign	Retorna um valor indicando o sinal de um número.	<i>Math.Sign (x)</i> Onde x é numérico, do tipo <i>int</i> , <i>double</i> ou <i>decimal</i> .
Sin	Retorna o seno do ângulo especificado.	<i>Math.Sin(x)</i> Onde x é numérico, do tipo <i>double</i> .
Sinh	Retorna o seno hiperbólico do ângulo especificado.	<i>Math.Sinh(x)</i> Onde x é numérico, do tipo <i>double</i> .
Sqrt	Retorna a raiz quadrada do número especificado.	<i>Math.Sqrt(x)</i> Onde x é numérico, do tipo <i>double</i> .
Tan	Retorna a tangente do ângulo especificado.	<i>Math.Tan(x)</i> Onde x é numérico, do tipo <i>double</i> .
Tanh	Retorna a tangente hiperbólica do ângulo especificado.	<i>Math.Tanh(x)</i> Onde x é numérico, do tipo <i>double</i> .
Truncate	Calcula a parte integral de um número.	<i>Math.Truncate(x)</i> Onde x é numérico, do tipo <i>double</i> ou <i>decimal</i> .

Constantes:

Nome	Descrição	Valor	Sintaxe
E	Representa a natural logarítmica base, especificado pela constante <i>e</i> .	2.7182818284590452354	Math.E
PI	Representa a proporção entre a circunferência de um círculo ao seu diâmetro, especificado pela constante π .	3.14159265358979323846	Math.PI