Banco de Dados

Linguagem MDX

Multi**D**imensional e**X**pressions

Valéria Cesário Times
vct@cin.ufpe.br

Tópicos

- Conceitos Básicos
- Sintaxe MDX
- Consultas e Expressões em MDX
- Principais Funções
- Usando mais de dois eixos
- Bibliografia

Definições

- Cubo OLAP
  - Organiza os dados em **dimensões** e **medidas**

QuantidadeVendida

	Produto			
Tempo	Milho	Ervilha	Azeitona	Palmito
Abril	16	23	12	4
Maio	14	12	23	6
Junho	34	19	19	8
Julho	17	22	14	4

  - Rótulos das colunas são **membros** da dimensão Produto
  - Rótulos das linhas são **membros** da dimensão Tempo
  - Células contêm valores da medida QuantidadeVendida

Definições

- Cubo OLAP
  - Trocar colunas e linhas sem alterar o significado do dado

QuantidadeVendida

	Tempo			
Produto	Abril	Maio	Junho	Julho
Milho	16	14	34	17
Ervilha	23	12	19	22
Azeitona	12	23	19	14
Palmito	4	6	8	4

  - Cada célula é descrita em termos de um membro de cada dimensão
    - Quantidade de latas de milho vendidas em maio = 14

Definições

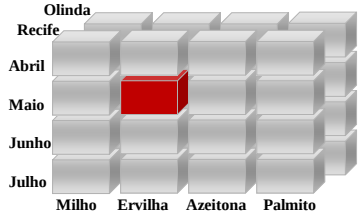
- Cubo OLAP
  - Cada valor da medida (**fato**) ocorre em uma única interseção entre membros de dimensões distintas
  - Um cubo pode exibir valores de mais de uma medida
    - Valores exibidos em planilhas separadas ou em única planilha

QuantidadeVendida / ValorVendido

	Produto							
Tempo	Milho		Ervilha		Azeitona		Palmito	
Abril	16	48,00	23	115,00	12	120,00	4	80,00
Maio	14	42,00	12	60,00	23	230,00	6	120,00
Junho	34	102,00	19	95,00	19	190,00	8	160,00
Julho	17	51,00	22	110,00	14	140,00	4	80,00

Definições

- Cubo OLAP
  - Cubos podem ter mais de duas dimensões
    - *Analysis Services* permite até 1024 medidas e até 128 dimensões



  - Cada eixo representa uma dimensão

Definições

- Hierarquias e Agregações
  - Membros de uma dimensão podem ser organizados em hierarquias (e geralmente são!)
  - Maioria dos cubos têm uma dimensão temporal a qual quase sempre é hierárquica
  - Tempo pode ser organizado em vários **níveis**

    Mês → Trimestre → Ano
  - Hierarquias têm níveis. Níveis têm membros.
  - A célula da interseção entre Milho e 2º. Trimestre contém o valor agregado para Milho de Abril à Junho.

Definições

- Hierarquias e Agregações
  - QuantidadeVendida

		Produto			
Tempo		Milho	Ervilha	Azeitona	Palmito
Ano	2000	242	199	196	65
	2001	232	201	219	75
	2002	294	214	209	86

		Produto			
Tempo		Milho	Ervilha	Azeitona	Palmito
Trimestre	Q1	61	36	58	21
	Q2	64	54	54	18
	Q3	45	59	33	12
	Q4	72	50	51	14

Definições

- Hierarquias e Agregações
  - QuantidadeVendida

		Produto			
Tempo		Milho	Ervilha	Azeitona	Palmito
Ano	2000	242	199	196	65
	2001	232	201	219	75
	2002	294	214	209	86

		Produto			
Tempo		Milho	Ervilha	Azeitona	Palmito
Trimestre	Q1	61	36	58	21
	Q2	64	54	54	18
	Q3	45	59	33	12
	Q4	72	50	51	14

Definições

- Hierarquias e Agregações
  - QuantidadeVendida

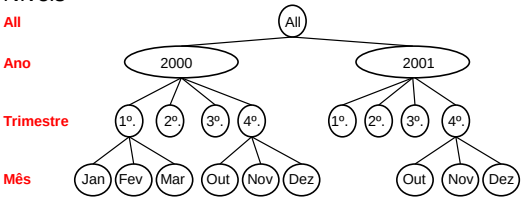
			Ervilha	Azeitona
2000	Q3	Julho	17	22
		Agosto	16	18
		Setembro	12	19
	Q3 Total		45	59
2000	Q4	Outubro	27	19
		Novembro	24	19
		Dezembro	21	12
	Q4 Total		72	50
2000 Total			242	199

Fatos Detalhados

Fato Agregado

Definições

- Níveis



  - Maioria das dimensões possuem All
    - É o nível mais alto da hierarquia
    - Contém a informação mais agregada representada pela menor quantidade de membros

Definições

- Nomeação
  - Cada nível da hierarquia possui um nome descritivo
  - Para identificar um membro, especifica-se:
    - Nome da dimensão
    - Membros de cada nível da hierarquia até o membro de interesse.
    - Por exemplo: [Loja] . [All] . [Pernambuco] . [Recife] . [Filial1]
  - Forma abreviada: [Loja] . [Filial1]
    - Porém, esta forma abreviada não funciona em todos os casos
    - Por exemplo: [Tempo] . [Out] não identifica o membro desejado.
    - Para resolver: [Tempo] . [Out-2000] ou

      [Tempo] . [All] . [2000] . [Trim4] . [Out]

Definições

- Tuplas
  - Interseção vários membros, sendo cada um deles de uma dimensão diferente do cubo
    - Não requer um membro de cada dimensão do cubo
  - Tem o potencial para identificar uma única célula do cubo
  - Não usa mais de um membro da mesma dimensão
  - Coleção de membros de dimensões diferentes
  - Sintaxe:
    - ([*Dim1*] . [*Membro*], [*Dim2*] . [*Membro*], [*Dim3*] . [*Membro*])
  - Exemplo:
    - ([Produto] . [Milho], [Tempo] . [Maio], [Loja] . [Recife])

Definições

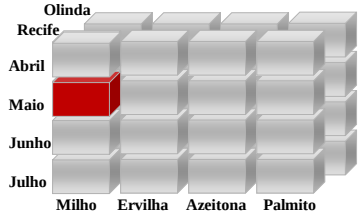
- Tuplas
  - Interseção vários membros, sendo cada um deles de uma dimensão diferente do cubo
    - Não requer um membro de cada dimensão do cubo
  - Tem o potencial para identificar uma única célula do cubo
  - Não usa mais de um membro da mesma dimensão
  - Coleção de membros de dimensões diferentes
  - Sintaxe:
    - ([*Dim1*] . [*Membro*], [*Dim2*] . [*Membro*], [*Dim3*] . [*Membro*])
  - Exemplo:
    - ([Produto] . [Milho], [Loja] . [Recife] , [Tempo] . [Maio])

Definições

- Conjuntos
  - Coleção de zero ou mais tuplas definidas sobre as mesmas dimensões (mesma dimensionalidade)
    - Pode ter tuplas repetidas e a ordem é importante
  - Sintaxe:
    - { ([*Dim1*] . [*Membro*], [*Dim2*] . [*Membro*], [*Dim3*] . [*Membro*]) , ([*Dim1*] . [*Membro*], [*Dim2*] . [*Membro*], [*Dim3*] . [*Membro*]), ... }
  - Exemplo:
    - { ([Produto] . [Milho], [Tempo] . [Maio], [Loja] . [Recife]) , ([Produto] . [Ervilha], [Tempo] . [Maio], [Loja] . [Recife]) }
    - Ambas tuplas têm um membro de Produto, Tempo e Loja.
      - Elas têm a mesma dimensionalidade

Definições

- Tuplas x Conjuntos
  - Considere o seguinte cubo:



  - O exemplo abaixo é uma tupla?

    ([Tempo] . [Maio], [Loja] . [Recife], [Produto] . [Milho])

Definições

- Tuplas x Conjuntos
  - E quanto aos exemplos dados abaixo?
    1. ([Loja] . [Recife], [Produto] . [Milho])
    2. { ([Tempo] . [Abril], [Loja] . [Recife], [Produto] . [Milho]) , ([Tempo] . [Maio], [Loja] . [Recife], [Produto] . [Milho]) , ([Tempo] . [Junho], [Loja] . [Recife], [Produto] . [Milho]), ([Tempo] . [Julho], [Loja] . [Recife], [Produto] . [Milho]) }

Ambos retornam as mesmas células do cubo !

Definições

- Tuplas x Hierarquias
  - Considere o seguinte exemplo:
    - Suponha um cubo com uma dimensão Tempo
    - Esta dimensão possui os níveis:

      Mês → Trimestre → Ano
    - Existem dados para os anos 1999, 2000 e 2001.
    - Então, a expressão abaixo resulta em uma única célula do cubo?

      ([Produto] . [Milho] , [Loja] . [Recife] , [Tempo] . [2000])
    - Células com valores agregados sempre existem no cubo.
      - Alguns destes valores são materializados (calculados e armazenados), enquanto outros são derivados em tempo de execução.
      - Tuplas podem recuperar o valor mais agregado ou detalhado

Definições

- Medidas x Dimensões
  - Se um cubo possui mais de uma medida, então a medida desejada é requisitada de forma semelhante à consulta sobre membros de uma dimensão
  - Exemplo em pseudo-MDX:

    ([Produto] . [Milho], [Tempo] . [Maio], [Loja] . [Recife] ,
    [Measures] . [QuantidadeVendida])
  - Se a medida desejada não for especificada, então a medida *default* é usada
    - Esta opção *default* geralmente pode ser alterada

Definições

- Propriedades de Membros
  - Existem informações no cubo que não se relacionam logicamente com **todas** as suas dimensões
  - Elas se relacionam com membros de uma única dimensão
    - Área de uma loja
    - Peso do produto
  - Tais informações não são armazenadas como medidas
  - São definidas como propriedades de um membro da dimensão relacionada
    - Nome da loja (membro da dimensão Loja)
    - Nome do produto (membro da dimensão Produto)

Definições

- Propriedades de Membros
  - Existem informações no cubo que não se relacionam logicamente com **todas** as suas dimensões
  - Elas se relacionam com membros de uma única dimensão
    - Área de uma loja
    - Peso do produto
  - Tais informações não são armazenadas como medidas
  - São definidas como propriedades de um membro da dimensão relacionada
    - Nome da loja (membro da dimensão Loja)
    - Nome do produto (membro da dimensão Produto)

MDX - Sintaxe

- Colchetes [ ]
  - Nomes de dimensões e de membros são usados entre [ ]
    - Porém , o uso de [ ] é apenas obrigatório se o nome contém:
      - Números
      - Espaços em branco
      - Palavras reservadas (e.g. [FROM])
      - Caracteres especiais
    - Exemplos de nomes corretos para dimensões ou membros
      - [Todos Produtos]
      - Bebida
      - [Bebida]
      - [Trim3]

MDX - Sintaxe

- Pontos
  - Nome de uma dimensão seguido por vários nomes de membros - devem vir separados pelo ponto
    - [Produto] . [Bebida] . [Cerveja]
- Parênteses ( )
  - São usados para denotar Tuplas
    - Exemplo: ([Produto] . [Bebida] . [Cerveja])
  - Tuplas contêm membros de uma ou mais dimensões
    - Membros são separados por vírgulas
    - Exemplos: ([Produto] . [Bebida] . [Cerveja] , [Cliente] . [PE])

      ([Produto] . [Bebida] . [Cerveja] , [Cliente] . [PE] , [Tempo] . [2003])

MDX - Sintaxe

- Chaves { }
  - São usadas para denotar Conjuntos
    - Exemplos:

      { ([Produto] . [Bebida] . [Cerveja]) , ([Produto] . [Comida] . [Milho]) }

      { [Clientes] . [All] . [PE] . Children }

      { [Produtos] . [All] . Children }

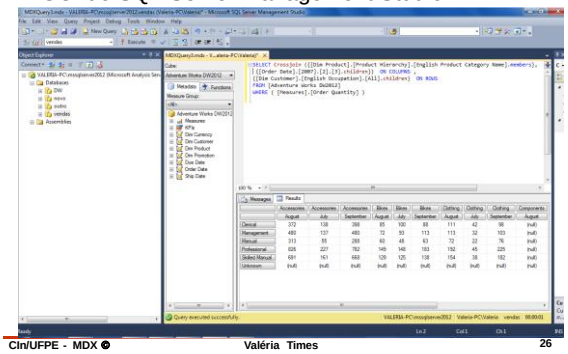
      { [Produtos] . [All] . [Enlatados] . [Milho] ,
      [Produtos] . [All] . [Enlatados] . [Ervilha] ,
      [Produtos] . [All] . [Enlatados] . [Azeitona] ,
      [Produtos] . [All] . [Enlatados] . [Palmito] }

MDX - Consultas

- Aspectos Importantes
  - Resultados das consultas MDX são sub-cubos
  - Não pode haver duplicidade de dimensões na consulta
  - Dimensões são mapeadas para eixos do sub-cubo
  - Dimensões não associadas aos eixos podem ser usadas na cláusula WHERE
  - Uma consulta MDX pode ter mais de um eixo
    - SQL Server Management Studio só permite no máximo dois eixos (*columns* e *rows*)
  - Linguagem de consulta *ad hoc* para suporte à decisão
    - Servidores OLAP não implementam todas as funcionalidades

MDX - Consultas

- GUI do SQL Server Management Studio



MDX - Consultas

- Exemplo de uma consulta em MDX

```
SELECT
{ [Cliente] . [All ] }  ON COLUMNS ,
{ [Measures] . [QuantidadeVendida] }  ON ROWS
FROM [Vendas]
WHERE   [Tempo] . [1998]
```

- MDX é considerada como uma extensão de SQL?
  - Apesar das semelhanças aparentes, elas diferem em vários aspectos importantes!

MDX - Consultas

- Componentes de uma consulta MDX:
  - Elementos das Colunas
  - Elementos das Linhas

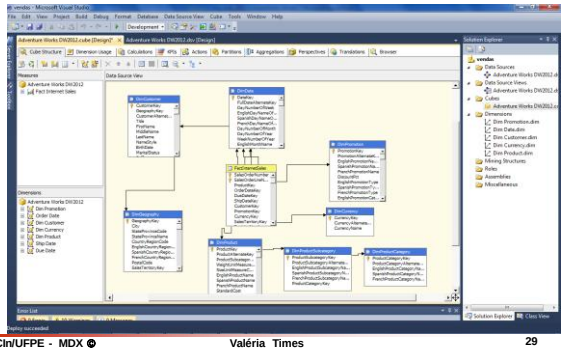
  } **Conjuntos**
  - Nome do cubo desejado

```
SELECT
{ conjunto de elementos das colunas }  ON COLUMNS ,
{ conjunto de elementos das linhas }  ON ROWS
FROM [ nome do cubo ]
```

  - Se uma dimensão não for especificada e não existir o nível *All* com o membro *default (All), será escolhido o 1º.* membro do nível mais alto

MDX - Consultas

- Cubo *FactInternetSales*



MDX - Consultas

- Exemplo de inversão de linhas e colunas

```
SELECT
{ [Product] . [All Products] }  ON COLUMNS ,
{ [Measures] . [Unit Sales] }  ON ROWS
FROM [Sales]
```

	All Products
Unit Sales	266.773,00

```
SELECT
{ [Measures] . [Unit Sales] } ON COLUMNS ,
{ [Product] . [All Products] } ON ROWS
FROM [Sales]
```

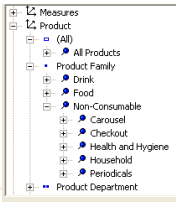
	Unit Sales
All Products	266.773,00

MDX - Consultas

- Exemplo de aumento do nível de detalhes

```
SELECT
 { [Measures] . [Unit Sales]  } ON COLUMNS,
 { [Product] . [All Products] . [Food] ,
   [Product] . [All Products] . [Drink] ,
   [Product] . [All Products] . [Non-Consumable]  } ON ROWS
FROM [Sales]
```

	Unit Sales
Food	191.940,00
Drink	24.597,00
Non-Consumable	50.236,00



MDX - Consultas

- Aumentando ainda mais o nível de detalhes

```
SELECT  { [Measures] . [Unit Sales]  } ON COLUMNS,
{ [Product] . [Product Family] . [Food] . [Baked Goods] , [Product] . [Product Family] . [Food] . [Baking Goods] , [Product] . [Product Family] . [Food] . [Breakfast Foods] , [Product] . [Product Family] . [Food] . [Canned Foods] ,
 [Product] . [Product Family] . [Food] . [Canned Products] , [Product] . [Product Family]. [Food] . [Dairy], [Product] . [Product Family] . [Food] . [Baked Goods] ,
 [Product] . [Product Family] . [Food] . [Deli] , [Product] . [Product Family] . [Food] . [Eggs] , [Product] . [Product Family] . [Food] . [Frozen Foods] ,
 [Product] . [Product Family] . [Food] . [Meat] , [Product] . [Product Family] . [Food] . [Produce] , [Product] . [Product Family] . [Food] . [Seafood] ,
 [Product] . [Product Family] . [Food] . [Snack Foods] ,  [Product] .[Product Family]. [Food] . [Starchy Foods]  } ON ROWS
FROM [Sales]
```


MDX - Funções

- Função Children
  - Retorna o conjunto dos membros do nível diretamente abaixo do membro ao qual a função é aplicada
  - Sintaxe: [Dim] . [Member] . Children
  - Exemplo:

```
SELECT
{ [Measures] . [Unit Sales]  } ON COLUMNS ,
{ [Product] . [Product Family] . [Food] . Children }
     ON ROWS
FROM  [Sales]
```

	Unit Sales
Baked Goods	7.870,00
Baking Goods	20.245,00
Breakfast Foods	3.317,00
Canned Foods	19.026,00
Canned Products	1.812,00
Dairy	12.885,00
Deli	7.870,00
Eggs	12.037,00
Frozen Foods	4.132,00
Meat	26.655,00
Produce	1.714,00
Seafood	37.792,00
Snack Foods	1.764,00
Starchy Foods	30.545,00

MDX - Funções

- Função Children - Outro Exemplo:

```
SELECT
{ [Customers] . [All Customers] . [USA] . Children }  ON COLUMNS ,
{ [Product] . [Product Family] . [Food] . Children }  ON ROWS
FROM  [Sales]
```

Cadê a medida?
Se nenhuma medida for especificada, então:
- medida *default* é usada ou
- 1ª. medida é escolhida

	CA	OR	WA
Baked Goods	2.150,00	2.013,00	3.707,00
Baking Goods	5.799,00	4.810,00	9.636,00
Breakfast Foods	938,00	862,00	1.517,00
Canned Foods	5.268,00	4.889,00	8.869,00
Canned Products	448,00	464,00	900,00
Dairy	3.534,00	3.131,00	6.220,00
Deli	3.393,00	3.038,00	5.606,00
Eggs	1.116,00	1.119,00	1.897,00
Frozen Foods	7.505,00	6.575,00	12.575,00
Meat	527,00	469,00	718,00
Produce	10.588,00	9.744,00	17.460,00
Seafood	441,00	451,00	872,00
Snack Foods	8.543,00	7.789,00	14.213,00
Snacks	1.958,00	1.831,00	3.095,00
Starchy Foods	1.448,00	1.352,00	2.462,00

MDX - Funções

- Função Where
  - Suponha que se deseje reescrever a consulta anterior para exibir valores de uma dada medida

```
SELECT
 { [Customers] . [All Customers] . [USA] . Children }  ON COLUMNS ,
 { [Product] . [Product Family] . [Food] . Children }  ON ROWS
 FROM  [Sales]
 WHERE ( [Measures] . [Sales Average] )
```

  - Mas não se aplica apenas às medidas e nem é restrita a uma única dimensão!

MDX - Funções

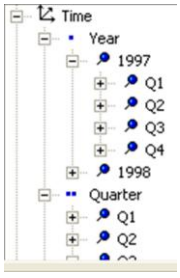
- Função Where
  - A consulta abaixo exibe os mesmos dados da anterior só que para o ano 1998 apenas!

```
SELECT
 { [Customers] . [All Customers] . [USA] . Children }  ON COLUMNS ,
 { [Product] . [Product Family] . [Food] . Children }  ON ROWS
 FROM  [Sales]
 WHERE ( [Measures] . [Sales Average] , [Time] . [1998] )
```

  - Qualquer dimensão que não estiver explicitamente indicada na consulta é chamada de *slicer dimension*
    - Filtrará os dados de acordo com o valor do membro *default* (All)

MDX - Funções

- Função FirstChild
  - Retorna o primeiro filho de um membro
  - Sintaxe: [*Membro*]. FirstChild
  - Exemplo: [Time].[1997].FirstChild
    retorna Q1



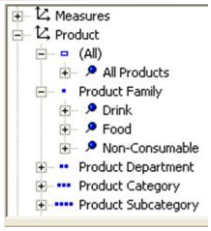
- Função LastChild
  - Retorna o último filho de um membro
  - Sintaxe: [*Membro*]. LastChild
  - Exemplo: [Time].[1997].LastChild retorna Q4

MDX - Funções

- Função Parent
  - Retorna o pai de um dado membro
  - Sintaxe: [*Membro*]. Parent
  - Exemplo:

[Product].[All Products].[Drink].Parent

Retorna [All Products]



  - Outros exemplos:

[Product].[All Products].[Food].Parent retorna [All Products]

[Time] . [1997] . [Q1] . [1] . Parent retorna [Q1]

MDX – Funções em Expressões

- Expressões
  - Recebem uma tupla ou um conjunto como parâmetro
  - Retornam sempre um valor (pode ser o valor nulo)
  - São usadas para definir:
    - Membros calculados ou medidas calculadas
    - Conjuntos
    - Propriedades de membros
  - Medidas calculadas: seus valores são derivados de medidas já existentes (e.g. Lucro = Vendas – Custo)
  - Membros calculados: seus valores são derivados de membros já existentes e não possuem propriedades

MDX - Funções

- Função CurrentMember
  - Retorna o membro atual de uma dimensão definido durante uma interação
    - O membro retornado é relativo ao contexto da interação
  - Sintaxe: [*Dimensao*] . CurrentMember
  - Exemplo:

```
SELECT { [Customers] . CurrentMember } ON COLUMNS ,
{ [Product] . [Product Family] . [Food] . Children } ON ROWS
FROM [Sales]
WHERE ( [Measures] . [Sales Average] )
```


MDX - Funções

- Função PrevMember
  - Retorna o membro anterior do nível que contém o membro ao qual a função é aplicada
  - Sintaxe: [*Membro*] . PrevMember
  - Exemplo: Se o nível Ano contém: [1994], [1995] e [1996], então: [1996].PrevMember retorna o membro [1995]
- Função NextMember
  - Retorna o membro posterior do nível que contém o membro ao qual a função é aplicada
  - Sintaxe: [*Membro*] . NextMember
  - Exemplo: [1994].NextMember retorna [1995]

MDX - Funções

- Função Lag
  - Retorna o membro anterior que está distante do membro ao qual a função é aplicada em um número específico de posições
  - Sintaxe: [*Membro*] . Lag (*expressao_numerica*)
  - Observações:
    - Lag (0) retorna o próprio membro ao qual a função é aplicada
    - Lag (1) é equivalente a PrevMember
    - Lag (-1) é equivalente a NextMember
  - Exemplo: Se a dimensão Tempo tem os níveis Ano e Mês, então: [1995] . [Fev] . Lag(3) retorna [1994] . [Nov]

MDX - Funções

- Função Lead
  - Retorna o membro posterior que está distante do membro ao qual a função é aplicada em um número específico de posições
  - Sintaxe: [*Membro*] . Lead (*expressao_numerica*)
  - Observações:
    - Lead (0) retorna o próprio membro ao qual a função é aplicada
    - Lead (1) é equivalente a NextMember
    - Lead (-1) é equivalente a PrevMember
  - Exemplo: Se a dimensão Tempo tem os níveis Ano e Mês, então: [1994] . [Nov] . Lead(3) retorna [1995] . [Fev]

MDX - Funções

- Comparando valores

Ano	Trimestre	Mês	Vendas
2000			790
	Trim1		120
		Jan	30
		Fev	40
		Mar	50
	Trim2		200
		Abr	65
		Mai	45
		Jun	90

  - Criando uma medida calculada:

([Time] . CurrentMember , [Measures] . [Store Sales]) -
([Time] . CurrentMember . PrevMember , [Measures] . [Store Sales])

MDX - Funções

- Considere a medida calculada *Crescimento*

			MeasuresLevel	
- Year	- Quarter	Month	Store Sales	Crescimento
- 1997	1997 Total		R$ 565.238,13	R$ 565.238,13
	- Q1	Q1 Total	R$ 139.628,35	R$ 139.628,35
		1	R$ 45.539,69	R$ 45.539,69
		2	R$ 44.058,79	(R$ 1.480,90)
		3	R$ 50.029,87	R$ 5.971,08
	+ Q2	Q2 Total	R$ 132.666,27	(R$ 6.962,08)
	+ Q3	Q3 Total	R$ 140.271,89	R$ 7.605,62
	+ Q4	Q4 Total	R$ 152.671,62	R$ 12.399,73

  - Os valores entre ( ) consistem em valores negativos
  - Exibição de valores negativos pode ser configurada

MDX - Funções

- Modificar a medida anterior dada abaixo:

([Time] . CurrentMember , [Measures] . [Store Sales]) -
([Time] . CurrentMember . PrevMember , [Measures] . [Store Sales])

- Considerando os membros do nível Mês:

([Time] . CurrentMember , [Measures] . [Store Sales]) -
([Time] . CurrentMember . Lag (12) , [Measures] . [Store Sales])

Funciona?
Sim. Mas para o nível mês apenas porque:
- Nível trimeste: Lag (12) implica em 12 trimestres anteriores = 3 anos!
- No nível ano: Lag (12) implica em 12 anos anteriores!

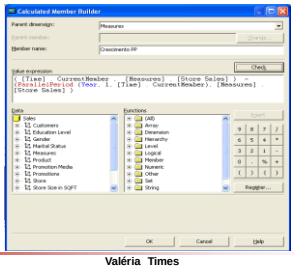
MDX - Funções

- Função ParallelPeriod
  - Recebe 3 parâmetros:
    - Nível (e.g. Ano, Trimestre ou Mês) ou unidade de tempo
    - Expressão numérica para indicar quantas unidades do primeiro parâmetro deve-se voltar atrás no tempo
    - Indicação do membro sobre o qual a comparação deve ser feita
  - Retorna o membro de um dado período anterior com a mesma posição relativa ao membro dado como parâmetro
  - Sintaxe: ParallelPeriod (*nivel*, *expressao_numerica*, [*membro*])
  - Exemplo: ParallelPeriod (Year , 2, [Fev96]) retorna [Fev94]

MDX - Funções

- Como comparar dados do período corrente com dados de um período anterior de forma genérica?

([Time] . CurrentMember , [Measures] . [Store Sales]) -
(ParallelPeriod (Year, 1, [Time] . CurrentMember), [Measures] . [Store Sales])



MDX - Funções

- Função Sum
  - Retorna a soma de uma expressão numérica avaliada sobre um conjunto
  - Sintaxe Sum ({*conjunto*} , *expressao_numerica*)
  - Exemplos:

Sum ({ [Produto].[Milho] , [Produto].[Ervilha] , [Produto].[Azeitona] , [Produto].[Palmito] } , [Measures].[QuantidadeVendida])

Sum ({ [Product].[All Products].[Food] , [Product].[All Products].[Drink] , [Product].[All Products].[Non-Consumable], } , [Measures].[Unit Sales])

MDX - Funções

- Função YTD
  - Recebe um membro como parâmetro
  - Retorna um conjunto formado pelo parâmetro de entrada e pelos membros anteriores do nível Ano
  - Sintaxe: YTD ([*membro*])
  - Exemplo:

Sum (YTD ([Time] . CurrentMember), [Measures] . [Unit Sales])

Retorna a soma da quantidade vendida para o conjunto de membros identificados pela função YTD

MDX - Funções

- Considere a medida calculada *medidaYTD*

			MeasuresLevel	
- Year	- Quarter	Month	Unit Sales	meddaYTD
- 1997	1997 Total		266.773,00	266.773,00
	- Q1	Q1 Total	66.291,00	66.291,00
		1	21.628,00	21.628,00
		2	20.957,00	42.585,00
		3	23.706,00	66.291,00
	+ Q2	Q2 Total	62.610,00	128.901,00
	+ Q3	Q3 Total	65.848,00	194.749,00
	+ Q4	Q4 Total	72.024,00	266.773,00
+ 1998	1998 Total			

  - Em cada nível, os valores de cada membro são acumulados com os valores dos membros anteriores

MDX - Funções

- Função PeriodsToDate
  - Recebe um nível e um membro como parâmetros
  - Retorna um conjunto formado pelo 2o. parâmetro e pelos membros anteriores do nível dado como 1o. parâmetro
  - Sintaxe: PeriodsToDate (*nivel,* [*membro*])
  - PeriodsToDate (Ano , [membro]) = YTD ([membro])
  - Exemplo:

SELECT
{[Measures].[Unit Sales]} ON COLUMNS,
{PeriodsToDate (Quarter, [9])} ON ROWS
FROM Sales

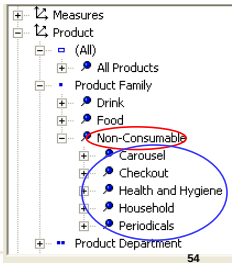
	Unit Sales
7	23.763,00
8	21.697,00
9	20.388,00

MDX - Funções

- Uso de Funções Aninhadas

1) [Product] . [All Products] . LastChild . Children
2) [Product] . [All Products] . LastChild . Parent
3) [Product] . [All Products] . Children . Parent

Não funciona porque *Children* retorna um conjunto de membros e *Parent* deve ser aplicado a um único membro!



MDX - Funções

- Valores de Membros Nulos
  - Dados de fora do limites podem ser solicitados:
    - Pai do membro de último nível
    - Filho do último membro
  - Nestes casos, valores de membros nulos são retornados
  - Exemplo: [Products] . [All Products] . Parent . FirstChild

Retorna NULL!

  - Para escrever expressões em MDX tem-se que explicitar todos os membros?

MDX - Funções

- Criar uma medida calculada de modo que:
  - a quantidade de itens vendidos de qualquer membro seja expressa em termos de um percentual sobre a quantidade de itens vendidos de seu membro pai

([Product] . CurrentMember , [Measures] . [Unit Sales]) /

([Product] . CurrentMember . Parent , [Measures] . [Unit Sales]) * 100

  - a quantidade de itens vendidos de qualquer membro seja expressa em termos de um percentual sobre a quantidade de todos os produtos vendidos

([Product] . CurrentMember , [Measures] . [Unit Sales]) /

([Product] . [All Products], [Measures] . [Unit Sales]) * 100

MDX - Funções

- Função Descendants
  - Recebe um nível e um membro como argumentos
  - Retorna o conjunto de descendentes de um membro em um dado nível (ou distância)
  - Sintaxe: Descendants ([*nivel*] , [*membro*])
  - Exemplos:

Descendants ([Time] . [1997] , [Quarter]) retorna Q1, Q2, Q3 e Q4

Descendants ([Time] . [1998] , [Month]) retorna 1, 2, 3, 4, ..., 12

Descendants ([Time] . [1998] . [Q2] , [Month]) retorna 4, 5 e 6

  - Possui variações quanto aos tipos e números de argumentos recebidos

MDX - Funções

- Função Descendants - variações
  - Pode também receber como argumentos:
    - Membro cujos descendentes devem ser obtidos
    - Número relativo à quantidade de níveis que se deseja descer na hierarquia
  - Sintaxe: Descendants ([*membro*] , *numero*)
  - Exemplos:

Descendants ([Time] . [1998] , 2) retorna 1, 2, 3, 4, ..., 12

Descendants ([Time] . [1998] . [Q2] , 1) retorna 4, 5 e 6

Descendants ([Time] . [1998] , [Month])

Descendants ([Time] . [1998] . [Q2] , [Month])

MDX - Funções

- Função Descendants - variações
  - Permite que o nível não seja especificado
  - Sintaxe: Descendants ([*membro*])
  - Exemplos:

Descendants ([Time] . [1997]) retorna 1997, Q1, Q2, Q3 e Q4 e 1, ..., 12

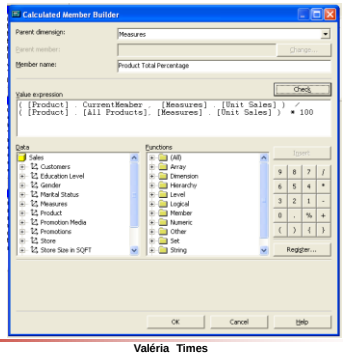
Descendants ([Time] . [1998] . [Q2]) retorna Q2 e 4, 5 e 6

Sem a definição do nível cujos descendentes devem ser retornados, esta função retorna o próprio membro + todos membros abaixo!

  - Qual a diferença entre *Children* e *Descendants*?
    - Com *Children* não se especifica o nível
    - *Children* não retorna os membros de todos níveis abaixo

MDX - Funções

- Criando medida calculada *Product Total Percentage*



MDX - Funções

- Medidas Calculadas em Consultas

```
WITH MEMBER [Measures].[Product Total Percentage] AS
'( [Product].CurrentMember , [Measures] . [Unit Sales] ) /
 ( [Product] . [All Products], [Measures] . [Unit Sales] ) * 100 '
 SELECT Descendants ( [Time] . [1997].[Q1] ) ON COLUMNS ,
        { [Product] . [All Products]. Children } ON ROWS FROM Sales
WHERE ([Measures] . [Product Total Percentage])
```

	Q1	1	2	3
Drink	9,01	8,83	9,31	8,92
Food	72,12	72,15	72,25	71,98
Non-Consum	18,87	19,02	18,44	19,10

MDX - Funções

- Função Ancestor
  - Recebe um membro e um nível como argumentos
  - Retorna o membro antecendente de um membro em um dado nível (ou distância)
  - Sintaxe: Ancestor ([*membro*] , [*nivel*])
  - Exemplos:

    Ancestor (Los Angeles , Country) retorna [USA]

    Ancestor (Los Angeles , State) retorna [California]

    Ancestor (Los Angeles, 0) retorna [Los Angeles]

    Ancestor (Los Angeles, 1) retorna [California]

    Ancestor (Los Angeles, 2) retorna [USA]

MDX - Funções

- Função Siblings
  - Retorna o conjunto de irmãos de um dado membro, incluindo o próprio membro ao qual a função é aplicada
  - Retorna um conjunto de membros de um dado nível que possuem o mesmo pai
  - Sintaxe: [*membro*] . Siblings
  - Exemplo:

    [Time] .[All Time] .[1998] . [Q1] . [Jan] . Siblings retorna Jan, Fev e Mar
  - *FirstSibling* e *LastSibiling* são semelhantes à *FirstChild* e *LastChild*, respectivamente.

MDX - Funções

- Função Cousin
  - Recebe dois membros como argumento
  - Retorna o filho do segundo membro que possui a mesma posição relativa ao primeiro membro dado de entrada
  - Sintaxe: Cousin ([*membro1*] , [*membro2*])
  - Exemplos:

Cousin ([Time] .[1998] . [Q1] . [Jan] , [Time] .[1998] . [Q2]) retorna [Abr]

Cousin ([Time] .[1998] . [Q4] , [1997]) retorna [Q4] de 1997

Como a posição de cada membro determina quem são seus primos, a aplicação desta função requer simetria na dimensão!

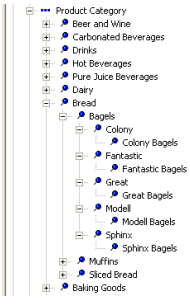
MDX - Exercícios

1. SELECT { [Time] . [1997] } ON COLUMNS ,
   Descendants ([Product] . [Bread] , [Product Subcategory]) ON ROWS
   FROM Sales
   WHERE ([Measures] . [Unit Sales])

	1997
Bagels	815,00
Muffins	3.497,00
Sliced Bread	3.558,00

MDX - Exercícios

2. SELECT { [Time] . [1997] } ON COLUMNS ,
   { Descendants ([Product] . [Bread]) } ON ROWS FROM
   Sales WHERE ([Measures] .[Unit Sales])



	1997
Bread	7.870,00
Bagels	815,00
Colony	163,00
Colony Bagels	163,00
Fantastic	160,00
Fantastic Bagels	160,00
Great	145,00
Great Bagels	145,00
Modell	165,00
Modell Bagels	165,00
Sphinx	182,00
Sphinx Bagels	182,00
Muffins	3.497,00
Colony	740,00
Colony Blueberry Muffins	193,00
Colony Cranberry Muffins	182,00
Colony English Muffins	161,00
Colony Muffins	204,00

MDX - Exercícios

3. SELECT
   { Cousin ([Time] . [1] , [Q2]) } ON COLUMNS ,
   { Descendants ([Product] . [All Products]) } ON ROWS
   FROM Sales

4. SELECT
   { Descendants ([Time] . [1997] . [Q1] . [1] , [Month]) } ON COLUMNS ,
   { [Customers] . [All Customers] } ON ROWS
   FROM Sales
   WHERE ([Measures] . [Unit Sales])

MDX - Exercícios

5. SELECT
   { [Product] . [All Products] } ON COLUMNS ,
   { [Time] . [1997] . [Q1] . Parent } ON ROWS
   FROM Sales

6. SELECT
   { Descendants ([Time] . [1997] , Month) } ON COLUMNS ,
   { [Customers] . [All Customers] } ON ROWS
   FROM Sales
   WHERE ([Measures] . [Unit Sales])

MDX - Funções

- Calculando a quantidade média

```
Sum ( Descendants ( [Time] . CurrentMember , [Month] ) ,
                                    [Measures] . [Unit Sales] ) /
Count ( Descendants ( [Time] . CurrentMember , [Month] ) )
```

- Função Avg
  - Retorna o valor médio de uma expressão numérica avaliada sobre um conjunto
  - Sintaxe: Avg ({*conjunto*} , *expressao_numerica*)

```
Avg ( Descendants ( [Time] . CurrentMember , [Month] ) ,
                                    [Measures] . [Unit Sales] )
```


MDX - Funções

- Obtendo a quantidade do último mês do período

```
(Descendants ( [Time] . CurrentMember , [Month] ), [Measures].[Unit Sales] )
```

Qual o problema com a expressão acima?

- Função Tail
  - Retorna o subconjunto do final de um conjunto
  - Recebe como entrada um conjunto e a quantidade dos n-últimos elementos a serem removidos do conjunto.
  - Sintaxe: Tail ({*conjunto*} , *n*)
  - Exemplo: Tail({USA, Canada, França, Alemanha, Japão }, 3)

MDX - Funções

- Obtendo a quantidade do último mês do período

```
( Tail ( Descendants ( [Time] . CurrentMember , [Month] ) , 1 ) ,
                                    [Measures] . [Unit Sales] )
```

A função *Tail* retorna um conjunto contendo o último elemento do conjunto de entrada

- Função Item
  - Localiza um dado membro em um conjunto
  - Indexa os elementos do conjunto com valores de 0, 1, 2, ...
  - Sintaxe: {*conjunto*} . Item (*numero*)
  - Exemplo: (Tail (Descendants ([Time] . CurrentMember , [Month]) , 1) . Item (0) , [Measures] . [Unit Sales])

MDX - Funções

- Obtendo a quantidade do último mês do período
- Função ClosingPeriod
  - Recebe como entrada um membro e um nível
  - Retorna o último irmão dos descendentes de um membro para um dado nível
  - Sintaxe: ClosingPeriod ([*nivel*] , [*membro*])
  - Existe também a função OpeningPeriod()
    - Possui os mesmos parâmetros de entrada
    - Retorna o primeiro irmão dos descendentes de um membro em um dado nível

MDX - Exemplo de ClosingPeriod

```
WITH MEMBER [Measures].[FechamentoVendas] AS
'(ClosingPeriod ([Month], [Time].CurrentMember),[Measures].[Unit Sales])'
SELECT { [Measures].[FechamentoVendas] } ON COLUMNS,
       { Descendants ( [Time] . [1997] ) } ON ROWS FROM Sales
```

	Unit Sales	Fechamento
1997	266.773,00	26.796,00
Q1	66.291,00	23.706,00
1	21.628,00	21.628,00
2	20.957,00	20.957,00
3	23.706,00	23.706,00
Q2	62.610,00	21.350,00
4	20.179,00	20.179,00
5	21.081,00	21.081,00
6	21.350,00	21.350,00
Q3	65.848,00	20.388,00
7	23.763,00	23.763,00
8	21.697,00	21.697,00
9	20.388,00	20.388,00
Q4	72.024,00	26.796,00
10	19.958,00	19.958,00
11	25.270,00	25.270,00
12	26.796,00	26.796,00

MDX - Exemplo de ClosingPeriod

```
WITH   MEMBER  [Measures].[FechamentoVendas]  AS
'(ClosingPeriod ([Month], [Time].CurrentMember),[Measures].[Unit Sales])'
SELECT   { [Measures].[FechamentoVendas]  } ON COLUMNS,
          { Descendants ( [Time] . [1997] ) }  ON ROWS   FROM Sales
```

	Unit Sales	Fechamento
1997	266.773,00	26.796,00
Q1	66.291,00	23.706,00
1	21.628,00	21.628,00
2	20.957,00	20.957,00
3	23.706,00	23.706,00
Q2	62.610,00	21.350,00
4	20.179,00	20.179,00
5	21.081,00	21.081,00
6	21.350,00	21.350,00
Q3	65.848,00	20.388,00
7	23.763,00	23.763,00
8	21.697,00	21.697,00
9	20.388,00	20.388,00
Q4	72.024,00	26.796,00
10	19.958,00	19.958,00
11	25.270,00	25.270,00
12	26.796,00	26.796,00

MDX - Exemplo de ClosingPeriod

```
WITH   MEMBER  [Measures].[FechamentoVendas]  AS
'(ClosingPeriod ([Month], [Time].CurrentMember),[Measures].[Unit Sales])'
SELECT   { [Measures].[FechamentoVendas]  } ON COLUMNS,
          { Descendants ( [Time] . [1997] ) }  ON ROWS   FROM Sales
```

	Unit Sales	Fechamento
1997	266.773,00	26.796,00
Q1	66.291,00	23.706,00
1	21.628,00	21.628,00
2	20.957,00	20.957,00
3	23.706,00	23.706,00
Q2	62.610,00	21.350,00
4	20.179,00	20.179,00
5	21.081,00	21.081,00
6	21.350,00	21.350,00
Q3	65.848,00	20.388,00
7	23.763,00	23.763,00
8	21.697,00	21.697,00
9	20.388,00	20.388,00
Q4	72.024,00	26.796,00
10	19.958,00	19.958,00
11	25.270,00	25.270,00
12	26.796,00	26.796,00

MDX - Exemplo de ClosingPeriod

```
WITH   MEMBER  [Measures].[FechamentoVendas]  AS
'(ClosingPeriod ([Month], [Time].CurrentMember),[Measures].[Unit Sales])'
SELECT   { [Measures].[FechamentoVendas]  } ON COLUMNS,
          { Descendants ( [Time] . [1997] ) }  ON ROWS   FROM Sales
```

	Unit Sales	Fechamento
1997	266.773,00	26.796,00
Q1	66.291,00	23.706,00
1	21.628,00	21.628,00
2	20.957,00	20.957,00
3	23.706,00	23.706,00
Q2	62.610,00	21.350,00
4	20.179,00	20.179,00
5	21.081,00	21.081,00
6	21.350,00	21.350,00
Q3	65.848,00	20.388,00
7	23.763,00	23.763,00
8	21.697,00	21.697,00
9	20.388,00	20.388,00
Q4	72.024,00	26.796,00
10	19.958,00	19.958,00
11	25.270,00	25.270,00
12	26.796,00	26.796,00

MDX - Exemplo de ClosingPeriod

```
WITH   MEMBER  [Measures].[FechamentoVendas]  AS
'(ClosingPeriod ([Month], [Time].CurrentMember),[Measures].[Unit Sales])'
SELECT   { [Measures].[FechamentoVendas]  } ON COLUMNS,
          { Descendants ( [Time] . [1997] ) }  ON ROWS   FROM Sales
```

	Unit Sales	Fechamento
1997	266.773,00	26.796,00
Q1	66.291,00	23.706,00
1	21.628,00	21.628,00
2	20.957,00	20.957,00
3	23.706,00	23.706,00
Q2	62.610,00	21.350,00
4	20.179,00	20.179,00
5	21.081,00	21.081,00
6	21.350,00	21.350,00
Q3	65.848,00	20.388,00
7	23.763,00	23.763,00
8	21.697,00	21.697,00
9	20.388,00	20.388,00
Q4	72.024,00	26.796,00
10	19.958,00	19.958,00
11	25.270,00	25.270,00
12	26.796,00	26.796,00

MDX - Funções

- Função Max
  - Recebe um conjunto e uma expressão numérica
  - Retorna o maior valor do conjunto computado pela expressão numérica
  - Sintaxe: Max ({*conjunto*}, *expressao_numerica*)
  - Exemplo: Se as quantidades vendidas para os países abaixo são 1000, 2000 e 3000, respectivamente:

Max ({ USA, Canada, Mexico}, [Measures]. [Unit Sales]) retorna 3000

Existe uma função semelhante Min que recebe os mesmos argumentos e retorna o menor valor do conjunto.

MDX - Exemplo de Max

```
WITH   MEMBER [Measures].[MaioresVendas]    AS
'(Max(Descendants([Time].CurrentMember),[Measures].[Unit Sales]))'
SELECT   { [Measures].[MaioresVendas]  } ON COLUMNS,
          { Descendants ( [Time] . [1997] ) }  ON ROWS   FROM  Sales
```

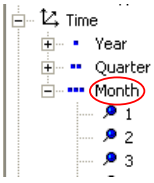
	Unit Sales	MaioresVendas
1997	266.773,00	26.796,00
Q1	66.291,00	23.706,00
1	21.628,00	21.628,00
2	20.957,00	20.957,00
3	23.706,00	23.706,00
Q2	62.610,00	21.350,00
4	20.179,00	20.179,00
5	21.081,00	21.081,00
6	21.350,00	21.350,00
Q3	65.848,00	23.763,00
7	23.763,00	23.763,00
8	21.697,00	21.697,00
9	20.388,00	20.388,00
Q4	72.024,00	26.796,00
10	19.958,00	19.958,00
11	25.270,00	25.270,00
12	26.796,00	26.796,00

MDX - Funções

- Função Members
  - Retorna o conjunto de membros de uma dimensão (ou hierarquia ou nível) na qual a função é aplicada
  - Sintaxe: [*Dimensao*]. Members
    [*Hierarquia*]. Members
    [*Nivel*]. Members
  - Exemplos:
    [Time] . Members
    [Time] . [Year] . [Quarter] . Members
    [Year] . Members

MDX - Funções

- Função Level
  - Retorna o nível de um membro
  - Sintaxe: [*Membro*]. Level
  - Exemplo:
    [1] . Level retorna Month



- Função Name
  - Retorna o nome de um nível, dimensão ou membro
  - Sintaxe: [*Nivel*] . Name
    [*Dimensao*] . *Name*
    [*Membro*] . *Name*

MDX - Funções

- Operador de Intervalo (:)

```
SELECT { [Time] . [1997] . [Q1] . [01], [Time] . [1997] . [Q1] . [02] ,
  [Time] . [1997] . [Q1] . [03], [Time] . [1997] . [Q2] . [04] ,
  [Time] . [1997] . [Q2] . [05], [Time] . [1997] . [Q2] . [06] } ON COLUMNS
FROM Sales
```

```
SELECT
{ [Time] . [1997] . [Q1] . [01] :
  [Time] . [1997] . [Q2] . [06] } ON COLUMNS
FROM Sales
```


MDX - Funções

- Calculando uma média móvel (*moving average*)

```
WITH MEMBER [Measures].[MediaMovel] AS
'( Avg ( [Time] . CurrentMember . Lag(2) : [Time] . CurrentMember ,
                                  [Measures].[Unit Sales] ) )'
SELECT { [Measures].[MediaMovel] } ON COLUMNS,
{ Descendants ( [Time] . [1997] , [Month]) } ON ROWS FROM Sales
```

	Unit Sales
1	21.628,00
2	20.957,00
3	23.706,00
4	20.179,00
5	21.081,00
6	21.350,00
7	23.763,00
8	21.697,00
9	20.388,00
10	19.958,00
11	25.270,00
12	26.796,00

	MediaMovel
1	21.628,00
2	21.292,50
3	22.097,00
4	21.614,00
5	21.655,33
6	20.870,00
7	22.064,67
8	22.270,00
9	21.949,33
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel (*moving average*)

```
WITH MEMBER [Measures].[MediaMovel] AS
'( Avg ( [Time] . CurrentMember . Lag(2) : [Time] . CurrentMember ,
                                  [Measures].[Unit Sales] ) )'
SELECT { [Measures].[MediaMovel] } ON COLUMNS,
{ Descendants ( [Time] . [1997] , [Month]) } ON ROWS FROM Sales
```

	Unit Sales
1	21.628,00
2	20.957,00
3	23.706,00
4	20.179,00
5	21.081,00
6	21.350,00
7	23.763,00
8	21.697,00
9	20.388,00
10	19.958,00
11	25.270,00
12	26.796,00

	MediaMovel
1	21.628,00
2	21.292,50
3	22.097,00
4	21.614,00
5	21.655,33
6	20.870,00
7	22.064,67
8	22.270,00
9	21.949,33
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel (*moving average*)

```
WITH MEMBER [Measures].[MediaMovel] AS
'( Avg ( [Time] . CurrentMember . Lag(2) : [Time] . CurrentMember ,
                                  [Measures].[Unit Sales] ) )'
SELECT { [Measures].[MediaMovel] } ON COLUMNS,
{ Descendants ( [Time] . [1997] , [Month]) } ON ROWS FROM Sales
```

	Unit Sales
1	21.628,00
2	20.957,00
3	23.706,00
4	20.179,00
5	21.081,00
6	21.350,00
7	23.763,00
8	21.697,00
9	20.388,00
10	19.958,00
11	25.270,00
12	26.796,00

	MediaMovel
1	21.628,00
2	21.292,50
3	22.097,00
4	21.614,00
5	21.655,33
6	20.870,00
7	22.064,67
8	22.270,00
9	21.949,33
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel (*moving average*)

```
WITH   MEMBER [Measures].[MediaMovel]   AS
'( Avg ( [Time] . CurrentMember . Lag(2)  :  [Time] . CurrentMember ,
                                        [Measures].[Unit Sales] ) )'
SELECT    { [Measures].[MediaMovel]  } ON COLUMNS,
{ Descendants ( [Time] . [1997] , [Month]) }  ON ROWS   FROM  Sales
```

	Unit Sales
1	21.628,00
2	20.957,00
3	23.706,00
4	20.179,00
5	21.081,00
6	21.350,00
7	23.763,00
8	21.697,00
9	20.388,00
10	19.958,00
11	25.270,00
12	26.796,00

	MediaMovel
1	21.628,00
2	21.292,50
3	22.097,00
4	21.614,00
5	21.655,33
6	20.870,00
7	22.064,67
8	22.270,00
9	21.949,33
10	20.681,00
11	21.872,00
12	24.008,00

E assim por diante...

MDX - Funções

- Calculando uma média móvel (*moving average*)

```
WITH   MEMBER [Measures].[MediaMovel]   AS
'( Avg ( [Time] . CurrentMember . Lag(2)  :  [Time] . CurrentMember ,
                                        [Measures].[Unit Sales] ) )'
SELECT    { [Measures].[MediaMovel]  } ON COLUMNS,
{ Descendants ( [Time] . [1997] , [Month]) }  ON ROWS   FROM  Sales
```

	Unit Sales
1	21.628,00
2	20.957,00
3	23.706,00
4	20.179,00
5	21.081,00
6	21.350,00
7	23.763,00
8	21.697,00
9	20.388,00
10	19.958,00
11	25.270,00
12	26.796,00

	MediaMovel
1	21.628,00
2	21.292,50
3	22.097,00
4	21.614,00
5	21.655,33
6	20.870,00
7	22.064,67
8	22.270,00
9	21.949,33
10	20.681,00
11	21.872,00
12	24.008,00

E quanto aos meses 1 e 2?

Como fazer *Lag()* operar diferentemente?

MDX - Funções

- Função Iif (*Immediate If*)
  - Recebe 3 parâmetros de entrada:
    - Expressão lógica a ser avaliada
    - String ou valor numérico retornado pela função se a expressão lógica for verdadeira
    - String ou valor numérico retornado pela função se a expressão lógica for falsa
  - Sintaxe: Iif (*expressao_logica* , *numerico1*, *numerico2*)

    Iif (*expressao_logica* , *string1*, *string2*)

```
Iif ( [Time].CurrentMember.Level  is  [Time].Month,  2 ,
      Iif ( [Time].CurrentMember.Level  is  [Time].Quarter ,  1, 0 )
```


MDX - Funções

- Calculando uma média móvel *inteligente*

```
WITH   MEMBER [Measures].[MMI]   AS
'(Avg ( [Time].CurrentMember  .Lag
( Iif  ( [Time].CurrentMember.Level  is  [Time].Month, 2,
      Iif  ( [Time].CurrentMember.Level  is  [Time].Quarter, 1, 0 ) ) )  :
                  [Time].CurrentMember,  [Measures].[Unit Sales] ) )'
SELECT    { [Measures].[MMI]  } ON COLUMNS,
          { Descendants ( [Time] . [1997]) }  ON ROWS
FROM  Sales
```


MDX - Funções

- Calculando uma média móvel *inteligente*

	Unit Sales
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	20.957,00
3	23.706,00
Q2	62.610,00
4	20.179,00
5	21.081,00
6	21.350,00
Q3	65.848,00
7	23.763,00
8	21.697,00
9	20.388,00
Q4	72.024,00
10	19.958,00
11	25.270,00
12	26.796,00

	MMI
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	21.292,50
3	22.097,00
Q2	64.450,50
4	21.614,00
5	21.655,33
6	20.870,00
Q3	64.229,00
7	22.064,67
8	22.270,00
9	21.949,33
Q4	68.936,00
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel *inteligente*

	Unit Sales
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	20.957,00
3	23.706,00
Q2	62.610,00
4	20.179,00
5	21.081,00
6	21.350,00
Q3	65.848,00
7	23.763,00
8	21.697,00
9	20.388,00
Q4	72.024,00
10	19.958,00
11	25.270,00
12	26.796,00

	MMI
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	21.292,50
3	22.097,00
Q2	64.450,50
4	21.614,00
5	21.655,33
6	20.870,00
Q3	64.229,00
7	22.064,67
8	22.270,00
9	21.949,33
Q4	68.936,00
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel *inteligente*

	Unit Sales
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	20.957,00
3	23.706,00
Q2	62.610,00
4	20.179,00
5	21.081,00
6	21.350,00
Q3	65.848,00
7	23.763,00
8	21.697,00
9	20.388,00
Q4	72.024,00
10	19.958,00
11	25.270,00
12	26.796,00

	MMI
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	21.292,50
3	22.097,00
Q2	64.450,50
4	21.614,00
5	21.655,33
6	20.870,00
Q3	64.229,00
7	22.064,67
8	22.270,00
9	21.949,33
Q4	68.936,00
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel *inteligente*

	Unit Sales
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	20.957,00
3	23.706,00
Q2	62.610,00
4	20.179,00
5	21.081,00
6	21.350,00
Q3	65.848,00
7	23.763,00
8	21.697,00
9	20.388,00
Q4	72.024,00
10	19.958,00
11	25.270,00
12	26.796,00

	MMI
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	21.292,50
3	22.097,00
Q2	64.450,50
4	21.614,00
5	21.655,33
6	20.870,00
Q3	64.229,00
7	22.064,67
8	22.270,00
9	21.949,33
Q4	68.936,00
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel *inteligente*

	Unit Sales
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	20.957,00
3	23.706,00
Q2	62.610,00
4	20.179,00
5	21.081,00
6	21.350,00
Q3	65.848,00
7	23.763,00
8	21.697,00
9	20.388,00
Q4	72.024,00
10	19.958,00
11	25.270,00
12	26.796,00

	MMI
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	21.292,50
3	22.097,00
Q2	64.450,50
4	21.614,00
5	21.655,33
6	20.870,00
Q3	64.229,00
7	22.064,67
8	22.270,00
9	21.949,33
Q4	68.936,00
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel *inteligente*

	Unit Sales
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	20.957,00
3	23.706,00
Q2	62.610,00
4	20.179,00
5	21.081,00
6	21.350,00
Q3	65.848,00
7	23.763,00
8	21.697,00
9	20.388,00
Q4	72.024,00
10	19.958,00
11	25.270,00
12	26.796,00

	MMI
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	21.292,50
3	22.097,00
Q2	64.450,50
4	21.614,00
5	21.655,33
6	20.870,00
Q3	64.229,00
7	22.064,67
8	22.270,00
9	21.949,33
Q4	68.936,00
10	20.681,00
11	21.872,00
12	24.008,00

MDX - Funções

- Calculando uma média móvel *inteligente*

	Unit Sales
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	20.957,00
3	23.706,00
Q2	62.610,00
4	20.179,00
5	21.081,00
6	21.350,00
Q3	65.848,00
7	23.763,00
8	21.697,00
9	20.388,00
Q4	72.024,00
10	19.958,00
11	25.270,00
12	26.796,00

	MMI
1997	266.773,00
Q1	66.291,00
1	21.628,00
2	21.292,50
3	22.097,00
Q2	64.450,50
4	21.614,00
5	21.655,33
6	20.870,00
Q3	64.229,00
7	22.064,67
8	22.270,00
9	21.949,33
Q4	68.936,00
10	20.681,00
11	21.872,00
12	24.008,00

E assim por diante...

MDX - Funções

- Função Filter
  - Retorna o conjunto resultante da verificação de uma condição sobre um outro conjunto dado de entrada
  - Sintaxe: Filter ({*Conjunto*} , *Condiçao*)
  - Exemplo:

```
Count (Filter ( Descendants ( [Product] . CurrentMember, [Product Name] ) ,
  ( [Time].CurrentMember, [Unit Sales] ) <
  ( [Time].CurrentMember.PrevMember, [Unit Sales] ) ) ) /
Count (Descendants ( [Product] . CurrentMember, [Product Name] ) )
```

MDX - Funções

- Função Filter
  - Retorna o conjunto resultante da verificação de uma condição sobre um outro conjunto dado de entrada
  - Sintaxe: Filter ({*Conjunto*} , *Condiçao*)
  - Exemplo:

```
Count (Filter ( Descendants ( [Product] . CurrentMember, [Product Name] ) ,
  ( [Time].CurrentMember, [Unit Sales] ) <
  ( [Time].CurrentMember.PrevMember, [Unit Sales] ) ) ) /
Count (Descendants ( [Product] . CurrentMember, [Product Name] ) )
```


MDX - Funções

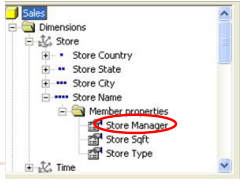
- Função Filter
  - Retorna o conjunto resultante da verificação de uma condição sobre um outro conjunto dado de entrada
  - Sintaxe: Filter ({*Conjunto*} , *Condiçao*)
  - Exemplo:

```
Count (Filter ( Descendants ( [Product] . CurrentMember, [Product Name] ) ,
  ( [Time].CurrentMember, [Unit Sales] ) <
  ( [Time].CurrentMember.PrevMember, [Unit Sales] ) ) ) /
Count (Descendants ( [Product] . CurrentMember, [Product Name] ) )
```


MDX - Funções

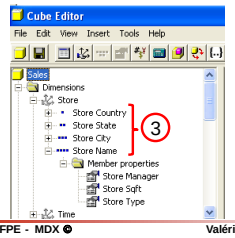
- Função Properties
  - Retorna uma *string* contendo o valor da propriedade de um membro
  - Recebe como parâmetro o nome da propriedade
  - Sintaxe: [*Membro*] . Properties (*string*)
  - Exemplo: Store . CurrentMember . Properties ("Store Manager")

No Editor de Cubos...



MDX – Exemplo de Properties

```
WITH MEMBER [Measures].[Store Mgr] AS
'Store.CurrentMember.Properties("Store Manager")'
SELECT { [Time] . [1998] } ON COLUMNS,
{ Descendants ( [Store].[All Stores].[USA] , 3 ) } ON ROWS
FROM Sales WHERE ([Measures].[Store Mgr])
```



	1998
HQ	
Store 6	Maris
Store 7	White
Store 24	Byrd
Store 14	Strehlo
Store 11	Erickson
Store 13	Inmon
Store 2	Smith
Store 3	Davis
Store 15	Ollom
Store 16	Mantle
Store 17	Mays
Store 22	Byrg
Store 23	Johnson

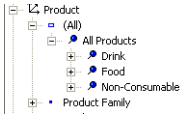
MDX - Funções

- Atribuindo Nomes a Conjuntos

```
WITH SET [Kids] AS '{[Product].[All Products].Children}'
SELECT { [Kids] } ON COLUMNS,
{ [Measures].[Unit Sales] } ON ROWS FROM Sales
```

```
SELECT { [Product] . [All Products].Children } ON COLUMNS,
{ [Measures].[Unit Sales] } ON ROWS FROM Sales
```

	Drink	Food	Non-Consumable
Unit Sales	24.597,00	191.940,00	50.236,00



MDX – Função Crossjoin

- Considere a consulta abaixo

```
SELECT { [Time] . [1997].Children } ON COLUMNS,
{ [Product].[All Products] . Children } ON ROWS FROM Sales
WHERE ( [Measures].[Unit Sales] )
```

	Q1	Q2	Q3	Q4
Drink	5.976,00	5.895,00	6.065,00	6.661,00
Food	47.809,00	44.825,00	47.440,00	51.866,00
Non-Consumable	12.506,00	11.890,00	12.343,00	13.497,00

Como exibir duas dimensões em um mesmo eixo?

CROSSJOIN ({*Conjunto* } , { *Conjunto* })

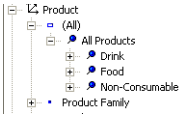
{ {*Conjunto* } * {*Conjunto* } }

MDX – Função Crossjoin

- Considere a consulta abaixo

```
SELECT { [Time] . [1997].Children } ON COLUMNS,
{ [Product].[All Products] . Children } ON ROWS FROM Sales
WHERE ( [Measures].[Unit Sales] )
```

	Q1	Q2	Q3	Q4
Drink	5.976,00	5.895,00	6.065,00	6.661,00
Food	47.809,00	44.825,00	47.440,00	51.866,00
Non-Consumable	12.506,00	11.890,00	12.343,00	13.497,00



Como exibir duas dimensões em um mesmo eixo?

CROSSJOIN ({*Conjunto* } , { *Conjunto* })

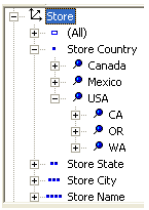
{ {*Conjunto* } * {*Conjunto* } }

MDX – Função Crossjoin

- Exemplo

```
SELECT { [Time] . [1997].Children } ON COLUMNS,
CROSSJOIN ( { [Store] . [Store Country] . [USA] . Children} ,
                { [Product]. [All Products] . Children } ) ON ROWS
FROM Sales WHERE ( [Measures].[Unit Sales] )
```

		Q1	Q2	Q3	Q4
CA	Drink	1.654,00	1.608,00	1.792,00	2.048,00
	Food	12.064,00	13.074,00	13.135,00	15.383,00
	Non-Consumable	3.172,00	3.370,00	3.443,00	4.005,00
OR	Drink	1.643,00	1.478,00	1.486,00	1.499,00
	Food	13.737,00	10.726,00	12.325,00	11.749,00
	Non-Consumable	3.907,00	2.875,00	3.129,00	3.105,00
WA	Drink	2.679,00	2.809,00	2.787,00	3.114,00
	Food	22.008,00	21.025,00	21.980,00	24.734,00
	Non-Consumable	5.427,00	5.645,00	5.771,00	6.387,00



MDX – Função Crossjoin

- Mesmo exemplo anterior mas com asterisco

```
SELECT { [Time] . [1997].Children } ON COLUMNS,
{ { [Store] . [Store Country] . [USA] . Children} *
                { [Product]. [All Products] . Children } } ON ROWS
FROM Sales WHERE ( [Measures].[Unit Sales] )
```

- Exemplo semelhante

```
SELECT
CROSSJOIN ( { [Store] . [Store Country] . [USA] . Children} ,
                { [Product]. [All Products] . Children } ) ON COLUMNS,
                { [Time] . [1997].Children } ON ROWS
FROM Sales WHERE ( [Measures].[Unit Sales] )
```


MDX – Função Crossjoin

- Usando duas vezes na mesma consulta

```
SELECT CROSSJOIN ( { [Store] . [Store Country] . [USA] . Children} ,
                { [Product] . [All Products] . Children } ) ON COLUMNS,
CROSSJOIN ( { [Time] . [1997].Children } ,
                { [Customers] . [USA] . Children} ) ON ROWS
FROM Sales WHERE ( [Measures].[Unit Sales] )
```

- Pode ser encadeado

```
SELECT CROSSJOIN ( { [Product]. [All Products] . Children} , {
            CROSSJOIN ( { [Time] . [1997].Children } ,
                { [Store] .[Store Country]. [USA] . Children} ) } )
                ON COLUMNS
FROM Sales WHERE ( [Measures].[Unit Sales] )
```


MDX - Funções

- Eliminando células vazias

```
SELECT { [Product]. [All Products] . Children} ON COLUMNS ,
        { [Customers].[All Customers] . Children} ON ROWS
FROM Sales
```

	Drink	Food	Non-Consumable
Canada			
Mexico			
USA	24.597,00	191.940,00	50.236,00

```
SELECT NON EMPTY { [Product].[All Products]. Children} ON COLUMNS ,
        NON EMPTY { [Customers].[All Customers].Children} ON ROWS
FROM Sales
```

	Drink	Food	Non-Consumable
USA	24.597,00	191.940,00	50.236,00

MDX - Funções

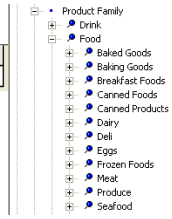
- Classificando o resultado da consulta
- Função Topcount
  - Recebe como entrada 3 parâmetros
    - Um conjunto
    - Um número *N*
    - Uma expressão aritmética
  - Retorna o conjunto dos *N* elementos que possuem o maior valor computado pela expressão aritmética
  - Sintaxe: Topcount ({*conjunto*} , *N* , *expressao_aritmetica*)
  - Ela quebra a hierarquia

MDX - Funções

- Função TOPCOUNT
  - Exemplo:

SELECT {Topcount ([Product] . [Product Family] . [Food] . Children, 4, [Measures].[Unit Sales]) } ON COLUMNS
FROM Sales

Produce	Snack Foods	Frozen Foods	Baking Goods
37.792,00	30.545,00	26.655,00	20.245,00



MDX - Funções

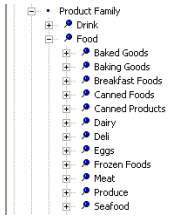
- Classificando o resultado da consulta
- Função Bottomcount
  - Recebe como entrada 3 parâmetros
    - Um conjunto
    - Um número *N*
    - Uma expressão aritmética
  - Retorna o conjunto dos *N* elementos que possuem o menor valor computado pela expressão aritmética
  - Sintaxe: Bottomcount({*conjunto*}, *N*, *expressao_aritmetica*)
  - Ela quebra a hierarquia

MDX - Funções

- Função Bottomcount
  - Exemplo:

SELECT {Bottomcount ([Product] . [Product Family] . [Food] . Children, 2, [Measures].[Unit Sales]) } ON COLUMNS ,
{ [Store] .[Store Country] .[USA] .Children} ON ROWS
FROM Sales

	Meat	Seafood
CA	527,00	441,00
OR	469,00	451,00
WA	718,00	872,00



MDX - Funções

- Ordenando o resultado da consulta
- Função Order
  - Recebe como entrada 3 parâmetros
    - Um conjunto
    - Um critério de ordenação (ASC, DESC, BASC, BDESC)
    - Uma expressão aritmética
  - Retorna o conjunto ordenado segundo o critério dado
  - Sintaxe: Order({*conjunto*}, *expressao_aritmetica* , *criterio*)
  - Pode opcionalmente quebrar a hierarquia

MDX - Funções

- Função Order
  - Possui duas variações:
    - Ordenação hierárquica (ASC, DESC)
      - Primeiro organiza os membros de acordo com a posição deles na hieraquia
      - Em seguida, ordena os níveis
    - Ordenação não hierárquica (BASC, BDESC)
      - B denota *Break hierarchy*
      - Organiza os membros em um conjunto e os ordena
  - ASC é a opção *default*

MDX - Funções

- Função Order
  - Exemplo:

SELECT {Order ([Product] . [All Products] . Children, [Measures].[Unit Sales], ASC) } ON COLUMNS ,
{ [Store] .[Store Country] .[USA] .Children} ON ROWS
FROM Sales

	Drink	Non-Consumable	Food
CA	7.102,00	13.990,00	53.656,00
OR	6.106,00	13.016,00	48.537,00
WA	11.389,00	23.230,00	89.747,00

MDX - Funções

- Navegando na hierarquia
- Função DrillDownMember
  - Recebe dois conjuntos como entrada
  - Realiza *drill down* em um conjunto de membros que pertencem ao segundo conjunto
  - Sintaxe: DrillDownMember({*conjunto1*}, {*conjunto2*})
- Função DrillUpMember
  - Realiza *drill up* em um conjunto de membros que pertencem ao segundo conjunto
  - Sintaxe: DrillUpMember({*conjunto1*}, {*conjunto2*})

MDX - Funções

- Função DrillDownMember
  - Exemplo:

```
SELECT { DrillDownMember( { [Time] . [1997] . [Q1] ,
    [Time] . [1997] . [Q2], [Time] . [1997] . [Q3], [Time] . [1997] . [Q4] } ,
    { [Time] . [1997] . [Q1] } ) } ON COLUMNS
FROM Sales
```

Q1	1	2	3	Q2	Q3	Q4
66.291,00	21.628,00	20.957,00	23.706,00	62.610,00	65.848,00	72.024,00

MDX - Funções

- Função DrillDownLevel
  - Recebe um conjunto como entrada
  - Realiza *drill down* no conjunto de entrada para um nível abaixo do menor nível encontrado no conjunto
  - Sintaxe: DrillDownLevel ({*conjunto*})
  - Exemplo:

```
SELECT { DrillDownLevel( { [Time].[Quarter].Members } ) } ON COLUMNS
FROM Sales
```

Conjunto de Entrada = { Q1, Q2, Q3 ,Q4 }
Menor Nível = Trimestre
Nível Abaixo = Mês

MDX - Funções

- Função DrillUpLevel
  - Realiza drill up no conjunto de entrada para um nível acima do menor nível encontrado no conjunto
  - Sintaxe: DrillUpLevel({*conjunto*})
  - Exemplo:

```
SELECT { DrillUpLevel ( { [Time] . [1997] . [Q1], [Time] . [1997],
  [Time].[1997] .[Q1] .[2] } ) } ON COLUMNS  FROM Sales
```

Q1	1997
66.291,00	266.773,00

MDX – Mais de dois eixos

- Usando mais de duas dimensões
  - ON COLUMNS e ON ROWS especificam como os dados do cubo são exibidos em dois eixos
  - É possível especificar três eixos adicionais em MDX
    - PAGES, SECTIONS e CHAPTERS
  - Exemplo:

```
SELECT { [Customers]. [All Customers]. [USA]. Children } ON COLUMNS ,
       { [Product]. [All Products]. Children } ON ROWS ,
       { [Time]. [1998]. Children } ON PAGES
FROM Sales
```

MDX Sample Application não permite mais de 2 dimensões!

MDX – Mais de dois eixos

- Usando mais de duas dimensões
  - MDX permite a especificação de até 128 dimensões!
  - Para usar mais de 5 dimensões na consulta deve-se seguir uma convenção de nomes alternativa baseada em números
    - AXIS (0) (ou apenas 0) para colunas
    - AXIS (1) (ou apenas 1) para linhas
    - AXIS (2) (ou apenas 2) para páginas
    - AXIS (3) (ou apenas 1) para seções
    - AXIS (4) (ou apenas 4) para capítulos
    - E assim por diante até AXIS(127) ou apenas 127!

MDX – Mais de dois eixos

- Exemplos

```
SELECT { [Customers]. [All Customers]. [USA]. Children } ON AXIS(0) ,
       { [Product]. [All Products]. Children } ON AXIS(1) ,
       { [Time]. [1998]. Children } ON AXIS(2) ,
       { [Store]. [USA]. Children } ON AXIS(3) ,
       { [Measures]. Members } ON AXIS(4)      FROM Sales

SELECT { [Customers]. [All Customers]. [USA]. Children } ON  0 ,
       { [Product]. [All Products]. Children } ON  1 ,
       { [Time]. [1998]. Children } ON  2 ,
       { [Store]. [USA]. Children } ON 3 ,
       { [Measures]. Members } ON  4 ,  FROM Sales
```


MDX – Mais de dois eixos

- Usando mais de duas dimensões
  - O que não pode ser feito?
    - Ordem deve ser respeitada independente da convenção de nomes escolhida
      - Um dado eixo não pode ser pulado (i.e. 1, 3, 4)
      - *Chapters* não pode ser especificado se na consulta não existe um eixo *Sections*
  - É possível que muitos usuários prefiram visualizar resultados apenas de forma bi-dimensional.

MDX – Bibliografia básica

- Referências
  - MDX at First Glance: Introduction to SQL Server MDX Essentials
    http://www.databasejournal.com/features/mssql/article.php/1495511
  - MDX Solutions with Microsoft SQL Server Analysis Services. George Spofford. John Wiley & Sons.
  - MDX Language Reference (MDX)
    - http://msdn2.microsoft.com/en-us/library/ms145595.aspx

Projeto Prático

Projeto Prático

- Roteiro para desenvolvimento do Projeto BD MD
  - Criar Minimundo e definir modelo ER da aplicação operacional
  - Mapeamento do ER para o esquema MultiDim
  - Criar o esquema lógico correspondente
  - Definir o esquema do cubo a ser criado e implementar o cubo no SQL Server
  - Implementar consultas MDX usando os operadores e funções OLAP vistos na aula
  - Testar e colocar o sistema em funcionamento para permitir a declaração das consultas e visualização dos resultados
  - Datas de Entrega: 27 / 10 / 15 e 29 / 10 / 15

Projeto Prático

- Roteiro para a nota máxima
  - Descrição de minimundo, modelagem conceitual e estrela e do cubo ⇒ Corretamente
  - Implementar consultas MDX usando:
    - Sets,Tuples, vírgula (,) e dois pontos (:)
    - Children, Descendants, FirstChild, Parent
    - TopCount, TopPercent, TopSum
    - DrilldownMember, DrilldownLevel
    - Count, Max, Sum, Avg
    - Crossjoint, NonEmptyCrossJoin, Non Empty
    - CurrentMember, Members
    - Hierarchize, Order, Rank
    - Properties


cin.ufpe.br
Centro de Informática
U·F·P·E
Universidade Federal de Pernambuco
129