QUERYING WITH MDX

TABLE OF FACTS → "SALES" CUBE

Dimension	Level (less to great detail)	Description
Customers	Country ← State or Province← City ← Name	Geographical hierarchy for registered customers of our stores.
Gender	Gender	"M" or "F"
Marital Status	Marital Status	"S" or "M"
Yearly Income	Yearly Income	Income of customer.
Product	Product Family ← Product Category ← Product Subcategory ← Brand Name ← Product Name	Products on sale in the *FoodMart* stores.
Promotion Media	Media Type	The media used for promotions
Promotions	Promotion Name	Identifies promotion that triggered the sale.
Store	Store Country ← Store State ← Store City ← Store Name	Geographical location of stores.
Store Size in SQFT	Store Square Feet	Area occupied by store, in square feet.
Store Type	Store Type	Type of store: "Deluxe Supermarket", "Small Grocery", etc.
Time	Year ← Quarter ← Month	Period of the sales.
Measures		
Unit Sales	Number of units sold.	
Store Cost	Cost of goods sold.	
Store Sales	Value of sales transactions.	

	MDX Queries

1	```SELECT Measures.MEMBERS ON COLUMNS,```
	``` [Store].MEMBERS ON ROWS```
	```FROM [Sales]```

2	```SELECT Measures.MEMBERS ON COLUMNS,```
	``` {[Store].[Store State].[CA],```
	``` [Store].[Store State].[WA]} ON ROWS```
	```FROM [Sales]```

3	```SELECT Measures.MEMBERS ON COLUMNS,```
	``` {[Store].[Store State].[CA].CHILDREN,```
	``` [Store].[Store State].[WA].CHILDREN} ON ROWS```
	```FROM [Sales]```

4	```SELECT Measures.MEMBERS ON COLUMNS,```
	``` {[Store].[Store State].[CA],```
	``` DESCENDANTS([Store].[Store State].[CA], [Store City])} ON ROWS```
	```FROM [Sales]```

5	```SELECT {[Store Type].[Store Type].MEMBERS} ON COLUMNS,```
	``` {[Store].[Store State].MEMBERS} ON ROWS```
	```FROM [Sales]```
	```WHERE (Measures.[Store Sales], [Time].[Year].[1997])```

| 6 | ```
WITH MEMBER Measures.Profit AS
 '(Measures.[Store Sales] - Measures.[Store Cost]) /(Measures.[Store Cost])'
 MEMBER [Time].[First Half 97]  AS '[Time].[1997].[Q1] + [Time].[1997].[Q2]'
 MEMBER [Time].[Second Half 97] AS '[Time].[1997].[Q3] + [Time].[1997].[Q4]'

SELECT {[Time].[First Half 97], [Time].[Second Half 97],
        [Time].[1997].CHILDREN} ON COLUMNS,
       {[Store].[Store Name].MEMBERS} ON ROWS
FROM [Sales]
``` |
| 7 | ```
WITH SET [Quarter1] AS
   'GENERATE([Time].[Year].MEMBERS, {[Time].CURRENTMEMBER.FIRSTCHILD})'

SELECT [Quarter1] ON COLUMNS,
       [Store].[Store Name].MEMBERS ON ROWS
FROM   [Sales] WHERE (Measures.[Profit])
``` |
| 8 | ```
WITH MEMBER Measures.PercentageSales AS
   '([Product].CURRENTMEMBER, Measures.[Unit Sales]) /
    ([Product].CURRENTMEMBER.PARENT, Measures.[Unit Sales])'

SELECT {Measures.[Unit Sales], Measures.PercentageSales} ON COLUMNS,
       [Product].[Brand Name].MEMBERS ON ROWS
FROM [Sales]
``` |
| | ```
WITH MEMBER Measures.PercentageSales AS
 '([Product].CURRENTMEMBER, Measures.[Unit Sales]) /
  (ANCESTOR([Product].CURRENTMEMBER, [Product Category]),Measures.[Unit Sales])'
WITH SET [PromotionSales] AS
    'EXCEPT({[Promotions].[All Promotions].CHILDREN},
            {[Promotions].[No Promotion]})'
    MEMBER Measures.PercentageSales AS
      '([Promotions].CURRENTMEMBER, Measures.[Unit Sales]) /
       SUM([PromotionSales], Measures.[Unit Sales])',
      FORMAT_STRING = '#.00%'

SELECT {Measures.[Unit Sales], Measures.PercentageSales} ON COLUMNS,
       [PromotionSales] ON ROWS
FROM [Sales]
``` |
| 9 | ```
WITH MEMBER Measures.[Profit Growth] AS
    '(Measures.[Profit]) - (Measures.[Profit], [Time].PREVMEMBER)'

SELECT {Measures.[Profit], Measures.[Profit Growth]} ON COLUMNS,
       {DESCENDANTS([Time].[1997], [Month])} ON ROWS
FROM [Sales]
``` |
| 10 | ```
SELECT [Product].[Product Family].MEMBERS ON COLUMNS,
{CROSSJOIN([Customers].[City].MEMBERS, [Time].[Quarter].MEMBERS)} ON ROWS
FROM [Sales]
WHERE (Measures.[Unit Sales])
``` |
| 11 | ```
SELECT NON EMPTY {[Store Type].[Store Type].MEMBERS} ON COLUMNS,
       FILTER({[Store].[Store City].MEMBERS},
              (Measures.[Unit Sales], [Time].[1997])>25000) ON ROWS
FROM [Sales] WHERE (Measures.[Profit], [Time].[Year].[1997])
``` |
| 12 | ```
SELECT Measures.MEMBERS ON COLUMNS,
HEAD(ORDER({[Store].[Store City].MEMBERS},Measures.[Unit Sales], BDESC), 12)
ON ROWS
FROM [Sales]
``` |

BDESC sort members in descendant order breaking their hierarchical order.
CROSSJOIN is the Cartesian product of two sets.