Multidimensional Databases

Microsoft SQL Server and MDX

Peter Scheuermann

MS SQL Server 2008

- Microsoft’s RDBMS
  - Runs on Windows OS only
- Nice features built-in
  - Analysis Services
  - Integration Services
  - Reporting Services
- Easy to use
  - Graphical “Management Studio” and “BI Developer Studio”

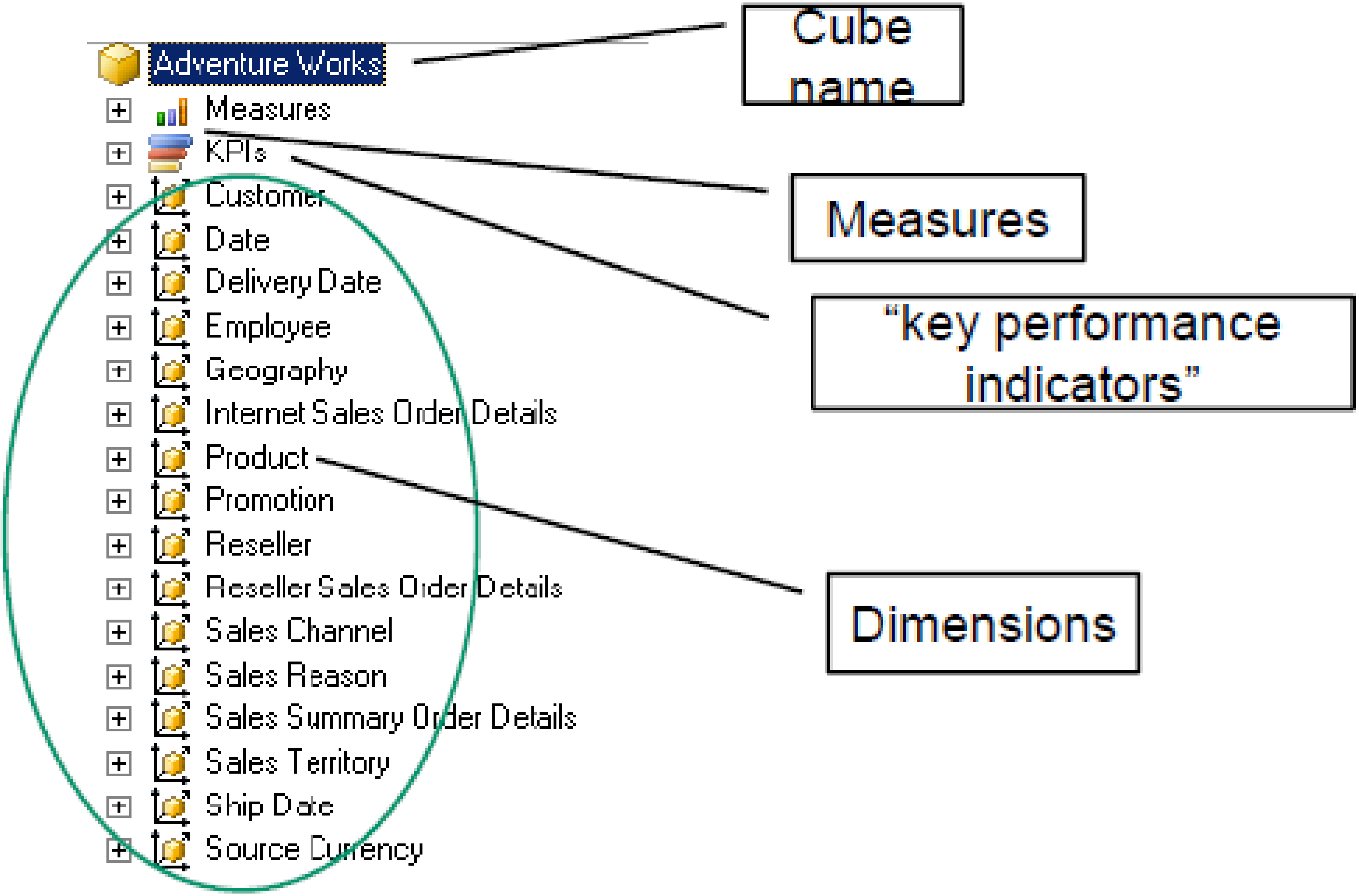
MS Analysis Services

- Cheap, easy to use, good, and widely used
- Supports ROLAP, MOLAP, HOLAP technology
- Intelligent pre-aggregation (for improving query performance)
- Programming: MS OLE DB for OLAP interface
- Uses the query language MDX (***M***ulti***D***imensional e***X***pressions)

SQL Server Data types

- Character data
  - CHAR, VARCHAR, ……
- Binary data
  - BINARY, VARBINARY, ……
- Date and time data
  - DATETIME, SMALLDATETIME
  - DATEADD(SS,dwml.dbo.[sales].[date],'19700101') converts UNIX time
- Numeric data
  - INT, FLOAT, ……
- Keys: IDENTITY property generates unique integer keys
  - Useful for generating DW (surrogate) keys during ETL

SSAS(SQL Server Analysis Service) –Data Cubes



Adventure Works

- Measures
- KPIs
- Customer
- Date
- Delivery Date
- Employee
- Geography
- Internet Sales Order Details
- Product
- Promotion
- Reseller
- Reseller Sales Order Details
- Sales Channel
- Sales Reason
- Sales Summary Order Details
- Sales Territory
- Ship Date
- Source Currency

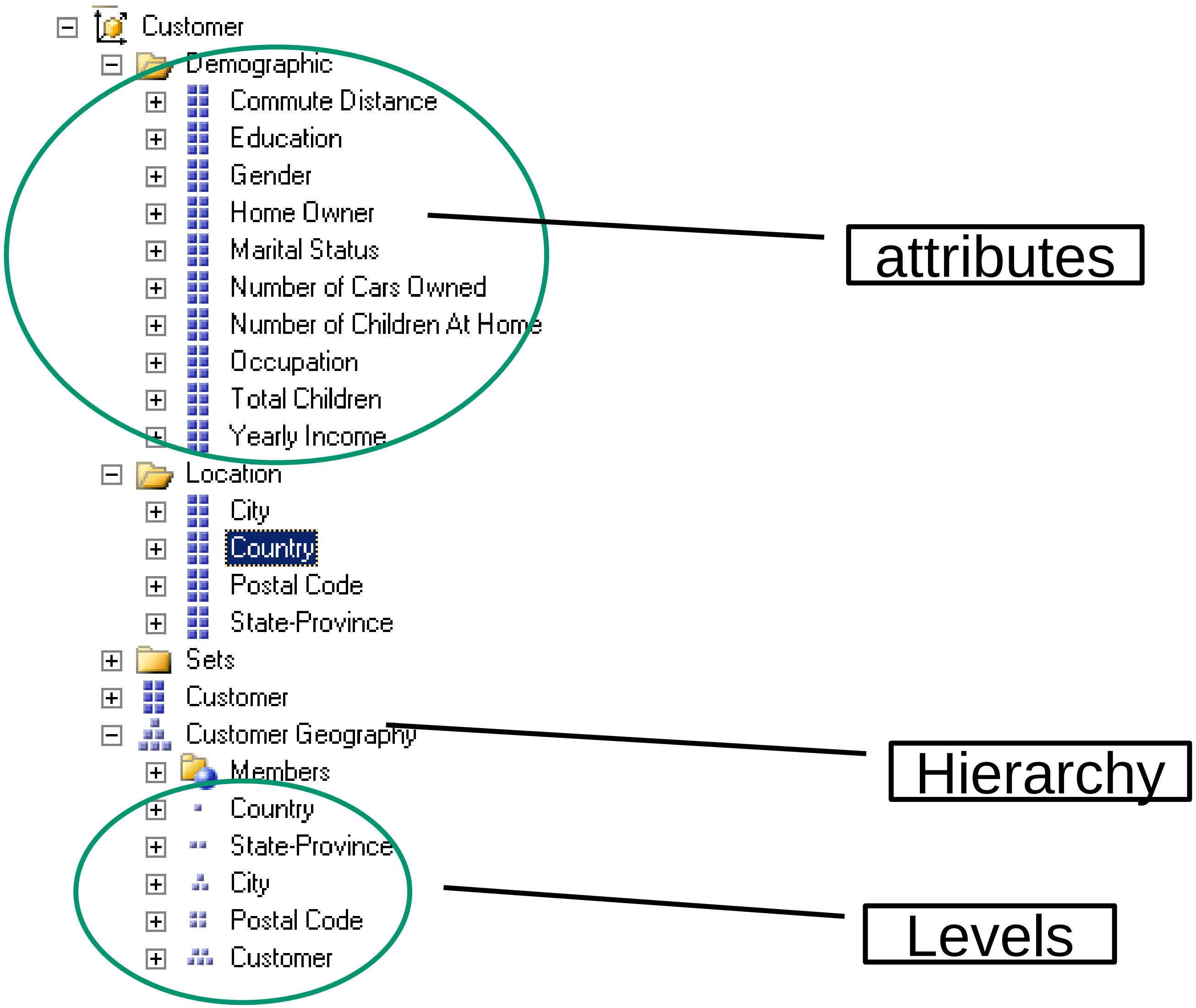
Cube name

Measures

"key performance indicators"

Dimensions

SSAS - Dimension



- Customer
  - Demographic
    - Commute Distance
    - Education
    - Gender
    - Home Owner
    - Marital Status
    - Number of Cars Owned
    - Number of Children At Home
    - Occupation
    - Total Children
    - Yearly Income
  - Location
    - City
    - Country
    - Postal Code
    - State-Province
  - Sets
  - Customer
  - Customer Geography
    - Members
    - Country
    - State-Province
    - City
    - Postal Code
    - Customer

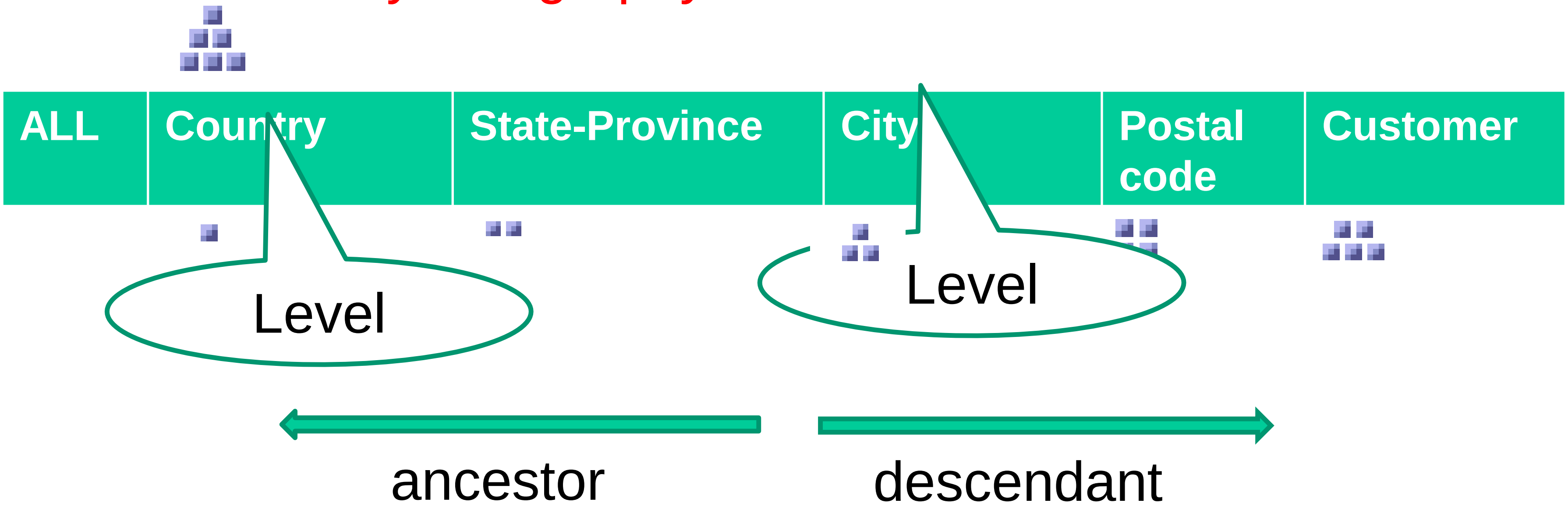
attributes

Hierarchy

Levels

Hierarchy, Level

Hierarchy Geography on Dimension Customer



ALL | Country | State-Province | City | Postal code | Customer

Level

Level

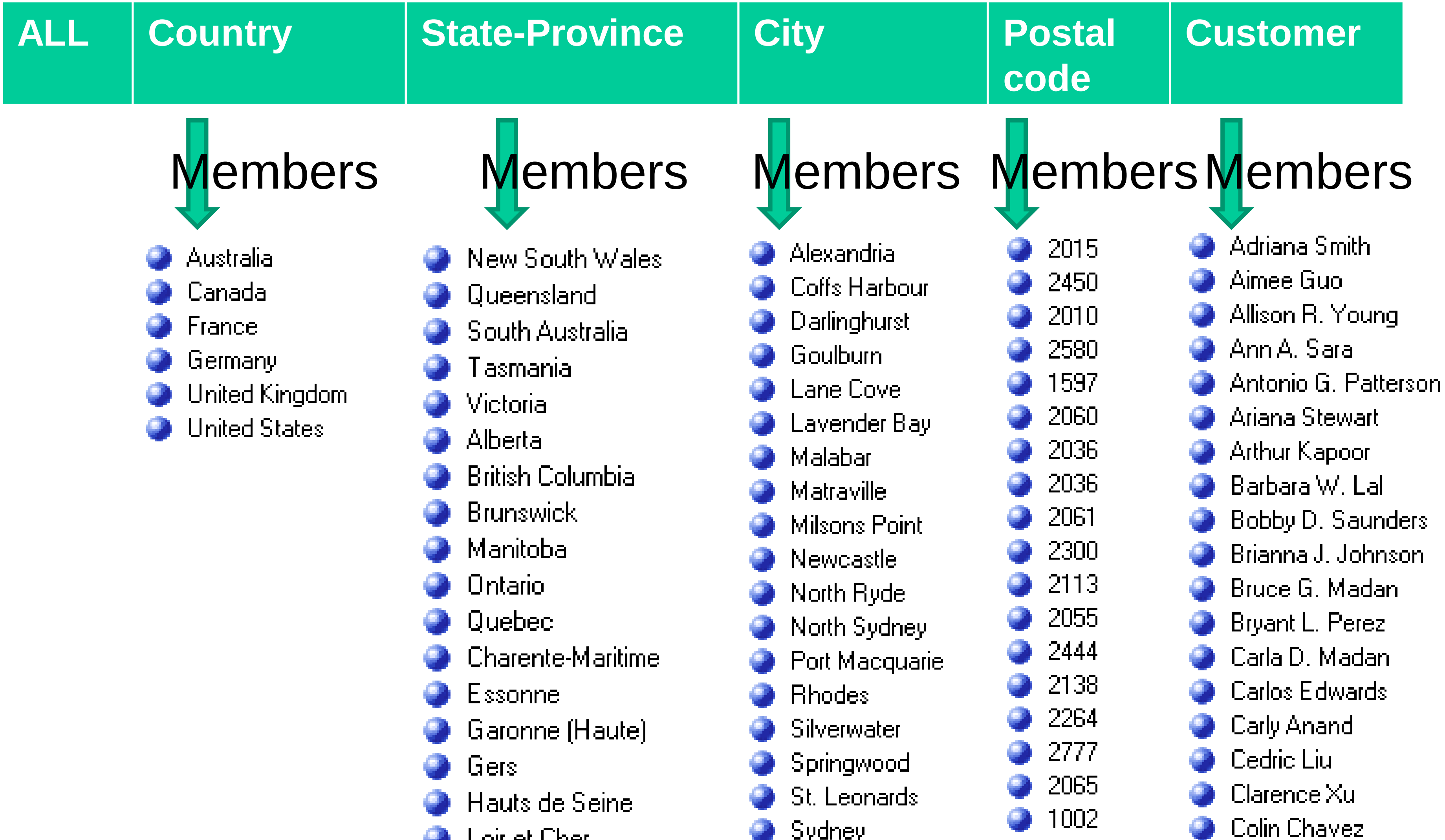
ancestor

descendant

- One dimension can have multiple hierarchies
- Hierarchies consist of *levels*
- Levels are in a linear order

Member

Hierarchy Geography on Dimension Customer



ALL	Country	State-Province	City	Postal code	Customer

Members

- Australia
- Canada
- France
- Germany
- United Kingdom
- United States

Members

- New South Wales
- Queensland
- South Australia
- Tasmania
- Victoria
- Alberta
- British Columbia
- Brunswick
- Manitoba
- Ontario
- Quebec
- Charente-Maritime
- Essonne
- Garonne (Haute)
- Gers
- Hauts de Seine
- Loir et Cher

Members

- Alexandria
- Coffs Harbour
- Darlinghurst
- Goulburn
- Lane Cove
- Lavender Bay
- Malabar
- Matraville
- Milsons Point
- Newcastle
- North Ryde
- North Sydney
- Port Macquarie
- Rhodes
- Silverwater
- Springwood
- St. Leonards
- Sydney

Members

- 2015
- 2450
- 2010
- 2580
- 1597
- 2060
- 2036
- 2036
- 2061
- 2300
- 2113
- 2055
- 2444
- 2138
- 2264
- 2777
- 2065
- 1002

Members

- Adriana Smith
- Aimee Guo
- Allison R. Young
- Ann A. Sara
- Antonio G. Patterson
- Ariana Stewart
- Arthur Kapoor
- Barbara W. Lal
- Bobby D. Saunders
- Brianna J. Johnson
- Bruce G. Madan
- Bryant L. Perez
- Carla D. Madan
- Carlos Edwards
- Carly Anand
- Cedric Liu
- Clarence Xu
- Colin Chavez

Sample Star Schema of Sales Cube

Dimension tables:

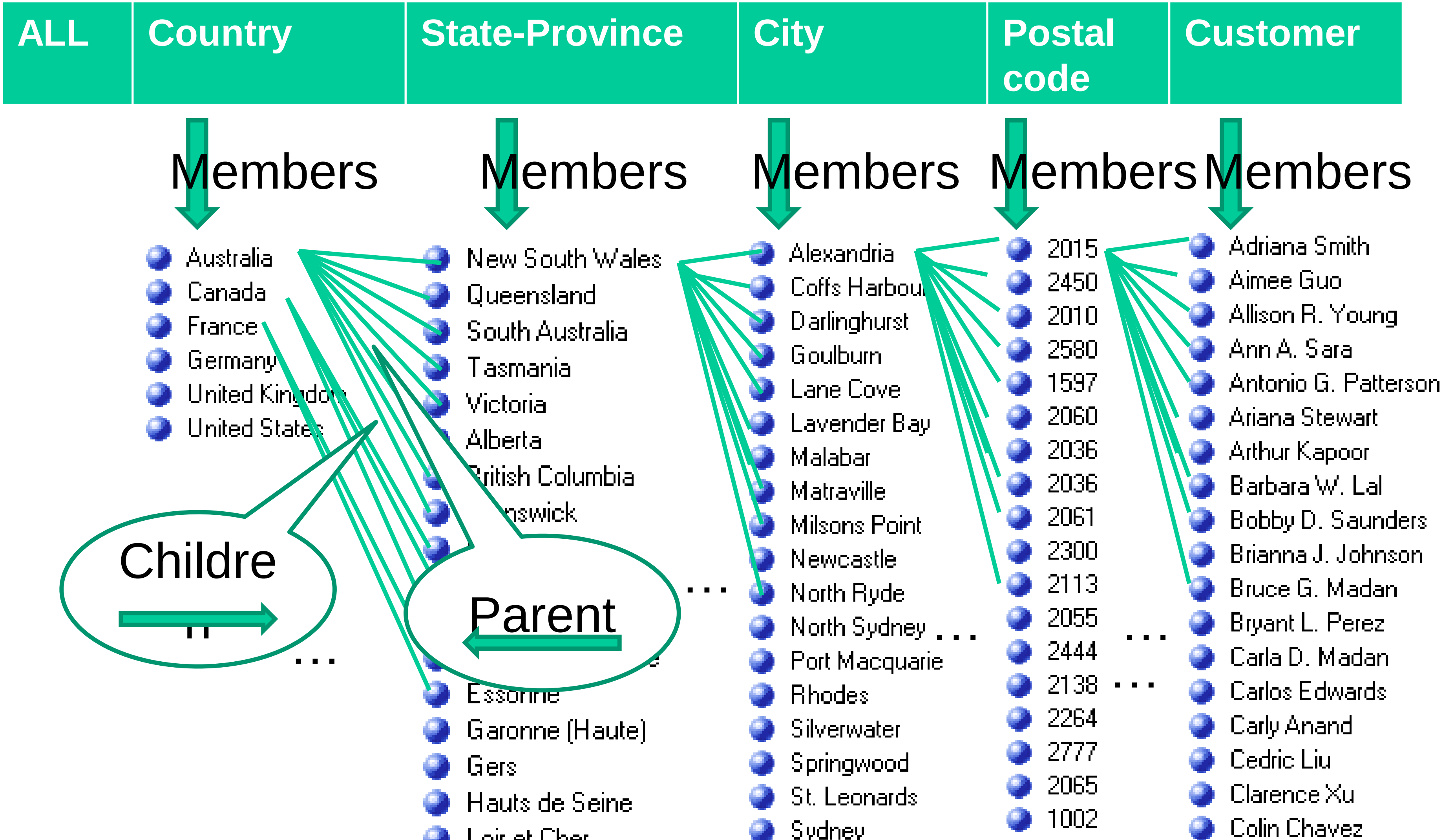
[Gender].[Gender Members]
[Product].[Product Name]
[Marital Status].[All Maritaal status]
[Promotions].[All Promotions],
[Store].[All Stores],
[Store Size in SQFT].[All],
[Store Type].[All],
[Yearly Income].[All Yearly Income]
[Time].[Year]

Fact table:

[Measures].[Unit Sales],
[Measures].[Store Cost],
[Measures].[Store Sales],
[Measures].[Sales Count],
[Measures].[Store Sales Net]

Children, Parent

Hierarchy Geography on Dimension Customer



ALL	Country	State-Province	City	Postal code	Customer

Members Members Members Members Members

Australia
Canada
France
Germany
United Kingdo
United State

New South Wales
Queensland
South Australia
Tasmania
Victoria
Alberta
ritish Columbia
nswick
Essonne
Garonne (Haute)
Gers
Hauts de Seine
Loir et Cher

Alexandria
Coffs Harbou
Darlinghurst
Goulburn
Lane Cove
Lavender Bay
Malabar
Matraville
Milsons Point
Newcastle
North Ryde
North Sydney
Port Macquarie
Rhodes
Silverwater
Springwood
St. Leonards
Sydney

2015
2450
2010
2580
1597
2060
2036
2036
2061
2300
2113
2055
2444
2138
2264
2777
2065
1002

Adriana Smith
Aimee Guo
Allison R. Young
Ann A. Sara
Antonio G. Patterson
Ariana Stewart
Arthur Kapoor
Barbara W. Lal
Bobby D. Saunders
Brianna J. Johnson
Bruce G. Madan
Bryant L. Perez
Carla D. Madan
Carlos Edwards
Carly Anand
Cedric Liu
Clarence Xu
Colin Chavez

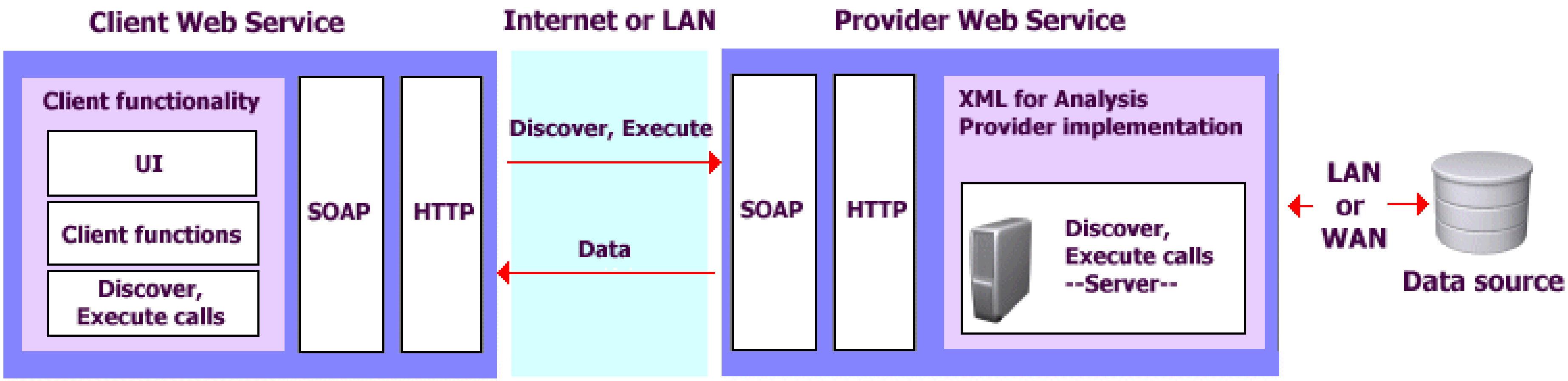
. . .

Childre
n

Parent

MDX

- Multidimensional Expressions (MDX) is a query language for cubes
  - Supported by many data warehousing systems
    - MS SQL Server, SAS OLAP Server, drivers for MDX for Oracle OLAP, …
  - Works on cubes
  - Part of XMLA (XML for Analysis) http://xmla.org



MDX query structure:

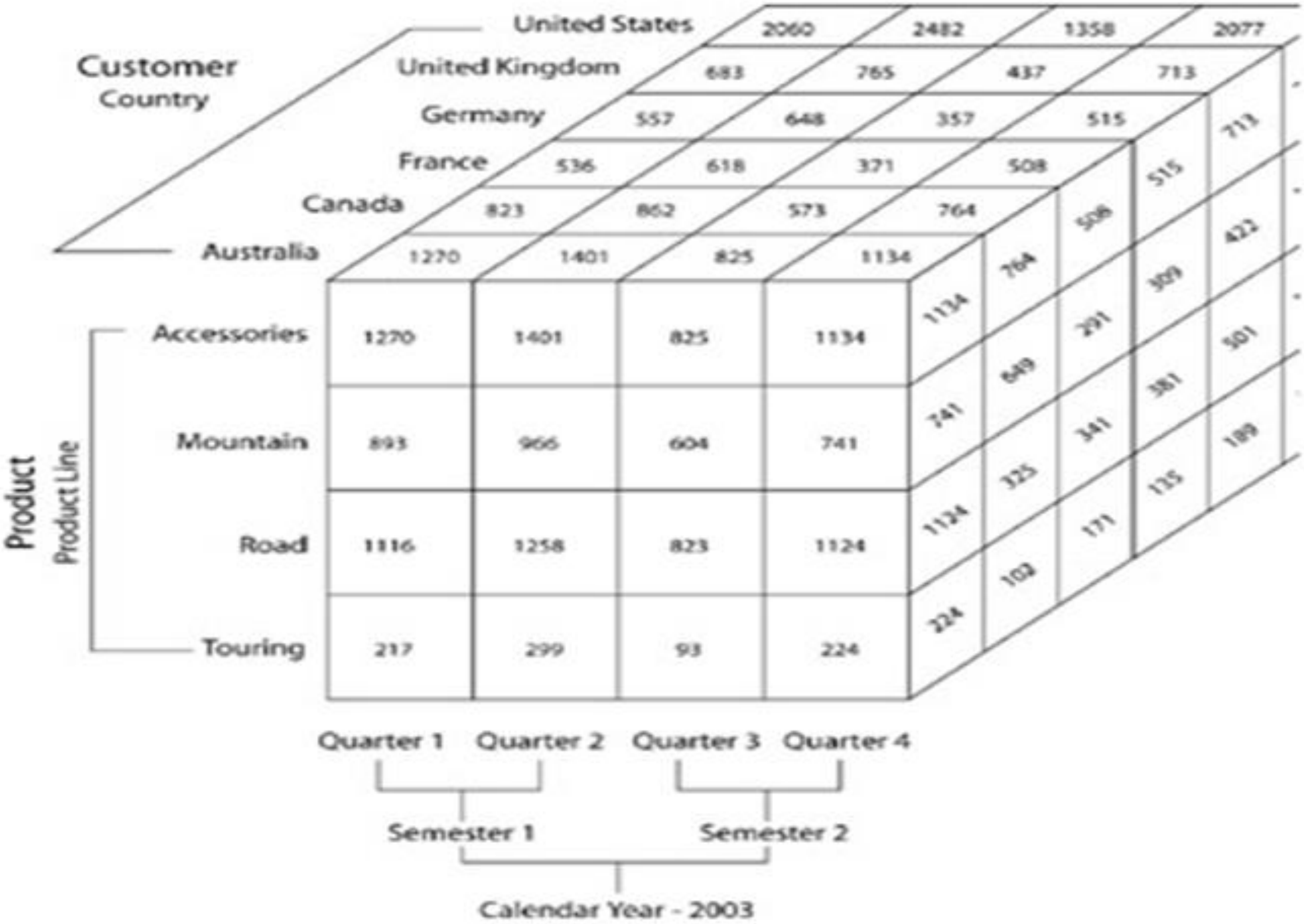
- The basic MDX query has the following structure:

SELECT $axis_{A1,......,}$ $axis_{An}$ ON COLUMNS,
$axis_{B1,......,}$ $axis_{Bn}$ ON ROWS
FROM *cube*

- Let’s compare that to the similar SQL statement:

SELECT column$_1$, column$_2$, …, column$_n$
FROM table

Cube Example



	Quarter 1	Quarter 2	Quarter 3	Quarter 4
United States	2060	2482	1358	2077
United Kingdom	683	765	437	713
Germany	557	648	357	515
France	536	618	371	508
Canada	823	862	573	764
Australia	1270	1401	825	1134

	Quarter 1	Quarter 2	Quarter 3	Quarter 4
Accessories	1270	1401	825	1134
Mountain	893	966	604	741
Road	1116	1258	823	1124
Touring	217	299	93	224

Every MDX query generates a cube !

```
SELECT Measures.[Internet Sales Amount]
                                        on COLUMNS,
   {[Customer].[Country].[France],
   [Customer].[Country].[Germany],
   [Customer].[Country].[United Kingdom]}
                                        on ROWS
FROM [Adventure Works]
```

MDX : Axis

- Axis:

Columns	Axis(0)
Rows	Axis(1)
Pages	Axis(2)
Chapters	Axis(3)
Sections	Axis(4)

SELECT Statement and Axis Specification

Country	Internet Sales Amount
France	120,000
Germany	999,999
United Kingdom	55,000

MDX : Axis Specification

- Axis specification: selection of members
  - In principle:

    [Dimension].[Hierarchy].[Level].[member]
  - Parts can be omitted if no ambiguity

    [Customer].[Customer Geography].[Coffs Harbour]

    → [Coffs Harbour]
  - Square brackets [ ] only needed when the name contains a space

MDX : Axis Specification

- Axis specification: selection of members
  - If member is missing: members of the level
  - If level and member missing: DefaultMember
  - MEMBERS: all members of the level/hierarchy
  - CHILDREN: all children of a member

MDX : Axis Specification

- Caveat! Attributes have an attribute hierarchy, including ALL and its values

Country attribute of Customer

Part of hierarchy [Customer Geography]

[Country]

[Country].[Country]

[Customer Geography].[Country]

WHERE Clause and Slicer Specification

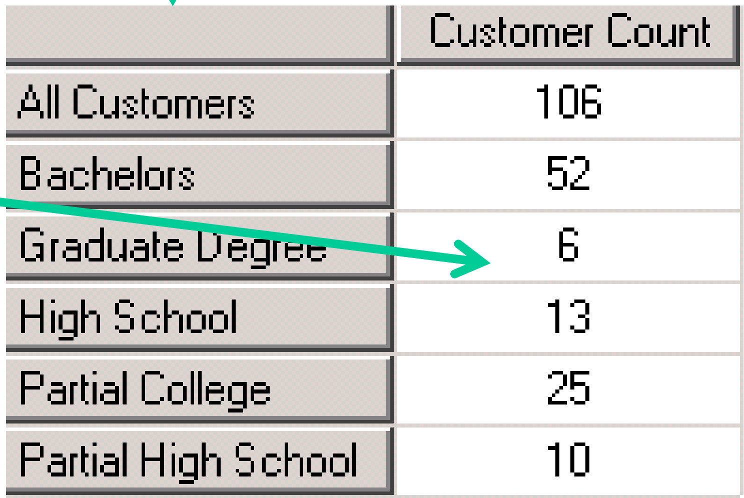
```
SELECT Measures.[Sales] on COLUMS

FROM [Adventure Works]

WHERE ([Product].[Color].[Silver]) //slicer
```

Example: MDX Specification

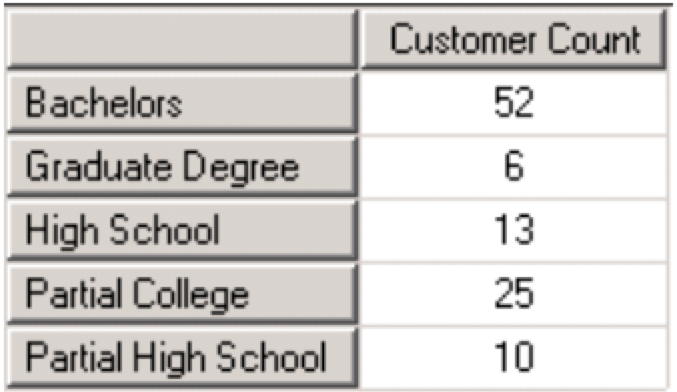
```
select
    [Measures].[Customer Count] on columns,
    [Customer].[Education].members on rows
from [Adventure Works]
where [Customer].
    [Customer Geography].
    [Coffs Harbour]
```



	Customer Count
All Customers	106
Bachelors	52
Graduate Degree	6
High School	13
Partial College	25
Partial High School	10

MDX : Axis Specification

```
select
    [Customer Count] on columns,
    [Education].[Education].members on rows
from [Adventure Works]
where [Coffs Harbour]
```



	Customer Count
Bachelors	52
Graduate Degree	6
High School	13
Partial College	25
Partial High School	10

MDX : Slicer

- If no measure on columns or rows: slicer must include measure!
- Built in the same way as the axis specification → list of members

Slicer building

Specify list of members

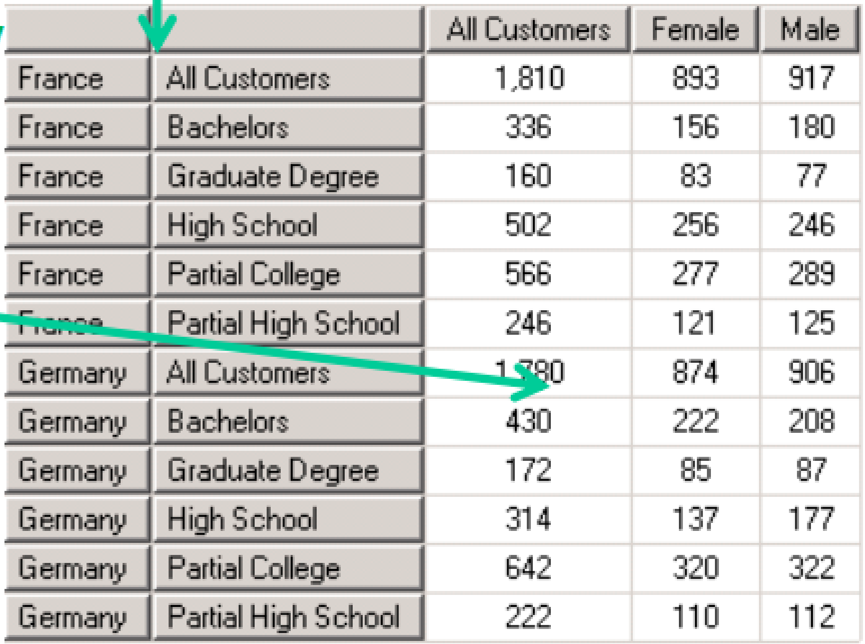
```
select
   [Customer].[Gender].members on columns,
   ( { [France], [Germany] }, education.members ) on rows
from [Adventure Works]

where (  [Customer Count],
   {[Commute Distance].[0-1 Miles],
   [Commute Distance].[1-2 Miles]}  )
```

Cross Tabulation

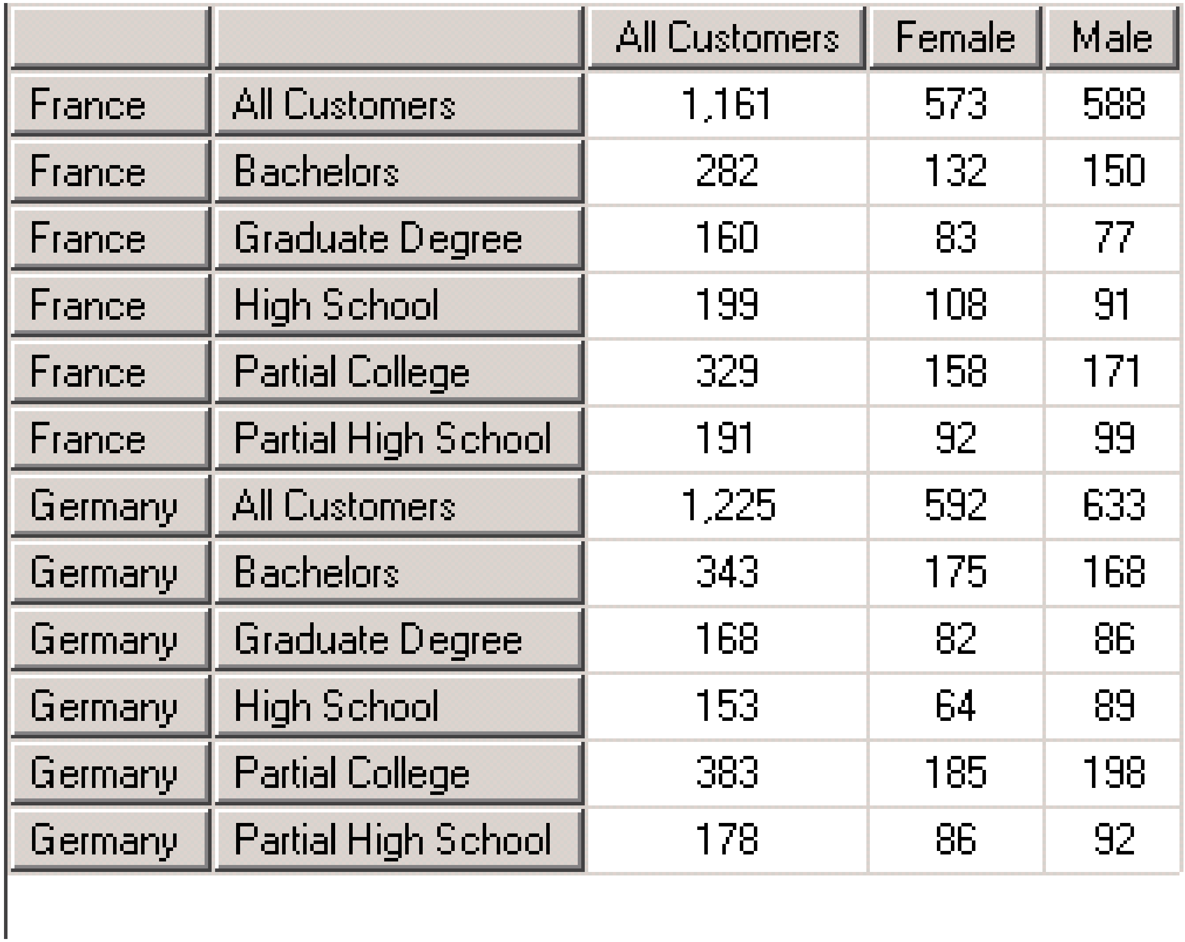
```
select
    [Customer].[Gender].members on columns,
    ( { [France], [Germany] }, education.members ) on rows
from [Adventure Works]
where [Customer Count]
```

Note: Customer count is actually
        a slicer



		All Customers	Female	Male
France	All Customers	1,810	893	917
France	Bachelors	336	156	180
France	Graduate Degree	160	83	77
France	High School	502	256	246
France	Partial College	566	277	289
France	Partial High School	246	121	125
Germany	All Customers	1,780	874	906
Germany	Bachelors	430	222	208
Germany	Graduate Degree	172	85	87
Germany	High School	314	137	177
Germany	Partial College	642	320	322
Germany	Partial High School	222	110	112

Example: Result



		All Customers	Female	Male
France	All Customers	1,161	573	588
France	Bachelors	282	132	150
France	Graduate Degree	160	83	77
France	High School	199	108	91
France	Partial College	329	158	171
France	Partial High School	191	92	99
Germany	All Customers	1,225	592	633
Germany	Bachelors	343	175	168
Germany	Graduate Degree	168	82	86
Germany	High School	153	64	89
Germany	Partial College	383	185	198
Germany	Partial High School	178	86	92

Example on Drill-down

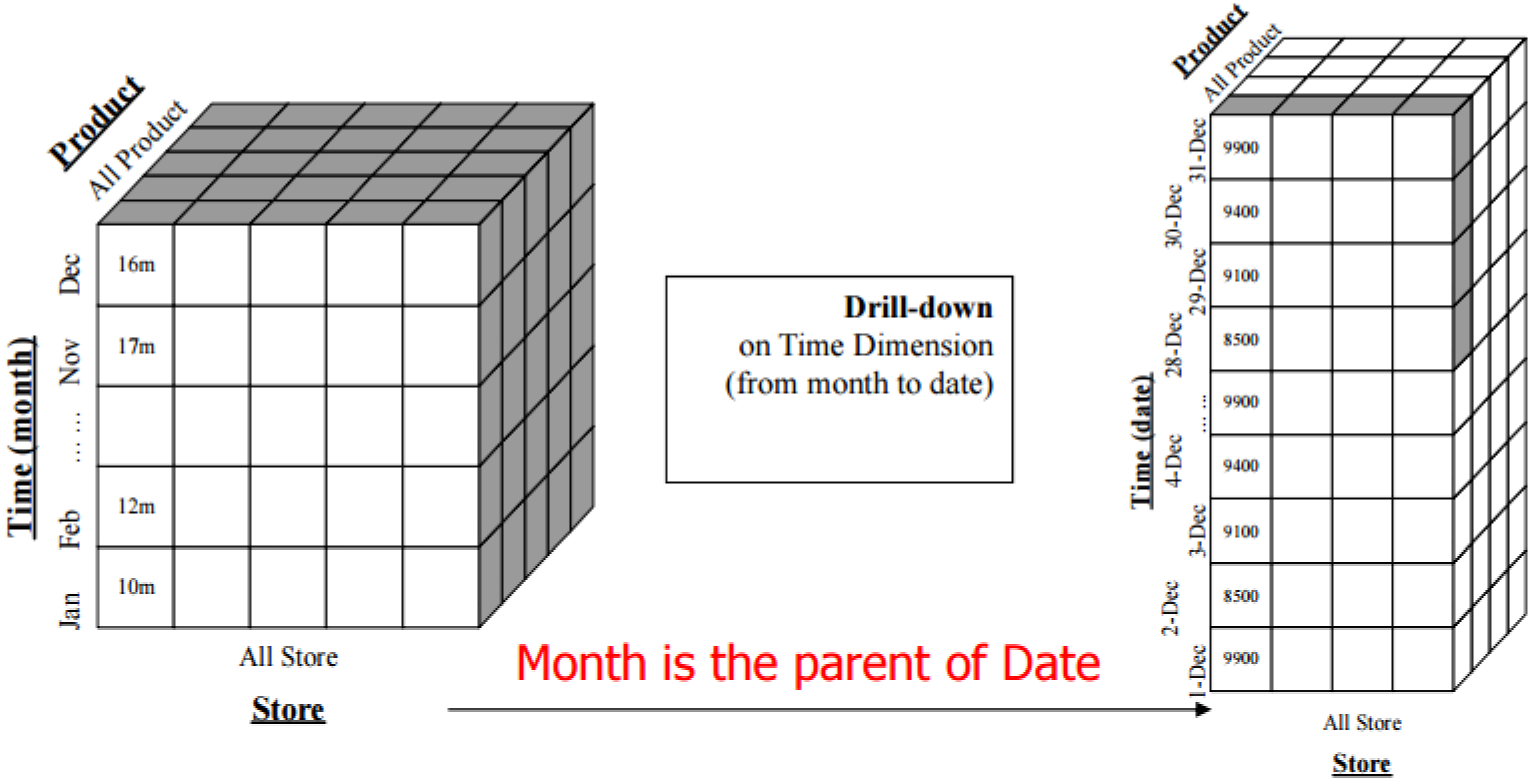
MDX

```
SELECT [SALES].[AMOUNT] ON COLUMNS,
[time].[2003].[Q4].[Dec].[31],
[time].[2003].[Q4].[Dec].[30],… …,
[time].[2003].[Q4].[Dec].[2],
[time].[2003].[Q4].[Dec].[1] ON ROWS FROM SALES
```

compare with SQL

```
select sum(amount), the_date
from SALES
where  (the_date='2003-Dec-31')
or (the_date='2003-Dec-30')
or… …or (the_date='2003-Dec-2')
or (the_date='2003-Dec-1') group by the_date
```

Graphical Description of Drill-down Example



Drill-down on Time Dimension (from month to date)

Month is the parent of Date

Displaying results from multiple axes

-
```
SELECT [Measures].[Store Sales]
                                      on COLUMNS,
   [Year].Members                     on ROWS
   [Products].[Soda].Members          on PAGES

FROM [Sales]
```

- Generates 3-D cube , very hard to display

CrossJoin

- Displaying multiple dimension members on a single axis

```
SELECT
     [Measures].[Store Sales]
                                 on COLUMNS,
      {Crossjoin({[Year].Members},
      {[Products].[Soda].Members})
                                 on ROWS
FROM [Sales]
```

		Sales
1997	Coke	100
1998	Coke	200

- Generates 2-D cube

CrossJoin -revisited

Problem with previous query: many members in the ROW axis are empty.

Need a FILTERING mechanism

```
SELECT
      {[Measures].[Store Sales]}
                                          on COLUMNS,
       NonEmpty (
       (Crossjoin({[Year].Members},
       [Products].[Soda].Members})
                                          on ROWS
FROM [Sales]
```

		Sales
1997	Coke	100
1998	Coke	200

- Generates 2-D cube

Filtering a set based on a particular condition

```
SELECT

{[Measures]. [Unit Sales]} ON COLUMNS,
{Filter({[Product]. [Product Name},
([Gender]. [All Gender].[F],[Measures].[Unit Sales]) > 10000)} ON
   ROWS

FROM [Sales]
```

Filter function

- The filter function produces a set of product departments meeting the filter criteria:

The set returned on the row axis consists of product departments for which unit sales to females are greater then $10,000