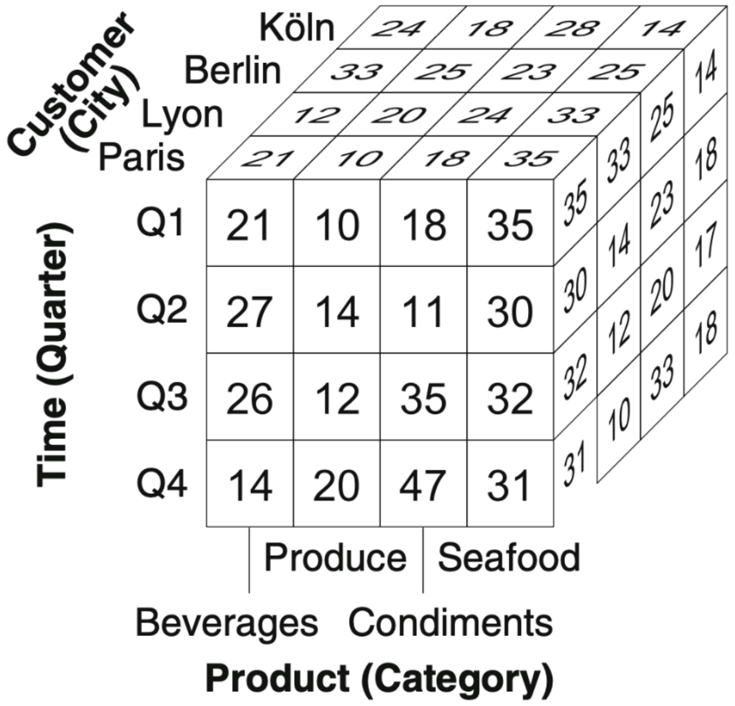
Chapter 4: Querying Data Warehouses

Data Warehouse

4.1 Introduction to MDX

4.1 Introduction to MDX

- 4.1.1 Tuples and Sets, Two fundamental concepts in MDX are tuples and sets. Intuitively, a tuple identifies a single cell in a multidimensional cube. A tuple is defined by stating one member from one or several dimensions of the cube.



4.1 Introduction to MDX

- 4.1.2 Basic Queries,
- The syntax of a typical MDX query is as follows:
  - SELECT ⟨ axis specification ⟩
  - FROM ⟨ cube ⟩
  - [WHERE⟨ slicer specification ⟩]
- As can be seen, at a first glance, MDX resembles SQL, but as we will see in this chapter, the two languages differ in several significant ways.

4.1 Introduction to MDX

- 4.1.3 Slicing, Removes a dimension from a cube by fixing a single value in a level of the dimension.
- To restrict the result to Belgium, we can write the following query:
  - SELECT Measures.MEMBERS ON COLUMNS, [Order Date].Year.MEMBERS ON ROWS
  - FROM Sales
  - WHERE (Customer.Country.Belgium)

- 4.1.4 Navigation
  - SELECT [Order Date].Year.MEMBERS ON COLUMNS,

    {Customer.Country.France,Customer.Country.Italy} ON ROWS
  - FROM Sales
  - WHERE Measures.[Sales Amount]

  - SELECT [Order Date].Year.MEMBERS ON COLUMNS,

    NON EMPTY { Customer.France.CHILDREN, Customer.Italy.CHILDREN } ON ROWS
  - FROM Sales
  - WHERE Measures.[Sales Amount]

4.1 Introduction to MDX

- 4.1.5 Cross Join
  - SELECT Product.Category.MEMBERS ON COLUMNS,
    CROSSJOIN(Customer.Country.MEMBERS,
    [Order Date].Calendar.Quarter.MEMBERS) ON ROWS
  - FROM Sales
  - WHERE Measures.[Sales Amount]

  - SELECT Product.Category.MEMBERS ON COLUMNS,
    Customer.Country.MEMBERS *
    [Order Date].Calendar.Quarter.MEMBERS ON ROWS
  - FROM Sales
  - WHERE Measures.[Sales Amount]

4.1 Introduction to MDX

- 4.1.6 Subqueries
  - SELECT Measures.[Sales Amount] ON COLUMNS,
    [Order Date].Calendar.Quarter.MEMBERS ON ROWS
  - FROM (SELECT { Product.Category.Beverages,
    Product.Category.Condiments } ON COLUMNS
    FROM Sales)

  - SELECT Measures.[Sales Amount] ON COLUMNS,
    [Order Date].Calendar.Quarter.MEMBERS *
    Product.Category.MEMBERS ON ROWS
  - FROM (SELECT { Product.Category.Beverages,
    Product.Category.Condiments } ON COLUMNS
    FROM Sales)

4.1 Introduction to MDX

- 4.1.7 Calculated Members and Named Sets
  - WITH MEMBER Measures.Profit% AS
    (Measures.[Sales Amount] - Measures.[Freight]) / (Measures.[Sales Amount]),
    FORMAT STRING = '#0.00%'
  - SELECT { [Sales Amount], Freight, Profit% } ON COLUMNS,
    Customer.Country ON ROWS
  - FROM Sales

  - WITH MEMBER Product.Categories.[All].[Meat & Fish] AS
    Product.Categories.[Meat/Poultry] + Product.Categories.[Seafood]
  - SELECT { Measures.[Unit Price], Measures.Quantity, Measures.Discount,
    Measures.[Sales Amount] } ON COLUMNS,
    Category.ALLMEMBERS ON ROWS
  - FROM Sales

- 4.1.8 Relative Navigation
  - WITH MEMBER Measures.[Percentage Sales] AS

    (Measures.[Sales Amount], Customer.Geography.CURRENTMEMBER) /

    (Measures.[Sales Amount], Customer.Geography.CURRENTMEMBER.PARENT),

    FORMAT STRING = '#0.00%’
  - SELECT {Measures.[Sales Amount], Measures.[Percentage Sales] }

    ON COLUMNS, DESCENDANTS(Customer.Europe,

    Customer.Country, SELF AND BEFORE) ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.8 Relative Navigation
  - SELECT Product.Category.MEMBERS ON COLUMNS,

    GENERATE({Customer.Belgium, Customer.France},

    DESCENDANTS(Customer.Geography.CURRENTMEMBER,

    [Company Name])) ON ROWS
  - FROM Sales
  - WHERE Measures.[Sales Amount]

4.1 Introduction to MDX

- 4.1.9 Time Series Functions
  - WITH MEMBER Measures.[Previous Year] AS

    (Measures.[Net Sales],

    PARALLELPERIOD([Order Date].Calendar.Quarter, 4)),

    FORMAT STRING = '$###,##0.00'
  - MEMBER Measures.[Net Sales Growth] AS

    Measures.[Net Sales] - Measures.[Previous Year],

    FORMAT STRING = '$###,##0.00; $-###,##0.00'
  - SELECT { [Net Sales], [Previous Year], [Net Sales Growth] } ON COLUMNS,

    [Order Date].Calendar.Quarter ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.9 Time Series Functions
  - WITH MEMBER Measures.[Quantity Difference] AS

    (Measures.[Quantity]) - (Measures.[Quantity],

    OPENINGPERIOD([Order Date].Calendar.Month,

    [Order Date].Calendar.CURRENTMEMBER.PARENT))
  - SELECT { Measures.[Quantity], Measures.[Quantity Difference] } ON COLUMNS,

    [Order Date].Calendar.[Month] ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.10 Filtering
    - SELECT Product.Category.MEMBERS ON COLUMNS,

      FILTER(Customer.City.MEMBERS, (Measures.[Sales Amount],

      [Order Date].Calendar.[1997]) > 25000) ON ROWS
    - FROM Sales
    - WHERE (Measures.[Net Sales Growth], [Order Date].Calendar.[1997])

4.1 Introduction to MDX

- 4.1.10 Filtering
  - WITH MEMBER Measures.[Profit%] AS
    (Measures.[Sales Amount] - Measures.[Freight]) /
    (Measures.[Sales Amount]), FORMAT STRING = '#0.00%'
  - MEMBER Measures.[Profit%City] AS
    (Measures.[Profit%],
    Customer.Geography.CURRENTMEMBER.PARENT),
    FORMAT STRING = '#0.00%'
  - SELECT { Measures.[Sales Amount], Measures.[Freight], Measures.[Net Sales],
    Measures.[Profit%], Measures.[Profit%City] } ON COLUMNS,
    FILTER(NONEMPTY(Customer.Customer.MEMBERS),
    (Measures.[Profit%]) < (Measures.[Profit%City])) ON ROWS
  - FROM Sales
  - WHERE [Order Date].Calendar.[1997]

4.1 Introduction to MDX

- 4.1.11 Sorting
  - SELECT Measures.[Sales Amount] ON COLUMNS,

    NON EMPTY GENERATE(

    ORDER(Customer.Geography.Continent.ALLMEMBERS,

    Customer.Geography.CURRENTMEMBER.NAME, BASC),

    ORDER({ Customer.Geography.CURRENTMEMBER } *

    Product.Categories.Category.ALLMEMBERS,

    Product.Categories.CURRENTMEMBER.NAME, BASC)) ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.12 Top and Bottom Analysis
  - SELECT Measures.MEMBERS ON COLUMNS,

    HEAD(ORDER(Customer.Geography.City.MEMBERS,

    Measures.[Sales Amount], BDESC), 3) ON ROWS
  - FROM Sales

  - SELECT Measures.MEMBERS ON COLUMNS,

    TOPCOUNT(Customer.Geography.City.MEMBERS, 3,

    Measures.[Sales Amount]) ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.12 Top and Bottom Analysis
  - WITH SET SetTop3Cities AS TOPCOUNT(

    Customer.Geography.City.MEMBERS, 3, [Sales Amount])
  - MEMBER Customer.Geography.[Top 3 Cities] AS

    AGGREGATE(SetTop3Cities)
  - MEMBER Customer.Geography.[Other Cities] AS (Customer.[All]) –

    (Customer.[Top 3 Cities])
  - SELECT Measures.MEMBERS ON COLUMNS,

    { SetTop3Cities, [Top 3 Cities], [Other Cities], Customer.[All] } ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.13 Aggregation Functions
  - WITH MEMBER Measures.[Maximum Sales] AS
    MAX(DESCENDANTS([Order Date].Calendar.Year.[1997],
    [Order Date].Calendar.Month), Measures.[Sales Amount])
  - MEMBER Measures.[Minimum Sales] AS
    MIN(DESCENDANTS([Order Date].Calendar.Year.[1997],
    [Order Date].Calendar.Month), Measures.[Sales Amount])
  - MEMBER Measures.[Average Sales] AS
    AVG(DESCENDANTS([Order Date].Calendar.Year.[1997],
    [Order Date].Calendar.Month), Measures.[Sales Amount])
  - SELECT { [Sales Amount], [Maximum Sales],
    [Minimum Sales], [Average Sales] } ON COLUMNS,
    Product.Categories.Category.MEMBERS ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.13 Aggregation Functions
  - WITH MEMBER Measures.[Maximum Sales] AS

    MAX(DESCENDANTS([Order Date].Calendar.Year.[1997],

    [Order Date].Calendar.Month), Measures.[Sales Amount])
  - MEMBER Measures.[Maximum Period] AS

    TOPCOUNT(DESCENDANTS([Order Date].Calendar.Year.[1997],

    [Order Date].Calendar.Month), 1,

    Measures.[Sales Amount]).ITEM(0).NAME
  - SELECT { [Maximum Sales], [Maximum Period] } ON COLUMNS,

    Product.Categories.Category.MEMBERS ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.13 Aggregation Functions
  - WITH MEMBER Measures.[Maximum Sales] AS
    MAX(DESCENDANTS([Order Date].Calendar.Year.[1997],
    [Order Date].Calendar.[Month]), Measures.[Sales Amount])
  - MEMBER Measures.[Maximum Period] AS
    TOPCOUNT(DESCENDANTS([Order Date].Calendar.Year.[1997],
    [Order Date].Calendar.[Month]), 1,
    Measures.[Sales Amount]).ITEM(0).NAME
  - SELECT { [Maximum Sales], [Maximum Period] } ON COLUMNS,
    Product.Categories.Category.MEMBERS *
    Customer.Geography.Country.MEMBERS ON ROWS
  - FROM Sales

4.1 Introduction to MDX

- 4.1.13 Aggregation Functions
  - WITH MEMBER Measures.[Customer Count] AS

    COUNT({Measures.[Sales Amount] *

    [Customer].[Company Name].MEMBERS}, EXCLUDEEMPTY)
  - SELECT { Measures.[Sales Amount], Measures.[Customer Count] } ON COLUMNS,

    Product.Category.MEMBERS ON ROWS
  - FROM Sales

4.2 Querying the Northwind Cube in MDX

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - SELECT [Order Date].Year.CHILDREN ON COLUMNS,

    NON EMPTY Customer.[Company Name].CHILDREN *

    Product.[Category Name].CHILDREN ON ROWS
  - FROM Sales
  - WHERE Measures.[Sales Amount]
  - SELECT [Order Date].Year.MEMBERS ON COLUMNS,

    NON EMPTY Customer.Country.MEMBERS *

    Supplier.Country.MEMBERS ON ROWS
  - FROM Sales
  - WHERE Measures.[Sales Amount]

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.[Previous Year] AS

    (Measures.[Sales Amount],

    PARALLELPERIOD([Order Date].Calendar.Month,12)),

    FORMAT STRING = '$###,##0.00'
  - SELECT { Measures.[Sales Amount], Measures.[Previous Year] } ON COLUMNS,

    NON EMPTY ORDER(Customer.Geography.State.MEMBERS,

    Customer.Geography.CURRENTMEMBER.NAME, BASC) *

    [Order Date].Calendar.Month.MEMBERS ON ROWS
  - FROM Sales

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.[Previous Month] AS
    (Measures.[Sales Amount],
    [Order Date].Calendar.CURRENTMEMBER.PREVMEMBER),
    FORMAT STRING = '$###,##0.00’
  - MEMBER Measures.[Sales Growth] AS
    (Measures.[Sales Amount]) - (Measures.[Previous Month]),
    FORMAT STRING = '$###,##0.00; $-###,##0.00’
  - SELECT { Measures.[Sales Amount], Measures.[Previous Month],
    Measures.[Sales Growth] } ON COLUMNS,
    NON EMPTY ORDER(Product.Categories.Product.MEMBERS,
    Product.Categories.CURRENTMEMBER.NAME, BASC) *
    [Order Date].Calendar.Month.MEMBERS ON ROWS
  - FROM Sales

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - SELECT Measures.[Sales Amount] ON COLUMNS,
    TOPCOUNT(Employee.[Full Name].CHILDREN, 3,
    Measures.[Sales Amount]) ON ROWS
  - FROM Sales

  - SELECT Measures.[Sales Amount] ON COLUMNS,
    { Customer.Geography.[All],
    TOPPERCENT([Customer].Geography.Country.MEMBERS, 50,
    Measures.[Sales Amount]) } ON ROWS
  - FROM Sales

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.[Avg Monthly Sales] AS

    AVG(DESCENDANTS([Order Date].Calendar.CURRENTMEMBER,

    [Order Date].Calendar.Month),Measures.[Sales Amount]),

    FORMAT STRING = '$###,##0.00'
  - SELECT { Measures.[Sales Amount], Measures.[Avg Monthly Sales] } ON COLUMNS,

    Employee.[Full Name].CHILDREN *

    [Order Date].Calendar.Year.MEMBERS ON ROWS
  - FROM Sales

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.[TotalDisc] AS

    Measures.Discount * Measures.Quantity *

    Measures.[Unit Price], FORMAT STRING = '$###,##0.00'
  - SELECT { Measures.[Sales Amount], [TotalDisc] } ON COLUMNS,

    NON EMPTY ORDER(Product.Categories.Product.MEMBERS,

    Product.Categories.CURRENTMEMBER.NAME, BASC) *

    [Order Date].Calendar.Month.MEMBERS ON ROWS
  - FROM Sales

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.YTDSales AS

    SUM(PERIODSTODATE([Order Date].Calendar.[Year],

    [Order Date].Calendar.CURRENTMEMBER),

    Measures.[Sales Amount]), FORMAT STRING = '###,##0.00'
  - SELECT DESCENDANTS([Order Date].[1996], [Order Date].[Month])

    ON COLUMNS, Product.[Category].MEMBERS ON ROWS
  - FROM Sales
  - WHERE (Measures.YTDSales)

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.MovAvg3Months AS

    AVG([Order Date].Calendar.CURRENTMEMBER.LAG(2):

    [Order Date].Calendar.CURRENTMEMBER,

    Measures.[Sales Amount]), FORMAT STRING = '$###,##0.00'
  - SELECT [Order Date].Calendar.Month.MEMBERS ON COLUMNS,

    Product.[Category].MEMBERS ON ROWS
  - FROM Sales
  - WHERE (Measures.MovAvg3Months)

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.[Personal Sales] AS

    (Employee.Supervision.DATAMEMBER, [Measures].[Sales Amount]),

    FORMAT STRING = '$###,##0.00'
  - SELECT { Measures.[Personal Sales], Measures.[Sales Amount] } ON COLUMNS,

    ORDER(Employee.Supervision.MEMBERS - Employee.Supervision.[All],

    Employee.Supervision.CURRENTMEMBER.NAME, BASC) ON ROWS
  - FROM Sales
  - WHERE [Order Date].Calendar.Year.[1997]

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - SELECT Measures.[Sales Amount] on COLUMNS,

    Employee.[Full Name].CHILDREN ON ROWS
  - FROM Sales
  - WHERE [Order Date].Calendar.Year.[1997]
  - WITH MEMBER Measures.[NbProducts] AS

    COUNT(NONEMPTY([Order].[Order No].CURRENTMEMBER *

    [Order].[Order Line].MEMBERS))
  - SELECT { Measures.[Sales Amount], NbProducts, Quantity } on COLUMNS,

    [Order].[Order No].CHILDREN ON ROWS
  - FROM Sales

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER Measures.AvgSales AS

    Measures.[Sales Amount]/Measures.[Order No],

    FORMAT STRING = '$###,##0.00'
  - SELECT { Measures.[Order No], [Sales Amount], AvgSales } ON COLUMNS,

    NON EMPTY [Order Date].Calendar.Month.MEMBERS ON ROWS
  - FROM Sales

4.2 Querying the Northwind Cube in MDX

- 4.2 Querying the Northwind Cube in MDX
  - WITH MEMBER NoCities AS

    Measures.[Territories Count]
  - MEMBER NoStates AS

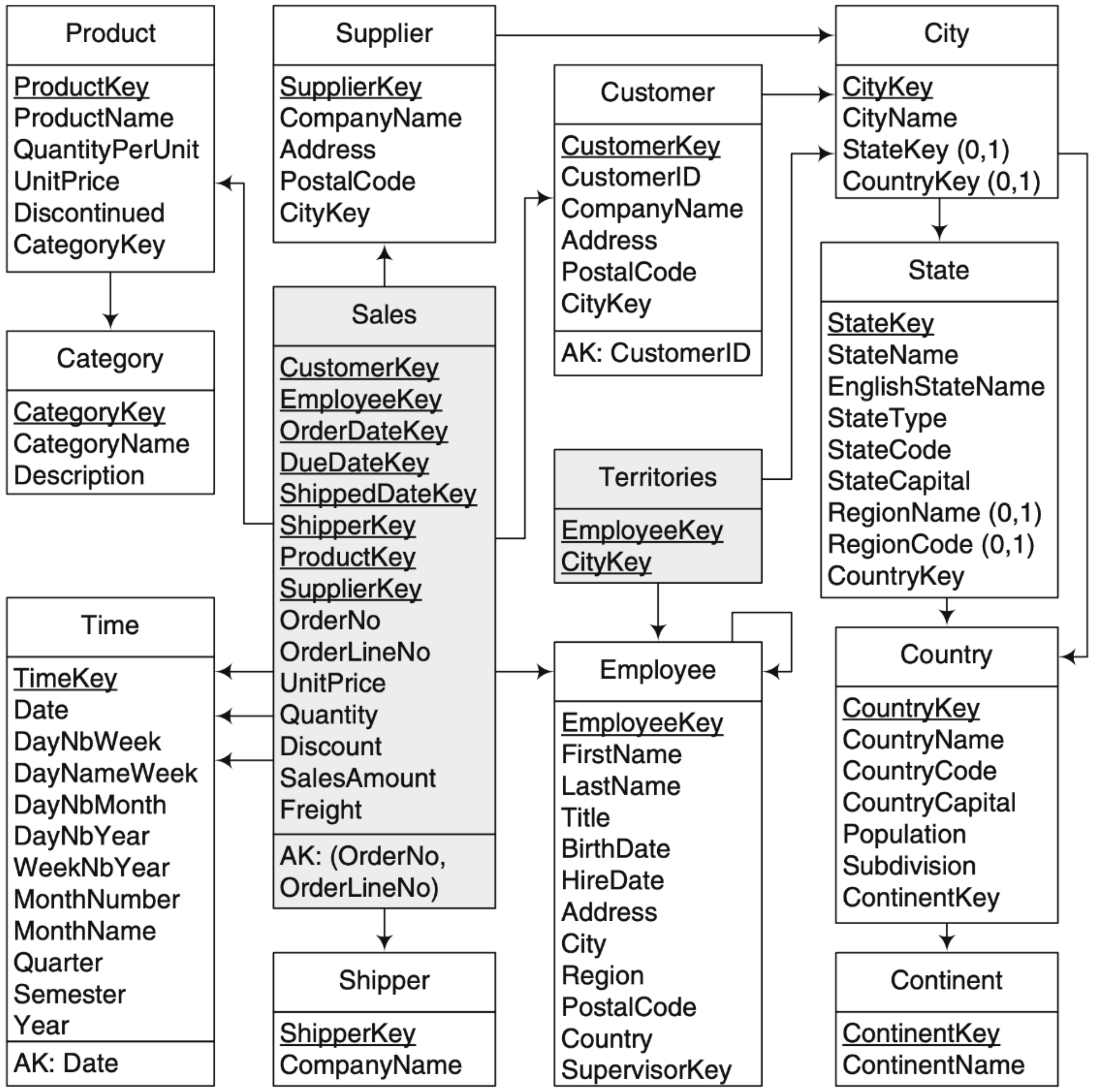
    DISTINCTCOUNT(Employee.[Full Name].CURRENTMEMBER *

    City.Geography.State.MEMBERS)
  - SELECT { Measures.[Sales Amount], Measures.NoCities, Measures.NoStates }

    ON COLUMNS, Employee.[Full Name].CHILDREN ON ROWS
  - FROM Sales

4.3 Querying the Northwind Data Warehouse in SQL

4.3 Querying the Northwind Data Warehouse in SQL



4.3 Querying the Northwind Data Warehouse in SQL

- 4.3.1 Total sales amount per customer, year, and product category
  - SELECT C.CompanyName, T.Year, A.CategoryName,

    FORMAT(SUM(SalesAmount),'$###,##0.00') AS SalesAmount
  - FROM Sales S, Customer C, Time T, Product P, Category A
  - WHERE S.CustomerKey = C.CustomerKey AND

    S.OrderDateKey = T.TimeKey AND

    S.ProductKey = P.ProductKey AND P.CategoryKey = A.CategoryKey
  - GROUP BY C.CompanyName, T.Year, A.CategoryName

4.3 Querying the Northwind Data Warehouse in SQL

- 4.3.2 Yearly sales amount for each pair of customer country and supplier countries
  - SELECT CO.CountryName AS CustomerCountry,
    SO.CountryName AS SupplierCountry, T.Year,
    FORMAT(SUM(SalesAmount),'$###,##0.00') AS SalesAmount
  - FROM Sales F, Customer C, City CC, State CS, Country CO,
    Supplier S, City SC, State SS, Country SO, Time T
  - WHERE F.CustomerKey = C.CustomerKey AND C.CityKey = CC.CityKey AND
    CC.StateKey = CS.StateKey AND CS.CountryKey = CO.CountryKey AND
    F.SupplierKey = S.SupplierKey AND S.CityKey = SC.CityKey AND SC.StateKey = SS.StateKey AND
    SS.CountryKey = SO.CountryKey AND F.OrderDateKey = T.TimeKey
  - GROUP BY CO.CountryName, SO.CountryName, T.Year
  - ORDER BY CO.CountryName, SO.CountryName, T.Year

4.3 Querying the Northwind Data Warehouse in SQL

- 4.3.3 Three best-selling employees
  - SELECT TOP(3) E.FirstName + ' ' + E.LastName AS EmployeeName,

    FORMAT(SUM(F.SalesAmount), '$###,##0.00') AS SalesAmount
  - FROM Sales F, Employee E
  - WHERE F.EmployeeKey = E.EmployeeKey
  - GROUP BY E.FirstName, E.LastName
  - ORDER BY SUM(F.SalesAmount) DESC

4.3 Querying the Northwind Data Warehouse in SQL

- 4.3.4 Best-selling employee per product and year
  - WITH SalesProdYearEmp AS (
    SELECT P.ProductName, T.Year, SUM(S.SalesAmount) AS SalesAmount,
    E.FirstName + ' ' + E.LastName AS EmployeeName
    FROM Sales S, Employee E, Time T, Product P
    WHERE S.EmployeeKey = E.EmployeeKey AND
    S.OrderDateKey = T.TimeKey AND S.ProductKey = P.ProductKey
    GROUP BY P.ProductName, T.Year, E.FirstName, E.LastName)
  - SELECT ProductName, Year,
    FORMAT(SalesAmount,'$###,##0.00') AS TopSales, EmployeeName AS TopEmployee
  - FROM SalesProdYearEmp S1
  - WHERE S1.SalesAmount = (
    SELECT MAX(SalesAmount) FROM SalesProdYearEmp S2
    WHERE S1.ProductName = S2.ProductName AND S1.Year = S2.Year)

4.3 Querying the Northwind Data Warehouse in SQL

- 4.3.5 For each month, total number of orders, total sales amount, and average sales amount by order
  - WITH OrderAgg AS (

    SELECT OrderNo, OrderDateKey, SUM(SalesAmount) AS SalesAmount Sales

    FROM Sales GROUP BY OrderNo, OrderDateKey)
  - SELECT MonthYear(MonthNumber,Year) AS Month, COUNT(OrderNo) AS NoOrders,

    FORMAT(SUM(SalesAmount), '$###,##0.00') AS SalesAmount,

    FORMAT(AVG(SalesAmount), '$###,##0.00') AS AvgAmount
  - FROM OrderAgg O, Time T
  - WHERE O.OrderDateKey = T.TimeKey
  - GROUP BY Year, MonthNumber
  - ORDER BY Year, MonthNumber

4.3 Querying the Northwind Data Warehouse in SQL

- 4.3.6 For each employee, total sales amount, number of cities, and number of states to which she is assigned
  - SELECT FirstName + ' ' + LastName AS FullName,

    FORMAT(SUM(SalesAmount) / COUNT(DISTINCT CityName), '$###,##0.00') AS TotalSales,

    COUNT(DISTINCT CityName) AS NoCities,

    COUNT(DISTINCT StateName) AS NoStates
  - FROM Sales F, Employee E, Territories T, City C, State S
  - WHERE F.EmployeeKey = E.EmployeeKey AND

    E.EmployeeKey = T.EmployeeKey AND T.CityKey = C.CityKey AND C.StateKey = S.StateKey
  - GROUP BY FirstName + ' ' + LastName
  - ORDER BY FirstName + ' ' + LastName

4.3 Querying the Northwind Data Warehouse in SQL

- 4.3.7 Countries that account for top 50% of the sales amount.
- 4.3.8 Total sales and average monthly sales by employee and year.
- 4.3.9 Total sales amount and total discount amount per product and month.
- 4.3.10 Monthly year-to-date sales for each product category.
- 4.3.11 Moving average over the last 3 months of the sales amount by product category.
- 4.3.12 Personal sales amount made by an employee compared with the total sales amount made by herself and her subordinates during 1997.
- 4.3.13 Total sales amount, number of products, and sum of the quantities sold for each order.
- 4.3.14 For each month, total number of orders, total sales amount, and average sales amount by order.
- 4.3.15 For each employee, total sales amount, number of cities, and number of states to which she is assigned.

4.4 Comparison of MDX and SQL

4.4 Comparison of MDX and SQL

MDX	SQL
Advantages • Data modeling: definition of dimensions, hierarchies, measure groups, from various data sources • Simple navigation within time dimension and hierarchies • Relatively simple expressions for often used business requests • Fast, due to the existence of aggregations	**Advantages** • Large user base • Easy-to-understand semantics of queries • Results are easy to visualize: scalars or 2D tables • Various ways of relating tables: joins, derived tables, correlated queries, common table expressions, etc.
Disadvantages • Extra effort for designing a cube and setting up aggregations • Steep learning curve: manipulating an *n*-dimensional space • Hard-to-grasp concepts: current context, execution phases, etc. • Some operations are difficult to express, such as ordering on multiple criteria	**Disadvantages** • Tables must be joined explicitly inside a query • Sometimes not intuitive and complex syntax for expressing analytical queries • No concept of row ordering and hierarchies: navigation dimensions may be complex • Not so performant for the types of queries used in data analysis

Enjoy the Course…!