Data Warehousing Introduction

Esteban Zimanyi
ezimanyi@ulb.ac.be
Slides from Toon Calders

ECOLE
POLYTECHNIQUE
DE BRUXELLES

Course Organization

- Lectures on Tuesday 14:00 and Thursday 16:00
  - Check http://gehol.ulb.ac.be/ for room
- Most exercises in computer class
  - Tutorial MS SQL Server tools
    - MS Sequel Server, SSIS, SSAS, SSRS
- Contributions from associated partners
  - IBM (TBC)
  - Teradata (TBC)

Course Organization

- Grading:
  - Written exam (14/20)
  - Project (6/20)
    - 2 practical assignments in groups of 3-4
      - TPC-DS benchmark
      - TPC-DI benchmark

Motivation for the Course

- Database = a piece of software to handle data:
  - Store, maintain, and query
- Most ideal system situation-dependent
  - data type: simple / semi-structured / complex / …
  - types of queries: simple lookup / analytical / …
  - type of usage: multi-user / single-user / distributed / …

Online Transaction Processing (OLTP)

- Relational database management systems are mainly to support transaction processing
  - Concurrent access
  - Data consistency, non-redundancy
  - Ad-hoc Querying
  - Efficiency

Atomicity

- Consider a Bank transaction; John transfers 100 euro to Mary
    1. Check if Balance John > 100 euro?
    2. Balance John -100 euro
    3. Balance Mary +100 euro
- What can go wrong when the banking system crashes?

Atomicity

- Consider a Bank transaction; John transfers 100 euro to Mary

  1. Check if Balance John > 100 euro?
  2. Balance John -100 euro

  CRASH

  3. Balance Mary +100 euro

- What can go wrong when the banking system crashes?
  - When the system is restarted, John has 100 euro less, but Mary did not receive it!

Consistency

- Consider a Bank transaction; John transfers 100 euro to Mary
    1. Balance John -100 euro
    2. Balance Mary +100 euro
- Suppose consistency rule:
  Balance should always ≥ 0
    - After the transaction, the database should still be consistent
    - Otherwise: roll-back

Durability

- Consider a Bank transaction; John transfers 100 euro to Mary

  1. Check if Balance John > 100 euro?
  2. Balance John -100 euro
  3. Balance Mary +100 euro

  COMMIT

CRASH

- After commit, transaction result should persist

Isolation

- Consider a Bank transaction; John withdraws 100 euro from an ATM; his wife Mary pays 50 Euro in a shop at the same time, from the same account.

John	Mary
Get balance	Get balance
Subtract 100 euro	Subtract 50 euro
Store new balance	Store new balance

- Possible problems?

Isolation

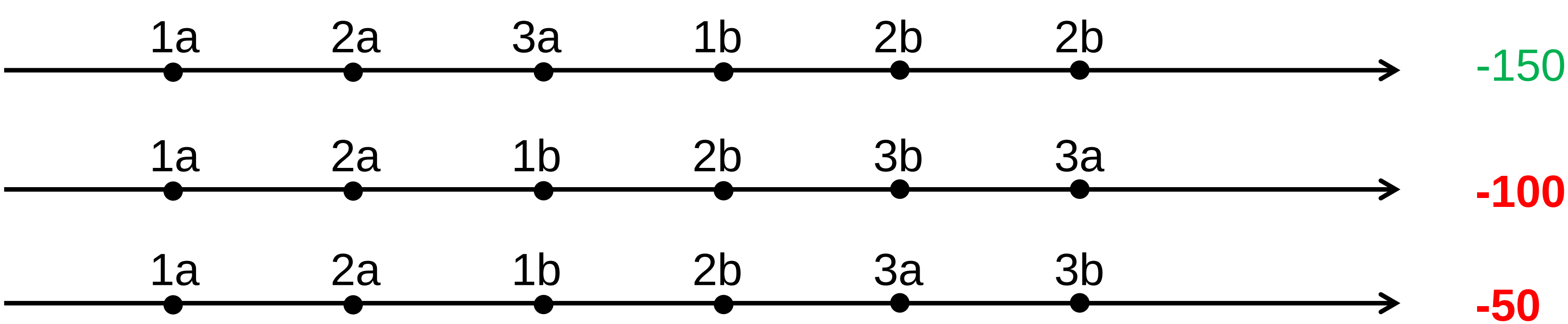
- Consider a Bank transaction; John withdraws 100 euro from an ATM; his wife Mary pays 50 Euro in a shop at the same time, from the same account.

John	Mary
1a. Get balance	1b. Get balance
2a. Subtract 100 euro	2b. Subtract 50 euro
3a. Store new balance	3b. Store new balance

Isolation

- Consider a Bank transaction; John withdraws 100 euro from an ATM; his wife Mary pays 50 Euro in a shop at the same time, from the same account.

John	Mary
1a. Get balance	1b. Get balance
2a. Subtract 100 euro	2b. Subtract 50 euro
3a. Store new balance	3b. Store new balance



Concurrent Access

- Multiple users
  - Concurrent access
  - Frequent inserts, deletes, updates

  → need for ACID
- Extremely important to have most recent information
- Enforced by “protocols” based on *locking*

Online Transaction Processing (OLTP)

- Relational database management systems are mainly to support transaction processing
  - Concurrent access
  - Data consistency, non-redundancy
  - Ad-hoc Querying
  - Efficiency

Design Theory

- Which instance do you prefer? Why?

Student	Code	Name	Semester	Lecturer	Grade
Phil	2ID45	Advanced Databases	Spring 2011	Calders	A+
Mary	2ID45	Advanced Databases	Spring 2011	Calders	C
John	2ID45	Advanced Databases	Spring 2011	Calders	B-
Paul	2ID05	Databases I	Spring 2011	Fletcher	C

Courses

Code	Name
2ID45	Advanced Databases
2ID05	Databases I

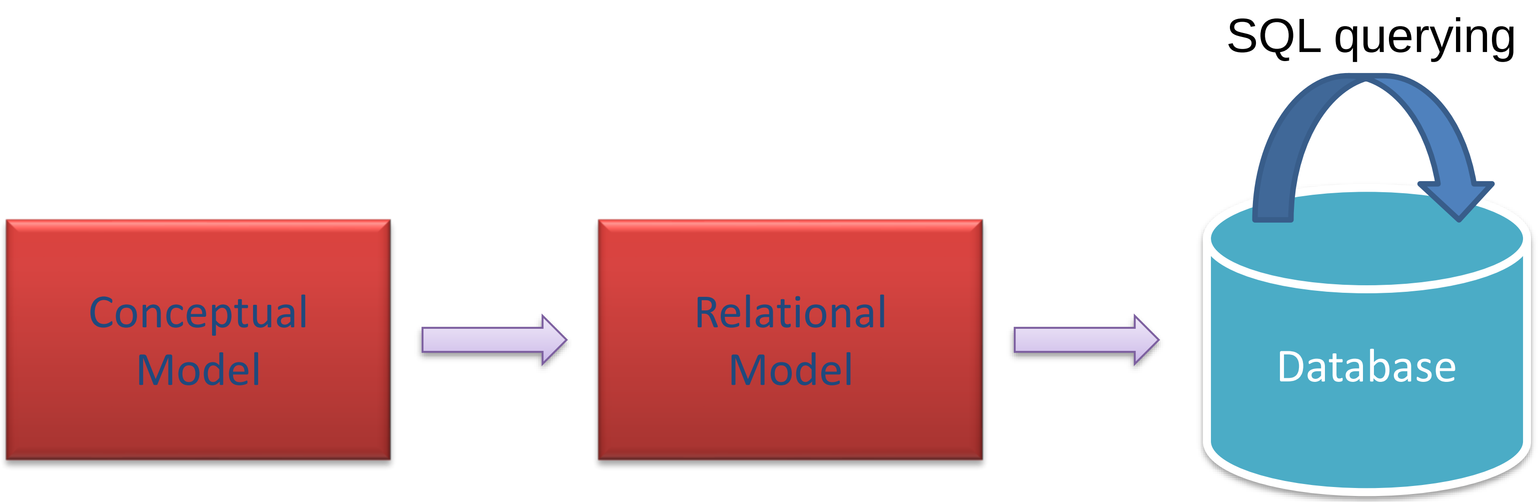
Offerings

Code	Semester	Lecturer
2ID45	Spring 2011	Calders
2ID05	Spring 2011	Fletcher

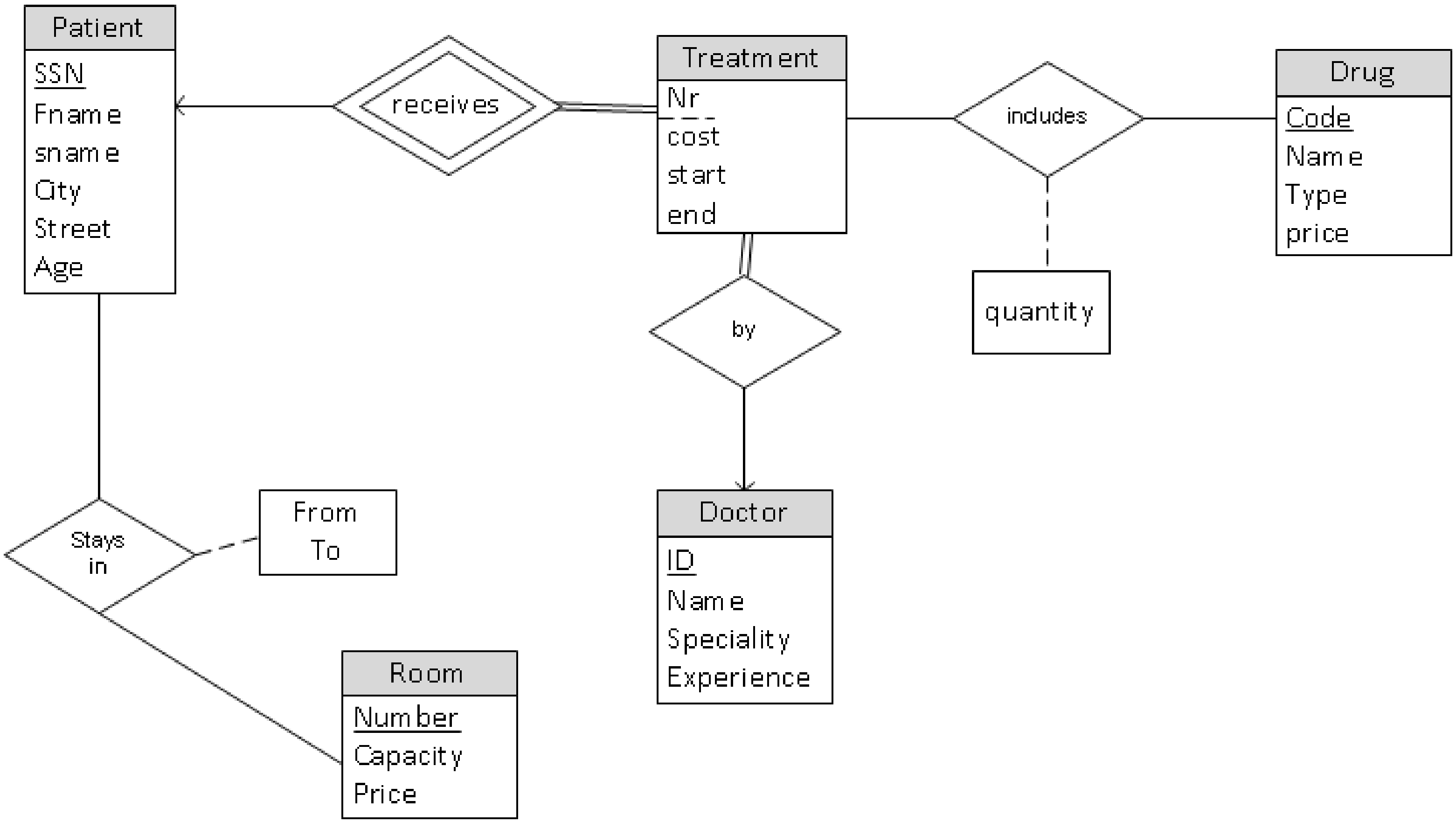
Follows

Student	Code	Semester	Grade
Phil	2ID45	Spring 2011	A+
Mary	2ID45	Spring 2011	C
John	2ID45	Spring 2011	B-
Paul	2ID05	Spring 2011	C

Revisiting Relational Database



ER Diagram



- Models entities and relations between them
  - “language” to write down constraints
  - documentation of the database design

Relational Model

- Relational Databases store the data in tables

  patient(<u>SSN</u>,fname,sname,city,street,age)
  doctor(<u>ID</u>,name,speciality,experience)
  treatment(<u>SSN</u>,<u>Nr</u>,ID,cost,start,end)
  drug(<u>code</u>,name,type,price)
  includes(<u>SSN</u>,<u>Nr</u>,<u>code</u>,quantity)
  room(<u>rnr</u>,capacity,price)
  stay(<u>SSN</u>,<u>rnr</u>,from,to)

- Good design =
  - No redundancy ➔ limit danger of inconsistencies
  - Constraints as much as possible covered by the design of the tables

Online Transaction Processing (OLTP)

- Relational database management systems are mainly to support transaction processing
  - Concurrent access
  - Data consistency, non-redundancy
  - Ad-hoc Querying
  - Efficiency

Powerful Language SQL

- Ad-hoc querying

```
SELECT fname, sname
FROM Customer
Where SSN="778944";
```

```
SELECT distinct S.name
FROM supplier S, transaction T, customer C
WHERE C.city="Brussels"
        and S.SID=T.SID and C.SSN=T.SSN;
```

```
SELECT S.City, sum(T.price), avg(T.price)
FROM supplier S, transaction T, customer C
WHERE C.city="Brussels"
        and S.SID=T.SID and C.SSN=T.SSN
GROUP BY S.City;
```

General-Purpose Language SQL

- Database engine optimizes queries
  - Makes a *query plan*
  - Using database statistics

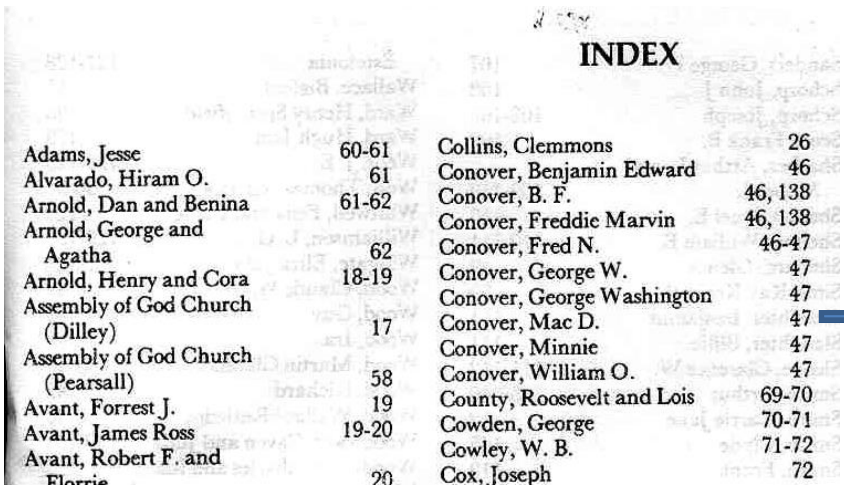
- General rule of thumb:
  The more powerful the query language, the more difficult it is to automatically optimize it

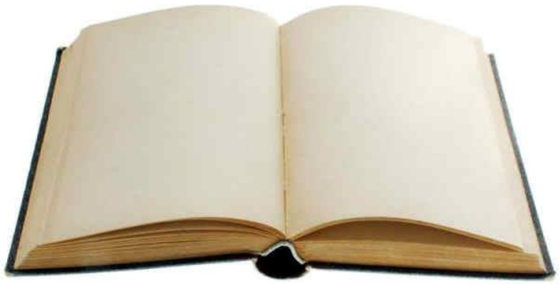
Online Transaction Processing (OLTP)

- Relational database management systems are mainly to support transaction processing
  - Concurrent access
  - Data consistency, non-redundancy
  - Ad-hoc Querying
  - Efficiency

Indexing Principle

No index


INDEX

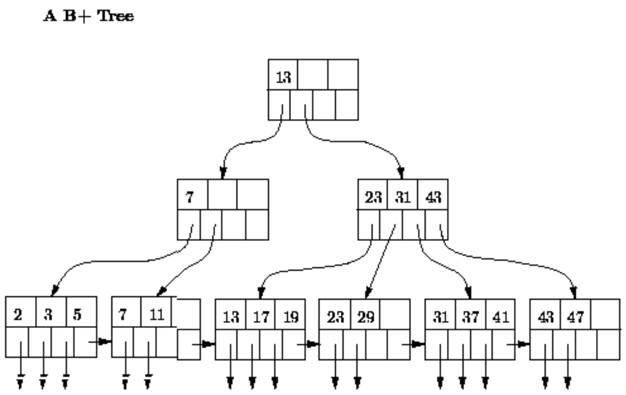


Indexing Principle

- Database Equivalent

No index → Expensive

Full table scan



→ Inexpensive

index lookup

+ Retrieve data page

Summary: Relational DBMS

- Strong in supporting OLTP
- Mainly aimed towards many, frequent, concurrent, small, ad-hoc queries

What About Decision Support?

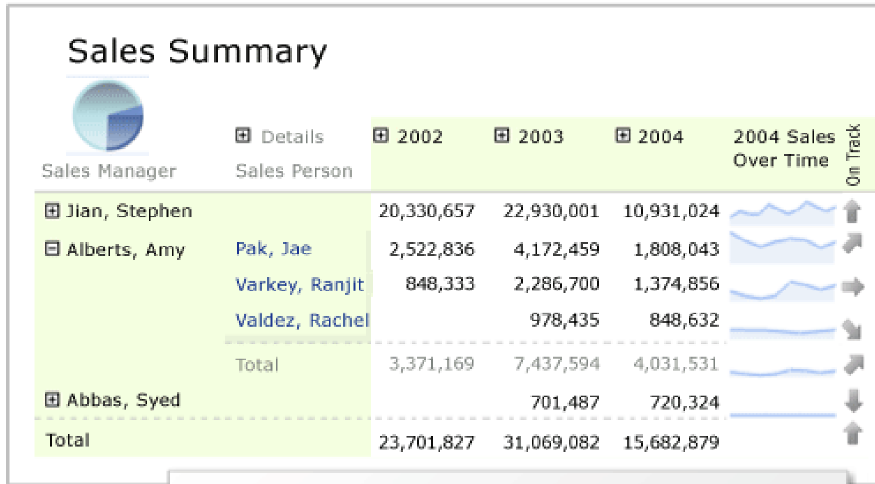
Decision support

- Off-line setting
- « Historical » data
- Summarized data
- Integrate different databases
- Statistical queries

Flight company

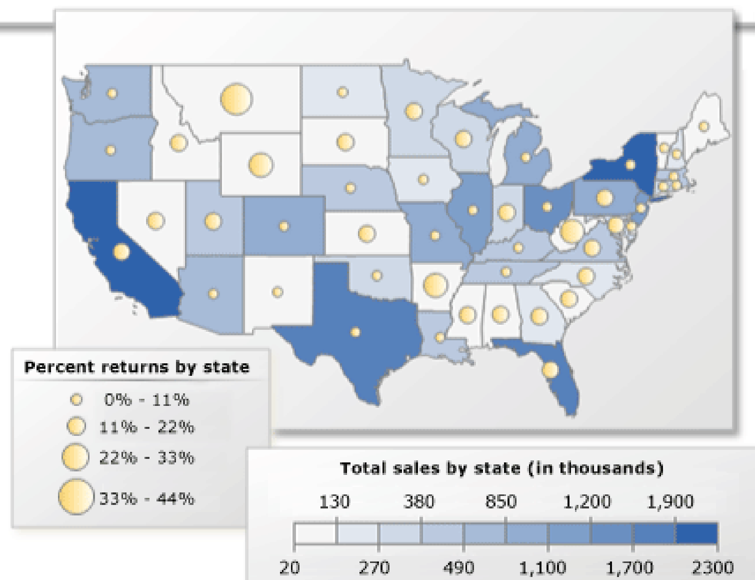
- Evaluate ROI flights
- Flights of last year
- # passengers per carrier for destination X
- Passengers, fuel costs, maintenance info
- Average % of seats sold/month/destination

Create Reports



Sales Summary

Sales Manager	Details Sales Person	2002	2003	2004	2004 Sales Over Time	On Track
Jian, Stephen		20,330,657	22,930,001	10,931,024		
Alberts, Amy	Pak, Jae	2,522,836	4,172,459	1,808,043		
	Varkey, Ranjit	848,333	2,286,700	1,374,856		
	Valdez, Rachel		978,435	848,632		
	Total	3,371,169	7,437,594	4,031,531		
Abbas, Syed			701,487	720,324		
Total		23,701,827	31,069,082	15,682,879		



Browse Data

Cube Viewer | Charts | MDX query editor | Layout | Log viewer | Drill Through viewer

				All Times				
					Q1	Q2	Q3	Q4
All Stores			Unit Sales	509,987	137,078	135,745	139,412	97,752
			Store Sales	1,079,147.47	290,873.18	287,009.99	295,040.55	206,223.75
	Canada		Unit Sales	46,157	11,160	12,885	12,966	9,146
			Store Sales	98,045.46	23,881.13	27,685.00	27,176.30	19,303.03
		BC	Unit Sales	46,157	11,160	12,885	12,966	9,146
			Store Sales	98,045.46	23,881.13	27,685.00	27,176.30	19,303.03
	Mexico		Unit Sales	203,914	56,133	54,005	57,872	35,904
			Store Sales	430,293.59	118,589.41	113,830.59	122,706.05	75,167.54
		DF	Unit Sales	45,223	12,058	12,818	12,962	7,385
			Store Sales	95,526.40	25,590.39	27,096.37	27,350.86	15,488.78
		Guerrero	Unit Sales	23,226	7,042	5,885	6,008	4,291
			Store Sales	49,090.03	15,063.14	12,301.53	12,755.76	8,969.60
		Jalisco	Unit Sales	2,124	666	637	492	329
			Store Sales	4,328.87	1,356.81	1,246.77	1,035.42	689.87
		Veracruz	Unit Sales	24,696	6,711	6,119	6,947	4,919
			Store Sales	52,142.07	13,970.82	13,114.47	14,727.55	10,329.23
		Yucatan	Unit Sales	37,143	9,766	9,372	11,205	6,800
			Store Sales	79,063.13	20,592.65	19,909.69	24,247.97	14,312.82
		Zacatecas	Unit Sales	71,502	19,890	19,174	20,258	12,180
			Store Sales	150,143.09	42,015.60	40,161.76	42,588.49	25,377.24
	USA		Unit Sales	259,916	69,785	68,855	68,574	52,702
			Store Sales	550,808.42	148,402.64	145,494.40	145,158.20	111,753.18

Example: Business Case

- Company selling different products
  - “units” of a high-tech material
  - different parameters
  - base product for other (high-tech) products
  - B2B scenario
- Company sees profit is dropping
  - Why?

Example: Business Case

- Different salesmen sell the products to their customers
  - Different price; result of negotiation
  - Transaction stored in sales database
    - Some transactions are to “compensate” incorrect transactions
  - There are seasonal effects (less sales in winter)
  - Data spread over different branches; formats are slightly different

Example: Business Case

Example Inc., August 2012

P&L Statement x1000 EUR	Actual 2012 August	Actual 2012 ytd August	Reference 2011	Budget 2012	Forecast 2012	Estimate 2012	Difference BE	Difference BF	Notes
Sales	4,237	32,916	3,987	53,000	49,374	52,000	1,000	3,626	
Total sales	**4,237**	**32,916**	**3,987**	**53,000**	**49,374**	**52,000**	**1,000**	**3,626**	
Costs of goods sold	1,983	15,405	1,866	24,804	23,107	24,336	468	1,697	Standard %
% of total sales	46.8%	46.8%	46.8%	46.8%	46.8%	46.8%			
Distribution cost	1,215	9,612	998	13,875	14,418	15,000	-1,125	-543	
% of total sales	28.7%	29.2%	25.0%	26.2%	29.2%	28.8%			
Gross margin	**1,039**	**7,899**	**1,123**	**14,321**	**11,849**	**12,664**	**1,657**	**2,472**	
% of total sales	**24.5%**	**24.0%**	**28.2%**	**27.0%**	**24.0%**	**24.4%**			
Expenses	214	1,712	211	2,568	2,568	2,568	0	0	Fixed
% of total sales	5.1%	5.2%	5.3%	4.8%	5.2%	4.9%			
Admin	115	920	112	1,380	1,380	1,380	0	0	Fixed
% of total sales	2.7%	2.8%	2.8%	2.6%	2.8%	2.7%			
R&D	36	312	42	465	468	465	0	-3	
% of total sales	0.8%	0.9%	1.1%	0.9%	0.9%	0.9%			
other	0	0	0	0	0	0	0	0	
EBITA	**674**	**4,955**	**758**	**9,908**	**7,433**	**8,251**	**1,657**	**2,475**	
% of total sales	**15.9%**	**15.1%**	**19.0%**	**18.7%**	**15.1%**	**15.9%**			
Depreciation	410	3,280	410	4,920	4,920	4,920	0	0	Fixed
% of total sales	9.7%	10.0%	10.3%	9.3%	10.0%	9.5%			
EBITDA	**264**	**1,675**	**348**	**4,988**	**2,513**	**3,331**	**1,657**	**2,475**	
% of total sales	**6.2%**	**5.1%**	**8.7%**	**9.4%**	**5.1%**	**6.4%**			

Figure 1.1: An example P&L statement.

Picture from MSc thesis Gerard de Ruig, TU/e

Example: Business Case

- Gathering the sales data took considerable time
- Data needed to be cleaned
- Analysis questions
  - Average, minimal, maximal price per region/salesman for comparable transactions
  - Average sales per product type and region
  - Evolution of sales this year over time, compared to last year’s sales

Example: Business Case

- Typically: want to browse the data
  - Explore
  - Concentrate on certain slices of the data
  - Refine analysis in a suspicious region
  - …
- Almost impossible using original data sources and OLTP-geared systems

Requirements for Decision Support?

- Concurrent access
  - →not really
  - →read-only

- Data consistency, non-redundancy
  - →data comes from consistent sources (sort of)
  - →data does not change during analysis; once clean, always clean

Requirements for Decision Support?

- Ad-hoc Querying
  - →No longer true;
  - →Spread-sheet like queries
  - →Long-running queries, touching large parts of the database
    - →In combination with transactions, kills the database

- Efficiency
  - → Relational DBMS optimized for other types of queries

Requirements for Decision Support?

- OLTP systems not very efficient for data analysis tasks
  - analysis queries might stall operational systems
  - architecture suboptimal
    - different indexing stuctures
    - denormalization
  - need of historical data versus only current data

Outline

Online Analytical Processing

- Data Warehouses
- Conceptual model: Data Cubes
- Query languages for supporting OLAP
  - Typical data cube operations
  - SQL extensions
  - MDX
- Database Explosion Problem

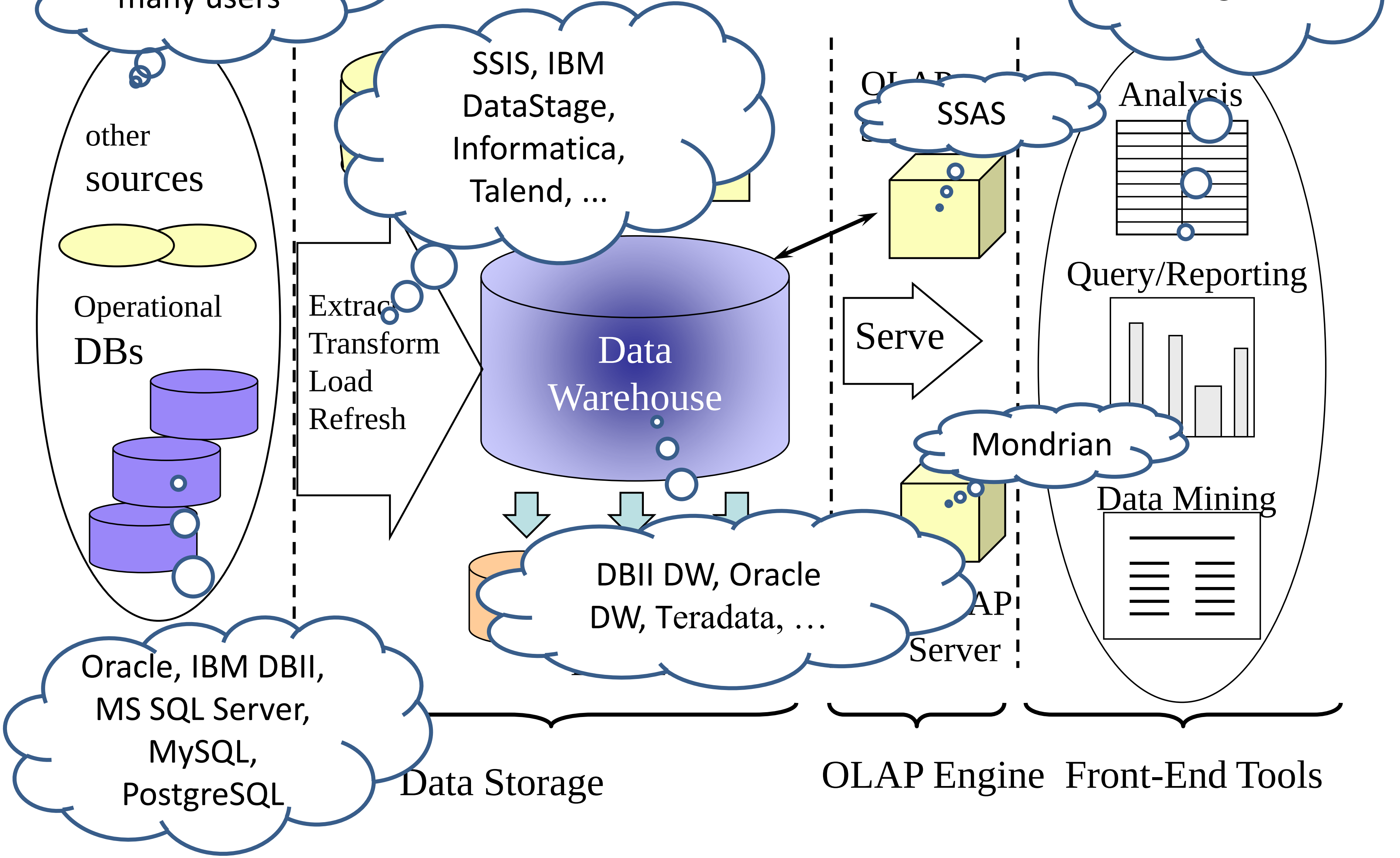
Data Warehouse

- A decision support DB maintained separately from the operational databases.
- Why Separate Data Warehouse?
  - Different functions
    - DBMS— tuned for OLTP
    - Warehouse—tuned for OLAP
  - Different data
    - Decision support requires historical data
  - Integration of data from heterogeneous sources

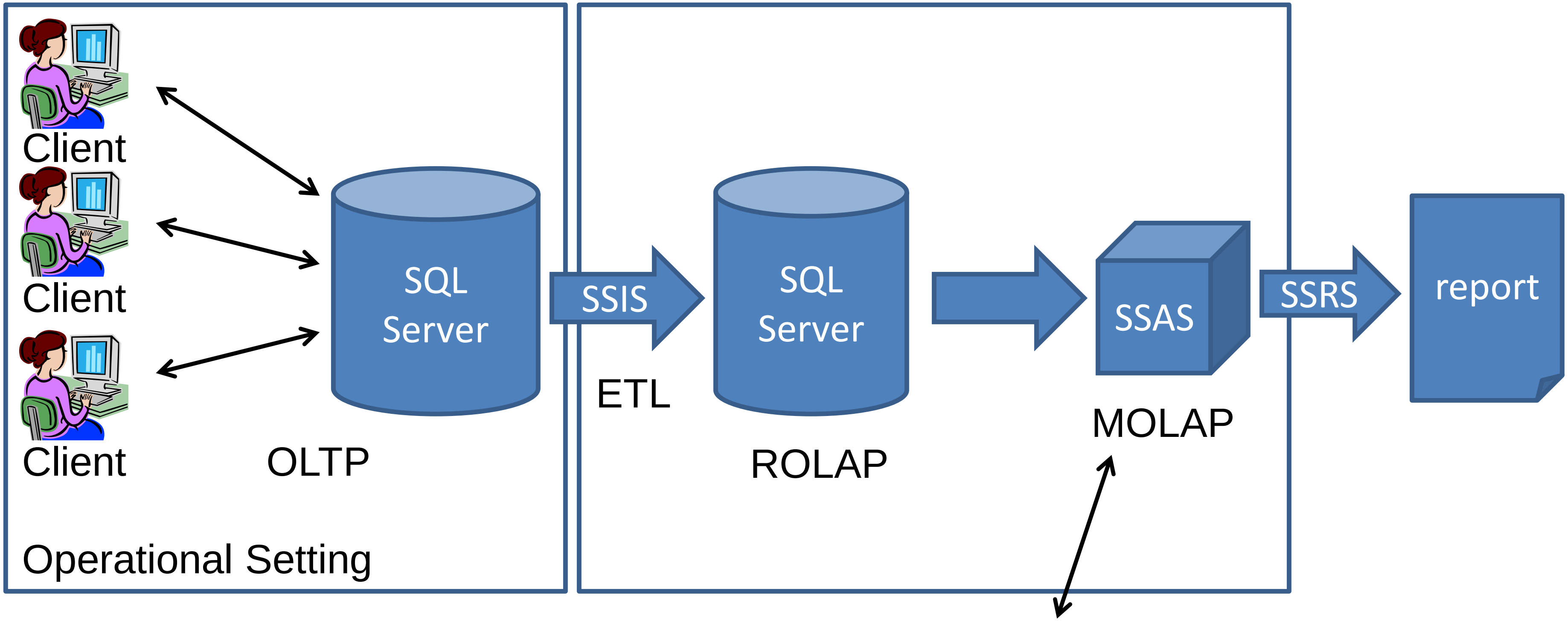
Data Warehouse

- Data Warehouse is
  - Subject-oriented (vs function-oriented)
  - Non-volatile (vs only holding most recent version)
  - Integrated (different data sources)
  - Time-variant (can be related to time)
  - Supporting decision support

ee-Tier Architectur



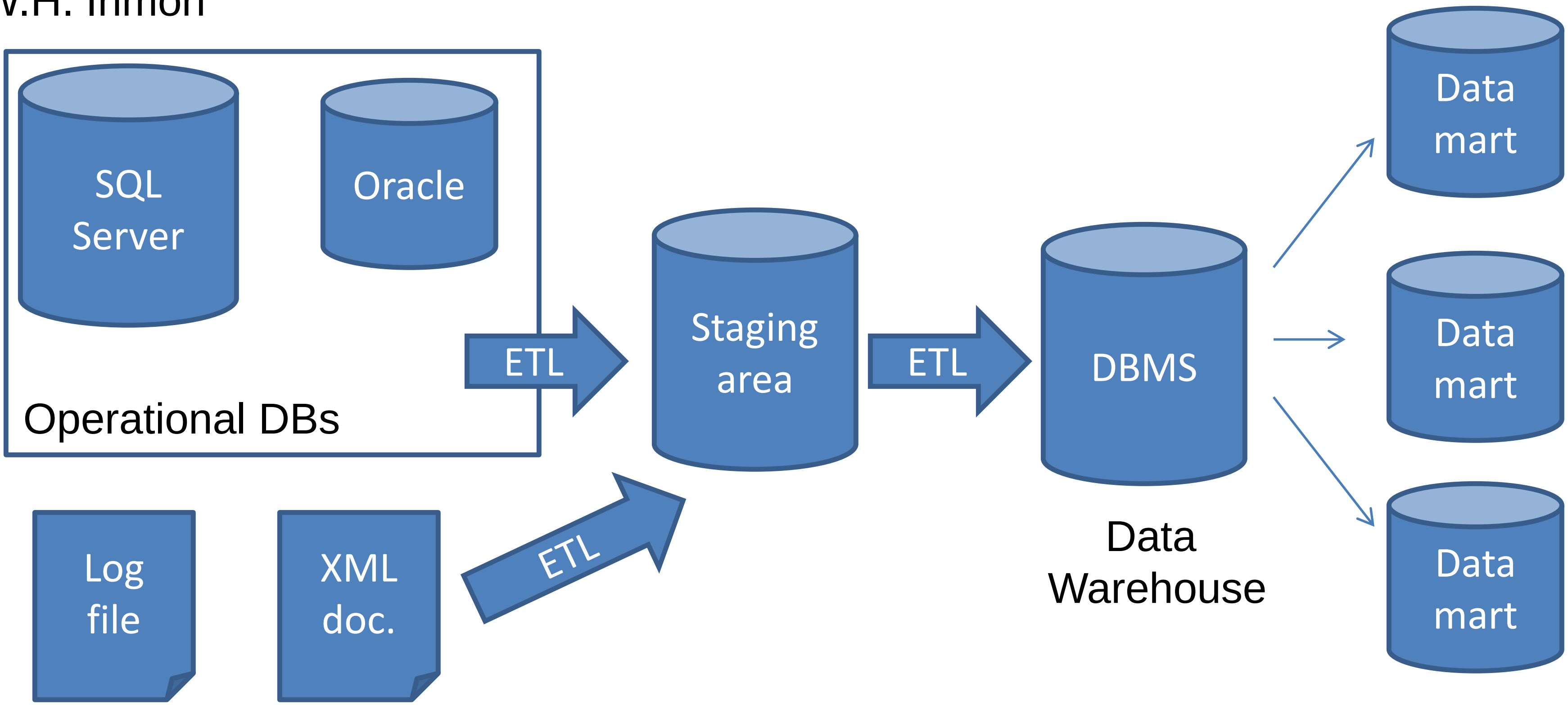
Example: MS SQLSERVER



SSIS: SQL Server Integration Services
SSAS: SQL Server Analysis Services
SSRS: SQL Server Reporting Services

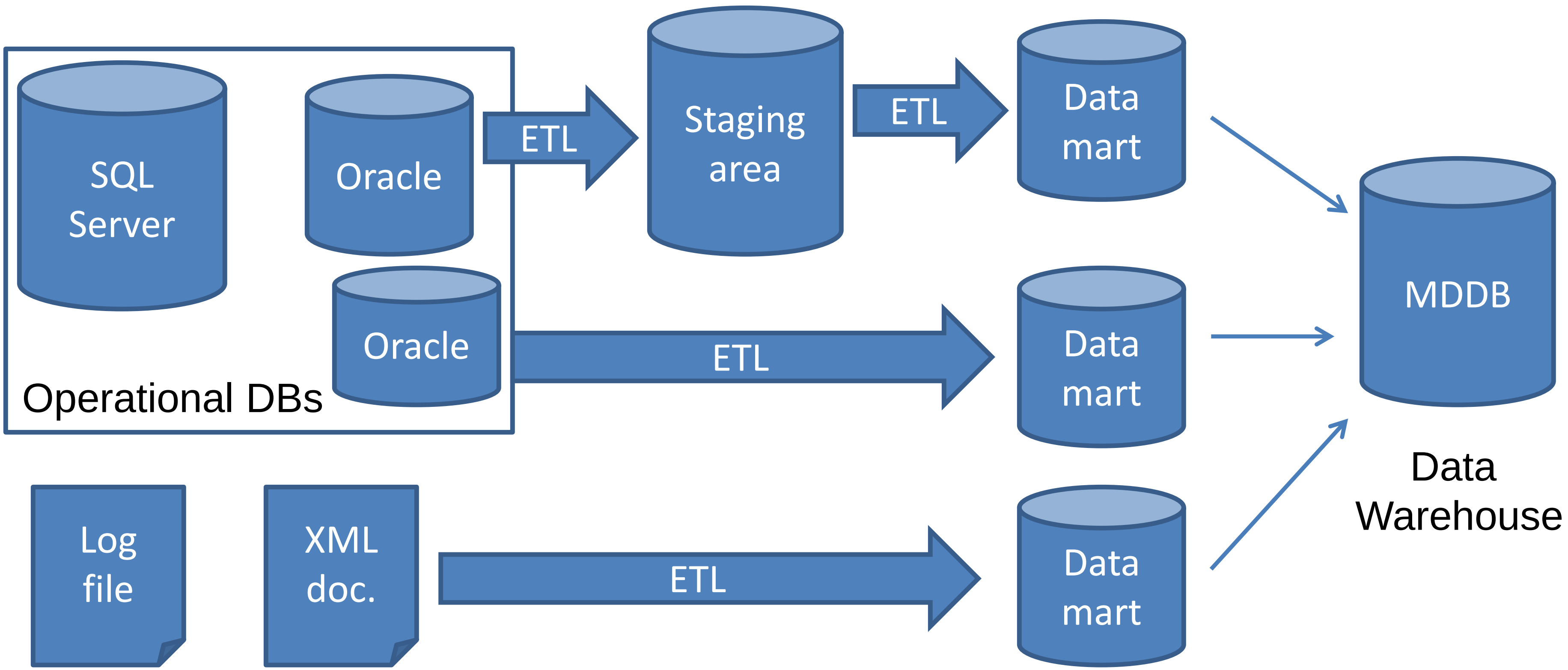
W.H. Inmon

Example: Top-Down



Ralph Kimball

Example: Bottom-Up



OLAP

- OLAP = OnLine Analytical Processing
  - Online = no waiting for answers

- OLAP system = system that supports *analytical queries* that are *dimensional* in nature.

- Most data warehousing systems support OLAP functionalities

Outline

Online Analytical Processing

- Conceptual model: Data cubes
- Query languages for supporting OLAP
  - Typical data cube operations
  - SQL extensions
  - MDX
- Database Explosion Problem

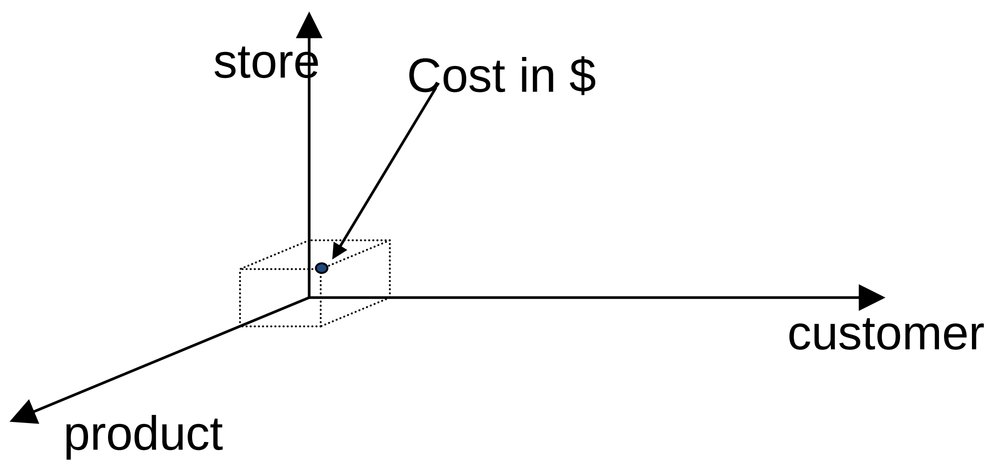
Supermarket Example

- Evaluate the sales of products
  - Product cost in $
  - Customer: ID, city, state, country,
  - Store: chain, size, location,
  - Product: brand, type, …
  - …

measure

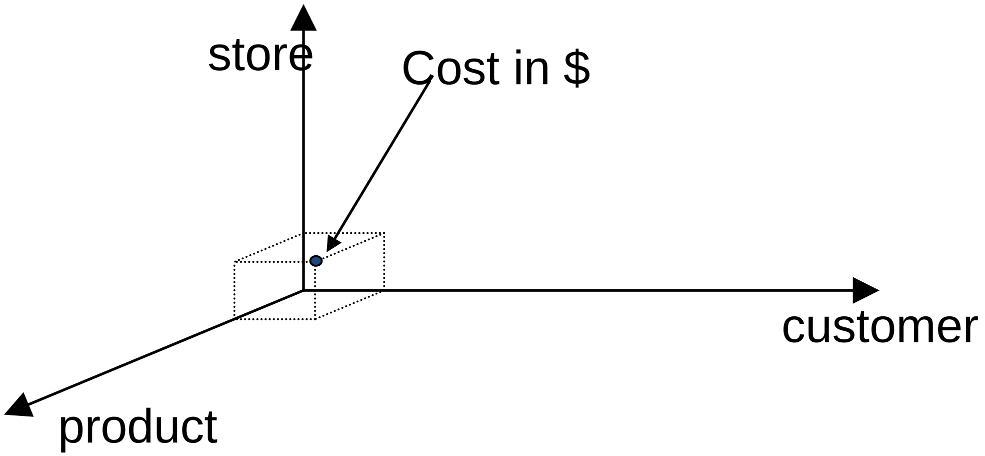
Dim.

hierarchies



Supermarket Example

- Multi-dimensional view on data



Cross Tabulation

- Cross-tabulations are highly useful
  - Sales of clothes June→August '06

Product: color

Date:month, June→August 2006

	Blue	Red	Orange	Total
June	51	25	158	234
July	58	20	120	198
August	65	22	51	138
Total	174	67	329	570

Data Cubes

- Extension of Cross-Tables to multiple dimensions
  - *Conceptual* notion

Dimensions

	Blue	Red	Orange	Total
June	51	25	158	234
July	58	20	120	198
August	65	22	51	138
Total	174	67	329	570

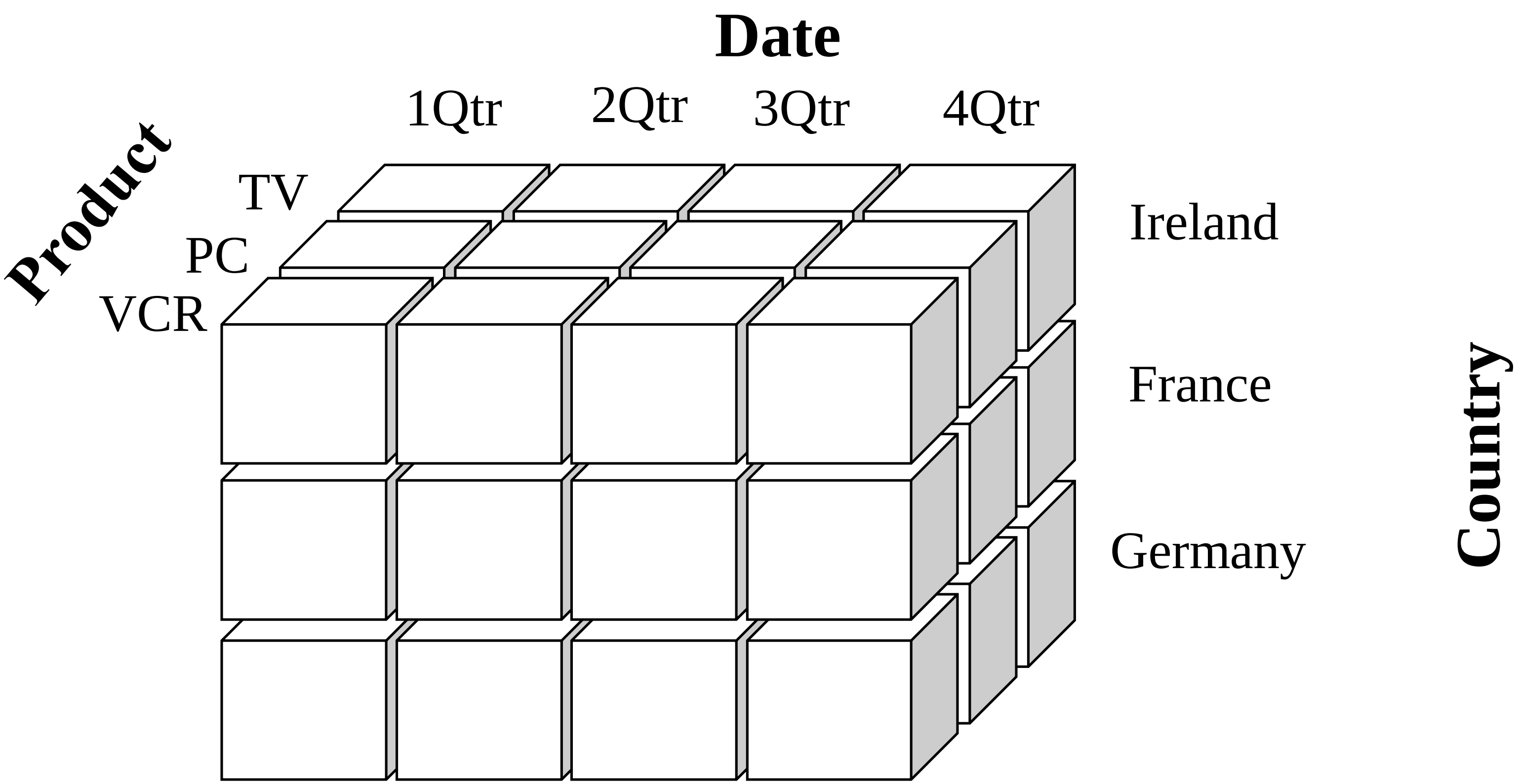
Data Points/
1st level of aggregation

Aggregated w.r.t. X-dim

Aggregated w.r.t. Y-dim

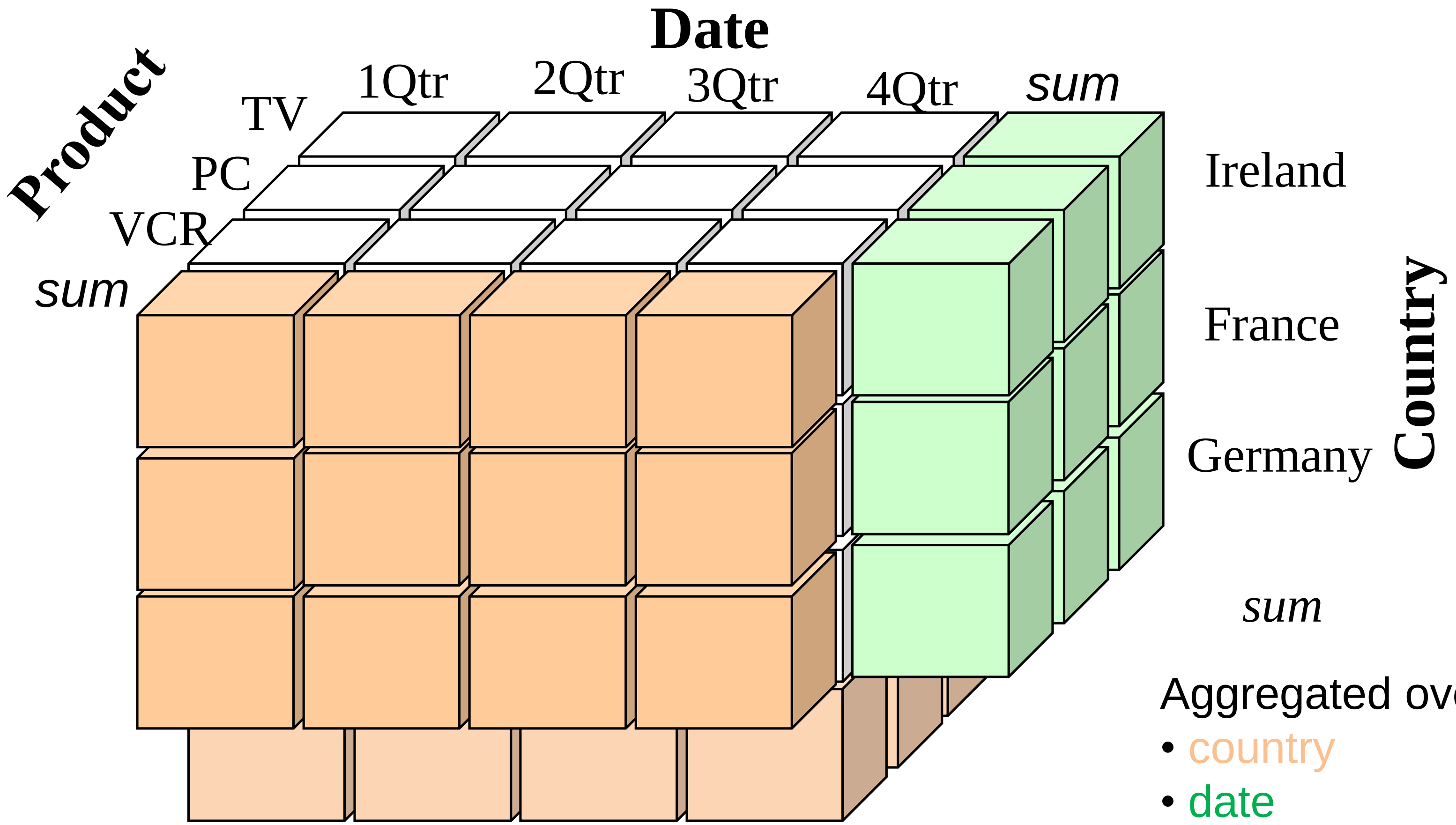
Aggregated w.r.t. X and Y

Data Cubes



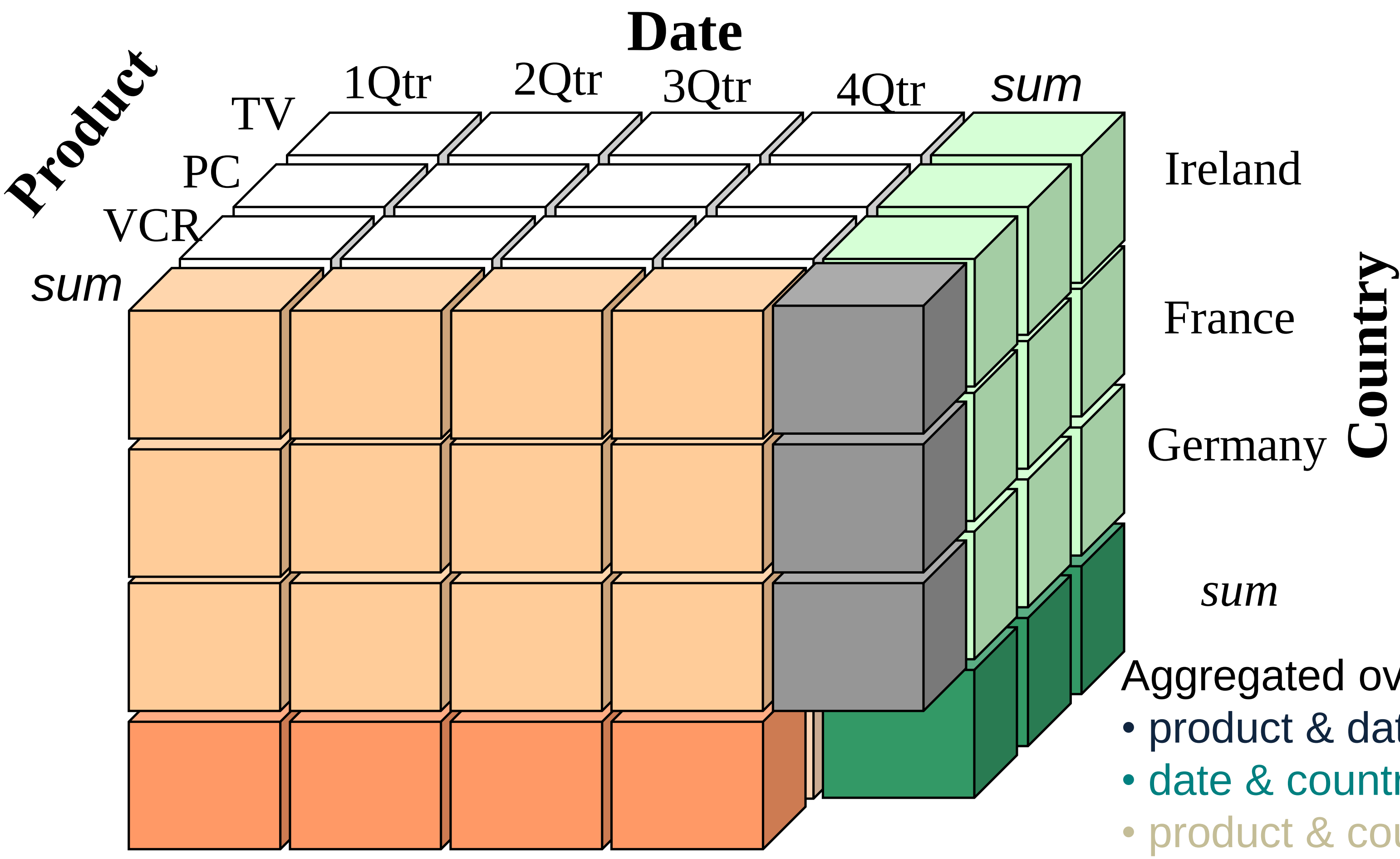
Date
1Qtr
2Qtr
3Qtr
4Qtr
Product
TV
PC
VCR
Ireland
France
Germany
Country

Data Cubes



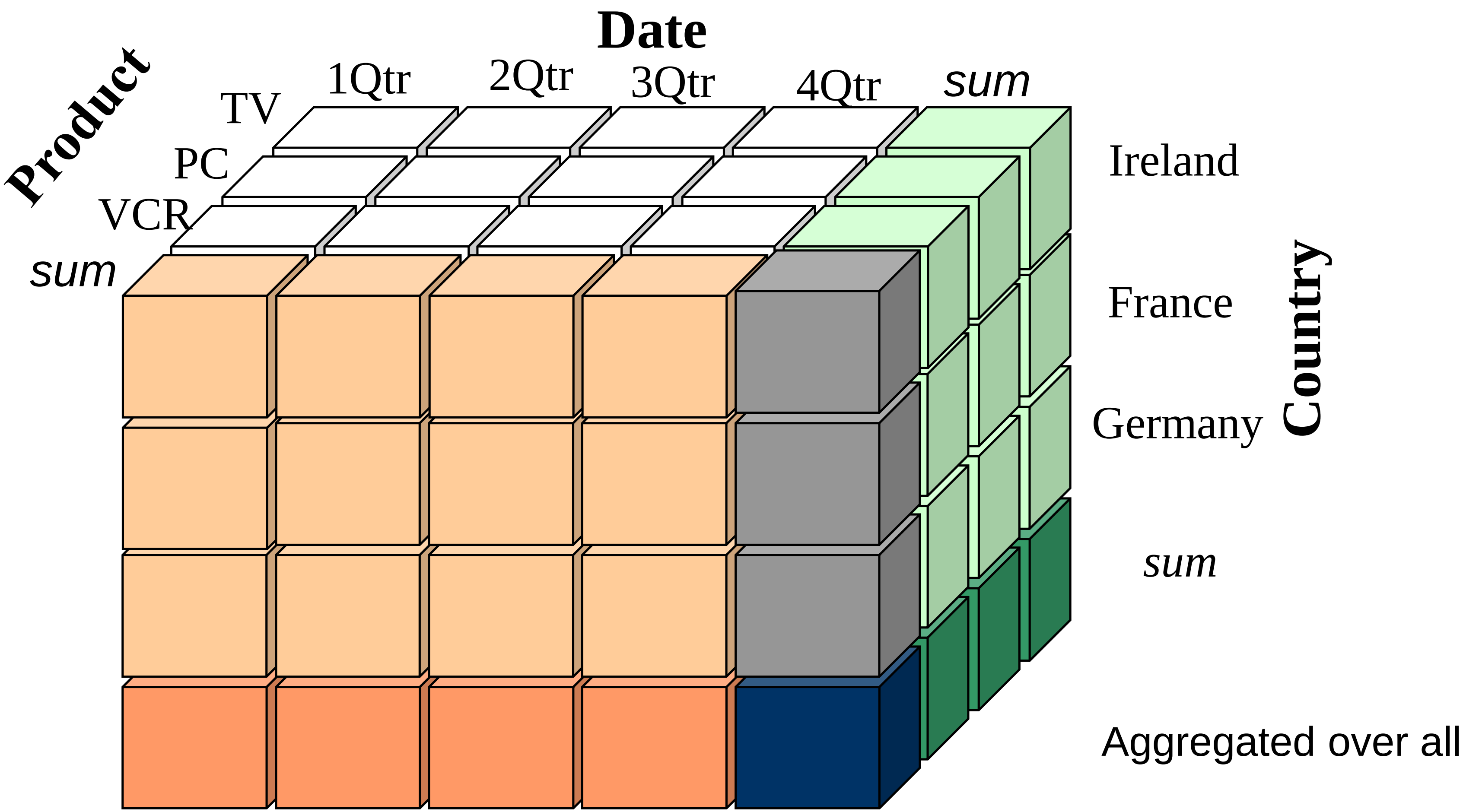
Aggregated over
- country
- date
- product

Data Cubes

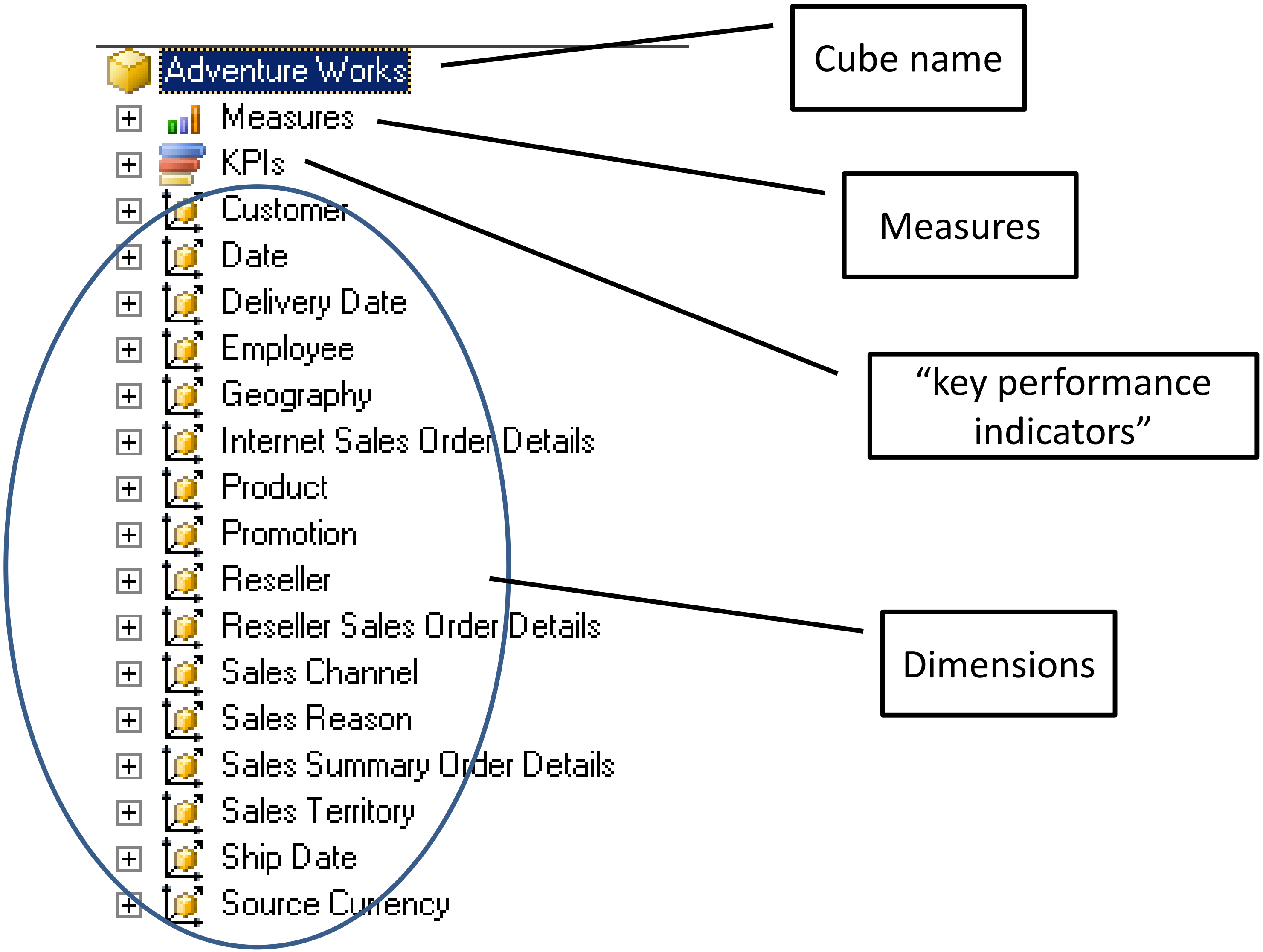


Aggregated over
- product & date
- date & country
- product & country

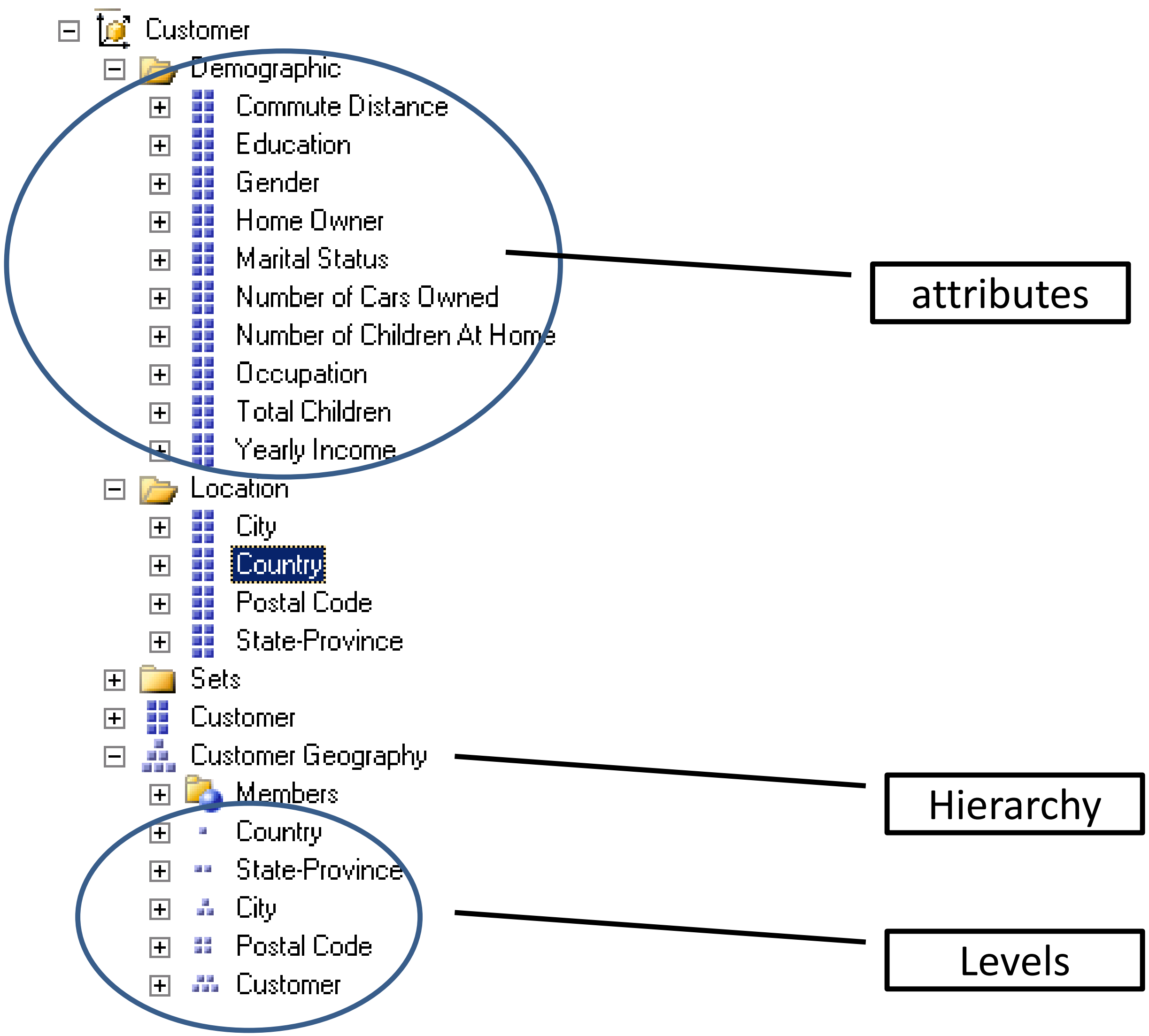
Data Cubes


Date
1Qtr
2Qtr
3Qtr
4Qtr
sum
Product
TV
PC
VCR
sum
Ireland
France
Germany
sum
Country
Aggregated over all

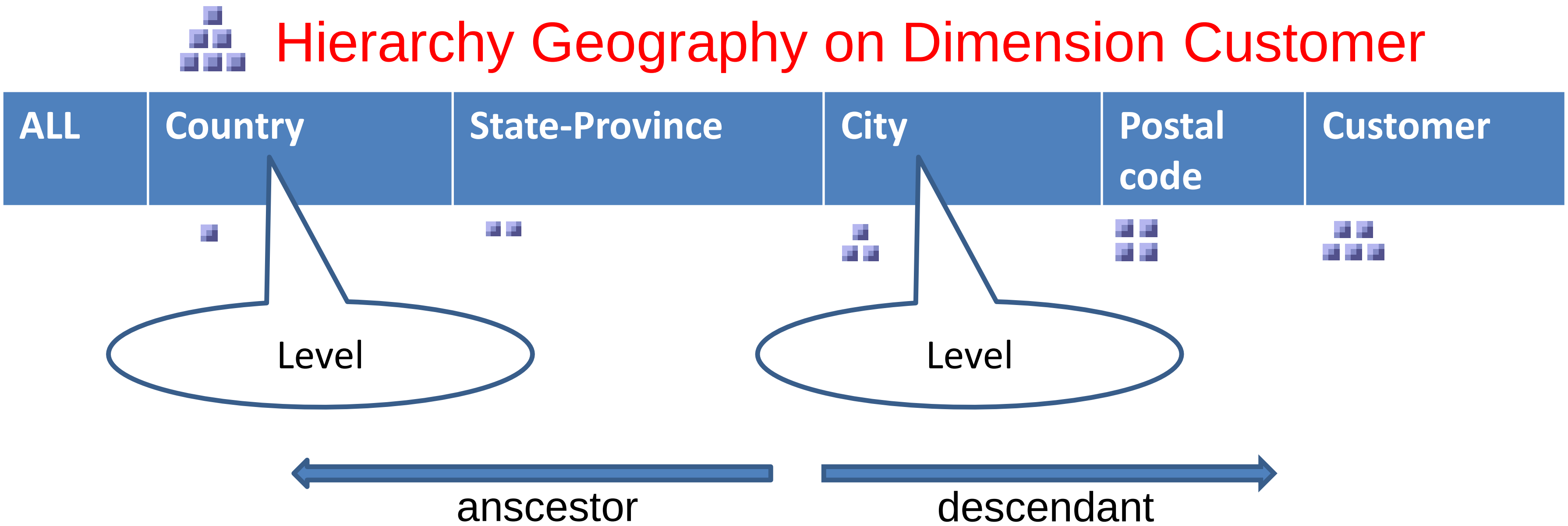
SSAS – Data Cubes



SSAS - Dimension



Hierarchy, Level



- 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

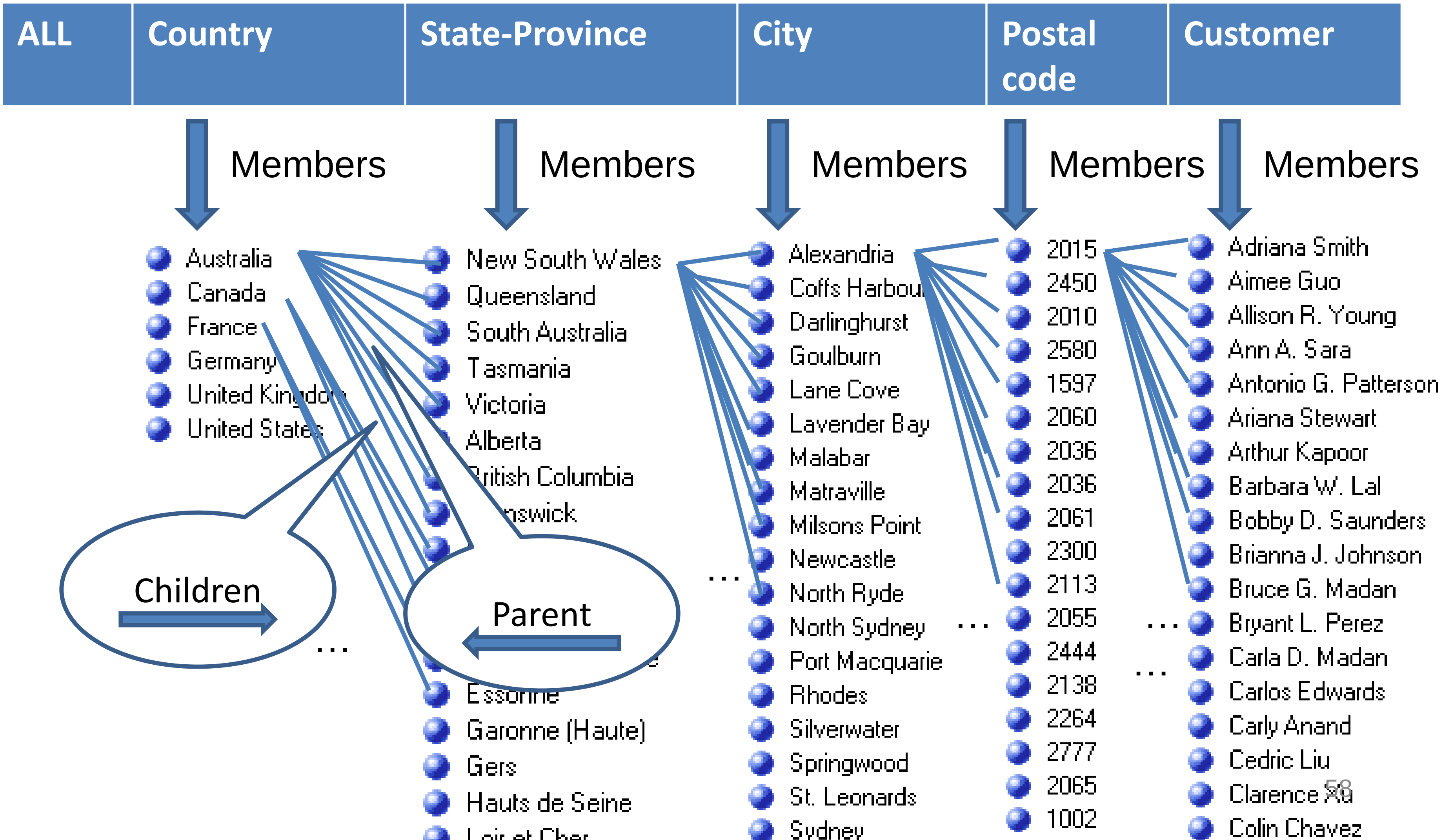
- 2015
- 2450
- 2010
- 2580
- 1597
- 2060
- 2036
- 2036
- 2061
- 2300
- 2113
- 2055
- 2444
- 2138
- 2264
- 2777
- 2065
- 1002

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

Children, Parent

Hierarchy Geography on Dimension Customer



Outline

Online Analytical Processing

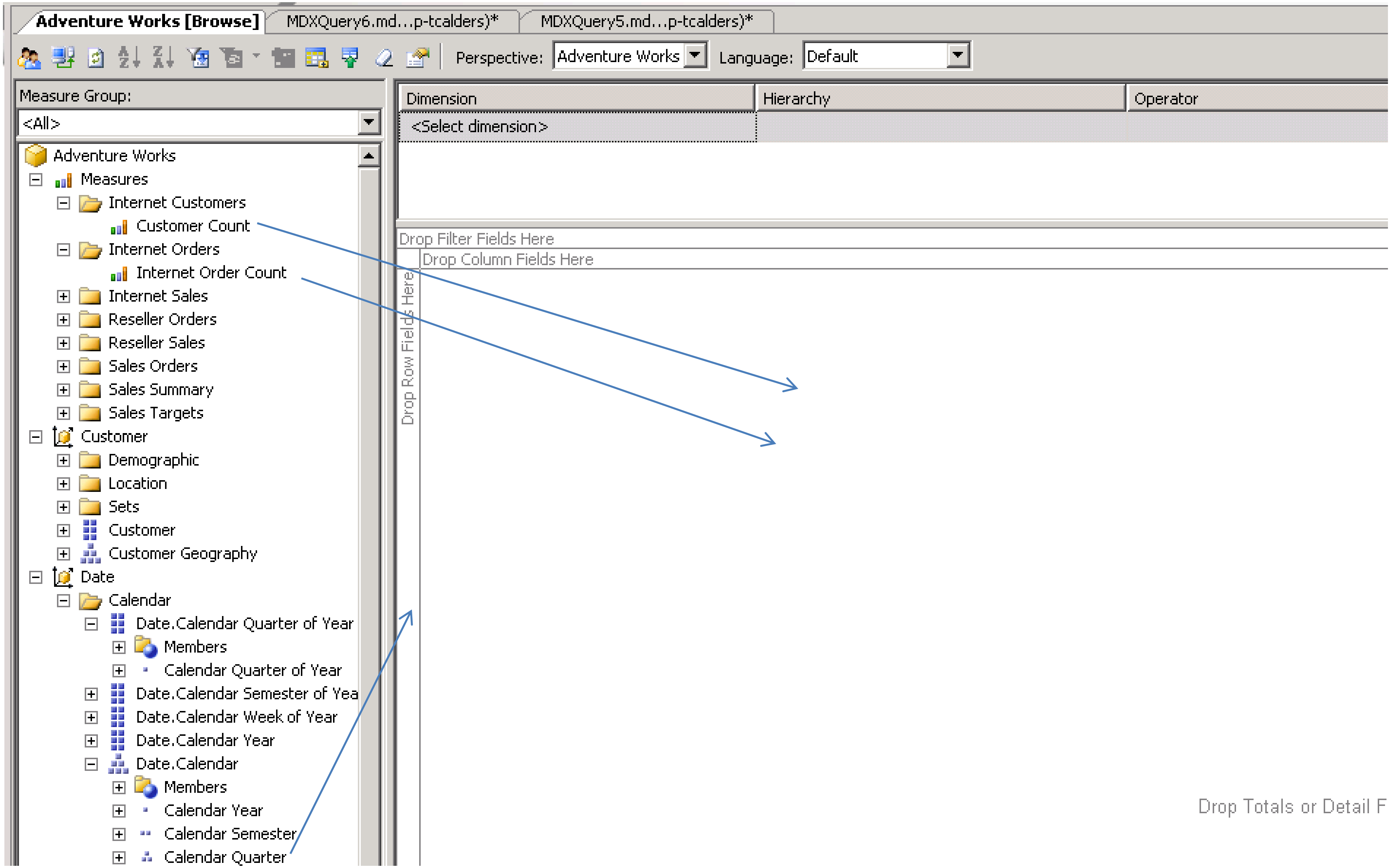
- Conceptual model: Data cubes
- Query languages for supporting OLAP
  - Typical data cube operations
  - SQL extensions
  - MDX
- Database Explosion Problem

Pivoting

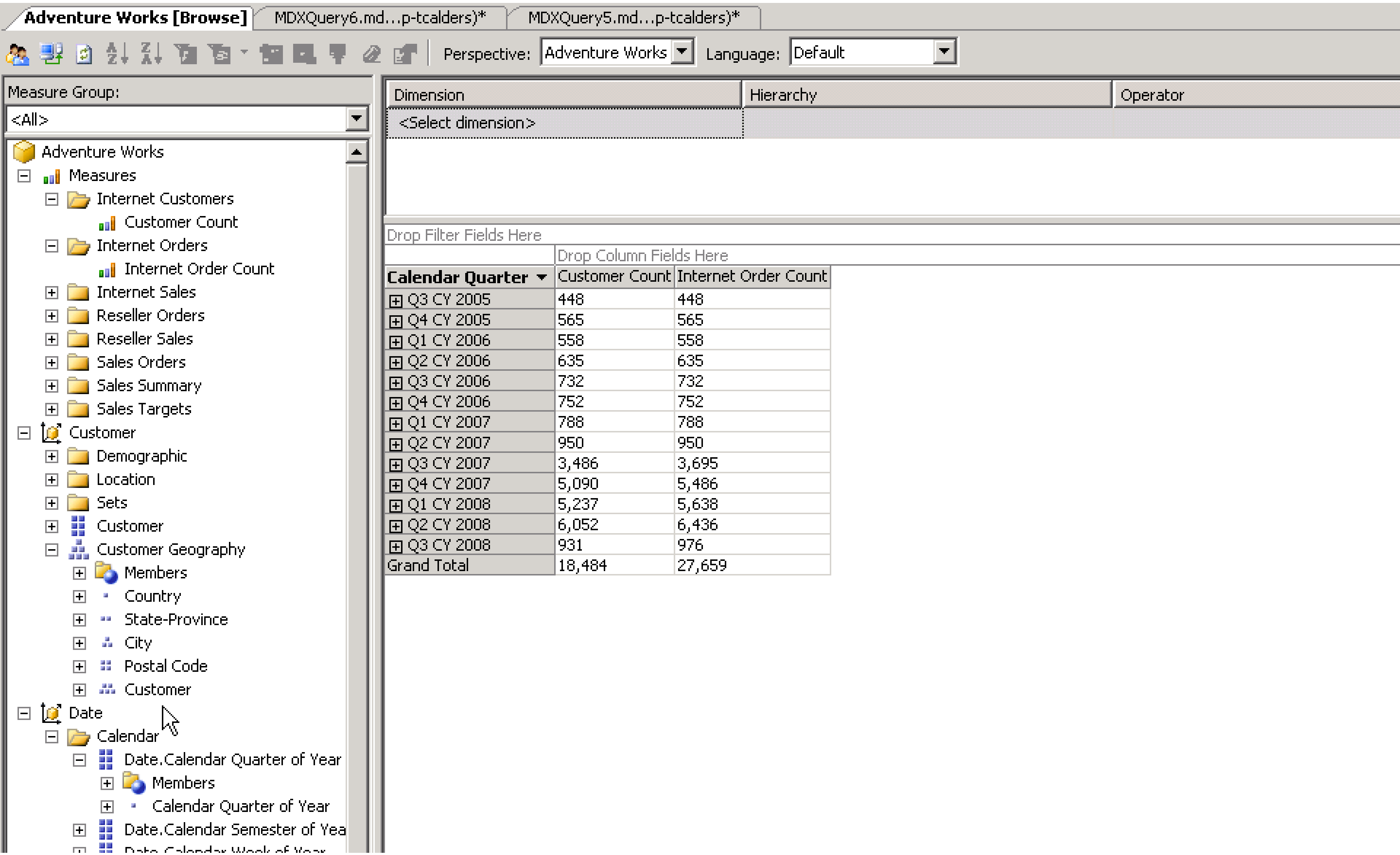
- Change the dimensions that are "displayed"; select a cross-tab.
  - look at the cross-table for product-date
  - display cross-table for date-customer

Sales	Date			
		1st sem	2nd sem	Total
Country	Ireland	20	23	**43**
	France	126	138	**264**
	Germany	56	48	**104**
	Total	**202**	**209**	**411**

Browsing a Cube

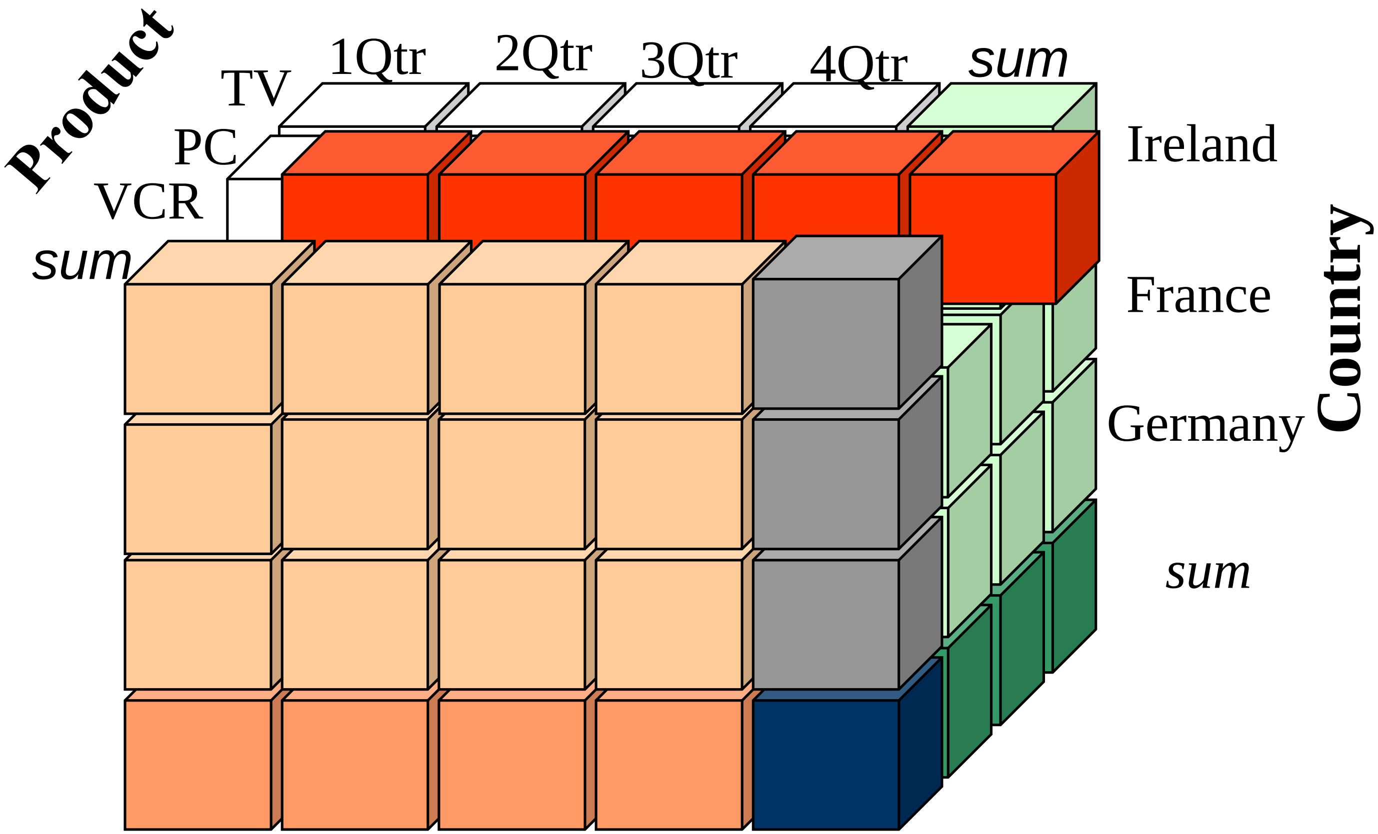


Browsing a Cube

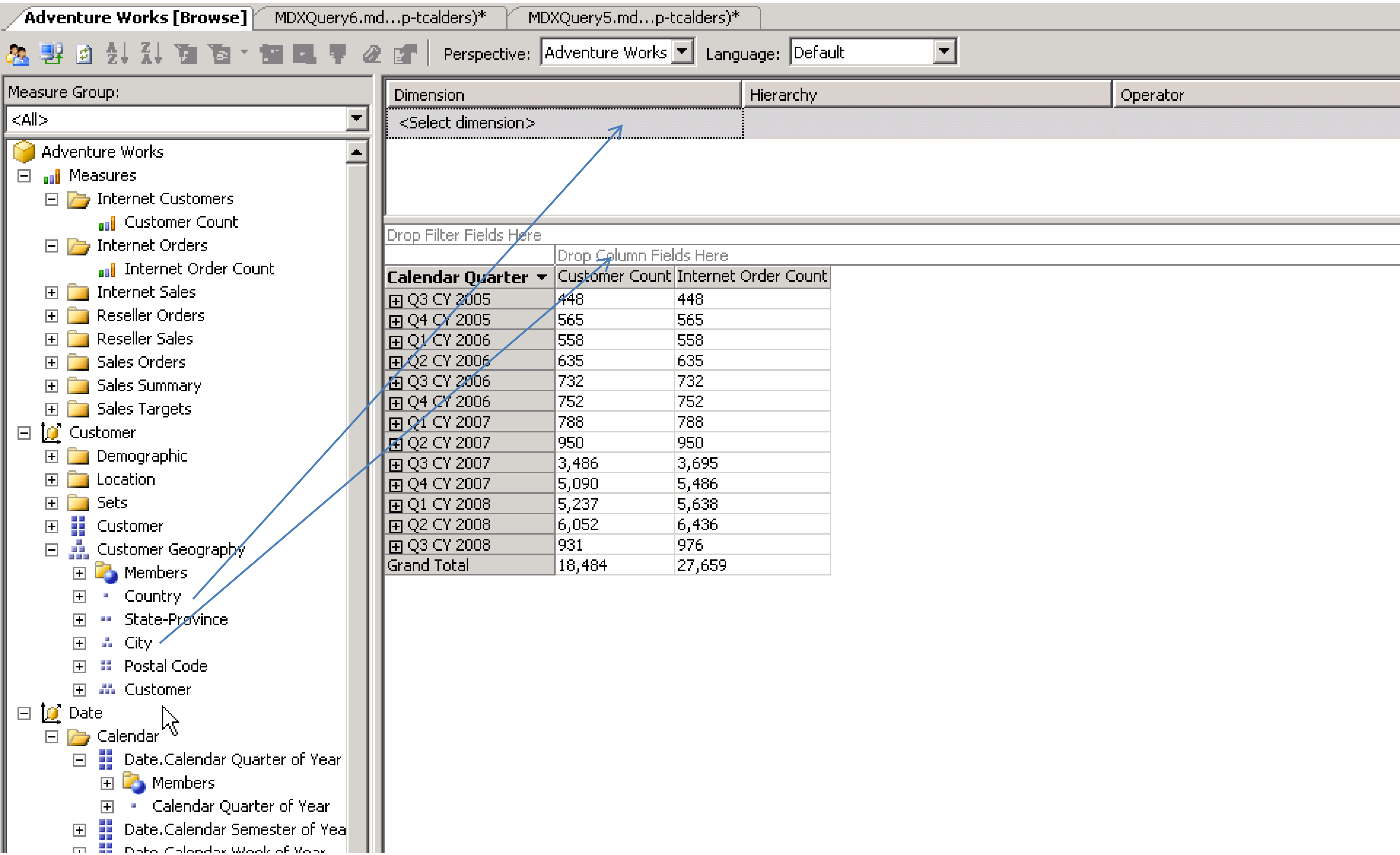


Slicing

- Select a part of the cube by restricting one or more dimensions to some values



Browsing a Cube



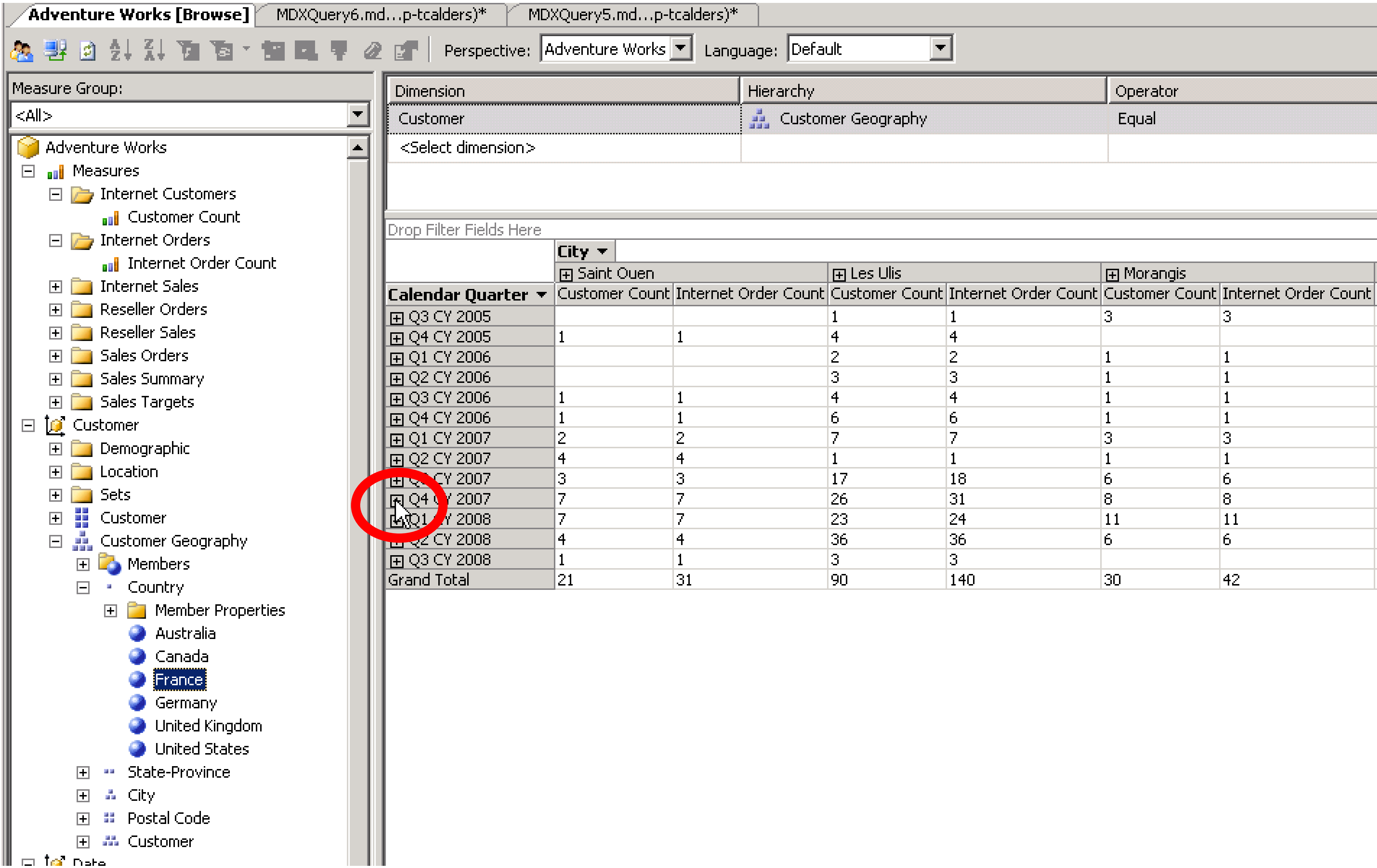
Calendar Quarter	Customer Count	Internet Order Count
Q3 CY 2005	448	448
Q4 CY 2005	565	565
Q1 CY 2006	558	558
Q2 CY 2006	635	635
Q3 CY 2006	732	732
Q4 CY 2006	752	752
Q1 CY 2007	788	788
Q2 CY 2007	950	950
Q3 CY 2007	3,486	3,695
Q4 CY 2007	5,090	5,486
Q1 CY 2008	5,237	5,638
Q2 CY 2008	6,052	6,436
Q3 CY 2008	931	976
Grand Total	18,484	27,659

Drill-down and Roll-Up

- Change level to a descendant in the hierarchy
  - city → store
  - country → cities
  - product type → product
- Roll-up = inverse operation

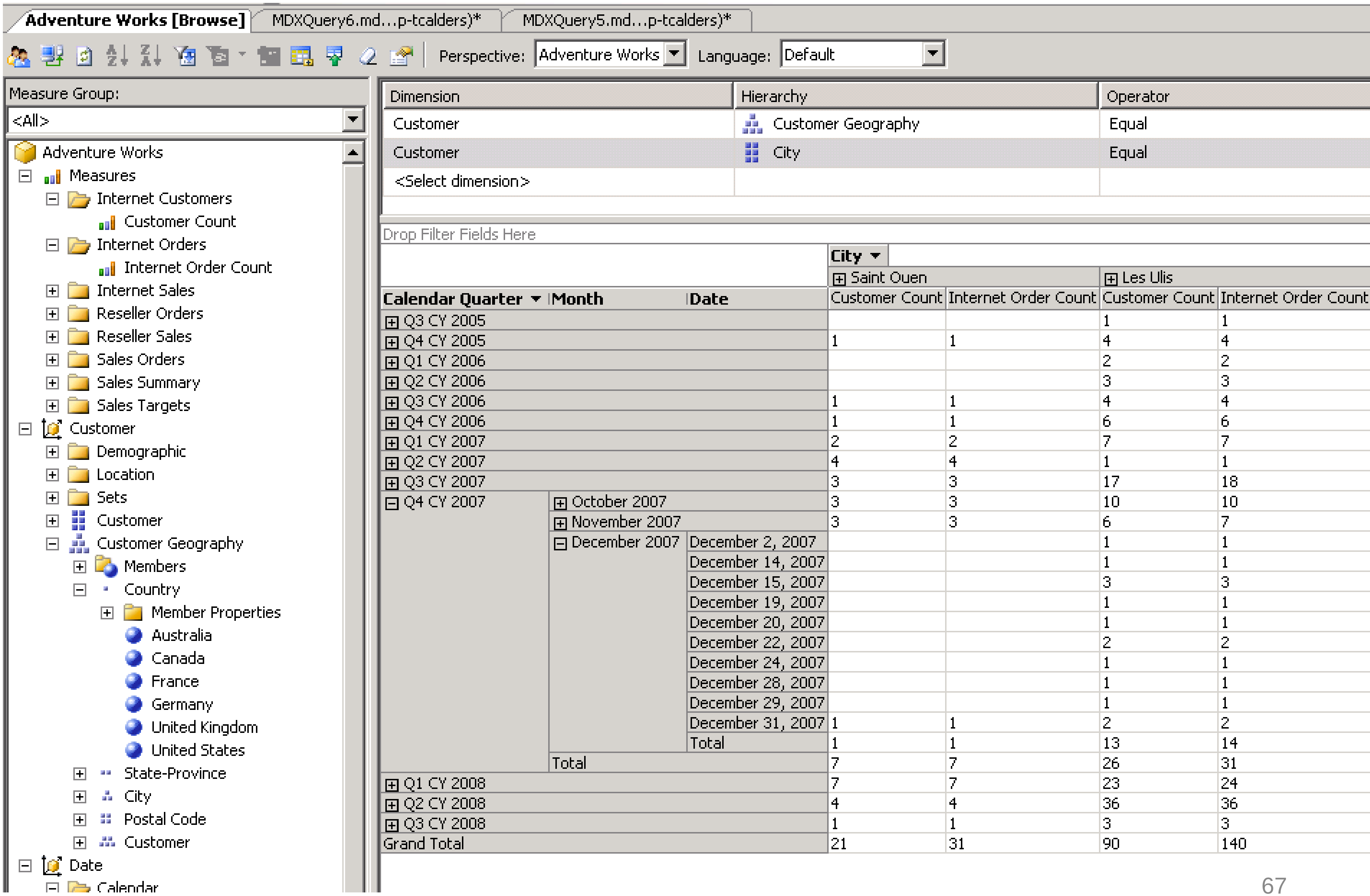
- Drill-through:
  - go back to the original, individual data records

Browsing a Cube

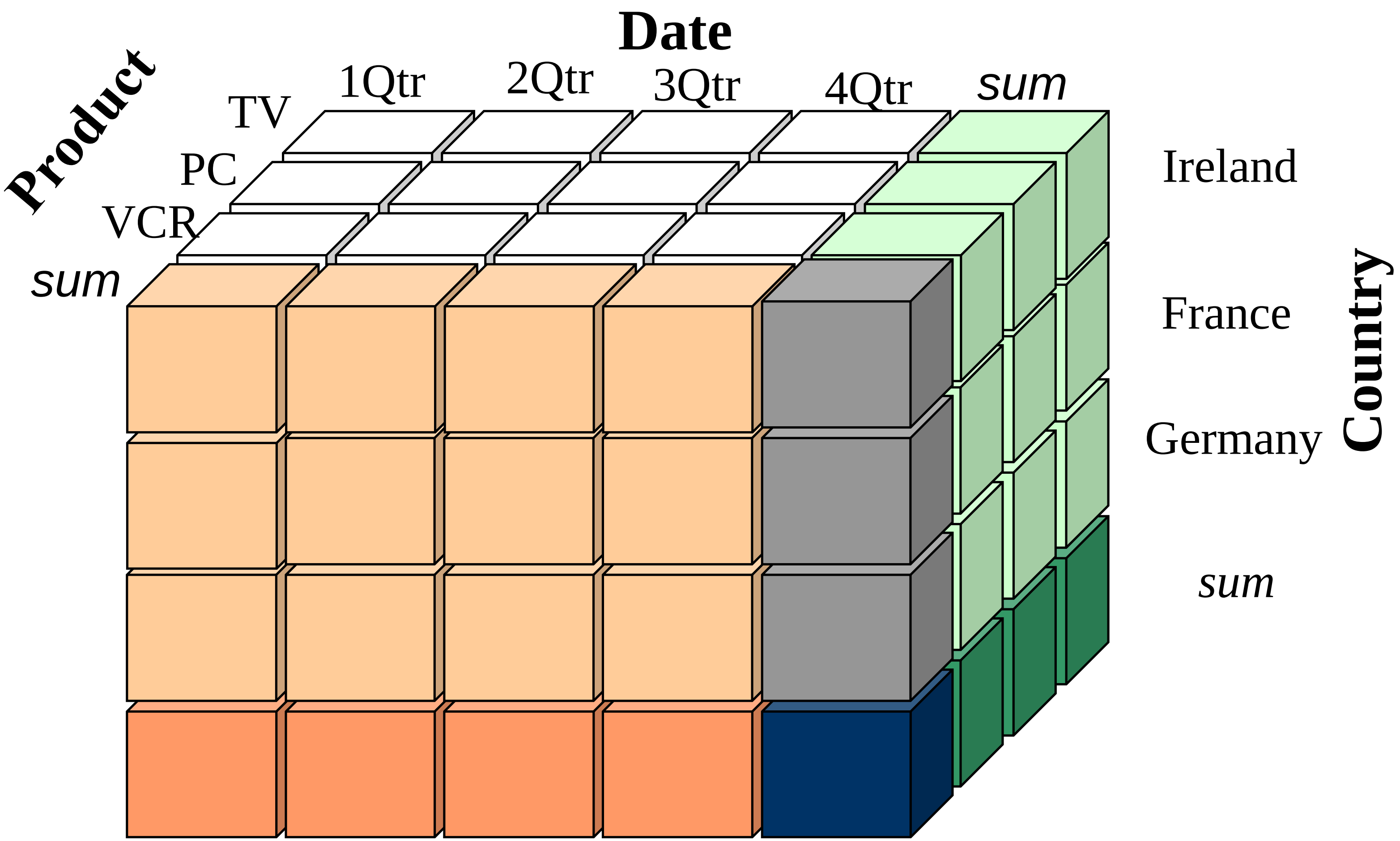


	Saint Ouen		Les Ulis		Morangis	
Calendar Quarter	Customer Count	Internet Order Count	Customer Count	Internet Order Count	Customer Count	Internet Order Count
Q3 CY 2005			1	1	3	3
Q4 CY 2005	1	1	4	4		
Q1 CY 2006			2	2	1	1
Q2 CY 2006			3	3	1	1
Q3 CY 2006	1	1	4	4	1	1
Q4 CY 2006	1	1	6	6	1	1
Q1 CY 2007	2	2	7	7	3	3
Q2 CY 2007	4	4	1	1	1	1
Q3 CY 2007	3	3	17	18	6	6
Q4 CY 2007	7	7	26	31	8	8
Q1 CY 2008	7	7	23	24	11	11
Q2 CY 2008	4	4	36	36	6	6
Q3 CY 2008	1	1	3	3		
Grand Total	21	31	90	140	30	42

Browsing a Cube

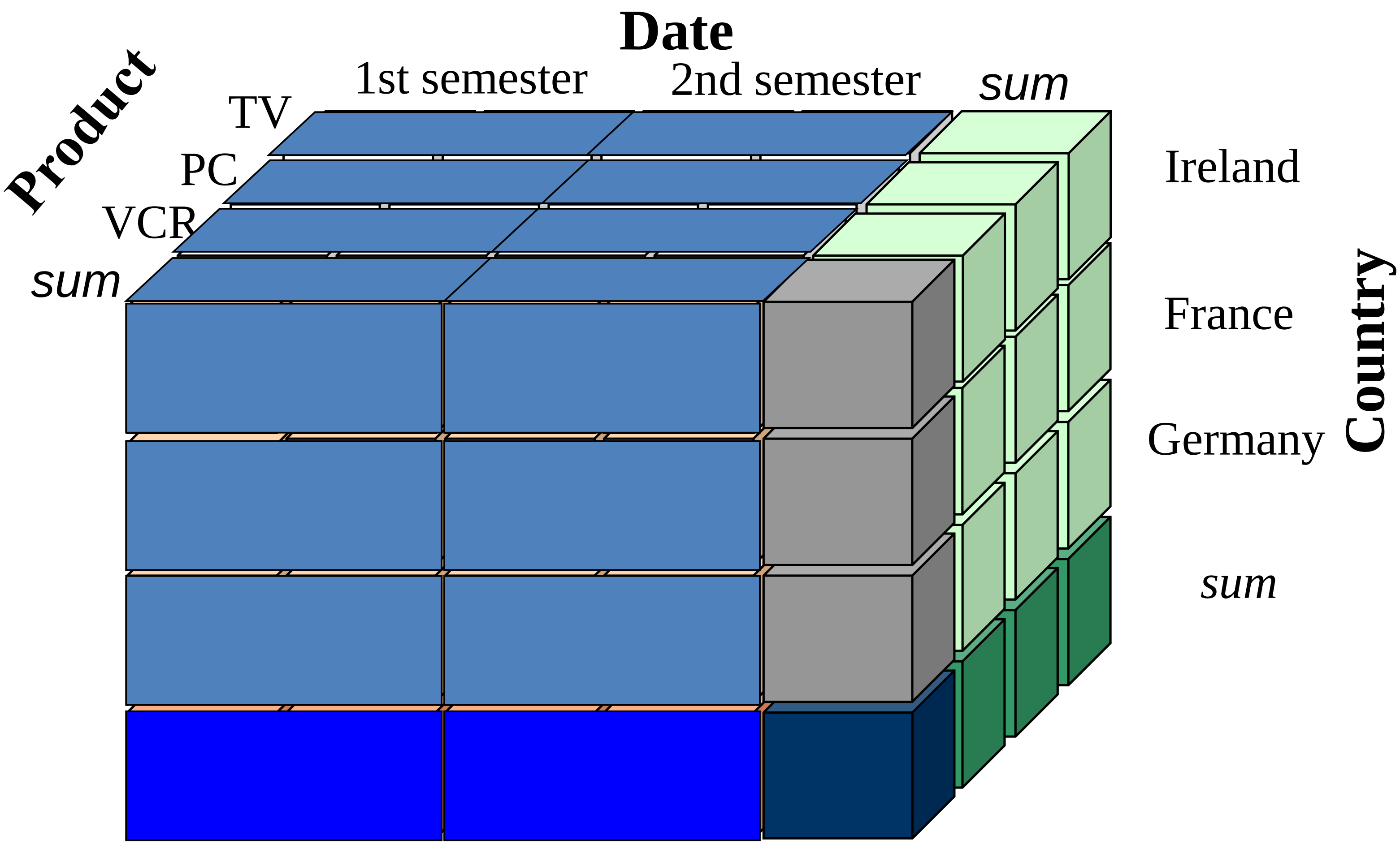


Roll-Up



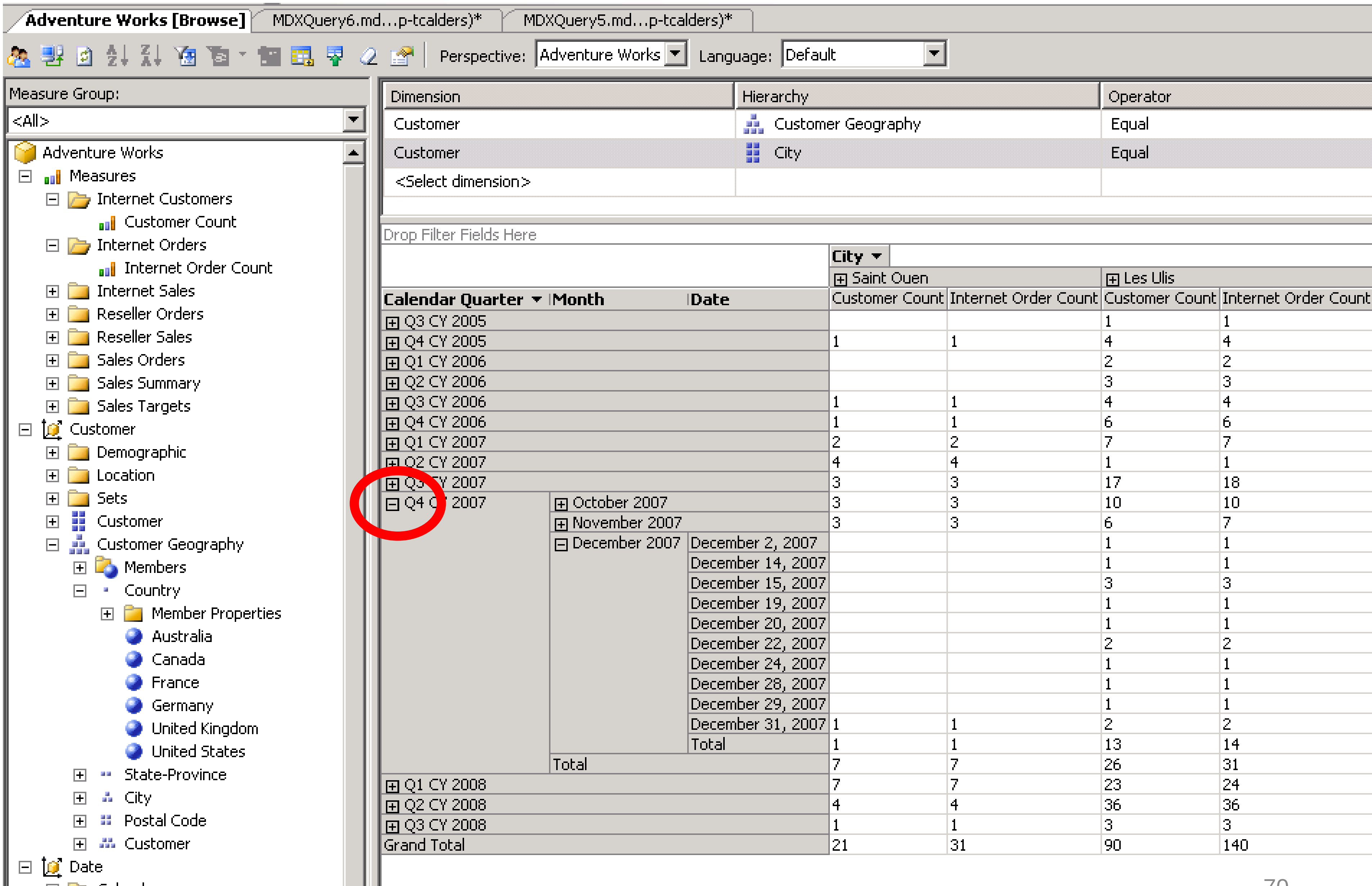
Date
Product
1Qtr
2Qtr
3Qtr
4Qtr
sum
TV
PC
VCR
sum
Ireland
France
Germany
sum
Country

Roll-Up



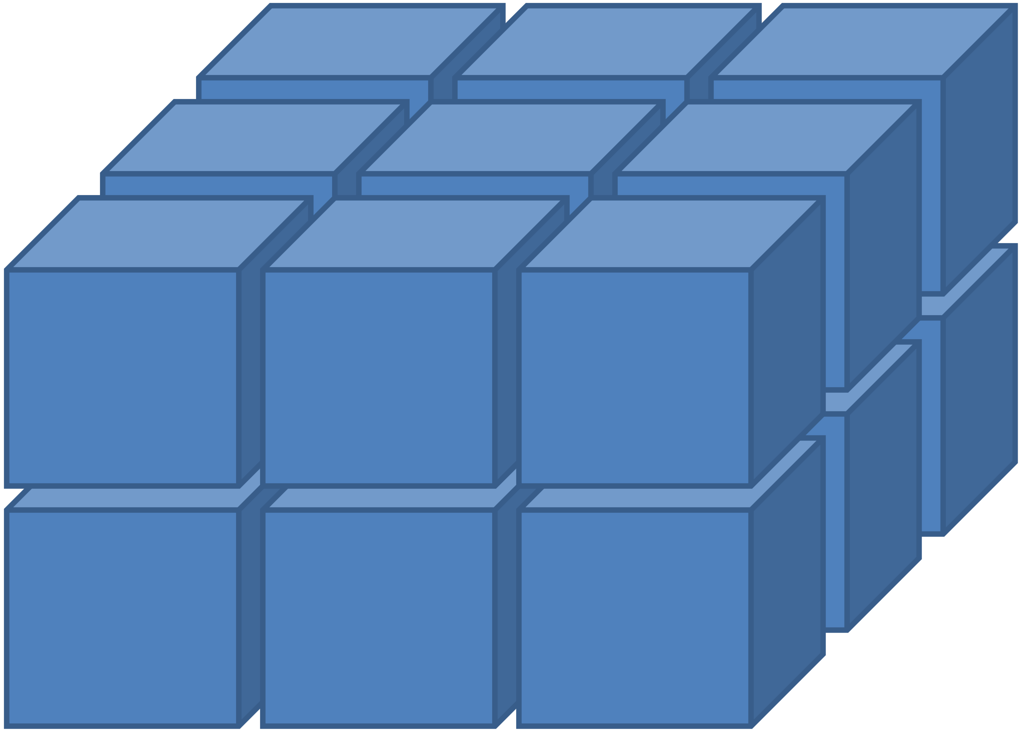
Date
1st semester
2nd semester
sum
Product
TV
PC
VCR
sum
Ireland
France
Germany
sum
Country

Browsing a Cube



Dicing

- Roll-up on multiple dimensions at once



Outline

Online Analytical Processing

- Conceptual model: Data cubes
- Query languages for supporting OLAP
  - Typical data cube operations
  - SQL extensions
  - MDX
- Database Explosion Problem

Extended Aggregation

- SQL-92 aggregation quite limited
  - Many useful aggregates are either very hard or impossible to specify
    - Data cube
    - Complex aggregates (median, variance)
    - binary aggregates (correlation, regression curves)
    - ranking queries ("assign each student a rank based on the total marks")
- SQL:1999 adds several OLAP extensions
  - Group by cube/by rollup

Representing the Cube

Sales	Date			
Country		1st sem	2nd sem	Total
	Ireland	20	23	**43**
	France	126	138	**264**
	Germany	56	48	**104**
	Total	**202**	**209**	**411**

Representing the Cube

- Special value « null » is used:

Date	Country	Sales
1st semester	Ireland	20
1st semester	France	126
1st semester	Germany	56
1st semester	**null**	**202**
2nd semester	Ireland	23
2nd semester	France	138
2nd semester	Germany	48
2nd semester	**null**	**209**
null	Ireland	**43**
null	France	**264**
null	Germany	**104**
null	**null**	**411**

Group by Cube

- group by cube:

```
select item-name, color, size, sum(number)
from sales
group by cube(item-name, color, size)
```

Computes the union of eight different groupings of the *sales* relation:

```
{ (item-name, color, size),(item-name, color),
 (item-name, size),(color, size),(item-name),
          (color),(size), ( ) }
```

Group by Cube

- Relational representation of the date-country-sales cube can be computed as follows:

```
select semester as date, country, sum(sales)
from sales
group by cube(semester,country)
```

Instead of:

```
select semester as date, country, sum(sales)
from sales group by semester, country
UNION select null as date, country, sum(sales)
from sales group by country
UNION select semester as date, null as country,
sum(sales) from sales group by country
UNION select null as date, null as country,
sum(sales) from sales
```

Group by Rollup

- rollup construct generates union on every prefix of specified list of attributes

```
select country, province, city, sum(number)
from sales
group by rollup(country, province, city)
```

Generates union of groupings:

```
{(country, province, city), (country, province),
 (country), ( ) }
```

Useful when there is a hierarchy between items

e.g., `group by (province)` does not make sense in the presence of `group by (country, province)`

Group by Cube & Rollup

- Multiple rollups and cubes can be used in a single group by clause

```
select country, province, city,
 category, product,
 sum(number) from sales
group by rollup(country, province, city),
 rollup(category, product)
```

Generates 12 groups; all combinations of:

```
{(country, province, city), (country, province),
    (country), ( ) }
```

and

```
{(category, product), (category),()}
```

Outline

Online Analytical Processing

- Conceptual model: Data cubes
- Query languages for supporting OLAP
  - Typical data cube operations
  - SQL extensions
  - MDX
- Database Explosion Problem

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, generates Pivot Tables

```
SELECT { [Measures].[Store Sales] } ON COLUMNS,
  { [Date].[2002], [Date].[2003] } ON ROWS
FROM Sales
WHERE ( [Store].[USA].[CA] )
```

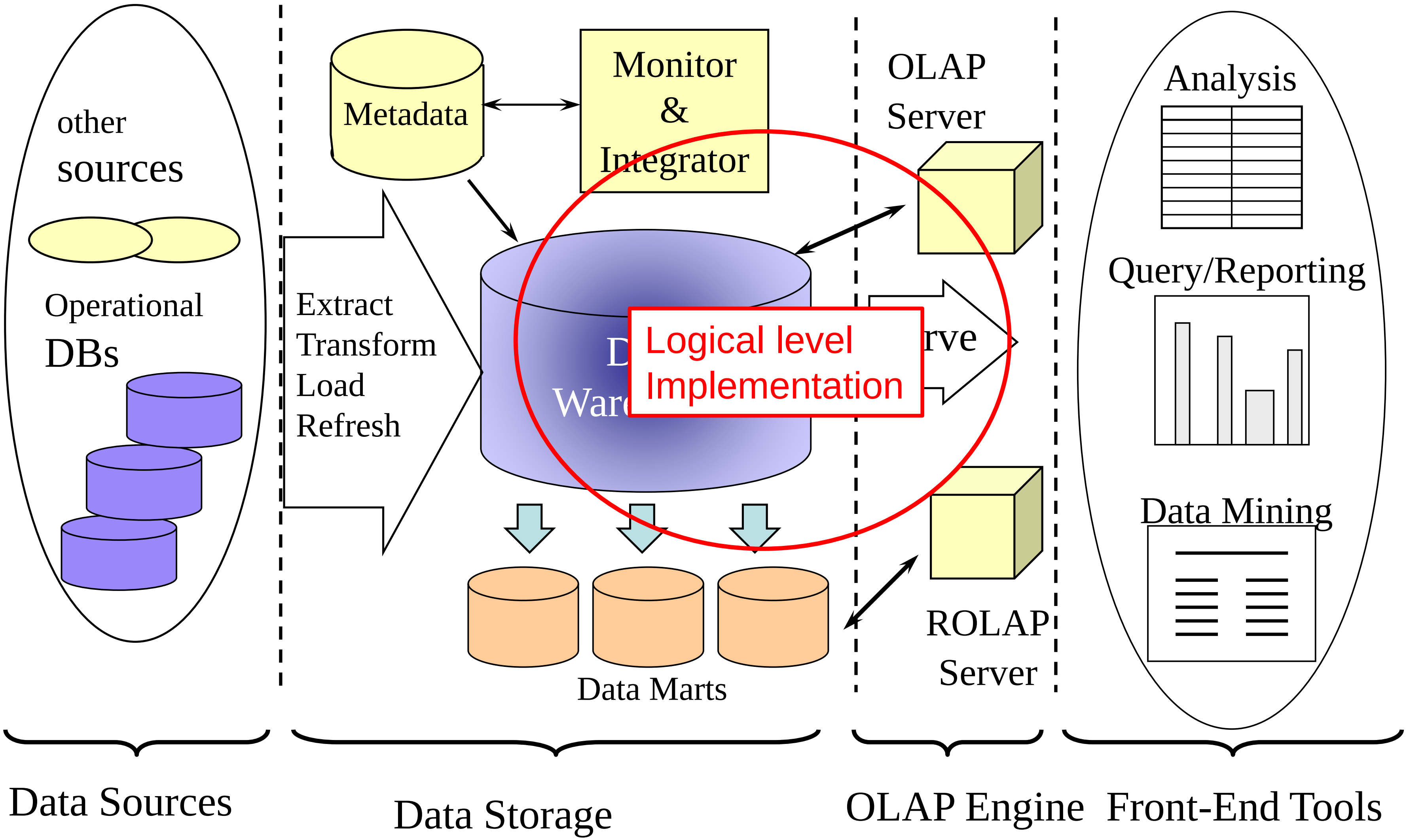
```
SELECT { continent.[Europe], continent.[Asia]} ON Axis(0),
  { Product.[Computers], Product.[Printers]} ON Axis(1),
  { Years.[1996], Years.[1997] } ON Axis(2)
FROM Sales
```

Outline

Online Analytical Processing

- Data Warehouses
- Conceptual model: Data cubes
- Query languages for supporting OLAP
  - SQL extensions
  - MDX
- Database Explosion Problem

Three-Tier Architecture



Implementation

- To make query answering more efficient: consolidate (materialize) all aggregations
- Early implementations used a multidimensional array.
  - Fast lookup: cell(prod. p, date d, prom. pr):
    - look up index of p1, index of d, index of pr:
      index = (p x D x PR) + (d x PR) + pr
- Obvious problem: sparse data
  - easy to solve, though;
    - binary search tree, hash table, …

- Nevertheless: very quickly people were confronted with the *Data Explosion Problem*

Data Explosion Problem

- Why?
  - n dimensions, every dimension has d values
    - d^n possible tuples.
  - Number of cells in the cube: $(d+1)^n$
    - Only a factor d increase
- However, most data is not dense, but *sparse*
  - not all d^n tuples are there in the source data.

Example: 10 dimensions with 10 values

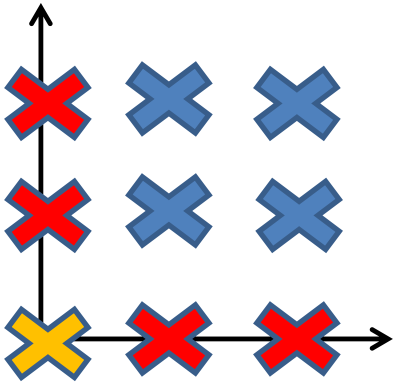
10 000 000 000 possibilities

One tuple increases the count of 2^{10} cells

How many for N tuples?

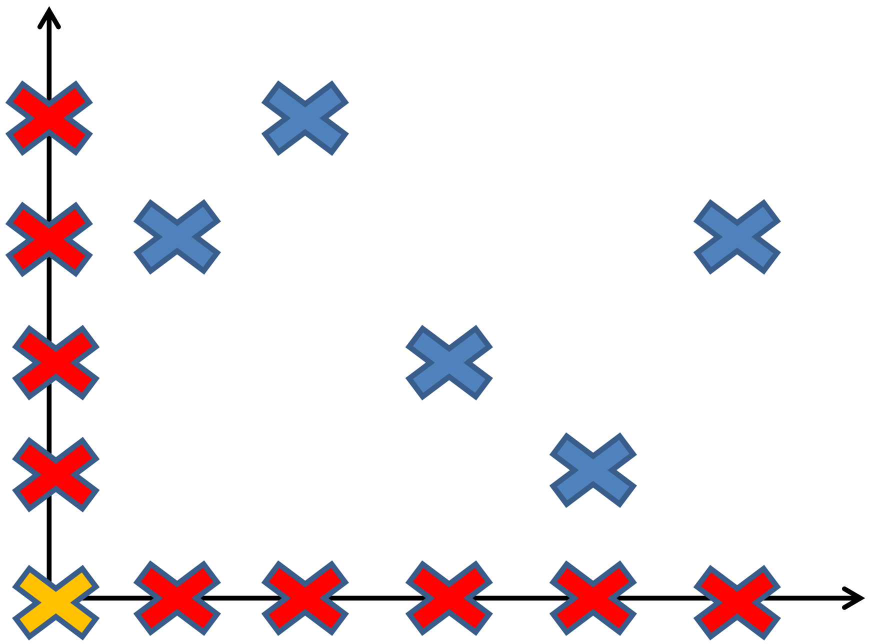
Dense Cube

Country	Brand	Sales
FR	A	123
FR	B	456
BE	A	678
BE	B	254



Explosion Problem: Sparsity

Country	Brand	Sales
FR	A	123
NL	B	456
BE	C	678
US	D	254
US	E	134

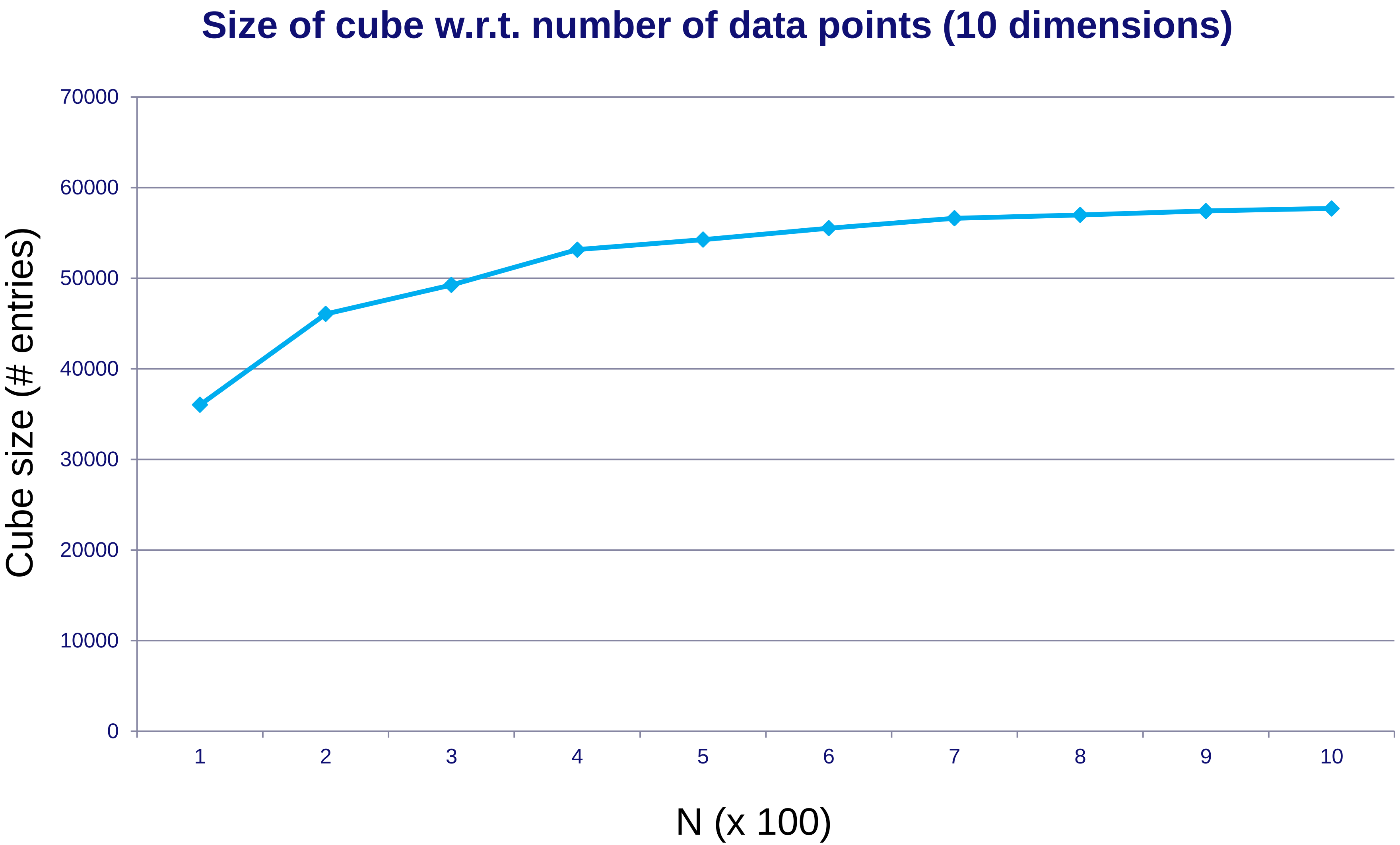


Data Explosion Problem

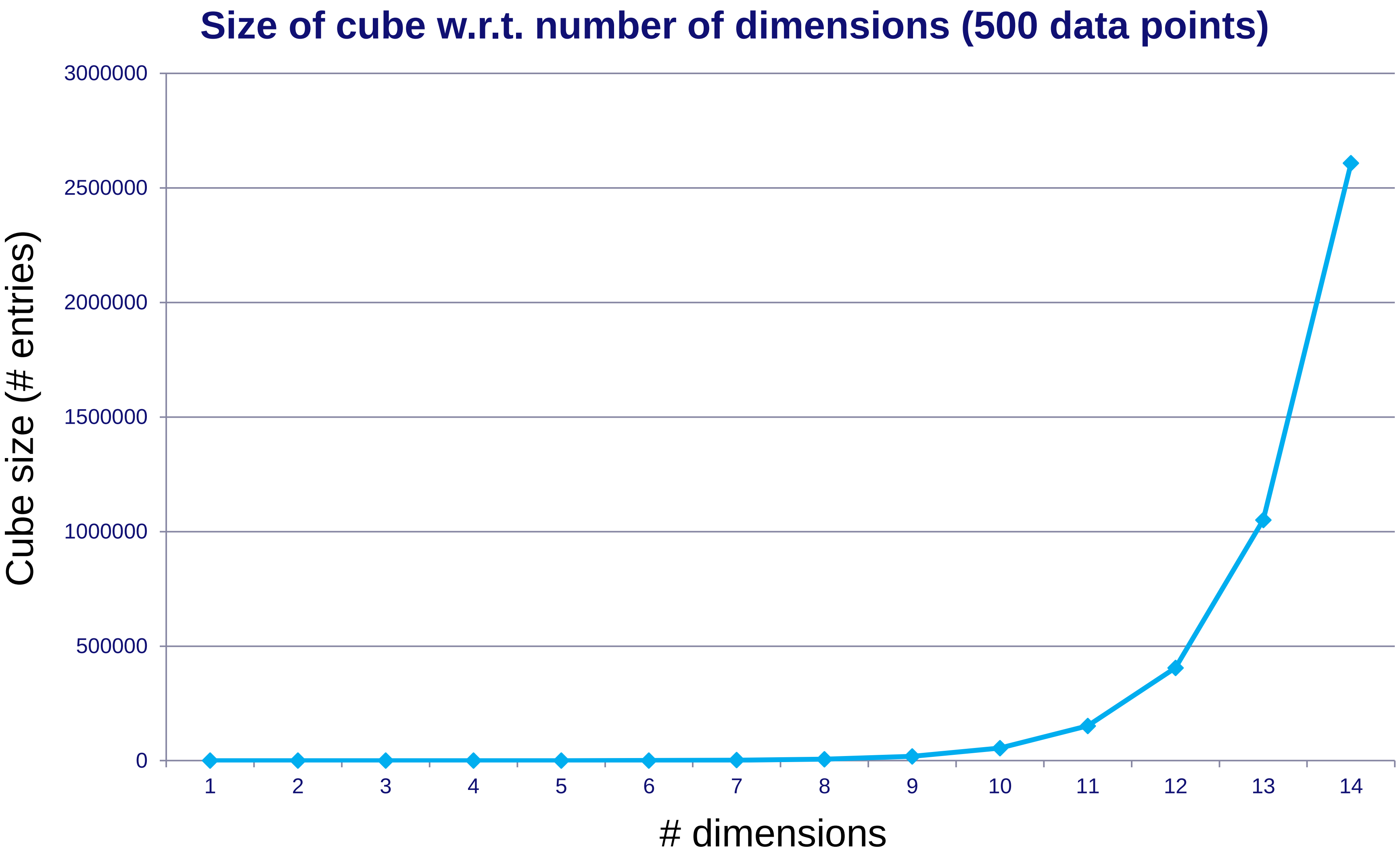
- Suppose:
  - m dimensions
  - n data points
  - dimensions are i.i.d.
  - all values drawn uniformly from { 0, 1 }

- Under these settings we will analyze how the size of the cube grows with the number of dimensions

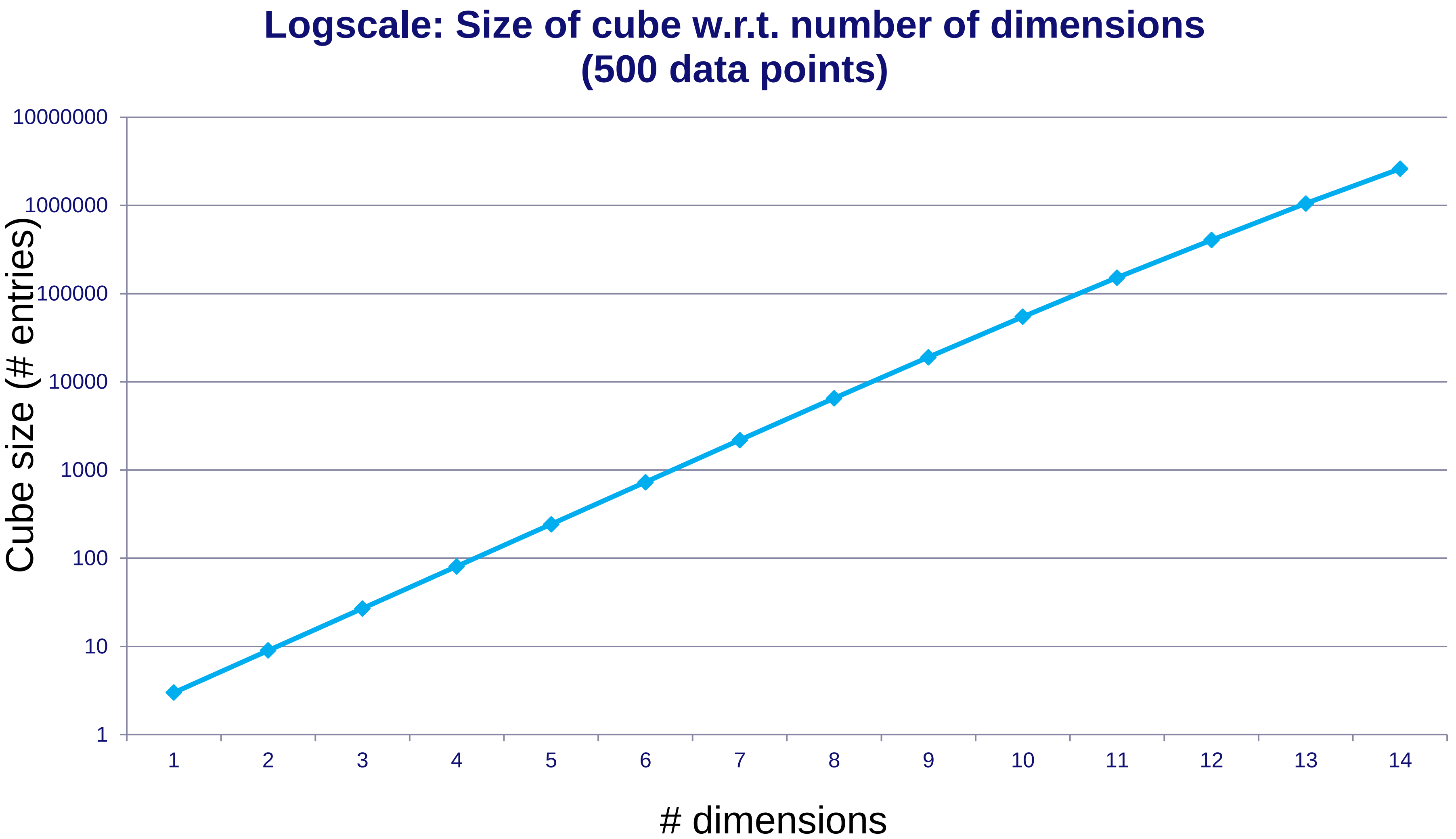
Data Explosion Problem



Data Explosion Problem



Data Explosion Problem



Summary

- Datawarehouses supporting OLAP for *decision support*
- Data Cubes as a *conceptual* model
  - Measurement, dimensions, hierarchy, aggregation
- Queries
  - Roll-up, Drill-down, Slice and dice, pivoting…
  - SQL:1999 extensions for supporting OLAP
- Straightforward implementation is problematic