Data Warehousing and OLAP

Toon Calders
t.calders@tue.nl

TU/e Technische Universiteit Eindhoven University of Technology

Motivation

- « Traditional » relational databases are geared towards online transaction processing:
  - bank terminal
  - flight reservations
  - student administration

- Decision support systems have different requirements

Transaction Processing

Transaction processing

- Operational setting
- Up-to-date = critical
- Simple data
- Simple queries; only « touch » a small part of the database

Flight reservations

- ticket sales
- do not sell a seat twice
- reservation, date, name
- Give flight details of X, List flights to Y

Transaction Processing

- Database must support
  - simple data
    - tables
  - simple queries
    - select from where …
  - consistency & integrity CRITICAL
  - concurrency
- Relational databases, Object-Oriented, Object-Relational

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

PART I Concepts

Outline

Online Analytical Processing

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

Technische Universiteit
Eindhoven
University of Technology

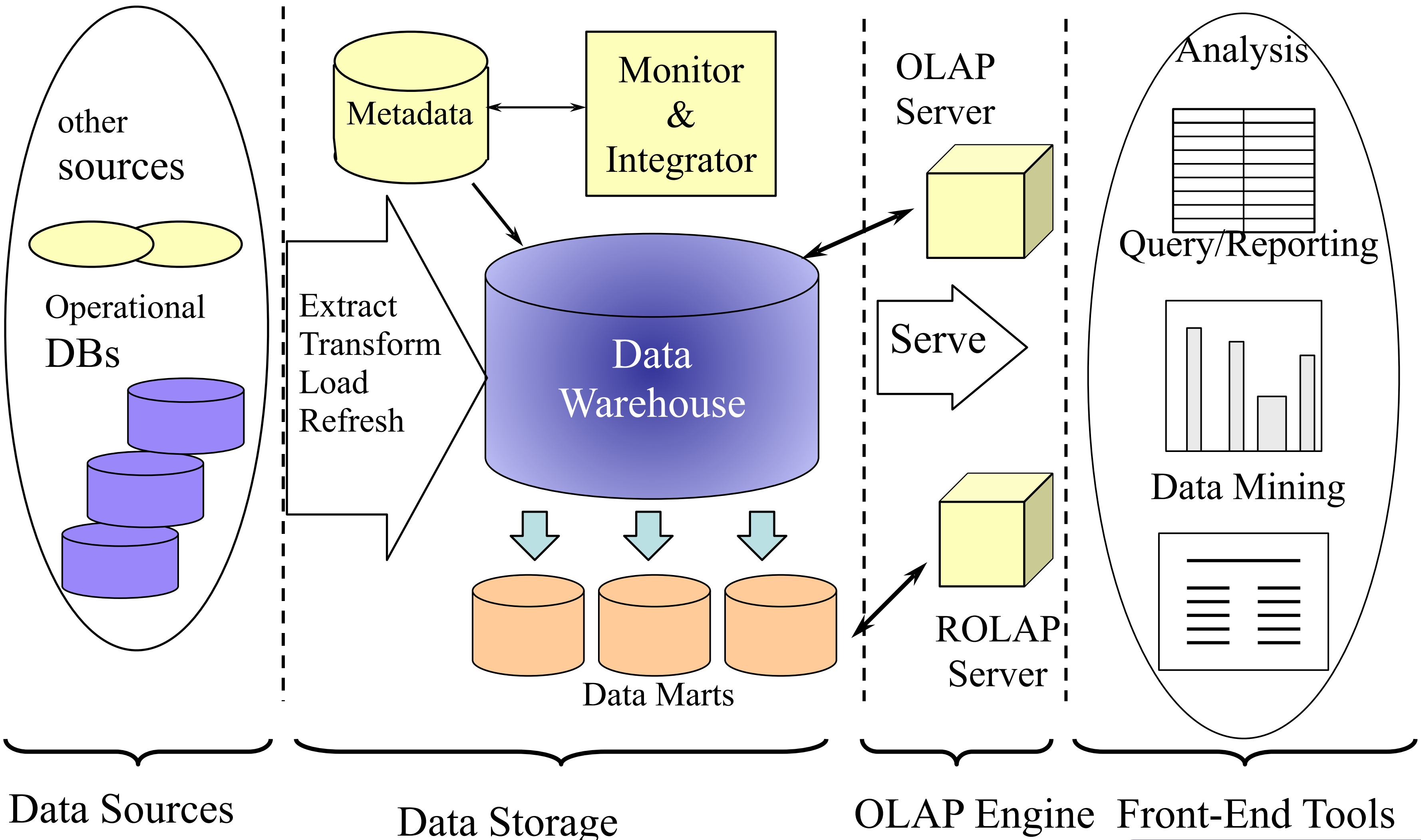
Data Warehouse

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

Three-Tier Architecture



Three-Tier Architecture

- Extract-Transform-Load
  - Get data from different sources; data integration
  - Cleaning the data
  - Takes significant part of the effort (up to 80%!)
- Refresh
  - Keep the data warehouse up to date when source data changes

Three-Tier Architecture

- Data storage
  - Optimized for OLAP
  - Specialized data structures
  - Specialized indexing structures

- Data marts
  - common term to refer to “less ambitious data warehouses”
  - Task oriented, departmental level

OLAP

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

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

Outline

Online Analytical Processing

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

Examples of Queries

- Flight company: evaluate ticket sales
  - give total, average, minimal, maximal amount
  - per date: week, month, year
  - by destination/source port/country/continent
  - by ticket type
  - by # of connections
  - …

Common Characteristics

- One (or few) special attribute(s): amount
  → measure
- Other attributes: select relevant *regions*
  → dimensions
- Different levels of generality (month, year)
  → hierarchies
- Measurement data is summarized: sum, min, max, average
  → aggregations

Supermarket Example

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

Dim.

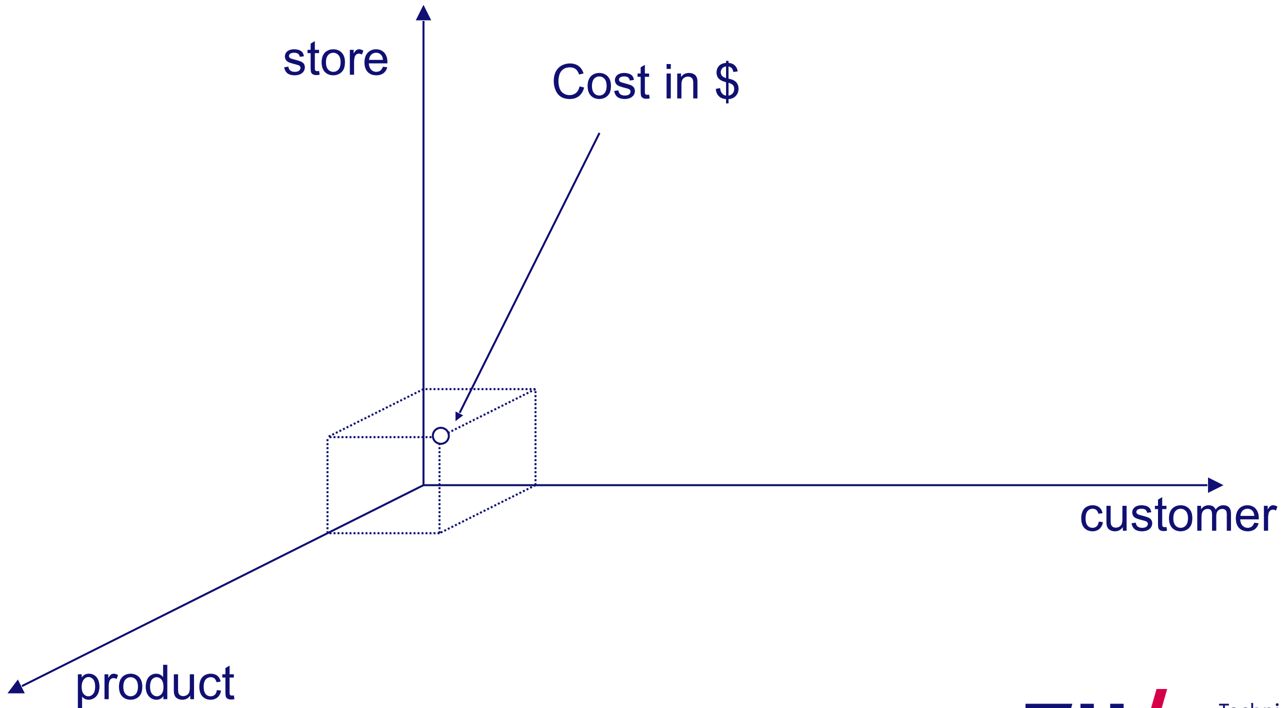
measure

hierarchies

- **What are the measure and dimensional attributes, where are the hierarchies?**

Why Dimensions?

- Multidimensional view on the 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

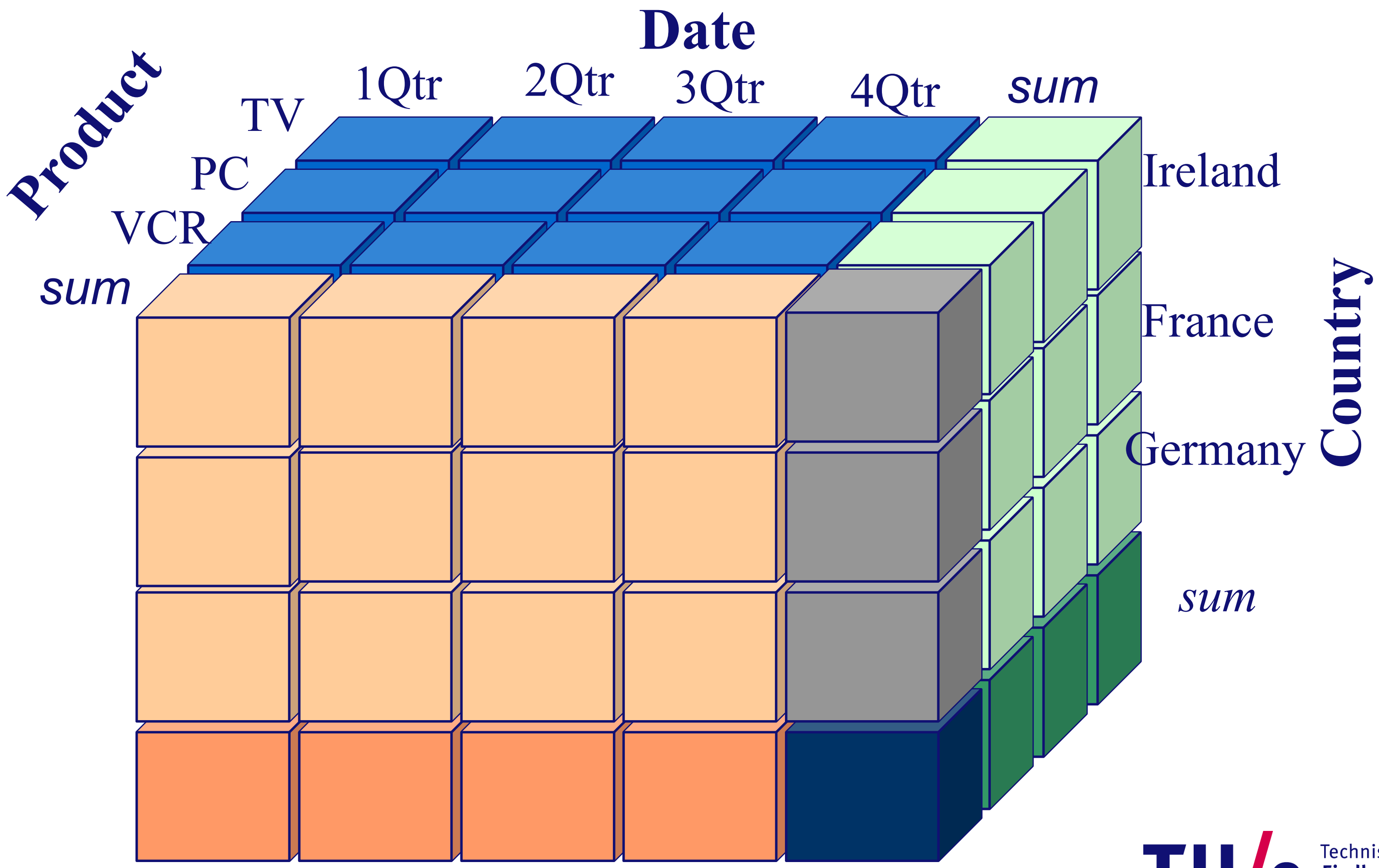
Data Points/
1st level of aggregation

Aggregated
w.r.t. X-dim

Aggregated
w.r.t. Y-dim

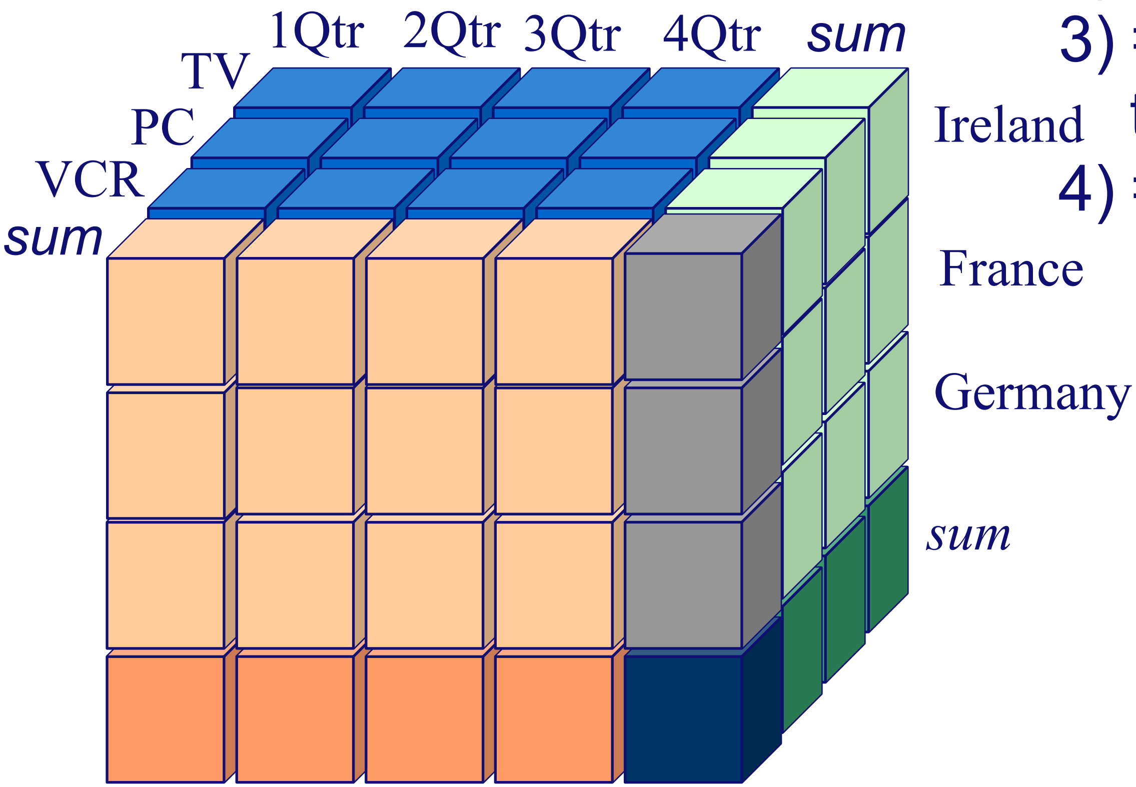
Aggregated
w.r.t. X and Y

Data Cubes



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

Data Cubes

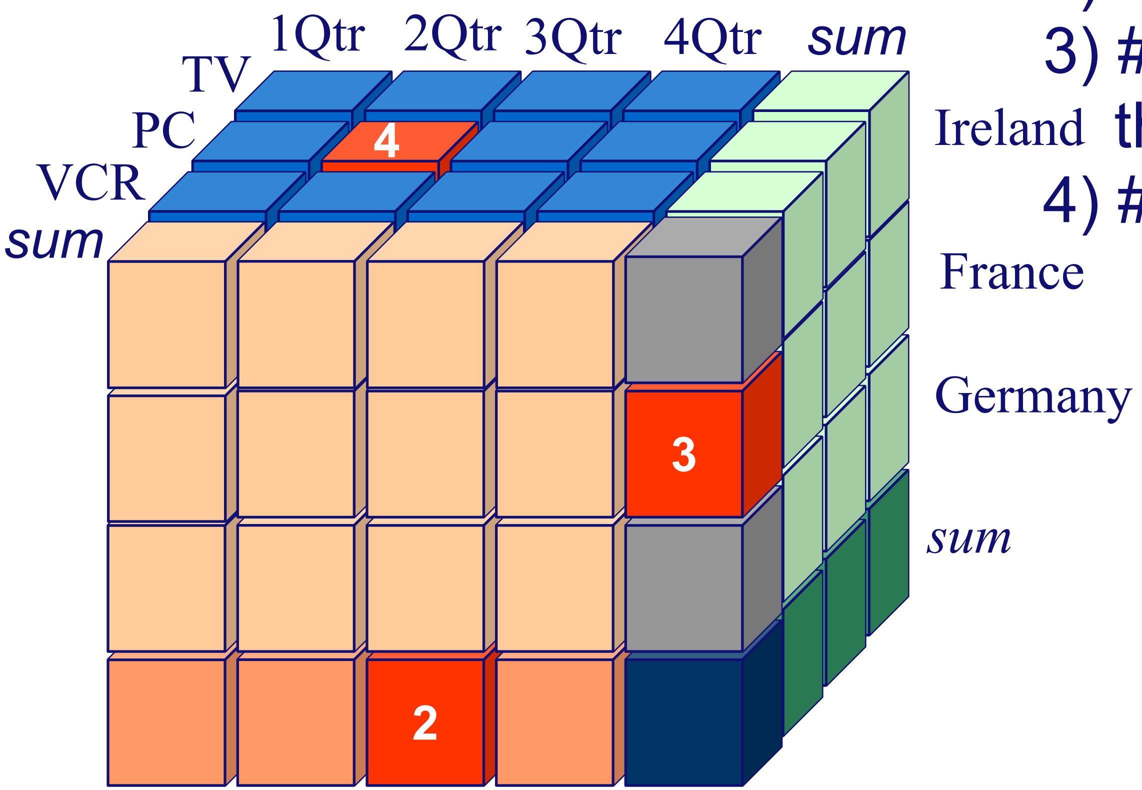


1) #TV's sold in the 2Qtr?
2) Total sales in 3Qtr?
3) #products sold in France this year
4) #PC's in Ireland in 2Qtr?

Data Cubes

In the back, at the bottom, second column

1) #TV's sold in the 2Qtr?
2) Total sales in 3Qtr?
3) #products sold in France this year
4) #PC's in Ireland in 2Qtr?



Outline

Online Analytical Processing

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

Operations with Data Cubes

- **What operations can you think of that an analyst might find useful? (e.g., store)**

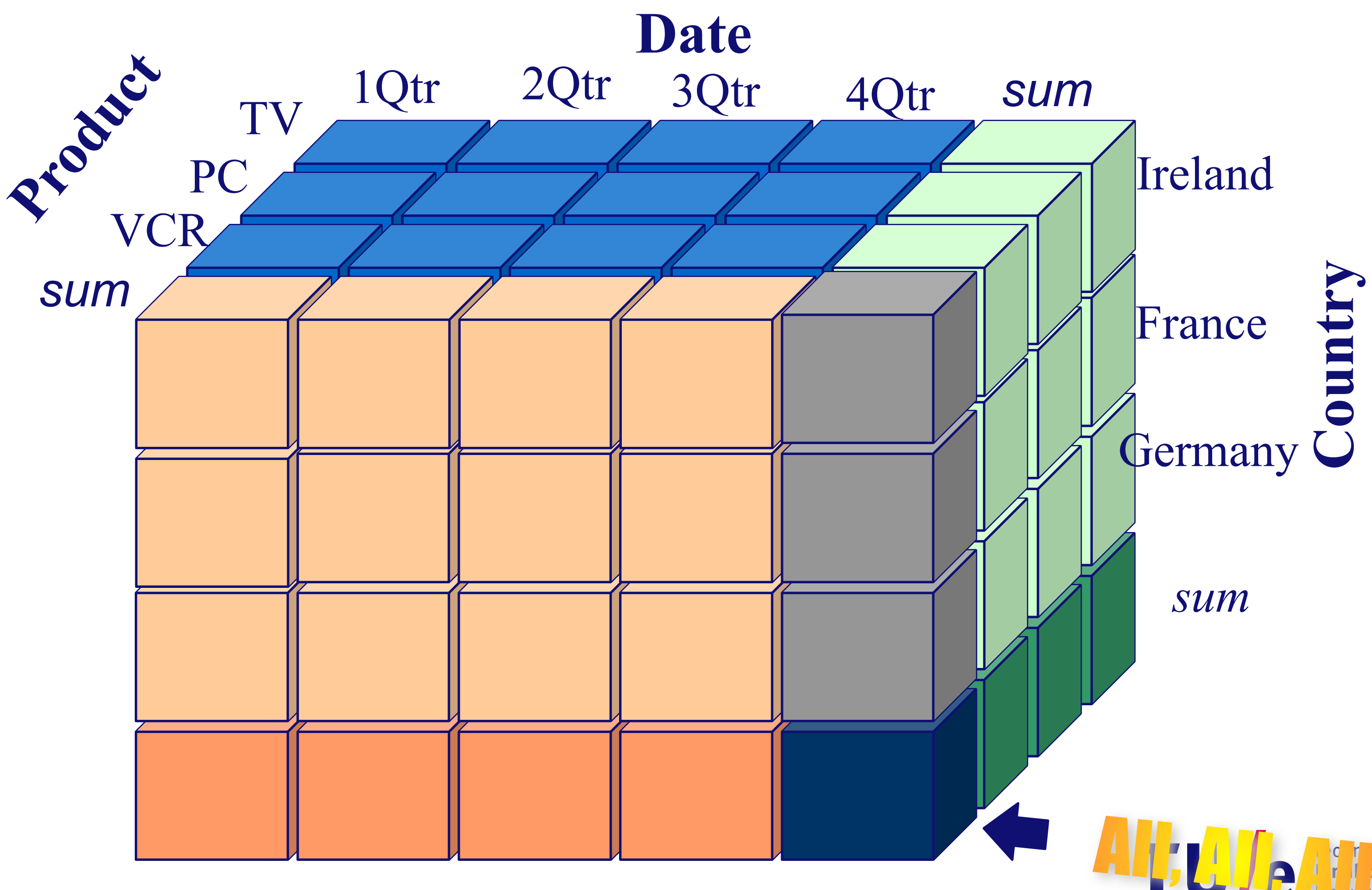
Operations with Data Cubes

- What operations can you think of that an analyst might find useful? (e.g., store)
  - only look at stores in the Netherlands
  - look at cities instead of individual stores
  - look at the cross-table for product-date
  - restrict analysis to 2006, product O1
  - go back to a finer granularity at the store level

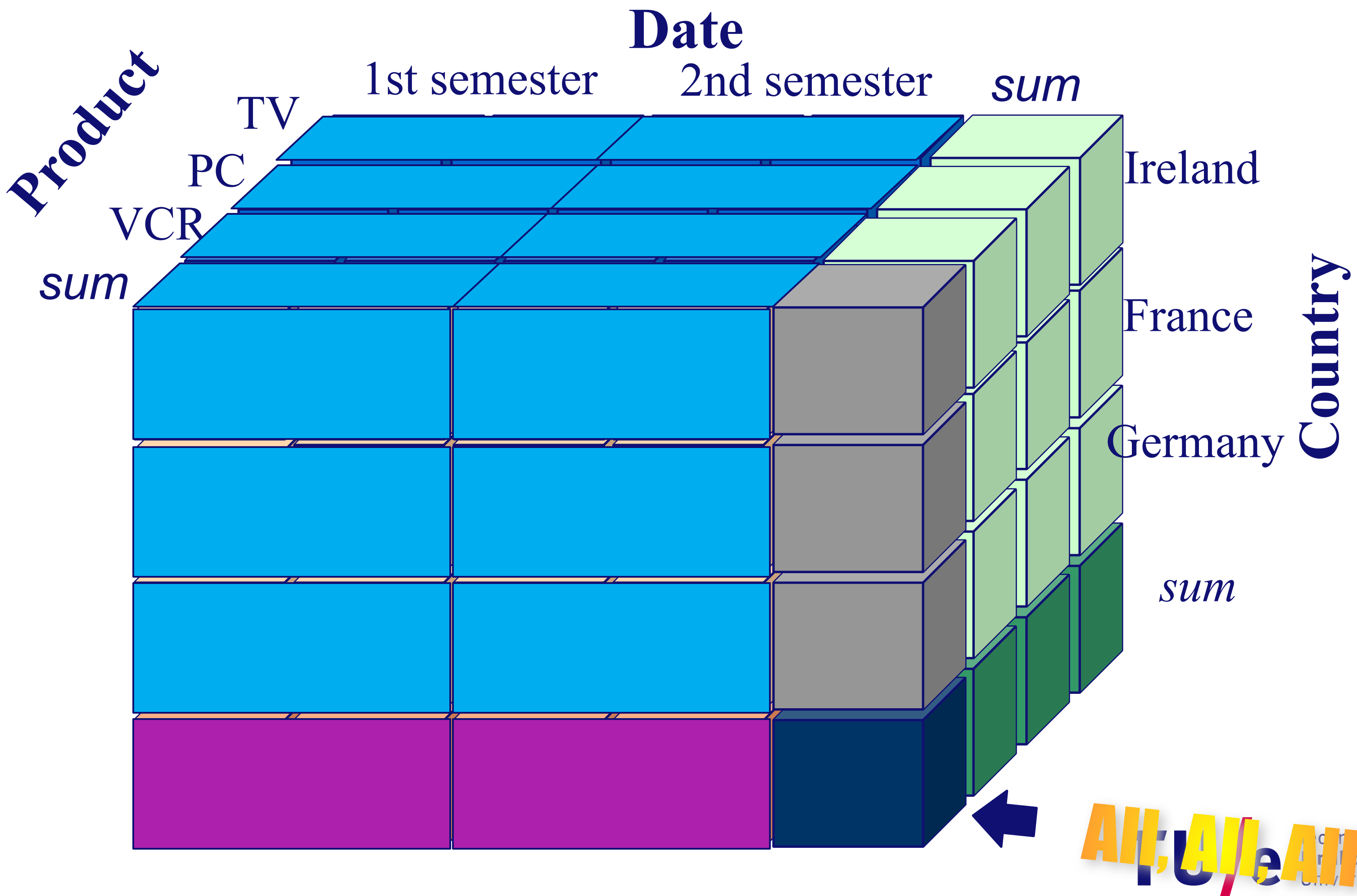
Roll-Up

- **Move in one dimension from a lower granularity to a higher one**
  - **store → city**
  - **cities → country**
  - **product → product type**
  - **Quarter → Semester**

Roll-Up


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

Roll-Up



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

Drill-down

- Inverse operation
- Move in one dimension from a higher granularity to a lower one
  - city → store
  - country → cities
  - product type → product

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

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

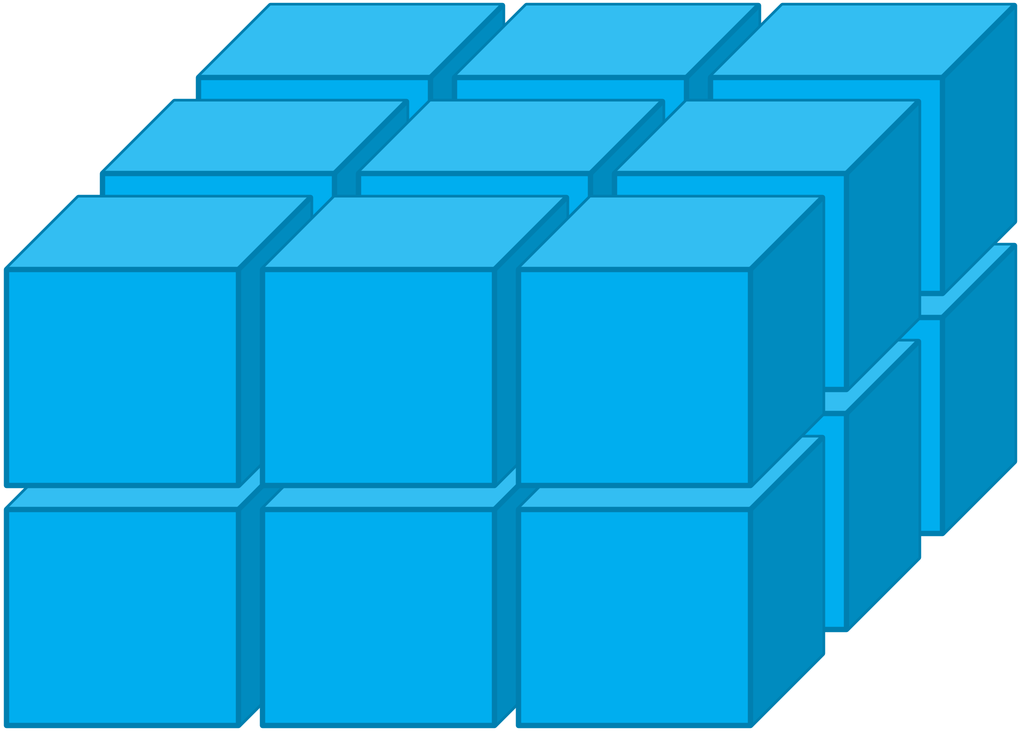
Pivoting

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

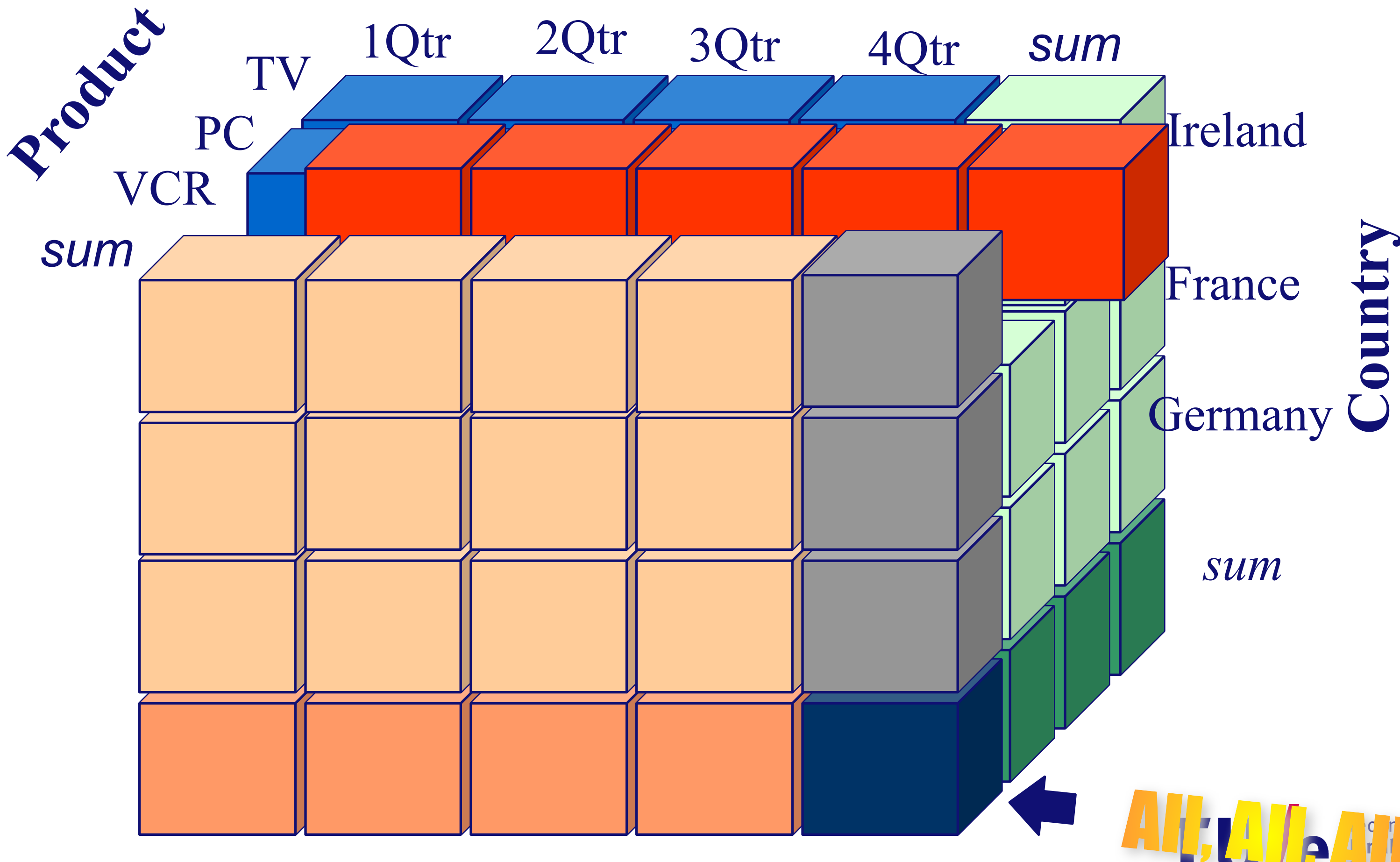
Sales	Date			
		1st sem	2nd sem	Total
Country	Ireland	20	23	**43**
	France	126	138	**264**
	Germany	56	48	**104**
	Total	**202**	**209**	**411**

Slice & dice

- **Roll-up on multiple dimensions at once**

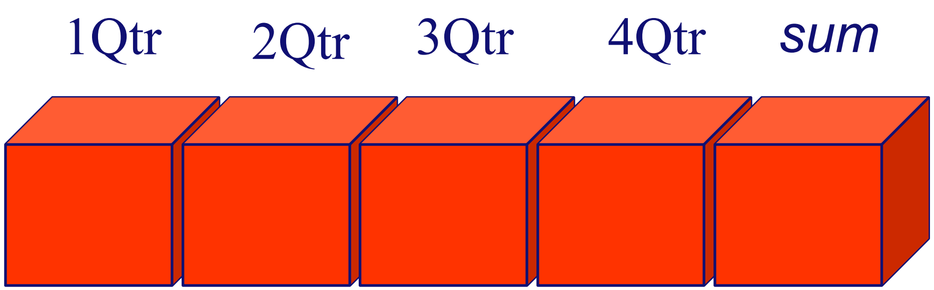


Select


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

Select

- **Select a part of the cube by restricting one or more dimensions**
  - **restrict analysis to Ireland and VCR**



Outline

Online Analytical Processing

- Data Warehouses
- Conceptual model: Data cubes
- Query languages for supporting OLAP
  - 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 OLAP extensions

Representing the Cube

- **Special value « null » is used:**

Sales	Date			
		1st sem	2nd sem	Total
Country	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*)

  - **grouping() and decode() can be applied to replace "null" by other constant:**
    - **decode(grouping(semester), 1, 'all', *semester*)**

Group by Rollup

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

  select *item-name*, *color*, *size*, **sum**(*number*)
  from *sales*
  group by rollup(*item-name, color, size*)

  Generates union of four groupings:

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

Group by Rollup

- Rollup can be used to generate aggregates at multiple levels.
- E.g., suppose *itemcategory*(*item-name, category*) gives category of each item.

  select *category, item-name*, sum(*number*)
  from *sales, itemcategory*
  where *sales.item-name* = *itemcategory.item-name*
  group by rollup(*category, item-name*)

  gives a hierarchical summary by *item-name* and by *category.*

Group by Cube & Rollup

- **Multiple rollups and cubes can be used in a single group by clause**
  - **Each generates set of group by lists, cross product of sets gives overall set of group by lists**

Example

select ***item-name, color, size*****, sum(*****number*****)**
from ***sales***
group by rollup(***item-name*****), rollup(*****color, size*****)**

generates the groupings

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

=

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

Technische Universiteit Eindhoven University of Technology

MDX

- Multidimensional Expressions (MDX) is a query language for cubes
  - Supported by many data warehouses
  - Input and output are cubes

SELECT { [Measures].[Store Sales] } ON COLUMNS, { [Date].[2002], [Date].[2003] } ON ROWS FROM Sales

WHERE ([Store].[USA].[CA])

II.2 Data storage and indexing

- How is the data stored?
  - relational database (ROLAP)
  - Specialized structures (MOLAP)

- How can we speed up computation?
  - Indexing structures
    - bitmap index
    - join index

Implementation

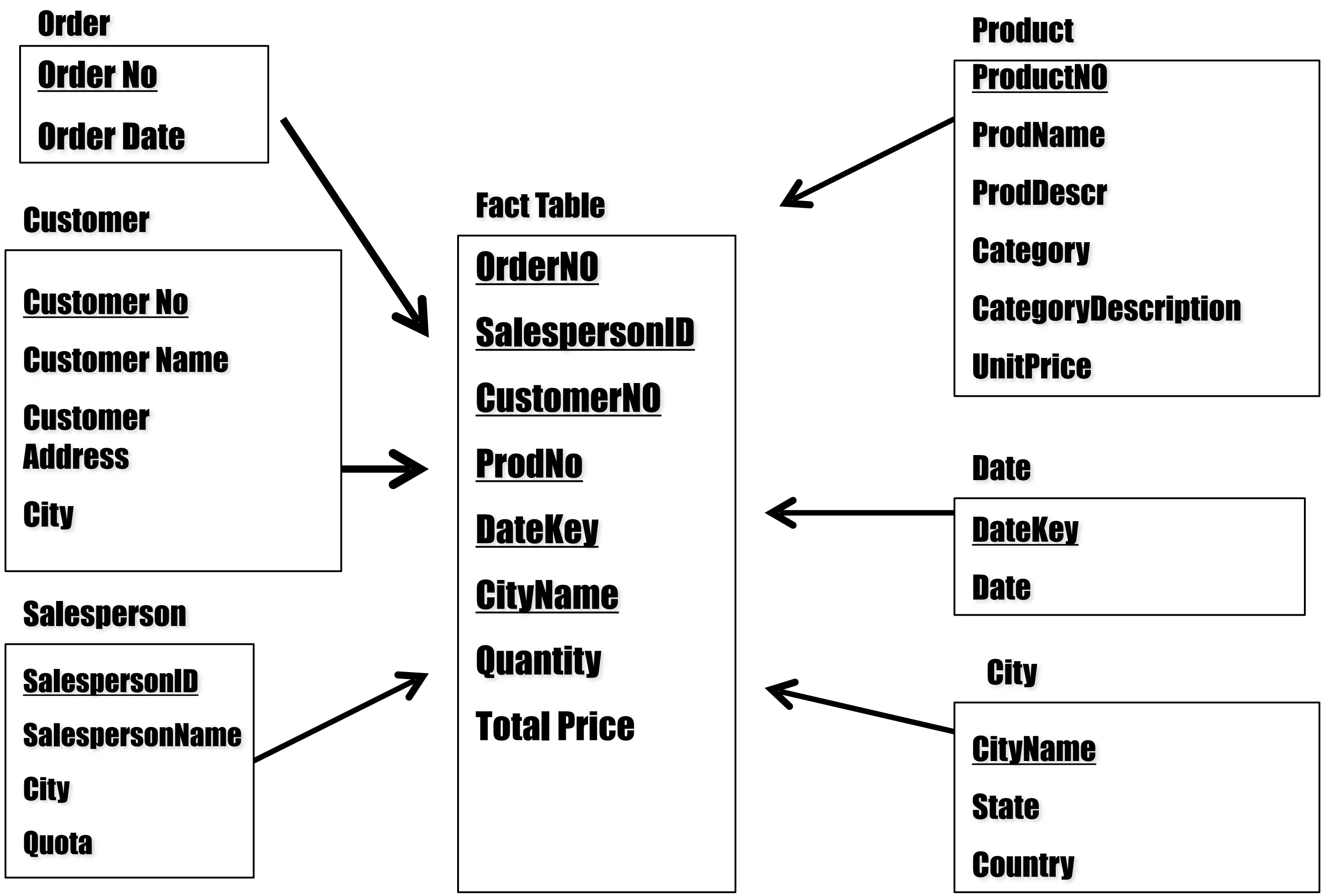
Nowadays systems can be divided in three categories:

- **ROLAP (Relational OLAP)**
  - **OLAP supported on top of a relational database**
- **MOLAP (Multi-Dimensional OLAP)**
  - **Use of special multi-dimensional data structures**
- **HOLAP: (Hybrid)**
  - **combination of previous two**

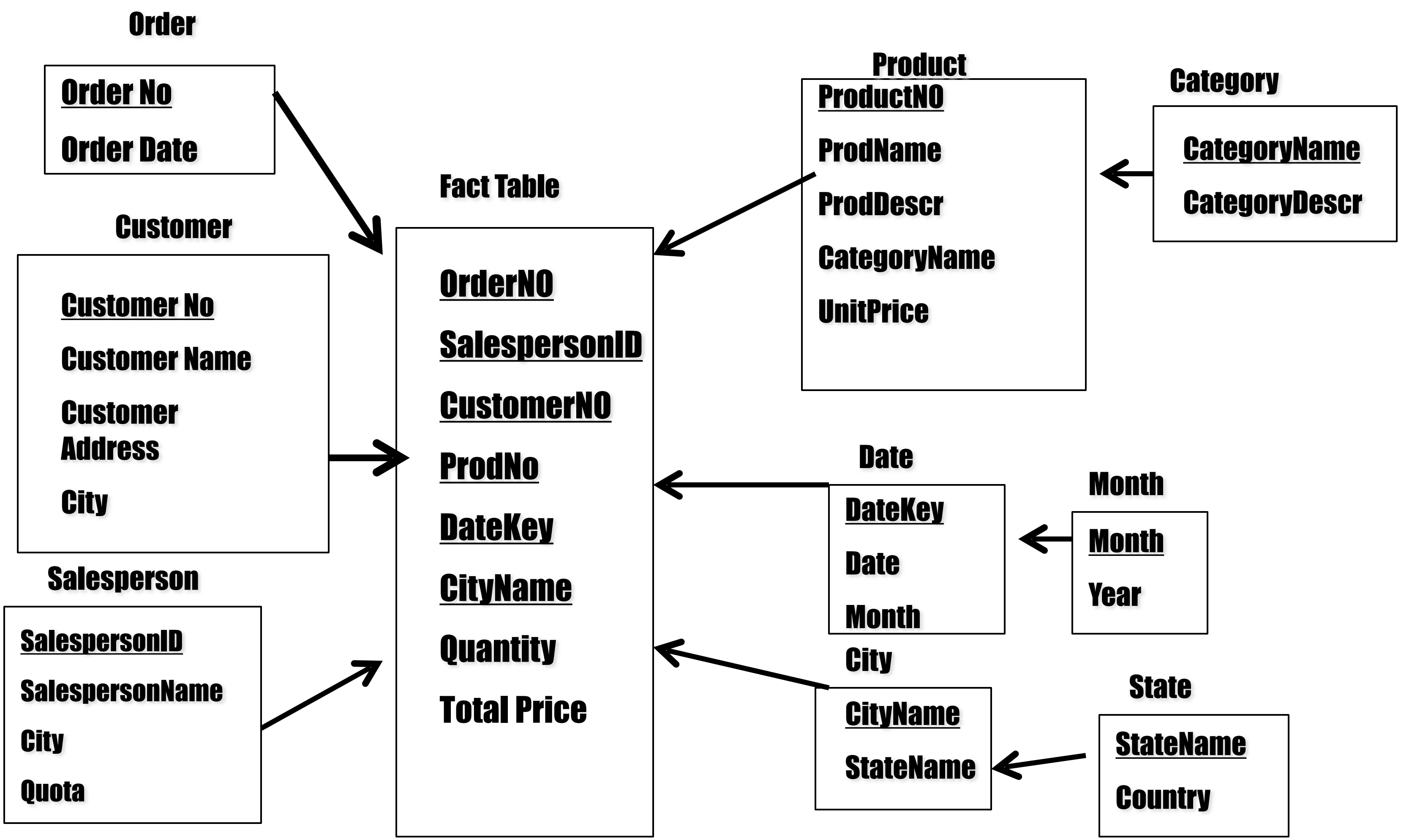
ROLAP

- Typical database scheme:
  - star schema
    - fact table is central
    - links to dimensional tables
  - Extensions:
    - snowflake schema
      - dimensions have hierarchy/extra information attached
    - Star constellation
      - multiple star schemas sharing dimensions

Example of a Star Schema

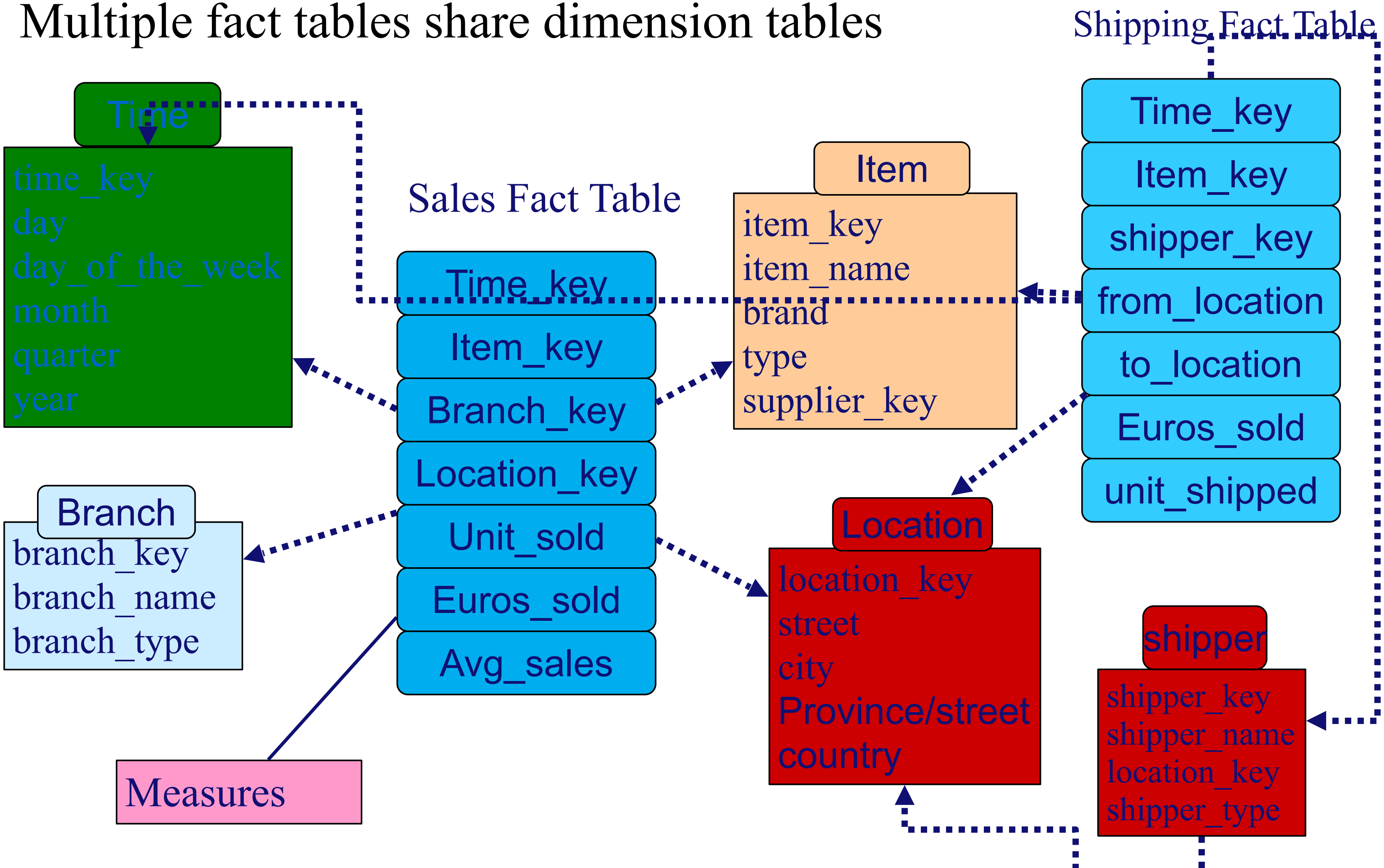


Example of a Snowflake Schema



Example of Fact Constellation

Multiple fact tables share dimension tables



This Lecture

- How is the data stored?
  - Relational database (ROLAP)
  - Specialized structures (MOLAP)

- How can we speed up computation?
  - Indexing structures
    - bitmap index
    - join index

MOLAP

- Not on top of relational database
  - most popular design
  - specialized data structures
    - Multicubes vs Hypercubes
  - Not all subcubes are materialized