Chapter 4. Data Warehousing and On-line Analytical Processing (OLAP)

Meng Jiang

CS412 Summer 2017:

Introduction to Data Mining

Data Warehouse

- **Basic Concepts**
- Modeling: Data Cube and OLAP
- Design and Usage
- Implementation

Data Warehouse

- Defined in many different ways, but not rigorously
  - A decision support database that is maintained separately from the organization's operational database

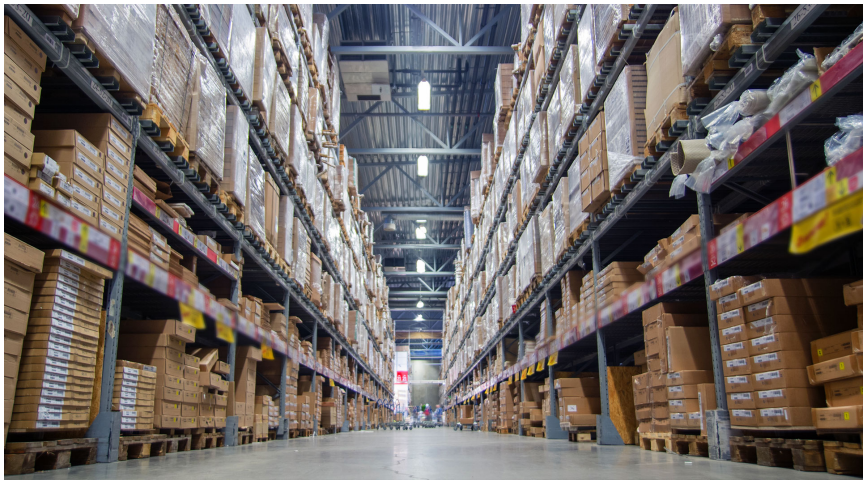
Operational Databases

(Data) Marts

(Data) Warehouse

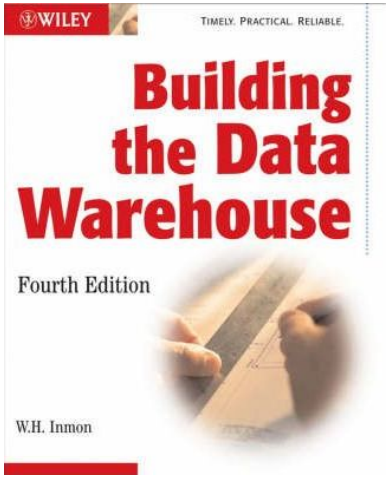
Data Warehouse

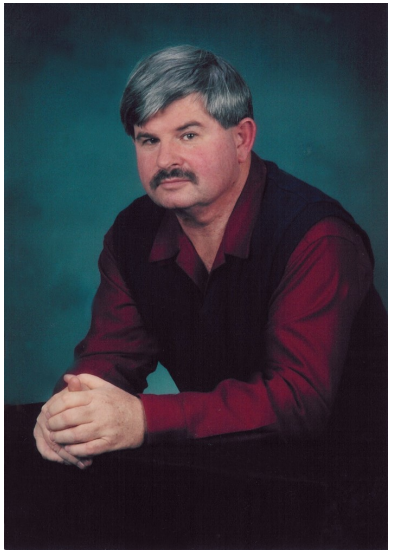
- Defined in many different ways, but not rigorously
  - A decision support database that is maintained separately from the organization's operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis



Data Warehouse

- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."—William H. (Bill) Inmon





- Data warehousing:
  - The process of constructing and using data warehouses

(1) Subject-Oriented

- Organized around major subjects, such as customer, product, sales
- Focusing on the modeling and analysis of data for decision makers, NOT on daily operations or transaction processing
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process

(2) Integrated

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied
  - Ensure **consistency** in naming conventions, encoding structures, attribute measures, etc. among different data sources
    - Ex. Hotel price: differences on currency, tax, breakfast covered, and parking

(3) Time-Variant

- The time horizon for the data warehouse is significantly **longer** than that of operational systems
  - Operational database: current value data
  - Data warehouse data: provide information from a **historical** perspective (e.g., past 5-10 years)
- **Every key** structure in the data warehouse
  - Contains an element of **time**, explicitly or implicitly
  - But the key of operational data may or may not contain "time element"

(4) Nonvolatile

- Independence
  - A physically separate store of data transformed from the operational environment
- Static: Operational update of data does NOT occur in the data warehouse environment
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - initial loading of data and access of data

OLTP vs OLAP

- OLTP: **Online** transactional processing
  - DBMS operations
  - Query and transactional processing
- OLAP: **Online** analytical processing
  - Data warehouse operations (drilling, slicing, dicing, etc.)
  - Data analysis to support decision making

OLTP vs OLAP

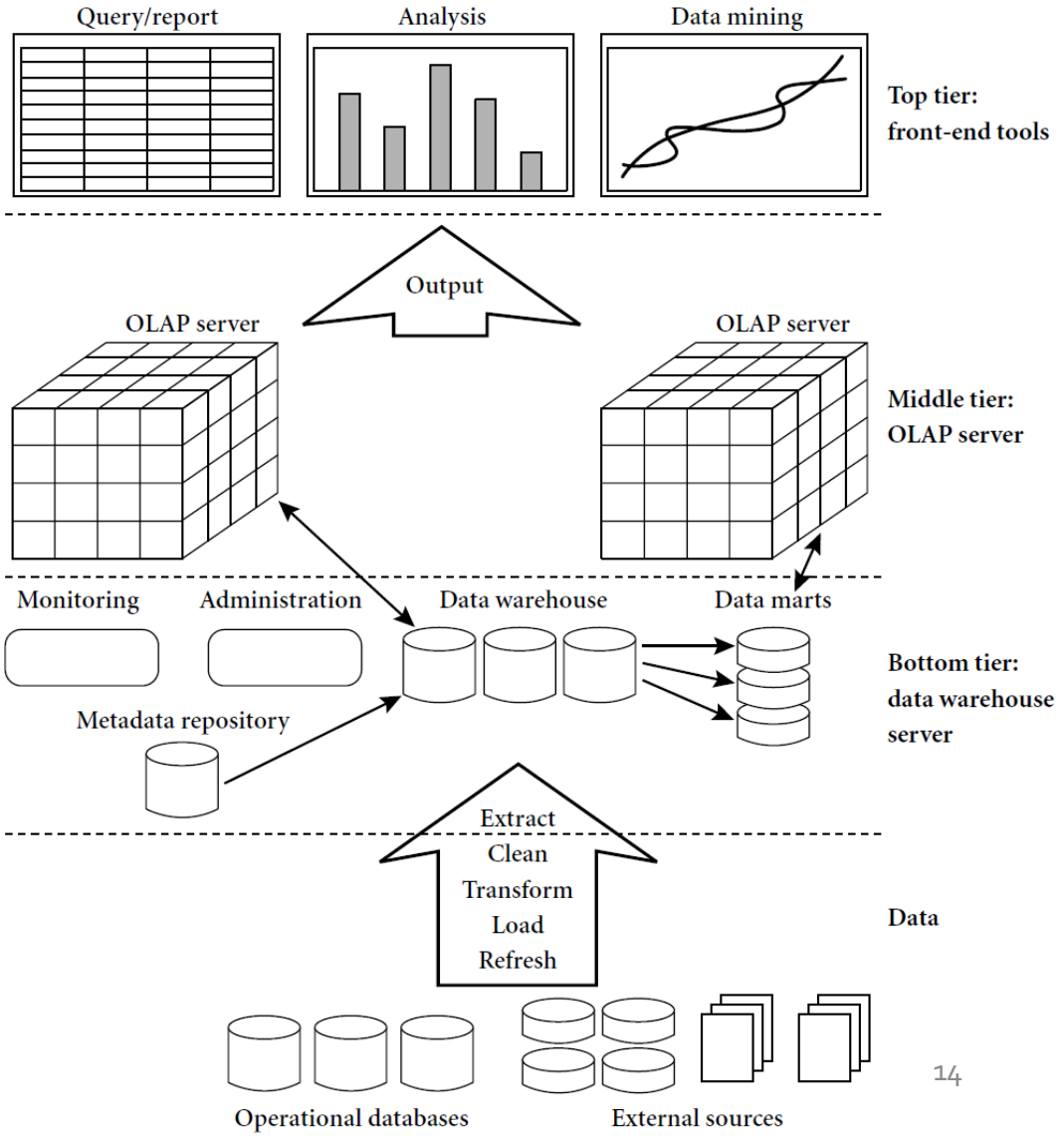
	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

Why a Separate Data Warehouse?

- High performance for both systems
  - DBMS— tuned for **OLTP**: access methods, indexing, concurrency control, recovery
  - Warehouse—tuned for **OLAP**: complex OLAP queries, multidimensional view, consolidation
- Different functions and different data:
  - Decision support requires historical data which operational DBs do not typically maintain
  - DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  - Different sources typically use inconsistent data representations, codes and formats which have to be reconciled

Data Warehouse: A Multi-Tiered Architecture

- Top Tier: Front-End Tools
- Middle Tier: OLAP Server
- Bottom Tier: Data Warehouse Server
- Data

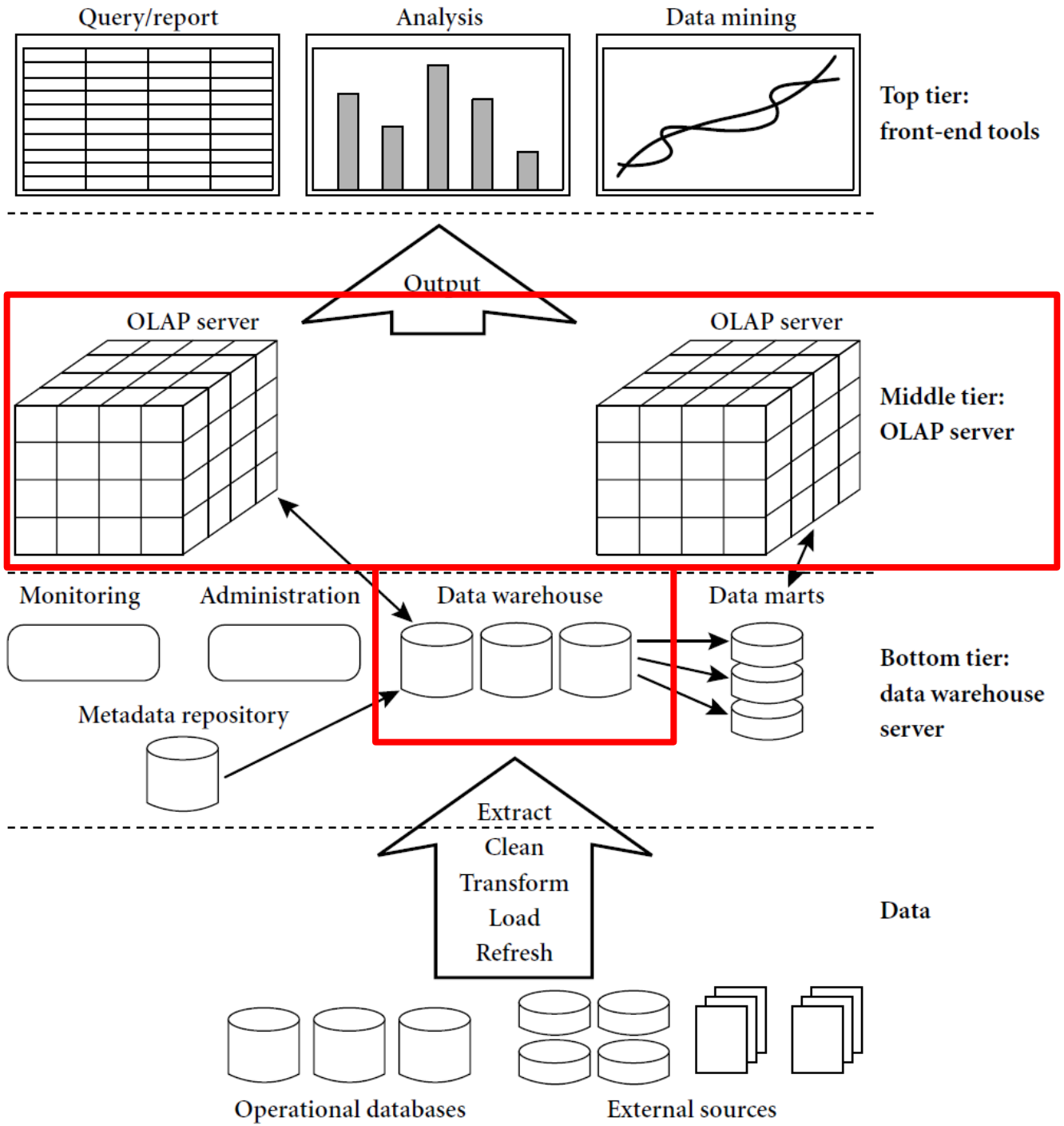


From Data to Data Warehouse: Extraction, Transformation, and Loading (ETL)

- **Data extraction**
  - get data from multiple, heterogeneous, and external sources
- **Data cleaning**
  - detect errors in the data and rectify them when possible
- **Data transformation**
  - convert data from legacy or host format to warehouse format
- **Load**
  - sort, summarize, consolidate, compute views, check integrity, and build indices and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

Metadata Repository

- **Meta data** is the data defining warehouse objects. It stores:
  - Description of the structure of the data warehouse
    - schema, view, dimensions, hierarchies, derived data definition, data mart locations and contents
  - Operational meta-data
    - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)
  - The algorithms used for summarization
  - The mapping from operational environment to the data warehouse
  - Data related to system performance
    - warehouse schema, view and derived data definitions
  - Business data
    - business terms and definitions, ownership of data, charging policies


Query/report
Analysis
Data mining
Top tier:
front-end tools
Output
OLAP server
OLAP server
Middle tier:
OLAP server
Monitoring
Administration
Data warehouse
Data marts
Metadata repository
Bottom tier:
data warehouse
server
Extract
Clean
Transform
Load
Refresh
Data
Operational databases
External sources

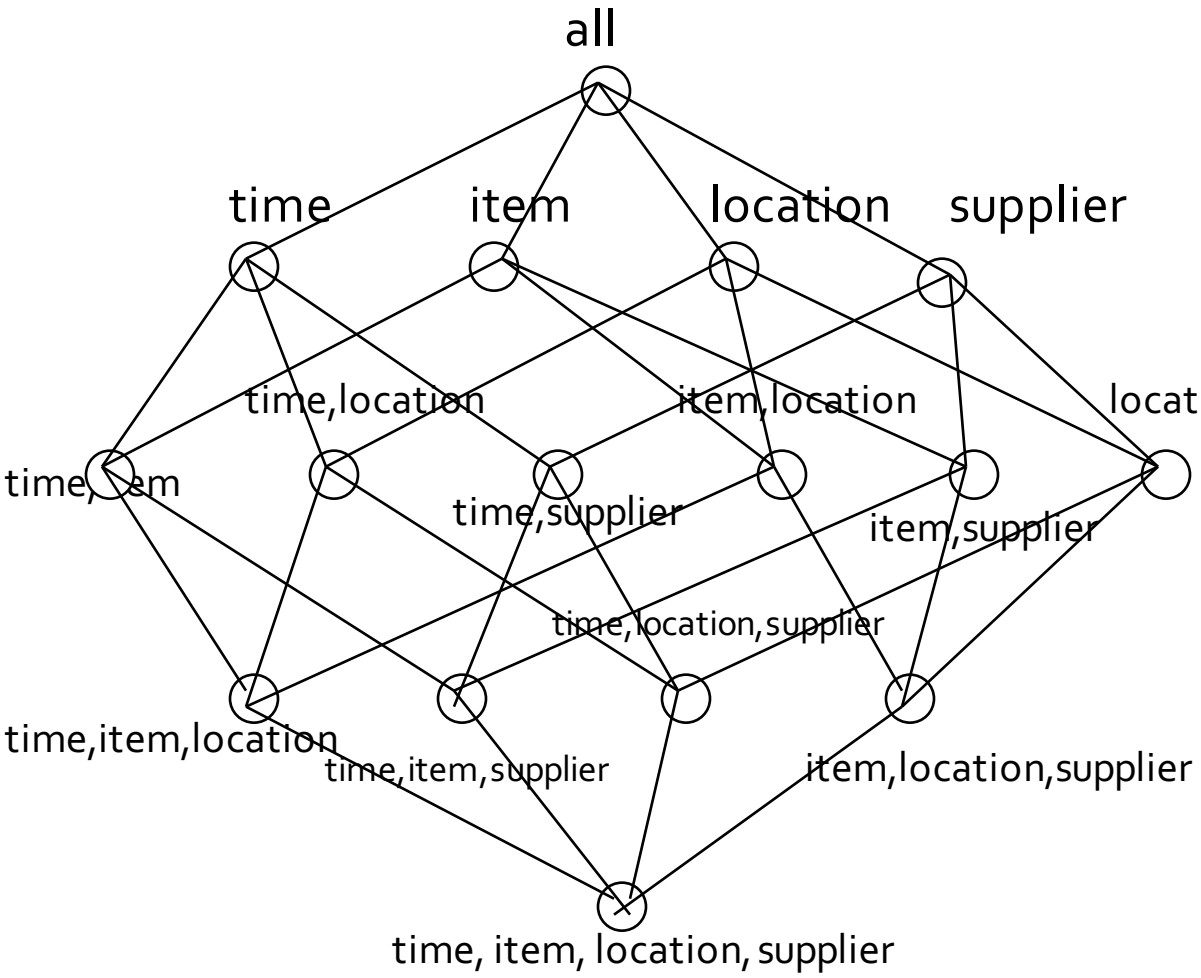
Data Warehouse

- Basic Concepts
- **Modeling: Data Cube and OLAP**
- Design and Usage
- Implementation

From Tables and Spreadsheets to Data Cubes

- A **data warehouse** is based on a multidimensional data model which views data in the form of a **data cube**
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - **Dimension tables**, such as item (item_name, brand, type), or time (day, week, month, quarter, year)
  - **Fact table** contains **measures** (such as dollars_sold) and keys to each of the related dimension tables
- **Data cube**: A lattice of cuboids
  - In data warehousing literature, an **n-D base cube** is called a **base cuboid**
  - The top most **o-D cuboid**, which holds the highest-level of summarization, is called the **apex cuboid**
  - The lattice of cuboids forms a **data cube**

Data Cube: A Lattice of Cuboids



0-D (*apex*) cuboid

1-D cuboids

2-D cuboids

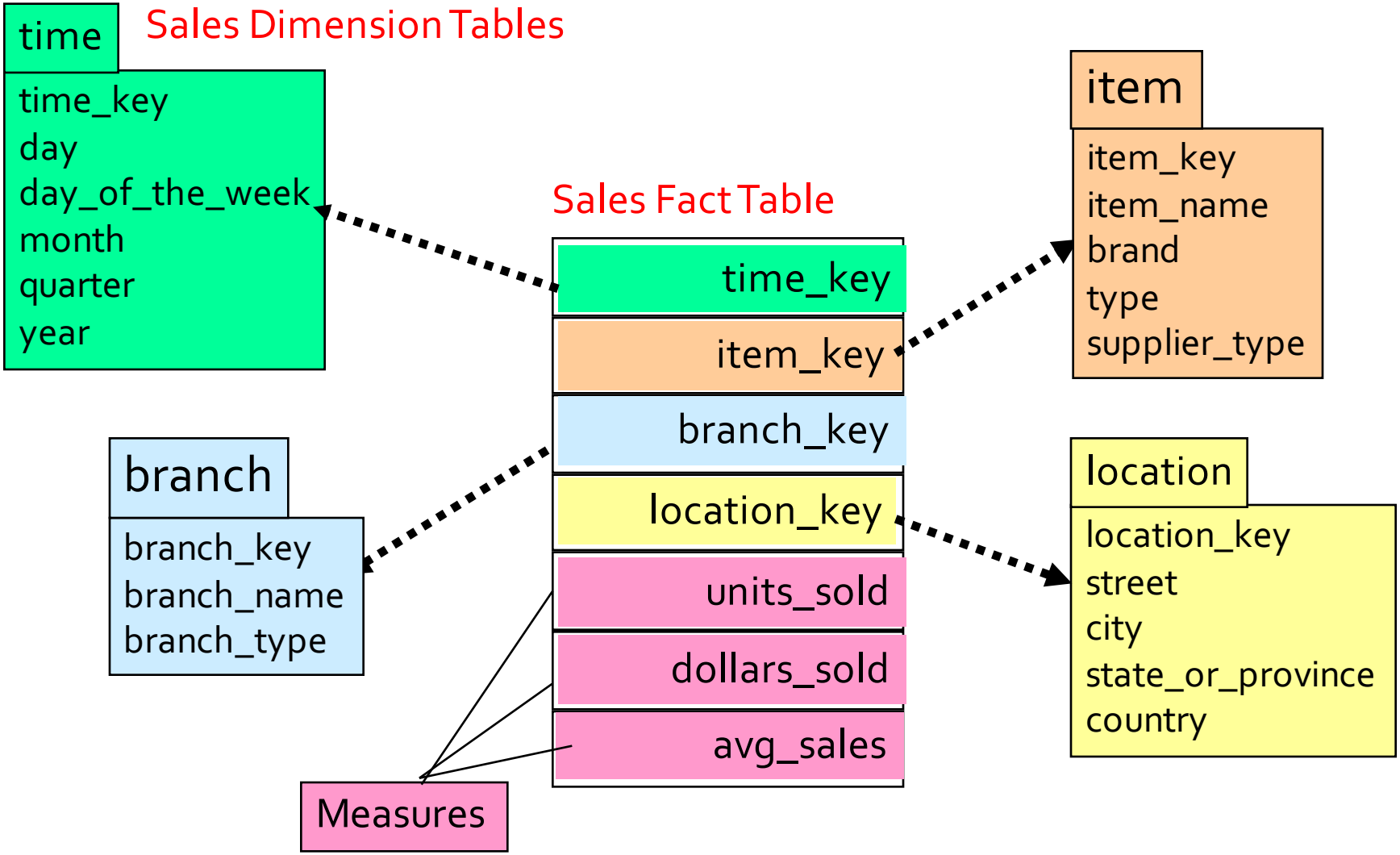
3-D cuboids

4-D (*base*) cuboid

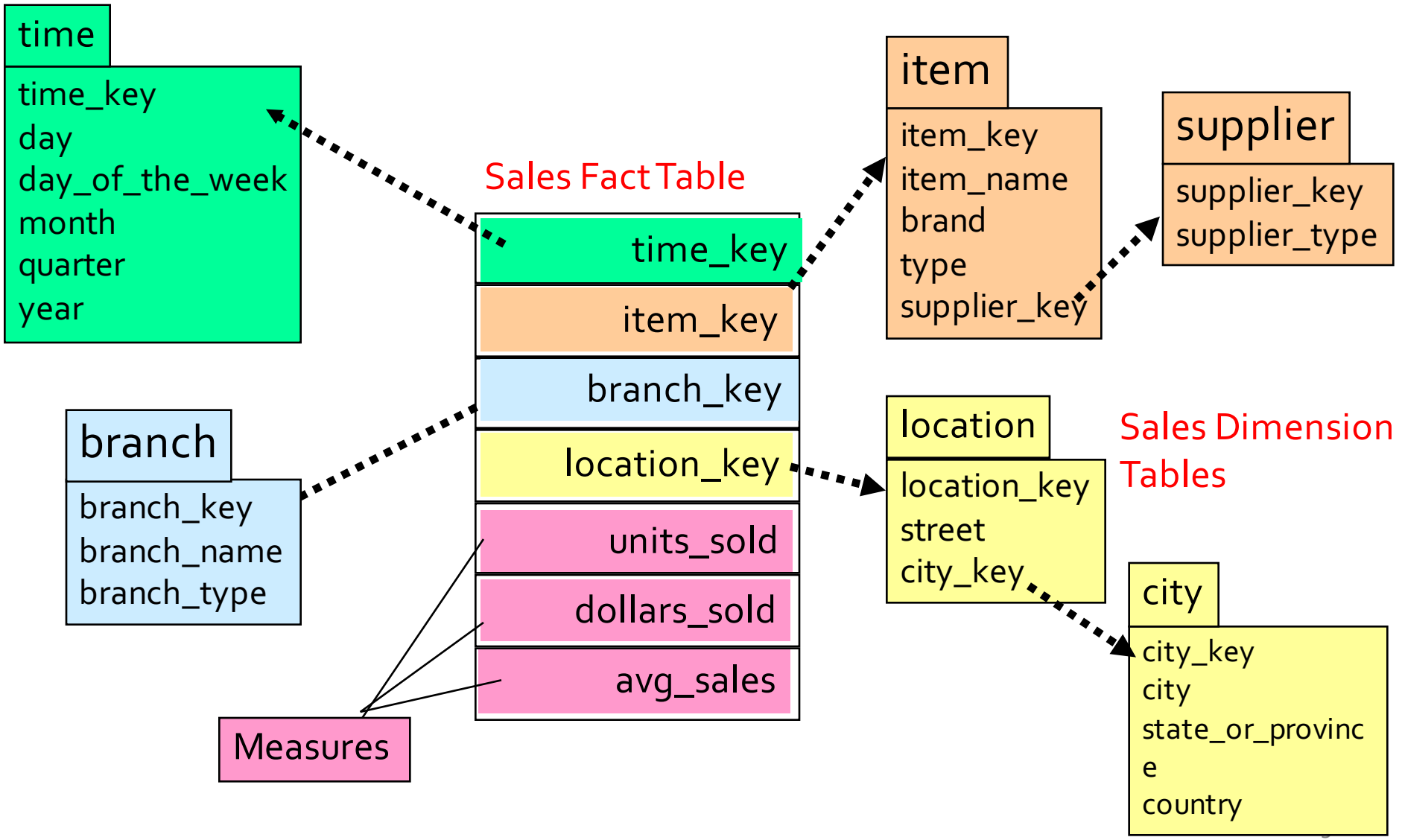
Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - **Schemas: Dimension tables and Fact tables**

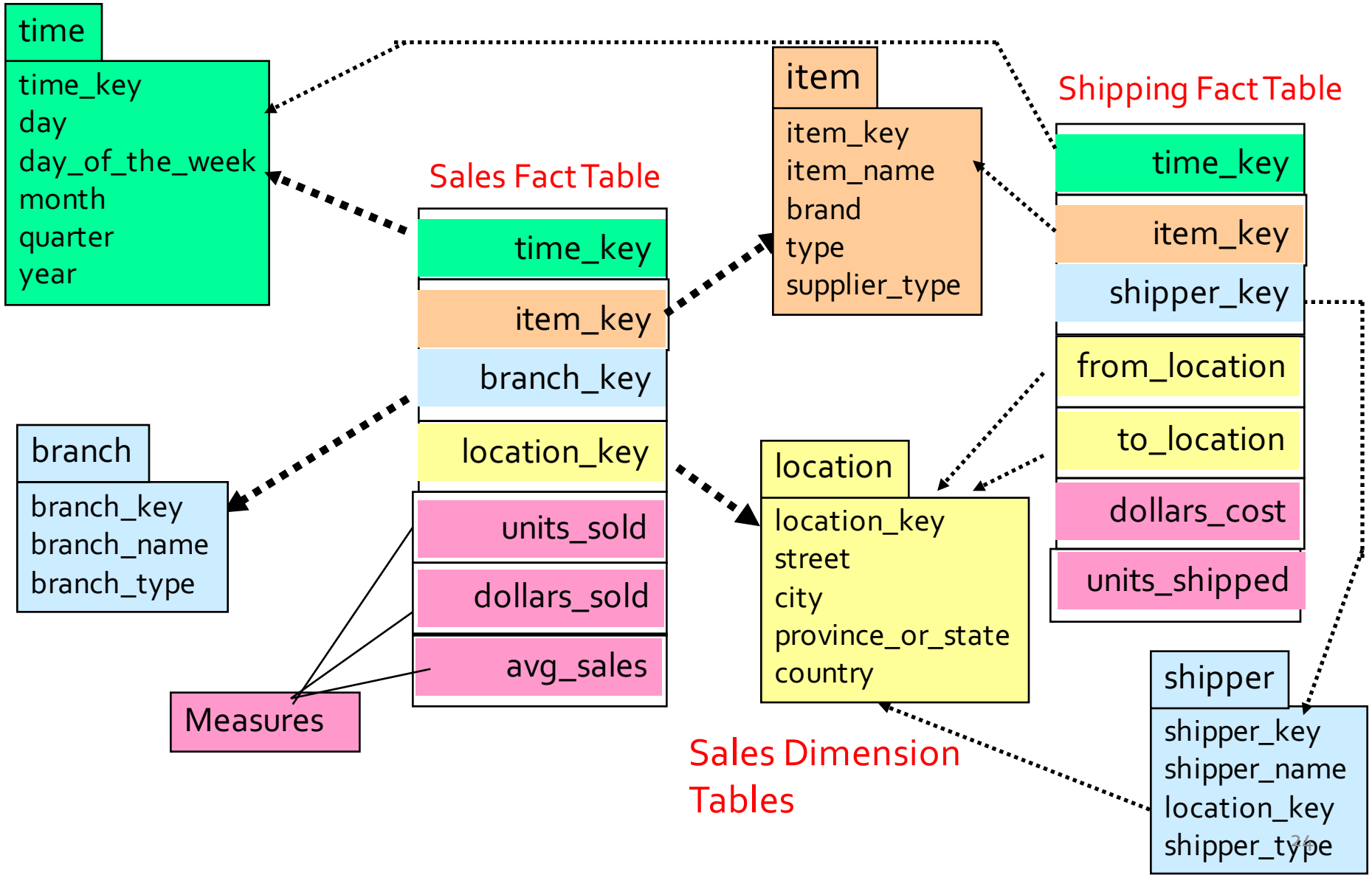
Star Schema


time
Sales Dimension Tables
time_key
day
day_of_the_week
month
quarter
year
Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales
item
item_key
item_name
brand
type
supplier_type
branch
branch_key
branch_name
branch_type
location
location_key
street
city
state_or_province
country
Measures

Snowflake Schema



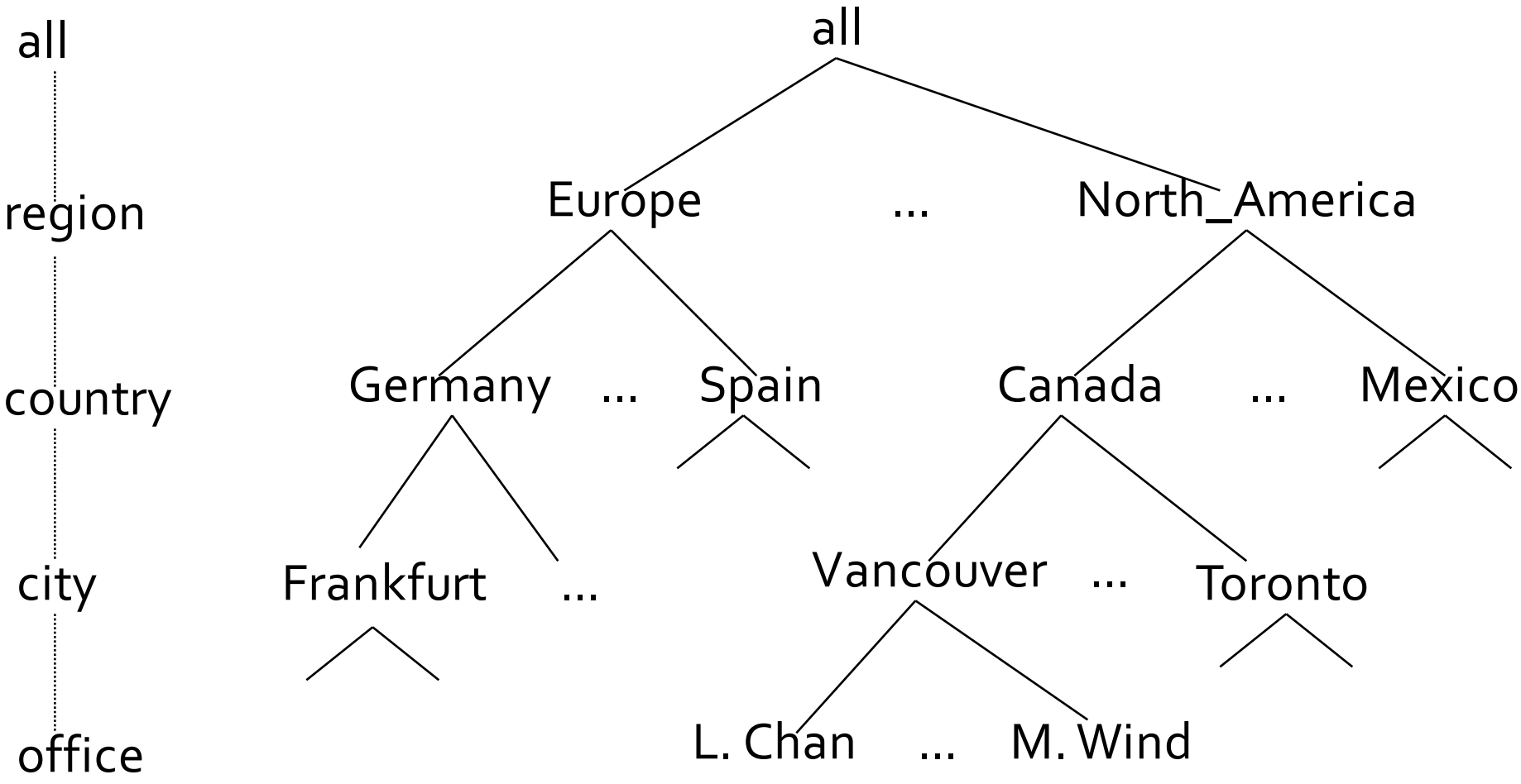
Fact Constellation



Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - **Star schema:** A fact table in the middle connected to a set of dimension tables
  - **Snowflake schema:** A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - **Fact constellations:** Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

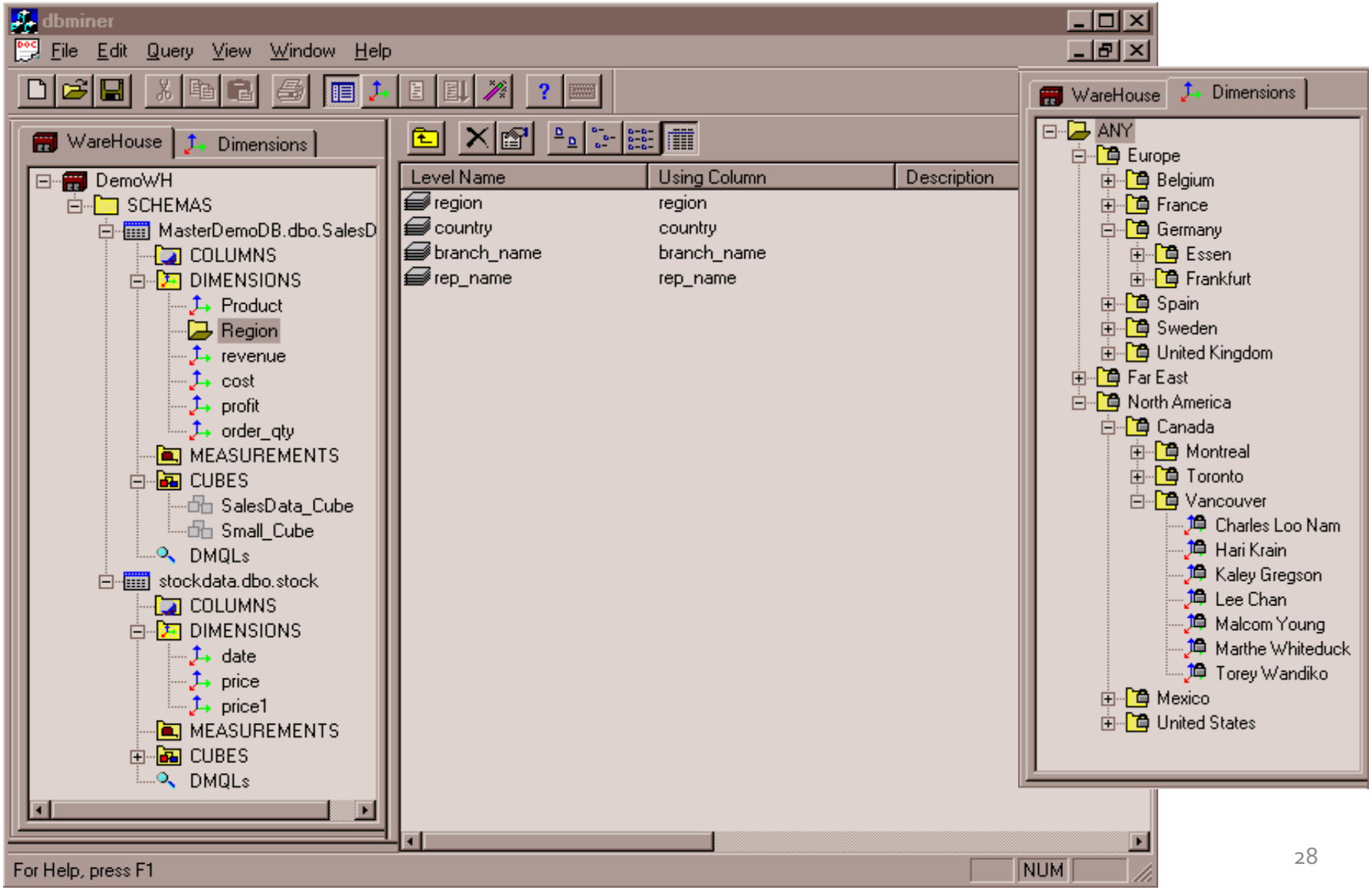
A Concept Hierarchy for a **Dimension** (location)



Data Cube Measures: Three Categories

- Distributive: if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning
    - E.g., count(), sum(), min(), max()
- Algebraic: if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function
    - avg(x) = sum(x) / count(x)
- Holistic: if there is no constant bound on the storage size needed to describe a sub-aggregate.
    - E.g., median(), mode(), rank()
- Q: How about standard_deviation(), Q1(), Q3()?

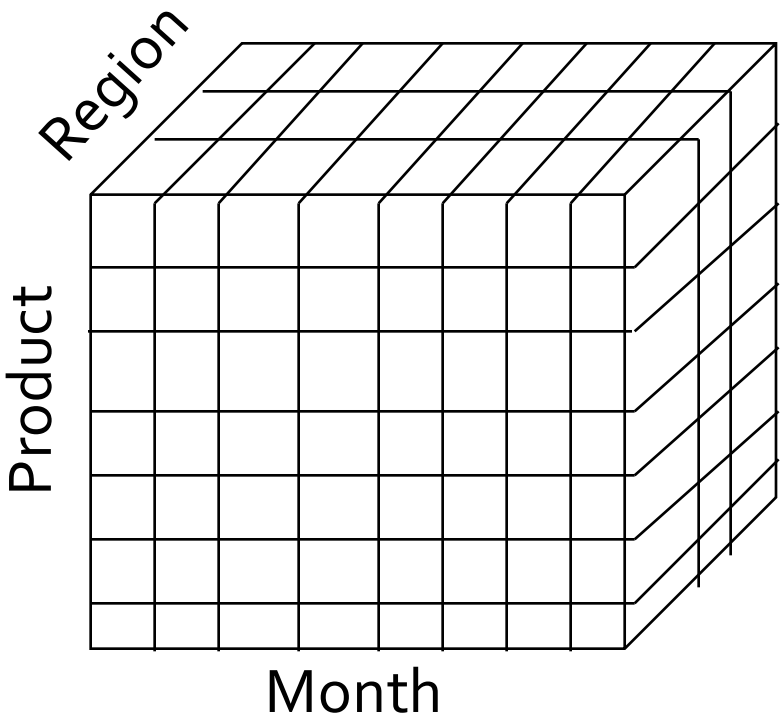
View of Warehouses and Hierarchies

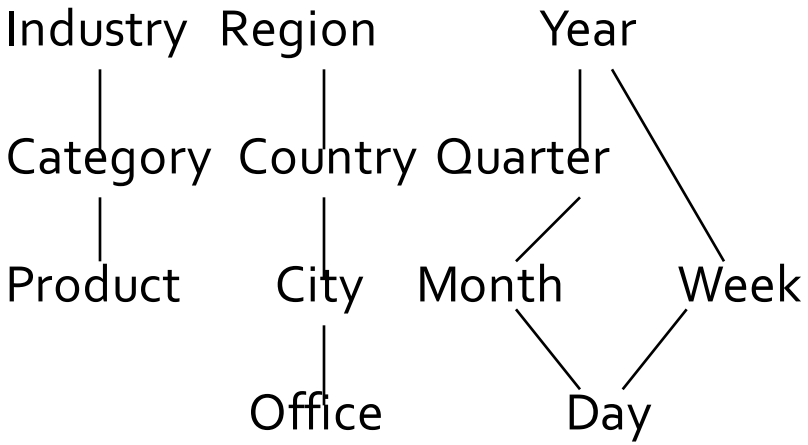


Multidimensional Data

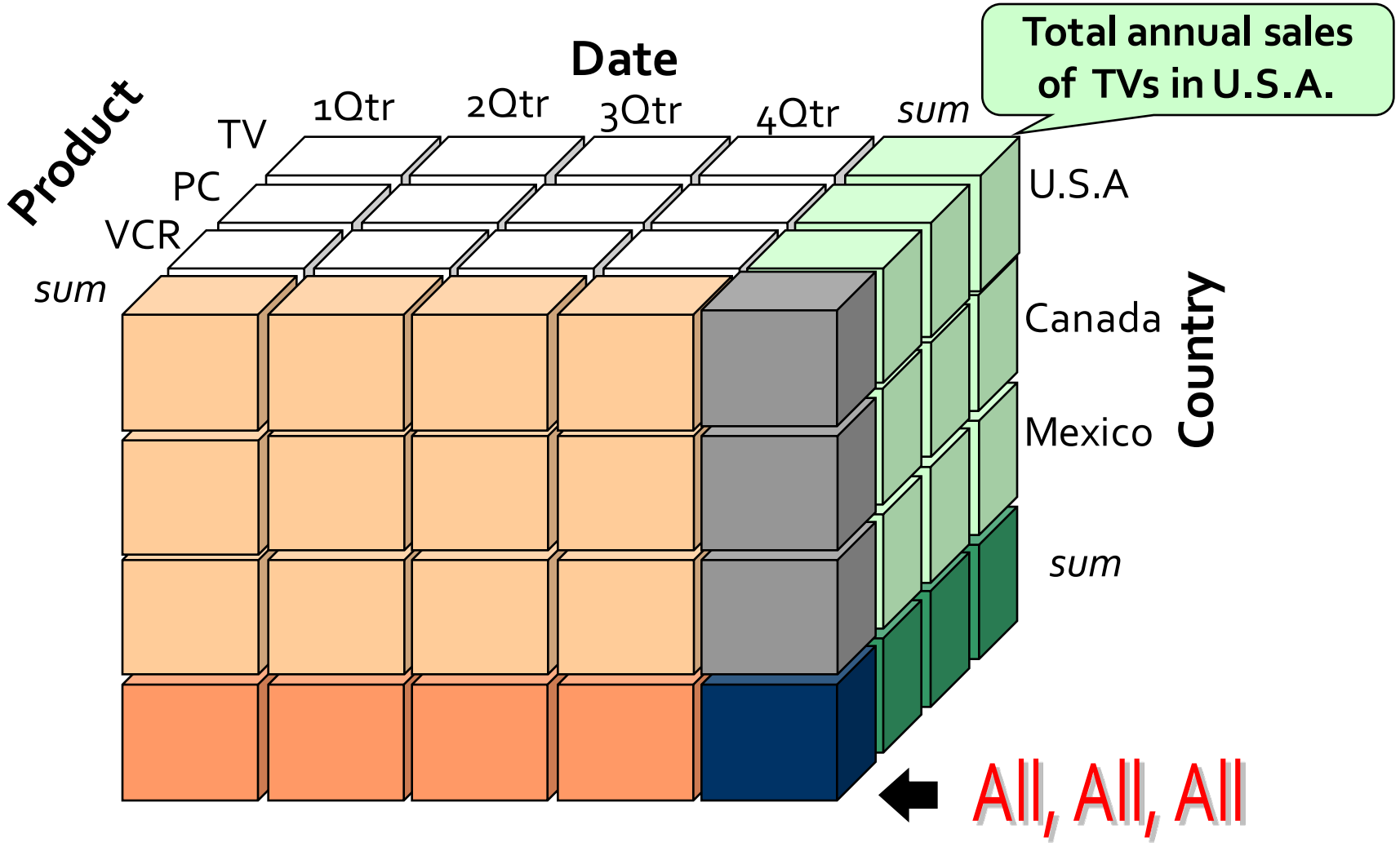
- Sales volume as a function of product, month, and region

Dimensions: *Product, Location, Time*
Hierarchical summarization paths



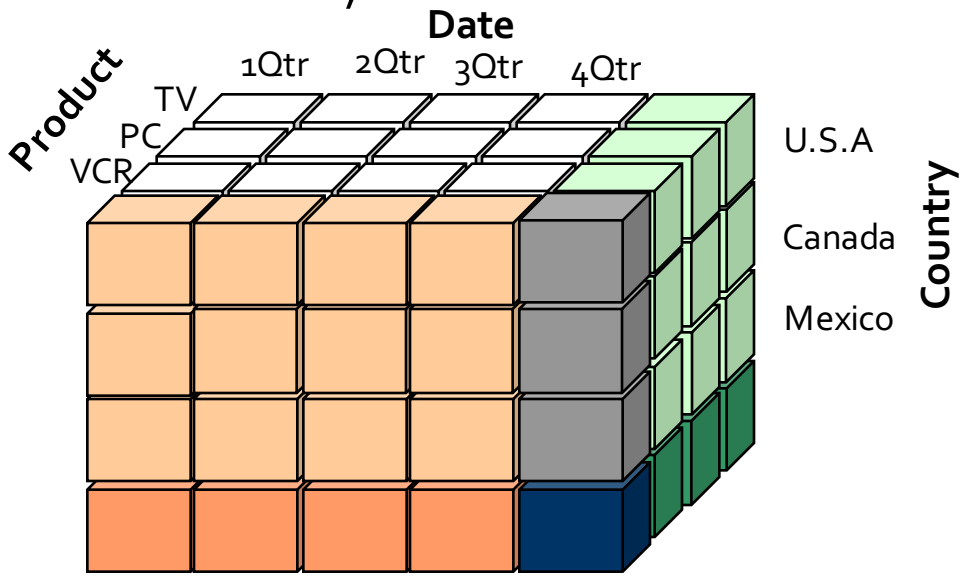


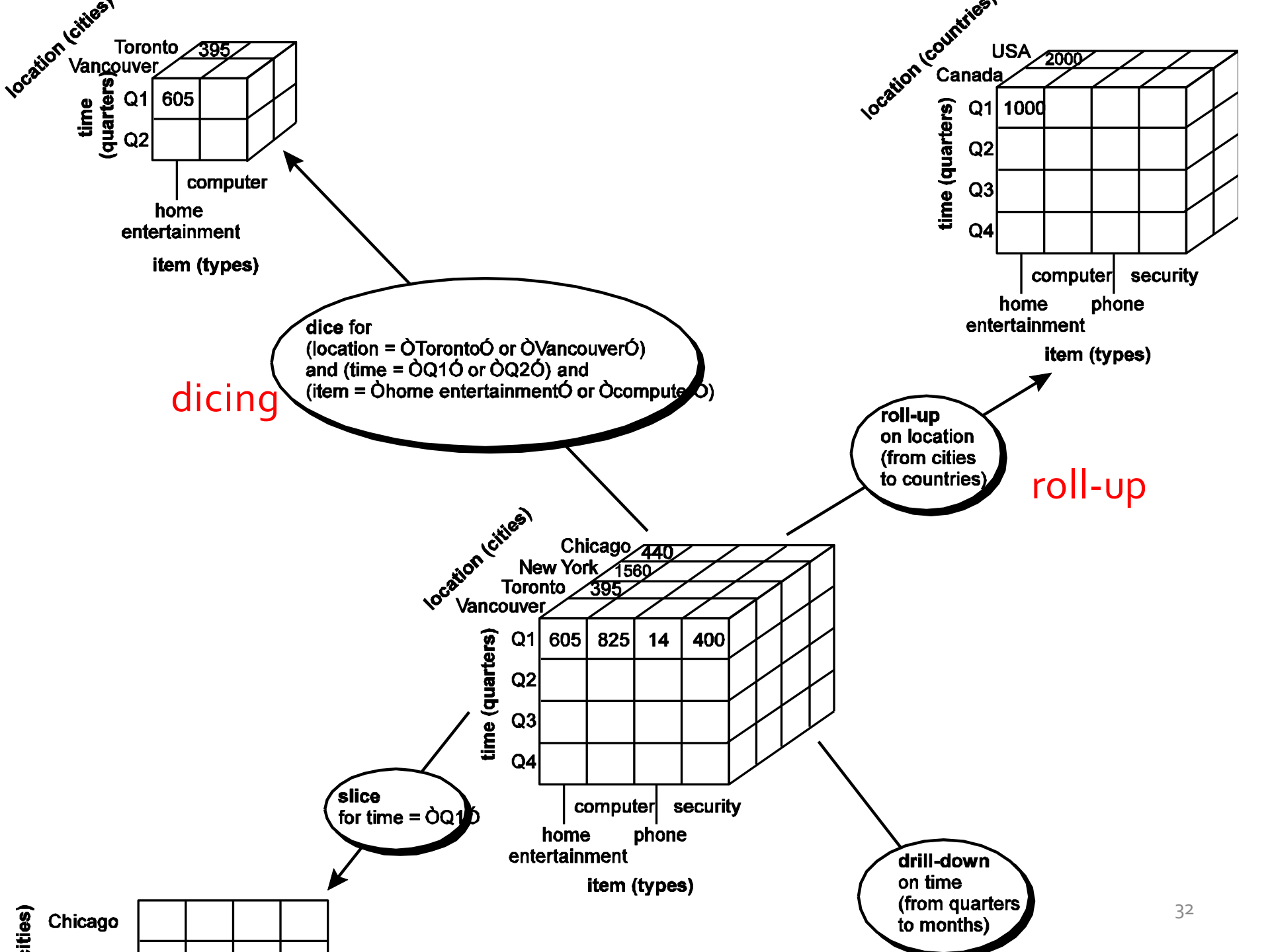
A Sample Data Cube

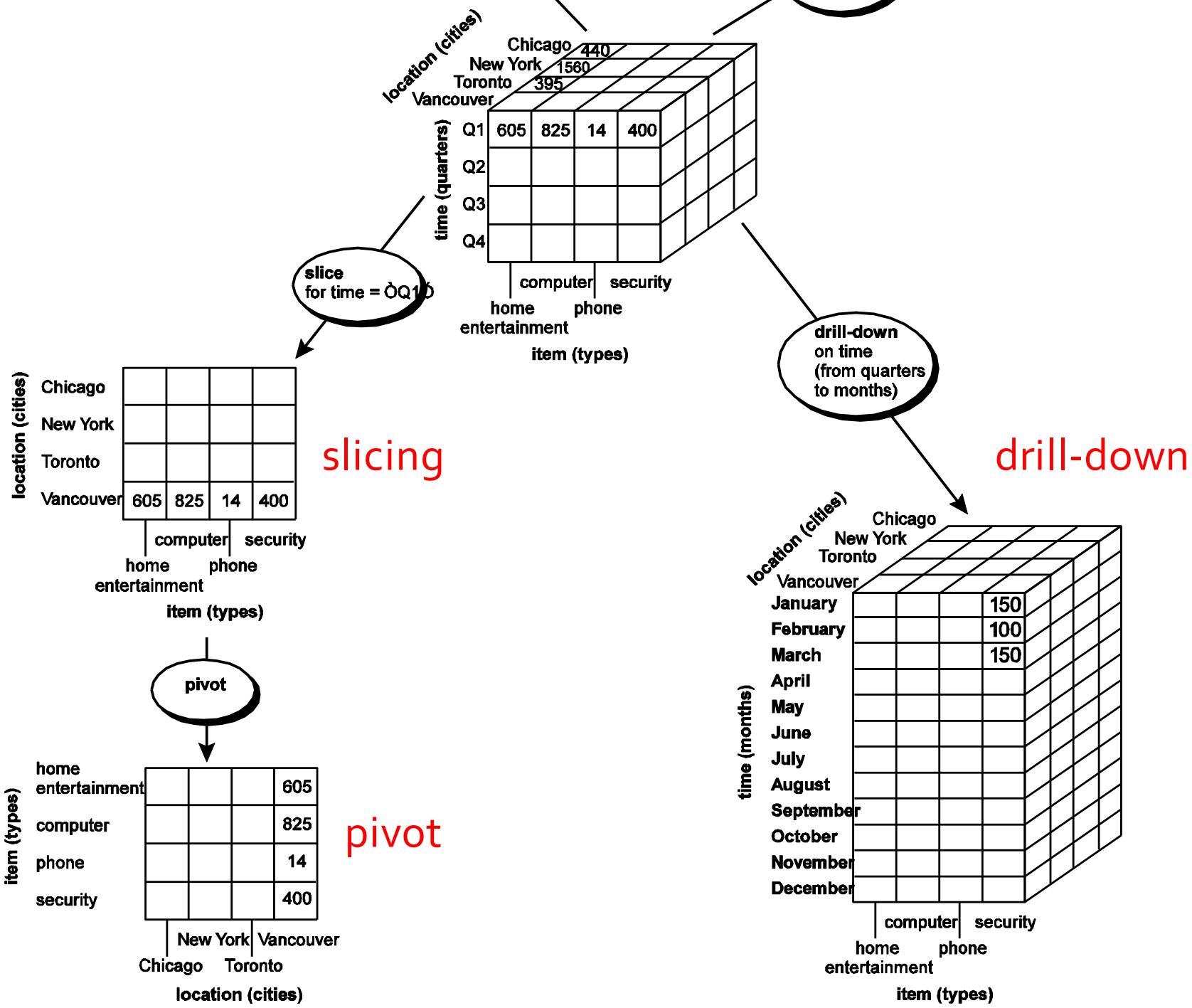


Typical OLAP Operations

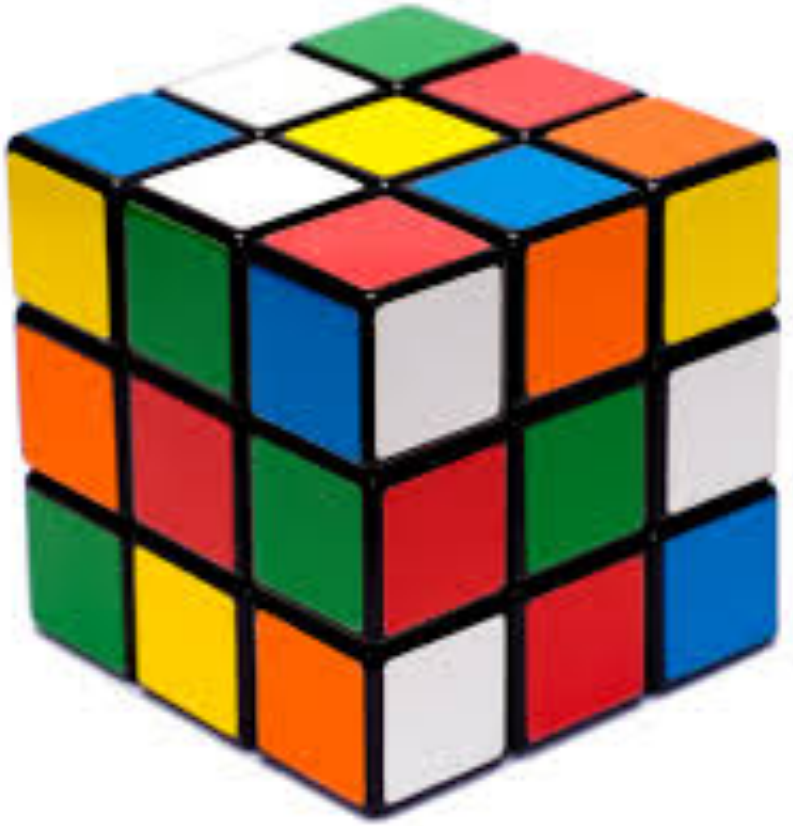
- Roll up (drill-up): summarize data
  - *by climbing up hierarchy or by dimension reduction*
- Drill down (roll down): reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- Slice and dice: *project and select*
- Pivot (rotate): *reorient the cube, visualization*




location (cities)
Toronto
Vancouver
395
time (quarters)
Q1
605
Q2
computer
home entertainment
item (types)
location (countries)
USA
Canada
2000
Q1
1000
Q2
Q3
Q4
computer
security
home entertainment
phone
item (types)
dice for
(location = ÒTorontoÓ or ÒVancouverÓ)
and (time = ÒQ1Ó or ÒQ2Ó) and
(item = Òhome entertainmentÓ or Òcompute
dicing
roll-up
on location
(from cities
to countries)
roll-up
location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
time (quarters)
Q1
605
825
14
400
Q2
Q3
Q4
computer
security
home entertainment
phone
item (types)
slice
for time = ÒQ1Ó
drill-down
on time
(from quarters
to months)
Chicago


location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
computer
security
home entertainment
phone
item (types)
slice
for time = ÒQ1Ó
drill-down
on time
(from quarters
to months)
location (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
computer
security
home entertainment
phone
item (types)
slicing
drill-down
pivot
item (types)
home entertainment
computer
phone
security
605
825
14
400
New York
Vancouver
Chicago
Toronto
location (cities)
pivot
location (cities)
Chicago
New York
Toronto
Vancouver
time (months)
January
February
March
April
May
June
July
August
September
October
November
December
150
100
150
computer
security
home entertainment
phone
item (types)

Chapter 4. Data Warehousing and On-line Analytical Processing (OLAP)



Meng Jiang

CS412 Summer 2017:

Introduction to Data Mining

Data Warehouse

- Basic Concepts
- Modeling: Data Cube and OLAP
- **Design and Usage**
- Implementation

Design of Data Warehouse: A Business Analysis Framework

- Four views regarding the design of a data warehouse
  - Top-down view
    - allows selection of the relevant information necessary for the data warehouse
  - Data source view
    - exposes the information being captured, stored, and managed by operational systems
  - Data warehouse view
    - consists of fact tables and dimension tables
  - Business query view
    - sees the perspectives of data in the warehouse from the view of end-user

Data Warehouse Design Process

- **Top-down, bottom-up approaches or a combination** of both
  - Top-down: Starts with overall design and planning (mature)
  - Bottom-up: Starts with experiments and prototypes (rapid)
- **From software engineering point of view**
  - Waterfall: structured and systematic analysis at each step before proceeding to the next
  - Spiral: rapid generation of increasingly functional systems, short intervals between successive releases
- **Typical data warehouse design process**
  - Choose a business process to model, e.g., orders, invoices, etc.
  - Choose the *grain* (*atomic level of data*) of the business process
  - Choose the dimensions that will apply to each fact table record
  - Choose the measure that will populate each fact table record

Data Warehouse Usage

- Three kinds of data warehouse applications
  - Information processing
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - Analytical processing
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - Data mining
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools

From On-Line Analytical Processing (OLAP) to On-Line Analytical Mining (OLAM)

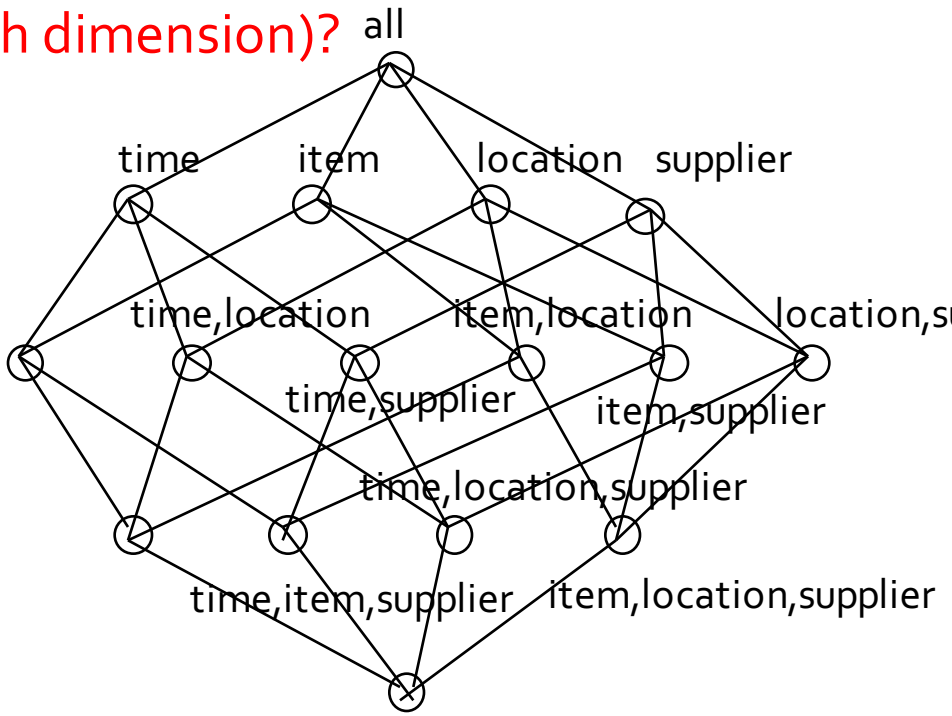
- Why **online analytical mining**?
  - High quality of data in data warehouses
    - DW contains integrated, consistent, cleaned data
  - Available information processing structure surrounding data warehouses
    - Open Database Connectivity (ODBC), Object Linking and Embedding, Database (OLEDB), Web accessing, service facilities, reporting and OLAP tools
  - OLAP-based exploratory data analysis
    - Mining with drilling, dicing, pivoting, etc.
  - On-line selection of data mining functions
    - Integration and swapping of multiple mining functions, algorithms, and tasks

Data Warehouse

- Basic Concepts
- Modeling: Data Cube and OLAP
- Design and Usage
- **Implementation**

Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L_i levels (at the i-th dimension)?



0-D (*apex*) cuboid

1-D cuboids

2-D cuboids

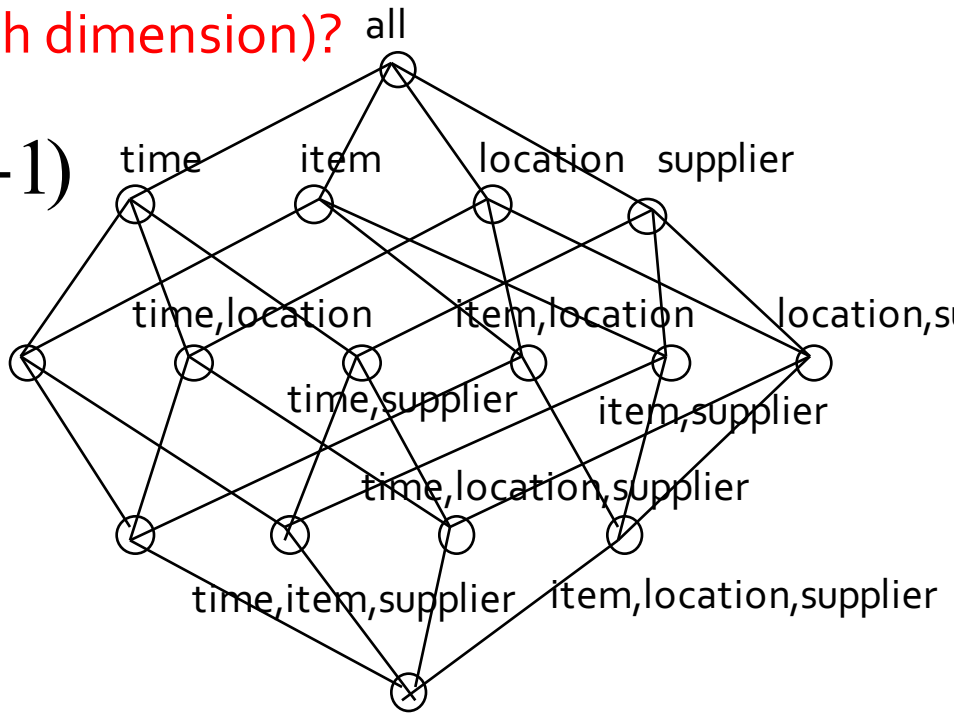
3-D cuboids

4-D (*base*) cuboid

Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L_i levels (at the i-th dimension)?

$$T = \prod_{i=1}^{n} (L_i + 1)$$



0-D (*apex*) cuboid

1-D cuboids

2-D cuboids

3-D cuboids

4-D (*base*) cuboid

Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L levels?
- Materialization of data cube
  - **Full materialization**: Materialize every (cuboid)
  - **No materialization**: Materialize none (cuboid)
  - **Partial materialization**: Materialize some cuboids
    - Which cuboids to materialize?
      - Selection based on size, sharing, access frequency, etc.

The "Compute Cube" Operator

- Cube definition and computation in DMQL

  define cube sales [item, city, year]: sum (sales_in_dollars)

  compute cube sales

- Transform it into a SQL-like language (with a new operator cube by, introduced by **Gray et al.'97**)

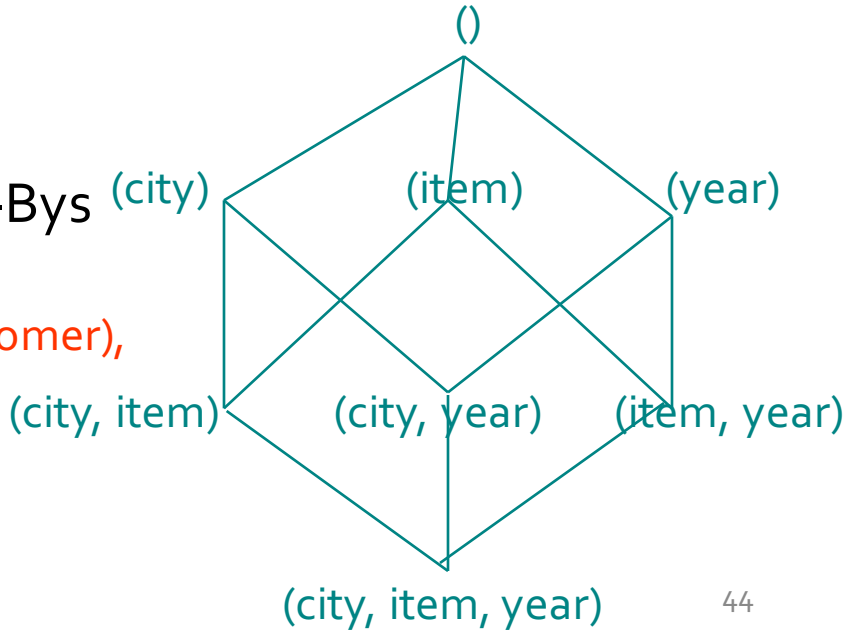
  SELECT item, city, year, SUM (amount)

  FROM SALES

  CUBE BY item, city, year

- Need compute the following Group-Bys

(year, product, customer),
(year, product), (year, customer), (product, customer),
(year), (product), (customer)
()



Data Cube

Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals
J Gray, S Chaudhuri, A Bosworth, A Layman, D Reichart, M Venkatrao, ...
Data Mining and Knowledge Discovery 1 (1), 29-53
2981 1997

Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals

Jim Gray
Surajit Chaudhuri
Adam Bosworth
Andrew Layman
Don Reichart
Murali Venkatrao
Frank Pellow
Hamid Pirahesh[1]

May 1997

Technical Report
MSR-TR-97-32

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

This paper appeared in *Data Mining and Knowledge Discov*

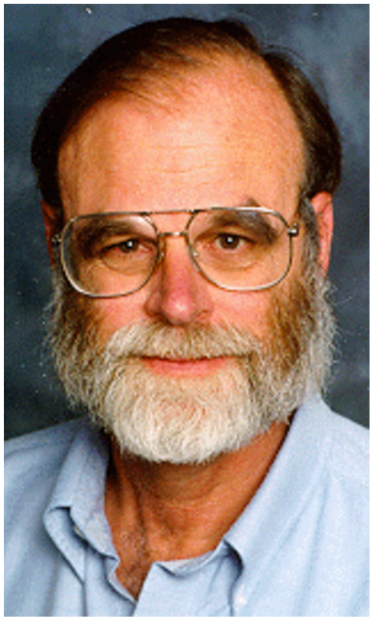
Surajit Chaudhuri *is a computer scientist best known for his contributions to database management systems. He is currently a* ***distinguished scientist at Microsoft Research****, where he leads the Data Management, Exploration and Mining group.*

Adam Bosworth *is a former* ***Vice President of Product Management at Google Inc****. from 2004–2007; prior to that, he was senior VP Engineering and Chief Software Architect at BEA Systems responsible for ...*

Hamid Pirahesh*, Ph.D., is* ***an IBM fellow, ACM Fellow and a senior manager responsible for the exploratory database department at IBM Research*** *- Almaden in San Jose, California. Dr. Hamid Pirahesh is the senior manager at IBM Almaden Research Center in San Jose, California.*

https://jimgray.azurewebsites.net/

Jim Gray Summary Home Page



Microsoft eScience Group

As you may be aware, Jim Gray has gone missing.

We (his colleagues in Microsoft Research) have heard from many of his collaborators about projects and collaborations that he had underway with them and who are unsure how to proceed. If you find yourself in this situation, please email grayproj@microsoft.com and we will follow up with you to find the best way forward.

Jim Gray is a researcher and manager of Microsoft Research's eScience Group. His primary research interests are in databases and transaction processing systems -- with particular focus on using computers to make scientists more productive. He and his group are working in the areas of astronomy, geography, hydrology, oceanography, biology, and health care. He continues a long-standing interest on building supercomputers with commodity components, thereby reducing the cost of storage, processing, and networking by factors of 10x to 1000x over low-volume solutions. This includes work on building fast networks, on building huge web servers with *CyberBricks*, and building very inexpensive and very high-performance storage servers.

Jim also is working with the astronomy community to build the world-wide telescope and has been active in building online databases like http://terraService.Net and http://skyserver.sdss.org. When the entire world's astronomy data is on the Internet and is accessible as a single distributed database, the Internet will be the world's best telescope. This is part of the larger agenda of getting all information online and easily accessible (digital libraries, digital government, online science ...). More generally, he is working with the science community (Oceanography, Hydrology, environmental monitoring, ..) to build the world-wide digital library that integrates all the world's scientific literature and the data in one easily-accessible collection. He is active in the research community, is an ACM, NAE, NAS, and AAAS Fellow, and received the ACM Turing Award for his work on transaction processing. He also edits of a series of books on data management.

https://en.wikipedia.org/wiki/Jim_Gray_(computer_scientist)

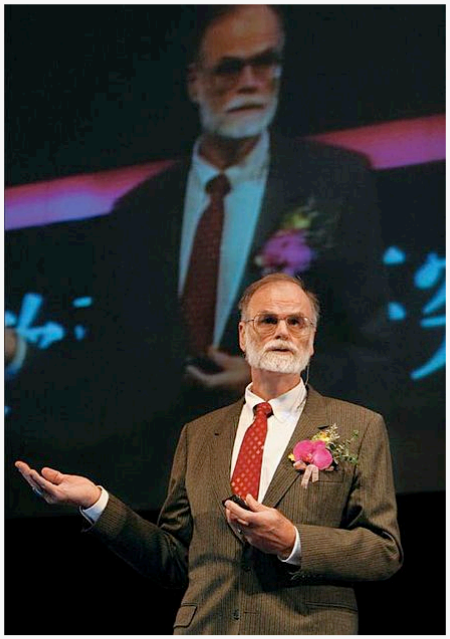
James Nicholas "Jim" Gray (born January 12, 1944; presumed lost at sea January 28, 2007; declared deceased May 16, 2012[4]) was an American computer scientist who received the Turing Award[5] in 1998 "for seminal contributions to database and transaction processing research and technical leadership in system implementation."

Contents [hide]

Early years [edit]

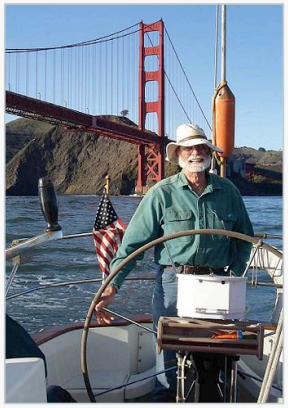
Gray was born in San Francisco, California, the second child of a mother who was a teacher and a father in the U.S. Army; the family moved to Rome where Gray spent most of the first three years of his life, learning to speak Italian before English.[2] The family then moved to Virginia, spending about four years there, until Gray's parents divorced, after

Jim Gray



Gray in 2006

Born	James Nicholas Gray January 12, 1944[1] San Francisco, California[2]
Disappeared	January 28, 2007 (aged 63) Waters near San Francisco
Status	Dead in absentia, May 16, 2012 (aged 68)
Nationality	American
Alma mater	University of California, Berkeley (Ph.D)
Occupation	Computer scientist
Employer	IBM Tandem Computers DEC Microsoft



Jim Gray on the *Tenacious* in January 2006

On Sunday, January 28, 2007, during a short solo sailing trip to the Farallon Islands near San Francisco to scatter his mother's ashes, Gray and his 40-foot yacht, *Tenacious*, were reported missing by his wife, Donna Carnes. The Coast Guard searched for four days using a C-130 plane, helicopters, and patrol boats but found no sign of the vessel.[21][22][23][24]

Gray's boat was equipped with an automatically deployable EPIRB (Emergency Position-Indicating Radio Beacon), which should have deployed and begun transmitting the instant his vessel sank. The area around the Farallon Islands where Gray was sailing is well north of the East-West ship channel used by freighters entering and leaving San Francisco Bay. The weather was clear that day and no ships reported striking his boat, nor were any distress radio transmissions reported.

On February 1, 2007, the DigitalGlobe satellite did a scan of the area, generating thousands of images.[25] The images were posted to Amazon Mechanical Turk in order to distribute the work of searching through them, in hopes of spotting his boat.

In the immediate aftermath of the disappearance, many theories were put forward on how Gray disappeared.[26]

After being missing for five years, Gray was legally assumed to have died at sea on January 28, 2012.[4][33]

Efficient Processing OLAP Queries

- **Determine which operations** should be performed on the available cuboids
  - Transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g., dice = selection + projection
- **Determine which materialized cuboid(s)** should be selected for OLAP op.
  - Let the query to be processed be on {*brand, province_or_state*} with the condition "*year = 2004*", and there are 4 materialized cuboids available:

    1) {*year, item_name, city*}

    2) {*year, brand, country*}

    3) {*year, brand, province_or_state*}

    4) {*item_name, province_or_state*} where *year = 2004*

    Which should be selected to process the query?

Efficient Processing OLAP Queries

- **Determine which operations** should be performed on the available cuboids
  - Transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g., dice = selection + projection
- **Determine which materialized cuboid(s)** should be selected for OLAP op.
  - Let the query to be processed be on {*brand, province_or_state*} with the condition "*year = 2004*", and there are 4 materialized cuboids available:

    1) {*year, item_name, city*}

    2) {*year, brand, country*}

    3) {*year, brand, province_or_state*} √

    4) {*item_name, province_or_state*} where *year = 2004*

    Which should be selected to process the query?

OLAP Server Architectures

- **Relational OLAP (ROLAP)**
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware
  - Include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
  - Greater scalability
- **Multidimensional OLAP (MOLAP)**
  - Sparse array-based multidimensional storage engine
  - Fast indexing to pre-computed summarized data
- **Hybrid OLAP (HOLAP)** (e.g., Microsoft SQLServer)
  - Flexibility, e.g., low level: relational, high-level: array
- Specialized SQL servers (e.g., Redbricks)
  - Specialized support for SQL queries over star/snowflake schemas

Summary

- Data warehousing: A multi-dimensional model of a data warehouse
  - A data cube consists of *dimensions* & *measures*
  - Star schema, snowflake schema, fact constellations
  - OLAP operations: drilling, rolling, slicing, dicing and pivoting
- Data Warehouse Architecture, Design, and Usage
  - Multi-tiered architecture
  - Business analysis design framework
  - Information processing, analytical processing, data mining, OLAM
- Implementation: Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - OLAP query processing
  - OLAP servers: ROLAP, MOLAP, HOLAP

References

- S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. VLDB'96
- D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek. Efficient view maintenance in data warehouses. SIGMOD'97
- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. ICDE'97
- **S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997**
- J. Gray, et al. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Mining and Knowledge Discovery, 1:29-54, 1997.
- A. Gupta and I. S. Mumick. Materialized Views: Techniques, Implementations, and Applications. MIT Press, 1999
- J. Han. Towards on-line analytical mining in large databases. *SIGMOD Record*, 1998
- V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. SIGMOD'96

References (cont.)

- C. Imhoff, N. Galemmo, and J. G. Geiger. Mastering Data Warehouse Design: Relational and Dimensional Techniques. John Wiley, 2003
- W. H. Inmon. Building the Data Warehouse. John Wiley, 1996
- R. Kimball and M. Ross. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2ed. John Wiley, 2002
- P. O'Neil and D. Quass. Improved query performance with variant indexes. SIGMOD'97
- S. Sarawagi and M. Stonebraker. Efficient organization of large multidimensional arrays. ICDE'94
- P. Valduriez. Join indices. ACM Trans. Database Systems, 12:218-246, 1987.
- J. Widom. Research problems in data warehousing. CIKM'95.
- K. Wu, E. Otoo, and A. Shoshani, Optimal Bitmap Indices with Efficient Compression, ACM Trans. on Database Systems (TODS), 31(1), 2006, pp. 1-38.