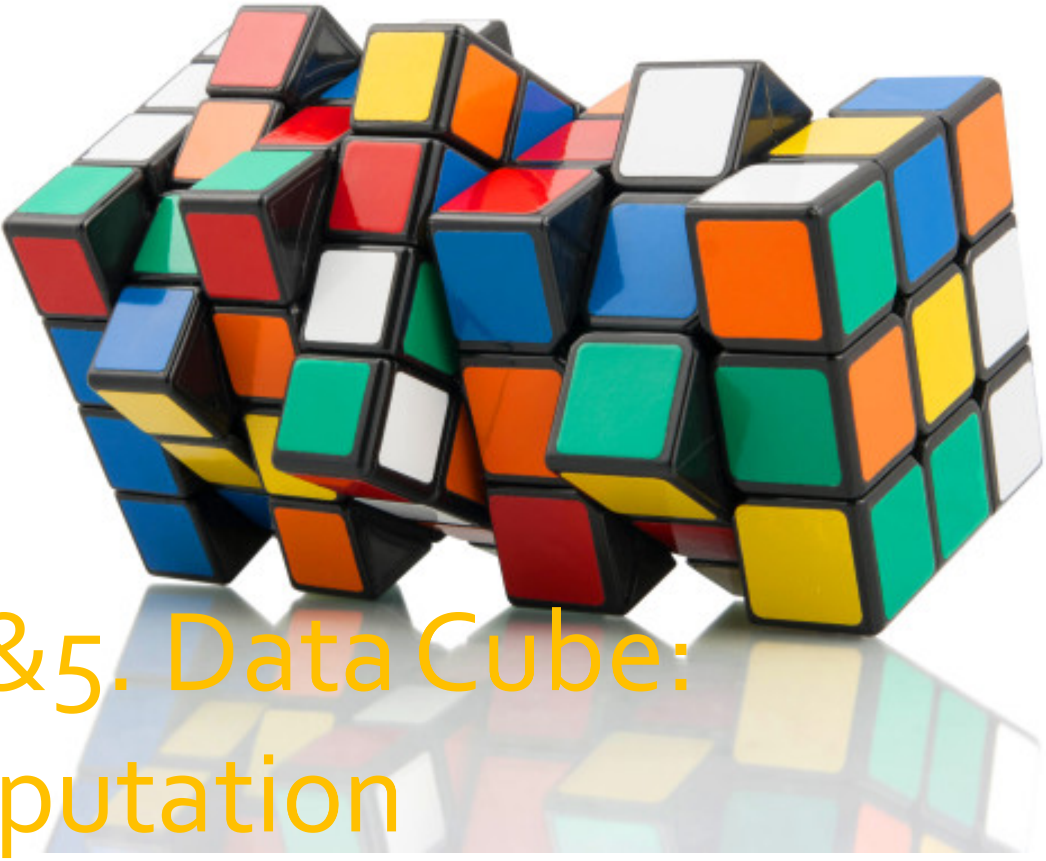


Chapter 4&5. Data Cube: Cube Computation

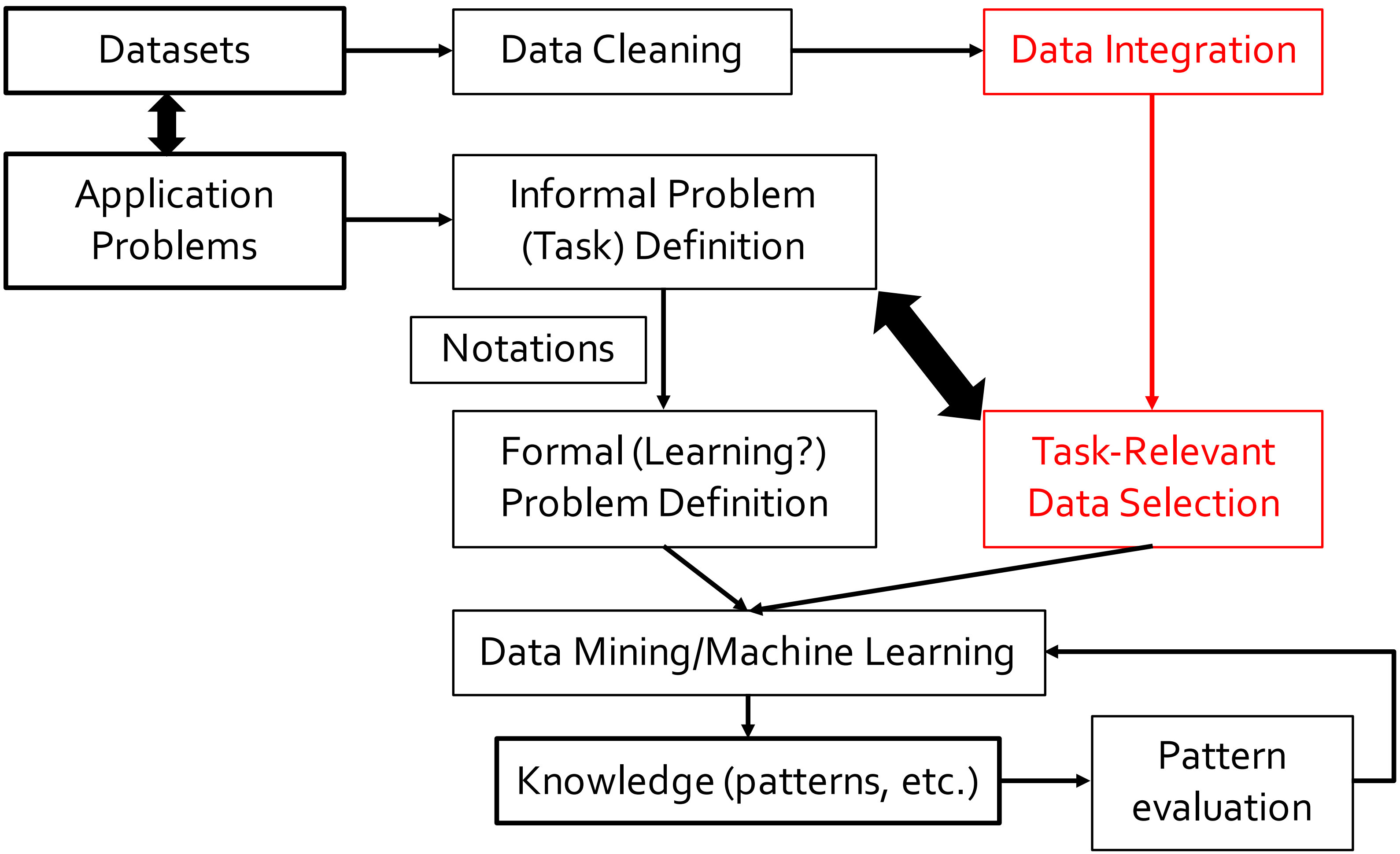
Meng Jiang

CSE 40647/60647 Data Science Fall 2017

Introduction to Data Mining

Previously on Data Science …

- Chapter 1. Introduction.



Previously on Data Science …

- Chapter 2. Get to Know Your Data.
  - Data Objects and Attribute Types
  - Basic Statistical Descriptions
    - Central tendency (mean, median, mode, etc.)
    - Outlierness (variance, standard deviation, z-score, etc.)
  - Data Visualization
    - Box plot, Histogram, Bar chart, Q plot, Q-Q plot, Scatter plot, etc.
  - Measuring Data Similarity and Dissimilarity
    - Minkowski distances
    - Jaccard/cosine similarity
    - KL divergence

Previously on Data Science …

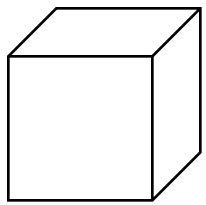
- Chapter 3. Data Processing.
  - Data cleaning: Missing data, Noisy data
  - Data integration: Redundant data
    - Correlation analysis: Chi-square test, Covariance
  - Data reduction
    - Regression analysis: Linear, non-Linear
    - Histogram, Clustering, Sampling
    - Normalization: Min-max, Z-score, Decimal scaling
  - Dimensionality reduction
    - Feature selection
    - Feature extraction: PCA (eigenvectors), etc.

Concrete Learning Goals

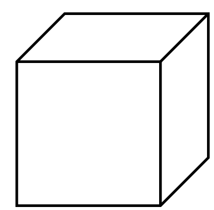
- **Can process raw data: data cleaning, data integration, data reduction, dimension reduction**
- Can describe data warehouse, OLAP, data cube concepts and technology that work on multi-dimensional datasets
- **Can use Apriori and FP-Growth for frequent pattern mining**
- Can describe diverse patterns, sequential patterns, graph patterns
- **Can use Decision Tree, Naïve Bayes, Ensembles for classification**
- Can describe SVMs and Neural Networks for classification
- **Can use K-Partitioning Methods (K-Means, etc.) for clustering**
- Can describe Kernel-based Clustering and Density-based Clustering
- **Can use appropriate measures to evaluate results of different functionalities**

Cells: Dimension, Dimension Level and Dimension Value

A cell of transactions:



A cell of papers:

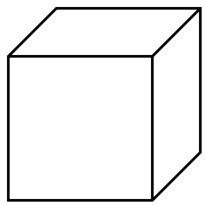


Cells: Dimension Level and **Concept Hierarchy**

A cell of transactions:

Time: Year-Quarter-Month-Week-Day
Location: Country-State-City-Street
Item: Department-Product-Model

Time Date **09/16**



milk Product Item

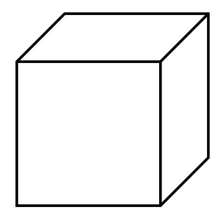
South Bend

City

Location

A cell of papers:

Dataset - **Twitter**

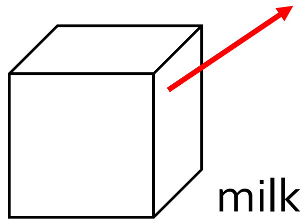


matrix factorization - Method

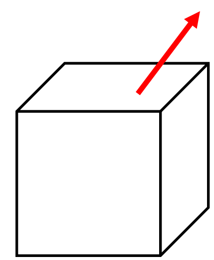
social spam detection - Problem

Cells: Facts or Measures

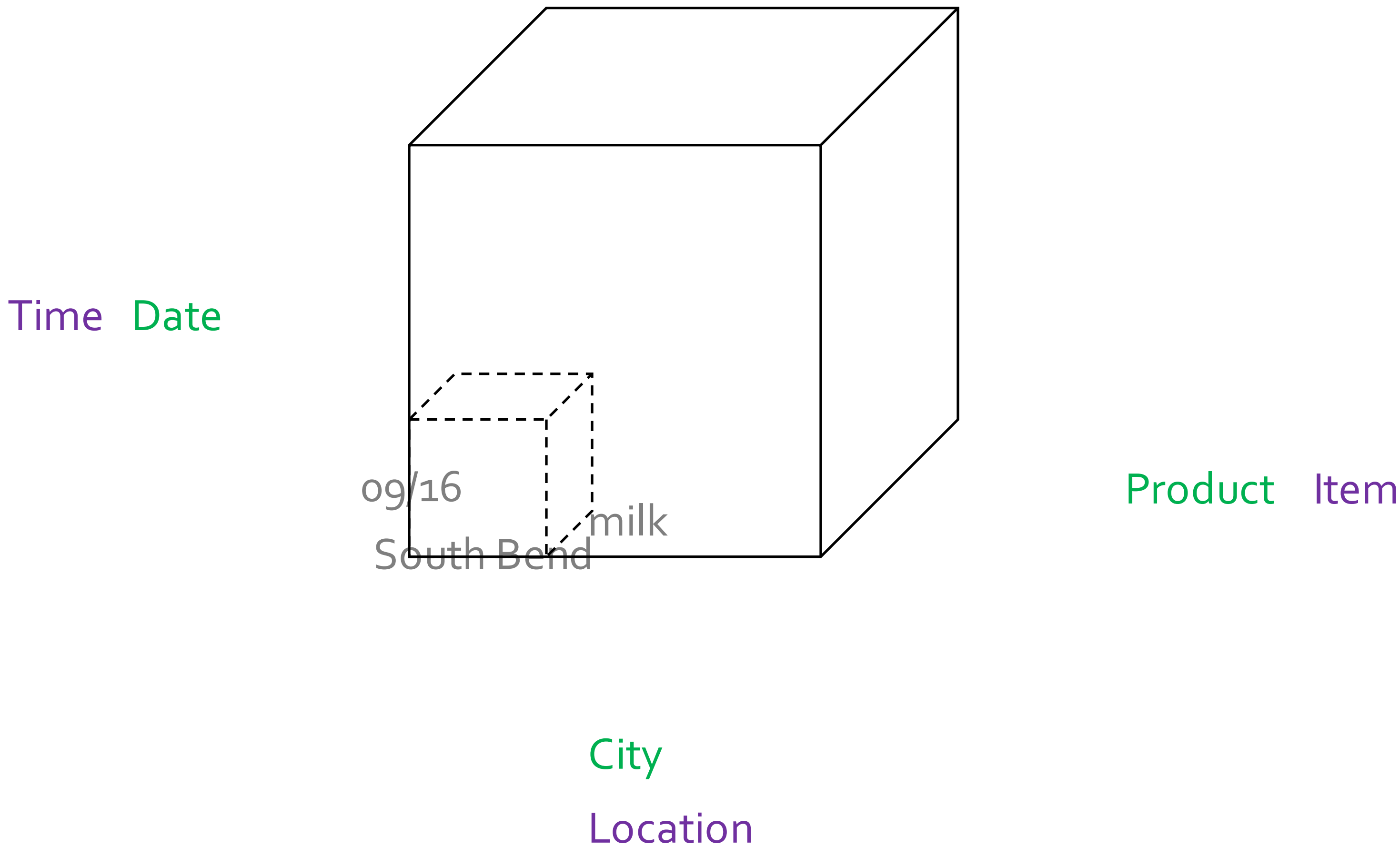
A cell of transactions:



A cell of papers:



Cuboids: Dimension, Dimension Level



Base Cells and Aggregate Cells

- Suppose a cuboid has 3 dimensions (time, location, item) at specific dimension levels (date, city, product).
- Base cells
  - (09/16, South Bend, milk)
- Aggregate cells
  - (*, South Bend, milk)
  - (09/16, *, milk)
  - (09/16, South Bend, *)
  - (*, *, milk)
  - (*, South Bend, *)
  - (09/16, *, *)
  - (*, *, *), called the Apex cell

parent vs child cells

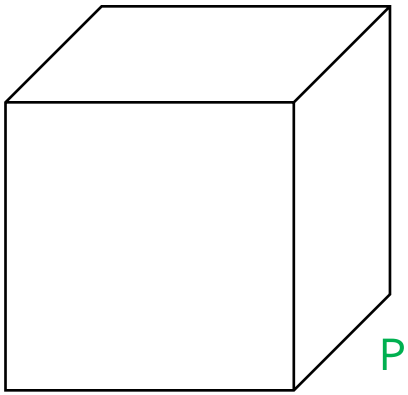
ancestor vs descendant cells

sibling cell:

(09/16, Mishawaka, milk)

Base Cuboids and Aggregate Cuboids

Time Date



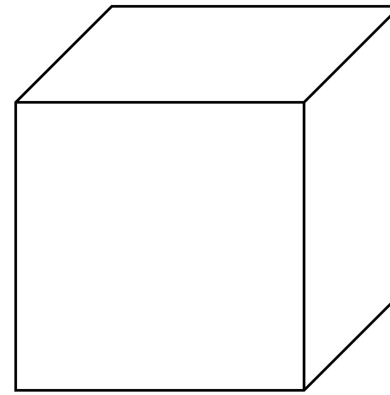
Product Item

City

Location

(Date, City, Product)

Time Month



Department

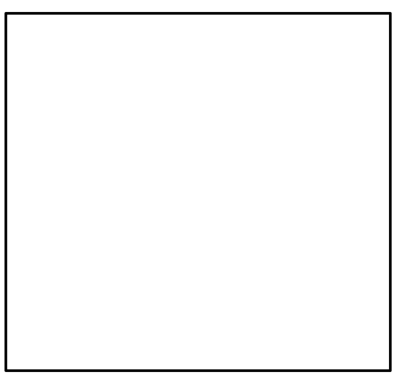
Item

Country

Location

(Month, Country, Department)

Time Date



City

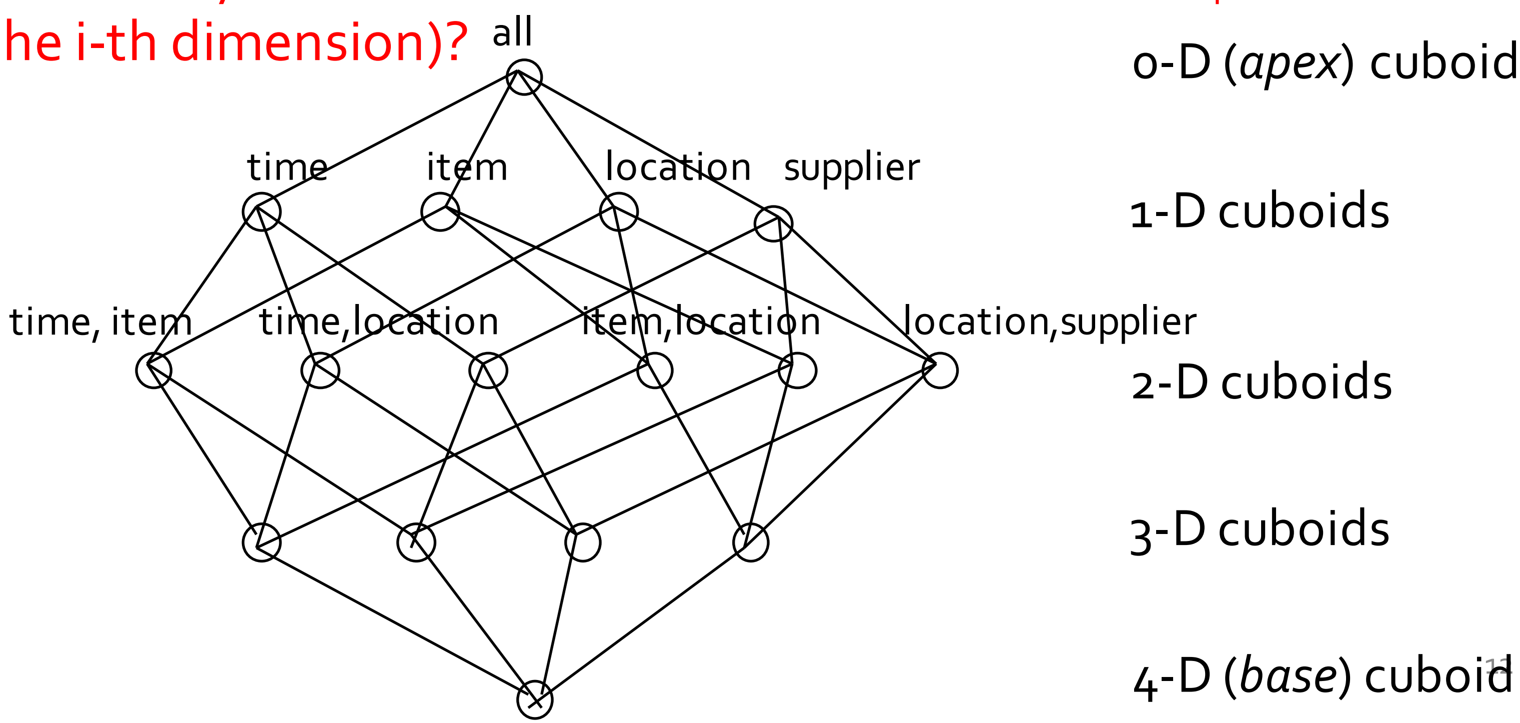
Location

(Date, City, *)

Apex cuboid: (*, *, *)

(N-Dimensional) Data Cube

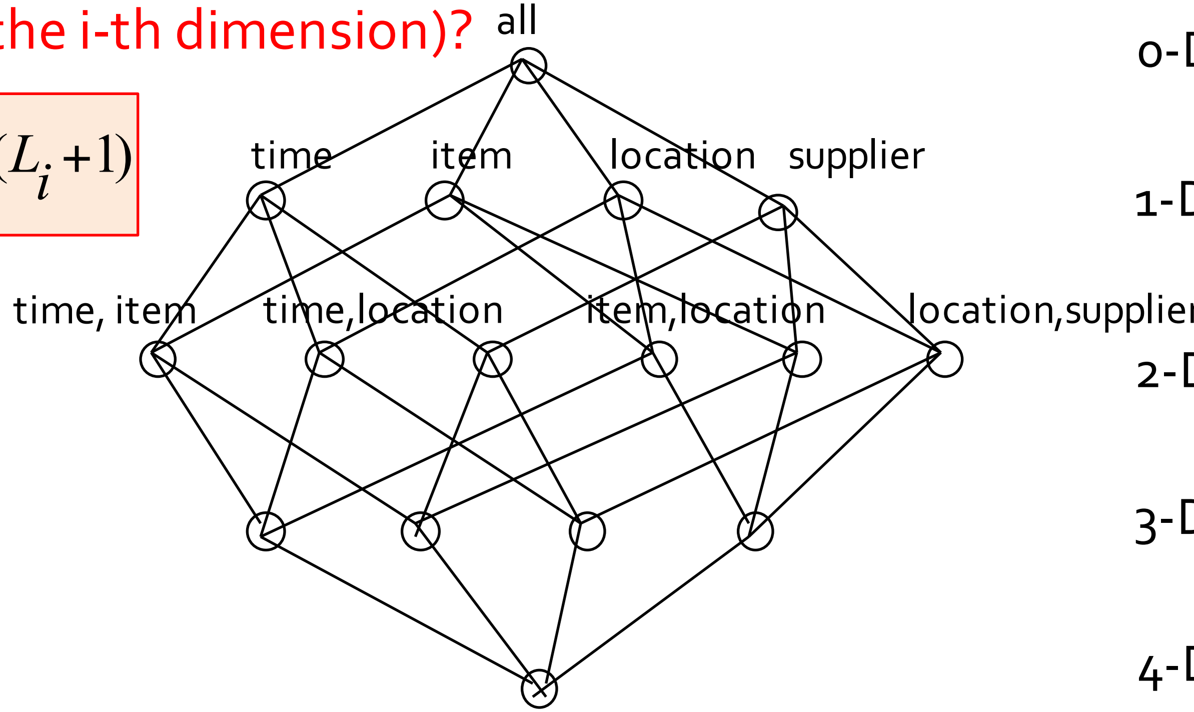
- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L_i levels (at the i-th dimension)?



(N-Dimensional) Data Cube

- 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

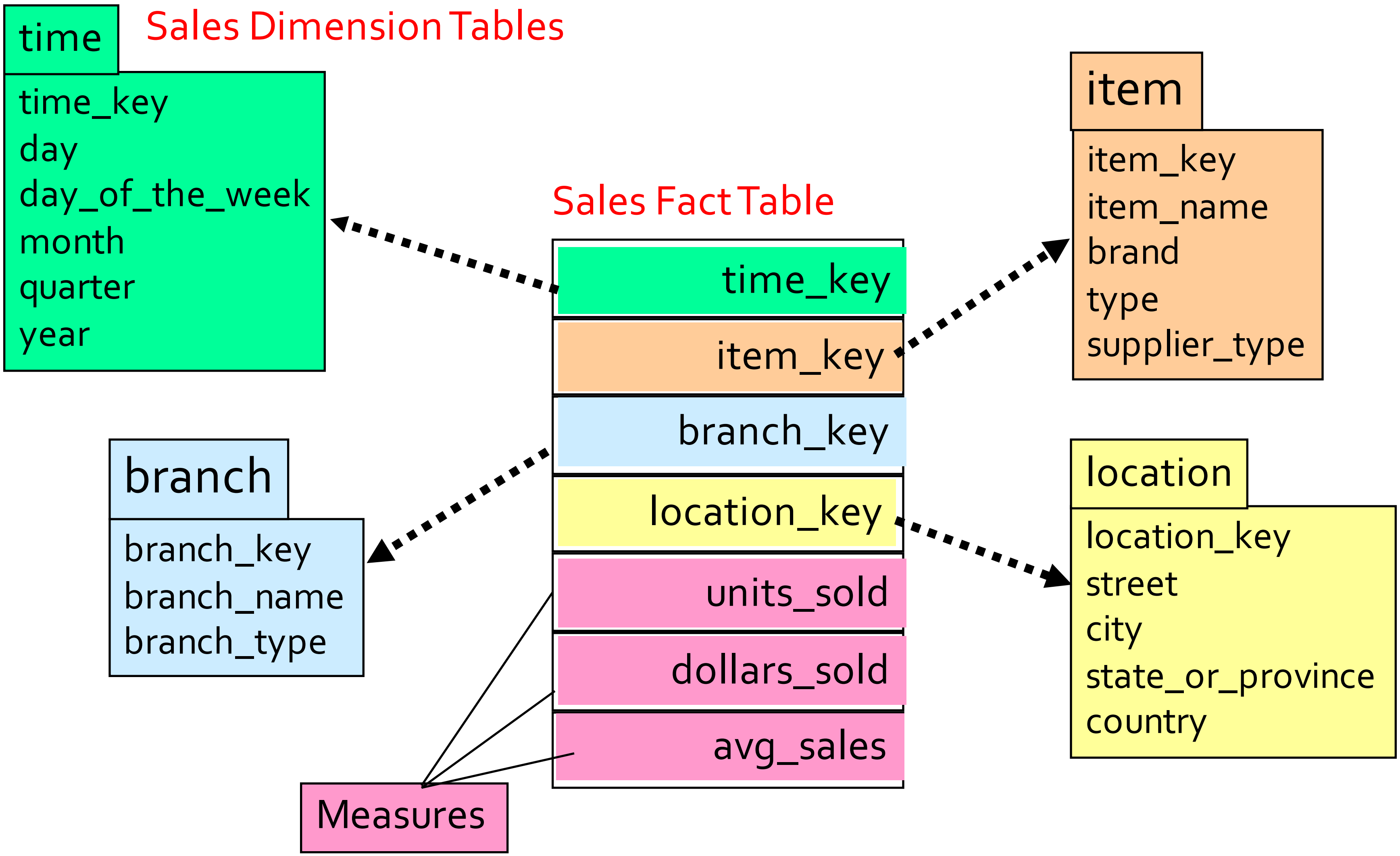
3-D cuboids

4-D (*base*) cuboid

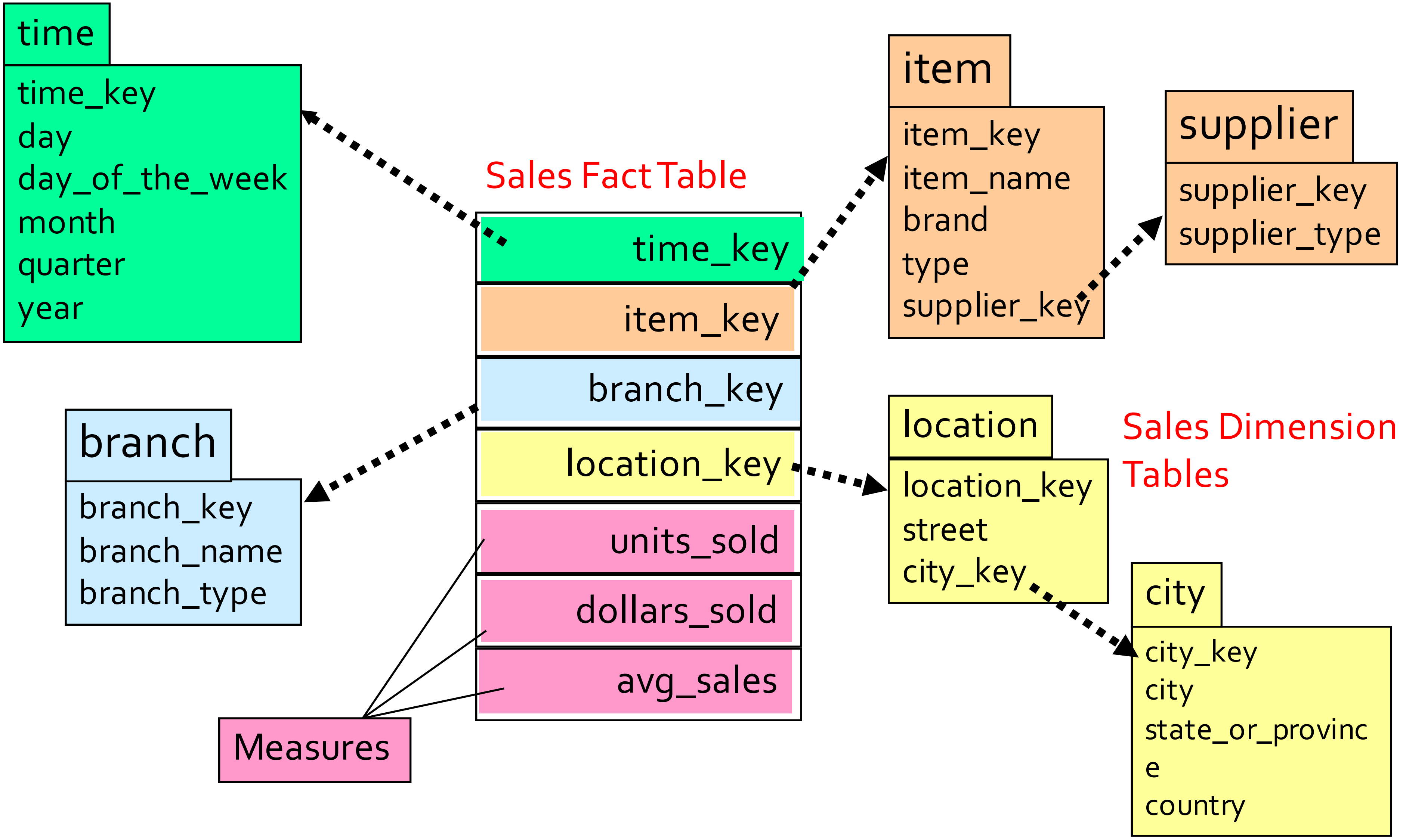
Data Cube: Definition

- **Data cube**: A lattice of cuboids
  - In data warehousing literature, an **n-D base cube** is called a **base cuboid**
  - The top most **0-D cuboid**, which holds the highest-level of summarization, is called the **apex cuboid**
  - The lattice of cuboids forms 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
  - <u>Schemas: Dimension tables and Fact tables</u>

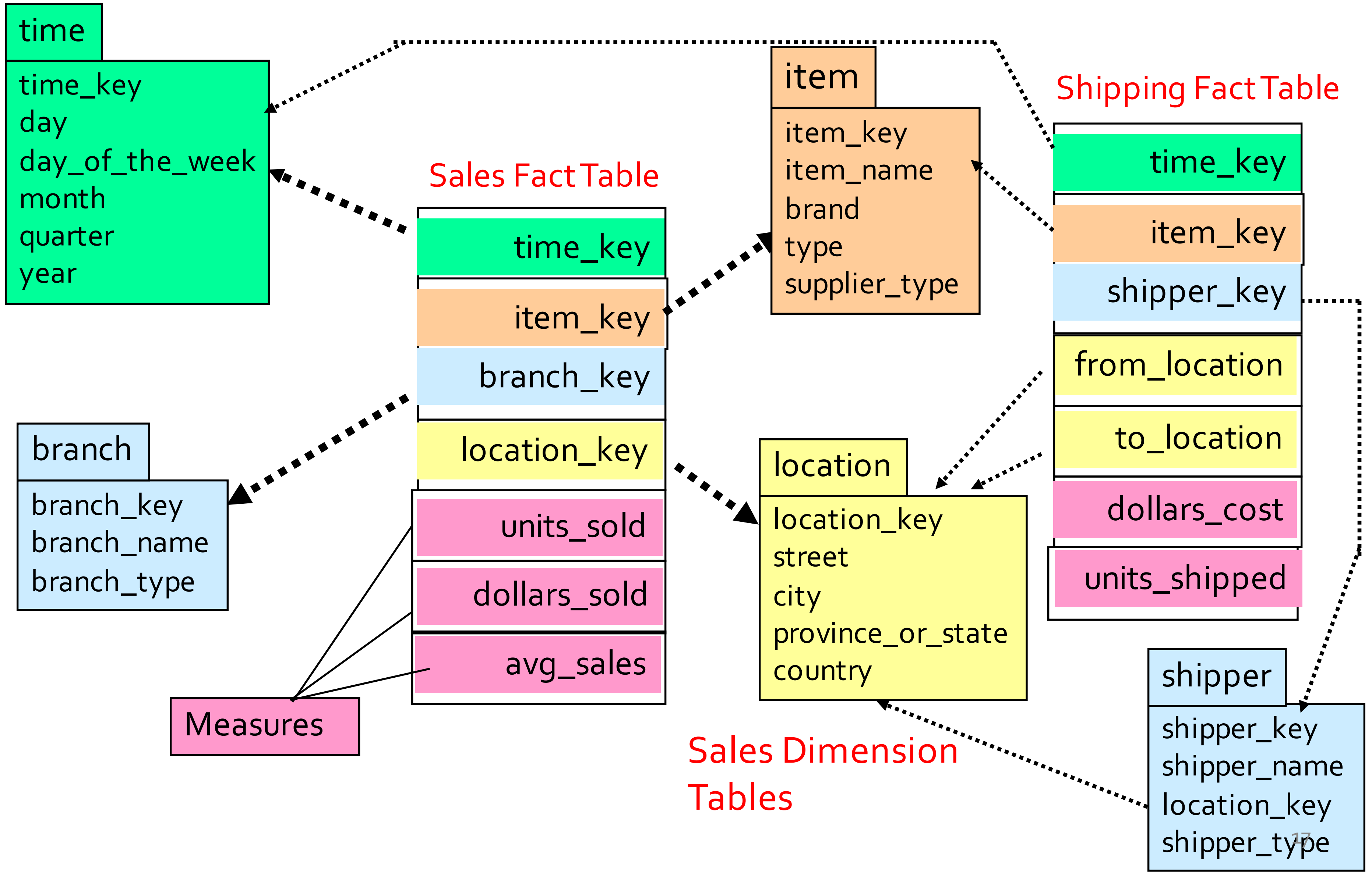
Star Schema


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



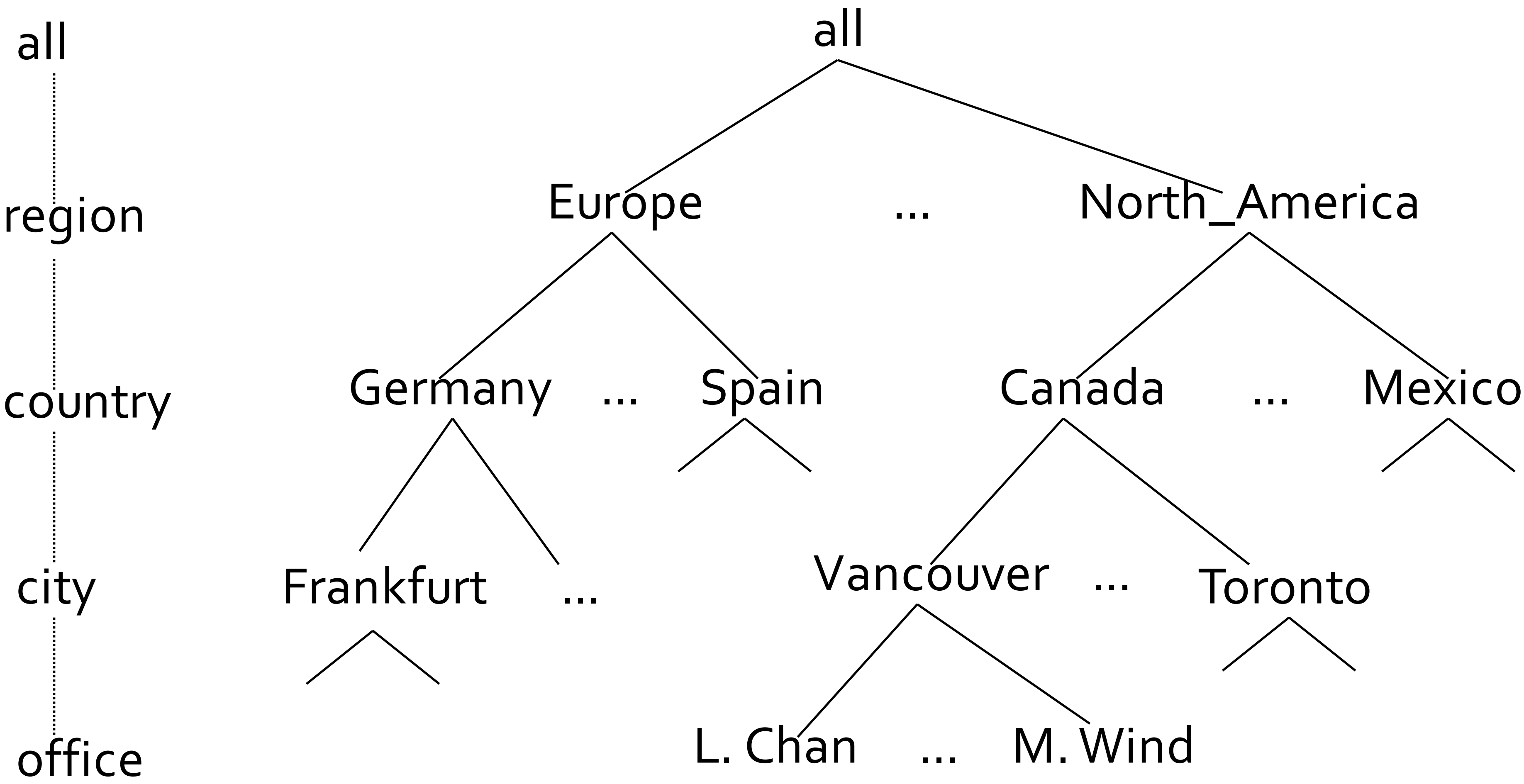
Fact Constellation



Modeling of Data Cubes

- Modeling data cubes: 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

Concept Hierarchy: Dimension Level and Dimension Value

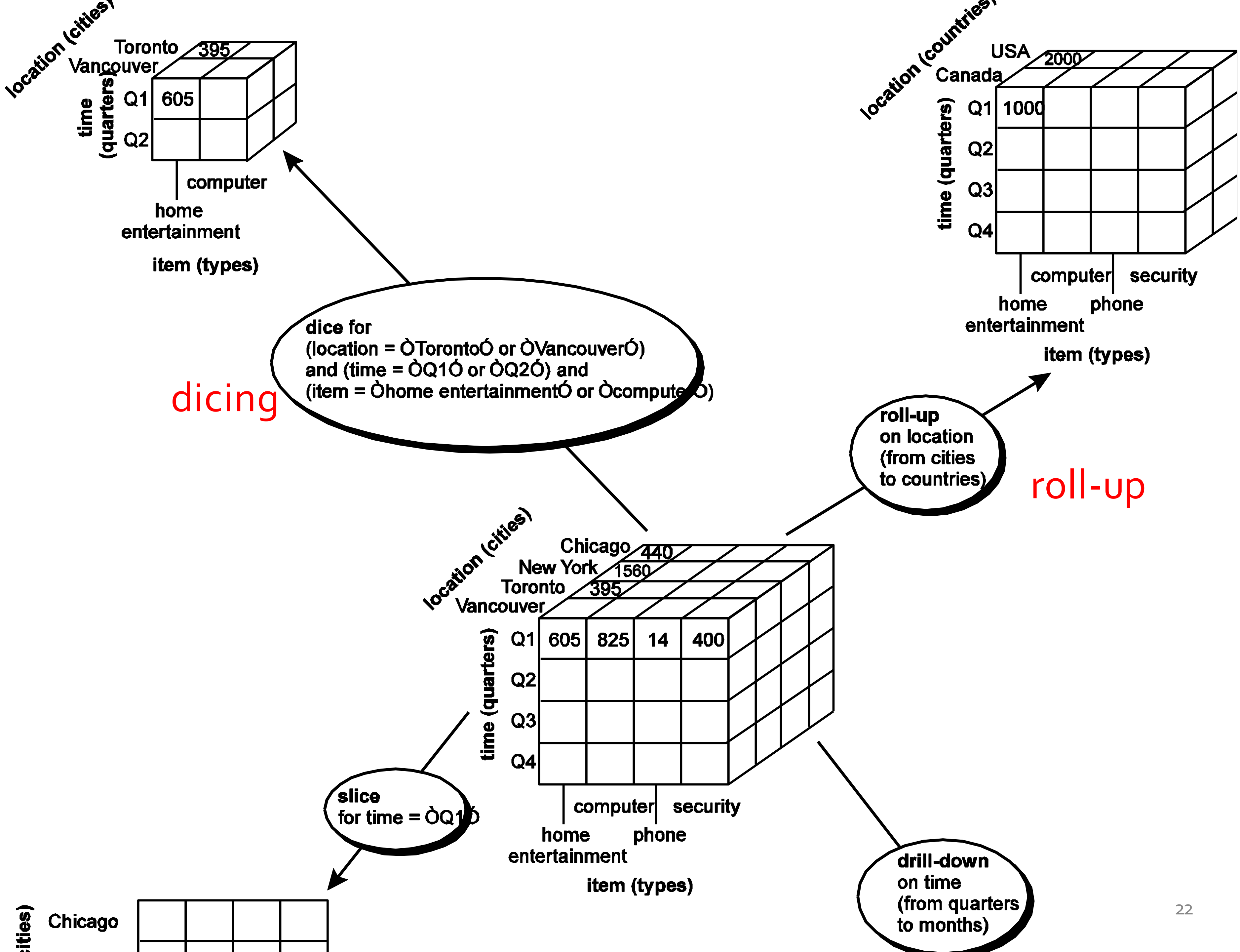


Data Cube Measures: Three Categories

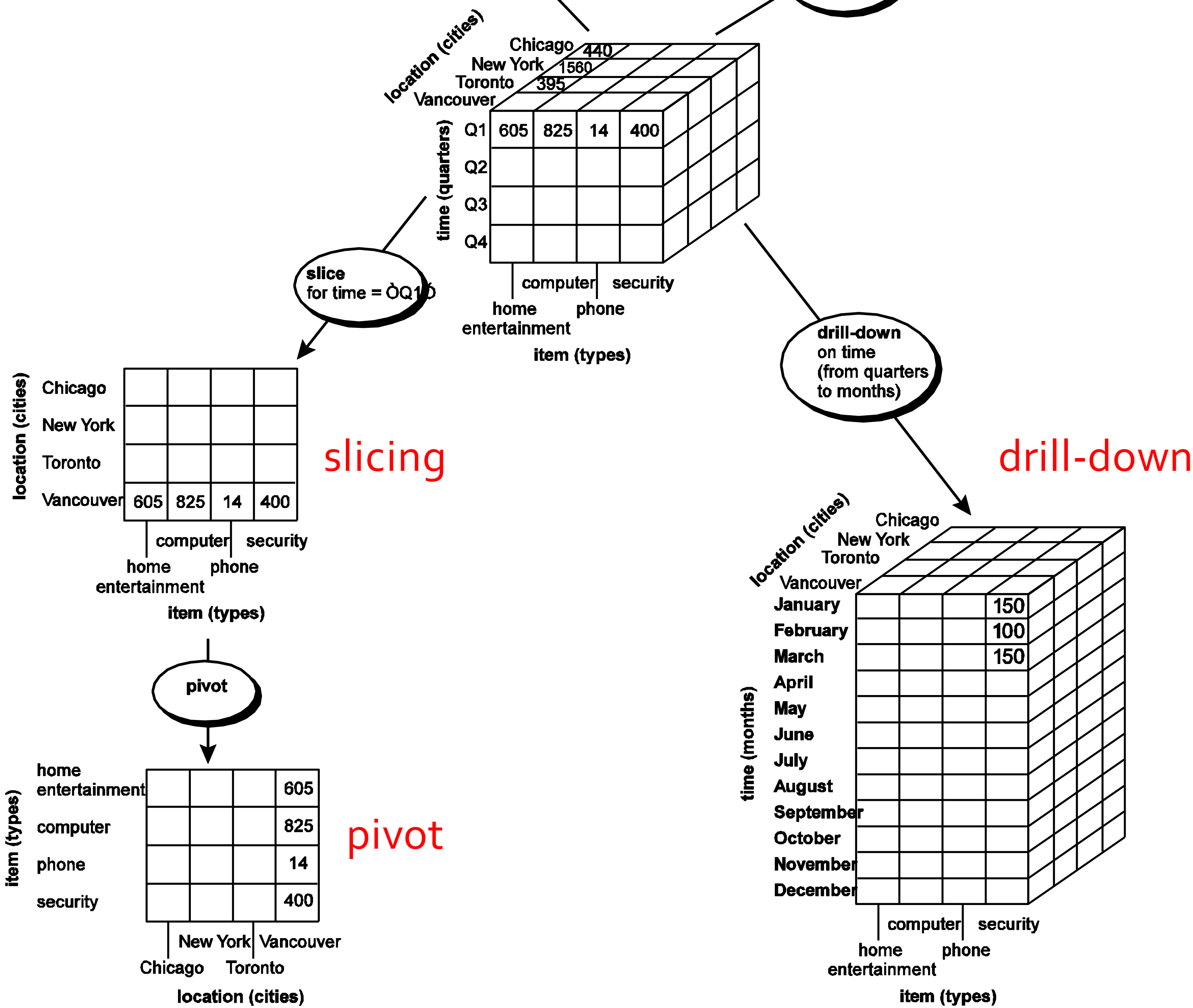
- Distributive: if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning
  - E.g., count(), sum(), min(), max()
- Algebraic: if it can be computed by an algebraic function with *M* arguments (where *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()?

Typical Data Cube 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
395
Vancouver
time (quarters)
Q1
605
Q2
computer
home entertainment
item (types)
location (countries)
USA
2000
Canada
time (quarters)
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 ÒcomputerÓ)
dicing
roll-up
on location
(from cities
to countries)
roll-up
location (cities)
Chicago
440
New York
1560
Toronto
395
Vancouver
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)
slicing
drill-down
pivot
January
February
March
April
May
June
July
August
September
October
November
December
150
100
150
time (months)
location (cities)

The “Compute Cube” Operator

- Cube definition and computation

  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

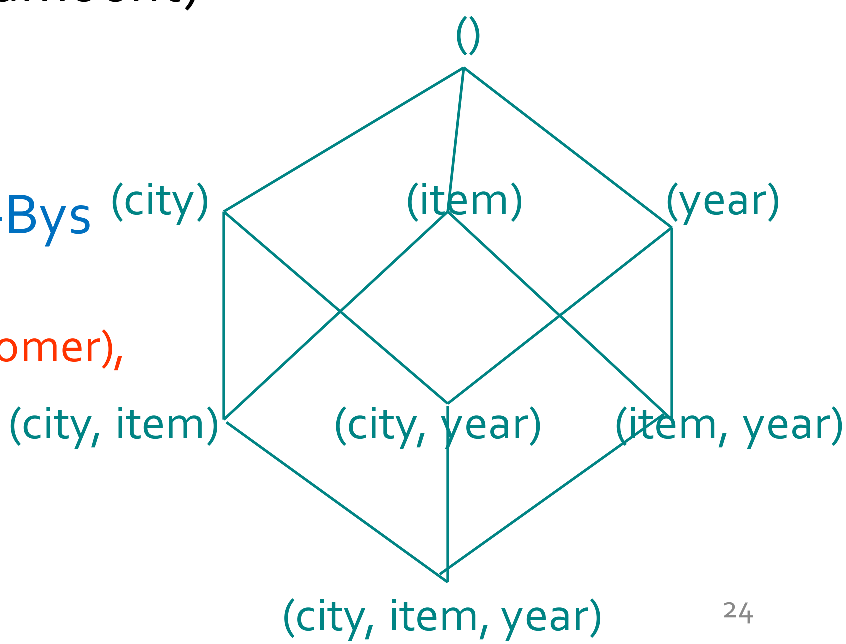
- Need compute the following Group-Bys

(year, product, customer),

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

(year), (product), (customer)

()



Data Cube History

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*

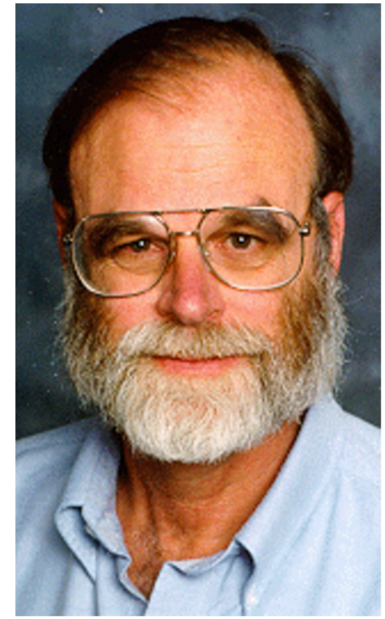
Surajit Chaudhuri *is a computer scientist best known for his contributions to database management systems. He is currently a* ***distinguished scientist at Microsoft Research****, where he leads the Data Management, Exploration and Mining group.*

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 graypro[j@microsoft.com](mailto: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)

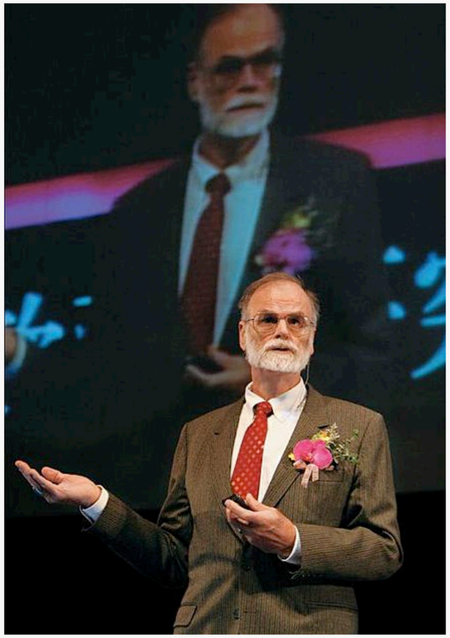
James Nicholas "Jim" Gray (born January 12, 1944; presumed lost at sea January 28, 2007; declared deceased May 16, 2012[4]) was an American computer scientist who received the Turing Award[5] in 1998 "for seminal contributions to database and transaction processing research and technical leadership in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

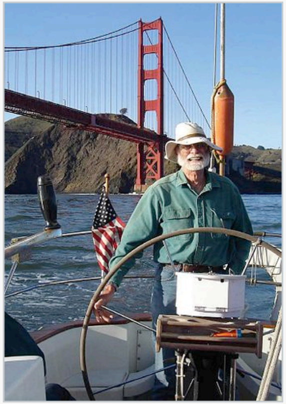
Jim Gray



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

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]



Jim Gray on the *Tenacious* in January 2006

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 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?

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

- 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.

Review: Data Cube

- Concepts
  - Cell, Cuboid, Cube
  - Dimension, Dimension Level, Dimension Value
  - Base/Aggregate Cell/Cuboid
- Components
  - Dimension tables and Fact tables
  - Concept hierarchy and Measures
  - Schemas
- Operations
- Materialization

Partial materialization:

Which cuboids to materialize?

Q: What do they hate the most?

Iceberg

Cube Materialization: Full Cube vs. Iceberg Cube

- Full cube vs. iceberg cube

  compute cube sales iceberg as

  select date, product, city, department, count(*)

  from salesInfo

  cube by date, product, city

  having count(*) >= min support
- Compute *only* the **cells** whose **measure** satisfies the iceberg condition
- Only a small portion of cells may be "above the water" in a sparse cube
- Ex.: Show only those cells whose **count** is no less than 100

Why Iceberg Cube?

- Advantages of computing iceberg cubes
  - No need to save nor show those cells whose value is below the threshold (iceberg condition)
  - Efficient methods may even avoid computing the un-needed, intermediate cells
  - Avoid explosive growth
- Example: A cube with 100 dimensions
  - Suppose it contains only 2 base cells and the count of each cell is 1:
    - $\{(a_1, a_2, a_3, ..., a_{100}) : 1, (a_1, a_2, b_3, ..., b_{100}) : 1\}$
  - How many **aggregate cells** if "having count >= 1" (**non-empty**)?
  - What are the **iceberg cells** with condition "having count >= 2"?

Suppose it contains only 2 base cells: $\{(a_1, a_2, a_3, \ldots, a_{100}), (a_1, a_2, b_3, \ldots, b_{100})\}$

How many non-empty aggregate cells?

For $\{(a_1, a_2, a_3 \ldots, a_{100}), (a_1, a_2, b_3, \ldots, b_{100})\}$, the total # of non-base cells should be 2 * (2^{100} – 1) – 4.

This is calculated as follows:

- (a1, a2, a3 . . . , a100) will generate 2^{100} - 1 non-base cells
- (a1, a2, b3, . . . , b100) will generate 2^{100} - 1 non-base cells

Among these, 4 cells are overlapped and thus minus 4 so we get:

2*2^{100} - 2 – 4

These 4 cells are:

- (a1, a2, *, …, *): 2
- (a1, *, *, …, *): 2
- (*, a2, *, …, *): 2
- (*, *, *, …, *): 2

Is Iceberg Cube Good Enough? Closed Cube & Cube Shell

- Let cube P have only 2 base cells: $\{(a_1, a_2, a_3 \ldots, a_{100}):10, (a_1, a_2, b_3, \ldots, b_{100}):10\}$
  - How many cells will the iceberg cube contain if "having count(*) ≥ 10"?
    - Answer: $2^{101} - 4$ (base+aggregate; still too big!)
- **Close cube:**
  - A cell c is ***closed*** if there exists no cell *d*, such that *d* is a descendant of *c*, and *d* has the same measure value as *c*
    - Ex. The same cube P has only 3 closed cells:
  - $\{(a_1, a_2, *, \ldots, *): 20, (a_1, a_2, a_3 \ldots, a_{100}): 10, (a_1, a_2, b_3, \ldots, b_{100}): 10\}$
  - A ***closed cube*** is a cube consisting of only closed cells

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
- K. Beyer and R. Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg CUBEs.. SIGMOD'99
- J. Han, J. Pei, G. Dong, K. Wang. Efficient Computation of Iceberg Cubes With Complex Measures. SIGMOD'01
- L. V. S. Lakshmanan, J. Pei, and J. Han, Quotient Cube: How to Summarize the Semantics of a Data Cube, VLDB'02
- X. Li, J. Han, and H. Gonzalez, High-Dimensional OLAP: A Minimal Cubing Approach, VLDB'04
- X. Li, J. Han, Z. Yin, J.-G. Lee, Y. Sun, "Sampling Cube: A Framework for Statistical OLAP over Sampling Data", SIGMOD'08
- K. Ross and D. Srivastava. Fast computation of sparse datacubes. VLDB'97
- D. Xin, J. Han, X. Li, B. W. Wah, Star-Cubing: Computing Iceberg Cubes by Top-Down and Bottom-Up Integration, VLDB'03
- Y. Zhao, P. M. Deshpande, and J. F. Naughton. An array-based algorithm for simultaneous multidimensional aggregates. SIGMOD'97
- D. Burdick, P. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. OLAP over uncertain and imprecise data. VLDB'05

References (cont.)

- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. ICDE'97
- B.-C. Chen, L. Chen, Y. Lin, and R. Ramakrishnan. Prediction cubes. VLDB'05
- B.-C. Chen, R. Ramakrishnan, J.W. Shavlik, and P. Tamma. Bellwether analysis: Predicting global aggregates from local regions. VLDB'06
- Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang, Multi-Dimensional Regression Analysis of Time-Series Data Streams, VLDB'02
- R. Fagin, R. V. Guha, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins. Multi-structural databases. PODS'05
- J. Han. Towards on-line analytical mining in large databases. SIGMOD Record, 27:97–107, 1998
- T. Imielinski, L. Khachiyan, and A. Abdulghani. Cubegrades: Generalizing association rules. Data Mining & Knowledge Discovery, 6:219–258, 2002.
- R. Ramakrishnan and B.-C. Chen. Exploratory mining in cube space. Data Mining and Knowledge Discovery, 15:29–54, 2007.
- K. A. Ross, D. Srivastava, and D. Chatziantoniou. Complex aggregation at multiple granularities. EDBT'98
- S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. EDBT'98
- G. Sathe and S. Sarawagi. Intelligent Rollups in Multidimensional OLAP Data. *VLDB'01*