CS 412 Intro. to Data Mining

Chapter 4. Data Warehousing and On-line Analytical Processing

Qi Li, Computer Science, Univ. Illinois at Urbana-Champaign, 2018

Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Implementation
- Summary

What is a Data Warehouse?

- Defined in many different ways, but not rigorously
  - Support decision
  - Maintained Separately
  - Information processing
- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."—W. H. Inmon
- Data warehousing:
  - The process of constructing and using data warehouses

Data Warehouse—Subject-Oriented

- Help make decisions
  - A **simple** and **concise** view (modeling and analysis)
  - Not details (transaction processing)
  - Organizing around major subjects, such as customer, product, sales
  - Excluding data that are not useful in the decision support process

Data Warehouse—Integrated

- Integrating Multiple, heterogeneous sources
  - Ex. relational databases, flat files, on-line transaction records
- Consistency
  - Data cleaning and data integration techniques are applied.
  - Ex. Hotel price: differences on currency, tax, breakfast covered, and parking
  - When data is moved to the warehouse, it is converted

Data Warehouse—Time Variant

Data Warehouse	Operational Database
Long time horizon (e.g., past 5-10 years)	current value data
Contains an element of time, explicitly or implicitly	data may or may not contain “time element”

Data Warehouse—Nonvolatile

- Independence – A physically separate store
- Static – No data management (updates, transaction processing, recovery, and concurrency control mechanisms)
- Requires only two operations in data accessing:
  - *initial loading of data* and *access of data*

Why a Separate Data Warehouse?

- Different functions and different data:
  - missing data: Decision support requires historical data which operational DBs do not typically maintain
  - data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  - data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled
- Note: There are more and more systems which perform OLAP (online analytical processing) analysis directly on relational databases

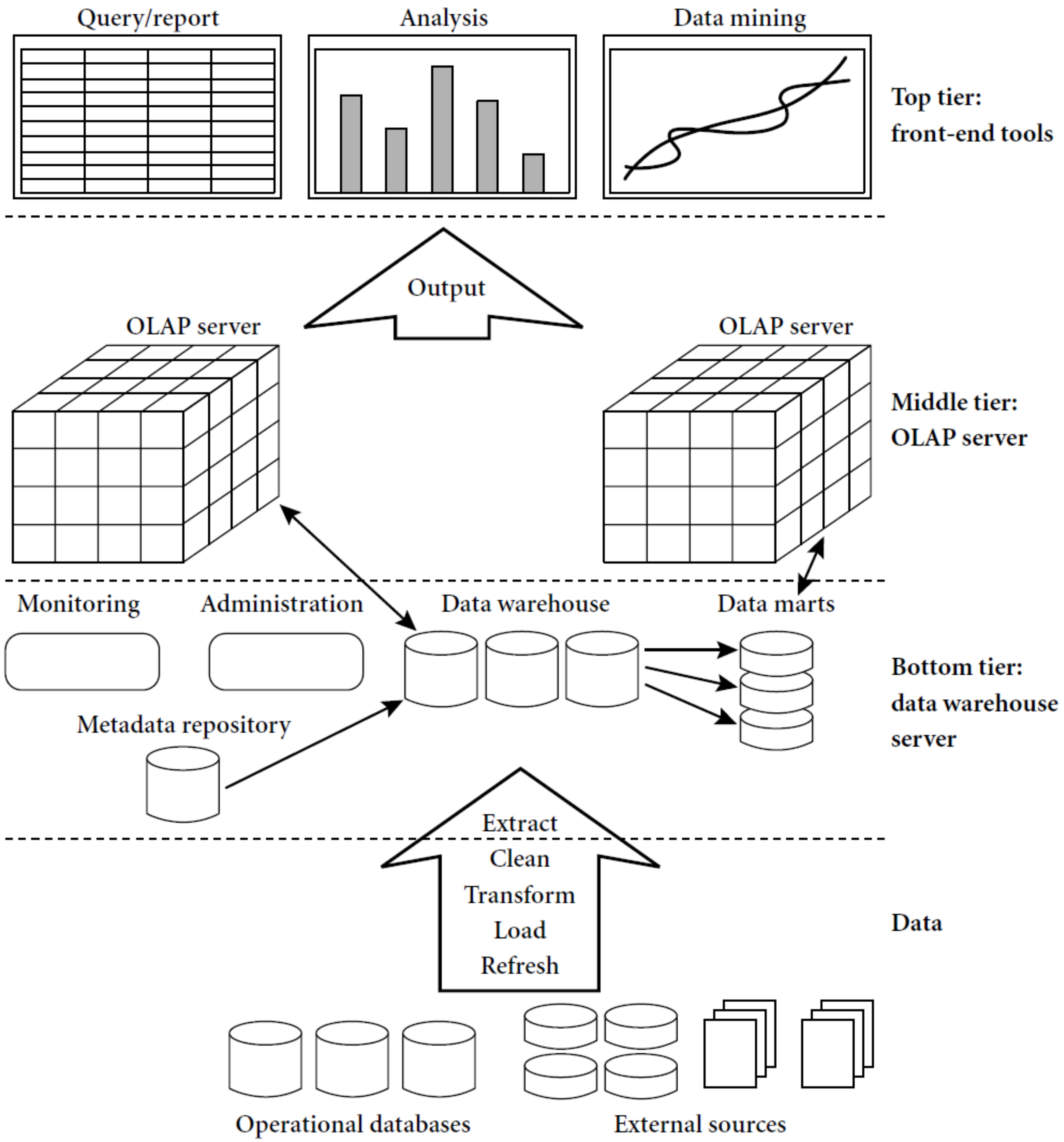
OLTP vs. OLAP

- OLTP: Online transactional processing
  - DBMS operations
  - Query and transactional processing
- OLAP: Online analytical processing
  - Data warehouse operations
  - Drilling, slicing, dicing, etc.

	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

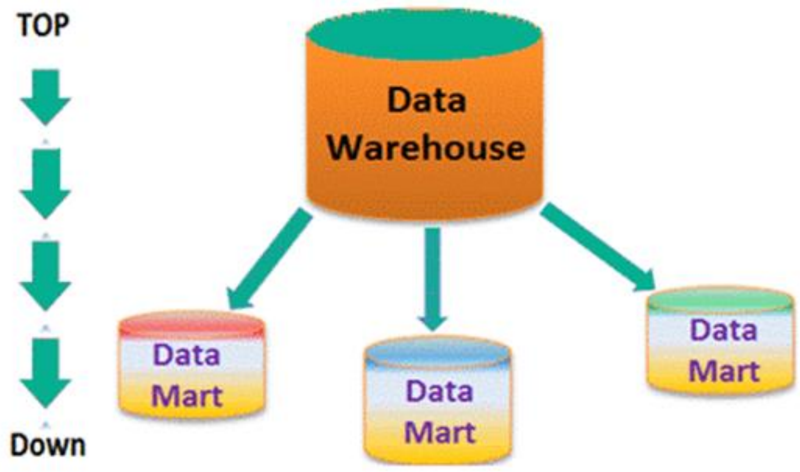
Data Warehouse: A Multi-Tiered Architecture

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



Three Data Warehouse Models

- **Enterprise warehouse -** Specially designed for the entire organization
- **Data Mart**
  - Specific, selected groups
  - Independent vs. dependent (directly from warehouse) data mart
- **Virtual warehouse**
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized



https://www.guru99.com/data-warehouse-vs-data-mart.html

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 indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

Metadata Repository

- **Meta data** is data about data. It stores:
  - Description of structure (schema, etc.)
  - 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

Chapter 4: Data Warehousing and On-line Analytical Processing

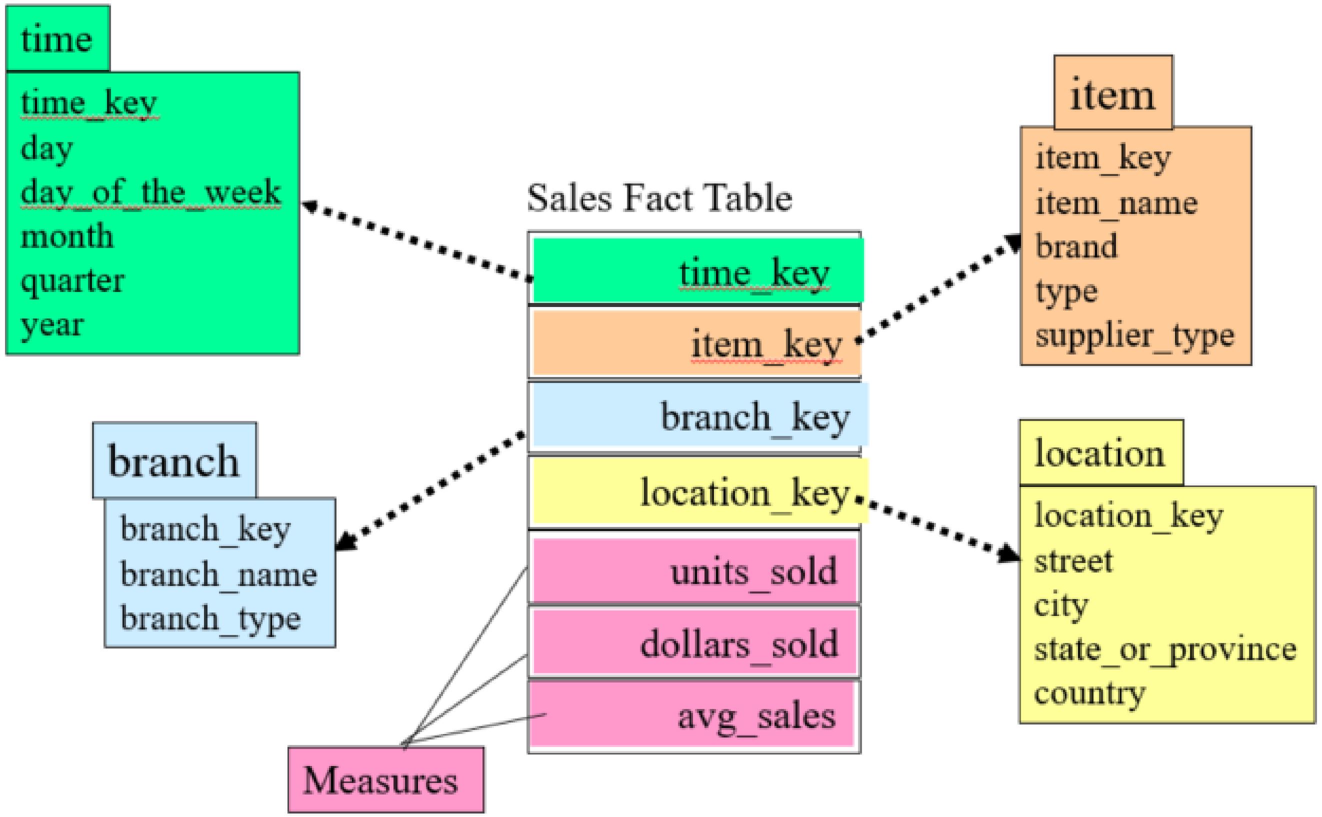
- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Implementation
- Summary

From Tables and Spreadsheets to Data Cubes

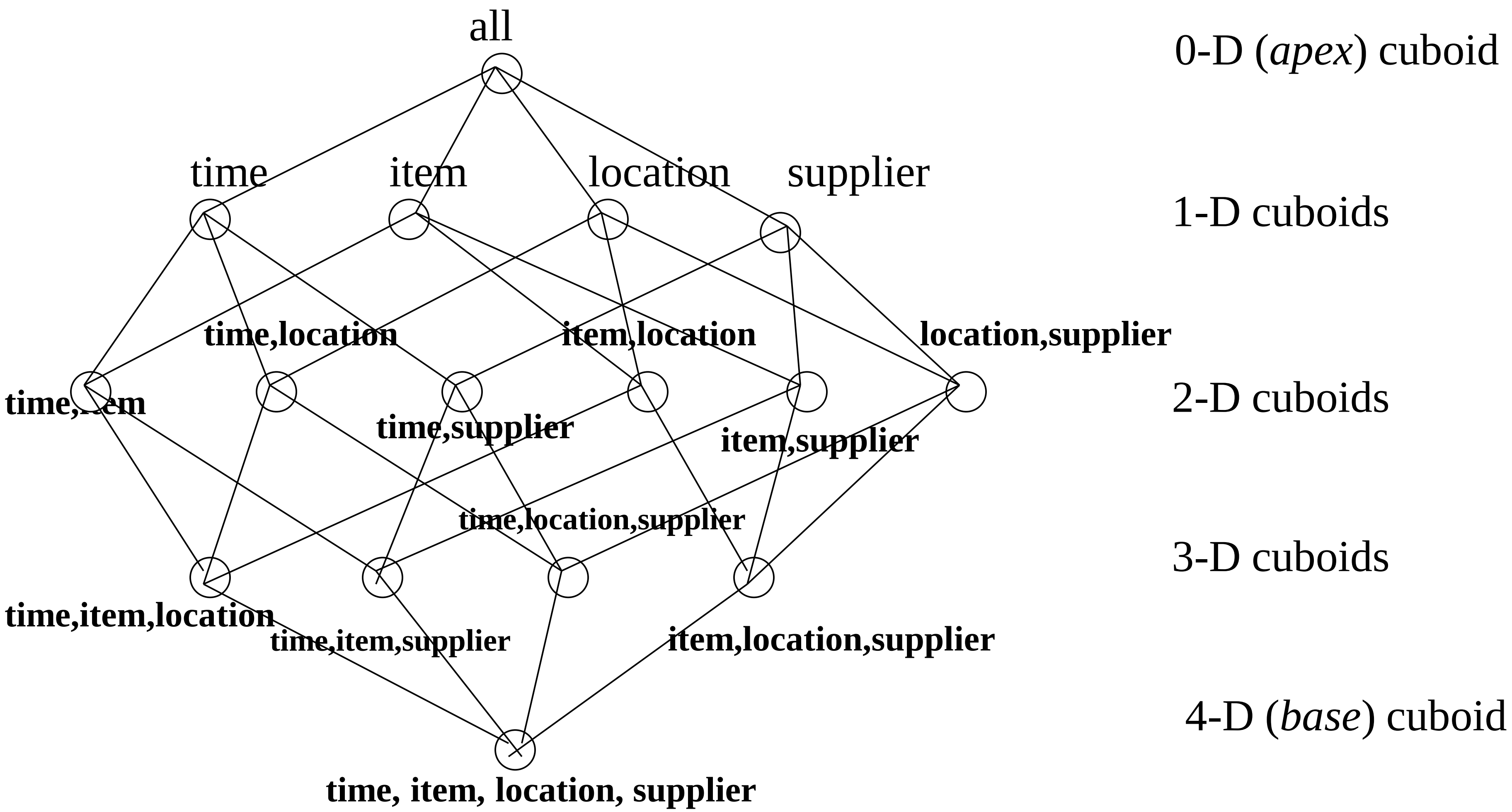
- A **data warehouse** is based on a multidimensional data model which views data in the form of a data cube
- Main function is to provide summarizations of the data
  - E.g., summarize the units or dollars sold at a particular store over a particular time period
- Can compute summarizations online (as they are requested)
  - Can be very slow
- Better to precalculate some summarizations

Design of Data Warehouses

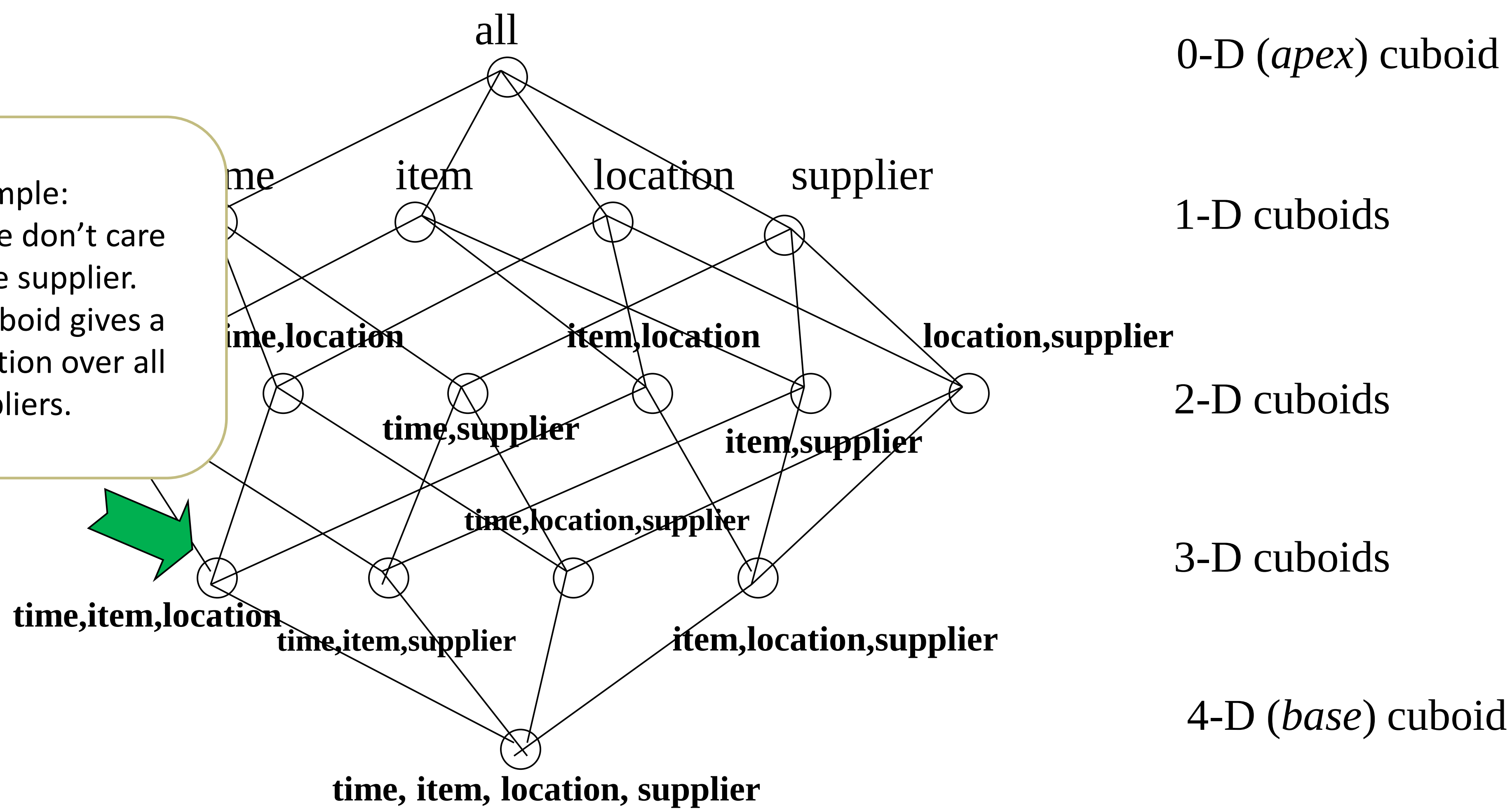
- **Dimension tables**, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
- **Fact table** contains **measures** (such as dollars_sold) and keys to each of the related dimension tables
- Different schema exist
  - Star
  - Snowflake
  - Fact constellation



Data Cube: A Lattice of Cuboids

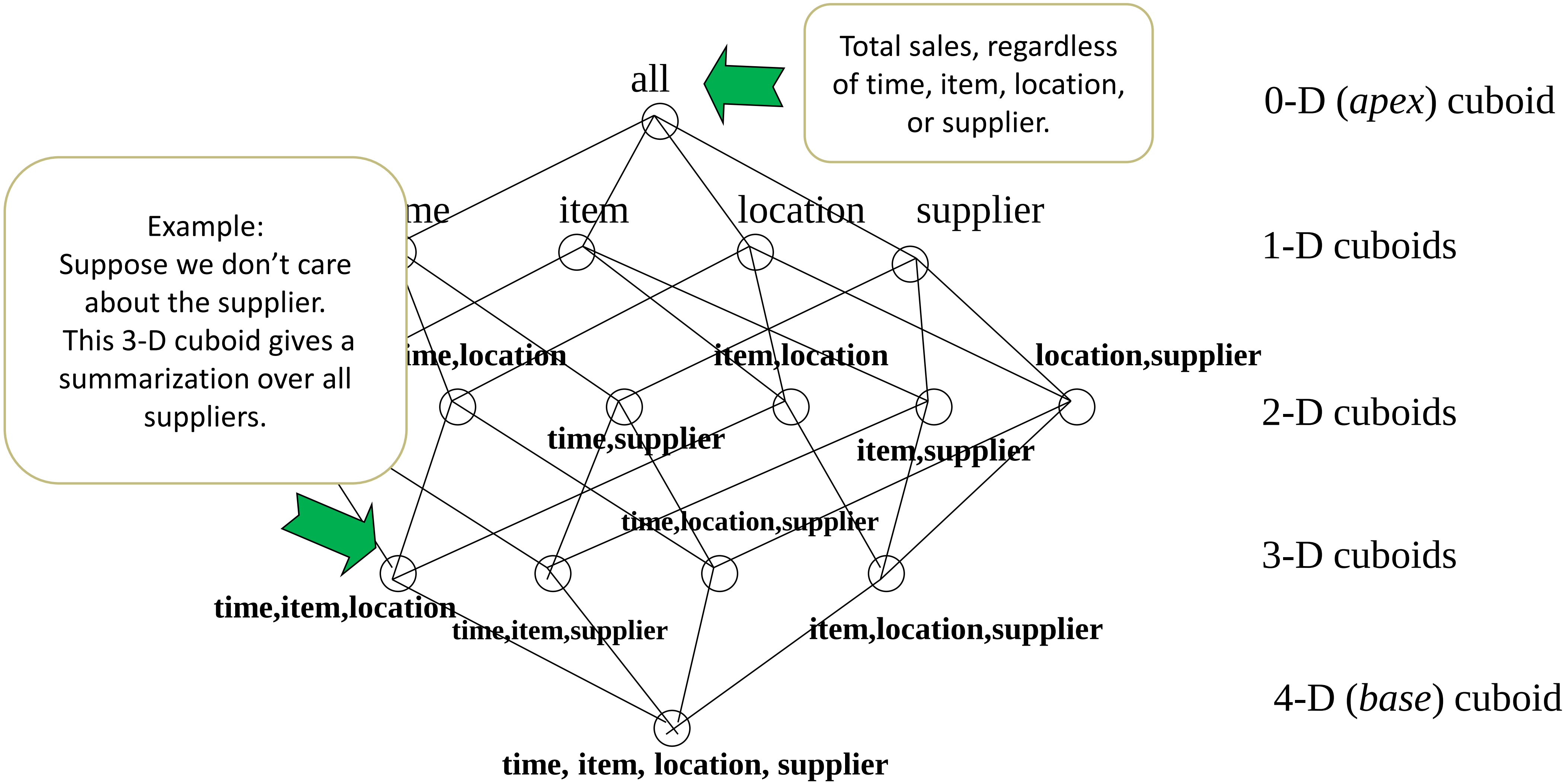


Data Cube: A Lattice of Cuboids



Example:
Suppose we don't care about the supplier.
This 3-D cuboid gives a summarization over all suppliers.

Data Cube: A Lattice of Cuboids



Calculating Number of Cuboids

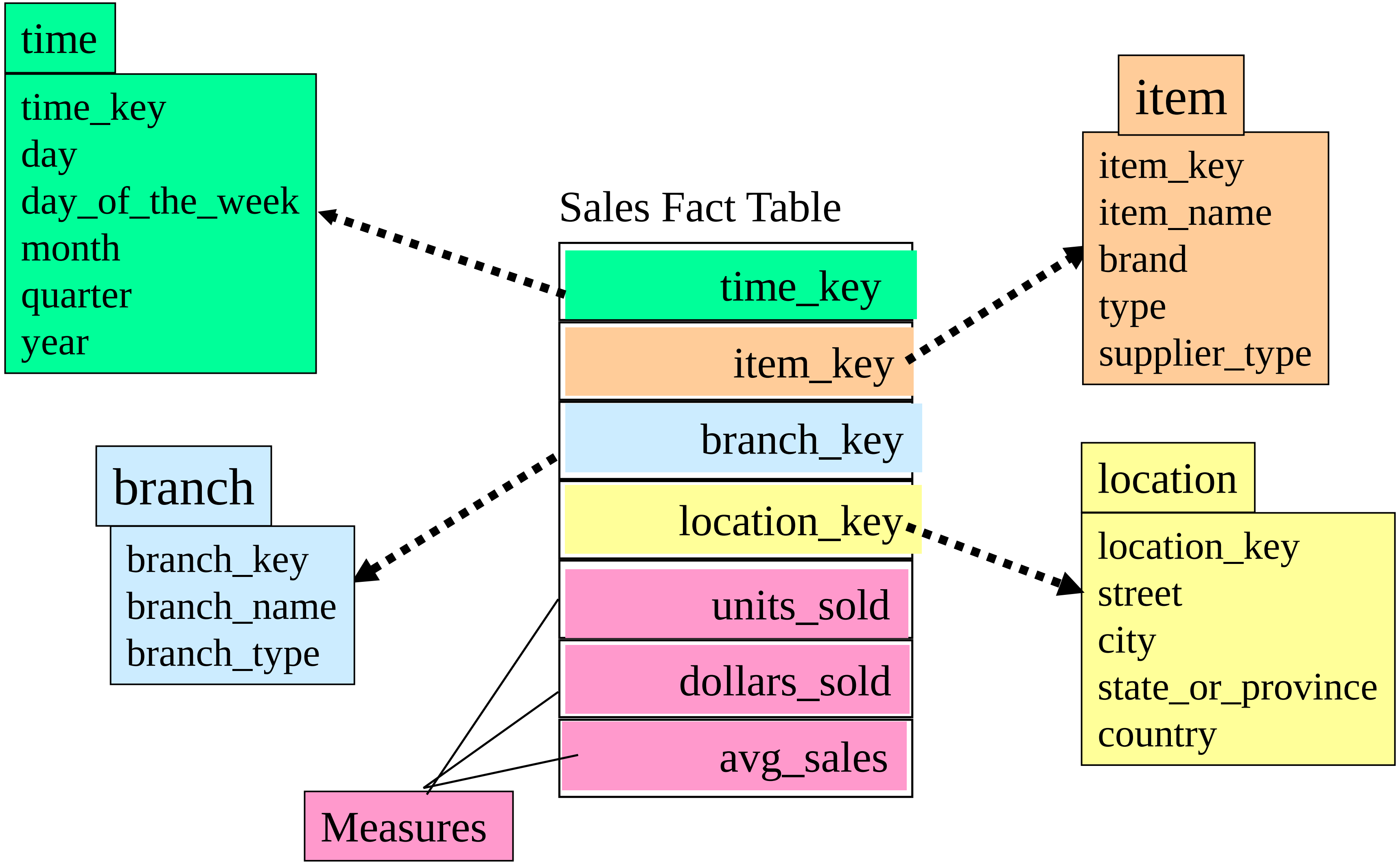
- Consider dimensions as binary numbers
- Example: 4 dimensions
  - Each is either in the cuboid, or not in the cuboid
  - (,   ,   ,   ) ← choice of 0 or 1 for each element of vector
  - Sum up for each position: $2^3 + 2^2 + 2^1 + 2^0 + 1$ (0-d cuboid) = 2^4
- In general, 2^d cuboids (d = number of dimensions)

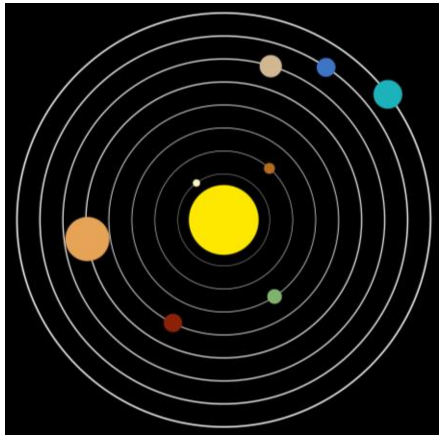
Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - Star schema
  - Snowflake schema
  - Fact constellations

Star Schema: An Example

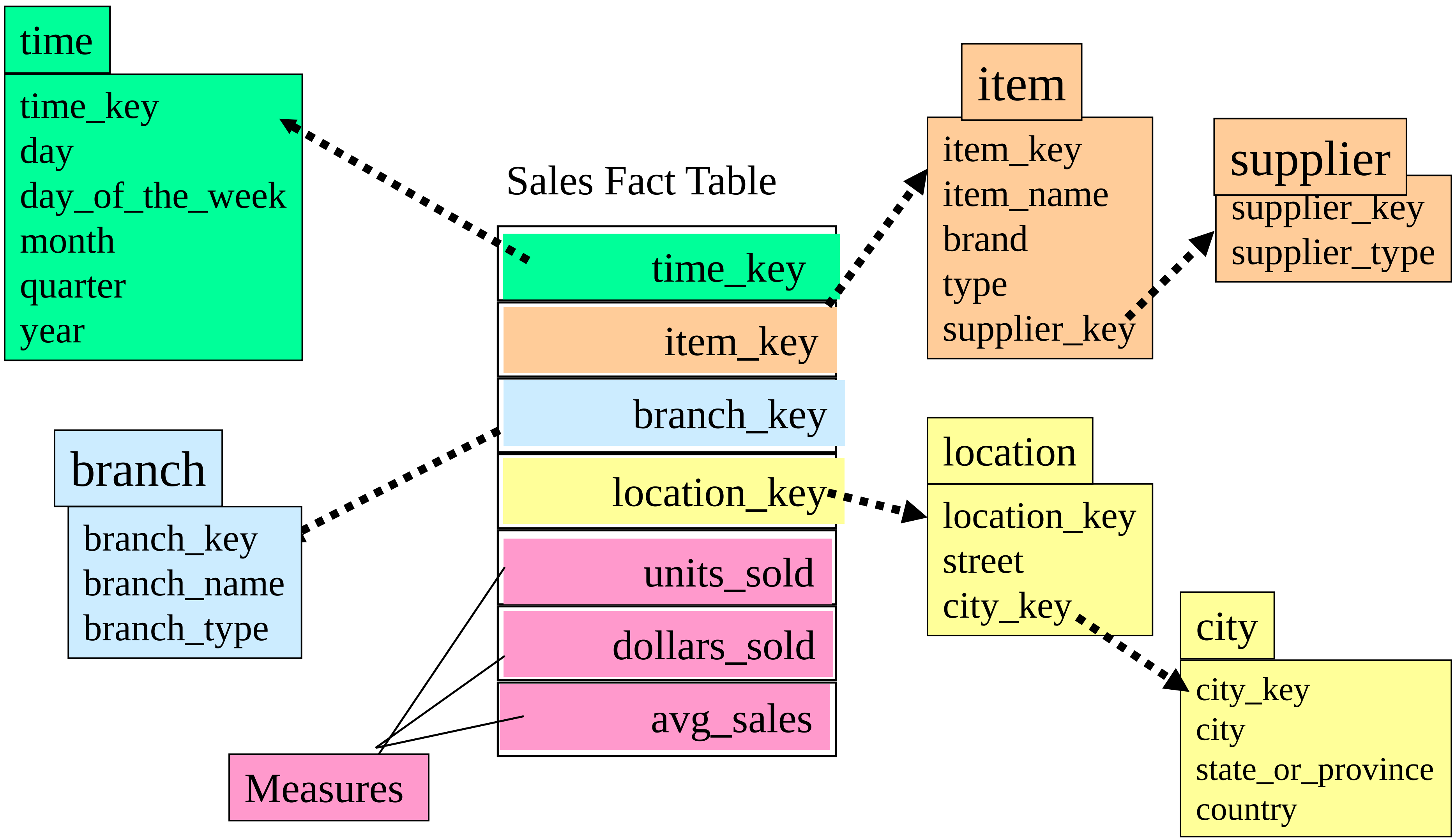
A fact table in the middle connected to a set of dimension tables





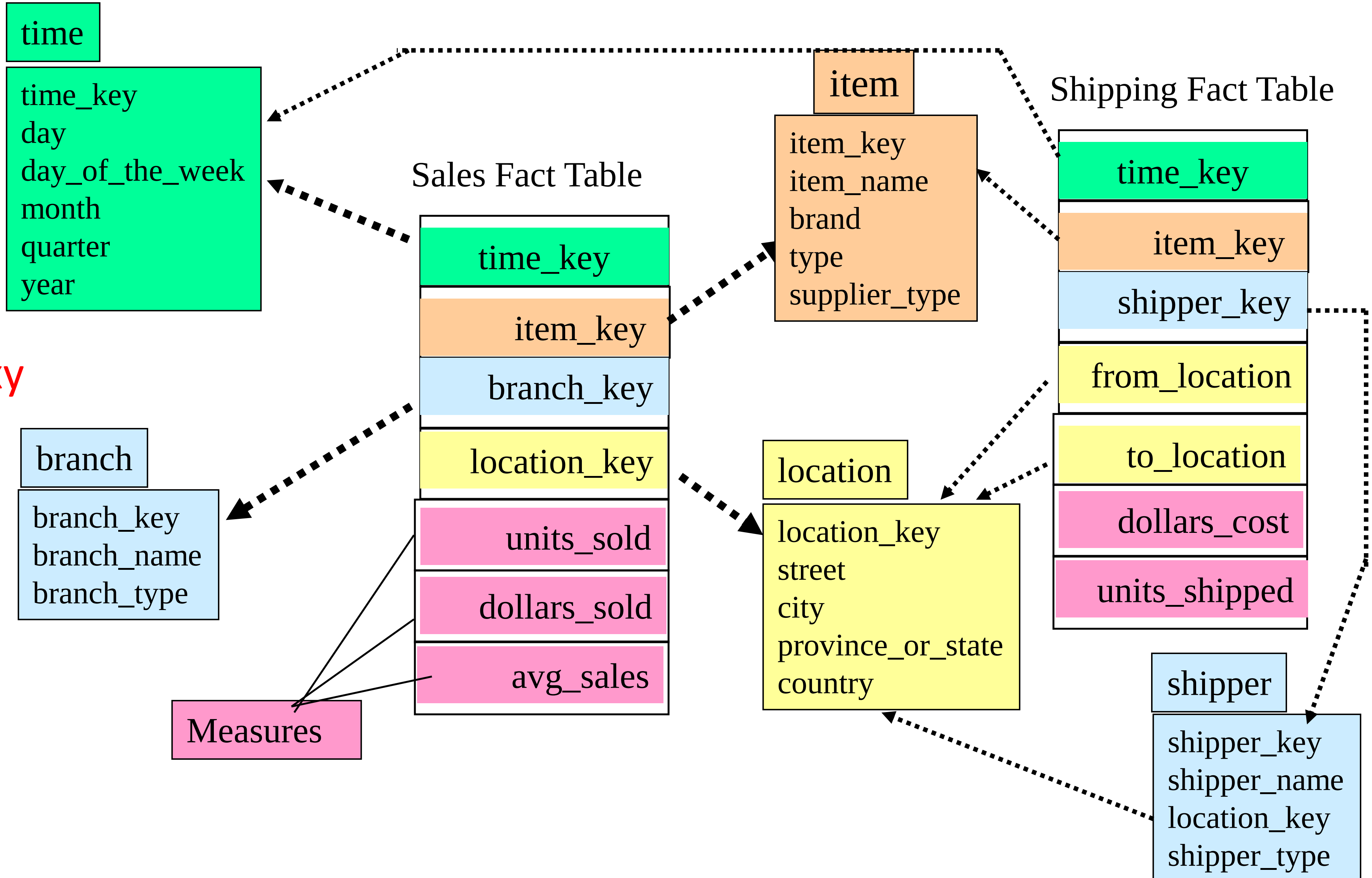
Snowflake Schema: An Example

A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake

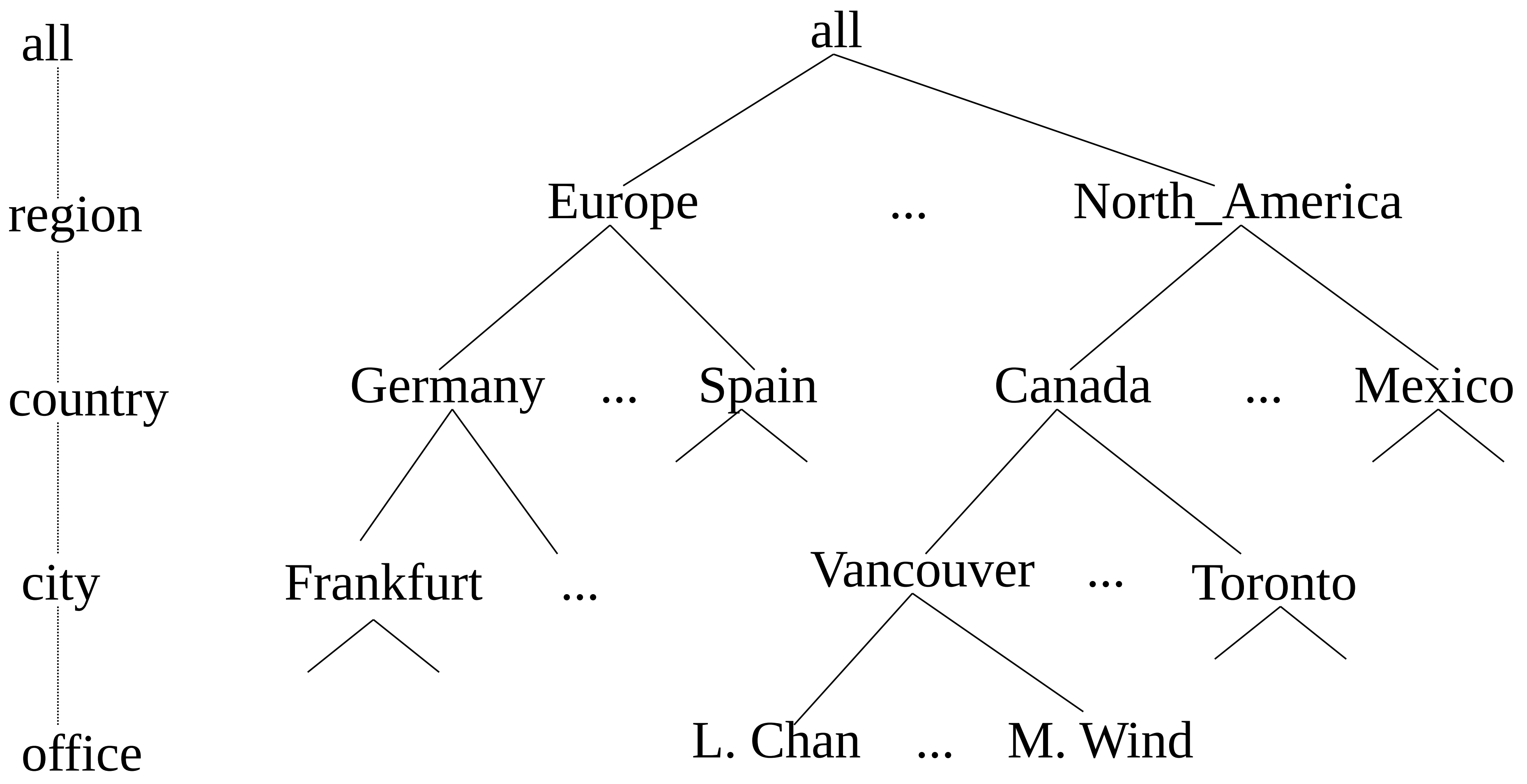


Fact Constellation: An Example

Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

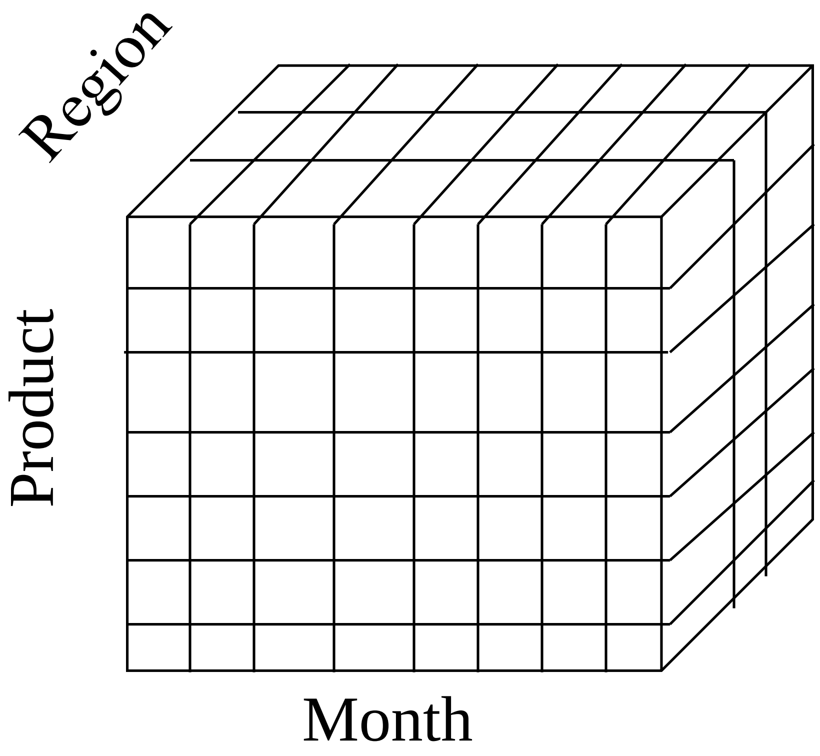


A Concept Hierarchy for a Dimension (location)

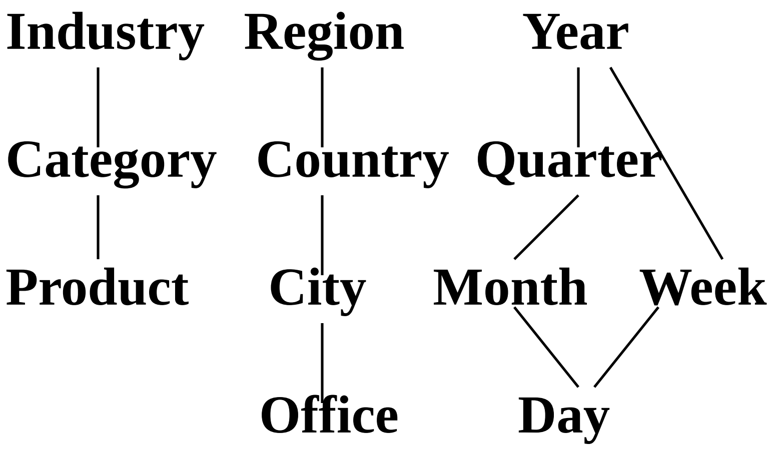


Multidimensional Data

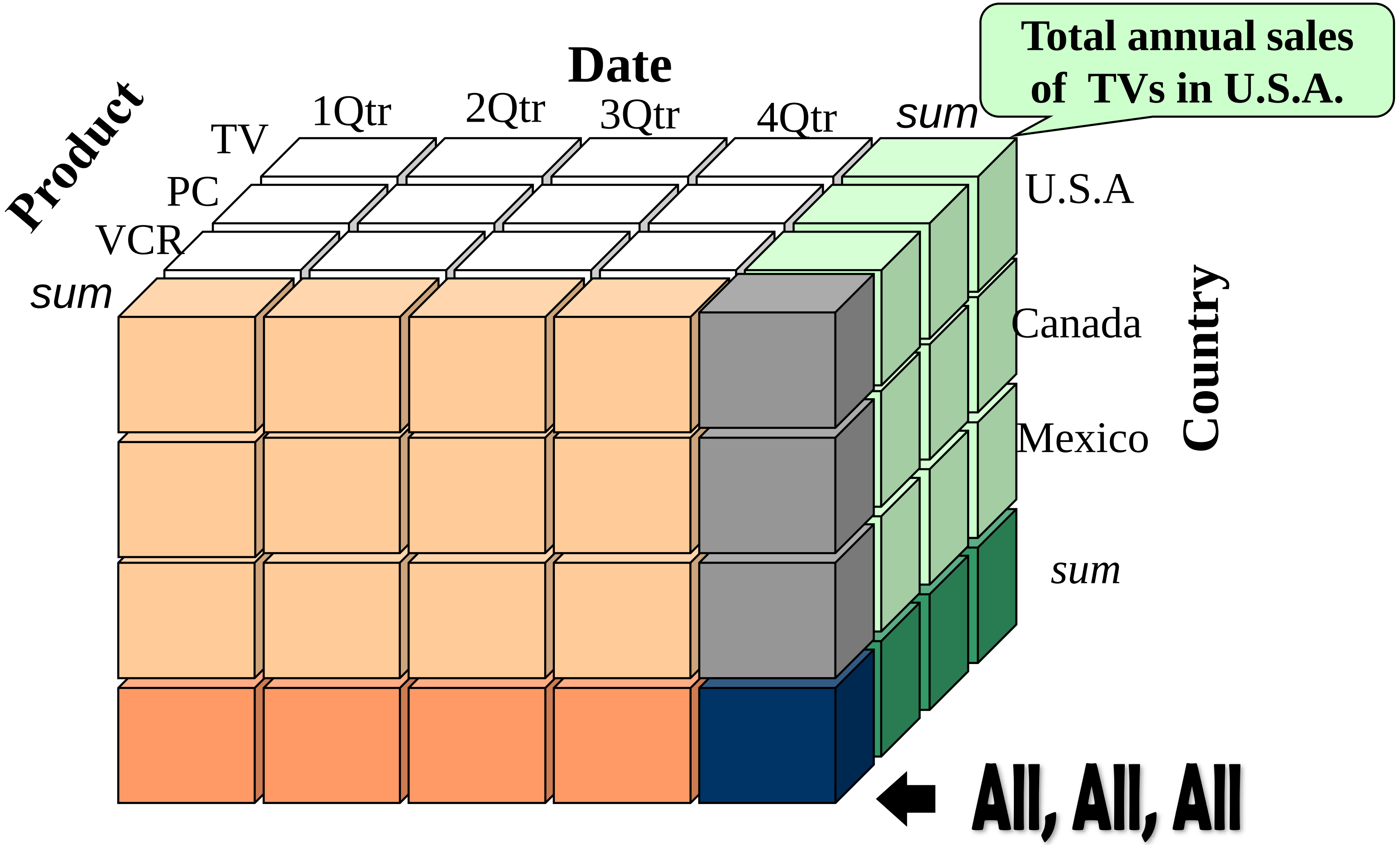
- Sales volume as a function of product, month, and region

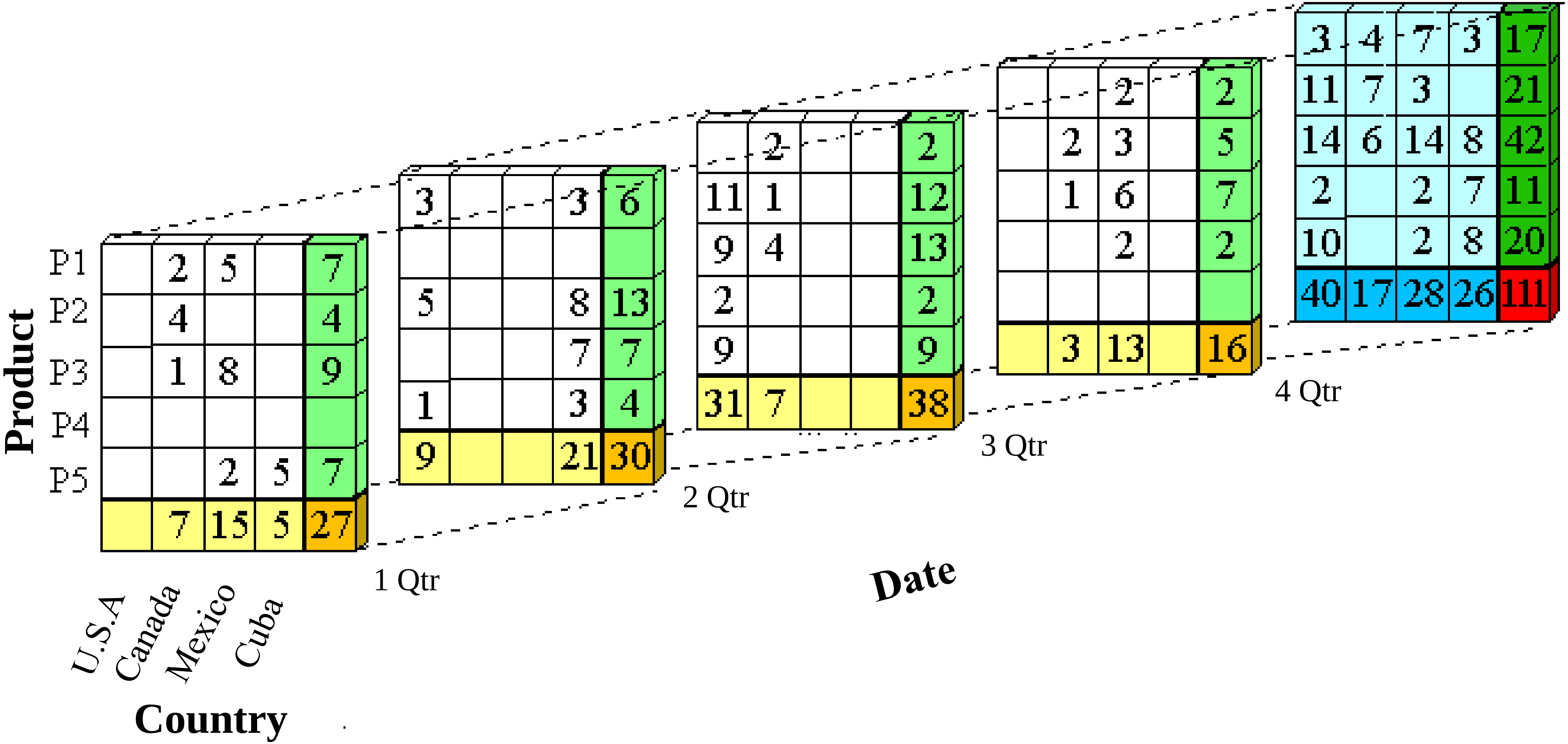


Dimensions: *Product, Location, Time*
Hierarchical summarization paths



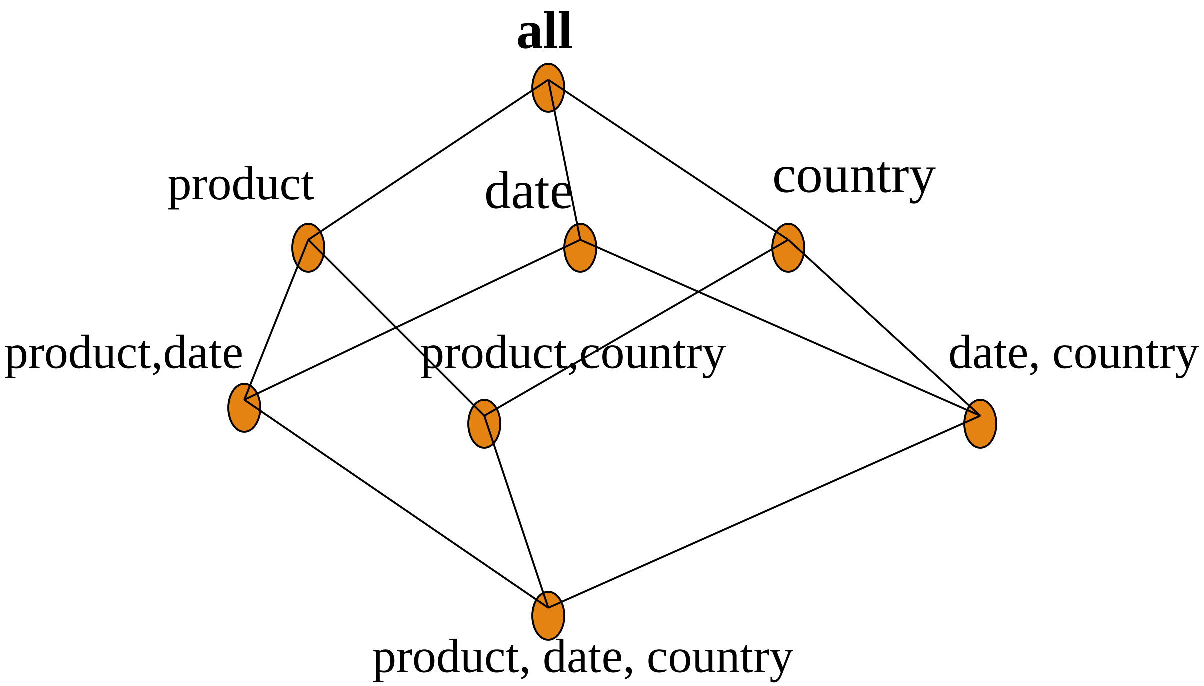
A Sample Data Cube




Product
P1
P2
P3
P4
P5
U.S.A
Canada
Mexico
Cuba
Country
1 Qtr
2 Qtr
3 Qtr
4 Qtr
Date

28

Cuboids Corresponding to the Cube



0-D (*apex*) cuboid

1-D cuboids

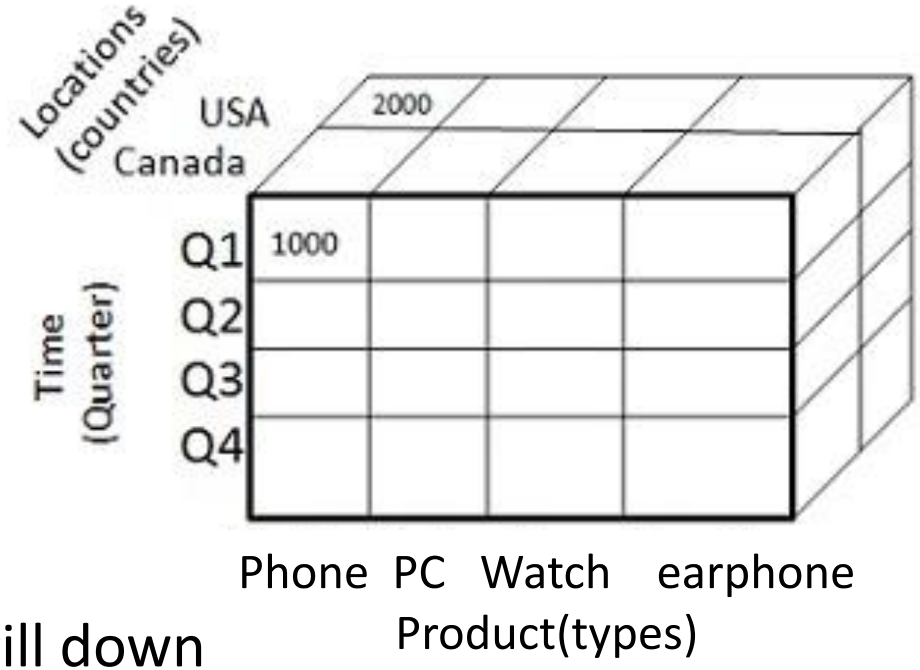
2-D cuboids

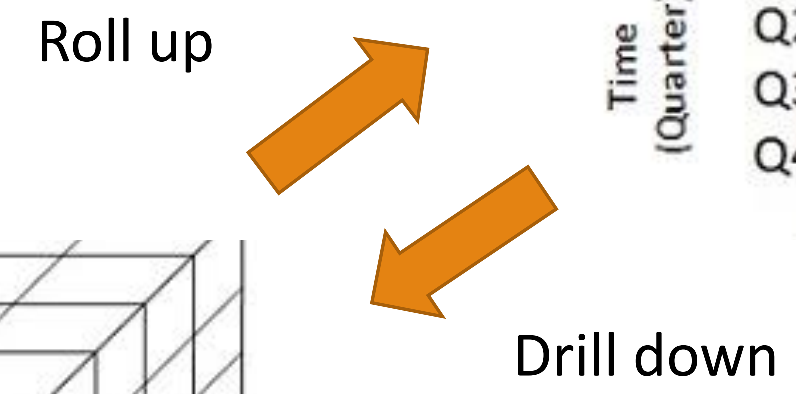
3-D (*base*) cuboid

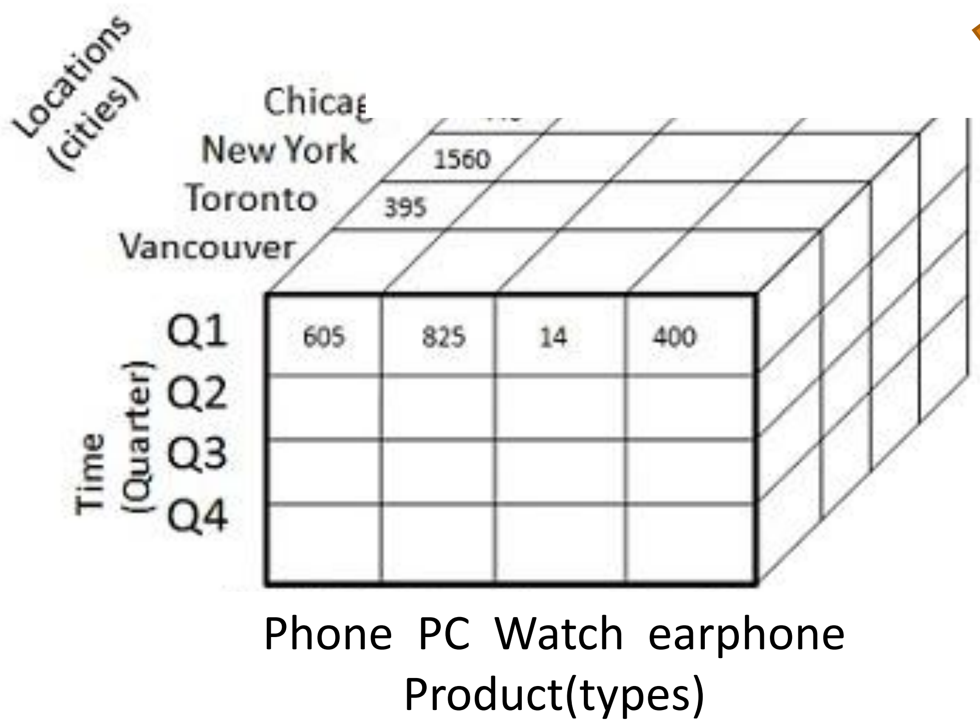
How can we play with the Cube?

Roll up & Drill down

Roll up (drill-up): summarize data
by climbing up hierarchy or by dimension reduction

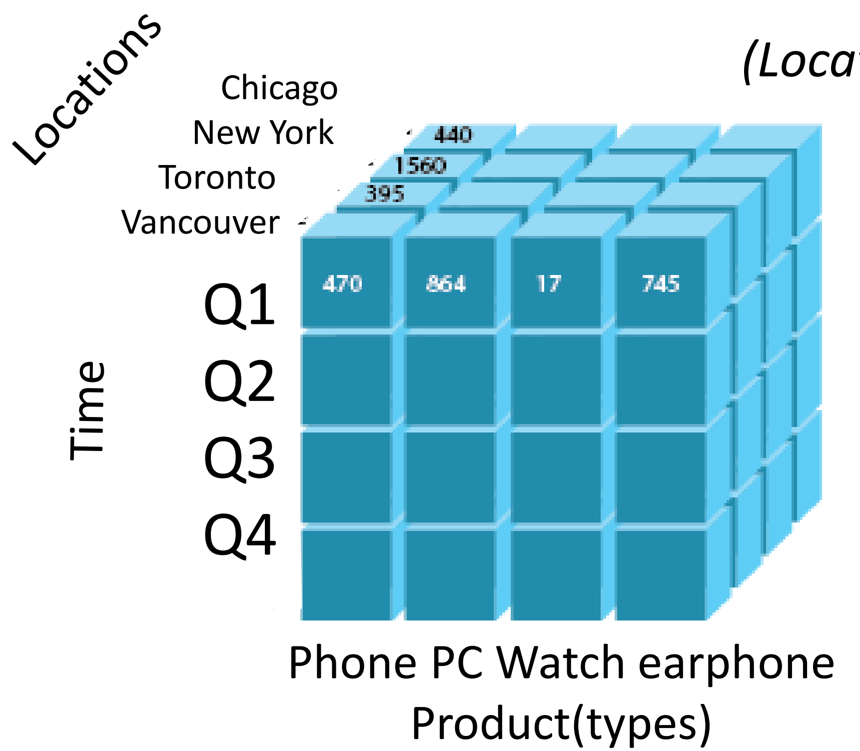




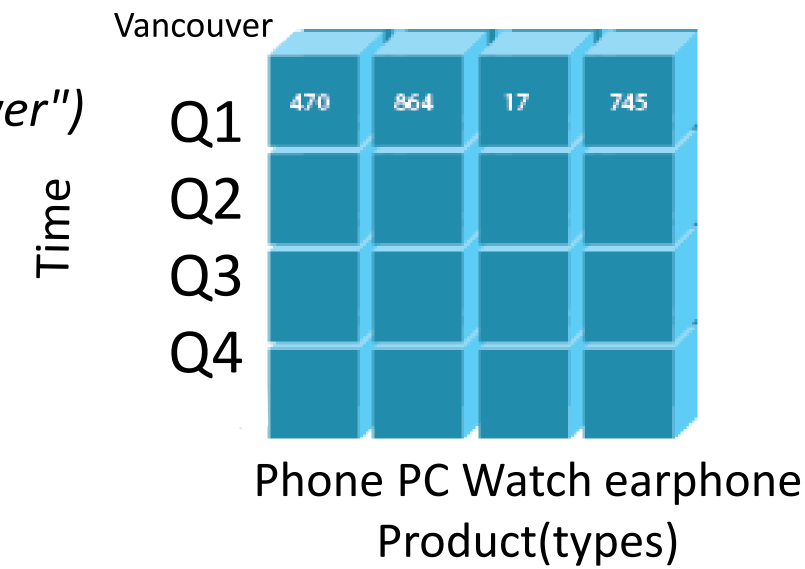


Drill down (roll down): reverse of roll-up
from higher level summary to lower level summary or detailed data, or introducing new dimensions

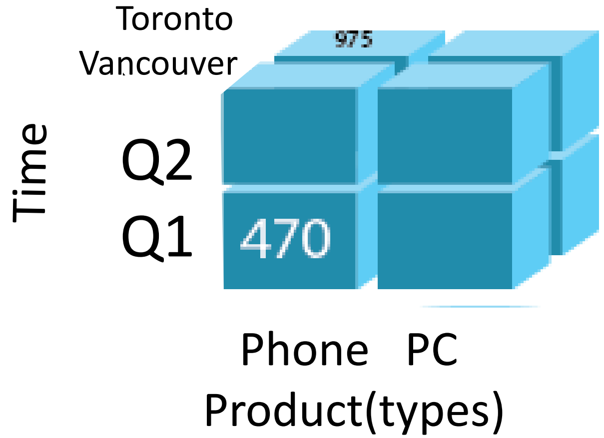
Dice and Slice



Slice for
(Location = "Vancouver")



Dice for
(Location = "Toronto" or "Vancouver")
and (Time = "Q2" or "Q1") and
(Product = "Phone" or "PC")



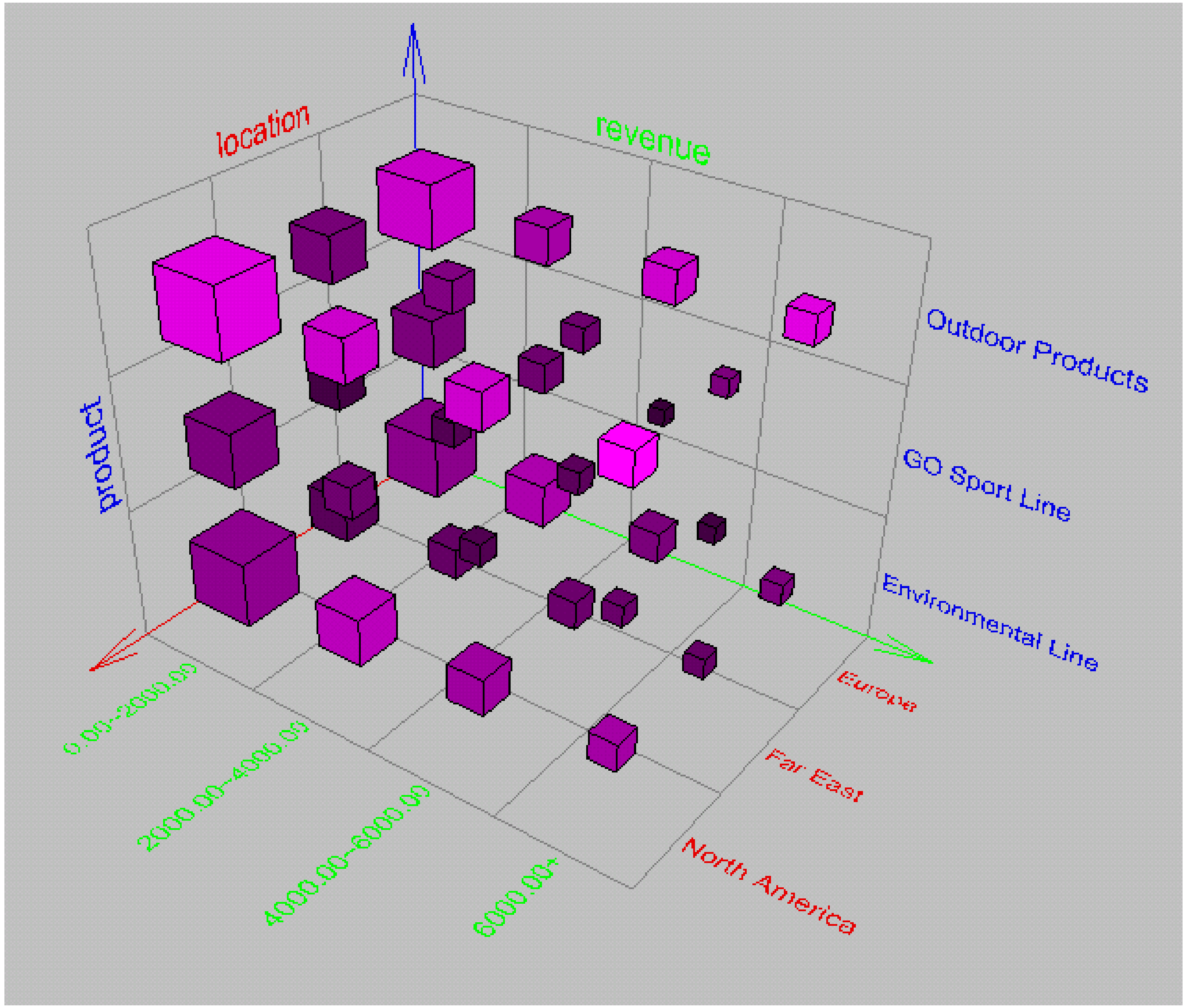
Other Typical OLAP Operations

- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes*
- Drill across:
  - *involving (across) more than one fact table*
- Drill through:
  - *through the bottom level of the cube to its back-end relational tables (using SQL)*

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)
  - Is min_N() an algebraic measure? How about standard_deviation()
- Holistic: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., median(), mode(), rank()

Browsing a Data Cube



- Visualization
- OLAP capabilities
- Interactive manipulation

Chapter 4: Data Warehousing and On-line Analytical Processing

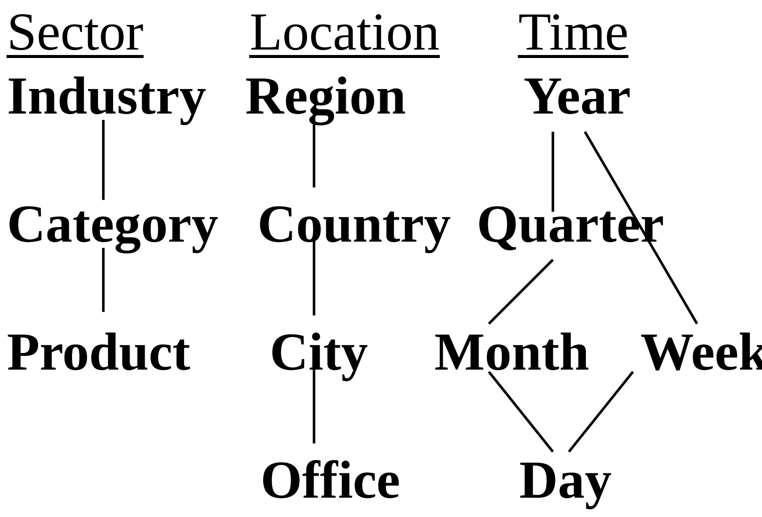
- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Implementation
- Summary

Efficient Data Cube Computation

- If I have ***n*** dimensions, each with L_i levels, how many cuboids are needed to preprocess all?
- Calculating **all** cuboids is costly in computation and time.
- How to decide which cuboid be pre-calculated (Materialization)?
  - Based on size of data, sharing, access frequency, etc.
  - Example: I know my users always search by Quarter, so that cuboid should be pre-calculated.
  - Example: If I pre-calculate days, I can use days as input to Months (30 or 31 days), or weeks (7 days), etc.

Why this formula?

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



The "Compute Cube" Operator

- Cube definition and computation in DMQL(Data Mining Query Language)

  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.'96)

  SELECT item, city, year, SUM (amount)

  FROM SALES

  CUBE BY item, city, year

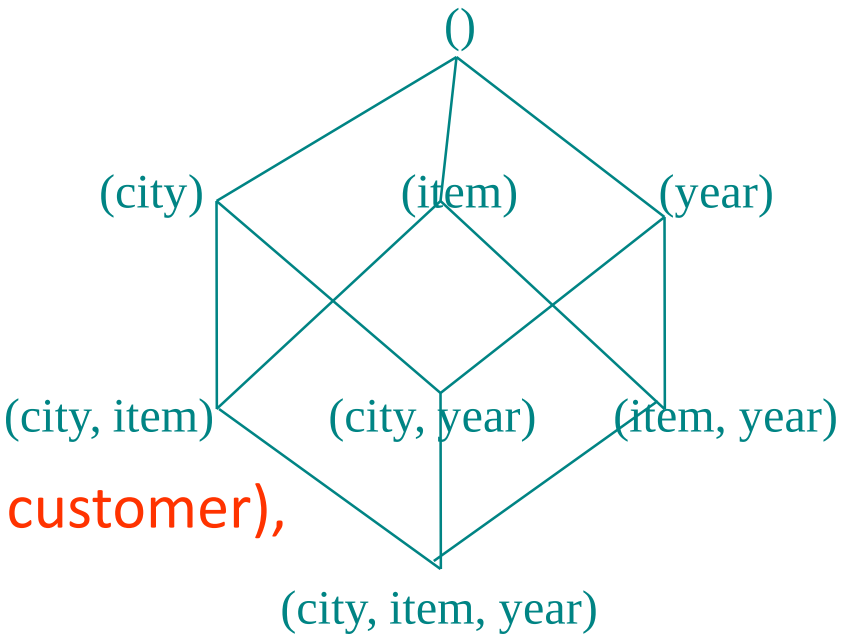
- Need compute the following Group-Bys ***(2 ^ 3)***

  3D Cuboid → (date, product, customer),

  2D Cuboid → (date, product),(date, customer), (product, customer),

  1D Cuboid → (date), (product), (customer)

  0D (Apex) Cuboid → ()



Indexing OLAP Data

- **Indexing**
  - Main purpose of indexing is to make the calculation faster/efficient
- **Common Warehouse Index: Bitmap Index**
  - **Benefits in Warehousing:**
    - Reduced response time for large classes of ad hoc queries.
    - Reduced storage requirements compared to other indexing techniques.
    - Dramatic performance gains even on hardware with a relatively small number of CPUs or a small amount of memory.

https://docs.oracle.com/database/121/DWHSG/schemas.htm#DWHSG9041

Indexing OLAP Data: Bitmap Index

- Index on a particular column
  - Each value in the column has a bit vector: bit-op is fast
  - The length of the bit vector: # of records in the base table
  - The *i*-th bit is set if the *i*-th row of the base table has the value for the indexed column
  - Not suitable for high cardinality domains. (WHY?)
- A recent bit compression technique, Word-Aligned Hybrid (WAH), makes it work for high cardinality domain as well [Wu, et al. TODS'06]

Base table

Cust	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retail
C5	Europe	Dealer

Index on Region

RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on Type

RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0
5	0	1

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)**
  - Data is stored in a relational database.
  - Greater scalability
- **Multidimensional OLAP (MOLAP)**
  - Everything is in multi-dimensional storage (see page 26 for an example)
  - Fast indexing to pre-computed summarized data
- **Hybrid OLAP (HOLAP)**
  - Used by : Microsoft SQLServer
  - Combines both ROLAP & MOLAP
  - Theoretically provides best performance

Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Implementation
- Summary

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
- Implementation: Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Indexing OALP data: Bitmap index and join index
  - OLAP query processing
  - OLAP servers: ROLAP, MOLAP, HOLAP

References (I)

- 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 (II)

- 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.