DATA WAREHOUSING

Hari Sundaram

hs1@illinois.edu

http://sundaram.cs.illinois.edu

adapted from slides by Jiawei Han and Kevin Chang

BASIC CONCEPTS

Data Cube and OLAP   Usage   Implementation   Summary

Supports information processing by providing a solid platform of consolidated, historical data for analysis.

what is a data warehouse?

A decision support database that is maintained separately from the organization's operational database

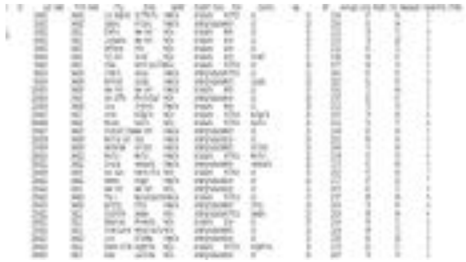
> A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.
>
> *-JH Inmon*

SUBJECT ORIENTED

Subjects: Organized around major subjects, such as customer, product

Decision Making: Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing

Simple and Concise: Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process



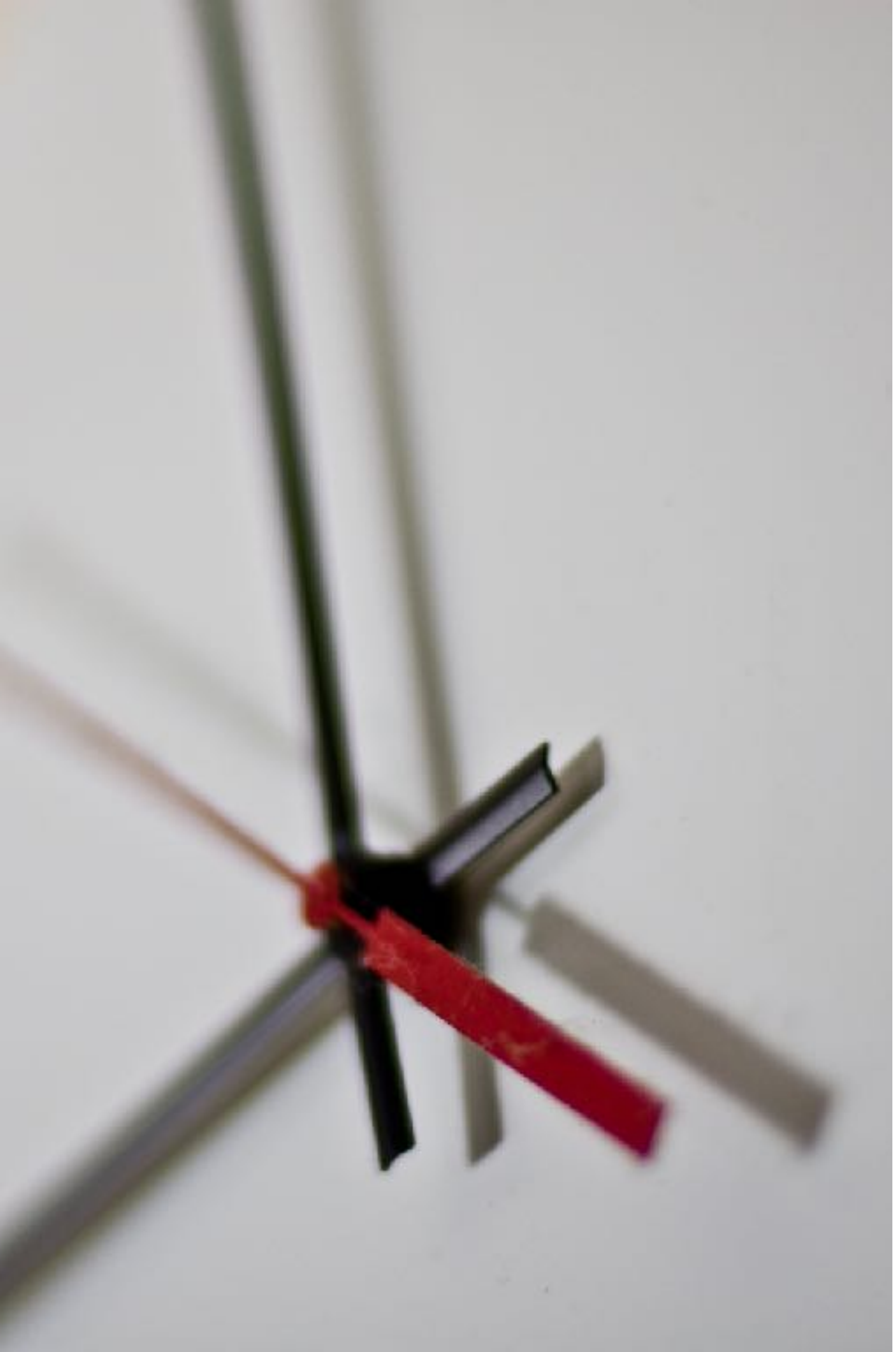
INTEGRATED

Data Integration: Constructed by integrating multiple, heterogeneous data sources

- relational databases, flat files, on-line transaction records

Data Preprocessing: Data cleaning and data integration techniques are applied.

- Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
  - E.g., Hotel price: currency, tax, breakfast covered, etc.
- When data is moved to the warehouse, it is converted.



TIME HORIZON

- Historical Perspective: 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)
- Time Related: 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"

NON VOLATILE

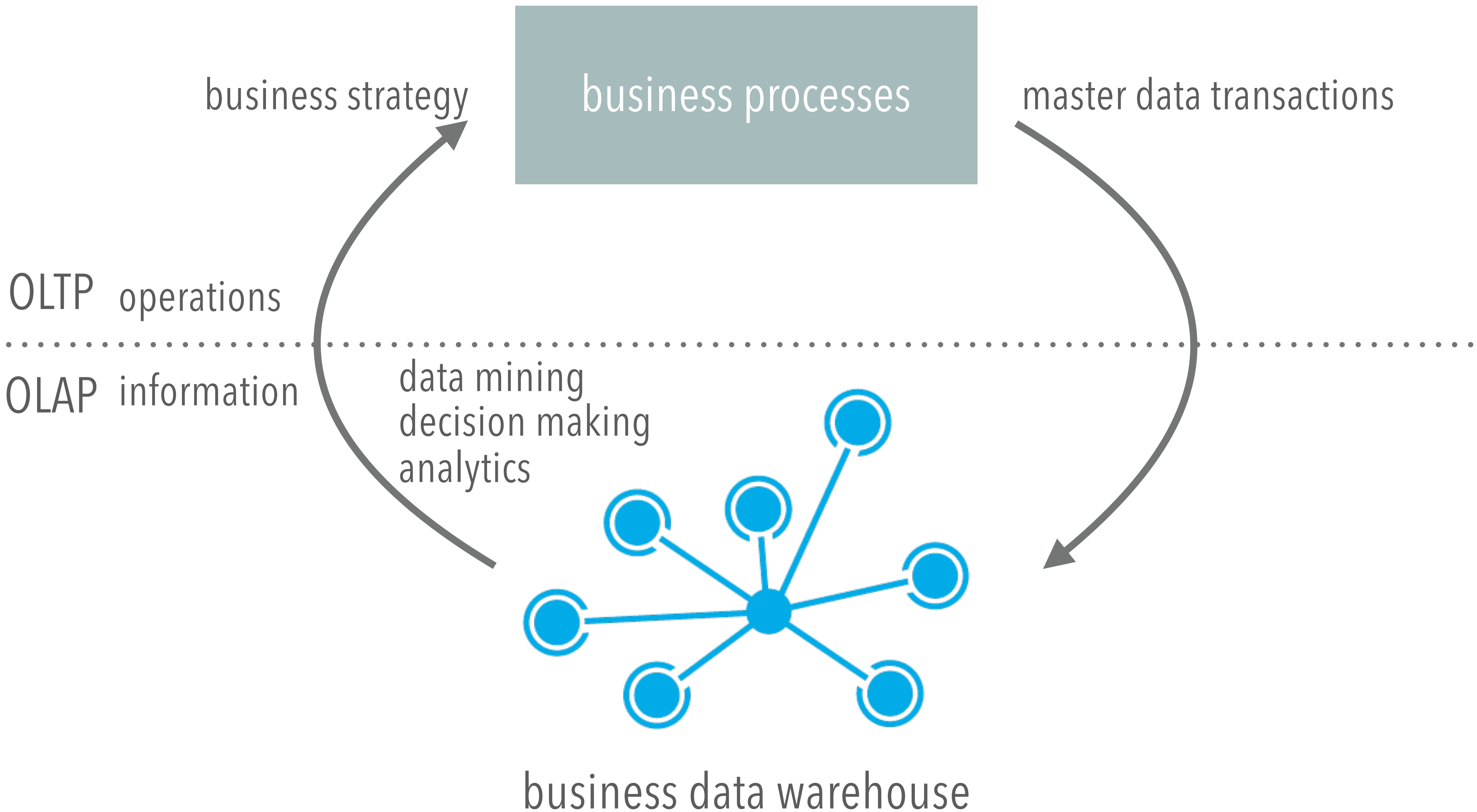
OLTP

Independence: A physically separate store of data transformed from the operational environment. Keep in high performance for both systems (OLTP vs. OLAP)

Static Status: 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

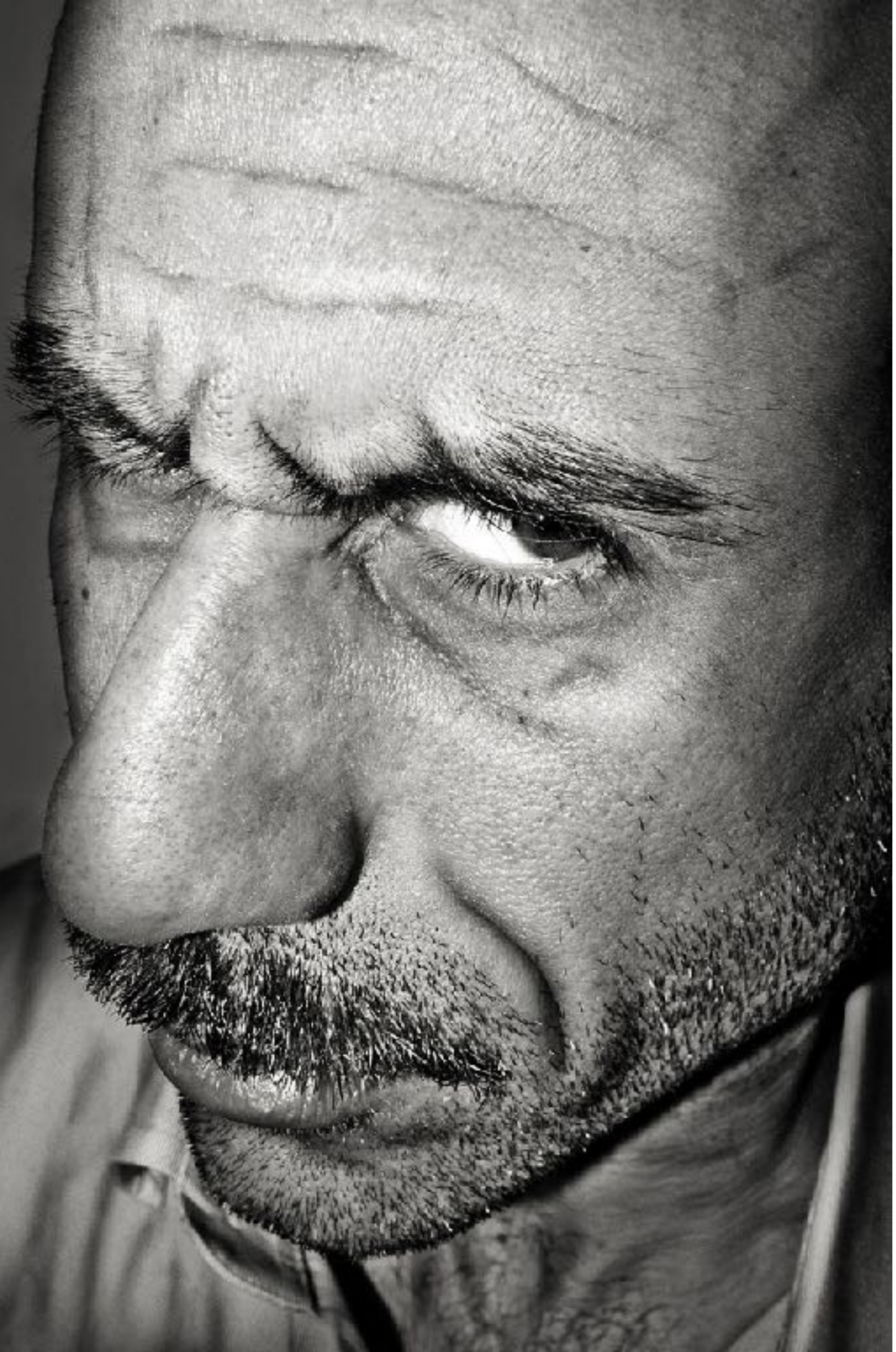
OLAP vs. OLTP



business strategy
business processes
master data transactions
OLTP operations
OLAP information
data mining
decision making
analytics
business data warehouse

OLAP vs. OLTP

Feature	*OLTP*	*OLAP*
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements decision support
DB design	ER-based, application-oriented	star/snowflake, subject-oriented
Data	current, guaranteed up-to-date	historic, accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	GB to high-order GB	≥ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time



WHY THE SEPARATION?

High performance for both systems

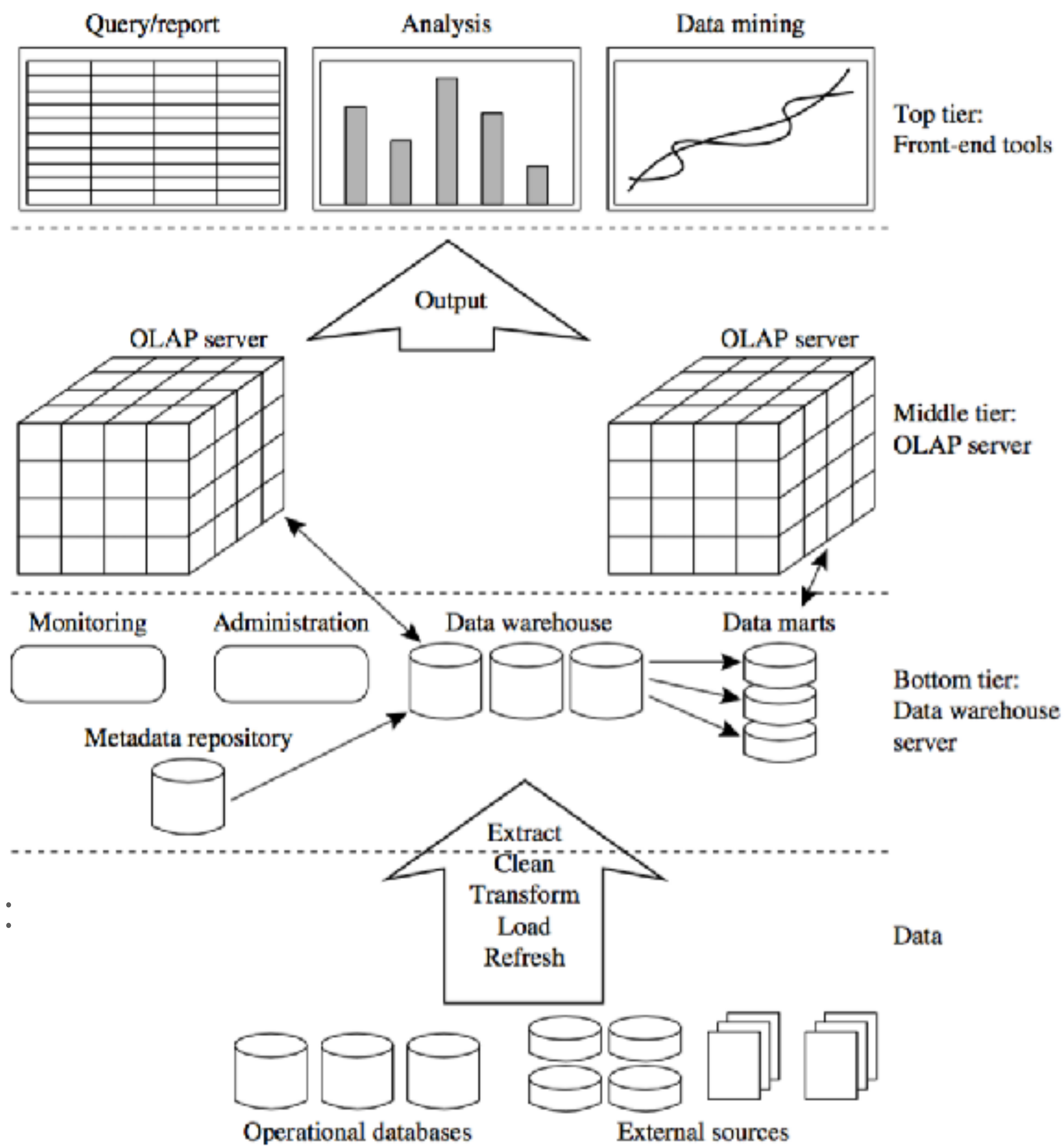
- DBMS— tuned for OLTP: access methods, indexing, concurrency control, recovery
- Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation

Different functions and different data:

- missing data: Decision support requires historical data which operational DBs do not typically maintain
- data consolidation: Decision support 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: Increasingly, systems perform OLAP analysis directly on relational databases

Data Warehouse: A Multi-tiered Architecture



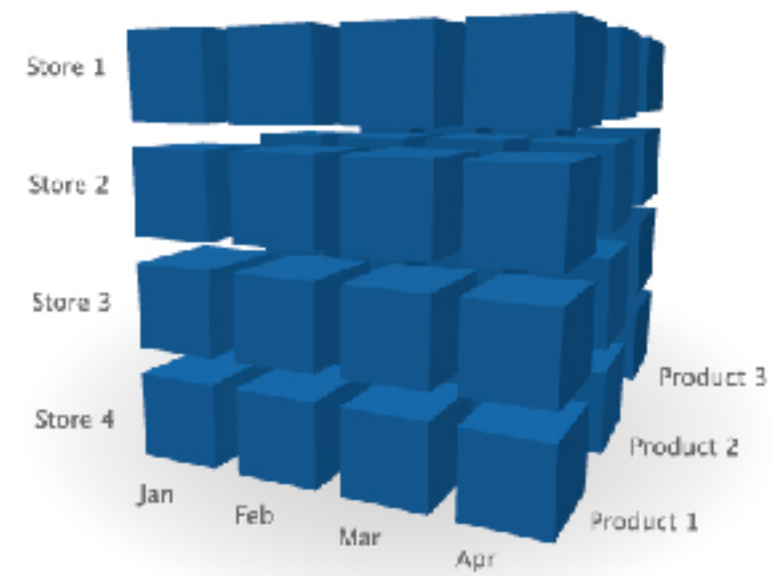
A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.

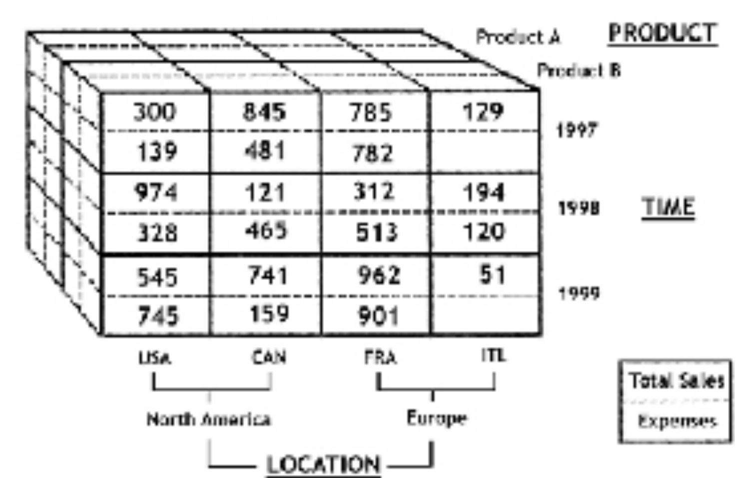


DATA WAREHOUSE SUMMARY

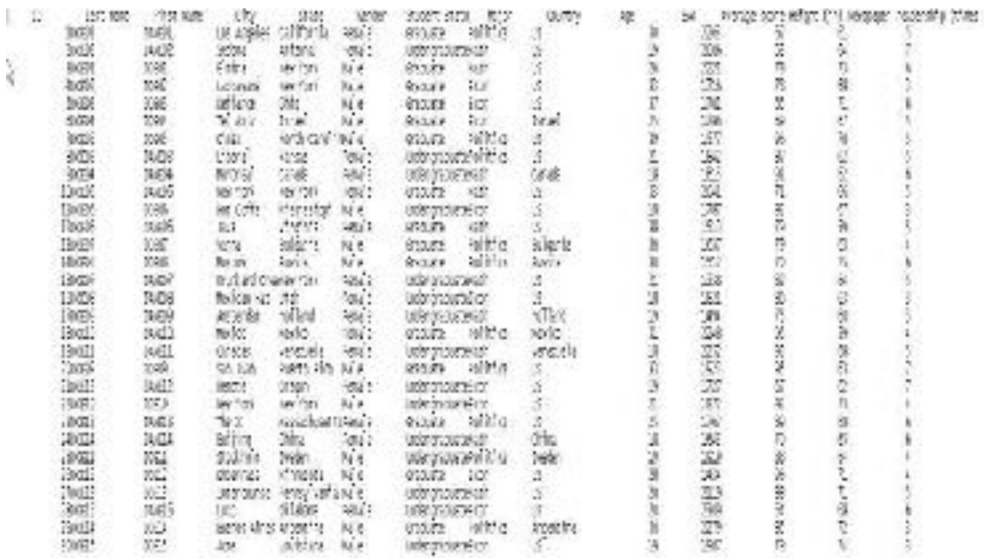

OLAP vs. OLTP

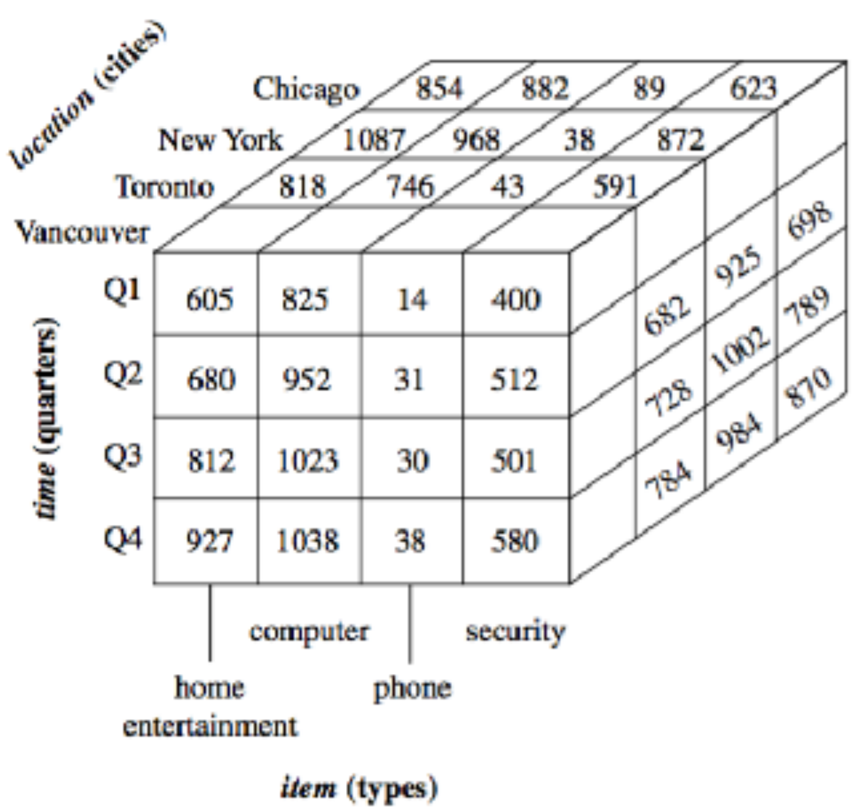
DATA CUBE AND OLAP





Basic Concepts Usage Implementation Summary





FROM 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 is a multidimensional generalization of data spreadsheet.

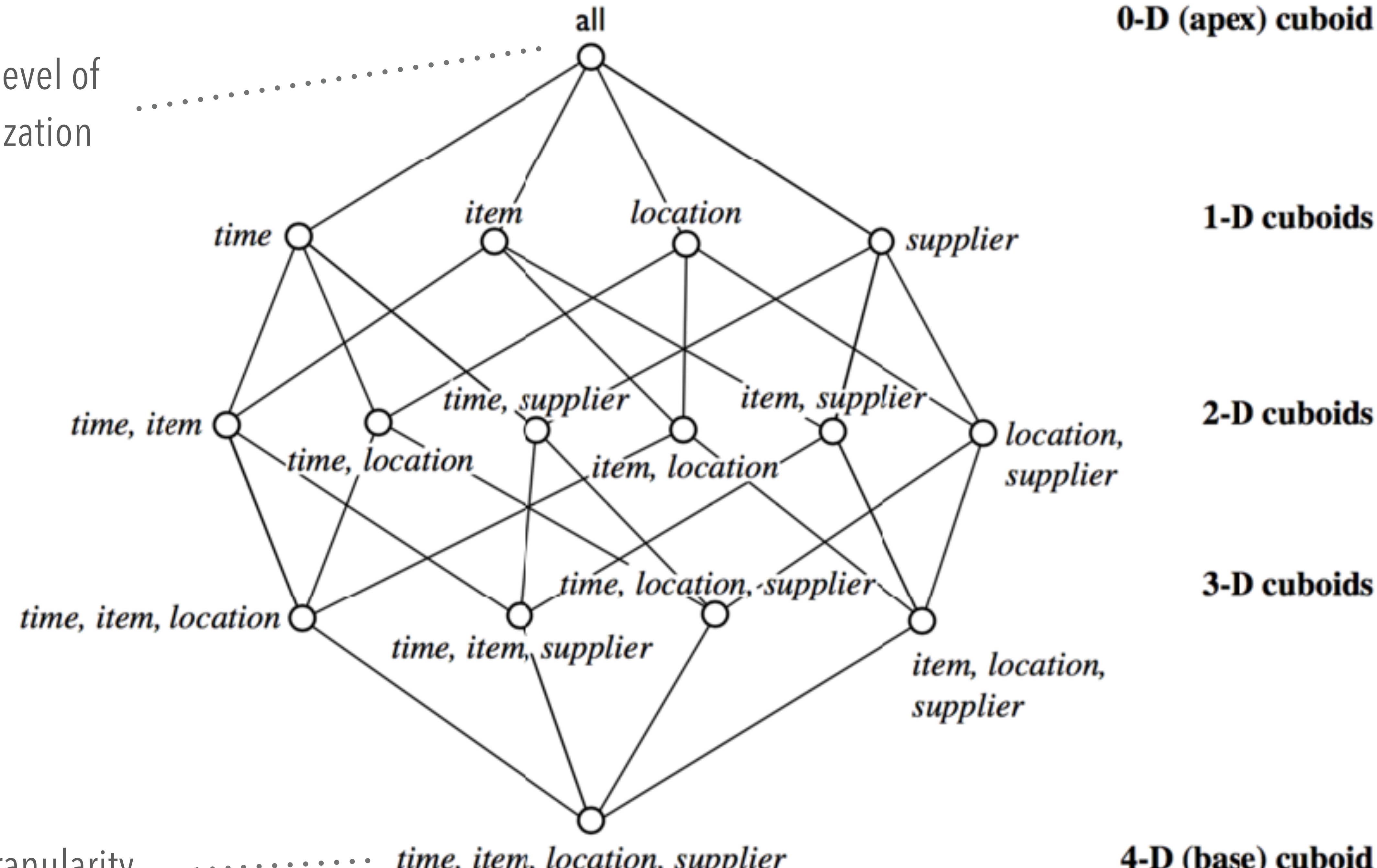
A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions

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

data cube

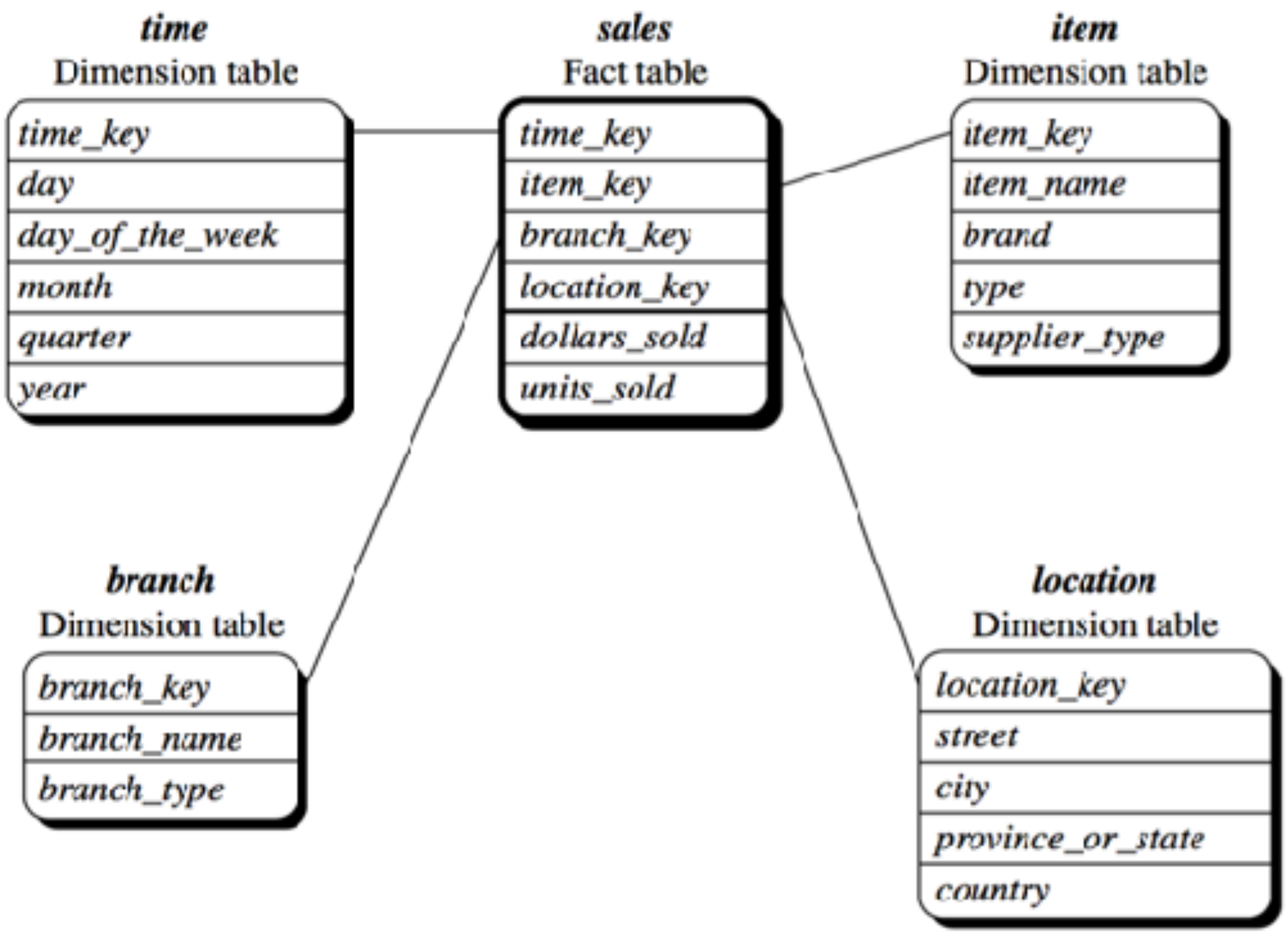
A data cuboid is a subset of data cube.

highest-level of summarization



finest granularity

CONCEPTUAL MODELS

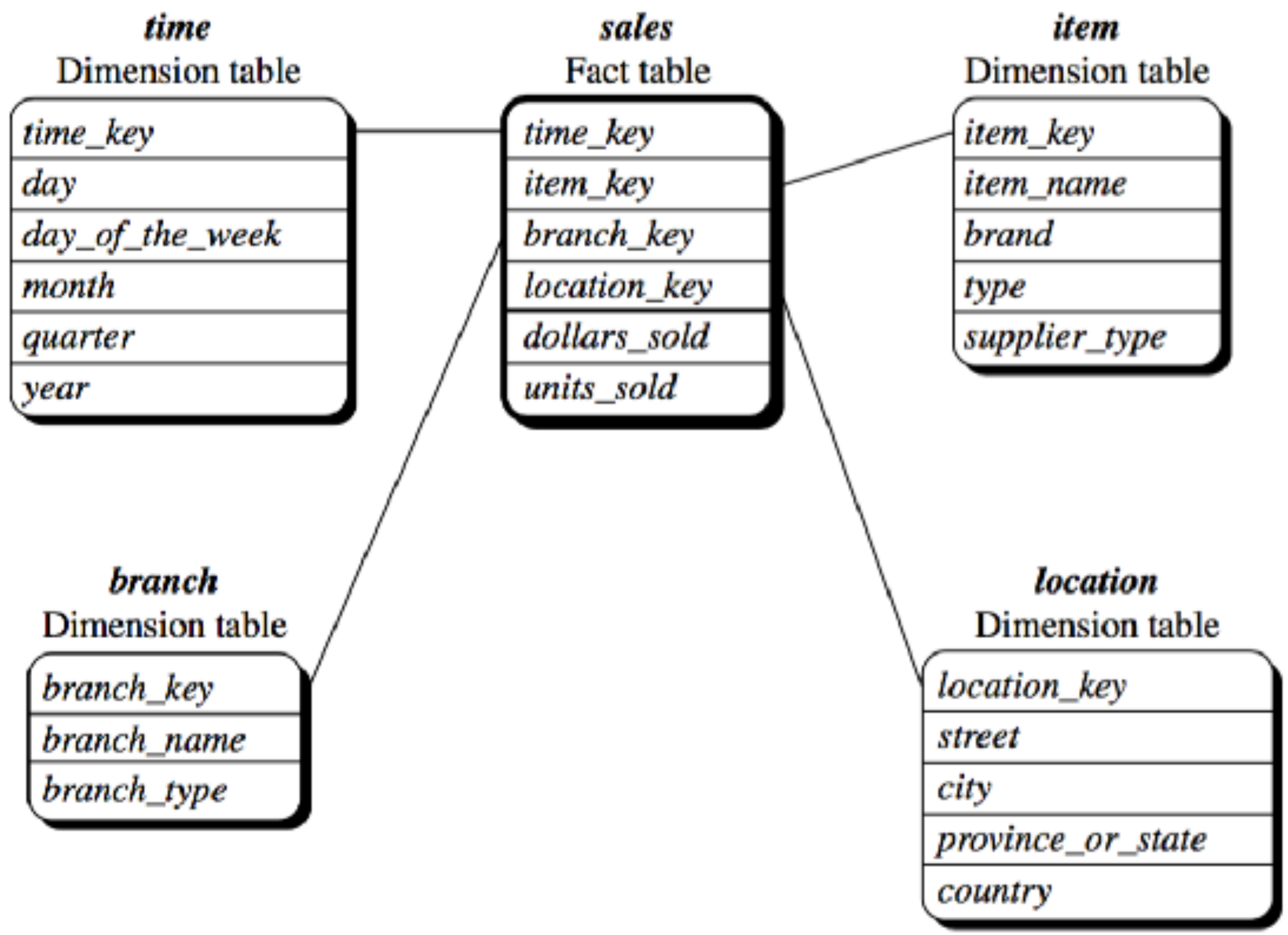


star schema

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

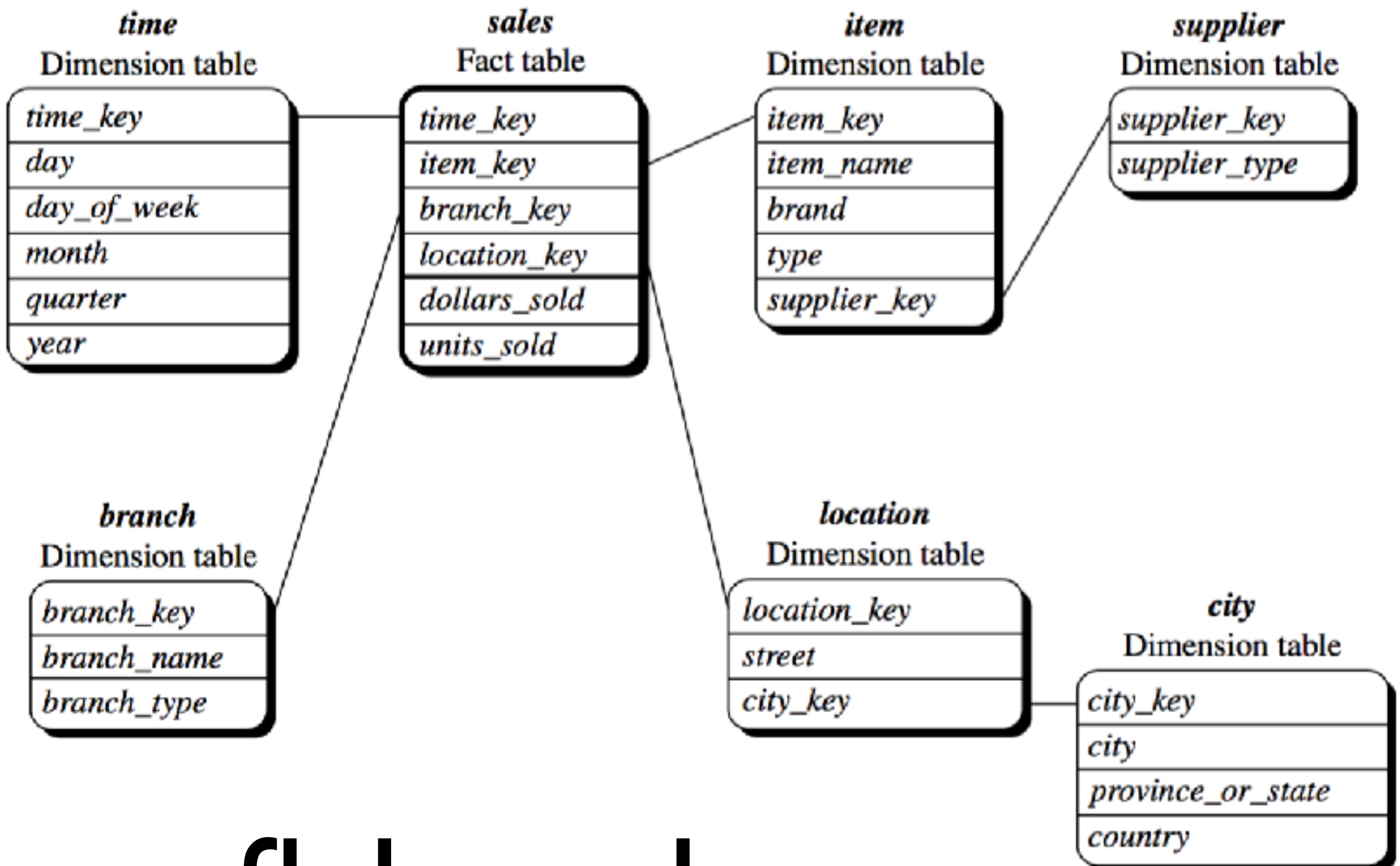
Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake

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

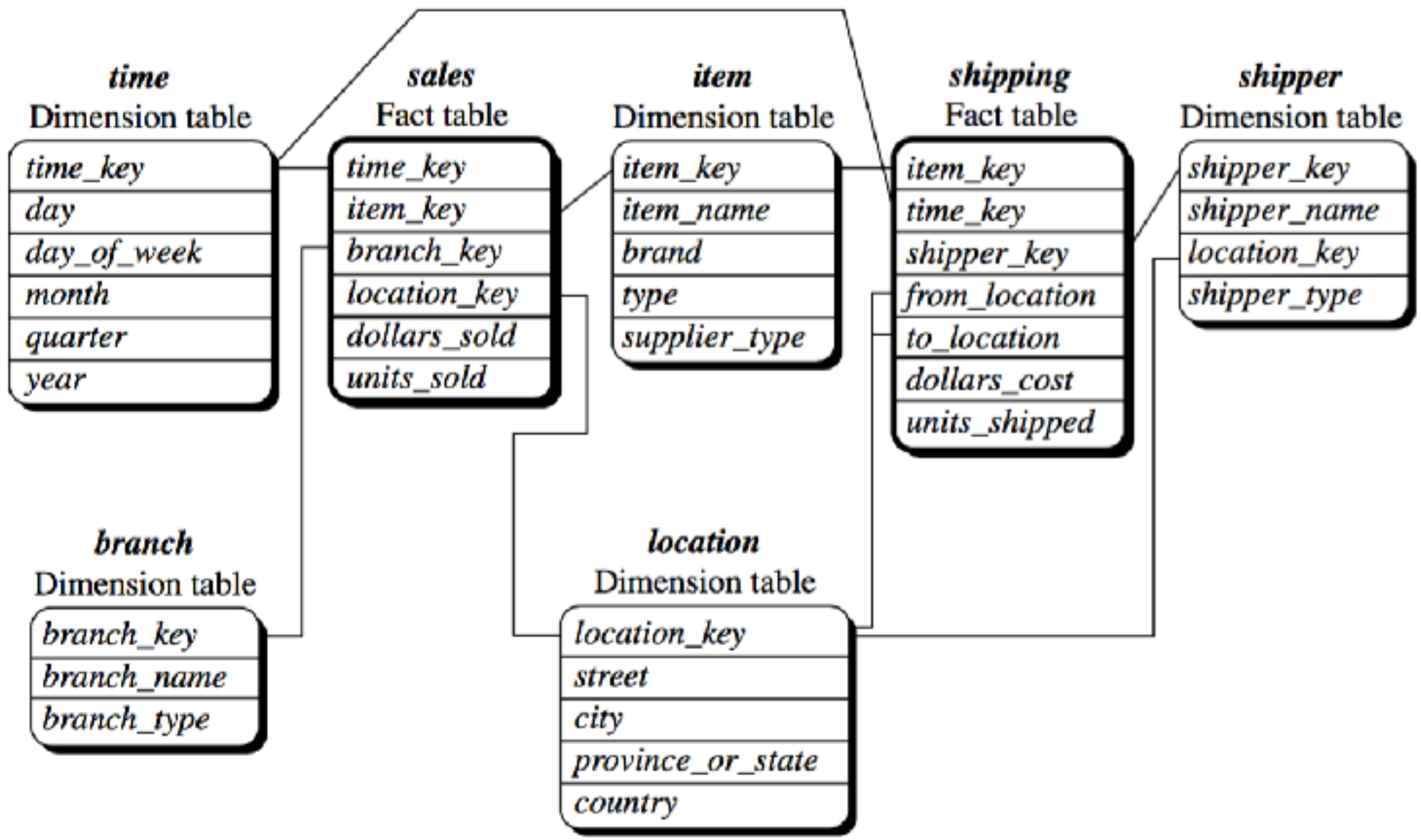


time
Dimension table
time_key
day
day_of_the_week
month
quarter
year
sales
Fact table
time_key
item_key
branch_key
location_key
dollars_sold
units_sold
item
Dimension table
item_key
item_name
brand
type
supplier_type
branch
Dimension table
branch_key
branch_name
branch_type
location
Dimension table
location_key
street
city
province_or_state
country

star schema

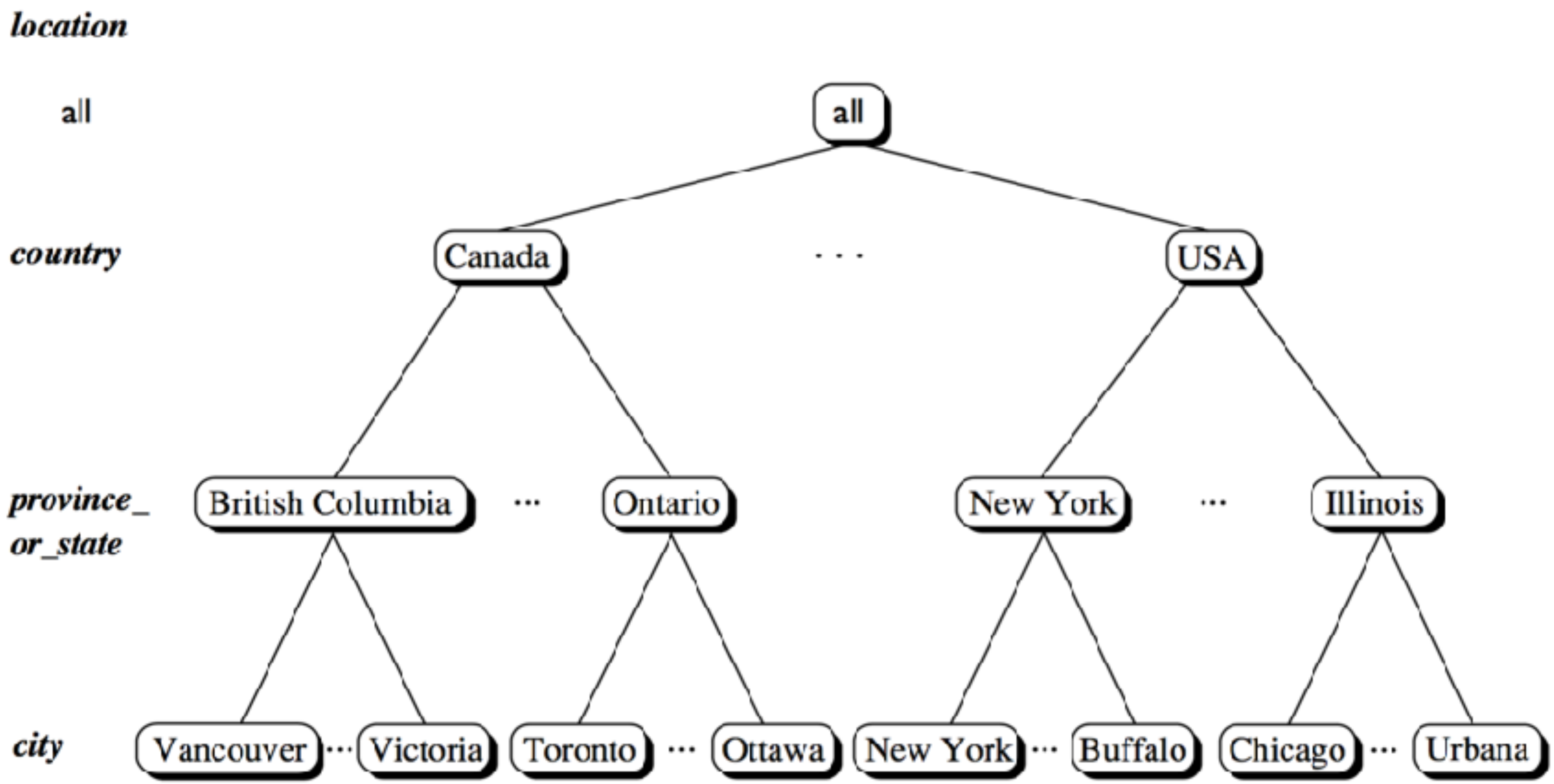


snowflake schema



fact constellation

two fact tables share dimensions



concept hierarchy

Exercise!

sum

median

average

5,6,8,10,3,7,11

count(a+b+c) = count(a+b) + count(c)

sum: (3 + 4 + 5) = ((3+4) + 5) = (3+(4+5)

distributive measure

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

Exercise!

sum

median

average

5,6,8,10,3,7,11

which are 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 distributively calculated

$$f\,\underbrace{(x, y, z)}_{\text{bounded number of arguments}}$$

distributively calculated

average(x) = sum(x) / count(x)

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 median, mode

holistic

not distributive, not algebraic

If there is no constant bound on the storage size needed to describe a sub-aggregate.

min

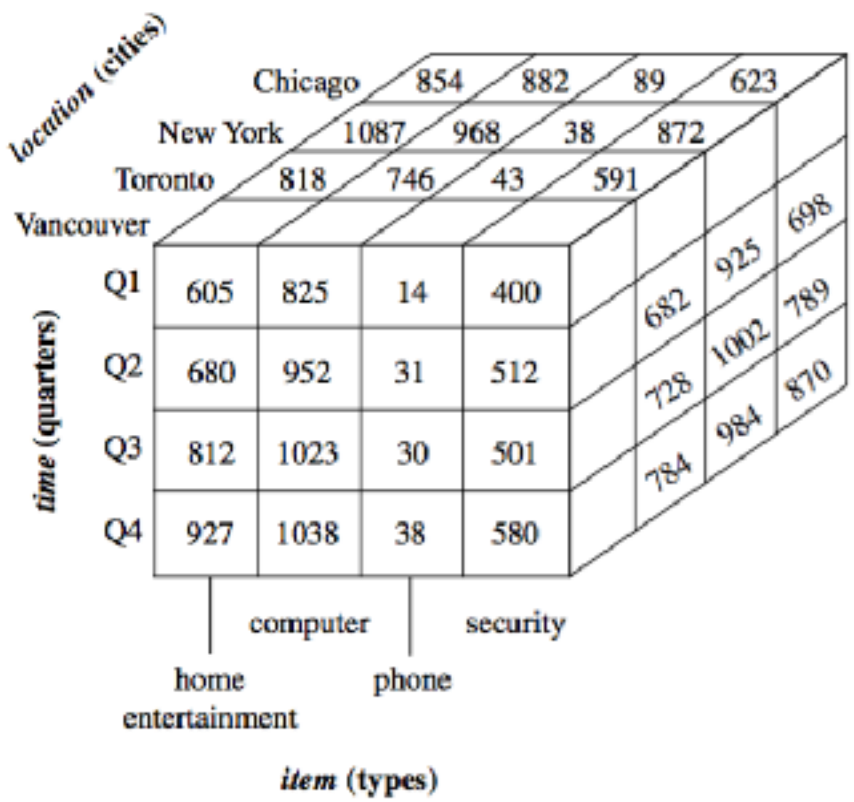
max

standard deviation

rank

5,6,8,10,3,7,11

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

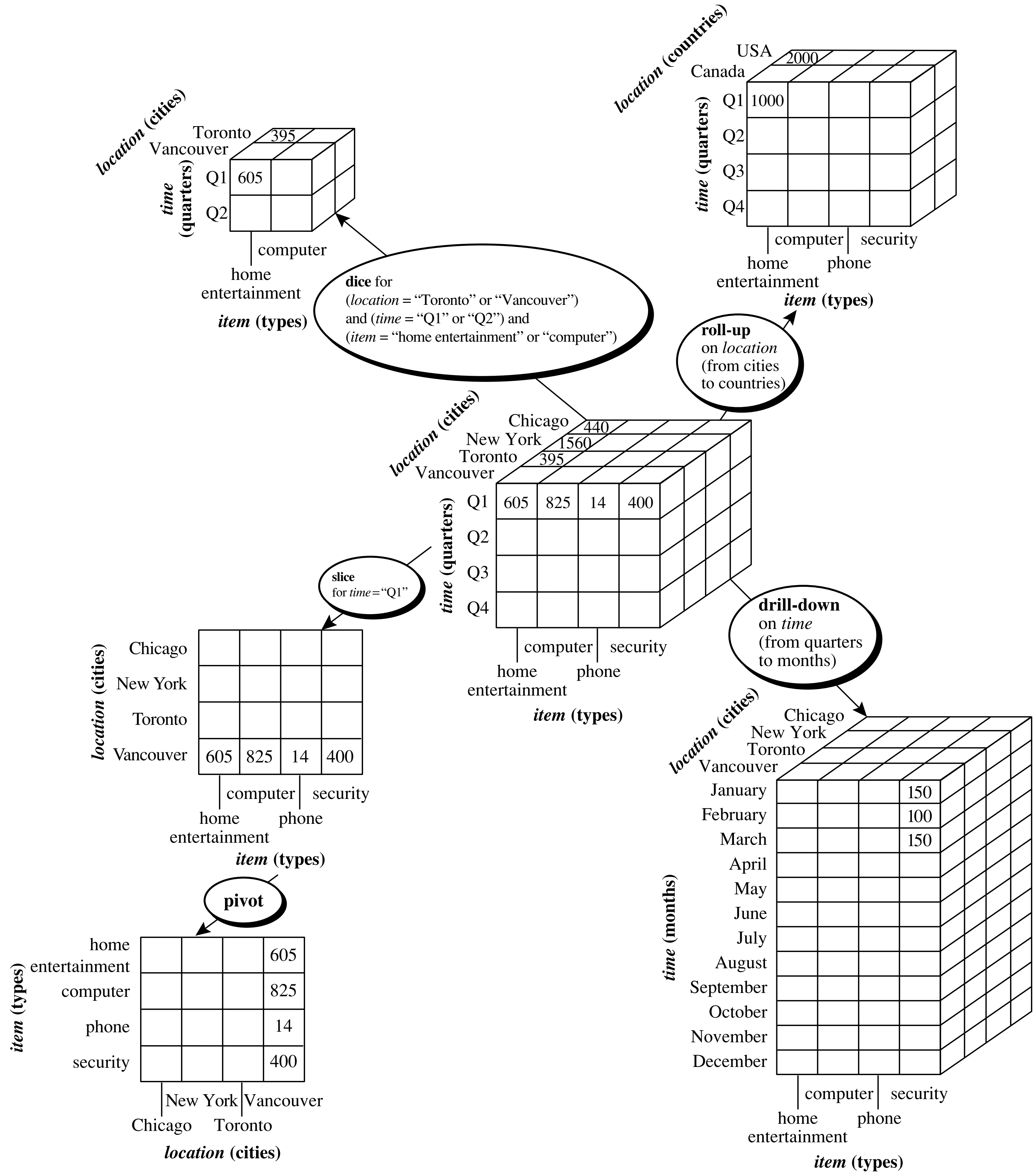
Slice and dice: selection on one and multiple dimensions, respectively.

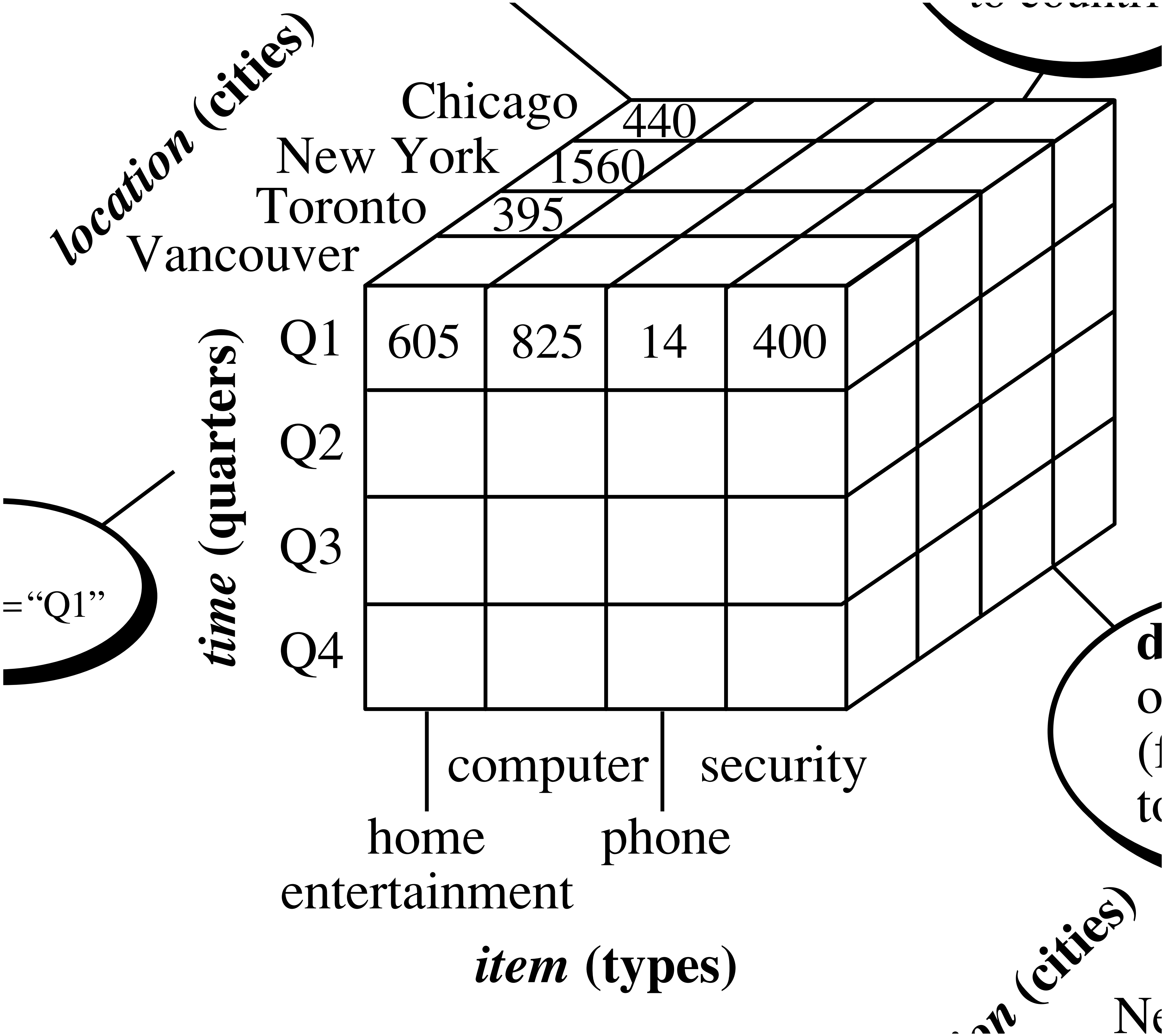
Pivot (rotate):

- reorient the cube, visualization, 3D to series of 2D planes

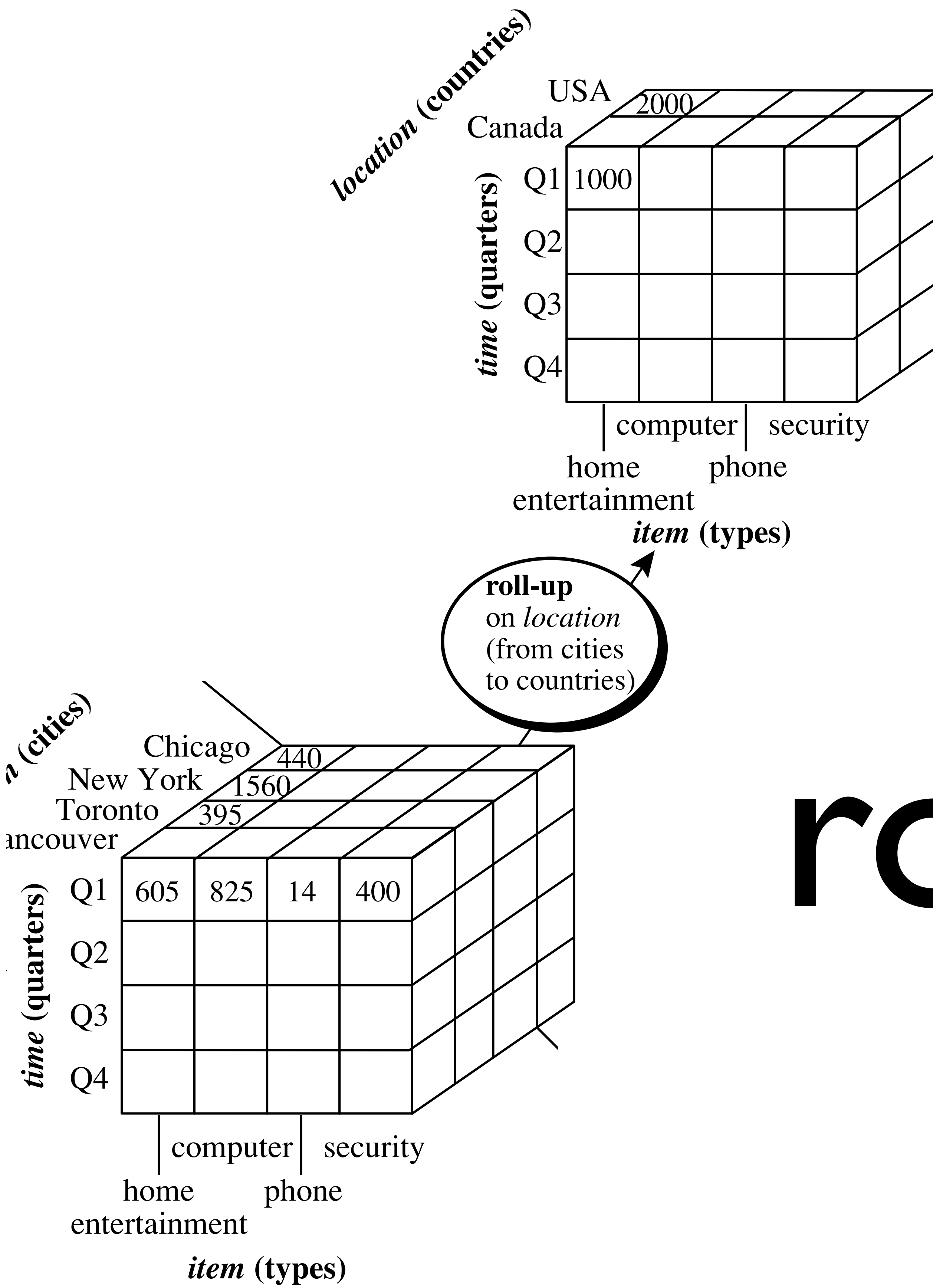
Other operations

- 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)

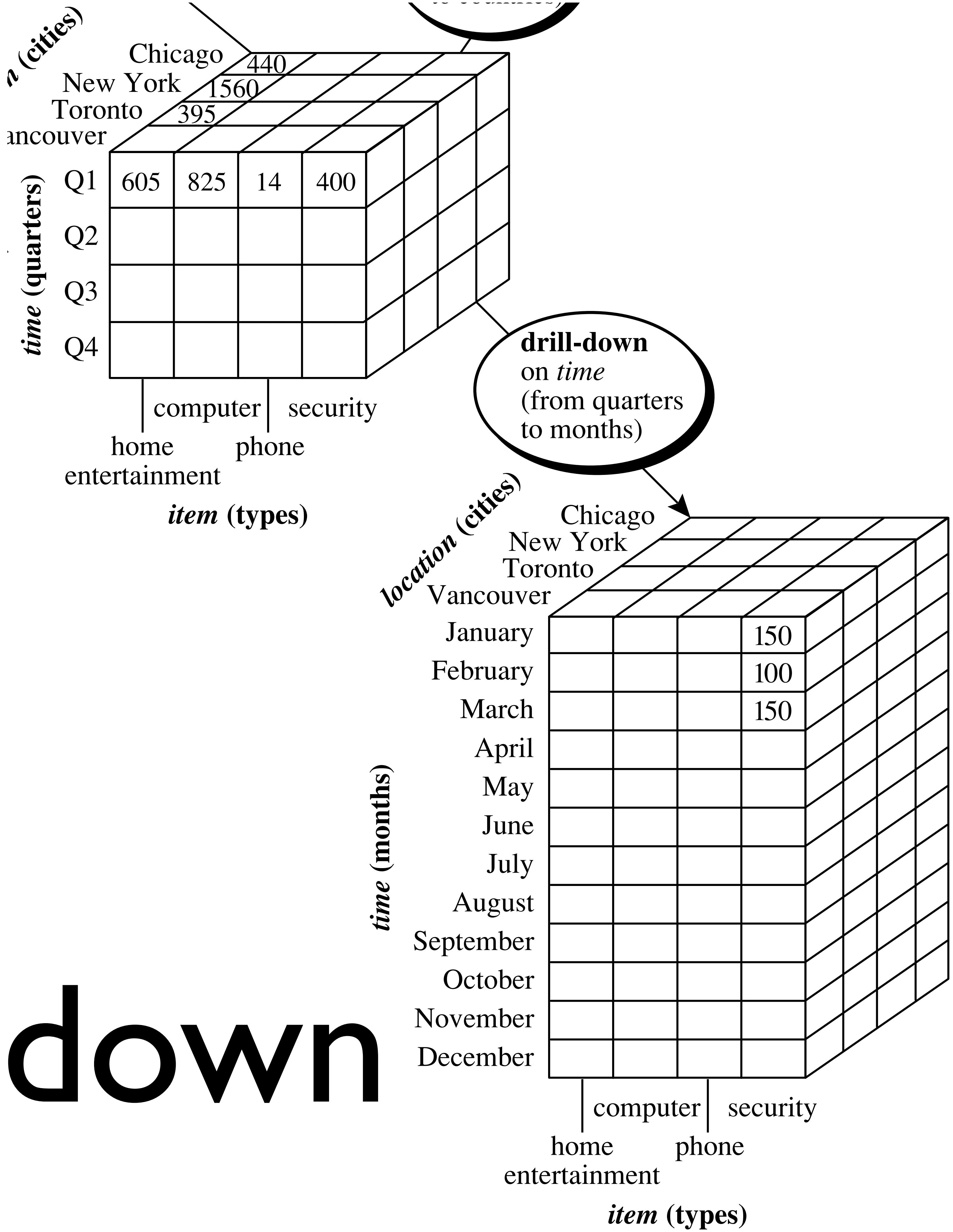

location (cities)
Toronto
Vancouver
395
Q1
605
Q2
time (quarters)
computer
home entertainment
item (types)
dice for (location = “Toronto” or “Vancouver”) and (time = “Q1” or “Q2”) and (item = “home entertainment” or “computer”)
location (countries)
USA
Canada
2000
Q1
1000
Q2
Q3
Q4
time (quarters)
computer
security
home entertainment
phone
item (types)
roll-up on location (from cities to countries)
location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Q1
Q2
Q3
Q4
605
825
14
400
time (quarters)
computer
security
home entertainment
phone
item (types)
slice for time = “Q1”
Chicago
New York
Toronto
Vancouver
location (cities)
605
825
14
400
computer
security
home entertainment
phone
item (types)
pivot
home entertainment
computer
phone
security
item (types)
605
825
14
400
New York
Vancouver
Chicago
Toronto
location (cities)
drill-down on time (from quarters to months)
location (cities)
Chicago
New York
Toronto
Vancouver
January
February
March
April
May
June
July
August
September
October
November
December
time (months)
150
100
150
computer
security
home entertainment
phone
item (types)



location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
="Q1"
home entertainment
computer
phone
security
item (types)
(cities)

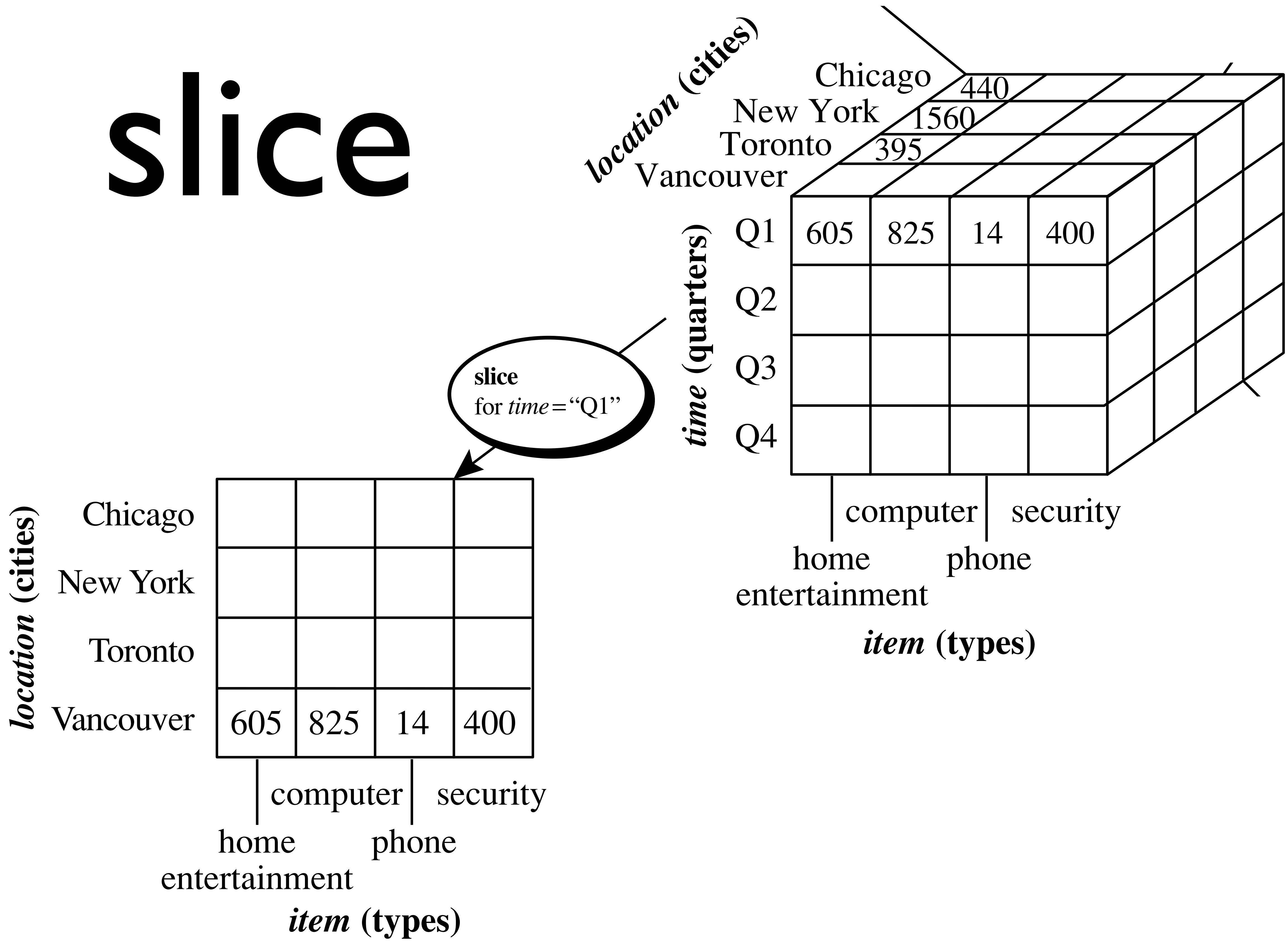

location (countries)
USA
Canada
2000
time (quarters)
Q1
Q2
Q3
Q4
1000
computer
security
home
entertainment
phone
item (types)
roll-up
on location
(from cities
to countries)
(cities)
Chicago
New York
Toronto
ancouver
440
1560
395
Q1
Q2
Q3
Q4
605
825
14
400
time (quarters)
computer
security
home
entertainment
phone
item (types)

roll up


(cities)
Chicago
New York
Toronto
ancouver
440
1560
395
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
computer
security
home
entertainment
phone
item (types)
drill-down
on time
(from quarters
to months)
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)

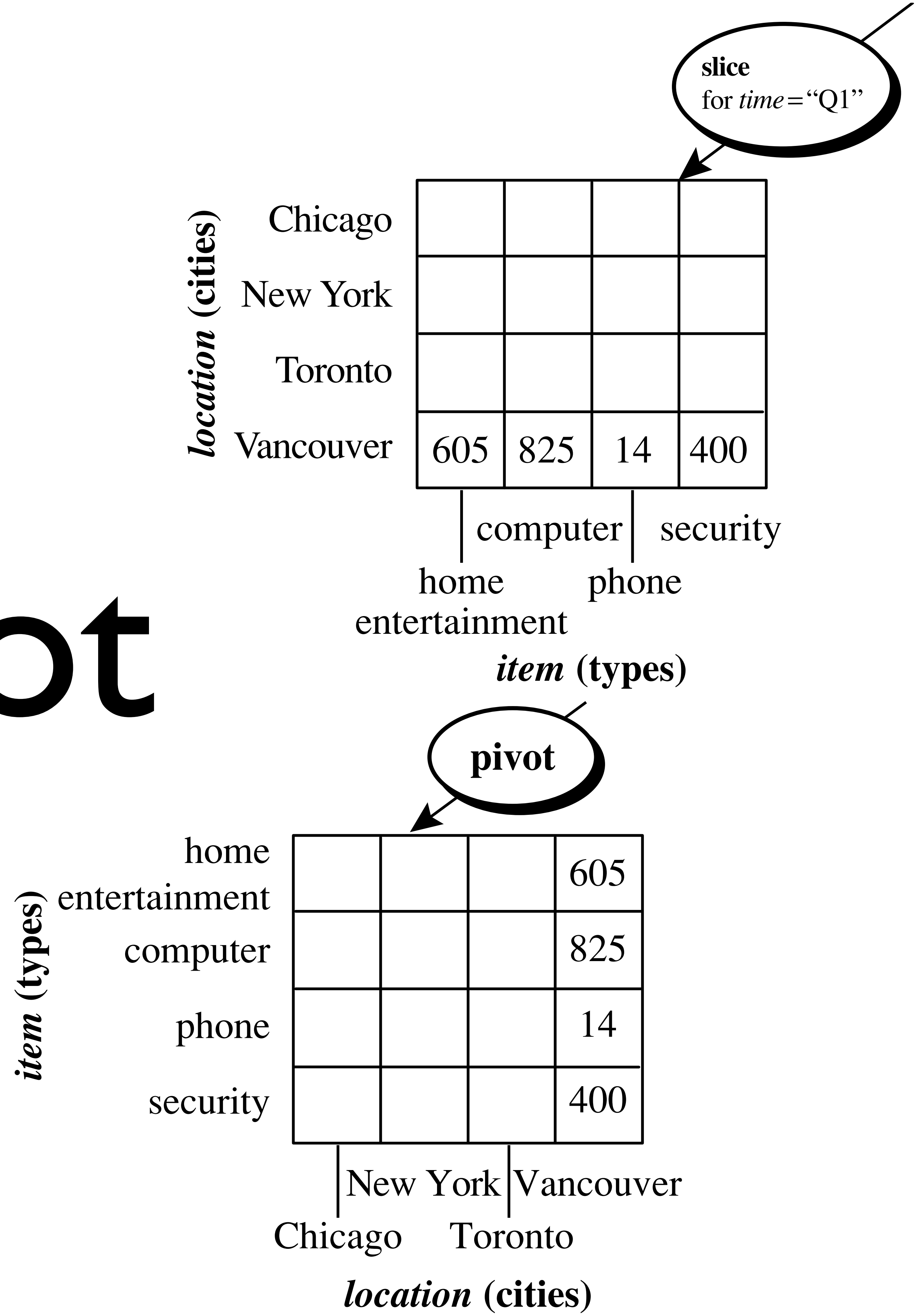
drill down

slice

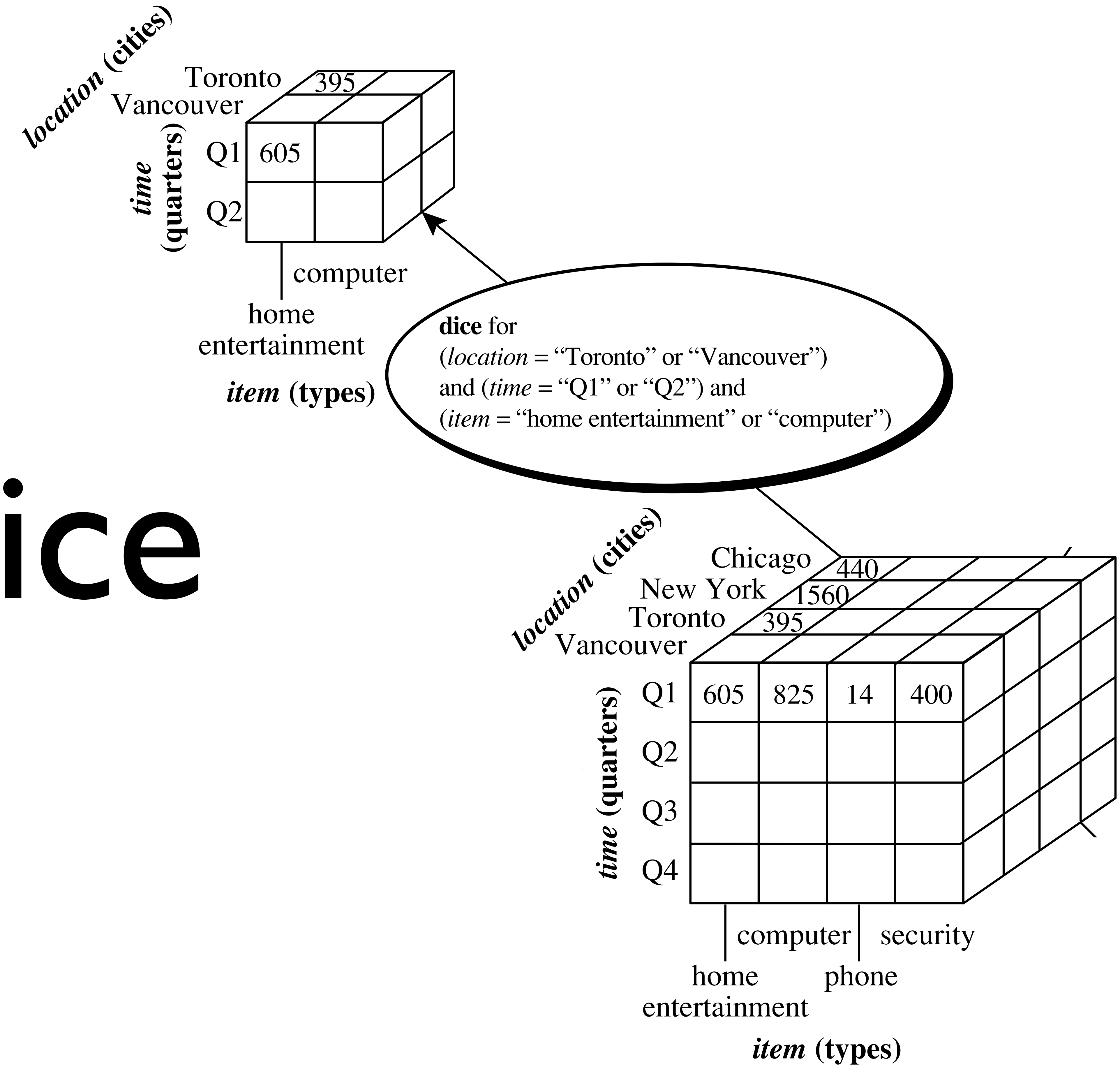


location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
computer
security
home entertainment
phone
item (types)
slice
for time = "Q1"
location (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
computer
security
home entertainment
phone
item (types)

pivot


slice
for time="Q1"
Chicago
New York
Toronto
Vancouver
605
825
14
400
location (cities)
home entertainment
computer
phone
security
item (types)
pivot
home entertainment
computer
phone
security
item (types)
605
825
14
400
Chicago
New York
Toronto
Vancouver
location (cities)

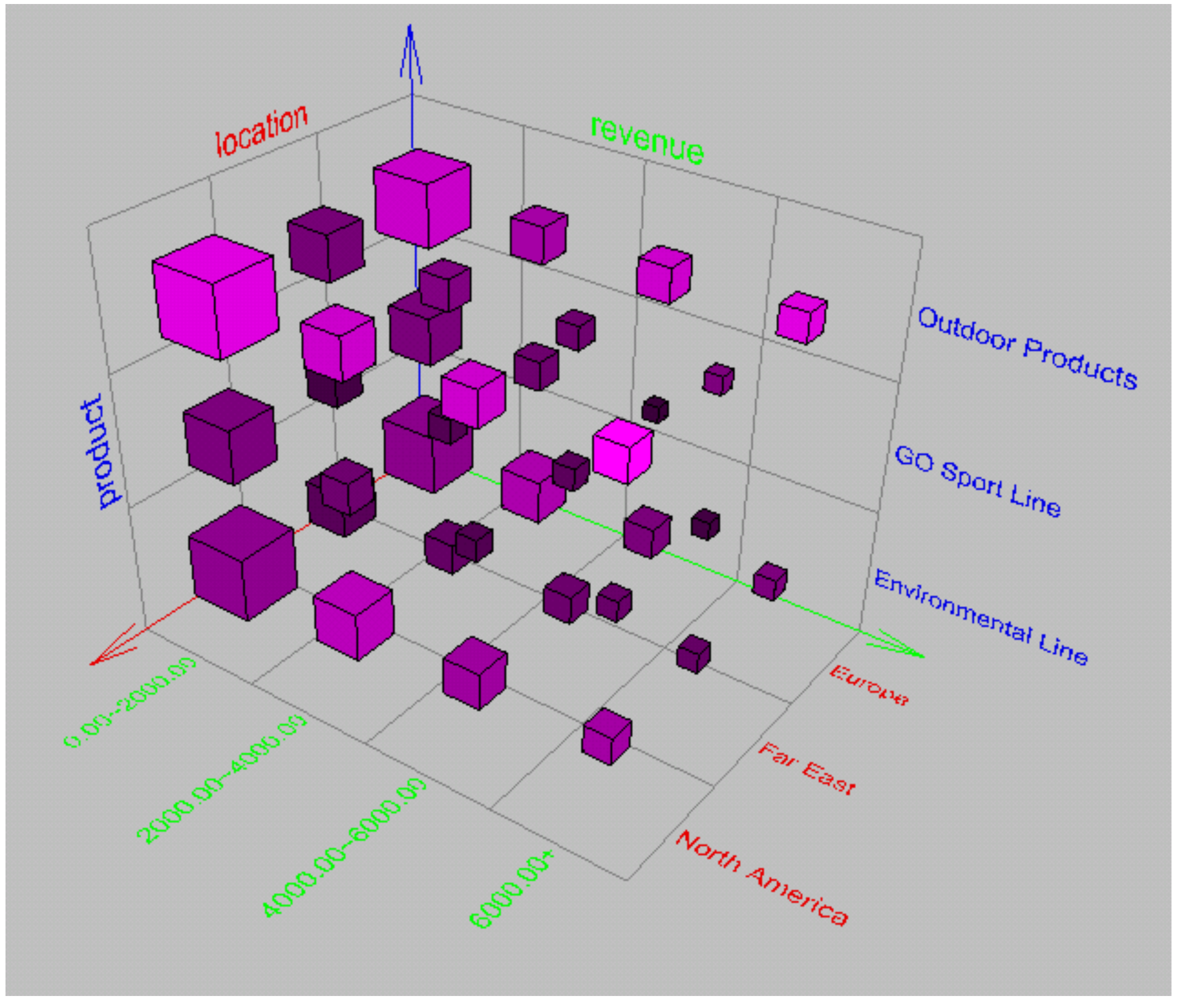
dice


location (cities)
Toronto
Vancouver
395
time (quarters)
Q1
Q2
605
computer
home entertainment
item (types)
dice for
(location = "Toronto" or "Vancouver")
and (time = "Q1" or "Q2") and
(item = "home entertainment" or "computer")
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)

browsing a data cube

Visualization

OLAP capabilities



Interactive manipulation

Objective

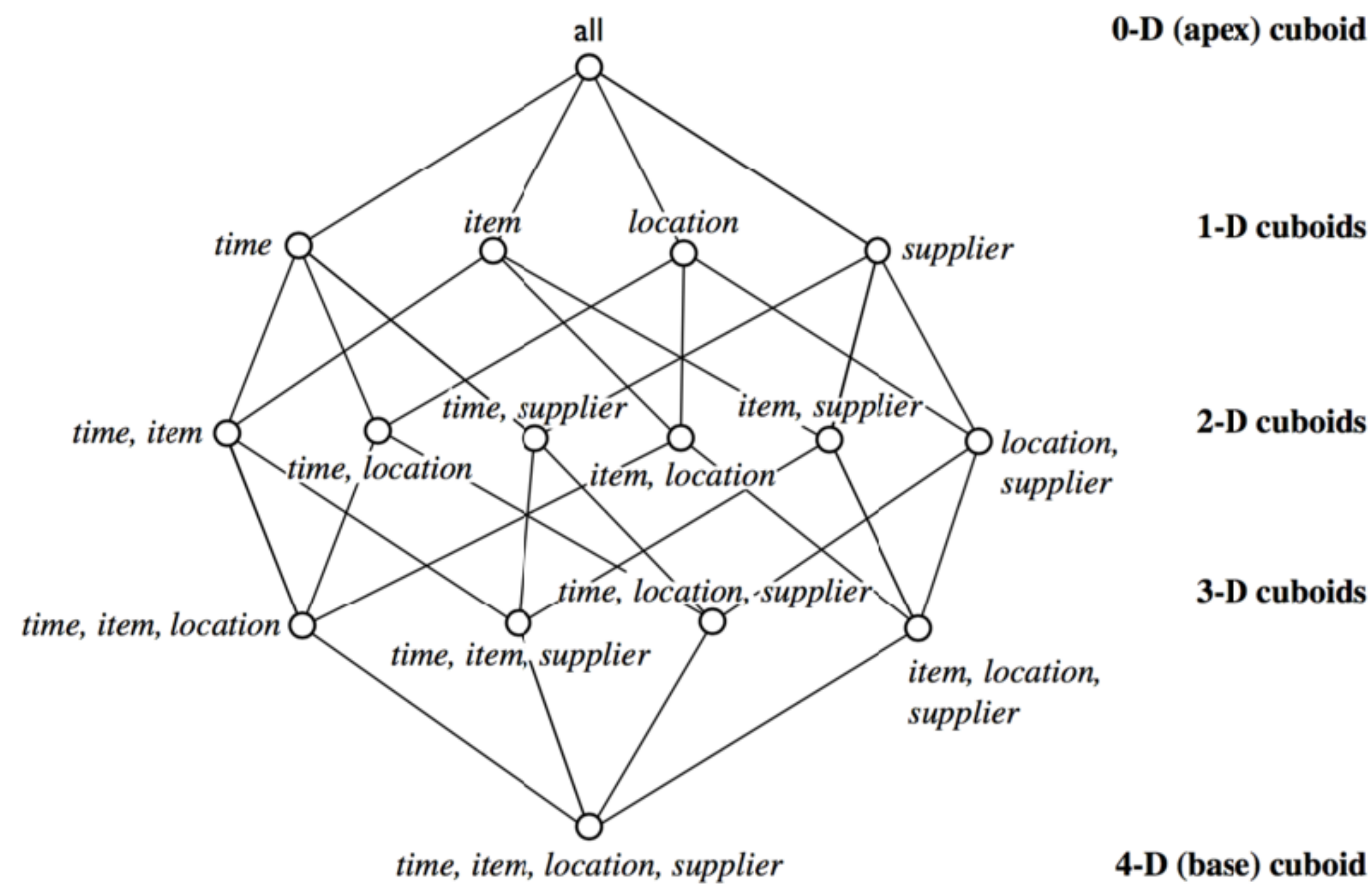
Exercise!

Understanding how **incremental updates** work in a datacube

Exercise!

Suppose we need to record two measures in a datacube: **min** and **average**.

Design an efficient computation and storage method for each measure given that the cube allows data to be deleted incrementally (i.e., in small portions at a time) from the cube.



Solution

measure=**min** amount spent

Base Data Table

#	Location	Product	Amount Spent
~~1~~	~~**Meijer**~~	~~**Milk**~~	~~**$3**~~
2	**Co-Op**	**Chocolate**	**$5**
…	**…**	**…**	**…**

	Meijer	Co-Op
Milk	**$3**	**$2**
Chocolate	**$7**	**$5**

base cuboid

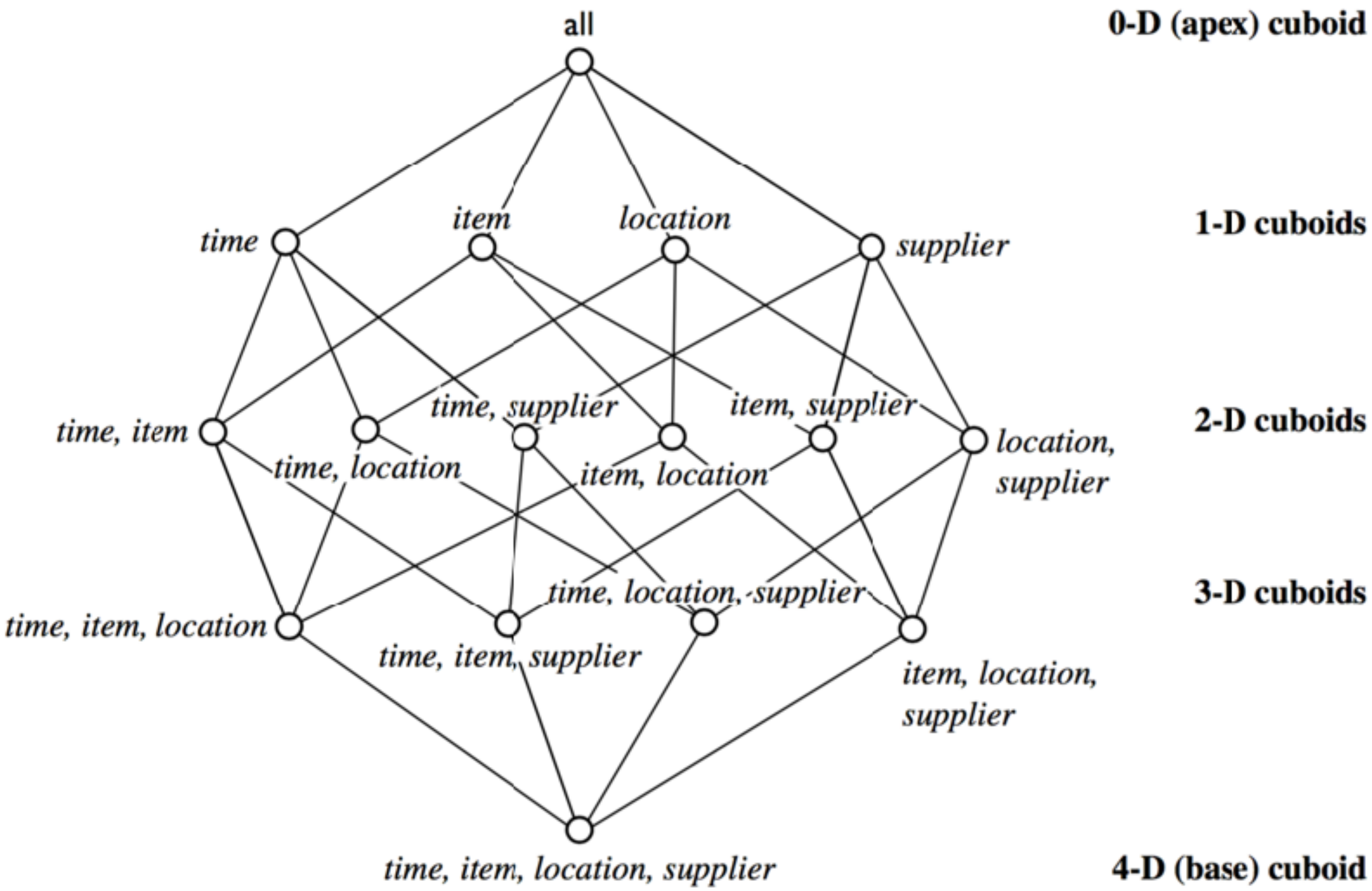
Solution

Exercise!

For **min**, keep the <**min_val**, **count**> pair for each cuboid to register the smallest value and its count.

For each deleted tuple, if its value is greater than **min_val**, do nothing.

Otherwise, decrement the count of the corresponding node. If a count goes to zero, recalculate the structure.

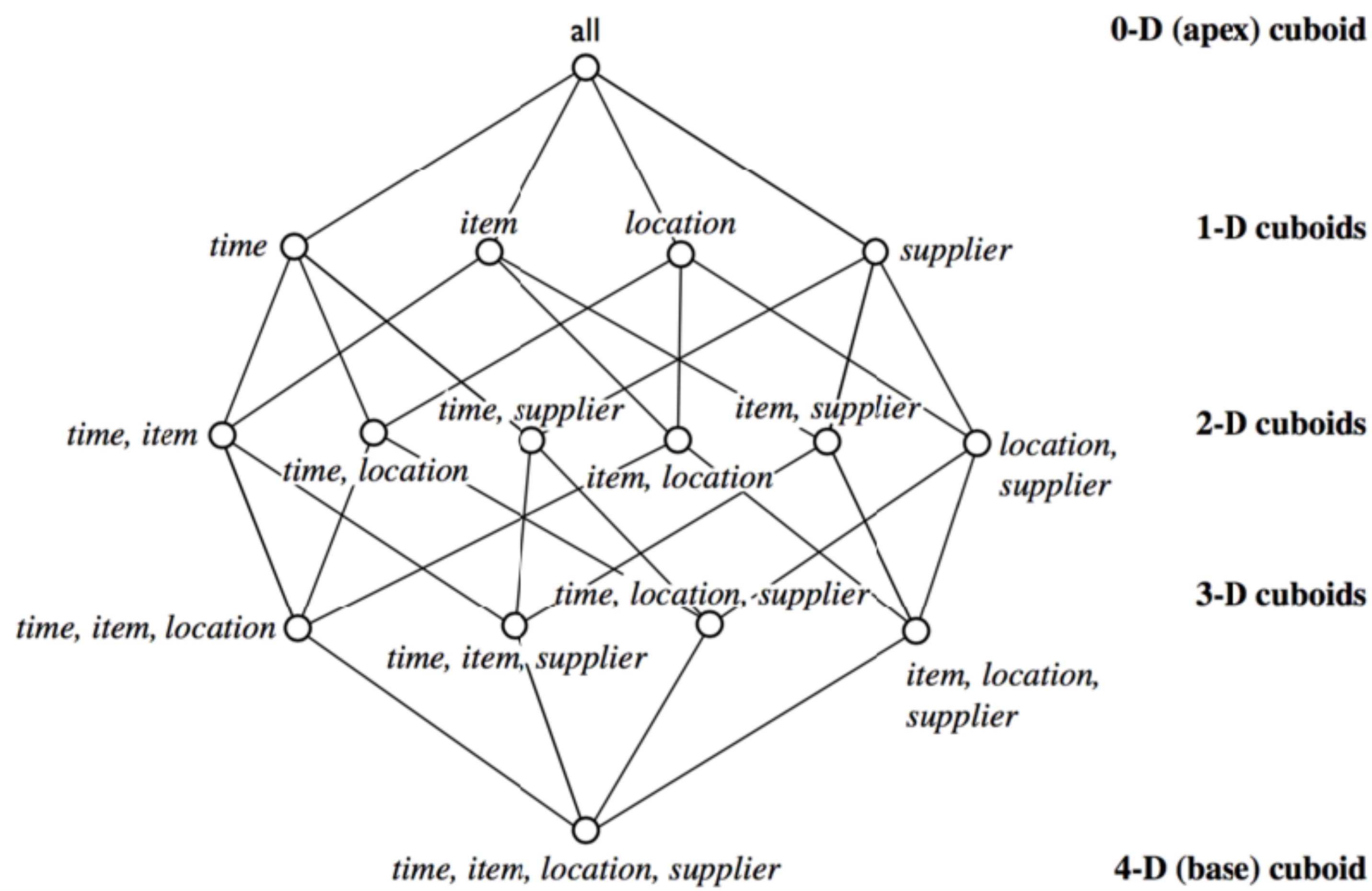


Solution

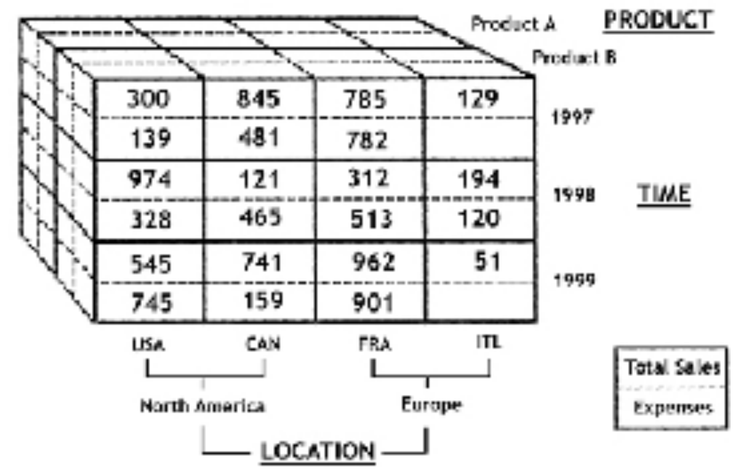
Exercise!

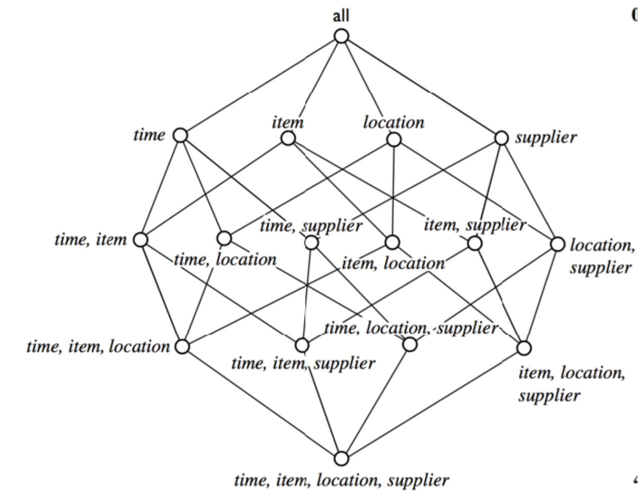
For **average**, keep a pair <**sum**, **count**> for each cuboid.

For each deleted tuple with value=N, decrement the count and subtract N from the sum, and **average** = **sum**/**count**.



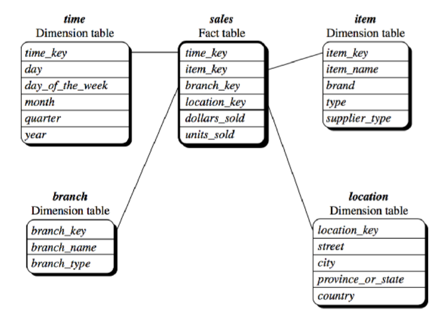
A data warehouse is based on a multidimensional data model which views data in the form of a data cube





DATA CUBE & OLAP SUMMARY

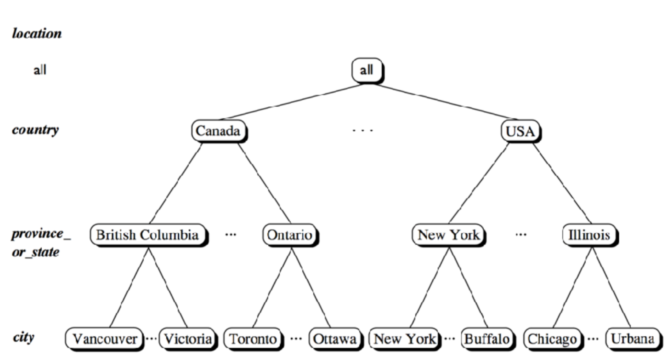
Fact and Dimension tables



star, snowflake and galaxy schemas

distributive, algebraic and holistic measures

min, average, median



concept hierarchy are essential

basic cube operations

roll up, drill down, slice, pivot, dice

WAREHOUSE USAGE

Basic Concepts Data Cube and OLAP Implementation Summary

USAGE

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

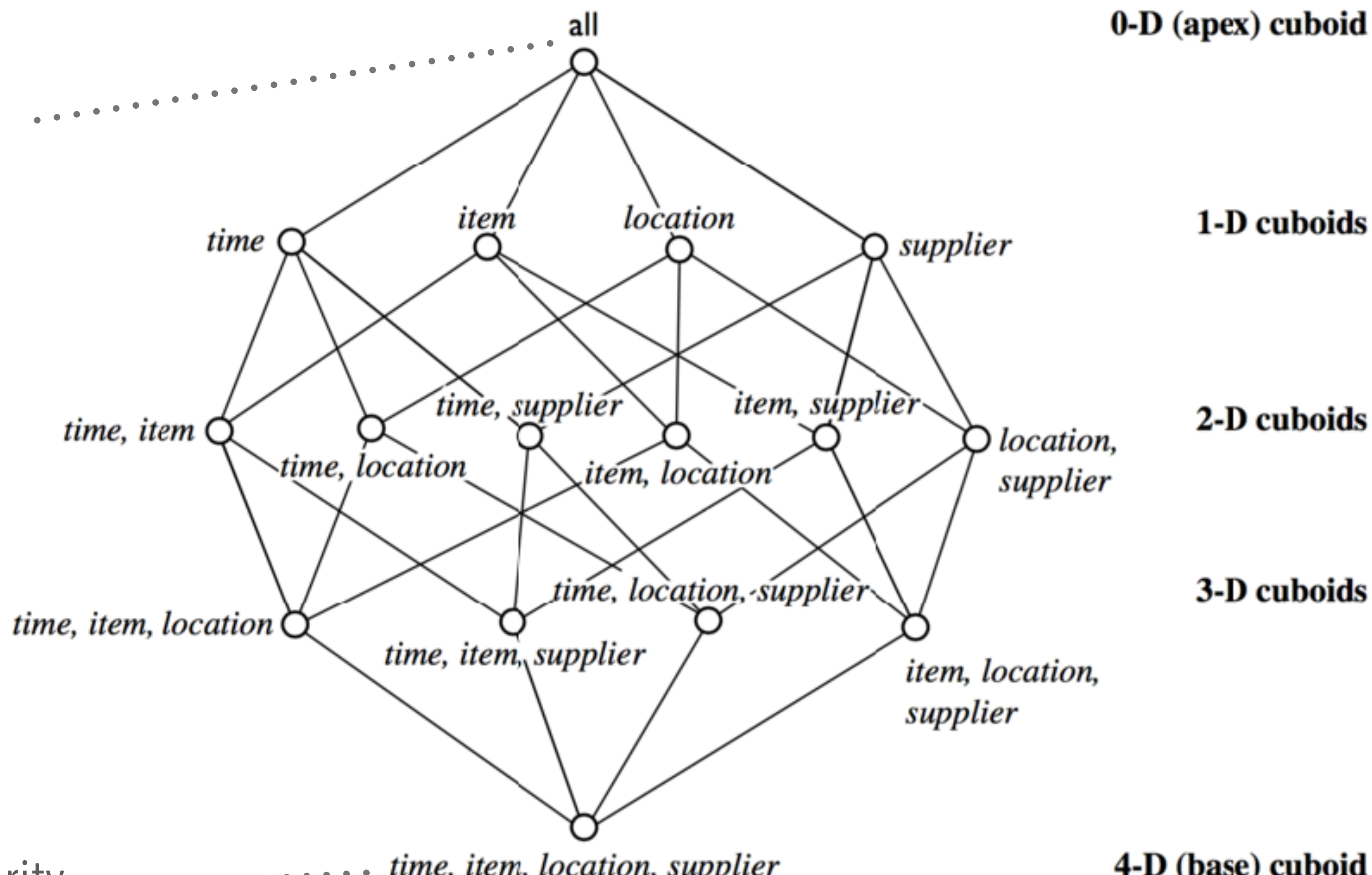
supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools

IMPLEMENTATION

how many cuboids?

highest-level of summarization

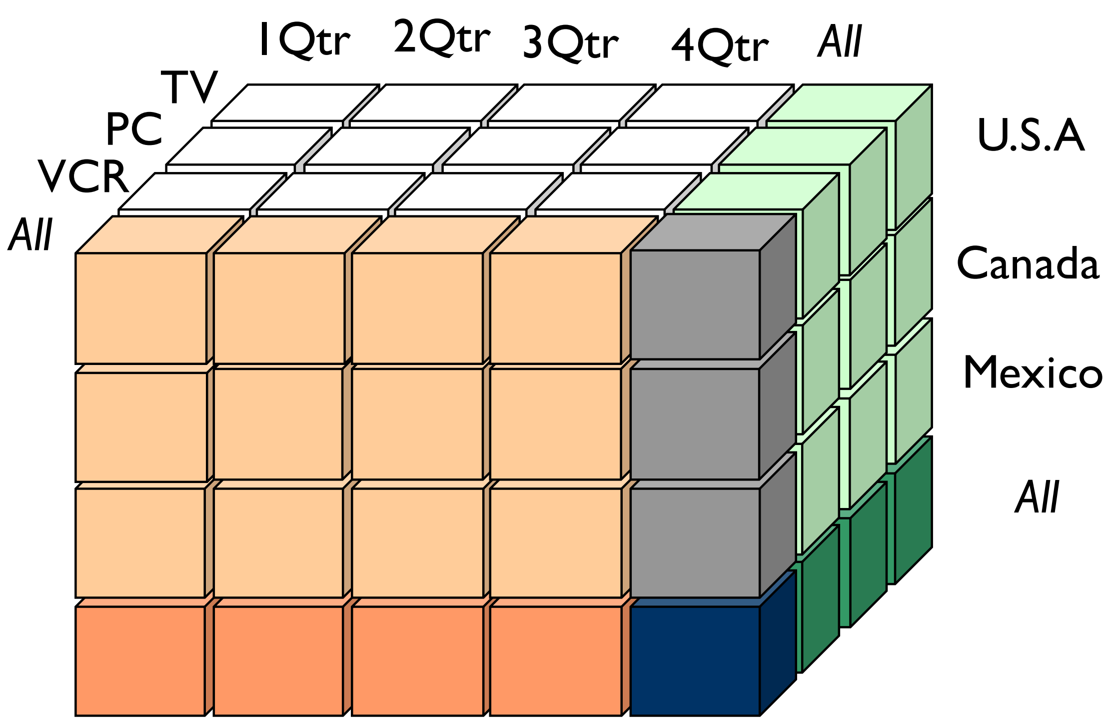
$$T = \prod_{i=1}^{n} (L_i + 1)$$

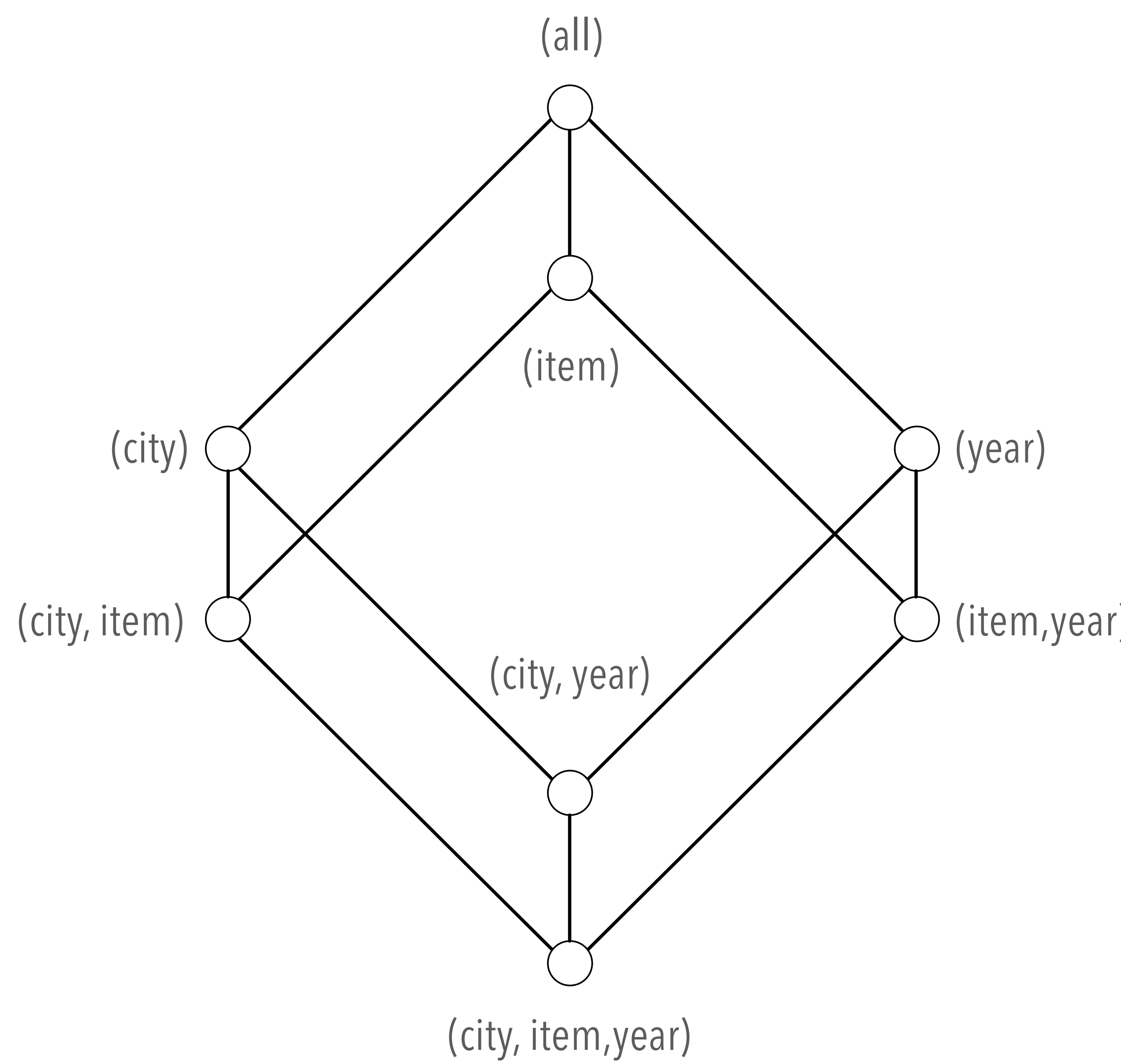


finest granularity

A data cuboid is a subset of data cube.

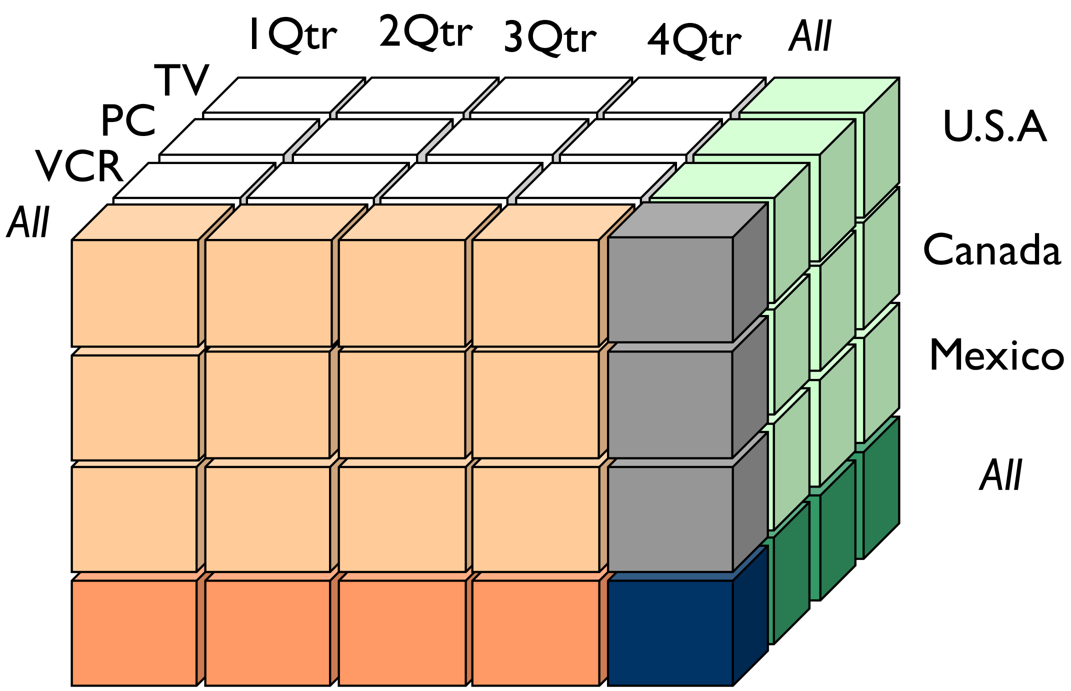
to see why $T = \prod_{i=1}^{n} (L_i + 1)$



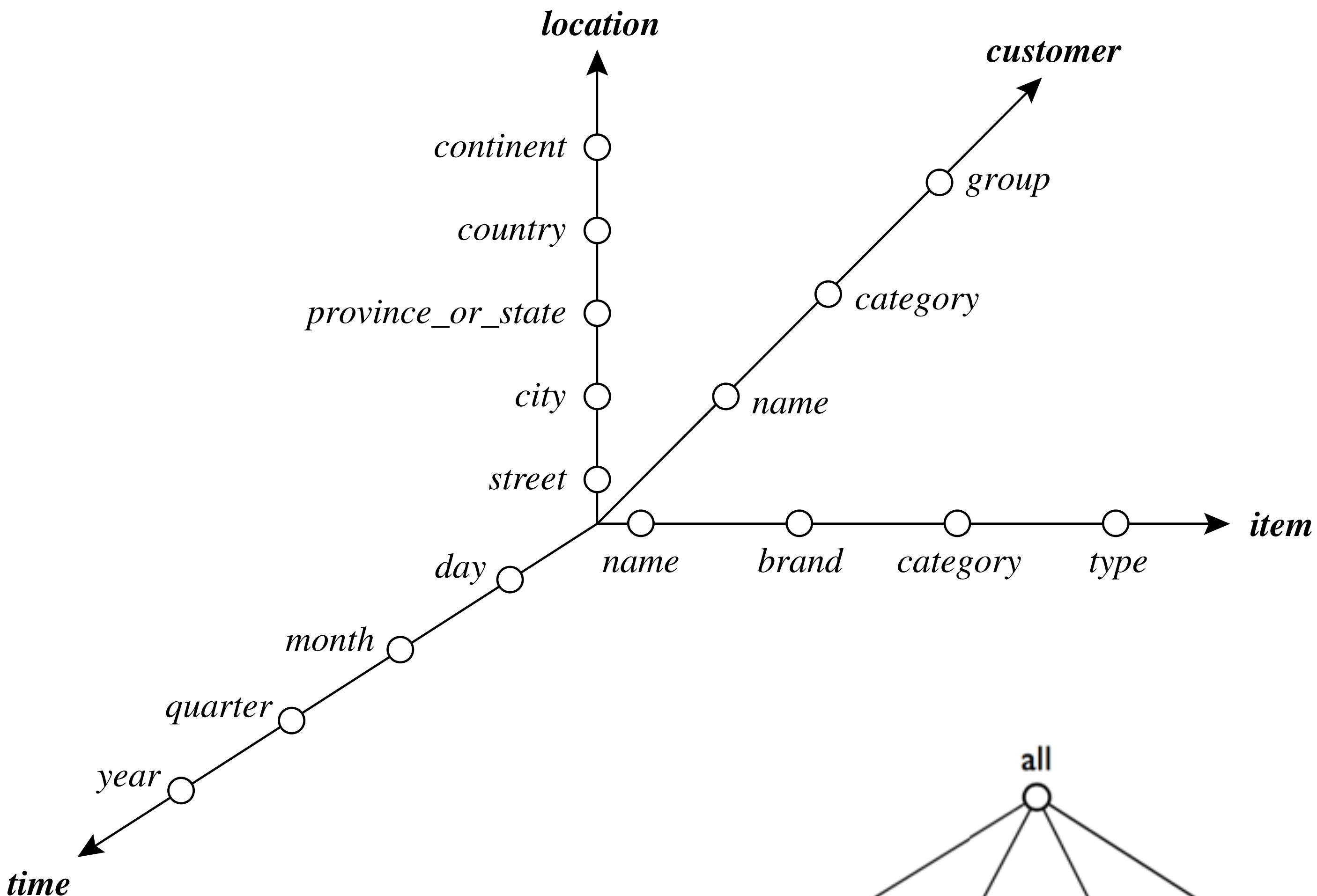


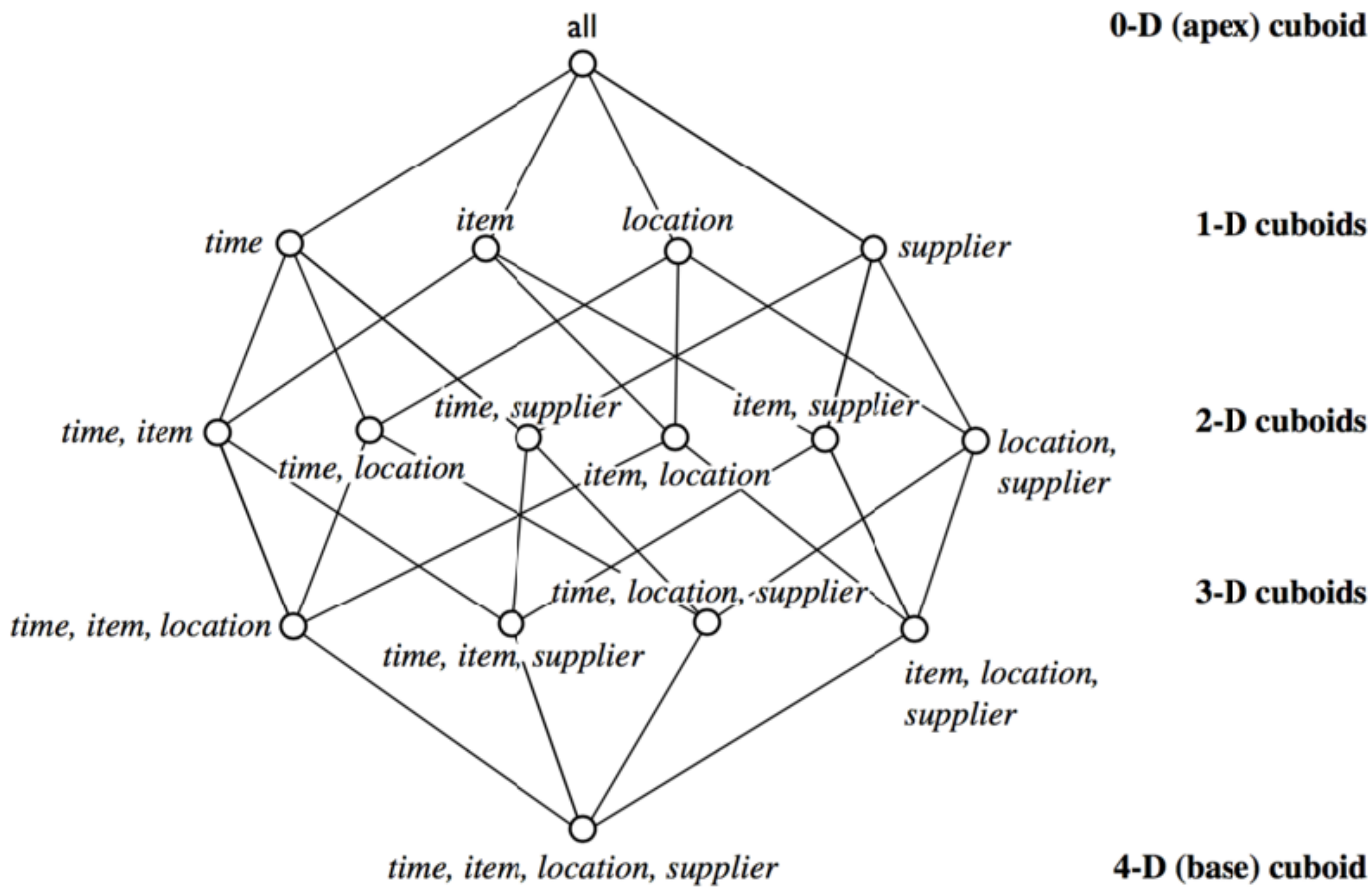
what is a challenge here?

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

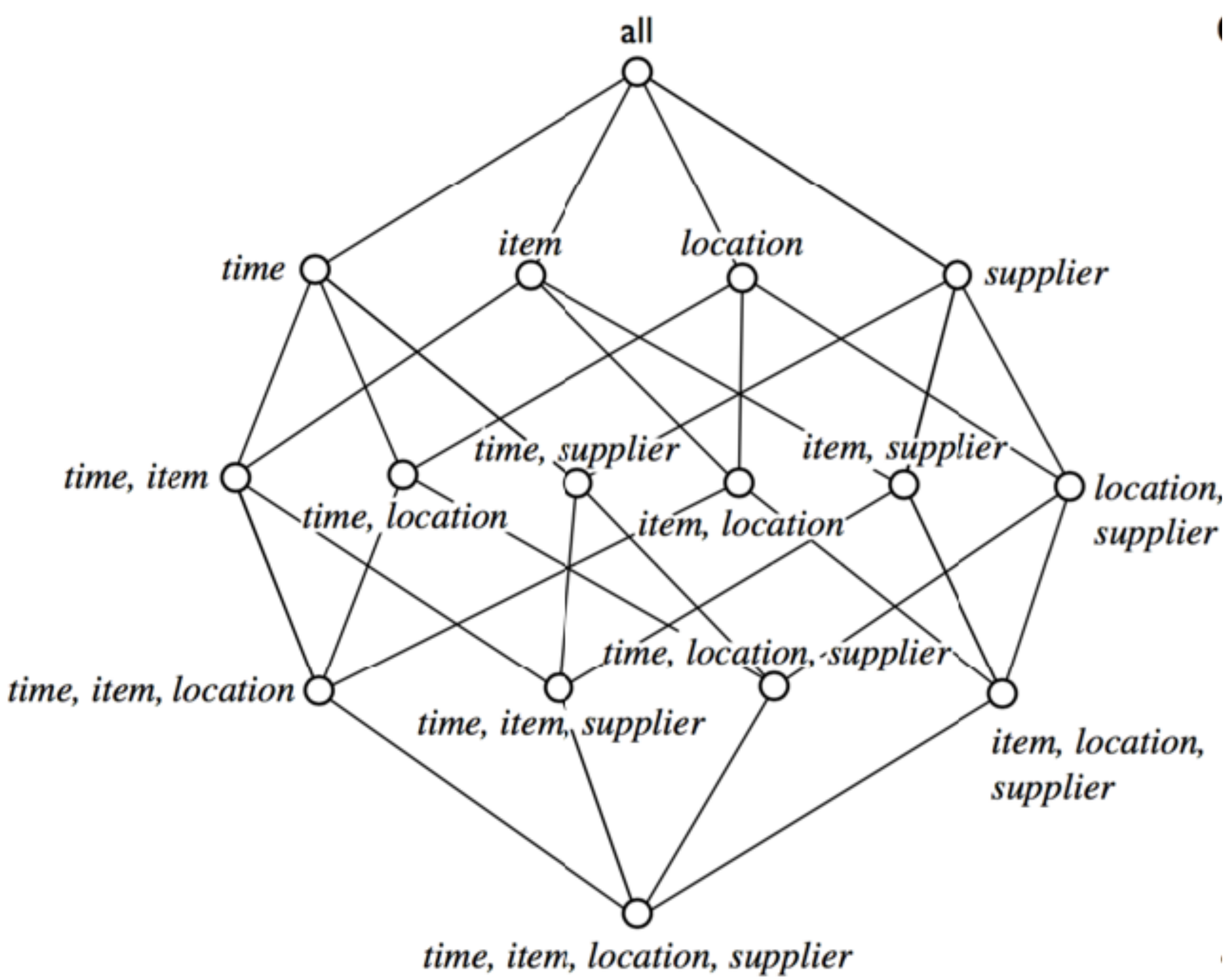


starnet query model





CUBE MATERIALIZATION



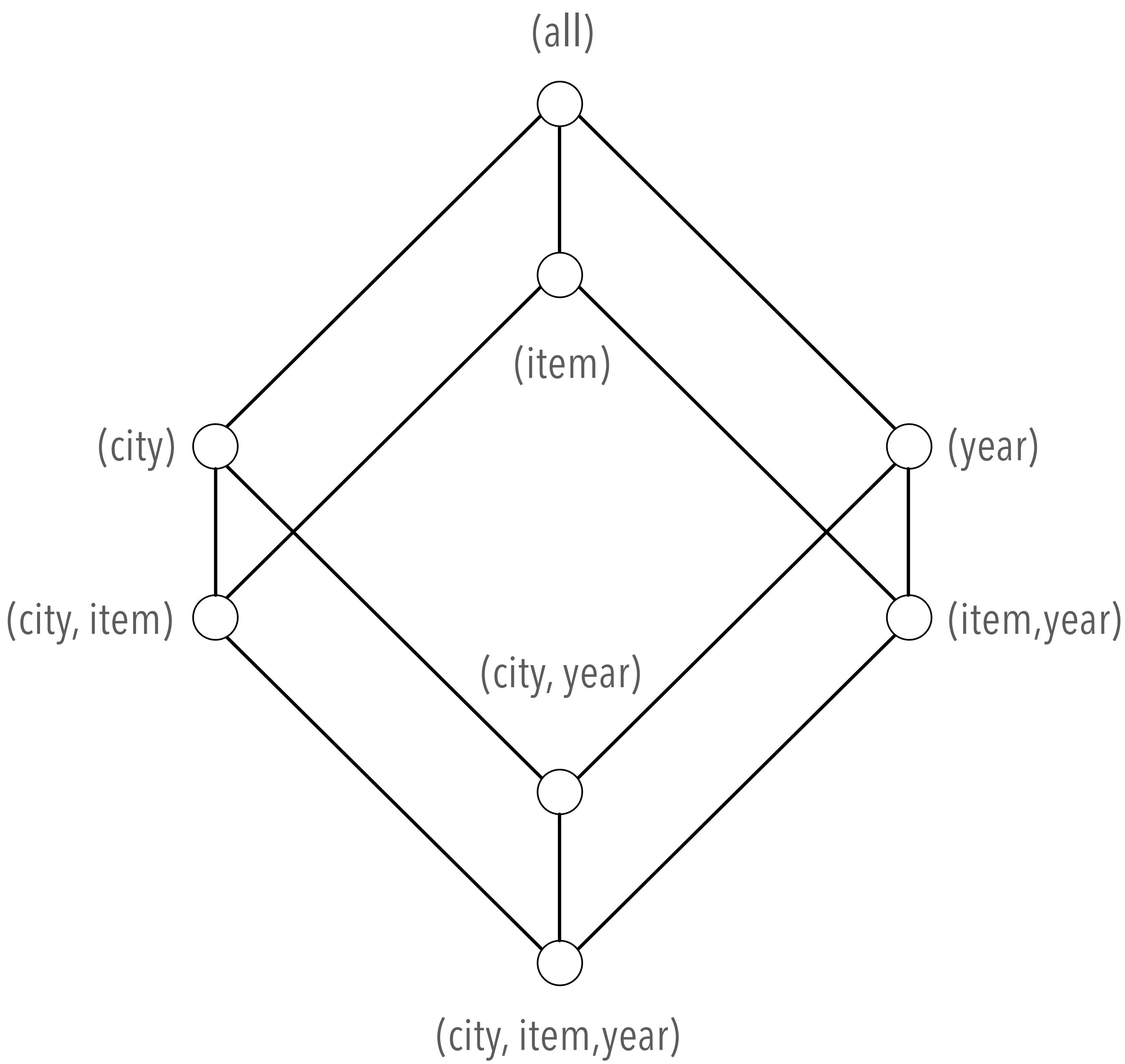
Options:

- Full: materialize every cuboid
- None: no materialization
- Partial: partial materialization

Selection of which cuboids to materialize

- Based on size, sharing, access frequency, etc.

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.'96)

SELECT item, city, year, SUM (amount)

FROM SALES

CUBE BY item, city, year

Need compute the following Group-Bys

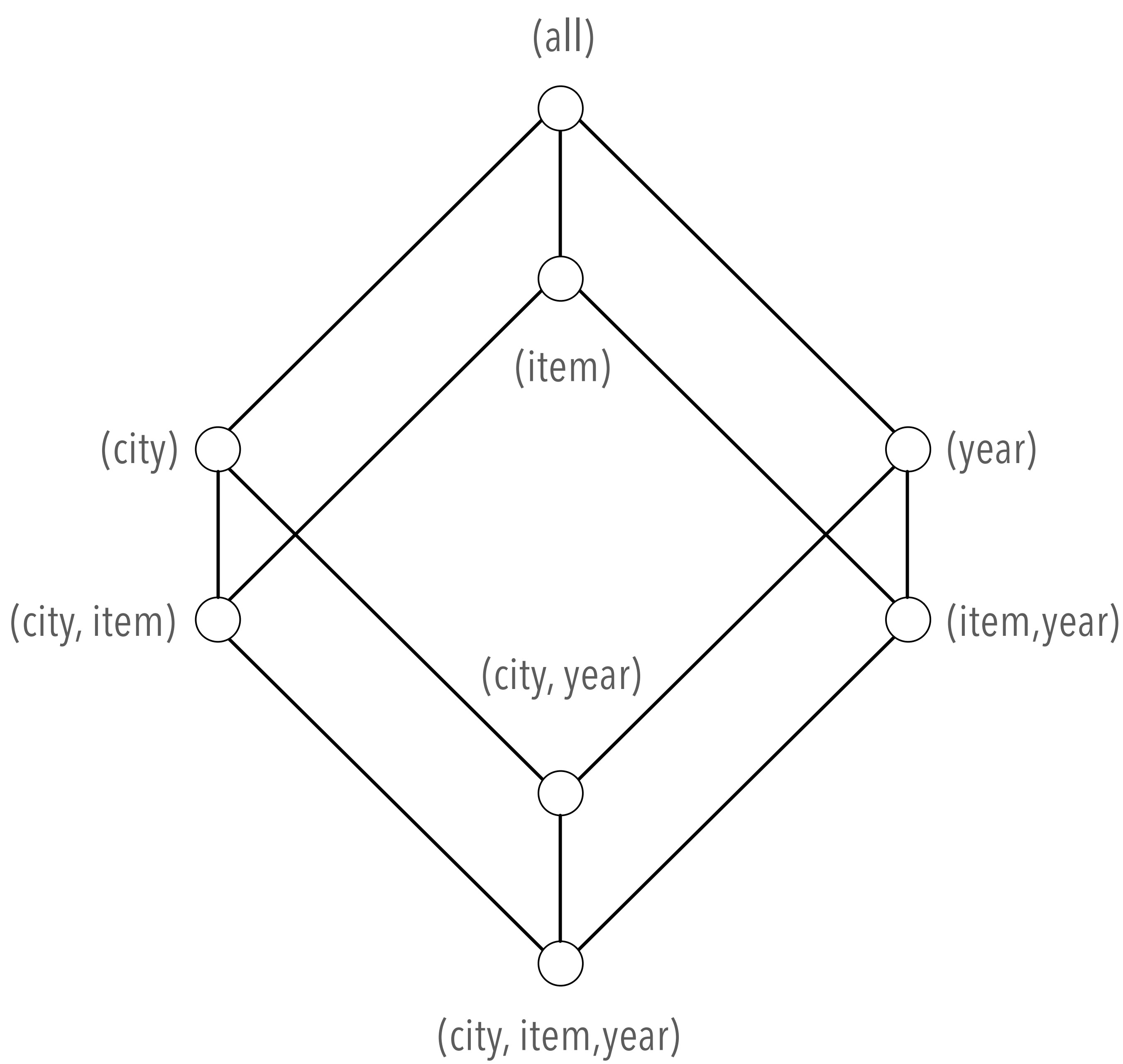
(date, product, customer),

(date,product),(date, customer), (product, customer),

(date), (product), (customer)

()

EFFICIENT 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?

Explore indexing structures and compressed vs. dense array structs in MOLAP

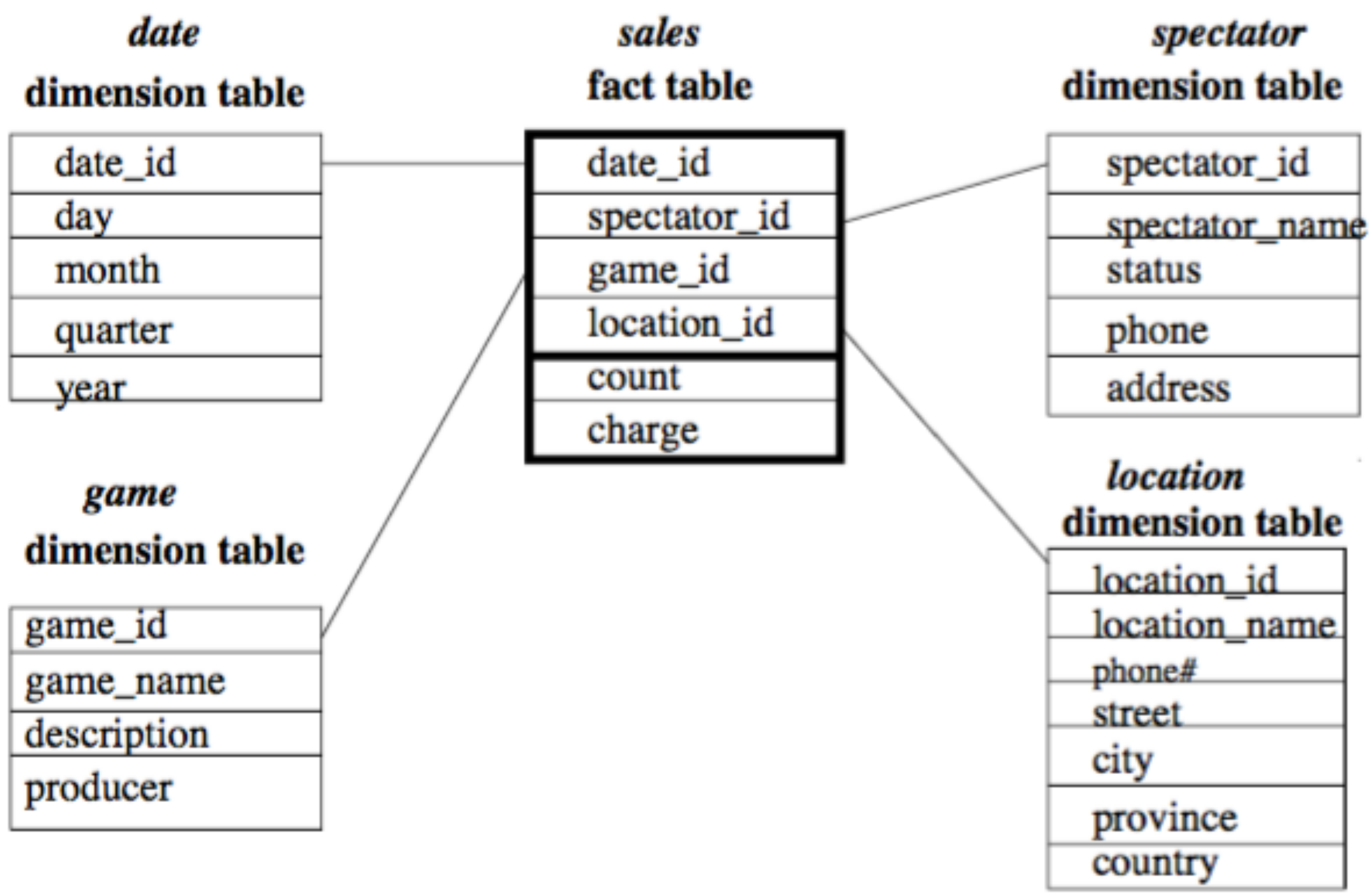
Objective

Exercise!

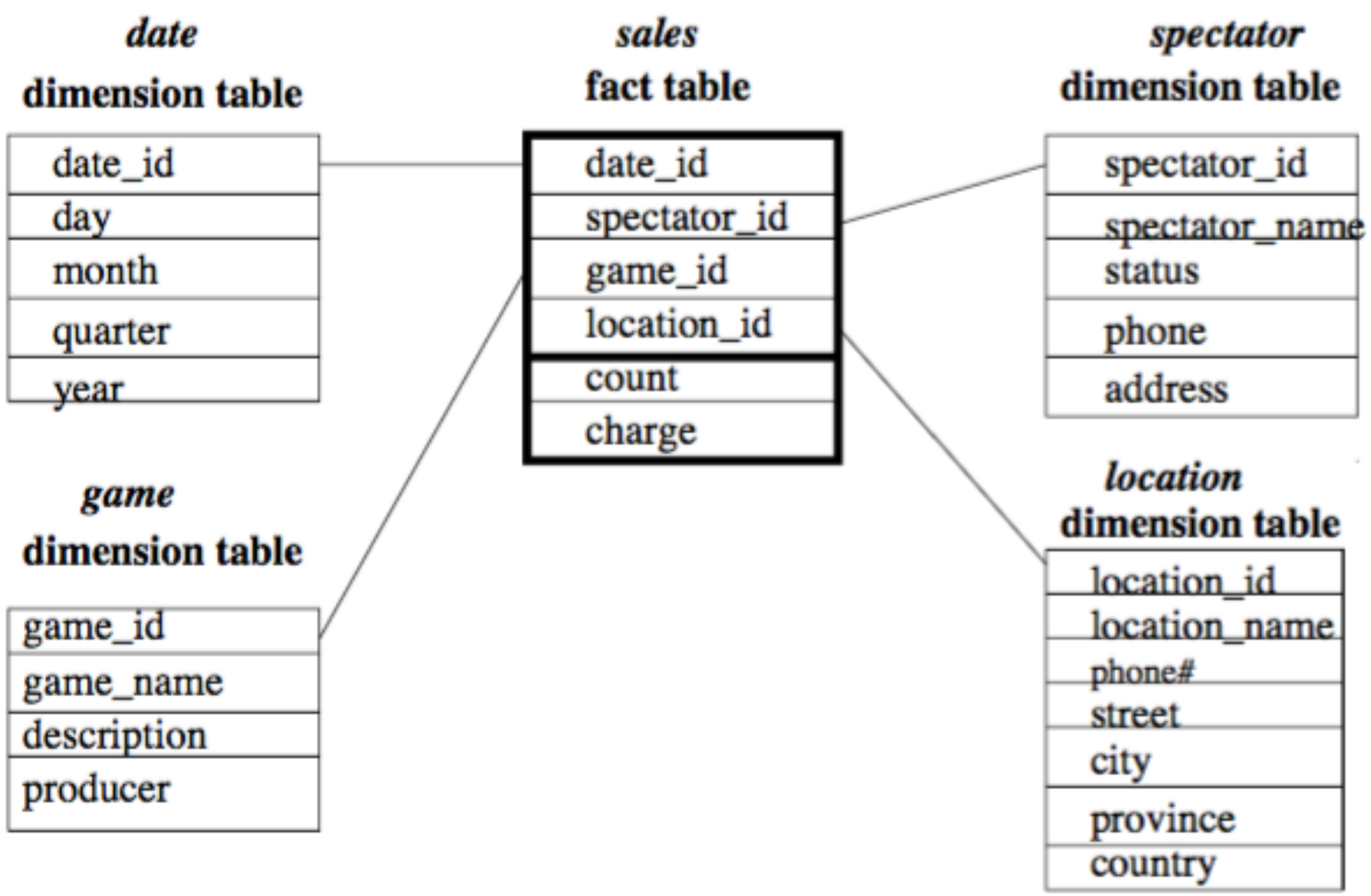
Understanding how **querying** works in a datacube

Exercise!

Starting with the base cuboid [**date**, **spectator**, **location**, **game**], what specific OLAP operations should one perform in order to list the total charge paid by student **spectators** at **location** GM_Place in **year** 2010?



SOLUTION



Roll-up on **date** from date_id to year

Roll-up on **game** from game_id to all.

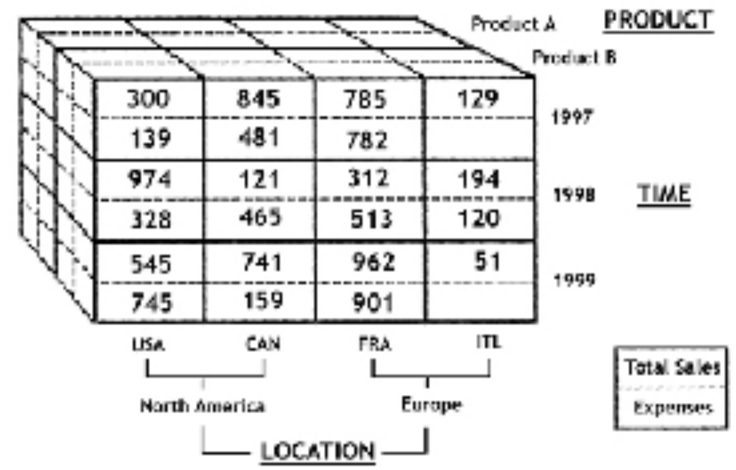
Roll-up on **location** from location_id to location_name

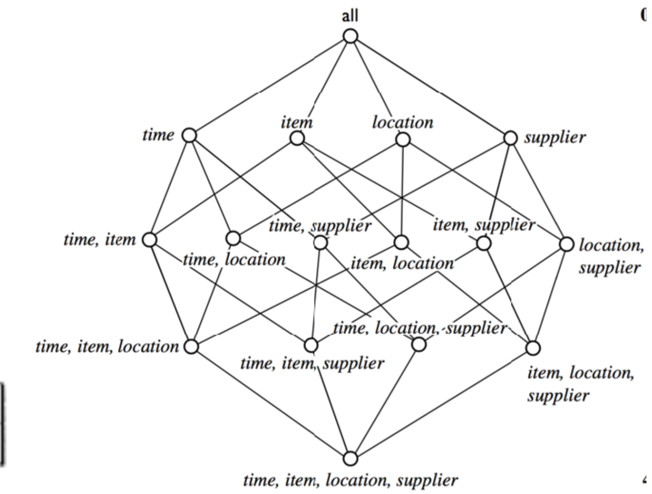
Roll-up on **spectator** from spectator_id to status

Dice with **status** = "students", location_name = "GM_Place", and year = 2010.

SUMMARY

A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.

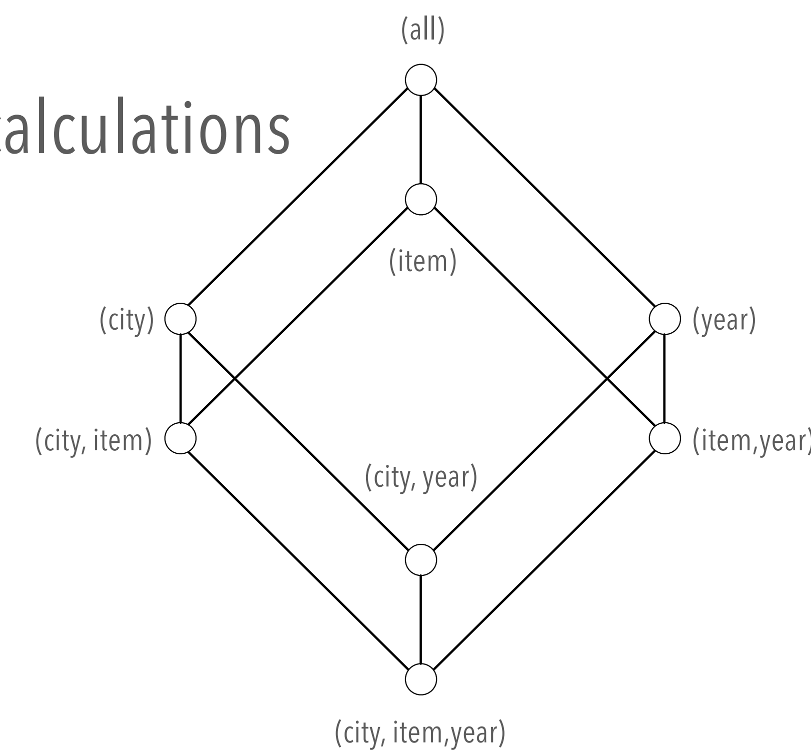




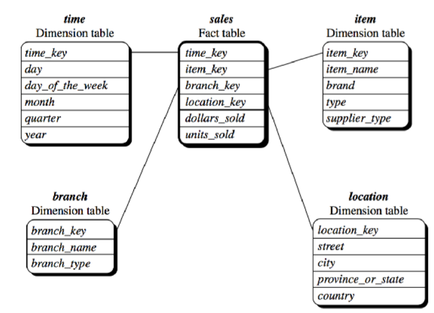
Efficient Data Cube calculations

Full, partial, no materialization

cube operator



Fact and Dimension tables



star, snowflake and galaxy schemas

distributive, algebraic and holistic measures