



CS 412 Intro. to Data Mining

Chapter 4. Data Warehousing and On-line Analytical Processing

Jiawei Han, Computer Science, Univ. Illinois at Urbana-Champaign, 2017



Chapter 4: Data Warehousing and On-line Analytical Processing

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



What is a 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
- “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

នៅរាជធានីភ្នំពេញមាត្រា ?!

Data Warehouse—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**

↪ ຂົດມາບກວມທີ່ນີ້

Data Warehouse—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
 - When data is moved to the warehouse, it is converted

Data Warehouse—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”

Data Warehouse—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.

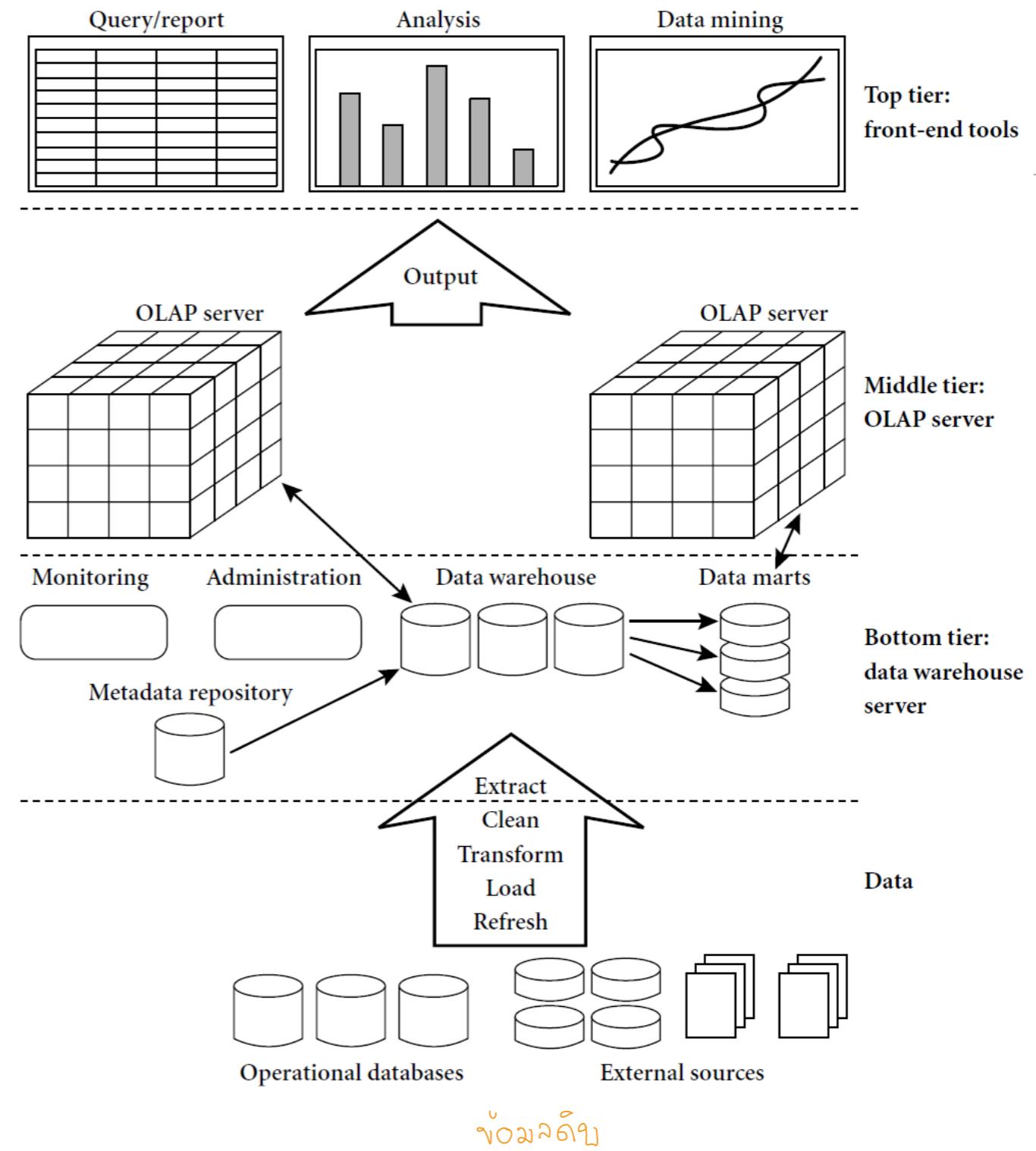
	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day-to-day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
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:
 - 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 analysis directly on relational databases

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**
 - Collects all of the information about subjects spanning the entire organization
- Data Mart**
 - A subset of corporate-wide data that is of value to a specific groups of users
 - Its scope is confined to specific, selected groups, such as marketing data mart
 - 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

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 defn, 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

Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- 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
- 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 0-D cuboid, which holds the highest-level of summarization, is called the **apex cuboid**
 - The lattice of cuboids forms a **data cube**.

2D

Table 4.2: A 2-D view of sales data for *AllElectronics* according to the dimensions *time* and *item*, where the sales are from branches located in the city of Vancouver. The measure displayed is *dollars-sold* (in thousands).

location = “Vancouver”

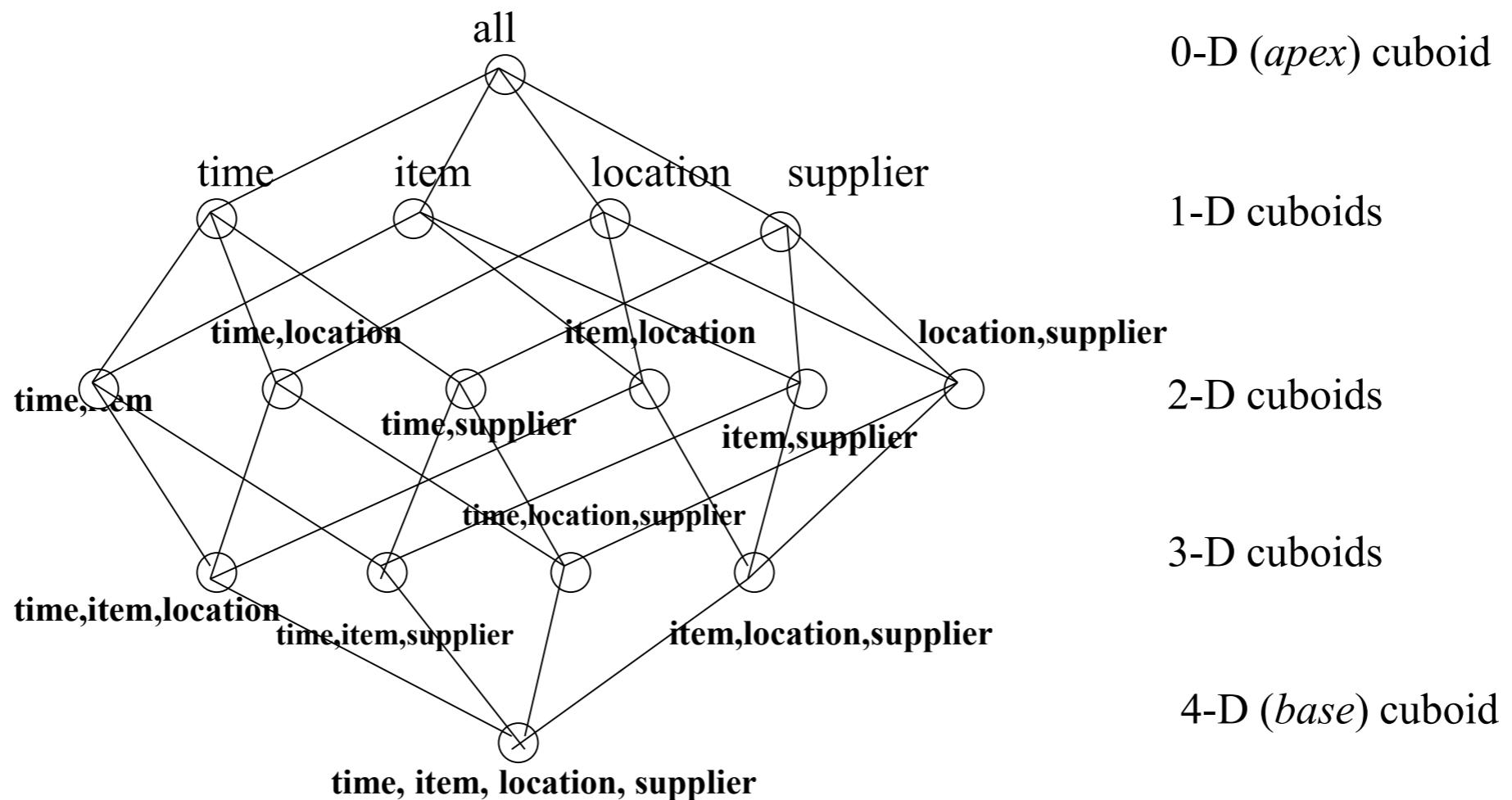
<i>time</i> (quarter)	<i>item</i> (type)			
	<i>home</i>	<i>entertainment</i>	<i>computer</i>	<i>phone</i>
Q1	605		825	14
Q2	680		952	31
Q3	812		1023	30
Q4	927		1038	38

3D

Table 4.3: A 3-D view of sales data for *AllElectronics*, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

<i>location</i> = "Chicago"				<i>location</i> = "New York"				<i>location</i> = "Toronto"				<i>location</i> = "Vancouver"				
<i>item</i>				<i>item</i>				<i>item</i>				<i>item</i>				
<i>home</i>				<i>home</i>				<i>home</i>				<i>home</i>				
<i>time</i>	ent.	comp.	phone	<i>time</i>	ent.	comp.	phone	<i>time</i>	ent.	comp.	phone	<i>time</i>	ent.	comp.	phone	
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

Data Cube: A Lattice of Cuboids

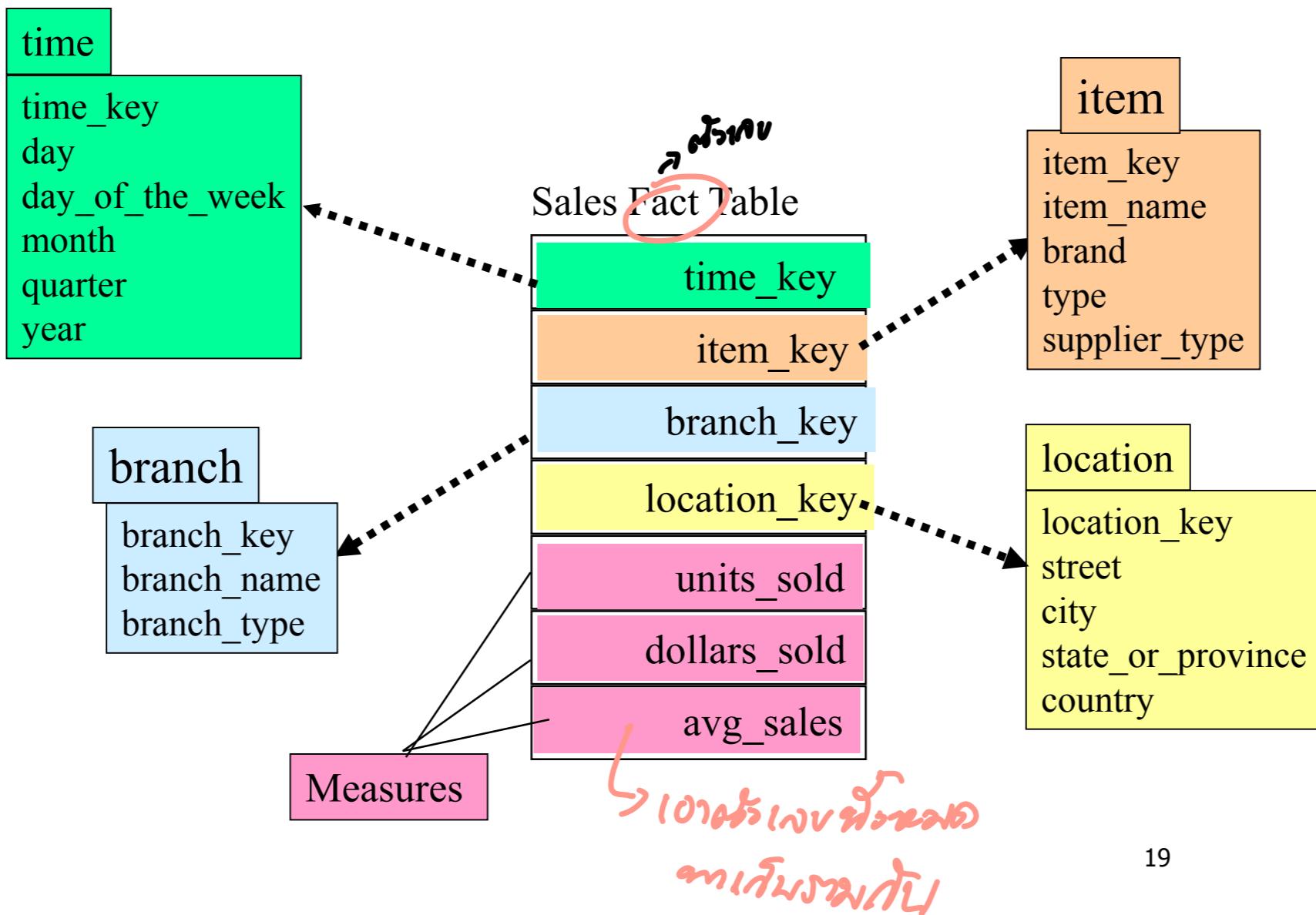


Conceptual Modeling of Data Warehouses

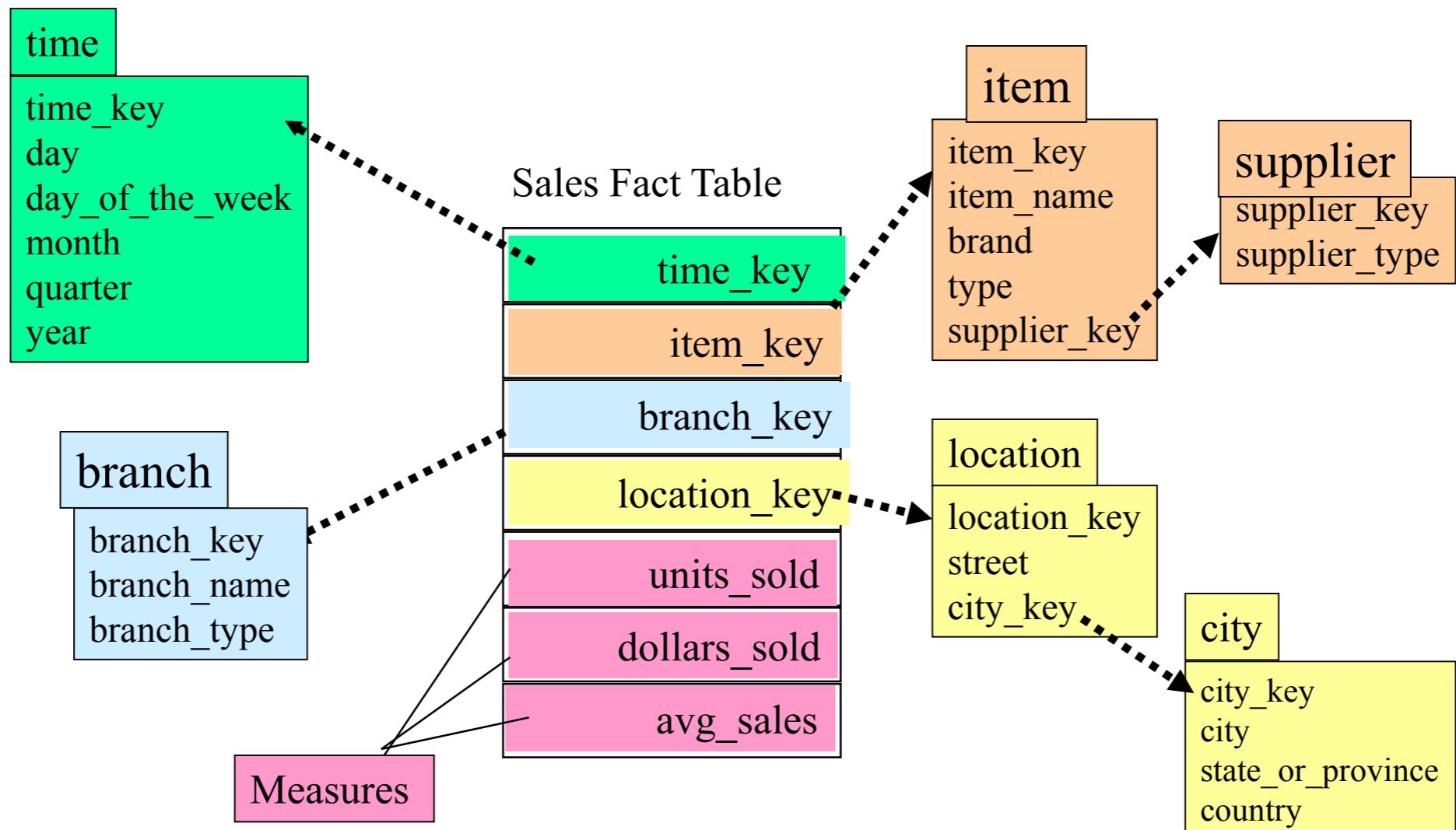
- Modeling data warehouses: dimensions & measures
 - Star schema: A fact table in the middle connected to a set of dimension tables
 - Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake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

వ్యాపకాలు, ఉంగాకాలాలు మొదటి Dimension
అధికారికాలుగా దృఢమైన

Star Schema: An Example



Snowflake Schema: An Example

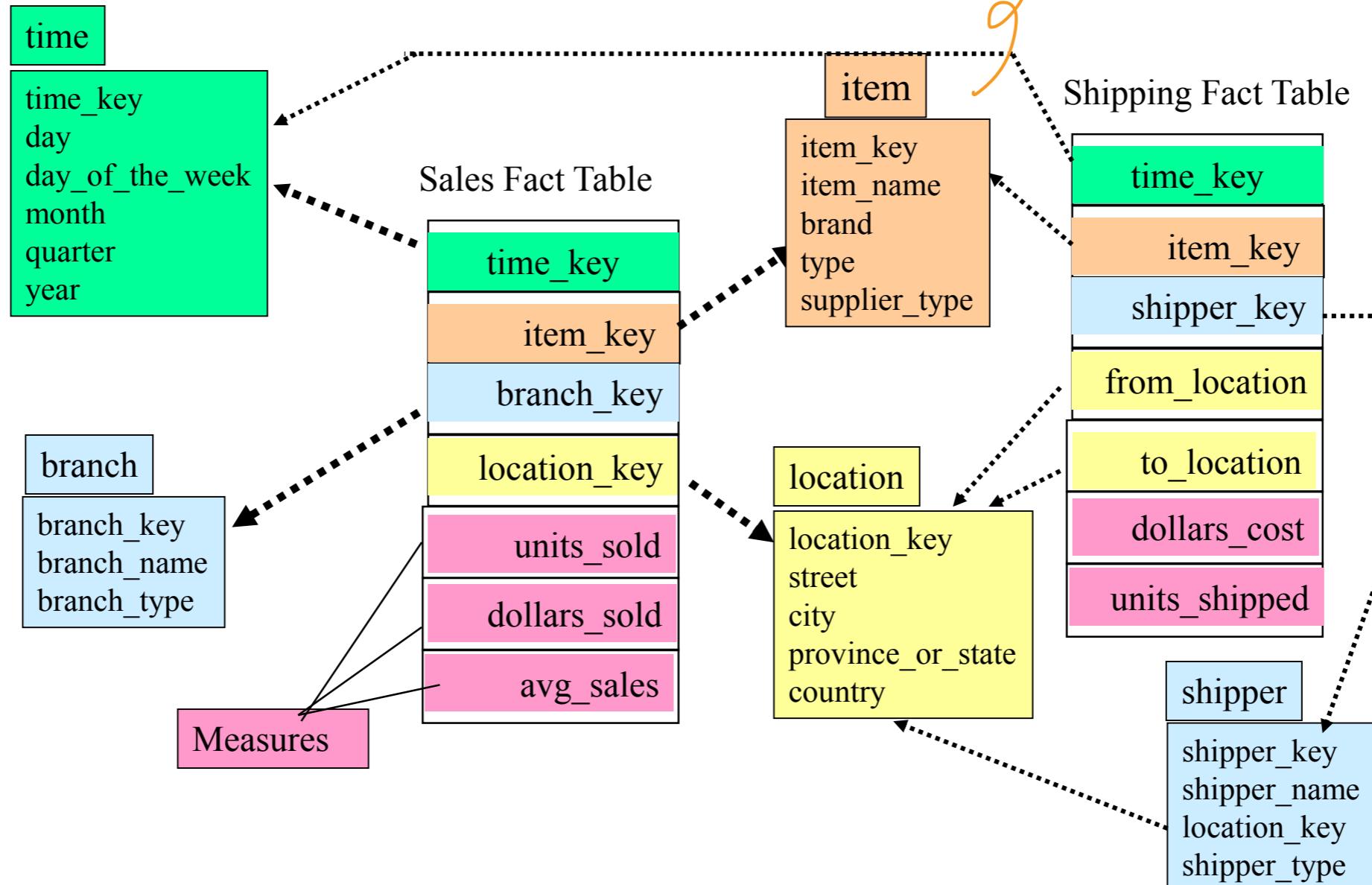


ជីវិទ្យាអំពីការបង្កើតរបស់ Fact នៃសម្រាប់ការបង្ហាញ

Dimension នូវការ

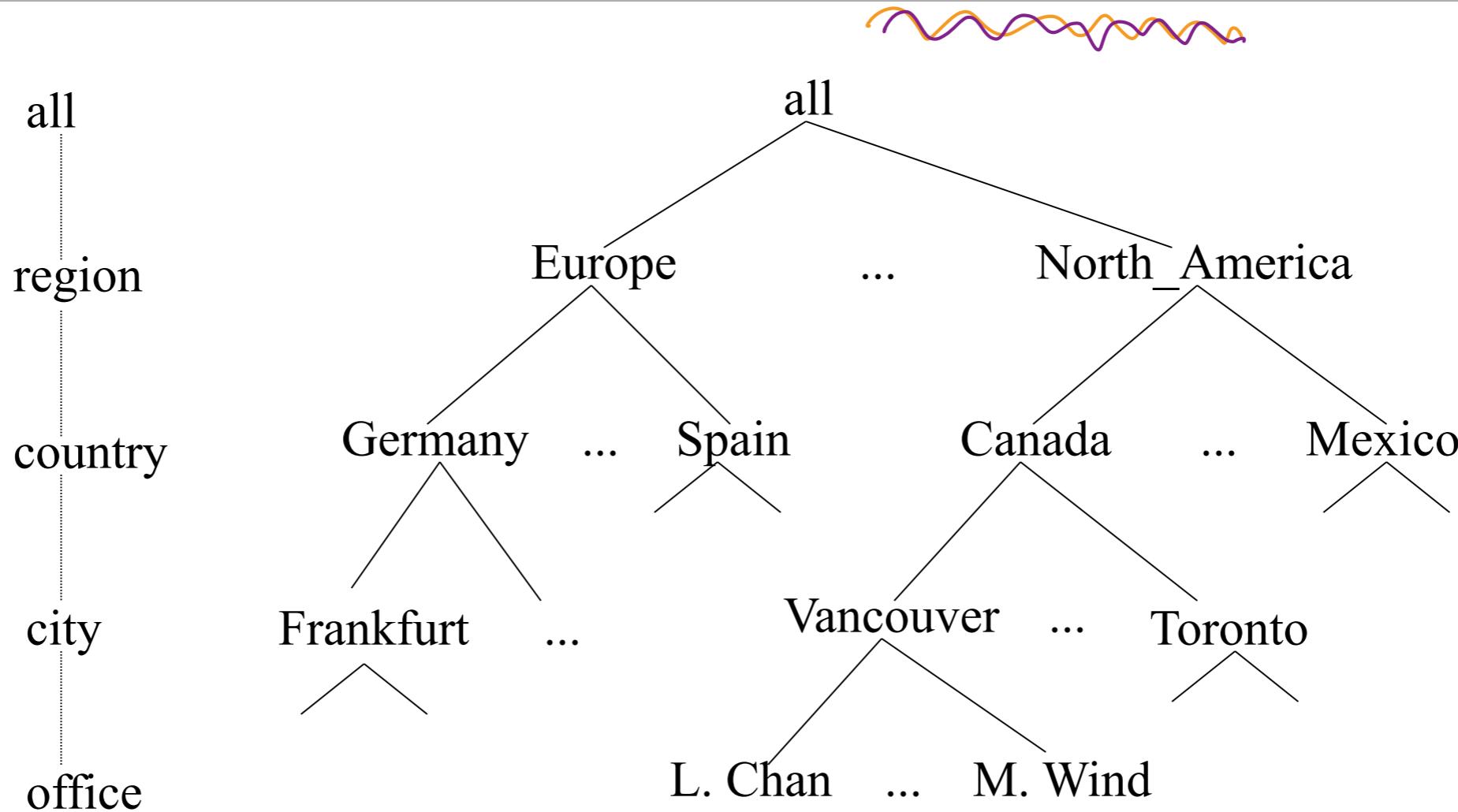
ករណីអាជីវ

Fact Constellation: An Example





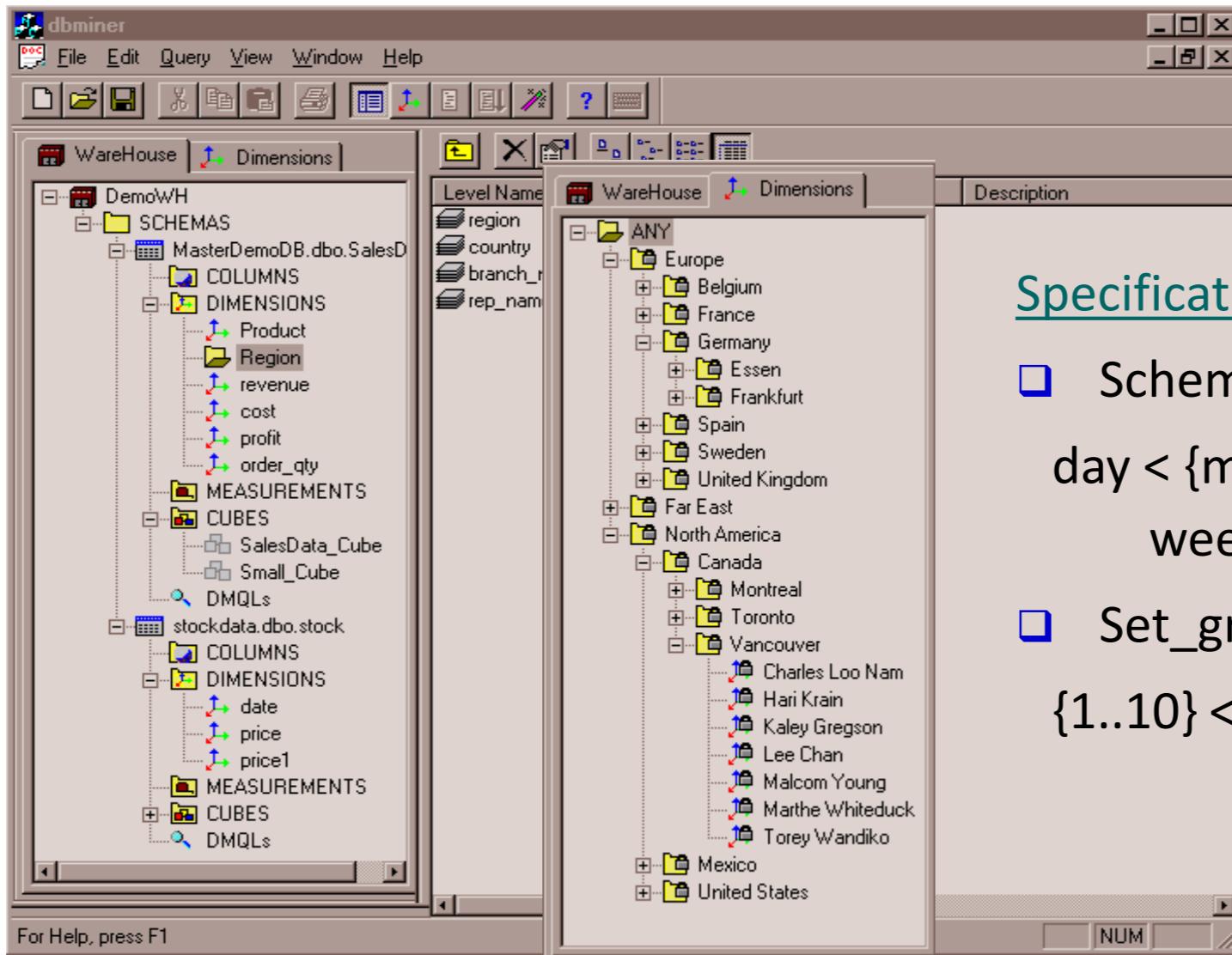
A Concept Hierarchy for a Dimension (location)



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
 - $\text{avg}(x) = \text{sum}(x) / \text{count}(x)$
 - Is $\text{min_N}()$ an algebraic measure? How about $\text{standard_deviation}()$?
- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
 - E.g., median(), mode(), rank()

View of Warehouses and Hierarchies



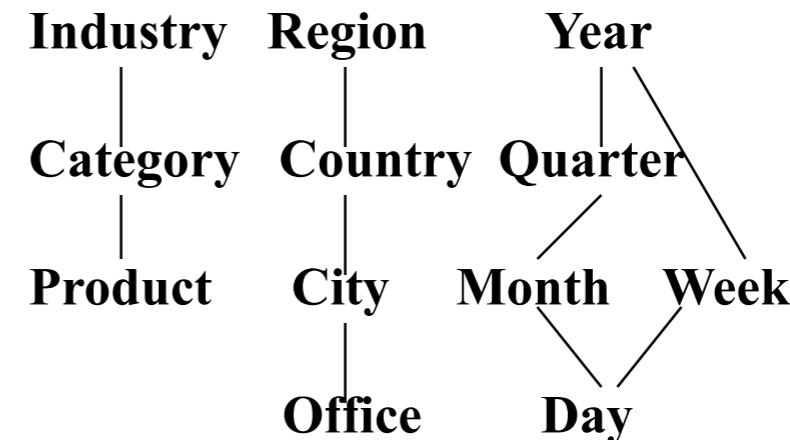
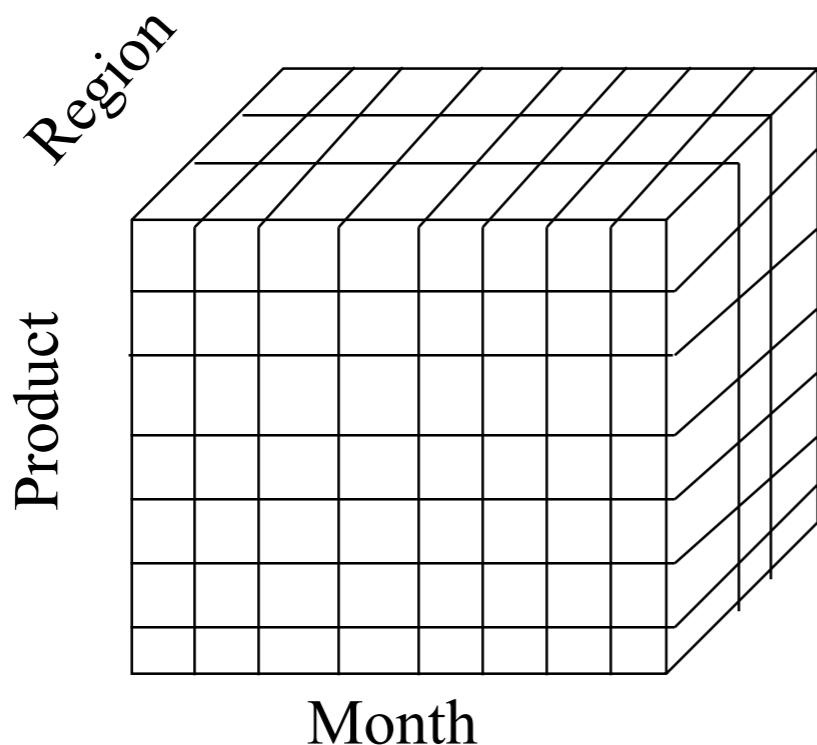
Specification of hierarchies

- Schema hierarchy
day < {month < quarter;
week} < year
- Set_grouping hierarchy
{1..10} < inexpensive

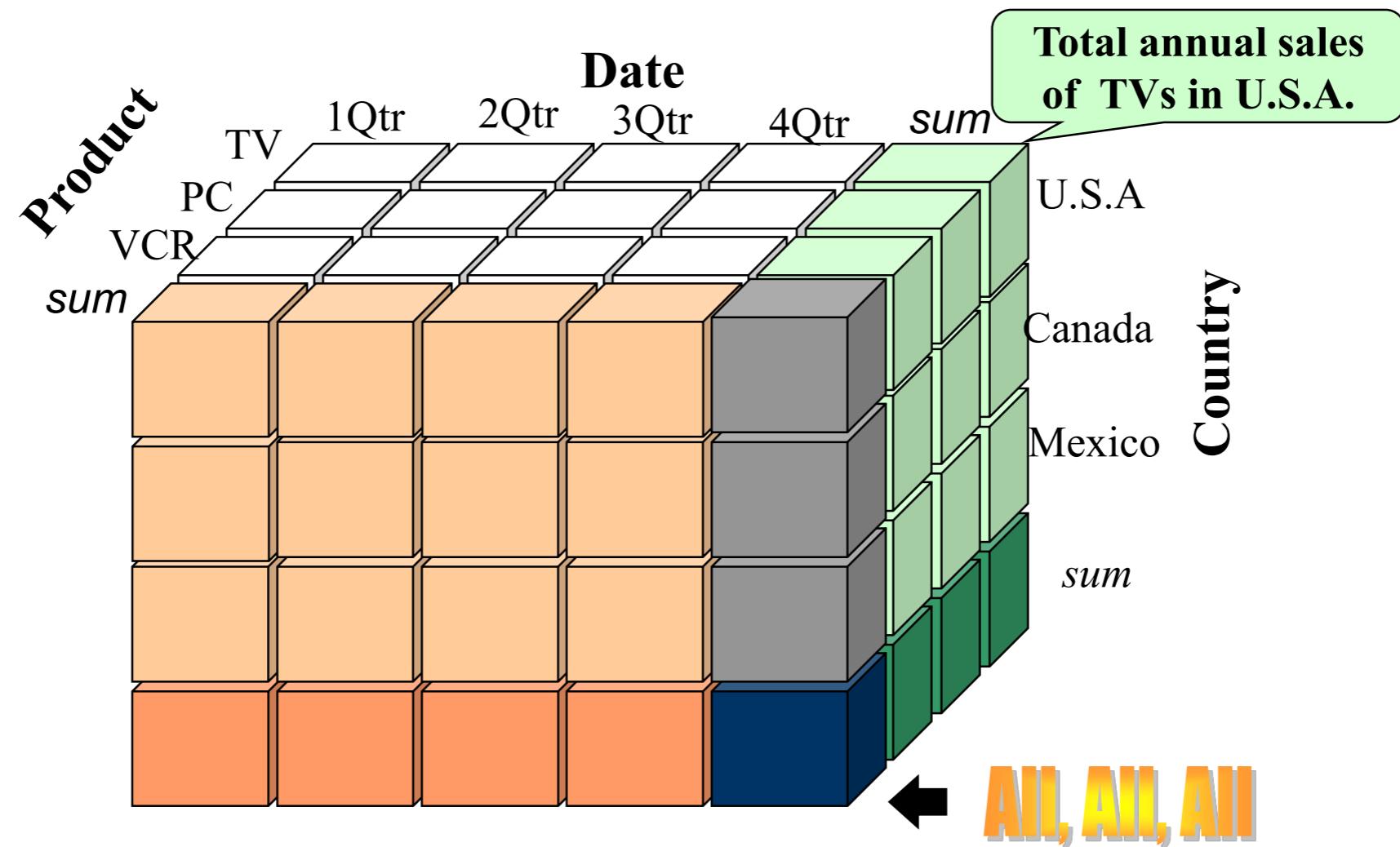
Multidimensional Data

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

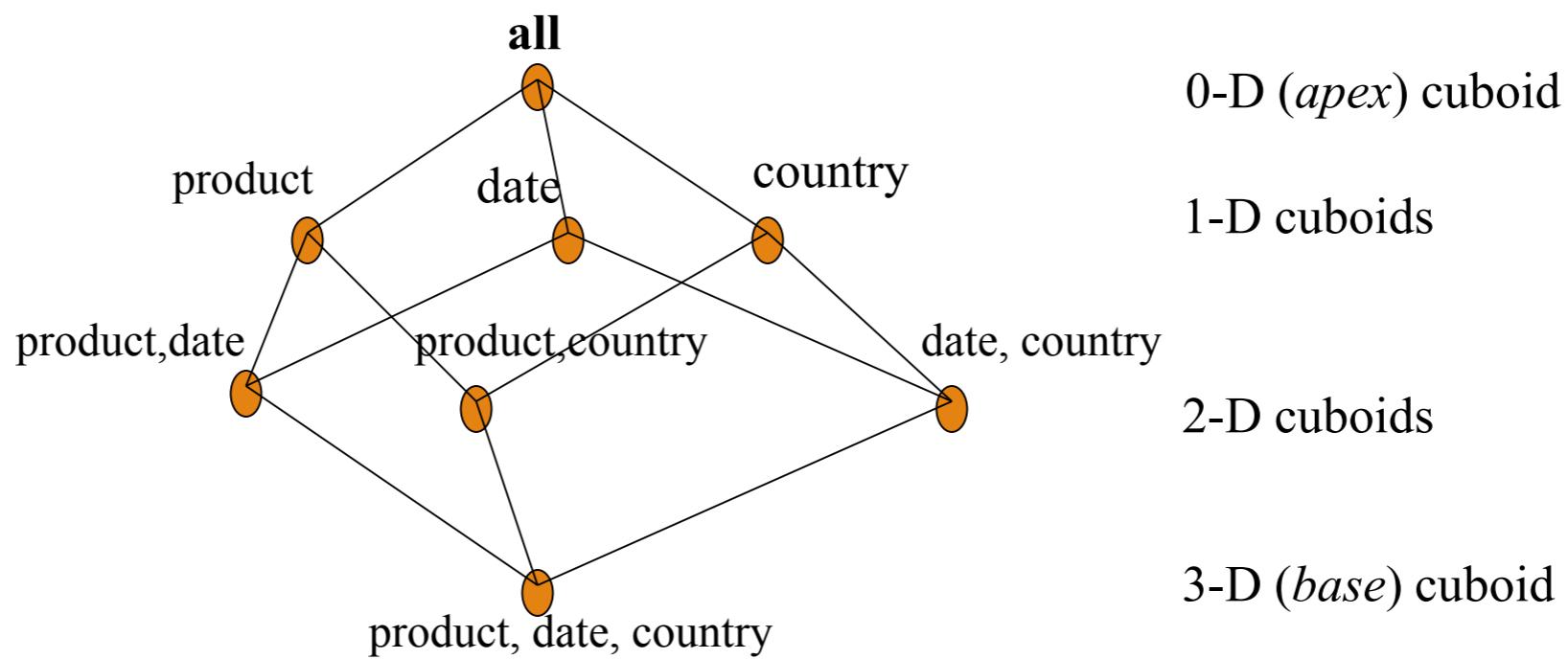
Dimensions: *Product, Location, Time*
Hierarchical summarization paths



A Sample Data Cube



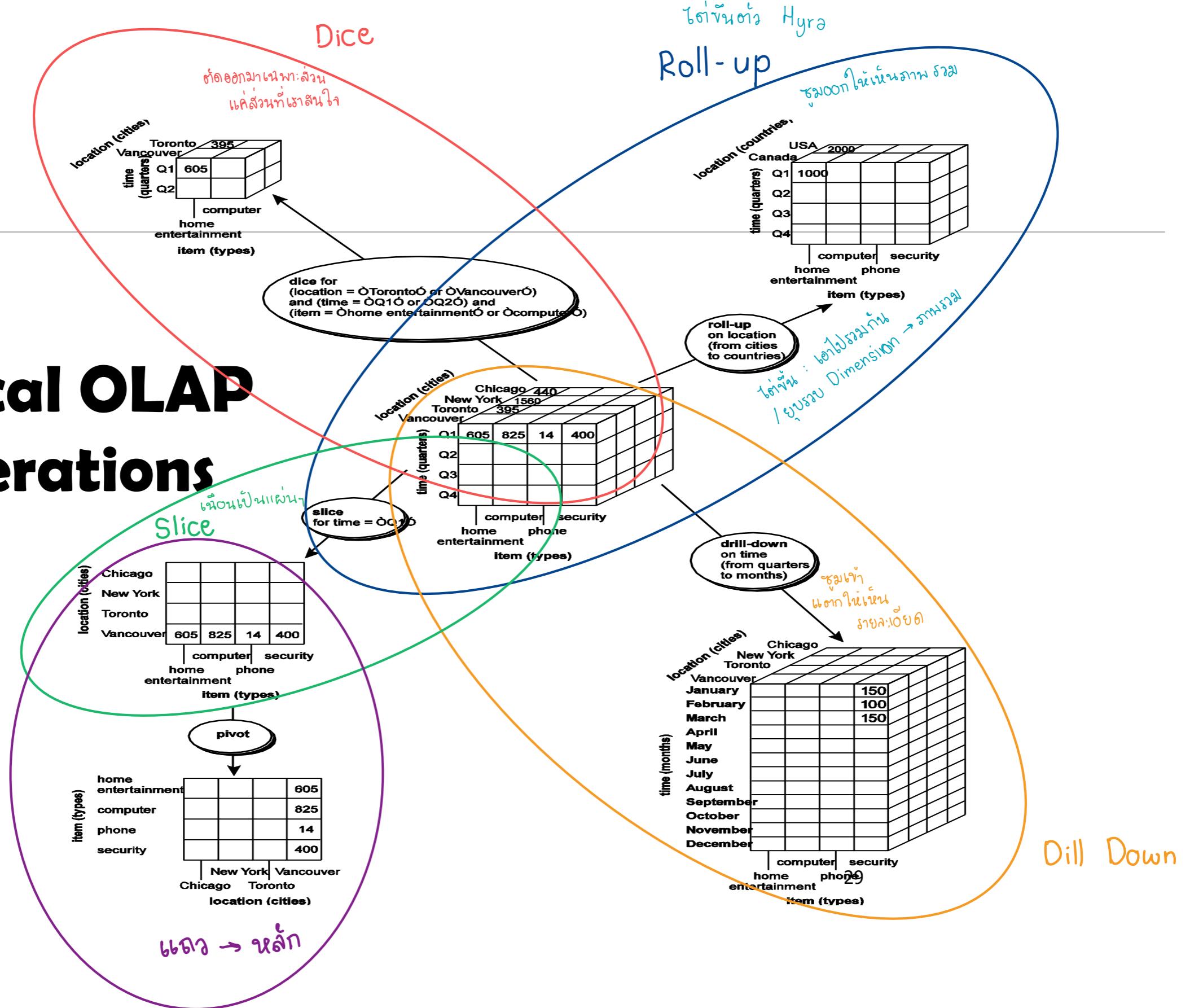
Cuboids Corresponding to the Cube



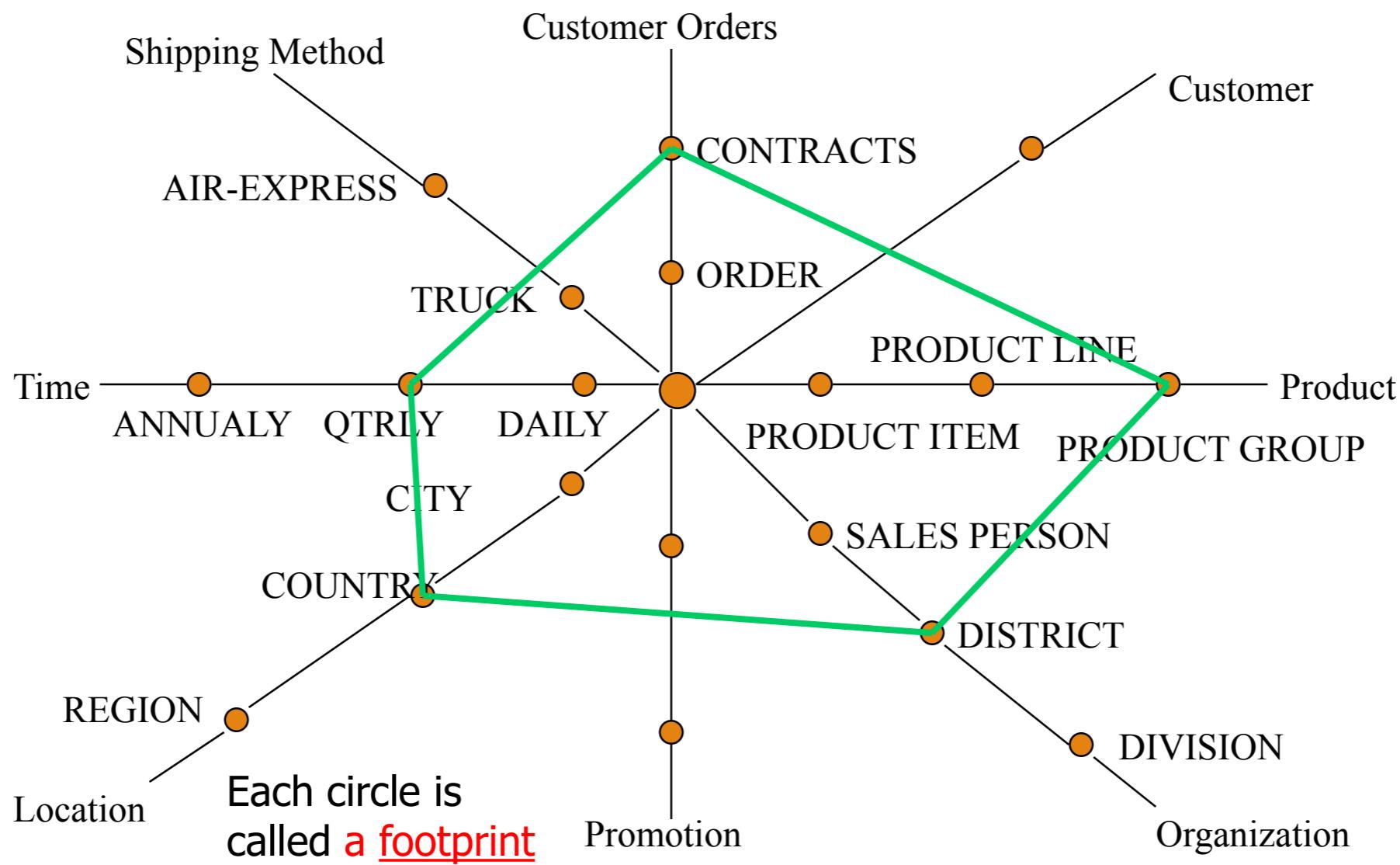
Typical OLAP Operations

- 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, 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)*

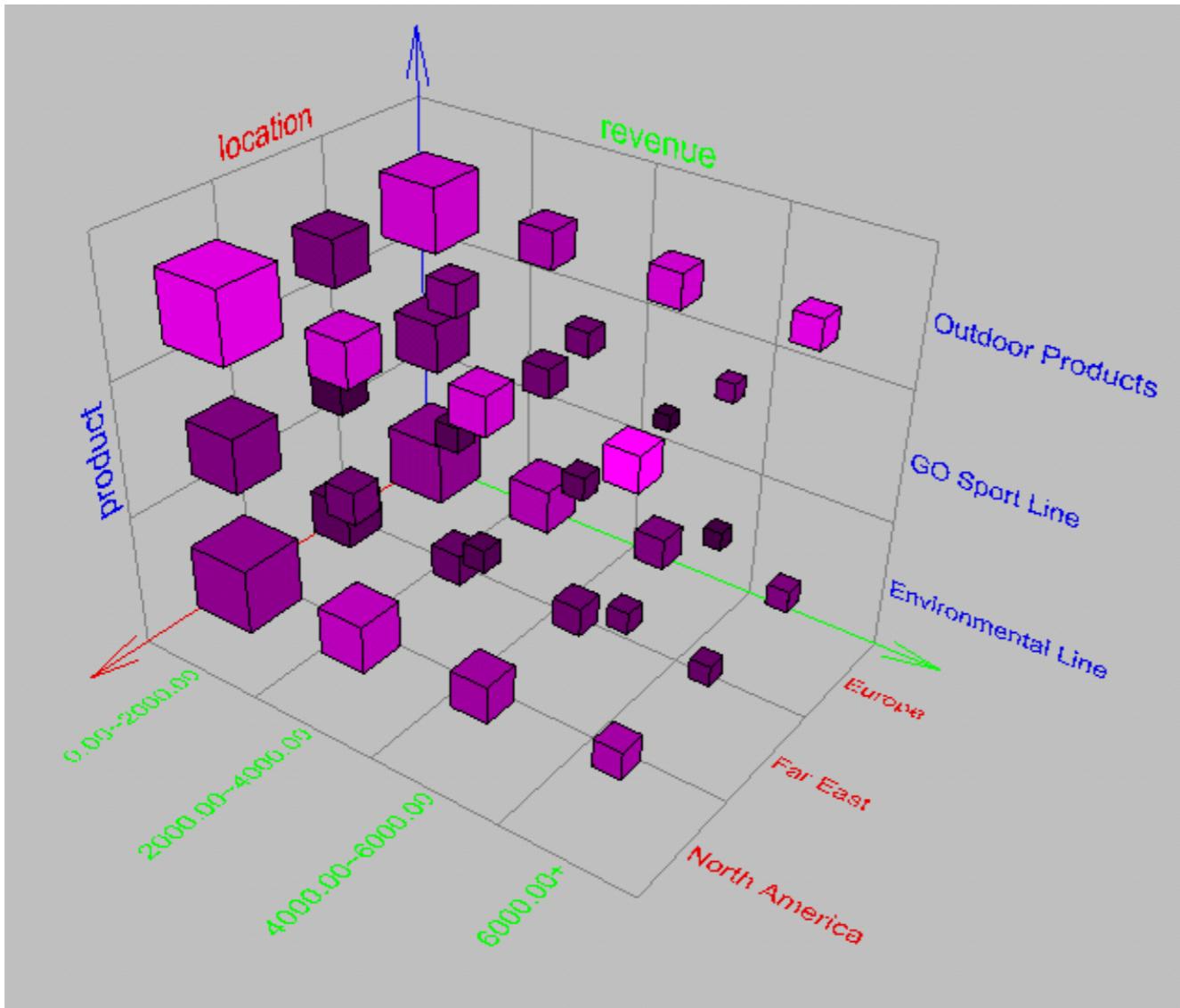
Typical OLAP Operations



A Star-Net Query Model



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 Design and Usage
- Data Warehouse Implementation
- Summary



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 turn around time, quick turn around
- **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
 - ODBC, 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

Chapter 4: Data Warehousing and On-line Analytical Processing

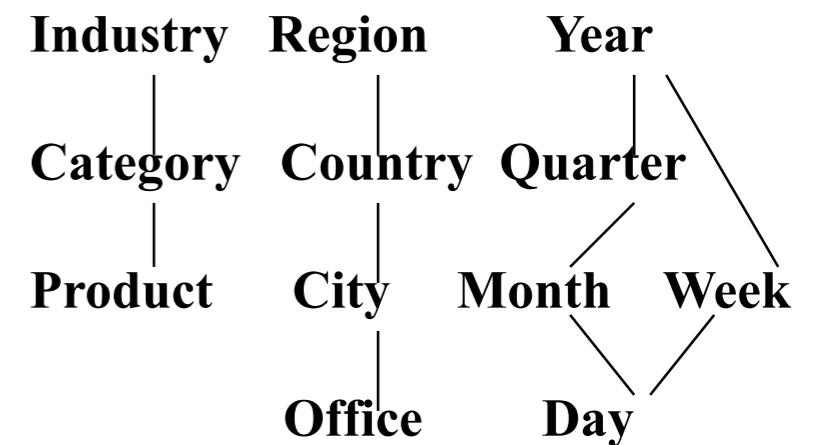
- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- Data Warehouse Implementation 
- Summary

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.

Why this formula?

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



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.'96)

```
SELECT item, city, year, SUM (amount)
```

```
FROM SALES
```

```
CUBE BY item, city, year
```

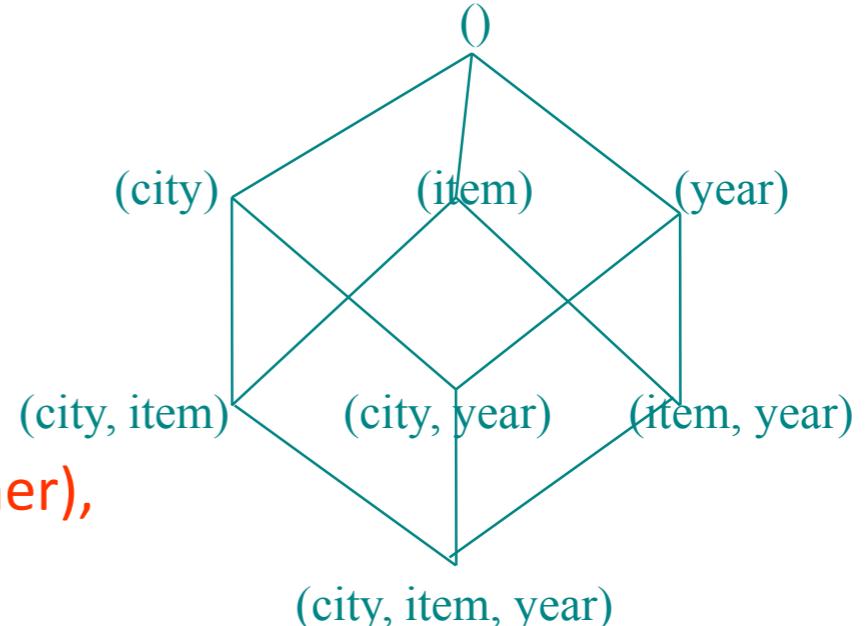
- ❑ Need compute the following Group-Bys

(date, product, customer),

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

(date), (product), (customer)

()



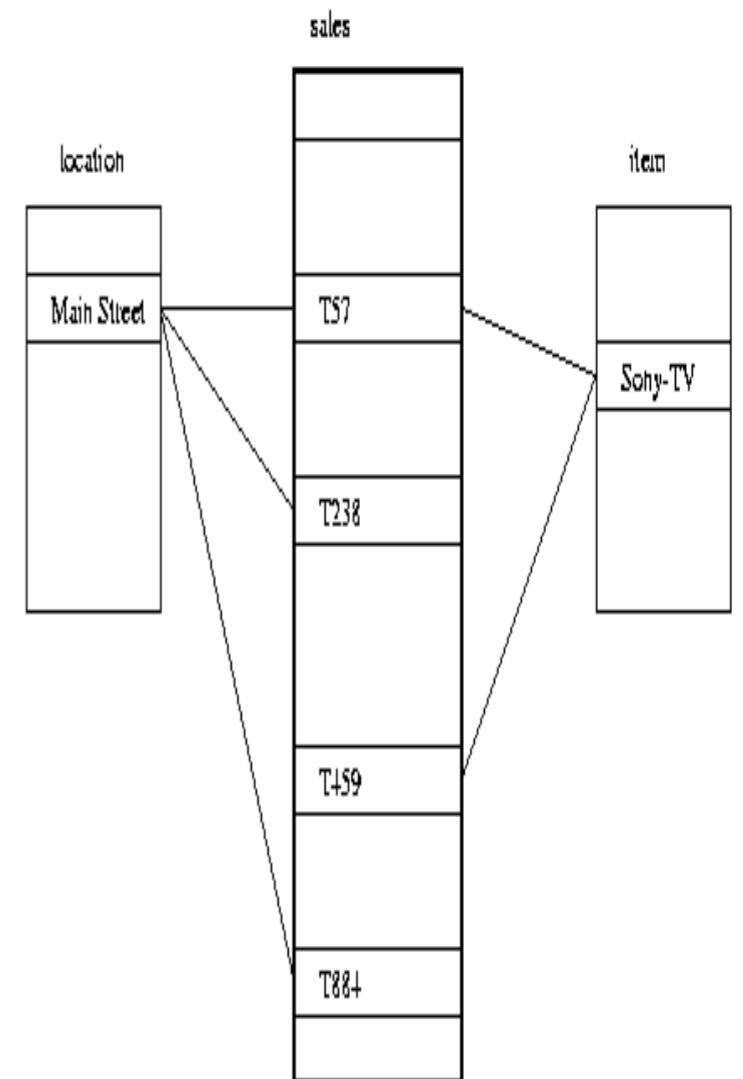
Indexing OLAP Data: Bitmap Index

- 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
- 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			Index on Region				Index on Type		
Cust	Region	Type	RecID	Asia	Europe	America	RecID	Retail	Dealer
C1	Asia	Retail	1	1	0	0	1	1	0
C2	Europe	Dealer	2	0	1	0	2	0	1
C3	Asia	Dealer	3	1	0	0	3	0	1
C4	America	Retail	4	0	0	1	4	1	0
C5	Europe	Dealer	5	0	1	0	5	0	1

Indexing OLAP Data: Join Indices

- Join index: $JI(R\text{-id}, S\text{-id})$ where $R (R\text{-id}, \dots) \bowtie S (S\text{-id}, \dots)$
- Traditional indices map the values to a list of record ids
 - It materializes relational join in JI file and speeds up relational join
- In data warehouses, join index relates the values of the dimensions of a start schema to rows in the fact table.
 - E.g., fact table: *Sales* and two dimensions *city* and *product*
 - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
 - Join indices can span multiple dimensions



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

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

Chapter 4: Data Warehousing and On-line Analytical Processing

- Data Warehouse: Basic Concepts
 - Data Warehouse Modeling: Data Cube and OLAP
 - Data Warehouse Design and Usage
 - 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, OLAM
- Implementation: Efficient computation of data cubes
 - Partial vs. full vs. no materialization
 - Indexing OLAP 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. Galembo, 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.

