



Data mining

MAS-ICT

Professor
Dr. Laura E. RAILEANU

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Organization

- Course content
 - theoretical courses on fundamental topics
 - theoretical and practical assignments (labs)
- Schedule
 - 8 hours (45minutes) * 3 days
- Evaluation
 - assignments
 - report

Course content (first part)

- Data Warehousing
- Data Preprocessing
- Data Mining Techniques
 - Market Basket Analysis
 - Classification
 - Clustering
 - Estimation
 - Prediction
 - Description

References

- « Data Mining: Concepts and Techniques », Jiawei Han and Micheline Kamber, 3rd edition, Morgan Kaufmann, 2011.
- « Handbook of Statistical Analysis and Data Mining Applications », Robert Nisbet, John Elder IV, and Gary Miner, 2009.
- « Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations », Ian H. Witten, Eibe Frank, 1999.
- « Data Mining (Techniques appliquées au marketing, à la vente et aux services clients) », Berry, 1997.
- « Principles of Data Mining », David J. Hand, Heikki Mannila, Padhraic Smyth, 2001.
- « Data Mining », Pieter Adriaans , Dolf Zantige, 1996.
- « Data Mining and Statistical Analysis Using SQL », Robert P. Trueblood, John N. Lovett, Jr, 2001.
- « Applied Data Mining (Statistical Methods for Business and Industry) », Paolo Giudici, 2003.
- « Data Mining (Introductory and Advanced Topics) », Margaret H. Dunham, 2003.
- « Data Mining (A Tutorial-Based Primer) », Richard J. Roiger, Michael W. Geatz, 2003.

Chapter 1: Data Warehouses and OLAP

Motivation

- *Data Warehouses (DW)* generalize and consolidate data in **multidimensional space**
- The construction of DW is an important preprocessing step for data mining involving data cleaning, data integration, data transformation


Motivation (ctd.)

- DW provide *on-line analytical processing (OLAP)* tools for the interactive analysis of multidimensional data of varied granularities
 - facilitates effective data generalization and data mining
- Data mining functions (association, classification, prediction, clustering) can be integrated with OLAP operations
 - to enhance interactive mining of knowledge at multiple levels of abstraction

Motivation (ctd.)

- The DW : platform for data analysis, on-line analytical processing and data mining
- Data warehousing and OLAP form an essential step in the *Knowledge Discovery Process (KDD)*.

Outline

- Data Warehouse: Basic Concepts 
- Data Warehouse Modeling: Data Cube and OLAP
- Data Warehouse Design and Usage
- 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.
- W. H. Inmon: “A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.”
- Data warehousing
 - The process of constructing and using data warehouses

Data Warehouse – Subject Oriented

- Organized around major subjects, such as for e.g.,
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

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 to
 - 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

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

- A **physically separate store** of data transformed from the operational environment
- 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*

Differences between OLTP and DW (OLAP)

- The on-line operational database systems perform on-line transaction and query processing : *on-line transaction processing (OLTP) systems*.
- The OLTP systems cover most of the day-to-day operations of an organization (purchasing, inventory, manufacturing, banking, payroll, registration, and accounting).
- DW systems serve users or knowledge workers in the role of data analysis and decision making.
- They can organize and present data in various formats in order to accommodate the diverse needs of the different users. These systems are known as *on-line analytical processing (OLAP) systems*.

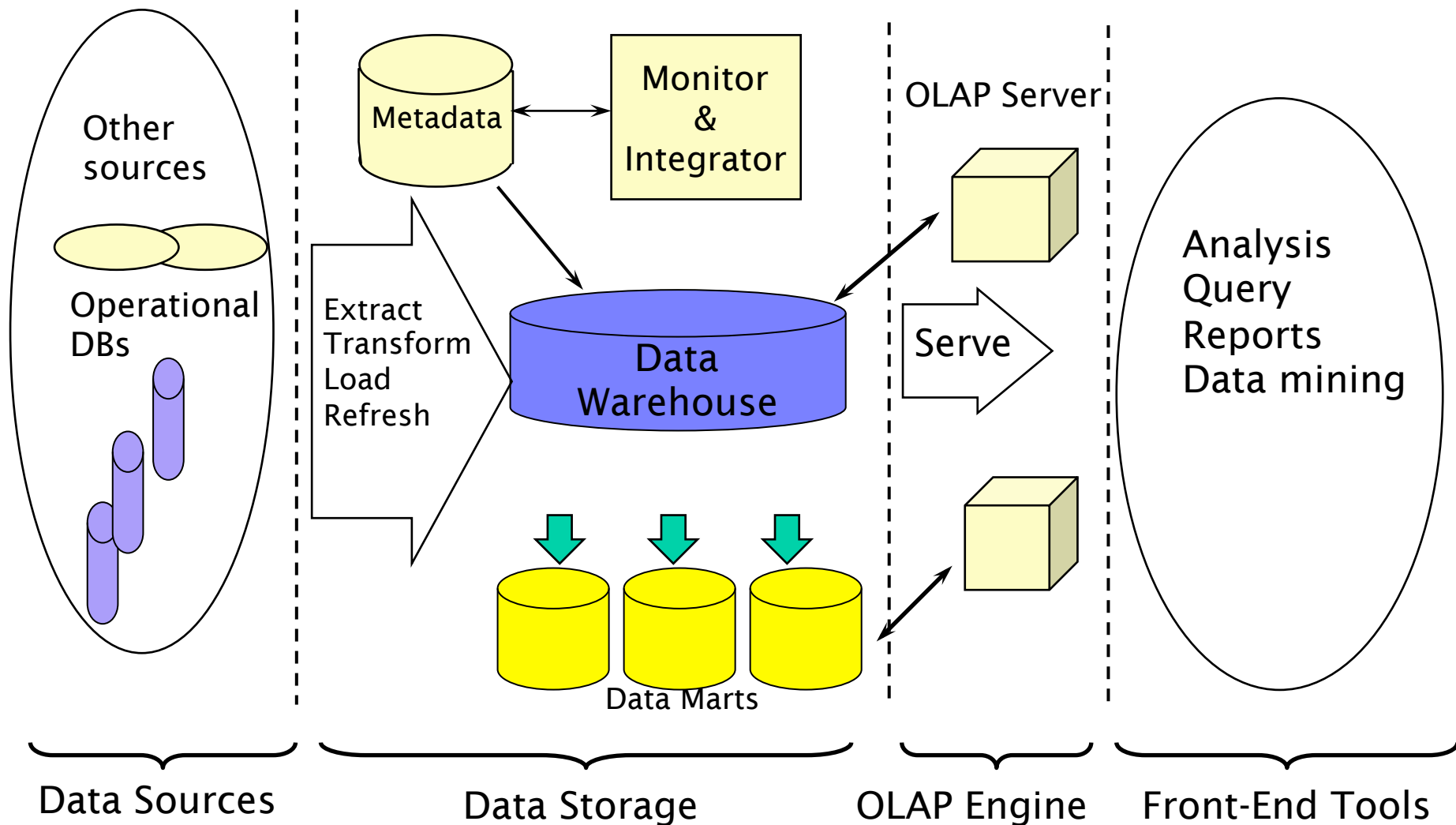
OLTP vs. OLAP

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

Why a Separate Data Warehouse?

- High performance for both systems
 - DBMS - tuned for OLTP: access methods, indexing, concurrency control, recovery
 - Warehouse - tuned for OLAP: complex OLAP queries, multidimensional view, consolidation
- Different functions and different data:
 - 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 cArchitecture



Data Warehouse: A Multi-Tiered Architecture (ctd.)

- The bottom tier: **a warehouse database server** (almost always a relational database system)
 - feeded with data from operational databases or external sources by **back-end tools and utilities** (data extraction, cleaning, transformation, load and refresh functions)
 - the data are extracted using application program interfaces known as **gateways**. A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server.
 - ODBC (Open Database Connection), OLEDB (Object Linking and Embedding Database) by Microsoft and JDBC (Java Database Connection).
 - it also contains a **metadata repository**, which stores information about the data warehouse and its contents.

Data Warehouse: A Multi-Tiered Architecture (ctd.)

- The middle tier: **an OLAP server** that is typically implemented using either
 - a relational OLAP (ROLAP) model: an extended relational DBMS that maps operations on multidimensional data to standard relational operations;
 - a multidimensional OLAP (MOLAP) model: a special-purpose server that directly implements multidimensional data and operations.
- The top tier: **a front-end client layer**
 - query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction)

Three Data Warehouse Models

- Enterprise warehouse

- collects all of the information about subjects spanning the entire organization
- provides corporate-wide data integration
- contains detailed and summarized data

- 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
- data contained inside tend to be summarized
 - Independent vs. dependent (directly from warehouse) data mart

Three Data Warehouse Models (ctd.)

- **Virtual warehouse**

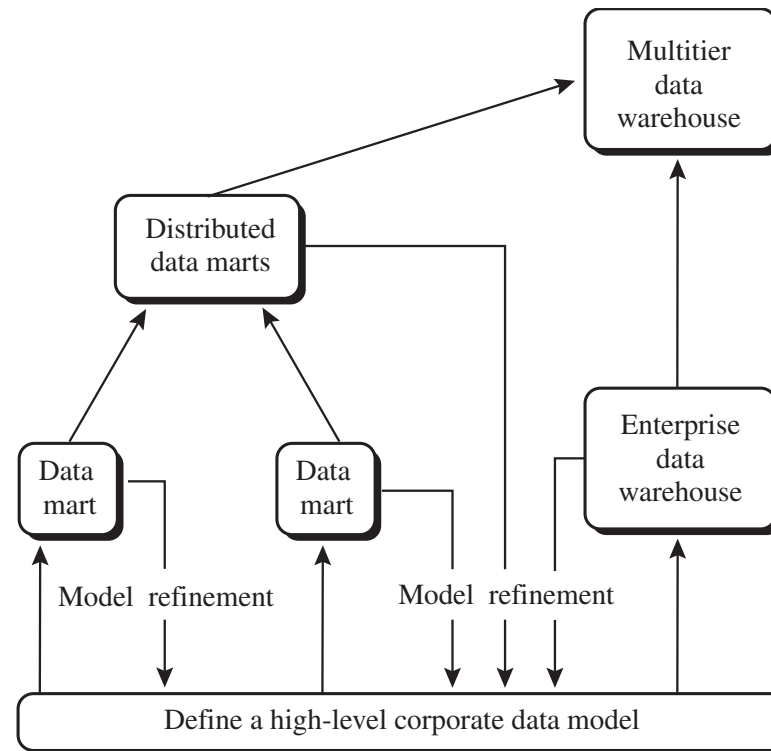
- a set of views over operational databases
- only some of the possible summary views may be materialized
- easy to build but requires excess capacity on operational database server

Data warehouse development approaches

- *The top-down* development:
 - a systematic solution, minimizes integration problems
 - expensive, takes a long time to develop, and lacks flexibility due to the difficulty in achieving consistency and consensus for a common data model for the entire organization
- *The bottom-up* approach to the design, development, and deployment of independent data marts:
 - provides flexibility, low cost, and rapid return of investment
 - problems when integrating various disparate data marts into a consistent enterprise data warehouse

Data warehouse development approaches (ctd.)

- A recommended method for the development of data warehouse systems is to implement the warehouse in an incremental and evolutionary manner



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)

Metadata Repository (ctd.)

- It stores:
 - 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

Data Warehousing and On-line Analytical Processing

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

A 2-D cube representation (time, item)

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”

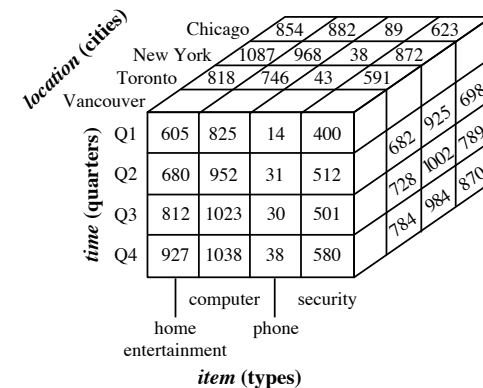
<i>time</i> (quarter)	<i>item</i> (type)			
	<i>home</i> <i>entertainment</i>	<i>computer</i>	<i>phone</i>	<i>security</i>
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”, 2011

A 3-D data cube representation (temps, article, adresse)

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>	<i>ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>	<i>ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>	<i>ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>	<i>ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>
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



Jiawei Han and
Micheline Kamber,

“Data Mining: Concepts
and Techniques”, 2011

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

A 4-D data cube representation (time, item, location, supplier)

- We can imagine a 4-D cube as being a series of 3-D cubes

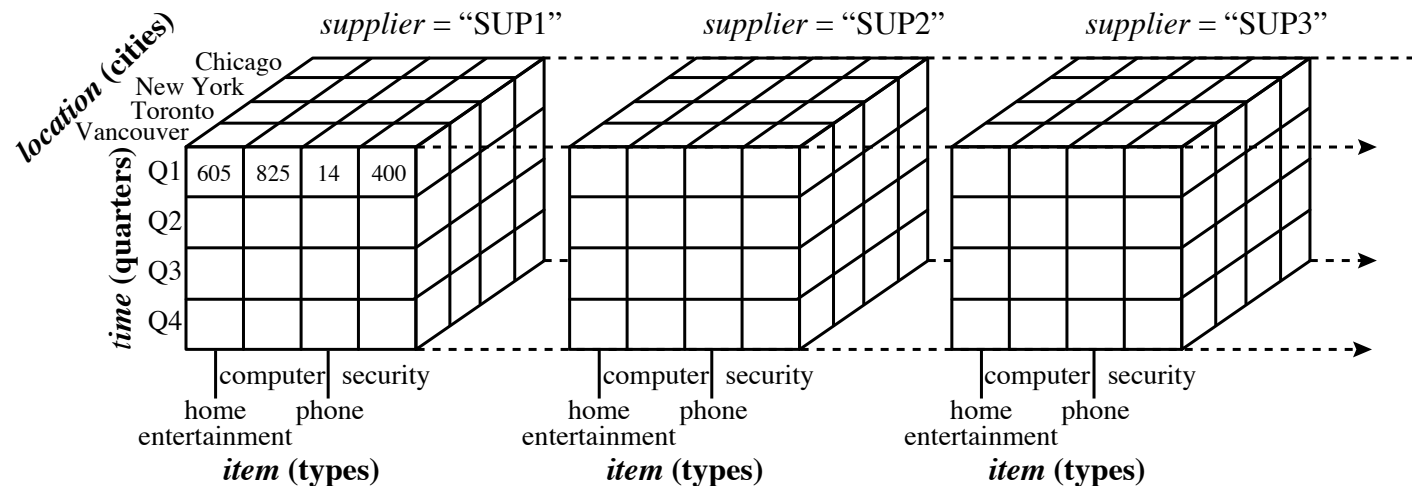
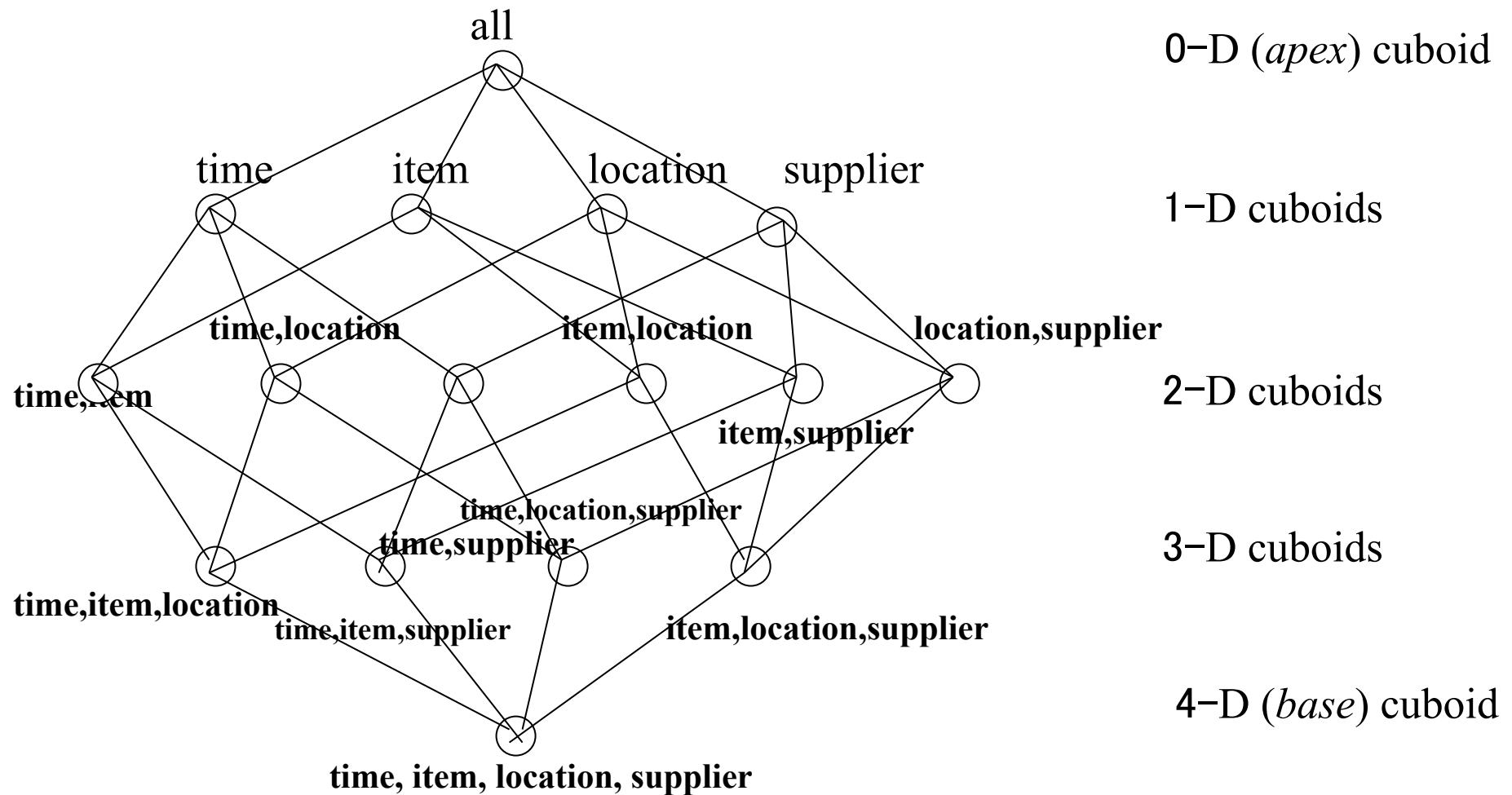


Figure 4.4: A 4-D data cube representation of sales data, according to the dimensions *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars_sold* (in thousands). For improved readability, only some of the cube values are shown.

Data Cubes

- 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**.

Cube: A Lattice of Cuboids



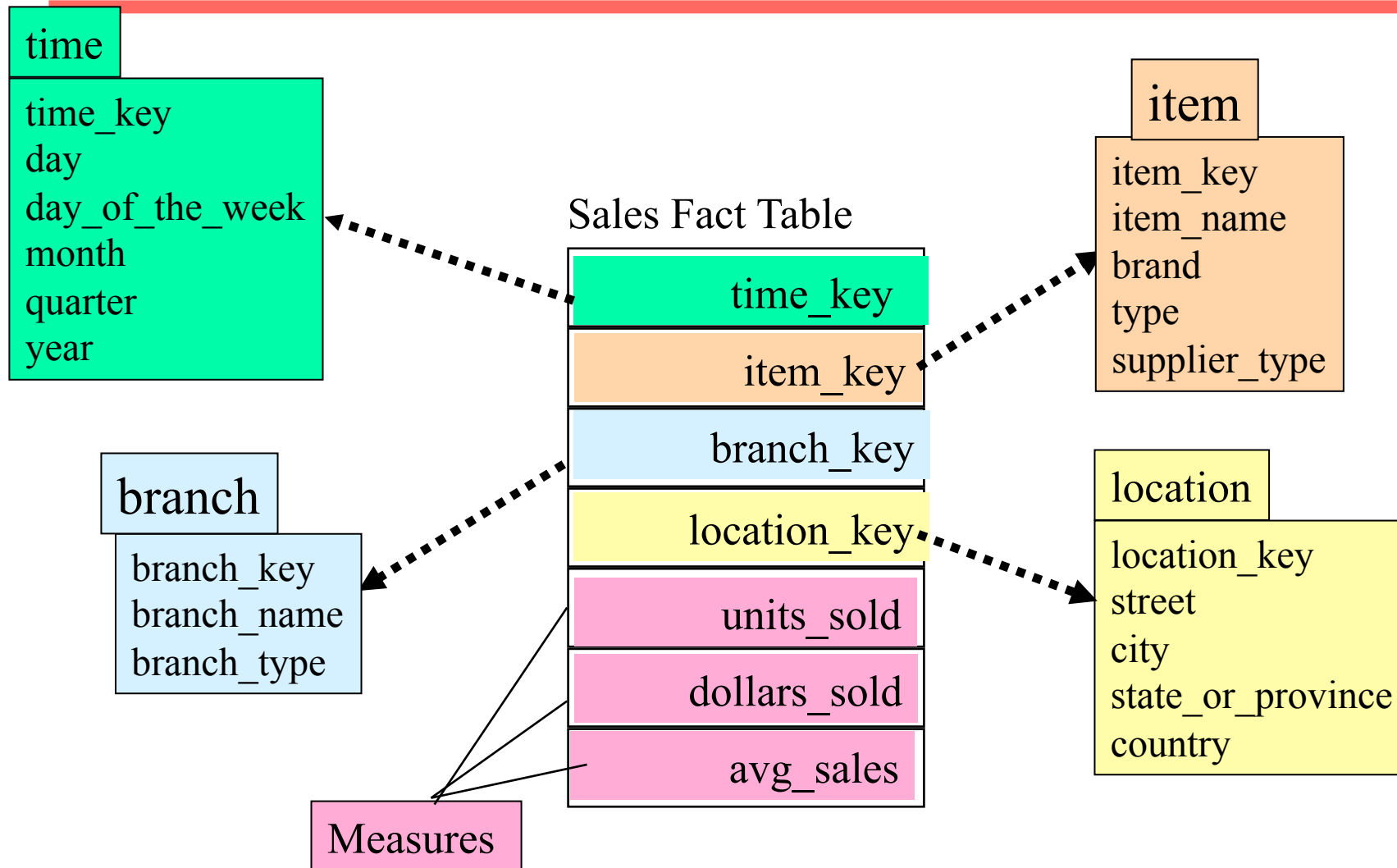
Conceptual Modeling of Data Warehouses

- 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

Star schema

- The most common modeling paradigm is the star schema, the data warehouse contains
 - a large central table (**fact table**) containing the bulk of the data, with no redundancy
 - a set of smaller attendant tables (**dimension tables**), one for each dimension
- The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

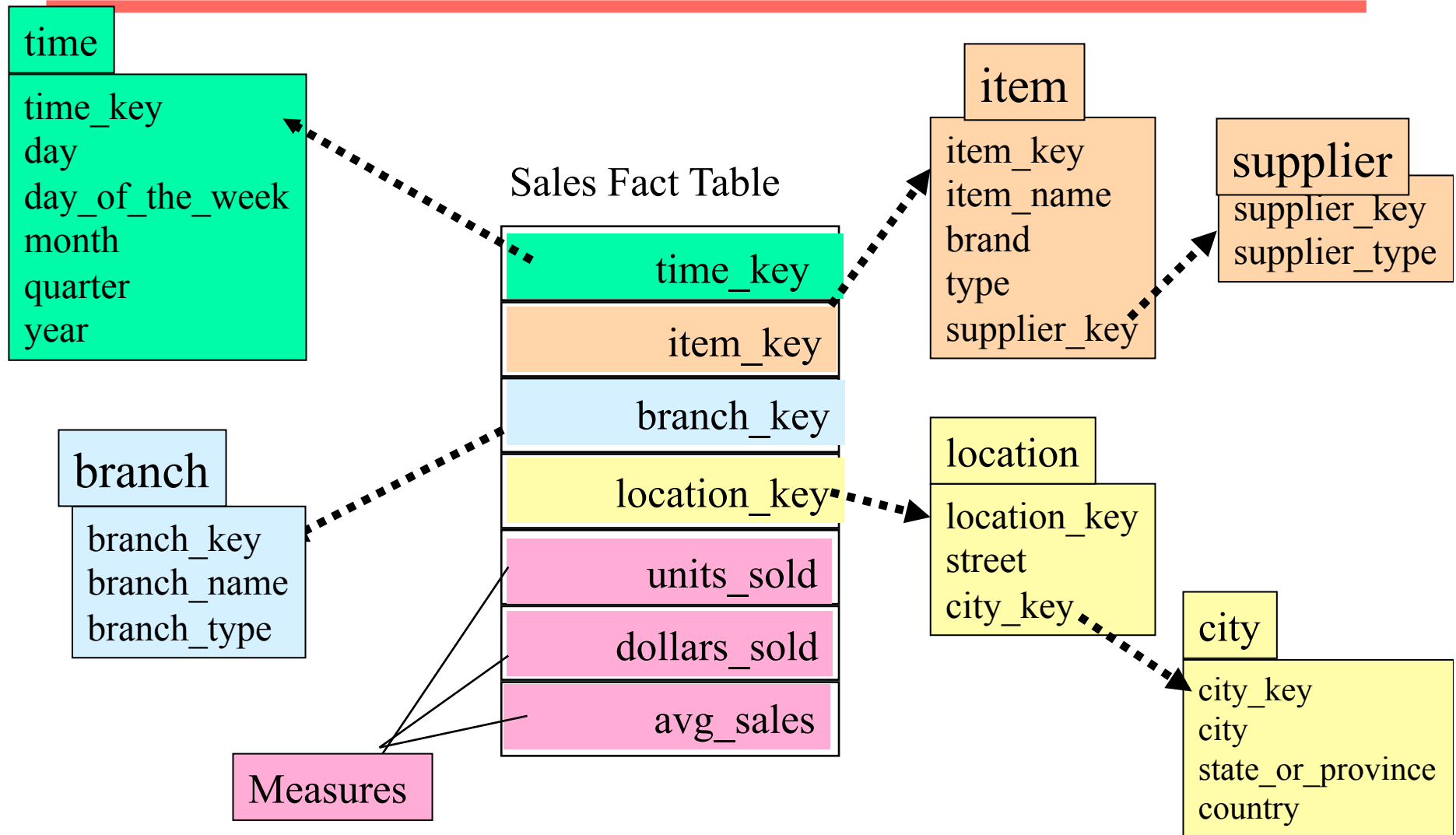
Example of Star Schema



Snowflake schema

- A variant of the star schema model, where **some dimension tables are *normalized***, thereby further splitting the data into additional tables
- The resulting schema graph forms a shape similar to a snowflake.
- The dimension tables of the snowflake model may be kept in normalized form to reduce redundancies.
- But, the snowflake structure can reduce the effectiveness of browsing, since more joins will be needed to execute a query.
- Although the snowflake schema reduces redundancy, it is not as popular as the star schema in data warehouse design.

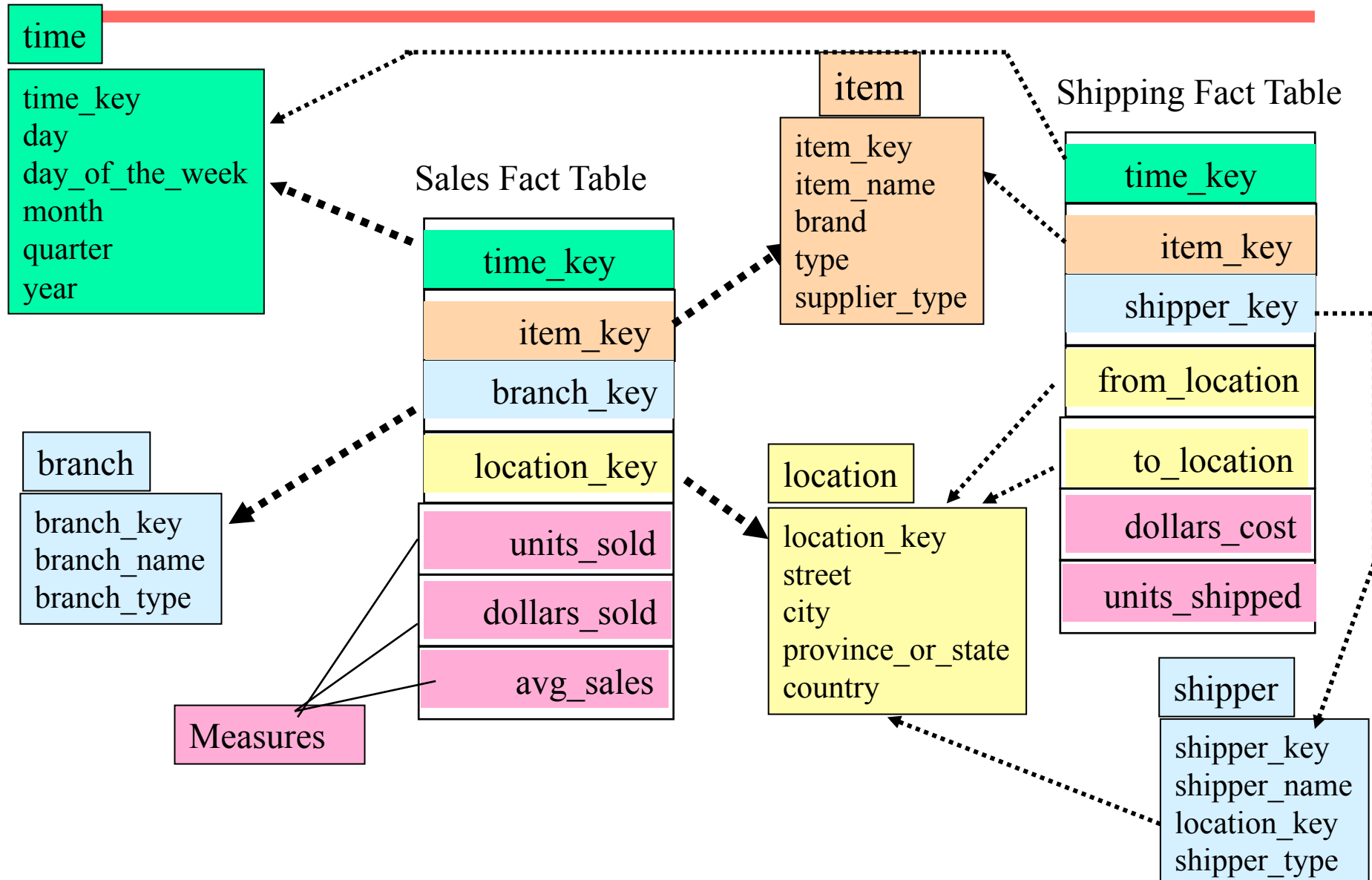
Example of Snowflake Schema



Fact constellation schema

- Sophisticated applications may require **multiple fact tables** to *share dimension tables*.
- This kind of schema can be viewed as a collection of stars, and hence is called **a galaxy schema or a fact constellation**.

Example of Fact Constellation



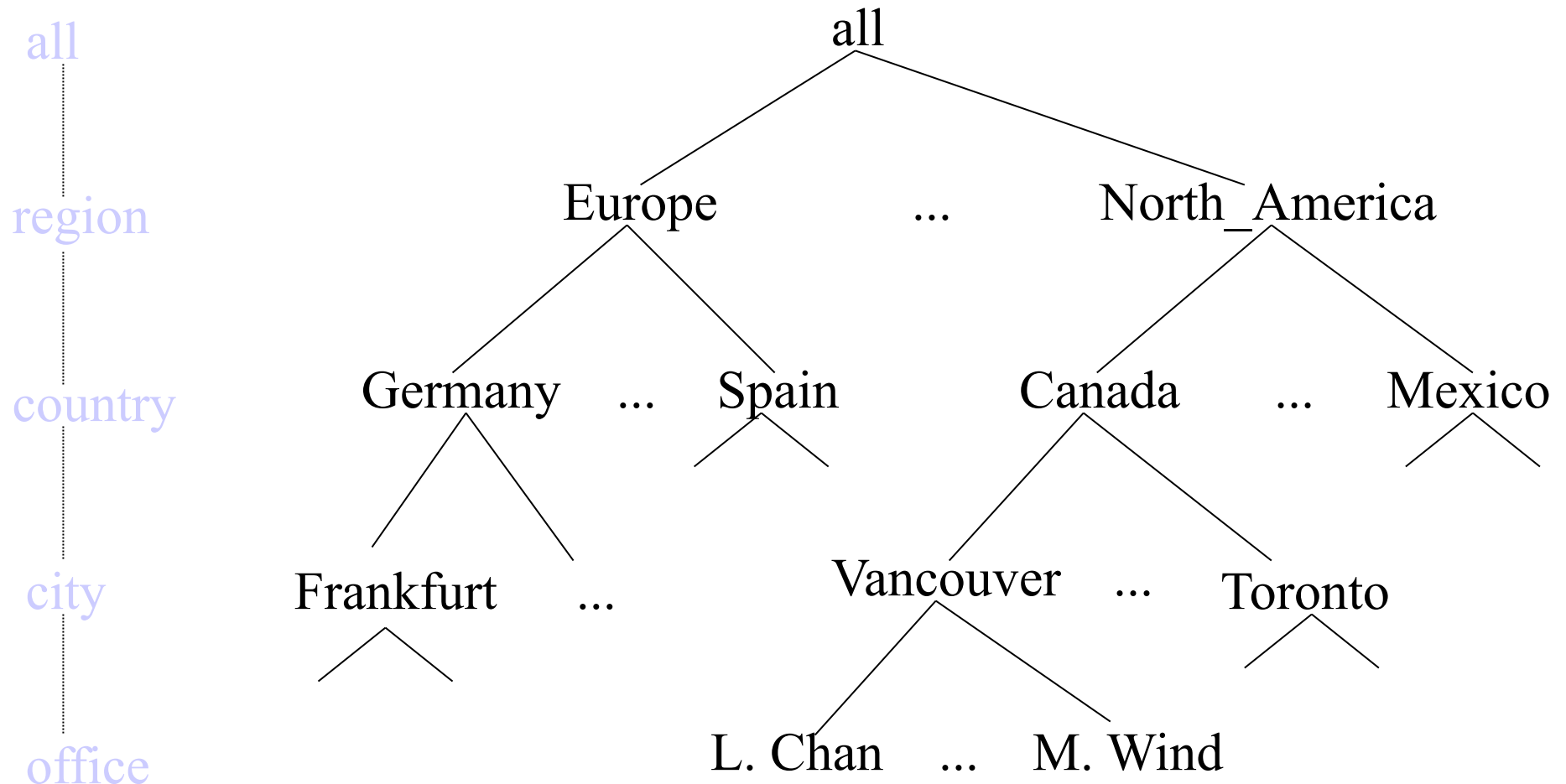
Use of the schemas

- A data warehouse collects information about subjects that span the *entire organization*, such as *customers, items, sales, assets*, and *personnel*, and thus its scope is *enterprise-wide*.
 - For data warehouses, the fact constellation schema is commonly used, since it can model multiple, interrelated subjects.
- A data mart, on the other hand, is a department subset of the data warehouse that focuses on selected subjects, and thus its scope is *department-wide*.
 - For data marts, the *star* or *snowflake* schema are commonly used, since both are geared toward modeling single subjects, although the star schema is more popular and efficient.

A Concept Hierarchy

- A *concept hierarchy* defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.
 - Exemple: location
 - City values for *location* : Vancouver, Toronto, NewYork, and Chicago
 - Each city, can be mapped to the province or state to which it belongs: Vancouver can be mapped to British Columbia, and Chicago to Illinois
 - The provinces and states can in turn be mapped to the country to which they belong, such as Canada or the USA
 - These mappings form a concept hierarchy for the dimension *location*, mapping a set of low-level concepts (i.e., cities) to higher-level,more general concepts (i.e., countries).

A Concept Hierarchy: Dimension (location)



Data Cube Measures

- A *data cube measure* is a **numerical function** that can be evaluated at each point in the data cube space.
 - *E.g., a multidimensional point in the data cube space :time = "Q1", location = "Vancouver", item = "computer"*
- A measure value is computed for a given point by aggregating the data corresponding to the respective dimension-value pairs defining the given point.

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
 - Distributive measures can be computed efficiently because of the way the computation can be partitioned
 - 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
 - E.g., `avg()`, `min_N()`, `standard_deviation()`
- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
 - It is difficult to compute holistic measures efficiently. Efficient techniques to *approximate*. The computation of some holistic measures, however, do exist.
 - E.g., `median()`, `mode()`, `rank()`

Interpreting measures for data cubes

- Measures can be computed by relational aggregation operations

- E.g., the relational database schema of AllElectronics:

```
time(time key, day, day of week, month, quarter, year)
item(item key, item name, brand, type, supplier type)
branch(branch key, branch name, branch type)
location(location key, street, city, province or state,
          country)
sales(time key, item key, branch key, location key, number of
       units sold, price)
```

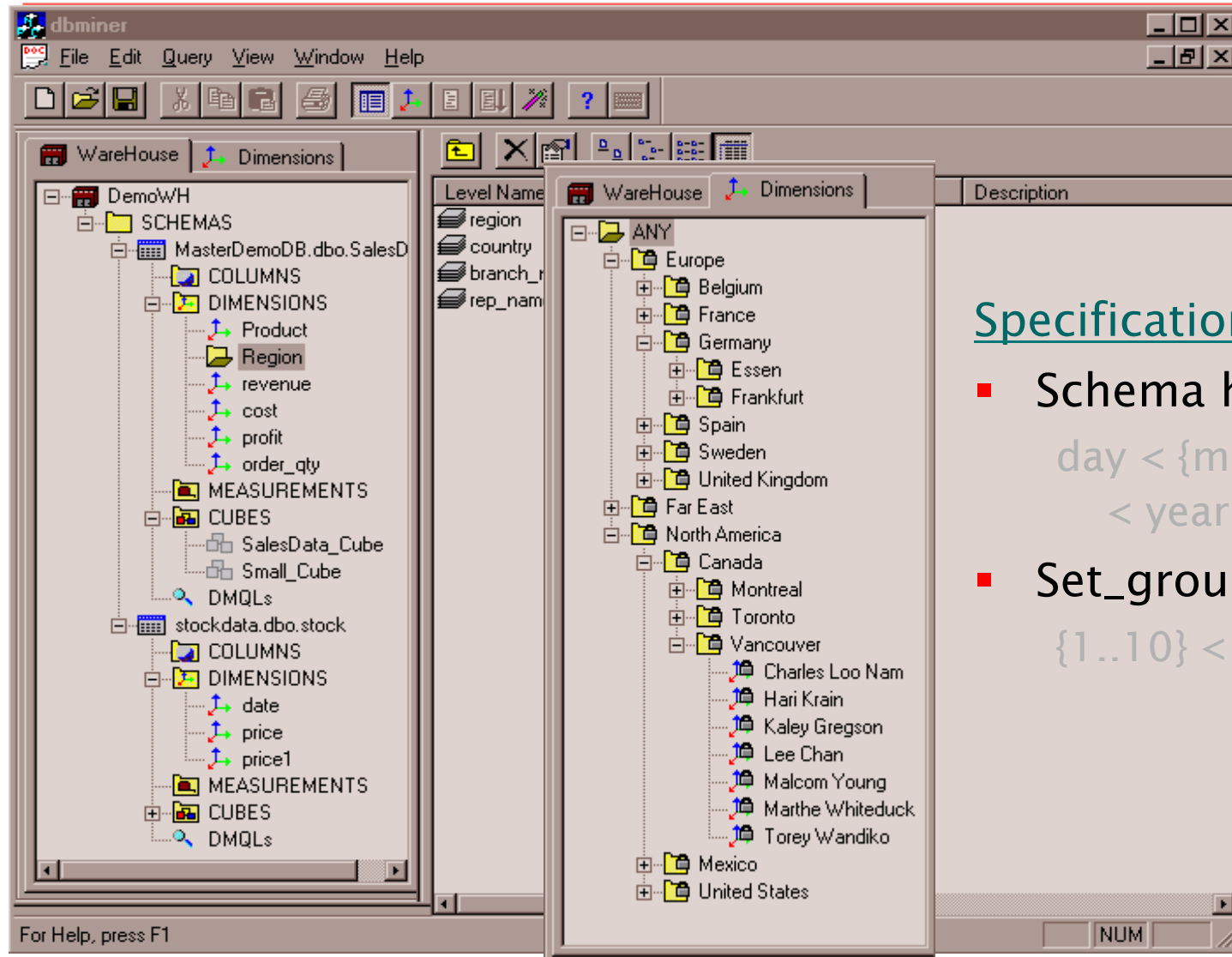
Interpreting measures for data cubes (ctd.)

- The base cuboid of the sales star data cube is obtained using the SQL query:

```
select s.time key, s.item key, s.branch key,  
       s.location key,  
       sum(s.number of units sold * s.price), sum(s.number of  
       units sold)  
from time t, item i, branch b, location l, sales s,  
where s.time key = t.time key and s.item key = i.item  
       key  
and s.branch key = b.branch key and s.location key =  
       l.location key  
group by s.time key, s.item key, s.branch key,  
       s.location key
```

- The sum aggregate function, is used to compute both dollars sold and units sold.
- The cube contains all of the dimensions specified in the data cube definition, where the granularity of each dimension is at the join key level.
- A join key is a key that links a fact table and a dimension table.
- The fact table associated with a base cuboid is sometimes referred to as the base fact table.

View of Warehouses and Hierarchies



Specification of hierarchies

- Schema hierarchy

day < {month < quarter; w
< year

- Set_grouping hierarchy

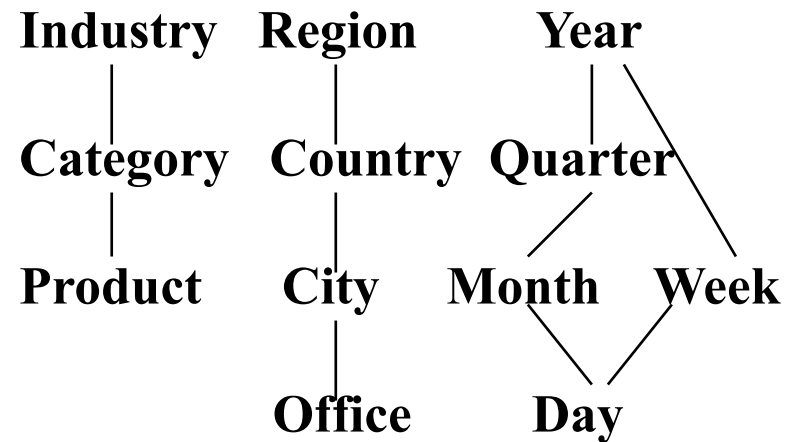
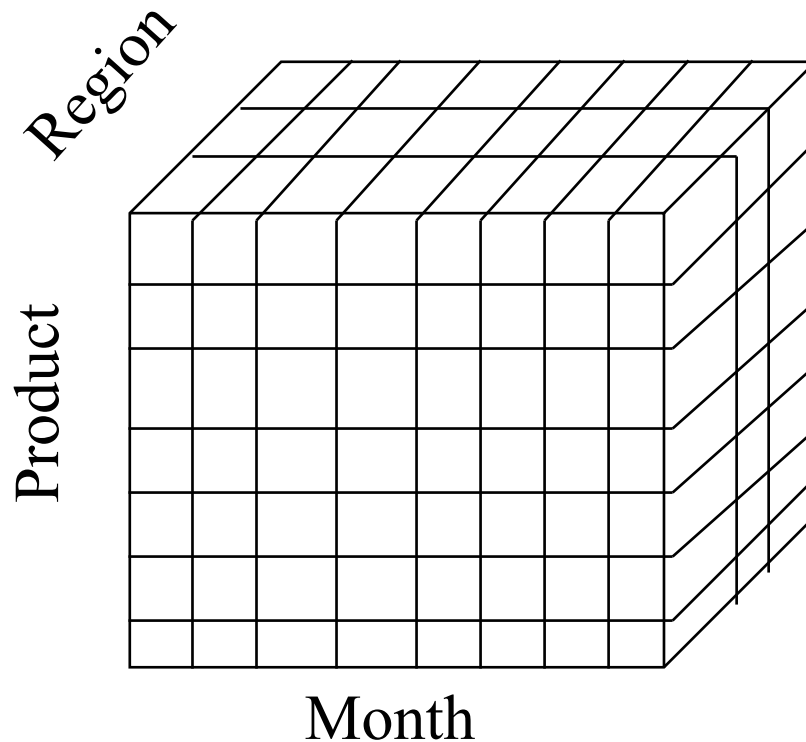
{1..10} < inexpensive

Multidimensional Data

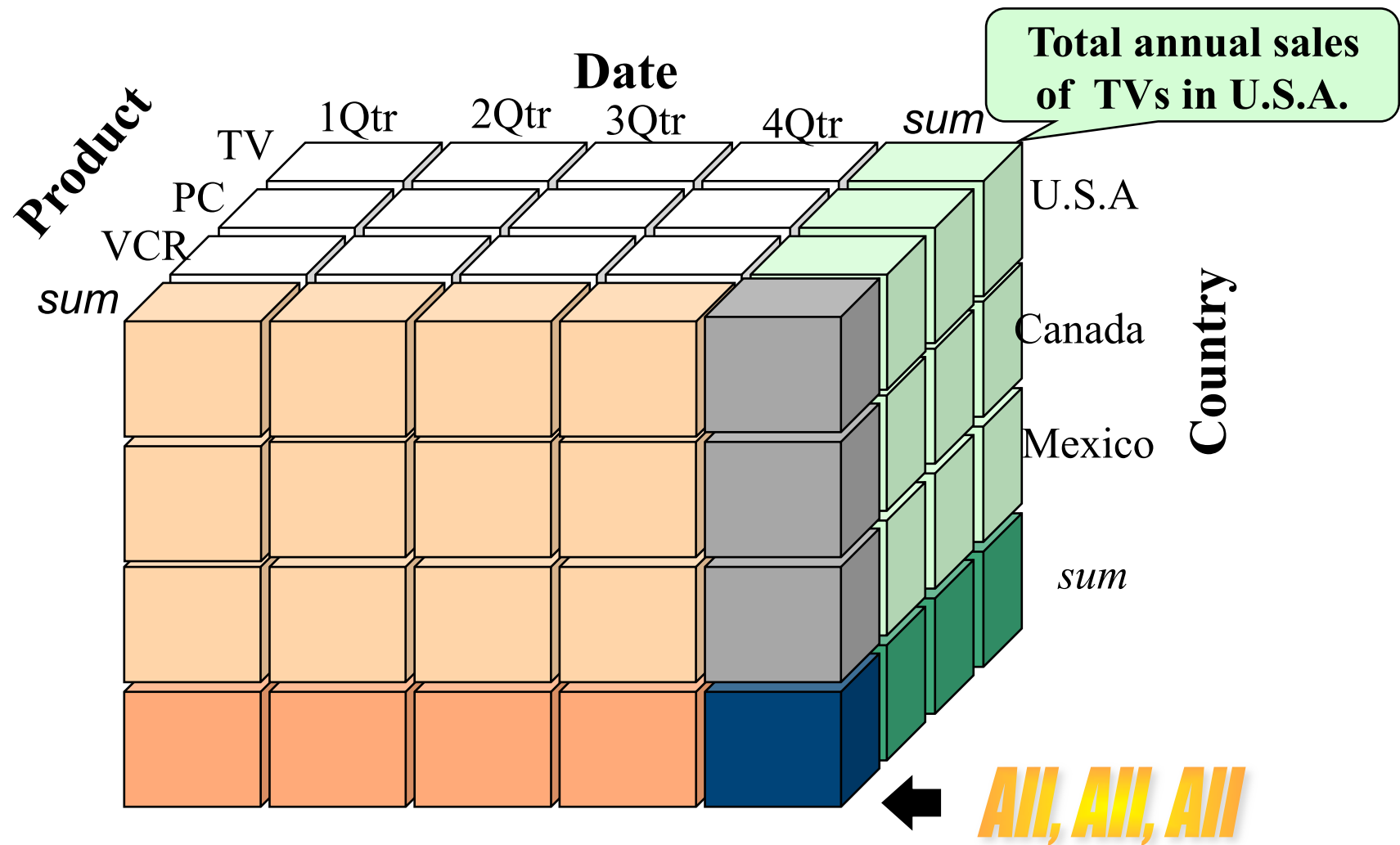
- Sales volume as a function of product, month, and region

Dimensions: *Product, Location, Time*

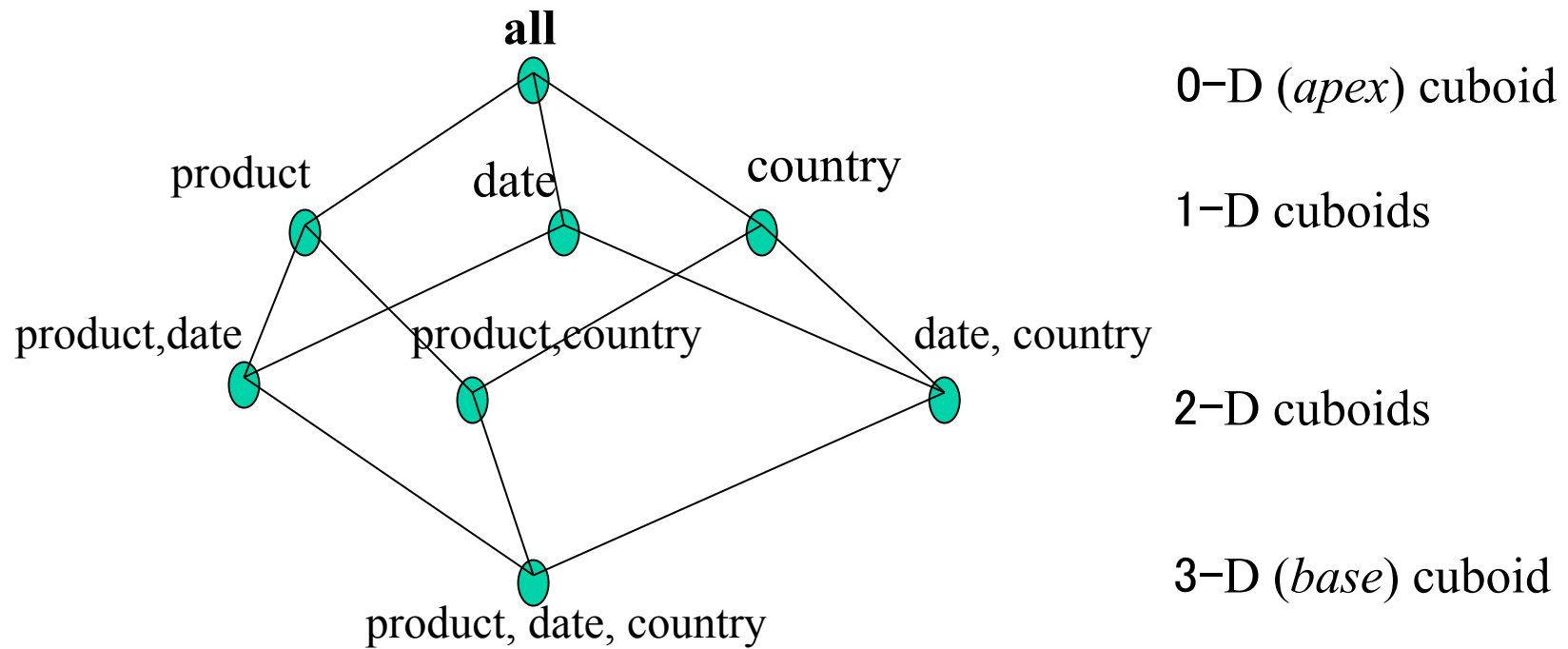
Hierarchical summarization paths



A Sample Data Cube



Cuboids Corresponding to the Cube



Cube Definition Syntax in DMQL

- Can use OLE DB for DM and other languages used in Oracle. Here we use DMQL:
- Cube Definition (Fact Table)
`define cube <cube_name> [<dimension_list>]:
 <measure_list>`
- Dimension Definition (Dimension Table)
`define dimension <dimension_name> as
 (<attribute_or_subdimension_list>)`
- Special Case (Shared Dimension Tables)
 - First time as “cube definition”
 - `define dimension <dimension_name> as
 <dimension_name_first_time> in cube
 <cube_name_first_time>`

Example of a star schema definition in DMQL

```
define cube sales star [time, item, branch,  
    location]: dollars sold = sum(sales in  
    dollars), units sold = count(*)  
define dimension time as (time key, day, day of  
    week, month, quarter, year)  
define dimension item as (item key, item name,  
    brand, type, supplier type)  
define dimension branch as (branch key, branch  
    name, branch type)  
define dimension location as (location key,  
    street, city, province or state, country)
```

Example of a snowflake schema definition in DMQL

```
define cube sales snowflake [time, item, branch, location]:  
dollars sold = sum(sales in dollars), units sold = count(*)  
define dimension time as (time key, day, day of week, month,  
    quarter, year)  
define dimension item as (item key, item name, brand, type,  
    supplier  
    (supplier key, supplier type))  
define dimension branch as (branch key, branch name, branch  
    type)  
define dimension location as (location key, street, city  
    (city key, city, province or state, country))
```

- The *item* and *location* dimension tables are normalized
- Defining *supplier* in this way implicitly creates a *supplier key* in the *item* dimension table definition
- A *city key* is implicitly created in the *location* dimension table definition

Example of a fact constellation schema definition in DMQL

```
define cube sales [time, item, branch, location]:
dollars sold = sum(sales in dollars), units sold = count(*)
define dimension time as (time key, day, day of week, month,
    quarter, year)
define dimension item as (item key, item name, brand, type,
    supplier type)
define dimension branch as (branch key, branch name, branch
    type)
define dimension location as (location key, street, city,
    province or state,
    country)
define cube shipping [time, item, shipper, from location, to
    location]:
dollars cost = sum(cost in dollars), units shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper key, shipper name,
    location as location in cube sales, shipper type)
define dimension from location as location in cube sales
define dimension to location as location in cube sales
```


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)

Roll-up (Drill-up)

- **Example 1**

- A roll-up operation performed on the central cube by climbing up the concept hierarchy for *location* from the level of *city* to the level of *country*
 - Hierarchy definition (total order) “*street < city < province or state < country.*”

The roll-up operation aggregates the data by ascending the *location* hierarchy, the resulting cube groups the data by country.

Roll-up (Drill-up)

- **Example 2:**
 - A roll-up performed by dimension reduction, is considered on a sales data cube (with two dimensions *location* and *time*) by removing, the *time* dimension, resulting in an aggregation of the total sales by location, rather than by location and by time.

Drill-down

- **Example 3**

- A drill-down operation performed on the central cube by stepping down a concept hierarchy for *time* defined as “*day < month < quarter < year.*” , by descending the *time* hierarchy from the level of *quarter* to the more detailed level of *month*.

The resulting data cube details the total sales per month rather than summarizing them by quarter.

Drill-down

- **Example 4**

- A drill-down performed by adding new dimensions to a cube is done on the central cube by introducing an additional dimension, such as *customer group*.

Slice and Dice

- **Example 5**

- The *slice* operation performs a selection on one dimension of the given cube, resulting in a subcube.
 - The sales data are selected from the central cube for the dimension *time* using the criterion *time* = “Q1”

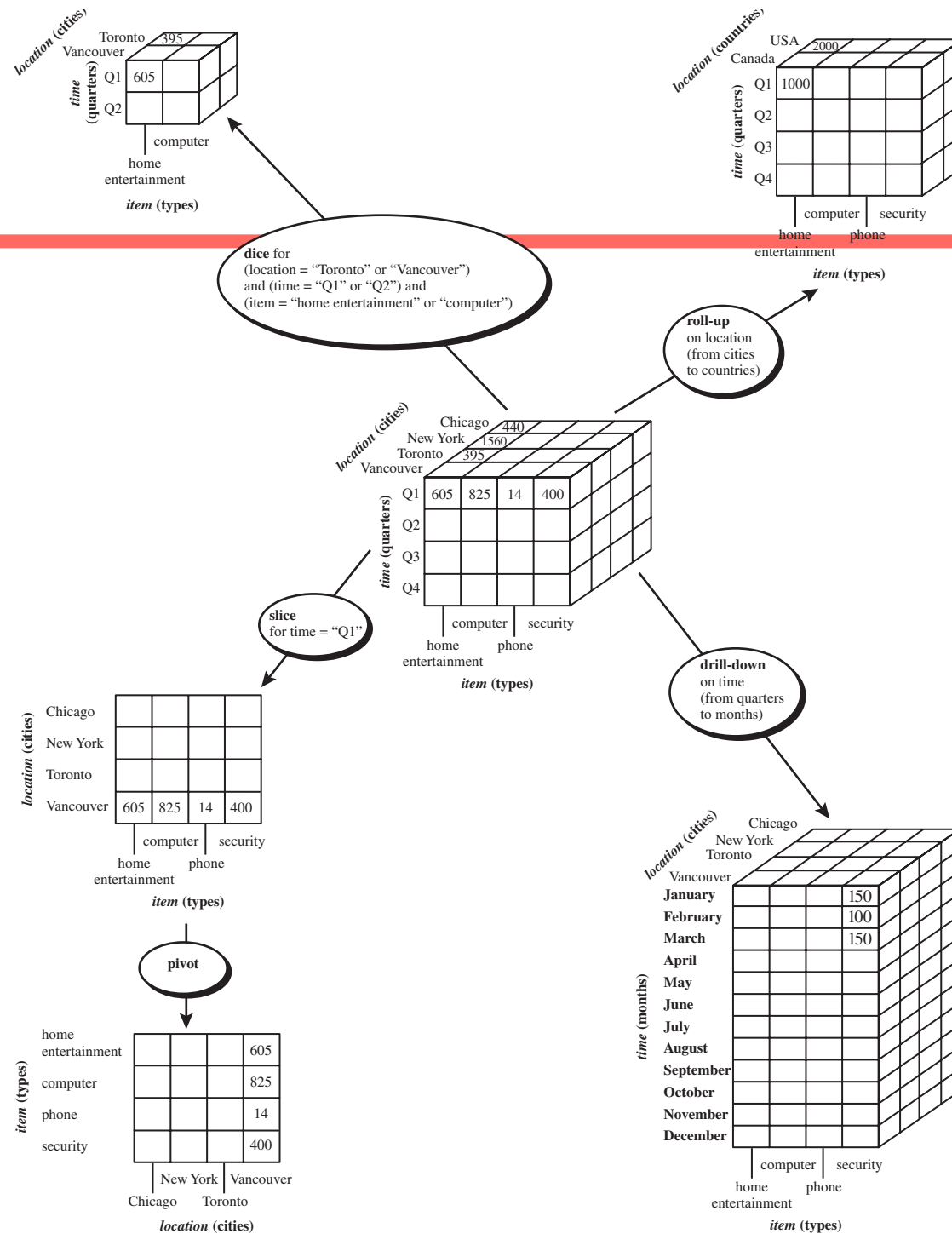
Slice and Dice

- **Example 6**

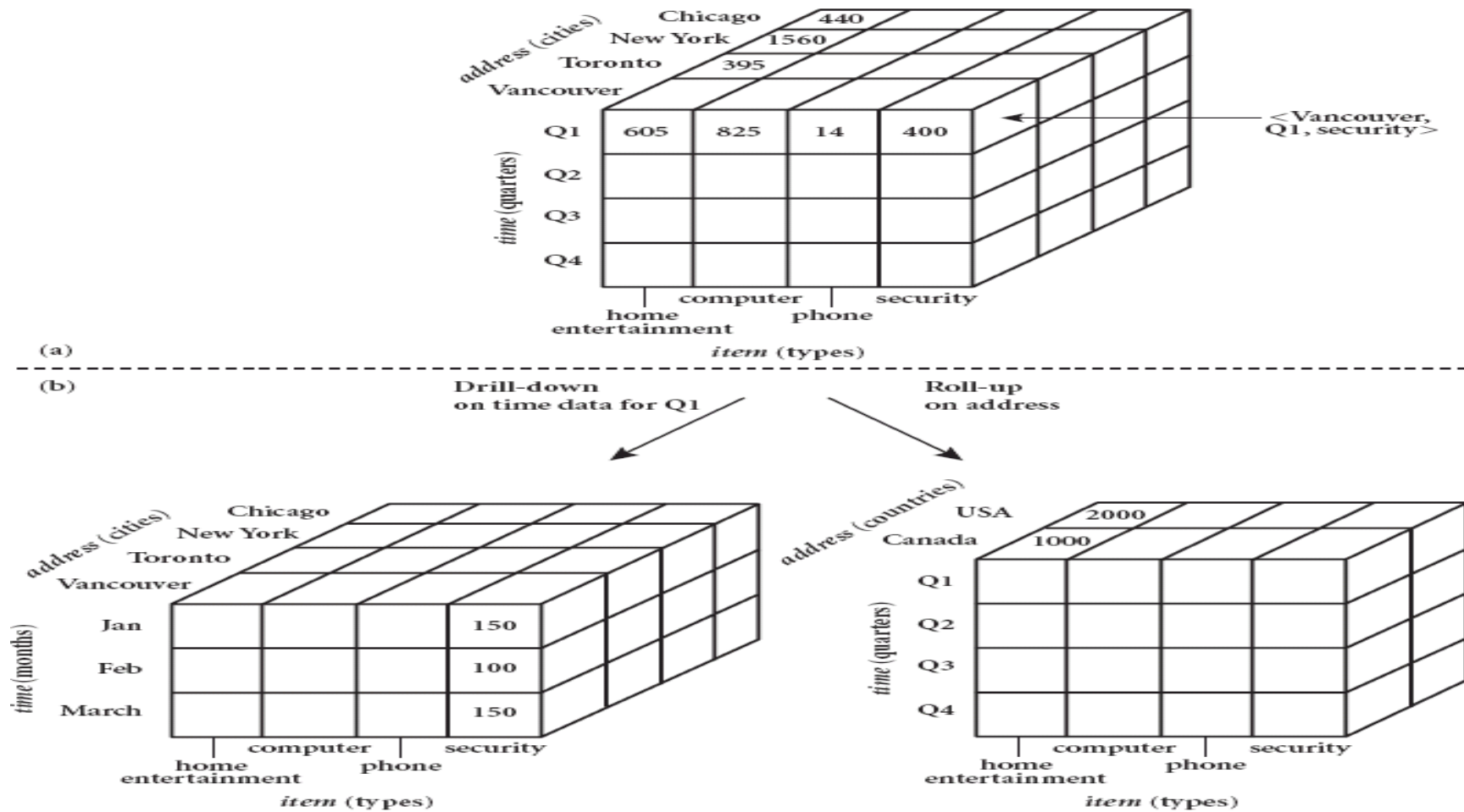
- The *dice* operation defines a subcube by performing a selection on two or more dimensions.
 - The selection criteria involve three dimensions: (*location* = “Toronto” or “Vancouver”) and (*time* = “Q1” or “Q2”) and (*item* = “home entertainment” or “computer”).

Pivot (Rotate)

- A visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data.
- **Example 7:**
 - A pivot operation where the *item* and *location* axes in a 2-D slice are rotated.
- **Example 8:**
 - Rotating the axes in a 3-D cube, or transforming a 3-D cube into a series of 2-D planes.



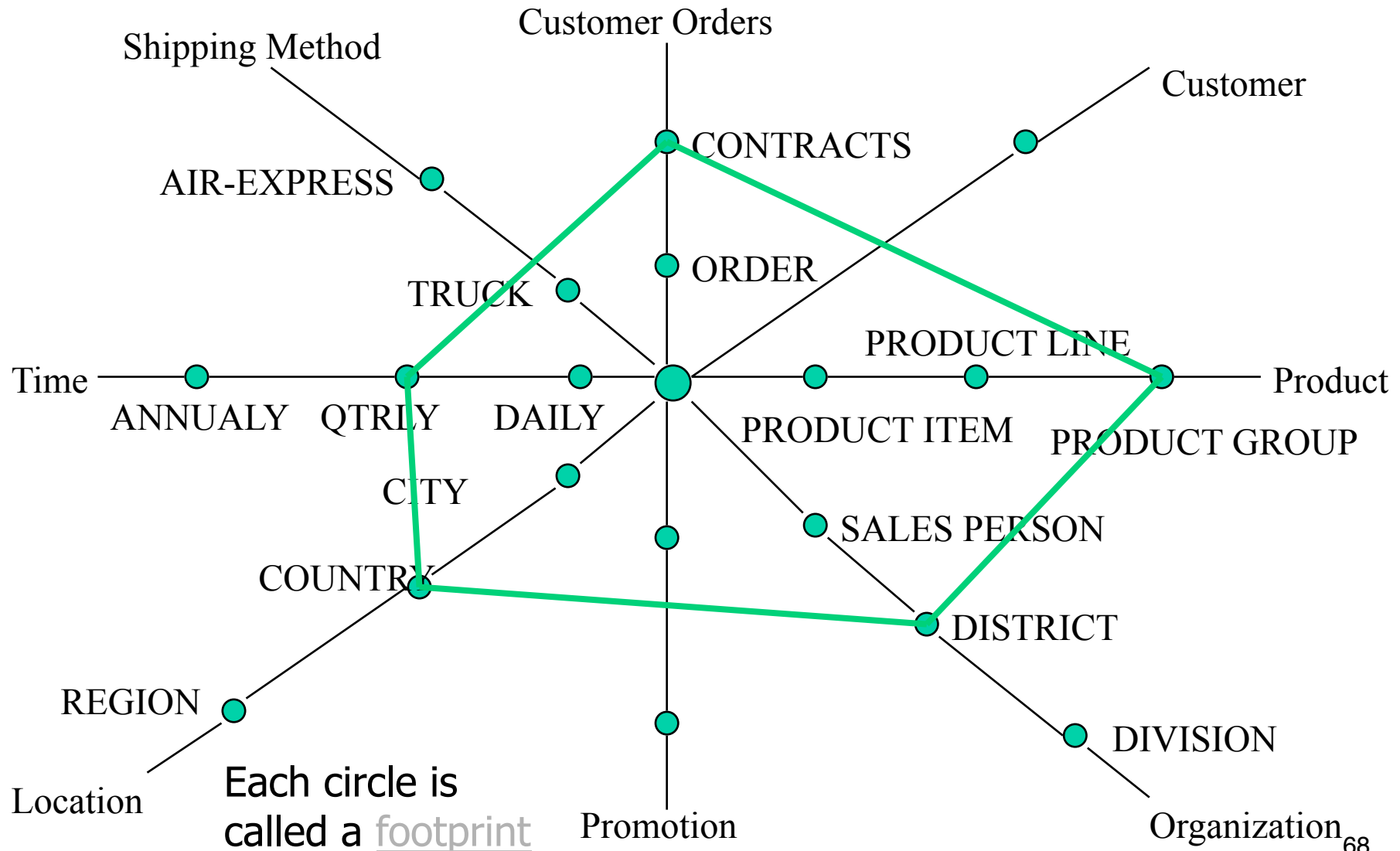
Typical OLAP Operations



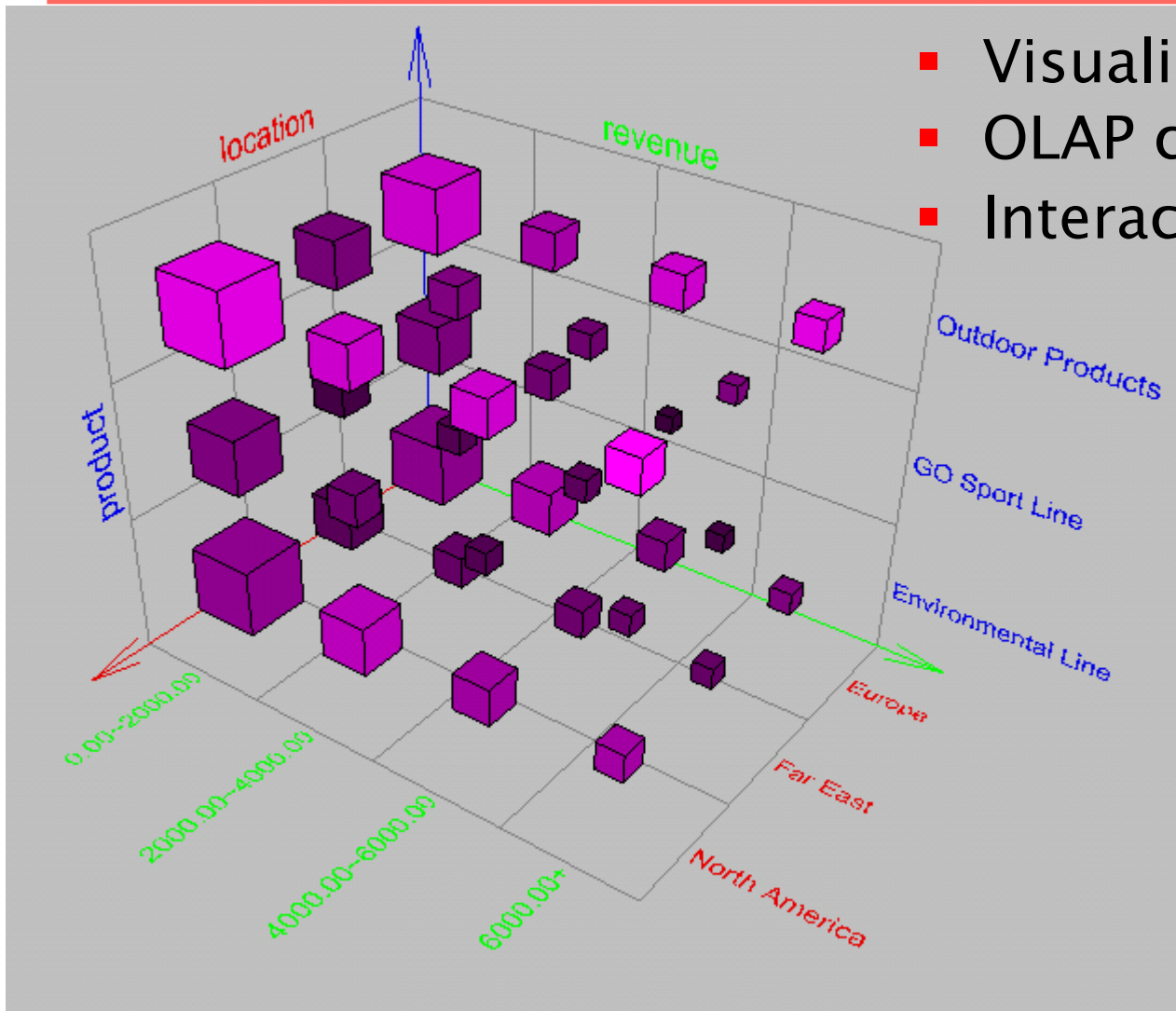
A Starnet Query Model for Querying Multidimensional Databases

- The **querying** of multidimensional databases can be based on a **starnet model**.
- A starnet model consists of radial lines emanating from a central point, where each line represents a concept hierarchy for a dimension.
- Each abstraction level in the hierarchy is called a *footprint*. These represent the granularities available for use by OLAP operations such as drill-down and roll-up.

A Star-Net Query Model




Browsing a Data Cube



- Visualization
- OLAP capabilities
- Interactive manipulation

Data Warehousing and On-line Analytical Processing

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

Design of Data Warehouse: A Business Analysis Framework

- Why a DW?
 - Provide *competitive advantage*
 - measure performance and make critical adjustments
 - Enhance business *productivity*
 - information that accurately describes the organization
 - Facilitate *customer relationship management*
 - consistent view of customers and items across all lines of business, all departments, and all market
 - Bring about *cost reduction*
 - by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner

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 (current and future business needs)
 - 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

Skills to build and use a DW

- *Business skills*
 - To know systems storage and management data
 - To build extractors, warehouse, to refresh software
 - To understand the significance of the data
 - To translate the business requirements into queries
- *Technology skills*
 - To understand how to make assessments from quantitative information
 - To discover patterns and trends, to extrapolate them based on history and look for anomalies
 - To present coherent managerial recommendations
- *Program management skills*
 - To interface with technologies, vendors, and end users, to deliver results in a timely and cost-effective manner

Data Warehouse Design Process

- Top-down, bottom-up approaches or a combination of both
 - Top-down: Starts with overall design and planning, useful when the technology is mature and well known, and where the business problems that must be solved are clear and well understood.
 - Bottom-up: Starts with experiments and prototypes (rapid), useful in the early stage of business modeling and technology development. It is less expensive and evaluate the benefits of the technology before making significant commitments.

Data Warehouse Design Process (ctd.)

- **From software engineering point of view**
 - steps: *planning, requirements study, problem analysis, warehouse design, data integration and testing, and the deployment of the DW*
 - 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
 - a good choice for DW development, especially for data marts, because the turnaround

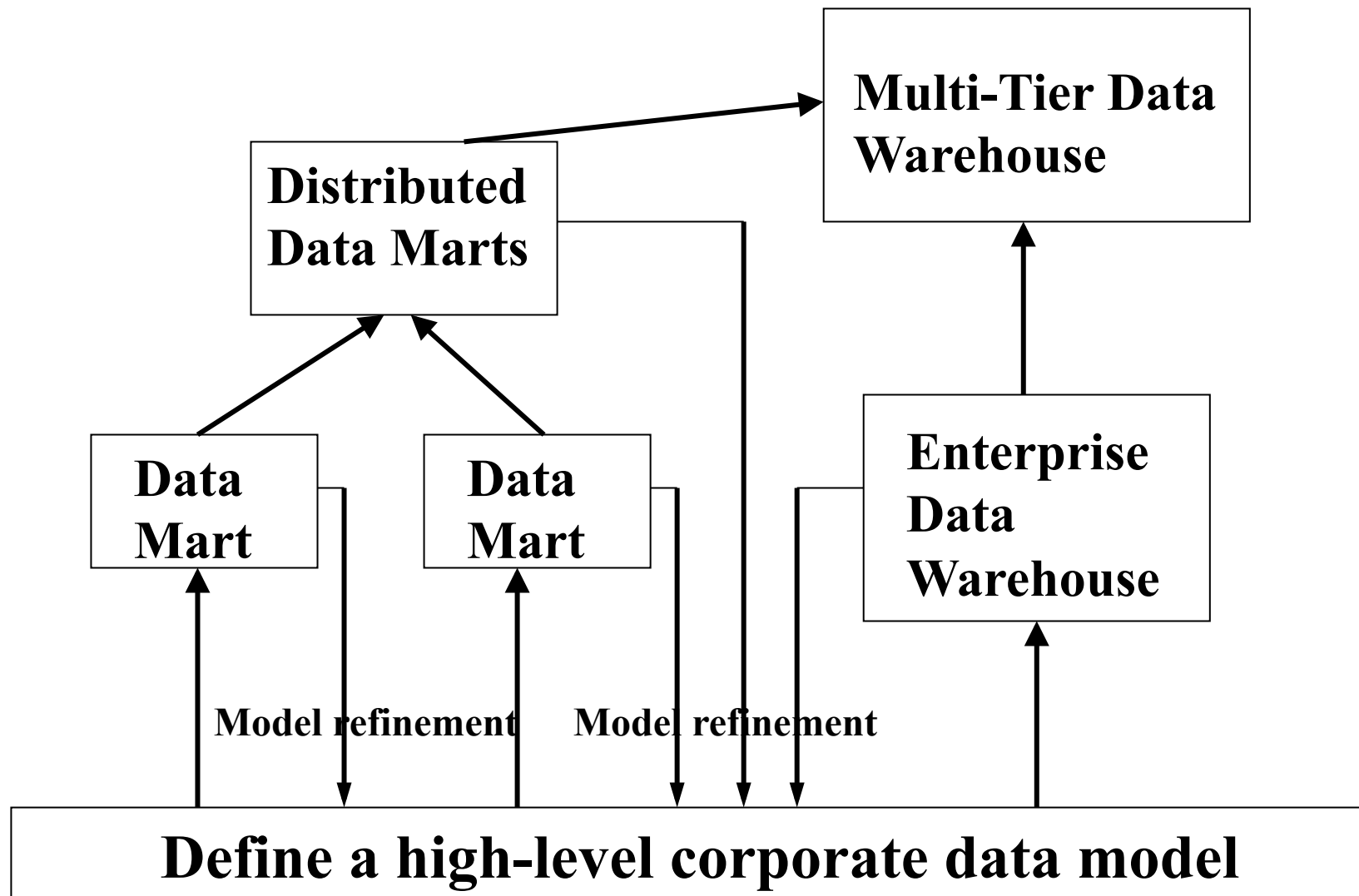
Data Warehouse Design Process (ctd.)

- Typical data warehouse design process
 - Choose a business process to model
 - e.g., orders, invoices
 - Choose the grain (atomic level of data) of the business process
 - e.g., individual transactions, individual daily snapshots
 - Choose the dimensions that will apply to each fact table record,
 - e.g., time, item, customer, supplier, warehouse, transaction, type, and status.
 - Choose the measure that will populate each fact table record
 - e.g, dollars sold and units sold.

Data Warehouse Deployment

- Initial installation, planning, training
- DW administration: data refreshment, data source synchronization, planning for disaster recovery, managing access control and security, managing data growth, managing database performance, and DW enhancement and extension
- Scope management: controlling the number and range of queries, dimensions, and reports; limiting the size of the DW
- DW development tools: functions to define and edit metadata repository contents (such as schemas, scripts, or rules), answer queries, output reports, and ship metadata to and from relational database system catalogue

Data Warehouse Development: A Recommended Approach



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

Summary

- **Data Warehouse: Basic Concepts**
 - (a) What Is a Data Warehouse?
 - (b) Data Warehouse: A Multi-Tiered Architecture
 - (c) Three Data Warehouse Models: Enterprise Warehouse, Data Mart, and Virtual Warehouse
 - (d) Extraction, Transformation and Loading
 - (e) Metadata Repository
- **Data Warehouse Modeling: Data Cube and OLAP**
 - (a) Cube: A Lattice of Cuboids
 - (b) Conceptual Modeling of Data Warehouses
 - (c) Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Databases
 - (d) Dimensions: The Role of Concept Hierarchy
 - (e) Measures: Their Categorization and Computation
 - (f) Cube Definitions in Database systems
 - (g) Typical OLAP Operations
 - (h) A Starlet Query Model for Querying Multidimensional Databases
- **Data Warehouse Design and Usage**
 - (a) Design of Data Warehouses: A Business Analysis Framework
 - (b) Data Warehouses Design Processes
 - (c) Data Warehouse Usage
 - (d) From On-Line Analytical Processing to On-Line Analytical Mining