

# Data Mining

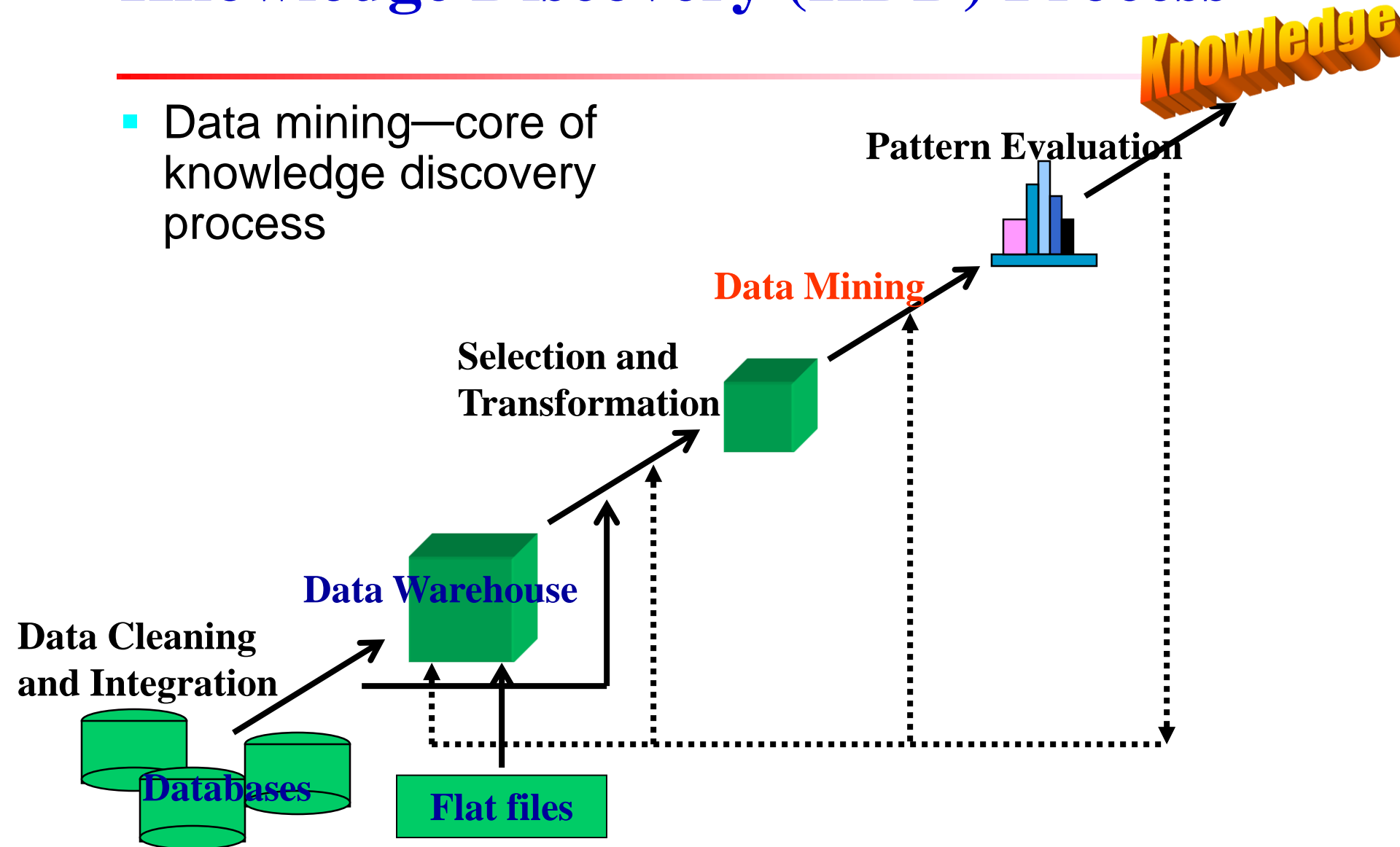
---

**Ying Liu, Prof., Ph.D**

University of Chinese Academy of Sciences

# Knowledge Discovery (KDD) Process

- Data mining—core of knowledge discovery process



# Data Warehouse and OLAP Technology Overview

---

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

# What is Data Warehouse?

---

- “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
- 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

# Data Warehouse

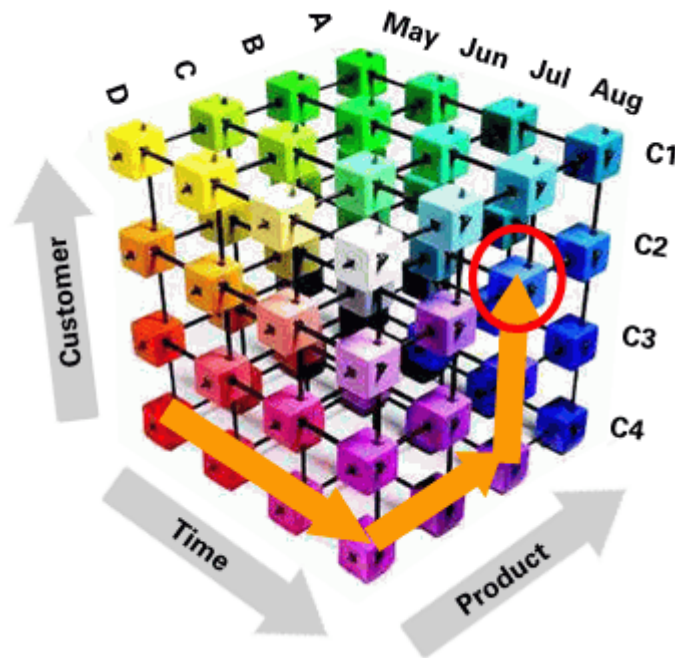
---

- 数据仓库将分布在企业网络中不同信息岛上的业务数据集成到一起，存储在一个单一的集成关系型数据库中，利用这样的集成信息，可方便用户对信息访问，可使决策人员对一段时间内的历史数据进行分析，研究事务的发展走势—**Informix 公司**
- 数据仓库是一种管理技术，旨在通过通畅、合理、全面的信息管理，达到有效的决策支持—**SAS软件研究所**
- 数据仓库是集成信息的存储中心，这些信息可用于查询或分析—**Stanford University**

# Example

---

- Customer relationship management



- Banking decision support system
- Insurance decision support system

# Example

---

- Weather forecasting
  - Air pressure, temperature, longitude/latitude, humidity, time, etc.
  - Slice, drill down, roll up, etc.
  - Query
  - Multi-dimensional visualization

# Data Warehouse—Subject-Oriented

---

- Organized around major subjects, such as **customer, product, sales**
- Focus 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
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
  - 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*

# Data Warehouse vs. Operational DBMS

---

- OLTP (on-line transaction processing)
  - Major task of traditional relational DBMS
  - Day-to-day operations: e.g. purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
  - Major task of data warehouse system
  - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
  - User and system orientation: customer vs. market
  - Data contents: current, detailed vs. historical, consolidated
  - View: current, local vs. evolutionary, integrated
  - Access patterns: update vs. read-only but complex queries

# OLTP vs. OLAP

---

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

# Data Warehouse vs. Heterogeneous DBMS

---

- Traditional heterogeneous DB integration: A **query driven** approach
  - Build **wrappers/mediators** on top of heterogeneous databases
  - When a query is posed to a client site, a meta-dictionary is used to translate the query into queries appropriate for individual heterogeneous sites involved, and the results are integrated into a global answer set
  - Complex information filtering, compete for resources
- Data warehouse: **update-driven**, high performance
  - Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis

# Why 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**: 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: There are more and more systems which perform OLAP analysis directly on relational databases

# Exercise

---

1. For the integration of multiple heterogeneous information sources, many companies in industry prefer the *update-driven* approach (which constructs and uses data warehouses), rather than the *query-driven* approach (which applies wrappers and integrators). Please describe situations where the *query-driven* approach is preferable over the *update-driven* approach.



# Data Warehouse and OLAP Technology Overview

---

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

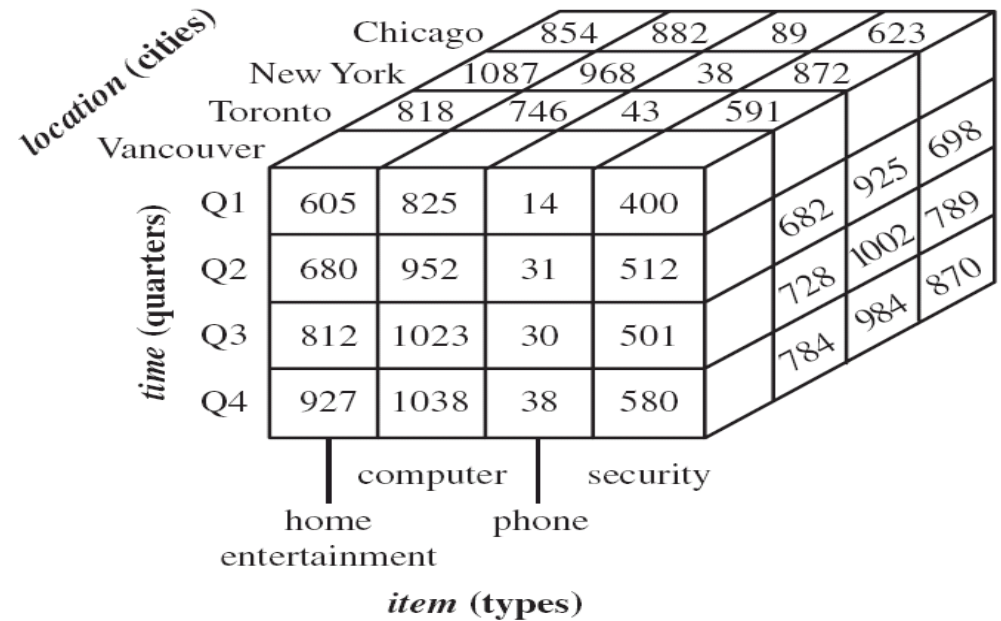
# From Tables and Spreadsheets to Data Cubes

---

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

# From Tables and Spreadsheets to Data Cubes

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

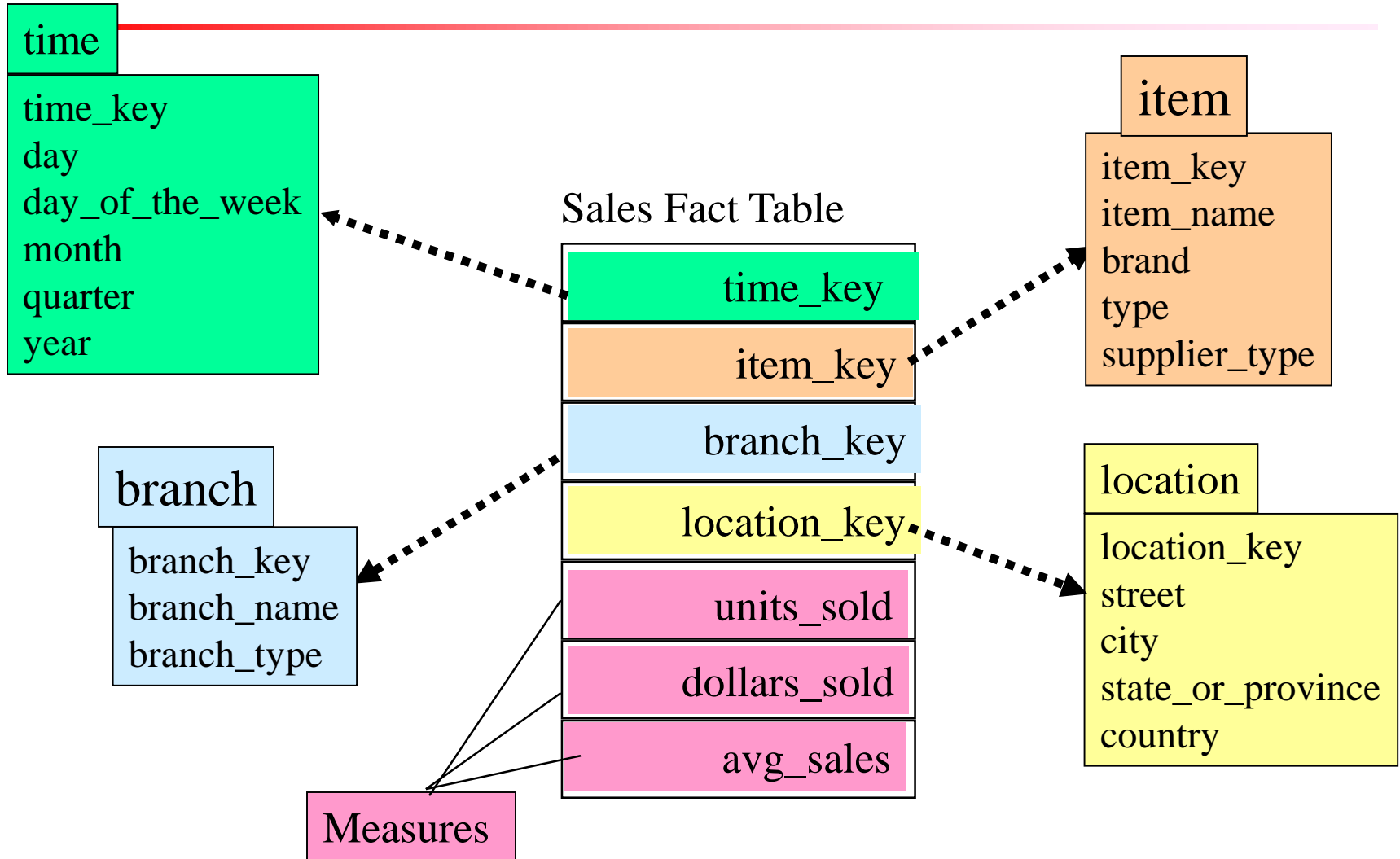


# Conceptual Modeling of Data Warehouses

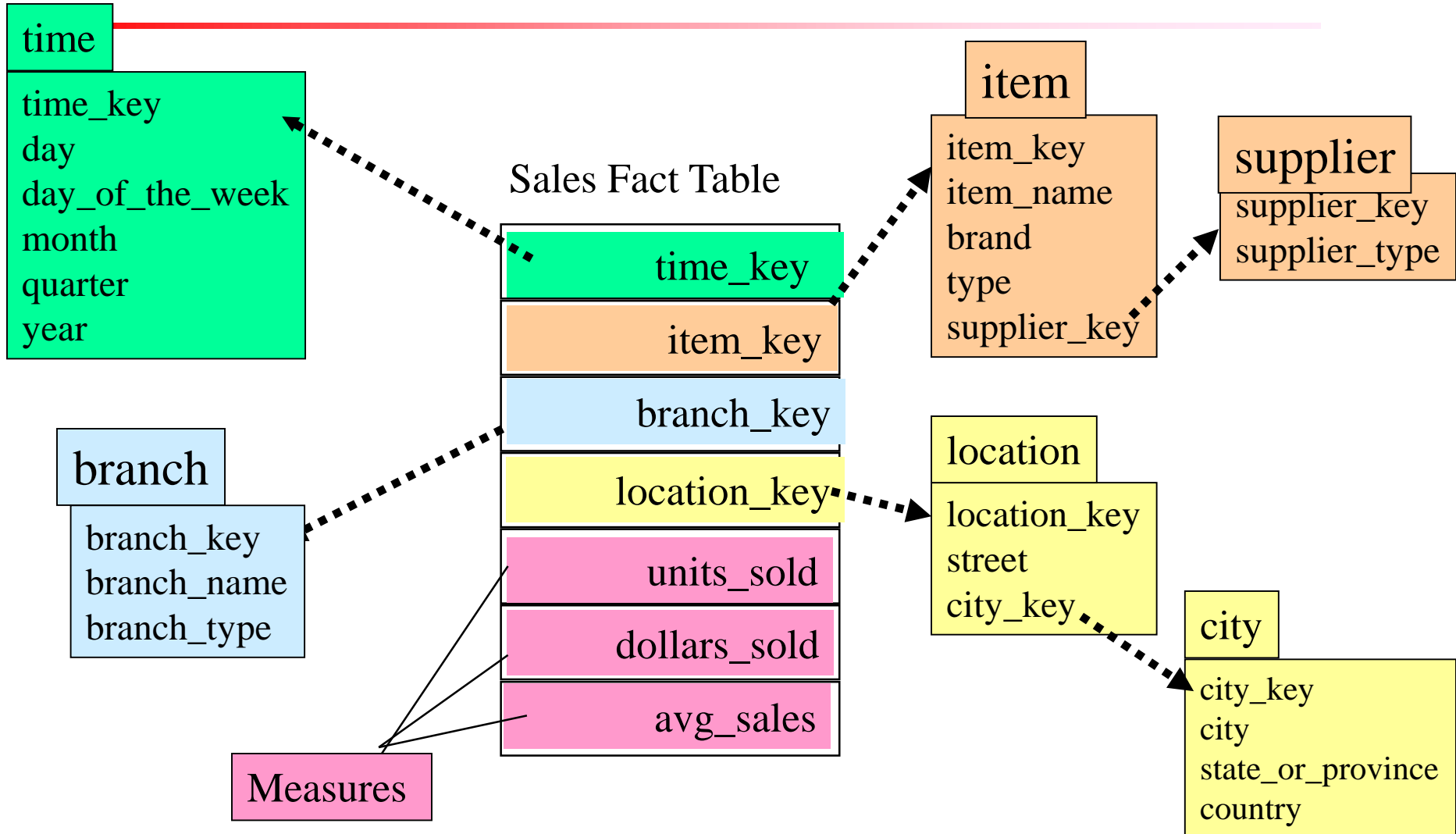
---

- 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

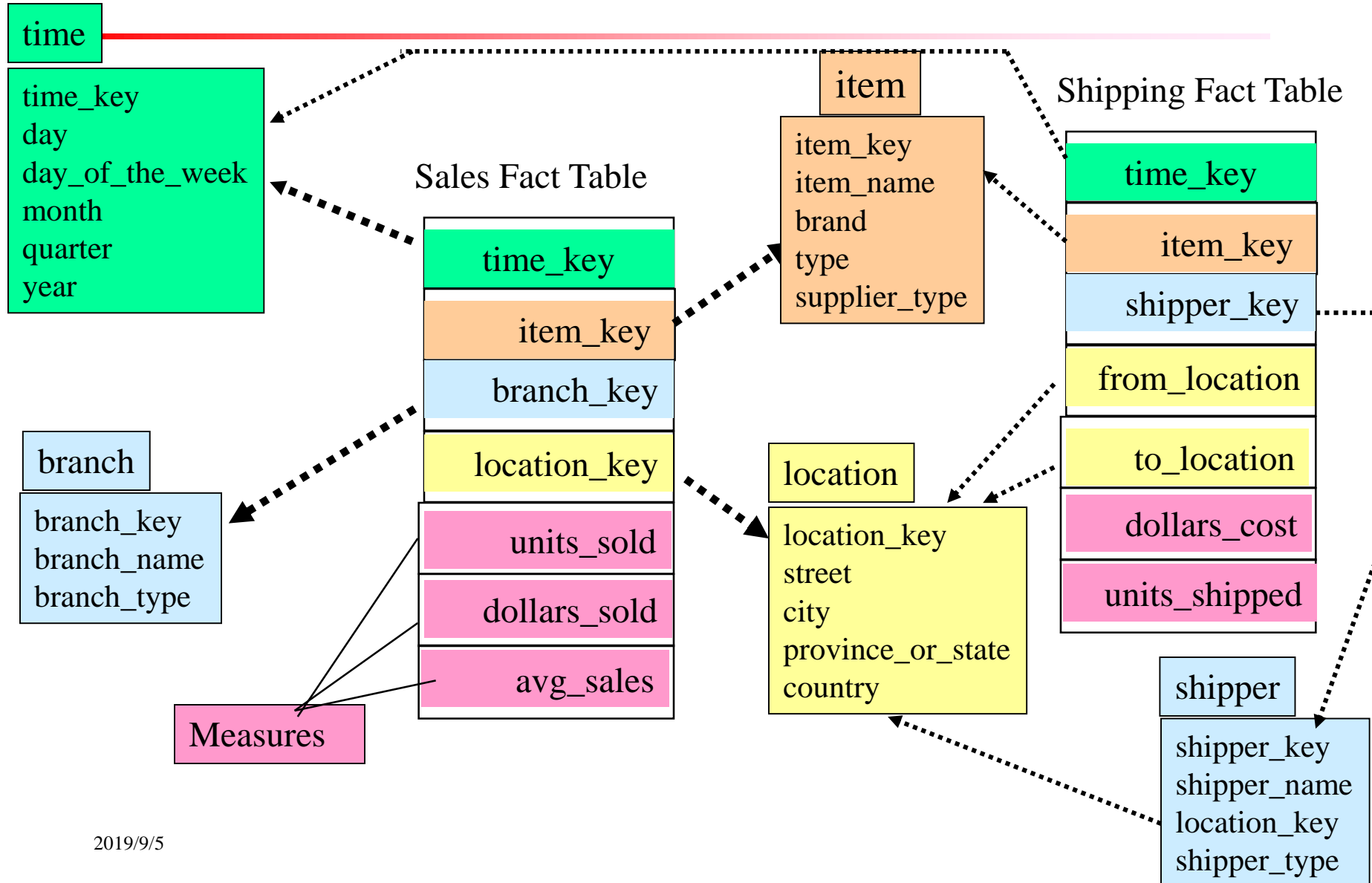
# Example of Star Schema



# Example of Snowflake Schema



# Example of Fact Constellation



# Cube Definition Syntax in 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>



# Defining Star Schema in DML

---

**define cube** sales\_star [time, item, branch, location]:

dollars\_sold, avg\_sales, units\_sold

**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)

# Defining Snowflake Schema in DMQL

---

**define cube** sales\_snowflake [time, item, branch, location]:

dollars\_sold, avg\_sales, units\_sold

**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, province\_or\_state, country))

# Defining Fact Constellation in DMQL

---

```
define cube sales [time, item, branch, location]:  
    dollars_sold, avg_sales, units_sold  
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]:  
    dollar_cost, unit_shipped  
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
```

# Exercise

---

1. Suppose that a data warehouse consists of three dimensions *time*, *doctor*, and *patient*, and two measures count and charge, where charge is the fee that a doctor charges a patient for a visit.  
  
(1) Draw a schema diagram for the data warehouse.

# How to Generate a Specified Data Cube?

---

- DMQL specification is translated into SQL query

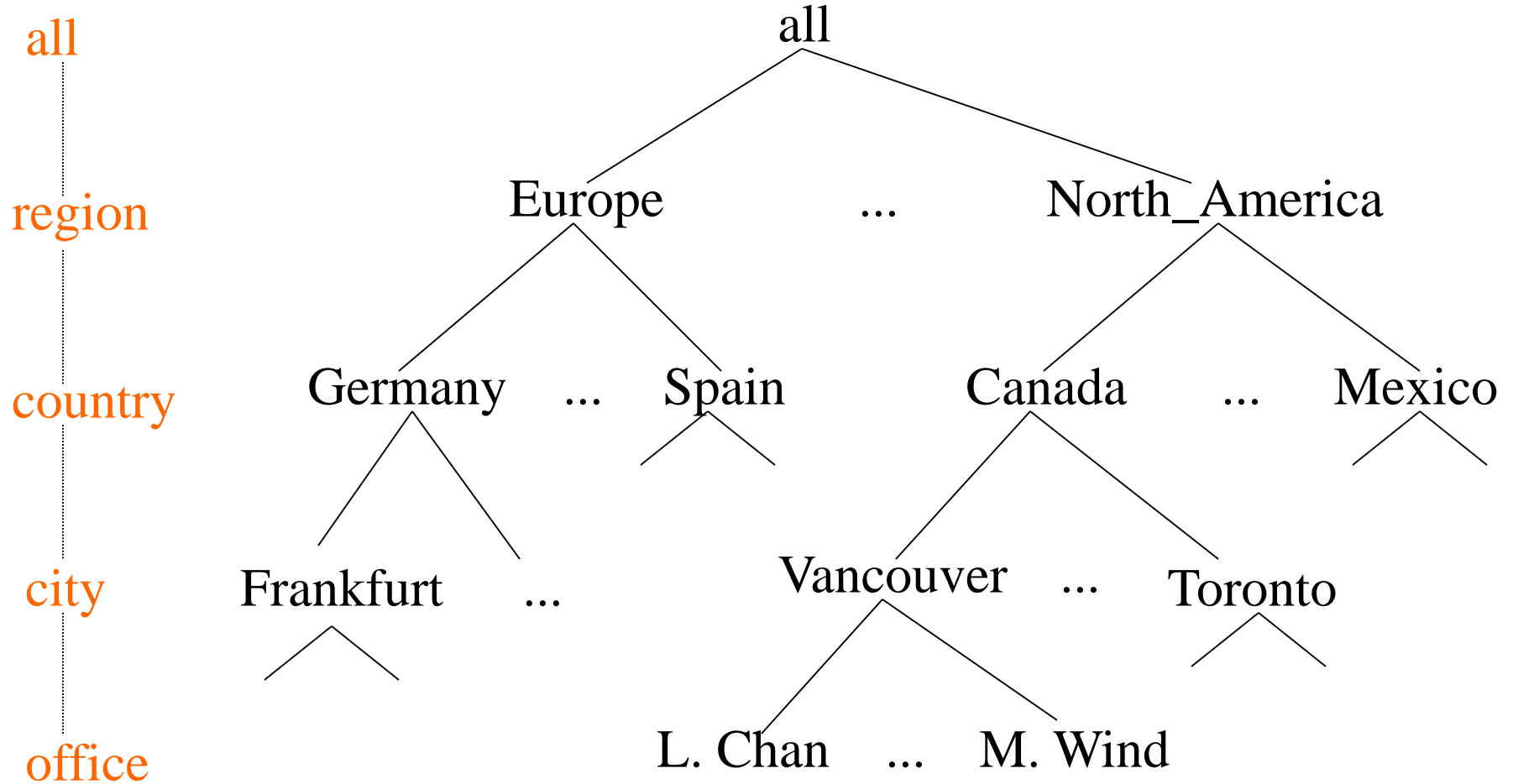
**define cube** sales\_star [time, item, branch, location]:  
dollars\_sold, units\_sold, units\_sold

*translator* ↓

```
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
```

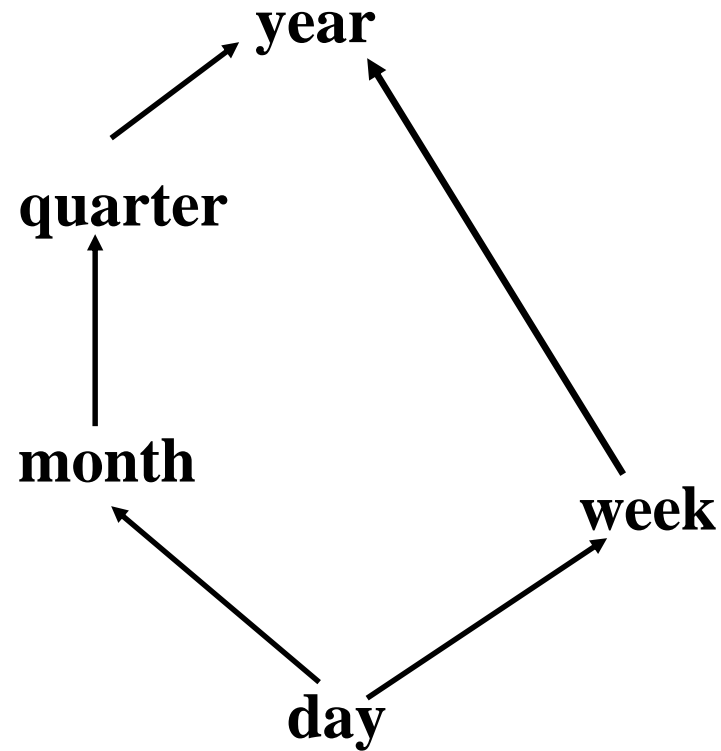
# A Concept Hierarchy: Dimension (location)

---



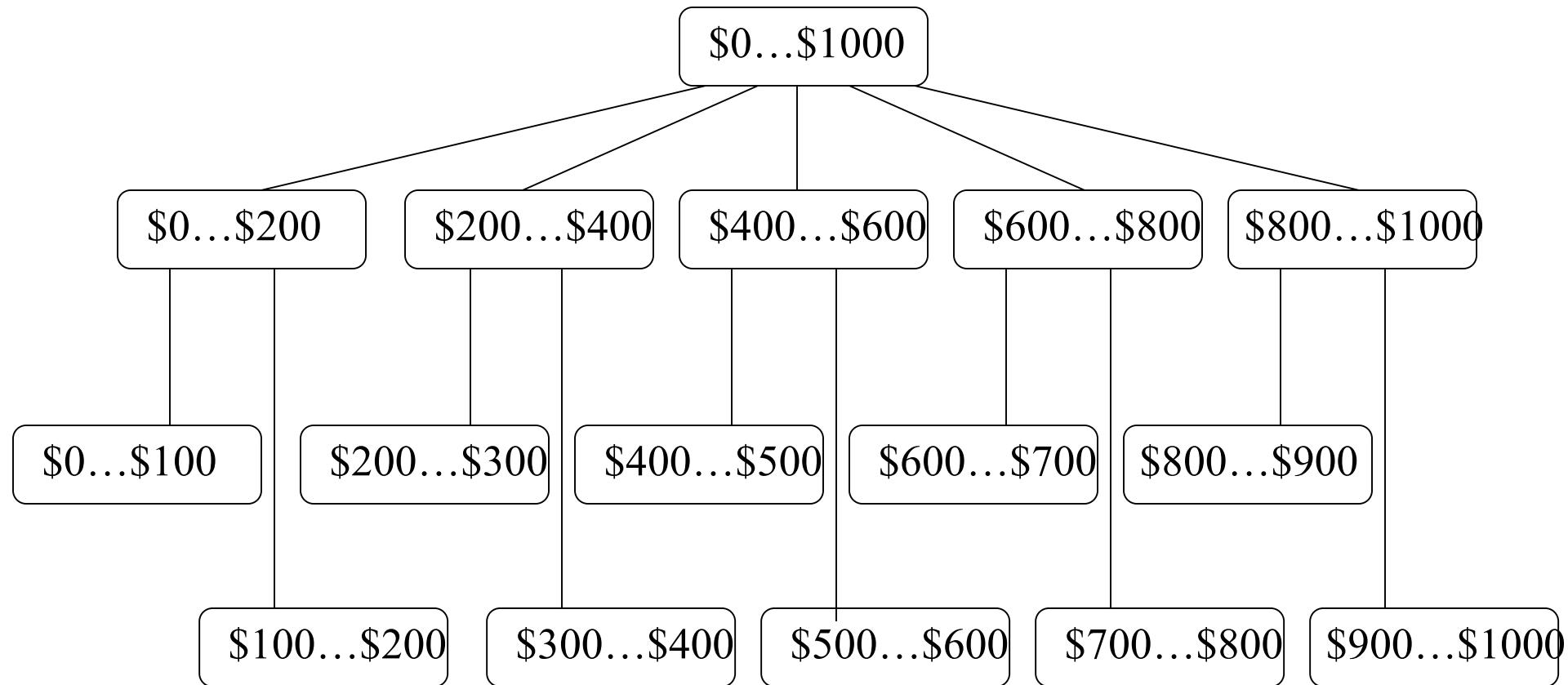
# A Concept Hierarchy: Dimension (time)

---



# A Concept Hierarchy for Numeric Values

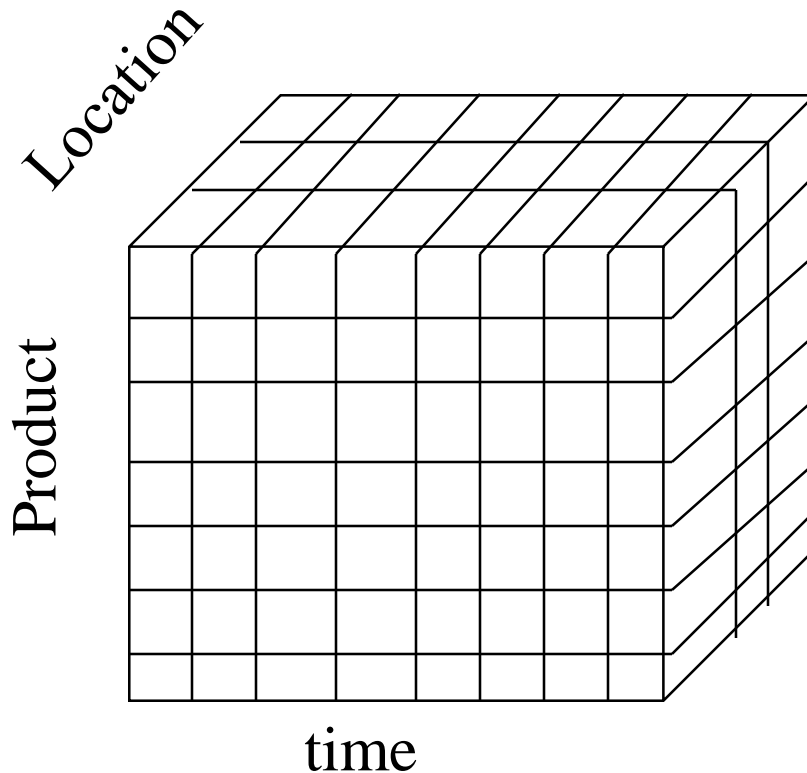
---



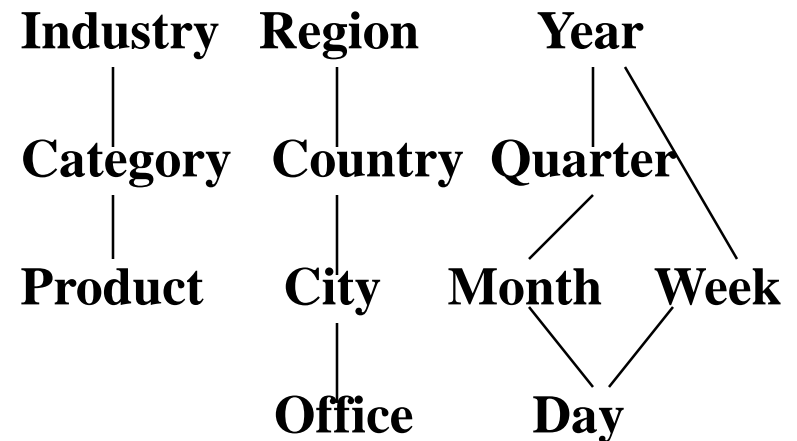


# Multidimensional Data

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



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

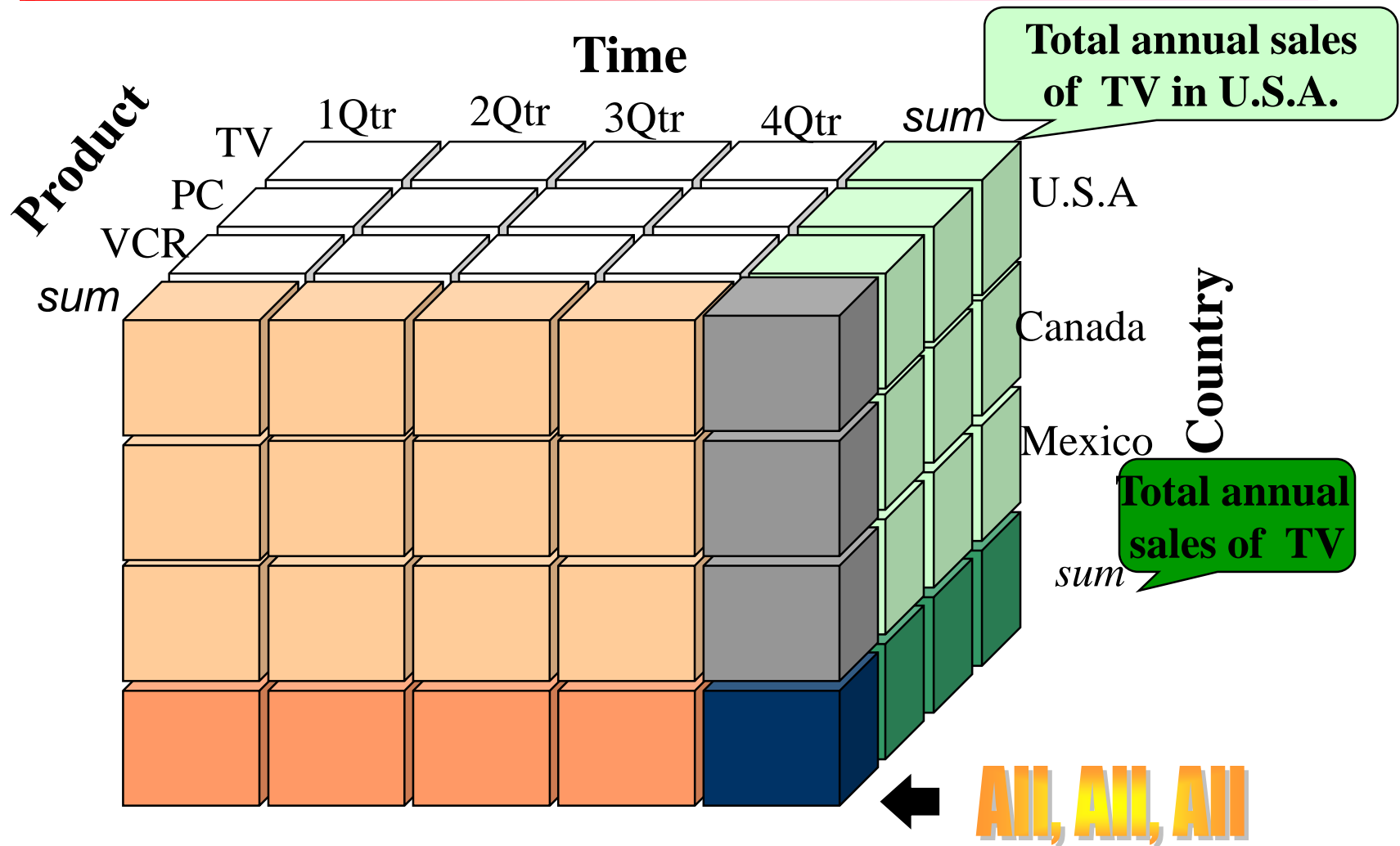


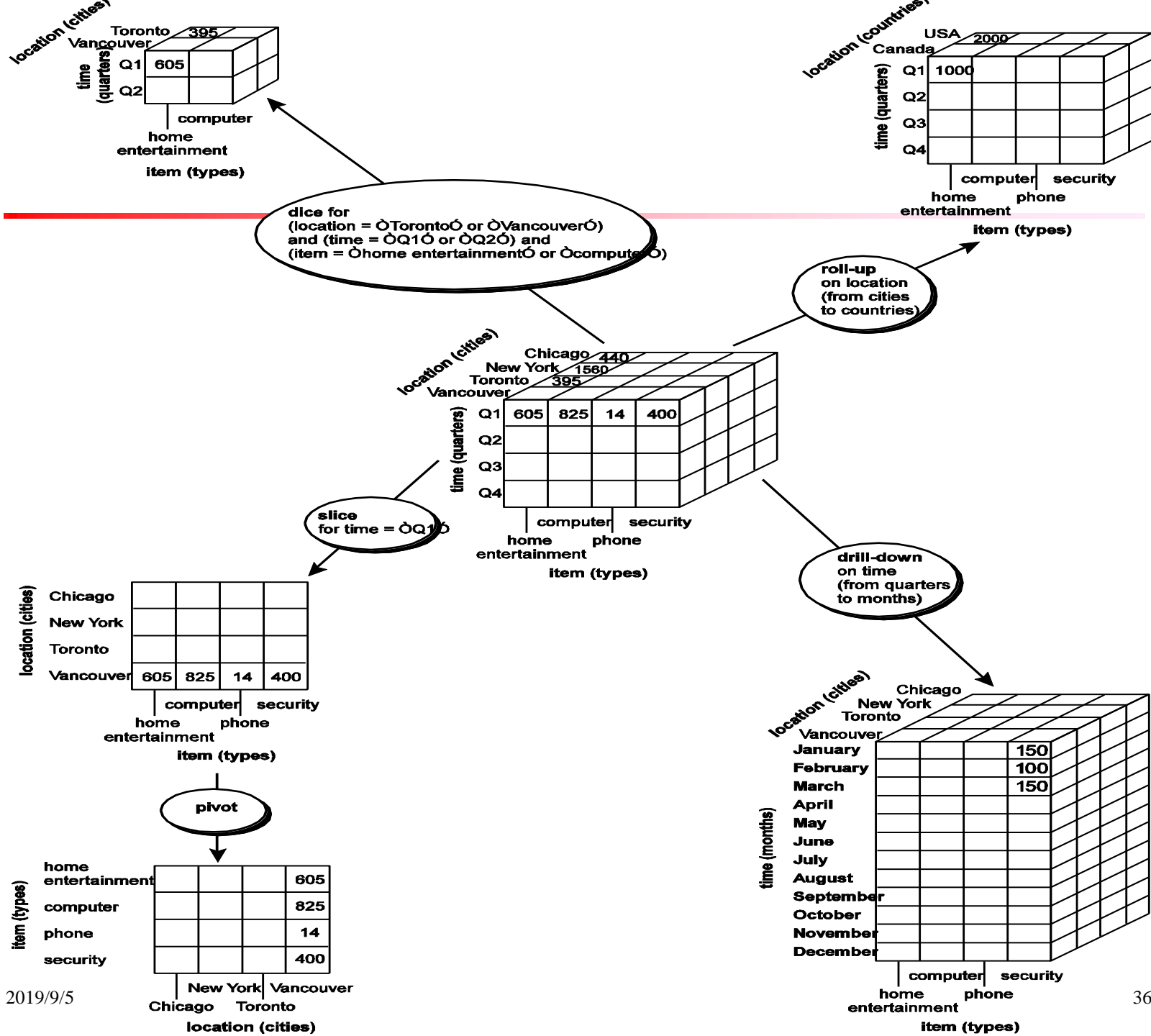
# Typical OLAP Operations

---

- **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*

# A Sample Data Cube





# OLAP Operations

---

## ■ 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)*
- *rank top N or bottom N items in lists*
- *Compute average, variance, deviation*

# Exercise

---

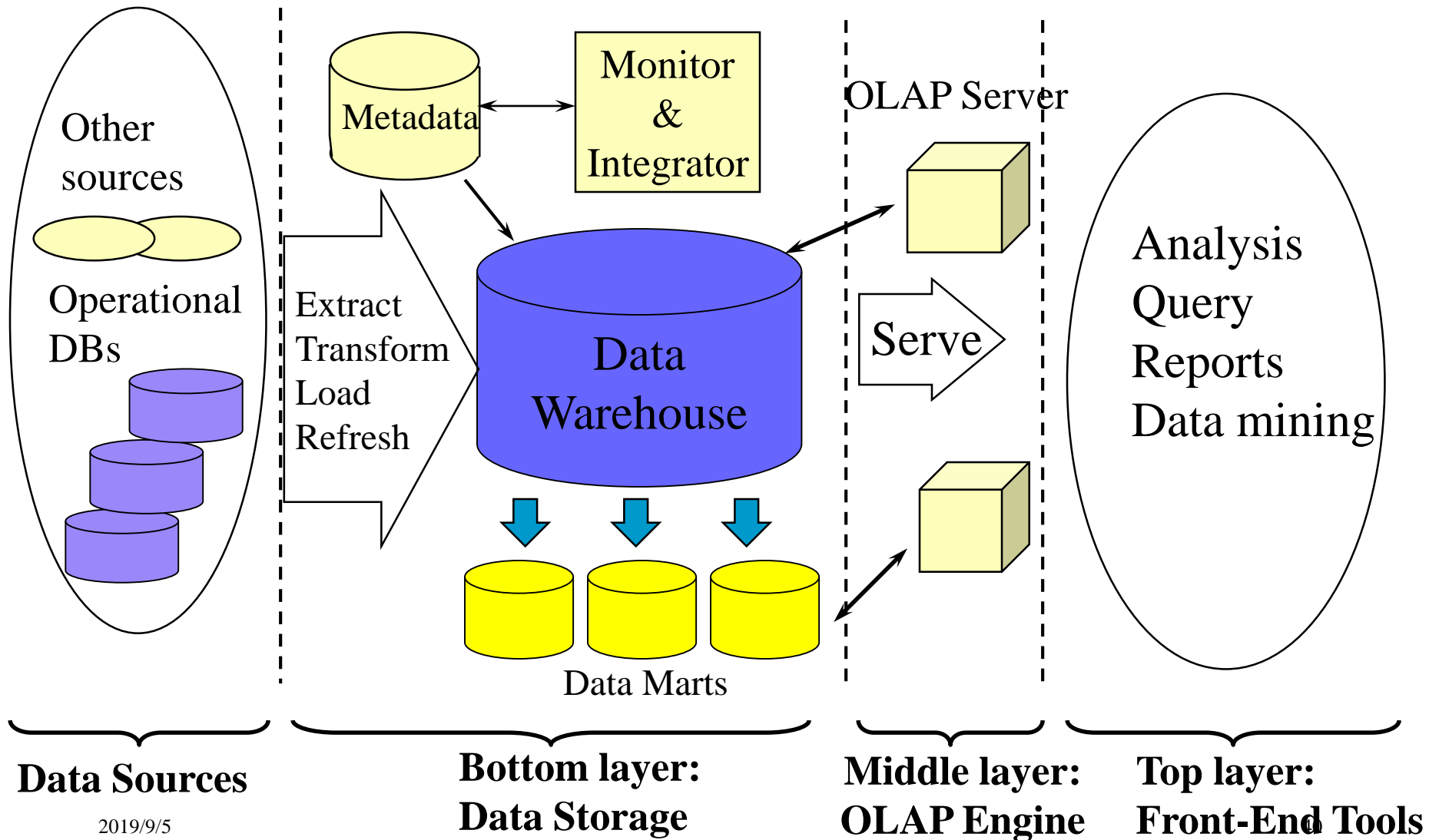
1. Suppose that a data warehouse consists of three dimensions *time*, *doctor*, and *patient*, and two measures count and charge, where charge is the fee that a doctor charges a patient for a visit.
- (2) Starting with the base cuboid [day, doctor, patient], what OLAP operations should be performed in order to list the total fee collected by each doctor in 1999?

# Data Warehouse and OLAP Technology Overview

---

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

# Data Warehouse: A Three-Layer Architecture





# Data Warehouse Back-End Tools and Utilities

---

- 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
- Refresh
  - propagate the updates from the data sources to the warehouse

# Three Data Warehouse Models

---

## ■ Enterprise warehouse

- collect 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 group 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

# Data Mart

## ■ Credit scoring

C_id	sex	age	income	edu	# credit cards	Payment ratio per month	# loans	Payment ratio per month	...
12	0	34	50K	BS.	1	100%	1	100%	...
14	1	29	60K	BS.	2	20%	1	50%	...
135	1	46	100K	MS.	4	100%	2	100%	...
...	...	...	...	...	...	...	...	...	...

## ■ Utility mining

C_id	T_id	A	Profit(A)	B	Profit(B)	C	Profit(C)	D	Profit(D)	...
12	01	0	0	4	5.2	1	0.9	3	5.7	...
14	123	3	6.0	0	0	1	0.9	2	3.8	...
135	12	1	2.0	1	1.3	2	1.8	1	1.9	...
...	...	...	...	...	...	...	...	...	...	...

# Metadata Repository

---

- Meta data is data about data. It contains:
  - Description of the structure of the data warehouse
    - schema, view, dimensions, hierarchies, derived data definition, 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

---

- 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

# 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
- Use parallel computing, bitmap indexing, etc.

# OLAP Server Architectures

---

- **Multidimensional OLAP (MOLAP)**
  - Sparse array-based multidimensional storage engine
  - Fast indexing to pre-computed summarized data
  - Sparse matrix compression technique
- **Hybrid OLAP (HOLAP)** (e.g., Microsoft SQLServer)
  - Flexibility, e.g., low level: relational, high-level: array

# Data Warehouse and OLAP Technology Overview

---

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining



# Cube Operation

- 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

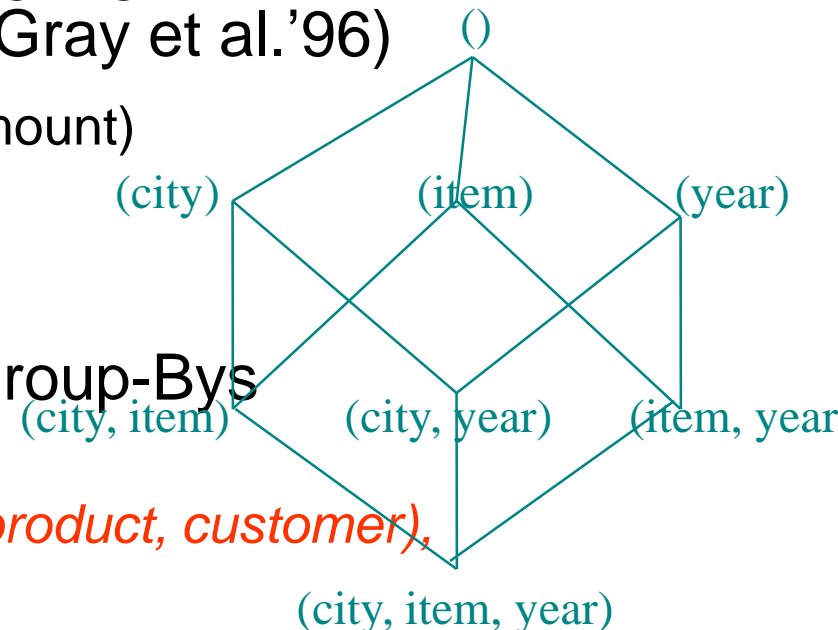
- Need to compute the following Group-Bys

*(date, product, customer),*

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

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

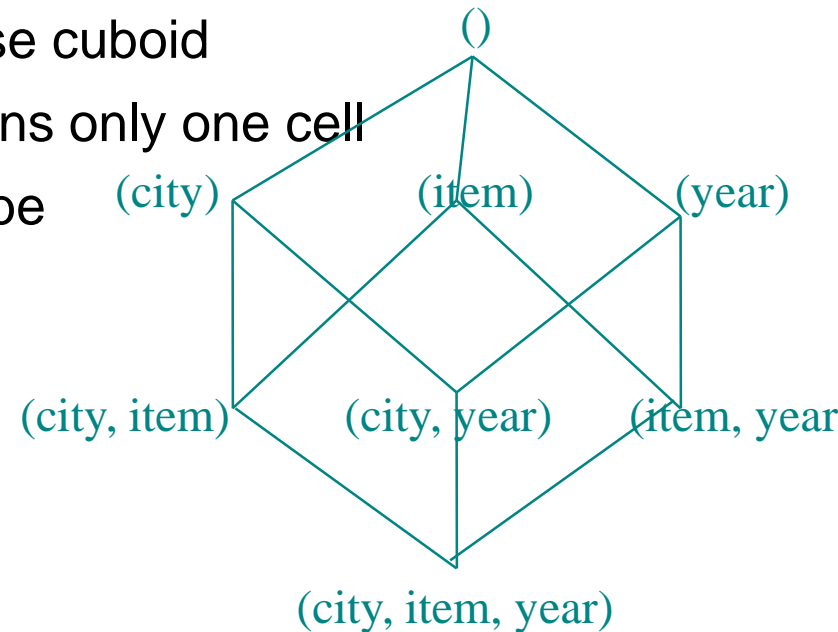
*()*



# 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
- $2^n$  cuboids in an n-dimensional cube



## ■ Materialization of data cube

- Materialize *every* (cuboid) (full materialization), *none* (no materialization), or **some (partial materialization)**
- Selection of which cuboids to materialize
  - Based on size, sharing, access frequency, etc.

# 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

**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**

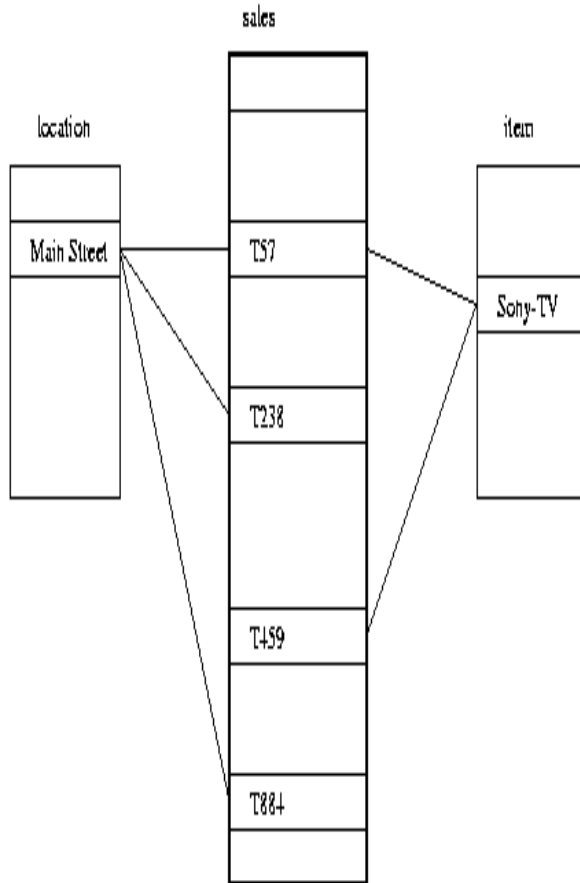
RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0
5	0	1

# Indexing OLAP Data: Join Indices

---

- Join index:  $Jl(R\text{-id}, S\text{-id})$  where  $R(R\text{-id}, \dots) \triangleright \triangleleft S(S\text{-id}, \dots)$
- Traditional indices map the values to a list of record ids
  - It materializes relational join in Join Index file and speeds up relational join
- In data warehouses, join index relates the values of the dimensions of a star 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

# Indexing OLAP Data: Join Indices



Join index table for  
*location/sales*

<i>location</i>	<i>sales_key</i>
...	...
Main Street	T57
Main Street	T238
Main Street	T884
...	...

Join index table for  
*item/sales*

<i>item</i>	<i>sales_key</i>
...	...
Sony-TV	T57
Sony-TV	T459
...	...

Join index table linking two dimensions  
*location/item/sales*

<i>location</i>	<i>item</i>	<i>sales_key</i>
...	...	...
Main Street	Sony-TV	T57
...	...	...

# Exercise

---

1. Suppose a data warehouse for *Big\_University* consists of four dimensions *student*, *course*, *semester*, and *instructor*, and two measures *count* and *score*.
  - (a) Draw a snowflake schema diagram for this data warehouse.
  - (b) Starting with the base cuboid [*student*, *course*, *semester*, *instructor*], what specific OLAP operations should you perform to list the number of CS courses for each *Big\_University* student?
  - (c) If each dimension has five concept levels (including *all*), such as “*student* < *major* < *status* < *university* < *all*”, how many cuboids will this cube contain?
  - (d) Taking this cube as an example, discuss advantages and problems of using a bitmap index structure.

# Exercise

---

2. Suppose a data warehouse has 20 dimensions, each with five concept levels.
  - (a) Users are mainly interested in four particular dimensions, each having three frequently accessed levels for rolling up and drilling down. How would you design a data cube to efficiently support this preference?
  - (b) Occasionally, a user may want to drill through the cube down to its raw relational database for one or two particular dimensions. How would you support this feature?

# Data Warehouse and OLAP Technology Overview

---

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining



# Data Warehouse Usage

---

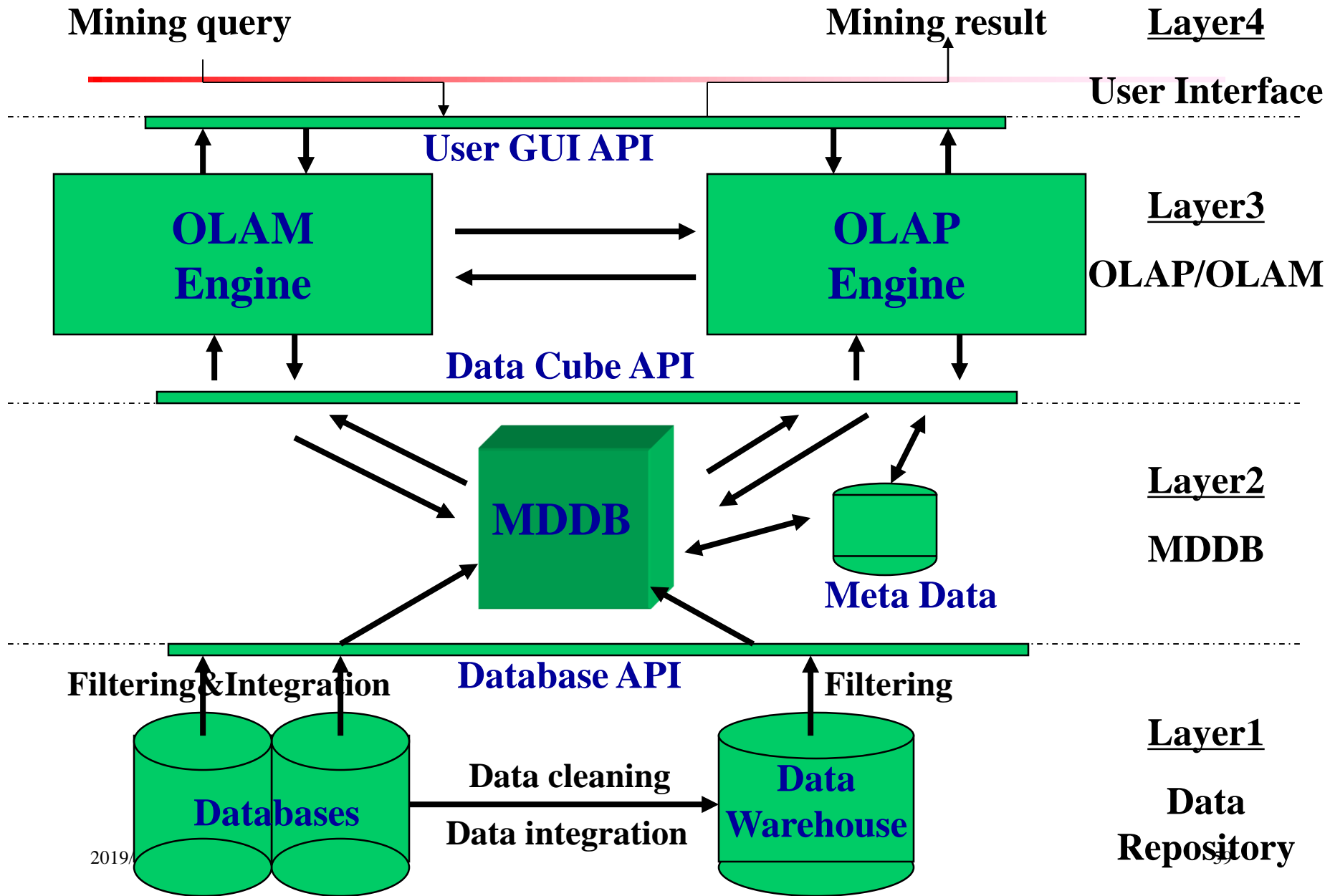
- Three kinds of data warehouse applications
  - Information processing
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - Analytical processing
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - Data mining
    - knowledge 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

# An OLAM System Architecture



# Summary

---

- Why data warehousing?
- A multi-dimensional model of a data warehouse
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures
- OLAP operations: drilling, rolling, slicing, dicing and pivoting
- Data warehouse architecture
- OLAP servers: ROLAP, MOLAP, HOLAP
- Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Indexing OLAP data: Bitmap index and join index
- From OLAP to OLAM (on-line analytical mining)