


# Outline

- ❑ Data Warehouse: Basic Concepts 
- ❑ Data Warehouse Modeling: Data Cube and OLAP
- ❑ Data Warehouse Design and Usage
- ❑ Data Warehouse Implementation
- ❑ Data Generalization by Attribute-Oriented Induction
- ❑ Summary

# What is Datawarehouse ?

- ❑ 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.
- ❑ It is a repository of multiple heterogeneous data sources organized under a unified schema at a single site to facilitate management decision making
- ❑ “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

# Datawarehouse –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**

# 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
    - 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*

# OLTP vs. OLAP

	<b>OLTP</b>	<b>OLAP</b>
<b>users</b>	clerk, IT professional	knowledge worker
<b>function</b>	day to day transactional operations	decision support and data analysis
<b>Data structure</b>	normalized	Star schema or snowflake schema
<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
	Oracle,MySQL,SQL server,DB2	Tableau, Power BI, and SAP

# Why a Separate Data Warehouse?

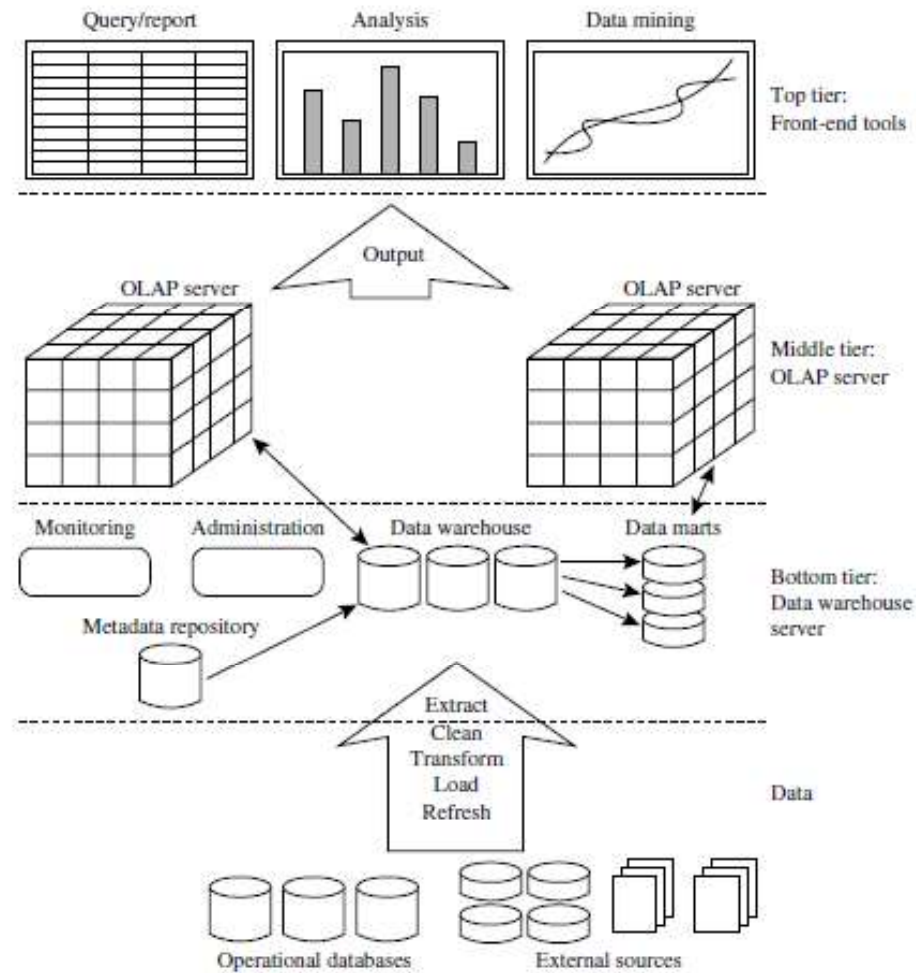
- High performance for both systems
  - DBMS—tuned for OLTP: access methods, indexing, concurrency control, recovery
  - Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation
- Different functions and different data:
  - **missing data**: Decision support requires historical 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



# 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



# 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

# 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

- Data Warehouse: Basic Concepts
- Data Warehouse Modeling: Data Cube and OLAP 
- Data Warehouse Design and Usage
- Data Warehouse Implementation
- Data Generalization by Attribute-Oriented Induction
- 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
- 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**
- A lattice of cuboids is a structure that represents data in a data warehouse as a data cube, which is a multidimensional model that allows data to be viewed in multiple dimensions.



## 2-D Data Cube (Table) Example

2-D View of Sales Data for *AllElectronics* According to *time* and *item*

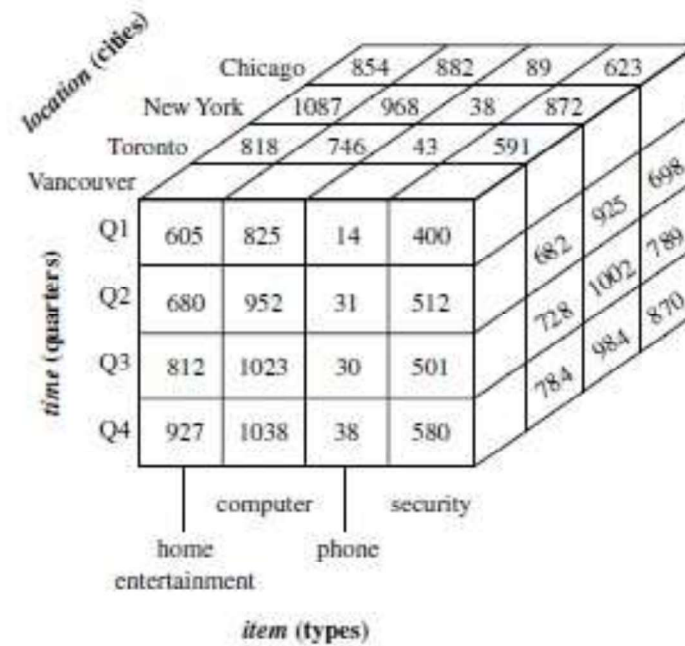
<i>location</i> = "Vancouver"				
<i>time</i> (quarter)	<i>item</i> (type)			
	<i>home entertainment</i>	<i>computer</i>	<i>phone</i>	<i>security</i>
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

## 3-D Data Cube Represented as 2 or more 2-D Datacubes

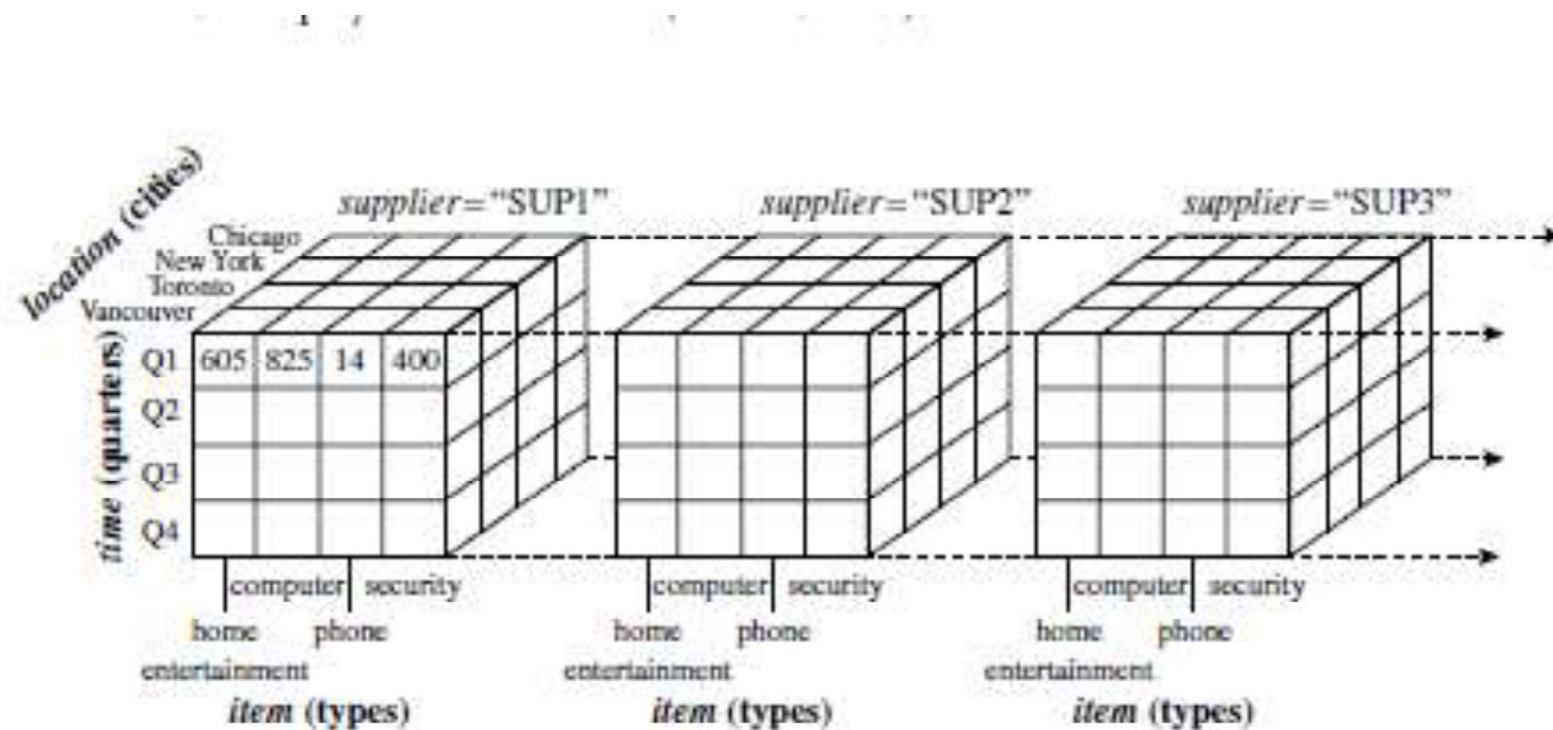
**Table 4.3** 3-D View of Sales Data for *AllElectronics* According to *time*, *item*, and *location*

<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>time</i>	<i>home</i>				<i>home</i>					<i>home</i>					<i>home</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	

# 3-D Data Cube



## 4-D Data Cube represented as 2 or more 3-D Datacubes



# Example for 3-D and 2-D cuboids

3-D view of sales

	location = BVM				location = Vijaya Durga			
	item				item			
time	H.E	C	P	S	H.E	C	P	S
Q <sub>1</sub>	5	5	3	6	2	20	10	5
Q <sub>2</sub>	3	4	4	2	3	10	15	8
Q <sub>3</sub>	2	5	5	7	5	10	20	6
Q <sub>4</sub>	1	2	8	9	10	10	20	4

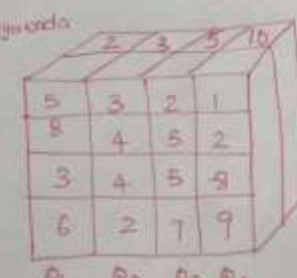
If we aggregate along 'location' dimension we get

	item			
time	H.E	C	P	S
Q <sub>1</sub>	7	28	13	11
Q <sub>2</sub>	6	14	19	10
Q <sub>3</sub>	7	15	25	13
Q <sub>4</sub>	11	12	33	13

Vijaya Durga

BVM

	H.E	C	P	S
Q <sub>1</sub>	5	3	2	1
Q <sub>2</sub>	8	4	5	2
Q <sub>3</sub>	3	4	5	8
Q <sub>4</sub>	6	2	7	9

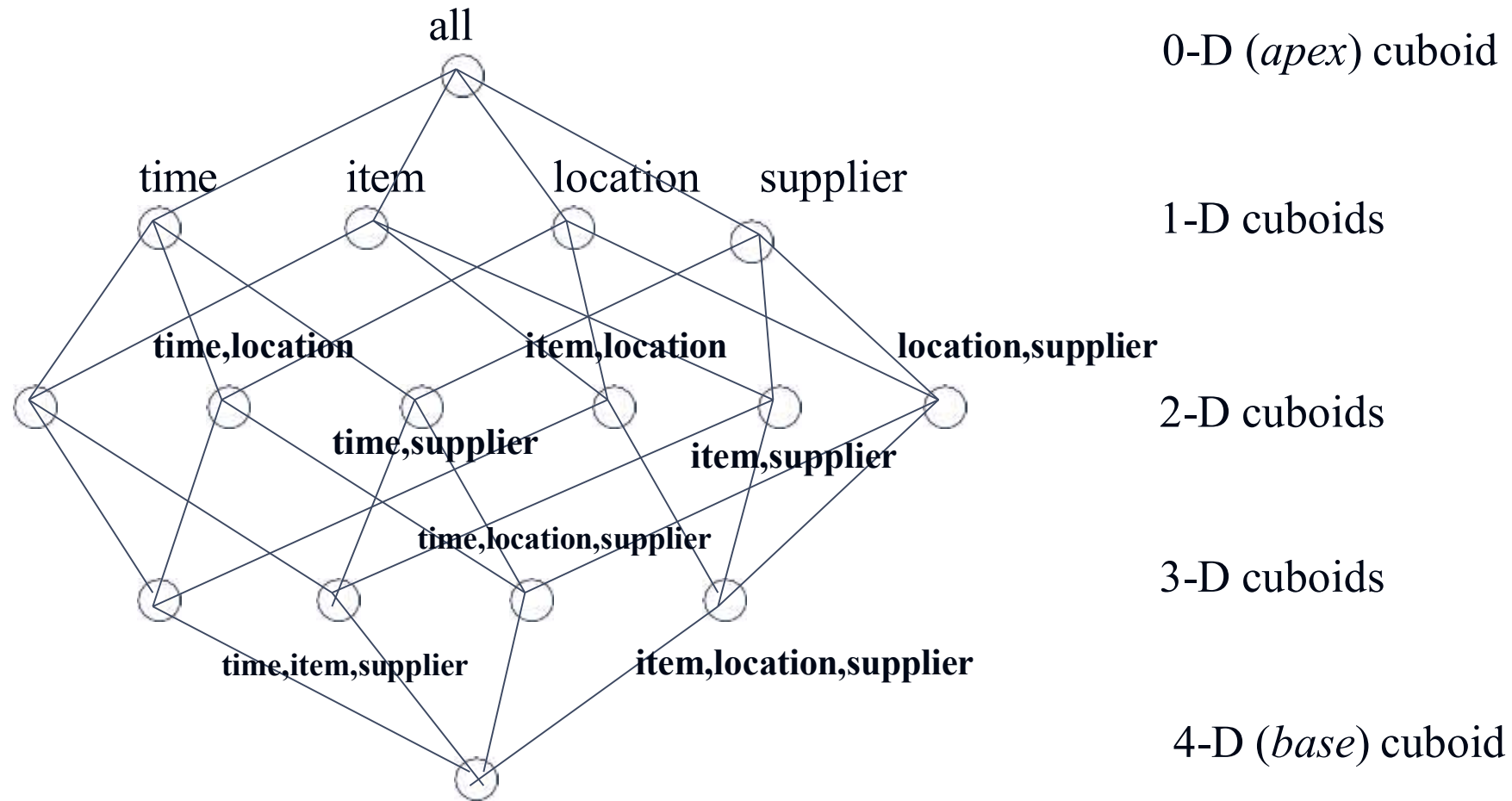


3-D Data Cube shown as two 2-D tables

We get 2-D Data Cube if we aggregate the above data along location dimension

3-D Data Cube

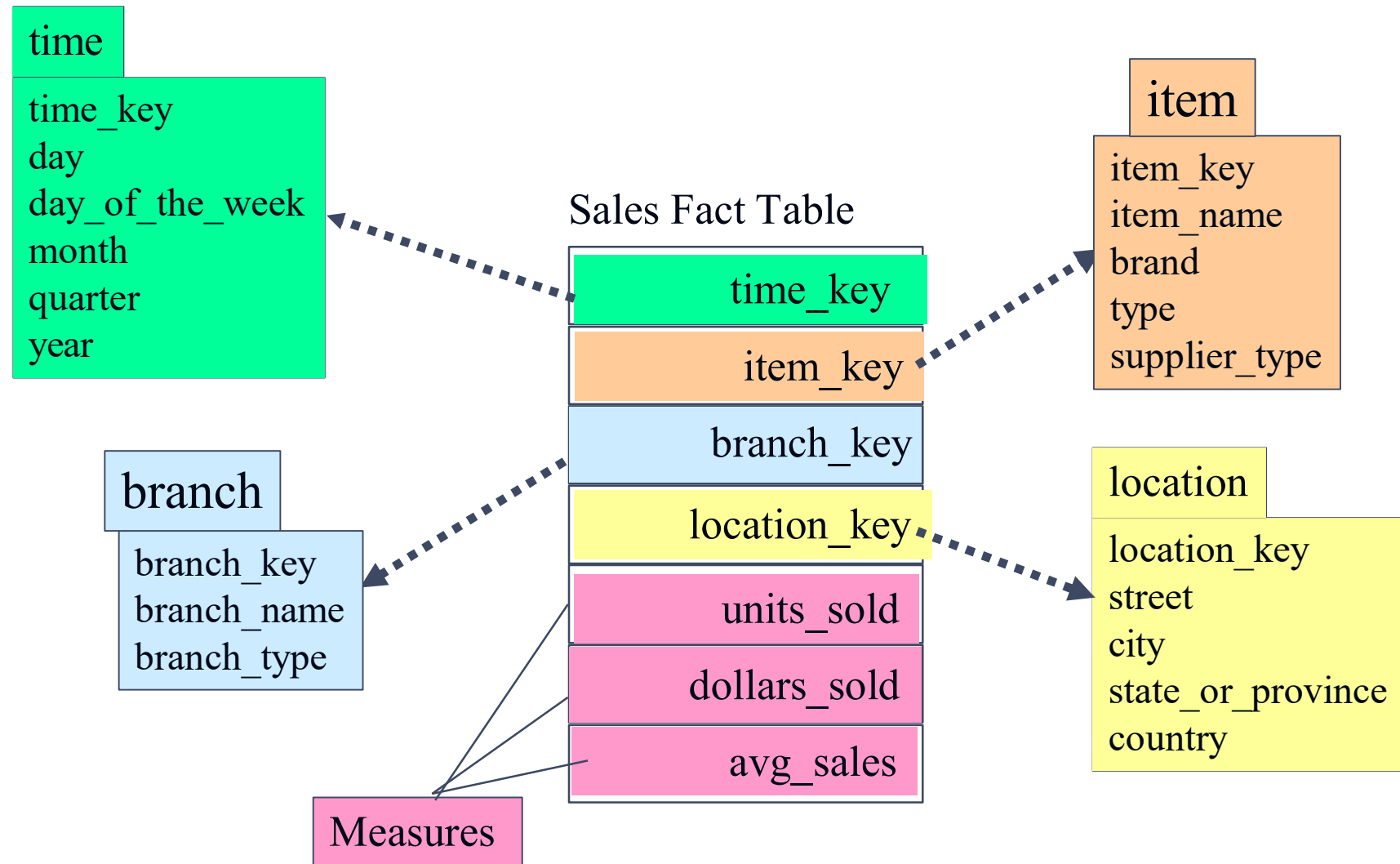
# Cube: A Lattice of Cuboids



# Conceptual Modeling of Data Warehouses

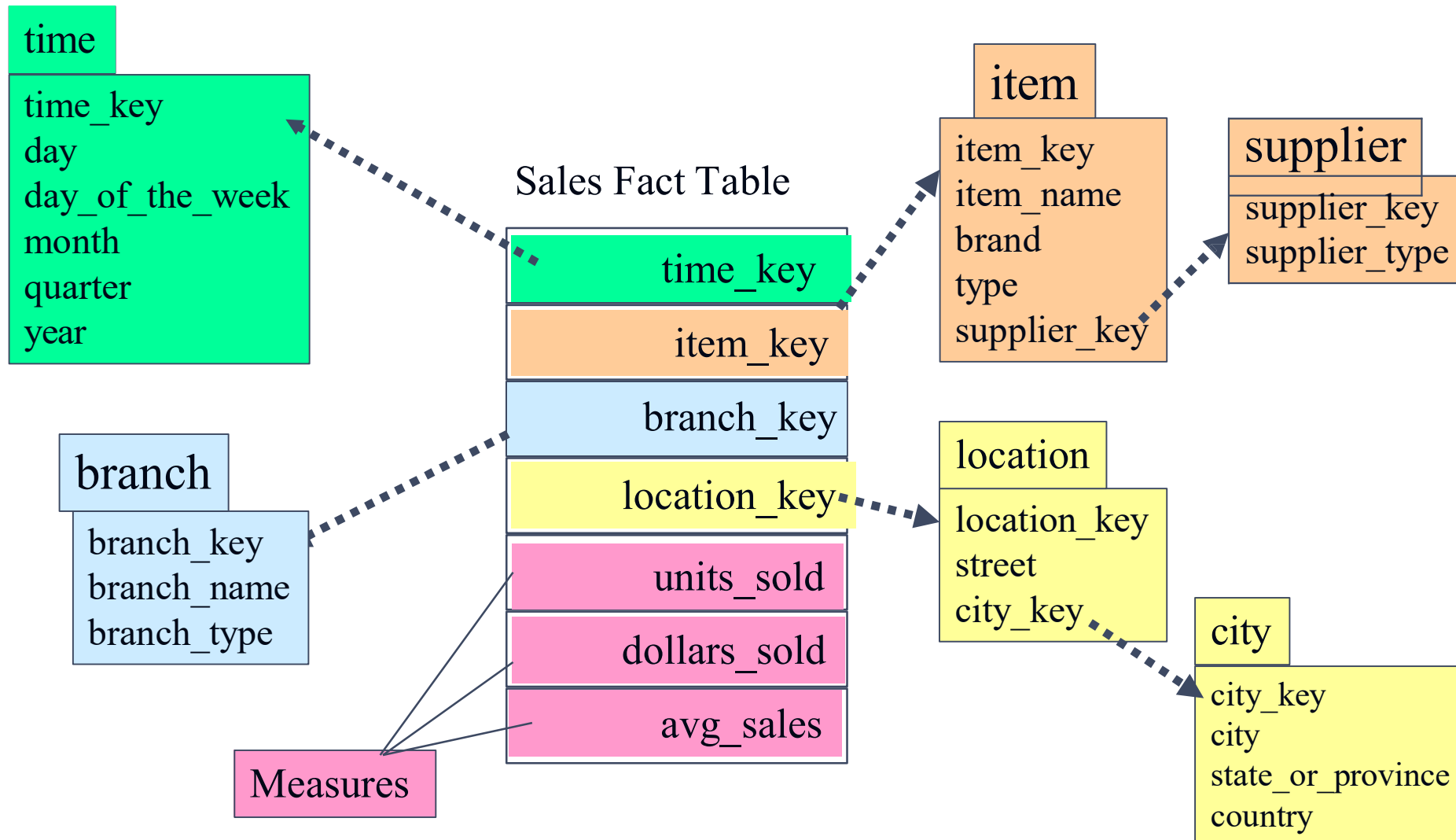
- Modeling data warehouses: dimensions & measures
  - **Star schema**: A fact table in the middle connected to a set of dimension tables
  - **Snowflake schema**: A refinement of star schema where some dimensional hierarchy is **normalized** into a set of smaller dimension tables, forming a shape similar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

# Example of Star Schema

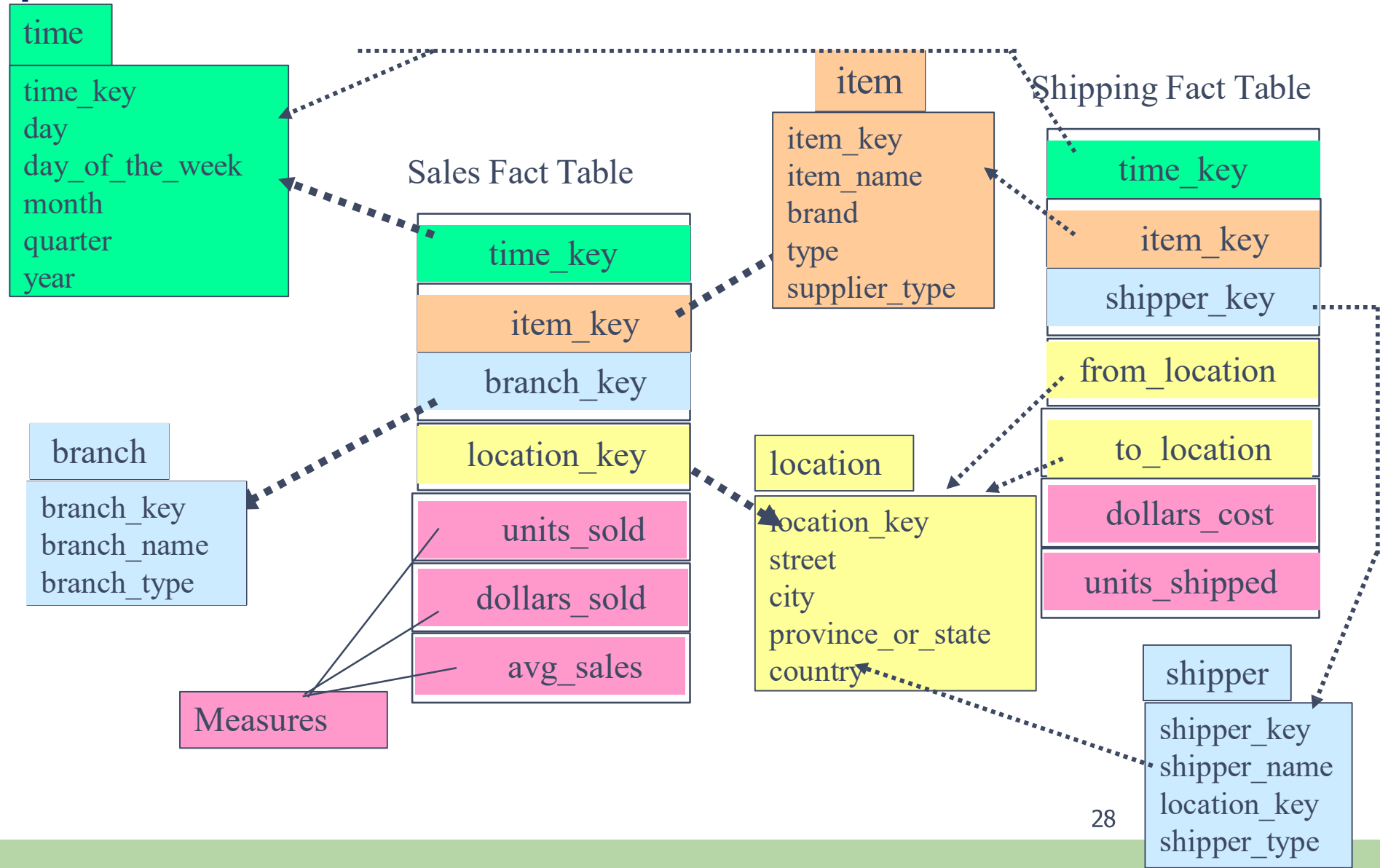




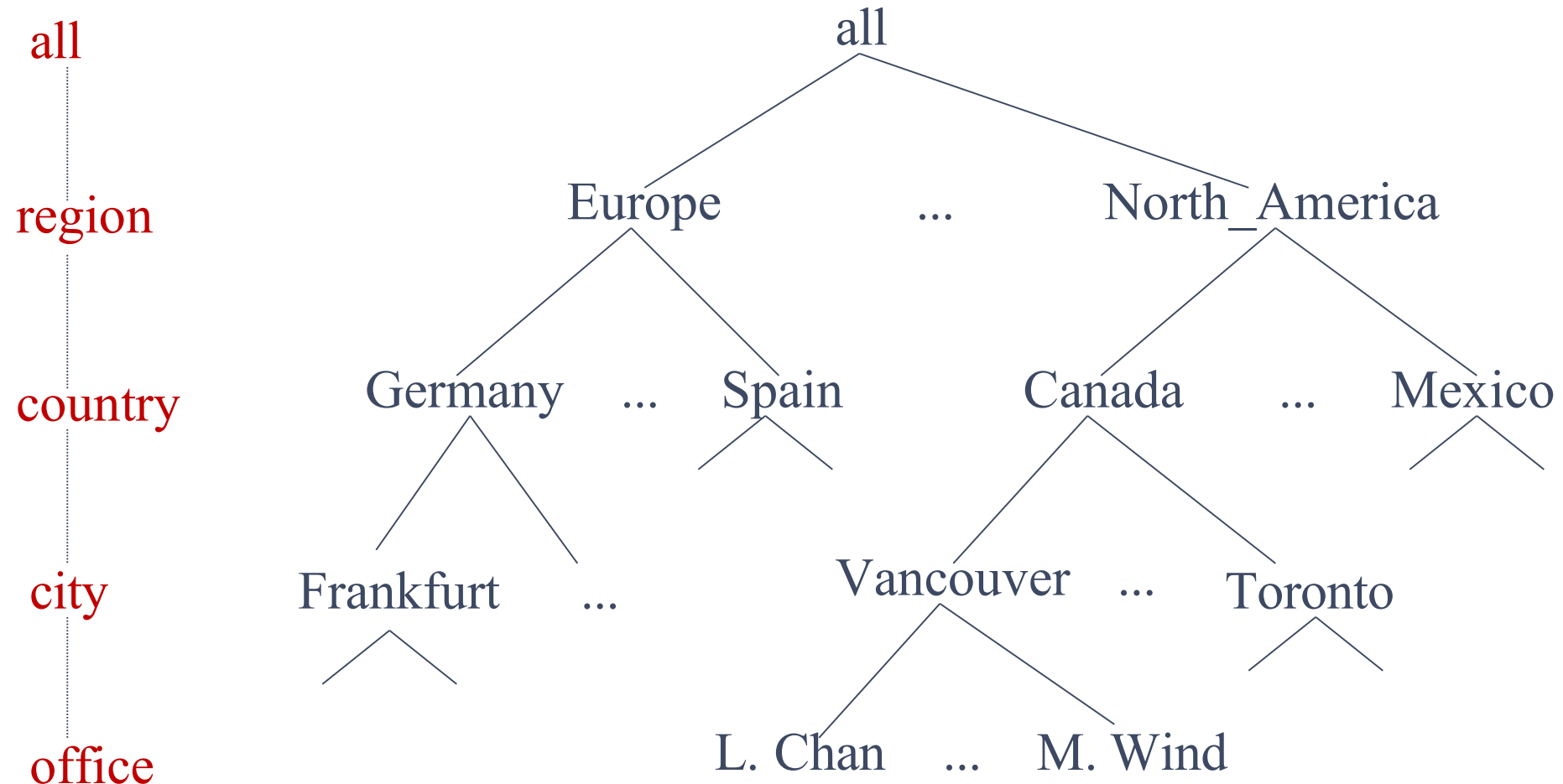
# Example of Snowflake Schema



# Example of Fact Constellation



# A Concept Hierarchy: Dimension (location)



# Concept Hierarchy (Time)

