Data Mining:

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

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

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 and OLAP

- Differences between Operational Database Systems and Data Warehouses.

- The major task of online operational database systems is to perform online transaction and query processing. These systems are called online transaction processing (OLTP) systems.

- Data warehouse systems, on the other hand, serve users or knowledge workers in the role of data analysis and decision making. These systems are known as online analytical processing (OLAP) systems.

OLTP and OLAP

- An OLTP system is customer-oriented and is used for transaction and query processing by clerks, clients, etc.
- An OLAP system is market-oriented and is used for data analysis by knowledge workers, including managers, executives, and analysts.
- An OLTP system manages current data that, typically, are too detailed to be easily used for decision making.
- An OLAP system manages large amounts of historic data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity.
- These features make the data easier to use for informed decision making.

OLTP and OLAP

- An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design. An OLAP system typically adopts either a star or a snowflake model (see Section 4.2.2) and a subject-oriented database design.
- An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data.
- In contrast, an OLAP system often spans multiple versions of a database schema, due to the evolutionary process.
- The access patterns of an OLTP system consist mainly of short, atomic transactions.
- However, accesses to OLAP systems are mostly read-only operations (because most data warehouses store historic rather than up-to-date information).

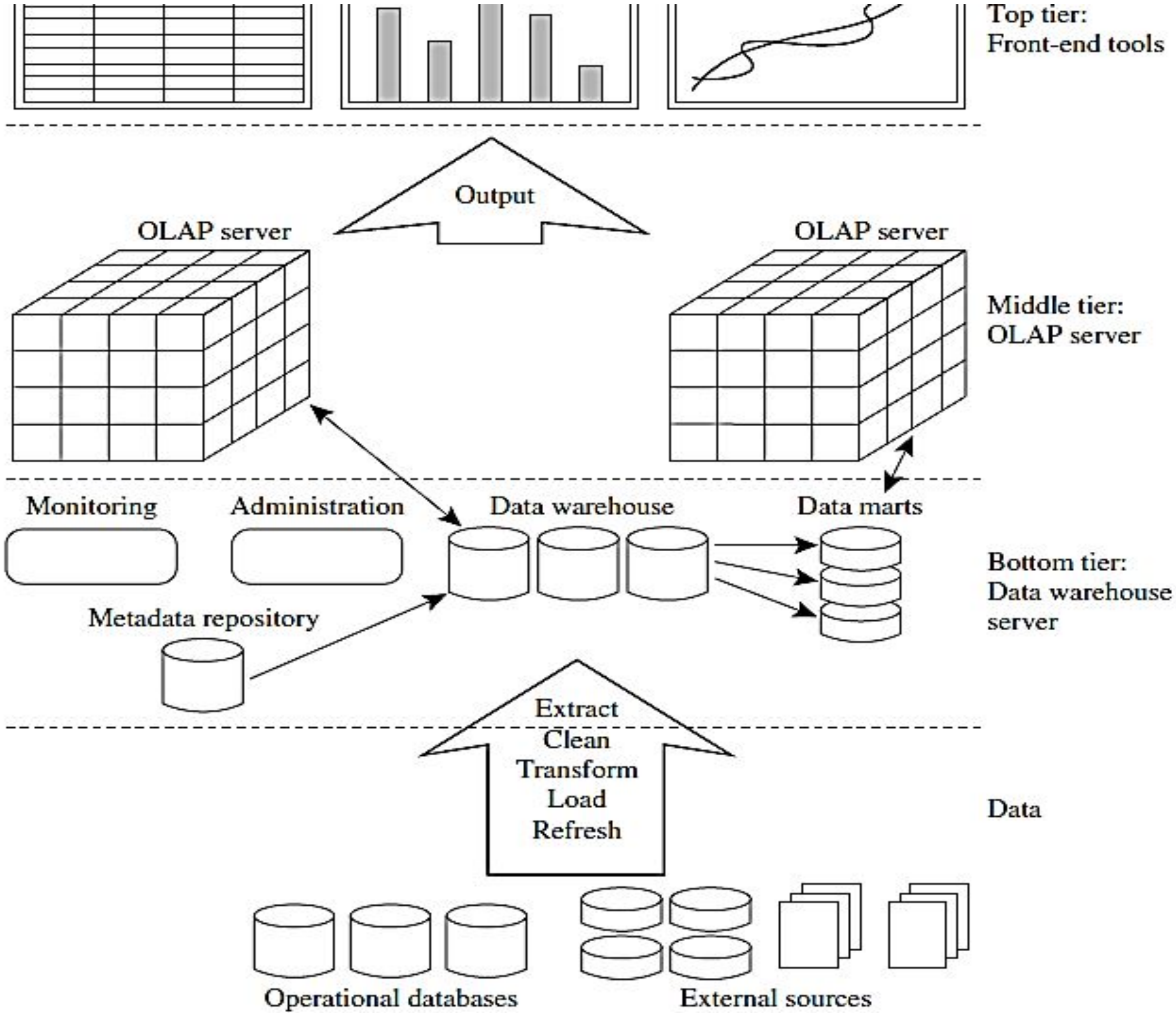
OLTP vs. OLAP

	OLTP	OLAP
characteristic	Operational	information
Orientation	transaction	analysis
users	clerk, IT professional	knowledge worker (analyst, manger)
function	day to day operations	decision support
DB design	ER based, application-oriented	Star/snowflake, 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 mostly read
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
Focus	Data in	Information out

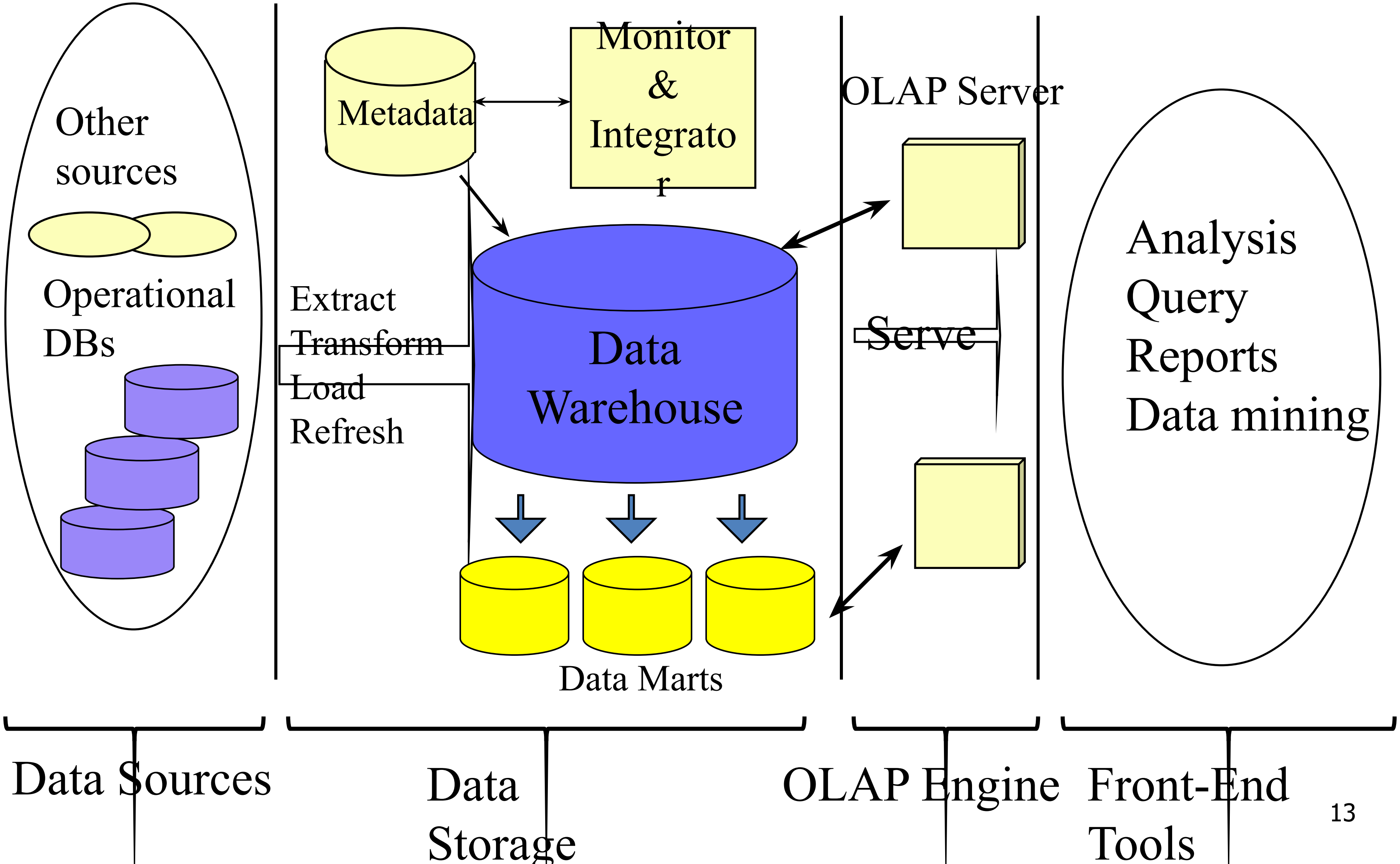
Why a Separate Data Warehouse?

- 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



Data Warehouse: A Multi-Tiered Architecture



Other sources

Operational DBs

Metadata

Monitor & Integrator

Extract
Transform
Load
Refresh

Data Warehouse

Data Marts

OLAP Server

Serve

Analysis
Query
Reports
Data mining

Data Sources

Data Storage

OLAP Engine

Front-End Tools

Three Data Warehouse Models

- Enterprise warehouse
  - collects all of the information about subjects spanning the entire organization. It typically contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond.
- Data Mart
  - a subset of corporate-wide data that is of value to a specific groups of users. For example, a marketing data mart may confine its subjects to customer, item, and sales. 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 indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

Metadata Repository

- **When used in a data warehouse, Meta data** are the data defining warehouse objects. It stores:
- 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)
- 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 Modeling: Data Cube and OLAP

- Data Cube: A Multidimensional Data Model

Dimension Table

- All-Electronics may create a sales data warehouse in order to keep records of the store’s sales with respect to the dimensions time, item, branch, and location.
- Each dimension may have a table associated with it, called a dimension table

Data Warehouse Modeling: Data Cube and OLAP

- Data Cube: A Multidimensional Data Model

Fact Table

- A multidimensional data model is typically organized around a central theme, such as sales. This theme is represented by a fact table. Facts are numeric measures.
- The fact table contains the names of the facts, or measures, as well as keys to each of the related dimension tables.

Data Warehouse Modeling: Data Cube and OLAP

- Data Cube: A Multidimensional Data Model

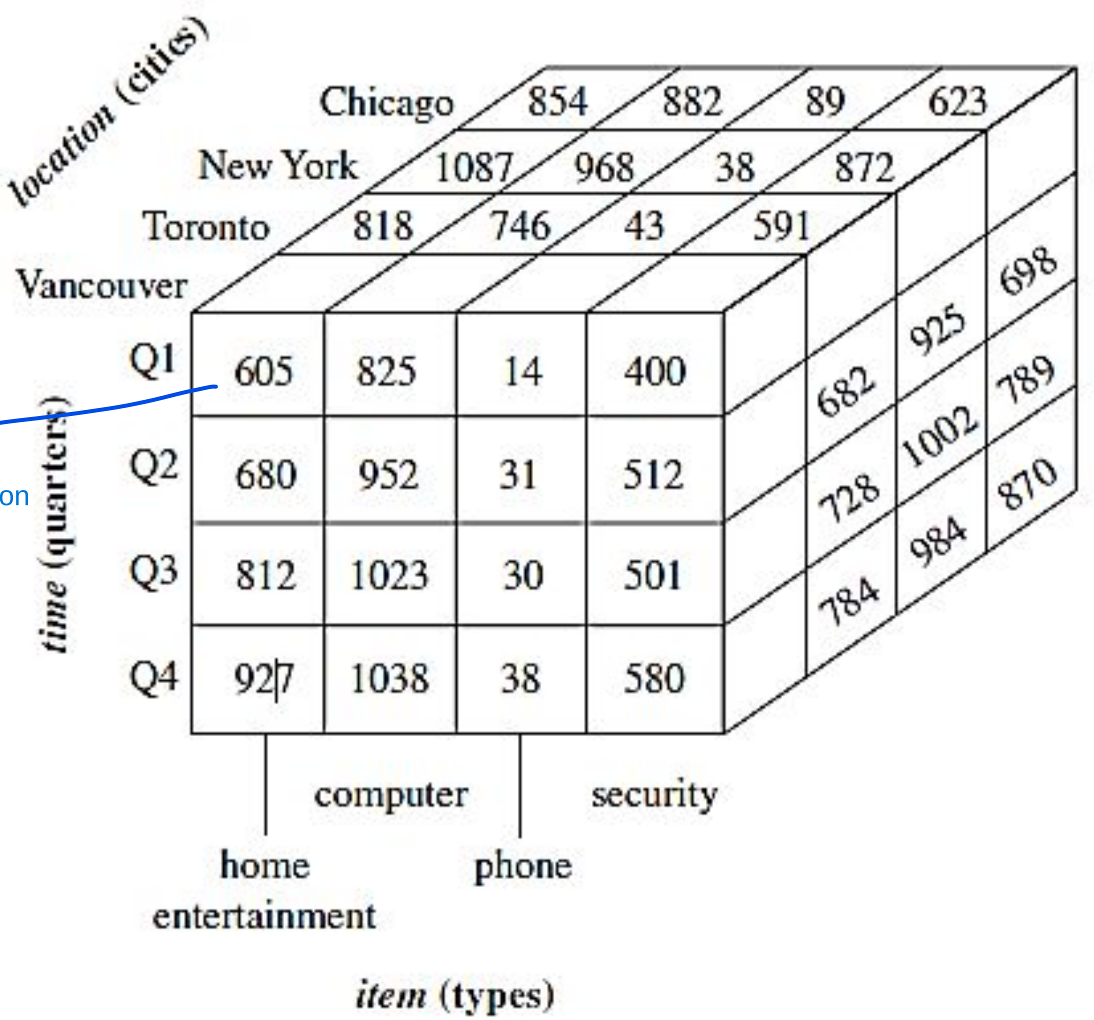
Fact Table

- Facts are numeric measures. Think of them as the quantities by which we want to analyze relationships between dimensions.
- Examples of facts for a sales data warehouse include dollars sold (sales amount in dollars), units sold (number of units sold), and amount budgeted.

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.

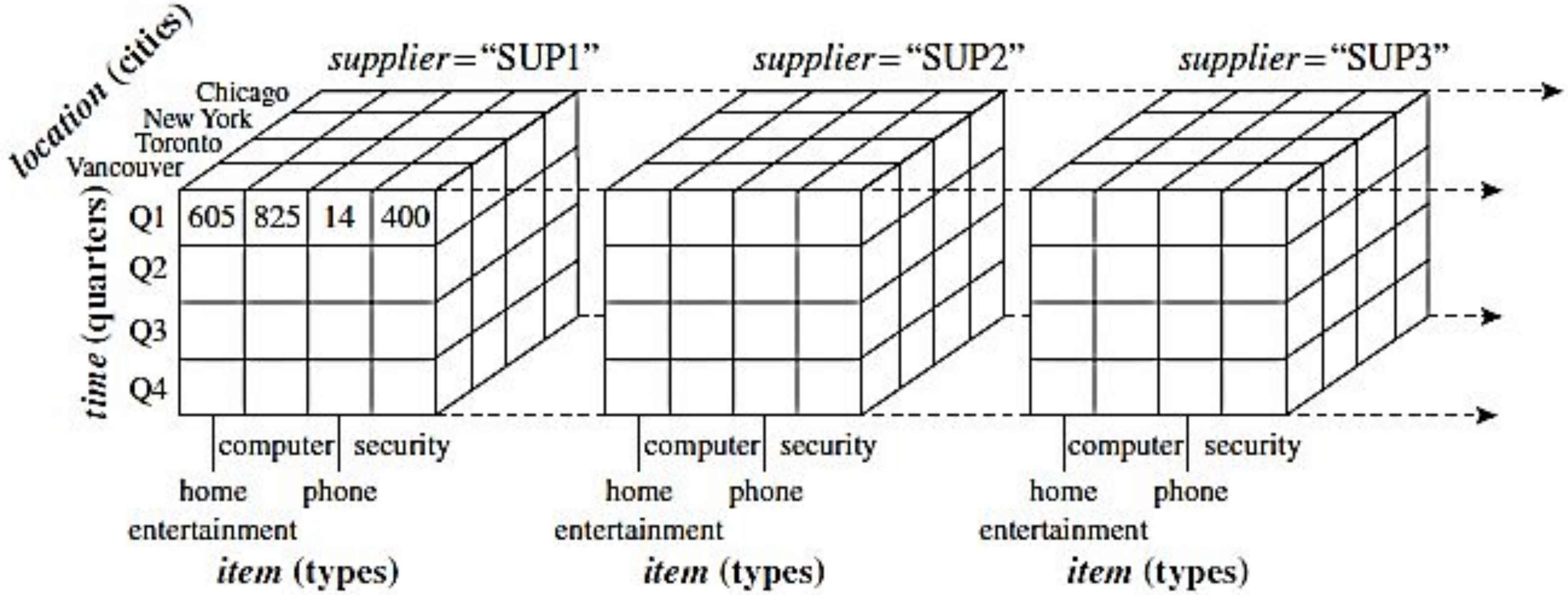
3-D Data Cube



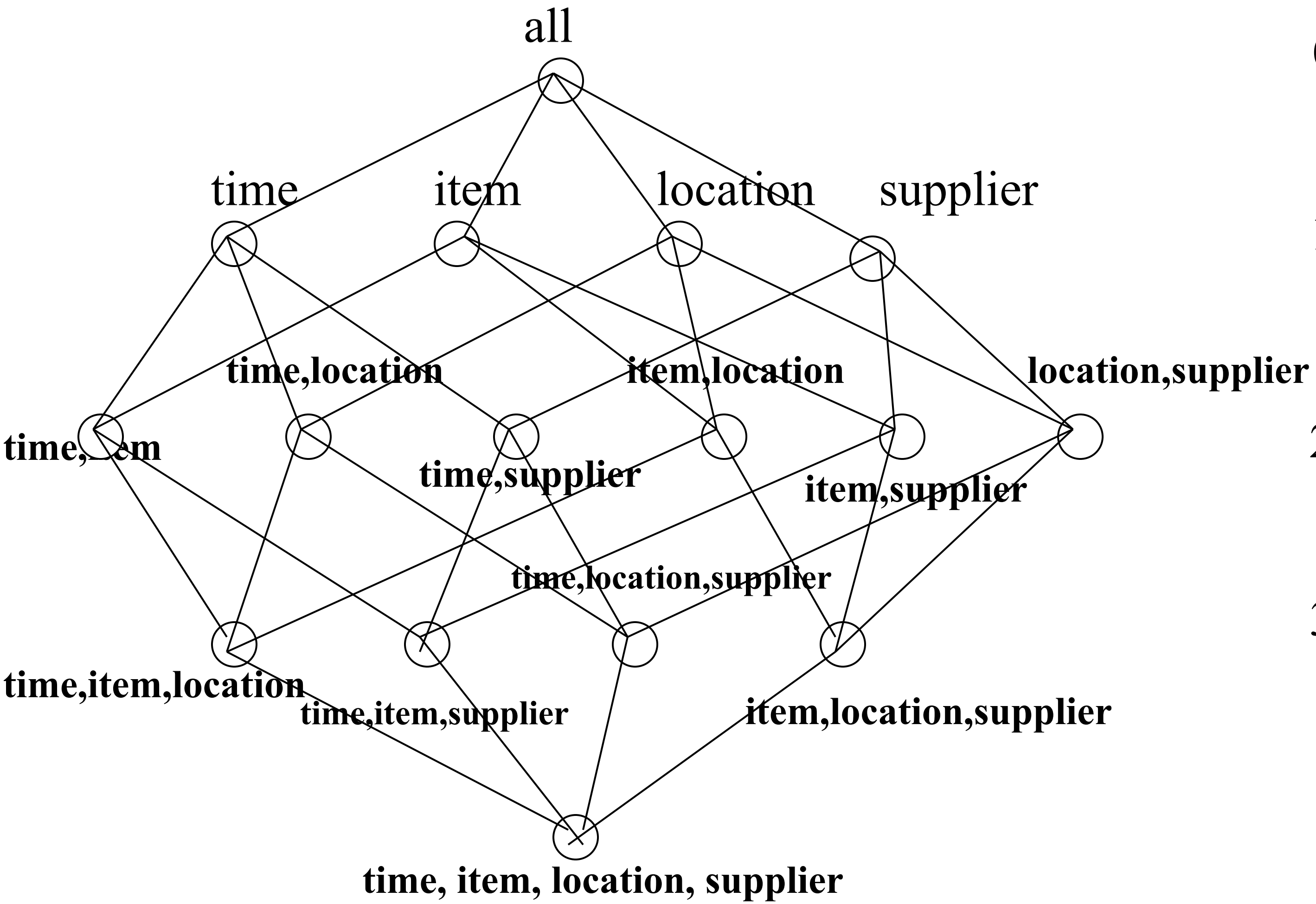
	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

sales of what item what location
at which time

4-D Data Cube


location (cities)
Chicago
New York
Toronto
Vancouver
supplier="SUP1"
supplier="SUP2"
supplier="SUP3"
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
computer
security
home entertainment
phone
item (types)

Cube: A Lattice of Cuboids



0-D (*apex*) cuboid

1-D cuboids

2-D cuboids

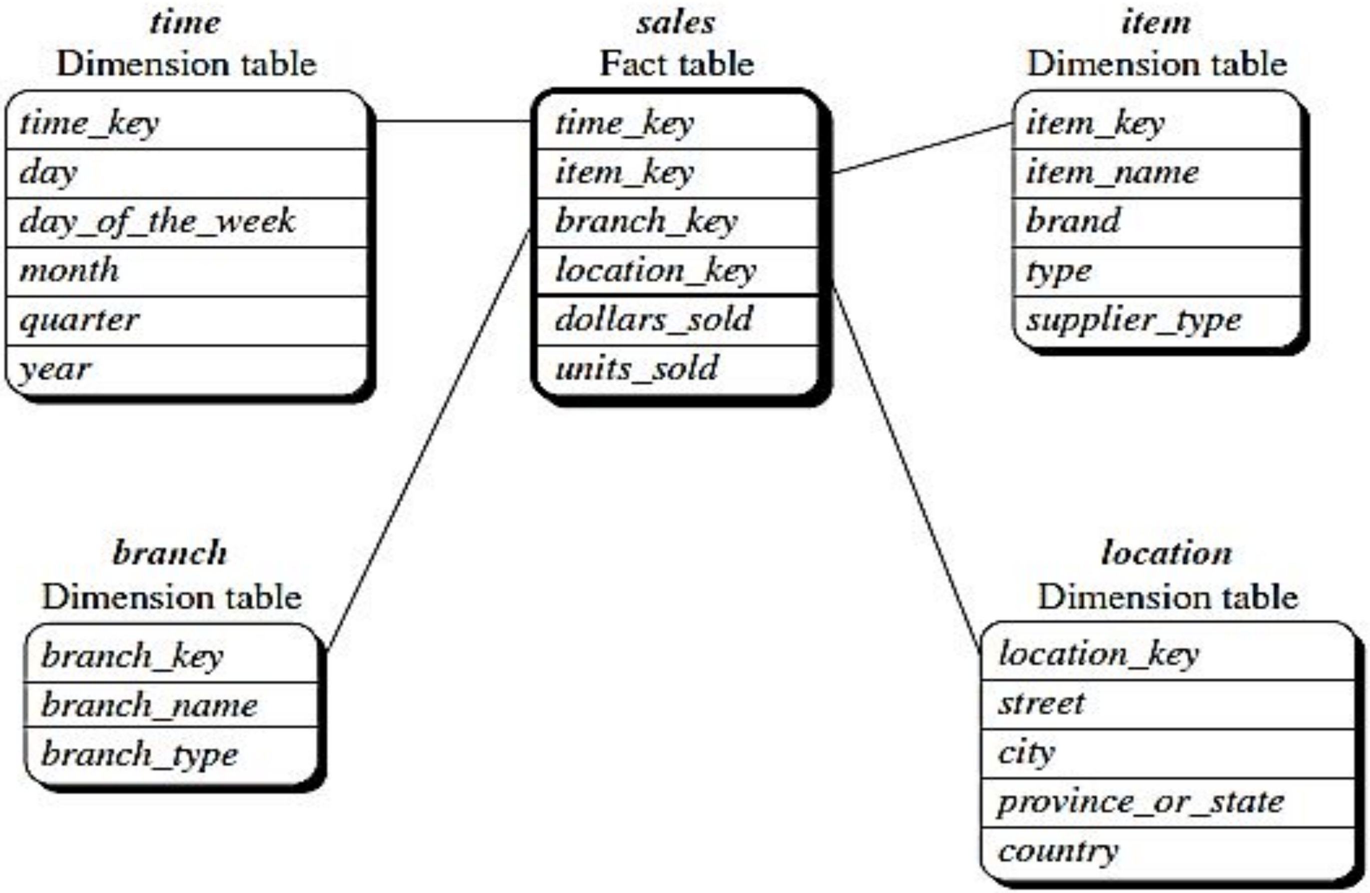
3-D cuboids

4-D (*base*) cuboid

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



time
Dimension table

time_key
day
day_of_the_week
month
quarter
year

sales
Fact table

time_key
item_key
branch_key
location_key
dollars_sold
units_sold

item
Dimension table

item_key
item_name
brand
type
supplier_type

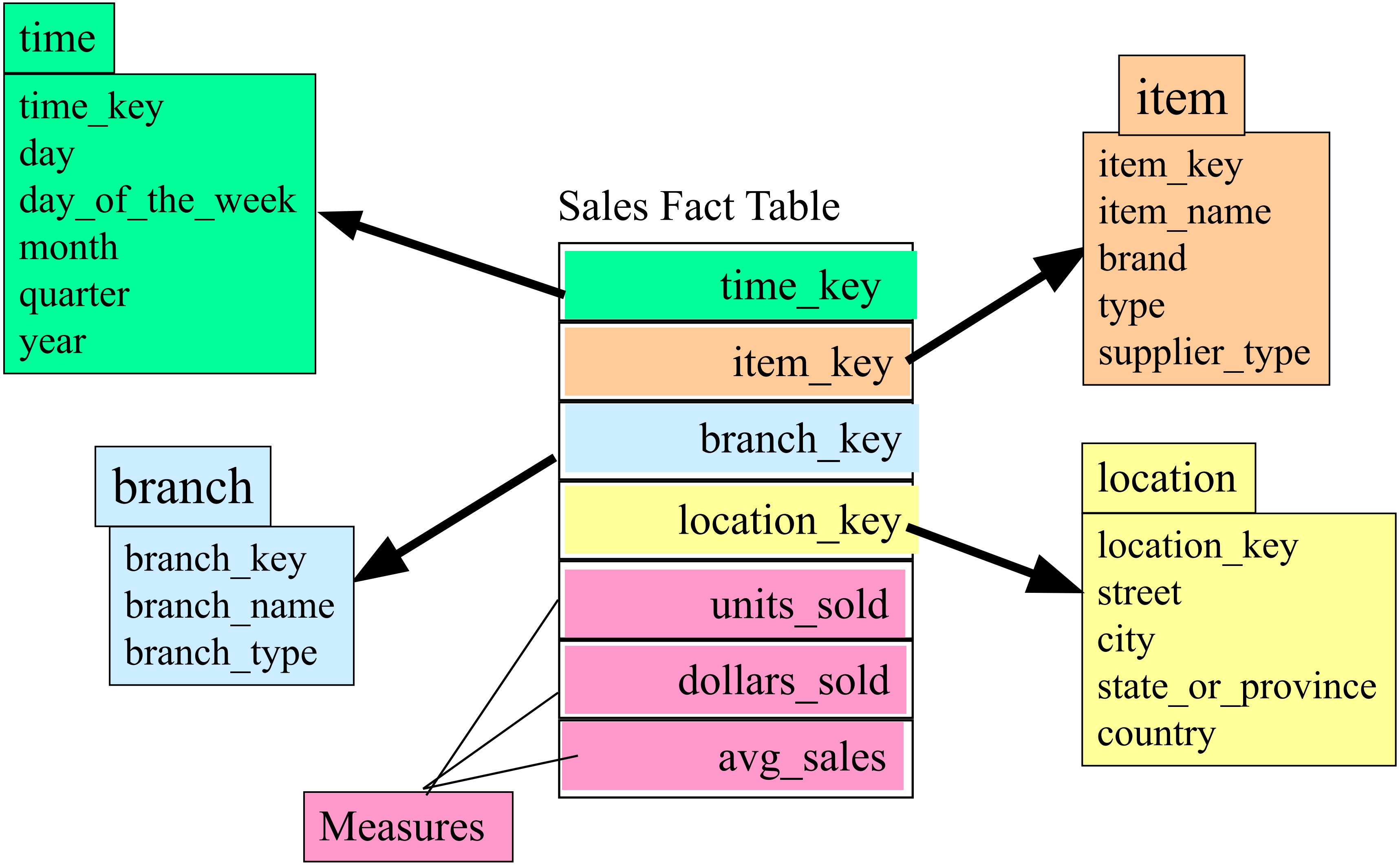
branch
Dimension table

branch_key
branch_name
branch_type

location
Dimension table

location_key
street
city
province_or_state
country

Example of Star Schema



time

time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table

time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

item

item_key
item_name
brand
type
supplier_type

branch

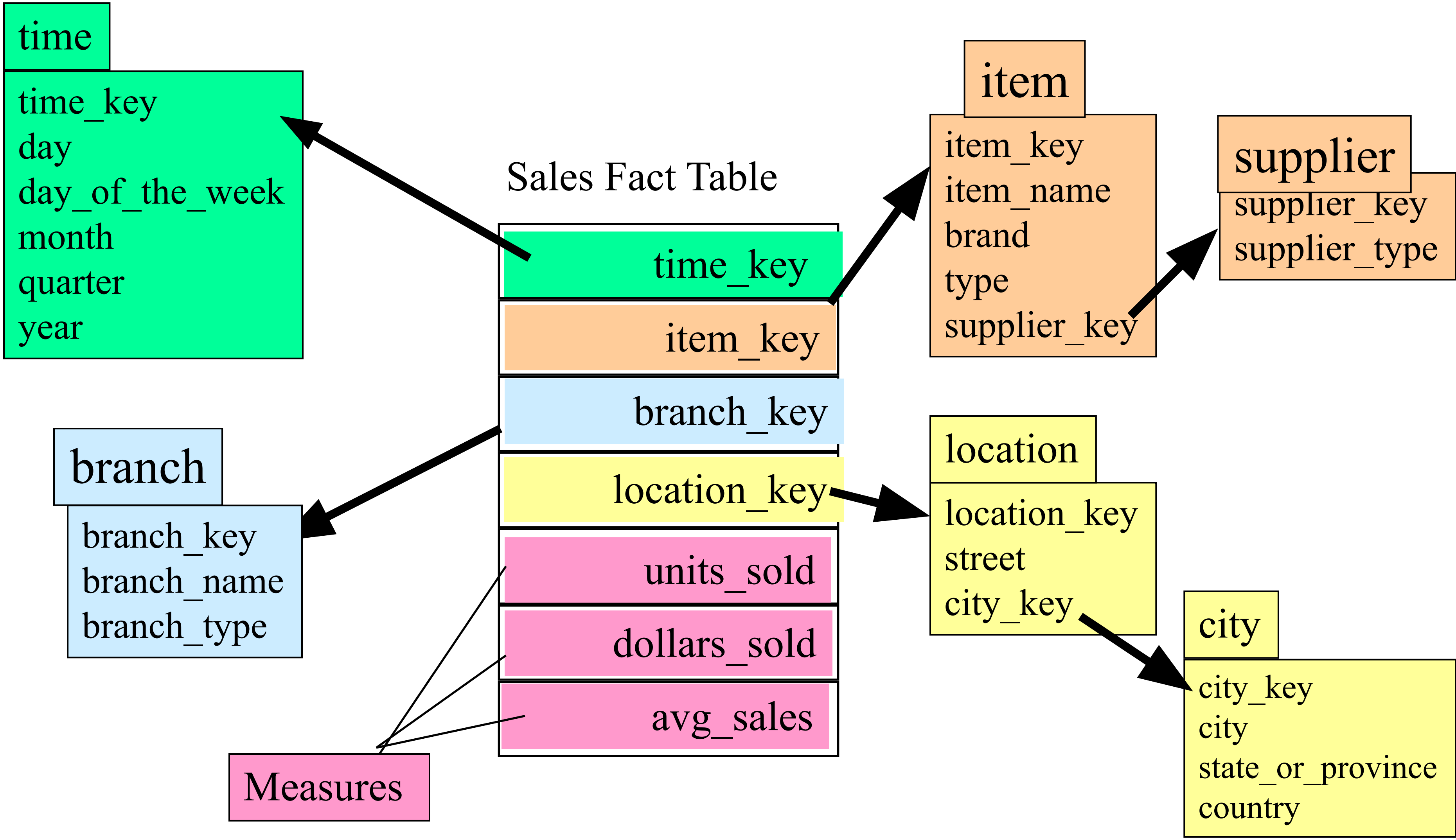
branch_key
branch_name
branch_type

location

location_key
street
city
state_or_province
country

Measures

Example of Snowflake Schema



time
time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

Measures

item
item_key
item_name
brand
type
supplier_key

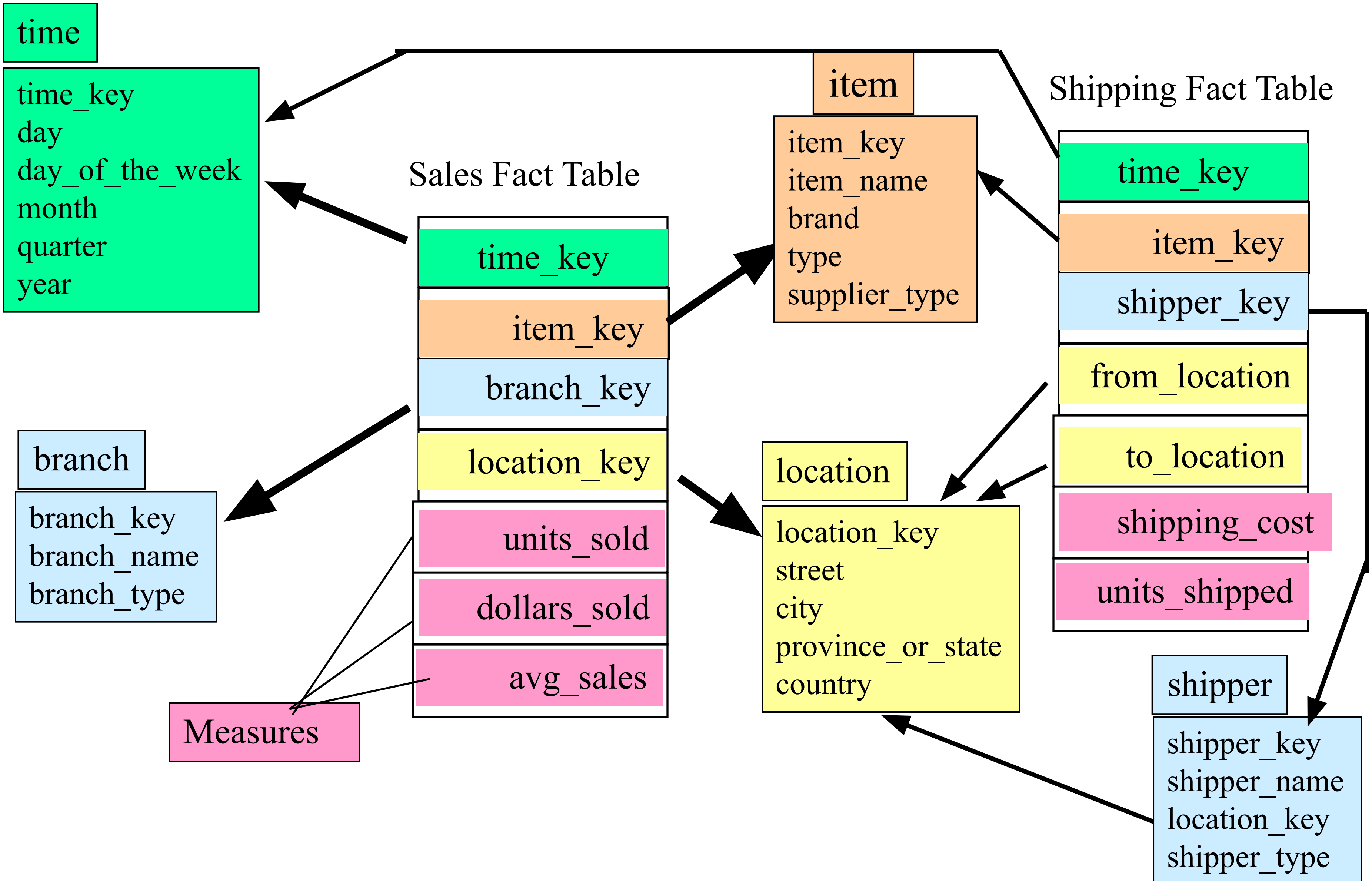
supplier
supplier_key
supplier_type

branch
branch_key
branch_name
branch_type

location
location_key
street
city_key

city
city_key
city
state_or_province
country

Example of Fact Constellation



time

time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table

time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

item

item_key
item_name
brand
type
supplier_type

Shipping Fact Table

time_key
item_key
shipper_key
from_location
to_location
shipping_cost
units_shipped

branch

branch_key
branch_name
branch_type

location

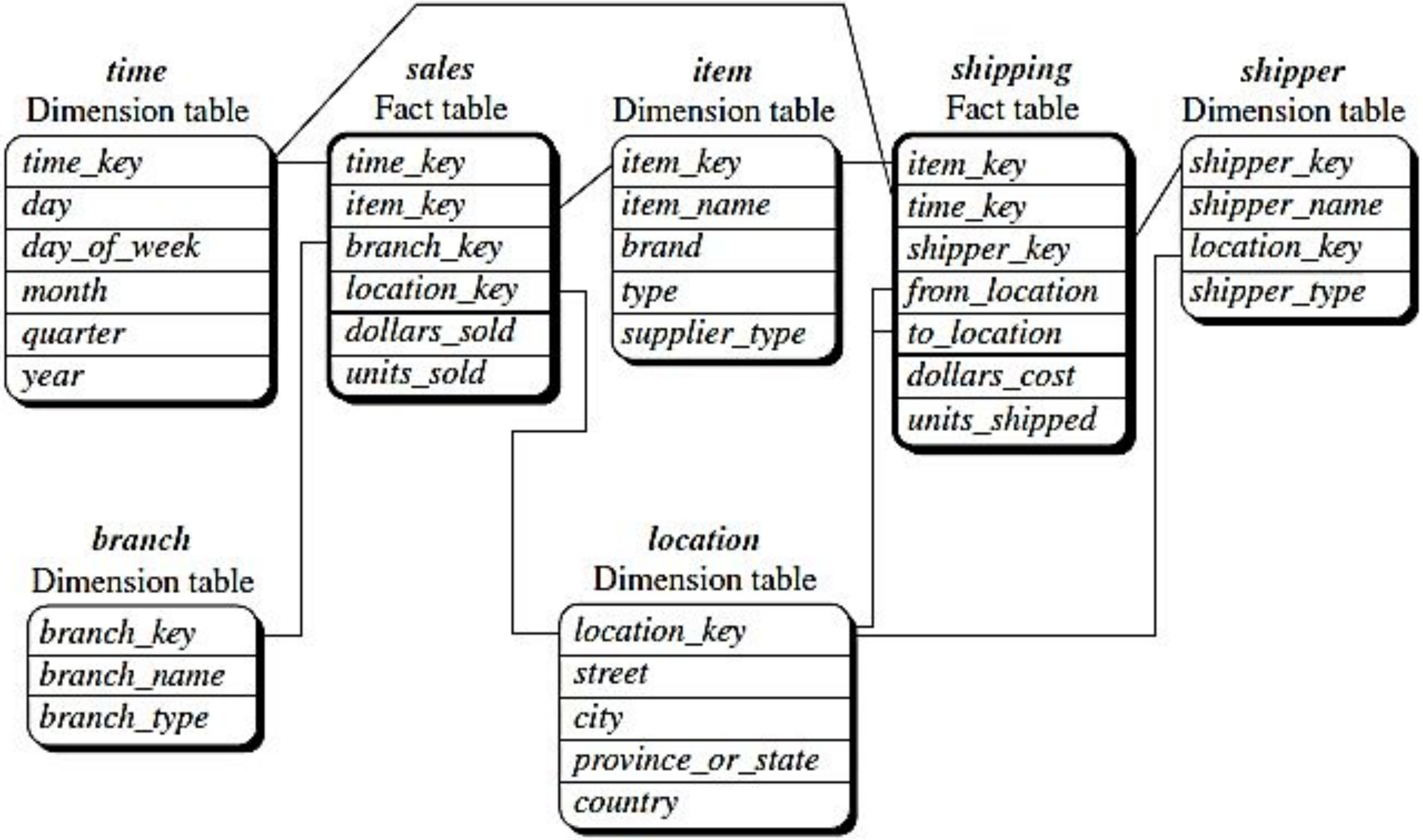
location_key
street
city
province_or_state
country

shipper

shipper_key
shipper_name
location_key
shipper_type

Measures

Example of Fact Constellation



time
Dimension table

time
time_key
day
day_of_week
month
quarter
year

sales
Fact table

sales
time_key
item_key
branch_key
location_key
dollars_sold
units_sold

item
Dimension table

item
item_key
item_name
brand
type
supplier_type

shipping
Fact table

shipping
item_key
time_key
shipper_key
from_location
to_location
dollars_cost
units_shipped

shipper
Dimension table

shipper
shipper_key
shipper_name
location_key
shipper_type

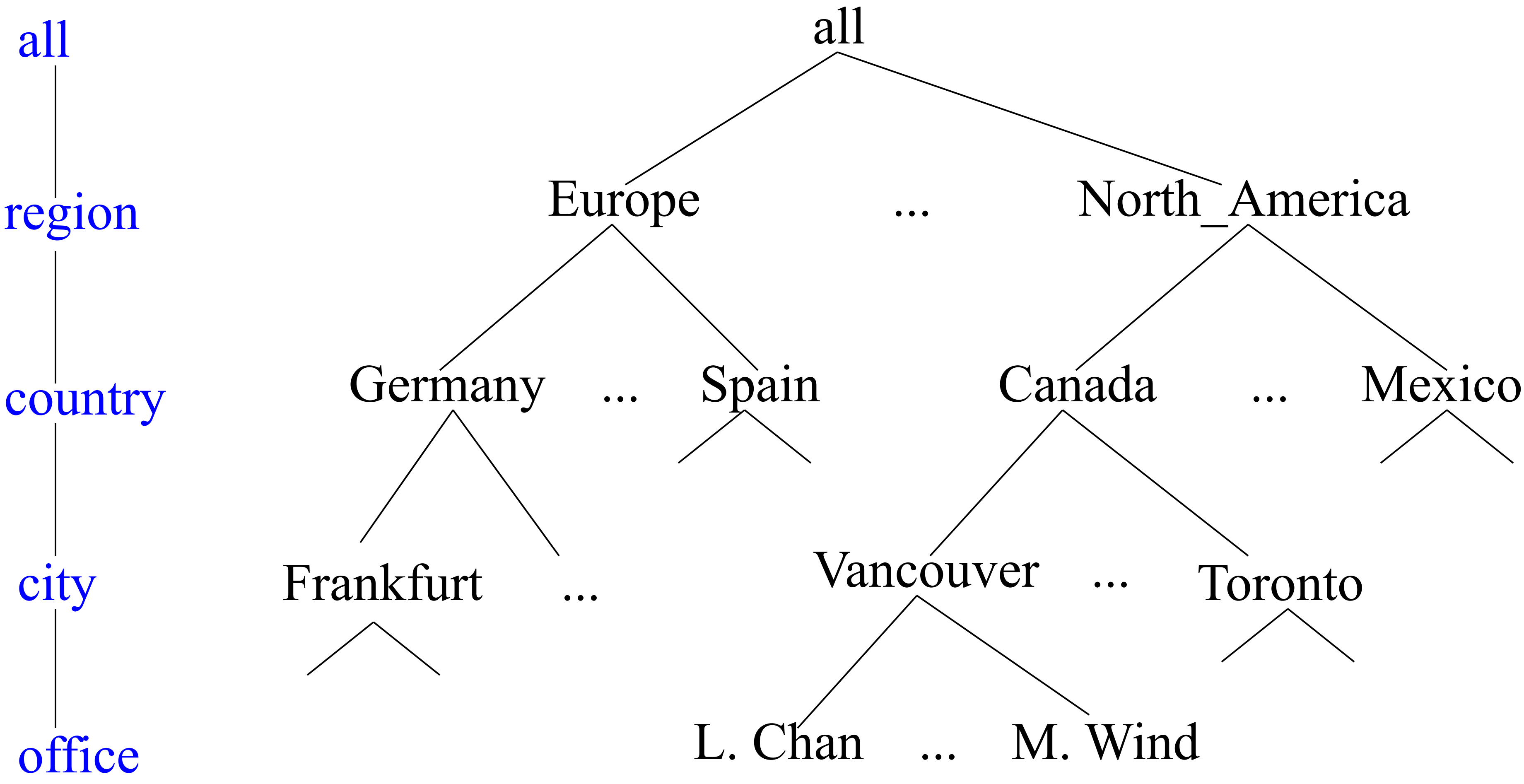
branch
Dimension table

branch
branch_key
branch_name
branch_type

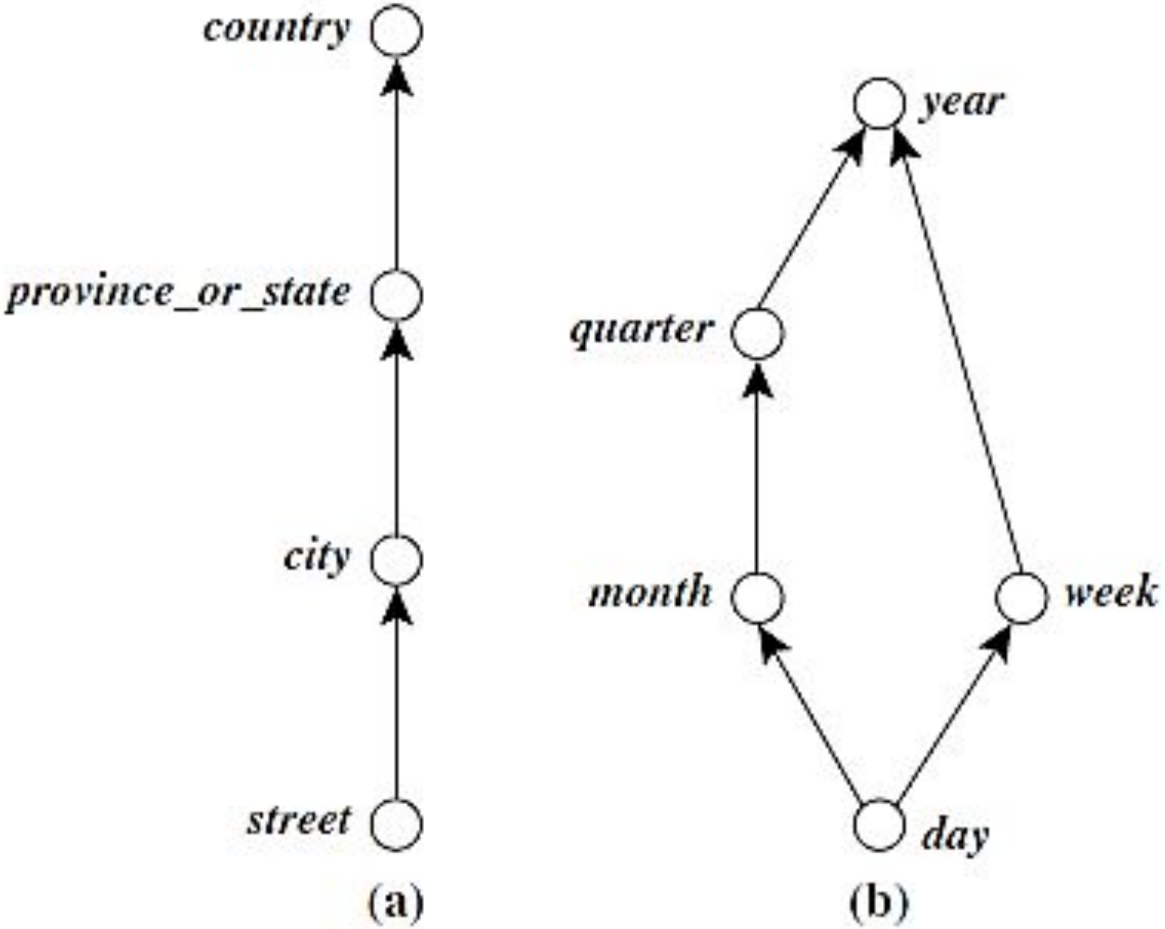
location
Dimension table

location
location_key
street
city
province_or_state
country

A Concept Hierarchy: Dimension (location)



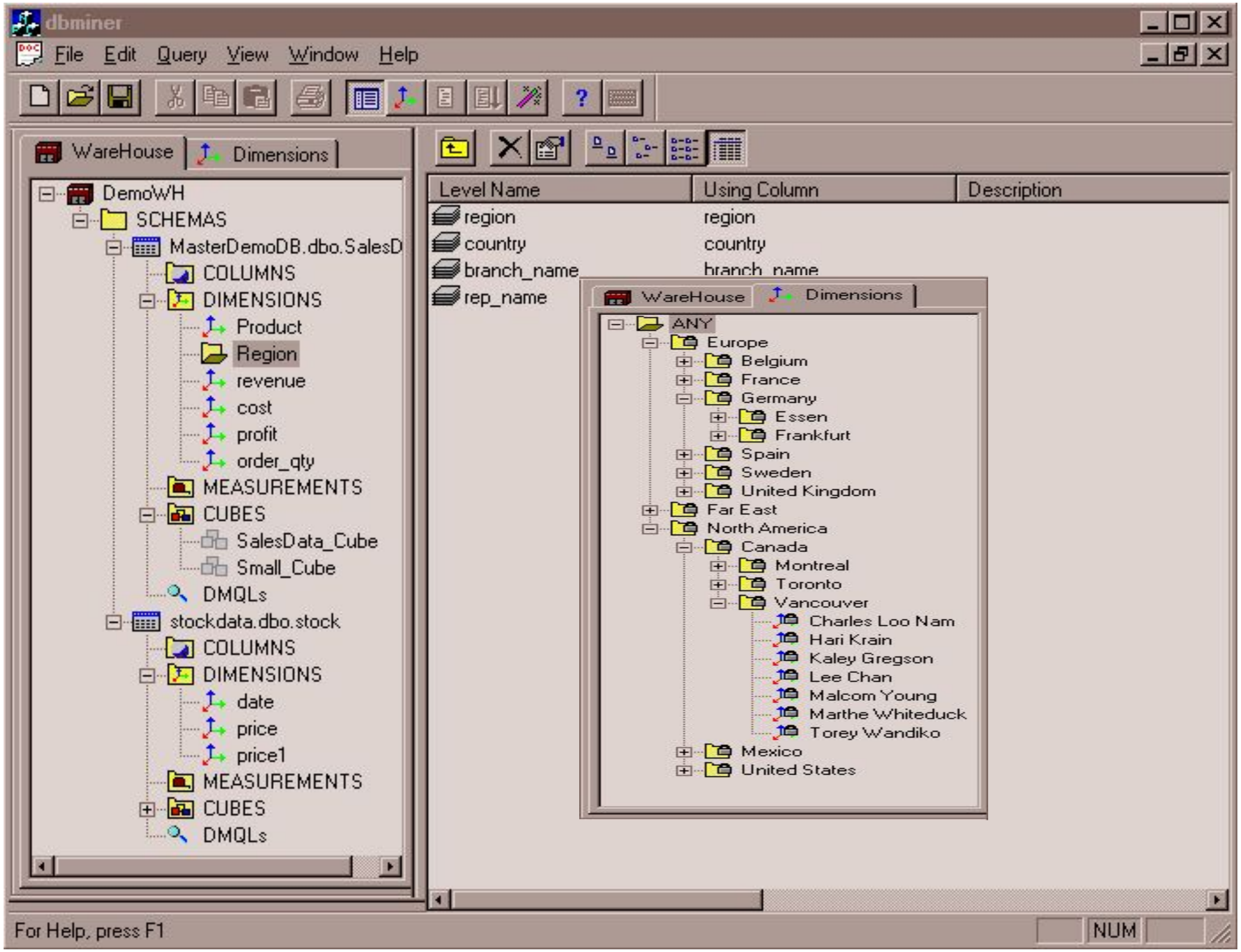
A Concept Hierarchy: Dimension (location)



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
    - E.g., avg(), min_N(), standard_deviation()
- Holistic: if there is no constant bound on the storage size needed to describe a sub-aggregate.
    - E.g., median(), mode(), rank()

View of Warehouses and Hierarchies



Level Name	Using Column	Description
region	region	
country	country	
branch_name	branch_name	
rep_name		

Specification of hierarchies

- Schema hierarchy

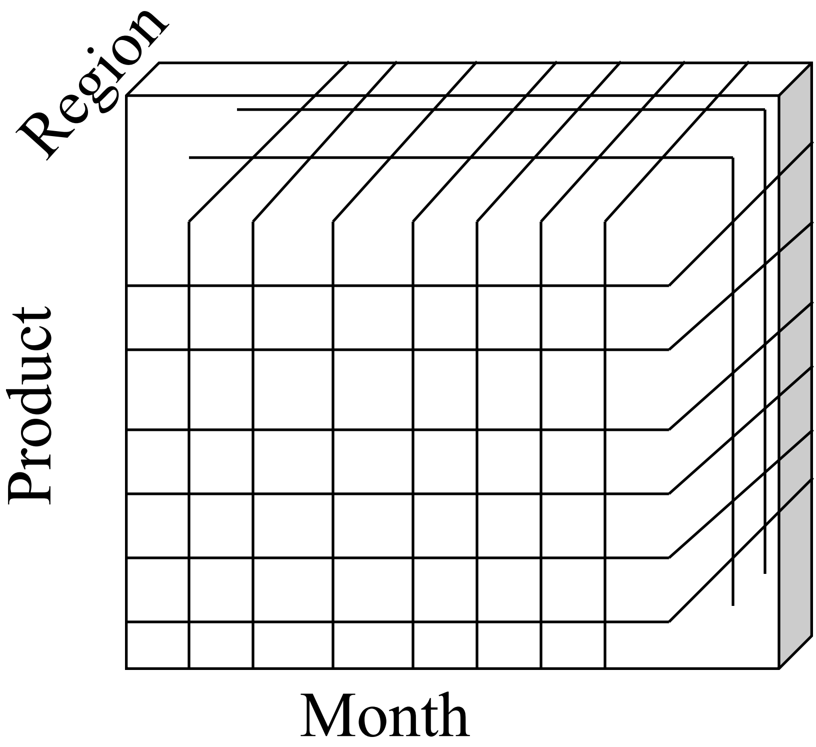
  day < {month < quarter; week} < year

- Set_grouping hierarchy

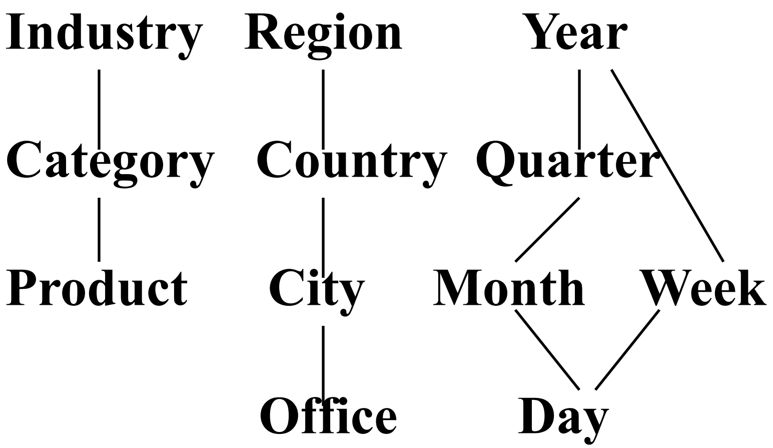
  {1..10} < inexpensive

Multidimensional Data

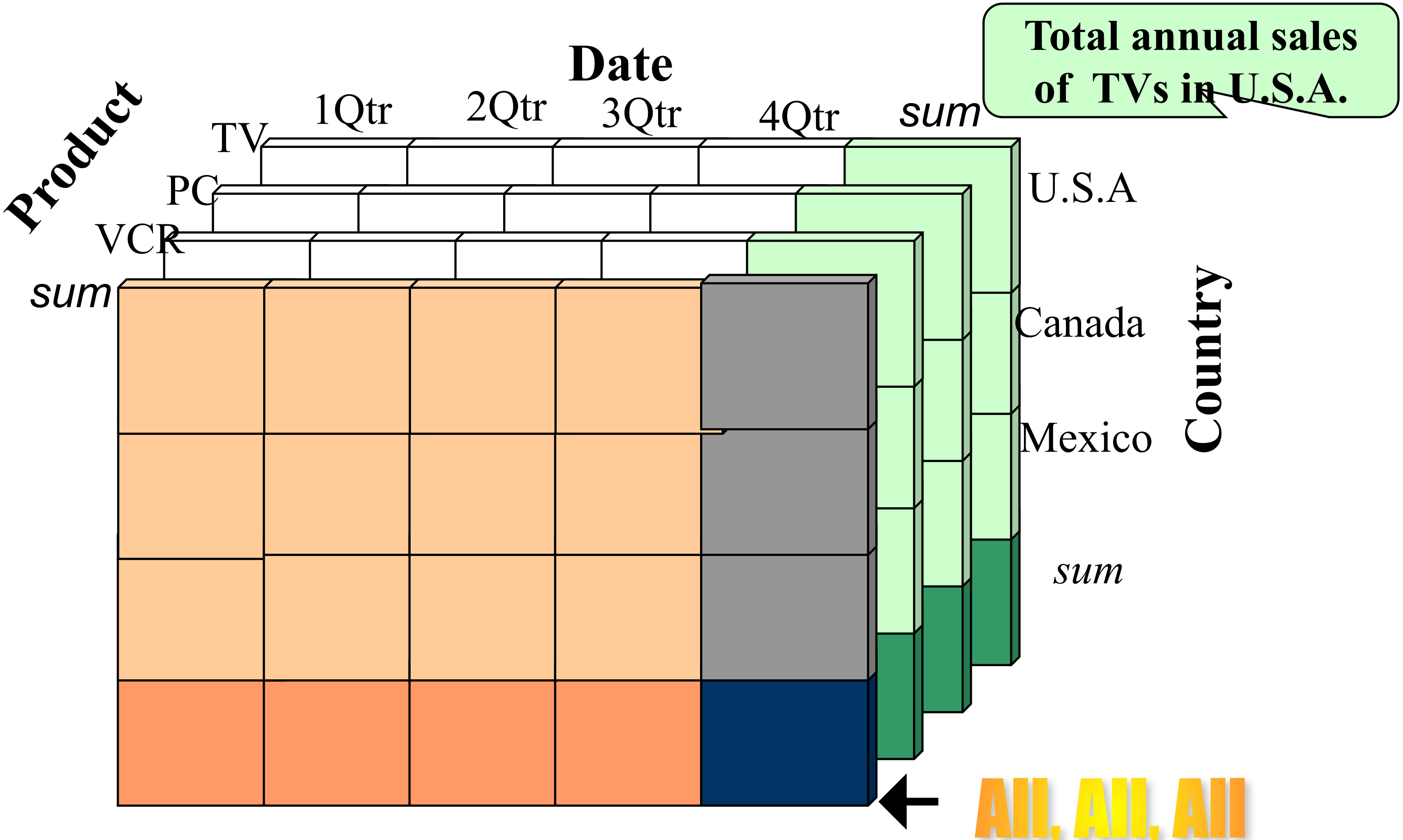
- Sales volume as a function of product, month, and region



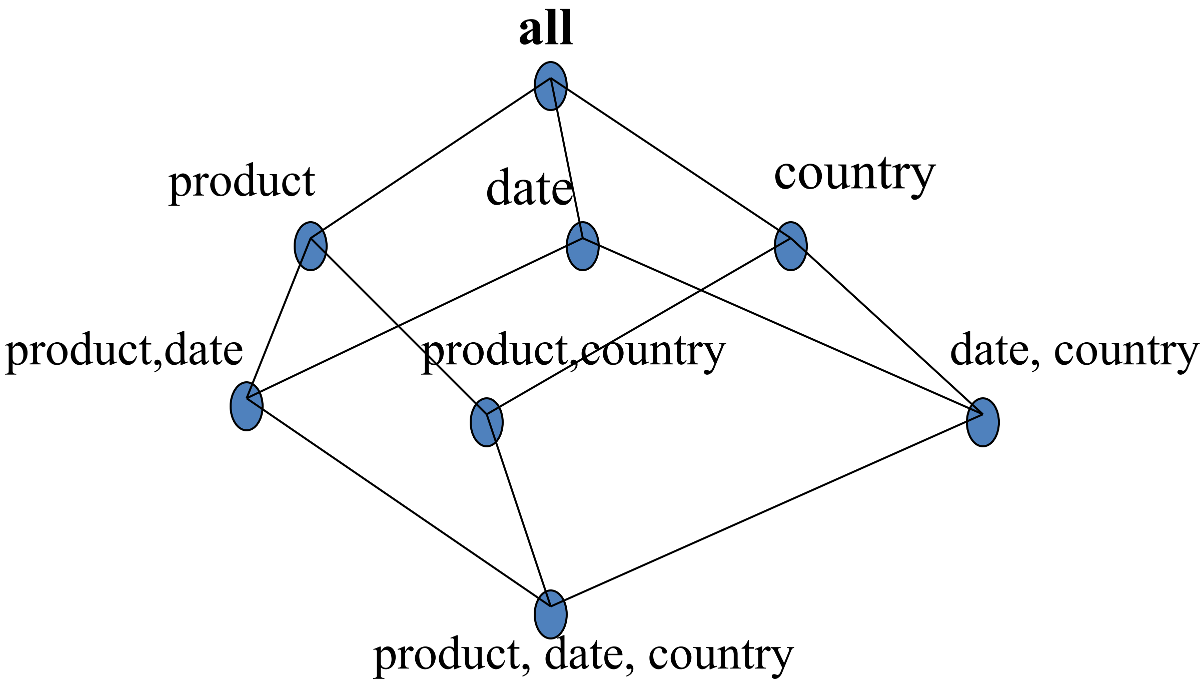
Dimensions: ***Product, Location, Time***
Hierarchical summarization paths



A Sample Data Cube



Cuboids Corresponding to the Cube



0-D (*apex*) cuboid

1-D cuboids

2-D cuboids

3-D (*base*) cuboid