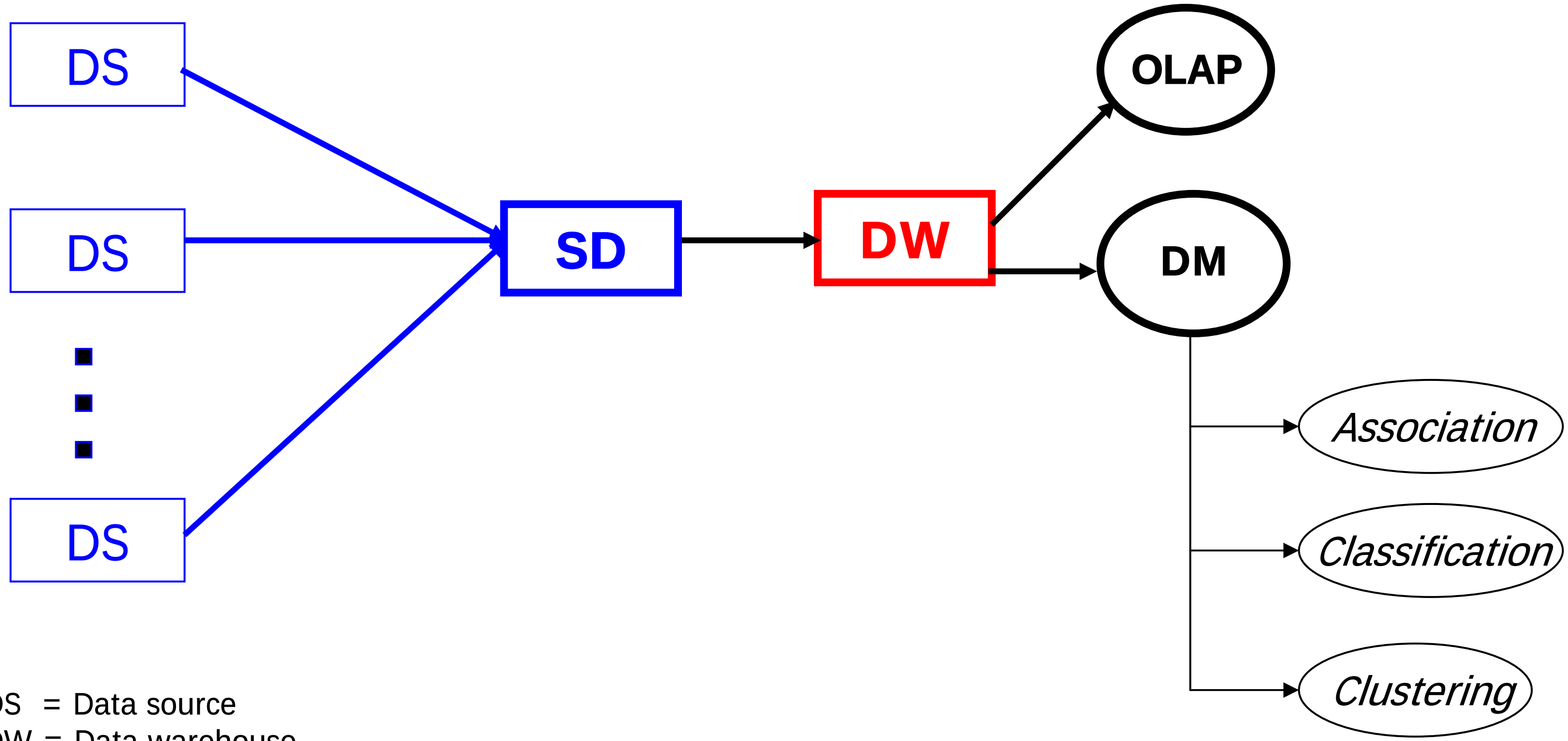
# Data Warehousing and OLAP Technology

Chapter 3

## - The Course




DS = Data source
DW = Data warehouse
DM = Data Mining
SD = Staging Database

# Chapter Outline

- What is a data warehouse?
- How to construct a Data Warehouse
  - What is the Data Model used in data warehouse?
  - Data warehouse architecture
  - Data warehouse implementation

# - 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

- Data warehousing:
  - The process of constructing and using data warehouses

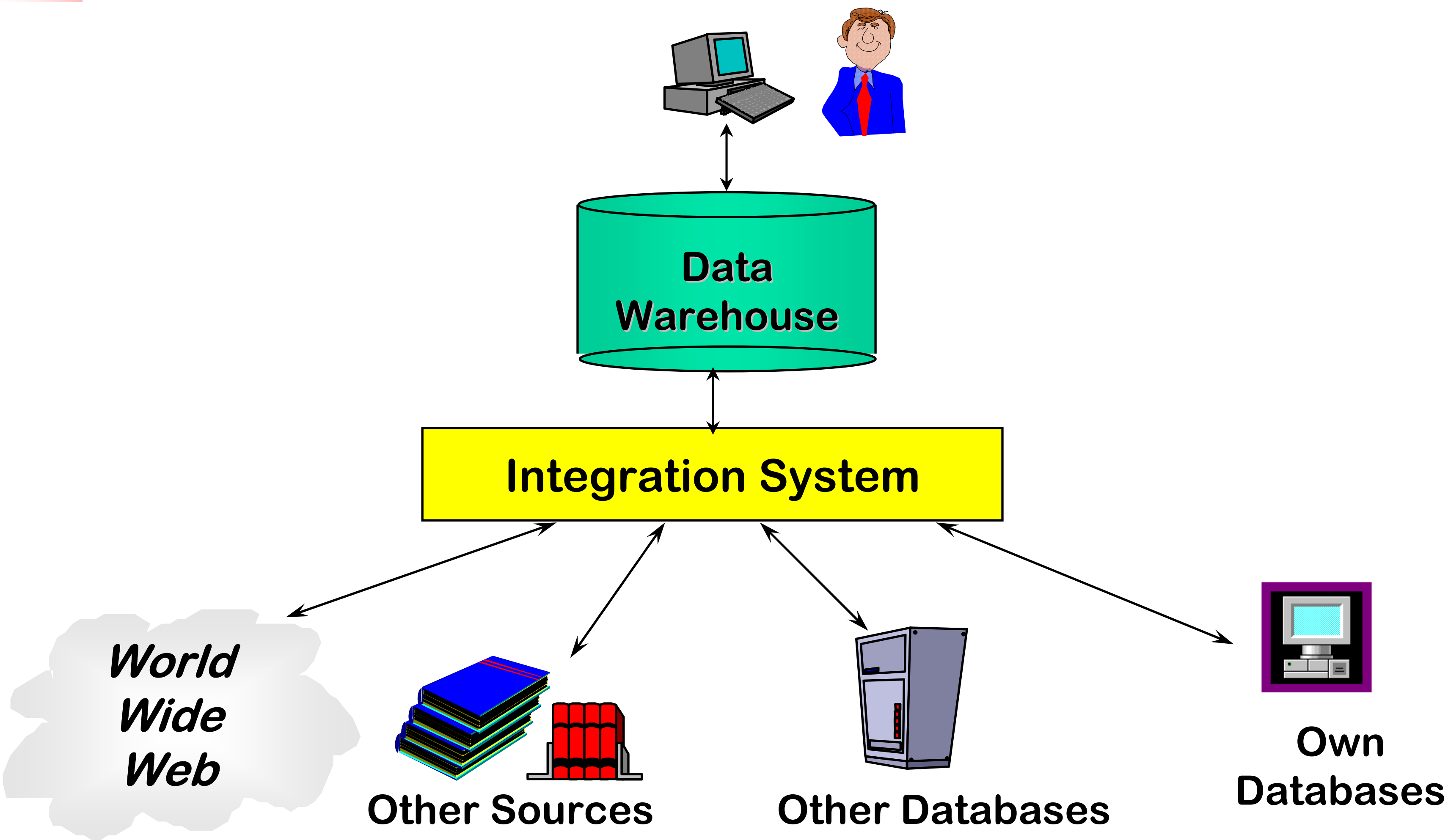
# -- Data Warehouse—Subject-Oriented

- 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 - Integrated



# -- 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”

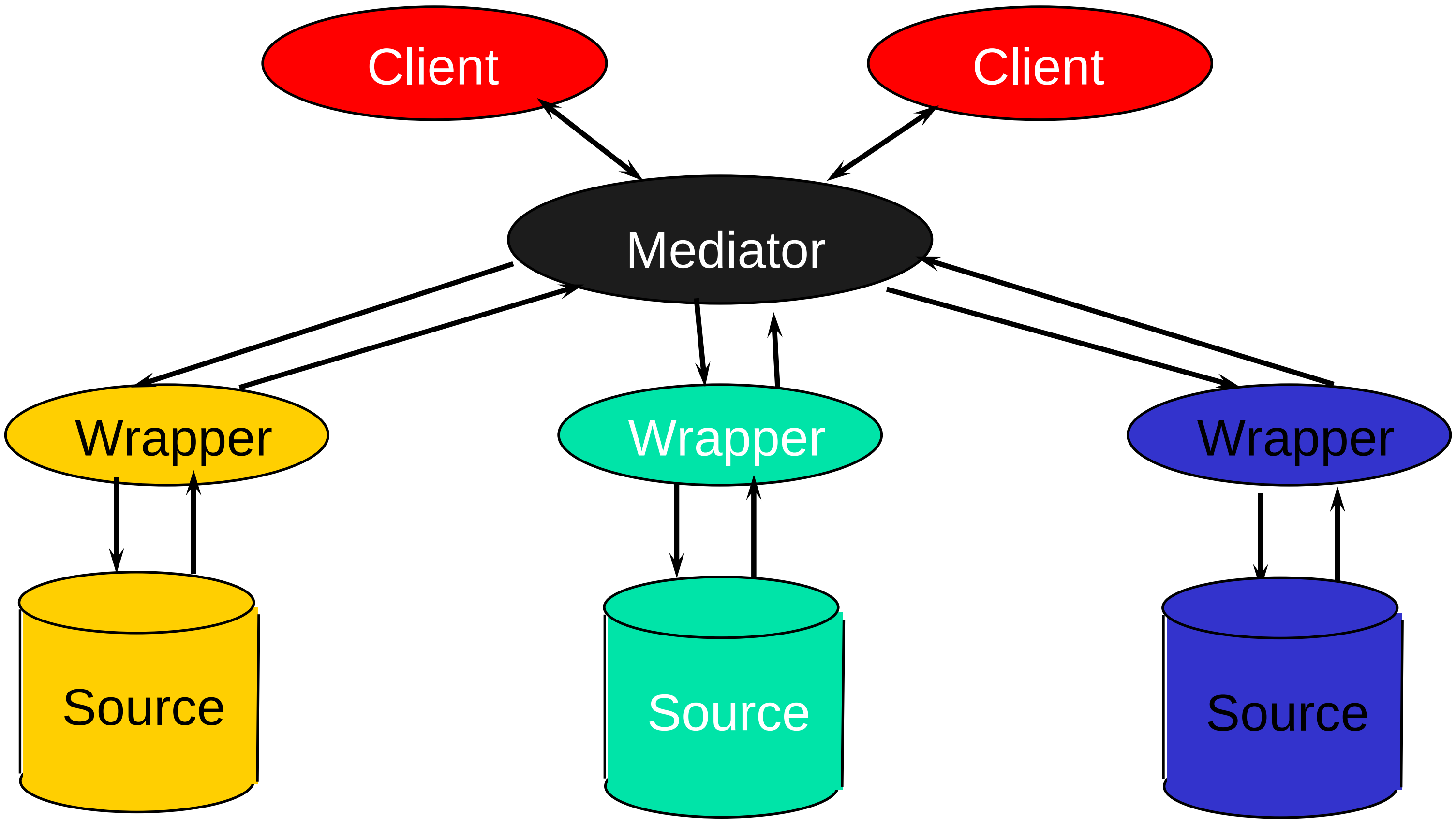
# -- Data Warehouse—Nonvolatile

- 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*

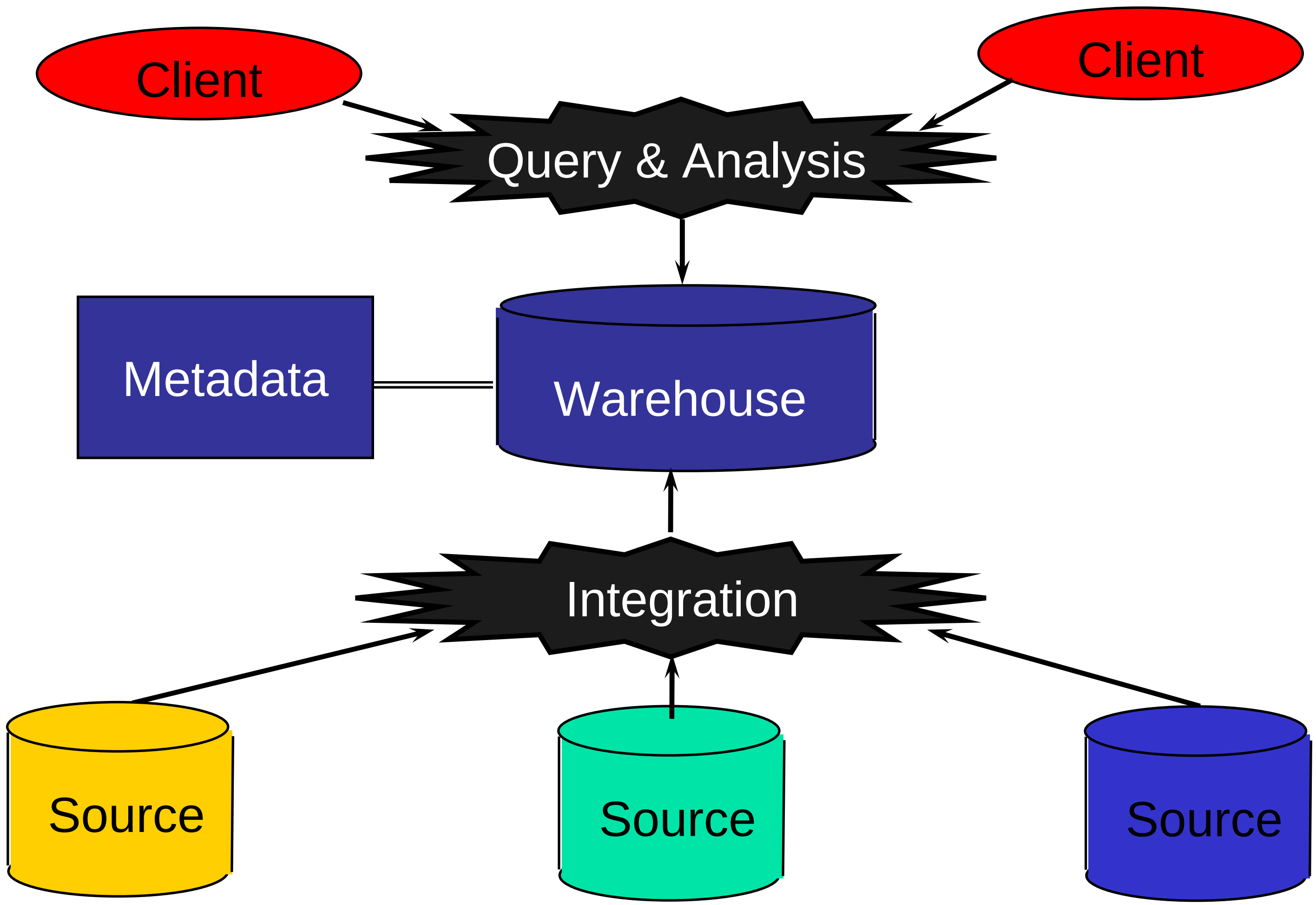
# -- Data Warehouse vs. Heterogeneous DBMS ...

- 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

# Query-Driven Approach



# The Warehousing Approach



# -- Data Warehouse vs. Operational DBMS

- OLTP (on-line transaction processing)
  - Major task of traditional relational DBMS
  - Day-to-day operations: 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
  - Database design: ER + application vs. star + subject
  - View: current, local vs. evolutionary, integrated
  - Access patterns: update vs. read-only but complex queries

# -- 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<br>detailed, flat relational<br>isolated | historical,<br>summarized, multidimensional<br>integrated, consolidated |
| **usage** | repetitive | ad-hoc |
| **access** | read/write<br>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 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

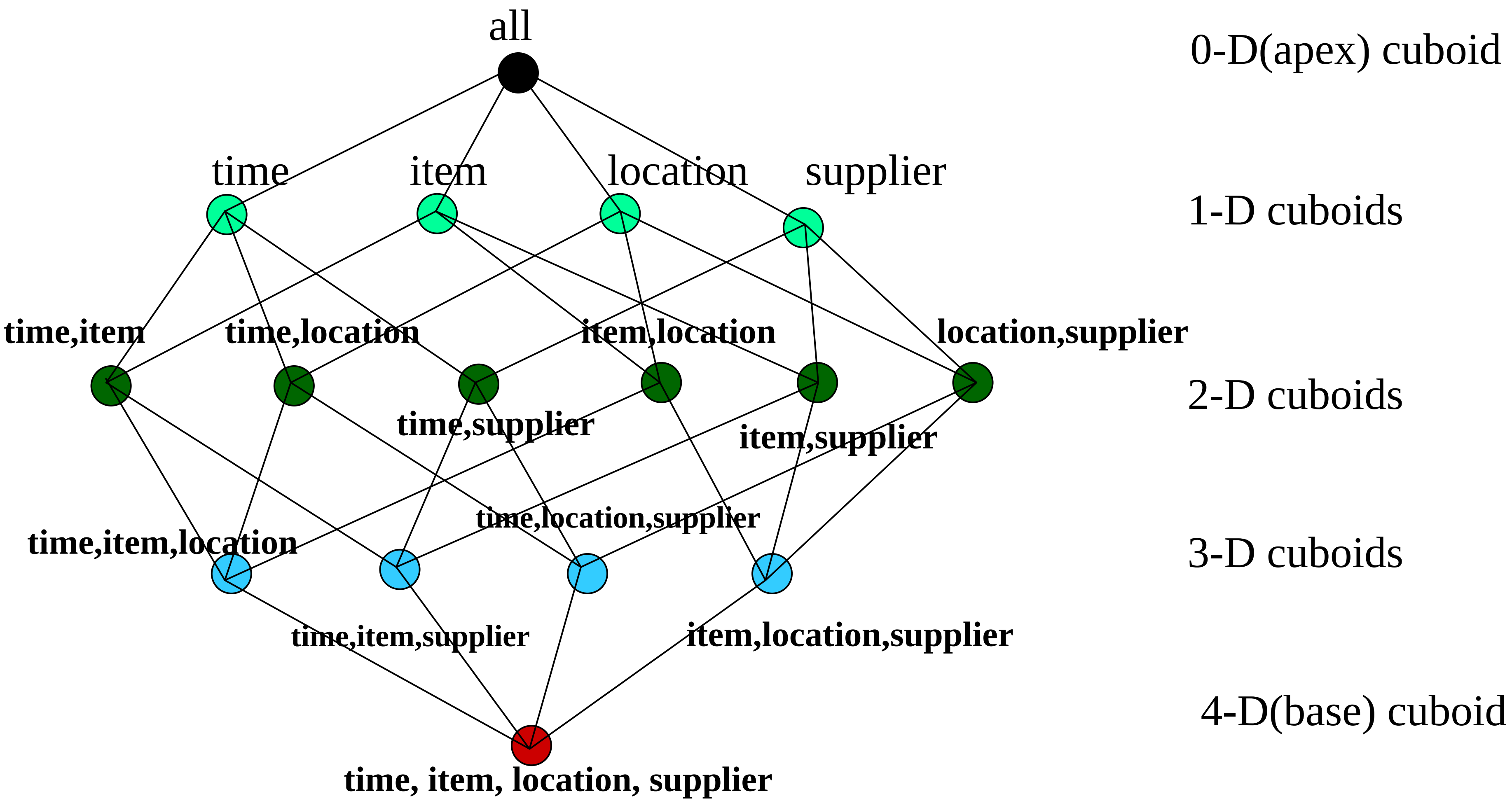
# Chapter Outline

- What is a data warehouse?
- How to construct a Data Warehouse
  - What is the Data Model used in data warehouse?
  - Data warehouse architecture
  - Data warehouse implementation

# -- 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.

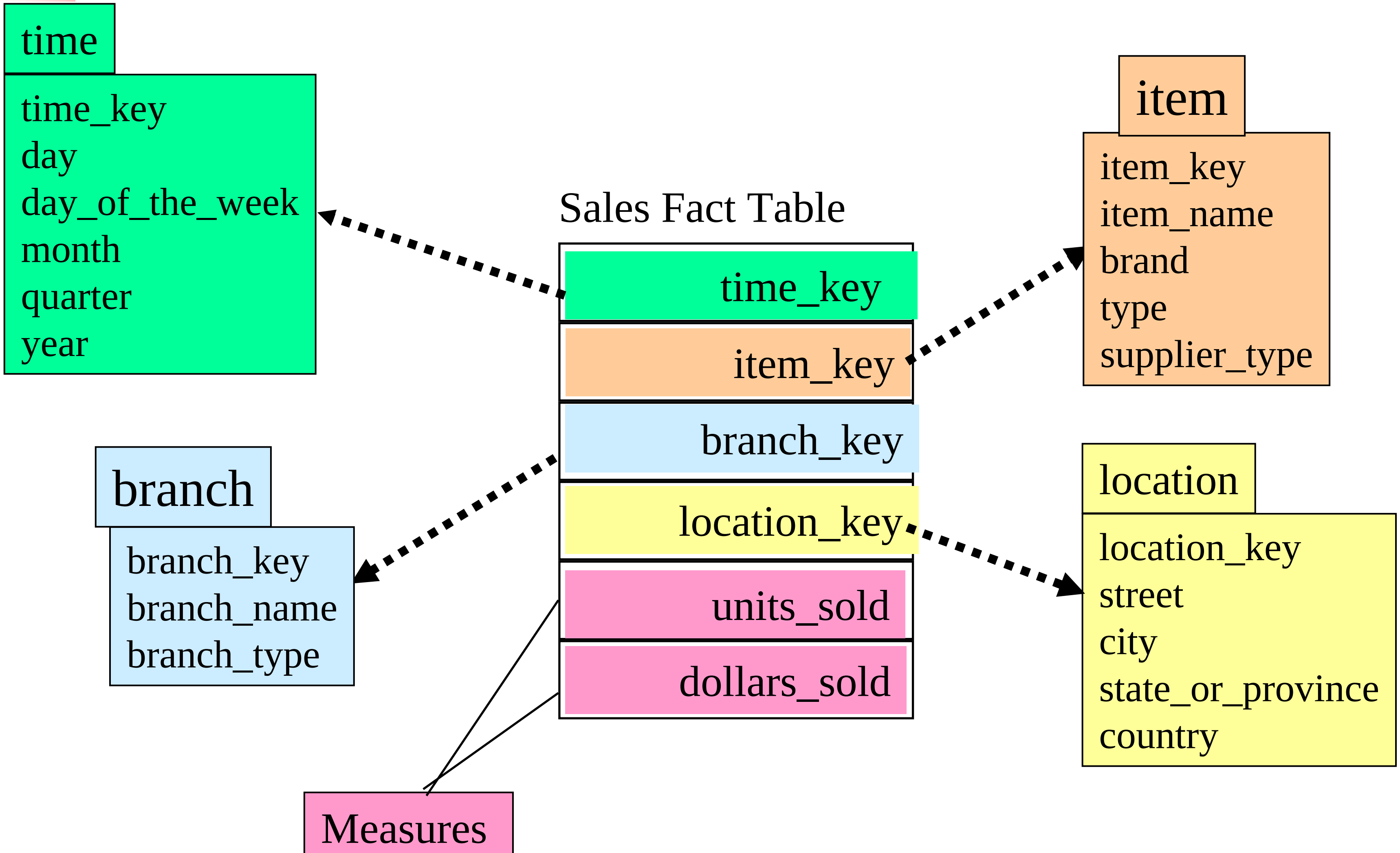
# Cube: A Lattice of Cuboids



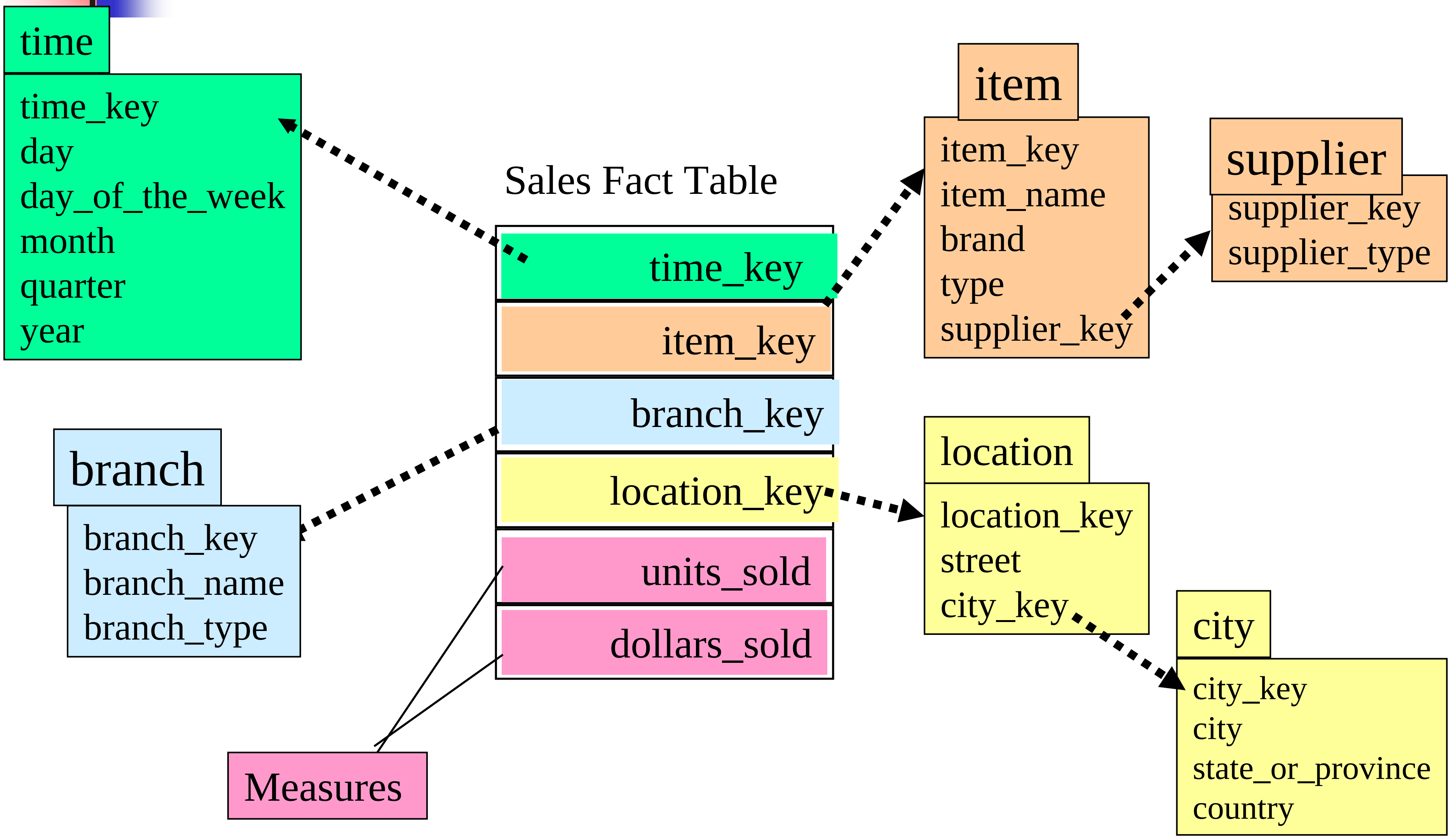
# -- Conceptual Modeling of Data Warehouses

- 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

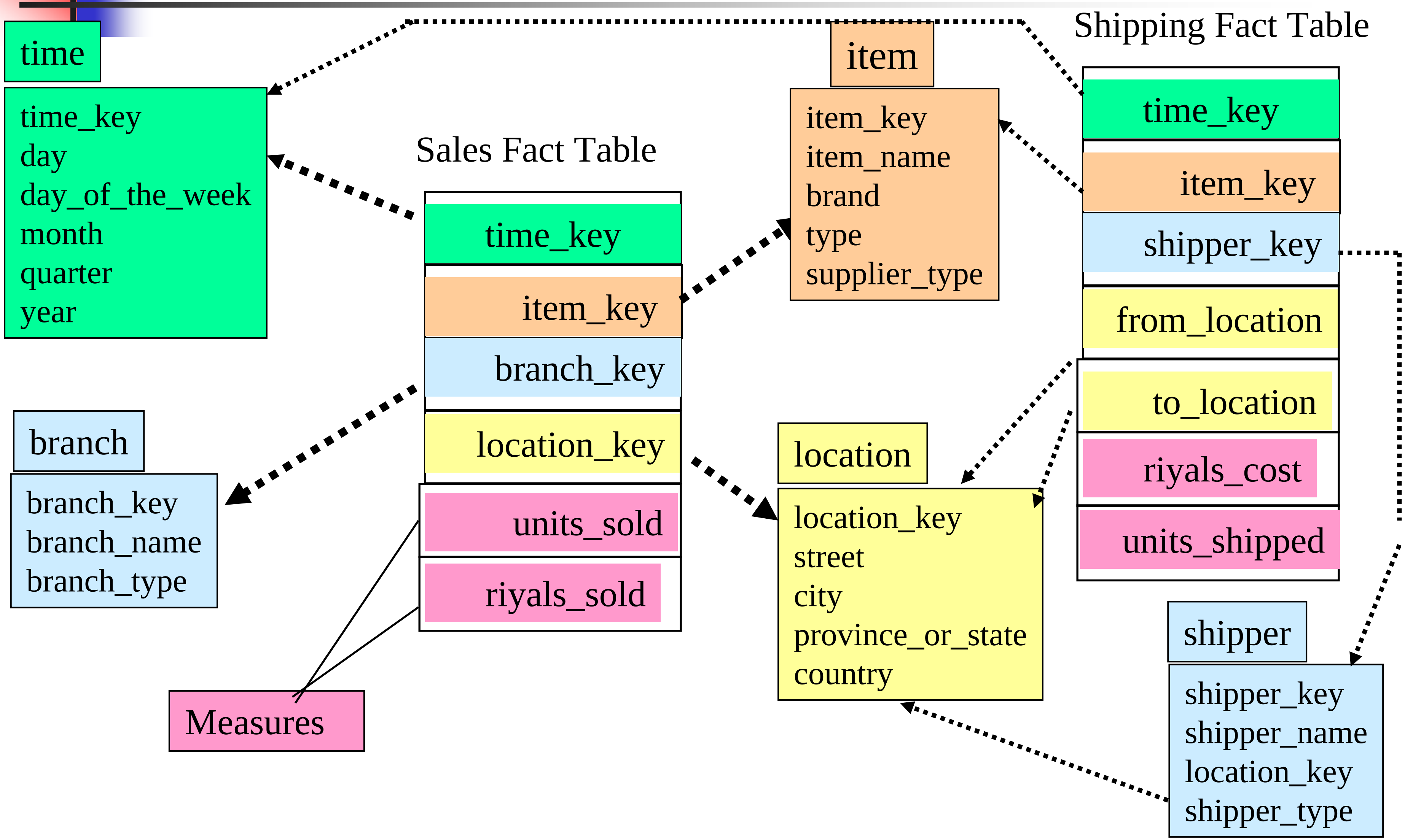
# --- Example of Star Schema



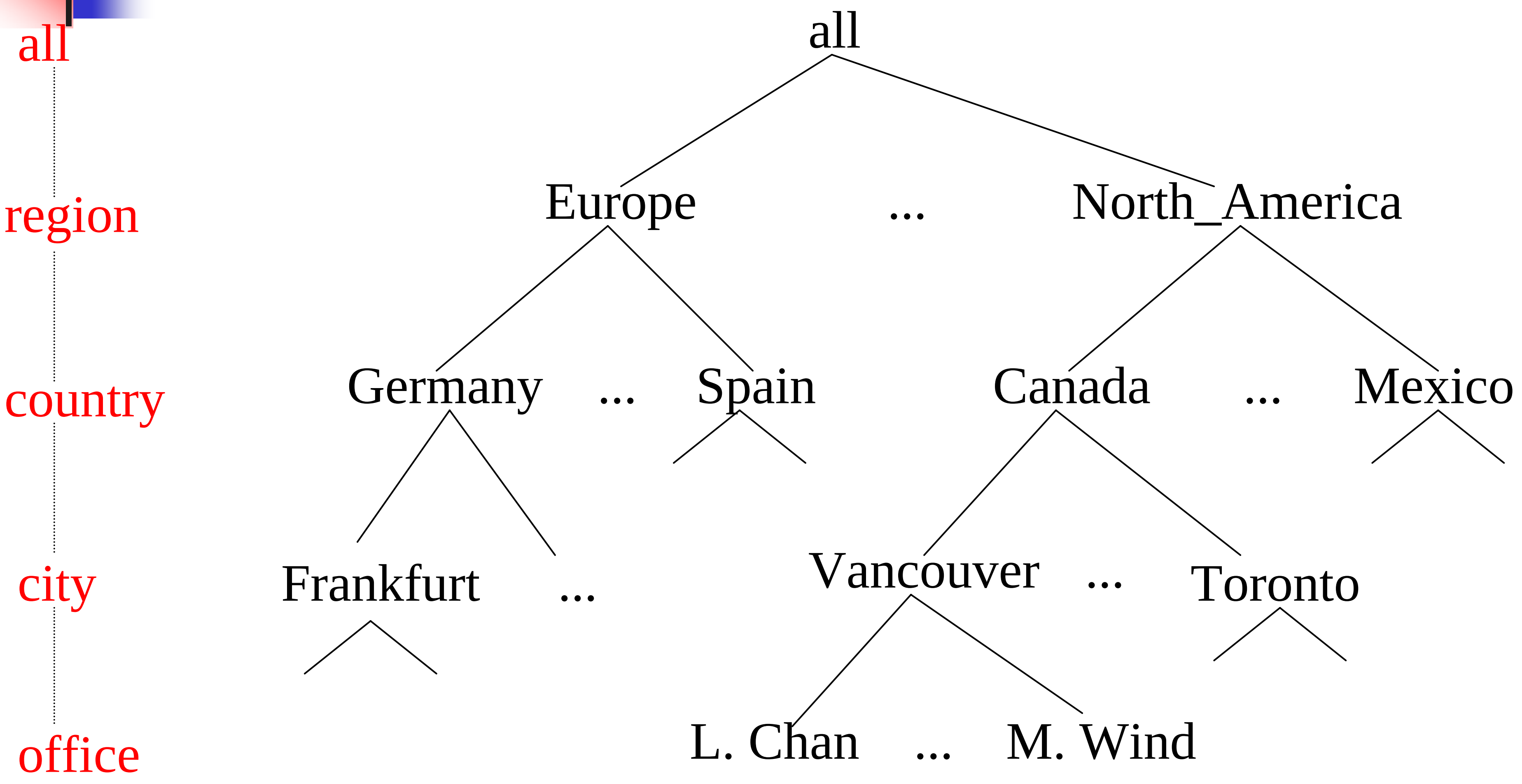
# -- Example of Snowflake Schema



# -- Example of Fact Constellation

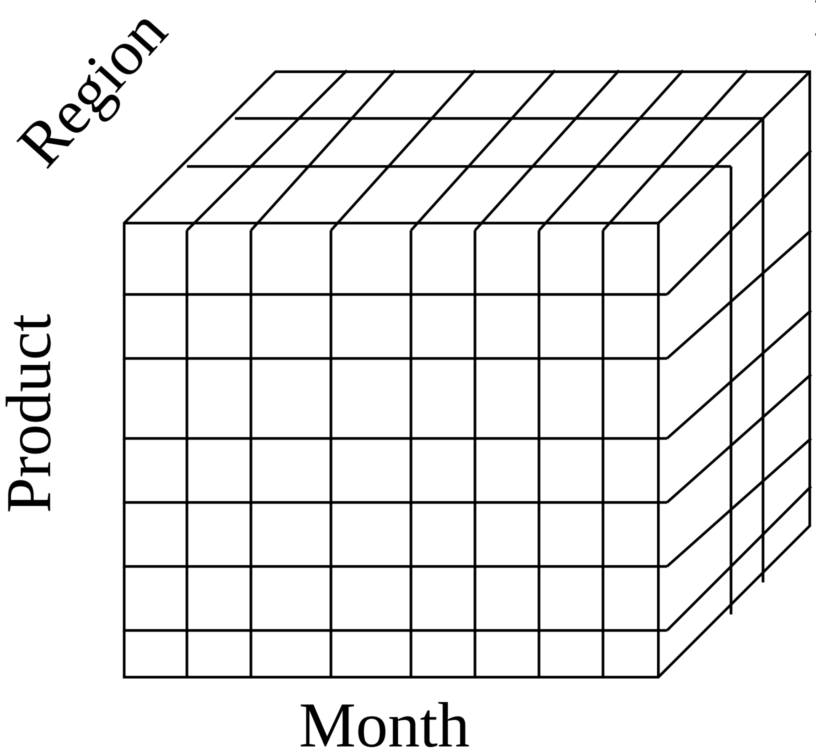


# -- A Concept Hierarchy: Dimension (location)

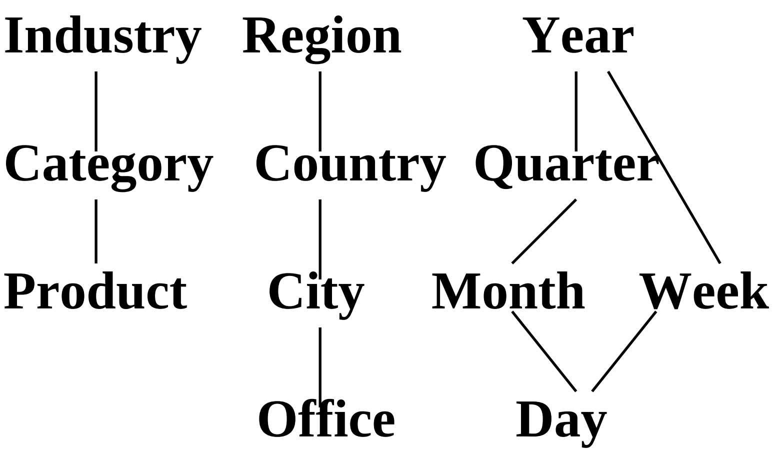


## -- Multidimensional Data

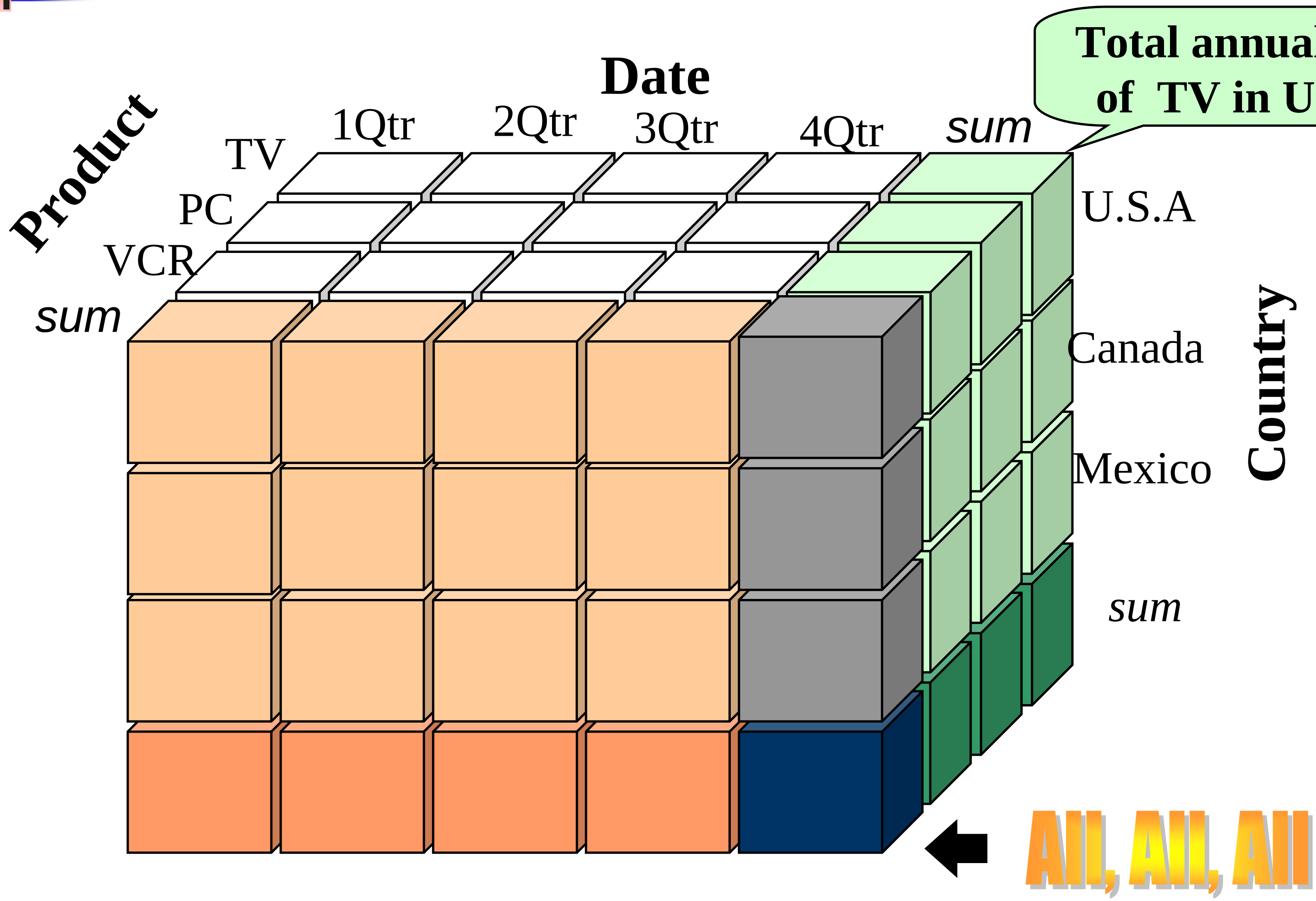
- Sales volume as a function of product, month, and region



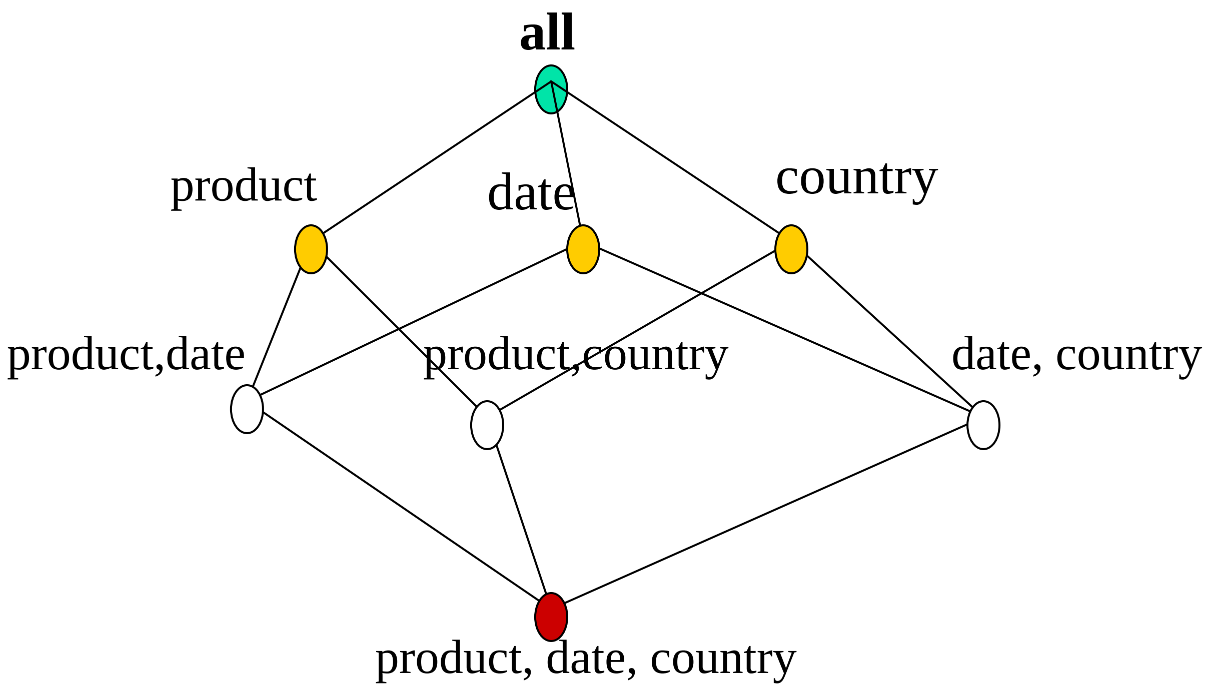

**Dimensions: Product, Location, Time**
**Hierarchical summarization paths**



# --- A Sample Data Cube



# Cuboids Corresponding to the Cube



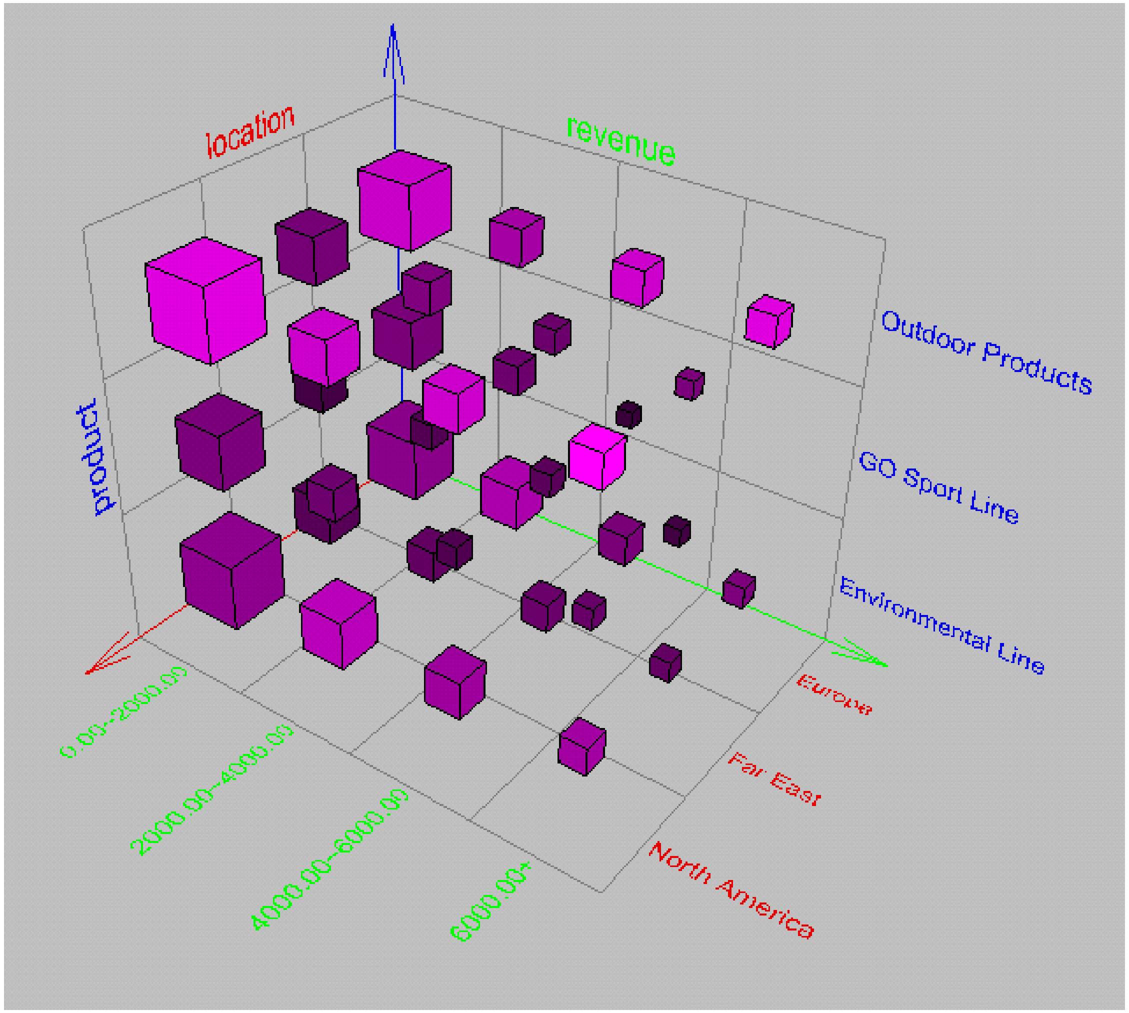

0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D(base) cuboid

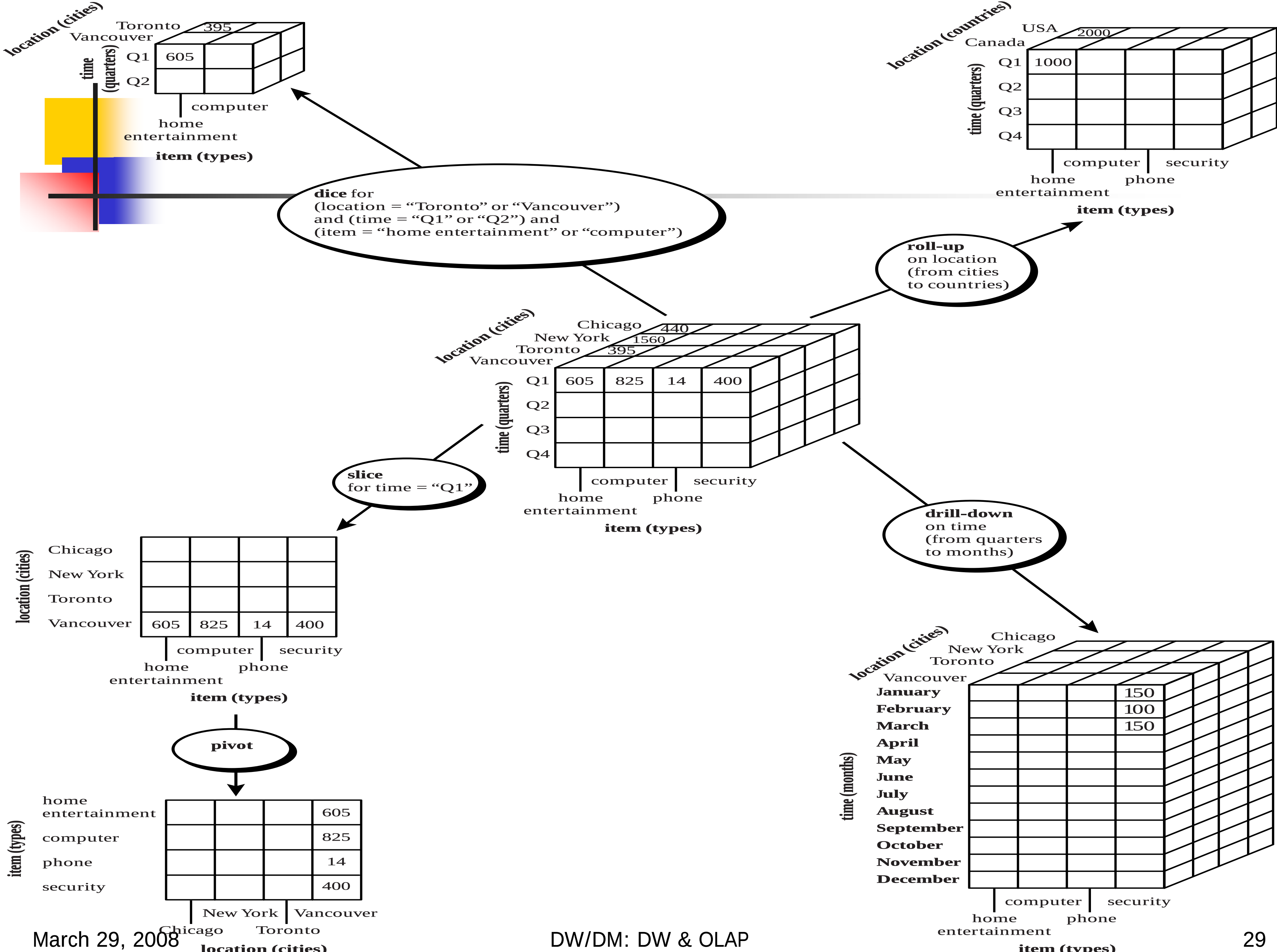
# --- Browsing a Data Cube




- Visualization
- OLAP capabilities
- Interactive manipulation

# -- 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*

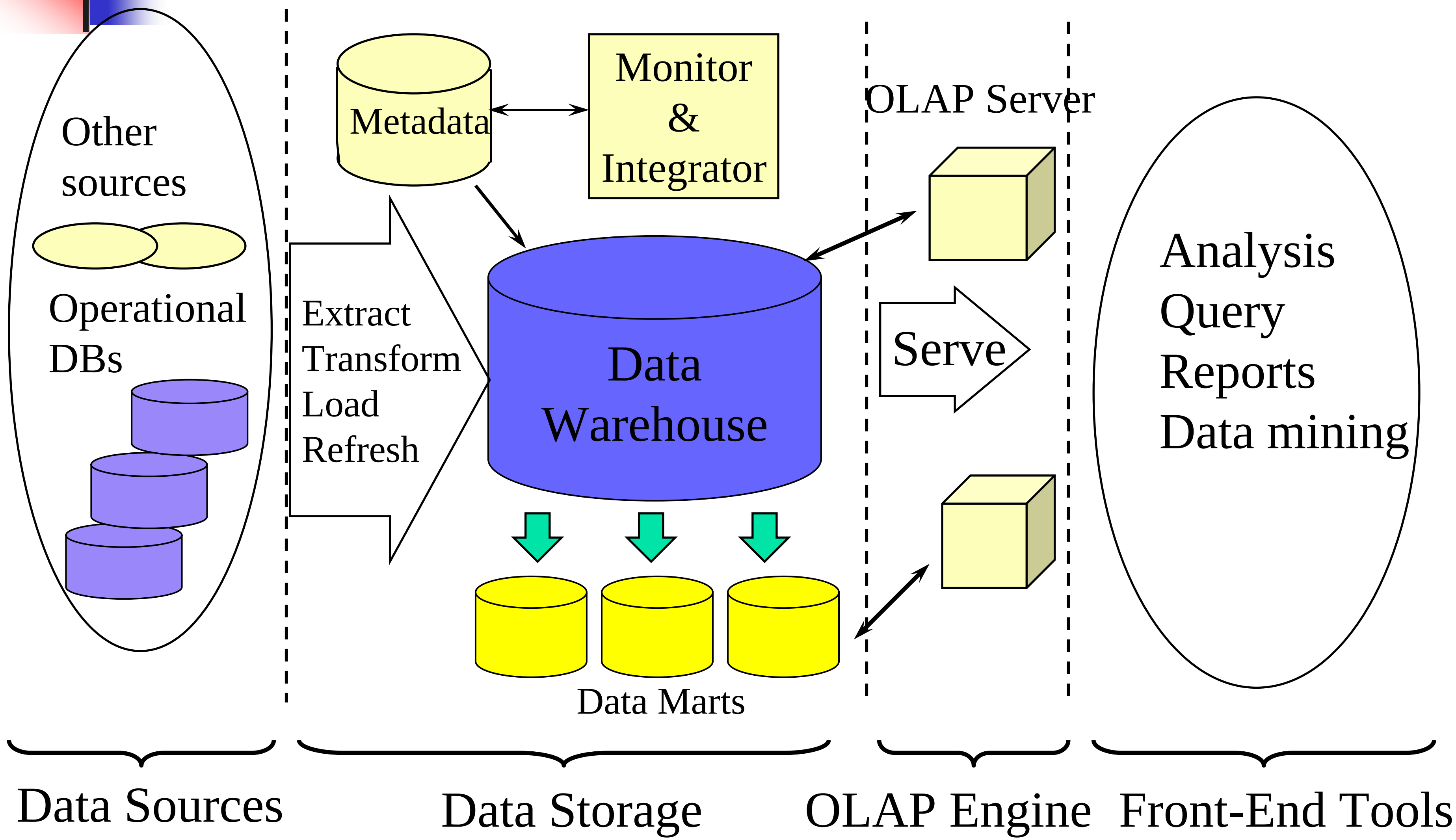


location (cities)
Toronto
Vancouver
395
605
Q1
Q2
time (quarters)
computer
home entertainment
item (types)
location (countries)
USA
Canada
2000
1000
Q1
Q2
Q3
Q4
time (quarters)
computer
security
home entertainment
phone
item (types)
dice for
(location = “Toronto” or “Vancouver”)
and (time = “Q1” or “Q2”) and
(item = “home entertainment” or “computer”)
roll-up
on location
(from cities
to countries)
location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Q1
Q2
Q3
Q4
605
825
14
400
time (quarters)
computer
security
home entertainment
phone
item (types)
slice
for time = “Q1”
drill-down
on time
(from quarters
to months)
location (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
computer
security
home entertainment
phone
item (types)
pivot
item (types)
home entertainment
computer
phone
security
605
825
14
400
New York
Vancouver
Chicago
Toronto
location (cities)
location (cities)
Chicago
New York
Toronto
Vancouver
January
February
March
April
May
June
July
August
September
October
November
December
150
100
150
time (months)
computer
security
home entertainment
phone
item (types)

# DW and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation

# Data Warehouse: A Multi-Tiered Architecture



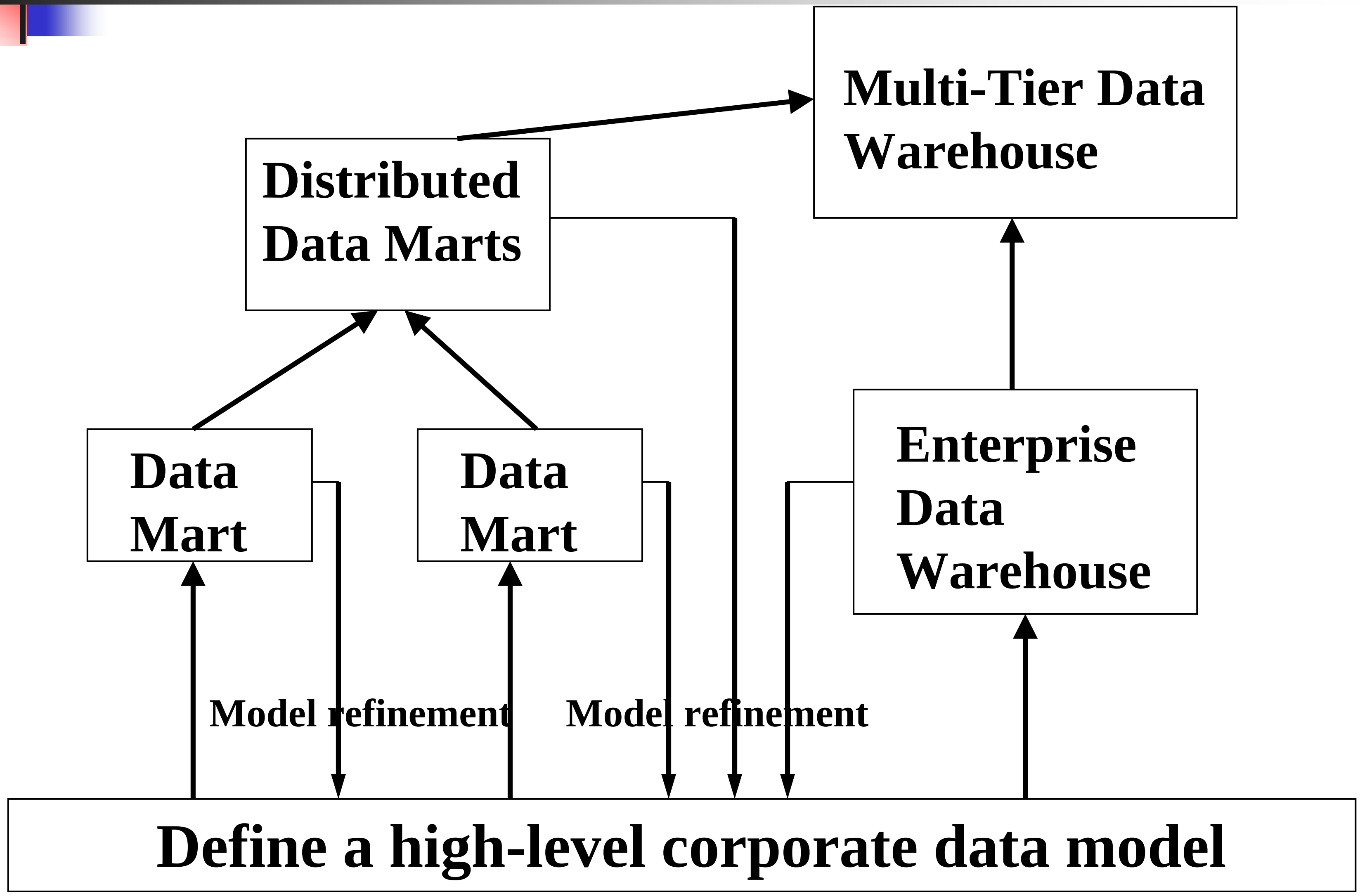
# -- DW Design Process

- Top-down, bottom-up approaches or a combination of both
  - Top-down: Starts with overall design and planning (mature)
  - Bottom-up: Starts with experiments and prototypes (rapid)
- Typical data warehouse design process
  - Choose a business process to model, e.g., orders, invoices, etc.
  - Choose the *grain* (*atomic level of data*) of the business process
  - Choose the dimensions that will apply to each fact table record
  - Choose the measure that will populate each fact table record

## -- Three DW 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

# -- DW Development: A Recommended Approach



# -- 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, 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 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
  - Measure and dimension definition algorithms
  - Data granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports

# ... -- Metadata Repository

- The mapping from operational environment to the data warehouse
  - Source databases and their contents,
  - Gateway descriptions, data partitions, data extraction, cleaning, transformation rules, and defaults, data refresh and purge rules
  - security
- Data related to system performance
  - Indices, profiles
  - Timing and scheduling of refresh
- 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
  - Greater scalability
- Multidimensional OLAP (MOLAP)
  - Sparse array-based multidimensional storage engine
  - Fast indexing to pre-computed summarized data
- Hybrid OLAP (HOLAP) (e.g., Microsoft SQLServer)
  - Flexibility, e.g., low level: relational, high-level: array

# Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation

# -- Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - In an n-dimensional cube there are:

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

    Cuboids where Li is the levels in dimension I

  - So the questions is how many cuboids can be materialized
    - Materialize every (cuboid) (full materialization)
    - some (partial materialization) or
    - none (no materialization)

# -- Cube Operation

- Cube definition and computation in DMQL

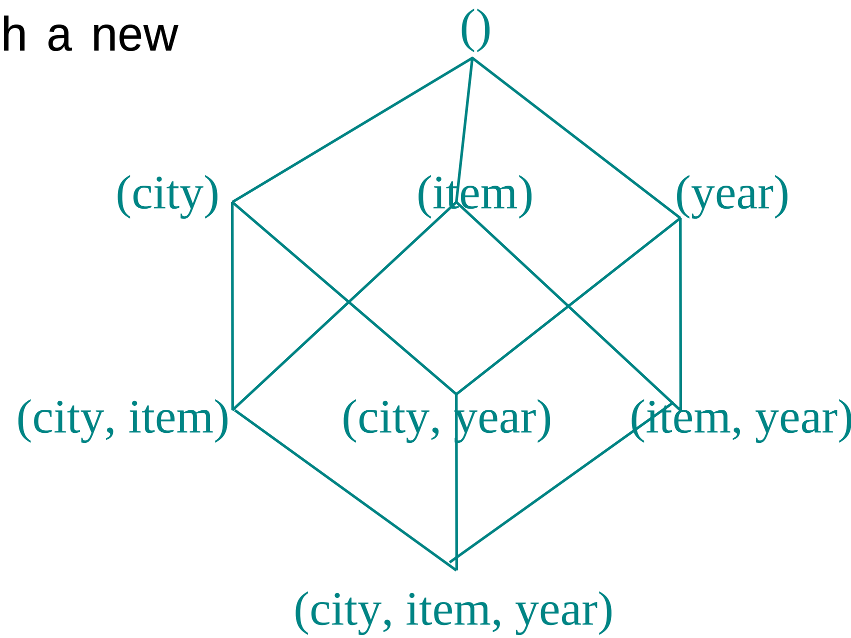
    define cube sales[item,city,year]: sum(sales_in_dollars)
    compute cube sales

- Transform it into a SQL-like language (with a new operator cube by)

    SELECT item, city, year, SUM (amount)
    FROM SALES
    CUBE BY item, city, year

- Need compute the following Group-Bys

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



# -- Indexing OLAP Data: Bitmap Index

- Index on a particular column
- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # of records in the base table
- The *i*-th bit is set if the *i*-th row of the base table has the value for the indexed column
- not suitable for high cardinality domains

**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: JI(R-id, S-id) where R (R-id, ...) ▷◁ S (S-id, ...)
- Traditional indices map the values to a list of record ids
  - It materializes relational join in JI file and speeds up relational join

- In data warehouses, join index relates the values of the dimensions of a start schema to rows in the fact table.
  - E.g. fact table: *Sales* and two dimensions *city* and *product*
    - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
  - Join indices can span multiple dimensions

# -- Efficient Processing OLAP Queries

- Determine which operations should be performed on the available cuboids
  - Transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g., dice = selection + projection
- Determine which materialized cuboid(s) should be selected for OLAP op.
  - Let the query to be processed be on {brand, province_or_state} with the condition "year = 2004", and there are 4 materialized cuboids available:

    1) {year, item_name, city}

    2) {year, brand, country}

    3) {year, brand, province_or_state}

    4) {item_name, province_or_state} where year = 2004

    Which should be selected to process the query?
- Explore indexing structures and compressed vs. dense array structs in MOLAP

# End