UNIT-1 Introduction

Lecture-1 Motivation: Why data mining?

Lecture-2 What is data mining?

Lecture-3 Data Mining: On what kind of data?

Lecture-4 Data mining functionality

Lecture-5 Classification of data mining systems

Lecture-6 Major issues in data mining

Unit-1 Data warehouse and OLAP

Lecture-7	What is a data warehouse?
Lecture-8	A multi-dimensional data model
Lecture-9	Data warehouse architecture
Lecture-10&11	Data warehouse implementation
Lecture-12	From data warehousing to data mining

Lecture-1

Motivation: Why data mining?

Evolution of Database Technology

- 1960s and earlier:
- Data Collection and Database Creation
  - Primitive file processing

Evolution of Database Technology

- 1970s - early 1980s:
- Data Base Management Systems
  - Hieratical and network database systems
  - Relational database Systems
  - Query languages: SQL
  - Transactions, concurrency control and recovery.
  - On-line transaction processing (OLTP)

Evolution of Database Technology

- Mid -1980s - present:
  - Advanced data models
    - Extended relational, object-relational
  - Advanced application-oriented DBMS
    - spatial, scientific, engineering, temporal, multimedia, active, stream and sensor, knowledge-based

Evolution of Database Technology

- Late 1980s-present
  - Advanced Data Analysis
    - Data warehouse and OLAP
    - Data mining and knowledge discovery
    - Advanced data mining appliations
    - Data mining and socity
- 1990s-present:
  - XML-based database systems
  - Integration with information retrieval
  - Data and information integreation

Evolution of Database Technology

- Present – future:
  - New generation of integrated data and information system.

Lecture-2

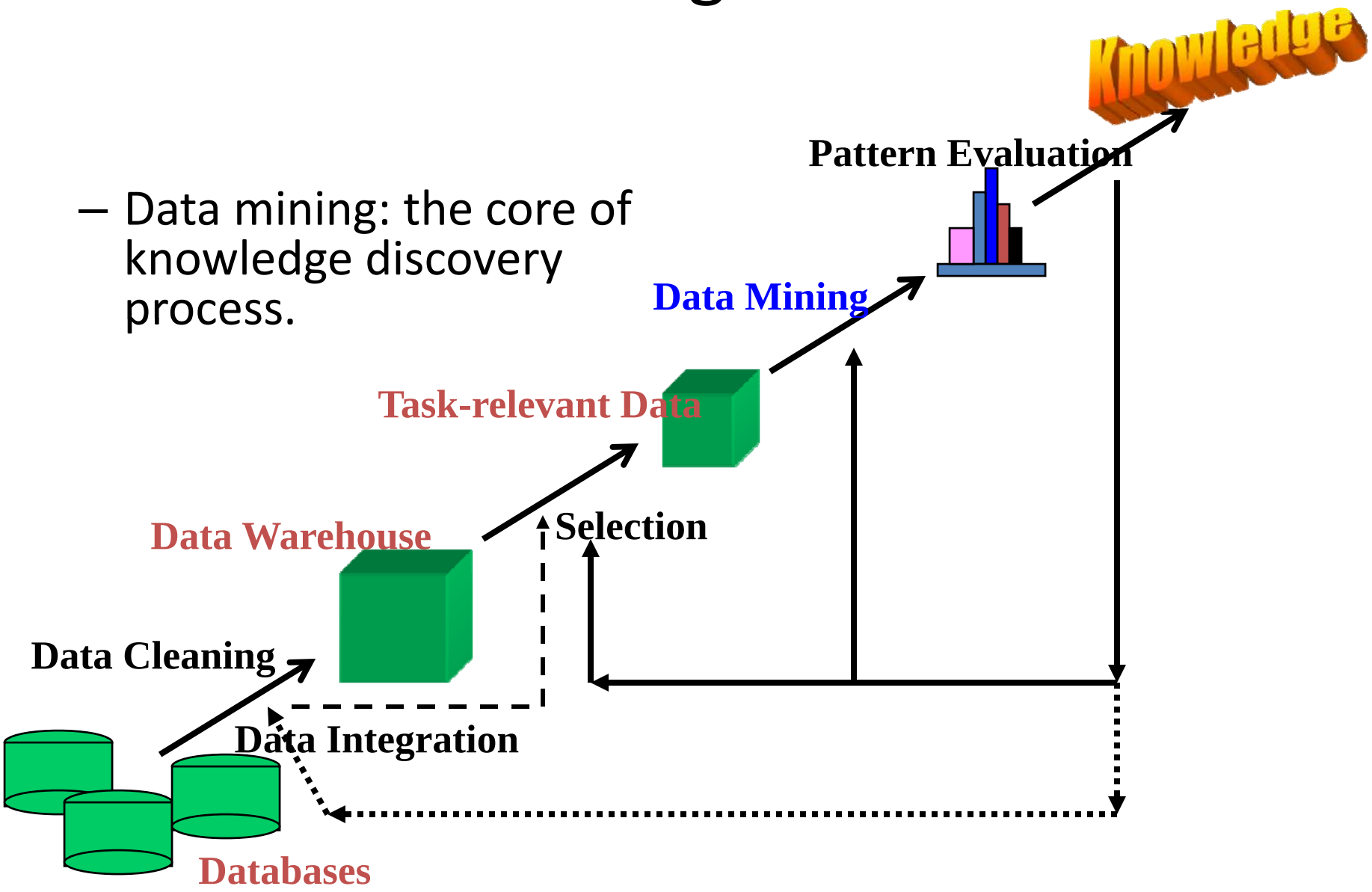
What Is Data Mining?

What Is Data Mining?

- Data mining refers to extracting or mining knowledge from large amounts of data.
- Mining of gold from rocks or sand
- Knowledge mining from data, knowledge extraction, data/pattern analysis, data archeology, and data dreding.
- Knowledge Discovery from data, or KDD

Data Mining: A KDD Process

– Data mining: the core of knowledge discovery process.



Steps of a KDD Process

1. Data cleaning
2. Data integration
3. Data selection
4. Data transformation
5. Data mining
6. Pattern evaluation
7. Knowledge presentaion

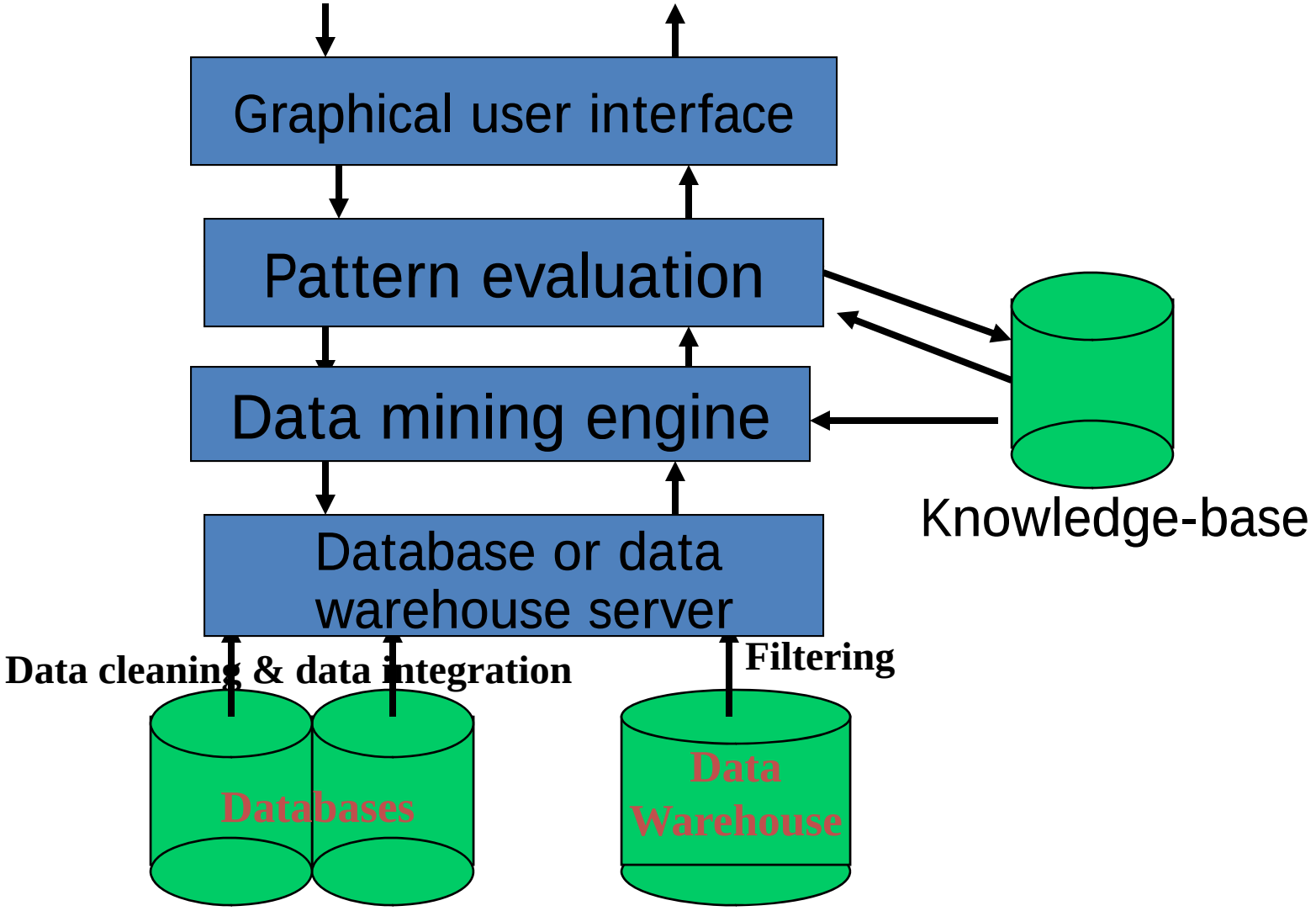
Steps of a KDD Process

- Learning the application domain:
  - relevant prior knowledge and goals of application
- Creating a target data set: data selection
- Data cleaning and preprocessing
- Data reduction and transformation:
  - Find useful features, dimensionality/variable reduction, invariant representation.

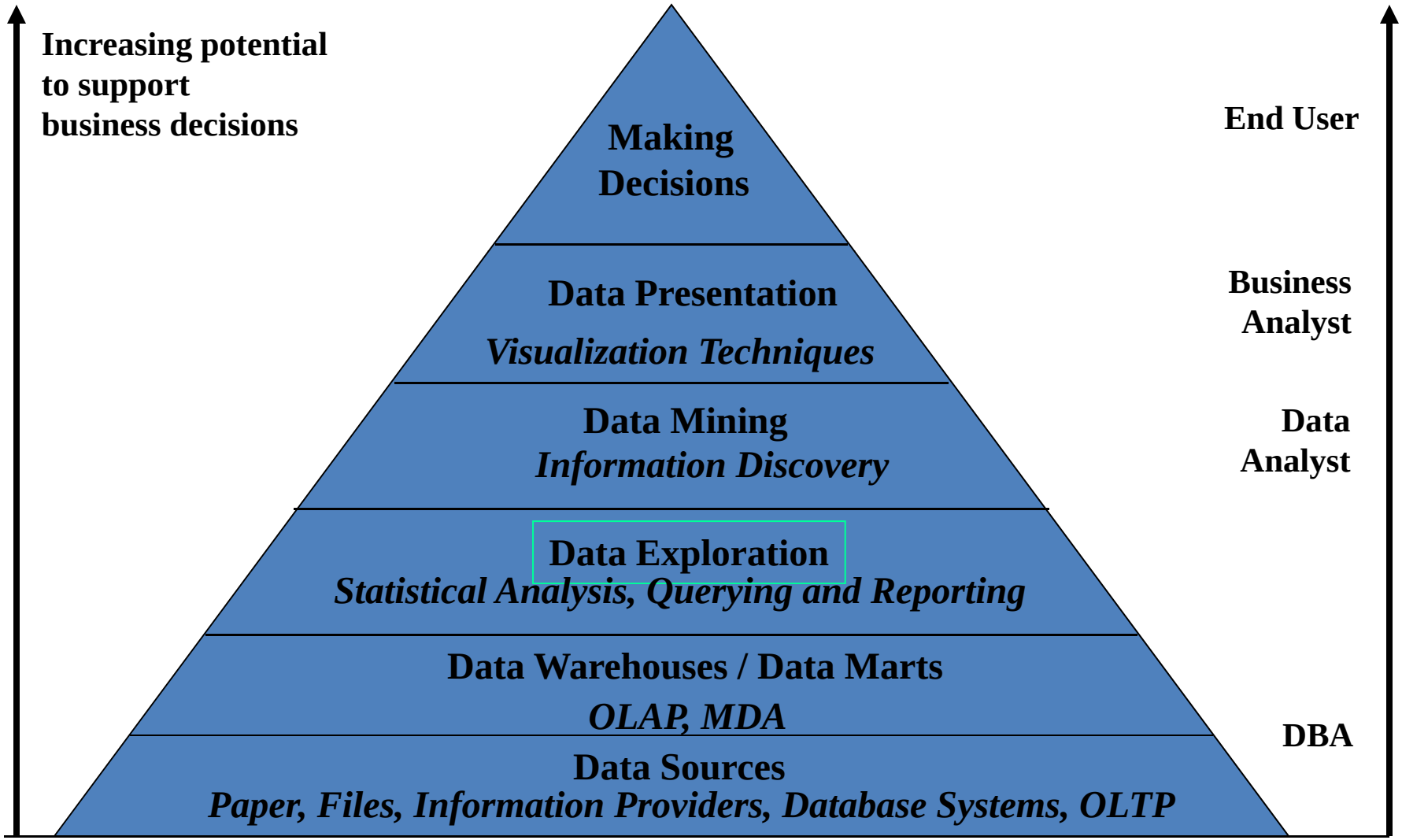
Steps of a KDD Process

- Choosing functions of data mining
  - summarization, classification, regression, association, clustering.
- Choosing the mining algorithms
- Data mining: search for patterns of interest
- Pattern evaluation and knowledge presentation
  - visualization, transformation, removing redundant patterns, etc.
- Use of discovered knowledge

Architecture of a Typical Data Mining System



Data Mining and Business Intelligence



Lecture-3

Data Mining: On What Kind of Data?

Data Mining: On What Kind of Data?

- Relational databases
- Data warehouses
- Transactional databases

Data Mining: On What Kind of Data?

- Advanced DB and information repositories
  - Object-oriented and object-relational databases
  - Spatial databases
  - Time-series data and temporal data
  - Text databases and multimedia databases
  - Heterogeneous and legacy databases
  - WWW

Lecture-4
Data Mining Functionalities

Data Mining Functionalities

- Concept description: Characterization and discrimination
  - Data can be associated with classes or concepts
  - Ex. AllElectronics store classes of items for sale include computer and printers.
  - Description of class or concept called class/concept description.
  - Data characterization
  - Data discrimination

Data Mining Functionalities

- Mining Frequent Patterns, Associations, and Correlations
  - Frequent patters- patterns occurs frequently
  - Item sets, subsequences and substructures
  - Frequent item set
  - Sequential patterns
  - Structured patterns

Data Mining Functionalities

- Association Analysis
  - Multi-dimensional vs. single-dimensional association
  - age(X, “20..29”) ^ income(X, “20..29K”) => buys(X, “PC”) [support = 2%, confidence = 60%]
  - contains(T, “computer”) => contains(x, “software”) [support=1%, confidence=75%]

Data Mining Functionalities

- Classification and Prediction
  - Finding models (functions) that describe and distinguish data classes or concepts for predict the class whose label is unknown
  - E.g., classify countries based on climate, or classify cars based on gas mileage
  - Models: decision-tree, classification rules (if-then), neural network
  - Prediction: Predict some unknown or missing numerical values

Data Mining Functionalities

- Cluster analysis
  - Analyze class-labeled data objects, clustering analyze data objects without consulting a known class label.
  - Clustering based on the principle: maximizing the intra-class similarity and minimizing the interclass similarity

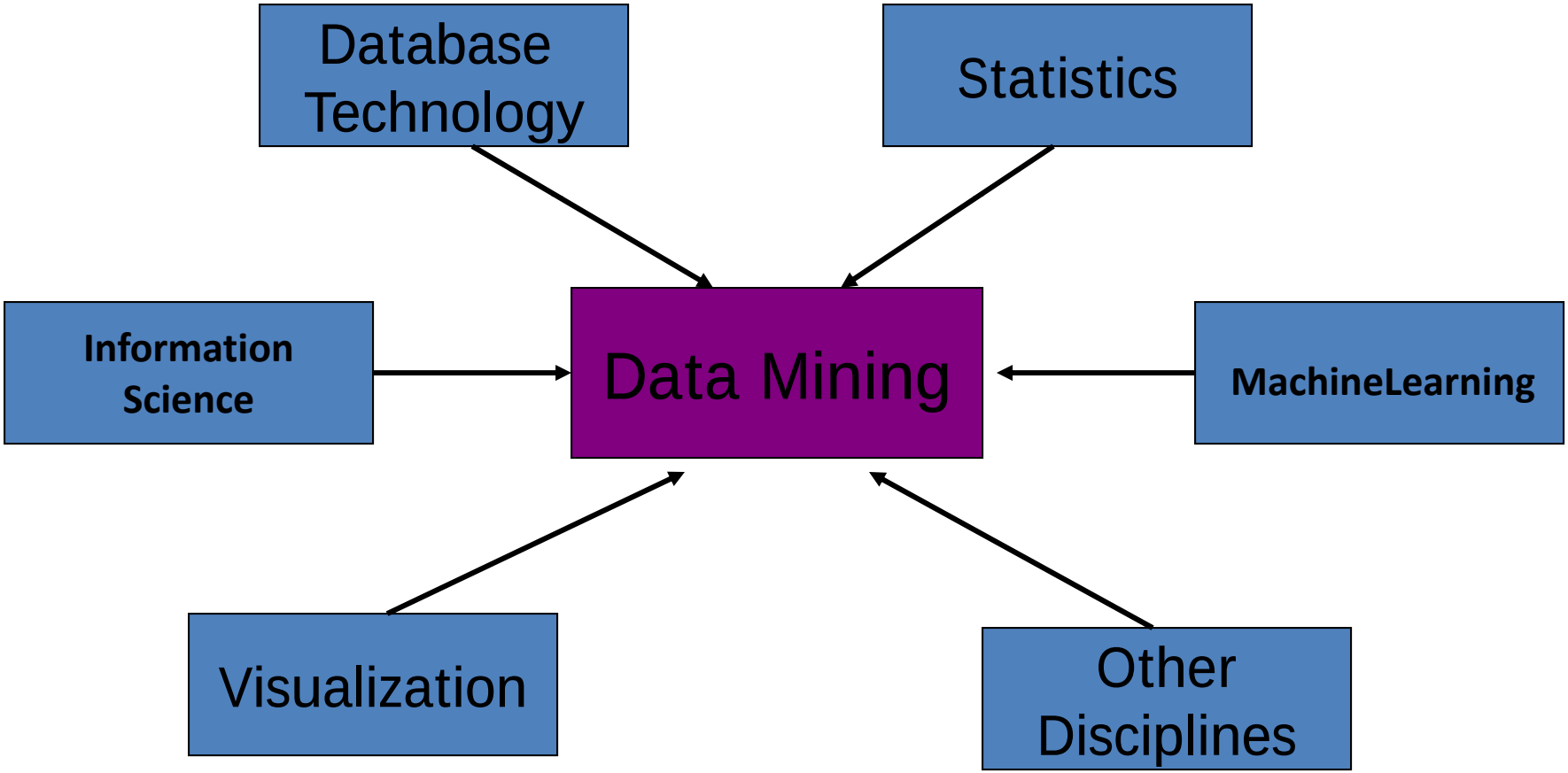
Data Mining Functionalities

- Outlier analysis
  - Outlier: a data object that does not comply with the general behavior of the model of the data
  - It can be considered as noise or exception but is quite useful in fraud detection, rare events analysis
- Trend and evolution analysis
  - Trend and deviation: regression analysis
  - Sequential pattern mining, periodicity analysis
  - Similarity-based analysis

Lecture-5

Data Mining: Classification Schemes

Data Mining: Confluence of Multiple Disciplines



Data Mining: Classification Schemes

- General functionality
  - Descriptive data mining
  - Predictive data mining
- Data mining various criteria's:
  - Kinds of databases to be mined
  - Kinds of knowledge to be discovered
  - Kinds of techniques utilized
  - Kinds of applications adapted

Data Mining: Classification Schemes

- Databases to be mined
  - Relational, transactional, object-oriented, object-relational, active, spatial, time-series, text, multi-media, heterogeneous, legacy, WWW, etc.
- Knowledge to be mined
  - Characterization, discrimination, association, classification, clustering, trend, deviation and outlier analysis, etc.
  - Multiple/integrated functions and mining at multiple levels
- analysis, Web mining, Weblog analysis, etc.

Data Mining: Classification Schemes

- Techniques utilized
  - Database-oriented, data warehouse (OLAP), machine learning, statistics, visualization, neural network, etc.
- Applications adapted
  - Retail, telecommunication, banking, fraud analysis, DNA mining, stock market

Lecture-6

Major Issues in Data Mining

Major Issues in Data Mining

- Mining methodology and user interaction issues
  - Mining different kinds of knowledge in databases
  - Interactive mining of knowledge at multiple levels of abstraction
  - Incorporation of background knowledge
  - Data mining query languages and ad-hoc data mining
  - Expression and visualization of data mining results
  - Handling noise and incomplete data
  - Pattern evaluation: the interestingness problem

Major Issues in Data Mining

- Performance issues
  - Efficiency and scalability of data mining algorithms
  - Parallel, distributed and incremental mining methods

Major Issues in Data Mining

- Issues relating to the diversity of data types
  - Handling relational and complex types of data
  - Mining information from heterogeneous databases and global information systems (WWW)

Lecture-7

What is Data Warehouse?

What is Data Warehouse?

- Defined in many different ways
  - 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—Non-Volatile

- 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

- 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

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

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 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.

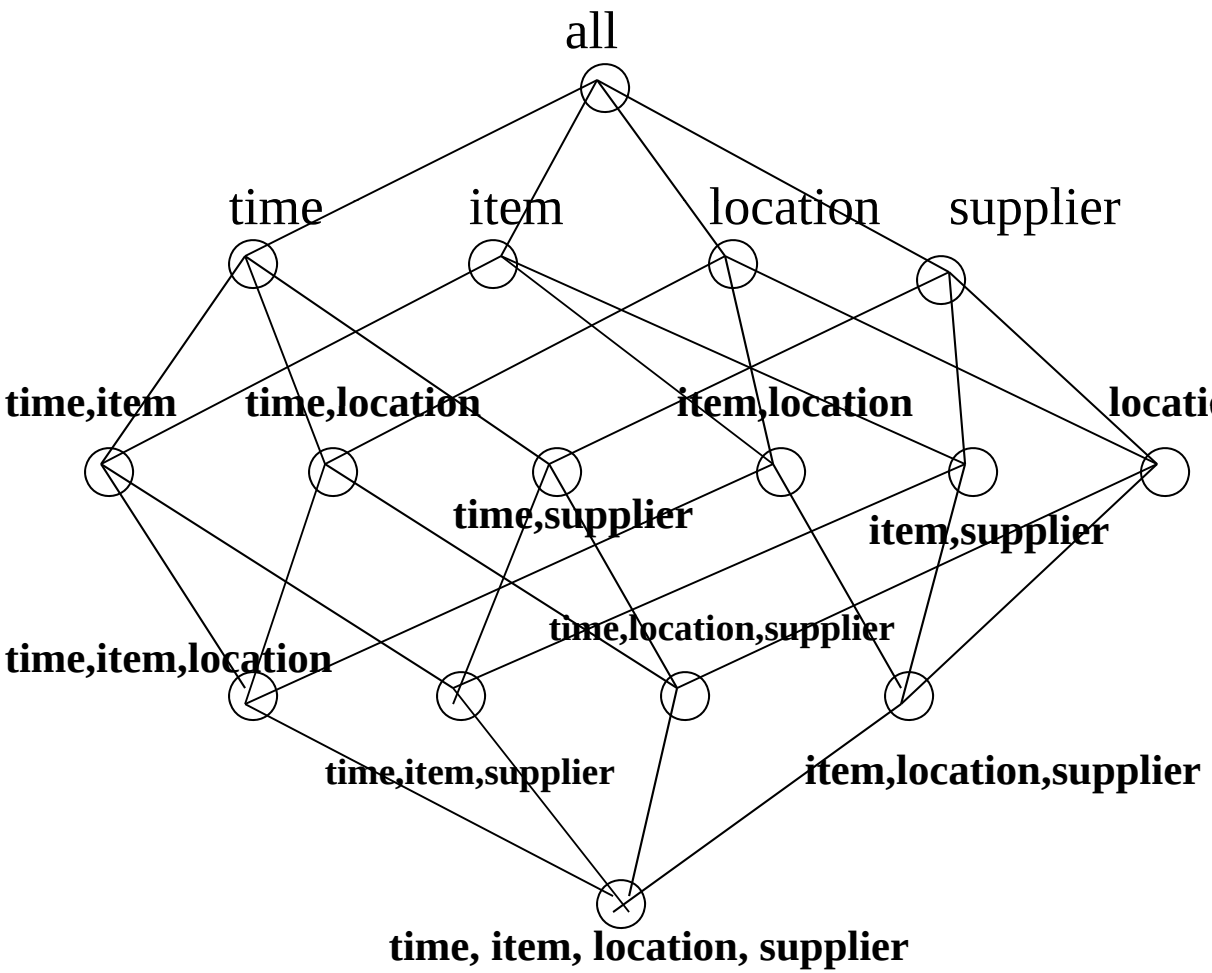
Why Separate Data Warehouse?

- 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

Lecture-8

A multi-dimensional data model

Cube: A Lattice of Cuboids



0-D(apex) cuboid

1-D cuboids

2-D cuboids

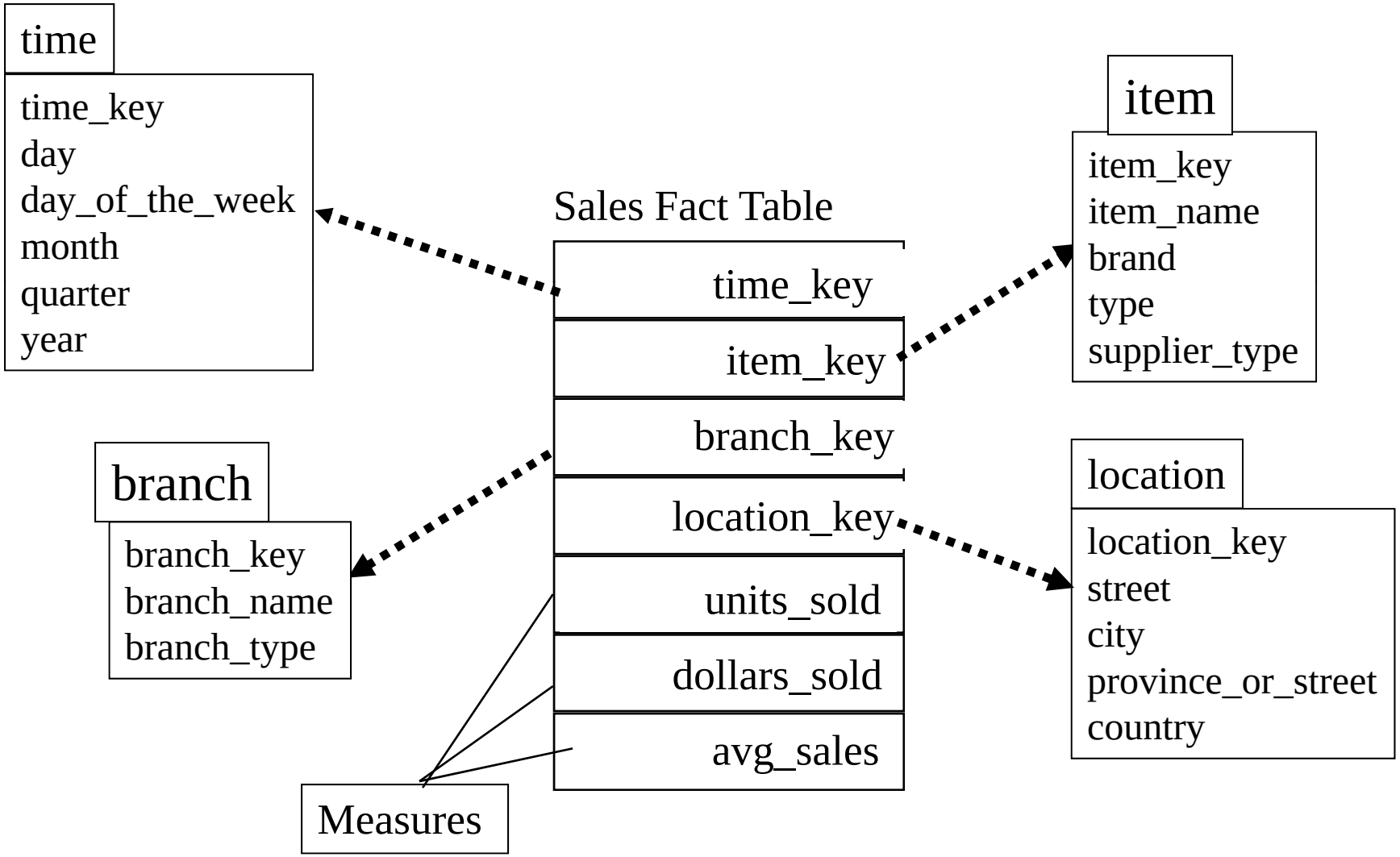
3-D cuboids

4-D(base) cuboid

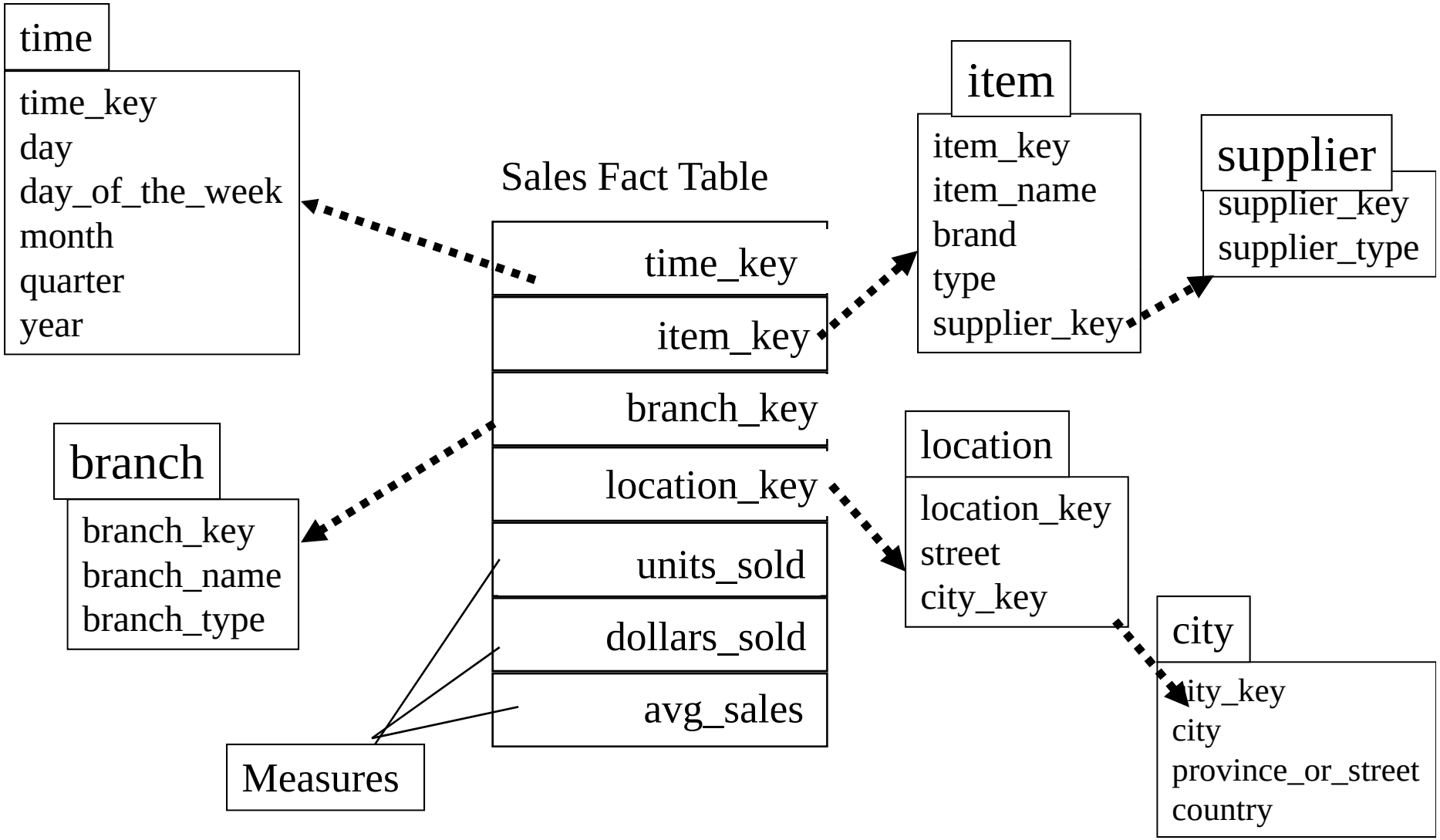
Conceptual Modeling of Data Warehouses

- 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

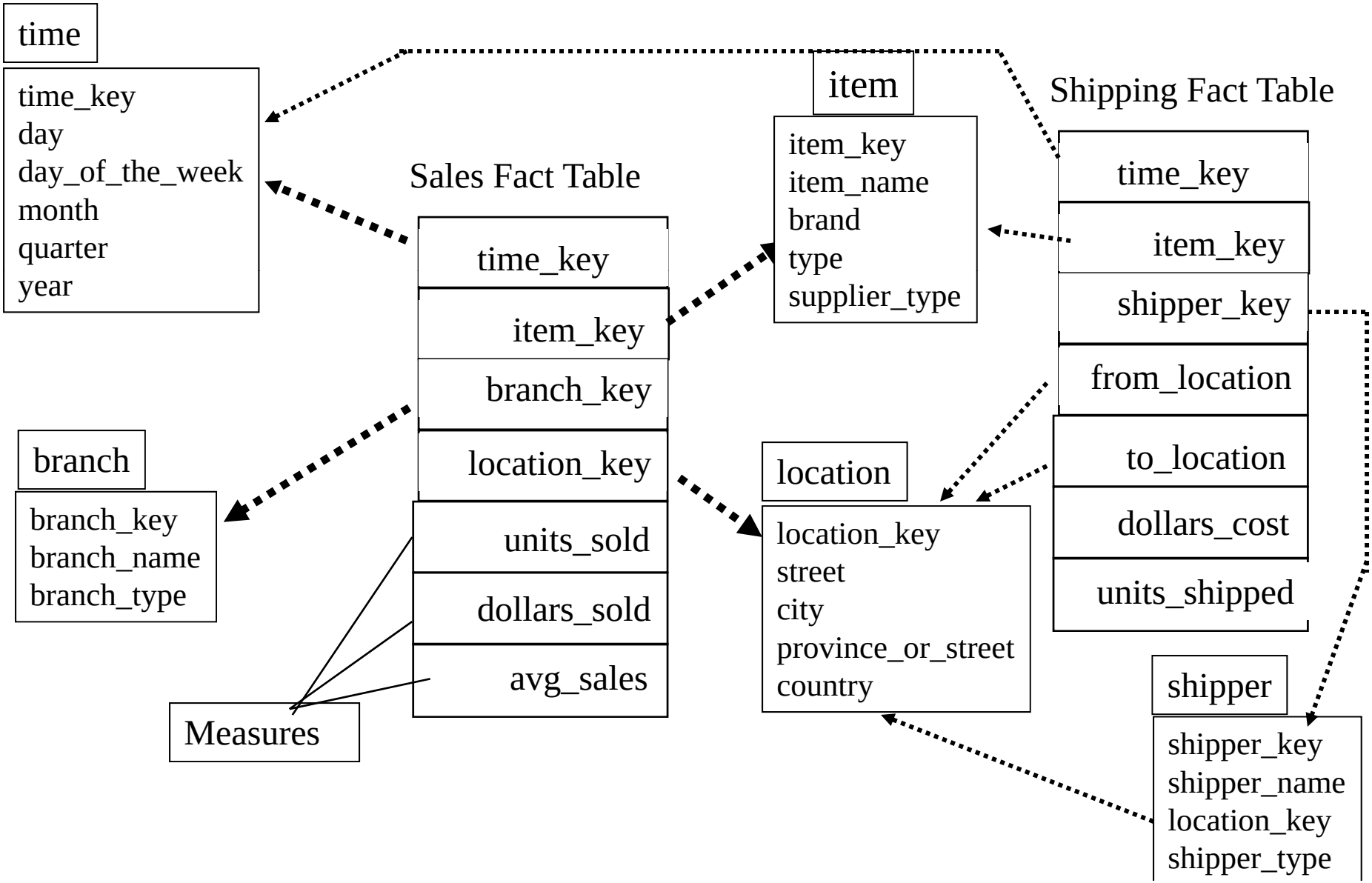
Example of Star Schema



Example of Snowflake Schema



Example of Fact Constellation



A Data Mining Query Language, DMQL: Language Primitives

- 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 a Star Schema in DMQL

define cube sales_star [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(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)

Defining a Snowflake Schema in DMQL

define cube sales_snowflake [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(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))

Defining a Snowflake Schema in DMQL

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 a Fact Constellation in DMQL

define cube sales [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(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)

Defining a Fact Constellation in DMQL

define cube shipping [time, item, shipper, from_location, to_location]:

dollar_cost = sum(cost_in_dollars), unit_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

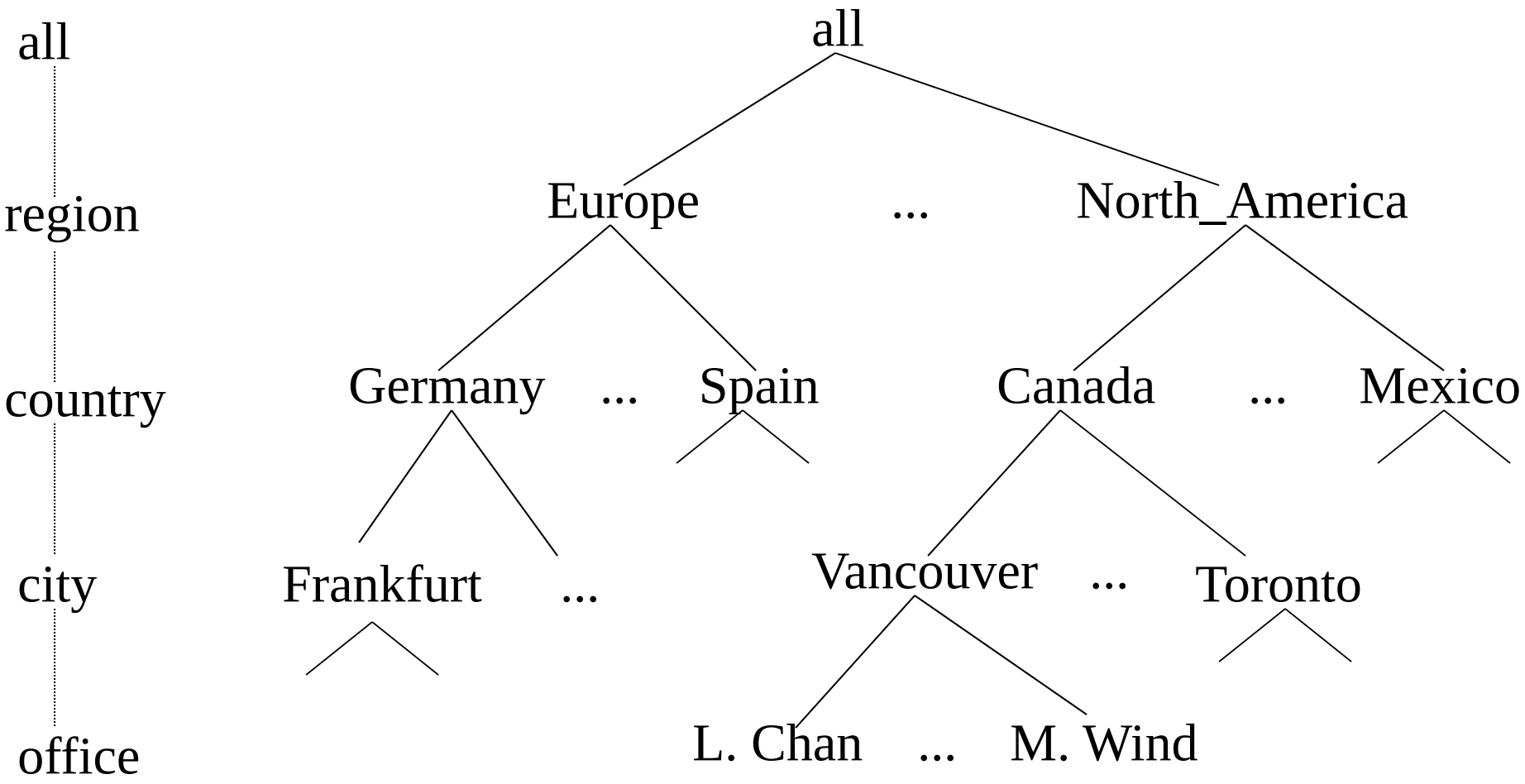
Measures: Three Categories

- 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().

Measures: Three Categories

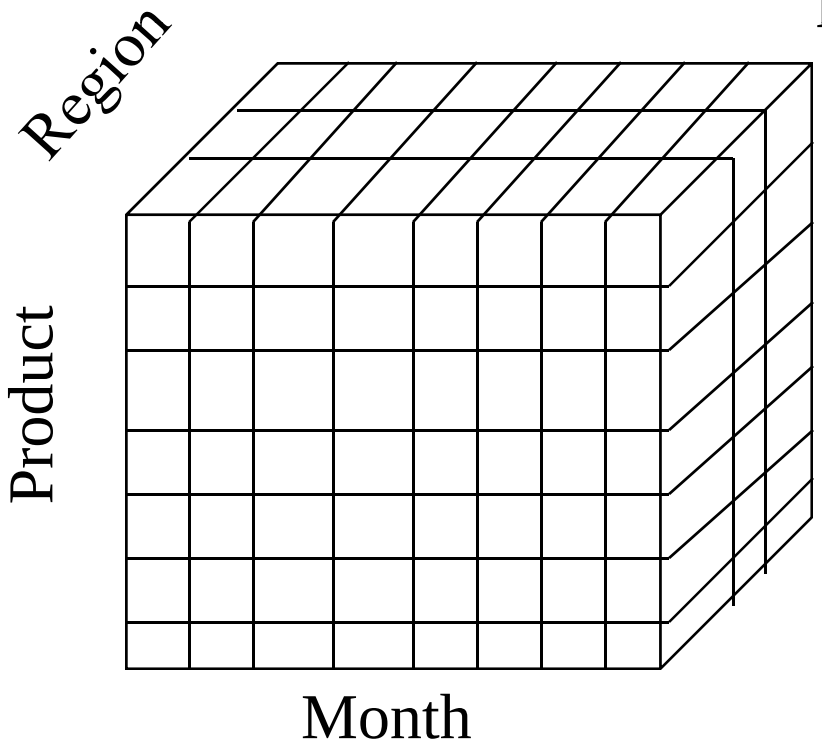
- holistic: if there is no constant bound on the storage size needed to describe a sub aggregate.
    - E.g., median(), mode(), rank().

A Concept Hierarchy: Dimension (location)

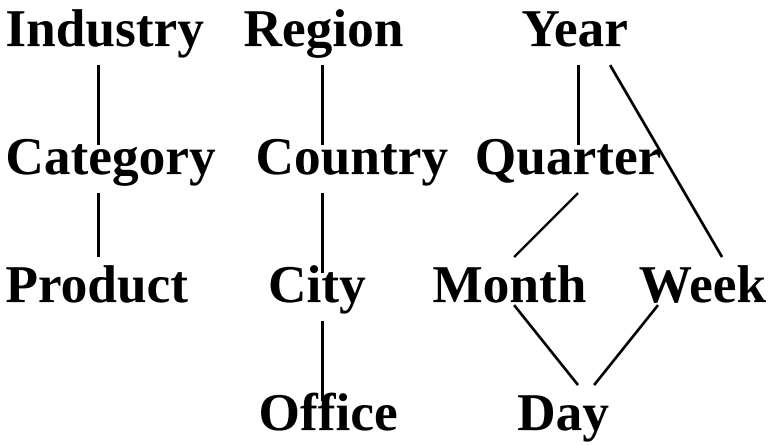


Multidimensional Data

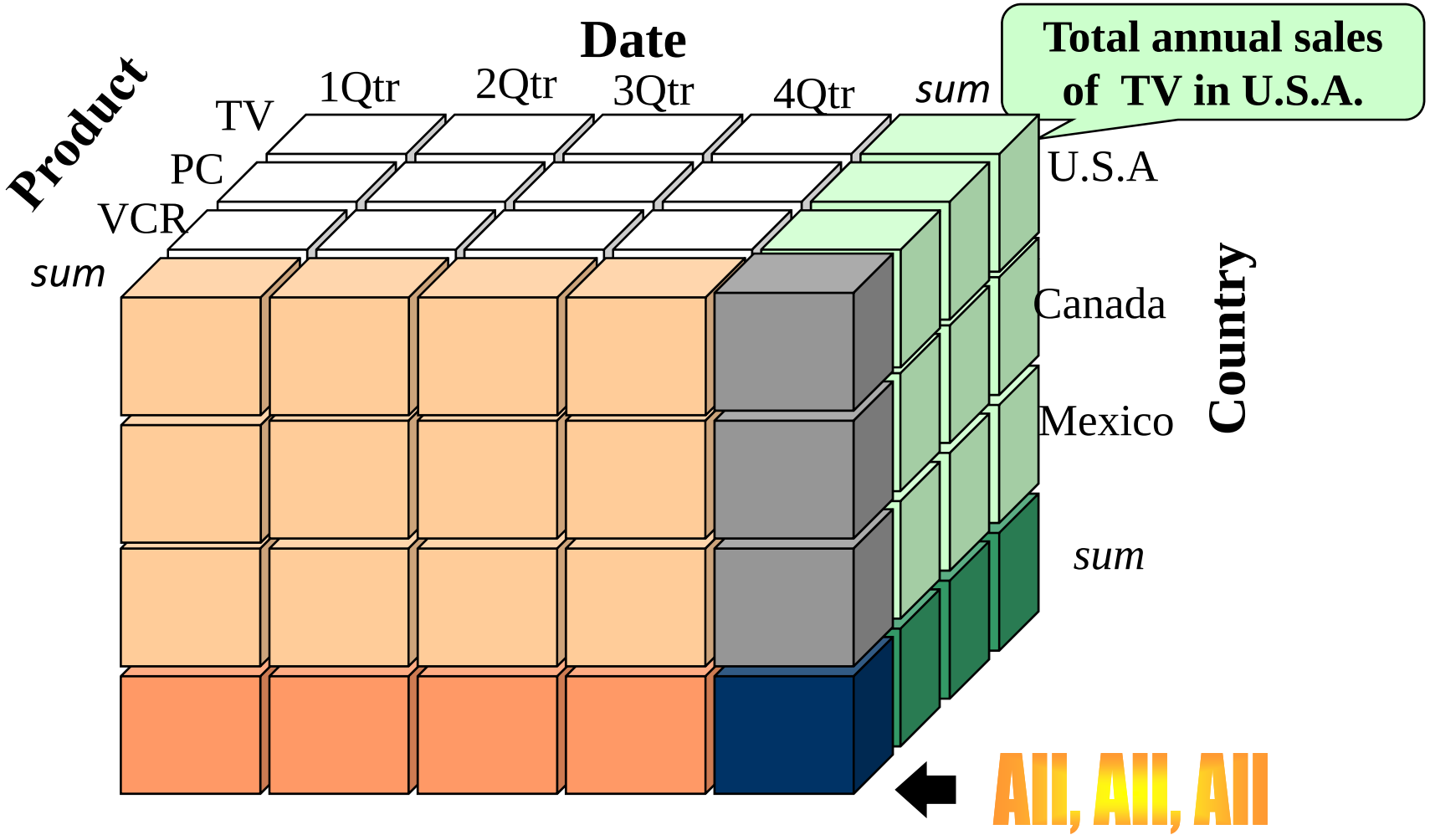
- Sales volume as a function of product, month, and region



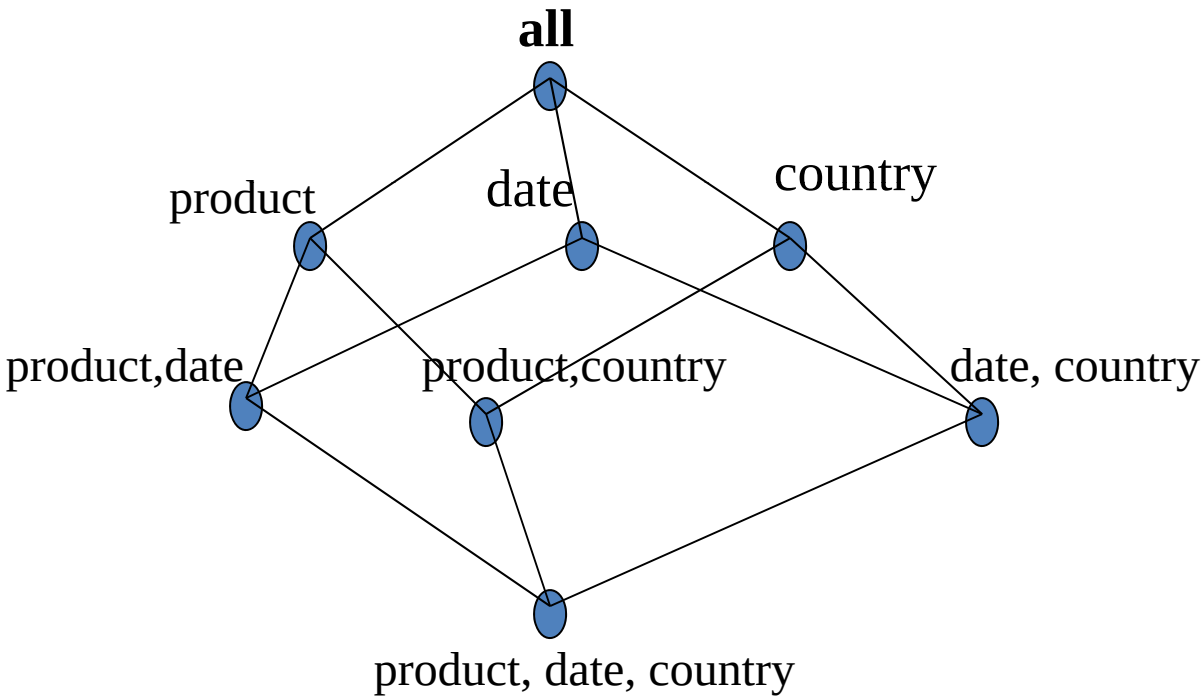
Dimensions: Product, Location, Time
Hierarchical summarization paths



A Sample Data Cube



Cuboids Corresponding to the Cube



0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D(base) cuboid

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*

OLAP Operations

- 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)*

Lecture-9

Data warehouse architecture

Steps for the Design and Construction of Data Warehouse

- The design of a data warehouse: a business analysis framework
- The process of data warehouse design
- A three-tier data ware house architecture

Design of a 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

Design of a Data Warehouse: A Business Analysis Framework

- Data warehouse view
  - consists of fact tables and dimension tables
- Data source view
  - exposes the information being captured, stored, and managed by operational systems
- Business query view
  - sees the perspectives

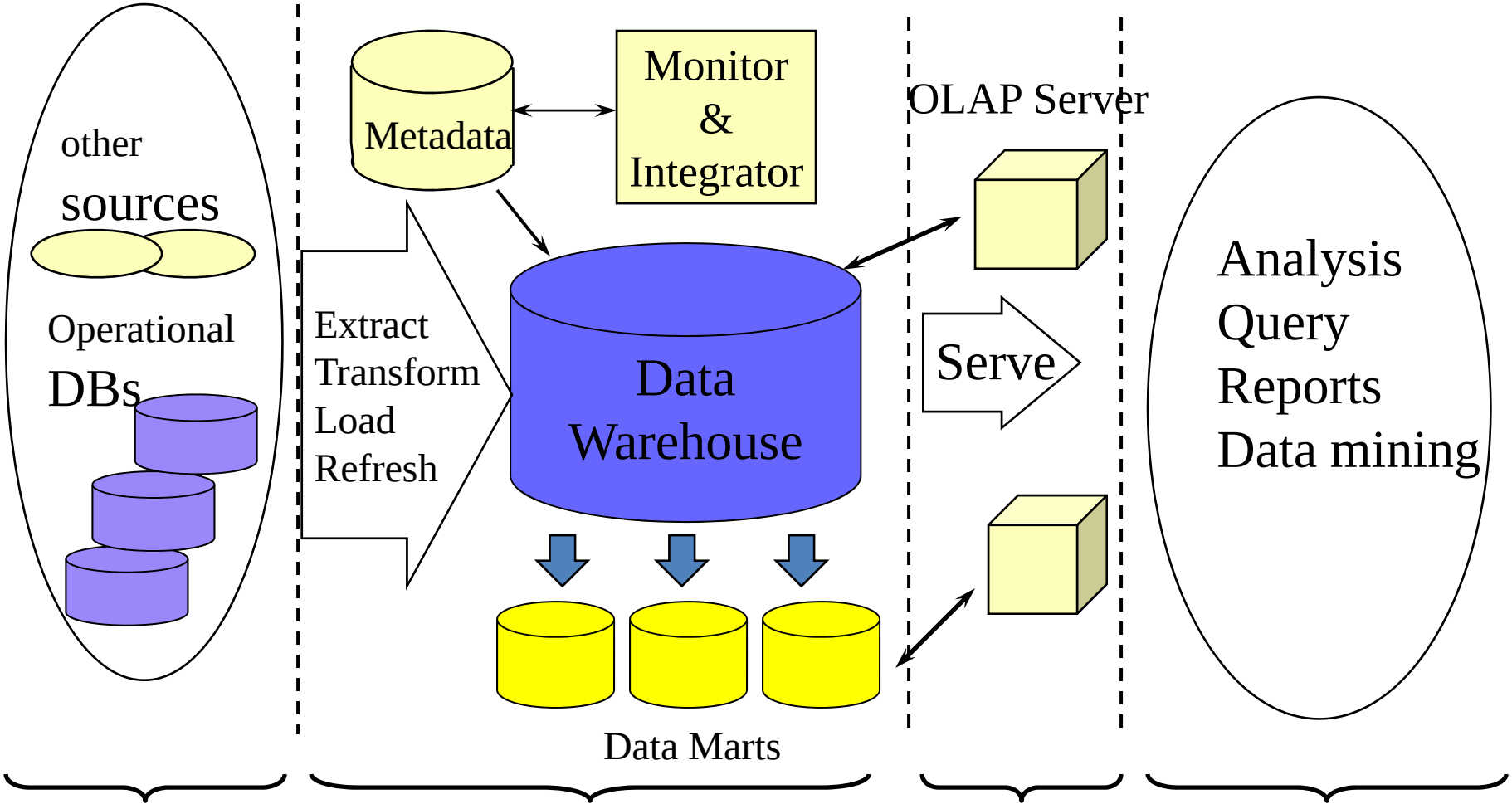
Data Warehouse Design Process

- 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)
- From software engineering point of view
  - 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

Data Warehouse Design Process

- 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

Multi-Tiered Architecture



Metadata Repository

- Meta data is the data defining warehouse objects. It has the following kinds
  - Description of the structure of the 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

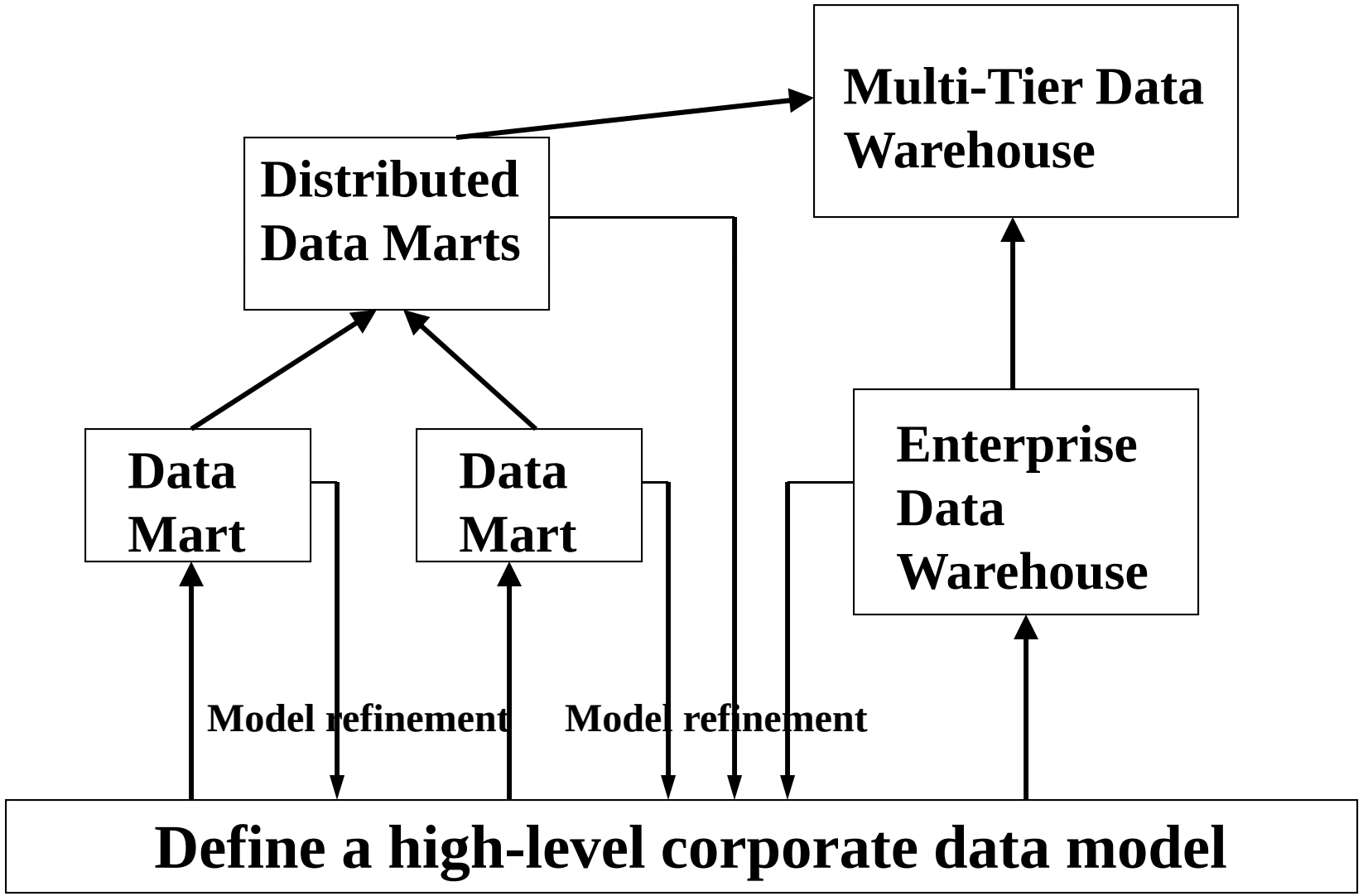
Data Warehouse Back-End Tools and Utilities

- 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

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

Data Warehouse Development: A Recommended Approach



Types of OLAP Servers

- Relational OLAP (ROLAP)
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware to support missing pieces
  - Include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
  - greater scalability
- Multidimensional OLAP (MOLAP)
  - Array-based multidimensional storage engine (sparse matrix techniques)
  - fast indexing to pre-computed summarized data

Types of OLAP Servers

- Hybrid OLAP (HOLAP)
  - User flexibility, e.g., low level: relational, high-level: array
- Specialized SQL servers
  - specialized support for SQL queries over star/snowflake schemas

Lecture-10 & 11

Data warehouse implementation

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
  - How many cuboids in an n-dimensional cube with L levels?

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

- 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.

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

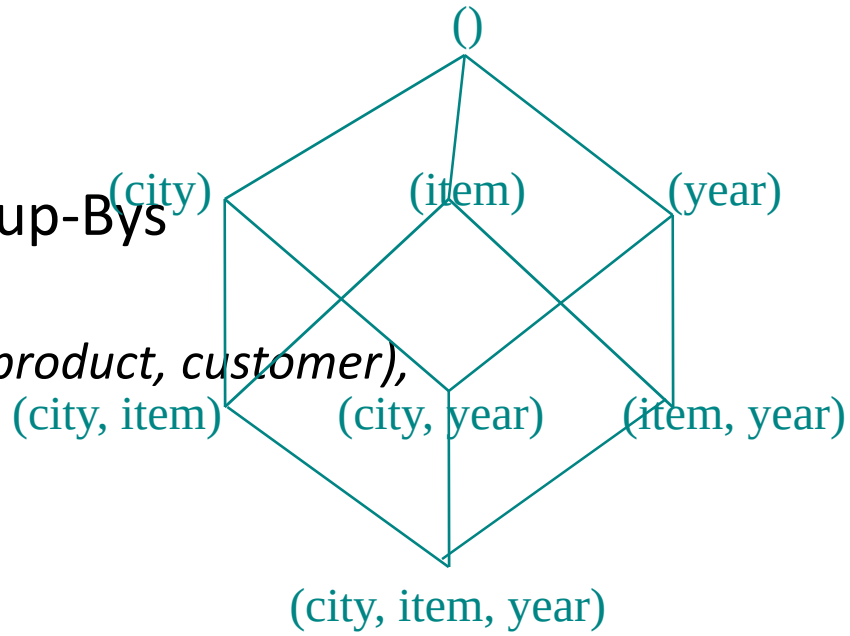
- Need compute the following Group-Bys

  (date, product, customer),

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

  (date), (product), (customer)

  ()



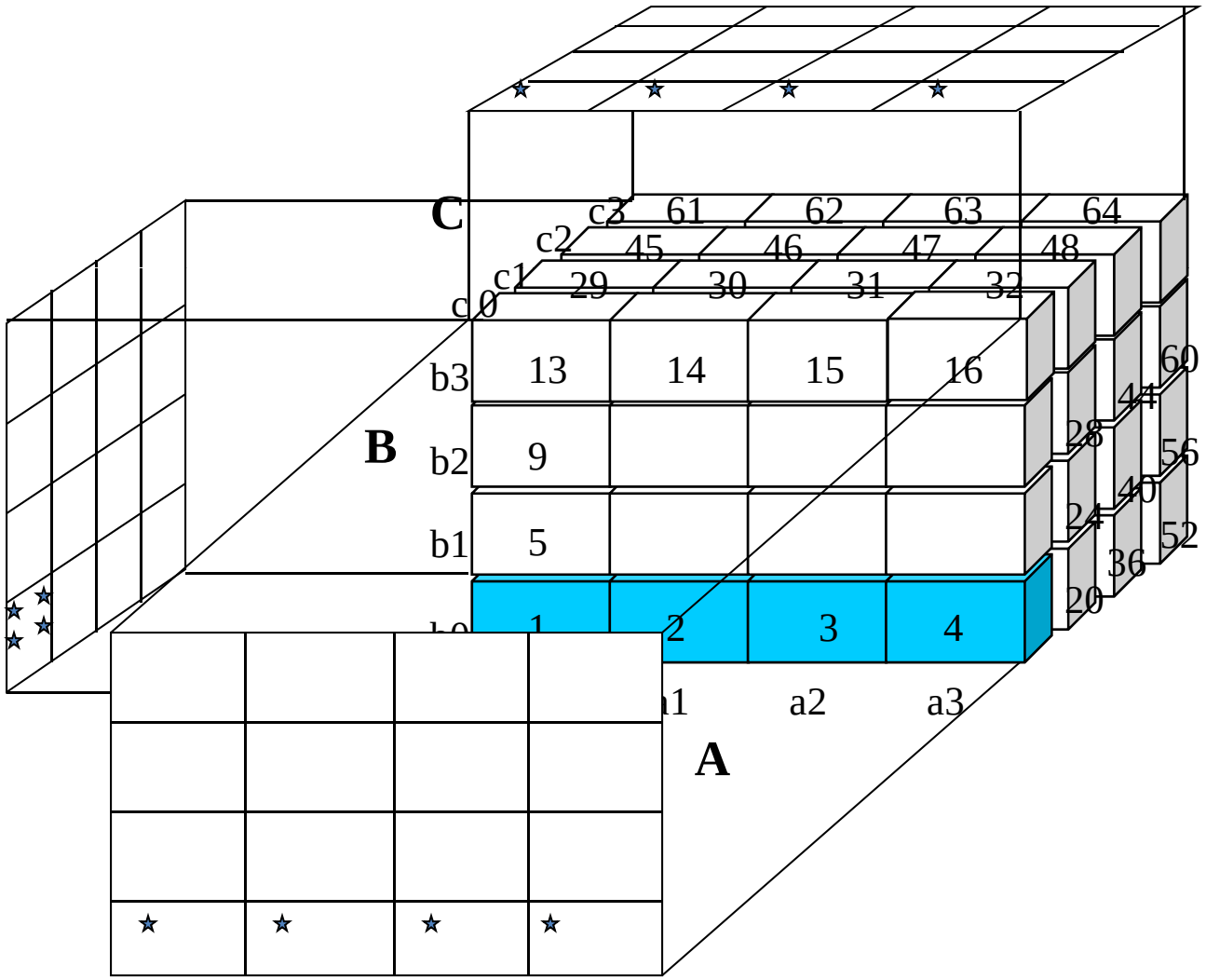
Cube Computation: ROLAP-Based Method

- Efficient cube computation methods
  - ROLAP-based cubing algorithms (Agarwal et al'96)
  - Array-based cubing algorithm (Zhao et al'97)
  - Bottom-up computation method (Bayer & Ramarkrishnan'99)
- ROLAP-based cubing algorithms
  - Sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples
  - Grouping is performed on some sub aggregates as a "partial grouping step"
  - Aggregates may be computed from previously computed aggregates, rather than from the base fact table

Multi-way Array Aggregation for Cube Computation

- Partition arrays into chunks (a small sub cube which fits in memory).
- Compressed sparse array addressing: (chunk_id, offset)
- Compute aggregates in “multi way” by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost.

Multi-way Array Aggregation for Cube Computation



Multi-Way Array Aggregation for Cube Computation

- Method: the planes should be sorted and computed according to their size in ascending order.
  - Idea: keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane
- Limitation of the method: computing well only for a small number of dimensions
  - If there are a large number of dimensions, "bottom-up computation" and iceberg cube computation methods can be explored

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

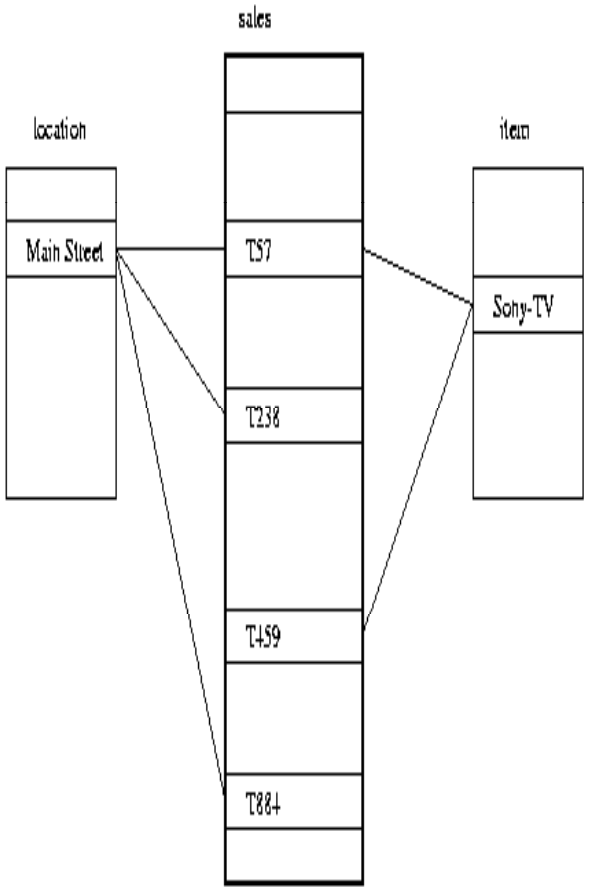
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

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 — a rather costly operation
- 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 to which materialized cuboid(s) the relevant operations should be applied.
- Exploring indexing structures and compressed vs. dense array structures in MOLAP

Lecture-12

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
    - 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.
- Differences among the three tasks

From On-Line Analytical Processing 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.
- Architecture of OLAM

An OLAM Architecture