MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

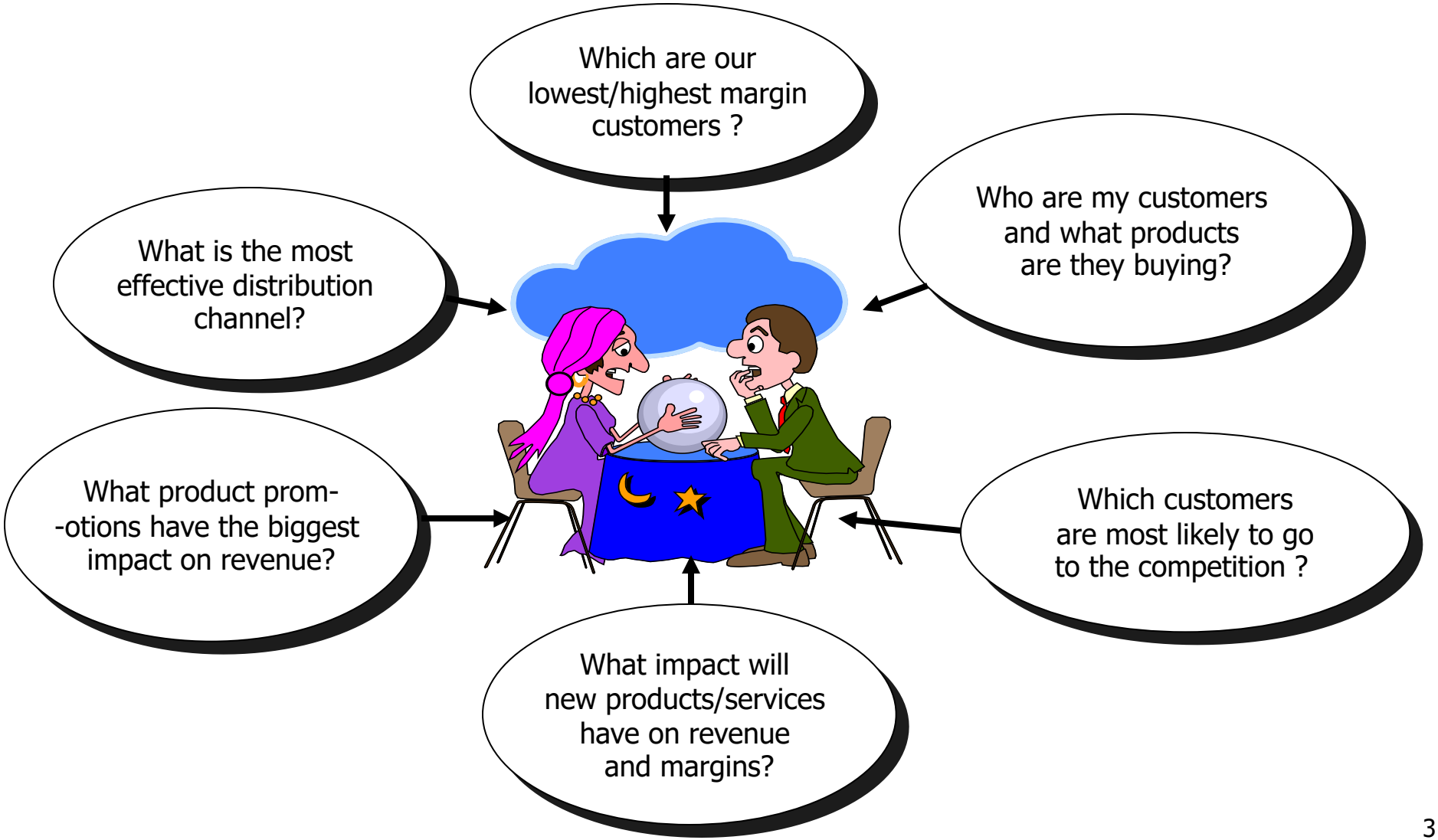
(Computer Engineering and Technology)
S.Y.B. Tech. CSE (AIDS)
Data Engineering Techniques

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

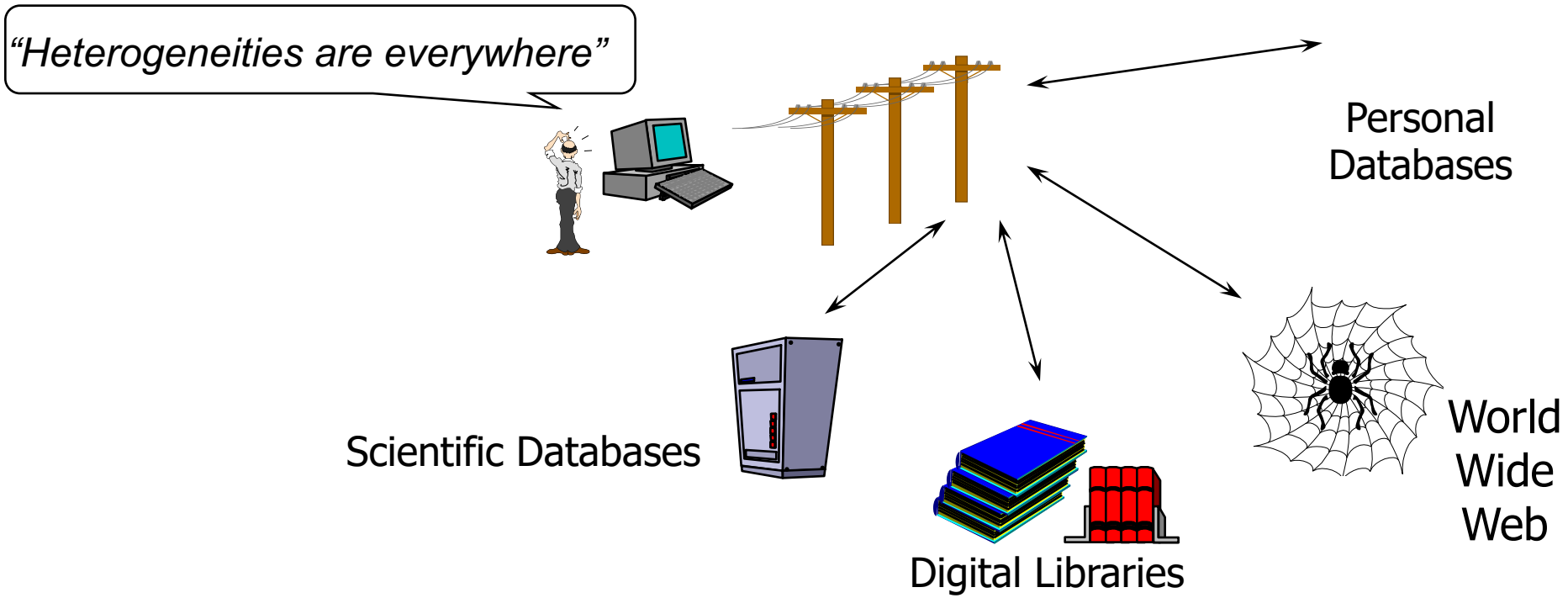
UNIT II-Syllabus on Data Warehouse

- **Unit II: DATA WAREHOUSE**
- Introduction to Data Warehouse
- OLTP and OLAP
- Data Warehouse architecture
- Data Warehouse Modeling: A Multidimensional Data Model
- Data Warehouse Design: Stars, Snowflakes, and Fact Constellations Schemas
- Dimensions: The Role of Concept Hierarchies
- Measures: Their Categorization and Computation
- Typical OLAP Operations, ROLAP, MOLAP,
- Materialized views
- Integration of Data Warehouse with other technologies

Scenario:A producer wants to know....

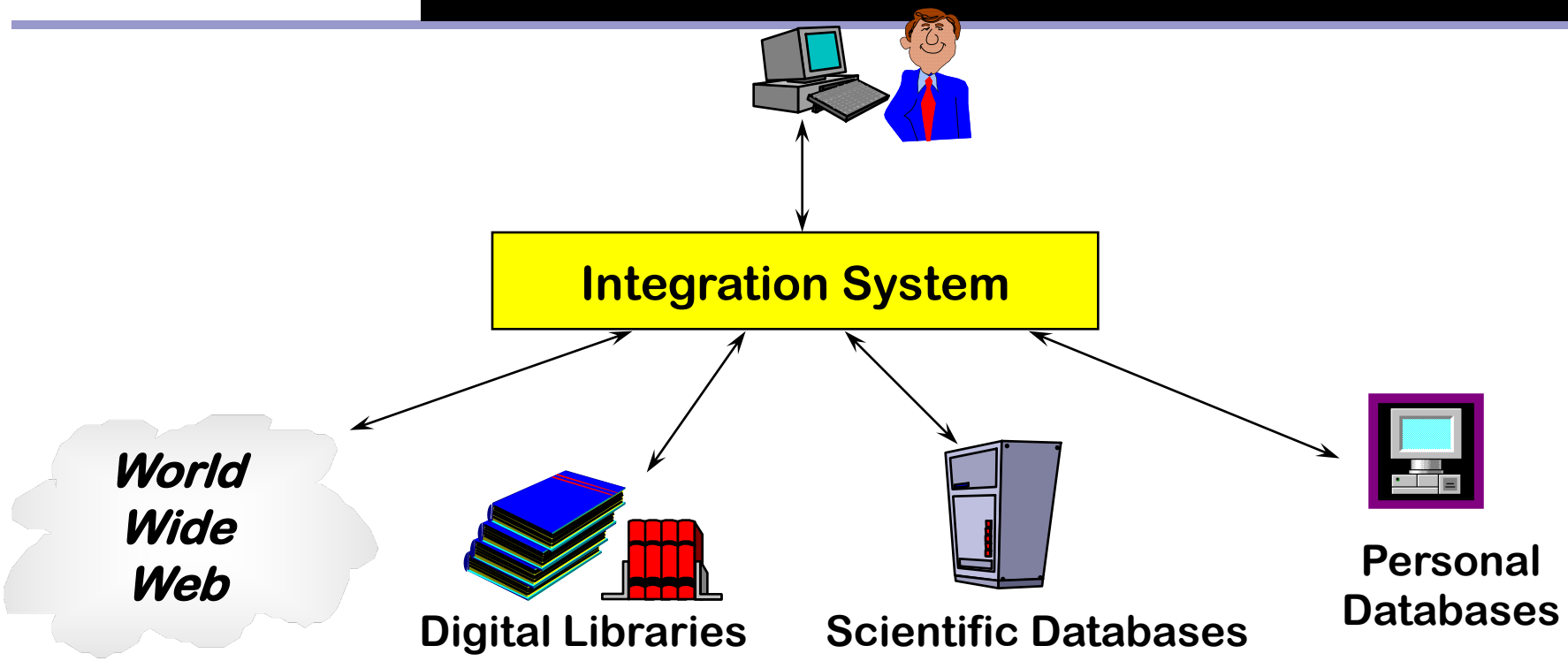


Problem: Heterogeneous Information Sources



- Different interfaces
- Different data representations
- Duplicate and inconsistent information

Unified Access to Data



- Collects and combines information
- Provides integrated view, uniform user interface
- Supports sharing

Data, Data everywhere yet ...

- Can't find the data needed
  - data is scattered over the network
  - many versions, subtle differences
- Can't get the data needed
  - need an expert to get the data
- Can't understand the data found
  - available data poorly documented
- Can't use the data found
  - results are unexpected
  - data needs to be transformed from one form to other

What is Data Warehouse?

- A single, complete and consistent store of data obtained from a variety of different sources, made available to end users in a way they can understand and use in a business context. [Barry Devlin]
- A decision support database that is maintained separately from the organization's operational databases.
- 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

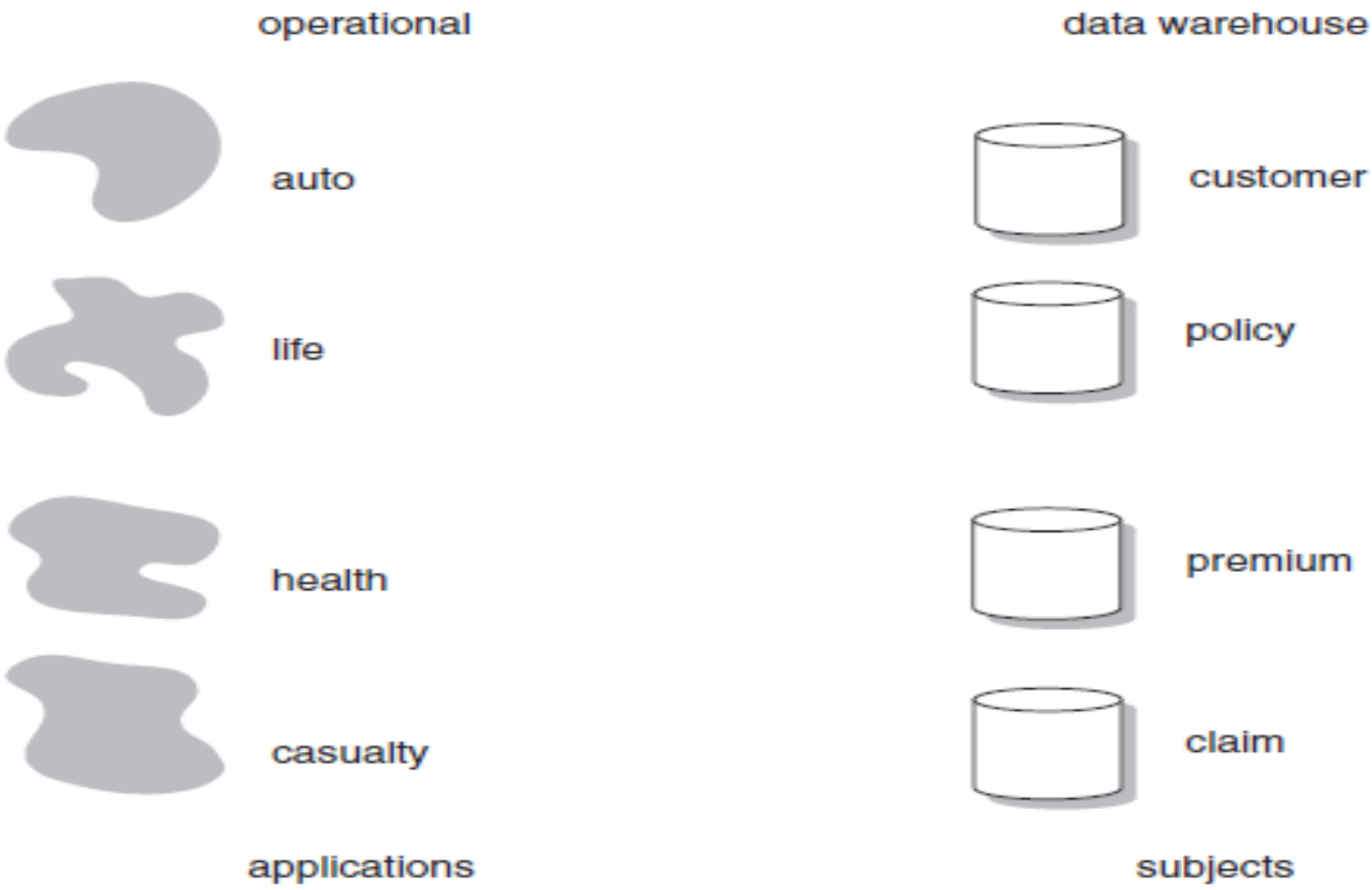
Data warehousing is the process of collecting, integrating, storing and managing data from multiple sources in a central repository. It enables organizations to organize large volumes of current and historical data for efficient querying, analysis and reporting.

Note: *The main goal of data warehousing is to support decision-making by providing clean, consistent and timely access to data. It ensures fast data retrieval even when working with massive datasets.*

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

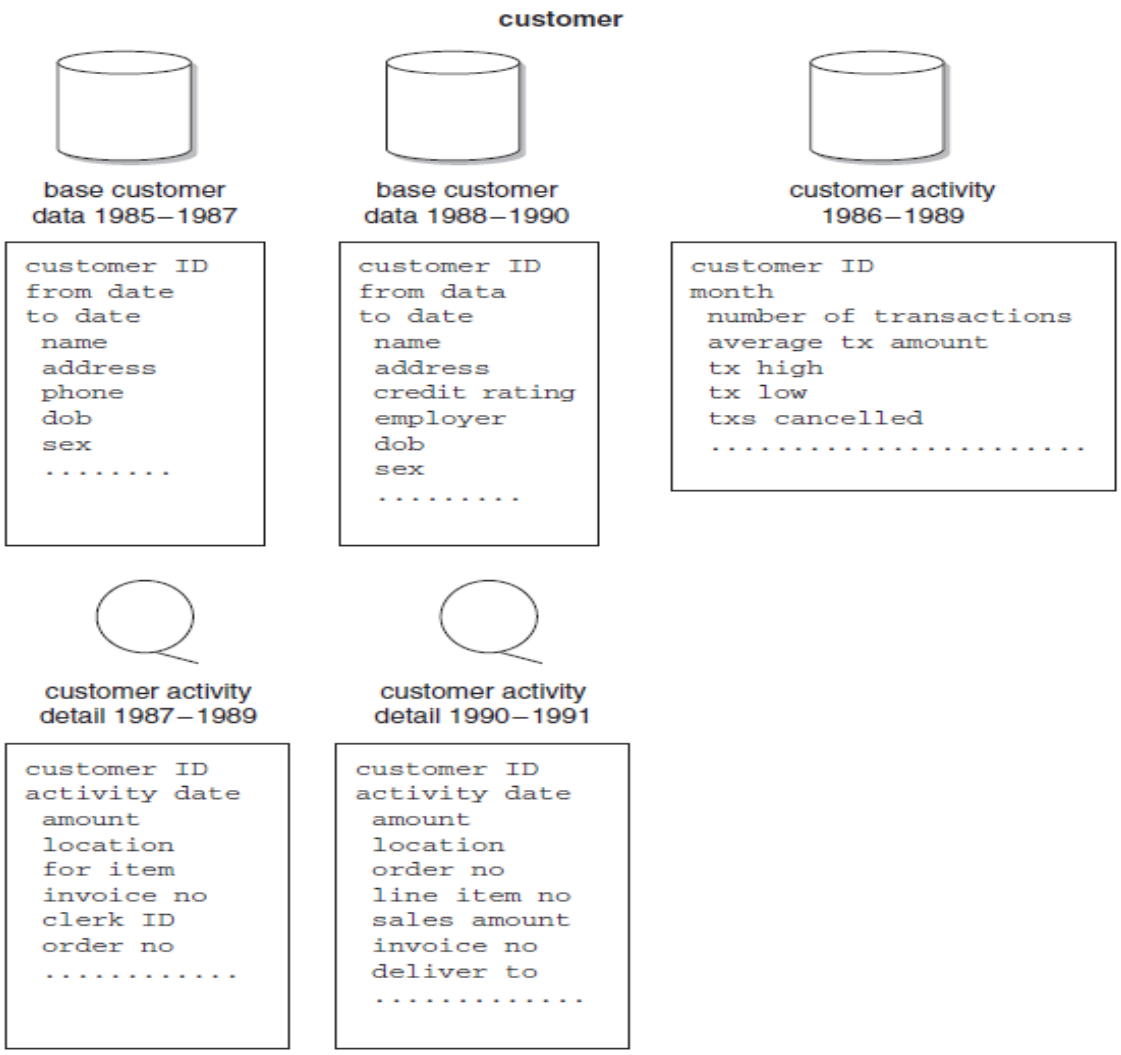
Data Warehouse -Subject-Oriented



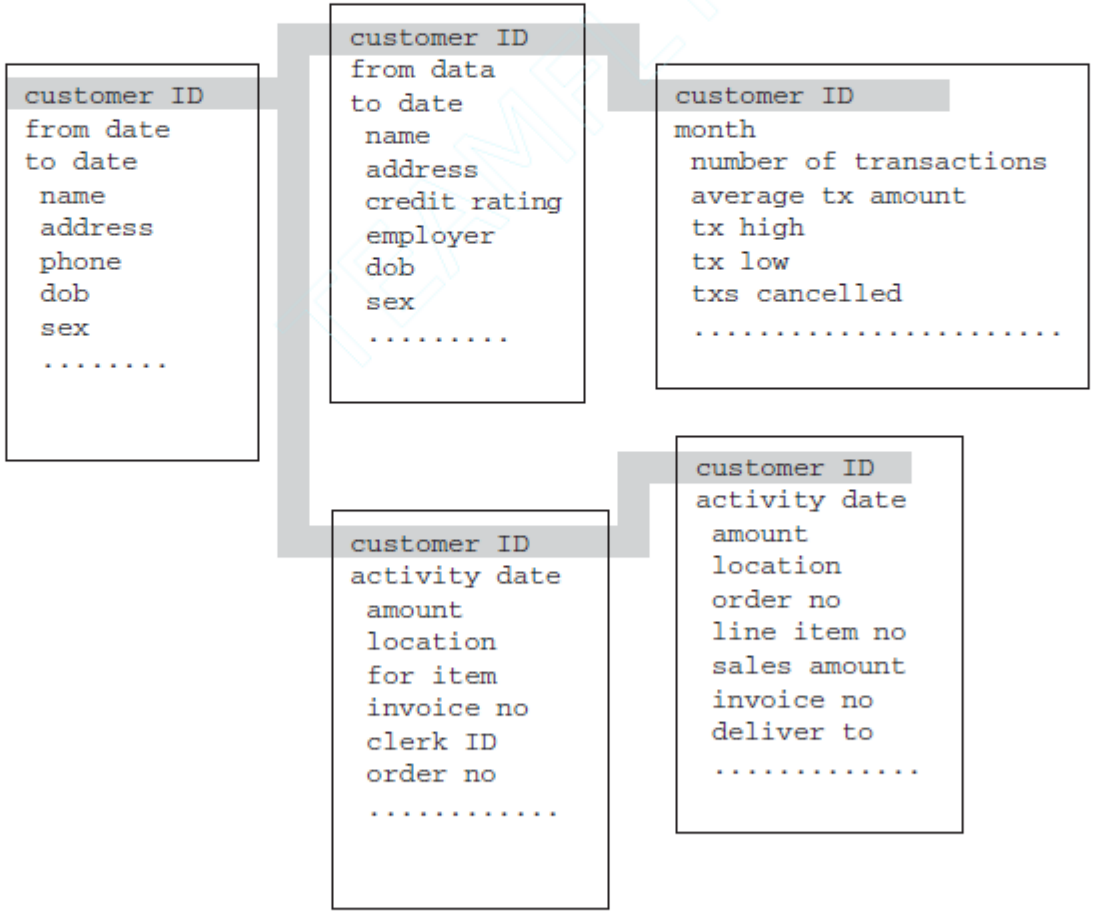
Example : Subject area—customer

- There is a base table for customer information as defined from 1985 to 1987.
- There is another for the definition of customer data between 1988 and 1990.
- There is a cumulative customer activity table for activities between 1986 and 1989.
- Each month a summary record is written for each customer record based on customer activity for the month.
- There are detailed activity files by customer for 1987 through 1989 and another one for 1990 through 1991.
- The definition of the data in the files is different, based on the year.
- All of the physical tables for the customer subject area are related by a common key.
- Figure shows that the key—customer ID—connects all of the data

Example : Subject area—customer



Example : Subject area—customer



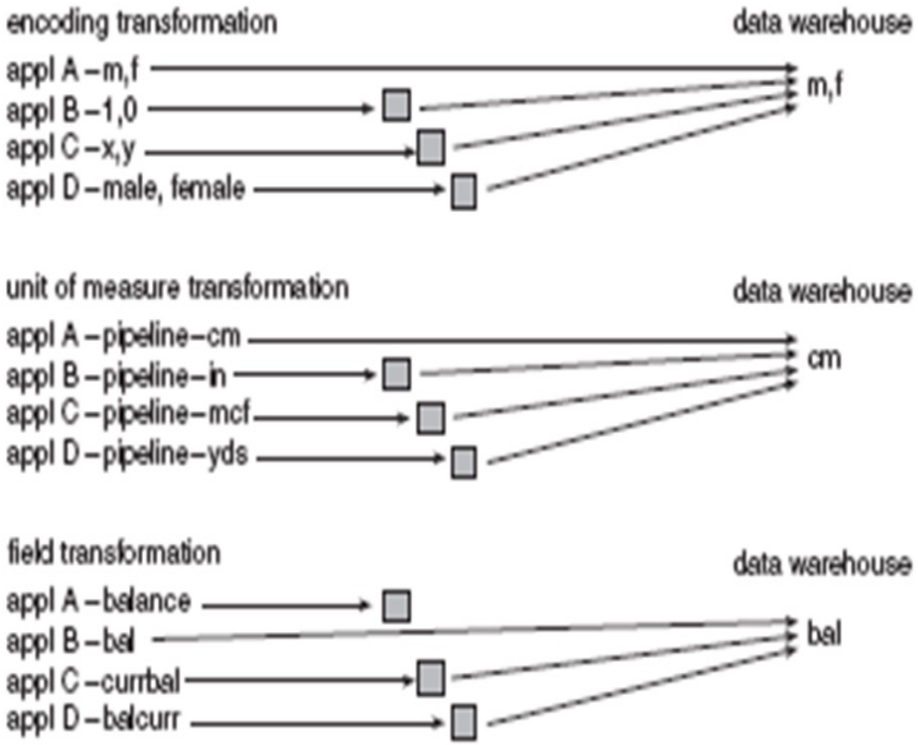
Data Warehouse—Integrated

- 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

- Across multiple applications there is no application consistency in encoding, naming conventions, physical attributes, measurement of attributes, and so forth.
- Each application designer has had free rein to make his or her own design decisions.

Data Warehouse—Integrated



Data Warehouse—Time Variant

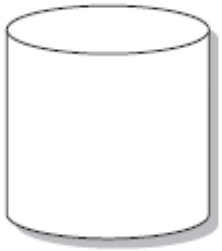
- The time horizon for the data warehouse is significantly longer than that of operational systems
  - Operational database: current value data
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain "time element"

Data Warehouse—Time Variant

- Different environments have different time horizons. A time horizon is the parameters of time represented in an environment.
- The collective time horizon for the data found inside a data warehouse is significantly longer than that of operational systems.
- A 60-to-90-day time horizon is normal for operational systems;
- A 5-to-10-year time horizon is normal for the data warehouse.
- As a result of this difference in time horizons, the data warehouse contains *much* more history than any other environment

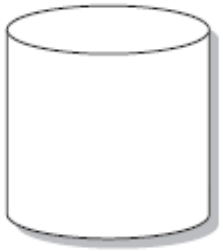
Data Warehouse—Time Variant

operational



- time horizon—current to 60–90 days
- update of records
- key structure may/may not contain an element of time

data warehouse



- time horizon—5–10 years
- sophisticated snapshots of data
- key structure contains an element of time

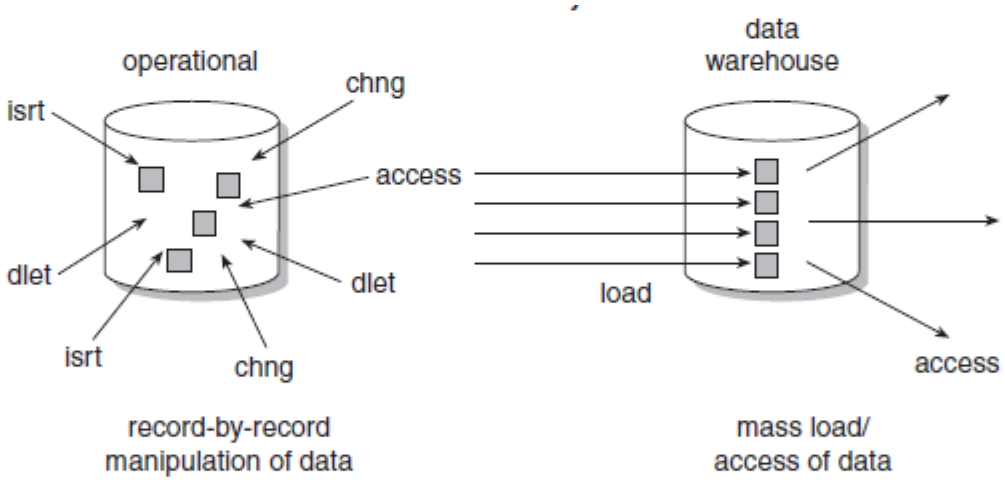
Data Warehouse—Nonvolatile

- 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—Nonvolatile

- Operational data is regularly accessed and manipulated one record at a time.
- Data is updated in the operational environment as a regular matter of course, but data warehouse data exhibits a very different set of characteristics.
- Data warehouse data is loaded (usually en masse) and accessed, but it is not updated (in the general sense).
- Instead, when data in the data warehouse is loaded, it is loaded in a snapshot, static format.
- When subsequent changes occur, a new snapshot record is written.
- In doing so a history of data is kept in the data warehouse.

Data Warehouse—Nonvolatile

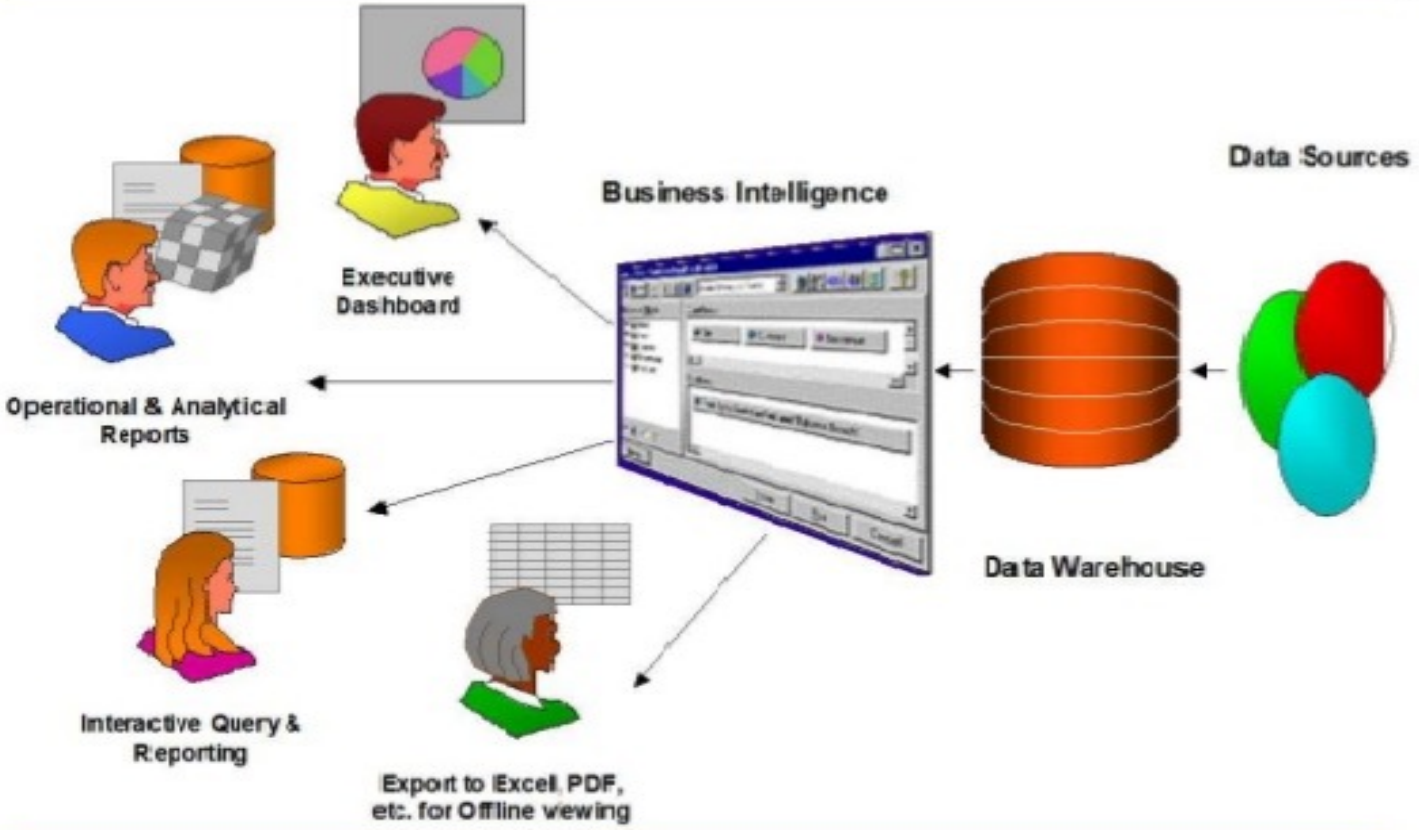


Data Warehouse

Applications

Industry	Application
Finance	Credit card Analysis
Insurance	Claims, Fraud Analysis
Telecommunication	Call record Analysis
Transport	Logistics management
Consumer goods	Promotion Analysis

Data Warehouse:Users



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.

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

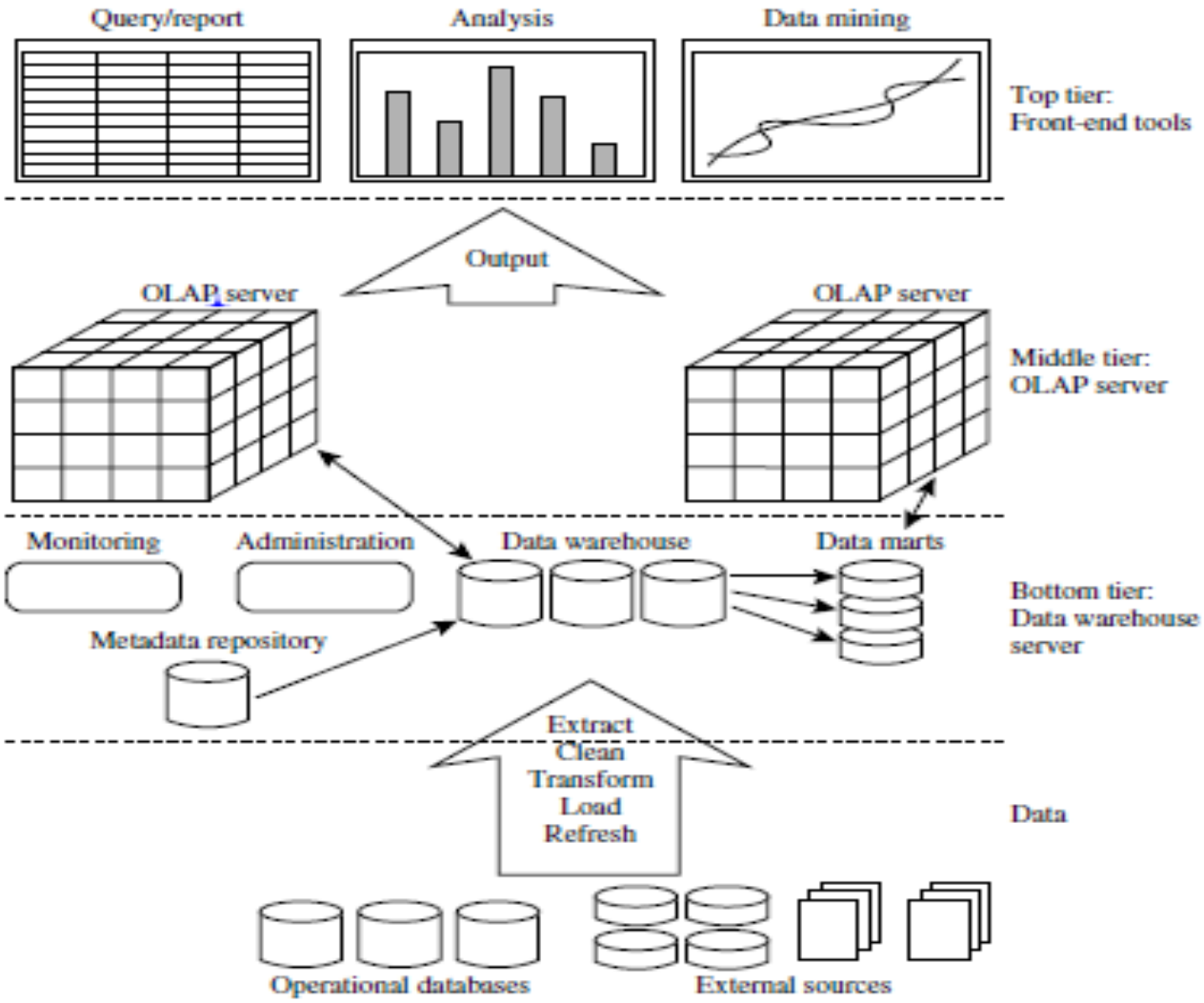
OLTP vs. OLAP

	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 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
- 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

DataWarehousing: A Three-tire Architecture



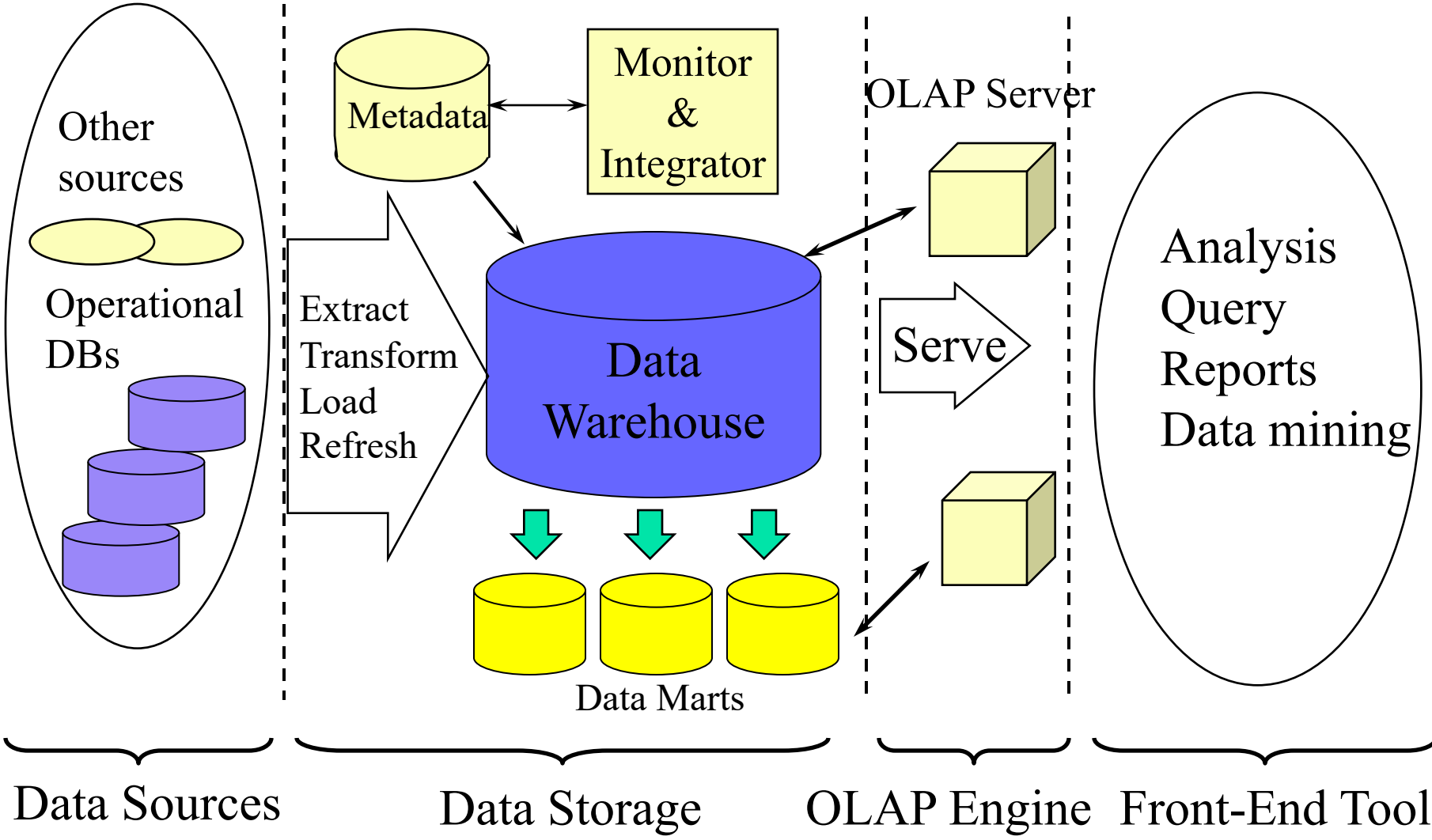
Data Warehouse Models

- 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
  - Cloud-based data warehouse
  - They can typically perform complex analytical queries much faster because they are massively parallel processing (MPP).

DW vs Data Mart

DATA WAREHOUSE	DATA MART
♦ Corporate/Enterprise-wide	♦ Departmental
♦ Union of all data marts	♦ A single business process
♦ Data received from staging area	♦ Star-join (facts & dimensions)
♦ Queries on presentation resource	♦ Technology optimal for data access and analysis
♦ Structure for corporate view of data	♦ Structure to suit the departmental view of data
♦ Organized on E-R model	

Data Warehouse: A Multi-Tiered Arhitecture



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 indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

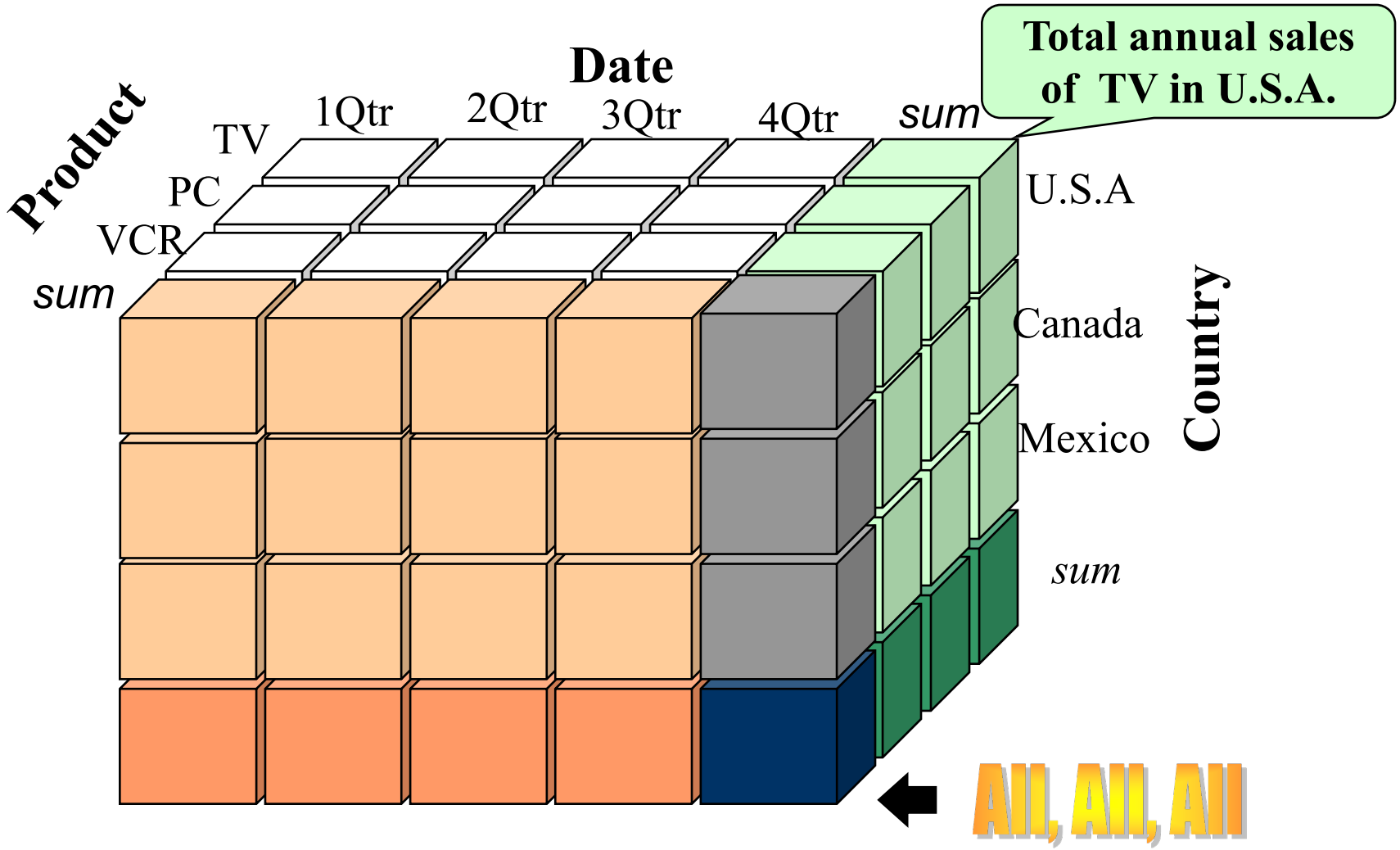
Metadata Repository

- 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

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.

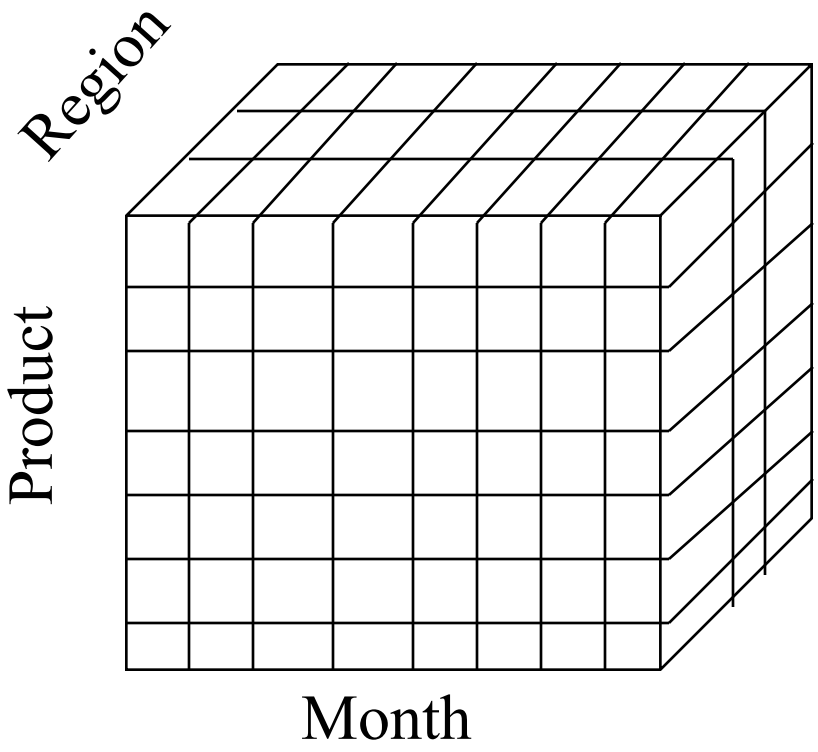
A Sample Data Cube



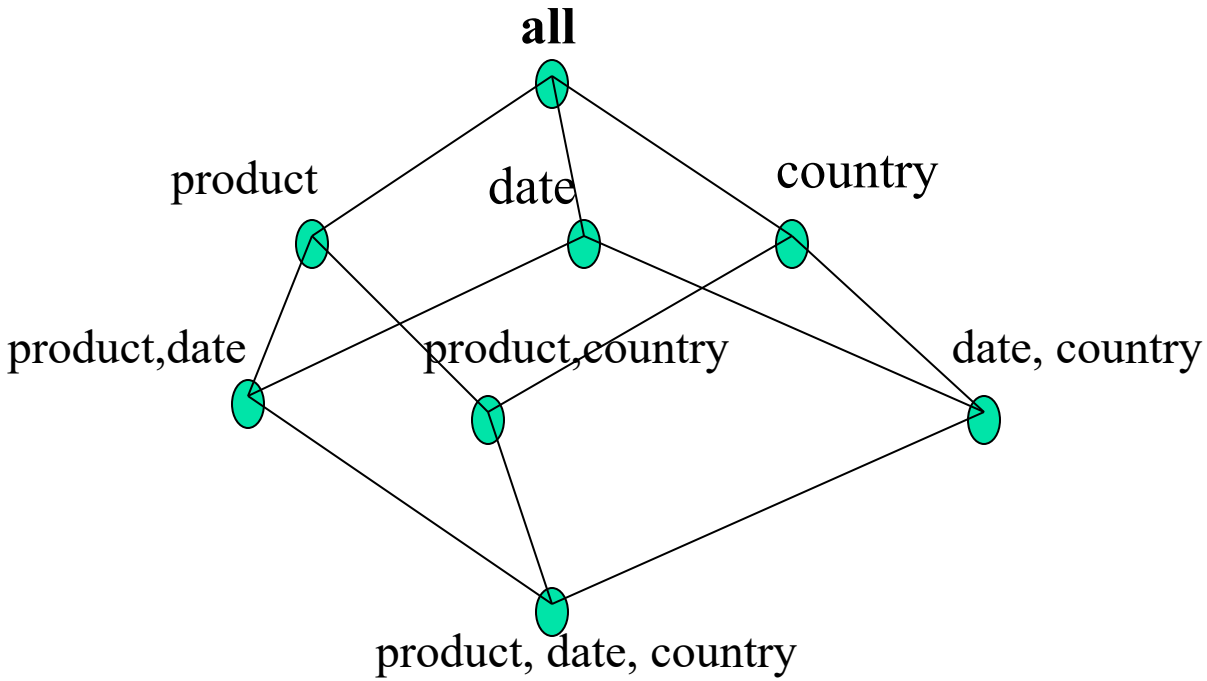
Multidimensional Data

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

Dimensions: Product, Location, Time



Cuboids Corresponding to the Cube



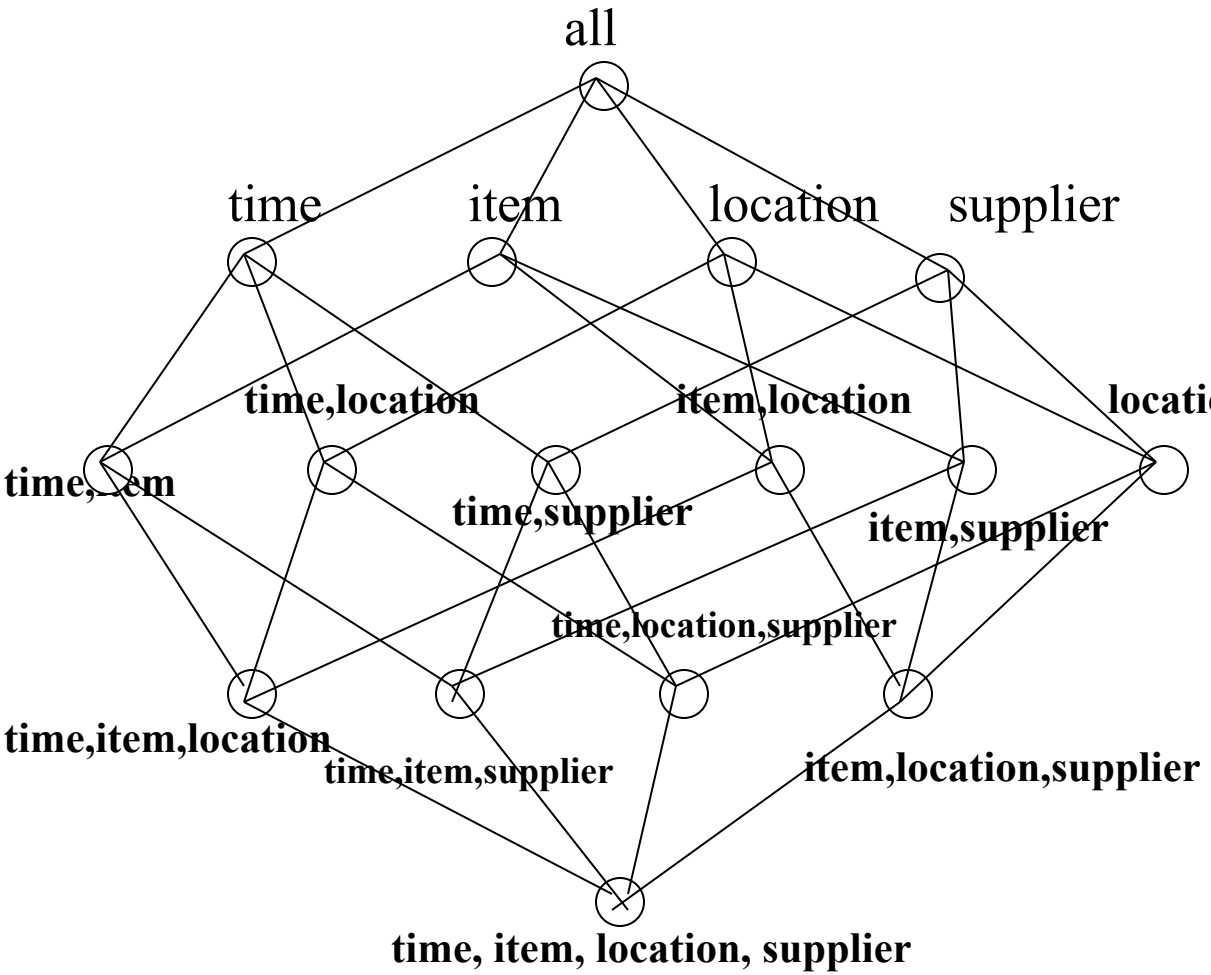
0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D(base) cuboid

Cube: A Lattice of Cuboids



0-D(apex) cuboid

1-D cuboids

2-D cuboids

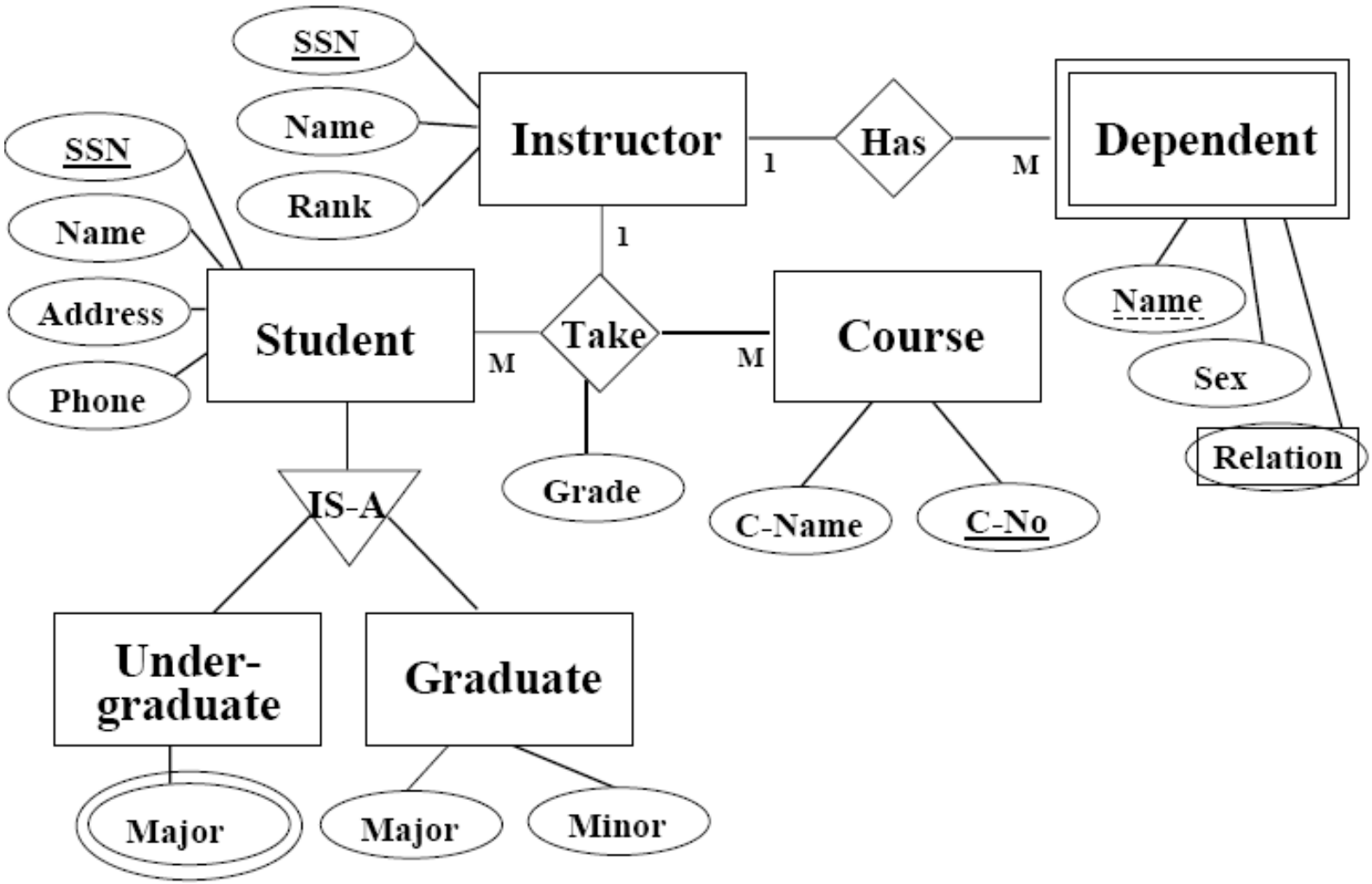
3-D cuboids

4-D(base) cuboid

ER Modelling

- A logical design technique that seeks to eliminate data redundancy
- Illuminates the microscopic relationships among data elements
- Perfect for OLTP systems
- Responsible for success of transaction processing in Relational Databases

Transactional ER system is not user friendly for retrieving data



Problems with ER Model

ER models are NOT suitable for DW?

- End user cannot understand or remember an ER Model
- Many DWs have failed because of overly complex ER designs
- Not optimized for complex, ad-hoc queries
- Data retrieval becomes difficult due to normalization
- Browsing becomes difficult

Design Requirements

- Design of the DW must directly reflect the way the managers look at the business
- Should capture the measurements of importance along with parameters by which these parameters are viewed
- Must facilitate data analysis, i.e., answering business questions

Dimensional nature of business data

- Managers think of business in terms of business dimensions
- **Marketing vice president** is interested in revenue numbers broken down by

  -month

  -division

  -customer

  -demographic

  -sales office

  -product version

  -plan.

These are the business dimensions

Dimensional nature of business data

- **Marketing Manager** business dimensions are

  -product

  -product category

  -time(day,week,month)

  -sale district

  -distribution channel

Dimensional nature of business data

- **Financial Controller** business dimensions are

  -budget line

  -time(month,quarter,year)

  -district

  -division

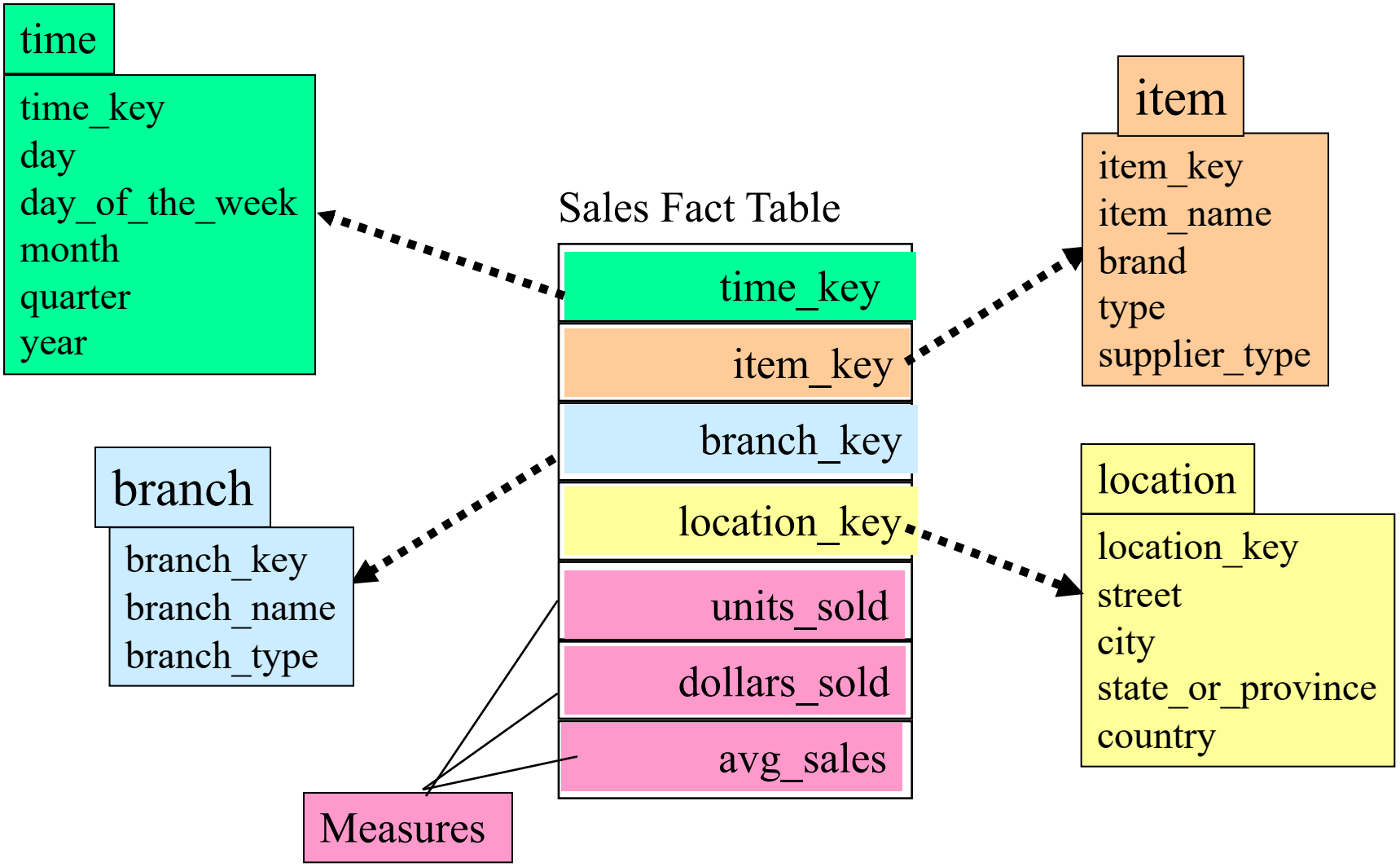
Modeling of Data Warehouses

- 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

Star Schema

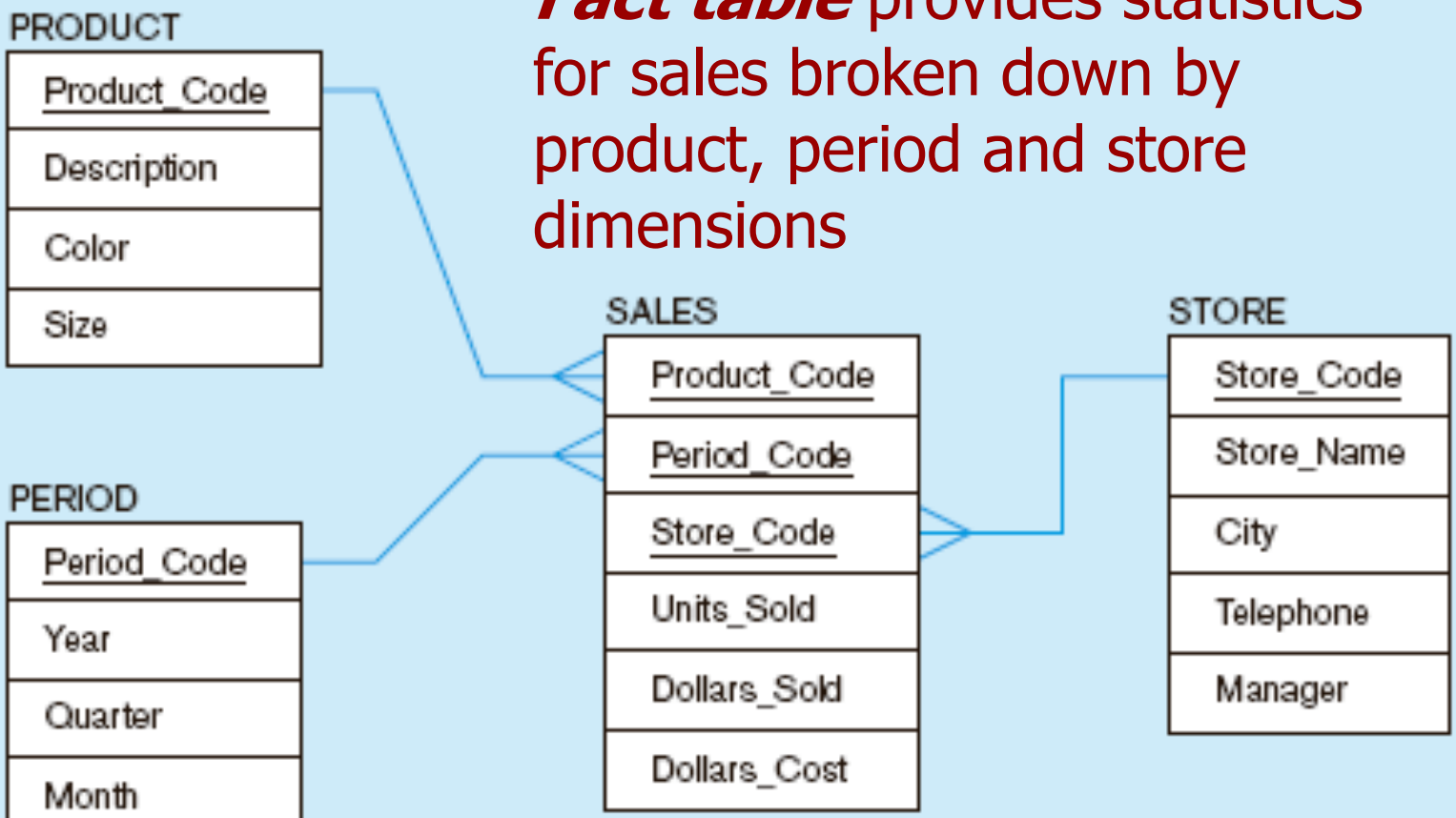
- A single fact table and a single table for each dimension
- Every fact points to one tuple in each of the dimensions and has additional attributes
- Does not capture hierarchies directly
- Generated keys are used for performance and maintenance reasons

Example of Star Schema

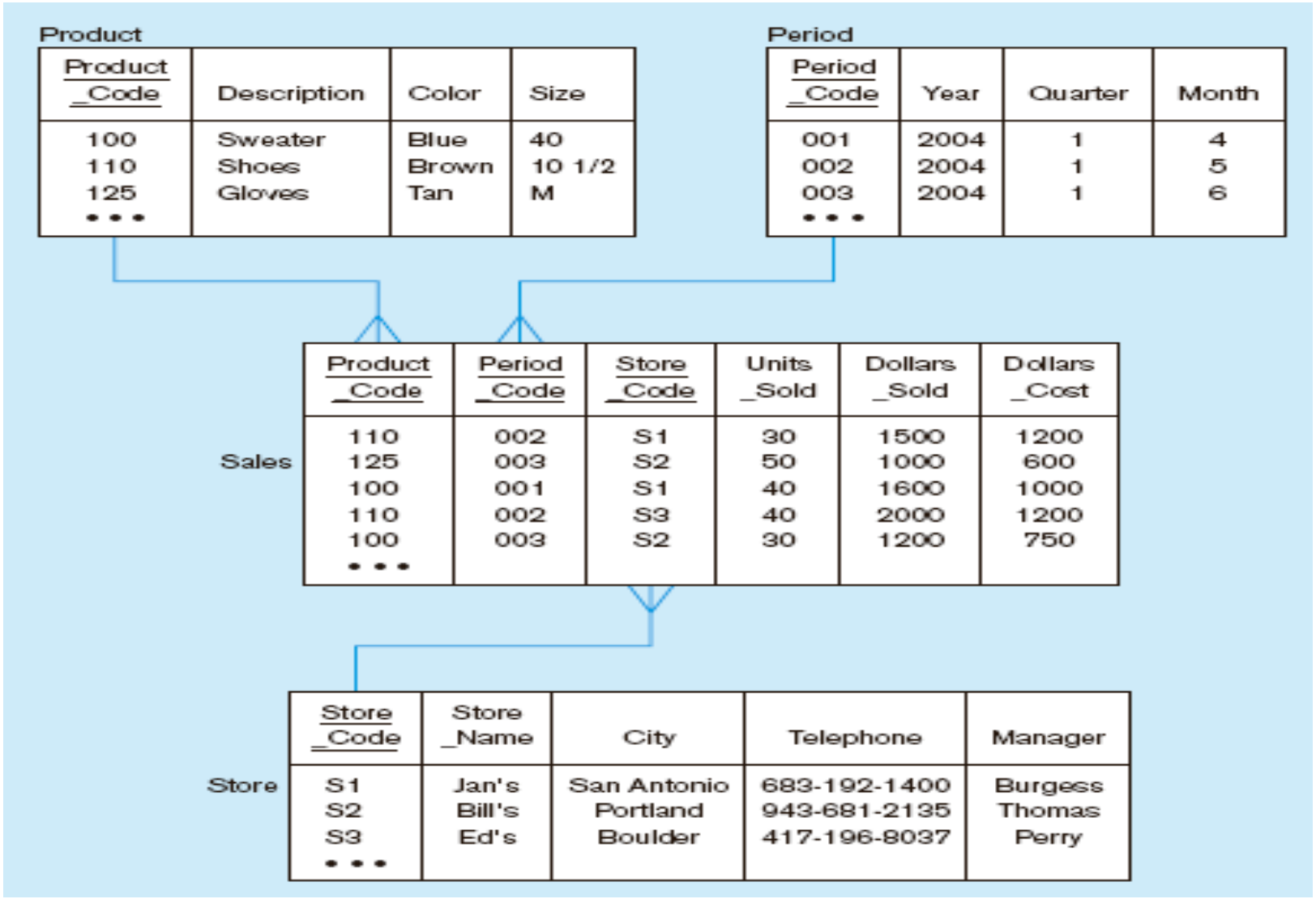


Star schema example

Fact table provides statistics for sales broken down by product, period and store dimensions



Star schema with sample data



STAR Schema keys

- Primary Keys-Uniquely identifies a a record in dimension table

- Surrogate Keys- System generated sequence numbers
  - Do not have any built in meanings.
  - Operational system keys are stored as additional attributes

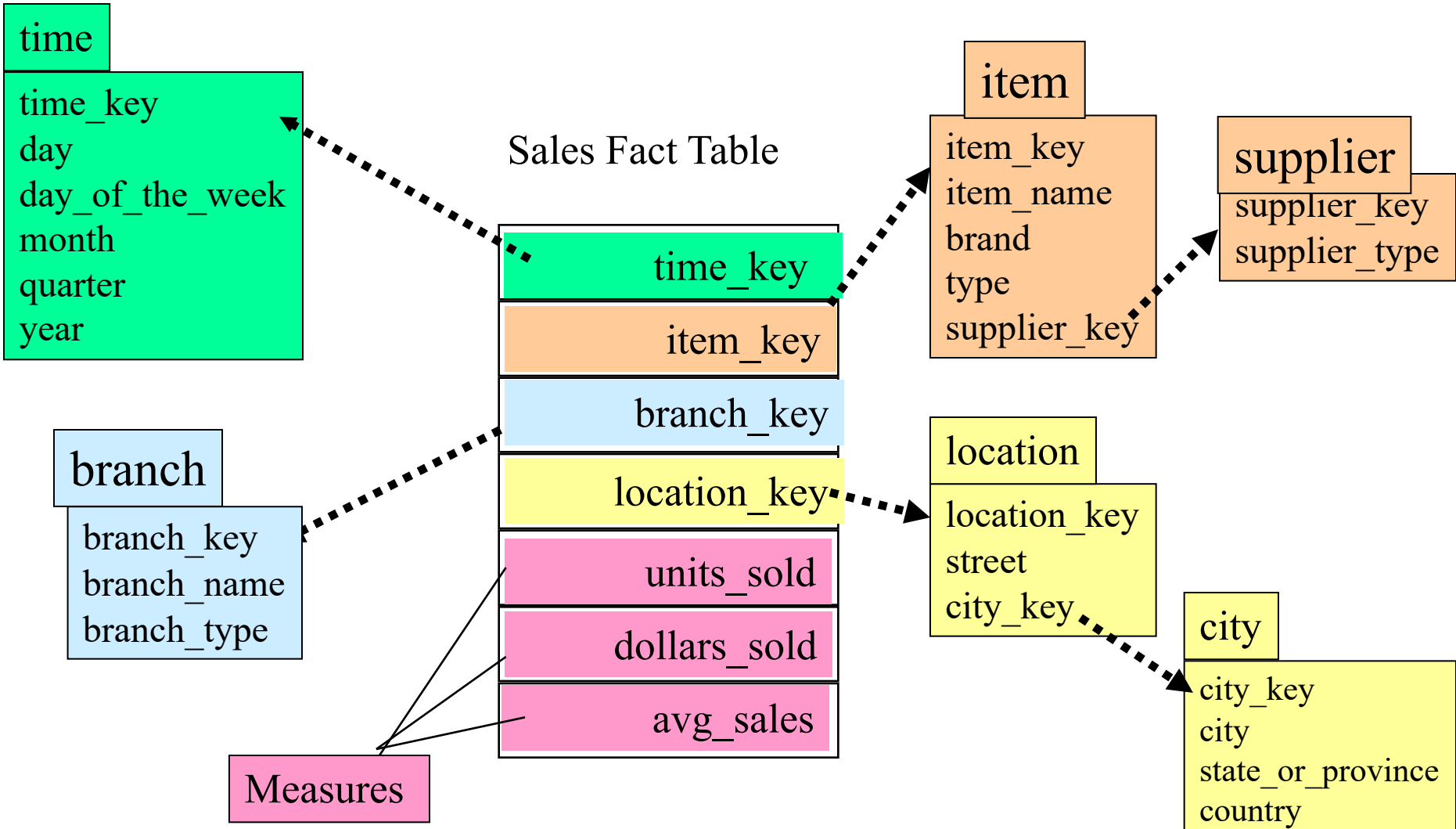
Pros & Cons of STAR Schema

- **Star schema pros**
  - simple design;
  - fast read and queries;
  - easy data aggregation
  - easy integration with OLAP systems and data cubes.
- **Star schema cons**
  - redundant data makes for larger storage on disk;
  - potential for data abnormalities, errors and inconsistencies;
  - slower queries;
  - limited flexibility on non-dimensional data.

Snowflake Schema

- A snowflake schema database is similar to a star schema in that it has a **single fact table and many dimension tables.**
- For a snowflake schema, each dimension table might have foreign keys that relate to other dimension tables.
- **A snowflake schema is more normalized.**
- **To manage the size of derived table.**

Example of Snowflake Schema



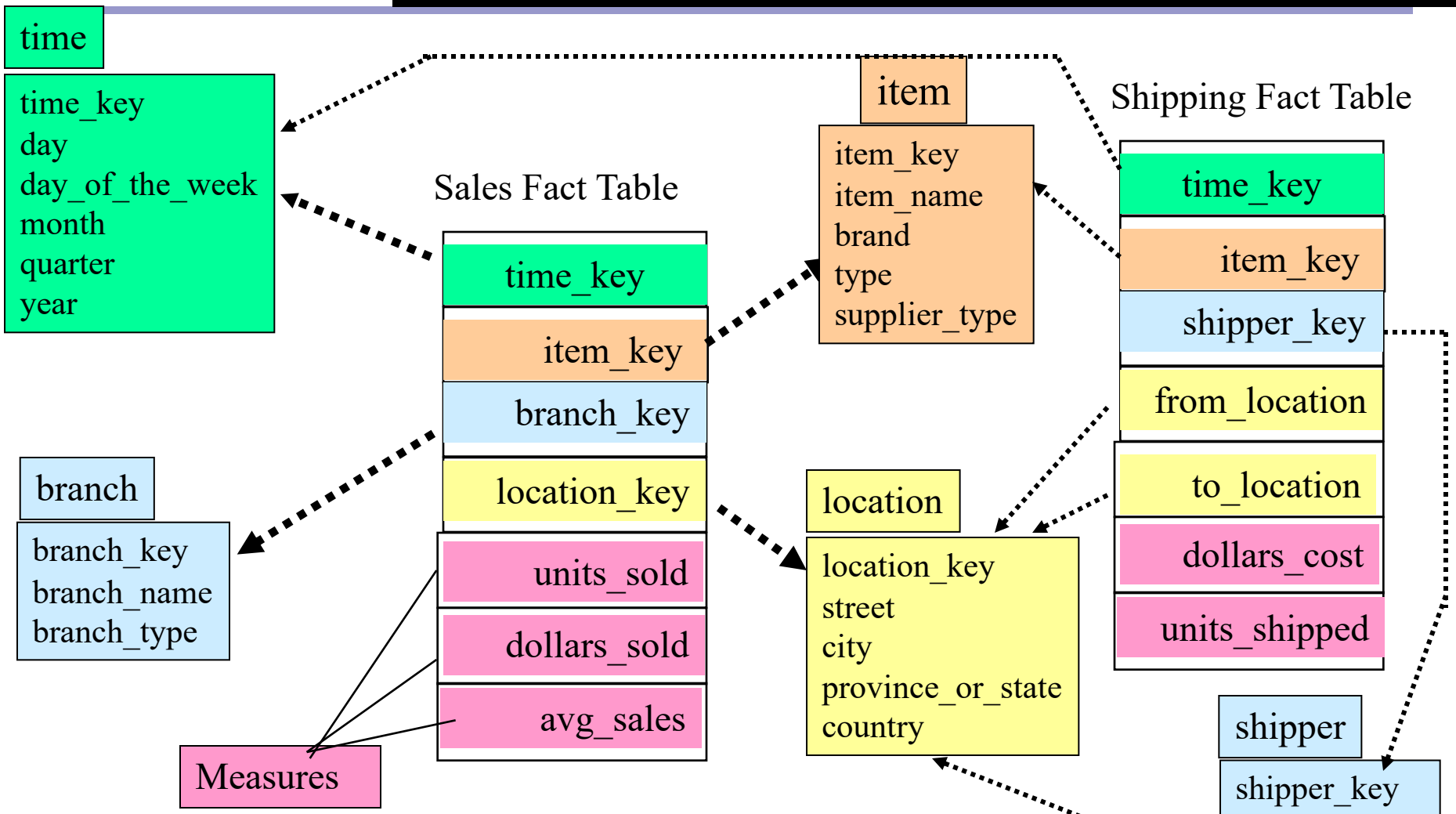
Snowflake Schema

- Advantages
  - Memory save
  - Normalized structures easier to update and maintain
- Disadvantages
  - Increases complexity
  - Browsing difficult
  - Degrades performance because of additional joins

Fact Constellation Schema

- Some applications are complex in nature and require multiple fact tables or sharing of dimension tables.
- This schema is one of the widely used data warehouse design methodology and is also called **Galaxy schema.**

Example of Fact Constellation



Pros & Cons of Fact Constellation Schema

- Pros:
  - It fact constellation schema offers more flexibility.
  - Multiple fact tables are explicitly assigned to dimension tables.
- Cons:
  - The structure is more complex and sophisticated.
  - It is difficult to maintain.
  - The number of aggregations are high in constellation.

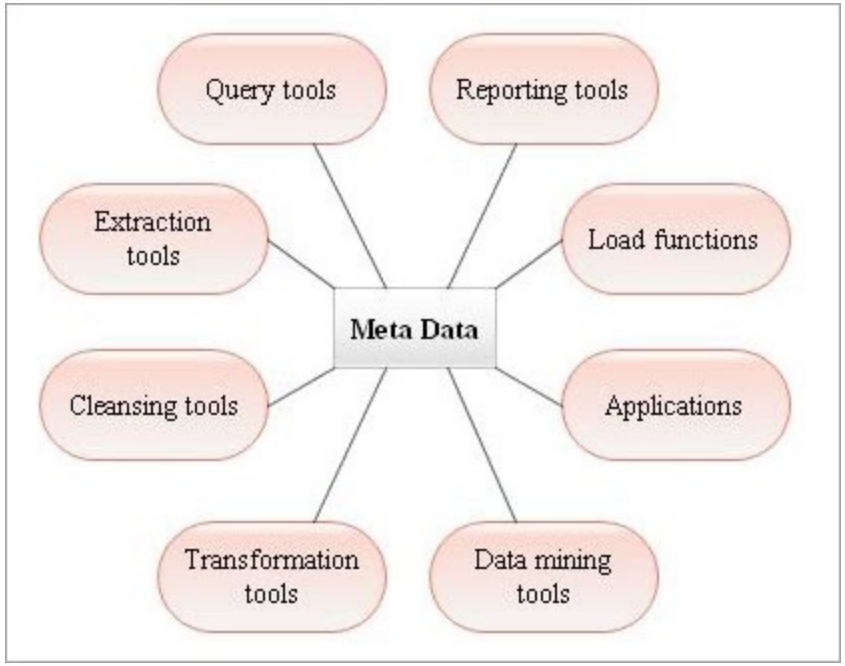
Role of Metadata

- Metadata in a data warehouse is data that describes and contextualizes other data. It can provide information about the content, format, structure, and other characteristics of data. Metadata can help improve the organization, discoverability, and accessibility of data, and is considered the backbone of effective data management
- Data understanding : Metadata can help users understand the data stored within the system. For example, structural metadata can describe how different elements of a compound data object are assembled, such as the page layout of a document or the chapter structure of an ebook.

Role of Metadata

- Data governance : Metadata can help ensure data accuracy and compliance, and enforce the definition of business terms to business end-users.
- Data integration : Metadata can help validate data transformation and ensure the accuracy of calculations.
- Data accessibility : Metadata can help improve the accessibility of data. For example, a data catalog can combine metadata with data management and search tools to help analysts and other data users find data.

Role of Metadata

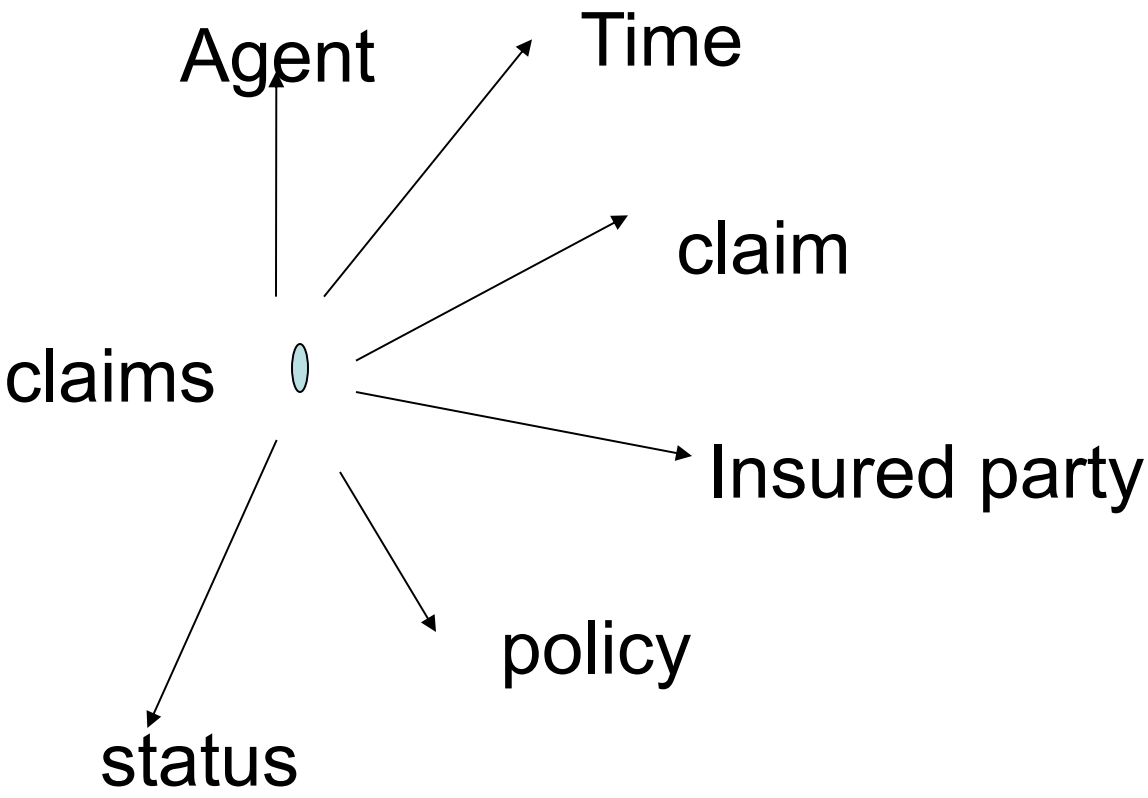


Datawarehouse : business dimensions

- Identify dimensions for building a DW for claim analysis of insurance business.

Snowflake Schema	Galaxy or Fact Constellation Schema
• Extension of star schema that adds additional dimensions	• Splits one star schema into more star schemas
• Called snowflake because its diagram resembles a snowflake	• Schema is viewed as a collection of star hence named as galaxy schema
• One fact table	• Two or more fact tables
• Data splits into different dimension tables	• The schema is helpful for aggregating fact tables for better understanding.
• One fact table surrounded by dimension table which are in turn surrounded by dimension table.	• Dimensions In this schema are separated into separate dimensions based on various levels of hierarchies.

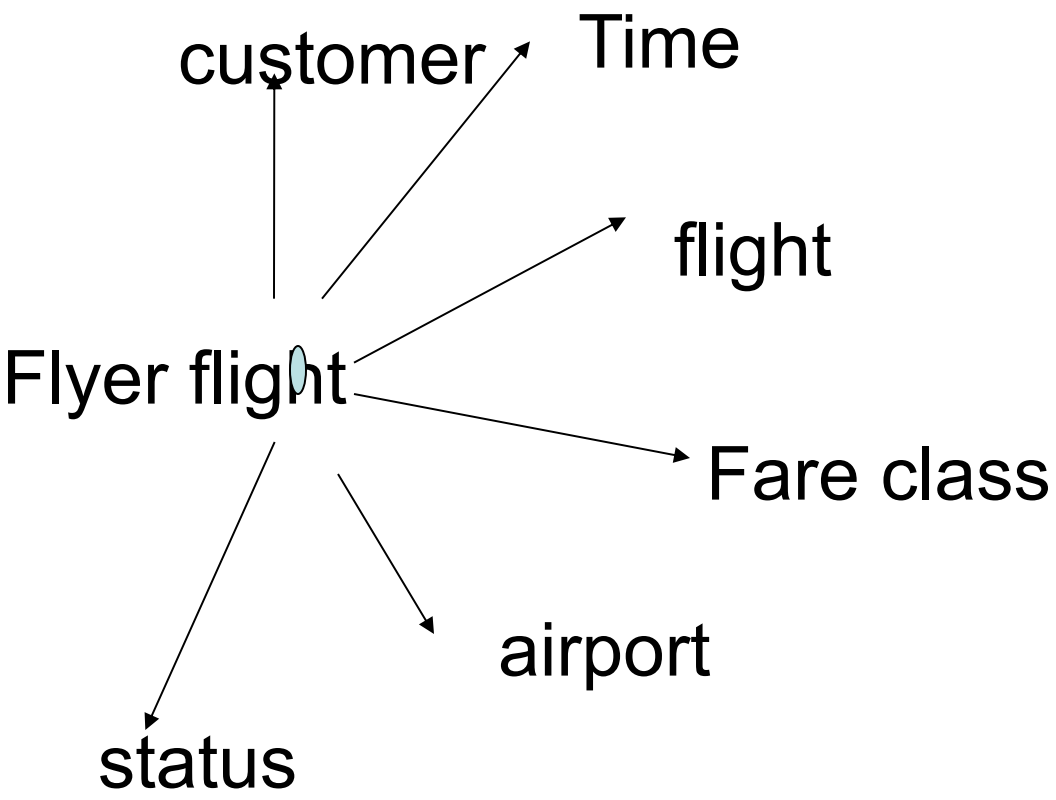
Example of business dimensions



Datawarehouse : business dimensions

- Identify dimensions for building a DW for flyer flight analysis of Airline Company

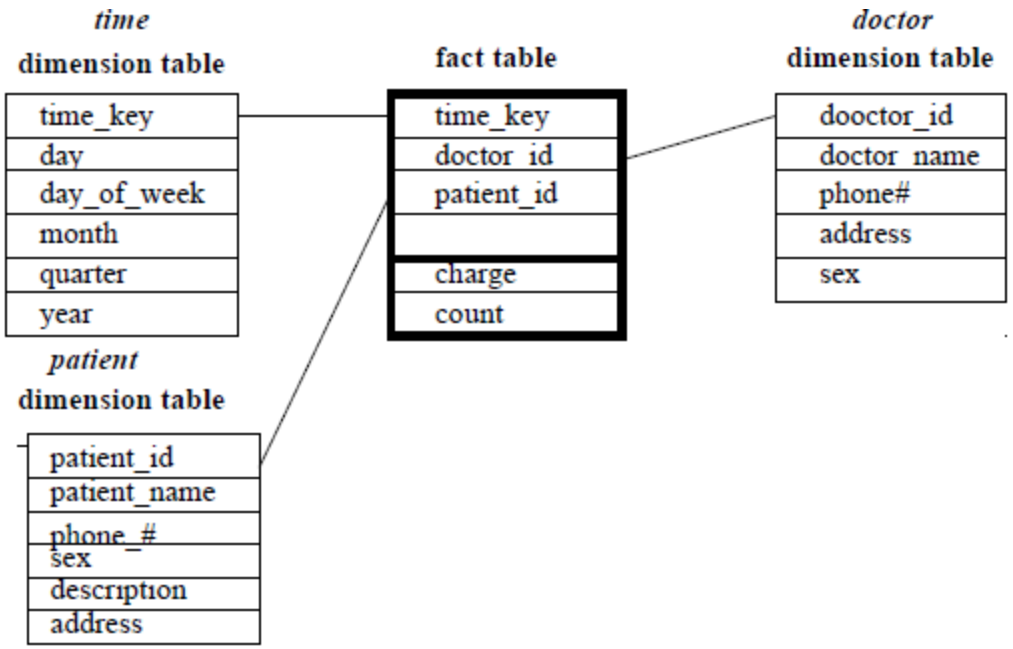
Example of business dimensions



Exercise

Suppose that a data warehouse consists of the three dimensions time, doctor, and patient and the two measures count and charge, where change is that a doctor charges a patient for a visit.

STAR Schema


time
dimension table
time_key
day
day_of_week
month
quarter
year
patient
dimension table
patient_id
patient_name
phone_#
sex
description
address
fact table
time_key
doctor_id
patient_id
charge
count
doctor
dimension table
dooctor_id
doctor_name
phone#
address
sex

Exercise

Suppose that a data warehouse consists of the four dimensions, *date*, *spectator*, *location*, and *game*, and the two measures, *count* and *charge*, where *charge* is the fare that a spectator pays when watching a game on a given date. Spectators may be students, adults, or seniors, with each category having its own charge rate.

(a) Draw a *star schema* diagram for the data warehouse

STAR Schema

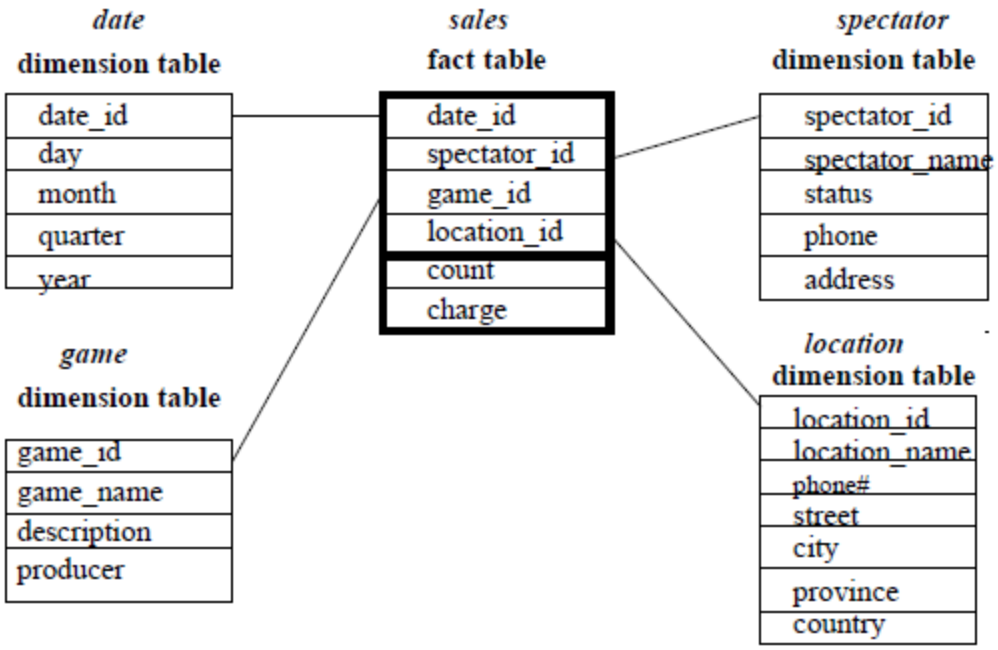
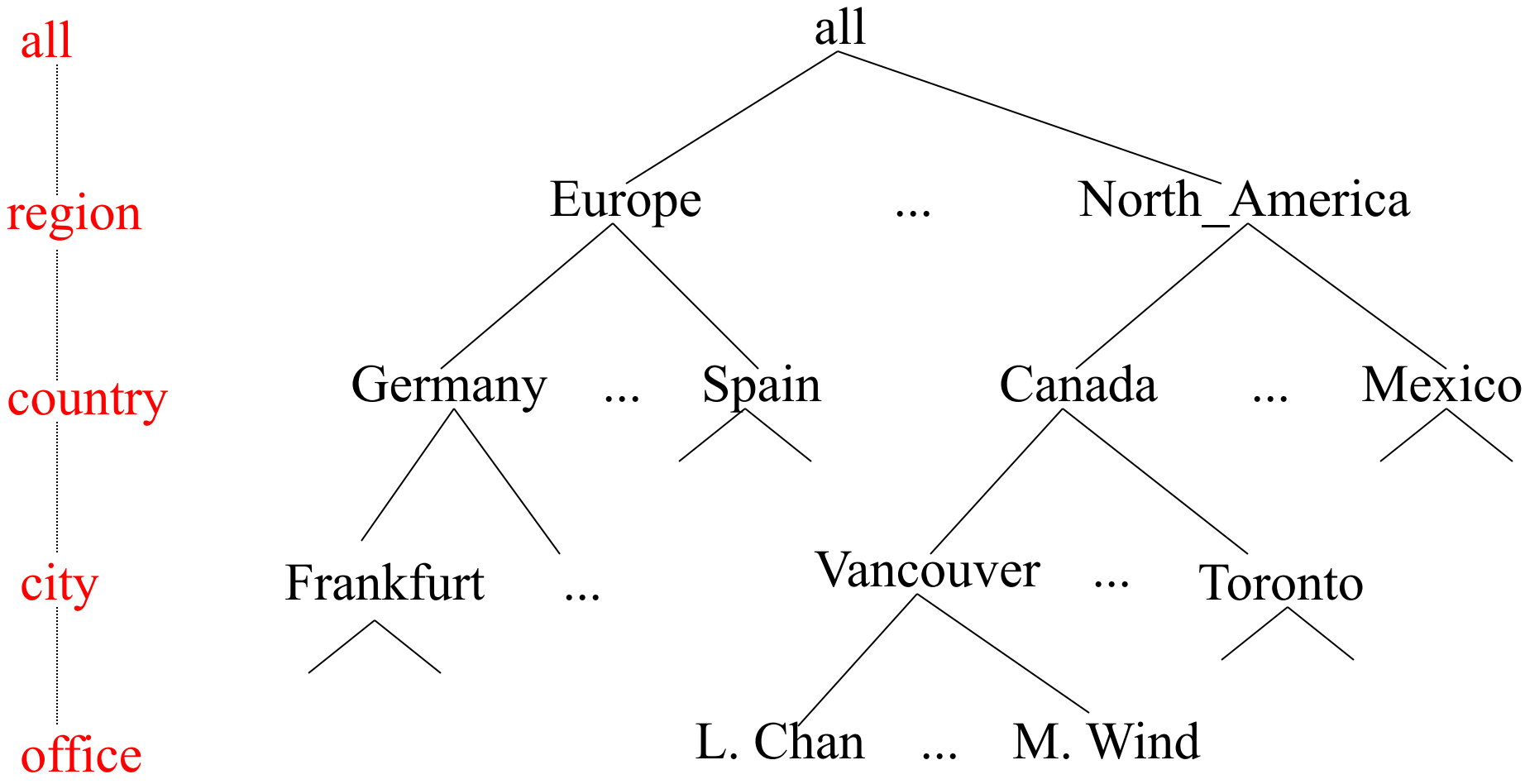


Figure 3.3: A star schema for data warehouse of Exercise 3.5.

Measures of Data Cube: Three Categories

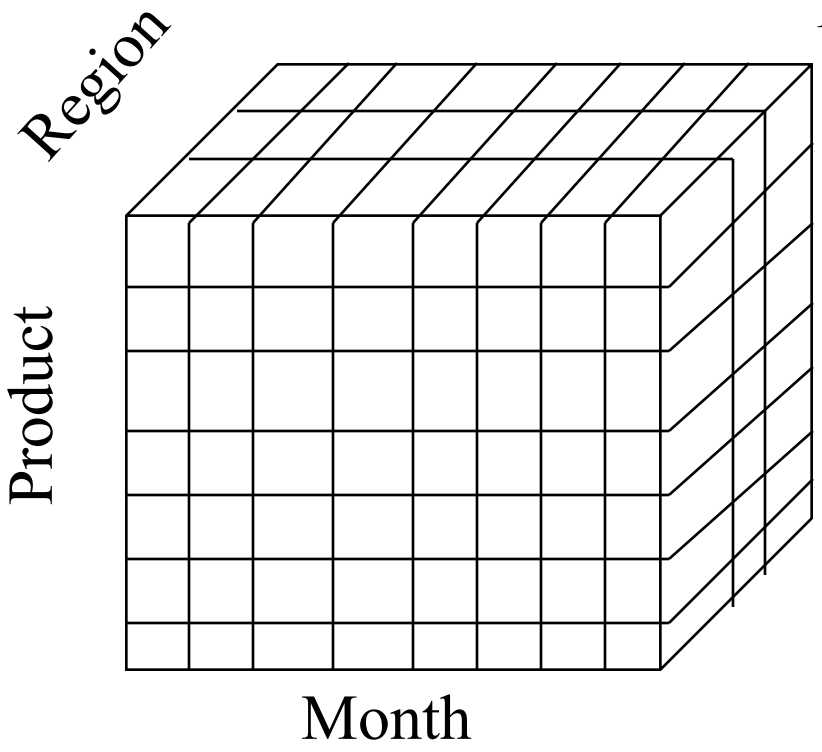
- Distributive: if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning
  - E.g., count(), sum(), min(), max()
- Algebraic: if it can be computed by an algebraic function with *M* arguments (where *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 subaggregate.
  - E.g., median(), mode(), rank()

A Concept Hierarchy: Dimension (location)

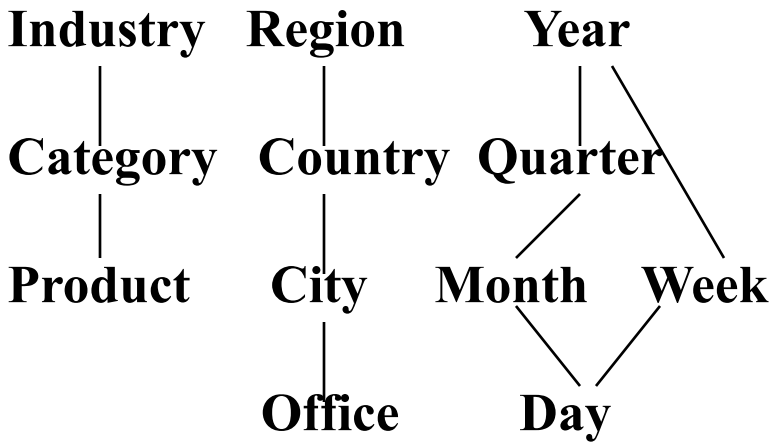


Multidimensional Data

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



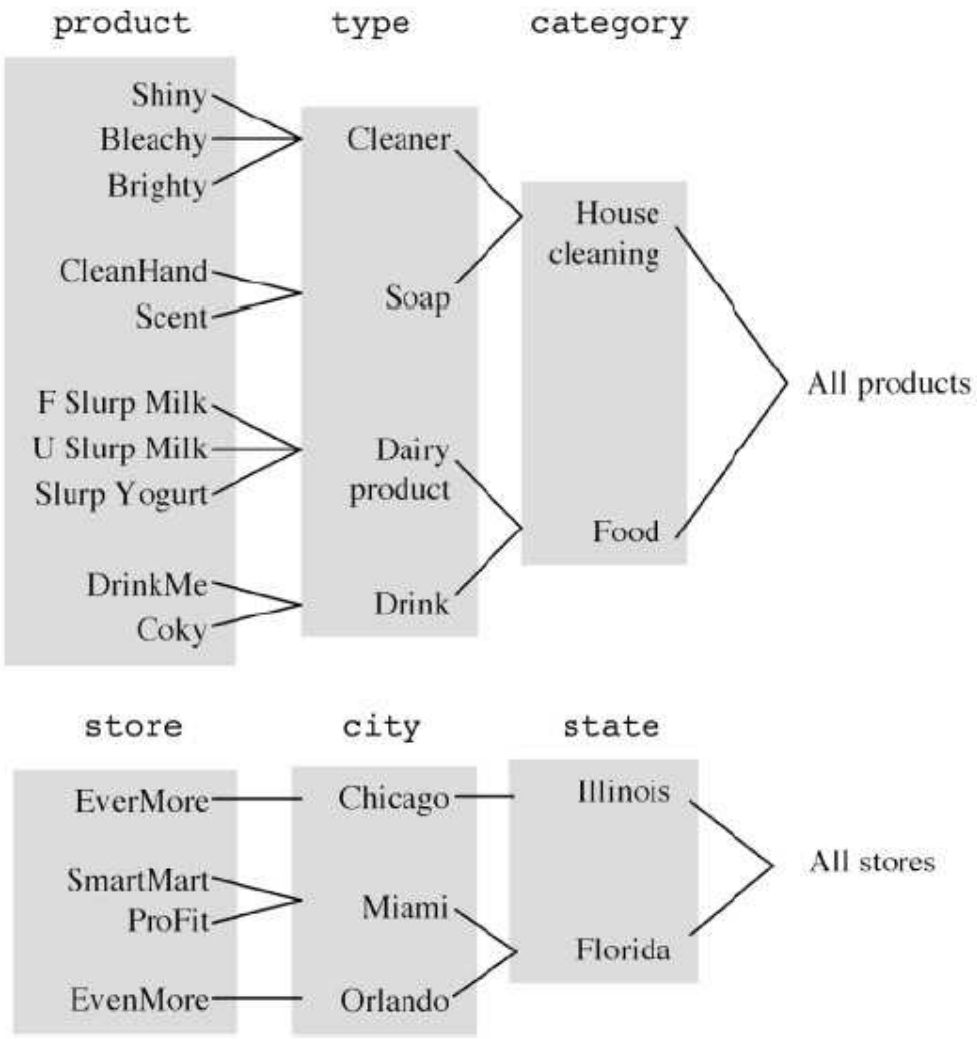
Dimensions: Product, Location, Time
Hierarchical summarization paths



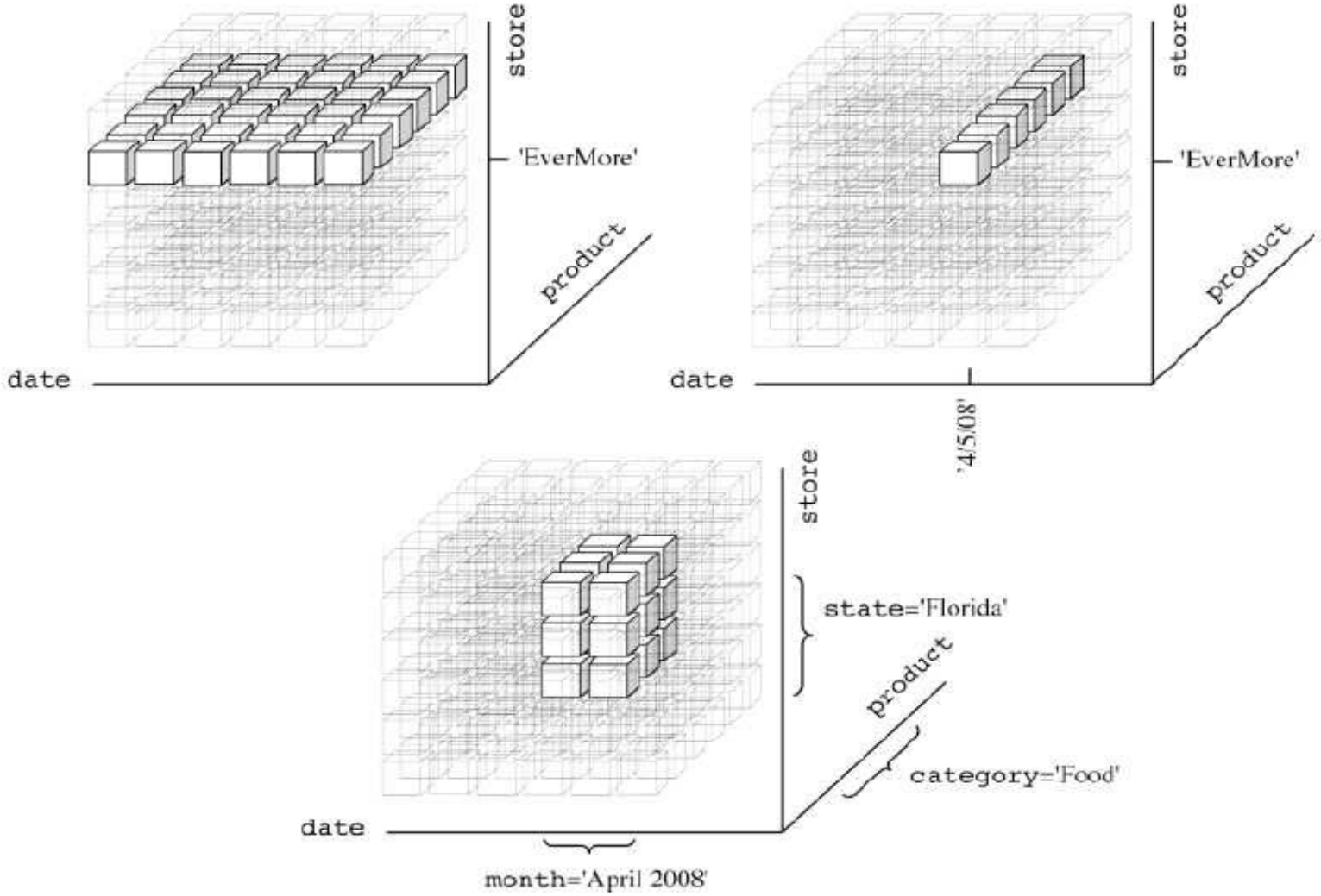
Typical OLAP Operations

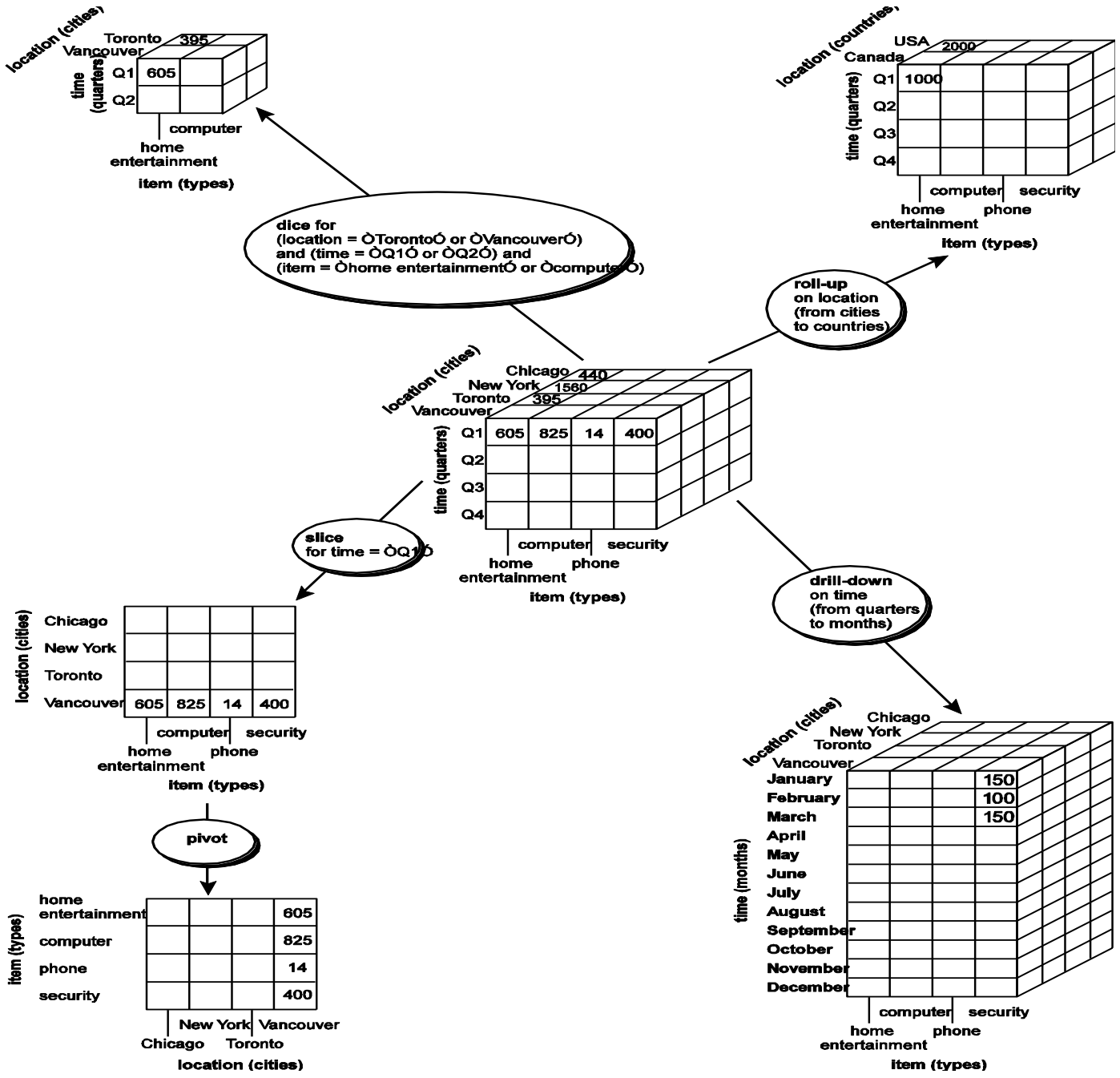
- Roll up (drill-up): summarize data
  - *by climbing up hierarchy or by dimension reduction*
- Drill down (roll down): reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- Slice and dice: *project and select*
- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes*
- 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)*

Roll up (drill-up):



Slice and dice




location (cities)
Toronto 395
Vancouver
time (quarters)
Q1 605
Q2
computer
home entertainment
item (types)
location (countries)
USA 2000
Canada
Q1 1000
Q2
Q3
Q4
time (quarters)
computer security
home entertainment phone
item (types)
dice for
(location = ÒTorontoÓ or ÒVancouverÓ)
and (time = ÒQ1Ó or ÒQ2Ó) and
(item = Òhome entertainmentÓ or Òcomputer Ó)
roll-up
on location
(from cities
to countries)
location (cities)
Chicago 440
New York 1560
Toronto 395
Vancouver
Q1 605 825 14 400
Q2
Q3
Q4
time (quarters)
computer security
home entertainment phone
item (types)
slice
for time = ÒQ1Ó
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 605
computer 825
phone 14
security 400
New York Vancouver
Chicago Toronto
location (cities)
location (cities)
Chicago
New York
Toronto
Vancouver
January 150
February 100
March 150
April
May
June
July
August
September
October
November
December
time (months)
computer security
home entertainment phone
item (types)

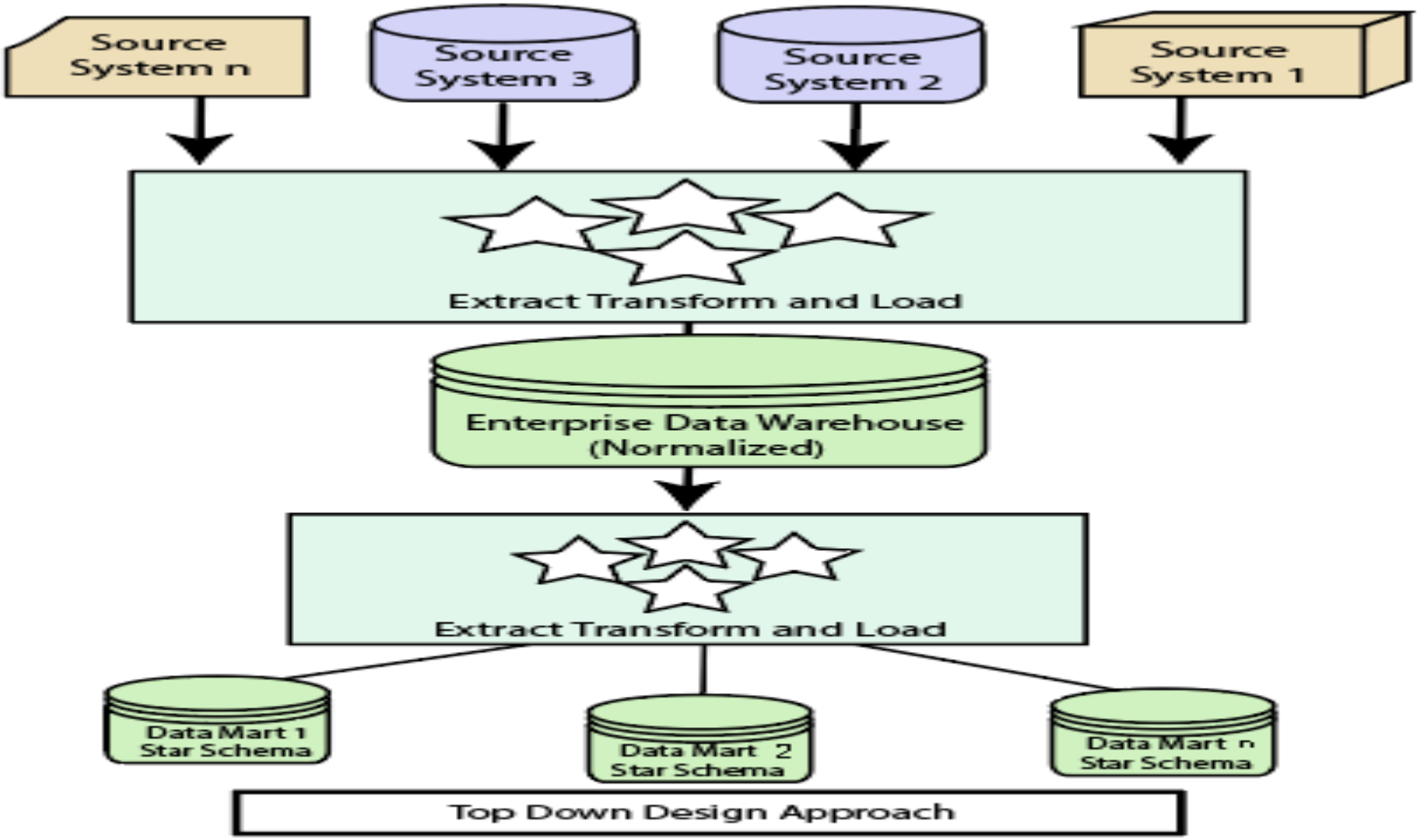
Design of Data Warehouse: A Business Analysis Framework

- Four views regarding the design of a data warehouse
  - Top-down view
    - allows selection of the relevant information necessary for the data warehouse
  - Data source view
    - exposes the information being captured, stored, and managed by operational systems
  - Data warehouse view
    - consists of fact tables and dimension tables
  - Business query view
    - sees the perspectives of data in the warehouse from the view of end-user

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
- 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

Top Down Approach

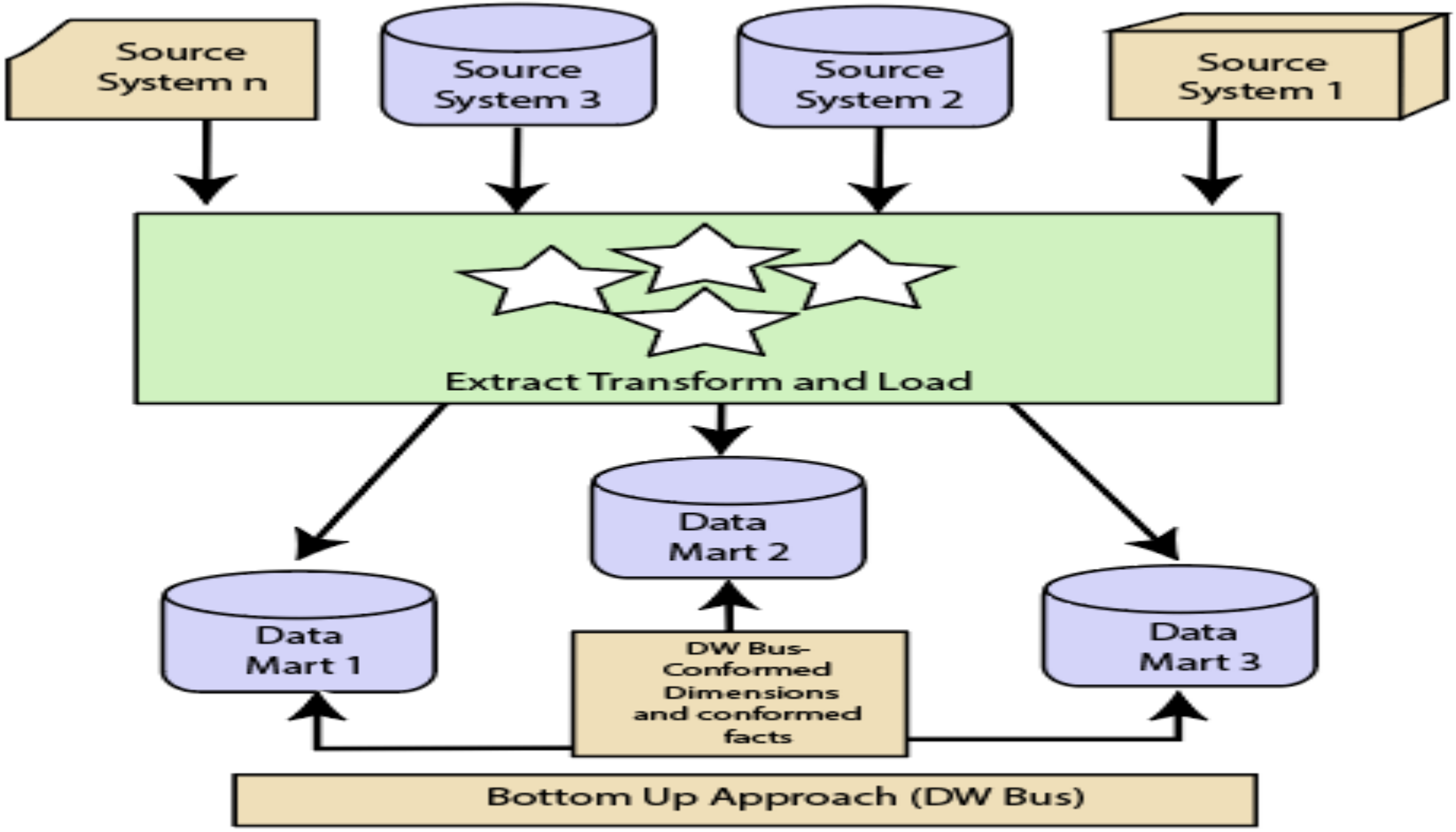


Top Down Design Approach

Top Down Approach

- **Advantages Of Top Down Design :**
  - It is easier to maintain Top Down Design
  - Provides consistent dimensional views of data across data marts, as all data marts are loaded from the data warehouse.
  - This approach is robust against business changes. Creating a new data mart from the data warehouse is very easy.
  - Initial cost is high but subsequent project development cost is lower
- **Disadvantages Of Top Down Design :**
  - It represents a very large project and the cost of implementing the project is significant.
  - It is time consuming and more time required for initial set up
  - Highly skilled people required for set up

Bottom Up Approach



Bottom Up Design Approach

Bottom Up Approach:

- **Advantages Of Bottom Up Design :**
  - This model contains consistent data marts and these data marts can be delivered quickly.
  - The data marts are created first to provide reporting capability
  - It is easier to extend the data warehouse as it can easily accommodate new business units. It is just creating new data marts and then integrating with other data marts.
  - This Approach take less time. Initial set up is very quickly
- **Disadvantage of Bottom Up Design :**
  - Initial cost is low but each subsequent phase will cost same
  - **The positions of the data warehouse and the data marts are reversed in the bottom-up approach design.**
  - It is difficult to maintain and often redundant and subject to revisions

Summary

Feature	Inmon Architecture	Kimball Architecture
Approach	Top-Down	Bottom-Up
Data Model	Normalized (3NF)	Denormalized (Star Schema)
Implementation Time	Longer	Faster
Query Performance	Slower due to normalization	Faster due to denormalization
Scalability	High	Moderate
Flexibility	Less flexible due to centralized design	More flexible with incremental data marts
User-Friendly	Less intuitive for end-users	More intuitive for end-users
Data Redundancy	Minimal	Higher
Complexity	More complex to manage	Easier to manage
Use Case	Large enterprises with comprehensive data integration needs	Smaller to mid-sized organizations with specific business needs

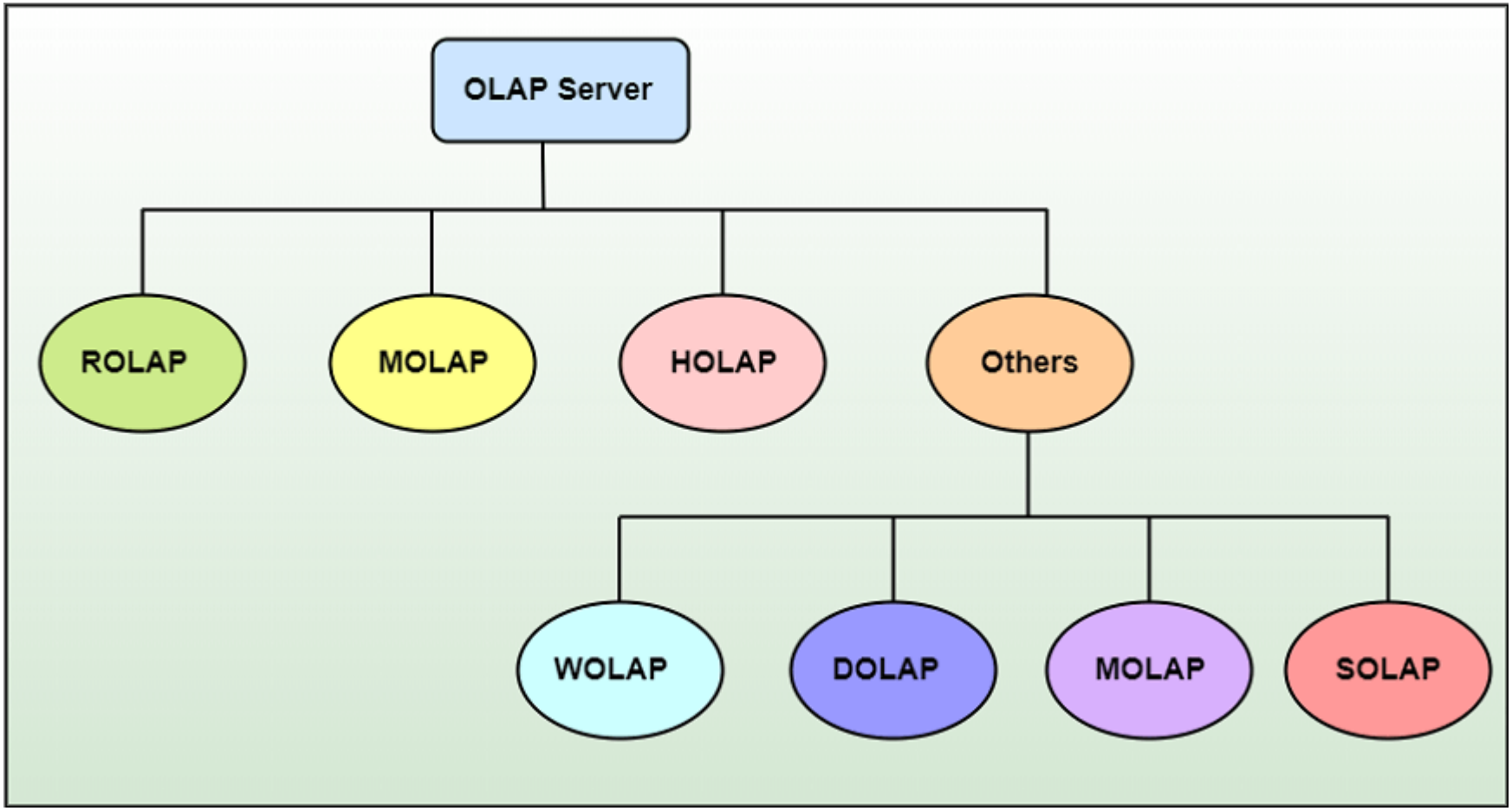
Data Warehouse Usage

- 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

OLAP Server Architectures

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP) (e.g., Microsoft SQLServer)
- Specialized SQL servers (e.g., Redbricks)
  - Specialized support for SQL queries over star/snowflake schemas

OLAP Server Architectures



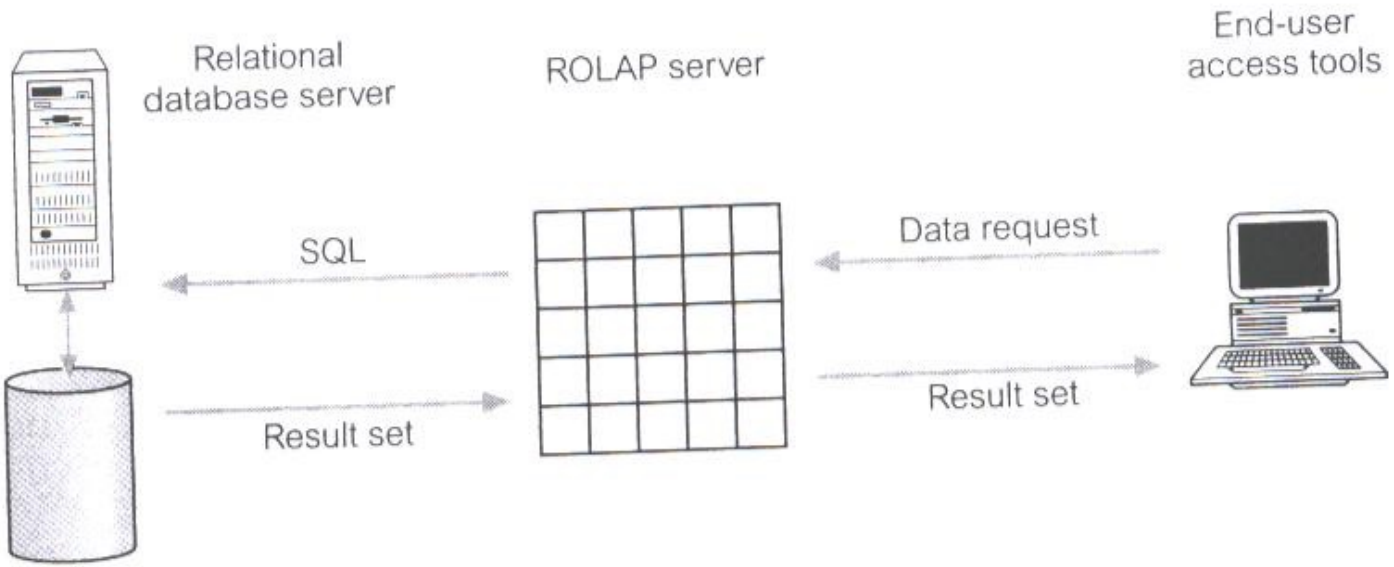
Relational OLAP (ROLAP)

- 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
- ROLAP supports RDBMS products through the use of a metadata layer, thus avoiding the requirement to create a static multi-dimensional data structure.
- This facilitates the creation of multiple multi-dimensional views of the two-dimensional relation.
- To improve performance, some ROLAP products have enhanced SQL engines to support the complexity of multi-dimensional analysis, while others recommend, or require, the use of highly denormalized database designs such as the star schema.

Relational OLAP (ROLAP)

- The development issues associated with ROLAP technology:
  - Performance problems associated with the processing of complex queries that require multiple passes through the relational data.
  - Development of middleware to facilitate the development of multi-dimensional applications.
  - Development of an option to create persistent multi-dimensional structures, together with facilities o assist in the administration of these structures.

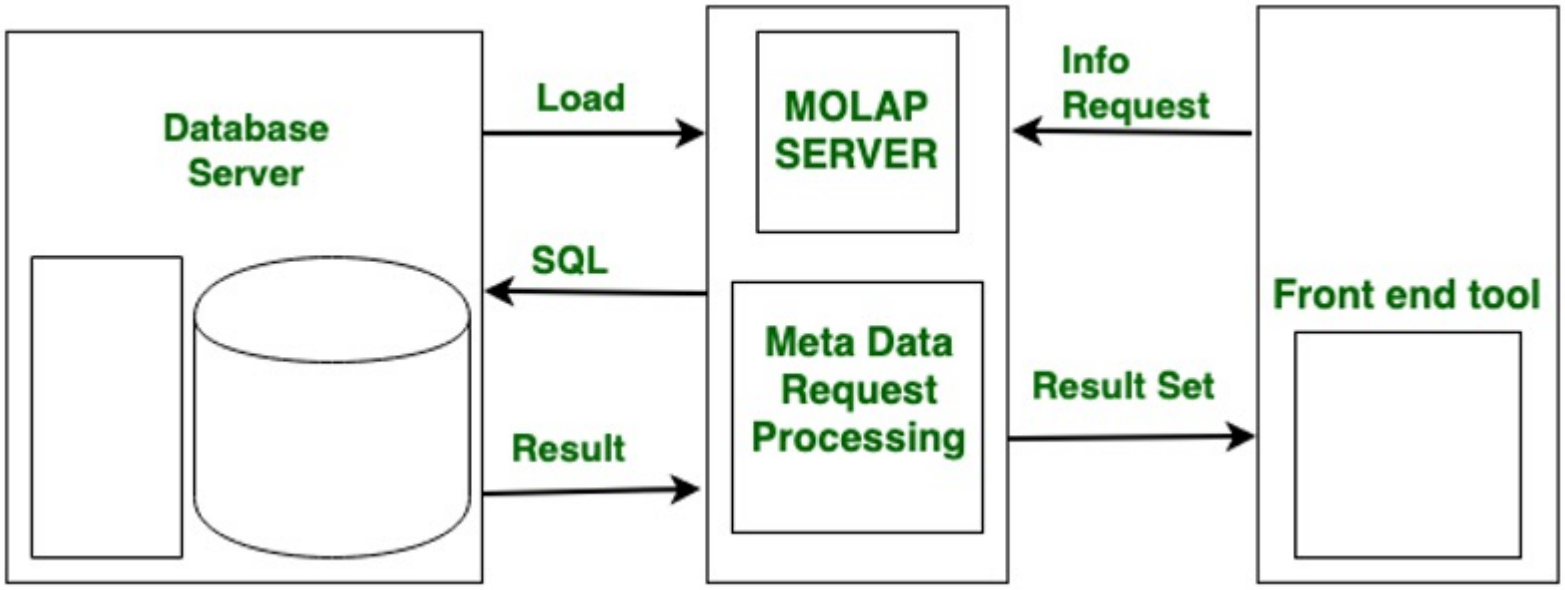
Relational OLAP (ROLAP)



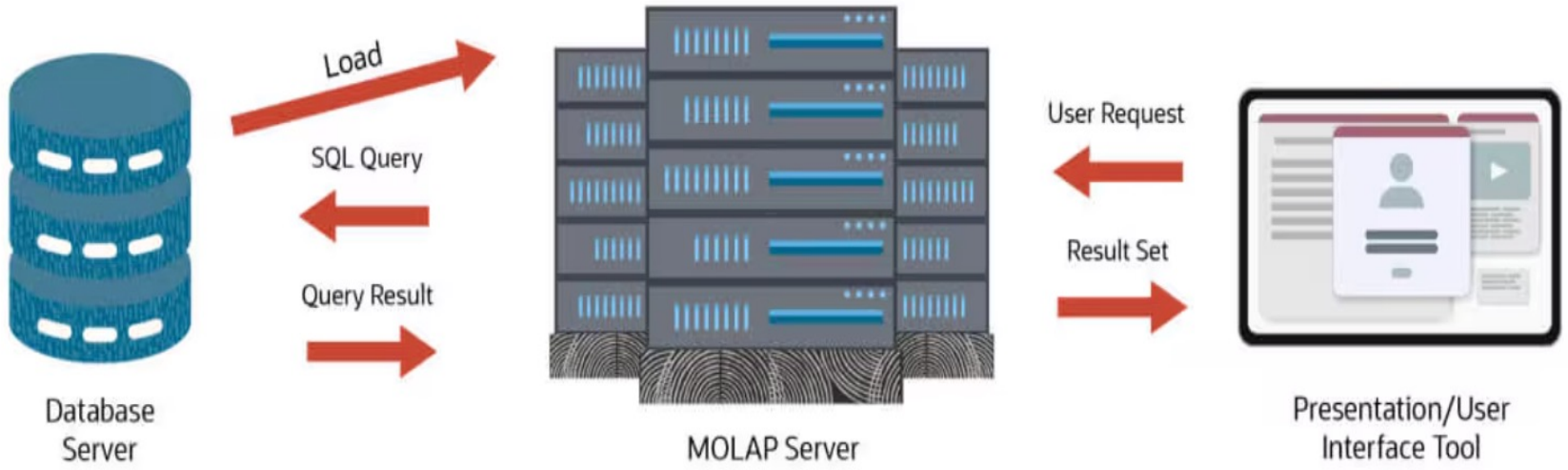
Multidimensional OLAP (MOLAP)

- MOLAP tools use specialized data structures and multi-dimensional database management systems (MDDBMS) to organize, navigate, and analyze data.
- Sparse array-based multidimensional storage engine that minimize the disk space requirements through sparse data management.
- Fast indexing to pre-computed summarized data
- The development issues associated with MOLAP:
  - Only a limited amount of data can be efficiently stored and analyzed.
  - Navigation and analysis of data are limited because the data is designed according to previously determined requirements.
  - MOLAP products require a different set of skills and tools to build and maintain the database.

Multidimensional OLAP (MOLAP)



MOLAP Architecture Components

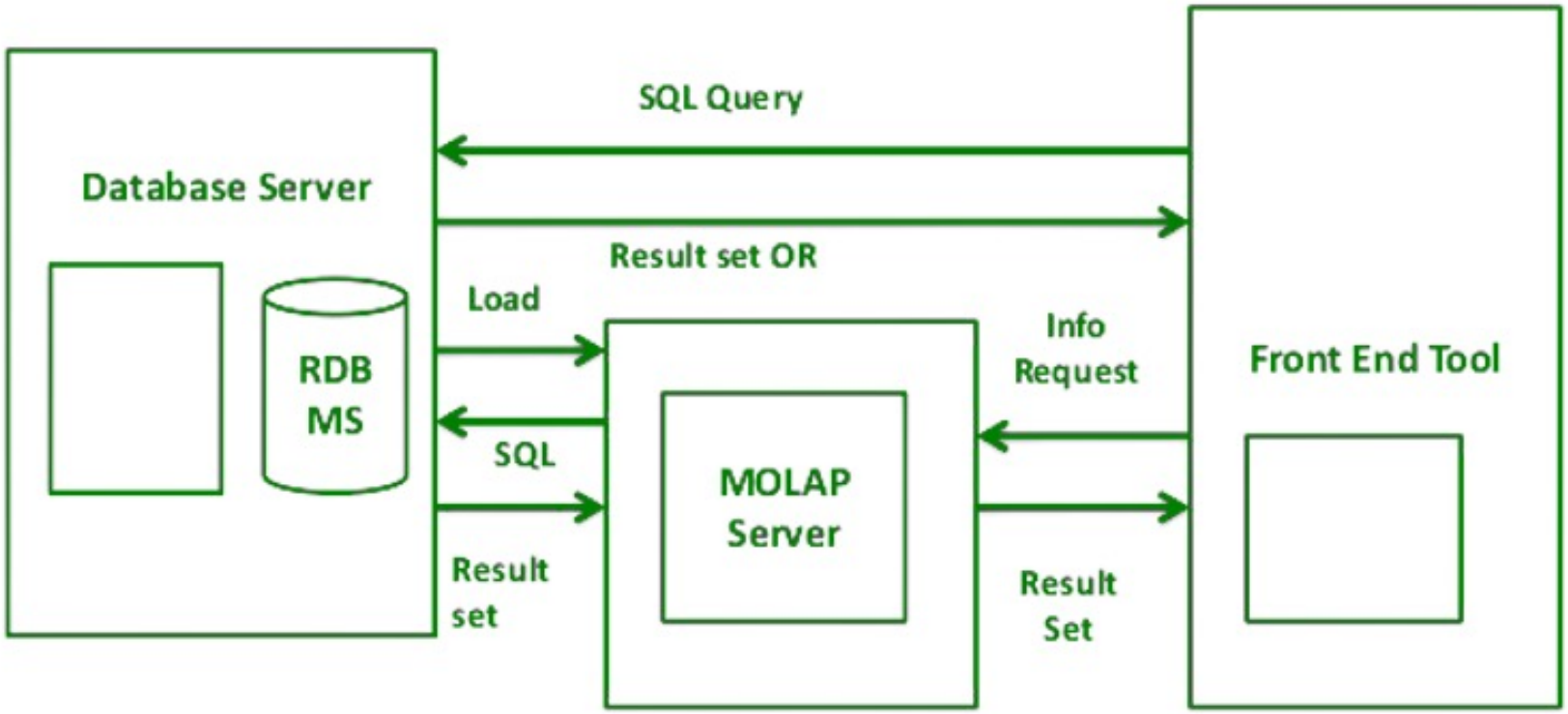


MOLAP systems typically have a three-tier architecture: a database server (1st tier), or repository, from which a MOLAP server (2nd tier) extracts data and preps it into data cubes that are optimized to respond to expected user queries that come from the presentation/user interface (3rd tier).

Hybrid OLAP (HOLAP)

- HOLAP tools provide limited analysis capability, either directly against RDBMS products, or by using an intermediate MOLAP server.
- HOLAP tools deliver selected data directly from DBMS or via MOLAP server to the desktop (or local server) in the form of data cube, where it is stored, analyzed, and maintained locally is the fastest-growing type of OLAP tools.
- Flexibility, e.g., low level: relational, high-level: array
- The issues associated with HOLAP tools:
  - The architecture results in significant data redundancy and may cause problems for networks that support many users.
  - Ability of each user to build a custom data cube may cause a lack of data consistency among users.
  - Only a limited amount of data can be efficiently maintained.

Hybrid OLAP (HOLAP)



Other Types

- **Web-Enabled OLAP (WOLAP) Server**
- WOLAP pertains to OLAP application which is accessible via the web browser. Unlike traditional client/server OLAP applications, WOLAP is considered to have a three-tiered architecture which consists of three components: a client, a middleware, and a database server.
- **Desktop OLAP (DOLAP) Server**
- DOLAP permits a user to download a section of the data from the database or source, and work with that dataset locally, or on their desktop.
- **Mobile OLAP (MOLAP) Server**
- Mobile OLAP enables users to access and work on OLAP data and applications remotely through the use of their mobile devices.
- **Spatial OLAP (SOLAP) Server**
- SOLAP includes the capabilities of both Geographic Information Systems (GIS) and OLAP into a single user interface. It facilitates the management of both spatial and non-spatial data.

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.

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?

Case Study

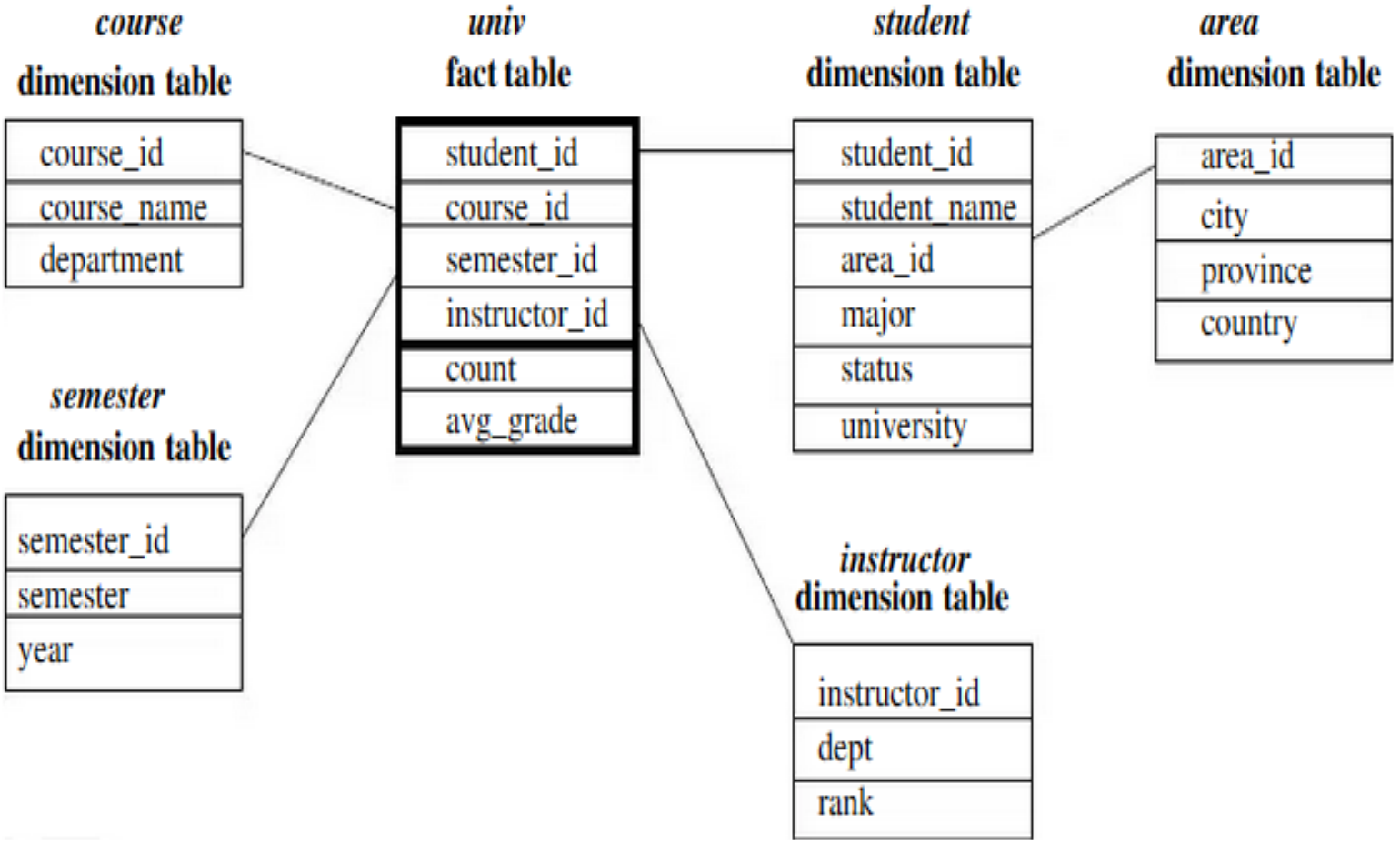
- **Suppose that a data warehouse for Big University consists of the following four dimensions: student, course, semester, and instructor , and two measures count and avg grade . When at the lowest conceptual level (e.g.,for a given student, course, semester, and instructor combination), the avg grade measure stores the actualcourse grade of the student. At higher conceptual levels, avg grade stores the average grade for the givencombination.**

  (a) Draw a snowflake schema diagram for the data warehouse.

  (b) Starting with the base cuboid [student,course,semester,instructor], what specific OLAP operations (e.g., roll-up from semester to year) should one perform in order to list the average grade of CS coursesfor each Big University student.

  (c) If each dimension has five levels (including all), such as " student < major < status < university < all ",how many cuboids will this cube contain (including the base and apex cuboids)?

Snowflake Schema diagram



- **Starting with the base cuboid [student,course,semester,instructor], what specific OLAP operations (e.g., roll-up from semester to year) should one perform in order to list the average grade of CS courses for each Big University student. The specific OLAP operations to be performed are:**
  - Roll-up on course from course id to department .
  - Roll-up on student from student id to university.
  - Dice on course, student with department="CS" and university = "Big University".
  - Drill-down on student from university to student name

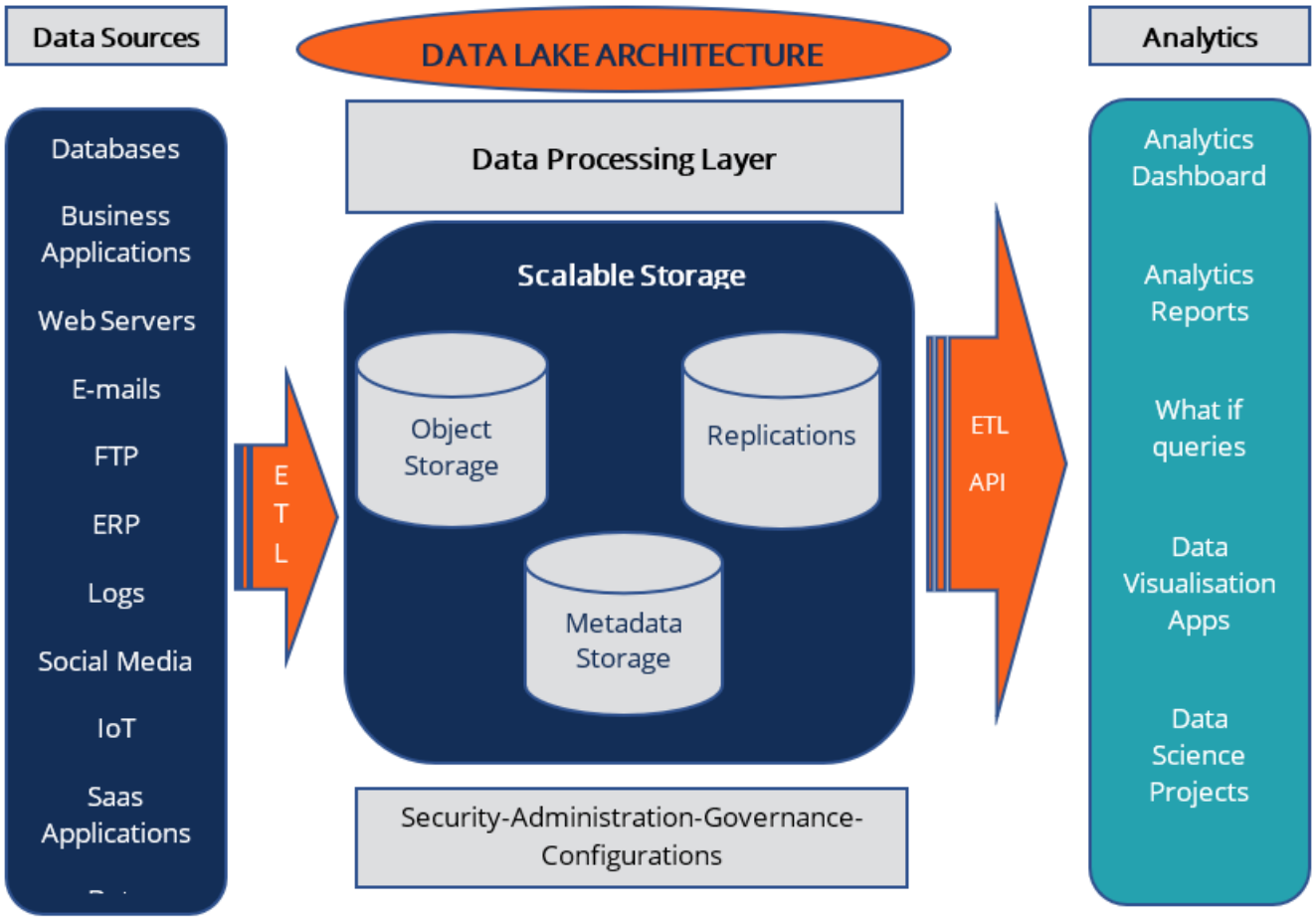
- **(c) If each dimension has five levels (including all), such as student < major < status < university < all,how many cuboids will this cube contain (including the base and apex cuboids)**
- This cube will contain 54= 625 cuboids

Data Lakes

- A **Data Lake** is a centralized repository that stores **raw data** in its **original format** (structured, semi-structured, and unstructured) at **any scale**.
- Think of it like a **big lake** where all kinds of data flow in—you decide **later** how to process and analyze it.

- **Types of Data Stored**
- **Structured** → tables, CSV, RDBMS data
- **Semi-structured** → JSON, XML, logs
- **Unstructured** → images, videos, audio, PDFs, text

a concept diagram for a data lake structure:



Data lake vs data warehouse: Key differences

The key differences between a data lake and a data warehouse are as follows [1, 2]:

Parameters	Data Lake	Data Warehouse
Data type	Raw (all types, no matter source of structure)	Processed (data stored according to metrics and attributes)
Data purpose	To be determined	Currently being used
Process	Extract Load Transform (ELT)	Extract Transform Load (ETL)
Schema position	After data storage, to offer agility and easy data capture	Before data storage, to offer security and high performance
Users	Data scientists, those who need in-depth analysis and tools (such as predictive modelling) to understand it	Business professionals, those who need it for operations
Accessibility	Accessible and easy to update	Complicated to make changes
History	Relatively new for big data	The concept has been around for decades

Data Lakehouse

- A **Data Lakehouse** is a **hybrid data architecture** that **combines the flexibility of a Data Lake** with the **performance, reliability, and structure of a Data Warehouse**.
- Data Lake + Data Warehouse = Data Lakehouse
- **Why Lakehouse Was Needed?**
- Problems with:
- **Data Lakes** → no ACID transactions, data quality issues, "data swamp"
- **Data Warehouses** → expensive, rigid schema, limited for AI/ML
- 💡 Lakehouse solves **both**.

- **Data Lakehouse Architecture (Layered View)**
- **Storage Layer**
  - Cloud object storage (S3, ADLS, GCS)
  - Open formats: **Parquet, ORC**
- **Metadata & Table Format Layer**
  - **Delta Lake**
  - **Apache Iceberg**
  - **Apache Hudi**
- **Processing Layer**
  - Apache Spark, Flink, Trino, Presto
- **Analytics & BI**
  - Power BI, Tableau, Looker
- **ML & AI**
  - MLflow, TensorFlow, PyTorch
- **Governance & Security**
  - Access control, versioning, auditing

Key Features of Data Lakehouse

- ACID transactions
- Schema enforcement & evolution
- Time travel (data versioning)
- High query performance
- Single source of truth
- Supports BI + ML together

Comparison: Data Lake vs Data Warehouse vs Lakehouse

Feature	Data Lake	Data Warehouse	Lakehouse
Data Type	All	Structured	All
Schema	On-read	On-write	Flexible
ACID	❌	✅	✅
Cost	Low	High	Medium
BI Support	Limited	Excellent	Excellent
ML Support	Excellent	Limited	Excellent