Data Warehousing and Data Mining (CS610)

Dr. Mahesha P.

Introduction

- DBMSs widely used to maintain transactional data
- Users involved in the operational aspects (adding, retrieving, deleting and updating) of larger system as efficiently as possible.
- Ex. Railway Reservation system.
  - How efficiently a system can be designed to perform reservation, modification and cancellation and so on.
- That is operational aspects of database system is a day to day operation.

Introduction

- Attempts to use of these data for analysis, exploration, identification of trends etc. has led to Decision Support Systems.
- We are going to discuss slightly different topic from the conventional idea of databases because the kind of users that the database is going to change.

Introduction

- There are other kind of users (High level) who use the database system as well.
- They are the users who take **strategic decision**
- These kind of strategic decisions are of qualitatively different in nature than operational system
- The strategic decision Ex.:
  - In Railways, best location to place next reservation counter
  - Which part of the city most people are travelling by train
  - Which part of the city has most people travelling 1 AC
  - Best time to offer concession, etc.,

Introduction

- The same database that are fed and retrieved in operational situation are required for making strategic decision as well.

A producer wants to know....

Which are our lowest/highest margin customers ?

Who are my customers and what products are they buying?

What is the most effective distribution channel?

What product prom--otions have the biggest impact on revenue?

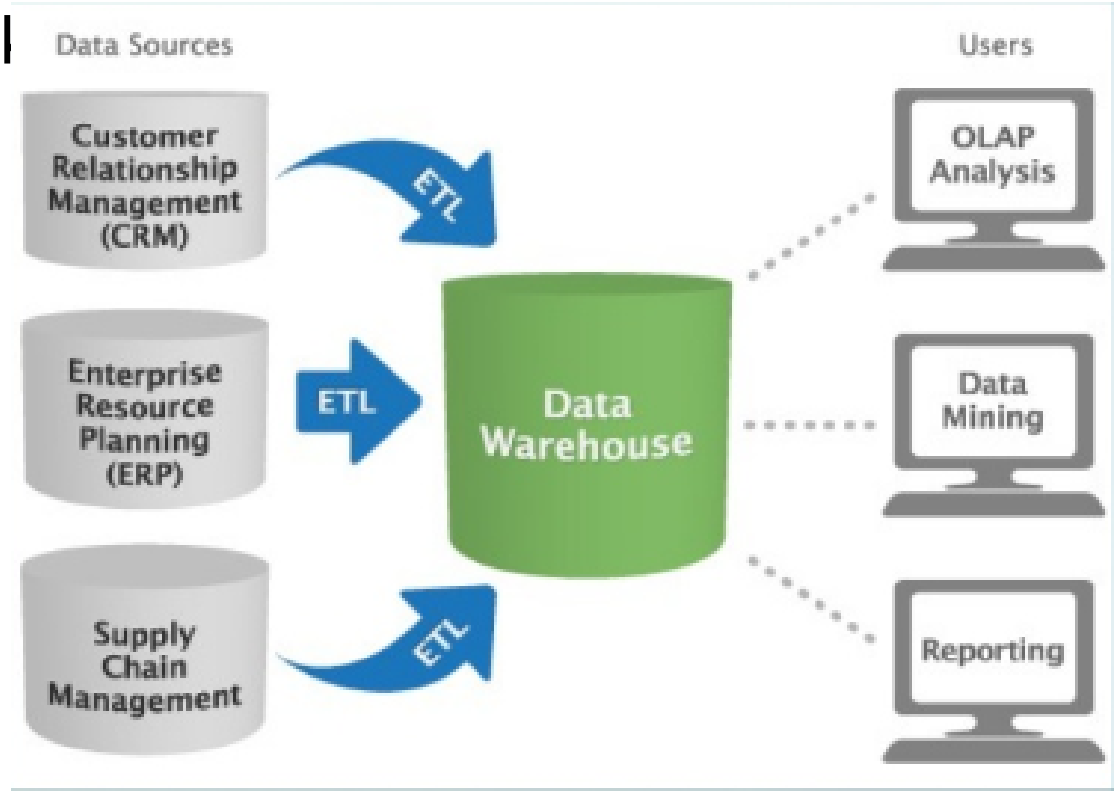
Which customers are most likely to go to the competition ?

What impact will new products/services have on revenue and margins?

What is Data Warehouse ?

- A Data Warehouse is a single, complete and consistent store of data obtained from a variety of different sources made available to end users in a what they can understand and use in a business context

- Data Warehousing is a process of transforming data into information and making it available to users in a timely enough manner to mal



DATA WAREHOUSE

- Distinctive characteristics:
  - Separate from operational databases
  - Subject oriented: provides a simple, concise view on one or more selected areas, in support of the decision process
  - Constructed by integrating multiple, heterogeneous data sources
  - Contains historical data
  - (Mostly) Read-Only access: periodic, infrequent updates

Data Mining

- The concept of data mining is gaining popularity in the e-commerce business activity in general.
- The amount of data being generated and stored is growing exponentially, due to the continuing advances in computer technology.
- Organizations are storing, processing and analysing data more than any time in the history and the trends continue to grow

So what is data mining ?

- Data mining is
  - data mining (sometimes called data or knowledge discovery) is the process of analysing data from different perspectives and summarising it into useful information
  - i.e. the objective analysis of the information that you already collect.
- Simply put, any organization which has data and processes it can be analysed with data mining
- Results are information, actionable information that can be used to organizations to
  - increase revenue and productivity, cuts costs,
  - fine tune their processes and increase efficiency

Why Data mining ?

- **The Explosive Growth of Data: from terabytes to petabytes**
  - Data collection and data availability
    - Automated data collection tools, database systems, Web, computerized society
  - Major sources of abundant data
    - Business: Web, e-commerce, transactions, stocks, …
    - Science: Remote sensing, bioinformatics, scientific simulation, …
    - Society and everyone: news, images, video, documents
    - Internet …

2019 This Is What Happens In An Internet Minute

What Can Data Mining Do ?

- Classification
  - » Classify credit applicants as low, medium, high risk
  - » Classify insurance claims as normal, suspicious
- Estimation
  - » Estimate the probability of a direct mailing response
  - » Estimate the lifetime value of a customer
- Prediction
  - » Predict which customers will leave within six months
  - » Predict the size of the balance that will be transferred by a credit card prospect
- Association
  - » Find out items customers are likely to buy together
  - » Find out what books to recommend to Amazon.com users
- Clustering
  - » Difference from classification: classes are unknown!

Some applications of Data Mining

- **Business data analysis and decision support**
  - Marketing focalization
    - Recognizing specific market segments that respond to particular characteristics
    - Return on mailing campaign (target marketing)
  - Customer Profiling
    - Segmentation of customer for marketing strategies and/or product offerings
    - Customer behavior understanding
    - Customer retention and loyalty
    - Mass customization / personalization

Some applications of Data Mining

- **Fraud detection**
  - Detecting telephone fraud:
    - Telephone call model: destination of the call, duration, time of day or week
    - Analyze patterns that deviate from an expected norm
  - Detection of credit-card fraud
  - Detecting suspicious money transactions (money laundering)
- **Text mining:**
  - Message filtering (e-mail, newsgroups, etc.)
  - Newspaper articles analysis
  - Text and document categorization
- **Web Mining**
  - Mining patterns from the content, usage, and structure of Web resources

Data warehousing vs Data mining

- Data warehousing is the process of compiling and organizing data into one common database, and
- Data mining is the process of extracting meaningful data from that database. The data mining process relies on the data compiled in the data warehousing phase in order to detect meaningful patterns.

Remember that data warehousing is a process that must occur before any data mining can take place.

Course Outline

- **UNIT 1: Data Warehousing and Online Analytical Processing (OLAP):**
  - In this chapter, we study definition of the data warehouse
  - We also look at data warehouse architecture, including steps on data warehouse design and construction. An overview of data warehouse implementation.
  - In particular, we study the *data cube, a multidimensional data model for data warehouses* and OLAP, as well as OLAP operations such as roll-up, drill-down, slicing, and dicing.
  - OLAP data indexing, and OLAP query processing

Course Outline

- UNIT 2 : **Data Mining Introduction**
  - Why Data Mining,
  - Data Mining Tasks :
    - Predictive tasks, to predict the value of a particular attribute based on the values of other attributes
    - Descriptive tasks, are often exploratory in nature and require post-processing techniques to validate and explain the results.
  - What Kinds of data can be Mined, What Kinds of patterns can be Mined, Which Technologies are used, Which types of Applications are Targeted, Major issues in Data Mining.
  - **Getting to Know Your Data** – Types of Data : data sets are grouped into three types: record data, graph based data, and ordered data.
  - Data Objects and Attribute Types, Basic Statistical Description of

Course Outline

- Unit – 3 : Preprocessing
  - An Overview, Major Tasks in Data Preprocessing, Data Cleaning: Missing Values, Noisy Data, Data Integration: Entity Identification Problem, Tuple Duplication, Data Value Conflict Detection and Resolution,
  - **Data Reduction:** Overview of Data Reduction Strategies, Principal Components Analysis, Attribute Subset Selection, Histograms, Clustering, Sampling,
  - **Data Transformation and Data Discretization:** Data Transformation Strategies Overview, Data Transformation by Normalization.

Course Outline

- Unit – 4: **Mining Frequent Patterns, Associations, and Correlations:**
  - Market Basket Analysis,
  - Association Analysis : Frequent Itemset Generation : Apriori Algorithm : Finding Frequent Itemsets by Confined Candidate Generation, A Pattern- Growth Approach For Mining Frequent Itemsets
  - Sequential Pattern Discovery (object associated with its own *timeline of events*)
  - **Classification**
    - General approach to solve classification problem,
    - Decision Trees, Rule Based Classifiers, Nearest Neighbor Classifiers.
    - Bayesian Classifiers, Estimating Predictive accuracy of classification methods,
    - Improving accuracy of clarification methods, Evaluation criteria for classification methods, Multiclass Problem.

Course Outline

- **Unit 5: Clustering Techniques**
  - Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships
  - Types of Cluster Analysis Methods,
  - Partitional Methods, :*k*-Means: A Centroid-Based Technique
  - Hierarchical Methods: Agglomerative versus Divisive Hierarchical Clustering.

Textbooks

- Textbook
  - Jiawei Han and Micheline Kamber: Data Mining-Concepts and Techniques, **3rd Edition**, Morgan Kaufmann Publisher, 2014.
- **Reference Books:**
  - Alex Berson and Stephen J. Smith, "Data Warehousing, Data Mining & OLAP", Tata McGraw–Hill Edition, Tenth Reprint2007
  - Pang Ning Tan, Michael Steinbach and Vipin kumar, Introduction to Data Mining, Pearson, 2006.
  - G.K. Gupta: Introduction to Data Mining with Case Studies, 3rd Edition, PHI, New Delhi, 2009.

Unit – 1
Data Warehousing and Online Analytical Processing (OLAP):

Contents

- Why do you need a warehouse ?
- Operational and informational systems
- Key features of Data Warehouse
- Data warehouse architecture
- Design Issues,
- Guidelines for Data Warehouse Implementation
- OLAP systems

Scenario 1

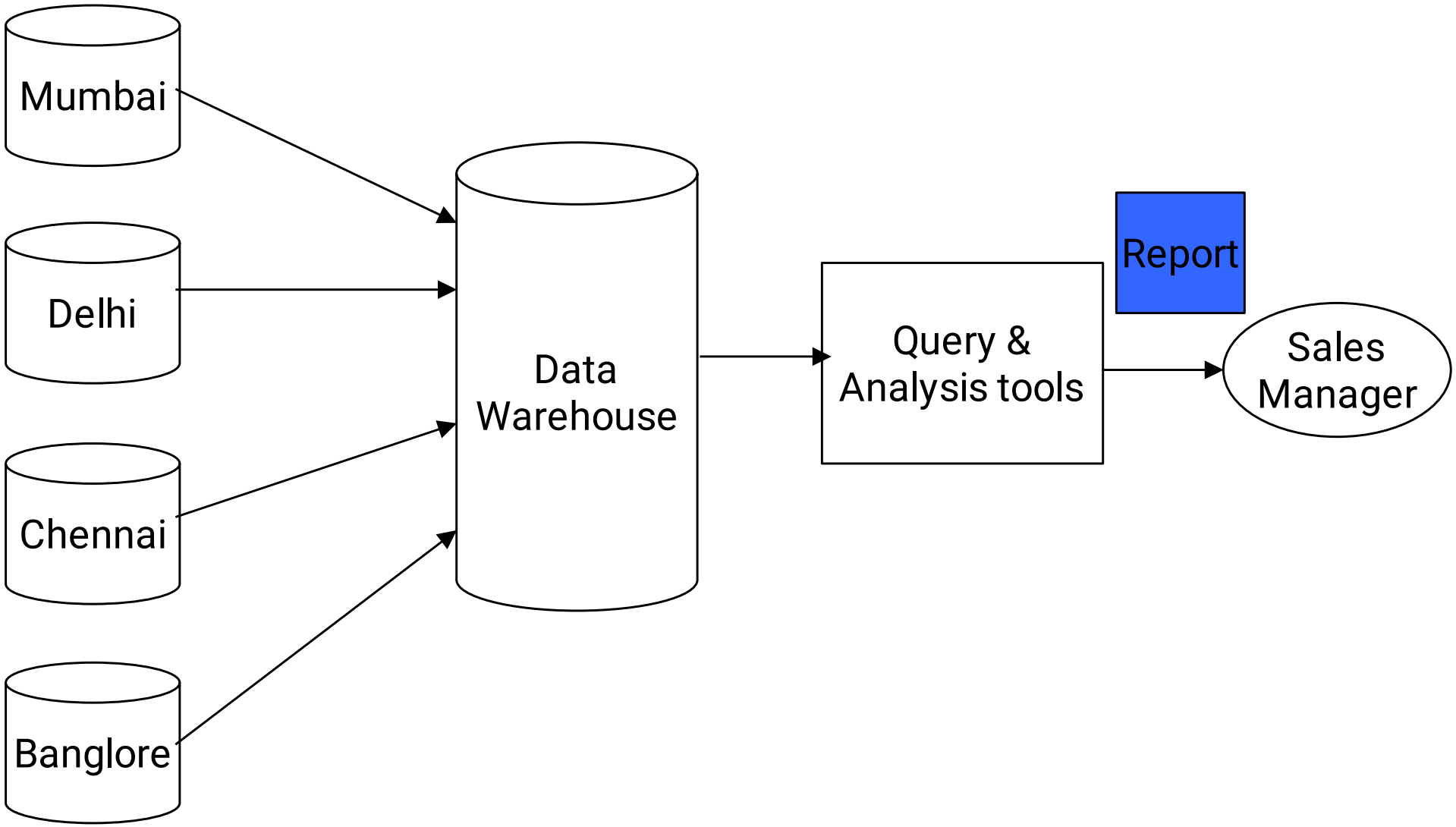
- ABC Pvt. Ltd is a company with branches at Mumbai, Delhi, Chennai and Bangalore.
- Each branch has a separate operational system.
- The Sales Manager wants quarterly sales report.

Scenario 1 : ABC Pvt Ltd.

Solution :ABC Pvt Ltd.

- Extract sales information from each database.
- Store the information in a common repository at a single site.

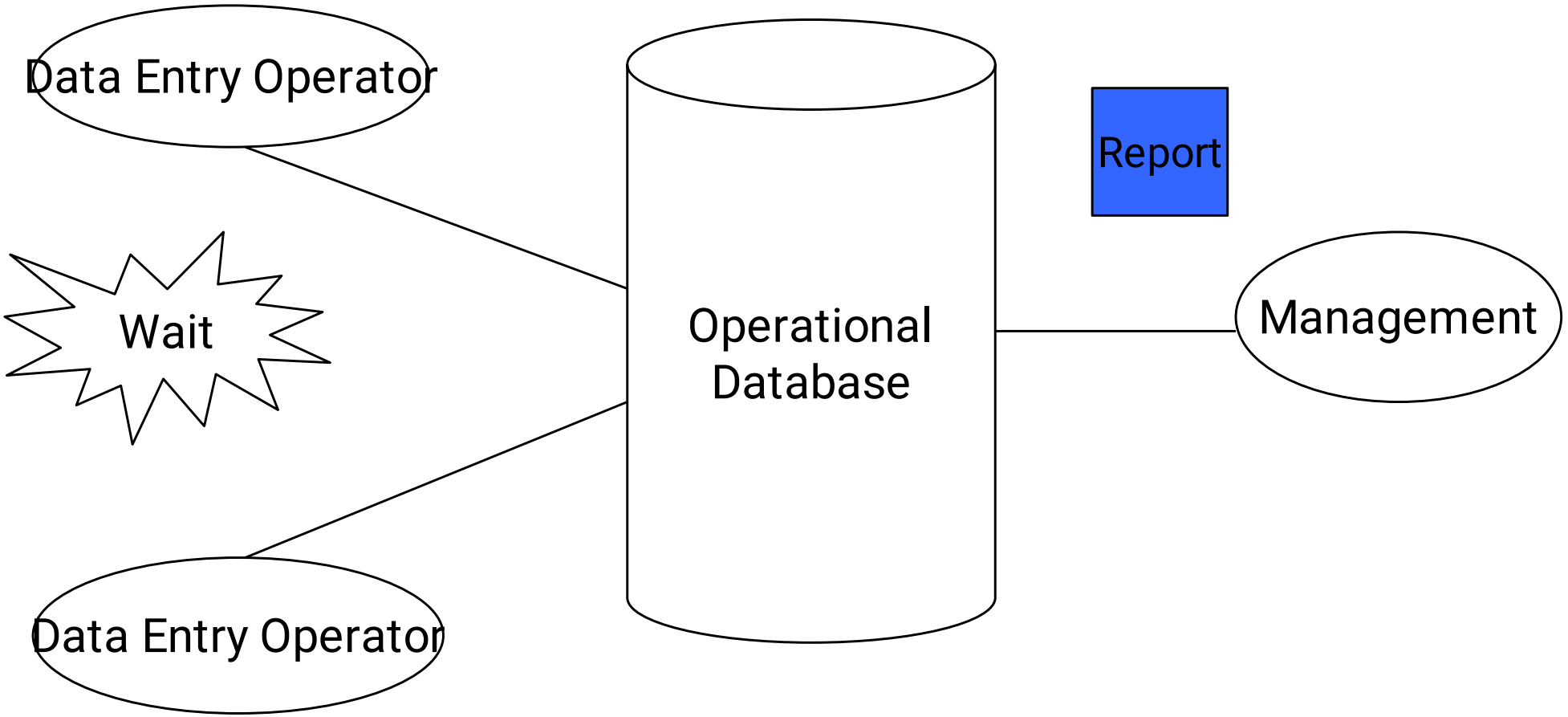
Solution :ABC Pvt Ltd.



Scenario 2

- One Stop Shopping Super Market has huge operational database.
- Whenever Executives wants some report the OLTP system becomes slow and data entry operators have to wait for some time.

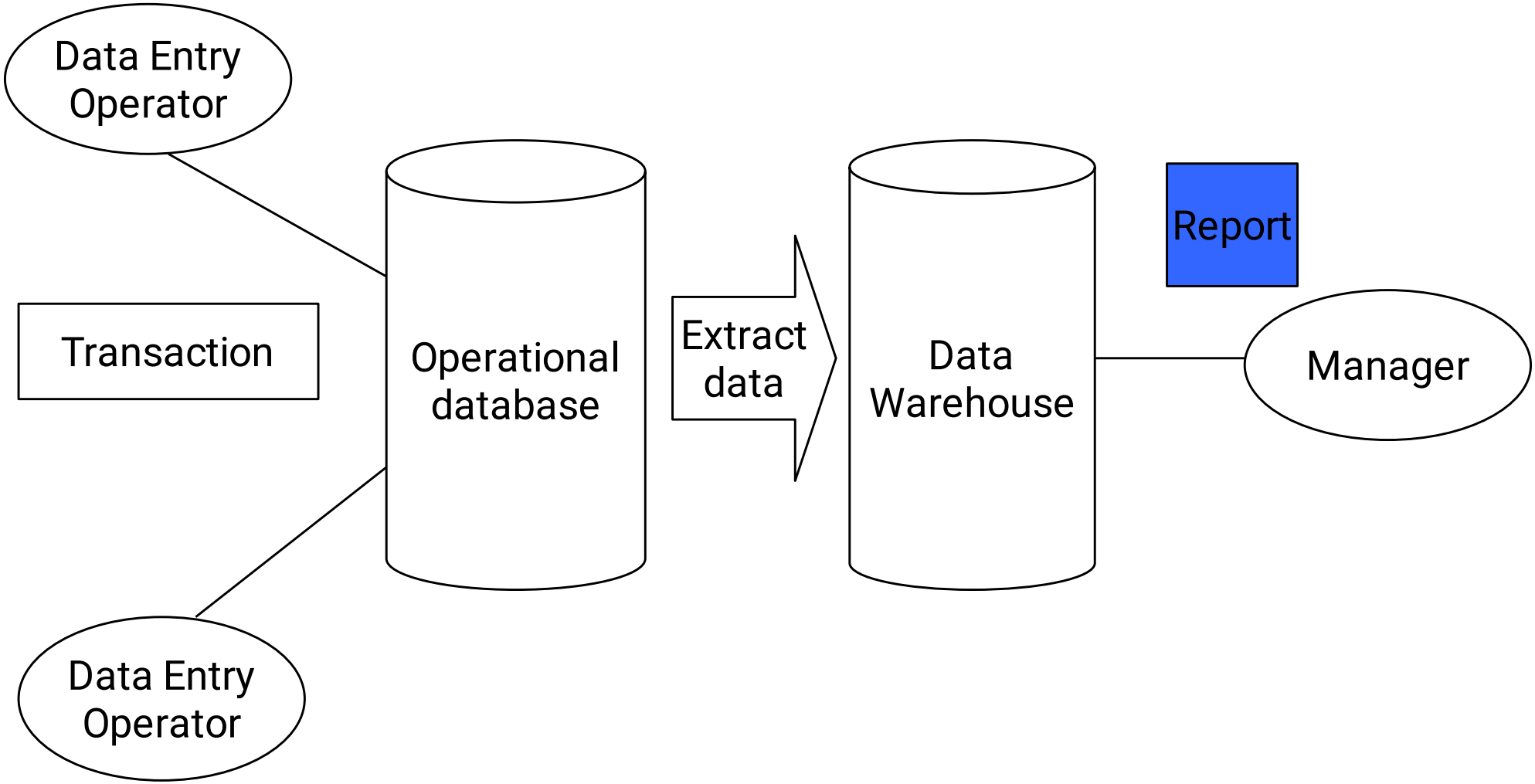
Scenario 2 : One Stop Shopping



Solution

- Extract data needed for analysis from operational database and store it in another system (data warehouse).
- Refresh warehouse at regular intervals so that it contains up to date information for analysis.
- Warehouse will contain data with historical perspective.

Solution


Data Entry Operator
Transaction
Data Entry Operator
Operational database
Extract data
Data Warehouse
Report
Manager

Why do you need a warehouse?

- The executives and managers who are responsible for keeping the enterprise competitive need information to make proper decisions.

Need for strategic information

- Executives and managers need information for the following purposes:
  - To get in-depth knowledge of their company's operations
  - Monitor how the business factors change over time
  - Compare their company's performance relative to the competition and to industry benchmarks.
  - Customers' needs and preferences.
  - Sales and marketing results.
  - Quality levels of products and services.

Key features of Data Warehouse

- The Key features of data warehouse are
  - subject-oriented,
  - integrated,
  - time-variant,
  - nonvolatile

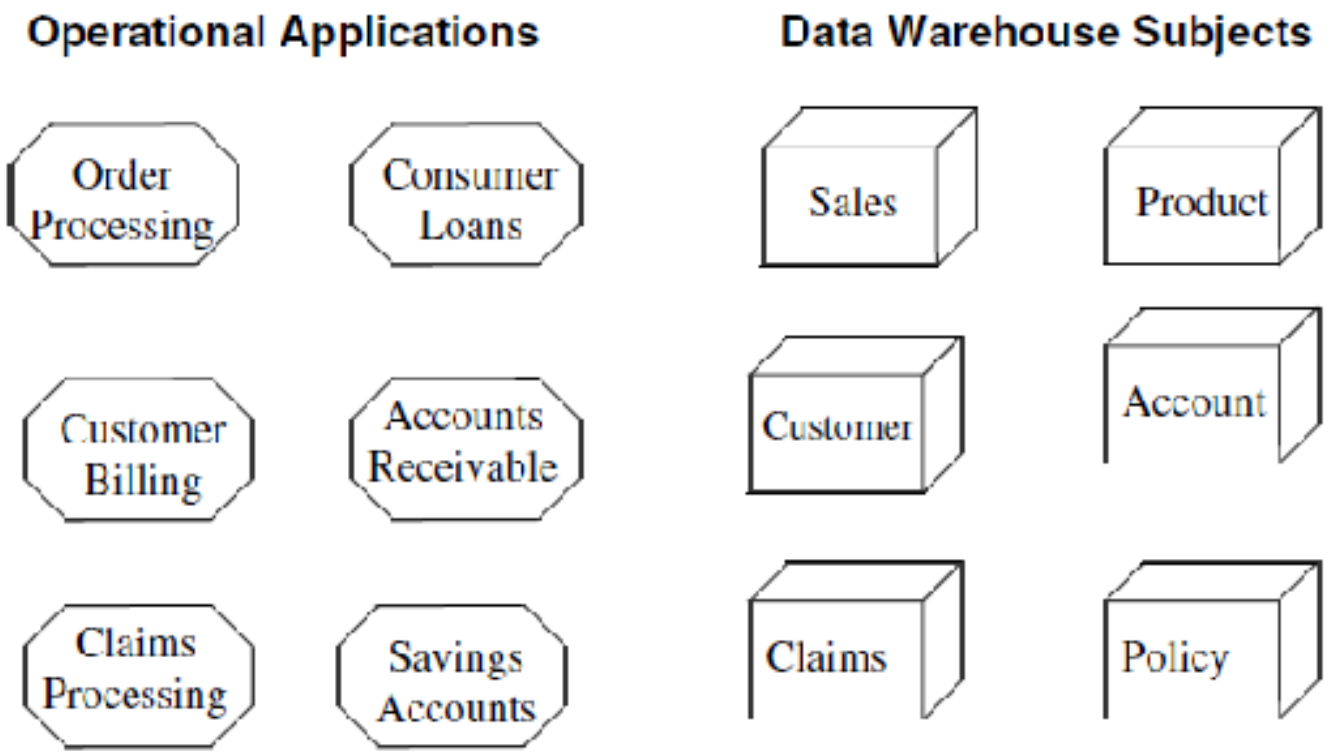
Key features of Data Warehouse

- **Subject-oriented**
  - A data warehouse is organized around major subjects, such as customer, supplier, product, and sales.
  - Rather than concentrating on the day-to-day operations and transaction processing of an organization,
  - Hence, data warehouses provide a simple and concise view around particular subject issues by excluding data that are not useful

Key features of Data Warehouse

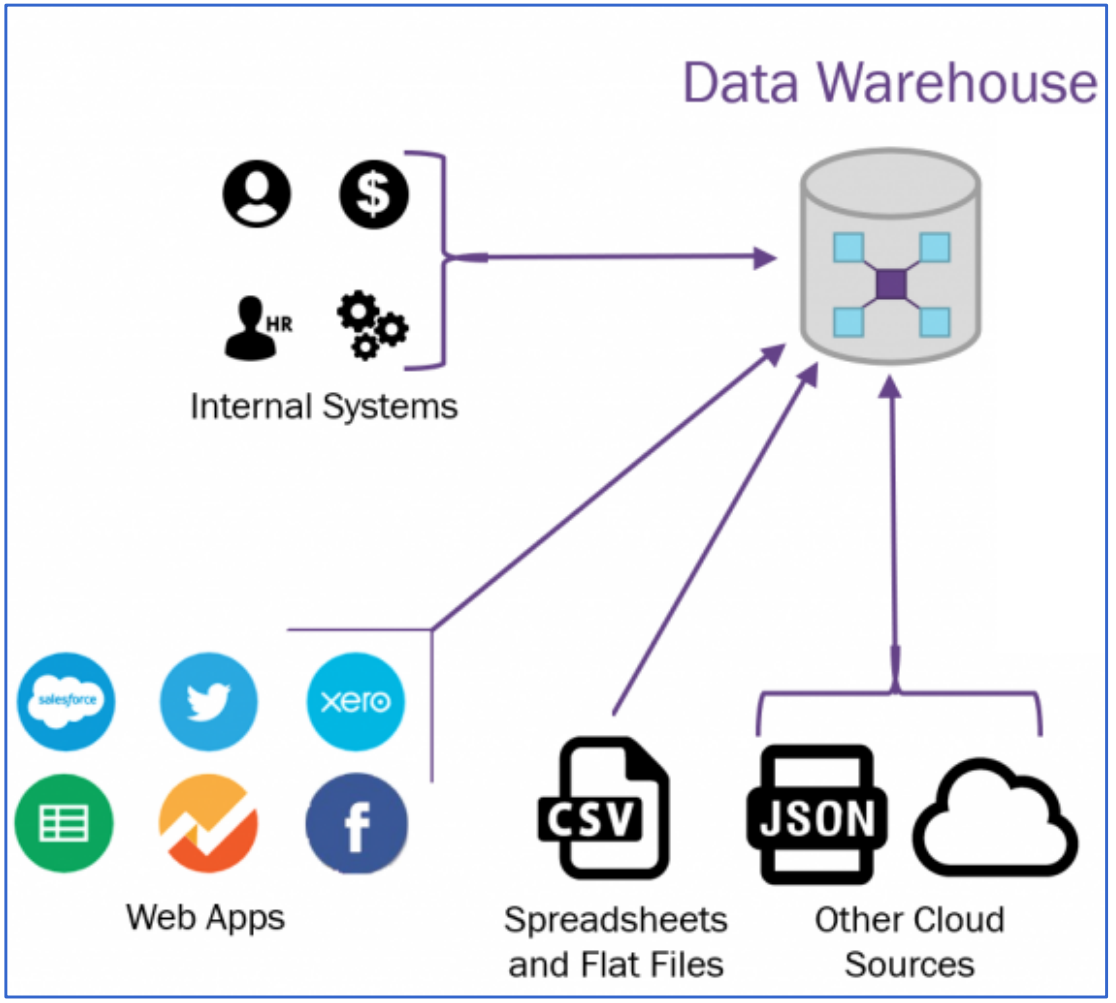
Subject-oriented

In the data warehouse, data is not stored by operational applications, but by business subjects.



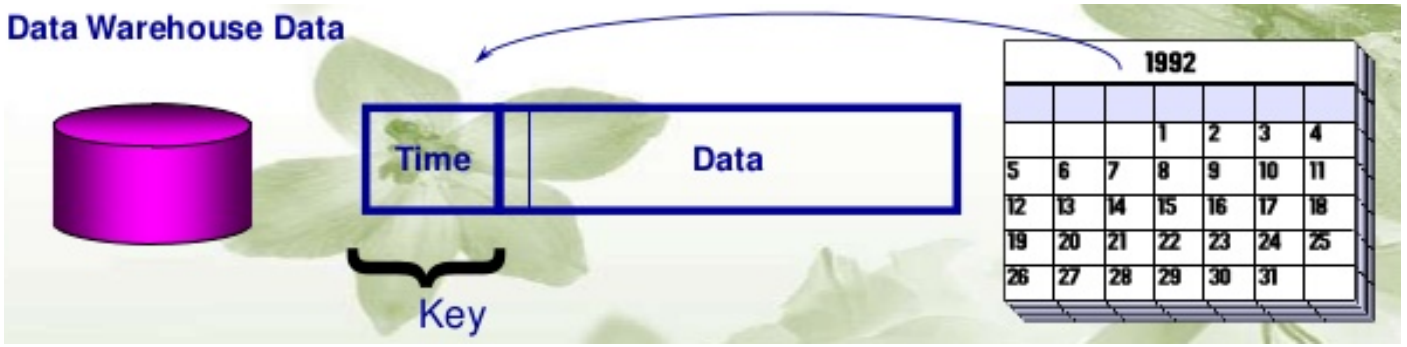
Key features of Data Warehouse

- **Integration**
  - A data warehouse is constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and online transaction records.
  - Data cleaning and data integration techniques are applied to ensure consistency in naming conventions



Key features of Data Warehouse

- Time-variant
  - Data is stored as a series of snapshots or views, which record how it is collected across time.



  - Data is tagged with some element of time - *creation date, as of date* , etc.
  - The time-variant nature in a DW allows, for *analysis of the past* , *Relates information to the present* , Enables *forecasts for the future*

Key features of Data Warehouse

- **Non-volatile**
  - A data warehouse is always a physically separate store of data transformed from the operational environment.
  - Due to this it does not require transaction processing, recovery, and concurrency control mechanisms.
  - It requires only two operations in data accessing:
    - Initial loading of data
    - Access of data.
    - Timely updates

Operational and informational systems

Operational Systems

- The major task of operational database systems is to perform on-line transaction and query processing.
- These systems are called on-line transaction processing (OLTP) systems.
- They cover most of the day-to-day operations of an organization, such as purchasing, banking, payroll, registration, and accounting.

Operational and informational systems

Informational Systems

- On the other hand, there are other functions that go on within the enterprise
- Functions like “marketing planning” and “financial analysis” also require information systems to support them.
- These systems are known as on-line analytical processing (OLAP) systems.

Operational and informational systems

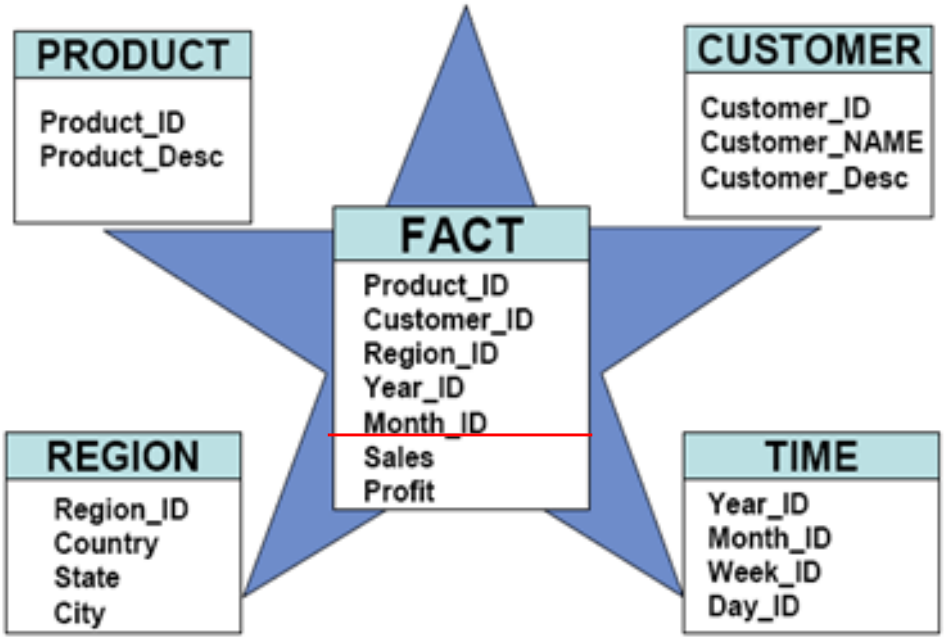
- The major distinguishing features between OLTP and OLAP are summarized as follows:
  - Users and system orientation
  - Data contents
  - Database design
  - View
  - Access patterns

Operational and informational systems

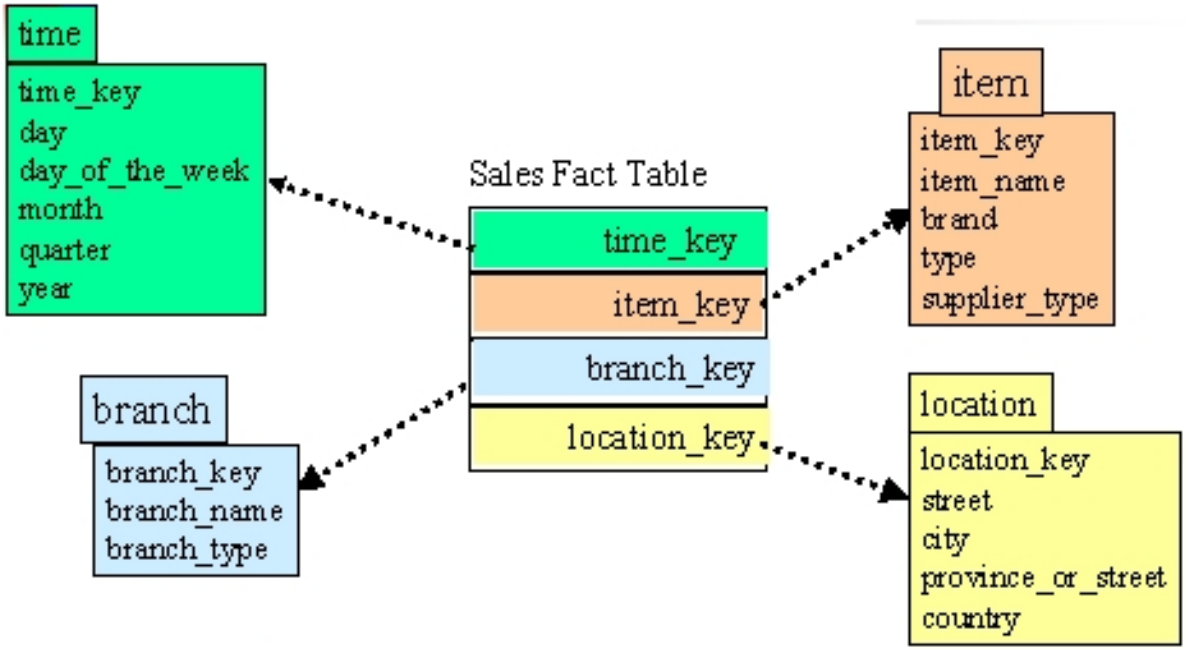
- **Users and system orientation**
  - OLTP : customer-oriented and is used for transaction and query processing by *clerks, clients, and professionals* .
  - OLAP : market-oriented and is used for data analysis by knowledge workers, including *managers, executives, and analysts*
- **Data contents**
  - OLTP, manages current data that, typically, are *too detailed* to be used for decision making.
  - OLAP, manages large amounts of *historical data* , provides facilities for summarization and aggregation

Operational and informational systems

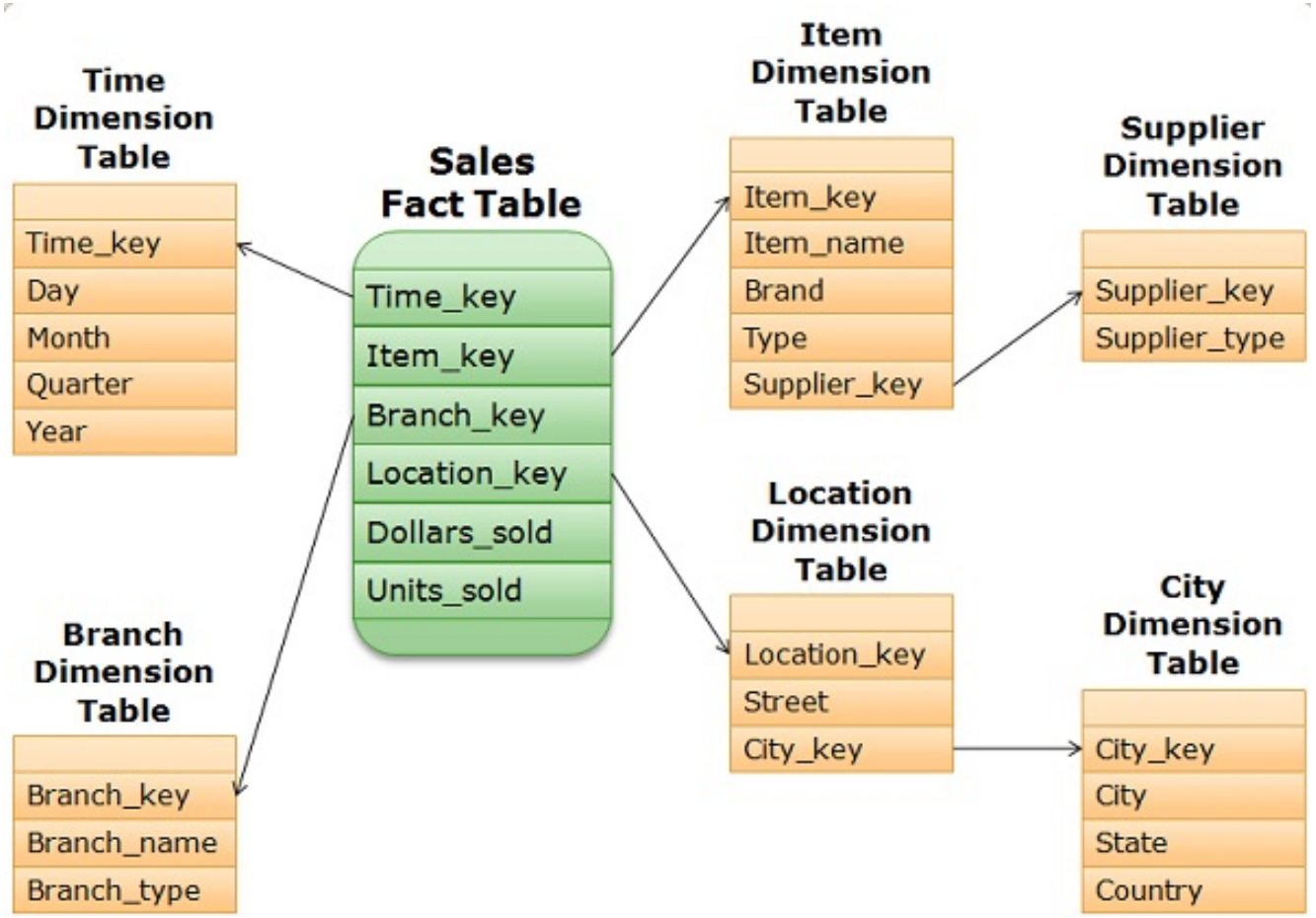
- Database design
  - OLTP, adopts an entity-relationship (ER) data model and an application-oriented database design.
  - OLAP, adopts either a *star or snowflake model and a subject oriented* database design.
    - **Star schema :** A fact table in the middle connected to a set of dimension tables
    - **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


PRODUCT
Product_ID
Product_Desc
CUSTOMER
Customer_ID
Customer_NAME
Customer_Desc
FACT
Product_ID
Customer_ID
Region_ID
Year_ID
Month_ID
Sales
Profit
REGION
Region_ID
Country
State
City
TIME
Year_ID
Month_ID
Week_ID
Day_ID

Star Schema


time
time_key
day
day_of_the_week
month
quarter
year
Sales Fact Table
time_key
item_key
branch_key
location_key
item
item_key
item_name
brand
type
supplier_type
branch
branch_key
branch_name
branch_type
location
location_key
street
city
province_or_street
country

Snowflake schema



Operational and informational systems

- View
  - OLTP *focuses mainly on the current data* within an enterprise or department without referring to historical data
  - OLAP systems deal with information that *originates and integrated from many data stores.*

Operational and informational systems

- **Access patterns**
  - The access patterns of an OLTP system consist mainly of short, atomic transactions.
  - Such a system requires concurrency control and recovery mechanisms.
  - However, accesses to OLAP systems are mostly read-only operations

- Other features that distinguish between OLTP and OLAP systems are summarized in the following Table

Table 4.1 Comparison of OLTP and OLAP Systems

Feature	*OLTP*	*OLAP*
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements decision support
DB design	ER-based, application-oriented	star/snowflake, subject-oriented
Data	current, guaranteed up-to-date	historic, accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	GB to high-order GB	$\geq$ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

Why Have a Separate DataWarehouse?

"Why not perform online analytical processing directly on such databases instead of spending additional time and resources to construct a separate data warehouse?"

Why Have a Separate Data Warehouse?

- A major reason for such a separation is to help promote the *high performance of both systems*
  - **Operational database** is tuned from known tasks like, searching for particular records, and optimizing "canned" queries
  - **Warehouse** is tuned for OLAP; complex OLAP queries, multidimensional view, consolidation
  - They involve the computation of large groups of data at summarized levels, and require the *use of special data organization, access, and implementation methods based on multidimensional*
  - Processing OLAP queries in operational databases would substantially degrade the performance of operational tasks.

Why Have a Separate Data Warehouse?

- Moreover, an operational database supports the concurrent processing of multiple transactions.
  - In Operational systems, Concurrency control and recovery mechanisms, such as locking and logging, are required to ensure the consistency and robustness of transactions
  - Concurrency control and recovery mechanisms, if applied for such OLAP operations, may jeopardize the execution of concurrent transactions and thus substantially reduce the throughput of an OLTP system

Why Have a Separate Data Warehouse?

- Different functions and different data
  - Decision support requires consolidation (aggregation, summarization) of data from heterogeneous sources. Resulting in high-quality, clean, and integrated data.
  - In contrast, operational databases contain only detailed raw data, such as transactions, which need to be consolidated before analysis.
- Because the two systems provide quite different functionalities and require different kinds of data, *it is necessary to maintain separate databases.*

Data Warehousing: A Multitiered Architecture

- Data warehouses often adopt a three-tier architecture, as presented in Figure

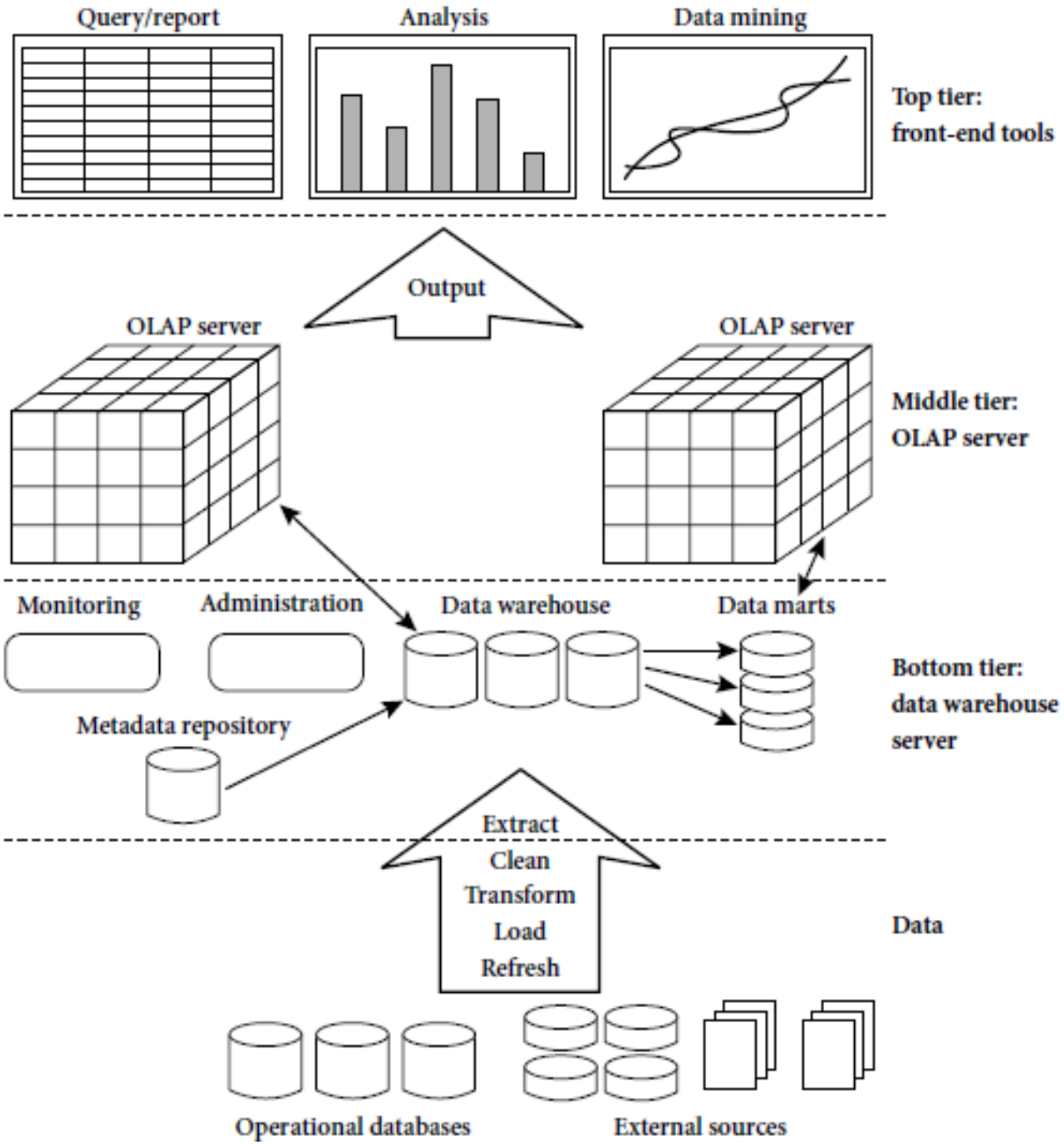
Three–tier Data warehouse architecture

Presentation Layer

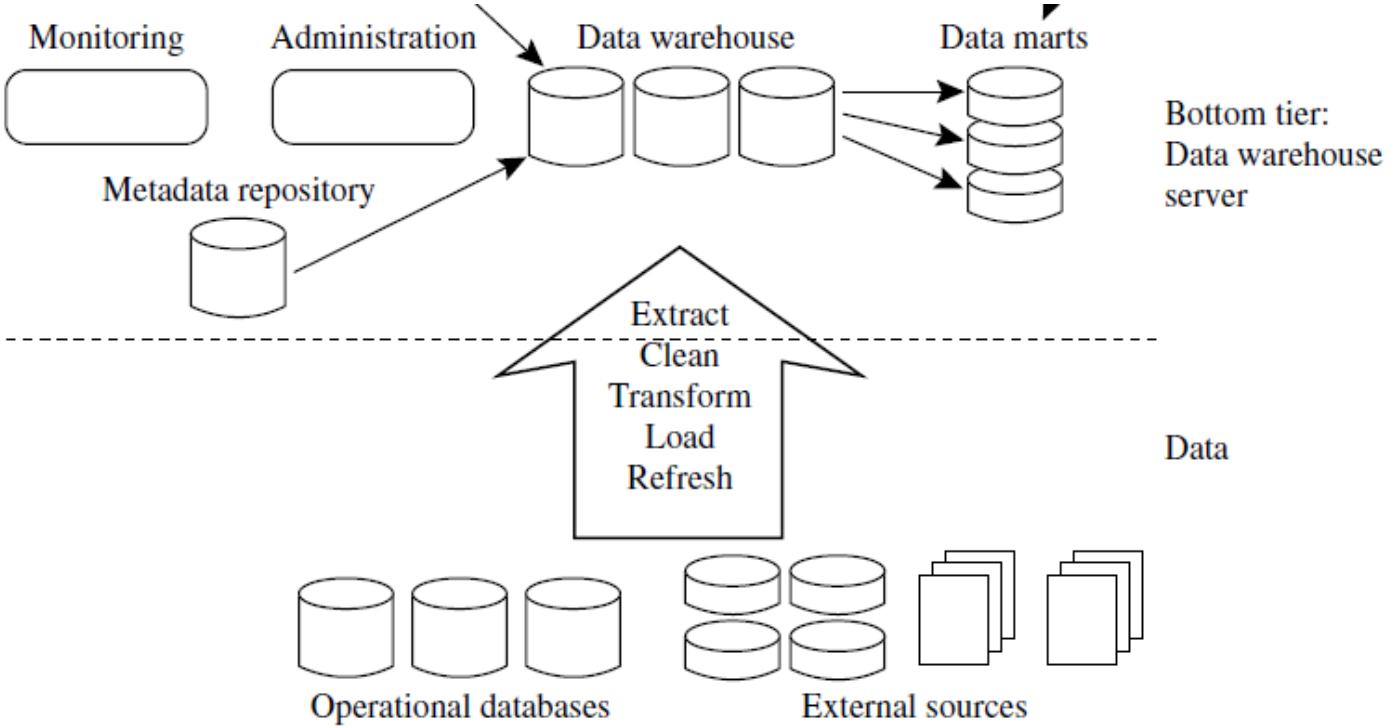
Data warehouse Layer

Data Integration Layer

Source Layer



Data warehouse architecture



- Bottom tier
  - The bottom tier is a **warehouse database server**
  - Back-end tools and utilities are used to feed data into the bottom tier from operational databases
  - These tools and utilities perform data extraction, cleaning, and transformation as well as load and refresh functions to update the data warehouse

Data warehouse architecture

- Bottom Tier
  - The data are extracted using API known as gateways.
  - A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server (Ex ODBC, JDBC)
  - This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

Data warehouse architecture
Extraction, Transformation, and Loading (ETL)

- Data warehouse systems use back-end tools and utilities to populate and refresh their data.
- These tools and utilities include the following functions:

  Data extraction

  Data cleaning

  Data transform

  Load

  Refresh

Data warehouse architecture - ETL

- **Data extraction**
  - get data from multiple, heterogeneous, and external sources
- **Data cleaning**
  - detect errors in the data and rectify them when possible
- **Data transformation**
  - convert data from legacy or host format to warehouse format
- **Load**
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

Data warehouse architecture - Metadata Repository

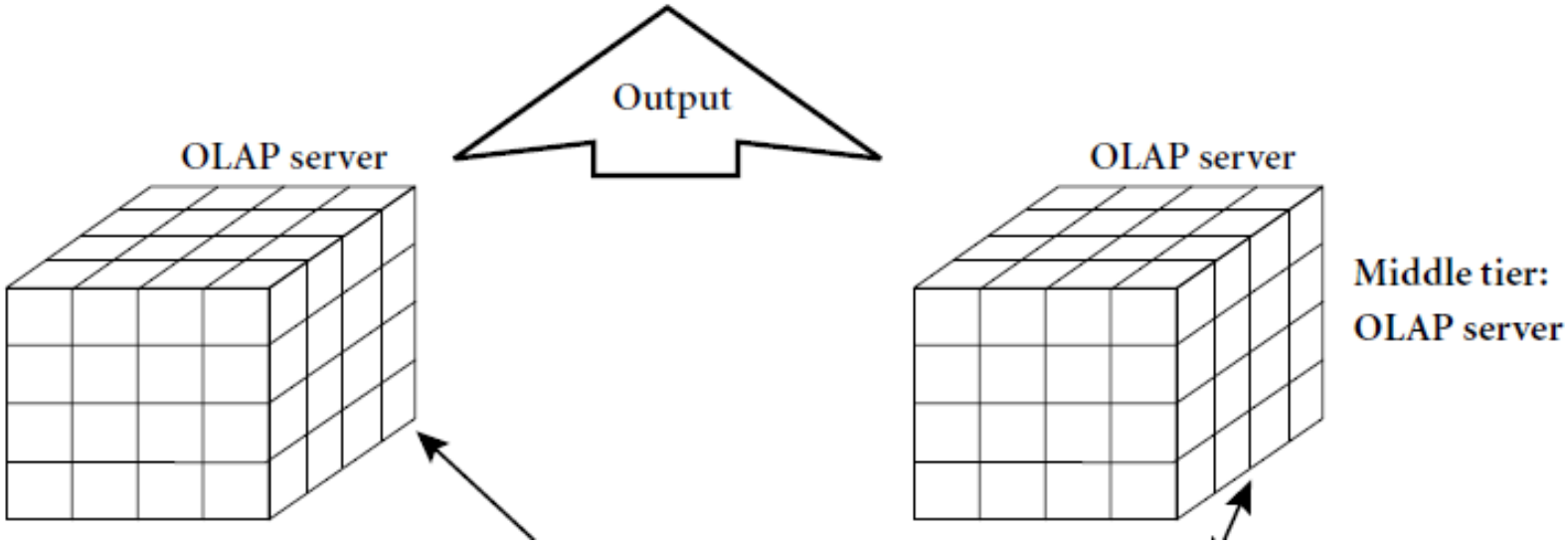
- **Metadata** is the data defining warehouse objects. It stores:
  - Description of the structure of the data warehouse
    - schema, view, dimensions, hierarchies, derived data defn, data mart locations and contents
  - Operational meta-data
    - data lineage (history of migrated data and transformation path), monitoring information (warehouse usage statistics, error reports, audit trails)
  - The algorithms used for summarization
    - include measure and dimension definition algorithms, data on subject areas, aggregation, summarization, and predefined queries and reports.

Data warehouse architecture - Metadata Repository

- **Meta data** is the data defining warehouse objects. It stores:
  - The mapping from operational environment to the data warehouse
    - Includes source databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and data refresh
  - Data related to system performance
    - which include indices and profiles that improve data access and retrieval performance, warehouse schema, view and derived data definitions
  - Business data
    - business terms and definitions, ownership of data, charging policies

Data warehouse architecture

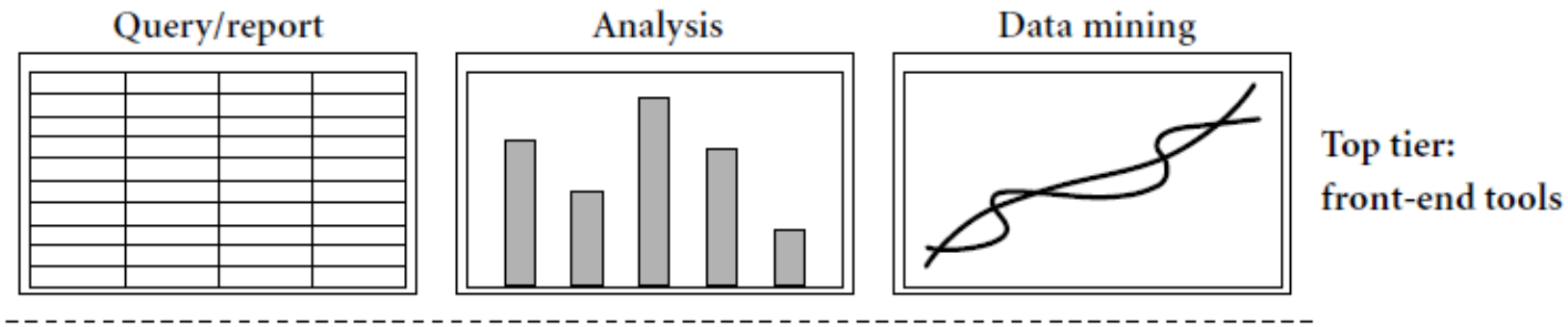
- **Middle Tier**



- It presents the users a **multidimensional data from data warehouse** or data marts.
  - A relational OLAP (ROLAP) model, presents data in relational tables
  - A multidimensional OLAP (MOLAP) model, presents data in array based structure means map directly to data cube array structure

Data warehouse architecture

- **Top Tier**



  - The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on).

Data Warehouse Models:

- From the architecture point of view, there are three data warehouse models:
  - the *enterprise warehouse,*
  - *the data mart, and*
  - *the virtual warehouse.*

Data Warehouse Models:

- **Enterprise warehouse**
  - An enterprise data warehouse is a unified database that
    - Collects all of the information about subjects spanning the entire organization makes it accessible all across the company.
    - Ability to classify data according to subject and give access according to those divisions (sales, finance, inventory and so on)

Data Warehouse Models:

- **Data mart:**
  - A data mart is focused on a single functional area of an organization and contains a subset of data stored in a Data Warehouse.
  - Its scope is confined to specific, selected groups, E.g., Marketing, Sales, HR or finance
  - Depending on the source of data, data marts can be
    - ***Independent data marts*** are sourced from data captured from one or more operational systems or external information providers
    - ***Dependent data marts*** are sourced directly from enterprise data warehouses.

Data Warehouse Models:

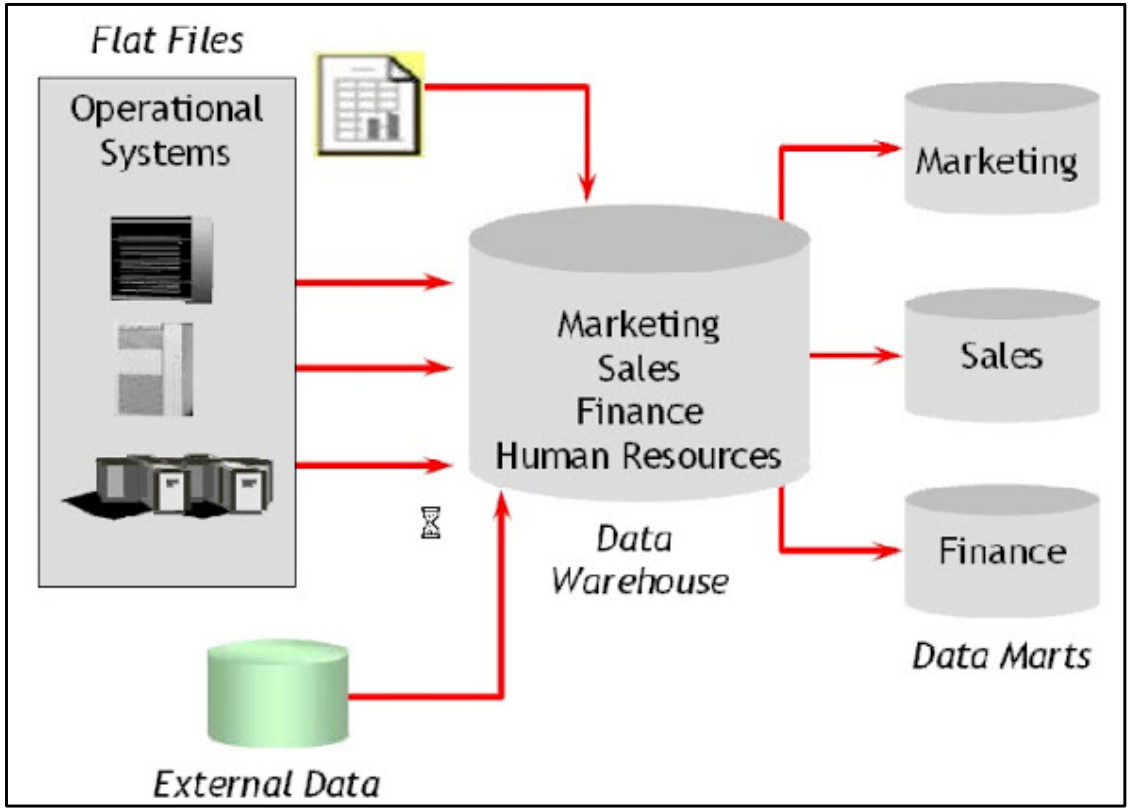
- **Virtual warehouse**
  - A virtual warehouse is a set of views over operational databases.
  - Virtual warehouse have a logical description of all the database and their structure

Data Warehouse Design Process

- Top-down: Starts with overall design and planning (mature)
- Bottom-up: Starts with experiments and prototypes (rapid)

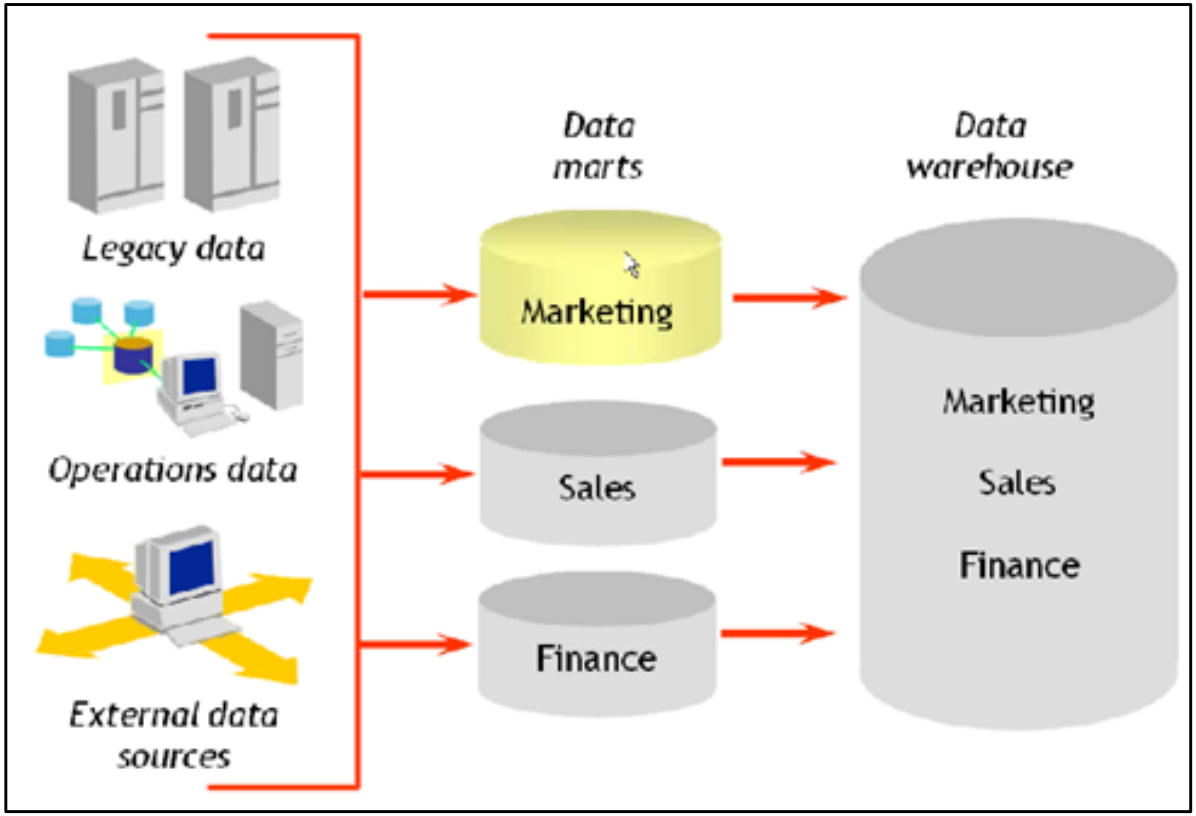
Top-down approach

- The top-down approach
  - It is useful in cases where the technology is mature and well known, and where the business problems that must be solved are clear and well understood.
  - In this approach the, data warehouse is built first. The data marts are then created from the data warehouse



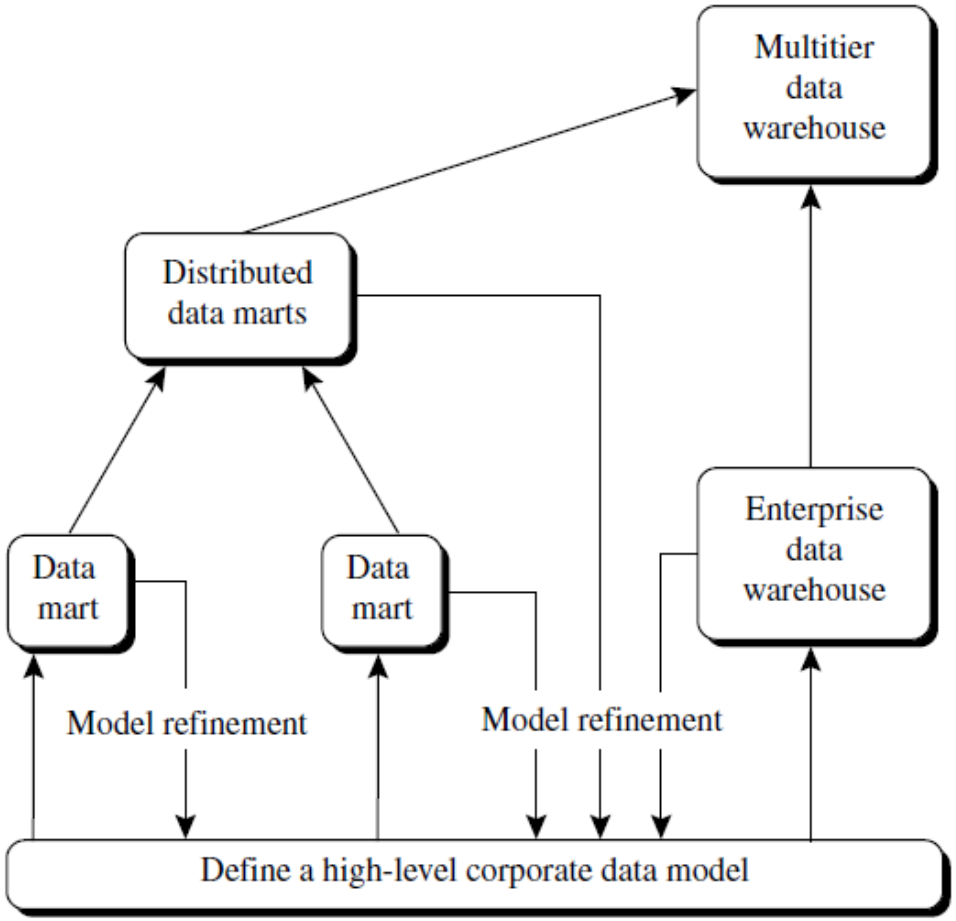
Bottom-up Approach

- The bottom-up approach
  - This is useful in the early stage of business modelling and technology development.
  - In the bottom-up design approach, the data marts are created first to provide reporting capability. A data mart addresses a single business area such as sales, Finance etc

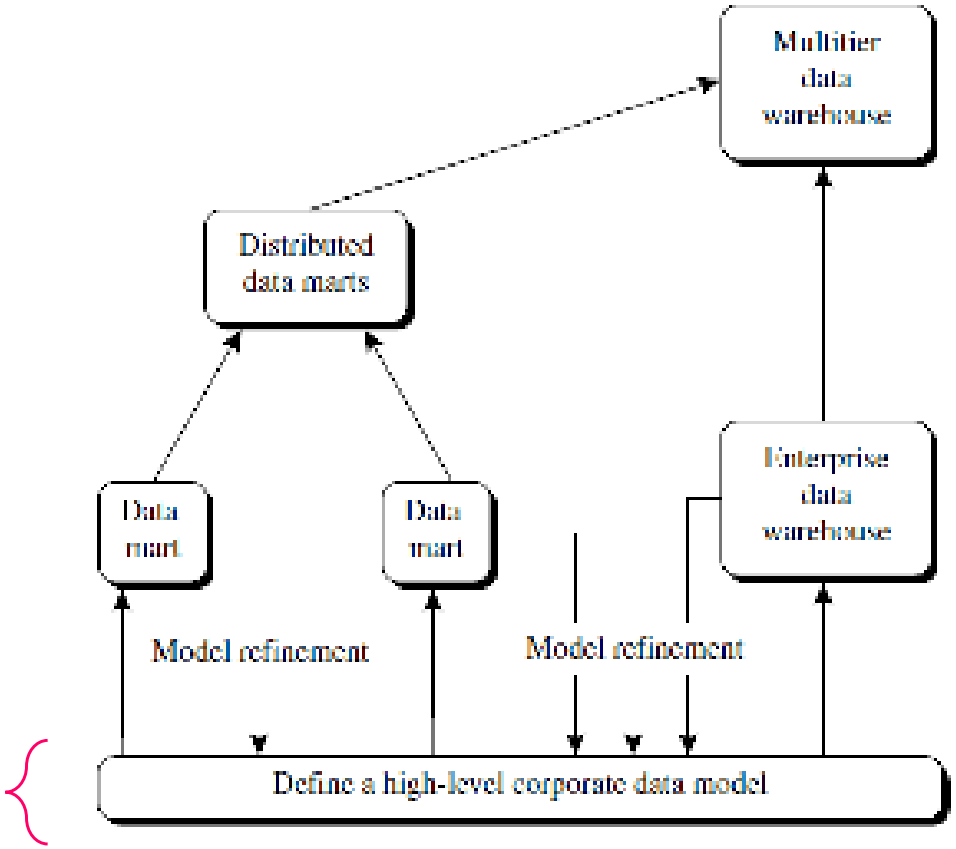


Data Warehouse Development: A Recommended Approach

- A recommended method for the development of data warehouse systems is
  - to implement the warehouse in an incremental and evolutionary mann
  - As shown in figure

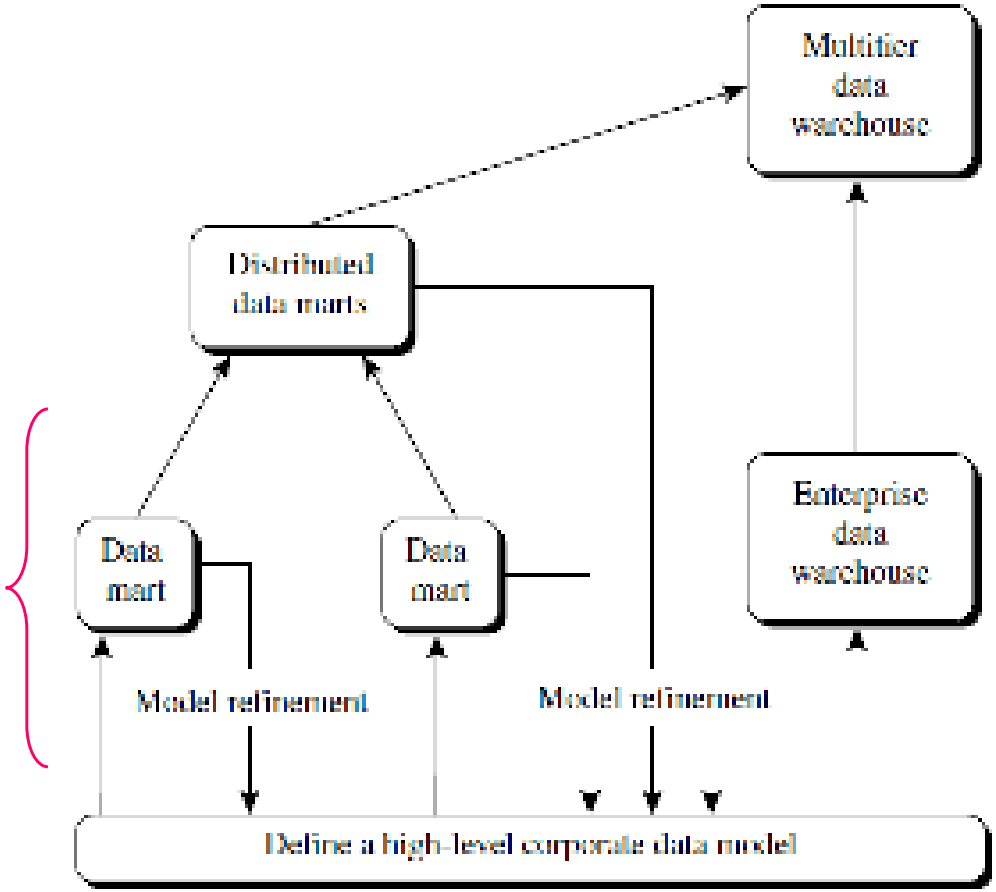


Data Warehouse Development: A Recommended Approach



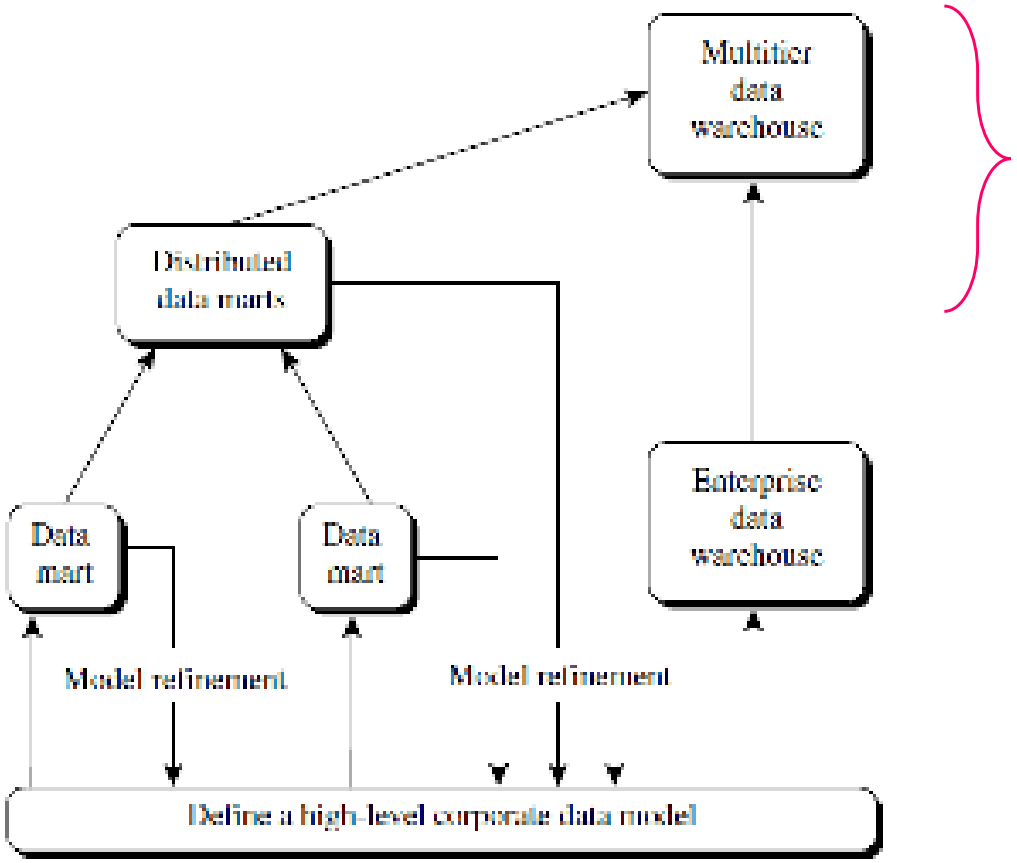
- First, a high-level corporate data model is defined that provides a corporate-wide, consistent, integrated view of data
- This high-level model, need to be refined in the further development of enterprise data warehouses and departmental data marts

Data Warehouse Development: A Recommended Approach



- Second, independent data marts can be implemented in parallel with the enterprise
- Third, distributed data marts can be constructed to integrate different data marts

Data Warehouse Development: A Recommended Approach



- Finally, a multitier data warehouse is constructed custodian of all warehouse data .

A Multidimensional Data Model

- Data warehouses and OLAP tools are based on a multidimensional data model.
- This model views data in the form of a *data cube*.
- *In this section, you will learn how data* cubes model *n-dimensional data.*
- *You will also learn about concept hierarchies and how* they can be used in basic OLAP operations to allow interactive mining at multiple levels of abstraction.

From Tables and Spreadsheets to Data Cubes

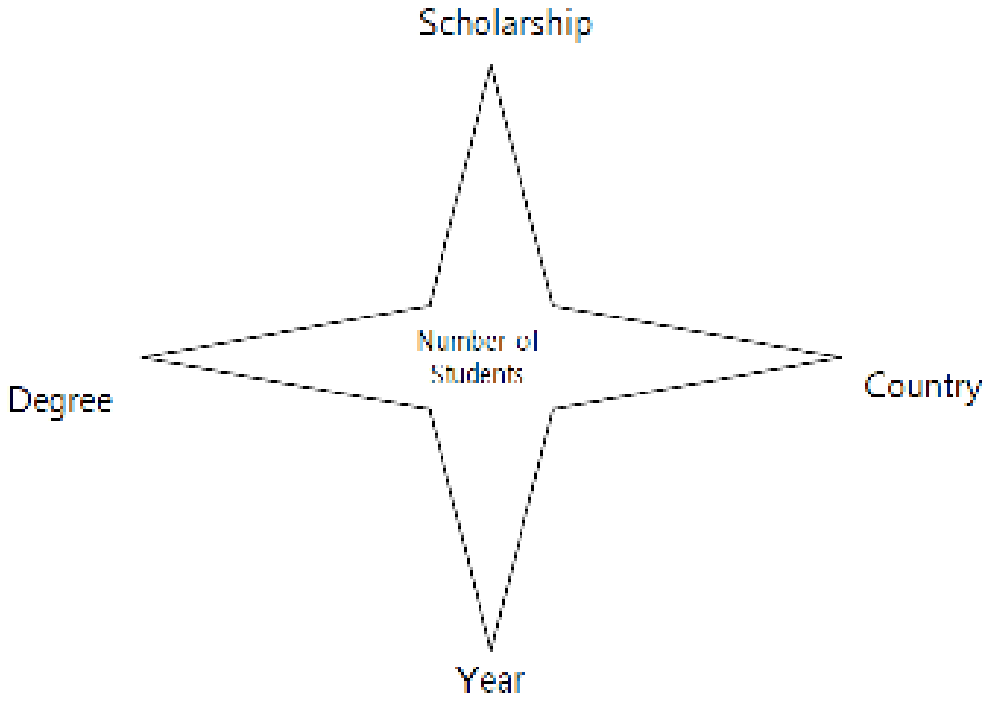
- *What is a data cube?"*
  - *A data cube allows data to be modeled and viewed in multiple* dimensions. It is defined by dimensions and facts
- In general terms, dimensions are the perspectives or entities with respect to which an organization wants to keep records
- For example,*AllElectronics may create a sales* data warehouse in order to keep records of the store's sales with respect to the dimensions *time, item, branch, and location.*
- *These dimensions allow the store to* keep track of things like monthly sales of items and the branches and locations

Data warehouse design

- A data warehouse model often consists of a central fact table and a set of surrounding dimension tables on which the facts depend.
- Such a model is called a star schema because of the shape of the model representation

Data warehouse design

- A simple example of such a schema is shown in F



- For a university where we assume that the number of students is given by the four dimensions – degree, year, country and scholarship.
- These four dimensions were chosen because we are interested in finding out how many students come to each degree program, each year, from each country under each scholarship scheme

Data warehouse design

- A characteristic of a star schema is that all the dimensions directly link to the fact table.
- The fact table may look like table 1 and the dimension tables may look Tables 2 to 5.

Table 1 An example of the fact table

Year	*Degree name*	*Country name*	*Scholarship name*	*Number*
2003	BSc	Australia	Govt	35
1999	MBBS	Canada	None	50
2000	LLB	USA	ABC	22
1999	BCom	UK	Commonwealth	7
2001	LLB	Australia	Equity	2

Data warehouse design

- The first dimension is the degree dimension. An example of this dimension table is Table 2.

Table 2 An example of the degree dimension table

Name	*Faculty*	*Scholarship eligibility*	*Number of semesters*
BSc	Science	Yes	6
MBBS	Medicine	No	10
LLB	Law	Yes	8
BCom	Business	No	6
LLB	Arts	No	6

Data warehouse design

- We now present the second dimension, the country dimension. An example of this dimension table is Table 3.

Table 3 An example of the country dimension table

Name	*Continent*	*Education Level*	*Major religion*
Nepal	Asia	Low	Hinduism
Indonesia	Asia	Low	Islam
Norway	Europe	High	Christianity
Singapore	Asia	High	NULL
Colombia	South America	Low	Christianity

Data warehouse design

- The third dimension is the scholarship dimension. The dimension table is given in Table 4

Table 4 An example of the scholarship dimension table

Name	*Amount (%)*	*Scholarship eligibility*	*Number*
Colombo	100	All	6
Equity	100	Low income	10
Asia	50	Top 5%	8
Merit	75	Top 5%	5
Bursary	25	Low income	12

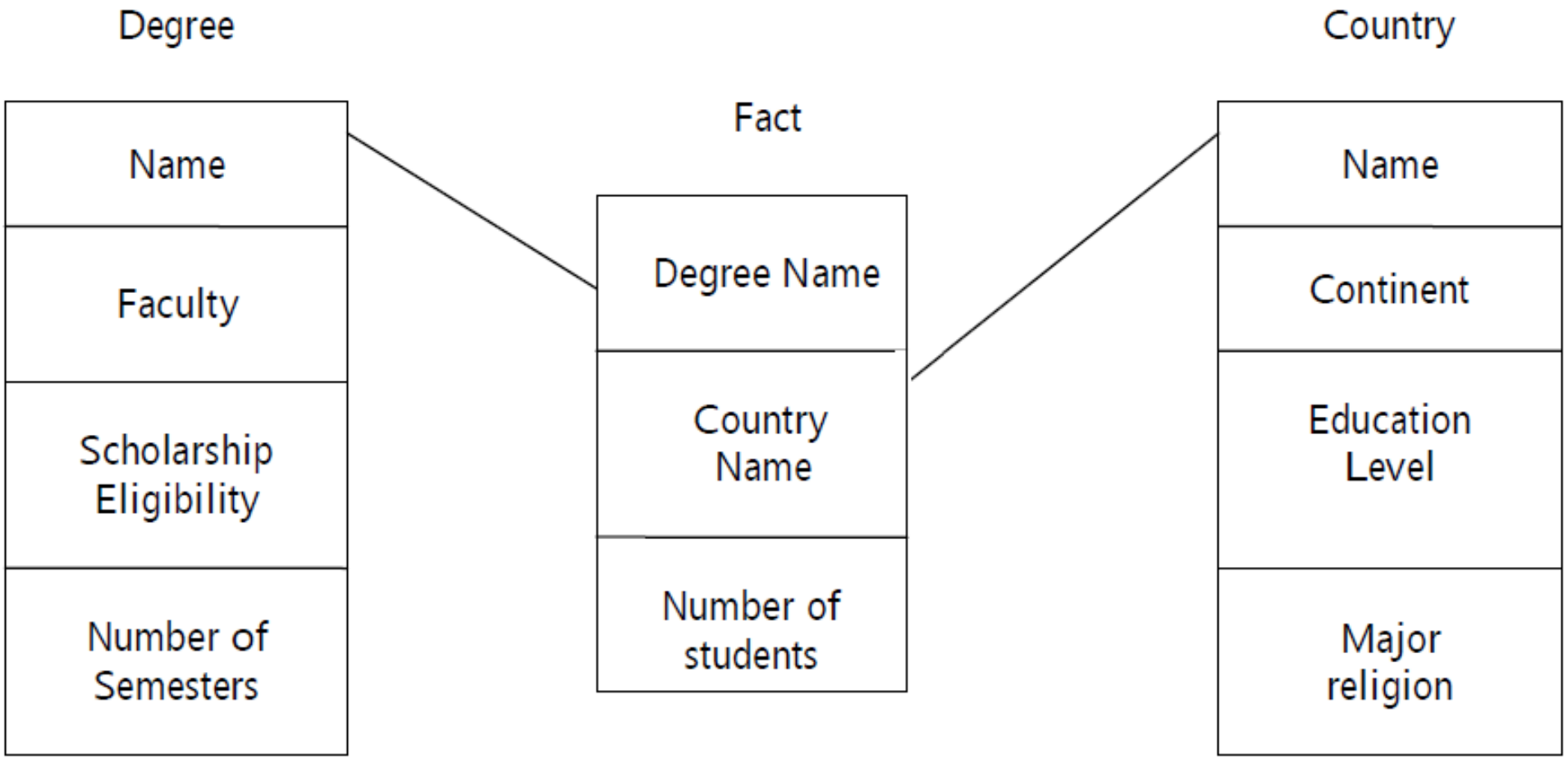
- The fourth dimension is the year dimension. The dimension table is given in Table 7.5.

Table 5 An example of the year dimension table

Year	Degree name	Number
2003	B.Sc	35
1999	MBBS	50
2000	LLB	22
1999	B.Com	7
2001	LLB	2

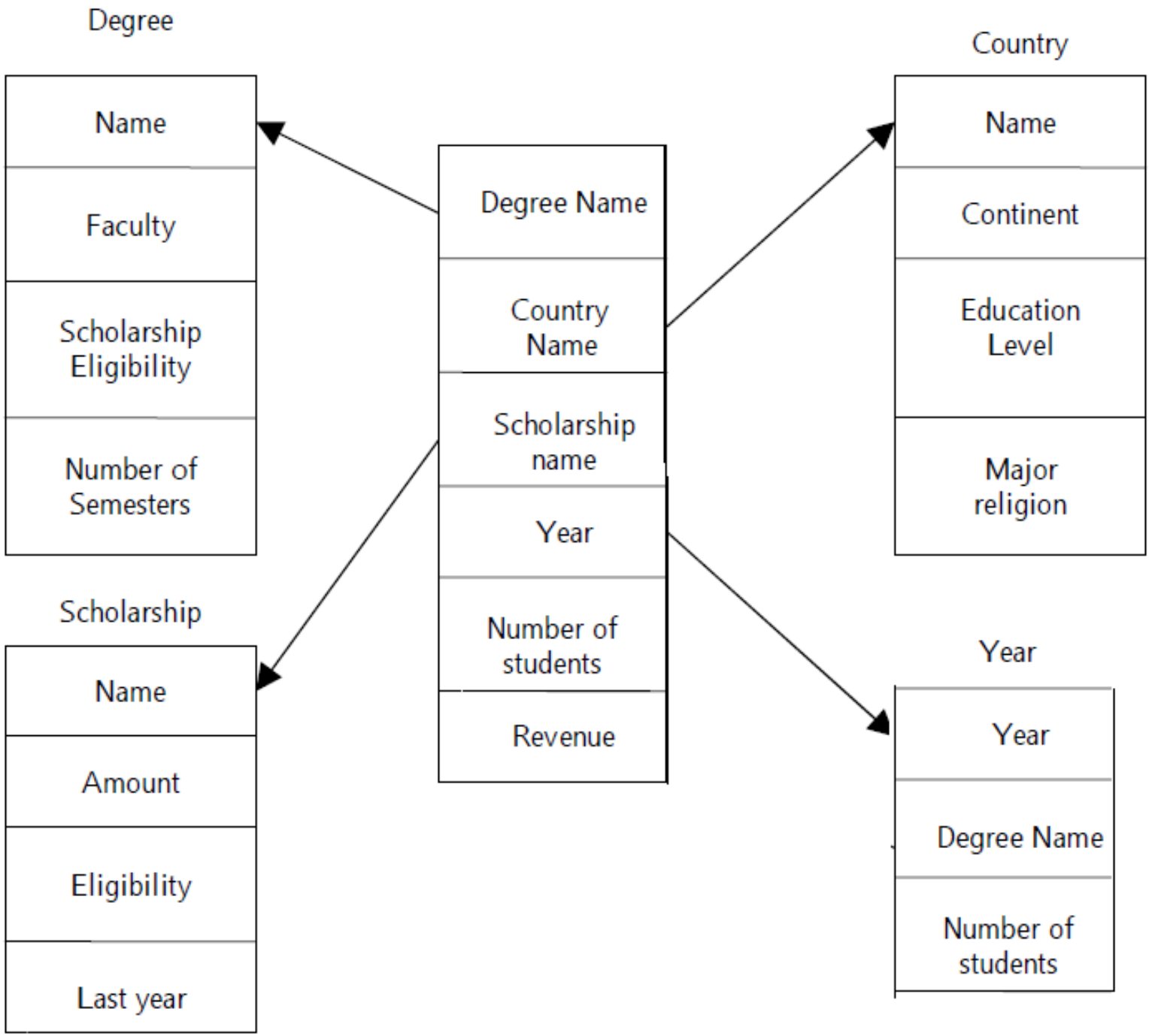
Data warehouse design

- We now present further examples of the star schema. Figure shows a star schema for a two-dimensional example



Data warehouse design

- Figure shows a star schema for a model with four dimensions.

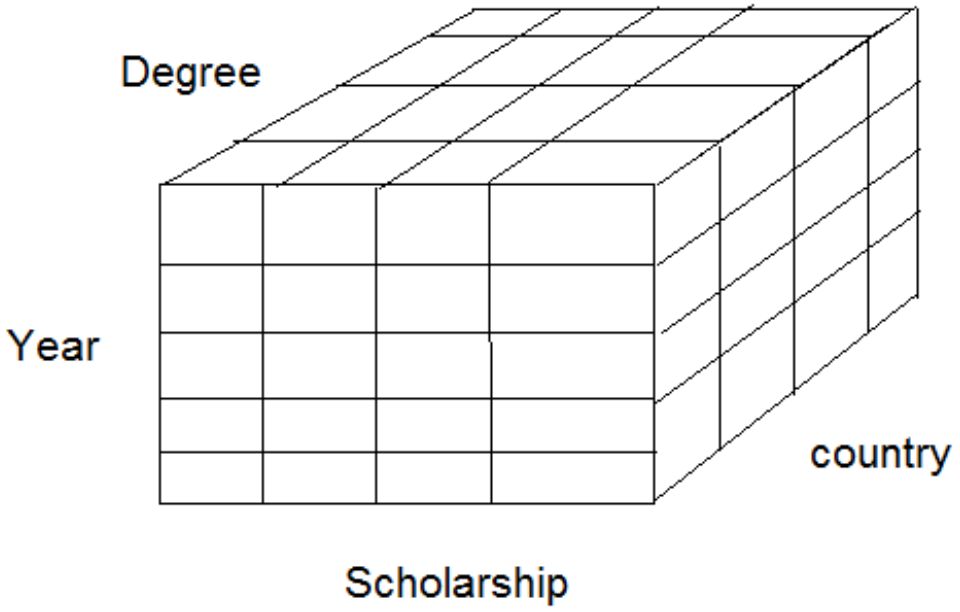


Data warehouse design

- The star schemas are intuitive, easy to understand can be easily extended by adding new attributes or new dimensions.
- Each dimension may be considered an entity in which the primary key is the combination of the foreign keys that refer to the dimensions.
- **Measures** are the core of the dimensional model and are data elements that can be summed, averaged, or mathematically manipulated. Fact tables are at the center of the **star schema**

Data warehouse design

- The dimensional structure of the star schema is called a multidimensional cube in OALP.
- The cubes may be pre-computed to provide very quick response to manage OLAP queries regardless of the size of the data warehouse



Data Cube: A Multidimensional Data Model

- A data cube is defined by facts and dimensions
  - Facts are data which data warehouse focus on
    - Fact tables contain numeric measures (such as dollars_sold) and keys to each of the related dimension tables
  - Dimensions are perspectives with respect to fact
    - Dimension tables describe the dimension with attributes. For example, item (item_name, brand, type), or time(day, week, month, quarter, year)

Exercise

Q1

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

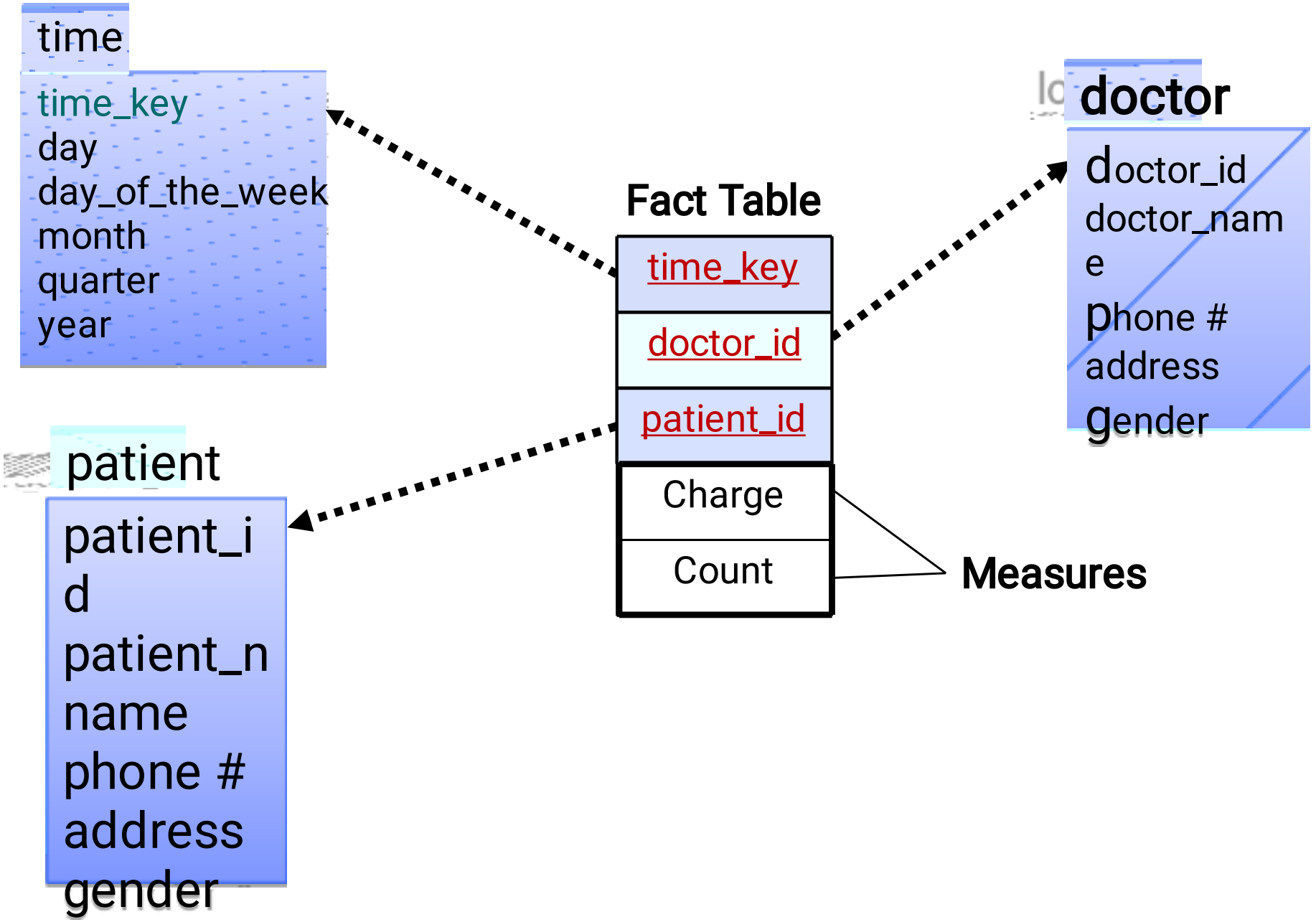
(a) Draw a *star schema diagram*

Q 2

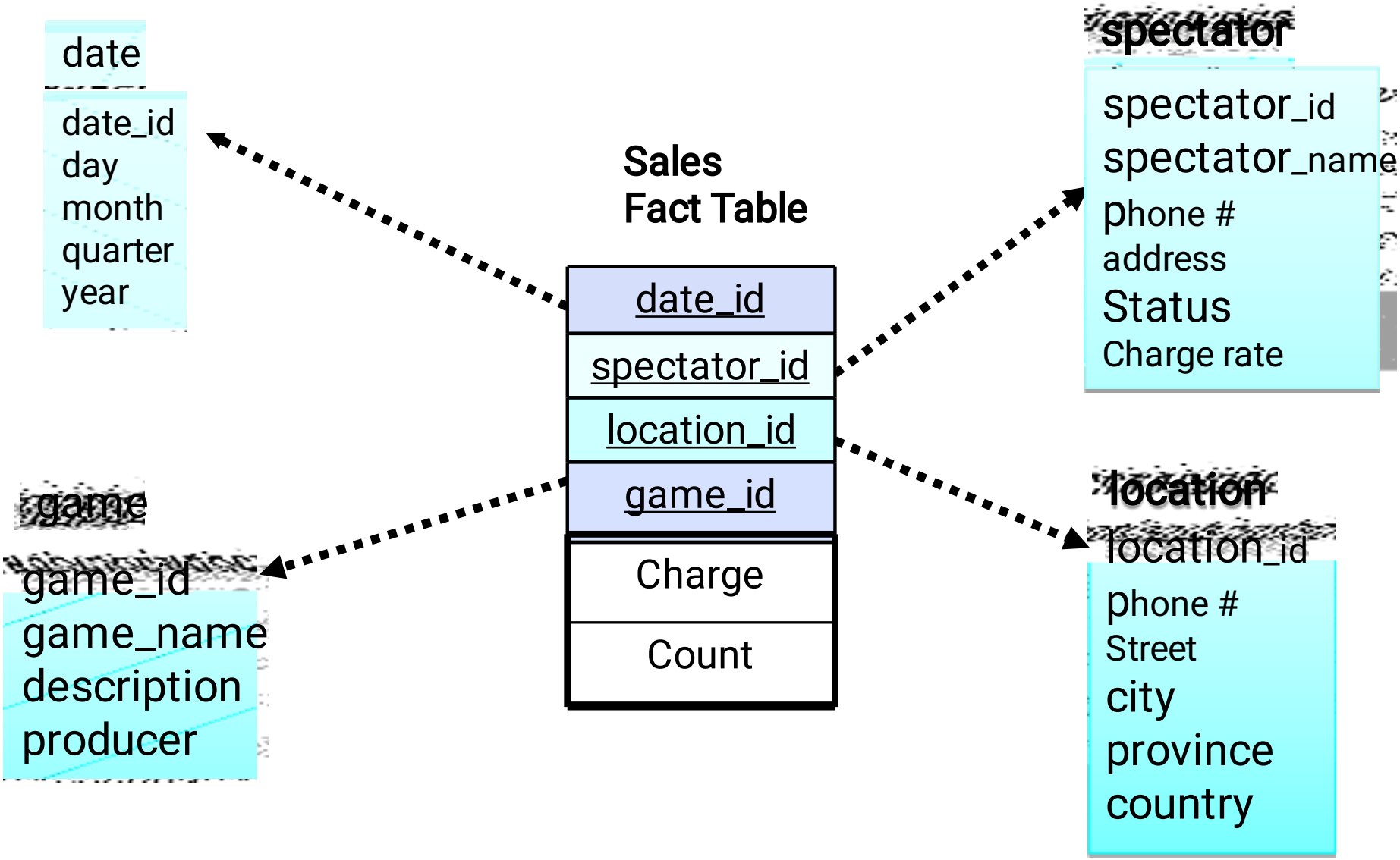
Suppose that a data warehouse consists of the four dimensions, date, spectator, location, and game, and the two measures, count and charge, where charge is the fare that a spectator pays when watching a game on a given date. Spectators may be students, adults, or seniors, with each category having its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

Draw a star schema diagram for the data warehouse.

Q1 Star Schema



Q 2. Star Schema



Data Cube: A Multidimensional Data Model

- To gain a better understanding of data cubes and the multidimensional data model, let's start by looking at a simple 2-D data cube
- Consider sales data from *AllElectronics*
- In particular, we will look at the *AllElectronics sales data for items sold per quarter in the city of Vancouver.*

Data Cube: A Multidimensional Data Model

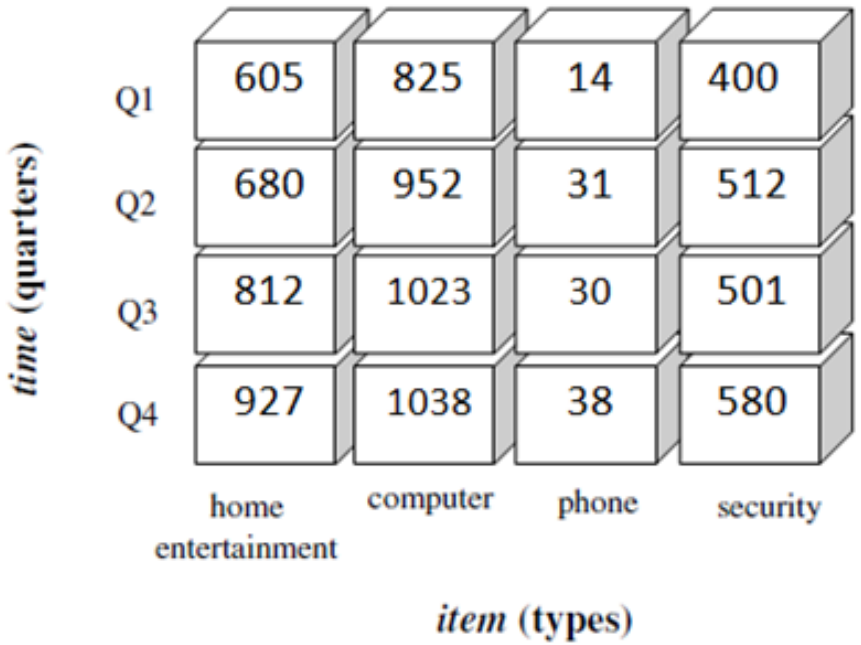
	location = "Vancouver"			
	item (type)			
time (quarter)	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Note: The sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

- In this 2-D representation, the sales for Vancouver are shown with respect to
  - the *time dimension (organized in quarters) and*
  - the *item dimension (organized according to the types of items sold).*
  - *The fact or measure* displayed is *dollars sold (in thousands)*

Data Cube: A Multidimensional Data Model

- Representation *AllElectronics sales data for items sold per quarter in the city of Vancouver*



Data Cube: A Multidimensional Data Model

- Now, suppose that we would like to view the sales data with a third dimension.
  - View the data according to *time* and *item*, as well as *location*, for the cities *Chicago, New York, Toronto, and Vancouver*.
  - These 3-D data are shown in Table ….

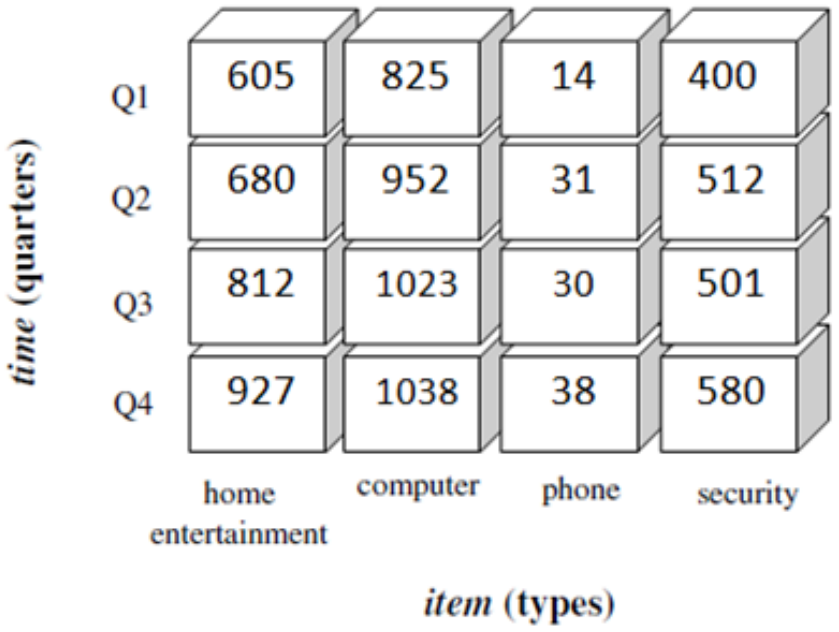
Table 4.3 3-D View of Sales Data for *AllElectronics* According to *time*, *item*, and *location*

	location = "Vancouver"				**location** = "Toronto"				**location** = "New York"				**location** = "Chicago"			
	item				**item**				**item**				**item**			
	home				*home*				*home*				*home*			
time	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*	*ent.*	*comp.*	*phone*	*sec.*
Q1	605	825	14	400	818	746	43	591	1087	968	38	872	854	882	89	623
Q2	680	952	31	512	894	769	52	682	1130	1024	41	925	943	890	64	698
Q3	812	1023	30	501	940	795	58	728	1034	1048	45	1002	1032	924	59	789
Q4	927	1038	38	580	978	864	59	784	1142	1091	54	984	1129	992	63	870

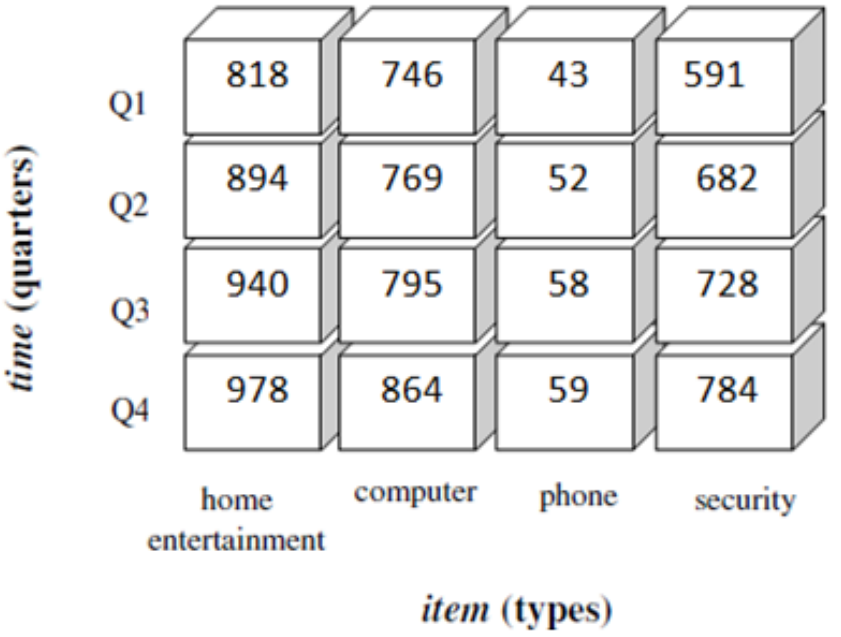
Note: The measure displayed is *dollars_sold* (in thousands).

- The 3-D data in the table are represented as a series of 2-D tables
- Conceptually, we may also represent the same data in the form of a 3-D data cube

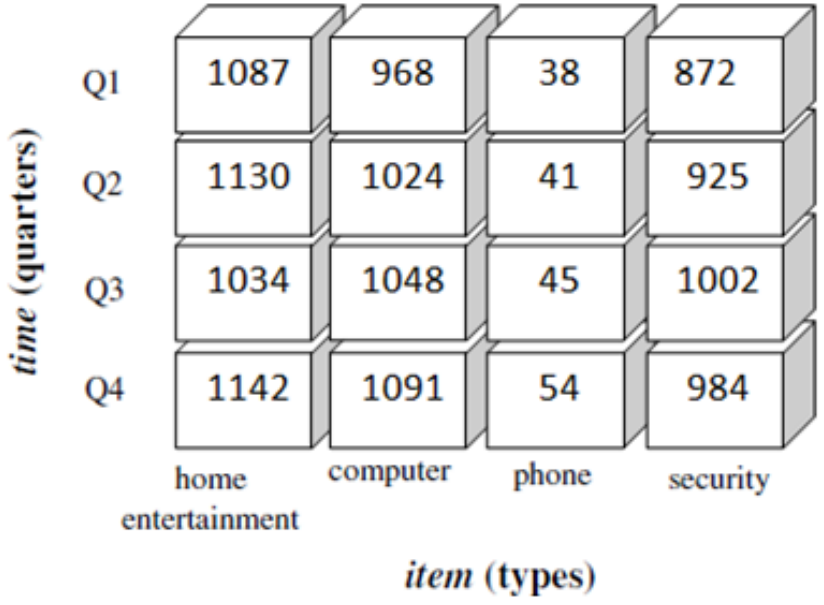
"Vancouver"



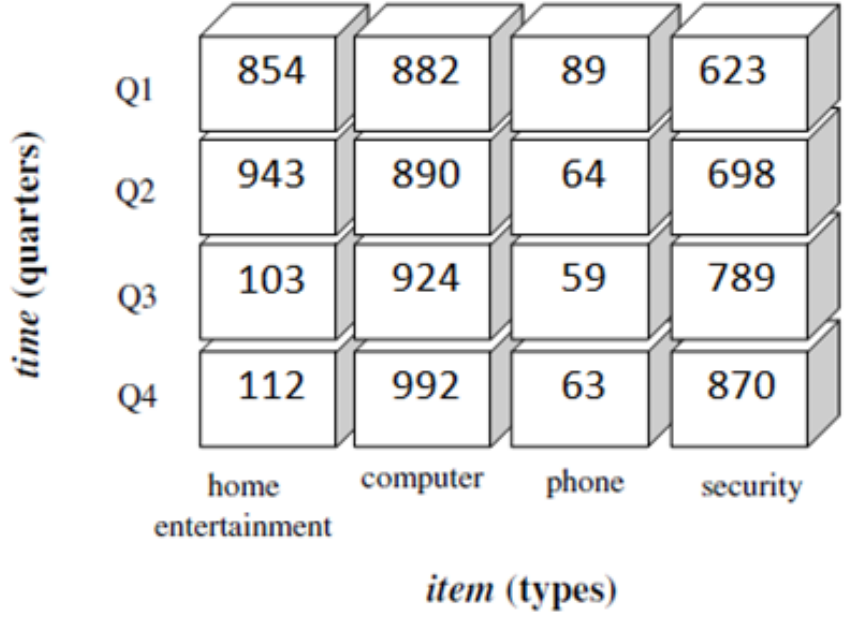
"Toronto"

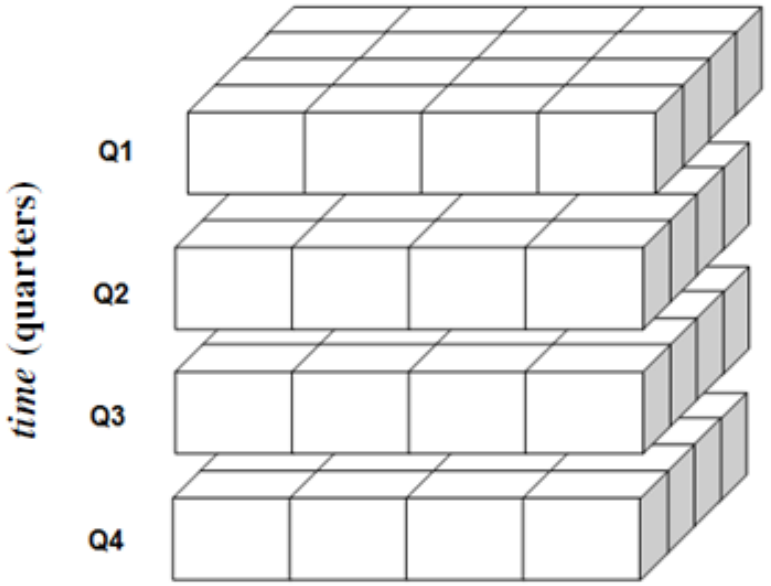


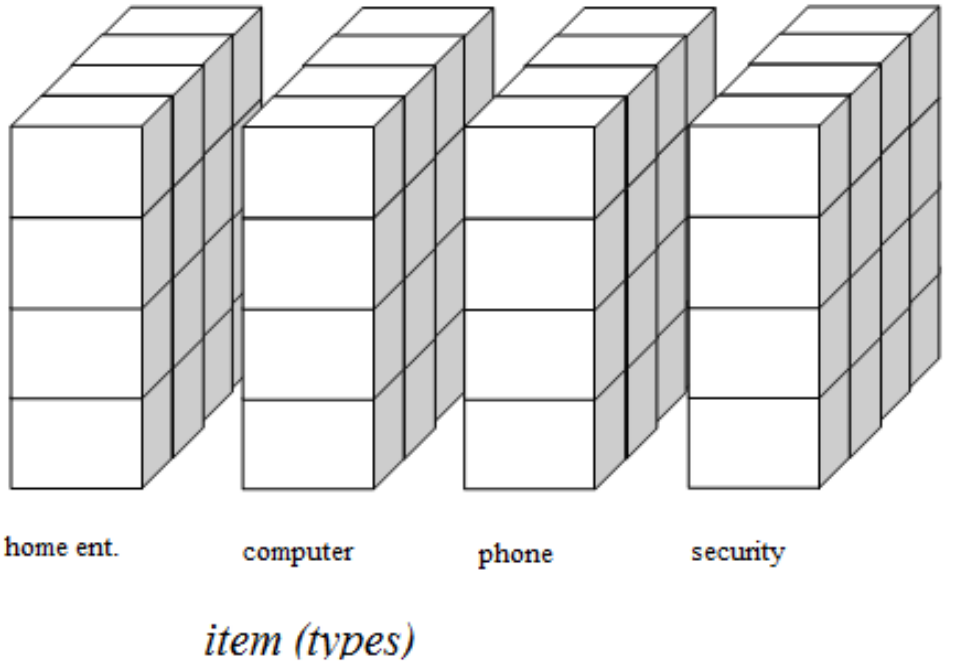
"New York"

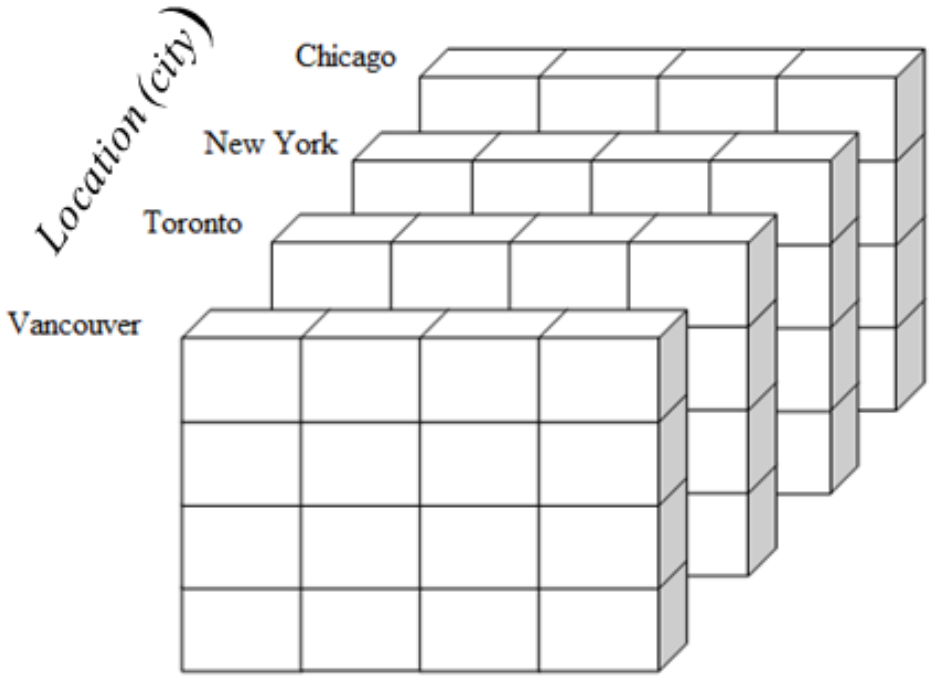


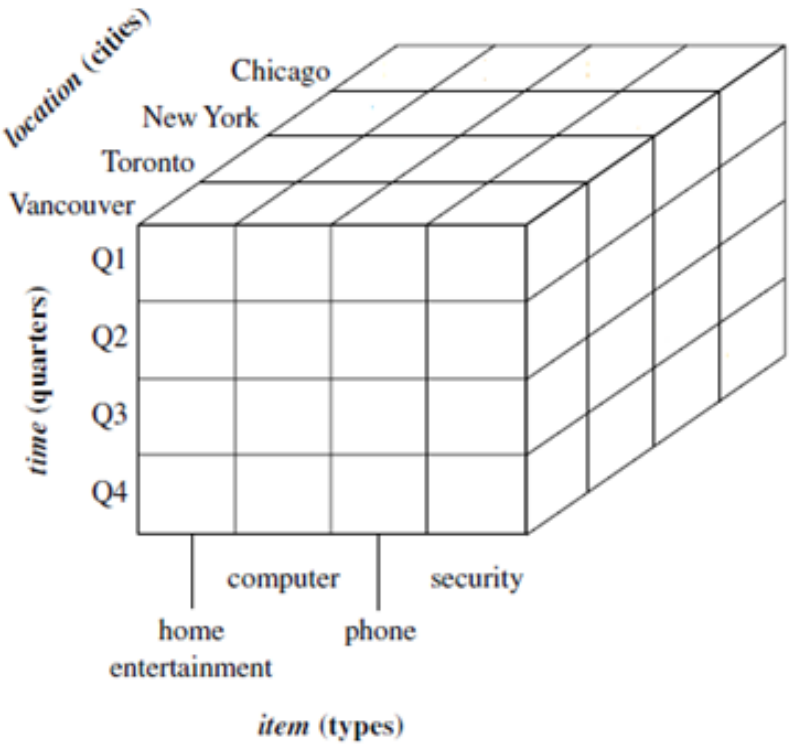
"Chicago"



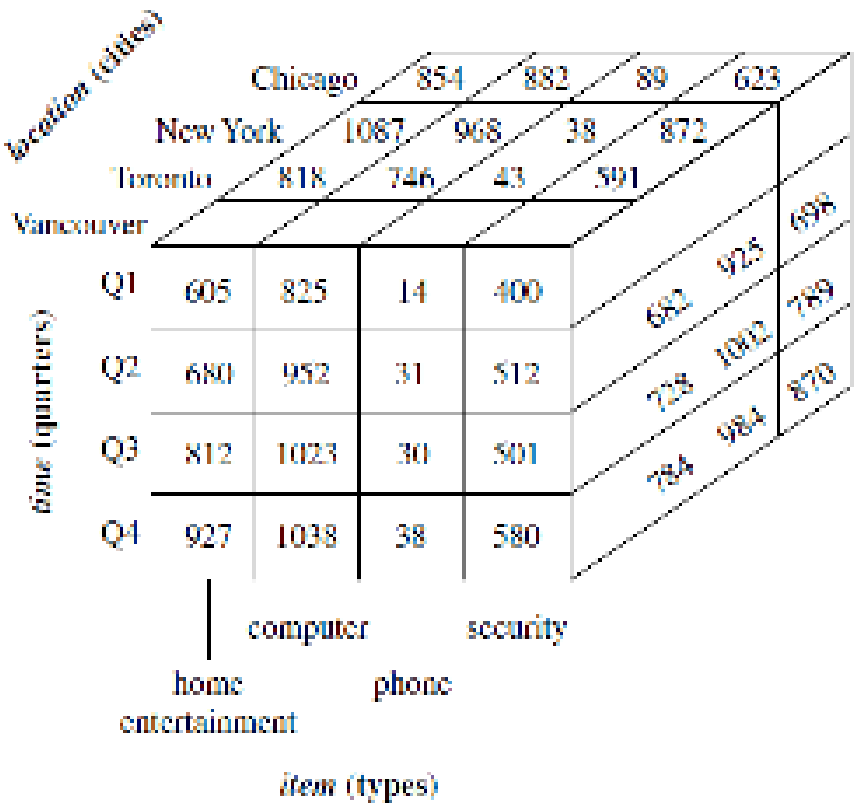

time (quarters)
Q1
Q2
Q3
Q4


home ent.
computer
phone
security
item (types)


Location (city)
Chicago
New York
Toronto
Vancouver

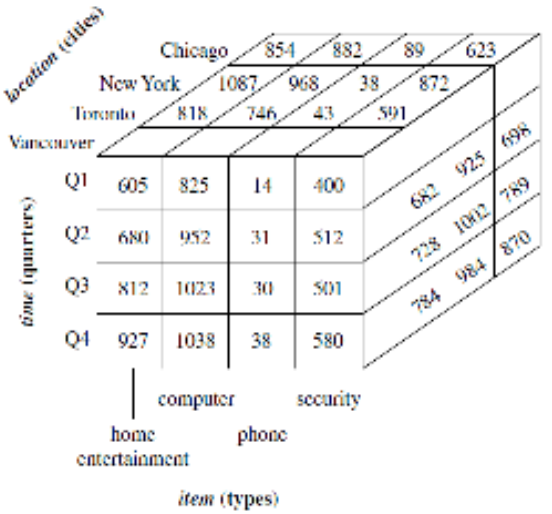

location (cities)
Chicago
New York
Toronto
Vancouver
time (quarters)
Q1
Q2
Q3
Q4
computer
security
home entertainment
phone
item (types)

- Finally, data cube for ALL Electronics ……



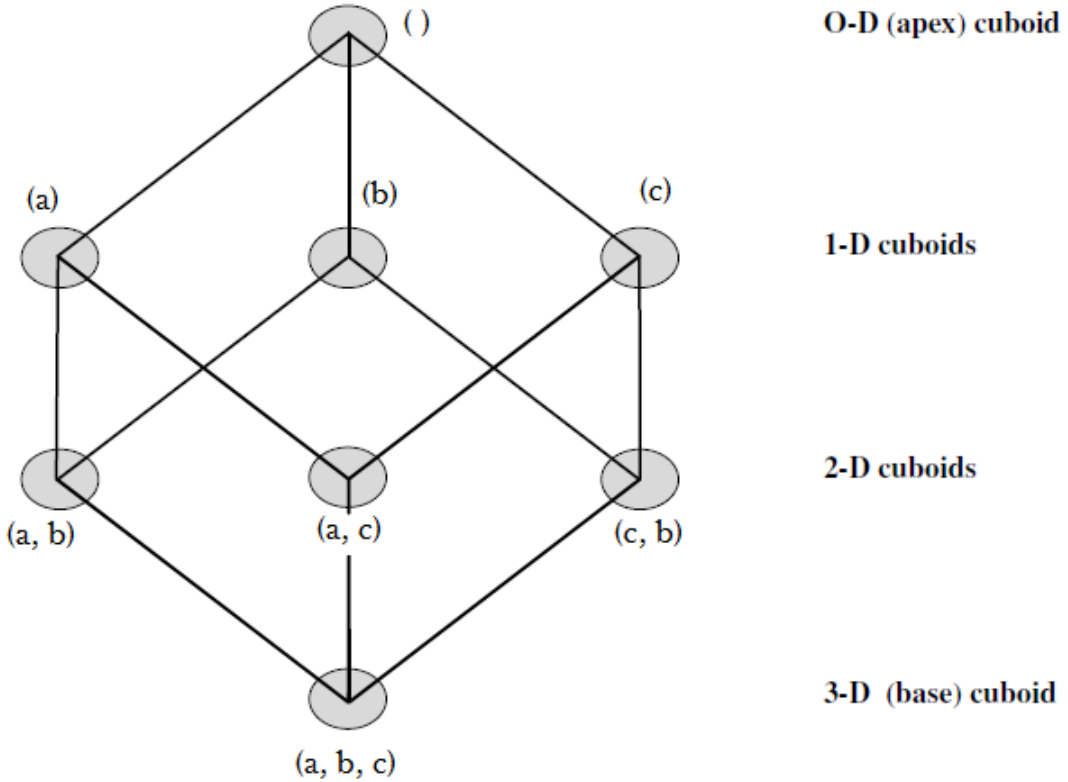
- Data cube is metaphor for multidimensional data storage
- The actual physical storage of such data may differ from its logical representation.
- The important thing to remember is that data cubes are *n-dimensional and do not confine data to 3-D.*

	location = "Vancouver"				location = "Toronto"				location = "New York"				location = "Chicago"			
	item				item				item				item			
time	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	605	825	14	400	818	746	43	591	1087	968	38	872	854	882	89	623
Q2	680	952	31	512	894	769	52	682	1130	1024	41	925	943	890	64	698
Q3	812	1023	30	501	940	795	58	728	1034	1048	45	1002	1032	924	59	789
Q4	927	1038	38	580	978	864	59	784	1142	1091	54	984	1129	992	63	870



- Table show the data at different degrees of summarization
- In the data warehousing research literature, a data cube like those shown in Figures is often referred to as a **cuboid.**
- Given a set of dimensions, we *can generate a cuboid for each of the possible subsets of the given dimensions*
- The result would form a *lattice of* cuboids, each showing the data at a *different level of summarization, or group-by*
- The ***lattice of cuboid*** is then referred to as Data cube as shown in figure…

- Figure shows a 3-D data cube for the dimensions *A, B, and C, and an aggregate measure, M.*
- *Commonly used measures include count(), sum(), min(), max(), and total sales().*



- A data cube is a lattice of cuboids. Each cuboid represents a group-by.
- *ABC is the base* cuboid, containing all three of the dimensions.
  - Here, the aggregate measure, *M, is computed* for each possible combination of the three dimensions
- The base cuboid is the least generalized of all the cuboids in the data cube
- The most generalized cuboid is the apex cuboid, commonly

Number of Cuboids

- How many cuboids are there in an n-dimensional data cube?
  - If there were no hierarchies associated with each dimension, then the total number of cuboids for an *n-dimensional data cube, as we have seen is* 2^n
  - For dimensions (*Product, Region, City*), $2^n = 2^3 = 8$ *cuboids*
  - However, in practice, many dimensions do have hierarchies.
  - For an *n-dimensional* data cube, the total number of cuboids that can be generated including hierarchies is

$$Total\ number\ of\ cuboids = \prod_{i=1}^{n}(L_i + 1),$$

where L_i is the number of levels associated with dimension i. One is added to L_i in Equation to include the *virtual* top level, all.

Efficient Computation of Data Cubes

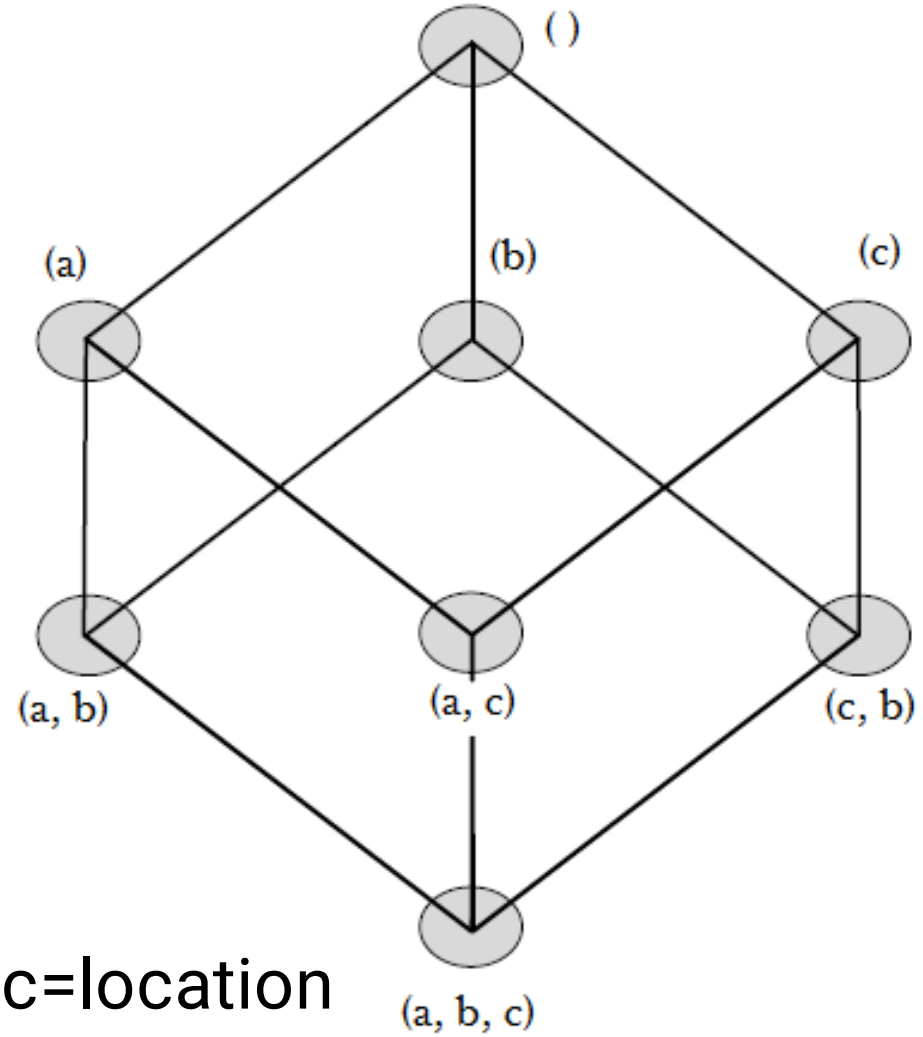
- Figure shows lattice of cuboid forming a data cube for three dimensions (time, item, location)

3 dimension

2^3 = 8 cuboids

the possible group-by's are

{(a,b,c), (a,b), (a,c), (b,c), (a), (b), (c), () }



a = time, b=item, c=location

Example

- Suppose that you would like to create a data cube for *AllElectronics sales that contains the following: city, item, year and sales as measurse in dollars. You* would like to to analyze the data, with queries such as the following:
  - *"Compute the sum of sales, grouping by city and item."*
  - *"Compute the sum of sales, grouping by city."*
  - *"Compute the sum of sales, grouping by item."*

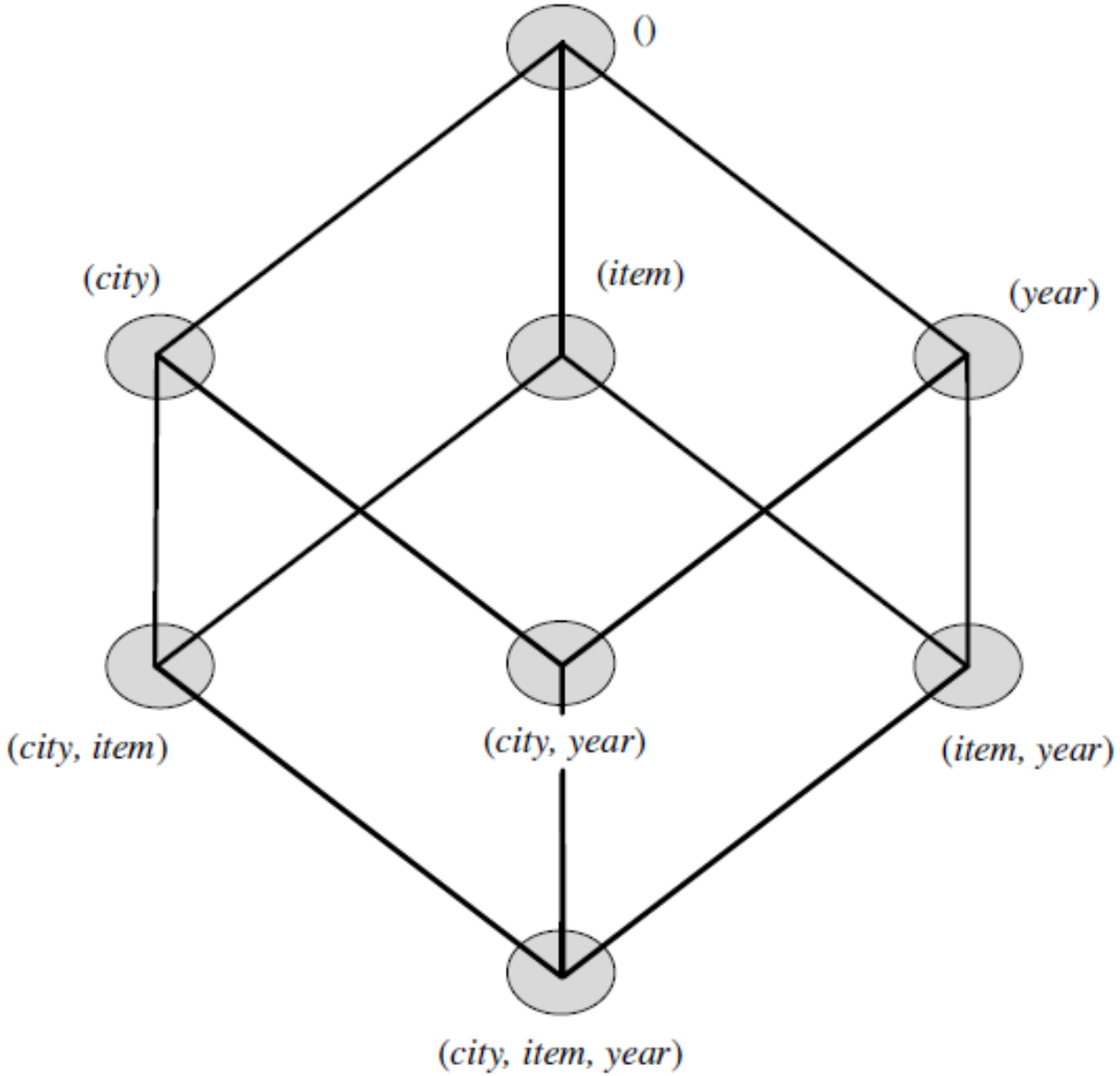
What is the total number of cuboids, or group-by's, that can be computed for this data cube?

- the total number of cuboids, or group by's, that can be computed for this data cube is $2^3 = 8$.
- The possible group-by's are the following:

{ (city, item, year),

(city, item), (city, year), (item, year),

(city), (item), (year), () }

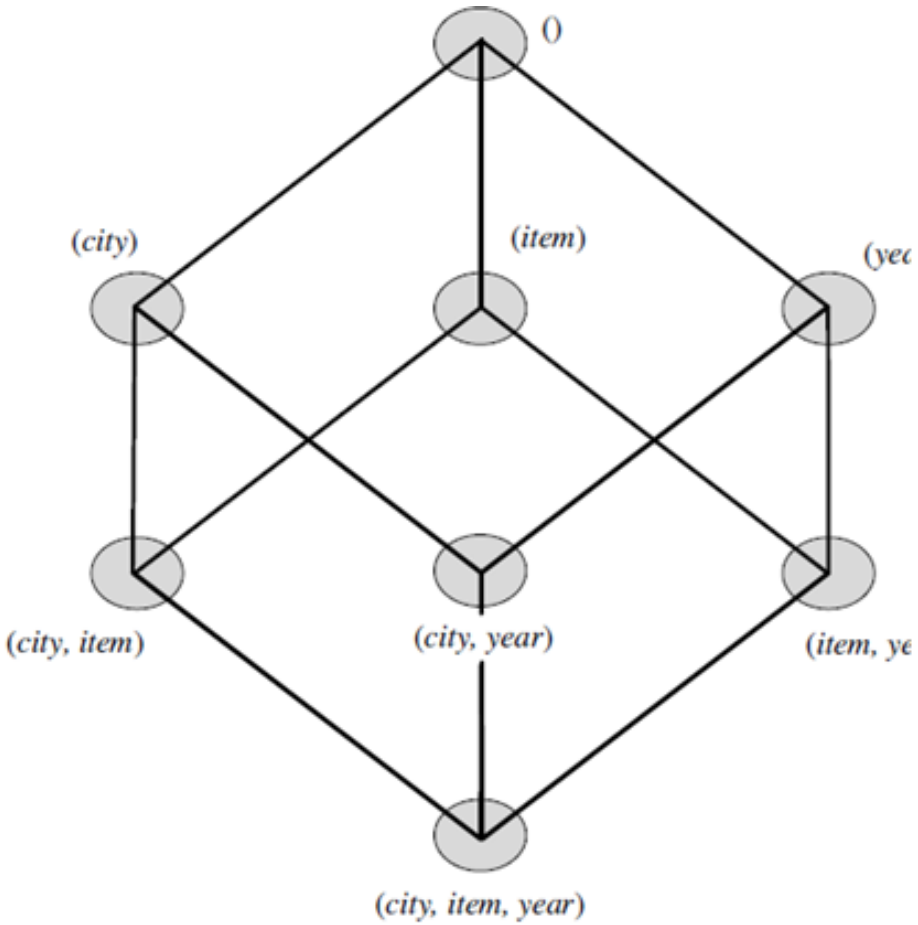


O-D (apex) cuboid

1-D cuboids

2-D cuboids

3-D (base) cuboid



O-D (apex) cuboid

1-D cuboids

2-D cuboids

3-D (base) cuboid

An SQL query containing

no group-by, such as “compute the sum of total sales” is a *zero-dimensional operation*

one group-by, such as “compute the sum of sales, group by city,” is a *one-dimensional operation*

one group-by, such as “compute the sum of sales, group by city and item” is a *two-dimensional operation*

- Construct a lattice of cuboids forming a data cube for the dimensions :*time, item, location, supplier.*
  - What is the total number of cuboids, or group-by's, that can be computed for this data cube?

- Construct a lattice of cuboids forming a data cube for the dimensions :*time, item, location, supplier*.

The possible group-by's are the following:

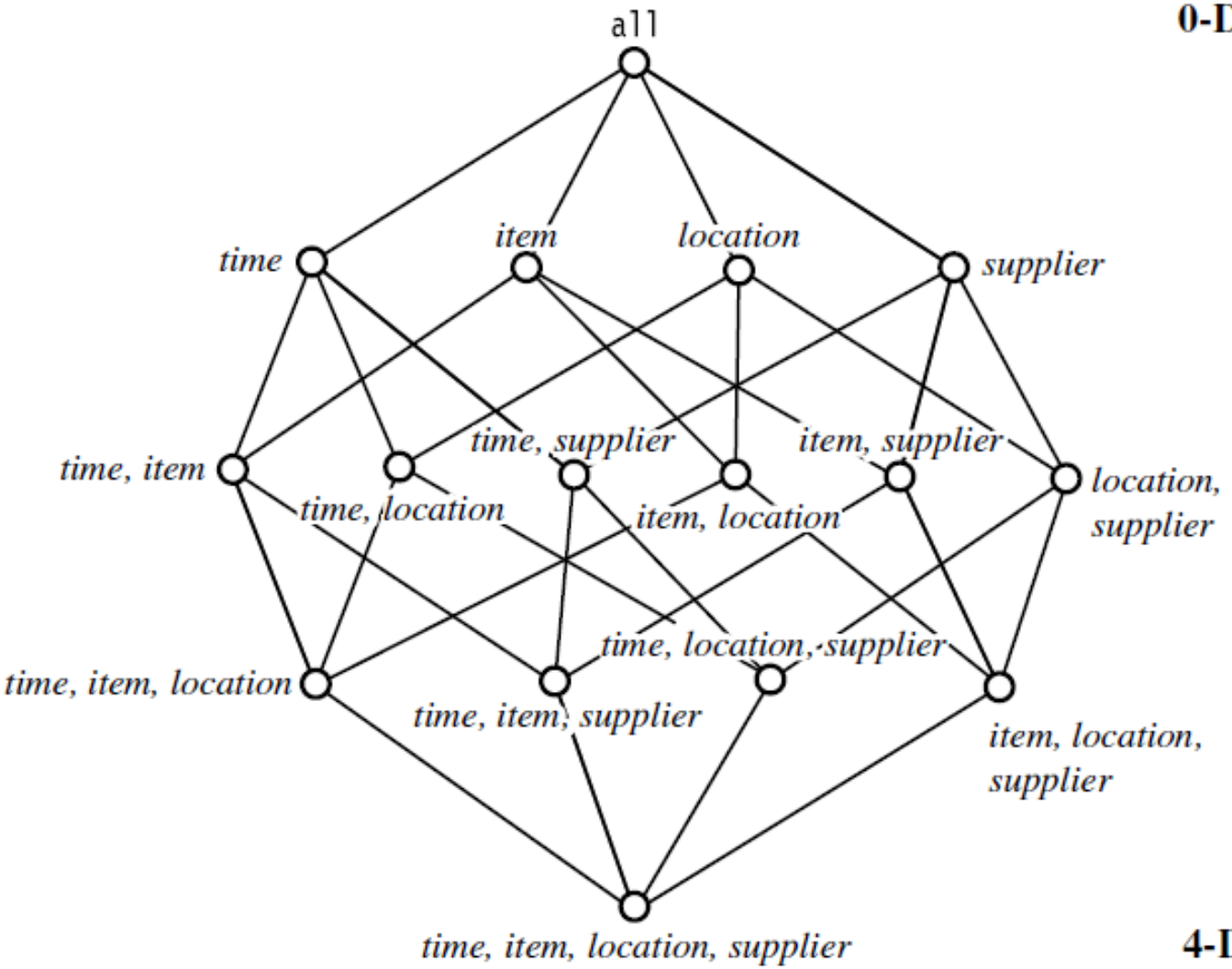
() all

{time} {item} {location} {supplier}

{time, item} {time, location}{time, supplier}, {item, location}
, {item, supplier}, {location, supplier}

{time, item ,location}
{time ,item, supplier}
{time, location, supplier}
{item, location,supplier}

{time, item, location, supplier}


all
time
item
location
supplier
time, item
time, location
time, supplier
item, location
item, supplier
location, supplier
time, item, location
time, item, supplier
time, location, supplier
item, location, supplier
time, item, location, supplier
0-D (apex) cuboid
1-D cuboids
2-D cuboids
3-D cuboids
4-D (base) cuboid

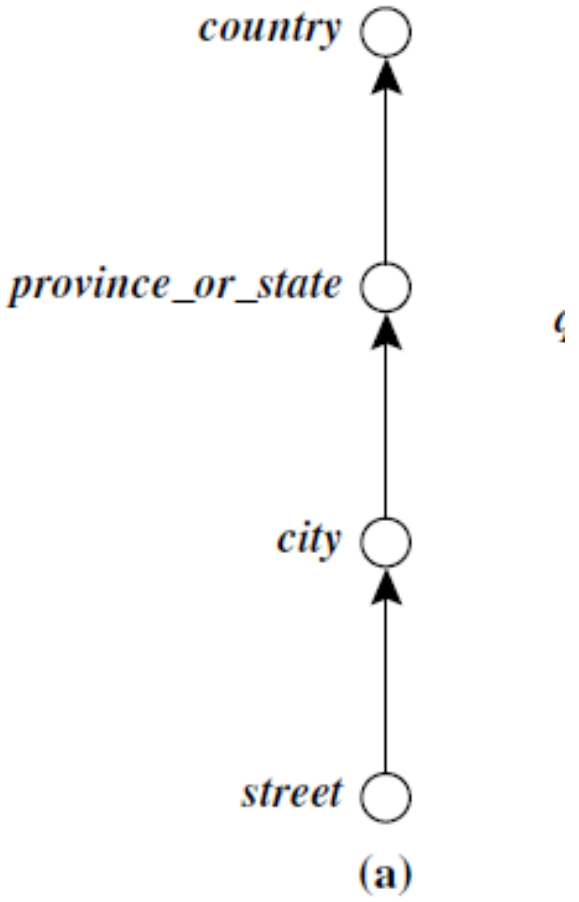
The Role of Concept Hierarchies

- A **concept hierarchy** defines a sequence of mappings **from a set of low-level concepts** to higher-level,
- Concept hierarchy organizes concepts (attribute values) hierarchically and is usually associated with each dimension
- Concept hierarchy facilitate drilling and rolling in data ware houses to view data in multiple granularity
- Hierarchies can be explicitly specified by domain experts and/or data ware house designers
- Concept hierarchies allow data to be handled at varying levels of abstraction,

The Role of Concept Hierarchies

- For example, suppose that the dimension*location is described by the attributes street, city, province or state,* and*country.*
- These attributes are related by a total order, forming a concept hierarchy such as $street < city < province\_or\_state < country$

This hierarchy is shown in Figure



Number of Cuboids

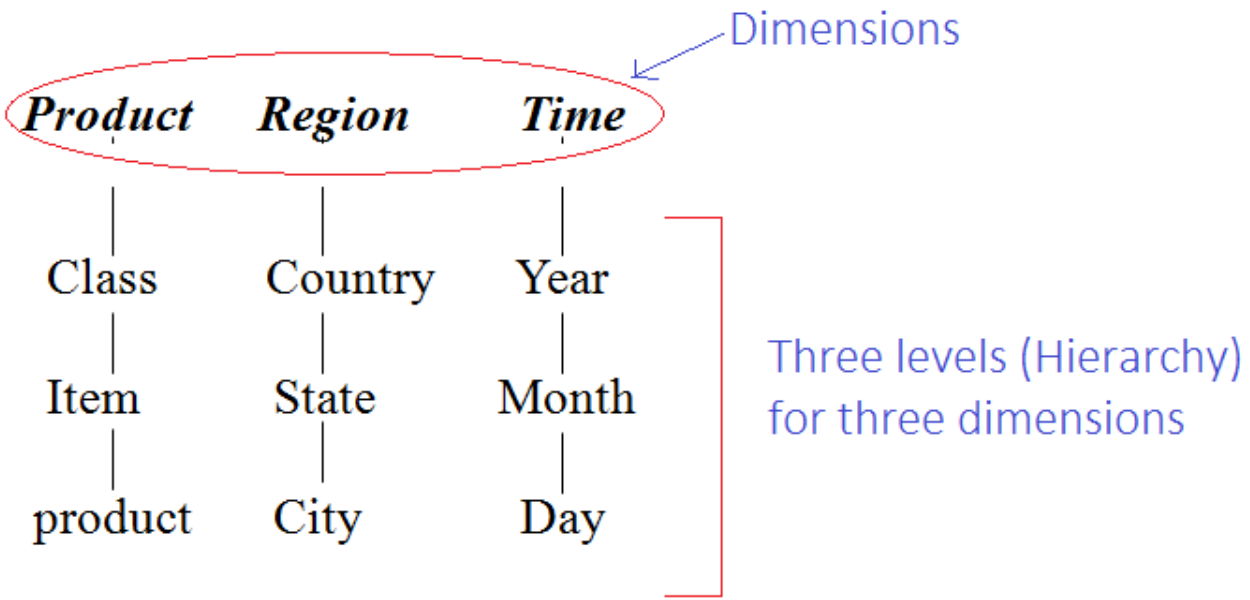
- How many cuboids are there in an n-dimensional data cube?
  - If there were no hierarchies associated with each dimension, then the total number of cuboids for a *n-dimensional data cube, as we have seen is* 2^n
  - For dimensions (*Product, Region, City*), $2^n = 2^3 = 8$ *cuboids*
  - For an *n-dimensional* data cube, the total number of cuboids that can be generated including hierarchies is

$$Total\ number\ of\ cuboids = \prod_{i=1}^{n}(L_i + 1),$$

where L_i is the number of levels associated with dimension i. One is added to L_i in Equation to include the *virtual* top level, all.

Number of Cuboids

- Consider 3 dimensions
- Each dimension has three levels or hierarchy
- How many cuboids are there in a data cube?



Number of Cuboids

Product			*Region*			*Time*		
Class	*Item*	*Product*	*Country*	*State*	*City*	*Year*	*Month*	*Day*
Class 1	Item 1	Camera	India	Karnataka	Mysore	2016	2	3
Class 2	Item 2	DVD	India	Tamilnadu	Salem	2015	4	2
Class 3	Item 3	LED	India	Kerala	Kozhikode	2014	5	1
…	…	…	…	….	…	….	….	…
…	…	…	…	….	…	….	….	…

- The Product dimension has three hierarchies *(class, item, product)*
- The Region dimension has three hierarchies *(country, state, city)*
- The Time dimension has three hierarchies *(year, month, day)*
- Thus, this cu…

$$\prod_{i=1}^{n}(L_i+1) = (3+1) * (3+1) * (3+1) = 64$$

- *Cuboids such as {(product, city, day), (product, city, month), (product,*

Number of Cuboids

- Similarly, If n=10 and each dimension has one level, then

  ??

- If n=10 and each dimension has 4 levels, then

  ??

Number of Cuboids

- Similarly, If n=10 and each dimension has one level, then

$$T = (2)^{10} = 1024$$

- If n=10 and each dimension has 4 levels, then

$$T = (5)^{10} = 9765625$$

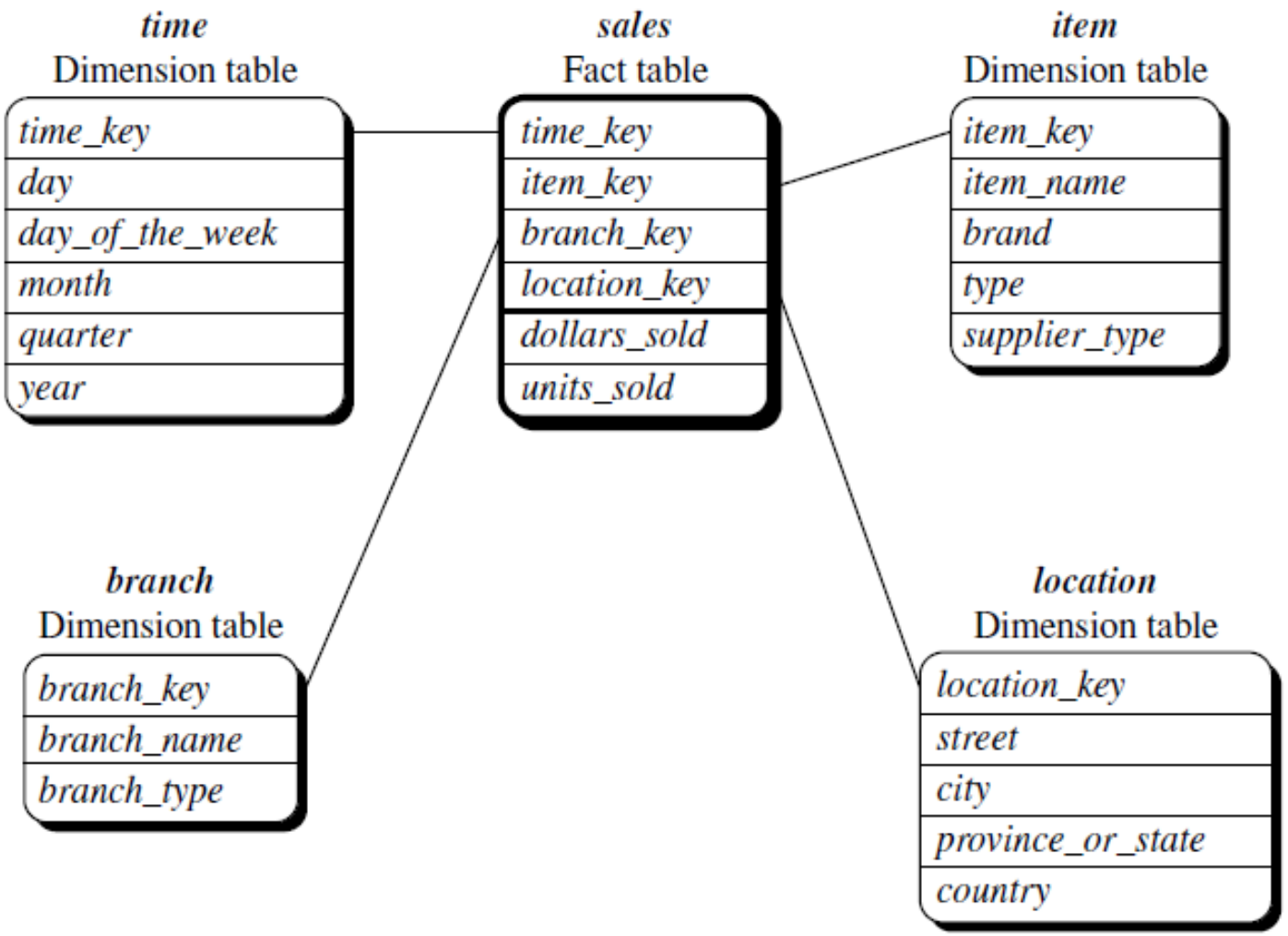
Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Data Models

- **Fact table**
  - A fact table typically includes : fact columns and foreign keys to the dimensions.
  - A fact table contains either detail-level facts or facts that have been aggregated.
- **Hierarchy**:
  - A hierarchy defines the navigating path for rolling up and drilling down. All attributes in a hierarchy belong to the same dimension.
- **Star Schema**:
  - A common form of dimensional model. In a star schema, each dimension is represented by a single dimension table.
- **Snowflake Schema**:
  - In a snowflake schema, different hierarchies in a dimension can be extended into their own dimensional tables. Therefore, a dimension can have more than a single dimension table.

Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Data Models

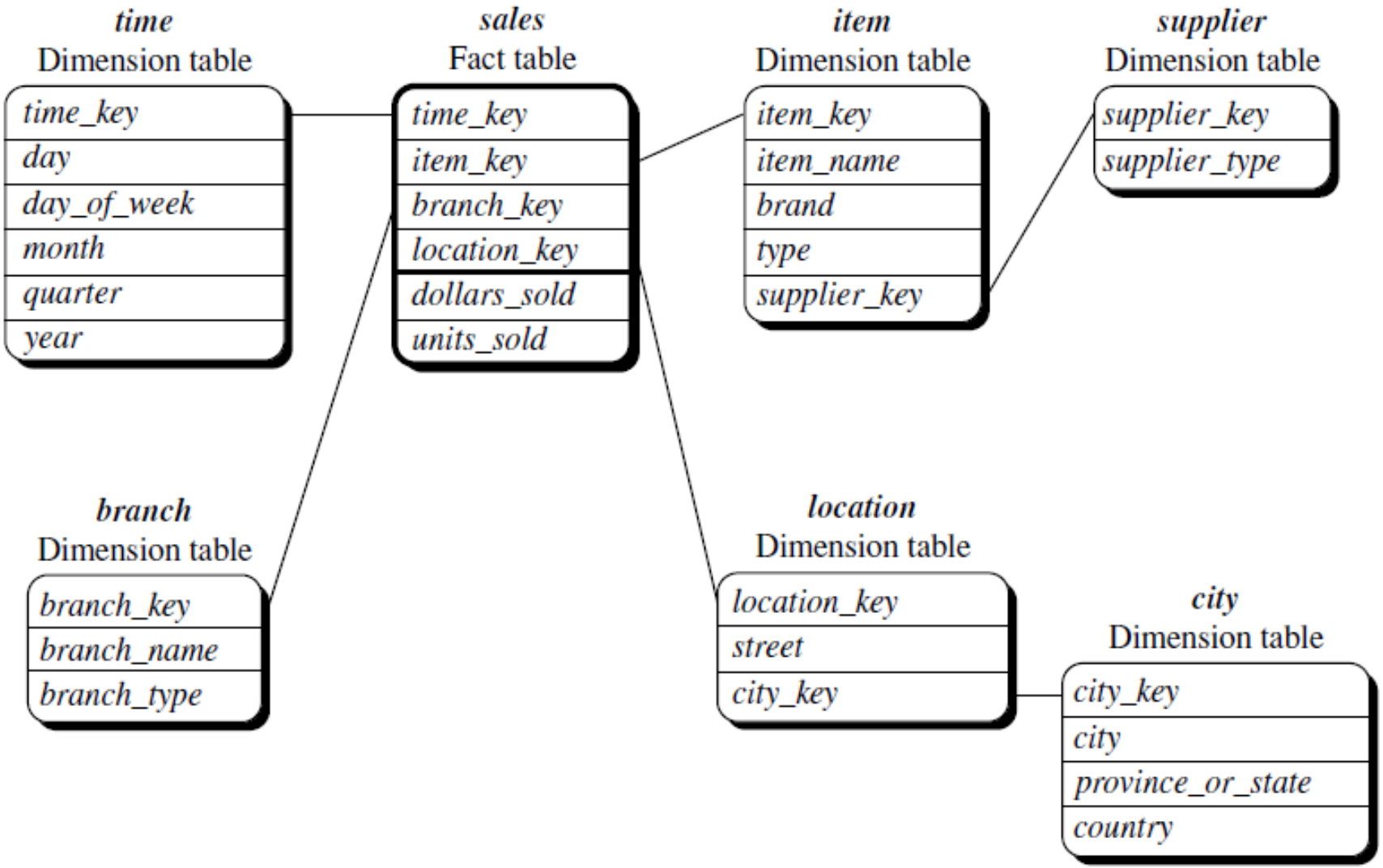
- **Fact constellation:**
  - **Sophisticated applications may require multiple fact tables to *share*** dimension tables.
  - This kind of schema can be viewed as a collection of stars, and hence is called a **galaxy schema or a fact constellation.**

Star schema


time
Dimension table
time_key
day
day_of_the_week
month
quarter
year
sales
Fact table
time_key
item_key
branch_key
location_key
dollars_sold
units_sold
item
Dimension table
item_key
item_name
brand
type
supplier_type
branch
Dimension table
branch_key
branch_name
branch_type
location
Dimension table
location_key
street
city
province_or_state
country

- **Snowflake schema:**
  - **The snowflake schema is a variant of the star schema model,** where some dimension tables are *normalized, thereby further splitting the data into* additional tables.
  - The resulting schema graph forms a shape similar to a snowflake.

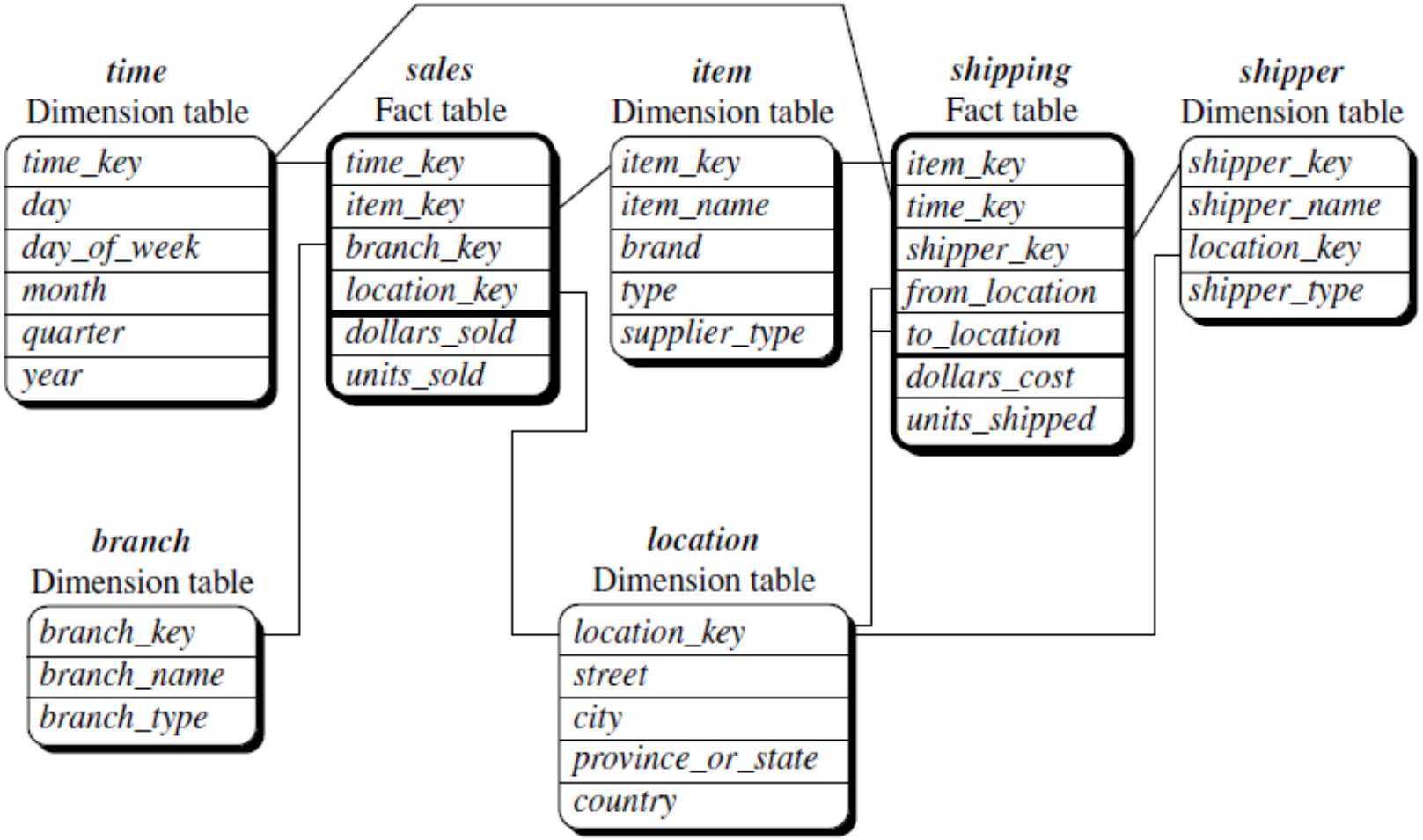
Snowflake schema



Fact constellation

- **Fact constellation:**
  - **Sophisticated applications may require multiple fact tables to *share*** dimension tables.
  - This kind of schema can be viewed as a collection of stars, and hence is called a **galaxy schema or a fact constellation.**

Fact constellation schema of a sales and shipping data warehouse



- This schema specifies two fact tables, *sales and shipping.* The *sales table definition is identical to that of* the star schema. The *shipping table has five dimensions*
- A fact constellation schema allows dimension tables to be shared between fact tables.
- For example, the dimensions tables for *time, item, and location are* shared between the *sales and shipping fact tables*

Data Mining Query Language (DMQL)

- The DMQL was proposed by Han Fu Wang, et al. for the DBMiner data mining system.
- The DMQL is based on the SQL
- Designed to support ad hoc and interactive data mining.
- Provides commands for specifying primitives.
- Particularly we examine how to define Cube, Dimension and Shared Dimensions

Examples for Defining Star, Snowflake,

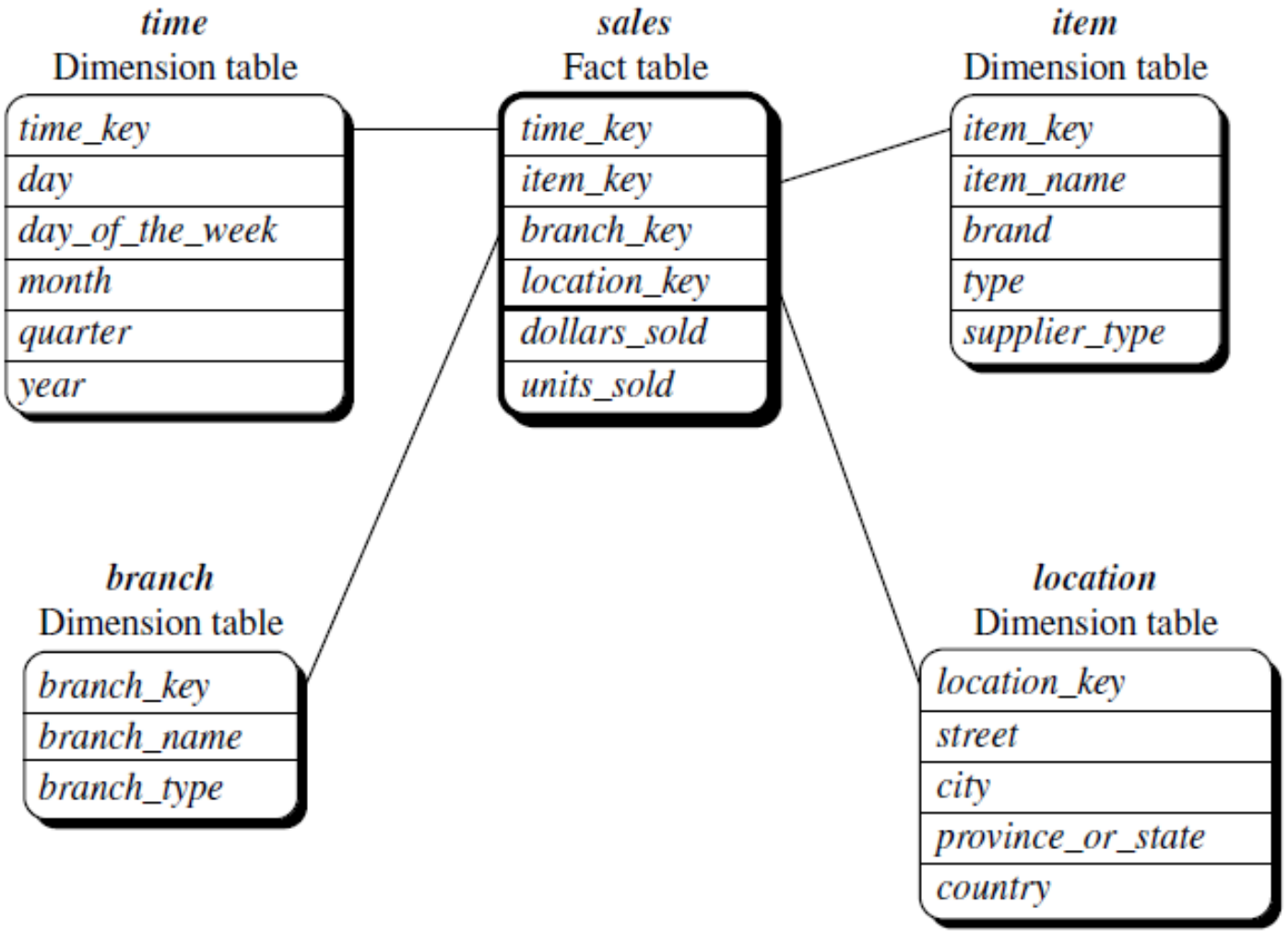
- The *cube definition* statement has the following syntax:

define cube ⟨cube_name⟩ [⟨dimension_list⟩]: ⟨measure_list⟩

- The *dimension definition* statement has the following syntax:

define dimension ⟨dimension_name⟩ as (⟨attribute_or_dimension_list⟩)

The star schema of following Example in DMQL



Defining a Star Schema in DMQL

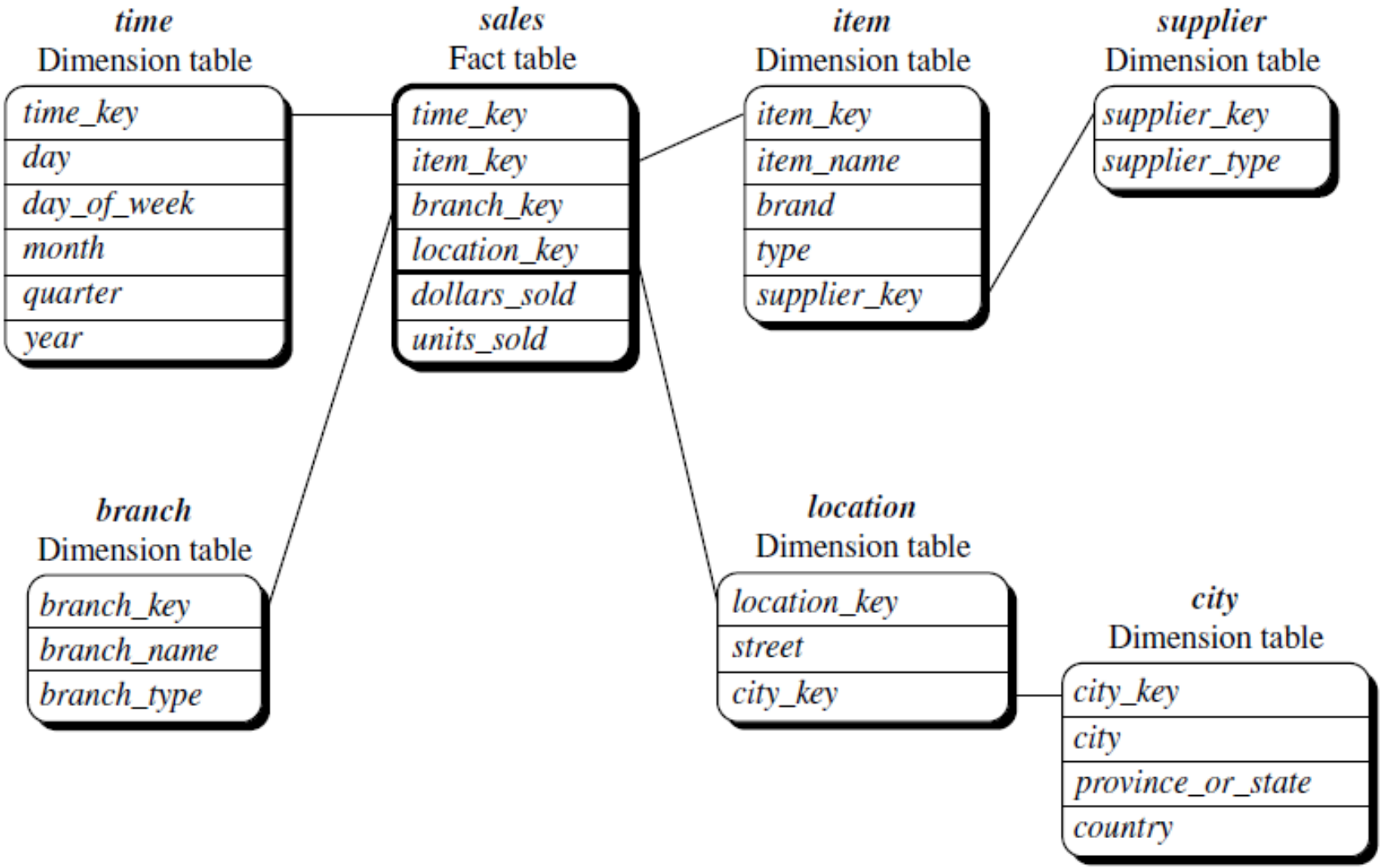
```
define cube sales_star [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month,
    quarter, year)
define dimension item as (item_key, item_name, brand, type,
    supplier_type)
define dimension branch as (branch_key, branch_name,
    branch_type)
define dimension location as (location_key, street, city,
    province_or_state, country)
```

- A statement such a

compute cube sales_star

would explicitly instruct the system to compute the sales aggregate cuboids for all 16 subsets of the set *{time, item, branch, location} including the empty subset*

Snowflake schema



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))
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city(city_key,
    province_or_state, country))
```

OLAP Operations in the Multidimensional Data Model

- In the multidimensional model, data are organized into multiple dimensions,
- and each dimension contains multiple levels of abstraction defined by concept hierarchies.
- This organization provides users with the flexibility to view data from different perspectives
- A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand.

OLAP Operations

- OLAP operations in multidimensional data.
  - Roll – up
  - Drill – down
  - Slice
  - Dice
  - Pivoting (rotate)

OLAP Operations

- Roll – up : Aggregation to higher level
  - Perform aggregation on a data cube by.
  - Climbing up a concept hierarchy for a dimension
  - Example : In a given hierarchy for location dimension

    "street < city < province or state < country."
  - In order to find aggregate of sales in country
    - The roll-up operation aggregates the data by ascending the *location hierarchy from the level of*

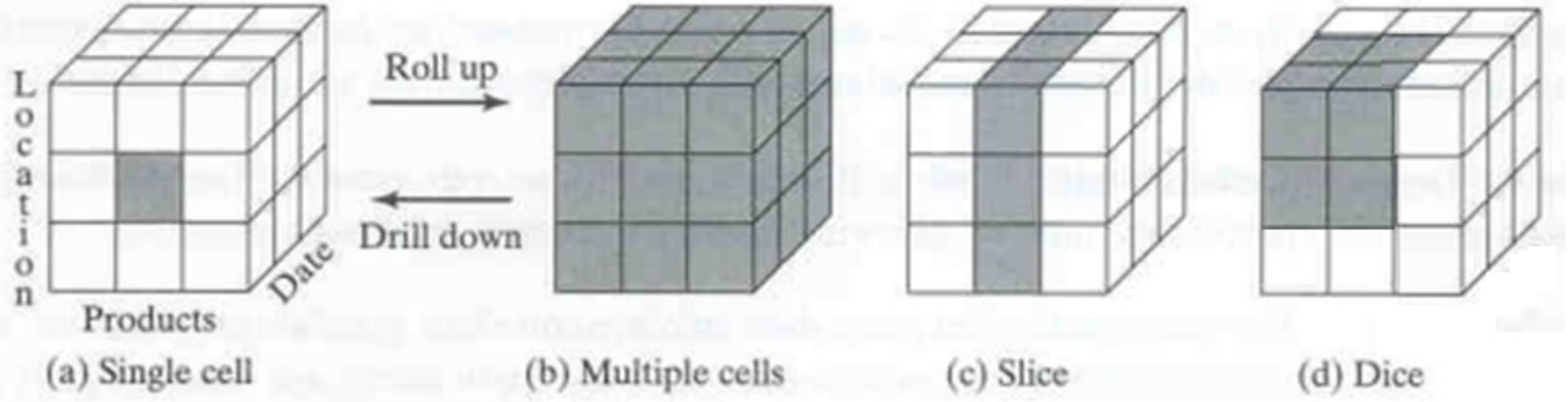
OLAP Operations

- Drill – down : Recalculation with more details
  - Reverse of roll-up
  - Navigates from less detailed data to more detailed data by
    - Stepping down a concept hierarchy for a dimension
    - Introducing additional dimensions

OLAP Operations

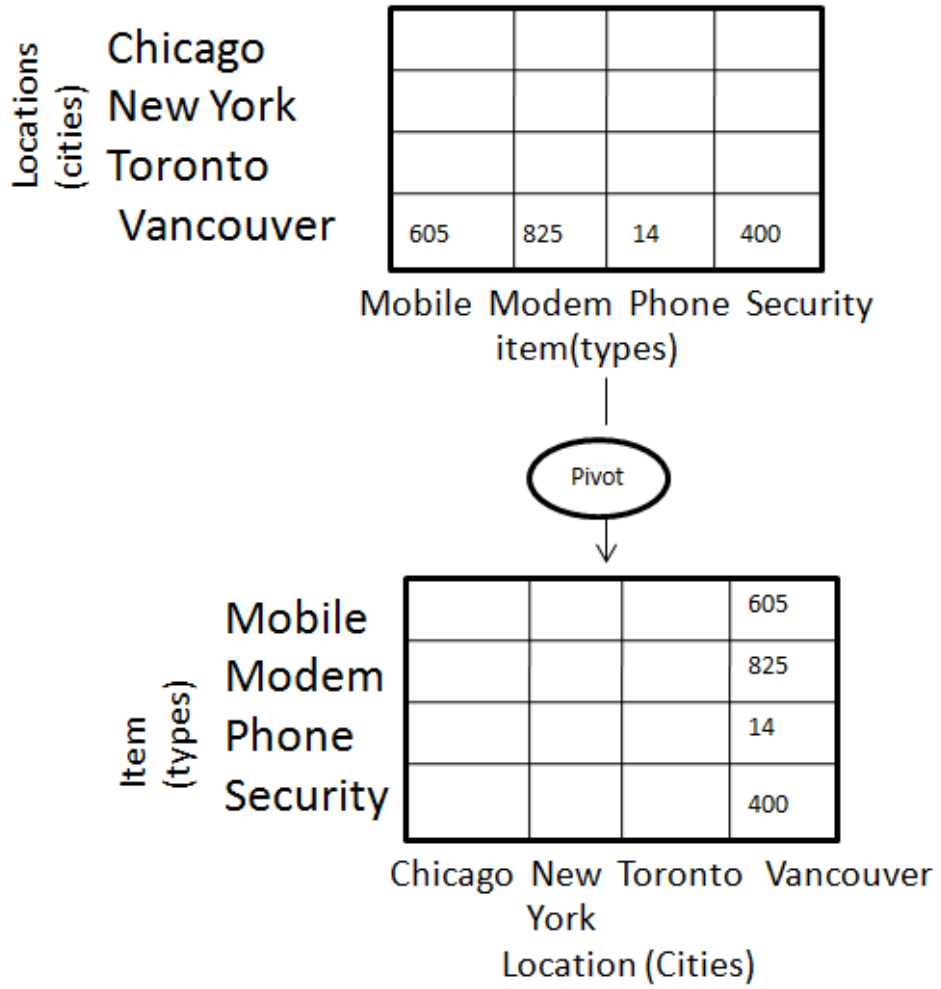
- Slice
  - The slice operation selects one particular dimension from a given cube and provides a new sub-cube
- Dice
  - Defines a sub-cube by performing a selection on two or more dimensions
- Pivoting
  - The pivot operation is also known as rotation.
  - It rotates the data axes in view in order to provide an alternative presentation of data.

OLAP Operations



OLAP Operations

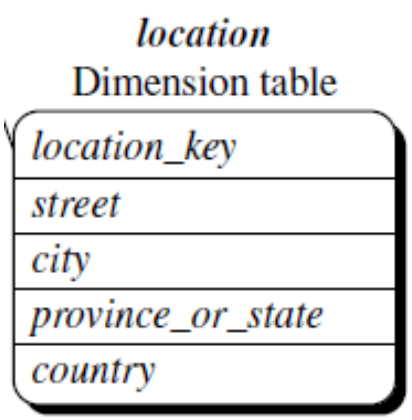
- Pivoting
  - It rotates the data axes in view in order to provide an alternative presentation of data.

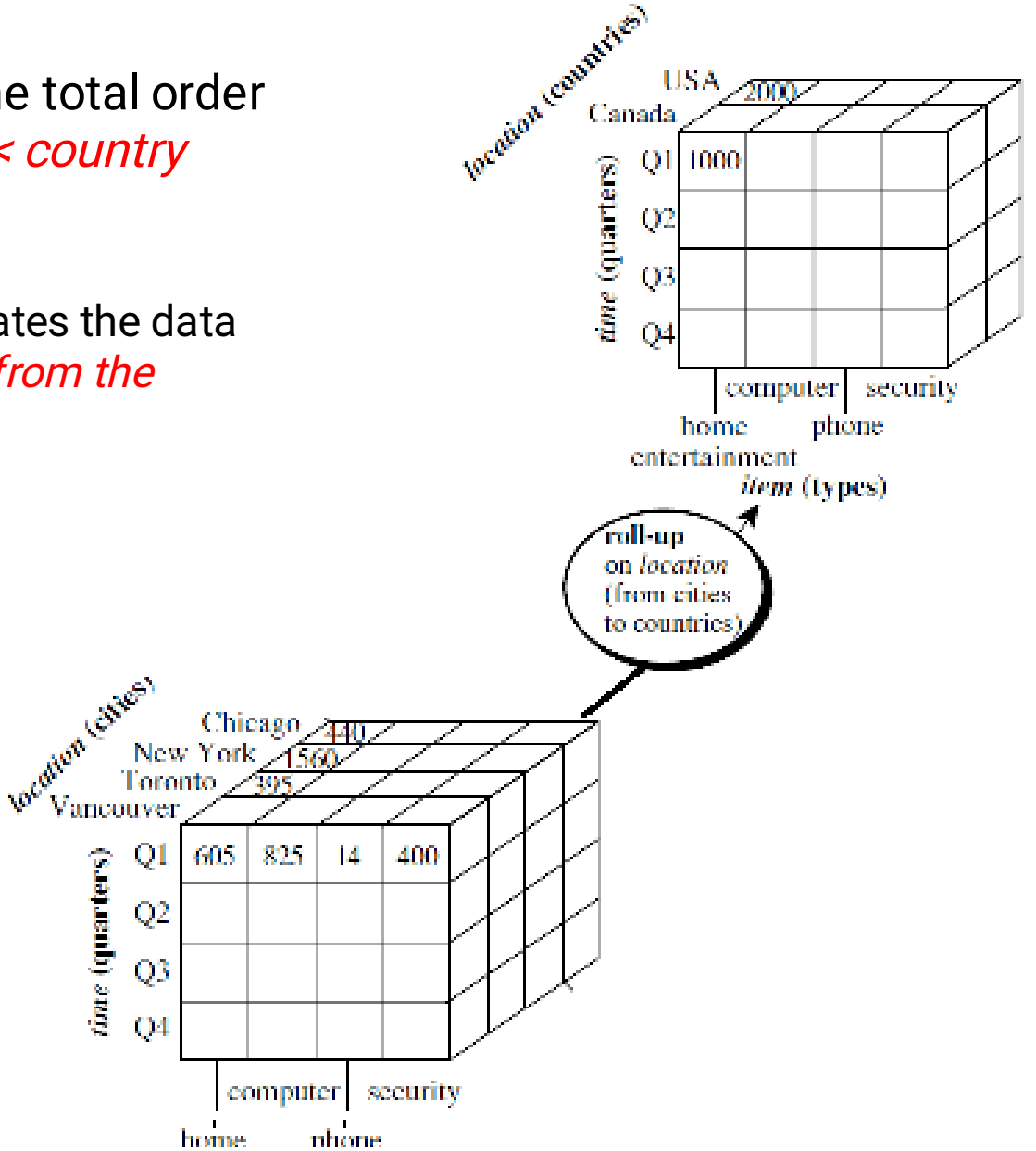


- Roll- up
  - The roll-up operation performs aggregation on a data cube
  - Example: The result of a roll-up Operation performed on the central cube by climbing up the concept hierarchy for *location*

This hierarchy was defined as the total order
street < city < province or state < country

The roll-up operation shown aggregates the data by ascending the *location hierarchy from the level of city to the level of country*

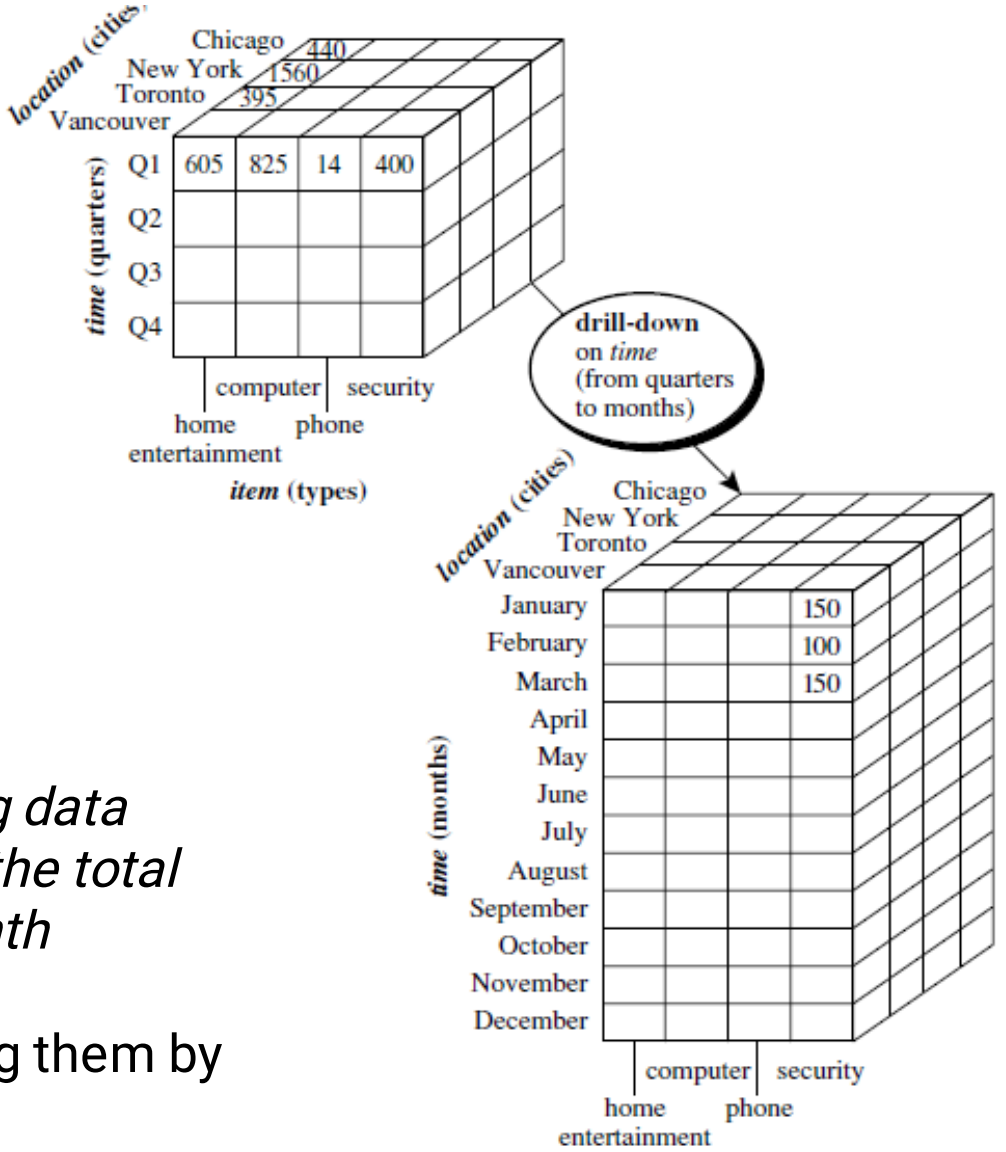




- **Drill-down** is the reverse of roll-up. It navigates from less detailed data to more detailed data.
- Drill-down can be realized by either *stepping down a concept hierarchy for a dimension or introducing additional dimensions*

drill-down operation performed on the central cube by stepping down a concept hierarchy for *time* defined as *"day < month < quarter < year."*

Drill-down occurs by descending the *time hierarchy from the level of quarter to the more detailed* level of *month.*



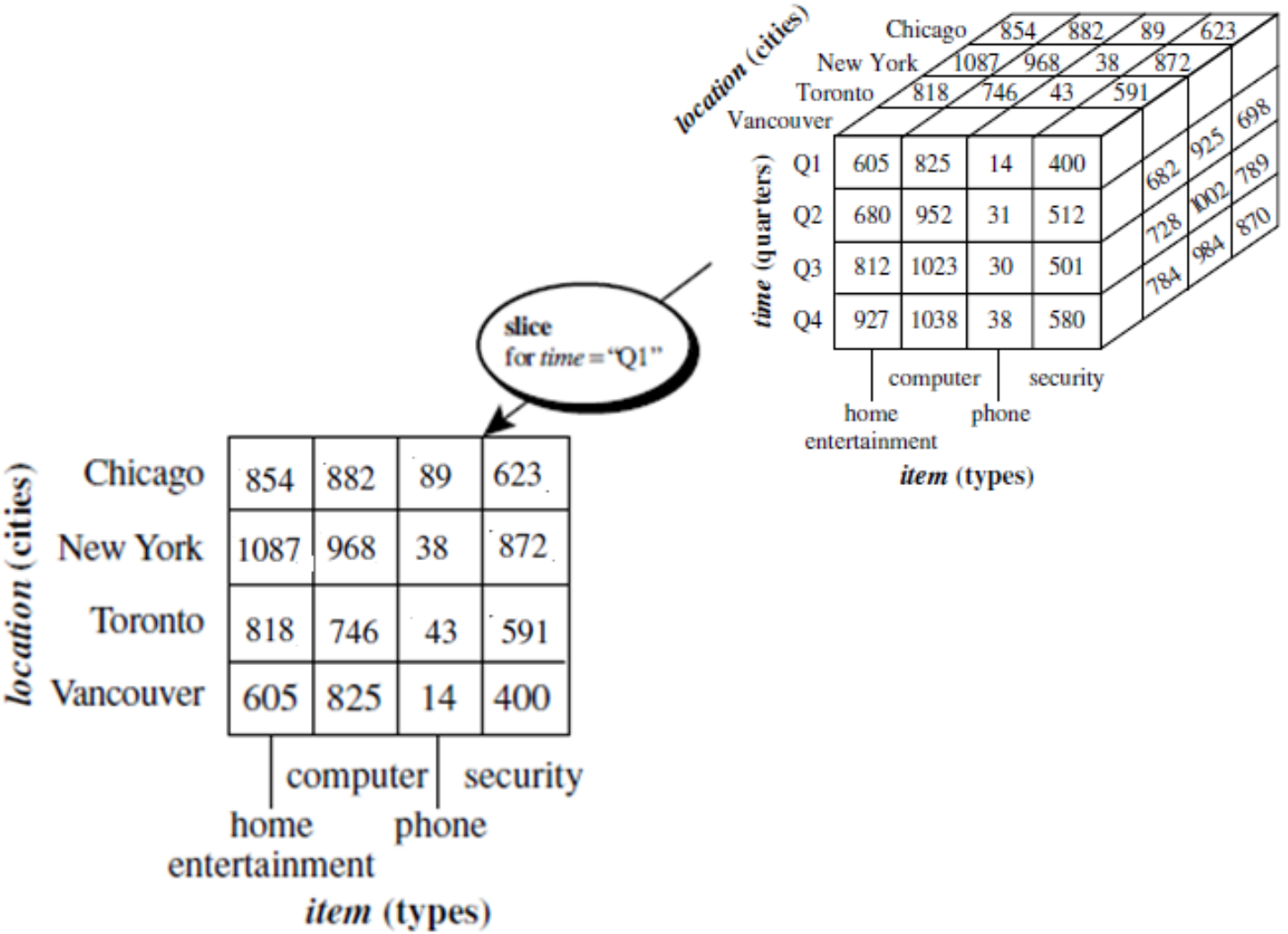
time
Dimension table

time_key
day
day_of_the_week
month
quarter
year

The resulting data cube details the total sales per month rather than summarizing them by quarter

- The ***slice*** *operation performs a selection on one dimension of the given* cube

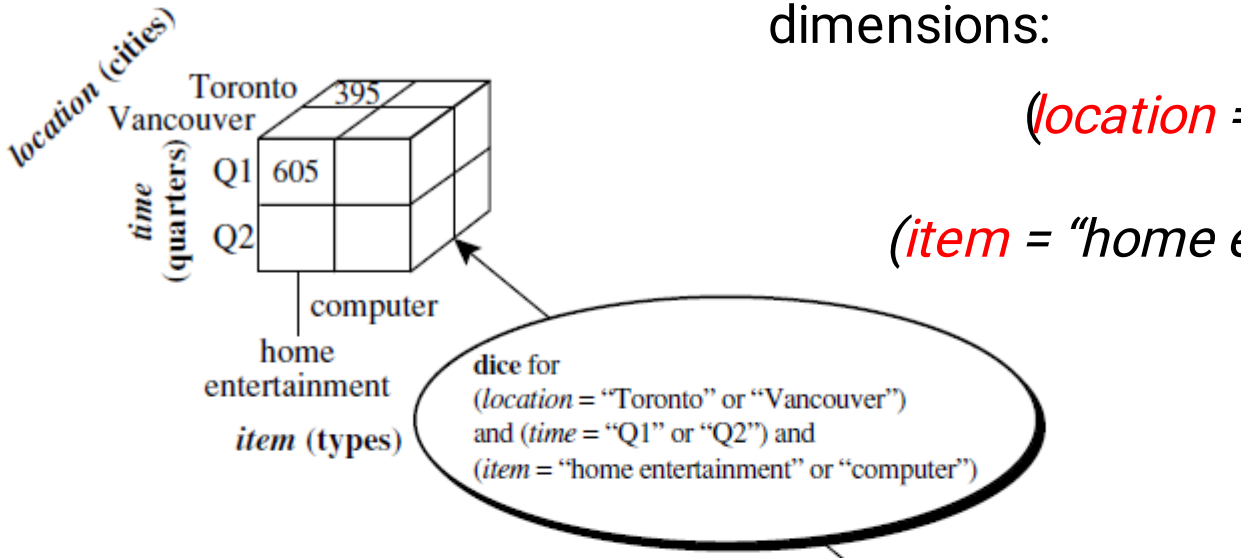
  The sales data are selected from the central cube for the dimension *time using the criterion time="Q1."*

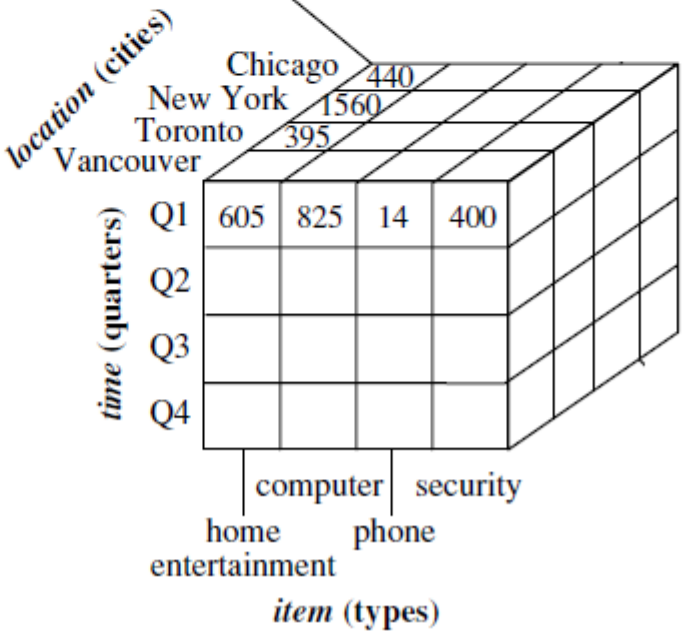


- The *dice operation defines a subcube by performing a selection on two* or more dimensions

Dice operation on the central cube based on the following selection criteria that involve three dimensions:

(*location* = *"Toronto"* or "Vancouver") and
(*time* = *"Q1" or "Q2") and*
(*item* = *"home entertainment" or* "computer").





- *Pivot (also called rotate) is a visualization operation that rotates the data* axes in view to provide an alternative data presentation.

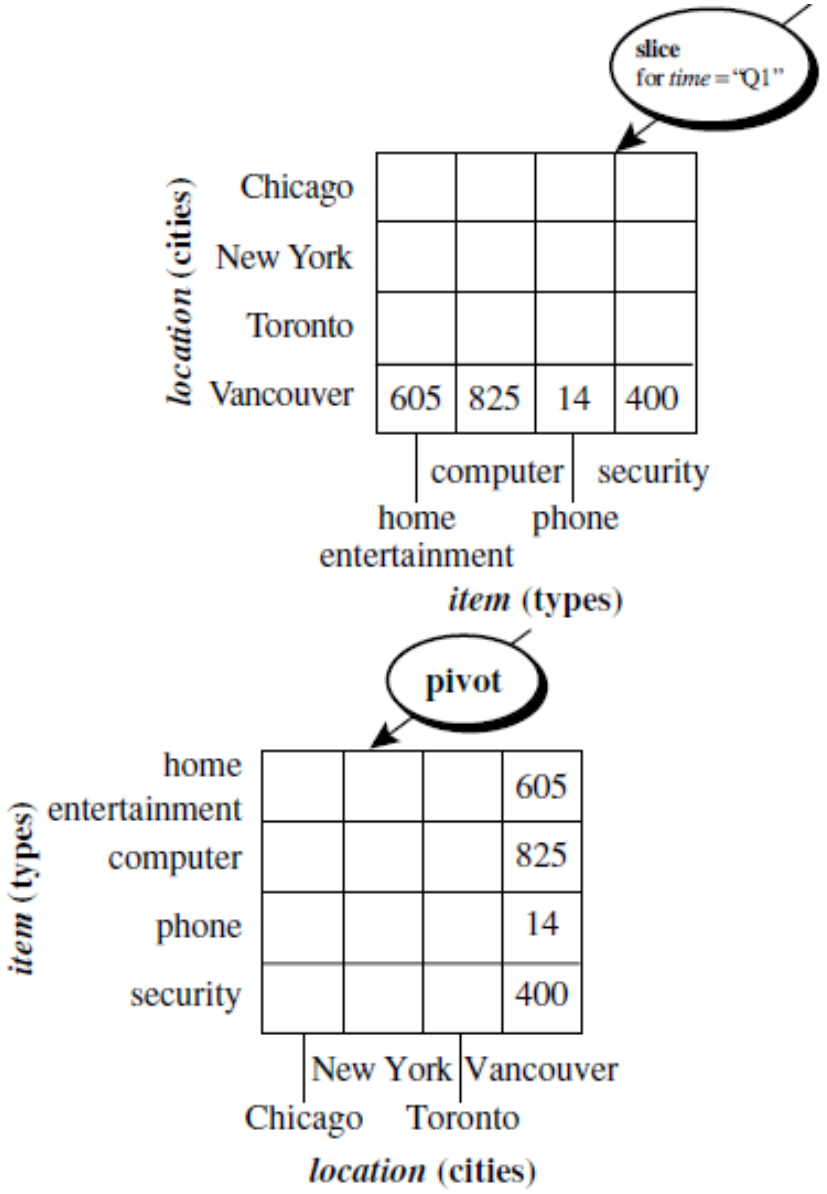
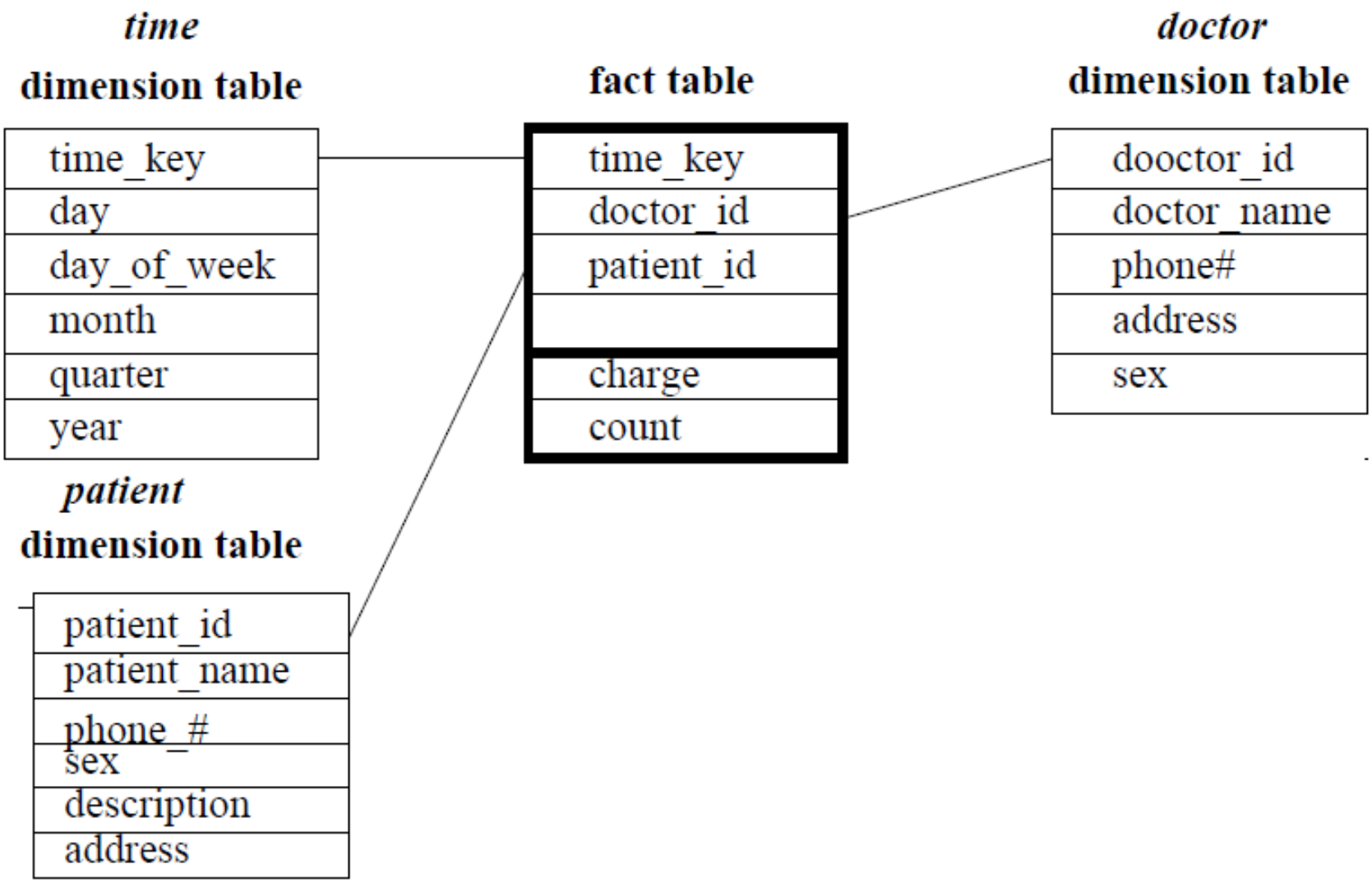


Figure shows a pivot operation where the *item and location axes in a 2-D slice are rotated*

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

1. Draw as tar schema diagram for the above data warehouse
2. Starting with the base cuboid [*day; doctor; patient], what specific OLAP operations should be per* formed in order to list the total fee collected by doctors in 2004?
3. To obtain the same list, write an SQL query assuming the data is stored in a relational database with the schema *fee (day, month, year, doctor, hospital, patient, count, charge).*

a) Solution


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

Starting with the base cuboid [*day; doctor; patient*], *what specific OLAP operations should be per* formed in order to list the total *fee* collected by doctors in 2004 ?

We need to give total fee collected in 2004 from patients (all)

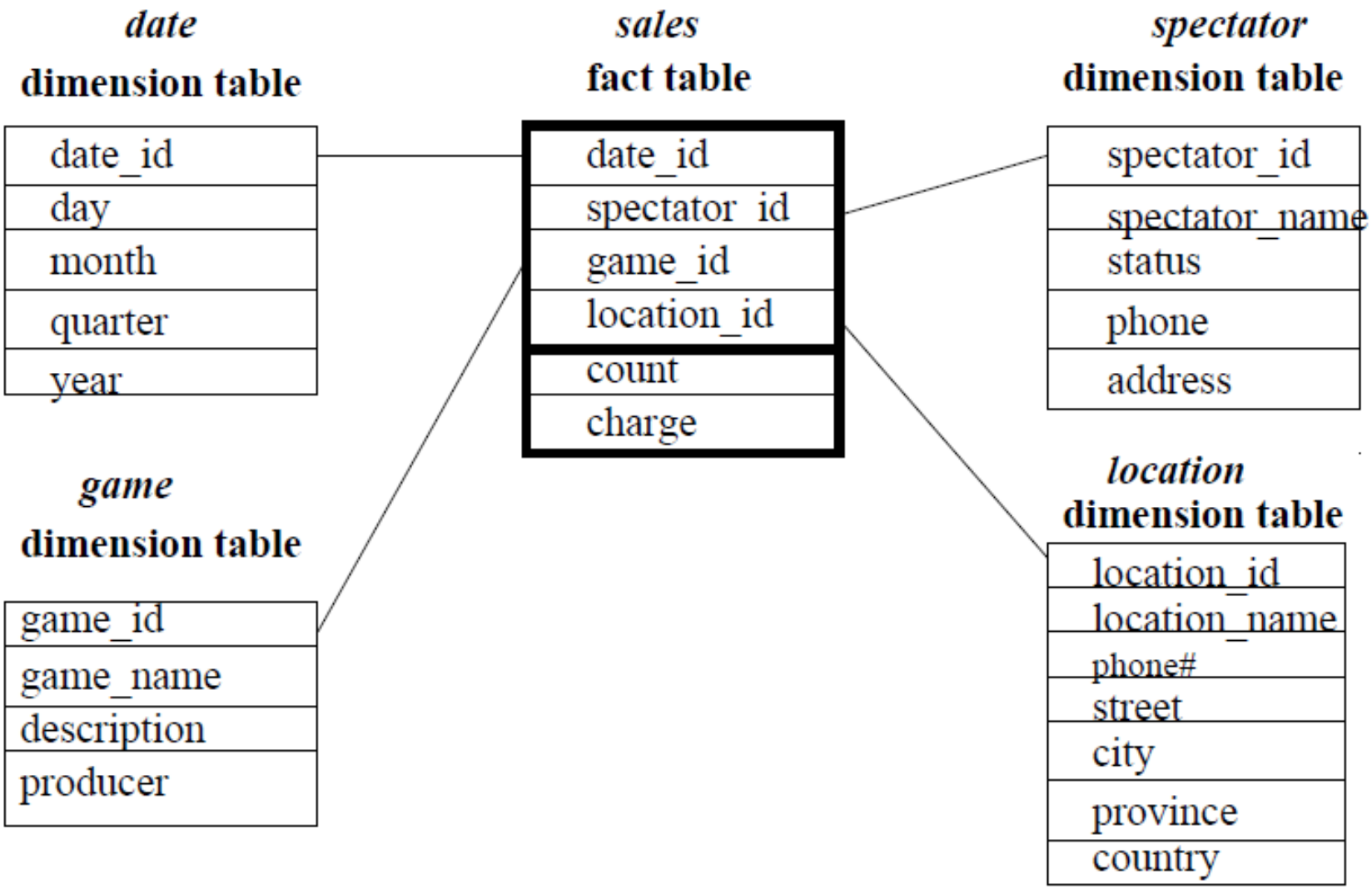
fee : measure ; year : time

- Roll-up *on time from day to year.*
- Slice for *time=2004.*
- Roll-up on *patient from individual patient to all.*

- 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.*

(b) Starting with the base cuboid [*date; spectator; location; game*], *what specific OLAP operations should* one perform in order to list the total charge paid by student spectators at GM Place in 2004?

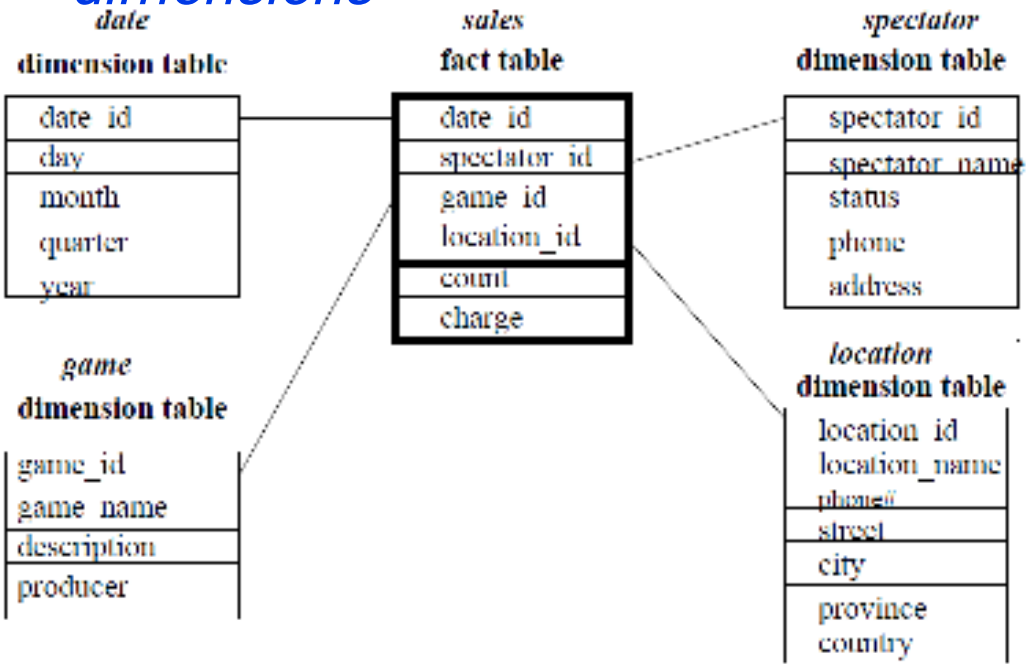

date
dimension table
date_id
day
month
quarter
year
sales
fact table
date_id
spectator_id
game_id
location_id
count
charge
spectator
dimension table
spectator_id
spectator_name
status
phone
address
game
dimension table
game_id
game_name
description
producer
location
dimension table
location_id
location_name
phone#
street
city
province
country

(b)Starting with the base cuboid [*date; spectator; location; game*], *what specific OLAP operations should* one perform in order to list the total *charge* paid by student spectators at GM Place in 2004?

charge – measure

Student - *Spectator*
GM Place - *Location*
2004 – *Date*
For all the games – Game

Spectator, Location, Date, Game are dimensions



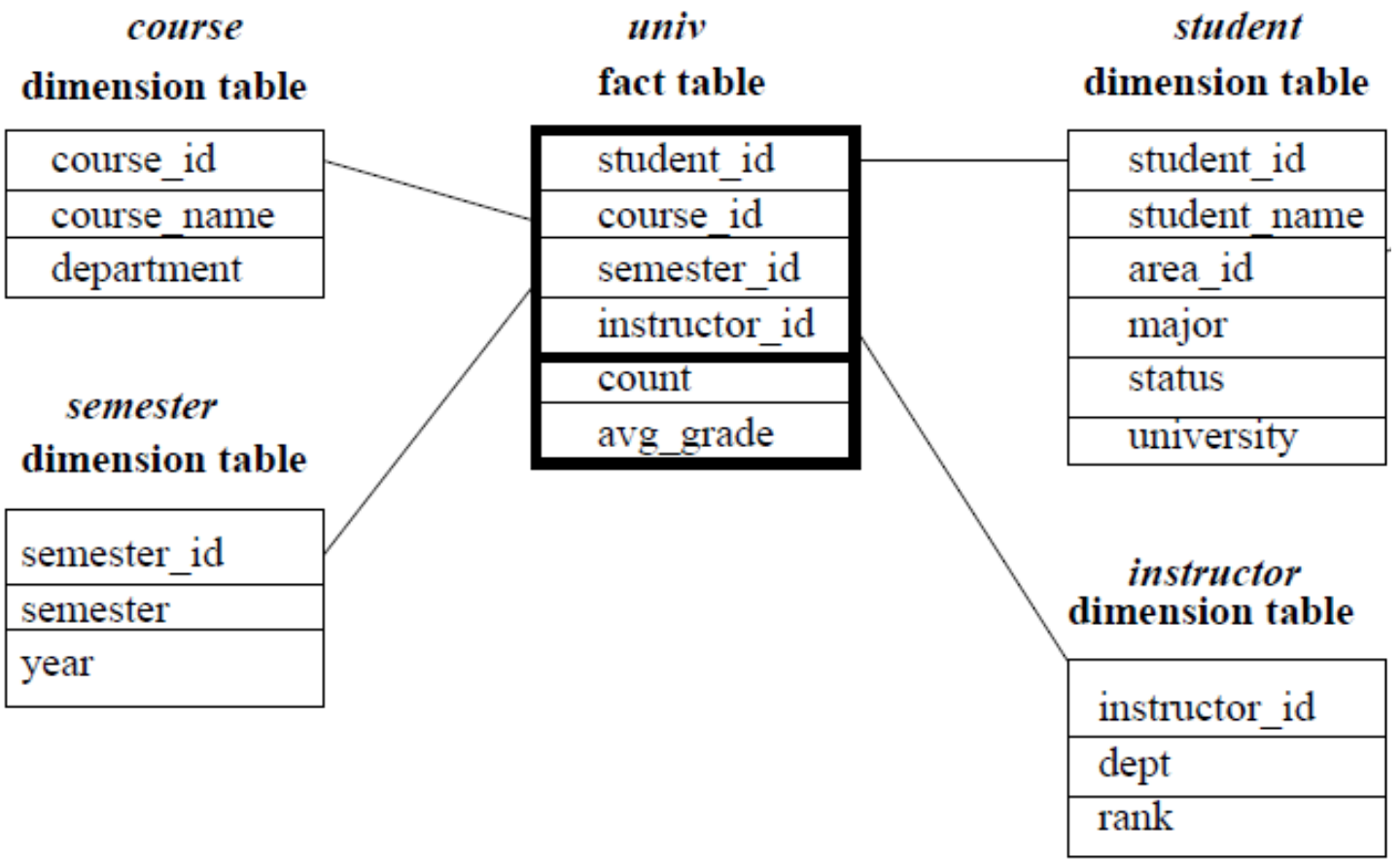
- **Roll-up** on *date from date_id to year.*
- **Roll-up** *on location from location_id to location_name.*
- **Roll-up** *on spectator from spectator id to status.*
- **Dice** *with status="students", location_name="GM Place" and year=2004.*
- **Roll-up** *on game from game_id to all.*

- 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 actual* course grade of the student. At higher conceptual levels, *avg grade stores the average grade for the given* combination.

(a) Draw a *snow-flake 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 courses* for each *Big University student.*

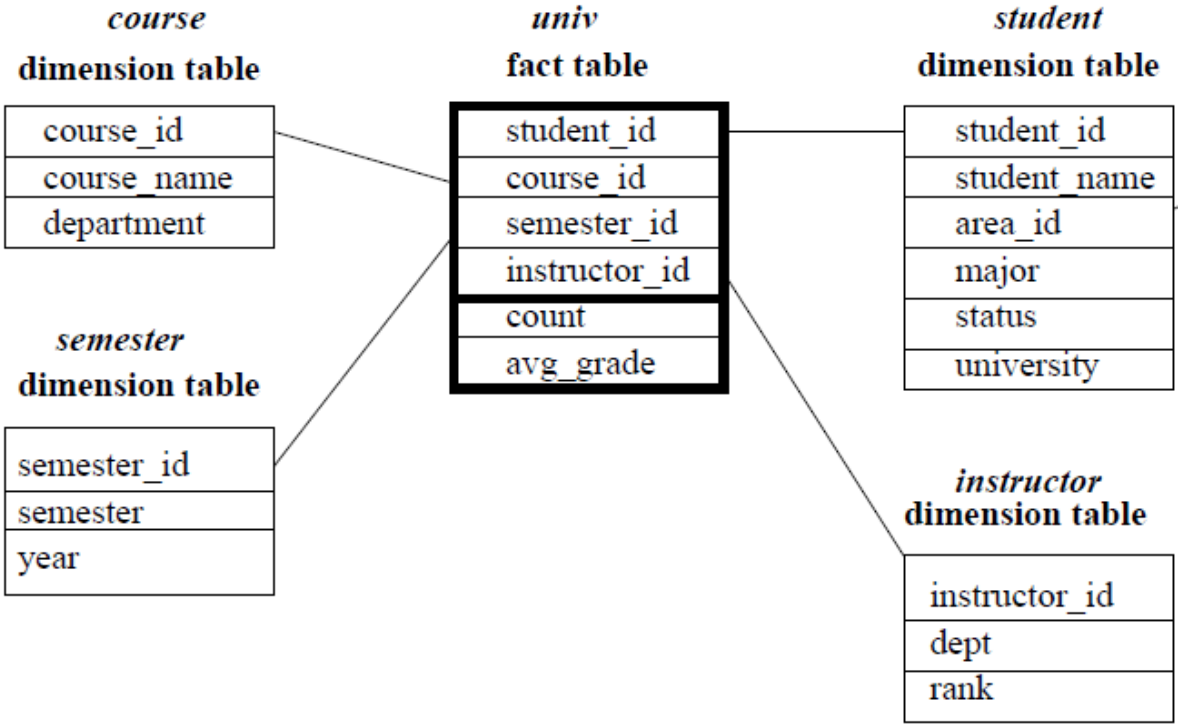
(c) If each dimension has ¯ve levels (including all), such as *student < major < status < university < all",* how many cuboids will this cube contain (including the base and apex cuboids)?


course
dimension table
course_id
course_name
department
semester
dimension table
semester_id
semester
year
univ
fact table
student_id
course_id
semester_id
instructor_id
count
avg_grade
student
dimension table
student_id
student_name
area_id
major
status
university
instructor
dimension table
instructor_id
dept
rank

what specific OLAP operations should one perform in order to list the average grade of CS courses for each *Big University student.*

avg grade is measure

CS - COURSE
BIG University – STUDENT
For each student - STUDENT



what specific OLAP operations should one perform in order to list the average grade of CS courses for each *Big University student.*

- 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 $5^4 = 625$ cuboids.

Star schema:

Sales(keyTime, keyProduct, keyLocation, quantity, customer)

Time(keyTime, day, week, month, quarter, year)

Product(keyProduct, type, brand, category, group)

Location(keyLocation, city, region, country, continent)

SQL query over the star schema:

```
SELECT customer
FROM Sales, Product, Time, Location
WHERE Sales.keyTime=Time.keyTime AND
   Sales.keyProduct=Product.keyProduct AND
   Sales.keyLocation=Location.keyLocation AND
   Time.year="2015"  AND
   Product.category="Car" AND
   Location.country="Italy"
```

Exercise: Data Warehouse design for a wholesale furniture company

Exercise

Wholesale furniture company

Design the data warehouse for a wholesale furniture company. The data warehouse has to allow to analyze the company's situation at least with respect to the Furniture, Customers and Time.
Moreover, the company needs to analyze:

- the furniture with respect to its type (chair, table, wardrobe, cabinet...), category (kitchen, living room, bedroom, bathroom, office...) and material (wood, marble...)
- the customers with respect to their spatial location, by considering at least cities, regions and states

The company is interested in learning at least the quantity, income and discount of its sales.

Questions

1. Identify facts, dimensions and measures
2. For each fact:
   - design the star or snowflake schema and write the following SQL queries:
     - Find the quantity, the total income and discount with respect to each city, type of furniture and the month
     - Find the average quantity, income and discount with respect to each country, furniture material and year
     - Determine the 5 most sold furnitures during the May month

A possible solution

Facts, dimensions, measures, attribute tree, fact schema

FACT Sales

MEASURES Quantity, Income, Discount

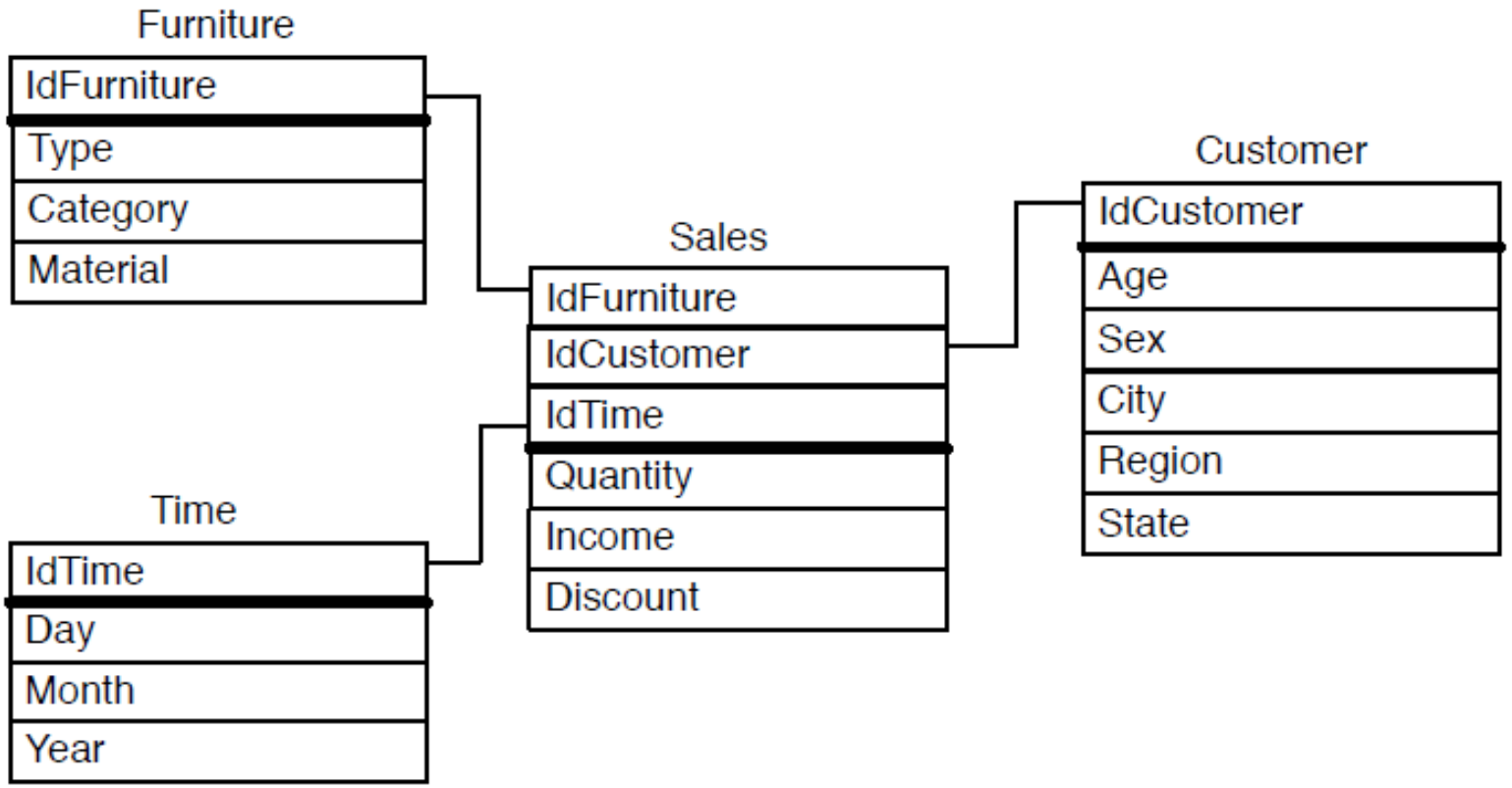
DIMENSIONS Furniture (Type, Category, Material)

Customer (Age, Sex, City $\rightarrow$ Region $\rightarrow$ State)

Time (Day $\rightarrow$ Month $\rightarrow$ Year)

A possible solution

Star schema



- Find the quantity, the total income and discount with respect to each city, type of furniture and the month

```
SELECT C.City, F.Type, T.Month,
       SUM(S.Quantity), SUM(S.Income), SUM(S.Discount)
FROM Sales S, Customer C, Time T, Furniture F
WHERE S.IdCustomer = C.IdCustomer AND
       S.IdTime = T.IdTime AND
       S.IdFurniture = F.IdFurniture
GROUP BY T.Month, F.Type, C.City
```

- Find the average quantity, income and discount with respect to each country, furniture material and year

```
SELECT C.Country, F.Material, T.Year,
           AVG(S.Quantity), AVG(S.Income), AVG(S.Discount)
FROM Sales S, Customer C, Time T, Furniture F
WHERE S.IdCustomer = C.IdCustomer AND
           S.IdTime = T.IdTime AND
           S.IdFurniture = F.IdFurniture
GROUP BY T.Year, C.Country, F.Material
```

- Determine the 5 most sold furnitures during the May month

```
SELECT F.Type, SUM(S.Quantity)
FROM (
   SELECT F.Type, SUM(S.Quantity) AS TotQuantity,
      RANK() OVER (ORDER BY SUM(S.Quantity) DESC)
                              AS Rank
   FROM Sale S, Furniture F, Time T
   WHERE S.IdFurniture = F.IdFurniture AND
            S.IdTime = T.IdTime AND
            T.Month = "May")
WHERE rank <=5
```