component of an information system. Data mining queries and functions are optimized based on mining query analysis, data structures, indexing schemes, and query processing methods of a DB or DW system. With further technology advances, DM, DB, and DW systems will evolve and integrate together as one information system with multiple functionalities. This will provide a uniform information processing environment.

This approach is highly desirable because it facilitates efficient implementations of data mining functions, high system performance, and an integrated information processing environment.

With this analysis, it is easy to see that a data mining system should be coupled with a DB/DW system. Loose coupling, though not efficient, is better than no coupling because it uses both data and system facilities of a DB/DW system. Tight coupling is highly desirable, but its implementation is nontrivial and more research is needed in this area. Semitight coupling is a compromise between loose and tight coupling. It is important to identify commonly used data mining primitives and provide efficient implementations of such primitives in DB or DW systems.

Major Issues in Data Mining

The scope of this book addresses major issues in data mining regarding mining methodology, user interaction, performance, and diverse data types. These issues are introduced below:

Mining methodology and user interaction issues: These reflect the kinds of knowledge mined, the ability to mine knowledge at multiple granularities, the use of domain knowledge, ad hoc mining, and knowledge visualization.

Mining different kinds of knowledge in databases: Because different users can be interested in different kinds of knowledge, data mining should cover a wide spectrum of data analysis and knowledge discovery tasks, including data characterization, discrimination, association and correlation analysis, classification, prediction, clustering, outlier analysis, and evolution analysis (which includes trend and similarity analysis). These tasks may use the same database in different ways and require the development of numerous data mining techniques.

Interactive mining of knowledge at multiple levels of abstraction: Because it is difficult to know exactly what can be discovered within a database, the data mining process should be *interactive*. For databases containing a huge amount of data, appropriate sampling techniques can first be applied to facilitate interactive data exploration. Interactive mining allows users to focus the search for patterns, providing and refining data mining requests based on returned results. Specifically, knowledge should be mined by drilling down, rolling up,

and pivoting through the data space and knowledge space interactively, similar to what OLAP can do on data cubes. In this way, the user can interact with the data mining system to view data and discovered patterns at multiple granularities and from different angles.

- *Incorporation of background knowledge:* Background knowledge, or information regarding the domain under study, may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction. Domain knowledge related to databases, such as integrity constraints and deduction rules, can help focus and speed up a data mining process, or judge the interestingness of discovered patterns.

- *Data mining query languages and ad hoc data mining:* Relational query languages (such as SQL) allow users to pose ad hoc queries for data retrieval. In a similar vein, high-level data mining query languages need to be developed to allow users to describe ad hoc data mining tasks by facilitating the specification of the relevant sets of data for analysis, the domain knowledge, the kinds of knowledge to be mined, and the conditions and constraints to be enforced on the discovered patterns. Such a language should be integrated with a database or data warehouse query language and optimized for efficient and flexible data mining.

- *Presentation and visualization of data mining results:* Discovered knowledge should be expressed in high-level languages, visual representations, or other expressive forms so that the knowledge can be easily understood and directly usable by humans. This is especially crucial if the data mining system is to be interactive. This requires the system to adopt expressive knowledge representation techniques, such as trees, tables, rules, graphs, charts, crosstabs, matrices, or curves.

- *Handling noisy or incomplete data:* The data stored in a database may reflect noise, exceptional cases, or incomplete data objects. When mining data regularities, these objects may confuse the process, causing the knowledge model constructed to overfit the data. As a result, the accuracy of the discovered patterns can be poor. Data cleaning methods and data analysis methods that can handle noise are required, as well as outlier mining methods for the discovery and analysis of exceptional cases.

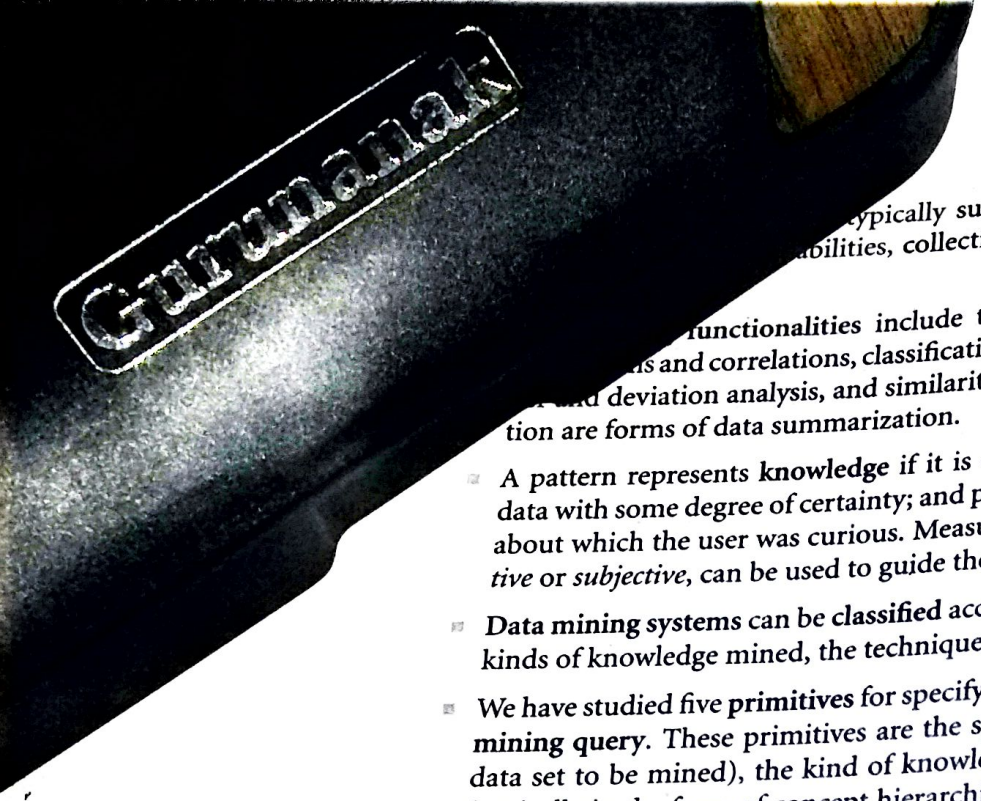
- *Pattern evaluation—the interestingness problem:* A data mining system can uncover thousands of patterns. Many of the patterns discovered may be uninteresting to the given user, either because they represent common knowledge or lack novelty. Several challenges remain regarding the development of techniques to assess the interestingness of discovered patterns, particularly with regard to subjective measures that estimate the value of patterns with respect to a given user class, based on user beliefs or expectations. The use of interestingness measures or user-specified constraints to guide the discovery process and reduce the search space is another active area of research.

Performance issues: These include efficiency, scalability, and parallelization of data mining algorithms.

- *Efficiency and scalability of data mining algorithms:* To effectively extract information from a huge amount of data in databases, data mining algorithms must be efficient and scalable. In other words, the running time of a data mining algorithm must be predictable and acceptable in large databases. From a database perspective on knowledge discovery, efficiency and scalability are key issues in the implementation of data mining systems. Many of the issues discussed above under *mining methodology and user interaction* must also consider efficiency and scalability.
- *Parallel, distributed, and incremental mining algorithms:* The huge size of many databases, the wide distribution of data, and the computational complexity of some data mining methods are factors motivating the development of **parallel and distributed data mining algorithms**. Such algorithms divide the data into partitions, which are processed in parallel. The results from the partitions are then merged. Moreover, the high cost of some data mining processes promotes the need for **incremental** data mining algorithms that incorporate database updates without having to mine the entire data again "from scratch." Such algorithms perform knowledge modification incrementally to amend and strengthen what was previously discovered.

Issues relating to the diversity of database types:

- *Handling of relational and complex types of data:* Because relational databases and data warehouses are widely used, the development of efficient and effective data mining systems for such data is important. However, other databases may contain complex data objects, hypertext and multimedia data, spatial data, temporal data, or transaction data. It is unrealistic to expect one system to mine all kinds of data, given the diversity of data types and different goals of data mining. Specific data mining systems should be constructed for mining specific kinds of data. Therefore, one may expect to have different data mining systems for different kinds of data.
- *Mining information from heterogeneous databases and global information systems:* Local- and wide-area computer networks (such as the Internet) connect many sources of data, forming huge, distributed, and heterogeneous databases. The discovery of knowledge from different sources of structured, semistructured, or unstructured data with diverse data semantics poses great challenges to data mining. Data mining may help disclose high-level data regularities in multiple heterogeneous databases that are unlikely to be discovered by simple query systems and may improve information exchange and interoperability in heterogeneous databases. Web mining, which uncovers interesting knowledge about Web contents, Web structures, Web usage, and Web dynamics, becomes a very challenging and fast-evolving field in data mining.



ypically summarized. Data warehouse systems provide
bilities, collectively referred to as **OLAP (on-line analytical**

functionalities include the discovery of concept/class descriptions,
s and correlations, classification, prediction, clustering, trend analysis, out-
deviation analysis, and similarity analysis. Characterization and discrimina-
tion are forms of data summarization.

- A pattern represents **knowledge** if it is easily understood by humans; valid on test data with some degree of certainty; and potentially useful, novel, or validates a hunch about which the user was curious. Measures of **pattern interestingness**, either *objective* or *subjective*, can be used to guide the discovery process.
- **Data mining systems** can be **classified** according to the kinds of databases mined, the kinds of knowledge mined, the techniques used, or the applications adapted.
- We have studied five **primitives** for specifying a data mining task in the form of a **data mining query**. These primitives are the specification of task-relevant data (i.e., the data set to be mined), the kind of knowledge to be mined, background knowledge (typically in the form of concept hierarchies), interestingness measures, and knowledge presentation and visualization techniques to be used for displaying the discovered patterns.
- **Data mining query languages** can be designed to support ad hoc and interactive data mining. A data mining query language, such as DMQL, should provide commands for specifying each of the data mining primitives. Such query languages are SQL-based and may eventually form a standard on which graphical user interfaces for data mining can be based.
- Efficient and effective data mining in large databases poses numerous requirements and great challenges to researchers and developers. The issues involved include data mining methodology, user interaction, performance and scalability, and the processing of a large variety of data types. Other issues include the exploration of data mining applications and their social impacts.

Exercises

1.1 What is *data mining*? In your answer, address the following:

(a) Is it another hype?

(b) Is it a simple transformation of technology developed from databases, statistics, and machine learning?

(c) Explain how the evolution of database technology led to data mining.

(d) Describe the steps involved in data mining when viewed as a process of knowledge discovery.

1.2 Present an example where data mining is crucial to the success of a business. What *data mining functions* does this business need? Can they be performed alternatively by data query processing or simple statistical analysis?

1.3 Suppose your task as a software engineer at *Big University* is to design a data mining system to examine the university course database, which contains the following information: the name, address, and status (e.g., undergraduate or graduate) of each student, the courses taken, and the cumulative grade point average (GPA). Describe the *architecture* you would choose. What is the purpose of each component of this architecture?

1.4 How is a *data warehouse* different from a database? How are they similar?

1.5 Briefly describe the following *advanced database systems* and applications: object-relational databases, spatial databases, text databases, multimedia databases, stream data, the World Wide Web.

1.6 Define each of the following *data mining functionalities*: characterization, discrimination, association and correlation analysis, classification, prediction, clustering, and evolution analysis. Give examples of each data mining functionality, using a real-life database with which you are familiar.

1.7 What is the difference between discrimination and classification? Between characterization and clustering? Between classification and prediction? For each of these pairs of tasks, how are they similar?

1.8 Based on your observation, describe another possible kind of knowledge that needs to be discovered by data mining methods but has not been listed in this chapter. Does it require a mining methodology that is quite different from those outlined in this chapter?

1.9 List and describe the five *primitives* for specifying a data mining task.

1.10 Describe why *concept hierarchies* are useful in data mining.

1.11 *Outliers* are often discarded as noise. However, one person's garbage could be another's treasure. For example, exceptions in credit card transactions can help us detect the fraudulent use of credit cards. Taking fraudulence detection as an example, propose two methods that can be used to detect outliers and discuss which one is more reliable.

1.12 Recent applications pay special attention to spatiotemporal data streams. A *spatiotemporal data stream* contains spatial information that changes over time, and is in the form of stream data (i.e., the data flow in and out like possibly infinite streams).

(a) Present three application examples of spatiotemporal data streams.

(b) Discuss what kind of interesting knowledge can be mined from such data streams, with limited time and resources.

(c) Identify and discuss the major challenges in spatiotemporal data mining.

(d) Using one application example, sketch a method to mine one kind of knowledge from such stream data efficiently.

1.13 Describe the differences between the following approaches for the integration of a data mining system with a database or data warehouse system: *no coupling, loose coupling,*

semitight coupling, and *tight coupling*. State which approach you think is the most popular, and why.

1.14 Describe three challenges to data mining regarding *data mining methodology* and *user interaction issues*.

1.15 What are the major challenges of mining a huge amount of data (such as billions of tuples) in comparison with mining a small amount of data (such as a few hundred tuple data set)?

1.16 Outline the major research challenges of data mining in one specific application domain, such as stream/sensor data analysis, spatiotemporal data analysis, or bioinformatics.

Bibliographic Notes

The book *Knowledge Discovery in Databases*, edited by Piatetsky-Shapiro and Frawley [PSF91], is an early collection of research papers on knowledge discovery from data. The book *Advances in Knowledge Discovery and Data Mining*, edited by Fayyad, Piatetsky-Shapiro, Smyth, and Uthurusamy [FPSSe96], is a collection of later research results on knowledge discovery and data mining. There have been many data mining books published in recent years, including *Predictive Data Mining* by Weiss and Indurkhya [WI98], *Data Mining Solutions: Methods and Tools for Solving Real-World Problems* by Westphal and Blaxton [WB98], *Mastering Data Mining: The Art and Science of Customer Relationship Management* by Berry and Linoff [BL99], *Building Data Mining Applications for CRM* by Berson, Smith, and Thearling [BST99], *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations* by Witten and Frank [WF05], *Principles of Data Mining (Adaptive Computation and Machine Learning)* by Hand, Mannila, and Smyth [HMS01], *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman [HTF01], *Data Mining: Introductory and Advanced Topics* by Dunham [Dun03], *Data Mining: Multimedia, Soft Computing, and Bioinformatics* by Mitra and Acharya [MA03], *Introduction to Data Mining* by Tan, Steinbach and Kumar [TSK05]. There

Section-3 Data Warehouse

Data Warehouse

It is a subject oriented, integrated, time variant and non-volatile collection of data in support of management's decision making process.

Subject oriented → It means, rather than focusing on day to day operations, it focuses on the modeling and analysis of data for decision makers.

time variant → It provides information from a historical perspective. (The past 5-10 years)

integrated → since data are collected from multiple sources and stored, it needs data cleaning, data integration to maintain consistency.

Non-volatile → It is physically separated store of data that doesn't require transaction processing, recovery, concurrency control mechanism. It usually requires only two operations

(1) Initial loading of data

(2) Access of data.

DW is an architecture constructed by integrating data from multiple heterogeneous sources to support queries, analytical processing and decision making.

Differences between Database and Datawarehouse

OLTP (online transaction processing)	OLAP (online analytical processing)
(1) These are the online operational processing database systems.	(1) online informational data warehouse systems
(2) Major tasks - online transaction and query processing. (Day to day operation)	(2) Data analysis and decision making
(3) This is customer oriented	(3) Market oriented
(4) It manages current data that are too detailed	(4) Historical data. (time variant)
(5) It usually uses an E-R data model, an application oriented database design.	(5) uses either a star or snowflake model and subject oriented database design.
(6) Database size 100 MB - GB	(6) 100 GB - TB
(7) User → DBA, clerk, database professionals	(7) Knowledge worker (managers, executive, Analyst)
(8) Focuses on data in.	(8) Focuses on information out.

Table 3.3 A 3-D view of sales data for *AllElectronics*, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

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

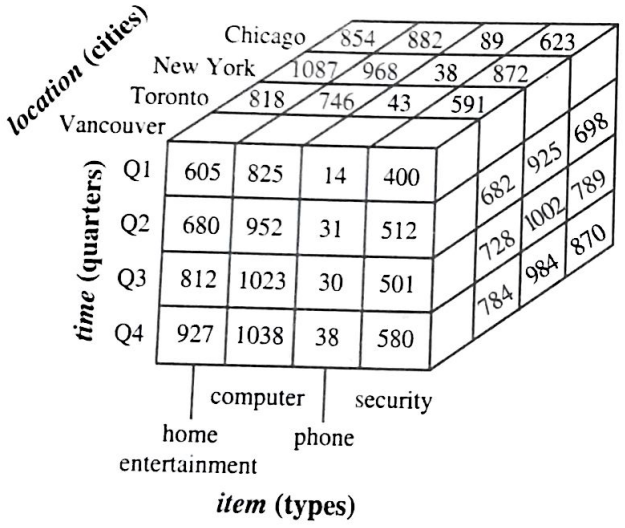


Figure 3.1 A 3-D data cube representation of the data in Table 3.3, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

Suppose that we would now like to view our sales data with an additional fourth dimension, such as *supplier*. Viewing things in 4-D becomes tricky. However, we can think of a 4-D cube as being a series of 3-D cubes, as shown in Figure 3.2. If we continue in this way, we may display any n-D data as a series of $(n-1)$-D "cubes." The data cube is a 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.

The above tables show the data at different degrees of summarization. In the data warehousing research literature, a data cube such as each of the above is often referred to

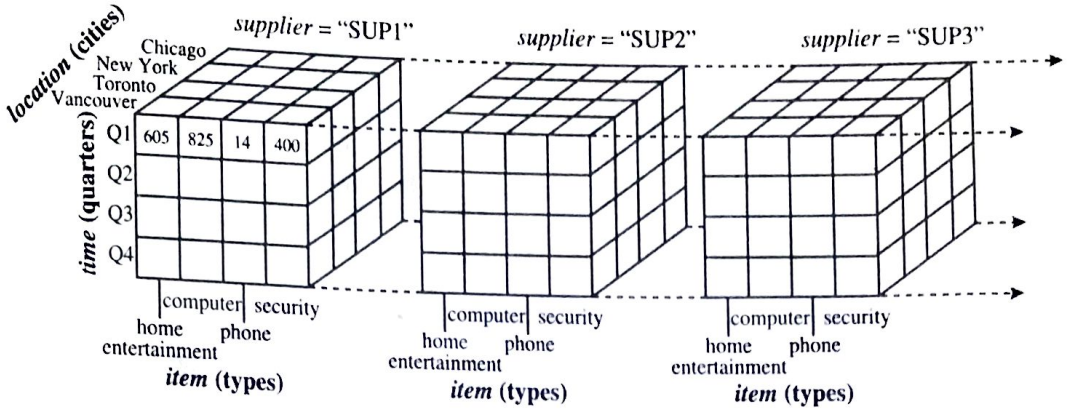


Figure 3.2 A 4-D data cube representation of sales data, according to the dimensions *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars_sold* (in thousands). For improved readability, only some of the cube values are shown.

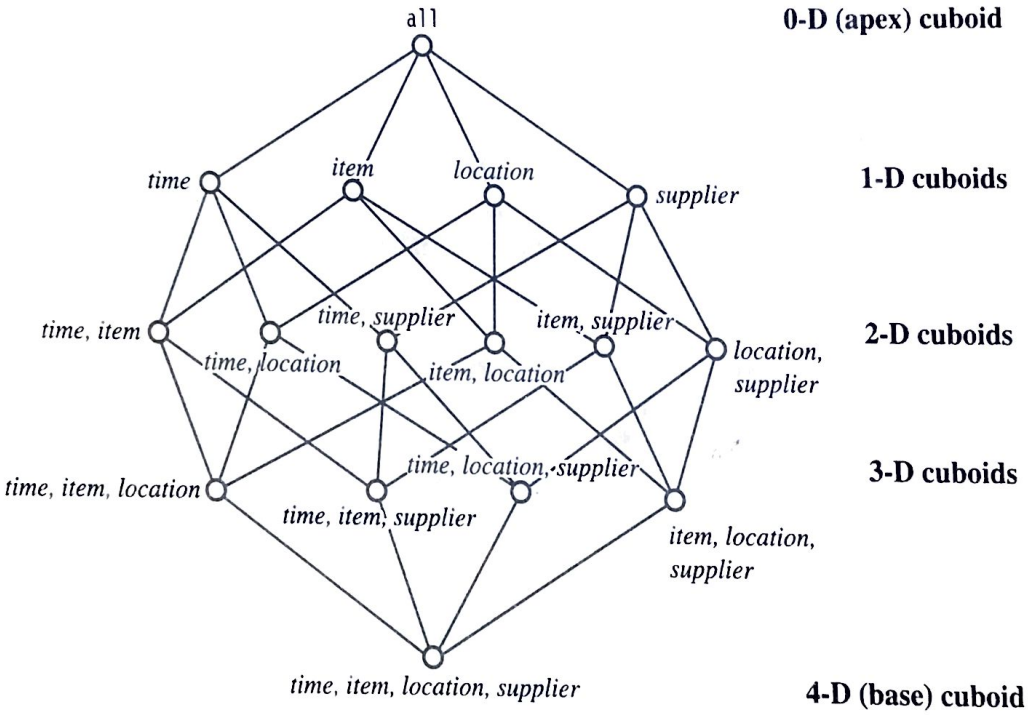


Figure 3.3 Lattice of cuboids, making up a 4-D data cube for the dimensions *time*, *item*, *location*, and *supplier*. Each cuboid represents a different degree of summarization.

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 cuboids is then referred to as a data cube. Figure 3.3 shows a lattice of cuboids forming a data cube for the dimensions *time*, *item*, *location*, and *supplier*.

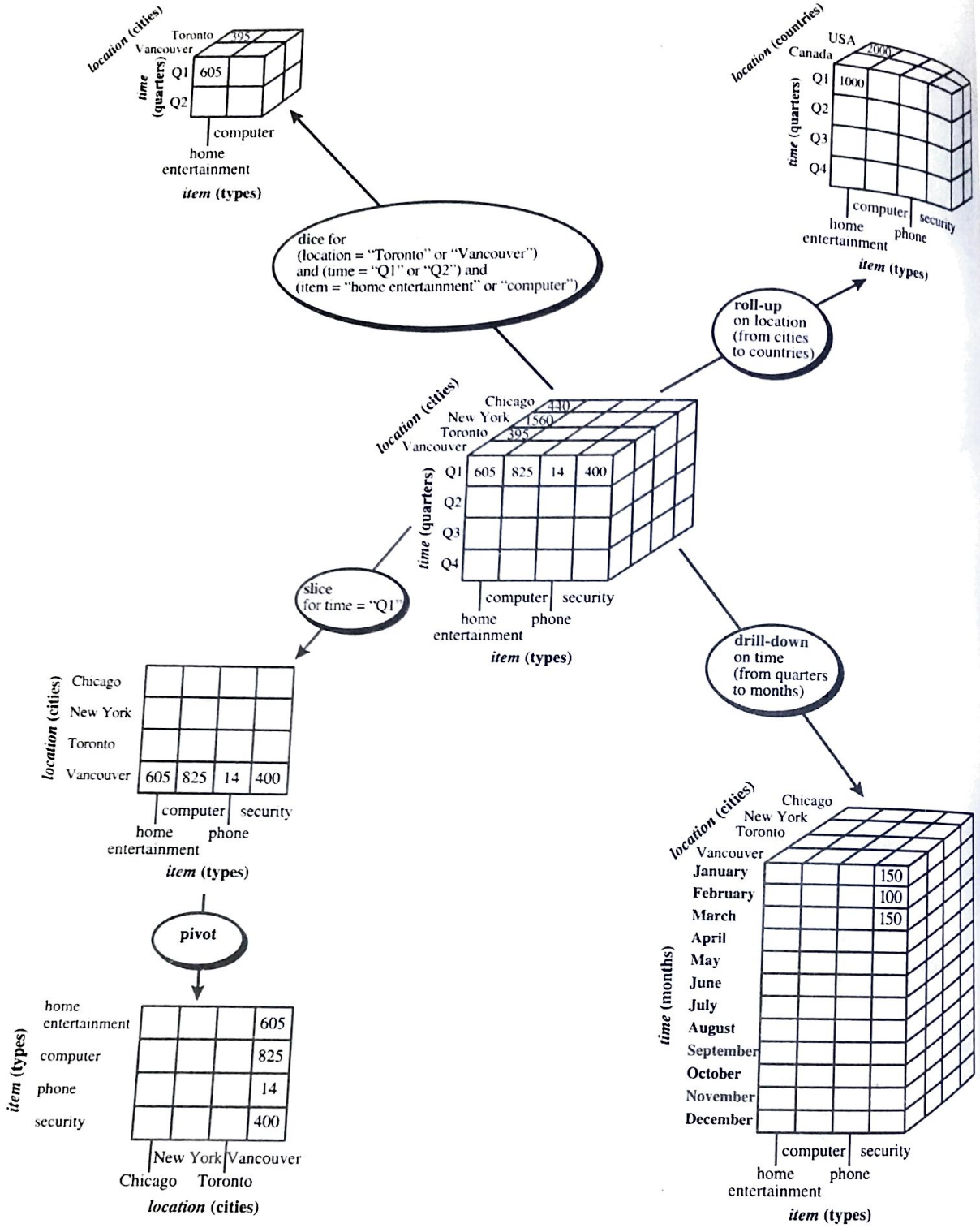


Figure 3.10 Examples of typical OLAP operations on multidimensional data.

of the data in the

OLAP operations on multidimensional data

(1) Rollup → The roll up operation performs aggregation on a data cube, either by climbing up on a dimension or dimension reduction.

(2) Drill-down :- navigates from less detailed data to more detailed data. It can be realized by stepping down or by introducing additional dimensions.

(3) Slice and dice

The slice operation performs a selection on one dimension of the given cube, resulting in a subcube.

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

(4) Pivot (rotate) :- rotates the data axes in view in order to provide an alternative presentation of the data.

The most popular multidimensional models are

(1) star schema.

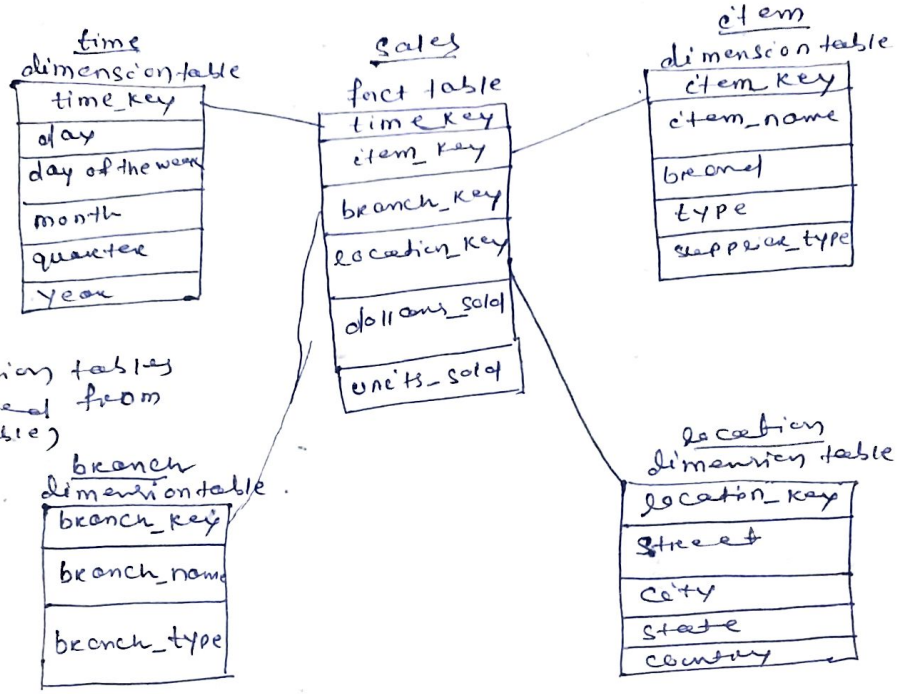
(2) snowflake "

(3) Fact constellation schema. or Galaxy schema

Star schema

Here the data warehouse contains a fact table with a bulk of data with no redundancy in the the centre and a set of smaller dimension tables one for each dimension.

Ex



(All dimension tables are derived from a fact table)

(Star schema of a data warehouse for sales)

Snowflake schema

It is a modification to star schema where some dimensional tables are normalized into a set of smaller dimension tables forming a shape similar to a snowflake

Q suppose that a data warehouse consists of the four dimensions time, item, branch, location and two measures dollars_sold, units_sold. Draw a star schema for the DW.

The most popular multidimensional models are

(1) Star schema.

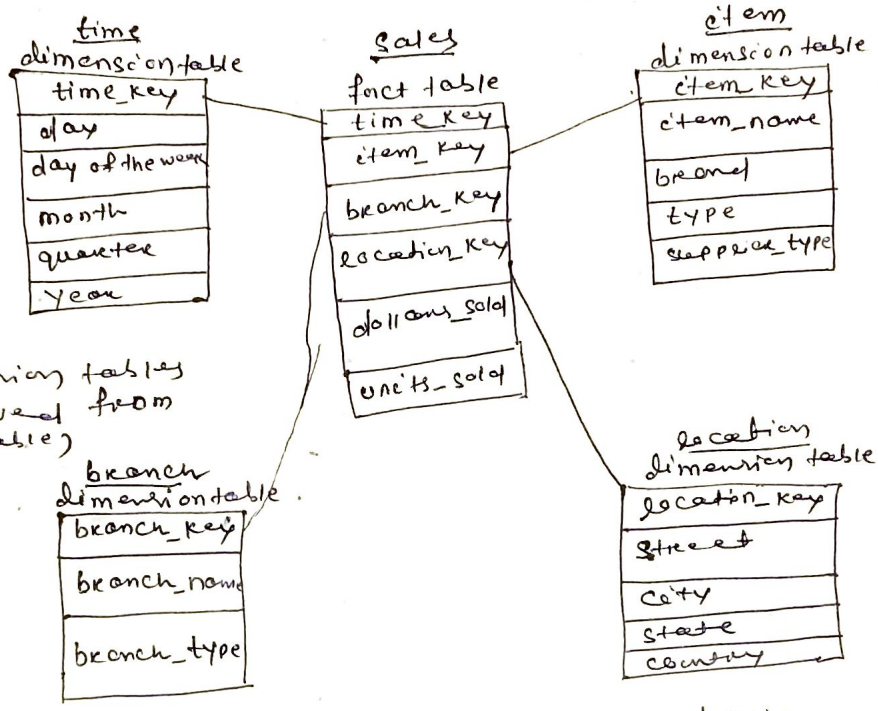
(2) Snowflake "

(3) Fact constellation schema. or Galaxy schema

Star schema

Here the data warehouse contains a fact table with a bulk of data with no redundancy in the the centre and a set of smaller dimension tables one for each dimension.

Ex



(Can dimension tables are derived from a fact table)

(Star schema of a data warehouse for sales)

Snowflake schema

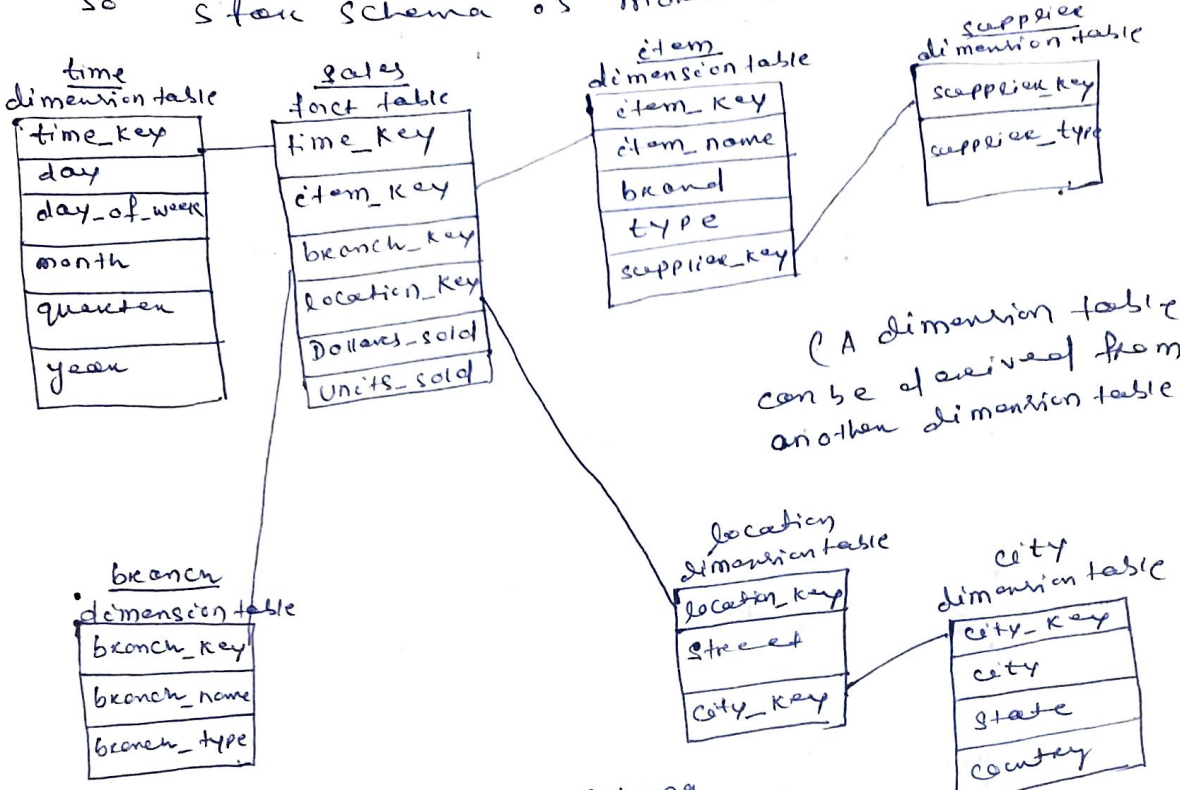
It is a modification to star schema where some dimensional tables are normalized into a set of smaller dimension tables forming a shape similar to a snowflake

Q Suppose that a data warehouse consists of the four dimensions time, item, branch, location and two measures dollars_sold, units_sold. Draw a star schema for the DW.

Since normalized it is easy to maintain and also it saves memory space and also reduces redundancies.

But the saving of memory space is negligible and also more joins are required to execute a query.

So star schema is more popular than it.



(Snowflake schema of a data warehouse for sales)

Fact constellation or Galaxy schema

Multiple fact tables share the dimension tables so viewed as a collection of stars.

Let us add a fourth dimension to the sales table, supplier. Data cubes are n-dimensional and do not confine to 3-D. We can think a 4D-cube as being a series of 3-D cubes. Each of these 3-D cubes are known as cuboid. dimension. Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the

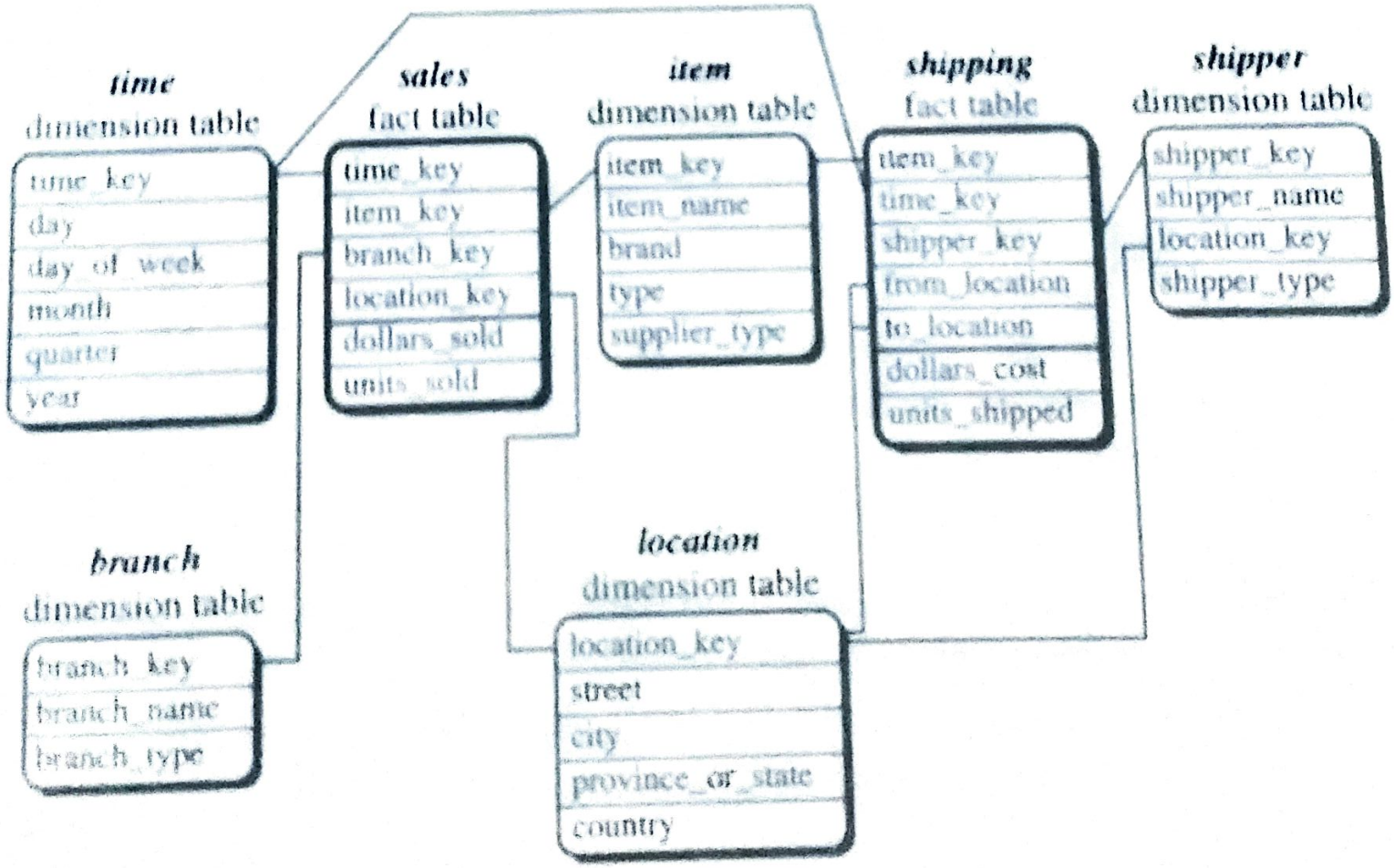


Figure 3.6 Fact constellation schema of a data warehouse for sales and shipping.

3.2.3 Examples for Defining Star, Snowflake,

data by OLAP operations), and *data mining* (which supports knowledge discovery). OLAP-based data mining is referred to as **OLAP mining**, or on-line analytical mining (**OLAM**), which emphasizes the interactive and exploratory nature of OLAP mining.

Exercises

3.1 State why, for the integration of multiple heterogeneous information sources, many companies in industry prefer the *update-driven approach* (which constructs and uses data warehouses), rather than the *query-driven approach* (which applies wrappers and integrators). Describe situations where the query-driven approach is preferable over the update-driven approach.

3.2 Briefly compare the following concepts. You may use an example to explain your point(s).

(a) Snowflake schema, fact constellation, starnet query model

(b) Data cleaning, data transformation, refresh

(c) Enterprise warehouse, data mart, virtual warehouse

3.3 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) Enumerate three classes of schemas that are popularly used for modeling data warehouses.

(b) Draw a schema diagram for the above data warehouse using one of the schema classes listed in (a).

(c) Starting with the base cuboid [*day*, *doctor*, *patient*], what specific *OLAP operations* should be performed in order to list the total fee collected by each doctor in 2004?

(d) To obtain the same list, write an SQL query assuming the data are stored in a relational database with the schema *fee* (*day*, *month*, *year*, *doctor*, *hospital*, *patient*, *count*, *charge*).

3.4 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 *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* courses for 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)?

3.5 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?

(c) *Bitmap indexing* is useful in data warehousing. Taking this cube as an example, briefly discuss advantages and problems of using a bitmap index structure.

3.6 A data warehouse can be modeled by either a *star schema* or a *snowflake schema*. Briefly describe the similarities and the differences of the two models, and then analyze their advantages and disadvantages with regard to one another. Give your opinion of which might be more empirically useful and state the reasons behind your answer.

3.7 Design a data warehouse for a regional weather bureau. The weather bureau has about 1,000 probes, which are scattered throughout various land and ocean locations in the region to collect basic weather data, including air pressure, temperature, and precipitation at each hour. All data are sent to the central station, which has collected such data for over 10 years. Your design should facilitate efficient querying and on-line analytical processing, and derive general weather patterns in multidimensional space.

3.8 A popular data warehouse implementation is to construct a multidimensional database, known as a data cube. Unfortunately, this may often generate a huge, yet very sparse multidimensional matrix. Present an example illustrating such a huge and sparse data cube.

3.9 Regarding the *computation of measures* in a data cube:

(a) Enumerate three categories of measures, based on the kind of aggregate functions used in computing a data cube.

(b) For a data cube with the three dimensions *time, location,* and *item*, which category does the function *variance* belong to? Describe how to compute it if the cube is partitioned into many chunks.
Hint: The formula for computing *variance* is $\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x}_i)^2$, where $\bar{x}_i$ is the average of $N x_i$s.

(c) Suppose the function is "*top 10 sales*". Discuss how to efficiently compute this measure in a data cube.

3.10 Suppose that we need to record three measures in a data cube: **min**, **average**, and **median**. Design an efficient computation and storage method for each measure given