Data Warehousing and Data Mining

4th Class

1st course

Edited By

Dr. Khalil I. Ghathwan

2021-2020

History of Data

The past couple of decades have seen a dramatic increase in the amount of Information or data being stored in electronic format. This accumulation of data has taken place at an explosive rate. It has been estimated that the amount of information in the world doubles every 20 months and the sizes as well as number of databases are increasing even faster. There are many examples that can be cited. Point of sale data in retail, policy and claim data in insurance, medical history data in healthcare, financial data in banking and securities, are some instances of the types of data that is being collected.

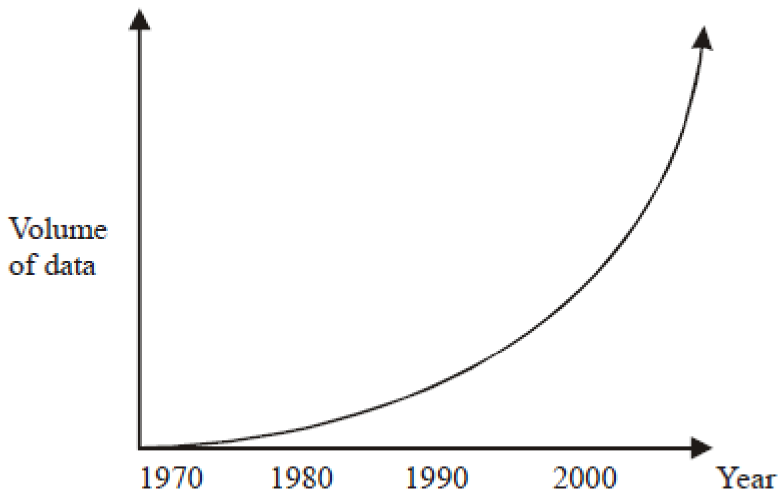


Fig. 1.1 The growing base of data

What is a Data Warehouse?

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. In addition to a relational database, a data warehouse environment includes an extraction, transportation a transformation, and an online analytical processing

(OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.

A common way of introducing data warehousing is to refer to the characteristics of a data warehouse:

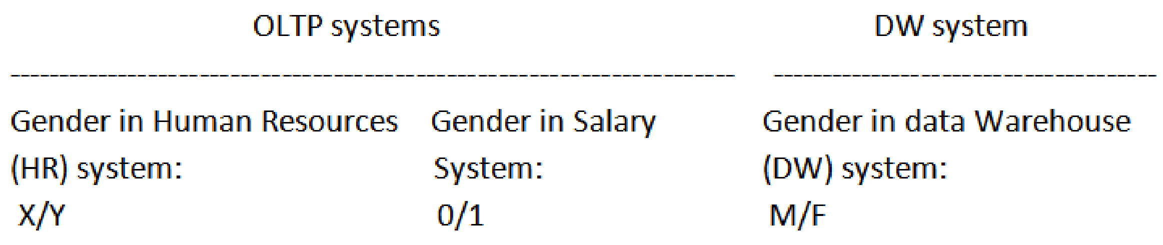
1. Subject Oriented
2. Integrated
3. Nonvolatile
4. Time Variant

1. Subject Oriented

Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?" This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented.

2. Integrated

Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated. For example, gender field might be represented differently from one transactional system to another. i.e. X/Y in human resources system or 0/1 in the salary system, while the data warehouse has one consistent format like M/F as illustrated in figure 2.



OLTP systems		DW system
Gender in Human Resources (HR) system:	Gender in Salary System:	Gender in data Warehouse (DW) system:
X/Y	0/1	M/F

Figure 2: Same attribute with different formats in dissimilar systems.

3. Nonvolatile

The final crucial characteristic of the data warehouse is Non-Volatile, data inside operational systems are updated (Insert, Delete, or Read and Write) also known as “CRUD”. While, data warehouse data are considered static or not up-to-date but only for Read. DW refreshed from time to time in a batch mode. This property enables DW to be heavily optimized for query processing; figure 3 illustrates a simple comparison of this feature between operational and informational systems.

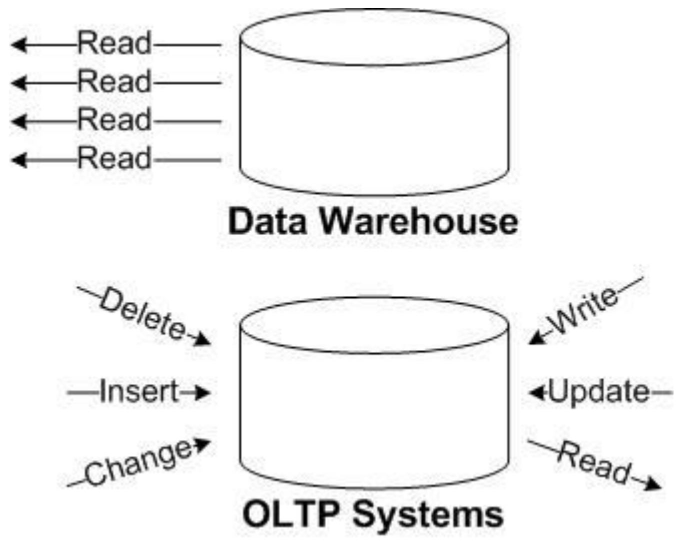


Figure 3: Simple comparison between OLTP and DW.

4. Time Variant

Operational systems store current up-to-date data for fast and atomic transactions.

Old data transferred to archives since it affects performance, while the data warehouse contains a large amount of historical information that helps the analysts to discover trends in business. Every entry in the data warehouse is time-stamped through time dimension. Noteworthy, data in OLTP systems maintained for few months or less than a year, while DW data kept for over three years (3 - 10).

Differences between Operational Database Systems and Data Warehouses

The major task of online operational database systems is to perform online transaction and query processing. These systems are called online transaction processing (OLTP) systems. They cover most of the day-to-day operations of an organization such as purchasing, inventory, banking, payroll, registration, and accounting. Data warehouse systems, on the other hand, serve users or knowledge workers in the role of data analysis and decision making. These systems are known as online analytical processing (OLAP) systems. The major distinguishing features of OLTP and OLAP are summarized as follows:

1. Users and system orientation: An OLTP system is application-oriented and is used for transaction and query processing by clerks, clients, and information technology professionals. An OLAP system is subject-oriented and is used for data analysis by knowledge workers, including managers, executives, and analysts.

2. Data contents: An OLTP system manages current data that, typically, are too detailed to be easily used for decision making, while OLAP system manages large amounts of historic data, provides facilities for summarization and aggregation, these features make the data easier to use for informed decision making.

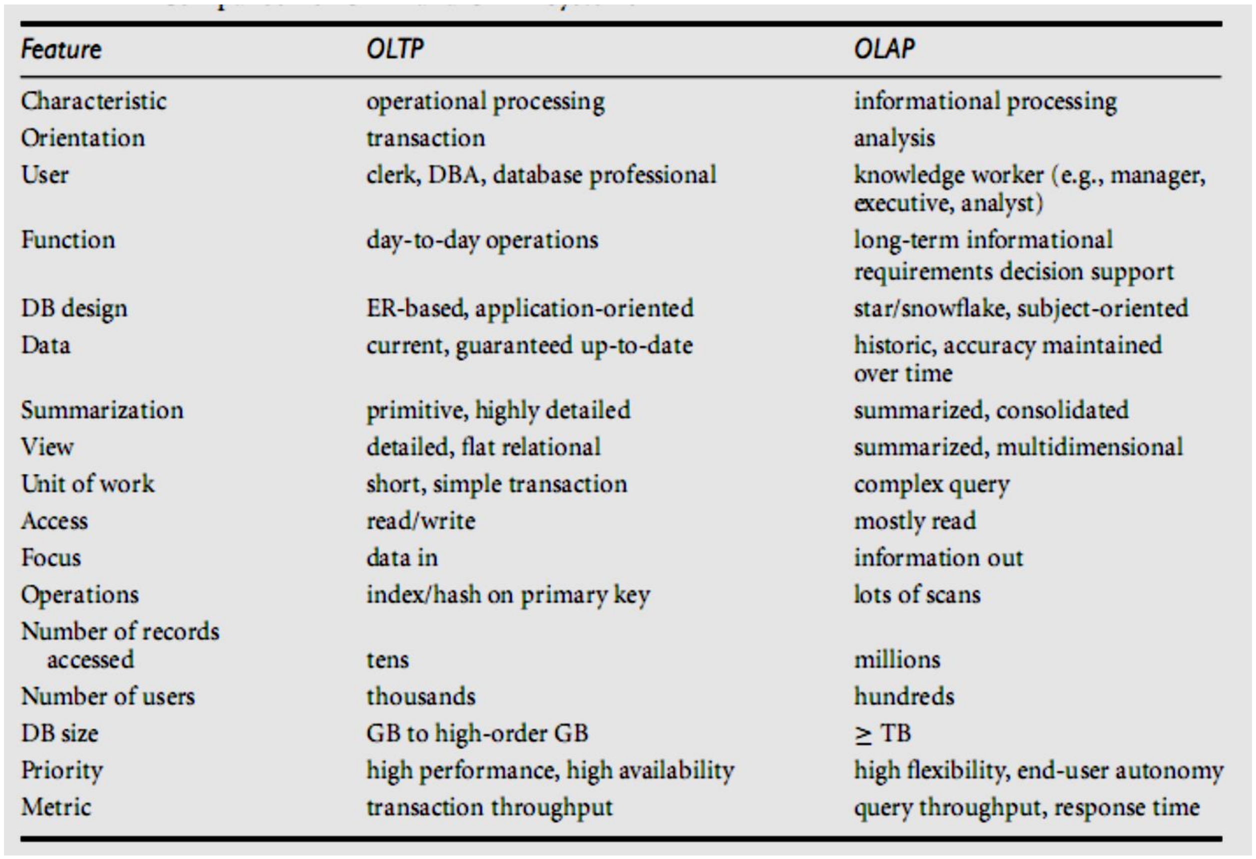
3. Database design: An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design. An OLAP system typically adopts either a star or a snowflake model and a subject-oriented database design.

4. View: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores.

5. 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 (because most data warehouses store historic rather than up-to-date information), although many could be complex queries.

Other features that distinguish between OLTP and OLAP systems include database size, frequency of operations, and performance metrics. These are summarized in Table 1.

Table 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	≥ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

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. Various multidimensional models are shown: star schema, snowflake schema, and fact constellation.

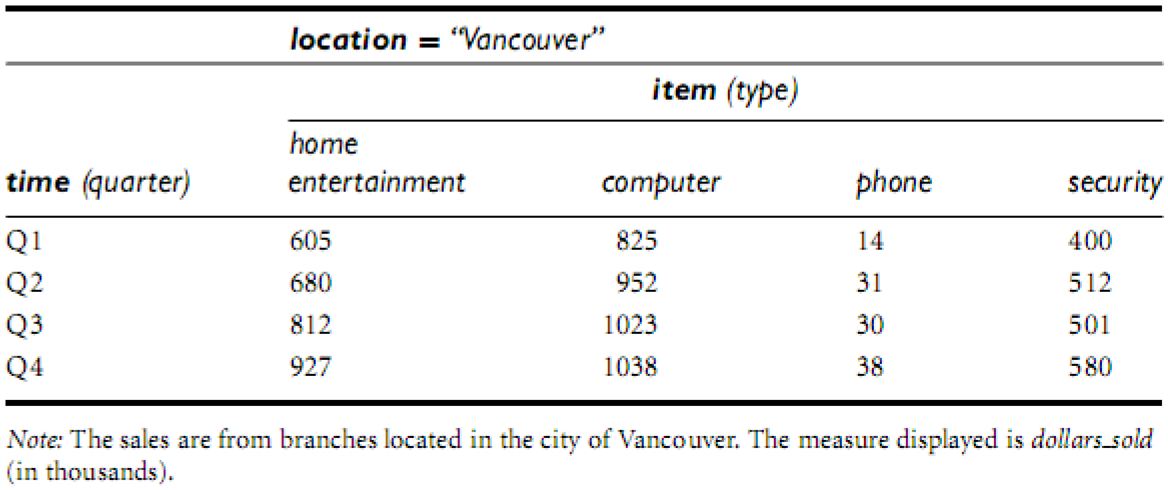
Data Cube: From Tables and Spreadsheets to Data Cubes

"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 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 at which the items were sold. Each dimension may have a table associated with it, called a dimension table. For example, a dimension table for item may contain the attributes item name, brand, and type. Dimension tables can be specified by users or experts, or automatically generated and adjusted based on data distributions.

A multidimensional data model is typically organized around a central theme, such as sales. This theme is represented by a fact table. Facts are numeric measures. Think of them as the quantities by which we want to analyze relationships between dimensions. Examples of facts for a sales data warehouse include dollars sold (sales amount in dollars) and units sold (number of units sold). The fact table contains the names of the facts, or measures, as well as keys to each of the related dimension tables.

We usually think of cubes as 3-D geometric structures, in data warehousing the data cube is n-dimensional. 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 that is, in fact, a table or spreadsheet for sales data from AllElectronics. In particular, we will look at the AllElectronics sales data for items sold per quarter in the city of Vancouver. These data are shown in Table 2. 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).

Table 2. 2-D View of Sales Data for *AllElectronics* According to *time* and *item*

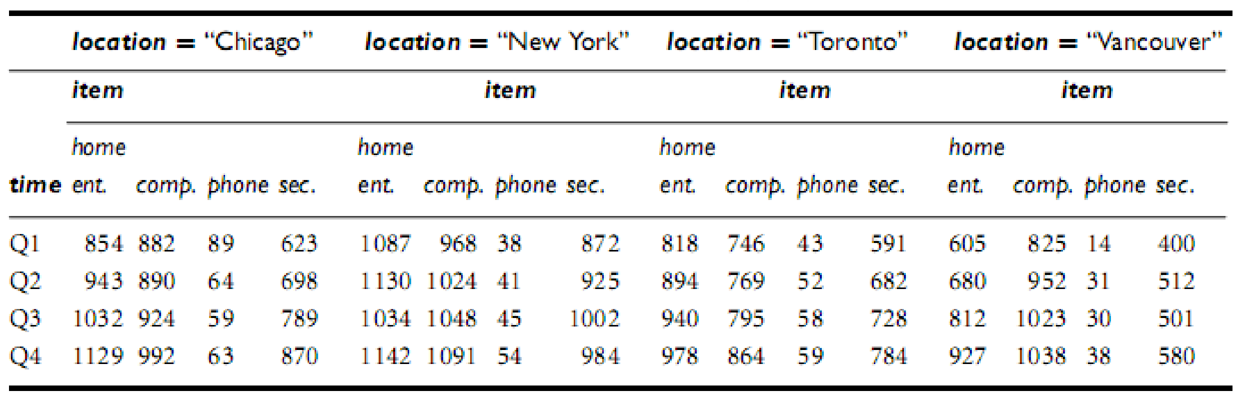


	location **= "Vancouver"**			
	item *(type)*			
time *(quarter)*	*home entertainment*	*computer*	*phone*	*security*
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

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

Now, suppose that we would like to view the sales data with a third dimension. For instance, suppose we would like to 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 3. 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, as in Figure 5.

Table 3. 3-D View of Sales Data for AllElectronics According to time, item, and Location.



	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
	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	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

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

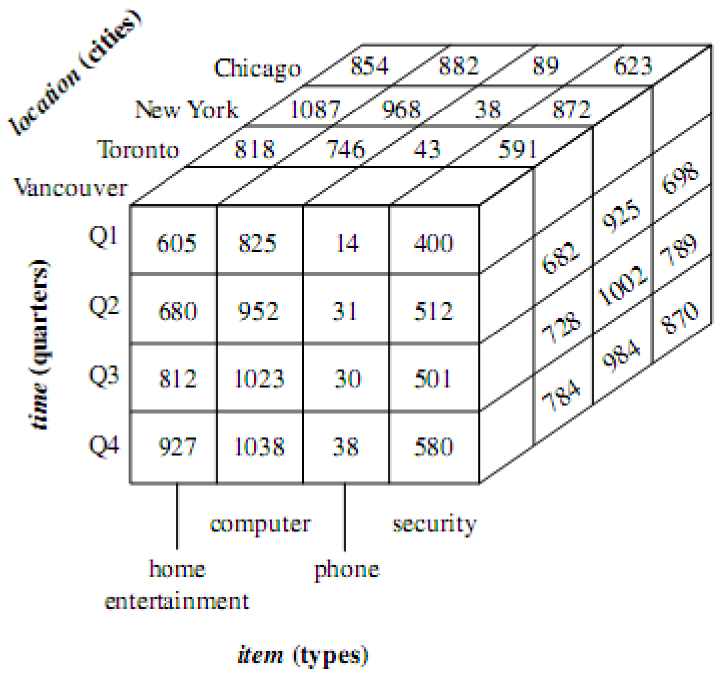


Figure 5. A 3-D data cube representation of the data in Table 3, according to time, item, and location.

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 6.

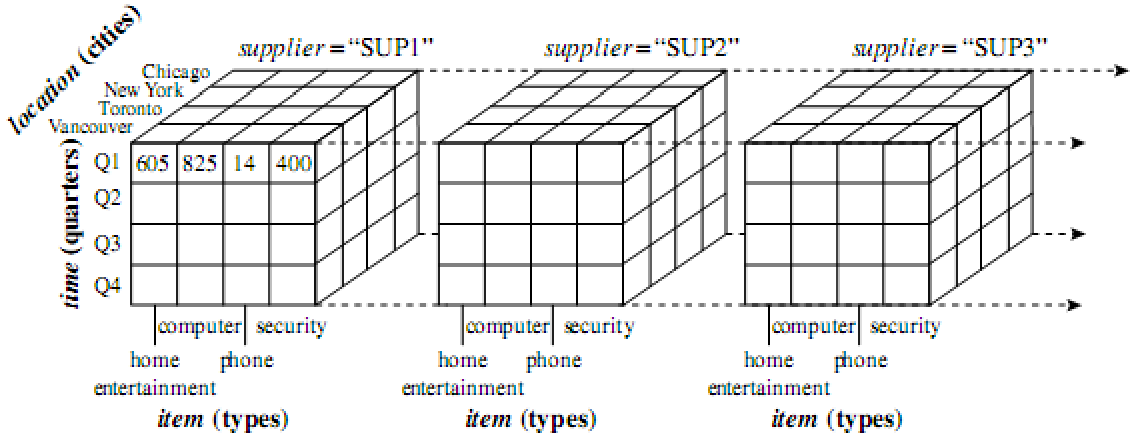


Figure 6. A 4-D data cube representation of sales data, according to *time*, *item*, *location*, and *supplier*. For improved readability, only some of the cube values are shown.

If we continue in this way, we may display any n-dimensional data as a series of (n-1) dimensional "cubes." The data cube is a metaphor for multidimensional data storage. The actual physical storage of such data may differ from its logical representation. Tables 2 and 3 show the data at different degrees of summarization. In the data warehousing research literature, a data cube like those shown in Figures 5 and 6 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 cuboids is then referred to as a data cube. Figure 7 shows a lattice of cuboids forming a data cube for the dimensions time, item, location, and supplier.

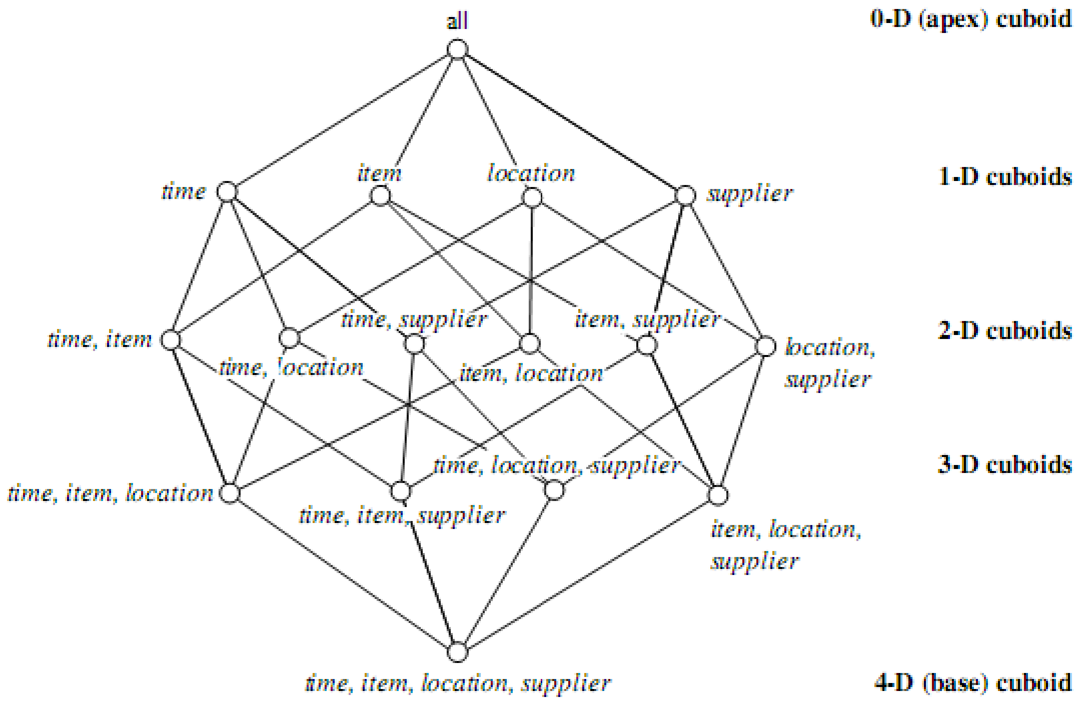


Figure 7. Lattice of cuboids, making up a 4-D data cube for time, item, location, and

Supplier. Each cuboid represents a different degree of summarization.

The cuboid that holds the lowest level of summarization is called the base cuboid. For example, the 4-D cuboid in Figure 6 is the base cuboid for the given time, item,

Location, and supplier dimensions. Figure 5 is a 3-D (nonbase) cuboid for time, item,

And location, summarized for all suppliers. The 0-D cuboid, which holds the highest

Level of summarization, is called the apex cuboid. In our example, this is the total sales, summarized over all four dimensions. The apex cuboid is typically denoted by all.

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

The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for online transaction processing. A data warehouse, however, requires a concise, subject-oriented schema that facilitates online data analysis. The most popular data model for a data warehouse is a multidimensional model, which can exist in the form of a star schema, a snowflake schema, or a fact constellation schema.

1. **Star schema**: The most common modeling paradigm is the star schema, in which the data warehouse contains (1) a large central table (fact table) containing the bulk of the data, with no redundancy, and (2) a set of smaller attendant tables (dimension tables), one for each dimension. The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

Example

A star schema for AllElectronics sales is shown in Figure 8. Sales are considered along four dimensions: time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold. Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes. For example, the location dimension table contains the attribute set location key, street, city, province or state, country. This constraint may introduce some redundancy. For example, “Urbana” and “Chicago” are both cities in the state of Illinois, USA. Entries for such cities in the location dimension table will create redundancy among the attributes province or state and country; that is ..., Urbana, IL, USA and ..., Chicago, IL, USA. Moreover, the attributes within a dimension table may form a hierarchy.

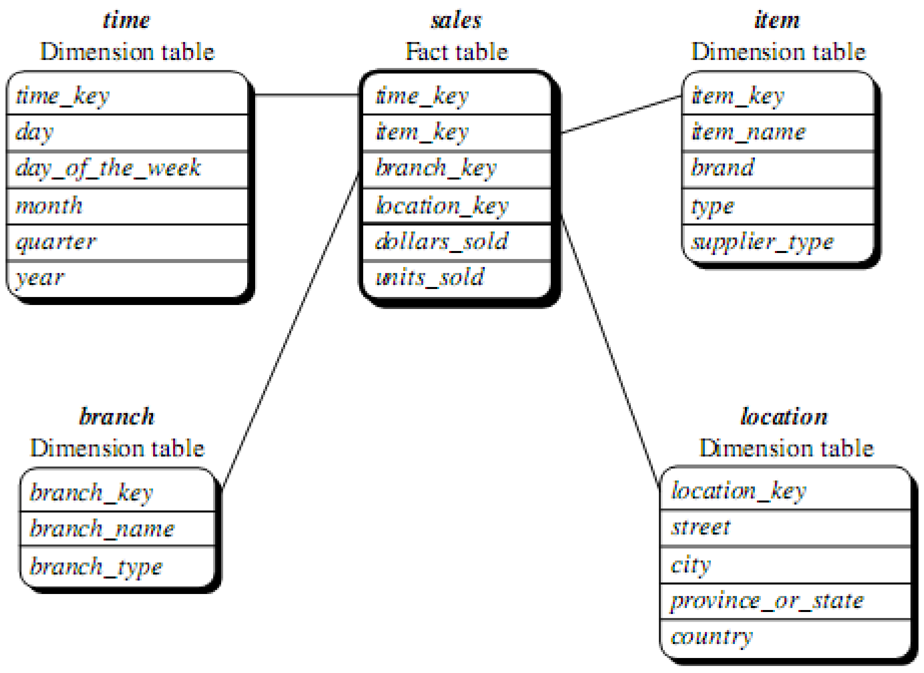


Figure 8. Star schema of sales data warehouse.

2. 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 major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies. Such a table is easy to maintain and saves storage space. However, this space savings is negligible in comparison to the typical magnitude of the fact table. Furthermore, the snowflake structure can reduce the effectiveness of browsing, since more joins will be needed to execute a query. Consequently, the system performance may be adversely impacted.

Example

A snowflake schema for AllElectronics sales is given in Figure 9. Here, the sales fact table is identical to that of the star schema in Figure 8. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables. For example, the item dimension table now contains the attributes item key, item name, brand, type, and supplier key, where supplier key is linked to the supplier dimension table, containing supplier key and supplier type information.

Similarly, the single dimension table for location in the star schema can be normalized into two new tables: location and city. The city key in the new location table links to the city dimension. Notice that, when desirable, further normalization can be performed on province or state and country in the snowflake schema.

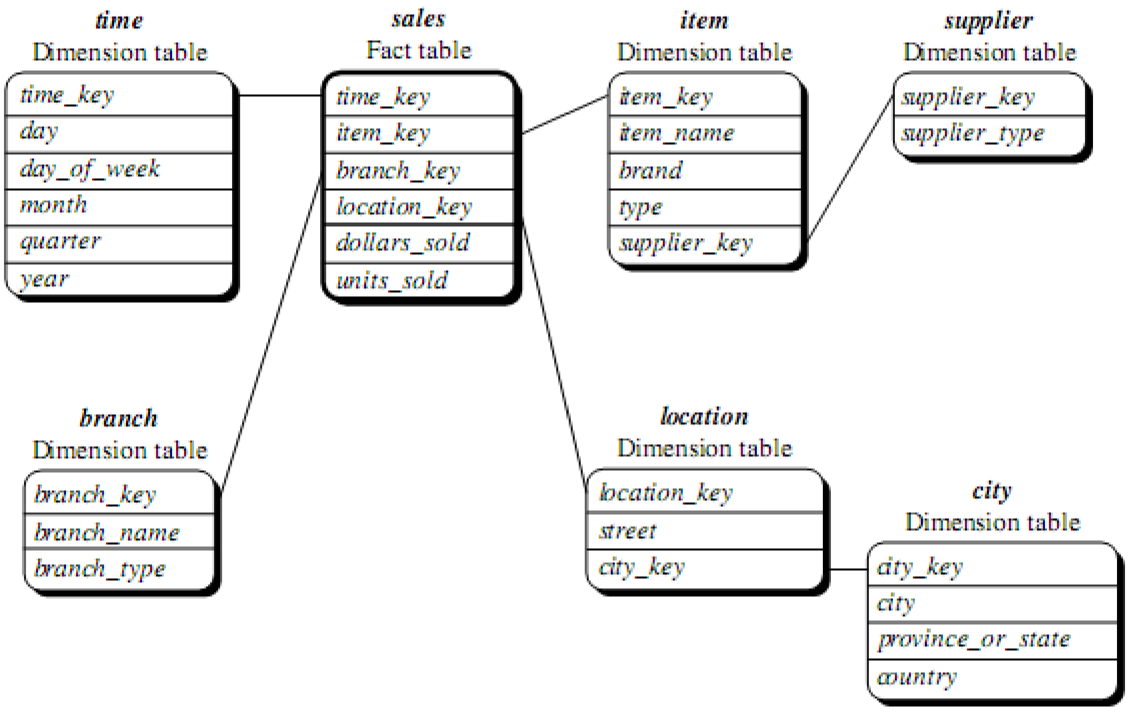


Figure 9. Snowflake schema of a sales data warehouse.

3. 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.

Example

A fact constellation schema is shown in Figure 10. This schema specifies two fact tables, sales and shipping. The sales table definition is identical to that of the star schema (Figure 8). The shipping table has five dimensions, or keys—item key, time key, shipper key, from_location, and to_location—and two measures—dollars cost and units shipped. 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.

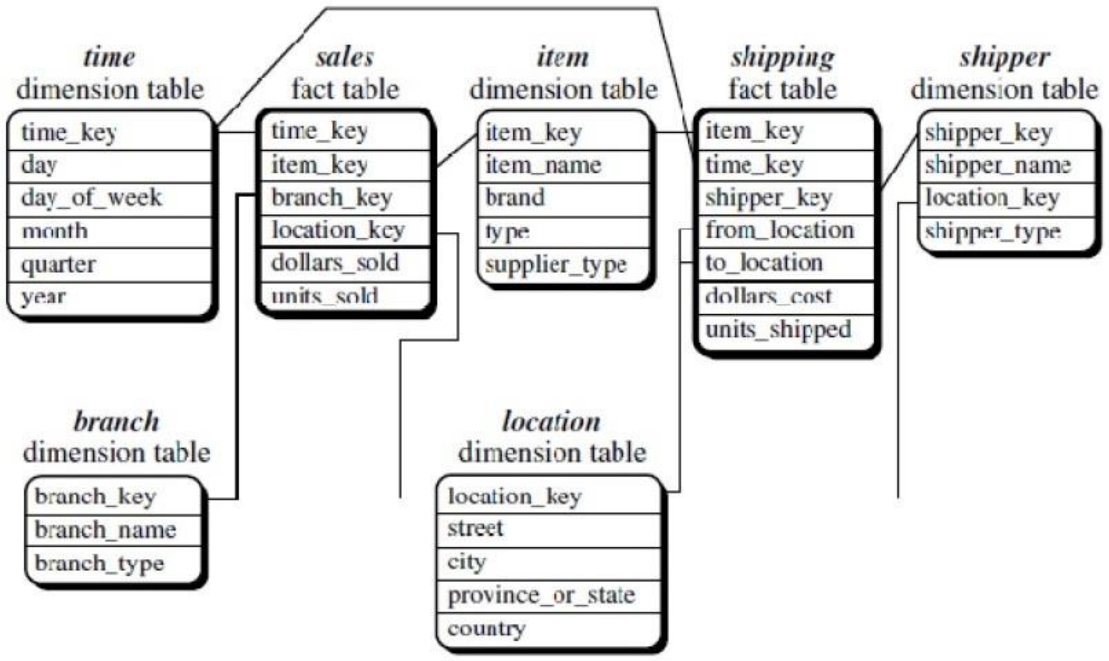


Figure 10: A fact constellation schema

OLAP Operations on a multidimensional data model

"How are concept hierarchies useful in OLAP?" 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 analysis of the data. Hence, OLAP provides a user-friendly environment for interactive data analysis.

Example

Let's look at some typical OLAP operations for multidimensional data. Each of the following operations described is illustrated in Figure 13. At the center of the figure is a data cube for AllElectronics sales. The cube contains the dimensions location, time, and item, where location is aggregated with respect to city values, time is aggregated with respect to quarters, and item is aggregated with respect to item types. To aid in our explanation, we refer to this cube as the central cube. The data examined are for the cities Chicago, New York, Toronto, and Vancouver.

Roll-up: The roll-up operation (also called the drill-up operation by some vendors) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction. Figure 13 shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for location given in Figure 10. 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. In other words, rather than grouping the data by city, the resulting cube groups the data by country.

When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube. For example, consider a sales data cube containing only the location and time dimensions. Roll-up may be performed by removing, say, the time dimension, resulting in an aggregation of the total sales by location, rather than by location and by time.

Drill-down: 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. Figure 13 shows the result of a 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. The resulting data cube details the total sales per month rather than summarizing them by quarter. Because a drill-down adds more detail to the given data, it can also be performed by adding new dimensions to a cube.

Slice and dice: The slice operation performs a selection on one dimension of the given cube, resulting in a sub cube. Figure 13 shows a slice operation where the sales data are selected from the central cube for the dimension time using the criterion time="Q1." The dice operation defines a sub cube by performing a selection on two or more dimensions. Figure 13 shows a 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 (rotate): Pivot (also called rotate) is a visualization operation that rotates the data axes in view to provide an alternative data presentation. Figure 11 shows a pivot operation where the item and location axes in a 2-D slice are rotated.

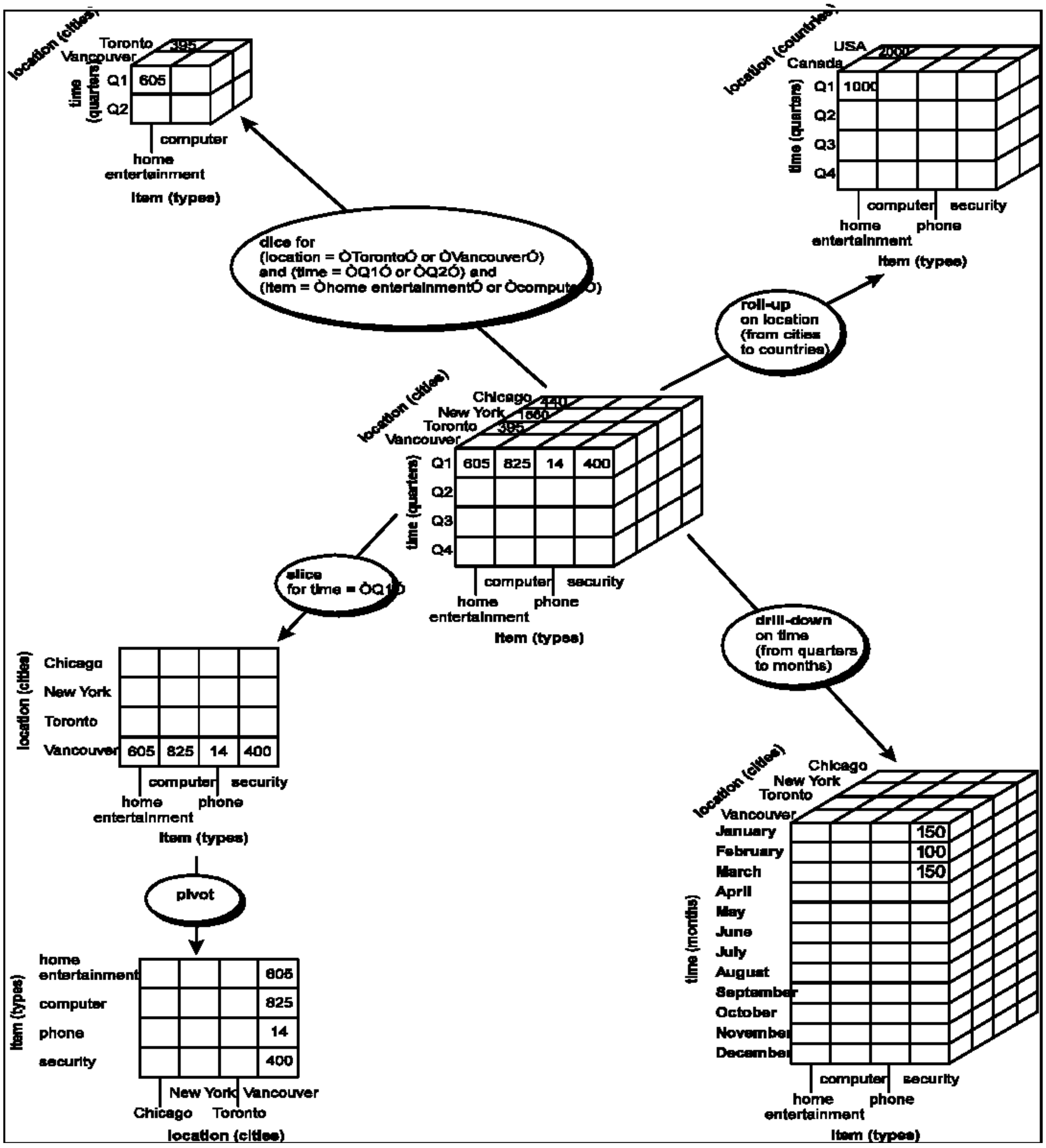


Figure 11. Examples of typical OLAP operations on multidimensional data.