Data Warehouse and OLAP

- **Data warehouses** generalize and consolidate data in multidimensional space. The construction of data warehouses involves data cleaning, data integration, and data transformation and can be viewed as an important preprocessing step for data mining. Moreover, data warehouses provide *on-line analytical processing (OLAP)* tools for the interactive analysis of multidimensional data of varied granularities, which facilitates effective data generalization and data mining. Many other data mining functions, such as association, classification, prediction, and clustering, can be integrated with OLAP operations to enhance interactive mining of knowledge at multiple levels of abstraction.

Data Warehouses

- According to William H. Inmon, a leading architect in the construction of data warehouse systems, "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process"
  - **Subject-oriented**: organized around major subjects, such as customer, supplier, product, and sales; focus on the modeling and analysis of data for decision makers.
  - **Integrated**: usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and on-line transaction records. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.
  - **Time-variant**: Data are stored to provide information from a historical perspective (e.g., the past 5–10 years). Every key structure in the data warehouse contains, either implicitly or explicitly, an element of time.
  - **Nonvolatile**: A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a data warehouse does not require transaction processing, recovery, and concurrency control mechanisms.

Data Warehouses

- Data warehousing is also very useful from the point of view of *heterogeneous database integration*.
  - **query-driven approach**: build wrappers and integrators (or mediators), on top of multiple, heterogeneous databases. When a query is posed to a client site, a metadata dictionary is used to translate the query into queries appropriate for the individual heterogeneous sites involved.
  - **update-driven approach**: information from multiple, heterogeneous sources is integrated in advance and stored in a warehouse for direct querying and analysis.

OLTP vs OLAP

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

- **Data contents**: An OLTP system manages current data that, typically, are too detailed to be easily used for decision making. An OLAP system manages large amounts of historical data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity. These features make the data easier to use in informed decision making.

- **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 *snowflake* model

OLTP vs OLAP

- **View**: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historical data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.

- **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 historical rather than up-to-date information), although many could be complex queries.

OLTP vs OLAP

Table 3.1 Comparison between 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	historical; 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	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

- *“why not perform on-line analytical processing directly on relational databases instead of spending additional time and resources to construct a separate data warehouse?”*
  - high performance of both systems
  - Concurrency control and recovery mechanisms, if applied for OLAP operations, may jeopardize the execution of concurrent transactions and thus substantially reduce the throughput of an OLTP system.
  - different structures, contents, and uses of the data in these two systems

A Multidimensional Data Model

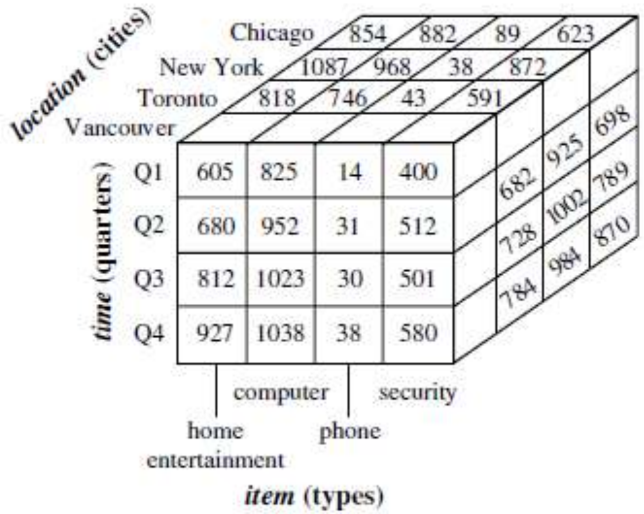
- A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by **dimensions** and **facts**.
  - **dimensions** are the perspectives or entities with respect to which an organization wants to keep records (Each dimension may have a table associated with it, called a dimension table, which further describes the dimension)
  - **facts** are numerical measures (the quantities by which we want to analyze relationships between dimensions)
- Although we usually think of cubes as 3-D geometric structures, in data warehousing the **data cube is *n*-dimensional**.

A 2-D view of sales data for *AllElectronics* according to the dimensions *time* and *item*, where the sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

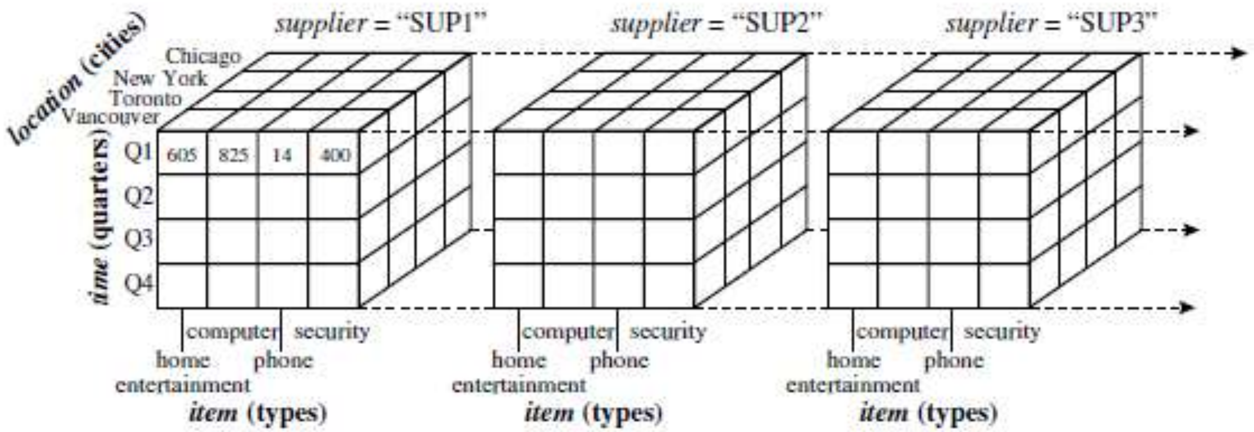
	location = "Vancouver"			
	item (*type*)			
time (*quarter*)	*home entertainment*	*computer*	*phone*	*security*
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

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			
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



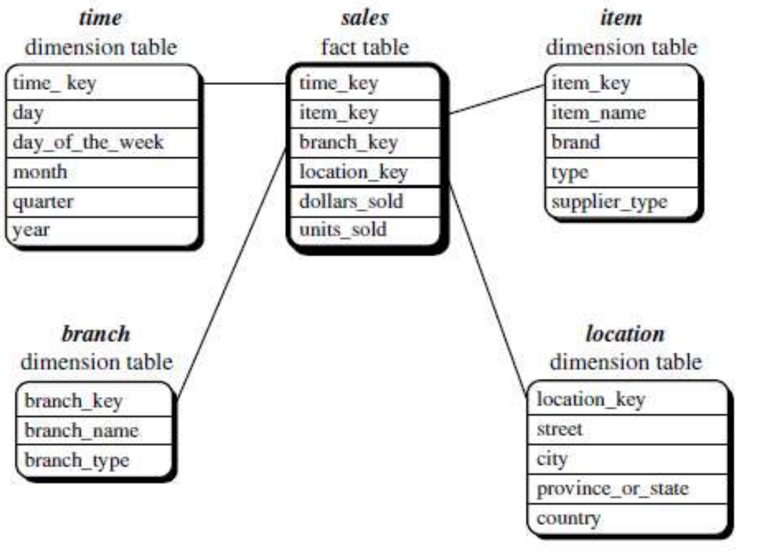
A 3-D data cube representation of the data in Table 3.3, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).



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.

Schemas for Multidimensional Databases

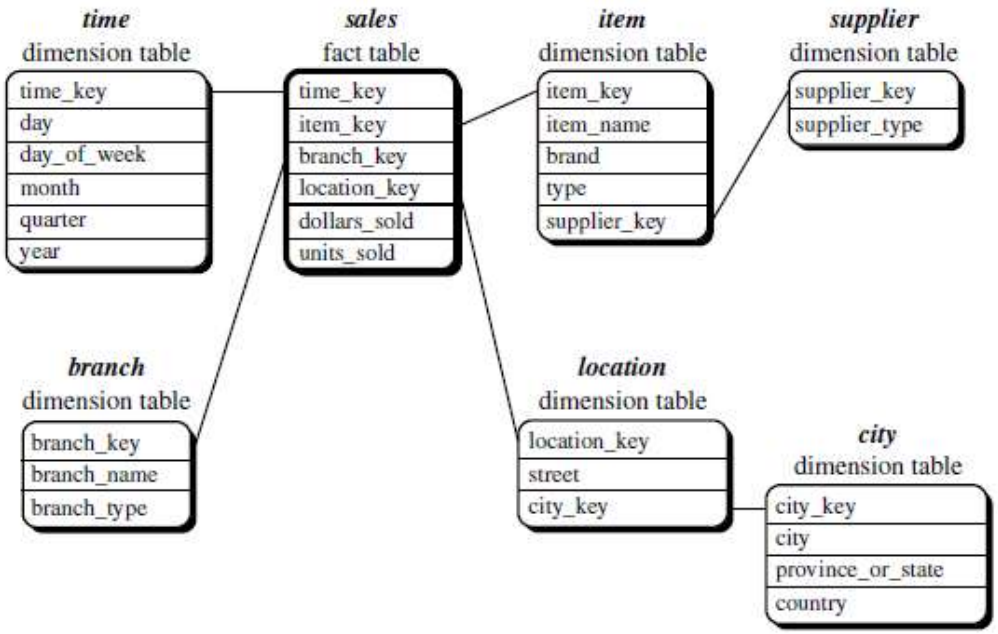
- **Star schema**: The most common modeling paradigm is the star schema, in which the data warehouse contains (1) a large central table (fact table) containing the bulk of the data, with no redundancy, and (2) a set of smaller attendant tables (dimension tables), one for each dimension. The schema graph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



Star schema of a data warehouse for sales.

Schemas for Multidimensional Databases

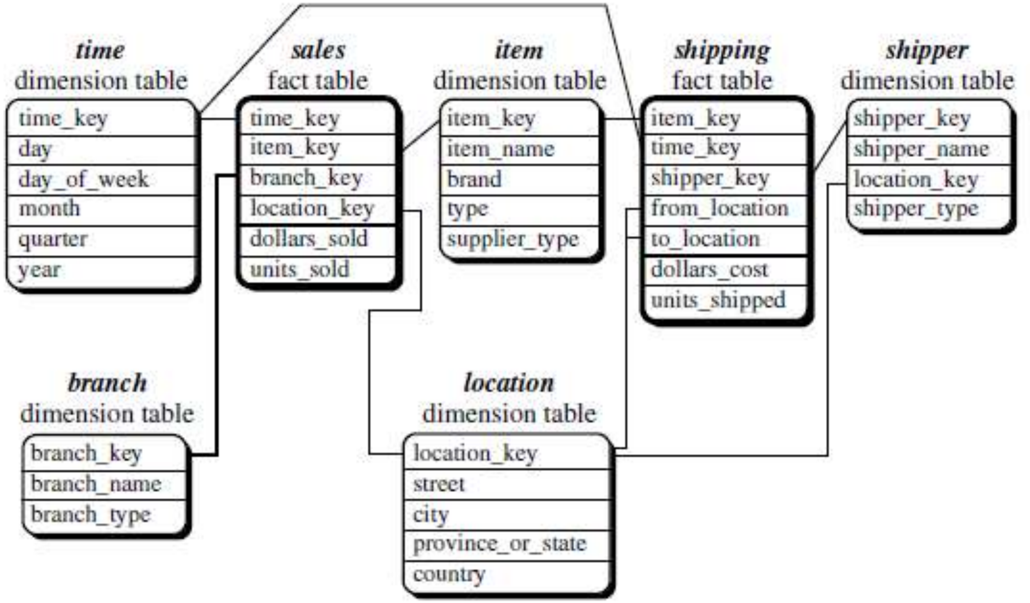
- **Snowflake schema:** The snowflake schema is a variant of the star schema model, where some dimension tables are *normalized*, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.



Snowflake schema of a data warehouse for sales.

Schemas for Multidimensional Databases

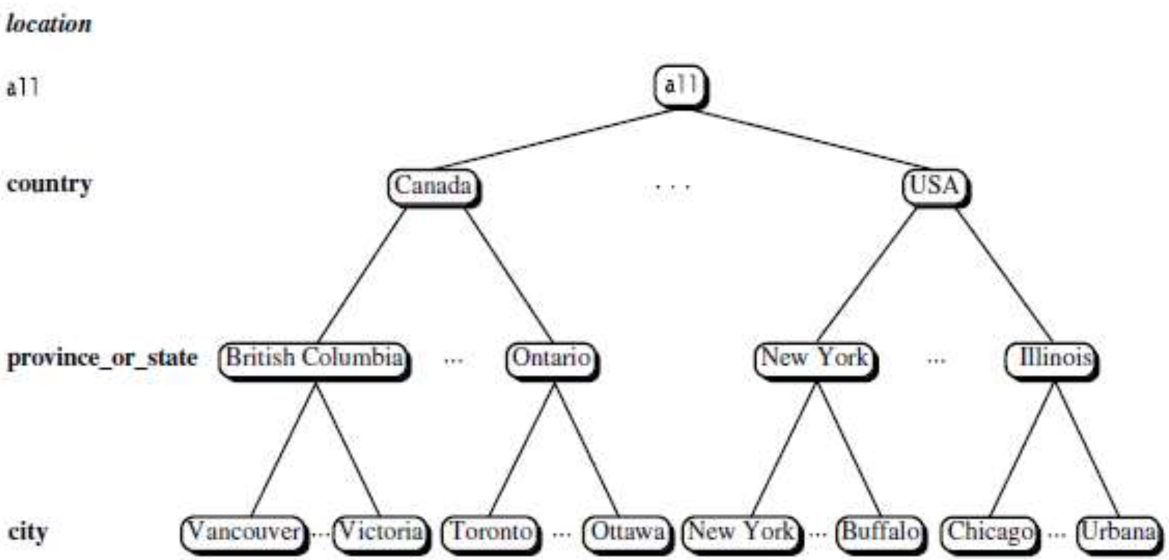
- **Fact constellation**: Sophisticated applications may require multiple fact tables to *share* dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.



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

Concept Hierarchies

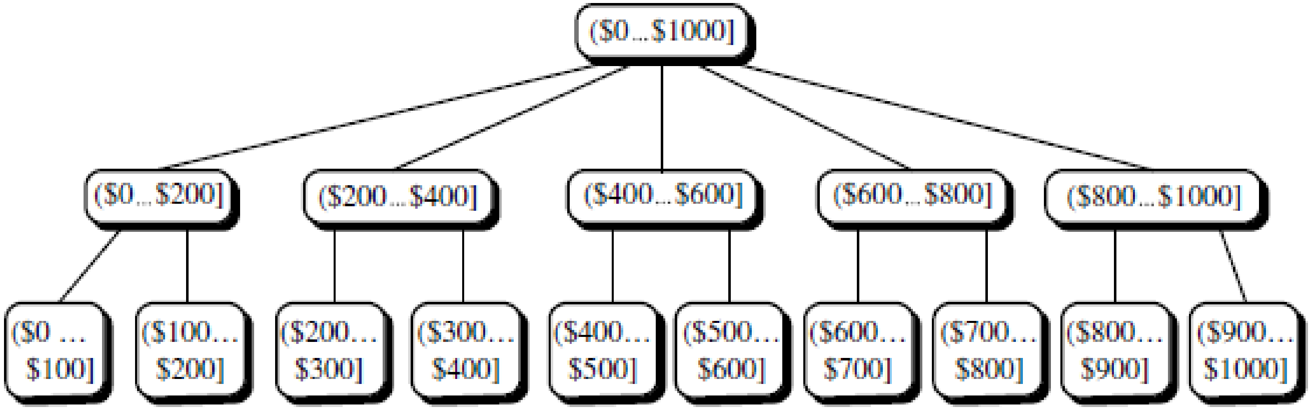
- A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.
- **Concept hierarchies allow data to be handled at varying levels of abstraction**



A concept hierarchy for the dimension *location*. Due to space limitations, not all of the nodes of the hierarchy are shown (as indicated by the use of "ellipsis" between nodes).

Concept Hierarchies

- Concept hierarchies may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a set-grouping hierarchy. A total or partial order can be defined among groups of values.
- There may be more than one concept hierarchy for a given attribute or dimension, based on different user viewpoints. For instance, a user may prefer to organize *price* by defining ranges for *inexpensive, moderately priced*, and *expensive*.

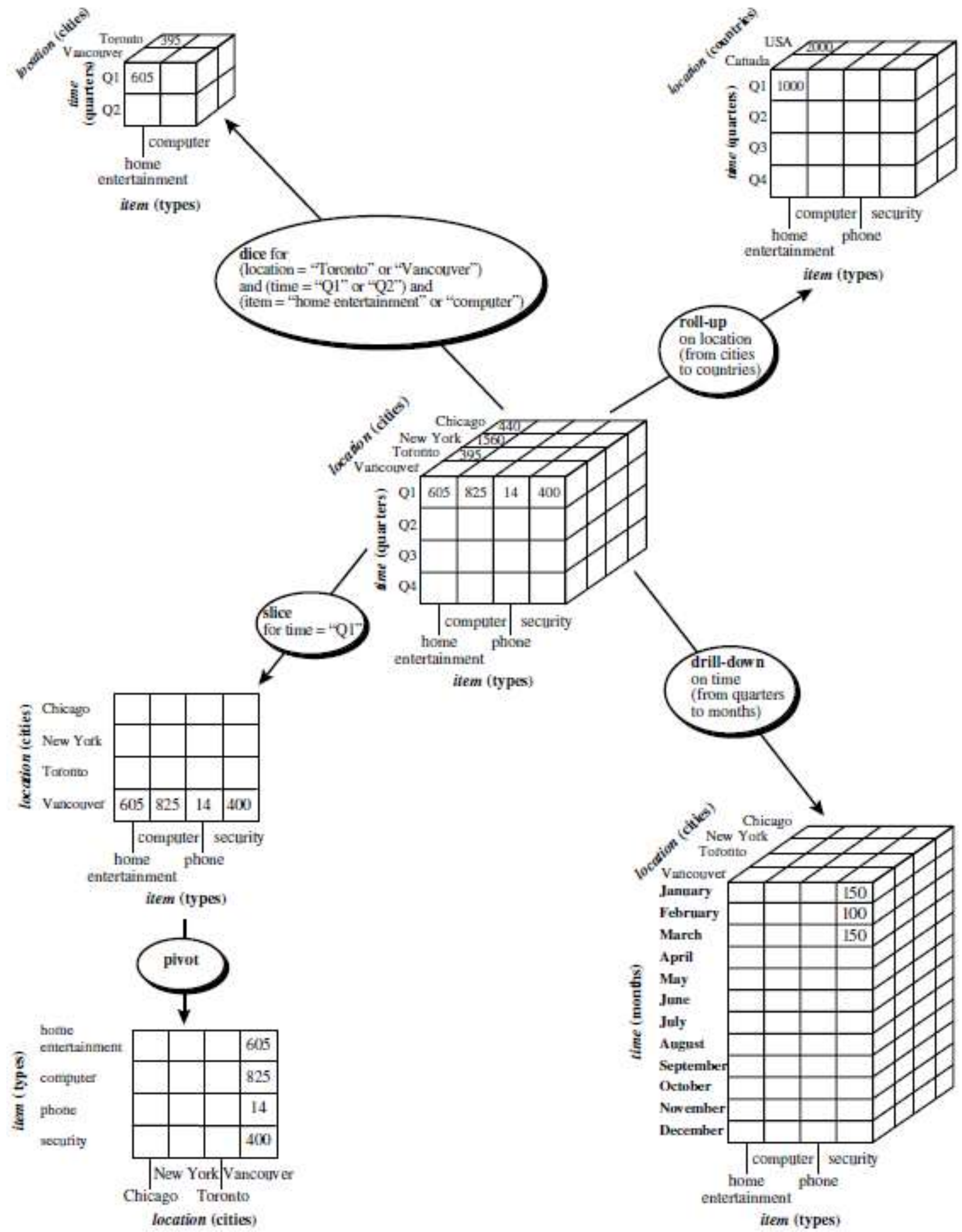


A concept hierarchy for the attribute *price*.

OLAP Operations in the 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 interactive querying and analysis of the data at hand. Hence, OLAP provides a user-friendly environment for interactive data analysis.



Examples of typical OLAP operations on multidimensional data.

OLAP Operations in the Multidimensional Data Model

- **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 (slide 19) shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for *location* given in Figure 3.7. 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 two dimensions *location* and *time*. 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.

OLAP Operations in the Multidimensional Data Model

- **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 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 fromthe 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. For example, a drill-down on the central cube of Figure 3.10 can occur by introducing an additional dimension, such as *customer group*.

OLAP Operations in the Multidimensional Data Model

- **Slice and dice:** The *slice* operation performs a selection on one dimension of the given cube, resulting in a subcube. Figure 3.10 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 subcube by performing a selection on two or more dimensions. Figure 3.10 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 in order to provide an alternative presentation of the data. Figure shows a pivot operation where the *item* and *location* axes in a 2-D slice are rotated. Other examples include rotating the axes in a 3-D cube, or transforming a 3-D cube into a series of 2-D planes.

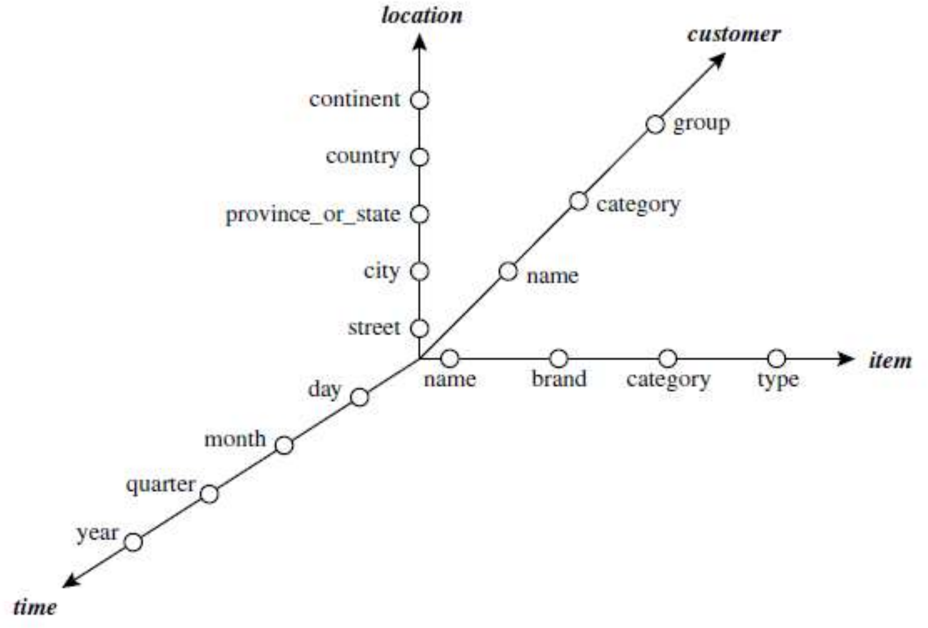
OLAP Operations in the Multidimensional Data Model

- Other OLAP operations: Some OLAP systems offer additional drilling operations. For example, drill-across executes queries involving (i.e., across) more than one fact table. The drill-through operation uses relational SQL facilities to drill through the bottom level of a data cube down to its back-end relational tables.

- Other OLAP operations may include ranking the top *N* or bottom *N* items in lists, as well as computing moving averages, growth rates, interests, internal rates of return, depreciation, currency conversions, and statistical functions.

A Starnet Query Model for Querying Multidimensional Databases

- The querying of multidimensional databases can be based on a starnet model. A starnet model consists of radial lines emanating from a central point, where each line represents a concept hierarchy for a dimension. Each abstraction level in the hierarchy is called a footprint. These represent the granularities available for use by OLAP operations such as drill-down and roll-up.



I Modeling business queries: a starnet model.

Indexing OLAP Data

- **Bitmap indexing method** - In the bitmap index for a given attribute, there is a distinct bit vector, ***Bv***, for each value *v* in the domain of the attribute. If the domain of a given attribute consists of *n* values, then *n* bits are needed for each entry in the bitmap index (i.e., there are *n* bit vectors). If the attribute has the value *v* for a given row in the data table, then the bit representing that value is set to 1 in the corresponding row of the bitmap index. All other bits for that row are set to 0.

Base table

RID	*item*	*city*
R1	H	V
R2	C	V
R3	P	V
R4	S	V
R5	H	T
R6	C	T
R7	P	T
R8	S	T

Item bitmap index table

RID	H	C	P	S
R1	1	0	0	0
R2	0	1	0	0
R3	0	0	1	0
R4	0	0	0	1
R5	1	0	0	0
R6	0	1	0	0
R7	0	0	1	0
R8	0	0	0	1

City bitmap index table

RID	V	T
R1	1	0
R2	1	0
R3	1	0
R4	1	0
R5	0	1
R6	0	1
R7	0	1
R8	0	1

Note: H for "home entertainment, " C for "computer, " P for "phone, " S for "security, " V for "Vancouver, " T for "Toronto."

Indexing OLAP data using bitmap indices.

Indexing OLAP Data

- The **join indexing method** gained popularity from its use in relational database query processing.

Join index table for
location/sales

location	*sales_key*
. . .	. . .
Main Street	T57
Main Street	T238
Main Street	T884
. . .	. . .

Join index table for
item/sales

item	*sales_key*
. . .	. . .
Sony-TV	T57
Sony-TV	T459
. . .	. . .

Join index table linking two dimensions
location/item/sales

location	*item*	*sales_key*
. . .	. . .	. . .
Main Street	Sony-TV	T57
. . .	. . .	. . .

Join index tables based on the linkages between the *sales* fact table and dimension tables for *location* and *item* shown in Figure 3.16.

From Data Warehousing to Data Mining

- **Data warehouse applications**: *information processing, analytical processing*, and *data mining*

  - **Information processing** supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts, or graphs. A current trend in data warehouse information processing is to construct low-cost Web-based accessing tools that are then integrated with Web browsers.

  - **Analytical processing** supports basic OLAP operations, including slice-and-dice, drill-down, roll-up, and pivoting. It generally operates on historical data in both summarized and detailed forms. The major strength of on-line analytical processing over information processing is the multidimensional data analysis of data warehouse data.

  - **Data mining** supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

From On-Line Analytical Processing (OLAP) to On-Line Analytical Mining (OLAM)

- On-line analytical mining (OLAM) (also called OLAP mining) integrates on-line analytical processing (OLAP) with data mining and mining knowledge in multidimensional databases. Among the many different paradigms and architectures of data mining systems, OLAM is particularly important for the following reasons:
  - High quality of data in data warehouses
  - Available information processing infrastructure surrounding data warehouses
  - OLAP-based exploratory data analysis
  - On-line selection of data mining functions