Module – 1

Data Warehousing &Modeling: Basic Concepts: Data Warehousing: A multitier Architecture, Data warehouse models: Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Data Cube: A multidimensional data model, Stars, Snowflakes and Fact constellations: Schemas for multidimensional Data models, Dimensions: The role of concept Hierarchies, Measures: Their Categorization and computation, Typical OLAP Operations.

1. Data warehouses

A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and which usually resides at a single site. Data warehouses are constructed via a process of data cleansing, data transformation, data integration, data loading, and periodic data refreshing.

1.1.1 What is a Data Warehouse?

Data warehousing provides architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions.

According to W. H. Inmon, a leading architect in the construction of data warehouse systems, **"A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management s decision-making process".**

The four keywords, subject-oriented, integrated, time-variant, and non-volatile, distinguish data warehouses from other data repository systems, such as relational database systems, transaction processing systems, and file systems. Let's take a closer look at each of these key features.

Subject-Oriented: Organized around major subjects, such as customer, product, sales. Focusing on the Modeling and analysis of data for decision makers, not on daily operations or transaction processing of an organization.A data warehouse focuses on the Modeling and analysis of data for decision makers. It provides a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

Integrated: Constructed by integrating multiple, heterogeneous data sources such as relational databases, flat files, on-line transaction records. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources E.g., Hotel price: currency, tax, breakfast covered, etc. When data is moved to the warehouse, it is converted.

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

Non-volatile: A physically separate store of data transformed from the operational environment Operational update of data does not occur in the data warehouse environment,does not require transaction processing, recovery, and concurrency control mechanisms. Requires only two operations in data accessing: **initial loading of data and access of data.**

A data warehouse is also often viewed as an architecture, constructed by integrating data from multiple heterogeneous sources to support structured and/or ad hoc queries, analytical reporting, and decision making.

Why,wecan view data warehousing as the process of constructing and using data warehouses.

A data warehouse is a semantically consistent data store that serves as a physical implementation of a decision support data model and stores the information on which an enterprise needs to make strategic decisions. A data warehouse is also often viewed as an architecture, constructed by integrating data from multiple heterogeneous sources to support structured and/or ad hoc queries, analytical reporting, and decision making.

The construction of a data warehouse requiresdata cleaning,dataintegration, anddata consolidation. The utilization of adata warehouse often necessitates a collection of decision support technologies. This allows "knowledge workers" (e.g., managers, analysts, and executives) to use the warehouse to quickly and conveniently obtain an overview of the data, and to make sound decisions based on information in the warehouse.

"How are organizations usingtheinformationfromdata warehouses?"

Many organizations use this information to support business decision-making activities, including

(1) Increasing customer focus, which includes the analysis of customer buying patterns (such as buying preference, buyingtime, budgetcycles, and appetites for spending);

(2) Repositioning products and managing product portfolios by comparing the performance of sales by quarter, by year, and by geographic regions inorderto fine-tuneproductionstrategies;

(3) Analysing operations and looking for sources ofprofit; and

(4) Managing customer relationships, making environmental corrections, and managing the cost of corporate assets.

Data warehouses can store and integrate historical information and support complex multidimensional queries. As a result, data warehousing has become popular in industry.

1.1.2 Differences between Operational Database Systems and Data Warehouses

Because most people are familiar with commercial relational database systems, itiseasy to understand whatadatawarehouse isby comparing these two kinds of systems.

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, manufacturing, banking, payroll, registration, and accounting.

Data warehouse systems, on the otherhand, serve users or knowledge workers in the role of data analysis and decision making. Such systems can organize and present data in various formats in order to accommodate the diverse needs of the different users. These systems are known as online analyticalprocessing (OLAP) systems.

The major distinguishing features between OLTP and OLAP are summarized as follows:

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 storesand manages information at different levels of granularity. These features make the data easier to use in informed decision making.

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

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.

Accesspatterns: The accesspatterns of anOLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms. However, accesses toOLAPsystems are mostly read-only operations(because mostdata warehouses storehistorical rather than up-to-date information), although many could be complex queries.

Other features that distinguish between OLTP and OLAP systems include database size,frequencyof operations, andperformance metrics. These are summarizedinTable4.1.

Table 4.1: 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 require-ments, 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	GB to high-order GB	$\geq$ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

1.1.3 Why a Separate Data Warehouse?

Because operational databases store huge amounts of data, you may wonder, "**why not perform online analytical processing directly on such databases instead of spending additional time and resources to construct a separate data warehouse?"**

A major reason for such a separation is to help promote the high performance of both systems. An operational database is designed and tuned from known tasks and workloads, such as indexing and hashing using primary keys,searchingforparticularrecords,and optimizing "canned" queries.

Onthe other hand, data warehouse queries are often complex. They involve the computationoflargegroups ofdata at summarizedlevels, and may requirethe use of specialdata organization, access, and implementation methodsbased on multidimensionalviews. Processing OLAP queries in operational databases would substantially degrade the performance of operational tasks.

Moreover, an operational database supports the concurrent processing of multiple transactions. Concurrency control and recovery mechanisms, such as locking and logging, are required to ensure the consistency and robustness of transactions. An OLAP query often needs read-only access of data records for summarization and aggregation. Concurrency control and recovery mechanisms, ifappliedfor suchOLAP operations, mayrisk the execution of concurrent transactions and thus substantially reduce the throughput of an OLTP system.

Finally, the separation of operational databases from data warehouses is basedon thedifferent structures, contents, and uses ofthedata in these two systems.

Decision support requires historical data, whereas operationaldatabases do not typically maintain historical data. In this context, the data in operational databases, though rich, is usually far from complete for decision making. Decision support requiresconsolidation(suchasaggregationand summarization) of datafromheterogeneous sources, resulting in high-quality, clean, and integrated data. In contrast, operational databases contain only detailedraw 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 presently necessary to maintain separate databases.

- Different functions and different data:
  - Missing data: Decision support requires historical data which operational DBs do not typically maintain.
  - Dataconsolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources.
  - Data quality: Different sources typically use inconsistent data representations, codes and formats which have to be reconciled.

1.1.4. Data Warehousing: A Multi-Tiered Architecture:

Data warehousesoften adoptathree-tier architecture, aspresentedinfollowing Figure.

1. The bottom tier is a warehouse database server that is almost always a relational database system. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources(such as customerprofileinformationprovidedby external

consultants). These tools and utilities perform data extraction, cleaning, and transformation(e.g. tomergesimilardatafromdifferentsources into a unified format), as well as load and refresh functions to update the data warehouse. The data are extracted using application program interfaces 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. Examples of gateways include ODBC (OpenDatabaseConnection) andOLEDB(ObjectLinking andEmbedding, Database) by Microsoft and JDBC (Java Database Connection).

This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

2. The middle tier is an OLAP server that is typically implemented using either

(1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or

(2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

3. The top tier is a front-end client layer, which contains query and reportingtools, analysistools, and/ordata miningtools (e.g., trendanalysis, prediction, and so on).

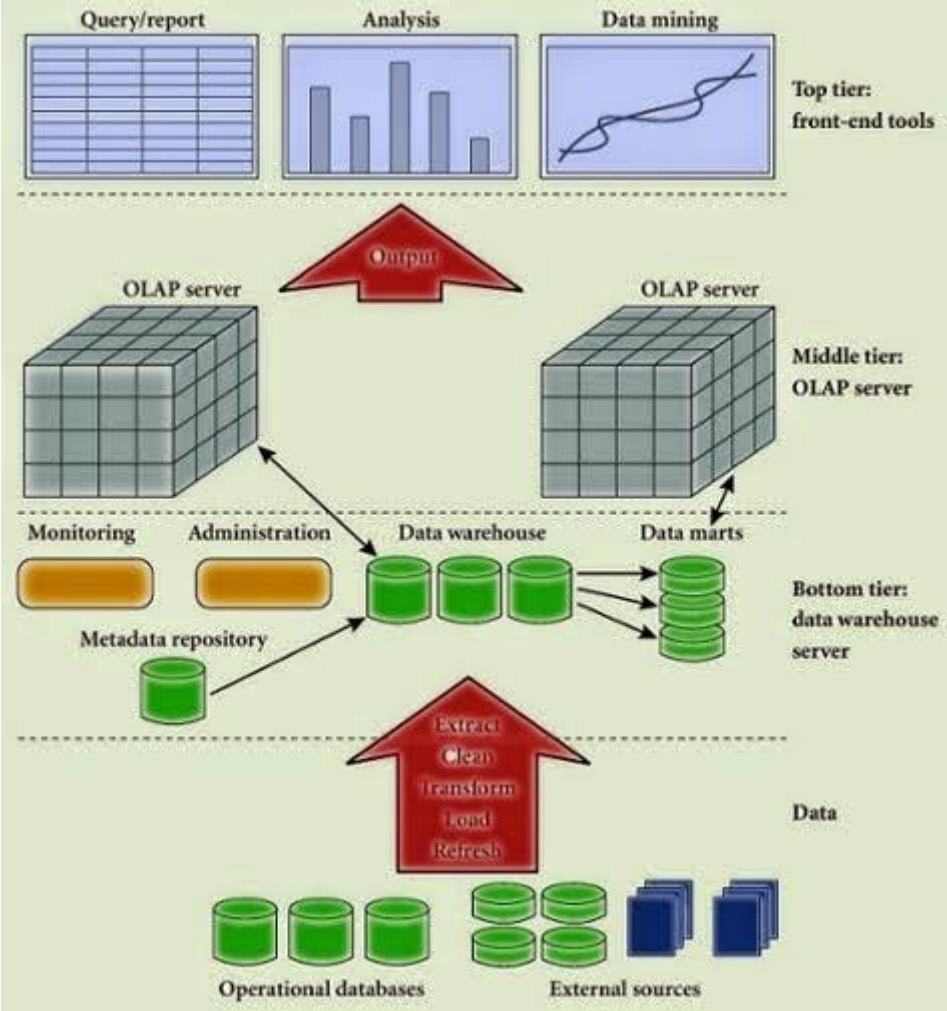


figure 1. 1

1.1.5 DataWarehouseModels: EnterpriseWarehouse,Data Mart, and Virtual Warehouse

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

Enterprise warehouse: An enterprise warehouse collects all of the information about subjects spanning the entire organization. It provides corporate-wide data integration, usually from one or more operational systems or externalinformationproviders, andis cross-functionalin scope.

It typically contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond. An enterprise data warehouse may be implemented on traditional mainframes, computer superservers, or parallel architecture platforms. It requires extensive business Modeling and may take years to design and build.

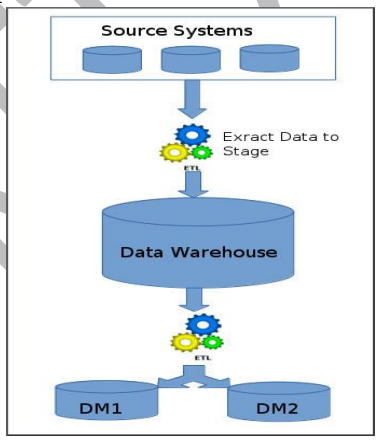
Data mart: A data mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales. The data contained in data marts tend to be summarized.

Data marts are usually implemented on low-cost departmental servers that are Unix/Linux-or Windows-based. The implementation cycle of a data mart is measured in weeks rather than months or years. However, it may involve complex integration in the long run if its design and planning were not enterprise-wide.

Depending on the source of data, data marts can be categorized as independent or dependent. Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. Dependent data marts are sourced directly from enterprise data warehouses.

Virtual warehouse: A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views maybe materialized. A virtual warehouseis easy to build but requires excess capacity on operational database servers.

"What are the pros and cons of the top-down and bottom-up approaches to data warehouse development?"

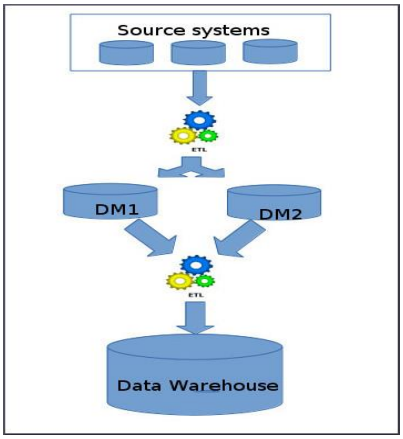


Top-down approach

The top-down development of an enterprise warehouse serves as a systematic solution and minimizes integration problems. However, it is expensive, takes a long time to develop, and lacks flexibility due to the difficulty in achieving consistency and consensus for a common data model for the entire organization.

Bottom-up approach

The bottom-up approach to the design, development, and deployment of independent data marts provides flexibility, low cost, and rapid return of investment. It, however, can lead to problems when integrating various disparate data marts into a consistent enterprise data warehouse.



A recommended method for the development of data warehouse systems is to implement the warehouse in an incremental and evolutionary manner, as shownin followingFigure.

First,ahigh-levelcorporatedatamodelisdefined withinareasonably shortperiod(such asone ortwomonths) thatprovidesa corporate-wide, consistent,integrated viewofdataamongdifferentsubjectsandpotential usages. This high-level model, although it will need to be refined in the further development of enterprise data warehouses and departmental data marts, will greatly reduce future integration problems.

Second, independent data marts can be implemented in parallel with the enterprise warehouse based on the same corporate data model set as above.

Third, distributed data marts can be constructed to integrate different data marts via hub servers.

Finally, **a multitier data warehouse** is constructed where the enterprise warehouse is the sole custodian of all warehouse data, which is then distributed to the various dependent data marts.

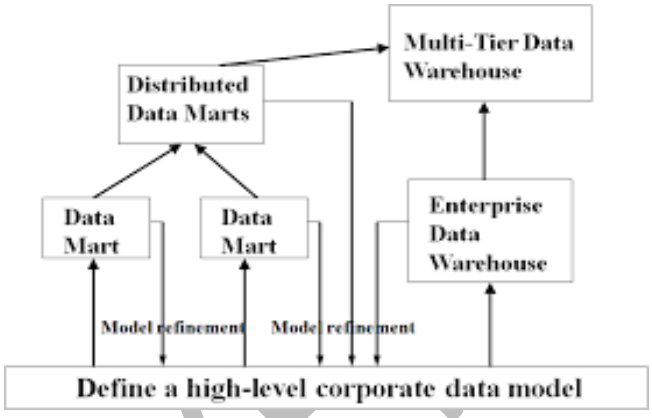


figure 1. 2

1.1.6Extraction, Transformation, Loading

Data warehouse systems useback-endtools and utilities topopulate and refresh theirdata(Figure Multi-tiered architecture). These tools and utilities include the following functions:

Data extraction which typically gathers data from multiple, heterogeneous, and external sources

Data cleaning which detects errors in the data and rectifies them when possible

Data transformation, which converts data from legacy or host format to warehouse format Load, which sorts, summarizes, consolidates, computes views, checks integrity, and builds indices and partitions

Load, which sorts, summarizes, consolidates, computes views, checks integrity, and builds indices and partitions

Refresh, which propagates the updates from the data sources to the warehouse

Besides cleaning, loading, refreshing, and metadata definition tools, data warehouse systems usually provide a good set of data warehouse management tools.

Data cleaning and data transformation are important steps in improving the quality of the data and, subsequently, of the data mining results. These two are used inDataPre-processing.

1.1.7 Metadata Repository

Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects.A metadata repository within the bottom tier of the data warehousing architecture. Metadata are created for the data names and definitions of the given warehouse. Additional metadata are created and captured for timestamping any extracted data, the source of the extracted data, and missing fields that have been added by data cleaning or integration processes.

A metadata repository should contain the following:

A description of the structure of the data warehouse, which includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents.

Operational metadata, which include data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived, or purged), and monitoring information (warehouse usage statistics, error reports, and audit trails)

The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports

The mapping from the operational environment to the data warehouse, which includes source data bases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control)

Data related to system performance, which include indices and profiles that improve data access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles

Business metadata, which include business terms and definitions, data ownership information, and chargingpolicies

Metadataplay a very different rolethan otherdata warehousedata and are important for many reasons. For example, metadata are used as a directory to help the decision support system analyst tolocate the contents of the data warehouse.

1.2 Data Warehouse Modeling: Data Cube and OLAP

Data warehouses and OLAP tools are based on a multidimensional data model. This model views data in the form of a data cube.

1.2.1 Data Cube: A Multidimensional Data Model

"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.**

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 at which the items were sold. Each dimension may have a table associated with it, called a **dimension table,** which describes the dimension. **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.

Amulti-dimensionaldata modelistypically organizedaround a centraltheme,like sales, forinstance. Thisthemeis representedby a**facttable.** Facts are numerical measures. These are thequantitiesby which wewanttoanalyze relationships between dimensions.

Examples of facts for a sales data warehouse include dollars_sold (sales amountindollars),units_sold (number ofunits sold), and amount_budgeted. The fact table contains the names of the facts, or measures, as well askeysto each of the relateddimensiontables.

From Tables and Spreadsheets to Data Cubes→

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 following Table .

Table 2.1 A 2-D view of sales data according to the dimension *time* and *item*, where the sales are from braches located in Vancouver. The measure shown is *dollar_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

In this 2-D representation, the sales for Vancouver are shown with respect to the time dimension (organizedinquarters) and the item dimension (organizedaccording to the types of items sold). The fact or 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 below.

Table 2.2 A 3-D view of sales data according to the dimension *time* and *item* and *location*. The measure shown is *dollar_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

The 3-D data in Table below 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 above Figure.

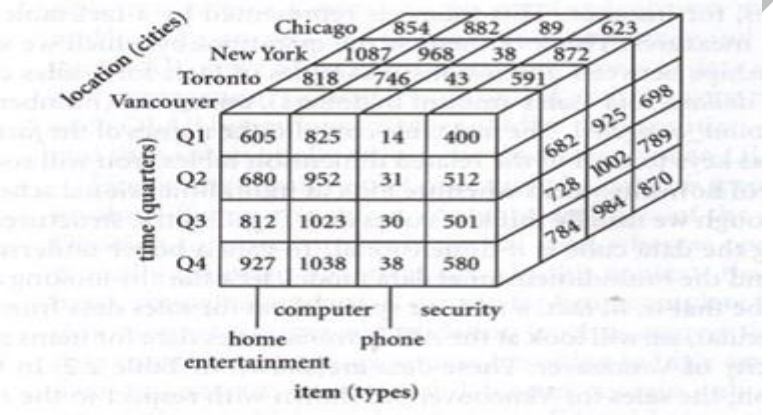


figure 1. 3

Figure : A 3-Ddata cube representation of the AllElectronicssalescdata, accordingto 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 fourthdimension, such as supplier. So, we can think of a 4-D cube as being a series of 3-D cubes, as shown in Figure below.

If we continue in this way, we may display any n-D data as a series of(n-1)-D "cubes."

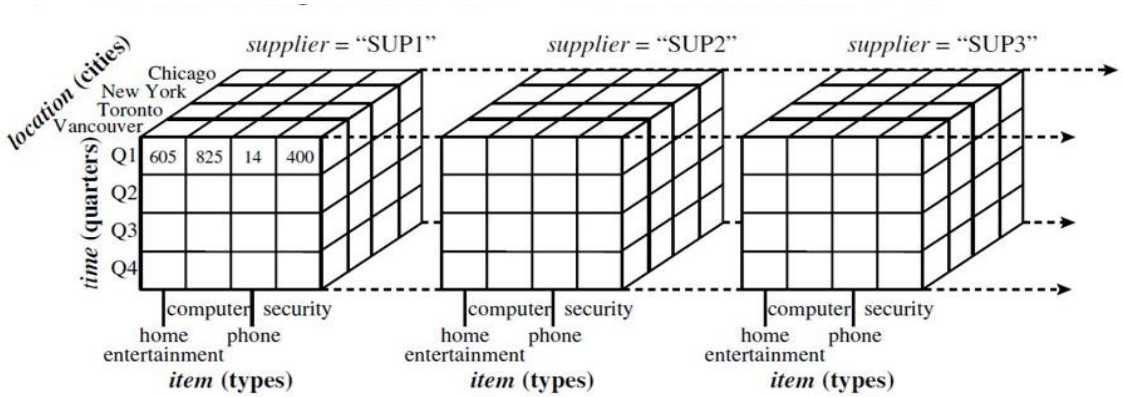


Figure: A 4-D data cube representation of sales data, according to the dimensions *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars sold* (in thousands). For improved readability, only some of the cube values are shown.

figure 1. 4

The data cube is a metaphor for multidimensional data storage. In the data warehousing research literature, a data cube such as each of the above is often referred to as a **cuboid.**

Given a set of dimensions, we can generatea lattice of cuboids for each of the possible subsets of the given dimensions,each showing the data at a different level of summarization, **or group by**. The lattice of cuboids forms a **data cube**. Figure below shows a lattice of cuboids forming a data cube for the dimensions time, item, location, and supplier.

- In data warehousing literature, an n-D base cube is called a base cuboid. The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid

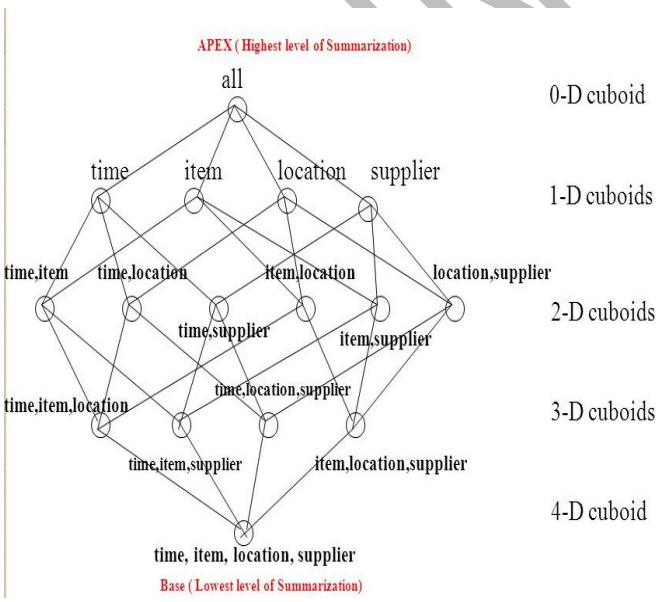


figure 1. 5

The cuboid that holds the lowest level of summarization is called the **base cuboid**. For example, the 4-D cuboid in the above figure1.4 is the base cuboid for the given time, item, location, and supplier dimensions.

Figure 1.3 is a 3-D (nonbase)cuboidfor time, item, and location, summarizedforall 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, or dollars sold, summarized over all four dimensions. The apex cuboid is typically denoted by all.

1.2.2 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 on-line transaction processing. A data warehouse, however, requires a concise, subject-oriented schema that facilitates online data analysis.

The mostpopulardatamodelfor adata warehouseis a multidimensional model. Such a model can exist in the form of a star schema,a snowflake schema, or a fact constellation schema. Let's look at each of these schema types.

Star schema: The most common modeling paradigm is the star schema, in which the data warehouse contains(1) alarge centraltable (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.

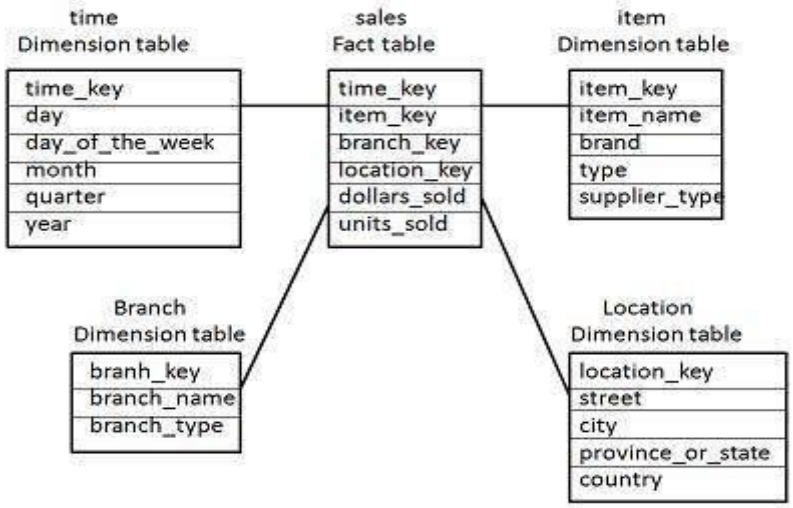


figure 1. 6

The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

In the star schema, eachdimensionis representedby only one table, andeachtable contains a set of attributes**. For example**, thelocation 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 either ahierarchy(total order) or alattice(partial order).

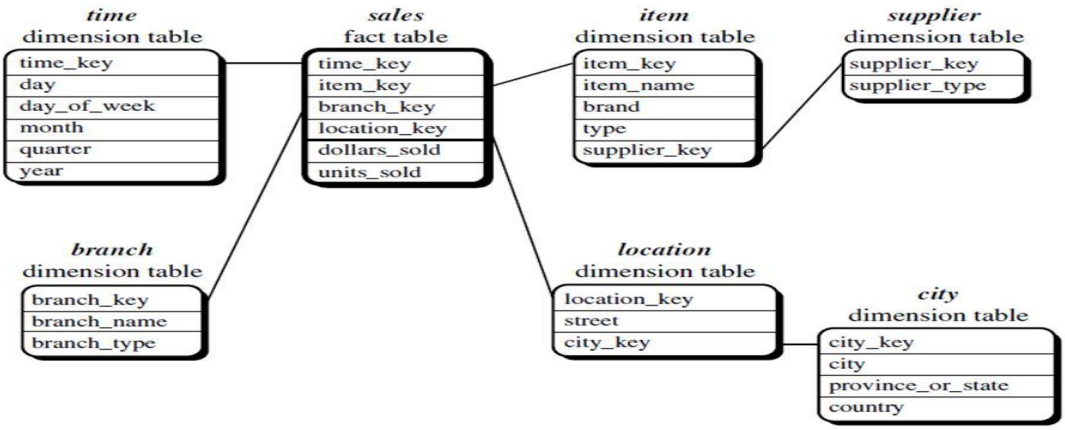


Figure: Snowflake schema of a data warehouse for sales

figure 1. 7

Snowflake schema: The snowflake schema is a variant of the star schema model, where somedimension tables are normalized, therebyfurther splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.

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 saving of space is negligible in comparison to the typical magnitude of the fact table.

The snowflake structure can reduce the effectiveness ofbrowsing, since morejoins willbe needed to execute a query. Consequently, the system performance may be adversely impacted. Hence, although the snowflake schema reduces redundancy, it is not as popular as the star schema in data warehouse design.

Snow flake schema	Star schema
No redundancy	*redundant*
More complex queries	*Less complex queries*
Lots of foreign keys, hence more execution time	*Quick execution*
Lots of joins	*Fewer joins*
More number of dimensions for single dimension	*Only one dimension*
Normalized	*denormalized*

Example: A snowflake schema for AllElectronics sales is given in Figure. Here, the sales fact table is identical to that of the star schema Figure.

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, containingsupplier 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.

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.**

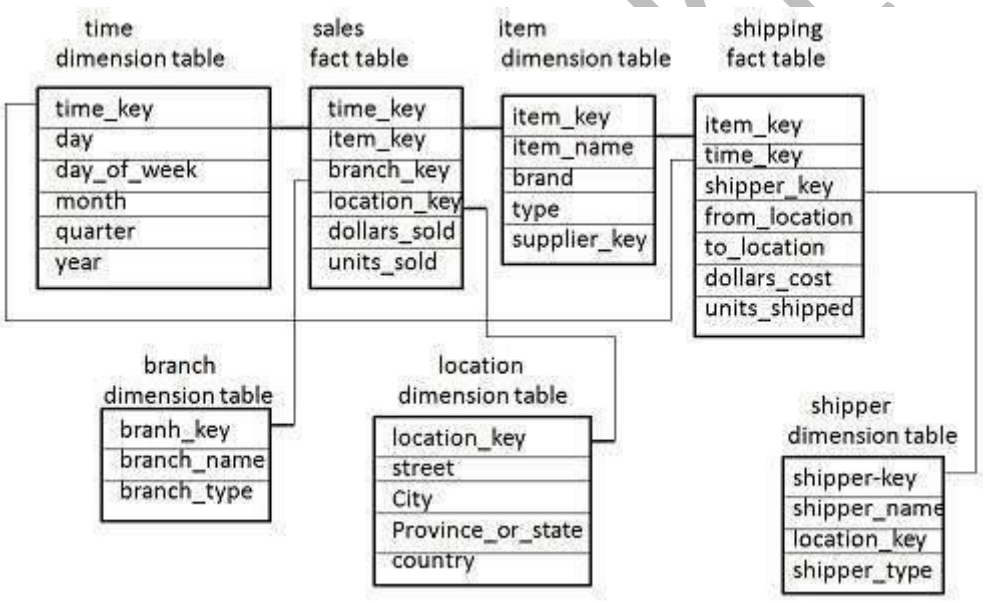


figure 1. 8

Example: Fact constellation: A fact constellation schema is shown in above Figure 1.8. This schema specifies two fact tables, sales and shipping. The sales table definition is identical to that of the star schema (Figure 1.6). 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 both the sales and shipping fact tables.

In data warehousing, there is a distinction between a data warehouse and a data mart. A data warehouse collects information about subjects that span the entire organization, such as customers, items, sales, assets, and personnel, and thus its scope is enterprise-wide. **For data warehouses, the fact constellation schema is commonly used,** since it can model multiple, interrelated subjects.

A data mart, on the other hand, is a department subset of the data warehouse thatfocuses on selected subjects, andthusits scopeis department-wide. **Fordata marts, the star or snowflake schema are commonly used**, although the starschemais morepopular and efficient.

1.2.3 Dimensions: The Role of Concept Hierarchies

Aconcepthierarchy **defines a sequence ofmappingsfrom a set oflow-levelconceptstohigher-level, moregeneral concepts.**

Consider a concepthierarchyforthe dimensionlocation. Cityvaluesfor locationincludeVancouver,Toronto,NewYork,andChicago.Eachcity,however,canbemappedtothepr ovinceorstatetowhichitbelongs.

Forexample,VancouvercanbemappedtoBritishColumbia,andChicago toIllinois. Theprovincesandstatescaninturnbemappedtothecountrytowhich they belong, such as Canada or the USA. These mappings form a concept hierarchyforthedimension location, mapping a set oflow-level concepts(i.e., cities)tohigher-level,moregeneral concepts (i.e., countries). The concepthierarchy described aboveisillustratedinFigure 1.9.

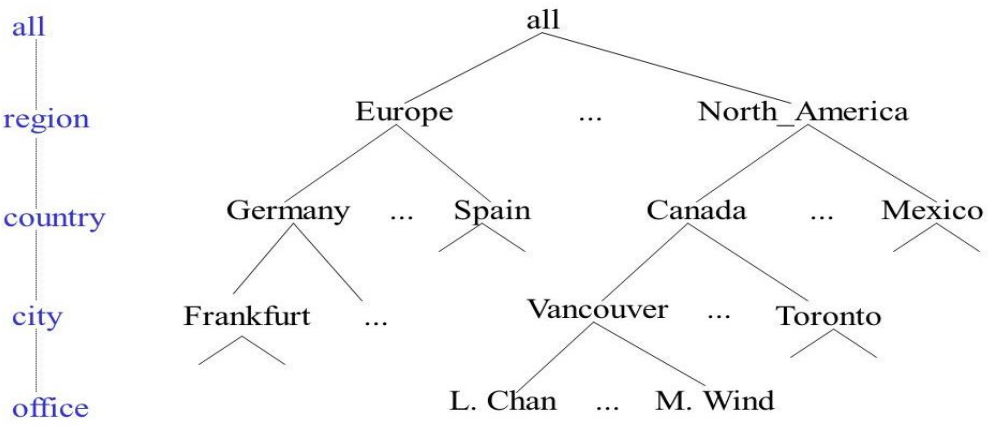


figure 1. 9

Many concept hierarchiesareimplicit withinthedatabase schema.

For example, suppose that the dimension location is described by the attributes number, street,city,province or state,zipcode,andcountry. Theseattributesarerelatedbyatotalorder,formingaconcepthierarchysuch as“street < city <province or state <country”. This hierarchy is shown in Figure 1.10(a). Alternatively, the attributes of adimension maybe organizedin apartial order,forming alattice. Anexample of a partial order for the time dimension based on the attributes day, week, month, quarter, and year is “day < {month <quarter; week}<year”.

This lattice structure is shown in Figure 1.10(b). A concept hierarchy that is atotal orpartial order among attributesin adatabase schemais called a schema.

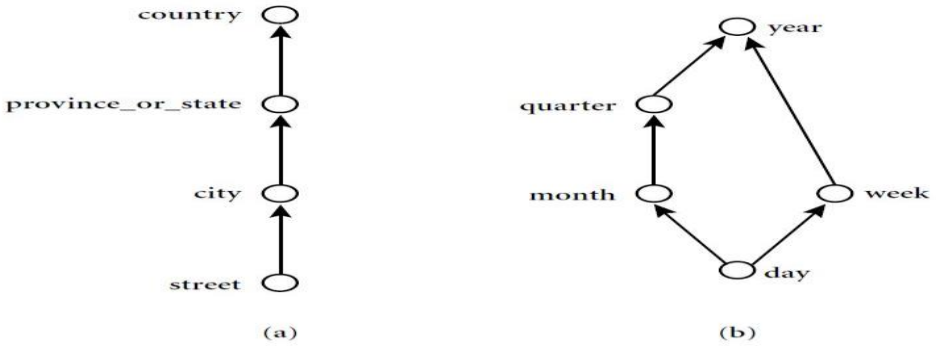


Figure: Hierarchical and lattice structures of attributes in warehouse dimensions: (a) a hierarchy for *location*; (b) a lattice for *time*.

figure 1. 10

hierarchy:

Concept hierarchies that are common to many applications may be predefined in the data mining system, such as the concept hierarchy for time. Data mining systems should provide users with the flexibility to tailor predefined hierarchies according to their particular needs.

For example, users may like to define a fiscal year starting on April 1 or an academic year starting on September 1.

Concept hierarchies may also be defined by 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. **An example** of a set-grouping hierarchy is shown in Figure 4.11 for the dimension price, where aninterval($X...$Y]denotes the range from $X(exclusive)to $Y (inclusive).

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.

Concept hierarchies may be provided manually by system users, domain experts, or knowledge engineers, or may be automatically generated based on statistical analysis ofthedatadistribution. Conceptthierarchies allowdata tobehandledat varyinglevels of abstraction.

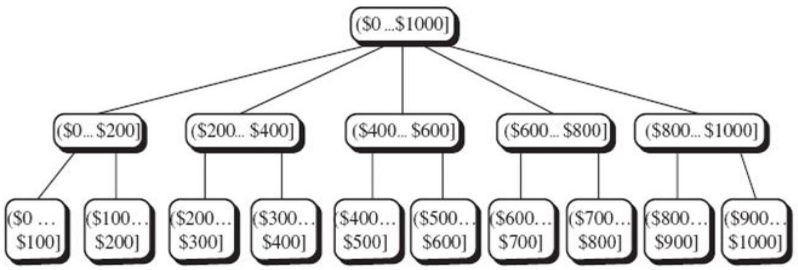


figure 1. 11

1.2.4 Measures: Their Categorization and Computation

"How are measures computed?" To answer this question, we first study how measures can be categorized. Note that a multidimensional point in the data cube space canbedefinedby a set ofdimension-valuepairs,for example, htime= "Q1", location = "Vancouver", item = "computer". A data cube measure is a numerical function that can be evaluated at each point in the data cube space. A measure value is computed for a given point by aggregating the data corresponding to the respective dimension-value pairs defining the given point.

Measurescanbe organizedintothreecategories(i.e.,distributive, algebraic,holistic), based on the kind of aggregate functions used.

Distributive: An aggregate function is distributive if it can be computed in a distributed manner as follows. Suppose the data are partitioned into nsets. We apply the function to each partition, resulting in n aggregate values. If the result derived by applying the function to the n aggregate values is the same as that derived by applying the function to the entire data set (without partitioning), the function can be computed in a distributed manner. For example, sum() can be computed for a data cube by firstpartitioning thecubeintoasetof subcubes,computing sum() for each subcube, and then summing up the counts obtained for each sub-cube. Hence, sum() is a distributive aggregate function. For the same reason, count(), min(), and max() are distributive aggregatefunctions. By treating the count value of each nonempty base cell as 1 by default, count() of any cell in a cube can be viewed as the sum of the count values of all of its corresponding child cells in its subcube. Thus, count()is distributive. A measure is distributive if it is obtained by applying a distributive aggregate function. Distributive measures can be computed efficiently because of the way the computation can be partitioned.

Algebraic: An aggregatefunctionisalgebraic ifitcanbe computedby an algebraicfunction withMarguments(where M is aboundedpositiveinteger), each of which is obtained by applying a distributive aggregate function.

For example, avg() (average) can be computed by sum()/count(), where both sum() and count() are distributive aggregate functions. Similarly, it can be shown that minN() and max N() (which findthe N minimum and Nmaximum values, respectively, in a given set) and standard deviation() are algebraic aggregate functions. A measure is algebraic if it is obtained by applying an algebraic aggregate function.

Holistic: An aggregate function is holistic if there is no constant bound onthe storage size needed todescribe a subaggregate. Thatis, theredoes not exist an algebraicfunction with M arguments(where M is a constant)that characterizes the computation. Common examples of holistic functions includemedian(), mode(), and rank(). Ameasureis holistic ifitis obtained by applying a holistic aggregate function.

1.2.5 Typical OLAP Operations

"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 ofOLAPdata cube operations exist to materialize

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

Example 1.4 OLAP operations. Let's look at some typical OLAP operations for multidimensional data. Each of the operations described below is illustrated in Figure 1.12. At the center of the figure is a data cube for AllElectronics sales.

The cube contains thedimensions location, time, and item, where location is aggregatedwith respect to city values, time is aggregatedwith respect toquarters,and item is aggregated with respect to item types. The measure displayed is dollars sold (in thousands). 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 1.12 shows the result of a roll-up operation performed on the central cube byclimbing up the concept hierarchy for location given in Figure 1.9. This hierarchy wasdefined asthetotal 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.

Drill-down: Drill-downis the reverse of roll-up. It navigatesfromlessdetailed data to more detailed data. Drill-down can be realized by either stepping down a concept hierarchy for adimension orintroducing additional dimensions. Figure 1.12 shows the result of a drill-down operation performed on the central cube by steppingdown a conceptthierarchyfor 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 performedbyadding newdimensions to a cube. For example, adrill-down on the central cube of Figure 1.12 can occur by introducing an additional dimension, such as customer group.

Slice and dice: The slice operation performs a selection on one dimension of thegivencube,resultinginasubcube. Figure1.12shows a sliceoperation 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 1.12 shows adice operation on the central cubebased on thefollowing 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 calledrotate)is a visualization operation that rotatesthedataaxesinviewin ordertoprovideanalternativepresentation of thedata. Figure 1.12 showsapivot operationwherethe item and location axes in a 2-D slice are rotated. Other examples include rotating the axesin a3-D cube, ortransforming a3-D cubeintoaseriesof2-Dplanes.

Other OLAP operations: Some OLAP systems offer additional drilling operations. For example, drill-acrossexecutesqueriesinvolving (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 itsback-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.

OLAP offers analytical Modeling capabilities, including a calculation engine forderiving ratios, variance, and so on, andfor computing measures across multiple dimensions. It cangenerate summarizations, aggregations, andhierarchies at each granularity level and at every dimension intersection. OLAP also supports functional models for forecasting, trend analysis, and statistical analysis.

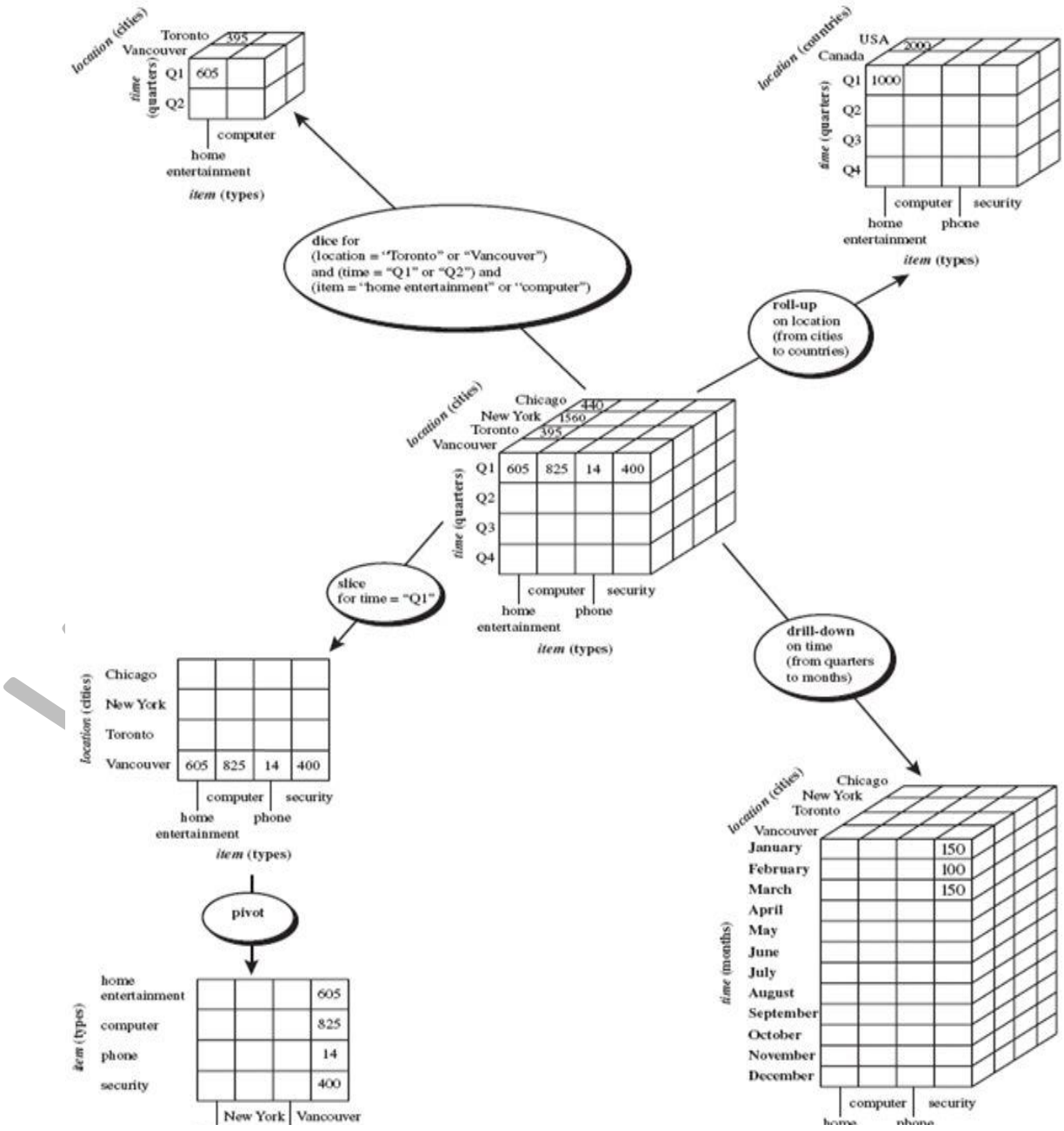


Figure 1.12

Question Bank

1. Define Data Warehouse .Explain the key features of Data Warehouse.
2. Differences between operational database and data warehouse.
3. Explain the three tier architecture of data warehouse.
4. Explain the different warehouse models.
5. Explain the tools and utilities used in the data warehouse architecture.
5. Define metadata explain the contents of metadata repository.
6. Define data cube. Explain with examples a multidimensional data cube.
7. Explain Star schema, snowflake schema and fact constellation with examples.
8. Explain the concept of hierarchies in multidimensional data.
9. Explain three different categories of measures.
10. Explain typical OLAP operations on multidimensional data.