Data Mining & Data Warehousing

18CS641

Pavan Kumar SP
VVCE MYSURU

Module 1 - Data Warehousing & modeling

1. Data Warehousing: Basic Concepts
   1.1. What Is a Data Warehouse?
   1.2. Differences between Operational Database Systems and Data Warehouses
   1.3. But, Why Have a Separate Data Warehouse?
   1.4. Data Warehousing: A Multitiered Architecture
   1.5. Data Warehouse Models: Enterprise Warehouse, Data Mart, and Virtual Warehouse
   1.6. Extraction, Transformation, and loading
2. Data Warehouse Modelling: Data Cube and OLAP
   2.1. Data Cube: A multidimensional data model
   2.2. Stars, Snowflakes, and Fact constellations: Schemas for multidimensional Data models
   2.3. Dimensions: The role of concept Hierarchies
   2.4. Measures: Their Categorization and computation
   2.5. Typical OLAP Operations

What is a Data Warehouse?

Definition: "Data Warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process" - ***William H. Inmon***

- It provides architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions**.**
- It is a data repository maintained separately from an organization's operational databases.
- It allows for the integration of a variety of application systems.
- It supports information processing by providing a solid platform of consolidated historical data for analysis
- It is also often viewed as an architecture constructed by integrating data from multiple heterogeneous sources to support structured and ad hoc queries, analytical reporting, and decision making
- Data warehouses can store and integrate historical information and support complex multidimensional queries.

key features of Data Warehouse are:

a. Subject oriented
b. Integrated
c. Time -variant
d. Non-volatile

Subject oriented

- Data warehouse is organized around significant ***subjects*** such as customer, supplier, product, and sales rather than day to day transaction

- It only focuses on modeling and analysis of data for decision-makers ***(subjects***)
- It provides a concise view of a particular ***subject*** issue, which is helpful for the decision-making process

Integrated

- A data warehouse is usually constructed by ***integrating*** multiple heterogeneous such as relational databases, flat files, and online transaction records.
- Data cleaning and ***integration*** techniques ensure consistency in naming conventions, encoding structures, and attribute measures.

Time-Variant

- Data is stored to provide information from a historical perspective (e.g., the past 5–10 years).
- Every key structure in the data warehouse contains a time element, either implicitly or explicitly.

Non-volatile:

- 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.
- It usually requires only two operations in data accessing: initial loading and access of data.

The construction of a data warehouse requires (Steps)

a. Data cleaning
b. Data integration
c. Data consolidation

How are organizations using the information from data warehouses?

Organizations use the data warehouse information for **decision-making activities** as follows.

1. **Increasing customer focus** – Analysis of customer buying patterns
2. **Repositioning of products / Managing product portfolios** – Comparing the sales by quarter/Year, by locations to fine-tune the production strategies.
3. Analysing operations and looking for sources of profit
4. Managing customer relationships, making environmental corrections, and managing the cost of corporate assets.

Heterogeneous database integration

- Organizations typically collects diverse kind of data,
- It maintains an extensive database from multiple, heterogeneous, autonomous, distributed information sources.
- The collected data is desirable and challenging to integrate.

Query-driven approach- Traditional approach to handle the heterogeneous database integration

The traditional database approach to heterogeneous database integration is to build ***wrappers*** and ***integrators*** (or mediators) on top of multiple, heterogeneous databases.

When a query is posed to a client site

1. **Metadata dictionary** translates the query into queries appropriate for the individual heterogeneous sites involved.
2. Queries are then mapped and sent to local **query processors**
3. The results from the different sites are integrated into a global answer set.

The drawback of the traditional approach

- This **query-driven approach** requires complex information filtering and integration processes and competes with local sites for processing resources.
- It is **inefficient** and **potentially expensive** for frequent queries, especially queries requiring aggregations**.**

Update driven approach

- Information from multiple, heterogeneous sources is integrated in advance and stored in a warehouse for direct querying and analysis.
- Data are copied, pre-processed, integrated, annotated, summarized, and restructured into one semantic datastore. Unlike online transaction processing databases, data warehouses do not contain the most current information.
- Query processing in data warehouses does not interfere with the processing at local sources.

Differences between Operational Database Systems and Data Warehouses

	Operational Database System	Data warehouse
Major task	is to perform an online transaction and query processing	Serve users or knowledge workers in the role of data analysis and decision making.
Called as	Online transaction processing (OLTP) systems	Online analytical processing (OLAP) systems
System orientation	OLTP system is **customer-oriented**	OLAP system is **market-oriented**

Used for	Transaction and query processing	Data analysis
Users	clerks, clients	knowledge workers, managers, executives, and analysts
Data contents	It manages current data that, typically, are too detailed to be easily used for decision making.	It manages large amounts of historical data and provides facilities for summarization and aggregation.
Database design	Usually adopts an entity-relationship (ER) data model and an application-oriented database design	Usually adopts a star or a snowflake model and a subject-oriented database design
View	• It focuses mainly on the current data within an enterprise or department	• It will focus on historical data • It deals with information from different organizations, integrating information from many data stores.
Access patterns	• Consist mainly of short, atomic transactions. • Such a system requires concurrency control and recovery mechanisms	• OLAP systems are mostly read-only operations

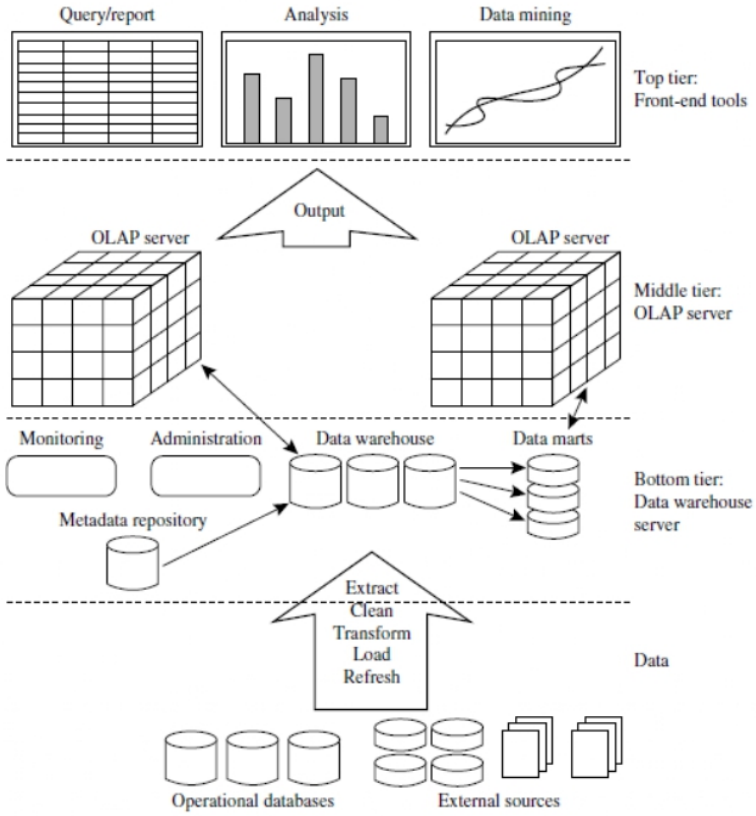
Table 4.1 Comparison of OLTP and OLAP Systems

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

Why Have a Separate Data Warehouse?

- This separation is to help promote the high performance of both systems
- Operational database is designed to have workloads like indexing and hashing using primary keys, searching for records, and optimizing "canned" queries
- Data warehouse queries are often complex and involve the computation of large data groups at summarized levels.
- Processing OLAP queries in operational databases would substantially degrade the performance of operational tasks.
- Operational database supports the concurrent processing of multiple transactions.
- Concurrency control and recovery mechanisms are required to ensure the consistency and robustness of transactions.
- An OLAP query often needs read-only access to data records for summarization and aggregation.
- Concurrency control and recovery mechanisms may reduce the throughput of an OLTP system if applied for such OLAP operations.
- Operational databases do not typically maintain historical data
- Decision support requires consolidation (e.g., aggregation and summarization) of data from heterogeneous sources, resulting in high-quality, clean, integrated data.
- In contrast, operational databases contain only detailed raw data, such as transactions, which need to be consolidated before analysis

Data Warehousing: A Multitiered Architecture



Bottom tier

- Back-end tools and utilities feed data from operational databases or other external sources into the bottom tier.
- These tools and utilities perform **data extraction, cleaning, transformation, and** 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.
- This tier also contains a **metadata repository**, which stores information about the data warehouse and its contents.

Middle tier

- is an **OLAP server** typically implemented using either a **relational OLAP(ROLAP)** or multidimensional OLAP (MOLAP) models.
- **ROLAP-** an extended relational DBMS that maps operations on multidimensional data to standard relational operations.
- **MOLAP** -special-purpose server that directly implements multidimensional data and operations.

Top tier

- It is a front-end client layer that contains query and reporting tools, analysis tools, and data mining tools

Data Warehouse Models

From the architectural point of view, there are three Data Warehouse Models

a. Enterprise Warehouse
b. Data mart
c. Virtual warehouse

Enterprise Warehouse

- it collects all information about subjects spanning the entire organization
- it provides corporate-wide integration (from one are more operational system/external information providers)
- it contains both detailed and summarized data
- size of the data varies from hundreds of gigabytes (GB) to terabytes (TB)
- it can be implemented in traditional mainframe computers/ computer Super servers /Parallel architecture platforms.
- it requires extensive business modeling
- implementation cycle is measured in years

Datamart

- it contains a subset of corporate-wide data
- the scope is confined to some specific selected subjects
- low-cost departmental servers usually implement its
- implementation cycle is measured in weeks
- if it's design and planning are not enterprise-wide, in the long run, we should involve a complex integration process
- Depending on data sources, the data mart is categorized into dependent and independent.

- If the data is sourced from one or more operational system /external information providers, we can call the data mart an independent data mart
- If the data is sourced directly from the enterprise data warehouse, we can use the data mart as a dependent data mart.

Virtual warehouse

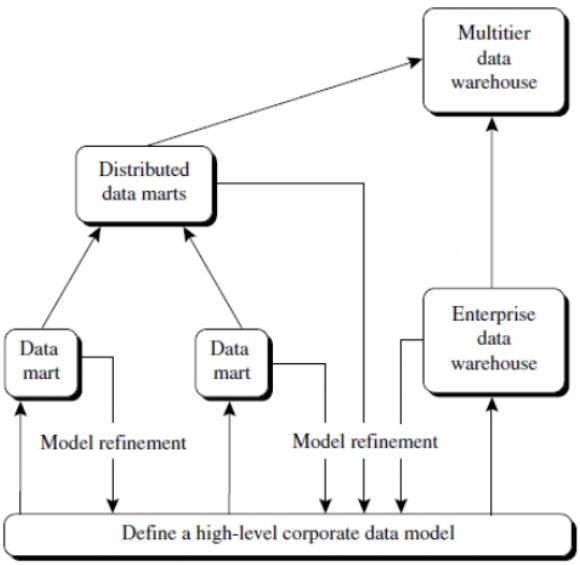
- It is a set of views over an operational database
- For efficient query processing, only some of the possible summary views may be materialized
- It is easy to build, but it requires excess capacity on the operational database server

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

	Top-down approach	Bottom-up approach
Pros	• It is a systematic solution • it minimizes the integration problem	• Design and development of independent data marts provide flexibility • Low cost • Rapid return on investment
Cons	• it is expensive • it takes a long time to develop • lack of flexibility	• it may lead to problems while integrating various disparate data marts into the consistent data warehouse

The recommended method for the development of data warehouse systems

- Implement the warehouse incrementally and evolutionarily, as shown in the below figure.



Step1

- Define a high-level corporate data model within a short period
- This data model provides a corporate-wide, integrated, and consistent view of subjects and potential usages.
- We need to refine this model further to reduce future integration problems.

Step2

- Independent data marts can be implemented in parallel with the enterprise warehouse based on the same corporate data model set noted before

Step 3

- Distributed data marts can be constructed to integrate different data marts via hub servers

Step 4

- 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

Extraction, Transformation, and Loading

- Data warehouses use back-end tools and utilities to extract and refresh their data.
- The flowing are the functionalities of tools and utilities
  1. **Data Extraction** – gathers data from external and multiple resources
  2. **Dara Cleaning** -it detects errors and noise in the data and rectifies them when possible
  3. **Data Transformation** – converts data from host format to warehouse format
  4. **Load** -sorts, summarizes, consolidates, computes views, checks the integrity, and builds indices and partitions
  5. **Refresh** – which propagates the updates from the data source to the warehouse

Metadata Repository

- Metadata is data about data
- It defines warehouse objects /data names /definition
- Additional metadata are created and captured
  a. For time stamping any extracted data
  b. Source of the extracted data
  c. Missing fields that have been added by the data clean and integration process
- Metadata is used as a directory to help the decision support system analyst locate the contents of the data warehouse
- It is used to guide the data mapping when data are transformed from the operational environment to the data warehouse environment.

- Metadata also guides the algorithms used to summarize the current detailed data and the lightly summarized data, and between the lightly summarized data and the highly summarized data.
- Metadata should be stored and managed persistently

A metadata repository should contain the following

1	Description of Datawarehouse structure	Which includes a. Warehouse schema view b. Dimensions c. Hierarchies d. Derived data definitions e. Data marts location and contents
2	Operational Metadata	This includes a. Data lineage- data migration and transformation b. Currency of data-active, archived -purged c. Monitoring Information -statistics, error reports, audit trails
3	The algorithm used for summarization	Which includes a. Measure and dimension definition algorithms b. Data on granularity c. Partitions d. Subject areas, e. Aggregation f. Predefined quarries and reports
4	Mapping from the operational environment to the data warehouse	which includes a. Source databases and their contents b. gateway descriptions c. data partitions/extraction/cleaning/ transformation rules d. data refresh and purging rules, e. Security (user authorization and access control).
5	Data related to system performance	which include a. indices and profiles that improve data access and retrieval performance b. Rules for the timing and scheduling of refresh, update, and replication cycles
6	Business metadata	which include a. Business terms and definitions b. data ownership information, c. charging policies

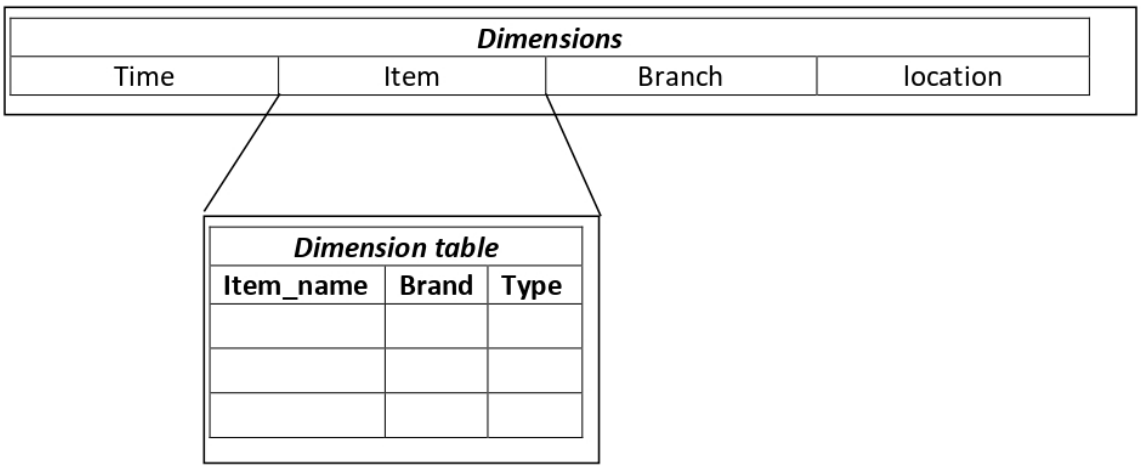
Data Cube: A multi-Dimensional Data model

- A-Data cube allows data to be **modeled** and **viewed** in multiple dimensions.
- It is defined by **dimensions** and **facts**
- **Dimensions** – are the perspectives/entities concerning which an organization wants to keep the records

Example

Organization: AllElectronics
Warehouse: Sales data Warehouse
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,** as shown in the below figure



- A multidimensional data model is typically organized around a central theme, such as **sales**
- A fact table represents this theme**.**
- **Facts** are numeric measures.

Example

Organization: AllElectronics
Warehouse: Sales data Warehouse
Dimensions: time, item, branch, and location
Theme: Sales
Facts: Dollor_sold and Unit_sold

- The fact table contains the **names of the facts** measures and **keys** to each of the related dimension tables.

Example for Two-Dimensional Data Model

Dimensions: Item and Time
Facts: Home Entertainment, Computer, Phone, Security, Q1 to Q4
Measure:

605 825 14 400
680 952 31 512
812 1023 30 501
927 1038 38 580

	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

- We will look at the AllElectronics sales data for items sold per quarter in the city of Vancouver.
- In this 2-D representation, the sales for **Vancouver** are shown concerning the **time dimension** (organized in quarters) and the **item dimension** (classified to the types of items sold).
- The fact or measure displayed is **dollars sold** (in thousands).

Example for Three-Dimensional Data Model

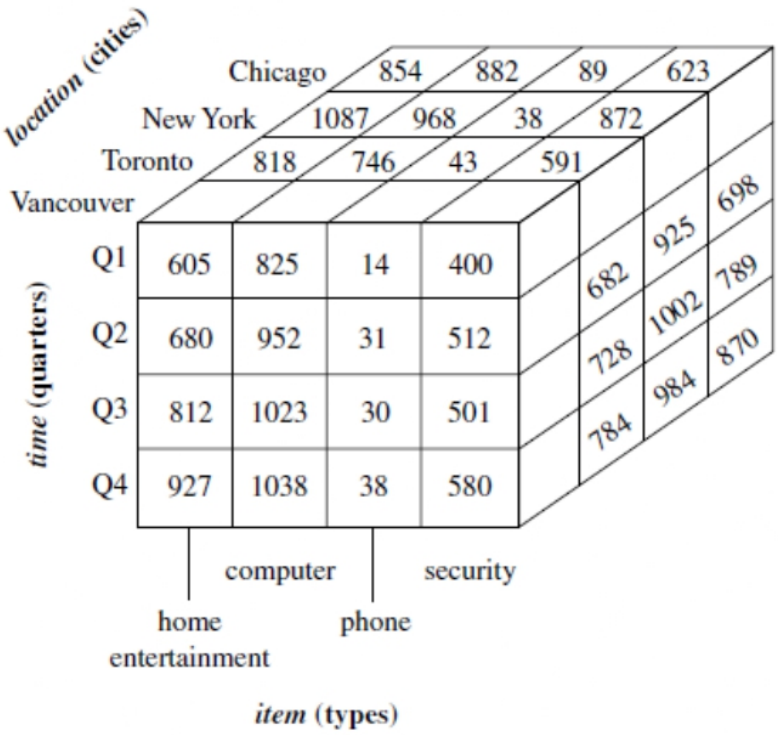
- Suppose we would like to view the data according to time and item, and location for the cities Chicago, New York, Toronto, and Vancouver.

Dimensions: Item and Time, Location
Facts: Home Entertainment, Computer, Phone, Security, Q1 to Q4, the cities Chicago, New York, Toronto, and Vancouver

- The 3-D data in the table are represented as a series of 2-D tables

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

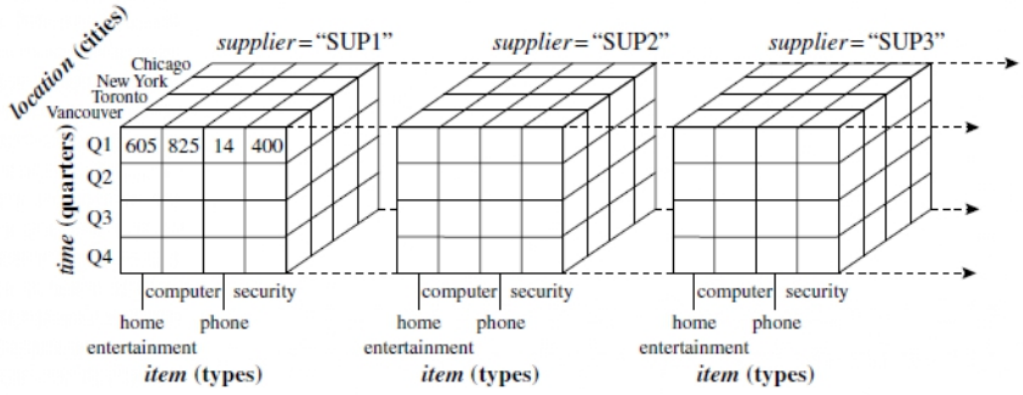
Conceptually, we may also represent the same data in a 3-D data cube, as shown below.



Example for Four-Dimensional Data Model

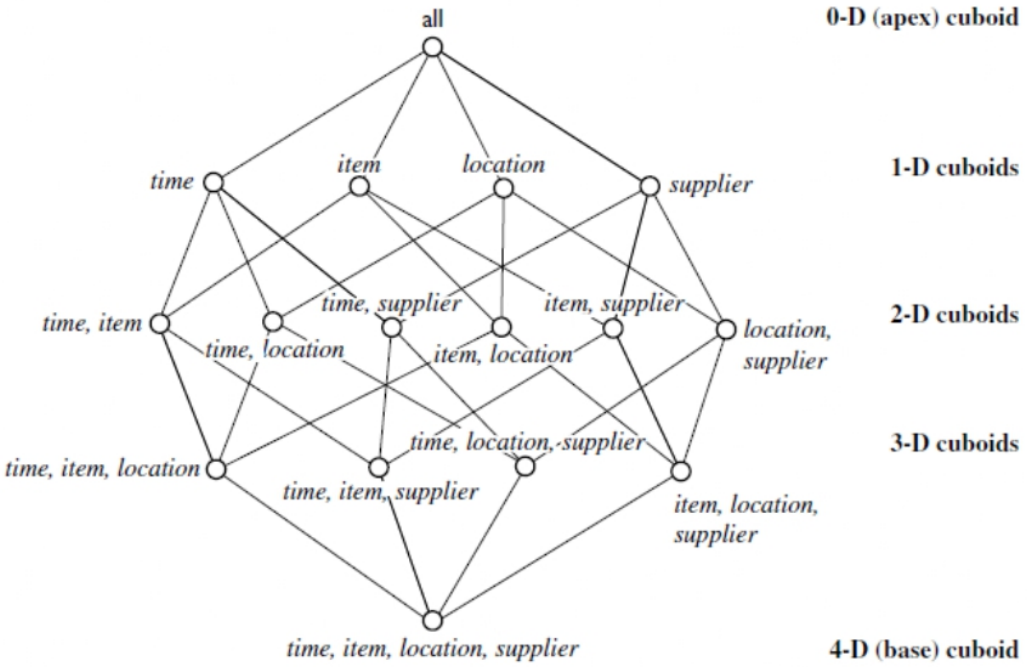
- Suppose that we would now like to view our sales data with an additional fourth dimension, such as a supplier.
- Viewing things in 4-D becomes tricky. However, we can think of a 4-D cube as being a series of 3-D cubes, as shown in the below figure

Dimensions: Item and Time, Location, Supplier
Facts: Home Entertainment, Computer, Phone, Security, Q1 to Q4, the cities Chicago, New York, Toronto, and Vancouver, Sup1, Sup2, Sup3



- We can have different degrees of summarization with different data dimensions

- 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
- The cuboid that holds the lowest level of summarization is called the base cuboid
- The 0-D cuboid, which holds the highest level of summarization, is called the apex cuboid.



Schemas for Multidimensional Data Models

- A data warehouse requires a concise **subject-oriented schema** that facilitates **online data analysis**
- The most popular schemas of the data warehouses are
  1. Star schema
  2. Snowflake schema
  3. Fact constellation schema

Star Schema

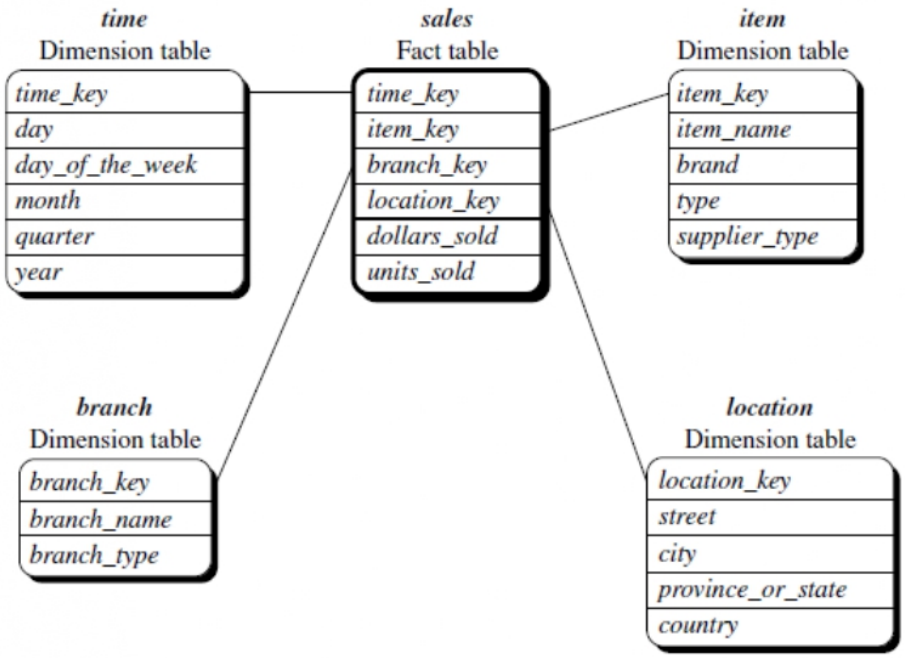
In a star schema, the data warehouse contains

1. **Fact Table** - a large central table containing the bulk of the data, with no redundancy
2. **Dimension tables**- a set of smaller attendant 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

Data warehouse: AllElectronics sales
Dimensions: time, item, branch, and location
Central fact table for sales: contains keys to each of the four dimensions along with two measures: dollars sold and units sold
Advantages



- Simplest and Easiest
- Optimises the navigation through the database
- Most suitable for query processing

Disadvantage

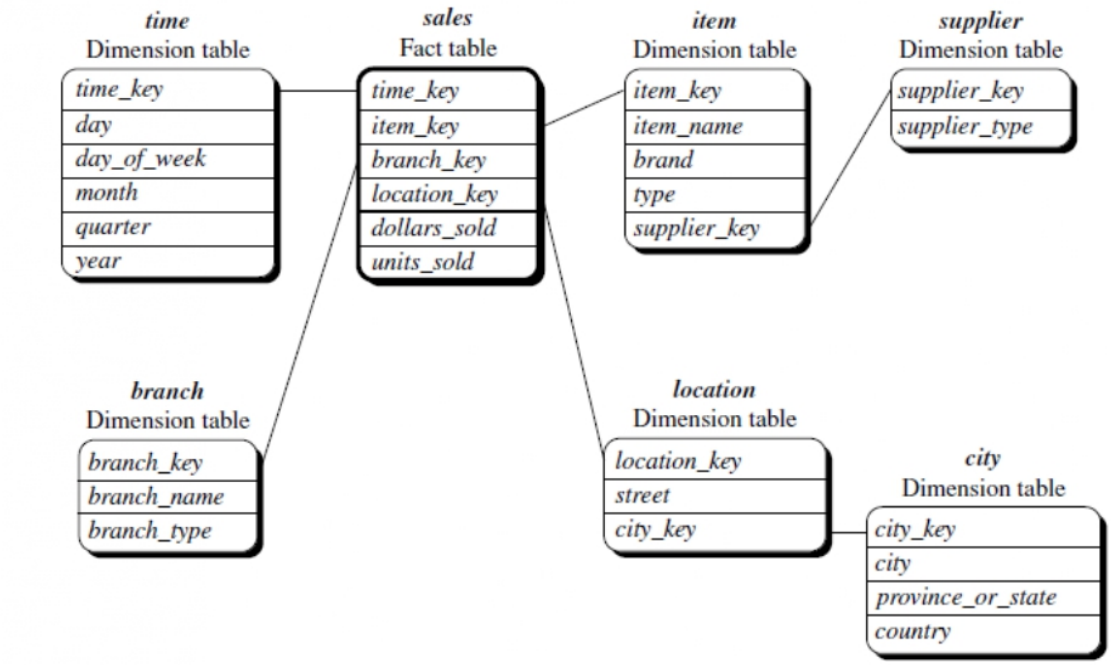
- few constraints may introduce redundancy

Snowflake schema

- The snowflake schema is a variant of the star schema model
- some of the dimension tables are normalized by splitting the data into additional tables.
- The resulting schema graph forms a shape like a snowflake
- The snowflake model may be kept in the normalized form to reduce redundancies
- Snowflake structure can reduce the effectiveness of browsing since more joins will be needed to execute a query

Example

Warehouse: AllElectronics sales

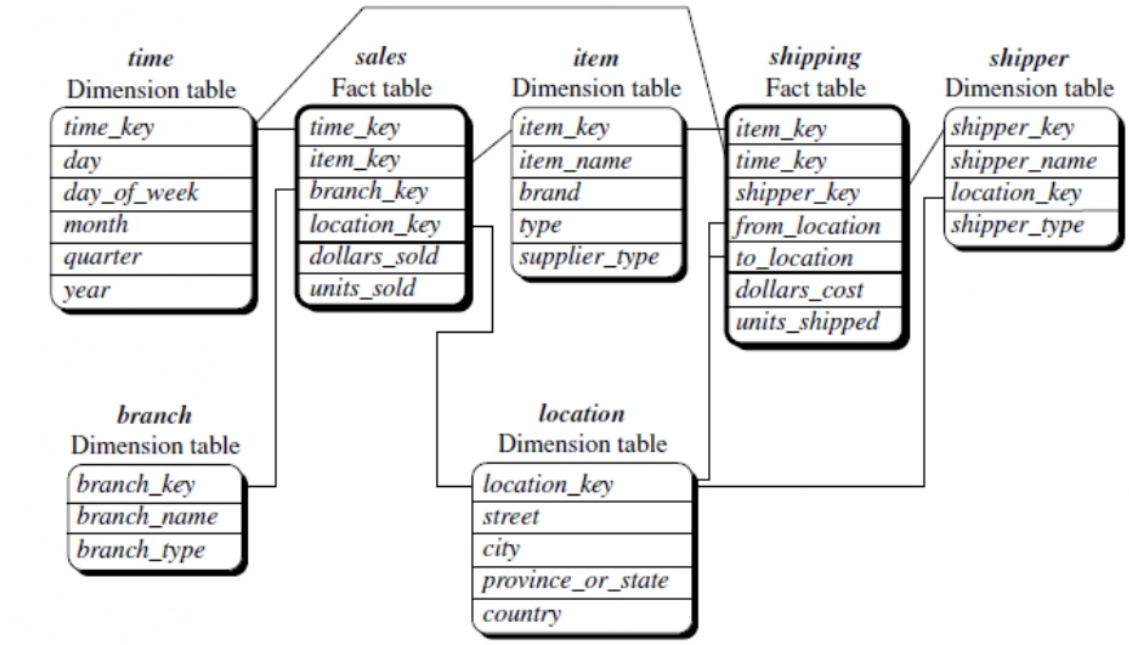


- The single dimension table for items in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables
- the item dimension table now contains the attributes item key, item name, brand, type, and supplier key,
- where supplier key is linked to the supplier dimension 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: area 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

Example



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

Note

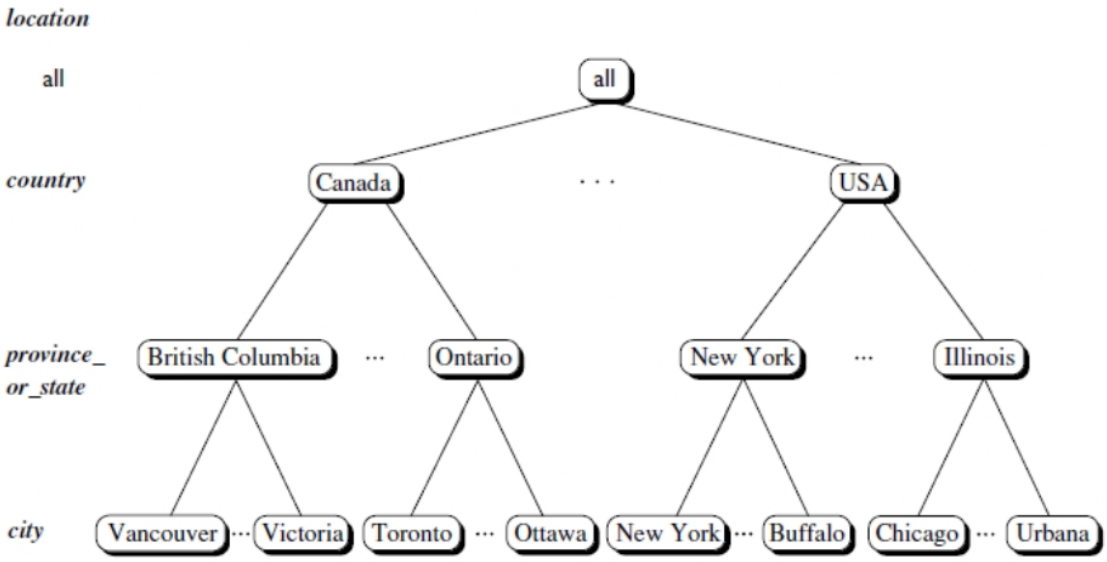
1. **For data warehouses, the fact constellation schema is commonly used.**
2. **For data marts, the star or snowflake schema is commonly used**

Dimensions: The Role of Concept Hierarchies

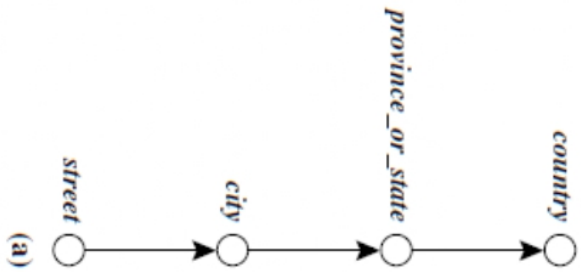
Concept Hierarchies

- It defines a sequence of mappings from a set of ***low-level concepts*** to ***higher-level***, more general concepts
  Example
  Dimension: Location
  Fact: City
  Values: Vancouver, Toronto, New York, and Chicago

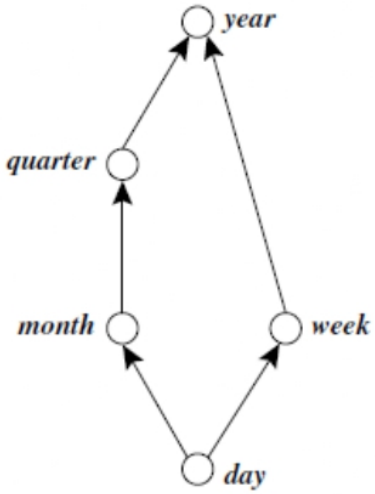
- Each city can be mapped to its respective state /province
  Example
  Vancouver can be mapped to British Columbia
- Each province and state can, in turn, be mapped to the country
  Example
  British Columbia mapped to Canada
- These mappings form a concept hierarchy for the dimension location, mapping low-level concepts (i.e., cities) to higher-level, more images (i.e., countries).



- Many concept hierarchies are implicit within the database schema.
  For example,
  Dimension location is described by the attributes number, street, city, province or state, zip code, and country
- These attributes are related by a total order, forming a concept hierarchy such as **"street < city < province or state < country"** as shown in the below figure.

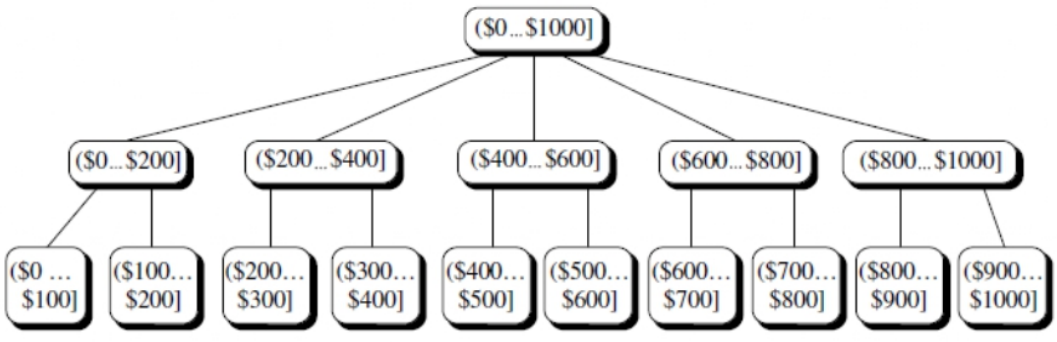


- A concept hierarchy for time



- A concept hierarchy that is a total or partial order among attributes in a database schema is called a **schema hierarchy**
- Concept hierarchies may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a **set-grouping hierarchy**

Example
Dimension: Price
Intervals: ($X …$Y)



Measures: Their Categorization and Computation

How are measures computed?

- Multidimensional points in the data cube space can be defined by dimensions–value pairs.
  Example
  <time = "Q1", location ="Vancouver", item = "computer"?>
- A **data cube measure** is a numeric function that can be evaluated at each point in the data cube space.
- A measured value is computed for a given point by aggregating the data corresponding to the **respective dimension–value pairs** defining the given point
- ***Measures*** can be organized into three categories based on the ***aggregation function***
  1. Distributive
  2. Algebraic
  3. Holistic

Distributive

- An aggregate function is distributive if it can be computed in a distributed manner.
- Suppose the data are partitioned into n sets.
- 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 first partitioning the cube into a set of subcubes, computing sum() for each subcube, and then summing up the counts obtained for each subcube. Hence, sum() is a distributive aggregate function. For the same reason, count(), min(), and max() are distributive aggregate functions.
- A measure is distributive if it is obtained by applying a distributive aggregate function.
- Distributive measures can be computed efficiently because the computation can be partitioned.

Algebraic

- An aggregate function is algebraic if it can be computed by an algebraic function with M arguments, Where M is a bounded positive integer) each of which is obtained by applying a distributive aggregate function.
  For example,
  1. avg() (average) can be computed by sum()/count(), where both sum() and count() are distributive aggregate functions.
  2. min N() and max N() (which find the N minimum and N maximum values, respectively, in a given set) and standard deviation() are algebraic aggregate functions.
  3. 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 on the storage size needed to describe a sub aggregate.
- That is, there does not exist an algebraic function with M arguments (where M is a constant) that characterizes the computation. Common examples of holistic functions include median (), mode(), and rank().
- A measure is holistic if it is obtained by applying a holistic aggregate function

Note

- Most large data cube applications require efficient computation of distributive and algebraic measures.
- Many efficient techniques for this exist.
- It is challenging to compute holistic measures efficiently.
- Efficient techniques to approximate the computation of some holistic measures do exist.

Typical OLAP Operations

- Several OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand.
- OLAP provides a user-friendly environment for interactive data analysis
- OLAP operations are
  1. Roll-Up-Up
  2. Drill -Down
  3. Slice and Dice
  4. Pivot(rotate)

Example

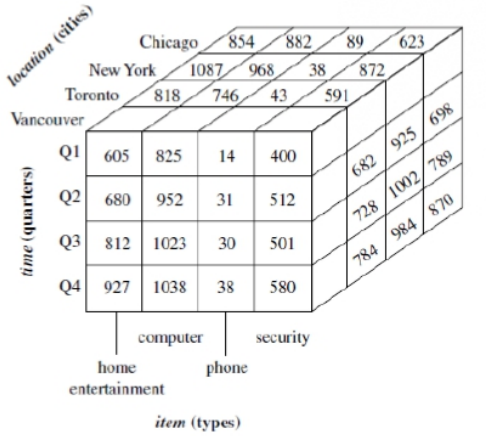
Data Cube represents: *AllElectronics* sales

Dimensions: location, time, and item

Where

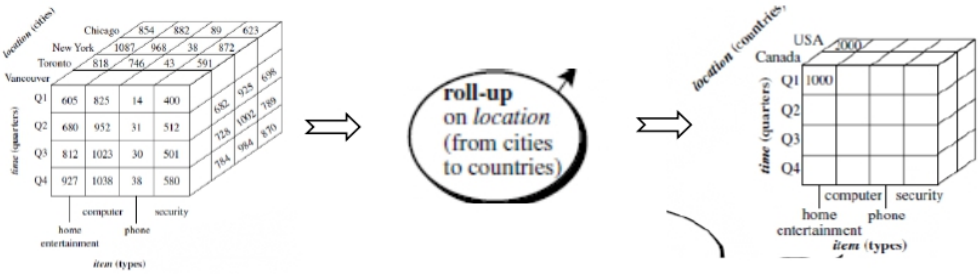
- location is aggregated to city values
- time is aggregated concerning quarters
- item is aggregated concerning item types

Measure: dollars sold



Roll-up:

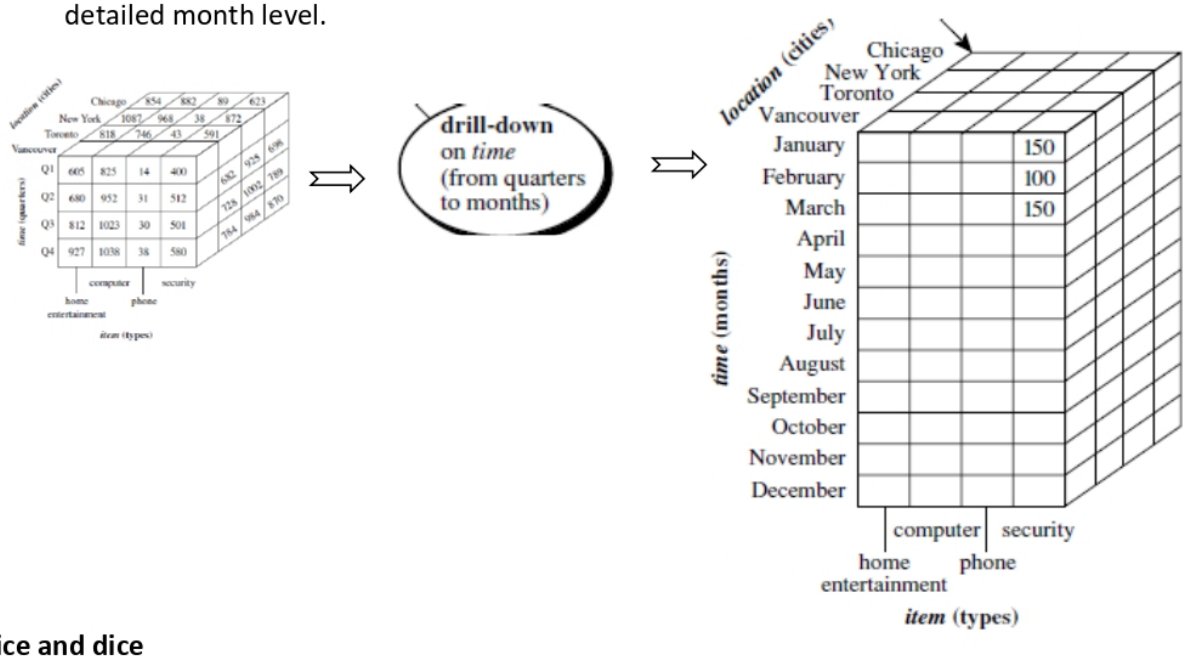
- The roll-up operation/ drill-up performs aggregation on a data cube by climbing up a concept hierarchy for a dimension or by dimension reduction.
- Below is the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for the location.



- This hierarchy was defined as the total order of "street < city < province or state < country.
- The roll-up operation shown aggregates the data by ascending the location hierarchy from the city level to the country level.
- When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube.

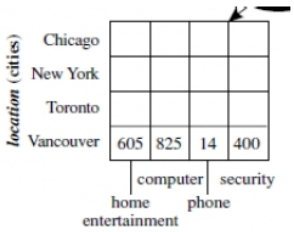
Drill-down

- Drill-down is the reverse of roll-up.
- It navigates from less detailed data to more complex data.
- Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.
- Below figure shows the result of a drill-down operation performed on the central cube by stepping down a concept hierarchy for the time defined as "day < month < quarter < year."
- Drill-down occurs by descending the time hierarchy from the quarter to the more detailed month level.



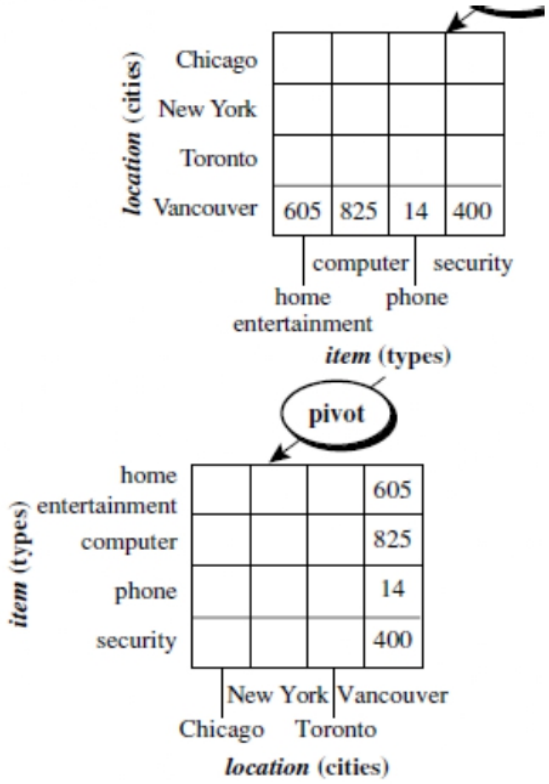
Slice and dice

- The slice operation selects one dimension of the given cube, resulting in a sub-cube.
- Below figure shows a slice operation where the sales data are selected from the central cube for the dimension time using the criterion time D "Q1."
- The dice operation defines a sub-cube by performing a selection on two or more dimensions



Pivot (rotate)

- Pivot (also called rotate) is a visualization operation that rotates the data axes in view to provide an alternative data presentation.
- Below figure shows a pivot operation where the item and location axes in a 2-D slice are rotated.



Other OLAP operations

- Some OLAP systems offer additional drilling operations.
  For example,
  1. Drill-across executes queries involving (i.e., across) more than one fact table.
  2. 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 return rates, depreciation, currency conversions, and statistical functions

Summary

- OLAP offers analytical modeling capabilities, including a calculation engine for deriving ratios, variance, and so on, for computing measures across multiple dimensions.
- It can generate summarizations, aggregations, and hierarchies at each granularity level and every dimension intersection.
- OLAP also supports functional models for forecasting, trend analysis, and statistical analysis.
- In this context, an OLAP engine is a powerful data analysis tool

OLAP Systems versus Statistical Databases

#	OLAP Systems	Statistical Databases
1	Targeted for business applications.	Focus on socioeconomic applications
2	No privacy issues	Privacy issues regarding concept hierarchies are a significant concern for SDBs.
3	Designed to support analytical application or decision making	Designed to help statistical applications.

Self-Learning Topic: **A Starnet Query Model for Querying Multidimensional Databases**

Source:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson, First impression,2014.

2. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining -Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.

Question Bank

1. What is a Data warehouse? Explain the three-tier architecture of the data warehouse. [8 Marks] [Dec 2019/Jan2020]
2. Explain the Schemas of multidimensional data models. [8 Marks] [Dec 2019/Jan2020]
3. What is a Data cube measure? Explain the categorization of measures. [8 Marks] [Dec 2019/Jan2020]
4. Explain data cube operations with examples. [8 Marks] [Dec 2019/Jan2020]
5. Describe a 3 -tier data warehouse architecture. [6 Marks] [June /July 2019] [8 Marks] [Aug /Sept 2020].
6. Compare OLTP and OLAP Systems. [6 Marks] [June /July 2019]
7. What is Data warehouse and what are its four key features? [4 Marks] [June /July 2019]
8. Explain with suitable examples the various OLAP operations in a multidimensional data model. [7 Marks] [June /July 2019] [8 Marks] [Aug /Sept 2020].
9. Explain the following terms with examples [9 Marks] [June /July 2019] [8 Marks] [Aug /Sept 2020].
   a. Snowflakes schema
   b. Fact constellation schema
   c. Star schema
10. What is Data Warehousing? Discuss various usage and trends in data warehousing . [8 Marks] [Aug /Sept 2020].