Data Warehousing & modeling: Basic Concepts: Difference between Operational Database systems and Data warehouse, Data Warehousing: A multitier Architecture, Data warehouse models: Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Metadata Repository, Data warehouse design and usage: Business Analysis framework, Data warehouse design process and usage for information processing, online analytical processing to multidimensional data mining. 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.

Basic Concepts

Data warehousing provides architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions. A data warehouse refers to a data repository that is maintained separately from an organization's operational databases. Data warehouse systems allow for integration of a variety of application systems. They support information processing by providing a solid platform of consolidated historic data for analysis.

- Data warehousing:The process of constructing and using data warehouses

The four keywords—*subject-oriented*, *integrated*, *time-variant*, and*nonvolatile*—*distinguish data warehouses from other data repository systems*, such asrelational database systems, transaction processing systems, and file systems.

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
- Provide 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
  - relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied.
  - 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: 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"

Nonvolatile:

- 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 comparison of operational systems and data warehousing systems

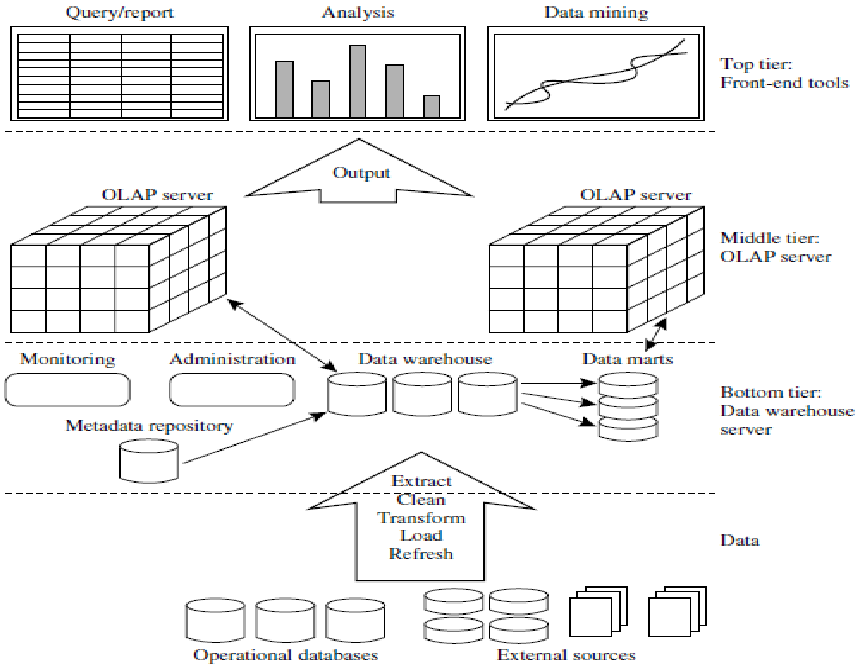
Operational systems	Data warehousing systems
Operational systems are generally designed to support high-volume *transaction processing* with minimal back-end reporting.	Data warehousing systems are generally designed to support high-volume *analytical processing* (i.e. *OLAP*) and subsequent, often elaborate *report generation*.
Operational systems are generally *process-oriented* or *process-driven*, meaning that they are focused on specific business processes or tasks. Example tasks include billing, registration, etc.	Data warehousing systems are generally *subject-oriented*, organized around business areas that the organization needs information about. Such subject areas are usually populated with data from one or more operational systems. As an example, revenue may be a subject area of a data warehouse that incorporates data from operational systems that contain student tuition data, alumni gift data, financial aid data, etc.
Operational systems are generally concerned with *current data*.	Data warehousing systems are generally concerned with *historical data*.
Data within operational systems are generally *updated regularly*according to need.	Data within a data warehouse is generally *non-volatile*, meaning that new data may be added regularly, but once loaded, the data is *rarely changed*, thus preserving an ever-growing *history of information*. In short, data within a data warehouse is generally *read-only*.
Operational systems are generally optimized to perform *fast inserts and updates* of relatively *small volumes of data*.	Data warehousing systems are generally optimized to perform *fast retrievals* of relatively *large volumes of data*.
Operational systems are generally *application-specific*, resulting in a multitude of partially or non-integrated systems and *redundant data*(e.g. billing data is not integrated with payroll data).	Data warehousing systems are generally *integrated* at a layer above the application layer, avoiding data redundancy problems.
Operational systems generally require a *non-trivial level of computing skills* amongst the end-user community.	Data warehousing systems generally appeal to an end-user community with a *wide range of computing skills*, from novice to expert users.

OLTP vs. OLAP

The Online Analytical Processing is designed to answer multi-dimensional queries, whereas the Online Transaction Processing is designed to facilitate and manage the usual business applications. While OLAP is customer-oriented, OLTP is market oriented. Both OLTP and OLAP are two of the common systems for the management of data. The OLTP is a category of systems that manages transaction processing.

	OLTP	**OLAP**
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-datedetailed, flat relational isolated	historical, summarized, multidimensionalintegrated, consolidated
usage	repetitive	ad-hoc
access	read/writeindex/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

Data Warehousing: A multitier Architecture



A Three-tier Data Warehouse Architecture

Generally a data warehouses adopts a three-tier architecture. Following are the three tiers of the data warehouse architecture.

Bottom Tier − The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back end tools and utilities to feed data into the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.

- **Data extraction**
  - get data from multiple, heterogeneous, and external sources
- **Data cleaning**
  - detect errors in the data and rectify them when possible
- **Data transformation**
  - convert data from legacy or host format to warehouse format
- **Load**
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

Metadata Repository:

- Meta data is the data defining warehouse objects. It stores:
- Description of the structure of the data warehouse
  - schema, view, dimensions, hierarchies, derived data defn, data mart locations and contents
- Operational meta-data
  - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)

Middle Tier − In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.

By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.

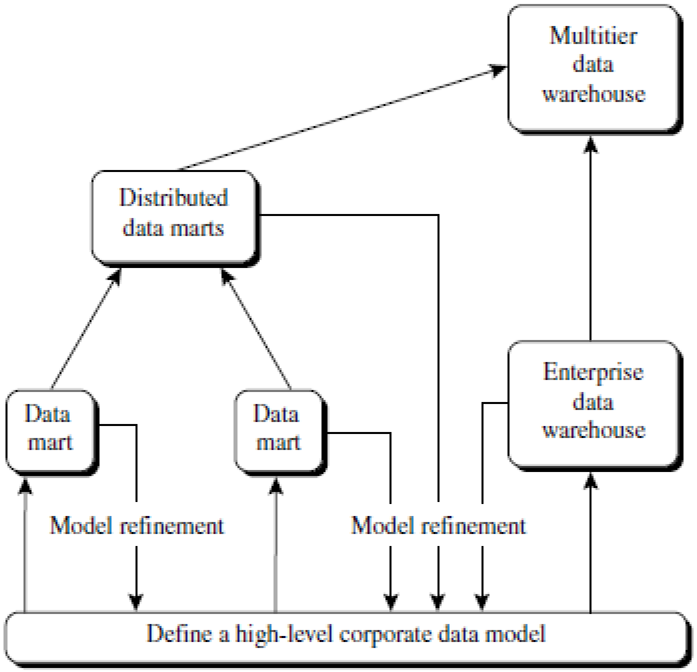
By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.

Top-Tier − This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

Data warehouse models

Three Data Warehouse Models:

- Enterprise warehouse
  - collects all of the information about subjects spanning the entire organization
- Data Mart
  - A subset of corporate-wide data that is of value to specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
- Virtual warehouse
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized



A recommended approach for data warehouse development.

Extraction, Transformation and loading

- **Data extraction**
  - get data from multiple, heterogeneous, and external sources
- **Data cleaning**
  - detect errors in the data and rectify them when possible
- **Data transformation**
  - convert data from legacy or host format to warehouse format
- **Load**
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- **Refresh**
  - propagate the updates from the data sources to the warehouse

Metadata Repository

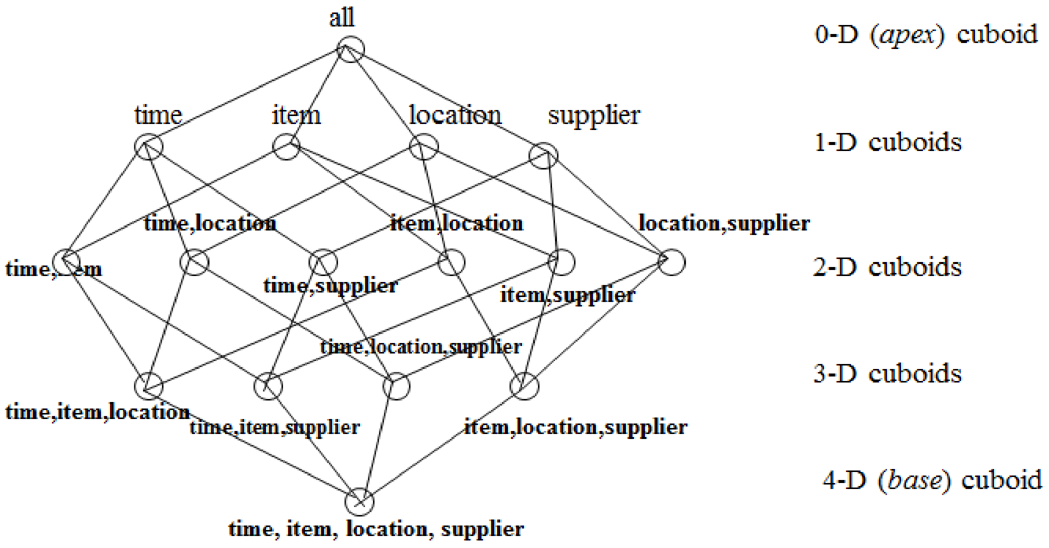
- Meta data is the data defining warehouse objects. It stores:
- Description of the structure of the data warehouse
  - schema, view, dimensions, hierarchies, derived data defn, data mart locations and contents
- Operational meta-data
  - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)

Data Cube

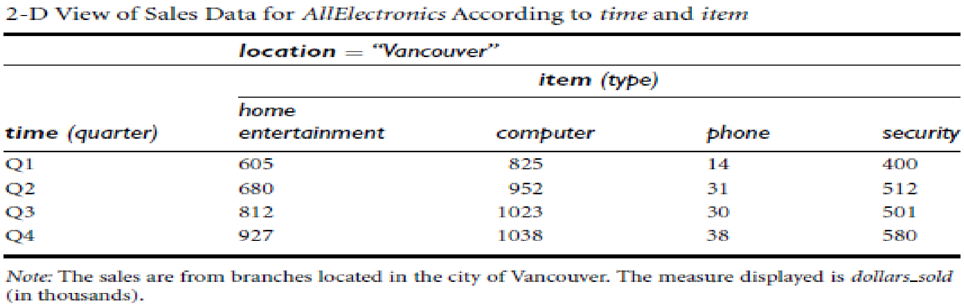
A data warehouse is based on a multidimensional data model which views data in the form of a **data cube.** A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions

- Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
- Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables

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. The lattice of cuboids forms a data cube.



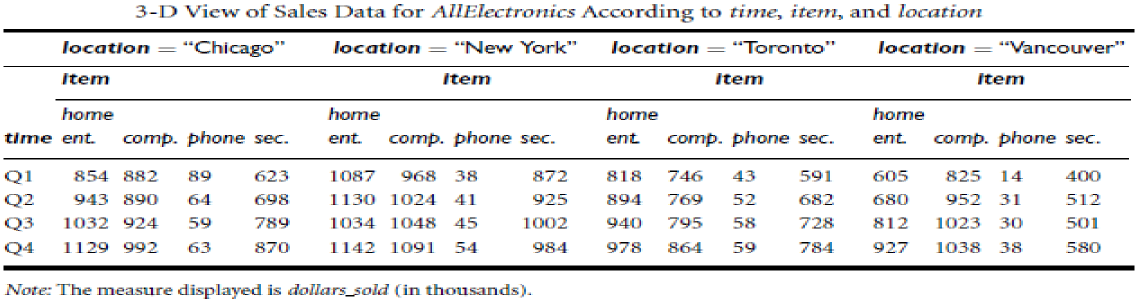
Cube: A lattice of cuboids



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

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

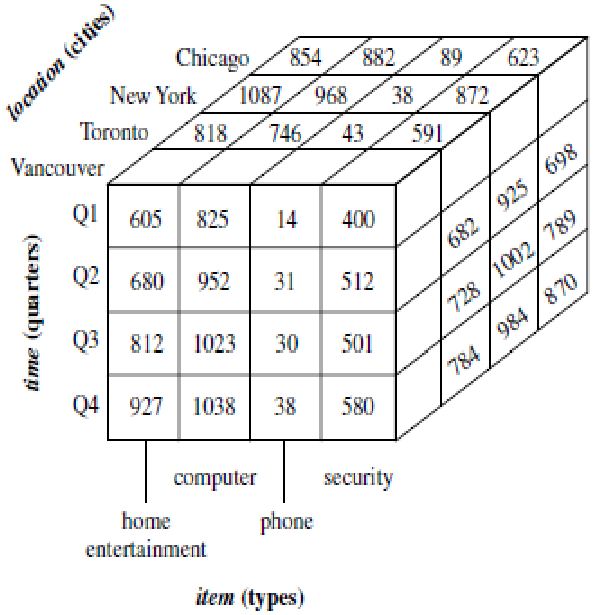
Note: The sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).



3-D View of Sales Data for *AllElectronics* According to *time*, *item*, and *location*

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

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



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

Schemas for multidimensional Data models

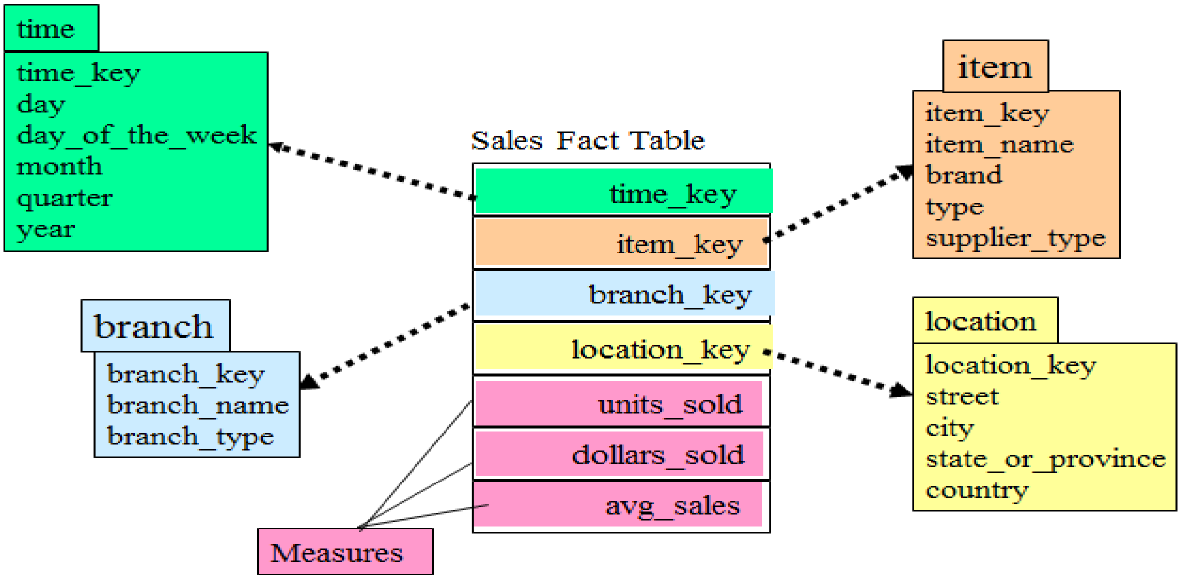
Fact Table: Refers to measurements for specific event. Along with numeric values, the table also consists of foreign keys pointing to the dimension tables. The table is designed in a way such that the facts included can store values at atomic level , which allows the storage of large number of records at a time.

Dimension Table: Since dimension tables support descriptive attribute data, the number of records inserted is relatively lesser than the fact table. The dimensions in a data warehouse can explain a variety of characteristics. The tables are also assigned a surrogate primary key (single-column integer data type). For instance, you can create a Sales fact table referring to an event with product key, customer key, promotion key, date key, items sold and revenue generated. For every fact table key, there is a dimension table like Product dimension table containing reference information as product name, product type, quantity, size, color, description and others.

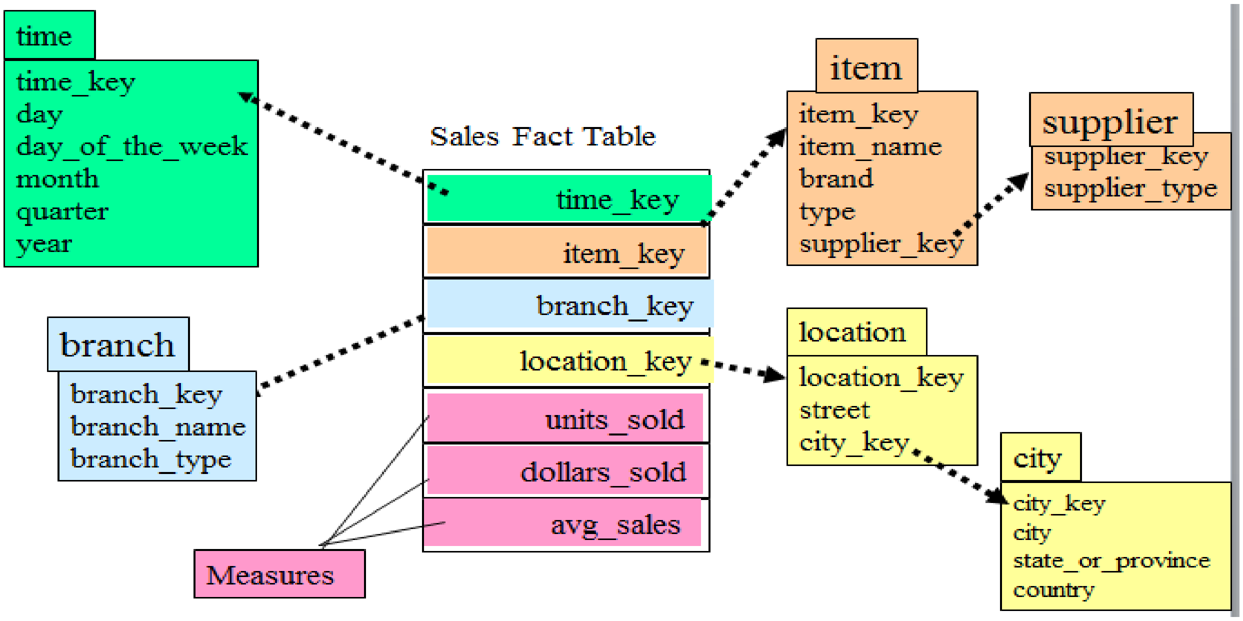
Star schema: star schema consists of data in the form of facts and dimensions. The reason behind the name 'Star Schema' is that this data model resembles a star with ends radiating from the center , where the center refers to the fact table and the radiating points are dimension tables.

Snowflake schema: Snowflake Schema is the extension to star schema such that the tables are arranged in a complex snowflake shape. The concept is similar to star schema with a center fact table and multiple dimension tables radiating from the center except that the tables described as dimensions are normalized and consist of more hierarchies.

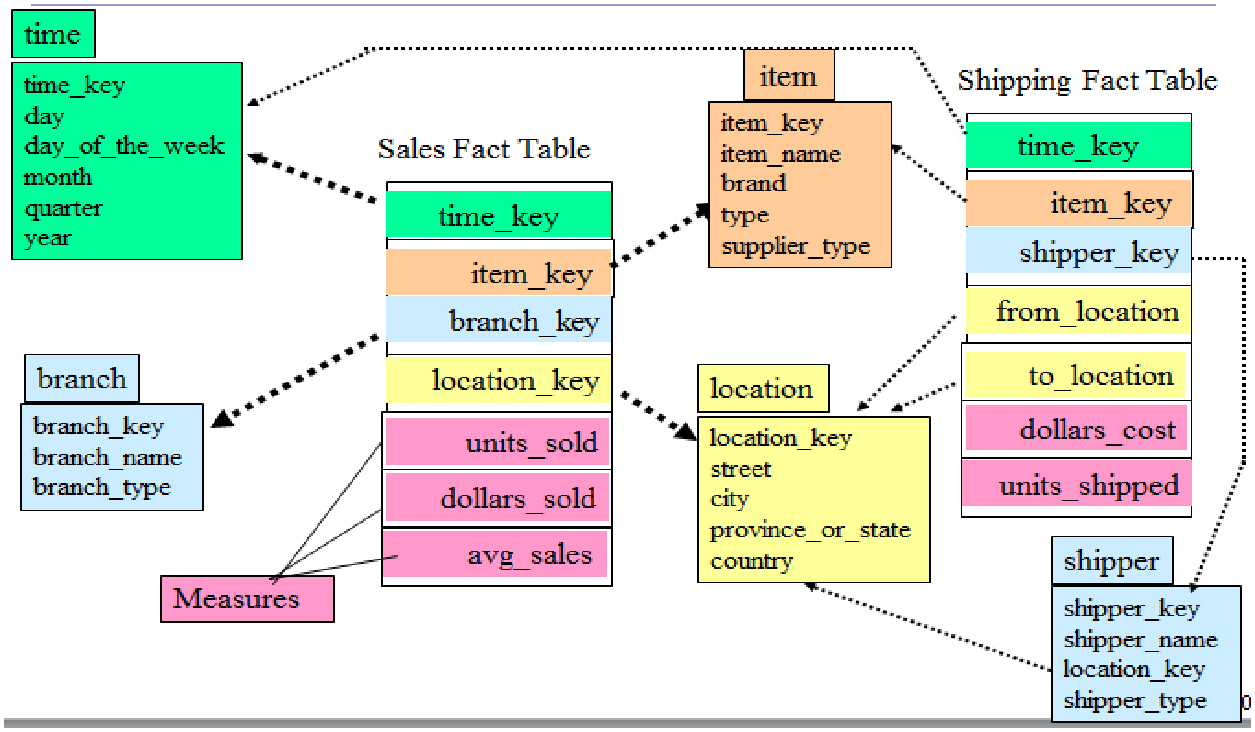
Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation



Star Schema



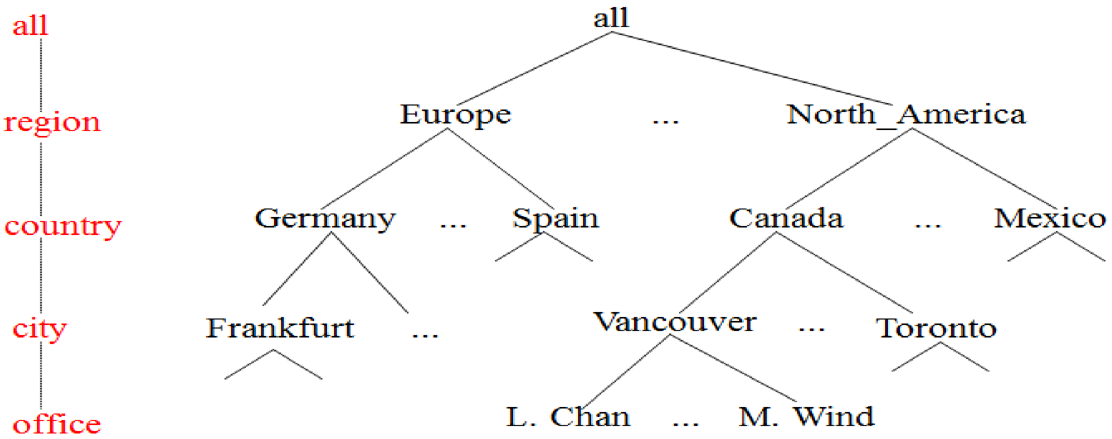
Snowflake Schema



Fact constellation (galaxy schema)

Dimensions:The role of conceptHierarchies

A Concept Hierarchy: Dimension (location)



A **concept hierarchy** defines a sequence of mappings from a set of low-level conceptsto higher-level, more general concepts. Consider a concept hierarchy for the dimension*location*. City values for *location* include Vancouver, Toronto, New York, and Chicago.Each city, however, can be mapped to the province or state to which it belongs. Forexample, Vancouver can be mapped to British Columbia, and Chicago to Illinois.The provinces and states can in turn be mapped to the country (e.g., Canada or theUnited States) to which they belong. These mappings form a concept hierarchy for thedimension *location*, mapping a set of low-level concepts (i.e., cities) to higher-level, moregeneral concepts (i.e., countries).

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* bydefining ranges for *inexpensive*, *moderately priced*, and *expensive*.Concept hierarchies may be provided manually by system users, domain experts, orknowledge engineers, or may be automatically generated based on statistical analysis ofthe data distribution.

Measures: Their Categorization and Computation

A data cube **measure** is a numeric function that can be evaluatedat each point in the data cube space. A measure value is computed for a given point byaggregating the data corresponding to the respective dimension–value pairs defining thegiven point.

Measures can be organized into three categories—distributive, algebraic, and holistic—based on the kind of aggregate functions used.

- Distributive: if the result derived by applying the function to *n*aggregate values is the same as that derived by applying the function on all the data without partitioning
    - E.g., count(), sum(), min(), max()
- Algebraic:if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function
    - E.g.,avg(), min_N(), standard_deviation()
- Holistic: if there is no constant bound on the storage size needed to describe a subaggregate.
    - E.g., median(), mode(), rank()

Typical OLAP Operations

OLAP provides a user-friendly environment for interactive data analysis. A number of OLAP data cube operations exist to materialize different views of data, allowing interactive querying and analysis of the data.

The most popular end user operations on dimensional data are:

Roll up

The roll-up operation (also called drill-up or aggregation operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by climbing down a concept hierarchy, i.e. dimension reduction. Let me explain roll up with an example:

Roll Down

The roll down operation (also called drill down) is the reverse of roll up. It navigates from less detailed data to more detailed data. It can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.

Slicing

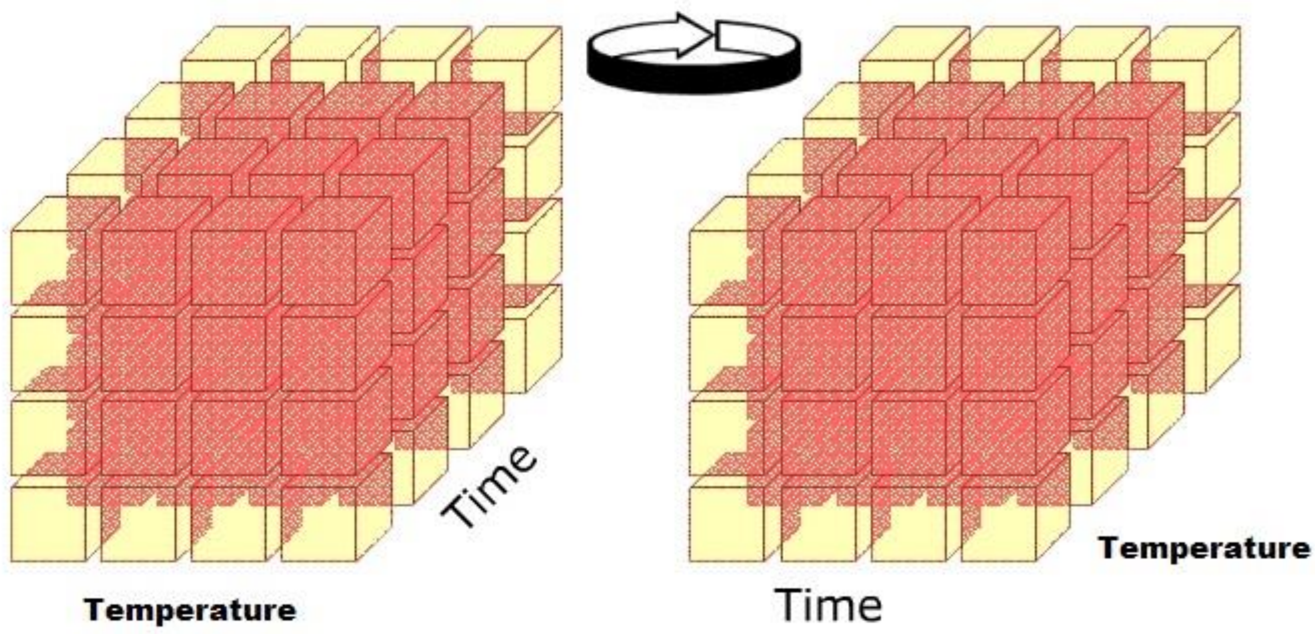
Slice performs a selection on one dimension of the given cube, thus resulting in a subcube.

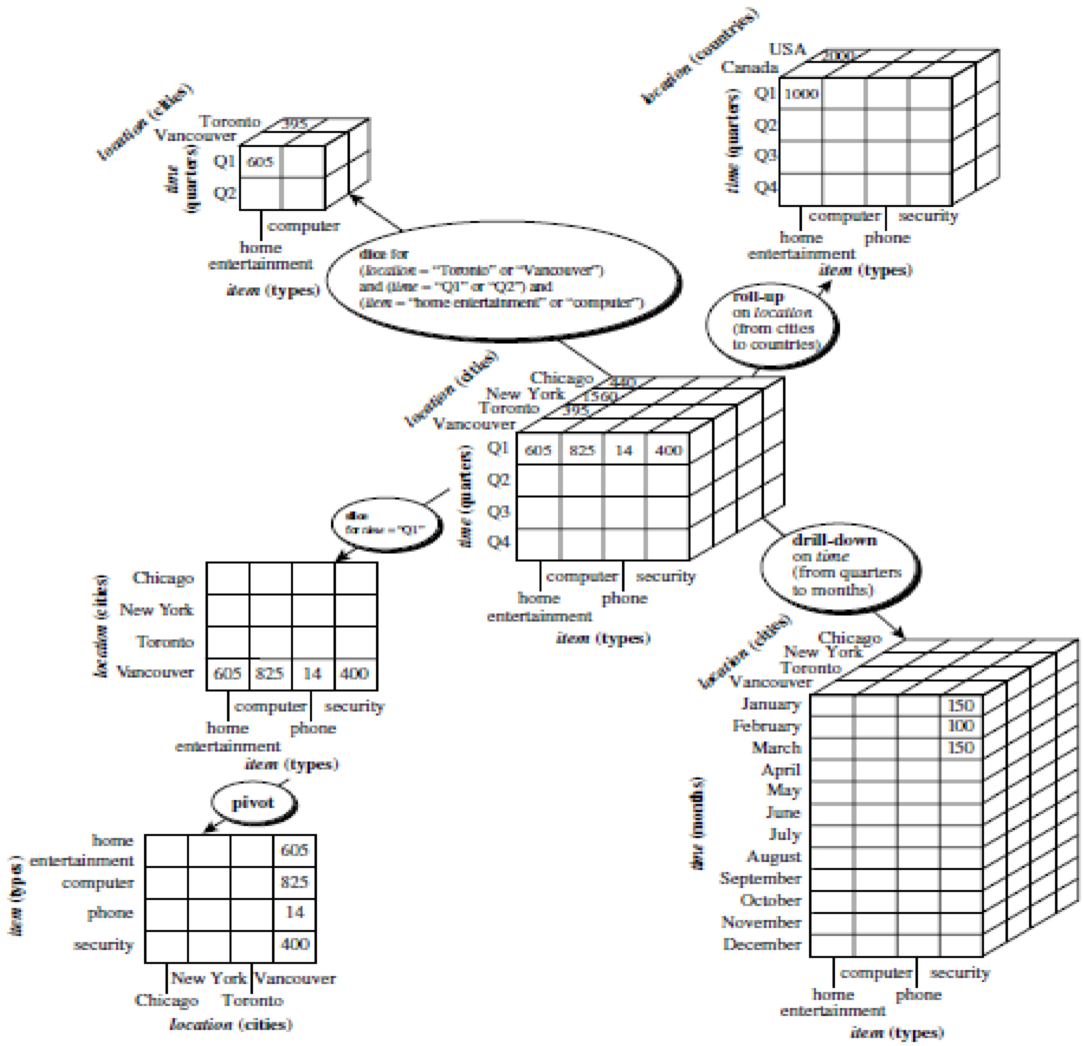
Dicing

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

Pivot

Pivot otheriwise known as Rotate changes the dimensional orientation of the cube, i.e. rotates the data axes to view the data from different perspectives. Pivot groups data with different dimensions. The below cubes shows 2D represntation of Pivot.





OLAP Operations