**Unit 1 : Introduction to Data Mining and Data Warehousing**

**What is Data?**

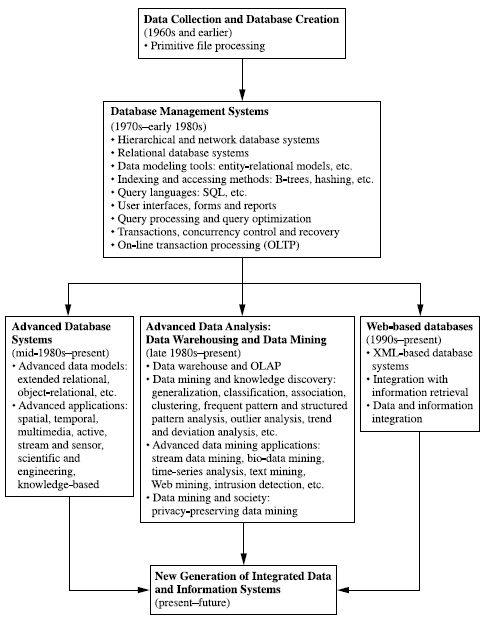
* A representation of facts, concepts, or instructions in a formal manner suitable for communication, interpretation, or processing by human beings or by computers.

**Review of basic concepts of data warehousing and data mining**

* The Explosive Growth of Data: from terabytes to petabytes
* Data accumulate and double every 9 months
* High-dimensionality of data
* High complexity of data
* New and sophisticated applications
* There is a big gap from stored data to knowledge; and the transition won’t occur automatically.
* Manual data analysis is not new but a bottleneck
* Fast developing Computer Science and Engineering generates new demands

**Evolution of Database Technology**

* 1960s:
  + Data collection, database creation, IMS and network DBMS
* 1970s:
  + Relational data model, relational DBMS implementation
* 1980s:
  + RDBMS, advanced data models (extended-relational, OO, deductive, etc.) and application-oriented DBMS (spatial, scientific, engineering, etc.)
* 1990s—2000s:
  + Data mining and data warehousing, multimedia databases, and Web databases



**Figure: The evolution of database system technology**

Very Large Databases

* Terabytes -- 10^12 bytes:
* Petabytes -- 10^15 bytes:
* Exabytes -- 10^18 bytes:
* Zettabytes -- 10^21 bytes:
* Zottabytes -- 10^24 bytes:

Data explosion problem

Automated data collection tools and mature database technology lead to tremendous amounts of data accumulated and/or to be analyzed in databases, data warehouses, and other information repositories

We are drowning in data, but starving for knowledge!

**Solution:**

“Necessity is the mother of invention”—**Data Warehousing and Data Mining**

What is Data Mining?

**Art/Science of extracting non-trivial, implicit,**

**previously unknown, valuable, and potentially**

**Useful information from a large database**

**Data mining is**

* A hot buzzword for a class of techniques that find patterns in data
* A user-centric, interactive process which leverages analysis technologies and computing power
* A group of techniques that find relationships that have not previously been discovered
* Not reliant on an existing database
* A relatively easy task that requires knowledge of the business problem/subject matter expertise

**Data mining is not**

Brute-force crunching of bulk data

“Blind” application of algorithms

Going to find relationships where none exist

Presenting data in different ways

A difficult to understand technology requiring an advanced degree in computer science

**Data mining is not**

* A cybernetic magic that will turn your data into gold. It’s the process and result of knowledge production, knowledge discovery and knowledge management.
* Once the patterns are found Data Mining process is finished.
* Queries to the database are not DM**.**

**What is Data Warehouse?**

* According to W. H. Inmon, a data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management decisions.
* “A data warehouse is a copy of transaction data specifically structured for querying and reporting” – Ralph Kimball
* Data Warehousing is the process of building a data warehouse for an organization.
* Data Warehousing is a process of transforming data into information and making it available to users in a timely enough manner to make a difference

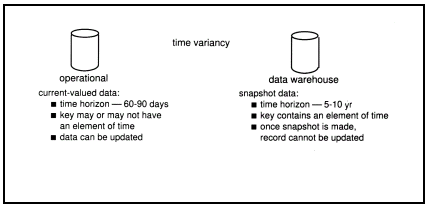
**Subject Oriented**

* Focus is on Subject Areas rather than Applications
* Organized around major subjects, such as customer, product, sales.
* 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
* Integration tasks handles naming conventions, physical attributes of data
* Must be made consistent.

**Time Variant**

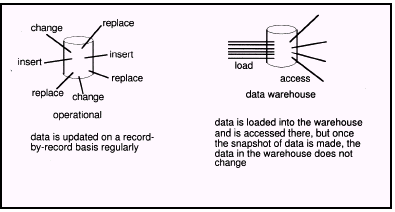
****

* Only accurate and valid at some point in time or over some time interval.
* The time horizon for the data warehouse is significantly longer than that of operational systems.

Operational database provides current value data.

Data warehouse data provide information from a historical perspective (e.g., past 5-10 years)

**Non Volatile**



* Data Warehouse is relatively Static in nature.
* Not updated in real-time but data in the data warehouse is loaded and refreshed from operational systems, it is not updated by end users.

Data warehousing helps business managers to :

* + Extract data from various source systems on different platforms
  + Transform huge data volumes into meaningful information
  + Analyze integrated data across multiple business dimensions
  + Provide access of the analyzed information to the business users anytime anywhere

**OLTP vs. Data Warehouse**

* Online Transaction Processing (OLTP) systems are tuned for known transactions and workloads while workload is not known a priori in a data warehouse
* OLTP applications normally automate clerical data processing tasks of an organization, like data entry and enquiry, transaction handling, etc. (access, read, update)
* Special data organization, access methods and implementation methods are needed to support data warehouse queries (typically multidimensional queries)
  + e.g*., average amount spent on phone calls between 9AM-5PM in Kathmandu during the month of March, 2012*
* OLTP
  + Application Oriented
  + Used to run business
  + Detailed data
  + Current up to date
  + Isolated Data
  + Repetitive access
  + Clerical User
* **Data Warehouse**
* Subject Oriented
* Used to analyze business
* Summarized and refined
* Snapshot data
* Integrated Data
* Ad-hoc access
* Knowledge User (Manager)
* **OLTP**
  + Transaction throughput is the performance metric
  + Thousands of users
  + Managed in entirety
* Data Warehouse
* Query throughput is the performance metric
* Hundreds of users
* Managed by subsets

**Why Data Mining?**

Because it can improve customer service, better target marketing campaigns, identify high-risk clients, and improve production processes. In short, because it can help you or your company make or save money.

Data mining has been used to:

* Identify unexpected shopping patterns in supermarkets.
* Optimize website profitability by making appropriate offers to each visitor.
* Predict customer response rates in marketing campaigns.
* Defining new customer groups for marketing purposes.
* Predict customer defections: which customers are likely to switch to an alternative supplier in the near future.
* Distinguish between profitable and unprofitable customers.
* Identify suspicious (unusual) behavior, as part of a fraud detection process.
* Data analysis and decision support
  + Market analysis and management
    - Target marketing, customer relationship management (CRM), market basket analysis, cross selling, market segmentation
  + Risk analysis and management
    - Forecasting, customer retention, improved underwriting, quality control, competitive analysis
  + Fraud detection and detection of unusual patterns (outliers)
* Other Applications
  + Text mining (news group, email, documents) and Web mining
  + Stream data mining
  + Bioinformatics and bio-data analysis

**Market Analysis and Management**

* Where does the data come from?—Credit card transactions, loyalty cards, discount coupons, customer complaint calls, plus (public) lifestyle studies
* Target marketing
  + Find clusters of “model” customers who share the same characteristics: interest, income level, spending habits, etc.
  + Determine customer purchasing patterns over time
* Cross-market analysis—Find associations/co-relations between product sales, & predict based on such association
* Customer profiling—What types of customers buy what products (clustering or classification)
* Customer requirement analysis
  + Identify the best products for different groups of customers
  + Predict what factors will attract new customers
* Provision of summary information
  + Multidimensional summary reports
  + Statistical summary information (data central tendency and variation)

**Corporate Analysis & Risk Management**

* Finance planning and asset evaluation
  + cash flow analysis and prediction
  + contingent claim analysis to evaluate assets
  + cross-sectional and time series analysis (financial-ratio, trend analysis, etc.)
* Resource planning
  + summarize and compare the resources and spending
* Competition
  + monitor competitors and market directions
  + group customers into classes and a class-based pricing procedure
  + set pricing strategy in a highly competitive market

**Fraud Detection & Mining Unusual Patterns**

* Approaches: Clustering & model construction for frauds, outlier analysis
* Applications: Health care, retail, credit card service, telecomm.
  + Auto insurance: ring of collisions
  + Money laundering: suspicious monetary transactions
  + Medical insurance
    - Professional patients, ring of doctors, and ring of references
    - Unnecessary or correlated screening tests
  + Telecommunications: phone-call fraud
    - Phone call model: destination of the call, duration, time of day or week. Analyze patterns that deviate from an expected norm
  + Retail industry
    - Analysts estimate that 38% of retail shrink is due to dishonest employees
  + Anti-terrorism

**Knowledge Discovery in Databases Process**

* Data selection
* Cleaning
* Enrichment
* Coding
* Data Mining
* Reporting

**Data Selection**

Once you have formulated your informational requirements, the nest logical step is to collect and select the data you need. Setting up a KDD activity is also a long term investment. A data environment will need to download from operational data on a regular basis, therefore investing in a data warehouse is an important aspect of the whole process.

**Cleaning**

Almost all databases in large organizations are polluted and when we start to look at the data from a data mining perspective, ideas concerning consistency of data change. Therefore, before we start the data mining process, we have to clean up the data as much as possible, and this can be done automatically in many cases.

**Enrichment**

Matching the information from bought-in databases with your own databases can be difficult. A well-known problem is the reconstruction of family relationships in databases. In a relational environment, we can simply join this information with our original data.

Some Alternative names to data mining are:

* + Knowledge discovery (mining) in databases (KDD)
  + Knowledge extraction
  + Data/pattern analysis
  + Data archeology
  + Data Dredging
  + Information Harvesting
  + Business intelligence, etc.

**Operational Data Sources:** Itmay include:

* Network databases.
* Departmental file systems and RDBMSs.
* Private workstations and servers.
* External systems (Internet, commercially available databases).

**Operational Data Store (ODS):** It is a repository of current and integrated operational data used for analysis.

* Often structured and supplied with data in same way as DW.
* May act simply as staging area for data to be moved into the warehouse.
* Provides users with the ease of use of a relational database while remaining distant from decision support functions of the DW.

**Warehouse Manager** (Data Manager):

* Operations performed include:
  + Analysis of data to ensure consistency.
  + Transformation/merging of source data from temp storage into DW
  + Creation of indexes.
  + Backing-up and archiving data.

**Query Manager** (Manages User Queries):

* Operations include:
  + directing queries to the appropriate tables and
  + scheduling the execution of queries.
* In some cases, the query manager also generates query profiles to allow the warehouse manager to determine which indexes and aggregations are appropriate.

**Meta Data:** This area of the DW stores all the meta-data (data about data) definitions used by all the processes in the warehouse.

* Used for a variety of purposes:
  + Extraction and loading processes
  + Warehouse management process
  + Query management process
* End-user access tools use meta-data to understand how to build a query.
* Most vendor tools for copy management and end-user data access use their own versions of meta-data.

**Lightly and Highly Summarized Data:** It stores all the pre-defined lightly and highly aggregated data generated by the warehouse manager.

* The purpose of summary info is to speed up the performance of queries.
* Removes the requirement to continually perform summary operations (such as sort or group by) in answering user queries.

**Archive/Backup Data:** It stores detailed and summarized data for the purposes of archiving and backup.

* May be necessary to backup online summary data if this data is kept beyond the retention period for detailed data.
* The data is transferred to storage archives such as magnetic tape or optical disk.

**End-User Access Tools:**

* The principal purpose of data warehousing is to provide information to business users for strategic decision-making.
* Users interact with the warehouse using end-user access tools.
* There are three main groups of access tools:
  1. Data reporting, query tools
  2. Online analytical processing (OLAP) tools *(Discussed later)*
  3. Data mining tools *(Discussed later)*

**Benefits of Data Warehousing**

* Queries do not impact Operational systems
* Provides quick response to queries for reporting
* Enables Subject Area Orientation
* Integrates data from multiple, diverse sources
* Enables multiple interpretations of same data by different users or groups
* Provides thorough analysis of data over a period of time
* Accuracy of Operational systems can be checked
* Provides analysis capabilities to decision makers
  + Increase customer profitability
  + Cost effective decision making
  + Manage customer and business partner relationships
  + Manage risk, assets and liabilities
  + Integrate inventory, operations and manufacturing
  + Reduction in time to locate, access, and analyze information (Link multiple locations and geographies)
  + Identify developing trends and reduce time to market
  + Strategic advantage over competitors
* Potential high returns on investment
* Competitive advantage
* Increased productivity of corporate decision-makers
* Provide reliable, High performance access
* Consistent view of Data: Same query, same data. All users should be warned if data load has not come in.
* Quality of data is a driver for business re-engineering.

**Applications of Data Mining**

* Data mining is an interdisciplinary field with wide and diverse applications
  + There exist nontrivial gaps between data mining principles and domain-specific applications
* Some application domains
  + Financial data analysis
  + Retail industry
  + Telecommunication industry
  + Biological data analysis

**Data Mining for Financial Data Analysis**

* Financial data collected in banks and financial institutions are often relatively complete, reliable, and of high quality
* Design and construction of data warehouses for multidimensional data analysis and data mining
  + View the debt and revenue changes by month, by region, by sector, and by other factors
  + Access statistical information such as max, min, total, average, trend, etc.
* Loan payment prediction/consumer credit policy analysis
  + feature selection and attribute relevance ranking
  + Loan payment performance
  + Consumer credit rating
* Classification and clustering of customers for targeted marketing
  + multidimensional segmentation by nearest-neighbor, classification, decision trees, etc. to identify customer groups or associate a new customer to an appropriate customer group
* Detection of money laundering and other financial crimes
  + integration of from multiple DBs (e.g., bank transactions, federal/state crime history DBs)
  + Tools: data visualization, linkage analysis, classification, clustering tools, outlier analysis, and sequential pattern analysis tools (find unusual access sequences)

**Data Mining for Retail Industry**

* Retail industry: huge amounts of data on sales, customer shopping history, etc.
* Applications of retail data mining
  + Identify customer buying behaviors
  + Discover customer shopping patterns and trends
  + Improve the quality of customer service
  + Achieve better customer retention and satisfaction
  + Enhance goods consumption ratios
  + Design more effective goods transportation and distribution policies
* Example 1. Design and construction of data warehouses based on the benefits of data mining
  + Multidimensional analysis of sales, customers, products, time, and region
* Example 2. Analysis of the effectiveness of sales campaigns
* Example 3. Customer retention: Analysis of customer loyalty
  + Use customer loyalty card information to register sequences of purchases of particular customers
  + Use sequential pattern mining to investigate changes in customer consumption or loyalty
  + Suggest adjustments on the pricing and variety of goods
* Example 4. Purchase recommendation and cross-reference of items

**Data Mining for Telecommunication Industry**

* A rapidly expanding and highly competitive industry and a great demand for data mining
  + Understand the business involved
  + Identify telecommunication patterns
  + Catch fraudulent activities
  + Make better use of resources
  + Improve the quality of service
* Multidimensional analysis of telecommunication data
  + Intrinsically multidimensional: calling-time, duration, location of caller, location of callee, type of call, etc.
* Fraudulent pattern analysis and the identification of unusual patterns
  + Identify potentially fraudulent users and their typical usage patterns
  + Detect attempts to gain fraudulent entry to customer accounts
  + Discover unusual patterns which may need special attention
* Multidimensional association and sequential pattern analysis
  + Find usage patterns for a set of communication services by customer group, by month, etc.
  + Promote the sales of specific services
  + Improve the availability of particular services in a region
* Use of visualization tools in telecommunication data analysis

**Biomedical Data Analysis**

* DNA sequences: 4 basic building blocks (nucleotides): adenine (A), cytosine (C), guanine (G), and thymine (T).
* Gene: a sequence of hundreds of individual nucleotides arranged in a particular order
* Humans have around 30,000 genes
* Tremendous number of ways that the nucleotides can be ordered and sequenced to form distinct genes
* Semantic integration of heterogeneous, distributed genome databases
  + Current: highly distributed, uncontrolled generation and use of a wide variety of DNA data
  + Data cleaning and data integration methods developed in data mining will help
* Similarity search and comparison among DNA sequences
  + Compare the frequently occurring patterns of each class (e.g., diseased and healthy)
  + Identify gene sequence patterns that play roles in various diseases
* Association analysis: identification of co-occurring gene sequences
  + Most diseases are not triggered by a single gene but by a combination of genes acting together
  + Association analysis may help determine the kinds of genes that are likely to co-occur together in target samples
* Path analysis: linking genes to different disease development stages
  + Different genes may become active at different stages of the disease
  + Develop pharmaceutical interventions that target the different stages separately
* Visualization tools and genetic data analysis

**Problems in Data Warehousing**

* Underestimation of resources for data loading
* Hidden problems with source systems
* Required data not captured
* Increased end-user demands
* Data homogenization
* High demand for resources
* Data ownership
* High maintenance
* Long duration projects
* Complexity of integration

**Major Challenges in Data Warehousing**

* Data mining requires single, separate, clean, integrated, and self-consistent source of data.
  + A DW is well equipped for providing data for mining.
* Data quality and consistency is essential to ensure the accuracy of the predictive models.
  + DWs are populated with clean, consistent data
* Advantageous to mine data from multiple sources to discover as many interrelationships as possible.
  + DWs contain data from a number of sources.
* Selecting relevant subsets of records and fields for data mining
  + requires query capabilities of the DW.
* Results of a data mining study are useful if can further investigate the uncovered patterns.
  + DWs provide capability to go back to the data source.
* The largest challenge a data miner may face is the sheer volume of data in the data warehouse.
* It is quite important, then, that summary data also be available to get the analysis started.
* A major problem is that this sheer volume may mask the important relationships the data miner is interested in.
* The ability to overcome the volume and be able to interpret the data is quite important.

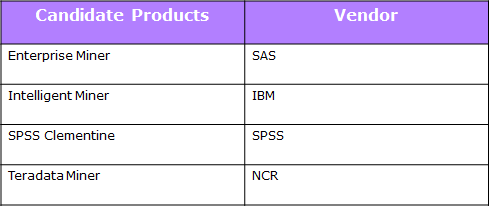
**Major Challenges in Data Mining**

* Efficiency and scalability of data mining algorithms
* Parallel, distributed, stream, and incremental mining methods
* Handling high-dimensionality
* Handling noise, uncertainty, and incompleteness of data
* Incorporation of constraints, expert knowledge, and background knowledge in data mining
* Pattern evaluation and knowledge integration
* Mining diverse and heterogeneous kinds of data: e.g., bioinformatics, Web, software/system engineering, information networks
* Application-oriented and domain-specific data mining
* Invisible data mining (embedded in other functional modules)
* Protection of security, integrity, and privacy in data mining

**Warehouse Products**

* Computer Associates -- CA-Ingres
* Hewlett-Packard -- Allbase/SQL
* Informix -- Informix, Informix XPS
* Microsoft -- SQL Server
* Oracle -- Oracle7, Oracle Parallel Server
* Red Brick -- Red Brick Warehouse
* SAS Institute -- SAS
* Software AG -- ADABAS
* Sybase -- SQL Server, IQ, MPP

**Data Mining Products**



**Unit 2: Data Warehouse Logical Design**

Database Development Process

* A conceptual data model include identification of important entities and the relationships among them. At this level, the objective is to identify the relationships among the different entities.

**What is a Logical Design??**

**Logical design** is the phase of a database design concerned with identifying the relationships among the data elements.

A logical design is conceptual and abstract. You do not deal with the physical implementation details yet. You deal only with defining the types of information that you need.

Logical design deals with concepts related to a certain kind of DBMS (e.g. relational, object oriented,) but are understandable by end users

The logical design should result in

1. A set of entities and attributes corresponding to fact tables and dimension tables.
2. A model of operational data from your source into subject-oriented information in your target data warehouse schema.

You can create the logical design using a pen and paper, or you can use a design tool such as Oracle Warehouse Builder (specifically designed to support modeling the ETL process) or Oracle Designer (a general purpose modeling tool).

The steps of the logical data model include identification of all entities and relationships among them. All attributes for each entity are identified and then the primary key and foreign key is identified. Normally normalization occurs at this level.

In data warehousing, it is common to combine the conceptual data model and the logical data model to a single step. The steps for logical data model are indicated below:

1. Identify all entities.
2. Identify primary keys for all entities.
3. Find the relationships between different entities.
4. Find all attributes for each entity.
5. Resolve all entity relationships that is many-to-many relationships.
6. Normalization if required.

The environment provides the infrastructure to carry out the specified process. It consists of:

* A *refined conceptual schema*, which is built from a conceptual multidimensional schema enriched with design guidelines.
* The *source schema* and the *DW schema*.
* Schema *mappings*, which are used to represent correspondences between the conceptual schema and the source schema.
* A set of *design rules*, which apply the *schema transformations* to the source schema in order to build the DW schema.
* A set of pre-defined *schema transformations* that build new relations from existing ones, applying DW design techniques.
* A *transformation trace*, which keeps the transformations that where applied, providing the mappings between source and DW schemas.

**Logical Design compared with Physical Design**

The process of logical design involves arranging data into a series of logical relationships called entities and attributes.

An [**entity**](http://docs.oracle.com/cd/B10501_01/server.920/a96520/glossary.htm) represents a chunk of information. In relational databases, an entity often maps to a table.

An [**attribute**](http://docs.oracle.com/cd/B10501_01/server.920/a96520/glossary.htm) is a component of an entity that helps define the uniqueness of the entity. In relational databases, an attribute maps to a column.

Relational database model’s structural and data independence enables us to view data *logically* rather than *physically*.

* The logical view allows a simpler file concept of data storage.
* The use of logically independent tables is easier to understand.
* Logical simplicity yields simpler and more effective database design methodologies.

**An entity is a person, place, event, or thing for which we intend to collect data.**

* + - **University** -- Students, Faculty Members, Courses
    - **Airlines** -- Pilots, Aircraft, Routes, Suppliers

**Each entity has certain characteristics known as attributes.**

* + - **Student** -- Student Number, Name, GPA, Date of Enrollment, Data of Birth, Home Address, Phone Number, Major
    - **Aircraft** -- Aircraft Number, Date of Last Maintenance, Total Hours Flown, Hours Flown since Last Maintenance

**A grouping of related entities becomes an entity set.**

* + - The STUDENT entity set contains all student entities.
    - The FACULTY entity set contains all faculty entities.
    - The AIRCRAFT entity set contains all aircraft entities
* A table contains a group of related entities -- i.e. an entity set.
* The terms entity set and table are often used interchangeably.
* A table is also called a relation.
* While entity-relationship diagramming has traditionally been associated with highly normalized models such as OLTP applications, the technique is still useful for data warehouse design in the form of dimensional modeling.
* 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
* The lattice of cuboids forms a data cube.

**Cube: A Lattice of Cuboids**

**Dimensions and Measures**

The database component of a data warehouse is described using a technique called dimensionality modelling.

Every dimensional model (DM) is composed of one table with a composite primary key, called the fact table, and a set of smaller tables called dimension tables.

Each dimension table has a simple (non-composite) primary key that corresponds exactly to one of the components of the composite key in the fact table.

**Fact Tables**

* A fact table is composed of two or more primary keys and usually also contains numeric data. Because it always contains at least two primary keys it is always a M-M relationship.
* Fact tables contain business event details for summarization.
* Because dimension tables contain records that describe facts, the fact table can be reduced to columns for dimension foreign keys and numeric fact values. Text and de-normalized data are typically not stored in the fact table.
* The logical model for a fact table contains a foreign key column for the primary keys of each dimension.
* The combination of these foreign keys defines the primary key for the fact table.
* Fact tables are often very large, containing hundreds of millions of rows and consuming hundreds of gigabytes or multiple terabytes of storage.

**Dimension Tables**

* Dimension tables encapsulate the attributes associated with facts and separate these attributes into logically distinct groupings, such as time, geography, products, customers, and so forth.
* A dimension table may be used in multiple places if the data warehouse contains multiple fact tables or contributes data to data marts.
* The data in a dimension is usually hierarchical in nature.
* Hierarchies are determined by the business need to group and summarize data into usable information.
* For example, a time dimension often contains the hierarchy elements: (all time), Year, Quarter, Month, Day, or (all time), Year Quarter, Week, Day.

**A Data Mining Query Language, DMQL: Language Primitives**

**Cube Definition (Fact Table)**

define cube <cube\_name> [<dimension\_list>]: <measure\_list>

**Dimension Definition ( Dimension Table )**

define dimension <dimension\_name> as (<attribute\_or\_subdimension\_list>)

**Special Case (Shared Dimension Tables)**

* + First time as “cube definition”
  + define dimension <dimension\_name> as <dimension\_name\_first\_time> in cube <cube\_name\_first\_time>

**Data Warehouse Schema**

* A data warehouse, however, requires a concise, subject-oriented schema that facilitates on-line data processing (OLAP).
* The most popular data model for a data warehouse is **a multidimensional model.**
* Such a model can exist in the following forms
  + a star schema
  + a snowflake schema
  + a fact constellation schema.
* The major focus will be on the star schema which is commonly used in the design of many data warehouse.

**Star Schema**

* The star schema is a data modeling technique used to map multidimensional decision support into a relational database.
* Star schemas yield an easily implemented model for multidimensional data analysis while still preserving the relational structure of the operational database.
* Others name: star-join schema, data cube, data list, grid file and multi-dimension schema
* The schema contains a central fact table for *sales* that contains keys to each of the four dimensions, along with two measures: *dollars\_sold, avg\_sales,* and *units\_sold*.
* To minimize the size of the fact table, dimension identifiers (such as *time key* and *item key*) are system-generated identifiers.
* Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes.
* For example, the *location* dimension table contains the attribute set {*location key, street, city, province or state, country*}

Defining a Star Schema in DMQL

define cube sales\_star [time, item, branch, location]:

dollars\_sold = sum(sales\_in\_dollars), avg\_sales = avg(sales\_in\_dollars), units\_sold = count(\*)

define dimension time as (time\_key, day, day\_of\_week, month, quarter, year)

define dimension item as (item\_key, item\_name, brand, type, supplier\_type)

define dimension branch as (branch\_key, branch\_name, branch\_type)

define dimension location as (location\_key, street, city, province\_or\_state, country)

**Advantages of Star Schema**

* Star Schema is very easy to understand, even for non technical business manager.
* Star Schema provides better performance and smaller query times
* Star Schema is easily extensible and will handle future changes easily

**Issues Regarding Star Schema**

* **Dimension table keys must be *surrogate* (non-intelligent and non-business related), because:**
  + Keys may change over time
  + Length/format consistency
* **Granularity of Fact Table–what level of detail do you want?** 
  + Transactional grain–finest level
  + Aggregated grain–more summarized
  + Finer grains 🡺 better *market basket analysis* capability
  + Finer grain 🡺 more dimension tables, more rows in fact table
* **Duration of the database–how much history should be kept?**
  + Natural duration–13 months or 5 quarters
  + Financial institutions may need longer duration
  + Older data is more difficult to source and cleanse

**Snowflake Schema**

A schema is called a *snowflake schema* if one or more dimension tables do not join directly to the fact table but must join through other dimension tables.

It is a variant of star schema model. It has a single, large and central fact table and one or more tables for each dimension.

Characteristics:

* Normalization of dimension tables
* Each hierarchical level has its own table
* less memory space is required
* a lot of joins can be required if they involve attributes in secondary dimension tables

**Defining a Snowflake Schema in DMQL**

define cube sales\_snowflake [time, item, branch, location]:

dollars\_sold = sum(sales\_in\_dollars), avg\_sales = avg(sales\_in\_dollars), units\_sold = count(\*)

define dimension time as (time\_key, day, day\_of\_week, month, quarter, year)

define dimension item as (item\_key, item\_name, brand, type, supplier(supplier\_key, supplier\_type))

define dimension branch as (branch\_key, branch\_name, branch\_type)

define dimension location as (location\_key, street, city(city\_key, province\_or\_state, country))

**Difference between Star Schema and Snow-flake Schema**

* Star Schema is a multi-dimension model where each of its disjoint dimension is represented in single table.
* Snow-flake is normalized multi-dimension schema when each of disjoint dimension is represent in multiple tables.
* Star schema can become a snow-flake
* Both star and snowflake schemas are dimensional models; the difference is in their physical implementations.
* Snowflake schemas support ease of dimension maintenance because they are more normalized.
* Star schemas are easier for direct user access and often support simpler and more efficient queries.
* It may be better to create a star version of the snowflaked dimension for presentation to the users

**Fact-Constellation Schema**

* Multiple fact tables share dimension tables.
* This schema is viewed as collection of stars hence called as galaxy schema or fact constellation.
* Sophisticated application requires such schema.
* In the Fact Constellations, aggregate tables are created separately from the detail, therefore, it is impossible to pick up, for example, Store detail when querying the District Fact Table.
* Fact Constellation is a good alternative to the Star, but when dimensions have very high cardinality, the sub-selects in the dimension tables can be a source of delay.

**Defining a Fact Constellation in DMQL**

define cube sales [time, item, branch, location]:

dollars\_sold = sum(sales\_in\_dollars), avg\_sales = avg(sales\_in\_dollars), units\_sold = count(\*)

define dimension time as (time\_key, day, day\_of\_week, month, quarter, year)

define dimension item as (item\_key, item\_name, brand, type, supplier\_type)

define dimension branch as (branch\_key, branch\_name, branch\_type)

define dimension location as (location\_key, street, city, province\_or\_state, country)

define cube shipping [time, item, shipper, from\_location, to\_location]:

dollar\_cost = sum(cost\_in\_dollars), unit\_shipped = count(\*)

define dimension time as time in cube sales

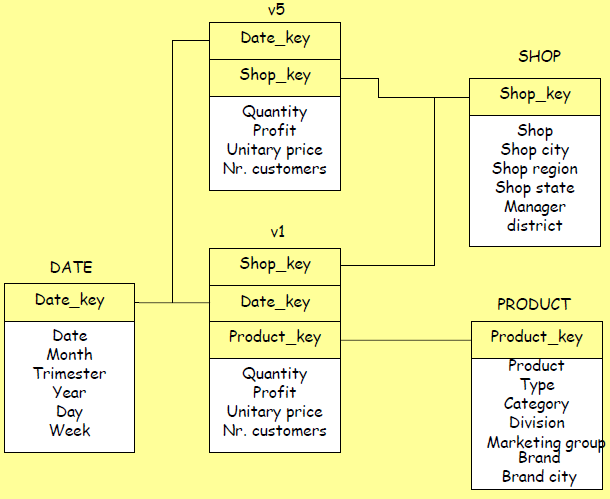
define dimension item as item in cube sales

define dimension shipper as (shipper\_key, shipper\_name, location as location in cube sales, shipper\_type)

define dimension from\_location as location in cube sales

define dimension to\_location as location in cube sales

**More Examples on Fact-Constellation Schema**



**Multidimensional Data Model**

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

*“What is a data cube?”* A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

In general terms, dimensions are the perspectives or entities with respect to which an organization wants to keep records.

Each dimension may have a table associated with it, called a dimension table, which further describes the dimension.

A multidimensional data model is typically organized around a central theme, like *sales*, for instance. This theme is represented by a fact table. Facts are numerical measures.

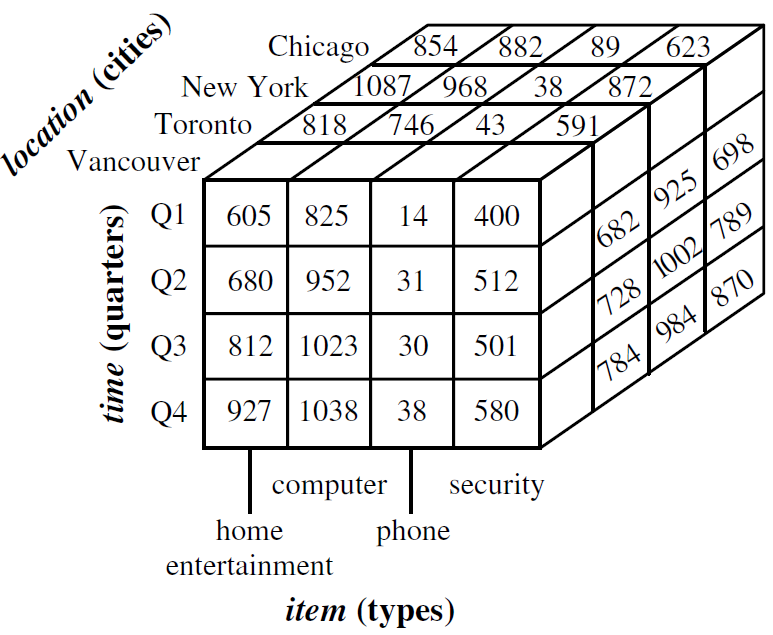


Figure: A 3-D data cube representation of the data in the table above, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars sold* (in thousands).

**Question?**

Suppose that we would now like to view our sales data with an additional fourth dimension, such as *supplier*.

What should we do??

Any Solution???

**Solution!!**

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 below:

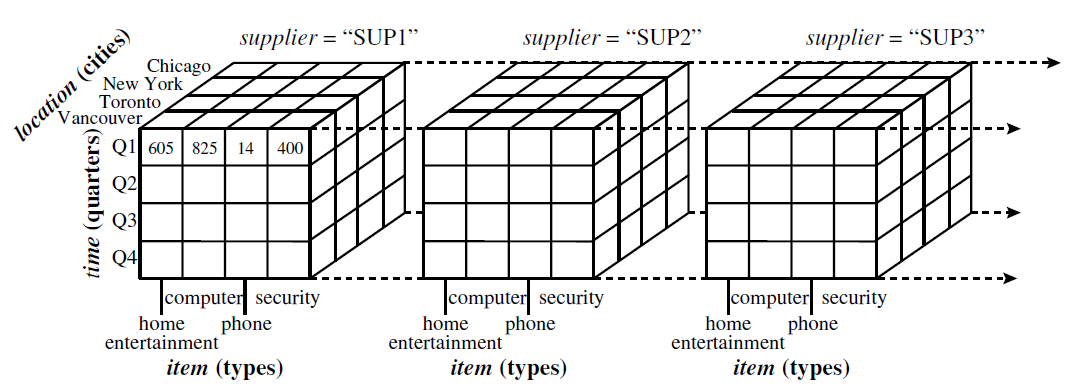


Figure: A 4-D data cube representation of sales data, according to the dimensions *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars sold* (in thousands). For improved readability, only some of the cube values are shown.

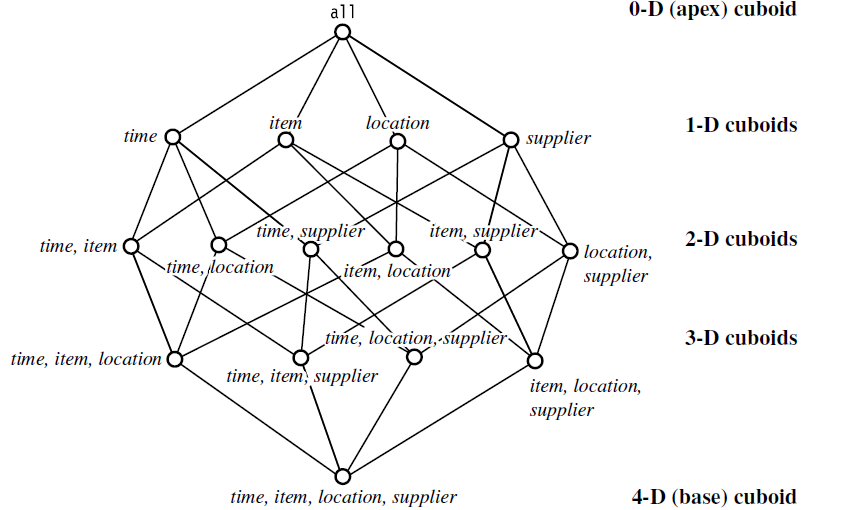


Figure: Lattice of cuboids, making up a 4-D data cube for the dimensions *time*, *item*, *location*, and *supplier*. Each cuboid represents a different degree of summarization.

**Measures: Their Categorization and Computation**

***“How are measures computed?”***

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.

Measures can be organized into three categories (i.e., distributive, algebraic, holistic), based on the kind of aggregate functions used.

**Distributive Measure**

* A measure is *distributive* if it is obtained by applying a distributive aggregate function.
* An aggregate function is *distributive* if it can be computed in a distributed manner.
* Example: count(), sum(), min(), max().

**Algebraic Measure**

* A measure is *algebraic* if it is obtained by applying an algebraic aggregate function.
* An aggregate function is *algebraic* if it can be computed by an algebraic function.
* Example: avg(), min\_N(), standard\_deviation().

**Holistic Measure**

* A measure is *holistic* if it is obtained by applying a holistic aggregate function.
* An aggregate function is *holistic* if there is no constant bound on the storage size needed to describe a sub-aggregate.
* Example: median(), mode(), rank().

**Concept Hierarchies**

* A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.

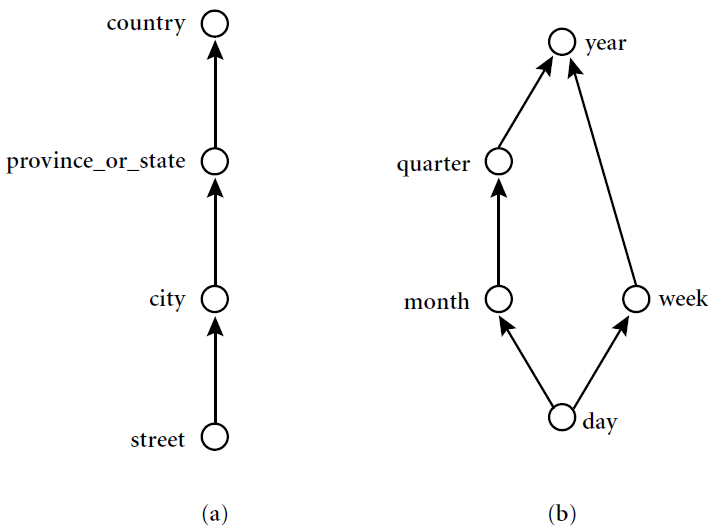


**Figure:** A concept hierarchy for the dimension *location*.

In the above figure we have considered 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.

For example, Vancouver can be mapped to British Columbia, and Chicago to Illinois. The provinces and states can in turn be mapped to the country to which they belong, such as Canada or the USA.

These mappings form a concept hierarchy for the dimension *location*, mapping a set of low-level concepts (i.e., cities) to higher-level, more general concepts (i.e., countries).



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

Many concept hierarchies are implicit within the database schema. For example, suppose that the 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*”. This hierarchy is shown in the figure (a) above.

Many concept hierarchies are implicit within the database schema. For example, suppose that the 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*”. This hierarchy is shown in the figure (a) above.

Alternatively, the attributes of a dimension may be organized in a partial order, forming a lattice.

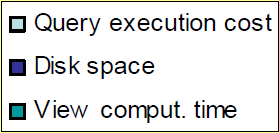
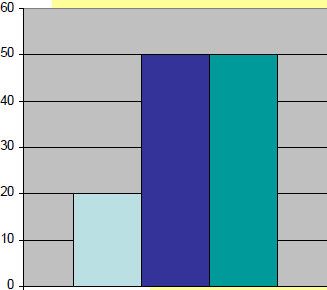
An example of a partial order for the *time* dimension based on the attributes *day, week, month, quarter*, and *year* is “*day* < {*month* <*quarter; week*} < *year*”.

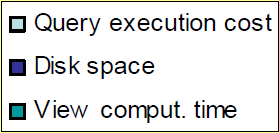
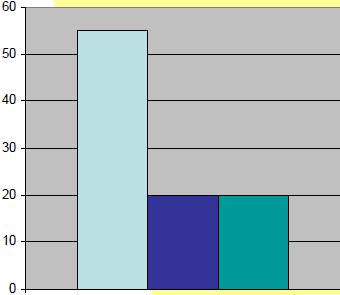
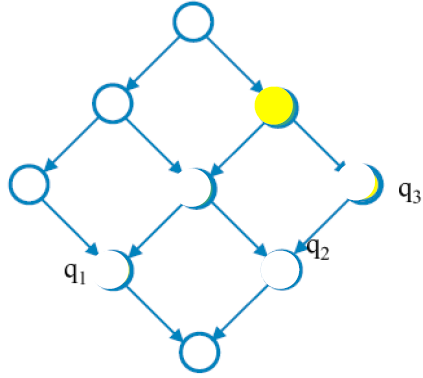
This lattice structure is shown in the figure (b) as above.

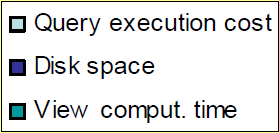
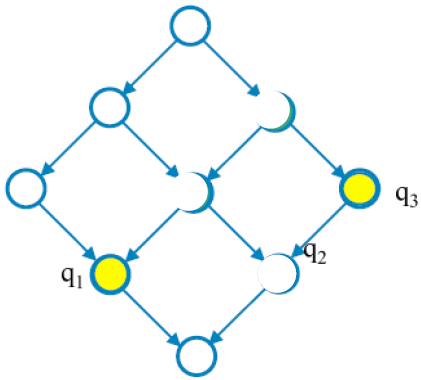
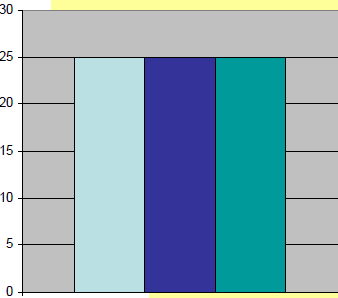
**Materialized View**

Materialized views are query results that have been stored in advance so long-running calculations are not necessary when you actually execute your SQL statements.

Materialized views can be best explained by Multidimensional lattice.







It is useful to materialize a view when:

* It directly solves a frequent query
* It reduce the costs of some queries

It is not useful to materialize a view when:

* Its aggregation pattern is the same as another materialized view
* Its materialization does not reduce the cost

**Class Assignment (1)**

Suppose that a data warehouse consists of the three dimensions *time, doctor*, and *patient*, and the two measures *count* and *charge*, where *charge* is the fee that a doctor charges a patient for a visit.

1. Enumerate three classes of schemas that are popularly used for modeling data warehouses.

(b) Draw a schema diagram for the above data warehouse using one of the schema classes listed in (a).

**Class Assignment (2)**

Suppose that a data warehouse for *XYZ University* consists of the following four dimensions: *student, course, semester*, and *instructor*, and two measures *count* and *avg grade*.

When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the *avg grade* measure stores the actual course grade of the student. At higher conceptual levels, *avg grade* stores the average grade for the given combination.

(a) Draw a *snowflake schema* diagram for the data warehouse.

**Class Assignment (3)**

A data warehouse can be modeled by either a *star schema* or a *snowflake schema*. Briefly describe the similarities and the differences of the two models, and then analyze their advantages and disadvantages with regard to one another.

Give your opinion of which might be more empirically useful and state the reasons behind your answer.

**Class Assignment (4)**

Let us consider the case of a real estate agency whose database is composed by the following tables:

OWNER (IDOwner, Name, Surname, Address, City, Phone)

ESTATE (IDEstate, IDOwner, Category, Area, City, Province, Rooms, Bedrooms, Garage, Meters)

CUSTOMER (IDCust, Name, Surname, Budget, Address, City, Phone)

AGENT (IDAgent, Name, Surname, Office, Address, City, Phone)

AGENDA (IDAgent, Data, Hour, IDEstate, ClientName)

VISIT (IDEstate, IDAgent, IDCust, Date, Duration)

SALE (IDEstate, IDAgent, IDCust, Date, AgreedPrice, Status)

RENT (IDEstate, IDAgent, IDCust, Date, Price, Status, Time)

Design a Star Schema or Snowflake Schema for the DW.

**Hints**

The following ideas will be used during the solution of the exercise:

* supervisors should be able to control the sales of the agency

FACT Sales

MEASURES OfferPrice, AgreedPrice, Status

DIMENSIONS EstateID, OwnerID, CustomerID, AgentID,

TimeID

* supervisors should be able to control the work of the agents by analyzing the visits to the estates, which the agents are in charge of

FACT Viewing

MEASURES Duration

DIMENSIONS EstateID, CustomerID, AgentID, TimeID

**Solution for Class Assignment (4)**

**Class Assignment (5)**

An online order wine company requires the designing of a data warehouse to record the quantity and sales of its wines to its customers. Part of the original database is composed by the following tables:

CUSTOMER (Code, Name, Address, Phone, BDay, Gender)

WINE (Code, Name, Type, Vintage, BottlePrice, CasePrice, Class)

CLASS (Code, Name, Region)

TIME (TimeStamp, Date, Year)

ORDER (Customer, Wine, Time, nrBottles, nrCases)

Note that the tables represent the main entities of the ER schema, thus it is necessary to derive the significant relationships among them in order to correctly design the data warehouse.

Construct Snowflake Schema.

**Solution for Class Assignment (5)**

**Unit 3 : Data Warehouse Physical Design**

**Physical Design**

Physical design is the phase of a database design following the logical design that identifies the actual database tables and index structures used to implement the logical design.

In the physical design, you look at the most effective way of storing and retrieving the objects as well as handling them from a transportation and backup/recovery perspective.

Physical design decisions are mainly driven by query performance and database maintenance aspects.

During the logical design phase, you defined a model for your data warehouse consisting of entities, attributes, and relationships. The entities are linked together using relationships. Attributes are used to describe the entities. The unique identifier (UID) distinguishes between one instance of an entity and another.

During the physical design process, you translate the expected schemas into actual database structures.

At this time, you have to map:

■ Entities to tables

■ Relationships to foreign key constraints

■ Attributes to columns

■ Primary unique identifiers to primary key constraints

■ Unique identifiers to unique key constraints

**Physical Data Model**

Features of physical data model include:

* Specification all tables and columns.
* Specification of Foreign keys.
* De-normalization may be performed if necessary.
* At this level, specification of logical data model is realized in the database.

The steps for physical data model design involves:

* Conversion of entities into tables,
* Conversion of relationships into foreign keys, Conversion of attributes into columns, and
* Changes to the physical data model based on the physical constraints.

**Physical Design Objectives**

* Involves tradeoffs among
  + Performance
  + Flexibility
  + Scalability
  + Ease of Administration
  + Data Integrity
  + Data Consistency
  + Data Availability
  + User Satisfaction

**Performance**

* Response time in DW typically > OLTP
  + Important to manage user expectations
* Poor performance may result from
  + Inadequate hardware
  + Inflexible data architecture
  + Poor physical design
  + Unrealistic user expectations
* Build performance bottom-up
  + Database Design and Optimization
  + Application design
  + Query efficiency
* Tune performance from top-down

**Flexibility**

* May include giving users flexibility to handle analysis, query, reporting needs
* Must accommodate change in today’s business environment

**Scalability**

* Old mainframes known for poor scalability
* Many adopt multi-server environment

**Physical Design Structures**

Once you have converted your logical design to a physical one, you will need to create some or all of the following structures:

■ Tablespaces

■ Tables and Partitioned Tables

■ Views

■ Integrity Constraints

■ Dimensions

Some of these structures require disk space. Others exist only in the data dictionary. Additionally, the following structures may be created for performance improvement:

■ Indexes and Partitioned Indexes

■ Materialized Views

**Tablespaces**

* A tablespace consists of one or more datafiles, which are physical structures within the operating system you are using.
* A datafile is associated with only one tablespace.
* From a design perspective, tablespaces are containers for physical design structures.

**Tables and Partitioned Tables**

* Tables are the basic unit of data storage. They are the container for the expected amount of raw data in your data warehouse.
* Using partitioned tables instead of non-partitioned ones addresses the key problem of supporting very large data volumes by allowing you to divide them into smaller and more manageable pieces.
* Partitioning large tables improves performance because each partitioned piece is more manageable.

**Views**

* A view is a tailored presentation of the data contained in one or more tables or other views.
* A view takes the output of a query and treats it as a table.
* Views do not require any space in the database.

**Integrity Constraints**

* Integrity constraints are used to enforce business rules associated with your database and to prevent having invalid information in the tables.
* In data warehousing environments, constraints are only used for query rewrite.
* NOT NULL constraints are particularly common in data warehouses.

**Indexes and Partitioned Indexes**

* Indexes are optional structures associated with tables.
* Indexes are just like tables in that you can partition them (but the partitioning strategy is not dependent upon the table structure)
* Partitioning indexes makes it easier to manage the data warehouse during refresh and improves query performance.

**Materialized Views**

* Materialized views are query results that have been stored in advance so long-running calculations are not necessary when you actually execute your SQL statements.
* From a physical design point of view, materialized views resemble tables or partitioned tables and behave like indexes in that they are used transparently and improve performance.

**Hardware and I/O Consideration**

* I/O performance should always be a key consideration for data warehouse designers and administrators.
* The typical workload in a data warehouse is especially I/O intensive, with operations such as large data loads and index builds, creation of materialized views, and queries over large volumes of data.
* The underlying I/O system for a data warehouse should be designed to meet these heavy requirements.
* In fact, one of the leading causes of performance issues in a data warehouse is poor I/O configuration.
* Database administrators who have previously managed other systems will likely need to pay more careful attention to the I/O configuration for a data warehouse than they may have previously done for other environments.
* The I/O configuration used by a data warehouse will depend on the characteristics of the specific storage and server capabilities
* There are following five high-level guidelines for data-warehouse I/O configurations:
* ■ Configure I/O for Bandwidth not Capacity
* ■ Stripe Far and Wide
* ■ Use Redundancy
* ■ Test the I/O System Before Building the Database
* ■ Plan for Growth

**Configure I/O for Bandwidth not Capacity**

* Storage configurations for a data warehouse should be chosen based on the I/O bandwidth that they can provide, and not necessarily on their overall storage capacity.
* Buying storage based solely on capacity has the potential for making a mistake, especially for systems less than 500GB is total size.
* The capacity of individual disk drives is growing faster than the I/O throughput rates provided by those disks, leading to a situation in which a small number of disks can store a large volume of data, but cannot provide the same I/O throughput as a larger number of small disks.
* While it may not be practical to estimate the I/O bandwidth that will be required by a data warehouse before a system is built, it is generally practical with the guidance of the hardware manufacturer to estimate how much I/O bandwidth a given server can potentially utilize, and ensure that the selected I/O configuration will be able to successfully feed the server.

**Conclusion:**

* There are many variables in sizing the I/O systems, but one basic rule of thumb is that your data warehouse system should have multiple disks for each CPU (at least two disks for each CPU at a bare minimum) in order to achieve optimal performance.

**Stripe Far and Wide**

* The guiding principle in configuring an I/O system for a data warehouse is to maximize I/O bandwidth by having multiple disks and channels access each database object.
* A striped file is a file distributed across multiple disks. This striping can be managed by software (such as a logical volume manager), or within the storage hardware.
* The goal is to ensure that each tablespace is striped across a large number of disks so that any database object can be accessed with the highest possible I/O bandwidth.

**Use Redundancy**

* Because data warehouses are often the largest database systems in a company, they have the most disks and thus are also the most susceptible to the failure of a single disk.
* Therefore, disk redundancy is a requirement for data warehouses to protect against a hardware failure.
* Like disk-striping, redundancy can be achieved in many ways using software or hardware.
* A key consideration is that occasionally a balance must be made between redundancy and performance.
* For example, a storage system like RAID configuration and its variants may be used.
* Redundancy is necessary for any data warehouse, but the approach to redundancy may vary depending upon the performance and cost constraints of each data warehouse.

**Test the I/O System Before Building the Database**

* The most important time to examine and tune the I/O system is before the database is even created.
* Once the database files are created, it is more difficult to reconfigure the files.
* When creating a data warehouse on a new system, the I/O bandwidth should be tested before creating all of the database datafiles to validate that the expected I/O levels are being achieved.

**Plan for Growth**

* A data warehouse designer should plan for future growth of a data warehouse.
* There are many approaches to handling the growth in a system, and the key consideration is to be able to grow the I/O system without compromising on the I/O bandwidth.

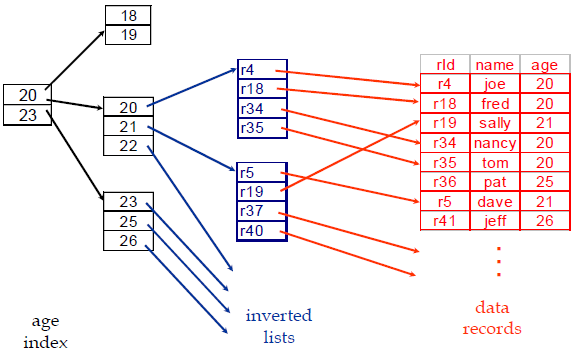
**Parallelism**

* Parallelism is the idea of breaking down a task so that, instead of one process doing all of the work in a query, many processes do part of the work at the same time.
* Parallel execution is sometimes called parallelism.
* Parallel execution dramatically reduces response time for data-intensive operations on large databases typically associated with decision support systems (DSS) and data warehouses.
* An example of this is when four processes handle four different quarters in a year instead of one process handling all four quarters by itself.
* Parallelism improves processing for:
* ■ Queries requiring large table scans, joins, or partitioned index scans
* ■ Creation of large indexes
* ■ Creation of large tables (including materialized views)
* ■ Bulk inserts, updates, merges, and deletes
* Parallelism benefits systems with all of the following characteristics:
* ■ Symmetric multiprocessors (SMPs), clusters, or massively parallel systems
* ■ Sufficient I/O bandwidth
* ■ Underutilized or intermittently used CPUs (for example, systems where CPU usage is typically less than 30%)
* ■ Sufficient memory to support additional memory-intensive processes, such as sorts, hashing, and I/O buffers

**Indexes**

* Indexes are optional structures associated with tables and clusters.
* Indexes are structures actually stored in the database, which users create, alter, and drop using SQL statements.
* You can create indexes on one or more columns of a table to speed SQL statement execution on that table.
* In a query-centric system like the data warehouse environment, the need to process queries faster dominates.
* Among the various methods to improve performance, indexing ranks very high.
* Indexesare typically used to speed up the retrieval of records in response to search conditions.
* **Indexes** can be unique or non-unique.
* Unique indexes guarantee that no two rows of a table have duplicate values in the key column (or columns).
* Non-unique indexes do not impose this restriction on the column values.
* Index structures applied in warehouses are:
  + Inverted lists
  + Bitmap indexes
  + Join indexes
  + Text indexes
  + B-Tree Index

**Inverted Lists**

**Query:** Get people with age = 20 and name = “fred”

List for age = 20: r4, r18, r34, r35

List for name = “fred”: r18, r52

**Answer is intersection: r18**

**Bitmap Indexes**

* The concept of bitmap index was first introduced by Professor Israel Spiegler and Rafi Maayan in their research "Storage and Retrieval Considerations of Binary Data Bases", published in 1985.
* A bitmap index is a special kind of database index that uses bitmaps and are used widely in multi-dimensional database implementation.
* Bitmap indexes are primarily intended for data warehousing applications where users query the data rather than update it.
* They are not suitable for OLTP applications with large numbers of concurrent transactions modifying the data.
* Bitmap indexes use bit arrays (commonly called bitmaps) and answer queries by performing bitwise logical operations on these bitmaps.
* In a bitmap index, a bitmap for each key value replaces a list of row ids.
* Each bit in the bitmap corresponds to a possible rowid, and if the bit is set, it means that the row with the corresponding rowid contains the key value.
* Each value in the indexed column has a bit vector (bitmaps).
* The length of the bit vector is the number of records in the base table.
* The i-th bit is set if the i-th row of the base table has the value for the indexed column.
* With efficient hardware support for bitmap operations (AND, OR, XOR, NOT), bitmap index offers better access methods for certain queries.

**Conclusion**:

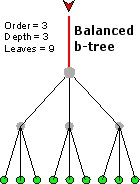
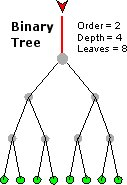
* Bitmap indexes are useful in data warehousing applications for joining a large fact table to smaller dimension tables such as those arranged in a star schema.

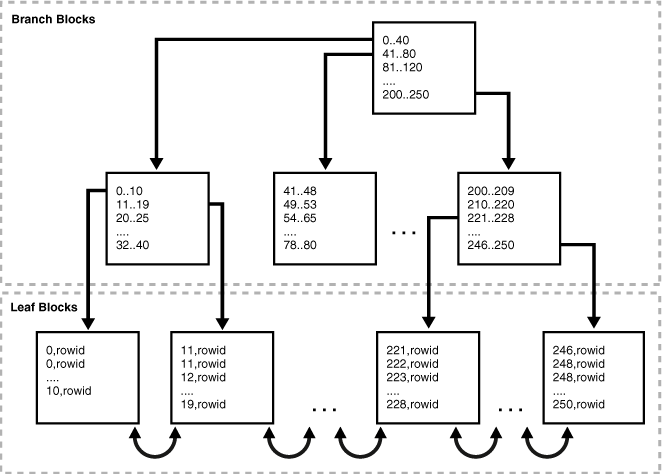
**Join Indexes**

* Join indexes map the tuples in the join result of two relations to the source tables.
* In data warehouse cases, join indexes relate the values of the dimensions of a star schema to rows in the fact table.
  + For a warehouse with a Sales fact table and dimension city, a join index on city maintains for each distinct city a list of RIDs of the tuples recording the sales in the city
* Join indexes can span multiple dimensions

**B-Tree Index**

* B-trees, short for **balanced trees**, are the most common type and default database index.
* A **B-tree** is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions.



* The B-tree is a generalization of a binary search tree in that a node can have more than two children.
* Figure below shows an example of a B-Tree Index.
* Example: A B-Tree IndexA B-tree index has two types of blocks: **branch blocks** for searching and **leaf blocks** that store values.
* The upper-level branch blocks of a B-tree index contain index data that points to lower-level index blocks.
* In above figure, the root branch block has an entry 0-40, which points to the leftmost block in the next branch level.
* This branch block contains entries such as 0-10 and 11-19.
* Each of these entries points to a leaf block that contains key values that fall in the range.
* Branch blocks store the minimum key prefix needed to make a branching decision between two keys.
* The branch blocks contain a pointer to the child block containing the key.
* The leaf blocks contain every indexed data value and a corresponding rowid used to locate the actual row.
* The leaf blocks themselves are also doubly linked.
* In above figure, the leftmost leaf block (0-10) is linked to the second leaf block (11-19).
* Notice the tree structure with the root at the top.
* The index consists of a B-Tree (a balanced binary tree) structure based on the values of the indexed column.
* Suppose we have to search value 25 in an indexed column, the query engine will first look in the “Root Node” to determine which node to refer in the “Branch Nodes”.
* In the above example first “Branch Node” has Value 1 to 20 and the second “Branch Node” has Value 21 to 40, so the query engine will go to the second “Branch Node” and will skip the first “Branch Node” as we have to search Value 25.

Conclusion:

* B-tree indexes are created to decrease the amount of I/O required to find and load a set of data.

**Assignment**

1. In your opinion what may be the other factors for hardware and i/o consideration while making Data Warehouse.
2. Discuss about parallelism and parallel computing. Mention and explain some of the parallelism technique that could be adopted.
3. Give some of the suitable examples for the various indexing schemes.

**Unit 4 : Data Warehousing Technologies and Implementation**

**Design of a Data Warehouse: A Business Analysis Framework**

Four views regarding the design of a data warehouse

* + Top-down view
    - allows selection of the relevant information necessary for the data warehouse
  + Data source view
    - exposes the information being captured, stored, and managed by operational systems
  + Data warehouse view
    - consists of fact tables and dimension tables
  + Business query view
    - sees the perspectives of data in the warehouse from the view of end-user

**Data Warehouse Design Process**

* Top-down, bottom-up approaches or a combination of both
  + **Top-down:** Starts with overall design and planning (mature)
  + **Bottom-up:** Starts with experiments and prototypes (rapid)
* From software engineering point of view
  + **Waterfall:** structured and systematic analysis at each step before proceeding to the next
  + **Spiral:** rapid generation of increasingly functional systems, short turn around time, quick turn around
* Typical data warehouse design process
  + Choose a business process to model, e.g., orders, invoices, etc.
  + Choose the *grain* (*atomic level of data*) of the business process
  + Choose the dimensions that will apply to each fact table record
  + Choose the measure that will populate each fact table record

**Design of a Data Warehouse: Three Data Warehouse Models**

* **Enterprise warehouse** 
  + collects all of the information about subjects spanning the entire organization
  + top down approach
  + the W. Inmon methodology
* **Data Mart** 
  + a subset of corporate-wide data that is of value to a 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
    - bottom up approach
    - the R. Kimball methodology
* **Virtual warehouse**
  + A set of views over operational databases
  + Only some of the possible summary views may be materialized

**The Data Mart Strategy**

* The most common approach
* Begins with a single mart and architected marts are added over time for more subject areas
* Relatively inexpensive and easy to implement
* Can be used as a proof of concept for data warehousing
* Can postpone difficult decisions and activities
* Requires an overall integration plan
* The key is to have an overall plan, processes, and technologies for integrating the different marts.
* The marts may be logically rather than physically separate.

**Enterprise Warehouse Strategy**

* A comprehensive warehouse is built initially
* An initial dependent data mart is built using a subset of the data in the warehouse
* Additional data marts are built using subsets of the data in the warehouse
* Like all complex projects, it is expensive, time consuming, and prone to failure
* When successful, it results in an integrated, scalable warehouse
* Even with the enterprise-wide strategy, the warehouse is developed in phases and each phase should be designed to deliver business value.

**Data Warehouse Development: A Recommended Approach**

**Extract, Transform and Load (ETL) Definition**

**Three separate functions combined into one development tool:**

1. Extract - Reads data from a specified source and extracts a desired subset of data.
2. Transform - Uses rules or lookup tables, or creating combinations with other data, to convert source data to the desired state.
3. Load - Writes the resulting data to a target database

**ETL Overview**

* **ETL, Short for *extract, transform, and load a*re the database functions that are combined into one tool.**
* **ETL is used to migrate data from one database to another, to form data marts and data warehouses and also to convert databases from one format or type to another.**
* **To get data out of the source and load it into the data warehouse – simply a process of copying data from one database to other**
* **Data is extracted from an OLTP database, transformed to match the data warehouse schema and loaded into the data warehouse database**
* **Many data warehouses also incorporate data from non-OLTP systems such as text files, legacy systems, and spreadsheets; such data also requires extraction, transformation, and loading**

**The ETL Cycle**

**ETL Processing**

ETL is independent yet interrelated steps.

It is important to look at the big picture.

Data acquisition time may include…

* ETL is often a complex combination of process and technology that consumes a significant portion of the data warehouse development efforts and requires the skills of business analysts, database designers, and application developers
* It is not a one time event as new data is added to the Data Warehouse periodically – i.e. monthly, daily, hourly
* Because ETL is an integral, ongoing, and recurring part of a data warehouse. It may be:
  + Automated
  + Well documented
  + Easily changeable
* When defining ETL for a data warehouse, it is important to think of ETL as a process, not a physical implementation

**Extraction, Transformation, and Loading (ETL) Processes**

**Data Extraction**

**Data Cleansing**

**Data Transformation**

**Data Loading**

**Data Refreshing**

**Data Extraction**

**Capture/Extract…obtaining a snapshot of a chosen subset of the source data for loading into the data warehouse**

* **ETL process needs to effectively integrate systems that have different:**
  + **DBMS**
  + **Operating Systems**
  + **Hardware**
  + **Communication protocols**
  + **Need to have a logical data map before the physical data can be transformed**
* **The logical data map describes the relationship between the extreme starting points and the extreme ending points of your ETL system usually presented in a table or spreadsheet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Target** | | | **Source** | | | **Transformation** |
| **Table Name** | **Column Name** | **Data Type** | **Table Name** | **Column Name** | **Data Type** |  |

* **The content of the logical data mapping document has been proven to be the critical element required to efficiently plan ETL processes.**
* **The table type gives us our queue for the ordinal position of our data load processes—first dimensions, then facts.**
* **This table must depict, without question, the course of action involved in the transformation process**
* **The transformation can contain anything from the absolute solution to nothing at all. Most often, the transformation can be expressed in SQL. The SQL may or may not be the complete statement**

**Some ETL Tools**

|  |  |
| --- | --- |
| **Tool** | **Vendor** |
| **Oracle Warehouse Builder (OWB)** | **Oracle** |
| **Data Integrator (BODI)** | **Business Objects** |
| **IBM Information Server (Ascential)** | **IBM** |
| **SAS Data Integration Studio** | **SAS Institute** |
| **PowerCenter** | **Informatica** |
| **Oracle Data Integrator (Sunopsis)** | **Oracle** |
| **Data Migrator** | **Information Builders** |
| **Integration Services** | **Microsoft** |
| **Talend Open Studio** | **Talend** |
| **DataFlow** | **Group 1 Software (Sagent)** |
| **Data Integrator** | **Pervasive** |
| **Transformation Server** | **DataMirror** |
| **Transformation Manager** | **ETL Solutions Ltd.** |
| **Data Manager** | **Cognos** |
| **DT/Studio** | **Embarcadero Technologies** |
| **ETL4ALL** | **IKAN** |
| **DB2 Warehouse Edition** | **IBM** |
| **Jitterbit** | **Jitterbit** |
| **Pentaho Data Integration** | **Pentaho** |

**Data Cleansing**

* **Data Warehouse is NOT just about arranging data, but should be clean for overall health of organization. “We drink clean water”!**
* **Sometime called as Data Scrubbing or Cleaning.**
* **ETL software contains rudimentary data cleansing capabilities**
* **Specialized data cleansing software is often used. Leading data cleansing vendors include Vality (Integrity), Harte-Hanks (Trillium), and Firstlogic (i.d.Centric)**

**Why Cleansing?**

* + **Data warehouse contains data that is analyzed for business decisions**
  + **Source systems contain “dirty data” that must be cleansed.**
  + **More data and multiple sources could mean more errors in the data and harder to trace such errors**
  + **Results in incorrect analysis**
  + **Enormous problem, as most data is dirty. (GIGO)**

**Reasons for “Dirty” Data**

* **Dummy Values**
* **Absence of Data**
* **Multipurpose Fields**
* **Cryptic Data**
* **Contradicting Data**
* **Inappropriate Use of Address Lines**
* **Violation of Business Rules**
* **Reused Primary Keys,**
* **Non-Unique Identifiers**
* **Data Integration Problems**

**Examples:**

* **Dummy Data Problem:**

**A clerk enters 999-99-9999 as a SSN rather than asking the customer for theirs.**

* **Reused Primary Keys:**

**A branch bank is closed. Several years later, a new branch is opened, and the old identifier is used again.**

**Inconsistent Data RepresentationsSame data, different representation**

**Date value representations  
Examples:  
 970314 1997-03-14  
 03/14/1997 14-MAR-1997**

**March 14 1997 2450521.5 (Julian date format)**

**Gender value representations  
Examples:  
 - Male/Female - M/F  
 - 0/1**

**Scrub/Cleanse…uses pattern recognition and AI techniques to upgrade data quality**

**Two Classes of Anomalies**

* **Coverage Problems**
  + **Missing values**
  + **Missing Tuples or records**
* **Key-based classification problems**
  + **Primary key problems**
  + **Non-Primary key problems**

**1. Coverage Problems**

* + **Missing Attribute**

**Result of omissions while collecting the data.**

**A constraint violation if we have null values for attributes where NOT NULL constraint exists.**

**Case more complicated where no such constraint exists.**

**Have to decide whether the value exists in the real world and has to be deduced here or not.**

* + **Why Missing Rows/Value?** **Equipment malfunction (bar code reader, keyboard etc.)**
  + **Inconsistent with other recorded data and thus deleted.**
  + **Data not entered due to misunderstanding/illegibility.**
  + **Data not considered important at the time of entry (e.g. Y2K).**
* **Handling missing dataDropping records.**
* **“Manually” filling missing values.**
* **Using a global constant as filler.**
* **Using the attribute mean (or median) as filler.**
* **Using the most probable value as filler.**

**2. Key-based Classification Problems**

**Primary key problems**

* + **Same PK but different data.**
  + **Same entity with different keys.**
  + **PK in one system but not in other.**
  + **Same PK but in different formats.**

**Non primary key problems**

* + **Different encoding in different sources.**
  + **Multiple ways to represent the same information.**
  + **Sources might contain invalid data.**
  + **Two fields with different data but same name.**
  + **Required fields left blank.**
  + **Data erroneous or incomplete.**
  + **Data contains null values.**

**Data Quality paradigm**

* **Correct**
* **Unambiguous**
* **Consistent**
* **Complete**

**Data quality checks are run at 2 places - after extraction and after cleaning and confirming additional check are run at this point**

* **Steps in Data CleansingParsing**
* **Correcting**
* **Standardizing**
* **Matching**
* **Consolidating**

**Data Quality paradigm**

* Correct
* Unambiguous
* Consistent
* Complete

Data quality checks are run at 2 places - after extraction and after cleaning and confirming additional check are run at this point

**Steps in Data Cleansing**

* Parsing
* Correcting
* Standardizing
* Matching
* Consolidating

**Parsing**

* The record is broken down into atomic data elements.
* Parsing locates and identifies individual data elements in the source files and then isolates these data elements in the target files.
* Examples include parsing the first, middle, and last name; street number and street name; and city and state.

**Correcting**

* External data, such as census data, is often used in this process.
* Corrects parsed individual data components using sophisticated data algorithms and secondary data sources.
* Example include replacing a vanity address and adding a zip code.

**Standardizing**

* Companies decide on the standards that they want to use.
* Standardizing applies conversion routines to transform data into its preferred (and consistent) format using both standard and custom business rules.
* Examples include adding a pre name, replacing a nickname, and using a preferred street name.

**Matching**

* Commercial data cleansing software often uses AI techniques to match records.
* Searching and matching records within and across the parsed, corrected and standardized data based on predefined business rules to eliminate duplications.
* Examples include identifying similar names and addresses.

**Consolidating**

* All of the data are now combined in a standard format.
* Analyzing and identifying relationships between matched records and consolidating/merging them into ONE representation.

**Data Staging**

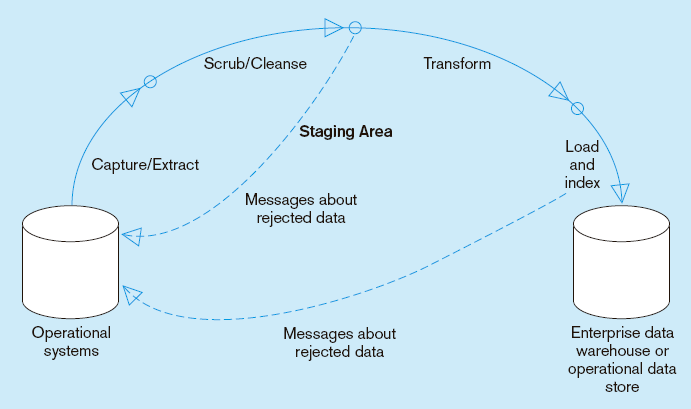
* Data staging is used in cleansing, transforming, and integrating the data.
* Often used as an interim step between data extraction and later steps
* Accumulates data from asynchronous sources using native interfaces, flat files, FTP sessions, or other processes
* At a predefined cutoff time, data in the staging file is transformed and loaded to the warehouse
* There is usually no end user access to the staging file
* An operational data store may be used for data staging

**Data Transformation**

**Data Transformation**

* It is the main step where the ETL adds value.
* Actually changes data and provides guidance whether data can be used for its intended purposes.
* Performed in staging area.

**Transform =** convert data from format of operational system to format of data warehouse



**Basic Tasks**

* 1. Selection
  2. Splitting/Joining
  3. Conversion
  4. Summarization
  5. Enrichment

**Data Transformation : Conversion**

* Convert common data elements into a consistent form i.e. name and address.

**Field format Field data**

First-Family-title Bijay Mishra, Lecturer

Family-title-comma-first Mishra Lecturer, Bijay

Family-comma-first-title Mishra, Bijay Lecturer

* Translation of dissimilar codes into a standard code.
* Data representation change
  + EBCIDIC to ASCII
* Operating System Change
  + Mainframe (MVS) to UNIX
  + UNIX to NT or XP
* Data type change
  + Program (Excel to Access), database format (FoxPro to Access).
  + Character, numeric and date type.
  + Fixed and variable length.

**Data Transformation : Summarization**

Values are summarized to obtain total figures which are subsequently calculated and stored at multiple levels as business fact in multidimensional fact tables.

**Data Transformation : Enrichment**

* Data elements are mapped from source tables and files to destination fact and dimension tables.
* Default values are used in the absence of source data.
* Fields are added for unique keys and time elements.

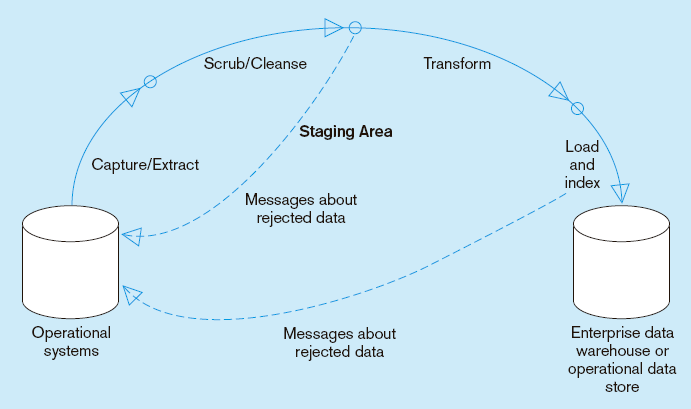
**Transformation – Confirming**

* Structure Enforcement
  + Tables have proper primary and foreign keys
  + Obey referential integrity
  + Data and Rule value enforcement
  + Simple business rules
  + Logical data checks

**Data Loading**

* Most loads involve only change data rather than a bulk reloading of all of the data in the warehouse.
* Data are physically moved to the data warehouse
* The loading takes place within a “load window”
* The trend is to near real time updates of the data warehouse as the warehouse is increasingly used for operational applications

**Load/Index=** place transformed data into the warehouse and create indexes



* The loading process can be broken down into 2 different types:
  + Initial Load
  + Continuous Load (loading over time)

**Initial Load**

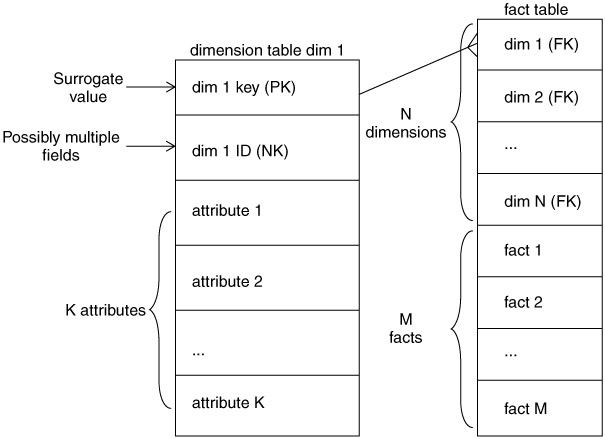
* Consists of populating tables in warehouse schema and verifying data readiness
* Examples:
  + DTS – data transformation services
  + Backup utility – batch copy
  + SQL\*Loader
  + Native Database Languages (T-SQL, PL/SQL, etc.)

**Continuous Loads**

* Must be scheduled and processed in a specific order to maintain integrity, completeness, and a satisfactory level of trust
* Should be the most carefully planned step in data warehousing or can lead to:
  + Error duplication
  + Exaggeration of inconsistencies in data
* Must be during a fixed batch window (usually overnight)
* Must maximize system resources to load data efficiently in allotted time
  + Ex. *Red Brick Loader* can validate, load, and index up to 12GB of data per hour on an SMP system

**Loading Dimensions**

* Physically built to have the minimal sets of components
* The primary key is a single field containing meaningless unique integer – Surrogate Keys
* The DW owns these keys and never allows any other entity to assign them
* De-normalized flat tables – all attributes in a dimension must take on a single value in the presence of a dimension primary key.
* Should possess one or more other fields that compose the natural key of the dimension



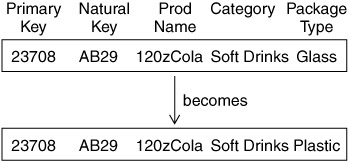
* The data loading module consists of all the steps required to administer slowly changing dimensions (SCD) and write the dimension to disk as a physical table in the proper dimensional format with correct primary keys, correct natural keys, and final descriptive attributes.
* Creating and assigning the surrogate keys occur in this module.
* The table is definitely staged, since it is the object to be loaded into the presentation system of the data warehouse.
* When DW receives notification that an existing row in dimension has changed it gives out 3 types of responses:

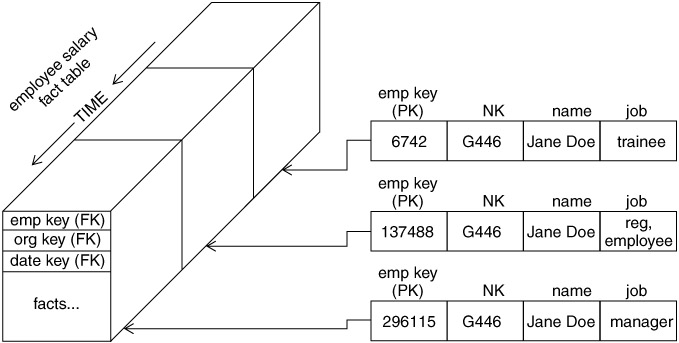
Type 1

Type 2

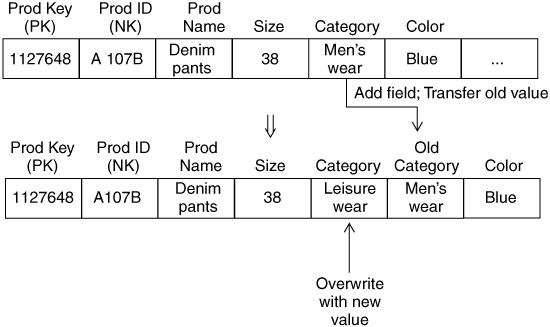
Type 3

**Type 1 Dimension**



**Type 2 Dimension**

**Type 3 Dimensions**

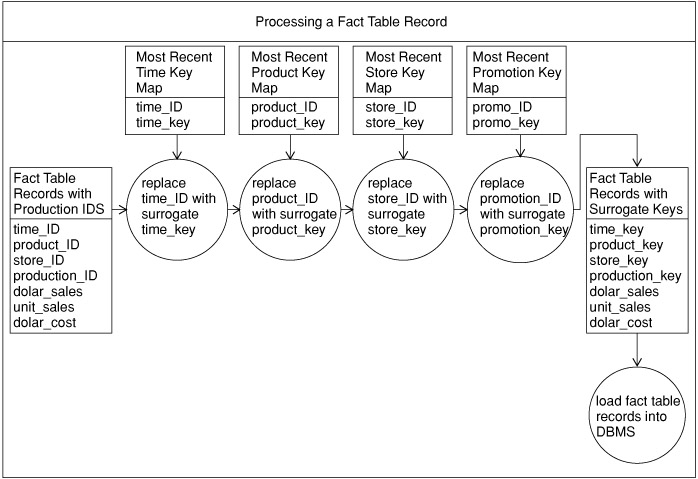


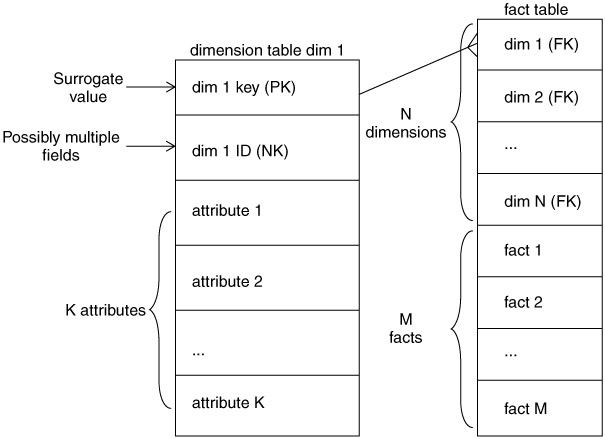
**Loading Facts**

* Fact tables hold the measurements of an enterprise.
* The relationship between fact tables and measurements is extremely simple.
* If a measurement exists, it can be modeled as a fact table row.
* If a fact table row exists, it is a measurement

**Key Building Process – Facts**

* When building a fact table, the final ETL step is converting the natural keys in the new input records into the correct, contemporary surrogate keys
* ETL maintains a special surrogate key lookup table for each dimension. This table is updated whenever a new dimension entity is created and whenever a Type 2 change occurs on an existing dimension entity
* All of the required lookup tables should be pinned in memory so that they can be randomly accessed as each incoming fact record presents its natural keys. This is one of the reasons for making the lookup tables separate from the original data warehouse dimension tables.



**Loading Fact Tables**

* Managing Indexes
  + Performance Killers at load time
  + Drop all indexes in pre-load time
  + Segregate Updates from inserts
  + Load updates
  + Rebuild indexes
* Managing Partitions
  + Partitions allow a table (and its indexes) to be physically divided into *minitables* for administrative purposes and to improve query performance
  + The most common partitioning strategy on fact tables is to partition the table by the date key. Because the date dimension is preloaded and static, you know exactly what the surrogate keys are
  + Need to partition the fact table on the key that joins to the date dimension for the optimizer to recognize the constraint.
  + The ETL team must be advised of any table partitions that need to be maintained.

**Data Refresh**

* + Propogate updates from sources to the warehouse
  + **Issues:**
  + when to refresh
  + how to refresh -- refresh techniques
  + Set by administrator depending on user needs and traffic

**When to Refresh?**

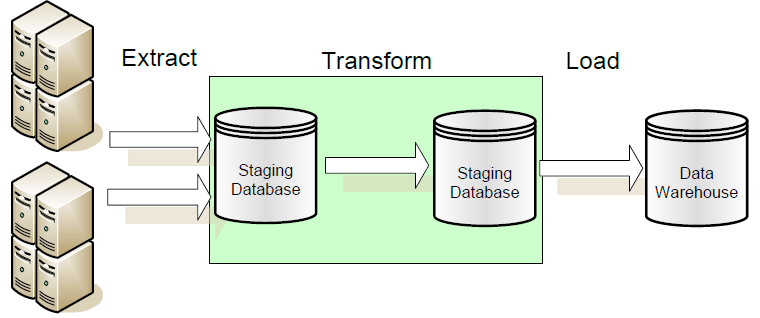
* periodically (e.g., every night, every week) or after significant events
* on every update: not warranted unless warehouse data require current data (up to the minute stock quotes)
* refresh policy set by administrator based on user needs and traffic
* possibly different policies for different sources

**Refresh Techniques**

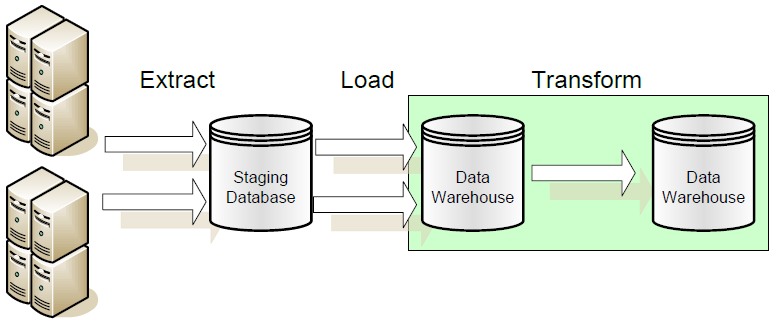
* Full Extract from base tables
  + read entire source table: too expensive
  + maybe the only choice for legacy systems

**ETL vs. ELT**

**ETL:** Extract, Transform, Load in which data transformation takes place on a separate transformation server.



**ELT:** Extract, Load, Transform in which data transformation takes place on the data warehouse server.

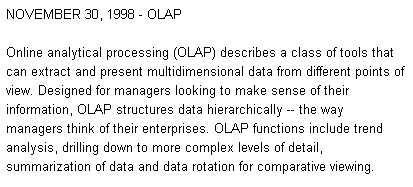


**Data warehouse support in SQL Server 2008/Oracle 11g**

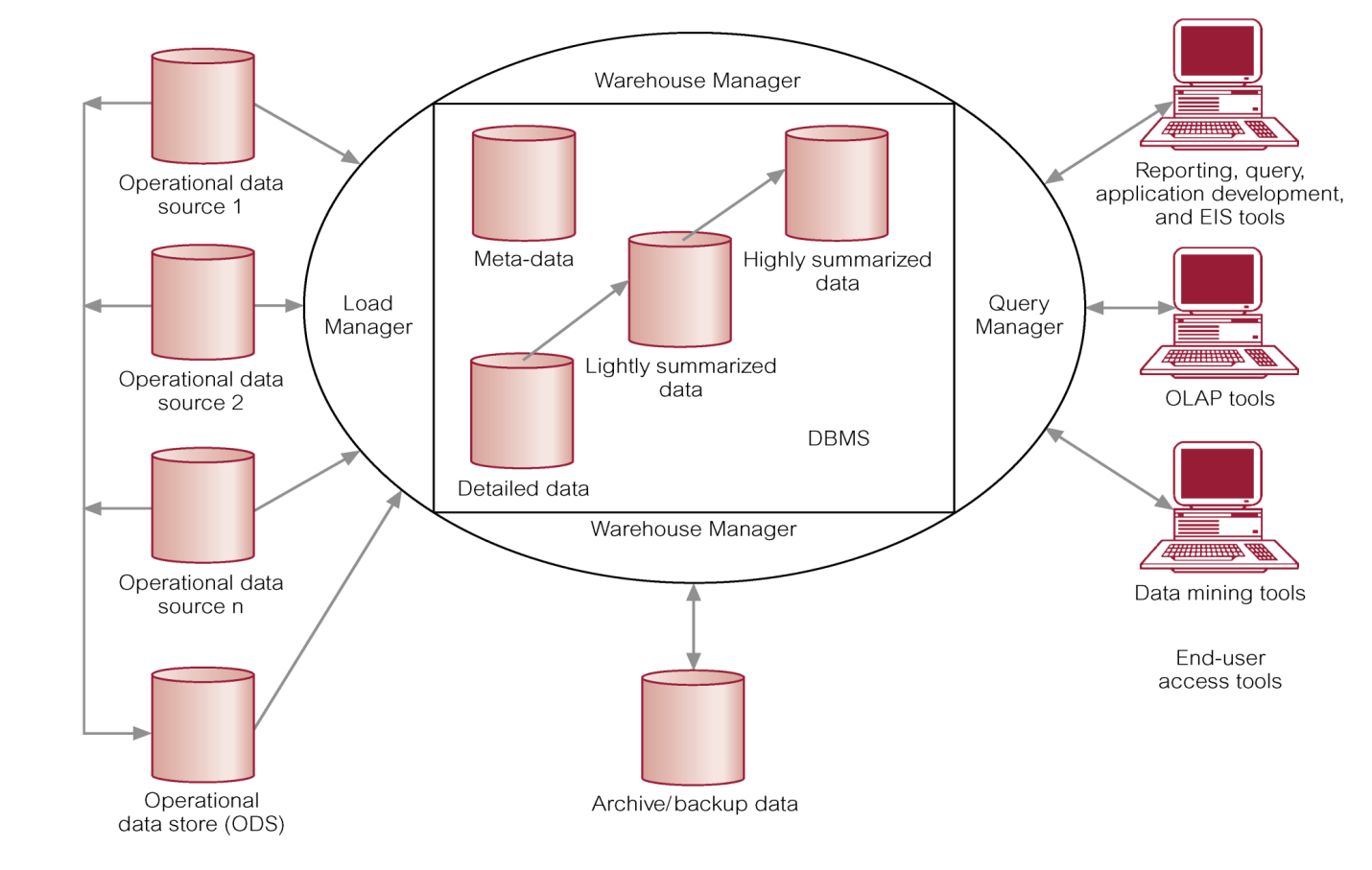
* **Oracle** supports the ETL process with their "Oracle Warehouse Builder" product. Many new features in the Oracle9i database will also make ETL processing easier.
* Data Warehouse Builder (or Oracle Data Mart builder), Oracle Designer, Oracle Express, Express Objects, etc. tools can be used to design and build a warehouse.
* **SQL Server 2008** introduced what we call the Management Data Warehouse.
* The Management Data Warehouse is a relational database that contains data that is collected from a server using the new SQL Server 2008 Data Collection mechanism.
* The Warehouse consists primarily of the following components:
  + An extensible data collector
  + Stored procedures which allow the DBA to create their own data collection set and own the resultant data collection items
  + Three Data Collections Sets which are delivered with SQL Server 2008 and which can be enabled at any time
  + Standard reports delivered with SQL Server 2008 Management Studio display data collected by the three predefined Data Collection Sets

**Unit 5 : Data Warehouse to Data Mining**

**OLAP – Online Analytical Processing**



**Data Warehouse Architecture**



* **OLAP** provides you with a very good view of *what is happening*, but can not predict *what will happen* *in the future* or *why it is happening*.
* **OLAP** is a term used to describe the analysis of complex data from the data warehouse.
* **OLAP** is an advanced data analysis environment that supports decision making, business modeling, and operations research activities.
* Can easily answer ‘who?’ and ‘what?’ questions, however, ability to answer ‘what if?’ and ‘why?’ type questions distinguishes OLAP from general-purpose query tools.
* Enables users to gain a deeper understanding and knowledge about various aspects of their corporate data through fast, consistent, interactive access to a wide variety of possible views of the data.
* Allows users to view corporate data in such a way that it is a better model of the true dimensionality of the enterprise.

OLAP is a category of applications/technology for

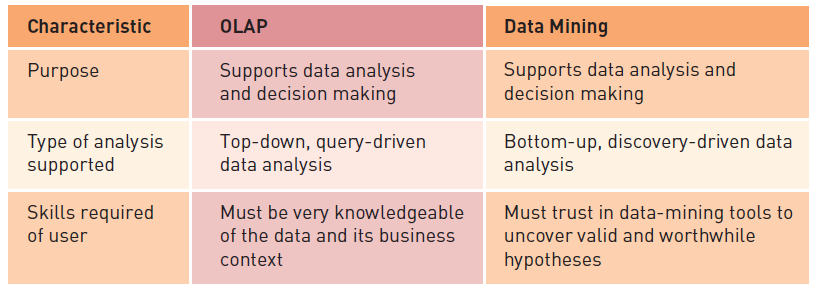
Collecting, managing, processing, and presenting

multidimensional data for analysis and management purposes.

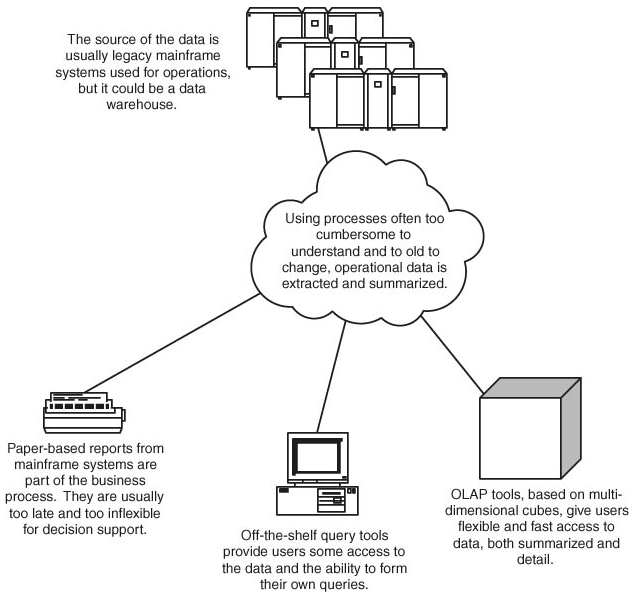
OLAP is **FASMI**

* + Fast
  + Analysis
  + Shared
  + Multidimensional
  + Information

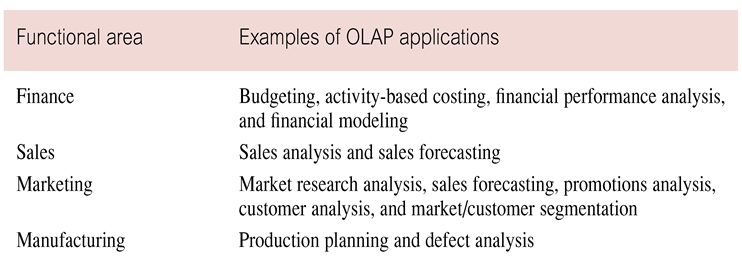
**Comparing OLAP and Data Mining**



**Where does OLAP fit in?**



**Examples of OLAP Applications in Various Functional Areas**



**OLAP Benefits**

* Increased productivity of end-users.
* Retention of organizational control over the integrity of corporate data.
* Reduced query drag and network traffic on OLTP systems or on the data warehouse.
* Improved potential revenue and profitability.

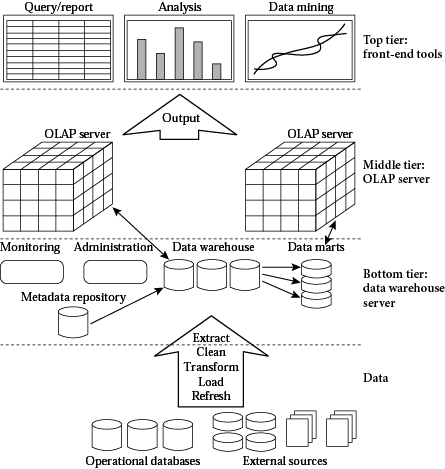
**Strengths of OLAP**

* It is a powerful visualization paradigm
* It provides fast, interactive response times
* It is good for analyzing time series
* It can be useful to find some clusters and outliers
* Many vendors offer OLAP tools

**OLAP for Decision Support**

* Goal of OLAP is to support ad-hoc querying for the business analyst
* Business analysts are familiar with spreadsheets
* Extend spreadsheet analysis model to work with warehouse data
  + Large data set
  + Semantically enriched to understand business terms (e.g., time, geography)
  + Combined with reporting features
* Multidimensional view of data is the foundation of OLAP

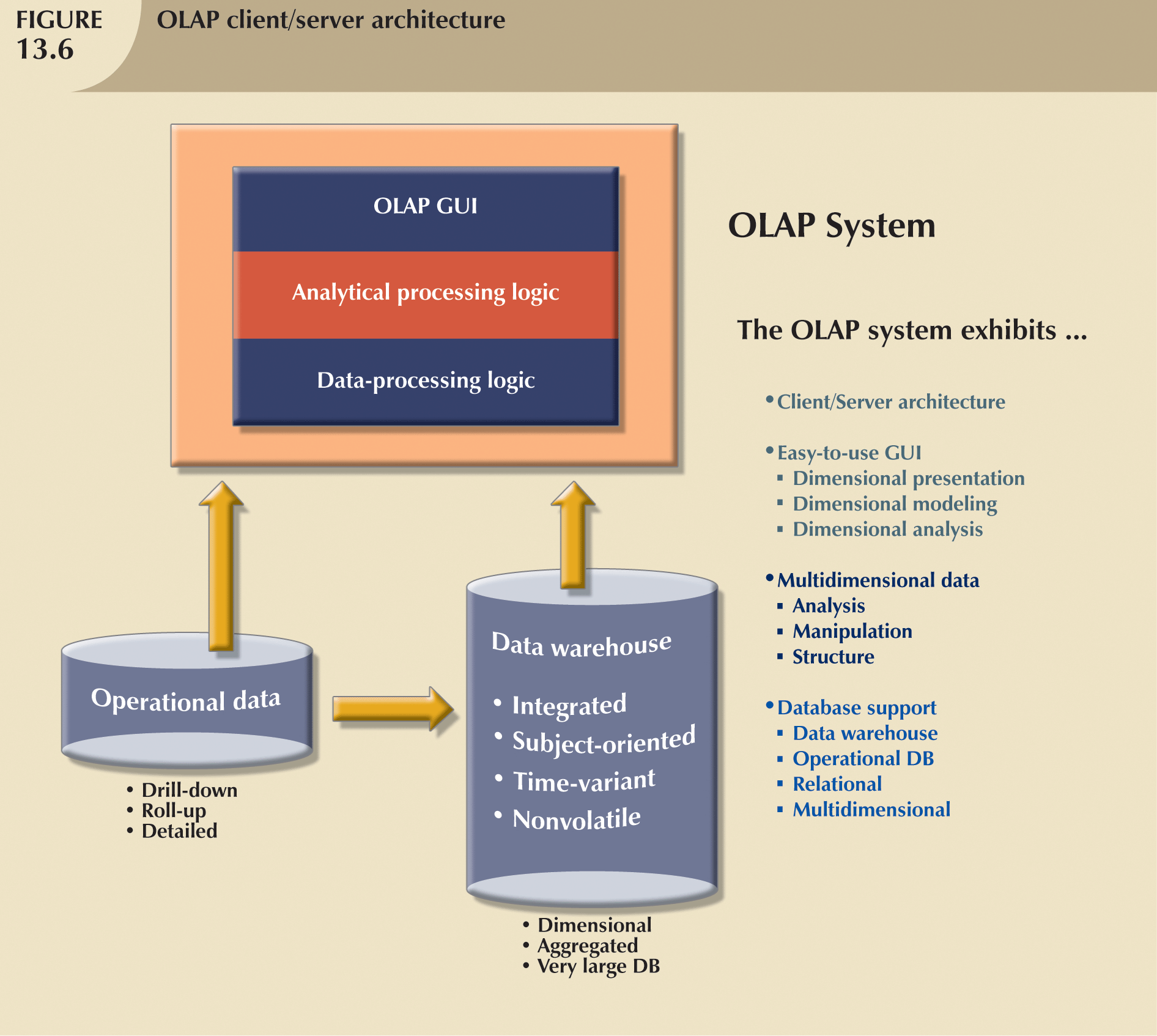
**OLAP Architecture**

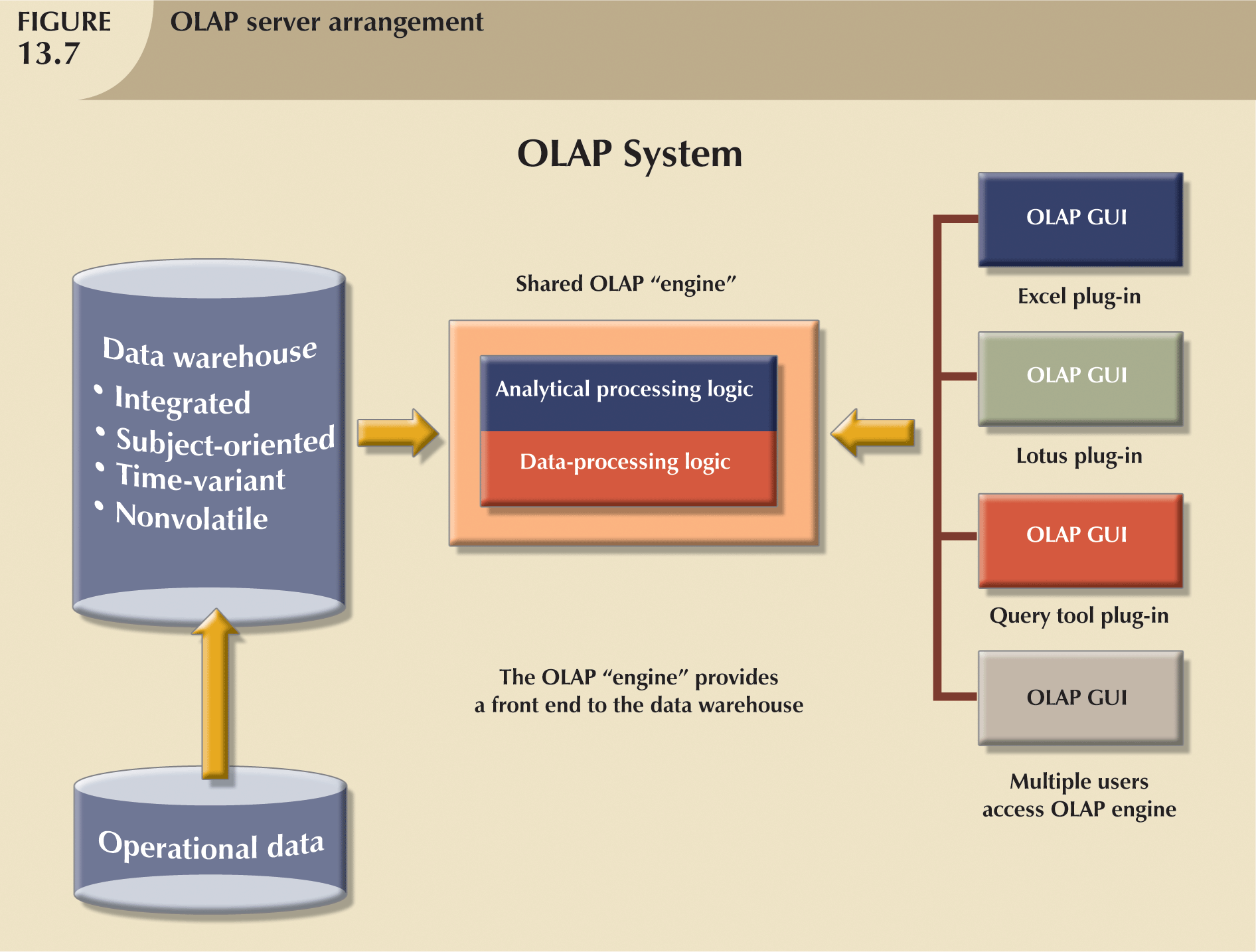


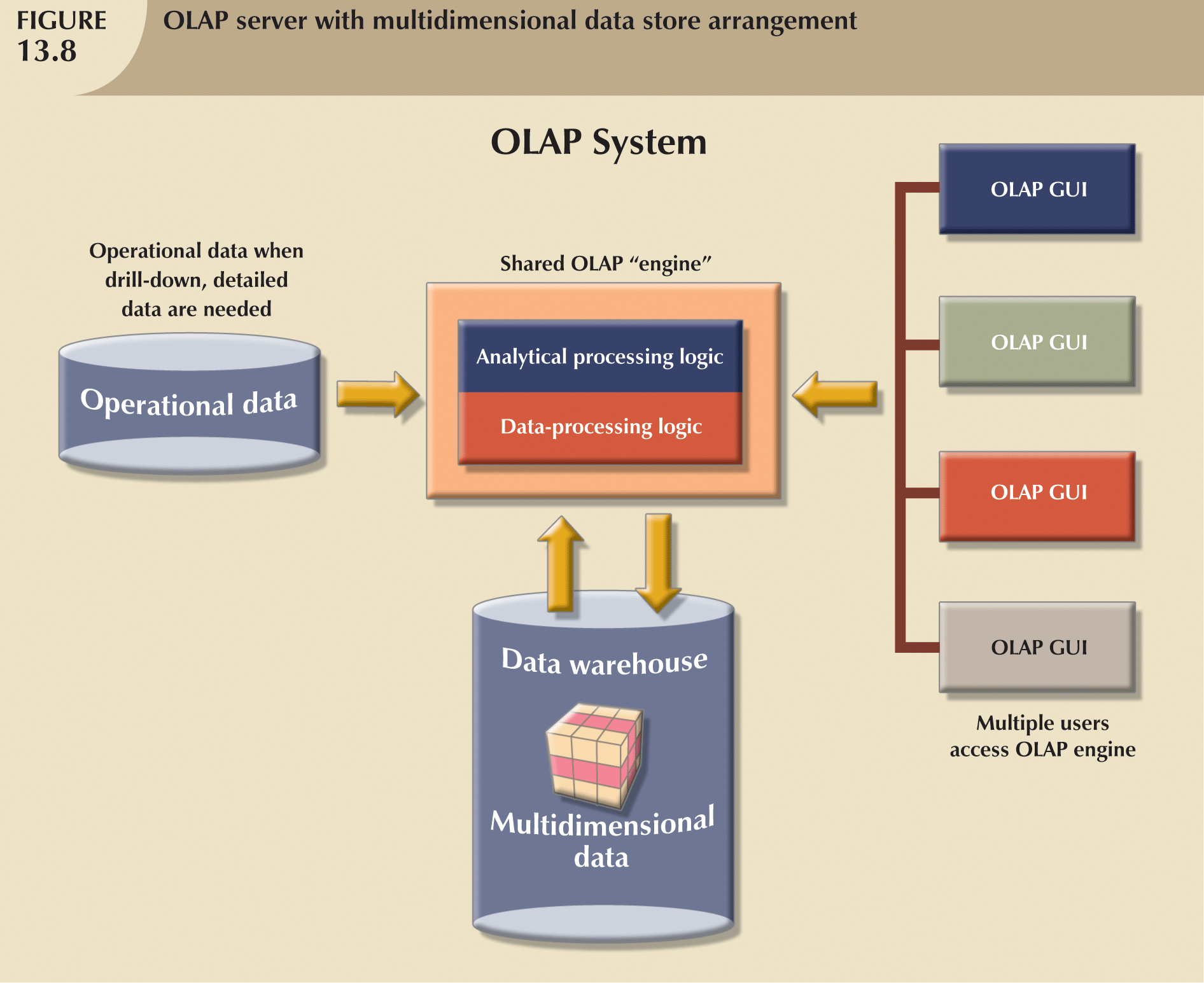
**OLAP Architecture (continued)**

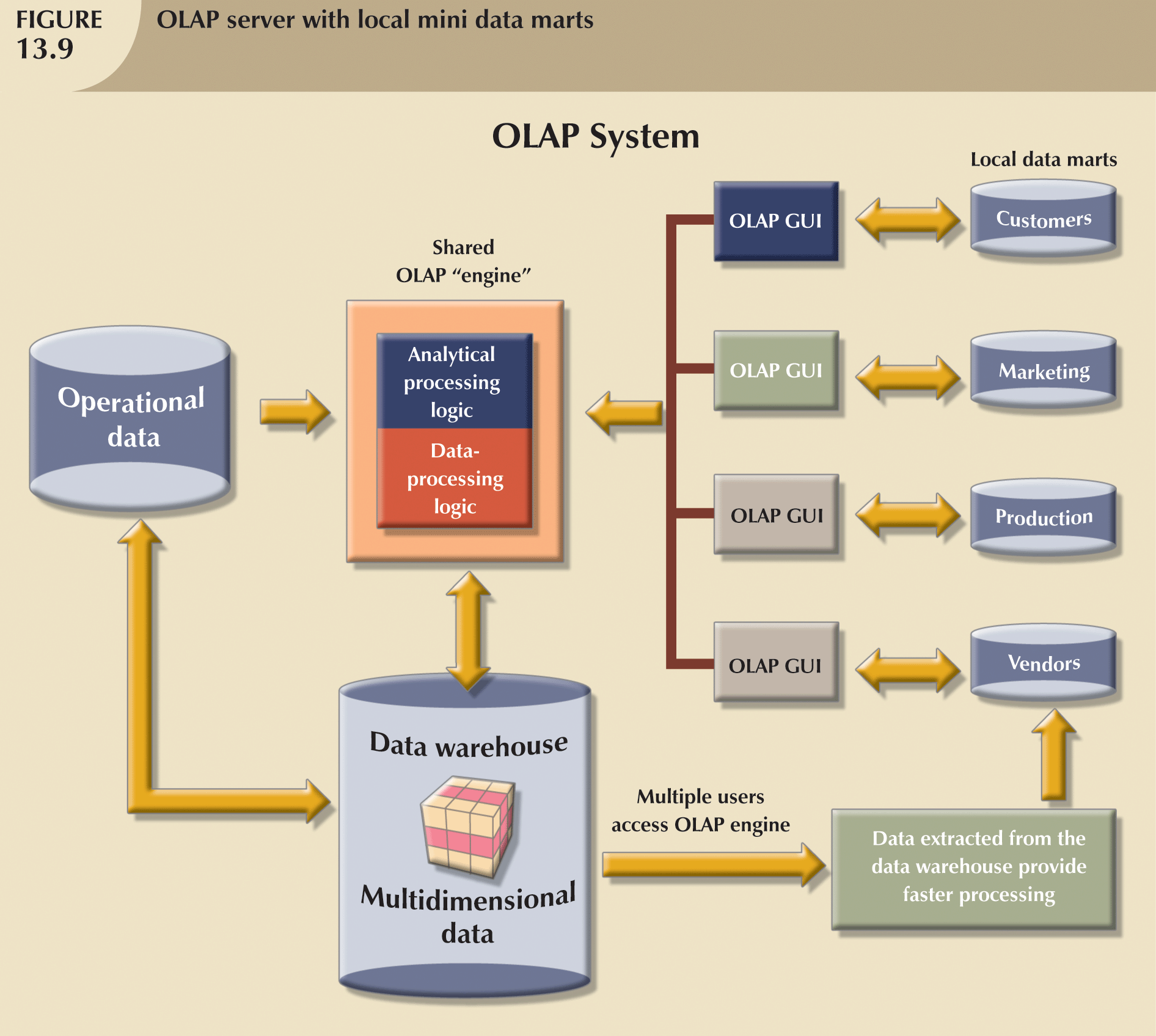
* Designed to use both operational and data warehouse data
* Defined as an “advanced data analysis environment that supports decision making, business modeling, and an operation’s research activities”
* In most implementations, data warehouse and OLAP are interrelated and complementary environments

**OLAP Client/Server Architecture**



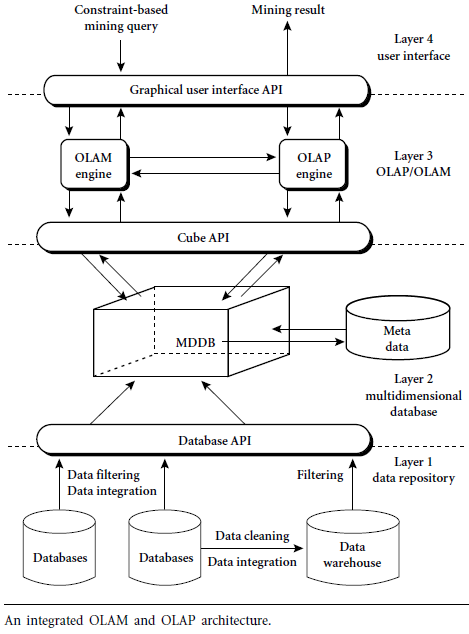






**On-Line Analytical Mining (OLAM)**

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



* An OLAM server performs analytical mining in data cubes in a similar manner as an OLAP server performs on-line analytical processing.
* An integrated OLAM and OLAP architecture is shown in figure above, where the OLAM and OLAP servers both accept user on-line queries (or commands) via a graphical user interface API and work with the data cube in the data analysis via a cube API.
* A metadata directory is used to guide the access of the data cube.
* The data cube can be constructed by accessing and/or integrating multiple databases via an MDDB API and/or by filtering a data warehouse via a database API that may support OLE DB or ODBC connections.

**Server Options**

**OLAP Server Options/Categories of OLAP Tools**

* OLAP tools are categorized according to the architecture of the underlying database.
* Three main categories of OLAP tools includes:
  + Relational OLAP (ROLAP)
  + Multi-dimensional OLAP (MOLAP or MD-OLAP)
  + DOLAP (Desktop OLAP)
  + Hybrid OLAP (HOLAP )

**Relational OLAP (ROLAP)**

Relational OLAP (ROLAP) implementations are similar in functionality to MOLAP. However, these systems use an underlying RDBMS, rather than a specialized MDDB. This gives them better scalability since they are able to handle larger volumes of data than the MOLAP architectures. Also, ROLAP implementations typically have better drill-through because the detail data resides on the same database as the multidimensional data .

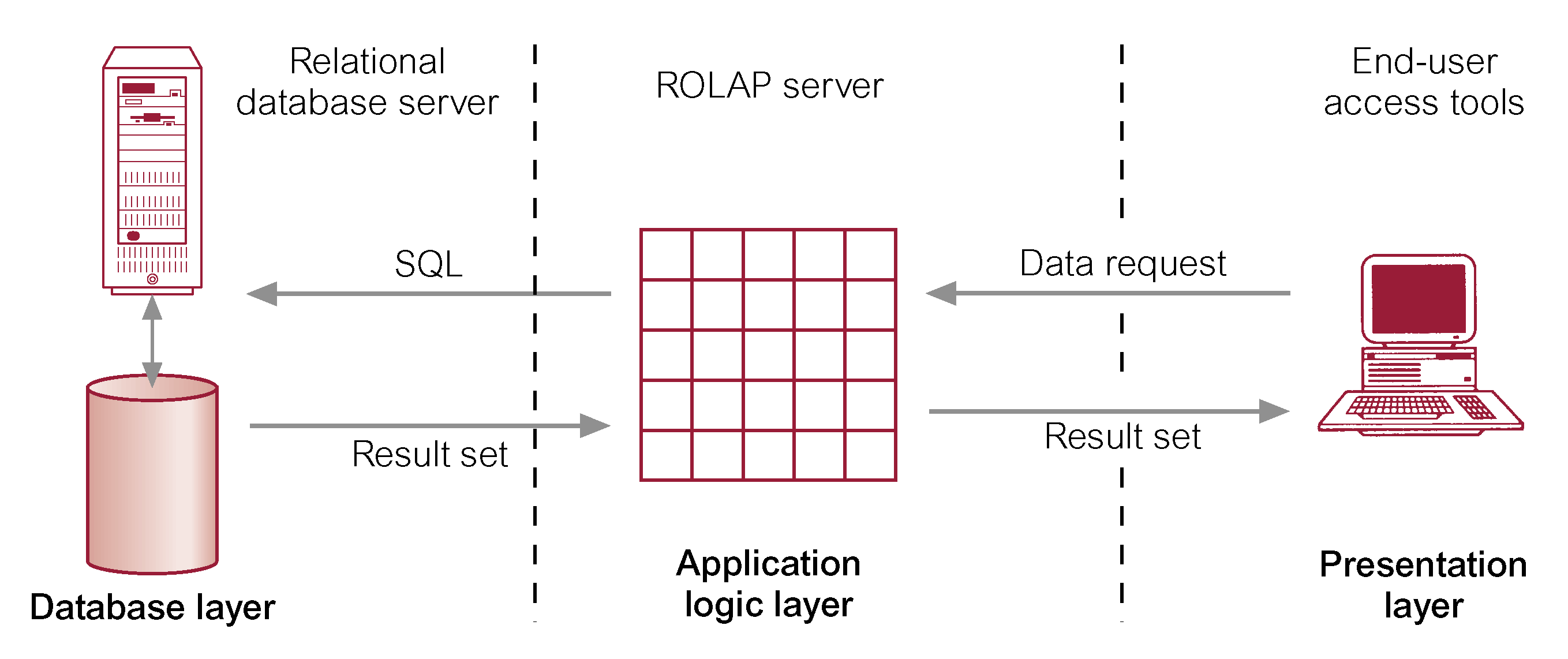
The ROLAP environment is typically based on the use of a data structure known as a star or snowflake schema. Analogous to a virtual MDDB, a star or snowflake schema is a way of representing multidimensional data in a two-dimensional RDBMS.

The data modeler builds a fact table, which is linked to multiple dimension tables. The dimension tables consist almost entirely of keys, such as location, time, and product, which point back to the detail records stored in the fact table. This type of data structure requires a great deal of initial planning and set up, and suffers from some of the same operational and flexibility concerns of MDDBs. Additionally, since the data structures are relational, SQL must be used to access the detail records.

Therefore, the ROLAP engine must perform additional work to do comparisons, such as comparing the current quarter with this quarter last year. Again, IT must be heavily involved in defining, implementing, and maintaining the database. Furthermore, the ROLAP architecture often restricts the user from performing OLAP operations in a mobile environment

* Relational Online Analytical Processing (ROLAP)
  + OLAP functionality using relational database and familiar query tools to store and analyze multidimensional data
* Adds following extensions to traditional RDBMS:
  + Multidimensional data schema support within RDBMS
  + Data access language and query performance optimized for multidimensional data
* Support for Very Large Databases
* Tune a relational DBMS to support star schemas.
* ROLAP is a fastest growing style of OLAP technology.
* Supports RDBMS products using a metadata layer - avoids need to create a static multi-dimensional data structure - facilitates the creation of multiple multi-dimensional views of the two-dimensional relation.
* To improve performance, some products use SQL engines to support complexity of multi-dimensional analysis, while others recommend, or require, the use of highly denormalized database designs such as the star schema.

**Typical Architecture for ROLAP Tools**



* With ROLAP data remains in the original relational tables, a separate set of relational tables is used to store and reference aggregation data. ROLAP is ideal for large databases or legacy data that is infrequently queried.
* ROLAP Products:
  + IBM DB2, Oracle, Sybase IQ, RedBrick, Informix
* ROLAP Tools
  + ORACLE 8i
  + ORACLE Reports; ORACLE Discoverer
  + ORACLE Warehouse Builder
  + Arbors Software’s Essbase

**Advantages of ROLAP**

* Define complex, multi-dimensional data with simple model
* Reduces the number of joins a query has to process
* Allows the data warehouse to evolve with rel. low maintenance
* HOWEVER! Star schema and relational DBMS are not the magic solution
  + Query optimization is still problematic

**Features of ROLAP:**

* Ask any question (not limited to the contents of the cube)
* Ability to drill down

Downsides of ROLAP:

* Slow Response
* Some limitations on scalability

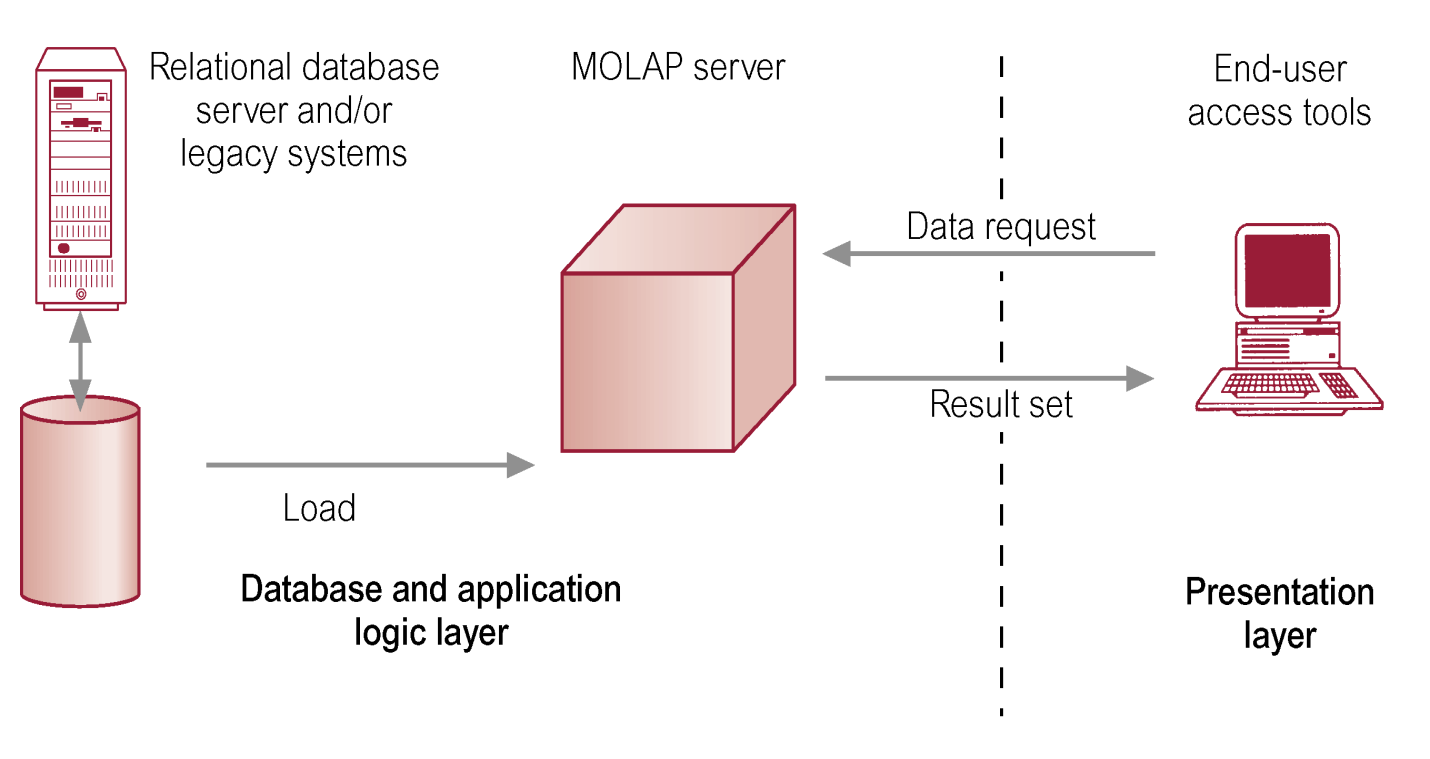
**Multi-Dimensional OLAP (MOLAP)**

The first generation of server-based multidimensional OLAP (MOLAP) solutions use multidimensional databases (MDDBs). The main advantage of an MDDB over an RDBMS is that an MDDB can provide information quickly since it is calculated and stored at the appropriate hierarchy level in advance. However, this limits the flexibility of the MDDB since the dimensions and aggregations are predefined. If a business analyst wants to examine a dimension that is not defined in the MDDB, a developer needs to define the dimension in the database and modify the routines used to locate and reformat the source data before an operator can load the dimension data.

Another important operational consideration is that the data in the MDDB must be periodically updated to remain current. This update process needs to be scheduled and managed. In addition, the updates need to go through a data cleansing and validation process to ensure data consistency. Finally, an administrator needs to allocate time for creating indexes and aggregations, a task that can consume considerable time once the raw data has been loaded. (These requirements also apply if the company is building a data warehouse that is acting as a source for the MDDB.)

Organizations typically need to invest significant resources in implementing MDDB systems and monitoring their daily operations. This complexity adds to implementation delays and costs, and requires significant IT involvement. This also results in the analyst, who is typically a business user, having a greater dependency on IT. Thus, one of the key benefits of this OLAP technology — the ability to analyze information without the use of IT professionals — may be significantly diminished.

**Typical Architecture for MOLAP Tools**



* Traditionally, require a tight coupling with the application layer and presentation layer.
* Recent trends segregate the OLAP from the data structures through the use of published application programming interfaces (APIs).
* MOLAP Products
  + Pilot, Arbor Essbase, Gentia
* MOLAP Tools
  + ORACLE Express Server
  + ORACLE Express Clients (C/S and Web)
  + MicroStrategy’s DSS server
  + Platinum Technologies’ Plantinum InfoBeacon
* Use array technology and efficient storage techniques that minimize the disk space requirements through sparse data management.
* Provides excellent performance when data is used as designed, and the focus is on data for a specific decision-support application.
* Features:

Very fast response

Ability to quickly write data into the cube

* Downsides:

Limited Scalability

Inability to contain detailed data

Load time

**Desktop OLAP (or Client OLAP)**

The desktop OLAP market resulted from the need for users to run business queries using relatively small data sets extracted from production systems. Most desktop OLAP systems were developed as extensions of production system report writers, while others were developed in the early days of client/server computing to take advantage of the power of the emerging (at that time) PC desktop. Desktop OLAP systems are popular and typically require relatively little IT investment to implement. They also provide highly mobile OLAP operations for users who may work remotely or travel extensively. However, most are limited to a single user and lack the ability to manage large data sets.



**Hybrid OLAP (HOLAP)**

Some vendors provide the ability to access relational databases directly from an MDDB, giving rise to the concept of hybrid OLAP environments. This implements the concept of "drill through," which automatically generates SQL to retrieve detail data records for further analysis. This gives end users the perception they are drilling past the multidimensional database into the source database.

The hybrid OLAP system combines the performance and functionality of the MDDB with the ability to access detail data, which provides greater value to some categories of users. However, these implementations are typically supported by a single vendor’s databases and are fairly complex to deploy and maintain. Additionally, they are typically somewhat restrictive in terms of their mobility.

* Can use data from either a RDBMS directly or a multi-dimension server.
* Equal treatment of MD and Relational Data
* Storage type at the discretion of the administrator
* Cube Partitioning
* HOLAP combines elements from MOLAP and ROLAP. HOLAP keeps the original data in relational tables but stores aggregations in a multidimensional format.
* Combines MOLAP & ROLAP
* Utilizes both pre-calculated cubes & relational data sources
* HOLAP Tools
  + ORACLE 8i
  + ORACLE Express Serve
  + ORACLE Relational Access Manager
  + ORACLE Express Clients (C/S and Web)
* HOLAP Products:
  + Oracle Express
  + Seagate Holos
  + Speedware Media/M
  + Microsoft OLAP Services

**HOLAP Features:**

* For summary type info – cube, (Faster response)
* Ability to drill down – relational data sources (drill through detail to underlying data)
* Source of data transparent to end-user

**OLAP Products**

|  |  |  |
| --- | --- | --- |
| **OLAP Category** | **Candidate Products** | **Vendor** |
| **ROLAP** | Microstrategy | Microstrategy |
|  | Business Objects | Business Objects |
|  | Crystal Holos (ROLAP Mode) | Business Objects |
|  | Essbase | Hyperion |
|  | Microsoft Analysis Services | Microsoft |
|  | Oracle Express (ROLAP Mode) | Oracle |
|  | Oracle Discoverer | Oracle |
| **MOLAP** | Crystal Holos | Business Objects |
|  | Essbase | Hyperion |
|  | Microsoft Analysis Services | Microsoft |
|  | Oracle Express | Oracle |
|  | Cognos Powerplay | Cognos |
| **HOLAP** | Hyperion Essbase+Intelligence | Hyperion |
|  | Cognos Powerplay+Impromptu | Cognos |
|  | Business Objects+Crystal Holos | Business Objects |

**Typical OLAP Operations**

**Roll up (drill-up) or Aggregation:** summarize data

* + *by climbing up hierarchy or by dimension reduction*
  + Data is summarized with increasing generalization
  + dimension reduction: e.g., total sales by city
  + summarization over aggregate hierarchy: e.g., total sales by city and year -> total sales by region and by year

**Drill down (roll down):** reverse of roll-up

* + *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
  + going from summary to more detailed views
  + Increasing levels of detail are revealed

**Slice and Dice:**

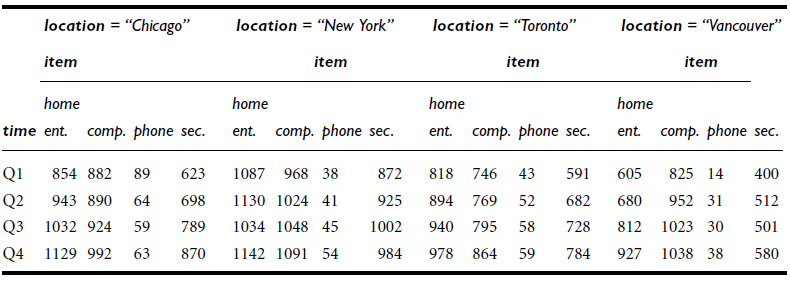
* + *project and select*
  + Performing projection operations on the dimensions.

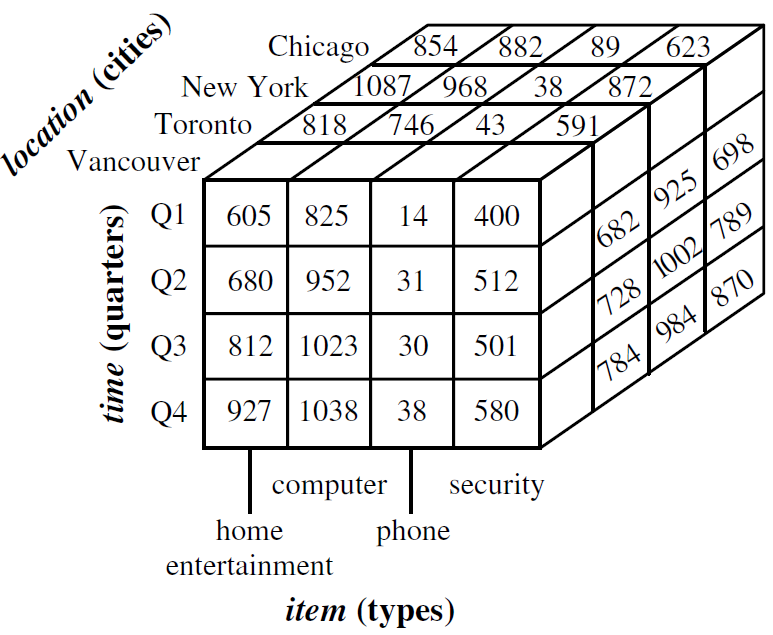
**Pivot (rotate):**

* + *reorient the cube, visualization, 3D to series of 2D planes.*
  + Cross tabulation is performed

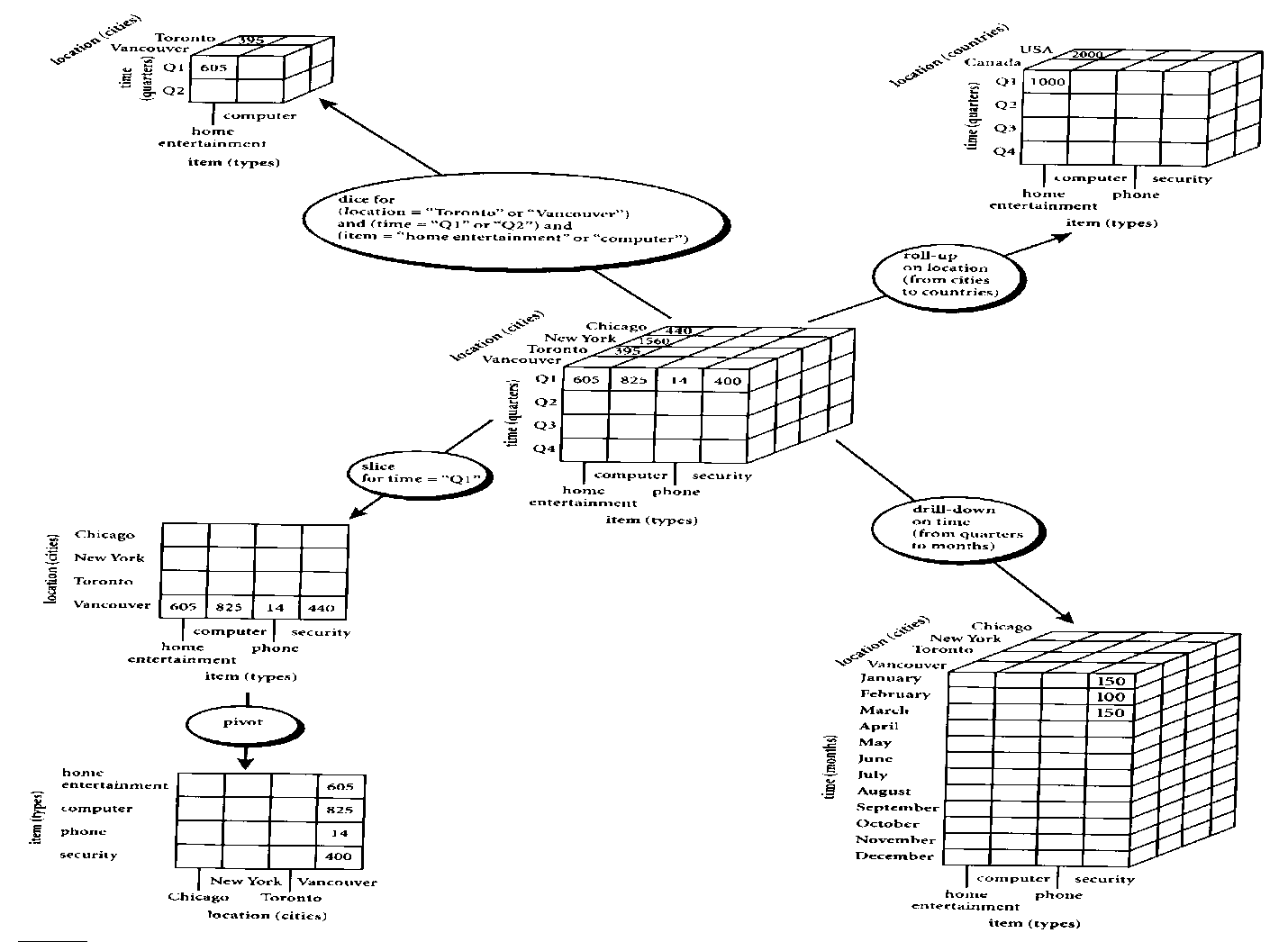
Other operations:

* + ***drill across:*** *involving (across) more than one fact table*
  + ***drill through:*** *through the bottom level of the cube to its back-end relational tables (using SQL)*

Table: A 3-D view of sales data according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars sold* (in thousands).



**Figure: A 3-D data cube representation of the data in the table above, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars sold* (in thousands).**

****

**Roll-up and Drill-down**

**The roll-up operation performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction such that one or more dimensions are removed from the given cube.**

**Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.**



Drill Down Example

**Slice and Dice**

The **slice** operation performs a selection on one dimension of the given cube, resulting in a sub cube.

The **dice** operation defines a sub cube by performing a selection on two or more dimensions.



**Rotation (Pivot Table)**

Example of Rotation (Pivot Table)



**Design and Query Processing**

Cube definition and computation in DMQL

define cube sales[item, city, year]: sum(sales\_in\_dollars)

compute cube sales

Transform it into a SQL-like language (with a new operator cube by)

SELECT item, city, year, SUM (amount)

FROM SALES

CUBE BY item, city, year

**Cube Operation**

Cube definition and computation in DMQL

define cube sales[item, city, year]: sum(sales\_in\_dollars)

compute cube sales

Transform it into a SQL-like language (with a new operator cube by)

SELECT item, city, year, SUM (amount)

FROM SALES

CUBE BY item, city, year

* Aggregate a measure on one or more dimension
* Summarize at different levels of a dimension hierarchy (state - city)
  + Total sales per city aggregated to obtain Total sales per State - *roll-up*
  + Total sales per state probed further to obtain Total sales per city - *drill-down*
* *Slicing* - an equality selection on one or more dimensions, possibly also with some dimensions projected out
* *Dicing* - range selection

**Note**: k dimensions, lead to 2k SQL queries

**SQL extension for OLAP**

**Aggregates**

* **Add up amounts for day 1**
* **In SQL: SELECT sum(amt) FROM SALE**

**WHERE date = 1**

* **Add up amounts by day**
* **In SQL: SELECT date, sum(amt) FROM SALE**

**GROUP BY date**

* **Add up amounts by day, product**
* **In SQL: SELECT date, sum(amt) FROM SALE**

**GROUP BY date, prodId**

**WEKA (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand.**

**WEKA is free software available under the GNU General Public License.**

**Features:**

* **Written in JAVA**
* **Has graphical user interfaces**
* **Contains a collection of visualization tools and algorithms for data analysis and predictive modeling**
* **Supports standard data mining tasks like data preprocessing, clustering, classification, regression, visualization, and feature selection**

**Usage:**

* **Apply a learning method to a dataset & analyze the result**
* **Use a learned model to make predictions on new instances**
* **Apply different learners to a dataset & compare results**

**MS Excel**

**In order to bridge the gap between the common user and the complex data mining process, Microsoft has introduced a new and efficient data mining tool ,the Microsoft SQL Server 2005 Data Mining Add-Ins for Office 2007 putting data mining within the reach of every user or desktop.**

**The add-in can be downloaded from the following link:**[**DOWNLOAD LINK**](http://www.microsoft.com/downloads/details.aspx?FamilyId=7c76e8df-8674-4c3b-a99b-55b17f3c4c51&displaylang=en)**.**

* **The software pre-requisites for using the add-in are:**
* **Microsoft Office 2007 installed.**
* **Microsoft SQL Server 2005 or above installed.**
  + **Microsoft .NET 2.0 framework or higher (for SQL server 2008 only).**
  + **Microsoft PowerShell (for SQL server 2008 only)**

**Once the add-in is installed, you can see the DATA MINING tab in the EXCEL ribbon. The tab contains different options like:**

**Data Preparation**

* + **Explore Data.**
  + **Clean Data.**
  + **Partition Data.**

**Data Modeling**

* + **Classify.**
  + **Estimate.**
  + **Cluster.**
  + **Associate.**
  + **Forecast.**
  + **Advanced.**

**Accuracy and Validation**

* + **Accuracy Chart.**
  + **Classification Matrix.**
  + **Profit Chart.**

**Model Usage**

* + **Browse.**
  + **Query.**

**Management**

**Connection**

* + **No Connection.**
  + **Trace.**

**Help**

**Conclusion:**

**The Microsoft SQL Server Data mining add-in for Microsoft Excel provides users with an easy to use interface that is capable of performing complex data mining tasks with ease. The add-in can be extremely useful for both, people who just want to get more out of their data and also for those interested in serious data mining.**

**Microsoft SQL Server**

**Microsoft SQL Server is a relational database server, developed by Microsoft: it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).**

* **Microsoft has introduced a wealth of new data mining features in Microsoft SQL Server 2008 that allow businesses to answer their concerns with data and mining for information in them.**
* **The current version of SQL Server, SQL Server 2008, (code-named "Katmai") aims to make data management self-tuning, self organizing, and self maintaining.**
* **SQL Server 2008 data mining features are integrated across all the SQL Server products, including SQL Server, SQL Server Integration Services, and Analysis Services.**
* **Accessing the data mining results is as simple as using an SQL-like language called Data Mining Extensions to SQL, or DMX.**

**Oracle**

**The Oracle Database (commonly referred to as *Oracle RDBMS* or simply as *Oracle*) is an object-relational database management system (ORDBMS) produced and marketed by Oracle Corporation.**

* **Oracle Data Mining (ODM) is used to incorporate data mining with the Oracle database.**
* **ODM is used for both supervised (where a particular target value should be specified) and unsupervised (where patterns in data are observed) data mining.**
* **The results of Oracle Data Mining can be viewed by the Oracle Business Intelligence’s reporting/publishing component.**
* **Oracle BI Standard Edition One is a product that is used to extract business information concealed in the data.**
* **Oracle Warehouse Builder (OWB) is used to create the logical and physical design of the data mart.**
* **The Oracle BI Server is used to build a repository of metadata from the data mart that was created using the Oracle Data warehouse builder.**
* **The users ultimately interact with Oracle BI Answers to extract useful information from the data mart created using OWB (Oracle Warehouse Builder).**
* **Oracle BI Interactive  Dashboards are used to publish the data extracted from the data mart so that the users can have an easy access to it.**
* **Oracle BI publisher is used to create reports that are very essential to any kind of business.**

**SPSS**

**SPSS (originally, Statistical Package for the Social Sciences)  is a computer program used for survey authoring and deployment (IBM SPSS Data Collection), data mining (IBM SPSS Modeler), text analytics, statistical analysis, and collaboration and deployment (batch and automated scoring services).**

**Assignments**

1. **Discuss the motivation behind *OLAP mining* (*OLAM*).**
2. **In data warehouse technology, a multiple dimensional view can be implemented by a relational database technique (*ROLAP*), or by a multidimensional database technique (*MOLAP*), or by a hybrid database technique (*HOLAP*).**

**(a) Briefly describe each implementation technique.**

**(b) For each technique, explain how each of the following functions may be implemented:**

**i. The generation of a data warehouse (including aggregation)**

**ii. Roll-up**

**iii. Drill-down**

**iv. Incremental updating**

**Which implementation techniques do you prefer, and why?**

**Unit 6 : Data Mining Approaches and Methods**

**Types of Data Mining Models**

**Predictive Model**

**(a)Classification -Data is mapped into predefined groups or classes. Also termed as supervised learning as classes are established prior to examination of data.**

**(b) Regression- Mapping of data item into known type of functions. These may be linear, logistic functions etc.**

**(c) Time Series Analysis- Value of an attribute are examined at evenly spaced times, as it varies with time.**

**(d) Prediction- It means fore telling future data states based on past and current data.**

**Descriptive Models**

**(a) Clustering- It is referred as unsupervised learning or segmentation/partitioning. In**

**clustering groups are not pre-defined.**

**(b) Summarization- Data is mapped into subsets with simple descriptions . Also termed as Characterization or generalization.**

**(c) Sequence Discovery- Sequential analysis or sequence discovery utilized to find out**

**sequential patterns in data. Similar to association but relationship is based on time.**

**(d) Association Rules- A model which identifies specific types of data associations.**

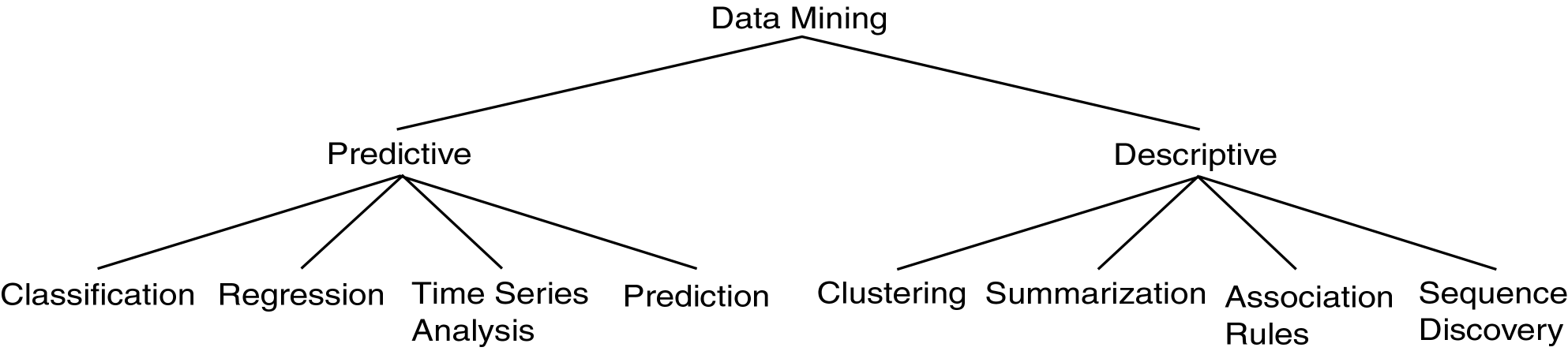
**Descriptive vs. Predictive Data Mining**

**Descriptive Mining:**

**It describes concepts or task-relevant data sets in concise, summarative, informative, discriminative forms.**

**Predictive Mining:**

**It is based on data and analysis, constructs models for the database, and predicts the trend and properties of unknown data.**

****

**Supervised and Unsupervised learning**

**Supervised learning:**

**The network answer to each input pattern is directly compared with the desired answer and a feedback is given to the network to correct possible errors**

**Unsupervised learning:**

**The target answer is unknown. The network groups the input patterns of the training sets into clusters, based on correlation and similarities.**

**Concept/Class Description**

**A concept typically refers to a collection of data such as *frequent buyers, graduate students*, and so on. As a data mining task, concept description is not a simple enumeration of the data. Instead, concept description generates descriptions for the *characterization* and *comparison* of the data. It is sometimes called class description, when the concept to be described refers to a class of objects.**

**Concept description is the most basic form of descriptive data mining. It describes a given set of task-relevant data in a concise and summarative manner, presenting interesting general properties of the data.**

**Data can be associated with classes or concepts. For example, in the *AllElectronics* store, classes of items for sale include *computers* and *printers*, and concepts of customers include *bigSpenders* and *budgetSpenders*. It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. Such descriptions of a class or a concept are called class/concept descriptions.**

**These descriptions can be derived via**

1. **Data characterization, by summarizing the data of the class under study (often called the target class) in general terms, or**
2. **Data discrimination, by comparison of the target class with one or a set of comparative classes (often called the contrasting classes), or**
3. **both data characterization and discrimination.**

**The output of data characterization can be presented in various forms. Examples include pie charts, bar charts, curves, multidimensional data cubes, and multidimensional tables, including crosstabs. The resulting descriptions can also be presented as generalized relations or in rule form(called characteristic rules).**

**Data Characterization**

**Data Characterization:**

**A data mining system should be able to produce a description summarizing the characteristics of customers.**

**Example:**

**The characteristics of customers who spend more than $1000 a year at XYZ store. The result can be a general profile such as age, employment status or credit ratings.**

**Data Discrimination**

**Data Discrimination:**

**It is a comparison of the general features of targeting class data objects with the general features of objects from one or a set of contrasting classes. User can specify target and contrasting classes.**

**Example:**

**The user may like to compare the general features of software products whose sales increased by 10% in the last year with those whose sales decreased by about 30% in the same duration.**

**Concept Description vs. OLAP**

* **Concept Description** 
  + **can handle complex data types of the attributes and their aggregations**
  + **a more automated process**
* **OLAP** 
  + **restricted to a small number of dimension and measure types**
  + **user-controlled process**

**Data Generalization**

Data generalization is a process that abstracts a large set of task-relevant data in a database from a relatively low conceptual level to higher conceptual levels.

Data generalization summarizes data by replacing relatively low-level values (such as numeric values for an attribute *age*) with higher-level concepts (such as *young*, *middleaged*, and *senior*).

Given the large amount of data stored in databases, it is useful to be able to describe concepts in concise and brief terms at generalized (rather than low) levels of abstraction. Allowing data sets to be generalized atmultiple levels of abstraction facilitates users in examining the general behavior of the data.

Data generalization approaches include data cube approach (OLAP Approach) and attribute oriented induction approach.

**Data Cube Approach (without using Attribute Oriented-Induction)**

It perform computations and store results in data cubes

Strength

* + An efficient implementation of data generalization
  + Computation of various kinds of measures
    - e.g., count( ), sum( ), average( ), max( )
  + Generalization and specialization can be performed on a data cube by *roll-up* and *drill-down*

Limitations

* + handle only dimensions of *simple nonnumeric data* and measures of *simple aggregated numeric values*.
  + Lack of intelligent analysis, can’t tell which dimensions should be used and what levels should the generalization reach

**Attribute-Oriented Induction**

* Proposed in 1989 (KDD ‘89 workshop)
* Not confined to categorical data nor particular measures.
* How it is done?
  + Collect the task-relevant data( *initial relation*) using a relational database query
  + Perform generalization by *attribute removal* or *attribute generalization*.
  + Apply aggregation by merging identical, generalized tuples and accumulating their respective counts.
  + Interactive presentation with users.

**Classification and Prediction**

Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help provide us with a better understanding of the data at large.

Whereas *classification* predicts categorical (discrete, unordered) labels, *prediction* models continuous valued functions.

For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

**Prediction**

Prediction is viewed as the construction and use of a model to assess the class of an unlabeled sample or to assess the value ranges of an attribute that a given sample is likely to have.

It is a statement or claim that a particular event will occur in the future in more certain terms than a forecast . It is similar to classification .It constructs a model to predict unknown or missing values. Prediction is the most prevalent grade level expectation on reasoning in state mathematics standards.

Generally it predicts a continuous value rather than categorical label. Numeric prediction predicts the continuous value. The most widely used approach for numeric prediction is regression.

Regression analysis is used to model the relationship between one or more independent or predictor variables and a dependent or response variable. In the context of Data Mining, predictor variables are attributes of interest describing the tuple.

**Linear Regression**

Regression is a statistical methodology developed by Sir Frances Galton in 1822-1911.Straight line regression analysis involves a response variable y and a single predictor variable x.

The simplest form of regression is

y = a + bx

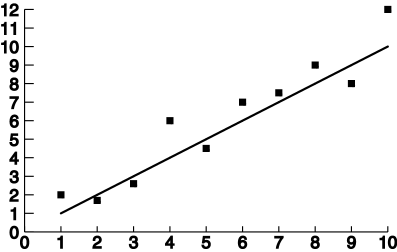
Where y is response variable and x is single predictor variable y is a linear function of x. a and b are regression coefficients.

As the regression coefficients are also considered as weights, we may write the above equation as:

y = w+w1x

These coefficients are solved by the method of least squares, which estimates the best fitting straight line as the one that minimizes the error between the actual data and the estimate of the line.

**Linear Regression**

****

**Classification**

**Classification can be described by a two step process given in appended block diagram:**

**Step 1**

**Also known as “supervised learning” as class labels are known. It is different than “Unsupervised learning” or clustering where class labels are not known. A model is built describing a predetermined set of data classes or concepts. The model is constructed by analyzing database tuples described by their attributes. Each tuple is assumed to belong to a predefined class and called as a class label attribute. Data tuples are also referred as Samples, Examples or Objects. Data tuples selected randomly form a training data set and are called training samples. The learning of the model is termed as” Supervised “ as it is told which class the training sample belongs. This is in contrast to Clustering which is termed unsupervised learning.**

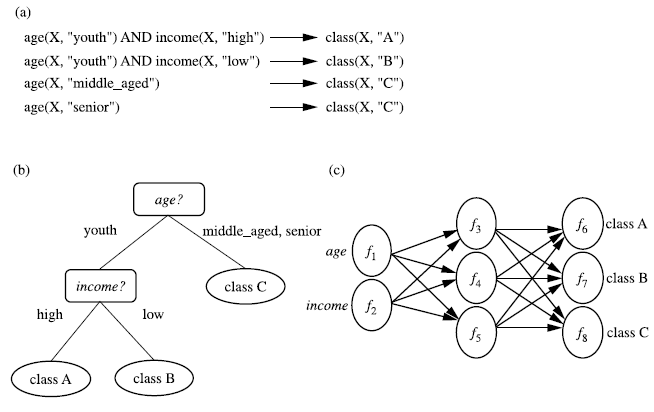
**Step 2**

**Test data verifies the accuracy of Classification Rules**

**The model is used for classification. First the predictive accuracy of the model is estimated. If the accuracy is acceptable the model can be used to classify future tuples or objects for which class label is not known.**

**Classification** is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).

*“How is the derived model presented?”* The derived model may be represented in various forms, such as *classification (IF-THEN) rules*, *decision trees*, *mathematical formulae*, or *neural networks.*

****

**Figure : A classification model can be represented in various forms, such as (a) IF-THEN rules, (b) a decision tree, or a (c) neural network.**

**Classification : Example of Grading**

**How does classification work?**

Data classification is a two-step process:

1. Model construction

Training data are analyzed by a classification algorithm. A classifier is built describing a predetermined set of data classes or concepts. Also called as training phase or learning stage.

1. Model usage

Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

**Model Construction**

**Model Usage**

**Examples of Classification Algorithms**

* **Decision Trees**
* **Neural Networks**
* **Bayesian Networks**

**Issues regarding classification and prediction**

**Issues (1): Data Preparation**

* Data cleaning

Preprocess data in order to reduce noise and handle missing values

* Relevance analysis (feature selection)

Remove the irrelevant or redundant attributes

* Data transformation

Generalize and/or normalize data

**Issues (2): Evaluating Classification Methods**

* Predictive accuracy
* Speed and scalability

time to construct the model

time to use the model

* Robustness

handling noise and missing values

* Scalability

efficiency in disk-resident databases

* Interpretability

understanding and insight provided by the model

* Goodness of rules

decision tree size

compactness of classification rules

**Decision Trees**

A decision tree is a predictive model that as its name implies can be viewed as a tree. Specifically each branch of the tree is a classification question and the leaves are partitions of data set with their classification.

A decision tree makes a prediction on the basis of a series of decisions. The decision trees are being built on historical data and are a part of the supervised learning. The machine learning technique for inducting a decision tree from data is called decision tree learning.

* + Internal node denotes a test on an attribute
  + Branch represents an outcome of the test
  + Leaf nodes represent class labels or class distribution

**In data mining, trees have three more descriptive categories/names:**

* *Classification tree* analysis - when the predicted outcome is the class to which the data belongs.
* *Regression tree* analysis - when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient’s length of stay in a hospital).
* *Classification And Regression Tree* (CART) analysis - when both of the above procedures are referred.

Decision tree generation consists of two phases

* Tree construction

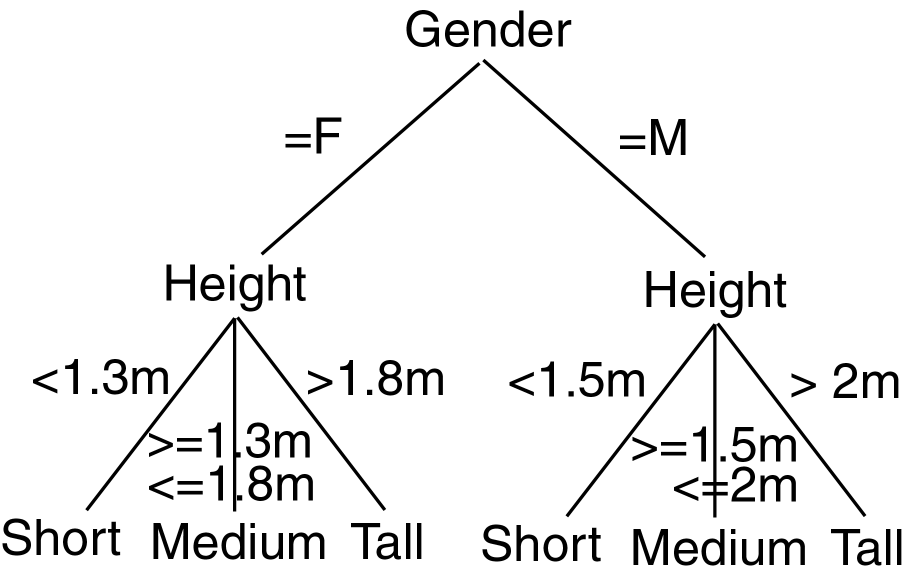
At start, all the training examples are at the root

Partition examples recursively based on selected attributes

* Tree pruning

Identify and remove branches that reflect noise or outliers

**Decision Tree Example**



**Decision Tree: Example**

**Day Outlook Temperature Humidity Wind Play Tennis**

**1 Sunny Hot High Weak No**

**2 Sunny Hot High Strong No**

**3 Overcast Hot High Weak Yes**

**4 Rain Mild High Weak Yes**

**5 Rain Cool Normal Weak Yes**

**6 Rain Cool Normal Strong No**

**7 Overcast Cool Normal Strong Yes**

**8 Sunny Mild High Weak No**

**9 Sunny Cool Normal Weak Yes**

**10 Rain Mild Normal Weak Yes**

**11 Sunny Mild Normal Strong Yes**

**12 Overcast Mild High Strong Yes**

**13 Overcast Hot Normal Weak Yes**

**14 Rain Mild High Strong No**

**Decision Tree Induction**

Basic algorithm (a greedy algorithm)

Tree is constructed in a top-down recursive divide-and-conquer manner

At start, all the training examples are at the root

Examples are partitioned recursively to *maximize purity*

Conditions for stopping partitioning

All samples belong to the same class

Leaf node smaller than a specified threshold

Tradeoff between complexity and generalizability

Predictions for new data:

Classification by majority voting is employed for classifying all members of the leaf

Probability based on training data that ended up in that leaf.

Class Probability estimates can be used also

**Algorithm for building Decision Trees**

Decision trees are a popular structure for supervised learning. They are constructed using attributes best able to differentiate the concepts to be learned.

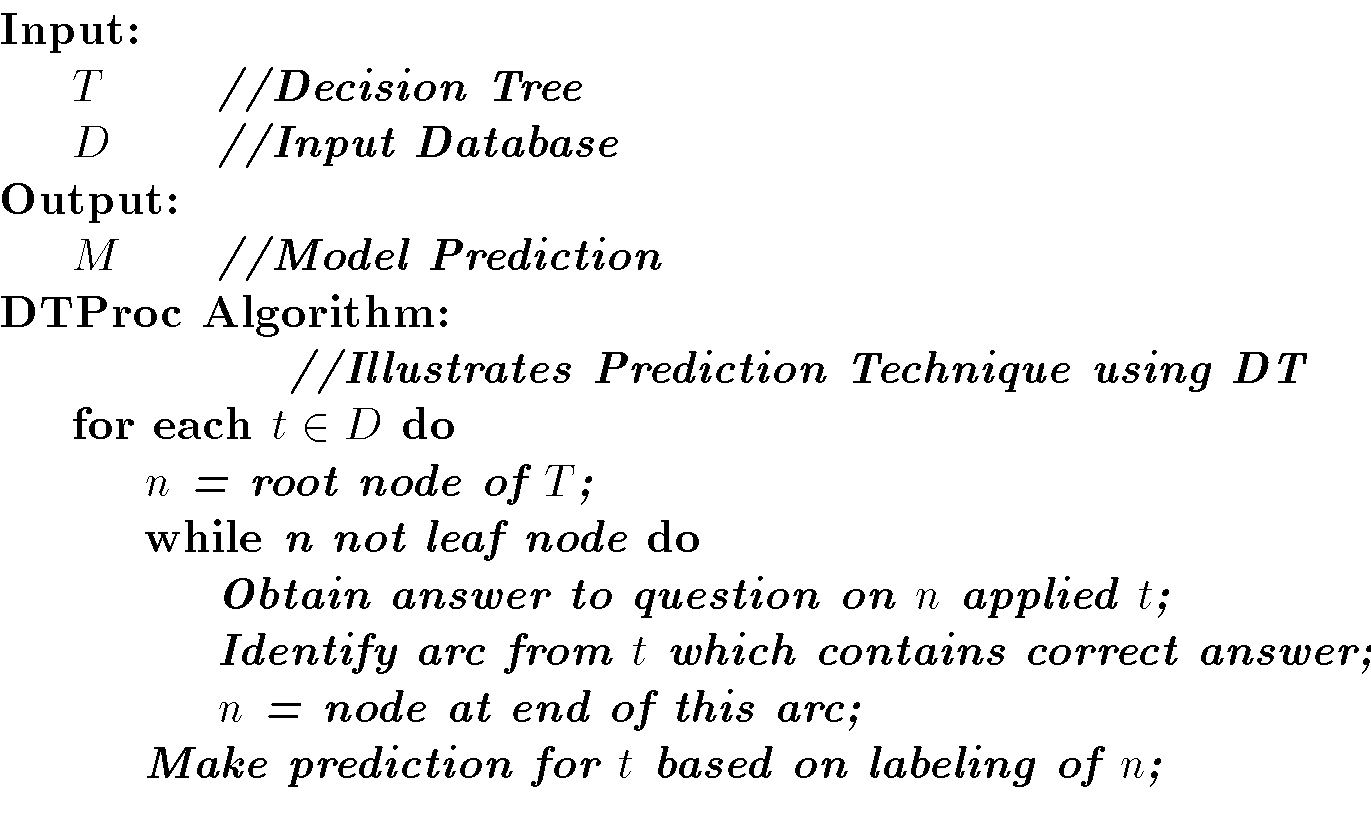
A decision tree is built by initially selecting a subset of instances from a training set. This subset is then used by the algorithm to construct a decision tree. The remaining training set instances test the accuracy of the constructed tree.

If the decision tree classified the instances correctly, the procedure terminates.

If an instance is incorrectly classified, the instance is added to the selected subset of training instances and a new tree is constructed.

This process continues until a tree that correctly classify all non-selected instances is created or the decision tree is built from the entire training set.

**Decision Tree Algorithm**



**Decision Tree Pseudocode**

node = tree-design(Data = {X,C})

For i = 1 to d

quality\_variable(i) = quality\_score(Xi, C)

end

node = {X\_split, Threshold } for max{quality\_variable}

{Data\_right, Data\_left} = split(Data, X\_split, Threshold)

if node == leaf?

return(node)

else

node\_right = tree-design(Data\_right)

node\_left = tree-design(Data\_left)

end

end

**Basic algorithm for inducing a decision tree**

Algorithm: Generate\_decision\_tree. Generate a decision tree from the given training data.

Input: The training samples, represented by discrete-valued attributes; the set of candidate attributes, attribute-list;

Output: A decision tree

Begin

Partition (S)

If (all records in S are of the same class or only 1 record found in S)

then return;

For each attribute Ai do

evaluate splits on attribute Ai;

Use best split found to partition S into S1 and S2 to grow a tree with two Partition (S1) and Partition (S2);

Repeat partitioning for Partition (S1) and (S2) until it meets tree stop growing criteria;

End;

**Decision Tree Algorithms & their main Issues**

1. **Tree Structure** - Selection of a tree structure like Balanced tree for improving performance.
2. **Training Data** - Structure of a tree depends on the training data. Selecting adequate data prevents either the tree to overfit and on the other hand good enough to work on a general data
3. **Stopping Criteria** - Construction of a tree stops on a Stopping criteria. It is essential to achieve a balance between too early or late to create a tree with right level.
4. **Pruning** - After constructing a tree, modify it to remove duplication or subtrees.
5. **Splitting** - Selection of the best splitting attribute and size of the training set are important factors in creating a decision tree algorithm. For example, Splitting attributes in the case of students may be gender, marks scored and electives chosen. The order in which splitting attributes are chosen are important for avoiding redundancy and unnecessary comparisons at different levels.

**Decision Tree Learning Algorithm - ID3**

**ID3 (Iterative Dichotomiser)** is a simple decision tree learning algorithm developed by Ross Quinlan (1983). ID3 follow non-backtracking approach in which decision trees are constructed in a top-down recursive “divide and conquer” manner to test each attribute at every tree node. This approach starts with a training set of tuples and their associated class labels. Training set is recursively partitioned into smaller subsets as the tree is being built.

**ID3 Algorithm:**

1.create a node N;

2. if tuples in D are all of the same class C then

3. return N as a leaf node labeled with the class C;

4. if attribute\_list is empty then

5. return N as a leaf node labled with the majority class in D;

6.apply Attribute\_selection\_method(D, attribute\_list) to find the "best"splitting\_criterion;

label node N with with splitting\_criterion;

7. if splitting\_attribute is discrete-valued and

multiway splits allowed then //not restricted to binary trees

attribute\_list (arrow mark) attribute \_ list - splitting\_attribute;

8. for each outcome j of splitting\_criterion

9. let (symbol)be the set of data tuples in D satisfying outcome j; // partition

10.if (symbol) is empty then

attach a leaf labeled with the majority class in D to node N;

11. else attach the node returned by

Generate\_decision\_tree(symbol,attribute\_list)to node N;

endfor

return N;

**Explanation of Algorithm:**

The above algorithm has three parameters D, attribute\_list and attribute\_selection\_method. D is data partition. It is a set of training tuples and their associated class labels. Attribute\_list contains a list of attributes describing the tuples.

Now tree starts as a single node N. It represents the training tuples in D. If the tuples in D are all of the same class then node N is considered as leaf. It is labeled with that class. It is occurring in step 2 and 3. Step 4 and 5 are terminating conditions. IF this condition does not follow then algorithm calls Attribute\_selection\_method to detemine the spilitting criterion. This criterion determines the best way to partition the tuples in D into individual classes(step 6). Step 7 serves as a test at the node. In steps 10 and 11, tuples in D are partitioned.

**Advantages of using ID3**

* Understandable prediction rules are created from the training data.
* Builds the fastest tree.
* Builds a short tree.
* Only need to test enough attributes until all data is classified.
* Finding leaf nodes enables test data to be pruned, reducing number of tests.
* Whole dataset is searched to create tree.

**Disadvantages of using ID3**

* Data may be over-fitted or over-classified, if a small sample is tested.
* Only one attribute at a time is tested for making a decision.
* Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

**Pros and Cons of Decision Tree**

**Pros**

* + no distributional assumptions
  + can handle real and nominal inputs
  + speed and scalability
  + robustness to outliers and missing values
  + interpretability
  + compactness of classification rules
  + They are easy to use.
  + Generated rules are easy to understand .
  + Amenable to scaling and the database size.

**Cons**

* + several tuning parameters to set with little guidance
  + decision boundary is non-continuous
  + Cannot handle continuous data.
  + Incapable of handling many problems which cannot be divided into attribute domains.
  + Can lead to over-fitting as the trees are constructed from training data.

**Neural Networks**

Neural Network is a set of connected INPUT/OUTPUT UNITS, where each connection has a WEIGHT associated with it. It is a case of SUPERVISED, INDUCTIVE or CLASSIFICATION learning.

Neural Network learns by adjusting the weights so as to be able to correctly classify the training data and hence, after testing phase, to classify unknown data. Neural Network needs long time for training. Neural Network has a high tolerance to noisy and incomplete data.

***A Neural Network (NN)*** is a directed graph F=<V,A> with vertices V={1,2,…,n} and arcs A={<i,j>|1<=i,j<=n}, with the following restrictions:

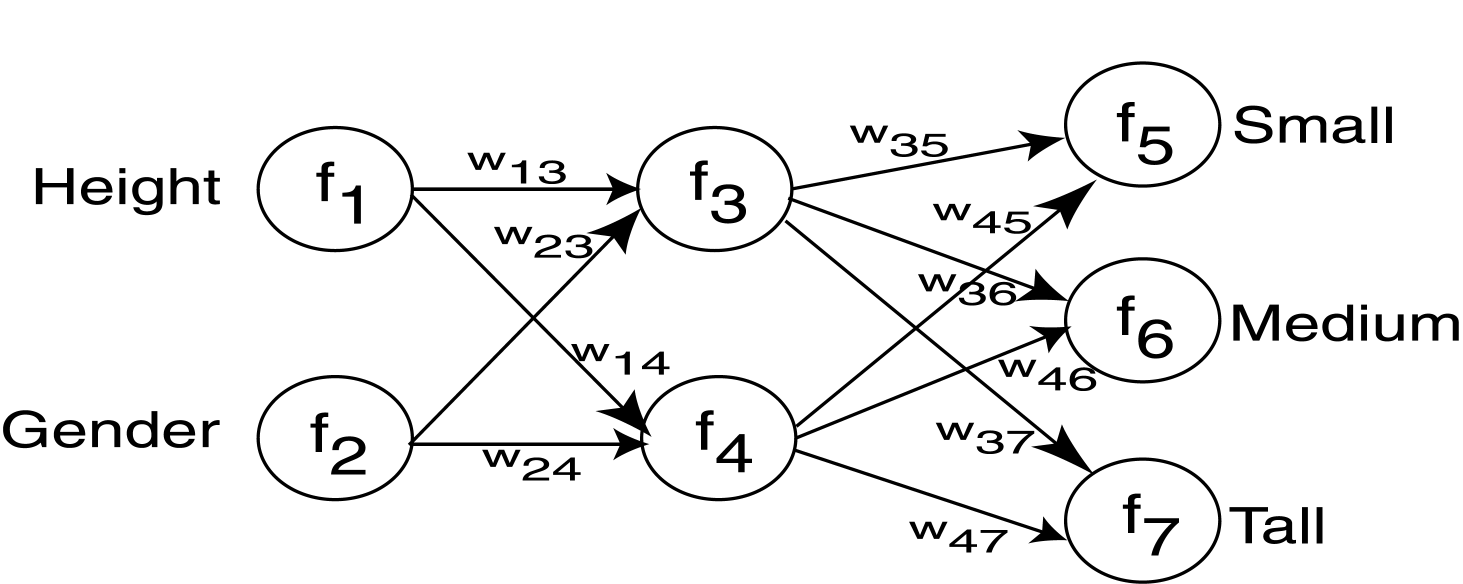
V is partitioned into a set of input nodes, VI, hidden nodes, VH, and output nodes, VO.

The vertices are also partitioned into layers

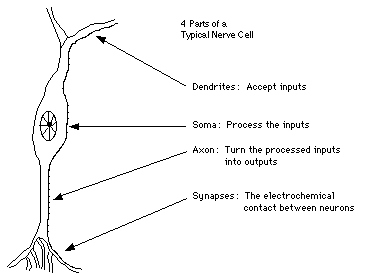
Any arc <i,j> must have node i in layer h-1 and node j in layer h.

Arc <i,j> is labeled with a numeric value wij.

Node i is labeled with a function fi.



**Similarity with Biological Network**

* Fundamental processing element of a   
  neural network is a neuron
* A human brain has 100 billion neurons
* An ant brain has 250,000 neurons

**A Neuron (= a Perceptron)**

The *n*-dimensional input vector *x* is mapped into variable *y* by means of the scalar product and a nonlinear function mapping

**Network Training**

The ultimate objective of training is to obtain a set of weights that makes almost all the tuples in the training data classified correctly.

**Steps**

Initialize weights with random values

Feed the input tuples into the network one by one

For each unit

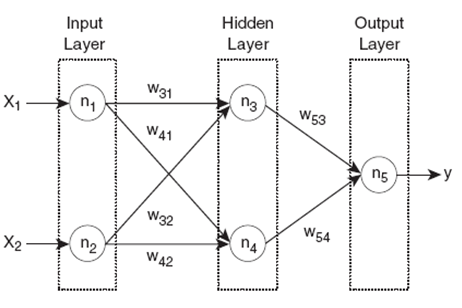
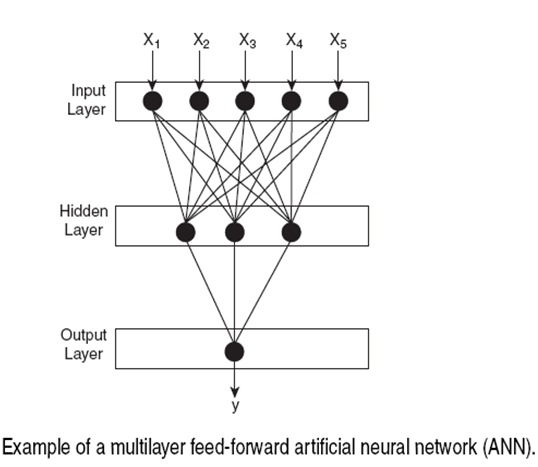
* + - Compute the net input to the unit as a linear combination of all the inputs to the unit
    - Compute the output value using the activation function
    - Compute the error
    - Update the weights and the bias

**Perceptron**

**Multi-Layer Perceptron**

**How A Multi-Layer Neural Network Works?**

* The **inputs** to the network correspond to the attributes measured for each training tuple
* Inputs are fed simultaneously into the units making up the **input layer**
* They are then weighted and fed simultaneously to a **hidden layer**
* The number of hidden layers is arbitrary, although usually only one
* The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction
* The network is **feed-forward** in that none of the weights cycles back to an input unit or to an output unit of a previous layer
* From a statistical point of view, networks perform **nonlinear regression**: Given enough hidden units and enough training samples, they can closely approximate any function



**Advantages of Neural Network**

* + prediction accuracy is generally high
  + robust, works when training examples contain errors
  + output may be discrete, real-valued, or a vector of several discrete or real-valued attributes
  + fast evaluation of the learned target function
  + High tolerance to noisy data
  + Ability to classify untrained patterns
  + Well-suited for continuous-valued inputs and outputs
  + Successful on a wide array of real-world data
  + Algorithms are inherently parallel
  + Techniques have recently been developed for the extraction of rules from trained neural networks

**Disadvantages of Neural Network**

* + long training time
  + difficult to understand the learned function (weights)
  + not easy to incorporate domain knowledge
  + Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."
  + *Poor interpretability:* Difficult to interpret the symbolic meaning behind the learned weights and of ``hidden units" in the network

**Association Rule**

* Proposed by Agrawal et al in 1993.
* It is an important data mining model studied extensively by the database and data mining community.
* Assume all data are categorical.
* No good algorithm for numeric data.
* Initially used for Market Basket Analysis to find how items purchased by customers are related.
* Given a set of records each of which contain some number of items from a given collection;

Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

**Applications:**

Basket data analysis, cross-marketing, catalog design, loss-leader analysis, clustering, classification, etc.

E.g., *98% of people who purchase tires and auto accessories also get automotive services done*

**Concepts:**

An *item*: an item/article in a basket

*I*: the set of all items sold in the store

A *transaction*: items purchased in a basket; it may have TID (transaction ID)

A *transactional* *dataset*: A set of transactions

**The model: rules**

A transaction *t* contains *X*, a set of items (itemset) in *I*, if *X* ⊆ *t*.

An association rule is an implication of the form:

*X* → *Y*, where *X*, *Y* ⊂ *I, and X* ∩*Y* = ∅

An itemset is a set of items.

E.g., X = {milk, bread, cereal} is an itemset.

A *k*-itemset is an itemset with *k* items.

E.g., {milk, bread, cereal} is a 3-itemset

**Rule Strength Measures**

Support:

The rule holds with support *sup* in *T* (the transaction data set) if sup% of transactionscontain *X* ∪ *Y*.

*sup* = Pr(*X* ∪ *Y*)

Confidence:

The rule holds in *T* with confidence *conf* if *conf*% of transactions that contain *X* also contain *Y.*

*conf* = Pr(*Y* | *X*)

An association rule is a pattern that states when *X* occurs, *Y* occurs with certain probability.

**Support and Confidence**

* *support* of  *X* in *D* is *count*(*X*)/|*D*|
* For an association rule *X⇒Y,* we can calculate

support (*X⇒Y*) = support (*XY*)

confidence (*X⇒Y)* = support (*XY*)/support (*X*)

* Relate Support (S) and Confidence (C) to Joint and Conditional probabilities
* There could be exponentially many A-rules
* Interesting association rules are (for now) those whose S and C are greater than minSup and minConf (some thresholds set by data miners)

**Support and Confidence**

Support count:

The support count of an itemset *X*, denoted by *X.count*, in a data set *T* is the number of transactions in *T* that contain *X*. Assume *T* has *n* transactions. Then,



**Mining Association Rules: Example**

For rule *A* ⇒ *C*:

support = support({*A*}∪{*C*}) = 50%

confidence = support({*A*}∪{*C*})/support({*A*}) = 66.6%

The Apriori principle:

Any subset of a frequent itemset must be frequent

**The Apriori principle:**

**Any subset of a frequent itemset must be frequent**

**Example of Association Rule**

**Frequent itemsets: Items that frequently appear together**

**I = {bread, peanut-butter}**

**I = {beer, bread}**

**Support count (σ): Frequency of occurrence of and itemset**

**σ ({bread, peanut-butter}) = 3**

**σ ({ beer, bread}) = 1**

**Support: Fraction of transactions that contain an itemset**

**s ({bread,peanut-butter}) = 3/5**

**s ({beer, bread}) = 1/5**

**Frequent itemset: An itemset whose support is greater than or equal to a minimum support threshold (minsup)**

**What’s an Interesting Rule?**

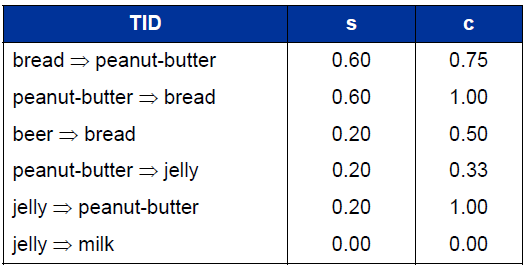
**An association rule is an implication of two itemsets:**

**X ⇒ Y**

**Many measures of interest. The two most used are:**

**Support (s): The occurring frequency of the rule, i.e., number of transactions that contain both X and Y**

**Confidence (c): The strength of the association, i e measures of how often items (X)**

****

* **Mining Association Rules:  
  What We Need to KnowGoal: Rules with high support/confidence**
* **How to compute?**

**Support: Find sets of items that occur frequently**

**Confidence: Find frequency of subsets of supported itemsets**

***If we have all frequently occurring sets of items (frequent itemsets), we can compute support and confidence!***

**The Apriori Algorithm**

**Pseudo-code:**

***Ck*: Candidate itemset of size k**

***Lk* : frequent itemset of size k**

***L1* = {frequent items};**

**for (*k* = 1; *Lk* !=∅; *k*++) do begin**

***Ck+1* = candidates generated from *Lk*;**

**for each transaction *t* in database do**

**increment the count of all candidates in *Ck+1* that are contained in *t***

***Lk+1* = candidates in *Ck+1* with min\_support**

**end**

**return ∪*k* *Lk*;**

**A-Priori Algorithm (in nutshell)**

**Clustering and Cluster Analysis**

**A c*luster* is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.**

***Clustering* is “the process of organizing objects into groups whose members are similar in some way”.**

***“Cluster Analysis* is a set of methods for constructing a (hopefully) sensible and informative classification of an initially unclassified set of data, using the variable values observed on each individual.”**

***- B. S. Everitt (1998), “The Cambridge Dictionary of Statistics”***

**Applications of Cluster Analysis**

* **Pattern Recognition**
* **Spatial Data Analysis** 
  + **Create thematic maps in GIS by clustering feature spaces**
  + **Detect spatial clusters or for other spatial mining tasks**
* **Image Processing**
* **Economic Science (especially market research)**
* **WWW**
  + **Document classification**
  + **Cluster Weblog data to discover groups of similar access patterns**

**Applications of Cluster Analysis**

* **Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs**
* **Land use: Identification of areas of similar land use in an earth observation database**
* **Insurance: Identifying groups of motor insurance policy holders with a high average claim cost**
* **City-planning: Identifying groups of houses according to their house type, value, and geographical location**
* **Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults**

**Objectives of Cluster Analysis**

**Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups**

**Types of Clusterings**

* Partitioning Clustering

A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors

Typical methods: k-means, k-medoids, CLARA (Clustering LARge Applications)

* Hierarchical clustering

A set of nested clusters organized as a hierarchical tree

Create a hierarchical decomposition of the set of data (or objects) using some criterion

Typical methods: DiAna (Divisive Analysis), AgNes (Agglomerative Nesting), BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), ROCK (RObust Clustering using linKs), CAMELEON

* Density-based Clustering

Based on connectivity and density functions

Typical methods: DBSACN (Density Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points To Identify the Clustering Structure), DenClue (DENsity-based CLUstEring )

* Grid-based Clustering
  + based on a multiple-level granularity structure
  + Typical methods: STING (STatistical INformation Grid ), WaveCluster, CLIQUE (Clustering In QUEst)
* Model-based Clustering
  + A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  + Typical methods:EM (Expectation Maximization), SOM (Self-Organizing Map), COBWEB
* Frequent pattern-based Clustering
  + Based on the analysis of frequent patterns
  + Typical methods: pCluster
* User-guided or constraint-based Clustering
  + Clustering by considering user-specified or application-specific constraints
  + Typical methods: COD, constrained clustering

**Partitioning Clustering**

**Hierarchical Clustering**

Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition

**Agglomerative (bottom up)**

1. start with 1 point (singleton)
2. recursively add two or more appropriate clusters
3. Stop when k number of clusters is achieved.

**Divisive (top down)**

1. Start with a big cluster
2. Recursively divide into smaller clusters
3. Stop when k number of clusters is achieved.

**Dendrogram**

* + Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.
  + A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

**Hierarchical Clustering**



**AGNES (Agglomerative Nesting)**

* Introduced in Kaufmann and Rousseeuw (1990)
* Implemented in statistical analysis packages, e.g., Splus
* Use the Single-Link method and the dissimilarity matrix.
* Merge nodes that have the least dissimilarity
* Go on in a non-descending fashion
* Eventually all nodes belong to the same cluster

**DIANA (Divisive Analysis)**

* Introduced in Kaufmann and Rousseeuw (1990)
* Implemented in statistical analysis packages, e.g., Splus
* Inverse order of AGNES
* Eventually each node forms a cluster on its own

**Hierarchical Clustering**

* Produces a set of nested clusters organized as a hierarchical tree
* Can be visualized as a dendrogram

A tree like diagram that records the sequences of merges or splits



**Strengths of Hierarchical Clustering**

* Do not have to assume any particular number of clusters

Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

* They may correspond to meaningful taxonomies

Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

**Hierarchical Clustering**

* Two main types of hierarchical clustering
  + Agglomerative (bottom up):

Start with the points as individual clusters

At each step, merge the closest pair of clusters until only one cluster (or k clusters) left

* + Divisive (top down):

Start with one, all-inclusive cluster

At each step, split a cluster until each cluster contains a point (or there are k clusters)

**Agglomerative Clustering Algorithm**

More popular hierarchical clustering technique

Basic algorithm

Compute the proximity matrix

Let each data point be a cluster

**Repeat**

Merge the two closest clusters

Update the proximity matrix

**Until** only a single cluster remains

**Note:**

Key operation is the computation of the proximity of two clusters

Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

* Start with clusters of individual points and a proximity matrix

Intermediate Situation

* After some merging steps, we have some clusters

Intermediate Situation

* We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.

After Merging

* The question is “How do we update the proximity matrix?”

**How to Define Inter-Cluster Similarity**

**How to Define Inter-Cluster Similarity**

* MIN
* MAX
* Group Average
* Distance Between Centroids

**How to Define Inter-Cluster Similarity**

MIN

MAX

Group Average

Distance Between Centroids

**How to Define Inter-Cluster Similarity**

**How to Define Inter-Cluster Similarity**

**Hierarchical Clustering: Problems and Limitations**

* Once a decision is made to combine two clusters, it cannot be undone
* Different schemes have problems with one or more of the following:

Sensitivity to noise and outliers

Difficulty handling different sized clusters and convex shapes

Breaking large clusters (divisive)

* Dendrogram correspond to a given hierarchical clustering is not unique, since for each merge one needs to specify which subtree should go on the left and which on the right
* They impose structure on the data, instead of revealing structure in these data.

**K-means Algorithm**

* Partitioning clustering approach
* Each cluster is associated with a centroid (center point or mean point)
* Each point is assigned to the cluster with the closest centroid
* Number of clusters, K, must be specified

The basic algorithm is very simple:

**The *k*-means partitioning algorithm.**

**Algorithm: *k*-means.** The *k*-means algorithm for partitioning, where each cluster’s center is represented by the mean value of the objects in the cluster.

**Input:**

*k*: the number of clusters,

*D*: a data set containing *n* objects.

**Output:** A set of *k* clusters.

**Method:**

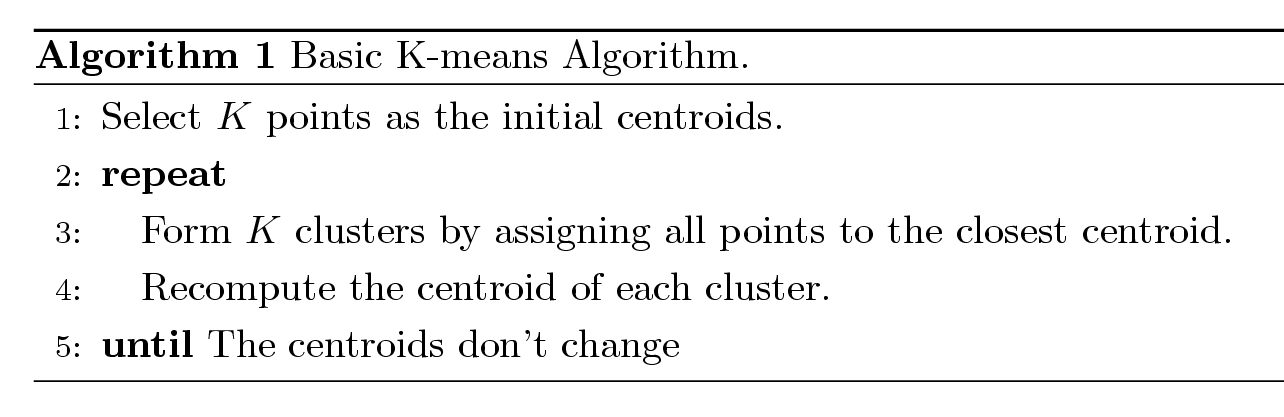
(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;

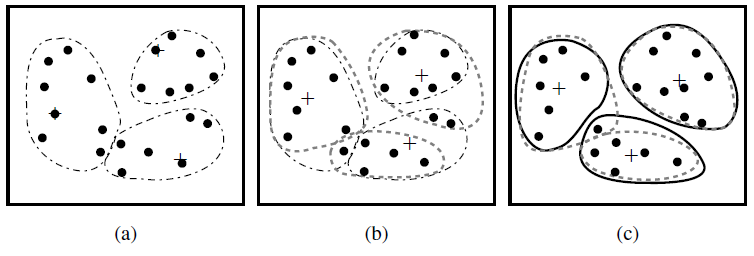
(2) repeat

(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;

(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;

(5) until no change;



**Figure:** Clustering of a set of objects based on the *k*-means method. (The mean of each cluster is marked by a “+”.)



**K-means Clustering – Details**

* Initial centroids are often chosen randomly.
  + Clusters produced vary from one run to another.
* The centroid is (typically) the mean of the points in the cluster.
* ‘Closeness’ is measured mostly by Euclidean distance, cosine similarity, correlation, etc.
* K-means will converge for common similarity measures mentioned above.
* Most of the convergence happens in the first few iterations.
  + Often the stopping condition is changed to ‘Until relatively few points change clusters’
* Complexity is O( n \* K \* I \* d )

n = number of points, K = number of clusters,   
I = number of iterations, d = number of attributes

**Issues and Limitations for K-means**

* How to choose initial centers?
* How to choose K?
* How to handle Outliers?
* Clusters different in
  + Shape
  + Density
  + Size
* Assumes clusters are spherical in vector space
  + Sensitive to coordinate changes

**K-means Algorithm**

**Pros**

* Simple
* Fast for low dimensional data
* It can find pure sub clusters if large number of clusters is specified

**Cons**

* K-Means cannot handle non-globular data of different sizes and densities
* K-Means will not identify outliers
* K-Means is restricted to data which has the notion of a center (centroid)
* Applicable only when *mean* is defined, then what about categorical data?
* Need to specify *k,* the *number* of clusters, in advance
* Unable to handle noisy data and *outliers*
* Not suitable to discover clusters with *non-convex shapes*

**Outliers**

What are outliers?

The set of objects are considerably dissimilar from the remainder of the data

Example: Sports: Michael Jordon, Randy Orton, Sachin Tendulkar ...

Applications:

Credit card fraud detection

Telecom fraud detection

Customer segmentation

Medical analysis

Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches

**How to handle Outliers?**

* The k-means algorithm is sensitive to outliers !

Since an object with an extremely large value may substantially distort the distribution of the data.

**K-Medoids:** Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

**Example:**

Use in finding Fraudulent usage of credit cards. Outlier Analysis may uncover Fraudulent usage of credit cards by detecting purchases of extremely large amounts for a given account number in comparison to regular charges incurred by the same account. Outlier values may also be detected with respect to the location and type of purchase or the purchase frequency.

The *K*-*Medoids* Clustering Method

* Find *representative* objects, called medoids, in clusters
* *PAM* (Partitioning Around Medoids, 1987)
  + starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  + *PAM* works effectively for small data sets, but does not scale well for large data sets
* *CLARA* (Kaufmann & Rousseeuw, 1990)
* *CLARANS* (Ng & Han, 1994): Randomized sampling
* Focusing + spatial data structure (Ester et al., 1995)

A Typical K-Medoids Algorithm (PAM)

PAM (Partitioning Around Medoids)

* PAM (Kaufman and Rousseeuw, 1987), built in Splus
* Use real object to represent the cluster

Select ***k*** representative objects arbitrarily

For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost ***TCih***

For each pair of ***i*** and ***h***,

If *TCih* < 0, ***i*** is replaced by ***h***

Then assign each non-selected object to the most similar representative object

repeat steps 2-3 until there is no change

PAM Clustering: Total swapping cost  *TCih=∑jCjih*

Cost calculations for example

The diagram illustrates the calculation of these six costs. We see that the minimum cost is 2 and that there are several ways to reduce this cost. Arbitrarily choosing the first swap, we get C and B as the new medoids with the clusters being {C, D} and {B, A, E}

An example

Initial five objects A, B, C, D, E, two clusters (A, C, D), (B, E), and centers {A, B}.

Evaluate swap enter A to center C.

Consider the new cost (new centers {B, C})

TCAC = CAAC + CBAC + CCAC + CDAC + CEAC

CAAC = CAB - CAA = 1 – 0 = 1

CBAC = CBB - CBB = 0 – 0 = 0

CCAC = CCC - CCA = 0 – 2 = -2

CDAC = CDC - CDA = 1 – 2 = -1

CEAC = CEB - CEB = 3 – 3 = 0

As a result, TCAC = 1 + 0 – 2 – 1 + 0 = – 2

The new center {B, C} is less costly. As a result, we should swan {A. B} to {B, C} by Medoid method

Comparison between K-means and K-medoids

The k-medoids method is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. However, its processing is more costly than the k-means method. Both methods require the user to specify k, the number of clusters.

**Unit 7 : Mining Complex Types of Data**

**Introduction**

Mining complex types of data include:

* Object data
* Spatial data
* Multimedia data
* Time-series data
* Text data
* Web data

**Spatial Data Mining**

* **Spatial data mining** is the process of discovering interesting, useful, non-trivial patterns from large spatial datasets.
* Spatial Data Mining = Mining Spatial Data Sets (i.e. Data Mining + Geographic Information Systems)
* Spatial data refer to any data about objects that occupy real physical space.
* Attributes for spatial data usually will include spatial information. Spatial information (metadata) is used to describe objects in space.
* Spatial information includes geometric metadata (e.g., location, shape, size, distance, area, perimeter) and topological metadata (e.g., “neighbor of”, “adjacent to”, “included in”, “includes”).
* Spatial data can contain both spatial and non-spatial features. Spatial data has location or geo-referenced features like:
  + Address, latitude/longitude (explicit)
  + Location-based partitions in databases (implicit)

**Spatial Data Mining**

**Spatial Data Warehouse** is an integrated, subject-oriented, time-variant, and nonvolatile spatial data repository for data analysis and decision making.

**Spatial Data Integration** is a big issue. It deals with:

Structure-specific formats (raster vs. vector-based, Object-Oriented vs. relational models, different storage and indexing, etc.)

Vendor-specific formats (ESRI, MapInfo, Integraph, etc.)

**Spatial Data Cube** is a multidimensional spatial database where both dimensions and measures may contain spatial components

Spatial Data Mining

**Special Cases**

**Spatial Classification and Spatial Trend Analysis**

* **Spatial Classification**

Analyze spatial objects to derive classification schemes, such as decision trees in relevance to certain spatial properties (district, highway, river, etc.)

*Example:* Classify regions in a province into *rich* vs. *poor* according to the average family income

* **Spatial Trend Analysis**

Detect changes and trends along a spatial dimension

Study the trend of non-spatial or spatial data changing with space

*Example:* Observe the trend of changes of the climate or vegetation with the increasing distance from an ocean

**Common Tasks dealing with Spatial Data**

* **Data focusing**

Spatial queries

Identifying interesting parts in spatial data

Progressive refinement can be applied in a tree structure

* **Feature extraction**

Extracting important/relevant features for an application

* **Classification or others**

Using training data to create classifiers

Many mining algorithms can be used

Classification, clustering, associations

**Spatial Mining Tasks**

* Spatial classification
* Spatial clustering
* Spatial association rules

**Spatial Classification**

* Use spatial information at different (coarse/fine) levels (different indexing trees) for data focusing
* Determine relevant spatial or non-spatial features
* Perform conventional supervised learning algorithms

e.g., Decision trees,

**Spatial Clustering**

* Also called **spatial segmentation**
* Use tree structures to index spatial data
* Examples: DBSCAN: R-tree, CLIQUE: Grid or Quad tree, etc.

**Input**

a table of area names and their corresponding attributes such as population density, number of adult illiterates etc.

Information about the neighbourhood relationships among the areas

A list of categories/classes of the attributes

**Output**

Grouped (segmented) areas where each group has areas with similar attribute values

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 2 | 2 | 2 | 2 |
|  |  |  |  | 2 | 2 | 2 | 2 |
|  |  |  |  | 2 | 2 | 2 | 2 |
| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| 1 | 1 | 1 | 1 |  |  |  |  |
| 1 | 1 | 1 | 1 |  |  |  |  |
| 1 | 1 | 1 | 1 |  |  |  |  |

* Spatial segmentation is performed in image processing

Identify regions (areas) of an image that have similar colour (or other image attributes).

Many image segmentation techniques are available

E.g. region-growing technique

**Region Growing Technique**

* There are many flavours of this technique
* One of them is described below:

Assign seed areas to each of the segments (classes of the attribute)

Add neighbouring areas to these segments if the incoming areas have similar values of attributes

Repeat the above step until all the regions are allocated to one of the segments

* Functionality to compute spatial relations i.e. neighbours are assumed.

**Multidimensional Analysis of Multimedia Data**

* **Multimedia Data Cube**

Design and construction similar to that of traditional data cubes from relational data

Contain additional dimensions and measures for multimedia information, such as color, texture, and shape

* The database does not store images but their descriptors.

**Feature descriptor:** a set of vectors for each visual characteristic

Color vector: contains the color histogram

MFC (Most Frequent Color) vector: five color centroids

MFO (Most Frequent Orientation) vector: five edge orientation centroids

**Layout descriptor:** contains a color layout vector and an edge layout vector

**Text mining** is the procedure of synthesizing information, by analyzing relations, patterns, and rules among textual data. These procedures contains text summarization, text categorization, and text clustering.

1. *Text summarization* is the procedure to extract its partial content reflecting its whole contents automatically.
2. *Text categorization* is the procedure of assigning a category to the text among categories predefined by users
3. *Text clustering* is the procedure of segmenting texts into several clusters, depending on the substantial relevance.

**Motivation for Text Mining**

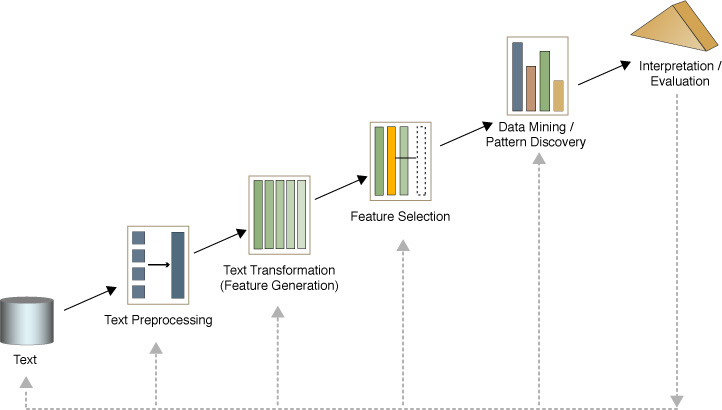
* Approximately **90%** of the world’s data is held in unstructured formats (Source: Oracle Corporation)
* Information intensive business processes demand that we transcend from simple document retrieval to “knowledge” discovery.

Text mining is well motivated, due to the fact that much of the world’s data can be found in free text form (newspaper articles, emails, literature, etc.). There is a lot of information available to mine.

While mining free text has the same goals as data mining, in general, extracting useful knowledge/stats/trends), text mining must overcome a major difficulty – there is no explicit structure.

Machines can reason will relational data well since schemas are explicitly available. Free text, however, encodes all semantic information within natural language. Our text mining algorithms, then, must make some sense out of this natural language representation. Humans are great at doing this, but this has proved to be a problem for machines.

**Text Mining Process**



**What’s Text Mining**

**Mining Text Data: An Introduction**

**Data Mining / Knowledge Discovery**

**Text Representation Issues**

* Each word has a dictionary meaning, or meanings

**Run** – (1) the verb. (2) the noun, in **cricket**

**Cricket –** (1) The game. (2) The insect.

Apple (the company) or apple (the fruit)

* Ambiguity and context sensitivity - Each word is used in various “senses”

Tendulkar made 100 runs

Because of an injury, Tendulkar can not run and will need a runner between the wickets

* Capturing the “meaning” of sentences is an important issue as well. Grammar, parts of speech, time sense could be easy!
* ***Order*** of words in the query

hot dog stand in the amusement park

hot amusement stand in the dog park

**Text Databases and IR**

**Text databases (document databases)**

Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.

Data stored is usually *semi-structured*

Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data

**Information retrieval**

A field developed in parallel with database systems

Information is organized into (a large number of) documents

*Information retrieval problem*: locating relevant documents based on user input, such as keywords or example documents

**Information Retrieval**

**Basic Measures for Text Retrieval**

**Precision:** the percentage of retrieved documents that are in fact relevant to the query (i.e., “correct” responses)

**Recall:** the percentage of documents that are relevant to the query and were, in fact, retrieved

**Application of Text Mining**

Text mining system provides a competitive edge for a company to process and take advantage of a large quantity of textual information. The potential applications are countless. We highlight a few below.

* Customer profile analysis, e.g., mining incoming emails for customers' complaint and feedback.
* Patent analysis, e.g., analyzing patent databases for major technology players, trends, and opportunities.
* Information dissemination, e.g., organizing and summarizing trade news and reports for personalized information services.
* Company resource planning, e.g., mining a company's reports and correspondences for activities, status, and problems reported.

**Text Mining vs. Data Mining**

**Product : Intelligent Miner for Text(IMT)**

**1. Feature extraction tools**

It recognizes significant **vocabulary** items in documents, and measures their importance to the document content.

**2. Clustering tools**

Clustering is used to segment a document collection into subsets, called **clusters**.

**3. Summarization tools**

Summarization is the process of **condensing a source text** into a shorter version **preserving its information content**.

**4. Categorization tool**

Categorization is used to assign objects to **predefined categories**, or **classes** from a taxonomy.

1. **Feature Extraction Tools**

1.1 Information extraction

* + - Extract linguistic items that represent document contents

1.2 Feature extraction

* + - Assign of different categories to vocabulary in documents,
    - Measure their importance to the document content.

1.3 Name extraction

* + - Locate names in text,
    - Determine what type of entity the name refers to

1.4 Term extraction

* + - Discover terms in text. Multiword technical terms
    - Recognize variants of the same concept

1.5 Abbreviation recognition

* + - Find abbreviation and math them with their full forms.

1.6 Relation extraction

**2. Clustering Tools**

2.1 Application

* + Provide a overview of content in a large document collection
  + Identify hidden structures between groups of objects
  + Improve the browsing process to find similar or related information
  + Find outstanding documents within a collection

2.2 Hierarchical clustering

Clusters are organized in a clustering tree and related clusters occurs in the same branch of tree.

2.3 Binary relational clustering

With Binary Relational Clustering, the tool finds topics hidden in a document collection and establishes *links* or *relations* between these topics.

**3. Summarization Tools**

3.1 Steps

* + the most relevant sentences → the relevancy of a sentence to a document

→ a summary of the document with length set by user

3.2 Applications

* + Judge the relevancy of a full text

Easily determine whether the document is relevant to read.

* + Enrich search results   
    The results of a query to a search engine can be enriched with a short summary of each document.
  + Get a fast overview over document collections

summary → full document

**4. Categorization Tool**

Applications

* + - Organize intranet documents
    - Assign documents to folders
    - Dispatch requests
    - Forward news to subscribers

**Mining World Wide Web (WWW)**

* The term **Web Mining** was coined by Orem Etzioni (1996) to denote the use of data mining techniques to automatically discover Web documents and services, extract information from Web resources, and uncover general patterns on the Web.
* The World Wide Web is a rich, enormous knowledge base that can be useful to many applications. The WWW is huge, widely distributed, global information service centre for news, advertisements, consumer information, financial management, education, government, e-commerce, hyperlink information, access and usage information.
* The Web’s large size and its unstructured and dynamic content, as well as its multilingual nature make extracting useful knowledge from it a challenging research problem.

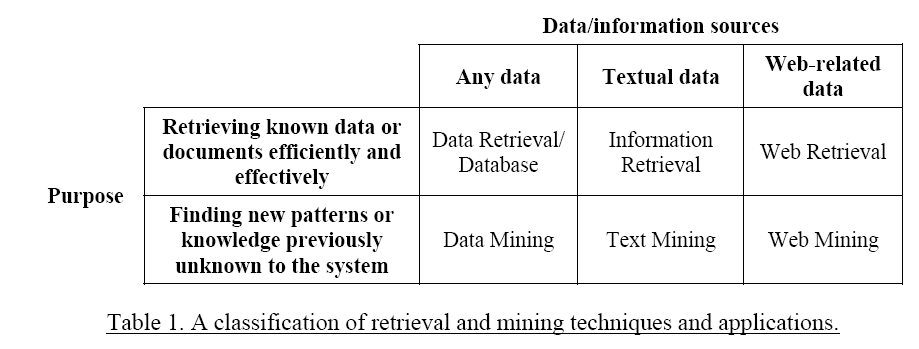
**Why Mining the World-Wide Web**

* Growing and changing very rapidly
* Broad diversity of user communities
* Only a small portion of the information on the Web is truly relevant or useful

99% of the Web information is useless to 99% of Web users

How can we find high-quality Web pages on a specified topic?

Web mining research overlaps substantially with other areas, including data mining, text mining, information retrieval, and web retrieval.



**Web Search Engines**

* **Index-based:** search the Web, index Web pages, and build and store huge keyword-based indices
* Help locate sets of Web pages containing certain keywords

**Deficiencies**

A topic of any breadth may easily contain hundreds of thousands of documents

Many documents that are highly relevant to a topic may not contain keywords defining them (polysemy)

**Web Mining: A More Challenging Task**

* Searches for

Web access patterns

Web structures

Regularity a

nd dynamics of Web contents

**Problems**

The “abundance” problem

Limited coverage of the Web: hidden Web sources, majority of data in DBMS

Limited query interface based on keyword-oriented search

Limited customization to individual users

**Web Mining Taxonomy**

Web Mining research can be classified into three categories:

**Web content mining** refers to the discovery of useful information from Web contents, including text, images, audio, video, etc.

**Web structure mining** studies the model underlying the link structures of the Web. It has been used for search engine result ranking and other Web applications.

**Web usage mining** focuses on using data mining techniques to analyze search logs to find interesting patterns. One of the main applications of Web usage mining is its use to learn user profiles.

**Mining the World-Wide Web**

**Mining the World-Wide Web**

**Mining the World-Wide Web**

**Mining the World-Wide Web**

**Mining the World-Wide Web**

**Web Usage Mining**

* Web servers, Web proxies, and client applications can quite easily capture **Web Usage data**.
  + Web server log: Every visit to the pages, what and when files have been requested, the IP address of the request, the error code, the number of bytes sent to user, and the type of browser used…
* By analyzing the Web usage data, web mining systems can discover useful knowledge about a **system’s usage characteristics** and the **users’ interests** which has various applications:
  + Personalization and Collaboration in Web-based systems
  + Marketing
  + Web site design and evaluation
  + Decision support (*e.g., Chen & Cooper, 2001; Marchionini, 2002*).
* Mining Web log records to discover user access patterns of Web pages

**Applications**

Target potential customers for electronic commerce

Enhance the quality and delivery of Internet information services to the end user

Improve Web server system performance

Identify potential prime advertisement locations

* Web logs provide rich information about Web dynamics

Typical Web log entry includes the URL requested, the IP address from which the request originated, and a timestamp

**Why Web Usage Mining?**

* Explosive growth of E-commerce

Provides an cost-efficient way doing business

Amazon.com: “online Wal-Mart”

* Hidden Useful information

Visitors’ profiles can be discovered

Measuring online marketing efforts, launching marketing campaigns, etc.

**Mining the World-Wide Web**

**Design of a Web Log Miner**

Web log is filtered to generate a relational database

A data cube is generated form database

OLAP is used to drill-down and roll-up in the cube

OLAM is used for mining interesting knowledge

* Web usage mining has been used for various purposes:

**A knowledge discovery process** for mining marketing intelligence information from Web data. *Buchner and Mulvenna (1998)*

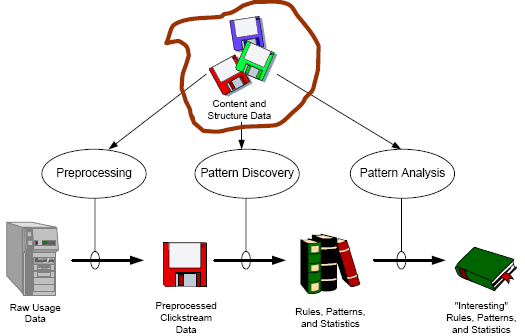
**Web traffic patterns** also can be extracted from Web usage logs in order to improve the performance of a Web site *(Cohen et al., 1998).*

**Commercial products**: *Web Trends developed by NetIQ, WebAnalyst by Megaputer and NetTracker by Sane Solutions.*

* Search engine transaction logs also provide valuable knowledge about **user behavior** on Web searching.
* Such information is very useful for a better understanding of users’ Web searching and information seeking behavior and can improve the design of Web search systems.
* One of the major goals of Web usage mining is to reveal **interesting trends and patterns** which can often provide important knowledge about the users of a system.
* The **Framework** for Web usage mining. *Srivastava et al. (2000)*
* **Preprocessing:** Data cleansing
* **Pattern discovery:**
* **Pattern analysis:**

Generic machine learning and Data mining techniques, such as association rule mining, classification, and clustering, often can be applied.

**Web Usage Mining – Procedure**



**Web Usage Mining – Model**



* Many Web applications aim to provide **personalized** information and services to users. **Web usage data** provide an excellent way to learn about users’ interest (*Srivastava et al., 2000*).

WebWatcher (*Armstrong et al., 1995*)

Letizia (*Lieberman, 1995*)

* Web usage mining on Web logs can help identify users who have accessed similar Web pages. The patterns that emerge can be very useful in **collaborative** Web searching and filtering.

*Amazon.com* uses **collaborative filtering** to recommend books to potential customers based on the preferences of other customers having similar interests or purchasing histories.

*Huang et al. (2002)* used **Hopfield Net** to model user interests and product profiles in an online bookstore in Taiwan.

**How to perform Web Usage Mining**

* Obtain web traffic data from

Web server log files

Corporate relational databases

Registration forms

* Apply data mining techniques and other Web mining techniques
* Two categories:

Pattern Discovery Tools

Pattern Analysis Tools

Pattern Analysis Tools

* Answer Questions like:

“How are people using this site?”

“which Pages are being accessed most frequently?”

* This requires the analysis of the structure of hyperlinks and the contents of the pages

Pattern Discovery Tools

* Data Pre-processing

Filtering/clean Web log files

eliminate outliers and irrelevant items

Integration of Web Usage data from:

Web Server Logs

Referral logs

Registration file

Corporate Database

* Converting IP addresses to Domain Names

Domain Name System does the conversion

Discover information from visitors’ domain names:

Ex: .ca(Canada), .cn(China), etc

* Converting URLs to Page Titles

Page Title: between <title> and </title>

Pattern Discovery Techniques

* Path Analysis

Uses Graph Model

Provide insights to navigational problems

Example of info. Discovered by Path analysis:

78% “company”-> “what’s new”->“sample”-> “order”

60% left sites after 4 or less page references

=> most important info must be within the first 4 pages of site entry points.

* Grouping

Groups similar info. to help draw higher-level conclusions

Ex: all URLs containing the word “Yahoo”…

* Filtering

Allows to answer specific questions like:

how many visitors to the site in *this week*?

Pattern Discovery Techniques

* Dynamic Site Analysis

Dynamic html links to the database, and requires parameters appended to URLs

<http://search.netscape.com/cgi-in/search?search=Federal+Tax+Return+Form&cp=ntserch>

Knowledge:

What the visitors looked for

What keywords S/B purchased from Search engineer

* Cookies

Randomly assigned ID by web server to browser

Cookies are beneficial to both web site developers and visitors

Cookie field entry in log file can be used by Web traffic analysis software to track repeat visitors 🡪 loyal customers.

Pattern Discovery Techniques

* Association Rules

help find spending patterns on related products

30% who accessed/company/products/bread.html, also accessed /company/products/milk.htm.

* Sequential Patterns

help find inter-transaction patterns

50% who bought items in /pcworld/computers/, also bought in /pcworld/accessories/ within 15 days

* Clustering

Identifies visitors with common characteristics based on visitors’ profiles

50% who applied discover platinum card in /discovercard/customerService/newcard, were in the 25-35 age group, with annual income between $40,000 – 50,000.

Pattern Discovery Techniques

* Decision Trees

a flow chart of questions leading to a decision

Ex: car buying decision tree

**Web Content Mining**

* **Text Mining for Web Documents**

Text mining for Web documents can be considered a sub-field of **Web content mining**.

**Information extraction techniques** have been applied to Web HTML documents

* + - E.g., *Chang and Lui (2001)* used a PAT tree to construct automatically a set of rules for information extraction.

**Text clustering algorithms** also have been applied to Web applications.

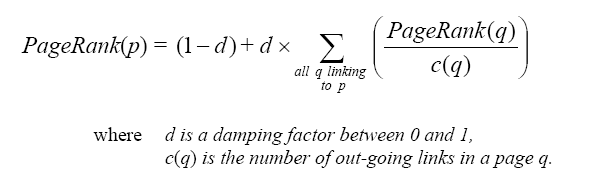
* + - E.g., *Chen et al. (2001; 2002)* used a combination of noun phrasing and SOM to cluster the search results of search agents that collect Web pages by meta-searching popular search engines.

**Web Structure Mining**

* **Web link structure** has been widely used to infer important web pages information.
* Web structure mining has been largely influenced by research in
  + **Social network analysis**
  + **Citation analysis** (bibliometrics).
    - *in-links:* the hyperlinks pointing to a page
    - *out-links:* the hyperlinks found in a page.
    - Usually, the **larger** the number of in-links, the **better** a page is.
    - By analyzing the pages containing a **URL**, we can also obtain
  + **Anchor text**: how other Web page authors annotate a page and can be useful in predicting the content of the target page.
* **Web structure mining algorithms**:

**The PageRank algorithm** is computed by weighting each in-link to a page **proportionally** to the quality of the page containing the in-link (*Brin & Page, 1998*).

The **qualities** of these referring pages also are determined by PageRank. Thus, a page *p* is calculated **recursively** as follows:



**Web structure mining algorithms:**

*Kleinberg (1998)* proposed the **HITS** (Hyperlink-Induced Topic Search) algorithm, which is similar to PageRank.

* + - **Authority pages**: high-quality pages related to a particular search query.
    - **Hub pages:** pagesprovide pointers to other authority pages.
    - A page to which many others point should be a good authority, and a page that points to many others should be a good hub.



Another application of Web structure mining is to understand the structure of the Web **as a whole**.

The core of the Web is a **strongly connected** component and that the Web’s graph structure is shaped like a bowtie. *Broder et al. (2000)*

**Strongly Connected Component** (**SCC**); 28% of the Web.

**IN**: every Web page contains a direct path to the SCC; 21% of Web

**OUT**: a direct path from SCC linking to it; 21% of Web

**TENDRILS**: pages hanging off from IN and OUT but without direct path to SCC; 22% of Web

**Isolated, Disconnected Components** that are not connected to the other 4 groups; 8% of Web

**Conclusion**

Spatial data mining is facilitated by Spatial warehousing, OLAP and mining and finds spatial associations, classifications and trends.

Multimedia data mining needs content-based retrieval and similarity search integrated with mining methods

Text mining goes beyond keyword-based and similarity-based information retrieval and discovers knowledge from semi-structured data using methods like keyword-based association and document classification.

Web mining includes mining Web link structures to identify authoritative Web pages, the automatic classification of Web documents, building a multilayered Web information base, and Weblog mining.

**Unit 8 : Research Trends in Data Warehousing and Data Mining**

**Data Mining Systems Products and Research Prototypes**

As a young discipline, data mining has a relatively short history and are constantly evolving-new data mining systems appear on the market every year; new functions, features, and visualization tools are added to existing systems on a constant basis; and efforts toward the standardization of data mining language have only just begun.

**How to Choose a Data Mining System?**

* Commercial data mining systems have little in common

Different data mining functionality or methodology

May even work with completely different kinds of data sets

* Need multiple dimensional view in selection
* Data types: relational, transactional, text, time sequence, spatial?
* **System issues**

running on only one or on several operating systems?

a client/server architecture?

Provide Web-based interfaces and allow XML data as input and/or output?

* Data sources

ASCII text files, multiple relational data sources

support ODBC connections (OLE DB, JDBC)?

* Data mining functions and methodologies

One vs. multiple data mining functions

One vs. variety of methods per function

More data mining functions and methods per function provide the user with greater flexibility and analysis power

* Coupling with Database and/or data warehouse systems

Four forms of coupling: no coupling, loose coupling, semitight coupling, and tight coupling

Ideally, a data mining system should be tightly coupled with a database system

* Scalability

Row (or database size) scalability

Column (or dimension) scalability

Curse of dimensionality: it is much more challenging to make a system column scalable that row scalable

* Visualization tools

“A picture is worth a thousand words”

Visualization categories: data visualization, mining result visualization, mining process visualization, and visual data mining

* Data mining query language and graphical user interface

Easy-to-use and high-quality graphical user interface

Essential for user-guided, highly interactive data mining

**Examples of Data Mining Systems**

* Microsoft SQL Server 2005

Integrate DB and OLAP with mining

Support OLEDB for DM standard

* IBM Intelligent Miner

Intelligent Miner is an IBM data-mining product

A wide range of data mining algorithms

Scalable mining algorithms

Toolkits: neural network algorithms, statistical methods, data preparation, and data visualization tools

Tight integration with IBM's DB2 relational database system

* SAS Enterprise Miner

SAS Institute Inc. developed Enterprise Miner

A variety of statistical analysis tools

Data warehouse tools and multiple data mining algorithms

* SGI MineSet

Silicon Graphics Inc. (SGI) developed MineSet

Multiple data mining algorithms and advanced statistics

Advanced visualization tools

* DBMiner

DBMiner Technology Inc developed DBMiner.

It provides multiple data mining algorithms including discovery-driven OLAP analysis, association, classification, and clustering

* + SPSS Clementine

Integral Solutions Ltd. (ISL) developed Clementine

Clementine has been acquired by SPSS Inc.

An integrated data mining development environment for end-users and developers

Multiple data mining algorithms and visualization tools including rule induction, neural nets, classification, and visualization tools

**Theoretical Foundations of Data Mining**

* Data reduction

The basis of data mining is to reduce the data representation

Trades accuracy for speed in response

* Data compression

The basis of data mining is to compress the given data by encoding in terms of bits, association rules, decision trees, clusters, etc.

* Pattern discovery

The basis of data mining is to discover patterns occurring in the database, such as associations, classification models, sequential patterns, etc.

* Probability theory

The basis of data mining is to discover joint probability distributions of random variables

* Microeconomic view

A view of utility: the task of data mining is finding patterns that are interesting only to the extent in that they can be used in the decision-making process of some enterprise

* Inductive databases

Data mining is the problem of performing inductive logic on databases,

The task is to query the data and the theory (i.e., patterns) of the database

Popular among many researchers in database systems

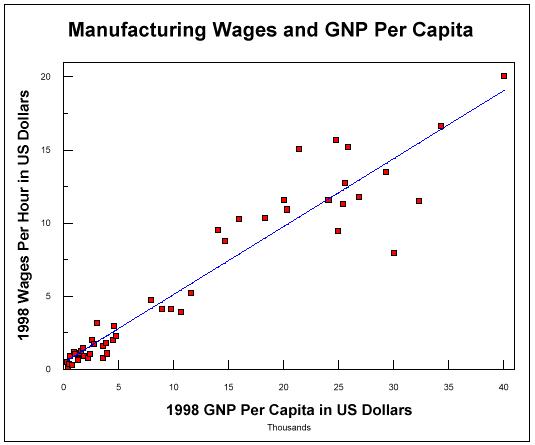
**Statistical Data Mining**

* There are many well-established statistical techniques for data analysis, particularly for numeric data

applied extensively to data from scientific experiments and data from economics and the social sciences

**Regression**

* + predict the value of a response (dependent) variable from one or more predictor (independent) variables where the variables are numeric
  + forms of regression: linear, multiple, weighted, polynomial, nonparametric, and robust



**Visual and Audio Data Mining**

Visualization: use of computer graphics to create visual images which aid in the understanding of complex, often massive representations of data

Visual Data Mining: the process of discovering implicit but useful knowledge from large data sets using visualization techniques

**Purpose of Visualization**

Gain insight into an information space by mapping data onto graphical primitives

Provide qualitative overview of large data sets

Search for patterns, trends, structure, irregularities, relationships among data.

Help find interesting regions and suitable parameters for further quantitative analysis.

Provide a visual proof of computer representations derived

* Integration of visualization and data mining

data visualization

data mining result visualization

data mining process visualization

interactive visual data mining

* Data visualization

Data in a database or data warehouse can be viewed

at different levels of granularity or abstraction

as different combinations of attributes or dimensions

Data can be presented in various visual forms

**Data Mining Result Visualization**

* Presentation of the results or knowledge obtained from data mining in visual forms
* Examples

Scatter plots and boxplots (obtained from descriptive data mining)

Decision trees

Association rules

Clusters

Outliers

Generalized rules

**Data Mining Process Visualization**

* Presentation of the various processes of data mining in visual forms so that users can see

Data extraction process

Where the data is extracted

How the data is cleaned, integrated, preprocessed, and mined

Method selected for data mining

Where the results are stored

How they may be viewed

**Interactive Visual Data Mining**

* Using visualization tools in the data mining process to help users make smart data mining decisions
* Example

Display the data distribution in a set of attributes using colored sectors or columns (depending on whether the whole space is represented by either a circle or a set of columns)

Use the display to which sector should first be selected for classification and where a good split point for this sector may be

**Audio Data Mining**

* Uses audio signals to indicate the patterns of data or the features of data mining results
* An interesting alternative to visual mining
* An inverse task of mining audio (such as music) databases which is to find patterns from audio data
* Visual data mining may disclose interesting patterns using graphical displays, but requires users to concentrate on watching patterns
* Instead, transform patterns into sound and music and listen to pitches, rhythms, tune, and melody in order to identify anything interesting or unusual

**Social Impact of Data Mining**

**Is Data Mining a Hype or Will It Be Persistent?**

* Data mining is a technology
* Technological life cycle

Innovators

Early adopters

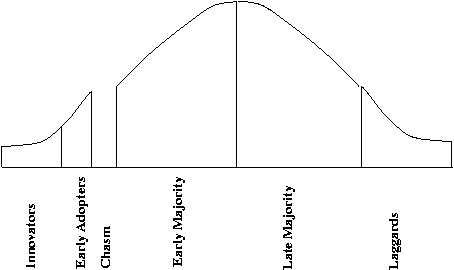
Chasm

Early majority

Late majority

Laggards

**Life Cycle of Technology Adoption**



* Data mining is at Chasm!?

Existing data mining systems are too generic

Need business-specific data mining solutions and smooth integration of business logic with data mining functions

**Social Impacts: Threat to Privacy**

* Is data mining a threat to privacy and data security?

“Big Brother”, “Big Banker”, and “Big Business” are carefully watching you

Profiling information is collected every time

You use your credit card, debit card, supermarket loyalty card, or frequent flyer card, or apply for any of the above

You surf the Web, reply to an Internet newsgroup, subscribe to a magazine, rent a video, join a club, fill out a contest entry form,

You pay for prescription drugs, or present you medical care number when visiting the doctor

Collection of personal data may be beneficial for companies and consumers, there is also potential for misuse

**Protect Privacy and Data Security**

* Fair information practices

International guidelines for data privacy protection

Cover aspects relating to data collection, purpose, use, quality, openness, individual participation, and accountability

Purpose specification and use limitation

Openness: Individuals have the right to know what information is collected about them, who has access to the data, and how the data are being used

* Develop and use data security-enhancing techniques

Blind signatures

Biometric encryption

Anonymous databases

**Trends in Data Mining**

* Application exploration

development of application-specific data mining system

Invisible data mining (mining as built-in function)

* Scalable data mining methods

Constraint-based mining: use of constraints to guide data mining systems in their search for interesting patterns

* Integration of data mining with database systems, data warehouse systems, and Web database systems
* Invisible data mining
* Standardization of data mining language

A standard will facilitate systematic development, improve interoperability, and promote the education and use of data mining systems in industry and society

* Visual data mining
* New methods for mining complex types of data

More research is required towards the integration of data mining methods with existing data analysis techniques for the complex types of data

* Web mining
* Privacy protection and information security in data mining