Medical Informatics

Prof. Dr. Nizamettin AYDIN

naydin@yildiz.edu.tr

http://www.yildiz.edu.tr/~naydin

1

Database Technology

- The purpose of a database to facilitate the management of data
  - a process that depends on
    - people
    - processes
    - the enabling technology
- Consider that the thousands of base pairs discovered every minute by the sequencing machines in public and private laboratories would be practically impossible to record, archive, and either publish or sell to other researchers without computer databases.

2

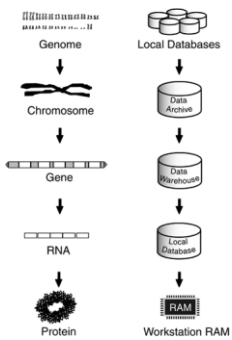
Database Technology

- The database technology empowers researchers to store their data in a way that it can be
  - quickly and easily accessed,
  - manipulated,
  - compared to other data,
  - shared with other researchers
- The volatility of the data, the concept of working memory, and the interrelatedness of data, regardless of the volume of data involved, are distinguishing features of the various forms of memory systems or databases.

3

Database Technology

- Organic Analog of Database Hierarchy.
- The database hierarchy has many parallels to the hierarchy in the human genome.
- Data stored in chromosomes, like a data archive, must be unpacked and transferred to a more immediately useful form before the data can be put to use.



4

Database Technology

- The data-archiving process involves
  - indexing,
  - selecting the appropriate software to manage the archive,
  - type of media as a function of frequency of use and expected useful life span of the data.
- From an implementation perspective, the key issues in selecting one particular archiving technology over another depends on
  - the size of the archive,
  - the types of data and data sources to be archived,
  - the intended use,
  - any existing or legacy archiving systems involved.

5

Database Technology

- A single source of data is generally much easier to work with than data from multiple, disparate sources in different and often non-compatible formats.
- In addition, hardware and software used in the archiving process should reflect the intended use of the data.
- For example,
  - seldom-used data can be archived using a much less powerful system, compared to data that must be accessed frequently.

6

Database Technology

- The simplest approach to managing bioinformatics data in a small laboratory is to establish a file server that is regularly backed up to a secure archive.
- To use the hardware most effectively, everyone connected to the server copies their files from their local hard drive to specific areas on the server's hard drive on a daily basis.
- The data on the server are in turn archived to magnetic tape or other high-capacity media by someone assigned to the task.
  - In this way, researchers can copy the file from the server to their local hard drive as needed.
- Similarly, if the server hardware fails for some reason,
  - then the archive can be used to reconstitute the data on a second server.

7

Database Technology

- From a database perspective, file servers used as archives have several limitations.
  - For example, because the data may be created using different applications, perhaps using different formats and operating systems, searching through the data may be difficult,
    - especially from a single interface other than with the search function that is part of the computer's operating system.
- Even then, there is no way of knowing what particular files hold.

8

Database Architecture

- From a structural or architectural perspective, database technology can be considered either centralized or distributed.
- In the centralized approach, typified by the data warehouse,
  - data are processed in order to fit into a central database.
- In a distributed architecture,
  - data are dispersed geographically,
    - even though they may appear to be in one location because of the database management system software.
- In each case, the goal is the same
  - providing researchers with some means of rapidly accessing and keeping track of data in a way that supports reuse.
    - This is especially critical in large biotech laboratories, where large, comprehensive patient and genomic databases support data mining and other methods that extract meaningful patterns from potentially millions of records.

9

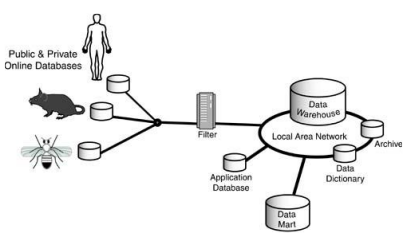
Database Architecture

- **Centralized Database Architecture**
  - concentrates all organizational activity in one location.
- This can be a formidable task, as it requires
  - cleaning,
  - encoding,
  - translation

  of data before they can be included in the central database.

10

Database Architecture



- A data warehouse isn't simply a large hard disk, but a database system implemented on a tiered storage system that reflects access time, cost, and data longevity constraints.

11

Database Architecture

- **Distributed Database Architecture**
  - characterized by physically disparate storage media.
  - supports the ability to use a variety of hardware and software in a laboratory,
    - allowing a group to use the software that makes their lives easiest, while still allowing a subset of data in each application to be shared throughout the organization.
- Separate applications, often running on separate machines and using proprietary data formats and storage facilities, share a subset of information with other applications.
  - A limitation of this common interface approach, compared to a central database, is that the amount of data that can be shared among applications is typically limited.
  - In addition, there is the computational overhead of communicating data between applications.

12

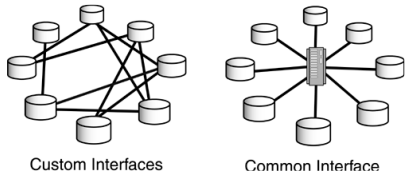
Database Architecture

- A challenge of using an integrated approach is developing the interfaces between the databases associated with each application.
- When there are only a few different applications and operating systems to contend with, developing custom interfaces between different databases may be tenable.
- However, with multiple applications and their associated databases, the number of custom interfaces that must be developed to allow sharing of data becomes prohibitive.
- A better solution to integrating incompatible databases is to write interfaces to a common standard.

13

Database Architecture

- Distributed Database Integration.



- Distributed databases can be configured to share data
  - through dedicated, one-to-one custom interfaces (left)
  - by writing to a common interface standard (right).
- Custom interfaces incur a work penalty on the order of two times the number of databases that are integrated.

14

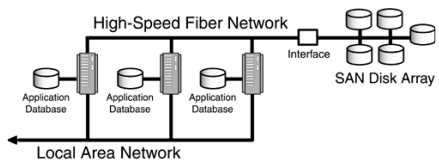
Database Architecture

- Full database integration is much more than simply moving data to a single hard disk.
  - A file server can store data from dozens of various applications and yet have no integration between applications.
- Similarly, just as a single hard disk can be formatted so that it appears as several logical volumes or drives, a distributed physical architecture can function like a logical centralized database.
- Taking this analogy one step further, there are hybrid database architectures that combine aspects of centralized and distributed architectures to provide enhanced functionality or reduced cost.
  - For example, the Storage Area Network (SAN) architecture is based on a separate, dedicated, high-speed network that provides storage under one interface (next slide).

15

Database Architecture

- Storage Area Network Architecture.



- A SAN is a dedicated network that connects servers and SAN-compatible storage devices.
- SAN devices can be added as needed, within the bandwidth limitations of the high-speed fiber network.

16

Database Architecture

- In addition to SANs, there is a variety of other network-dependent database architectures.
  - Network Attached Storage (NAS) is one method of adding storage to a networked system of workstations.
    - To users on the network, the NAS acts like a second hard drive on their workstations.
    - A NAS device, like a file server, must be managed and archived separately.
- A similar approach is to use a Storage Service Provider (SSP),
  - which functions as an Application Service Provider (ASP) with a database as the application.

17

Database Architecture

- With the increased reliance on the Internet, outsourcing storage through Internet-based SANs and SSPs is often used instead of purchasing huge servers in-house.
- The advantage of technologies such as SANs and SSPs is that they can provide virtually unlimited storage as part of huge server farms that may be located in geographically disparate areas.
- The downside is loss of control over the data and archiving process, as well as the risk that company providing the service may fail, resulting in the loss of valuable research and production data.
- In addition, like NAS, SANs and SSPs only address additional storage space, not integration.

18

Database Management Systems (DBMS)

- the set of software tools that works with a given architecture to create a practical database application.
- the interface between the low level hardware commands and the user,
  - allowing the user to think of data management in abstract, high-level terms using a variety of data models, instead of the bits and bytes on magnetic media.
- also provides views or high-level abstract models of portions of the conceptual database that are optimized for particular users.
- shields the user from the details of the underlying algorithms and data representation schemes.

19

Database Management Systems (DBMS)

- facilitates use by maximizing the efficiency of managing data with techniques
  - such as dynamically configuring operations to make use of a given hardware platform.
    - For example, a DBMS should recognize a server with large amounts of free RAM and make use of that RAM to speed serving the data.
- ensures data integrity by imposing data consistency constraints,
  - such as requiring numeric data in certain fields, free text in others, and image data elsewhere.
    - A researcher isn't allowed to insert a numerical sequence in the space assigned for a nucleotide sequence, for example.
- guards against data loss.
    - For example, a DBMS should support quick recovery from hardware or software failures.
- adds security to a database,
  - a properly constructed DBMS allows only users with permission to have access to specific data, normally down to the level of individual files.

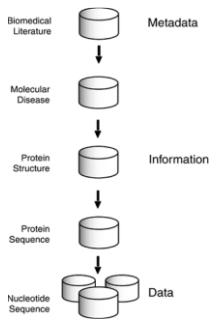
20

Database Management Systems (DBMS)

- A key issue in working with a DBMS is the use of metadata.
- For example,
  - one way to think about the application of metadata is to consider the high level biomedical literature a means of simplifying and synthesizing the underlying complexity of molecular disease, protein structure, protein alignment, and protein and DNA sequence data.
  - From this perspective, data are base pair identifiers derived from observation, experiment, or calculation, information is data in context, such as the relationship of DNA sequences to protein structure, and metadata is a descriptive summary of disease presentations that provides additional context to the underlying information.
- The use of metadata as an organizational theme makes the centralized data management approach easier to maintain and control.

21

Database Management Systems (DBMS)



- Metadata, Information, and Data in Bioinformatics.
  - Metadata labels, simplifies, and provides context for underlying information and data.

22

Database Management Systems (DBMS)

- DBMS can be described using three levels of abstraction:
  - the physical database,
  - the conceptual database,
  - the views.
- The point of using these abstractions is that they allow researchers to manipulate huge amounts of data that may be associated in very complex ways by shielding database designers and users from the underlying complexity of computer hardware.
- The physical database is the low-level data and framework that is defined in terms of media, bits, and bytes.
- This low-level abstraction is most useful for anyone who has to deal directly with data and files.

23

Database Management Systems (DBMS)

- The conceptual database is concerned with the most appropriate way to represent the data.
  - This level of abstraction more closely approximates the needs of database designers who deal with DBMS data representation and efficiency issues such as the data-dictionary design.
- The conceptual database is defined in terms of data structures (an organizational scheme, such as a record) and the properties of the data to be stored and manipulated.
  - The most common methods of representing the conceptual database are the entity-relationship model and the data model.

24

Database Management Systems (DBMS)

- The entity-relationship model focuses on entities and their interrelationships in a way that parallels how we categorize the world.
  - For example, common database entities in bioinformatics are the human being, protein sequences, nucleotide sequences, and disease processes about which data are recorded.
  - Similarly, every entity has some basic attribute, such as name, size, weight (a particular protein may have a known weight), or charge.
- Relationships within the model are classified according to how data are associated with each other, such as
  - one-to-one,
  - one-to-many,
  - many-to-many.
    - For example, a length of DNA may be translated to one mRNA sequence (a one-to-one relationship) and a gene may give rise to several proteins (a one-to-many relationship).
  - These and other relationships can be used to maintain the integrity of data.

25

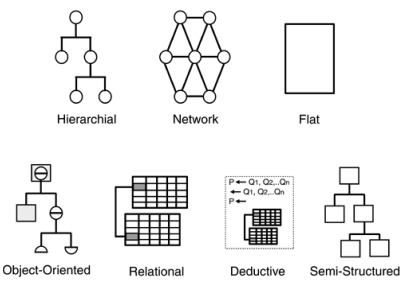
Database Management Systems (DBMS)

- The conceptual database can also be represented as a data model.
- Data models provide a means of representing and manipulating large amounts of data.
- A data model consists of two components
  - a mathematical notation for expressing data and relationships,
  - operations on the data that serve to express manipulations of the data.
- data models may also contain a collection of integrity rules that define valid data relationships.
- These various components work together to provide a formal means of representing and manipulating data.

26

Database Management Systems (DBMS)

- The most common data models supported by DBMS products are flat, network, hierarchical, relational, object-oriented, and deductive data models,



27

Database Management Systems (DBMS)

- The flat data model is simply a table without any embedded structure information to govern the relationships between records.
- As a result, a flat file database can only work with one table or file at a time.
  - Strictly speaking, a flat file doesn't really fit the criteria for a data model because it lacks an embedded structure.
  - However, the lack of an embedded structure is one reason for the popularity of the flat file database in bioinformatics, especially in capturing sequence data.
    - A sequence of a few dozen characters may be followed by a sequence of thousands of characters, with no known relationship between the sequences, other than perhaps the tissue sample or sequence run.
    - As such, a separate flat file can be used to efficiently store the sequence data from each sample or run.
- In order to make the management of large amounts of sequence or other data more tenable, a model with an embedded structure is required.

28

Database Management Systems (DBMS)

- The relational model is based on the concept of a data table
  - in which every row is unique.
- The records or rows in the table are called tuples;
  - the fields or columns are variably referred to attributes, predicates, or classes.
- Database queries are performed with the select operation,
  - which asks for all tuples in a certain relation that meet a certain criterion.

29

Database Management Systems (DBMS)

- To connect the data of two or more relations, an operation called a join is performed.
- A record is retrieved from the database by means of a key, or label,
  - that may consist of a field, part of a field, or a combination of several fields.
- A useful feature of the relational model is that records or rows from different files can be combined as long as the different files have one field in common.
  - The price paid for this flexibility is extended access time.

30

Database Management Systems (DBMS)

- In the hierarchical model, permanent hierarchical connections are defined when the database is created.
- Within the hierarchical database model, the smallest data entity is the record.
  - Unlike records in a relational model, records within a hierarchical database are not necessarily broken up into fields.
  - In addition, connections within the hierarchical model do not depend on the data.
- The hierarchical links, sometimes called the structure of the data, can best be thought of as forming an inverted tree,
  - with the parent file at the top and children files below.
- The relationship between parent and children is a one-to-many connection, in that one parent may produce multiple children.

31

Database Management Systems (DBMS)

- Because of the storage inefficiency of the hierarchical model for some types of data, the network model was developed in the late 1960s.
  - The network model is more flexible than the hierarchical one because multiple connections can be established between files.
  - These multiple connections enable the user to gain access to a particular file more effectively,
    - without traversing the entire hierarchy above that file.
- The network model is based on a many-to-one relationship.
  - The network model is significant in bioinformatics in that it may play a significant role in the architecture of the Great Global Grid and other Web-based computing initiatives.

32

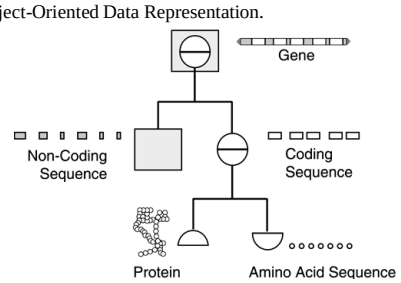
Database Management Systems (DBMS)

- In the object-oriented model, complex data structures are represented by composite objects,
  - which are objects that contain other objects.
- These objects may contain other objects in turn,
  - allowing structures to be nested to any degree.
- This metaphor is especially appealing to those who work with bioinformatics data
  - because this nesting of complexity complements the natural structure of genomic data (next slide).

33

Database Management Systems (DBMS)

- Object-Oriented Data Representation.



- The OO data model is natural for hiding the complexity of genomic data.

34

Database Management Systems (DBMS)

- The OO model combines the natural structure of the hierarchical model with the flexibility of the relational model.
  - The major advantage of the OO model is that it can be used to represent complex genomic information, including non-record-oriented data,
    - such as textual sequence data and images.
  - With an OO DBMS, it's possible to use arbitrary data types, and complex relationships can be queried without having to create resource-intensive joins between tables.
- The OO model is considered optimum for handling genomic data,
  - because it allows combinations of data to be treated as single entities.
    - Instead of thinking about a gene with exons, introns, mRNA, nucleotide sequences, associated proteins, and their 3D shapes as a separate sound file, a separate video file, and a separate text document, researchers can simply work with the gene object.

35

Database Management Systems (DBMS)

- Although the OO approach holds great promise in bioinformatics, it still lags far behind relational technology in the global database market.
- In addition, because of the flexibility and power of the relational design, many of the OO DBMS products on the market are based on extensions of commercial relational database packages.
- Because of the added overhead, the performance of these hybrid object-oriented systems is necessarily less than that of either a pure relational or an OO system.

36

Database Management Systems (DBMS)

- The deductive model is an extension of the relational database with a logic programming interface based on the principles of logic programming.
  - The logic programming interface is composed of rules, facts, and queries, using the relational database infrastructure to contain the facts.
- The database is termed deductive because from the set of rules and the facts it is possible to derive new facts not contained in the original set of facts.
- Unlike logic programming languages such as PROLOG, which search for a single answer to a query using a top-down search, deductive databases search from bottom-up, starting from the facts to find all answers to a query.
  - For example, using the format "patient (Patient ID, Sex, Mother Carrier, Father Trait Trait)," data in the deductive database describing a sex-linked recessive gene such as red-green color blindness could be represented in a relational table as in the next slide.

37

Database Management Systems (DBMS)

- Data for a Deductive Database (example)

Patient ID	Sex	Mother Carrier	Father Trait
001	Male	Yes	Yes
002	Female	Yes	No
003	Male	No	Yes
004	Female	No	Yes
005	Male	Yes	No
006	Male	No	No
007	Female	Yes	Yes

- A relevant rule in a deductive database would be:
  - Potential Carrier (Sex = Female) AND (Mother Carrier = Yes)

38

Database Management Systems (DBMS)

- The patient is a potential carrier if the sex of the patient is female and the patient's mother is a known carrier.
- Males with the gene exhibit the disease, or red-green color blindness. However, because the gene involved in color blindness is maternal, then the state of the father's color acuity is irrelevant.
- The query:

  ←Patient ID (X, potential carrier)
- would return the list of patients that should be tested for the genetic anomaly,
  - in this case Patient 002 and Patient 007 from the Table in previous slide.
- Despite the obvious uses of deductive databases in bioinformatics, most deductive databases are either academic projects or internally developed and have yet to enter the ranks of commercial relational database products.

39

Database Management Systems (DBMS)

- The semi-structured model is a hybrid between a flat file and a hierarchical model, typically written as a text document in XML.
- The major advantage of the semi-structured model is the ability to revise the structure to match new requirements on the-fly.
  - However, there is a likely repetition of data.
- Regardless of the model, at the highest level of abstraction of the DBMS is the view
  - views are abstract models of portions of the conceptual database.
  - Each view describes some of the database entities, attributes, and relationships between entities in a format convenient for a specific class of user or application.
    - For example, researchers in a pharmacogenomic firm working with an application to report sequencing results do not need to know about patient findings.
- The view abstraction has application in user interface design.

40

Interfaces

- Databases communicate with devices and users through external and user interfaces.
- Getting data into a database can come about programmatically as in the creation of a data warehouse or data mart through processing an existing database.
- More often, the data are derived from external sources,
  - such as user input through keyboard activity, or devices connected to a computer or network.

41

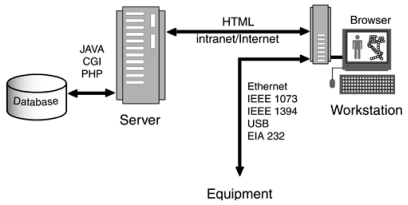
Interfaces

- Common sources of input data include mouse and keyboard activity, voice recognition, bar-code readers, wireless devices, and RF-ID tags.
- Electronic data recorders, sequencing machines, and a variety of test equipment can also provide data for inclusion in the database, according to device communications standards.
- A variety of standards, such as the IEEE 1073 Point of Care Medical Device Communications standard, define the format, speed, and protocol of communications between workstations and external devices (next slide).

42

Interfaces

- External Interfaces



- Databases communicate with equipment and users through a variety of external interfaces.

Interfaces

- Getting data into a database is of little value unless the data can also be retrieved.
- The most common methods for extracting data from a database are based on the Internet or an intranet and languages such as the Common Gateway Interface (CGI), the Hypertext Processor (PHP), and Java.
- In each case, the user issues a command from the workstation that is interpreted in the server.
- Results of the database query are then processed by language system and HTML is sent to the user's browser.
  - In this scenario, the computational overhead is borne by the server.

Interfaces

- Regardless of the language used to extract data from a database, the data have to be displayed on the user's monitor in an appropriate, understandable, and attractive way.
- This component of the user interface is most easily handled with a separate style sheet that defines the characteristics for the display device.
- In this paradigm, data to be displayed are first extracted from the database and coded in XML, a markup language for the Web that classifies content, but doesn't define how it should be displayed.
- A separate style sheet, in the form of an Extensible Stylesheet Language (XSL) document, specifies how the data are to be displayed in the user's browser.

Implementation

- Even with all of the public-domain databases accessible through the Internet, there will be research tasks that necessitate using a local database.
- The reasons vary from a need to collect, analyze, and publish sequence information inside a small laboratory to establishing a massive data warehouse as part of a pharmacogenomic R&D effort.
- In either case, the general issues and challenges are the same (next slide).

Bioinformatics Database Implementation Issues

Issue	Description
Accessibility	Ease of use, support for multiple mental models and database abstractions
Archiving	Support for the archival process, including software and hardware, and offsite storage facilities
Capacity	Local and remote data storage capacity, including space for expansion of the database
Connectivity	Connectivity through local and wide area networks, intranets, and the Internet
Control	Internal vs. third-party control of data, which may be an issue with storage service providers and other Internet-based commercial storage options
Cost	Initial, operating, and indirect (need to upgrade current network hardware and software, purchase additional peripherals) costs

Bioinformatics Database Implementation Issues

Issue	Description
Data Dictionary	Design, implementation, and maintenance of the data dictionary
Data Formats	Data formats supported by the database
Data Input	Hardware, software, and processes involved in feeding data into the database, from keyboard and voice recognition to direct instrument feed and the Internet
Data Model	Flat files, relational, hierarchical, network, object-oriented, or semistructured
Dependencies	Dependence on primary databases for populating the database, especially regarding update frequency provision for validating data to minimize propagation of errors

Bioinformatics Database Implementation Issues

Issue	Description
DBMS Software	Robustness, scalability, performance, cost, vendor reputation, support available (if open source)
Disaster Recovery	Procedural, hardware, and software provisions for disaster recovery, including error recovery mechanisms
Export/Import Capabilities	Provisions for importing and exporting data to and from different file formats
Hardware Requirements	Hard disks, controllers, backup hardware, production and staging servers for large database projects
Indexing	Indexing methodology, including selection and use of the most appropriate controlled vocabulary
Integration	Integration with other databases

49

Bioinformatics Database Implementation Issues

Issue	Description
Intellectual Property	Ownership of sequence data, images, and other data stored in and communicated through the database
Interfaces	Connectivity with other databases and applications
Legacy Systems	How to deal with legacy data and databases
Licensing	For vendor-supplied database systems, the most appropriate licensing arrangement
Life Span	The MTBF for the hardware as well as the likely useful life of the data
Load Testing	The maximum number of simultaneous users that can be supported by the DBMS
Maintenance	Cost and resource requirements
Media	The most appropriate disk, tape cartridges, and CD-ROM media

50

Bioinformatics Database Implementation Issues

Issue	Description
Normalization	Avoiding errors by representing data one way, one time, and in one place
Operating Environment	Ensuring proper power and operating temperature and humidity
Operating System	UNIX, Linux, Windows, MacOS, or mini/mainframe OS
Output	Format of database output
Performance	Access time and data throughput
Privacy	Provision for preserving confidentiality of data
Query Language	Proprietary or standard query language
Resource Requirements	Hardware, software, and operating and development personnel

51

Bioinformatics Database Implementation Issues

Issue	Description
Redundancy	Hot backups, shadowing, and RAID systems
Scalability	Ability to handle greater data volume with added hardware and/or software upgrades
Security	Limits on user access, from username-password combinations to biometrics, as well as encryption of sessions
Stand-Alone vs. Network	And multi- vs. single user
Standards	From media format to operating system, query language, and data models
Utilities	Availability of software tools for data recovery
Vendor Viability	Commercial viability of the hardware and software vendors supplying database tools and platform

52

Implementation

- For example, a milestone in designing and implementing a database is defining the type of data to be stored.
- This decision will then imply the most appropriate data model and type of DBMS to employ.
  - If the data are nucleotide sequences, then a reasonable choice would be a semi-structured database based on XML-tagged text files.
  - However, if the data are images of 3D protein structures and keywords, then either an object-oriented or a relational database would likely be more appropriate.
- Even though the representation of rows and columns may not be optimum for mapping protein structures onto a database, factors such as support from a commercial relational database vendor and support might dictate use of a relational product.

53

Implementation

- Consider the process involved in creating a central data warehouse of a scale appropriate for the pharmacogenomic laboratory.
- The six-stage process usually involves the following phases:
  - planning;
  - data consolidation;
  - data transformation;
  - selective archiving;
  - data distribution;
  - ongoing maintenance.

54

Implementation

- In the planning stage, representatives from administration, R&D, and information technology departments decide exactly what to include in the data warehouse.
  - Ideally, the data warehouse content should reflect the questions likely to be asked.
- In the consolidation phase, the selected data from each application database are restructured.
  - This typically involves adding fields and relations to reflect how the data will be used in the data warehouse.
  - The goal in the consolidation phase is to provide an efficient framework that supports queries likely to be asked, as determined in the planning stage.

55

Implementation

- The data transformation stage of data warehouse development involves transforming the consolidated data into a more useful form through summarization and packaging.
  - In summarization, the data are
    - selected, aggregated, and grouped into views more convenient and useful to users.
  - Packaging involves using the summarized data as the basis of graphical presentations, animations, and charts.

56

Implementation

- Selective archiving involves moving older or infrequently accessed data to tape, optical, or other longterm storage media.
  - Archiving saves money by sparing expensive magnetic, high-speed storage, and minimizes the performance hit imposed by locally storing data that is no longer necessary for outcomes analysis.
- The distribution phase makes data contained in the data warehouse available to users.
  - Providing for distribution encompasses front-end development so that users can easily and intuitively request and receive data, whether in real-time or in the form of routine reports.
  - Push technologies, including email alerts, can be used to distribute data to specific users.
  - The Web is also a major portal for accessing the data.

57

Implementation

- Maintenance is the final, ongoing stage of data warehouse development.
- However, creating a data warehouse involves much more than simply designing and implementing a database.
- Even if there is a process in place for extracting, cleaning, transporting, and loading data from sequence machines, bibliographic reference databases, and other molecular biology applications, and distribution tools are both powerful and intuitive, the data warehouse may not be sustainable in the long-term.
  - For example, the process of extracting, cleaning, and reloading data can be prohibitively expensive and time-consuming.
- A sustainable data warehouse provides a real benefit to users to the degree that not only is the return worth the original development,
  - but that it is valuable enough to warrant continual redesigning and evaluation to meet changing demands.

58

Infrastructure

- From a hardware perspective, implementing a database requires more than servers, large hard drives, perhaps a network and the associated cables and electronics.
- Power conditioners and uninterruptible power supplies are needed to protect sensitive equipment and the data they contain from power surges and sudden, unplanned power outages.
- Providing a secure environment for data includes the usual use of username and passwords to protect accounts.
- However, for higher levels of assurance against data theft or manipulation, secure ID cards, dongles, and biometrics (such as voice, fingerprint, and retinal recognition) may be appropriate.

59