Figure 14.2 Conversion of E-R diagram to relational, network, and hierarchical models.

Concept in E-R Model	Conversion to Relational Model	Conversion to Network Model	Conversion to Hierarchical Model
Entity set	Relation	Record type	Record type
1:1 and 1:N binary relationship	As a relation including the primary keys of entities involved or by appending the primary key of the "1" side relation in the other relation.	As a set type where the owner record type is the "1 side."	As a hierarchy with the parent is the "1 side" record type.
M:N relationship	As a relation including the primary keys of the entities in the relationship.	Introduce an intermediate record type that is a common member of two set types owned by the record types corresponding to the M:N relationship.	For symmetric access use two hierarchies with either duplication or use of virtual records.
IS_A relationship	As a relation for the superclass and each of the subclasses with the primary key of the superclass included, or as a relation for each subclass with the attributes of the superclass included.	A 1:1 set type with mandatory membership. At most one member record type occurrence is enforced by the application program.	A hierarchy with at most one occurrence of the child record type. The constraint of at most one record occurrence at the child level is enforced by the application program.

for attributes that do not apply to a given instance of the entity. An additional attribute to indicate the type of the tuple could be used in case the generalization-specialization is disjoint. For an overlapping generalization-specialization, a Boolean attribute for each possible type may be included. In this way, the nonrelevant attributes may be ignored.

If care is taken in the preliminary design to normalize the records, the database will satisfy structural constraints. To meet the performance requirements, a number of indexes will have to be generated for each relation. The attributes used in generating the indexes depend on the types of access required.

14.6.2 Designing the Conceptual Database—Network DBMS

In the network model, the entity is represented as a record type. A weak entity is represented as a set type where the strong entity is the owner record type. Alternatively, the weak entity type may be represented as a repeating group within the record type for the strong entity. A 1:N relationship is represented as a set type where the record type corresponding to the "1 side" is the owner record type. The attribute of the relationship is combined with the attributes of the member record type. However, if the member record type participates in more than one set, the representation of the relationship requires the introduction of a record type to hold the attributes of the relationship. This newly introduced record type now becomes the common member in two sets. One is a 1:N set involving the original owner record type as owner and the new record type as member. The other is a 1:1 set involving the original member record type as an owner and the new record type as member. An N:M relationship is represented by two set types involving an intermediate record type as the member. The intermediate record type represents the attributes of the relationship.

The *IS_A* relationship representing a generalization-specialization hierarchy of the E-R diagram is represented by a 1:1 set type where set membership is mandatory. The fact that a set can have only one member occurrence is enforced by the application program.

14.6.3 Designing the Conceptual Database—Hierarchical DBMS

In the hierarchical model, each entity type is represented by a record type. A 1:1 or a 1:N relationship is represented as a hierarchy with the record type for the "1 side" being the parent. Optionally the child record type is represented as a part of the parent record type. A weak entity is represented as a child record type in a hierarchy where the record type for the strong entity is the parent or as a repeating group in the strong entity record type. An N:M relationship is represented by duplications or by use of virtual records.

The *IS_A* relationship representing a generalization-specialization hierarchy of the E-R diagram is represented by a 1:1 hierarchy, the constraint that there can be only one child record type occurrence being enforced by the application program.

14.6.4 Designing the Physical Database

The primary keys of the records included in the database are chosen during the preliminary logical database design. The physical design includes decisions regarding the following aspects of the physical database:

- The choice of clustering of records
- The choice of the file organization
- The choice of supporting indexes
- The provision of links between records

Here the intent is to choose appropriate storage structures and access aids for optimum performance of the database system. Direct access is required where the file has a high rate of insertions and deletions and indexed-sequential access is suitable for a stable file.

Performance is measured in response time for online queries such as airline reservations or banking applications, or turnaround time for application programs such as payroll preparation. Performance depends on the size of records, the amount of data and its distribution on a number of storage devices, the presence of various indexes or direct access mechanisms.

For a given system the file structures that may be used are usually dictated by the DBMS. The expected types and frequencies of data manipulation operations are used to determine access aids that would be effective. If an attribute such as address is normally used for retrieval in an online system, a direct access path based on this attribute may be implemented.

Special care is taken to define indexes in a relational system for attributes participating in join operations. The storage structure and indexes may have to be modified during the fine tuning of the system, once it becomes operational and supports day-to-day operations.

Physical storage strategy includes decisions regarding the partitioning of a record into vertical, horizontal, or mixed fragments. **Vertical fragmentation** is appropriate if some of the record's fields are accessed more frequently than others. By removing the less frequently used fields along with the primary key into a separate record on a different physical file, the volume of data transfer is reduced. This would also be applicable if the vertical fragments were rarely used simultaneously. **Horizontal fragmentation** is appropriate if some occurrences of a record are more frequently used than others.

A strategy used in a relational system is to store the join of two relations, or at least those attributes of the joined relations that are frequently required. However, this strategy requires that all update operations must maintain the consistency of the database by updating such duplicated attributes.

In a relational database, a number of indexes are created for each record. The records themselves may be stored in a serial manner. The attributes used for creating secondary indexes are determined from the processing requirements of the database. Alternately, performance requirements may dictate that if a relation is retrieved using its primary key, which is also used for join operations, the relation may be stored using direct file structure. If the relation is to be stored as a sequential file, the ordering is on the attribute that is used frequently in retrieving tuples from the relation or for performing join operations. If more than one such attribute is needed, the relation may be stored in a serial file and secondary indexes created for each such attribute. The advantages of a serial file are ease of growth and shrinkage of the file size.

In a network system, access to member record types can be improved by storing the members close to the owner record type. However, if a record type participates as a member in more than one set type, this scheme is possible for only one set,

Key Terms

information requirements
processing requirements
integrity constraints
centralized scheme design
view-integration approach
top-down approach
bottom-up approach
vertical fragmentation
horizontal fragmentation

Exercises

14.1 Videobec is the leading corporation in the growing video rental business. It has the largest number of stores and prides itself on having the most comprehensive list of video movies and games. It also rents VCRs and video cameras to its members. As a convenience, it repairs video equipment, the actual work being contracted out to a number of repair shops who reap 80% of the repair charge. Each of Videobec's stores is run by a manager and assistant manager who are full-time employees. In addition, each store hires its own part-time help who are paid on a hourly basis.

The membership privilege is extended to customers for a period of one year and is renewable, unless a member has been habitually tardy in returning items borrowed. A member is allowed to rent up to 12 movie titles, 6 video games, 1 VCR, and 1 video-camera simultaneously. Movies and games can be returned to any store, but a VCR or video camera has to be returned to the store from which it was borrowed. Members have access to the online catalogue of titles and may reserve titles. A reserved title has to be picked up before 6 P.M., after which time the reservation is automatically canceled. Items are charged per day and borrowed items have to be returned before noon. Any late return bears a charge of one additional day. A discount of 20% is awarded on weekdays for all items rented. A total discount of 33% is also given on movie rentals on weekdays when more than three titles are borrowed at one time.

Movies are held by Videobec in both VHS and Beta format. The catalogue of movies contains the title of the movie, the studio or producer, the director, two leading actors, the category of the movie, number of cassettes per copy, and charge per day. The video grames catalogue contains the name of the game, the game system, and the charge per day. Videobec carries multiple copies of the same title, and a store could have been assigned any number of copies of each title. A store that has more copies of a given title than assigned to it will return these at the end of each week to Videobec's head office, which redistributes them to appropriate stores.

You are required to design and implement the database for Videobec's operational data using an appropriate DBMS package. Prepare a report documenting your design, including an E-R model of the database. The implementation of the database is to be made using the chosen DBMS on an appropriate computer system.

Your database implementation should allow the following types of queries to be made:

- Add new titles, equipment, stores, members, employees, part-time employees, repair shops.
- Remove titles, equipment stores, members, employees, part-time employees, repair shops.
- Update appropriate attributes of titles, equipment, stores. members, employees, part-time employees, repair shops.
- Show status of a member, including titles borrowed and amount outstanding for items rented.

- Show status of movies, games, and equipment.
- Show payment to employees for the week.
- Show payment to repair shops for the month.
- Show income of a given store for the month.
- Reserve titles by members.
- Note return of items by members and additional charges outstanding (e.g., $1.00 per cassette not rewound).
- Show rental of items and initial charge for the first day of the rental.

Start with the E-R model of your system and note the attributes of each entity and relationship. Choose the DBMS and the computer system. Convert the E-R model to that of your DBMS. Implement the applications indicated above and design a set of tests for your system.

Many ambiguities in this case study will have to be resolved. This should be done via observation of an actual video store and discussions with its management. You may make appropriate assumptions but you must be able to defend them.

14.2 Do Drive is a small driving school that is growing and feels the need for a database system. The school offers driving lessons on three different vehicles—cars, trucks, and buses. To get a driving certificate from the school, which is a prerequisite for getting a driver's licenses, each student should score more than 75% in five theoretical courses (Defensive Driving, Automobile Mechanics, Highway Code, Safe Driving, and Maintenance) and more than 85% in practical driving. After 10 hours of practical driving, a student's performance is assessed. If the student fails, he or she will be asked to take two hours of additional driving. If the student fails one of the theoretical courses, he or she will be asked to appear for a supplementary test in that course.

The fee for the driving course is $300.00 for car, $700.00 for bus, and $1,000.00 for trucks. The fee for one supplementary test or one hour of extra driving, 10% of the course fee. Students can pay their fees in installments; however, the certificate is withheld while the student owes money to the school.

The school employs three types of employees: salaried employees for administration, teachers who offer theoretical courses, and instructors who give practical lessons. The salaried employees are paid a monthly salary. Each teacher is paid $300.00 per course section and each instructor is paid $100.00 per student.

You may make the following assumptions:

- A teacher can offer one or more courses. A teacher can also offer more than one section of the same course.
- An instructor can offer practical lessons to several students.
- An instructor can offer practical lesson on more than one type of vehicle.
- The school owns more than one vehicle of each type.

Once a student gets a certificate, the details pertaining to the student can be removed from the online database.

Typical operations to be supported by the database are listed below:

- Add new students, teachers, instructors, salaried employees, and vehicles.
- Remove existing students, teachers, instructors, salaried employees, and vehicles.
- Compute various types of statistics for the student population.
- Compute the payments for employees.
- Prepare the schedules for the courses and driving lessons.
- Keep track of payments made by students and amounts outstanding.

Chapter

15

Distributed Databases

Contents

In this chapter we discuss some of the issues involved in the case of a distributed database. The components of such a database are located at a number of sites interconnected by means of a communications network. Each node consists of an independent computer system and its software. Advantages of distributing the database are the increased availability, reliability and the possibility of incremental growth. However, the costs and complexity of the system are higher. The partitioning of the database can be non-disjoint and some portions of the database could be replicated. Data distribution is covered in Section 15.3.

A query in a distributed database may need data from more than one node which entails communication costs. Distributed query processing is the topic of Section 15.5 wherein we introduce the semijoin operation. This operation is used to reduce the amount of data transmission. Consistency requirements are stressed in Section 15.6. Concurrency control in the case of a distributed database requires special treatment. A number of concurrency control alternatives are presented in Section 15.7. Section 15.8 introduces the failures peculiar to a distributed system and presents schemes for recovery from such failures. Distributed deadlock detection and prevention are covered in Section 15.9. Issues of security are the topic of Section 15.10. Examples of distributed systems are the subject of Section 15.11.

15.1 Introduction

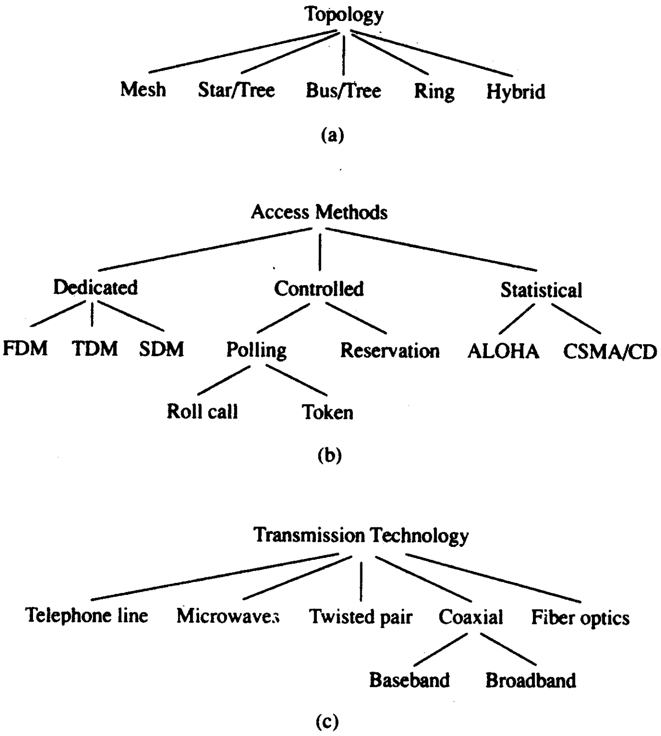
Independent or decentralized systems were the norm in the earlier days of information management, the 1950s and early 1960s. There was duplication of hardware and facilities. Incompatible procedures and lack of management control were the consequences of the evolving nature of the field. The latter may also have been partly due to the lack of understanding of the computer as a tool. In the late 1960s and early 1970s, the trend was toward the use of large general-purpose computers, heralded by the introduction of the IBM System/360. The same facility served a multitude of users with differing needs, leading to conflict and lack of responsiveness. A centralized database system is one such shared facility.

In a centralized database system, the DBMS and the data reside at a single location, and control and processing is limited to this location. However, many organizations have geographically dispersed operations. A case in point is the MUC Library system discussed in Chapters 8 and 9. For such organizations, accessing data from a centralized database creates problems. Data of concern to a particular location, such as the Lynn branch, has to be obtained from the central site. The reliability of the system is compromised since loss of messages between sites or failure of the communication links may occur. The excessive load on the system at the central site would likely cause all accesses to be delayed. Furthermore, the single central site would exhibit a sizable load of transactions, requiring a very large computing system.

An organization located in a single building with quasi-independent operational divisions, each with its own information processing needs and using a centralized database on a local network, would have similar problems.

The current trend is toward a distributed system. This is a central system connected to intelligent remote devices, each of which can itself be a computer or interconnected, possibly heterogeneous, computers. The distribution of processing power creates a feasible environment for data distribution. All distributed data must still be

Figure 15.1 Network design issues.



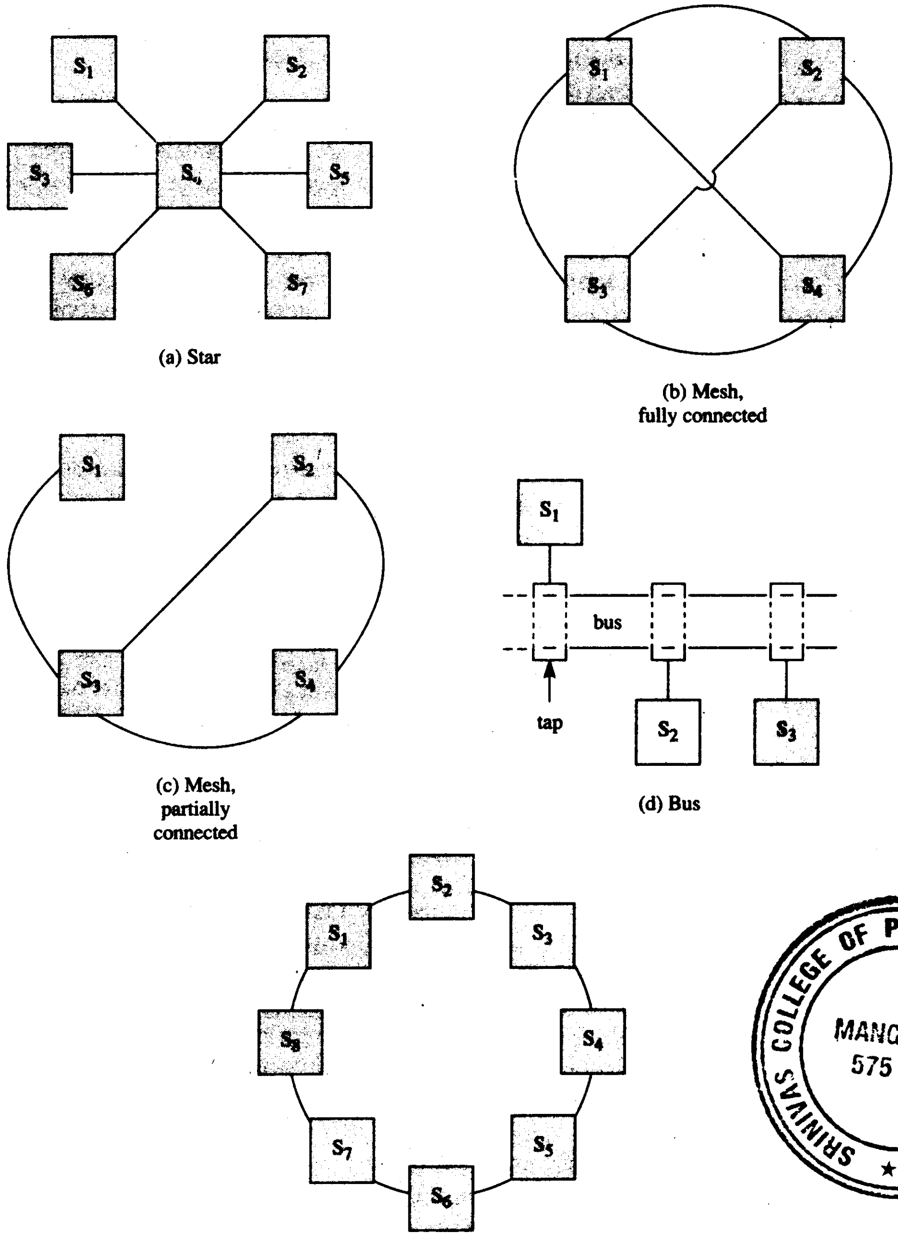
Network topology defines the structure of the network, as illustrated in Figure 15.2. In the **star topology,** all sites or nodes are connected to a central node, which is responsible for transmitting messages between the nodes. A star topology can be considered tree-based, if we consider the central node as the root node. In the **mesh connection,** the interconnecting could be variable or fully interconnected, where each node is connected directly to all other nodes. The nodes are connected by taps to a linear cable in the **bus network.** In **ring topology** each node is connected to the next by a point-to-point cable with the nodes forming a closed circuit.

A fully connected network (or mesh), in which every site is connected to all other sites, is very reliable, although expensive. Even when a link is down, a number of alternate paths exist. In practice, partially connected networks are more likely to occur. Based on traffic load and networking considerations, certain nodes are interconnected and this still allows some alternate paths between sites. In a partially connected network there may be links whose failures could partition the network.

The star network has a central node to which other nodes are connected. Some of these peripheral nodes may act as concentrators for nodes connected to them in a treelike fashion. The reliability of the system is critically dependent on the central node. Star networks or derivatives occur frequently in communication networks.

In ring topology, the nodes are connected to each other and form a closed loop. Data is transmitted in packets that circulate through the ring. The packets are inserted into the ring one at a time.

Figure 15.2 Network topologies.



(a) Star

(b) Mesh, fully connected

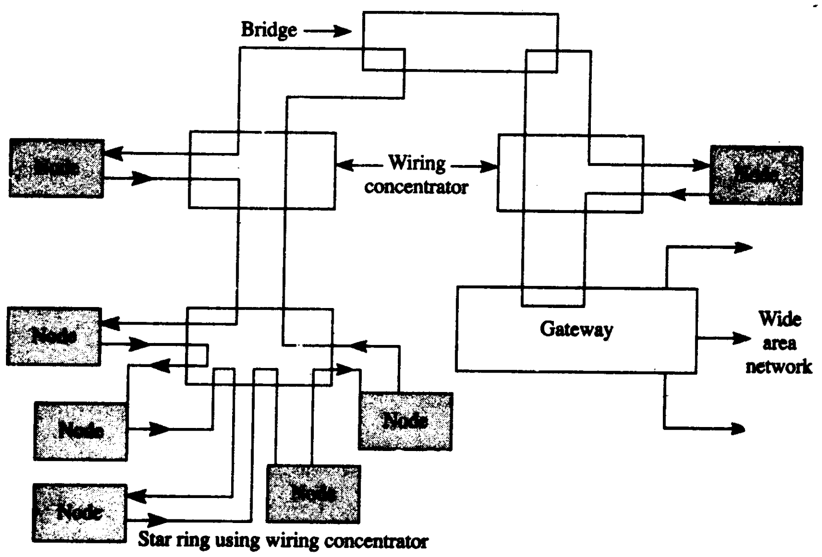
(c) Mesh, partially connected

(d) Bus

(e) Ring

The control and access method defines how the nodes on the system get control of and utilize the transmission media. In the dedicated approach, the communication media is shared in a dedicated fashion, based on ume or frequency. In **synchronous time-division multiplexing (SDM)**, the different sites connected to a shared channel

Figure 15.4 Use of wiring concentrator in ring networks.



15.2.1 Failures and Distributed Databases

Distributed databases are designed to be operational even when certain failures occur in the system. Failure is said to have occurred when a site does not receive messages on a particular link, or when it receives garbled messages. Three kinds of failures can easily be identified: node failure, loss of message, or communication link failure.

A simple decentralized scheme to detect these failures could be based on periodic message exchanges between terminal nodes of the links with each site maintaining a table of "up" and "down" sites. A site that detects the failure of another site or of the link between the sites informs all other sites (including the failed site). This eventually forces a recovery procedure to start at the failed site. A site that is recovering from failure (has been down and is now ready to be up) requires that the system initiate special procedures to allow it to be reintegrated into the system. These procedures allow the site database state to become consistent with the rest of the database.

Communication link or node failures, in certain cases, can result in the database system becoming partitioned, i.e., become two or more independent systems. Example 15.2 illustrates such **network partitioning**.

Example 15.2

Consider the partiaily connected mesh structured communication network of Figure A. A failure in the link between nodes B and D will not cause any disruption in communication, since an alternate path exists. A failure of the link between A and D will divide the network into two partitions. A failure of node D will divide the network in three partitions.