Chapter 5

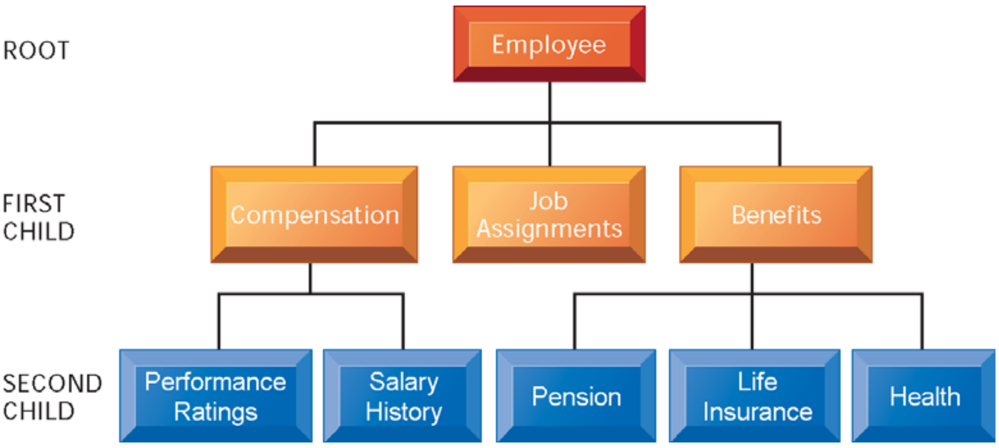
Foundations of Business Intelligence: Databases and Information Management

LEARNING TRACK 3: HIERARCHICAL AND NETWORK DATA MODELS

You can still find older systems that are based on a hierarchical or network data model. The hierarchical DBMS is used to model one-to-many relationships, presenting data to users in a treelike structure. Within each record, data elements are organized into pieces of records called segments. To the user, each record looks like an organizational chart with one top-level segment called the root. An upper segment is connected logically to a lower segment in a parent–child relationship. A parent segment can have more than one child, but a child can have only one parent.

Figure 1 shows a hierarchical structure that might be used for a human resources database. The root segment is Employee, which contains basic employee information such as name, address, and identification number. Immediately below it are three child segments: Compensation (containing salary and promotion data), Job Assignments (containing data about job positions and departments), and Benefits (containing data about beneficiaries and benefit options). The Compensation segment has two children below it: Performance Ratings (containing data about employees' job performance evaluations) and Salary History (containing historical data about employees' past salaries). Below the Benefits segment are child segments for Pension, Life Insurance, and Health, containing data about these benefit plans.

FIGURE 1 A hierarchical database for a human resources system.



The hierarchical database model looks like an organizational chart or a family tree. It has a single root segment (Employee) connected to lower level segments (Compensation, Job Assignments, and Benefits). Each subordinate segment, in turn, may connect to other subordinate segments. Here, Compensation connects to Performance Ratings and Salary History. Benefits connects to Pension, Life Insurance, and Health. Each subordinate segment is the child of the segment directly above it.

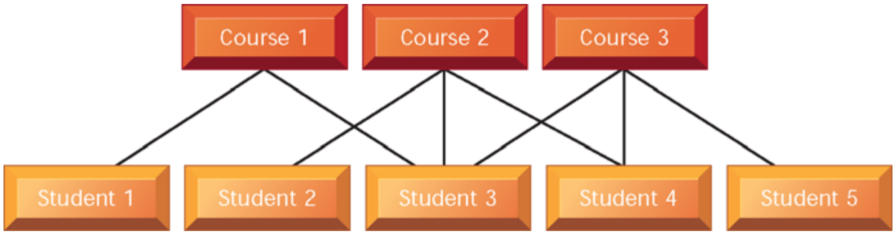
Whereas hierarchical structures depict one-to-many relationships, network DBMS depict data logically as many-to-many relationships. In other words, parents can have multiple children, and a child can have more than one parent. A typical many-to-many relationship for a network DBMS is the student–course relationship (see Figure 2). There are many courses in a university and many students. A student takes many courses, and a course has many students.

Hierarchical and network DBMS are considered outdated and are no longer used for building new database applications. They are much less flexible than relational DBMS and do not support ad hoc, English language–like inquiries for information. All paths for accessing data must be specified in advance and cannot be changed without a major programming effort. For instance, if you queried the human resources database illustrated in Figure 1 to find out the names of the employees with the job title of administrative assistant, you would discover that there is no way the system can find the answer in a reasonable amount of time. This path through the data was not specified in advance.

Relational DBMS, in contrast, have much more flexibility in providing data for ad hoc queries, combining information from different sources, and providing capability to add new data and records without disturbing existing programs and applications. However, these systems can be slowed down if they require many accesses to the data stored on disk to carry out the select, join, and project commands. Selecting one part number from among millions, one record at a time, can take a long time. Of course, the database can be tuned to speed up prespecified queries.

Hierarchical DBMS can still be found in large legacy systems that require intensive high-volume transaction processing. Banks, insurance companies, and other high-volume users continue to use reliable hierarchical databases, such as IBM's Information Management System (IMS) developed in 1969. As relational products acquire more muscle, firms will shift away completely from hierarchical DBMS, but this will happen over a long period of time.

FIGURE 2 The network data model.



This illustration of a network data model showing the relationship the students in a university have to the courses they take represents an example of logical many-to-many relationships.