Database

A **database** is an organized collection of data. The data is typically organized to model relevant aspects of reality (for example, the availability of rooms in hotels), in a way that supports processes requiring this information (for example, finding a hotel with vacancies).

Database management systems (DBMSs) are specially designed applications that interact with the user, other applications, and the database itself to capture and analyze data. A general-purpose **database management system (DBMS)** is a software system designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Microsoft Access, Oracle, SAP, dBASE, FoxPro, and IBM DB2. A database is not generally portable across different DBMS, but different DBMSs can inter-operate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one database.

Terminology and overview

Formally, the term "database" refers to the data itself and supporting data structures.

Databases are created to operate large quantities of information by inputting, storing, retrieving, and managing that information. Databases are set up, so that one set of software programs provides all users with access to all the data. Databases use a table format, that is made up of rows and columns. Each piece of information is entered into a row, which then creates a record. Once the records are created in the database, they can be organized and operated in a variety of ways that are limited mainly by the software being used. Databases are somewhat similar to spreadsheets, but databases are more demanding than spreadsheets because of their ability to manipulate the data that is stored. It is possible to do a number of functions with a database that would be more difficult to do with a spreadsheet. The word data is normally defined as facts from which information can be derived. A database may contain millions of such facts. From these facts the database management system (DBMS) can develop information.

A "database management system" (DBMS) is a suite of computer software providing the interface between users and a database or databases. Because they are so closely related, the term "database" when used casually often refers to both a DBMS and the data it manipulates.

Outside the world of professional information technology, the term *database* is sometimes used casually to refer to any collection of data (perhaps a spreadsheet, maybe even a card

index). This article is concerned only with databases where the size and usage requirements necessitate use of a database management system.

The interactions catered for by most existing DBMS fall into four main groups:

- Data definition. Defining new data structures for a database, removing data structures from the database, modifying the structure of existing data.
- Update. Inserting, modifying, and deleting data.
- Retrieval. Obtaining information either for end-user queries and reports or for processing by applications.
- Administration. Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information if the system fails.

A DBMS is responsible for maintaining the integrity and security of stored data, and for recovering information if the system fails.

Both a database and its DBMS conform to the principles of a particular database model. "Database system" refers collectively to the database model, database management system, and database.

Physically, database servers are dedicated computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most database applications. DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these functions. Since DBMSs comprise a significant economical market, computer and storage vendors often take into account DBMS requirements in their own development plans.

Databases and DBMSs can be categorized according to the database model(s) that they support (such as relational or XML), the type(s) of computer they run on (from a server cluster to a mobile phone), the query language(s) used to access the database (such as SQL or XQuery), and their internal engineering, which affects performance, scalability, resilience, and security.

Applications and roles

Most organizations in developed countries today depend on databases for their business operations. Increasingly, databases are not only used to support the internal operations of the

organization, but also to underpin its online interactions with customers and suppliers (see Enterprise software). Databases are not used only to hold administrative information, but are often embedded within applications to hold more specialized data: for example engineering data or economic models. Examples of database applications include computerized library systems, flight reservation systems, and computerized parts inventory systems.

Client-server or transactional DBMSs are often complex to maintain high performance, availability and security when many users are querying and updating the database at the same time. Personal, desktop-based database systems tend to be less complex. For example, FileMaker and Microsoft Access come with built-in graphical user interfaces.

General-purpose and special-purpose DBMSs

A DBMS has evolved into a complex software system and its development typically requires thousands of person-years of development effort. Some general-purpose DBMSs such as Adabas, Oracle and DB2 have been undergoing upgrades since the 1970s. General-purpose DBMSs aim to meet the needs of as many applications as possible, which add to the complexity. However, the fact that their development cost can be spread over a large number of users means that they are often the most cost-effective approach. However, a general-purpose DBMS is not always the optimal solution: in some cases a general-purpose DBMS may introduce unnecessary overhead. Therefore, there are many examples of systems that use special-purpose databases. A common example is an email system: email systems are designed to optimize the handling of email messages, and do not need significant portions of a general-purpose DBMS functionality.

Many databases have application software that accesses the database on behalf of end-users, without exposing the DBMS interface directly. Application programmers may use a wire protocol directly, or more likely through an application programming interface. Database designers and database administrators interact with the DBMS through dedicated interfaces to build and maintain the applications' databases, and thus need some more knowledge and understanding about how DBMSs operate and the DBMSs' external interfaces and tuning parameters.

General-purpose databases are usually developed by one organization or community of programmers, while a different group builds the applications that use it. In many companies, specialized database administrators maintain databases, run reports, and may work on code that runs on the databases themselves (rather than in the client application).

History

With the progress in technology in the areas of processors, computer memory, computer storage and computer networks, the sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitudes.

The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational. The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system, and the Codasyl model (Network model), implemented in a number of products such asIDMS.

The relational model, first proposed in 1970 by Edgar F. Codd, departed from this tradition by insisting that applications should search for data by content, rather than by following links. The relational model is made up of ledger-style tables, each used for a different type of entity. It was not until the mid-1980s that computing hardware became powerful enough to allow relational systems (DBMSs plus applications) to be widely deployed. By the early 1990s, however, relational systems were dominant for all large-scale data processing applications, and they remain dominant today (2012) except in niche areas. The dominant database language is the standard SQL for the relational model, which has influenced database languages for other data models.

Object databases were invented in the 1980s to overcome the inconvenience of object-relational impedance mismatch, which led to the coining of the term "post-relational" but also development of hybrid object-relational databases.

The next generation of post-relational databases in the 2000s became known as NoSQL databases, introducing fast key-value stores and document-oriented databases. A competing "next generation" known as NewSQL databases attempted new implementations that retained the relational/SQL model while aiming to match the high performance of NoSQL compared to commercially available relational DBMSs.