

STUDY UNIT FOURTEEN

INFORMATION TECHNOLOGY IV

14.1	Database Management Systems	1
14.2	Transaction Processing Modes	2
14.3	Application Processing Phases	4
14.4	Disaster Recovery and Business Continuity	6

This study unit is the fourth of five covering information technology (IT).

14.1 DATABASE MANAGEMENT SYSTEMS

1. A **database management system (DBMS)** is an integrated set of software tools superimposed on the data files that helps maintain the integrity of the underlying database.
 - a. Database management systems make the maintenance of vast relational databases practical. Without the sophisticated capabilities of database management systems, enforcing the rules that make the database relational would be overwhelmingly time-consuming.
 - b. A DBMS allows programmers and designers to work independently of the physical and logical structure of the database.
 - 1) Before the development of DBMSs, programmers and systems designers needed to consider the logical and physical structure of the database with the creation of every new application. This was extremely time-consuming and therefore expensive.
 - 2) With a DBMS, the physical structure of the database can be completely altered without having to change any of the programs using the data items. Thus, different users may define their own views of the data (called subschemas).
2. A particular database's design, called its **schema**, consists of the layouts of the tables and the constraints on entering new records. To a great extent, a DBMS automates the process of enforcing the schema.
 - a. Two vital parts of any DBMS are as follows:
 - 1) A data definition language, which allows the user to specify how the tables will look and what kinds of data elements they will hold.
 - 2) A data manipulation language, with which the DBMS retrieves, adds, deletes, or modifies records and data elements.
 - 3) Both of these roles are commonly fulfilled in the current generation of database management systems by **Structured Query Language (SQL)** or one of its many variants.
 - b. The **data dictionary** contains the physical and logical characteristics of every data element in a database.
 - 1) The data dictionary includes, for example, the name of the data element (e.g., employee name, part number), the amount of space required to store the data element (in bytes), and what kind of data is allowed in the data element (e.g., alphabetic, numeric).
 - 2) The data dictionary also describes the mapping of every data element to all applications where it is updated and vice versa.
 - 3) Thus, the data dictionary contains the size, format, usage, meaning, and ownership of every data element as well as what persons, programs, reports, and functions use the data element.

3. A DBMS can maintain a **distributed database**, meaning one that is stored in two or more physical sites.
 - a. In the replication, or snapshot, technique, the DBMS duplicates the entire database and sends it to multiple locations. Changes are periodically copied and similarly distributed.
 - b. In the fragmentation, or partitioning, method, specific records are stored where they are most needed. For example, a financial institution may store a particular customer's data at the branch where (s)he usually transacts his/her business. If the customer executes a transaction at another branch, the pertinent data are retrieved via communications lines.
4. A **deadly embrace**, also called a **deadlock**, can be resolved by a DBMS. This situation occurs when two transactions attempt to update a single data element simultaneously.
 - a. When a deadly embrace occurs, the DBMS selects one of the transactions as the "victim" and releases the data resources it controls so that the other transaction can run to completion. The victim transaction is then restarted and permitted to run.
5. Those in the IT function responsible for dealing with the DBMS are called database administrators (see item 1.a. in Subunit 11.3).

14.2 TRANSACTION PROCESSING MODES

1. **Batch processing.** In this mode, transactions are accumulated and submitted to the computer as a single "batch." In the early days of computers, this was the only way a job could be processed.
 - a. In batch processing, the user cannot influence the process once the job has begun (except to ask that it be aborted completely). (S)he must wait until the job is finished running to see if any transactions in the batch were rejected and failed to post.
 - b. Despite huge advances in computer technology, this accumulation of transactions for processing on a delayed basis is still widely used. It is very efficient for such applications as payroll, where large numbers of routine transactions must be processed on a regular schedule.
2. **Online processing.** In this mode, the computer processes each transaction individually as the user enters it.
 - a. The user is in direct communication with the computer and gets immediate feedback on whether the transaction was accepted or not.
 - b. A common example is an accounts payable system in which a payables clerk can enter each individual invoice as (s)he verifies the paperwork.
3. Many applications use **combined batch and online** modes.
 - a. In such systems, users continuously enter transactions in online mode throughout the workday, collecting them in batches. The computer can then take advantage of the efficiencies of batch mode overnight when there are fewer users logged on to the system.
4. **Real-time processing.** In some systems, having the latest information available at all times is crucial to the proper functioning of the system.
 - a. A thermostat is a common example, constantly monitoring the temperature in the room and engaging the heating or cooling accordingly.
 - b. Online, real-time systems combine the two modes of user data entry and instant update. A common example is an airline reservation system, which is constantly updated from moment to moment and must be available all the time.

5. **Centralization.** During the early days of computer processing, computers were very large and expensive and only organizations such as large banks and governmental agencies could afford them.
 - a. Of necessity, all processing and systems development were done at a single, central location. Users connected to the mainframe via “dumb terminals,” i.e., simple monitor-and-keyboard combinations with no processing power of their own.
 - b. Since hardware, information security, and data integrity functions were located in one office, economies of scale were achieved and controls were strong.
6. **Decentralization.** As the data processing industry evolved, computers became smaller (so-called minicomputers), and branch offices of large organizations could have their own.
 - a. Each branch could store and process its data onsite, transmitting the results overnight to the mainframe at the home office. This was an early form of **distributed processing**, in which parts of an organization’s computer operations could be performed in separate physical locations.
 - b. In this early distributed arrangement, the home office mainframe ran its programs and the branches ran theirs. The next evolution was for a single application to be split into pieces so the parts could run on separate hardware platforms.
 - 1) The decision was thus no longer whether an application should run centrally or locally, but rather, which parts of the application are better performed by small local computers and which parts are better performed at some other, possibly centralized, site.
 - c. Since IT functions were no longer concentrated in a single location, issues of security and training became more challenging.
7. **Client/server networks.** The key to the client/server model of distributed processing is that it runs processes on the platform most appropriate to that process while attempting to minimize traffic over the network.
 - a. A “server” is centrally located and devoted to the functions that are needed by all network users.
 - 1) Examples include mail servers (to handle electronic mail), application servers (to run application programs), file servers (to store databases and make user inquiries more efficient), Internet servers (to manage access to the Internet), and web servers (to host websites).
 - 2) Whether a device is classified as a server is not determined by its hardware configuration, but rather by the function it performs. A simple desktop computer can be a server.
 - b. Technically, a “client” is any object that uses the resources of another object. Thus, a client can be either a piece of hardware or a software program.
 - 1) In common usage, however, client generally refers to a device that requests services from a server. This use of the term encompasses anything from a Palm Pilot, to a desktop computer, to another server.
8. **Outsourcing.** Some organizations farm out all or part of their IT function to an outside provider. There are two common motivations for this practice:
 - a. The outside provider offers economies of scale that are not available to the organization.
 - 1) For instance, the organization needs its payroll processed every two weeks and does not wish to invest in the dedicated hardware that would allow it to do that processing itself. In a case like this, the organization may keep its own IT department and simply contract with the service bureau to perform certain specified functions.

- b. Management has decided that IT is not a core competency of the organization and that the entire IT function can be most efficiently provided by a firm specializing in providing IT services.
 - 1) This arrangement is fulfilled by a facilities management organization, which provides the personnel to manage and operate the client's internal IT activity.

14.3 APPLICATION PROCESSING PHASES

1. **Data capture.** In order to be processed, data must be entered into the system. This can be done in batch mode, by online entry (see Subunit 14.2), or even from a personal digital assistant.
2. **Edit routines.** These are controls programmed into the software that prevent certain types of errors from ever getting into the system. They include:
 - a. **Preformatting.** To avoid data entry errors in online systems, a preformatted screen may be designed to look exactly like the corresponding paper document.
 - b. **Field checks.** Some data elements can only contain certain characters, and any transaction that attempts to use an invalid character is halted. A typical example is a Social Security Number, which is not allowed to contain letters.
 - c. **Limit and range checks.** Based on known limits for given information, certain entries can be rejected by the system. For example, hours worked per week cannot exceed 80 without a special override by management; date of birth cannot be any date within the last 15 years, etc.
 - d. **Validity checks.** In order for a transaction to be processed, some other record must already exist in another file. For example, for the system to accept a transaction requesting payment of a vendor invoice, the vendor must already have a record on the vendor master file.
 - e. **Sequence checks.** Processing efficiency is greatly increased when files are sorted on some designated field(s), called the "key," before operations such as matching. For instance, the accounts payable transaction file and master file should both be sorted according to vendor number before the matching operation is attempted. If the system discovers a record out of order, it may indicate that the files were not properly prepared for processing.
 - f. **Self-checking digits.** An algorithm is applied to, for instance, a product number and incorporated into the number.

EXAMPLE:

- 1) A box of detergent has the product number 4187604. The last digit is actually a derived number, arrived at by applying the check-digit algorithm to the other digits.
- 2) The check digit is calculated by starting with the last position of the base product number and multiplying each successive digit to the left by 2, then by 1, then by 2, etc., and adding the results: $(0 \times 2) + (6 \times 1) + (7 \times 2) + (8 \times 1) + (1 \times 2) + (4 \times 1) = 0 + 6 + 14 + 8 + 2 + 4 = 34$. The last digit of this result becomes the check digit.
- 3) When the clerk enters 4187604 into the terminal, the system performs an immediate calculation and determines that this is a valid product number. This reduces keying errors such as dropped and transposed digits.
- g. **Zero-balance checks.** The system will reject any transaction or batch thereof in which the sum of all debits and credits does not equal zero.

3. **Output controls.** These procedures are performed at the end of processing to ensure that all transactions the user expected to be processed were. They include:
 - a. Error listings. All transactions rejected by the system are printed and distributed to the appropriate user department for resolution.
 - b. Record counts. The total number of records processed by the system is compared to the number the user expected to be processed.
 - c. Run-to-run control totals. The new financial balance should be the sum of the old balance plus the activity that was just processed.
 - d. Hash totals. These are totals without a defined meaning, such as the total of employee numbers or invoice numbers.
 - e. Proof account activity listing. This report shows all changes to master files. It can be sent to the appropriate user department to verify that the changes were authorized.
 - f. An audit trail of all processing activity should be generated. It summarizes any or all of the totals described above.
4. **Master file maintenance.** For a description of the two subtypes of master file, see Subunit 13.3.
 - a. The first subtype is only updated irregularly, for instance, when a new vendor is added or an existing one changes its mailing address.
 - b. The second subtype is updated regularly, for instance, with the daily postings of journal activity.
 - c. Whichever of the two subtypes is involved, the power to approve changes to a master file must be assigned in accord with a coherent organizational policy; e.g., the head of payroll cannot approve changes to the customer address file.
5. **Reporting, Accounting, Control, and Management**
 - a. Reports are subsets of the organization's total set of data that are presented in such a way as to (1) reveal the organization's performance or (2) help in decision making. Reports do not necessarily have to be in paper form; they can be viewed entirely onscreen. Examples are the company's statement of income and aging of accounts receivable.
 - b. No matter how powerful an organization's computing resources are, they are finite. If cost-effective, usage of the computer's resources must be measured and the allocated cost billed to the user departments who benefit from receiving IT services.
 - 1) Common bases for cost allocation and billing are number of CPU cycles and number of input/output operations. Special pieces of systems software track what quantity of these measures is consumed by which users.
6. **Query, Audit Trail, and Ad Hoc Reports**
 - a. Online information systems are often powerful enough to allow end users to perform their own queries, i.e., to ask questions directly of the database without the assistance of IT personnel. Such querying is enabled by fourth-generation programming languages (see item 3.d. in Subunit 12.2), which are user-friendly enough to require little technical knowledge.
 - 1) Closely related to the concept of the query is the ad hoc report, which is a "quick-and-dirty" report drawn from one of the organization's databases that fulfills a user need but for which there is not sufficient time or resources to request formally from the IT function.
 - b. An audit trail of activities is a crucial part of monitoring security over a system. The audit trail includes not only the reports described in item 3. above, but also such reports as logs of system sign-in and sign-out times to monitor who was doing what on the system.

14.4 DISASTER RECOVERY AND BUSINESS CONTINUITY

1. The information security goal of data availability (see item 1. in Subunit 13.2) is primarily the responsibility of the IT function.
 - a. **Contingency planning** is the name commonly given to this activity.
 - 1) **Disaster recovery** is the process of resuming normal information processing operations after the occurrence of a major interruption.
 - 2) **Business continuity** is the continuation of business by other means during the period in which computer processing is unavailable or less than normal.
 - b. Two major types of contingencies must be planned for: those in which the data center is physically available and those in which it is not.
 - 1) Examples of the first type of contingency are power failure, random intrusions such as viruses, and deliberate intrusions such as hacking incidents. The organization's physical facilities are sound, but immediate action is required to keep normal processing going.
 - 2) The second type of contingency is much more serious. This type is caused by disasters such as floods, fires, hurricanes, earthquakes, etc. An occurrence of this type necessitates the existence of an alternate processing facility [see item 4.c.1) on the next page].
2. **Periodic backup and offsite rotation** of computer files is the most basic part of any disaster recovery/business continuity plan.
 - a. It is a truth seldom grasped by those who are not computer professionals that an organization's data is more valuable than its hardware. Hardware can be replaced for a price, but each organization's data bundle is unique and is indispensable to carrying on business. If it is ever destroyed, it cannot be replaced. For this reason, periodic backup and rotation are essential.
 - b. A typical backup routine involves duplicating all data files and application programs once a month. Incremental changes are then backed up and taken to the offsite location once a week. (Application files must be backed up in addition to data since programs change too.)
 - c. The offsite location must be temperature- and humidity-controlled and guarded against physical intrusion. Just as important, it must be geographically remote enough from the site of the organization's main operations that it would not be affected by the same natural disaster. It does the organization no good to have adequate backup files if the files are not accessible or have been destroyed.
 - d. In case of an interruption of normal processing, the organization's systems can be restored such that at most seven days of business information is lost. This is not an ideal situation, but it is a far cry from a complete loss of a company's files, which could essentially put it out of business.
3. The **risk assessment** that forms the core of contingency planning involves:
 - a. Identifying and prioritizing the organization's critical applications
 - 1) Not all of an organization's systems are equally important. The firm must decide which vital applications it simply cannot do business without and in what order they should be brought back into operation.
 - b. Determining the minimum recovery time frames and minimum hardware requirements
 - 1) How long will it take to reinstall each critical application and what platform is required? If the interruption has been caused by an attack such as a virus or hacker, how long will it take to isolate the problem and eliminate it from the system?

- c. Developing a recovery plan
 - 1) Each type of contingency requires its own specific recovery procedures (see item 4. below).
- 4. Dealing with Specific Types of Contingencies
 - a. **Power failures** can be guarded against by the purchase of backup electrical generators. These can be programmed to automatically begin running as soon as a dip in the level of electric current is detected. This is a widespread practice in settings such as hospitals where 24-hour system availability is crucial.
 - b. Attacks such as **viruses** and denial-of-service call for a completely different response. The system must be brought down “gracefully” to halt the spread of the infection. The IT staff must be well trained in the nature of the latest virus threats to know how to isolate the damage and bring the system back to full operation.
 - c. The most extreme contingency is when the organization’s main facility is rendered uninhabitable by **flood, fire, earthquake, etc.** It is to prepare for these cases that organizations contract for alternate processing facilities.
 - 1) An **alternate processing facility** is a physical location maintained by an outside contractor for the express purpose of providing processing facilities for customers in case of disaster.
 - a) The recovery center, like the offsite storage location for backup files, must be far enough away that it will likely be unaffected by the same natural disaster that forced the abandonment of the main facility. Usually, companies contract for backup facilities in another city.
 - b) Once the determination is made that processing is no longer possible at the principal site, the backup files are retrieved from the secure storage location and taken to the recovery center.
 - c) Recovery centers can take many forms. Organizations determine which facility is best by calculating the tradeoff between the cost of the contract and the cost of downtime.
 - i) A hot site is a fully operational processing facility that is immediately available. A flying-start site is a hot site with the latest data and software that permit startup within a few minutes or even a few seconds.
 - ii) A warm site is a facility with limited hardware, such as communications and networking equipment, already installed but lacking the necessary servers and client terminals.
 - iii) A cold site is a shell facility lacking most infrastructure but readily available for the quick installation of hardware.
- 5. **Other technologies** that can assist in recovery from an interruption in processing:
 - a. Fault-tolerant computer systems have additional hardware and software as well as a backup power supply. A fault-tolerant computer has additional chips and disk storage. This technology is used for mission-critical applications that cannot afford to suffer downtime.
 - 1) The enabling technology for fault-tolerance is the redundant array of inexpensive discs, or RAID. It is a grouping of multiple hard drives with special software that allows for data delivery along multiple paths. If one drive fails, the other discs can compensate for the loss.
 - b. High-availability computing is used for less-critical applications because it provides for a short recovery time rather than the elimination of recovery time.