Database Management

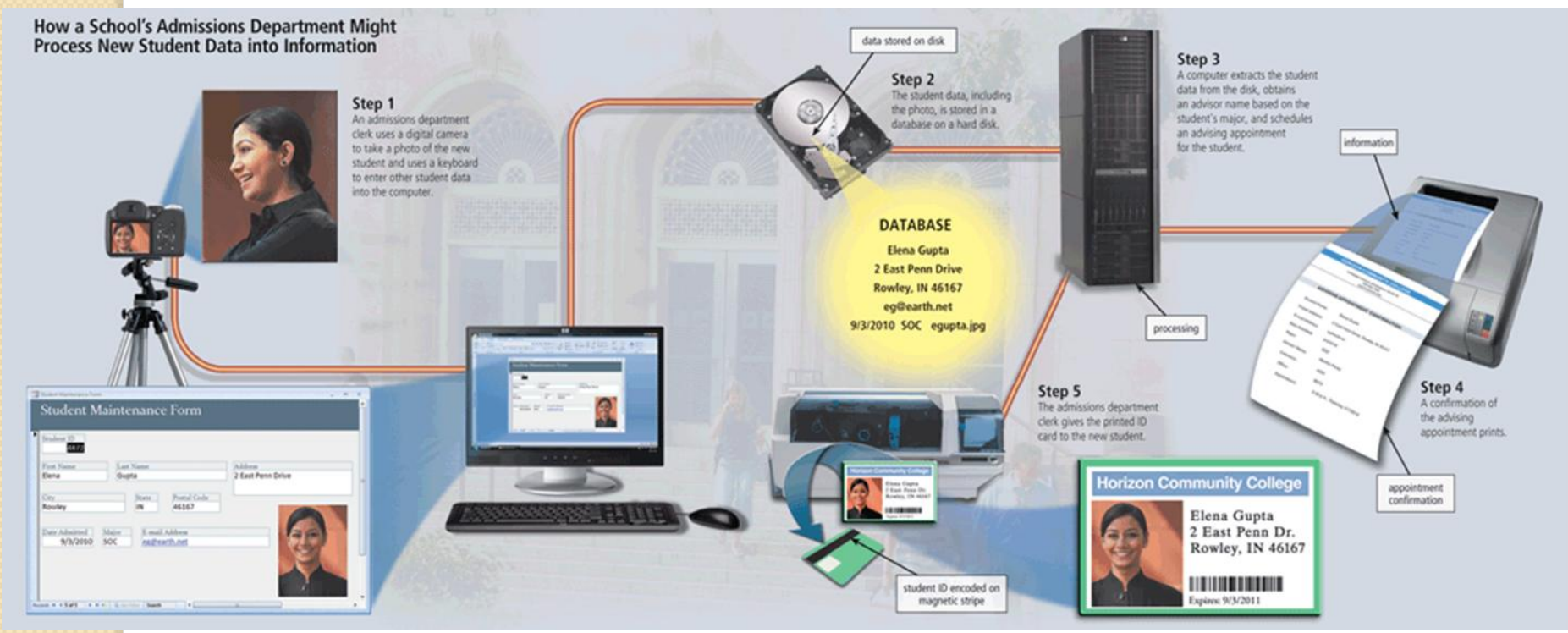
Databases, Data, and Information

- A **database** is a collection of data organized in a manner that allows access, retrieval, and use of that data.
- *Data* is a collection of unprocessed items, which can include text, numbers, images, audio, and video.
- *Information* is processed data that is organized, meaningful, and useful.

Databases, Data, and Information

- Computers process data in a database into information.
- The entered data is stored in a database on a server's hard disk.

Databases, Data, and Information



Databases, Data, and Information

- With **database software**, often called a **database management system (DBMS)**, users create a computerized database; add, modify, and delete data in the database; sort and retrieve data from the database; and create forms and reports from the data in the database.

Databases, Data, and Information

- Data is a valuable asset because it is used to generate information.
- Many business transactions take less time when employees have instant access to information.
- This makes employees more productive and customers more satisfied.

Databases, Data, and Information

- It is important for an organization to manage and protect the data, just like any other resource.
- Therefore, it is important to keep the data secure, such as encrypting and encoding sensitive items.

Data Integrity

- *Data integrity* identifies the quality of the data.
- When a database contains incorrect data, it loses its integrity.
- *Garbage in, garbage out (GIGO)* is a computing phrase that points out the accuracy of a computer's output depends on the accuracy of the input.
- If you input incorrect data (garbage in), the computer will produce incorrect information (garbage out).

Data Integrity

- Data integrity is important because computers and people use information to make decisions.
- If the data is incorrect, this could result in issues such as improper billing.

Qualities of Valuable Information

- To assist with sound decision making, information must have a value.
- For it to be valuable, it should be accurate, verifiable, timely, organized, accessible, useful, and cost-effective.

Qualities of Valuable Information

- *Accurate information* is error free.
  - If it is inaccurate, incorrect decisions could be made.
  - Ex. If your credit report incorrectly shows past due payments, a bank may not lend you money.
- *Verifiable information* can be proven as correct or incorrect.
  - Ex. Personnel at an airport request some type of photo identification.

Qualities of Valuable Information

- *Timely information* has an age suited to its use.
  - Most information loses or gains value over time, such as a transcript (gains).
- *Organized information* is arranged to suit the needs and requirements of the decision maker.
  - Different people may need the same information formatted differently.
  - Ex. List of out of stock items verses alphabetized by vendor.

Qualities of Valuable Information

- *Accessible information* is available when the decision maker needs it.
  - Having to wait may delay an important decision.
- *Useful information* has meaning to the person who receives it.
  - Ex. An announcement of an alumni association meeting is not useful to students who have not graduated.

Qualities of Valuable Information

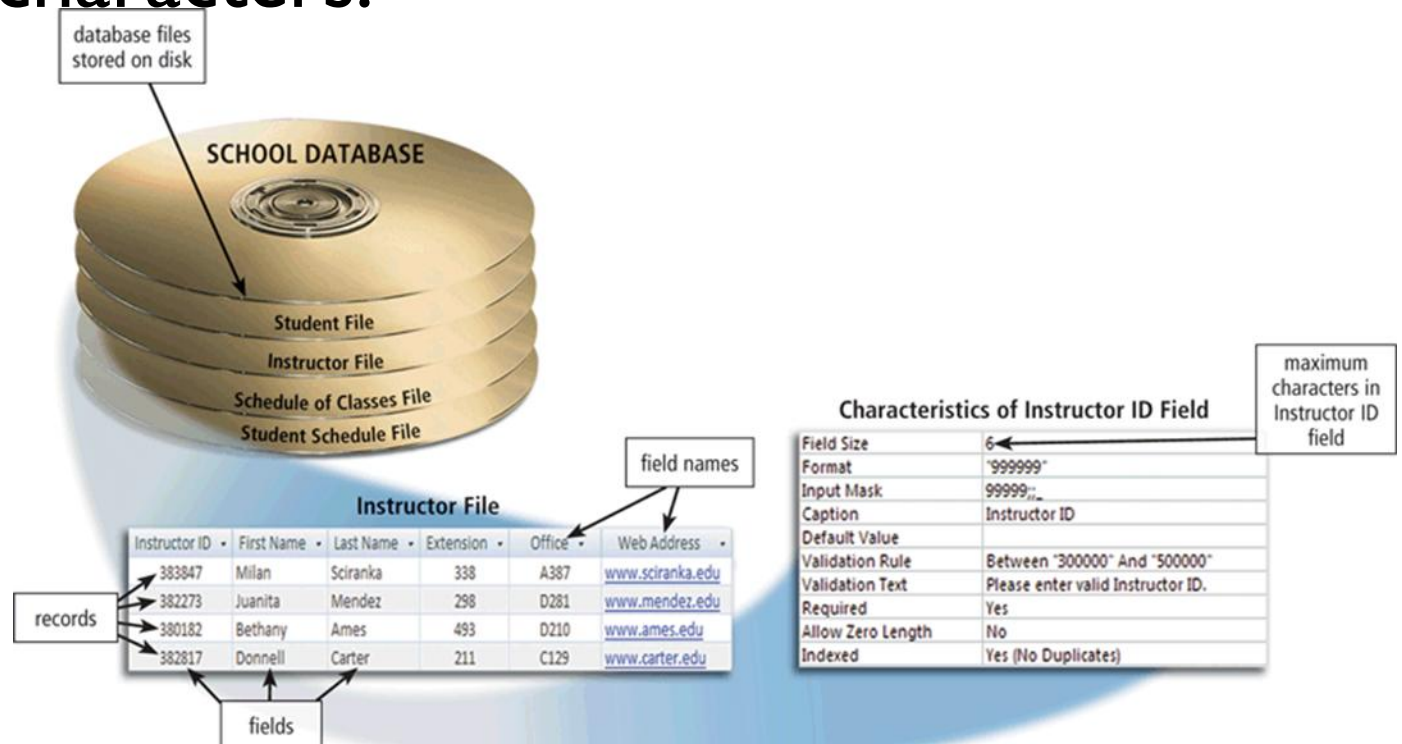
- *Cost-effective information* should give more value than it costs to produce.
  - An organization should occasionally review the info to still see if it is cost-effective.
  - Ex. Sending a printed benefits manual to each employee could be costly, so employees can access an online benefits manual.

The Hierarchy of Data

- Data is organized in layers and is classified in a hierarchy.
- Each higher level of data consists of one or more items from the lower level.
  - Ex. Student has an address, and an address consists of letters and numbers.

The Hierarchy of Data

- A database contains files (or tables), a file contains records, a record contains fields, and a field is made up of one or more characters.



Characters

- As previously mentioned, a bit is the smallest unit of data.
- Eight bits grouped together in a unit comprise a byte.
- In ASCII coding, each byte represents a single **character**, which can be a number, letter, space, punctuation mark, or other symbol.
- Unicode coding can use one or two bytes to represent a character.

Fields

- A **field** is a combination of one or more related characters or bytes and is the smallest unit of data a user accesses.
- A **field name** uniquely identifies each field.
- When searching for data, you often specify the field name.
- Ex. First names, last name, id, etc.

Fields

- The **field size** defines the maximum number of characters a field can contain.
- The **data type** specifies the kind of data a field can contain and how the field is used.
  - Text: letters and numbers, or special characters
  - Numeric: numbers only
  - AutoNumber: unique number automatically assigned to each record

Fields

- Currency: dollar and cent amounts, numbers with decimals
- Date: month, day year, and sometimes time
- Memo: lengthy text entry
- Yes/No (or *Boolean*): only yes or no (true or false)
- Hyperlink: email address or web address
- Object: photo, audio, video, document
- Attachment: document or image attached to the field

Fields

Instructor file

Instructor ID	Text
First Name	Text
Last Name	Text
Extension	Number
Office	Text
Web Address	Hyperlink

data types

Student file

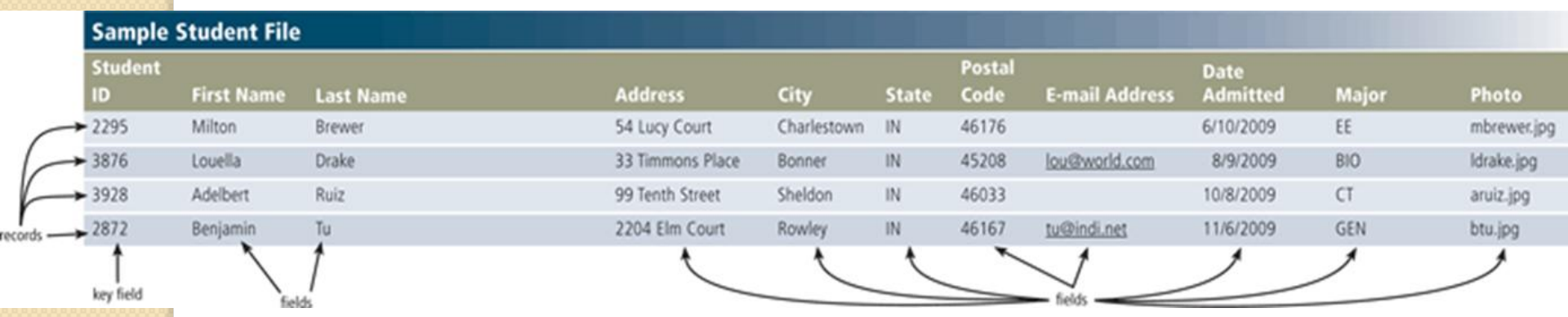
Student ID	AutoNumber
First Name	Text
Last Name	Text
Address	Text
City	Text
State	Text
Postal Code	Text
E-mail Address	Hyperlink
Date Admitted	Date/Time
Major	Text
Photo	Attachment

Records

- A **record** is a group of related fields.
  - Ex. A student record contains a set of fields about one student.
- A **primary key** is a field that uniquely identifies each record in a file.
  - Ex. Student ID, no two students have the same ID.
- In some tables, the primary key consists of multiple fields, called a *composite key*.

Files

- A **data file** is a collection of related records stored on a storage medium such as a hard disk or optical disc.
- A file may consist of thousands of individual records, each containing the same fields with different data.

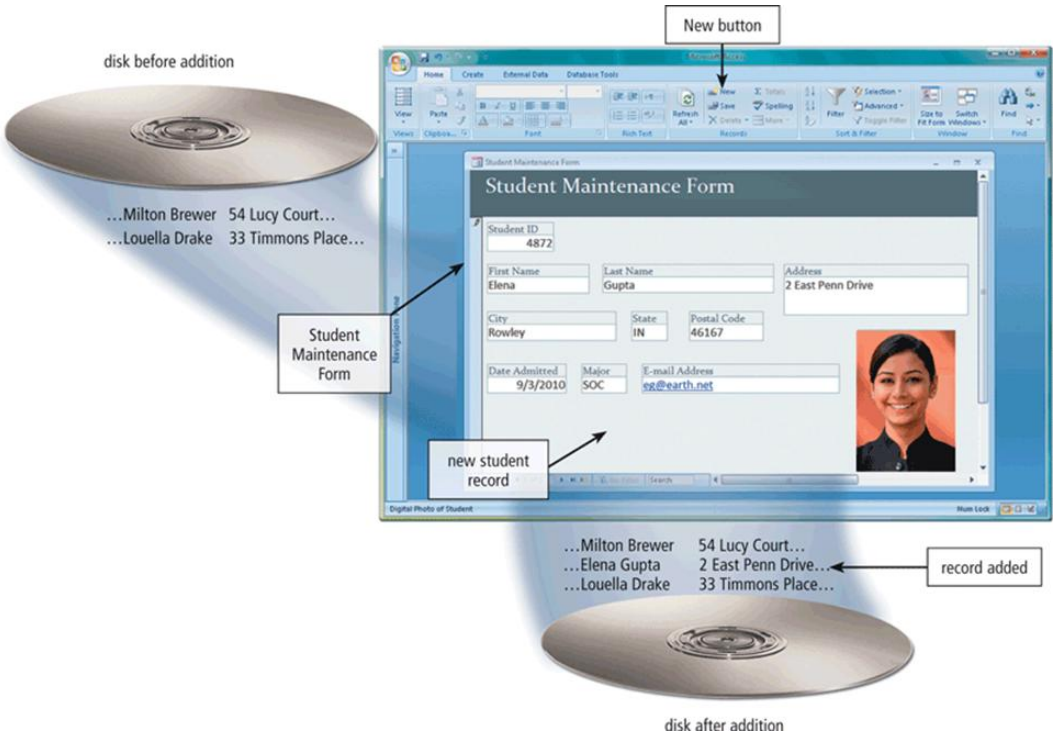


Maintaining Data

- **File maintenance** refers to the procedures that keep data current.
- Procedures include adding records to, modifying records in, and deleting records from a file.

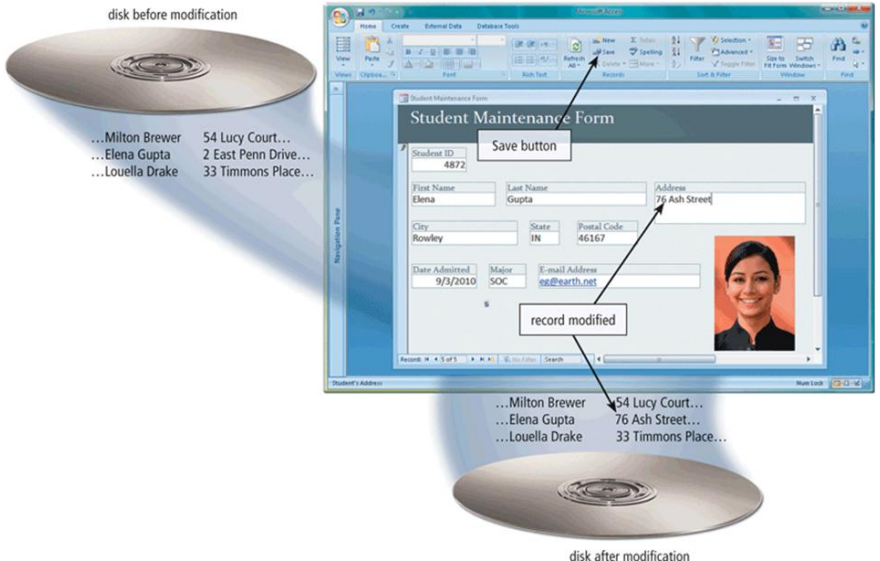
Adding Records

- Users add new records to a file when they obtain new data.
  - Ex. An admissions department clerk adds a new record to the Student file.



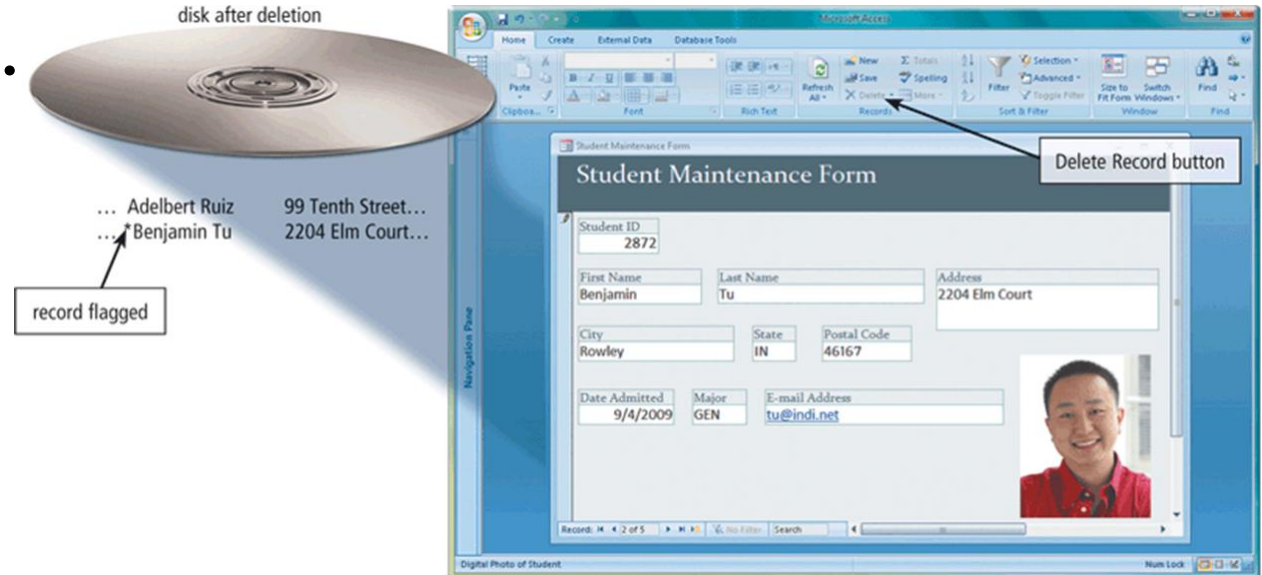
Modifying Records

- Generally, users modify a record in a file
  - To correct inaccurate data or
    - Ex. A student's email address was incorrectly entered initially.
  - To update old data with new data
    - Ex. A student moved and needs their address updated.



Deleting Records

- When a record is no longer needed, a user deletes it from file.
- In some cases, the record is removed completely and cannot be undone.
- Other DBMSs will just flag the entry as inactive.



Validating Data

- **Validation** is the process of comparing data with a set of rules or values to find out if the data is correct.
- Many programs perform a *validity check* that analyzes data, either as you enter it or after you enter it, to help ensure that it is correct.

Validating Data

- If data fails the validity check, the computer should not allow the data to be entered.
- Validity checks, sometimes called *validation rules*, reduce data entry errors and thus enhance the data's integrity.

Alphabetic/Numeric Check

- An *alphabetic check* ensures that users enter only alphabetic data into a field.
  - Ex. A first name should only contain letters.
- A *numeric check* ensures that users enter only numeric data into a field.
  - Ex. A postal code should only contain numbers.

Range Check

- A *range check* determines whether a number is within a specified range.
  - Ex. A lowest per credit hour fee is $75.00 and the highest is $370.75. A range check ensures the value is between $75.00 and $370.75.

Consistency Check

- A *consistency check* tests the data in two or more associated fields to ensure that the relationship is logical and their data is in the correct format.
  - Ex. A Date Admitted field cannot occur earlier than a Birth Date field.

Completeness Check

- A *completeness check* verifies that a required field contains data.
  - Ex. Some fields cannot be left blank, while others require a minimum number of characters.

Check Digit

- A *check digit* is a number(s) or character(s) that is appended to or inserted in a primary key value.
- It confirms the accuracy of a primary key value.
- Sensitive information like bank accounts or credit cards often include one or more check digits.

File Processing Verses Databases

- Almost all application programs use the file processing approach, the database approach, or a combination of both to store and manage data.

File Processing Systems

- In a typical **file processing system**, each department or area within an organization has its own set of files.
- The records in one file may not relate to the records in another.
- A lot of these systems have two major weaknesses: redundant and isolated data.

File Processing Systems

- Data Redundancy: Each department or area in an organization has its own files in a file processing system, thus, the same fields are stored in multiple files.
- Duplicating data in this manner wastes resources such as storage space and time.
- It also can increase the chance of errors.
  - Ex. If a student updates their address, it would need to be changed in all files.

File Processing Systems

- Isolated Data: Often it is difficult to access data stored in separate files in different departments.

The Database Approach

- When an organization uses a **database approach**, many programs and users share the data in the database.
- The database secures its data so that only authorized users can access certain data items.
- Instead of working directly with the DBMS, some users interact with a *front end*, which is a program that generally has a more user-friendly interface.
- The *back end* is an application that supports a front-end program.

The Database Approach

- The database approach addresses many of the weaknesses of the file processing systems.
  - Reduced Data Redundancy: Most data items are stored only in one file.
  - Improved Data Integrity: When users modify data in the database, they make changes to one file instead of multiple.
  - Shared Data: The data in a database environment belongs to and is shared by the entire organization.

The Database Approach

- Easier Access: The database approach allows nontechnical users to access and maintain data, providing they have the necessary privileges.
- Reduced Development Time: It often is easier and faster to develop programs that use the database approach.

The Database Approach

- Databases also have some disadvantages.
  - They can be more complex than file processing systems.
  - They require more memory, storage, and processing power.
  - Data can be more vulnerable.
- Despite these limitations, many businesses work with databases because of their tremendous advantages.
- Long term benefits exceed the initial cost.