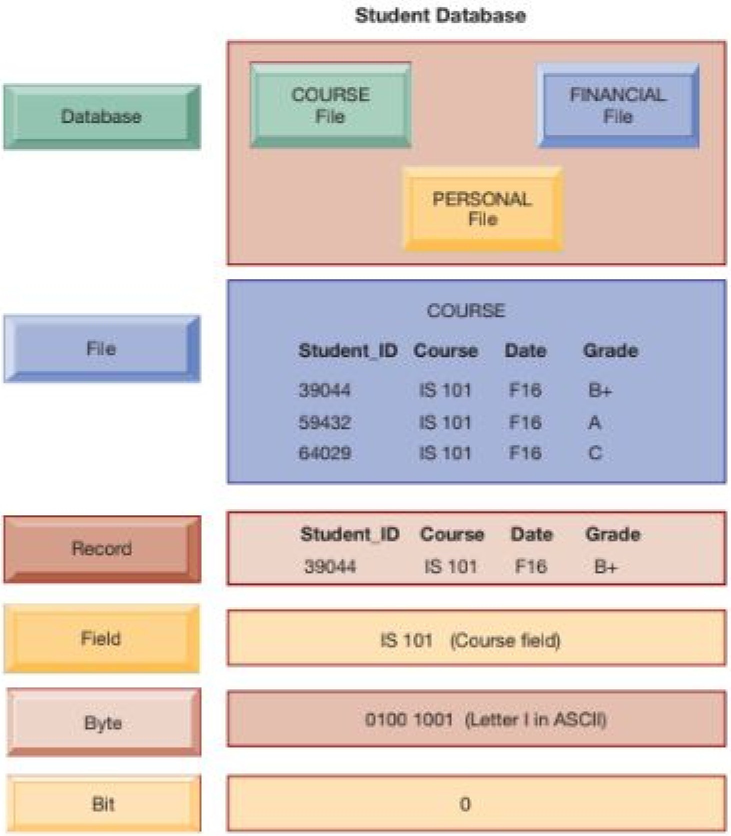
Chapter- 6

Foundation of Business Intelligence: Databases & IS

🔖 Topic- 10. 1: What are the problems of managing data resources in a traditional file environment?

10.1.1: File Organization Terms and Concepts

A computer system organizes data in a hierarchy that starts with bits and bytes and progresses to fields, records, files, and databases (see Figure 6. 1). A bit represents the smallest unit of data a computer can handle. A group of bits, called a byte, represents a single character, which can be a letter, a number, or another symbol. A grouping of characters into a word, a group of words, or a complete number (such as a person's name or age) is called a field. A group of related fields, such as the student's name, the course taken, the date, and the grade, comprises a record; a group of records of the same type is called a file .



Student_ID	Course	Date	Grade
39044	IS 101	F16	B+
59432	IS 101	F16	A
64029	IS 101	F16	C

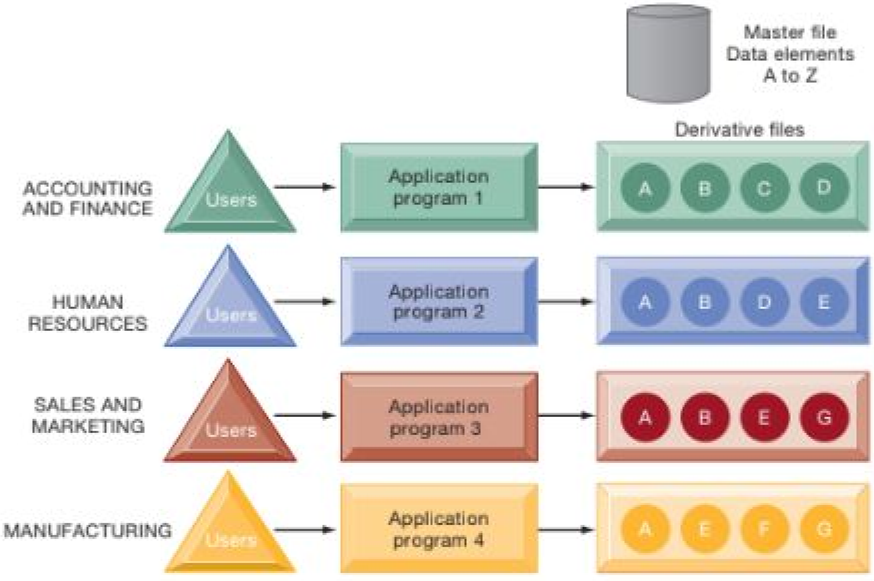
Student_ID	Course	Date	Grade
39044	IS 101	F16	B+

A computer system organizes data in a hierarchy that starts with the bit, which represents either a 0 or a 1. Bits can be grouped to form a byte to represent one character, number, or symbol. Bytes can be grouped to form a field, and related fields can be grouped to form a record. Related records can be collected to form a file, and related files can be organized into a database.

For example, the records in Figure 6. 1could constitute a student course file. A group of related files makes up a database. The student course file illustrated in Figure 6. 1could be grouped with files on students' personal histories and financial backgrounds to create a student database. A record describes an entity. An entity is a person, place, thing, or event on which we store and maintain information. Each characteristic or quality describing a particular entity is called an attribute. For example, Student_ID, Course, Date, and Grade are attributes of the entity COURSE. The specific values that these attributes can have are found in the fields of the record describing the entity COURSE.used to provide computing services that connect employees, customers, and suppliers into a coherent digital environment, including large mainframes, midrange computers, desktop and laptop computers, and mobile handheld and remote cloud computing services.

10.1.2: Problems with the Traditional File Environment

In most organizations, systems tended to grow independently without a companywide plan. Accounting, finance, manufacturing, human resources, and sales and marketing all developed their own systems and data files. Figure 6. 2 illustrates the traditional approach to information processing. Each application, of course, required its own files and its own computer program to operate. For example, the human resources functional area might have a personnel master file, a payroll file, a medical insurance file, a pension file, a mailing list file, and so forth, until tens, perhaps hundreds, of files and programs existed. In the company as a whole, this process led to multiple master files created, maintained, and operated by separate divisions or departments. As this process goes on for 5 or 10 years, the organization is saddled with hundreds of programs and applications that are very difficult to maintain and manage. The resulting problems are data redundancy and inconsistency, program-data dependence, inflexibility, poor data security, and an inability to share data among applications.



The use of a traditional approach to file processing encourages each functional area in a corporation to develop specialized applications. Each application requires a unique data file that is likely to be a subset of the master file. These subsets of the master file lead to data redundancy and inconsistency, processing inflexibility, and wasted storage resources.

- **Data Redundancy and Inconsistency** Data redundancy is the presence of duplicate data in multiple data files so that the same data are stored in more than one place or location. Data redundancy occurs when different groups in an organization independently collect the same piece of data and store it independently of each other. Data redundancy wastes storage resources and also leads to data inconsistency, where the same attribute may have different values.

- **Program-Data Dependence**, Program-data dependence refers to the coupling of data stored in files and the specific programs required to update and maintain those files such that changes in programs require changes to the data. Every traditional computer program has to describe the location and nature of the data with which it works. In a traditional file environment, any change in a software program could require a change in the data accessed by that program. One program might be modified from a five-digit to a nine-digit ZIP code. If the original data file were changed from five-digit to nine-digit ZIP codes, then other programs that required the five-digit ZIP code would no longer work properly. Such changes could cost millions of dollars to implement properly.

- **Lack of Flexibility** A traditional file system can deliver routine scheduled reports after extensive programming efforts, but it cannot deliver ad hoc reports or respond to unanticipated information requirements in a timely fashion. The information required by ad hoc requests is somewhere in the system but may be too expensive to retrieve. Several programmers might have to work for weeks to put together the required data items in a new file.

- **Poor Security** Because there is little control or management of data, access to and dissemination of information may be out of control. Management may have no way of knowing who is accessing or even making changes to the organization’s data.

- **Lack of Data Sharing and Availability** Because pieces of information in different files and different parts of the organization cannot be related to one another, it is virtually impossible for information to be shared or accessed in a timely manner. Information cannot flow freely across different functional areas or different parts of the organization. If users find different values of the same piece of information in two different systems, they may not want to use these systems because they cannot trust the accuracy of their data.

🔖 Topic- 10.2: What are the major capabilities of database management systems (DBMS), and why is a relational DBMS so powerful?

10.2.1: Database Management Systems

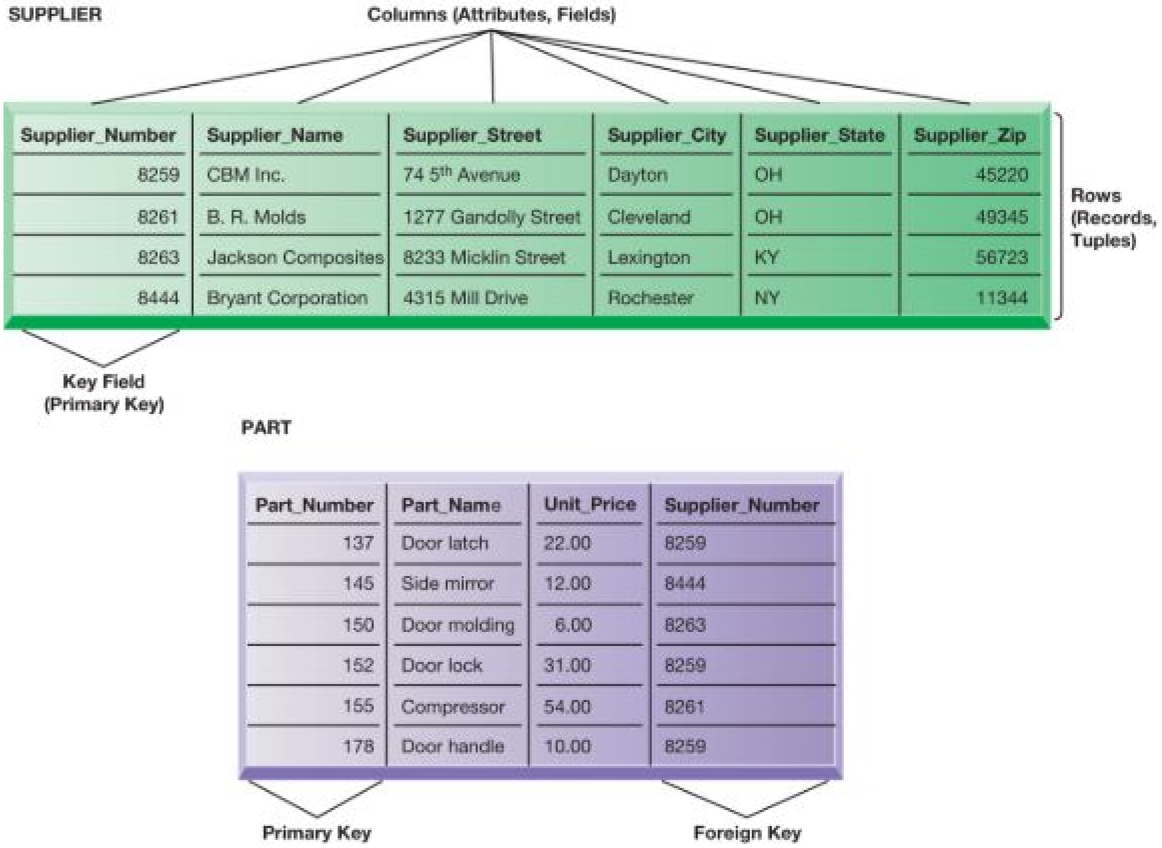
A database management system (DBMS) is software that permits an organization to centralize data, manage them efficiently, and provide access to the stored data by application programs. The DBMS acts as an interface between application programs and the physical data files. When the application program calls for a data item, such as gross pay, the DBMS finds this item in the database and presents it to the application program. Using traditional data files, the programmer would have to specify the size and format of each data element used in the program and then tell the computer where they were located.

- **How a DBMS Solves the Problems of the Traditional File Environment** A DBMS reduces data redundancy and inconsistency by minimizing isolated files in which the same data are repeated. The DBMS may not enable the organization to eliminate data redundancy entirely, but it can help control redundancy. Even if the organization maintains some redundant data, using a DBMS eliminates data inconsistency because the DBMS can help the organization ensure that every occurrence of redundant data has the same values. The DBMS uncouples programs and data, enabling data to stand on their own. The description of the data used by the program does not have to be specified in detail each time a different program is written. Access and availability of information will be increased and program development and maintenance costs reduced because users and programmers can perform ad hoc queries of the database for many simple applications without having to write complicated programs.

- **Relational DBMS** Contemporary DBMS use different database models to keep track of entities, attributes, and relationships. The most popular type of DBMS today for PCs as well as for larger computers and mainframes is the relational DBMS. Relational databases represent data as two-dimensional tables (called relations). Tables may be referred to as files. Each table contains data on an entity and its attributes. Microsoft Access is a relational DBMS for desktop systems, whereas DB2, Oracle Database, and Microsoft SQL Server are relational DBMS for large mainframes and midrange computers. MySQL is a popular open source DBMS.

  Let’s look at how a relational database organizes data about suppliers and parts (see Figure 6. 4). The database has a separate table for the entity SUPPLIER and a table for the entity PART. Each table consists of a grid of columns and rows of data. Each individual element of data for each entity is stored as a separate field, and each field represents an attribute for that entity. Fields in a relational database are also called columns. For the entity SUPPLIER, the supplier identification number, name, street, city, state, and ZIP code are stored as separate fields within the SUPPLIER table and each field represents an attribute for the entity SUPPLIER.

  The actual information about a single supplier that resides in a table is called a row. Rows are commonly referred to as records, or in very technical terms, as tuples . Data for the entity PART have their own separate table. The field for Supplier_Number in the SUPPLIER table uniquely identifies each record so that the record can be retrieved, updated, or sorted. It is called a key field. Each table in a relational database has one field that is designated as its primary key. This key field is the unique identifier for all the information in any row of the table and this primary key cannot be duplicated. Supplier_Number is the primary key for the SUPPLIER table and Part_Number is the primary key for the PART table. Note that Supplier_Number appears in both the SUPPLIER and PART tables. In the SUPPLIER table, Supplier_Number is the primary key. When the field Supplier_Number appears in the PART table, it is called a foreign key and is essentially a lookup field to look up data about the supplier of a specific part.


SUPPLIER
Columns (Attributes, Fields)
Rows (Records, Tuples)
Key Field (Primary Key)
PART
Primary Key
Foreign Key

Supplier_Number	Supplier_Name	Supplier_Street	Supplier_City	Supplier_State	Supplier_Zip
8259	CBM Inc.	74 5th Avenue	Dayton	OH	45220
8261	B. R. Molds	1277 Gandolly Street	Cleveland	OH	49345
8263	Jackson Composites	8233 Micklin Street	Lexington	KY	56723
8444	Bryant Corporation	4315 Mill Drive	Rochester	NY	11344

Part_Number	Part_Name	Unit_Price	Supplier_Number
137	Door latch	22.00	8259
145	Side mirror	12.00	8444
150	Door molding	6.00	8263
152	Door lock	31.00	8259
155	Compressor	54.00	8261
178	Door handle	10.00	8259

Entity Relationship Diagram Database designers document their data model with an entity-relationship diagram, illustrated in Figure 6. 11 . This diagram illustrates the relationship between the entities SUPPLIER, PART, LINE_ITEM, and ORDER. The boxes represent entities. The lines connecting the boxes represent relationships. A line connecting two entities that ends in two short marks designates a one-to-one relationship. A line connecting two entities that ends with a crow's foot topped by a short mark indicates a one-to-many relationship. Figure 6. 11shows that one ORDER can contain many LINE_ITEMs. (A PART can be ordered many times and appear many times as a line item in a single order.) Each PART can have only one SUPPLIER, but many PARTs can be provided by the same SUPPLIER. It can't be emphasized enough: If the business doesn't get its data model right, the system won't be able to serve the business well. The company's systems will not be as effective as they could be because they'll have to work with data that may be inaccurate, incomplete, or difficult to retrieve.

Understanding the organization’s data and how they should be represented in a database is perhaps the most important lesson you can learn from this course.