DATABASE MANAGEMENT SYSTEM

Topics Covered:

Database Management System (DBMS) and *Its Applications*
Advantage of dbms:
Disadvantage of dbms:
View of Data :
Data abstraction:
Data model and its types:

Database Management System:

A database-management system (**DBMS**) is a collection of inter -related data and a set of programs to access those data.

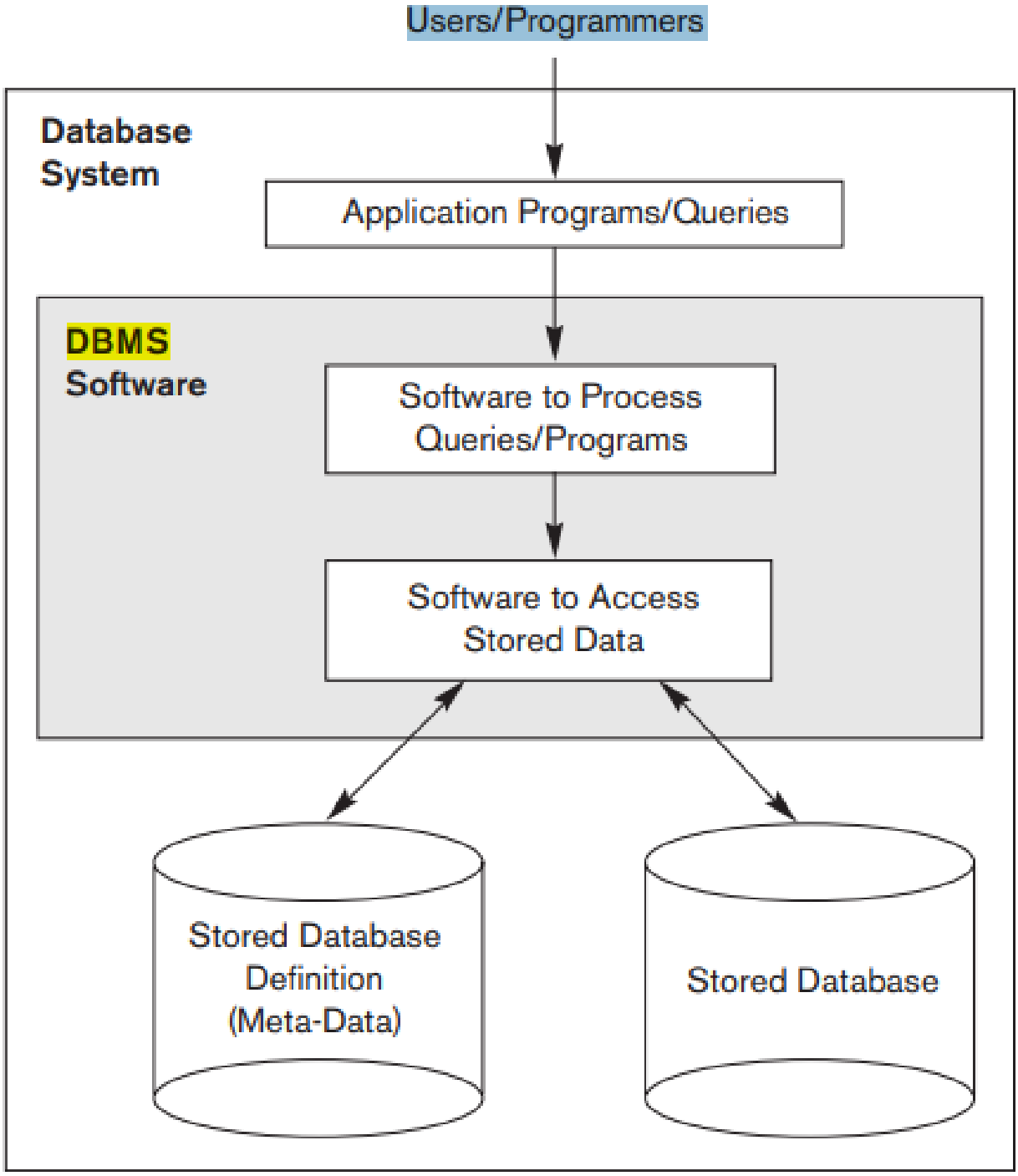
The primary goal of a **DBMS** is to provide a way to store and retrieve database information that is both convenient and efficient.

Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information.

In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

(DBMS) an*d Its Applications:*

- ➔ A Database management system is a computerized record-keeping system.
- ➔ It is a repository or a container for collection of computerized data files.
- ➔ The overall purpose of DBMS is to allow the users to define, store, retrieve and update the information contained in the database on demand.
- ➔ Information can be anything that is of significance to an individual or organization.


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Databases touch all aspects of our lives. Some of the major areas of application are as follows:

- ☐ Banking
- ☐ Airlines
- ☐ Universities
- ☐ Manufacturing and selling
- ☐ Human resources

Examples 😀

1.College(AMC) : For student information, course registration, and grades..(in addition to standard enterprise information such as human resources and accounting).

2. Banking and Finance: i)Banking: For customer information, accounts, loans, and banking transaction..etc….
ii)Credit card transactions : For purchase on credit cards and generation of monthly statements.

3.Sales(enterprise information): For customer, product, and purchase information.i)Accounting: for payments, receipts, account balances, assets and other accounting information.

Advantages of DBMS:

1. **Controlling of Redundancy:** Data redundancy refers to the duplication of data (i.e storing same data multiple times).

In a database system, by having a centralized database and centralized control of data by the DBA the unnecessary duplication of data is avoided.

It also eliminates the extra time for processing the large volume of data. It results in saving the storage space.

2. **Improved Data Sharing :** DBMS allows a user to share the data in any number of application programs.
3. **Data Integrity** : Integrity means that the data in the database is accurate. Centralized control of the data helps in permitting the administrator to define integrity constraints to the data in the database.
4. **Data Consistency** : By eliminating data redundancy, we greatly reduce the opportunities for inconsistency. For example: is a customer address is stored only once, Also updating data values is greatly simplified when each value is stored in one place only. Finally, we avoid the wasted storage that results from redundant data storage

Efficient Data Access :

In a database system, the data is managed by the DBMS and all access to the data is through the DBMS providing a key to effective data processing

- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
- **Reduce time:** It reduces development time and maintenance need.
- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

Disadvantages of DBMS 😂

- **Cost of Hardware and Software**: It requires a high speed of data processor and large memory size to run DBMS software.
- **Size**: It occupies a large space of disks and large memory to run them efficiently.
- **Complexity:** Database system creates additional complexity and requirements.
- **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

View of Data :

A database system is a collection of interrelated data and a set of programs that allow users to access and modify these data. A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored and maintained.

Data Abstraction:

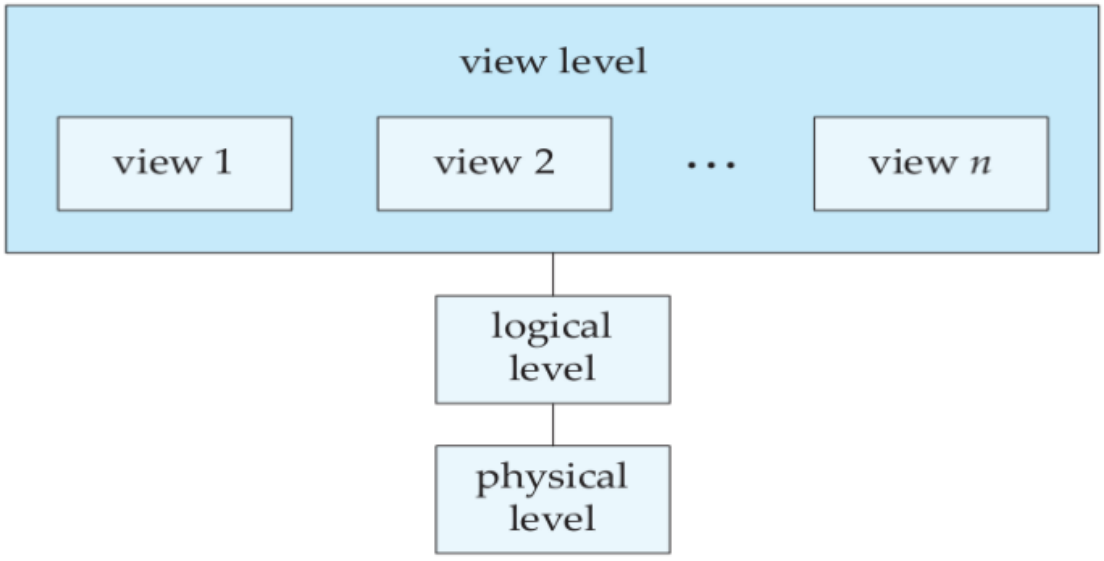
For the system to be usable, it must retrieve data efficiently. The need for efficiency has led designers to use complex data structures to represent data in the database.

- ❖ Since many database-system users are not computer trained,
- ❖ developers hide the complexity from users through several levels of abstraction,
- ❖ to simplify users' interactions with the system.

Purpose of Data Abstraction 😀

The main purpose of data abstraction is to hide irrelevant data and provide an abstract view of the data. With the help of data abstraction, developers hide irrelevant data from the user and provide them the relevant data. By doing this, users can access the data without any hassle, and the system will also work efficiently.
In DBMS, data abstraction is performed in layers which means there are levels of data abstraction in DBMS that we will further study in this article. Based on these levels, the database management system is designed.

Levels of Data Abstractions in DBMS 🙂



1. Physical or Internal Level:

The physical or internal layer is the lowest level of data abstraction in the database management system. It is the layer that defines how data is actually stored in the database.

2. Logical or Conceptual Level:

The logical or conceptual level is the intermediate or next level of data abstraction. It explains what data is going to be stored in the database and what the relationship is between them.

3. View or External Level:

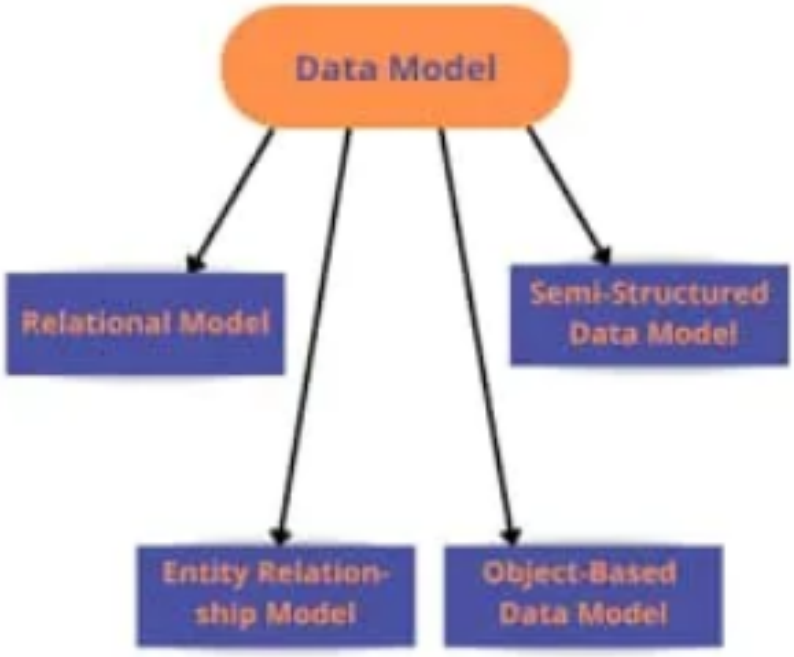
View or External Level is the highest level of data abstraction. There are different views at this level that define the parts of the overall data of the database. This level is for the end-user interaction; at this level, end users can access the data based on their queries.

Instances and Schemas

Databases change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an **instance of the database**. The overall design of the database is called the **database schema.**

Data Models

Data model: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. A data model provides a way to describe the design of a database at the physical, logical, and view levels.



1) Entity-Relationship Model:

Entity-Relationship Model or simply ER Model is a high-level data model diagram. In this model real-world problems are presented in pictorial format to make it simple for the stakeholders to understand. Additionally, designers can quickly comprehend the system by simply looking at the ER diagram. The ER diagram is a visual representation of an ER Model.

The ER diagram consists of the following three elements:

I. **Entities**: Entities are real-world objects. It could be a specific person, place, or even an idea. Entities are represented in a rectangle shape in ER diagram.

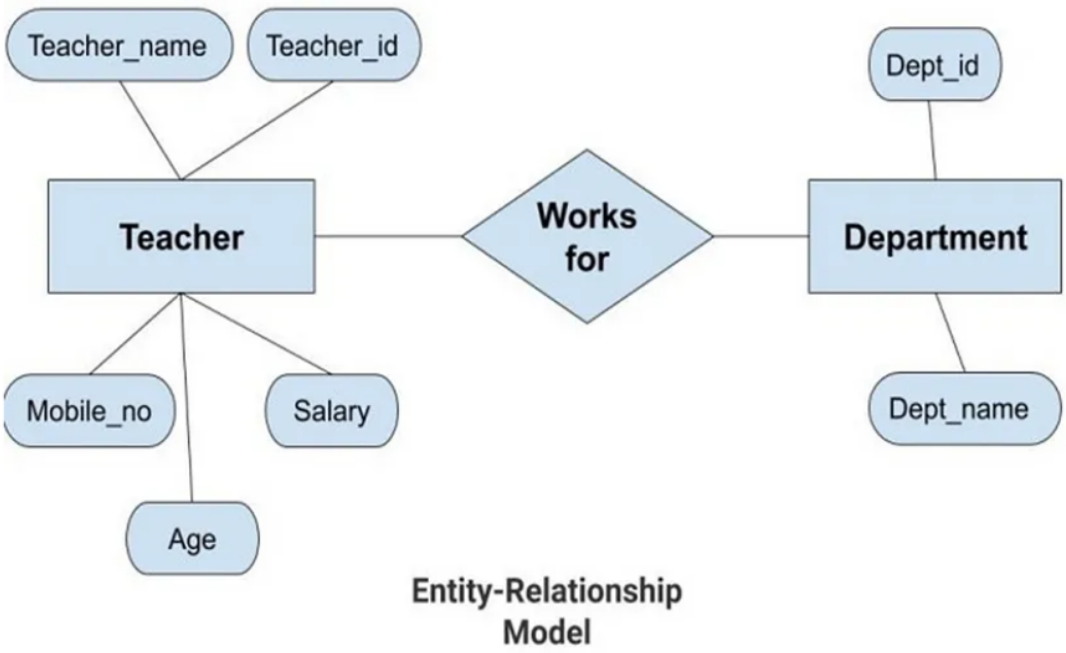
Ex. Teacher, Student, Son, College, etc.

II. **Attributes**: An entity has a real-world property known as an attribute. These are the traits associated with that property. Attributes are represented in an oval shape in ER diagram.

Ex. Entity Student has attributes like roll no, age, address, etc.

III. **Relationship**: Relationship describes the connection between two attributes. Relationship is represented in a diamond shape in ER diagram.

Ex. Teacher works in a department, Student enrolls in a course.

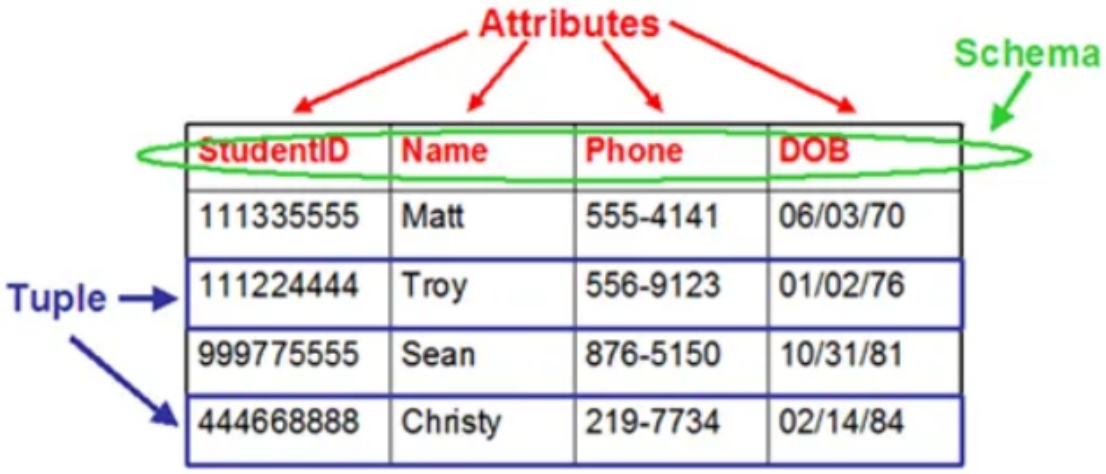


Entity-Relationship Model

2.Relational Model:

The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. Tables are also known as relations.

Relational data model is the primary data model, which is used widely around the world for data storage and processing. This model is simple, and it has all the properties and capabilities required to process data with storage efficiency.



- Relational Model represents data as collection a of tables.
- A table is also called a relation.
- Each row of the table is called a tuple.
- Column headers are Attributes.

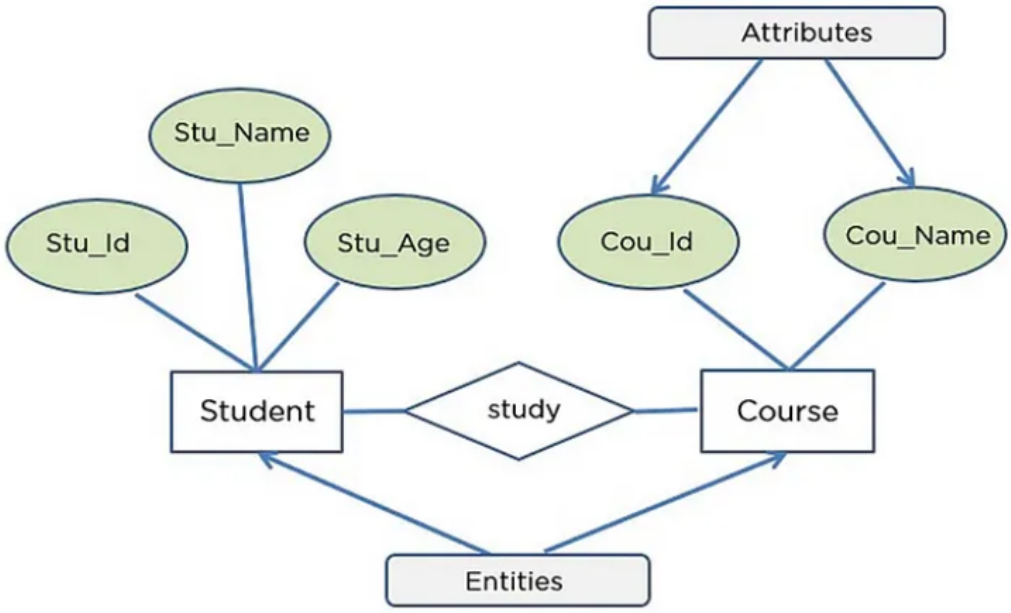
3)Semi-Structured data model:

The semi-structured data model permits the specification of data where individual data items of the same type may have different sets of attributes.

This is in contrast to the data models mentioned earlier, where every data item of a particular type must have the same set of attributes.

The Extensible Markup Language (XML) is widely used to representsemi-structured data.

ER diagram for Semi-structured data model

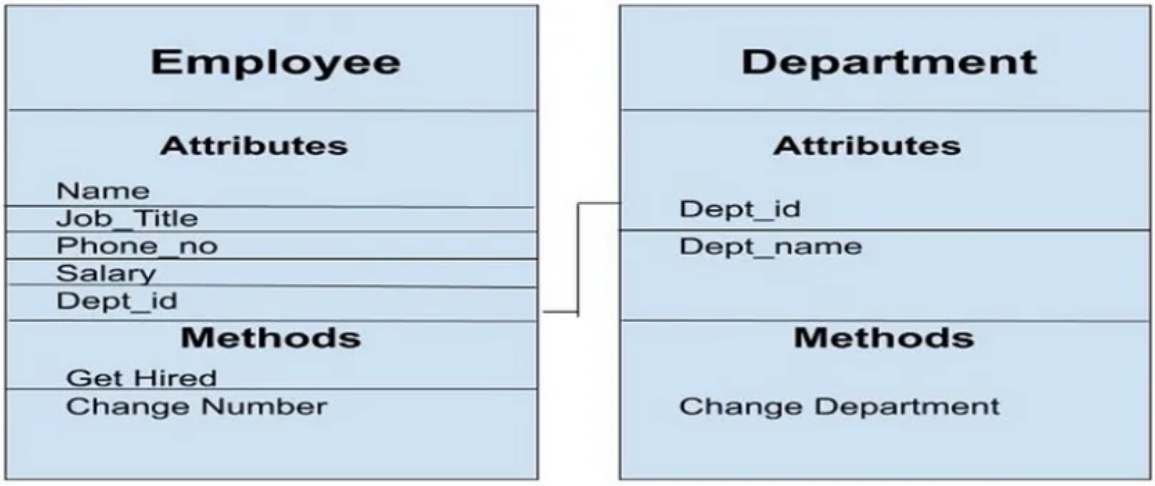


- **Entities**: The entities here are student and course
- **Attributes**: The attributes for entity student are roll number, name, address, and age, and for entity course id and course name are the attributes.
- **Relationship**: here relationship between entities student and course is study.

4)Object-Oriented Data Model:

★ object-oriented data model is a combination of object-oriented programming, and relational data model.

★ It is an extension of Entity-Relationship data model with an idea of functions, encapsulation, and object identity, as well. This model defines a database as a collection of objects with associated features and methods.
★ In object-oriented data model, both the data and relationship are present in a single structure known as an object.
★ We can store audio, video, images, etc in the database with the help of this data model.
★ In this model, two are more objects are connected through links. We use this link to relate one object to other objects. This can be understood by the example given below.



Object_Oriented_Model

In the above example, we have two objects Employee and Department. All the data and relationships of each object are contained as a single unit. The attributes like Name, Job_title of the employee and the methods which will be performed by that object are stored as a single object. The two objects are connected through a common attribute i.e. the Department_id and the communication between these two will be done with the help of this common id.

Historically, the **network data model** and the **hierarchical data** model preceded the relational data model. These models were tied closely to the underlying implementation, and complicated the task of modeling data. As a result they are used little now, except in old database code that isstill in service in some places