Topic Three: Data Models and Conceptual Modeling

- **A data model** is an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.
- A model is a representation of 'real world' objects and events, and their associations.
- It should provide the basic concepts and notations that will allow database designers and end-users. unambiguously and accurately to communicate their understanding of the organizational data.
- A data model can be thought of as comprising three components:
  - a **structural part**, consisting of a set of rules according to which databases can be constructed;
  - a **manipulative part**, defining the types of operation that are allowed on the data (this includes the operations that are used for updating or retrieving data from the database and for changing the structure of the database);
  - possibly a **set of integrity constraints**, which ensures that the data is accurate.
- The purpose of a data model is to represent data and to make the data understandable. If it does this, then it can be easily used to design a database.
- Data models fall into three broad categories: **object-based**, **record-based**, and **physical** data models. The first two are used to describe data at the conceptual and external levels, the latter is used to describe data at the internal level.

Object-Based Data Models

- Object-based data models use concepts such as entities, attributes, and relationships.
- An **entity** is a distinct object (a person, place, thing, concept, event) in the organization that is to be represented in the database.
- An **attribute** is a property that describes some aspect of the object that we wish to record, and a **relationship** is an association between entities.
- Some of the more common types of object-based data model are:
  - o Entity–Relationship
  - o Semantic
  - o Functional
  - o Object-Oriented.
- The Entity–Relationship model has emerged as one of the main techniques for database design and forms the basis for the database design methodology.
- The object-oriented data model extends the definition of an entity to include not only the attributes that describe the **state** of the object but also the actions that are associated with the object, that is, its **behavior**.
- The object is said to **encapsulate** both state and behavior.

Record-Based Data Models

- In a record-based model, the database consists of a number of fixed-format records possibly of differing types.
- Each record type defines a fixed number of fields, each typically of a fixed length.
- There are three principal types of record-based logical data model: the **relational data model**, the **network data model**, and the **hierarchical data model**.
- The hierarchical and network data models were developed almost a decade before the relational data model, so their links to traditional file processing concepts are more evident.

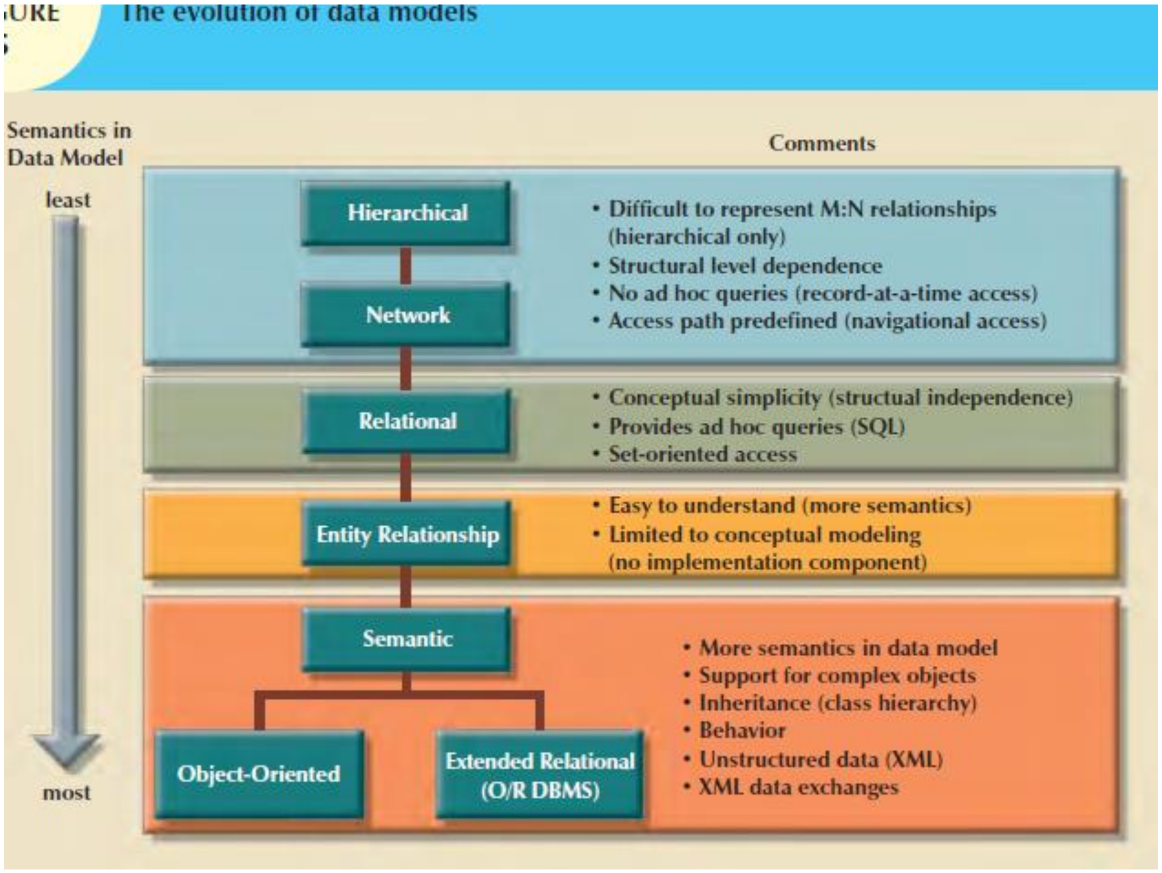
Physical Data Models

- Physical data models describe how data is stored in the computer, representing information such as record structures, record orderings, and access paths.
- There are not as many physical data models as logical data models, the most common ones being the *unifying model* and the *frame memory*.

Conceptual Modeling

- From an examination of the three-level architecture, we see that the conceptual schema is the 'heart' of the database.
- It supports all the external views and is, in **turn**, supported by the internal schema.
- However, the internal schema is merely the physical implementation of the conceptual schema.
- The conceptual schema should be a complete and accurate representation of the data requirements of the enterprise.
- **Conceptual modeling**, or **conceptual database design**, is the process of constructing a model of the information use in an enterprise that is independent of implementation details, such as the target DBMS, application programs, programming languages, or any other physical considerations.
- This model is called a **conceptual data model**.

Evolution of Database Models


URE
The evolution of data models
Semantics in Data Model
least
most
Comments
Hierarchical
Network
Relational
Entity Relationship
Semantic
Object-Oriented
Extended Relational (O/R DBMS)
• Difficult to represent M:N relationships (hierarchical only)
• Structural level dependence
• No ad hoc queries (record-at-a-time access)
• Access path predefined (navigational access)
• Conceptual simplicity (structual independence)
• Provides ad hoc queries (SQL)
• Set-oriented access
• Easy to understand (more semantics)
• Limited to conceptual modeling (no implementation component)
• More semantics in data model
• Support for complex objects
• Inheritance (class hierarchy)
• Behavior
• Unstructured data (XML)
• XML data exchanges

Hierarchical data model

- The hierarchical model is a restricted type of network model. Again, data is represented as collections of records and relationships are represented by sets.
- However, the hierarchical model allows a node to have only one parent.
- A hierarchical model can be represented as a tree graph, with records appearing as nodes (also called segments) and sets as edges.
- Record-based (logical) data models are used to specify the overall structure of the database and a higher-level description of the implementation.
- Its basic logical structure is represented by an upside-down tree. The hierarchical structure contains levels, or segments.
- A segment is the equivalent of a file system's record type.
- Within the hierarchy, a higher layer is perceived as the parent of the segment directly beneath it, which is called the child.
- The hierarchical model depicts a set of one-to-many (1:M) relationships between a parent and its children segments. (Each parent can have many children, but each child has only one parent.)

- Their main drawback lies in the fact that they do not provide adequate facilities for explicitly specifying constraints on the data, whereas the object-based data models lack the means of logical structure specification but provide more semantic substance by allowing the user to specify constraints on the data.

Network data model

- In the network model, data is represented as collections of **records**, and relationships are represented by **sets**.
- Compared with the relational model, relationships are explicitly modeled by the sets, which become pointers in the implementation.
- The records are organized as generalized graph structures with records appearing as **nodes** (also called **segments**) and sets as **edges** in the graph.
- It was created to represent complex data relationships more effectively than the hierarchical model, to improve database performance, and to impose a database standard.
- In the network model, the user perceives the network database as a collection of records in 1:M relationships. However, unlike the hierarchical model, the network model allows a record to have more than one parent.
- While the network database model is generally not used today, the definitions of standard database *concepts* that emerged with the network model are still used by modern data models. Some important concepts that were defined at this time are:
    - ❖ The **schema**, which is the conceptual organization of the entire database as viewed by the database administrator.
    - ❖ The **subschema**, which defines the portion of the database "seen" by the application programs that actually produce the desired information from the data contained within the database.
    - ❖ A **data management language** (**DML**), which defines the environment in which data can be managed and to work with the data in the database.
    - ❖ A schema **data definition language** (**DDL**), which enables the database administrator to define the schema components.

Note: The majority of modern commercial systems are based on the relational paradigm, whereas the early database systems were based on either the network or hierarchical data models. The latter two models require the user to have knowledge of the physical database being accessed, whereas the former provides a substantial amount of data independence. Hence, while relational systems adopt a **declarative** approach to database processing (that is, they specify *what* data is to be retrieved), network and hierarchical systems adopt a **navigational** approach (that is, they specify *how* the data is to be retrieved).

Relational data model

- The relational data model is based on the concept of mathematical relations.

- In the relational model, data and relationships are represented as tables, each of which has a number of columns with a unique name.
- Note that the relational data model requires only that the database be perceived by the user as tables.
- However, this perception applies only to the logical structure of the database, that is, the external and conceptual levels of the ANSI-SPARC architecture.
- It does not apply to the physical structure of the database, which can be implemented using a variety of storage structures.

The Entity Relationship Model

- Complex design activities require conceptual simplicity to yield successful results.
- Although the relational model was a vast improvement over the hierarchical and network models, it still lacked the features that would make it an effective database design tool.
- Because it is easier to examine structures graphically than to describe them in text, database
- designers prefer to use a graphical tool in which entities and their relationships are pictured. Thus, the entity relationship (ER) model, or ERM, has become a widely accepted standard for data modeling.
- Peter Chen first introduced the ER data model in 1976; it was the graphical representation of entities and their relationships in a database structure that quickly became popular because it complemented the relational data model concepts.
- The relational data model and ERM combined to provide the foundation for tightly structured database design.
- ER models are normally represented in an entity relationship diagram (ERD), which uses graphical representations to model database components.

The Object-Oriented (OO) Model

- Increasingly complex real-world problems demonstrated a need for a data model that more closely represented the real world.
- In the object-oriented data model (OODM), both data and their relationships are contained in a single structure known as an object.
- An OODM reflects a very different way to define and use entities. Like the relational model's entity, an object is described by its factual content.
- But quite unlike an entity, an object includes information about relationships between the facts within the object, as well as information about its relationships with other objects. Therefore, the facts within the object are given greater meaning.
- The OODM is said to be a semantic data model because semantic indicates meaning.
- Subsequent OODM development has allowed an object to also contain all operations that can be performed on it, such as changing its data values, finding a specific data value, and printing data values.

- Because objects include data, various types of relationships, and operational procedures, the object becomes self-contained, thus making the object—at least potentially—a basic building block for autonomous structures.

The OO data model is based on the following components:

- ❖ An object is an abstraction of a real-world entity. In general terms, an object may be considered equivalent to an ER model's entity.
- ❖ More precisely, an object represents only one occurrence of an entity. (The object's semantic content is defined through several of the items in this list.)
- ❖ Attributes describe the properties of an object. For example, a PERSON object includes the attributes Name, Identification Number, and Date of Birth.
- ❖ Objects that share similar characteristics are grouped in classes. A **class** is a collection of similar objects with shared structure (attributes) and behavior (methods).
- ❖ Classes are organized in a *class hierarchy*. The **class hierarchy** resembles an upside-down tree in which each class has only one parent. For example, the CUSTOMER class and the EMPLOYEE class share a parent PERSON class. (Note the similarity to the hierarchical data model in this respect.) **Inheritance** is the ability of an object within the class hierarchy to inherit the attributes and methods of the classes above it. For example, two classes, CUSTOMER and EMPLOYEE, can be created as subclasses from the class PERSON. In this case, CUSTOMER and EMPLOYEE will inherit all attributes and methods from PERSON.

- Object-oriented data models are typically depicted using Unified Modeling Language (UML) class diagrams.
- Unified Modeling Language (UML) is a language based on OO concepts that describes a set of diagrams and symbols that can be used to graphically model a system.
- UML class diagrams are used to represent data and their relationships within the larger UML object-oriented system's modeling language.

Object/Relational and XML

- Facing the demand to support more complex data representations, the relational model's main vendors evolved the model further and created the extended relational data model (ERDM).
- The ERDM adds many of the OO model's features within the inherently simpler relational database structure.
- The ERDM gave birth to a new generation of relational databases supporting OO features such as objects (encapsulated data and methods), extensible data types based on classes, and inheritance. That's why a DBMS based on the ERDM is often described as an object/relational database management system (O/R DBMS).

- The use of complex objects received a boost with the Internet revolution. When organizations integrated their business models with the Internet, they realized the potential of the Internet to access, distribute, and exchange critical business information.
- This resulted in the widespread adoption of the Internet as a business communication tool. It is in this environment that Extensible Markup Language (XML) emerged as the de facto standard for the efficient and effective exchange of structured, semistructured, and unstructured data.
- Organizations using XML data soon realized there was a need to manage the large amounts of unstructured data such as word-processing documents, Web pages, e-mails, diagrams, etc., found in most of today's organizations.
- To address this need, XML databases emerged to manage unstructured data within a native XML format

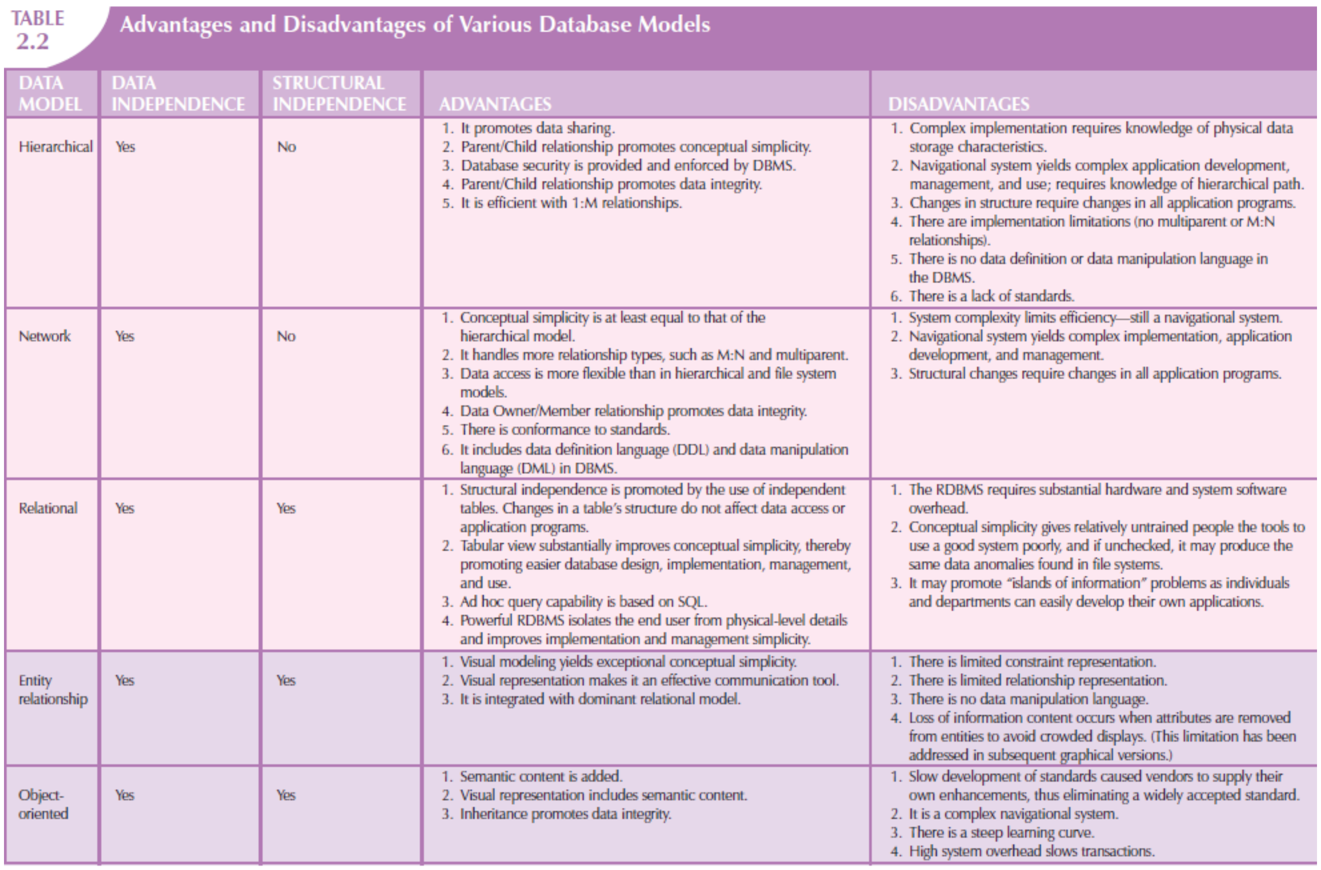


TABLE 2.2 Advantages and Disadvantages of Various Database Models

DATA MODEL	DATA INDEPENDENCE	STRUCTURAL INDEPENDENCE	ADVANTAGES	DISADVANTAGES
Hierarchical	Yes	No	1. It promotes data sharing. 2. Parent/Child relationship promotes conceptual simplicity. 3. Database security is provided and enforced by DBMS. 4. Parent/Child relationship promotes data integrity. 5. It is efficient with 1:M relationships.	1. Complex implementation requires knowledge of physical data storage characteristics. 2. Navigational system yields complex application development, management, and use; requires knowledge of hierarchical path. 3. Changes in structure require changes in all application programs. 4. There are implementation limitations (no multiparent or M:N relationships). 5. There is no data definition or data manipulation language in the DBMS. 6. There is a lack of standards.
Network	Yes	No	1. Conceptual simplicity is at least equal to that of the hierarchical model. 2. It handles more relationship types, such as M:N and multiparent. 3. Data access is more flexible than in hierarchical and file system models. 4. Data Owner/Member relationship promotes data integrity. 5. There is conformance to standards. 6. It includes data definition language (DDL) and data manipulation language (DML) in DBMS.	1. System complexity limits efficiency—still a navigational system. 2. Navigational system yields complex implementation, application development, and management. 3. Structural changes require changes in all application programs.
Relational	Yes	Yes	1. Structural independence is promoted by the use of independent tables. Changes in a table's structure do not affect data access or application programs. 2. Tabular view substantially improves conceptual simplicity, thereby promoting easier database design, implementation, management, and use. 3. Ad hoc query capability is based on SQL. 4. Powerful RDBMS isolates the end user from physical-level details and improves implementation and management simplicity.	1. The RDBMS requires substantial hardware and system software overhead. 2. Conceptual simplicity gives relatively untrained people the tools to use a good system poorly, and if unchecked, it may produce the same data anomalies found in file systems. 3. It may promote "islands of information" problems as individuals and departments can easily develop their own applications.
Entity relationship	Yes	Yes	1. Visual modeling yields exceptional conceptual simplicity. 2. Visual representation makes it an effective communication tool. 3. It is integrated with dominant relational model.	1. There is limited constraint representation. 2. There is limited relationship representation. 3. There is no data manipulation language. 4. Loss of information content occurs when attributes are removed from entities to avoid crowded displays. (This limitation has been addressed in subsequent graphical versions.)
Object-oriented	Yes	Yes	1. Semantic content is added. 2. Visual representation includes semantic content. 3. Inheritance promotes data integrity.	1. Slow development of standards caused vendors to supply their own enhancements, thus eliminating a widely accepted standard. 2. It is a complex navigational system. 3. There is a steep learning curve. 4. High system overhead slows transactions.