Database Models

- A database model is a collection of logical constructs used to represent the data structure and the data relationships found within the database.

- Two Categories of Database Models
  - **Conceptual models** focus on the logical nature of the data representation. They are concerned with *what* is represented rather than *how* it is represented.
  - **Implementation models** place the emphasis on *how* the data are represented in the database or on *how* the data structures are implemented.

Database Models

- Three Types of Relationships
  - **One-to-many relationships (1:M)**
    - A painter paints many different paintings, but each one of them is painted by only that painter.
      - PAINTER (1) paints PAINTING (M)
  - **Many-to-many relationships (M:N)**
    - An employee might learn many job skills, and each job skill might be learned by many employees.
      - EMPLOYEE (M) learns SKILL (N)
  - **One-to-one relationships (1:1)**
    - Each store is managed by a single employee and each store manager (employee) only manages a single store.
      - EMPLOYEE (1) manages STORE (1)

Database Models

- Three Types of Implementation Database Models
  - **Hierarchical database model**
  - **Network database model**
  - **Relational database model**

A Hierarchical Structure

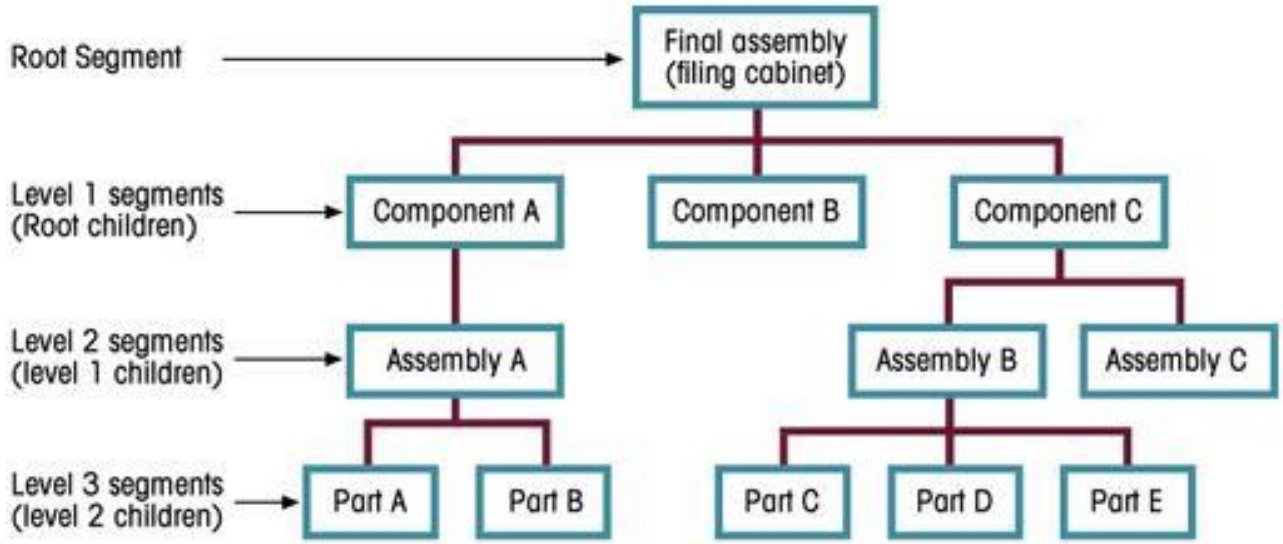

FIGURE 1.8 A HIERARCHICAL STRUCTURE

Figure 1.8

Database Models

- Hierarchical Database Model
  - **Basic Structure**
    - Collection of records logically organized to conform to the upside-down tree (hierarchical) structure.
    - The top layer is perceived as the parent of the segment directly beneath it.
    - The segments below other segments are the children of the segment above them.
    - A tree structure is represented as a hierarchical path on the computer's storage media.

Database Models

- Hierarchical Database Model
  - **Advantages**
    - Conceptual simplicity
    - Database security
    - Data independence
    - Database integrity
    - Efficiency dealing with a large database
  - **Disadvantages**
    - Complex implementation
    - Difficult to manage
    - Lacks structural independence
    - Applications programming and use complexity
    - Implementation limitations
    - Lack of standards

Child with Multiple Parents

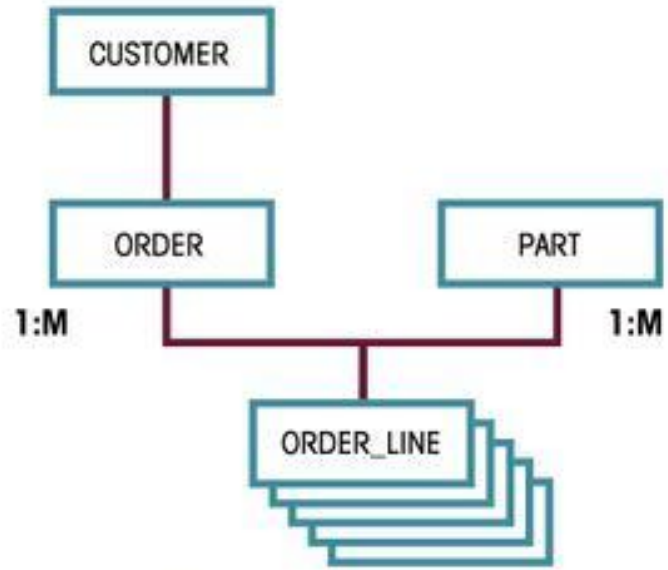

FIGURE 1.9 CHILD WITH MULTIPLE PARENTS

Figure 1.9

Database Models

- Network Database Model
  - **Basic Structure**
    - **Set** -- A relationship is called a set. Each set is composed of at least two record types: an owner (parent) record and a member (child) record.
    - A set is represents a 1:M relationship between the owner and the member.

A Network Database Model

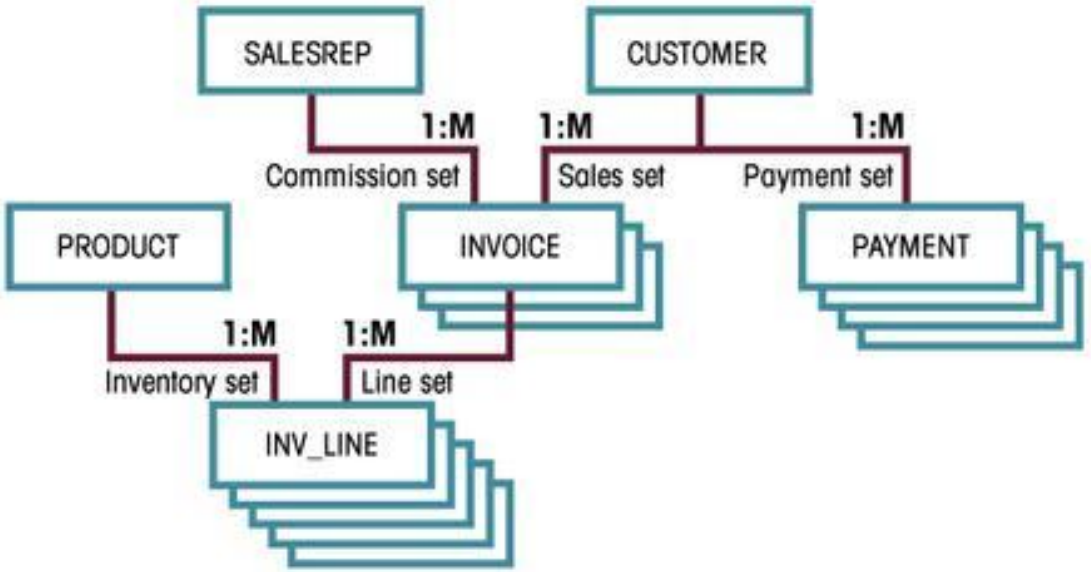

FIGURE 1.10 A NETWORK DATABASE MODEL

Figure 1.10

Database Models

- Network Database Model
  - **Advantages**
    - Conceptual simplicity
    - Handles more relationship types
    - Data access flexibility
    - Promotes database integrity
    - Data independence
    - Conformance to standards
  - **Disadvantages**
    - System complexity
    - Lack of structural independence

Database Models

- Relational Database Model
  - **Basic Structure**
    - RDBMS allows operations in a human logical environment.
    - The relational database is perceived as a collection of tables.
    - Each table consists of a series of row/column intersections.
    - Tables (or relations) are related to each other by sharing a common entity characteristic.
    - The relationship type is often shown in a relational schema.
    - A table yields complete data and structural independence.

Linking Relational Tables

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	3/12/99	502
10011	Dunne	Leona	K	713	894-1238	5/23/99	501
10012	Smith	Kathy	W	615	894-2285	1/5/98	502
10013	Olowski	Paul	F	615	894-2180	9/20/99	502
10014	Orlando	Myron		615	222-1672	12/4/99	501
10015	O'Brian	Amy	B	713	442-3381	8/29/99	503
10016	Brown	James	G	615	297-1228	3/1/99	502
10017	Williams	George		615	290-2556	6/23/99	503
10018	Farriss	Anne	G	713	382-7185	11/9/99	501
10019	Smith	Olette	K	615	297-3809	2/18/99	503

link throught AGENT_CODE

Table name: AGENT

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

FIGURE 1.11 LINKING RELATIONAL TABLES

Figure 1.11

Database Models

- Relational Database Model
  - **Advantages**
    - Structural independence
    - Improved conceptual simplicity
    - Easier database design, implementation, management, and use
    - Ad hoc query capability (SQL)
    - Powerful database management system
  - **Disadvantages**
    - Substantial hardware and system software overhead
    - Possibility of poor design and implementation
    - Potential “islands of information” problems

Relational Model

- Record- and table-based model
- **Relational database modeling is a *logical-level* model**
- Proposed by E.F. Codd
- Based on mathematical relations
- Uses relations, represented as tables
- Columns of tables represent attributes
- Tables represent relationships as well as entities
- Successor to earlier record-based models—network and hierarchical

A Relational Schema

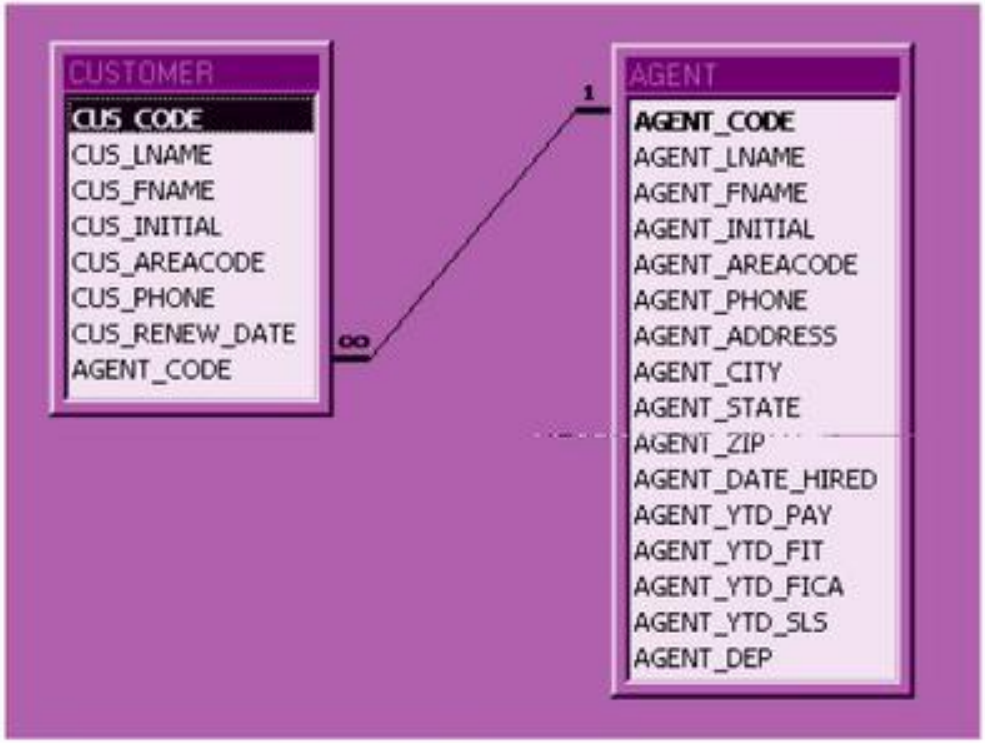

FIGURE 1.12 A RELATIONAL SCHEMA

Figure 1.12

Database Models

- Entity-Relationship Data Model
  - It is one of the most widely accepted graphical data modeling tools.
  - It graphically represents data as entities and their relationships in a database structure.
  - It complements the relational data model concepts.

Database Models

- Entity Relationship Data Model
  - **Basic Structure**
    - E-R models are normally represented in an **entity relationship diagram** (**ERD**).
    - An **entity** is represented by a rectangle.
    - Each entity is described by a set of attributes. An **attribute** describes a particular characteristics of the entity.
    - A **relationship** is represented by a diamond connected to the related entities.

Figure 1.13 Relationship Depiction: The ERD

A One-to-Many (1:M) Relationship: A PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE

FIGURE 1.13 RELATIONSHIP DEPICTION: THE ERD

Figure 1.14 Relationship Depiction: The Crow's Foot

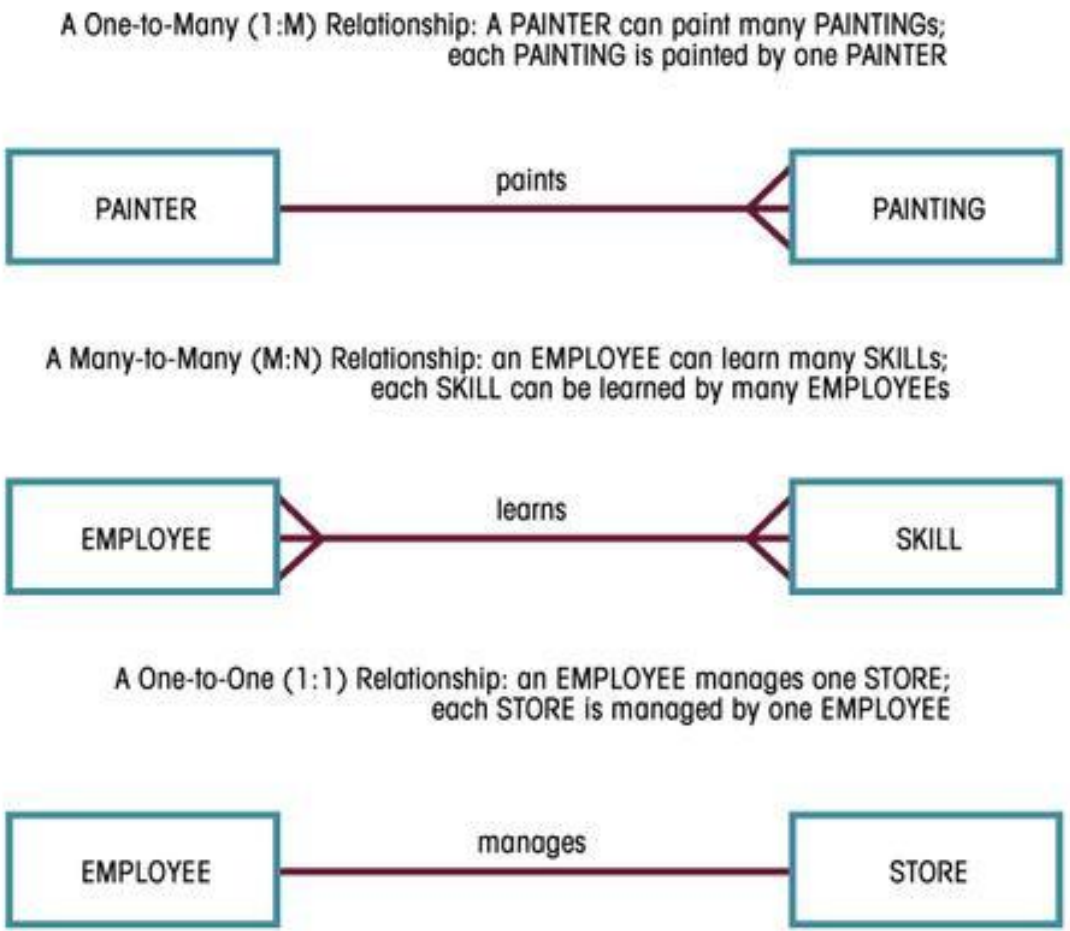

FIGURE 1.14 RELATIONSHIP DEPICTION: THE CROW'S FOOT

Database Models

- Entity-Relationship Data Model
  - **Advantages**
    - Exceptional conceptual simplicity
    - Visual representation
    - Effective communication tool
    - Integrated with the relational database model
  - **Disadvantages**
    - Limited constraint representation
    - Limited relationship representation
    - No data manipulation language
    - Loss of information content

Entity-Relationship Model

- A semantic model, captures meanings
- **E-R modeling is a *conceptual level* model**
- Proposed by P.P. Chen in 1970s
- **Entities** are real-world objects about which we collect data
- **Attributes** describe the entities
- **Relationships** are associations among entities
- **Entity set** – set of entities of the same type
- **Relationship set** – set of relationships of same type
- Relationships sets may have descriptive attributes

Database Models

- Object-Oriented Database Model
  - **Characteristics**
    - An object is described by its factual content.
    - An object includes information about relationships between the facts within the object, as well as with other objects.
    - An object is a self-contained building block for autonomous structures.

Database Models

- Object-Oriented Database Model
  - **Basic Structure**
    - **Objects** are abstractions of real-world entities or events.
    - **Attributes** describe the properties of an object.
    - 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.
    - An object can inherit the attributes and methods of the classes above it.

Object-oriented Model

- Uses the E-R modeling as a basis but extended to include **encapsulation**, **inheritance**
- Objects have both state and behavior
- **State** is defined by attributes
- **Behavior** is defined by methods (functions or procedures)
- Designer defines classes with attributes, methods, and relationships
- Class constructor method creates object instances
- Each object has a unique object ID
- Classes related by class hierarchies
- Database objects have persistence

Object-relational model

- Adds new complex datatypes to relational model
- Adds objects with attributes and methods
- Adds inheritance
- SQL extended to handle objects in SQL:1999
-

A Comparison: The OO Data Model and the ER Model

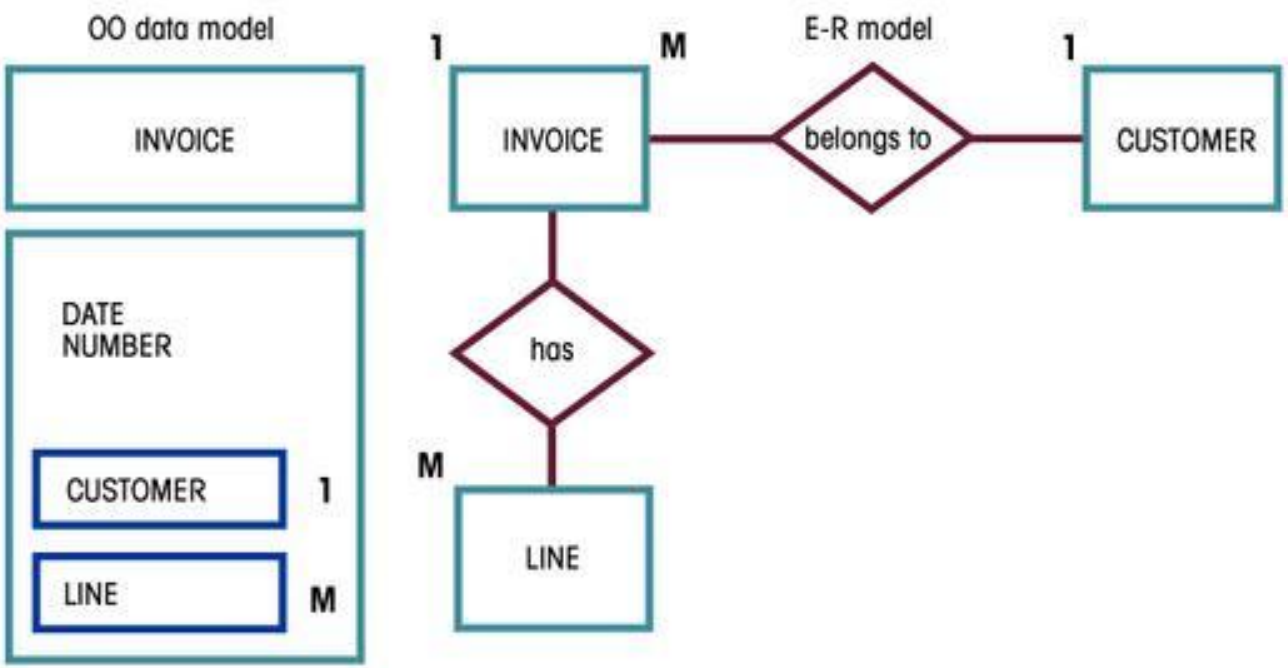

FIGURE 1.15 A COMPARISON OF THE OO DATA MODEL AND THE ER MODEL

Figure 1.15

Database Models

- Object-Oriented Database Model
  - **Advantages**
    - Add semantic content
    - Visual presentation includes semantic content
    - Database integrity
    - Both structural and data independence
  - **Disadvantages**
    - Lack of OODM standards
    - Complex navigational data access
    - Steep learning curve
    - High system overhead slows transactions

The Development of Data Models

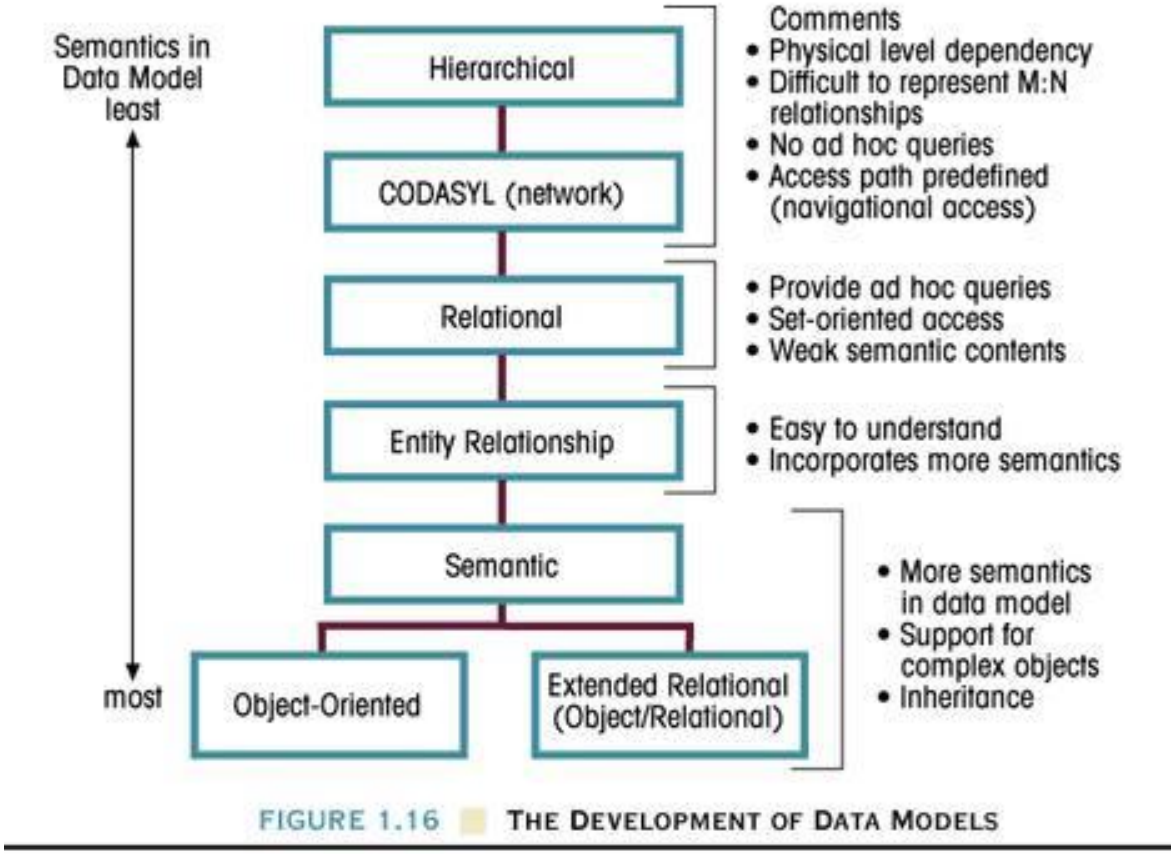

FIGURE 1.16 THE DEVELOPMENT OF DATA MODELS

Figure 1.16

- **Semi-structured Model**
- Collection of nodes, each with data, and with different schemas
- Each node contains a description of its own contents
- Can be used for integrating existing databases
- XML tags added to documents to describe structure
- XML tags identify elements, sub-elements, attributes in documents
- XML DTD (Document Type Definition) or XML Schema used to define structure