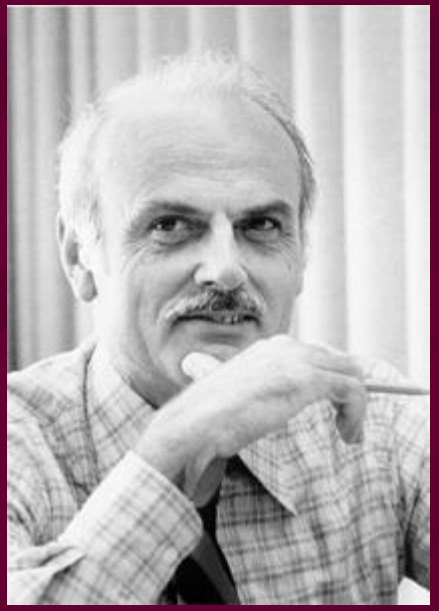
Database Models



Data Modeling and Data Models

- Data modeling: creating a specific data model for a determined problem domain
  - Model: abstraction of a more complex real-world object or event
  - Data model: simple representation of complex real-world data structures
    - Useful for supporting a specific problem domain

The Importance of Data Models

- The importance of data modeling cannot be overstated
  - Facilitates communication via visualization
    - Peers
    - Management/approval chain
  - Gives various views of the database
    - Different departments/people will be concerned with the data they need to work with
  - Organizes data for various users/departments
  - Provides an abstraction for the creation of good a database

Data Model Basic Building Blocks

- Entity: tangible or intangible object such as a person, place, thing, or event about which data will be collected and stored
  - Attribute: characteristic of an entity
  - Relationship: association among entities
    - One-to-many (1:M OR 1..*)
    - Many-to-many (M:N or *..*)
    - One-to-one (1:1 OR 1..1)
  - Constraint: restriction placed on data
    - Ensures data integrity

Business Rules

- Brief, precise, and unambiguous description of a policy, procedure, or principle
  - Create and enforce actions within that organization's environment
  - Establish entities, relationships, and constraints
- Example: in our company every order is associated with one and only one salesperson

Business Rules (con't)

- Data considerations are a subset of the total set of business rules

Business Rules Examples for Purchases Process

Process Activity/Step	Intention/ Objective	Authority/Action	Access Controls	Application Controls
Place in Inventory	Place items in proper inventory locations promptly	Person placing items in inventory must be different than person ordering goods	Person placing items in inventory can't modify POs	System specifies location for inventory items
Return Defective Items	Return defective items to suppliers promptly	Manager must approve returns of defective items	Person returning items can't modify vendor/supplier information	System must provide supplier return address
Pay Supplier	Pay suppliers accurately; take discounts if advantageous	Person making payment must not be the persons who ordered, received, or accepted items.	Person making payments can't modify POs, receipt records, or acceptance records	System supplies vendor payment info & amount of payment; date defaults to current date

Discovering Business Rules

- Sources of business rules
  - Company managers
  - Policy makers
  - Department managers
  - Written documentation
  - Artifacts such as reports and forms
  - Direct interviews with end users
  - Physical or logical restrictions

Discovering Business Rules (con't)

- Reasons for identifying and documenting business rules
  - Standardize company's view of data
  - Facilitate communications tool between users and designers
  - Assist designers
    - Understand the nature, role, scope of data, and business processes
    - Develop appropriate relationship participation rules and constraints
    - Create an accurate data model

Translating Business Rules into Data Model Components

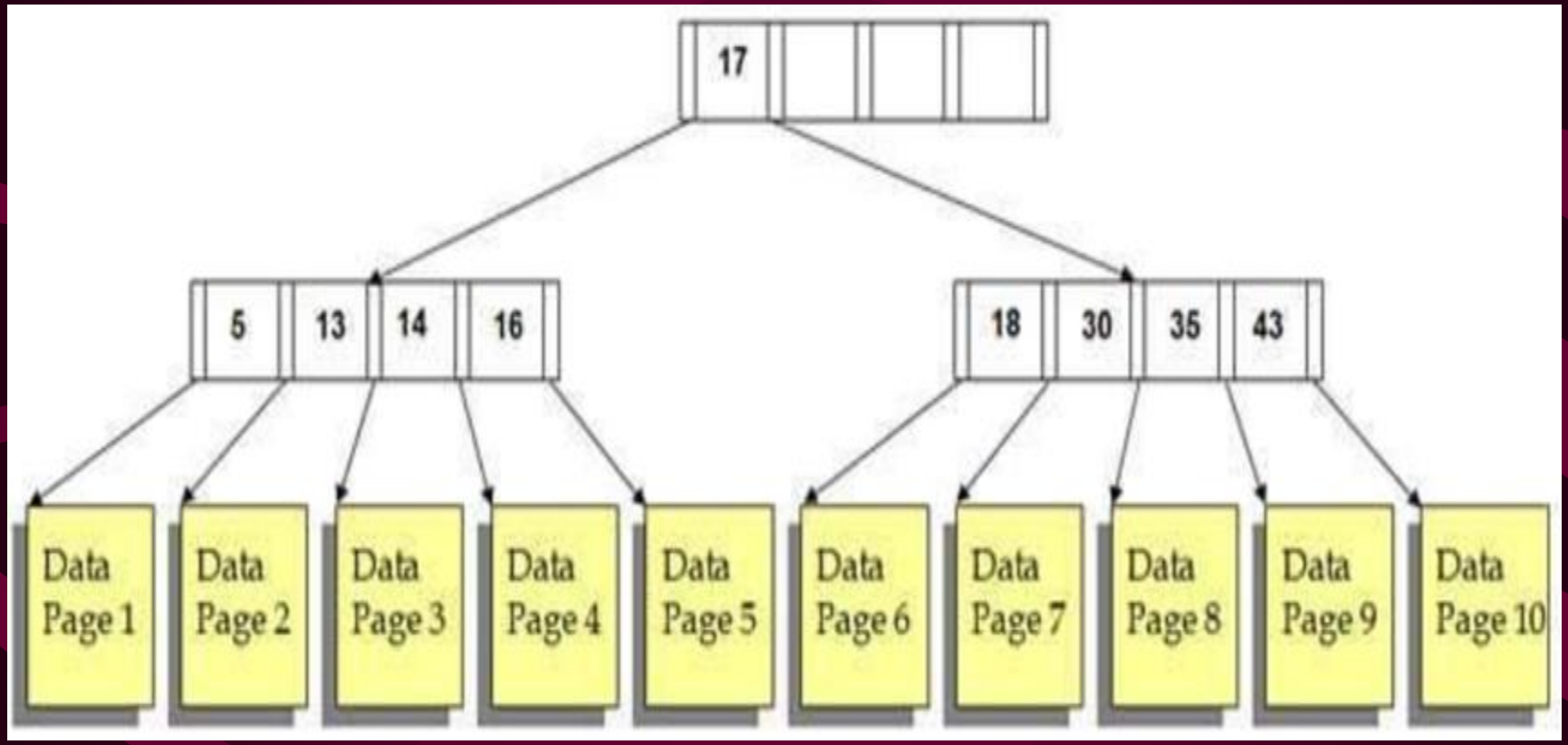
- Business rules set the stage for the proper identification of entities, attributes, relationships, and constraints
  - Nouns usually translate into entities
  - Verbs usually translate into relationships among entities
- Relationships are bidirectional

  - Questions to identify the relationship type
    - How many instances of B are related to each instance of A?
    - How many instances of A are related to each instance of B?

Database Evolution

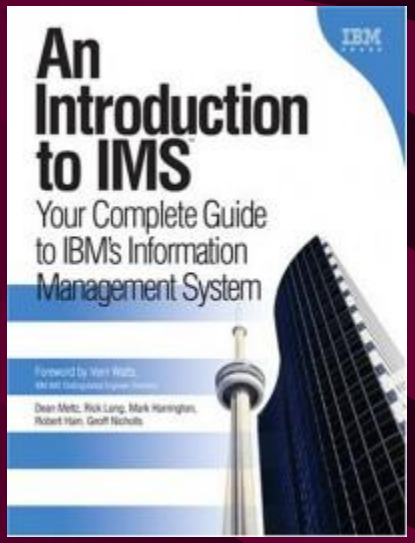
- "Flat Files" - sequential files
- Indexed Files (1950's)
  - Indexed Sequential Files ("ISAM")
  - Direct Access Method - Transform/Hashing
- Databases
  - Hierarchical (IBM's IMS – 1960's)
  - Network (CODASYL – 1970's)
  - Relational – 1980s and on
  - OO & OO/Relational
  - NoSQL & Blockchain

ISAM
(example index customer #)



Hierarchical Databases

- DL/I Standard (1960)
- IBM's IMS (Information Management System)



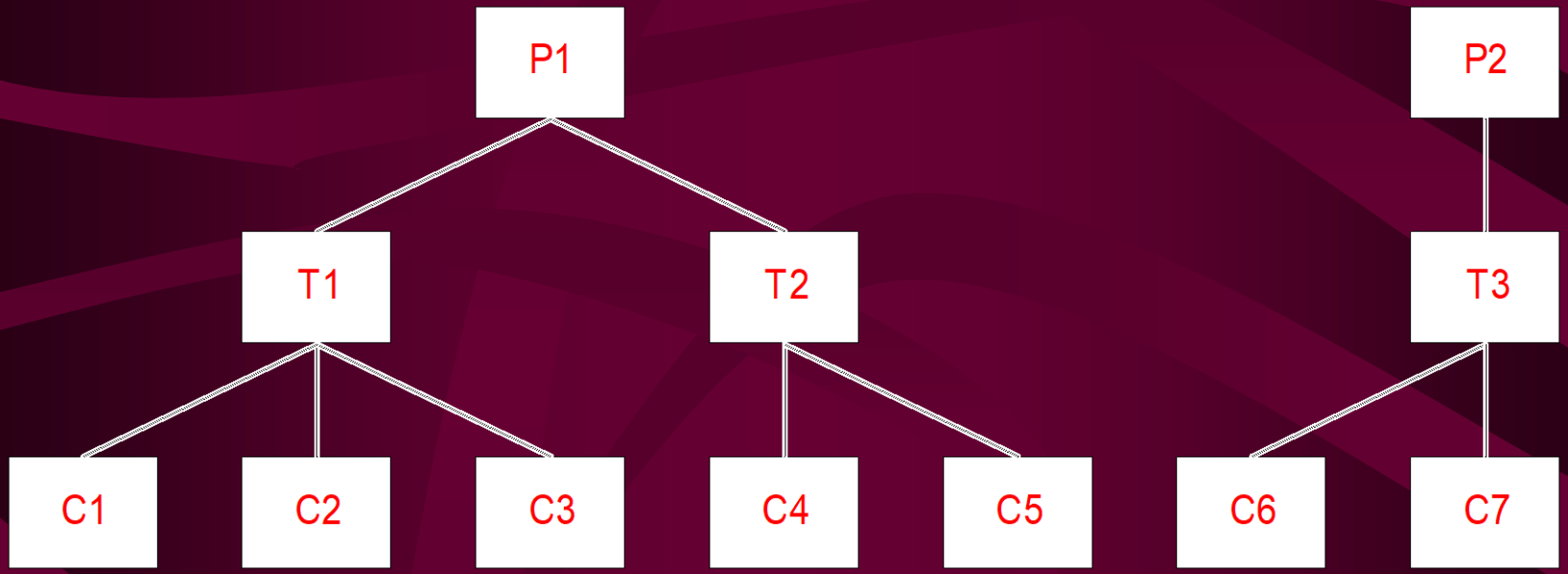
Hierarchical Databases (con't)

- A file level model
- All relationships must be transformed into trees
- Each entity class on the tree is a set (of its instances)
- Instances (of the same class) are linked to each other with pointers - *twin pointers*
- Related instances, the next level down on the tree, are linked by a *child pointer* to the first child

Case Study

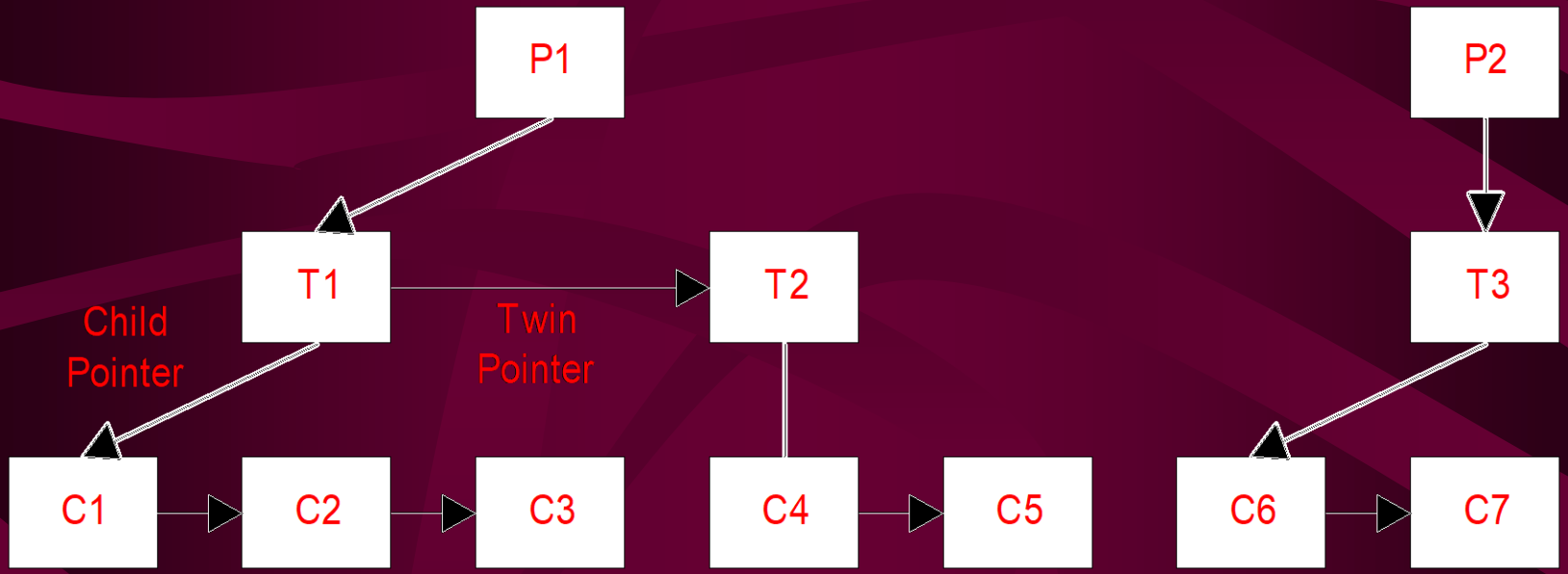
- Consider the entities:
  - PUBLISHER
  - TITLE
  - COPY
- Where a publisher prints copies of a particular title

Tree Representation (of instances)



Physical Representation [Hierarchical Databases]

[records on a disk file]

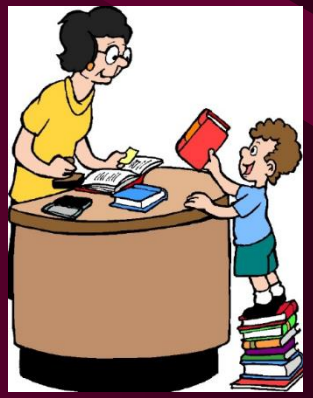


Hierarchical Databases (con't)

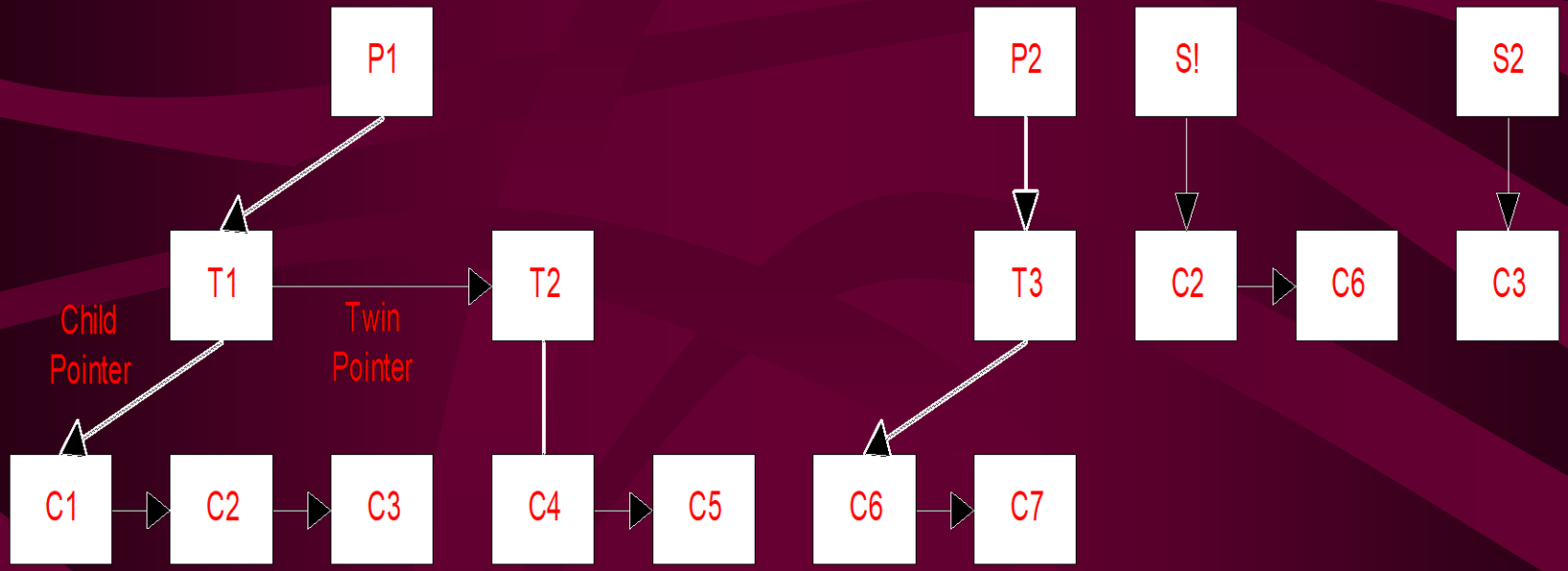
- Need to know how data is physically (or logically) organized to access it efficiently
- Therefore, a change in data organization means a change in application programs
- Duplicate data where multiple parents are involved - see following example
- Referential integrity problems (can delete a parent)
- Inverted Pointer systems (all twin pointers in parent record) - need variable length records with other associated problems

Hierarchical Databases (con't)

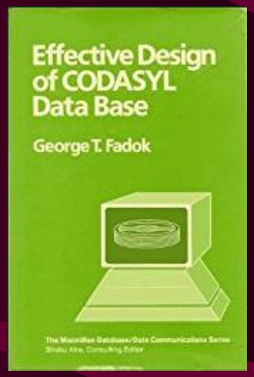
- Let's add to our case study the fact that we have students checking out these books from the library and we want to maintain information on which student currently has a particular book checked out



Duplicate COPY (C) Records

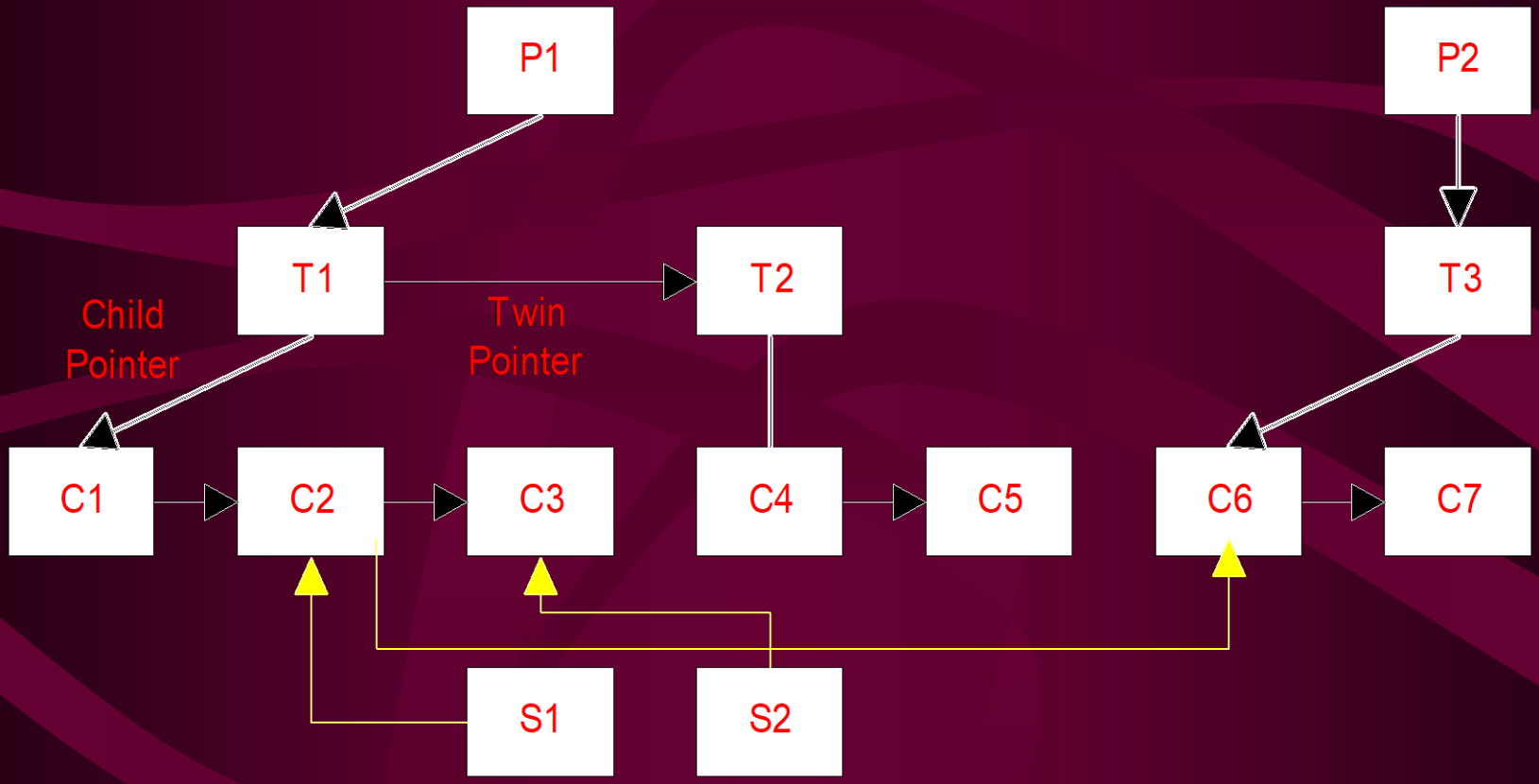


Network Databases



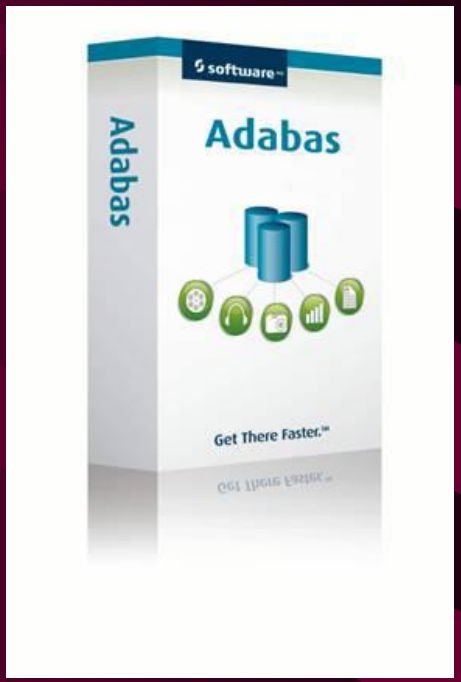
- File level models
- IDMS and TOTAL
- CODASYL [1981] standards
- All relationships must be transformed into sets
- Each entity class on the tree is a set of its instances, however it may belong to multiple sets
- Instances (of the same class) are linked to each other with pointers - *twin pointers*, *multithreaded (a set of twin pointers for each set)*
- Related instances, the next level down on the tree, are linked by a *child pointer* to the first child

For example, C2 has two threads running through it...



Network Databases (con't)

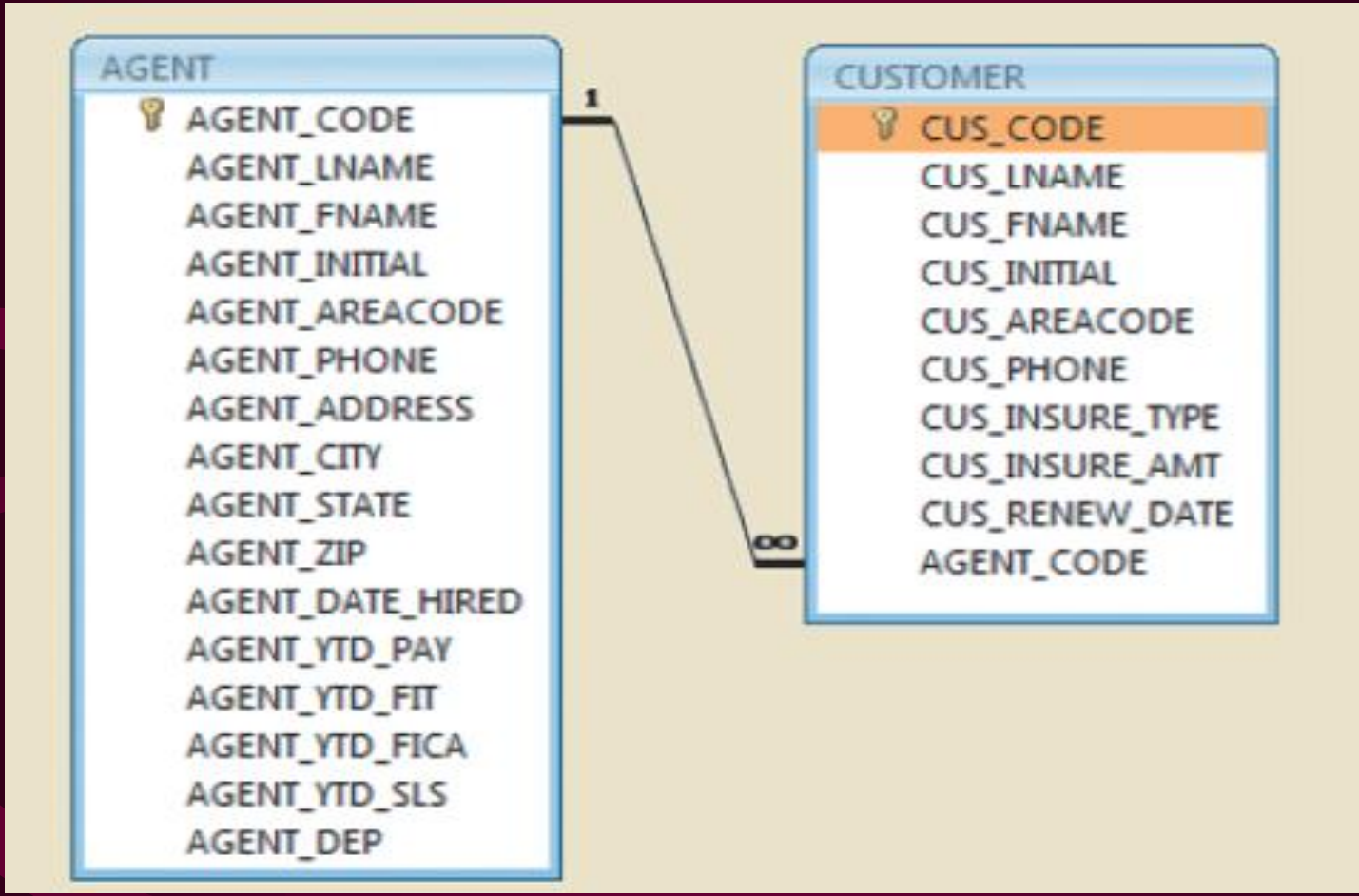
- Need to know how data is physically (or logically) organized to access it efficiently
- Therefore a change in data organization means a change in application programs
- Referential integrity problems
- Inverted pointer systems, such as ADABAS or DATACOM, (all twin pointers in parent record) - need variable length records with other associated problems



The Relational Model

- Based on a relation (i.e., mathematical table) matrix composed of intersecting tuples (rows) and attributes (columns) – table level model
- Describes a precise set of data manipulation constructs
- Relational database management system (RDBMS)
  - Performs basic functions provided by the hierarchical and network DBMS systems
  - Makes the relational data model easier to understand and implement
  - Hides the complexities of the relational model from the user

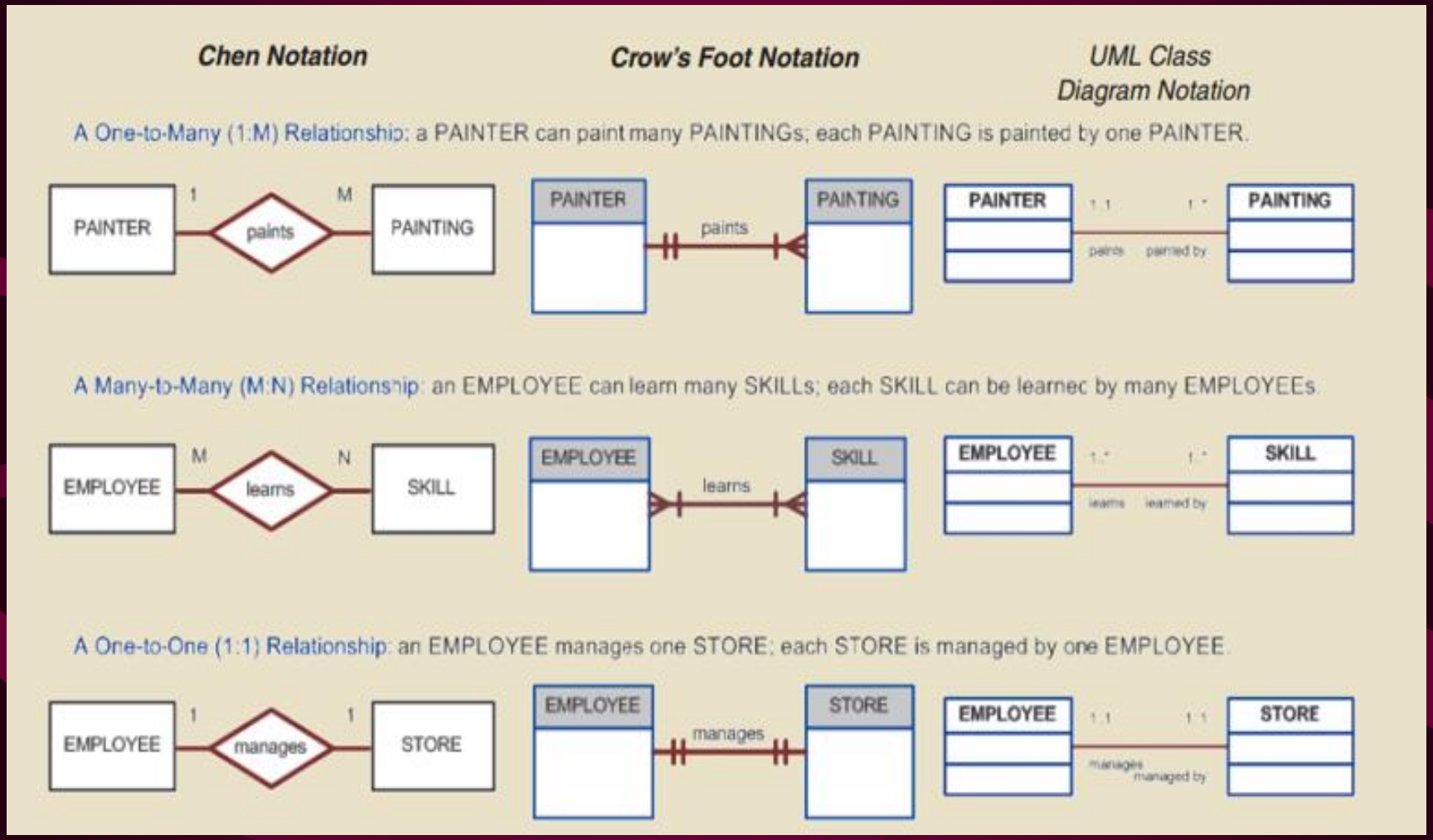
The Relational Model (con't)



The Entity Relationship Model

- An entity level model
- Graphical representation of entities and their relationships in a database structure
  - Entity relationship diagram (ERD): uses graphic representations to model database components
  - Attributes: describe particular characteristics
  - Connectivity: term used to label the relationship types

The Entity Relationship Model (con't)



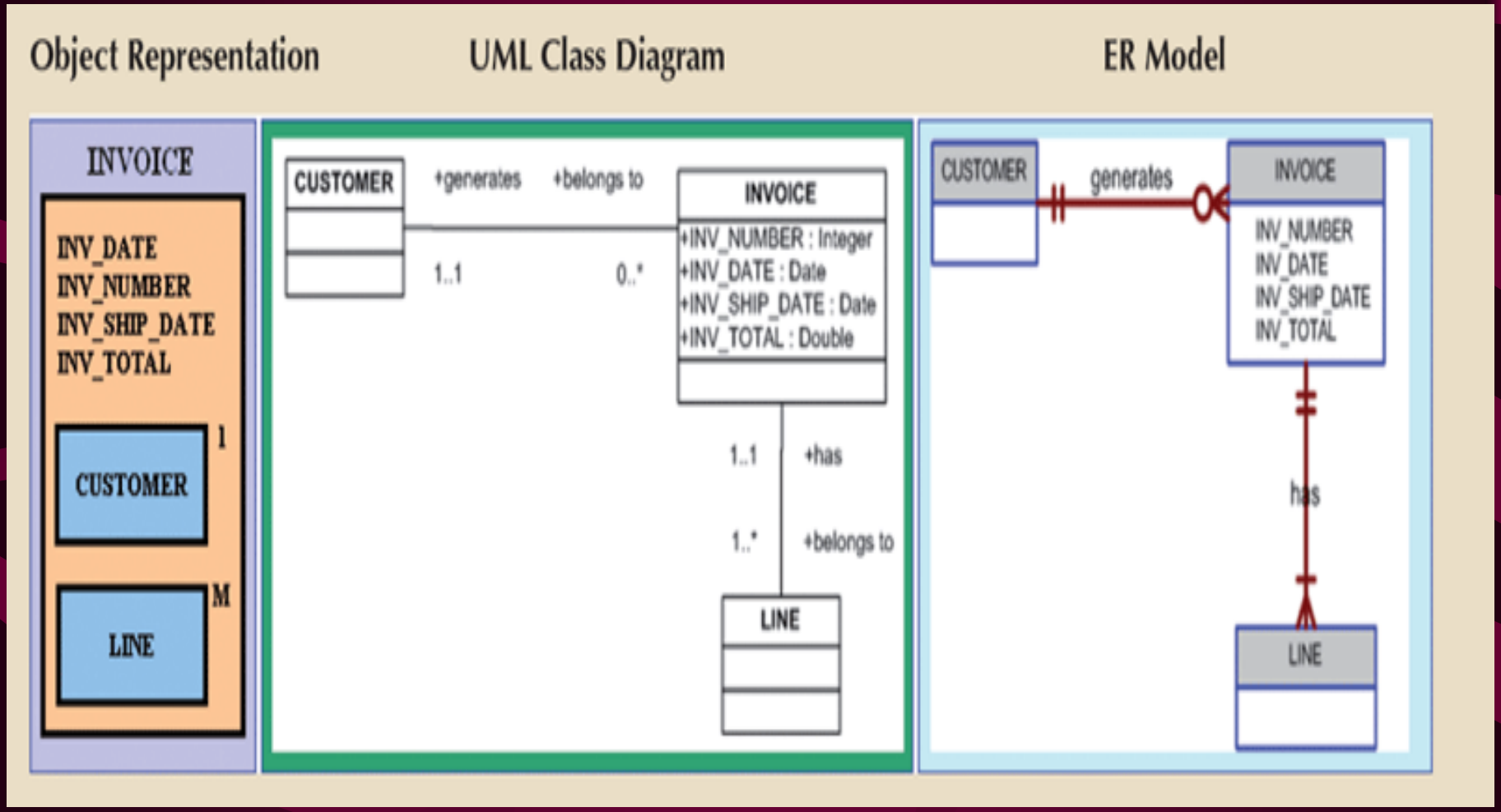
The Object-Oriented Data Model

- Both data and its relationships are contained in a single structure known as an object
  - Object-oriented database management system(OODBMS): based on OODM
- Object: contains data and their relationships with operations that are performed on it
  - Basic building block for autonomous structures
  - Abstraction of real-world entity
- Attribute: describes the properties of an object

The Object-Oriented Data Model (con't)

- Class: collection of similar objects with shared structure and behavior organized in a class hierarchy
- Class hierarchy: resembles a tree in which each class has only one parent
  - Inheritance: object inherits methods and attributes of classes above it
- Unified Modeling Language (UML): describes sets of diagrams and symbols to graphically model a system

Object-Oriented Model vs E-R Model



Extended Relational Data Model (ERDM)

- Extended relational data model (ERDM)
  - Supports some OO features, extensible data types based on classes, and inheritance
    - Object/relational database management system (O/R DBMS): based on ERDM

Big Data and NoSQL (not only SQL)

- Goals of Big Data
  - Find new and better ways to manage large amounts of data such as from web ranking and searching, social media, and sensor-generated data
  - Provide high performance at a reasonable cost
- Characteristics of Big Data (3 V's)
  - Volume
  - Velocity
  - Variety

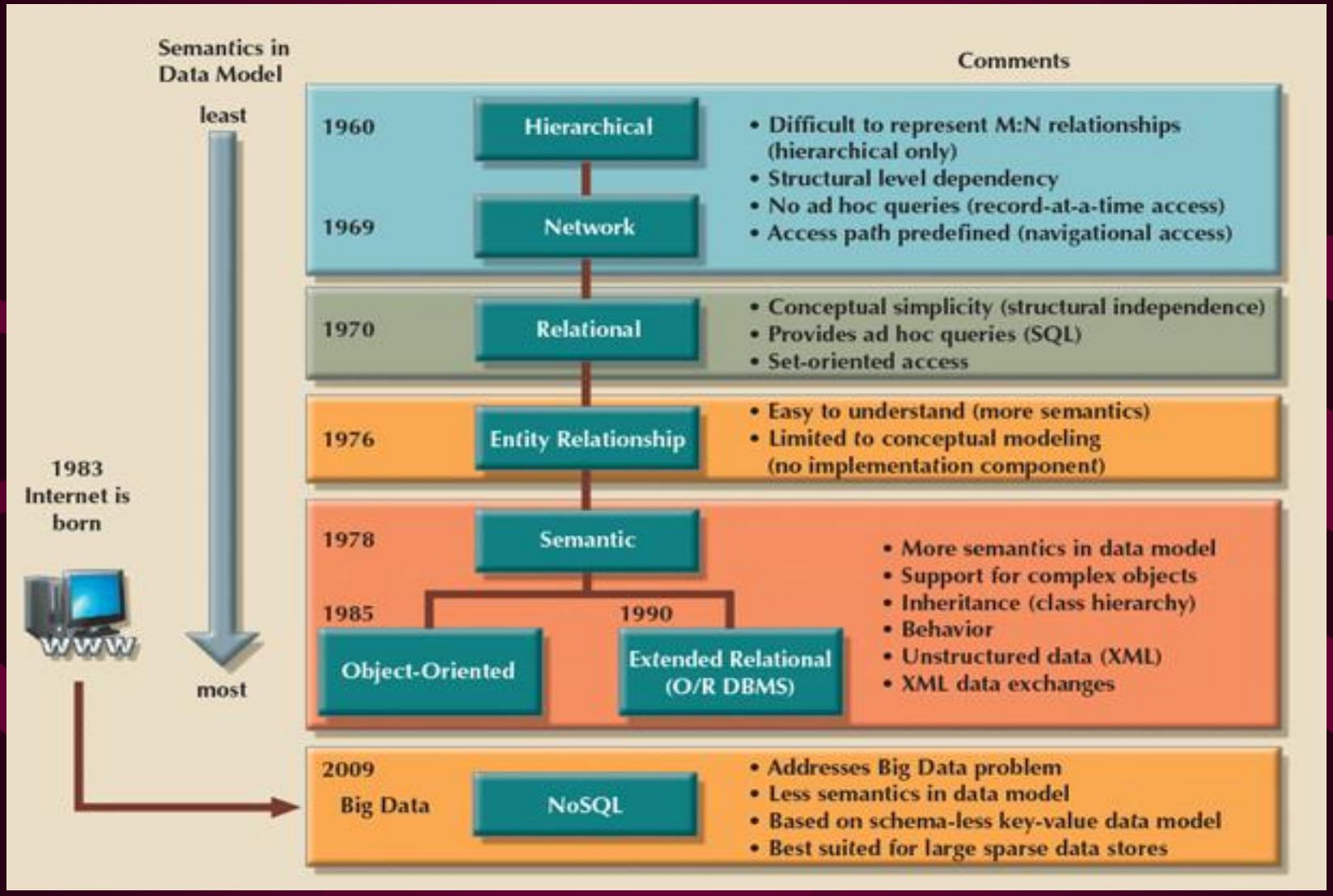
Big Data and NoSQL (con’t)

- Challenges of Big Data
  - Volume doesn’t allow usage of conventional structures
  - Expensive
  - OLAP tools proved inconsistent dealing with unstructured data
- New technologies of Big Data
  - Hadoop
  - Hadoop Distributed File System (HDFS)
  - MapReduce
  - Blockchain

Big Data and NoSQL (con't)

- NoSQL databases
  - Not based on the relational model
  - Support distributed database architectures
  - Provide high scalability, high availability, and fault tolerance
  - Support large amounts of sparse data
  - Geared toward performance rather than transaction consistency
  - Provides a broad umbrella for data storage and manipulation

Data Model Summary



Hierarchical Model Pros and Cons

- Advantages
  - Promotes data sharing
  - Parent/child relationship promotes conceptual simplicity and data integrity
  - Database security is provided and enforced by DBMS
  - Efficient with 1:M relationships
- Disadvantages
  - Requires knowledge of physical data storage characteristics
  - Navigational system requires knowledge of hierarchical path
  - Changes in structure require changes in all application programs
  - Implementation limitations
  - No data definition
  - Lack of standards

Network Model Pros and Cons

- Advantages
  - Conceptual simplicity
  - Handles more relationship types
  - Data access is flexible
  - Data owner/member relationship promotes data integrity
  - Conformance to standards
  - Includes data definition language (DDL) and data manipulation language (DML)
- Disadvantages
  - System complexity limits efficiency
  - Navigational system yields complex implementation, application development, and management
  - Structural changes require changes in all application programs

Relational Model Pros and Cons

- Advantages
  - Structural independence is promoted using independent tables
  - Tabular view improves conceptual simplicity
  - Ad hoc query capability is based on SQL
  - Isolates the end user from physical-level details
  - Improves implementation and management simplicity
- Disadvantages
  - Requires substantial hardware and system software overhead
  - Conceptual simplicity gives untrained people the tools to use a good system poorly
  - Not suited to “big data”

Entity Relationship Model

- Advantages
  - Visual modeling yields conceptual simplicity
  - Visual representation makes it an effective communication tool
  - Is integrated with the dominant relational model
- Disadvantages
  - Must be converted to a table level model to be implemented
  - Limited constraint representation
  - Limited relationship representation
  - No data manipulation language
  - Loss of information content occurs when attributes are removed from entities to avoid crowded displays

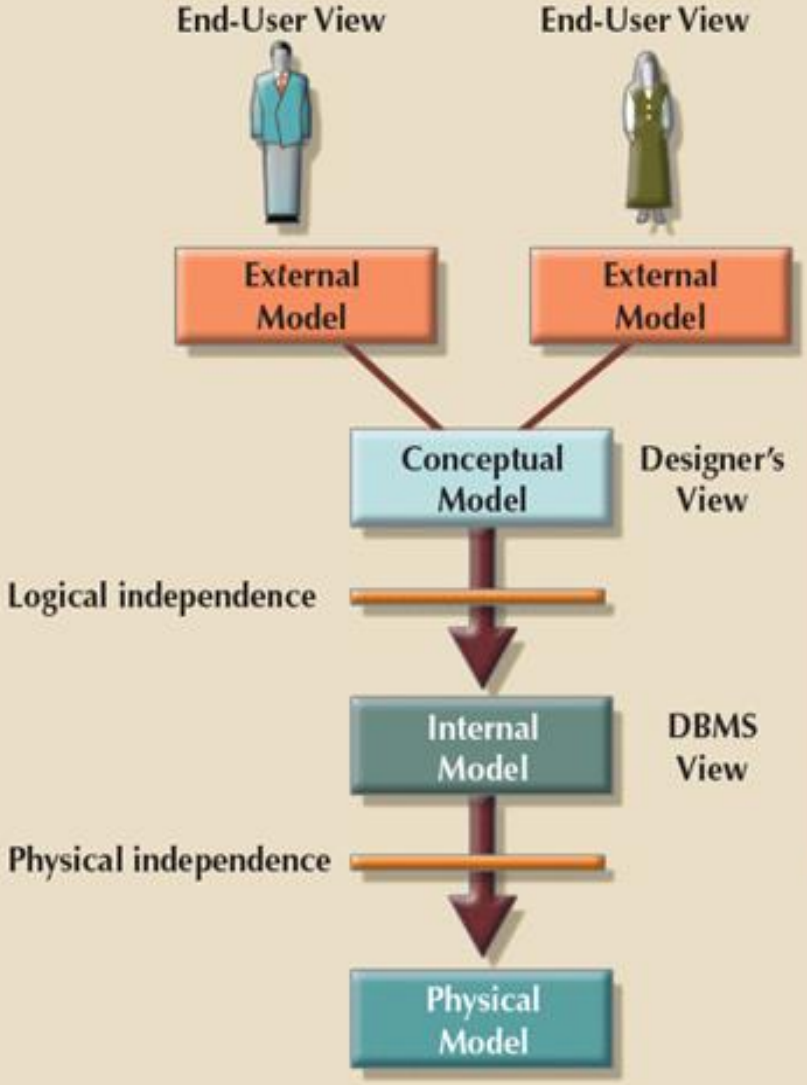
Object-Oriented Model Pros and Cons

- Advantages
  - Semantic content is added
  - Visual representation includes semantic content
  - Inheritance promotes data integrity
- Disadvantages
  - Slow development of standards caused vendors to supply their own enhancements
  - Complex navigational system
  - Learning curve is steep
  - High system overhead slows transactions

NoSQL Pros and Cons

- Advantages
  - High scalability, availability, and fault tolerance are provided
  - Uses low-cost commodity hardware
  - Supports Big Data
  - Key-value model improves storage efficiency
- Disadvantages
  - Complex programming is required
  - There is no relationship support
  - There is no transaction integrity support
  - In terms of data consistency, it provides an eventually consistent model

Degrees of Data Abstraction

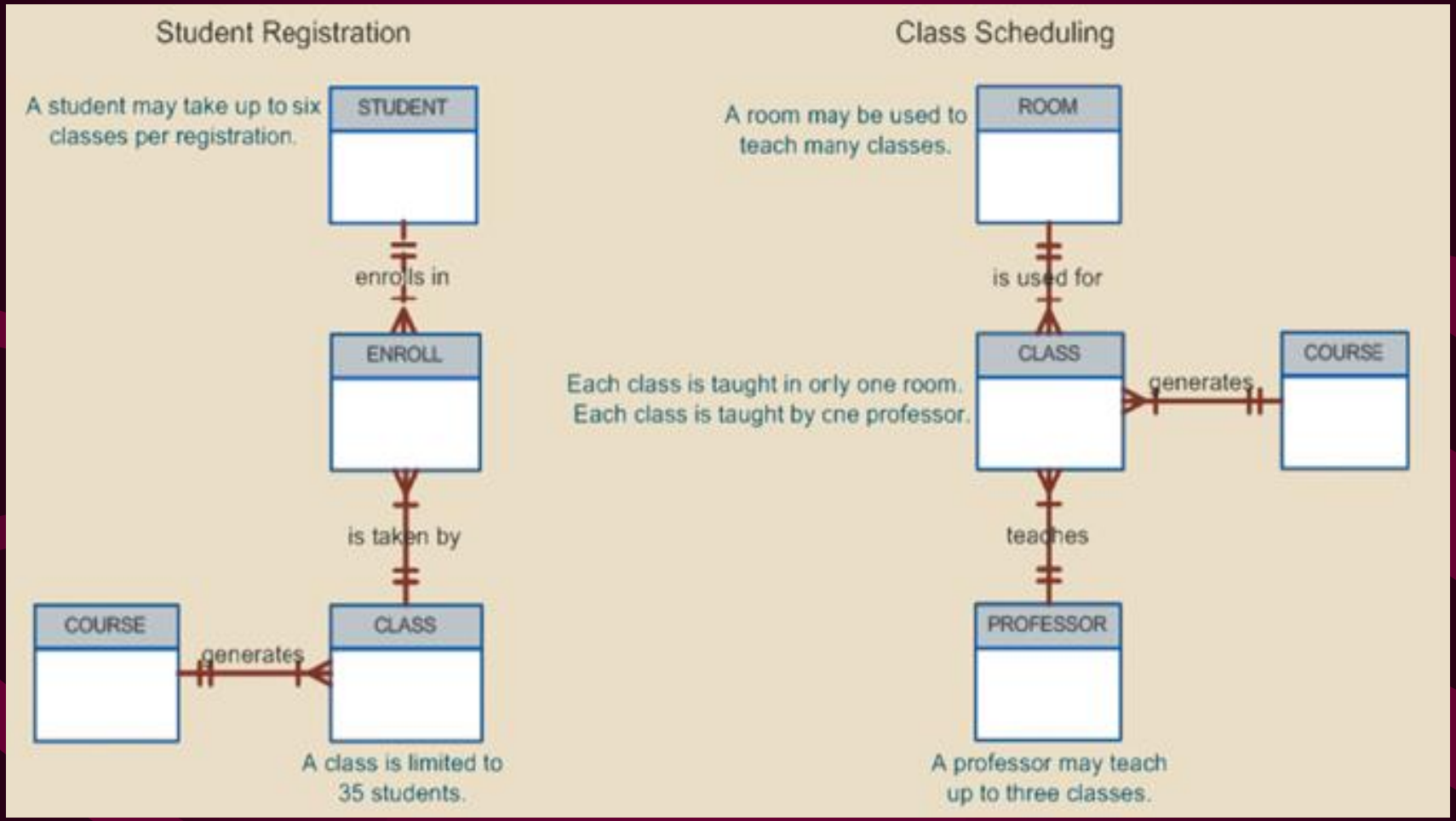


Degree of Abstraction		Characteristics
High	ER Object-Oriented	Hardware-independent Software-independent
Medium	Relational	Hardware-independent Software-dependent
Low	Network Hierarchical	Hardware-dependent Software-dependent

The External Model

- End users' view of the data environment
  - People who use the application programs to manipulate the data and generate information
- Different departments may have different views with some similar entities
- ER diagrams are typically used to represent the external views
  - External schema: specific representation of an external view

The External Model (con't)



Note: some entities are in one model but not the other.

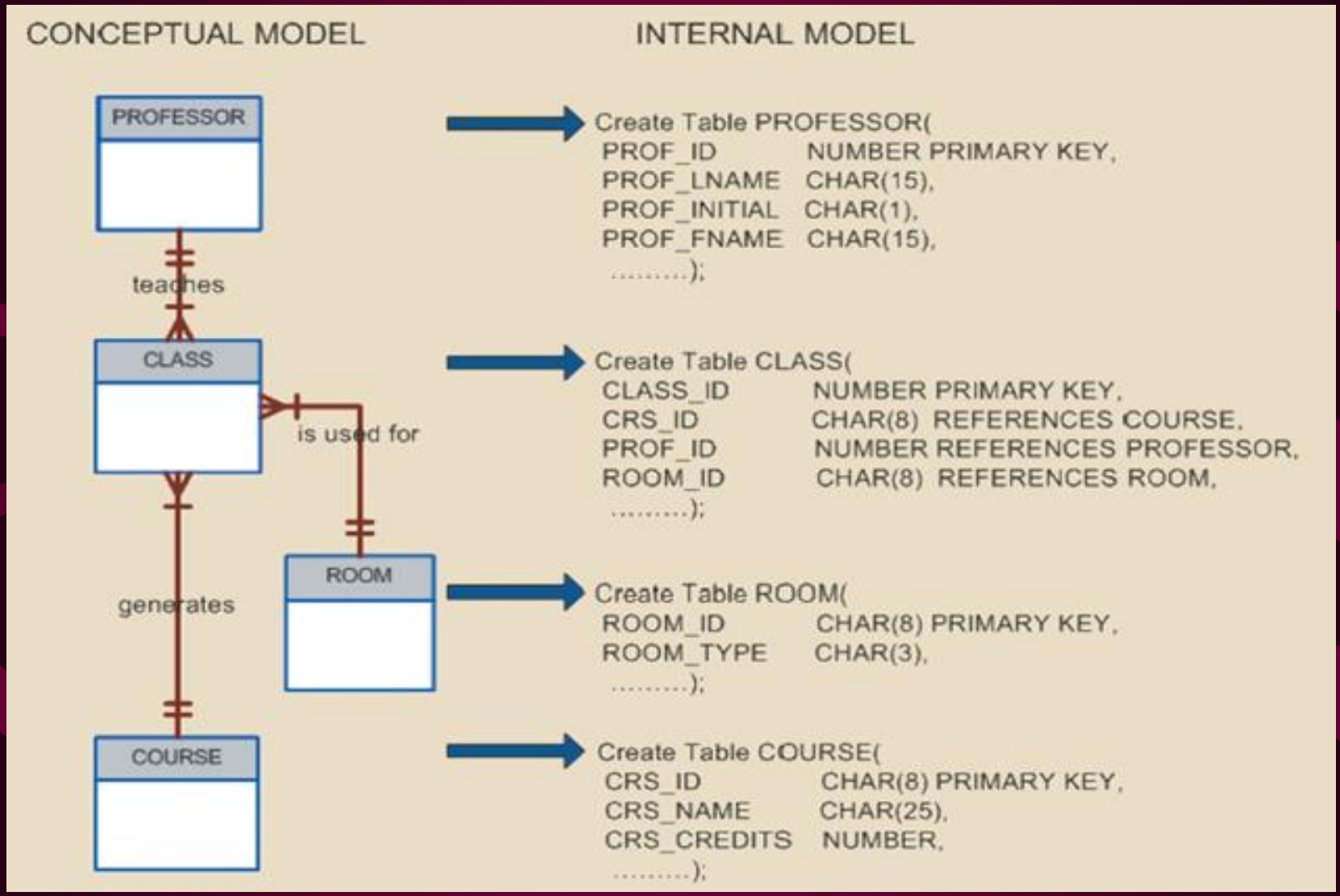
The Conceptual Model

- Represents a global view of the entire database by the entire organization
  - Integrates all external views (ER Diagrams)
  - Conceptual schema: basis for the identification and high-level description of the main data objects
  - Logical design: task of creating a conceptual data model
- Conceptual model advantages
  - Macro-level view of data environment
  - Software and hardware independent

The Internal Model

- Representing database as seen by the DBMS mapping conceptual model to the DBMS (such as the relational tables)
  - Internal schema: specific representation of an internal model, using the database constructs supported by the chosen database
  - Logical independence: changing internal model without affecting the conceptual model
  - Hardware independent: unaffected by the type of computer on which the software is installed

Conceptual Model and Internal Model



Internal Model Representations

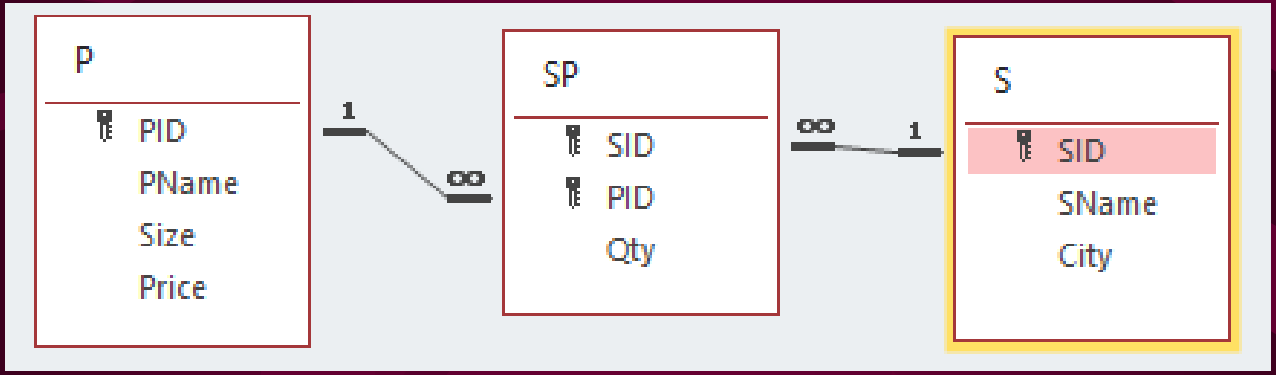
- Table notation:
  - Table name, attributes, primary key (underline), foreign keys (italicize)
    - SP (*SID, PID*, QTY)
- SQL notation:
  - CREATE TABLE SP (
  - SID Number,
  - PID Number,
  - QTY Number,
  - PRIMARY KEY (SID, PID),
  - FOREIGN KEY SID REFERENCES S,
  - FOREIGN KEY PID REFERENCES P);

The Physical Model

- Operates at lowest level of abstraction
  - Describes the way data are saved on storage media such as magnetic, solid state, or optical media
- Requires the definition of physical storage and data access methods
  - Software and hardware dependent
- Relational model aimed at logical level
  - Does not require physical-level details
- Physical independence: changes in physical model do not affect internal model

Example Physical Model

- Microsoft Access physical model:



Database Models

Model	Degree of Abstraction	Focus	Independent of
External	High	End-user views	Hardware and software
Conceptual	Medium-High	Global view of data (database model independent)	Hardware and software
Internal	Medium-Low	Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

Database Models and Levels

- Entity Level
- External & Conceptual models
  - ER diagram
- Table Level
- File Level
- Internal model
- Physical Model

Review Exercise

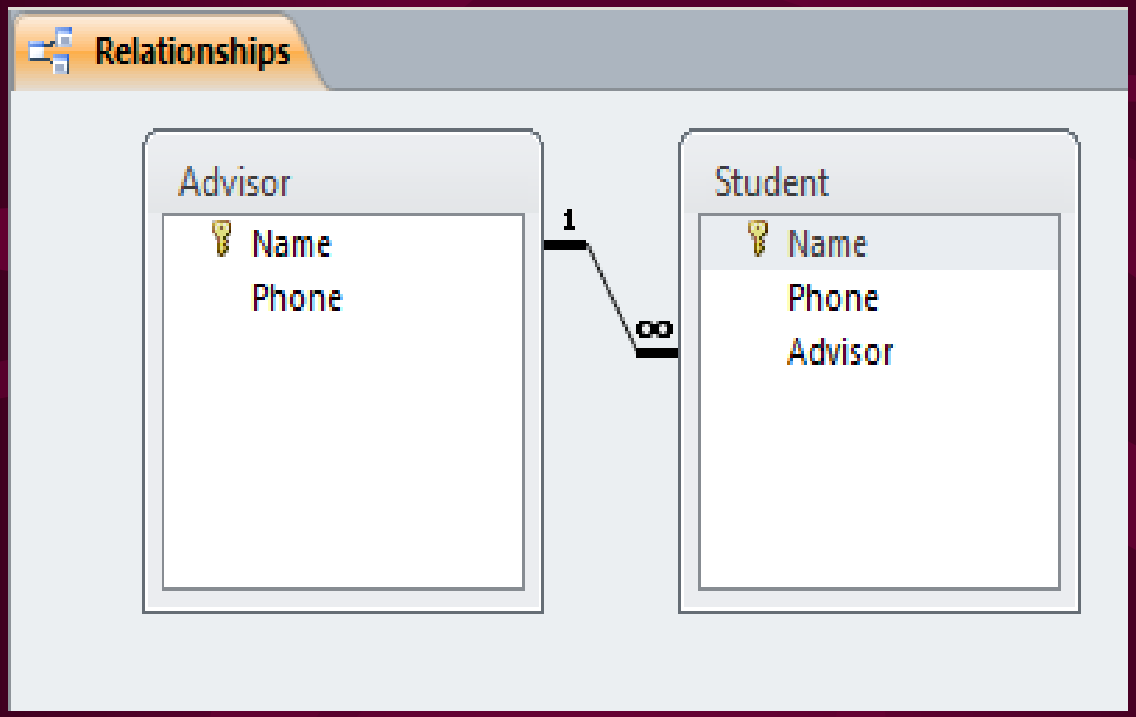
- Create an Access model (table level model) for the student/advisor data
  - Names are primary keys
  - Set up referential integrity
  - Enter test data

Student Name	Student Number	Advisor
Baker, Rex	232-8897	Parks
Charles, Mary	232-0099	Parks
Johnson, Beth	232-4487	Jones
Scott, Glenn	232-4444	Parks
zylog, Frita	232-5588	Jones

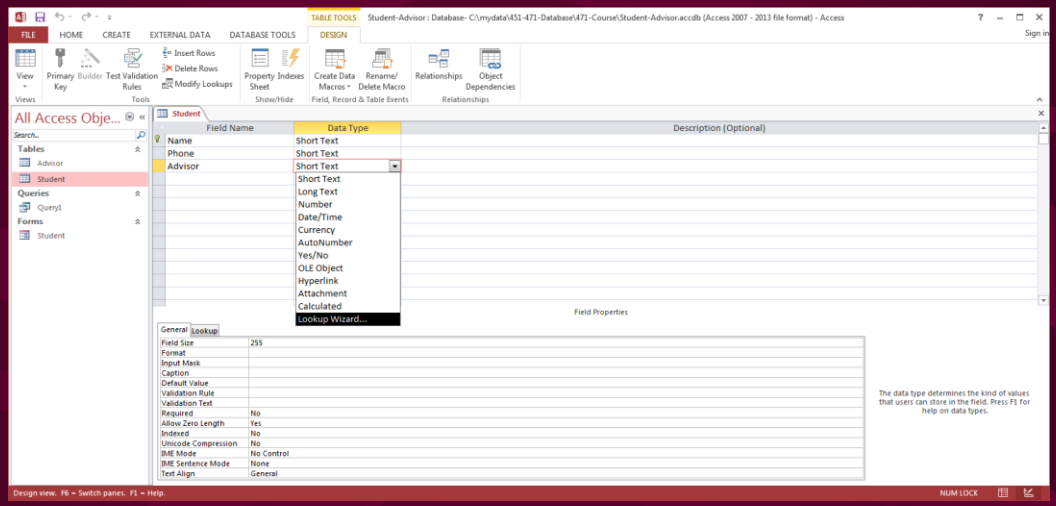
Advisor Name	Advisor Phone
Parks	236-0098
Jones	236-0110

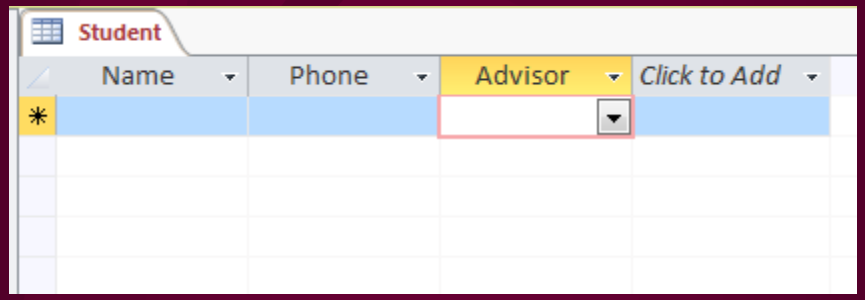
Don’t look ahead !

Access Model



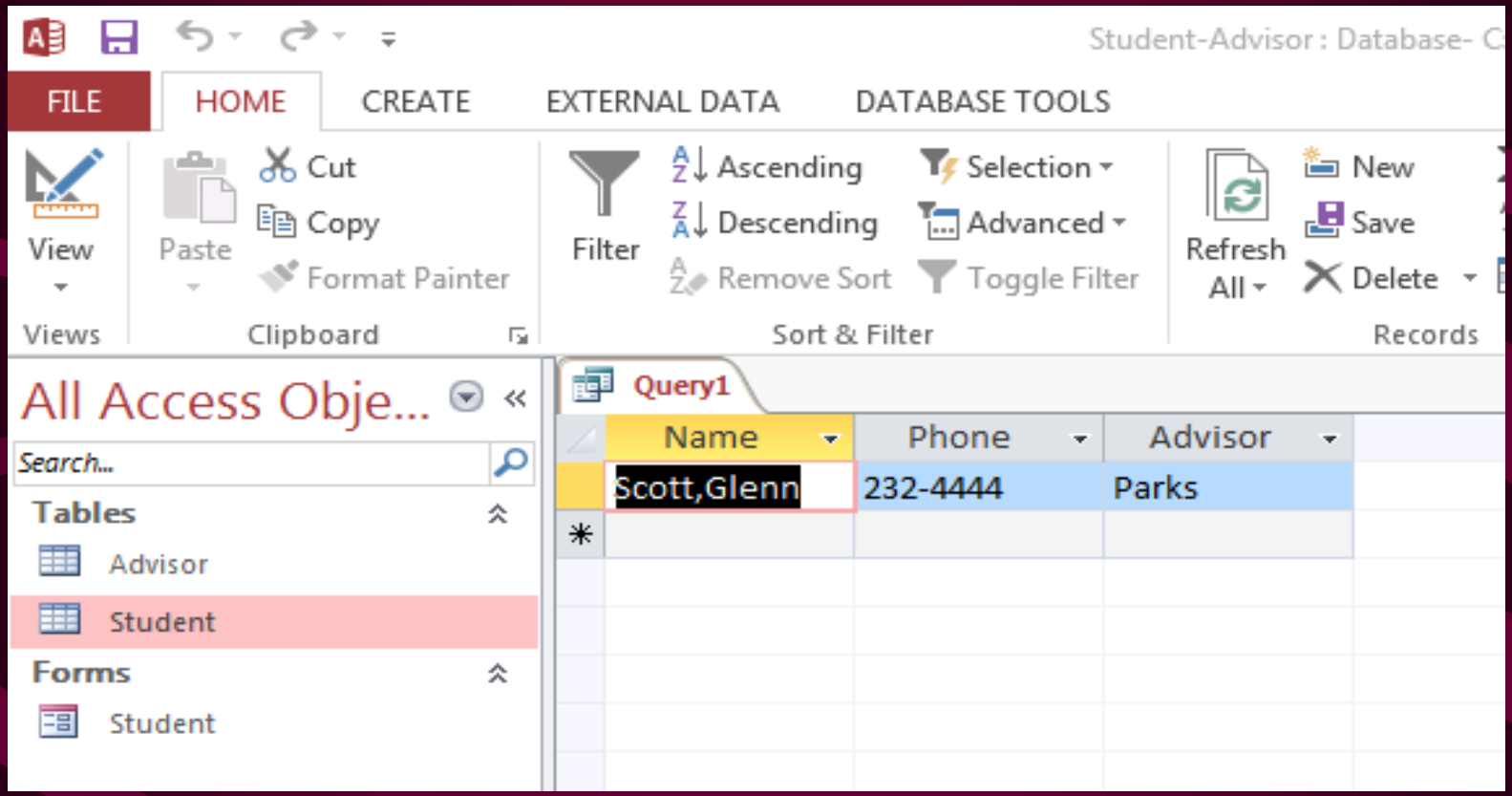
- Now make the advisor field in the student table a “lookup” field



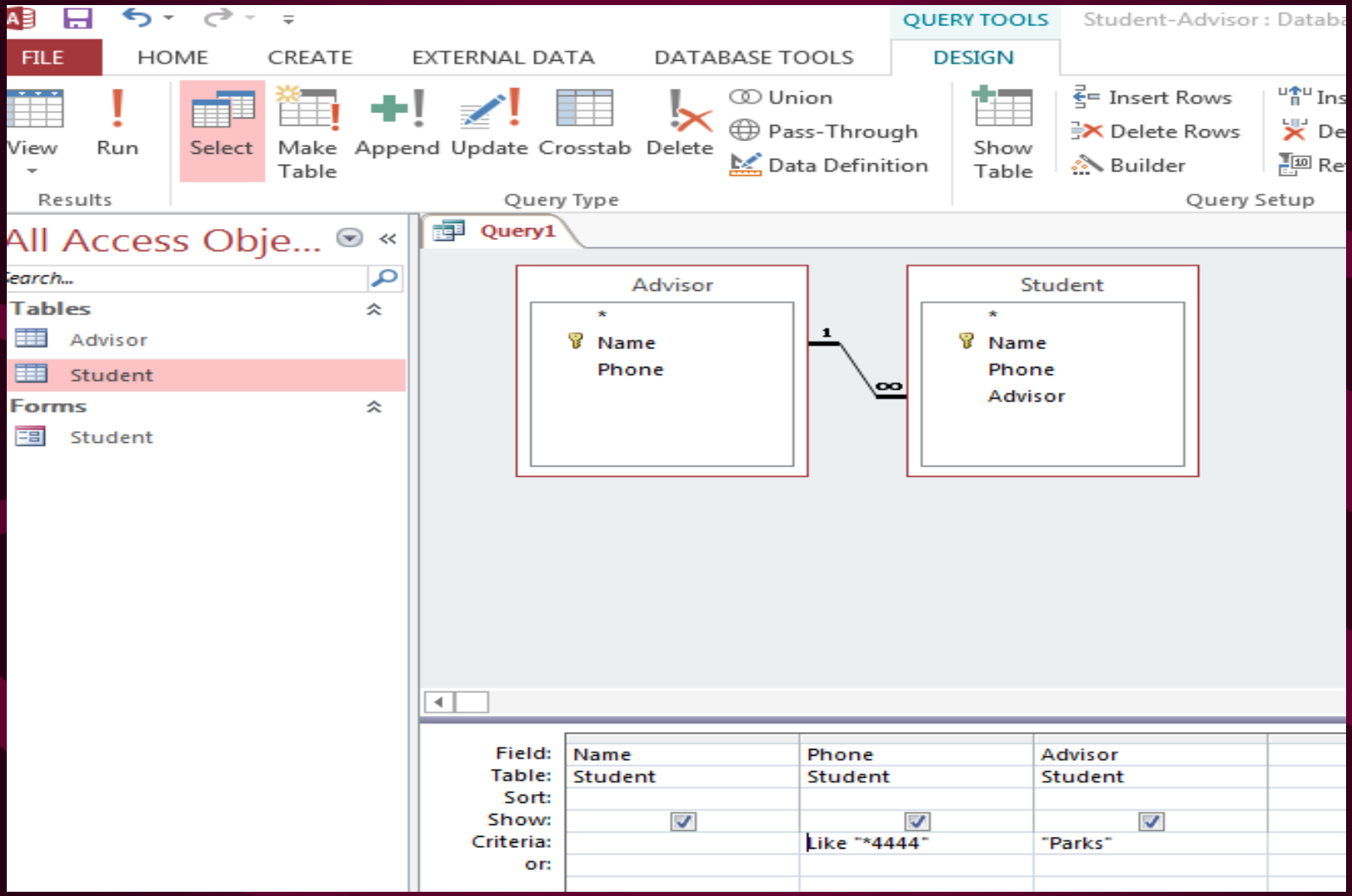


- Create a query to show only students who have Parks as an advisor and a phone extension of 4444

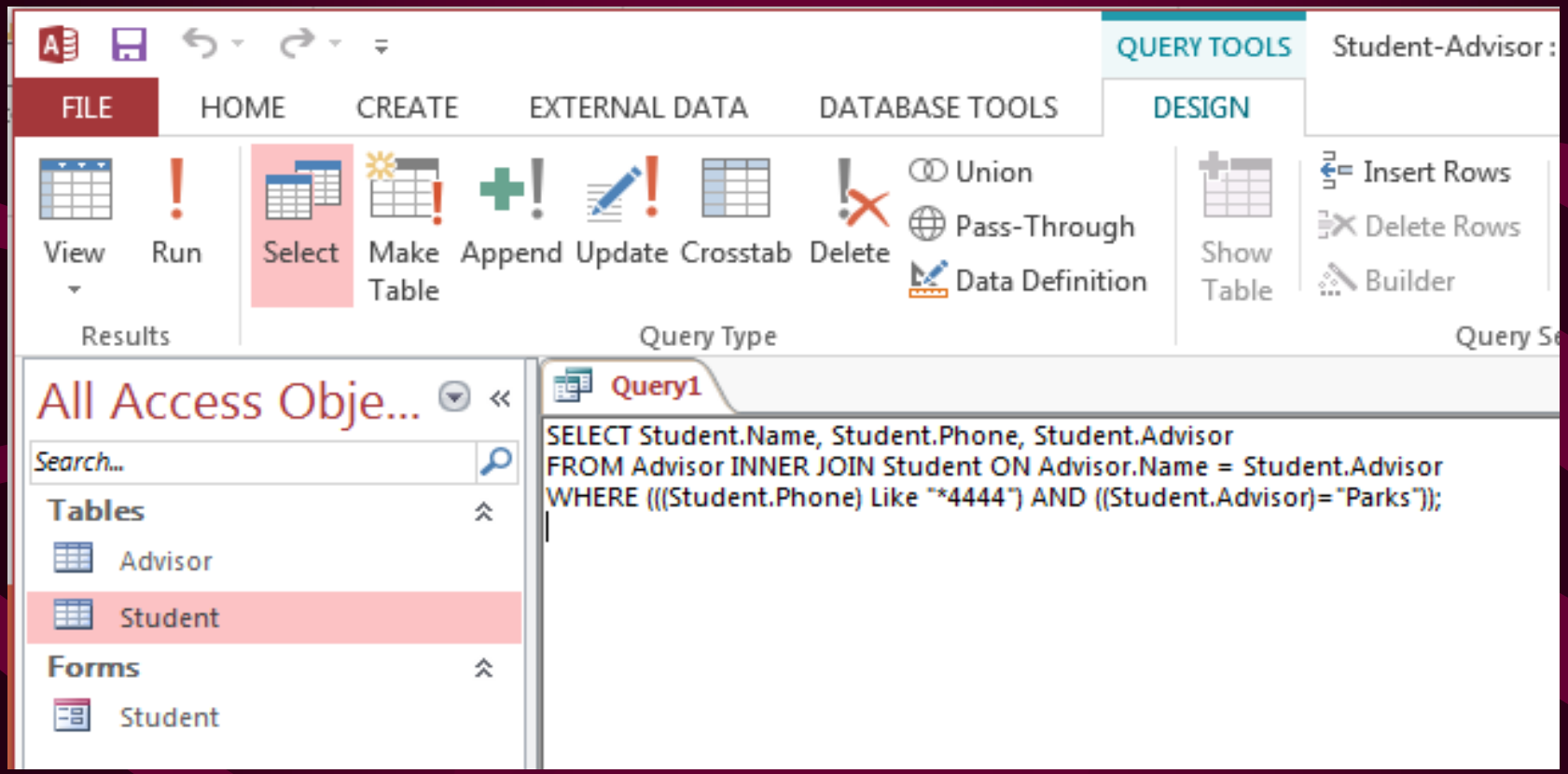
Don’t look ahead !



Query Grid Design View



SQL View



References

- The Data Model Resource Book: A Library of Universal Data Models for All Enterprises by Len Silverston
- Data Model Patterns: Conventions of Thought (Dorset House eBooks) by David Hay
- Access For Dummies by Laurie A. Ulrich and Ken Cook

Homework

- Read textbook Chapter 2
- Review Questions 1 thru 12
- Textbook Problems 1 thru 3
- Consider - for your project, for what things do you have to maintain information about (external model)