Chapter 2: Data Models

Data Model

- It is a collection of conceptual tools for describing data, data relationship, data semantics and consistency constraints.
- It provides a way to describe the design of a database at physical , logical and view level.
- Is a Communications tool to facilitate interaction among the designer, the applications programmer, and the end user
- Relatively simple representation, usually graphical, of complex real-world data structures
- Good database design uses an appropriate data model as its foundation

Importance of Data Modeling

- End-users have different views and needs for data
- Data model organizes data for various users

Data Model Basic Building Blocks

- Entity is anything about which data are to be collected and stored
- Attribute is a characteristic of an entity
- Relationship describes an association among (two or more) entities
  - **One-to-many (1:M) relationship**
  - **Many-to-many (M:N or M:M) relationship**
  - **One-to-one (1:1) relationship**

Business Rules

- Brief, precise, and unambiguous description of a policy, procedure, or principle within a specific organization's environment
- Apply to *any organization that stores and* uses data to generate information
- Description of operations that help to create and enforce actions within that organization's environment
- Must be rendered in writing
- Must be kept up to date
- Sometimes are external to the organization
- Must be easy to understand and widely disseminated
- Describe characteristics of the data *as viewed by the company*

Data models categorization

- American National Standards Institute/Standards Planning and Requirements Committee (ANSI/SPARC)

–Classified data models according to their degree of abstraction (1970s):

- Conceptual Model
- External Model
- Internal Model
- Physical Model

FIGURE 2.10 DATA ABSTRACTION LEVELS

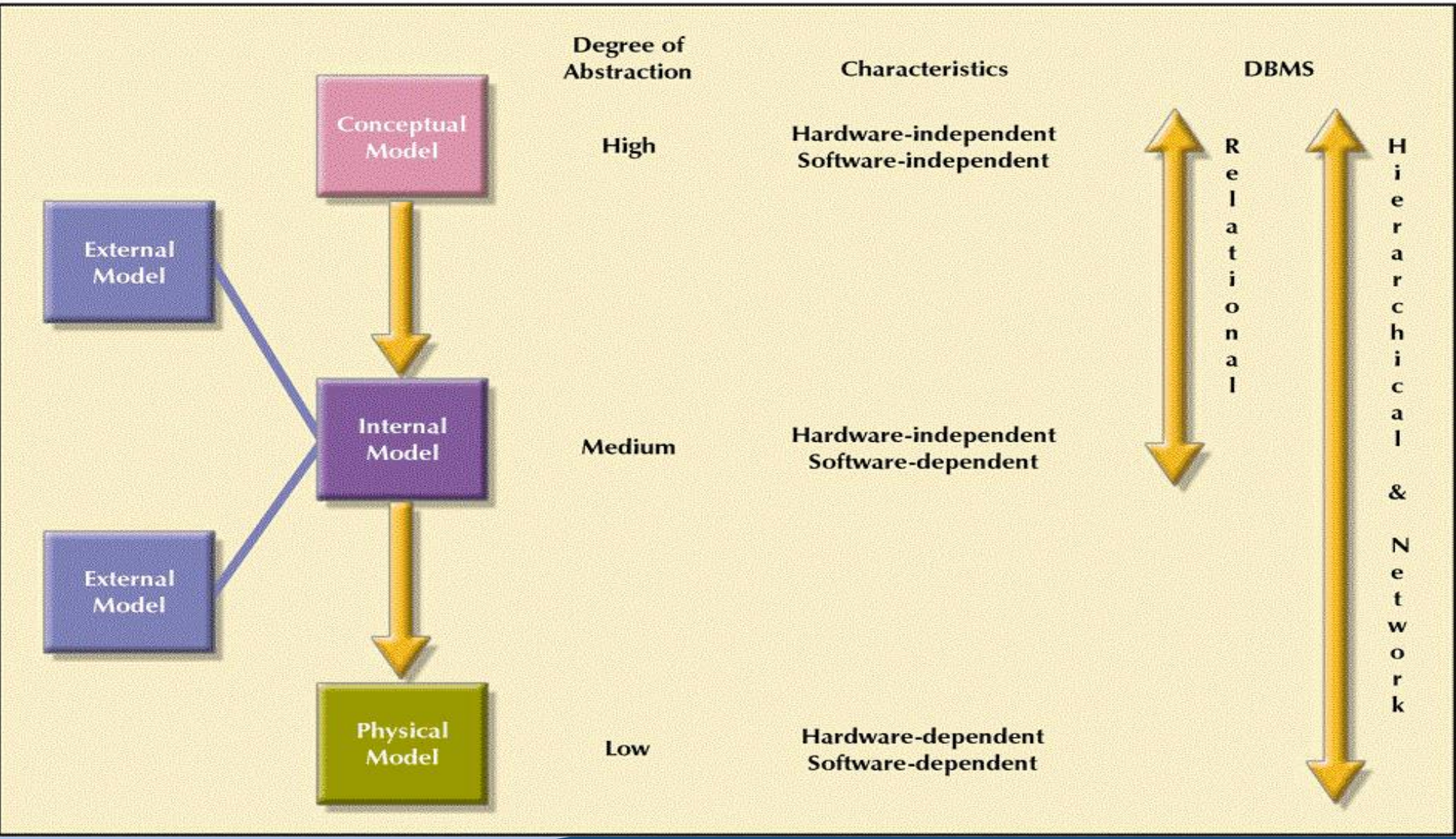


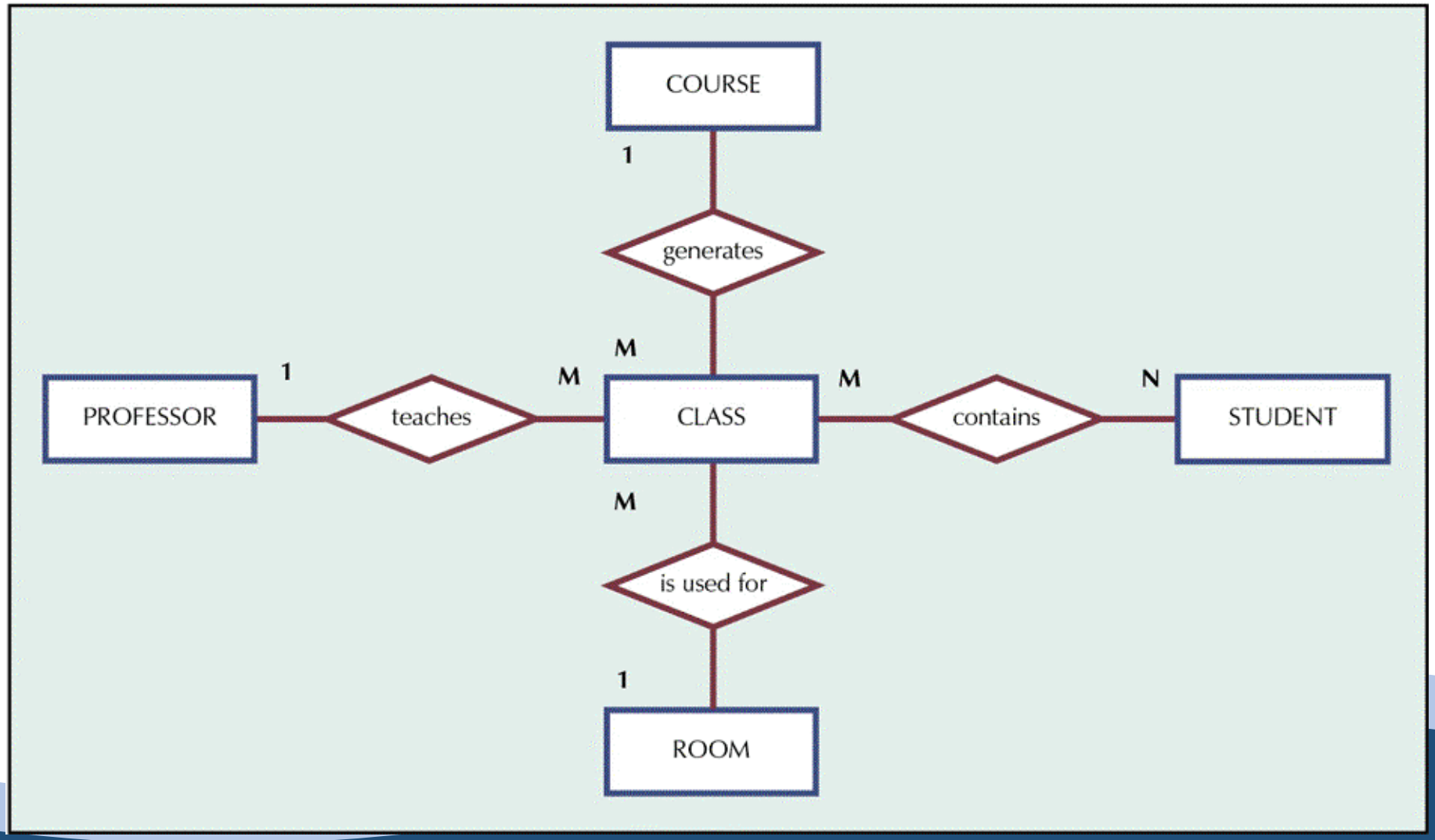
TABLE 2.2 LEVELS OF DATA ABSTRACTION

MODEL	DEGREE OF ABSTRACTION	DATA MODEL	FOCUS	INDEPENDENT OF
Conceptual	High	Entity	Global	Hardware and software
External		ER components	Subset	Hardware
Internal		Relational and others	Global	Hardware
Physical	Low	Physical storage methods	N/A	Neither hardware nor software

Conceptual Model

- Represents global view of the database
- Enterprise-wide representation of data as viewed by high-level managers
- Basis for identification and description of main data objects, avoiding details
- Most widely used conceptual model is the
- entity relationship (ER) model

FIGURE 2.12 A CONCEPTUAL MODEL FOR TINY COLLEGE



Internal Model (Logical Model)

- Representation of the database as “seen” by the DBMS
- Adapts the conceptual model to the DBMS
- Software dependent
- Hardware independent

External Model

- End users' view of the data environment
- Requires that the modeler subdivide set of requirements and constraints into functional modules that can be examined within the framework of their external models
- Good design should:

–Consider such relationships between views

–Provide programmers with a set of restrictions that govern common entities

- Use of database subsets makes application program development much simpler
- Facilitates designer's task by making it easier to identify specific data required to support each business unit's operations
- Provides feedback about the conceptual model's adequacy
- Creation of external models helps to ensure security constraints in the database design

FIGURE 2.13 A DIVISION OF AN INTERNAL MODEL INTO EXTERNAL MODELS

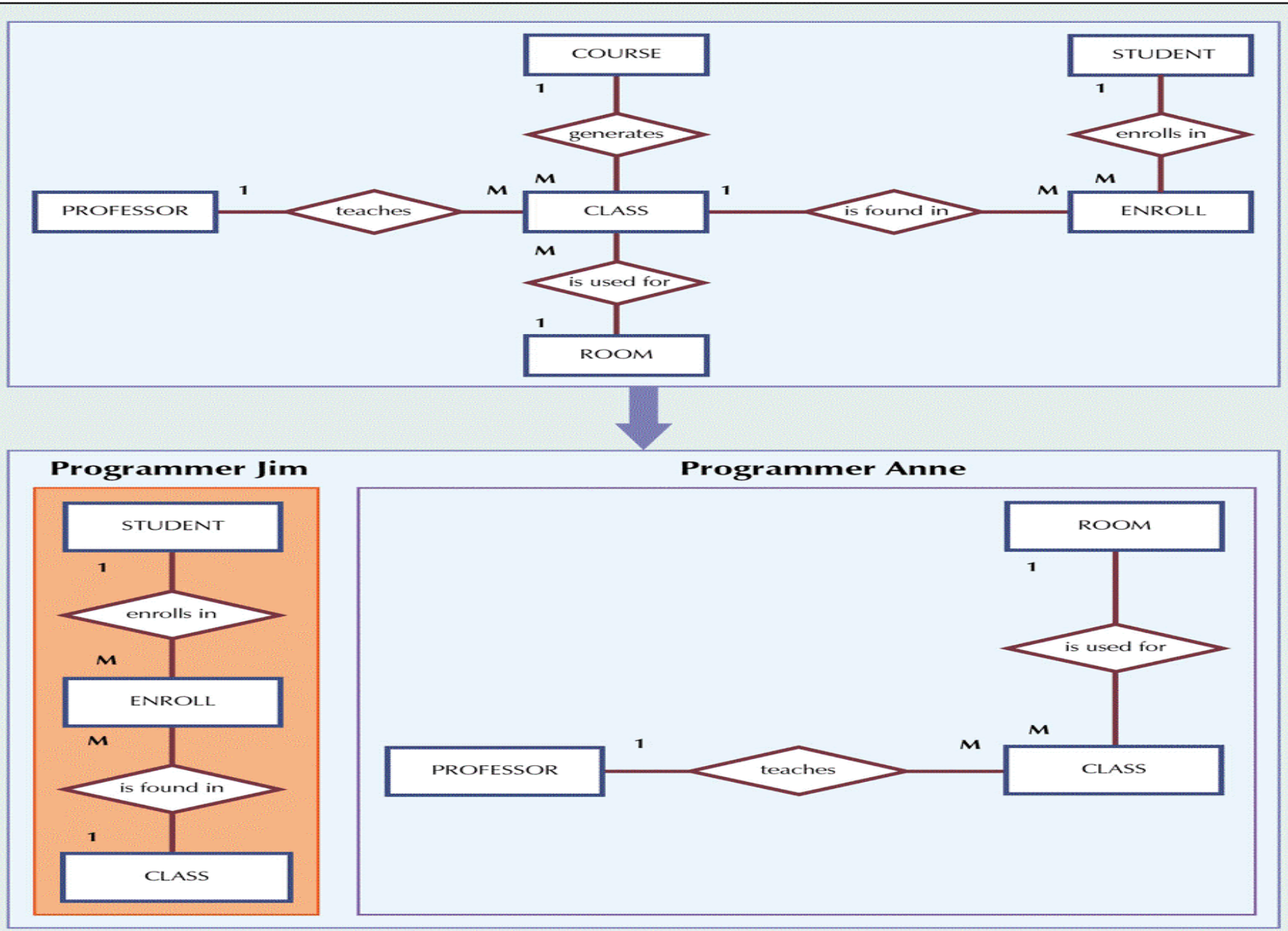
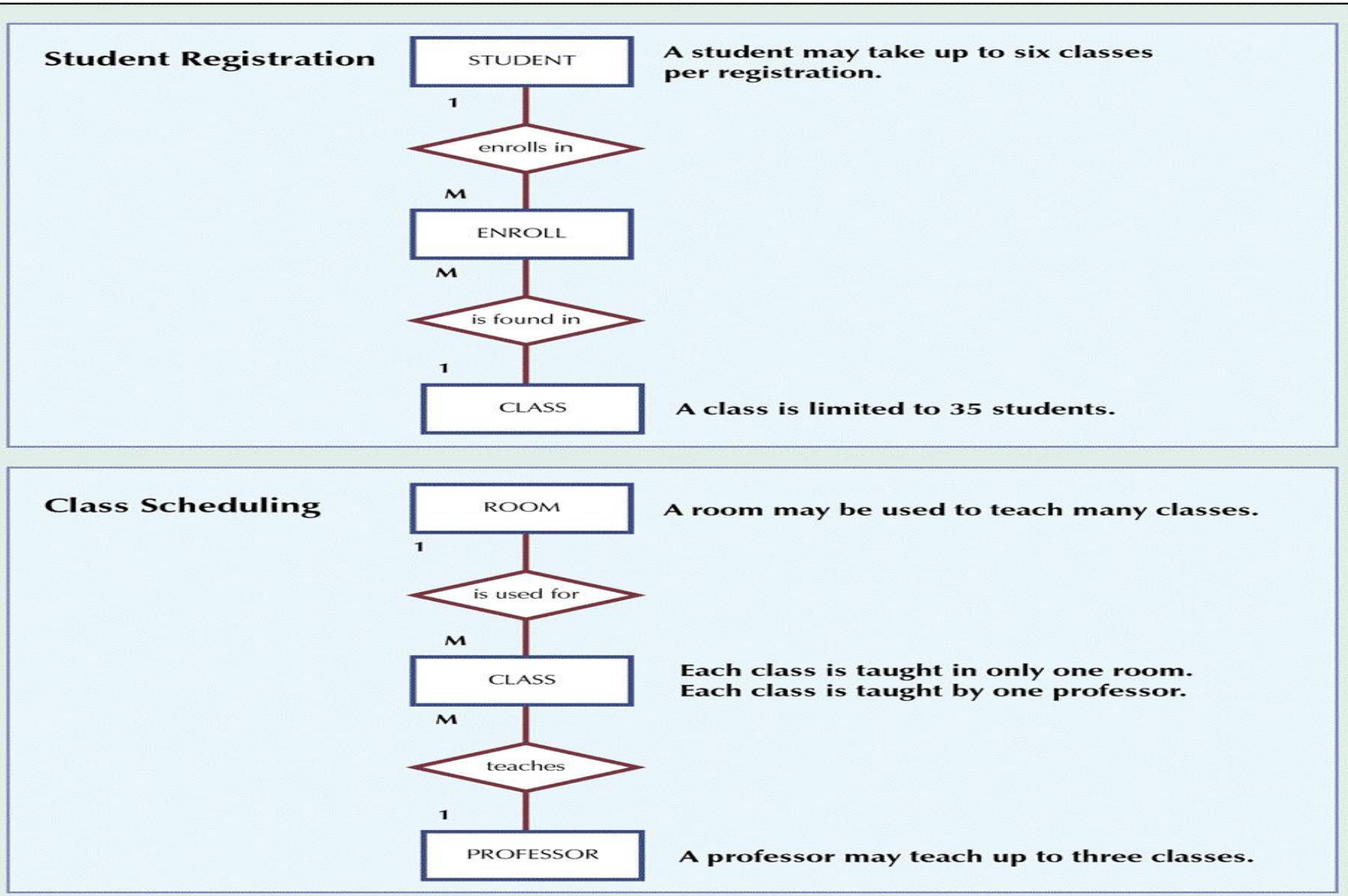


FIGURE 2.14 THE EXTERNAL MODELS FOR TINY COLLEGE

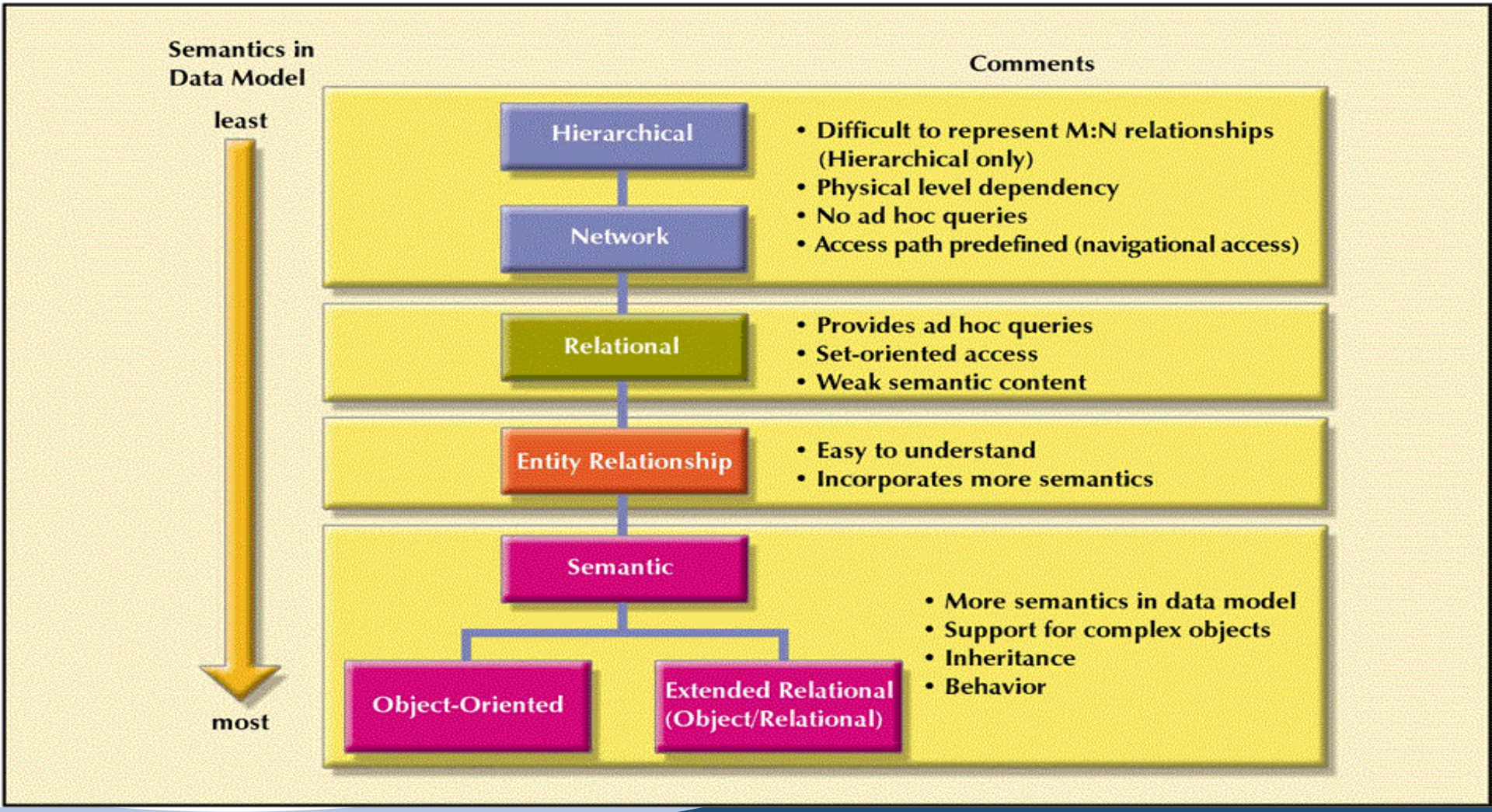


Physical Model

- Operates at lowest level of abstraction, describing the way data are saved on storage media such as disks or tapes
- Software and hardware dependent
- Requires that database designers have a detailed knowledge of the hardware and software used to implement database design

Evolution of Data Models

FIGURE 2.9 THE DEVELOPMENT OF DATA MODELS



- Hierarchical
- Network
- Relational
- Entity relationship
- Semantic
  - Object oriented
  - Object/ relational

Hierarchical Data Model

- GUAM (Generalized Update Access Method)
  - –Based on the recognition that the many smaller parts would come together as components of still larger components
- Information Management System (IMS)
  - –World's leading mainframe hierarchical database system in the 1970s and early 1980s

Characteristics

- Basic concepts form the basis for subsequent database development
- Limitations lead to a different way of looking at database design
- Each parent can have many children
- Each child has only one parent

- Tree is defined by path that traces parent segments to child segments, beginning from the left
- Hierarchical path
  - –Ordered sequencing of segments tracing hierarchical structure
- Preorder traversal or hierarchic sequence
  - –“Left-list”path

Advantages

-Conceptual simplicity

–Database security

–Data independence

–Database integrity

–Efficiency

Disadvantages

-Complex implementation

–Difficult to manage

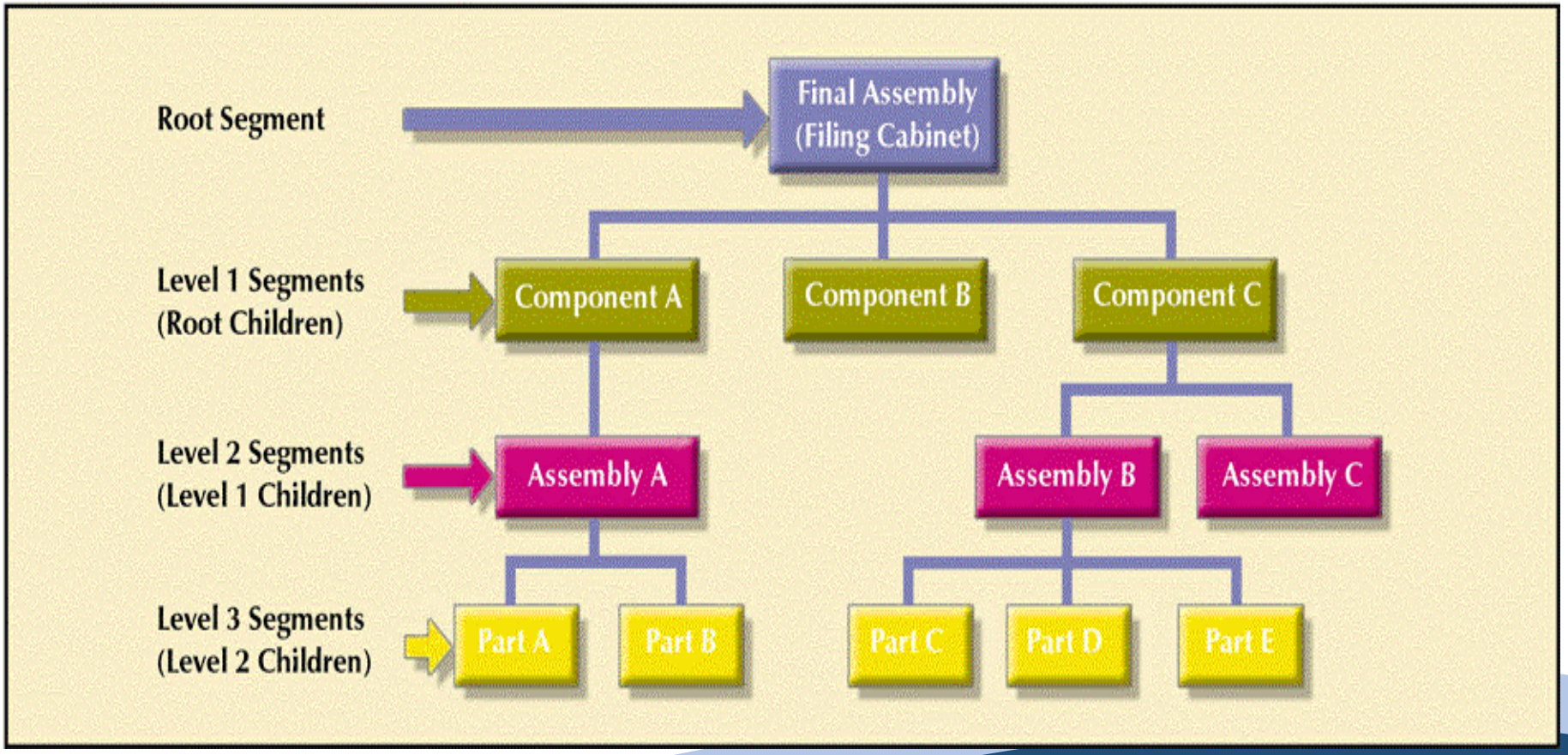
–Lacks structural independence

–Complex applications programming and use

–Implementation limitations

–Lack of standards

FIGURE 2.1 A HIERARCHICAL STRUCTURE



Network Model

- Created to:
  - –Represent complex data relationships more effectively
  - –Improve database performance
  - –Impose a database standard
- Conference on Data Systems Languages(CODASYL)
- American National Standards Institute (ANSI)
- Database Task Group (DBTG)

Crucial Database Components

- Schema
  - –Conceptual organization of entire database as viewed by the database administrator
- Subschema
  - –Defines database portion “seen” by the application programs that actually produce the desired information from data contained within the database
- Data Management Language (DML)
  - –Define data characteristics and data structure in order to manipulate the data
- Schema Data Definition Language (DDL)–Enables database administrator to define schema components
- Subschema DDL –Allows application programs to define database components that will be used
- DML –Manipulates database contents

Characteristics

- Resembles hierarchical model
- Collection of records in 1:M relationships
- Set –Relationship
  - –Composed of at least two record types
- Owner –Equivalent to the hierarchical model's parent
- Member –Equivalent to the hierarchical model's child

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

Relational Model

- Developed by Edgar Frank Codd (IBM) in 1970
- Considered ingenious but impractical in 1970
- Conceptually simple
- Computers lacked power to implement the relational model
- Today, microcomputers can run sophisticated relational database software

Basic Structure

- Relational Database Management System (RDBMS)
- Performs same basic functions provided by hierarchical and network DBMS systems, plus other functions
- Most important advantage of the RDBMS is its ability to let the user/designer operate in a human logical environment

- Table (relations) –Matrix consisting of a series of row/column intersections
  - –Related to each other by sharing a common entity characteristic
- Relational schema
  - –Visual representation of relational database's entities, attributes within those entities, and relationships between those entities

Relational Table

- Stores a collection of related entities –Resembles a file
- Relational table is purely logical structure
  - –How data are physically stored in the database is of no concern to the user or the designer
  - –This property became the source of a real database revolution

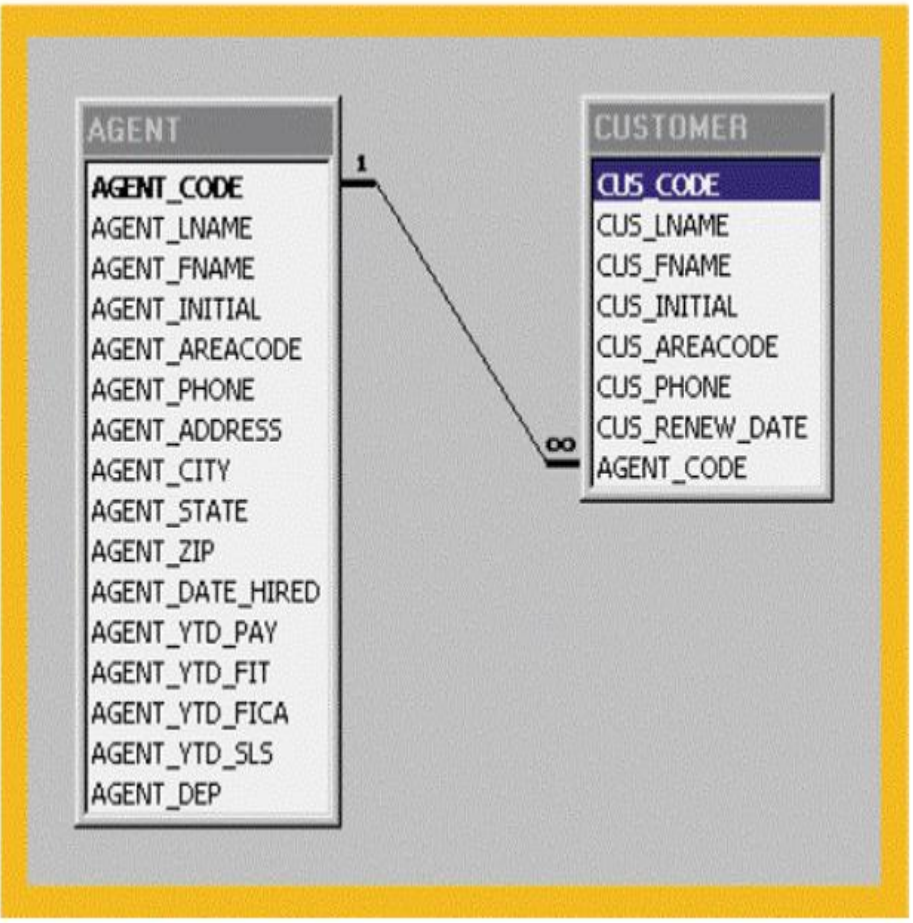


Fig: A relational Schema

Database name: Ch02_InsureCo Table name: AGENT (first six attributes)

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

Link through AGENT_CODE

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	05-Apr-2004	502
10011	Dunne	Leona	K	713	894-1238	16-Jun-2004	501
10012	Smith	Kathy	W	615	894-2285	29-Jan-2005	502
10013	Olowski	Paul	F	615	894-2180	14-Oct-2004	502
10014	Orlando	Myron		615	222-1672	28-Dec-2004	501
10015	O'Brian	Amy	B	713	442-3381	22-Sep-2004	503
10016	Brown	James	G	615	297-1228	25-Mar-2004	502
10017	Williams	George		615	290-2556	17-Jul-2004	503
10018	Farriss	Anne	G	713	382-7185	03-Dec-2004	501
10019	Smith	Olette	K	615	297-3809	14-Mar-2004	503

Fig: Linking relational Tables

Advantages

–Structural independence

–Improved conceptual simplicity

–Easier database design, management ,implementation, and use

–Ad hoc query capability

–Powerful database management system

Disadvantages

–Substantial hardware and system software overhead

–Can facilitate poor design and implementation

–May promote “islands of information” problems

Entity Relationship Model(ER Model)

- Widely accepted and adapted graphical tool for data modeling
- Introduced by Chen in 1976
- Graphical representation of entities and their relationships in a database structure

Basic Structure

- Entity relationship diagram (ERD)
  - –Uses graphic representations to model database components
  - –Entity is mapped to a relational table
- Entity instance (or occurrence) is row in table
- Entity set is collection of like entities
- Connectivity labels types of relationships
  - –Diamond connected to related entities through a relationship line

Figure: Basic Chen ERD notation

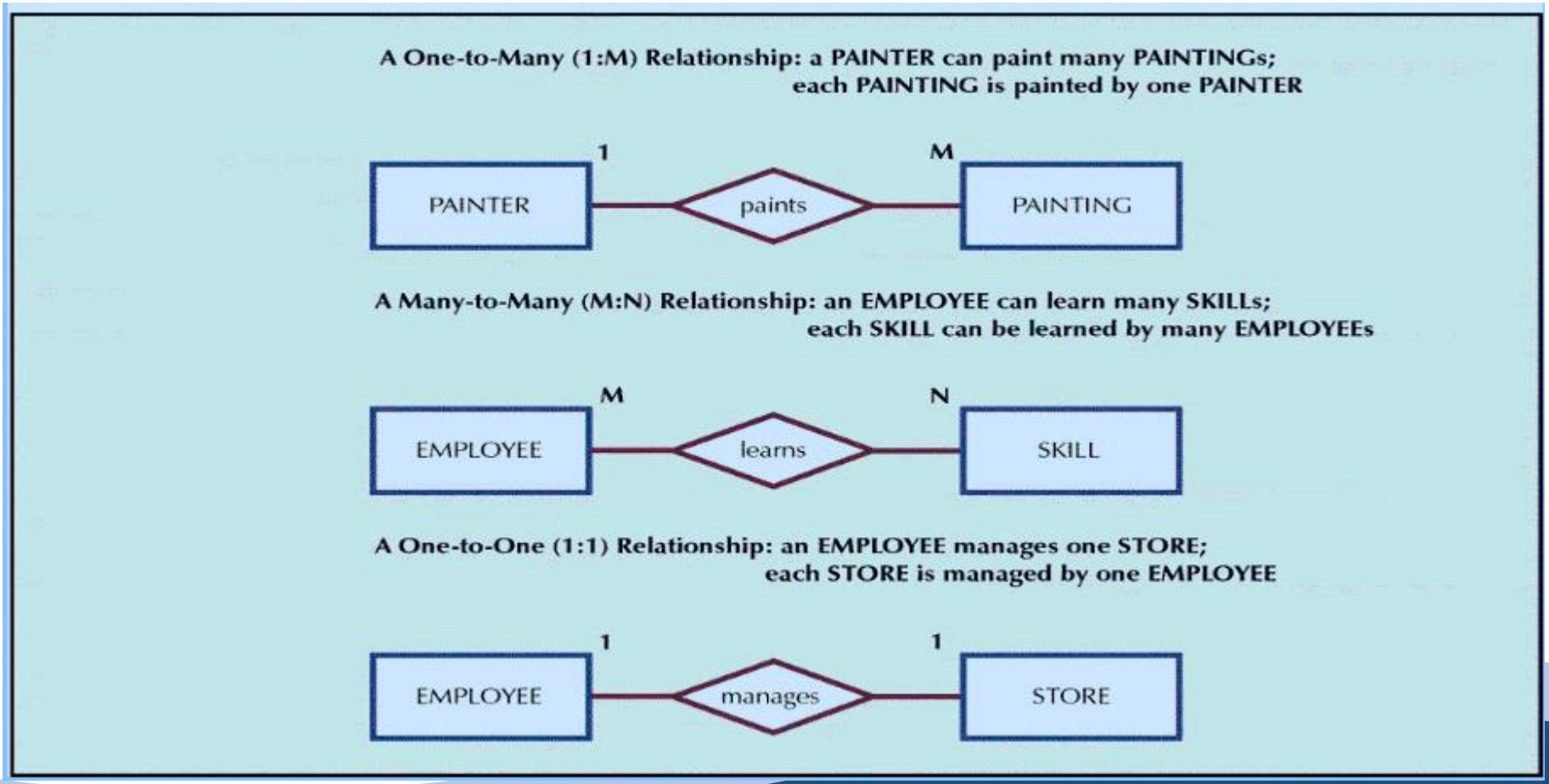
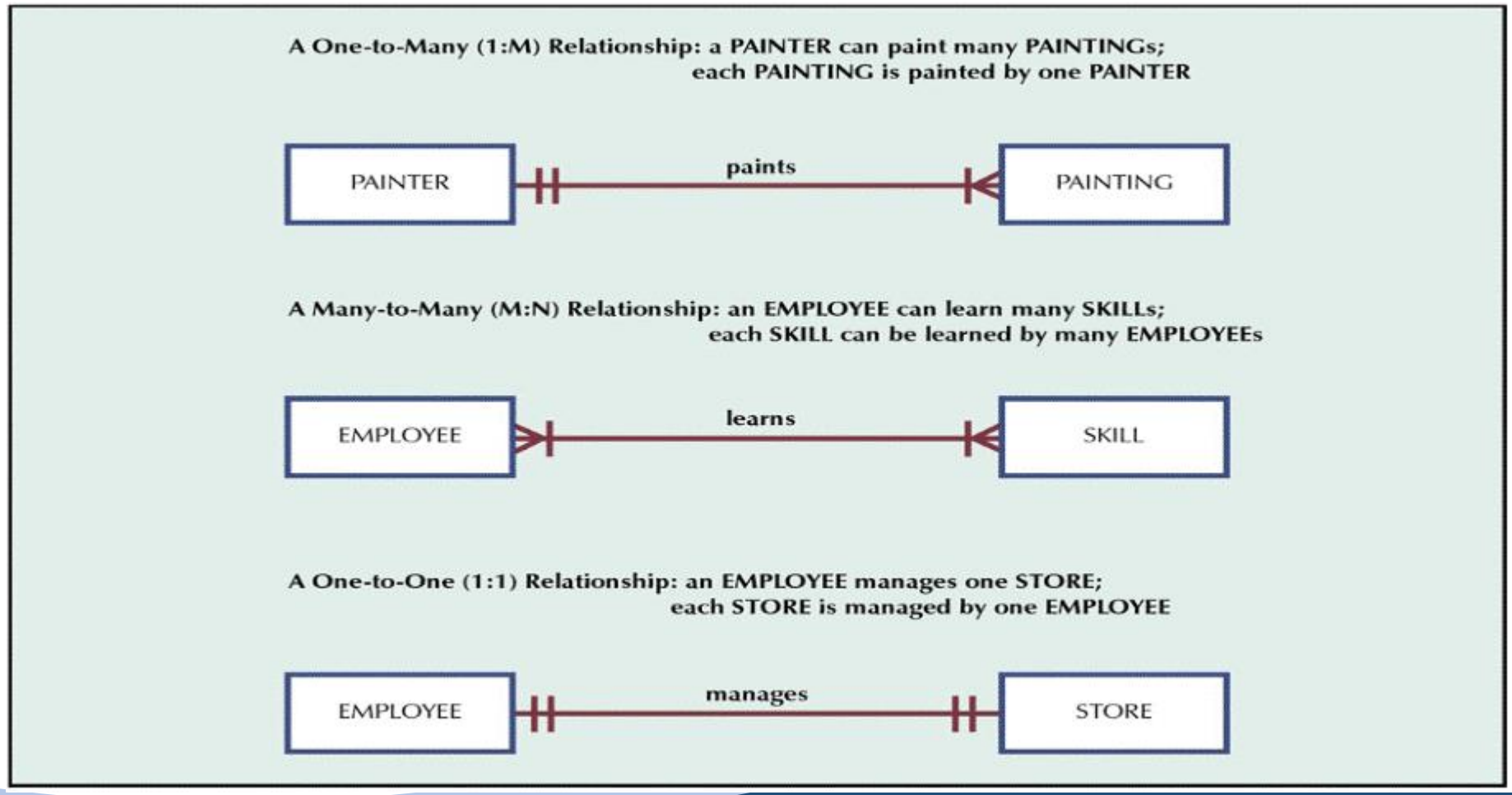


Figure: Basic Crow Foot ERD Notation



Advantages

–Exceptional conceptual simplicity

–Visual representation

–Effective communication tool

–Integrated with the relational data model

Disadvantages

–Limited constraint representation

–Limited relationship representation

–No data manipulation language

–Loss of information content

Object Oriented Model

- Semantic data model (SDM) developed by Hammer and McLeod in 1981
- Modeled both data and their relationships in a single structure known as an object
- Basis of object oriented data model (OODM)
- OODM becomes the basis for the object oriented database management system(OODBMS)
- Object is described by its factual content
  - –Like relational model's entity
- Includes information about relationships between facts within object and relationships with other objects
  - –Unlike relational model's entity

Contd..

- Subsequent OODM development allowed an object to also contain operations
- Object becomes basic building block for autonomous structures

Developments that Boosted OODM's Popularity

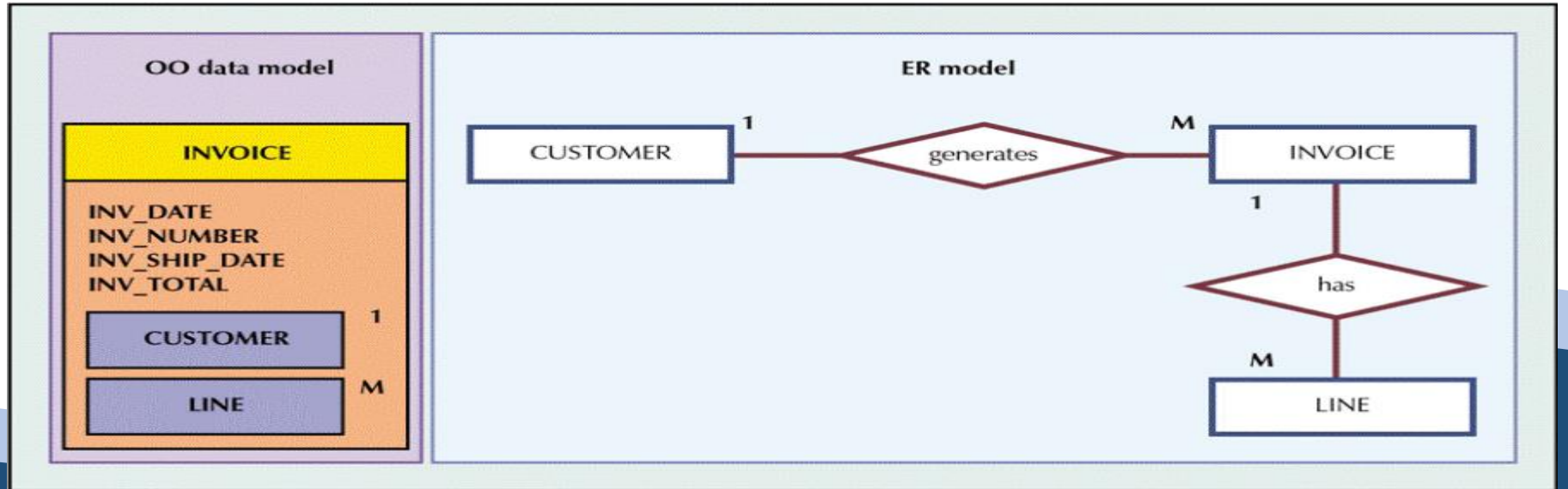
- Growing costs put a premium on code reusability
- Complex data types and system requirements became difficult to manage with a traditional RDBMS
- Became possible to support increasingly sophisticated transaction & information requirements
- Ever-increasing computing power made it possible to support the large computing overhead required

Object Oriented Data Model—Basic Structure

- Object: abstraction of a real-world entity
- Attributes describe the properties of an object
- Objects that share similar characteristics are grouped in classes
- Classes are organized in a class hierarchy
- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of classes above it

OODM

ER model



Advantages

–Adds semantic content

–Visual presentation includes semantic content

–Database integrity

–Both structural and data independence

Disadvantages

–Slow pace of OODM standards development

–Complex navigational data access

–Steep learning curve

–High system overhead slows transactions

–Lack of market penetration

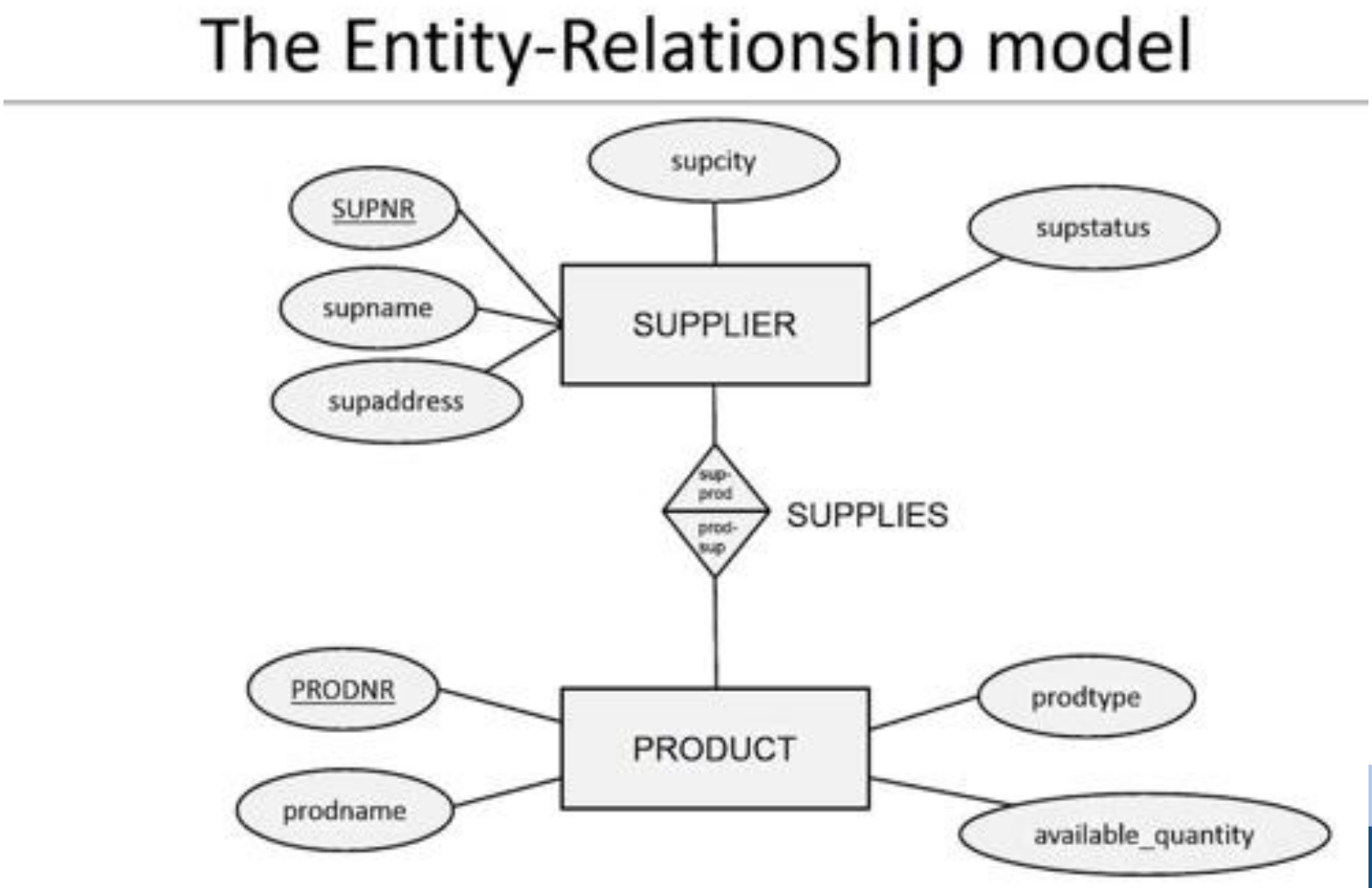
Entity Relationship Model(ER Model)

- An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).
- An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.
- The ER model employs three basic concepts : Entity Set, Relationship set and attributes.

ER Diagram:

- An ER diagram shows the relationship among entity sets.
- An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

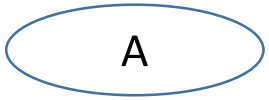
A simple ER diagram



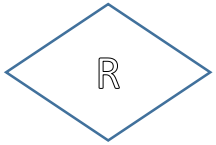
Symbols used for ER diagram

Entity

Attributes

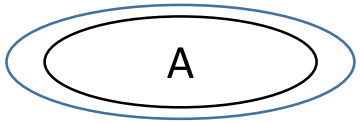


Relationship set



Links

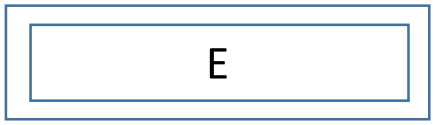
Multivalued Attribute



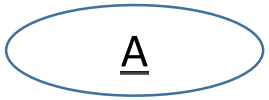
Derived Attribute

Total Participation

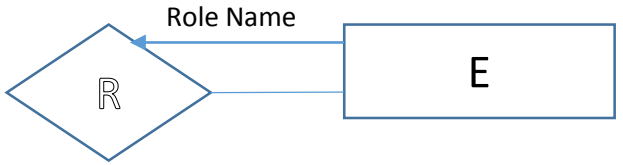
Weak Entity set



Primary Key



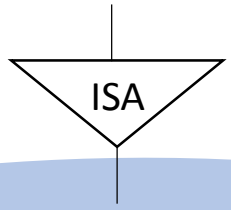
Role Name



Discriminating Attribute

Generalization/

Specialization



Entity:

- An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

For example: In the following ER diagram we have two entities Student and College and these two entities have many to one relationship as many students study in a single college.

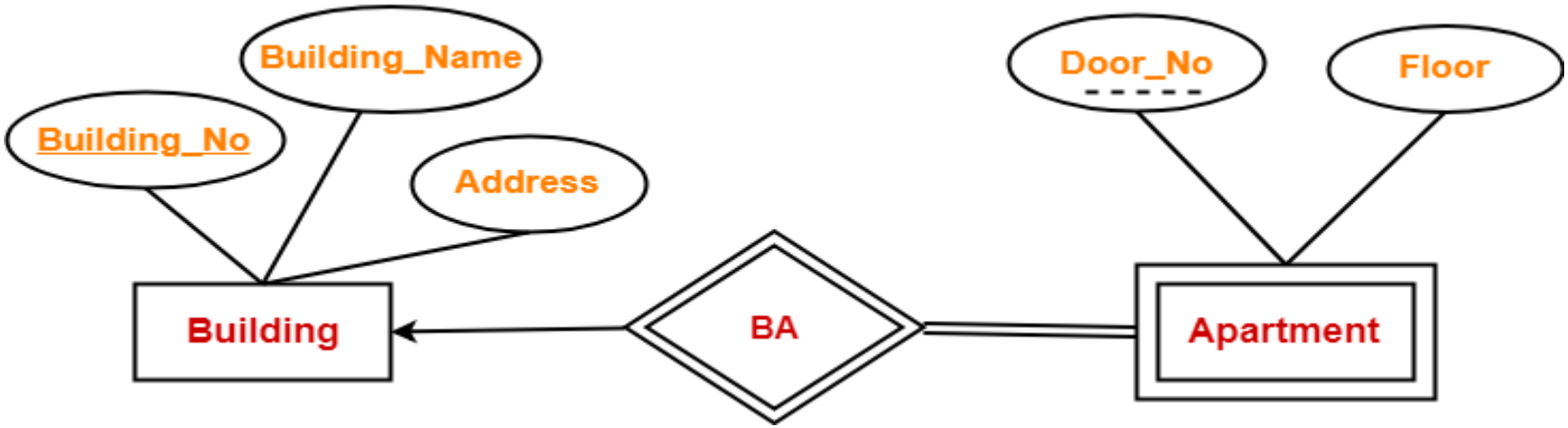
Strong Entity: an entity set which may have sufficient attributes to form a primary key is called strong entity

Weak Entity:

- An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity. The weak entity is represented by a double rectangle.

For example – a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so bank account is a weak entity.

Example: Strong and Weak Entity set



Strong entity set: Building
Primary key: Building_No
Weak Entity Set: Apartment
Discriminator: Door_No
Relationship set: BA
Note: relationship set associating the weak entity set is denoted by double diamond symbol

Attribute

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

1. Key attribute
2. Simple attribute
3. Composite attribute
4. Multivalued attribute
5. Derived attribute

Key Attribute:

A key attribute can uniquely identify an entity from an entity set.

For example, student roll number can uniquely identify a student from a set of students. Key attribute is represented by oval same as other attributes however the **text of key attribute is underlined**.

Simple Attribute: An attribute that cannot be divided in sub class. Eg. College_name.

Composite Attribute:

An attribute that is a combination of other attributes is known as composite attribute.

For example, In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country.

Multivalued attribute:

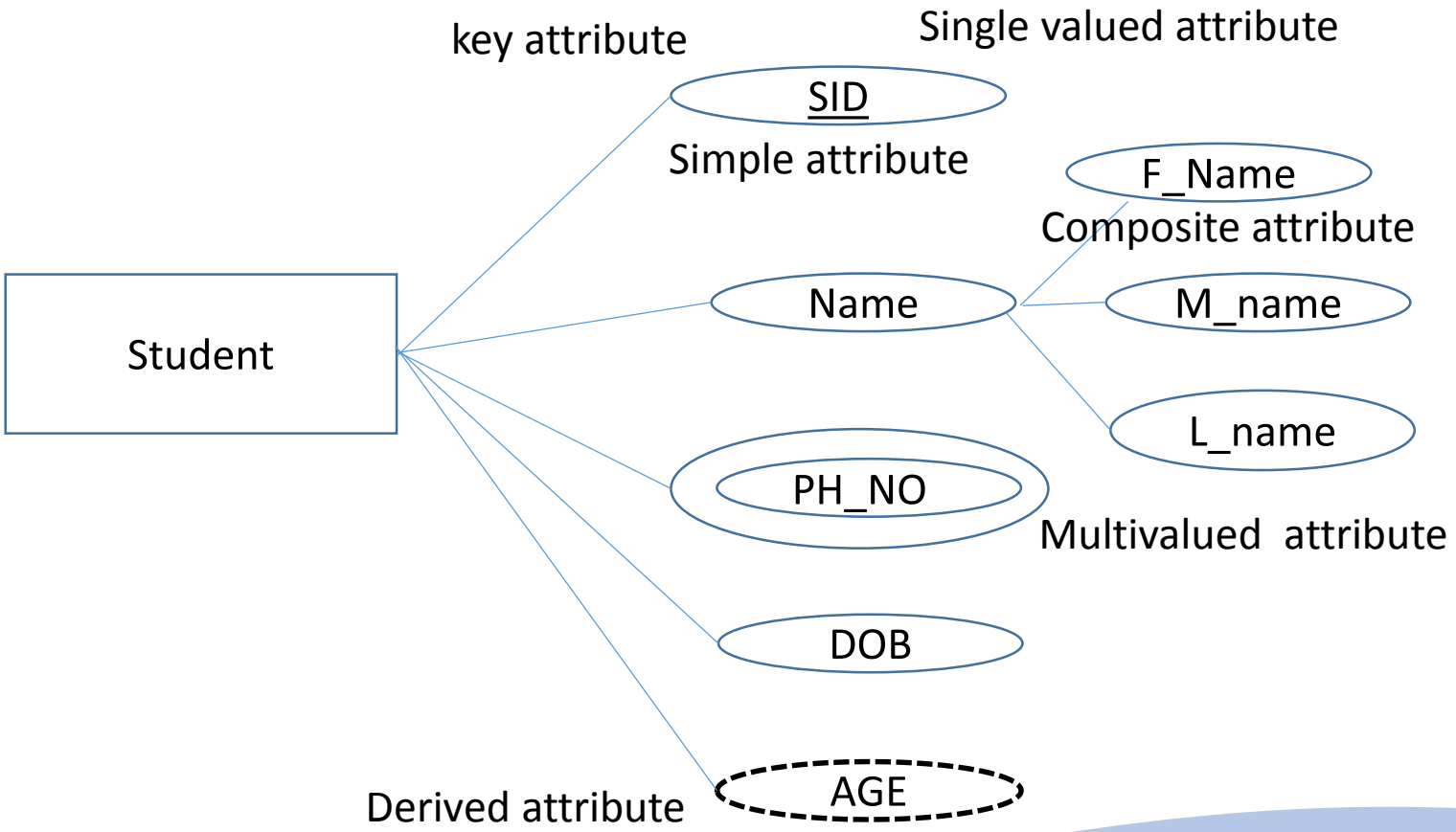
- An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram.

For example – A person can have more than one phone numbers so the phone number attribute is multivalued.

Derived attribute:

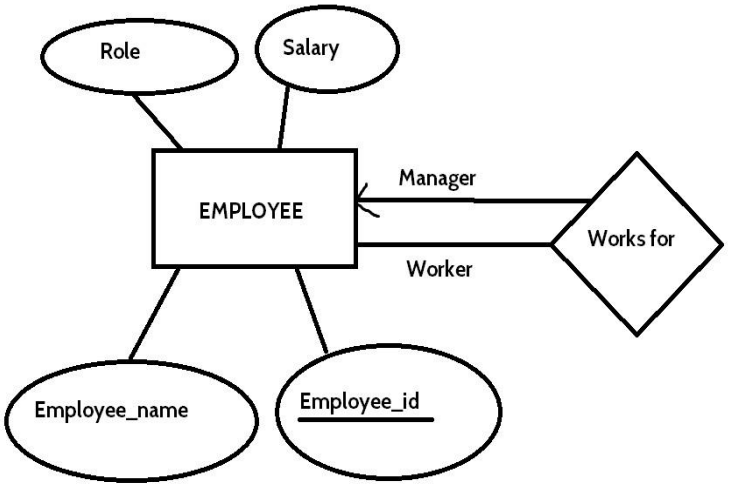
- A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram.
- For example – Person age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).

Example: Types of Attributes



Roles:

- A database role is a collection of any number of permissions/privileges that can be assigned to one or more users. A database role also is also given a name for that collection of privileges.

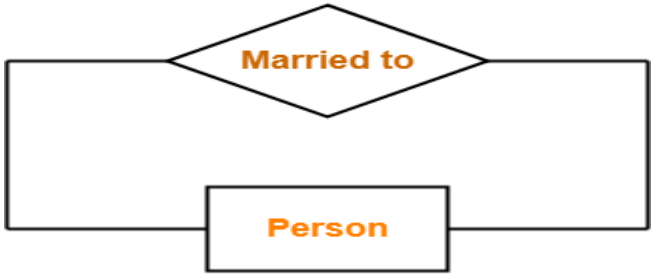


Degree of relationship:

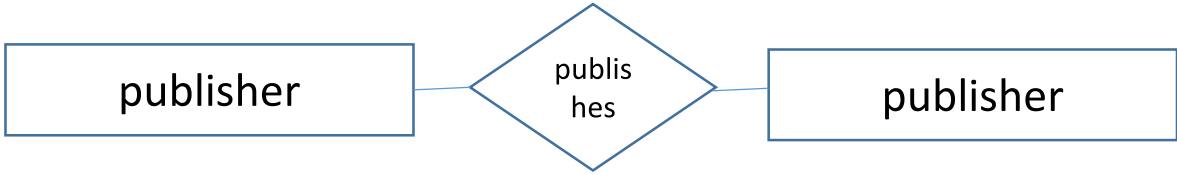
It is the number of entities associated with the relationship set. The n-ary relationship is the general form of degree n.

Types:
Unary relationship
Binary relationship
Ternary relationship
Quarternary relationship

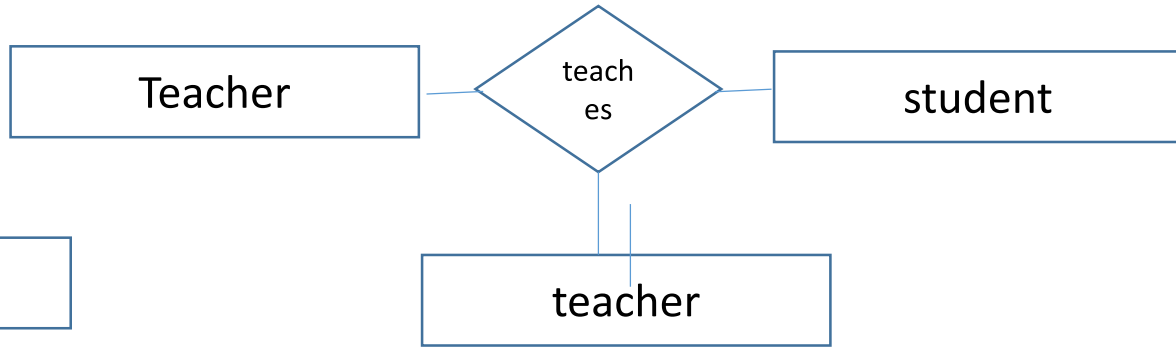
Example



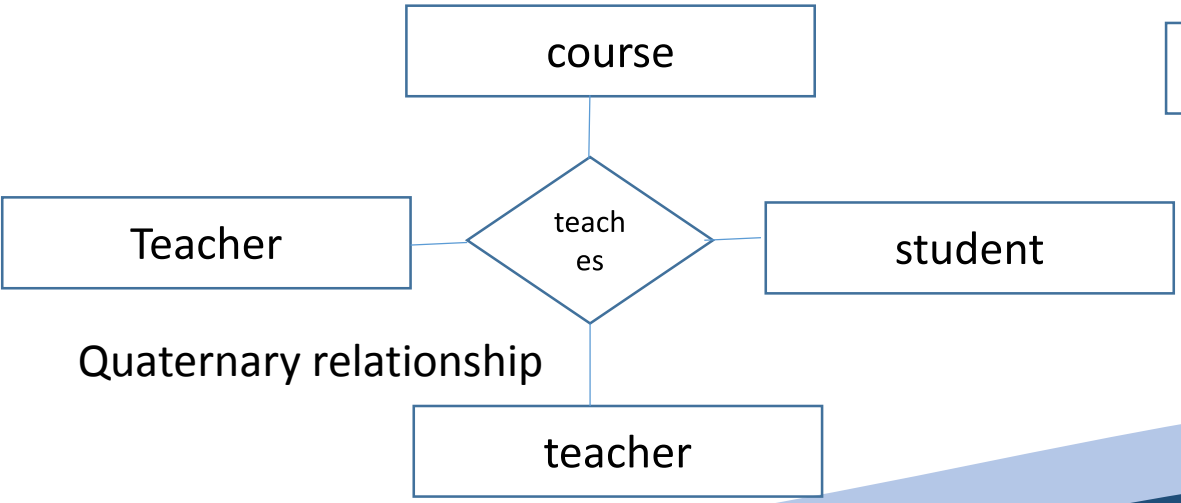
Unary Relationship Set



Binary relationship



Ternary relationship



Quaternary relationship

ER modeling - Constraints

a) Mapping Cardinalities

b) Participation Constraints

Mapping Cardinalities: express the number of entities to which another entity can be associated via a relationship.

For binary relationship sets between entity sets A and B, the mapping cardinality must be one of:

One-to-one: An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.

One-to-many: An entity in A is associated with any number in B. An entity in B is associated with at most one entity in A.

Many-to-one: An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A.

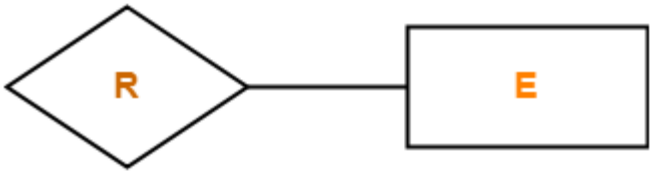
Many-to-many: Entities in A and B are associated with any number from each other. (Figure 2.6)

Example

Participation Constraint:

- Total Participation: It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.
- That is why, it is also called as **mandatory participation.**
- Total participation is represented using a double line between the entity set and relationship set.

- Partial Participation:It specifies that each entity in the entity set may or may not participate in the relationship instance in that relationship set.
- That is why, it is also called as **optional participation.**
- Partial participation is represented using a single line between the entity set and relationship set.



ER Diagram: A library management System

ER Diagram: A Hotel management system

ER Diagram: A Banking system

ER Diagram: A Hospital management system

ER Diagram: An Online BOOK Store

ER Diagram: An Airline Reservation system

Case Study 1

You are given the requirement for a simple database for National Football League(NFL). The NFL has many teams and each team has a name, a city, a coach, a captain and a set of players. Each player belongs to only one team and each player has a name, a position(left wing, mid fielder or goalkeeper) a skill level a set of injury records. A team captain is also a player and a game is played between two teams(referred as host team and guest team) and has a match date (such as June 11,2018) and score such as (2 or 5). Construct an ER diagram for the following NFL database.

Specialization and Generalization

Concept of subclass and super class

Subclass: is an entity type that is the part of super class that has attributes distinct from other sub groups

Super class: includes distinct subclasses.

- superclass attributes are shared among subclass.

Eg: vehicle (superclass)

motorcycle, car, truck (subclass)

Specialization: process of forming a subclass

- top down process
- identifying sub groups within an entity set which have attributes not shared by all entities(unique properties)

Generalization: process of forming a super class

- proceeds from the recognition of a number of entities that share common attributes

Example

Aggregation

- One limitation of ER model is that it can’t express relationship among relationships.
- Aggregation is the process of compiling information of an object
- It is an abstraction through which relationship are treated as higher level entities.

Ex: relationship between employee, branch, job, manager

Schema Diagram

- A database which confirm to an ER diagram can be represented by a collection of schema.

Rules for schema Diagram:

- Each relation appears in rectangular box
- Attributes are listed inside box and relation name is shown above it.
- If there is PK , we draw a horizontal line below the attribute.
- FK dependencies are shown by the arrow from foreign key attribute to the PK in reference relation.

EX:

branch(b_name, assets)

Account (acc_no, b_name)

Depositor(cname,acc_no)

Customer (cname,c_street,c_city)

Loan(loan_no, b_name,amount)

Borrower(cname,loan_no_

Rules of ER diagram to Tables

- Strong entity sets reduces to a table with same attribute
- A weak entity set becomes a table that includes a column for PK of reference relation plus column of its own entity set
- Tabular reprenestation of relationship set consit of PK representing the relations
- Tabular representation of generalization:

-create a table for the superclass entity set. For sub class entity set, create a table which includes the column of the subclass plus PK of superclass