Chapter 4: Data Models

In this chapter, you will learn:

- Why data models are important
- About the basic data-modeling building blocks
- What business rules are and how they affect database design
- How the major data models evolved, and their advantages and disadvantages
- How data models can be classified by level of abstraction

Data Model and Data Modeling

- **Data model** is a representation, usually graphical, of a real-world data structure.
- **Data modeling** refers to the process of creating a specific data model for a problem to determine what data and relationships should be stored in the database.
- Two major data models in use today are:
  - Entity-Relationship Model (ERD)
  - Class Diagram (Object-Oriented)

The Importance of Data Models

1. As a communication tool to facilitate interaction among system designers, applications programmers, and end users.
2. Good database design uses an appropriate data model as its foundation. **reduce redundancy, eliminate anomalies**
3. End-users have different views of data. Data model provides the whole view of data.
4. Data model organizes data for various users who have different needs for data.

Business Rules

- A **business rule** is a brief, precise, and unambiguous description of a policy, procedure, or principle within a specific organization.
- The main sources of business rules are company managers, policy makers, department managers, and written documentation such as a company's procedures, standards, and operations manuals.
- A faster and more direct source of business rules is direct interviews with end users. Unfortunately, because perceptions differ, end users are sometimes a less reliable source when it comes to specifying business rules. Interviews with several people who perform the same job often get different perceptions of what the job components are.

Examples of Business Rules

1. A customer may make many payments on account. Each payment on account is credited to only one customer.
2. A machine operator may not work more than 10 hours in any 24-hour period.
3. A business trip destination must be at least 100 miles away for an airline ticket to be purchased.
4. A training cannot be scheduled for fewer than 10 employees or for more than 30 employees.
5. A lab cannot be scheduled for demonstration purposes for more than one-hour per day.
6. A customer may receive many invoices. Each invoice is received by only one customer.

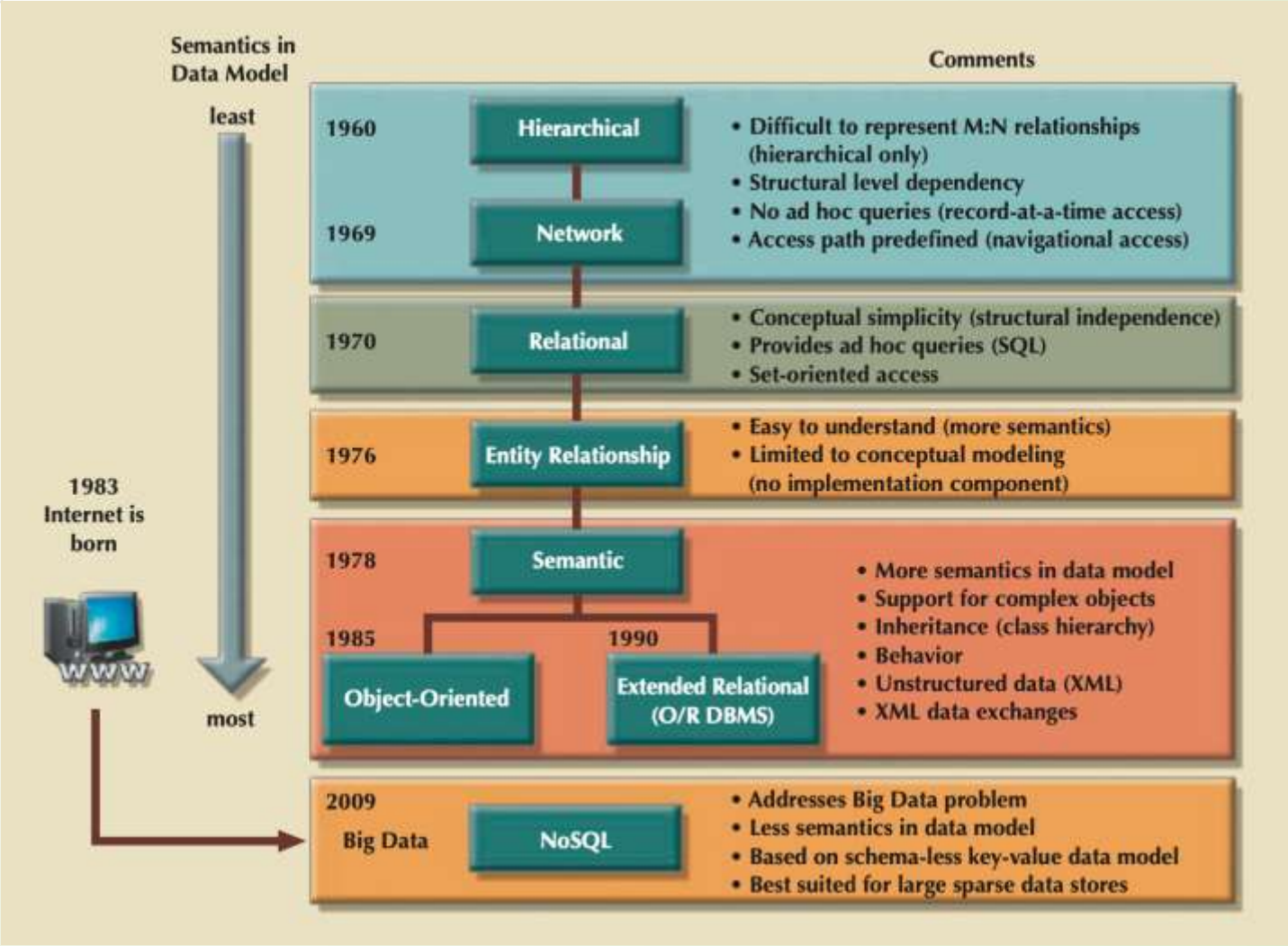
Importance of Business Rules

1. They allow the database designer to develop appropriate relationship participation rules and constraints and to create an accurate data model.
2. They standardize the company's view of data.
3. They serve as a communication tool between users and database designers.
4. They allow database designers to understand the nature, role, and scope of data.
5. They help designers to understand business processes.

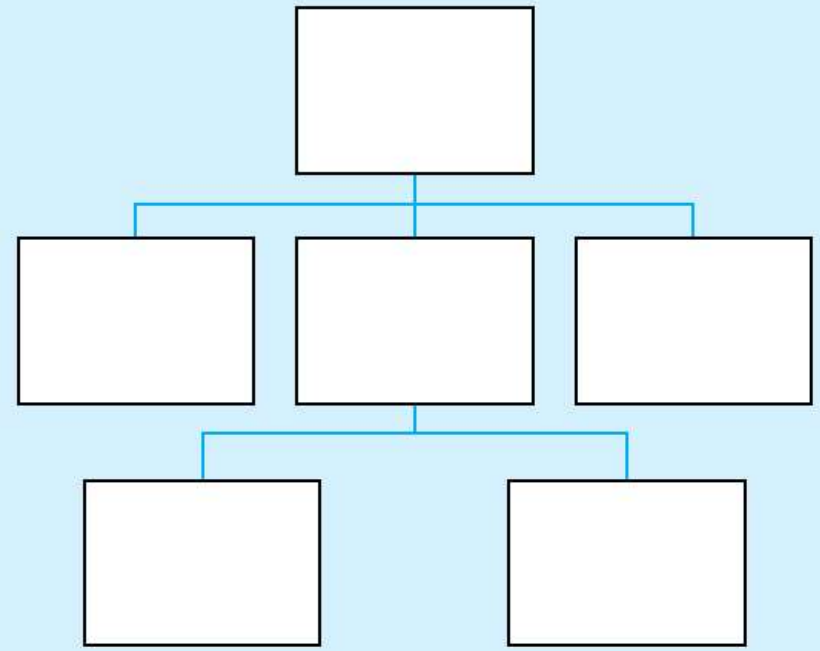
Translating Business Rules into Data Model Components

- Data model components are **entities (table/relation)**, **attributes (fields/column)**, **relationships**, and **constraints**.
- A **noun** in a business rule will translate into an **entity** in data model.
- A **verb** (active or passive) that associates the nouns will translate into a **relationship** among the entities.
- For example, the business rule "a customer may generate many invoices" contains two nouns (customer and invoices) and a verb (generate) that associates the nouns. From this business rule, you could deduce the following:
  - Customer and invoice are entities.
  - There is a *generate* relationship between customer and invoice.

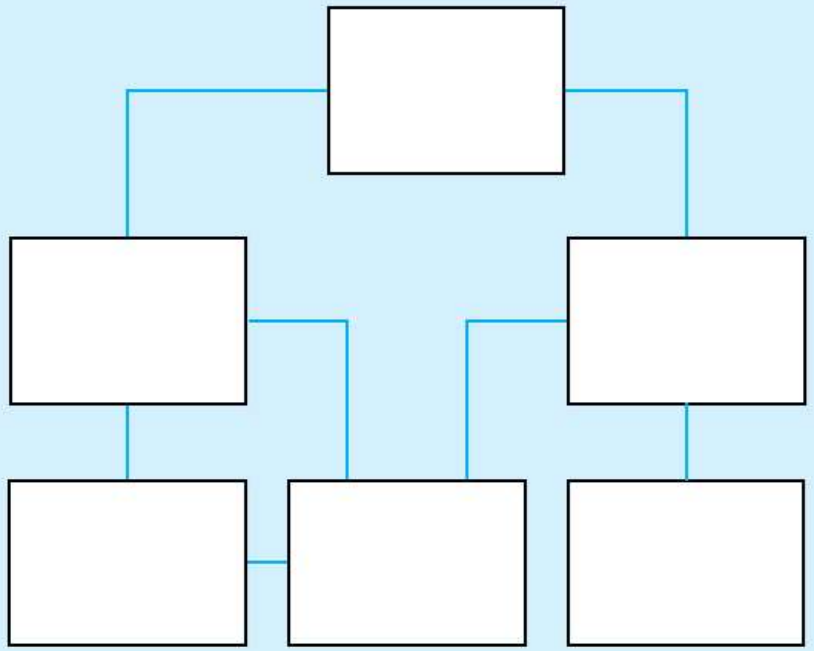
Evolution of Data Models



Evolution of Data Models



Hierarchical database model

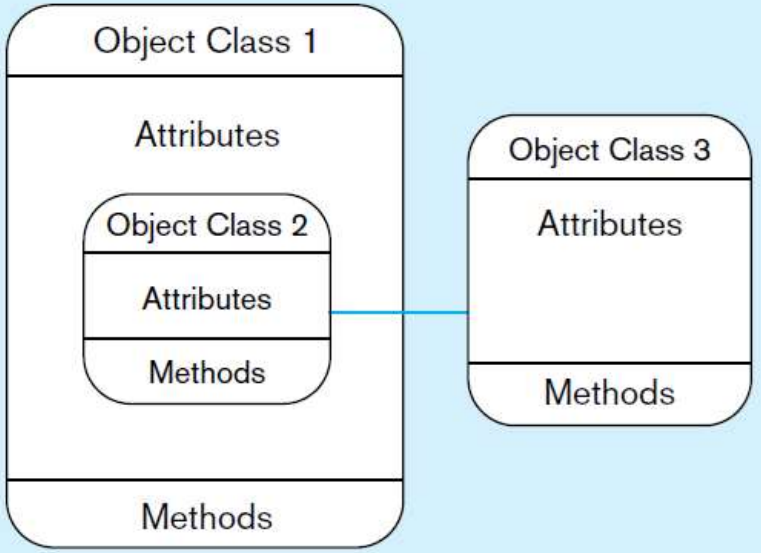


Network database model

RELATION 1 (PRIMARY KEY, ATTRIBUTES...)

RELATION 2 (PRIMARY KEY, FOREIGN KEY, ATTRIBUTES...)

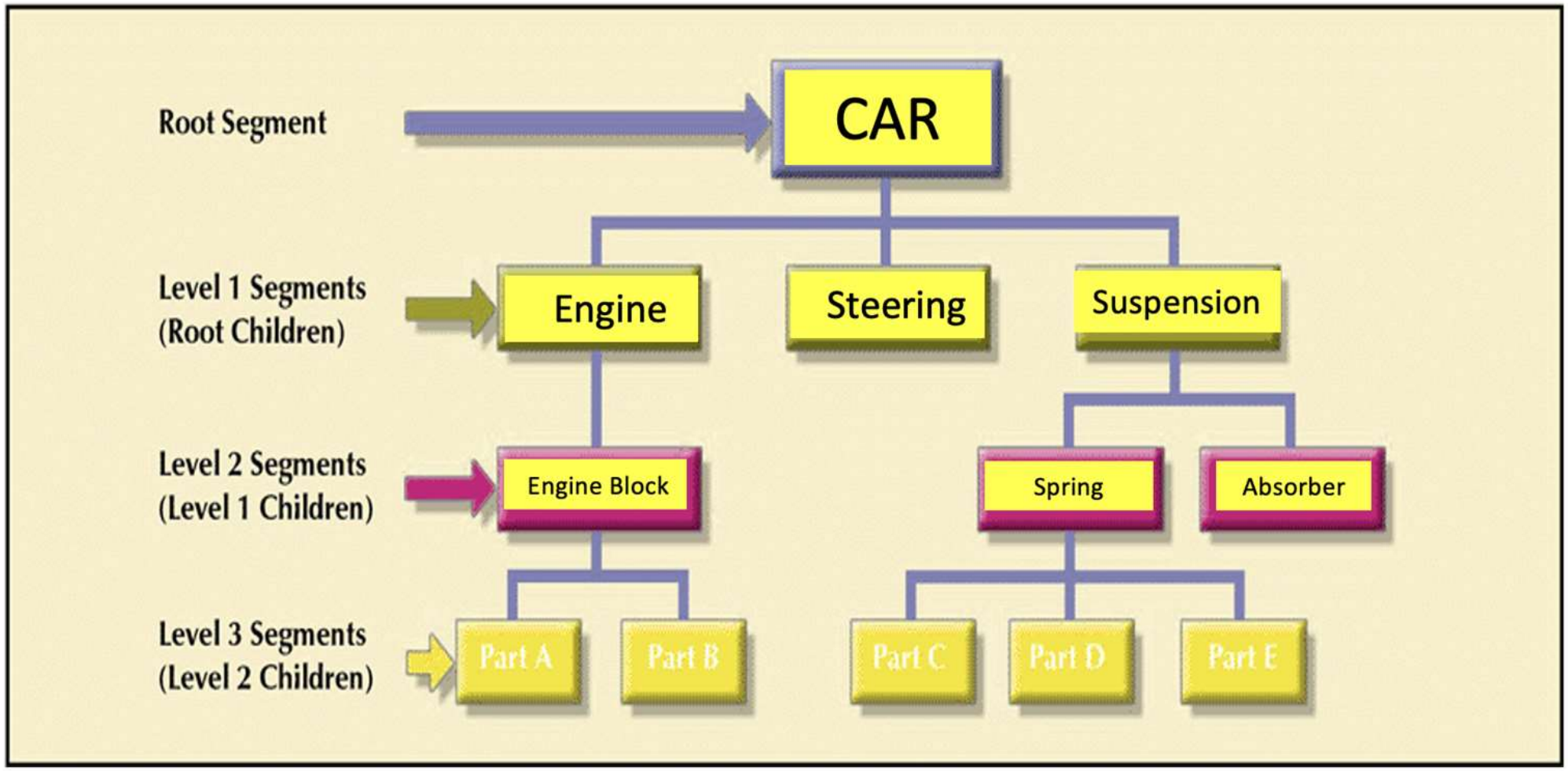
Relational database model



Object-oriented database model

A Hierarchical Structure

FIGURE 2.1 A HIERARCHICAL STRUCTURE



Characteristics of Hierarchical Model

- Hierarchical model is based on an upside-down tree structure which contains **levels** or **segments**.
- Within the hierarchy, a higher layer is perceived as the **parent** of the segment directly below it, which is called the **child**.
- Each parent can have many children.
- Each child has only one parent.
- A **tree** is defined by the path that traces the parent segments to child segments.

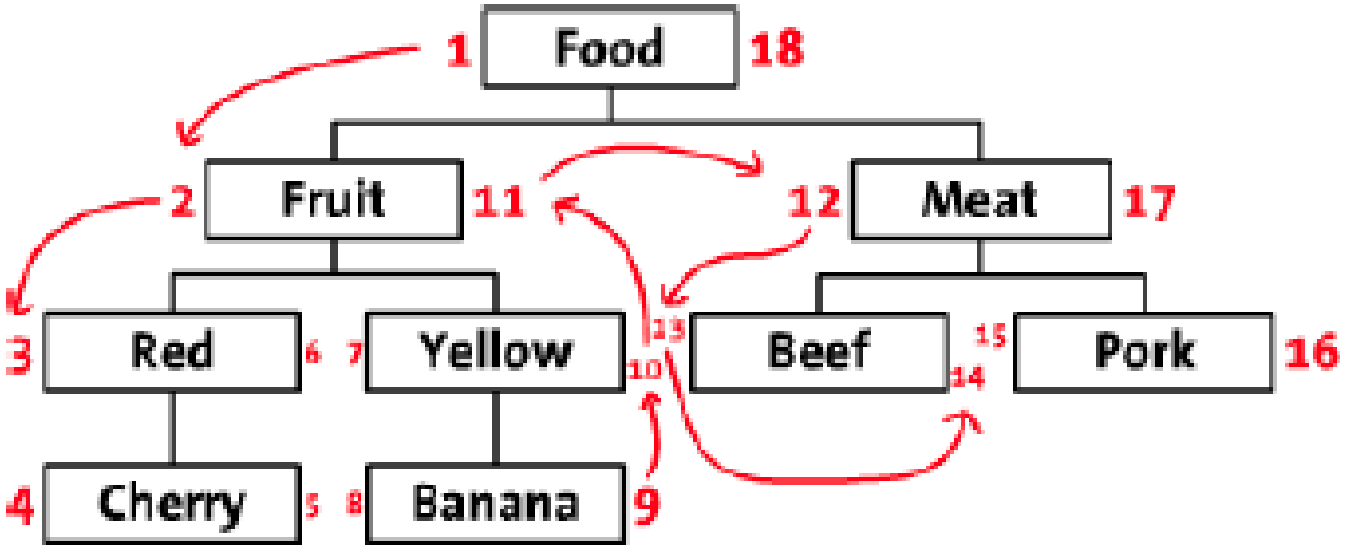
The Hierarchical Model

- **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

The Hierarchical Model

Hierarchical Database Model :

- The main problem with this database is searching a data is extremely difficult.
- Search a particular data in this database needs more time.
- This model was failed.



Preorder traversal or hierarchic sequence "Left-list" path takes 18 steps to find "Pork" and return its value.

The Network Model

- Network model was created to
  - Represent complex data relationships more effectively.
  - Improve database performance.
  - Impose a database standard.

- 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. E.g., schema, subschema, DML, and DDL.

ORACLE

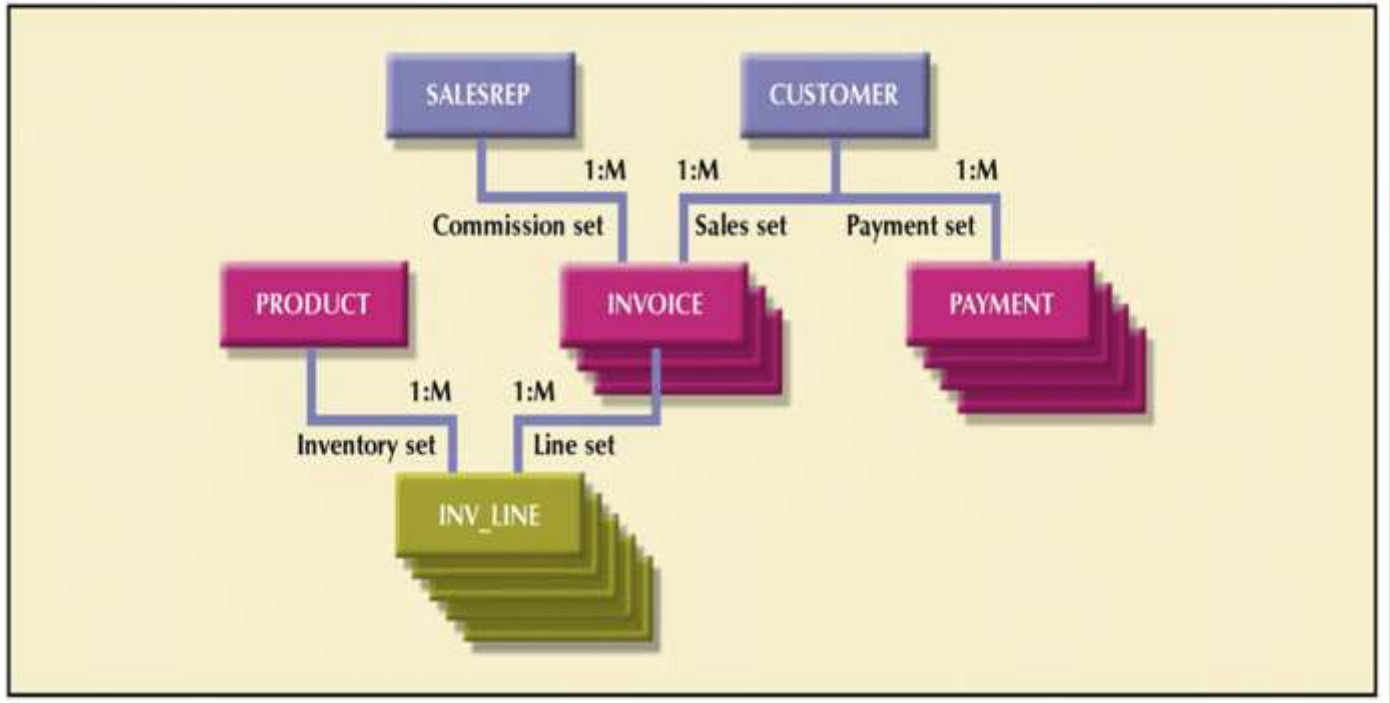
The Network Model

Network Database Model :

- Designed to reduce the problem caused by the hierarchical data model.
- Allows more links between the child databases.
- Created not only to reduce the chance of redundant data, but it also makes searching for data much easier.

A Network Data Model

FIGURE 2.3 A NETWORK DATA MODEL



The Network Data Model

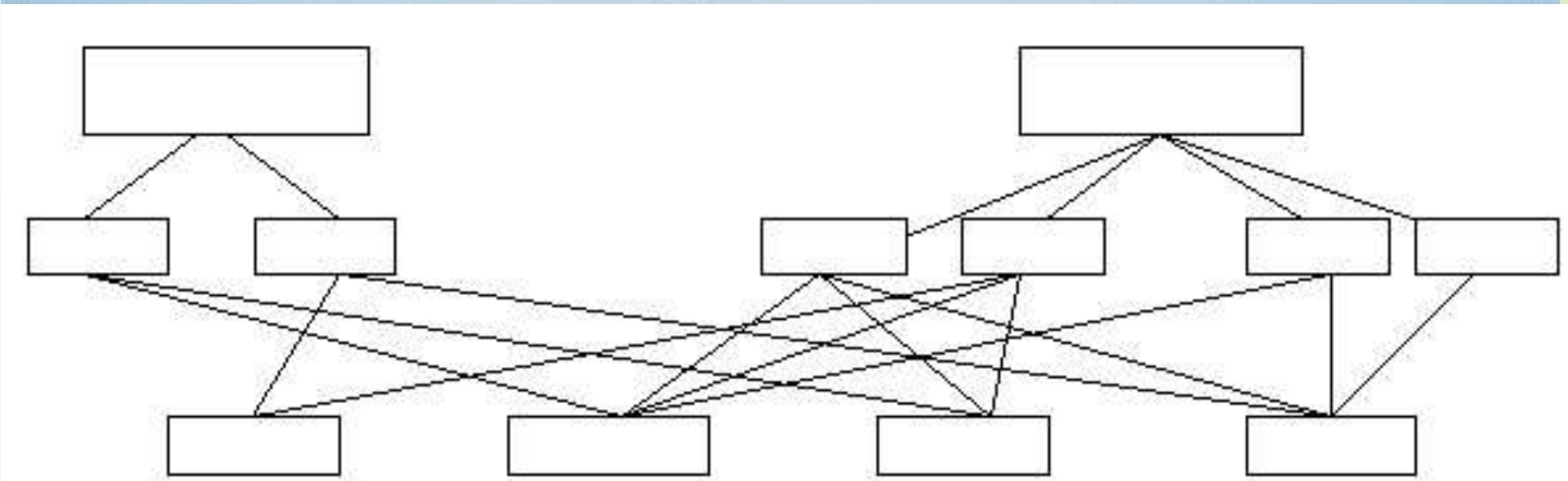
- **Advantages**
  - Conceptual simplicity
  - Handles more relationship types
  - Data access flexibility
  - Promotes database integrity
  - Data independence
  - Conformance to standards
- **Disadvantages**
  - System complexity
  - Lack of structural independence

The Network Model

Network Database Model :

Problem with this data model :

- Only database experts can use this database successfully.
- Very difficult for the general public to use network databases for real-life applications.



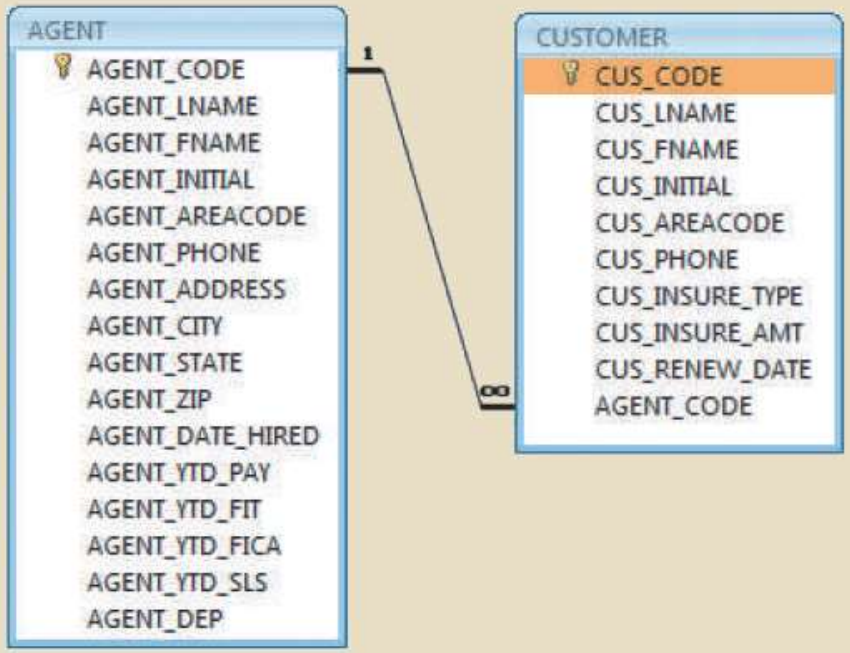
The Relational Model

- A **table** (sometimes called a **relation**) is a two-dimensional structure composed of intersecting rows and columns. Each row in a table is called a **record** (or **tuple**). Each **column** represents an **attribute**.
- The relational data model is implemented through a very sophisticated **relational database management system** (**RDBMS**). The RDBMS performs the same basic functions provided by the hierarchical and network DBMS systems, in addition to other functions that make the relational data model easier to understand and implement.

A Relational Diagram

- A **relational diagram** is a representation of the relational database's entities, the attributes within those entities, and the relationships between those entities.

FIGURE 2.2 A RELATIONAL DIAGRAM



Linking Relational Tables

FIGURE 2.1 LINKING RELATIONAL TABLES

Table name: AGENT (first six attributes)

Database name: Ch02_InsureCo

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

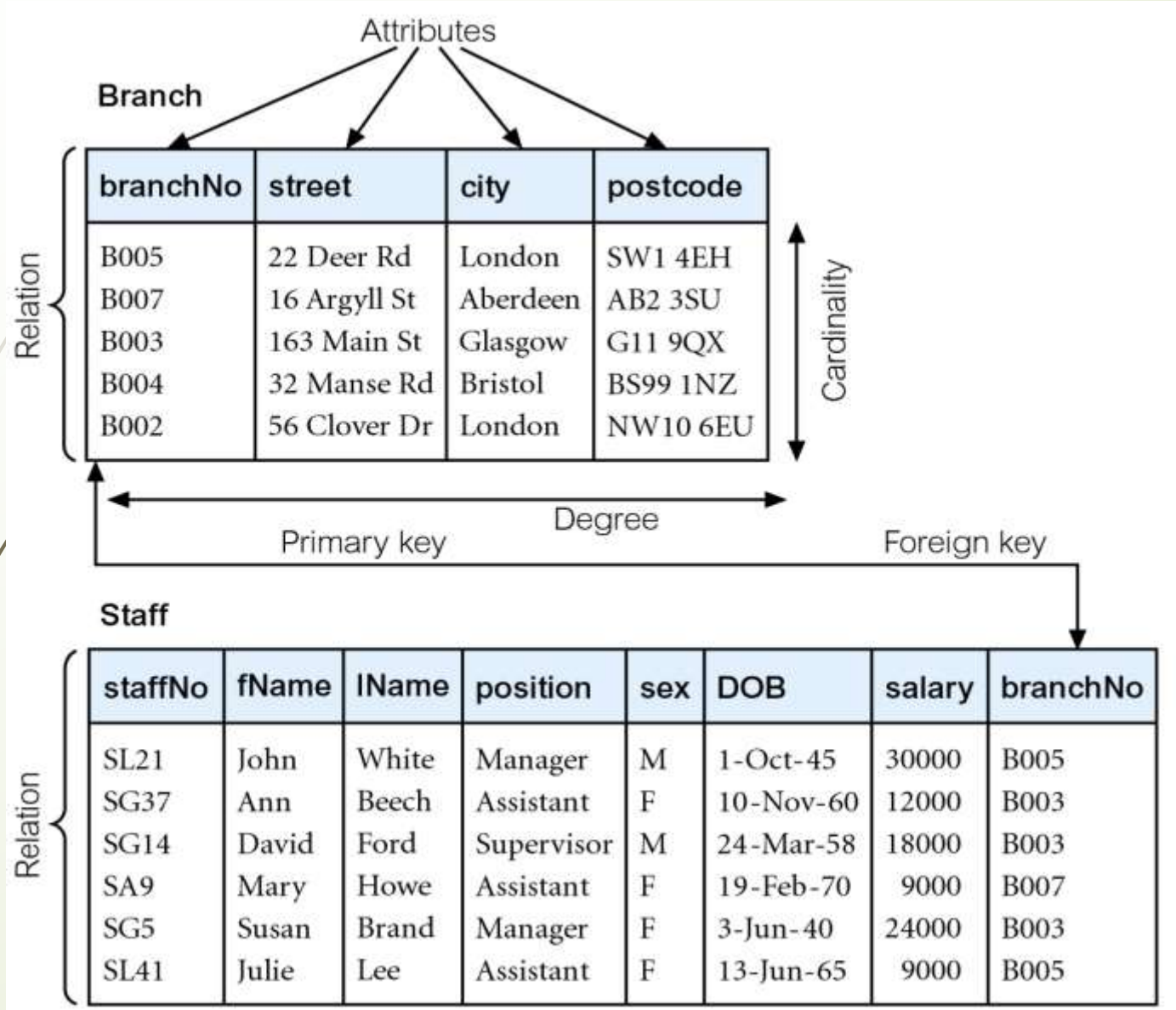
Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2018	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2018	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2019	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2018	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2019	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2018	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2019	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2018	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2018	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2019	503

Relational Table

- A relational table stores a collection of related entities. In this respect, the relational database table resembles a file, but there is a crucial difference between a table and a file: a table yields complete data and structural independence because it is a purely **logical** structure. How the data is **physically** stored in the database is of no concern to the user or the designer; the perception is what counts.

The Relational Model



- Data and relationships are represented by a collection of **tables**.
- Each **table** has a number of columns with unique names, e.g. *customer, account*.

The Relational Model

- **Advantages**
  - Structural independence
  - Improved conceptual simplicity
  - Easier database design, implementation, management, and use
  - Ad hoc query capability through the use of SQL
  - Powerful database management system

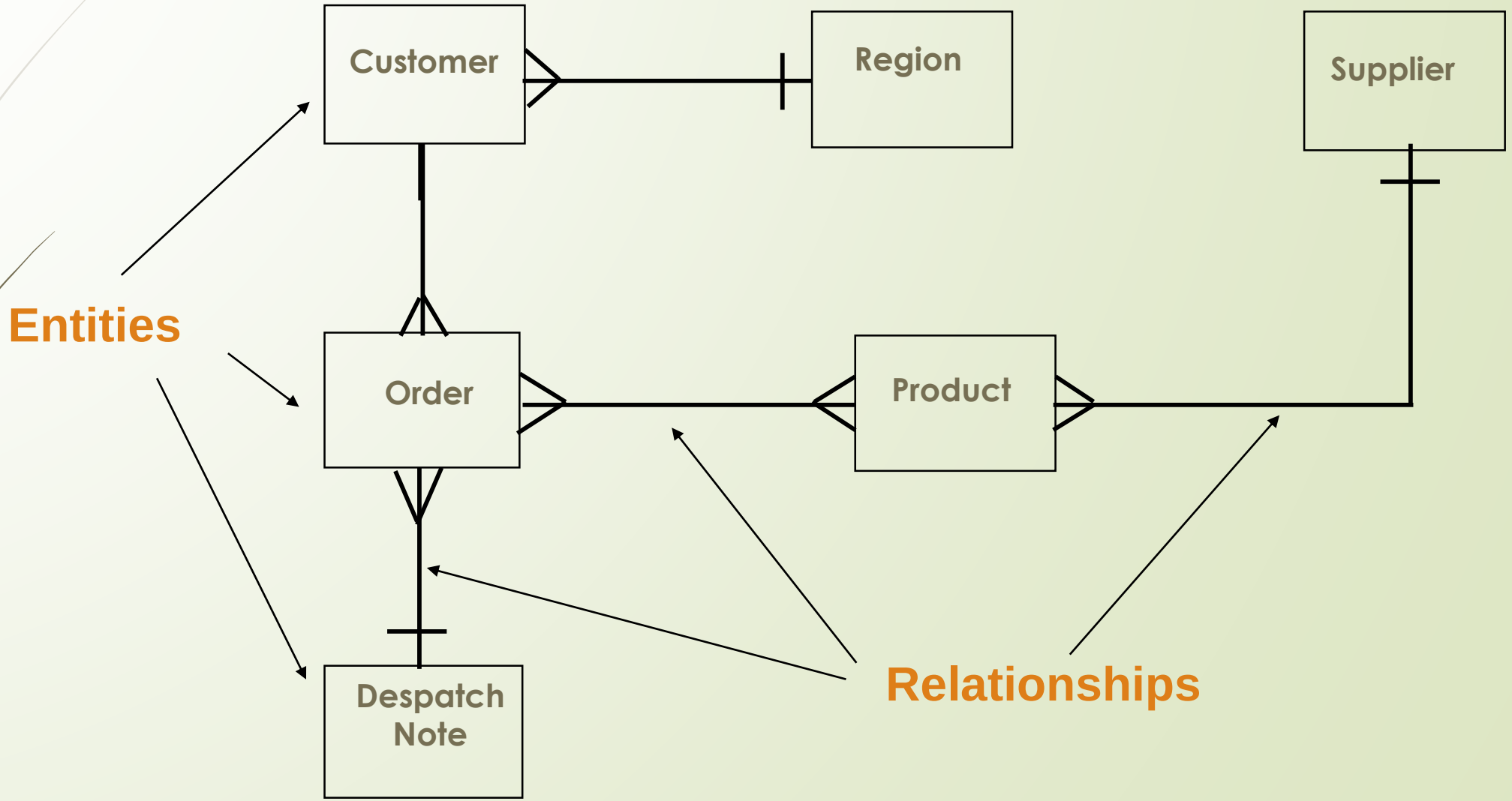
The Relational Model

- **Disadvantages**
  - Substantial hardware and system software overhead
  - Can facilitate poor design and implementation
  - May promote “islands of information” problems

The Entity Relationship Model

- Widely accepted and adapted graphical tool for data modeling
- Graphical representation of entities and their relationships in a database structure.
- A top-down approach to data modeling.

Building Blocks of Data Model



Entity

- Meaning. A class of persons, places, objects, events, or concepts about which we need to capture and store data (crudely, it is a *file* and described using a noun).
- Examples : customers, products, suppliers, employees, departments
  - Using singular or plural nouns is a matter of preference
- Notation - it is represented by a rectangle.

What Should an Entity Be?

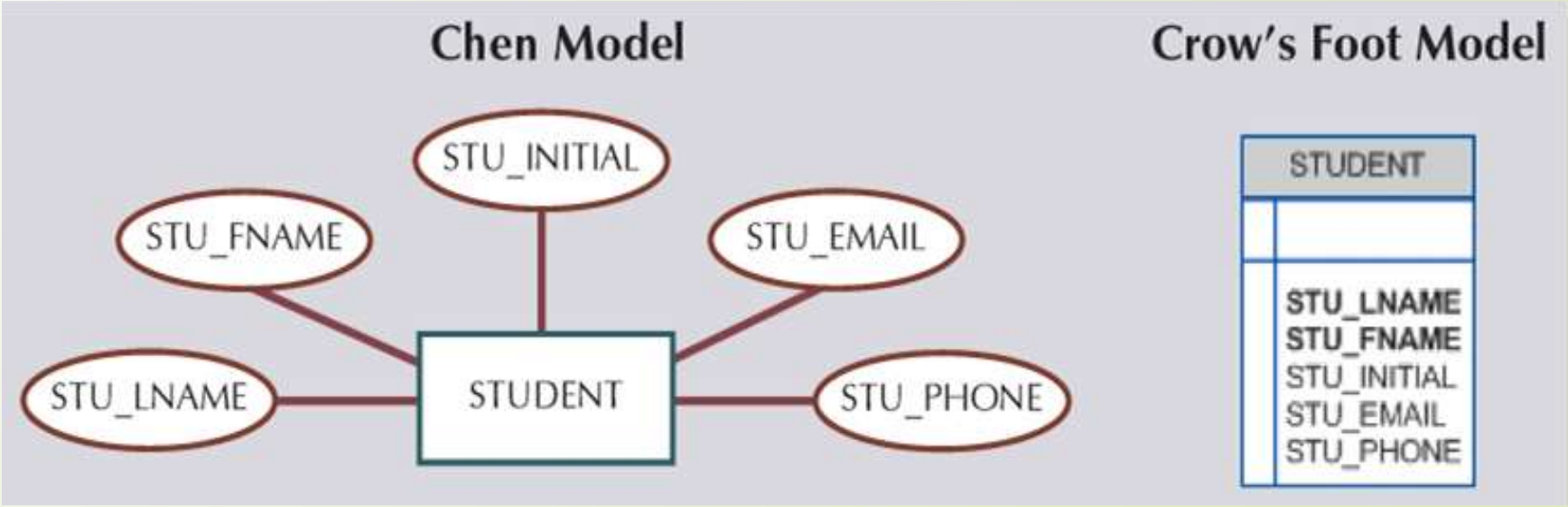
- SHOULD BE:
  - An object that will have many instances in the database
  - An object that will be composed of multiple attributes
  - An object that we are trying to model
- SHOULD NOT BE:
  - A user of the database system
  - An output of the database system (e.g. a report)

More Examples of Entities

- Persons : agency, contractor, customer, department, division, employee, instructor, student, supplier
- Places : sales region, building, room, branch office, campus
- Objects : book, machine, part, product, raw material, software license, software package, tool, vehicle model, vehicle
- Events : application, award, cancellation, class, flight, invoice, order, registration, renewal, requisition, reservation, sale, trip
- Concepts : account, block of time, bond, course, fund, qualification, stock

Attributes

- Meaning - is a characteristic, data field, or data element of an entity. There are many attributes for each entity.
- Example – Customer's attributes are name, address, telephone, email.
- Notation:

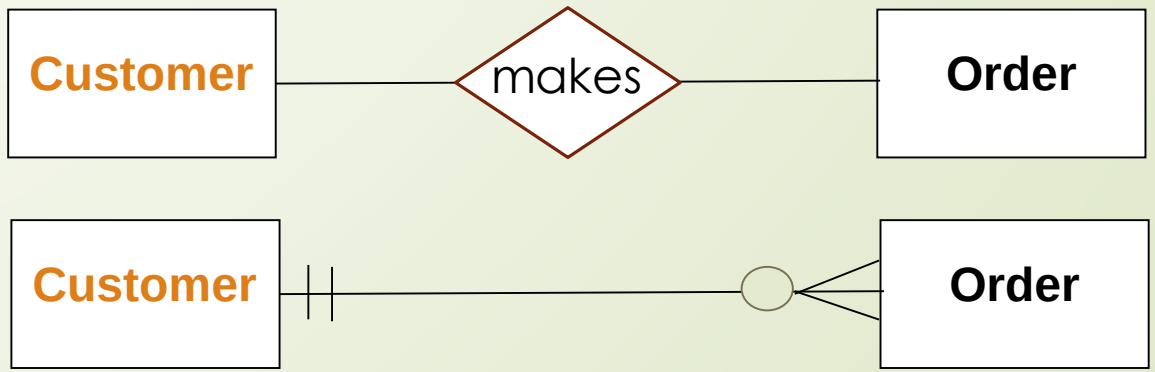


DBDL or Textual notation

STUDENT (STU_ID, STU_LNAME, STU_FNAME, STU_INITIAL, STU_EMAIL, STU_PHONE)

Relationship

- Meaning. The lines between the entities (boxes) representing relationships between them
- Example. Between CUSTOMER and ORDER
- Notation. The relationship is indicated by lines connecting the entities. There's a "crow's" foot or a diamond shaped box
- Description : A relationship is read in two directions (left and right). Example : Each customer places many orders, and each order is placed by only one customer.



Exercise for Students

- List 4 appropriate entities for each of the following systems :
  - Inventory System
  - Human Resource System
  - Purchasing System
- For each of the entities, list 4 attributes

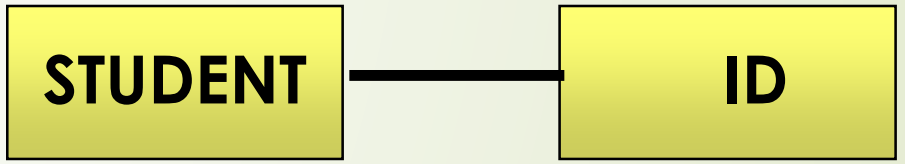
- Inventory System
  - Product (ProductID, Name, SKU, Quantity, Price) **SKU → Stock Keeping Unit
  - Warehouse (WarehouseID, Location, Capacity, Personnel)
  - Supplier (SupplierID, Name, ContactNo, DeliveryTime, ProductCatalogue)
  - Order (OrderID, OrderDate, Quantity, Status, DeliveryInformation)
- Human Resource System
  - Employee (EmployeeID, EmployeeName, ContactInformation, JobTitle, Salary)
  - Department (DepartmentID, DepartmentName, Manager, Budget, Goals)
  - Job Position (JobID, Title, JobDescription, SkillsRequired, SalaryRange)
  - Performance Review (PerformanceID, Date, PerformanceGoals, Feedback, Rating)
- Purchasing System
  - Vendor (VendorID, Name, Contact information, Product offerings, Lead time)
  - Purchase Order (PurchaseOrderID, Number, Date, Vendor, ItemsOrdered)
  - Budget (BudgetID, Amount, Department, Date, Status)
  - Shipment (ShippingID, Date, Quantity, Vendor, PurchaseOrderNumber)

Types of Relationships with Examples

ONE-TO-ONE:

1 ⟹ 1

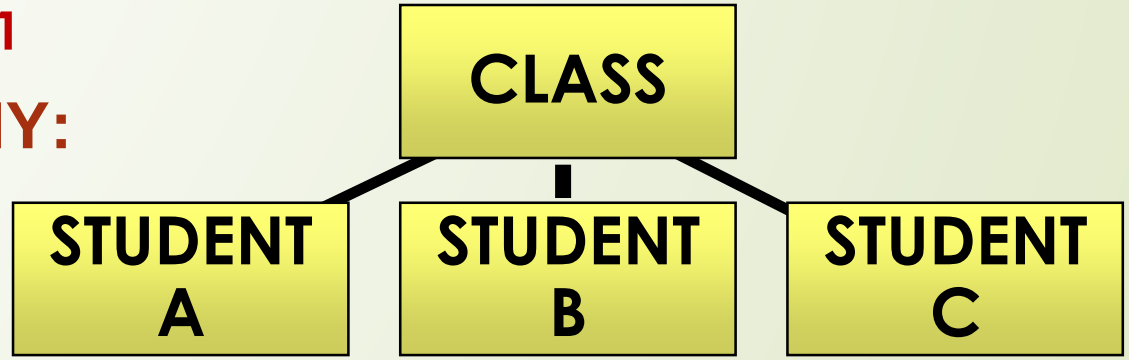
1 ⟸ 1



ONE-TO-MANY:

1 ⟹ M

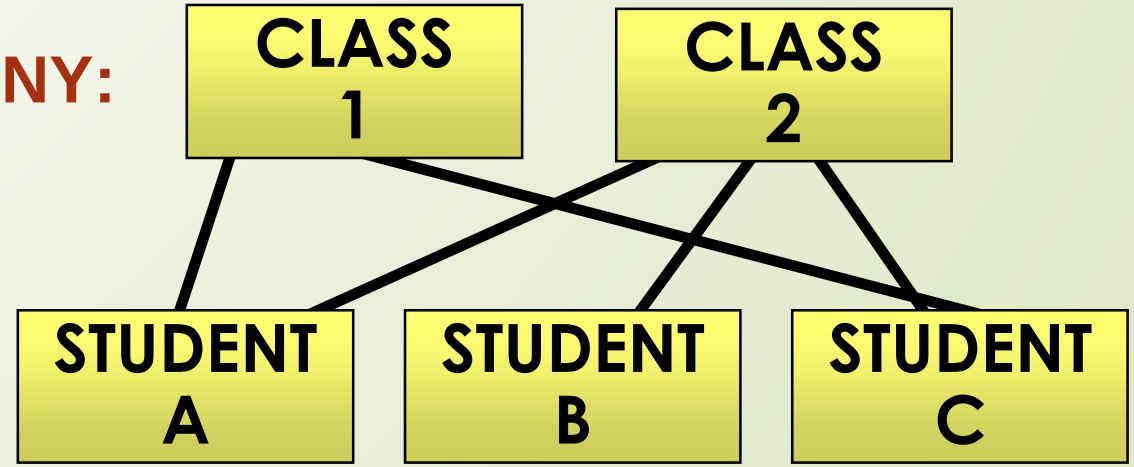
1 ⟸ 1



MANY-TO-MANY:

1 ⟹ M

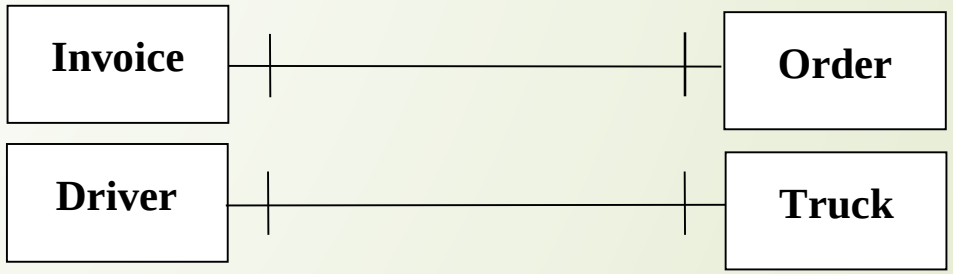
M ⟸ 1



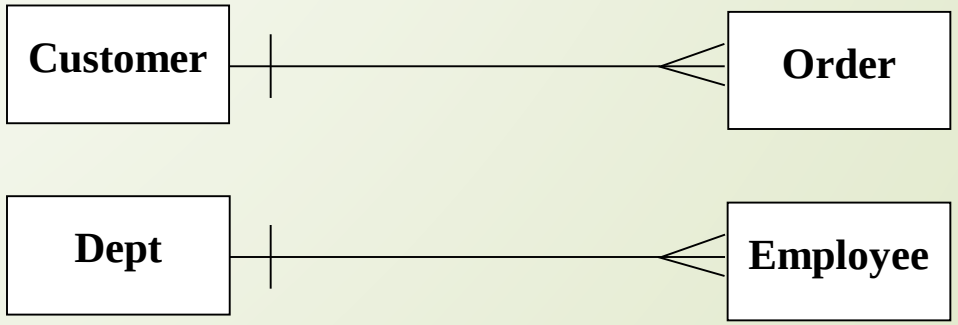
break down into 2 sets of one-to-many relationship

Types of Relationship

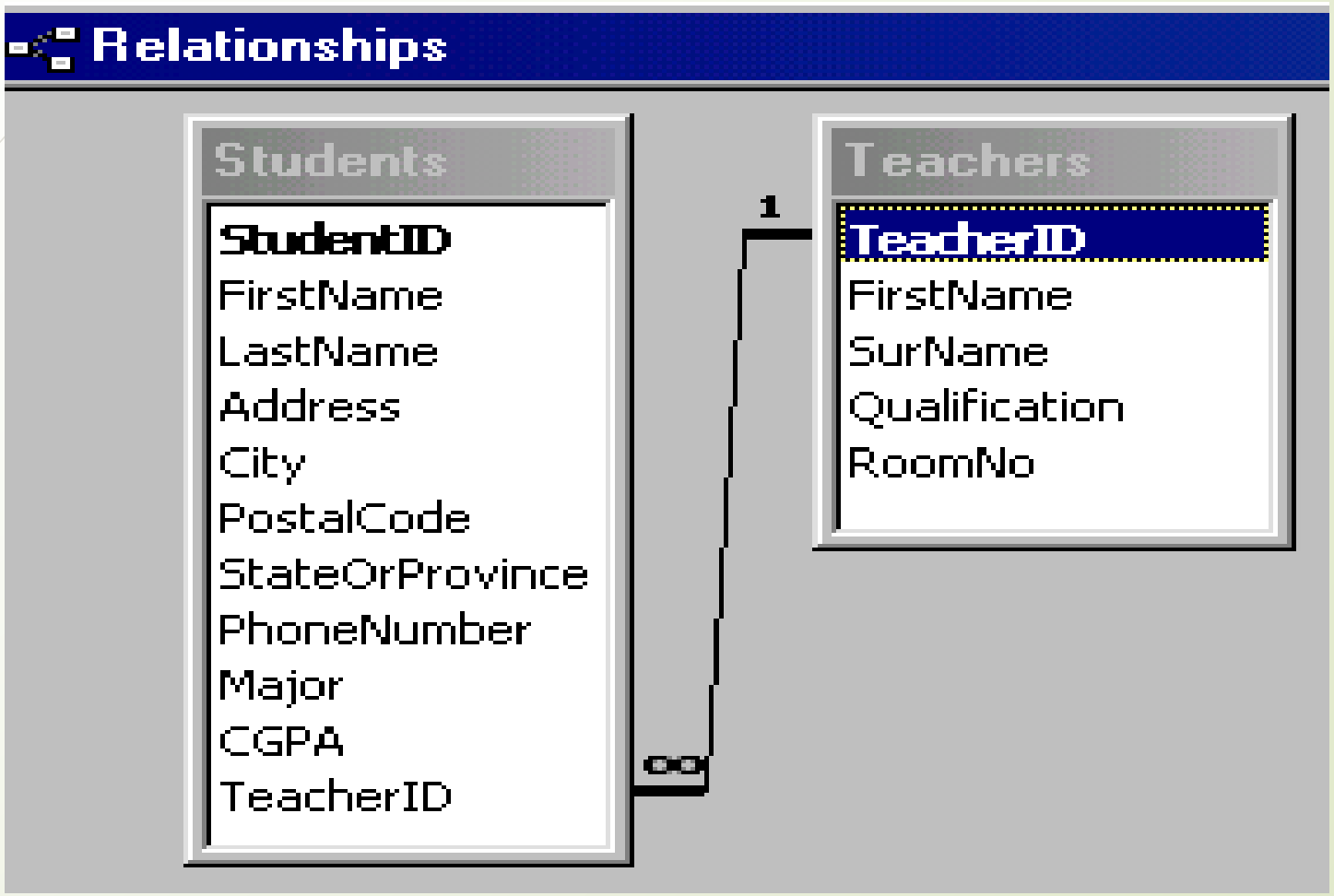
- **One-to-One (1 : 1)**



- **One-to-Many (1 : N)**



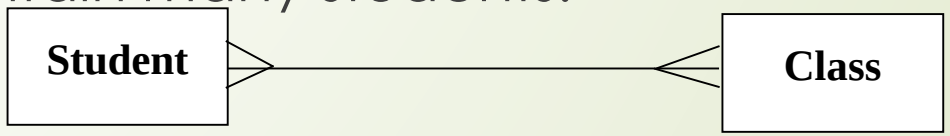
Example of One-to-Many Relationship (in Access)



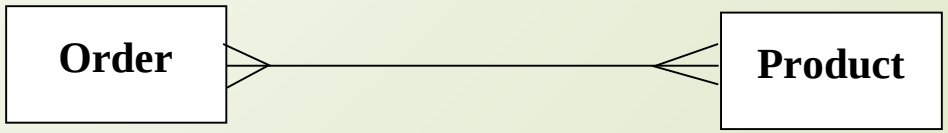
Example: A teacher can teach many students, and each student have one class teacher.

Many-to-Many Relationship

- Meaning - When an entry can occur more than once on both sides of the relationship.
- Examples
  - Each student can take many classes, and each class can contain many students.

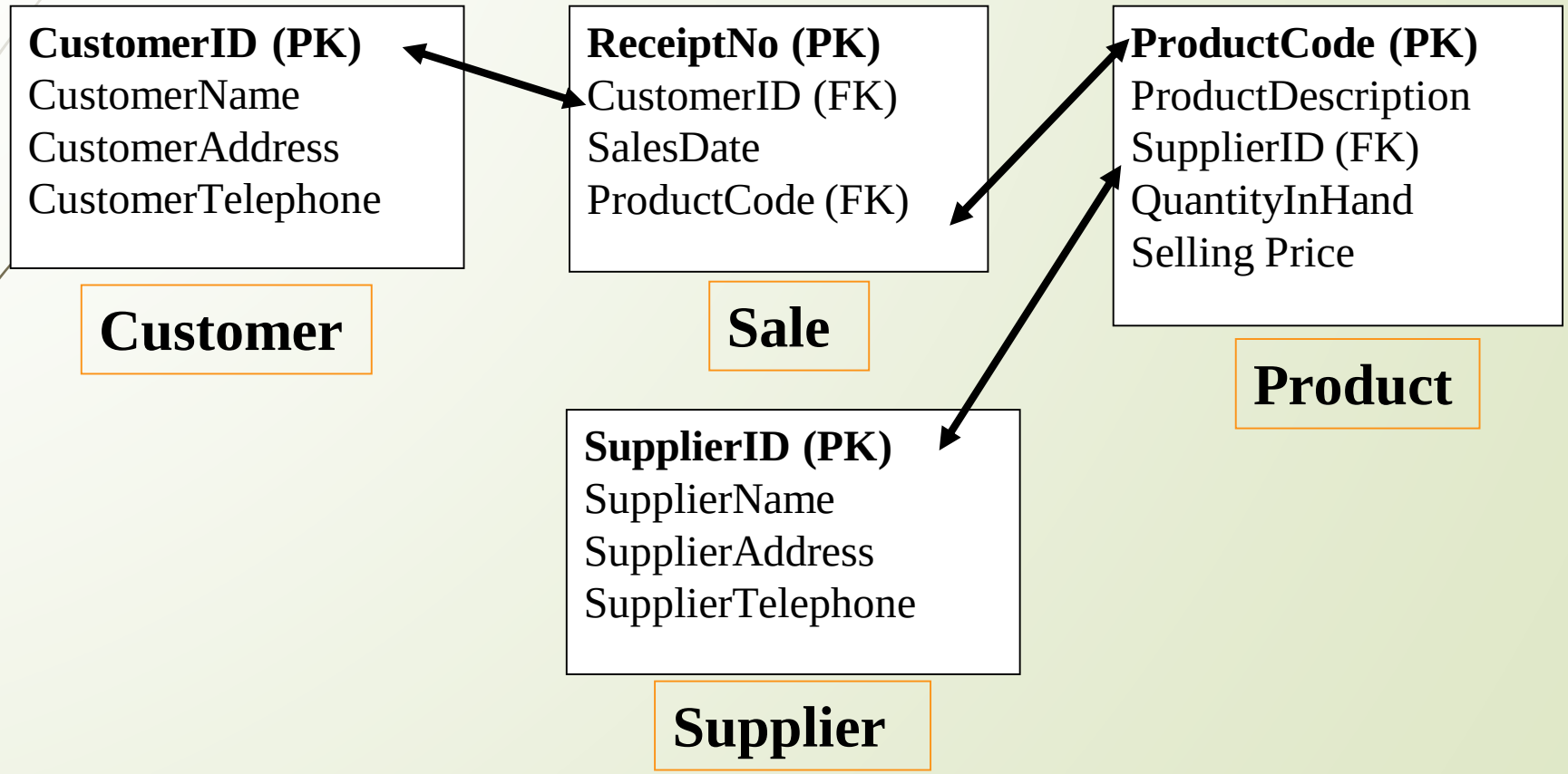


  - Each order contains many products, and each product can be listed in many orders.



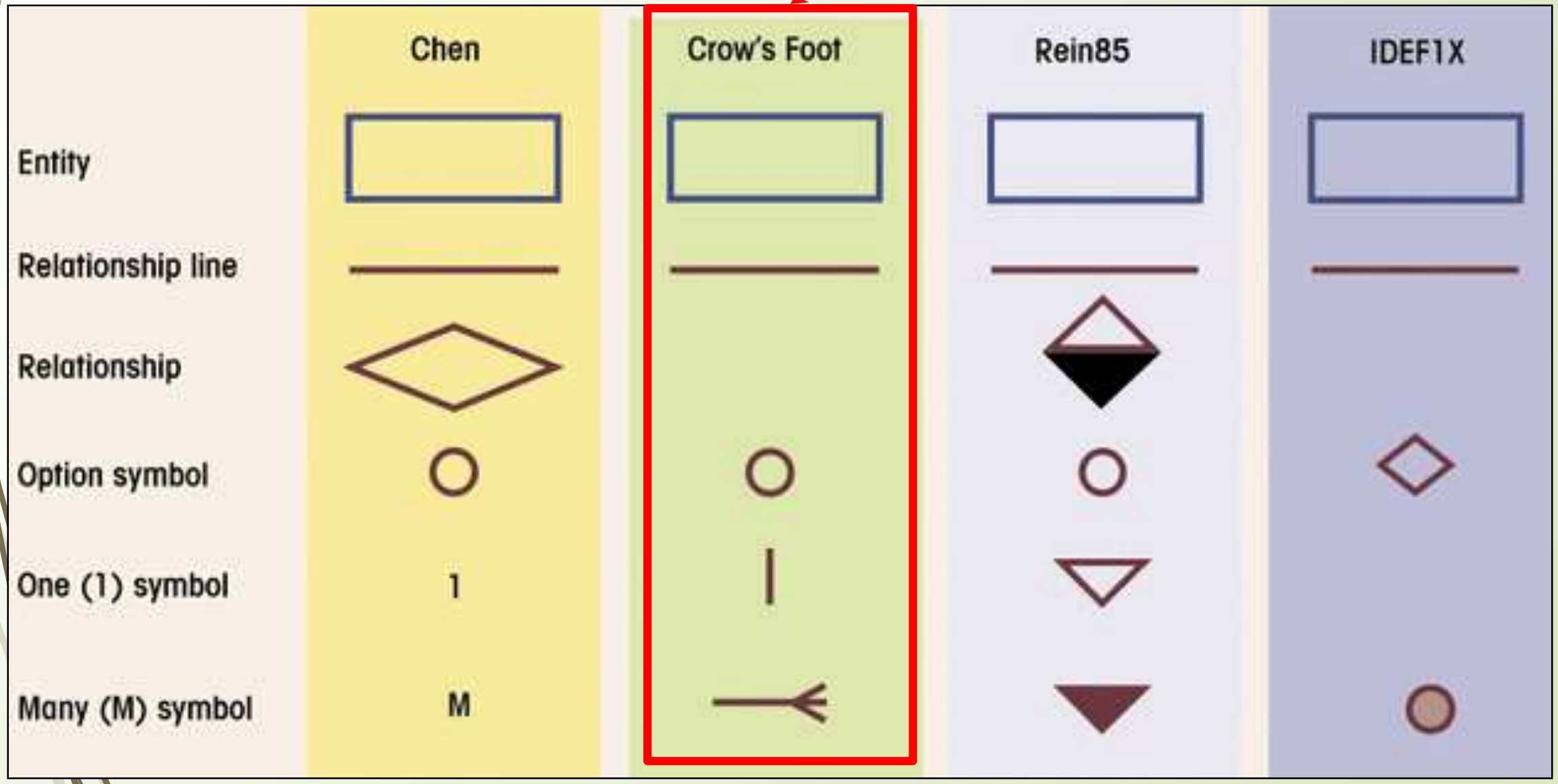
Exercise for Students

? Identify the entities, attributes and relationships in the following diagram.

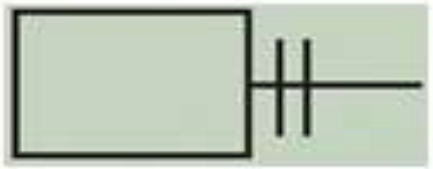
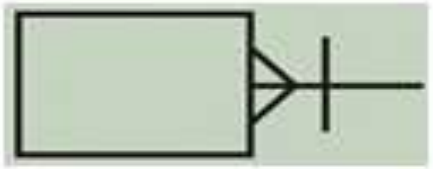
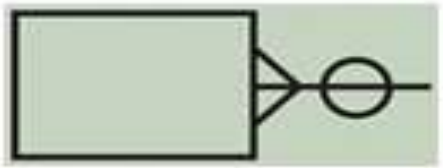
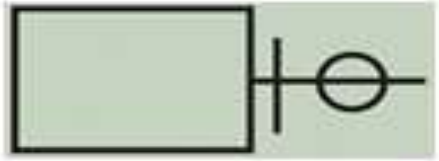


ERD Notation

We only use this for examination and assignment submission



Crow's Foot Notation

Symbols	Meaning
	One and only one
	One or many
	Zero, or one, or many
	Zero, or one

Cardinality

Expresses minimum and maximum number of entity occurrences associated with one occurrence of related entity

Established by very concise statements known as business rules

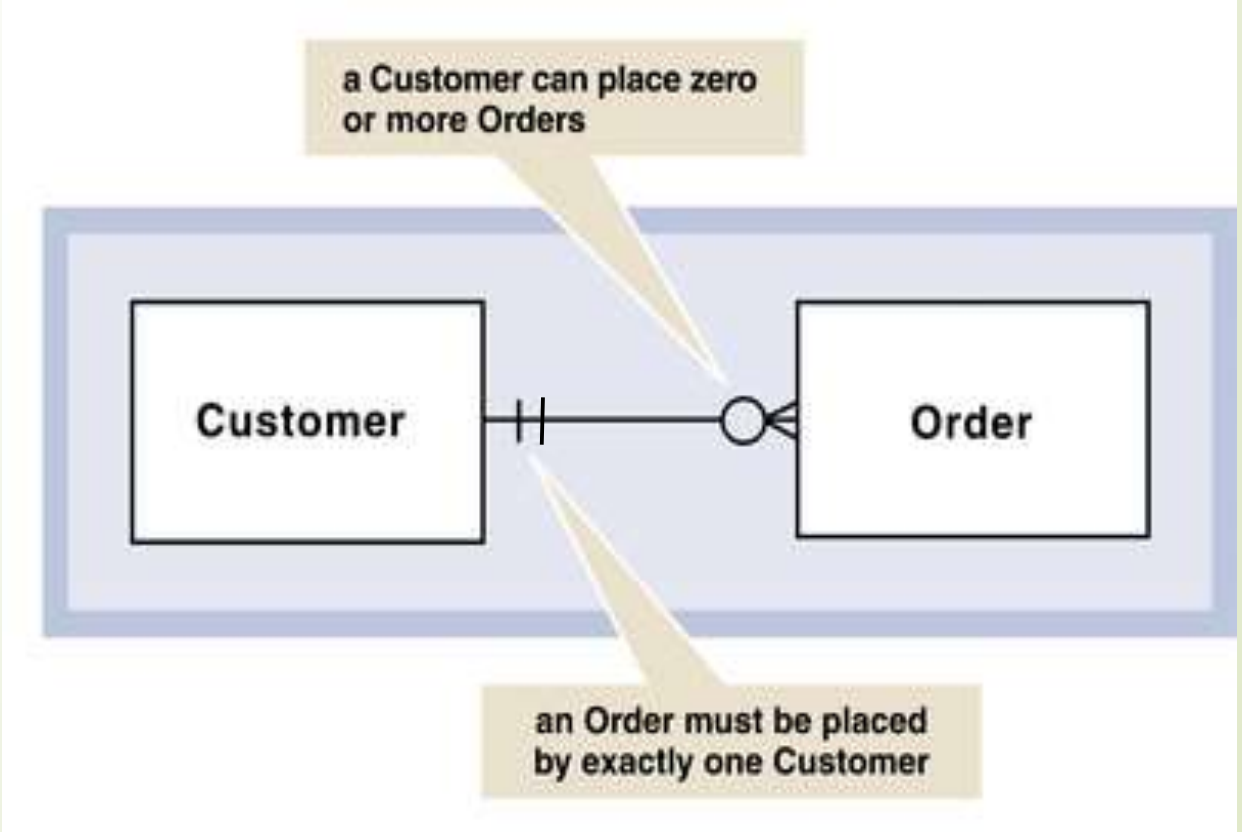
Important Note:

These are the only symbols to be used in tests and exam questions

From this slide onwards, these correct Crow's foot notation should be used

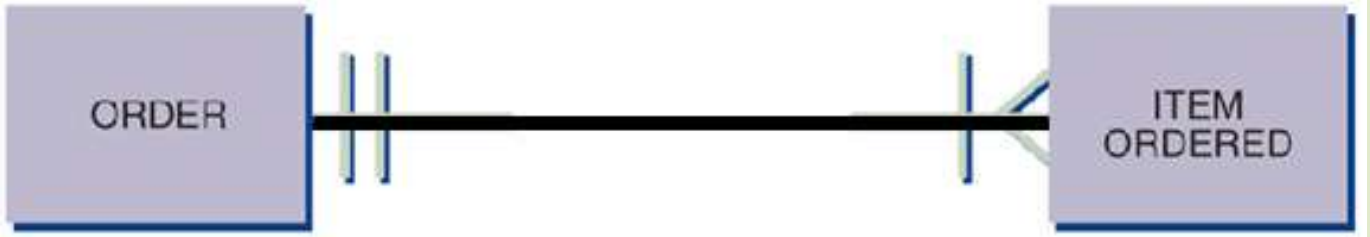
Crow's Foot Notation

- Describes the cardinality of relationship between two entities
- Shows how instances of one entity relate to instances of another entity.
- Because all relationships are bidirectional, cardinality must be defined in both directions for every relationship.



Crow's Foot Notation

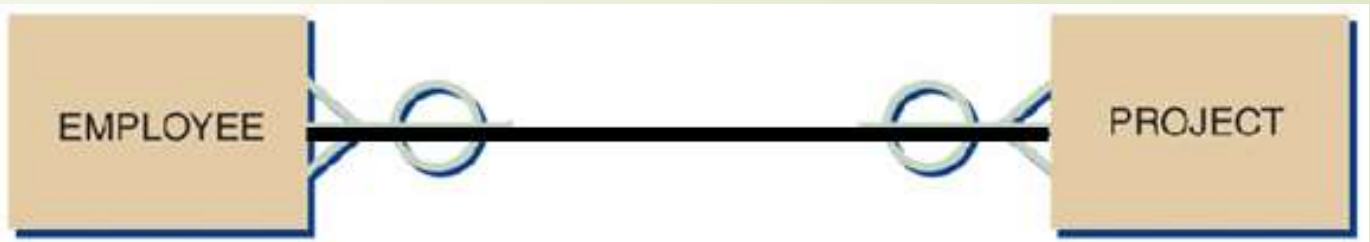
One and only one CUSTOMER can place anywhere from zero to many of the ORDER entity.



One and only one ORDER can include one ITEM ORDERED or many.

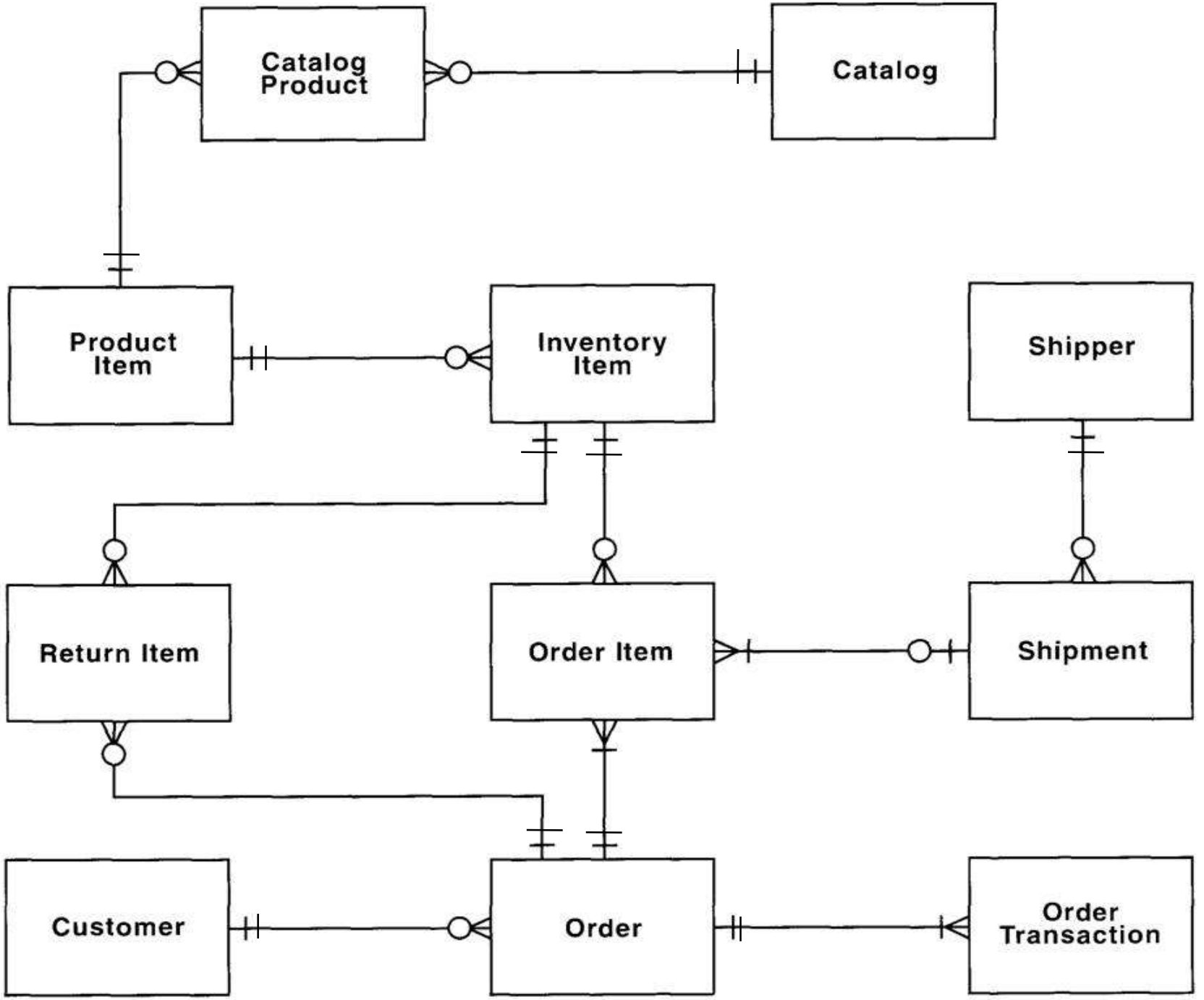
One and only one EMPLOYEE can have one SPOUSE or NONE.

Ex:
Rewrite the relationships in both directions

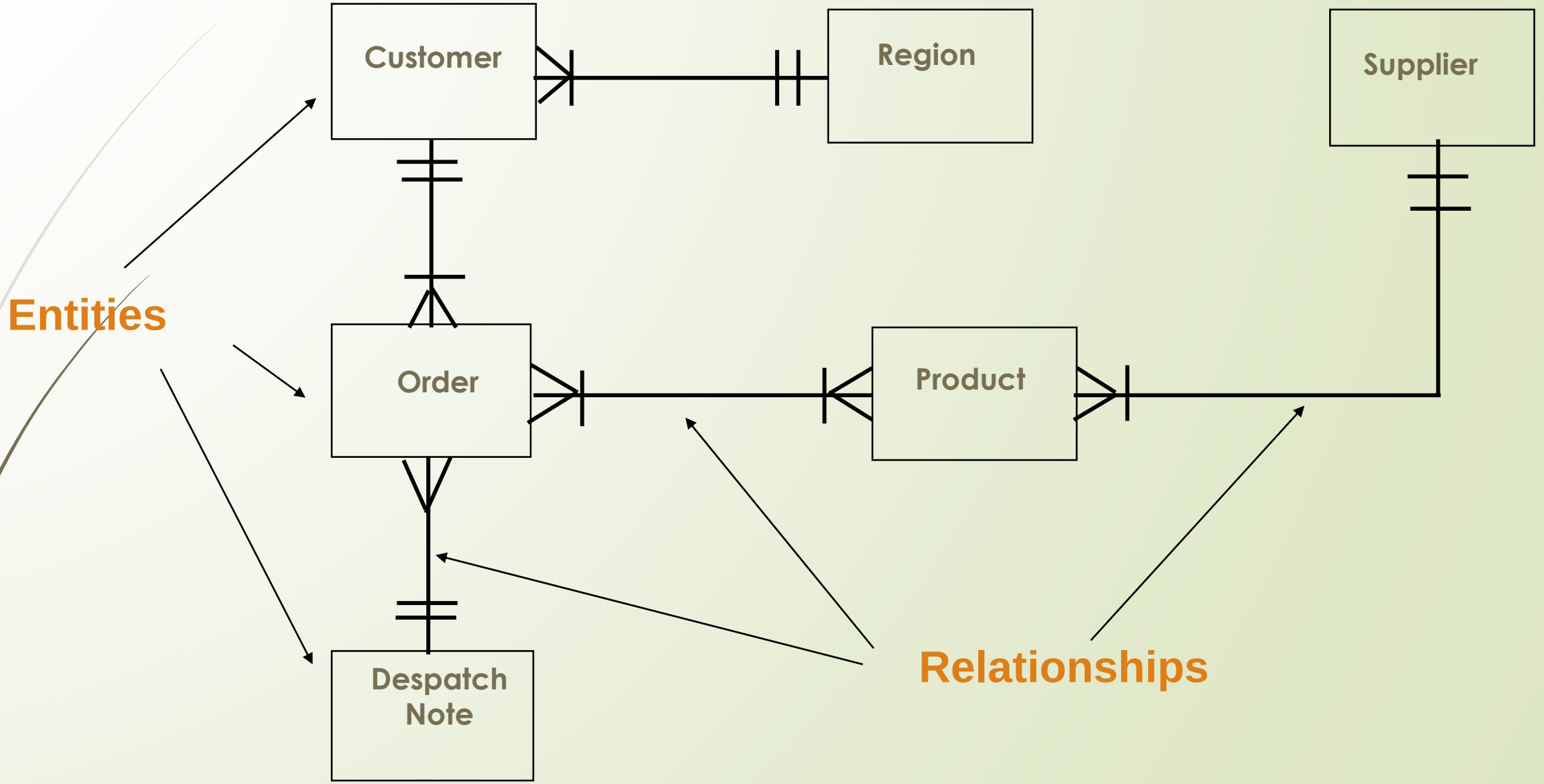


One EMPLOYEE, or many employees, or none, can be assigned to one PROJECT, or many projects, or none.

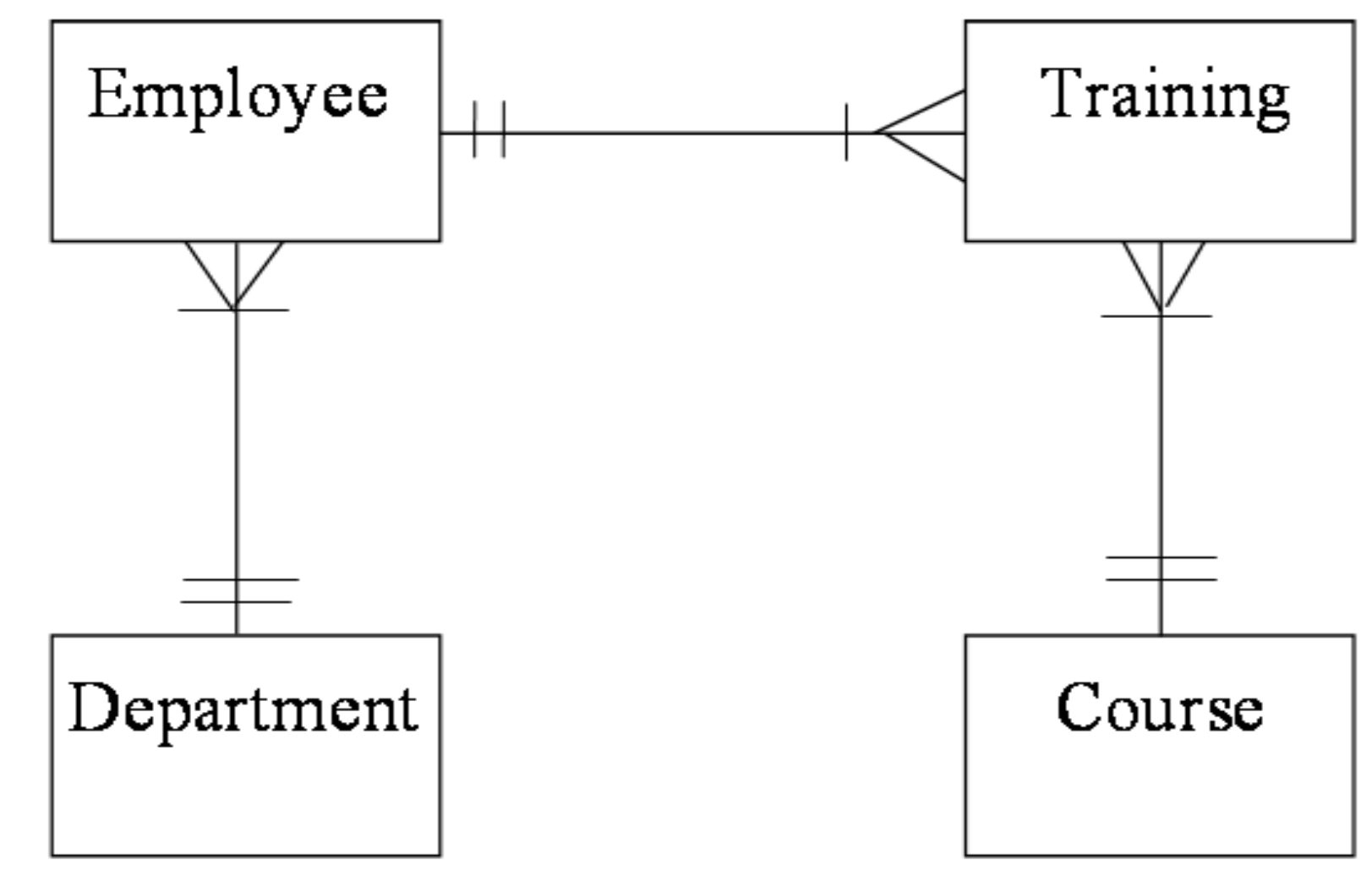
Example 1 of ERD



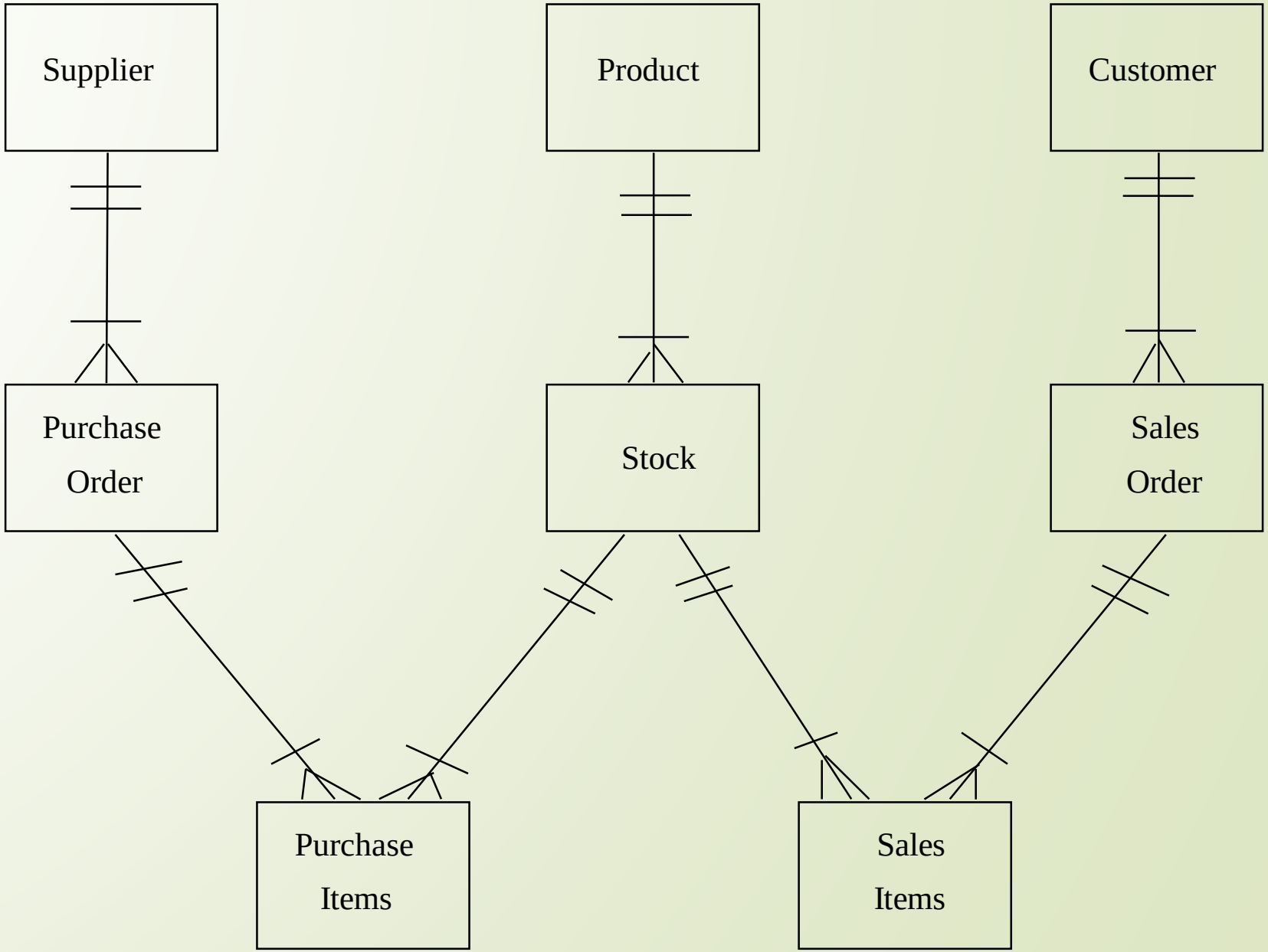
Example 2 of ERD



Example 3 of ERD



Example 4 of ERD



Resolving Many-to-Many Relationship

Associative Entities

- Also known as weak entities.
- Used to resolve M:N relationships.
- Composed of primary keys of each of the entities to be connected.
- By introducing an 'associative' entity to form a new 2 pairs of one-to-many relationships.
- M: N □ 1: M

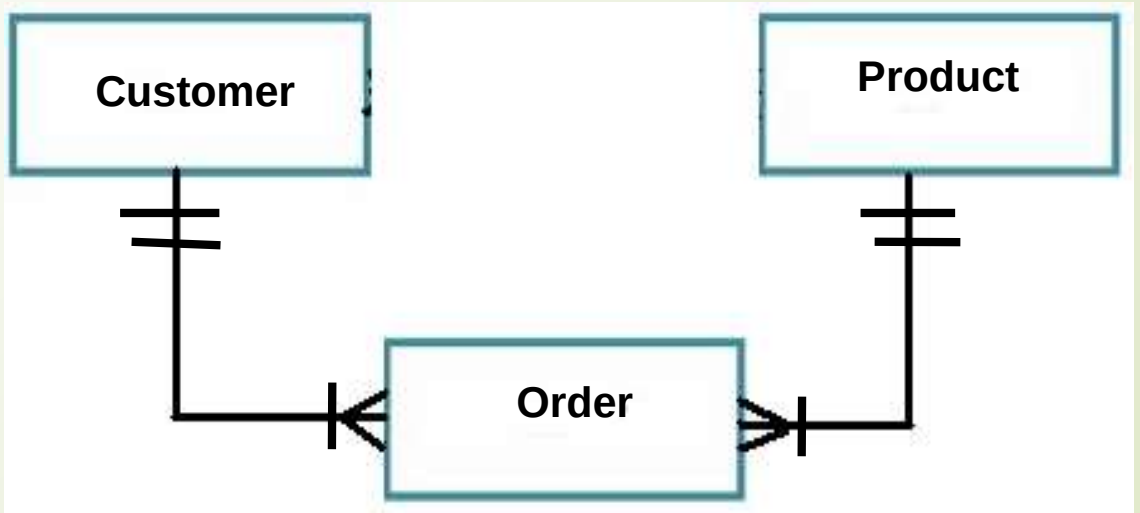
  M: 1

Example 1 – Associative Entity

? By introducing an 'associative' entity to form a new 2 pairs of one-to-many relationships.

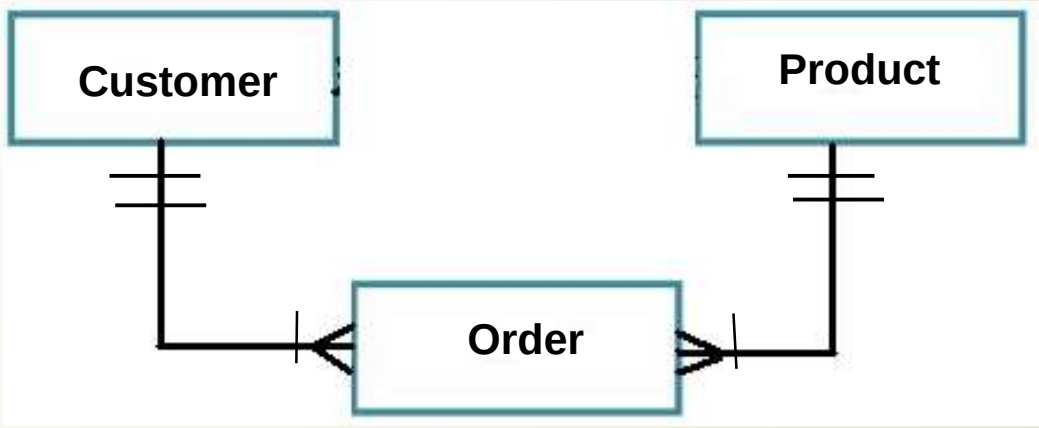
Before

After



Associative Entity

Example - Attributes



Reconsider relationship between Product and Order!!! **M:N**

CUSTOMER

CustomerID (PK)
LastName
FirstName
Address
PostCode
City
Phone

ORDER

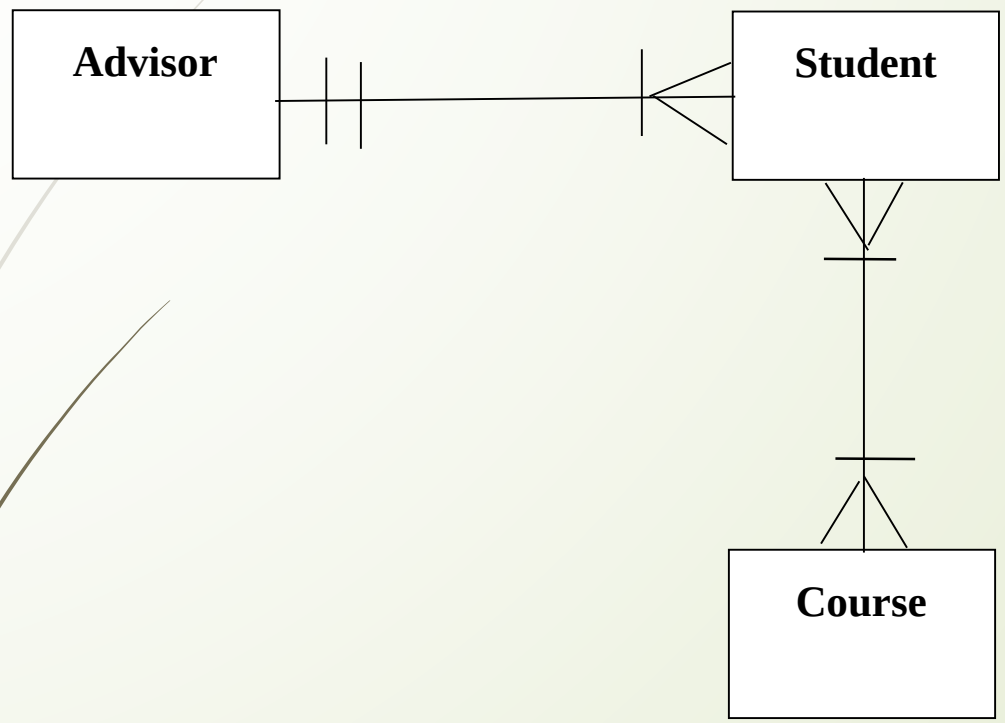
CustomerID
ProductID
PurchaseDate
Amount

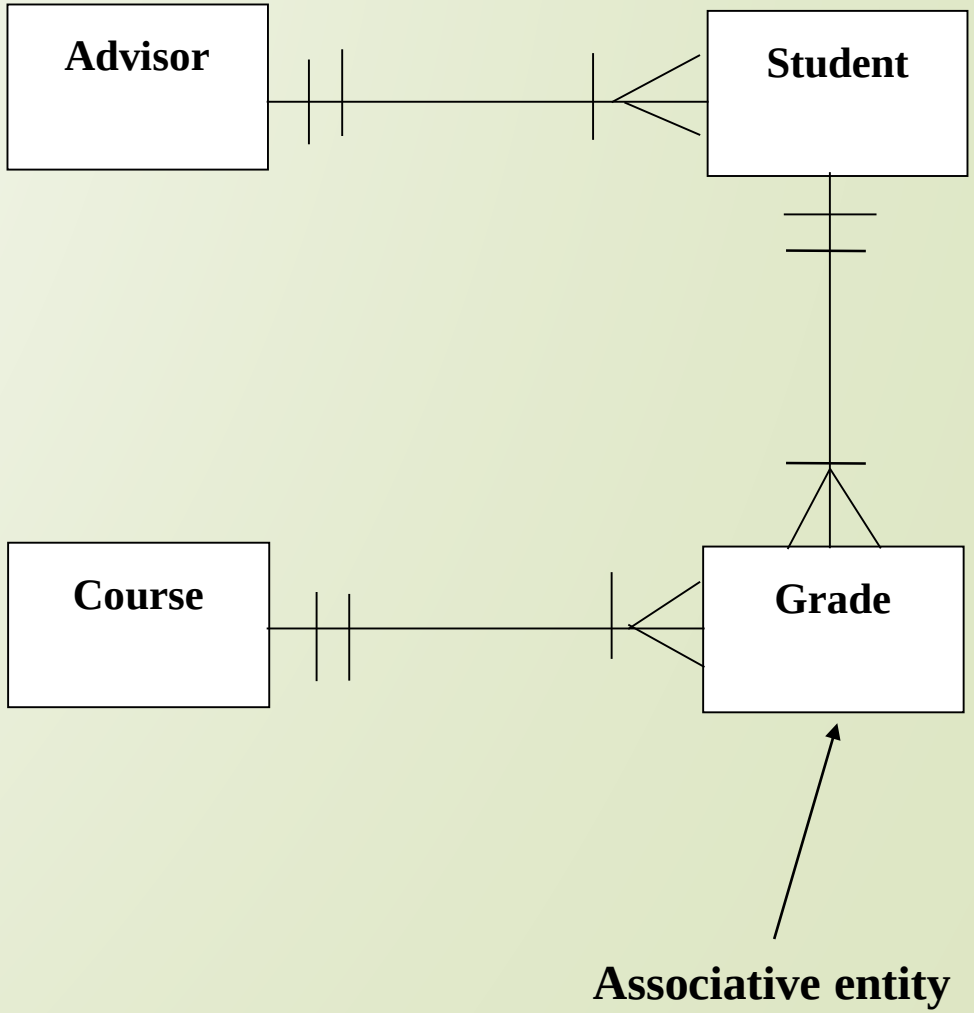
PRODUCT

ProductID (PK)
Name
UnitPrice

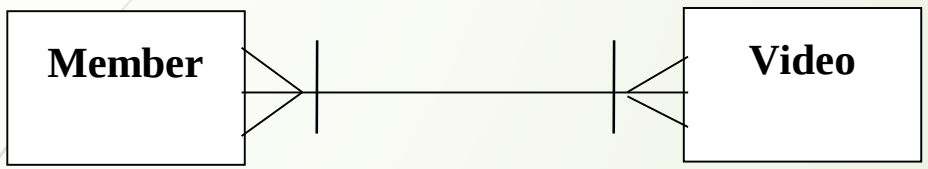
Identify the keys in ORDER entity

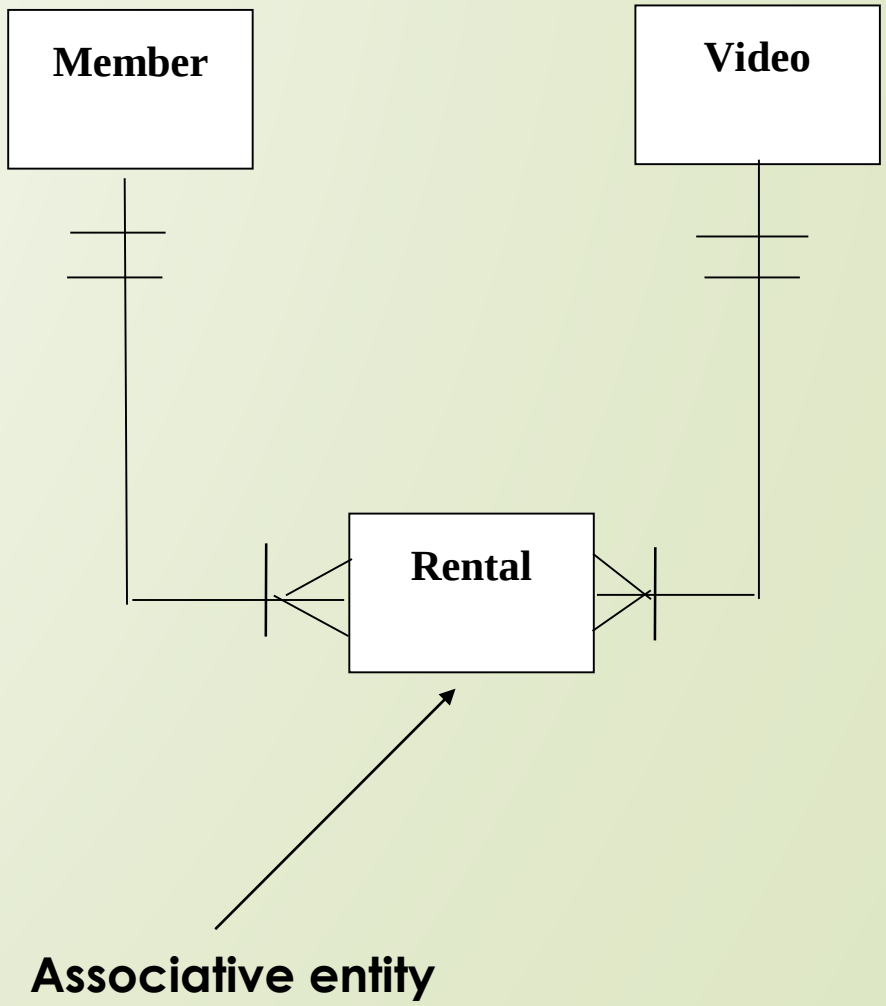
Example 2 - Resolving M:N Relationship



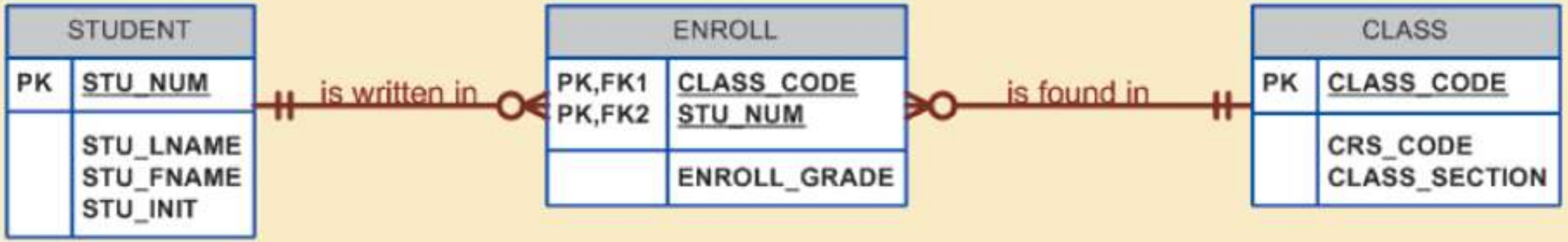


Example 3 – Resolving M:N Relationship





Example 4 – Resolving M:N Relationship



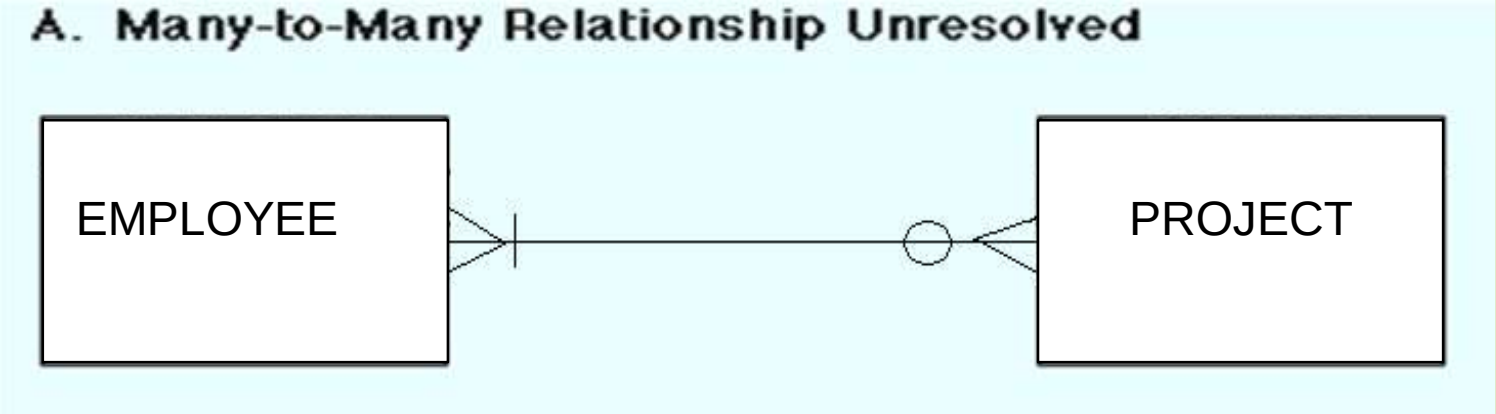
STU_NUM	STU_LNAME	STU_FNAME	STU_INIT
321452	Bowser	William	C
324257	Smithson	Anne	K
324258	Brewer	Juliette	
324269	Oblonski	Walter	H
324273	Smith	John	D
324274	Katinga	Raphael	P
324291	Robertson	Gerald	T
324299	Smith	John	B

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

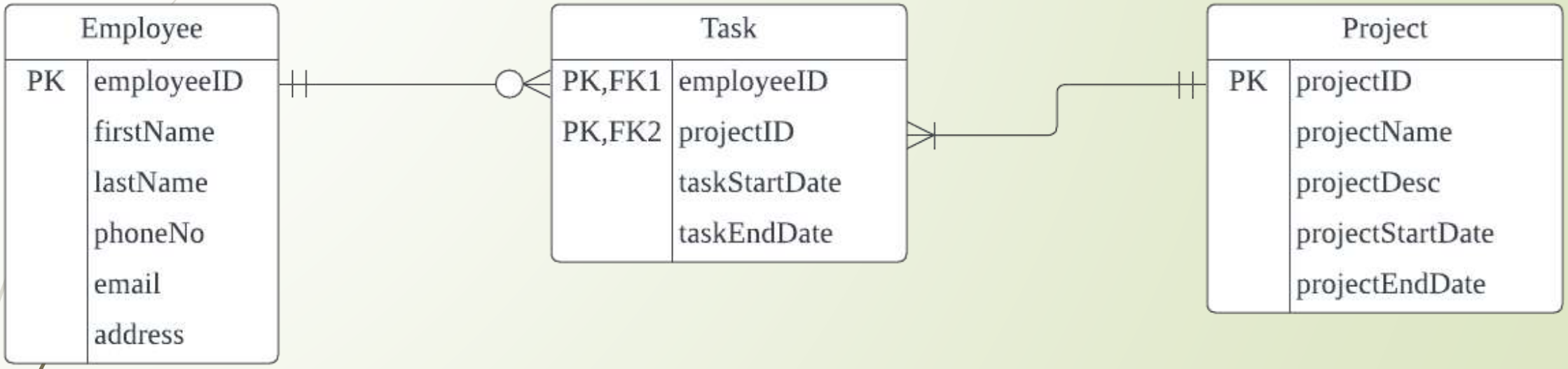
CLASS_CODE	CRS_CODE	CLASS_SECTION
10012	ACCT-211	1
10013	ACCT-211	2
10014	ACCT-211	3
10015	ACCT-212	1
10016	ACCT-212	2
10017	CIS-220	1
10018	CIS-220	2
10019	CIS-220	3
10020	CIS-420	1
10021	QM-261	1
10022	QM-261	2
10023	QM-362	1
10024	QM-362	2
10025	MATH-243	1

Exercise for Students

- We normally do NOT implement M:N relationships in a relational database environment because they create many redundancies.
- They must be split into 2 pairs of 1:M relationships by creating an associative entity.
- Question - Each employee can be assigned to many projects, and each project can be done by many employees (below).
  - Resolve the many-to-many relationship
  - List some attributes for each entity



Answer


Employee
PK
employeeID
firstName
lastName
phoneNo
email
address
Task
PK,FK1
employeeID
PK,FK2
projectID
taskStartDate
taskEndDate
Project
PK
projectID
projectName
projectDesc
projectStartDate
projectEndDate

Exercise for Students

Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

Database name: CH2_MUSEUM

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Itero	Julio	G

link

Table name: PAINTING
Primary key: PAINTER_NUM
Foreign key: PAINTER_NUM

PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

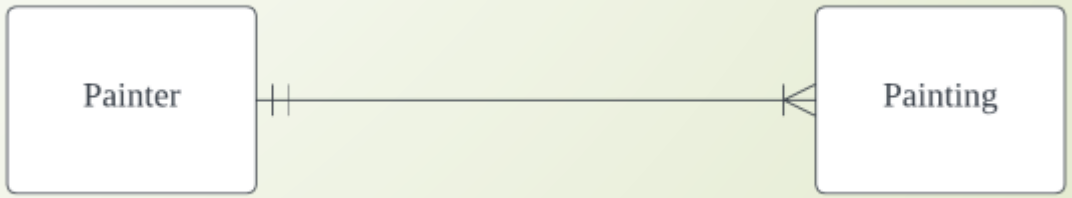
Task: Describe the relationship between the painter and the painting

Answer

Business Rule:

One painter is allow to paint for many painting. One painting is assign to only one painter.

Entity Relationship Diagram:



Exercise for Students

- Resolve the following many-to-many relationship

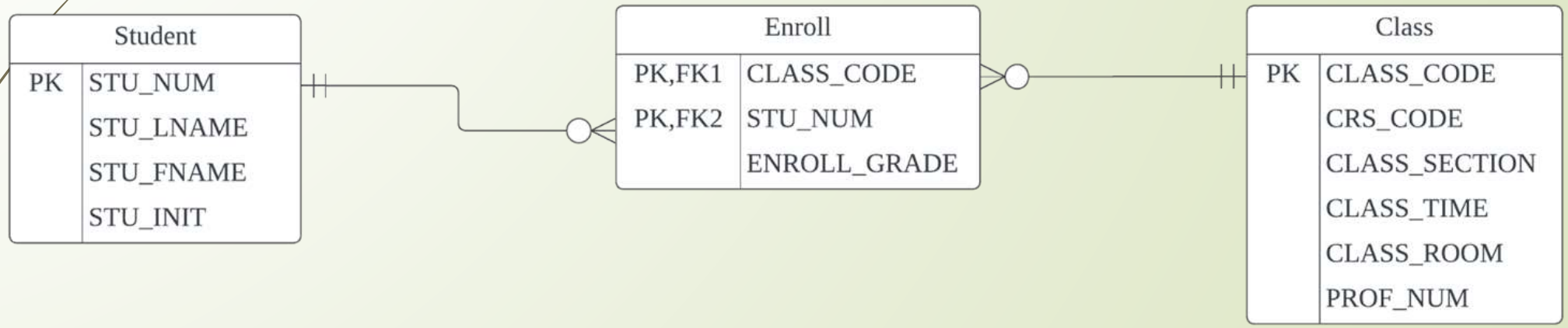
Table name: STUDENT_FIG2_24

STU_NUM	STU_LNAME	CLASS_CODE
321452	Bowser	10014
321452	Bowser	10018
321452	Bowser	10021
324257	Smithson	10014
324257	Smithson	10018
324257	Smithson	10021

Table name: CLASS_FIG2_24

CLASS_CODE	STU_NUM	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	321452	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10014	324257	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	321452	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10018	324257	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	321452	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10021	324257	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

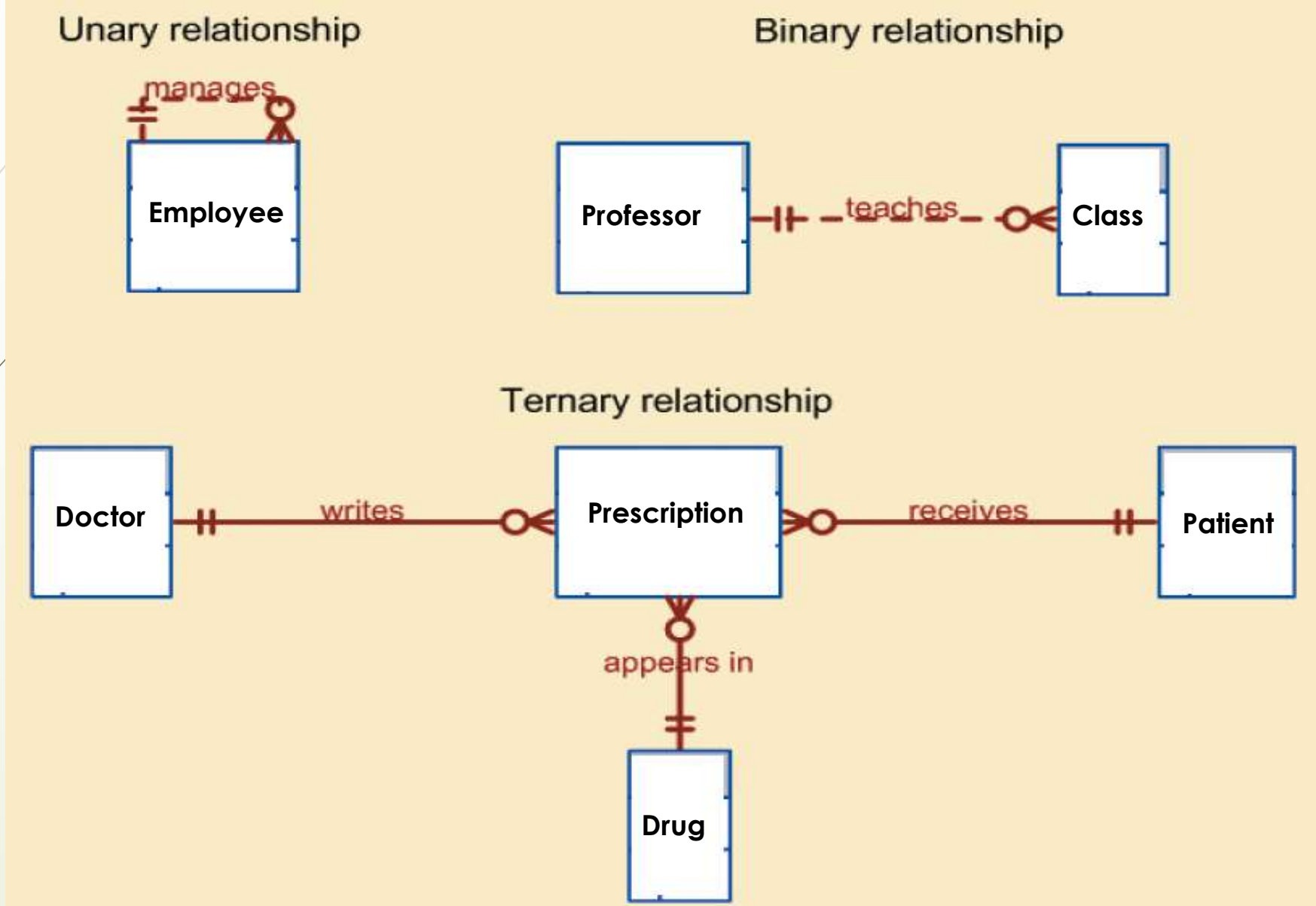
Answer for Students



Relationship Degree

- Indicates number of entities or participants associated with a relationship
- Unary relationship
  - Association is maintained within single entity
- Binary relationship
  - Two entities are associated
- Ternary relationship
  - Three entities are associated

Relationship Degree



Example of Ternary Relationship

Database name: Ch04_Clinic

Table name: DRUG

DRUG_CODE	DRUG_NAME	DRUG_PRICE
AF15	Afgapan-15	25.00
AF25	Afgapan-25	35.00
DRO	Droalene Chloride	111.89
DRZ	Druzocholar Cryptolene	18.99
KO15	Koliabar Oxyhexalene	65.75
OLE	Oleander-Drizapan	123.95
TRYP	Tryptolac Heptadimetric	79.45

Table name: PATIENT

PAT_NUM	PAT_TITLE	PAT_LNAME	PAT_FNAME	PAT_INITIAL	PAT_DOB	PAT_AREACODE	PAT_PHONE
100	Mr.	Kolmycz	George	D	15-Jun-1942	615	324-5456
101	Ms.	Lewis	Rhonda	G	19-Mar-2005	615	324-4472
102	Mr.	Vandam	Rhett		14-Nov-1958	901	675-8993
103	Ms.	Jones	Anne	M	16-Oct-1974	615	898-3456
104	Mr.	Lange	John	P	08-Nov-1971	901	504-4430
105	Mr.	Williams	Robert	D	14-Mar-1975	615	890-3220
106	Mrs.	Smith	Jeanine	K	12-Feb-2003	615	324-7883
107	Mr.	Diante	Jorge	D	21-Aug-1974	615	890-4567
108	Mr.	Wiesenbach	Paul	R	14-Feb-1966	615	897-4358
109	Mr.	Smith	George	K	18-Jun-1961	901	504-3339
110	Mrs.	Genkazi	Leighla	W	19-May-1970	901	569-0093
111	Mr.	Washington	Rupert	E	03-Jan-1966	615	890-4925
112	Mr.	Johnson	Edward	E	14-May-1961	615	898-4387
113	Ms.	Smythe	Melanie	P	15-Sep-1970	615	324-9006
114	Ms.	Brandon	Marie	G	02-Nov-1932	901	882-0845
115	Mrs.	Saranda	Hermine	R	25-Jul-1972	615	324-5505
116	Mr.	Smith	George	A	08-Nov-1965	615	890-2984

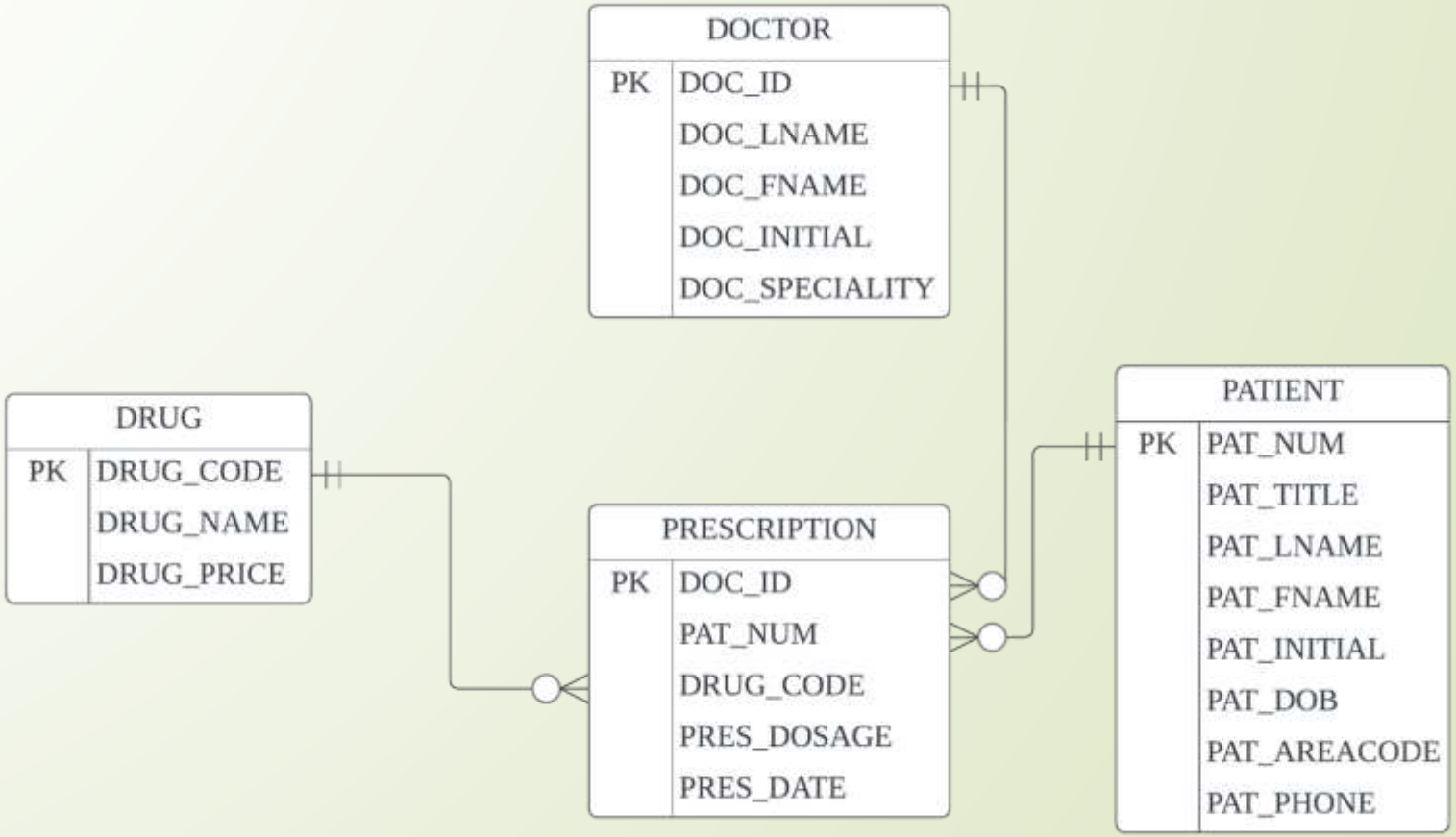
Table name: DOCTOR

DOC_ID	DOC_LNAME	DOC_FNAME	DOC_INITIAL	DOC_SPECIALTY
29827	Sanchez	Julio	J	Dermatology
32445	Jorgensen	Annelise	G	Neurology
33456	Korenski	Anatoly	A	Urology
33989	LeGrande	George		Pediatrics
34409	Washington	Dennis	F	Orthopaedics
36221	McPherson	Katye	H	Dermatology
36712	Dreifag	Herman	G	Psychiatry
38995	Minh	Tran		Neurology
40004	Chin	Ming	D	Orthopaedics
40028	Feinstein	Denise	L	Gynecology

Table name: PRESCRIPTION

DOC_ID	PAT_NUM	DRUG_CODE	PRES_DOSAGE	PRES_DATE
32445	102	DRZ	2 tablets every four hours -- 50 tablets total	12-Nov-07
32445	113	OLE	1 teaspoon with each meal -- 250 ml total	14-Nov-07
34409	101	KO15	1 tablet every six hours -- 30 tablets total	14-Nov-07
36221	109	DRO	2 tablets with every meal -- 60 tablets total	14-Nov-07
38995	107	KO15	1 tablet every six hours -- 30 tablets total	14-Nov-07

Example


DOCTOR
PK
DOC_ID
DOC_LNAME
DOC_FNAME
DOC_INITIAL
DOC_SPECIALITY
DRUG
PK
DRUG_CODE
DRUG_NAME
DRUG_PRICE
PRESCRIPTION
PK
DOC_ID
PAT_NUM
DRUG_CODE
PRES_DOSAGE
PRES_DATE
PATIENT
PK
PAT_NUM
PAT_TITLE
PAT_LNAME
PAT_FNAME
PAT_INITIAL
PAT_DOB
PAT_AREACODE
PAT_PHONE

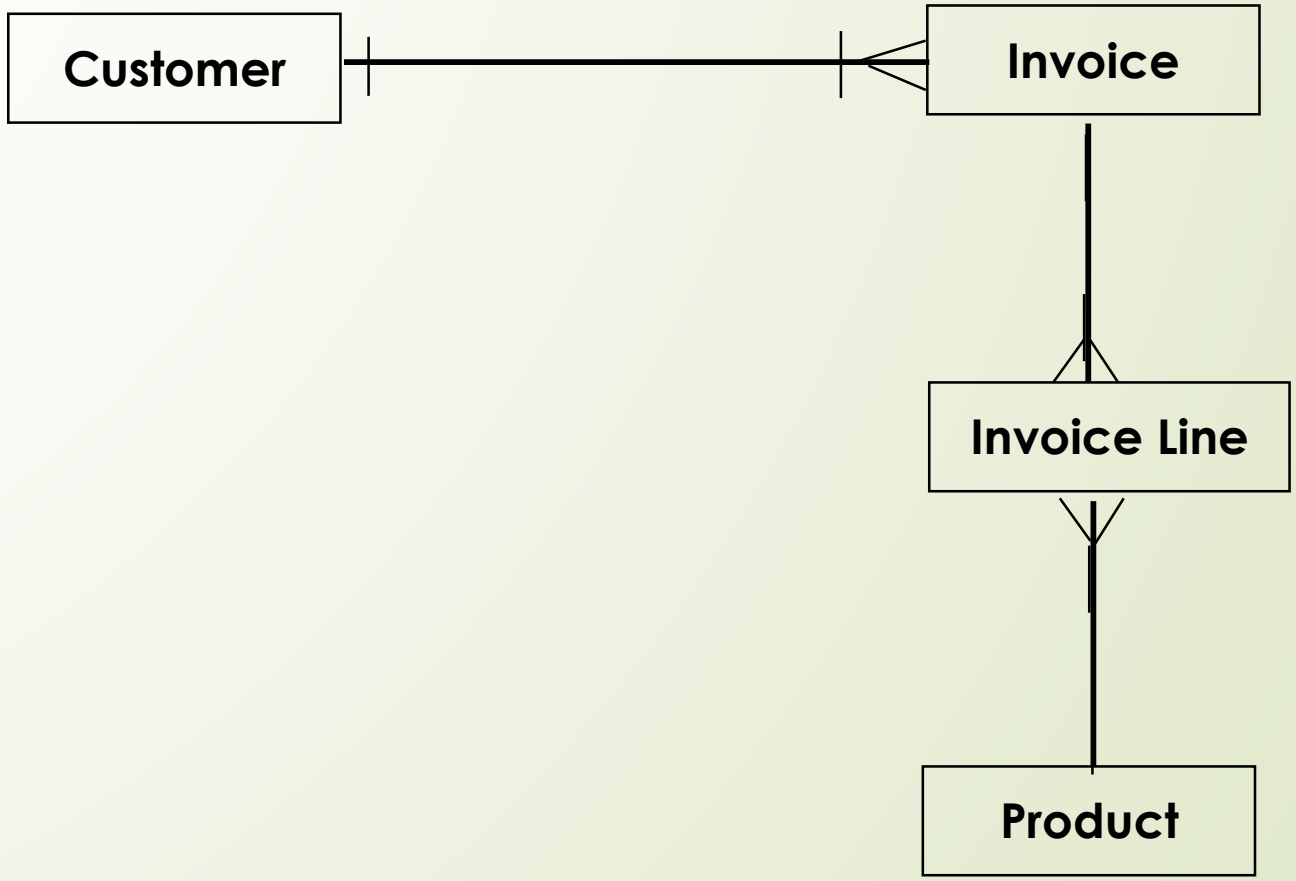
The Entity Relationship Model

- **Advantages**
  - Exceptional conceptual simplicity
  - Visual representation
  - Effective communication tool
  - Integrated with the relational data model

The Entity Relationship Model

- **Disadvantages**
  - Limited constraint representation
  - Limited relationship representation
  - No data manipulation language
  - Loss of information content

Exercise for Students



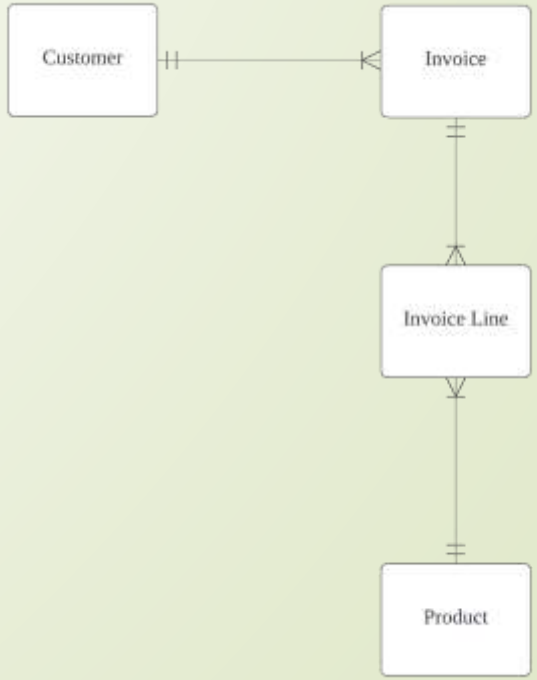
Describe the above relationships and correct the errors in the use of crows' foot symbols

Answer

Business Rules:

1. One customer will receive one or more invoice. One invoice is issued to only one customer.
2. One invoice exists in many invoice line. One invoice line belongs to one invoice.
3. One invoice line can have one product. One product is exists in many invoice line.

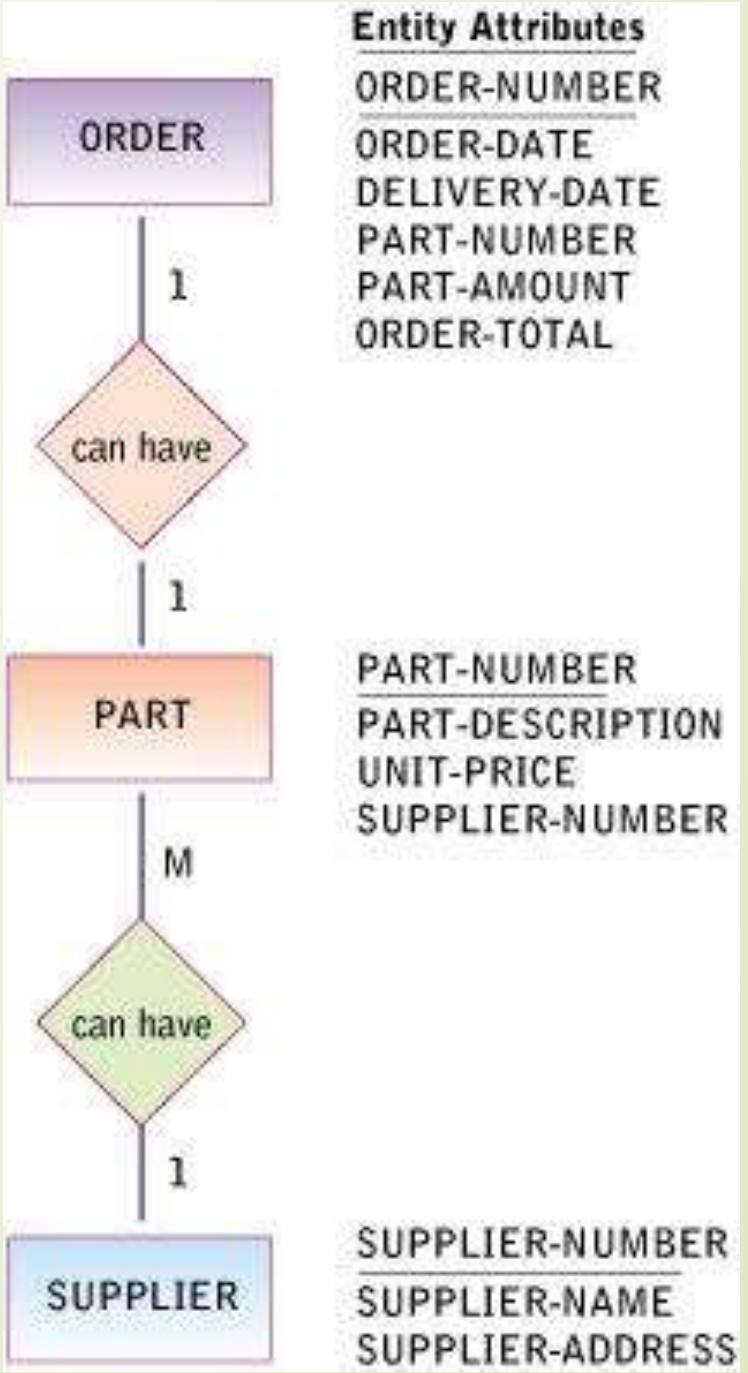
ERD in Crow's foot notation



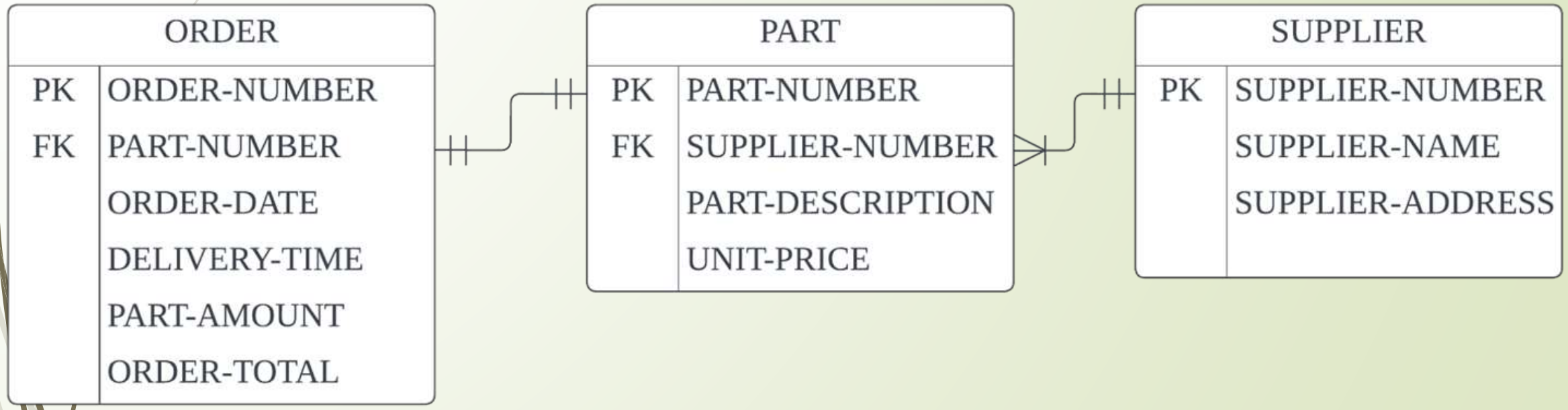
Example of ERD

EXERCISE:
Redraw this ERD using Crow's Foot notations.

Identify the various keys



Answer


ORDER
PK ORDER-NUMBER
FK PART-NUMBER
ORDER-DATE
DELIVERY-TIME
PART-AMOUNT
ORDER-TOTAL
PART
PK PART-NUMBER
FK SUPPLIER-NUMBER
PART-DESCRIPTION
UNIT-PRICE
SUPPLIER
PK SUPPLIER-NUMBER
SUPPLIER-NAME
SUPPLIER-ADDRESS

The Object-Oriented Model

- In the **object-oriented data model (OODM)**, both data and its relationships are contained in a single structure known as an object.
- OODM becomes the basis for the **object-oriented database management system (OODBMS).**

The Object-Oriented Model

Why OO?

- Conventional data models (e.g., relational) are
  - Inadequate
  - Can't model complex and unstructured data
  - Can't model processes (dynamic behaviour)
  - Doesn't support reuse

A more 'natural' way to represent the real world

- Encapsulation – incorporating both data and functions in a unit where they are protected from modification from outside.

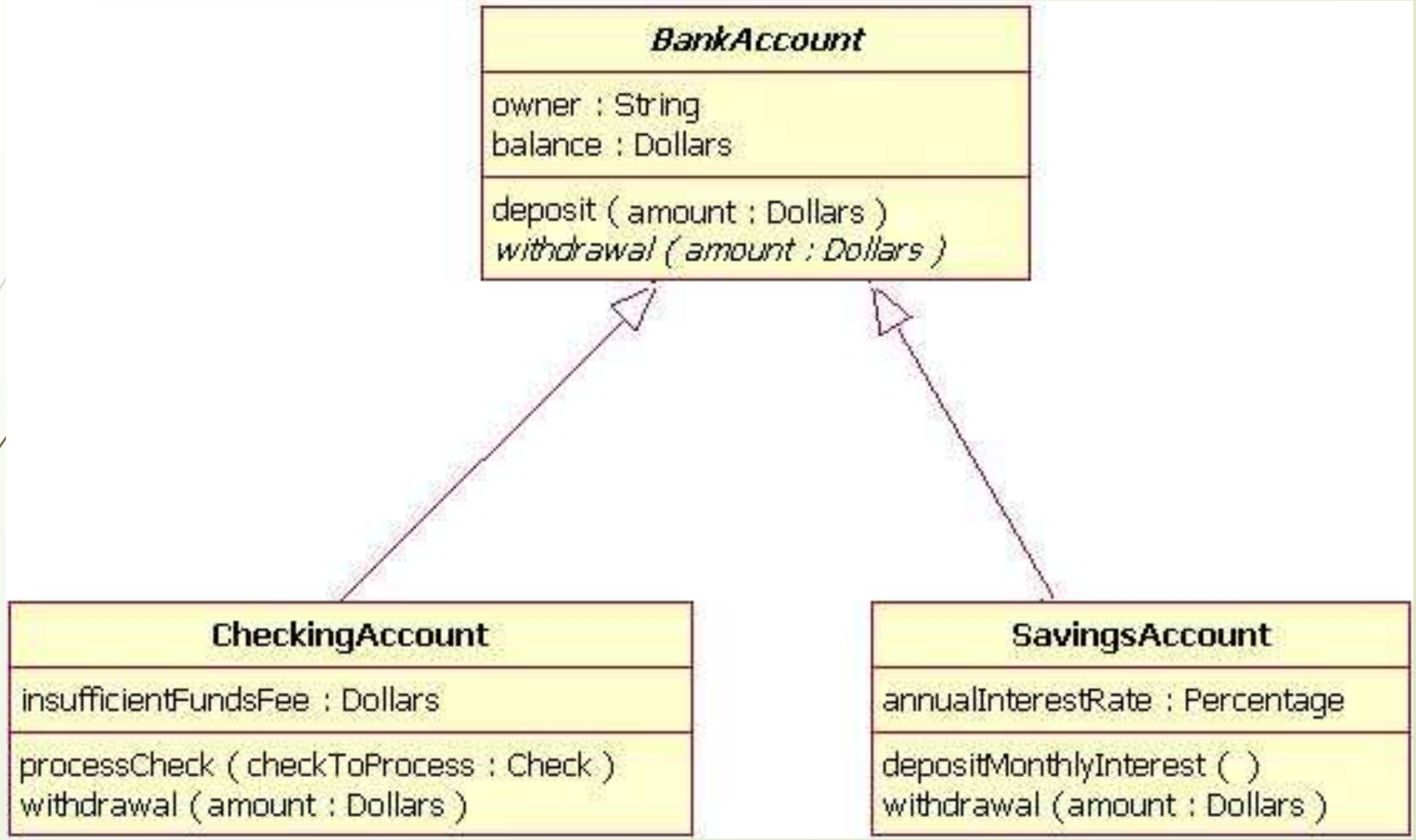
Object-Oriented Data Model - Basic Structure

- An **object** is an abstraction of a real-world entity.
- **Attributes** describe the properties of an object. For example, a PERSON object includes the attributes Name and Date of Birth.
- Classes are organized in a **class hierarchy**. For example, the CUSTOMER class and the EMPLOYEE class share a parent PERSON class.

Object-Oriented Data Model - Basic Structure

- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of 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. UML is a language based on OO concepts that describes a set of diagrams and symbols you can use to graphically model a system.

Example of Class Diagram in UML



The Object Oriented Model

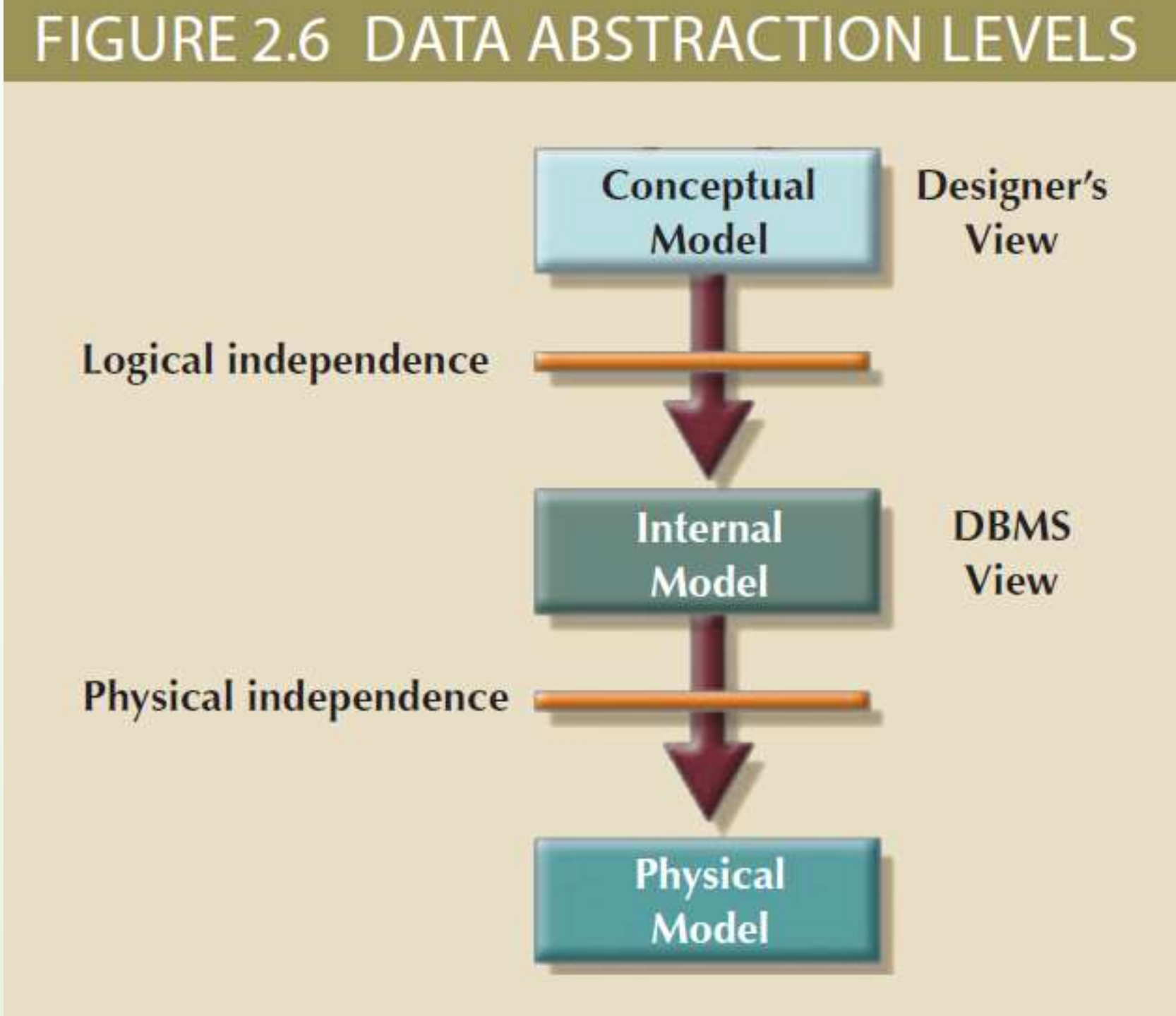
- **Advantages**
  - Adds semantic content
  - Visual presentation includes semantic content
  - Database integrity
  - Both structural and data independence

The Object Oriented Model (continued)

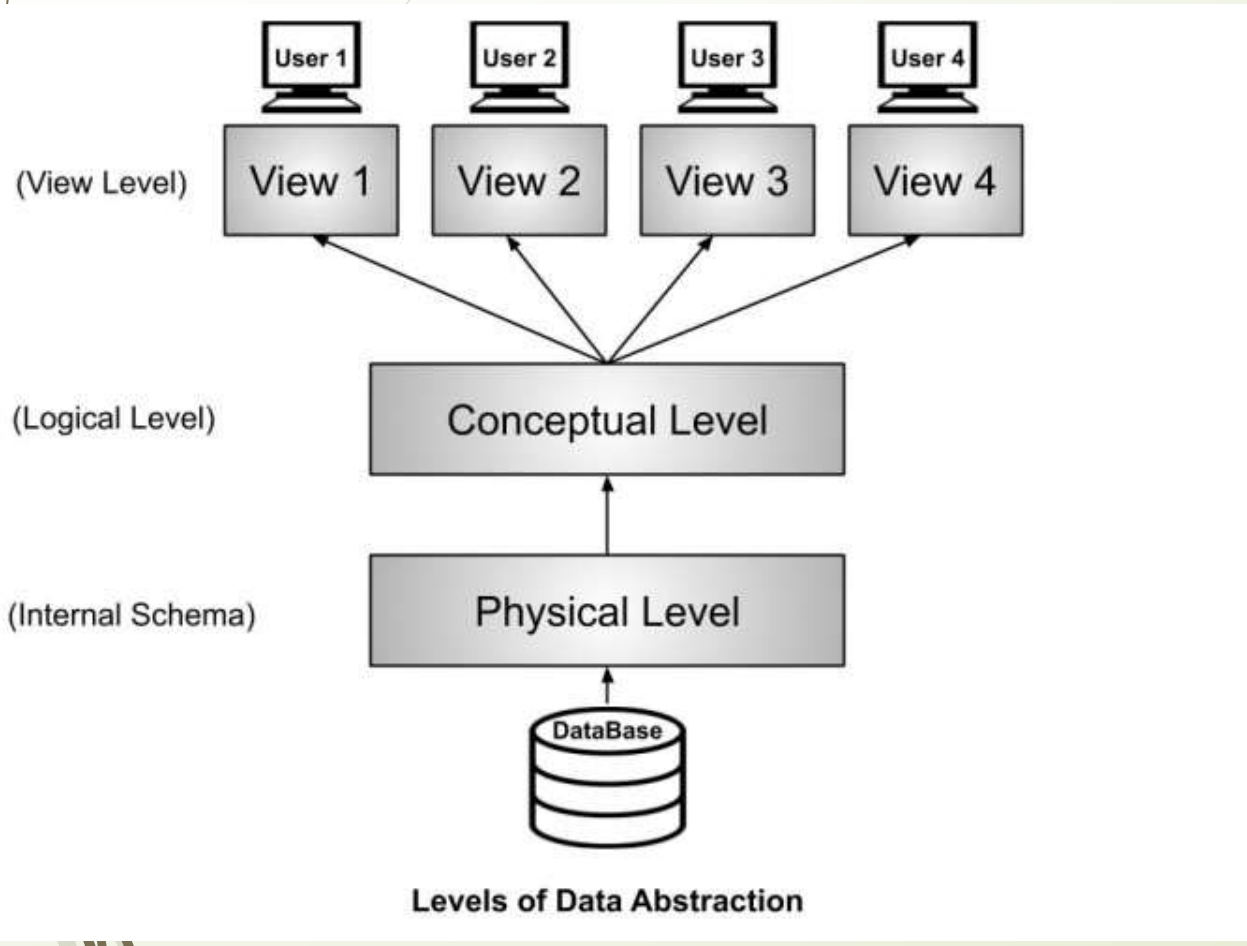
- **Disadvantages**
  - Slow pace of OODM standards development
  - Complex navigational data access
  - Steep learning curve
  - High system overhead slows transactions
  - Lack of market penetration

Data Abstraction Levels

Used in various stages of database design



Data Abstraction Levels



Levels of Data Abstraction

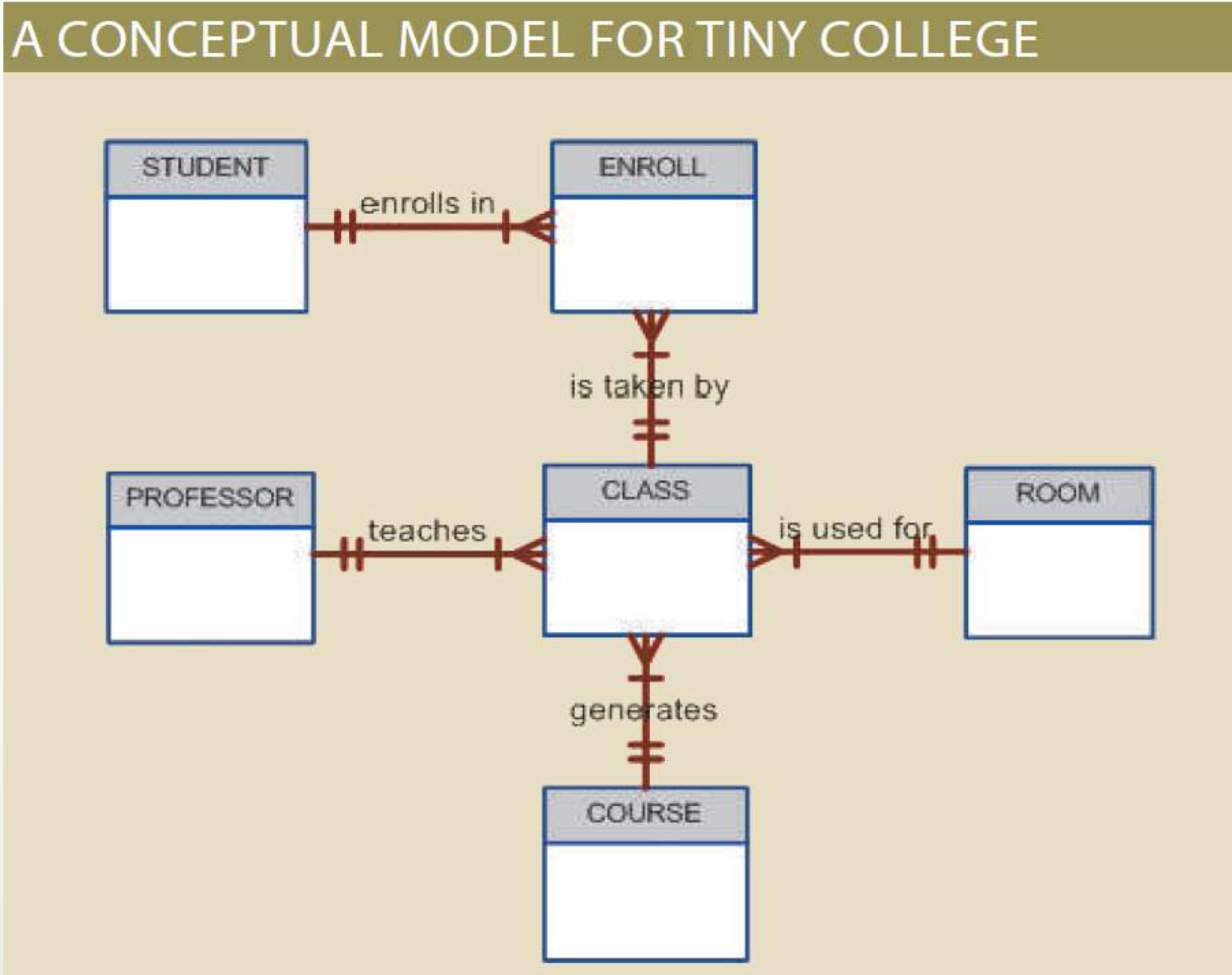
In DBMS, there are 3 levels of data abstraction:

- Physical Level or Internal Level: This is the layer where the raw data is stored in file format on physical hard drives.
- Logical Level or Conceptual Level: In this layer, the raw data is taken from the physical layer and organized in a proper structure, like in tabular format.
- View Level or External Level: At this level, the end users get the data depending on the queries. The same data can be viewed in multiple ways, like tables, graphs, or pie charts.

Conceptual Data Model

- A conceptual data model identifies the highest-level relationships between different entities.
- Features of conceptual data model include:
  - Includes the important entities and the relationships among them.
  - No attribute is specified.
  - No primary key is specified.

Example:

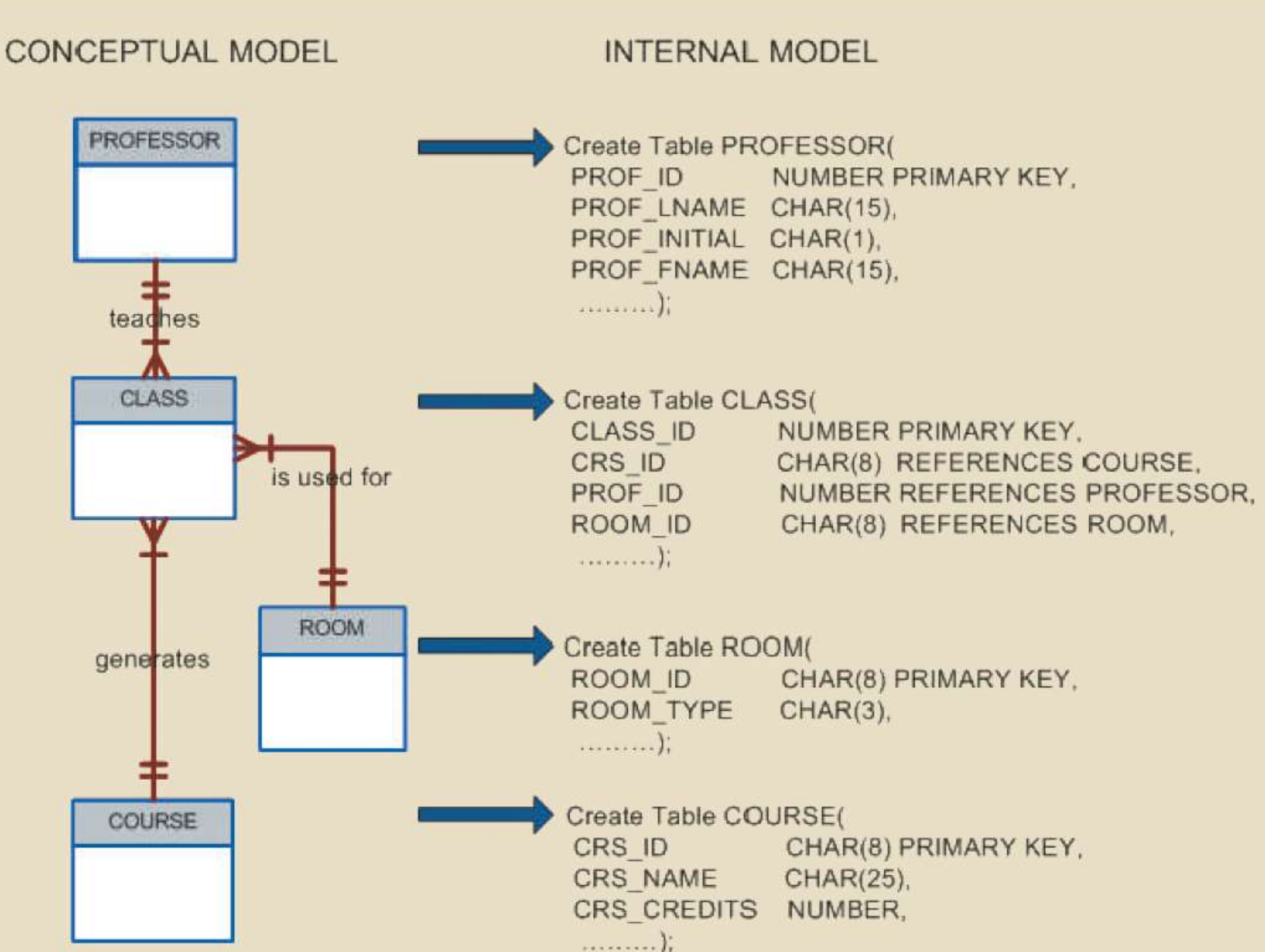

A CONCEPTUAL MODEL FOR TINY COLLEGE
STUDENT
enrolls in
ENROLL
is taken by
PROFESSOR
teaches
CLASS
is used for
ROOM
generates
COURSE

Advantages of Conceptual Model

1. It provides a bird's eye view of the data environment that is easy to understand.
2. It is independent of both software and hardware.
   - Software independence means that the model does not depend on the DBMS software.
   - Hardware independence means that the model does not depend on the hardware used in the implementation of the model.
   - Therefore, changes in either the hardware or the DBMS software will have no effect on the database design at the conceptual level.

From Conceptual Model To Internal Model

FIGURE 2.9 INTERNAL MODEL FOR TINY COLLEGE



Data Dictionary

- Data dictionary provides a detailed description of all tables in the database created by the user and designer. Thus, the data dictionary contains at least all of the attribute names and characteristics for each table in the system.
- In short, the data dictionary contains metadata (data about data).

A SAMPLE DATA DICTIONARY

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
CUSTOMER	CUS_CODE	Customer account code	CHAR(5)	99999	10000–99999	Y	PK	
	CUS_LNAME	Customer last name	VARCHAR(20)	Xxxxxxxx		Y		
	CUS_FNAME	Customer first name	VARCHAR(20)	Xxxxxxxx		Y		
	CUS_INITIAL	Customer initial	CHAR(1)	X				
	CUS_RENEW_DATE	Customer insurance renewal date	DATE	dd-mmm-yyyy				
	AGENT_CODE	Agent code	CHAR(3)	999			FK	AGENT
AGENT	AGENT_CODE	Agent code	CHAR(3)	999		Y	PK	
	AGENT_AREACODE	Agent area code	CHAR(3)	999		Y		
	AGENT_PHONE	Agent telephone number	CHAR(8)	999–9999		Y		
	AGENT_LNAME	Agent last name	VARCHAR(20)	Xxxxxxxx		Y		
	AGENT_YTD_SLS	Agent year-to-date sales	NUMBER(9,2)	9,999,999.99				

FK	= Foreign key
PK	= Primary key
CHAR	= Fixed character length data (1 – 255 characters)
VARCHAR	= Variable character length data (1 – 2,000 characters)
NUMBER	= Numeric data. NUMBER (9,2) is used to specify numbers with up to nine digits, including two digits to the right of the decimal place. Some RDBMS permit the use of a MONEY or CURRENCY data type.

Why are Data Dictionaries Important?

1. To document the data about the data (metadata) in the system.
2. To locate errors and omissions in the system.
3. To communicate a common meaning for all of the elements in the system.