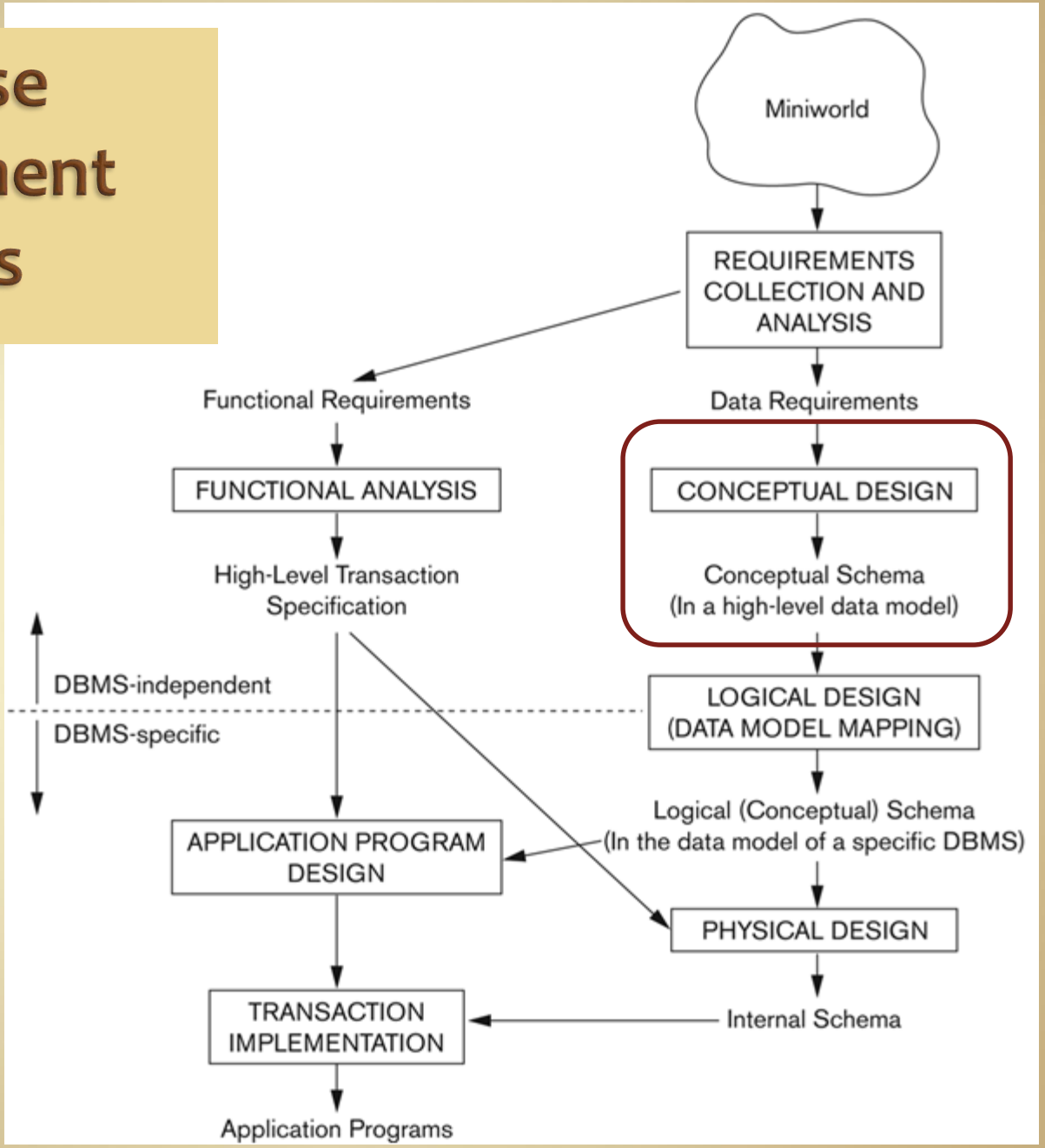
Advanced Database Management Systems

Lecture 2
Entity-Relationship Diagrams

Database Development Process



Common Data Models

UML/OO	ER	Relational
class	entity type	relation/table
object	entity	tuple/row
attribute	attribute	attribute/column
association	relationship	foreign key
	key attribute	primary key
inheritance	inheritance	foreign key

- We have standard techniques for translating between data models.

Common Data Models

- UML and ER have graphical notations
  - very good for conceptual design
  - very expressive: capture more application semantics
- Relational model is an implementation model
  - not good for conceptual design
  - not very expressive: many concepts map to same notation

Entity Types

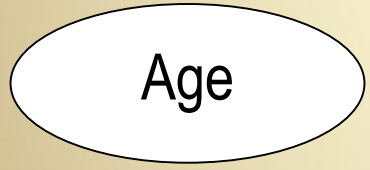
- Entity types →boxes
  Weak entity type → double box

Entity Types

- Entity types are similar to classes, they describe potential objects (entities) that will appear in the database.
- Weak entity types describe dependent entities, entities that depend on other entities for identity.

Attributes and Keys

- Attributes →ovals
- Key attributes → underlined name
- Partial key attributes →dotted underlined name



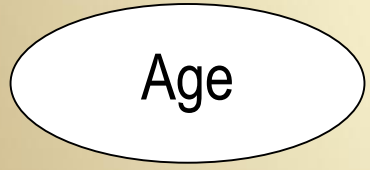
Attribute

Key Attribute

Partial Key Attribute

Attributes and Keys

- Key attributes must be unique for each entity
- Keys are used to identify particular entities
- Partial keys are only partially unique
  - used for weak entity types



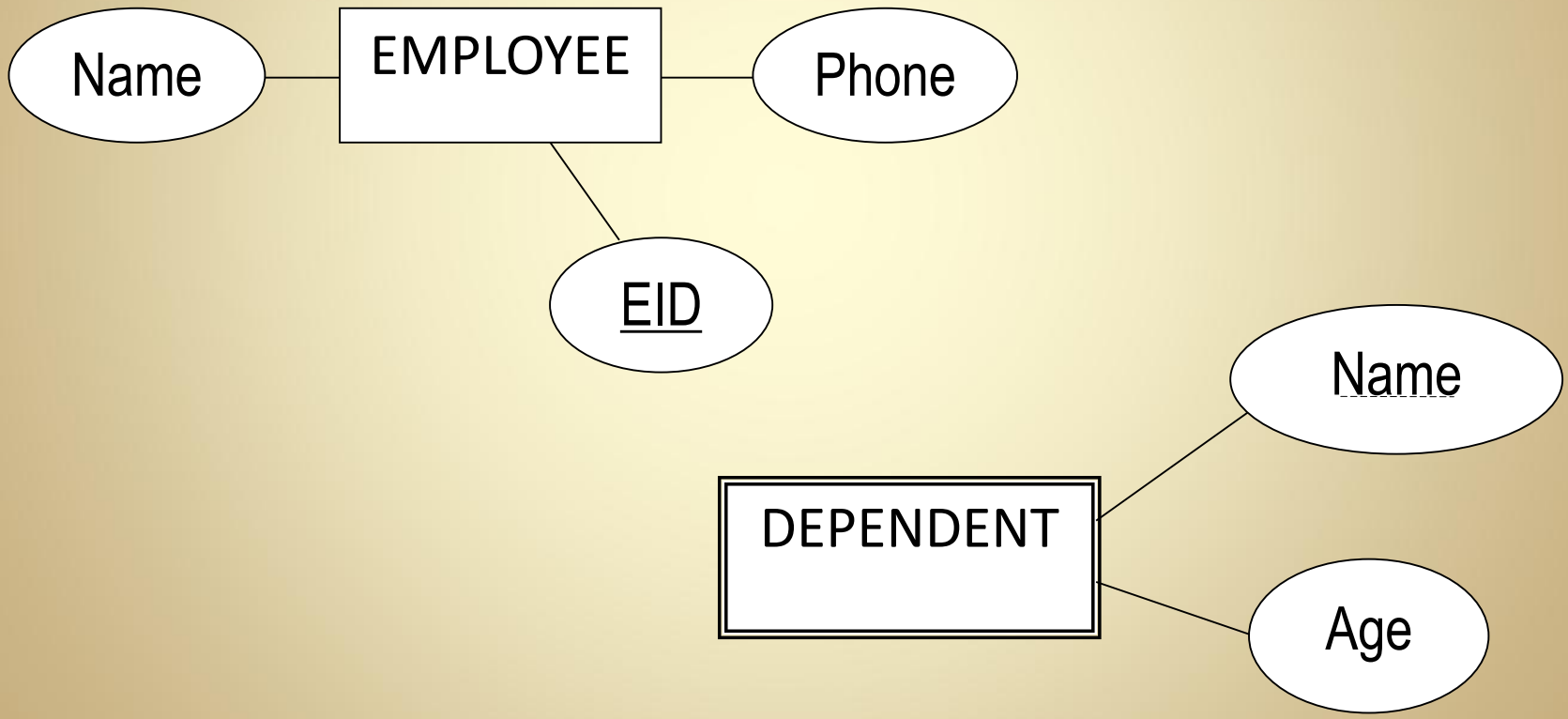
Attribute

Key Attribute

Partial Key Attribute

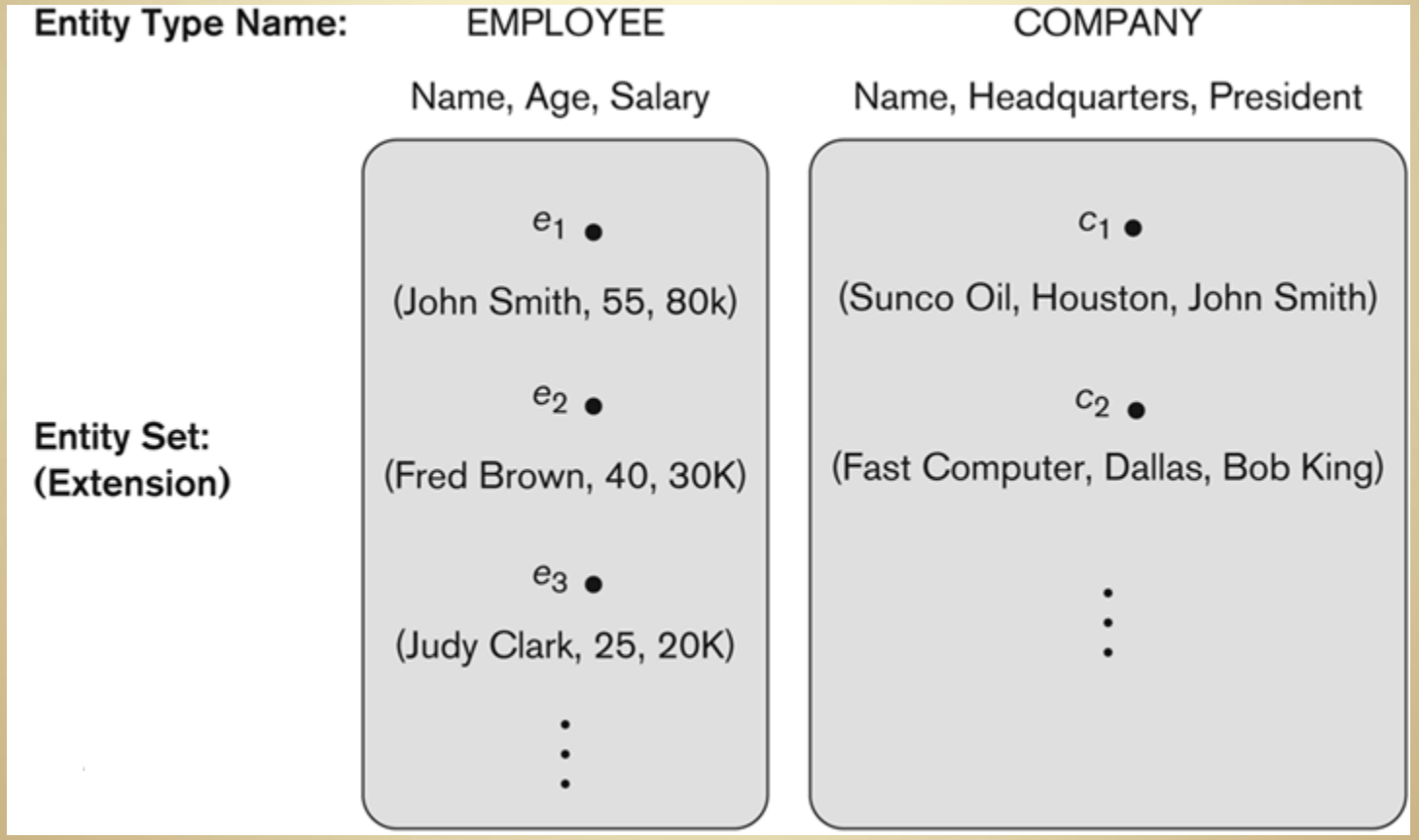
Entity Types and Attributes

- Attributes are connected to entity types by lines



Entity Types and Entities

- Entities are instances of an entity type.



Entity Types and Keys

- All regular entity types must have a key attribute or set of key attributes
- Weak entity types must have partial keys
- Weak entities get part of their key (and part of their identity) from some related entity.

Sets and Derived Attributes

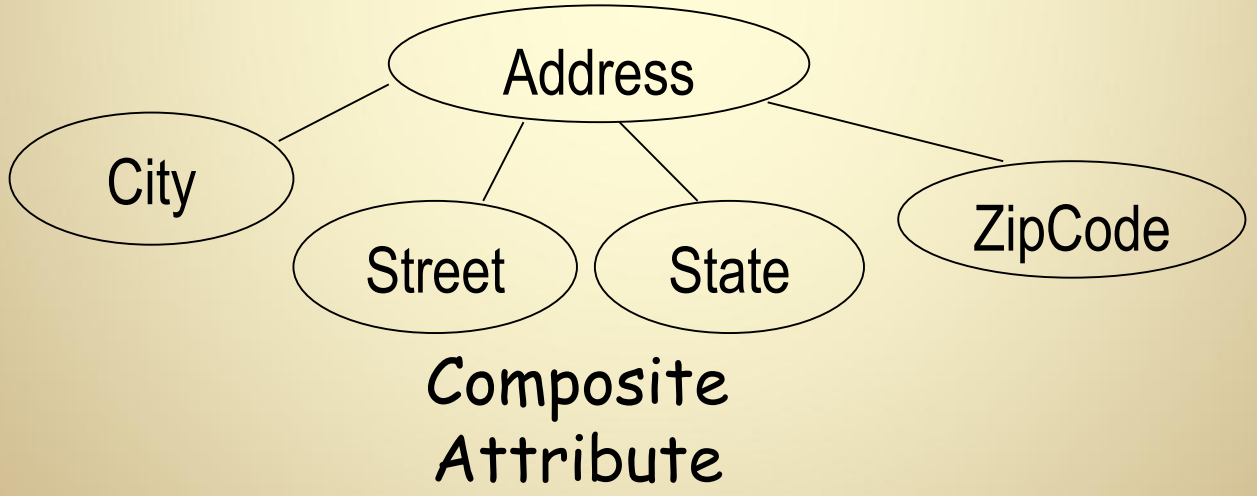
- Multivalued attributes → double lined oval
  - multivalued = set valued
  - that there may be more than one value for the attribute.
- Derived attributes → dashed line ovals
  - the attribute is computed from other data

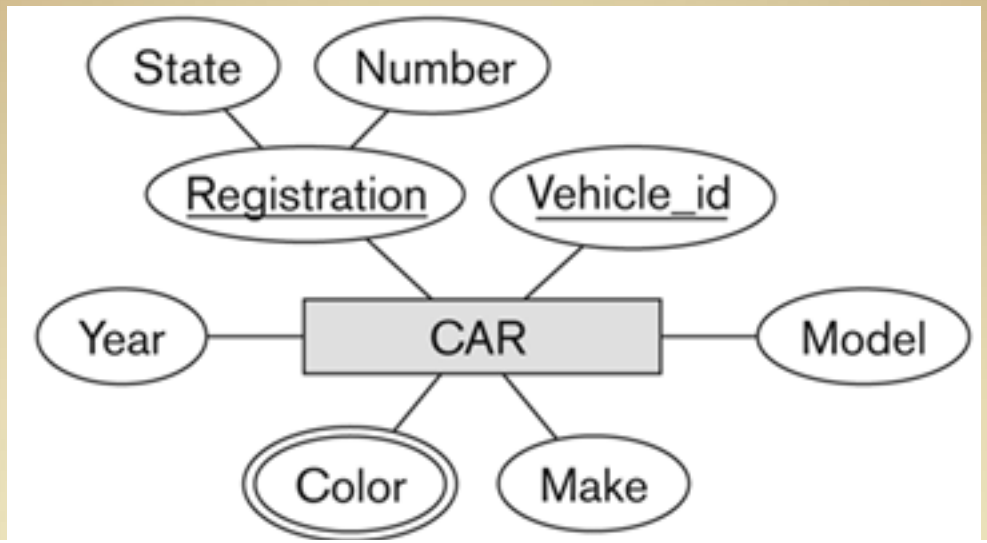
Multivalued Attribute

Derived Attribute

Composite Attributes

- Composite attributes → tree
  - composed of other attributes.
  - used for a set of related attributes, when the set is not a conceptual entity
  - the composite doesn't have identity … it doesn't have a key





CAR

Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

CAR_1

((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2

((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

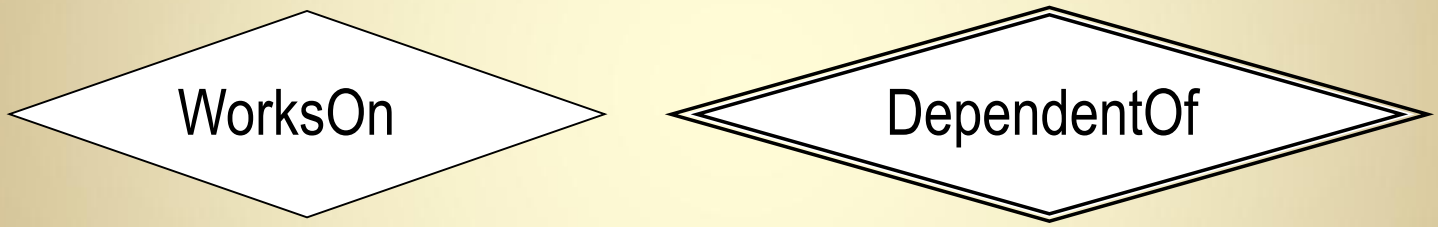
CAR_3

((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Relationships

- Relationships → diamonds
- Identifying relationship → double diamond

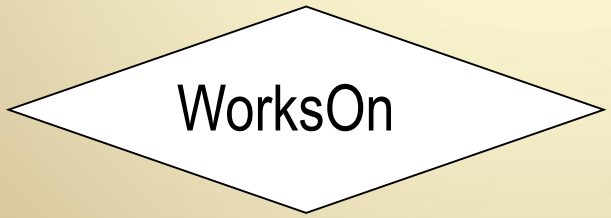


Relationship

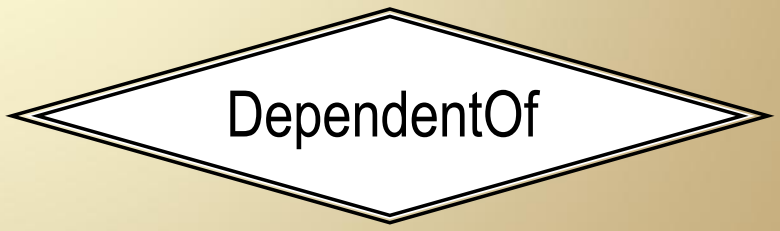
Identifying Relationship

Relationships

- Relationships indicate a meaningful connection between two entity types
- Relationships may have attributes, but they cannot have key attributes.
- Identifying relationships connect a weak entity type to some other entity type
  - indicates where the weak entity gets a key to complete its own partial key

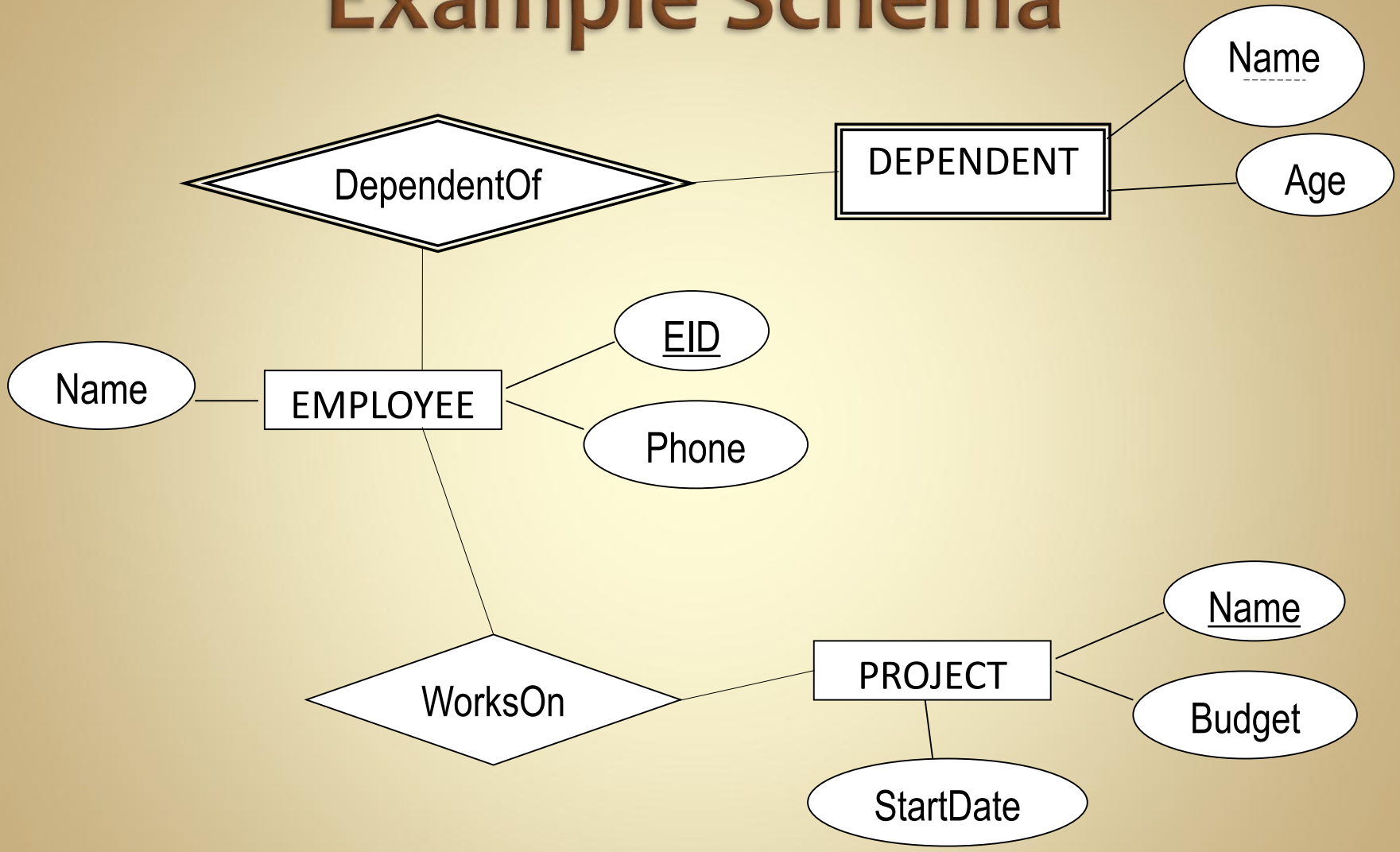


Relationship



Identifying Relationship

Example Schema



Participation and Cardinality

- Participation and cardinality define constraints on relationships
- Participation indicates whether an entity is required to take part in a relationship
- Cardinality ratios and structural constraints place limits on the number of entities that may participate in a relationship

Participation Constraints

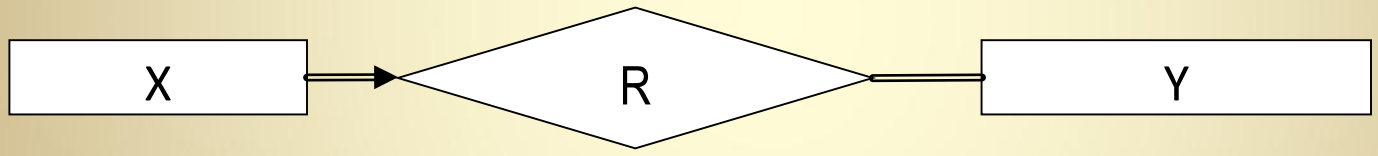
- Total participation → double or thick line
  - indicates required participation
- Partial participation → thin line
  - indicates optional participation

Total Participation

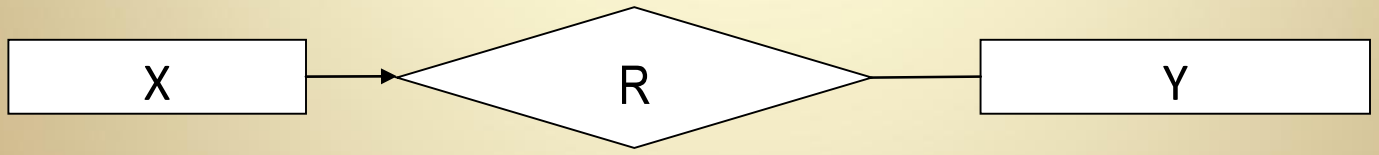
Partial Participation

Participation Constraints

- Arrowheads can be used to indicate an upper bound of 1 for participation
  - (not used in our textbook)



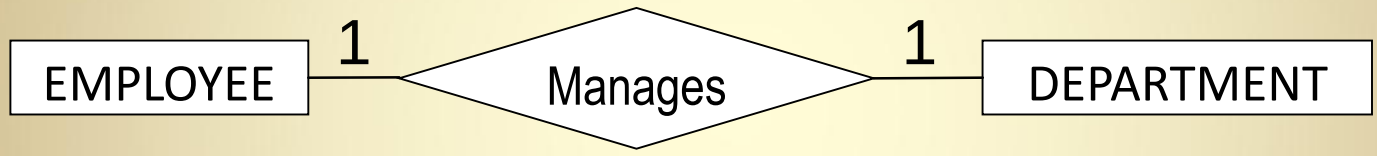
X must participate in exactly one R



X may participate in at most one R

Cardinality Ratios

- Cardinality ratios specify the *maximum* number of relationship instances that an entity may participate in



1:1 ratio

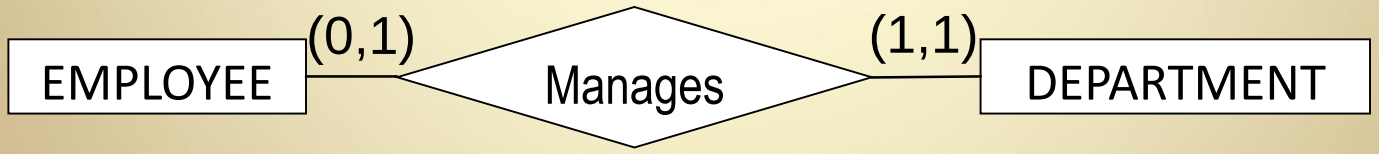
n:1 ratio

n:m ratio

Structural Constraints

- Structural constraints specify the *minimum* and *maximum* number of relationship instances that an entity may participate in

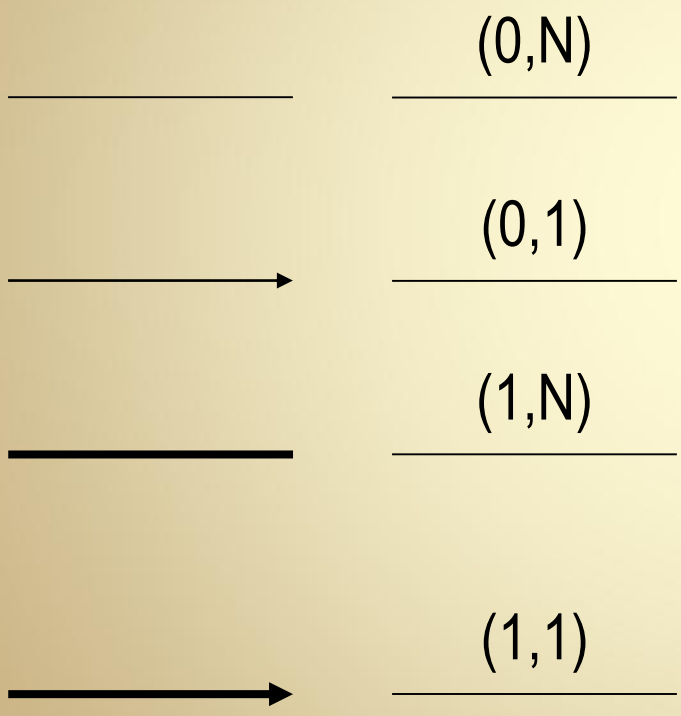
An employee must work for exactly 1 department.
A department must have at least 4 employees.



An employee may manage at most 1 department.
A department must have exactly 1 manager.

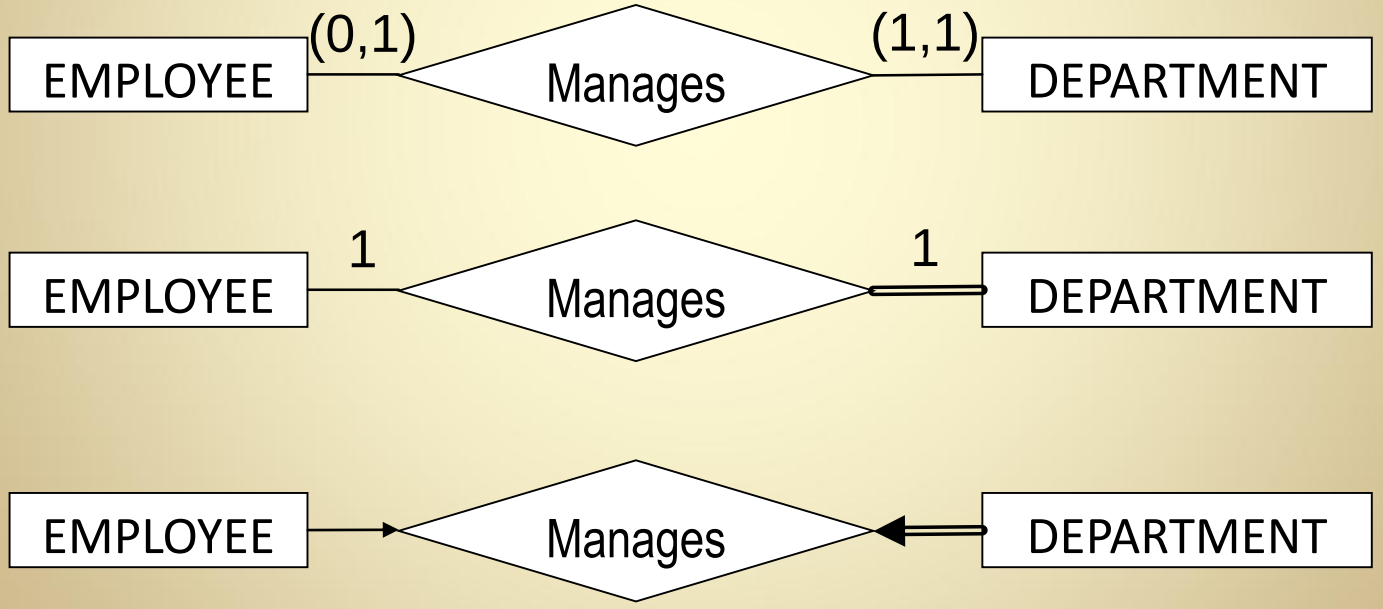
Participation and Cardinality

- There's generally numerous ways to express a relationship constraint.



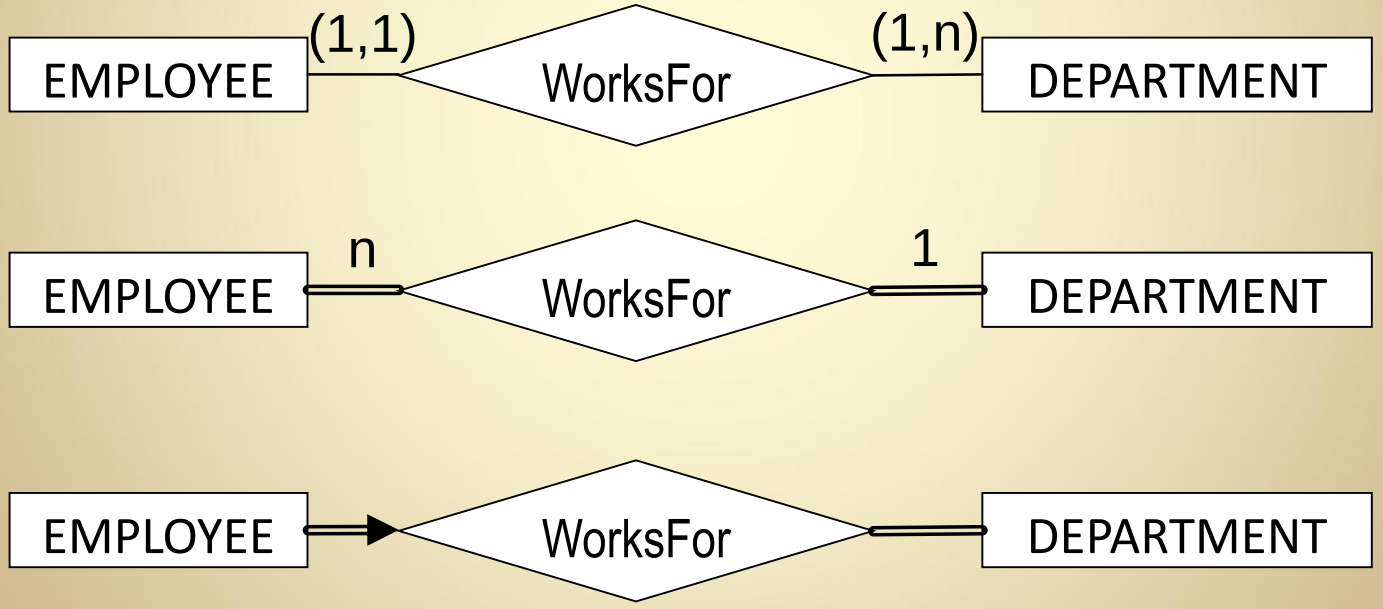
Equivalent Notations

- An employee can manage at most one department.
- A department must have exactly one manager.



Equivalent Notations

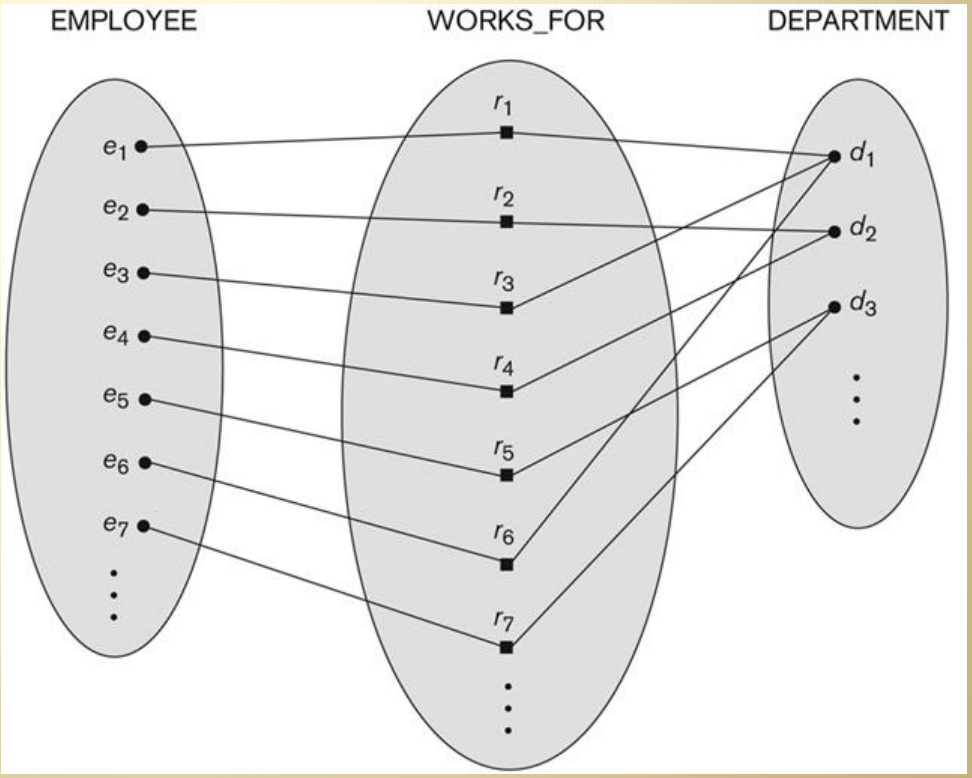
- An employee must work for exactly one department.
- A department must have at least one employee.



Relationship Instances

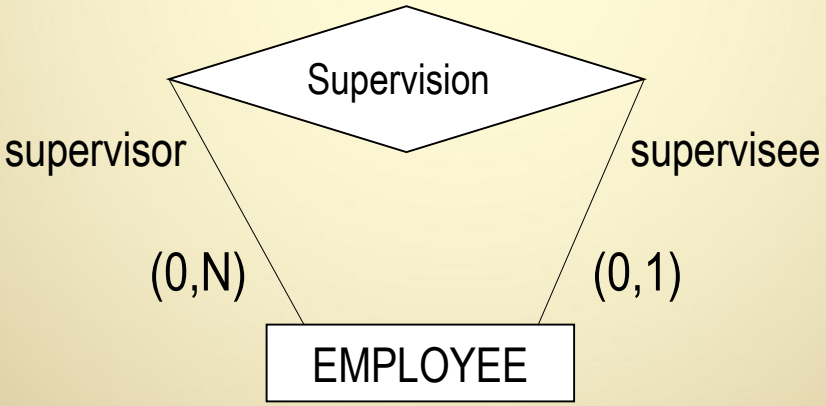
Each EMPLOYEE instance is connected to exactly one WORKS_FOR instance.

Each DEPARTMENT instance is connected to at least one WORKS_FOR instance.

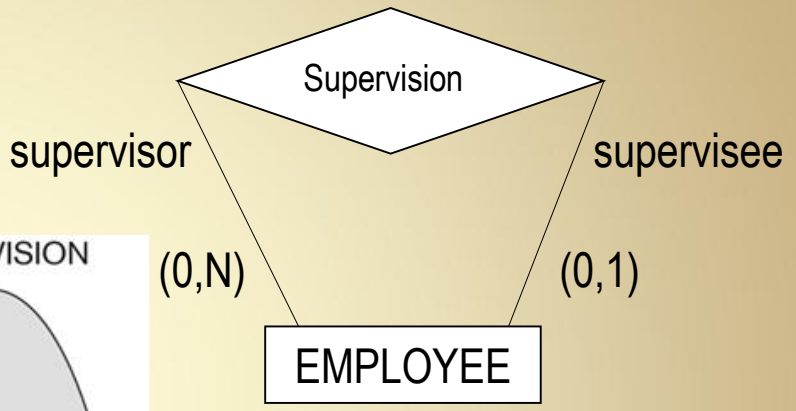


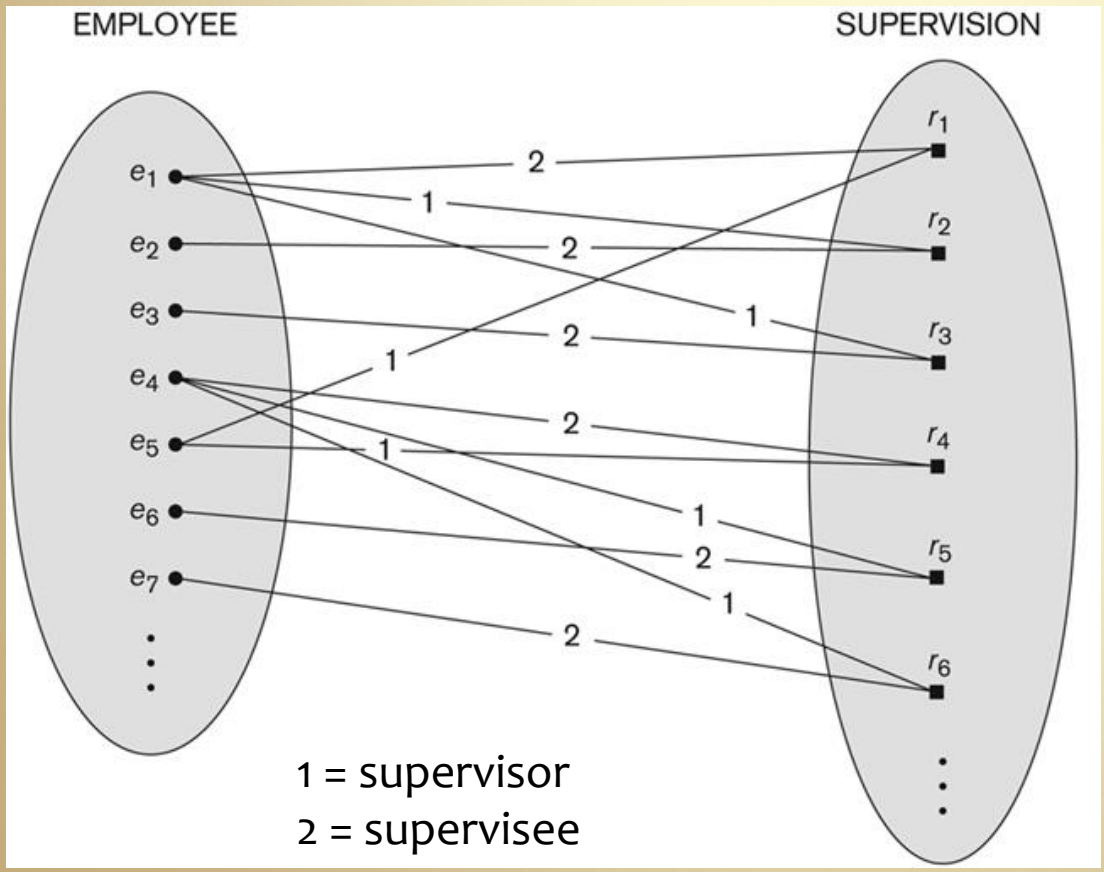
Relational Roles

- It is sometimes convenient to name an entity's role in a relationship.
  - particularly useful in recursive relationships
  - removes ambiguity in direction of relationship

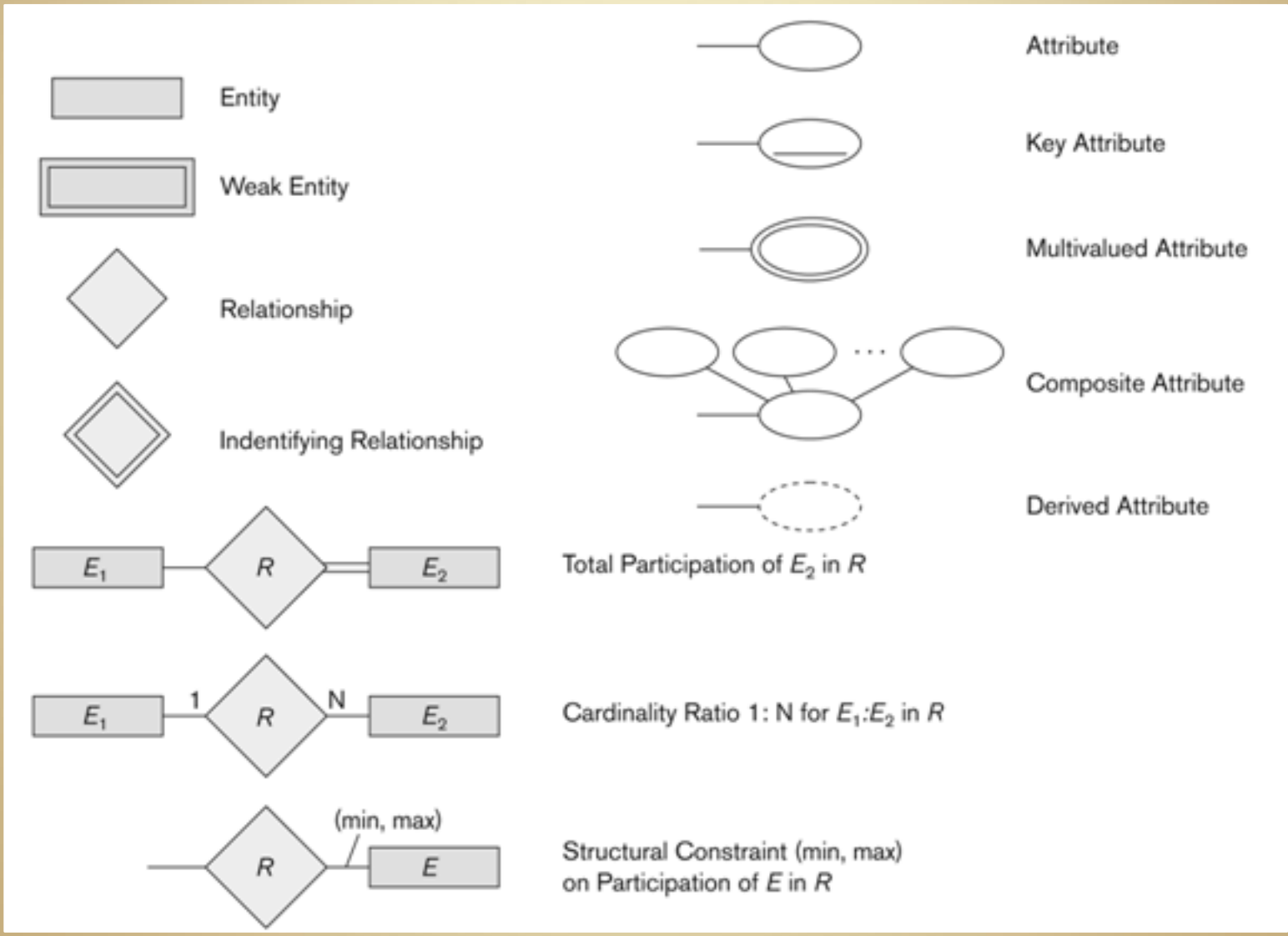


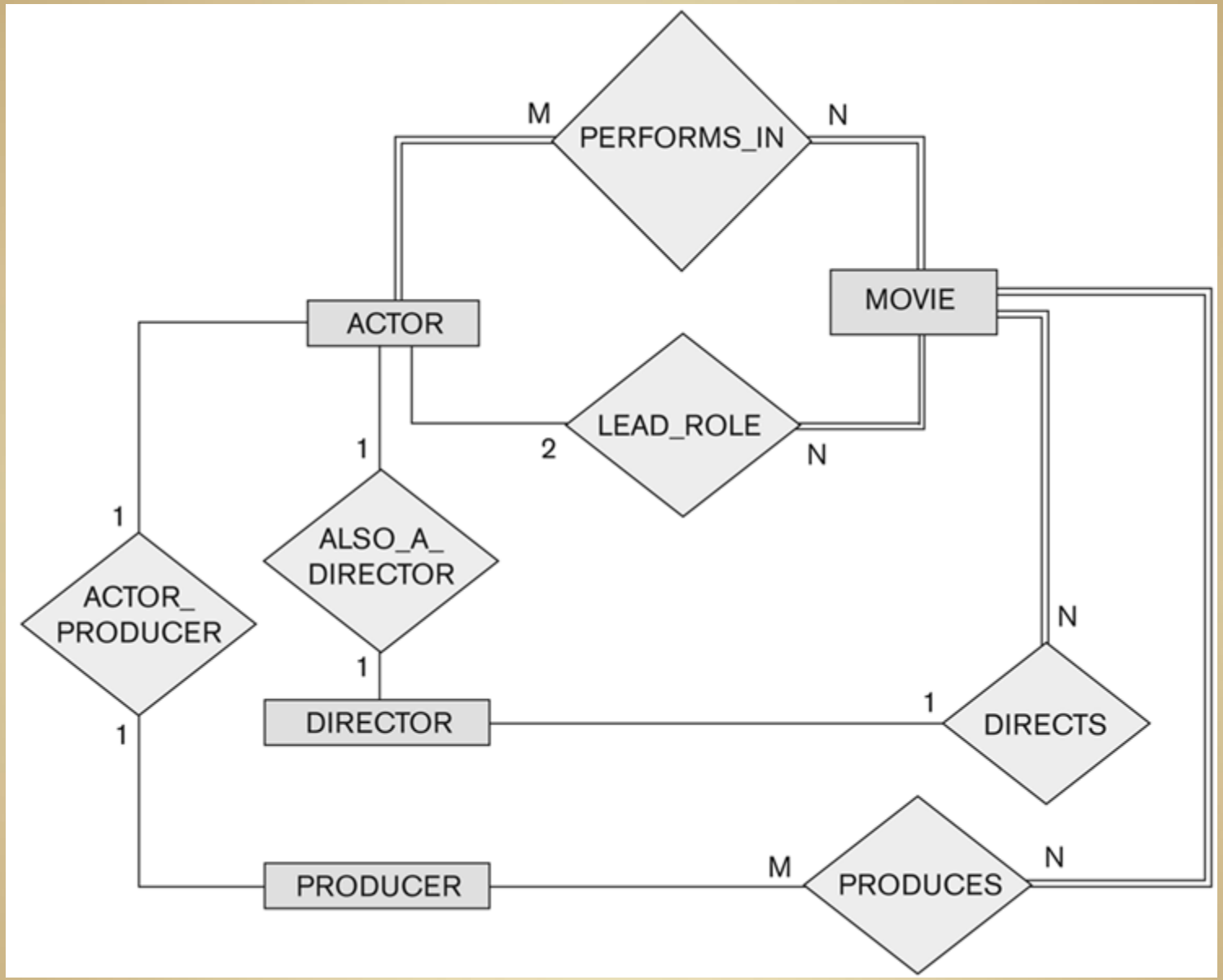
Recursive Relationship

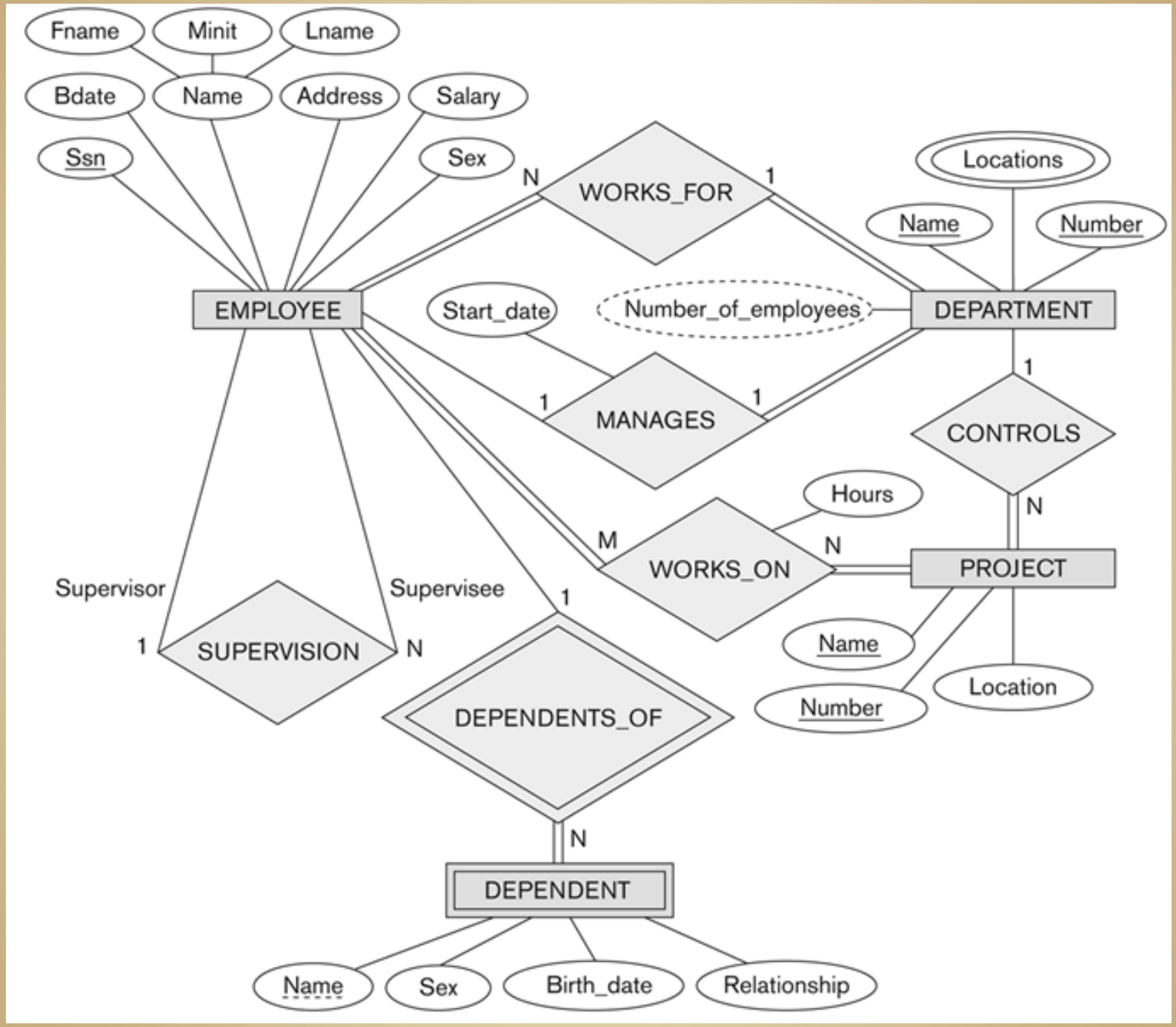


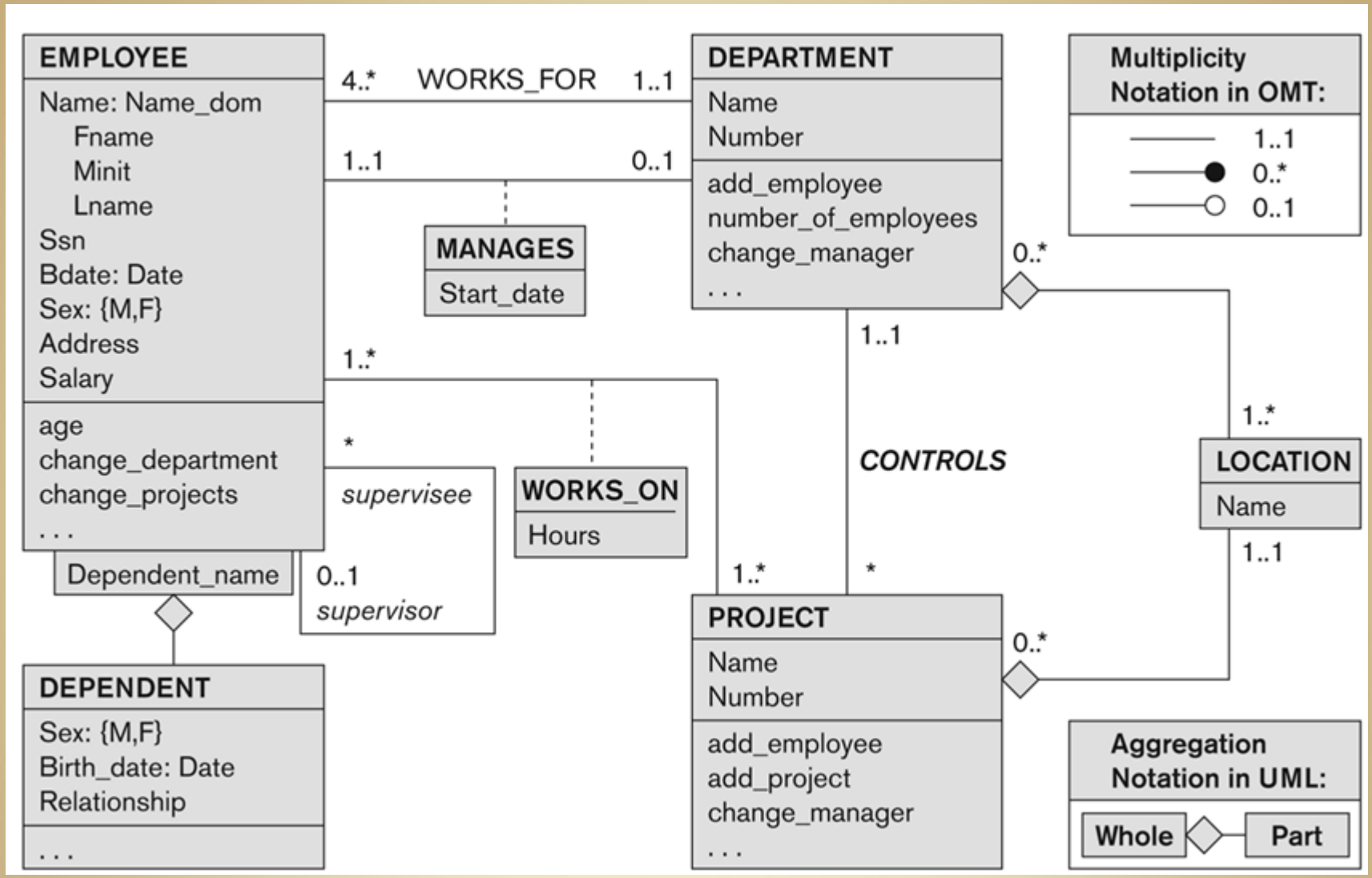


Notation Summary

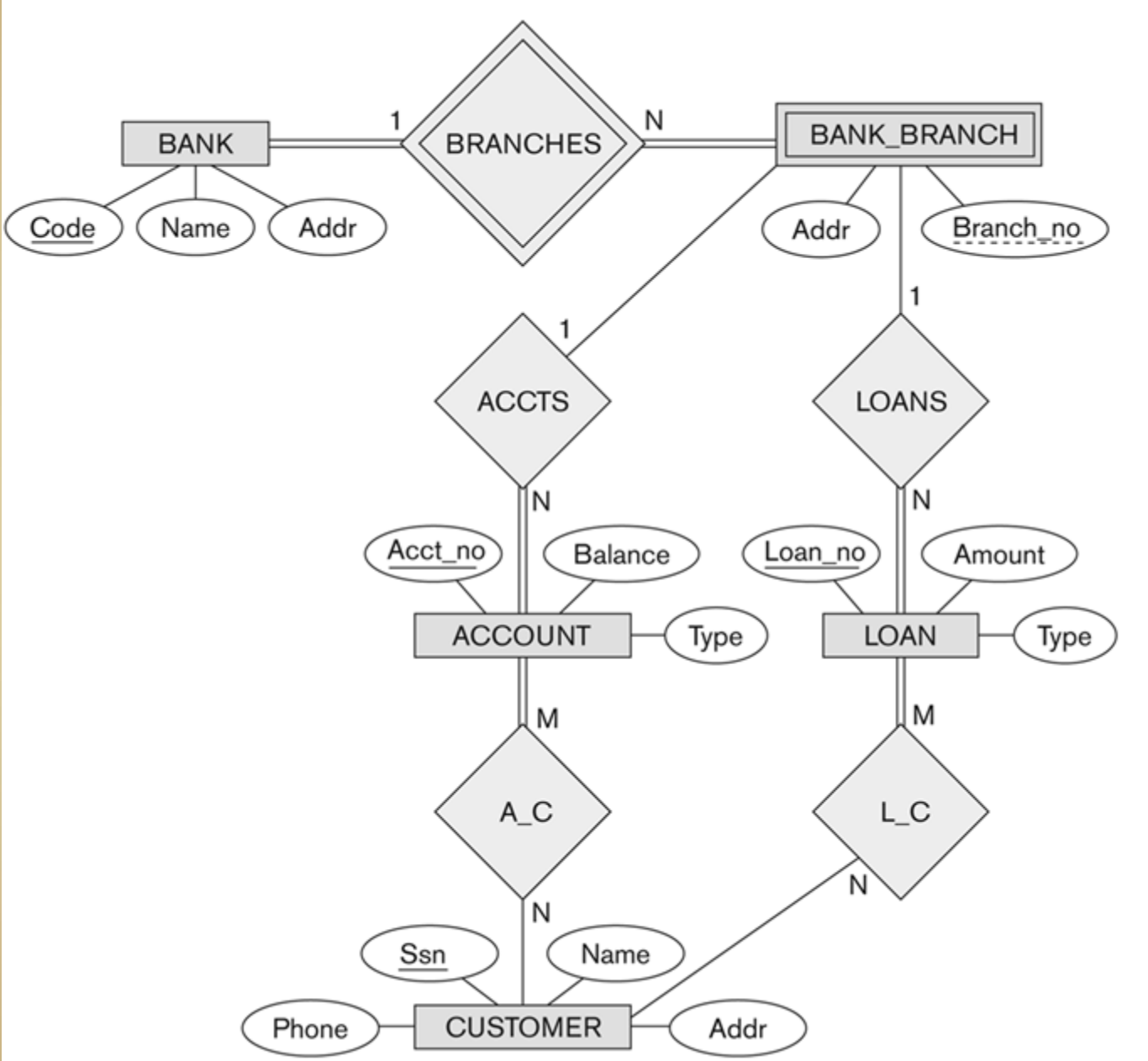


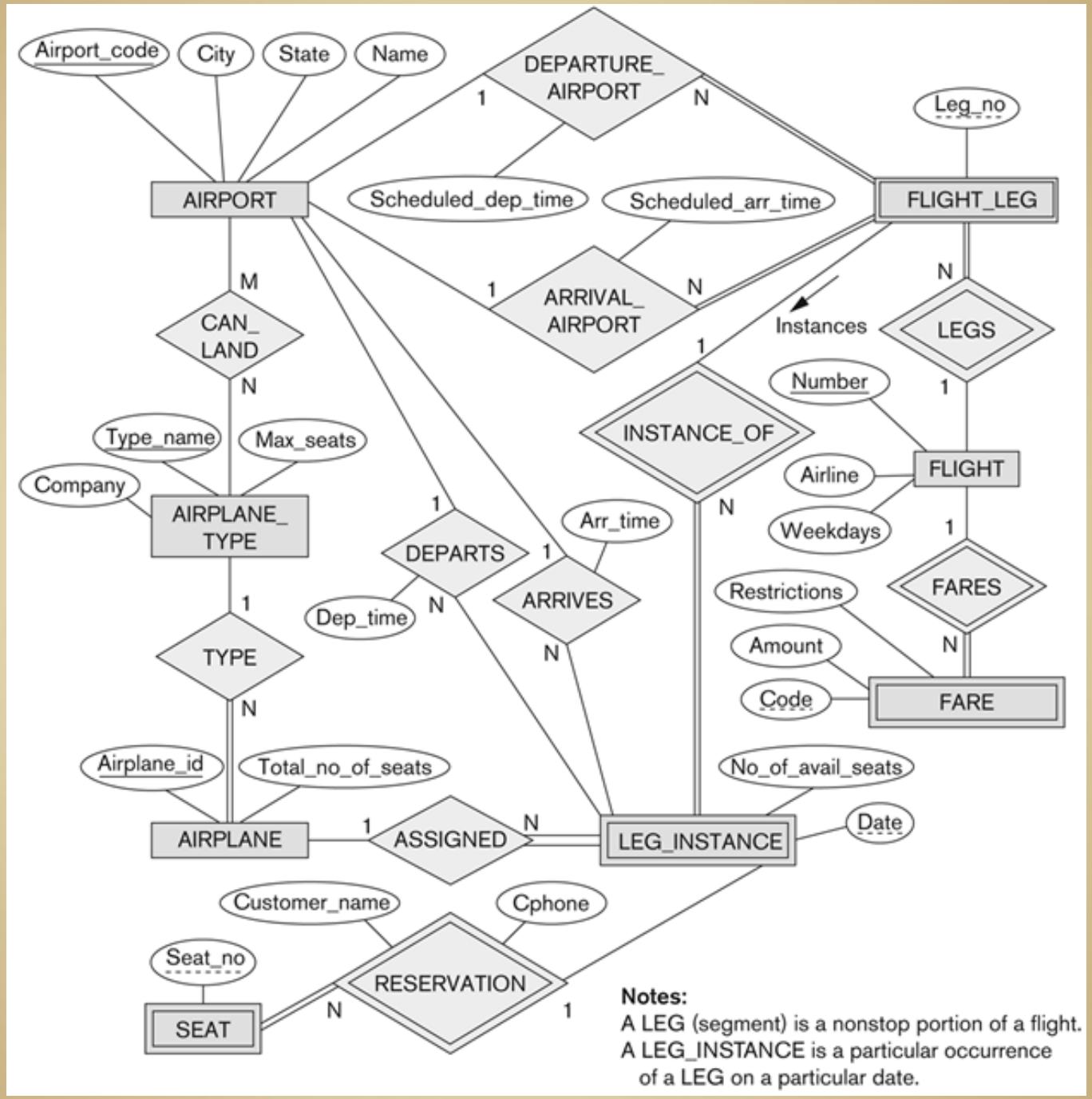

PERFORMS_IN
M
N
ACTOR
MOVIE
LEAD_ROLE
2
N
1
ALSO_A_
DIRECTOR
1
ACTOR_
PRODUCER
1
N
DIRECTOR
1
DIRECTS
1
PRODUCER
M
PRODUCES
N


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
WORKS_ON
N
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship



Equivalent Schema defined in UML

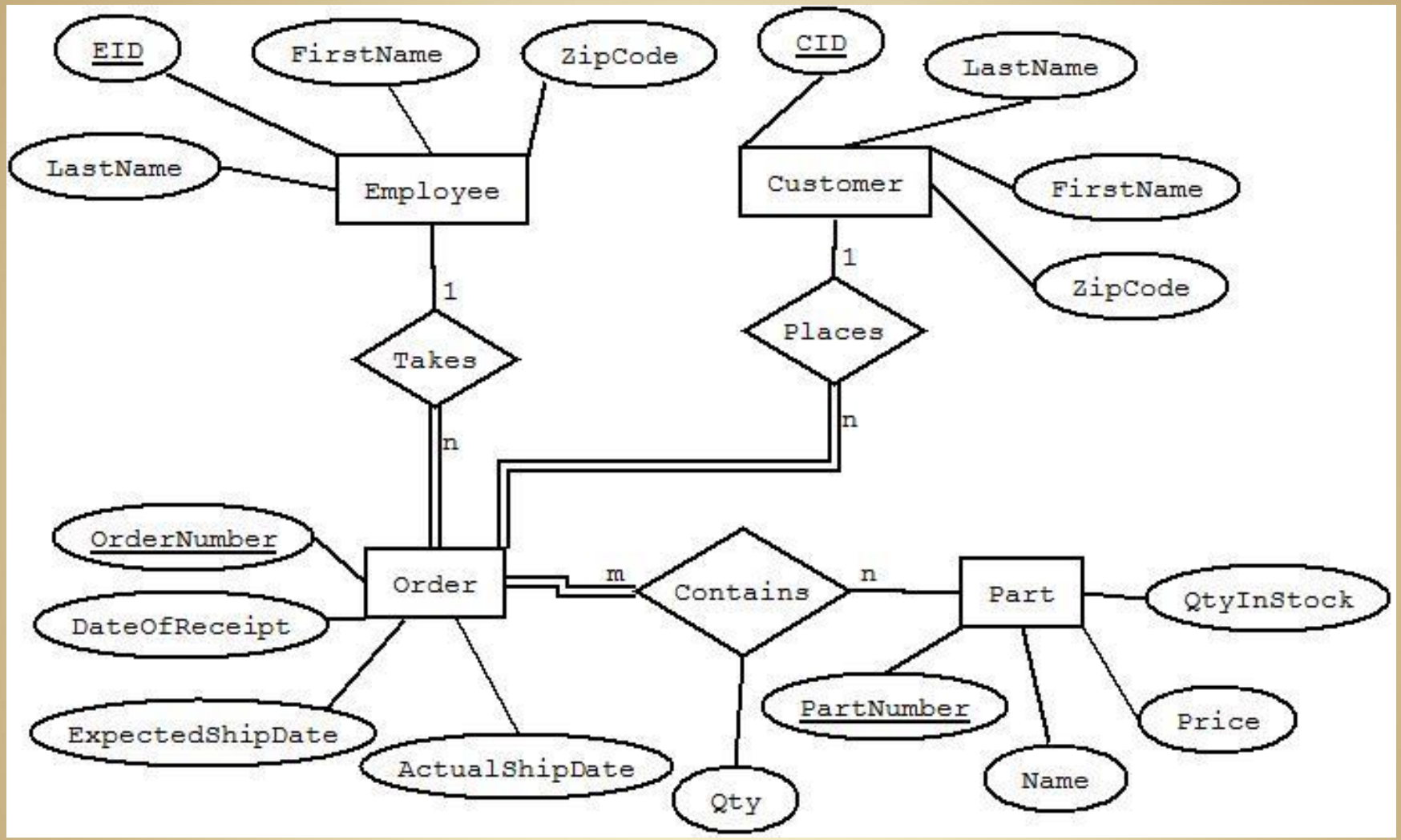

BANK
1
BRANCHES
N
BANK_BRANCH
Code
Name
Addr
Addr
Branch_no
1
1
ACCTS
LOANS
N
N
Acct_no
Balance
Loan_no
Amount
ACCOUNT
Type
LOAN
Type
M
M
A_C
L_C
N
N
Ssn
Name
Phone
CUSTOMER
Addr


Airport_code
City
State
Name
DEPARTURE_AIRPORT
1
N
Leg_no
AIRPORT
Scheduled_dep_time
Scheduled_arr_time
FLIGHT_LEG
M
1
ARRIVAL_AIRPORT
N
N
Instances
CAN_LAND
LEGS
N
1
1
Number
INSTANCE_OF
Type_name
Max_seats
Company
AIRPLANE_TYPE
1
DEPARTS
Arr_time
1
Airline
FLIGHT
N
Weekdays
1
ARRIVES
Restrictions
FARES
Dep_time
N
1
TYPE
N
Amount
N
Code
FARE
N
Airplane_id
Total_no_of_seats
No_of_avail_seats
AIRPLANE
1
ASSIGNED
N
LEG_INSTANCE
Date
Customer_name
Cphone
Seat_no
RESERVATION
SEAT
N
1
Notes:
A LEG (segment) is a nonstop portion of a flight.
A LEG_INSTANCE is a particular occurrence of a LEG on a particular date.

Design an ER schema for the following enterprise:

Consider a MAIL_ORDER database in which employees take orders for parts from customers. The data requirements are summarized as follows:

- The mail order company has employees, each identified by a unique employee number, first and last name, and ZIP code.
- Each customer of the company is identified by a unique customer number, first and last name, and ZIP code.
- Each part sold by the company is identified by a unique part number, a part name, price, and quantity in stock.
- Each order placed by a customer is taken by an employee and is given a unique order number. Each order contains specified quantities of one or more parts. Each order has a date of receipt as well as an expected ship date. The actual ship date is also recorded.


EID
FirstName
ZipCode
CID
LastName
LastName
Employee
Customer
FirstName
ZipCode
1
1
Takes
Places
n
n
OrderNumber
Order
m
Contains
n
Part
QtyInStock
DateOfReceipt
PartNumber
Price
ExpectedShipDate
ActualShipDate
Qty
Name