Database Design

Overview

- What is database?
- Why do we bother?
- Relational database
- Entity-Relationship Modeling
- Mapping class diagrams to tables

What Is Database?

- A tool that stores data, and lets you create, read, update, and delete the data
- Information container
- Various types of database
  - Flat files
  - spreadsheets
  - XML
  - relational databases
    - mySQL, Oracle, DB2, Access

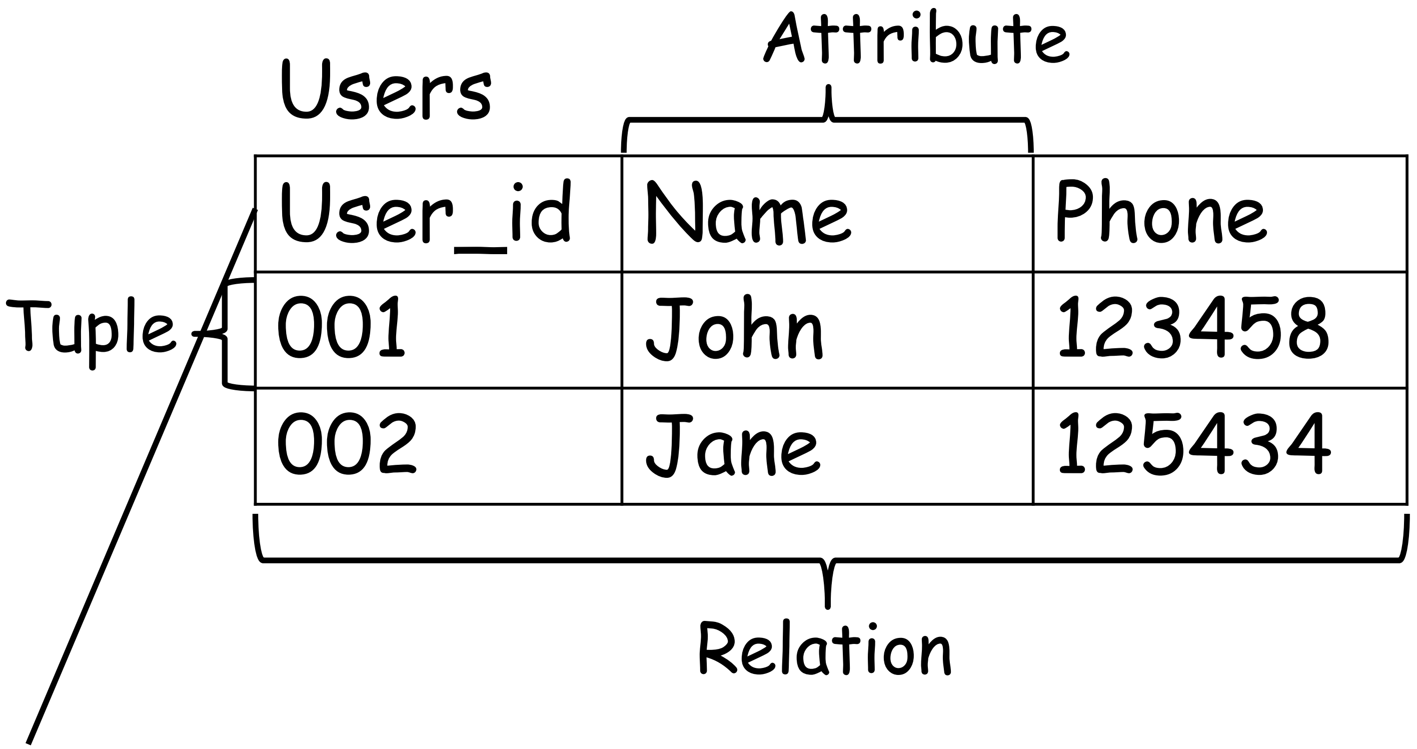
Why Do We Use Database?

- Every non-trivial application uses databases to keep program states, to store, manipulate, and retrieve data
- Database plays a critical role in applications
  - Corrupted data => execution failure
  - Poor data organization => poor performance
- A poorly designed database application allows developers to put in arbitrary data
  - Enter a string “none” as a phone number

Relational Database

- A digital database with a collection of tables
  - Each table contains rows and columns, with a unique key for each row
  - Each entity type described in a database has its own table
    - E.g., “Employee”, “Item”, “Order”
  - Each row represents an instance of the entity
    - E.g., “John Jenny”, “Soap”
  - Each column represents an attribute
    - E.g., “phone number”, “price”

Relational Databases (cont.)



Attribute

Users

User_id	Name	Phone
001	John	123458
002	Jane	125434

Tuple

Relation

Primary Key/Unique Key: to uniquely specify a tuple in a table

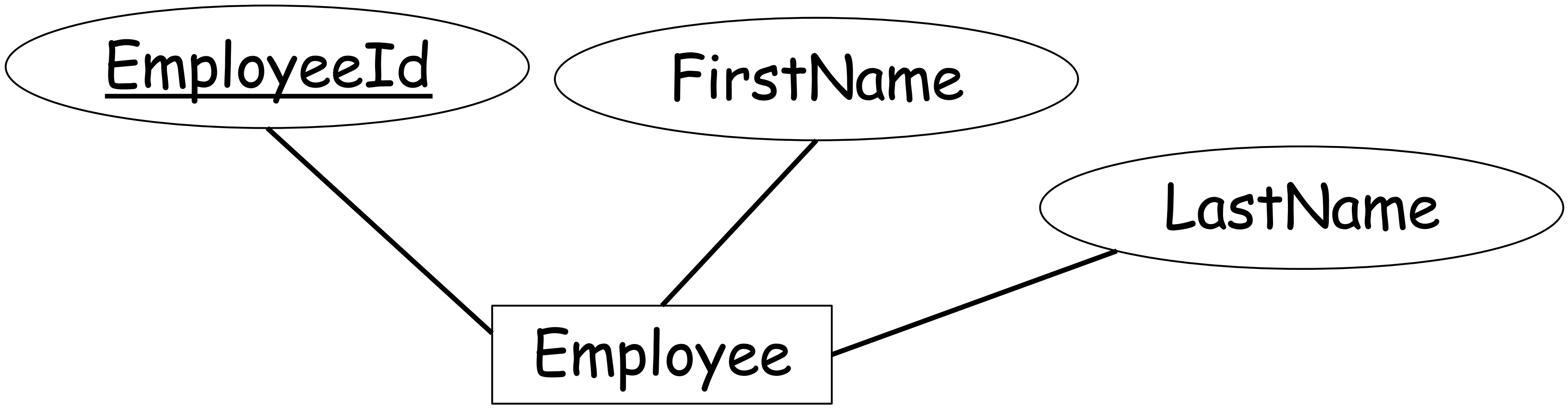
Foreign Key: an attribute in a relational table that matches the primary key column of another table. It can be used to cross-reference tables.

Entity-Relationship Models

- Entity-relationship (ER) diagrams are similar to semantic object modelings (class diagrams)
- It uses different notations
- Focuses more on relations and less on class structure

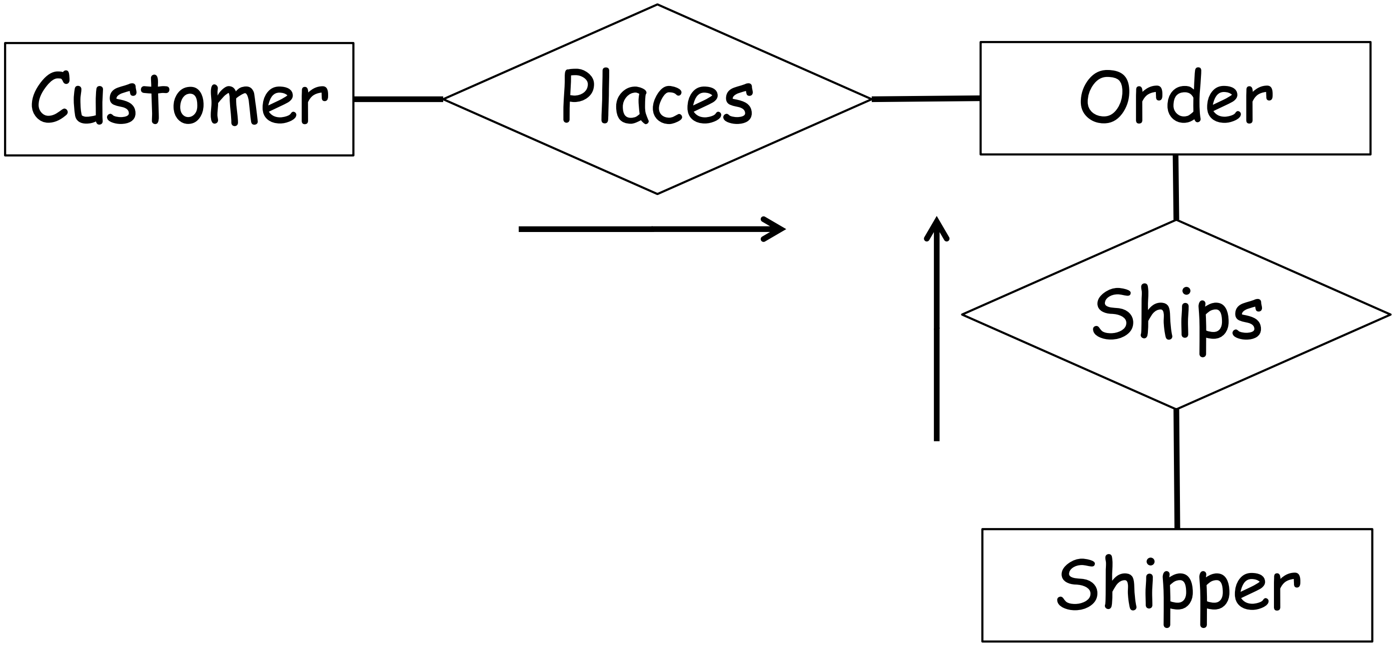
Entities and Attributes

- An entity is similar to a semantic object
- It includes attributes that describe the object



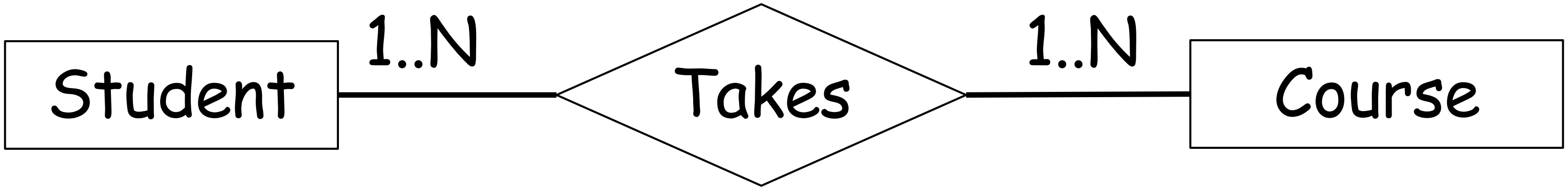
Relationships

- An ER diagram indicates a relationship between entities with a diamond
- Sometimes arrows are added to indicate direction of relationship



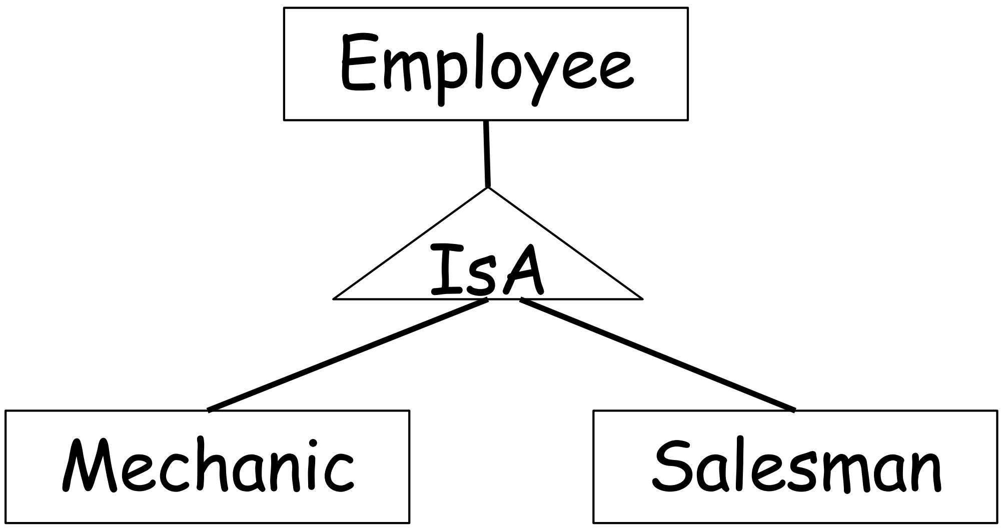
Cardinality

- Numbers used to describe relationship quantitatively



Inheritance

- A triangle named “IsA” represents the inheritance relationship



Mapping Class Diagrams to Tables

Mapping Classes to Tables

Course
courseId name description
…

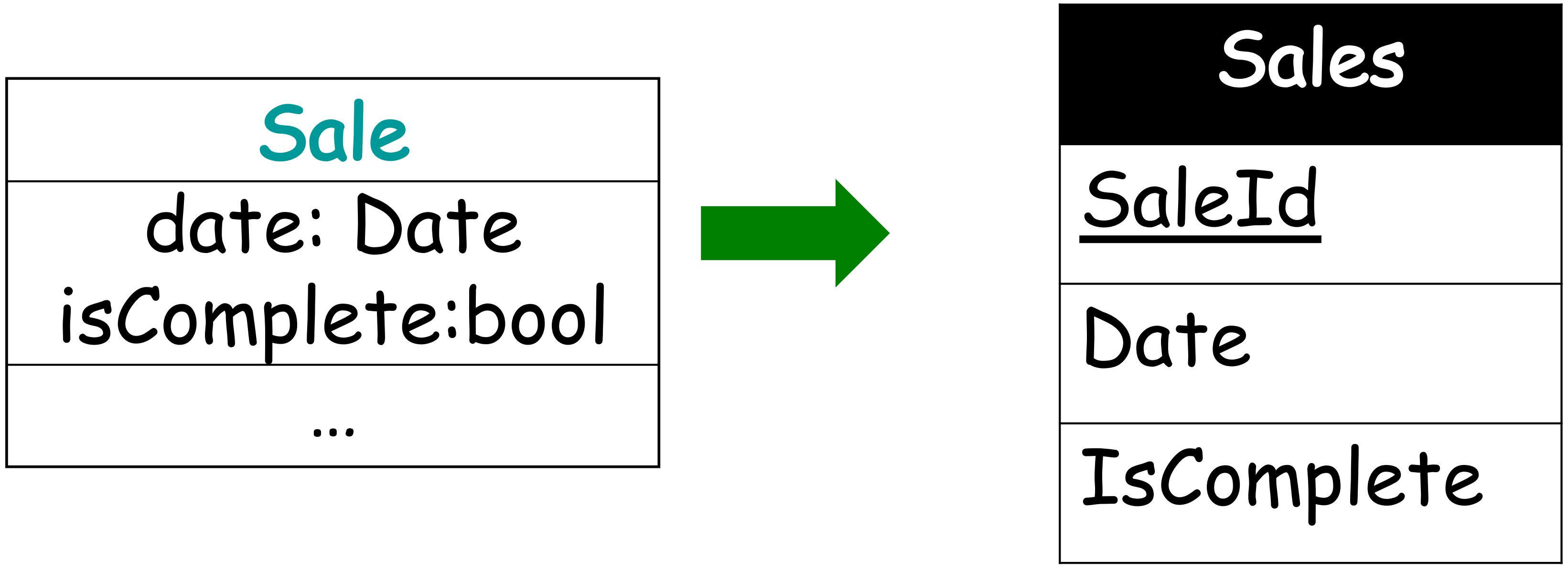
Student
studentId firstName lastName
…

Courses
CourseId
Name
Description

Students
StudentId
FirstName
LastName

Key Points about Tables

- Sometimes you need to explicitly add a primary key to distinguish data in tables
- Database usually provides functionality to automatically increment primary key



Sale

date: Date

isComplete:bool

…

Sales

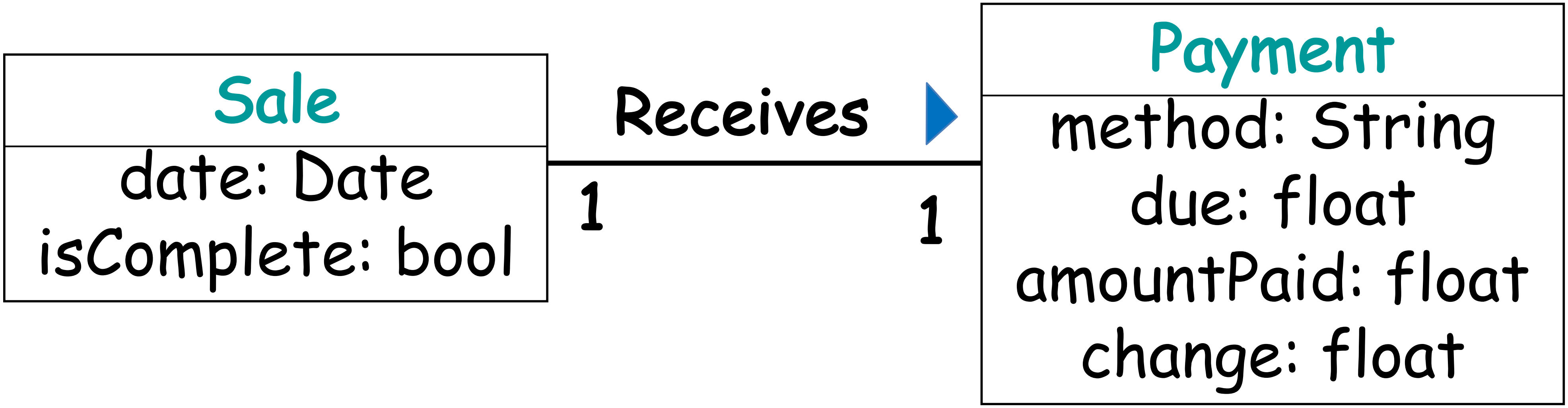
SaleId

Date

IsComplete

Mapping Associations

Mapping One-to-One Associations



Sale

date: Date
isComplete: bool

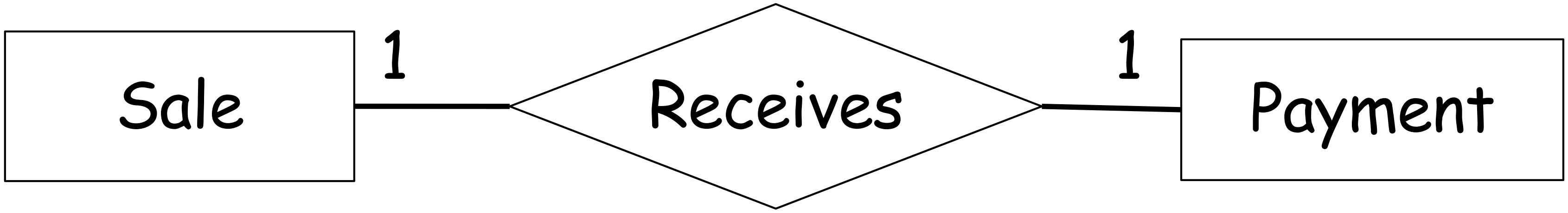
Receives ▶

1

1

Payment

method: String
due: float
amountPaid: float
change: float



Sale

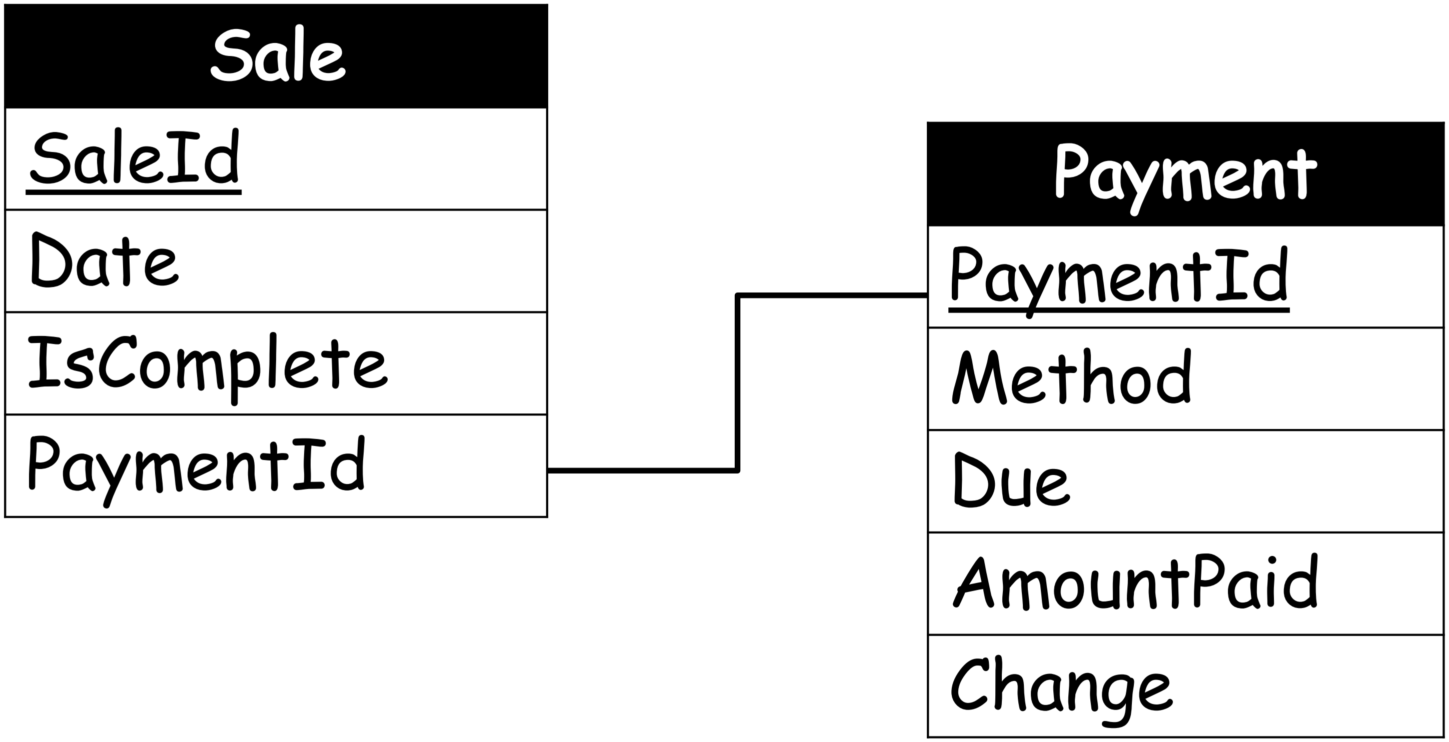
1

Receives

1

Payment

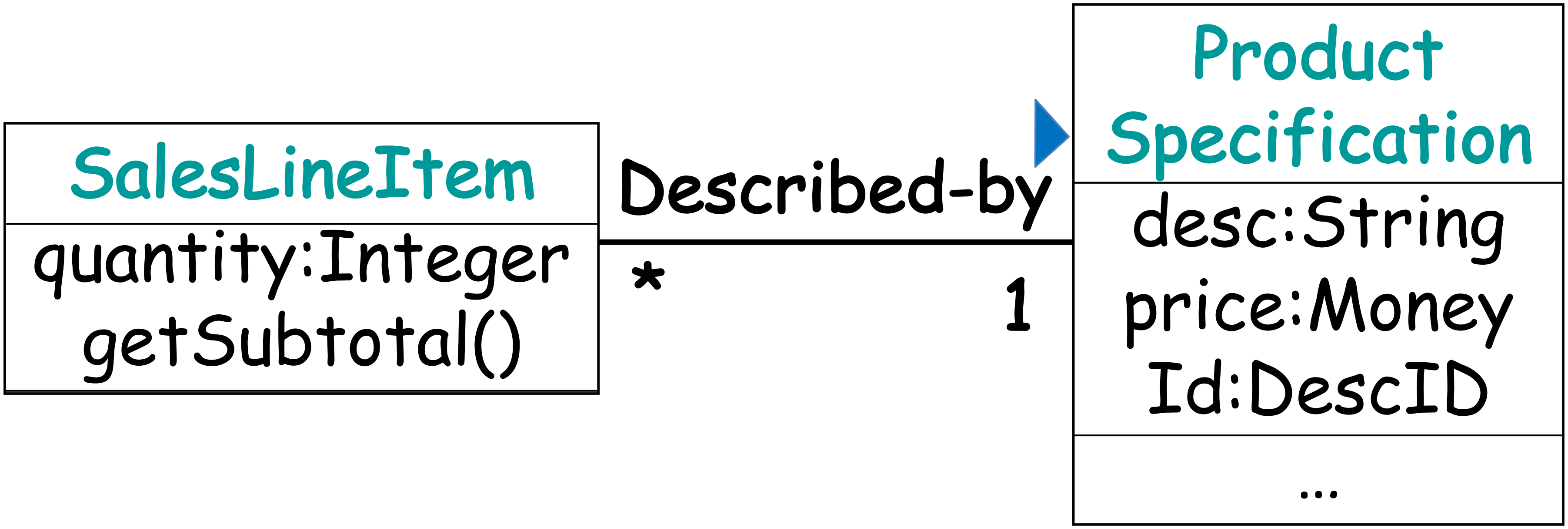
What Are the Tables?



Sale
SaleId
Date
IsComplete
PaymentId

Payment
PaymentId
Method
Due
AmountPaid
Change

Mapping One-to-Many Associations



SalesLineItem

quantity:Integer

getSubtotal()

Described-by

*

1

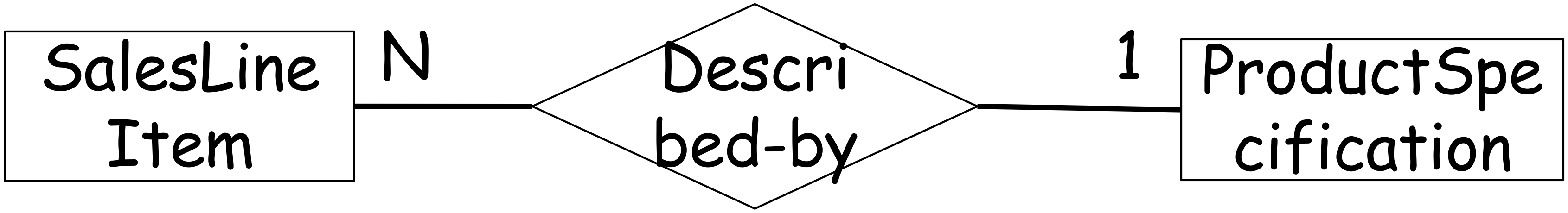
Product Specification

desc:String

price:Money

Id:DescID

...



SalesLine Item

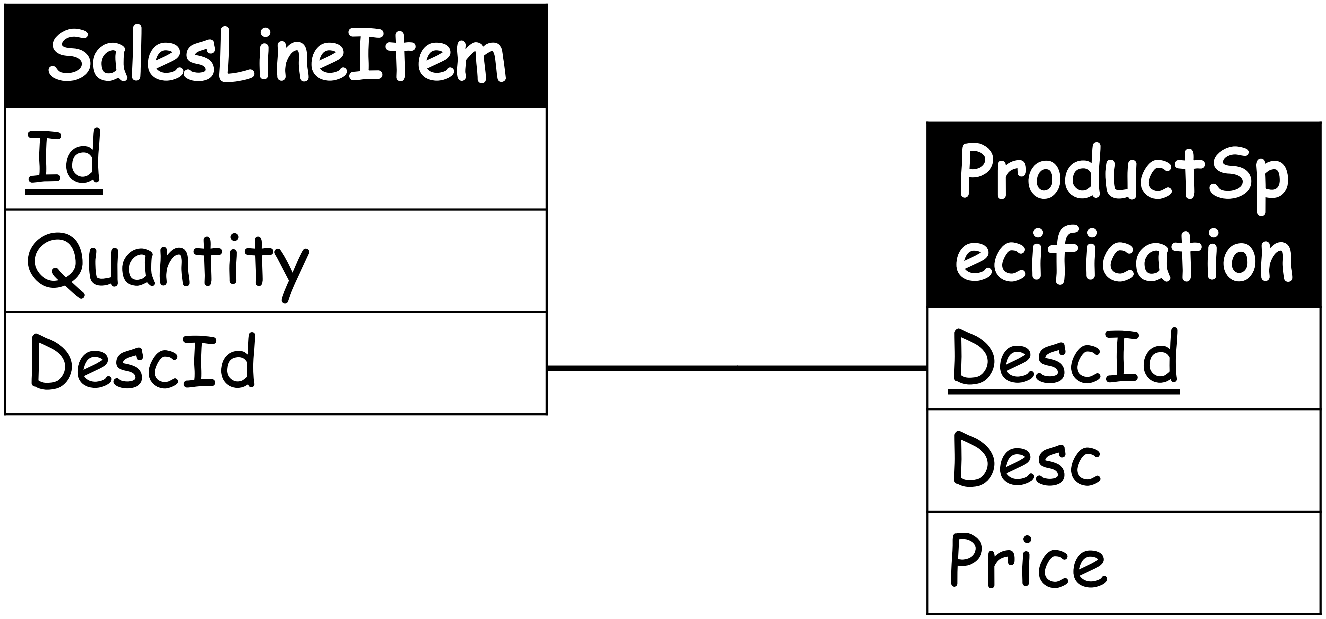
N

Described-by

1

ProductSpecification

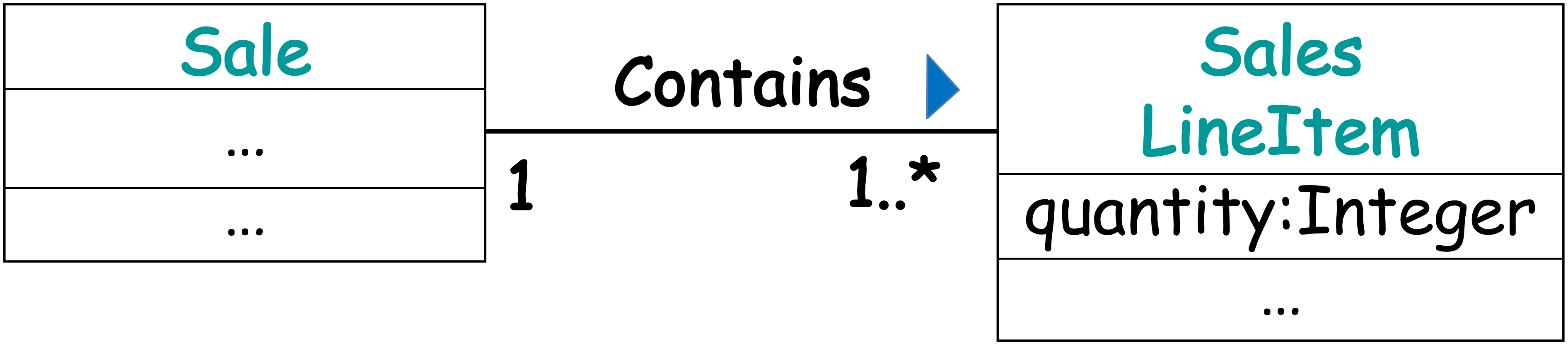
What Are the Tables?

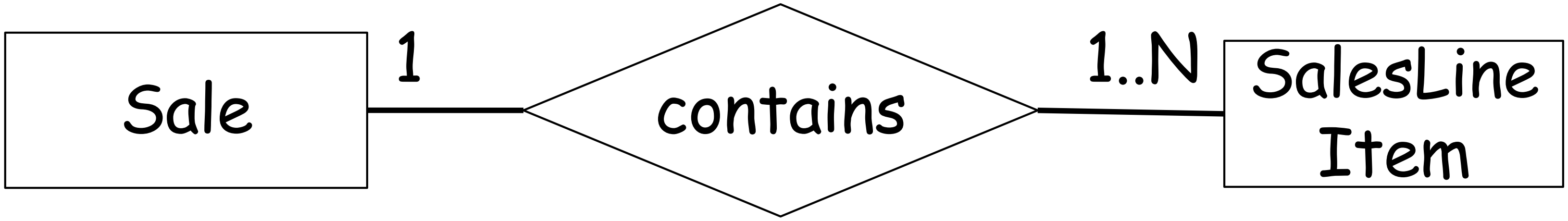


SalesLineItem
Id
Quantity
DescId

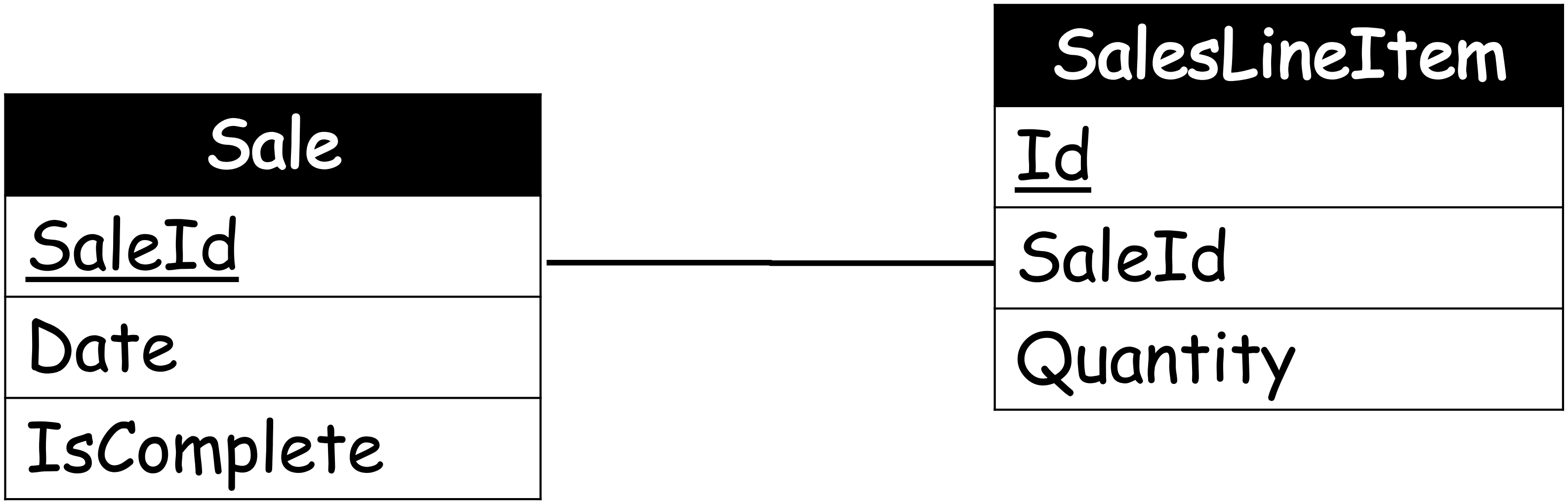
ProductSpecification
DescId
Desc
Price

Mapping Many-to-One Associations





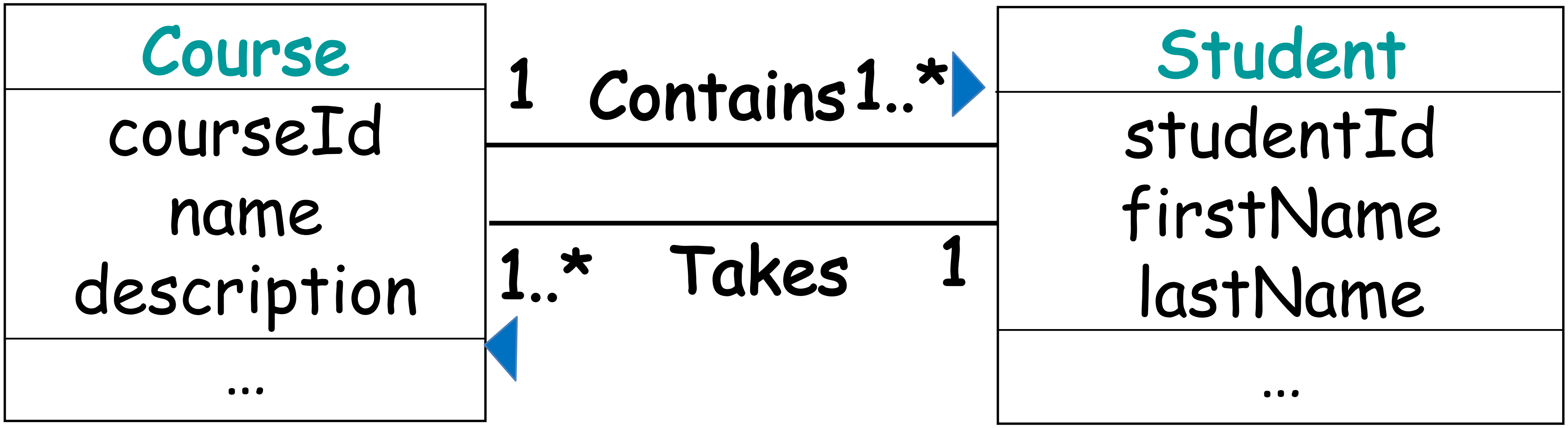
What Are the Tables?

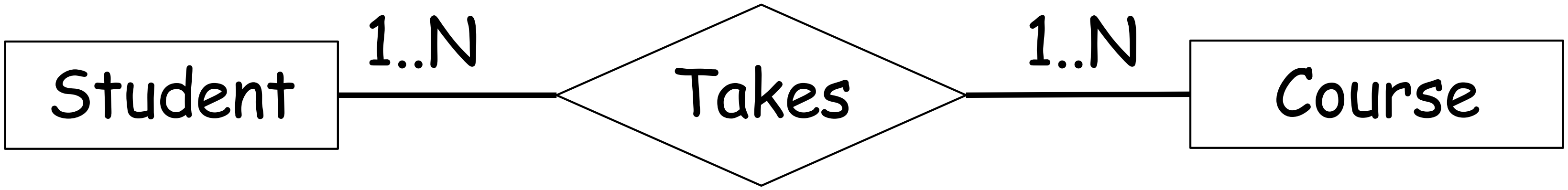


Sale
SaleId
Date
IsComplete

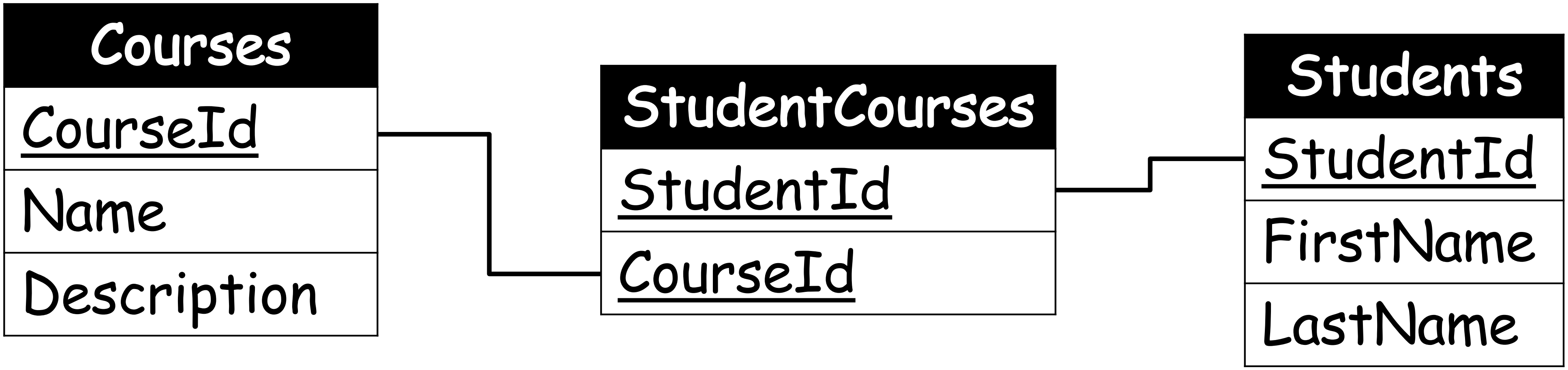
SalesLineItem
Id
SaleId
Quantity

Mapping Many-to-Many Associations to Table





What Are the Tables?



Courses

CourseId

Name

Description

StudentCourses

StudentId

CourseId

Students

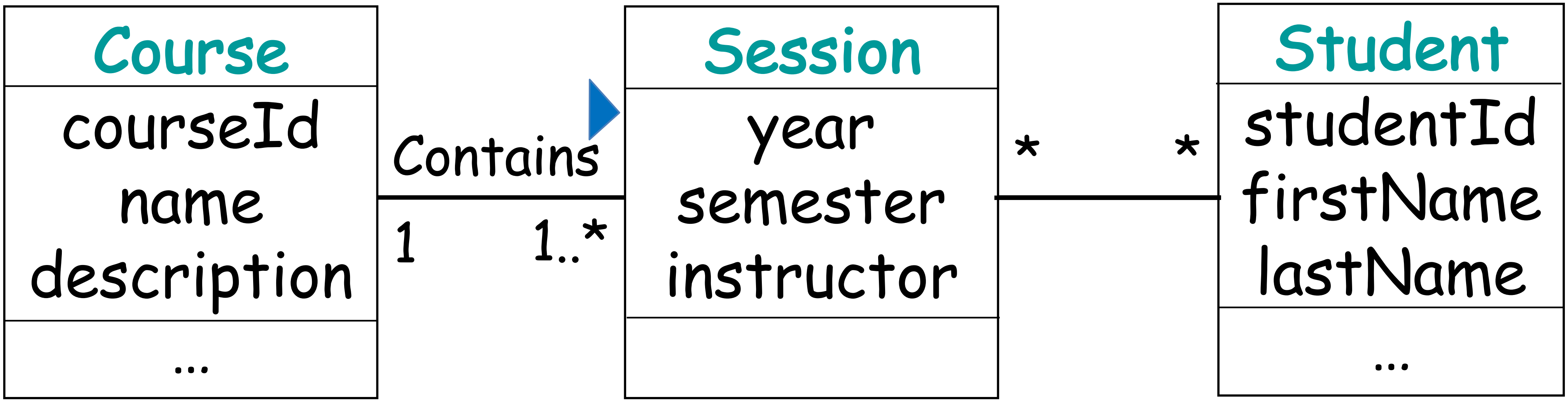
StudentId

FirstName

LastName

Multiple Many-to-Many

- What if we want to know students' enrollment over time for each year and semester
  - E.g., to distinguish students enrolled different time)?



Course

courseId
name
description

...

Contains

1

1..*

Session

year
semester
instructor

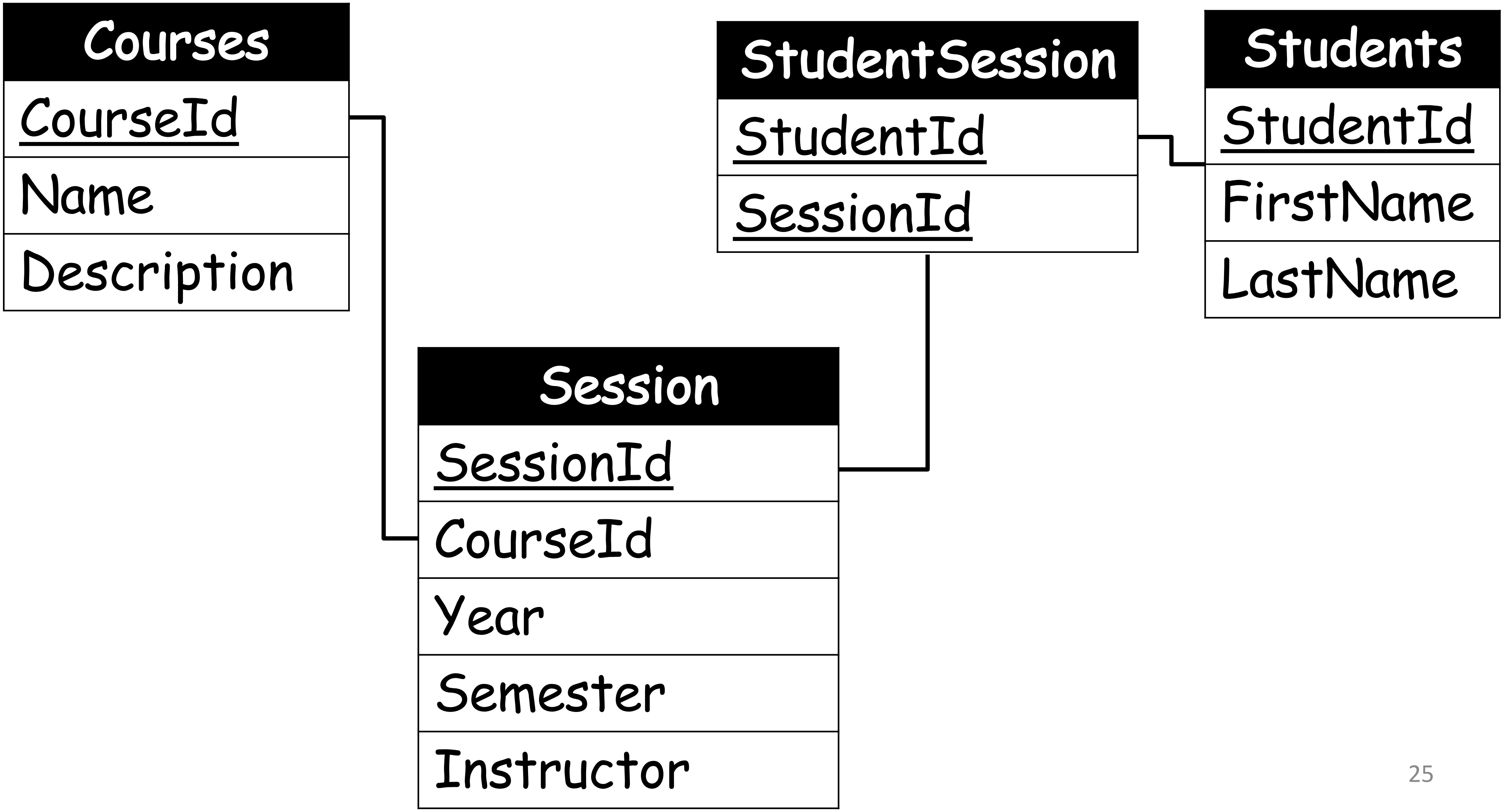
*

*

Student

studentId
firstName
lastName

...



Courses
CourseId
Name
Description

StudentSession
StudentId
SessionId

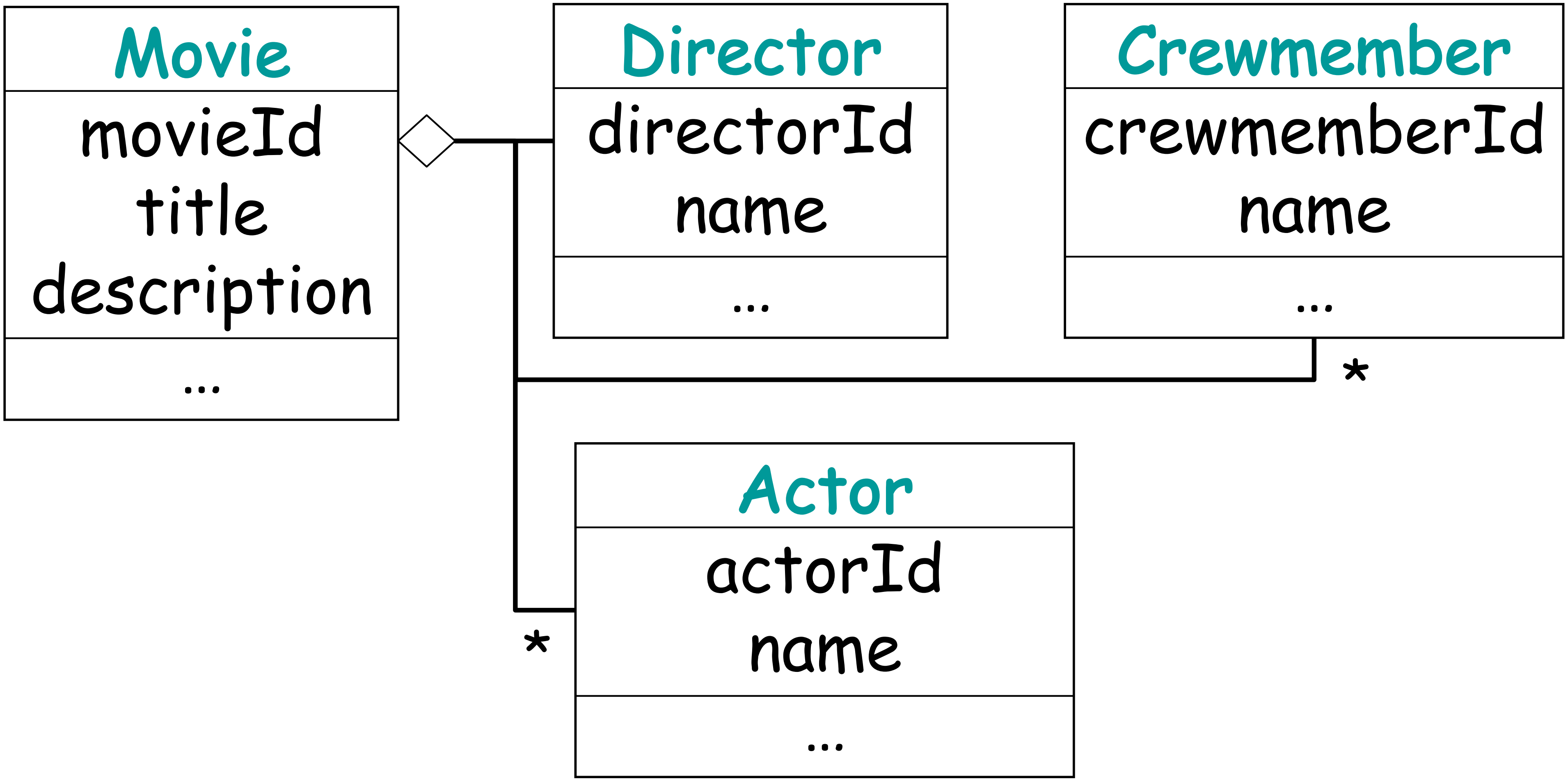
Students
StudentId
FirstName
LastName

Session
SessionId
CourseId
Year
Semester
Instructor

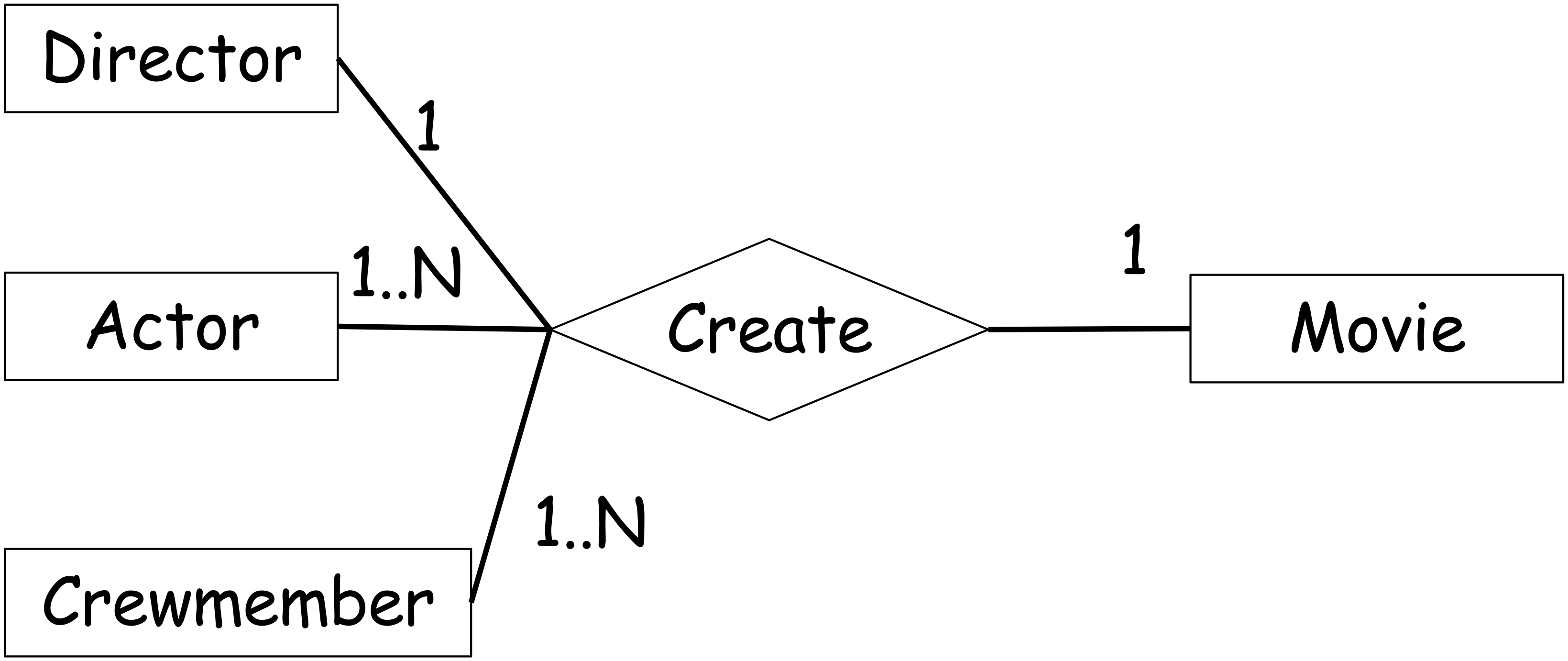
Multiple-Object Associations

- Definition
  - Many different kinds of objects are collectively associated with each other
- Case study
  - Making a movie requires a whole horde of people including a director, a bunch of actors, and a huge number of crew members

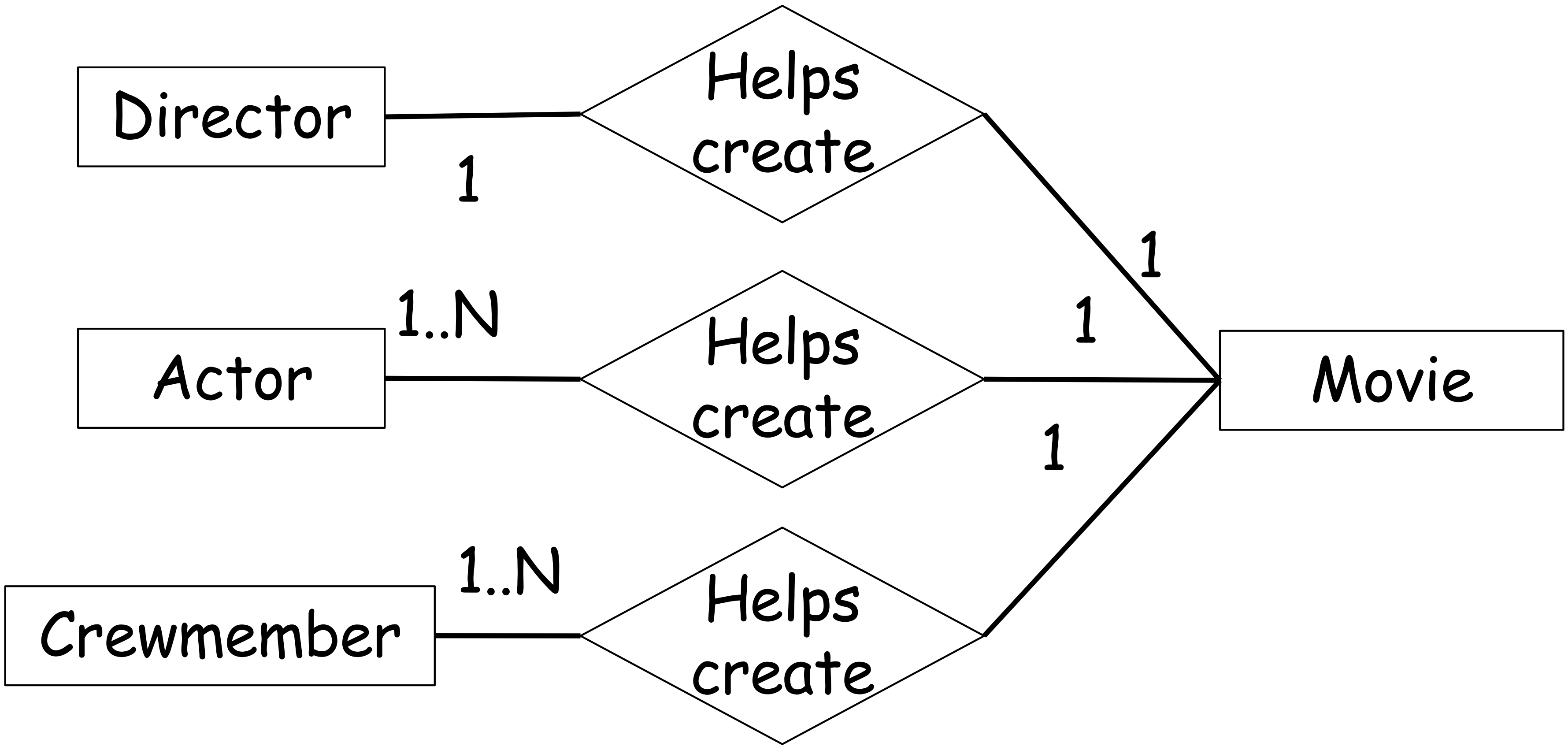
Class Diagram for Movie-making

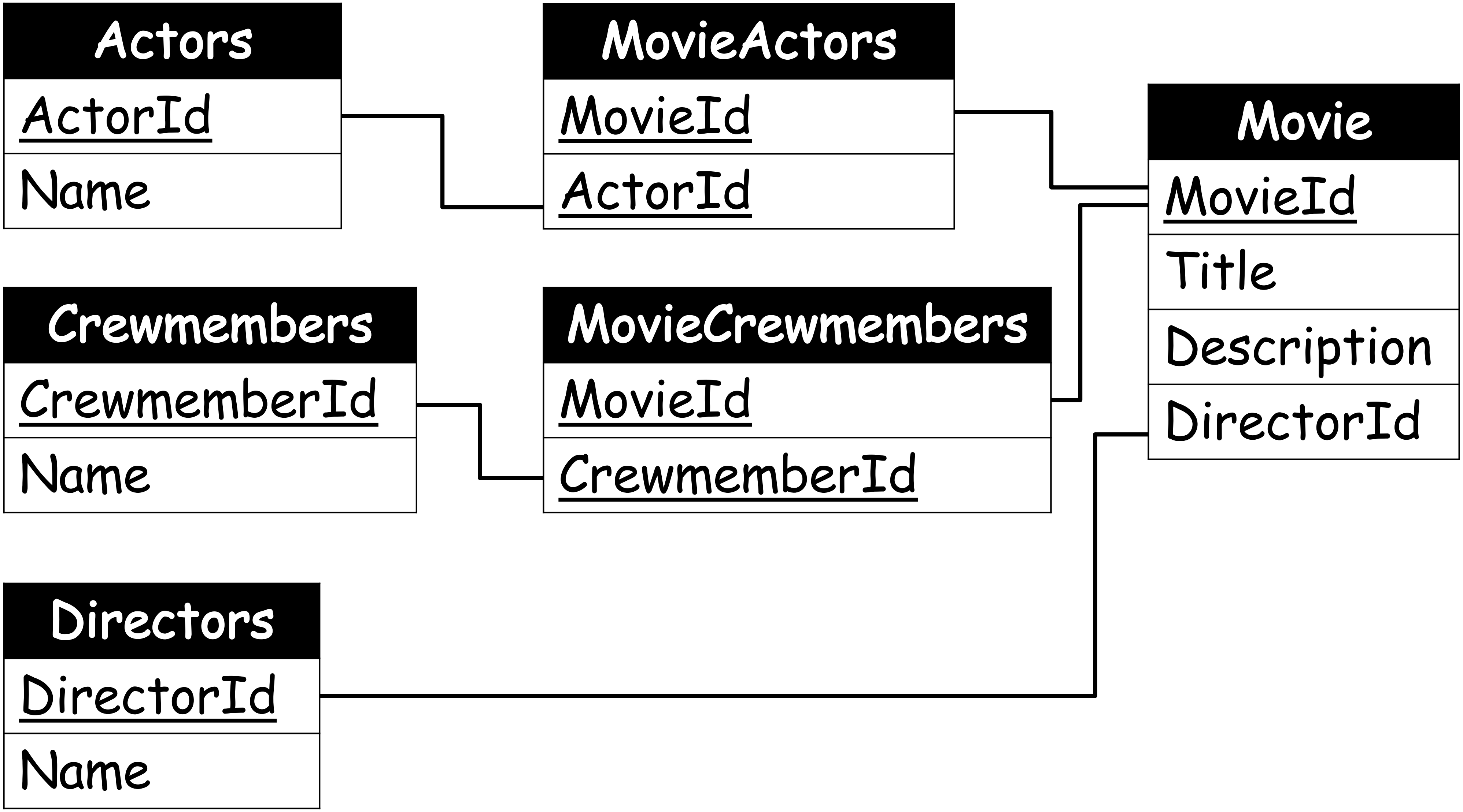


ER Diagram for Movie Making



Consider the Relationship as a Combination of Simpler Relationships





Actors
ActorId
Name

MovieActors
MovieId
ActorId

Movie
MovieId
Title
Description
DirectorId

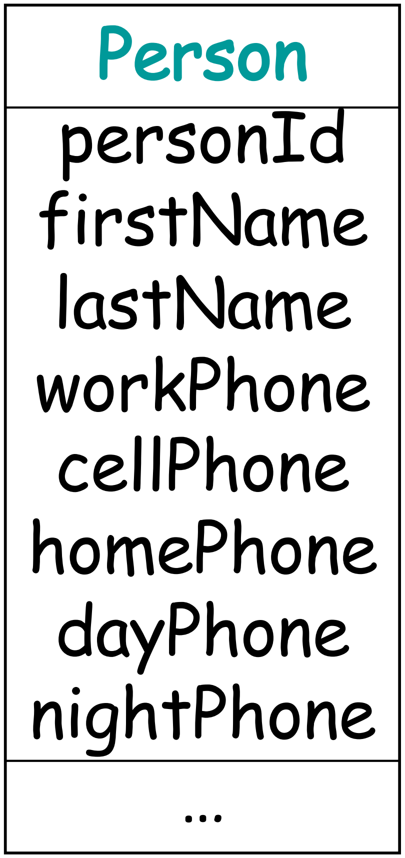
Crewmembers
CrewmemberId
Name

MovieCrewmembers
MovieId
CrewmemberId

Directors
DirectorId
Name

Repeated Attribute Associations

- Some entities have multiple attributes that represent either the same kind of data or very similar kinds of data
  - Some people may have multiple phone numbers for different purposes
  - Some people may have only one number to serve all purposes

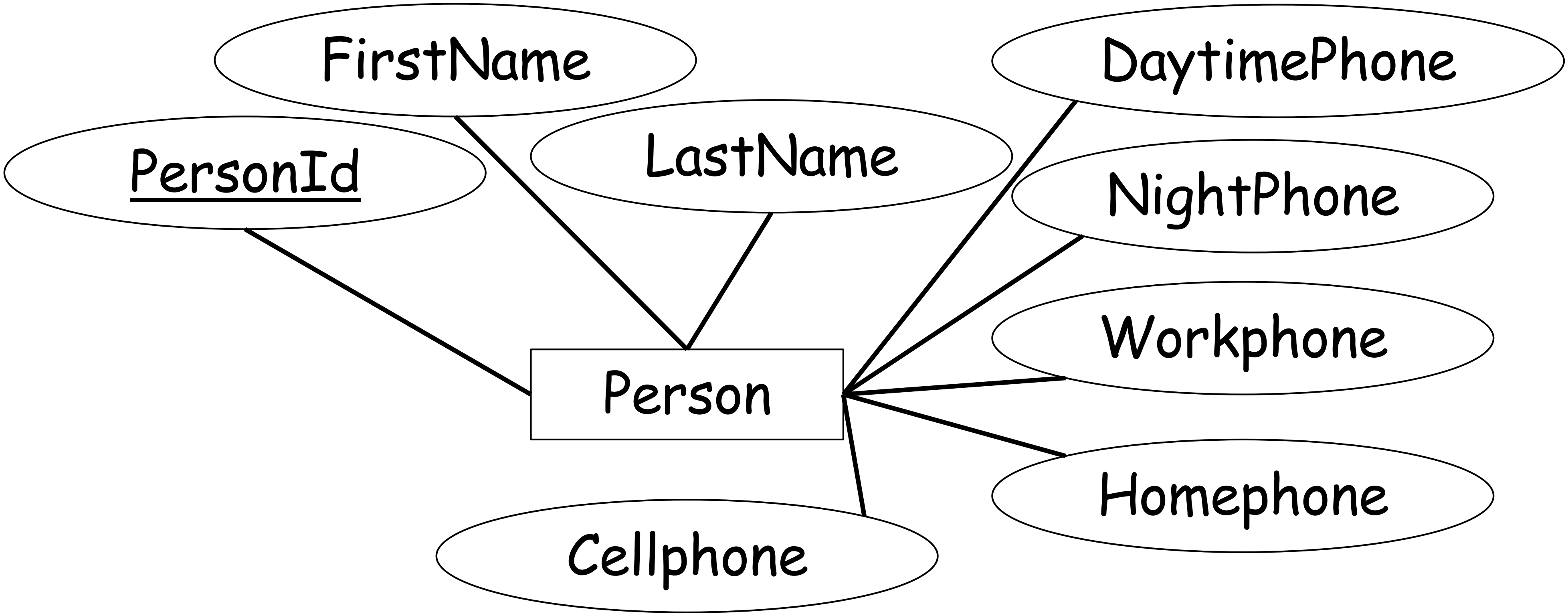


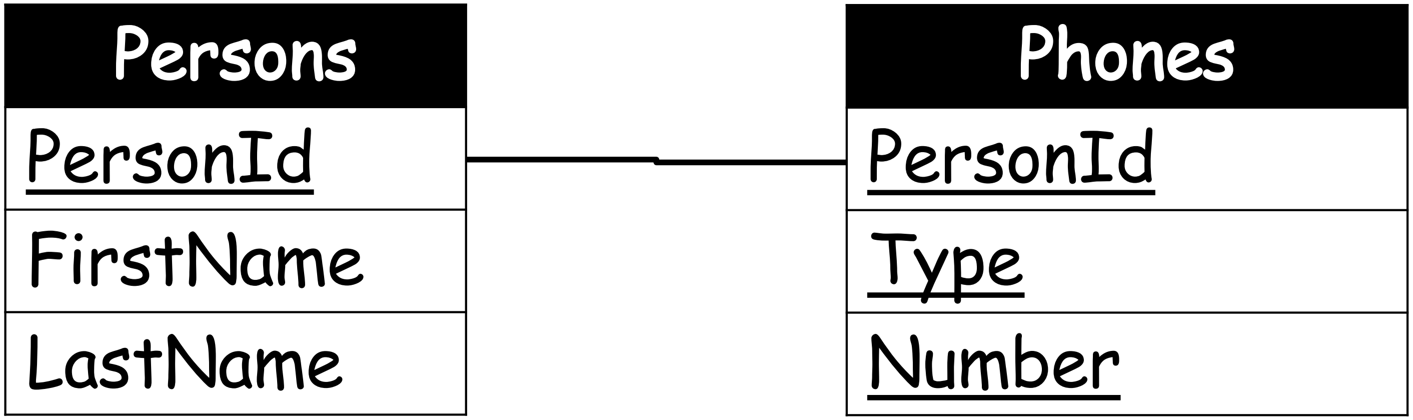
Person

personId
firstName
lastName
workPhone
cellPhone
homePhone
dayPhone
nightPhone

…

How to design the tables to avoid repetition or sparse data?





Persons

PersonId

FirstName

LastName

Phones

PersonId

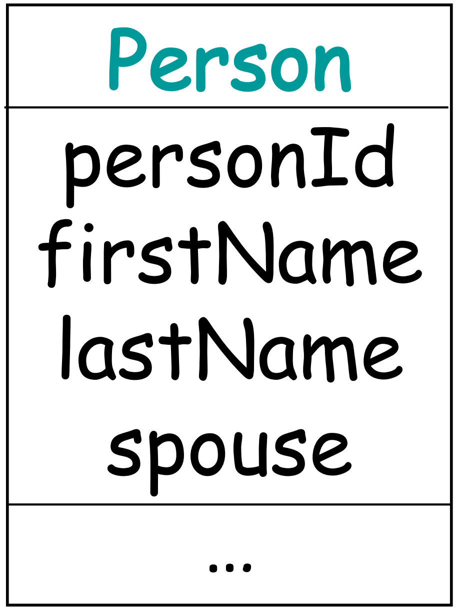
Type

Number

Reflexive Associations

- An object refers to an object of the same class
  - One-to-One reflexive association
  - One-to-Many reflexive association
  - Many-to-Many reflexive association

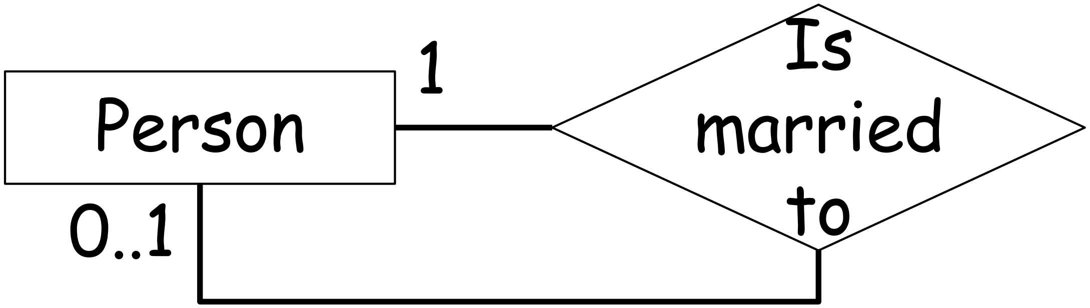
One-to-One Reflexive Association



Person

personId
firstName
lastName
spouse

...

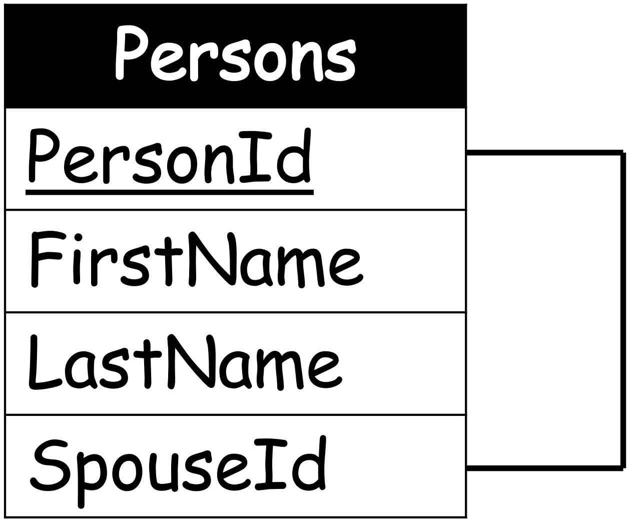


Person

1

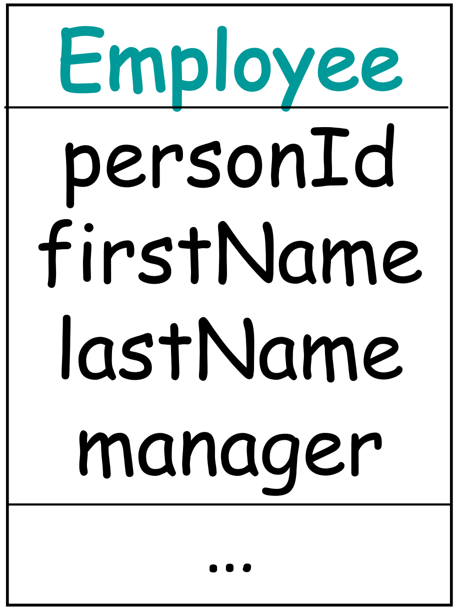
Is married to

0..1



Persons
PersonId
FirstName
LastName
SpouseId

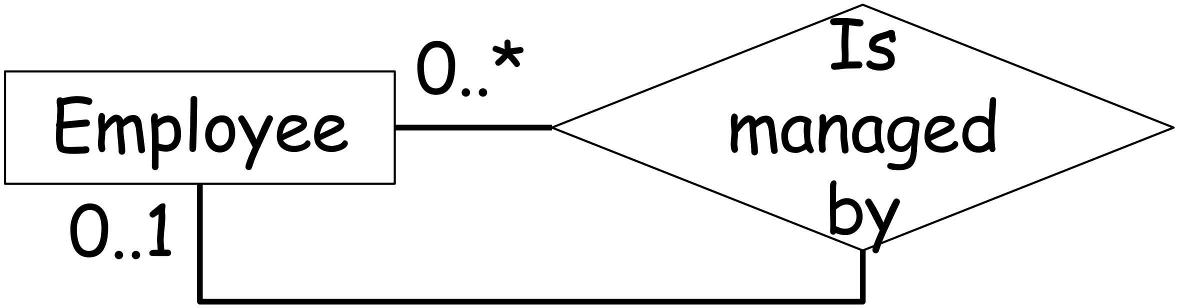
One-to-Many Reflexive Association

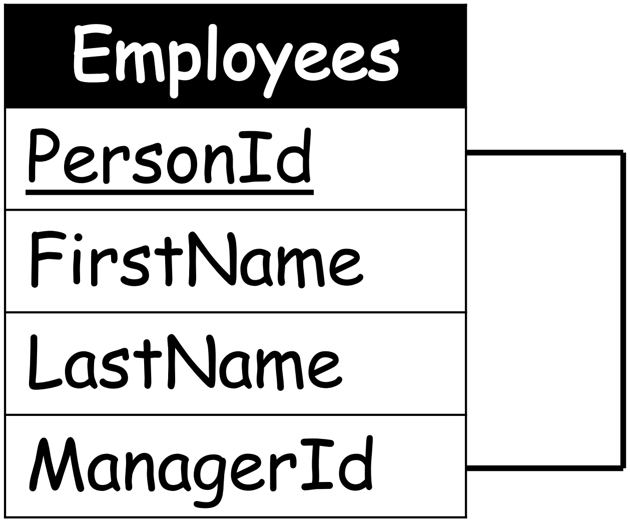


Employee

personId
firstName
lastName
manager

...





Employees

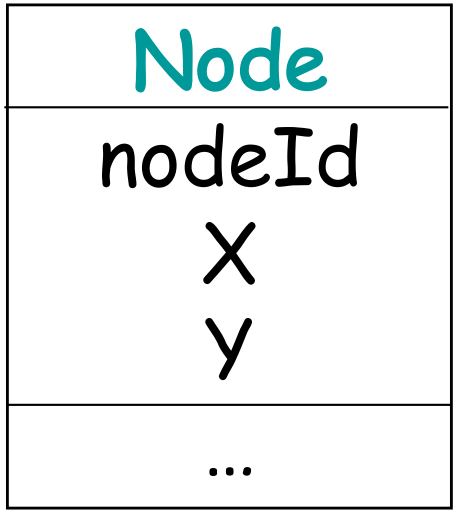
PersonId

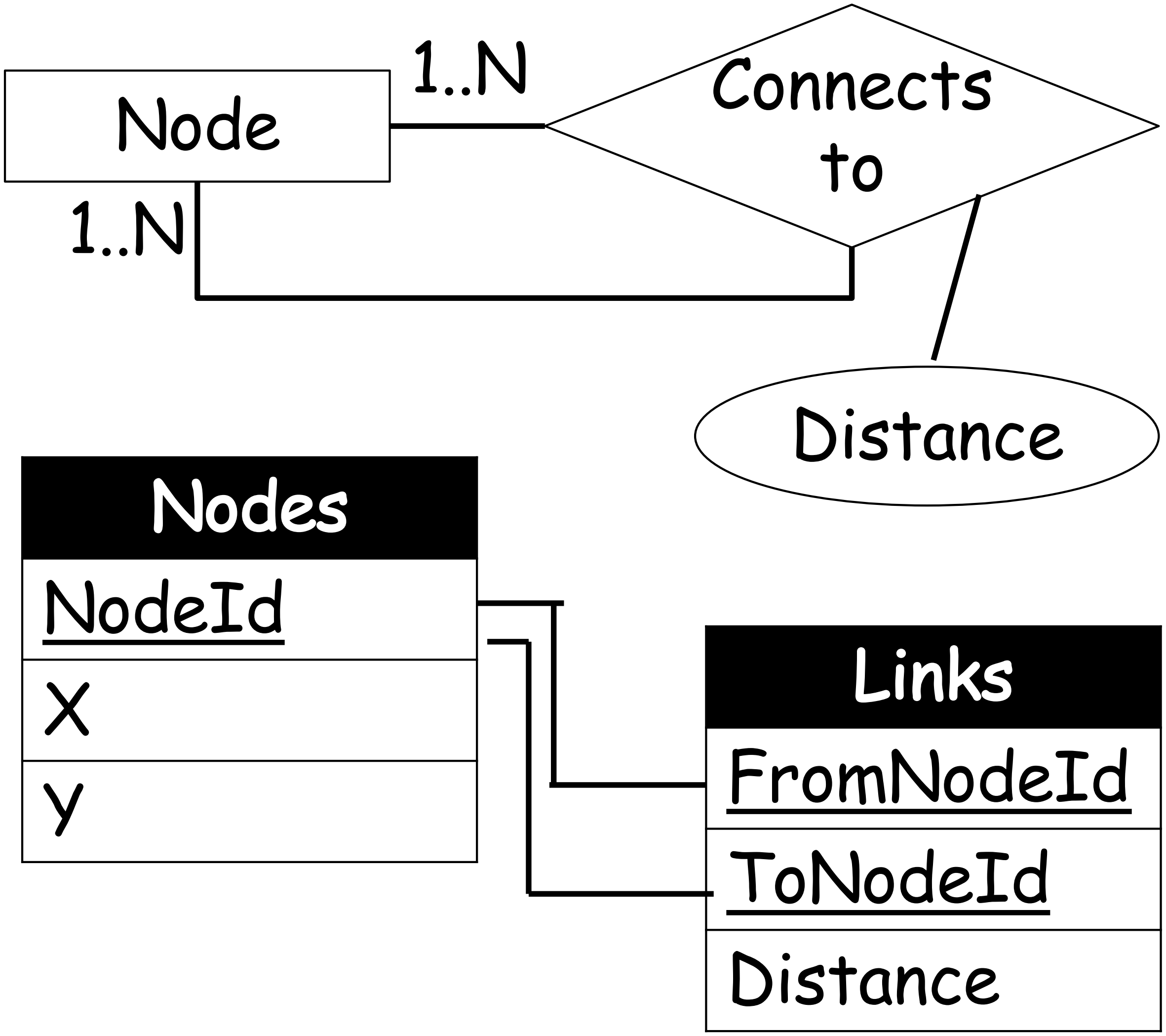
FirstName

LastName

ManagerId

Many-to-Many Reflexive Association

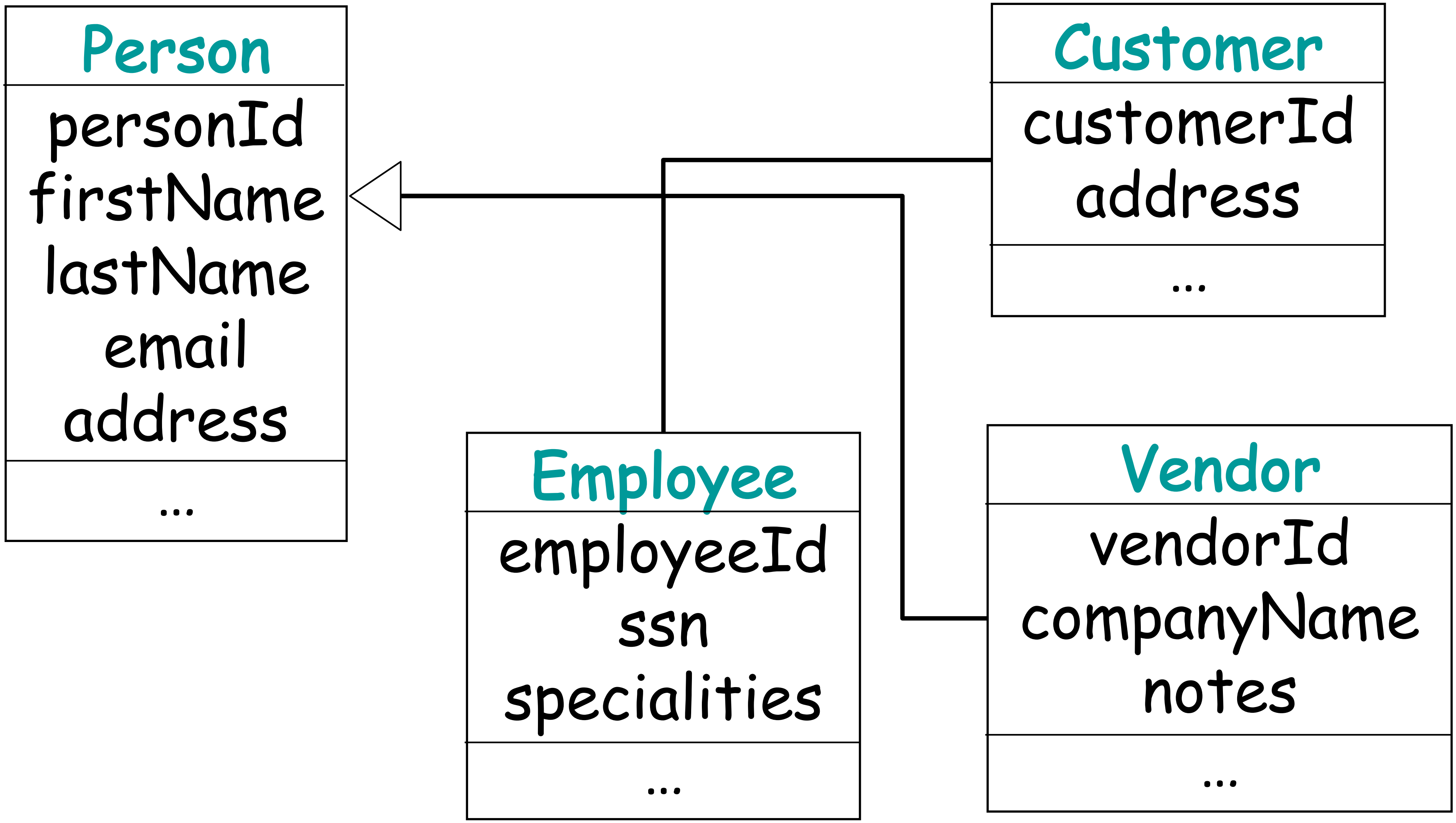




Nodes
<u>NodeId</u>
X
y

Links
<u>FromNodeId</u>
<u>ToNodeId</u>
Distance

Inheritance



Person

personId
firstName
lastName
email
address

...

Customer

customerId
address

...

Employee

employeeId
ssn
specialities

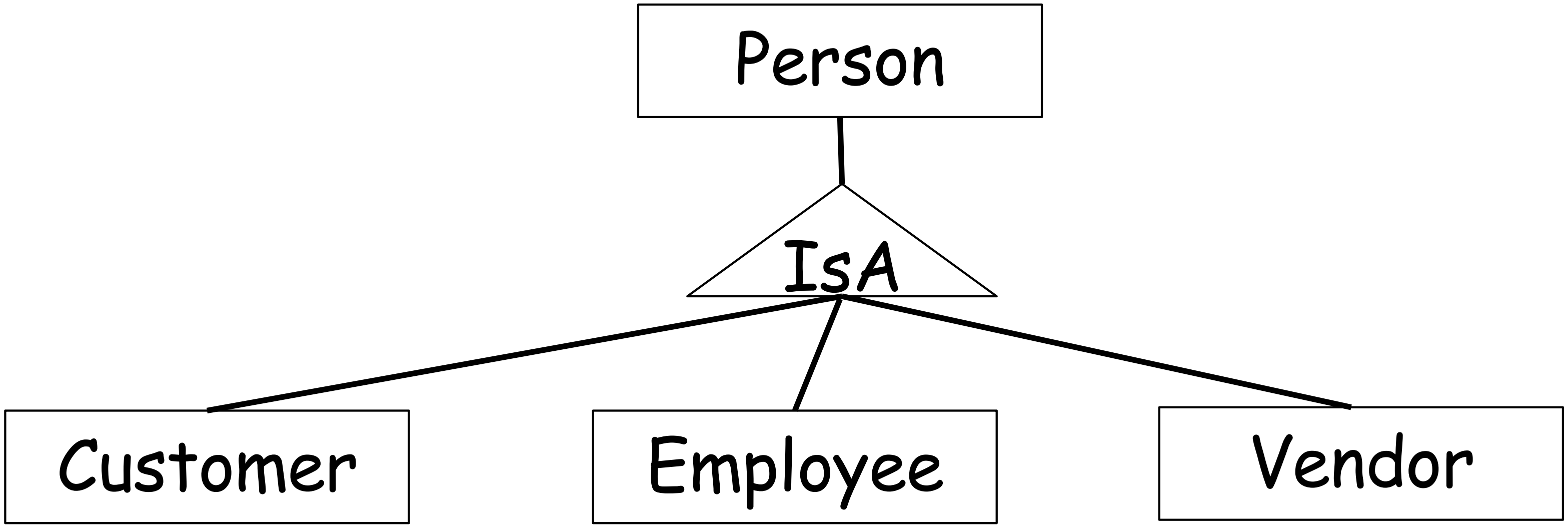
...

Vendor

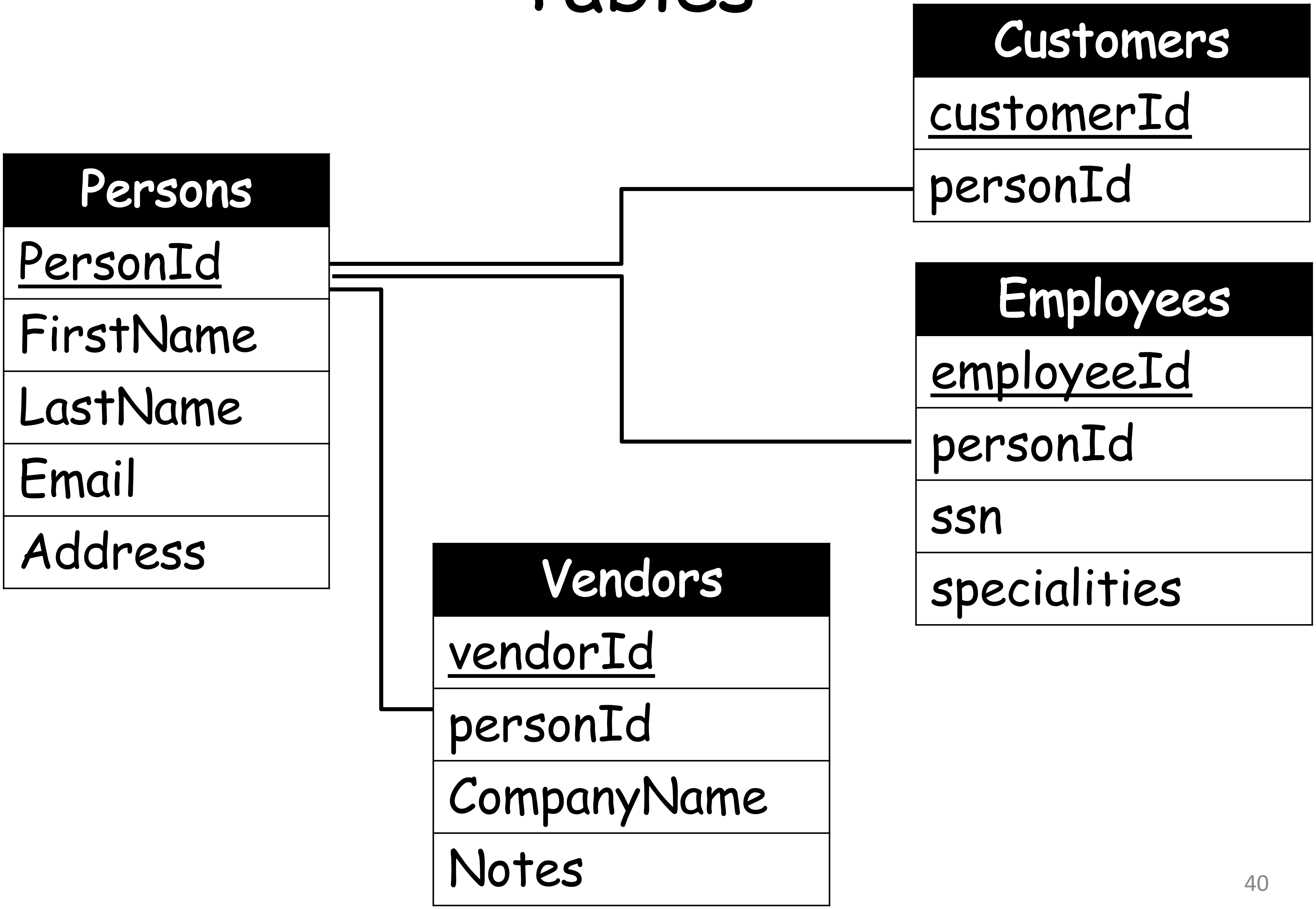
vendorId
companyName
notes

...

ER-Diagram



Tables



Persons
PersonId
FirstName
LastName
Email
Address

Customers
customerId
personId

Employees
employeeId
personId
ssn
specialities

Vendors
vendorId
personId
CompanyName
Notes