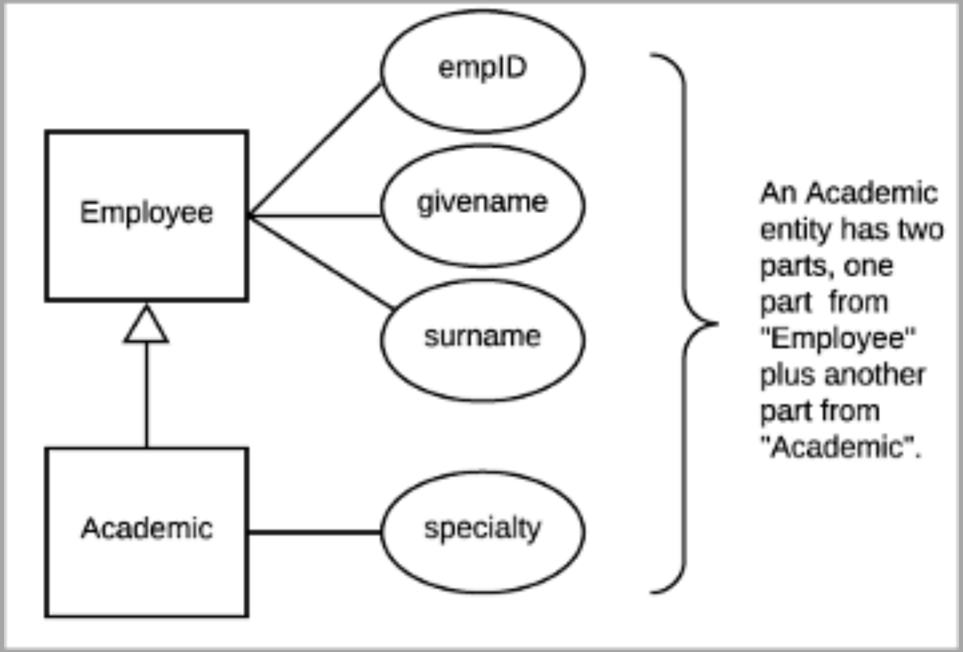
The Entity-Relationship Model 2: Advanced concepts and Mapping ERDs

- Reading:
    - The “High-level Database Models” chapter.

Subclasses in the ER Model

- An entity set includes some entities that have special attributes not associated with all entities.
  - Some employees are academics having the special attribute “specialty”.
- A special entity set, or subclass, is defined with its *own attributes* and possibly relationships. The subclass has the “isa” (in other words “is a”, or specialisation) relationship with the original entity set.
- An entity set can have several subclasses. But a subclass entity set has only one direct superclass entity set.
- Data for entities of a subclass entity set comprise its own attributes as well as all attributes from its superclass entity sets.

Subclasses in the ER model: Example


empID
Employee
givename
surname
Academic
specialty
An Academic entity has two parts, one part from "Employee" plus another part from "Academic".

The subclass concept is about “specialisation”.

Subclasses in UML

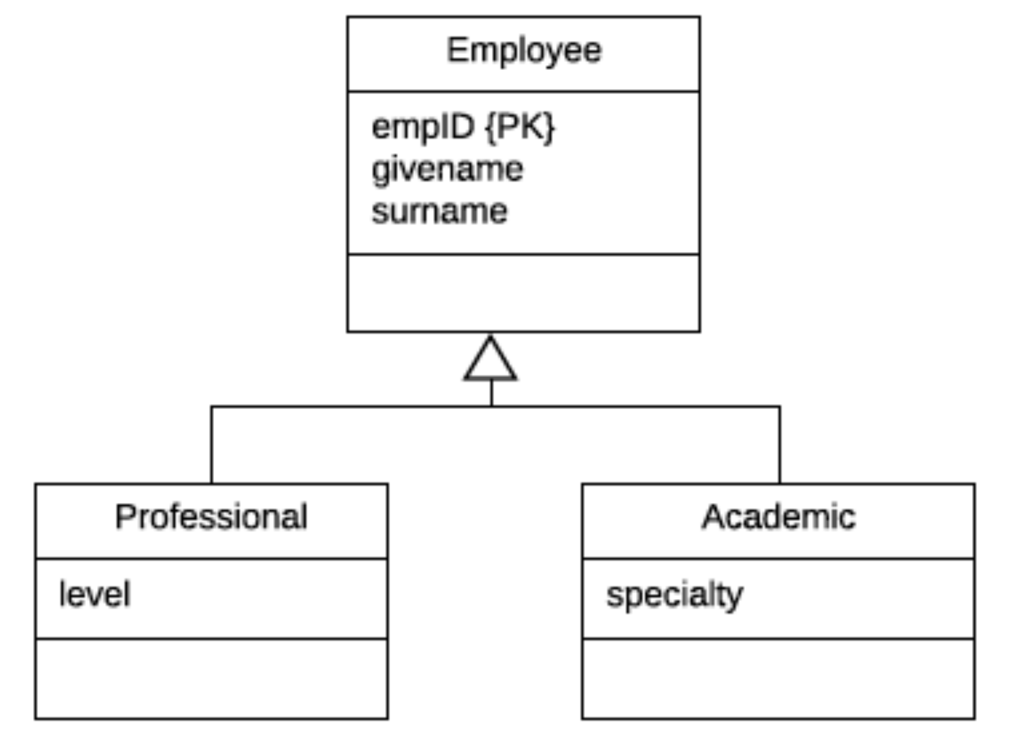
UML is an object-oriented modelling language. The subclass concept is based on “inheritance”, different from “specialisation” of the ER model.

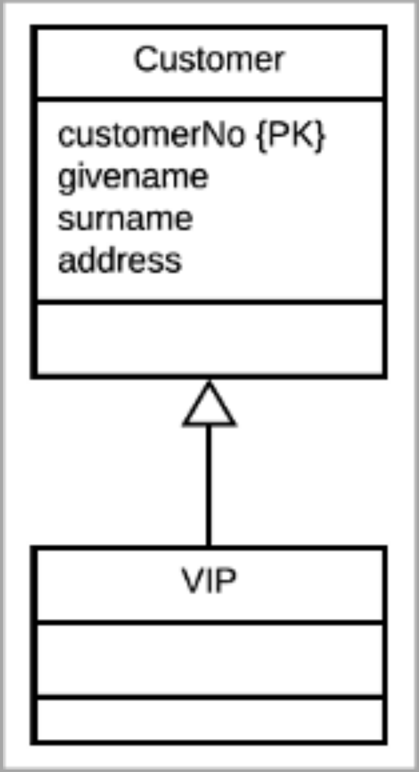
- A subclass inherits attributes and associations from its superclass.
- An object belongs to one class, and also conceptually “belongs” to its superclass due to inheritance.

UML has four types of subclasses:

- *Complete* or *Partial*. If every object in a class is a member of some subclass the subclasses are complete; otherwise partial.
- *Disjoint* or *Overlapping*. If an object can be in two or more (sub)classes, the subclasses are overlapping; otherwise disjoint.
    - Typically the object-oriented modelling only allows disjoint subclasses.

Subclasses in UML: Examples





Complete and disjoint subclasses

Partial subclass

Weak Entity Sets

- Occasionally, entities of an entity set need “help” to identify them uniquely.
- Entity set *E* is said to be weak if in order to identify entities of *E* uniquely, we need to follow relationships from *E* to some other supporting entity sets and include the key of entities of the connected entity sets.

Weak Entity Set: Example

A TV-show like The Simpsons, South Park, or Neighbours has many episodes.

- Some show episodes do not even have a title, and so title can not, and usually is not used to uniquely identify a show episode.
- Season number and Episode number are certainly not a key, since two episodes from different shows can have the same Season and Episode numbers.
- But season and episode numbers together with the show name where the episode belongs to should be unique.

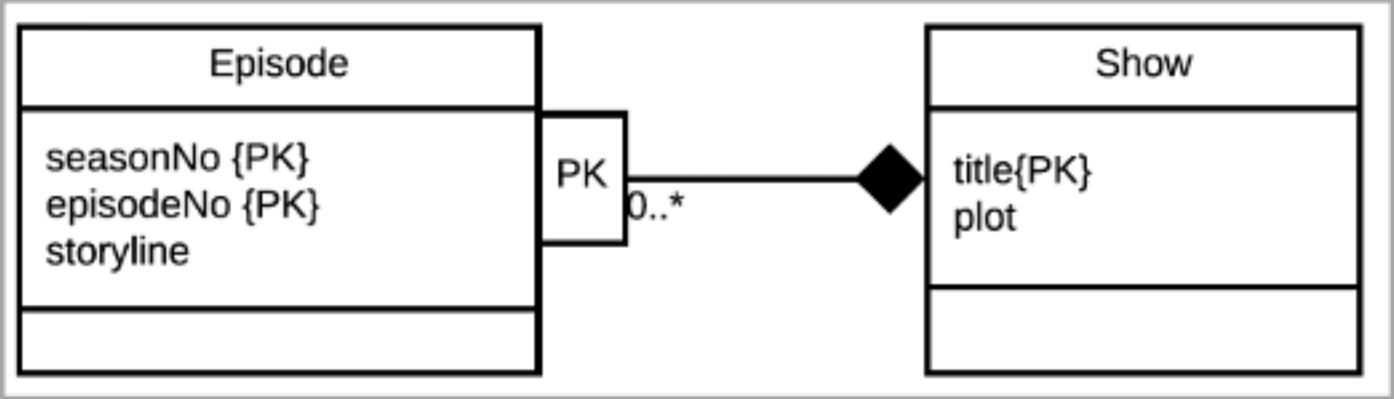
The UML representation for weak entity sets

- A special “supporting” composition relationship notation can be used to represent the “supporting” relationship between the weak entity set and the supporting entity set.
    - A “PK” box indicates “supporting” composition.
- The key for a weak entity set comprises of its own PK attributes as well as PK attributes of its supporting entity set(s).

The UML representation for weak entity sets: Example

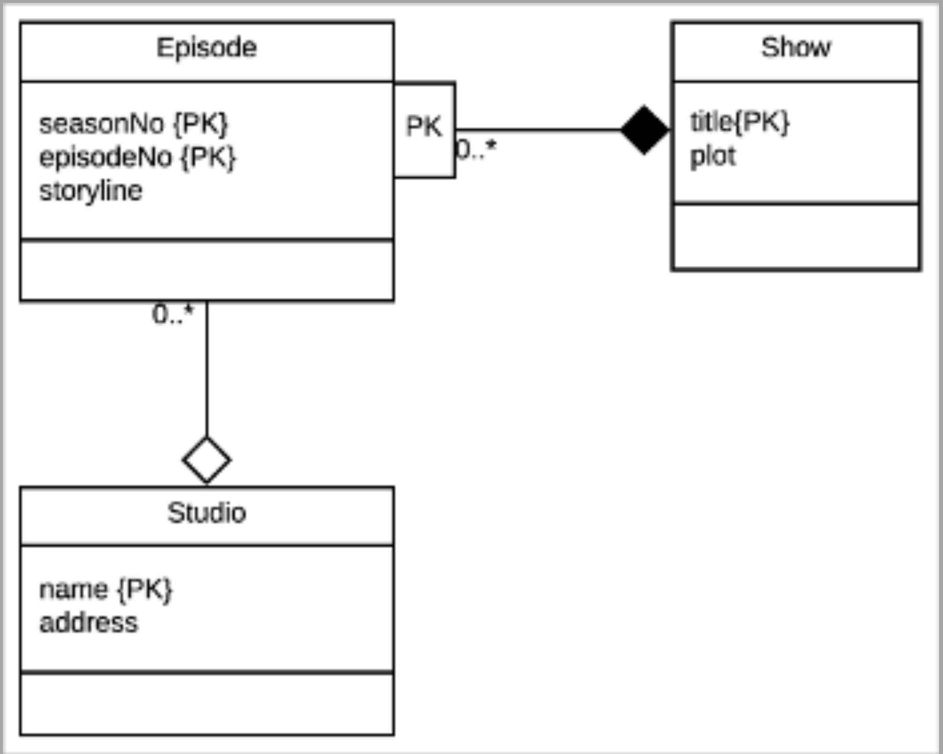
The Episode weak entity set is associated with the supporting "Show" entity set via the supporting composition relationship.

- The "PK" box at the left end indicates that the complete key for Episode comprises of PK attributes of Episode as well as the PK attribute of "Show".
- The solid diamond at the right end indicate "exactly one".



Weak entity sets can have regular relationships: Example

- Aggregation from the Episode weak entity set to Studio.



Multiway Relationships

- Multiway relationships connect more than two entity sets.
- Example: “A customer service consultant often books a delivery appointment with a delivery company for a customer”, which is a ternary relationship among entity sets Consultant, Delivery-company and Customer:
  - There do not exist any binary “booking” relationships between any two entity sets.

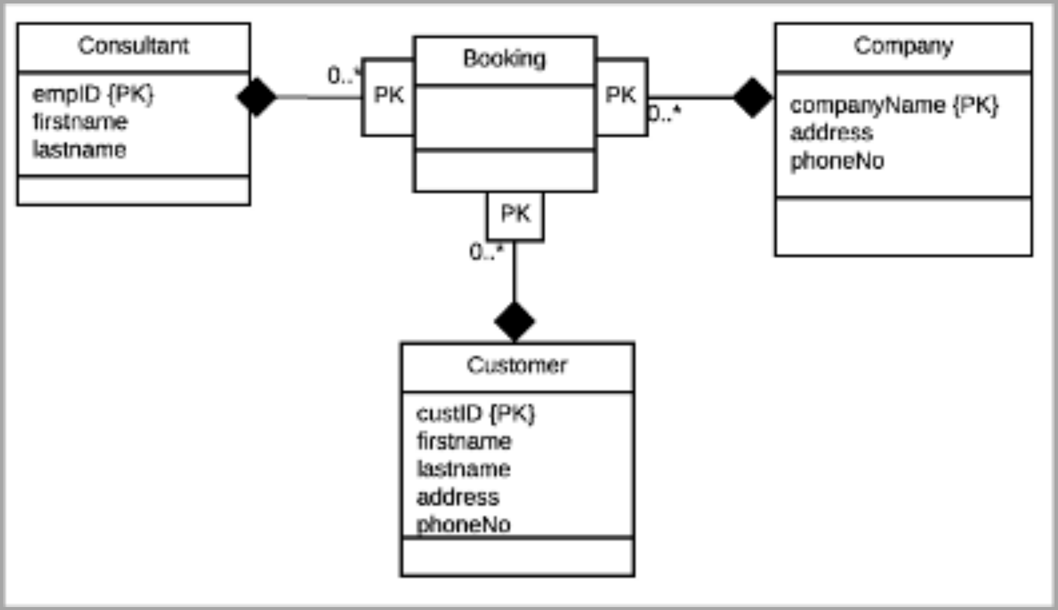
Converting multiway relationships to binary

- The ER model allows multiway relationships conceptually, but UML only has binary relationships.
- Converting multiway relationships to binary: create a weak entity set and then relationships from it to the entity sets in the original multiway relationship.

Example: Converting a ternary relationship to binary

The ternary relationship “A customer service consultant often books with a delivery company for a customer”:

- A weak entity set “Booking” is created
- Binary relationships between Booking and entity sets Consultant, Company and Customer.



Note that in this example the weak entity set “Booking” does not have attributes of its own. However, in other cases a weak entity could for example also have its own attributes (key or non-key)

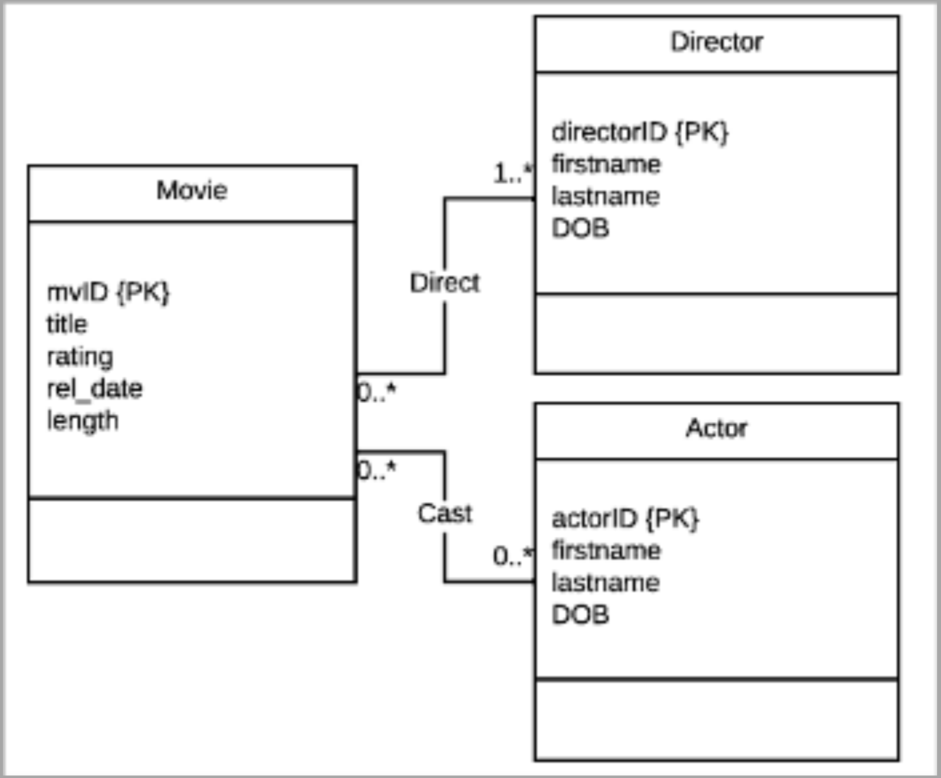
Map an ER diagram into a relational database schema

- Entity sets and their relationships are mapped to relations, under the relational model.
- The number of entity sets and relationships decide the number of relations and their attributes.

Mapping ER Diagrams to Relational Database Schemas: Basics

- Entity sets to Relations:
  - attributes → attributes.
  - PK: PK attributes of classes
- M:N Relationships to Relations:
  - Foreign keys pointing to relevant entity sets.
  - If an association class, add the attributes.
  - PK: foreign key attributes combined

Mapping M:N relationships: example



Movie(mvID, title, rating, rel_date, length)

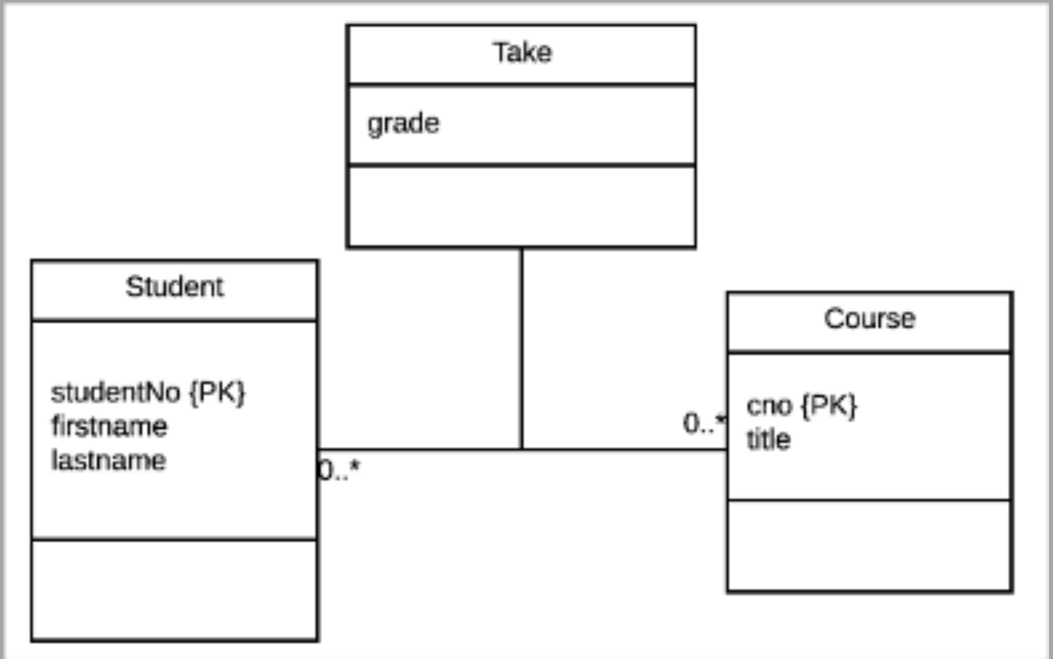
Director(directorID, firstname, lastname, DOB)

Actor(actorID, firstname, lastname, DOB)

Cast(mvID*, actorID*)

Direct(mvid*, directorID*)

Mapping M:N relationships: example



Student(studentNo, firstname, lastname)

Course(cno, title)

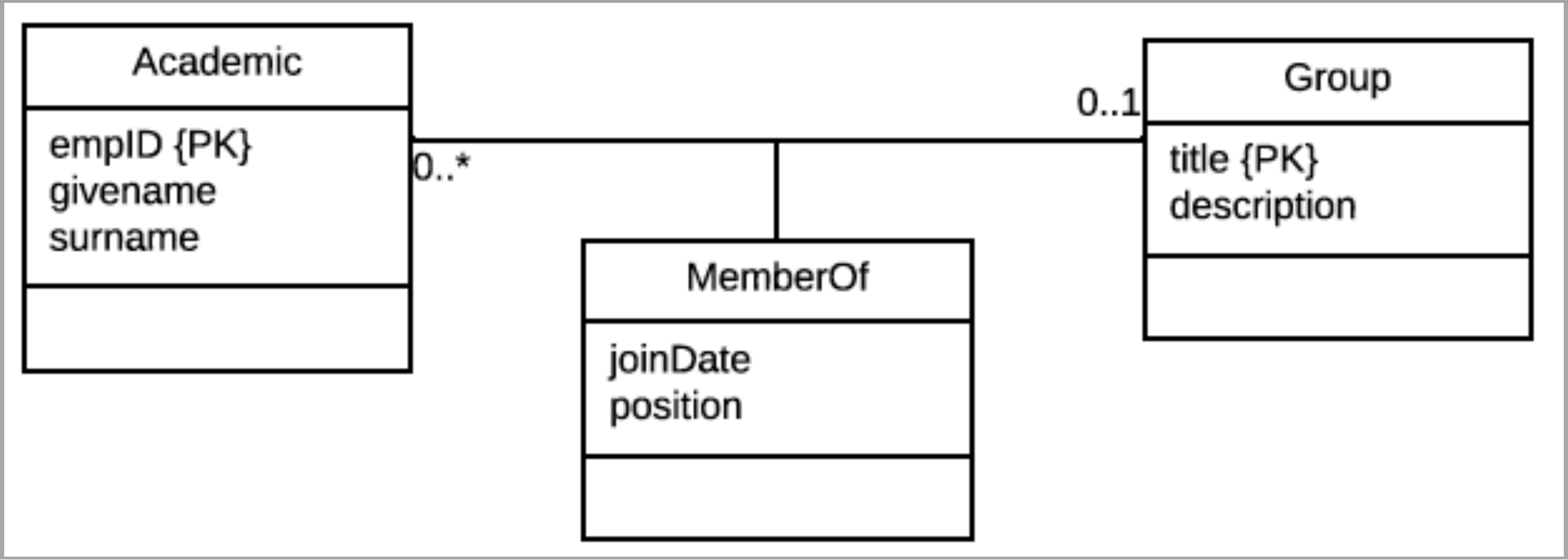
Take(studentNo*, cno*, grade)

Mapping M:1 relationships with attributes

M:1 relationships with attributes are mapped to relations (1:1 is a special M:1 relationship):

- Foreign keys pointing to relevant entity sets.
- Include the relationship attributes.
- PK: foreign key attributes to the entity set on the “many” end

Mapping M:1 relationships with attributes: example



Academic(empID, givename, surname)

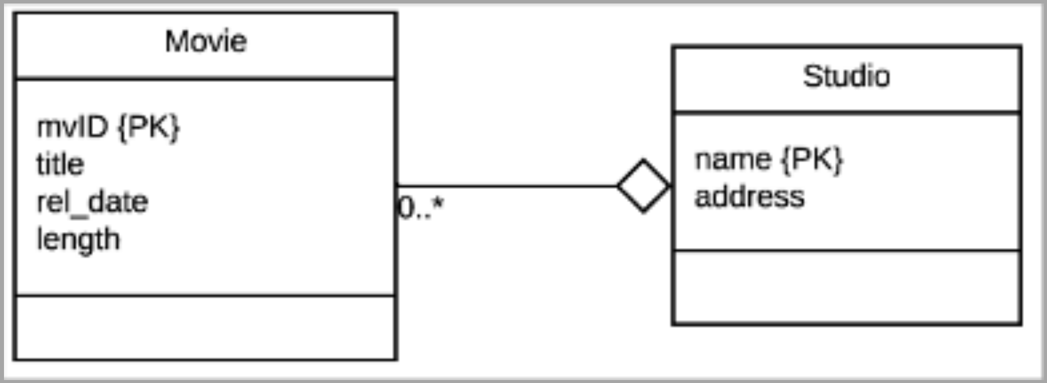
Group(title, description)

MemberOf (empID*, title*, joinDate, position)

Mapping Compositions and Aggregations (M:1 relationships without attributes)

- Compositions or Aggregations are NOT mapped to relations. Instead, add the PK attributes of the diamond-end class as foreign keys to the relation for the class at the other end.
- The added foreign key attributes can be null for aggregations.

Mapping Compositions and Aggregations: Example

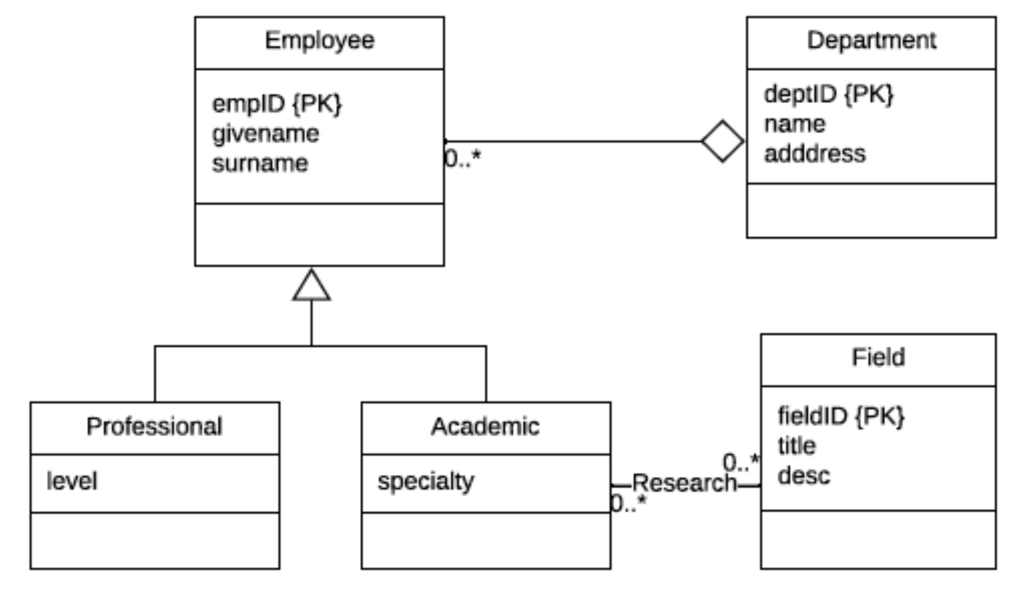


Movie(mvID, title, rel_date, length, studio_name*)
Studio(name, address)

Mapping subclass hierarchies

- The relation for entities of a subclass comprises attributes of its superclass as well as and its own attributes.
- Each subclass is mapped to a relation, including a foreign key referencing PK of the superclass as well as its own attributes.
- PK: PK attributes of the root class

Mapping the subclass hierarchies: an example



Note: the subclass hierarchy is complete.

Employee(empID, givename, surname, deptID*)

Professional(empID*, level)

Academic(empID*, specialty)

Department(deptID, name, address)

Field(fieldID, title, desc)

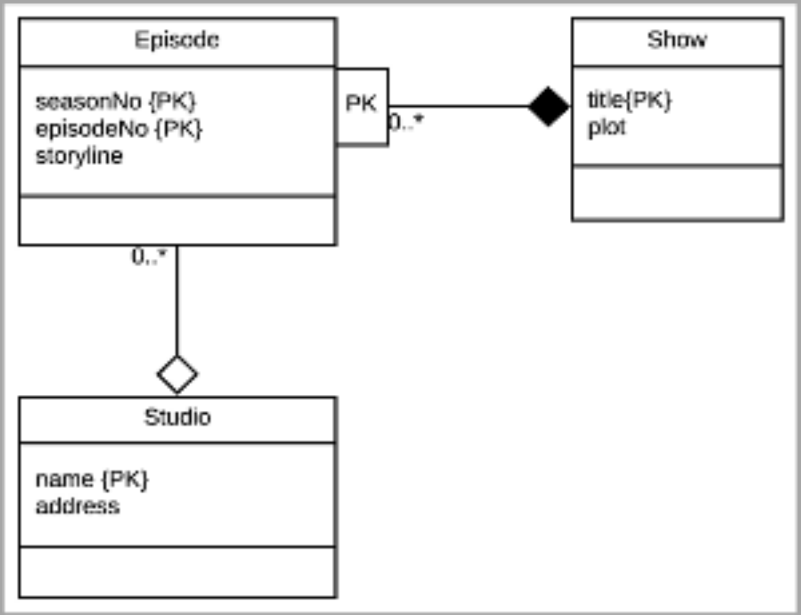
Research(empID*, fieldID*)

Mapping weak entity sets

A weak entity class is mapped to a relation, including

- foreign key pointing to the supporting class entity set(s)
- any attributes of the weak entity set itself
- PK: foreign key attributes to the supporting entity set(s) and any PK attributes of the weak entity set itself

Mapping weak entity sets: example



Show(title, plot)

Studio(name, address)

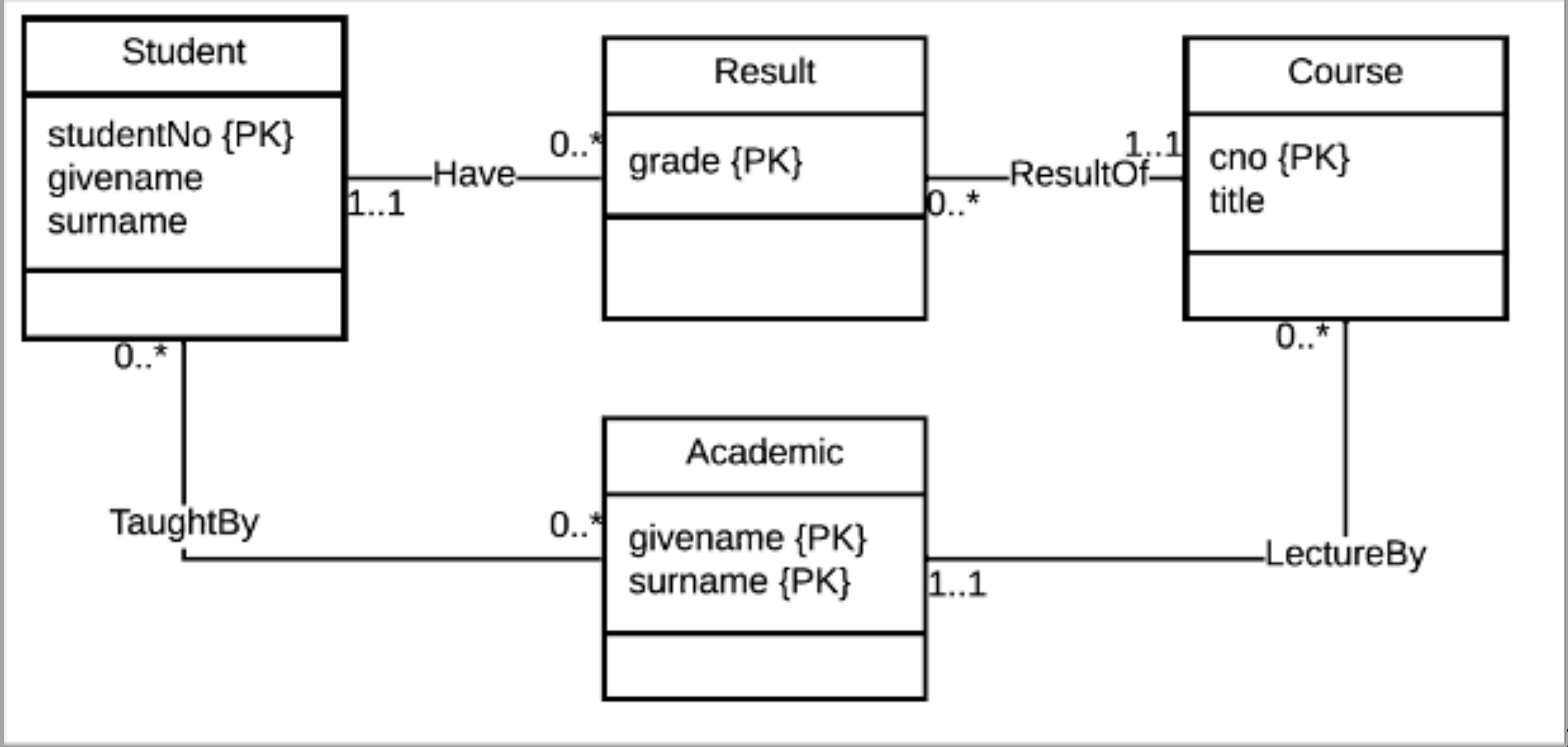
Episode(show-title*, seasonNo, episodeNo, storyline, studio-name*)

ER Diagram Design Principles

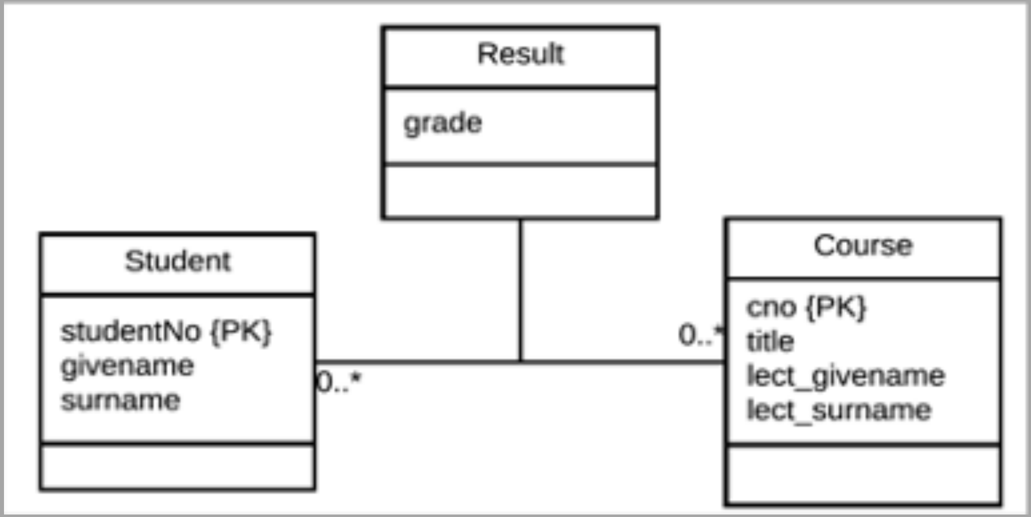
- An ER diagram is a database schema and should only include entities and relationships for which data will be kept.
  - Entities keep data.
  - Relationships also keep data.
- Avoid redundancy.
  - One piece of information is modelled only once.
- Simpler is better.
  - Don’t use an entity set when an attribute will do.

A bad design

The below ER diagram has redundancy and is unnecessarily complex.

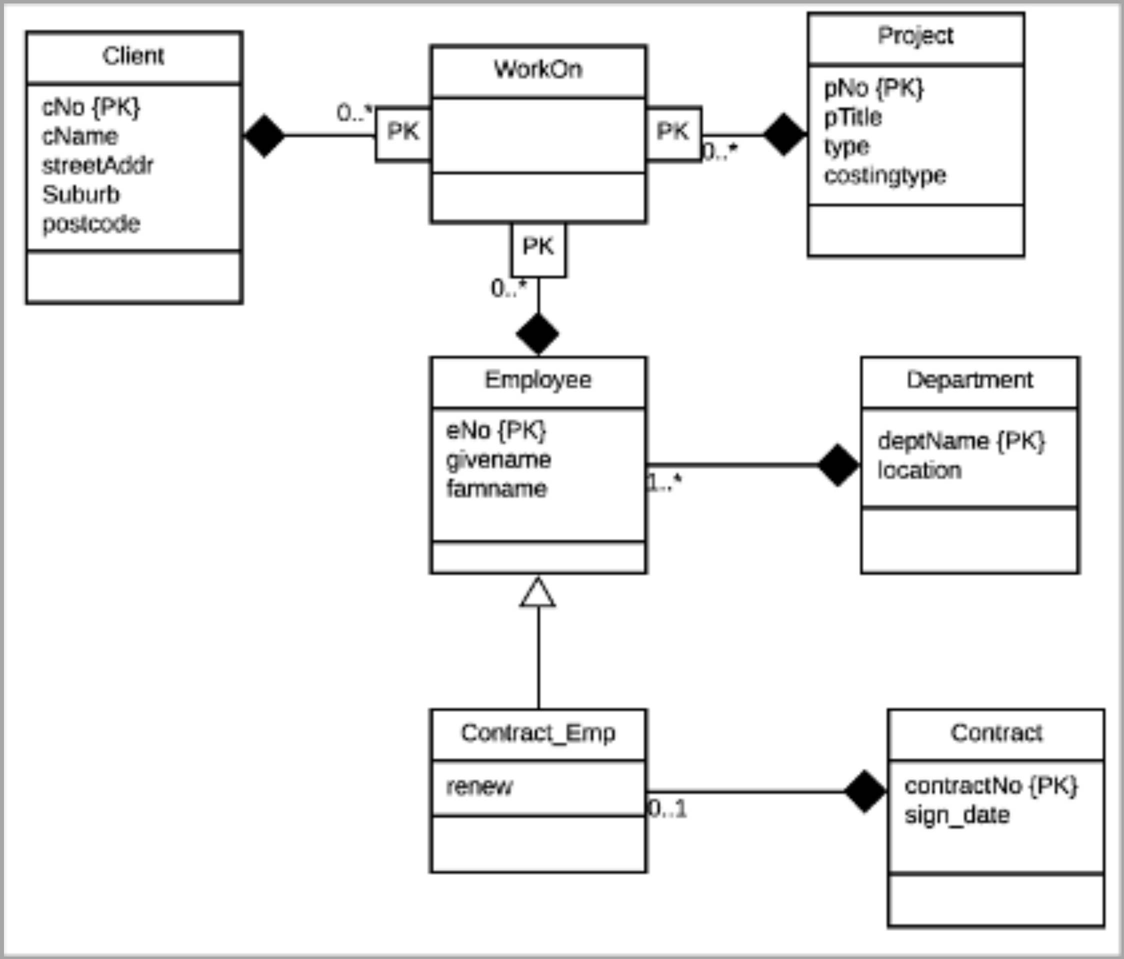


A good design and its mapping



Student(studentNo, givename, surname)
Course(cno, title, lect_givename, lect_surname)
Result(studentNo*, cno*, grade)

Example ERD: The Employee database



- The subclass hierarchy is partial.

Mapping into a relational database schema

Client(cNo, cName, streetAddr, suburb, postcode)

Project(pNo, pTitle, type, costingtype)

Department(deptName, location)

Employee(eNo, givename, famname, deptName*)

Contract_Emp(eNo*, renew, contractNo*)

Contract(contractNo, sign_date)

WorkOn(cNo*, pNo*, eNo*)

Summary

- Principle for ERD design: simpler is better.
- ERD is the first step for designing a database. ERD is mapped to a relational database schema.
- How to examine the ”goodness” of a relational database schema will be discussed in “Relational database design” and “Normalization”.