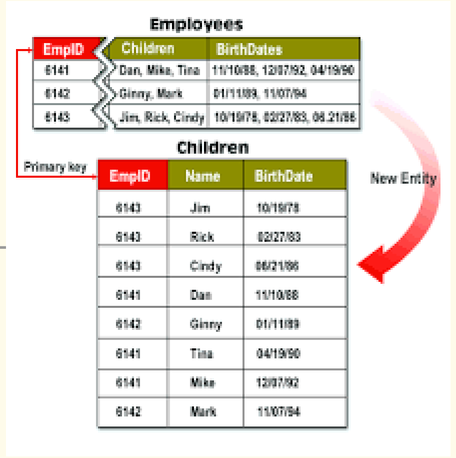
Mapping of Advanced ER Models



Employees

EmpID	Children	BirthDates
6141	Dan, Mike, Tina	11/10/88, 12/07/92, 04/19/90
6142	Ginny, Mark	01/11/89, 11/07/94
6143	Jim, Rick, Cindy	10/19/78, 02/27/83, 06.21/86

Children

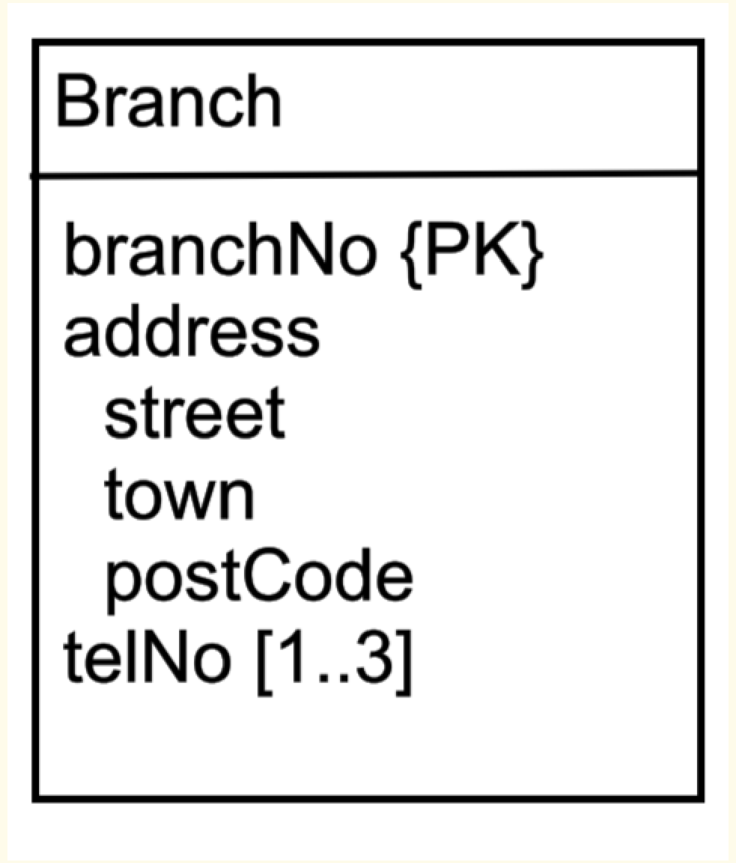
EmpID	Name	BirthDate
6143	Jim	10/19/78
6143	Rick	02/27/83
6143	Cindy	06/21/86
6141	Dan	11/10/88
6142	Ginny	01/11/89
6141	Tina	04/19/90
6141	Mike	12/07/92
6142	Mark	11/07/94

Topics List

- Multivalued Attributes
- Recursive Relationships
- Weak Entity Types

Multivalued Attributes
Modelling

- Recall that a multivalued attribute holds multiple values for each occurrence of an entity type.
- To model a multivalued attribute you write the attribute followed by square brackets [ ] and inside the square brackets you write down the min and max values.



Multivalued Attributes
Mapping

- To map a multivalued attribute, we ***create a new relation to represent the multi-valued attribute and include the primary key of original entity in the new relation, to act as a foreign key.***

- Unless the multi-valued attribute is itself an alternate key of the entity, the primary key of the new relation is the combination of the multi-valued attribute and the primary key of the entity.

Multivalued Attributes

Mapping

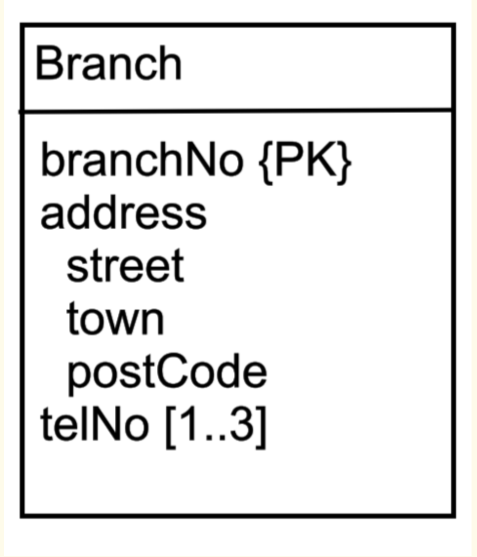
- **Example One:**

We have a Branch entity with attributes: branchNo, address (street, town, postCode), and telNo. The attribute telNo can have many values.

We map branchNo, street, town, and postcode into the first relation and map telNo into the second relation. To relate/link the relations we post branchNo into the second relation.

Branch(branchNo, street, town, postCode)
Primary key branchNo

BranchPhones(telNo, branchNo)
Primary key telNo
Foreign key branchNo references Branch(branchNo)



Note: Since telNo is unique to each Branch, telNo is sufficient as the Primary key of the new relation.

Multivalued Attributes

Mapping

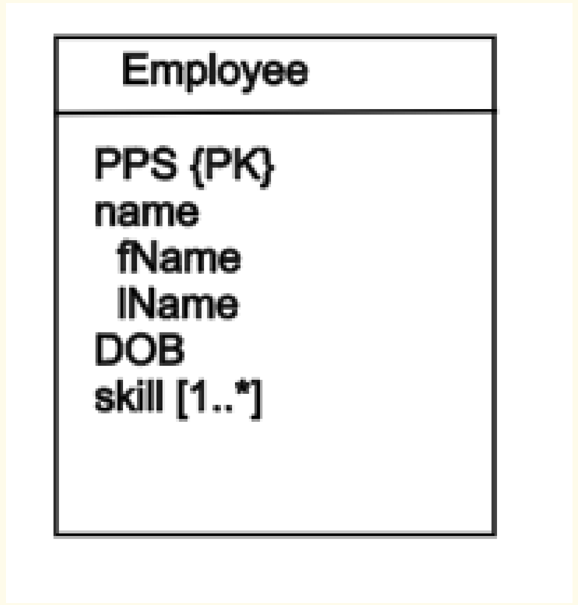
- **Example Two:**

  Employee(PPS, fName, lName, DOB)

  Primary key PPS

  EmpSkill(PPS, skill)
  Primary key PPS, skill
  Foreign key PPS references Employee(PPS)

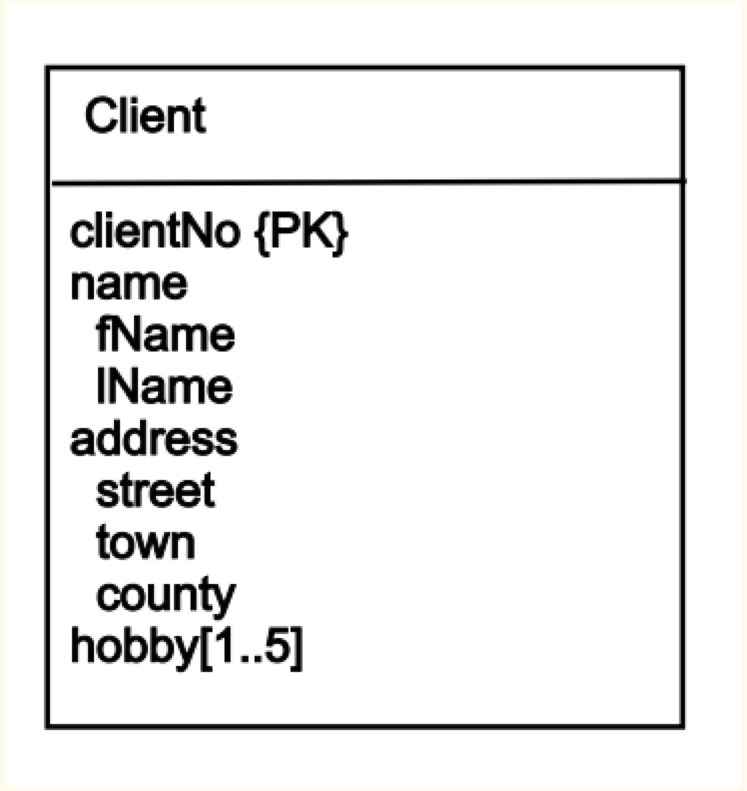
  Since skill is not unique to any person (employee), a composite primary key (PPS, skill) is required.



Multivalued Attributes

Mapping

- **Exercise**
  - Using the figure specified below, create a logical data model for the entity type *Client*:

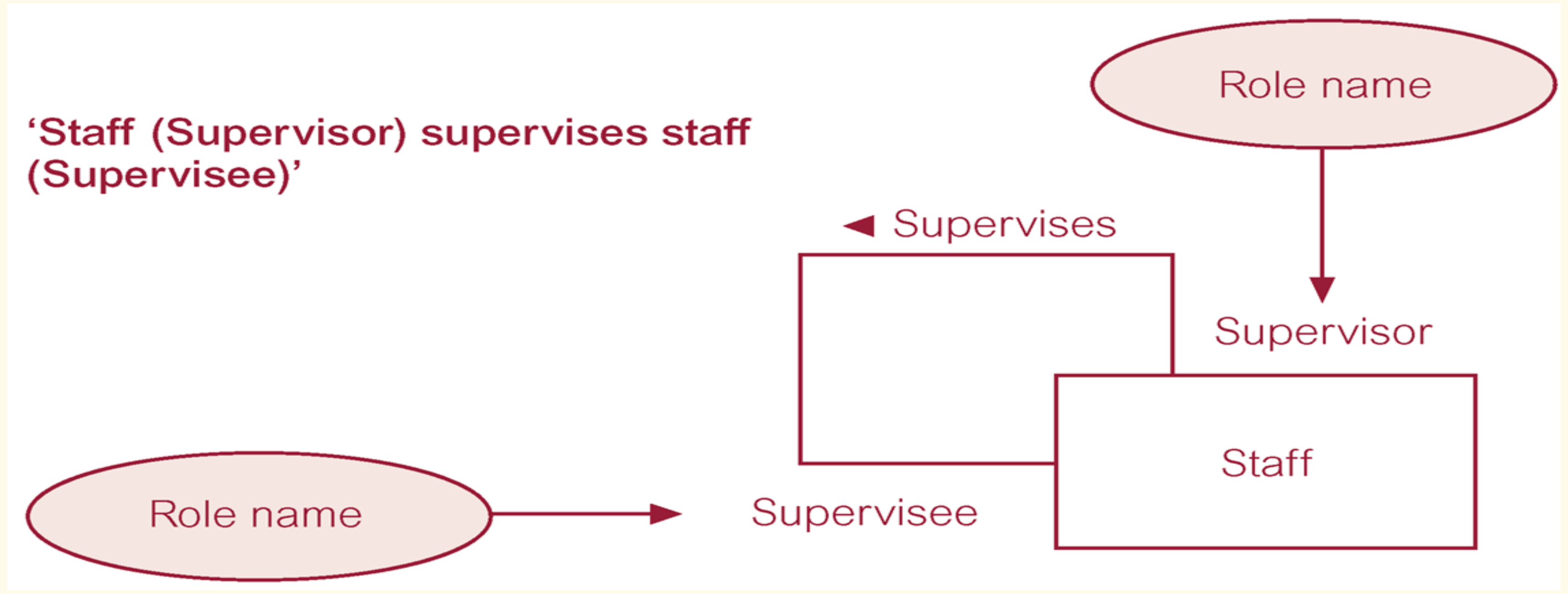


Topics List

- Multivalued Attributes
- Recursive Relationships
- Weak Entity Types

Recursive Relationships

- Recall that a ***Recursive Relationship*** is a relationship type where the *same* entity type participates more than once in *different* roles. Sometimes called *unary* relationships.



Recursive Relationships

- We need to map:
  - ***1:* recursive relationships***
  - ***1:1 recursive relationships***
  - ****:* recursive relationships***

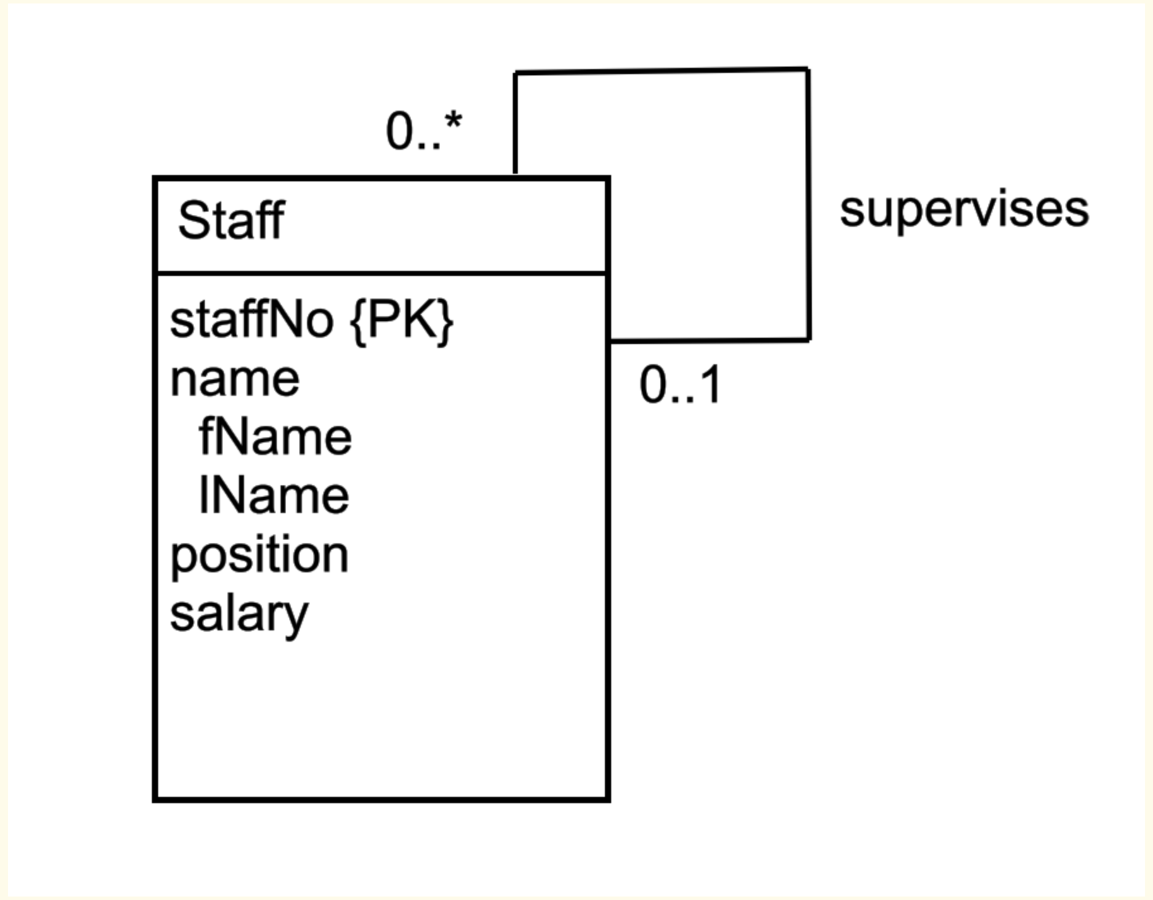
Recursive Relationships

Mapping

- ***1:* recursive relationships***
  - The representation of a 1:* recursive relationship is similar to 1:* binary relationship. However, in this case, both the parent and child entity is the *same* entity.
  - ***For a 1:* recursive relationship, post a copy of the primary key into the same entity (itself) to act as a foreign key. This new attribute is renamed to represent the relationship.***

Recursive Relationships
Mapping

- ***1:* recursive relationships***



Staff(staffNo, fName, lName, position, salary, supervisor)
Primary key staffNo
Foreign key supervisor references Staff(staffNo)

Recursive Relationships

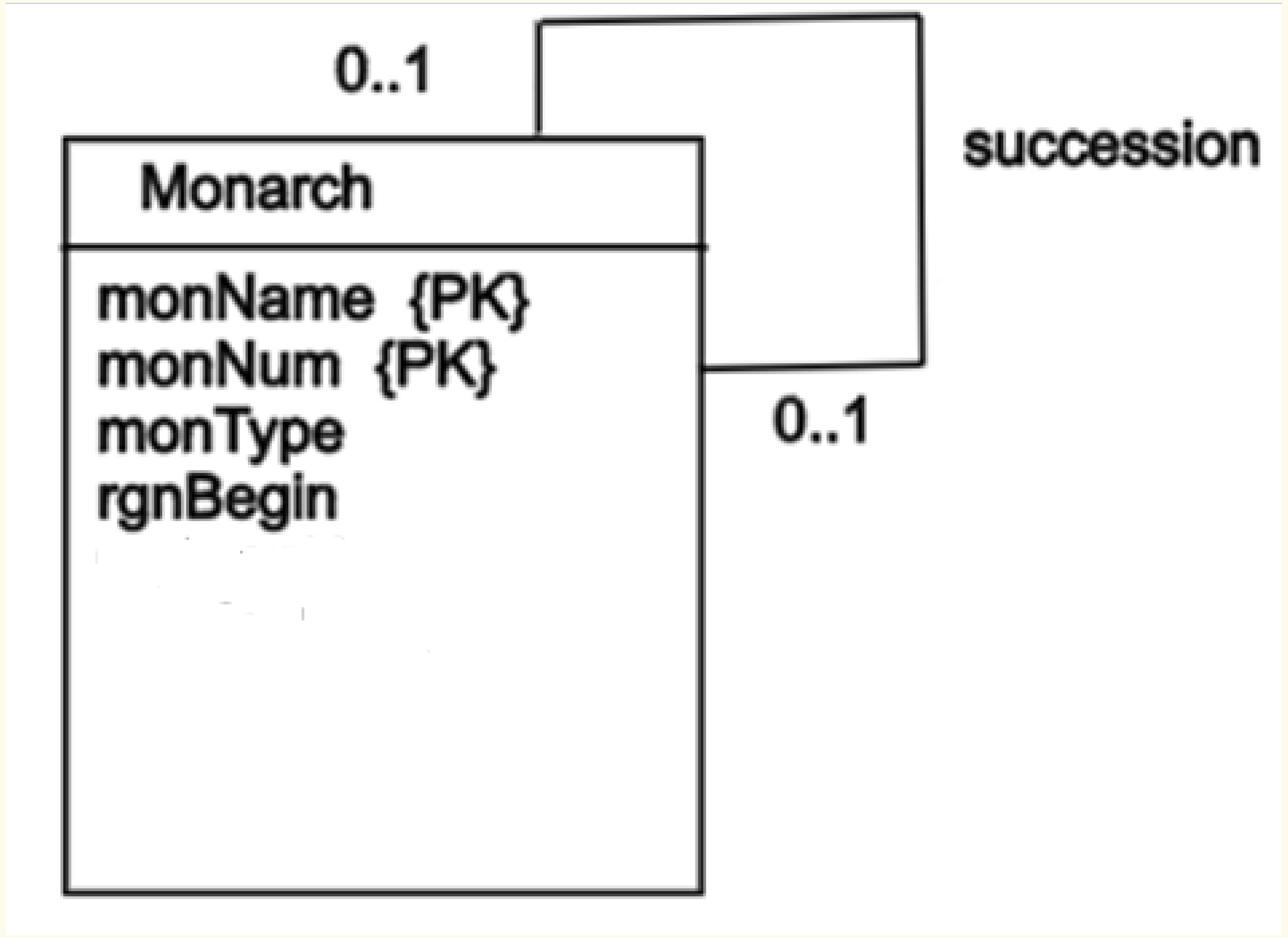
Mapping

- ***1:1 recursive relationships***
  - ***For a 1:1 recursive relationship, post a copy of the primary key into the same entity (itself) to act as a foreign key. This new attribute is renamed to represent the relationship.***

Recursive Relationships

Mapping

- ***1:1 recursive relationships***



Recursive Relationships

Mapping

- ***1:1 recursive relationships***

  monarch (monName, monNum, monType, rgnBegin,
  preMonName, preMonNum)
  Primary key monName, monNum,
  Foreign key preMonName, preMonNum references
  monarch(monName, monNum)

Recursive Relationships

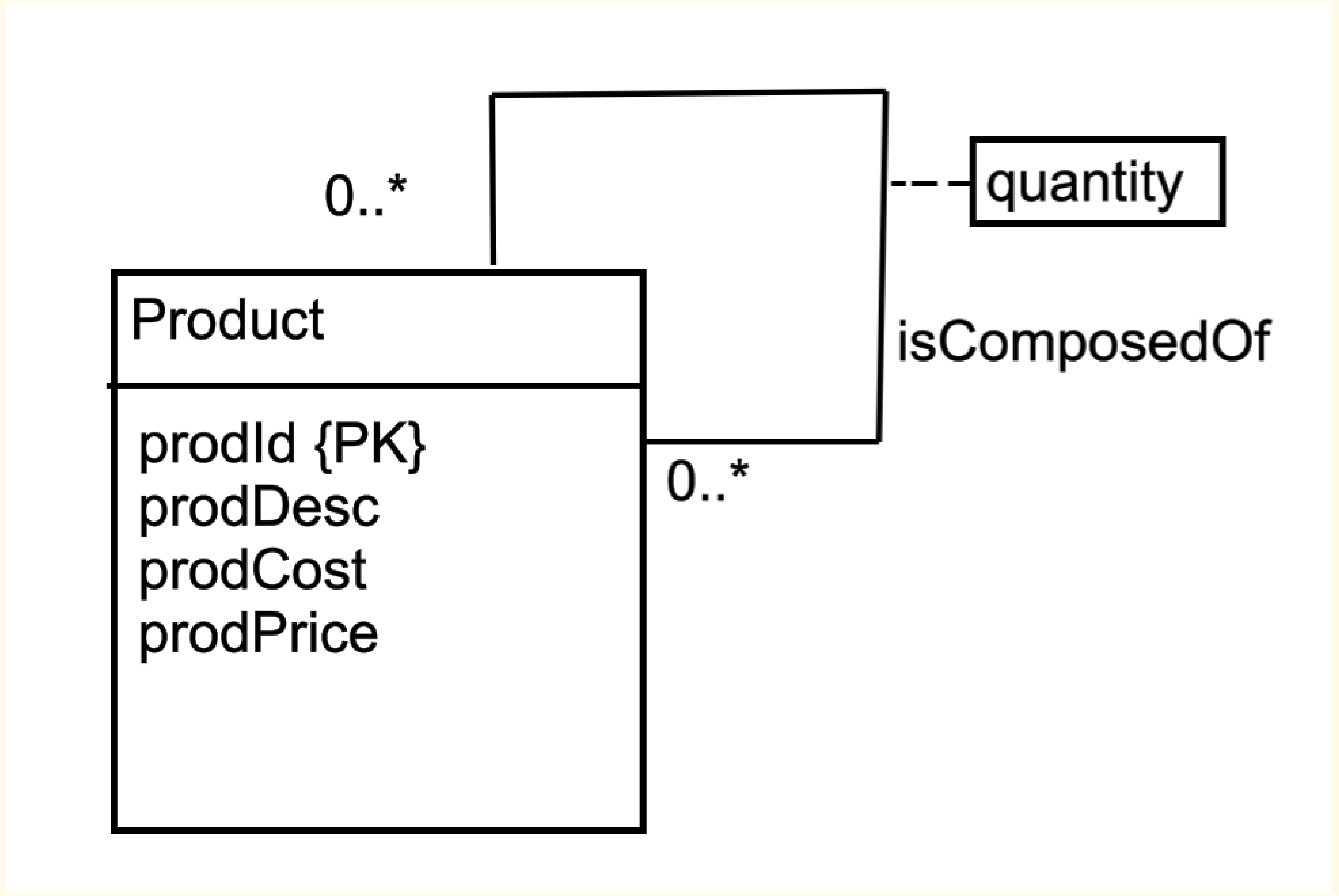
Mapping

- ***:* recursive relationships**
  - The representation of a *:* recursive relationship is similar to *:* binary relationship.
  - ***For a *:* recursive relationship, we will create a new relation which will hold two copies of the original primary key. Again one of the copies of the primary key will be renamed to represent the relationship.***

Recursive Relationships

Mapping

- ***:* recursive relationships***



Recursive Relationships
Mapping

- ***:* recursive relationships***

  product(prodId, prodDesc, prodCost, prodPrice)
  Primary key prodId

  assembly (prodId, subProdId, quantity)
  Primary key prodid, subprodid,
  Foreign key prodid references product(prodid),
  Foreign key subprodid references product(prodid)

Recursive Relationships

Mapping

- ***:* recursive relationships***

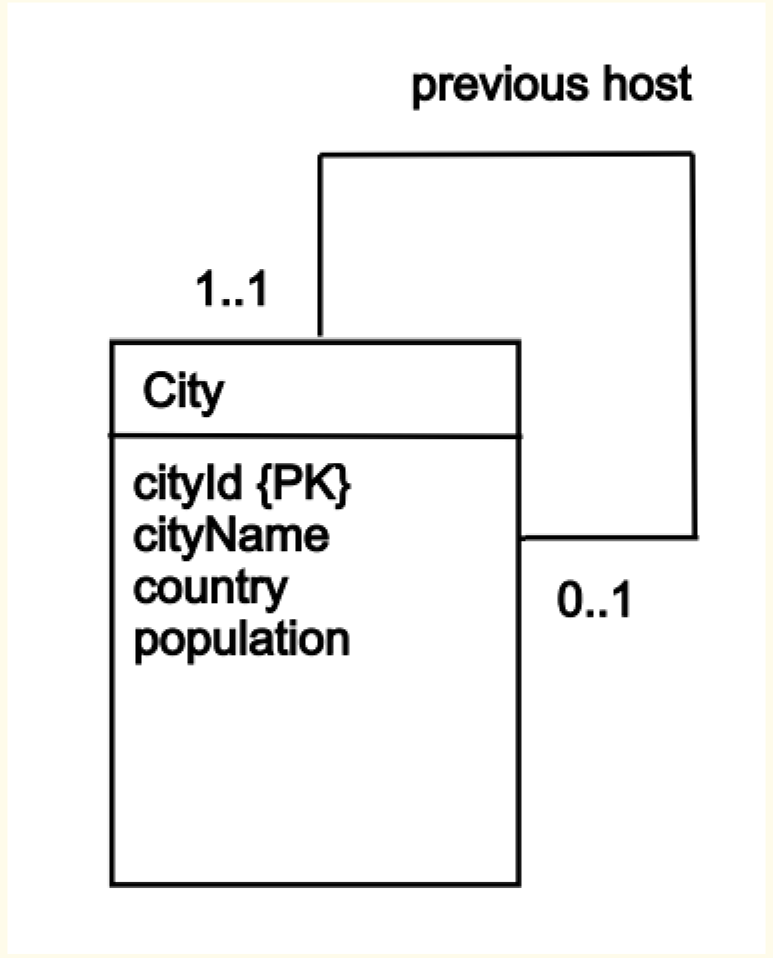
prodId	prodDesc	prodCost	prodPrice
1000	Animal photography kit		725
101	Camera	150	300
102	Camera case	10	15
103	70-210 zoom lens	125	200
104	28-85 zoom lens	115	185
105	Photographer's vest	25	40
106	Lens cleaning cloth	1	1.25
107	Tripod	35	45
108	16 GB SDHC memory card	30	30

prodId	subProdId	quantity
1000	101	1
1000	102	1
1000	103	1
1000	104	1
1000	105	1
1000	106	2
1000	107	1
1000	108	4

Recursive Relationships

Mapping

- **Exercise**
  - Using the figure specified below, create a logical data model for the entity type *City*:



Topics List

- Multivalued Attributes
- Recursive Relationships
- Weak Entity Types

Weak Entity Types

- Recall, a ***Weak Entity Type*** is an entity type that will depend on another entity type for its existence.

- Each entity occurrence cannot be uniquely identified using only the attributes associated with that entity type. A weak entity type does not exist on its own but must participate in a relationship with another (strong) entity type.

Weak Entity Types

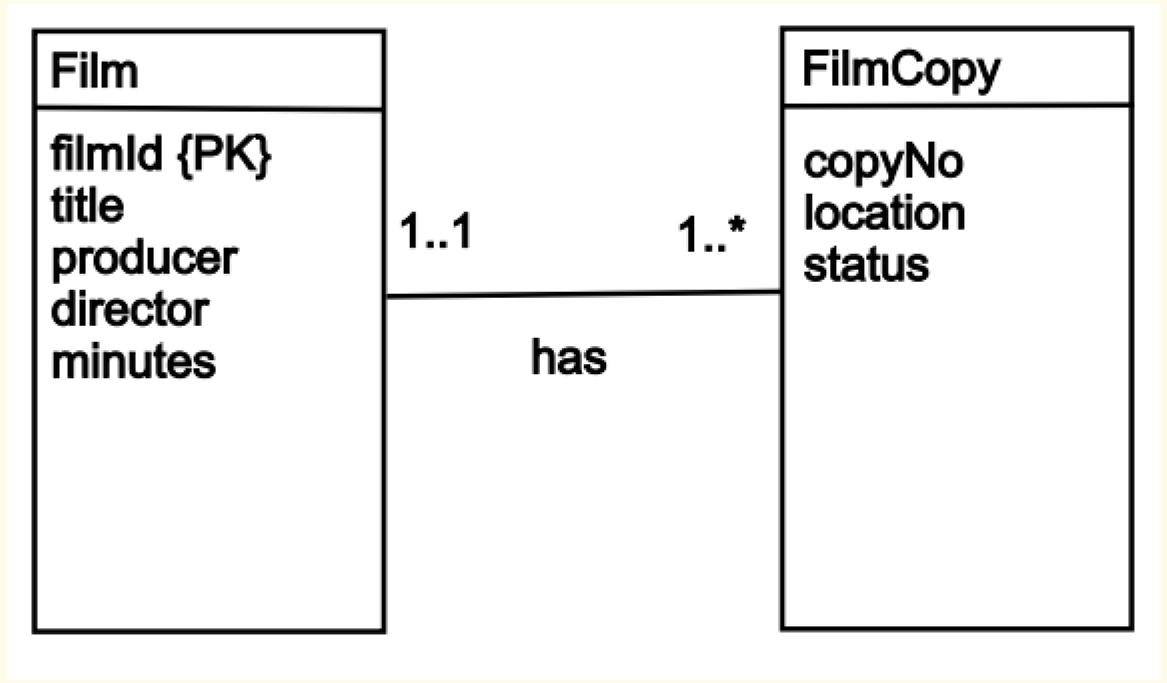
Mapping

- For each weak entity in the data model:
  - ***Create a relation that includes all the simple attributes of that entity.***
  - ***The primary key of a weak entity is partially or fully derived from each owner entity and so the identification of the primary key of a weak entity cannot be made until after all the relationships with the owner entities have been mapped.***

Weak Entity Types

Mapping

- For example 1, we will create a relation each for *Film* and *FilmCopy*.
- Since the relationship between *Film* and *FilmCopy* is one-to-many, we post a copy of the primary key from the Film entity type (filmId) as a foreign key into *FilmCopy*.
- This attribute now becomes part of the primary key of the *FilmCopy* relation (along with one or more other attributes as there may be many film copies).



Weak Entity Types

Mapping

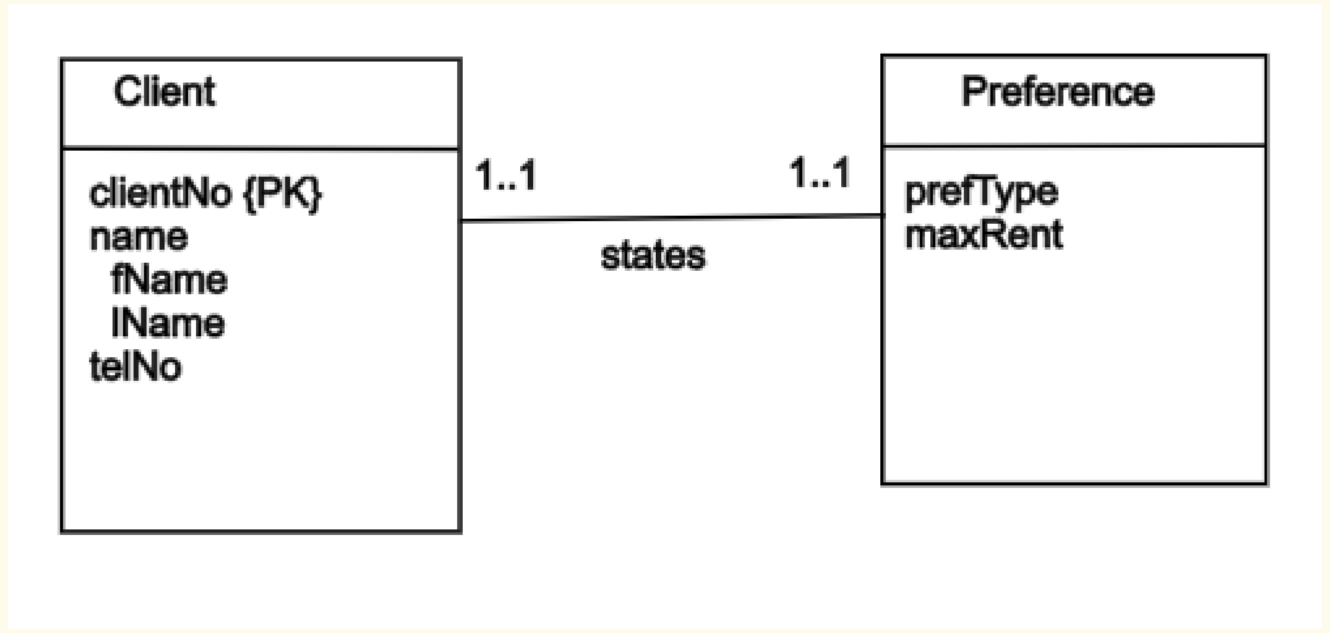
Film(filmId, title, producer, director, minutes)
Primary key filmId

FilmCopy(filmId, copyNo, location, status)
Primary key filmId, copyNo
Foreign key filmId references Film(filmId)

Weak Entity Types

Mapping

- For example 2, we will create a relation each for *Client* and *Preference*.
- Since the relationship between *Client* and *Preference* is one-to-one and fully mandatory, we post a copy of the primary key from one side to the other. Since there is only one primary key value, we post (clientNo) as a foreign key into Preference.



Weak Entity Types

Mapping

Client(clientNo, fName, lName, telNo)
Primary key clientNo

Preference(clientNo, prefType, maxRent)
Primary key clientNo
Foreign key clientNo references Client(clientNo)

Note: As the relationship is one-to-one, a Client will only have one preference, therefore *clientNo* is sufficient as the Primary key for the *Preference* relation.

Weak Entity Types
Mapping

OR

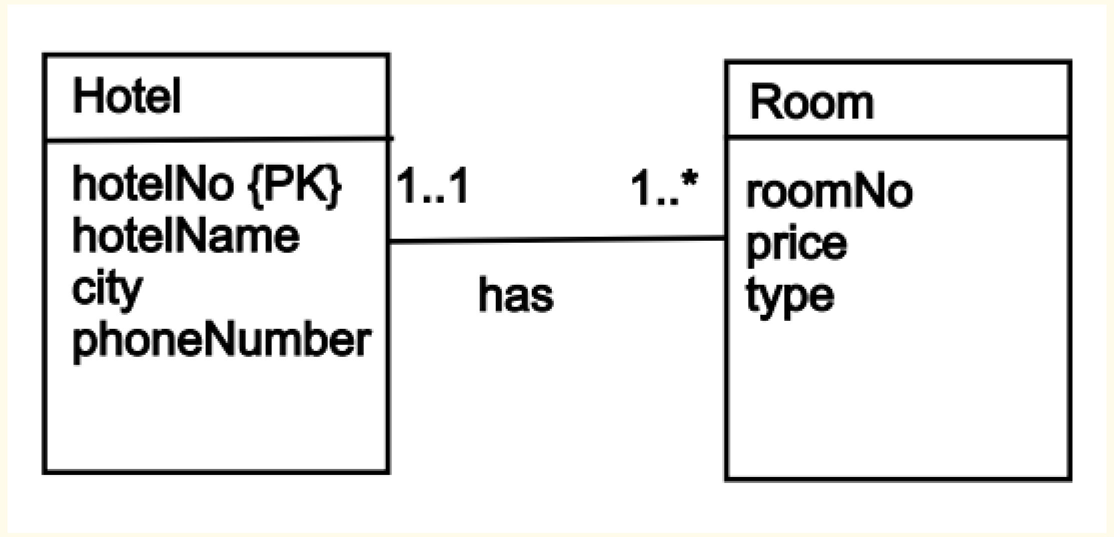
- Since the relationship is one:one (fully mandatory) we could merge the above 2 relations into one as follows:

  Client(clientNo, fName, lName, telNo, prefType, maxRent)
  Primary key clientNo

Weak Entity Types

Mapping

- **Exercise**
  - Using the figure specified below, create a logical data model for the conceptual data model:


Hotel
hotelNo {PK}
hotelName
city
phoneNumber
1..1
has
1..*
Room
roomNo
price
type