Logical Level | Relational Data Model

Last Week ×Q4U

Weak Entity? Partial Key?

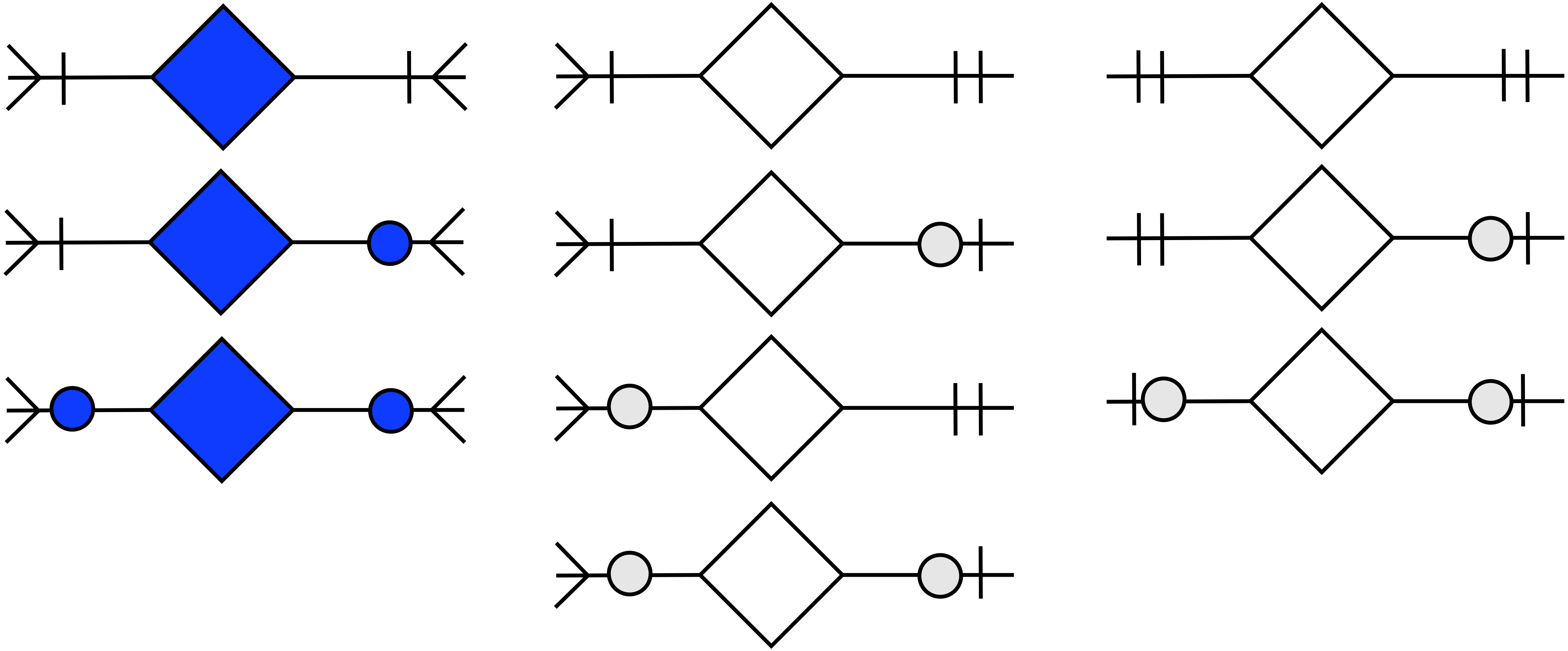
What is foreign key set?

Is foreign key set a key set?

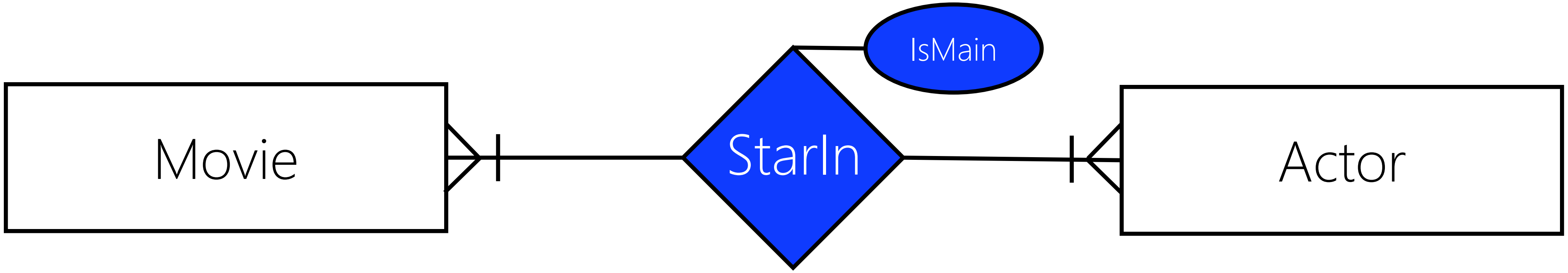
Is it ok to have foreign key set as attribute in ERD?

ISA, total vs. partial, overlapping vs. disjoint?

Relationship2Relation (R2R)

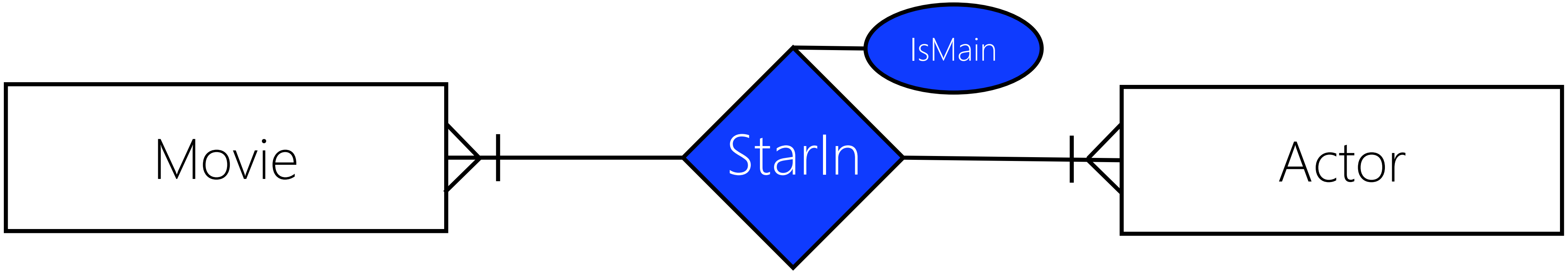


R2R × Many-Many



m_1	m_1, a_1, 0	a_1
m_2	m_1, a_2, 1	a_2
	m_1, a_3, 0	a_3
	m_2, a_1, 1	
	m_2, a_3, 0	

R2R × Many-Many



m_1

m_2

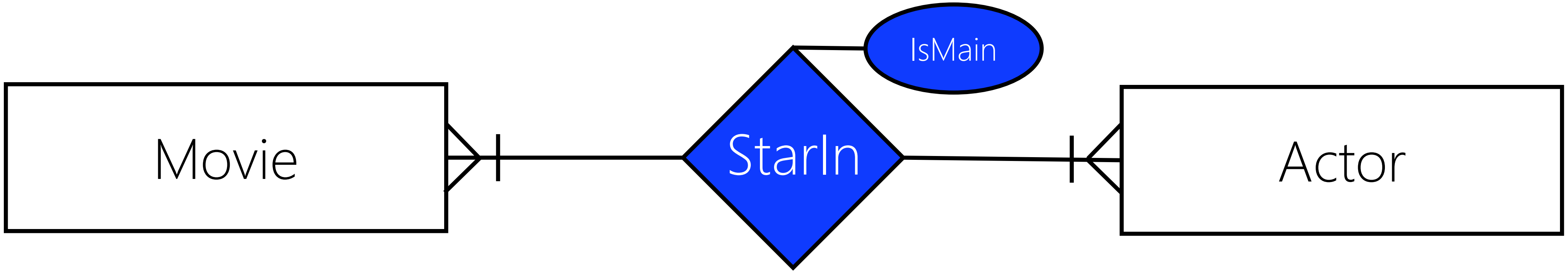
a_1, m_1, 0, m_2, 1

a_2, m_1, 1

a_3, m_1, 0, m_2, 0

Maximum number of movies is not known!

R2R × Many-Many



$m_1, a_3, 0, a_2, 1, a_1, 0$

$m_2, a_3, 0, a_1, 1$

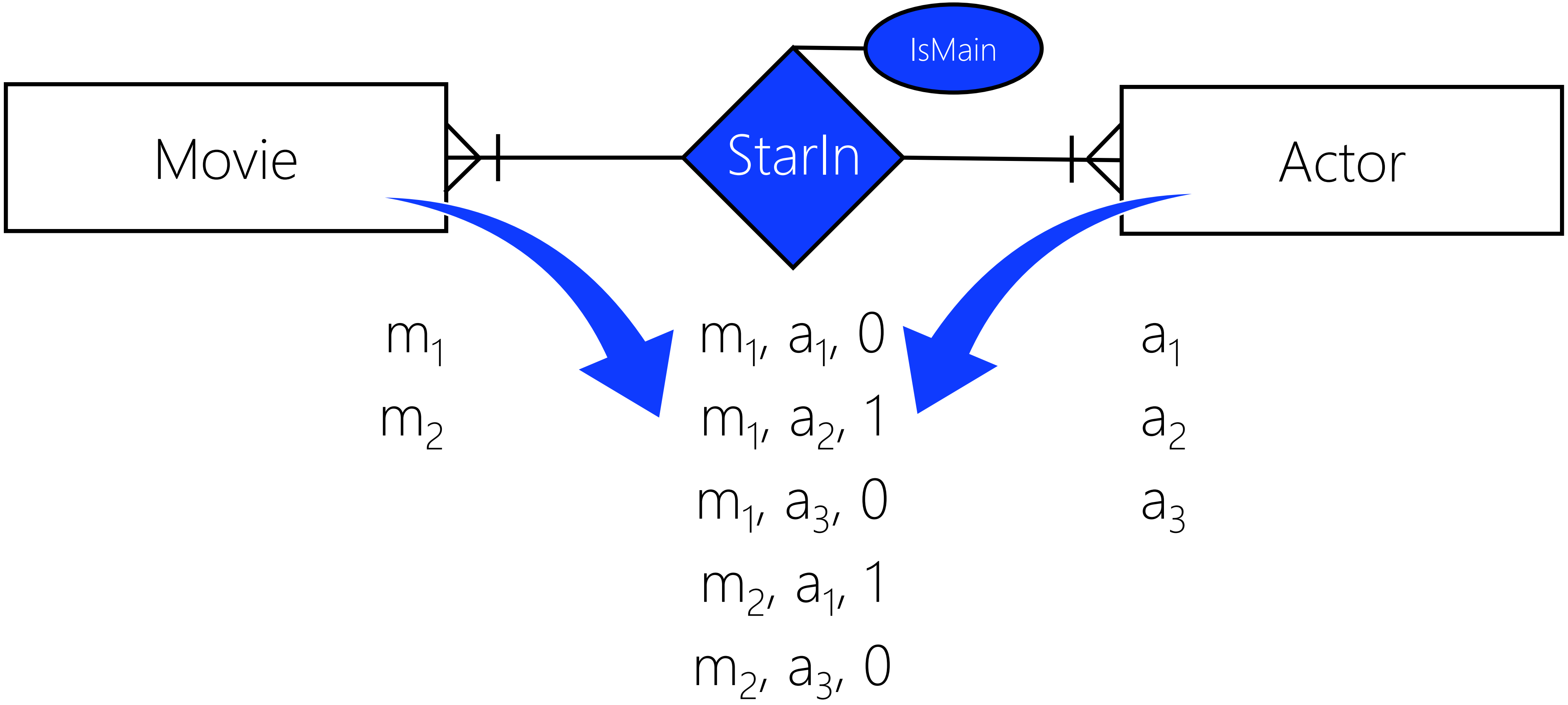
a_1

a_2

a_3

Maximum number of actors is not known!

R2R × Many-Many



m_1

m_2

$m_1, a_1, 0$

$m_1, a_2, 1$

$m_1, a_3, 0$

$m_2, a_1, 1$

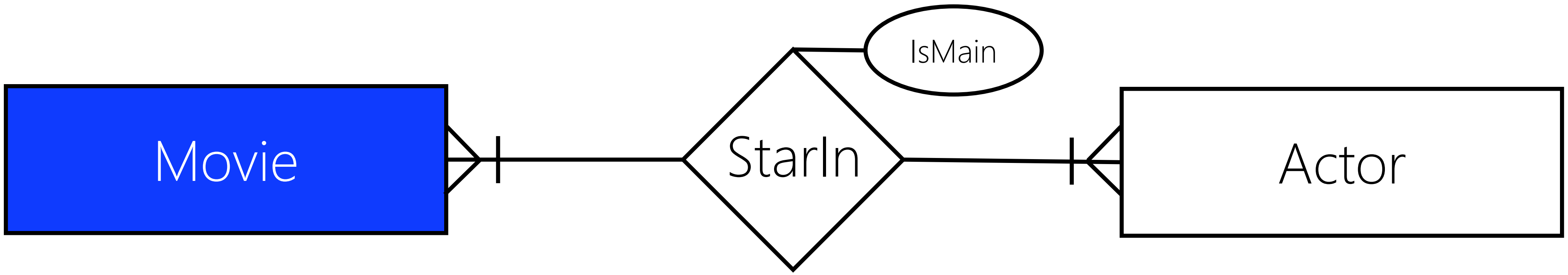
$m_2, a_3, 0$

a_1

a_2

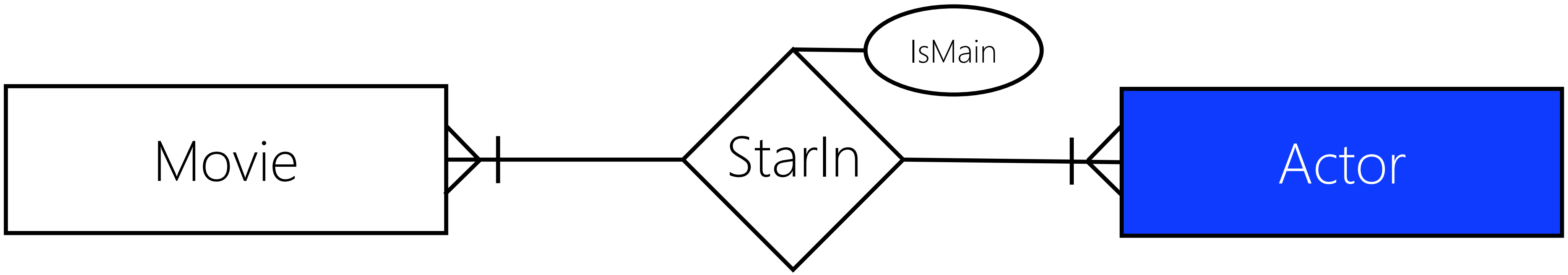
a_3

R2R × Many-Many



R_1: Movie(<u>Title</u>, <u>ReleaseYear</u>, <u>ReleaseMonth</u>, <u>ReleaseDay</u>, RunningTime, Age)

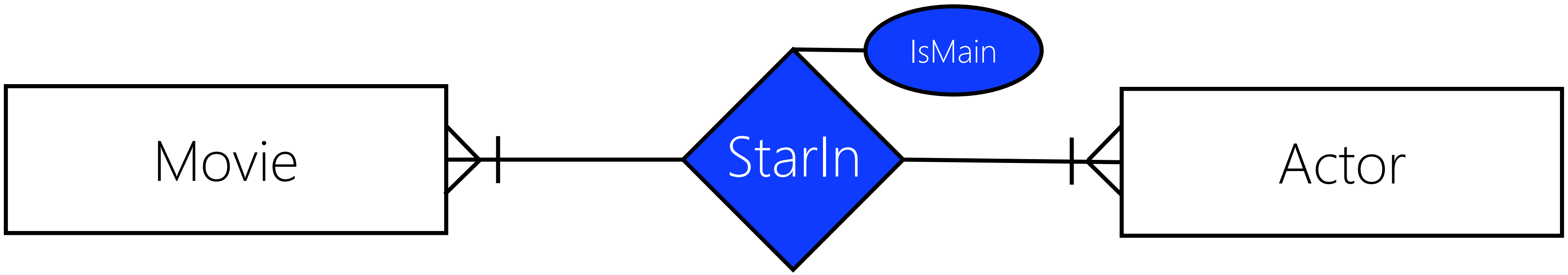
R2R × Many-Many



R_1: Movie(<u>Title</u>, <u>ReleaseYear</u>, <u>ReleaseMonth</u>, <u>ReleaseDay</u>, RunningTime, Age)

R_2: Actor(<u>FirstName</u>, <u>LastName</u>, <u>DateOfBirth</u>, PlaceOfBirth, Age)

R2R × Many-Many

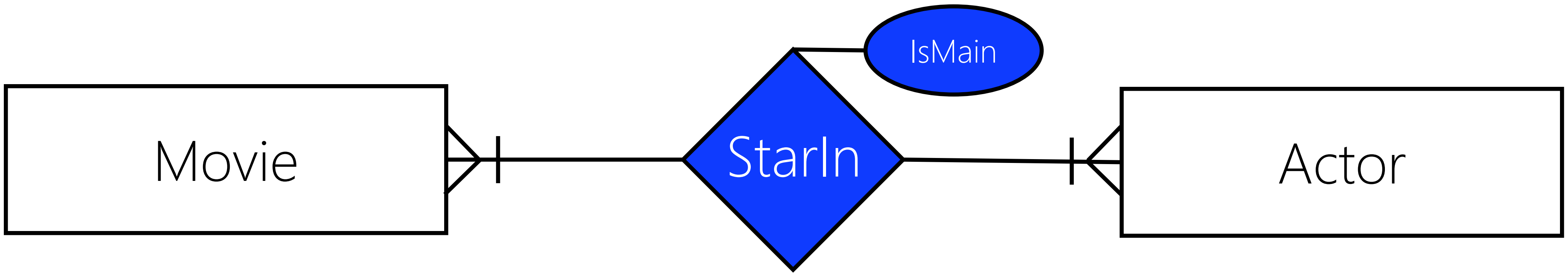


R_1: Movie(<u>Title</u>, <u>ReleaseYear</u>, <u>ReleaseMonth</u>, <u>ReleaseDay</u>, RunningTime, Age)

R_2: Actor(<u>FirstName</u>, <u>LastName</u>, <u>DateOfBirth</u>, PlaceOfBirth, Age)

R_3: StarIn(Movie.Title, Movie.ReleaseYear, Movie.ReleaseMonth, Movie.ReleaseDay,
Actor.FirstName, Actor.LastName, Actor.DateOfBirth,
IsMain)

R2R × Many-Many

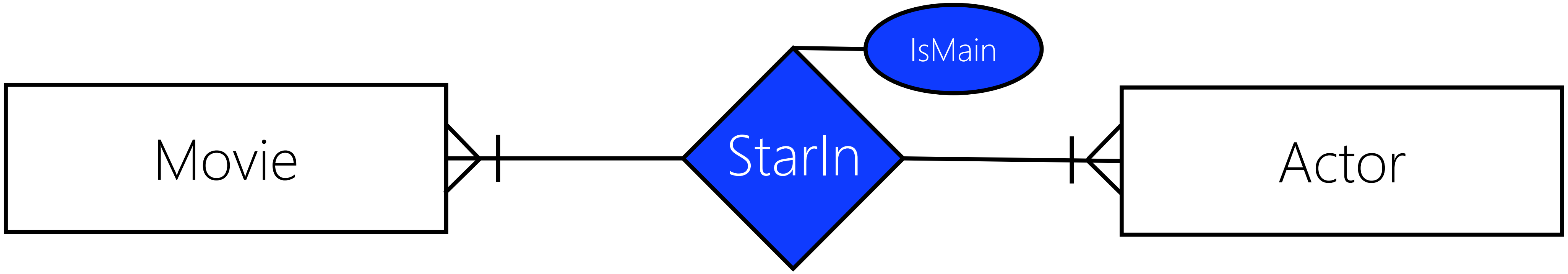


R_1: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age)

R_2: Actor(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age)

R_3: StarIn($\underbrace{\text{Movie.Title, Movie.ReleaseYear, Movie.ReleaseMonth, Movie.ReleaseDay,}}_{FK_1}$
$\underbrace{\text{Actor.FirstName, Actor.LastName, Actor.DateOfBirth,}}_{FK_2}$
IsMain)

R2R × Many-Many



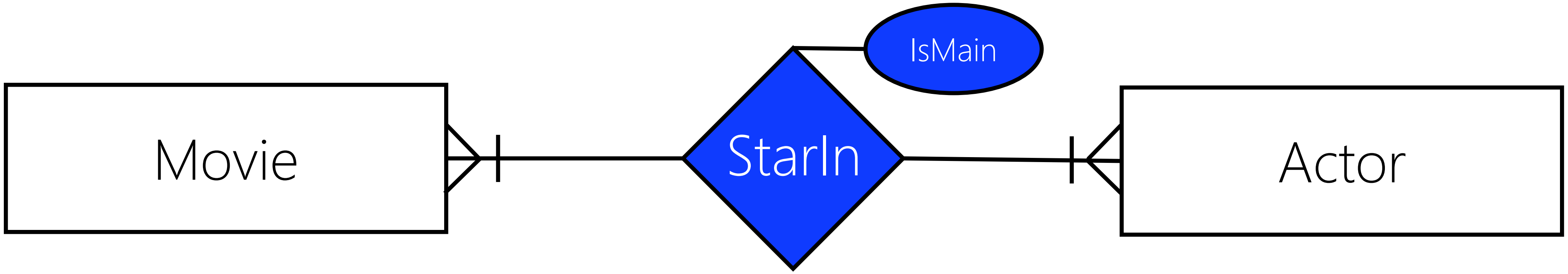
R_1: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age)

R_2: Actor(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age)

R_3: StarIn(Movie.Title, Movie.ReleaseYear, Movie.ReleaseMonth, Movie.ReleaseDay,
Actor.FirstName, Actor.LastName, Actor.DateOfBirth,
IsMain)

PK??

R2R × Many-Many



R_1: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age)

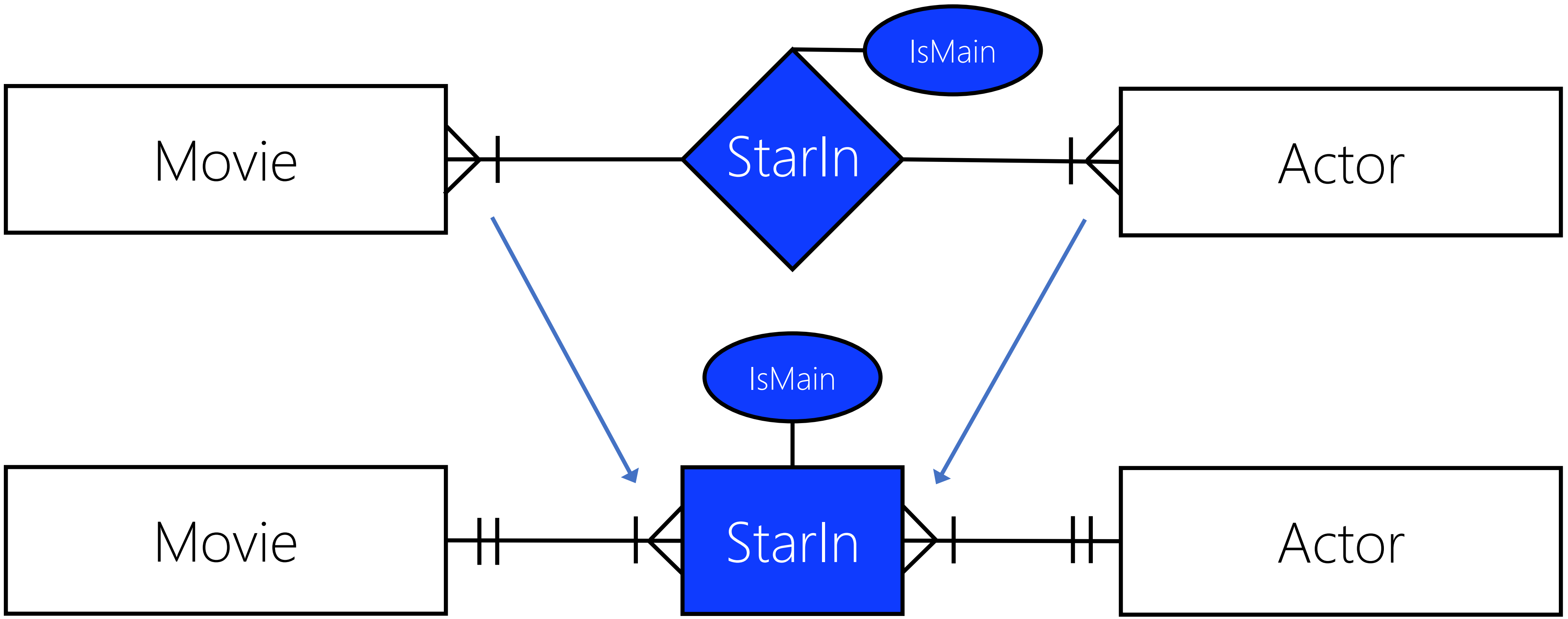
R_2: Actor(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age)

R_3: StarIn(Movie.Title, Movie.ReleaseYear, Movie.ReleaseMonth, Movie.ReleaseDay,
Actor.FirstName, Actor.LastName, Actor.DateOfBirth,
IsMain)

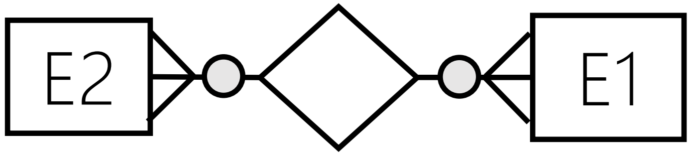
$PK = FK_1 \, U \, FK_2$

Ignore the next slide!

R2R × Many-Many



R2R × Many-Many

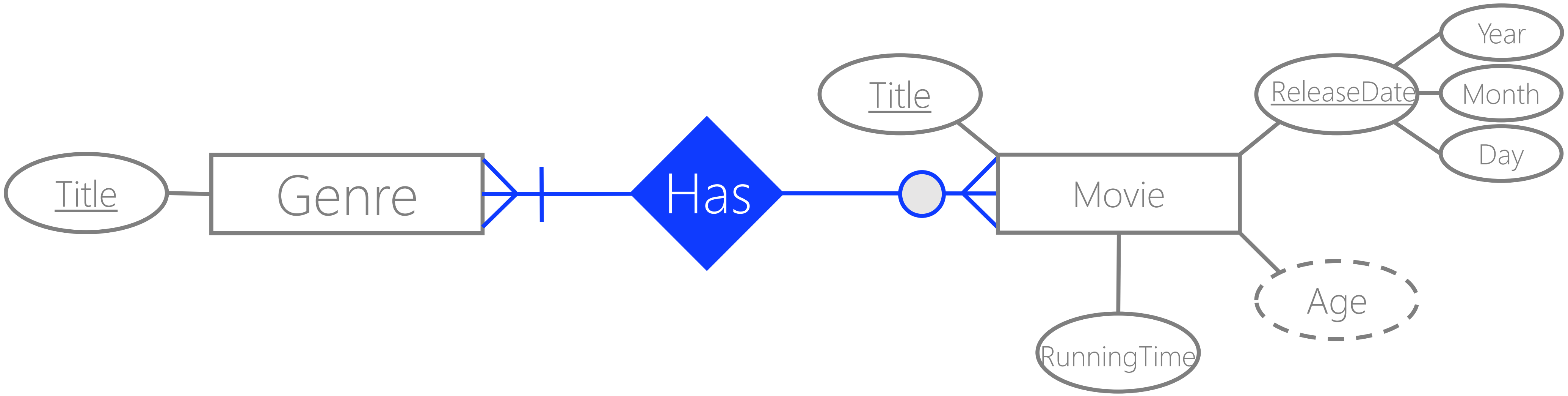


Input: Many-Many relationship btw. E2 and E1, i.e.,
Output: Relations R1 for E1, R2 for E2, and R3 for relationship.

1) For E1, create relation R1 with the same attributes and keys as in E1
2) For E2, create relation R2 with the same attributes and keys as in E2
3) For many-many relationship set, create new relation R3
4) [Foreign Key Set FK_1] Add key set of E1 to R3
5) [Foreign Key Set FK_2] Add key set of E2 to R3
6) Add attributes of relationship set to R3
7) [Primary Key Set] Create key set for R3 from foreign key sets (step 4 & 5)
   $PK = FK_1 \cup FK_2$

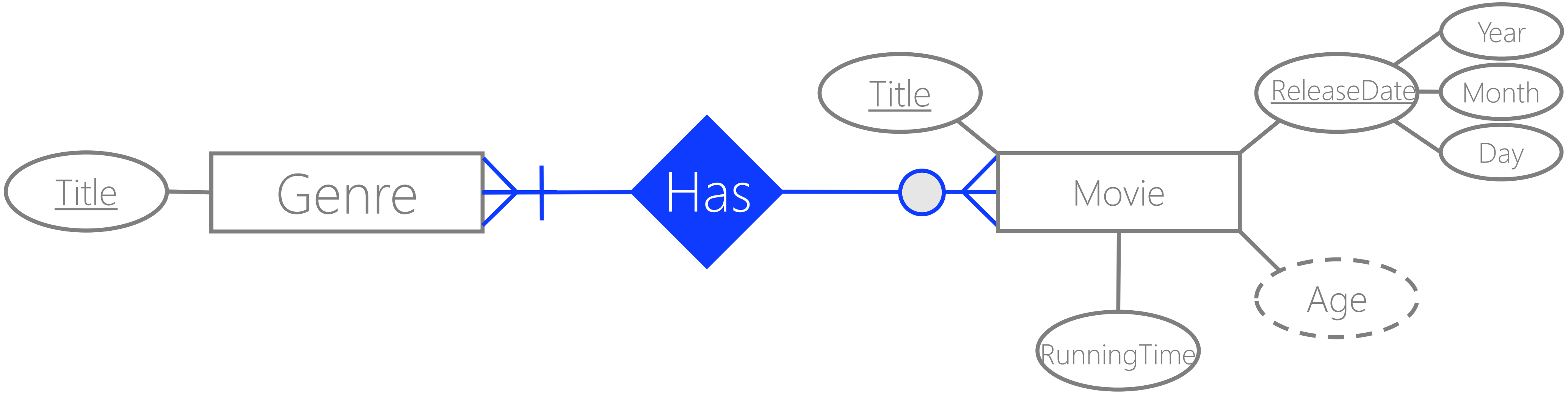
Herein, we do not care about ordinality! We fix it later.

E2R × Entity Set × Multivalued Attribute



R_1: Movie(<u>Title</u>, RunningTime, <u>ReleaseYear</u>, <u>ReleaseMonth</u>, <u>ReleaseDay</u>, Age)
R_2: Genre(<u>Title</u>)

E2R × Entity Set × Multivalued Attribute

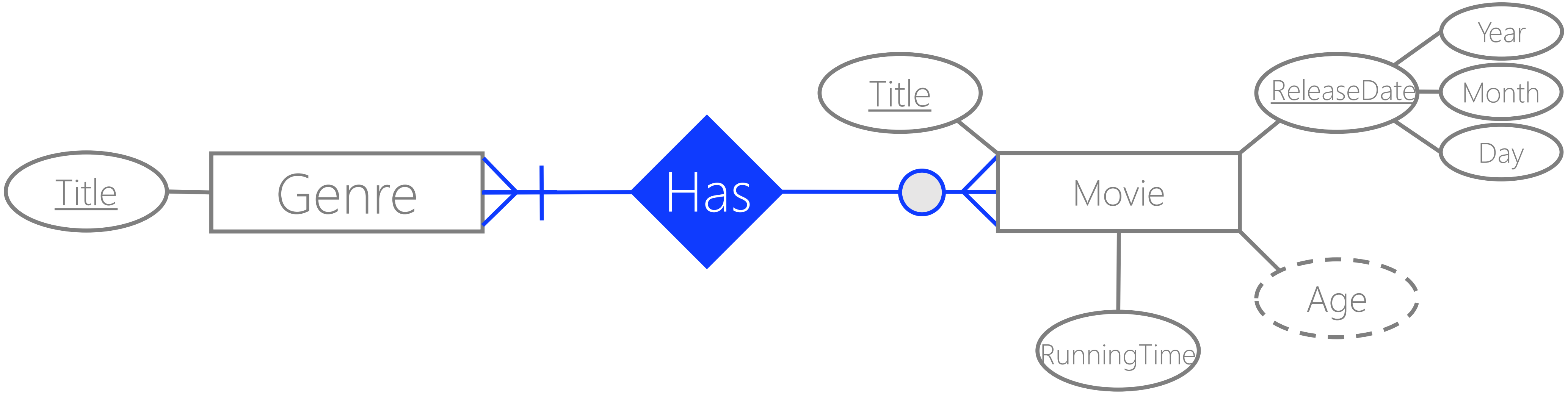


R_1: Movie(Title, RunningTime, ReleaseYear, ReleaseMonth, ReleaseDay, Age)

R_2: Genre(Title)

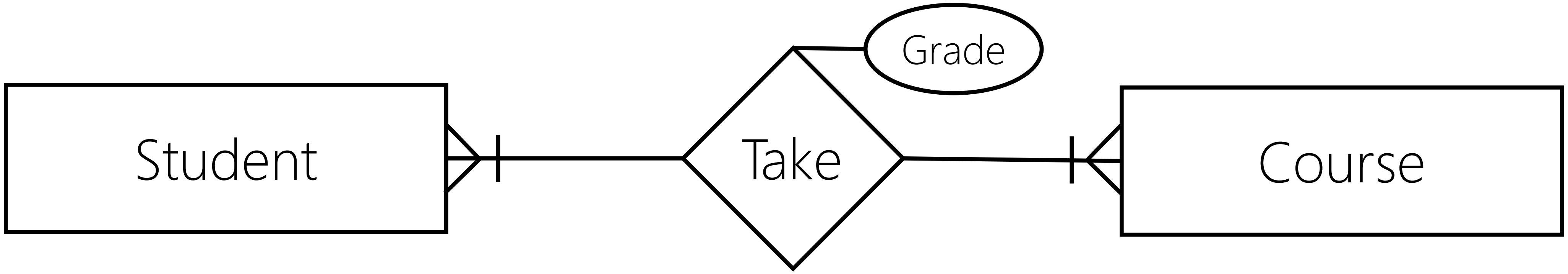
R_3: MovieGenre() #better naming than 'Has'

E2R × Entity Set × Multivalued Attribute

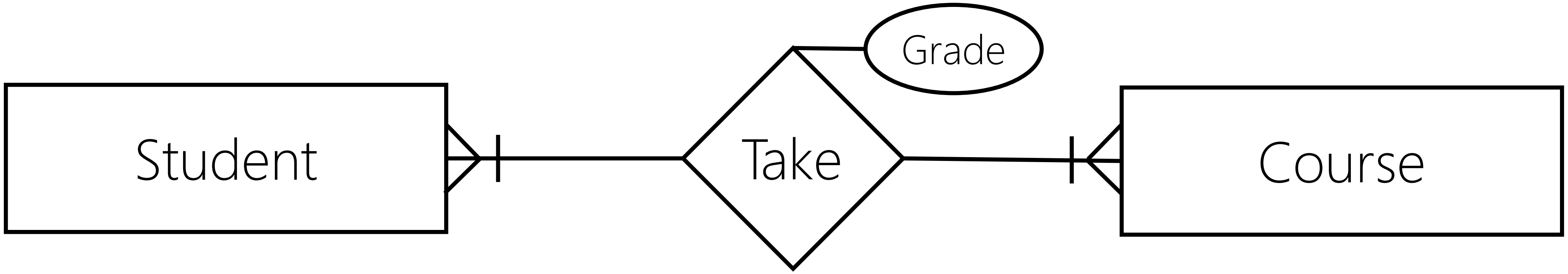


R_1: Movie(Title, RunningTime, ReleaseYear, ReleaseMonth, ReleaseDay, Age)
R_2: Genre(Title)
R_3: MovieGenre(Genre.Title, Movie.ReleaseYear, Movie.ReleaseMonth, Movie.ReleaseDay, Movie.Title)

R2R × Many-Many

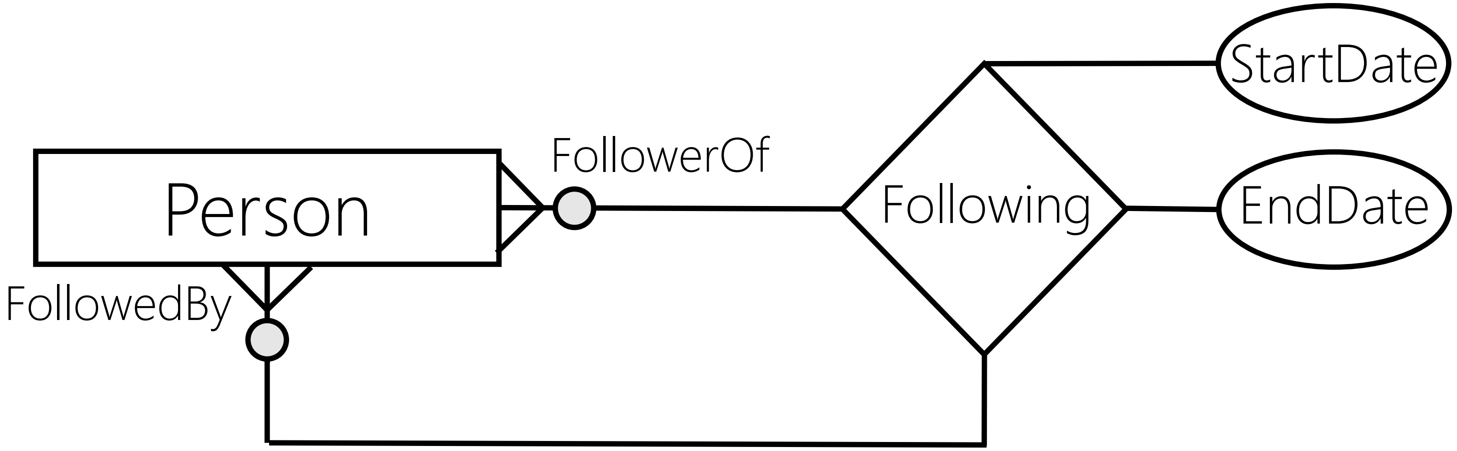


R2R × Many-Many

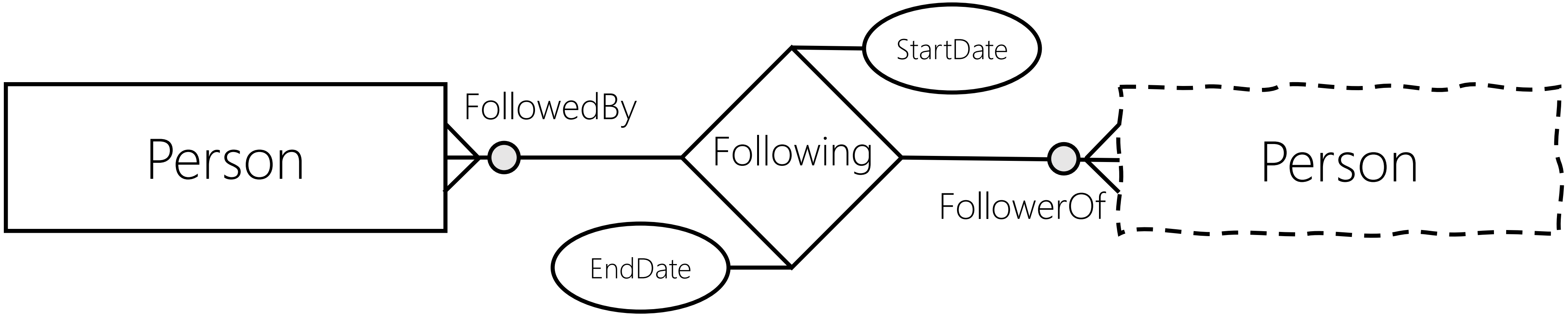


R_1: Student(<u>No</u>, FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_2: Course(Title, <u>Code</u>, Credit, IsRequired, …)
R_3: StudentCourse(<u>Student.No</u>, <u>Course.Code</u>, Grade)

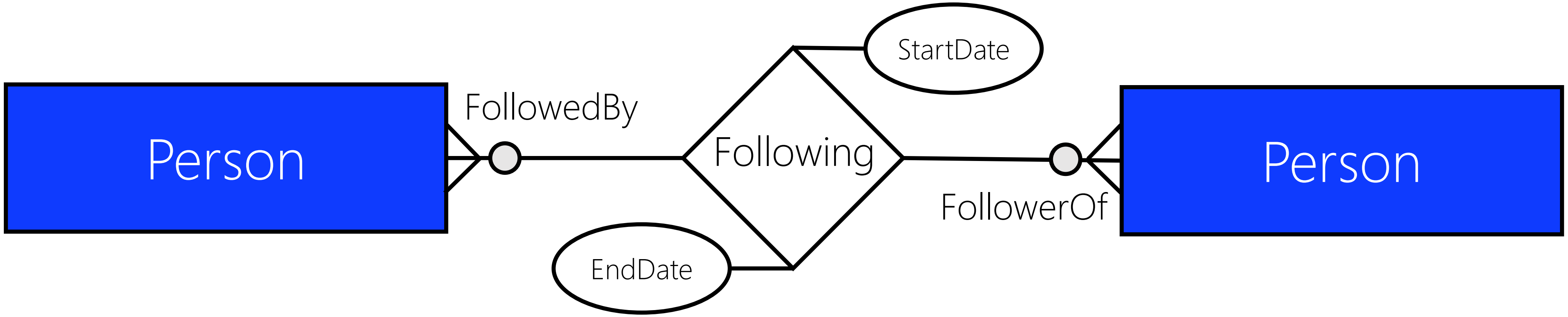
R2R × Many-Many × Self (Unary)



R2R × Many-Many × Self (Unary)



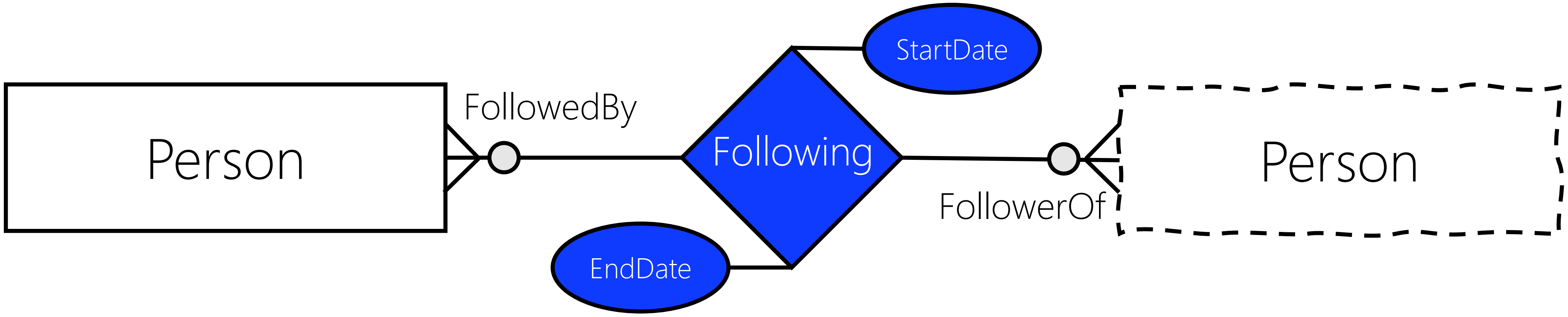
R2R × Many-Many × Self (Unary)



R_1: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, ...)

R_2: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, ...)

R2R × Many-Many × Self (Unary)



R_1: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)

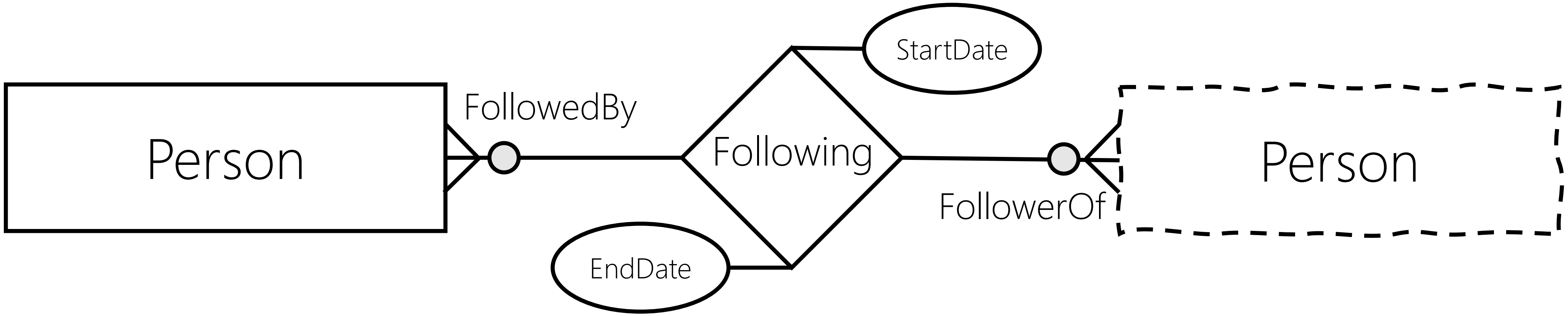
R_2: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)

R_3: Followership(FirstName, LastName, DateOfBirth, PlaceOfBirth, FK1

R_3: Followership FirstName, LastName, DateOfBirth, PlaceOfBirth, FK2

R_3: Followership StartDate, EndDate)

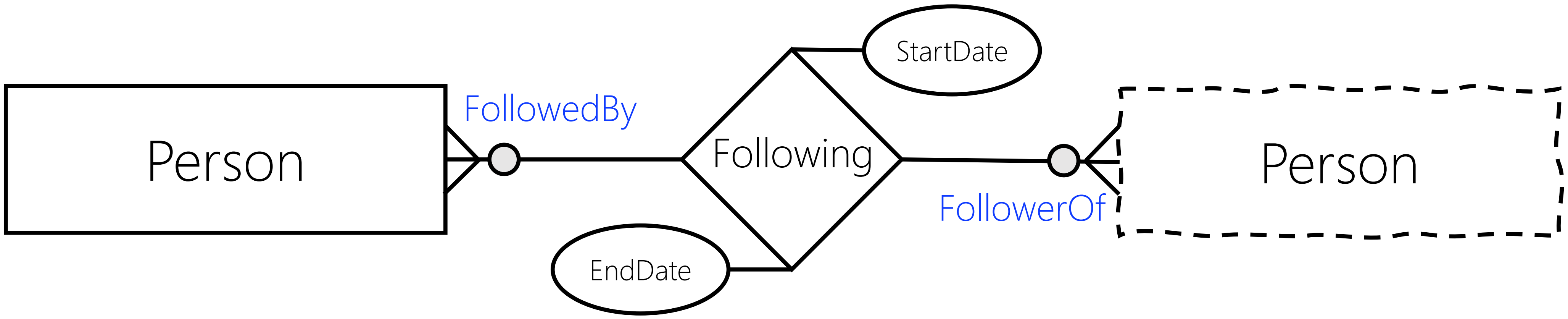
R2R × Many-Many × Self (Unary)



R_1: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_2: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_3: Followership(FirstName, LastName, DateOfBirth, PlaceOfBirth,
R_3: Followership FirstName, LastName, DateOfBirth, PlaceOfBirth,
R_3: Followership(StartDate, EndDate)

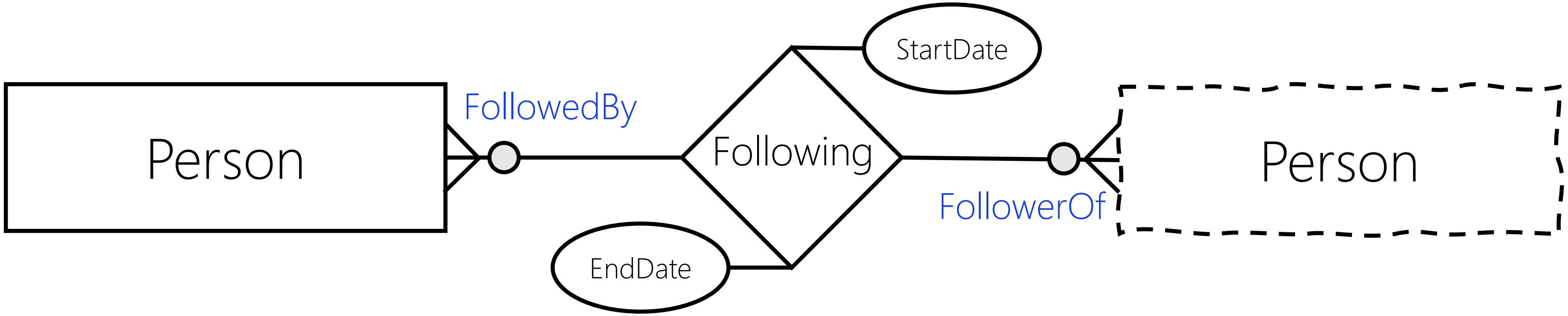
Naming conflict!

R2R × Many-Many × Self (Unary)



R_1: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_2: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_3: Followership(FollowerFirstName, FollowerLastName, FollowerDateOfBirth, FollowerPlaceOfBirth,
R_3: Followership FollowedFirstName, FollowedLastName, FollowedDateOfBirth, FollowedPlaceOfBirth,
R_3: Followership(StartDate, EndDate)

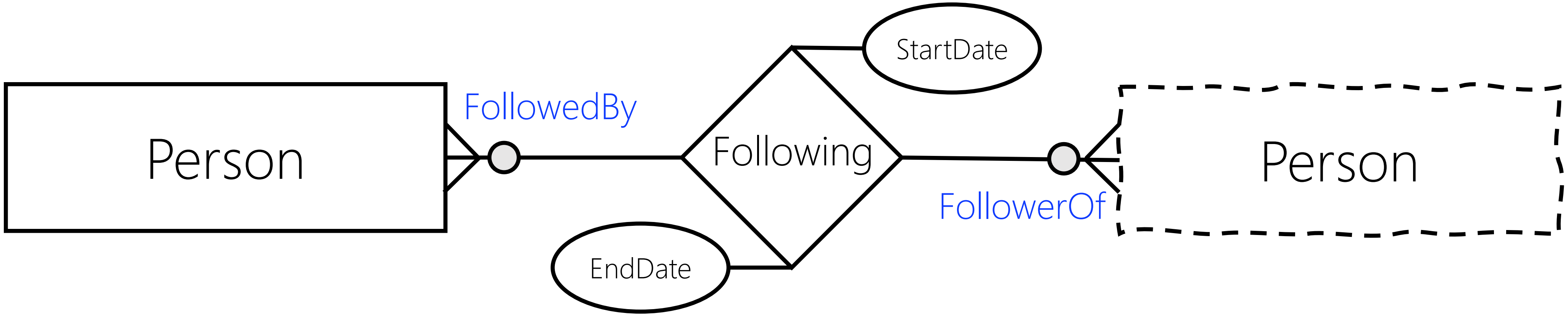
R2R × Many-Many × Self (Unary)



R_1: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_2: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_3: Followership(FollowerFirstName, FollowerLastName, FollowerDateOfBirth, FollowerPlaceOfBirth,
R_3: Followership FollowedFirstName, FollowedLastName, FollowedDateOfBirth, FollowedPlaceOfBirth,
R_3: Followership(StartDate, EndDate)

PK??

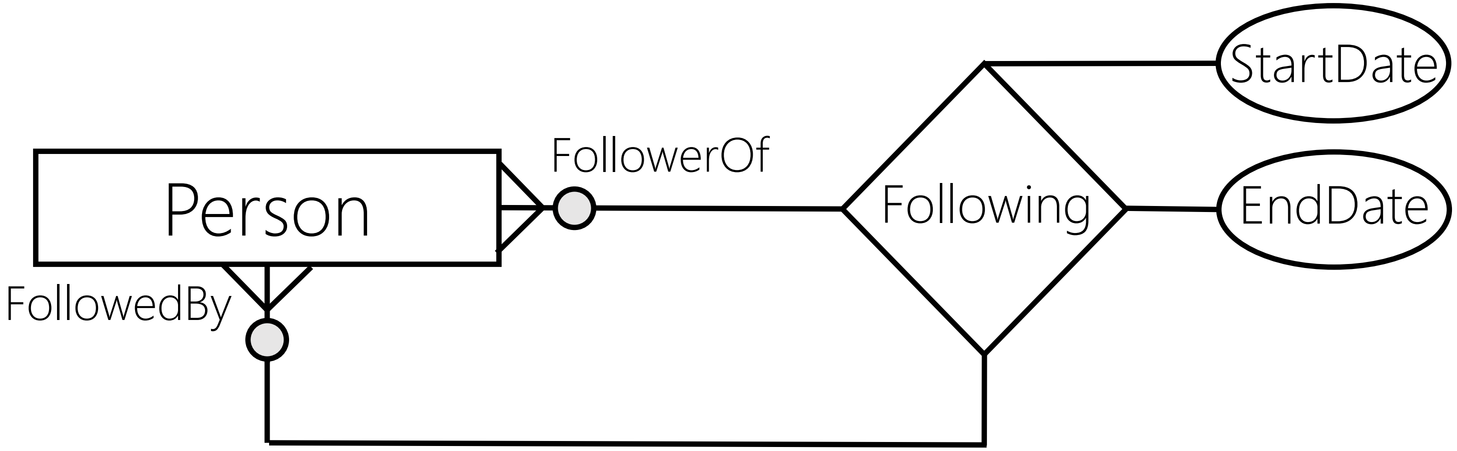
R2R × Many-Many × Self (Unary)



R_1: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_2: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, …)
R_3: Followership(FollowerFirstName, FollowerLastName, FollowerDateOfBirth, FollowerPlaceOfBirth,
R_3: Followership FollowedFirstName, FollowedLastName, FollowedDateOfBirth, FollowedPlaceOfBirth,
R_3: Followership(StartDate, EndDate)

Should we include StartDate?

R2R × Many-Many × Self (Unary)



R_1: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, ...)

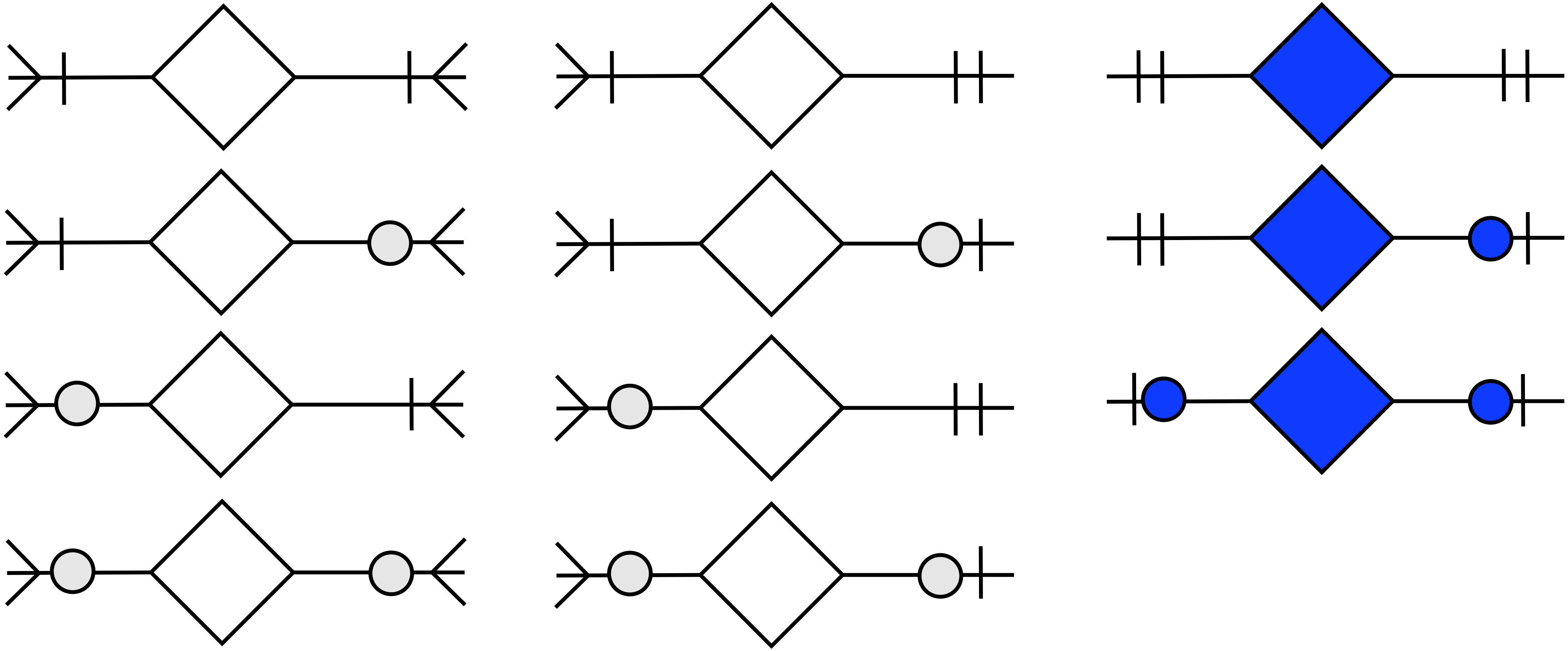
~~R_2: Person(FirstName, LastName, DateOfBirth, PlaceOfBirth, Age, ...)~~

R_2: Followership(FollowerName, FollowerName, FollowerDateOfBirth, FollowerPlaceOfBirth,

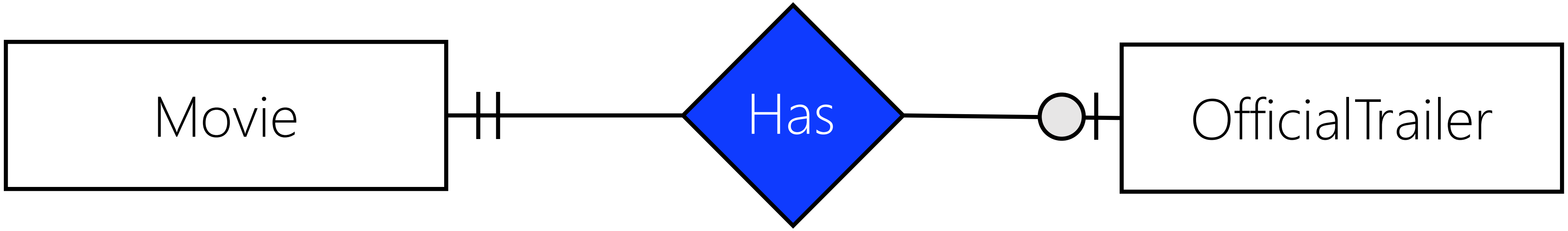
R_3: Followership(FollowedFirstName, FollowedLastName, FollowedDateOfBirth, FollowedPlaceOfBirth,

R_3: Followership(StartDate, EndDate)

Relationship2Relation (R2R)

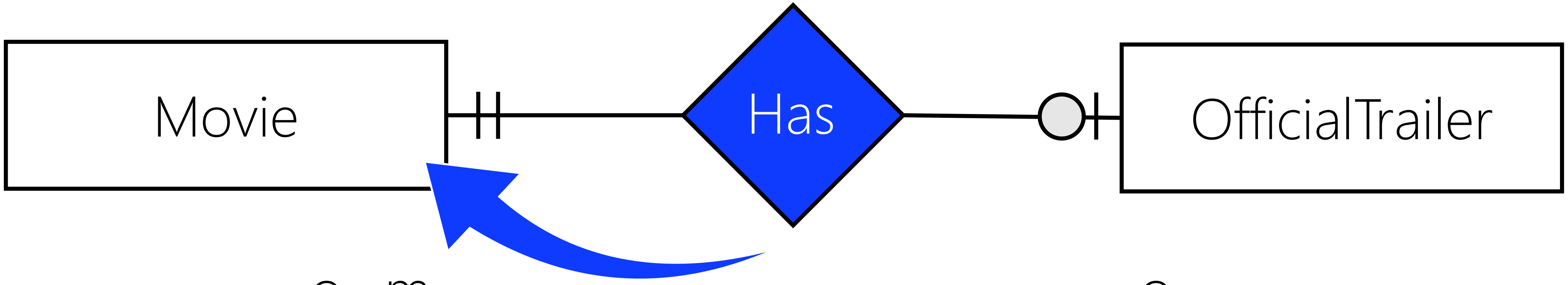


R2R × One-One



m_1	m_1, o_1	o_1
m_2	m_2, o_2	o_2
m_3	~~m_1, o_2~~	
	~~m_2, o_1~~	

R2R × One-One (Approach I)



o_1, m_1

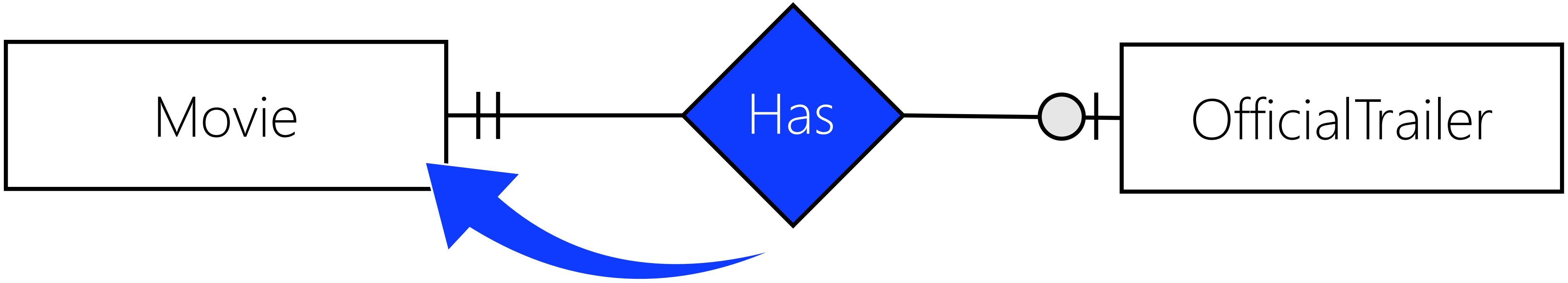
o_2, m_2

NULL, m_3

o_1

o_2

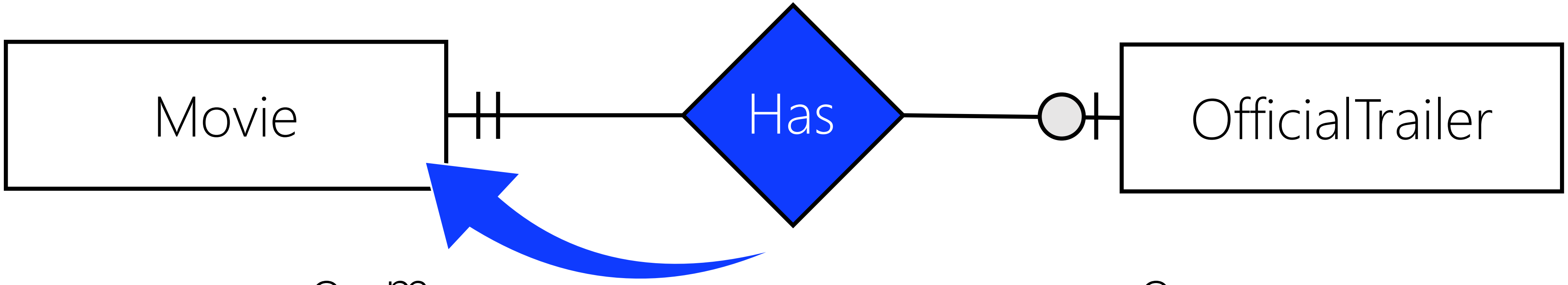
R2R × One-One (Approach I)



R_1: OfficialTrailer($\underline{\text{Url}}$, RunningTime)
R_2: Movie($\underline{\text{Title}}$, $\underline{\text{ReleaseYear}}$, $\underline{\text{ReleaseMonth}}$, $\underline{\text{ReleaseDay}}$, RunningTime, Age, $\underbrace{\text{OfficialTrailer.Url}}_{\text{FK}}$)

There is a problem! Where? Solution?

R2R × One-One (Approach I)



o_1, m_1 o_1

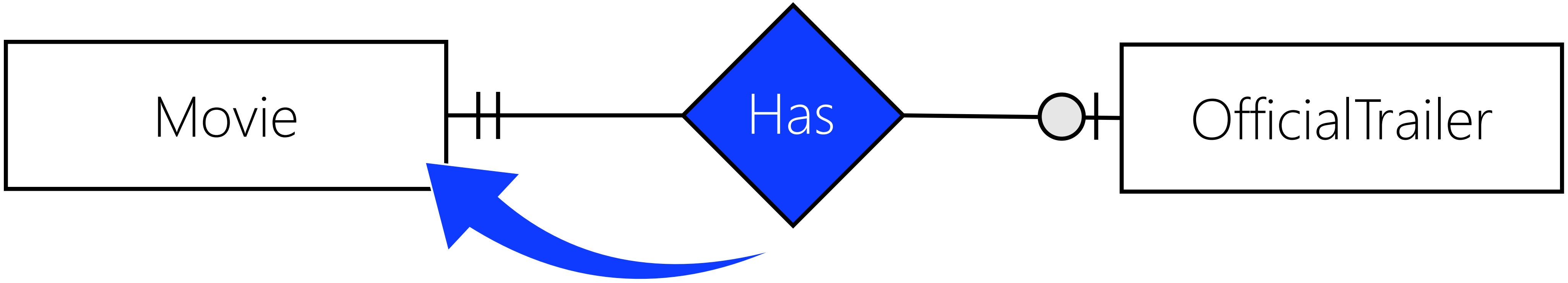
o_2, m_2 o_2

NULL, m_3

~~o_2, m_1~~ Not possible because of key.

o_2, m_4 Possible. Nothing prevents!

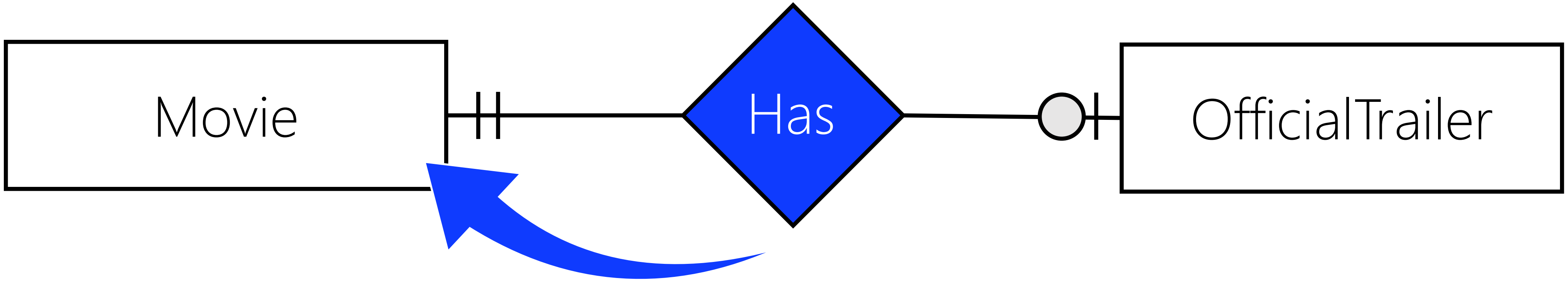
R2R × One-One (Approach I)



R_1: OfficialTrailer(Url, RunningTime)
R_2: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age, OfficialTrailer.Url)

The Movie relation allows an official trailer belongs to multiple movies!
We must make OfficialTrailer.Url unique. How? By being part of key set?

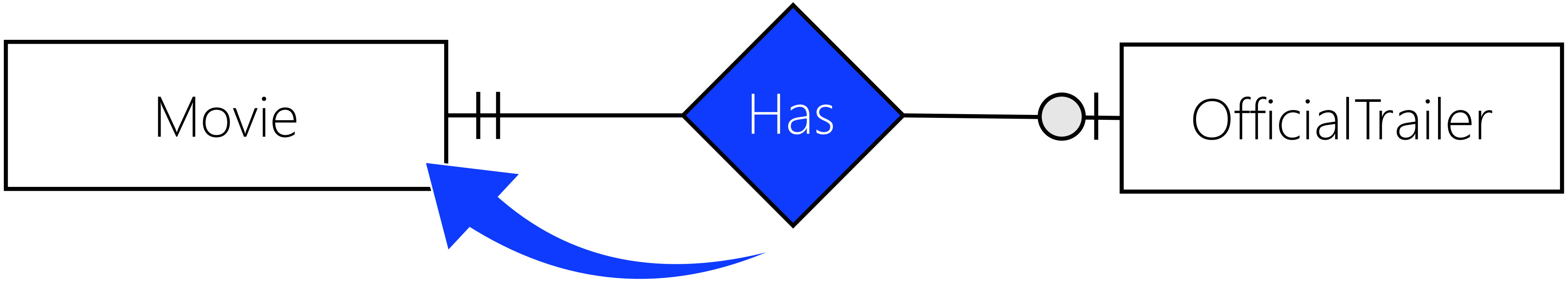
R2R × One-One (Approach I)



R_1: OfficialTrailer(Url, RunningTime)
R_2: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age, OfficialTrailer.Url)

The Movie relation allows an official trailer belongs to multiple movies!
We have to make OfficialTrailer.Url unique. How? By being part of key set? No! it does not solve the issue.
By being a new key set?

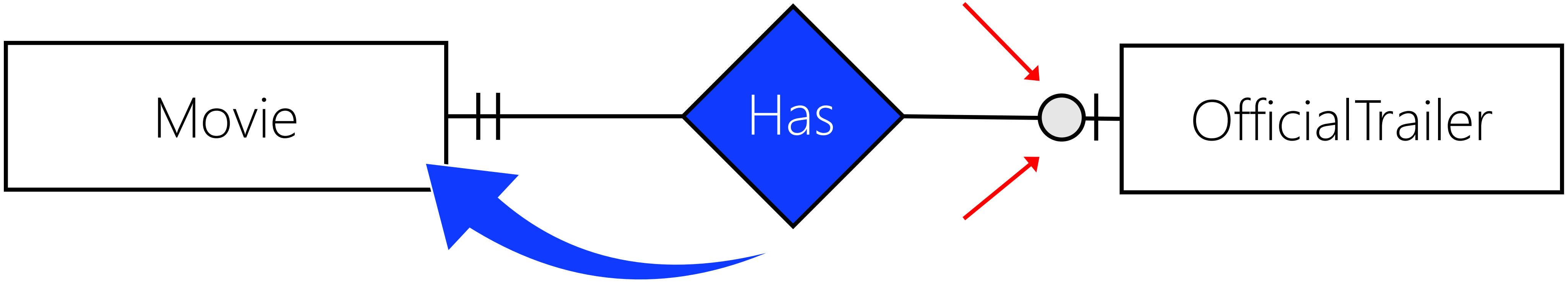
R2R × One-One (Approach I)



R_1: OfficialTrailer(<u>Url</u>, RunningTime)
R_2: Movie(<u>Title</u>, <u>ReleaseYear</u>, <u>ReleaseMonth</u>, <u>ReleaseDay</u>, RunningTime, Age, OfficialTrailer.Url)

New key set for R_2:
Secondary Key Set ={OfficialTrailer.Url}

R2R × One-One (Approach I)



R$_1$: OfficialTrailer(Url, RunningTime)
R$_2$: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age, OfficialTrailer.Url)

New key set for R$_2$:
SK={OfficialTrailer.Url} but if we choose this as key set, it should be mandatory!