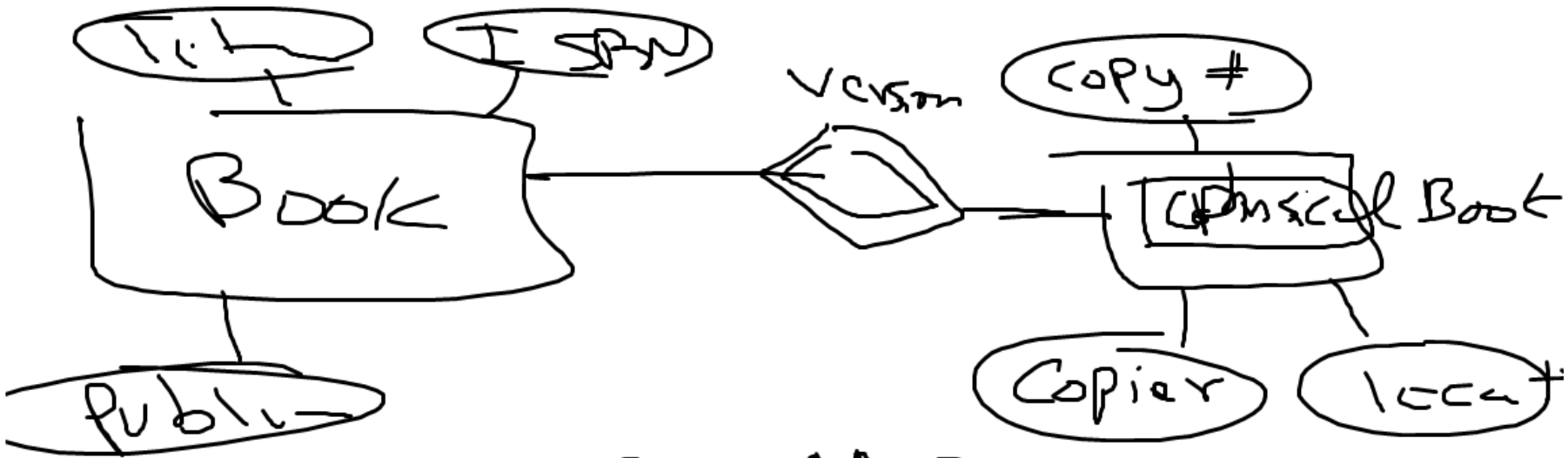
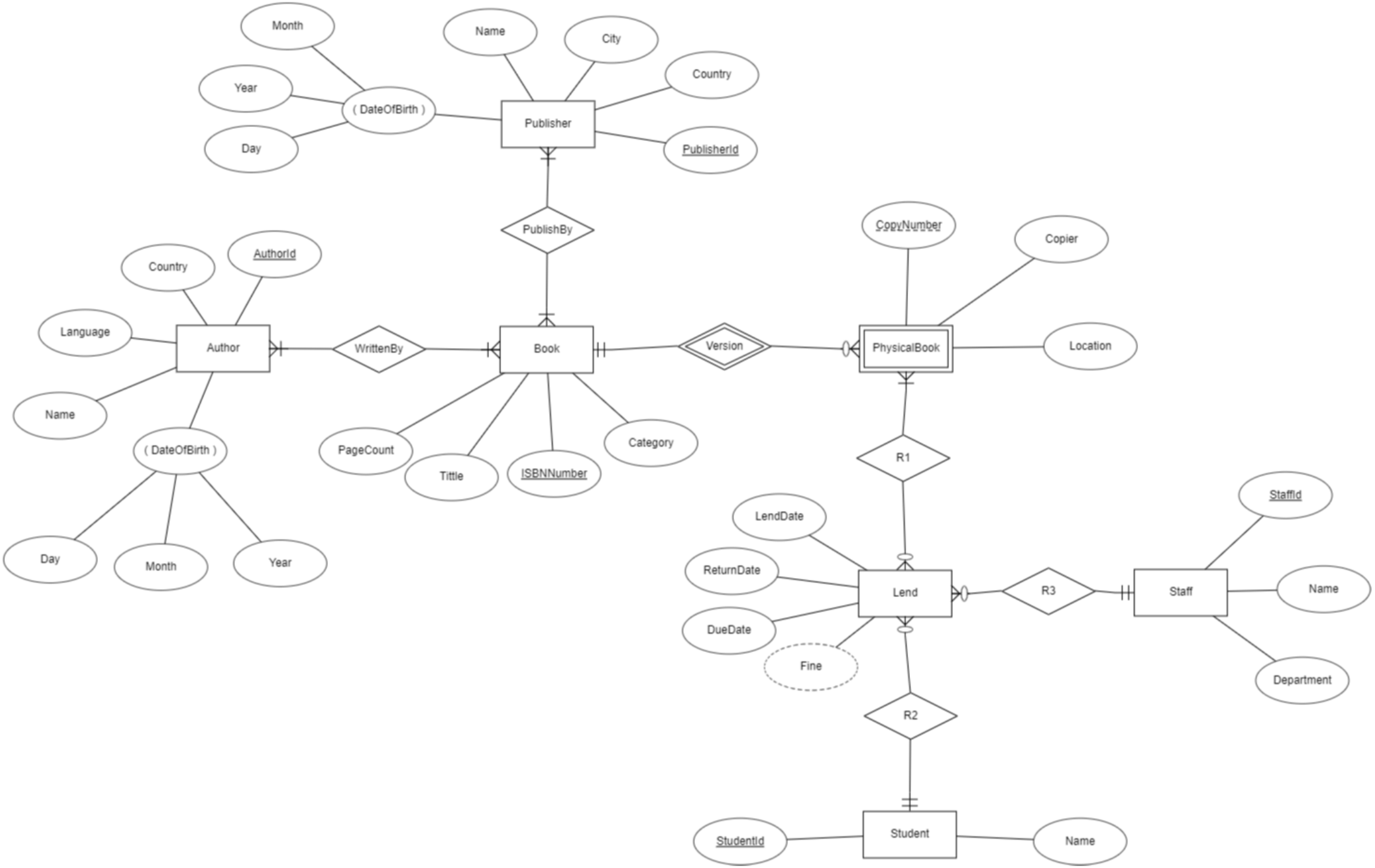

Book
Version
Copy #
Physical Book
Copier

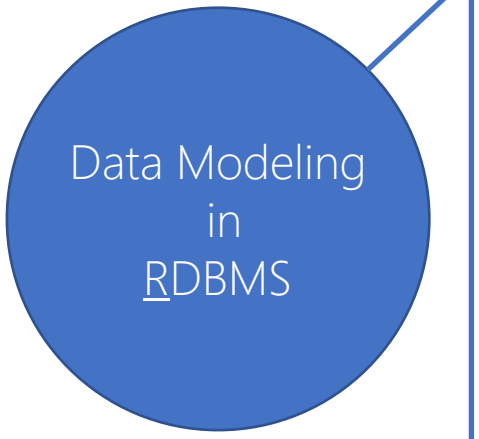
Weak Entity Set

An Example


Month
Year
Day
(DateOfBirth)
Name
City
Country
PublisherId
Publisher
PublishBy
CopyNumber
Copier
Location
PhysicalBook
Version
Book
WrittenBy
Author
Country
AuthorId
Language
Name
(DateOfBirth)
Day
Month
Year
PageCount
Tittle
ISBNNumber
Category
R1
LendDate
ReturnDate
DueDate
Fine
Lend
R3
Staff
StaffId
Name
Department
R2
Student
StudentId
Name

Logical Level | Relational Data Model

Today



Real World Entity

Conceptual Level | Entity-Relationship Model (E/R) Level

Conceptual Level | Logical Level | Relational Model

Conceptual Level | Logical Level | Physical Level | SQL

Conceptual Level | Logical Level | Computable Entity

Data Modeling × Logical Level

1. How entities, attributes, relationships instances should be represented.
2. Update Schema

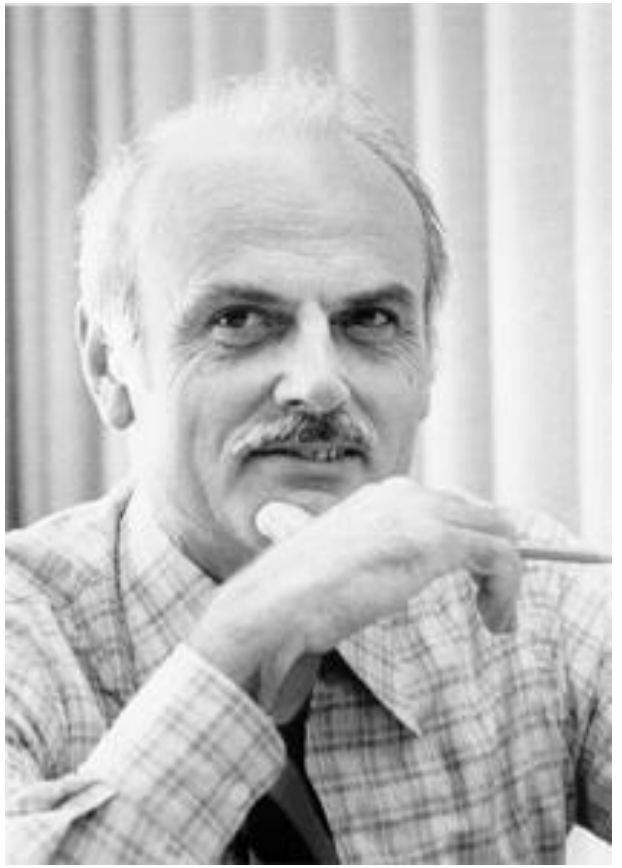
Data Modeling × Logical Level

There are other representations as well.
There are other logical models as well.
There are other data models at logical level as well.

- ~1960: Object Oriented
- ~1969: Relational: Mathematical Relation
- ~1996: XML

Relational

Edgar Frank "Ted" Codd, IBM, 1969, 1970



Information Retrieval

A Relational Model of Data for Large Shared Data Banks

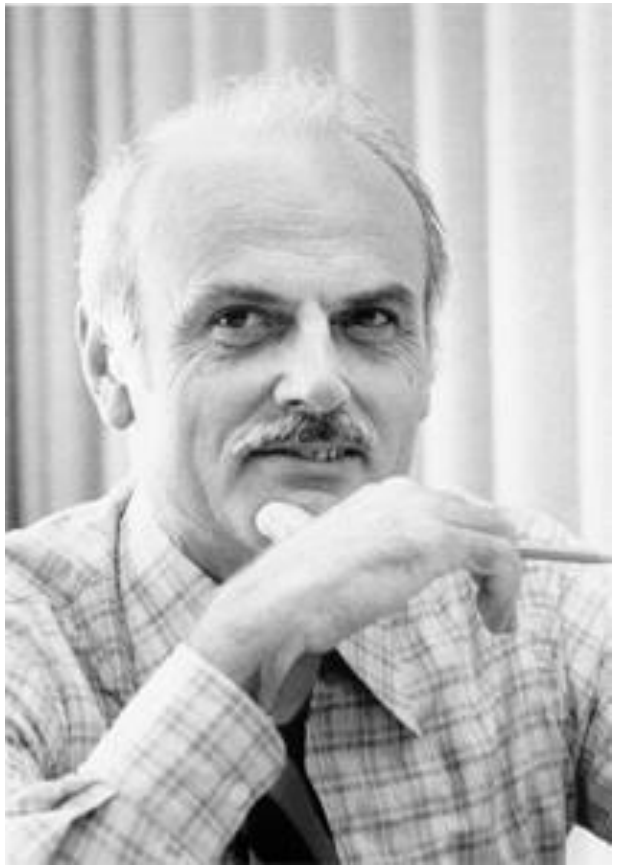
E. F. Codd
IBM Research Laboratory, San Jose, California

Relational

Data instance is represented in terms of Tuple
Tuples are grouped in Relation

Data Definition Language (DDL)
Data Manipulation Language (DML)
→ Relational Algebra

Solid Mathematical Model



Relational × Relation (R)

Two-dimensional table, e.g., Movie Relation

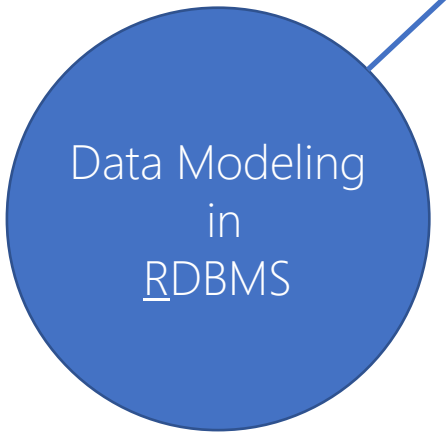
Schema	Title	Language	RunningTime
Tuple 1	*2001: A Space Odyssey*	*English*	*142*
Tuple 2	*Rosemary's Baby*	*English*	*136*
Tuple 3	*The Birds*	*English*	*119*
…	…	…	…

Informally: Relation | Table
Informally: Tuple | Row
Informally: Attribute | Column

Relational × Relation (R)

Everything is Relation (Table)

Entity & Relationship 2 Relation



Real World Entity

Conceptual Level | Entity & Relationship Level

Conceptual Level | Logical Level | Relation (Table)

Conceptual Level | Logical Level | Physical Level | SQL

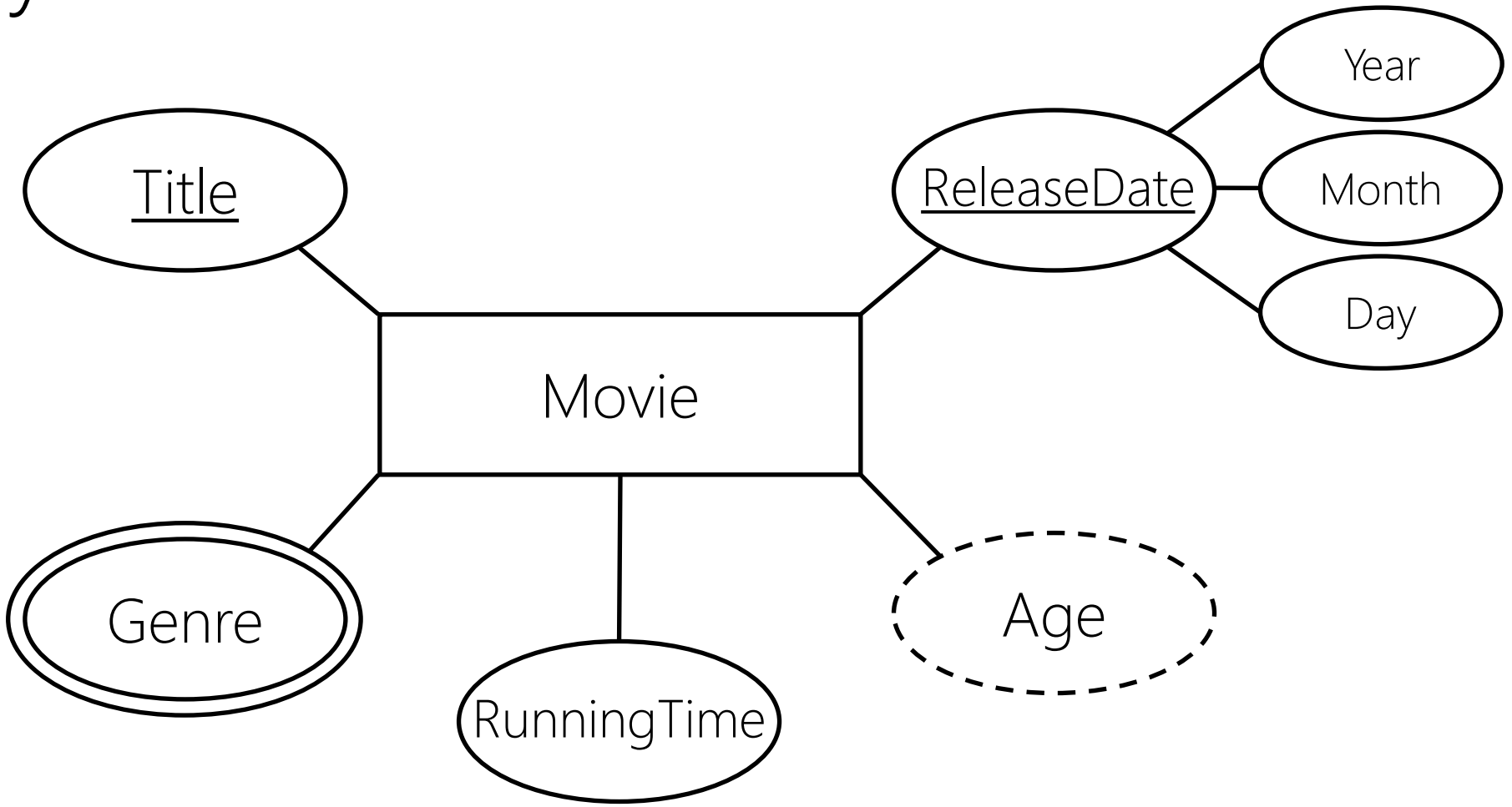
Conceptual Level | Logical Level | Computable Entity

Relational × Relation (R)

Everything is Relation (Table)

Entity → Relation
Relationship → Relation

Entity2Relation



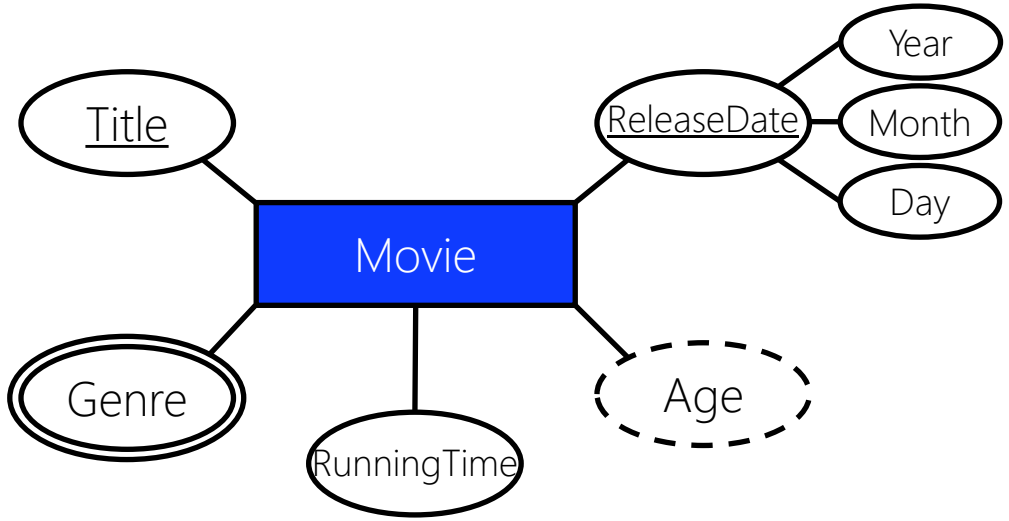
Entity2Relation (E2R)

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

R_2: Genre(Title)

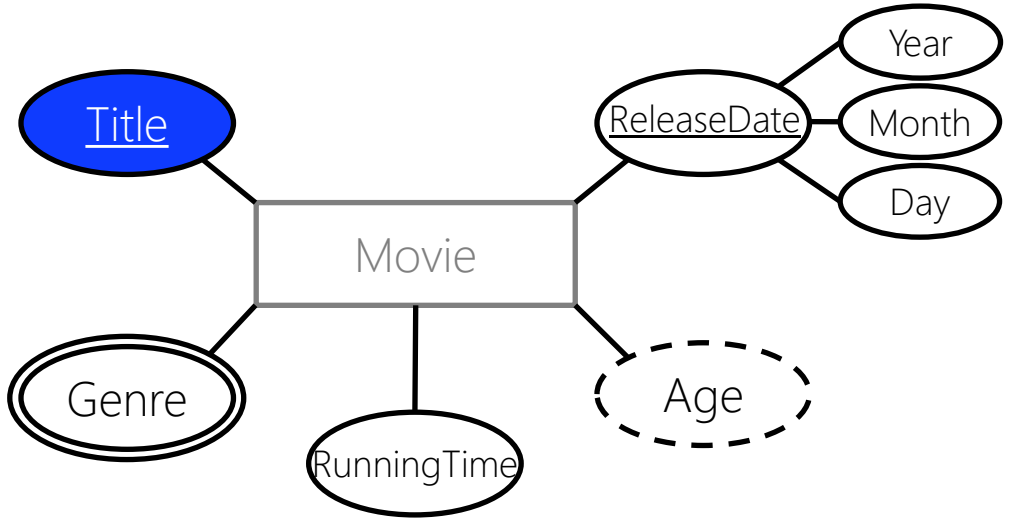
R_3: MovieGenre(Movie.Title, Movie.ReleaseYear, Movie.ReleaseMonth, Movie.ReleaseDay, Genre.Title)

E2R × Entity Set



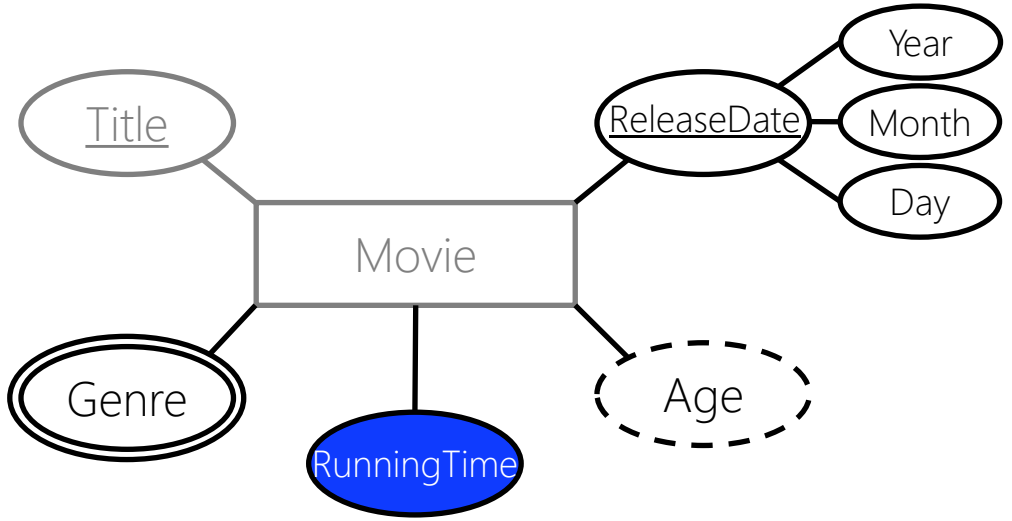
R_1: Movie()

E2R × Entity Set × Attribute



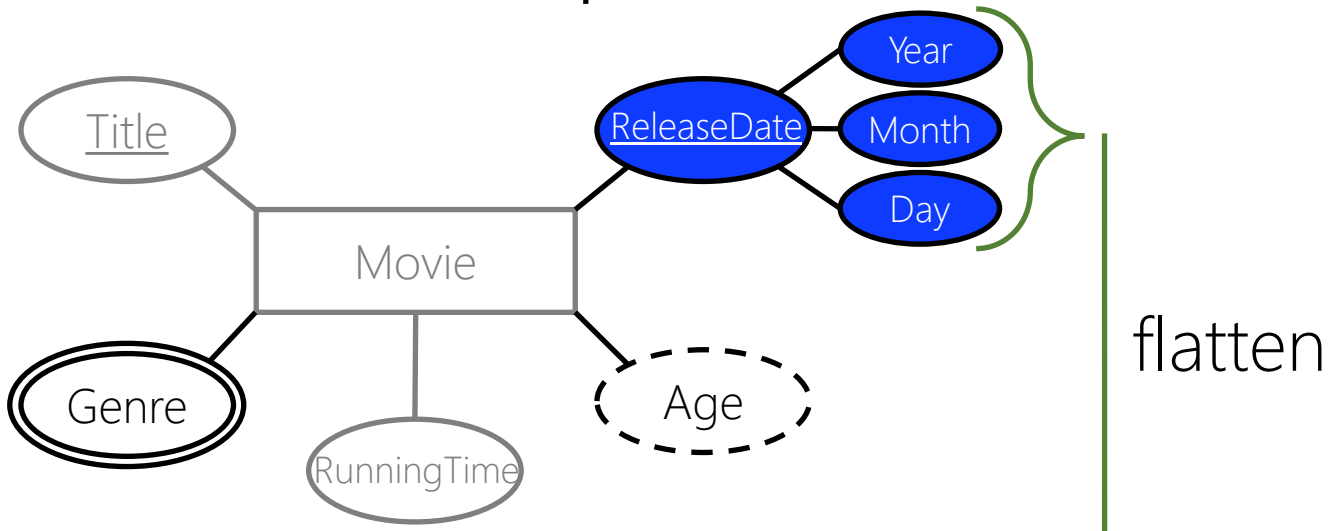
R_1: Movie(Title)

E2R × Entity Set × Attribute



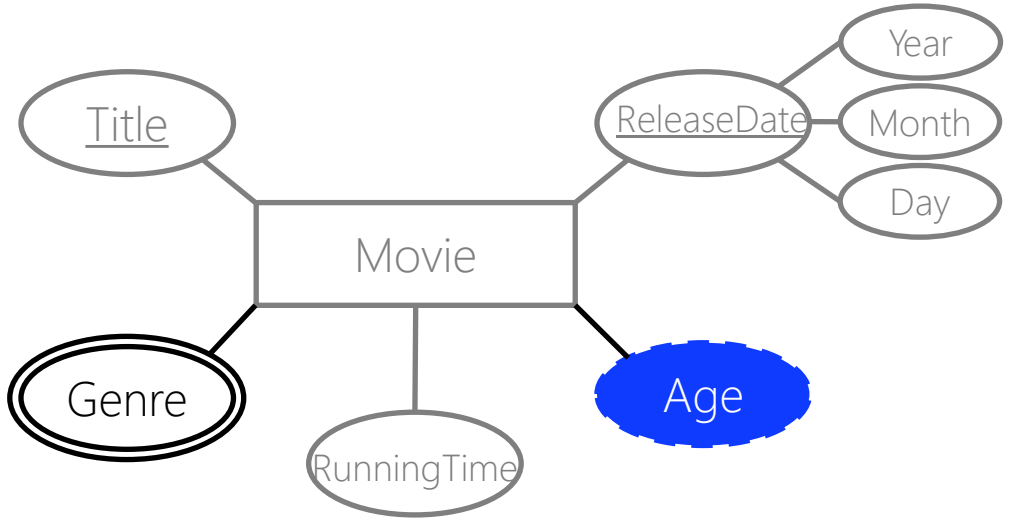
R_1: Movie(Title, RunningTime)

E2R × Entity Set × Composite Attribute



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

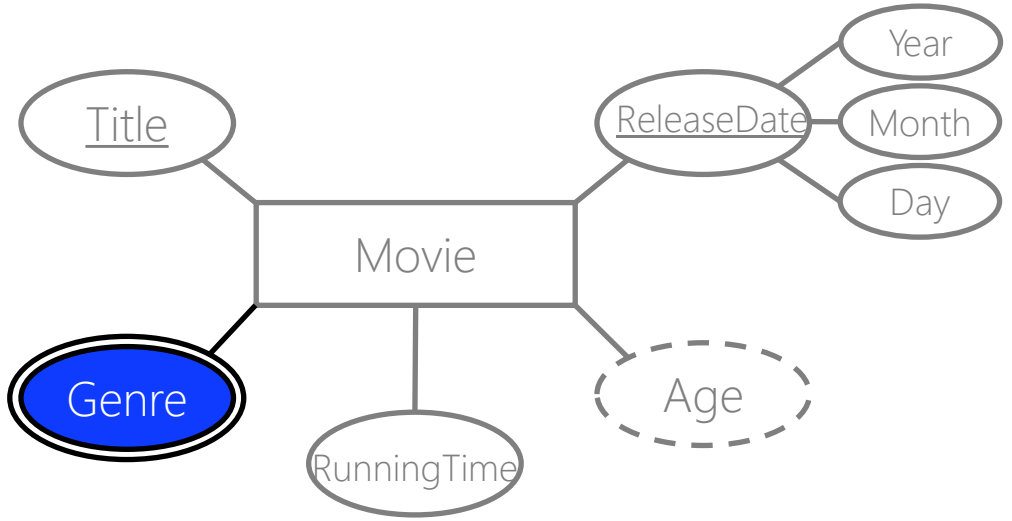
E2R × Entity Set × Derived Attribute



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

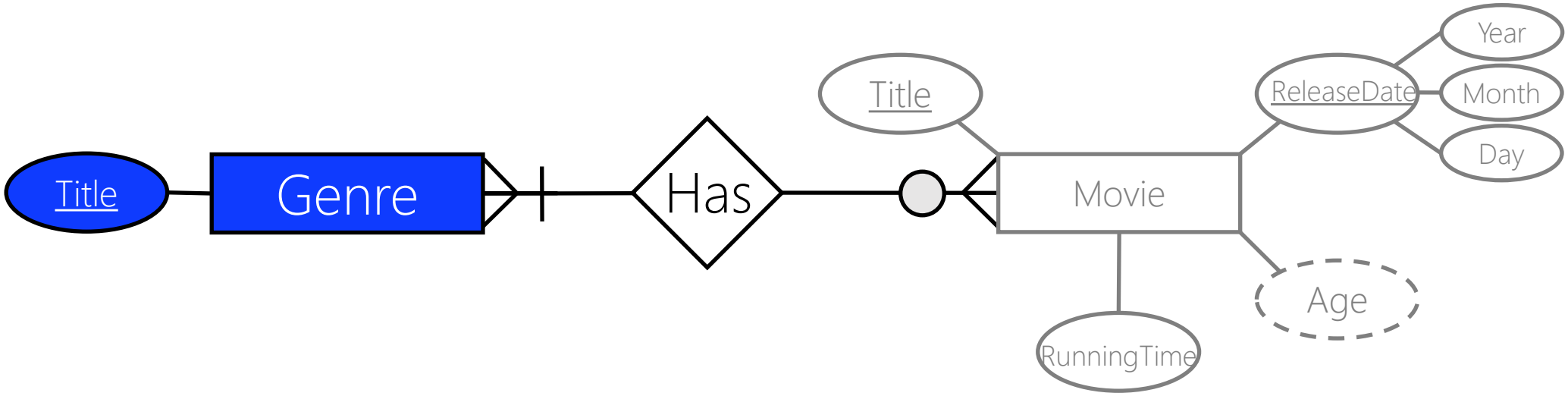
E2R × Entity Set × Multivalued Attribute

Relational model does not allow multivalued attributes!



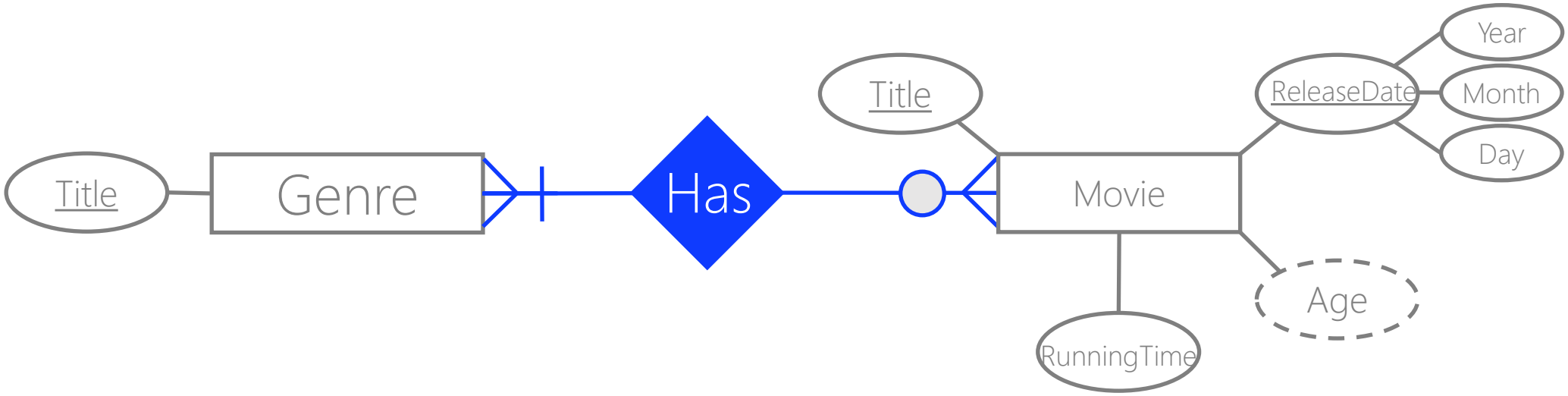
R_1: Movie(Title, RunningTime, ReleaseYear, ReleaseMonth, ReleaseDay, Age)

E2R × Entity Set × Multivalued Attribute



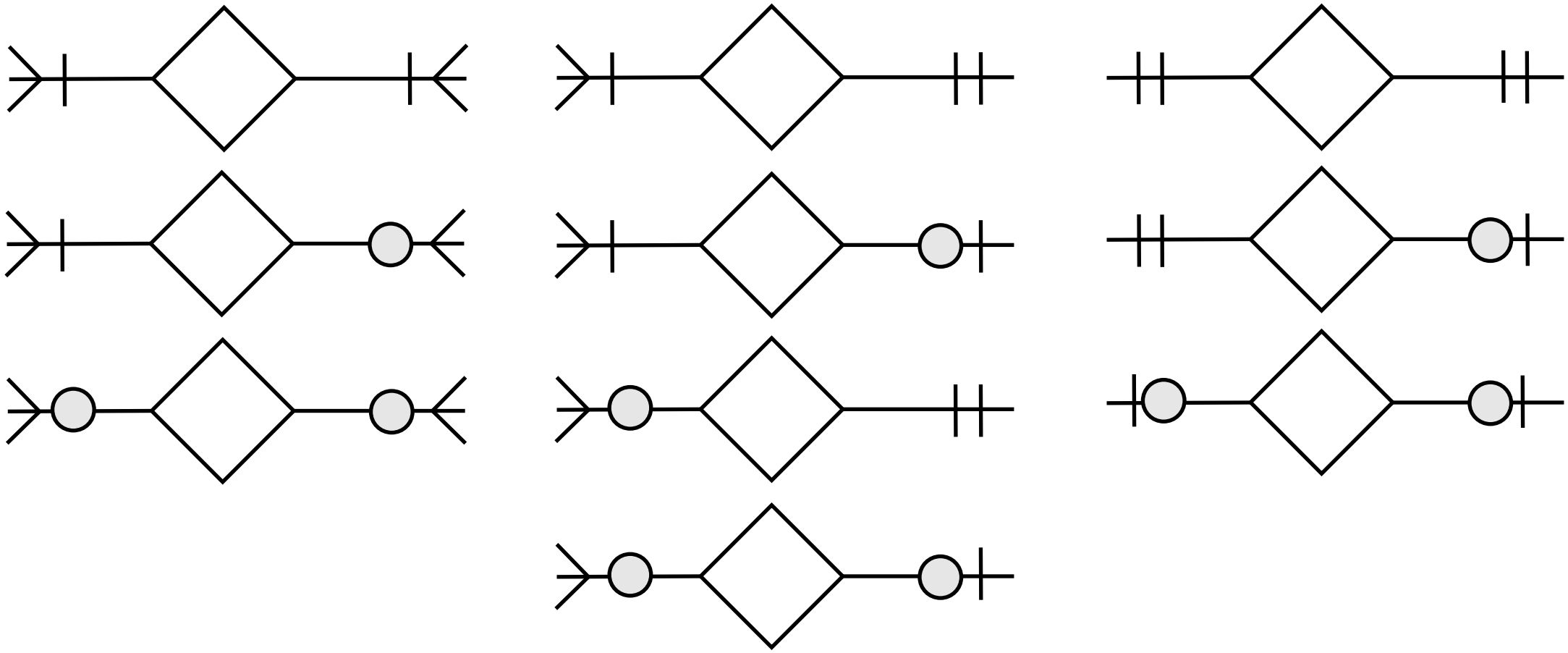
R_1: Movie(Title, RunningTime, ReleaseYear, ReleaseMonth, ReleaseDay, Age)
R_2: Genre(Title)

E2R × Entity Set × Multivalued Attribute

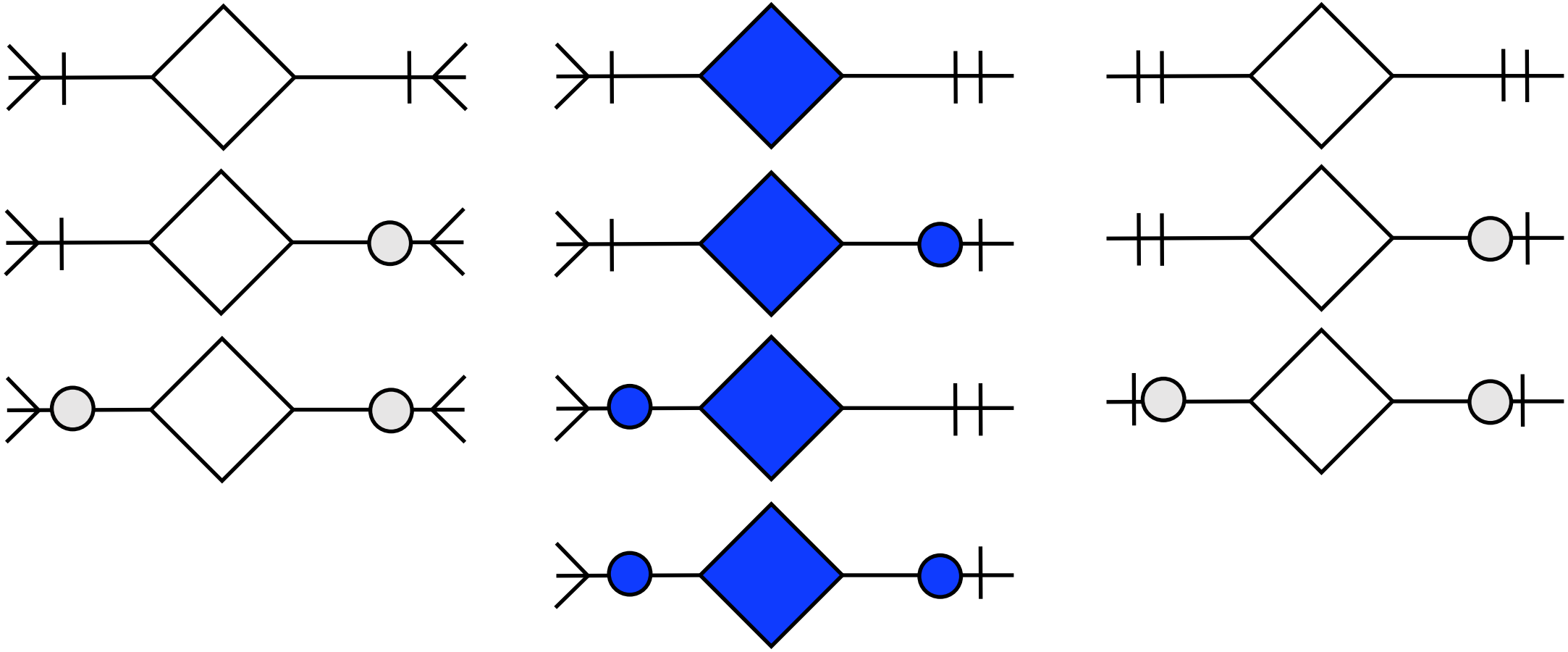


R_1: Movie(Title, RunningTime, ReleaseYear, ReleaseMonth, ReleaseDay, Age)
R_2: Genre(Title)

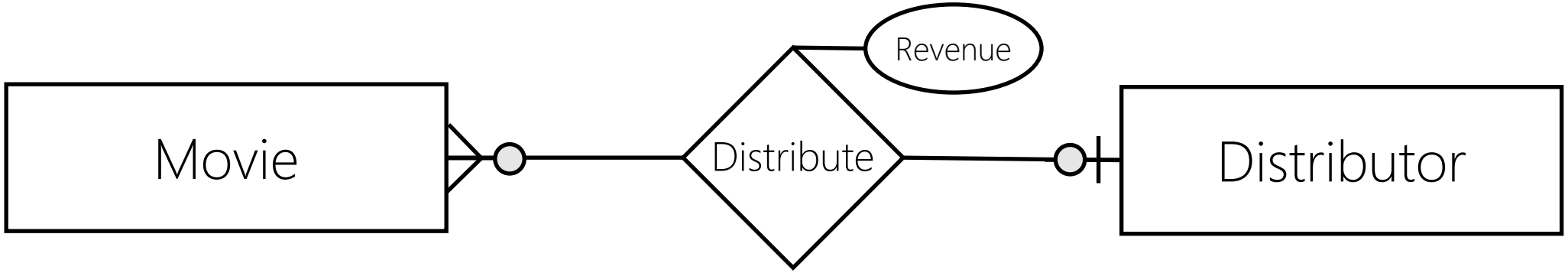
Relationship2Relation (R2R)



Relationship2Relation (R2R)

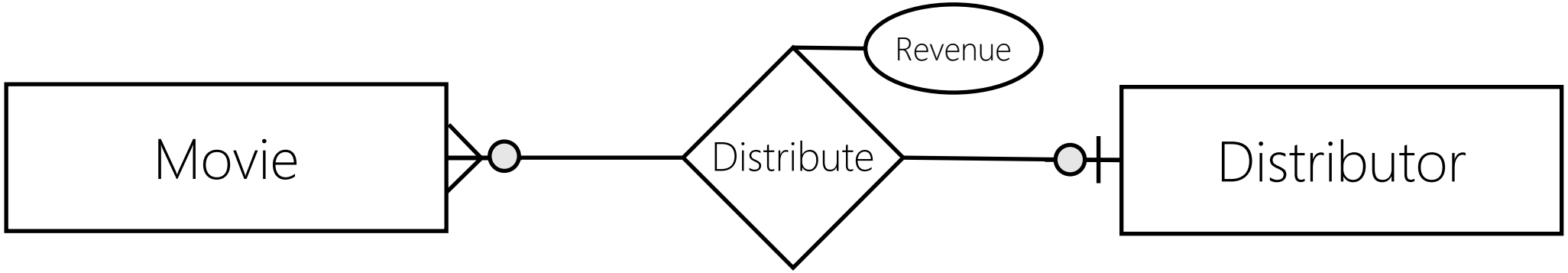


R2R × Many-One (One-Many)



m_1	m_2, d_3, \$20M	d_1
m_2	~~m_2, d_1, \$28M~~	d_2
m_3	m_3, d_3, \$35M	d_3
m_4	m_4, d_2, \$27M	d_4
m_5	m_5, d_1, \$13M	

R2R × Many-One (One-Many)



m_1

m_2

m_3

m_4

m_5

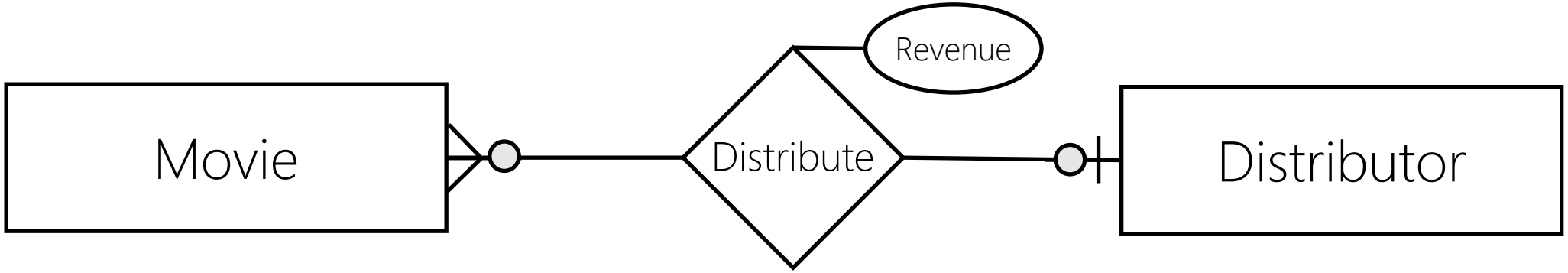
d_1, m_5, \$13M

d_2, m_4, \$27M

d_3, m_2, \$20M, m_3, \$35M

d_4

R2R × Many-One (One-Many)



NULL, NULL, m_1

d_3, \$20M, m_2

d_3, \$35M, m_3

d_2, \$27M, m_4

d_1, \$13M, m_5

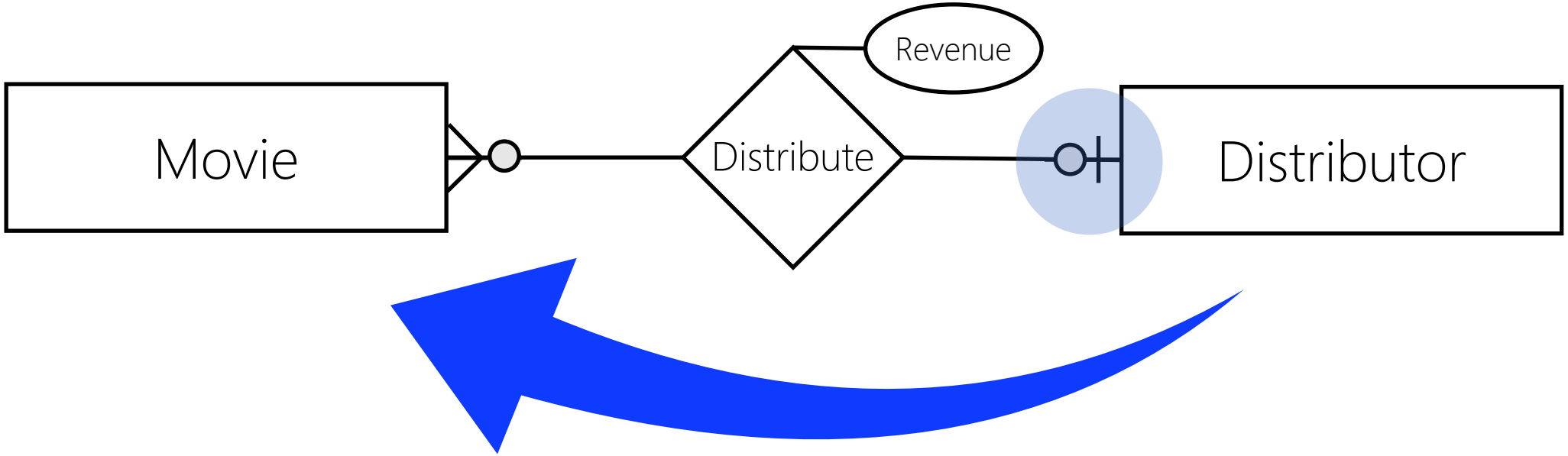
d_1

d_2

d_3

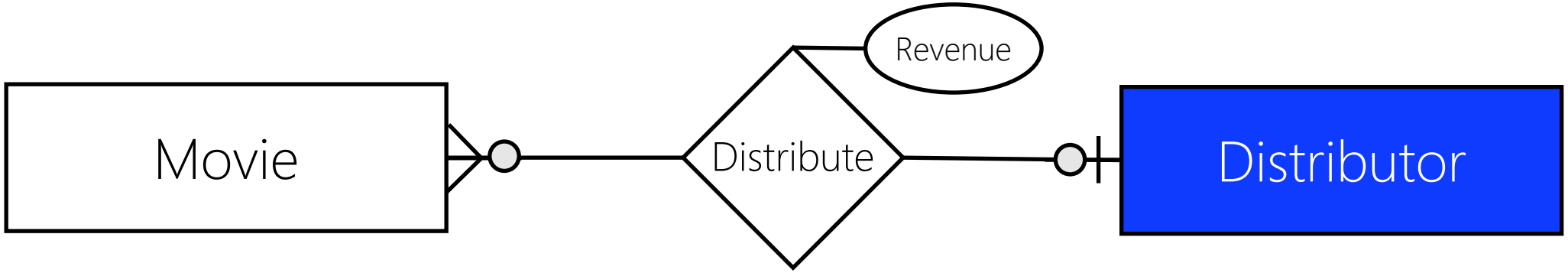
d_4

R2R × Many-One (One-Many)



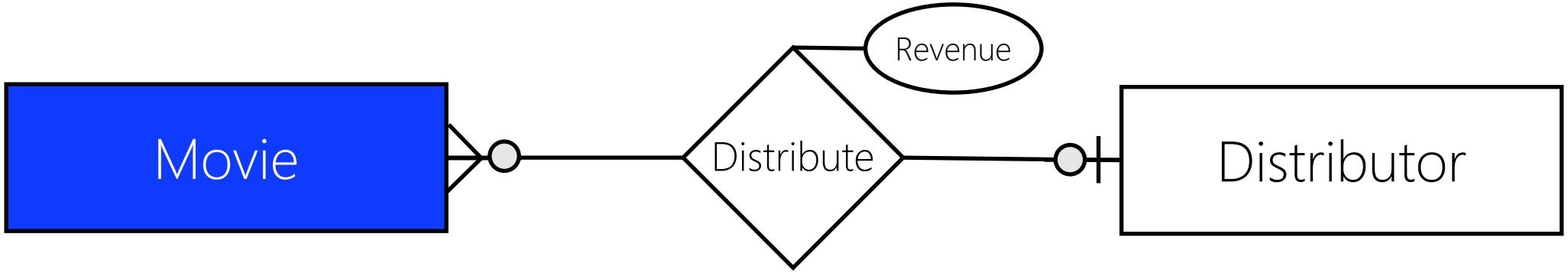
Everything goes to entity set with cardinality one (i.e., many side)
Because it only needs to store one entity from other entity set

R2R × Many-One (One-Many)



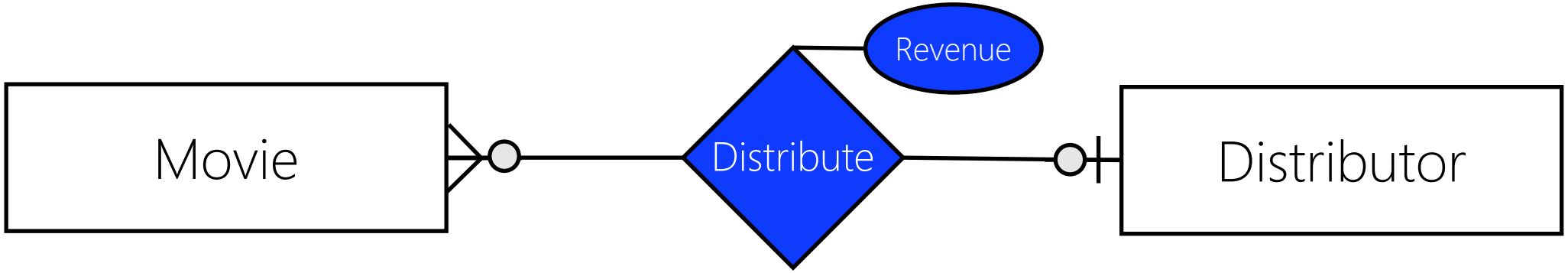
R_1: Distributor(Name, Address, POBox, Website, ...)

R2R × Many-One (One-Many)



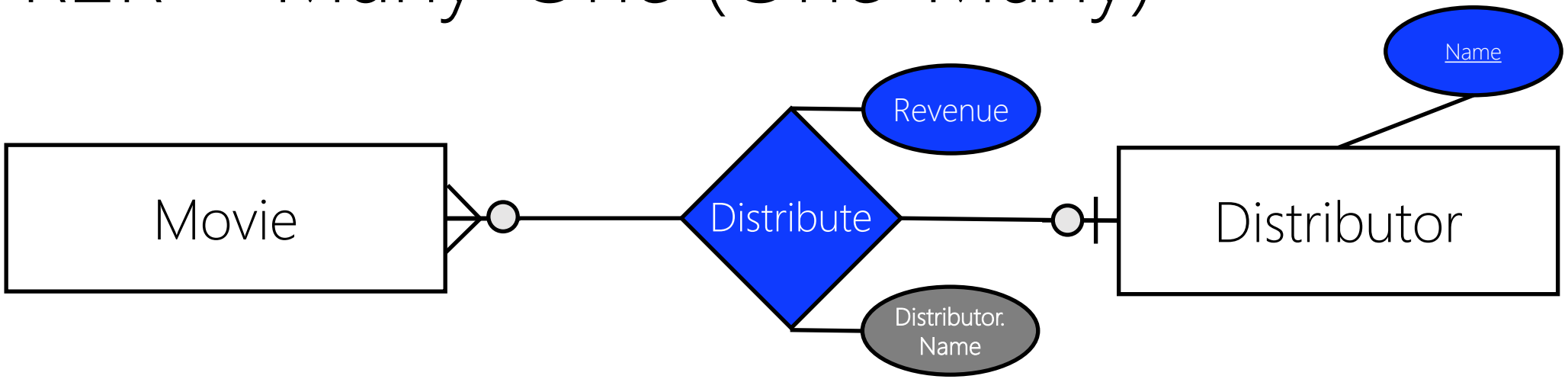
R_1: Distributor(Name, Address, POBox, Website, …)
R_2: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age)

R2R × Many-One (One-Many)



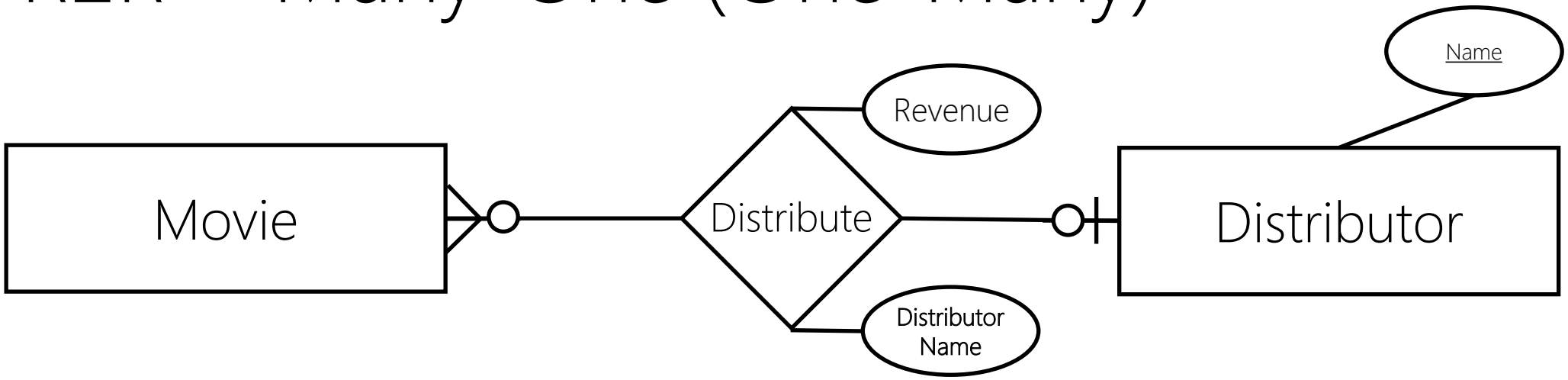
R_1: Distributor(Name, Address, POBox, Website, ...)
R_2: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age, ...
R_2: Movie Revenue)

R2R × Many-One (One-Many)



R_1: Distributor(Name, Address, POBox, Website, …)
R_2: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age, …
R_2: Movie Distributor.Name, Revenue)

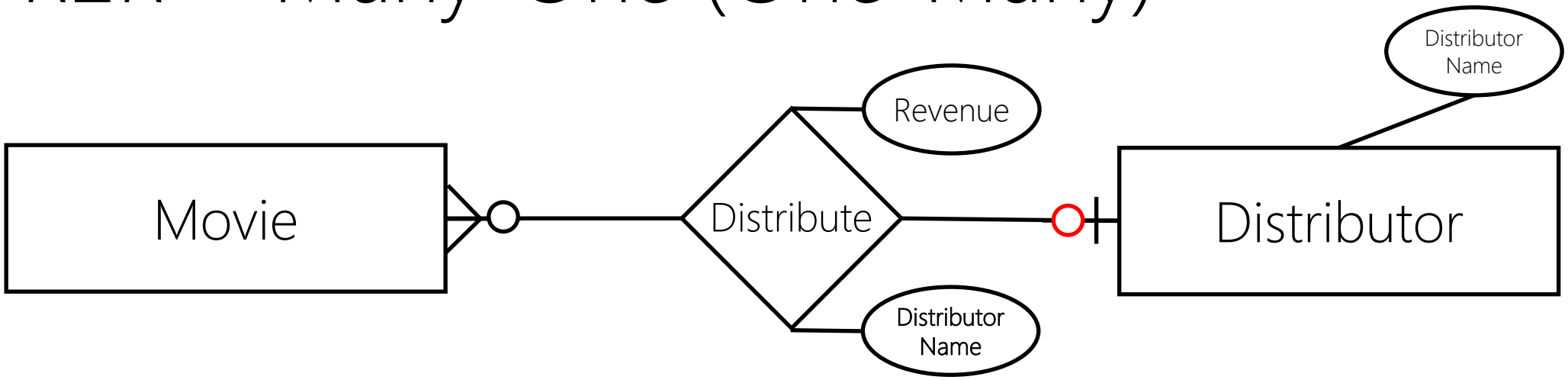
R2R × Many-One (One-Many)



R_1: Distributor(Name, Address, POBox, Website, …)
R_2: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age, …
R_2: Movie Distributor.Name, Revenue)

Primary Key (PK) from other relation: Foreign Key (FK)

R2R × Many-One (One-Many)

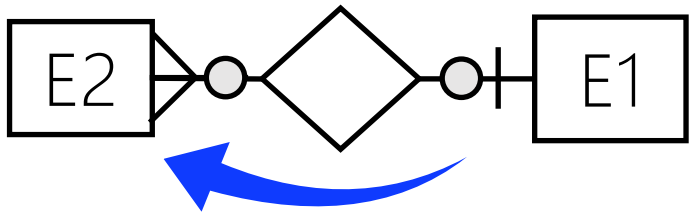


R_1: Distributor(Name, Address, POBox, Website, …)
R_2: Movie(Title, ReleaseYear, ReleaseMonth, ReleaseDay, RunningTime, Age, …
R_2: Movie Distributor.Name, Revenue)

Must be optional (Why?)

R2R × Many-One (One-Many)

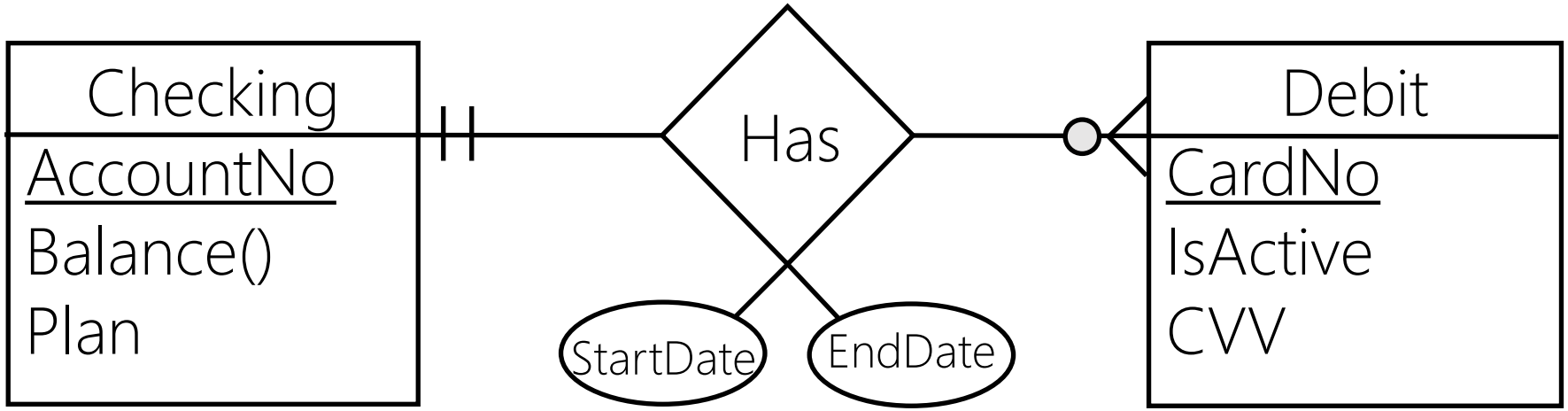
Input: Many-One relationship btw. E2 and E1, i.e.,
Output: Relations R1 for E1 and R2 for E2.



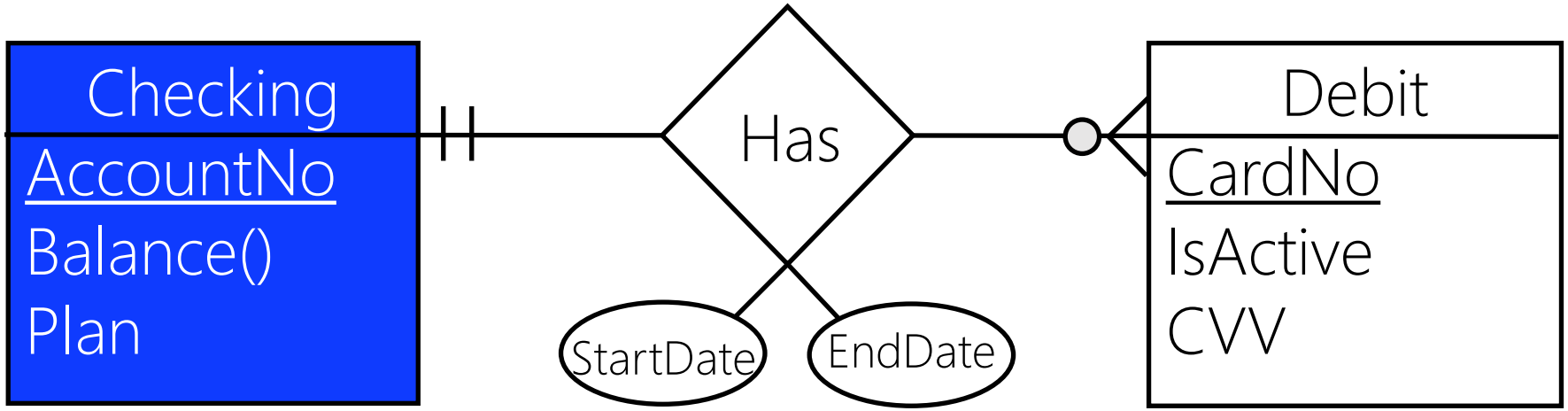
1) For E1, create relation R1 with the same attributes and keys as in E1
2) For E2, create relation R2 with the same attributes and keys as in E2
3) [Foreign Key Set] Add key set of E1 to R2
4) Add attributes of relationship set to R2
5) If E2 ordinality is optional then make foreign key set optional
6) If E2 ordinality is optional else make foreign key set mandatory

Herein, we do not care about E1's ordinality! We fix it later.

R2R × Many-One × Banking

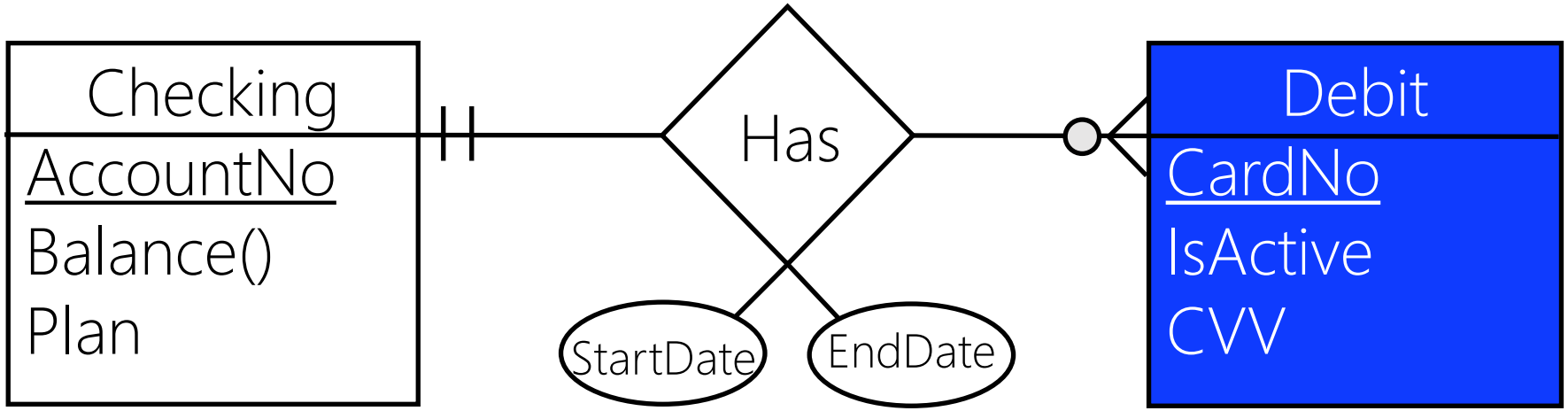


R2R × Many-One × Banking



R1: Checking(AccountNo, Balance, Plan)

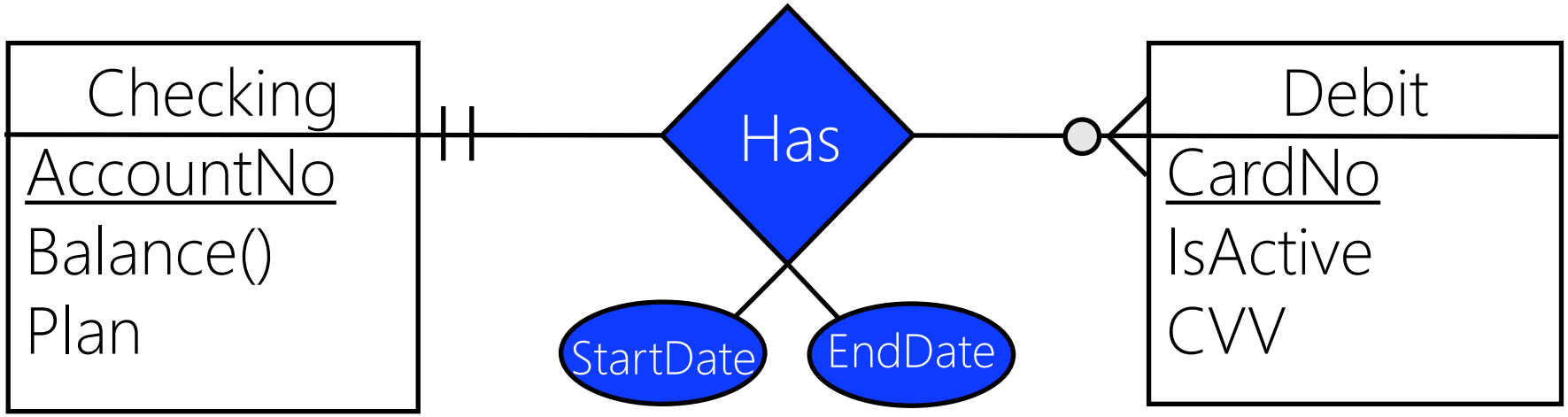
R2R × Many-One × Banking



R1: Checking(AccountNo, Balance, Plan)

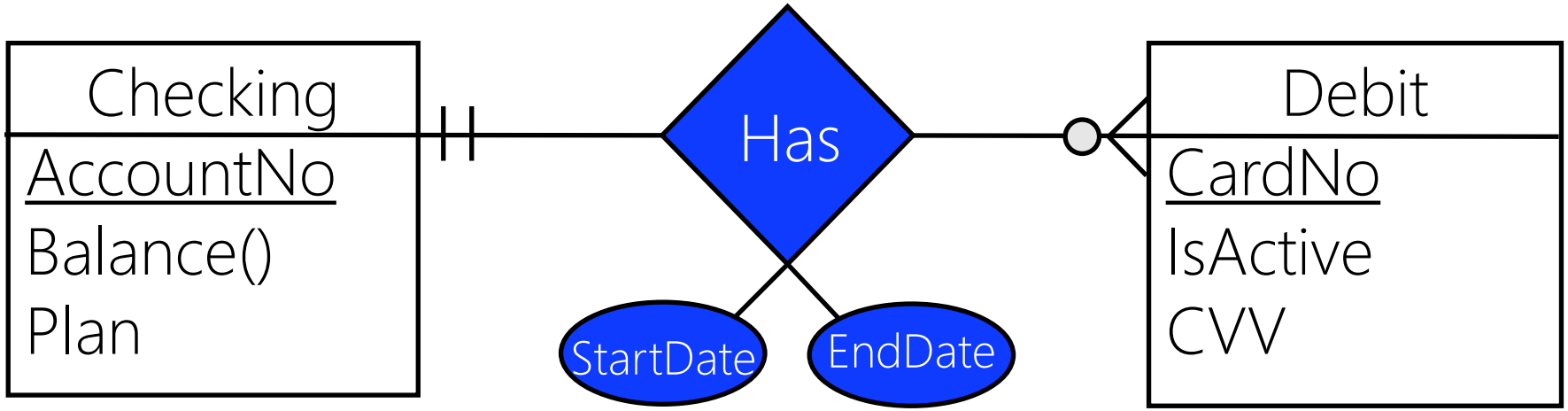
R2: Debit(CardNo, IsActive, CVV)

R2R × Many-One × Banking



R1: Checking(<u>AccountNo</u>, Balance, Plan)?
R2: Debit(<u>CardNo</u>, IsActive, CVV)?

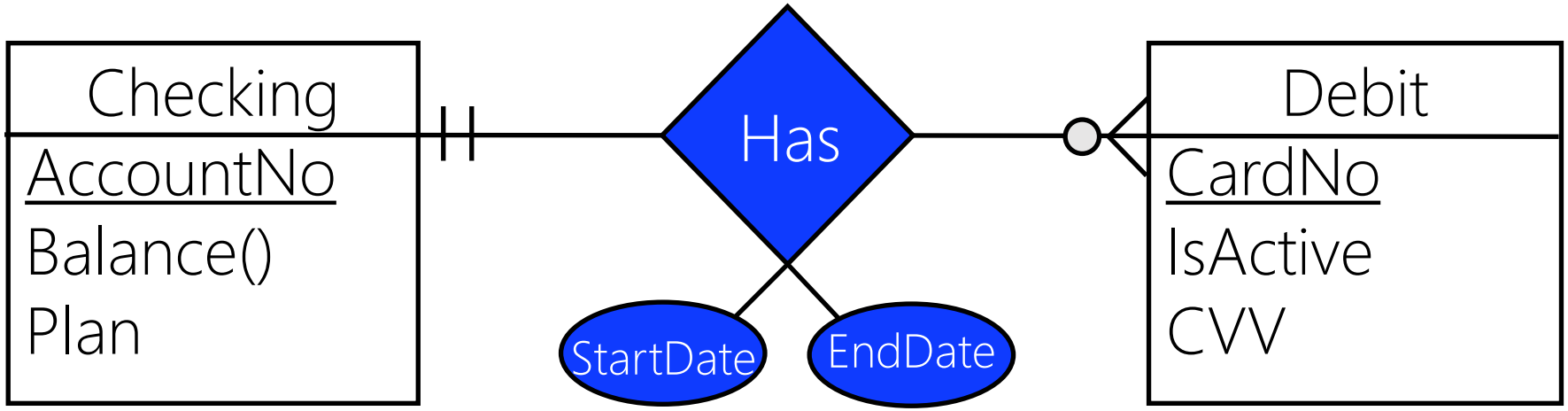
R2R × Many-One × Banking



R1: Checking(AccountNo, Balance, Plan)

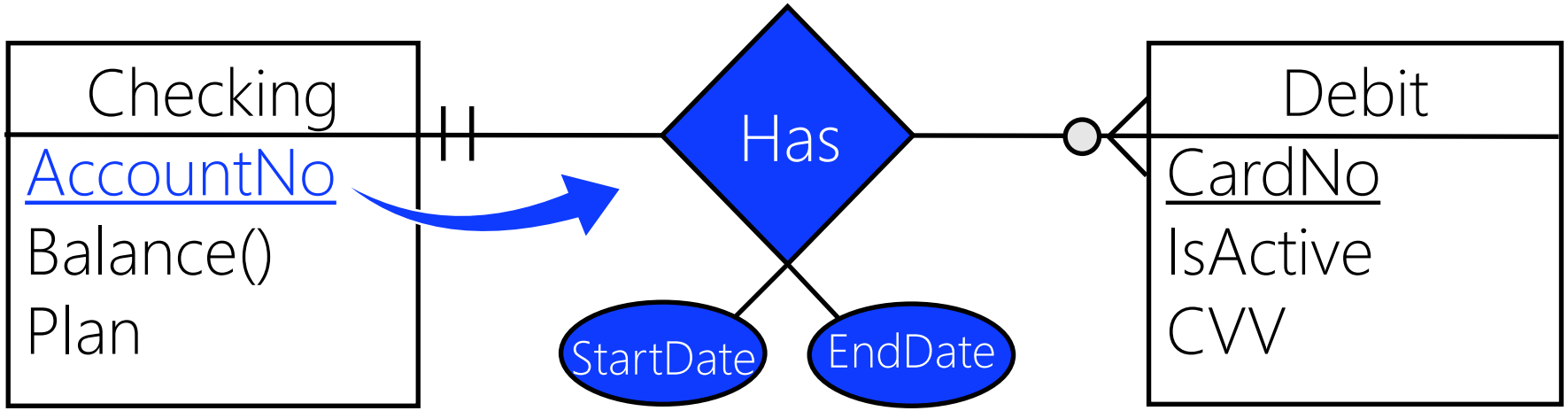
R2: Debit(CardNo, IsActive, CVV)

R2R × Many-One × Banking



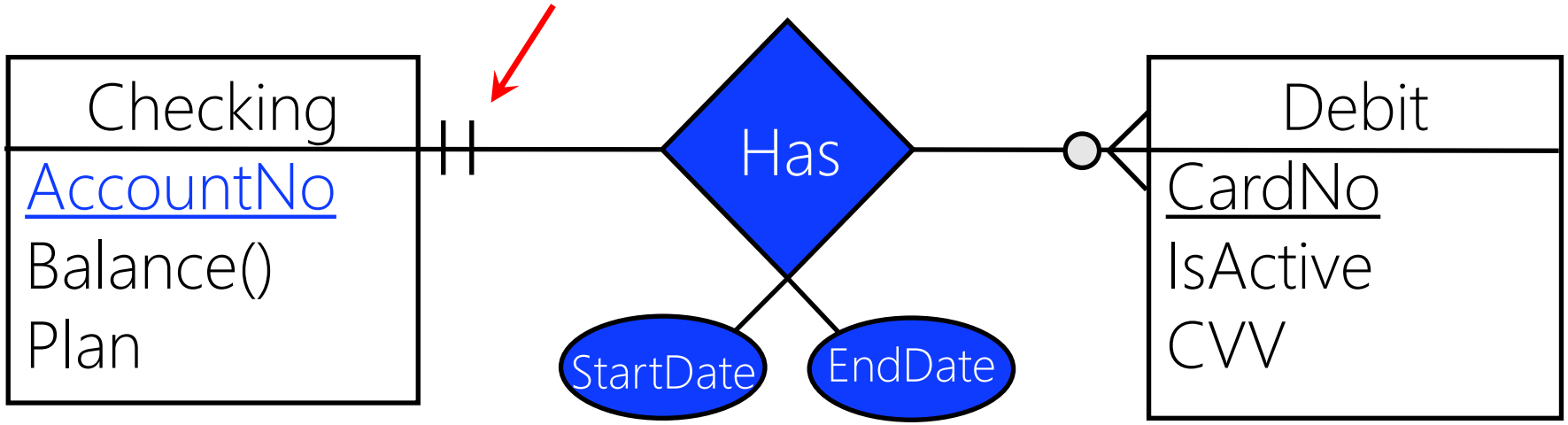
R1: Checking(AccountNo, Balance, Plan)
R2: Debit(CardNo, IsActive, CVV, StartDate, EndDate)

R2R × Many-One × Banking



R1: Checking(AccountNo, Balance, Plan)
R2: Debit(CardNo, IsActive, CVV, Checking.AccountNo, StartDate, EndDate)

R2R × Many-One × Banking

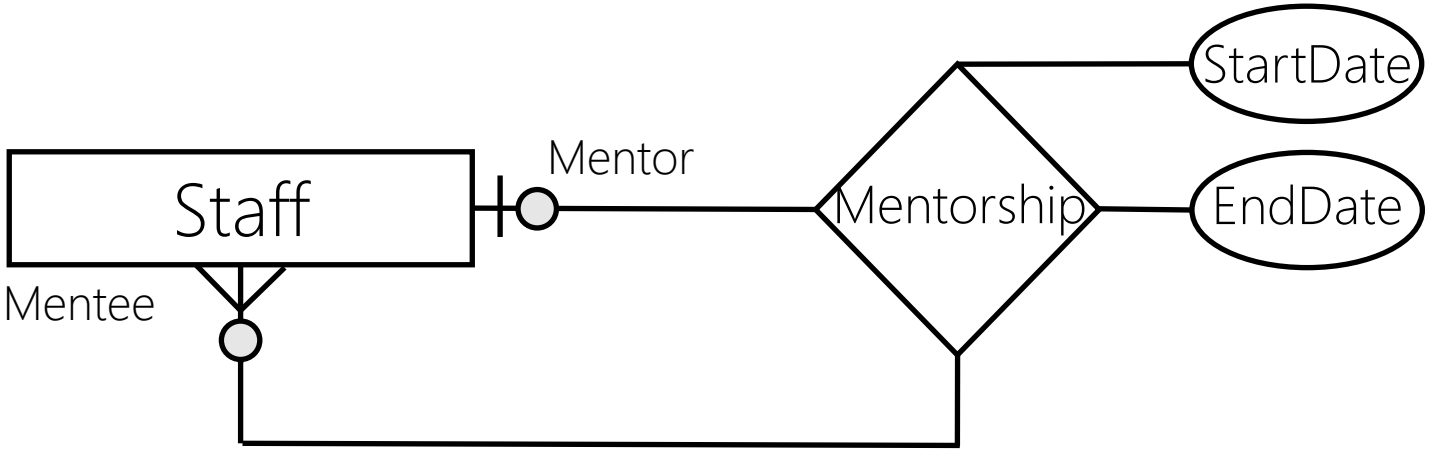


R1: Checking(AccountNo, Balance, Plan)

R2: Debit(CardNo, IsActive, CVV, Checking.AccountNo, StartDate, EndDate)

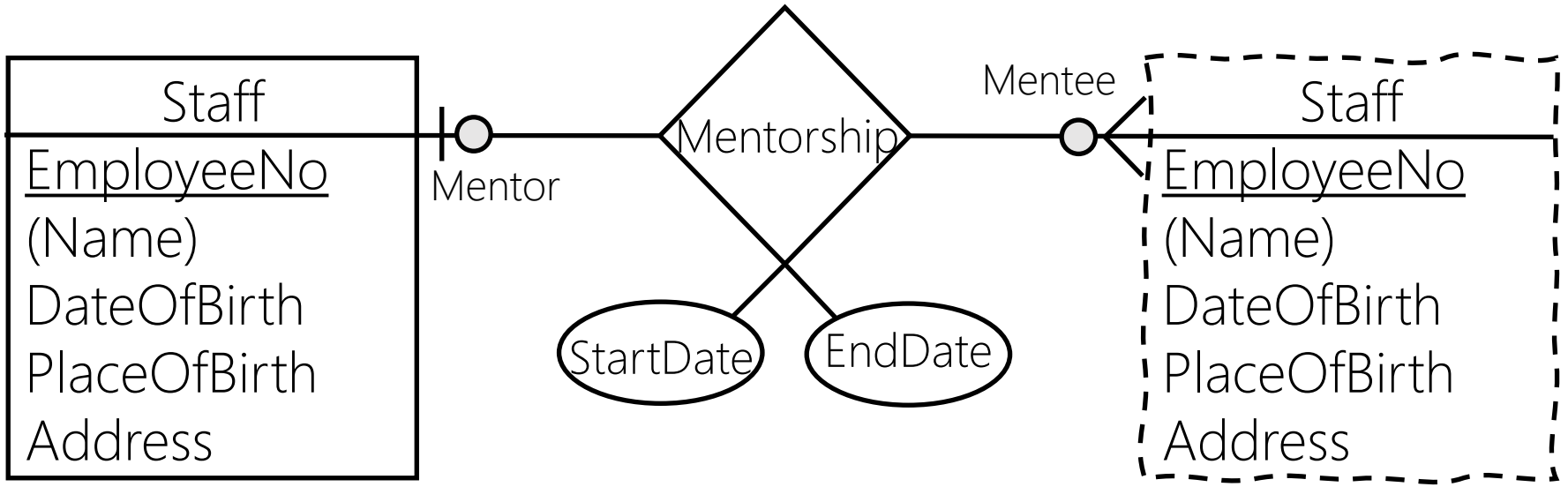
Must be mandatory! (Why?)

R2R × Many-One × Self (Unary)

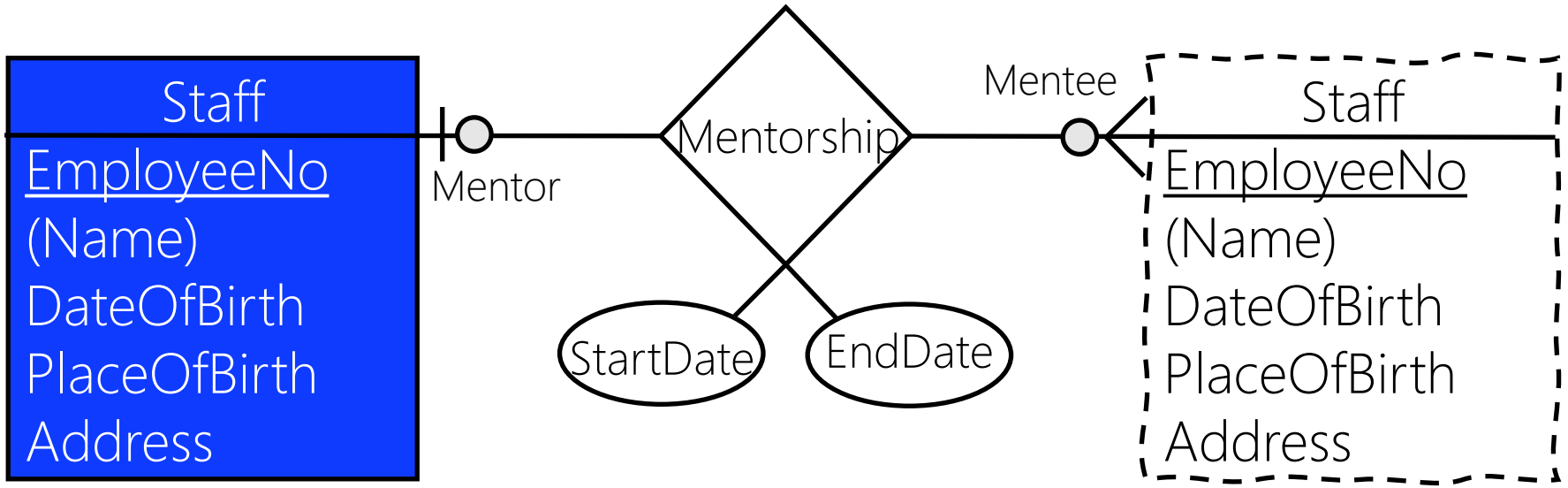


How many relations?

R2R × Many-One × Self (Unary)

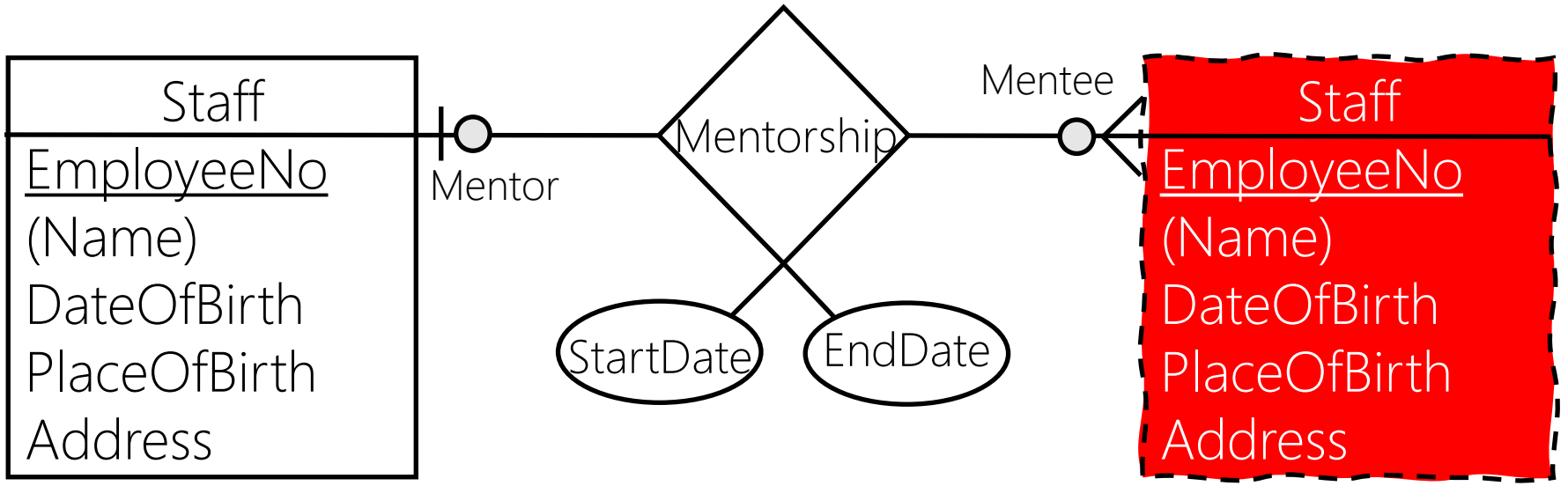


R2R × Many-One × Self (Unary)



R1: Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth)

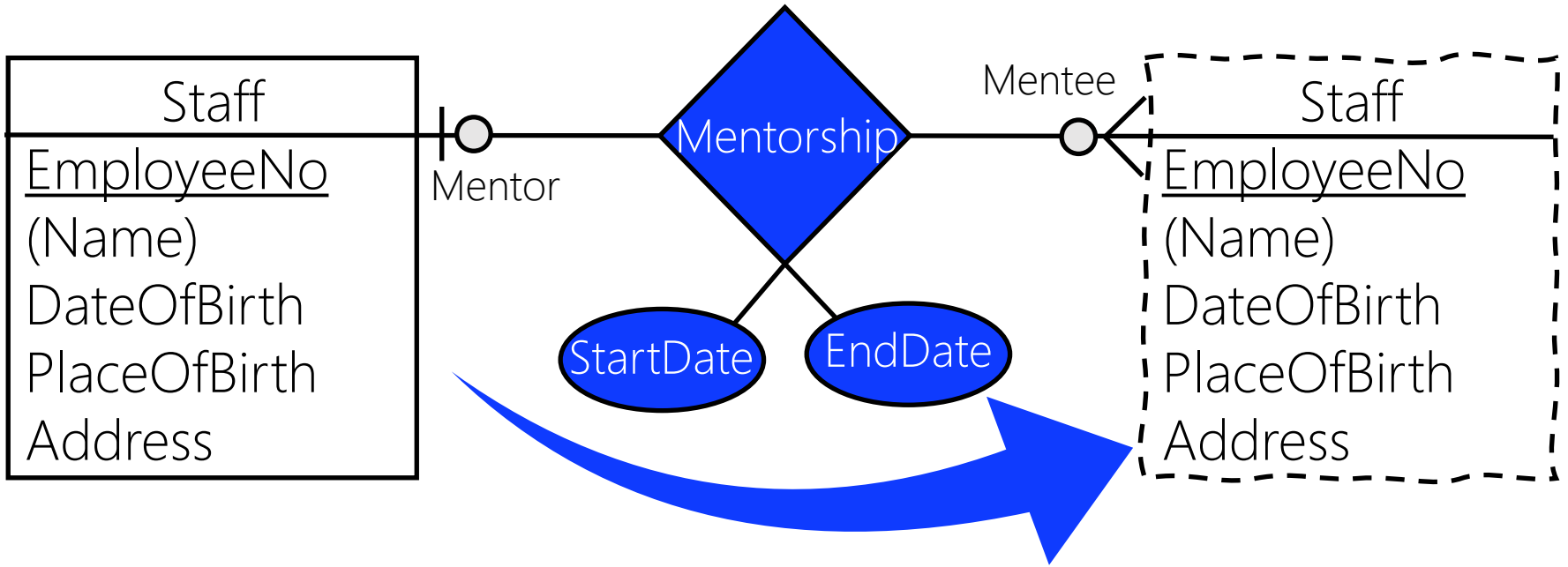
R2R × Many-One × Self (Unary)



R1: Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth)

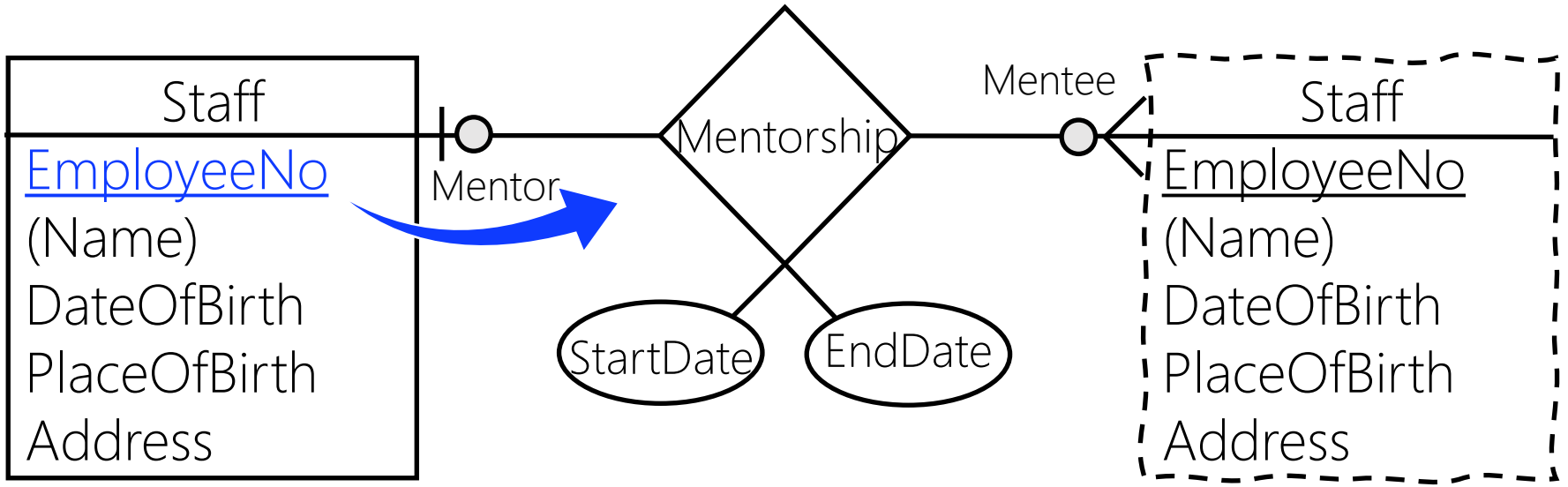
~~R2: Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth)~~

R2R × Many-One × Self (Unary)



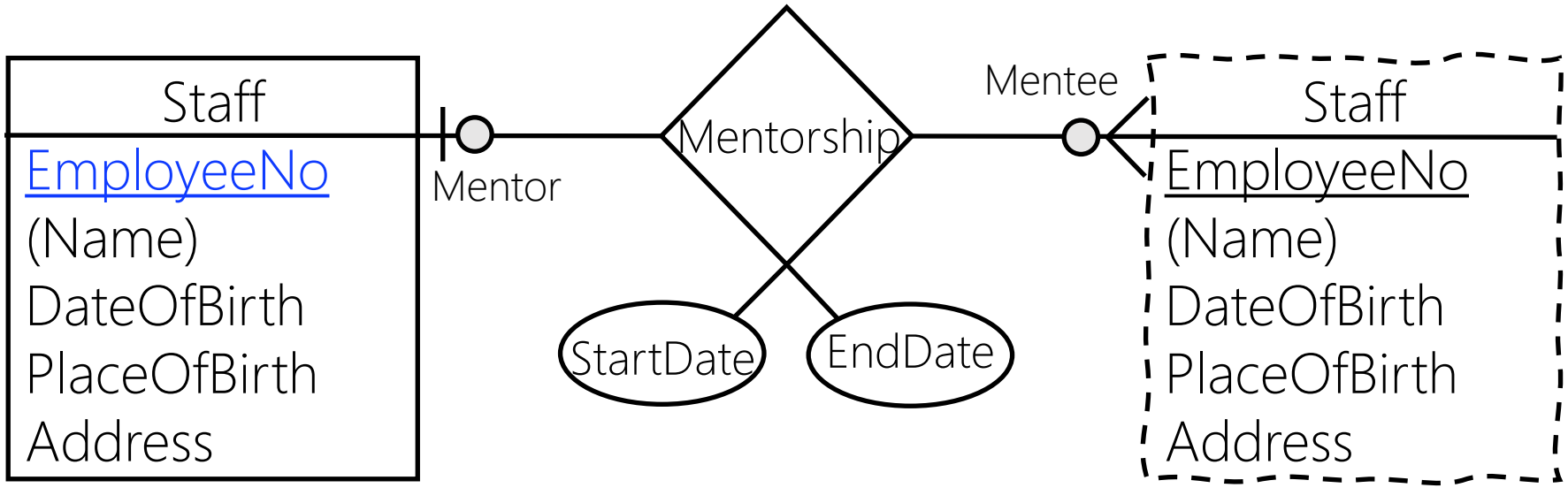
R1: Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth, StartDate, EndDate)

R2R × Many-One × Self (Unary)



R1: Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth, EmployeeNo, StartDate, EndDate)

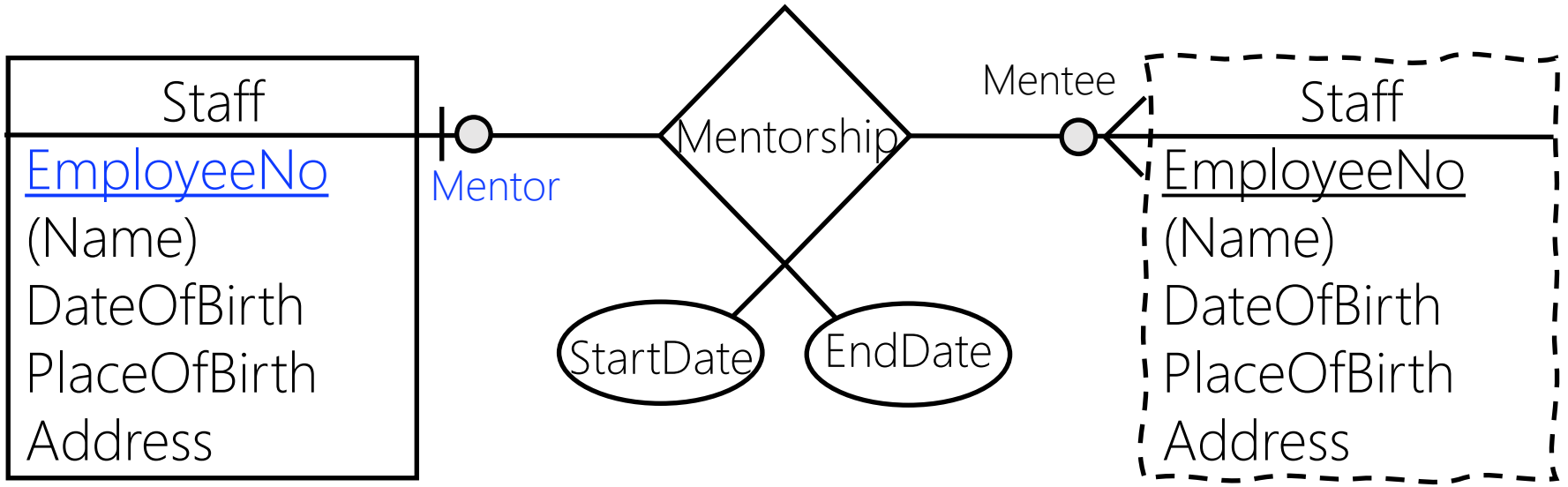
R2R × Many-One × Self (Unary)



Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth, EmployeeNo, StartDate, EndDate)

Attribute name conflict!

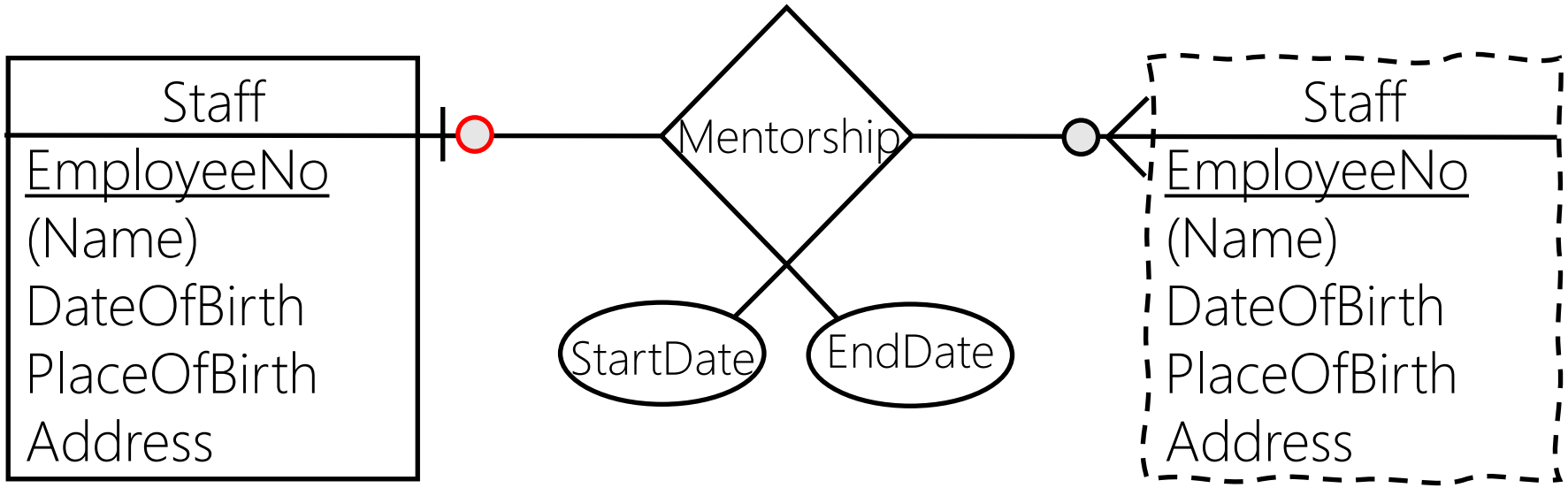
R2R × Many-One × Self (Unary)



Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth, MentorNo, StartDate, EndDate)

By convention, change attribute name to the role

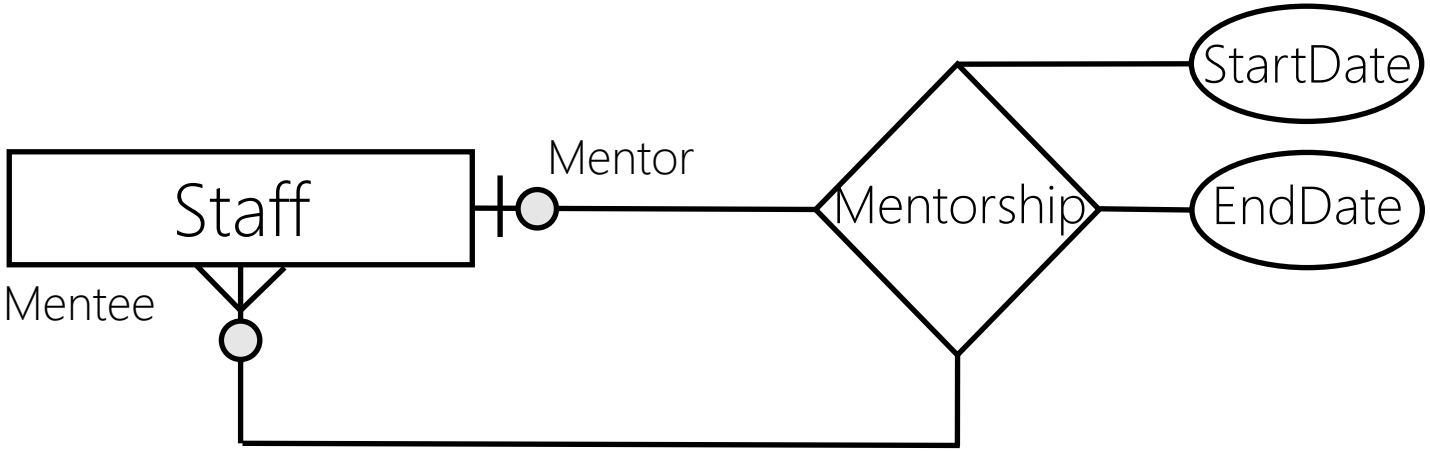
R2R × Many-One × Self (Unary)



Staff(EmployeeNo, Name, DateOfBirth, PlaceOfBirth, MentorNo, StartDate, EndDate)

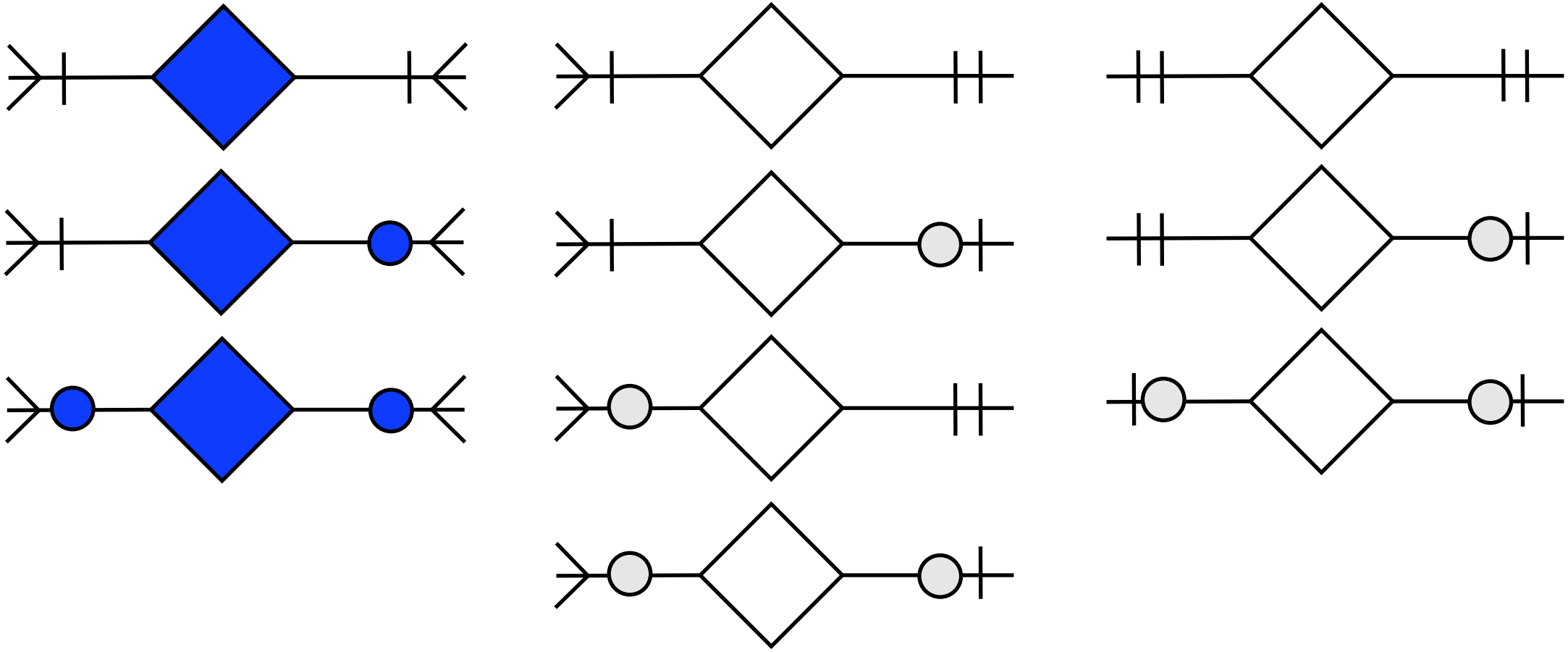
Must be optional!

R2R × Many-One × Self (Unary)

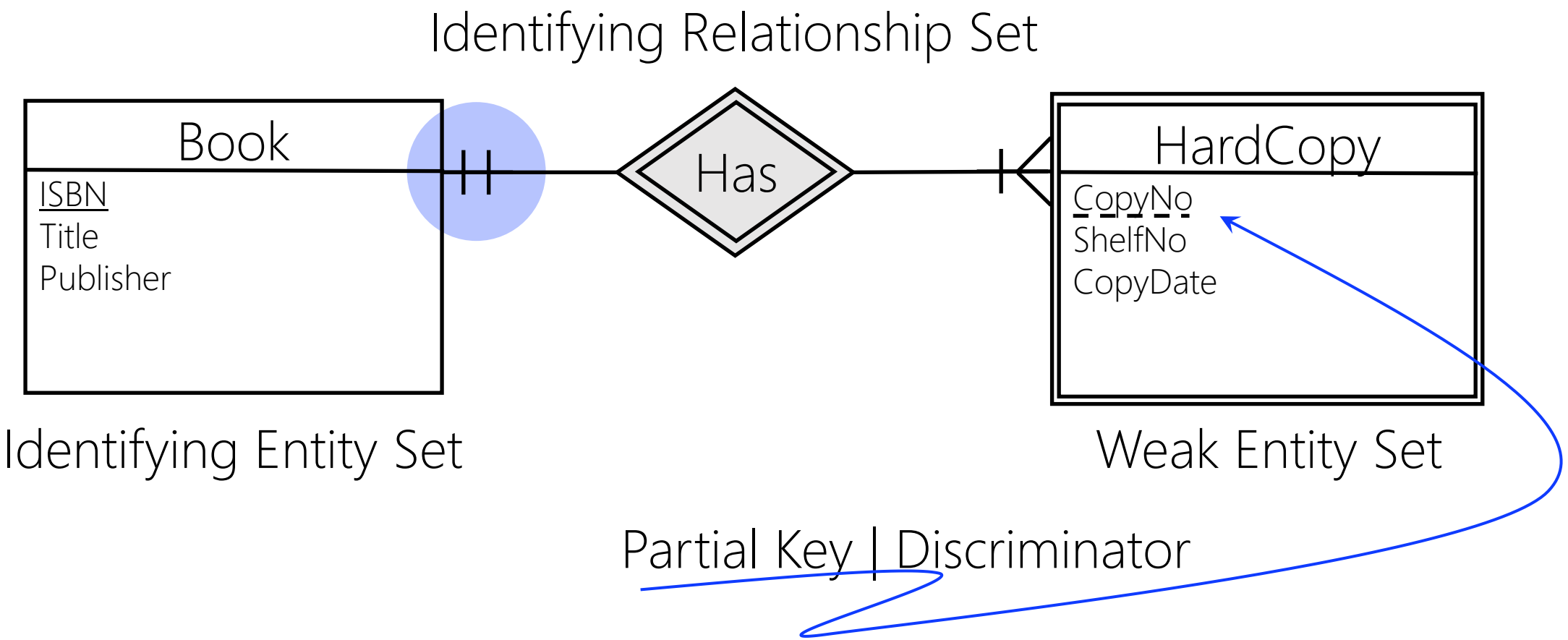


Staff(<u>EmployeeNo</u>, Name, DateOfBirth, PlaceOfBirth, MentorNo, StartDate, EndDate)

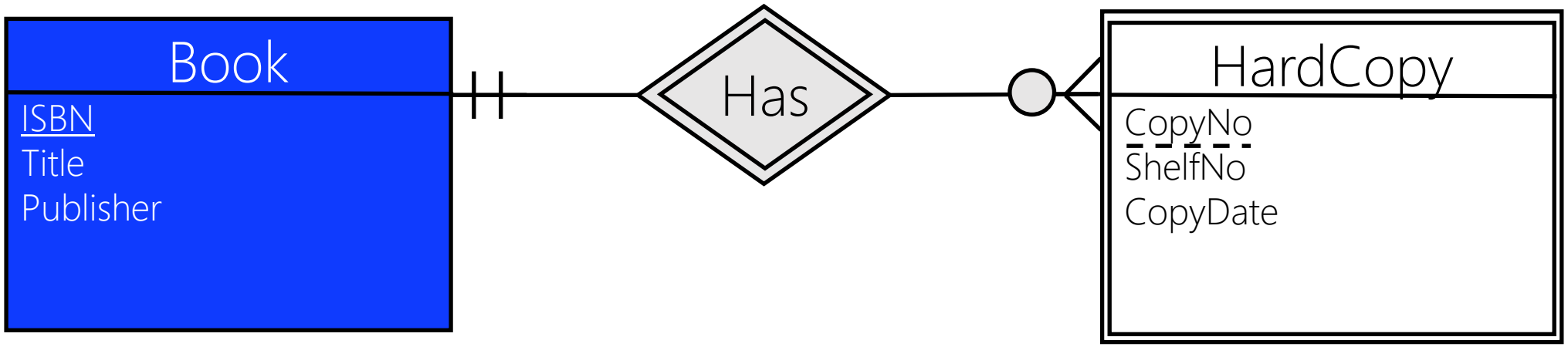
Relationship2Relation (R2R)



R2R × Many-One × Weak Entity Set

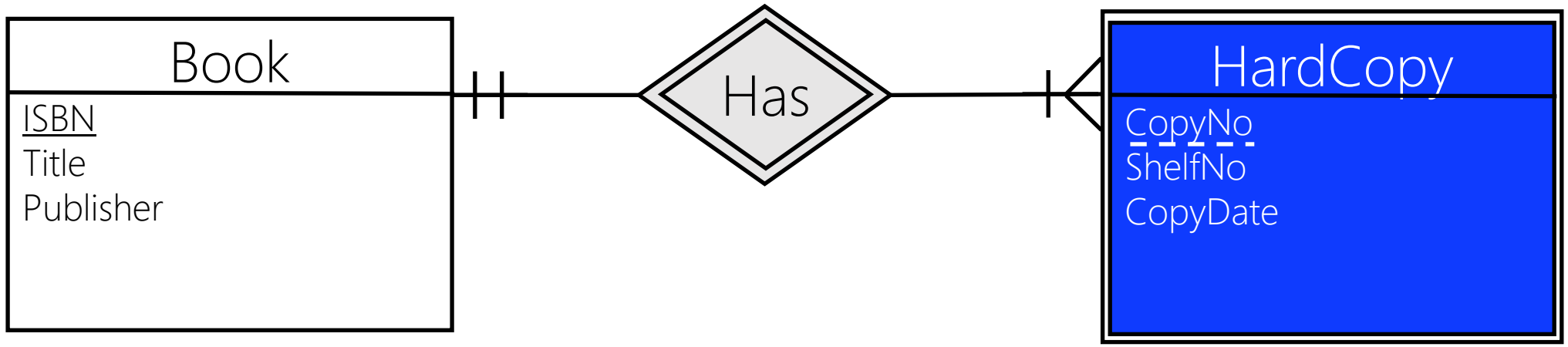


R2R × Many-One × Weak Entity Set



R1: Book(ISBN, Title, Publisher)

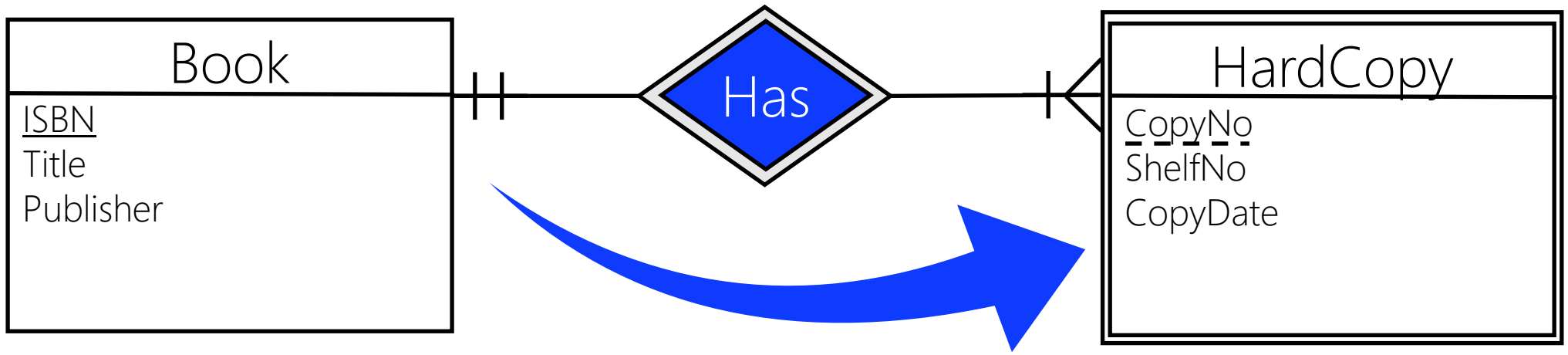
R2R × Many-One × Weak Entity Set



R1: Book(ISBN, Title, Publisher)

R2: HardCopy(CopyNo, ShelfNo, CopyDate)

R2R × Many-One × Weak Entity Set

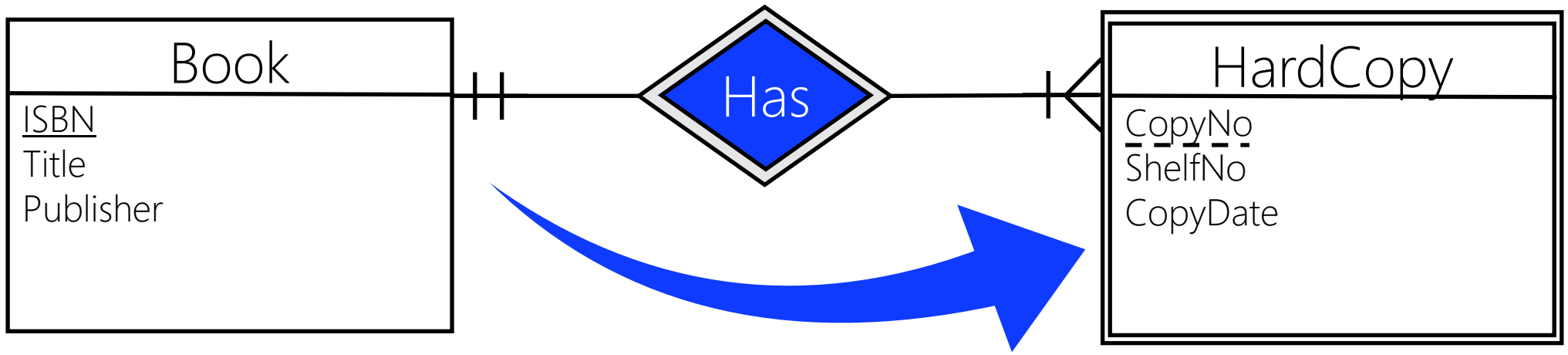


R1: Book(ISBN, Title, Publisher)

R2: HardCopy(CopyNo, ShelfNo, CopyDate, Book.ISBN)

Foreign Key

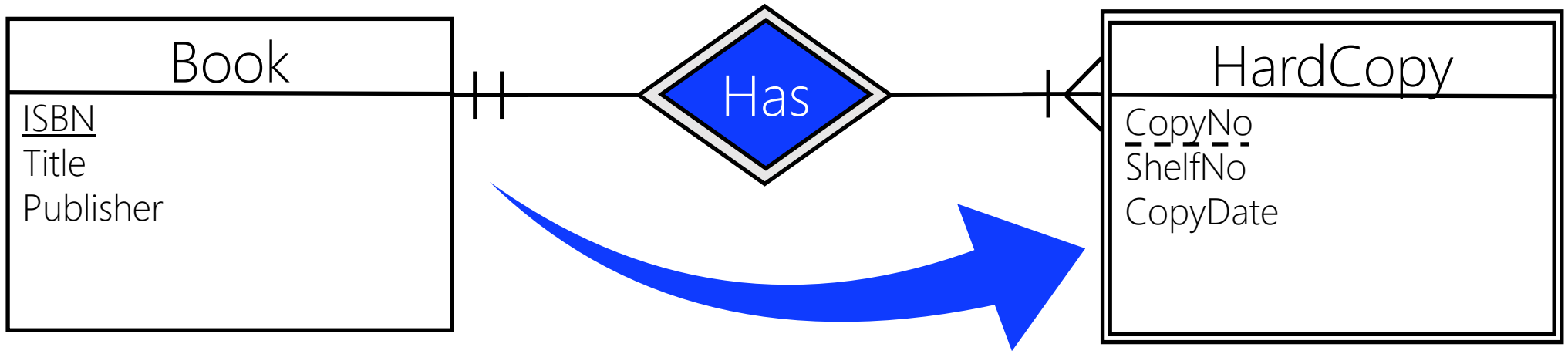
R2R × Many-One × Weak Entity Set



R1: Book(ISBN, Title, Publisher)
R2: HardCopy(CopyNo, ShelfNo, CopyDate, Book.ISBN)

Primary Key = {Partial Keyset} U {Primary Keyset of identifying entity set}

R2R × Many-One × Weak Entity Set

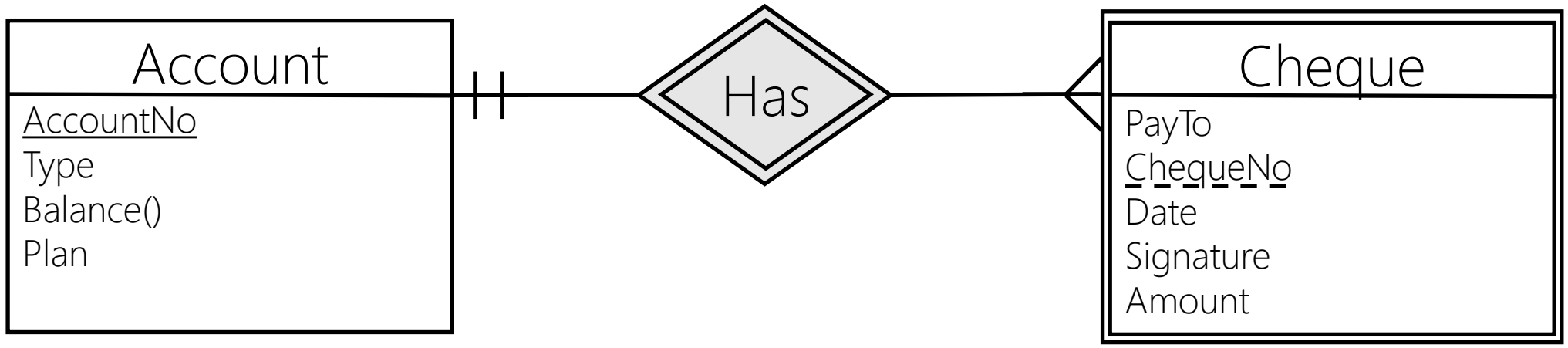


R1: Book(ISBN, Title, Publisher)

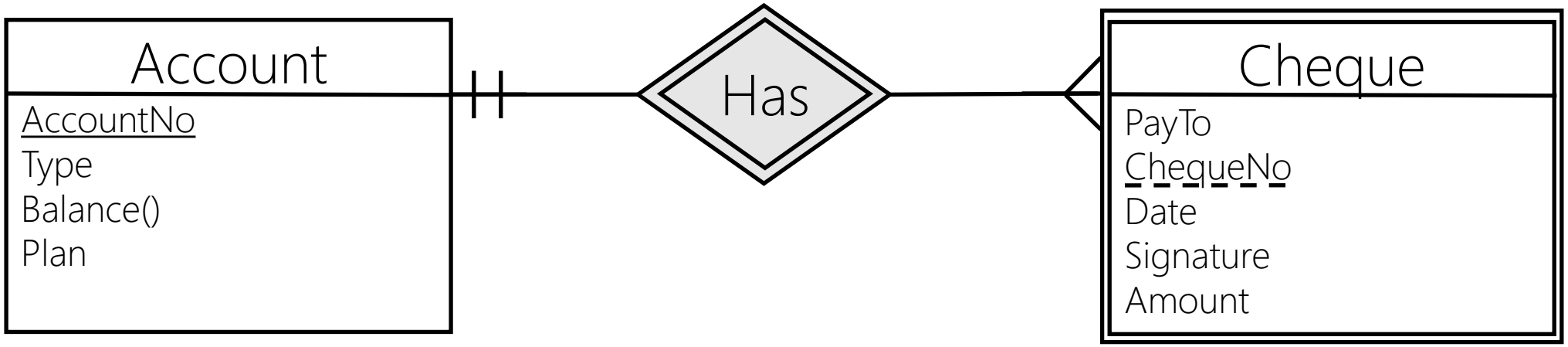
R2: HardCopy(Book.ISBN, CopyNo, ShelfNo, CopyDate,)

Already mandatory. (Why?)

R2R × Many-One × Weak Entity Set



R2R × Many-One × Weak Entity Set



R1: Account(AccountNo, Type, Balance, Plan)
R2: Cheque(AccountNo, ChequeNo, Date, Signature, PayTo, Amount)

Relationship2Relation (R2R)