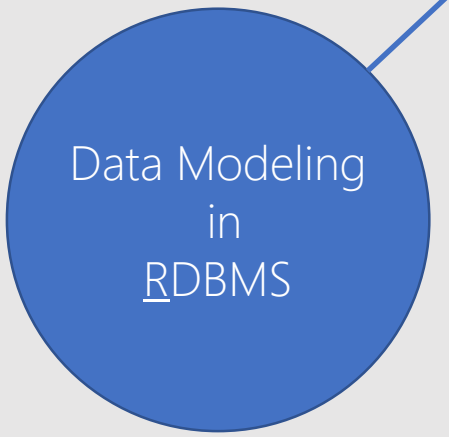
Today



Real World Entity

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

| Logical Level | Relational Model

| Physical Level | SQL

Computable Entity

Data Modeling × Logical Level 3

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

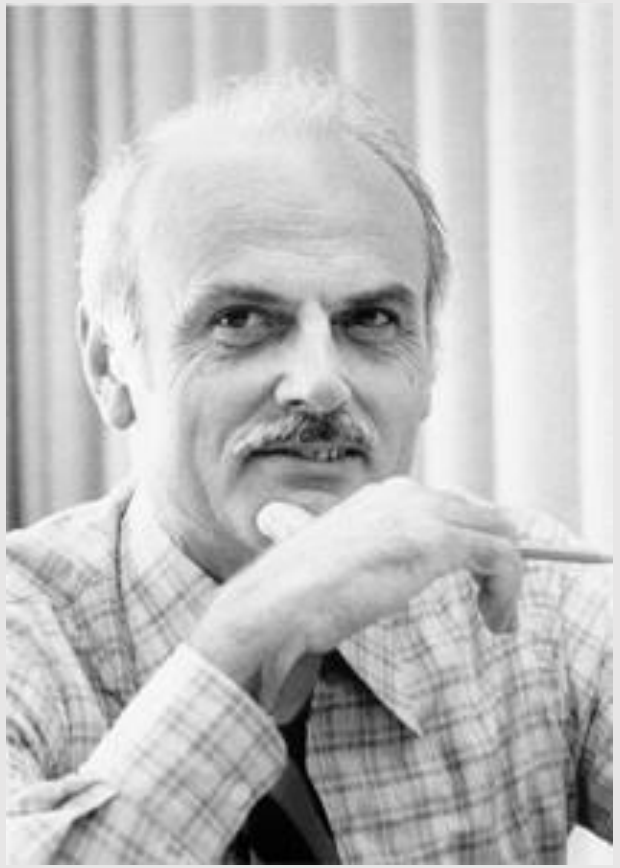
Data Modeling × Logical Level

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

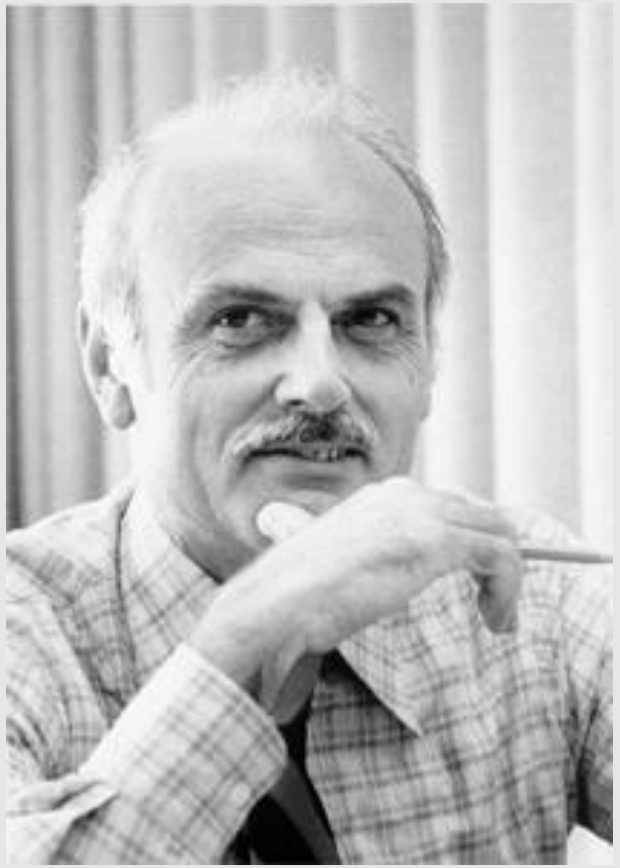
E. F. Codd
IBM Research Laboratory, San Jose, California

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

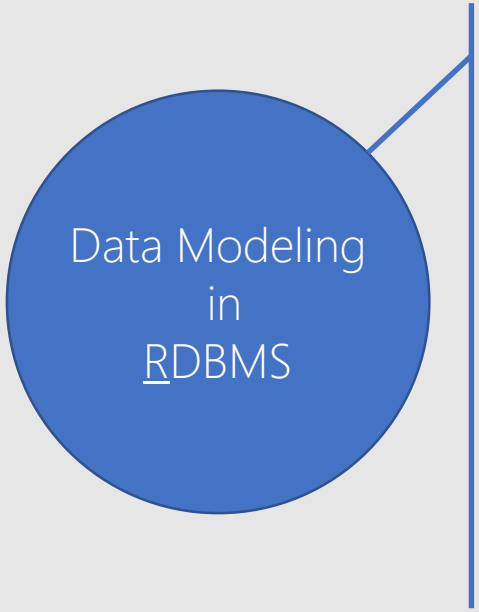
Schema	Title	Language	RunningTime
Tuple 1	*2001: A Space Odyssey*	*English*	*142*
Tuple 2	*Rosemary's Baby*	*English*	*136*
Tuple 3	*The Birds*	*English*	*119*
…	…	…	…

Informally: Relation | Table
Tuple | Row
Attribute | Column

Relational × Relation (R)

Everything is Relation (Table)

Entity & Relationship → Relation



Real World Entity

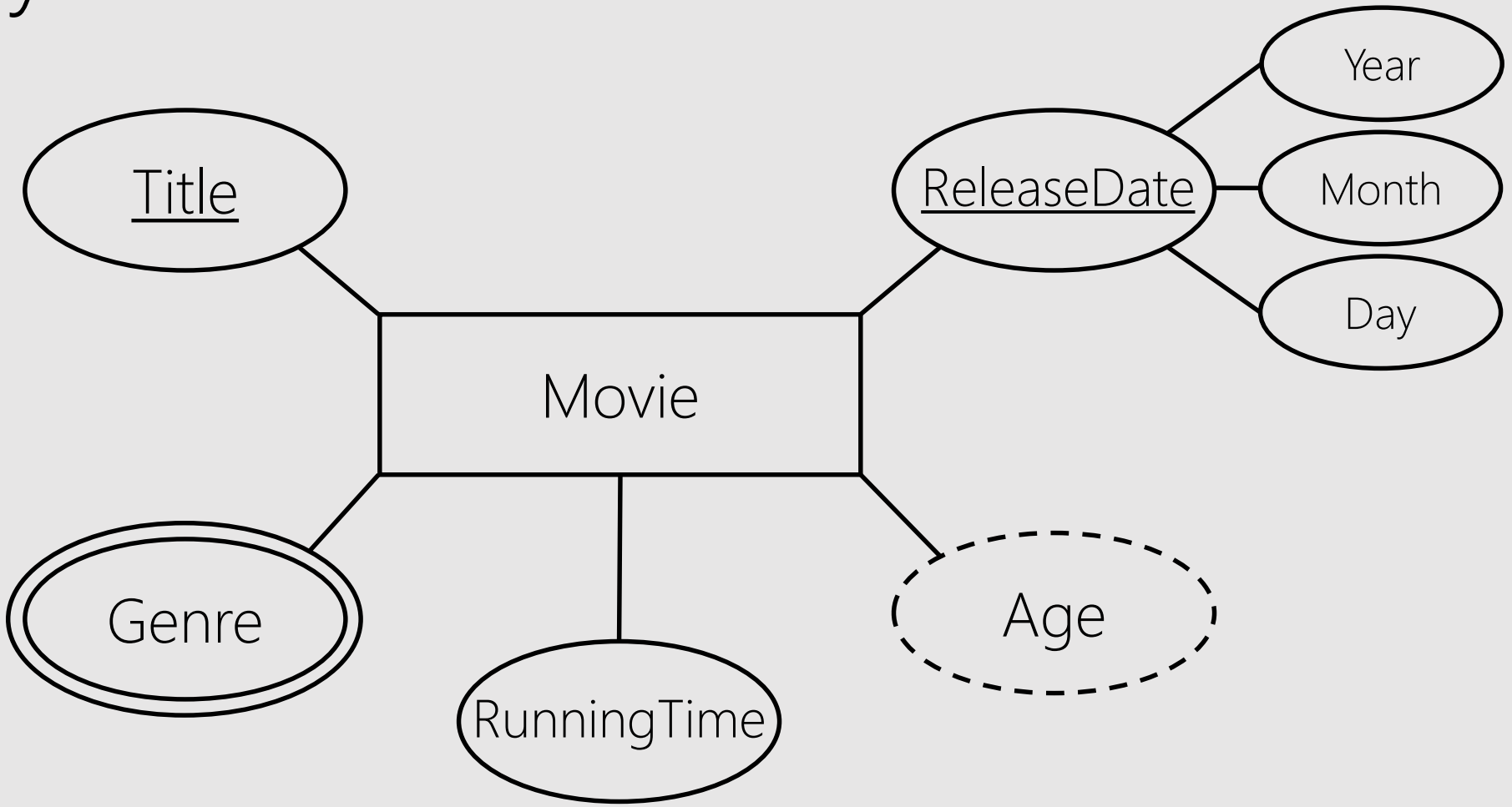
Conceptual Level | Entity & Relationship

| Logical Level | Relation (Table)

| Physical Level | SQL

Computable Entity

Entity2Relation



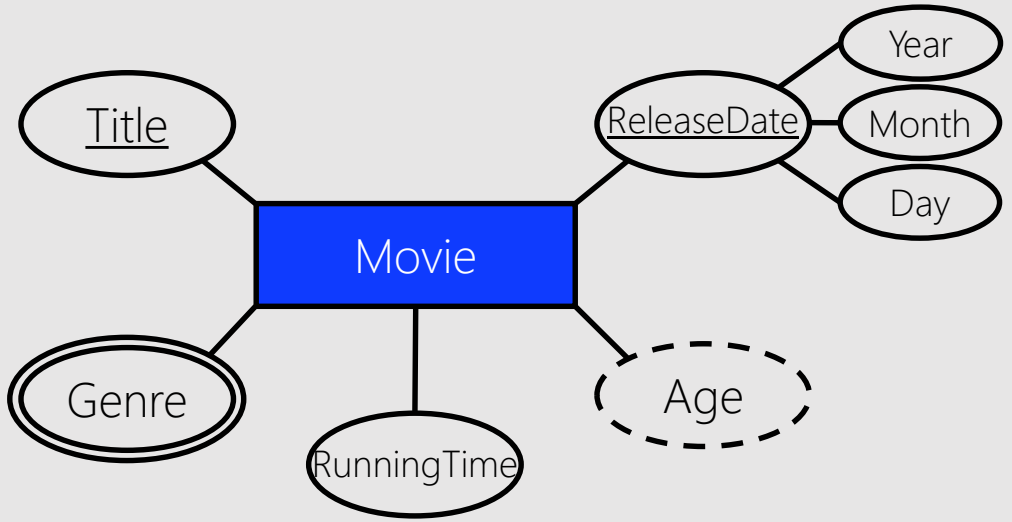
Entity2Relation (E2R)

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

R_2: Genre(<u>Title</u>)

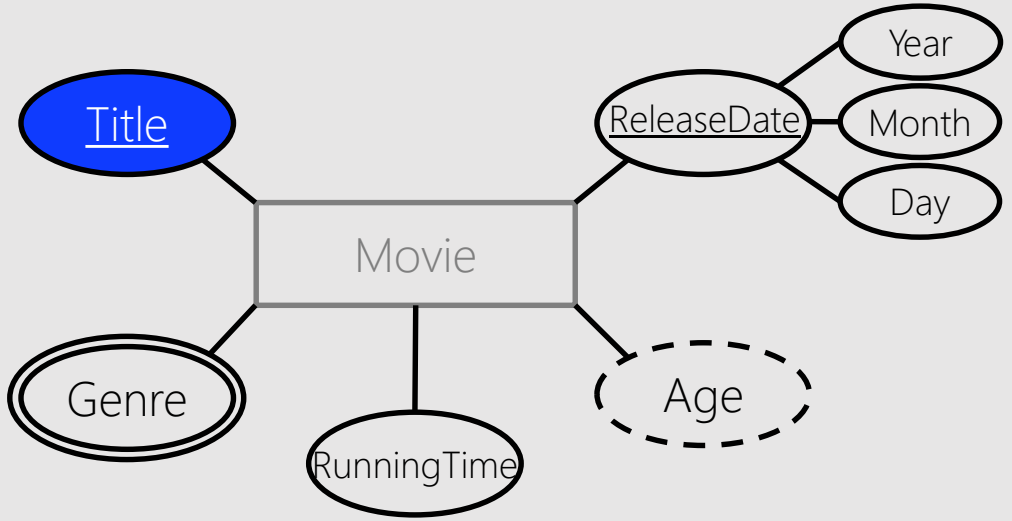
R_3: MovieGenre(Movie.<u>Title</u>, Movie.<u>ReleaseYear</u>, Movie.<u>ReleaseMonth</u>, Movie.<u>ReleaseDay</u>, Genre.<u>Title</u>)

E2R × Entity Set



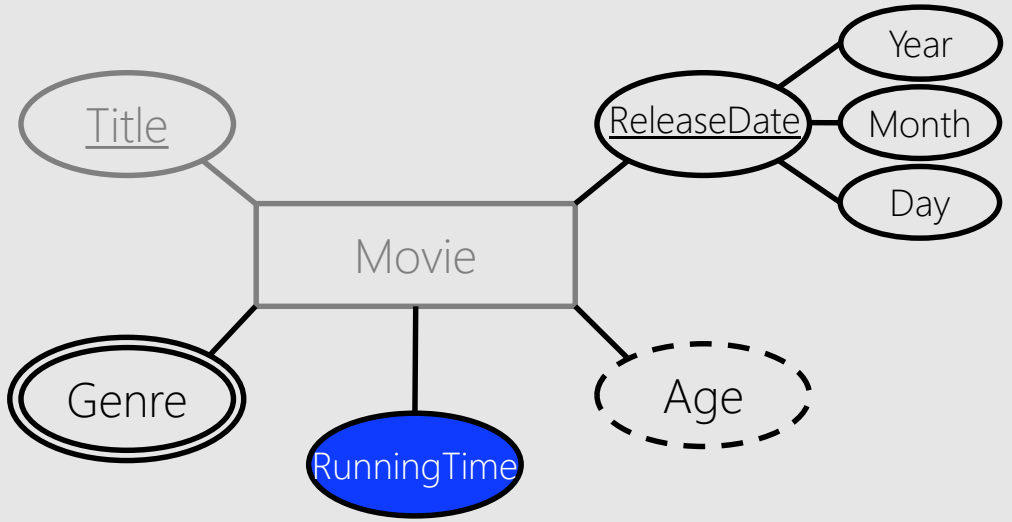
R_1: Movie()

E2R × Entity Set × Attribute



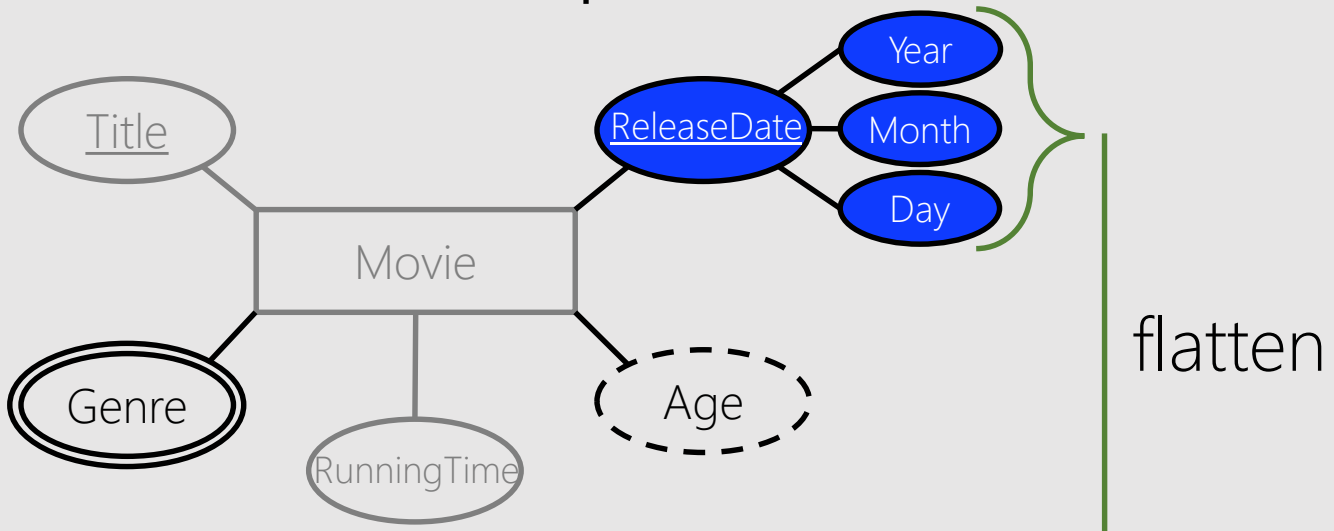
R_1: Movie(Title)

E2R × Entity Set × Attribute



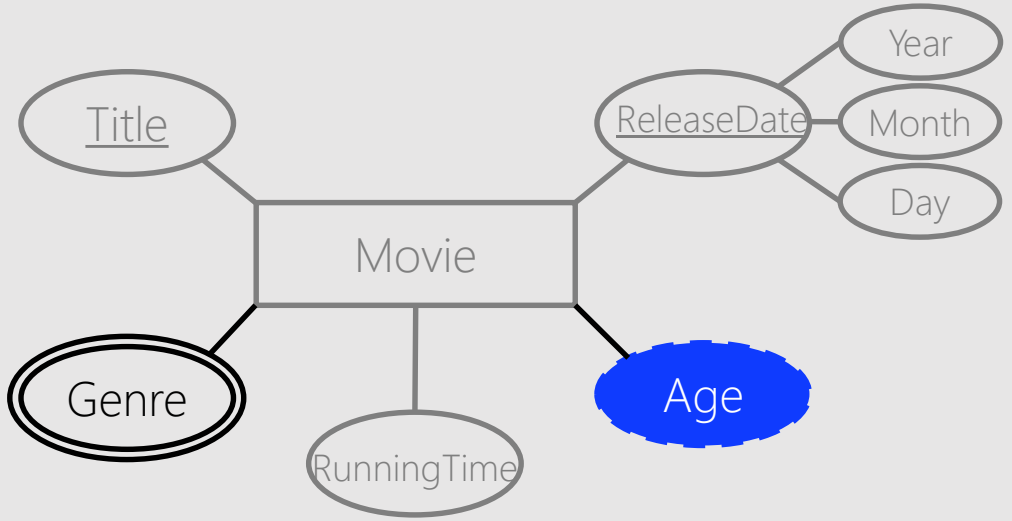
R_1: Movie(Title, RunningTime)

E2R × Entity Set × Composite Attribute



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

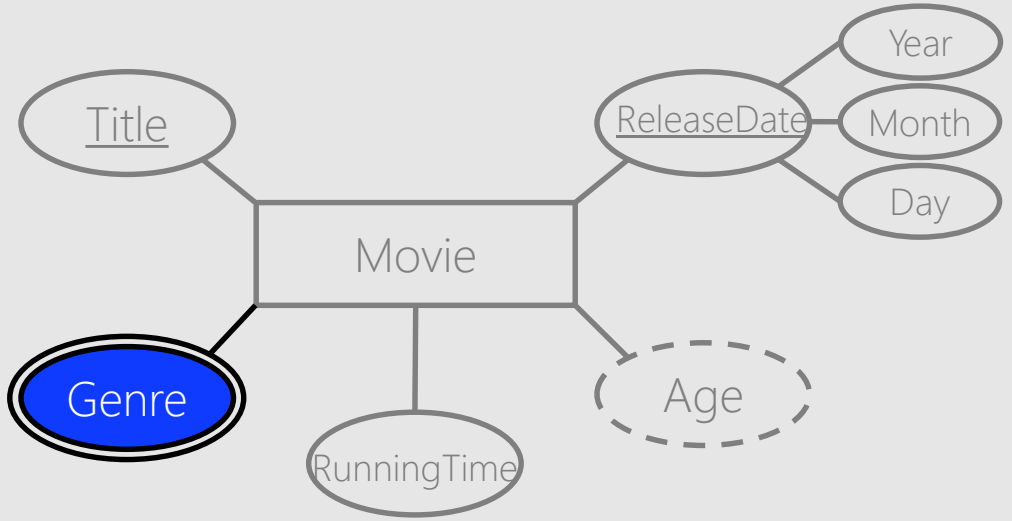
E2R × Entity Set × Derived Attribute



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

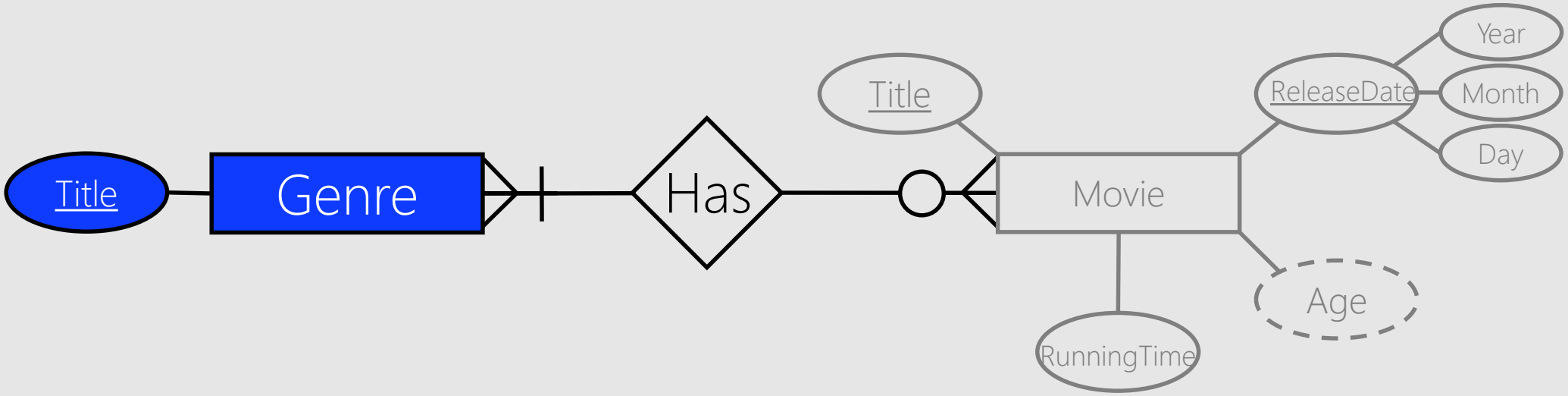
E2R × Entity Set × Multivalued Attribute 17

Relational model does not allow multivalued attributes!



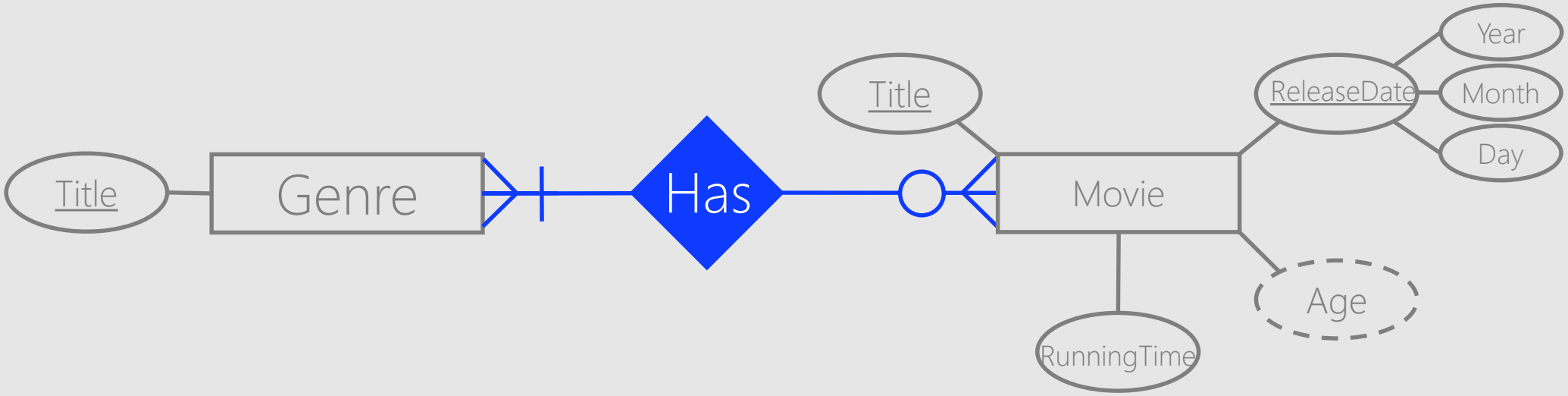
R_1: Movie(Title, RunningTime, ReleaseYear, ReleaseMonth, ReleaseDay, Age)

E2R × Entity Set × Multivalued Attribute 18



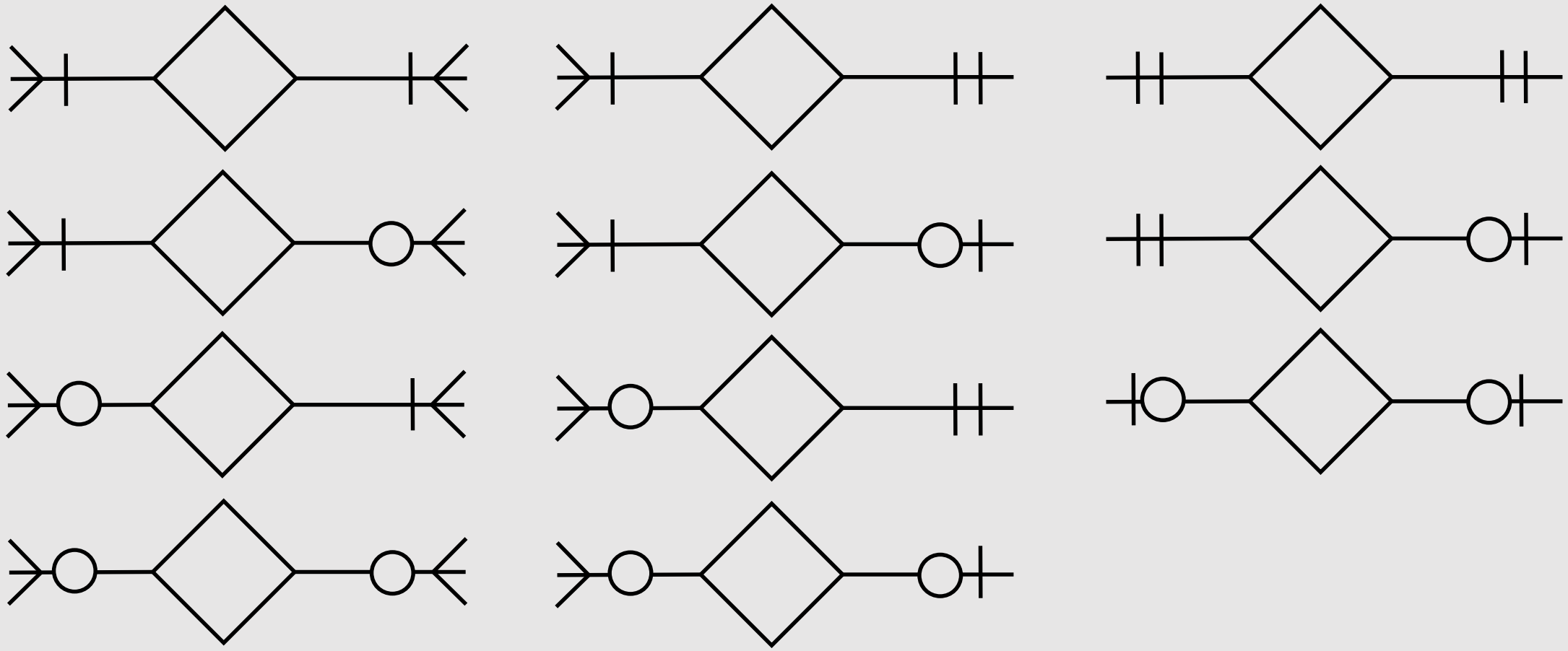
R_1: Movie(Title, RunningTime, ReleaseYear, ReleaseMonth, ReleaseDay, Age)
R_2: Genre(Title)

E2R × Entity Set × Multivalued Attribute 19

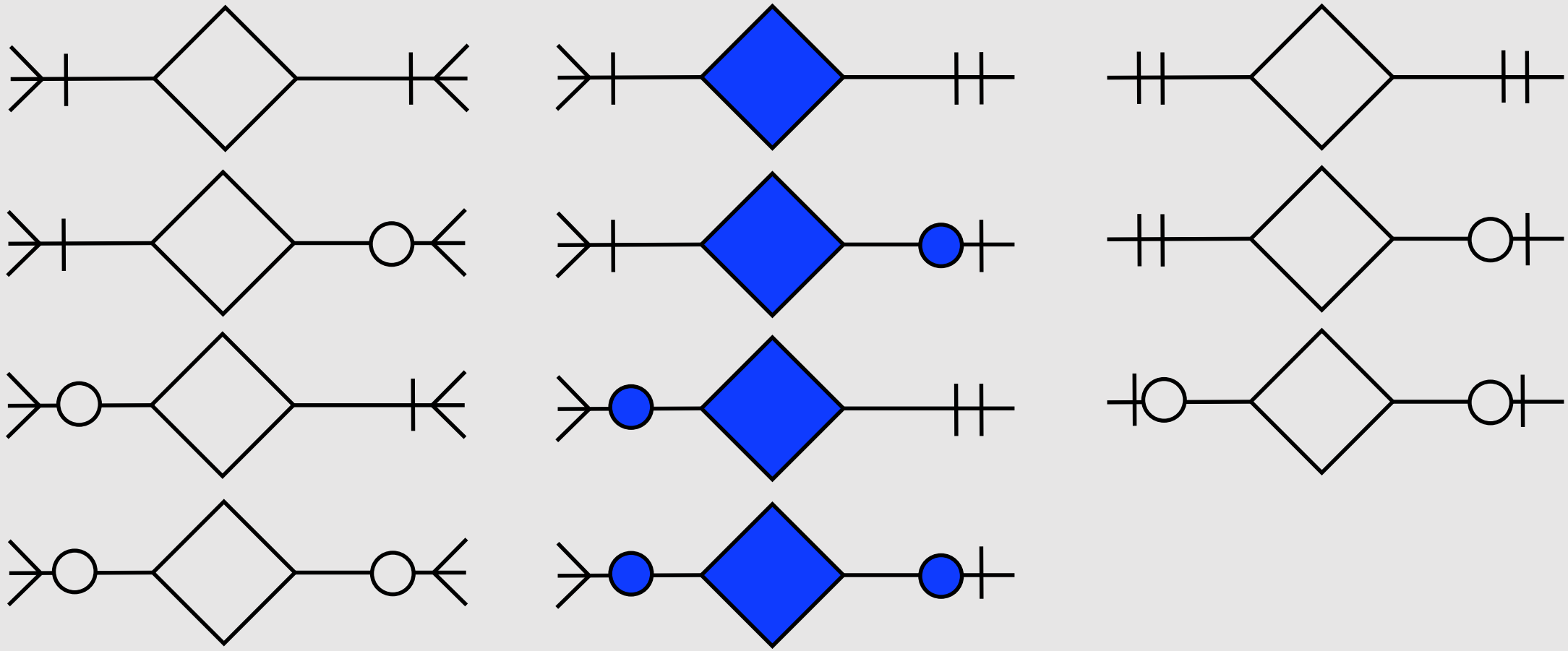


R_1: Movie(<u>Title</u>, RunningTime, <u>ReleaseYear</u>, <u>ReleaseMonth</u>, <u>ReleaseDay</u>, Age)
R_2: Genre(<u>Title</u>)

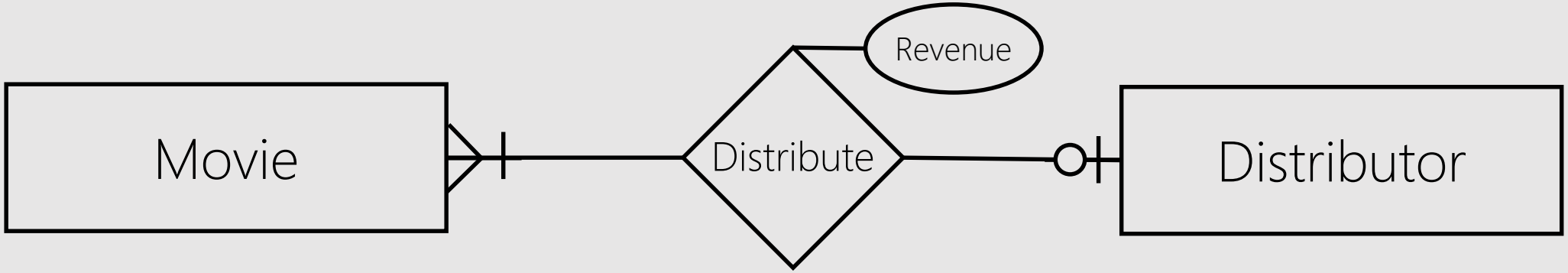
Relationship2Relation (R2R)



Relationship2Relation (R2R)

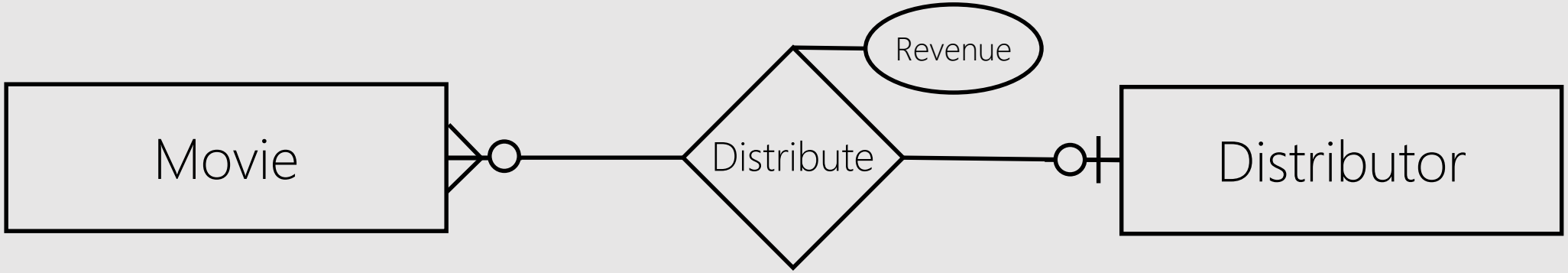


R2R × Many-One (One-Many)



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	
m_5	m_5, d_1, \$13M	

R2R × Many-One (One-Many)



m_1

m_2

m_3

m_4

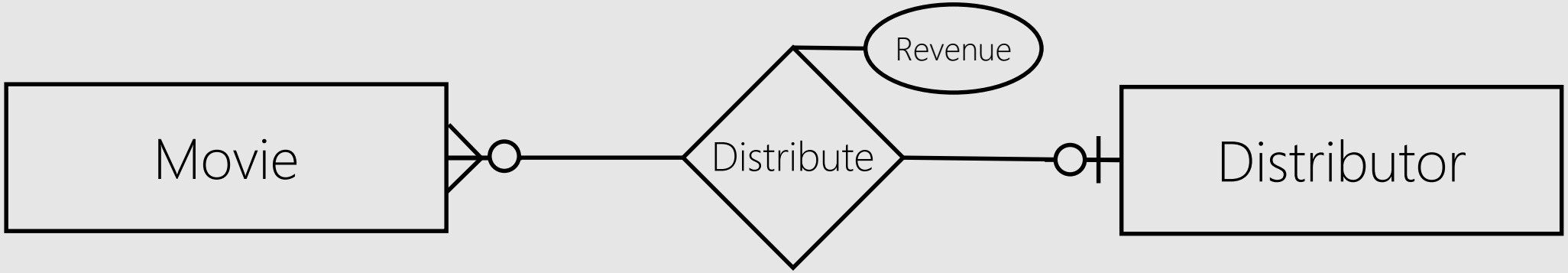
m_5

d_1, m_5, \$13M

d_2, m_4, \$27M

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

R2R × Many-One (One-Many)



NULL, NULL, m_1

d_3, \$20M, m_2

d_3, \$35M, m_3

d_2, \$27M, m_4

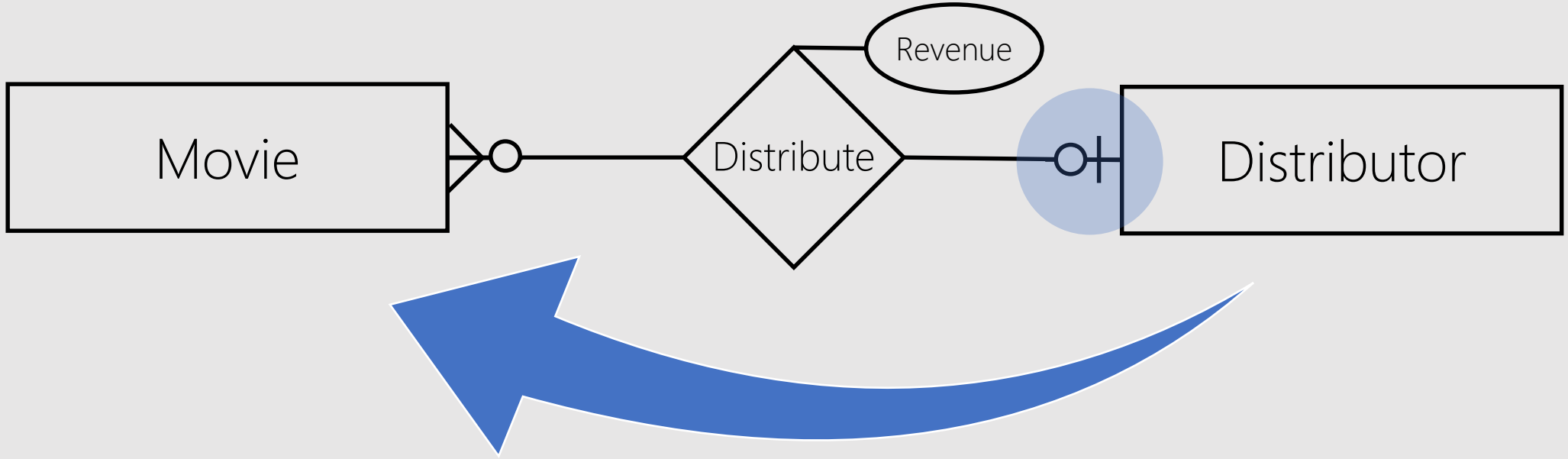
d_1, \$13M, m_5

d_1

d_2

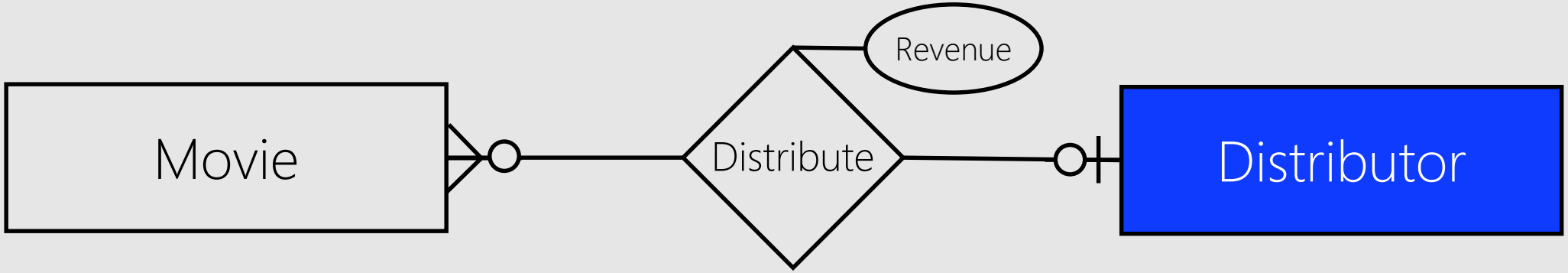
d_3

R2R × Many-One (One-Many)



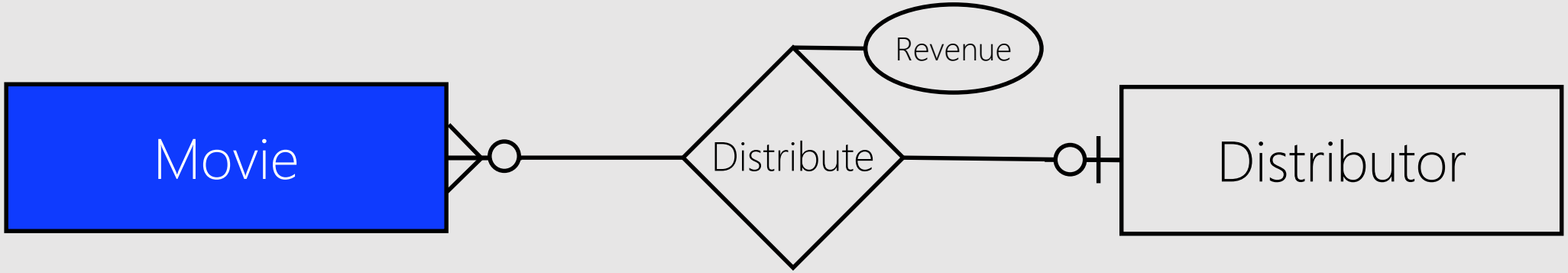
Everything goes to entity set with cardinality <u>one</u> (i.e., many side)
Because it only needs to store one entity from other entity set

R2R × Many-One (One-Many)



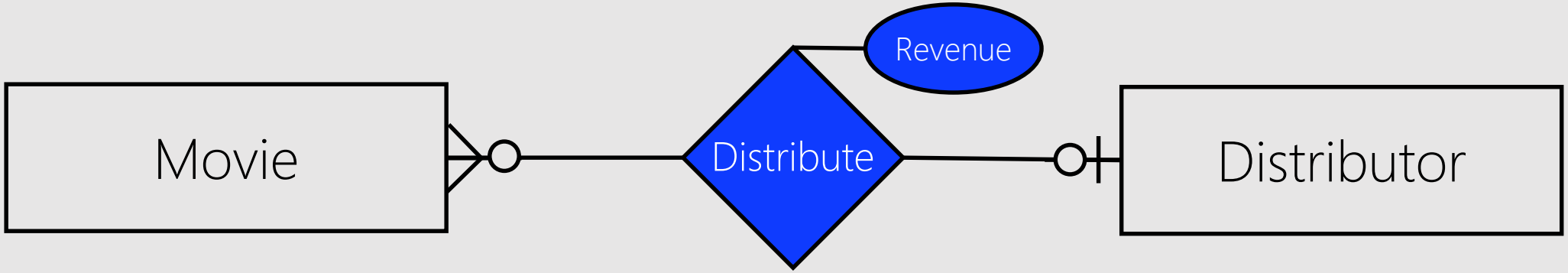
R_1: Distributor(<u>Name</u>, Address, POBox, Website, …)

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

R2R × Many-One (One-Many)

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

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

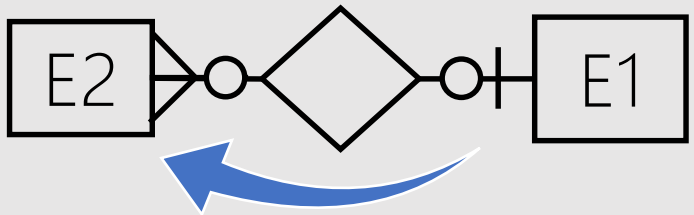
R2R × Many-One (One-Many)

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

must be optional (Why?)

R2R × Many-One (One-Many)

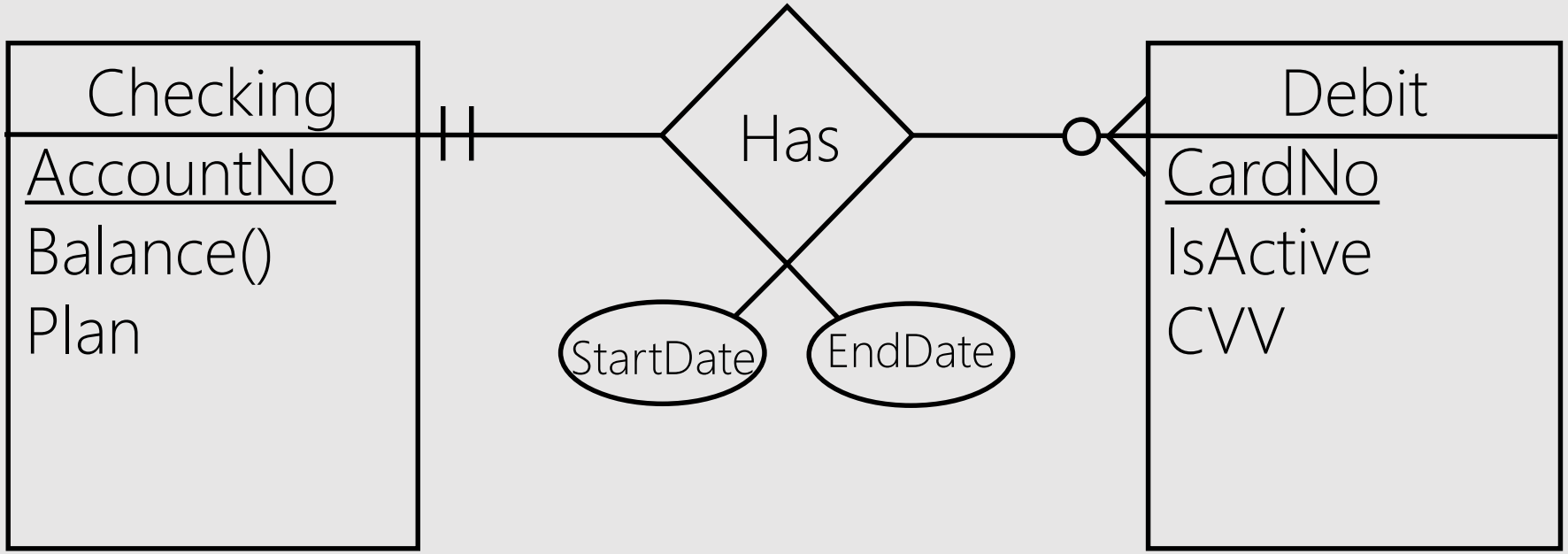
Input: Many-One relationship btw. E2 and E1, i.e.,
Output: Relations R1 for E1 and R2 for E2.



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
   else make foreign key set mandatory

Here, we do not care about E1's ordinality! Later we fix it.

R2R × Many-One × Banking

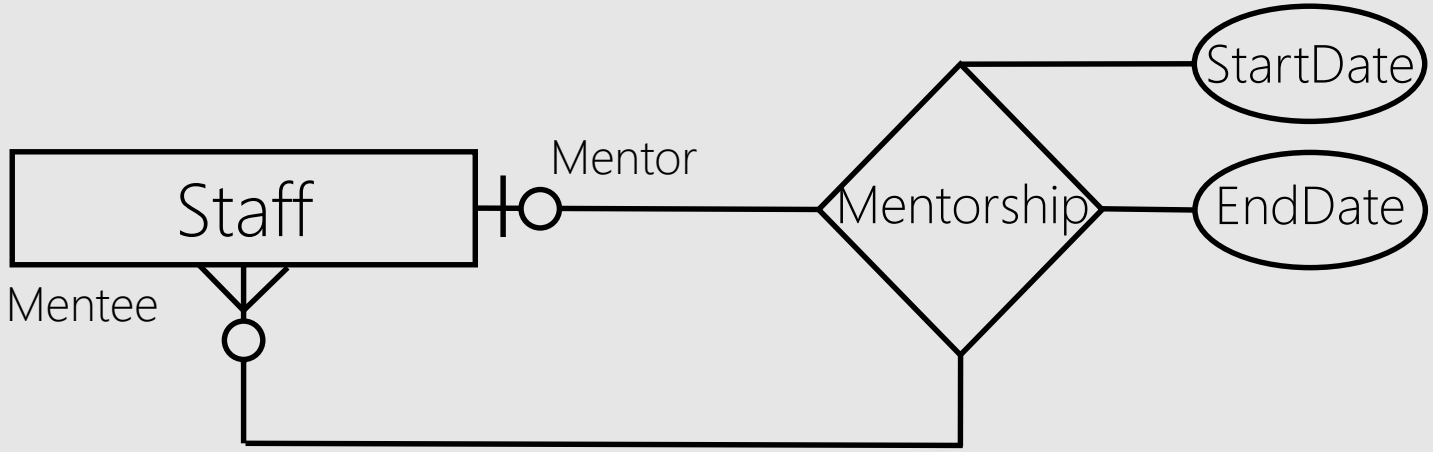


R2R × Many-One × Banking

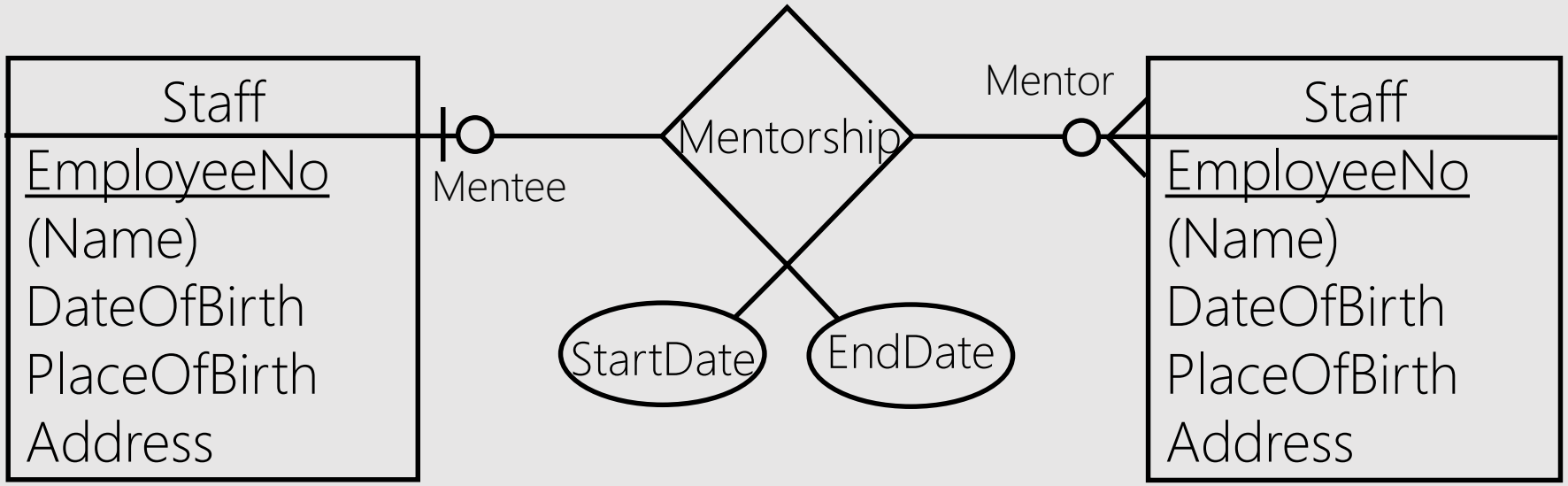
Checking(AccountNo, Balance, Plan)
Debit(CardNo, IsActive, CVV, Checking.AccountNo, StartDate, EndDate)

must be mandatory!

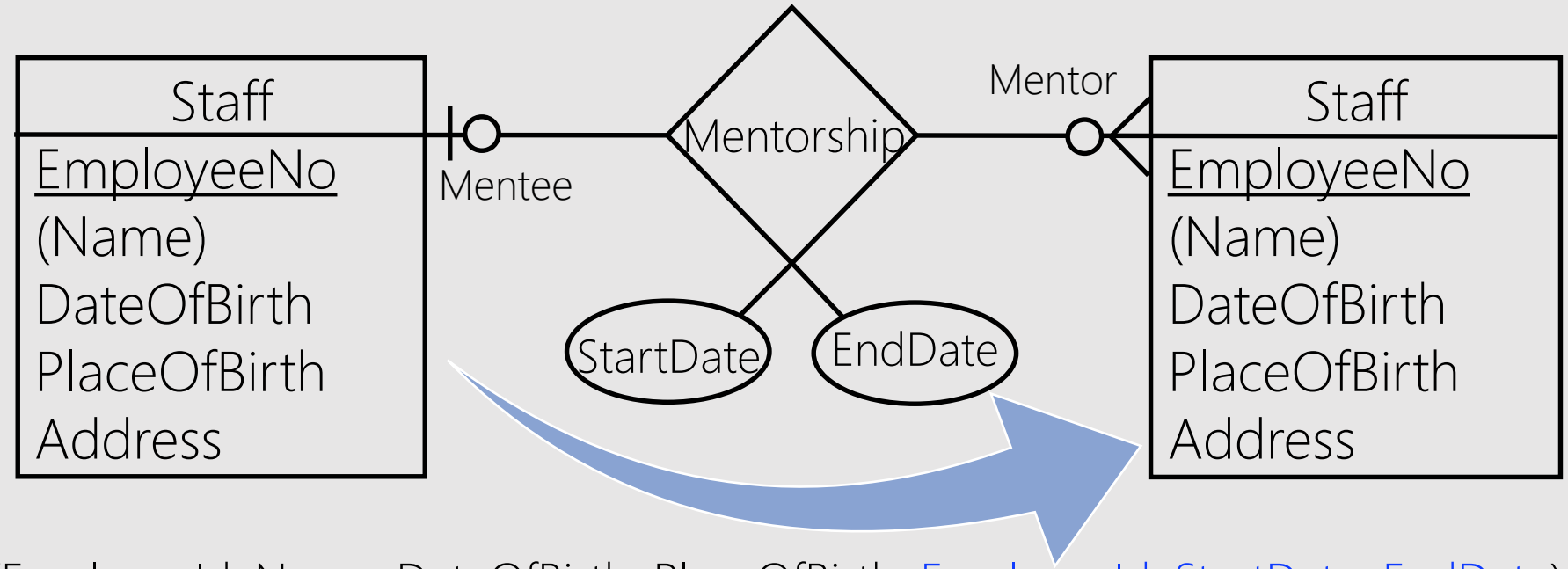
R2R × Many-One × Self



R2R × Many-One × Self

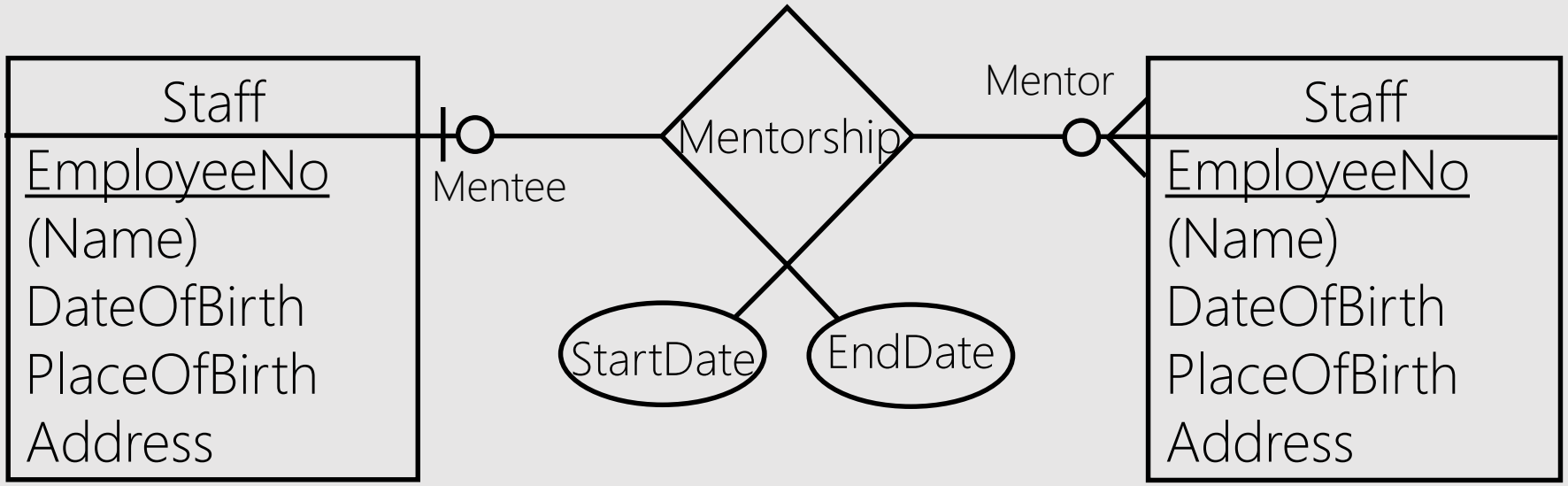


R2R × Many-One × Self



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

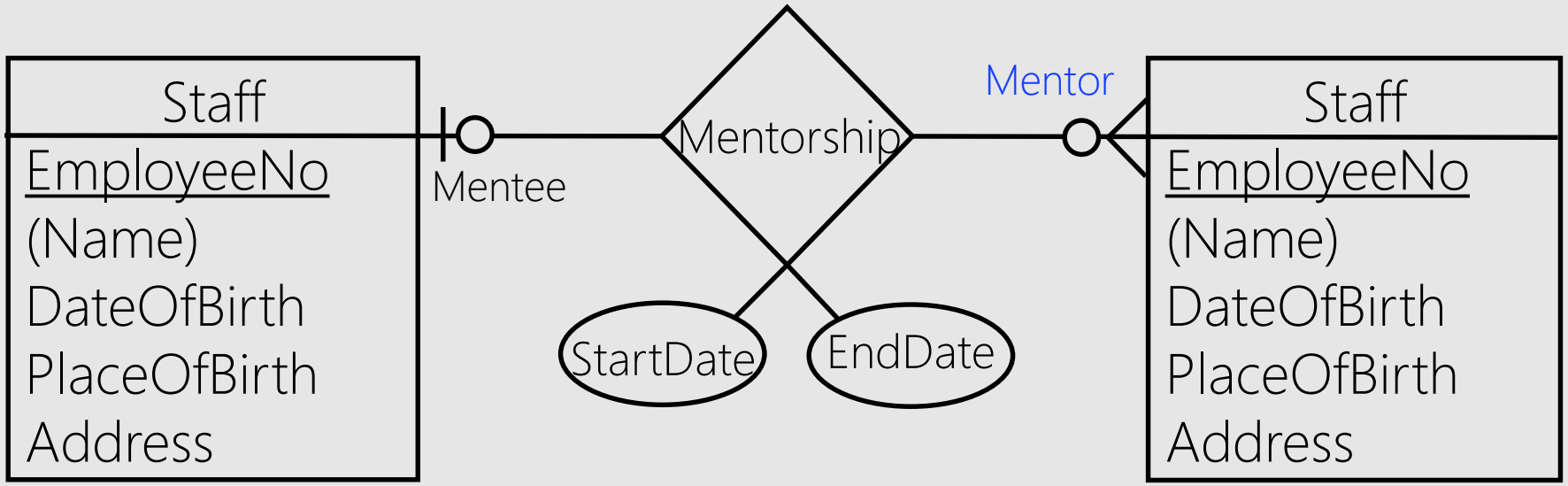
R2R × Many-One × Self



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

attribute name conflict!

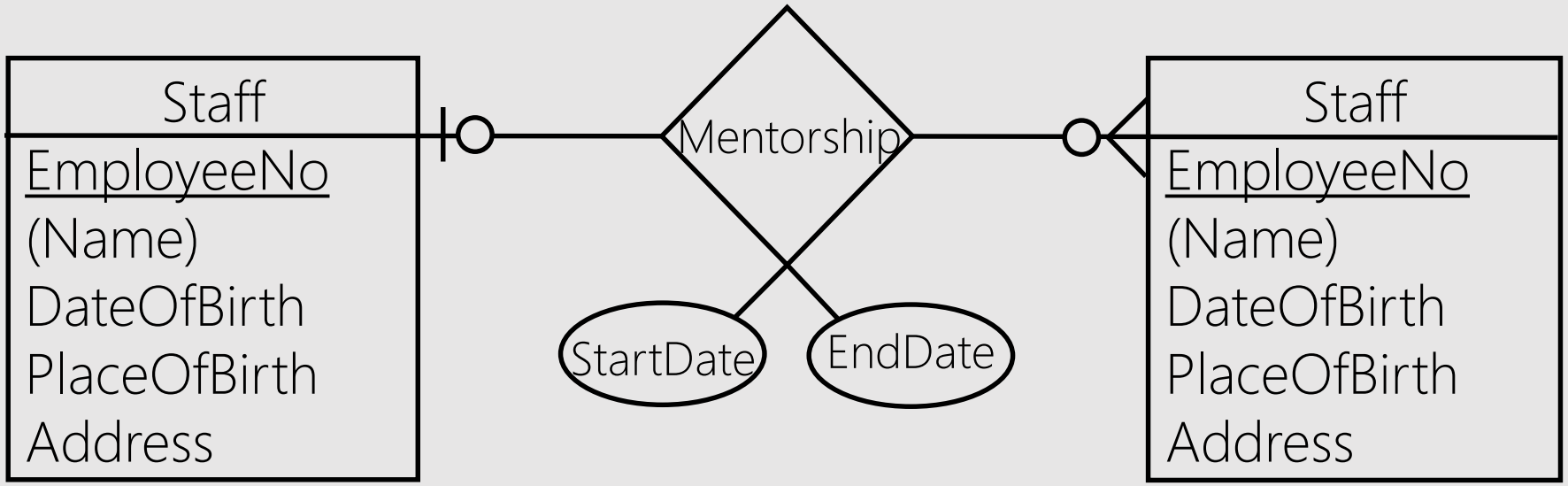
R2R × Many-One × Self



Staff(EmployeeId, Name, DateOfBirth, PlaceOfBirth, MentorId, StartDate, EndDate)

By convention, change attribute name to the role

R2R × Many-One × Self



Staff(EmployeeId, Name, DateOfBirth, PlaceOfBirth, MentorId, StartDate, EndDate)

must be optional!

R2R × Many-One × Self

Staff(EmployeeId, Name, DateOfBirth, PlaceOfBirth, MentorId, StartDate, EndDate)

Relationship2Relation (R2R)