CS 6016

Database Systems

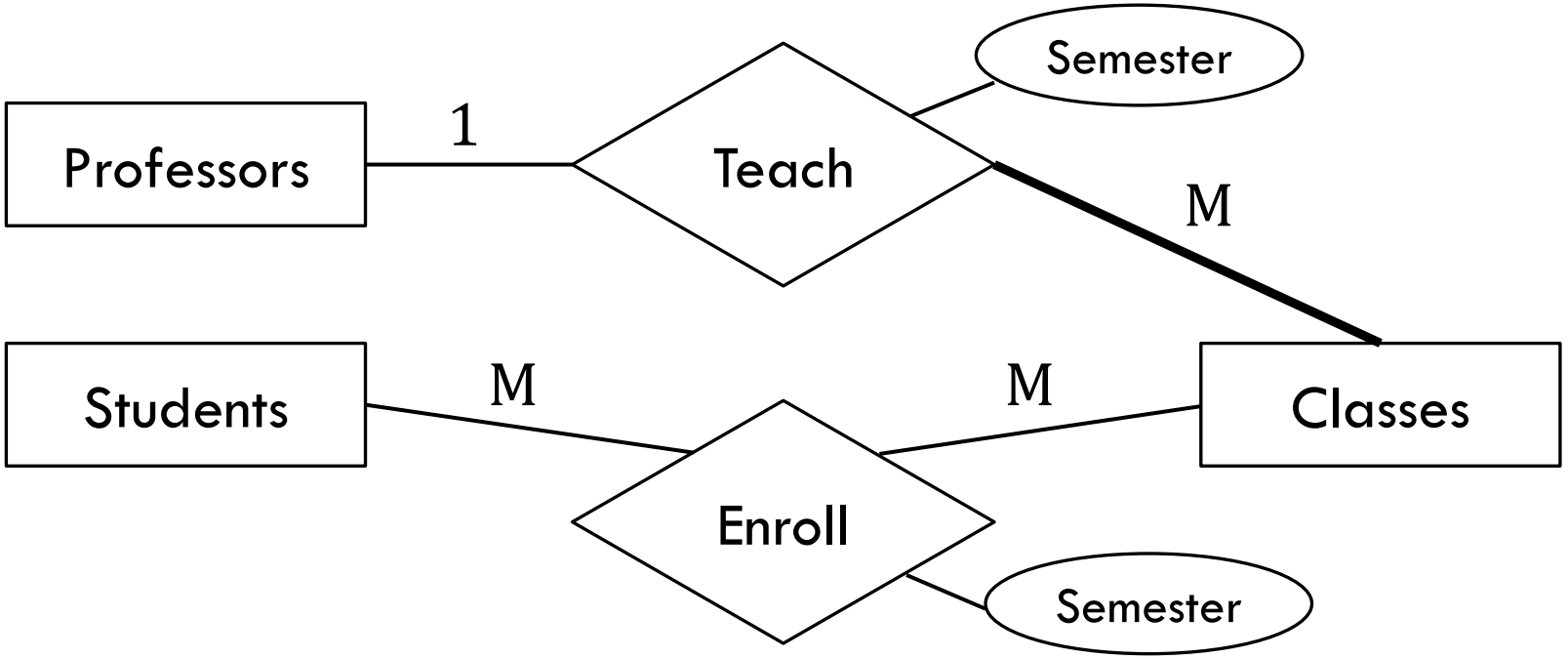
ER Model Cont.

SQL Tables

Reducing ER to Schema

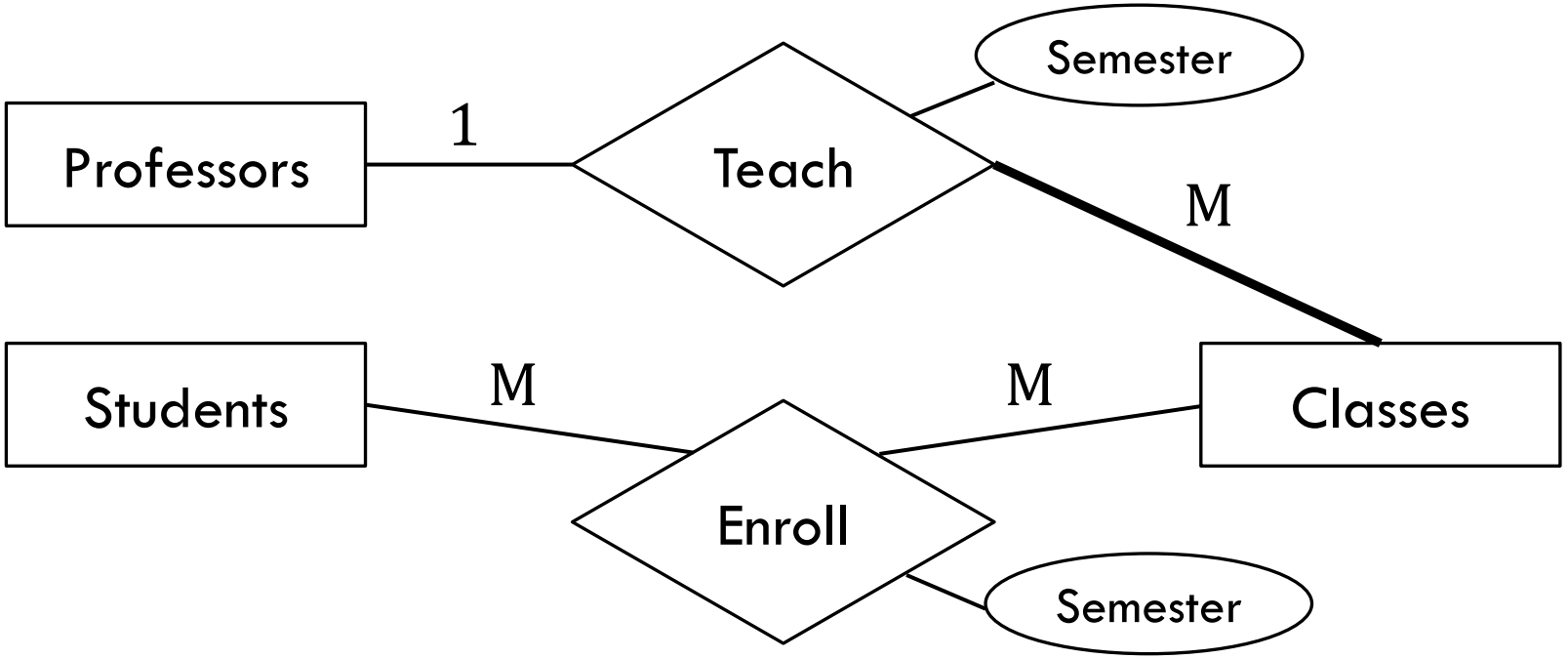
Practice (Translate Diagram)

- Can a professor take sabbatical?
- Can a class be co-taught?
- Can a class have no teacher?
- Can a class have no students?
- Can a student take multiple classes?
- Can a student take the semester off?



Practice (Translate Diagram)

- Can a professor take sabbatical? yes
- Can a class be co-taught? no
- Can a class have no teacher? no
- Can a class have no students? yes
- Can a student take multiple classes? yes
- Can a student take the semester off? yes



Hierarchical Types

```
class Employee { int SSN; string name; }


class HourlyEmployee extends Employee
{  float wage;  }


class SalaryEmployee extends Employee
{  float salary; Benefits b;  }
```

Hierarchical Types

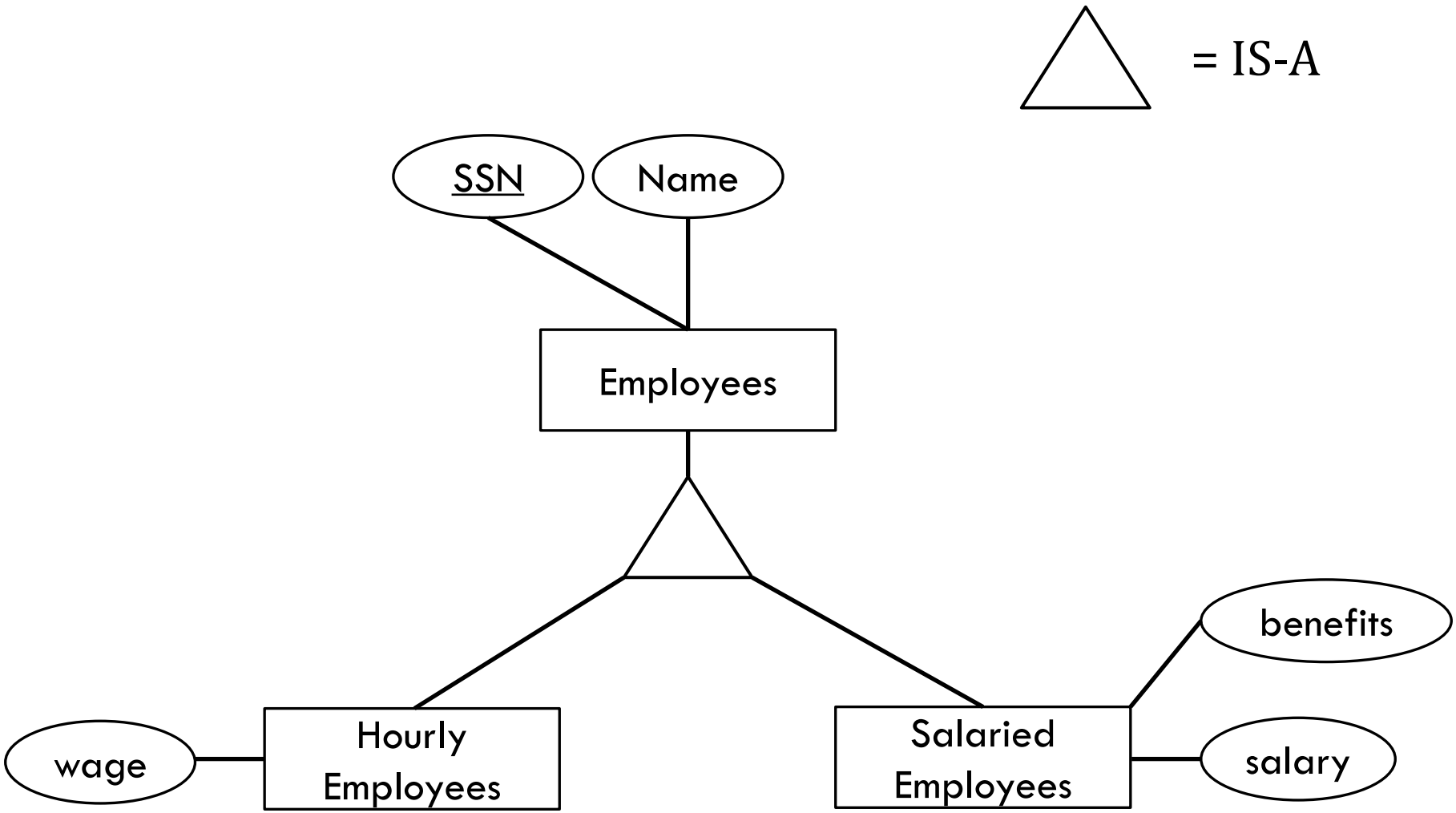
```
class Employee { int SSN; string name; }

class HourlyEmployee extends Employee
{  float wage;  }

class SalaryEmployee extends Employee
{  float salary; Benefits b;  }
```

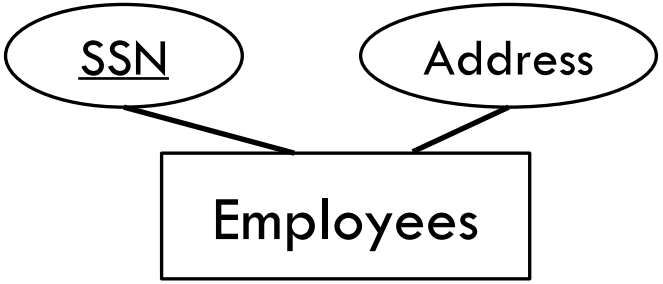
- `HourlyEmployee` "IS-A" `Employee`

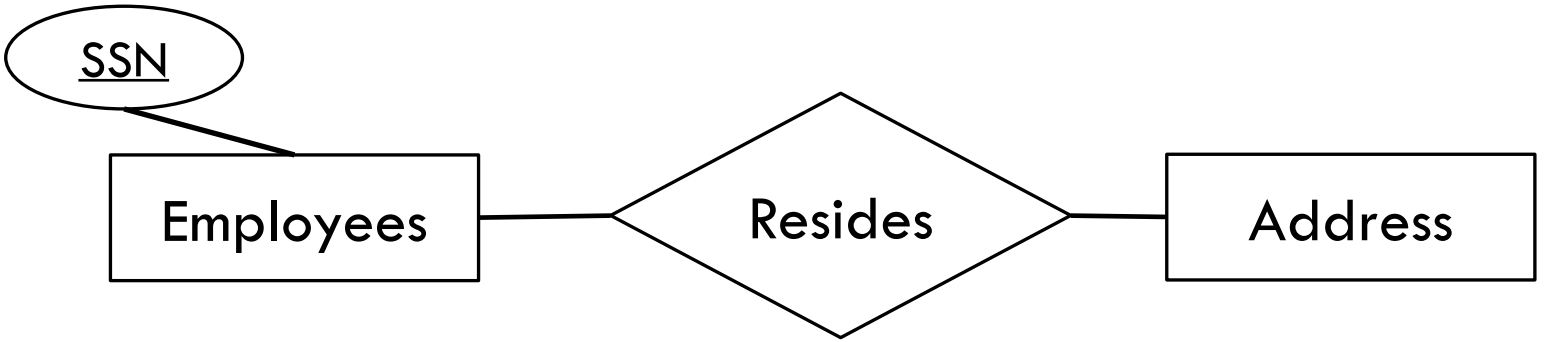
"IS-A"


= IS-A
SSN
Name
Employees
benefits
wage
Hourly Employees
Salaried Employees
salary

Entity or Attribute?

- An employee has an address





Entity or Attribute?

- It's usually obvious:
  - A student is an entity
  - A student ID is **not** an entity

Entity or Attribute?

- If it's not obvious, a few questions to ask:
  - Is it complex data that needs its own keys?

Entity or Attribute?

- If it's not obvious, a few questions to ask:
  - Is it complex data that needs its own keys?
  - Does the data type make sense in another relationship?

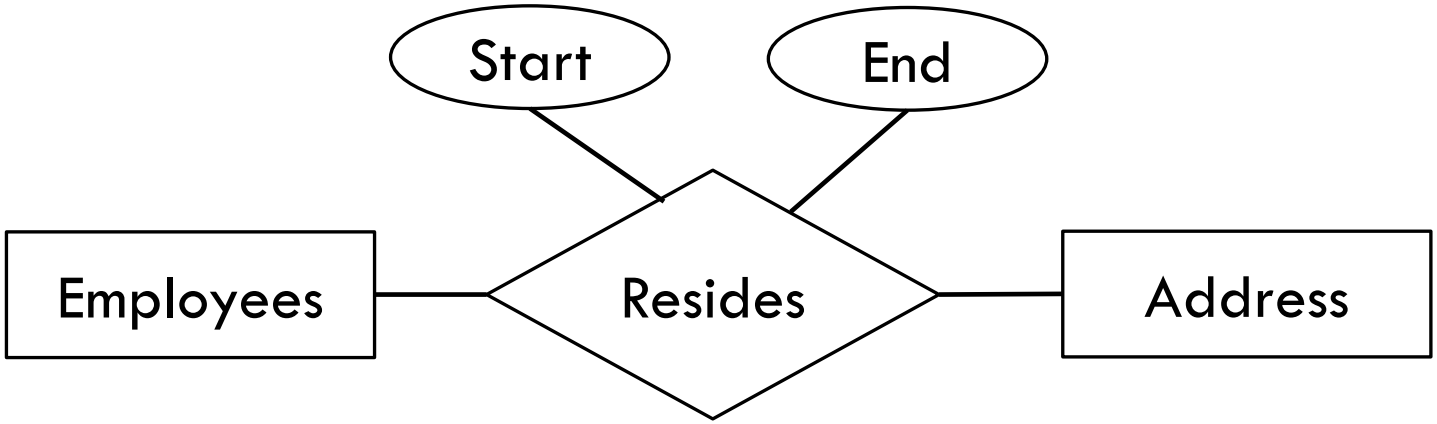
Entity or Attribute?

- If it's not obvious, a few questions to ask:
  - Is it complex data that needs its own keys?
  - Does the data type make sense in another relationship?
  - Can there be more than one?

- "yes" to these usually argues for an entity
  - But not always!

Entity or Attribute?

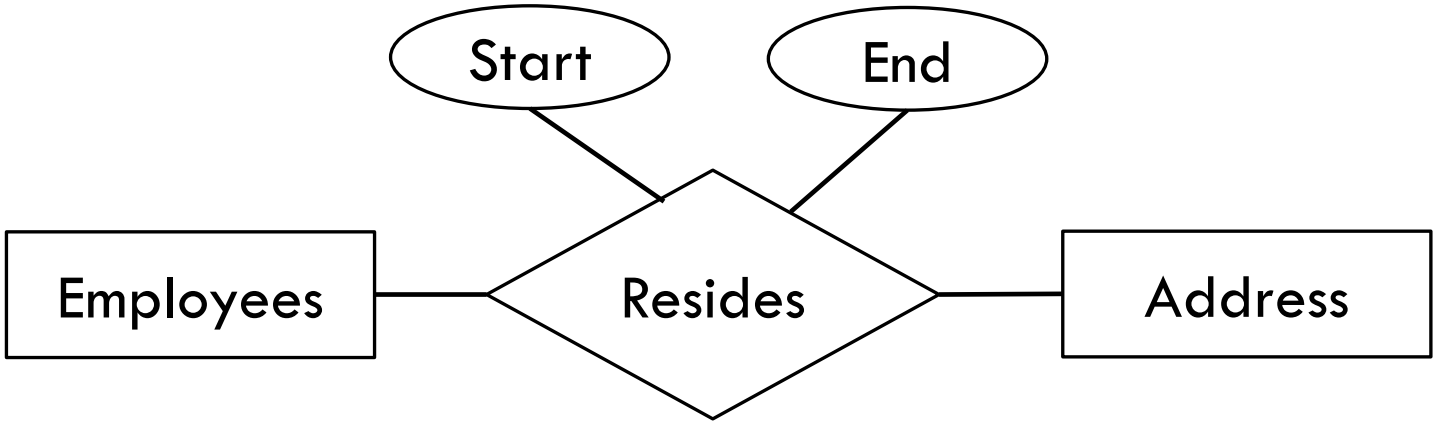
- Track full residence history



- Now an employee can have multiple addresses

Ponder

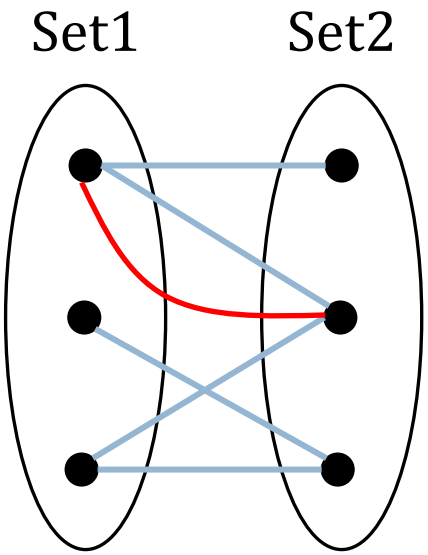
- What if the employee lives at the same residence two different times?



5/1/16 – 5/1/17, 123 Fake Street
5/1/17 – 5/1/18, 555 Creek Rd.
5/1/18 – 5/1/19, 123 Fake Street

Ponder

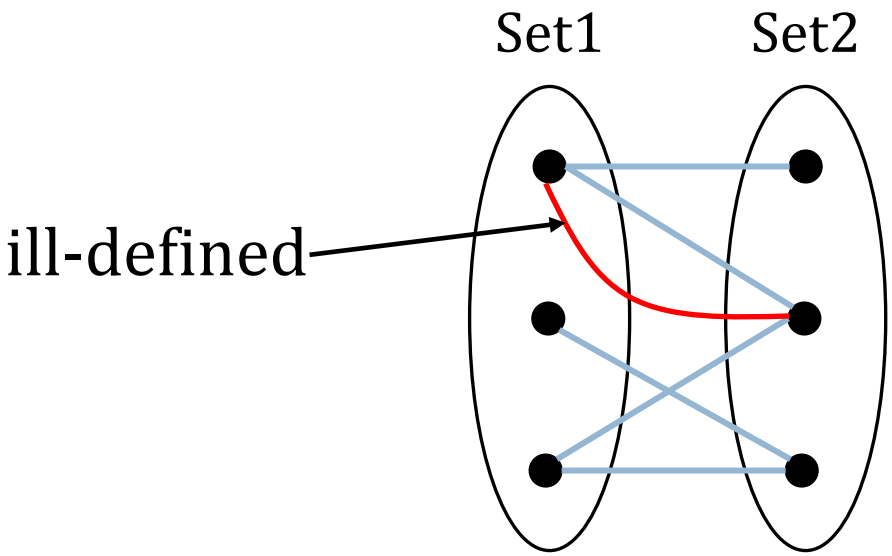
- What if the employee lives at the same residence two different times?



Many-to-Many

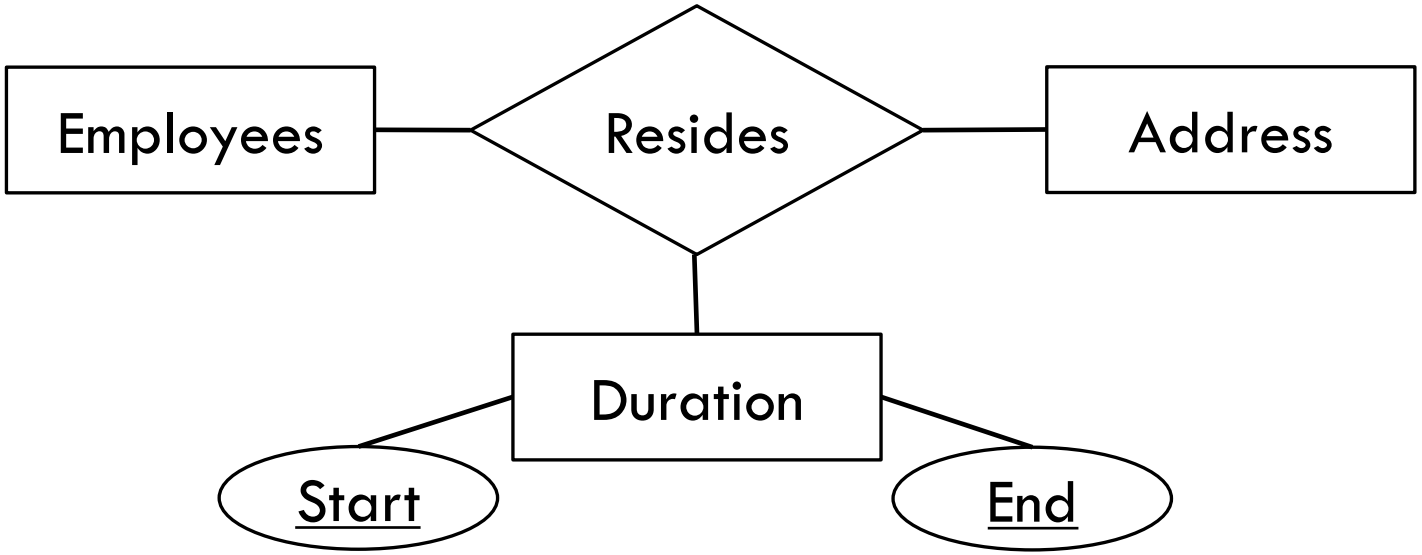
Ponder

- What if the employee lives at the same residence two different times?



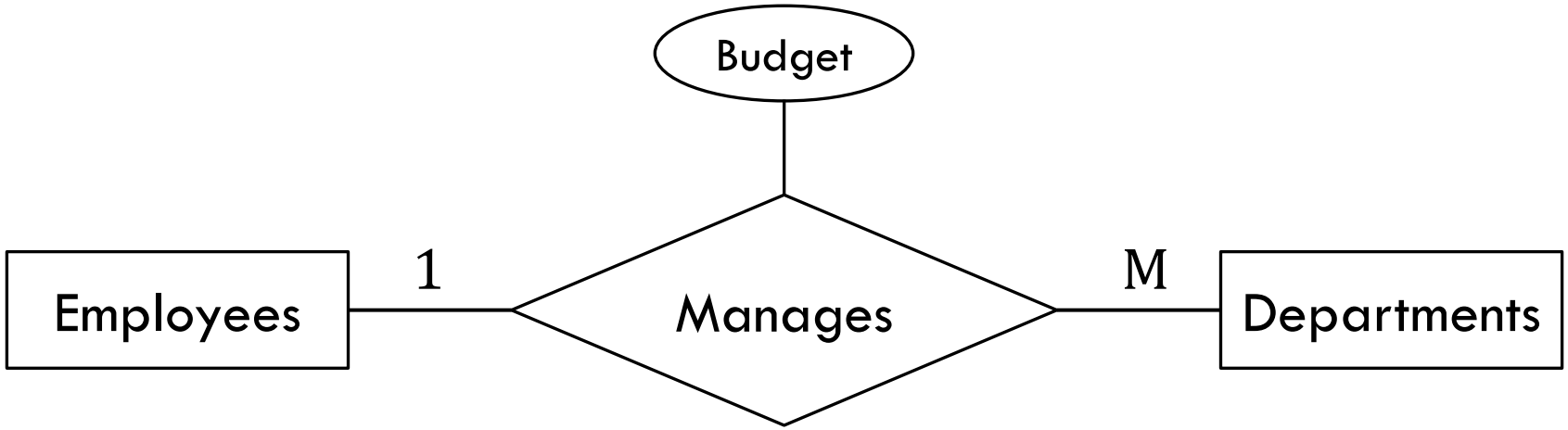
Many-to-Many

Ternary Relation



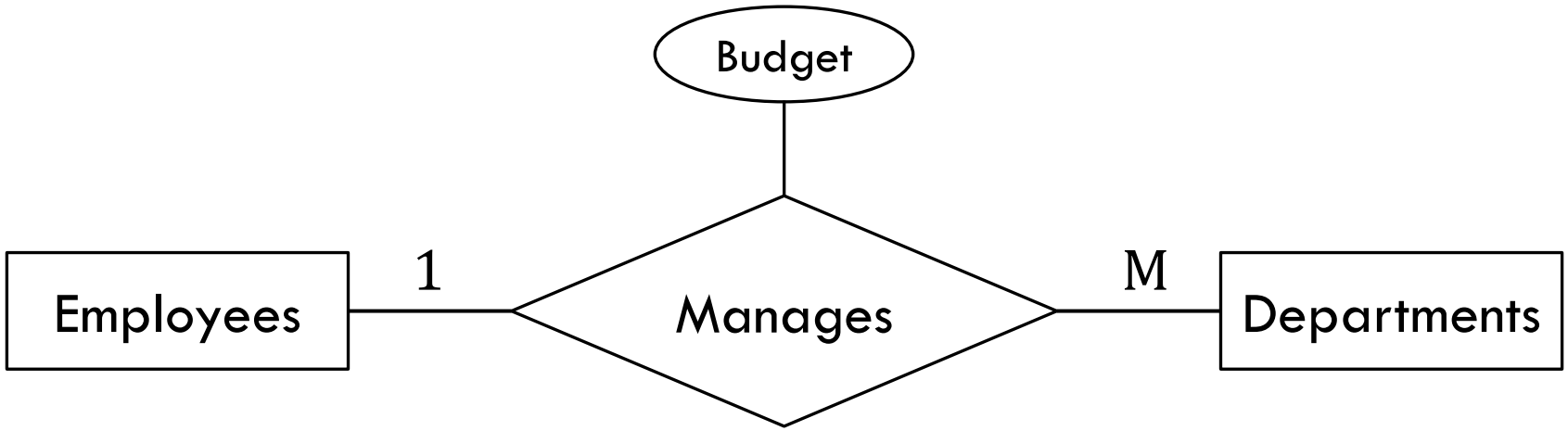
Entity vs. Relationship

- How to get a new entity out of a relationship?
- An employee can manage multiple departments
- With a different budget for each department



Entity vs. Relationship

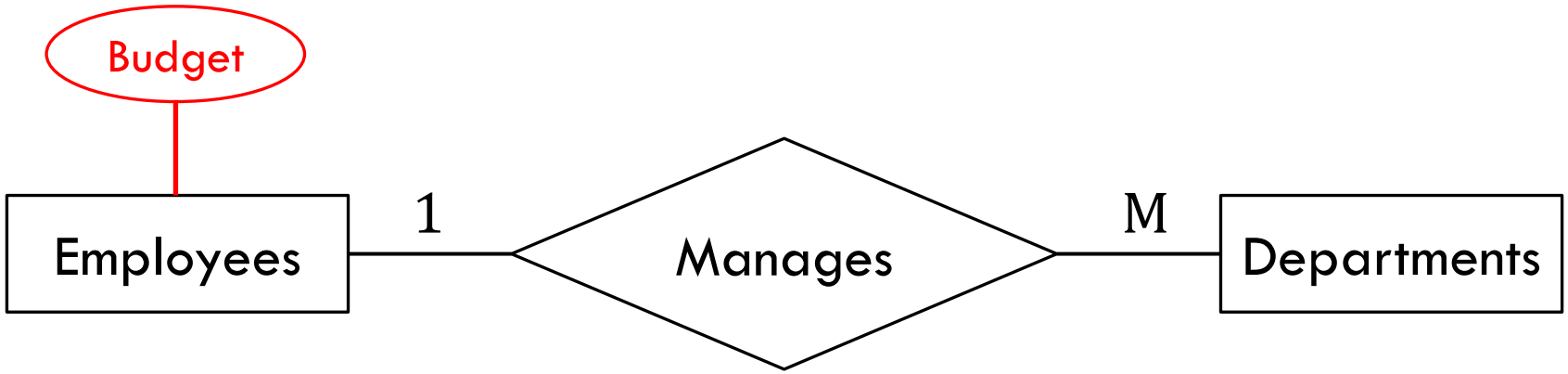
- An employee can manage multiple departments
- With a different budget for each department



- What if the manager has just one budget to split?

Entity vs. Relationship

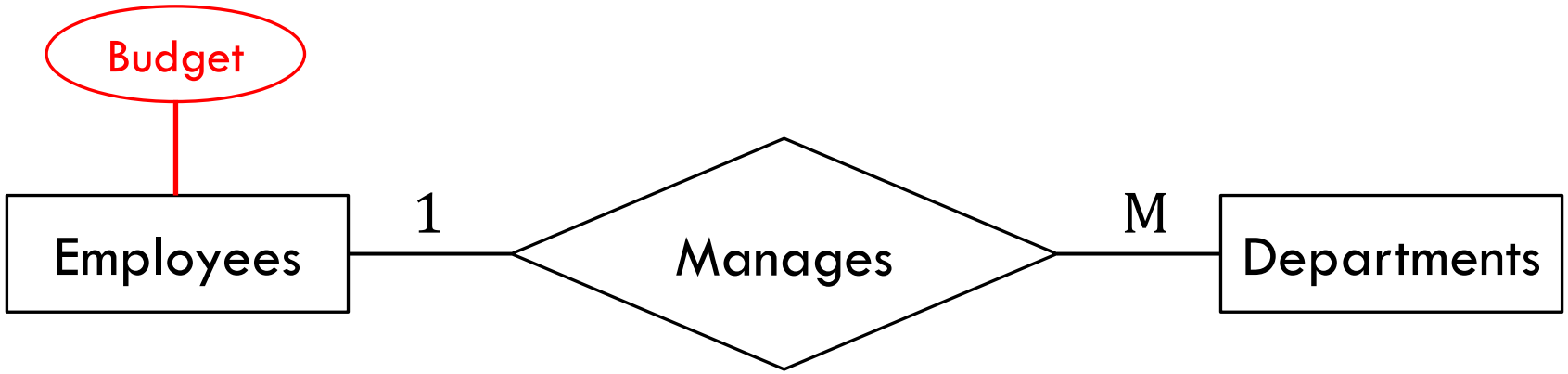
- An employee can manage multiple departments
- With a different budget for each department



- What if the manager has just one budget to split?

Entity vs. Relationship

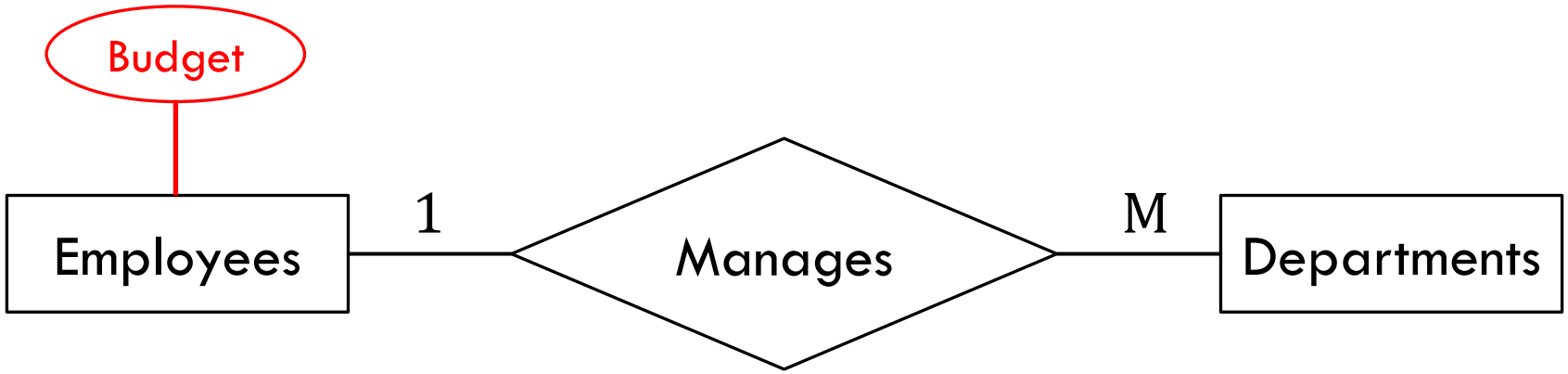
- An employee can manage multiple departments
- With a different budget for each department



- What if the manager has just one budget to split?
  - Bad: not all employees are managers

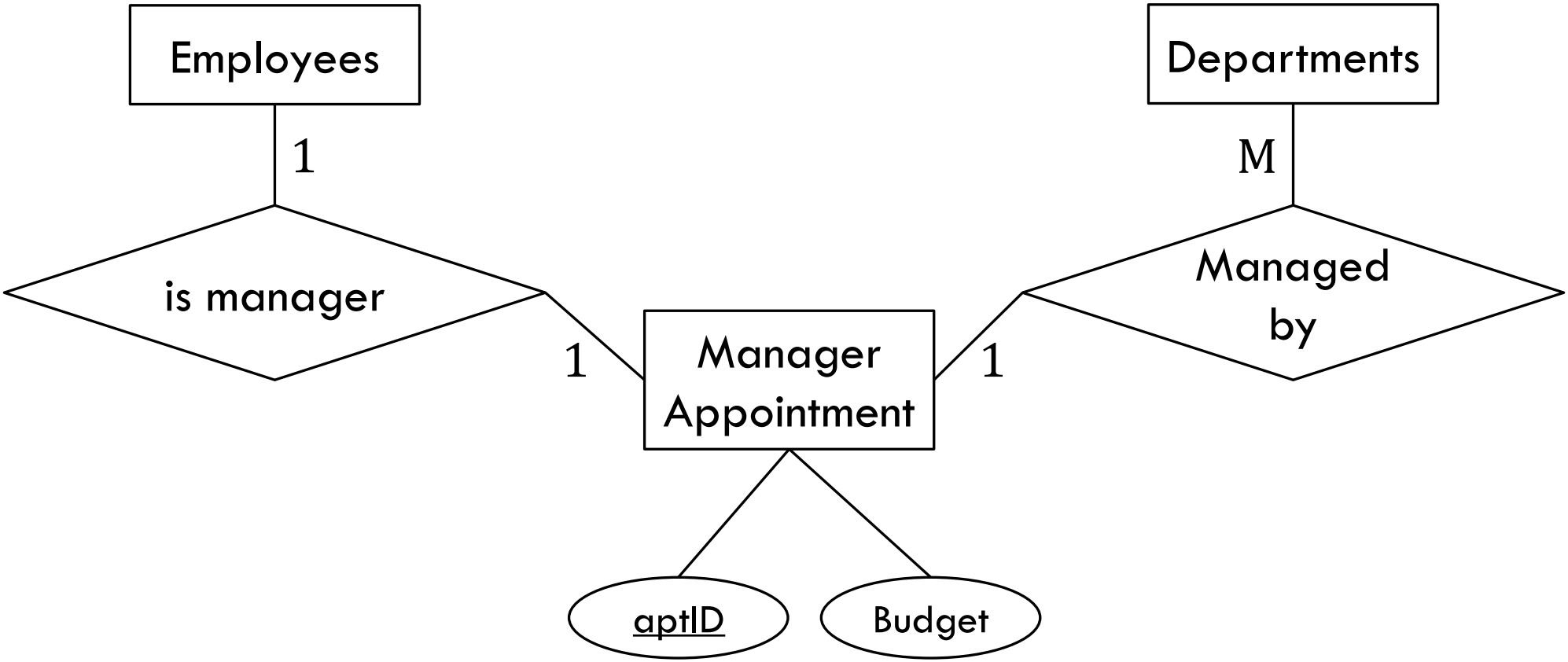
Entity vs. Relationship

- An employee can manage multiple departments
- With a different budget for each department

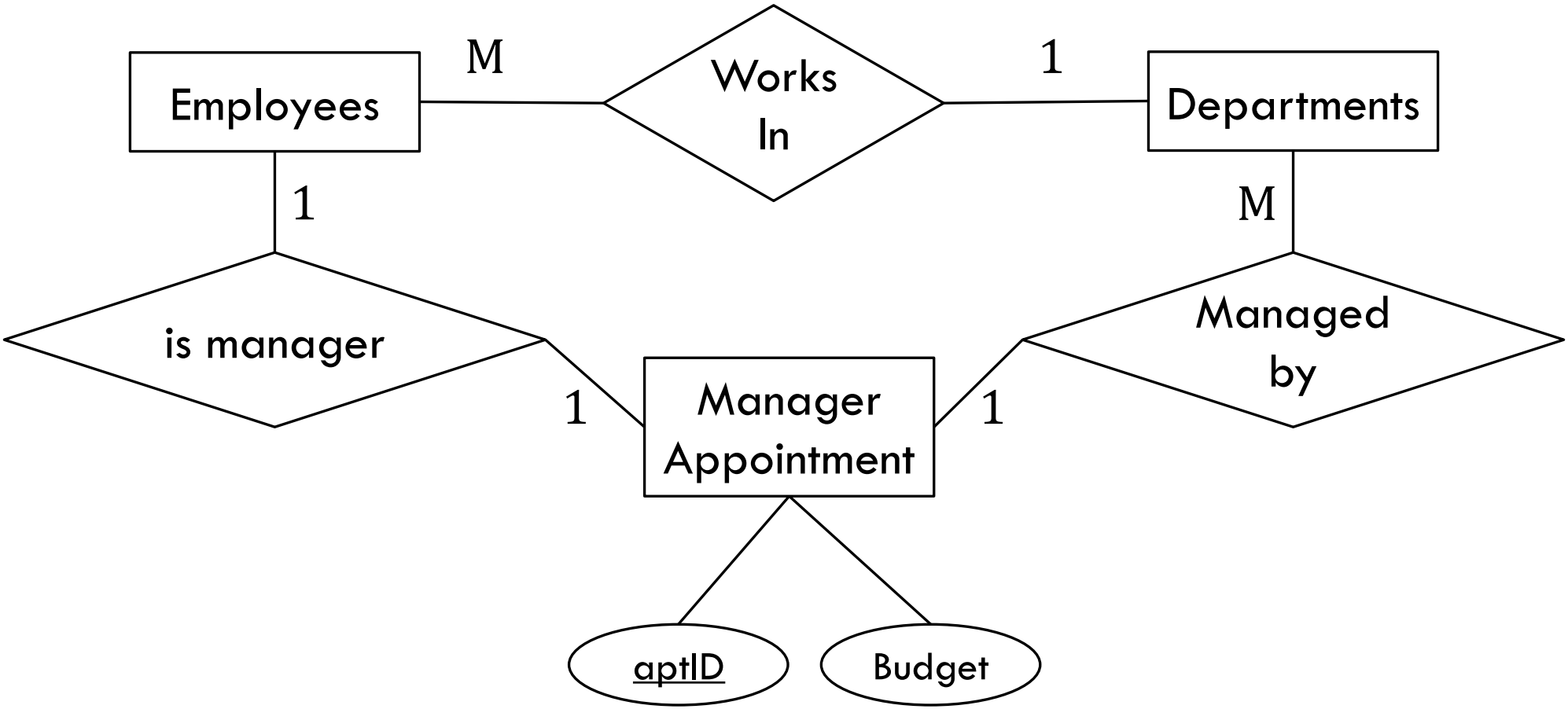


- What we need: something that defines a management role

Entity vs. Relationship



Almost Complete ER



Participation Constraints

1. Does a department have to have a manager?

2. Does a manager appointment have to be associated with at least one department?

3. Does an employee have to be a manager?

4. Does a manager appointment have to have a person associated with it?

5. Does an employee have to work for a department?

6. Does a department have to have “at least one” employee?

Participation Constraints

1. Does a department have to have a manager? **Yes, so bold.**

2. Does a manager appointment have to be associated with at least one department? **Yes, so bold.**

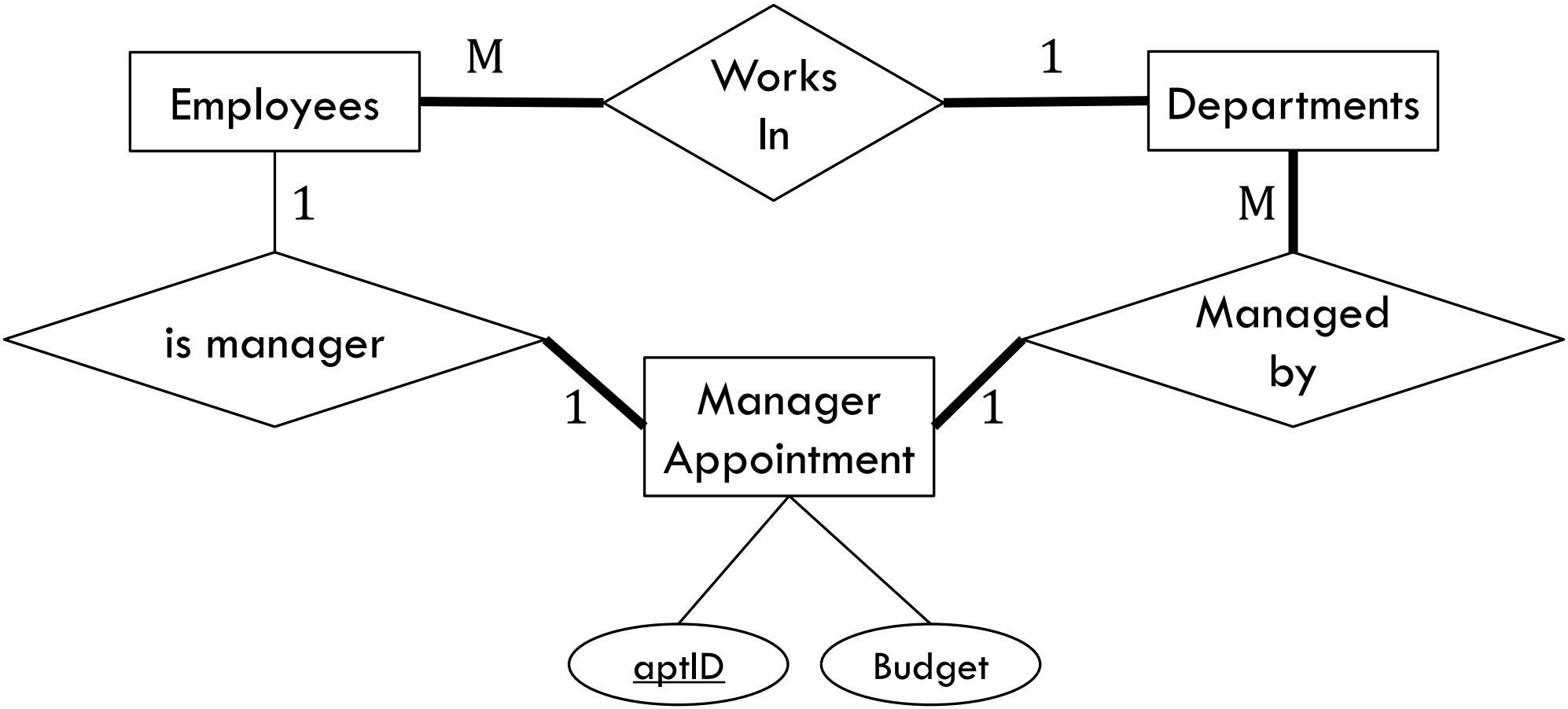
3. Does an employee have to be a manager? **No, so no bold.**

4. Does a manager appointment have to have a person associated with it? **Yes, so bold.**

5. Does an employee have to work for a department? **Yes, so bold.**

6. Does a department have to have “at least one” employee? **Yes, so bold.**

Almost Complete ER



Weak Entities

- A weak entity can't be identified by its own attributes

- It is identified by a combination of:
  - its own attribute(s)
  - and a foreign key (something else's key attribute)

- Can not be uniquely identified by its own attributes

Weak Entities

- A weak entity can't be identified by its own attributes

- It is identified by a combination of:
  - its own attribute(s)
  - and a foreign key (something else's key attribute)

- i.e., its existence only makes sense in the context of another entity

Example

- Consider the difference between a *course* and a *class*
  - Course
    - Subject
    - Number
    - Name
    - Description
- CS 6016 Database Systems, “in this class we study …”,

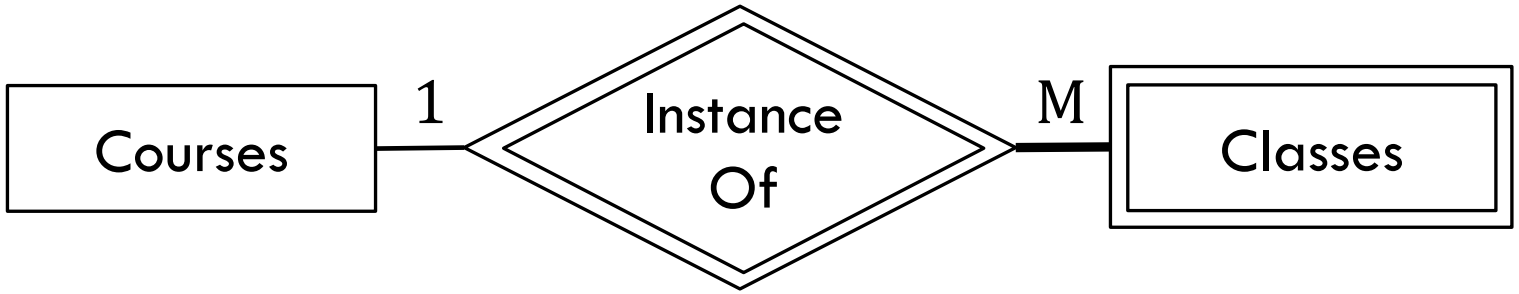
Example

- Consider the difference between a *course* and a *class*
  - Class
    - Semester
    - <reference to course>
    - <reference to teacher>

- Spring 21, <reference to CS6016>, <reference to teacher>

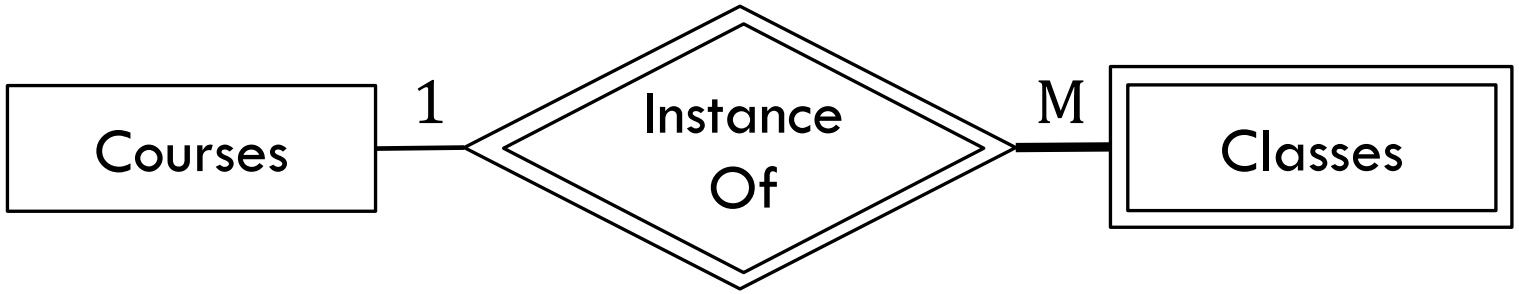
Weak Entity

- `Classes` is a weak entity set
- `Instance of` is a supporting relationship set



Weak Entity

- `Classes` is a weak entity set
  - Double rectangle for the entity set
  - Double diamond for the supporting relationship
  - Supporting relationship must be 1-to-M, and weak entity must participate



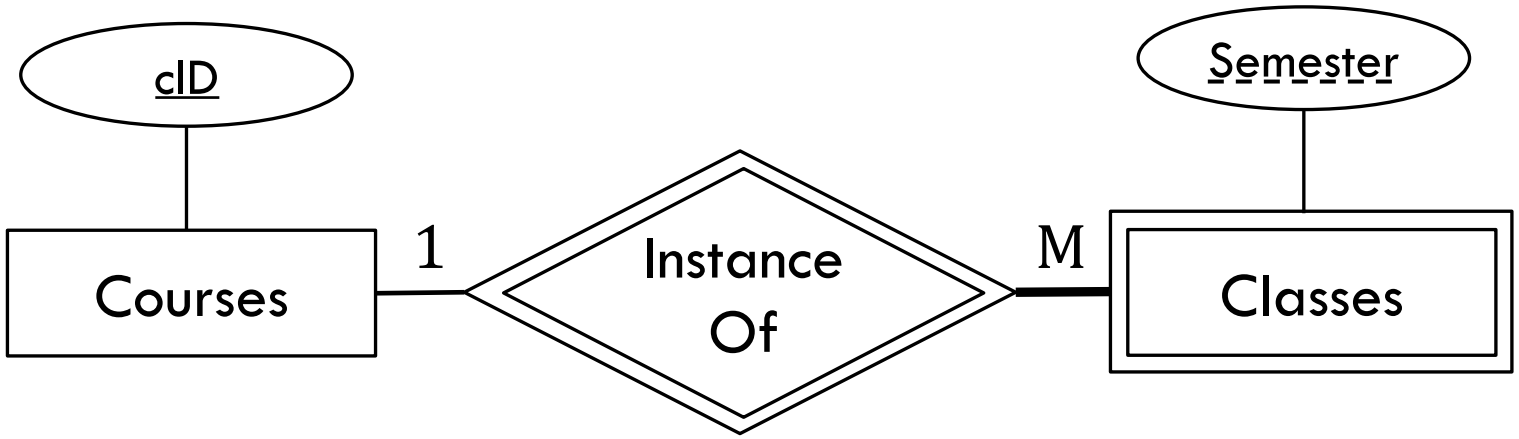
Weak Entities

- A weak entity can't be identified by its own attributes

- It is identified by a combination of:
  - its own attribute(s)
  - and a foreign key

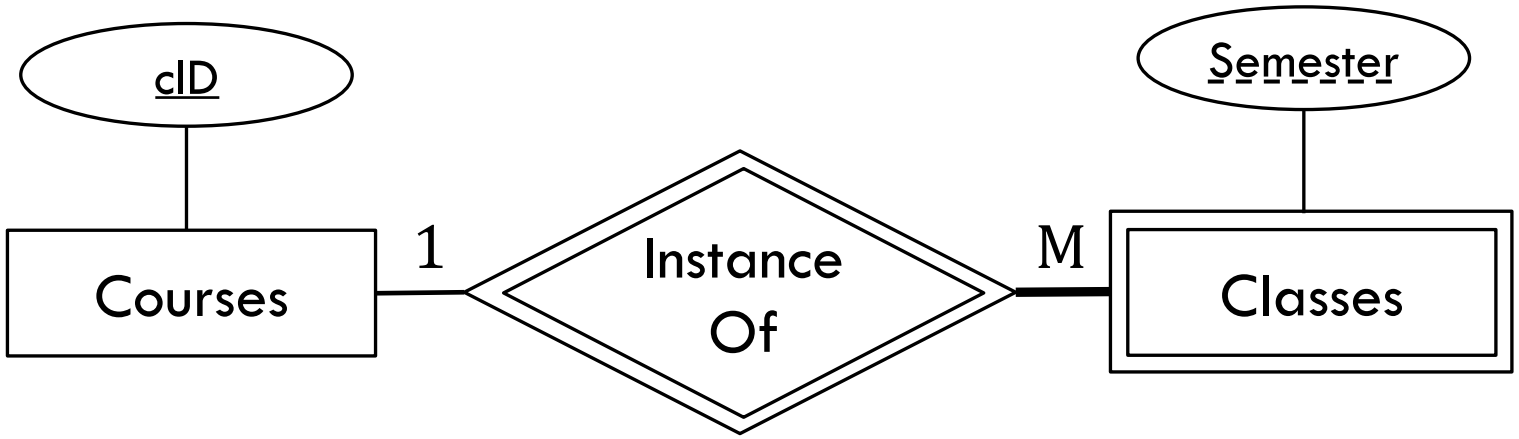
Partial Key

- A weak entity set has a partial key
  - Dashed underline



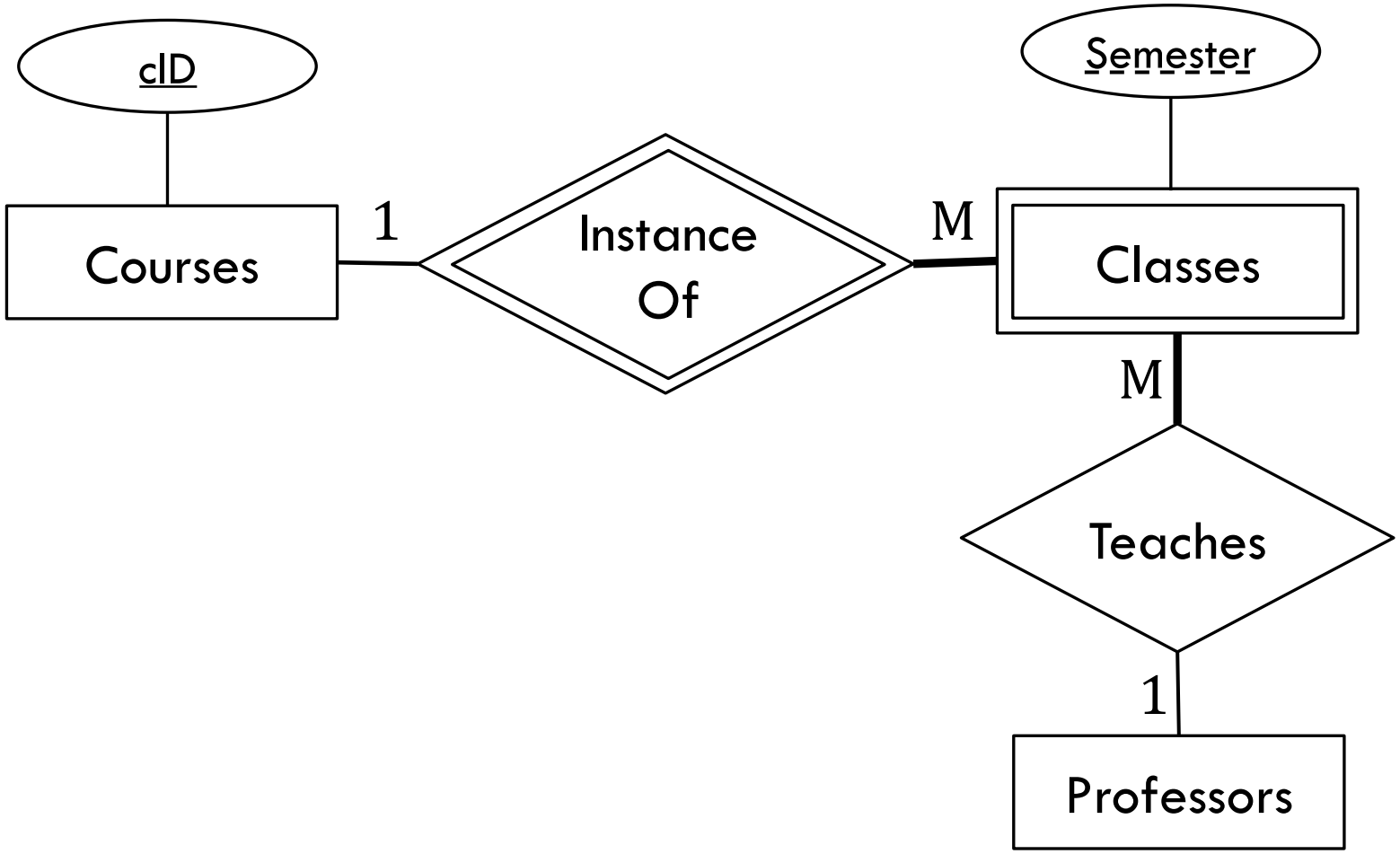
Partial Key

- A weak entity set has a partial key
  - Dashed underline
  - Must be combined with a foreign key
  - `{cID, Semester}` is the key for `Classes`



Partial Key

- Weak entity sets can have non-supporting relationships



Creating Tables in SQL

```
create table <name> (
  <column1Name> <type> <properties>,
  <column2Name> <type> <properties>,
  …
  <table properties>
);
```

- Properties are optional

MySQL Data Types

- Numeric
  - Integers – int, <tiny, small, medium, big>int, <unsigned>
  - Reals – float, double, decimal

MySQL Data Types

- Numeric
  - Integers – int, <tiny, small, medium, big>int, <unsigned>
  - Reals – float, double, decimal

- Dates – very important in DBs
  - Date, datetime, time stamp, time, year

MySQL Data Types

- Numeric
  - Integers – int, <tiny, small, medium, big>int, <unsigned>
  - Reals – float, double, decimal

- Dates – very important in DBs
  - Date, datetime, time stamp, time, year

- Strings
  - char(m), varchar(m)

MySQL Data Types

- Numeric
  - Integers – int, <tiny, small, medium, big>int, <unsigned>
  - Reals – float, double, decimal

- Dates – very important in DBs
  - Date, datetime, time stamp, time, year

- Strings
  - char(m), varchar(m)

- Blobs – Binary Large Objects

- Enums

Strings

- char(N) - exactly N characters
- varchar(N) - up to N characters

Quiz

- Best type for ‘CardNum’?

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Quiz

- Best type for ‘CardNum’?
  INT UNSIGNED

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Quiz

- Best type for 'Author'?

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Quiz

- Best type for 'Author'?
  varchar(…)

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Quiz

- Best type for 'ISBN'?

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Quiz

- Best type for 'ISBN'?
  - char(14)
  - unless we get rid of the dashes

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Strings

- CHAR(N) - exactly N characters
  - ISBN
  - Phone number
  - We could use BIGINT UNSIGNED if we left out the dash '-'

- VARCHAR(N) - up to N characters
  - Title
  - Author
  - Name

Strings

- How to pick (N) for VARCHAR(N)?
- Requires N bytes for storage plus 1 or 2 bytes for size
  - 1 size byte if N <= 255
  - 2 size bytes if N > 255

Strings

- How to pick (N) for VARCHAR(N)?
- Requires N bytes for storage plus 1 or 2 bytes for size
  - 1 size byte if N <= 255
  - 2 size bytes if N > 255
- Pick the smallest value that is always large enough

Properties

- Column properties:
  - `NOT NULL`
  - `DEFAULT 'hello'`
  - `AUTO_INCREMENT`

- Table properties:
  - `PRIMARY KEY (column1, ...)`
  - `UNIQUE (column1, ...)`

Not Null

- Improves index optimizations (faster tree-based searches on columns)

- … it's not just for table design

- Specifying `NOT NULL` is not required on key values
  - But SQL will automatically set it

Quiz

- Appropriate properties for ‘CardNum’?

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Quiz

- Appropriate properties for ‘CardNum’?

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

- `not null`
- `auto_increment`
- `primary key`

Quiz

- Appropriate properties for ‘CardNum’?

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Quiz

- Appropriate properties for ‘CardNum’?

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

- `not null`

Creating Tables

- Let's create this table (without the contents)

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Creating Tables

```
create table Titles (
  ISBN char(14) not null,
  Title varchar(255) not null,
  Author varchar(255) not null,
  primary key(ISBN)
);
```

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Exercise

- Command to create this table? (without the contents)

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427

```
create table <name> (
  <column1Name> <type> <properties>,
  <column2Name> <type> <properties>,
  <table properties>
);
```

Solution

```
create table Inventory (
   Serial int unsigned not null auto_increment,
   ISBN char(14) not null,
   primary key(Serial)
);
```

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427

Foreign Keys

```
FOREIGN KEY (<column>) REFERENCES
<table>(<table's key>)
ON DELETE <action>
ON UPDATE <action>
```

Foreign Keys

```
FOREIGN KEY (<column>) REFERENCES
<table>(<table's key>)
ON DELETE <action>
ON UPDATE <action>
```

- `<action>` can be:
  - `RESTRICT` (Default): disallow the change
  - `CASCADE`: also delete/update in child table
  - `SET NULL`: nullify key in child table
  - `SET DEFAULT`: set to column's default value

Foreign Key Example

```
CREATE TABLE Phones (
   ...
   FOREIGN KEY (CardNum) REFERENCES Patrons (CardNum)
   ON DELETE CASCADE
)
```

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

ER → RM Algorithm

- We've been using a "naïve", but correct relational model
  - Sometimes resulting in unnecessary tables

ER → RM Algorithm

- If we start with a good ER model, the translation to a good RM is mechanical

ER → RM Algorithm

- If we start with a good ER model, the translation to a good RM is mechanical

- Basic algorithm:

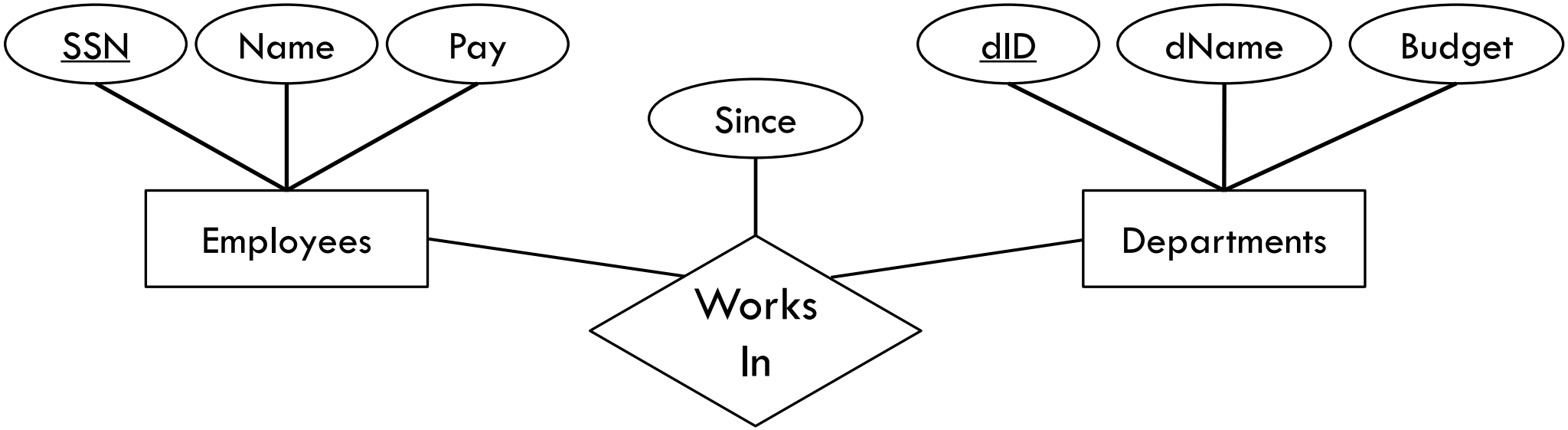
    1. Every entity set becomes a schema

    2. Relationship sets *might* become schema depending on cardinality, otherwise they are *merged*

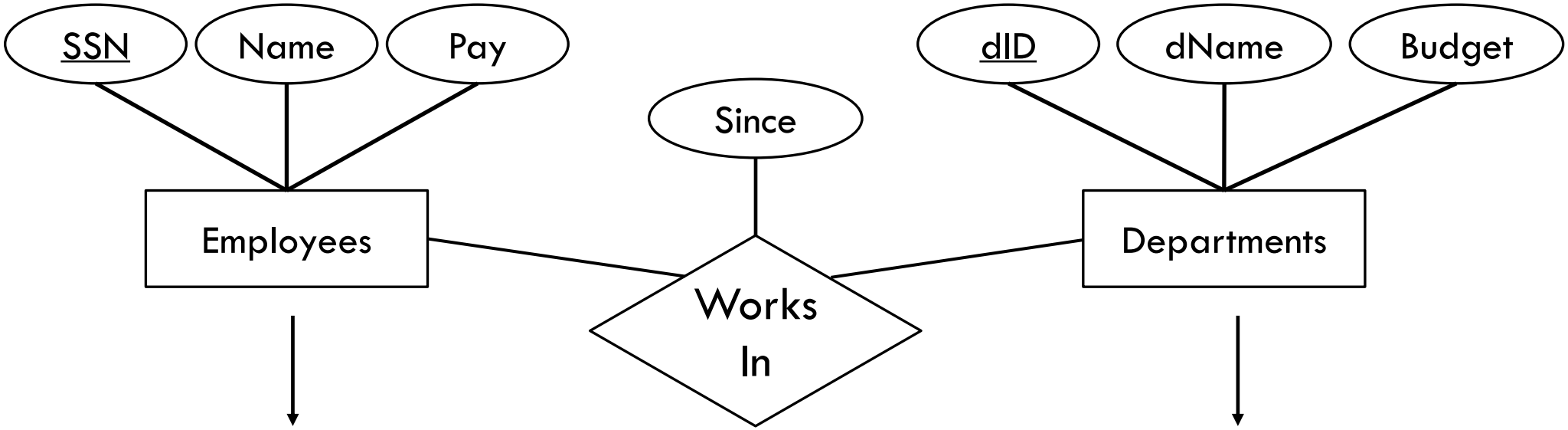
Entity Sets to Schema

- **Entity Sets**
- Every entity set translates directly to a schema
- Attributes become columns
- Pick one of the key attribute sets as the primary key

Entity Sets to Schema



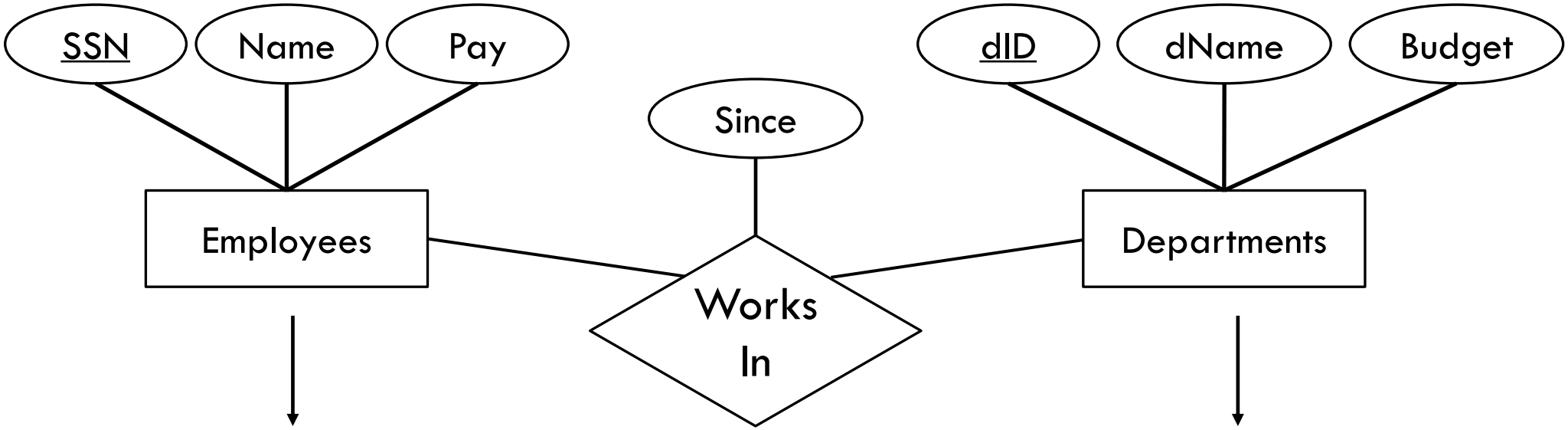
Entity Sets to Schema



SSN	Name	Pay
…	…	…
PK NN	NN	NN

dID	dName	Budget
…	…	…
PK NN	NN	NN

Entity Sets to Schema



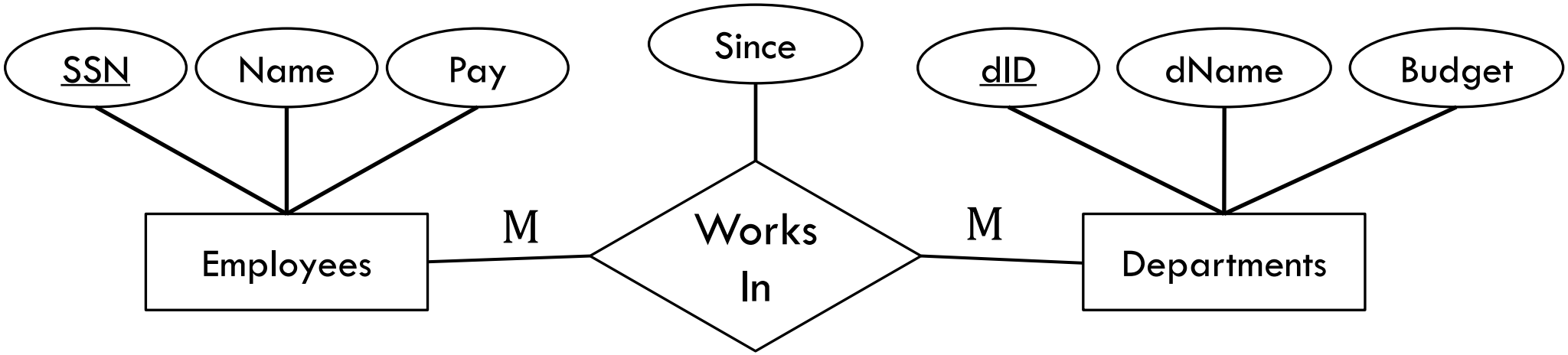
SSN	Name	Pay
…	…	…
PK NN	NN	NN

?

dID	dName	Budget
…	…	…
PK NN	NN	NN

Relationship Sets to Schema

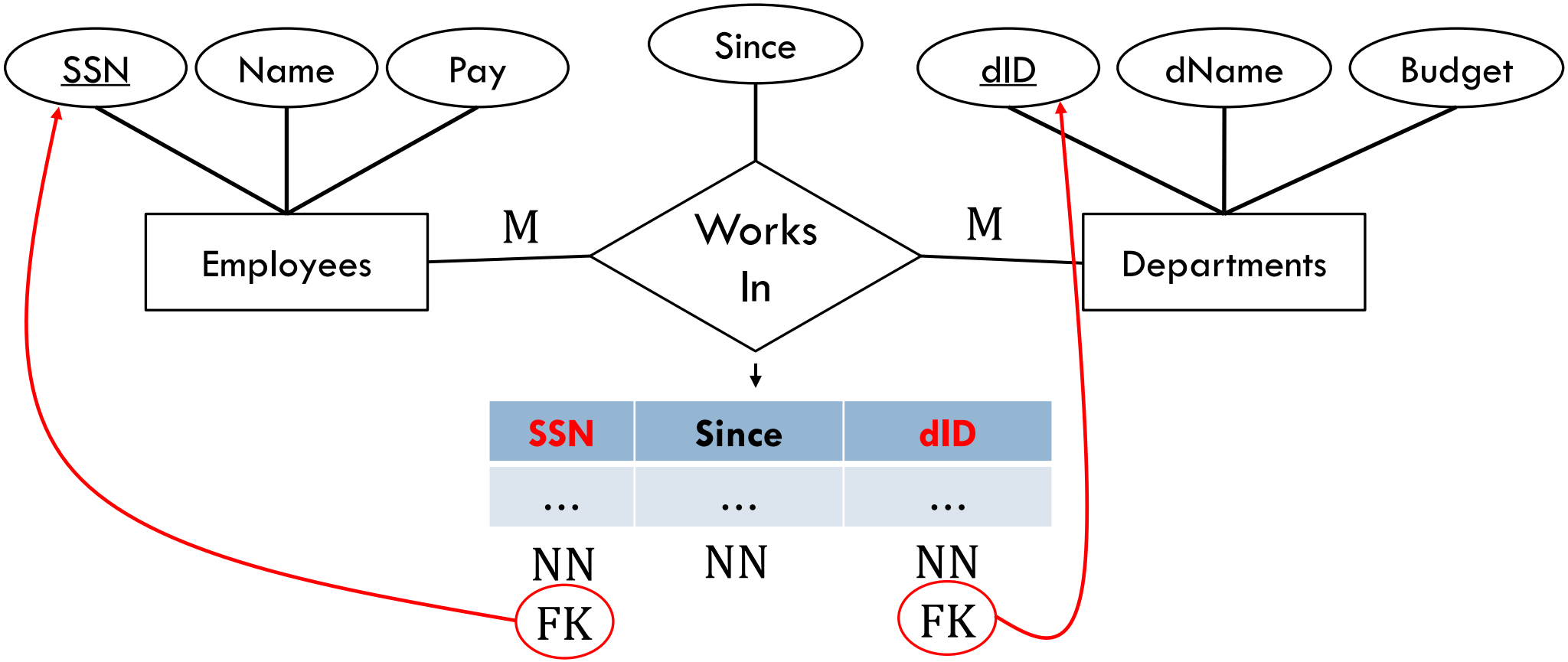
- **Many-to-Many**
  - New schema for the relationship



?	Since	?
…	…	…

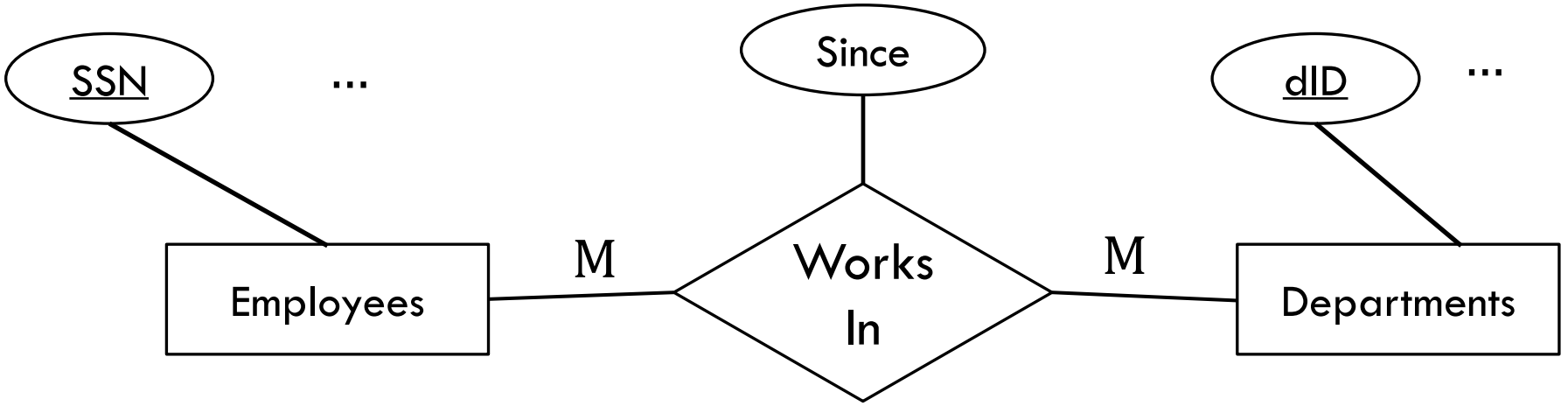
Relationship Sets to Schema

- **Many-to-Many**
  - Primary keys of relating entities as foreign keys



Relationship Sets to Schema

- **Many-to-Many**
  - What is the key of the `WorksIn` table?

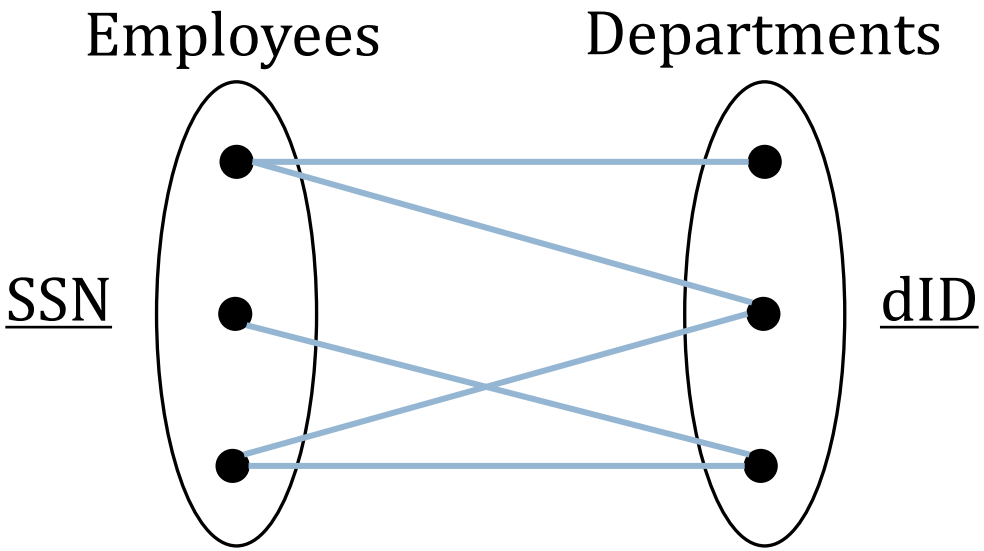


SSN	Since	dID
...	...	...
NN FK	NN	NN FK

PK?

Relationship Sets to Schema

- How do we uniquely identify a line (many-to-many)?

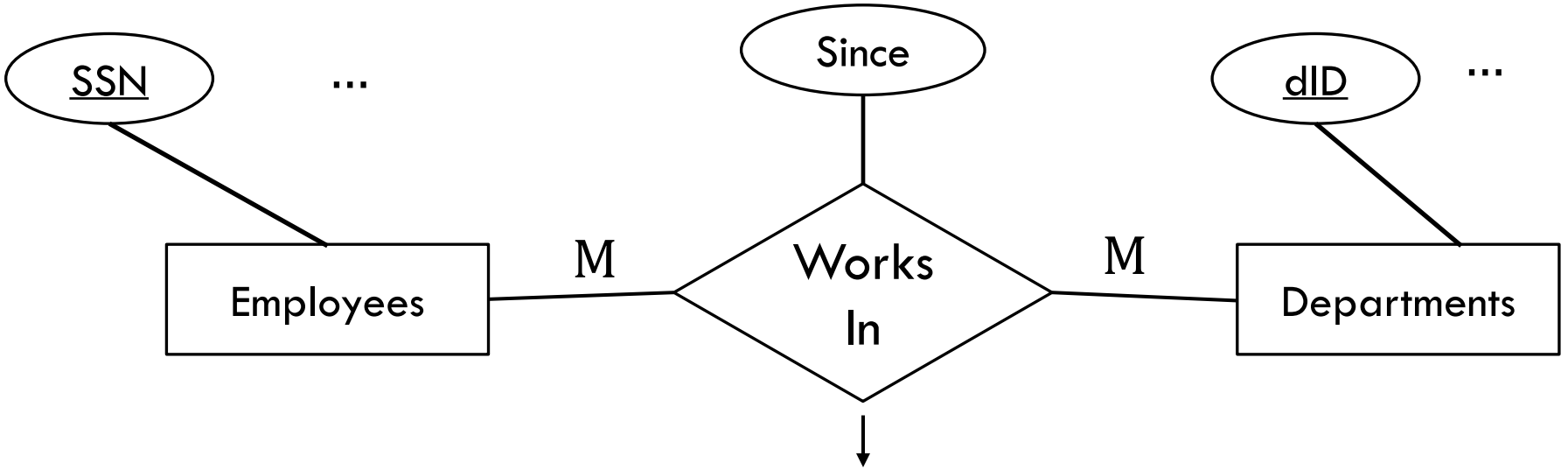


WorksIn

(SSN, dID)

Relationship Sets to Schema

- **Many-to-Many**
  - Key is the combination of the foreign keys

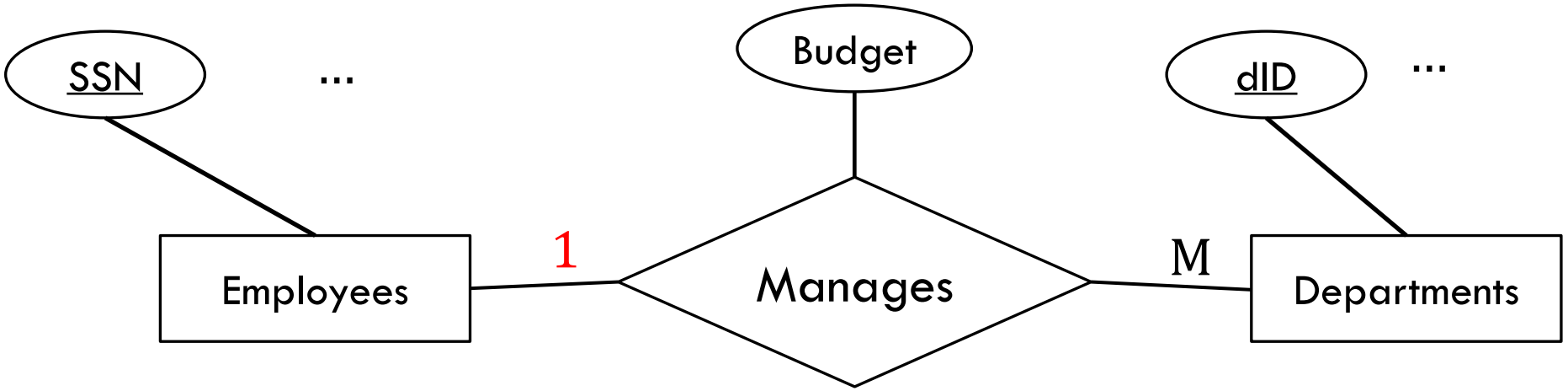


SSN	dID	Since
...	...	...
NN	NN	NN
FK	FK	

-------PK-------

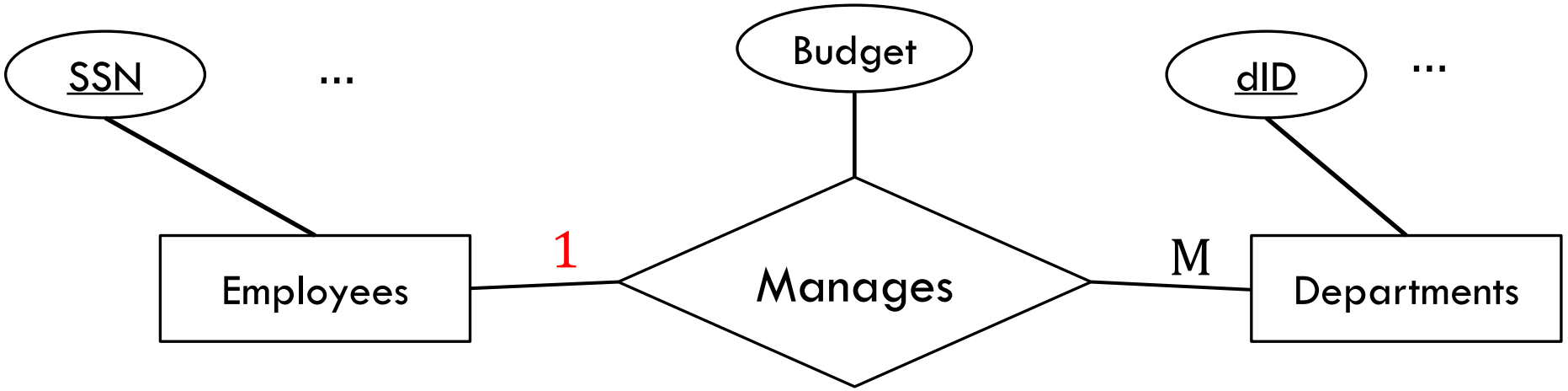
Relationship Sets to Schema

- 1-to-Many



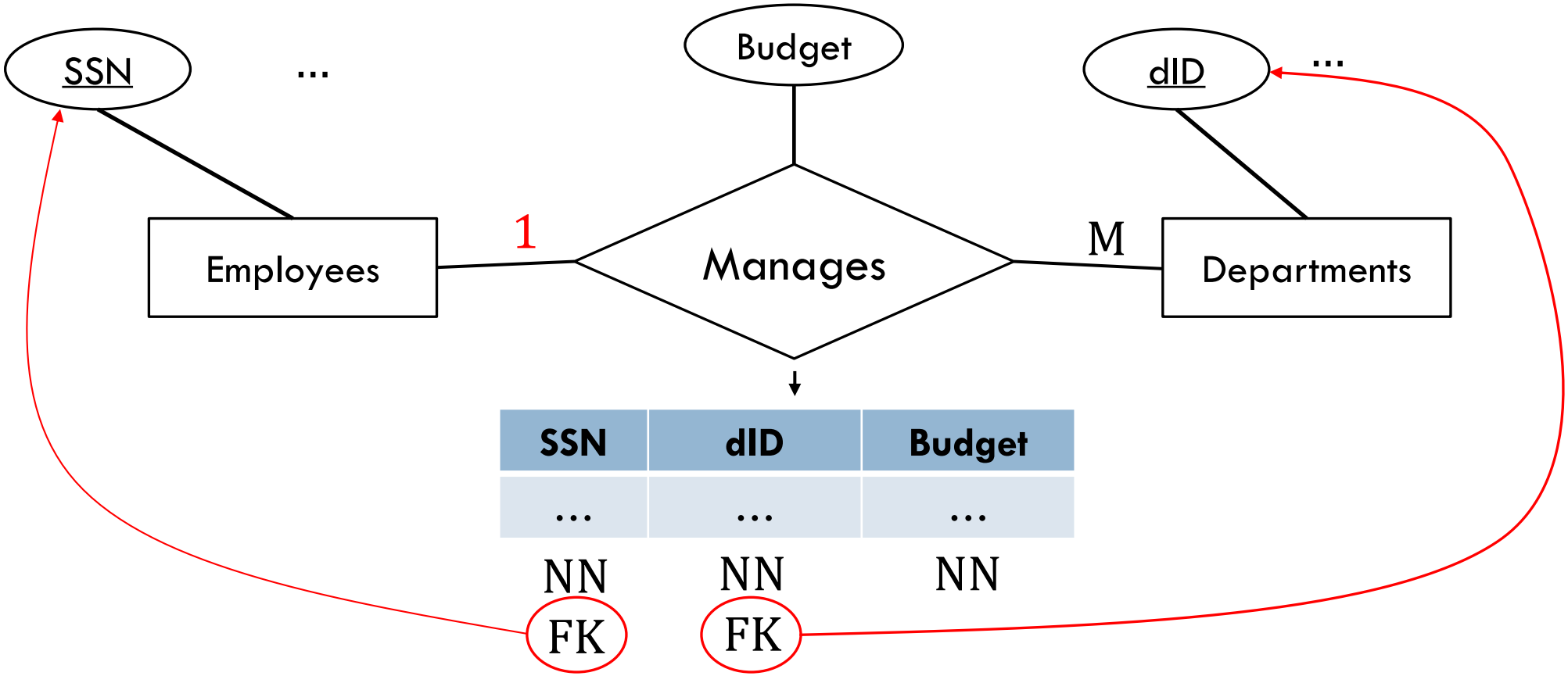
Relationship Sets to Schema

- **1-to-Many**
  - Department can only have 1 manager



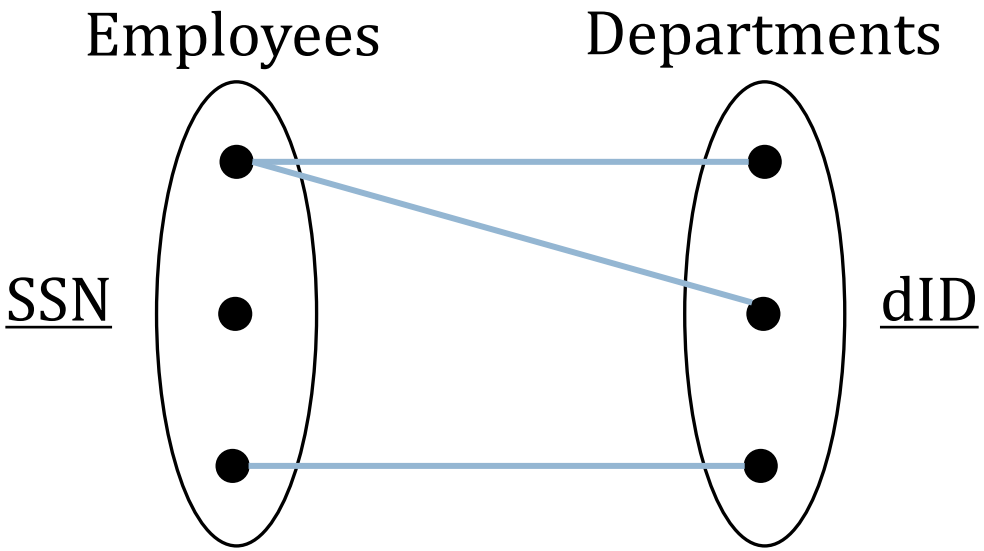
Relationship Sets to Schema

- **1-to-Many (first attempt)**
  - New schema for the relationship
  - Primary key of related entities as foreign keys



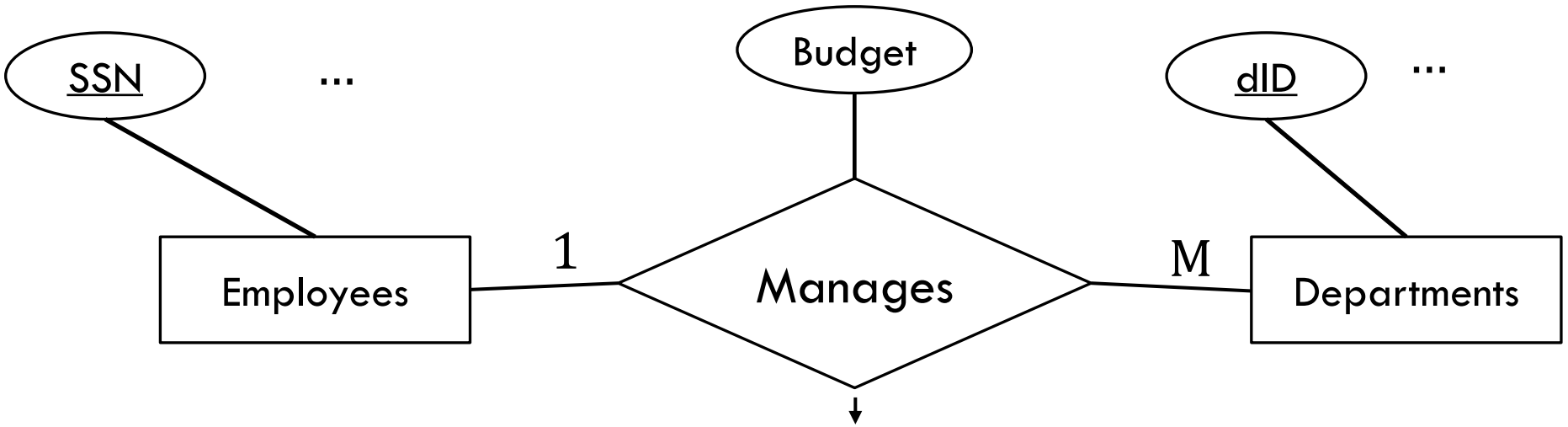
Relationship Sets to Schema

- How do we uniquely identify a line (1-to-M)?
  - "a dID can only appear once in Manages table"



Relationship Sets to Schema

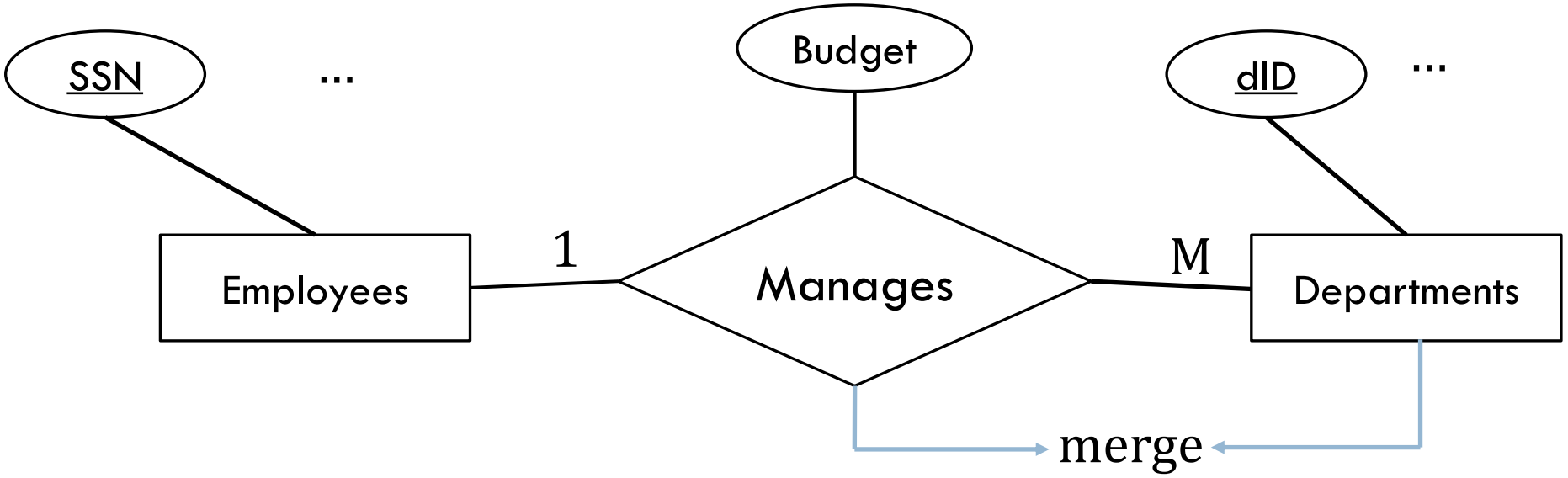
- **1-to-Many (first attempt)**
  - Key is the foreign key from the 'M' side



SSN	dID	Budget
...	...	...
NN	NN	NN
FK	FK	
	PK	

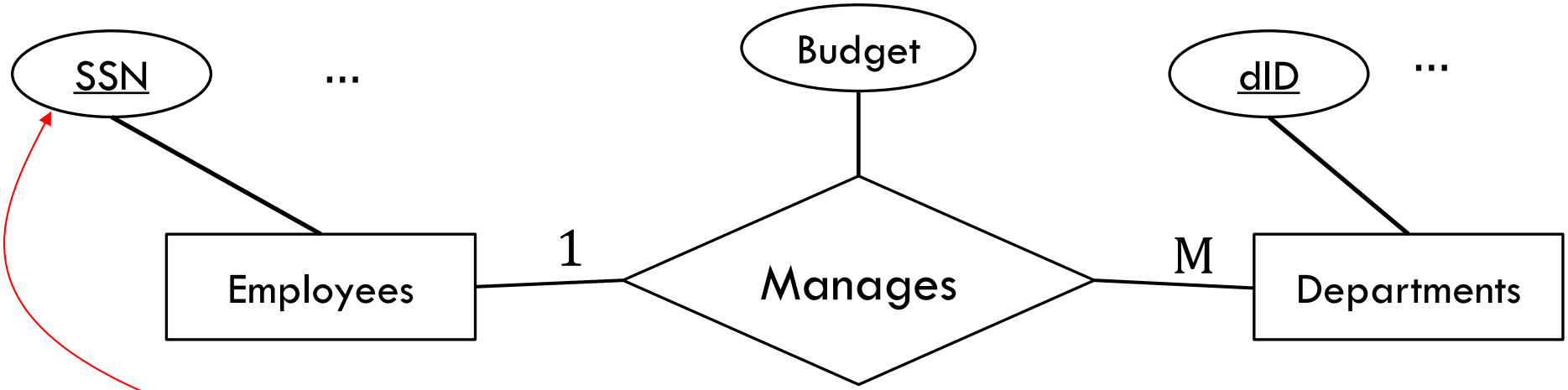
Relationship Sets to Schema

- **1-to-Many (better option)**
  - Department can only have 1 manager
  - Merge the relationship into the Entity



Relationship Sets to Schema

- **1-to-Many (better option)**
  - Department can only have 1 manager

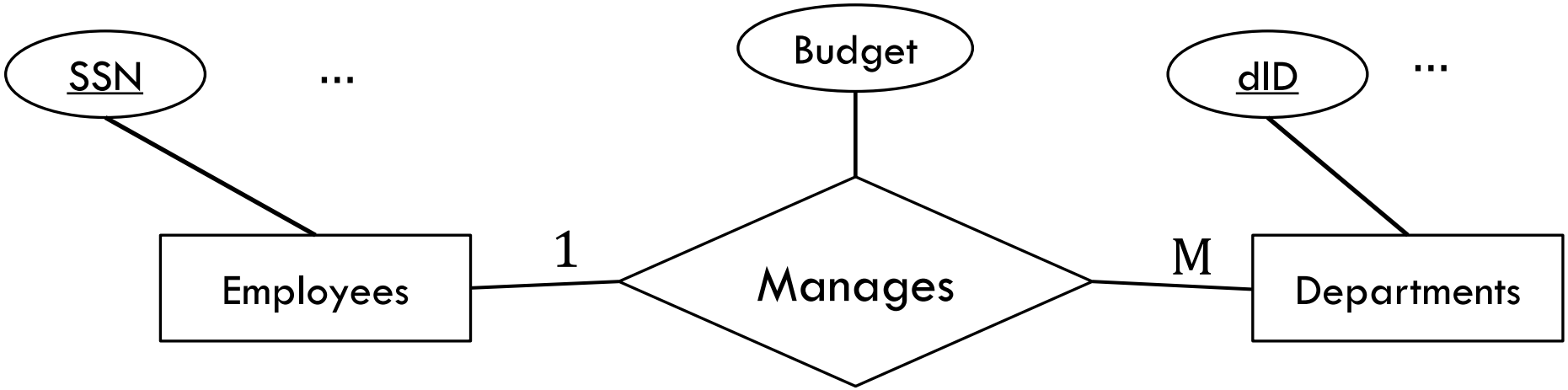


SSN	Name	Pay
...	...	...
PK NN	NN	NN

dID	dName	Mnger	Budget
...	...		
PK NN	NN	FK	

Relationship Sets to Schema

- **1-to-Many (better option)**
  - Can `Mnger` and `Budget` be NULL?

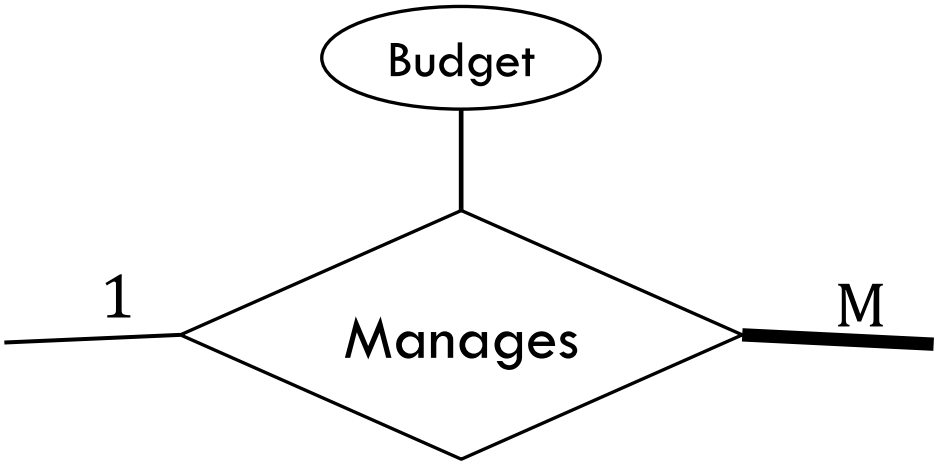


SSN	Name	Pay
...	...	...
PK NN	NN	NN

dID	dName	Mnger	Budget
...	...		
PK NN	NN	FK NN	NN

Participation Constraints

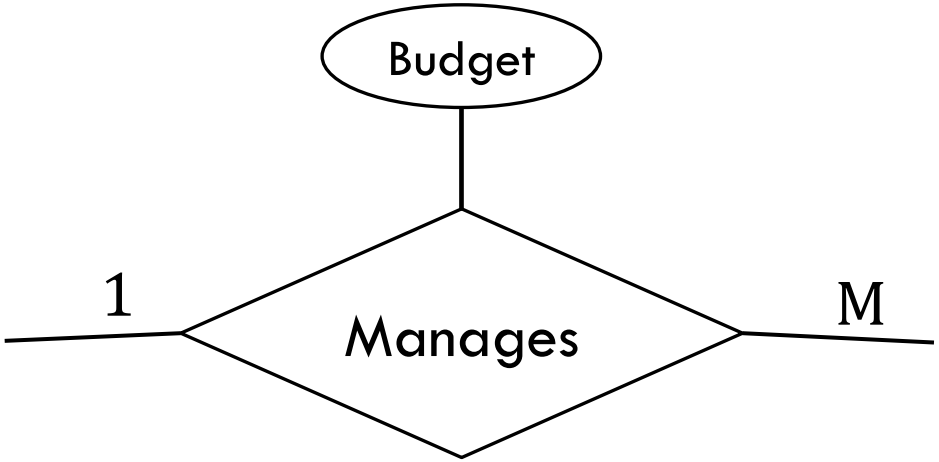
- If participation is required, set `NOT NULL`
- Else, `NULL` is allowed



…	Mnger	Budget
…		

FK NN

NN



…	Mnger	Budget
…		

FK

Foreign Key + NULL

- Referential integrity only enforced when inserting non-null data
- Not all departments have a manager

Employees

SSN	Name	Pay
1	Joe	…
2	Steve	…
3	Meg	…

Departments

dID	dName	Budget	Mnger
a	Chem	…	1
b	Phys	NULL	NULL
c	CS	…	3

FK

Naïve Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Reduced

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN	CheckedOutBy
1001	978-0590353427	1
1002	978-0590353427	NULL
1003	978-0679732242	NULL
1004	978-0394823379	1
1005	978-0394823379	4
1006	978-0062278791	4

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

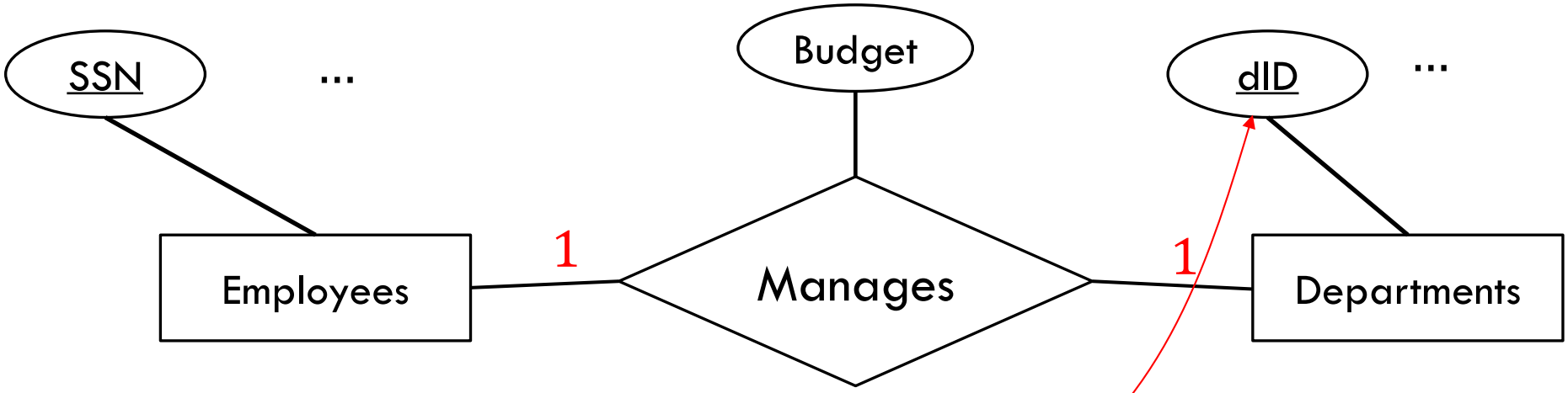
ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Performance

- In general:
  - Fewer joins = better performance
  - Fewer tables = fewer joins
- 1-to-M and M-to-1 don’t need their own tables

Relationship Sets to Schema

- **1-to-1**
  - Treat as 1-to-M
  - Merge relationship into one of the other tables

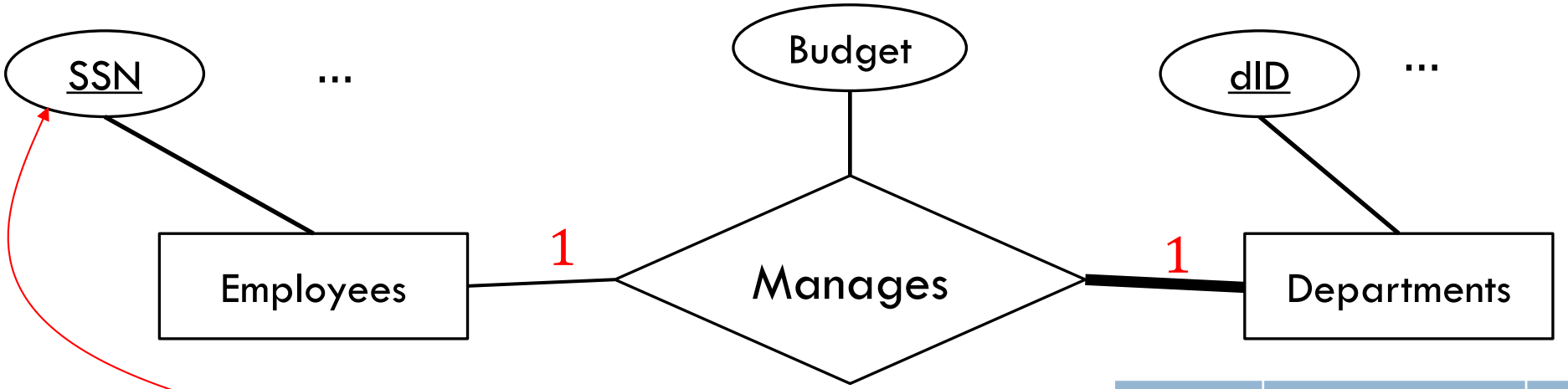


SSN	…	Manages	Budget
…	…	…	…
PK		FK	
NN		UNIQUE	

dID	…
…	…

Relationship Sets to Schema

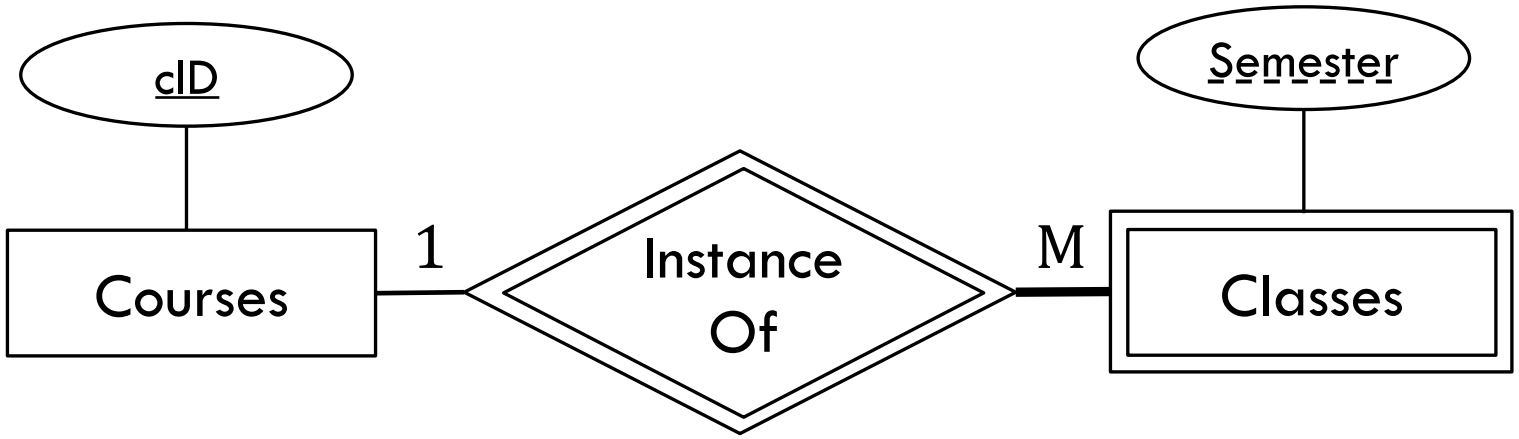
- **1-to-1**
  - If participation required on one side, use that side
  - Add `NOT NULL`



dID	ManagedBy	Budget
...	...	...
PK NN	FK UNIQUE NN	NN

Relationship Sets to Schema

- **Supporting Relationship**
  - A supporting relationship is 1-to-M
  - Same procedure, except supporting key is combined with partial key



Semester	cID	…
…	…	…
NN	NN FK	

-------PK-------

Double Participation

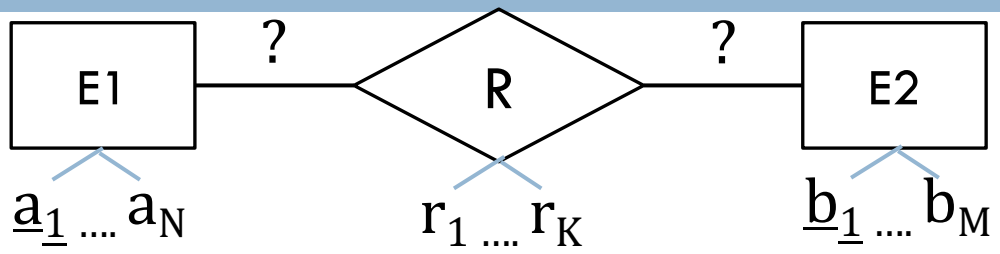
- If participation required on **both** sides…
  - Regardless of cardinality

Double Participation

- If participation required on **both** sides…
  - Regardless of cardinality
  - Chicken/egg problem

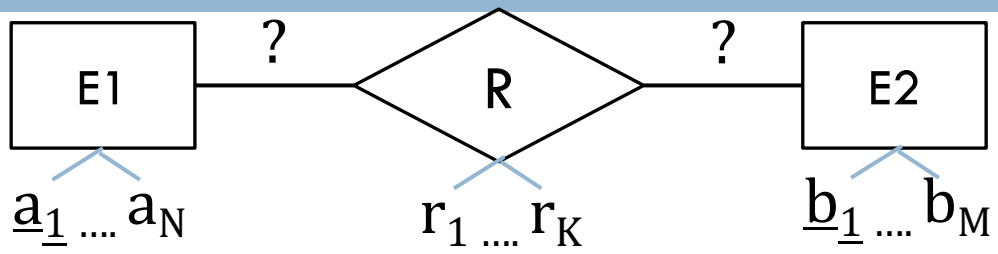
- Difficult to capture with schema design
  - Instead, enforce with SQL commands

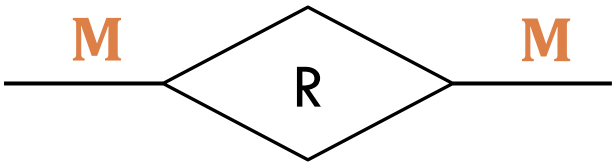
Algorithm Summary



Algorithm Summary

■ = FK



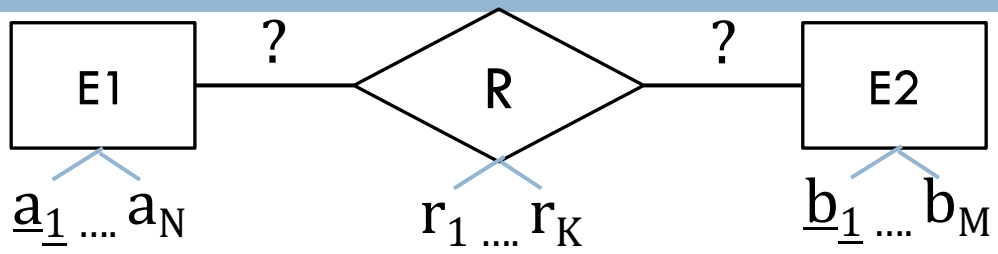


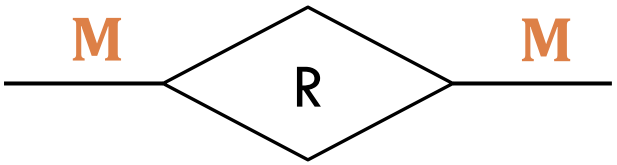
$E1 = \{\underline{a_1}, \ldots, a_N\}$ $E2 = \{\underline{b_1}, \ldots, b_N\}$

$R = \{\underline{a_1, b_1}, r_1, \ldots r_k\}$

Algorithm Summary

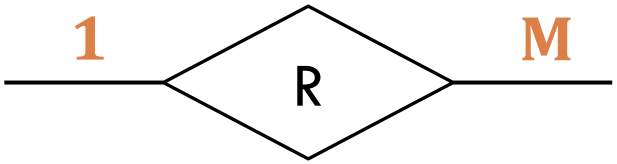
■ = FK





$E1 = \{\underline{a_1}, \ldots, a_N\}$ $E2 = \{\underline{b_1}, \ldots, b_N\}$

$R = \{\underline{a_1}, \underline{b_1}, r_1, \ldots r_k\}$

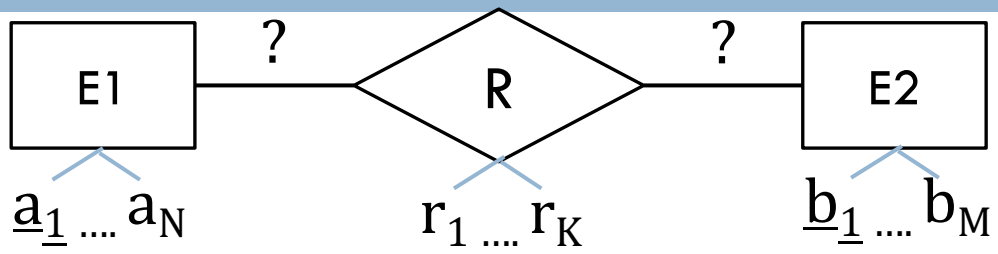


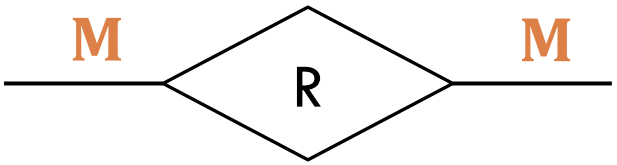
$E1 = \{\underline{a_1}, \ldots, a_N\}$

$E2 = \{\underline{b_1}, \ldots, b_N, a_1, r_1, \ldots r_k\}$

Algorithm Summary

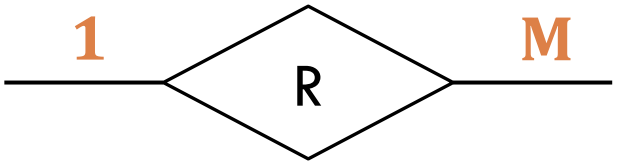
■ = FK





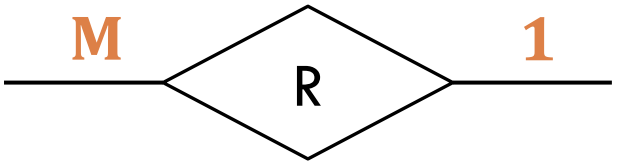
$E1 = \{\underline{a_1}, \ldots, a_N\}$ $E2 = \{\underline{b_1}, \ldots, b_N\}$

$R = \{\underline{a_1}, \underline{b_1}, r_1, \ldots r_k\}$



$E1 = \{\underline{a_1}, \ldots, a_N\}$

$E2 = \{\underline{b_1}, \ldots, b_N, a_1, r_1, \ldots r_k\}$

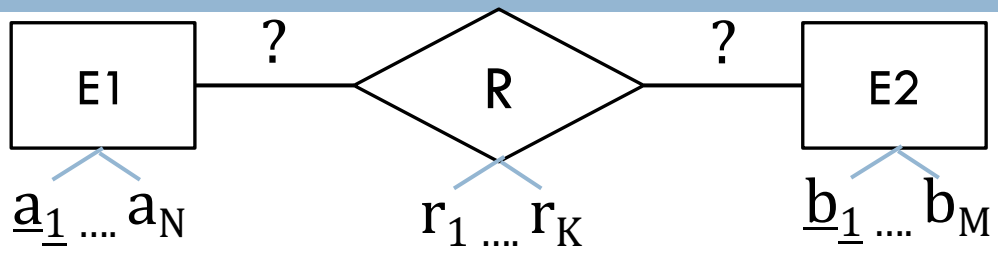


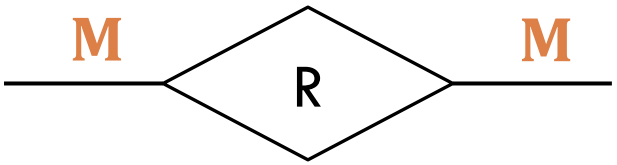
$E1 = \{\underline{a_1}, \ldots, a_N, b_1, r_1, \ldots r_k\}$

$E2 = \{\underline{b_1}, \ldots, b_N\}$

Algorithm Summary

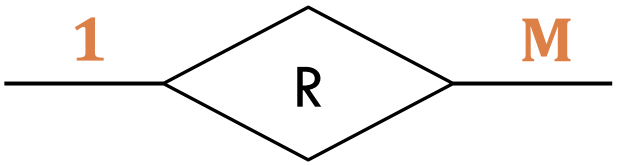
■ = FK





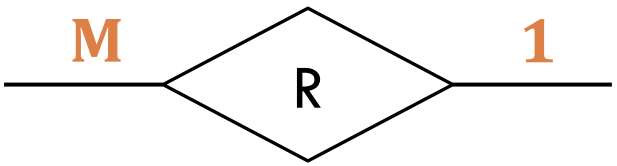
E1 = {a_1, …, a_N} E2 = {b_1, …, b_N}

R = {a_1, b_1, r_1, … r_k}



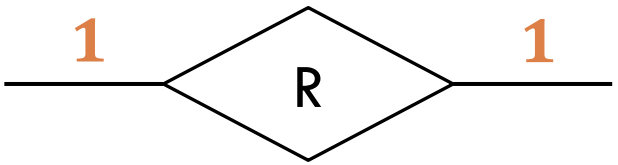
E1 = {a_1, …, a_N}

E2 = {b_1, …, b_N, a_1, r_1, … r_k}



E1 = {a_1, …, a_N, b_1, r_1, … r_k}

E2 = {b_1, …, b_N}



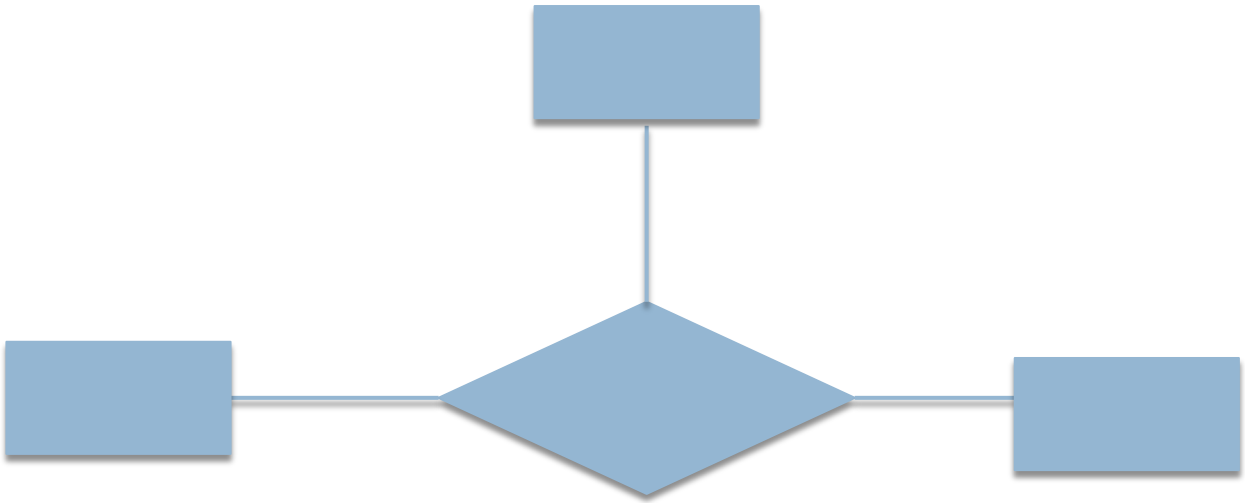
Treat as 1:M or M:1

Mark foreign key as unique

Algorithm Summary

- `NOT NULL` determined by participation constraints
- Total participation on both sides – don't capture with schema

Relationships with Arity > 2



Relation might be “1 to 1 to M” or “1 to M to M”:

If so, can add a foreign key to that entity set’s table

Otherwise, create a table for the relationship with foreign keys for each participating entity