Database Management Systems
L3

Umass Boston
Summer 2023
Cristina Maier

Part of slides are based on “Database Management System, 3rd ed., by Ramakrishnan and Gehrke

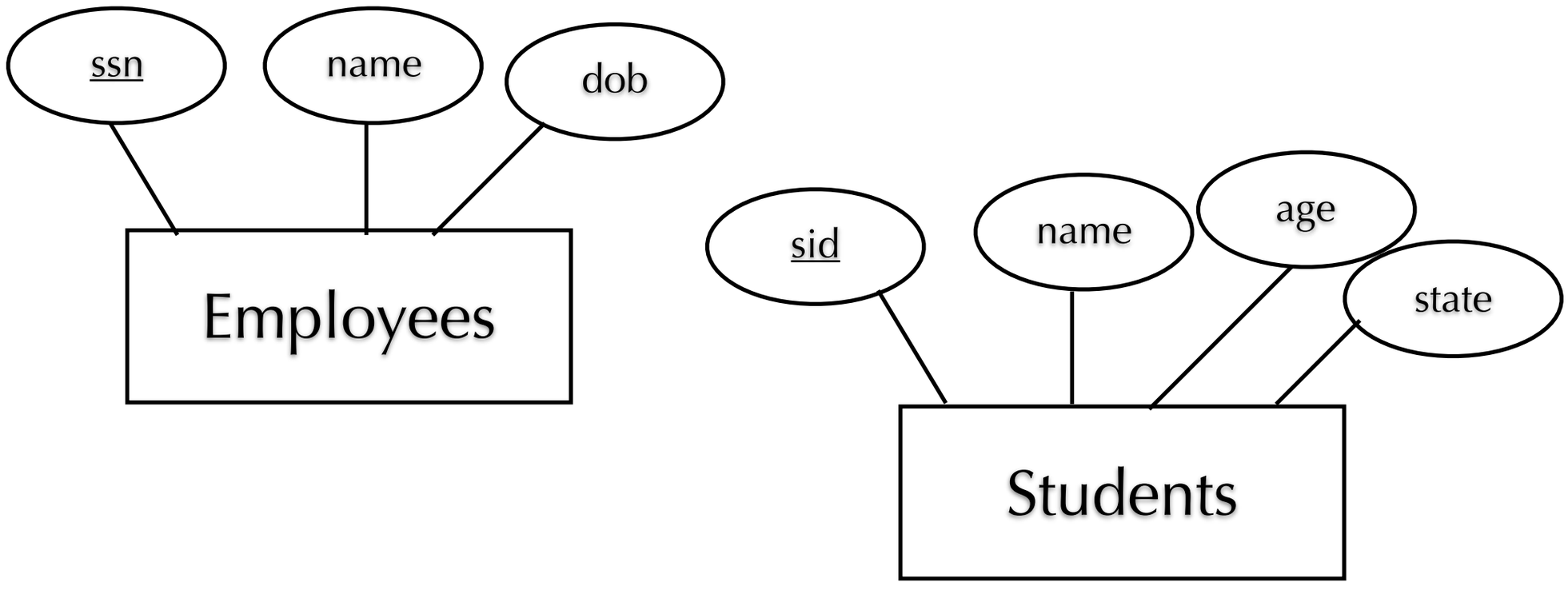
Topics

- Introduction to DBMS
- Relational Data Model
- *Relational Algebra*
- Conceptual Design: the Entity-Relationship Model
- Structured Query Language (SQL)
- Application Development (Java, Python)
- Schema Refinement and Normal Forms
- Database Security and Authorization
- Some NoSQL topics (If time permitted)

Representation (recap.)

- Entity set: represented as rectangle
- An attribute: represented as an oval
- A key: underlined attribute(s) name
- Edges: used to connect an entity set to its attributes

Example of Employees and Students Entity Sets (recap.)



Relationship and Relationship Set

- Relationship: association among two or more entities
  - A relationship can have descriptive attributes
    - But relationships must be determined only by entities
  - E.g.: Mary is a student at Umass Boston
    - Since 09/06/2022
- Relationship set: collection of similar relationships
- A relationship set can be seen as a set of n-tuples where $\{(e_1, e_2, \ldots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$, where $E_i, i \in [1,n]$ is an entity set
- Each n-tuple denotes a relationship involving n Entities Sets
- If a relationship between two entity sets, we have a binary relationship(2-tuples); three entity sets: ternary relationship (3-tuple)
- Several relationship sets might involve the same entity set

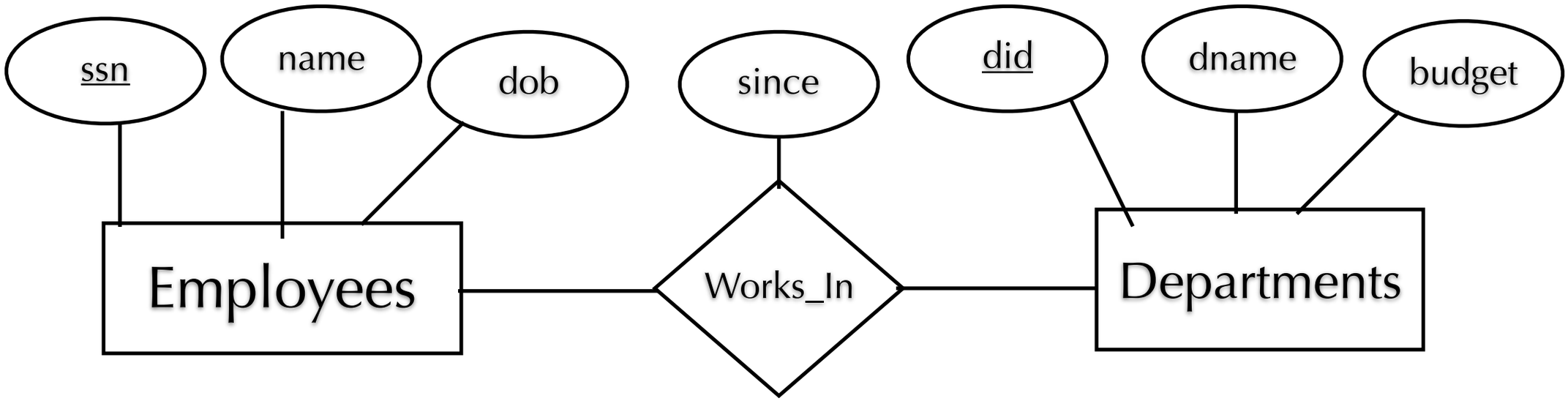
Representation

- A relationship set is represented by a diamond

- A relationship can have descriptive attributes, represented by ovals

- Edges connect a relationship set to entity sets. Edges also connect relationship sets to relationship sets attributes

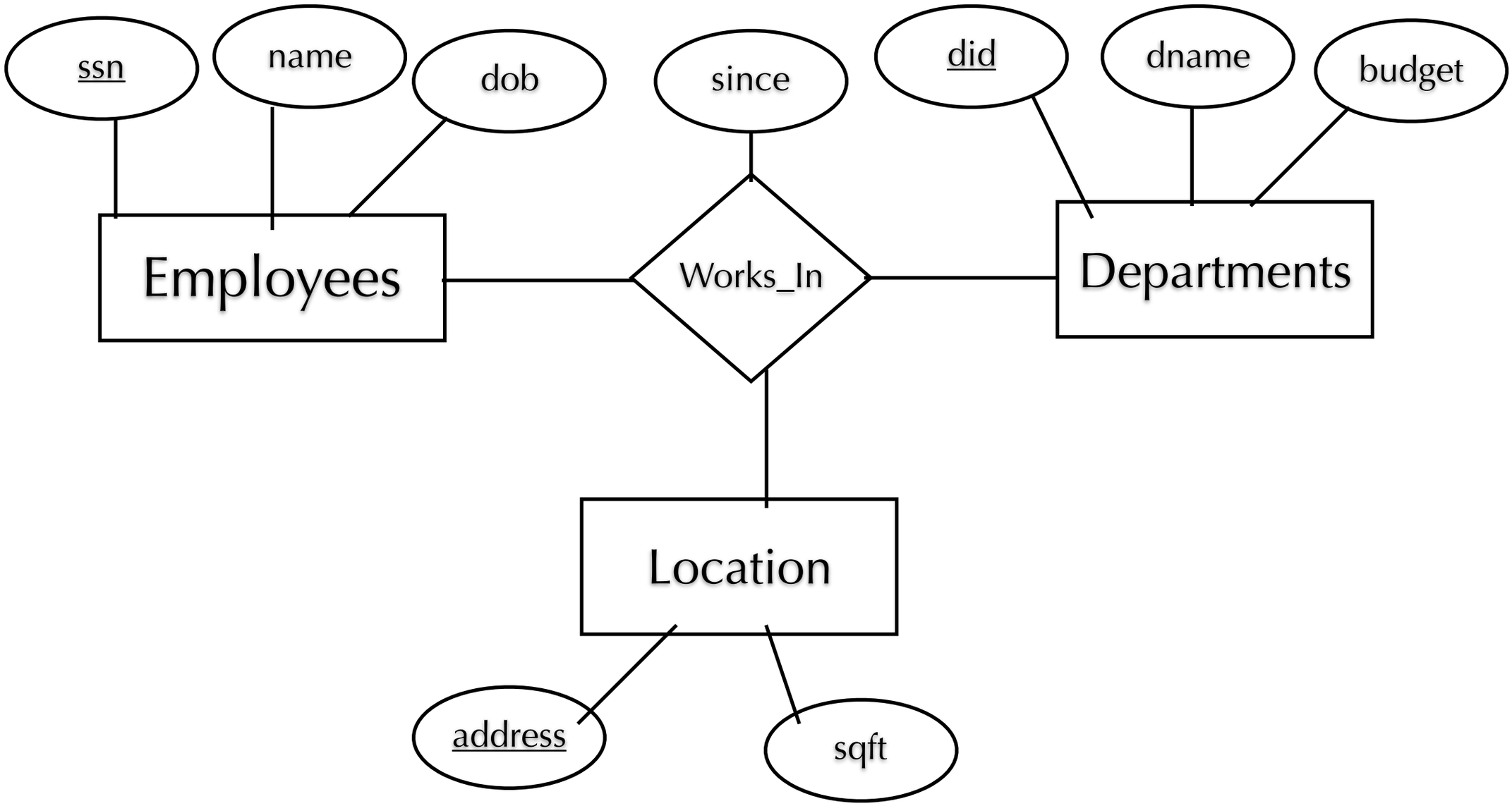
Example 1


ssn
name
dob
since
did
dname
budget
Employees
Works_In
Departments

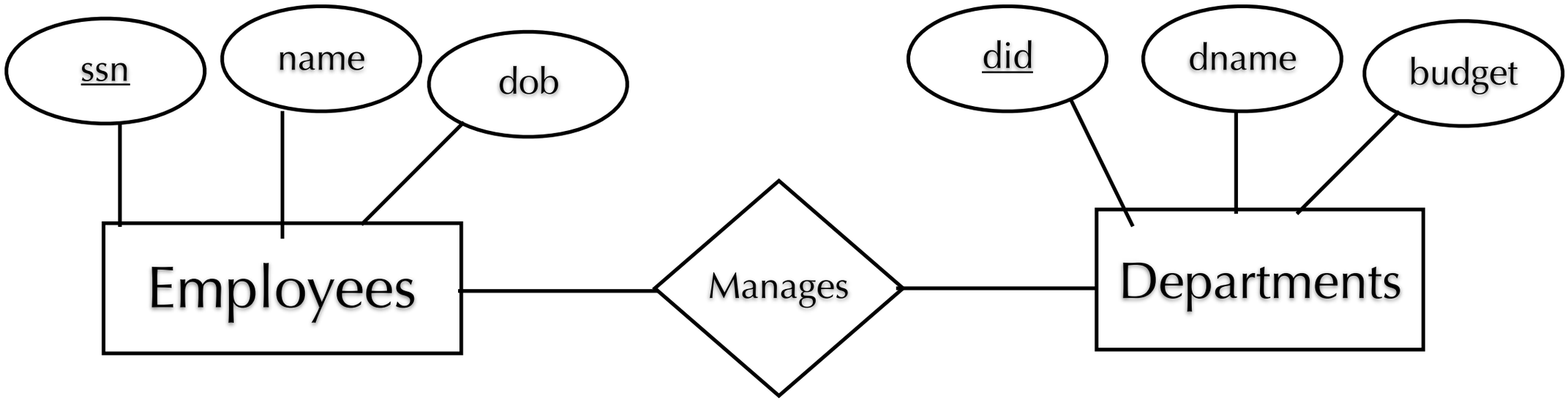
Relationship set

- A relationship must be uniquely identified by the entities, without referencing the relationship descriptive attributes
  - E.g. In the Works_In relationship set, a Works_In relationship is uniquely identified by a combination of employee ssn and the department did
    - This means for a given employee and department pair, we cannot have more than one associated ‘since’
- Instance of a relationship set: a ‘snapshot’ of relationship set at a given point in time

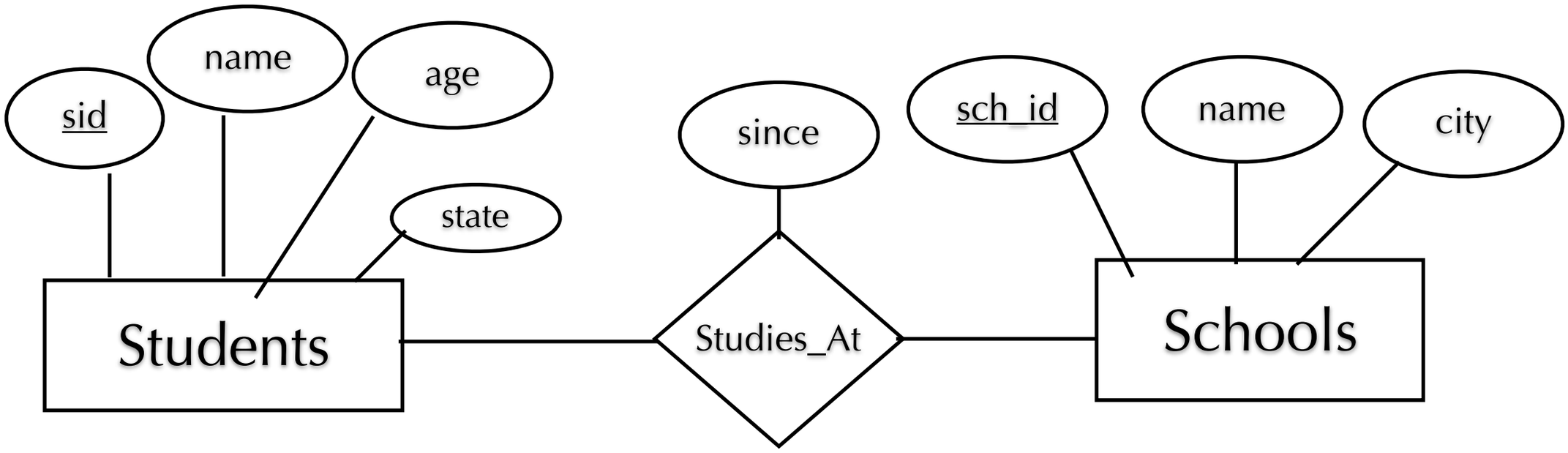
Example 2 (Ternary Relationship Set)



Example 3

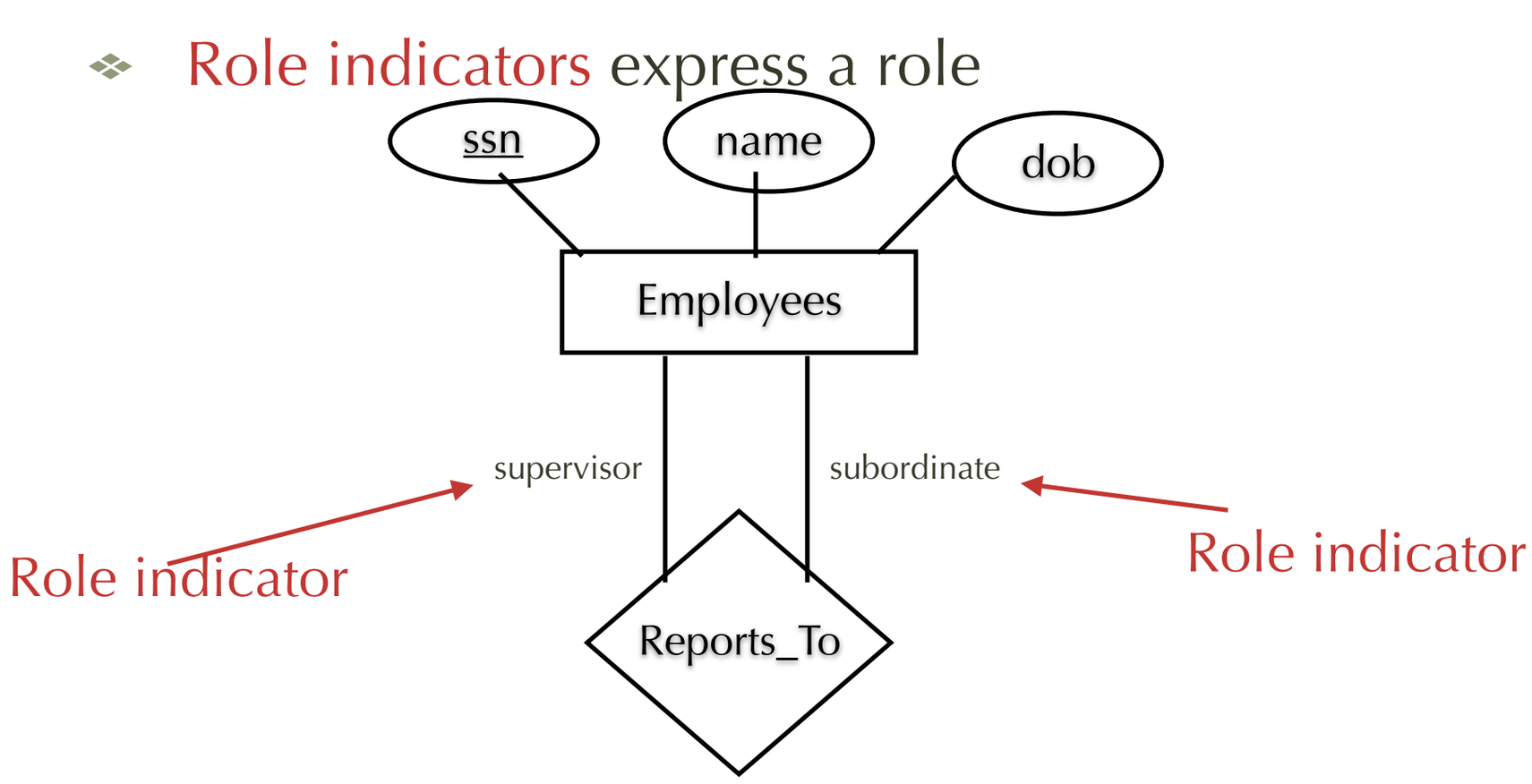

ssn
name
dob
Employees
Manages
did
dname
budget
Departments

Example 4

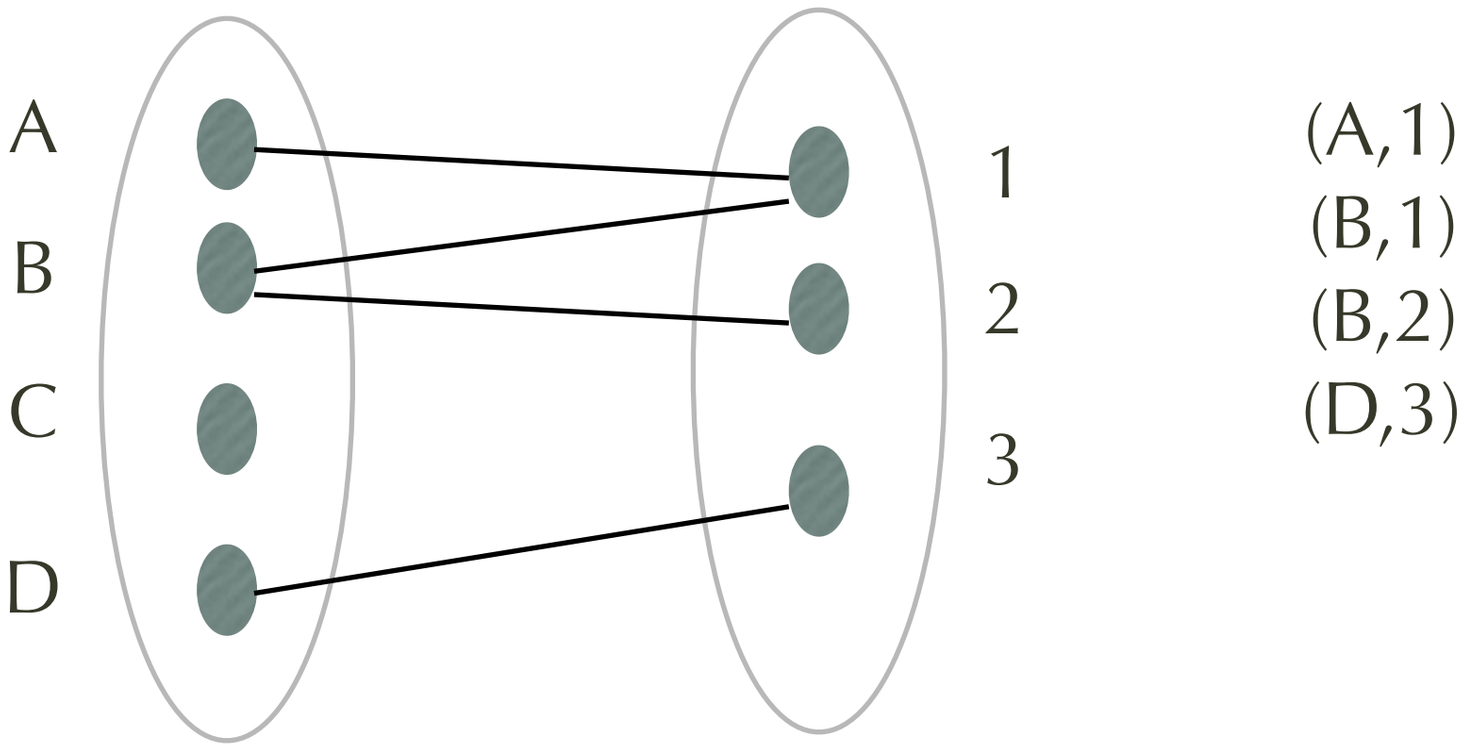

sid
name
age
state
Students
since
Studies_At
sch_id
name
city
Schools

A special case of relationship

- An entity set can participate in a relationship with itself
- Each entity set in the same relationship set plays a different role
- Role indicators express a role



Visualizing Relationships



(A,1)
(B,1)
(B,2)
(D,3)

Edge= Relationship
Edges=relationship set instance

Constraints

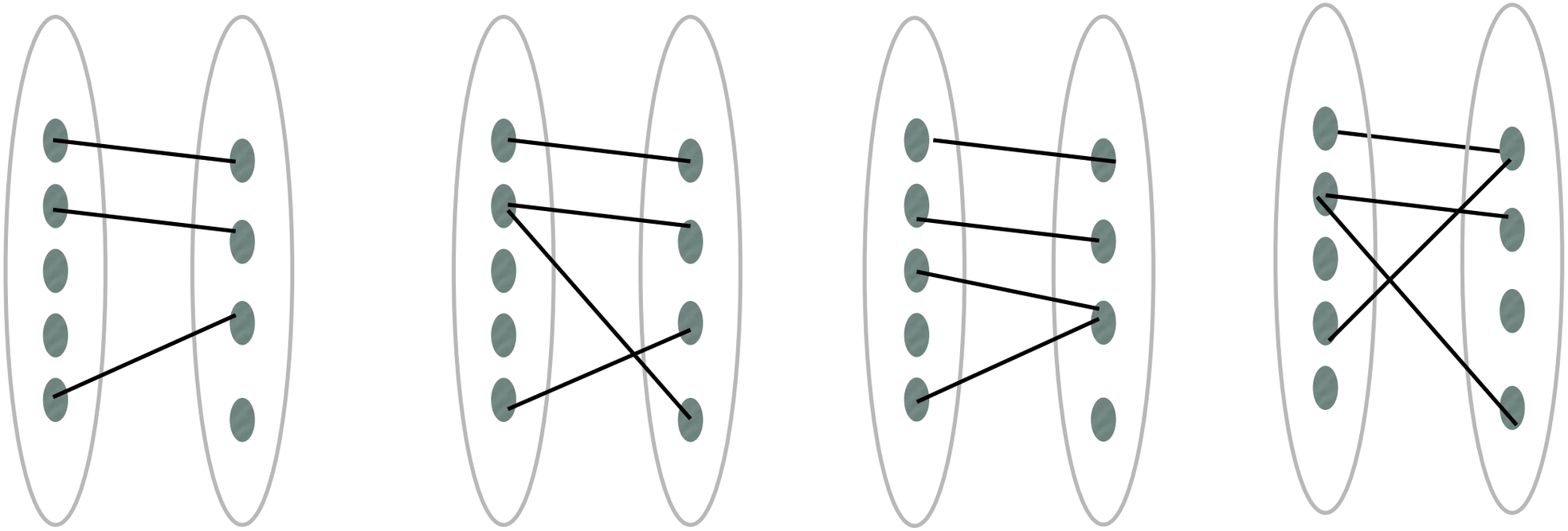
- To an ER model, we can also add constraints, such as:
  - Key constraints
  - Participation constraints

Key constraints

- Given two entity sets, E_1 and E_2. How many entities from E_2 can an entity from E_1 have a relationship with? How many entities from E_1 can an entity from E_2 have a relationship with?

- This is referred to as a relationship multiplicity

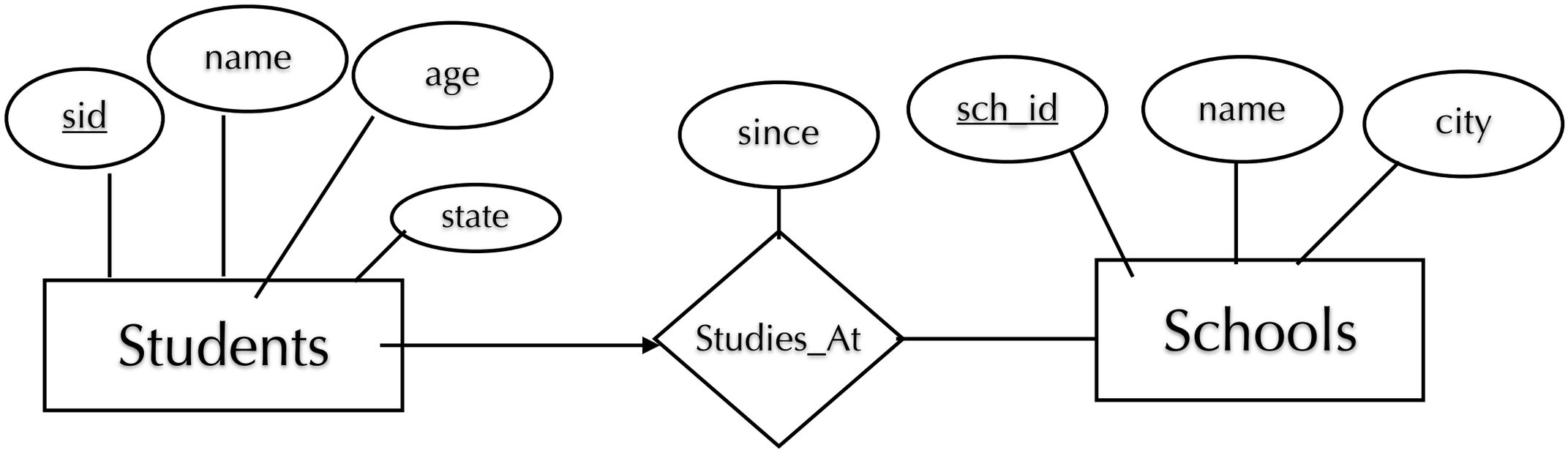
Relationship multiplicity



1-to-1 1-to-Many Many-to-1 Many-to-Many

Example

Constraint: A student can be student at a most one school.
More students can study at the same school.

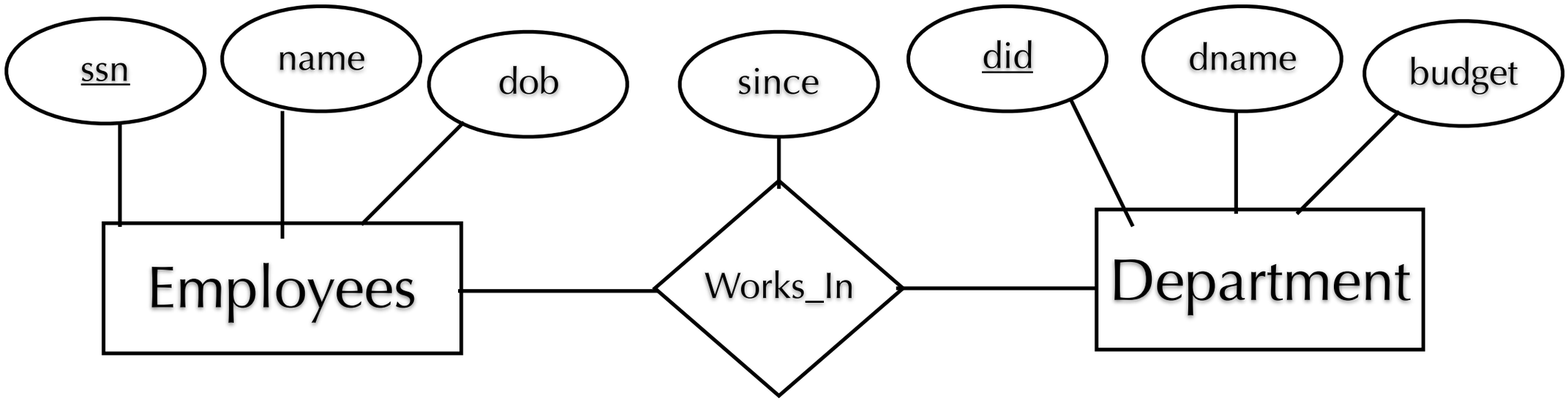


From Students to Schools we have Many-to-1
From Schools to Students we have 1-to-Many

Note where we use the arrow

Many-to-Many Example

An employee can work in multiple departments.
A department can have more employees.

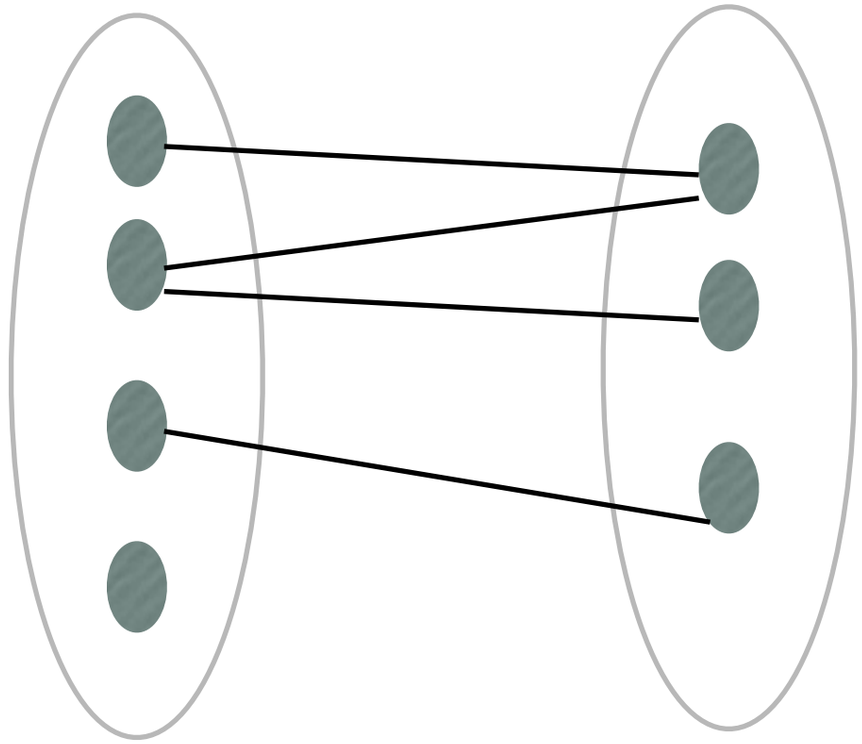


Participation Constraints

- Total Participation
  - Represented by thickened edge
- Partial Participation
  - Represented by regular (i.e. non-thickened) edge

Visualizing Relationships

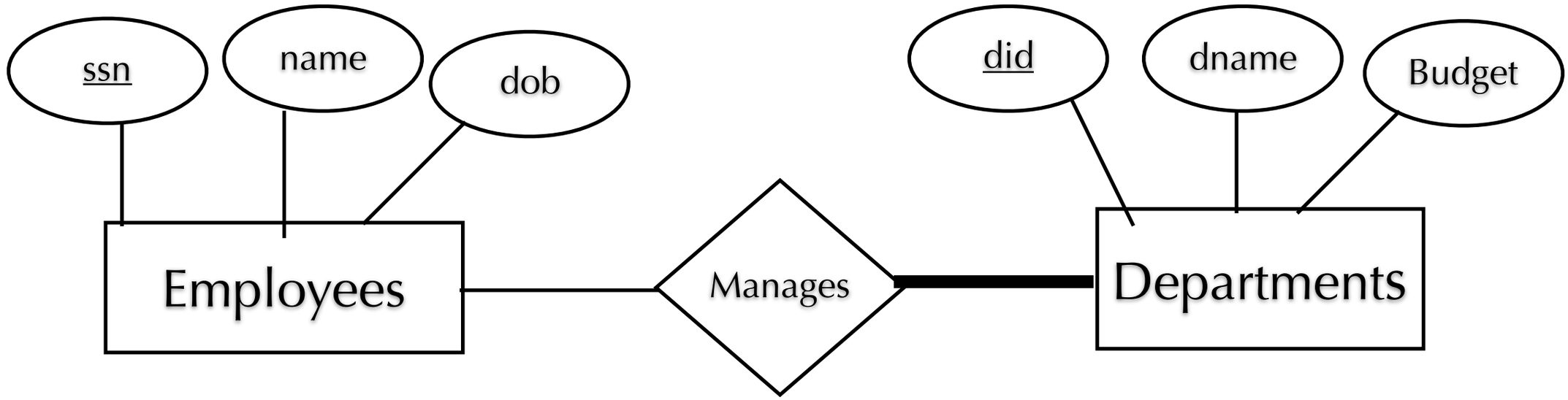
Partial



Total

Example 1

Participation Constraint: Each department must have at least one manager (total).
Not all employees are managers(partial)



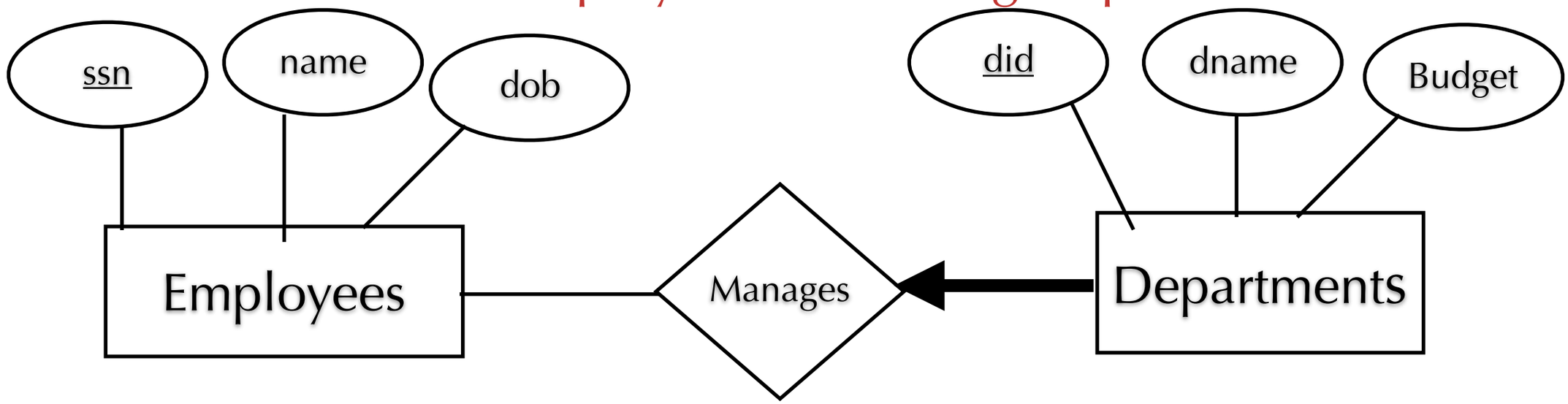
Between Departments And Manages
We use a thickened edge because
each department has at least one manager (total participation)

Participation Constraints

- A key constraint on Manages tells us that a department has at least one manager
- Does a department requires to have a manager?
  - Yes. This type of requirement is called a participation constraint. In such a case the participation is total
  - Otherwise, the participation is partial

Example 2

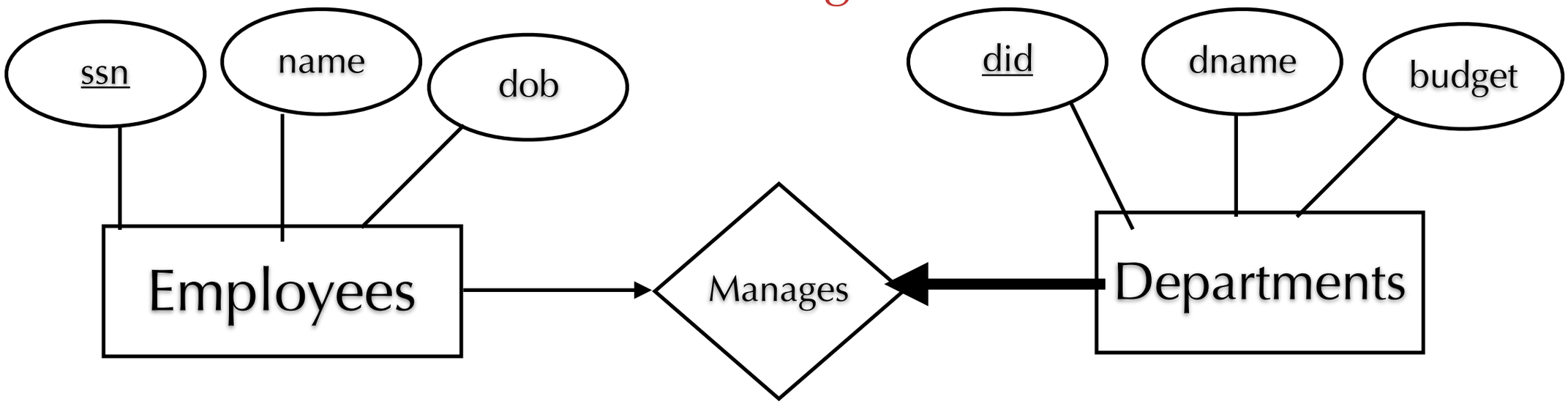
Constraint: Each department must have exactly one manager (total).
Not all employees are managers(partial)



Between Departments And Manages
we use thickened arrow because each department has one manager.

Example 3

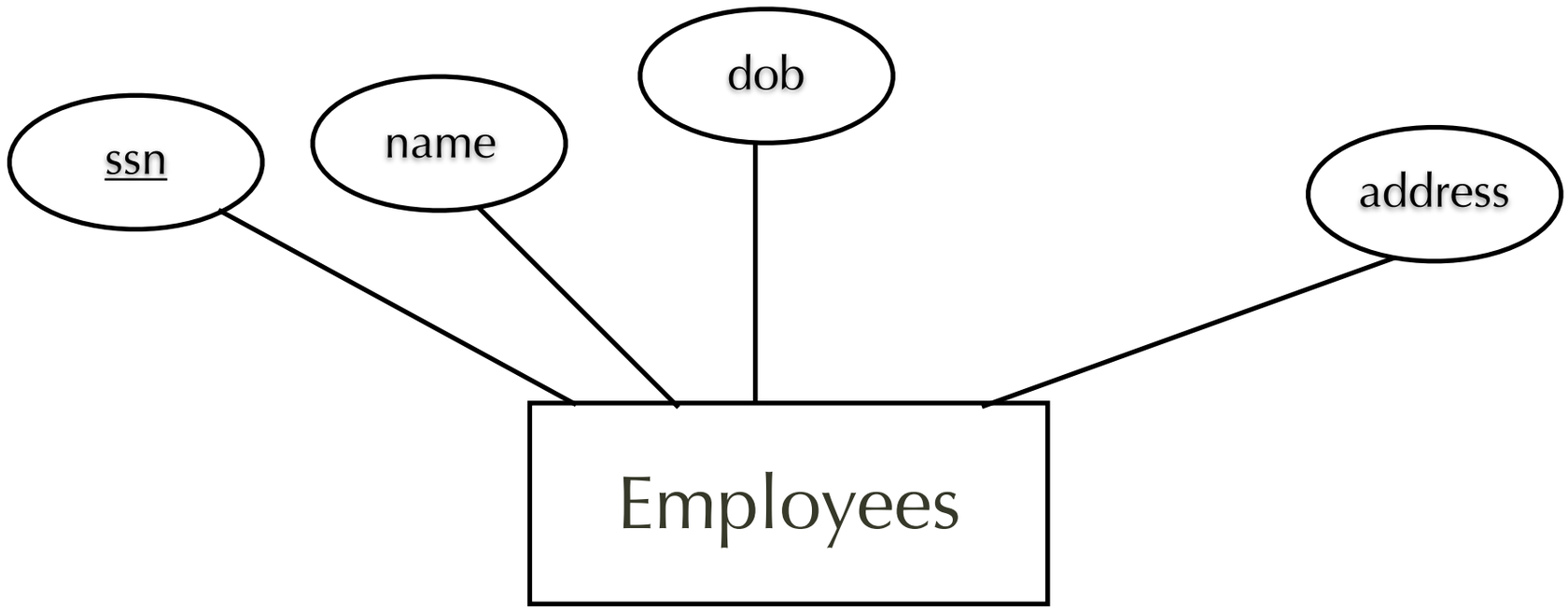
Constraint: An employee can be a manager of at most one department. One department must have exactly one manager



Design Choices in ER Model

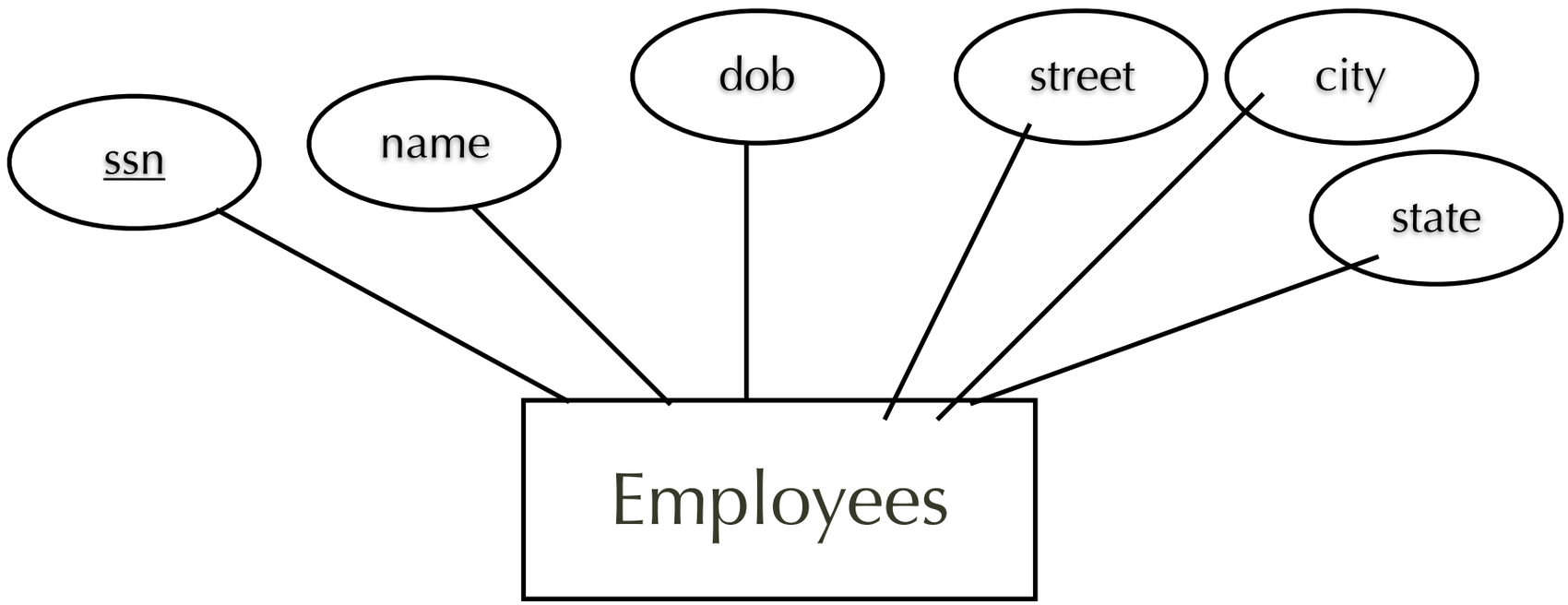
- Should a concept be modeled as an attribute or as an entity set?
  - E.g.: adding the address of an employee
    - Add one attribute address (appropriate if the employee has only one address). If employee has more addresses, this solution will not work
    - (works with multiple addresses per employee) Address modeled as a different entity set. Add a relationship set Lives_At

Example
-with one attribute address

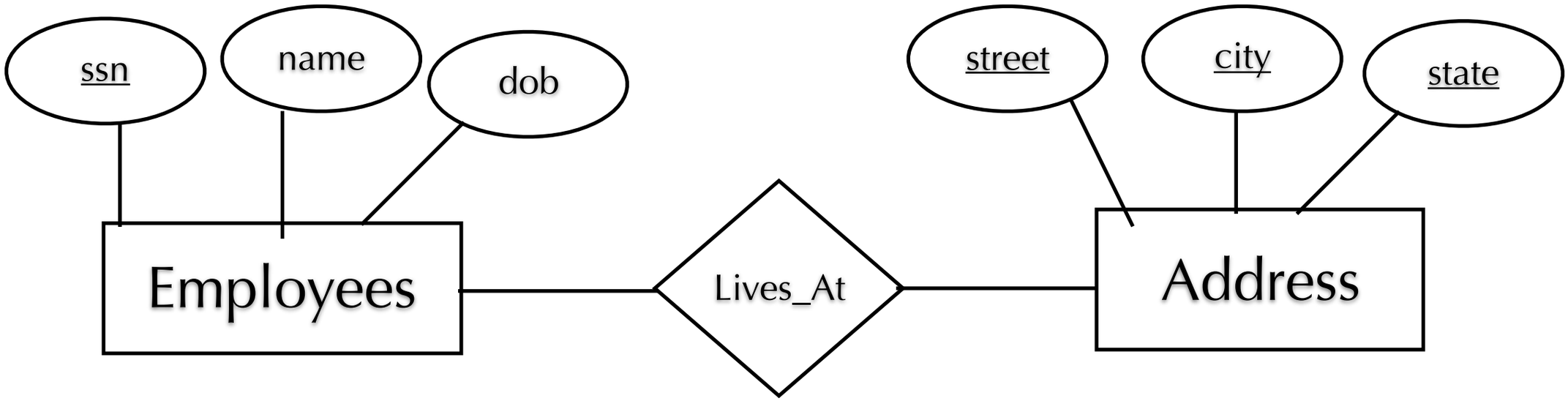


Example

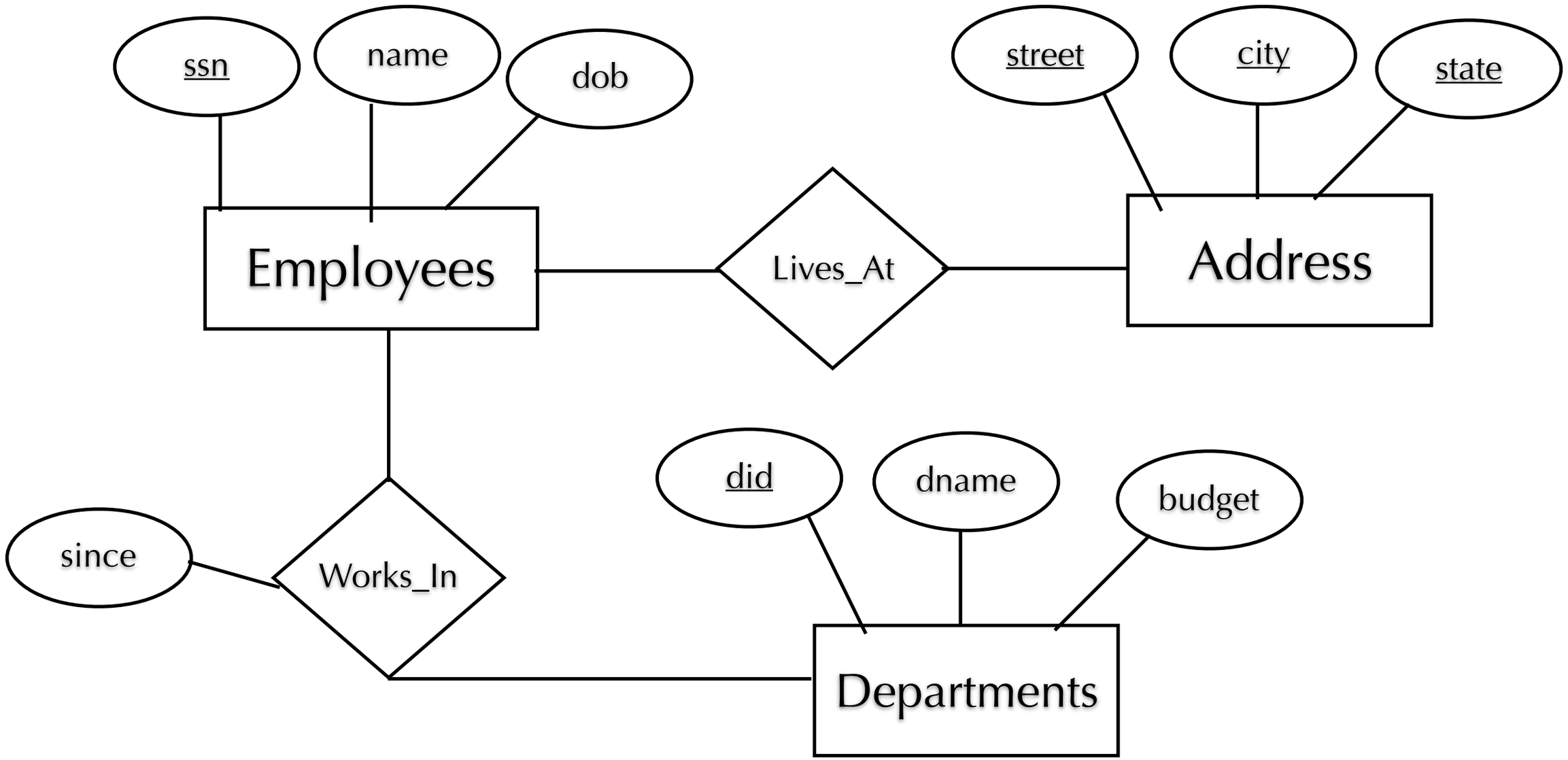
- if we want to be able to be able to search by state, city, or state, we split the attribute address into three attributes



Example
- if we want to allow an employee to have multiple addresses



Example with multiple entity sets and relationship sets

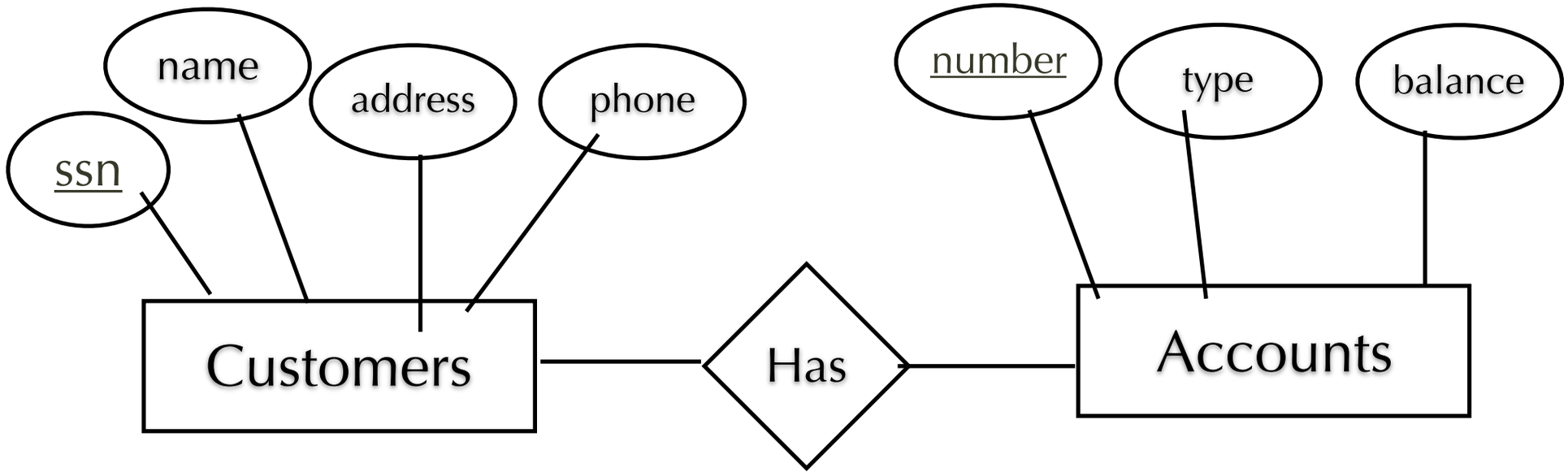


Design Exercise

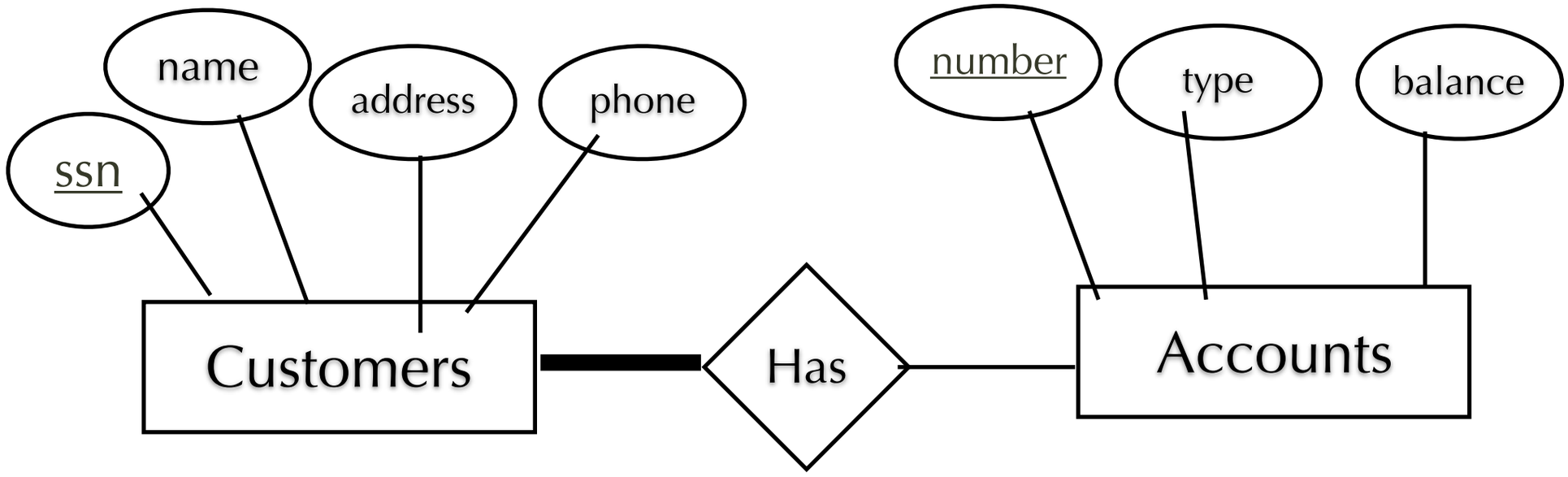
- Design a database for a bank, including information about customers and their accounts. Information about customers include their name, address, phone and SSN. Accounts have numbers, types (e.g. savings, checking) and balances.
  - a)Draw the ER diagram for this database
  - b)Modify the ER diagram such that each customer must have at least one account
  - c)Modify the diagram further, such that each account must have at most one customer
  - d)Modify the diagram such that each account must have exactly one customer
  - e) Modify the diagram further, to extend the database to include information about the bank (bankid, bname, baddress) the account is part of. Note that an account is part of exactly one bank, and one bank can contain multiple accounts .

Design Example

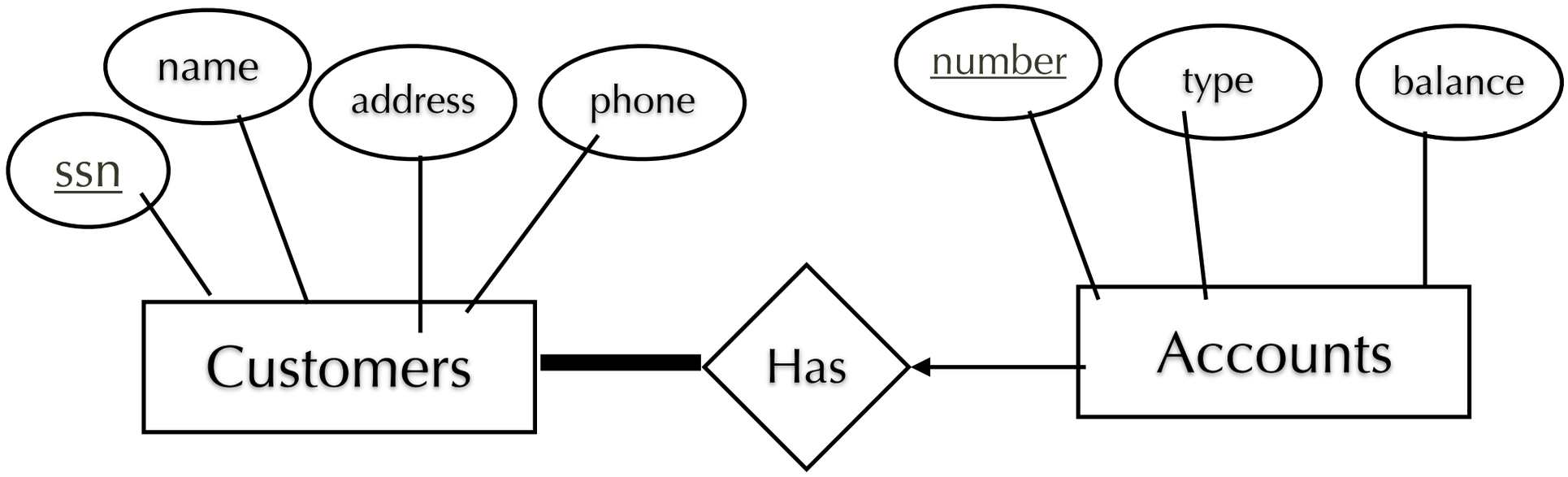
a)



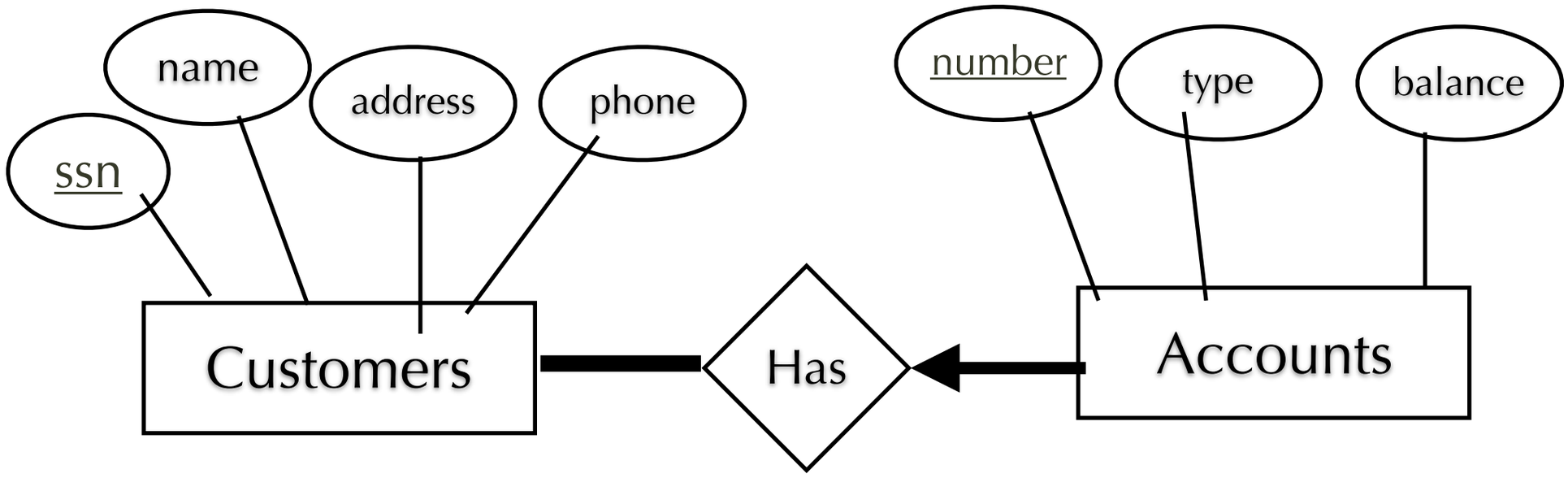
Design Example
b)



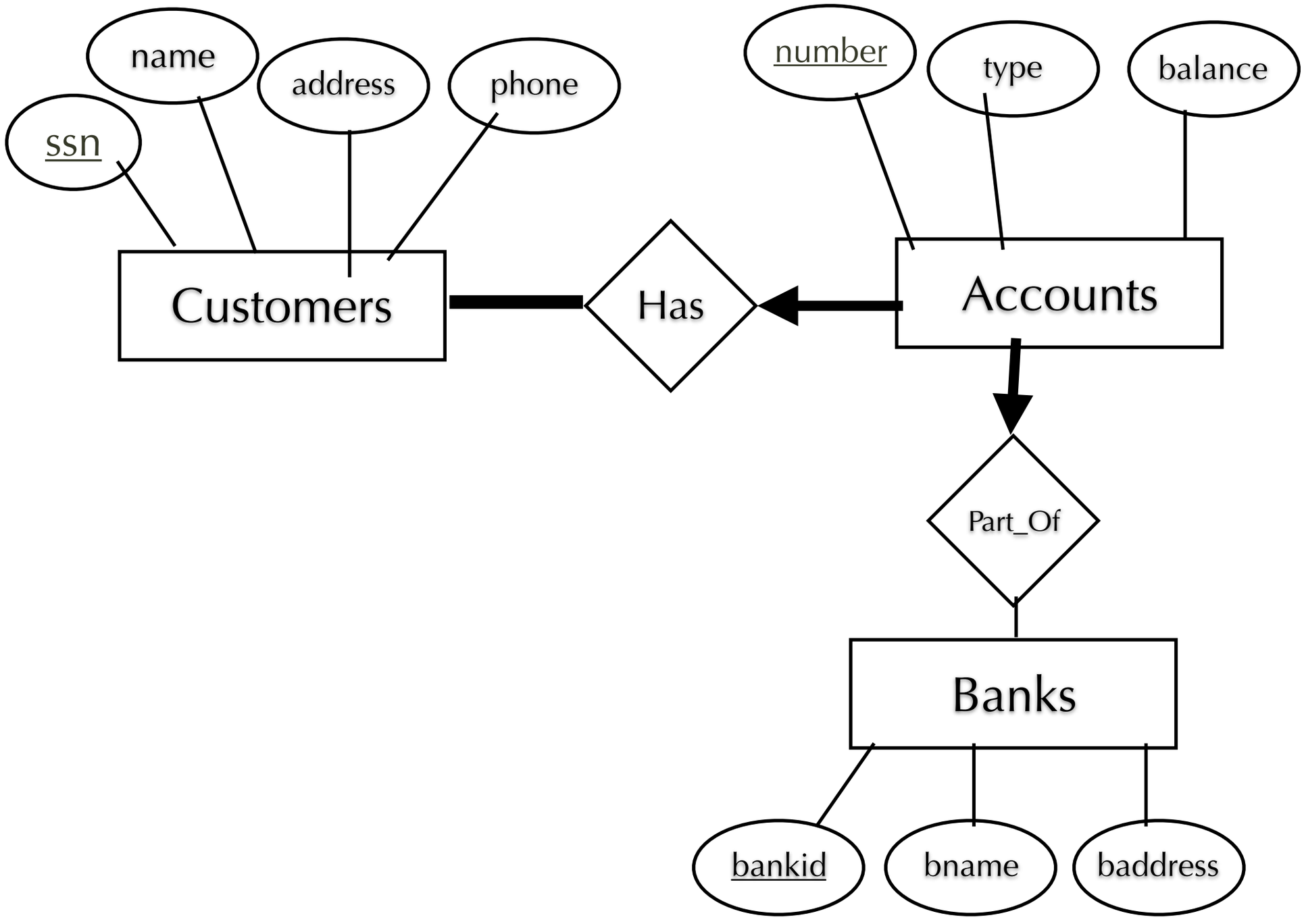
Design Example
c)



Design Example
d)



e)


name
address
phone
ssn
number
type
balance
Customers
Has
Accounts
Part_Of
Banks
bankid
bname
baddress

Create DB Schema for the above example?

Database Schema

- Customers(ssn: string,name:string, address:string, phone:string)

- Accounts(number: int, type:string, balance:real)

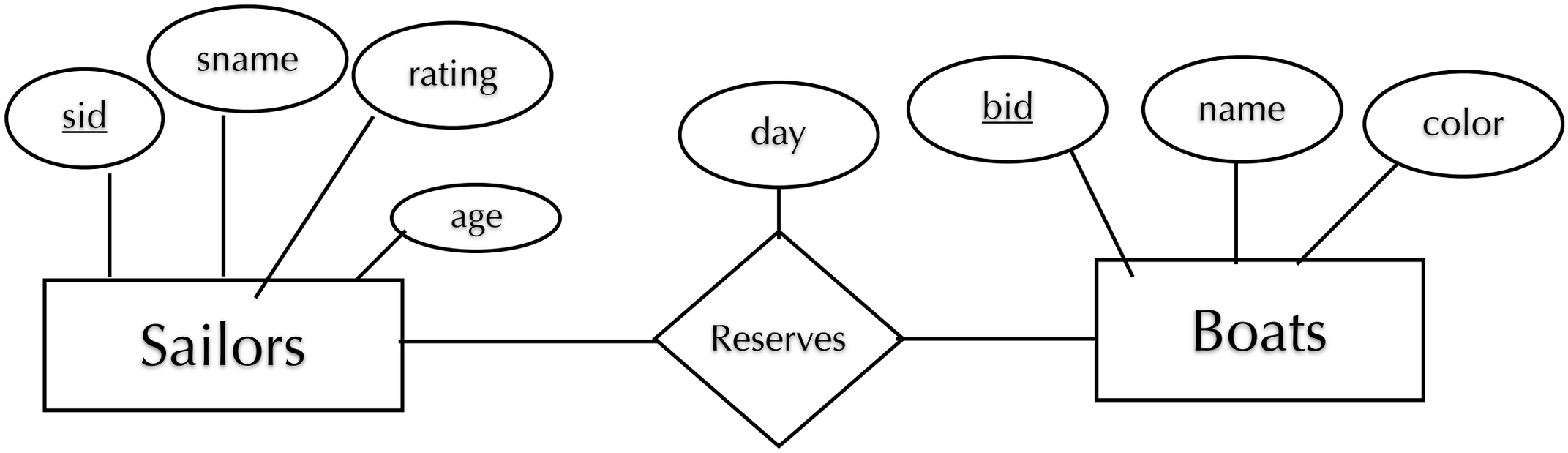
- Banks(bankid: int, bname:string, baddress: string)

How do we model relationship sets?

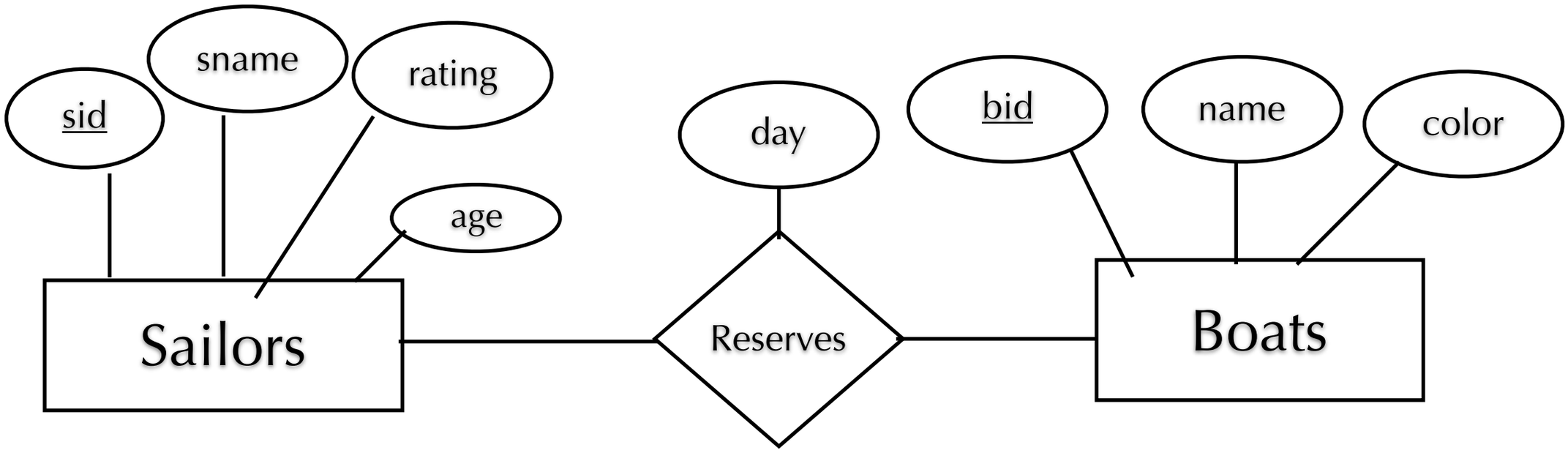
Database Schema

- Customers(ssn: string,name:string, address:string, phone:string)
- Accounts(number: int, type:string, balance:real)
- Banks(bankid: int, bname:string)
- Has(ssn:string, number:int)
- Part_Of(number: int, bankid:int)

Boats Sailors ER Model



Boats Sailors ER Model



- Sailors (sid: int, sname: string, rating:int, age: real)
- Boats(bid: int, name:string, color: string)
- Reserves(sid: int, bid:int, day: date)

Before SQL, A bit about constraints

- Domain Constraints
  - Attributes have a data type
- Referential Integrity Constraints
  - Foreign key
- Not Null Constraints
- Unique Constraints
- Key Constraints
- We'll talk more about constraints in SQL

Topics

- Introduction to DBMS
- Relational Data Model
- *Relational Algebra*
- Conceptual Design: the Entity-Relationship Model
- Structured Query Language (SQL)
- Application Development (Java, Python)
- Schema Refinement and Normal Forms
- Database Security and Authorization
- Some NoSQL topics (If time permitted)

Structured Query Language (SQL)

- 1974: IBM developed SQL (Structured Query language), initially called SEQUEL (Structured English Query Language), as part of the IBM System R database system
- Late 1980s: SQL was standardized. In 1986 adopted by the American National Standards Institute (ANSI), and in 1987 adopted by the International Organization for Standardization (ISO)
- SQL Standards:
  - 1986, 1989, 1992, 1999, 2003, 2006, 2008, 2011, 2016, 2019

SQL

- Offers support for:
  - Creating databases, tables, views, indexes..etc
  - Inserting, updating, deleting records
  - Querying the database
  - Controlling access
  - etc.

Creating Tables (in its simplest form)

```
Create table tablename (column1 datatype,
                        column2 datatype,
                        column3 datatype,
                        …);
```

We can also add constraints

Creating tables (in its simplest form)

```
create table tablename (column1 datatype constraint,
                        column2 datatype constraint,
                        column3 datatype constraint,
                        …);
```

Some constraints cannot be added after the column

Constraints could be

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- SET DEFAULT
- CREATE INDEX

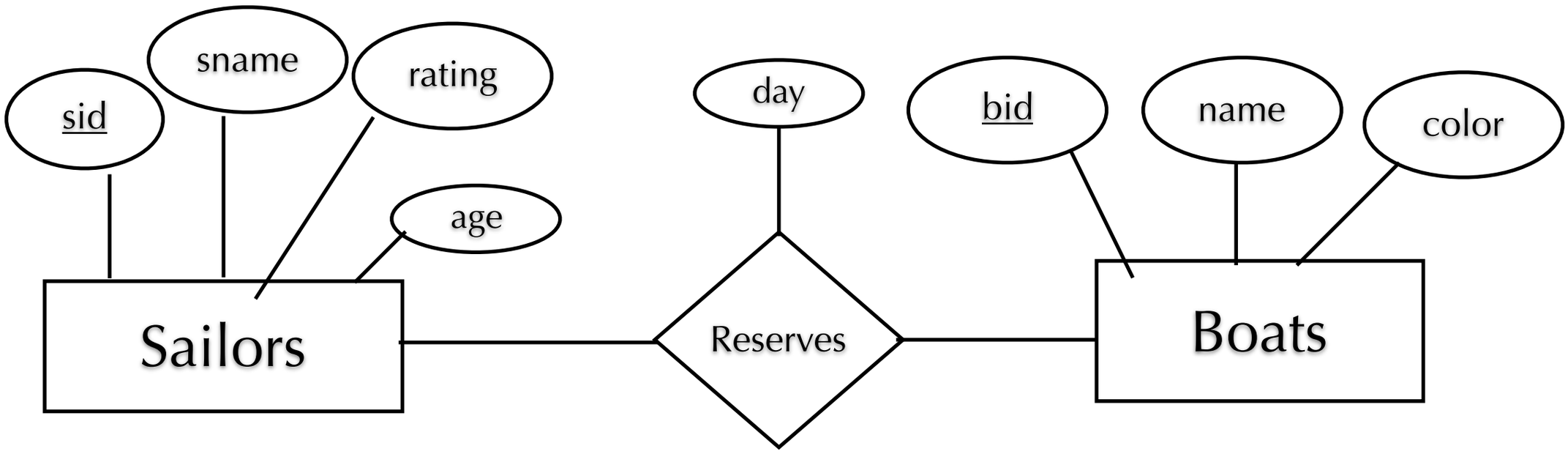
Data types

- CHAR(n): fixed length
- VARCHAR(n): variable length (in Oracle stored as VARCHAR2(n))
- DATE
- TIMESTAMP

Data types (cont.)

- DECIMAL(p,s), NUMERIC(p,s)
  - In Oracle implemented as NUMBER(p,s)
    - p= precision, s = scale.
      - e.g.: 2345.12 p=6, s=2
- NUMBER(p,s)
  - NUMBER: floating point with some default p,s
  - NUMBER(p) : same as an integer
- INT, SMALLINT
  - In Oracle implemented as NUMBER(38)
- REAL, FLOAT

Example
Boats Sailors ER Model and its Relational Database Schema



- Sailors (sid: int, sname: string, rating:int, age: real)
- Boats(bid: int, name:string, color: string)
- Reserves(sid: int, bid:int, day: date)

Example
Sailors Reservation Database
Create Sailors Table

```
create table sailors( sid number(9)
primary key,
                      sname varchar(20),
                      rating number(2),
                      age number(4,2)
);
```

Example
Sailors Reservation Database
Create Boats Table

```
create table boats(bid number(9) primary
key,
                   name varchar(20),
                   color varchar(20)
);
```

Example
Sailors Reservation Database
Create Reserves Table

```
create table reserves( sid number(9),
                       bid number(9),
                       day date,
                       primary key (sid,bid),
                       foreign key (sid) references sailors
                       foreign key (bid) references boats
);
```

What if we want to allow a sailor to reserve the same boat more than once?

```
create table reserves( sid number(9),
                       bid number(9),
                       day date,
                       primary key (sid,bid,day),
                       foreign key (sid) references sailors
                       foreign key (bid) references boats
);
```

Describe table

desc table;

- E.g.: desc sailors;

DELETING A TABLE

- DROP TABLE tablename;

Alter table

- Statement used to modify an existing table. Example:
  - Add new columns
  - Modify existing column
  - Drop columns
  - Rename column
  - Rename table

Alter table
Add columns

- ALTER TABLE table name

  ADD colname datatype constraints;

- ALTER TABLE table name

  ADD (colname1 datatype constraints,

  colname2 datatype constraints,

  ....

  colnamen datatype constraints

  );

Every value in the new column will be NULL by default. You can set a DEFAULT value different than NULL.

Alter Table Add Column Example

- ALTER TABLE reserves

  ADD invoiceday DATE;

Alter table MODIFY columns

- ALTER TABLE table name

  MODIFY colname datatype constraints;

- ALTER TABLE table name

  MODIFY (colname1 datatype constraints,

  colname2 datatype constraints,

  ....

  );

Alter Table Add Column Example

- ALTER TABLE reserves

  MODIFY day VARCHAR(30);

Alter Table Rename Column

- ALTER TABLE table name

  RENAME COLUMN colname TO newcolname;

E.g.:

ALTER TABLE reserves

  RENAME COLUMN day to reserveday;

Alter Table
Rename Table

- ALTER TABLE tablename

  RENAME TO newtablename

E.g.

ALTER TABLE reserves

    RENAME TO otherreserves;

Rename it back:

ALTER TABLE otherreserves

    RENAME TO reserves;

Alter Table
Drop Column

- ALTER TABLE tablename

  DROP COLUMN colname;

- E.g:

  ALTER TABLE reserves

  DROP COLUMN reserveday;

INSERT Statement

- INSERT INTO tablename (colname1, colnam2, …, colnamen)

  VALUES (val1, val2, …, valn);

INSERT multiple rows with one command (NOTE: this command with multi rows does not work in Oracle!)

- INSERT INTO tablename (colname1, colnam2, …, colnamen)

  VALUES (val1, val2, …, valn),

  (valb1, valb2, …, valbn),

  ….

  (valk1, valk2, …, valkn);

You can also insert data from the result of a SELECT statement!
We will talk about that later.

Structure of SELECT queries

- SELECT [DISTINCT] target-list

FROM relation-list

[WHERE conditions];

- **relation-list** : list of relation names
- **target-list**: list of attributes from relations in relation-list
- **conditions**: condition expressions connected by AND, OR
  - Condition is of forms:
    - **attribute** op **constant**
    - **attributeA** op **attributeB**
    - **op** can be: =, <=, >=, <>, or string operators
  - Later we will discuss more about this when we talk about nested queries
- **DISTINCT**: if present, eliminates duplicates from the result

Simplest SELECT statement

- SELECT * FROM tablename;
  - Selects all the columns from table tablename
  - The resulted relation will contain all data from tablename. Columns will appear in the same order they are defined in tablename
- SELECT col1, col2 FROM tablename;
  - Selects columns col1 and col2 from table tablename
  - You can give any column from table name
  - The resulted relation will contain all rows from tablename but extract only the columns from the list. The columns will be in the order they are specified in the list

Select and renaming column in the results

- SELECT col1 as newcol1, col2 as newcol2 FROM tablename;

  - Selects columns col1 and col2 from table tablename, and renames them as newcol1 and newcol2 in the result (NOT IN THE ORIGINAL TABLE!!)

Example Database Instance

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

Example Database Instance

Select query

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

```
select * from sailors;
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

Example Database Instance
Select query

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

select rating, sname from sailors;

rating	sname
7	dustin
8	lubber
10	rusty
10	rusty

Notice the difference between this result and the result of of a relational algebra expression with projection operator

Example Database Instance

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

select DISTINCT rating, sname from sailors;

rating	sname
7	dustin
8	lubber
10	rusty

Example Database Instance

Select with WHERE clause

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

select * from sailors where rating=10;

sid	sname	rating	age
58	rusty	10	35.0
59	rusty	10	45.0

Example Database Instance

Select with WHERE clause

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

select * from sailors where rating>9 or rating <=7;

sid	sname	rating	age
22	dustin	7	45.0
58	rusty	10	35.0
59	rusty	10	45.0

What about combining multiple tables?

```
select * from sailors, reserves where
sailors.sid=reserves.sid;
```

Same as:

```
select * from sailors s1, reserves r1 where
s1.sid=r1.sid;
```

Example Database Instance

Select with WHERE clause

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

select * from sailors, reserves where sailors.sid=reserves.sid;

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/22
22	dustin	7	45.0	22	102	10/20/22
58	rusty	10	35.0	58	101	10/11/22

Think about the cross product followed by a selection condition

Example Database Instance

Select with WHERE clause

sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.0
58	rusty	10	35.0
59	rusty	10	45.0

reserves

sid	bid	day
22	101	10/10/22
58	101	10/11/22
22	102	10/20/22

boats

bid	name	color
101	interlake	red
102	clipper	green

select * from sailors, reserves where sailors.sid=reserves.sid and bid=101;

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/22
58	rusty	10	35.0	58	101	10/11/22

Connecting to a Database

- Connect from:
    - SQL CLIENT (could be a CLI or one with GUI)
    - Code (programming) (e.g. Connecting from java, from python, etc)
- You usually have to specify:
    - Database hostname
    - Database
    - Port
    - Username
    - Password
- You also need a driver

Connecting to Oracle

- We will use sqlplus on the Unix machine
- You have to ssh into your unix account and start sqlplus

Questions?