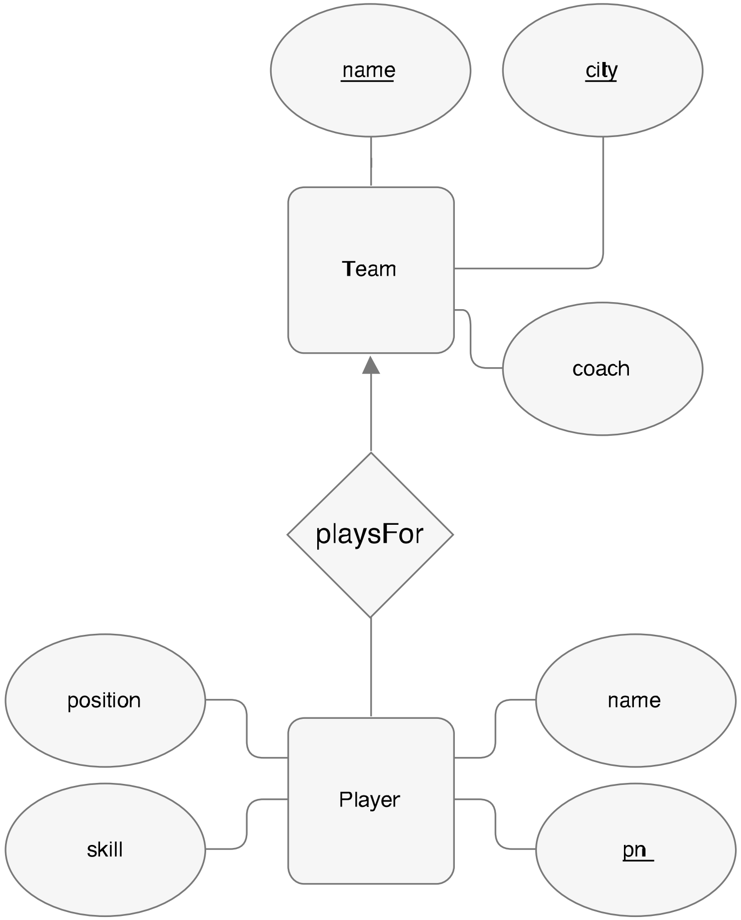
Lecture 7a

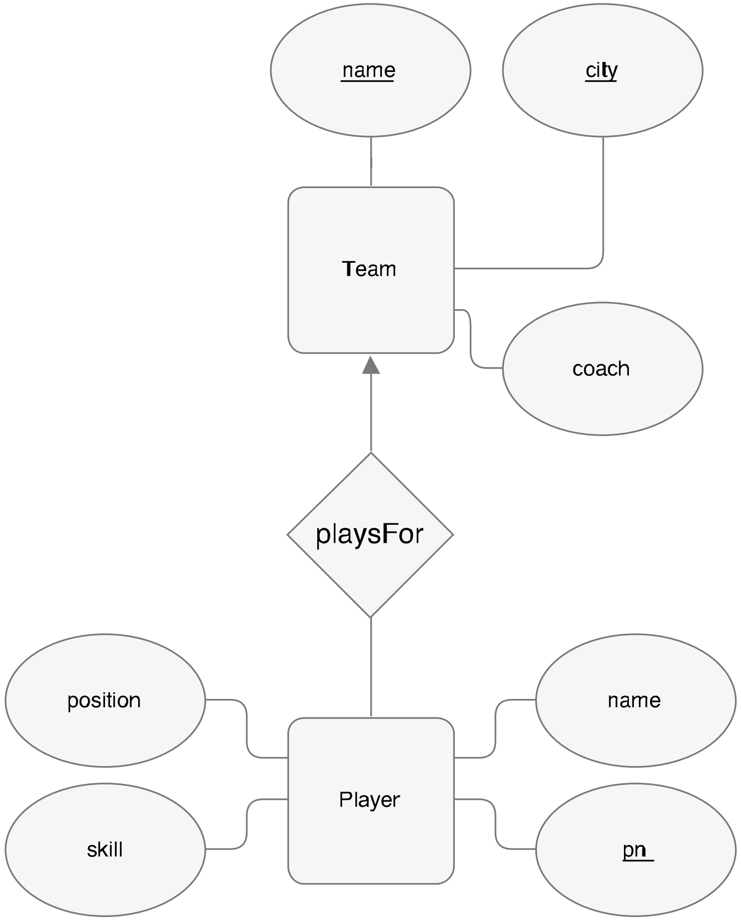
ER – Exercises + SQL

Ex 1



- A team’s name is only unique in their own city
- Each player belongs to maximum one team
- Each team has a name, a city, a coach and a set of players
- Each player has a name, a unique person number, a position, a skill level

Ex 1 (a)

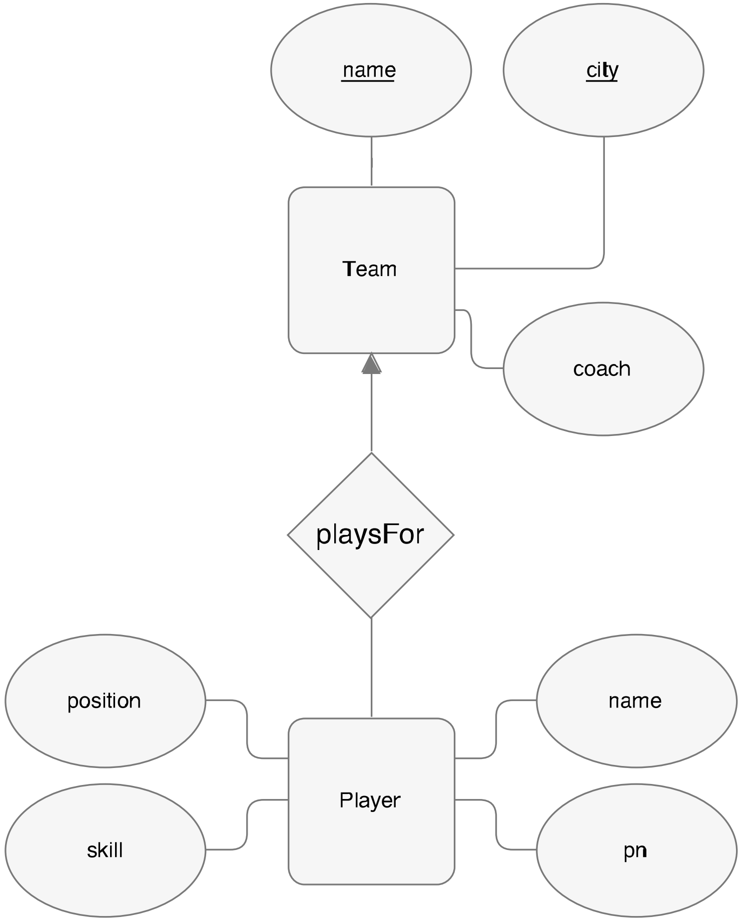


Team(<u>city</u>, <u>name</u>, coach)

Player(<u>pn</u>, name, position, skill)

PlaysFor(pn, city, name)
pn -> Player.pn
(city, name) -> Team.(city, name)

Ex 1 (b)



Team(city, name, coach)

Player(pn, name, position, skill, city, name)
(city, name) -> Team.(city, name)

Ex 1

Check slides from lecture 2

- Which one is better?

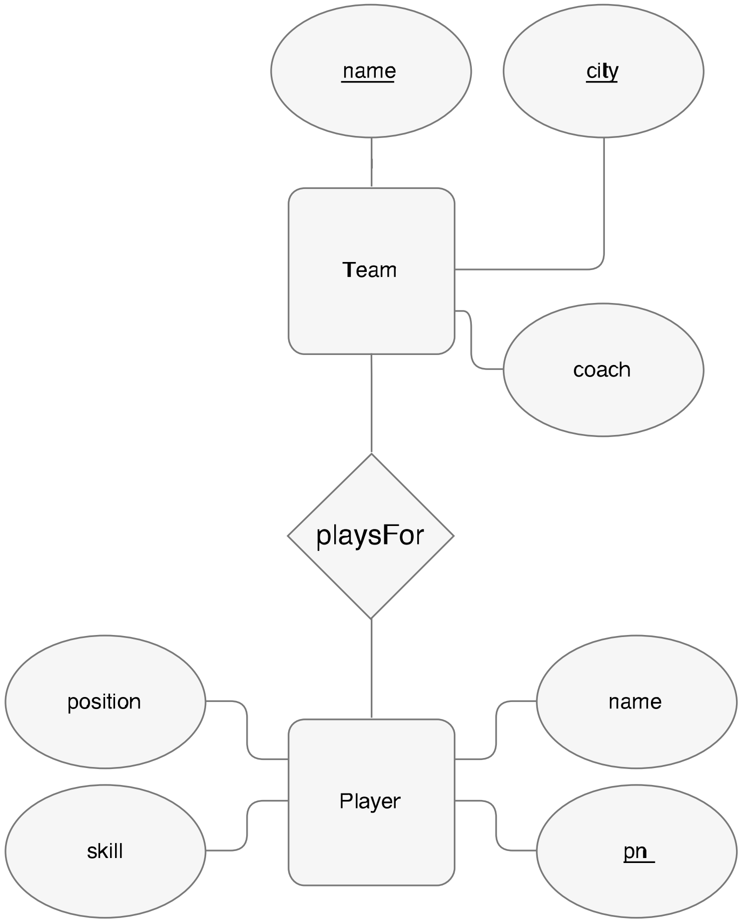
Ex 1 (a): NO-NULL Approach

- Safe translation - no NULLs anywhere.
- May lead to duplication of the pn.
- May lead to more *joins*.
- **Default translation rule**, use **unless** you have a good reason not to.

Ex 1 (b): NULL Approach

- Will lead to NULLs for players that have no team.
- Can **sometimes** be preferred when *not* having a team is an uncommon exception to the rule.
- Reduces the need for *joins.*

Ex 1 (c)

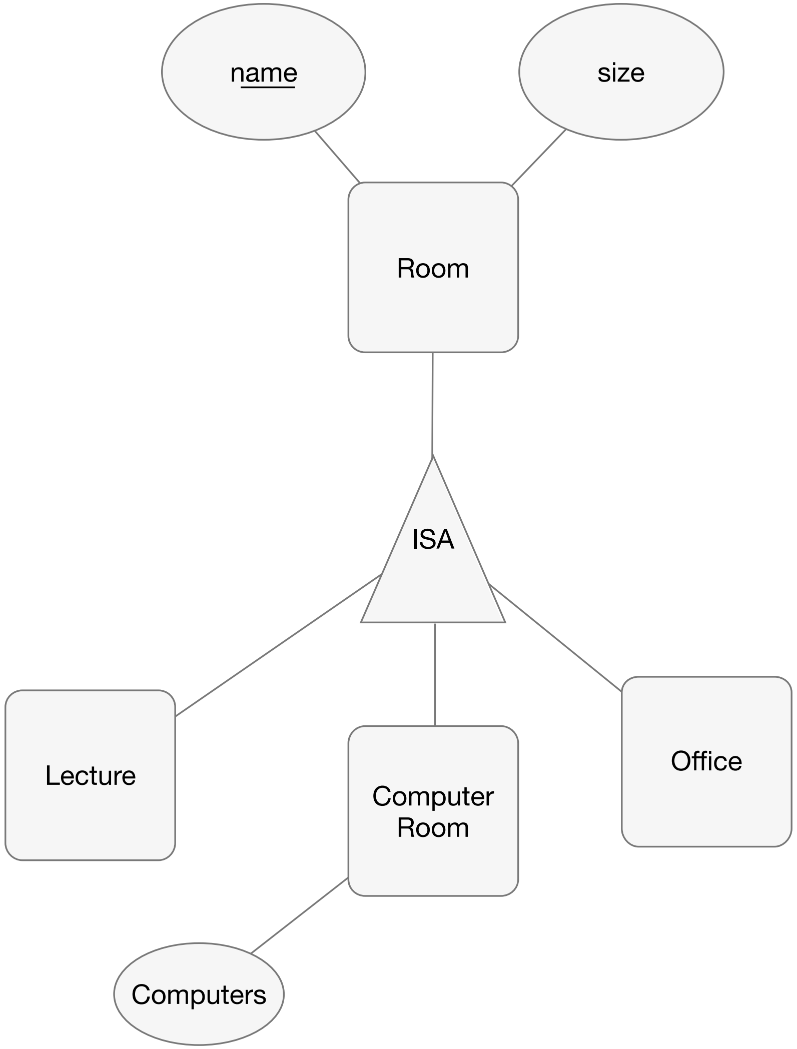


Team(<u>city</u>, <u>name</u>, coach)

Player(<u>pn</u>, name, position, skill)

PlaysFor(<u>pn</u>, <u>city</u>, <u>name</u>)
pn -> Player.pn
city -> Team.city
name -> Team.name

Ex 2



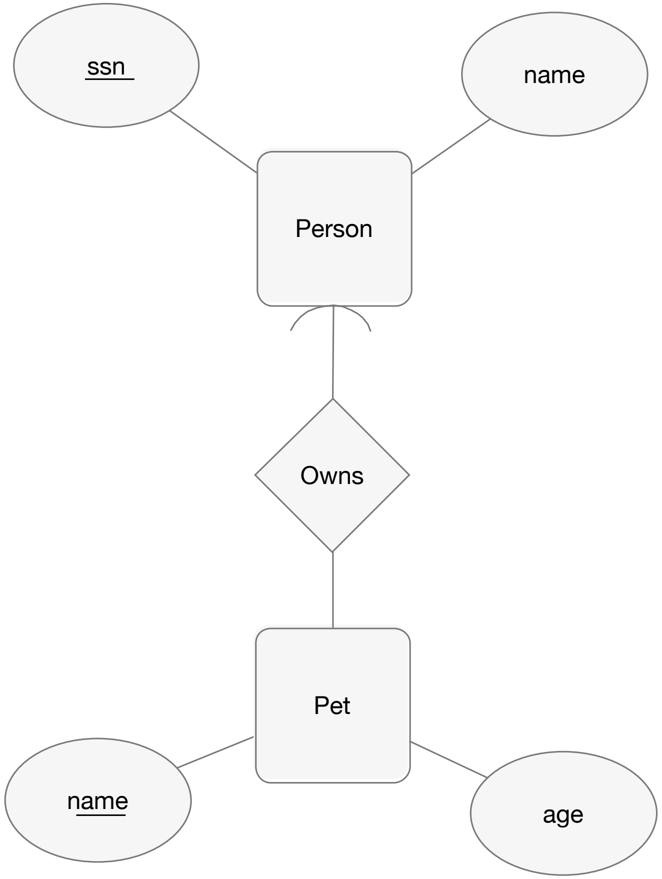
- Room (<u>name</u>, size)
- Lectures (name)
  name -> Room.name
- Office (<u>name</u>)
  name -> Room.name
- ComputerRoom (<u>name</u>, computers)
  name -> Room.name

Ex 3

Problem Definition

- Persons are identified with ssn
- Name of pets are unique per person
- Persons can have any number of pets
- One pet can only be owned by one person

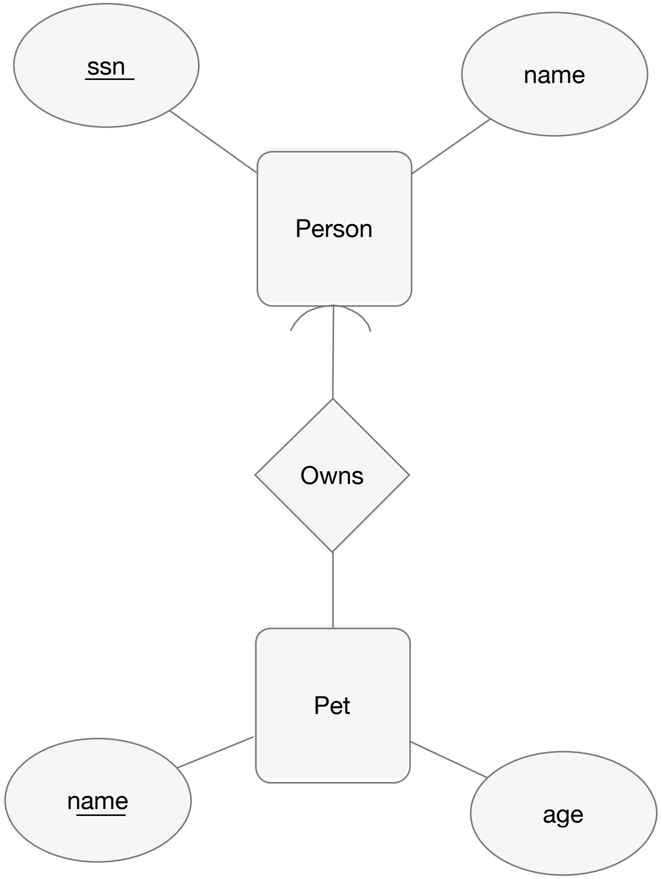
Ex 3a



```
CREATE TABLE Person (
  ssn INT PRIMARY KEY,
  name TEXT NOT NULL
);


CREATE TABLE Pet(
  name TEXT PRIMARY KEY,
  ssn INT REFERENCES Person NOT
NULL
);
```

We DO NOT address the constraints

Ex 3b

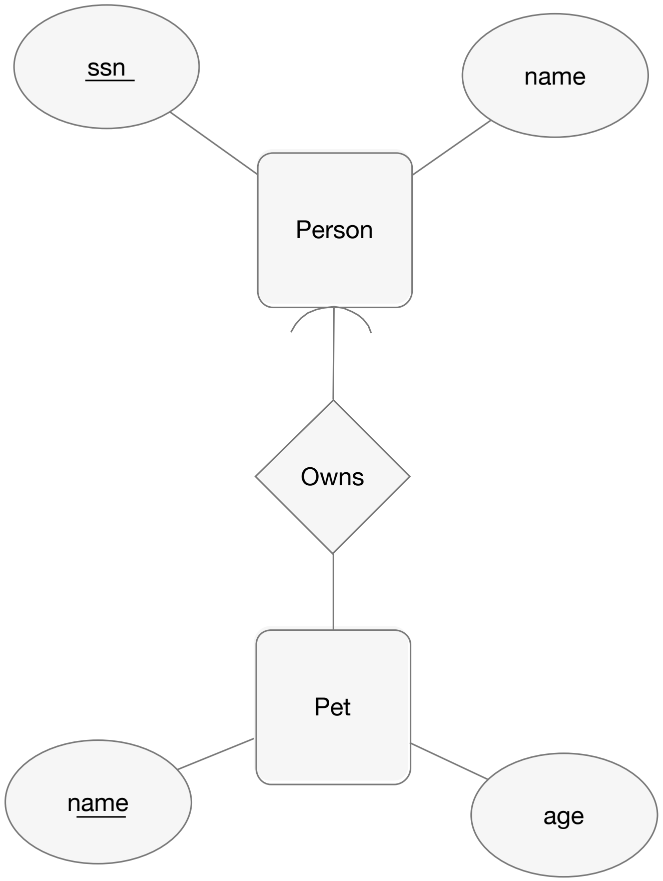


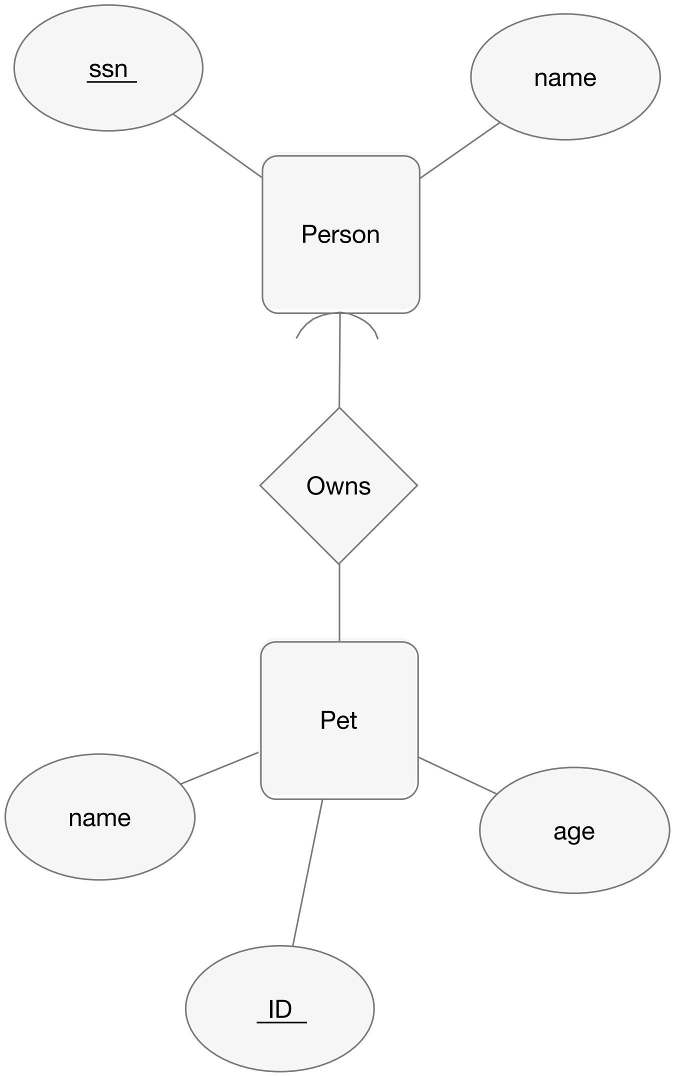
```
CREATE TABLE Person (
  ssn INT PRIMARY KEY,
  name TEXT NOT NULL
);


CREATE TABLE Pet(
  name TEXT,
  ssn INT REFERENCES Person NOT
NULL,
  PRIMARY KEY (name, ssn)
);
```

We DO address the constraints

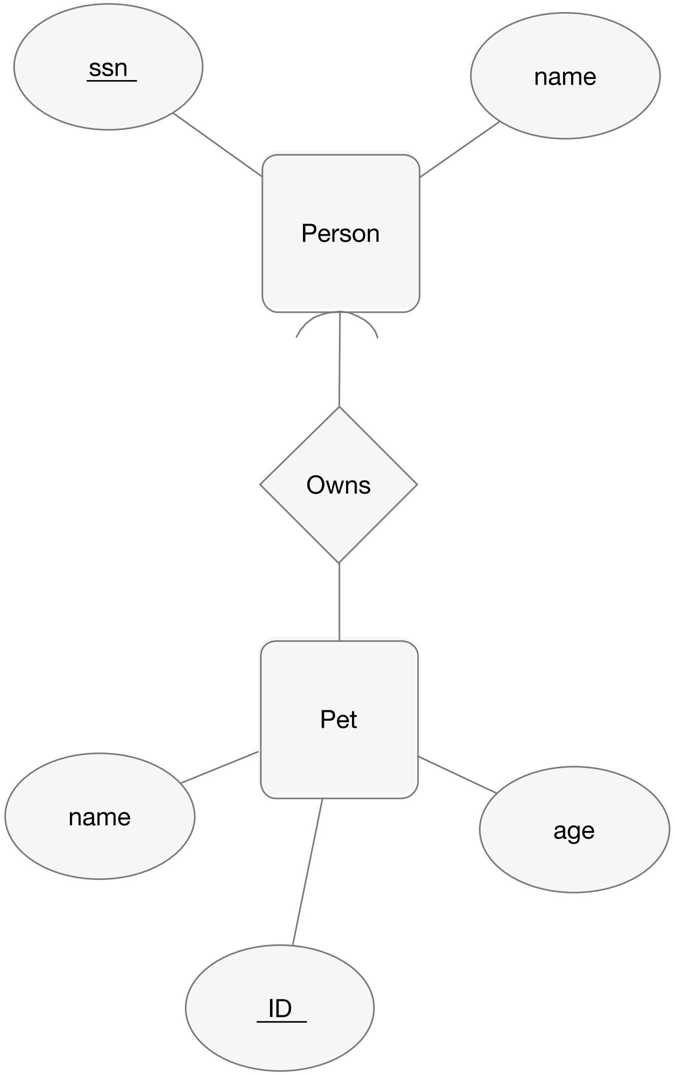
BUT we DO NOT model the E/R

Ex 3 (Solution)





Ex 3c

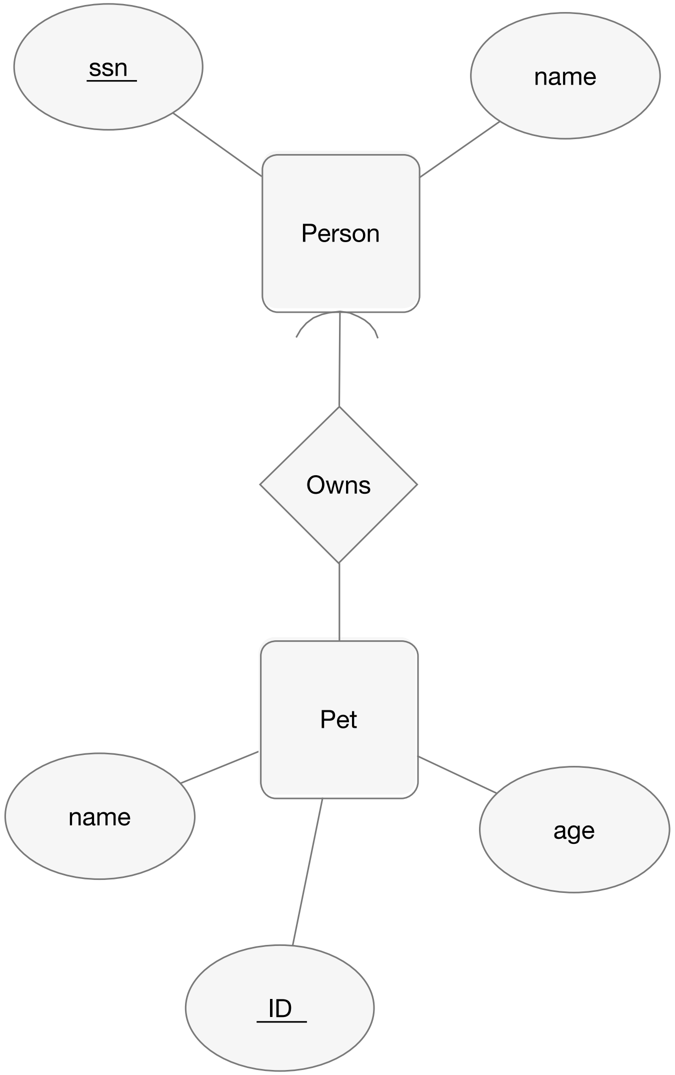


```
CREATE TABLE Person (
  ssn INT PRIMARY KEY,
  name TEXT NOT NULL
);


CREATE TABLE Pet(
  pet_id INT PRIMARY KEY,
  name TEXT,
  ssn INT REFERENCES Person NOT
NULL
);
```

We DO NOT address the constraints:

```
INSERT INTO Person VALUES (1, 'Name1');
INSERT INTO Pet VALUES (1,'Pet1',1);
INSERT INTO Pet VALUES (5,'Pet1',1);
```

Ex 3d



```
CREATE TABLE Person (
  ssn INT PRIMARY KEY,
  name TEXT NOT NULL
);


CREATE TABLE Pet(
  pet_id INT PRIMARY KEY,
  name TEXT,
  ssn INT REFERENCES Person NOT NULL,
  UNIQUE (ssn,name)
);
```