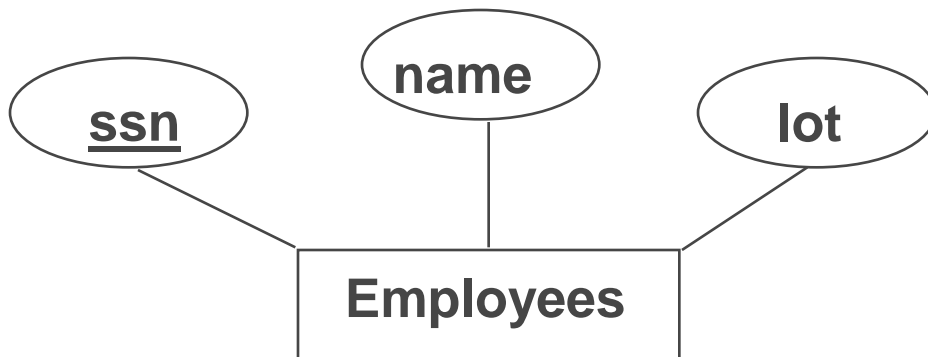


# Database Management System

## Lecture 7 ER to Relational

# Logical DB Design: ER to Relational

- ▶ Entity sets to tables:



```
CREATE TABLE Employees  
  (ssn CHAR(11),  
   name CHAR(20),  
   lot INTEGER,  
   PRIMARY KEY (ssn))
```

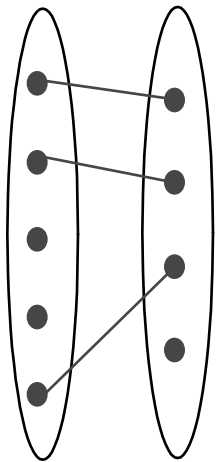
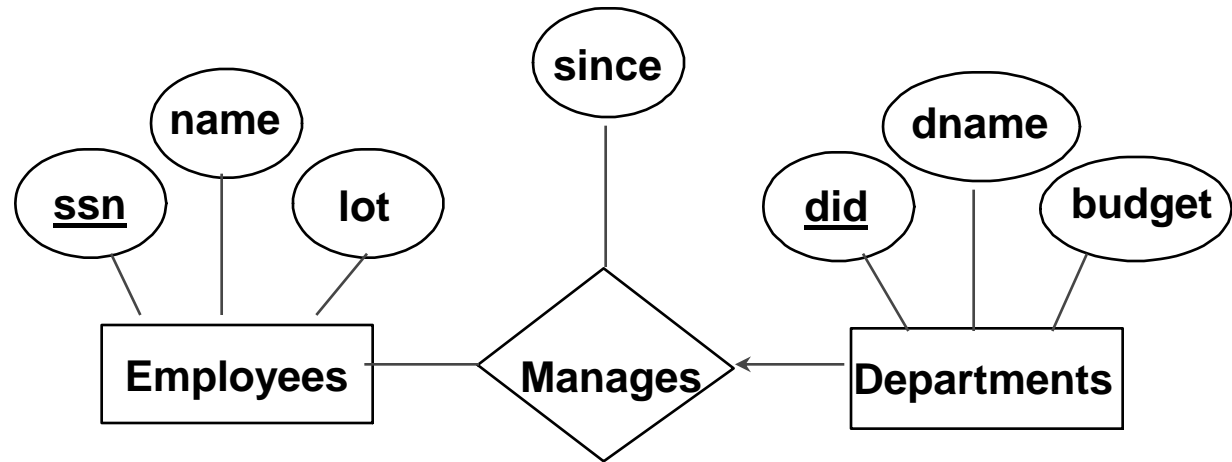
# Relationship Sets to Tables

- ▶ In translating a relationship set to a relation, attributes of the relation must include:
  - Keys for each participating entity set (as foreign keys).
    - This set of attributes forms a *superkey* for the relation.
  - All descriptive attributes.

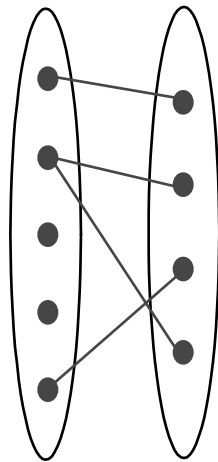
```
CREATE TABLE Works_In(  
    ssn CHAR(11),  
    did INTEGER,  
    since DATE,  
    PRIMARY KEY (ssn, did),  
    FOREIGN KEY (ssn)  
        REFERENCES Employees,  
    FOREIGN KEY (did)  
        REFERENCES Departments)
```

# Review: Key Constraints

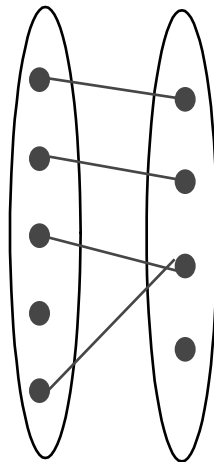
- ▶ Each dept has at most one manager, according to the key constraint on Manages.



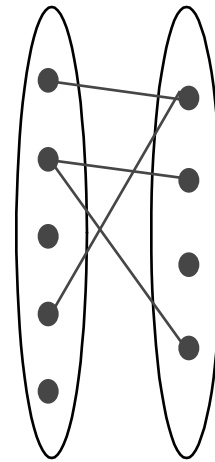
1-to-1



1-to Many



Many-to-1



Many-to-Many

*Translation to relational model?*

# Translating ER Diagrams with Key Constraints

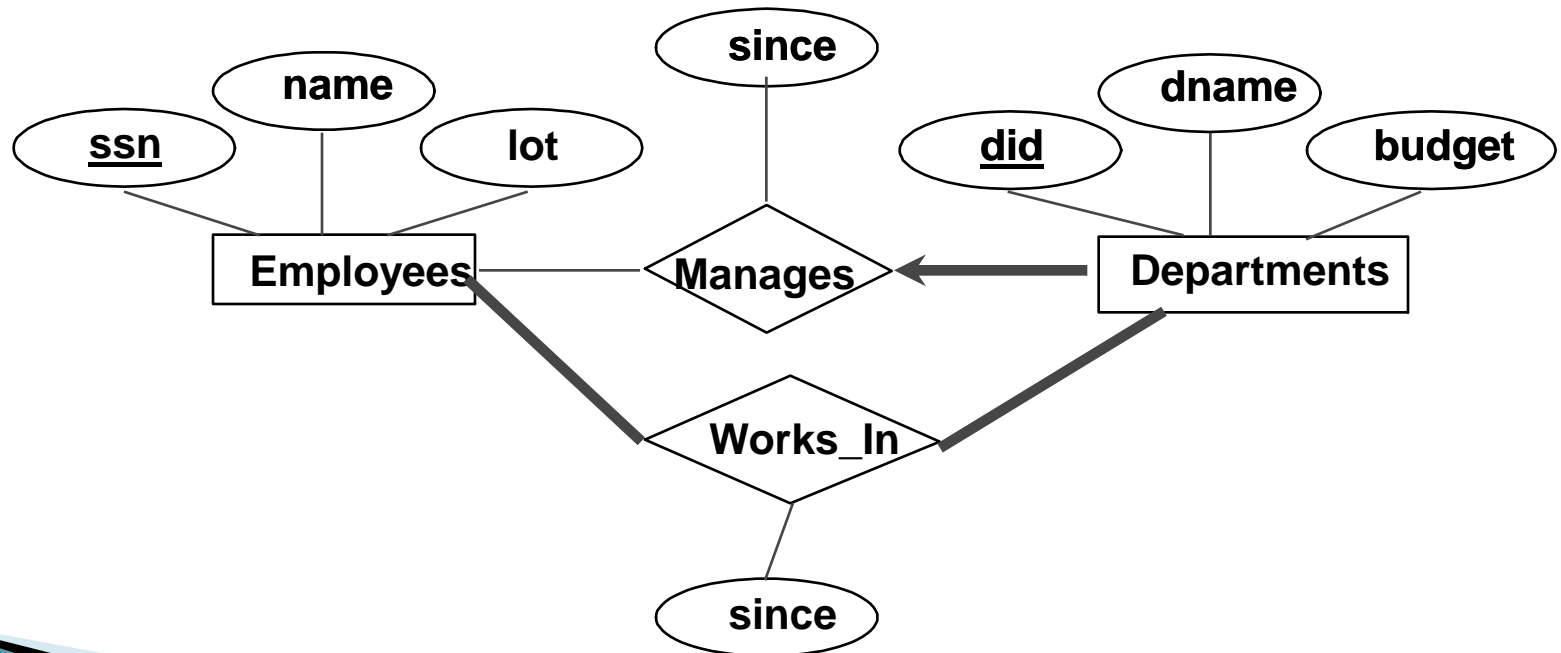
- ▶ Map relationship to a table:
  - Note that **did** is the key now!
  - Separate tables for Employees and Departments.
- ▶ Since each department has a unique manager, we could instead combine Manages and Departments.

```
CREATE TABLE Manages(  
    ssn CHAR(11),  
    did INTEGER,  
    since DATE,  
    PRIMARY KEY (did),  
    FOREIGN KEY (ssn) REFERENCES Employees,  
    FOREIGN KEY (did) REFERENCES Departments)
```

```
CREATE TABLE Dept_Mgr(  
    did INTEGER,  
    dname CHAR(20),  
    budget REAL,  
    ssn CHAR(11),  
    since DATE,  
    PRIMARY KEY (did),  
    FOREIGN KEY (ssn) REFERENCES Employees)
```

# Review: Participation Constraints

- ▶ Does every department have a manager?
  - If so, this is a *participation constraint*: the participation of Departments in Manages is said to be *total (vs. partial)*.
  - Every *did* value in Departments table must appear in a row of the Manages table (with a non-null *ssn* value!)



# Participation Constraints in SQL

- ▶ We can capture participation constraints involving one entity set in a binary relationship, but little else (without resorting to CHECK constraints).

```
CREATE TABLE Dept_Mgr(  
  did INTEGER,  
  dname CHAR(20),  
  budget REAL,  
  ssn CHAR(11) NOT NULL,  
  since DATE,  
  PRIMARY KEY (did),  
  FOREIGN KEY (ssn) REFERENCES Employees,  
  ON DELETE NO ACTION)
```