

# CSCI4333 Database Design & Implement

## **Lecture Nine – Relational Model**

**3**

Instructor: Dr. Yifeng Gao

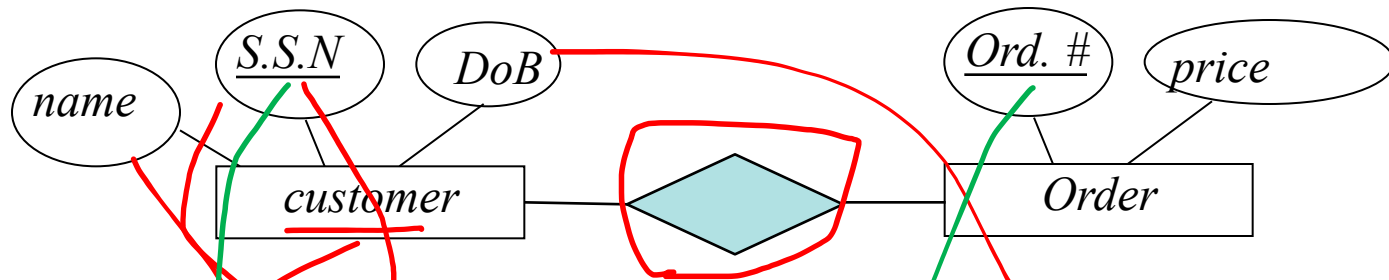
**Q1(5pt):** Please draw the E-R Diagram for this University Database. Your E-R Diagram should describe all conditions mentioned below:

1. For each professor, the database stores Name, SSN, and Date of Birth
2. For each student, the database stores student ID, Name, and Date of Birth.
3. Each course record is associated with a course ID, a section id.
4. Each professor can be uniquely identified by SSN
5. Each student can be uniquely identified by student ID
6. Each course can be uniquely identified by combination of course ID and section id
7. A professor must teach one or more courses. And a course must be taught by one and only one professor.
8. A student can take any number of courses and each courses must have at least one students.
9. A student can be supervised by only one (or zero) professor. A professor can supervise any number of students.

**\*Additional Notes:** "one or more" = "at least one"

$\geq 1$  —  
 $= 1$  —  
 $\leq 1$  —  
 $\times$  —

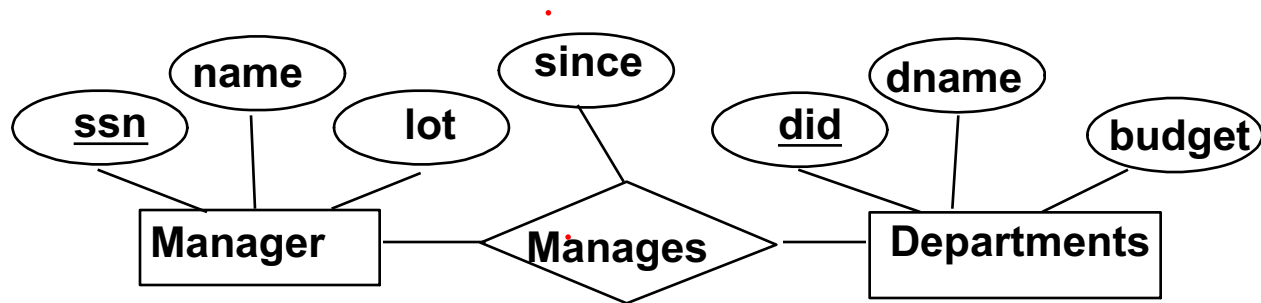
# ER-diagram Translation



*customer*(SSN : string, *name* : string, *DoB* : number)

*Order*(OrdNum : number, *price* : number)

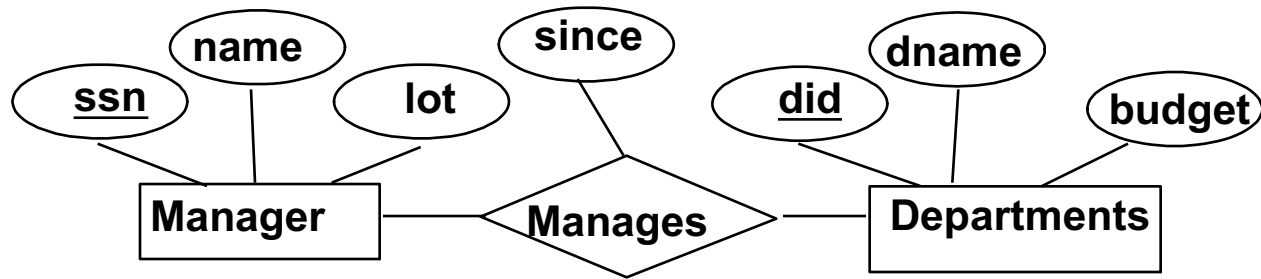
*purchase*(S.S.N : string, OrdNum : number)



*Manager(SSN: string, name : string, lot : number)*

*Manages(since:string, ssn:string, did:string)*

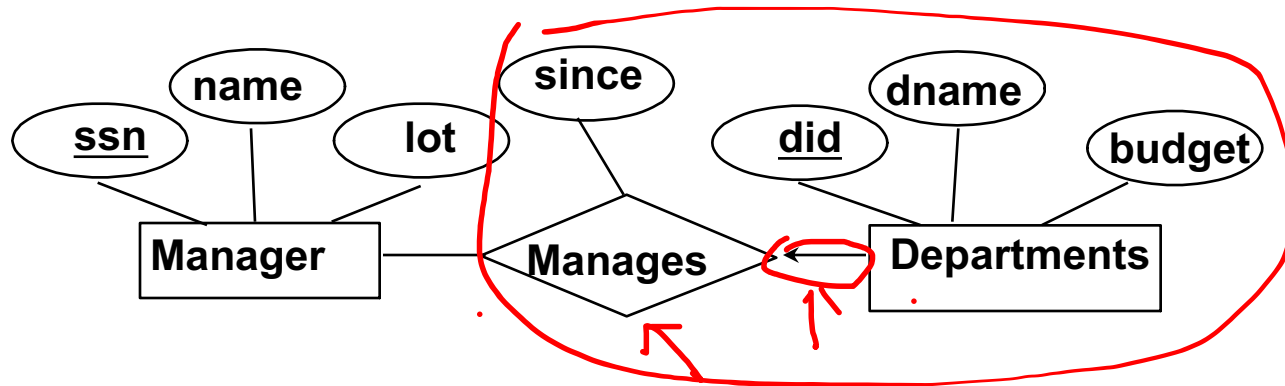
*Departments(did : string, dname: string, budget : number)*



*Manager(SSN : string, name : string, lot : number)*

*Manages(since:string, ssn:string, did:string)*

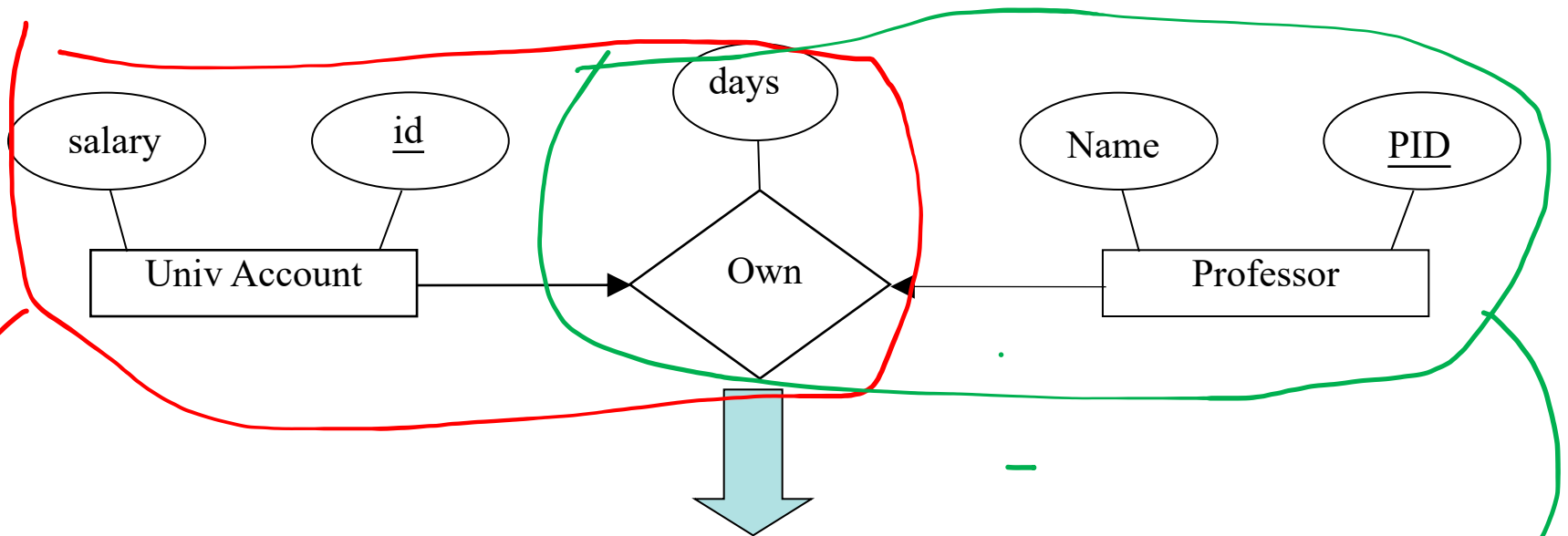
*Departments(did : string, dname: string, budget : number)*



*Manager*(SSN : string, name : string, ~~lot : number~~)

*Departments*(did : string, dname: string, ~~ssn: string,~~  
~~since: date,~~ budget : number)

# ER-diagram Translation



*Professor*(PID : string, id: string, Name: string, *days* : string)

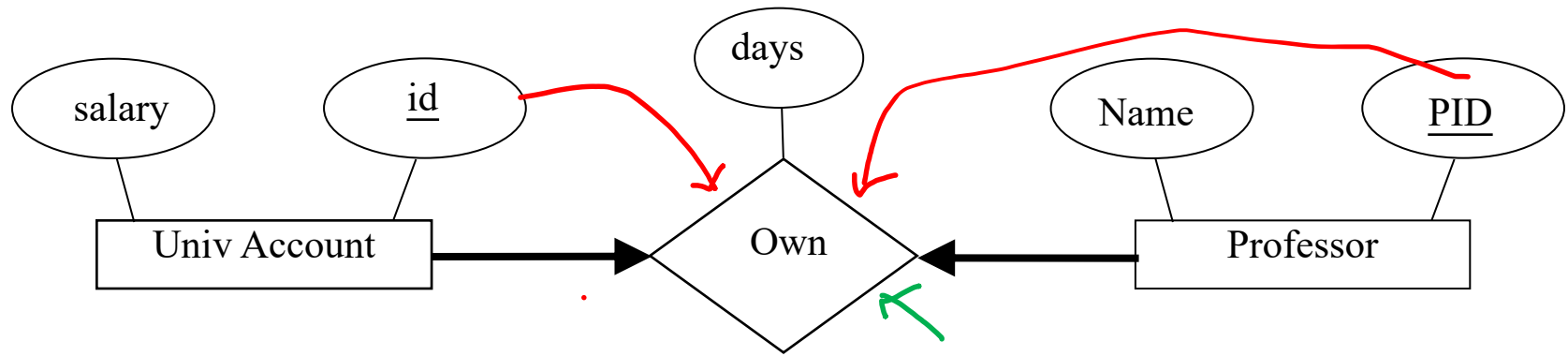
*UniversityAcc*(id : string, salary: number)

**or**

*UniversityAcc*(id : string, salary: number, *days* : string, *PID* : string)

*Professor*(PID : string, Name: string)

# ER-diagram Translation



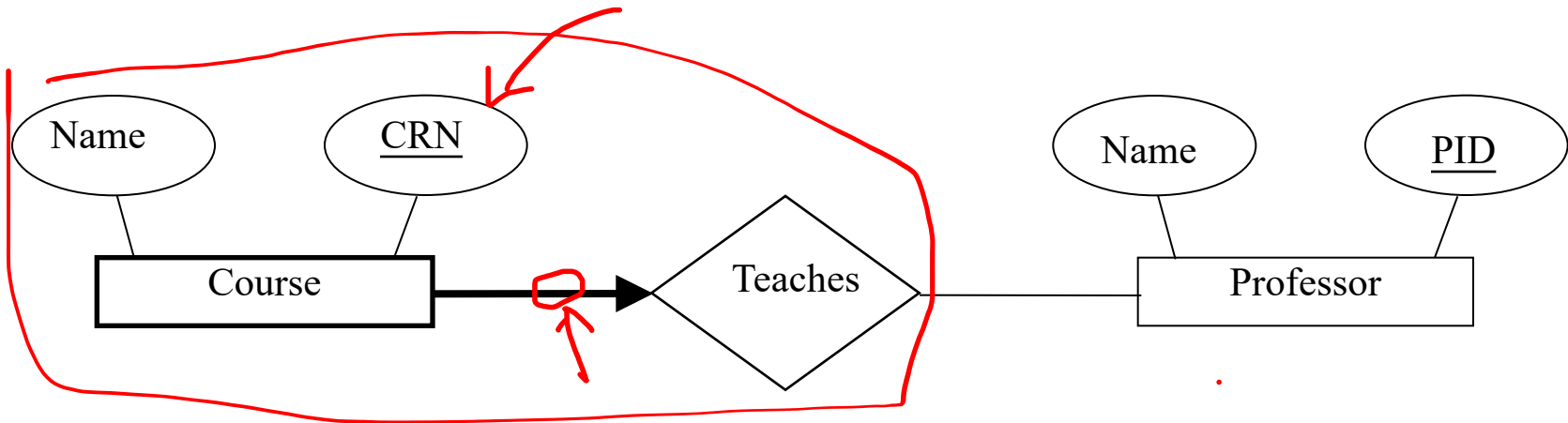
ProfessorAcc(PID: string, Name: string, id: string,  
days: date, salary: number)

or

ProfessorAcc(PID: string, Name: string, id: string,  
days: date, salary: number)



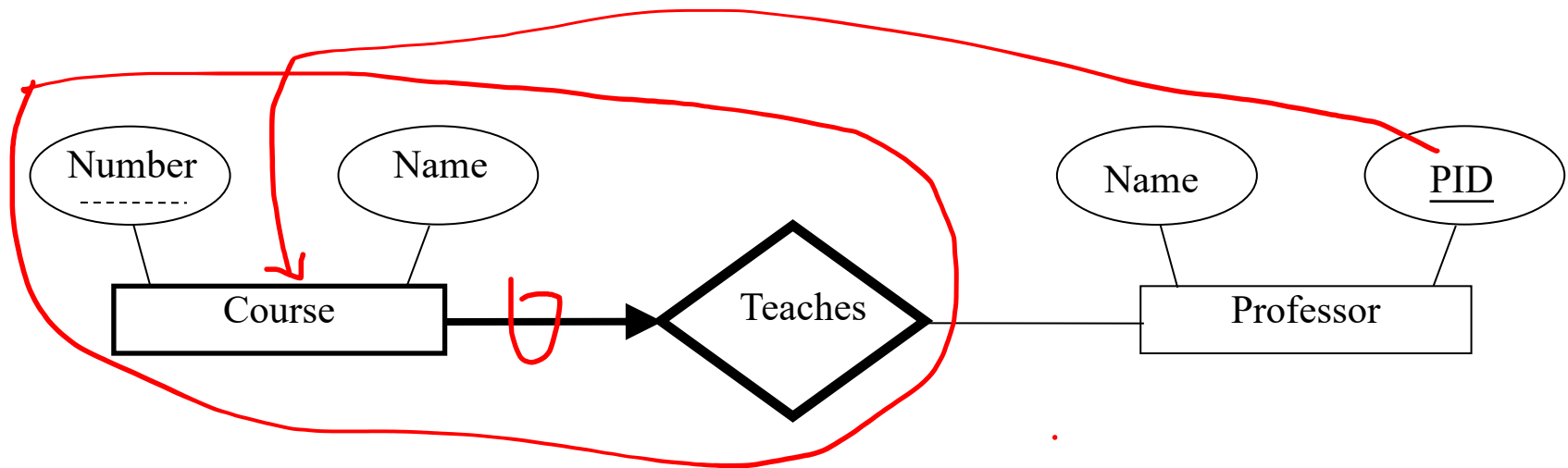
# ER-diagram Translation



*professor(PID : string, name : string)*

*Course(PID : string, name : string, CRN : string)*

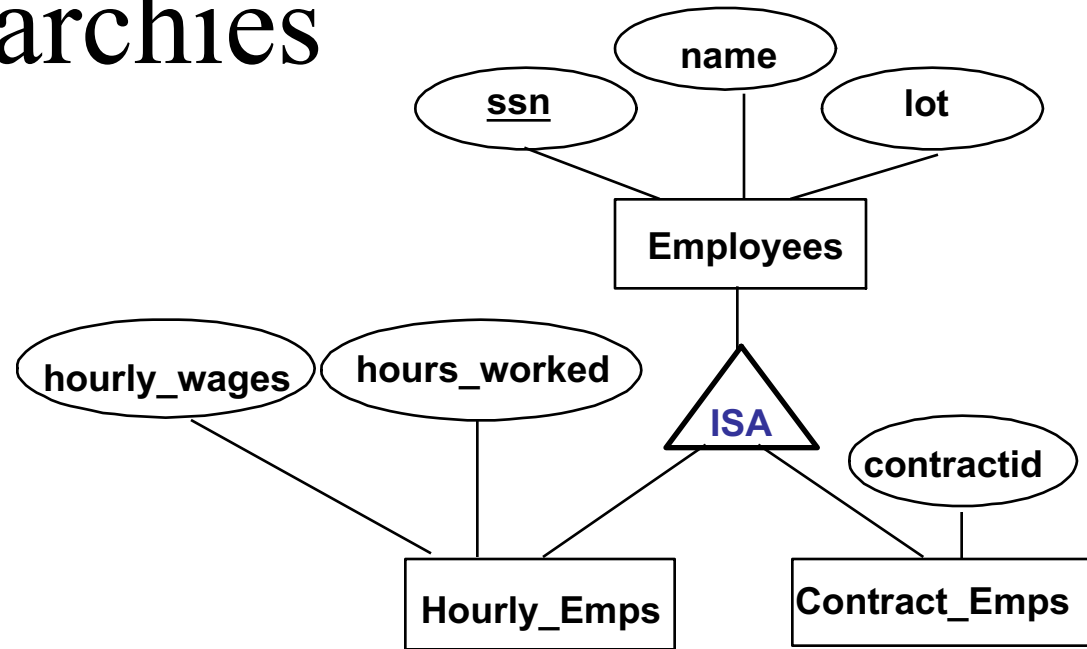
# ER-diagram Translation – Weak Entity



*professor(PID : string, name : string)*

*Course(PID : string, name : string, number : string)*

# ISA ('is a') Hierarchies

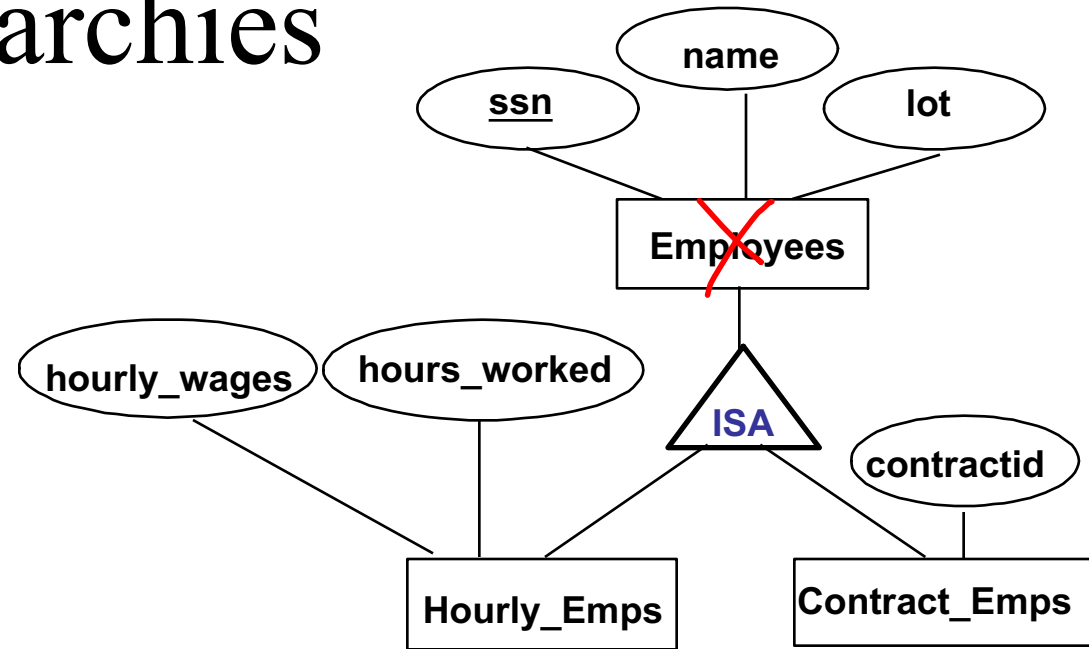


## General approach:

3 relations: Employees, Hourly\_Emps and Contract\_Emps.

- *Employee(ssn,name,lot)*
- *Hourly\_Emps(hourly\_wages, hours\_worked, ssn);*
  - must delete Hourly\_Emps tuple if referenced Employees tuple is deleted.
- *Contract\_Emps: Contract\_Emps (contract\_id,ssn);*
  - must delete Contract\_Emps tuple if referenced Employees tuple is deleted.

# ISA ('is a') Hierarchies



Special Case:

Hourly\_Emps and Contract\_Emps are sufficient if each employee **must be in one of these two subclasses**.

- *Hourly\_Emps*(*hourly\_wages*, *hours\_worked*, name, lot, ssn)
- Contract\_Emps: Contract\_Emps (contract\_id, name, lot, ssn)