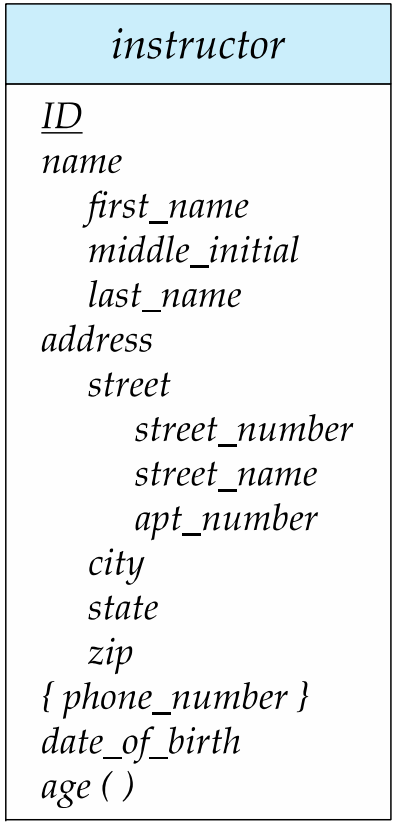
ER to Relational Model Mapping Rules

Step 1 : Entity

- From entity to relation/table
- Create a new relation that includes all the attributes
- Leave out multi-valued attributes (if any)
- Pick up appropriate PK



Instructor DDL

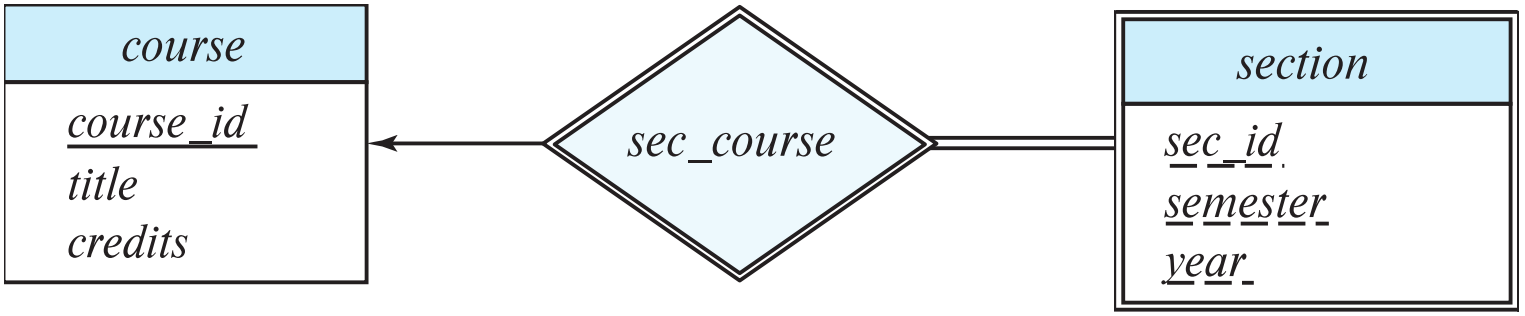
CREATE TABLE Instructor

(I_ID CHAR(20)
name CHAR(20),
dept_name CHAR(10),
salary (INTEGER)

PRIMARY KEY (I_ID))

Step 2 : Weak Entity

- Create a new relation
- Include all attributes
- Include FK pointing towards owner relation
- Example: adding ‘C_ID’ to the ‘section’ relation
- PK is the combination of the FK and the partial key
- **Section**(***C_ID, Sec_ID, semester, year, building, room_number***)



Weak Entity sets

- Weak entity set (section) and identifying relationship set (course) are translated into a single table Section_sec_course
- When the owner entity is deleted, all owned weak entities must also be deleted.

```
CREATE TABLE  Section_Sec_course (
 Sec_ID CHAR(20),
Semester CHAR (20),
Year CHAR (20),
Building CHAR(20),
Room number INTEGER,
C_ID  INTERGER  NOT NULL,
   PRIMARY KEY  (sec_ID, semester, year, C_ID),
   FOREIGN KEY  (C_ID) REFERENCES Course,
     ON DELETE CASCADE)
```

Cascading Actions in Referential Integrity

- When a referential-integrity constraint is violated, the normal procedure is to reject the action that caused the violation.
- An alternative, in case of delete or update is to cascade

```
create table course (
        (…
         dept_name varchar(20),
         foreign key (dept_name) references department
              on delete cascade
              on update cascade,
          . . .)
```

- Instead of cascade we can use :
  - **set null**,
  - **set default**

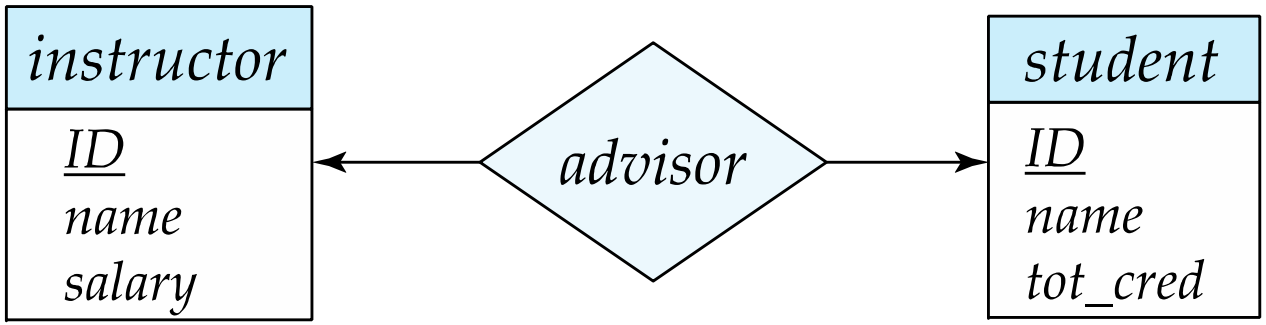
Course(courseID, title,dept_name, credits)

Department (dept_name, building, budget)

Step 3 : 1:1 Relationship

- Add to one of the participating relation a FK to other relation
- It is better to add to the relation that has a total participation in the relationship
- Include any relationship attributes
- Example: consider **instructor-advisor-student**
- Every student has at most one advisor and an instructor advises at most one student

- **Stu_advisor** ***(s_ID****, s_name, dept_name, tot_credit, I_ID, since)*
- Foreign Key *I_ID* references to **Instructor**

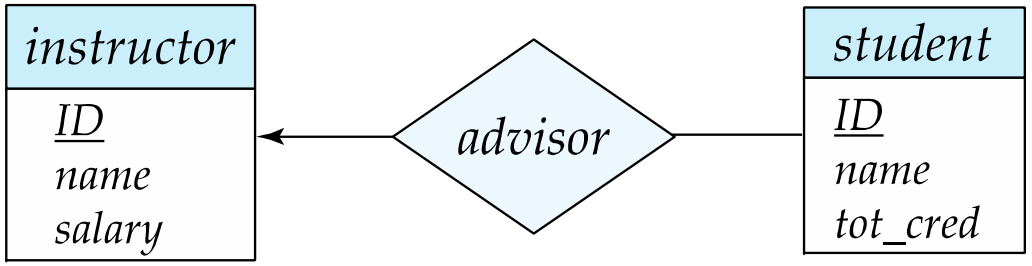


Every student must have an advisor

Step 4 : 1:N Relationship

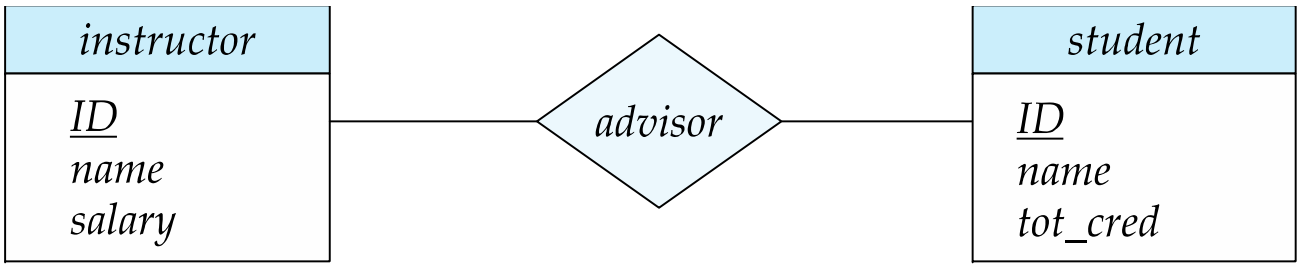
- Add to the relation a FK from the other side
- Include any relationship attributes
- Example: instructor-advisor-student
- 1:N
- Add to student relation I_ID
- Add attribute since
- Student (***S_ID***, *S_name, dept_name, total_cred, I_ID, since*)

  FK (I_ID) references Instructor



Step 5 : M:N Relationship

- Create a new relation containing FKs to both the participating relations
- Add relationship attributes (since)
- PK is the combination of both the FKs
- Instructor-advisor-student
- student can have many advisors and instructor can advise many students
- **Advisor**(***I_ID, S_ID***, *since)*



Step 6 : Multivalued Attributes

- Create a new relation
  - containing FK to the entity containing it
  - Attribute or attributes
- PK is FK plus the attributes

- Instructor (***I_ID,*** *Name, Dept_name, salary, {phone_no})*

- *Inst_ph (* ***I_ID, ph_no****)*

- It is all key relation
- FK (I_ID) references Instructor)

Step 7 : Ternary Relationship

- New relation containing a foreign key referencing each of the 3 entities
- Include relationship/descriptive attributes (if any)

Proj_guide (instructor, project, student)

- Proj_guide has descriptive attribute (since)
- **Proj_guide (I_ID, P_ID, S_ID, since)**