



ESCUELA
POLITÉCNICA
NACIONAL



BASES DE DATOS

Dra. Rosa Navarrete



Unidad 2. Video 1

Modelo Entidad - Relación

- Fases del Diseño
- Esquema del Modelo ER
- Diagramas ER

Fases del diseño

- La fase inicial del diseño de la base de datos es caracterizar completamente las necesidades de datos de los posibles usuarios de la base de datos.
- Luego, el diseñador elige un modelo de datos y, aplicando los conceptos del modelo de datos elegido, traduce estos requisitos en un esquema conceptual de la base de datos.
- Un esquema conceptual completamente desarrollado también indica los requisitos funcionales de la empresa. En una "especificación de requisitos funcionales", los usuarios describen los tipos de operaciones (o transacciones) que se realizarán en los datos.

Fases del diseño (Cont.)

El proceso de pasar de un modelo de datos abstractos a la implementación de la base de datos procede en dos fases finales de diseño.

- Diseño lógico: decisión sobre el esquema de la base de datos. El diseño de la base de datos requiere que encontremos una "buena" colección de esquemas de relación.
 - Decisión comercial: ¿qué atributos debemos registrar en la base de datos?
 - Decisión de la informática: ¿qué esquemas de relación deberíamos tener y cómo deberían distribuirse los atributos entre los diversos esquemas de relación?
- Diseño físico: decisión sobre el diseño físico de la base de datos

Enfoques de diseño

□ Modelo de entidad-relación

- Modela una empresa como una colección de entidades y relaciones
 - ▶ Entidad: una "cosa" u "objeto" en la empresa que se distingue de otros objetos, que es caracterizable (puede ser descrita). No necesita ser tangible, puede ser un “concepto”.
 - ▶ Descrito por un conjunto de atributos
- Relación: una asociación entre varias entidades.
- Representado esquemáticamente por un diagrama entidad-relación

□ Teoría de la normalización

- Formalizar diseño relacional

Esquema del Modelo ER

Modelo ER-- Modelamiento

- ❑ El modo de datos ER fue propuesto por Peter Chen (1976).
- ❑ Se desarrolló para facilitar el diseño de la base de datos al permitir la especificación de un esquema empresarial que representa la estructura lógica general de una base de datos.
- ❑ El modelo ER es muy útil para mapear los significados e interacciones de las empresas del mundo real en un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en conceptos del modelo ER.
- ❑ El modelo de datos ER emplea tres conceptos básicos:
 - ❑ conjuntos de entidades,
 - ❑ conjuntos de relaciones,
 - ❑ atributos.
- ❑ El modelo ER también tiene una representación esquemática asociada, el diagrama ER, que puede expresar la estructura lógica general de una base de datos gráficamente.

Entity Sets

- Una **entidad (entity)** es un objeto que existe y es distinguible de otros objetos. Es decir se puede describir y reconocer.
 - Ejemplo: estudiante, empresa, evento, factura, libro
- Un **conjunto de entidades (entity set)** es la colección de entidades del mismo tipo que comparten las mismas propiedades.
 - Ejemplo: conjunto de todos los estudiantes, conjunto de todos los libros...
- Una entidad está representada por un conjunto de atributos; es decir, propiedades descriptivas que poseen todos los miembros de un entity set.
 - Ejemplo:
$$\text{instructor} = (ID, name, street, city, salary)$$
$$\text{course} = (course_id, title, credits)$$
- Una **clave primaria (primary key)** del entity set es un subconjunto de los atributos; que identifica de manera única a cada miembro del conjunto.

Entity Sets -- *instructor* and *student*

instructor_ID instructor_name

76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

student-ID student_name

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Relaciones (Relationship Sets)

- Una **relación (relationship)** es una asociación entre varias entidades.

Ejemplo:

44553 (Peltier)	(tutor) <u>advisor</u>	22222 (<u>Einstein</u>)
<i>student</i> entity	relationship set	<i>instructor</i> entity

- Un **relationship set** es una relación matemática entre $n \geq 2$ entidades, cada una tomada de su respective entity set:

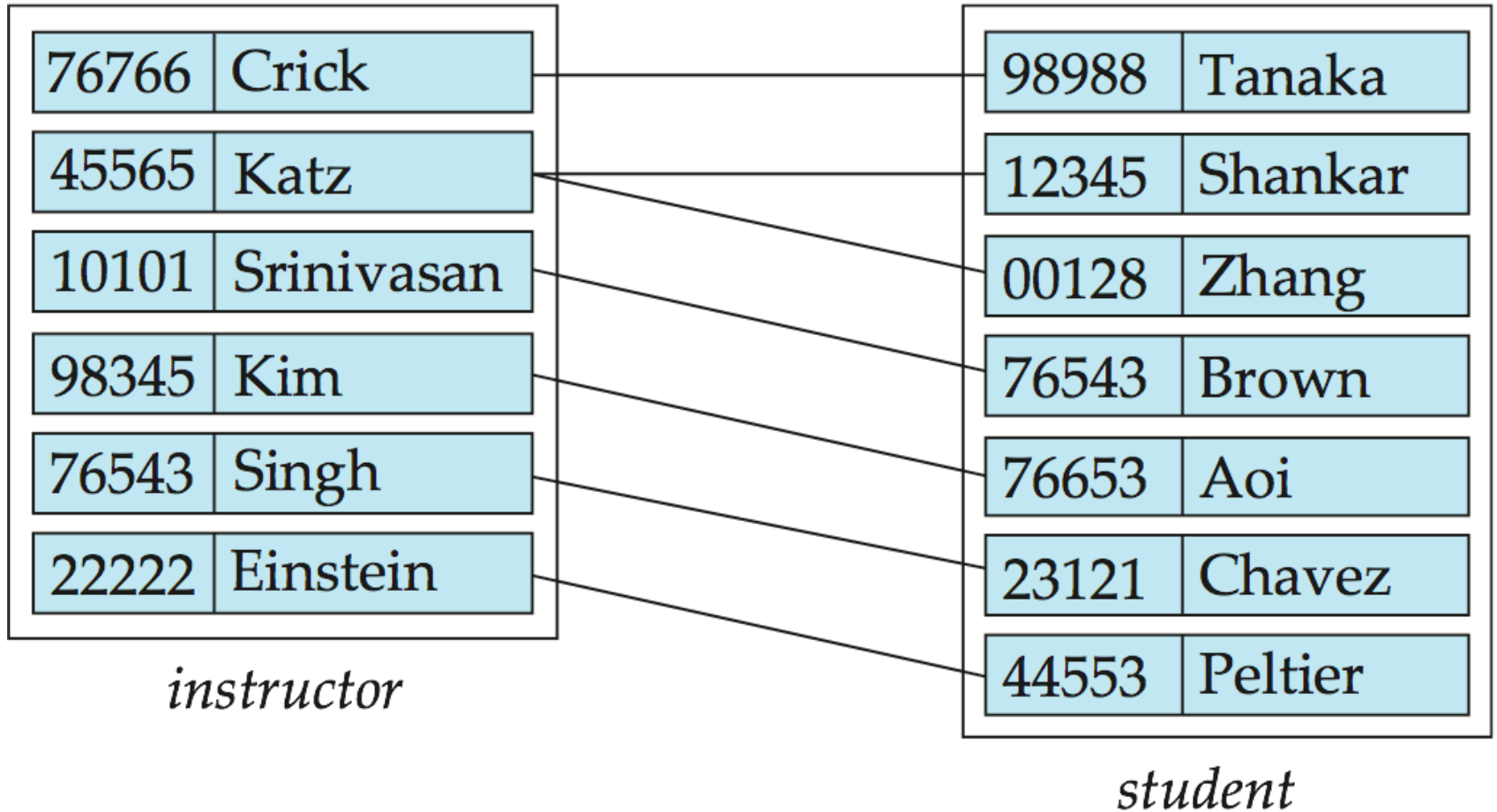
$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

donde (e_1, e_2, \dots, e_n) es una relación

- Ejemplo:

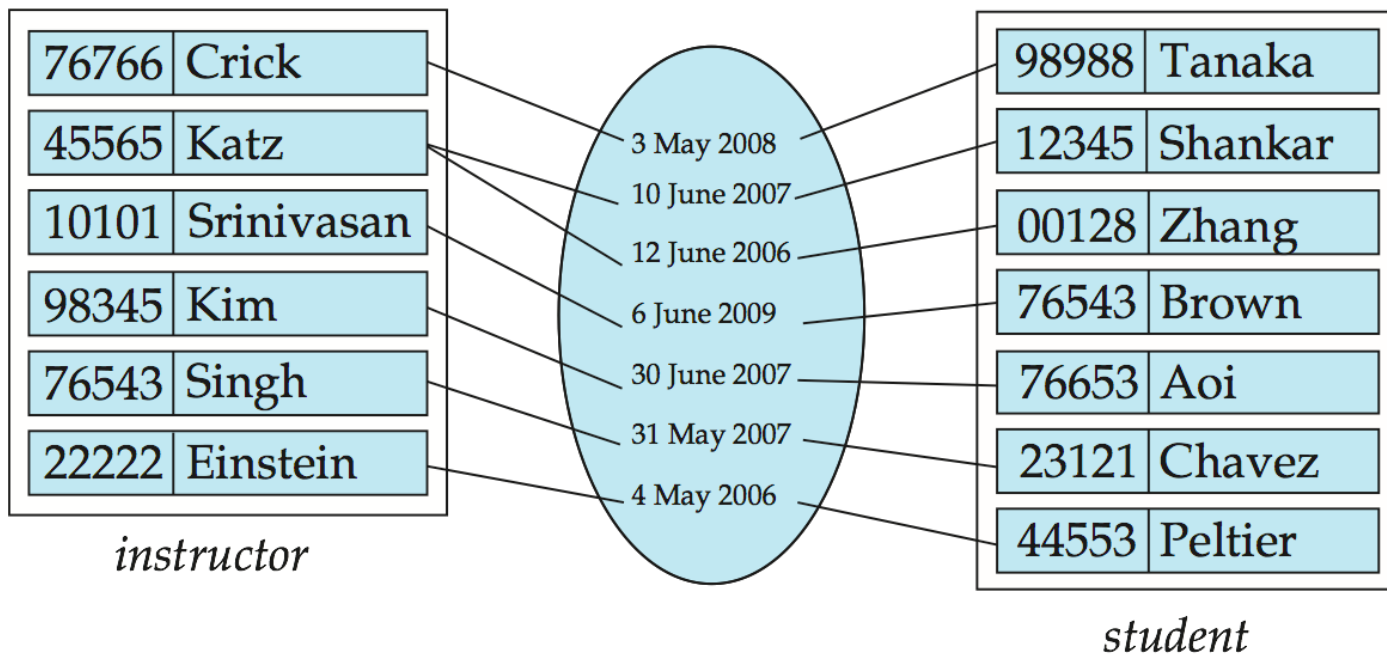
$$(44553, 22222) \in \text{advisor}$$

Relationship Set *advisor*



Relationship Sets (Cont.)

- Un atributo también se puede asociar con un conjunto de relaciones.
- Por ejemplo, la relación *advisor* entre los entity sets *instructor* y *student* puede tener como atributo la fecha (*date*) desde la cual el estudiante se asoció a un tutor.



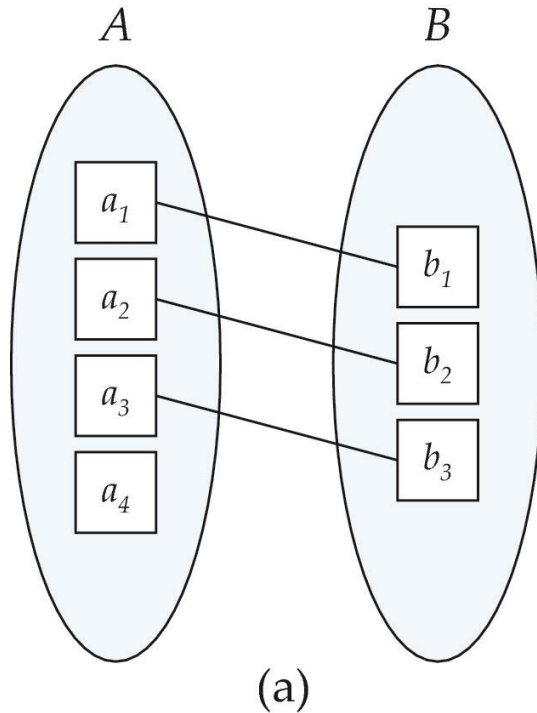
Grado de un Relationship Set

- Relaciones binarias (binary relationship)
 - Involucra dos entity sets (o es de grado dos).
 - Se espera que las relaciones en una base de datos sean mayoritariamente binarias o puedan ser representadas como relaciones binarias.
 - ▶ Ejemplo: *students* work en proyectos (*projects*) bajo la *supervision* de un *instructor*.
 - ▶ La relación *proj_guide* es una relación ternaria entre *instructor*, *student*, and *project*

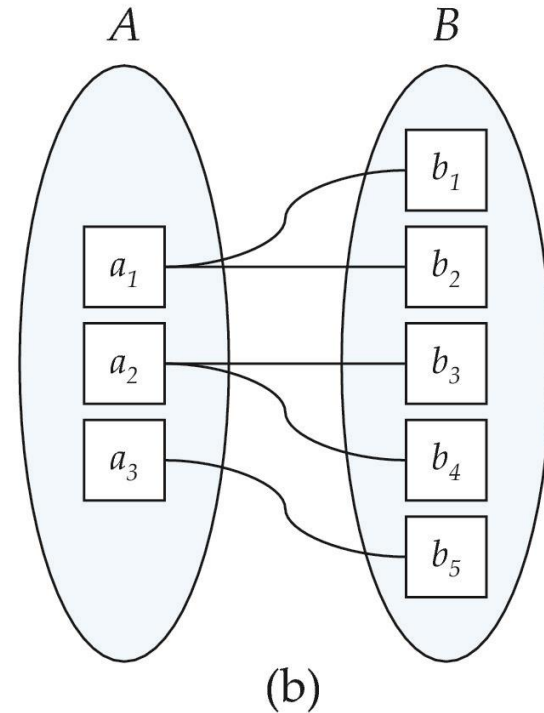
Mapeo de restricciones de cardinalidad

- Las restricciones de cardinalidad expresan el número de entidades a las que se puede asociar otra entidad a través de un conjunto de relaciones.
- Para una relación binaria (dos entity set que se relacionan), la cardinalidad puede ser
 - Una a una - One to one
 - Una a muchos - One to many
 - Muchos a una - Many to one
 - Muchos a muchos - Many to many

Maapeo de cardinalidades



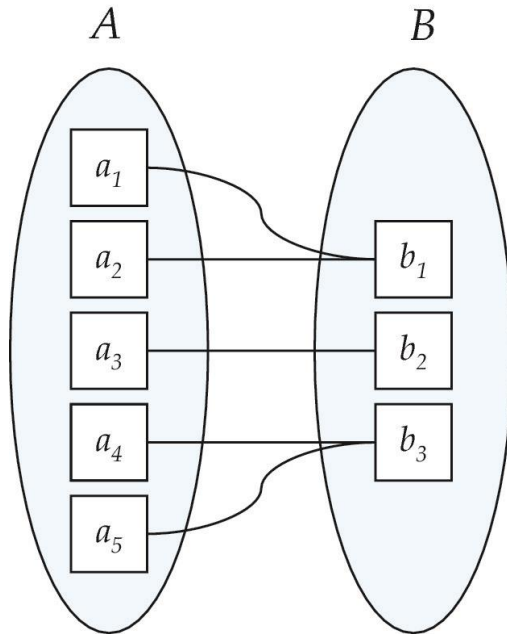
One to one



One to many

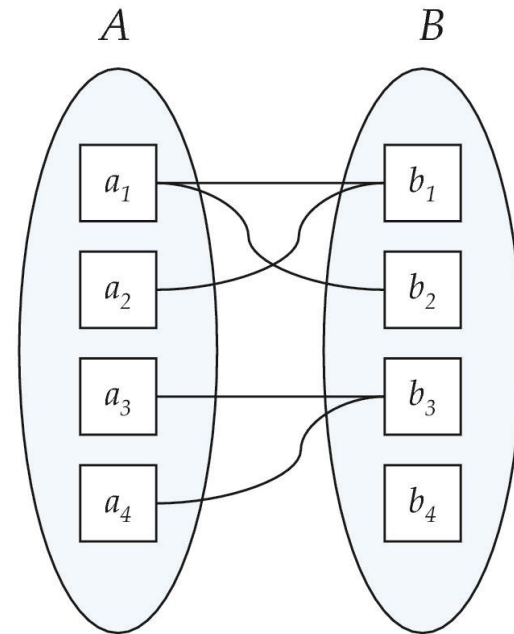
Nota: Algunos elementos en A y B pueden no estar asignados a ningún elemento en el otro conjunto

Mapeo de cardinalidades



(a)

Many to
one



(b)

Many to many

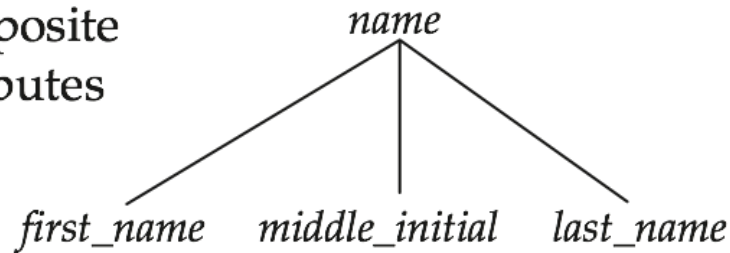
Nota: Algunos elementos en A y B pueden no estar asignados a ningún elemento en el otro conjunto

Tipos de Atributos

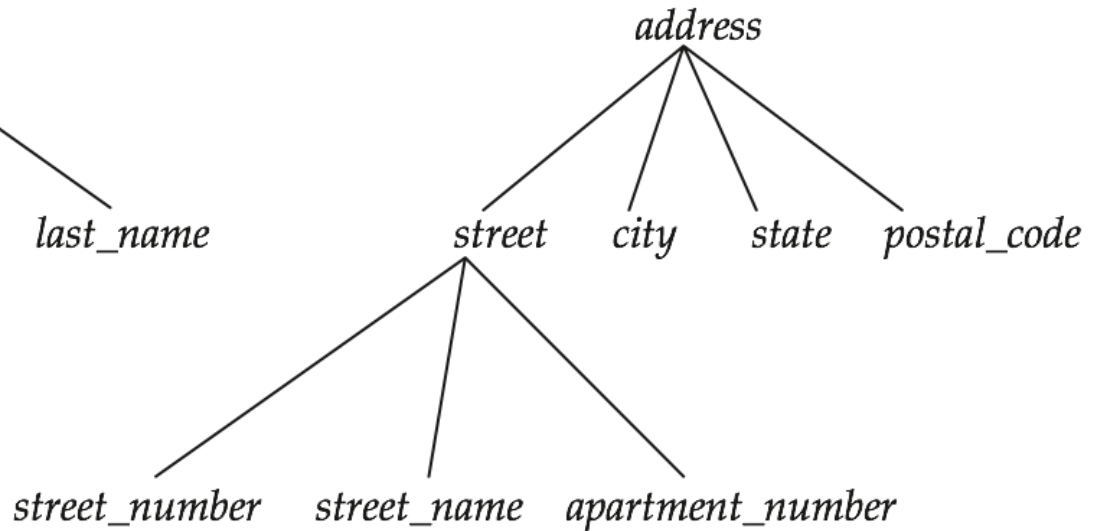
- Tipos de atributos:
 - **Simple** y **compuesto**.
 - **Monovaluado** y **multivaluado**
 - ▶ Ejemplo: atributo monovaluado (valor único): *sueldo*, *placa_auto* ,
 - ▶ atributo multivaluado (*más de un valor*): *numero_telefono*
 - **Derivado**
 - ▶ Puede ser calculado a partir de otro u otros atributos
 - ▶ Ejemplo: atributo edad, dado fecha_nacimiento
- **Dominio (Domain)** – conjunto de valores permitidos para cada atributo.
 - Ejemplo: nombre_estudiante (cadena de caracteres)
 - Fecha_Nacimiento (valor admisible de fecha)
 - Sueldo: Valor numérico (entre 450 y 1700)

Atributos compuestos

composite
attributes



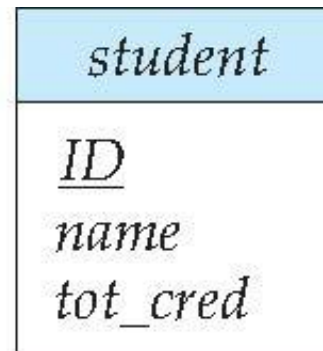
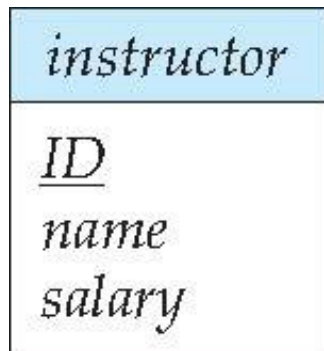
component
attributes



Diagramas E-R

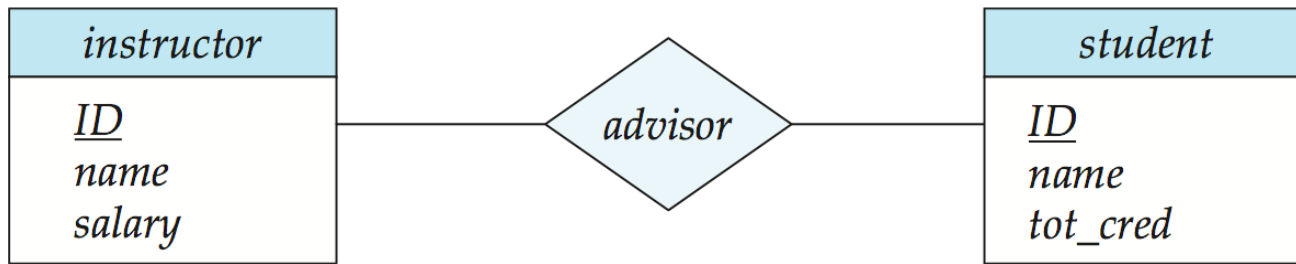
Clave primaria (Primary Key)

- ❑ Las entidades se representan con un rectángulo en cuya parte superior se identifica el nombre de la entidad.
 - ❑ Los atributos se listan dentro del rectángulo.
 - ❑ Los atributos que forman la clave primaria se subrayan.
 - ❑ La clave primaria puede ser un único atributo (deseable) o un conjunto de atributos, pero siempre, el mínimo conjunto, de tal forma que no existan atributos redundantes; es decir, aquellos que no contribuyen a la clave.



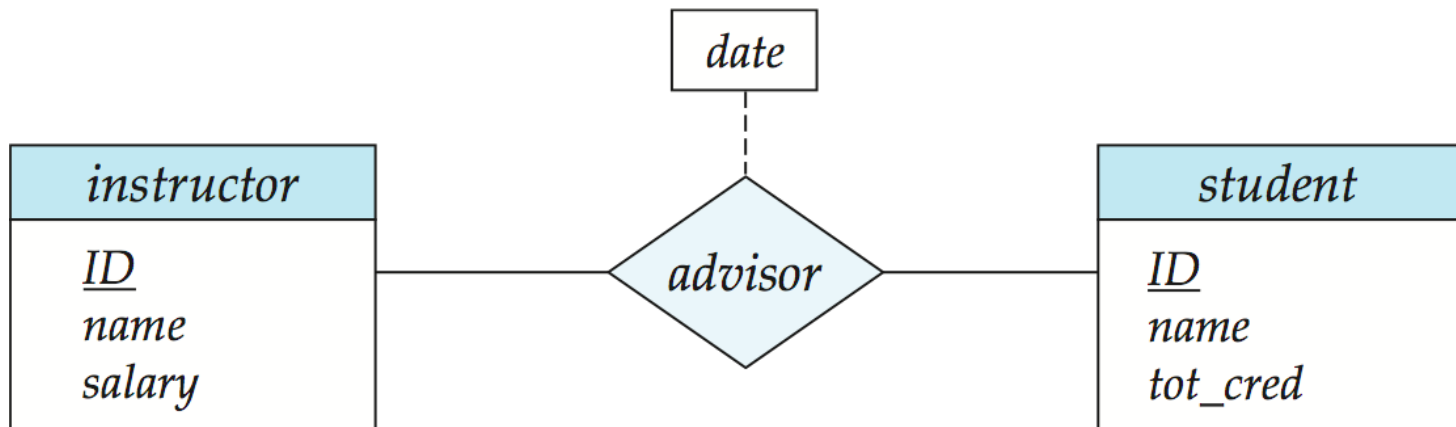
Relationship Sets

- Las relaciones (relationship sets) se representan con un rombo, en cuyo interior se escribe el nombre dado a la relación.



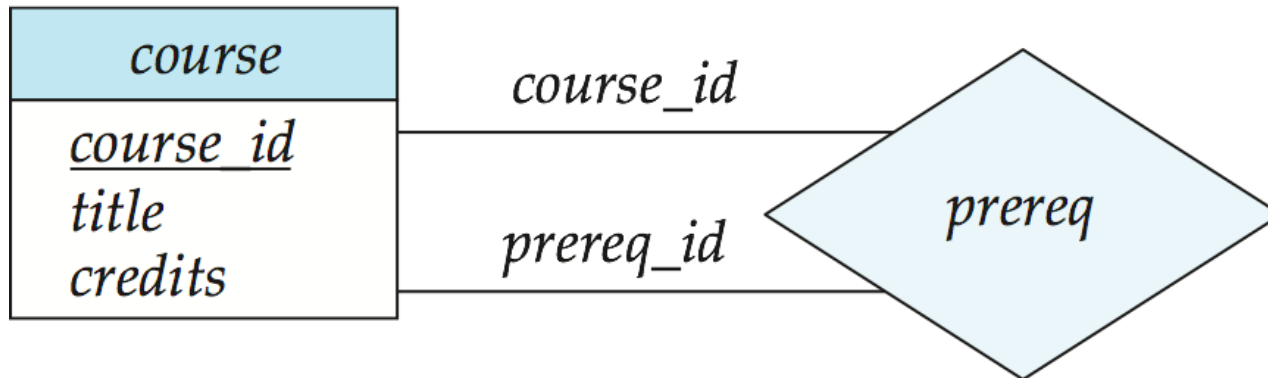
Conjunto de relaciones con atributos

- Si un conjunto de relaciones (relationship set) tiene atributos, estos aparecen en rectángulos unidos a la relación con una línea entrecortada.



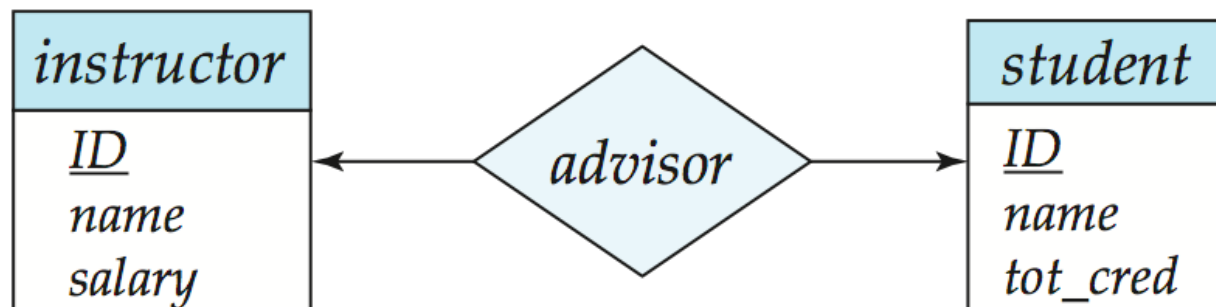
Roles

- ❑ Puede haber casos en que las relaciones muestren que una instancia de la relación se relaciona con otra u otras instancias de la misma relación. Esto representa una **relación recursiva**.
- ❑ Las entity set no necesariamente deben ser distintas.
 - ❑ Cada ocurrencia del entity set juega un “rol” en la relación.
- ❑ Las etiquetas “*course_id*” y “*prereq_id*” se llaman **roles**.



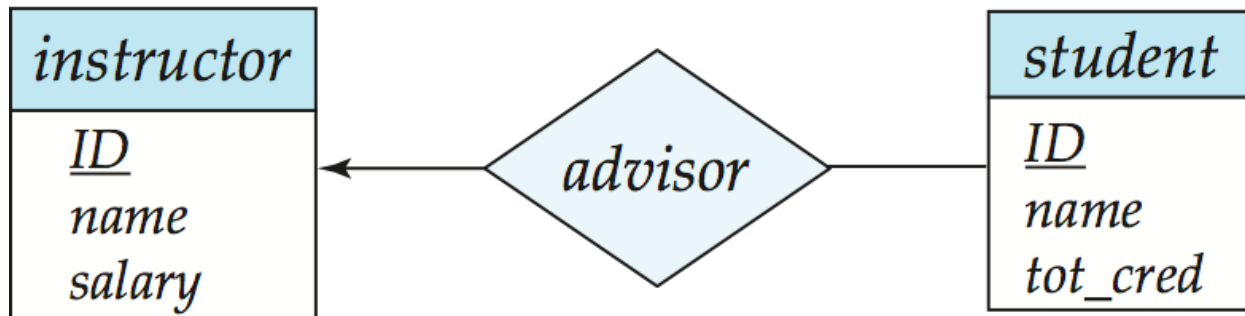
Restricciones de cardinalidad

- Expresamos restricciones de cardinalidad dibujando una línea dirigida (\rightarrow), que significa "uno", o una línea no dirigida ($—$), que significa "muchos", entre el conjunto de relaciones y el conjunto de entidades.
- Las relaciones Uno a Uno (One-to-one) entre un *instructor* y un *student* :
 - Un *student* se asocia con máximo un *instructor* via la relación *advisor*
 - Un *student* se asocia con máximo un *department* via la relación *stud_dept*



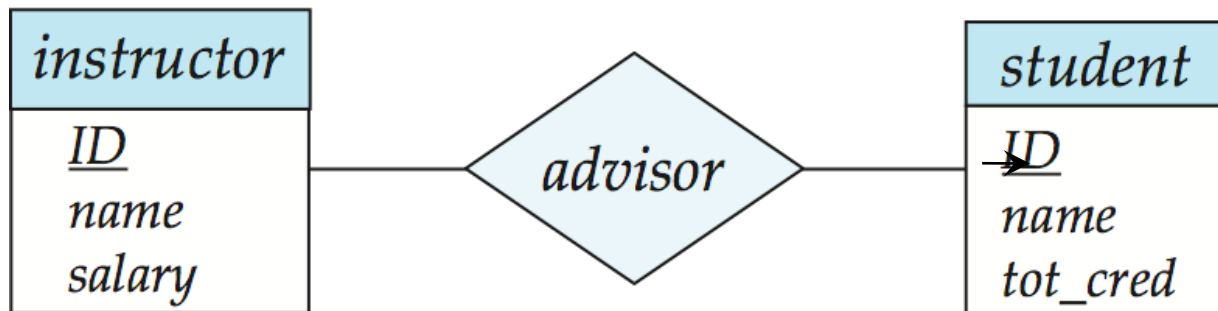
Relación Uno a Muchos (One-to-Many)

- Relación Uno a Muchos (one-to-many) entre un *instructor* y un *student*
 - Un *instructor* se asocia con muchos (incluso 0) *student* via la relación *advisor*
 - Un *student* se asocia al menos un *instructor* via la relación *advisor*



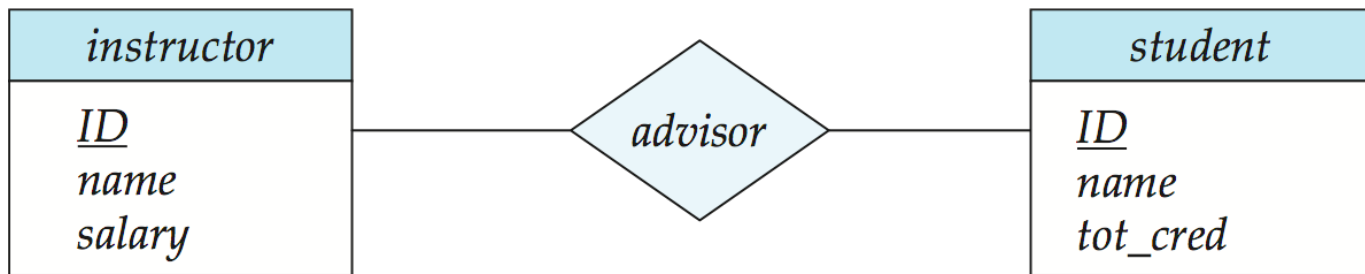
Relación Muchos a Uno (Many-to-One)

- En una relación Muchos a uno (many-to-one) entre un *instructor* y un *student*,
 - Un *instructor* se asocia con máximo un *student* via *advisor*,
 - Un *student* se asocia con muchos (incluso 0) *instructor* via *advisor*



Relación Muchos a Muchos (Many-to-Many)

- Un *instructor* se asocia con muchos (posiblemente 0) *student* via *advisor*
- Un *student* se asocia con muchos (posiblemente 0) *instructor* via *advisor*



Participación Total y Parcial en la relación

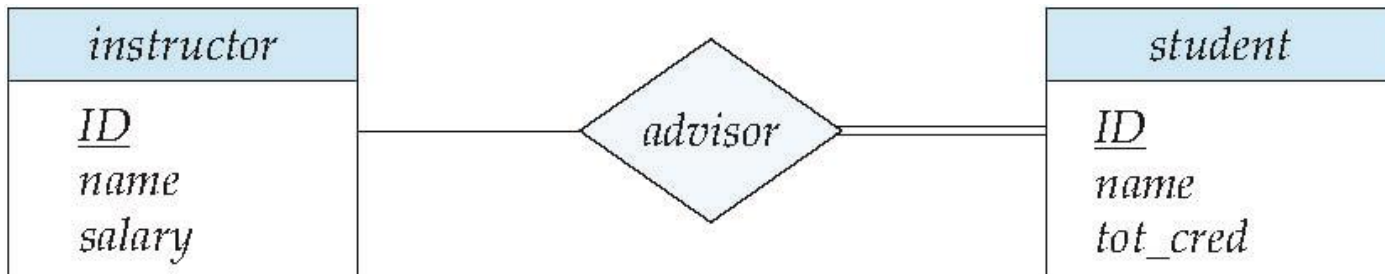
- La **participación total** (indicada con línea doble): cada entidad en el entity set participa en al menos una relación del relationship set.

La participación de *student* en la relación *advisor* relation es **total**

- ▶ Cada *student* **tiene que** estar asociado a un *instructor*

- La **participación parcial** (indicada con línea simple): algunas entidades pueden no participar en ninguna relación en el conjunto de relaciones

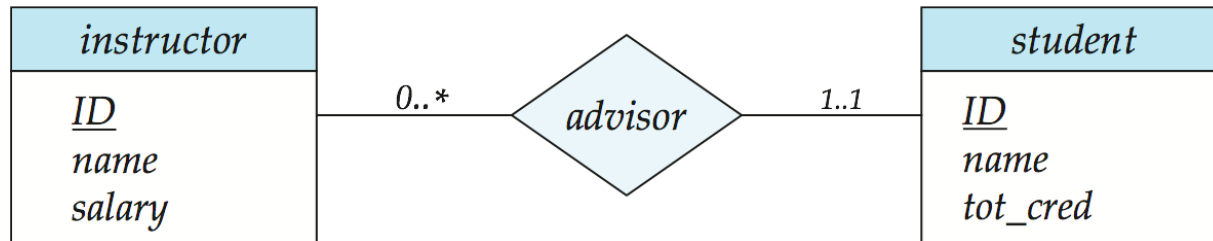
- ▶ La participación de *instructor* en la relación *advisor* es parcial. Un instructor **puede no estar asociado** con algún *student*



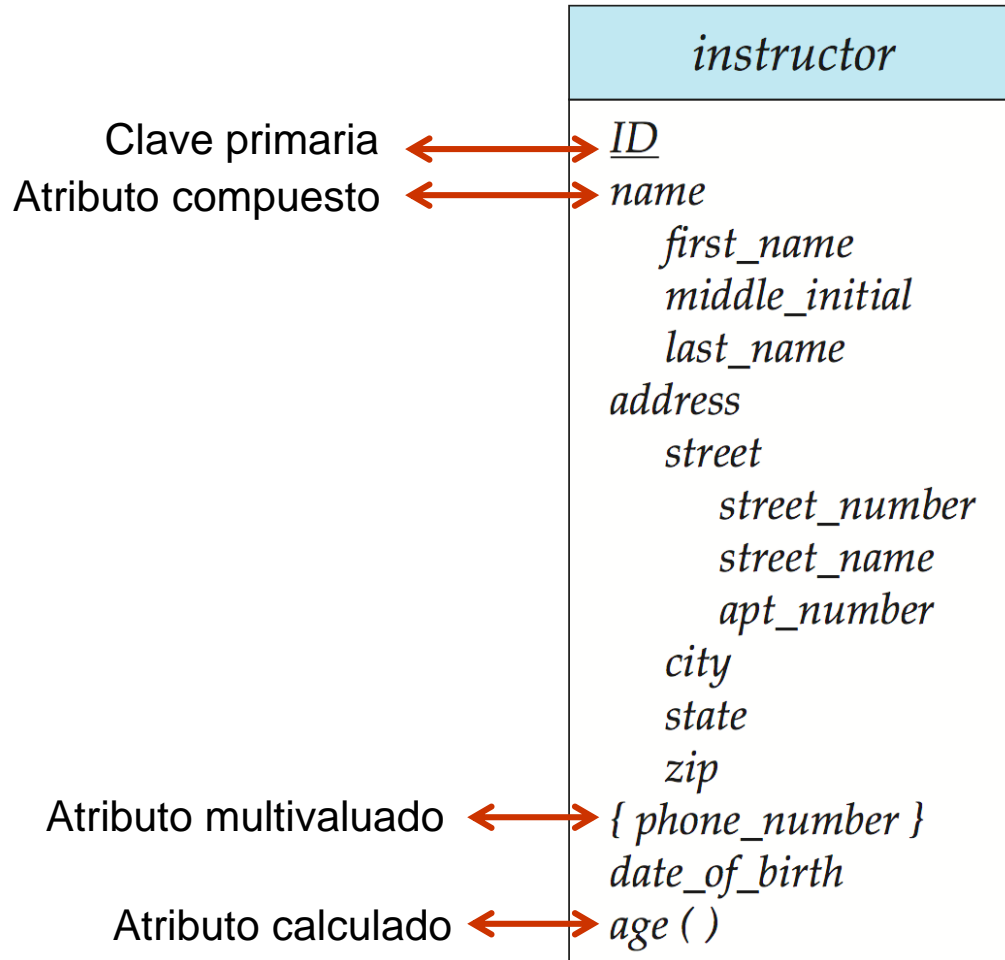
Notación para expresar cardinalidades complejas

- Se puede anotar una cardinalidad mínima y máxima asociada, que se muestra en la forma ***l..h***, donde *l* es la cardinalidad mínima y *h* la máxima.
- Si el valor mínimo es 1, entonces la participación es total (indica mínimo 1).
 - Un máximo valor de 1 indica que la entidad participa en como máximo 1 relación.
 - Un valor máximo representado con * indica que no hay límite.

Ejemplo: Un *instructor* puede tutoriar a 0 o más *student*. Un *student* debe estar asociado con 1 *instructor* para la tutoría. No puede tener muchos *instructor* asociados.



Notación de atributos

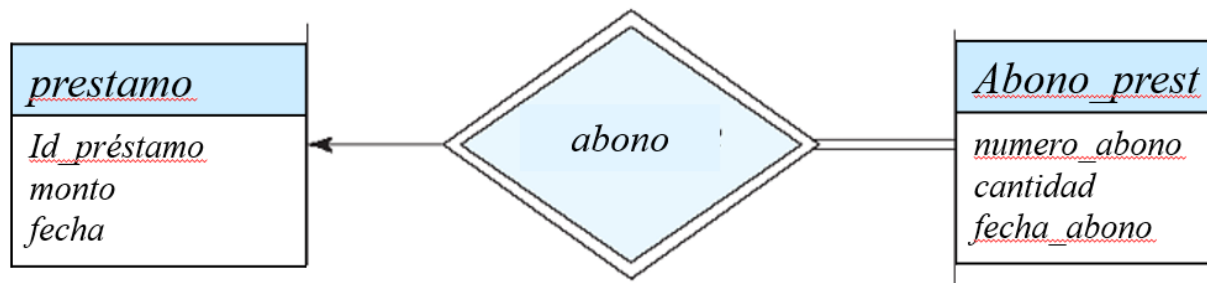


Entidades débiles

- Un conjunto de entidades débiles es aquel cuya existencia depende de otra entidad, llamada entidad de identificación.
- En lugar de asociar una clave primaria con una entidad débil, usamos la entidad de identificación, junto con atributos adicionales llamados discriminadores para identificar de manera única una entidad débil.
- Un conjunto de entidades que no es un conjunto de entidades débiles se denomina conjunto de entidades fuertes.

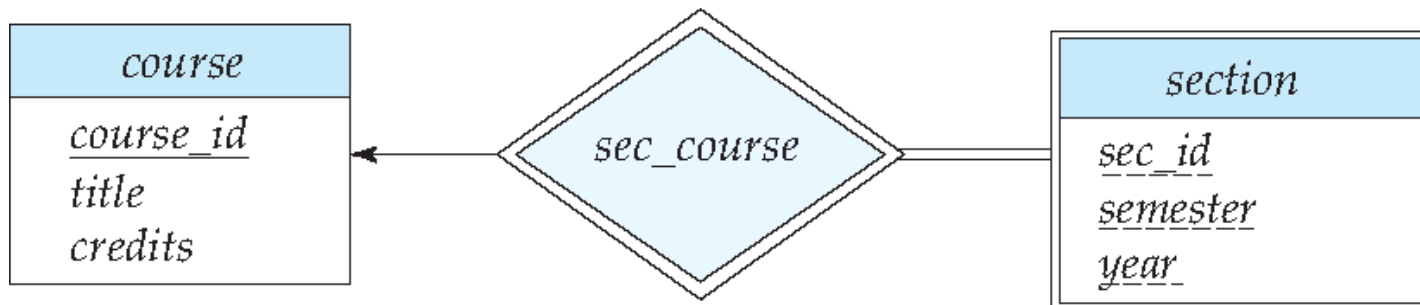
Entidades débiles

- ❑ Por ejemplo: Considere el conjunto de entidad, **Préstamo** y el conjunto de entidad **Abono_Prestamo**
- ❑ Si la entidad Prestamo tiene una clave primaria, esto identifica de forma única a cada préstamo.
- ❑ Pero, en el conjunto de entidad Abono_Prestamo, algunas entidades se van a referir a pagos del mismo préstamo.
- ❑ Para identificar de forma única a cada Abono_Prestamo hay que pedir prestado la clave primaria de Prestamo y agregar un atributo discriminador, que identifique a cada Pago de préstamo.

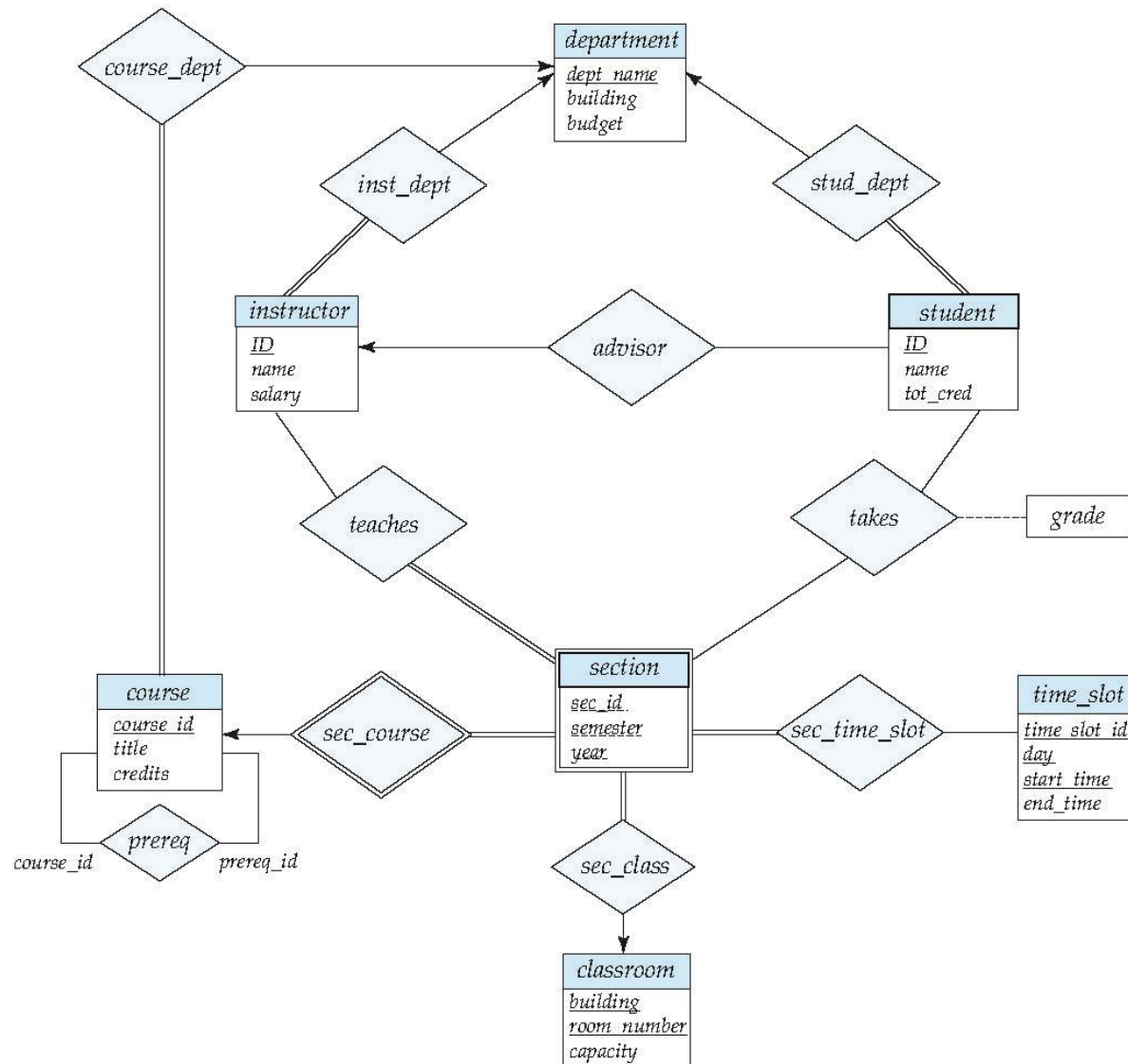


Expressing Weak Entity Sets

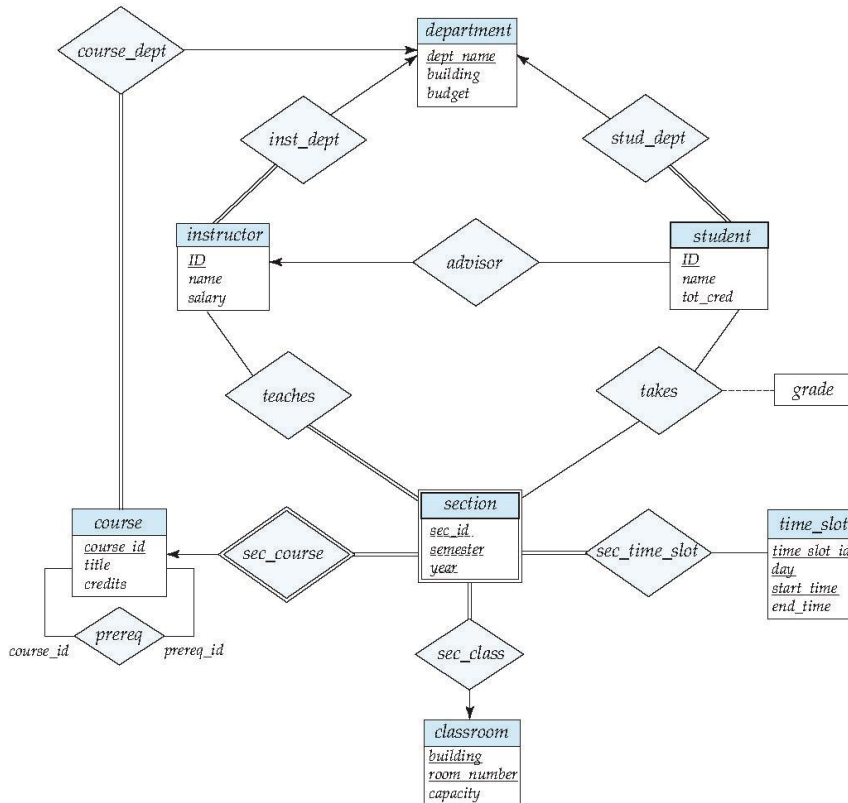
- En los diagramas E-R, una entidad débil se dibuja con un rectángulo de doble línea.
- Se identifica el discriminante de la entidad débil subrayándolo con una línea entrecortada.
- La relación que conecta la entidad débil con la entidad fuerte se dibuja con un rombo de doble línea.
- Primary key for *section* – (*course_id*, *sec_id*, *semester*, *year*)



E-R Diagram for a University Enterprise



E-R Diagram for a University Enterprise



- En que set entity se muestra una relación recursiva: *course* (un curso es prerequisite de otro curso)
- Todo *student* DEBE ESTAR relacionado con un *department*
- A un *department* se puede asociar muchos *student*
- Todo *instructor* DEBE ESTAR asociado con un *department*
- La única entidad débil es *section*

Gracias

Unidad 2: Video 1