Databases and Information Systems
CS303

Database design : ER Diagrams
31-08-2023

Database Design

- Clients come to the software designer with informal set of requirements
- Database architect should design the database schema before implementing it

Database Design Process

- Understand the data needs of prospective users and come up with specifications of user requirements (either diagrammatically or textually)

- Choose an appropriate data model and translate the requirements to a conceptual schema
    - For Relational databases conceptual schemas are represented as Entity-Relationship diagrams
    - ER Diagrams : Identify the entities and the relationships among them

- Ensure that the conceptual schema supports all functional requirements

- Implementation of the database
    - Logical Design Phase: Convert ER diagram into a relational schema
    - Physical Design Phase: Physical features of the database are implemented (file organization, index, data structures...)

Major Pitfalls

- Data Redundancy : Unnecessarily storing the same data multiple times
    - Causes Inconsistency during updates
    - Normalize the relational schema if there is redundancy

- Data Incompleteness : Inability to perform some functional requirements

- Choose wisely among the ‘good options’

Entity Relationship Diagrams

Entity Relationship Diagrams

- ER Diagrams : Represents overall logical structure of the database
  - Entity Sets
  - Relationship sets
  - Attributes

Entity sets

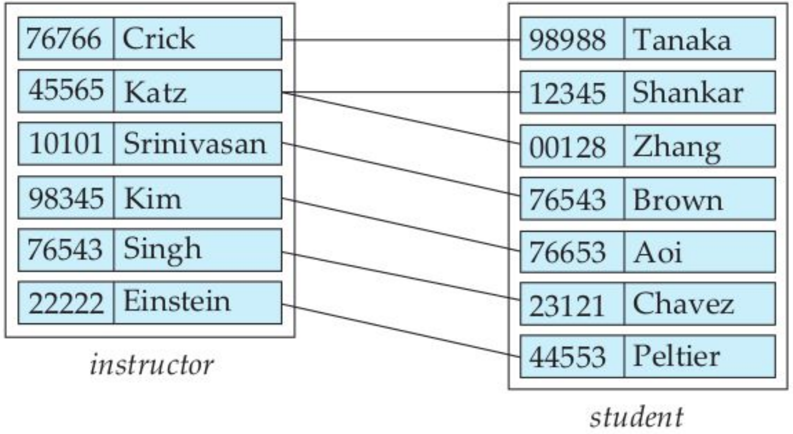
- Entity is anything from the real world
  - Every instructor, student, course etc in the university database
- Entity set is a set of entities of the same type (that share same properties)
  - Course, Instructor, Student
- Entity set is abstract (does not refer to any particular entity)
  - Similar to Class in Object Oriented Programming
- Extensions of the entity set refers to the actual collection of entities of a particular entity set
  - Similar to the set of all Objects of a Class in Object Oriented Programming

Entity sets

- Entity sets need not be disjoint
  - Person entity may be a part of Instructor entity or Student entity
- Entity is represented by a set of attributes
  - Instructor entity has attributes ID, name, dept_name, salary
  - Course entity has attributes ID, title, dept_name, credits
  - Each entity has values for each attribute
- Database contains a collection of entities

Relationship sets

- Describes relationship among entities



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

instructor

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

student

Advisor relationship

Relationship sets

- Relationship of arity n >= 2 over n entities (not necessarily distinct) $E_1\ E_2\ E_3 \ldots\ldots E_n$ is given by

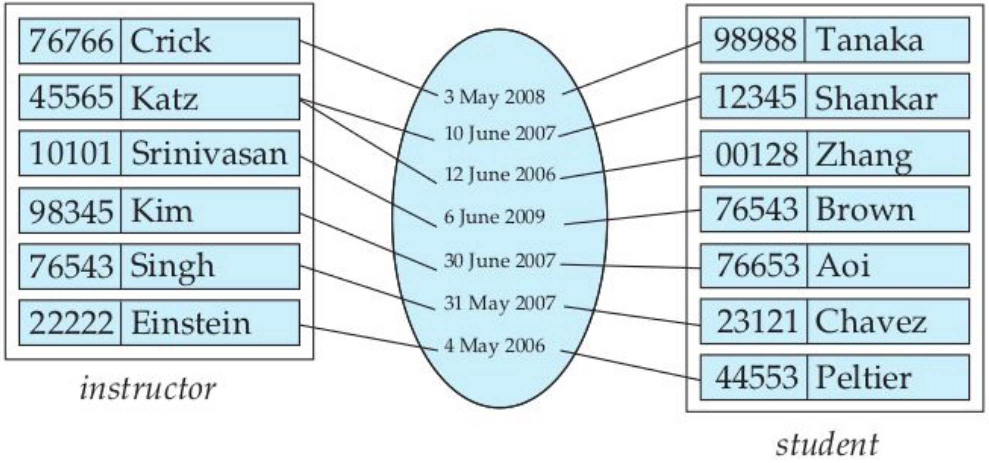
  $\{ (e_1\ e_2\ e_3 \ldots\ldots e_n) \mid e_1 \in E_1 ,\ e_2 \in E_2 \ldots\ldots e_n \in E_n \}$

  - Arity of a relationship : Number of participating entities

- Function that an entity plays in a relationship is called role
  - Roles are generally implicit and not explicitly specified
  - Might be harder in recursive relationships
    - Example : Prerequisite

Relationship sets

- Descriptive Attributes : Attributes of Relationships
  - Example: Date as an attribute for advisor-advisee relationship set



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

instructor

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

student

  - Example: Grade for takes relationship between student and section

Relationship sets

- Same entities can participate in different relationship sets

- Most practical relationship sets are binary
    - But occasionally there can be relationships that involve more than two entity sets
        - Example: Instructor(s) guiding student(s) on project(s) [Ternary relationship set]

Attribute

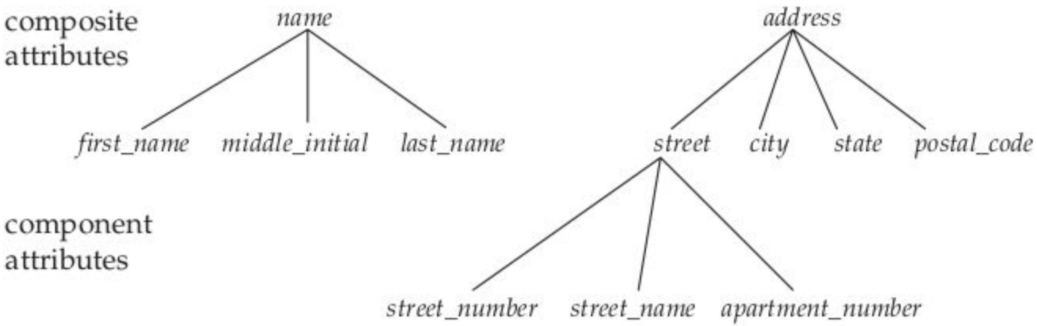
- Properties of Entity set or Relationship set
- Each attribute has a permitted set of values (domain)
- Attribute of an entity set is a function that maps the entities to domain

  Example: one tuple of instructor is:

  { (ID, 76766), (name, Crick), (dept_name, Biology), (salary, 72000) }

Attribute : Types

- Single / Composite :
  - Single: Cannot be divided into sub parts
  - Composite : Can be divided into sub parts.
    - Example:



Attribute : Types

- Single valued / Multi valued :
    - Single valued : An attribute that can take only one value
    - Multi valued : Phone_number attribute for student

      Multi valued attributes are denoted as sets {phone_number}

Attribute : Types

- Derived attribute :
    - Can be derived from other attributes stored
    - Not stored in the database, computed when required
        - Example : age (if date of birth is an attribute) ;
          number of students advised for an instructor

Attribute : Null

- An attribute for an entity can be null. It could mean:
    - Not applicable (Example: Thesis title for bachelor students)
    - Missing (Example: date of birth)
    - Not known (Example: House name in the address)

Constraints

- E-R diagrams allows us to specify constraints on the design
    - Mapping constraints
    - Participation constraints
    - Key constraints

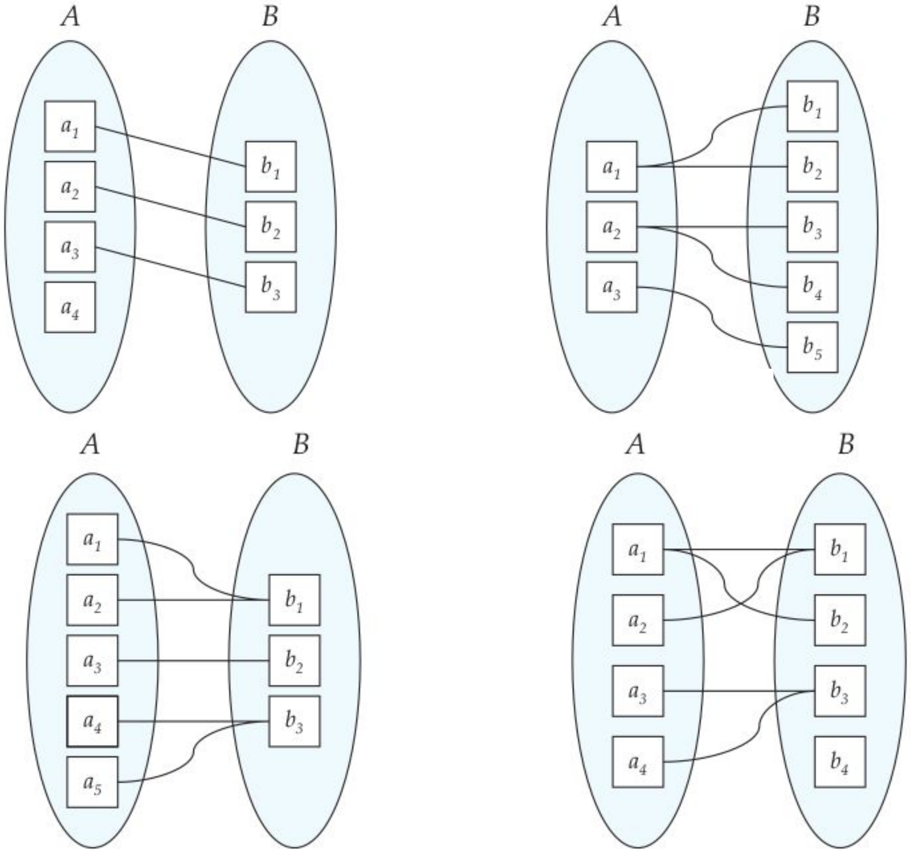
Constraints : Mapping constraints

- Mapping cardinalities express the number of entities to which another entity can be associated via a relationship set.

- Best suited to describe binary relationship sets

Constraints : Mapping constraints

- One-to-one
- One-to-many
- Many-to-one
- Many-to-many

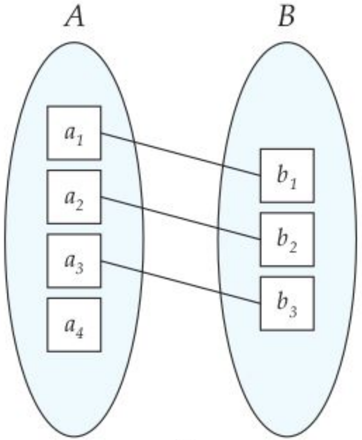


Constraints : Participating constraints

- Participation of an entity set E in a relationship set R is Total if every entity in E participates in at least one relationship in R.

- If only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial.

- Example:
  In advisor-advisee relationship
  Student participation is total;
  Instructor participation is partial



Constraints : Key constraints

- Two entities in an entity set are allowed to have the same values for all the attributes

- Notion of Keys also apply to entity sets (Super Key, Candidate Key, Primary Key)
  - Set of attributes that identify an entity uniquely
  - Primary Key of an entity uniquely identifies an en

Constraints : Key constraints

- Keys also identify relationships uniquely

- Designing Primary keys for relationship sets:
  - Suppose R is a relationship over the entities E_1 E_2 .... E_n then:
    - Attributes of R = primary-key(E_1) U primary-key(E_2) U .... U primary-key(E_n)
    - Primary Key of R = primary-key(E_1) U primary-key(E_2) U .... U primary-key(E_n)

- For Binary relationship R on E_1 and E_2, we can define primary key depending on the mapping constraints:

  - Many-Many : primary-key(E_1) U primary-key(E_2)
  - One-Many : primary-key (E_1)
  - Many-One : primary-key (E_2)
  - One-One : primary-key (E_1) or primary-key (E_2)

Overview of E-R diagram design

- Designing E-R diagram starts with the identification of the entity sets
- After this, the attributes of each entity set are identified
    - Choice of what attributes to include / what to leave out in the design is up to the designer who uses the domain knowledge
- In the next step the relationship sets are formed
    - This step may result in redundant attributes
- Example: Consider the entities instructor and department
    - instructor has attributes ID, name, dept_name, salary
    - department has dept_name, building and budget
    - A relationship set (inst_dept) associated each instructor to a department
    - Attribute dept_name appears in both entities
        - Primary key in department; so should be removed in instructor
    - dept_name gets added to instructor later (only if each instructor is part of single department)

Entities and relationships in the University database

- Entities:
  - classroom: with attributes (building, room number, capacity)
  - department: with attributes (dept name, building, budget)
  - course: with attributes (course id, title, credits)
  - instructor: with attributes (ID, name, salary)
  - section: with attributes (course id, sec id, semester, year)
  - student: with attributes (ID, name, tot cred)
  - time slot: with attributes (time slot id, {(day, start time, end time) })
- The relationship sets in our design are listed below:
  - inst_dept: relating instructors with departments
  - stud_dept: relating students with departments
  - teaches: relating instructors with sections
  - takes: relating students with sections, with a descriptive attribute grade
  - course dept: relating courses with departments
  - sec_course: relating sections with courses
  - sec_class: relating sections with classrooms
  - sec_time slot: relating sections with time slots
  - advisor: relating students with instructors
  - prereq: relating courses with prerequisite courses

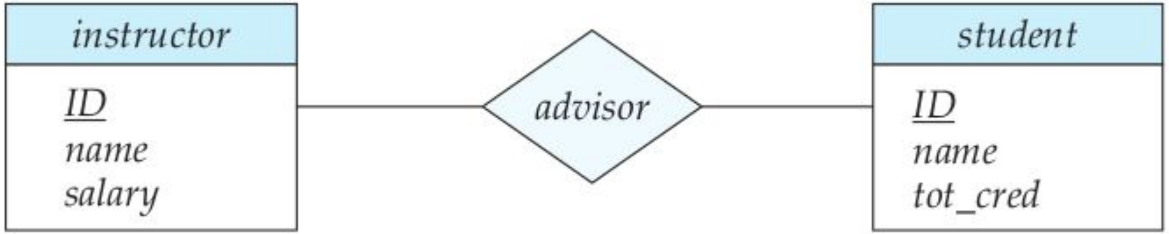
E-R diagrams

- Graphical representation of Entity sets and relationship sets

E-R diagram : Basic structures

- Rectangles:
  - Divided into two parts represent entity sets
    - First part contains the name of the entity set
    - Second part contains the names of all the attributes of the entity set
    - Attributes that are part of the primary key are underlined.

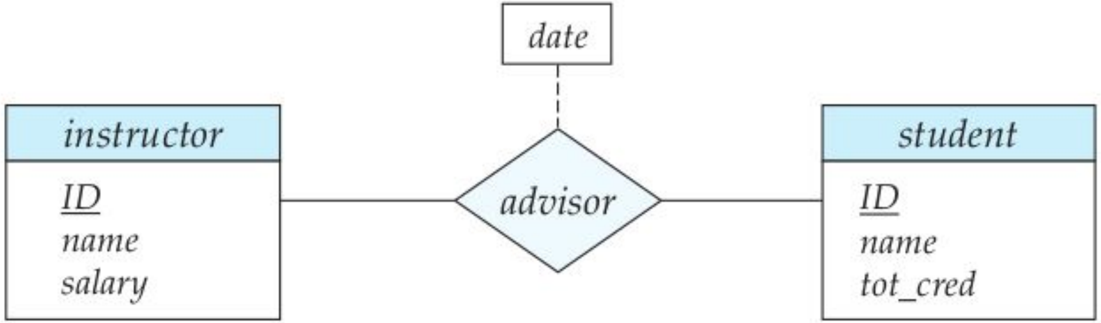
- Diamonds represent relationship sets



E-R diagram : Basic structures

- Lines link entity sets to relationship sets.

- Dashed lines link attributes of a relationship set to the relationship set.



E-R diagram : Basic structures

- Double lines indicate total participation of an entity in a relationship set.
- Double diamonds represent identifying relationship sets linked to weak entity sets

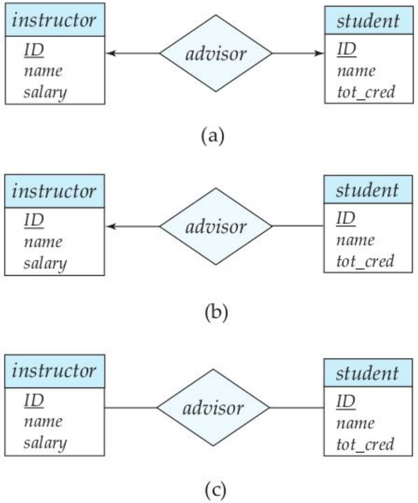
E-R diagram : Cardinality mapping

- Directed line for ‘One’ participation and Undirected line for ‘Many’ participation

- (a) One-to-One

- (b) One-to-Many

- (c) Many-to-Many



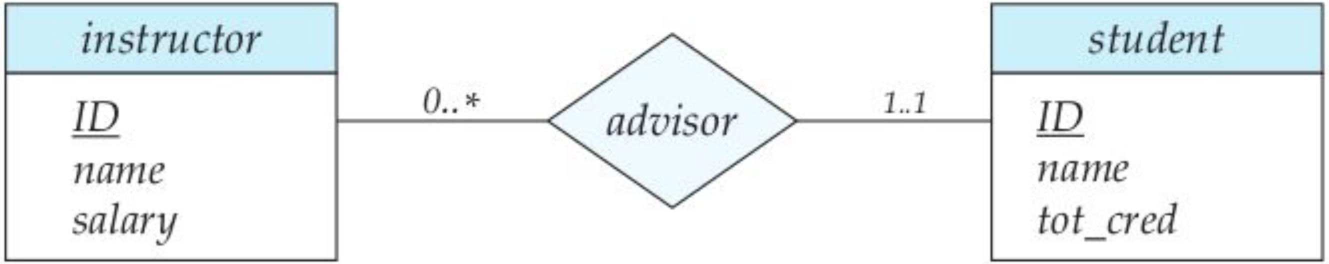
(a)

(b)

(c)

E-R diagram : Cardinality mapping

- Indicate m…n to mean at least m and at most n participations

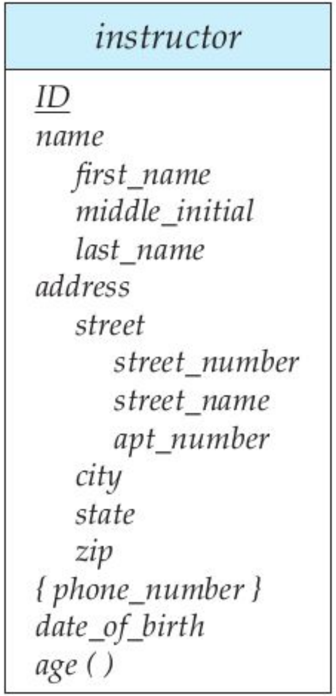


Complex attributes

- Complex attributes are indented

- Multi-valued attributes are represented in { }

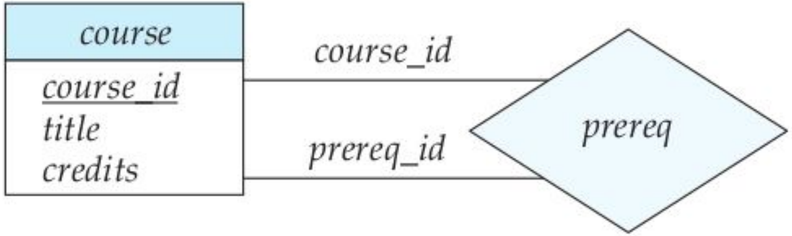
- Derived attributes have ( ) suffix



```
instructor
ID
name
    first_name
    middle_initial
    last_name
address
    street
        street_number
        street_name
        apt_number
    city
    state
    zip
{ phone_number }
date_of_birth
age ( )
```

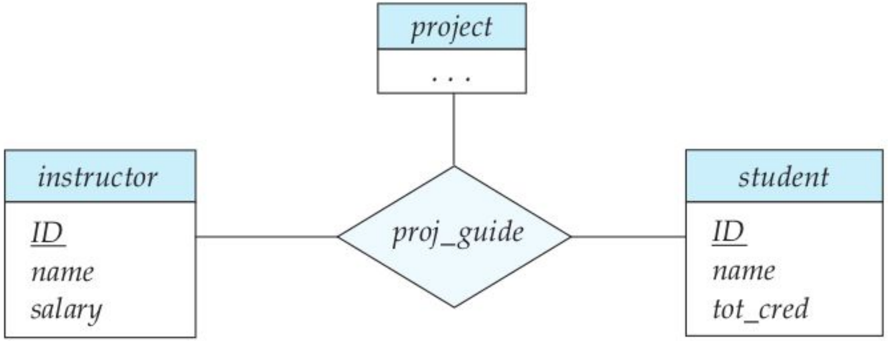
Roles

- Important for relationship of an entity set with itself
- Indicated as edge labels



Non binary relationship sets

- Suppose a student can have at most one instructor as a guide on a project:
    - This constraint can be specified by an arrow pointing to instructor on the edge from proj_guide.
- Can we have two entity sets with participating cardinality ‘One’ ?

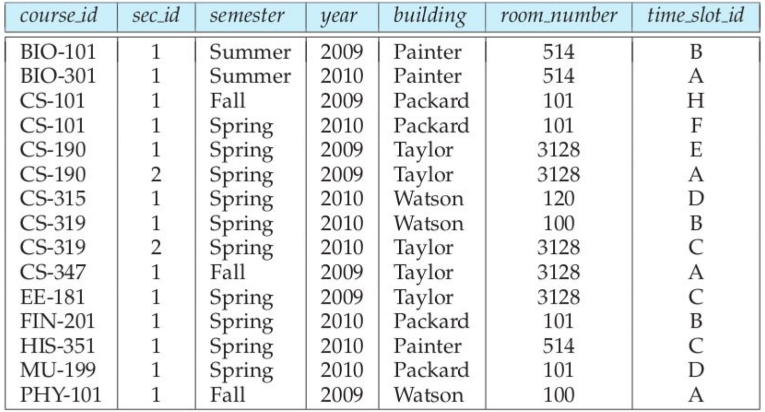


Non binary relationship sets

- Suppose R is a relationship over the entity sets E_1 E_2… E_i E_{i+1} …. E_m where E_1 E_2… E_i participate with cardinality 'One' then it can be interpreted in two ways:
  - Every combination of E_1 E_2… E_i can be associated with at most one combination of E_{i+1} …. E_m
    - Primary Key of R is the union of primary keys of E_1 E_2… E_i
  - For every E_k where 1<= k <= i : Each combination of the others entity sets can be associated with at most one entity from E_k
    - Each set $\{ E_1 , E_2 , . . . , E_{k-1} , E_{k+1} , . . . , E_n \}$, for $i < k \leq n$ forms a candidate key

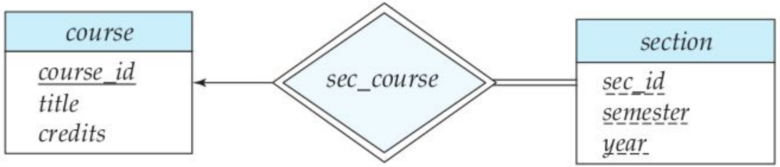
Weak Entity sets

- Weak Entity is an Entity set that does not have sufficient attributes to form a Primary Key
- Entity that has a primary key is called a strong entity
- Weak entity is always associated with a strong entity called identifying entity set (Owner entity set)
- The relationship that associates the weak entity set with its Owner entity set is called the Identifying relationship
- Identifying relationship is Many-to-One from Weak entity to Strong entity



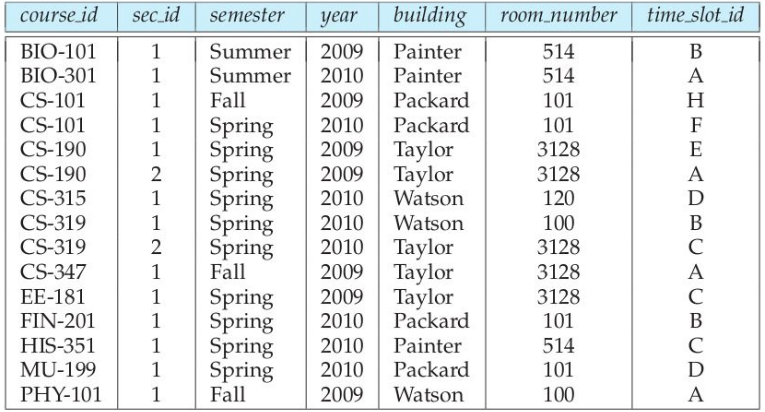
course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

section



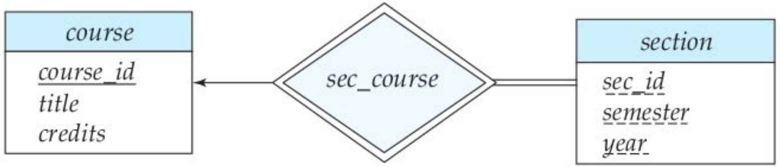
Weak Entity sets

- Discriminator of a weak entity is a set of attributes that allows distinguishing the weak entities that depend on the same owner entity
  - Also called Partial Key
- The discriminator of a weak entity is underlined with a dashed font.
- Identifying relationship is depicted by a double diamond.
- Weak entity set always 'totally participates' in the identifying relationship



course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

section

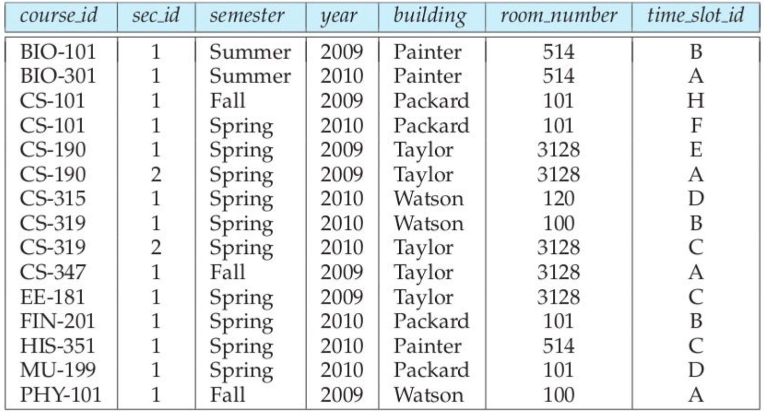


Weak Entity sets

- Weak entities can be alternatively described as composite attribute of the owner entity set

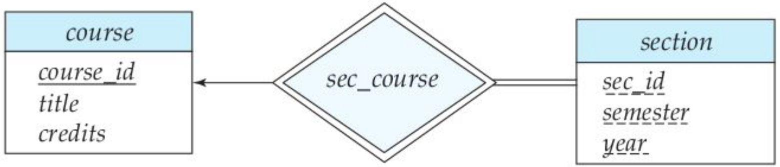
- Weak entity can also have multiple owner entity sets

- Weak entities can participate in other relationships with other entity sets

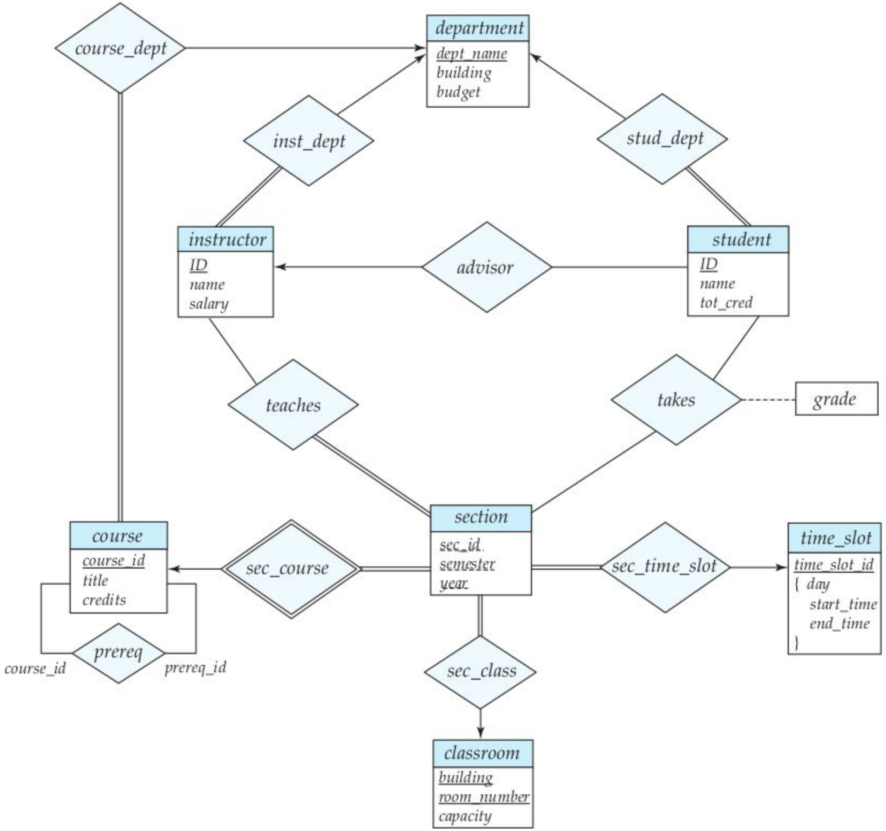


course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

section



E - R diagram for the University Database


course_dept
department
dept_name
building
budget
inst_dept
stud_dept
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
teaches
takes
grade
section
sec_id
semester
year
course
course_id
title
credits
sec_course
sec_time_slot
time_slot
time_slot_id
{ day
start_time
end_time
}
prereq
course_id
prereq_id
sec_class
classroom
building
room_number
capacity