DATABASE SYSTEMS

CS – 355/ CE – 373

Instructor: Maria N. Samad

September 9th, 2024

MINIMAL PRIMARY KEY OF RELATIONSIP SETS

- **Case 2**: ***Relationship set has descriptive attributes***
  - Primary keys of entity sets are included in the primary key of the relationship set exactly as in Case 1, based on mapping cardinalities
  - Descriptive attributes are part of the minimal primary keys of relationship set, ***ONLY IF*** they are needed to distinguish between relationships
  - Otherwise, even descriptive attributes can be skipped from the minimal primary keys of relationship sets

MINIMAL PRIMARY KEY OF RELATIONSIP SETS

- **Case 2**: ***Relationship set has descriptive attributes***
  - For example:
    - There are two entities: ***Customer*** (Primary Key: cust_ID) and ***Product*** (Primary Key: prod_ID)
    - The relationship is *purchase*, i.e. a customer purchases a product – this is a many-to-many relationship as multiple customers can purchase multiple products
    - There is a descriptive attribute, called date, which indicates the date of purchase
    - Superkeys of this relationship would then be: **{cust_ID, prod_ID, date}**
    - But is this also the minimal primary key for this relationship?
      - To do that, we must check if the purchase will become meaningful without the date as well
      - The same customer could purchase the same make and model of product multiple times, so in order to distinguish the results from database, we need this date
    - Thus, the minimal primary key is **{cust_ID, prod_ID, date}**

MINIMAL PRIMARY KEY OF RELATIONSIP SETS

- <u>**Case 2**</u>: ***Relationship set has descriptive attributes***
  - For example:
    - There are two entities: ***Student*** (Primary Key: <u>student_ID</u>) and ***Course*** (Primary Keys: <u>course_ID</u>, <u>semester</u>)
    - The relationship is <u>*takes*</u>, i.e. students take course in a specific semester – this is also a many-to-many relationship
    - Let's say this relationship also has descriptive attribute called <u>grade</u> which is the end grade of this course
    - Superkeys of this relationship would then be: **{<u>student_ID, course_ID, semester, grade</u>}**
    - But is this also the minimal primary key for this relationship?
      - To do that, we must check if the <u>*takes*</u> relationship will be meaningful without the <u>grade</u> as well
      - As the primary keys of entity sets are sufficient to distinguish results from the database for this relationship, then it'll be unnecessary to have <u>grade</u> as a part of primary key as well
    - Thus, the minimal primary key is **{<u>student_ID</u>, <u>course_ID</u>, <u>semester</u>}**

MINIMAL PRIMARY KEY – EXAMPLES

- Activity Sheet:
  - Attempt **Question 12**

MINIMAL PRIMARY KEY – EXAMPLES

- Activity Sheet **Question 12** Solution:
  - **ER Model Q12 Solution**

WEAK ENTITIES

- An entity type should always have a key attribute which uniquely identifies each entity in the entity set.
- However, sometimes, there exists some entity type for which key attribute can’t be defined, even if you combine them together
- These are called ***Weak Entities***.
- The entity sets that do not have sufficient attributes to form a primary key are known as ***weak entity sets*** and the entity sets which have a primary key are known as ***strong entity sets***.

WEAK ENTITIES

- An entity that depends on another entity needs to have a referencing attribute. For example, a ***section*** of a ***course***, so the ***section*** entity must have an attribute of *courseID* or *courseTitle*, that can specify the course of that section.
- We may or may not have unique *sectionID* as an attribute in the ***section*** relation, however, without the ***course*** reference, the ***section*** entity is **incomplete**
- This means ***section*** becomes a ***weak entity***, as it is dependent on another entity, i.e. ***course***, in this example

WEAK ENTITIES

- Therefore, we must check both the scenarios to deduce if a specific entity will be ***strong*** or ***weak***
- To deduce if an entity is ***weak*** or not, first check if that entity has a primary key or not. If not, then it is a ***weak entity***
- Then deduce, which entity does it depend on. That entity will become its ***owner*** or ***identifying entity set***

WEAK ENTITIES

- Examples:
  - Employees and their Dependents:
    - Employee (EmployeeID, EmployeeName, Position) ← **???**
      - **STRONG ENTITY**
    - Dependent (DependentName, Relationship) ← **???**
      - **WEAK ENTITY**
- In this example, the ***Dependent*** entity does not have appropriate attributes that can be used as a primary key
- Therefore, it is considered as a ***weak*** entity
- Then check for its ***identifying entity***, which in this case will be ***Employee***, because employees have dependents

WEAK ENTITIES

- Examples:
  - Event and its admission Ticket:
    - Event (EventID, EventName, EventLocation, EventDate) ← **???**
      - **STRONG ENTITY**
    - Ticket (TicketNumber, SeatNumber, Price) ← **???**
      - **WEAK ENTITY**
- In this example, you may assume *TicketNumber* is the primary key in the ***Ticket*** entity because for an event that can never be the same, but this means that after the first event, we can never use the same numbers for *TicketNumber*, as this will cause redundancy, which is not possible, so we cannot make it a primary key
- Therefore, it is also considered as a ***weak*** entity
- Then check for its ***owner entity***, which in this example will be ***Event***, because tickets are associated with specific events

WEAK ENTITIES

- Examples:
  - Project and its Task:
    - Project (ProjectID, Title, StartDate) ← **???**
      - **STRONG ENTITY**
    - Task (TaskID, TaskName, Deadline) ← **???**
      - **WEAK ENTITY**
- In this example, the ***Task*** entity may use *TaskID* as a primary key, and if multiple projects use similar tasks they can use this *TaskID* to refer to it uniquely, but why would you have a *Deadline* in it? This means *TaskID* is not defined globally, and so it may be repeated in the relation. Thus, it is not a primary key
- Therefore, it is considered as a ***weak*** entity
- Its associated ***identifying*** or ***owner entity*** will be ***Project***, because a task cannot exist without a project

WEAK ENTITIES

- In all the examples in previous slides, the specified ***weak entities*** may conveniently be converted to ***strong entities*** by adding <u>extra</u> attribute(s) in the same entity that can be used as primary key, and/or that can overcome the "incompleteness" of that particular entity; but if that is not possible, then we can only define them as weak entities

WEAK ENTITIES

- As the weak entities do not have any primary key and/or relevant attributes, they cannot be identified on their own, so they depend on some other entity (known as ***identifying*** or ***owner entity set***)
- Every weak entity must be associated with an identifying entity
- Hence, the weak entity set is said to be ***existence dependent*** on the ***owner entity set***
- Thus, weak entity set always has ***total participation*** but strong entity set may or may not have total participation

WEAK ENTITIES

- The relationship associating the weak entity set with the identifying entity set is called the ***identifying relationship***
- This is many-to-one relationship from the weak entity set to the identifying entity set
- The identifying relationship set should NOT have descriptive attributes, instead they will be part of the weak entity set

WEAK ENTITIES

- Weak entity is dependent on strong entity to ensure its existence
- Weak entity does not have any ***primary key***, however, there should still be a way to distinguish its records
- For that it contains a set of keys called ***discriminator key***, or the ***partial key*** of the entity set, that is close enough to define the weak entity
- The discriminator of a weak entity is underlined with a dashed line in the E-R diagram

WEAK ENTITIES

- For the given examples, discriminator/partial keys can be:
  - Employees and their Dependents:
    - Employee (EmployeeID, EmployeeName, Position) ← **STRONG ENTITY**
    - Dependent (DependentName, Relationship) ← **WEAK ENTITY**
  - Event and its admission Ticket:
    - Event (EventID, EventName, EventLocation, EventDate) ← **STRONG ENTITY**
    - Ticket (TicketNumber, SeatNumber, Price) ← **WEAK ENTITY**
  - Project and its Task
    - Project (ProjectID, Title, StartDate) ← **STRONG ENTITY**
    - Task (TaskID, TaskName, Deadline) ← **WEAK ENTITY**

WEAK ENTITIES

- The ***primary key*** of a weak entity is formed by the combination of:
  - The ***primary key*** of the identifying/owner entity set
  - The ***discriminator*** of the weak entity set
- Weak entity set can participate in relationships other than the identifying relationship, and may be an owner entity in another relationship

WEAK ENTITIES

- A weak entity set is depicted via a double rectangle with the discriminator being underlined with a dashed line.
- The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond.

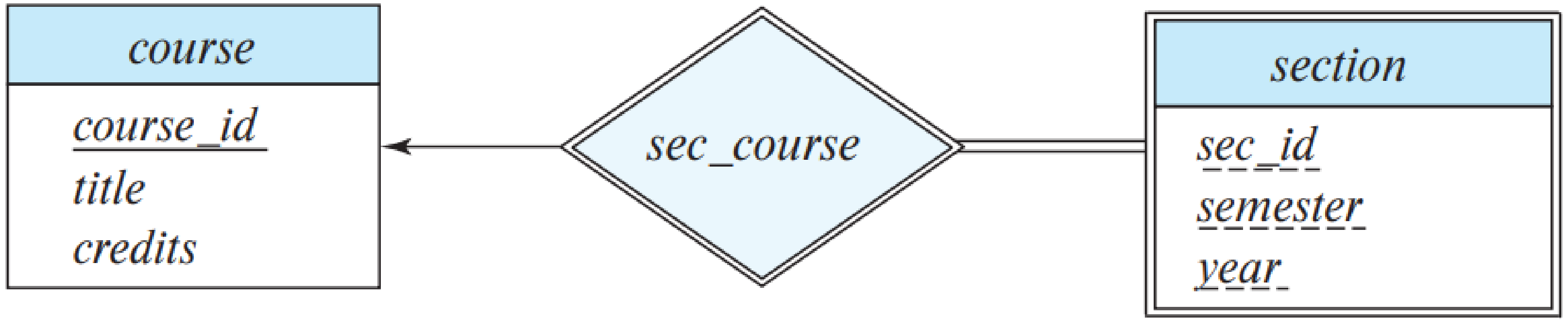

Figure 6.14 E-R diagram with a weak entity set.

WEAK ENTITIES

- In this example, ***Section*** entity does not have course_id as an attribute, which makes it insufficient to define a primary key, because multiple course can have same sec_id in a semester, for example L1 in Database Systems, and L1 in Object Oriented Programming. Hence it becomes the weak entity
- The relationship set, <u>*sec_course*</u> provides that extra information i.e. <u>course_id</u>, to identify the ***Section*** entities uniquely
- Here, the ***owner/identifier entity set*** for ***Section*** is ***Course***

WEAK ENTITIES

- The ***discriminator*** or ***partial key*** of the weak entity set, ***Section*** consists of the attributes: *sec_id*, *year*, and *semester*
- The ***primary key*** of ***Section*** will then be {*course_id, sec_id, year, semester*}
    - *course_id* is the primary key of identifying entity set, ***Course***
    - {*sec_id, year, semester*} is the discriminator of weak entity set, ***Section***
- ***Section*** being the weak entity set has total participation in the relationship
- It also has many-to-one relationship with the strong entity set, ***Course***

COMPLETE ERD

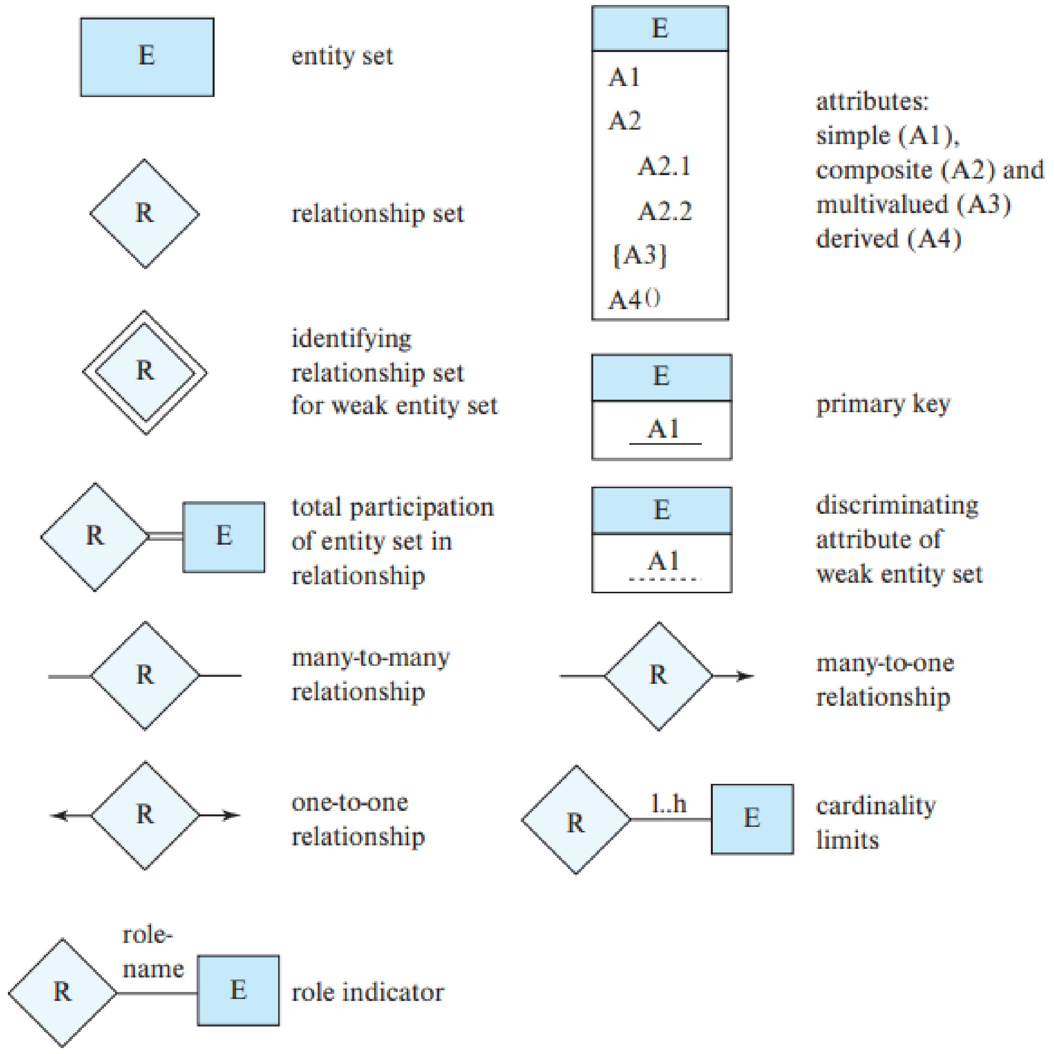
E
entity set
R
relationship set
R
identifying relationship set for weak entity set
R
E
total participation of entity set in relationship
R
many-to-many relationship
R
one-to-one relationship
R
role-name
E
role indicator
E
A1
A2
A2.1
A2.2
{A3}
A4()
attributes: simple (A1), composite (A2) and multivalued (A3) derived (A4)
E
A1
primary key
E
A1
discriminating attribute of weak entity set
R
many-to-one relationship
R
l..h
E
cardinality limits

Figure 6.26 Symbols used in the E-R notation.

COMPLETE ERD – EXAMPLE

- As per university policy, both the instructors and students can take a semester off during their time at university
- A department has the following attributes: dept_name, building and budget
- An instructor has the following attributes: ID, name, and salary
- A student has the following attributes: ID, name, and tot_cred
- A course has the following attributes: course_id, title and credits
- A department may or may not have an instructor, a student or a course associated with it
- Each instructor/student/course must have exactly one associated department
- Not all students need to be advised by an instructor, nor all instructors need to be advisors

COMPLETE ERD – EXAMPLE

- Different students takes different sections of courses
- A section has the following attributes: sec_id, semester, and year
- Multiple instructors teach multiple sections of courses
- Every section will always have an instructor
- Assume sec_course is an identifying relationship
- Multiple students can take multiple sections of courses
- Grade is generated after a student has completed the course each time
- Prereq is a recursion relationship with the course

COMPLETE ERD – EXAMPLE

- Every Section has a specific time slot when a particular class section meets
- A time slot must always be assigned to a section
- A time slot has the following attributes: time_slot_id and slot
- The slot attribute is a composite, as well as multivalued attribute, that can further be decomposed into day, start_time and end_time attributes

COMPLETE ERD – EXAMPLE

- Multiple sections can have classes in a particular classroom
- All sections will always take place in a classroom
- A classroom has the following attributes: building, room_number and capacity

Final ERD

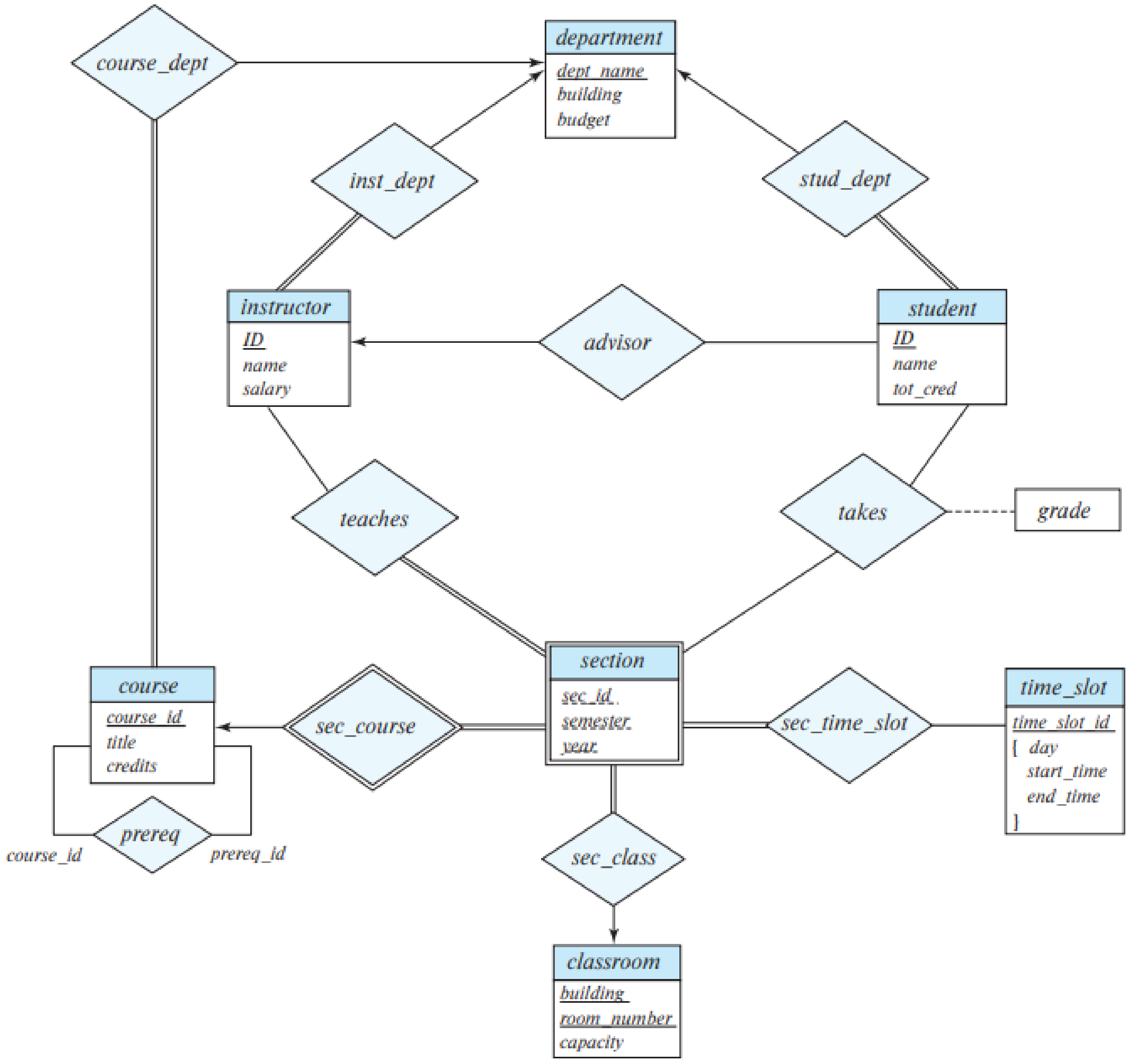

Figure 6.15 E-R diagram for a university enterprise.

ER DIAGRAMS

- Activity Sheet

ER DIAGRAMS

- Activity Sheet Solution:
  - ERD Solution