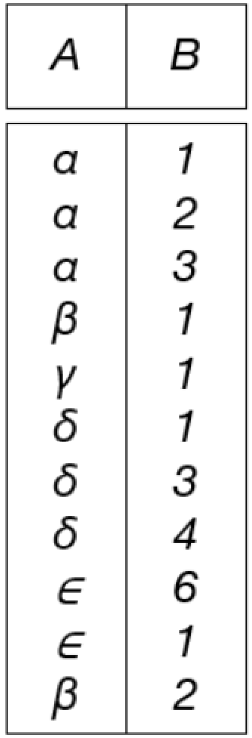
DBMS Week 4 TA Session

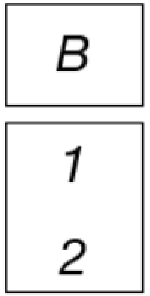
Division Operation

- Relations *r*, *s*:



A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ϵ	6
ϵ	1
β	2

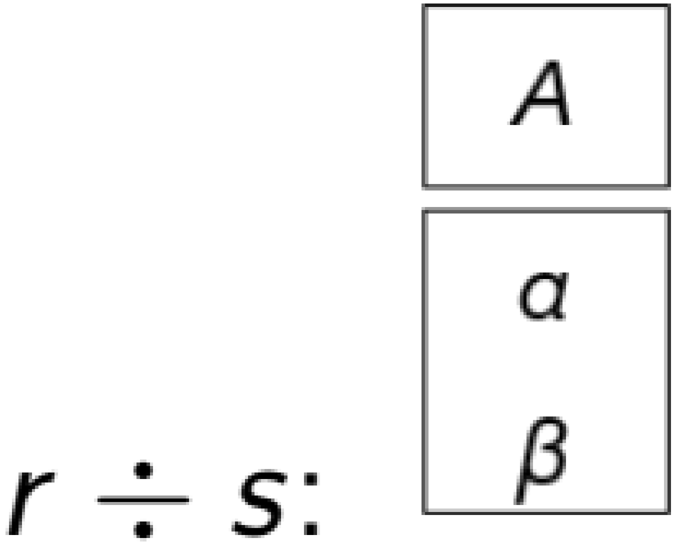
r



B
1
2

s

Division Operation (Continued)



$r \div s$:

A
α
β

Relational Algebra

It's a procedural query language

- σ - Select
- π - Project
- $\neg$ - Negation (not)
- $\cup$ - Union
- $\cap$ - Intersection
- $\times$ - Cartestion Product
- $-$ - Set Difference (Except)
- $\bowtie$ - Natural Join

Example

1. Find all the names of students whose age is greater than 25, or who are enrolled in Maths

$$\pi_{Name}(\sigma_{Age<25 \lor Subject='Maths'}(Students))$$

2. Find the name and sports of the student whose age is less than 25 and awards is greater than 3

$$\pi_{Name,Sports}(\sigma_{Age<25 \land Awards>3}(Students \bowtie Activity))$$

Tuple Relational Calculus

- TRC is a non-procedural query language, where each query is of the form

$$\{t \mid P(t)\}$$

where **t** = resulting tuples,

P(t) = known as predicate and these are the conditions that are used to fetch t.

Example

1. Find the name of the students whose age is 21

$$\{t \mid \exists\ s \in students(t.name = s.name \land s.age = 21)\}$$

2. Find the name of the employees who works in department manufacturing

employee(*id*, name, salary)
department(*id*, d_id, name, building)

$$\{M \mid \exists\ E \in employee\ \exists\ D \in department(E.id = D.id \land D.name =' Manufacturing' \land M.name = E.name)\}$$

Domain Relational Calculus

- A non-procedural query language equivalent in power to the tuple relational calculus
- Each query is an expression of the form:

$$\{< x_1, x_2, \ldots, x_n > \mid P(x_1, x_2, \ldots, x_n)\}$$

- $x_1, x_2, \ldots, x_n$ represent domain variables
- P represents a formula similar to that of the predicate calculus

Example

1. Find the name of the students whose age is 21

student(name, age, marks)

$$\{< a > \mid \exists\, b\, (< a, b, c > \in \mathit{students} \ \wedge \ b = 21)\}$$

2. Find the name of the employees who works in department manufacturing

employee(*id*, name, salary)
department(*id*, d_id, name, building)

$$\{< b > \mid \exists\, a, c, d\, (< a, b, c > \in \mathit{employee}) \ \wedge \ \exists\, y\, (< a, x, y, z > \in \mathit{department} \ \wedge y =' \mathit{Manufacturing}')\}$$

Entity Sets

- An **entity** is an object that exists and is distinguishable from object.
- An **entity set** is a set of entities of the same type that share the same properties

Strong Entity set

- A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities.
- A primary key exists for a strong entity sets

Weak Entity Set

- A weak entity set is an entity set that does not contain sufficient attributes to uniquely identify its entities.
- A primary does not exists for a weak entity set
- However, it contains a partial key called as a **discriminator**
- **Discriminator** represented by underlining with a dashed line.

Weak Entity set (continued)

- Weak entity set cannot exist independently since it doesn't have primary key
- It features in the model in relationship with a strong entity set. This is called the **identifying relationship**
- **Primary key of weak entity set = Discriminator + Primary key of Strong entity set**
- It must have **total participation** and **identifying relationship**

Attributes

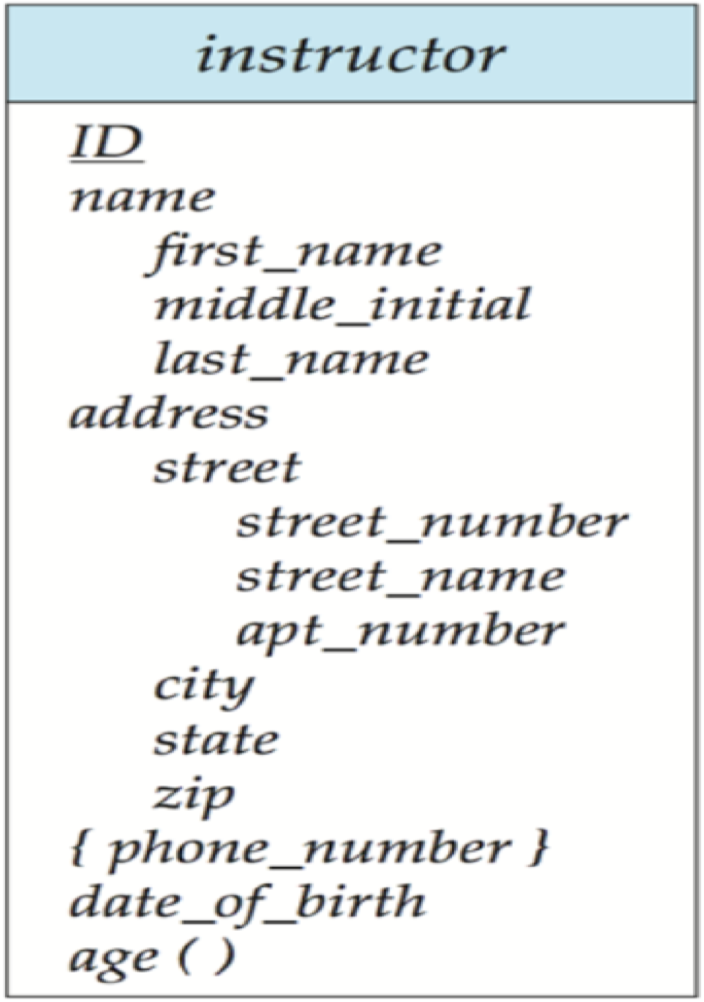
- An attribute is a property associated entity set.

Types of attributes

- Simple attribute
- Composite attribute - **Eg** - fname, mname, lname can consits in a name

- Multivalued attribute - **Eg** - **{phone_numbers}**
- Derived attribute - **Eg** - **age()** from date of birth

Example



instructor

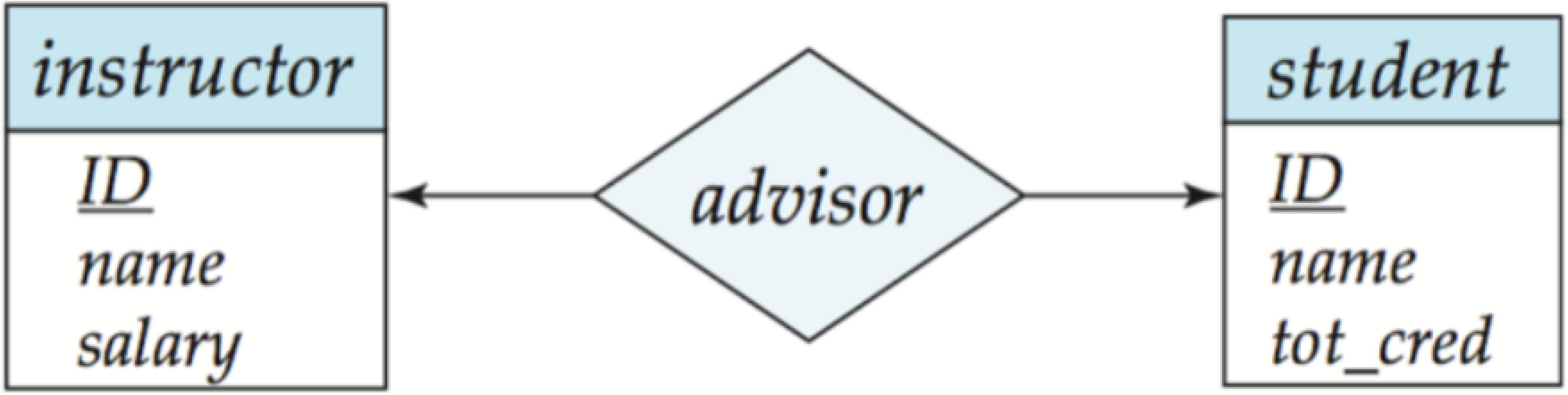
- *ID*
- *name*
  - *first_name*
  - *middle_initial*
  - *last_name*
- *address*
  - *street*
    - *street_number*
    - *street_name*
    - *apt_number*
  - *city*
  - *state*
  - *zip*
- *{ phone_number }*
- *date_of_birth*
- *age ( )*

ER Diagram

- Rectangles represent entities
- Attributes are listed inside the rectangle
- Underline indicates primary key
- Diamond represent relationship set
- Total Participation (indicated by double line)
- Weak entity set represented by double rectangle

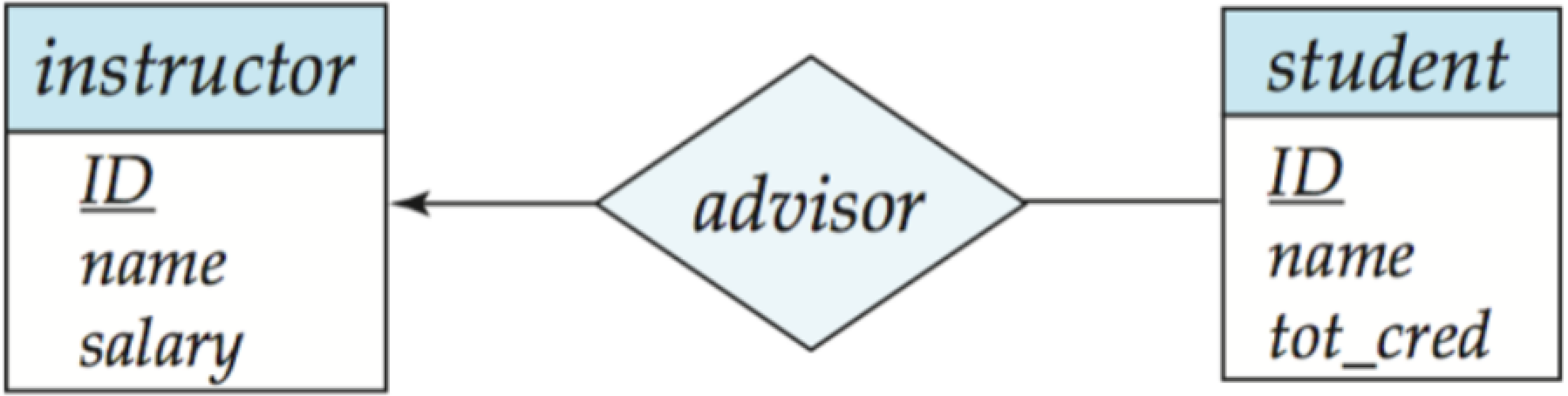
One to One relationship

- Each instructor has atmost one student
- Each student has atmost one instructor



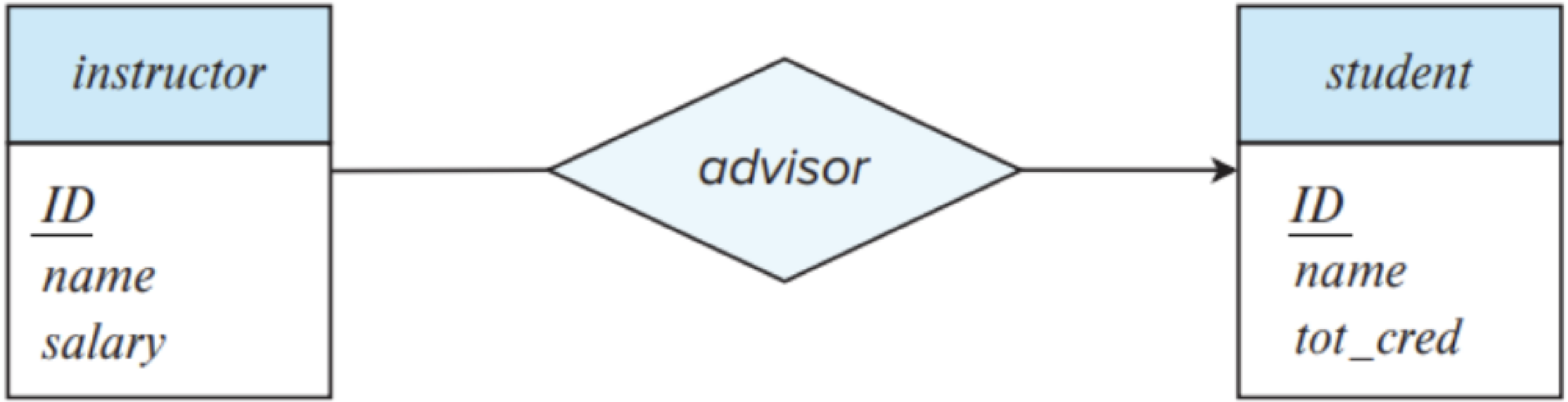
One to Many relationship

- Each instructor has one or many students
- Each student has at most one instructor



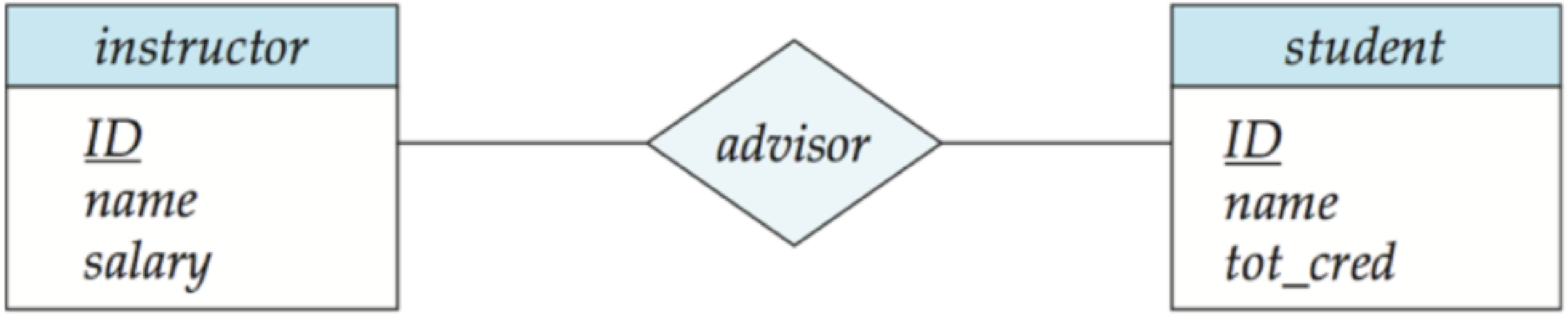
Many to one relationship

- Each instructor has at most one student
- Each student has one or many instructors



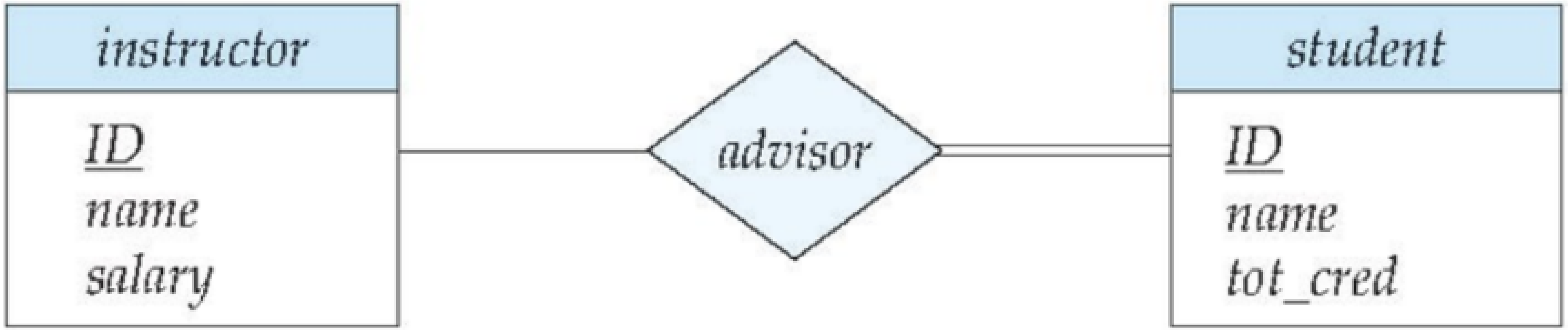
Many to Many relationship

- Each instructor has one or many students
- Each student has one or many instructors

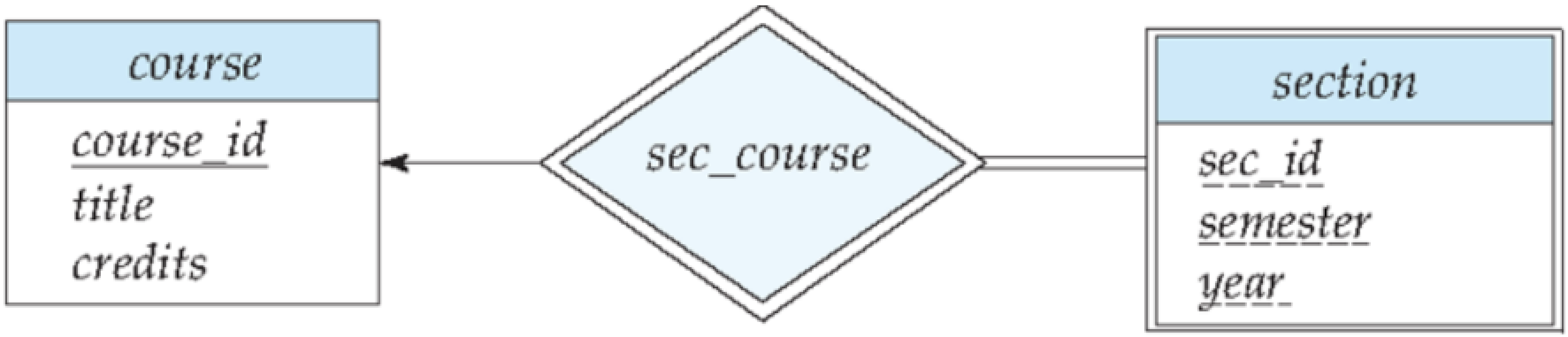


Total and Partial Participation

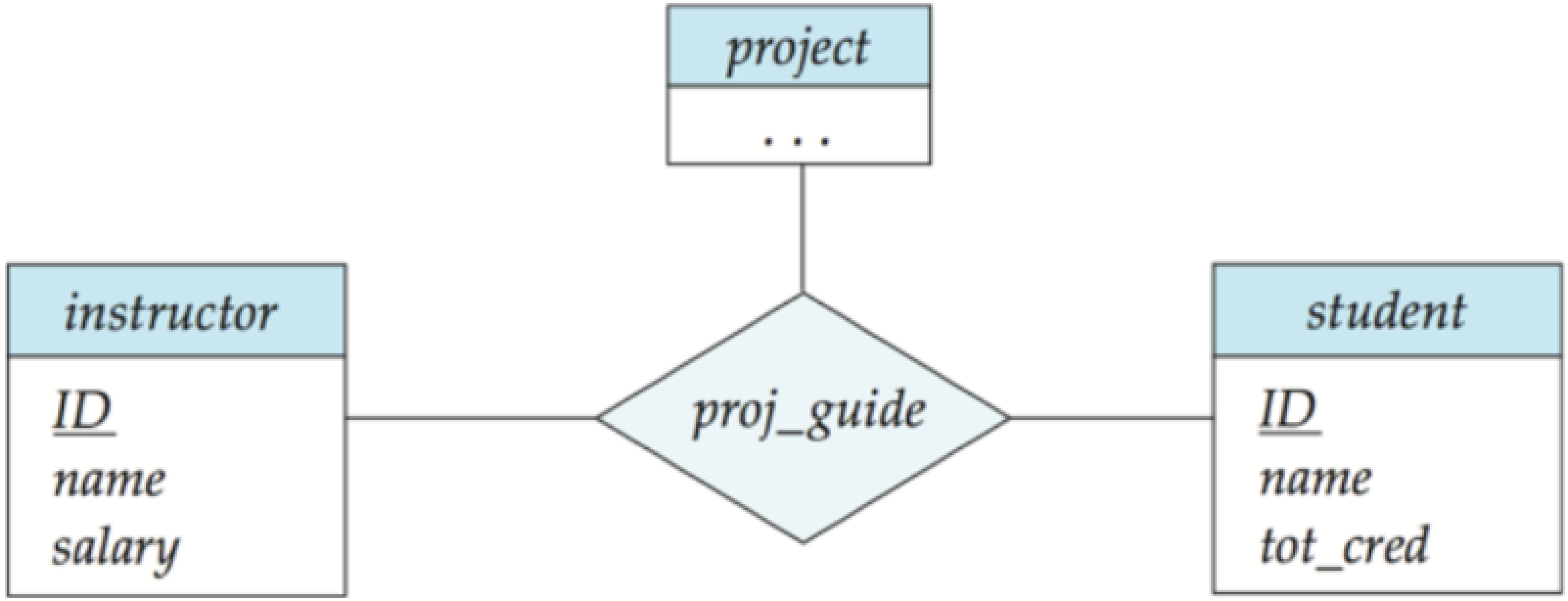
- Every student must have an instructor
- Instructor may or may not have a student



Expressing weak entity set



Ternary Relationship



Aggregation

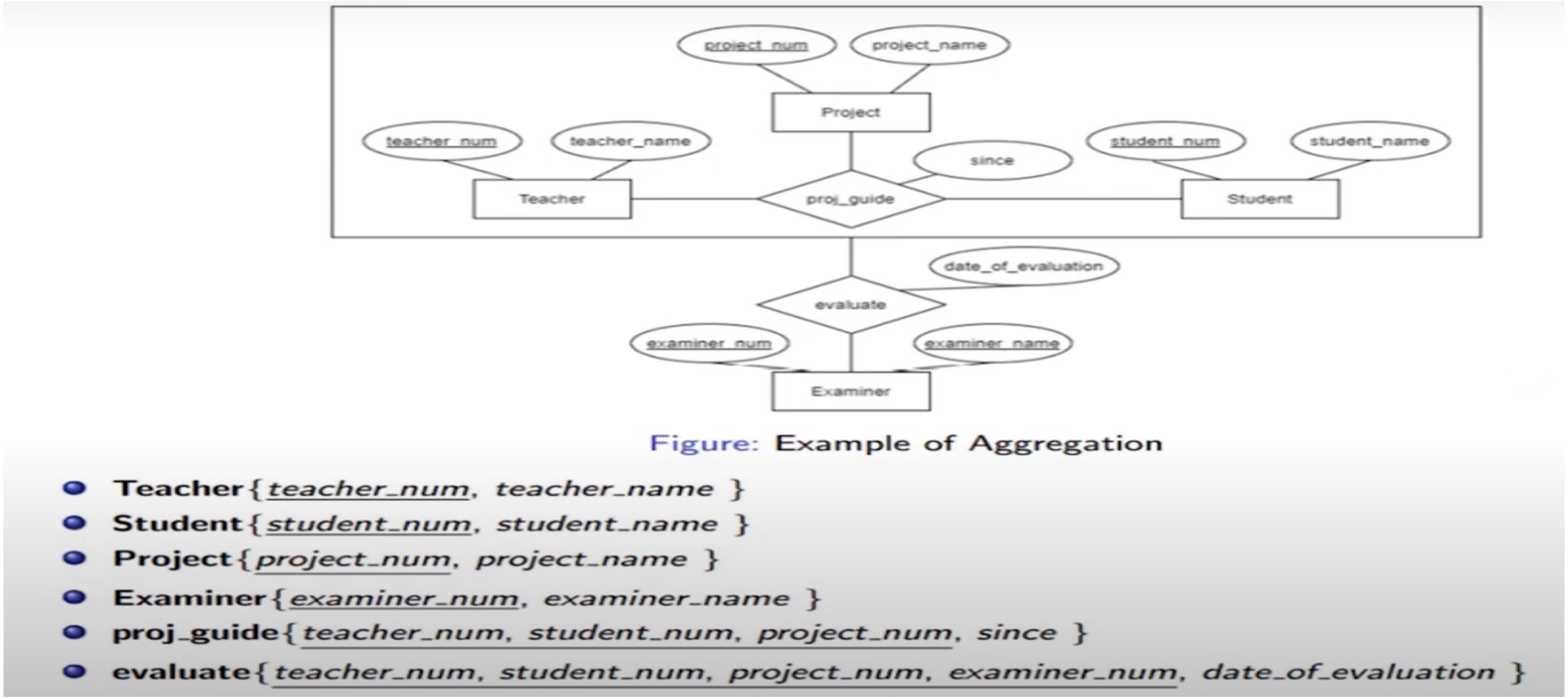


Figure: Example of Aggregation

- **Teacher**{*teacher_num*, *teacher_name* }
- **Student**{*student_num*, *student_name* }
- **Project**{*project_num*, *project_name* }
- **Examiner**{*examiner_num*, *examiner_name* }
- **proj_guide**{*teacher_num*, *student_num*, *project_num*, *since* }
- **evaluate**{*teacher_num*, *student_num*, *project_num*, *examiner_num*, *date_of_evaluation* }

Overlapping and Partial

- A person can be either student or faculty or both.
- There may be some persons who are just persons not belongs to faculty and students

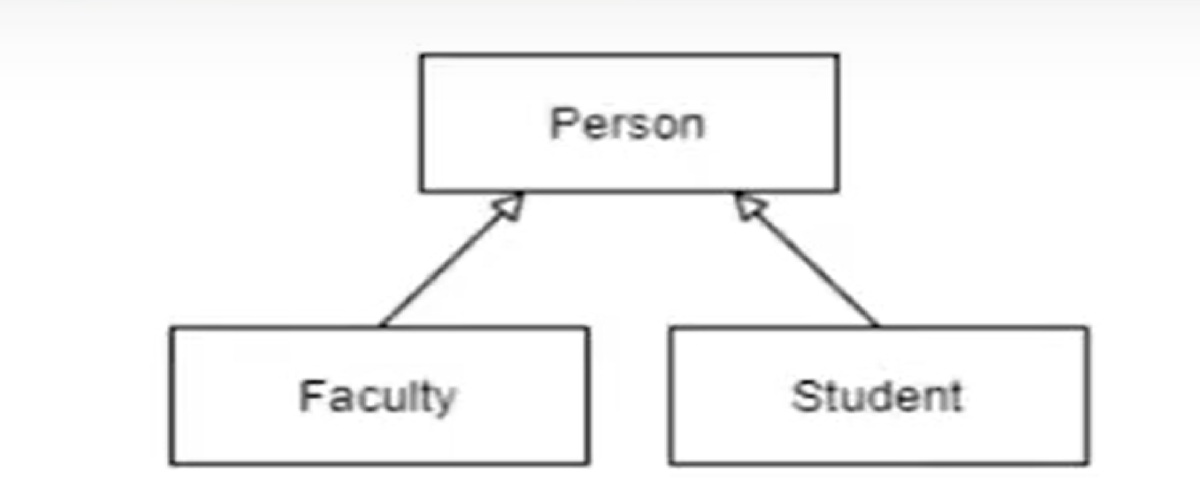


Figure: Overlapping and Partial

Disjoint and Partial

- A Student can be either UG students or PG students but they cannnot be both UG and PG students
- There may be students who are just students not belongs both UG and PG students

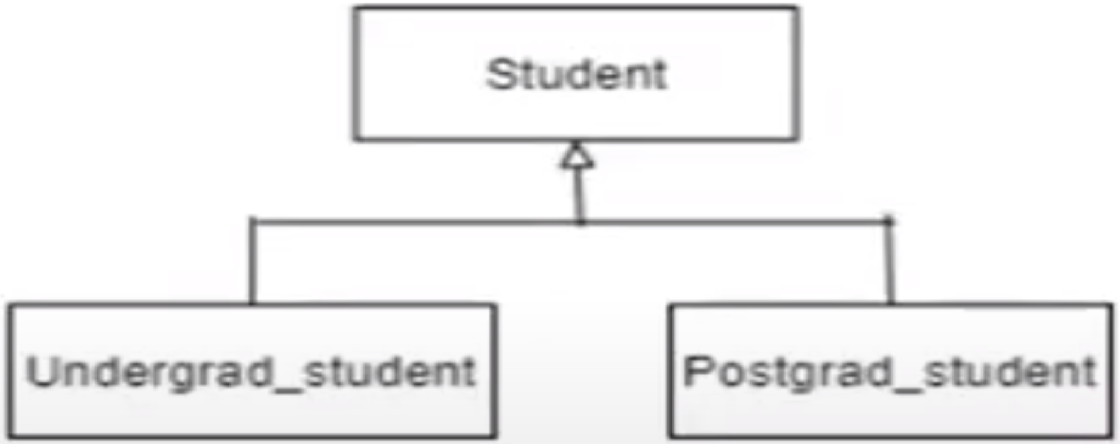


Figure: Disjoint and Partial

Disjoint and Complete

- Every Part must be present in either Purchased part or Manufactured part

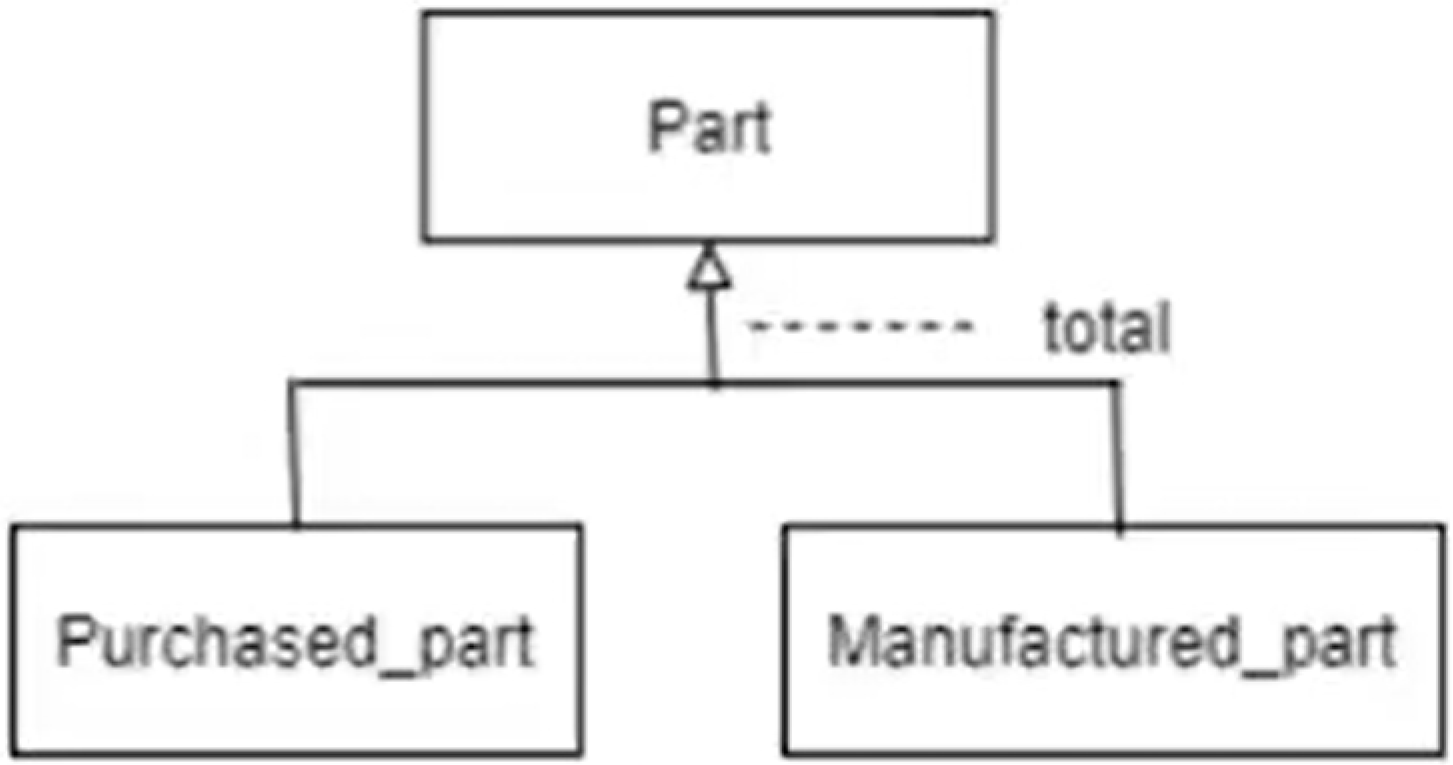


Figure: Disjoint and Complete

Strong Entity Set

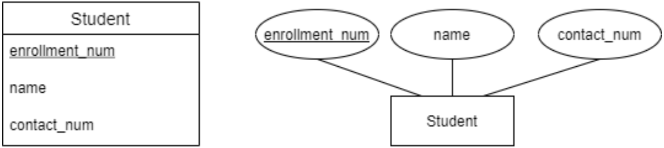
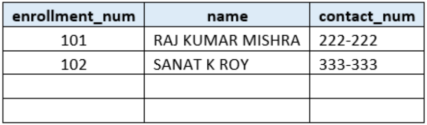


Figure: Strong entity with simple attributes

Student{*enrollment_num*, *name*, *contact_num*}



enrollment_num	name	contact_num
101	RAJ KUMAR MISHRA	222-222
102	SANAT K ROY	333-333

Figure: Table **Student**

Strong Entity Set with Composite Key

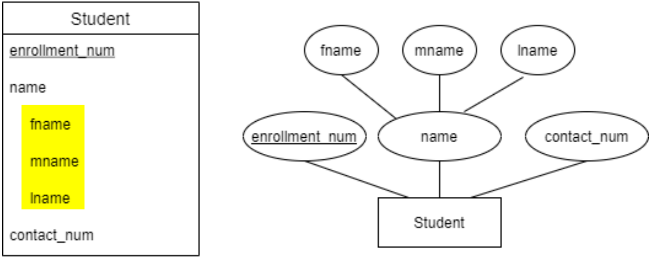


Figure: Entity set **Student** with simple and composite attributes

$\textbf{Student}\{\underline{\textit{enrollment\_num}}, \textit{fname, mname, lname}, \textit{contact\_num}\}$

enrollment_num	fname	mname	lname	contact_num
101	RAJ	KUMAR	MISHRA	222-222
102	SANAT	K	ROY	333-333

Figure: Table **Student**

Strong Entity Set with Composite Key + Multivalued Attribute

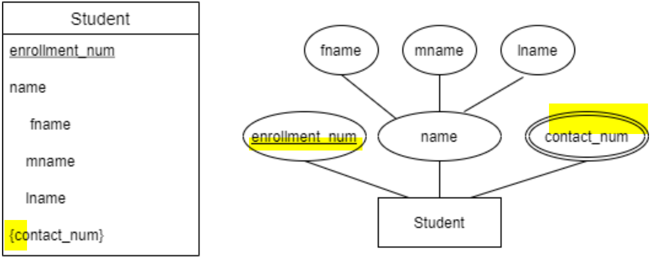
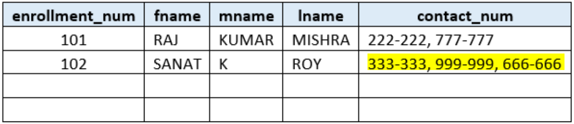


Figure: Entity set **Student** with simple, composite, and multivalued attributes

Student{*<u>enrollment_num</u>*, *fname*, *mname*, *lname*, *contact_num*}



enrollment_num	fname	mname	lname	contact_num
101	RAJ	KUMAR	MISHRA	222-222, 777-777
102	SANAT	K	ROY	333-333, 999-999, 666-666

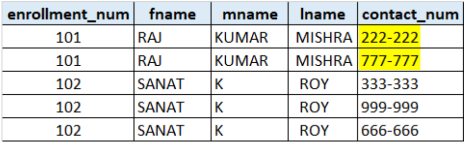
Figure: Table **Student**

causes problems when we our query has WHERE

SOLN: decompose the table into multiple tables

Strong Entity Set
with Composite Key + Multivalued Attribute

Student{*enrollment_num*, *fname*, *mname*, *lname*, *contact_num*}



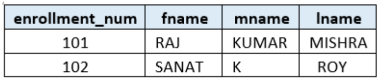
enrollment_num	fname	mname	lname	contact_num
101	RAJ	KUMAR	MISHRA	222-222
101	RAJ	KUMAR	MISHRA	777-777
102	SANAT	K	ROY	333-333
102	SANAT	K	ROY	999-999
102	SANAT	K	ROY	666-666

Figure: Table **Student**

OR

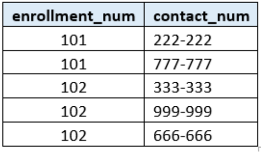
Contacts{*enrollment_num*, *contact_num*}

Student{*enrollment_num*, *fname*, *mname*, *lname* }



enrollment_num	fname	mname	lname
101	RAJ	KUMAR	MISHRA
102	SANAT	K	ROY

Figure: Table **Student**



enrollment_num	contact_num
101	222-222
101	777-777
102	333-333
102	999-999
102	666-666

Figure: Table **Student**

Strong Entity Set with Composite Key + Multivalued Attribute + Derived Attribute

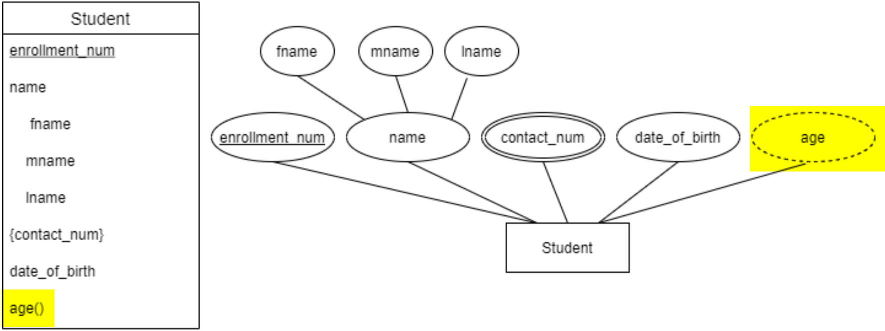


Figure: Entity set **Student** with simple, composite, multivalued attributes, and derived attribute

Student{*enrollment_num*, *fname*, *mname*, *lname*, *data_of_birth* }

Contacts{*enrollment_num*, *contact_num*}

Relationship: Cardinality Constraint (many-to-many)

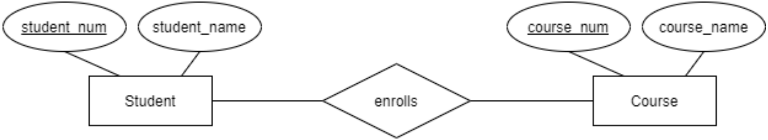
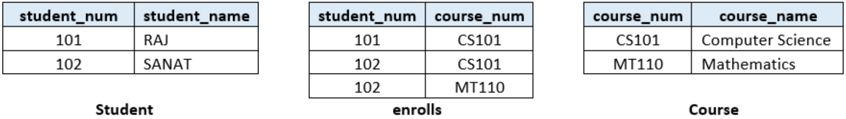


Figure: A many-to-many relationship between **Student** and **Course**

- **Student**{*student_num*, *student_name*}
- **Course**{*course_num*, *course_name*}
- **enrolls**{ *student_num*, *course_num* }



student_num	student_name
101	RAJ
102	SANAT

Student

student_num	course_num
101	CS101
102	CS101
102	MT110

enrolls

course_num	course_name
CS101	Computer Science
MT110	Mathematics

Course

Figure: Table: **Student**, **Course** and **enrolls**

Relationship: Cardinality Constraint (many-to-many) with Descriptive Attributes

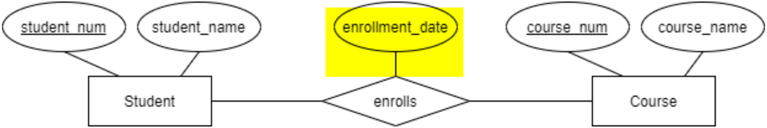


Figure: A many-to-many relationship between **Student** and **Course**

- **Student**{<u>*student_num*</u>, *student_name*}
- **Course**{<u>*course_num*</u>, *course_name*}
- **enrolls**{ <u>*student_num*</u>, <u>*course_num*</u>, *enrollment_date* }

Relationship: Cardinality Constraint (many-to-one)

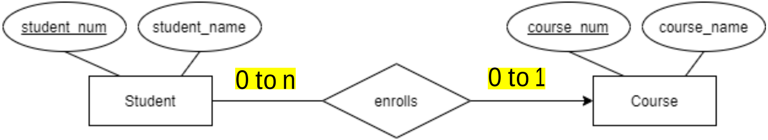


Figure: An many-to-one relationship between **Student** and **Course**

primary of one becomes attribute of many

- **Student**{*student_num*, *student_name*, *course_num*}
- **Course**{*course_num*, *course_name*}

must be nullable

Relationship: Cardinality Constraint (one-to-one)

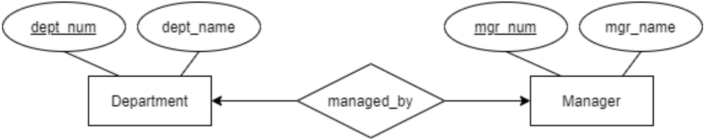


Figure: An one-to-one relationship between **Department** and **Manager**

- **Department**$\{\underline{dept\_num},\ dept\_name\}$
- **Manager**$\{\underline{mgr\_num},\ mgr\_name,\ dept\_num\}$

OR

- **Department**$\{\underline{dept\_num},\ dept\_name,\ mgr\_num\}$
- **Manager**$\{\underline{mgr\_num},\ mgr\_name\ \}$

Relationship: Cardinality Constraint (many-to-one) with Participation Constraint

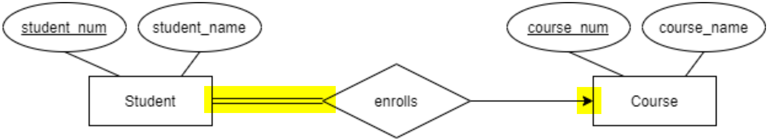


Figure: An many-to-one relationship between **Student** and **Course**

not null

- **Student**{<u>*student_num*</u>, *student_name*, *course_num*}
- **Course**{<u>*course_num*</u>, *course_name*}

Relationship: Cardinality Constraint (one-to-one) with Participation Constraint

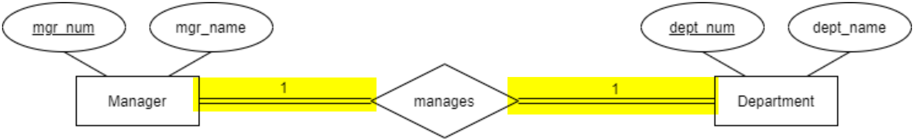


Figure: A one-to-one relationship between **Manager** and **Department**

- **Mgr_Dept**{*mgr_num, dept_num*, *mgr_name*, *dept_name* }

Weak Entity Set

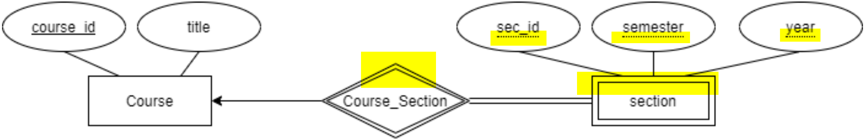


Figure: A relationship between **Course** and **Section**

- **Course**{*course_id*, *title* }
- **Section**{*course_id, sec_id, semester, year* }

Ternary Relationship

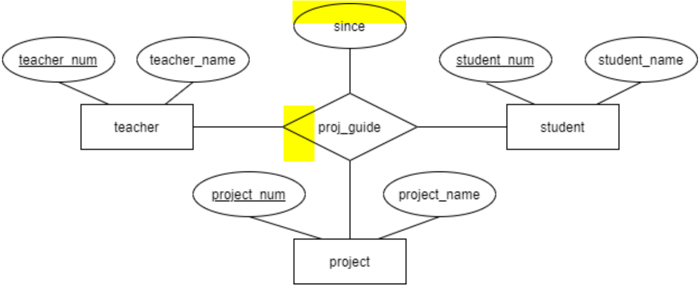


Figure: Example of ternary relationship

- **teacher**{*teacher_num*, *teacher_name* }
- **student**{*student_num*, *student_name* }
- **project**{*project_num*, *project_name* }
- **proj_guide**{*teacher_num*, *student_num*, *project_num*, *since* }

E-R Diagram with Aggregation

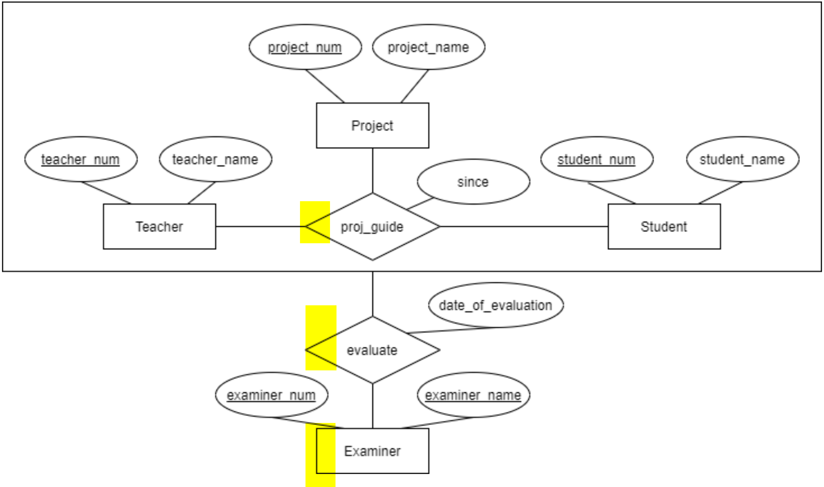
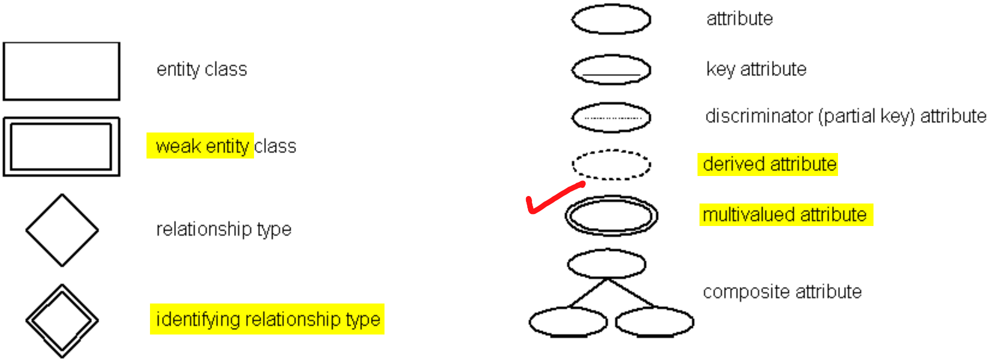


Figure: Example of Aggregation

- **proj_guide**{*teacher_num*, *student_num*, *project_num*, *since* }
- **Examiner**{*examiner_num*, *examiner_name* }
- **evaluate**{*teacher_num*, *student_num*, *project_num*, *examiner_num*, *date_of_evaluation* }

E-R diagrams symbols



entity class

weak entity class

relationship type

identifying relationship type

attribute

key attribute

discriminator (partial key) attribute

derived attribute

multivalued attribute

composite attribute