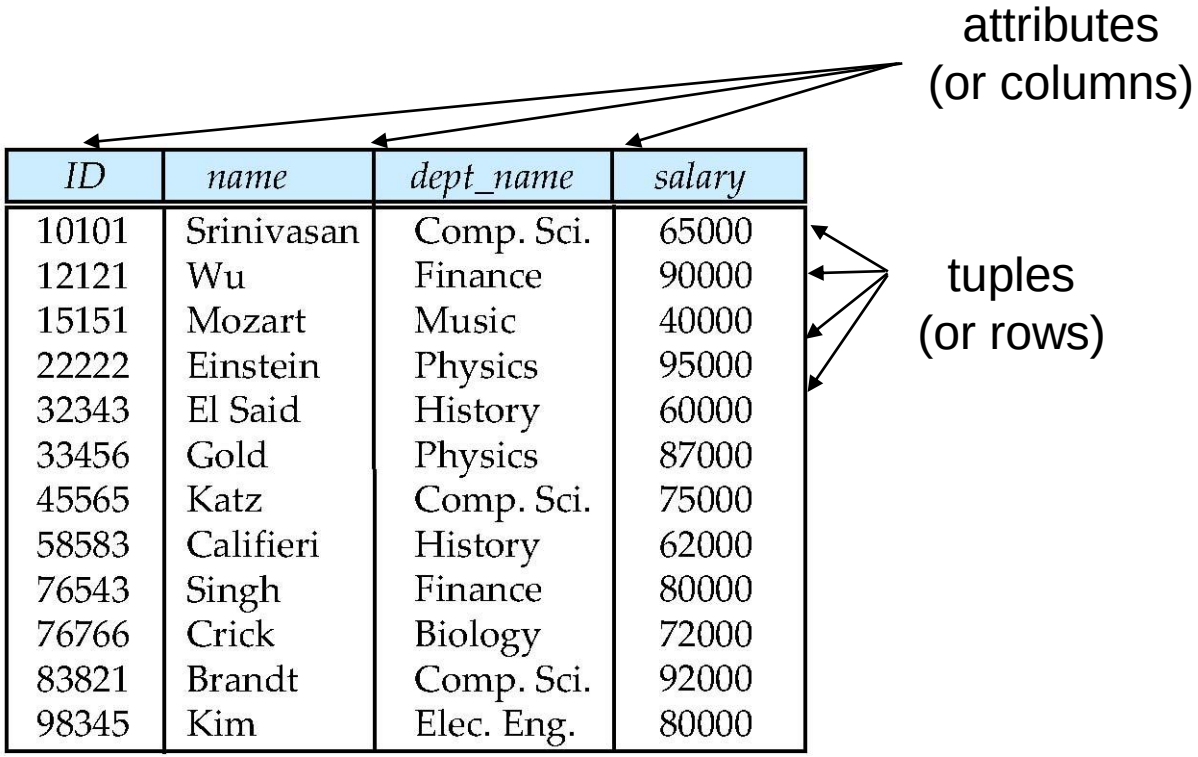
Database and Information Systems

Course Roadmap

Chapter 1	Introduction to Databases
Chapter 2	Integrity Constraints and ER Model
Chapter 3	Relational Databases and Schema Refinement
Chapter 4	Query Language
Chapter 5	Transaction and Concurrency Control
Chapter 6	Indexing

Example of Relation



ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Table: Faculty

Attribute Types

- Columns are attributes
- Rows are tuples or records
- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible

Candidate Key

- Key
  - Attribute of a relation/table
- Candidate Key
  - Attribute that **uniquely identify any two tuples** in a table
  - E.g., In Faculty table, Candidate Key (CK): {ID}
    - To identify if two faculties are same or different
  - In Student table, CK: {roll no, aadhaar no, registration no, email id}
    - To identify if two students are same or different
  - In a Table, there can be more than one CKs

Candidate Key

- Question?
  - Can we use DOB as CK?

Candidate Key

- Question
  - Can we use name as CK?

Primary Key

- Primary Key
  - One of the candidate keys that is not Null: **Unique + Not Null**
  - A Table can have **only one Primary Key**
- Examples
  - In Student table, Primary Key can be either Roll No. or Registration No.
  - In Aadhaar Card Database, Primary Key can be Aadhaar No.

Primary Key

- Question?
  - In Student Table, Aadhaar card can be one of the possible Primary Keys?

Primary and Alternate Keys

- SQL query to create Primary Key:
  - CREATE TABLE Student (
    RollNo int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (RollNo));
  - Primary Key is usually assigned by the organization
- Alternate Key
  - Set of **candidate keys that are not primary key**

Super Key

- Combination of all possible attributes which can uniquely identify two tuples in a table
- **Super set of a candidate is a super key**
  - Minimal super key is a candidate key
- For Example
  - Consider a relation/table **student(id, name)**, where id is a candidate key
    - Possible super keys = {id, (id name)}

Class Activity

- How many possible super keys
  - A set of n attributes with one attribute as candidate key

2^{n-1}

Class Activity

- How many possible super keys
  - A set of n attributes with two combined attribute as candidate key (composite key as candidate key)

2^{n-2}

Class Activity

- How many possible super keys
  - A set of n attributes with two attributes as candidate keys

$$2^{n-1} + 2^{n-1} - 2^{n-2}$$

$$= 3 \cdot 2^{n-2}$$

Class Activity

- Minimum and maximum super keys in a relation of n attributes?

1 $2^n - 1$

Super Key

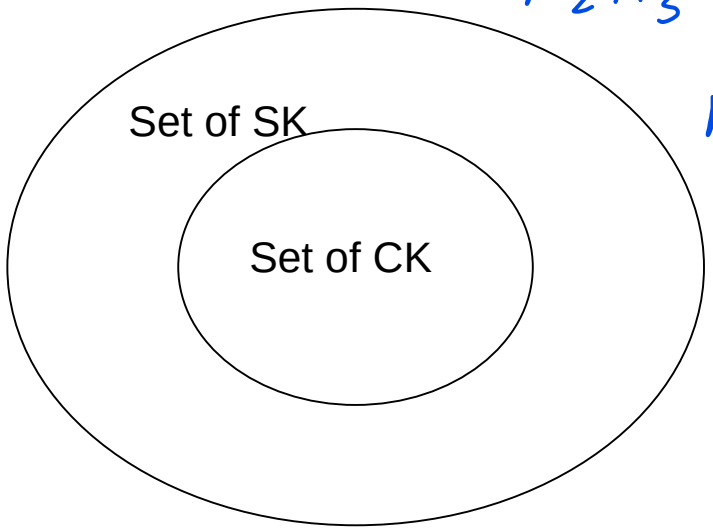
- Example: R(A_1, A_2, A_3, A_4), Candidate keys (CK) = {A_1, A_2A_3}
  - No of super keys (SK) = ?

$A_1 \rightarrow 2^3$

$A_2A_3 \rightarrow 2^2$

$A_1 + A_2A_3 \rightarrow 2$

$2^3 + 2^2 - 2$



Class Activity

- Consider a relation R(A_1, A_2, A_3, ….., A_n), where candidate keys are {A_1A_2, A_2A_3}
  - How many possible super keys?

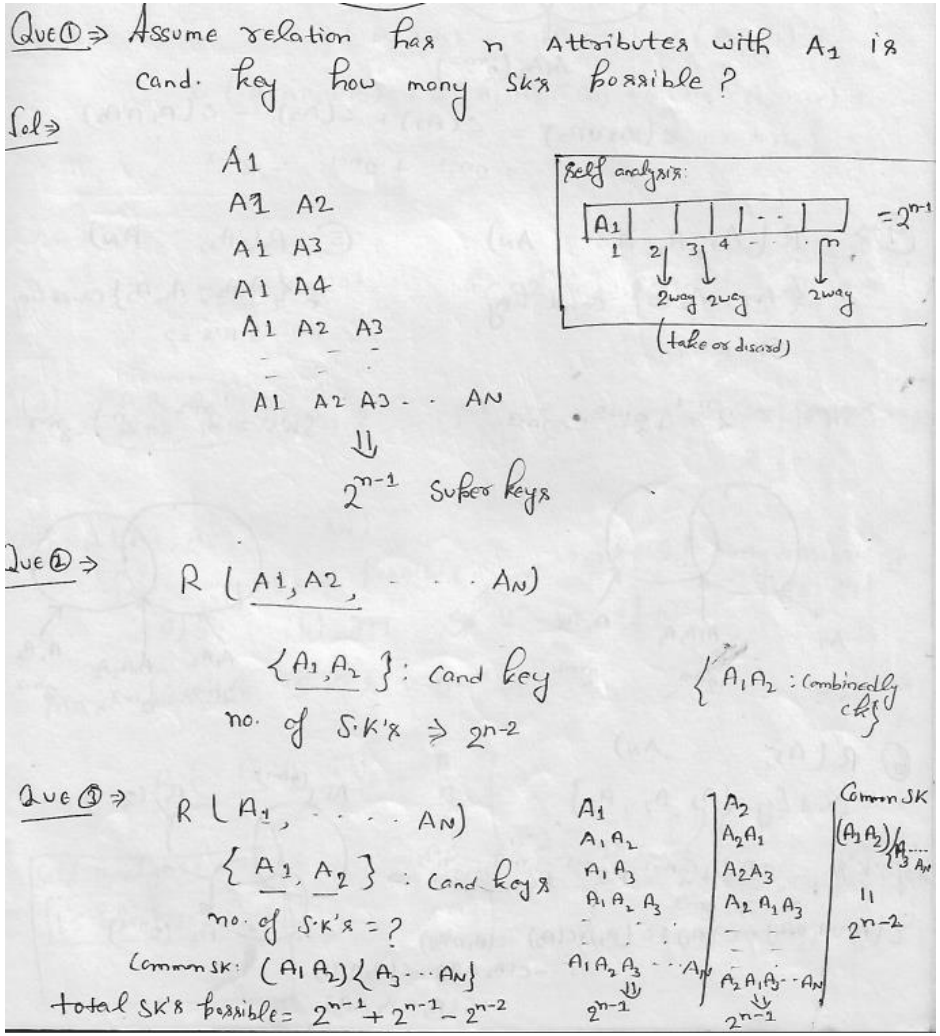
$A_1A_2 \rightarrow 2^{n-2}$

$A_2A_3 \rightarrow 2^{n-2}$

$A_1A_2A_3 \rightarrow 2^{n-3}$

$2^{n-2} + 2^{n-2} - 2^{n-3}$

Super Key



Question: Can you tell if A_2A_3 is a super key?

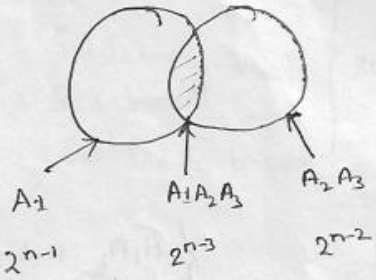
Super Key

④ → $R(A_1, A_2, A_3 \cdots A_N)$

$\{A_1, A_2A_3\}$ cand key

SK's = ?

$SK's = 2^{n-1} + 2^{n-2} - 2^{n-3}$

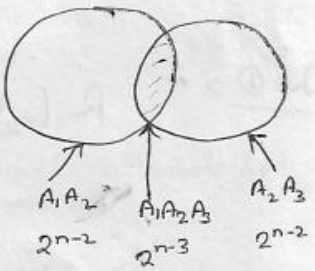


⑤ $R(A_1, \cdots A_N)$

$\{A_1A_2, A_2A_3\}$ cand key

S·K's = ?

$SK's = 2^{n-2} + 2^{n-2} - 2^{n-3}$

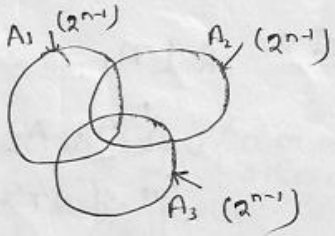


⑥ $R(A_1, \ldots, A_N)$

Cand key $\{A_1, A_2, A_3\}$

$\#\text{of S·K's} = 2^{n-1} + 2^{n-1} + 2^{n-1} + 2^{n-3} - 3 \cdot 2^{n-2}$

$= 7 \cdot 2^{n-3}$

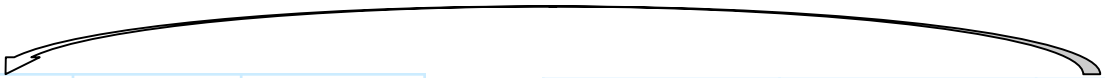


$C(A_1 \cup A_2 \cup A_3) = C(A_1) + C(A_2) + C(A_3) - C(A_1 \cap A_2) - C(A_2 \cap A_3) - C(A_3 \cap A_1) + C(A_1 \cap A_2 \cap A_3)$

Foreign Key

- Foreign Key
  - Attribute or set of attributes that references to primary key of same table or another table
  - Used to relate one table with other table
  - Maintains referential integrity
  - In a Table, there can be more than one foreign key
- Integrity
  - Maintain same value in database
- Foreign key integrity
  - Maintains referential integrity while inserting deleting and updating in the database
  - Example: Roll no 1 belongs to a CSE student, Ram

Foreign Key: Example



Roll_no (pk)	Name	Address
1	Ram	Delhi
2	Subham	Indore
3	Shyam	Delhi
4	Rohan	Mumbai
5	Bharat	Bangalore

Base Table/Referenced Table: **Student**

Course_id	Course_name	Roll_no (fk)
C1	Database	1
C2	Data Mining	2
C3	CN	1
C4	C	3
C5	OS	5

Referencing Table: **Course**

- To relate two tables, one attribute should be common in two tables
  - In the above tables, Roll_no is common and that is also acting as a Foreign Key
  - What is the meaning of Roll_no in Course table is taking reference from Roll_no in Student table?

Foreign Key

- How to create Foreign Key
  - CREATE TABLE Course (
    Course_id Varchar(10),
    Course_name Varchar(20),
    Roll_no int,
    PRIMARY KEY (Course_id),
    FOREIGN KEY (Roll_no) REFERENCES Student(Roll_no)
    );

Foreign Key: Class Activity

- Insertion in referenced table:
  - Will not cause problem
  - May cause problem
  - Will cause problem

Foreign Key: Class Activity

- Deletion in referenced table:
  - Will not cause problem
  - May cause problem
  - Will cause problem

Foreign Key: Class Activity

- Updation in referenced table:
  - Will not cause problem
  - May cause problem
  - Will cause problem

Foreign Key: Class Activity

- Insertion in referencing table:
  - Will not cause problem
  - May cause problem
  - Will cause problem

Foreign Key: Class Activity

- Deletion in referencing table:
  - Will not cause problem
  - May cause problem
  - Will cause problem

Foreign Key: Class Activity

- Updation in referencing table:
  - Will not cause problem
  - May cause problem
  - Will cause problem

Referential Integrity

- Insertion in Referenced Table or Base Table
  - No violation
- Deletion or updation in Referenced Table or Base Table
  - May create problem
- Insertion and updation in Referencing Table
  - May create problem
- Deletion in Referencing Table
  - No violation

Roll_no (pk)	Name	Address
1	Ram	Delhi
2	Subham	Indore
3	Shyam	Delhi
4	Rohan	Mumbai
5	Bharat	Bangalore

Referenced Table: Student

Course_id	Course_name	Roll_no (fk)
C1	Database	1
C2	Data Mining	2
c3	CN	3
c4	C	3
c5	OS	5

Referencing Table: Course

Maintain Referential Integrity

- Deletion in Referenced Table
  - On Delete Cascade
    - If referential integrity violation occurs, delete corresponding record from both the tables
  - On Delete No Action
    - Referenced attribute value deletion is restricted, if this attribute is referred by foreign key of referencing table
      - If it is essential to delete, delete in referencing table then in referenced table
  - On Delete Set Null
    - If referential integrity violation occurs, place NULL in the corresponding foreign key attribute

Class Activity

- Can you always apply On Delete Set Null?

Maintain Referential Integrity

- Updation in Referenced Table
  - On Update Cascade
  - On Update No Action
  - On Update set Null
- Insertion and Updation in Referencing Table
  - If any violation, restrict operations

Foreign Key

- How I will remember!!
  - GUI
  - DDU

ER Model

- **ER Model**
  - Logical representation of database
  - Used for high level database design
  - Has an associated diagrammatic representation, the **ER diagram**, which can express the overall logical structure of a database graphically
    - Creates a blueprint of database
- Main components in ER Model
  - Attribute, Entity, Relationship

ER Model

- Entity
  - An object in real world
    - Example: Student, course, faculty
- Attribute
  - Characteristics of an entity
    - Example: {roll no, age, address} can be attribute of Student entity
- Relationship
  - Connection or association between entities

Entity Sets

- An **entity** is an object that exists and is distinguishable from other objects
  - Example: specific person, company, event, plant
- An **entity set** is a set of entities of the same type that share the same properties
  - Example: set of all persons, companies, trees, holidays
- An entity is represented by a set of attributes; i.e., descriptive properties possessed by all members of an entity set
  - Example:

    instructor = (*ID, name, salary*)
    course= (*course_id, title, credits*)
- A subset of the attributes form a **primary key** of the entity set

Entity Sets -- *instructor* and *student*

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

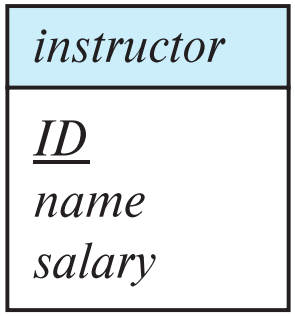
instructor

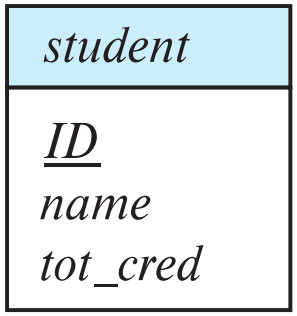
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

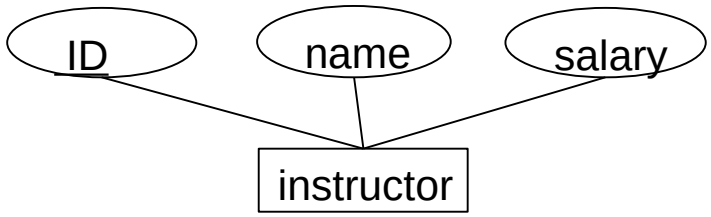
Representing Entity sets in ER Diagram

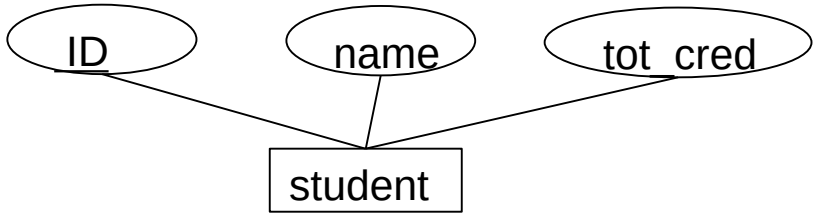
- Rectangles represent entity sets
- Attributes listed inside entity rectangle
  - Or eclipse represent attribute
  - Underline indicates primary key attributes





- Alternate representation





Relationship Sets

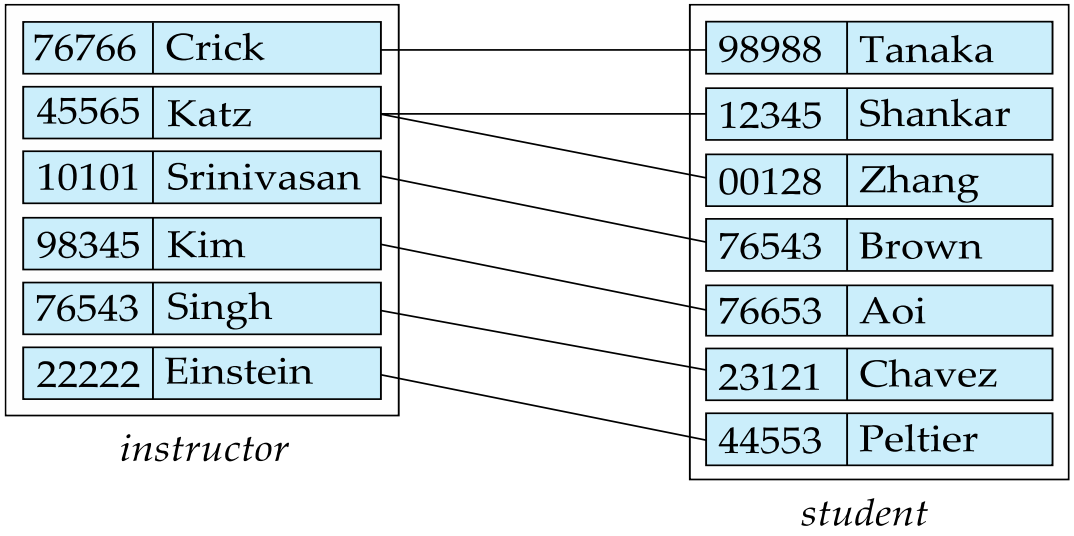
- A **relationship** is an association among several entities

  Example:

  44553 (Peltier) — *advisor* — 22222 (Einstein)

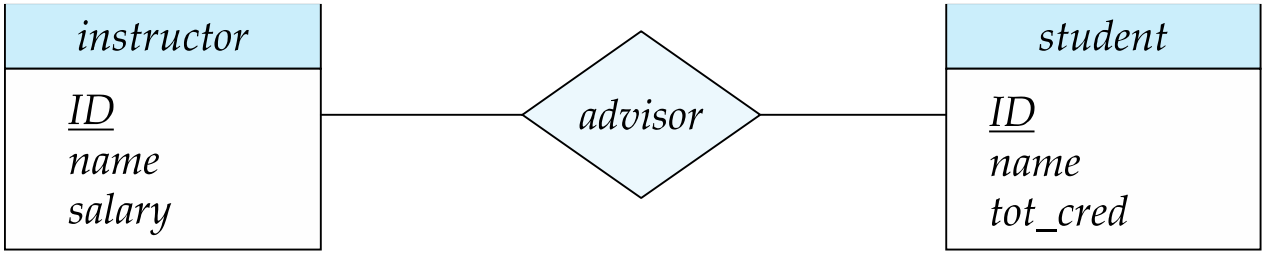
  student entity — relationship set — *instructor* entity

- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets
- Pictorially, we draw a line between related entities.



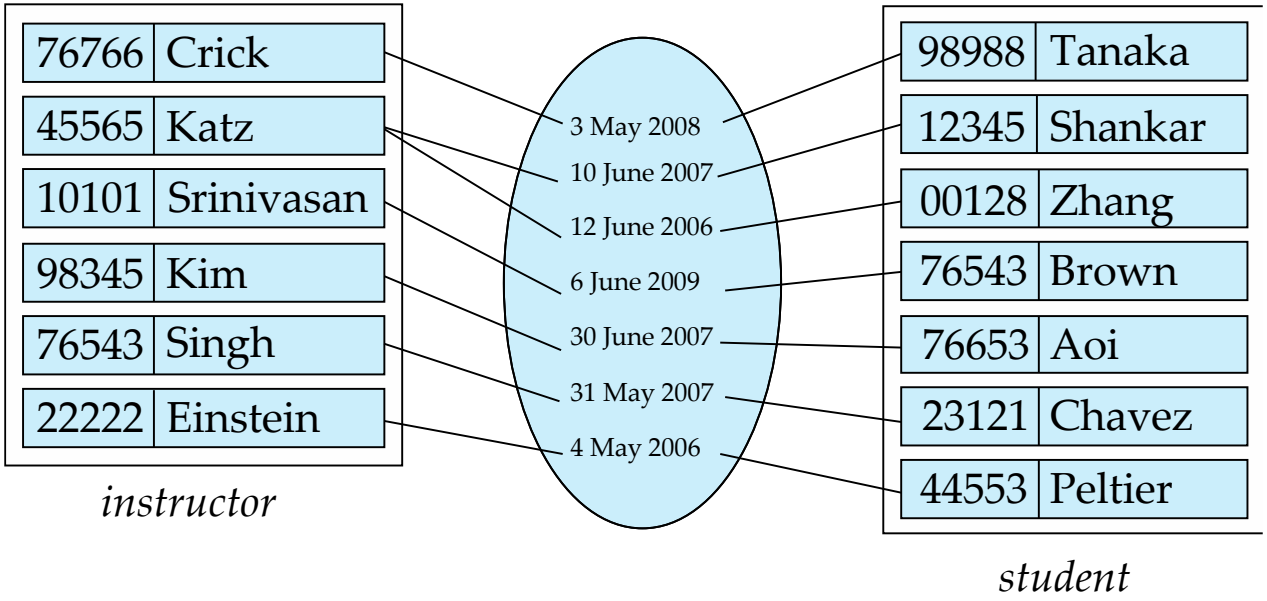
Representing Relationship Sets via ER Diagrams

- Diamonds represent relationship sets.

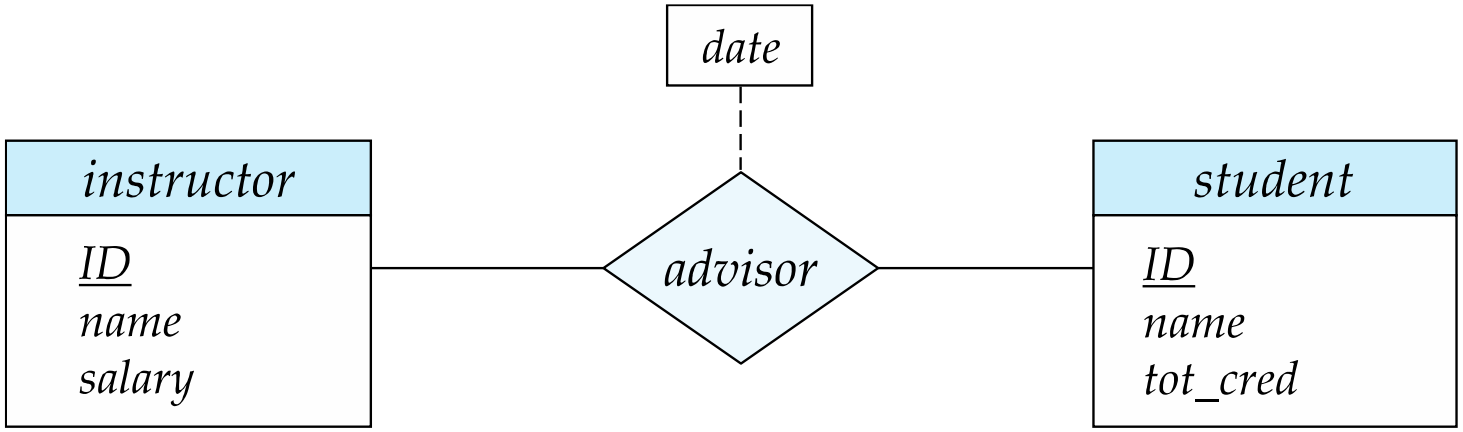


Relationship Sets (Cont.)

- An attribute can also be associated with a relationship set.
- For instance, the *advisor* relationship set between entity sets *instructor* and *student* may have the attribute *date* which tracks when the student started being associated with the advisor



Relationship Sets with Attributes

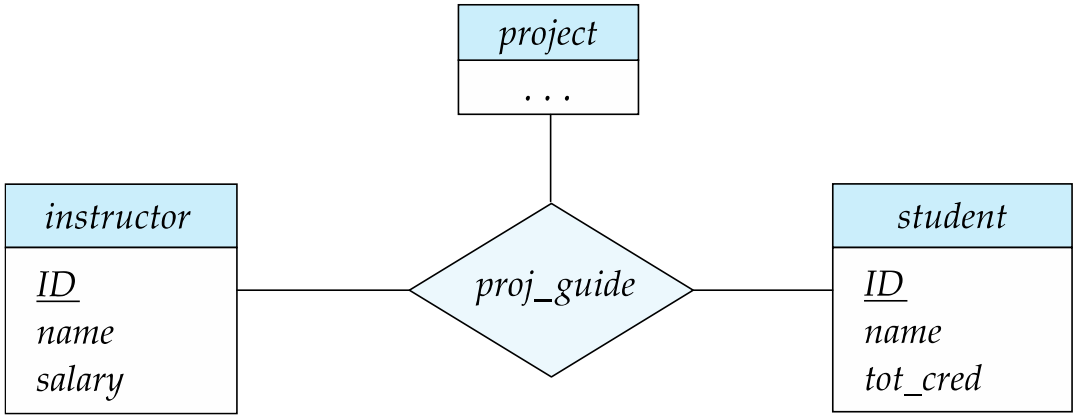


Degree of a Relationship Set

- Binary relationship
  - Involve two entity sets (or degree two)
  - Most relationship sets in a database system are binary
- Relationships between more than two entity sets are rare. Most relationships are binary
  - Example: *students* work on research *projects* under the guidance of an *instructor*.
  - Relationship *proj_guide* is a ternary relationship between *instructor, student,* and *project*

Non-binary Relationship Sets

- Most relationship sets are binary
- There are occasions when it is more convenient to represent relationships as non-binary
- E-R Diagram with a Ternary Relationship



Attributes

- Attribute types:
  - **Simple** and **composite** attributes
    - Simple Attribute cannot be divided further
      - Example: student age
    - Composite attribute can be divide further
      - Example: student name (first name, middle name, last name), student address
  - **Single-valued** and **Multi-valued** attributes
    - Single-valued attribute has only one value
      - Example: Student registration_number
    - Multi-valued attribute has more than one vlaue
      - Example: Student phone_numbers, address
  - Complex attributes
    - Composite + Multi-valued
      - Example: Two address of a student

Class Activity

- Can you tell few examples of single valued attributes?

Class Activity

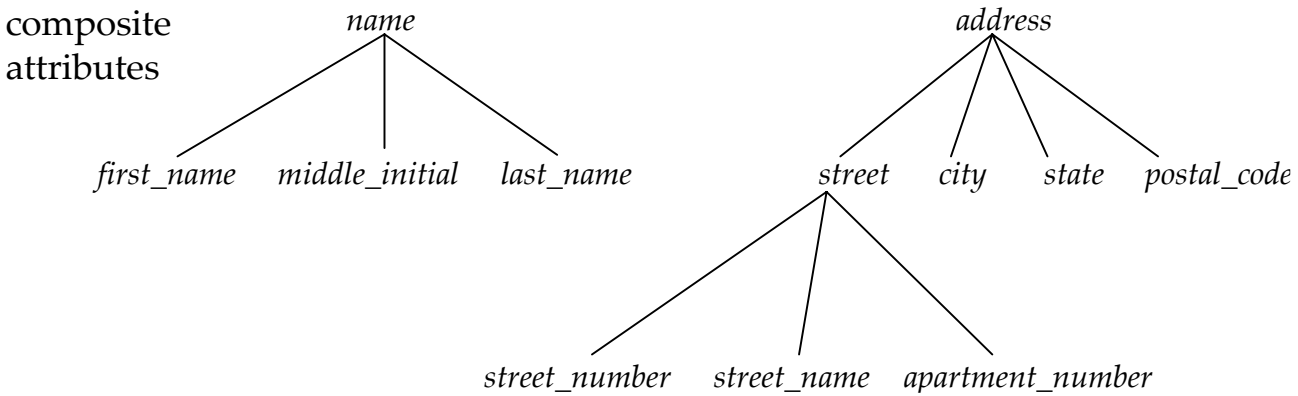
- Can you tell few examples of multi valued attributes?

Attributes

- Attribute types
  - Stored attributes
    - Cannot be derived
    - Example: date_of_birth
  - Derived attributes
    - Can be computed from other attributes
    - Example: age given date_of_birth
  - Key and non-key attributes
    - Key is an unique attribute
    - Example: registration number is a key attribute in student entity
- **Domain** – the set of permitted values for each attribute
- Representation of different types of attributes
  - Derived: represented in dotted eclipse
  - Multivalued: represented in double eclipse
  - Key Attribute: underlined in eclipse

Composite Attributes

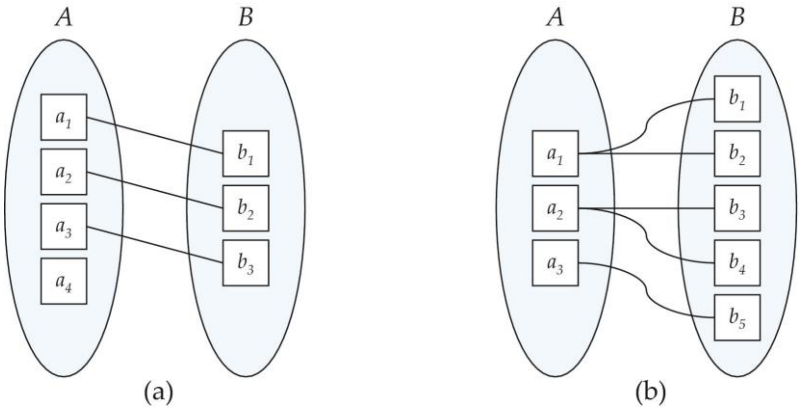
- Composite attributes allow us to divided attributes into subparts (other attributes).



Mapping Cardinality Constraints

- Express the number of entities to which another entity can be associated via a relationship set
- Most useful in describing binary relationship sets
  - **How two entities are associated with each other**
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one
  - One to many
  - Many to one
  - Many to many

Mapping Cardinalities

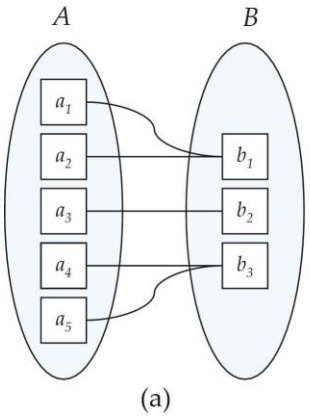


One to one

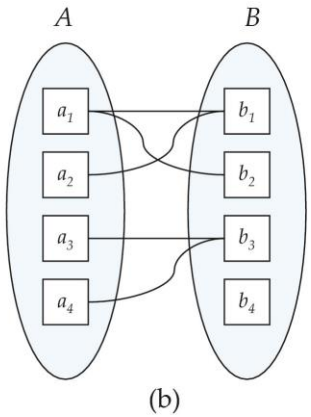
One to many

Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

Mapping Cardinalities



Many to one



Many to many

Representing Cardinality Constraints in ER Diagram

- We express cardinality constraints by drawing either a directed line (→), signifying “one”, or an undirected line (—), signifying “many”, between the relationship set and the entity set.
- **One-to-one relationship** between an *instructor* and a *student* :
  - A *student* is associated with at most one *instructor* via the relationship *advisor*
  - Similarly, an *instructor* is associated with at most one *student* with the same relationship

  - Example: HOD manages Department

One-to-One Relationship

- Can you draw an ER diagram for HOD Manage Department?

One-to-Many Relationship

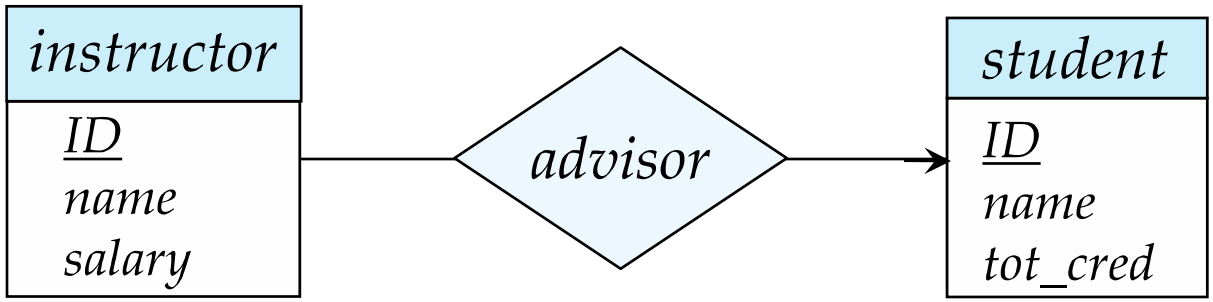
- One-to-many relationship between an *instructor* and a *student*
  - A student is associated with at most one instructor via *advisor*
  - An instructor is associated with several (including 0) students via *advisor*

One-to-Many Relationship

- Can you draw an ER diagram for Customer Give Order?

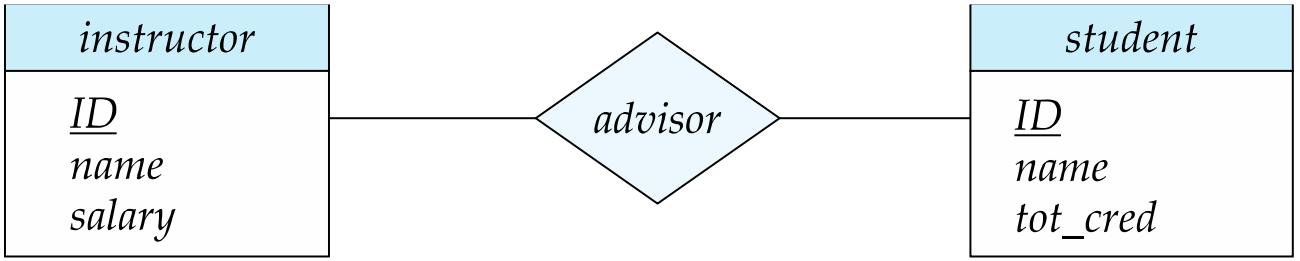
Many-to-One Relationships

- In a many-to-one relationship between an *instructor* and a *student,*
  - A student is associated with several (including 0) instructors via *advisor*
  - An instructor is associated with at most one student via *advisor*,



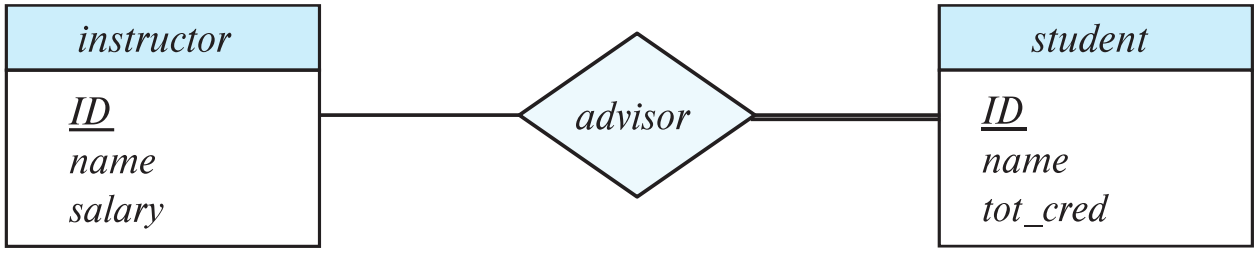
Many-to-Many Relationship

- An instructor is associated with several students via *advisor*
- A student is associated with several instructors via *advisor*



Total and Partial Participation

- **Total participation** (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

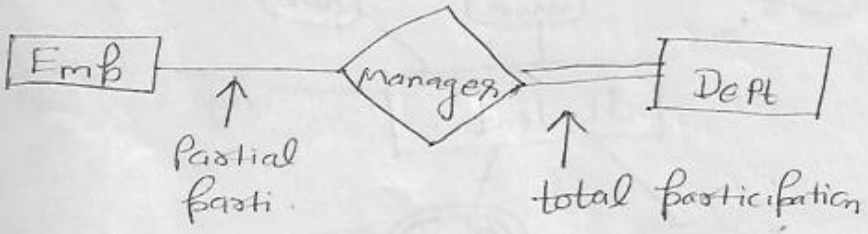


  - Participation of *student* in *advisor* relation is total
    - Every *student* must have an associated instructor
- **Partial participation**: Some entities may not participate in any relationship in the relationship set
  - Example: participation of *instructor* in *advisor* is partial

Total and Partial Participation

Participation :-

If every entity of entity set relating to relationship set then called TOTAL PARTICIPATION otherwise PARTIAL PARTICIPATION



- every dept there should be manager

Weak Entity Sets and Self Referential Relationship

- Weak entity set
  - Entity set with no key
- Attributes of weak entity set may not be able to differentiate the records
- Weak entity set is allowed in design of ER diagram
- RDBMS does not allow weak entity set
  - Combine relationship with entity

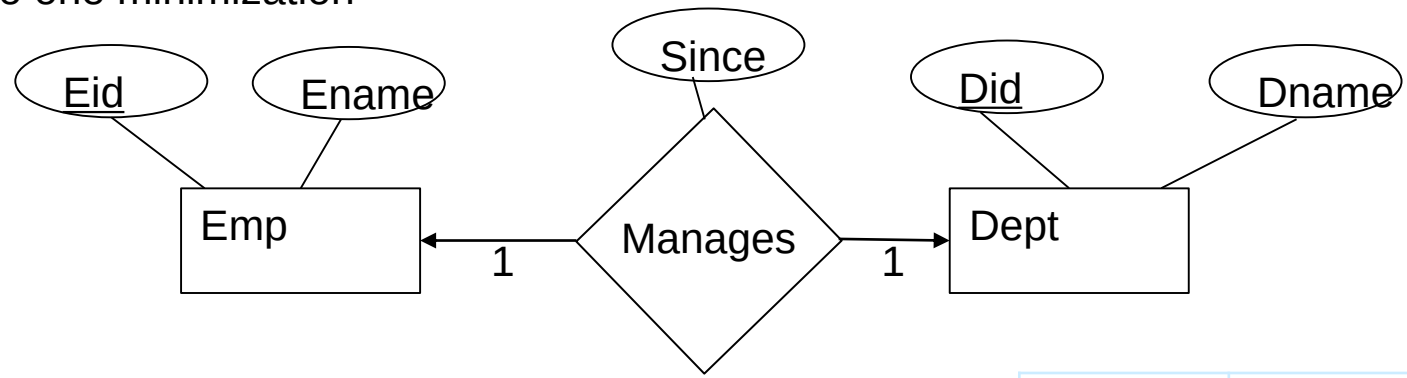
- Self Referential Relationship
  - One entity of entity set is related to other entity of same entity set
  - Example: In a company, a junior reports to a senior

ER Diagram Minimization

- ER Diagram Minimization
  - Convert ER Diagram into minimal such that minimum number of RDBMS tables would be formed
    - To reduce the number of overhead and complexity
  - One to one minimization
  - One to many minimization
  - Many to one minimization
  - Many to many minimization

ER Diagram Minimization

One to one minimization



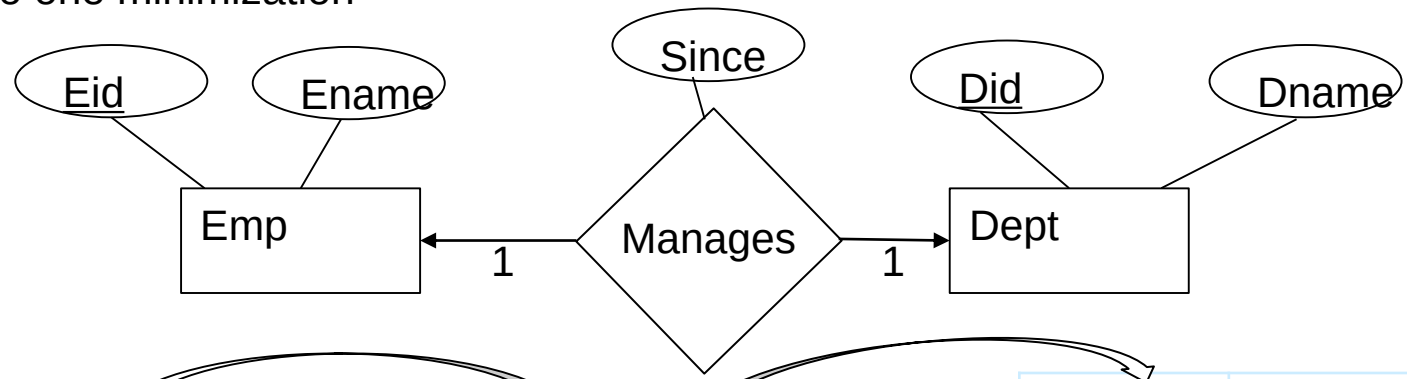
Eid	**Ename**
E1	A
E2	A
E3	B

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

- What will be attributes in relationship table, Manages?
- What will be Primary Key (PK) in relationship table, Manages?

ER Diagram Minimization

One to one minimization



Eid	**Ename**
E1	A
E2	A
E3	B

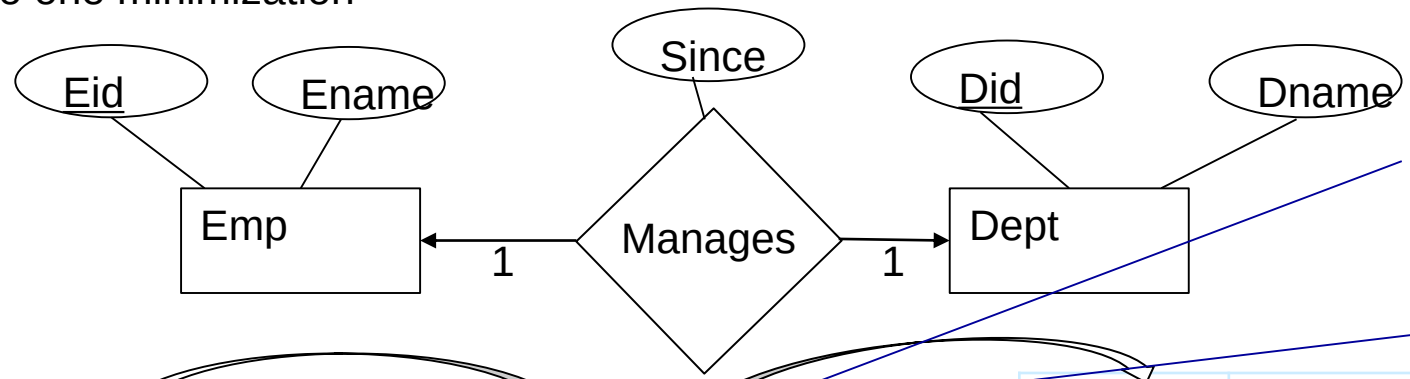
Eid	**Did**	**Since**
E1	D1	2009
E2	D2	2012
E3	D5	2016

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

- Can you tell if Manages Table is correct?

ER Diagram Minimization

One to one minimization



What will be attributes?

What will be Primary Key (PK)?

Eid	**Ename**
E1	A
E2	A
E3	B

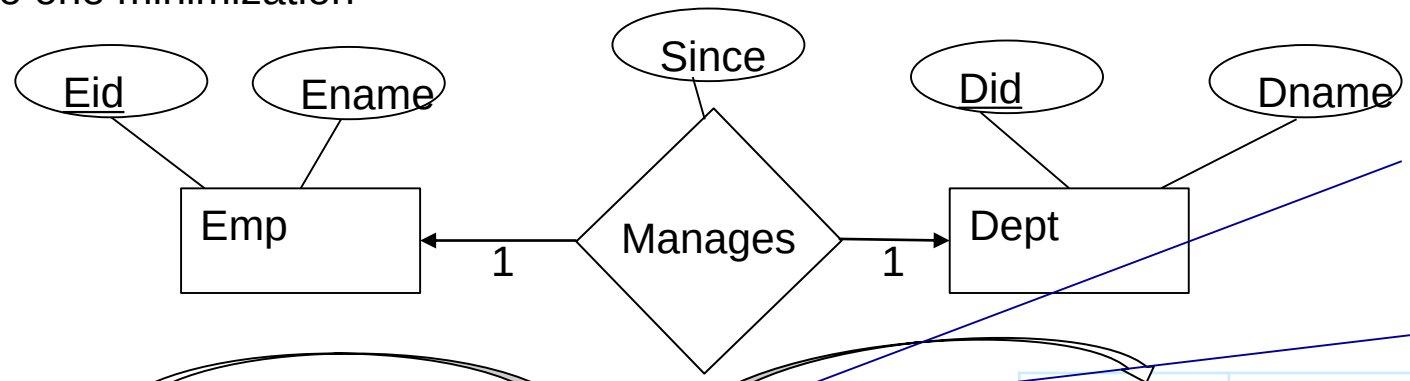
Eid	**Did**	**Since**
E1	D1	2009
E2	D2	2012
E3	D3	2016

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

Can we merge?
What will be PK?

ER Diagram Minimization

One to one minimization



What will be attributes?

What will be Primary Key (PK)?

Eid	**Ename**
E1	A
E2	A
E3	B

Eid	**Did**	**Since**
E1	D1	2009
E2	D2	2012
E3	D3	2016

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

Can we merge? What will be PK?

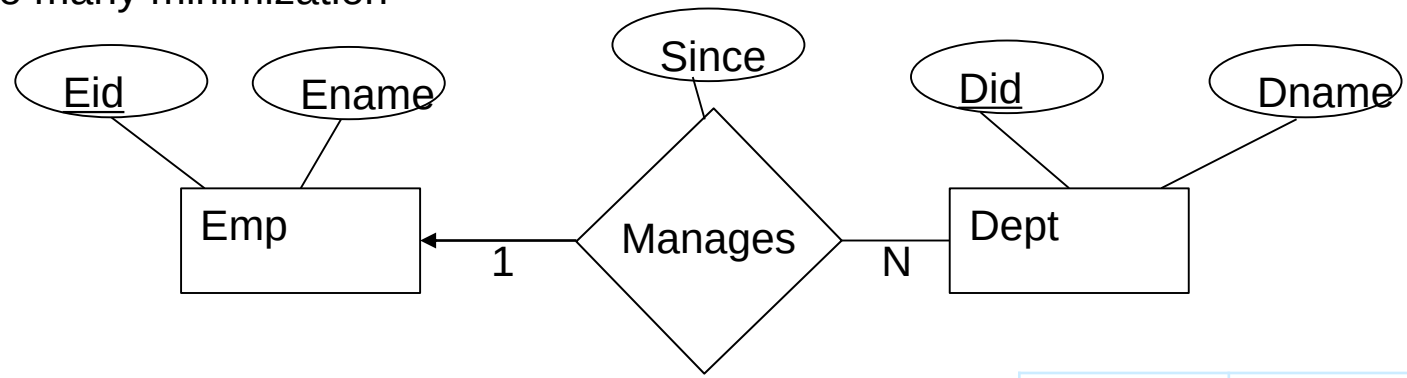
Eid	**Ename**
E1	A
E2	A
E3	B

Did	**Dname**	**Eid**	**Since**
D1	CS	E1	2009
D2	IT	E2	2012
D3	ME	E3	2016
D4	EE	NULL	NULL

Can we merge? What will be PK?

ER Diagram Minimization

One to many minimization



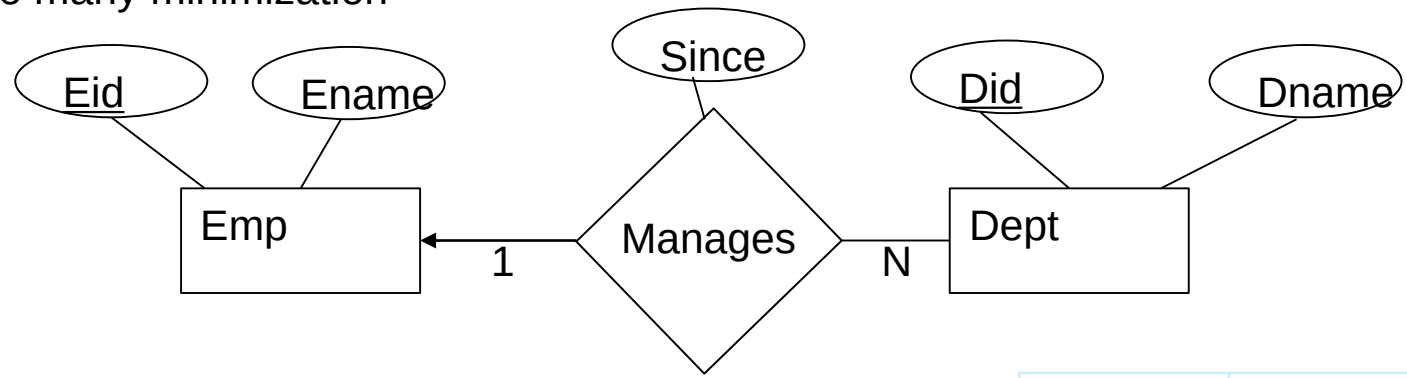
Eid	Ename
E1	A
E2	A
E3	B

Did	Dname
D1	CS
D2	IT
D3	ME
D4	EE

What will be Primary Key in Relationship Table, Manages (PK)?

ER Diagram Minimization

One to many minimization



Eid	**Ename**
E1	A
E2	A
E3	B

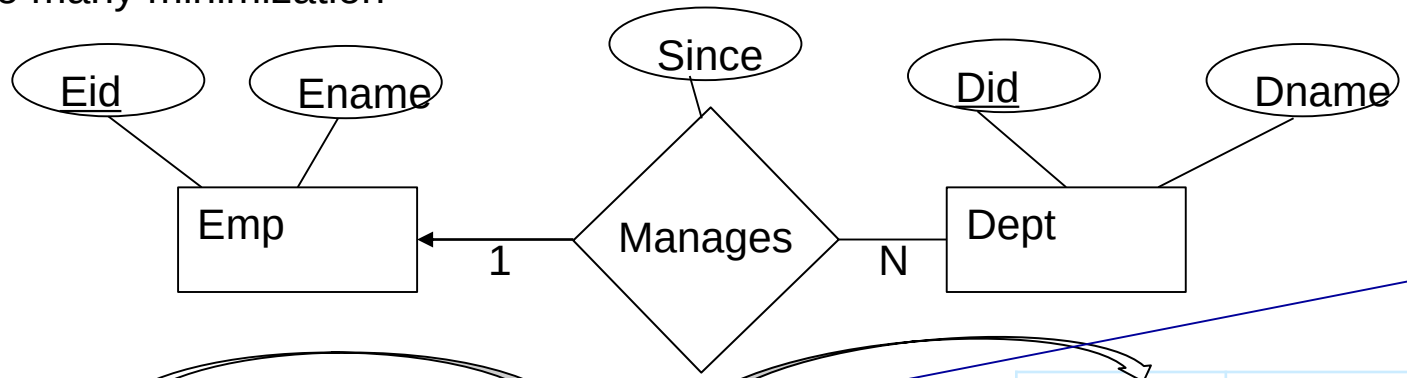
Eid	**Did**	**Since**
E1	D1	2009
E1	D2	2012
E2	D3	2016

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

What will be Primary Key in Relationship Table, Manages (PK)?
How I will remember? Easy way?

ER Diagram Minimization

One to many minimization



What will be Primary Key (PK)?

Eid	**Ename**
E1	A
E2	A
E3	B

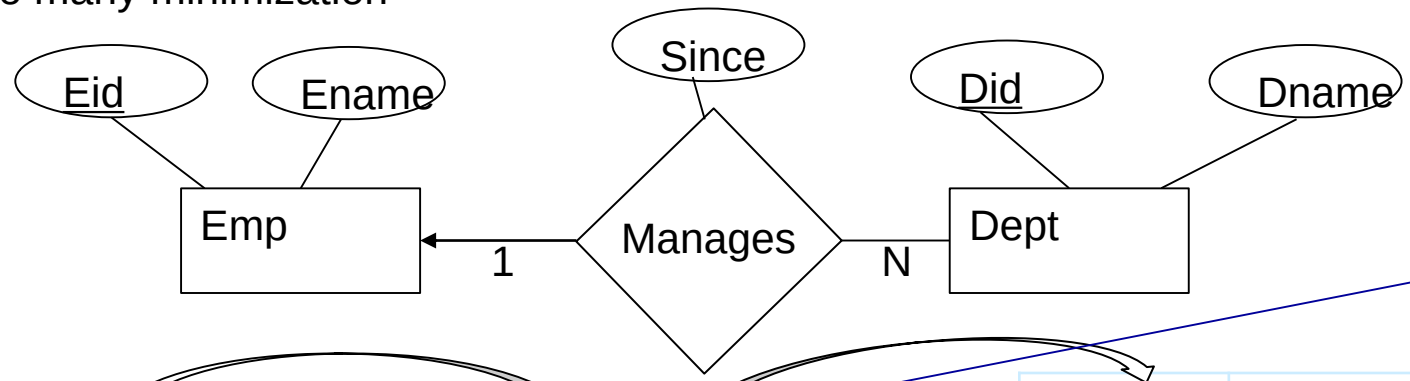
Eid	**Did**	**Since**
E1	D1	2009
E1	D2	2012
E2	D3	2016

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

Can we merge?

ER Diagram Minimization

One to many minimization



What will be Primary Key (PK)?

Eid	**Ename**
E1	A
E2	A
E3	B

Eid	**Did**	**Since**
E1	D1	2009
E1	D2	2012
E2	D3	2016

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

Can we merge?

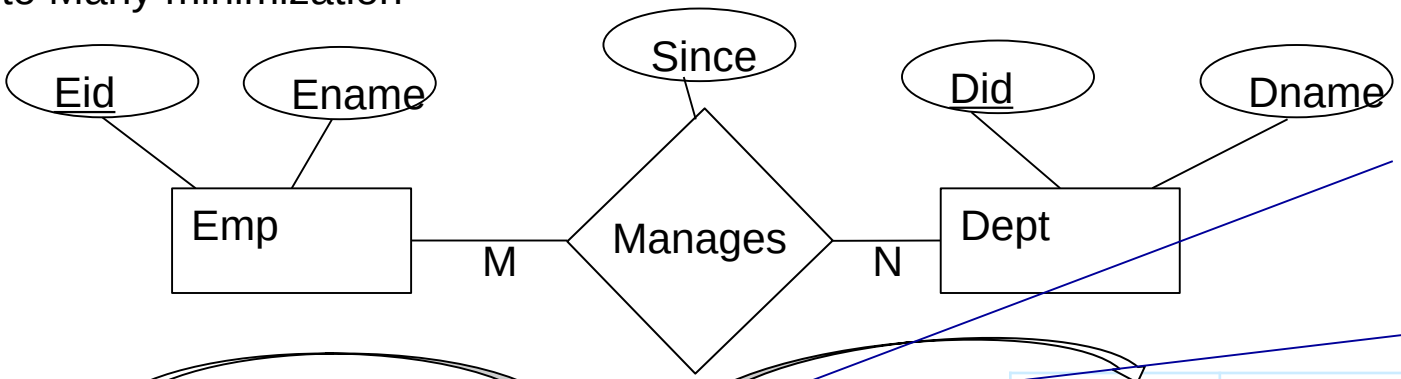
Eid	**Ename**
E1	A
E2	A
E3	B

Did	**Dname**	**Eid**	**Since**
D1	CS	E1	2009
D2	IT	E1	2012
D3	ME	E2	2016
D4	EE	NULL	NULL

Can we merge?

ER Diagram Minimization

Many to Many minimization



What will be attributes?

What will be Primary Key (PK)?

Eid	**Ename**
E1	A
E2	A
E3	B

Eid	**Did**	**Since**
E1	D1	2009
E1	D2	2012
E3	D2	2016

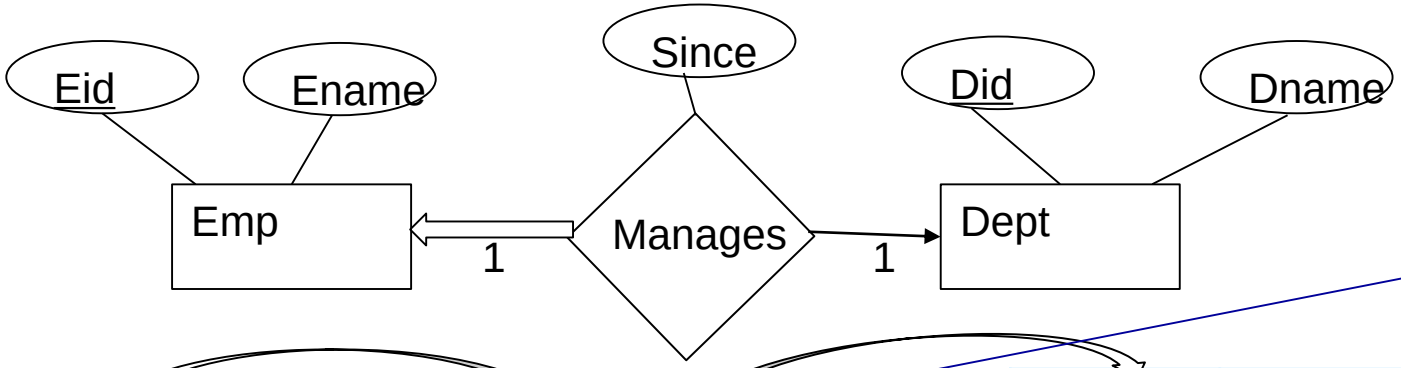
Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

Can we merge? What will be PK?

Can we merge? What will be PK?

ER Diagram Minimization

One to one minimization with total participation



What will be Primary Key (PK)?

Eid	**Ename**
E1	A
E2	A
E3	B

Eid	**Did**	**Since**
E1	D1	2009
E2	D2	2012
E3	D3	2016

Did	**Dname**
D1	CS
D2	IT
D3	ME
D4	EE

Can we merge?

Eid	**Ename**
E1	A
E2	A
E3	B

+

Did	**Dname**	**Eid**	**Since**
D1	CS	E1	2009
D2	IT	E2	2012
D3	ME	E3	2016
D4	EE	NULL	NULL

Can we merge?

References

- Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan. *Database system concepts*. Vol. 6. New York: McGraw-Hill, 1997.
- Ramez Elmasri, Shamkant B. Navathe. *Fundamentals of Database Systems.* Edition 6. Pearson, 2010.