

# CS & IT ENGINEERING



## Database Management System

DBMS

Lecture No. 9

By- Ravindrababu Ravula Sir



# Recap of Previous Lecture



Topic

Nested Queries



# Topics to be Covered



Topic

Topic

Topic

Topic

Topic

ER Model (Entry Relationship Diagram)

Binary Relationship Set

One many mapping

RDBMS Design

Many to Many Mapping



# Topics to be Covered



Topic

Topic

Self Referential Relationship Set

Weak Entity Set





# Topic: ER Model (Entry Relationship Diagram)



**High level DB Design:** ✓

Diagrammatic representation of DB design.

In SDLC (Software development life cycle), we have:

Requirements

Design [here we use UML/Flow chart etc. for designing in software]

Implement

In DB development, we use ER-diagrams for design.



## Topic: ER Model (Entry Relationship Diagram)

In SDLC (Software development life cycle), we have:

Requirements



Design [here we use UML/Flow chart etc. for designing in software]



Implement

In DB development, we use ER-diagram for design.

For every proper development, diagrammatic representation is a must.





## Topic: ER Model (Entry Relationship Diagram)



**There are several steps to design a DB.**

1. Requirements
2. Conceptual / Logical DB design (using ER diagram) after which we convert ER Diagram into RDBMS table.
3. Apply Normalization, to eliminate or reduce the redundancy.
4. Physical DB design(Indexing design) This is to reduce the excess cost.
5. Application / Security For allowing who can access software.



# Topic: ER Model (Entry Relationship Diagram)



**There are several steps to design a DB.**

- High level DB design }
  - 1. Requirements ✓
  - 2. Conceptual / Logical DB design (using ER diagram) after which we convert ER Diagram into RDBMS table. ✓
  
- Low level DB design }
  - 3. Apply Normalization, to eliminate or reduce the redundancy.
  - 4. Physical DB design(Indexing design) This is to reduce the excess cost.
  - 5. Application / Security For allowing who can access software.



# Topic: ER Model (Entry Relationship Diagram)



## Main Components of ER Diagrams:

1. Attributes ✓
2. Entity Sets ✓
3. Relationship Sets ✓



# Topic: ER Model (Entry Relationship Diagram)



## Attributes:

Attribute



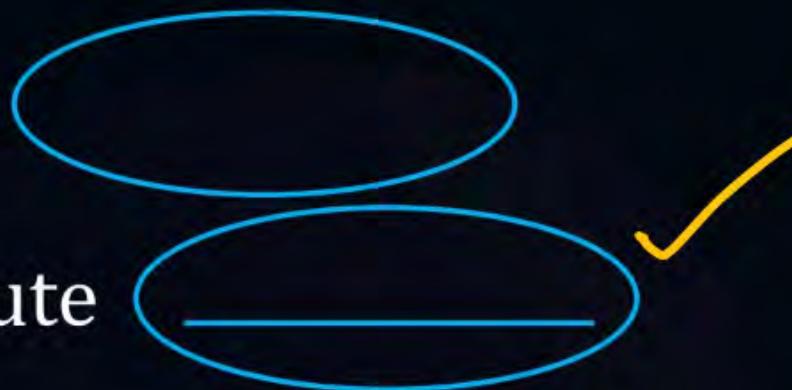


# Topic: ER Model (Entry Relationship Diagram)



## Attributes:

Attribute



Key Attribute



# Topic: ER Model (Entry Relationship Diagram)

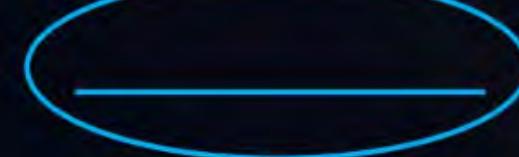


## Attributes:

Attribute



Key Attribute



Multi-Valued Attribute

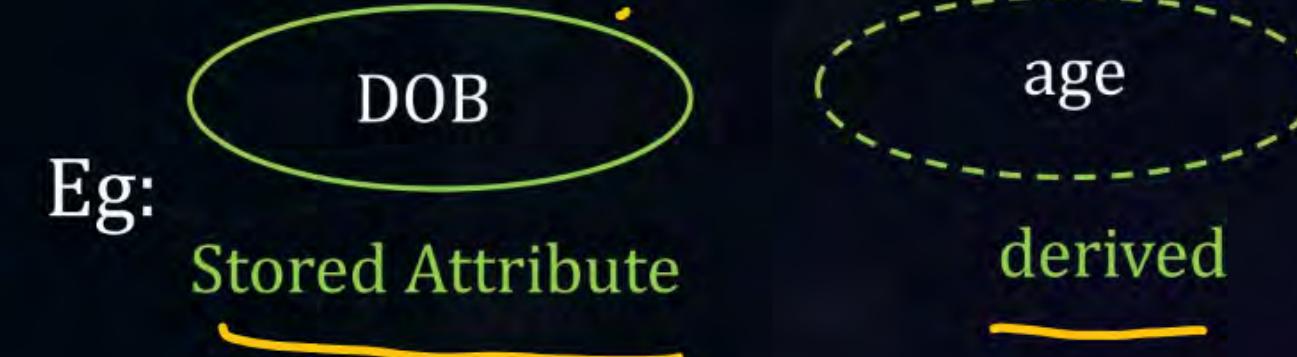
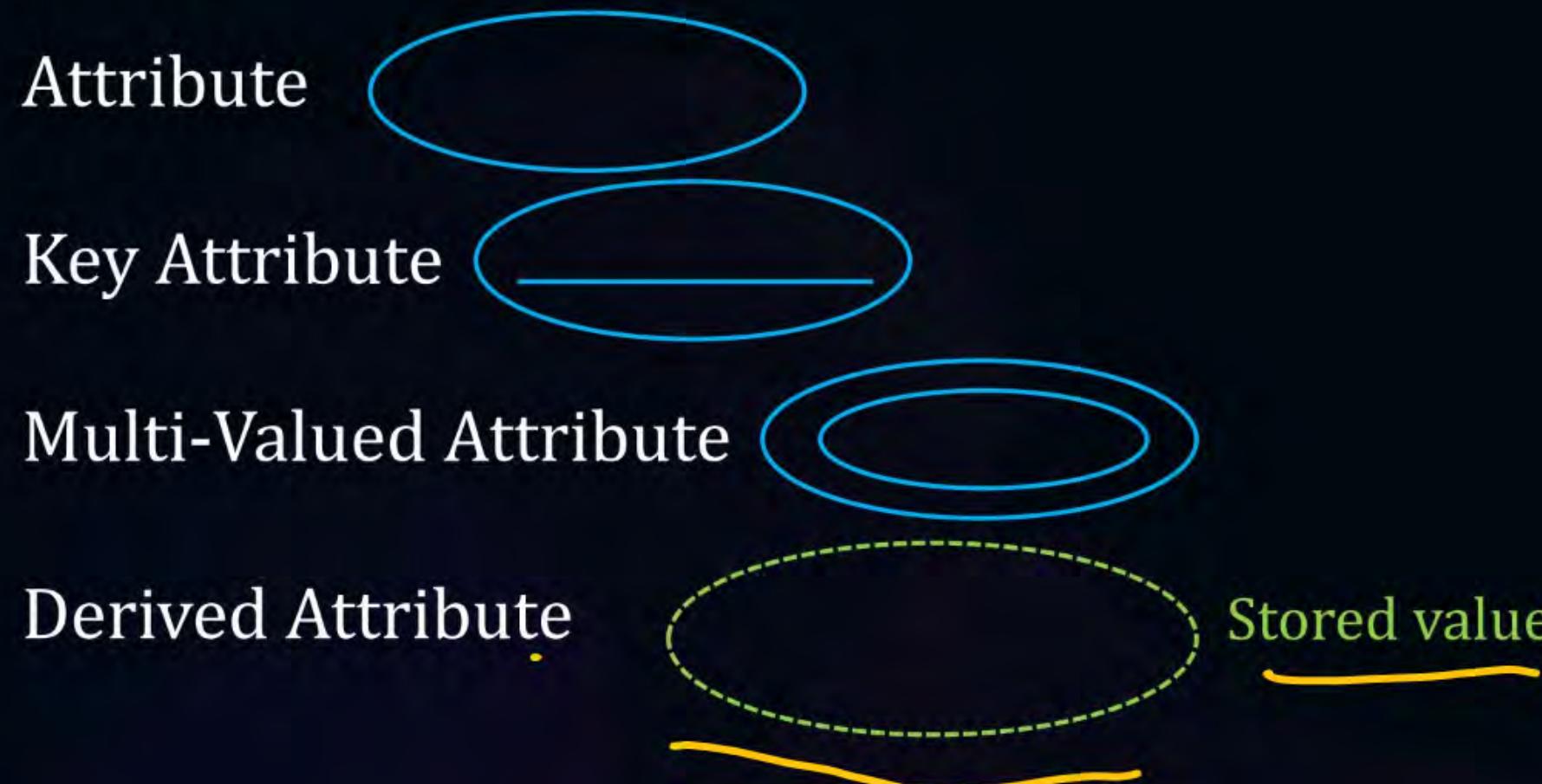




# Topic: ER Model (Entry Relationship Diagram)



## Attributes:





# Topic: ER Model (Entry Relationship Diagram)



## Attributes:

Attribute



Key Attribute



Multi-Valued Attribute



Derived Attribute



Composite Attribute: can be represented as 2 or more attr



Stored value

Eg:





## Topic: ER Model (Entry Relationship Diagram)



Entity Set: Set of similar entries (Tuples)  

*entities* ✓

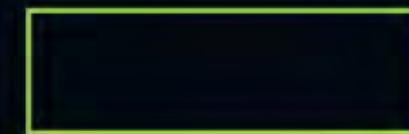
Ex: students



# Topic: ER Model (Entry Relationship Diagram)



**Entity Set:** Set of similar entries (Tuples)



Ex: students

**Relationship Set:** This relates two or more entity sets



Ex: Enroll

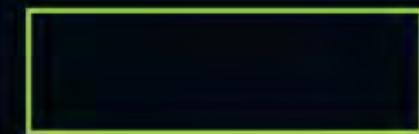




# Topic: ER Model (Entry Relationship Diagram)



**Entity Set:** Set of similar entries (Tuples)



Ex: students

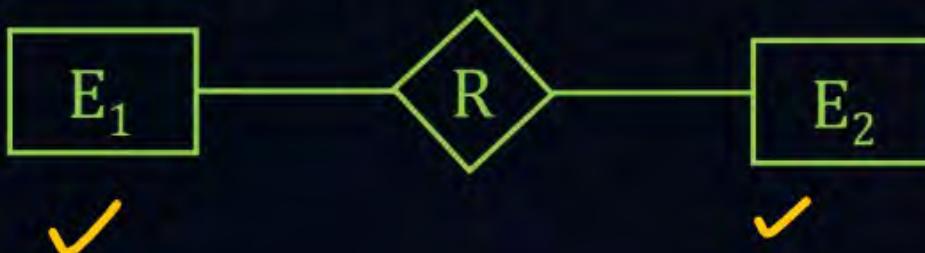
**Relationship Set:** This relates two or more entity sets



Ex: Enroll



Binary Relationship

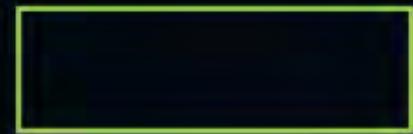




# Topic: ER Model (Entry Relationship Diagram)

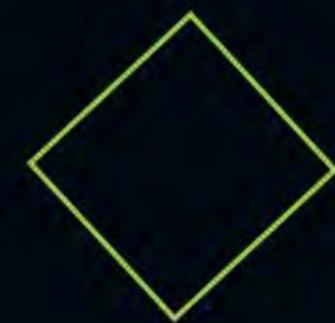


**Entity Set:** Set of similar entries (Tuples)



Ex: students

**Relationship Set:** This relates two or more entity sets

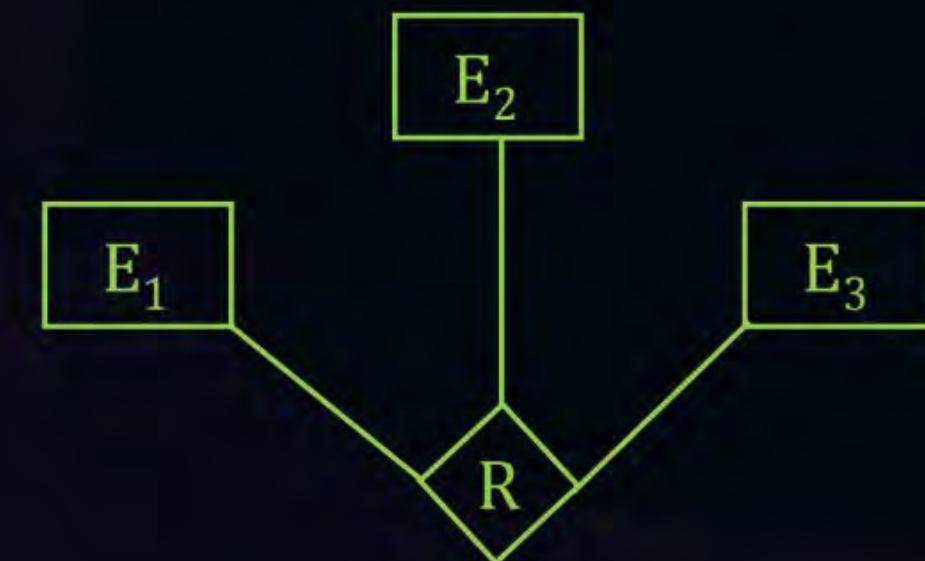


Ex: Enroll

Binary Relationship



Ternary Relationship

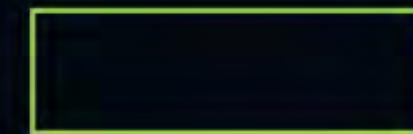




# Topic: ER Model (Entry Relationship Diagram)



**Entity Set:** Set of similar entries (Tuples)



Ex: students

**Relationship Set:** This relates two or more entity sets

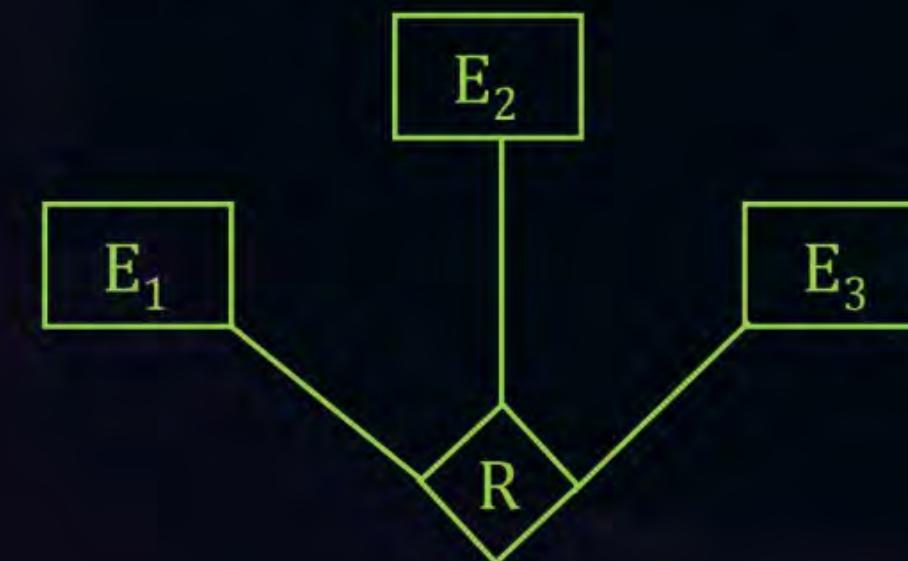


Ex: Enroll

Binary Relationship



Ternary Relationship



n-ary relationship





## Topic: ER Model (Entry Relationship Diagram)



### Participation:

If every entity of entity set is related to relationship set then

it is Total Participation  $\Rightarrow$  Must be 100% participation (denoted by  $=$ )

else Partial Participation, may/may not have 100% participation (denoted by  $-$ )



## Topic: ER Model (Entry Relationship Diagram)

**Example2:** EMP & Dept are entity Set. Manages is relationship set Such that every dept there must be manager.



## Topic: ER Model (Entry Relationship Diagram)

**Example2:** EMP & Dept are entity Set. Manages is relationship set Such that every dept there must be manager.

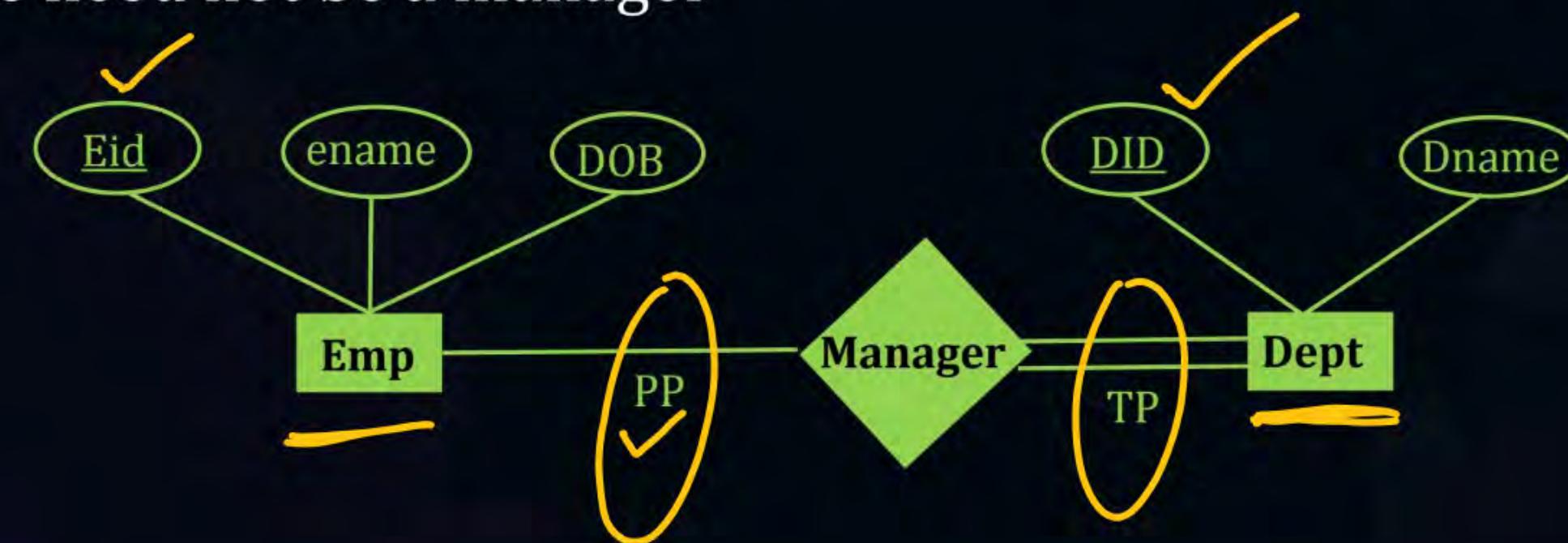
Note that every Dept must have a manager, meaning we have total participation  
Every employee need not be a manager



## Topic: ER Model (Entry Relationship Diagram)

**Example2:** EMP & Dept are entity Set. Manages is relationship set Such that every dept there must be manager.

Note that every Dept must have a manager, meaning we have **total participation**  
Every employee need not be a manager



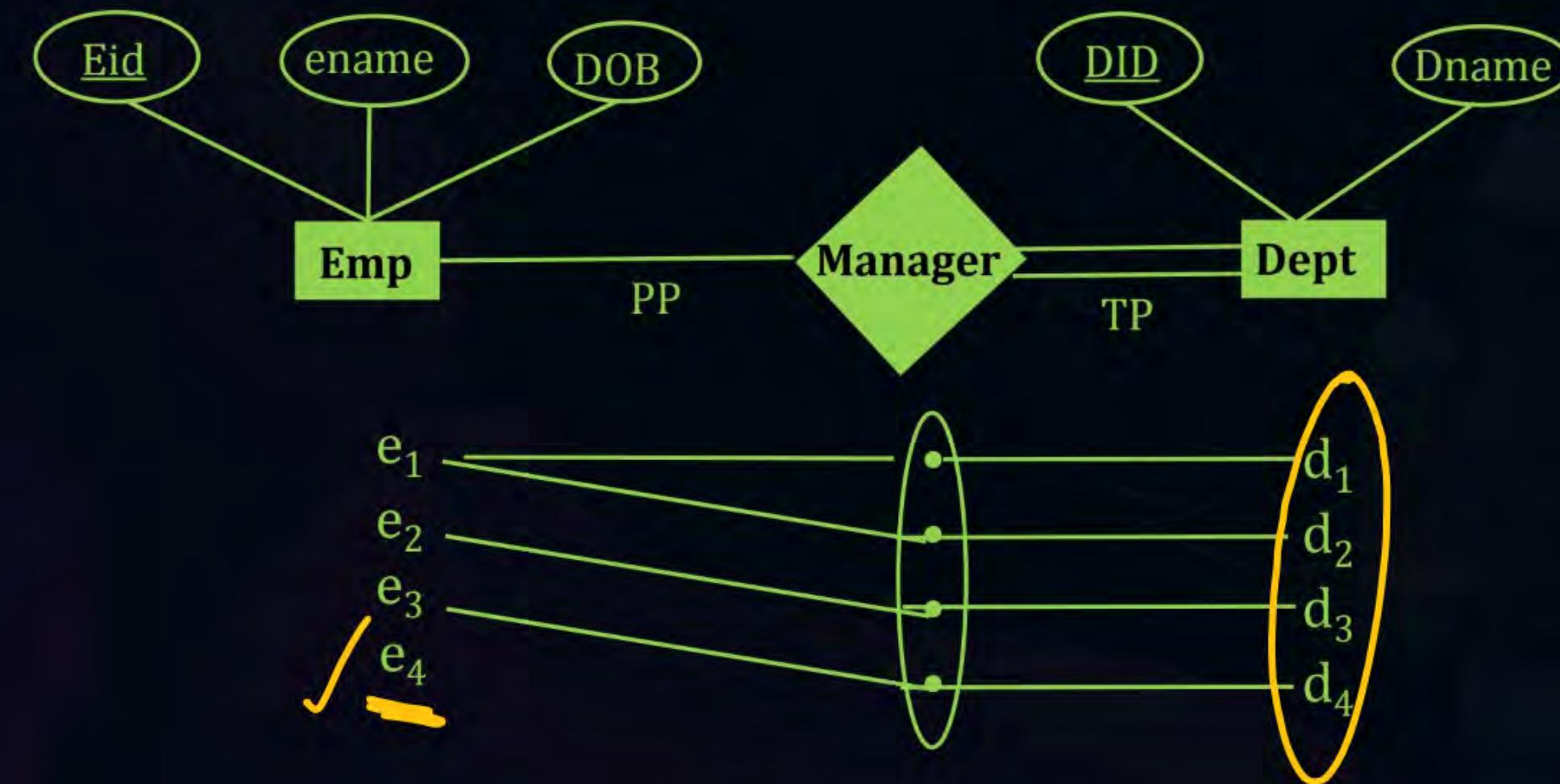


## Topic: ER Model (Entry Relationship Diagram)



**Example2:** EMP & Dept are entity Set. Manages is relationship set Such that every dept there must be manager.

Note that every Dept must have a manager, meaning we have **total participation**  
Every employee need not be a manager





## Topic: ER Model (Entry Relationship Diagram)



### Cardinality (mapping) of a relation:

If one employee can manage many dept. then its **many mapping**.  
if one employee can manage only one dept, then its **one mapping**.



## Topic: ER Model (Entry Relationship Diagram)

### Cardinality (mapping) of a relation:

If one employee can manage many dept, then its **many mapping**.

if one employee can manage only one dept, then its **one mapping**.

⇒ Mapping is how many objects of one entity can be mapped to an object of other entity



## Topic: ER Model (Entry Relationship Diagram)



### Cardinality (mapping) of a relation:

If one employee can manage many dept, then its **many mapping**.

if one employee can manage only one dept, then its **one mapping**.

⇒ Mapping is how many objects of one entity can be mapped to an object of other entity

#### 1. **ONE Mapping:** [0...1]

⇒ can relate to almost one object

⇒ either 0 or more 1

⇒ represented by (→)





## Topic: ER Model (Entry Relationship Diagram)



### Cardinality (mapping) of a relation:

If one employee can manage many dept, then its **many mapping**.

if one employee can manage only one dept, then its **one mapping**.

⇒ Mapping is how many objects of one entity can be mapped to an object of other entity

#### 1. **ONE Mapping:** [0...1]

- ⇒ can relate to almost one object
- ⇒ either 0 or more 1
- ⇒ represented by (→)



#### 2. **Many Mapping:** [0...\*]

- ⇒ 0 or more
- ⇒ [0...\*]
- ⇒ represented by (-)





## Topic: ER Model (Entry Relationship Diagram)



### Cardinality (mapping) of a relation:

If one employee can manage many dept, then its **many mapping**.

if one employee can manage only one dept, then its **one mapping**.

⇒ Mapping is how many objects of one entity can be mapped to an object of other entity

#### 1. **ONE Mapping:** [0...1]

- ⇒ can relate to almost one object
- ⇒ either 0 or more 1
- ⇒ represented by (→)

#### 2. **Many Mapping:** [0...\*]

- ⇒ 0 or more
- ⇒ [0...\*]
- ⇒ represented by (—)





## Topic: Binary Relationship Set



**Binary Relationship Set:** 4 possible Mapping.

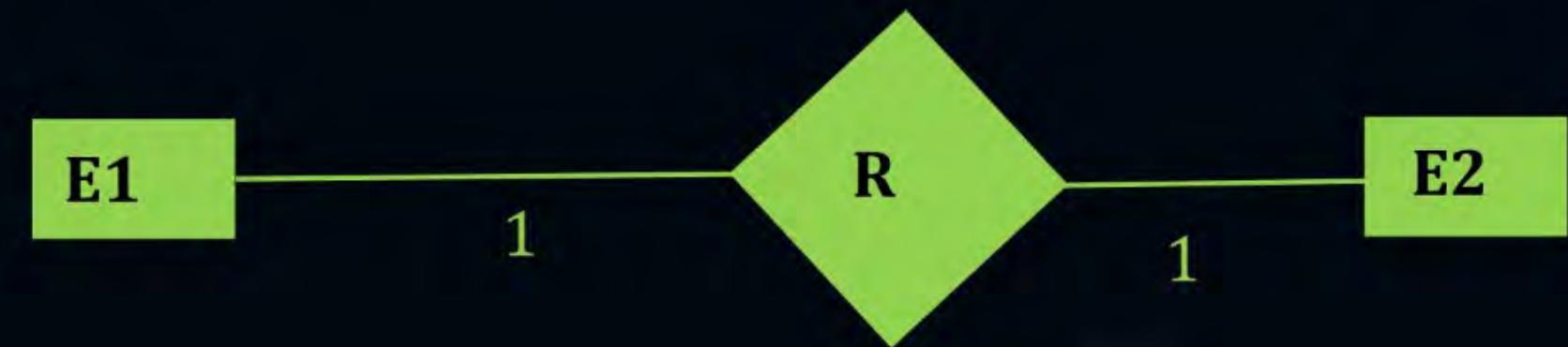




## Topic: Binary Relationship Set

**Binary Relationship Set:** 4 possible Mapping.

1. One : One Mapping(1:1)





## Topic: Binary Relationship Set

**Binary Relationship Set:** 4 possible Mapping.

1. One : One Mapping(1:1)
2. One : Many Mapping (1:M)





## Topic: Binary Relationship Set

**Binary Relationship Set:** 4 possible Mapping.

1. One : One Mapping(1:1)
2. One : Many Mapping (1:M)
3. Many : One Mapping (M:1)





## Topic: Binary Relationship Set

**Binary Relationship Set:** 4 possible Mapping.

1. One : One Mapping(1:1)
2. One : Many Mapping (1:M)
3. Many : One Mapping (M:1)
4. Many : Many Mapping (M:N)

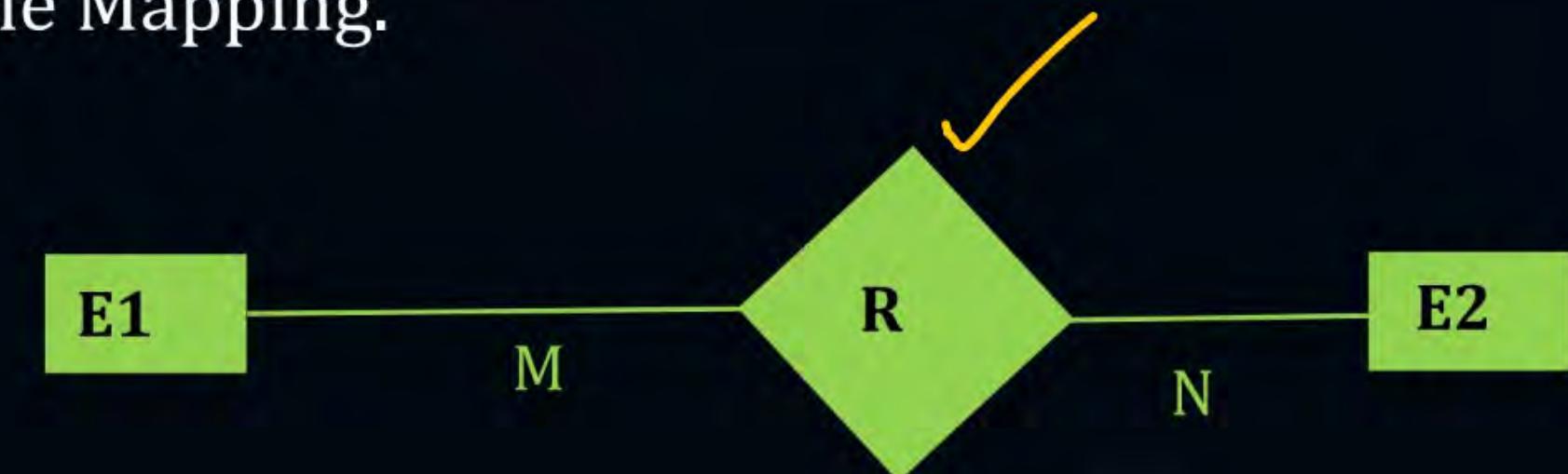




## Topic: Binary Relationship Set

**Binary Relationship Set:** 4 possible Mapping.

1. One : One Mapping(1:1)
2. One : Many Mapping (1:M)
3. Many : One Mapping (M:1)
4. Many : Many Mapping (M:N)



- Based on required mapping, the Candidate Key is designed for relationship Set.



## Topic: Ternary Relationship

### Ternary Relationship:

Here, total of 8 possible mappings are possible.

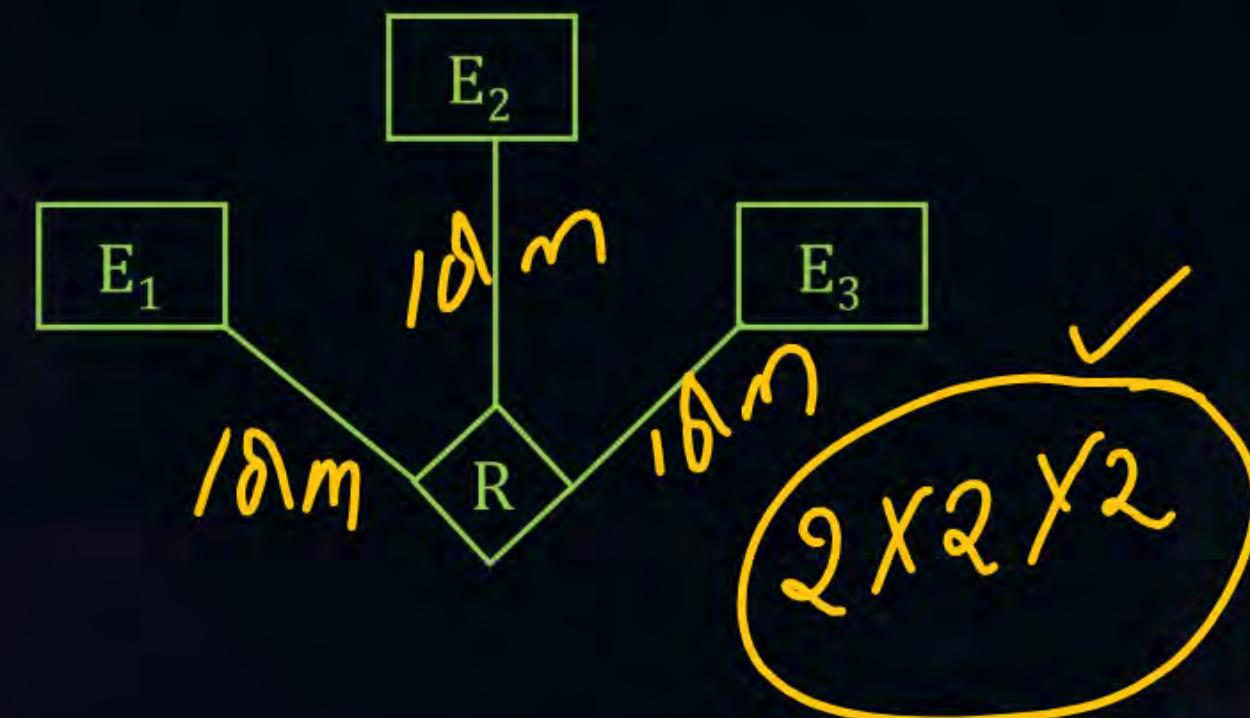
Again, based on the requirement of mapping, the Candidate key are defined.

8 possible mapping

1 : 1 : 1

1 : 1 : M

:





## Topic: One many mapping



We are gonna concentrate on 3 problems mainly:

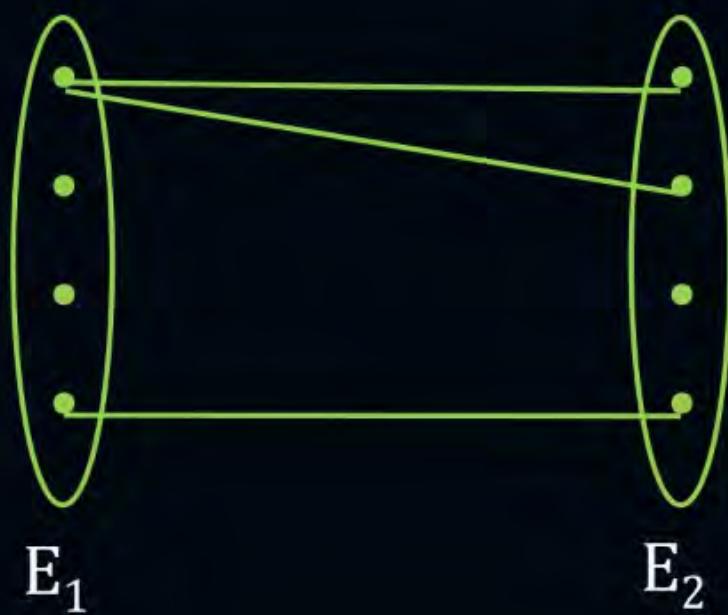
1. How to draw ER diagram for given RDMS mapping
2. What is CK of a relationship?
3. How many min RDBMS tables are required to store the ER diagram?





## Topic: One many mapping

### One to Many Relationship:



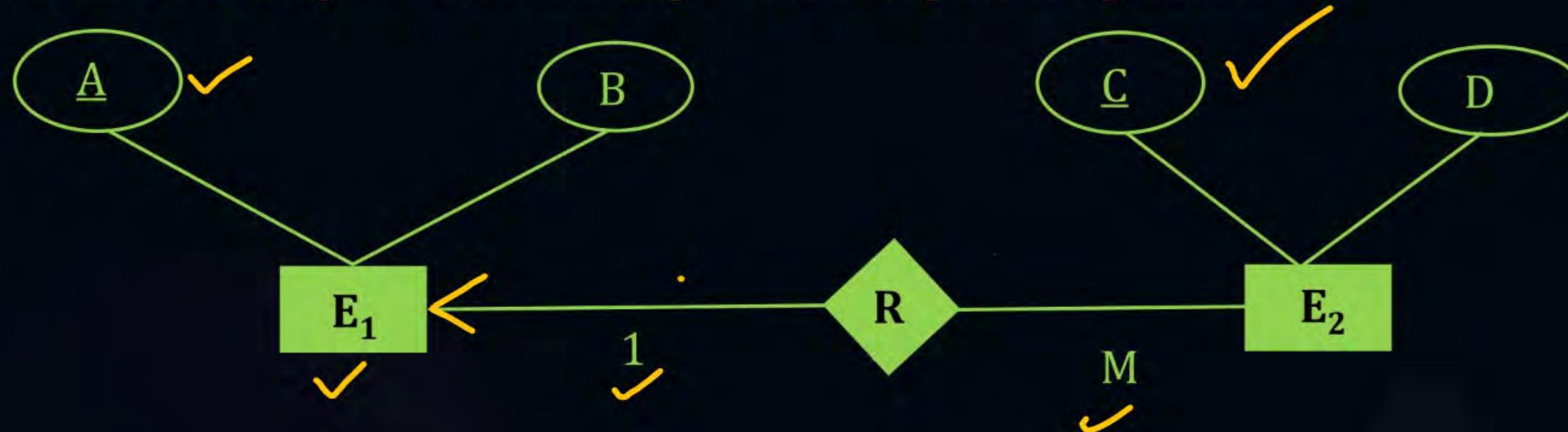
Say, One object of E<sub>1</sub> can relate to many object of E<sub>2</sub>

But each object of E<sub>2</sub> can relate to at most one object of E<sub>1</sub>



# Topic: One many mapping

## One to Many Relationship: Partial participation





## Topic: One many mapping

### One to Many Relationship: Partial participation



Attributes of the relationship set R are the key attribute of entities which are relating to a relationship.

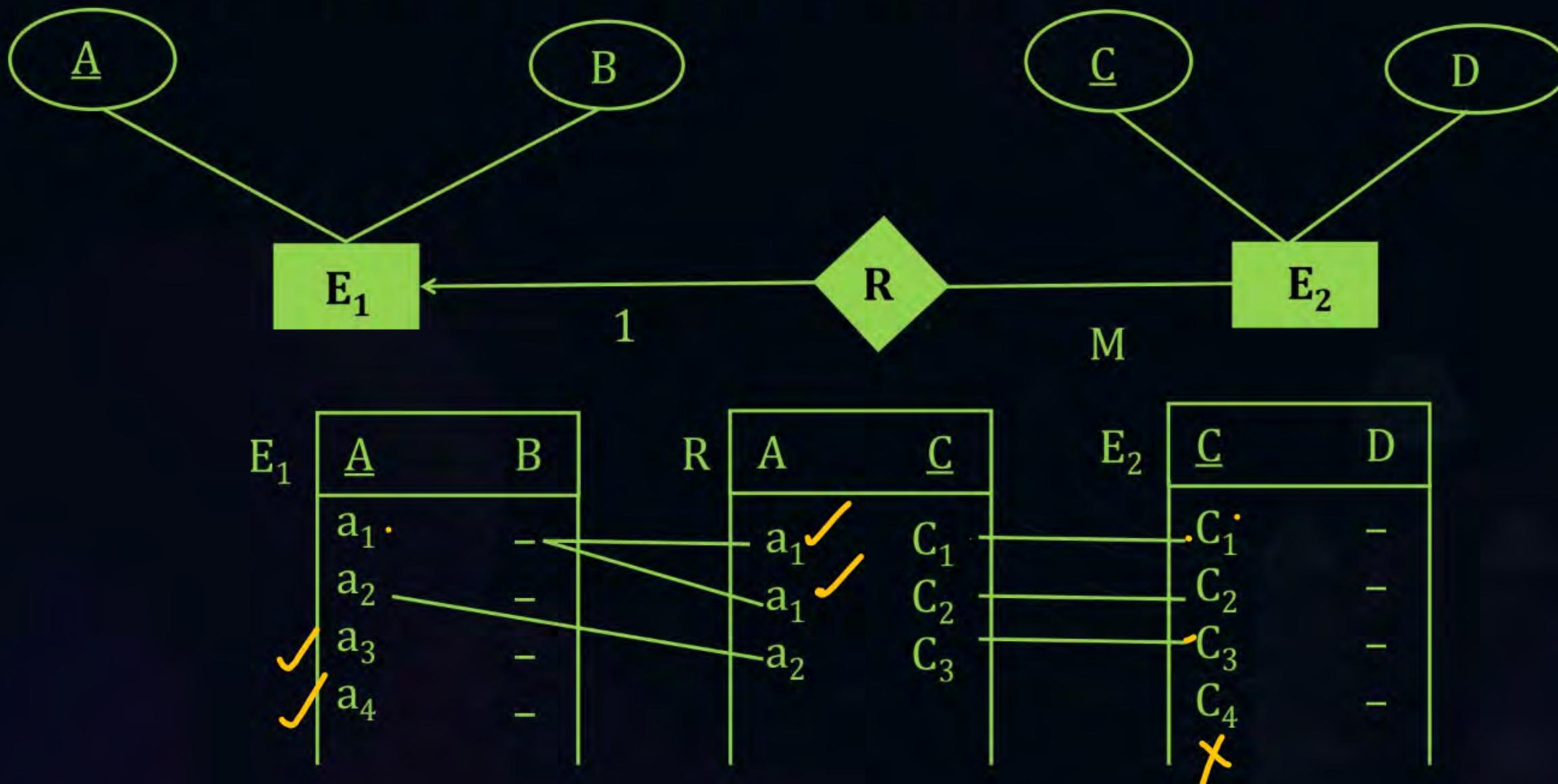
So, R attributes are key attribute of  $E_1$  and Key attribute of  $E_2$ .

So, A and C are attributes of Relationship R



# Topic: One many mapping

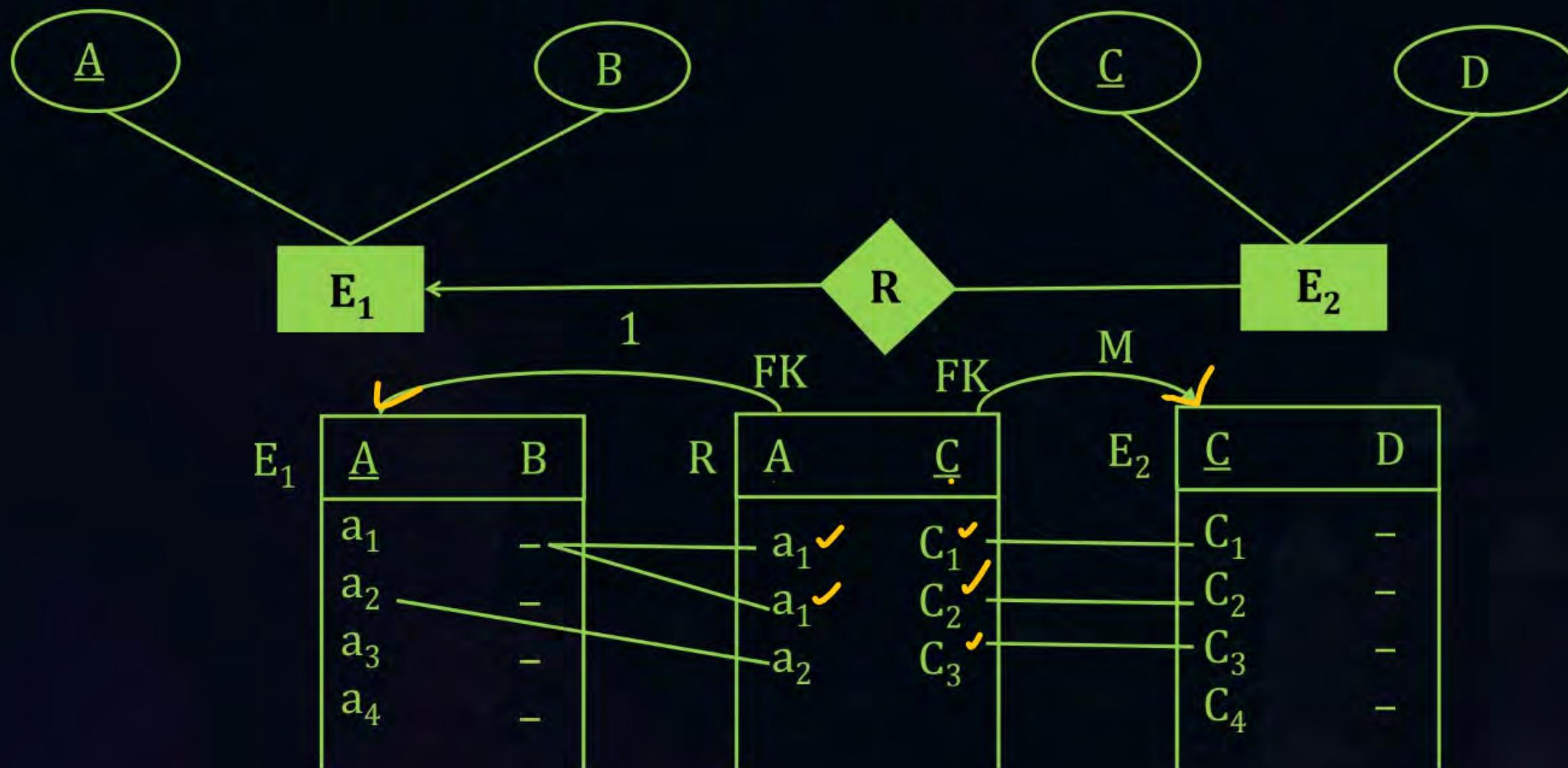
## One to Many Relationship: Partial participation





# Topic: One many mapping

## One to Many Relationship: Partial participation

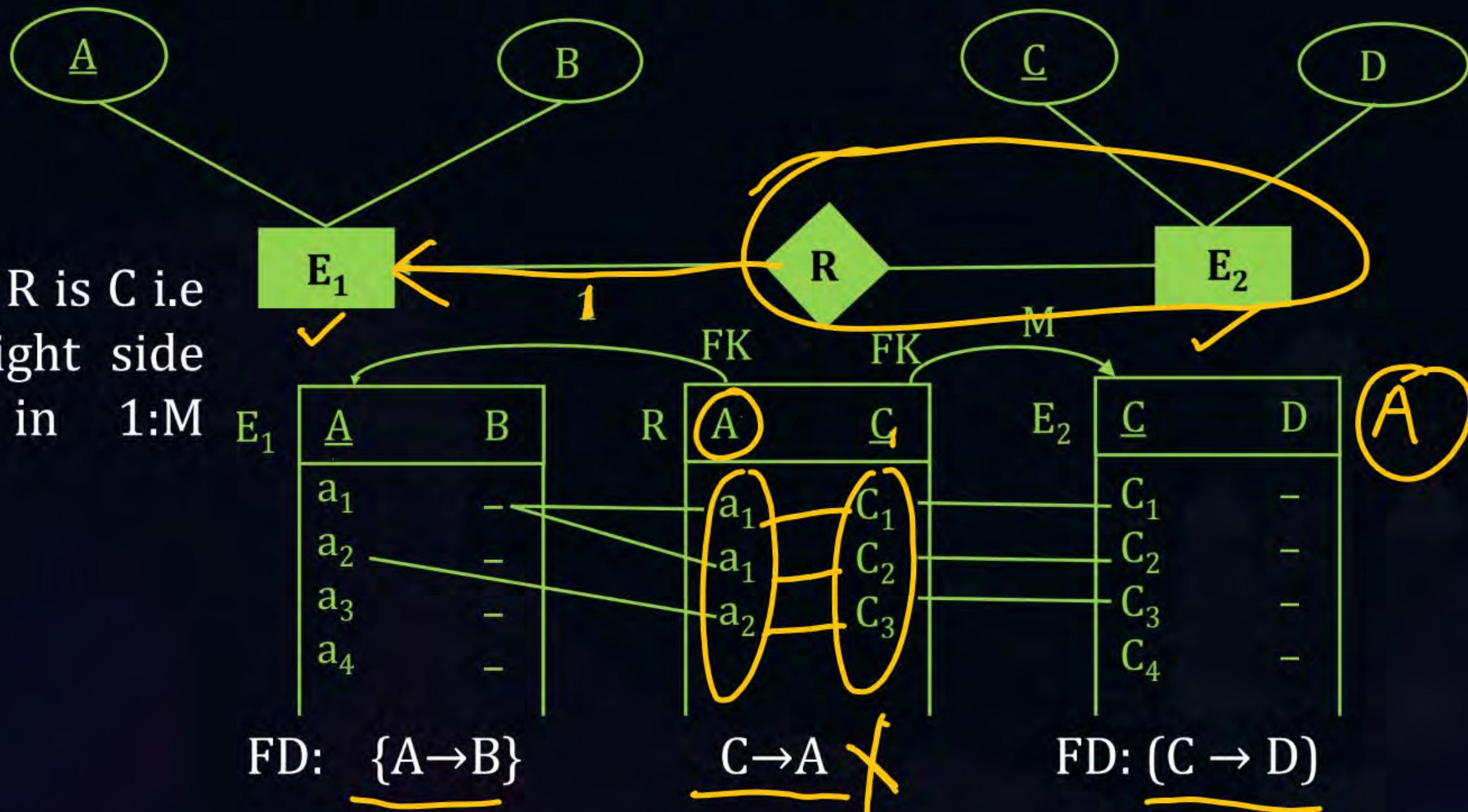




# Topic: One many mapping

## One to Many Relationship: Partial participation

So, the CK of R is C i.e  
the CK of right side  
entity set in 1:M  
mapping





# Topic: RDBMS Design

## One to Many Relationship: Partial participation

### RDBMS Design:

C is CK for R, so R and E<sub>2</sub> can be merged

E <sub>1</sub>	A	B
a <sub>1</sub>	-	
a <sub>2</sub>	-	
a <sub>3</sub>	-	
a <sub>4</sub>	-	

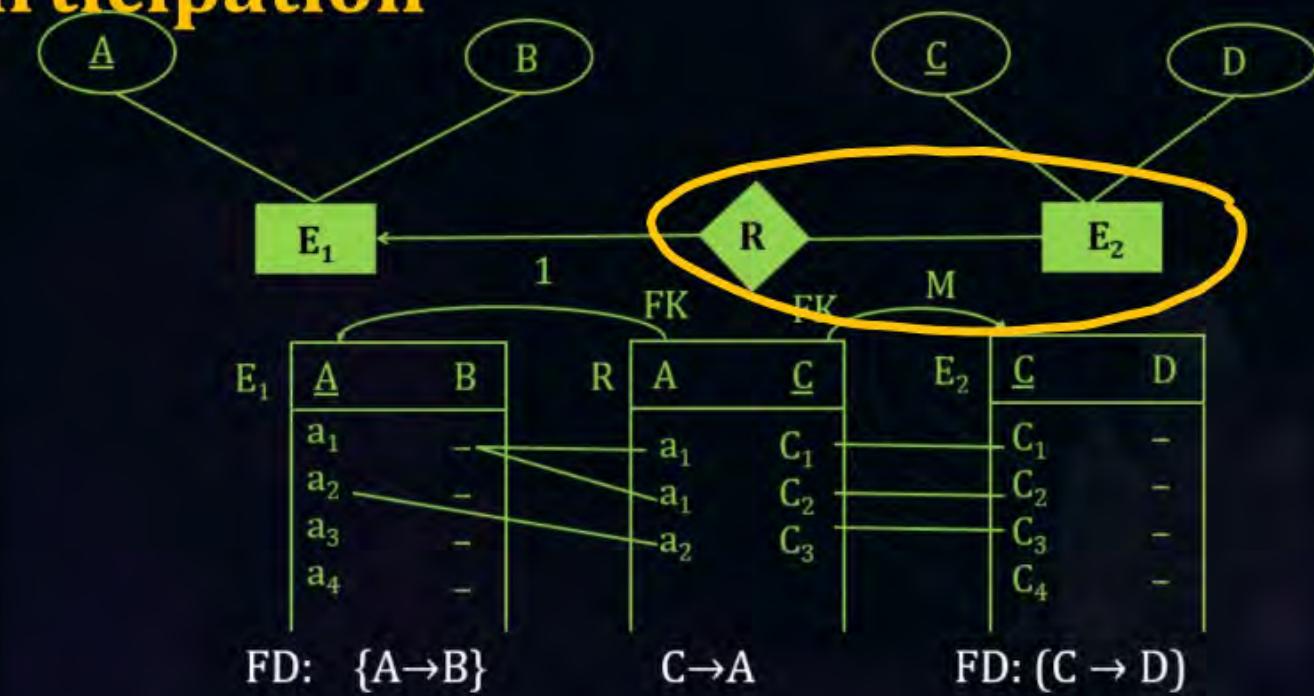
{A → B}

E <sub>2</sub> R	C	D	A
C <sub>1</sub>	-	a <sub>1</sub>	
C <sub>2</sub>	-	a <sub>1</sub>	
C <sub>3</sub>	-	a <sub>2</sub>	
C <sub>4</sub>	-	Null	

{C → DA}

Min 2 RDMS tables are required and 1 FK





## Topic: RDBMS Design

### One to Many Relationship: Total participation

Ex: EMP & Dept Entity Set, Manages relationship set is 1:M.

Each emp can manage many dept. Each dept **must** be managed by one emp



## Topic: RDBMS Design



### One to Many Relationship: Total participation

Ex: EMP & Dept Entity Set, Manages relationship set is 1:M.

Each emp can manage many dept. Each dept **must** be managed by one emp

#### Draw ERD:

We have 2 entities: Emp and Dept. Manages is establishing the relation.  
Emp to Dept is One to many mapping



## Topic: RDBMS Design



### One to Many Relationship: Total participation

Ex: EMP & Dept Entity Set, Manages relationship set is 1:M.

Each emp can manage many dept. Each dept **must** be managed by one emp

#### Draw ERD:

We have 2 entities: Emp and Dept. Manages is establishing the relation.  
Emp to Dept is One to many mapping





# Topic: RDBMS Design



## One to Many Relationship: Total participation

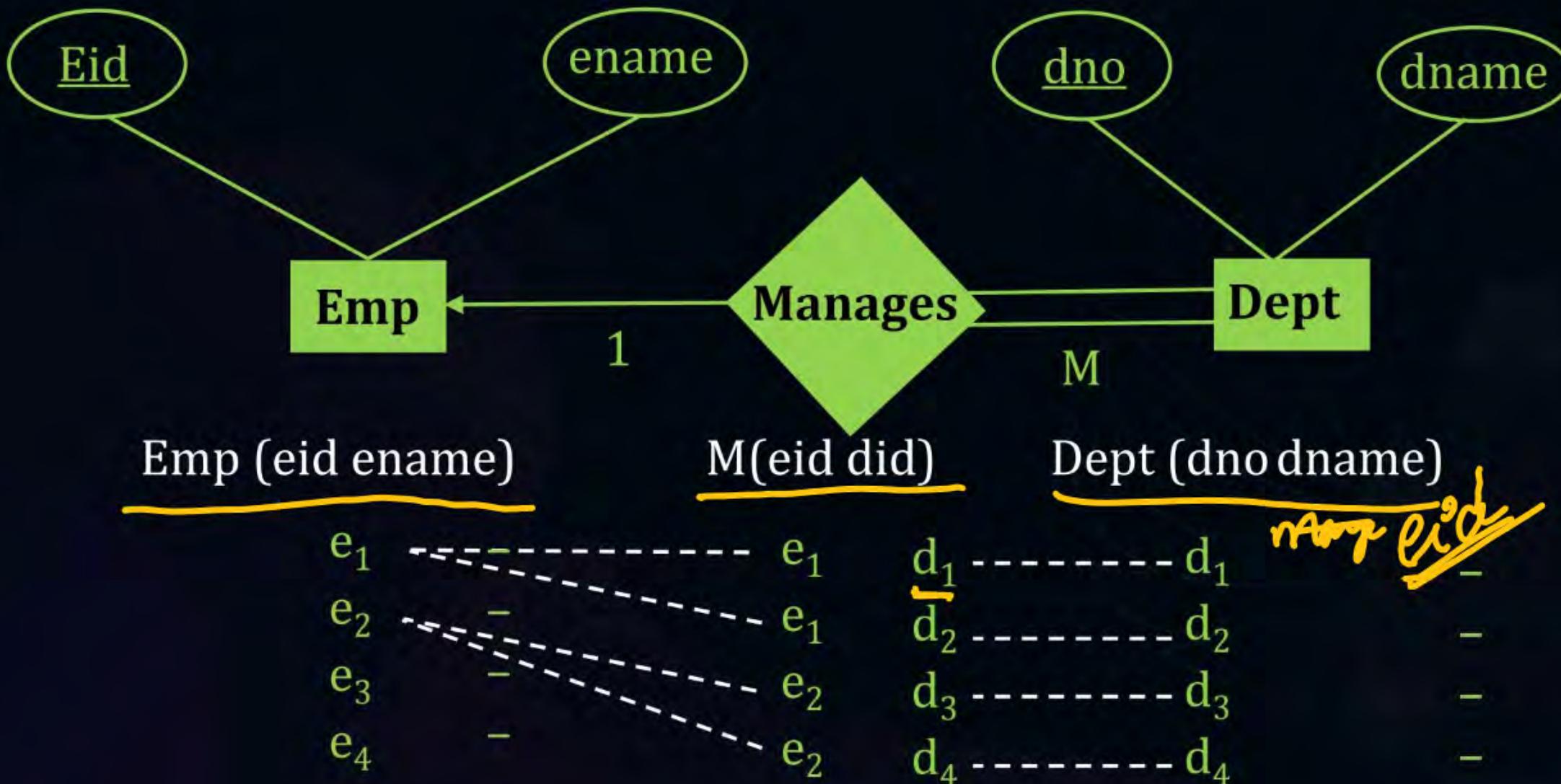




# Topic: RDBMS Design



## One to Many Relationship: Total participation

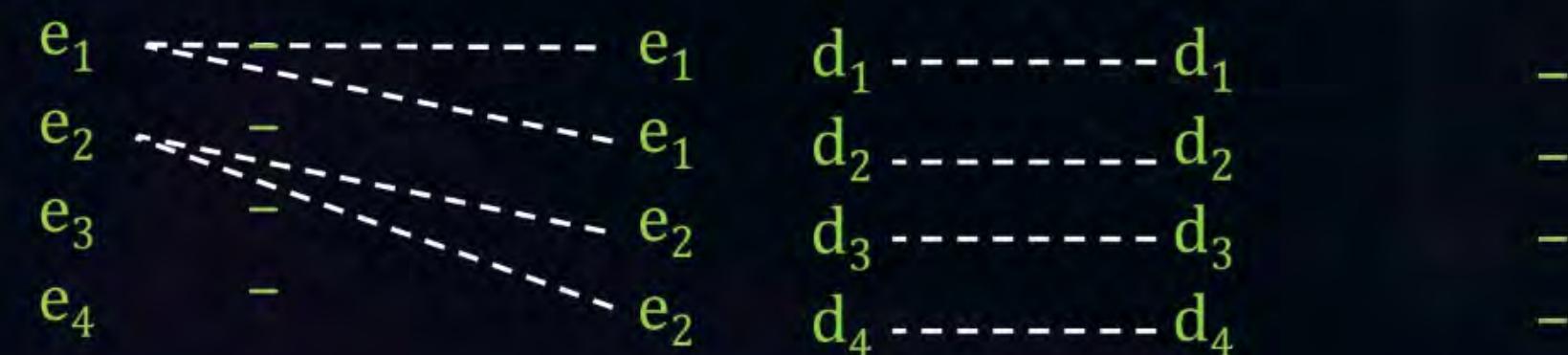
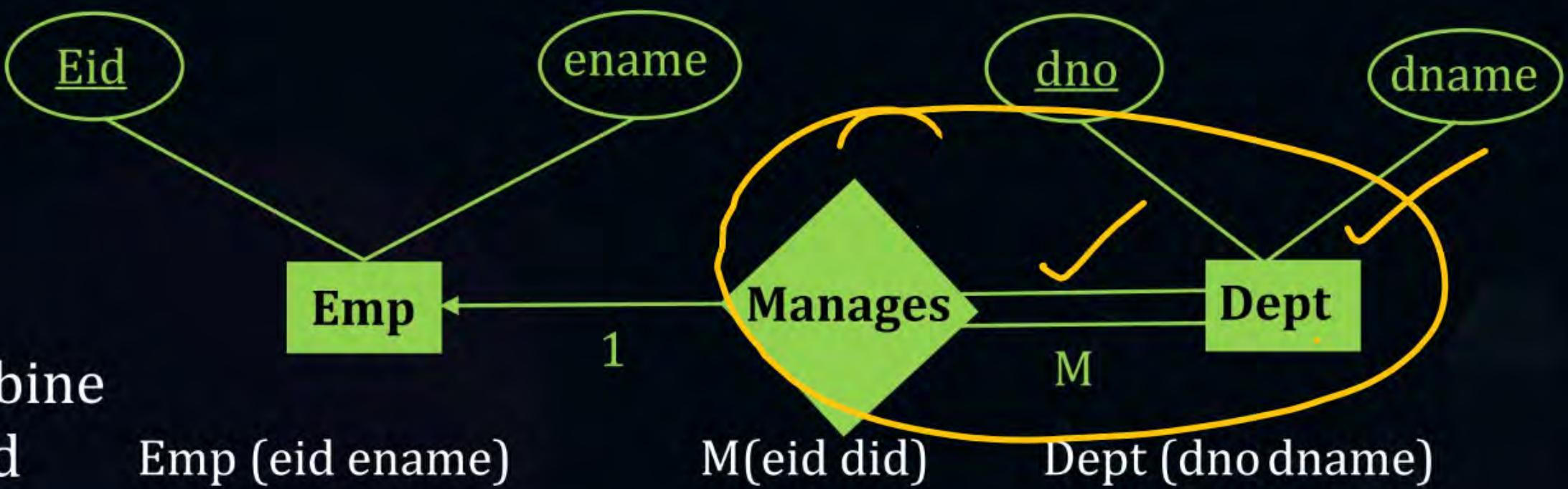




# Topic: RDBMS Design

## One to Many Relationship: Total participation

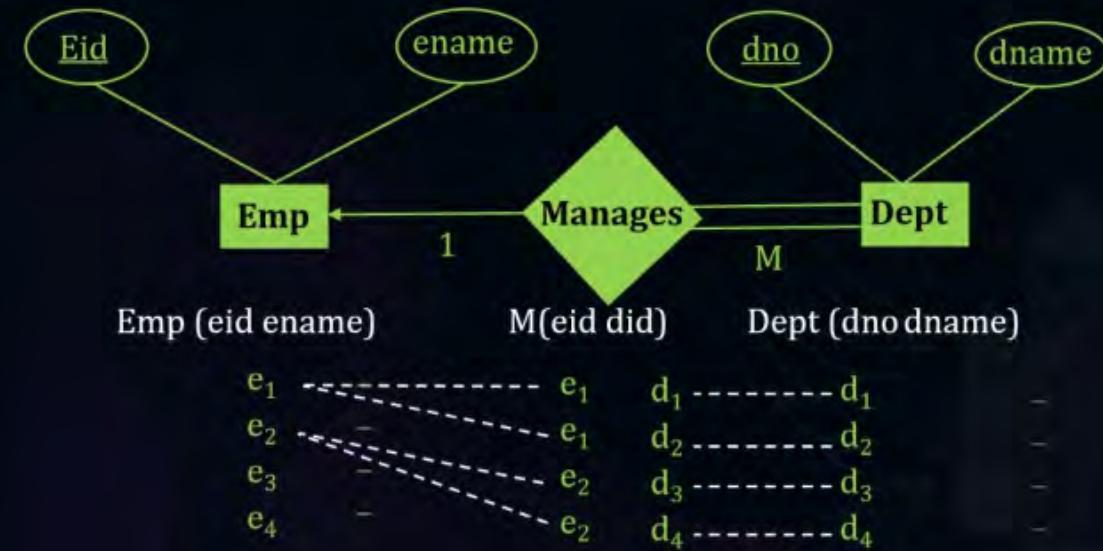
Also note we will have **Total participation** from Manages to Dept.





# Topic: RDBMS Design

**RDBMS design :** Very similar to me previous table.



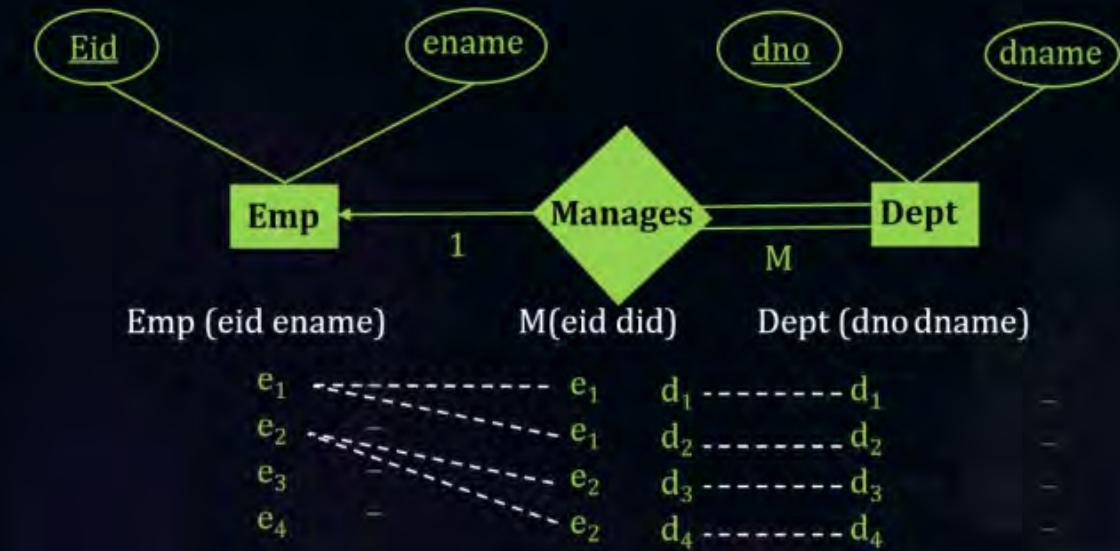


# Topic: RDBMS Design



**RDBMS design :** Very similar to me previous table.

Emp	Dept_M				
	eid	ename	did	dname	eid
e <sub>1</sub>			d <sub>1</sub>		e <sub>1</sub>
e <sub>2</sub>	-		d <sub>2</sub>	-	e <sub>2</sub>
e <sub>3</sub>	-		d <sub>3</sub>	-	e <sub>3</sub>
e <sub>4</sub>	-		d <sub>4</sub>	-	e <sub>4</sub>
					-



Note that in previous example, we also had a NULL entry, that we will not find here because of total participation. Min 2 RDBMS tables & 1 Foreign key.



## Topic: RDBMS Design



If we want to design this RDBMS table:

Create table emp ✓  
(eid varchar(10) Primary key ,  
ename varchar(20) );

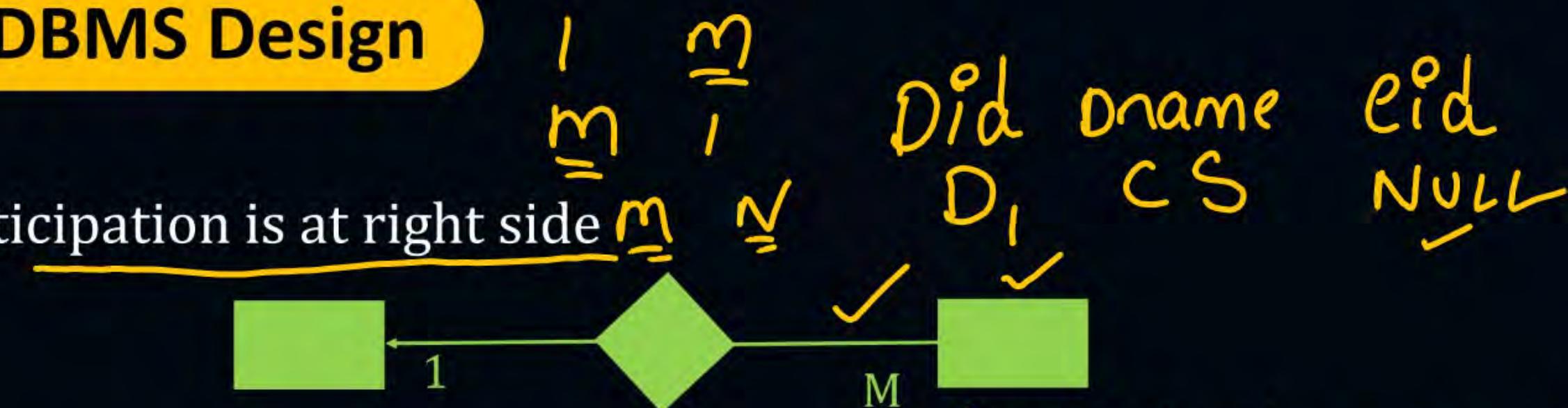
Create table DeptM  
(did varchar(10) PK,  
dname varchar(30),  
eid varchar(10) NOT NULL ✓  
Foreign key (eid) references emp);

Emp		Dept_M		
		did	dname	eid
	e <sub>1</sub>	d <sub>1</sub>	-	e <sub>1</sub>
	e <sub>2</sub>	d <sub>2</sub>	-	e <sub>2</sub>
	e <sub>3</sub>	d <sub>3</sub>	-	e <sub>3</sub>
	e <sub>4</sub>	d <sub>4</sub>	-	e <sub>4</sub>



## Topic: RDBMS Design

- If partial participation is at right side



then Foreign Key that we added into right side entity Set allows NULL.

- If we have total participation on right side of one-to-many mapping then FK that we added in right side entity should NOT contain NULL values  
∴ we took NON NULL



# Topic: Many to Many Mapping

## Many to Many Relationship:



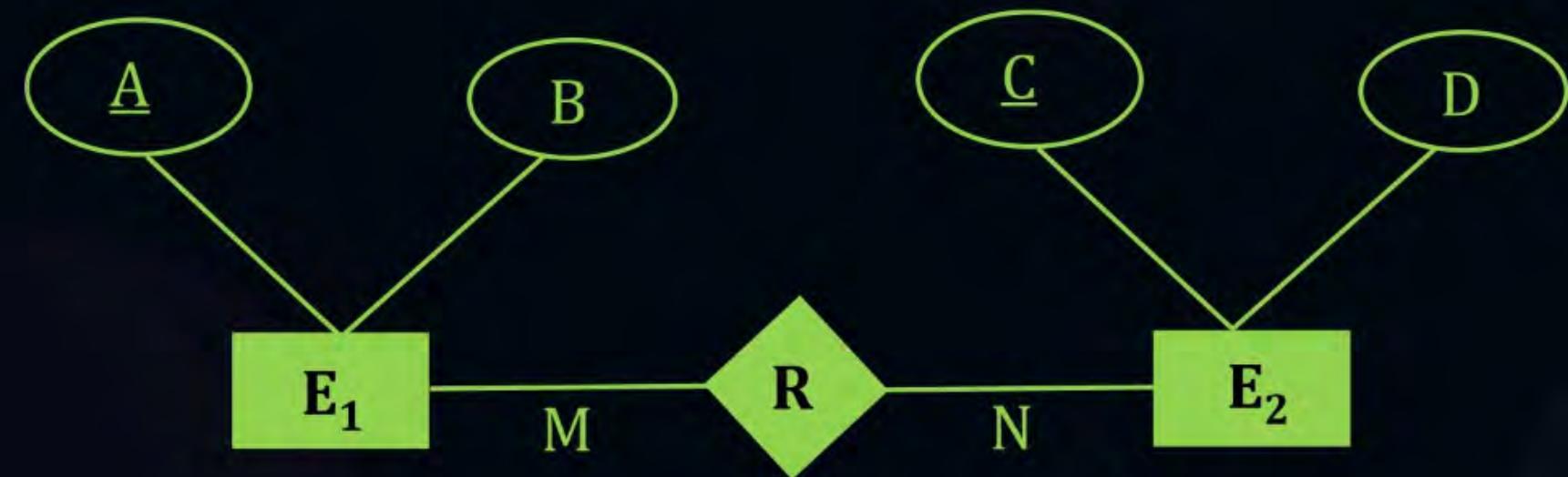
Each object of  $E_1$  can relate to many object entity object of  $E_2$ .

Similarly, each object of  $E_2$  can relate to many object of  $E_1$



# Topic: Many to Many Mapping

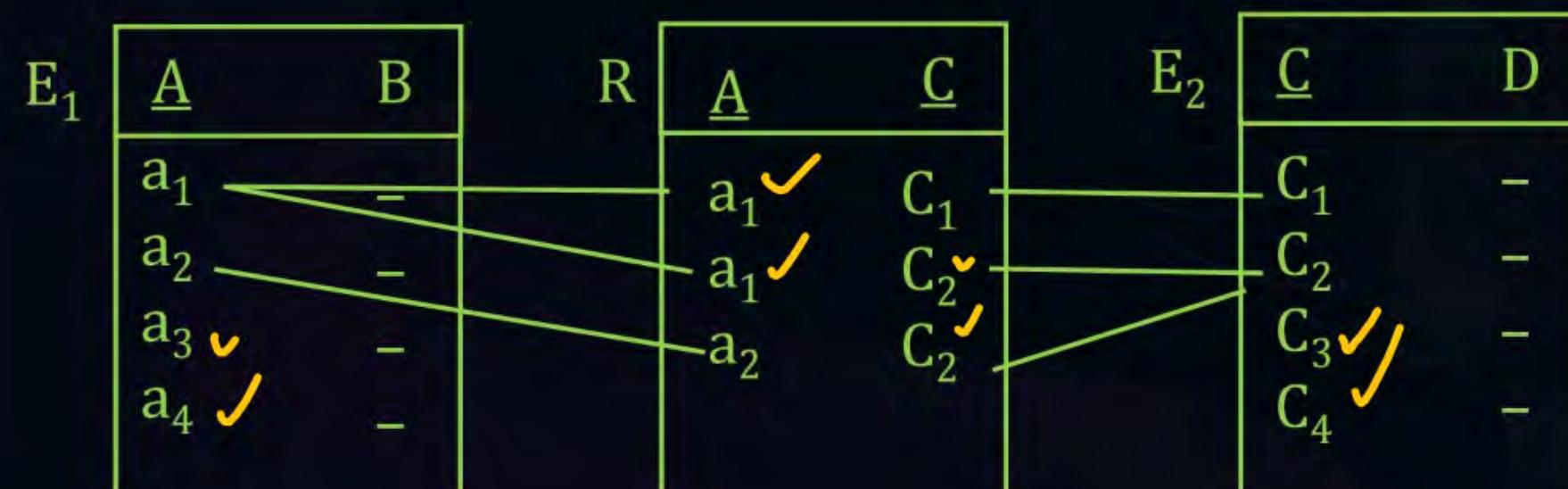
## Many to Many Relationship:





# Topic: Many to Many Mapping

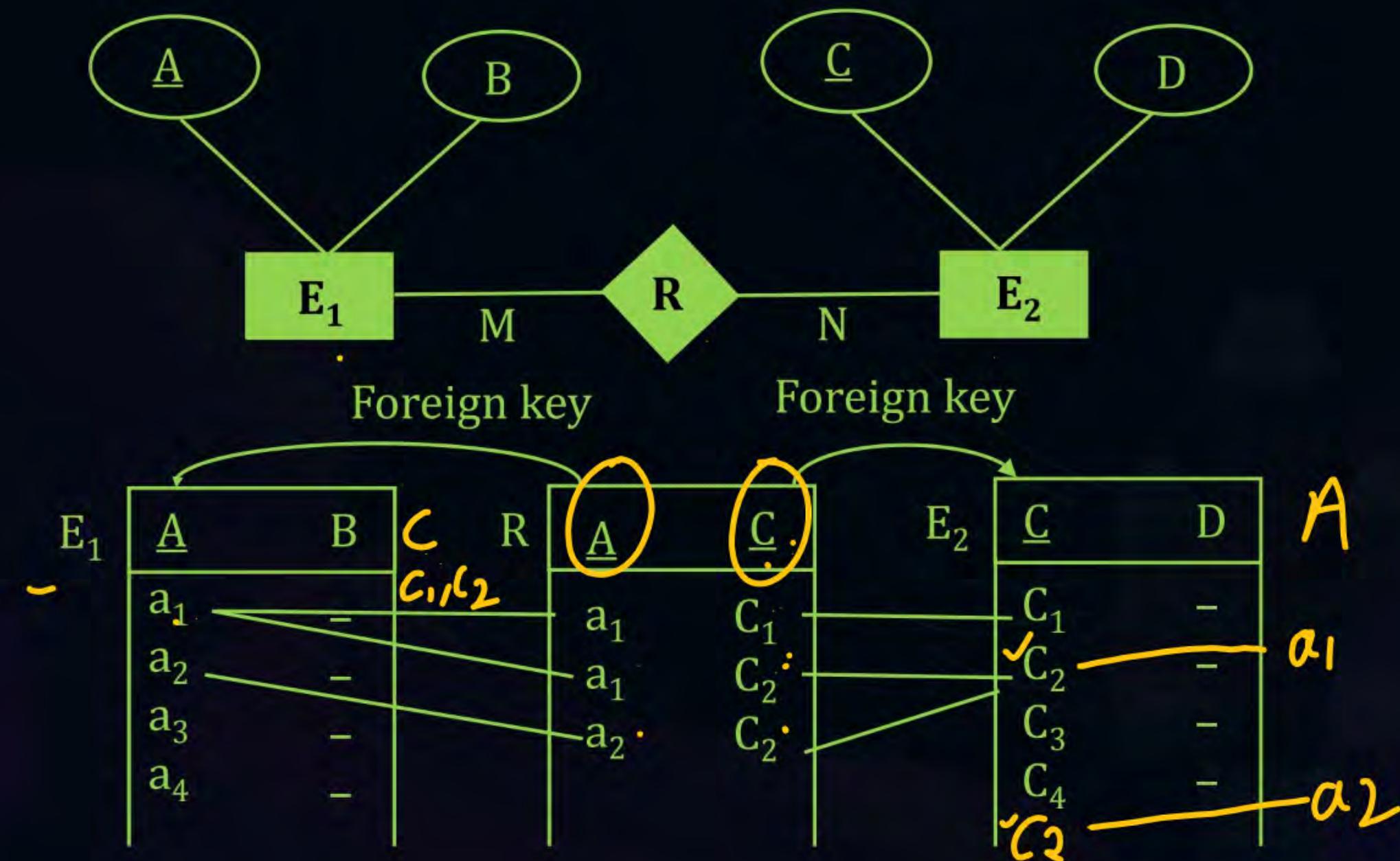
## Many to Many Relationship:





# Topic: Many to Many Mapping

## Many to Many Relationship:

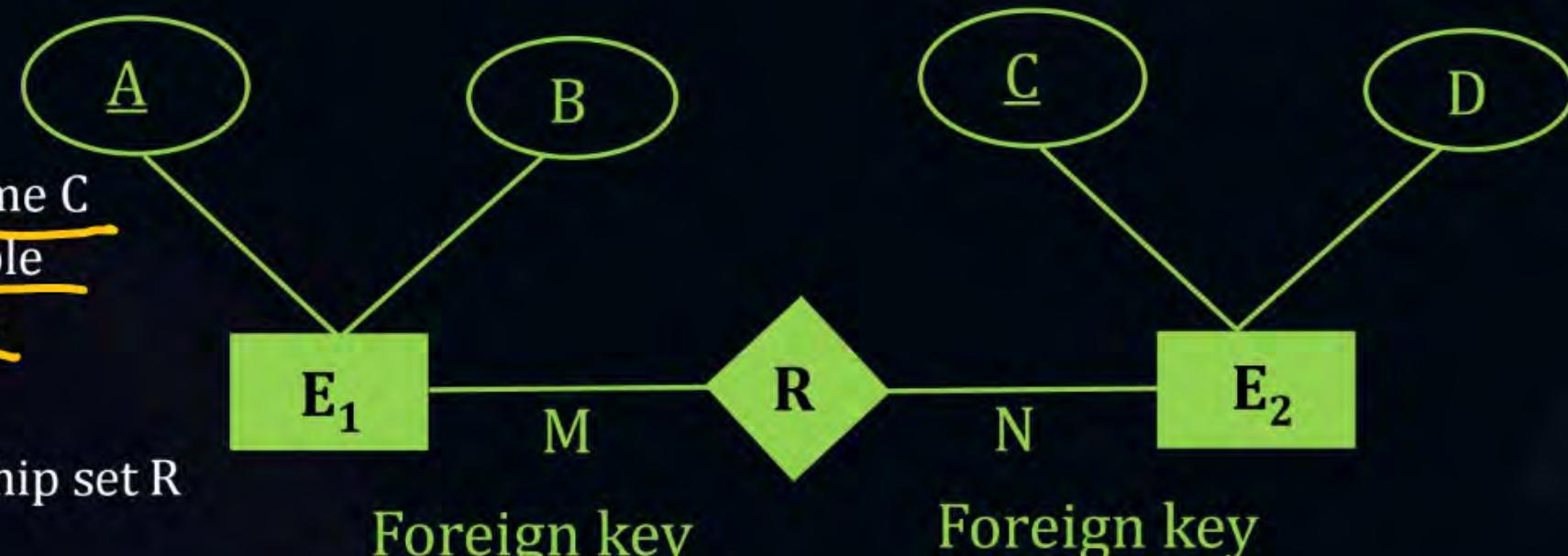




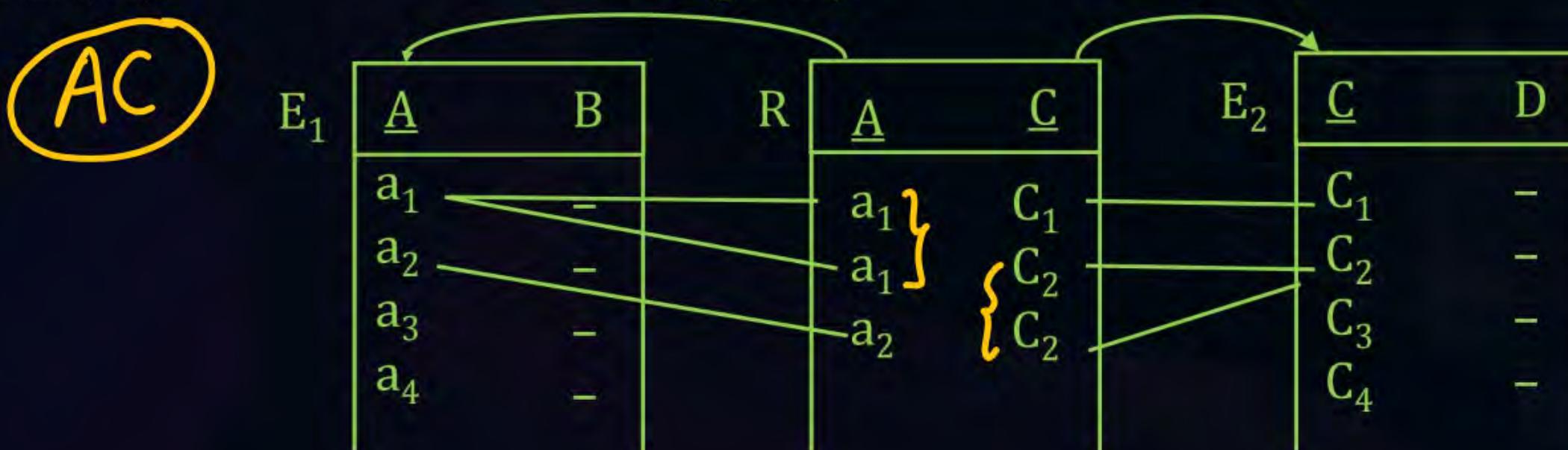
# Topic: Many to Many Mapping

## Many to Many Relationship:

Two or more records can have Same C values in Relationship, also multiple records can have same A values in relationship.



Hence, Candidate Key of relationship set R is, the combination of (AC).





# Topic: Many to Many Mapping

- R (A, C) {AC} is CK for M:N

∴ Some entity of E<sub>1</sub> can relate to many entities of E<sub>2</sub> and vice versa.

The A in Relationship Table is not unique

∴ We can't merge E<sub>2</sub> Table and Relationship table.

Similarly, C in Relationship table is not unique

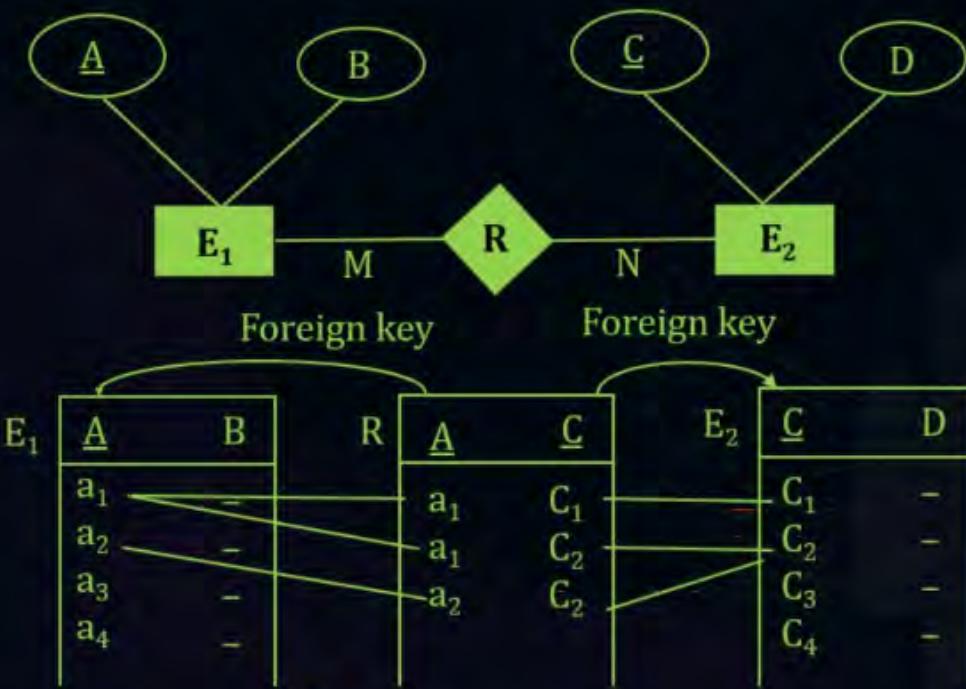
∴ We can't merge E<sub>1</sub> Table and Relationship table.

So, In RDBMS Table, minimum three tables are required and 2 FK





# Topic: Many to Many Mapping



[E<sub>1</sub>(A B), E<sub>2</sub>(C D), R(A C)]

A → B

C → D

No non-trivial key.

- AC was the candidate key for relation hence, we get no non-trivial FD for relationship.

Minimum 3 tables and 2FKs required



## Topic: One-One Mapping

(a) 1:1 mapping between E<sub>1</sub> and E<sub>2</sub> with partial participation on both ends

Each object of E<sub>1</sub> is mapped to atmost one object of E<sub>2</sub> and vice versa.



## Topic: One-One Mapping

(a) 1:1 mapping between  $E_1$  and  $E_2$  with partial participation on both ends

Each object of  $E_1$  is mapped to atmost one object of  $E_2$  and vice versa.

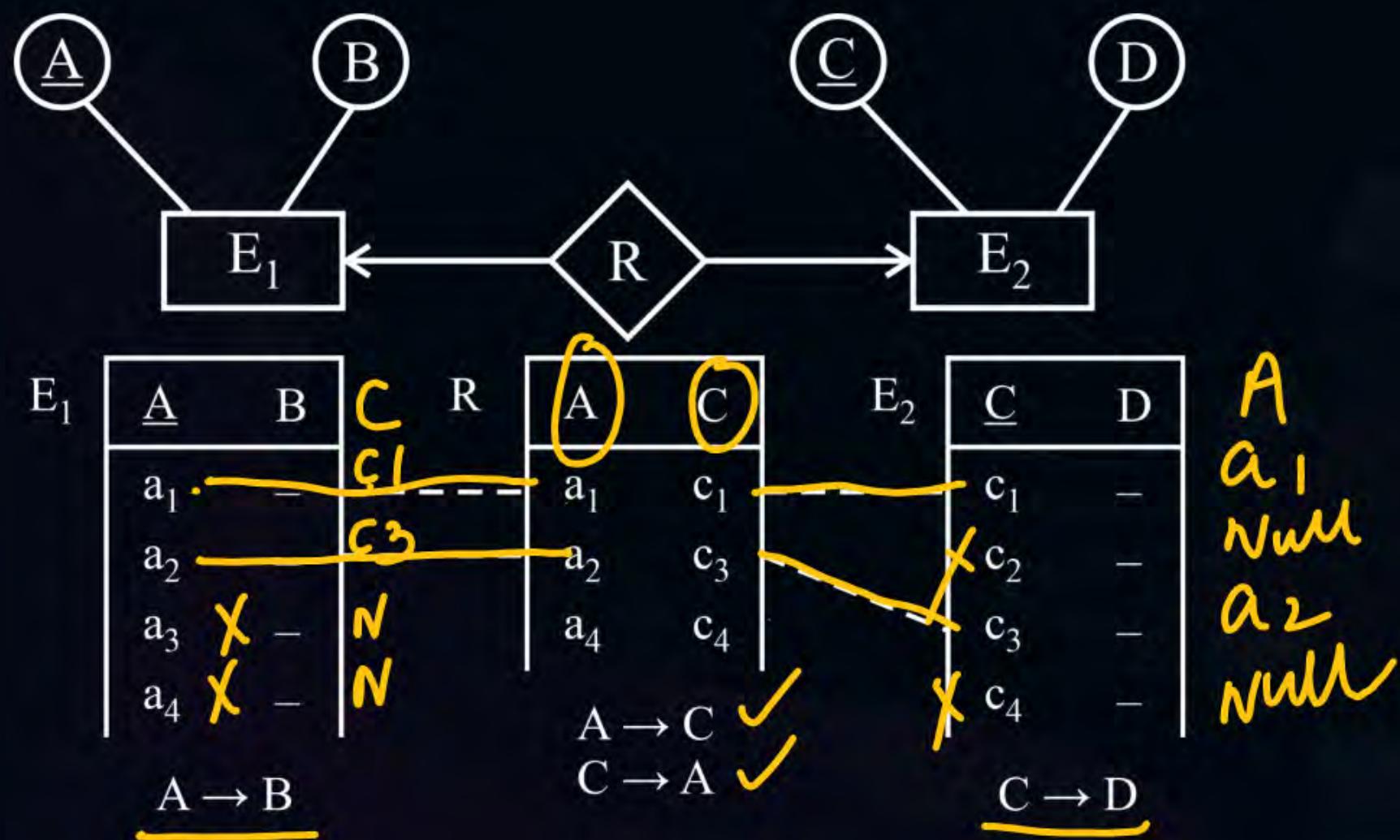




## Topic: One-One Mapping

(a) 1:1 mapping between  $E_1$  and  $E_2$  with partial participation on both ends

Each object of  $E_1$  is mapped to atmost one object of  $E_2$  and vice versa.





## Topic: One-One Mapping

(a) 1:1 mapping between  $E_1$  and  $E_2$  with partial participation on both ends

Each object of  $E_1$  is mapped to atmost one object of  $E_2$  and vice versa.

Since we have partial participation,  
some of the objects may not have any  
participation at all as we can see that  
from the table.

What should be the key?

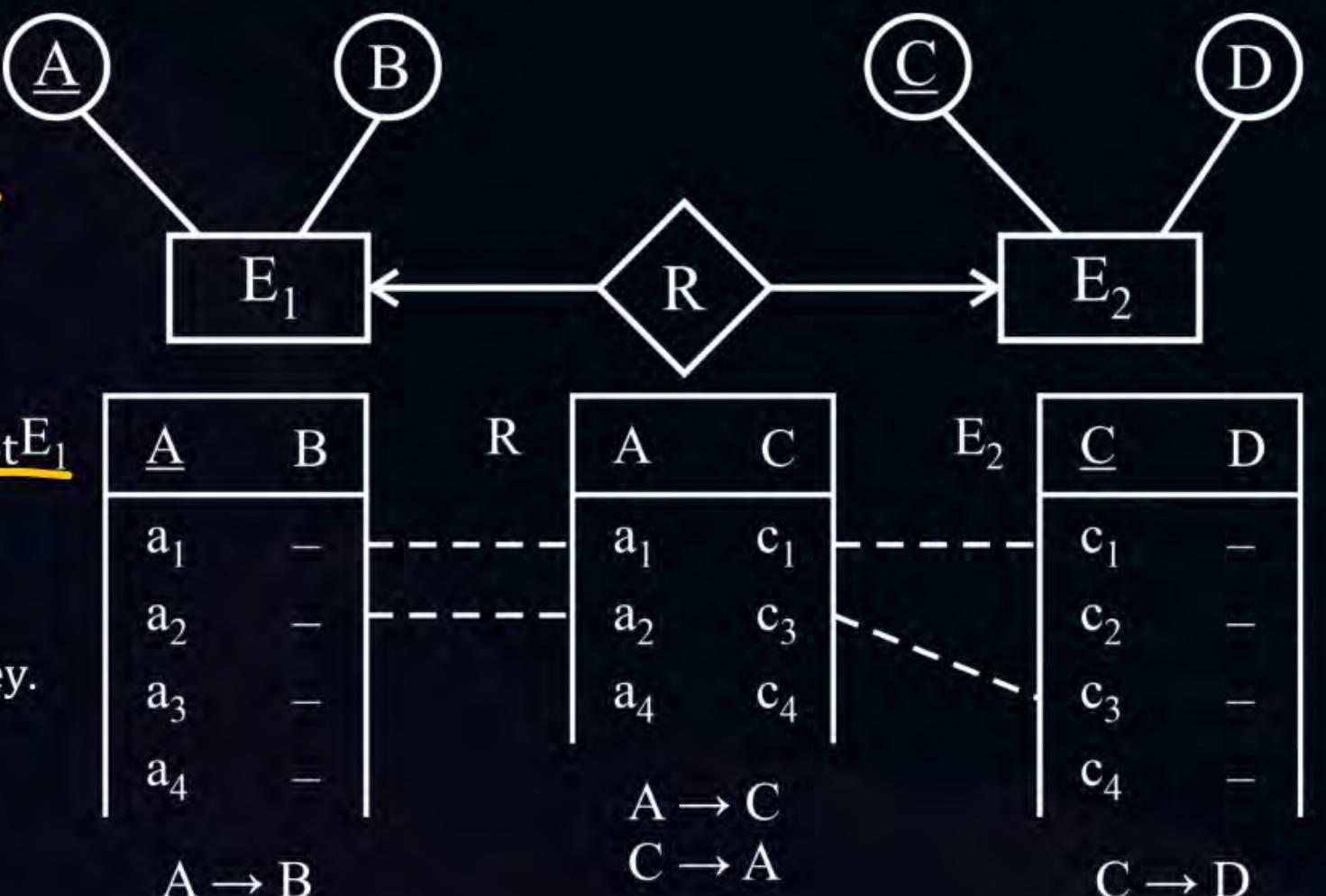
In R table, no two records can have  
same A, because one entity of  $E_1$  is not allowed to relate to more than one entity of  $E_2$ .

Hence, A should be key, but same is also true for C, hence C will also be key.

Candidate key for R is both {A, C}

$A \rightarrow C \ \& \ C \rightarrow A$

Candidate key of R (A, C) {A, C}



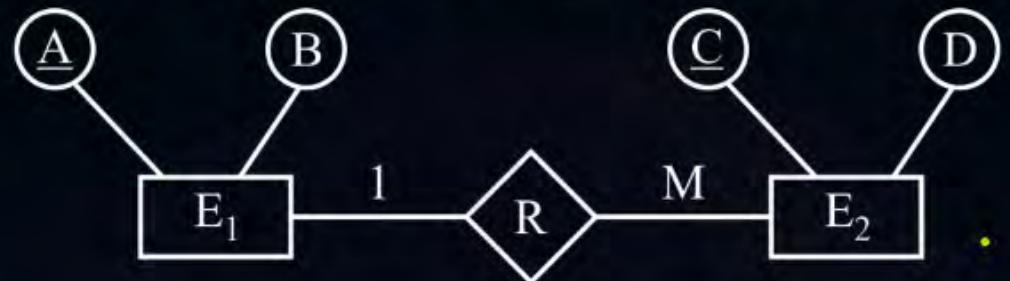


# Topic: Candidate Key

**What is the CK of different Relationships?**

**Candidate key of one-to-many relationship?**

Key of many side entity set, then key of  $E_2$ .



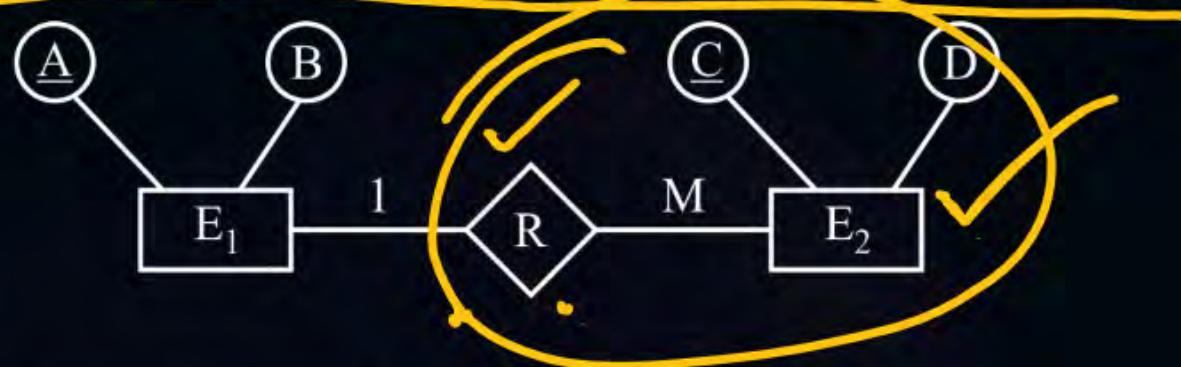


# Topic: Candidate Key

**What is the CK of different Relationships?**

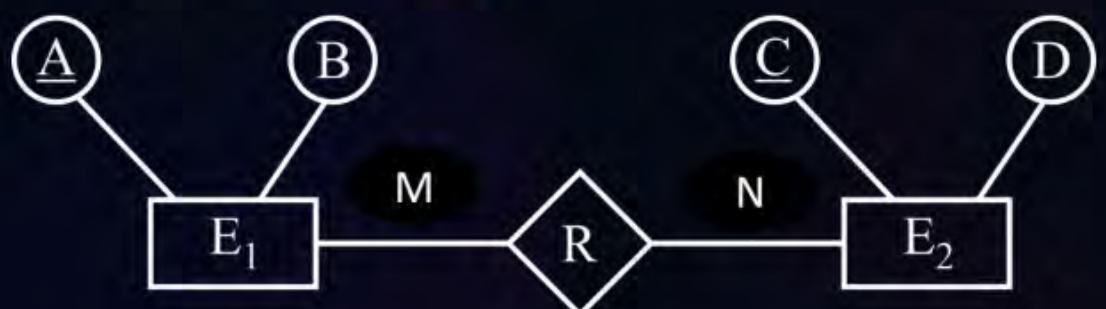
**Candidate key of one-to-many relationship?**

Key of many side entity set, then key of E<sub>2</sub>.



**Candidate key for Many-to-Many?**

Combination of {AC}



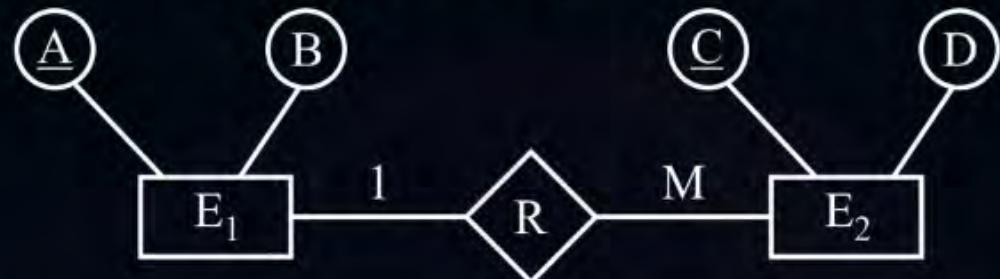


# Topic: Candidate Key

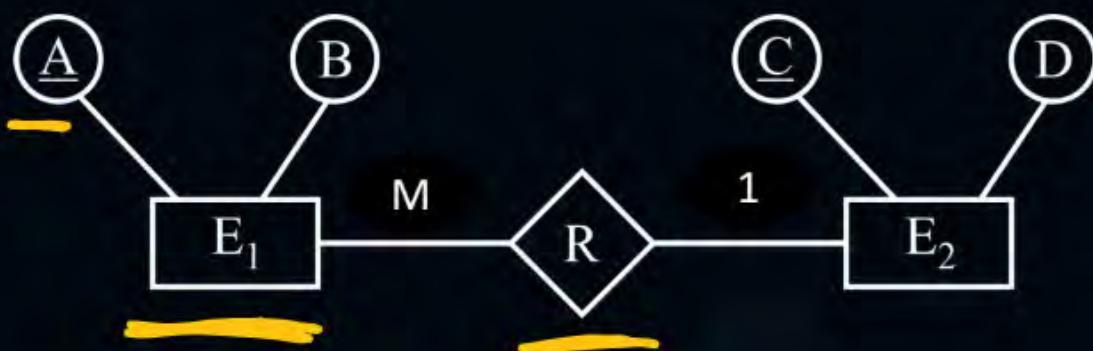
**What is the CK of different Relationships?**

**Candidate key of one-to-many relationship?**

Key of many side entity set, then key of  $E_2$ .

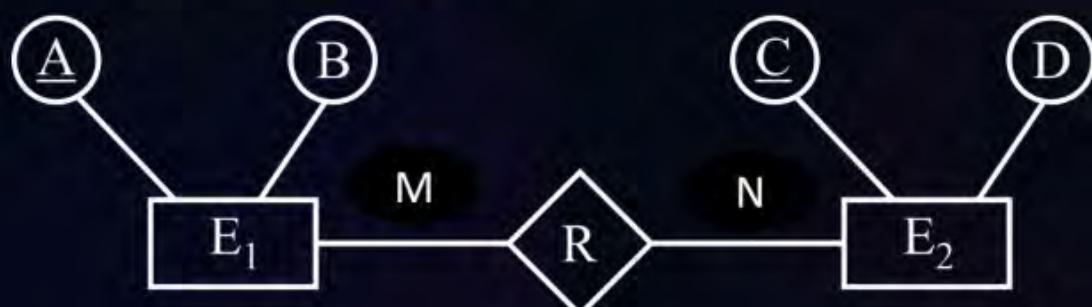


**Candidate key for Many-to-one** • CK of Many



**Candidate key for Many-to-Many?**

Combination of {AC}



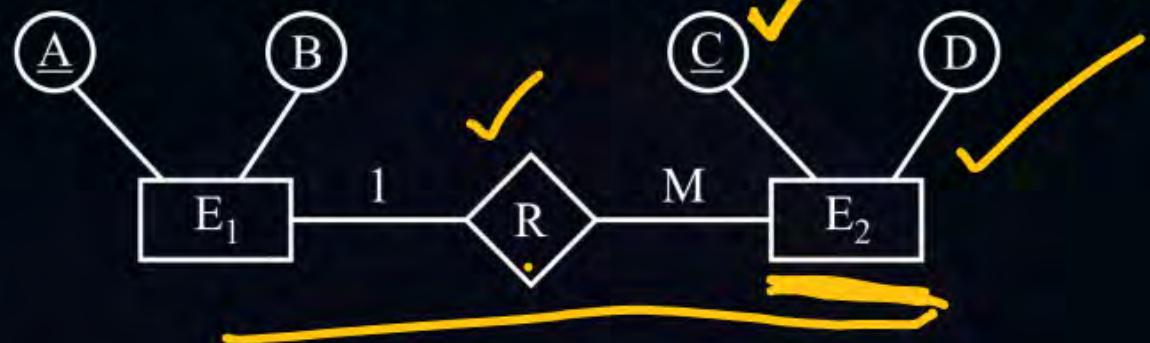


# Topic: Candidate Key

What is the CK of different Relationships?

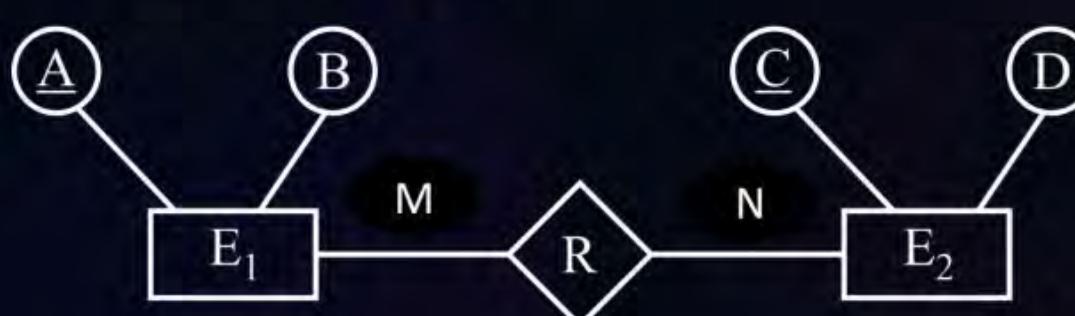
Candidate key of one-to-many relationship?

Key of many side entity set, then key of  $E_2$ .

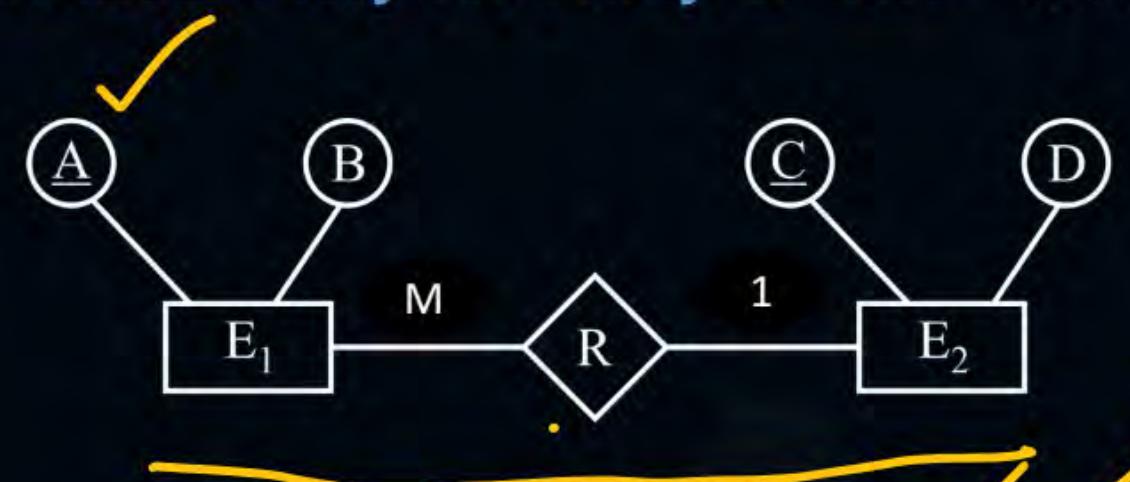


Candidate key for Many-to-Many?

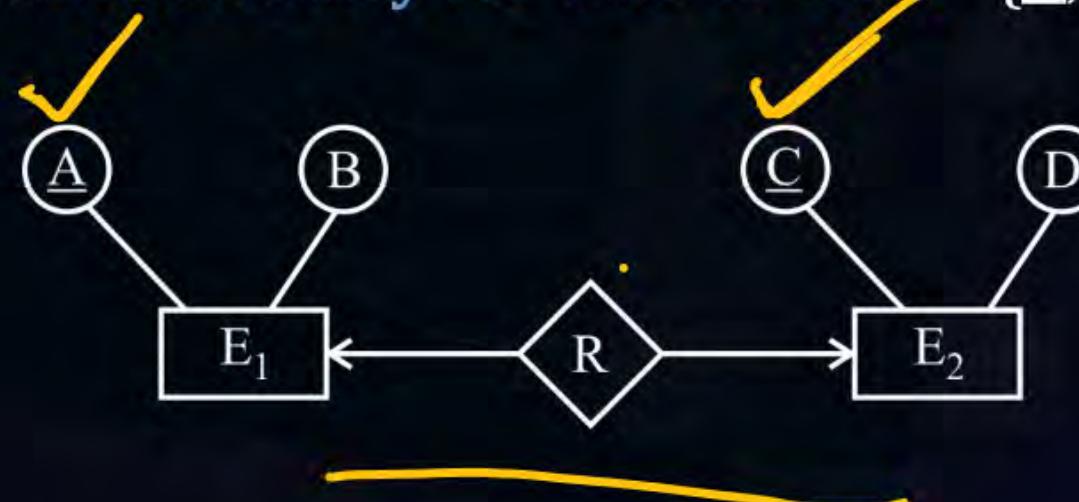
Combination of {AC}



Candidate key for Many-to-one • CK of Many



Candidate key for One-to-One? {A, C}





## Topic: One-One Mapping

### Candidate key for One-to-One relationship?

- $\{\underline{A}, \underline{C}\}$



- For this diagram, minimum of 2 RDBMS table required.

There are 2 possibilities, (a) R and E<sub>1</sub> can be merged or

(b) R and E<sub>2</sub> can be merged



# Topic: One-One Mapping

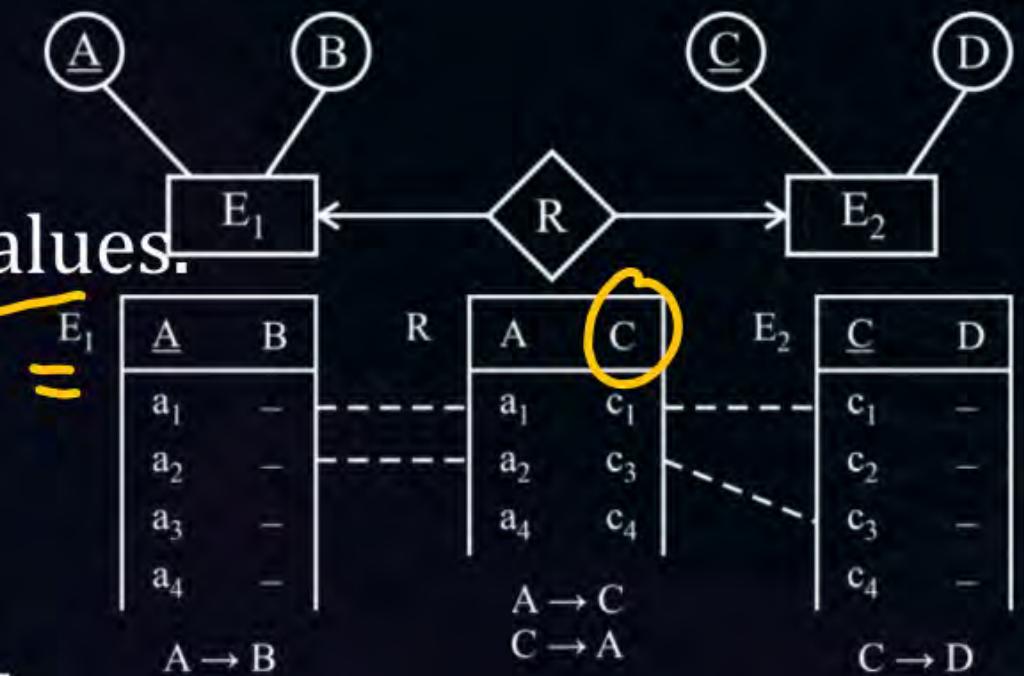
P  
W

(a) Lets combine  $E_1$  & R because both table have unique values.

CK of  $E_1 R$ : {A, C}

PK : A

Alternative key : C



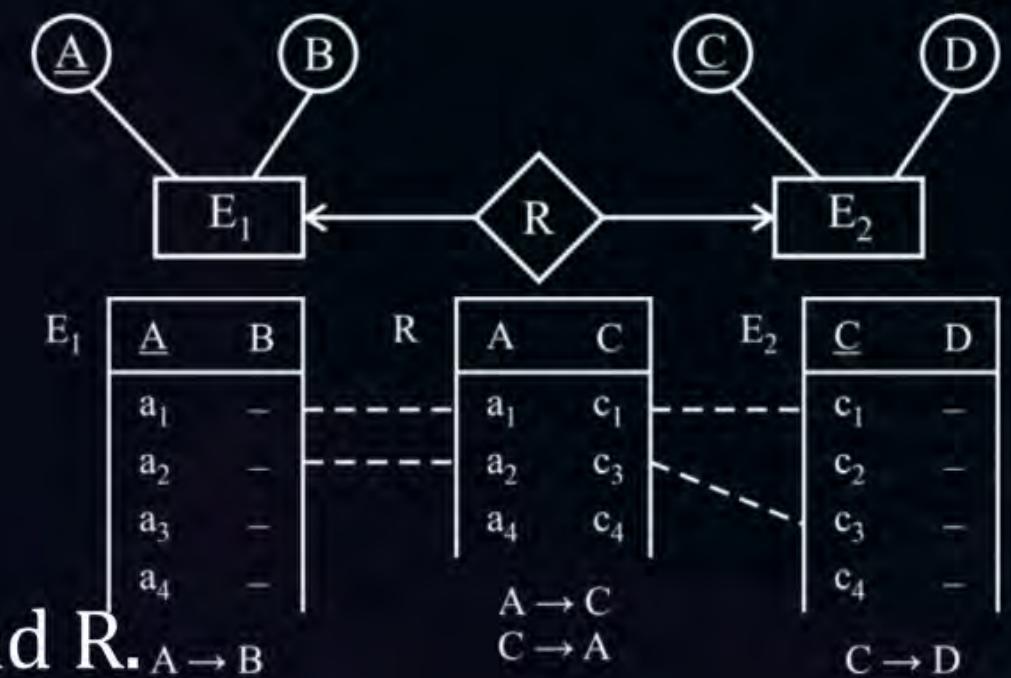


# Topic: One-One Mapping

P  
W

## Why we combined $E_1$ & R?

- The fundamental reason was, A was key for both  $E_1$  and R.
- Relationship  $E_2$  and R could also be combined since C is primary key in both.
- We can't combine all of them into a single table because both are partial participation. Hence, can't combine into single table.





# Topic: One-One Mapping

P  
W

Try combining all the data, what will happen?

- Say, we have

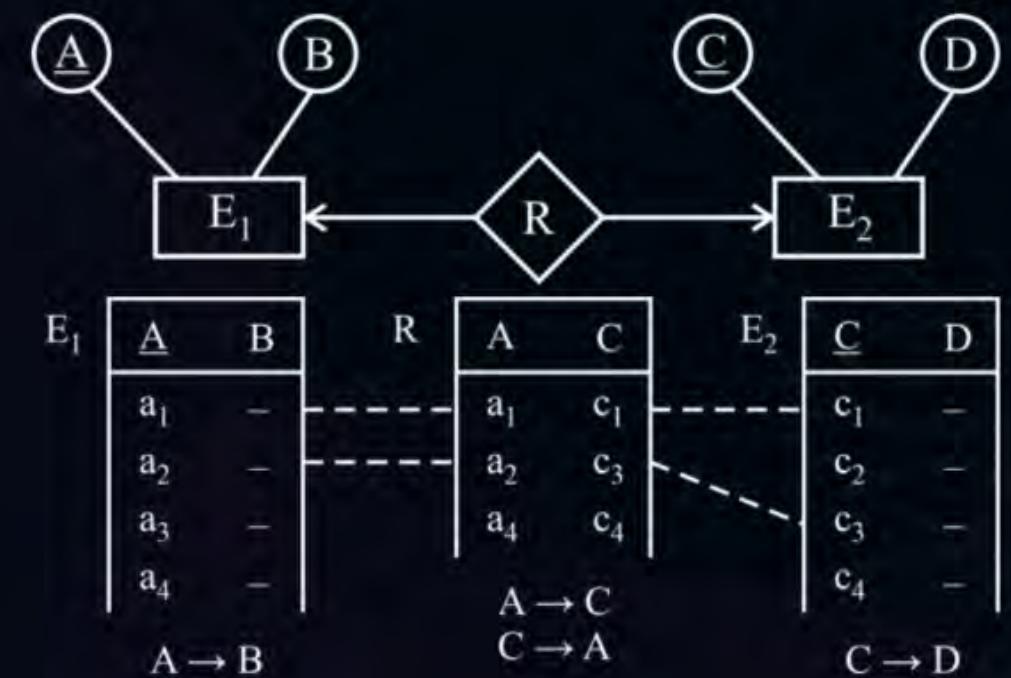
	A	B	C	D
E <sub>1</sub>	a <sub>1</sub>	-	c <sub>1</sub>	-
R	a <sub>2</sub>	-	c <sub>3</sub>	-
E <sub>2</sub>	a <sub>3</sub>	-	NULL	NULL
	a <sub>4</sub>	-	c <sub>4</sub>	-
	NULL	NULL	c <sub>2</sub>	-

- Assume the above,

When we try to combine, it might so happen that for some record in E<sub>1</sub>RE<sub>2</sub>

C can be null similarly, A can be NULL for some record

Therefore, No possibility to get primary key, hence it will become wrong design.





## Topic: One-One Mapping

(b) 1 : 1 mapping between  $E_1 E_2$  with

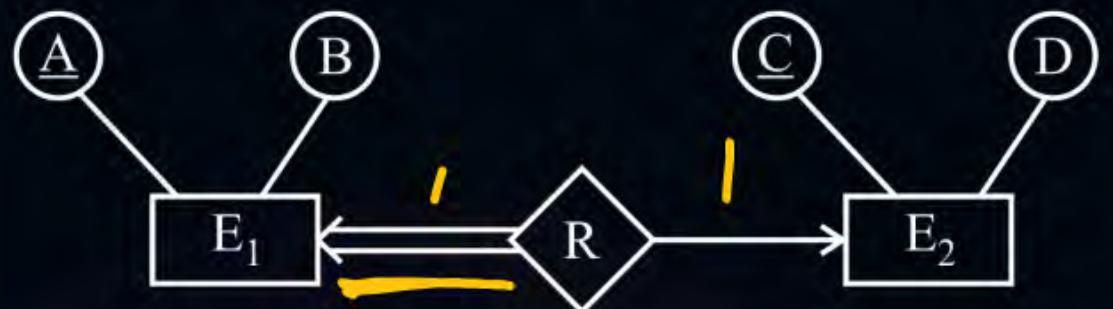
- (i) one end is total participation
- (ii) both ends are in total participation.



## Topic: One-One Mapping

- (i) One to one mapping with one end is total participation

Now here, because of total participation, we can combine  $E_1$  R  $E_2$  into single table.



- If we can combine entity together no need for foreign key.
- For every A of entity  $E_1$ , there is a mapping of C.

A	B	C	D
a <sub>1</sub>	—	NotNULL	

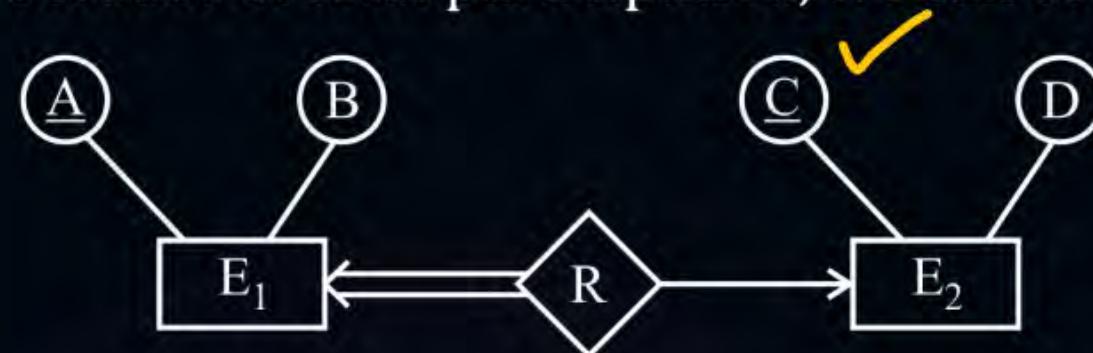
$A \rightarrow BC$      $C \rightarrow AD$      $\{\underline{A}, \underline{C}\}$  are CKs



## Topic: One-One Mapping

- (i) One to one mapping with one end is total participation

Now here, because of total participation, we can combine  $E_1$  R  $E_2$  into single table.



- If we can combine entity together no need for foreign key.
- For every A of entity  $E_1$ , there is a mapping of C.

$E_1$ R $E_2$	A	B	C	D
			NULL	
	a <sub>1</sub>	—	NotNULL	
.				

$A \rightarrow BC$      $C \rightarrow AD$      $\{\underline{A}, \underline{C}\}$  are CKs

A will have all unique values and no NULL values, C will have all Unique values but there will be NULL values

A is PK and C is AK

Minimum 1 RDBMS table and no foreign key.



## Topic: One-One Mapping

- (b) 1 : 1 mapping with both ends are in total participation.

Now here, because of total participation, we can combine  $E_1 \text{ R } E_2$  into single table.



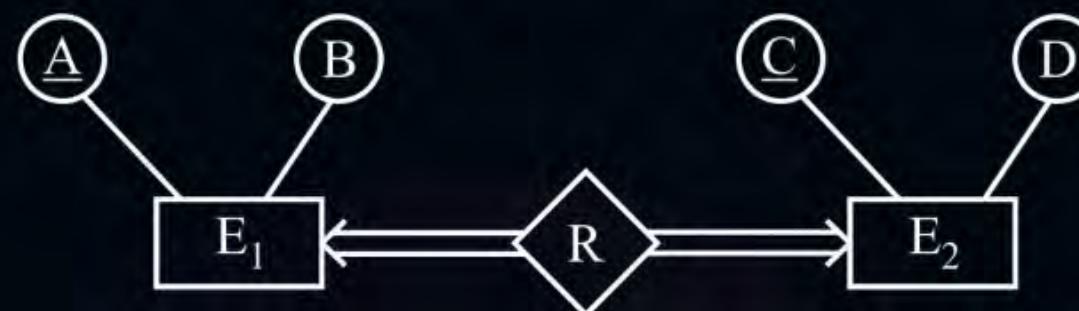
For each object of  $E_1$ , there is exact one object mapped in  $E_2$  and vice versa.



## Topic: One-One Mapping

- (b) 1 : 1 mapping with both ends are in total participation.

Now here, because of total participation, we can combine  $E_1 \text{ R } E_2$  into single table.



For each object of  $E_1$ , there is exact one object mapped in  $E_2$  and vice versa.

Hence, there is no chance of getting NULL for A or C

E <sub>1</sub> R E <sub>2</sub>			
A	B	C	D
A <sub>1</sub>		C <sub>1</sub>	

{ A , C }  
↑ Pk      ↑ Unique  
              Not Null

FD's will not change.

$$A \rightarrow BC \quad C \rightarrow AD$$

So, if there is total participation in either side, that is either in  $E_1$  or  $E_2$ , then we will have one table.

Minimum 1 RDBMS table and no foreign key.



# Topic: Conclusion

For One-to-Many mapping:

Minimum 2 table and 1FK

for both partial and total participation



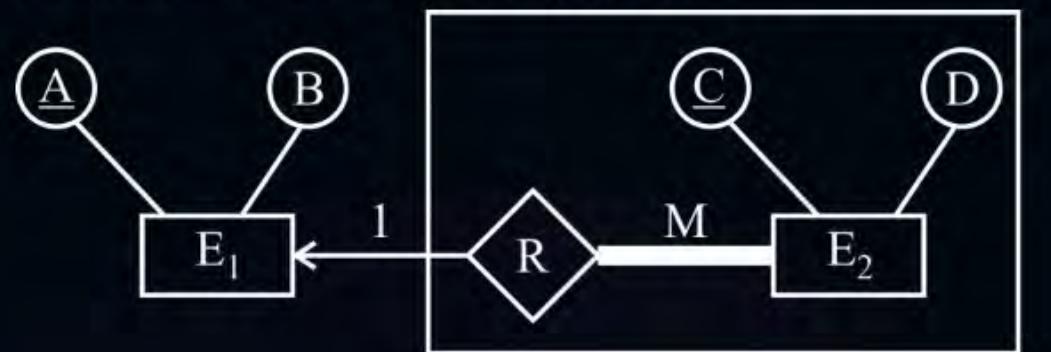


# Topic: Conclusion

**For One-to-Many mapping:**

Minimum 2 tables and 1FK

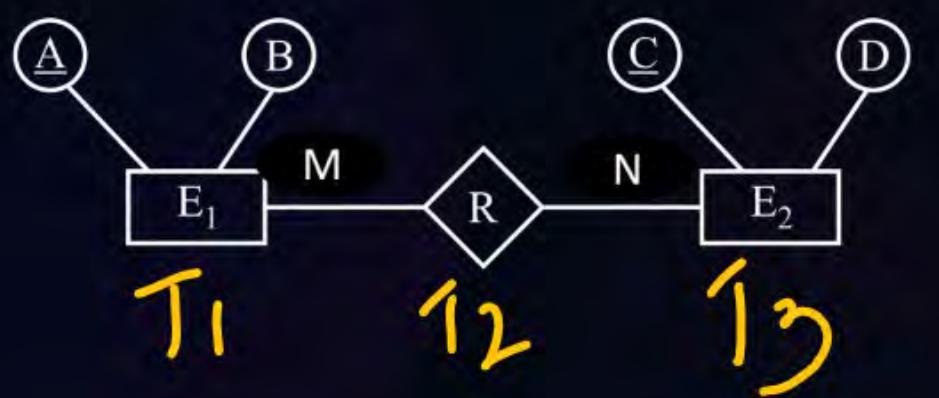
for both partial and total participation



**For Many-to-Many mapping:**

Key of R is {AC}

3 tables and 2FKs are required





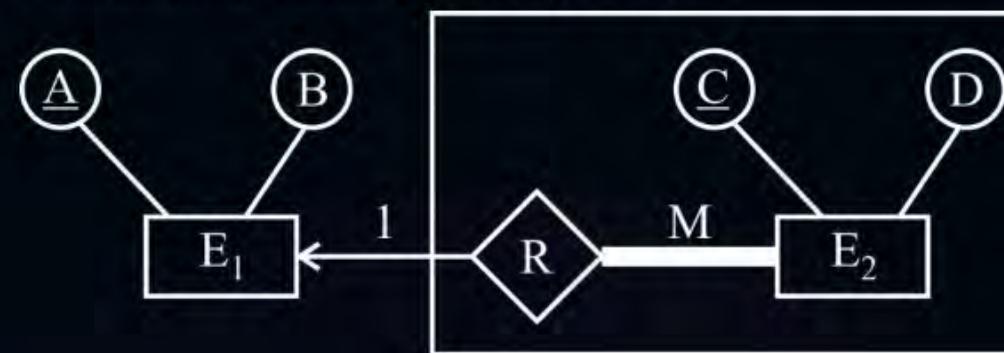
# Topic: Conclusion

**1 : 1 Mapping in  $E_1 E_2$  with Partial Participation**  
Minimum 2 tables and 1FK is required.

**For One-to-Many mapping:**

Minimum 2 table and 1FK

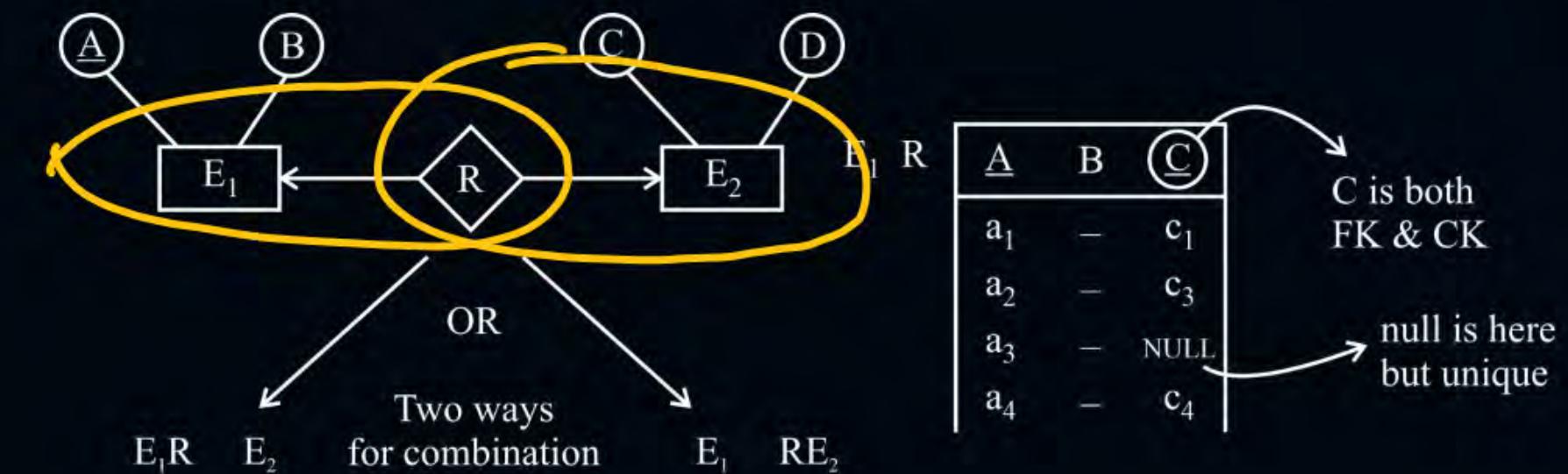
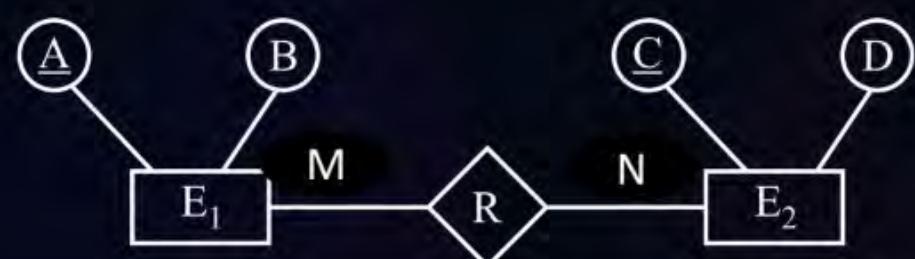
for both partial and total participation



**For Many-to-Many mapping:**

Key of R is {AC}

3 tables and 2FKs are required





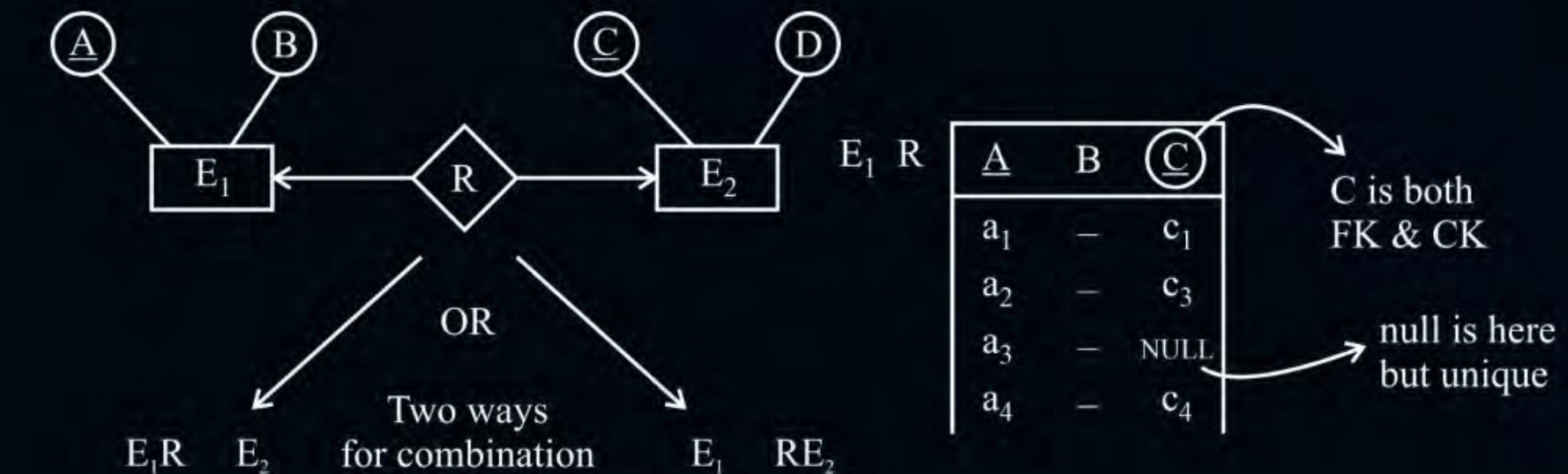
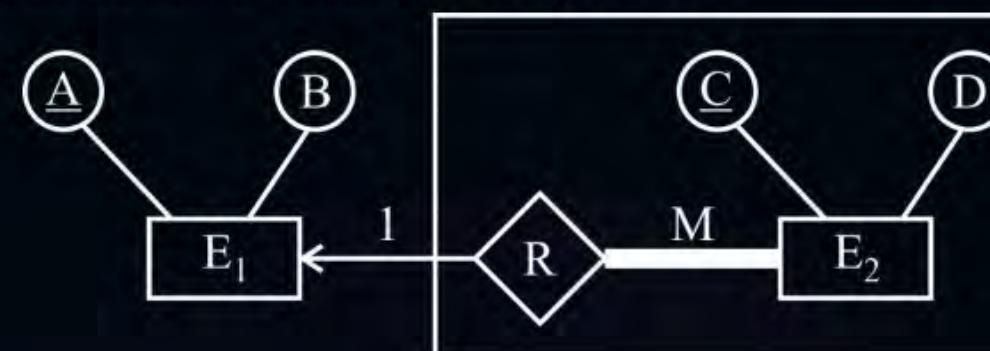
# Topic: Conclusion

**1 : 1 Mapping in E<sub>1</sub> E<sub>2</sub> with Partial Participation**  
Minimum 2 tables and 1FK is required.

**For One-to-Many mapping:**

Minimum 2 table and 1FK

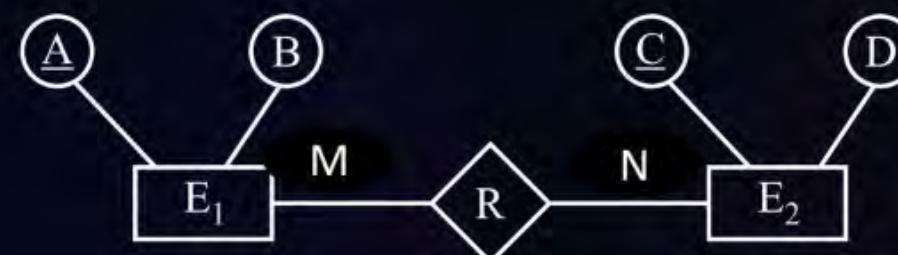
for both partial and total participation



**For Many-to-Many mapping:**

Key of R is {AC}

3 tables and 2FKs are required



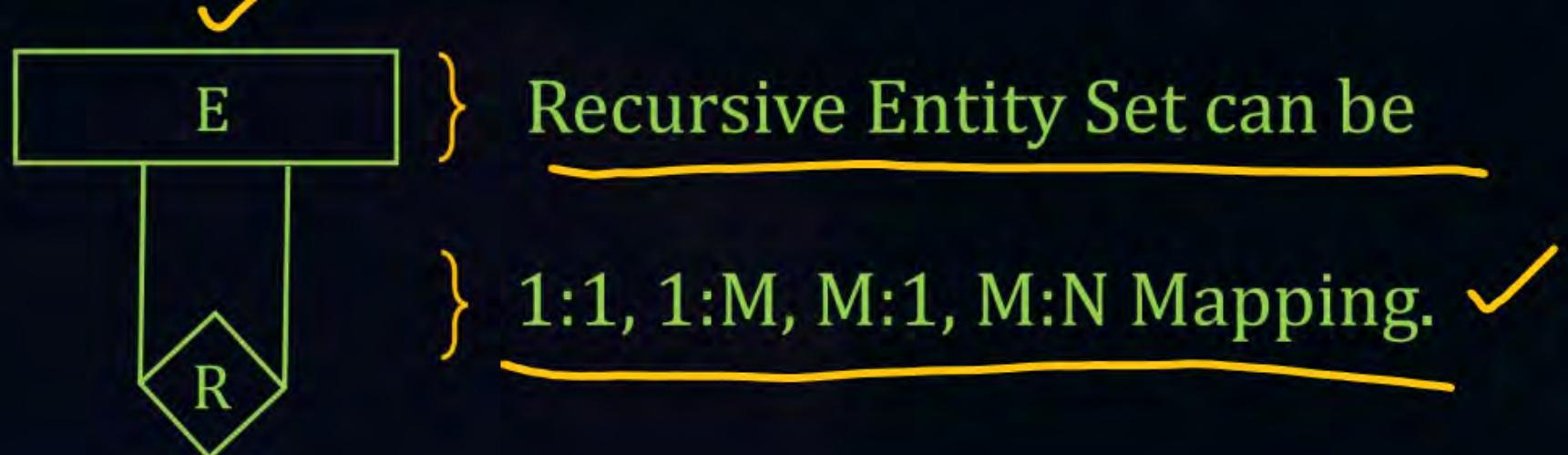
**1:1 mapping with atleast one total participation**



## Topic: Self Referential Relationship Set

**Self Referential Relationship Set:** Also called Recursive Entity Set

Entities of entity set (E) related to some other entities of same entity Set (E).





## Topic: Self Referential Relationship Set

✓  
**Example:** Emp is entity Set & reportsto is Relationship set related between Supervises & Subordinates

- (i) Each Supervisor can supervise many subordinates.  
Each subordinate reports to one supervisor.  
It is self referential because both Subordinate and Supervisor are employees.



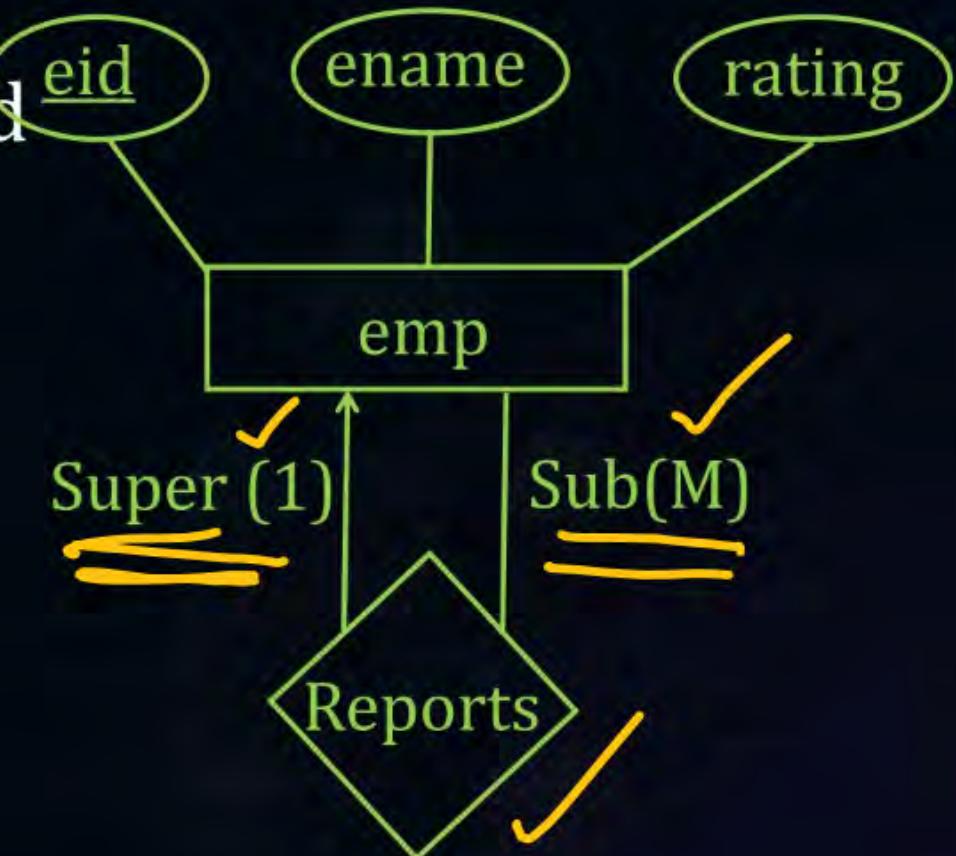
# Topic: Self Referential Relationship Set

**Example:** Emp is entity Set & reportsto is Relationship set related between Supervises & Subordinates

(i) Each Supervisor can supervise **many** subordinates.

Each subordinate reports to **one** supervisor.

It is self referential because both Subordinate and Supervisor are employees.



# Topic: Self Referential Relationship Set



Emp			
eid	ename	rating	
$e_1$	-	-	
$e_2$	-	-	
$e_3$	-	-	
$e_4$	-	-	

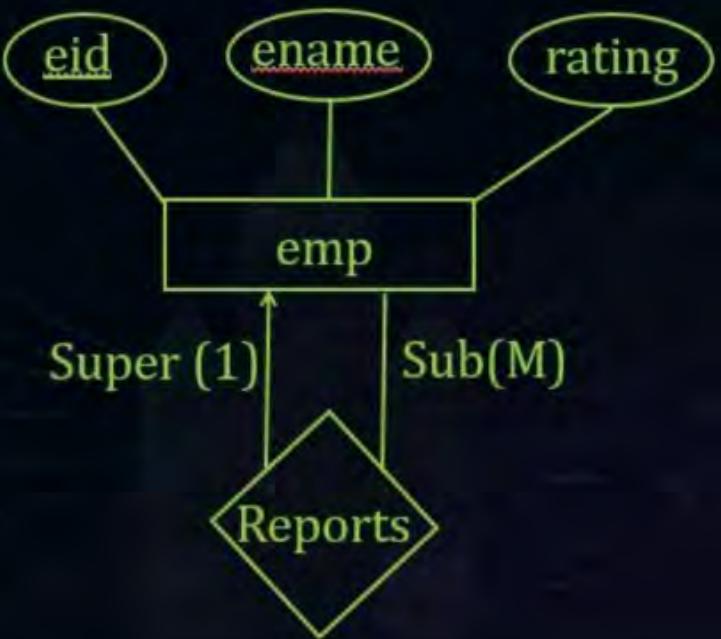
Emp -rep			
eid	ename	rating	Sup
$e_1$	-	-	Null
$e_2$	-	-	$e_1$
$e_3$	-	-	$e_1$
$e_4$	-	-	$e_2$

Rep (1: M)	
Sup	Sub
$e_1$	$e_2$
$e_1$	$e_3$
$e_2$	$e_4$



# Topic: Self Referential Relationship Set



Emp			
eid	ename	rating	
$e_1$	-	-	
$e_2$	-	-	
$e_3$	-	-	
$e_4$	-	-	

Rep (1: M)

Sup	Sub
$e_1$	$e_2$
$e_1$	$e_3$
$e_2$	$e_4$

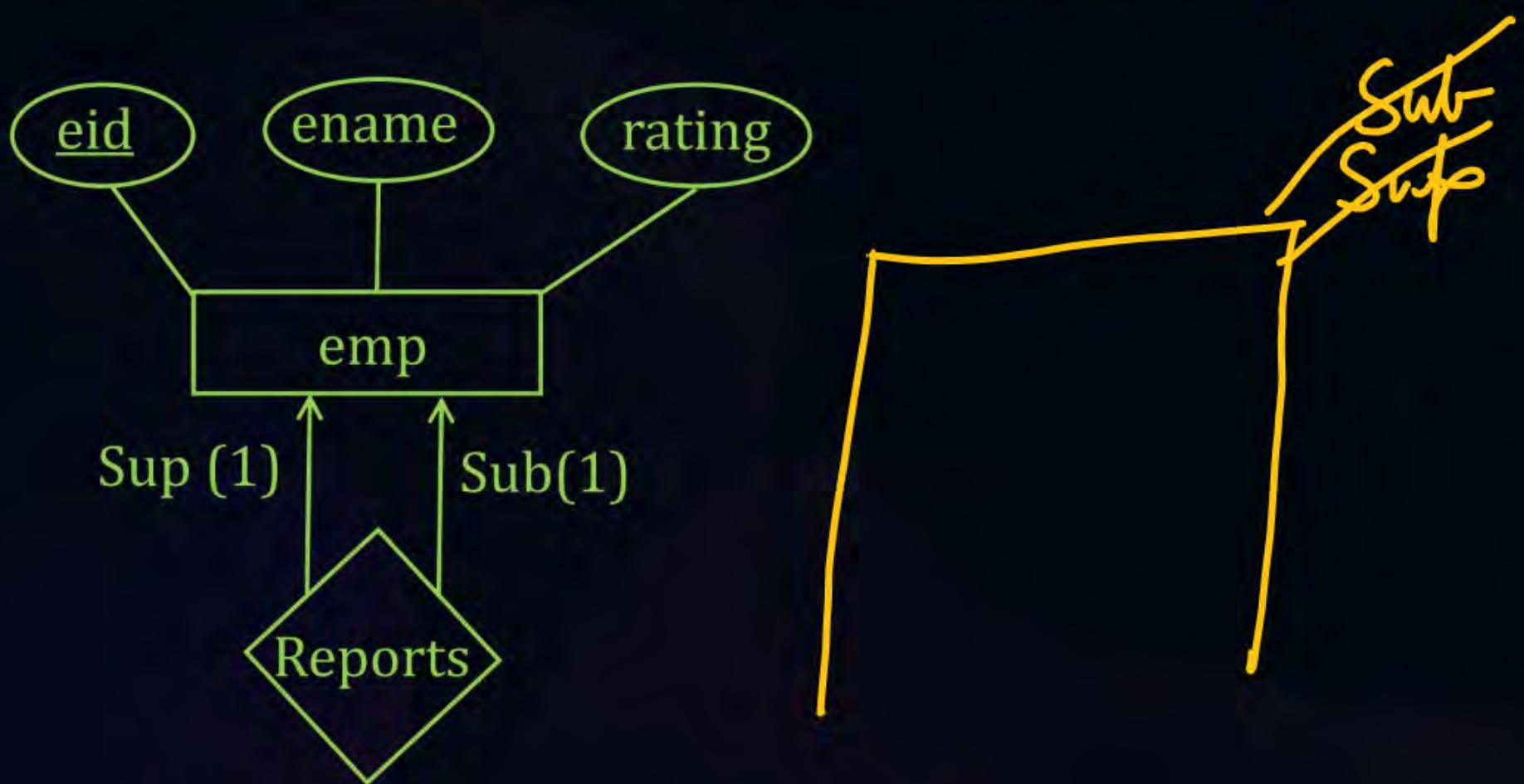
∴ Minimum 1 table and 1FK is required  
if it is self referential one to many  
relationship



## Topic: Self Referential Relationship Set

Let's take one-to-one mapping for same

- Each Supervisor can supervise one Subordinates.
- Each Subordinate reports to one Supervisor.

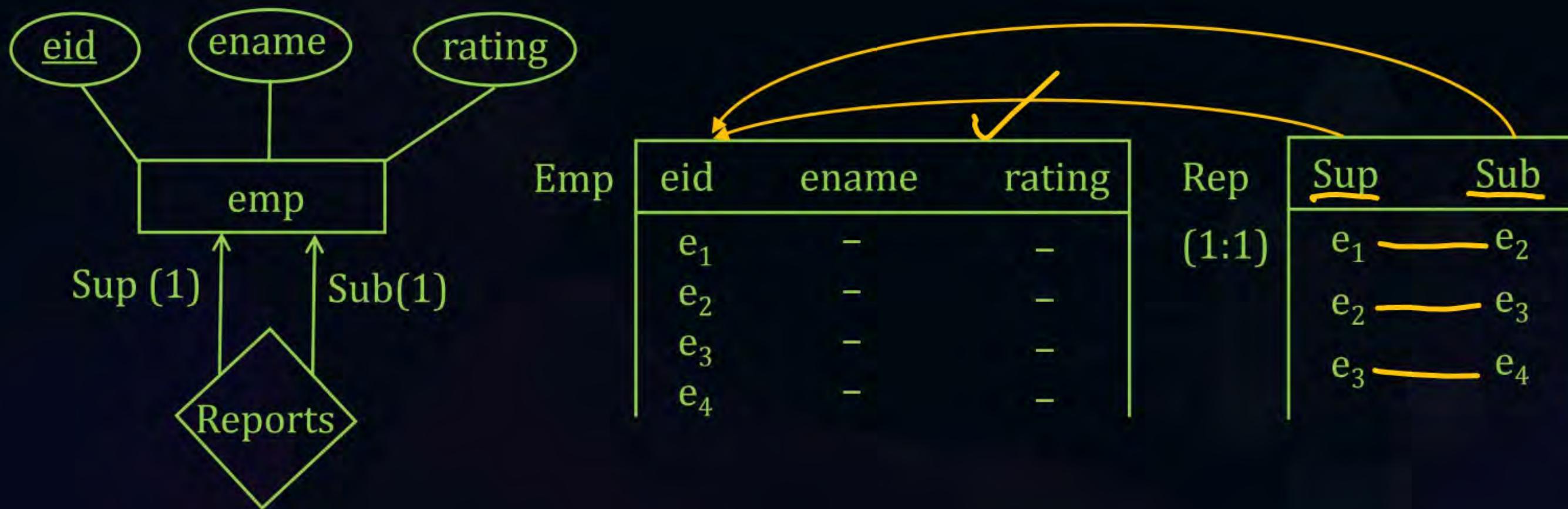




# Topic: Self Referential Relationship Set

Let's take one-to-one mapping for same

- Each Supervisor can supervise **one** Subordinates.
- Each Subordinate reports to **one** Supervisor.



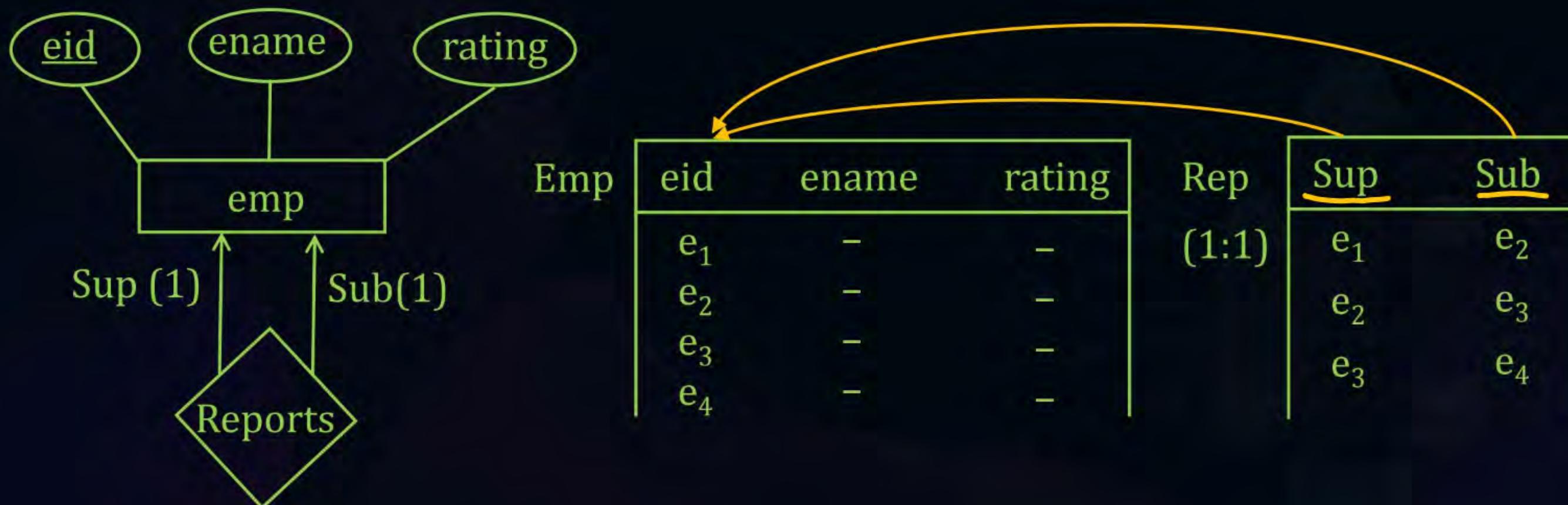


# Topic: Self Referential Relationship Set

Let's take one-to-one mapping for same

- Each Supervisor can supervise **one** Subordinates.
- Each Subordinate reports to **one** Supervisor.

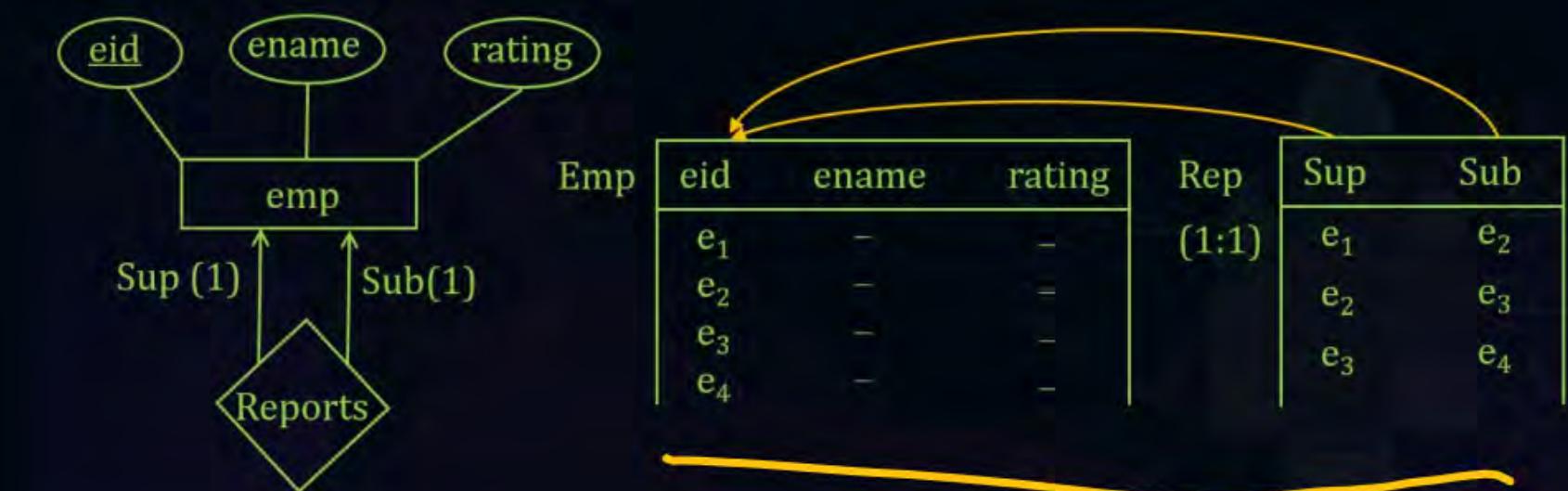
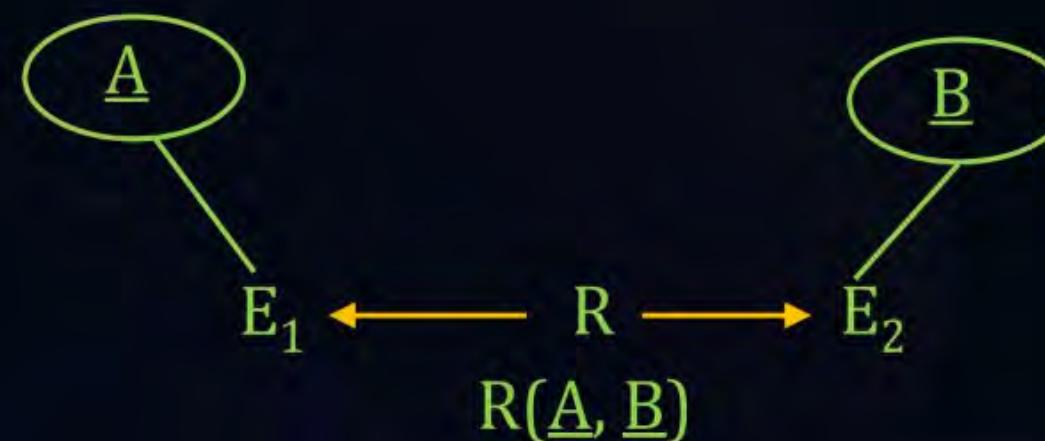
- In Rep. table, both sup and sub becomes key.  
Candidate key {sup, sub}





# Topic: Self Referential Relationship Set

Both are Keys because one Supervisor is not allowed to Supervise more than one Subordinate  
Subordinate is also unique, because one Subordinate is not allowed to report to more than one supervisor.





# Topic: Self Referential Relationship Set

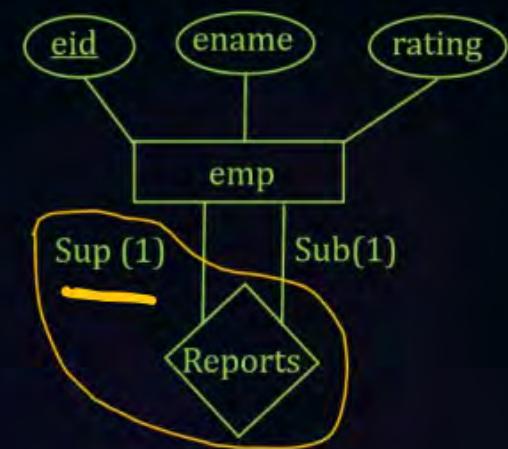
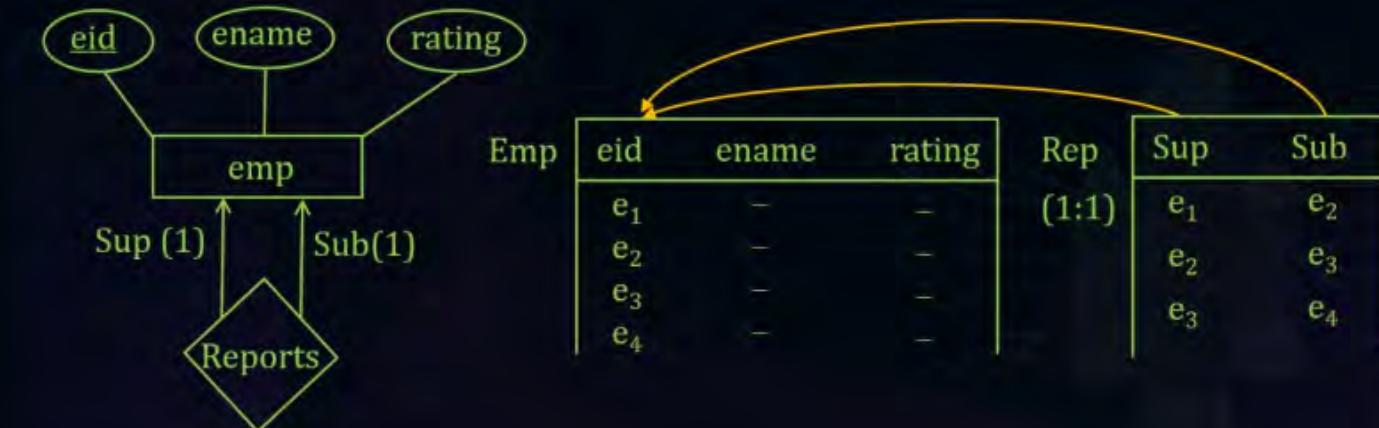
There could be two cases

$E_1 R E_2$

i.e. R combines with  $E_1$

$E_1 E_2 R$

i.e. R combines with  $E_2$





# Topic: Self Referential Relationship Set

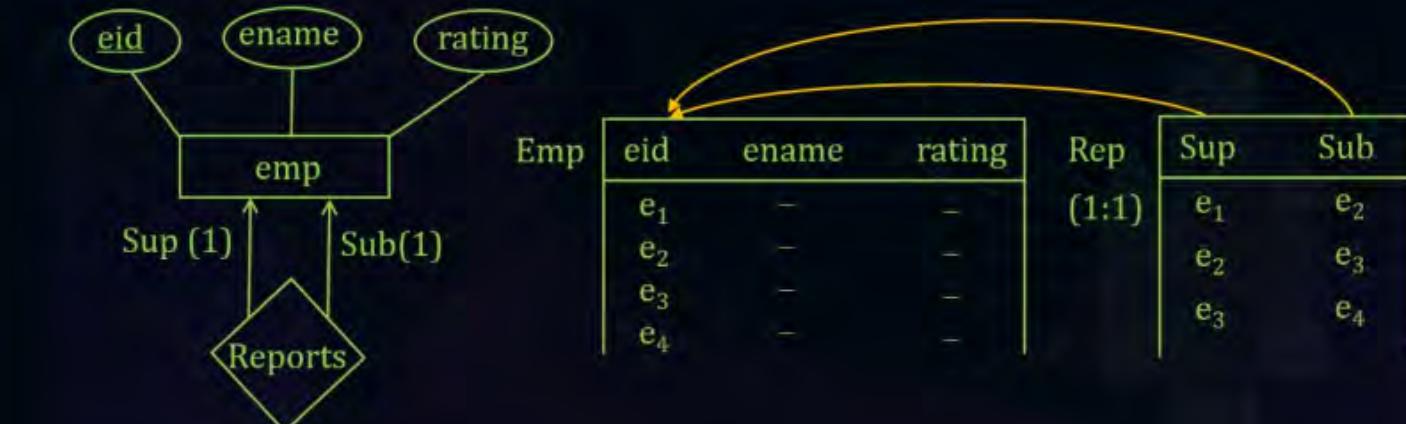
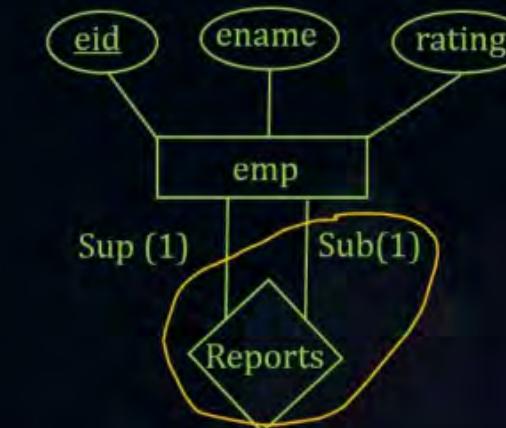
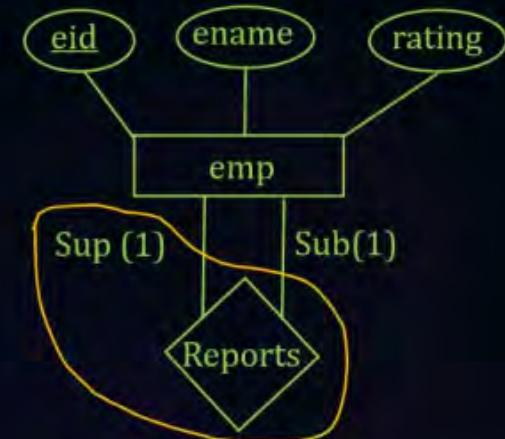
There could be two cases

$E_1 R E_2$

i.e. R combines with  $E_1$

$E_1 E_2 R$

i.e. R combines with  $E_2$



Emp -rep			
eid	ename	rating	Sup
e <sub>1</sub>	-	-	Null
e <sub>2</sub>	-	-	e <sub>1</sub>
e <sub>3</sub>	-	-	e <sub>2</sub>
e <sub>4</sub>	-	-	e <sub>2</sub>

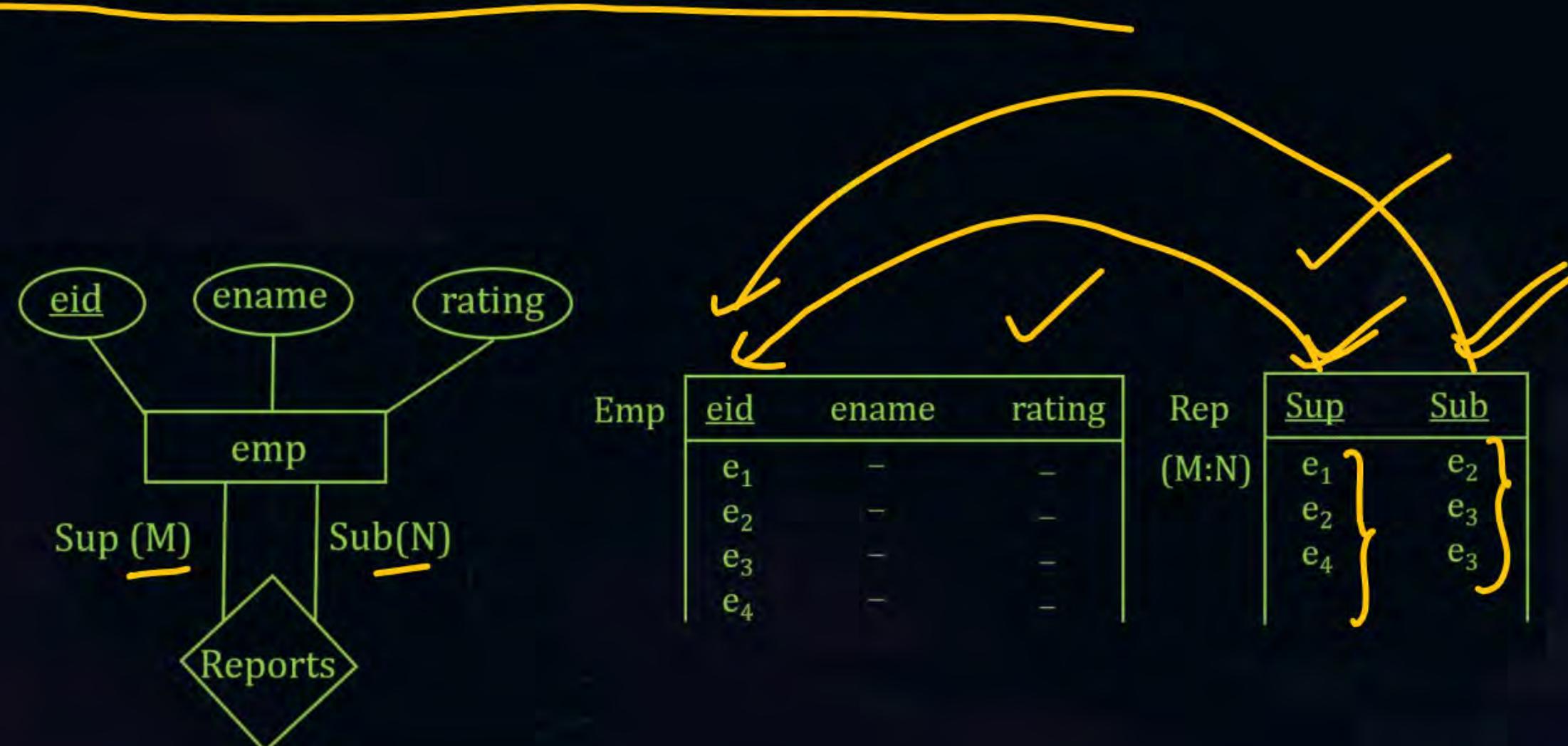
**Single Table** The only diff compared to previous one is here Sup is both FK and CK  
CKs are eid and Sup



## Topic: Self Referential Relationship Set

Each Supervisor can supervises many Subordinate.

Each Subordinate can report to many Supervisor.





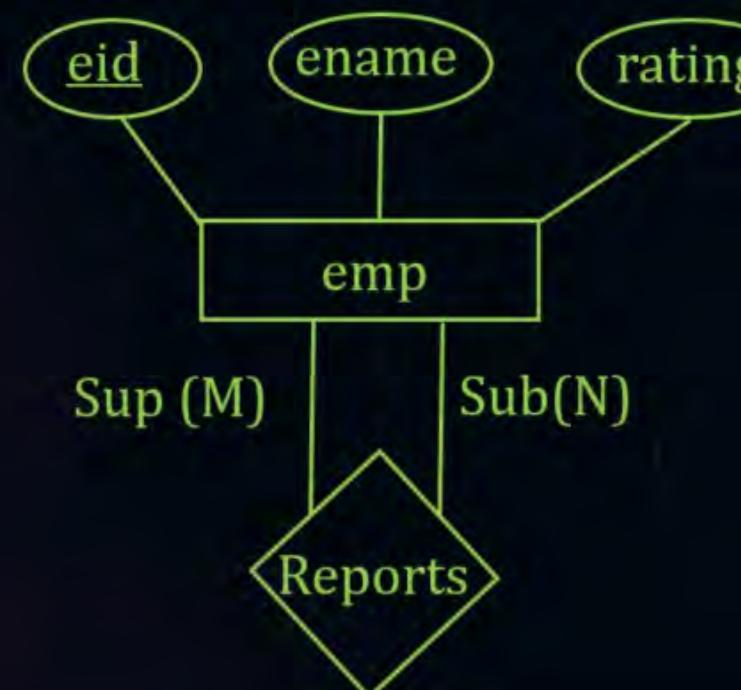
## Topic: Self Referential Relationship Set



Each Supervisor can supervises many Subordinate.

Each Subordinate can report to many Supervisor.

CK of Rep will be combination of both (Sub Sup) and we need a separate table, therefore 2 tables and 2FKs are required in total



Emp		
<u>eid</u>	ename	rating
e <sub>1</sub>	-	-
e <sub>2</sub>	-	-
e <sub>3</sub>	-	-
e <sub>4</sub>	-	-

Rep (M:N)	
Sup	Sub
e <sub>1</sub>	e <sub>2</sub>
e <sub>2</sub>	e <sub>3</sub>
e <sub>4</sub>	e <sub>3</sub>



## Topic: Self Referential Relationship Set

✓  
Except for M:N relationship for self-referential relations,  
1:1, 1:M, M:1 require only one table



## Topic: Self Referential Relationship Set

#Q. Binary one-to-many mapping

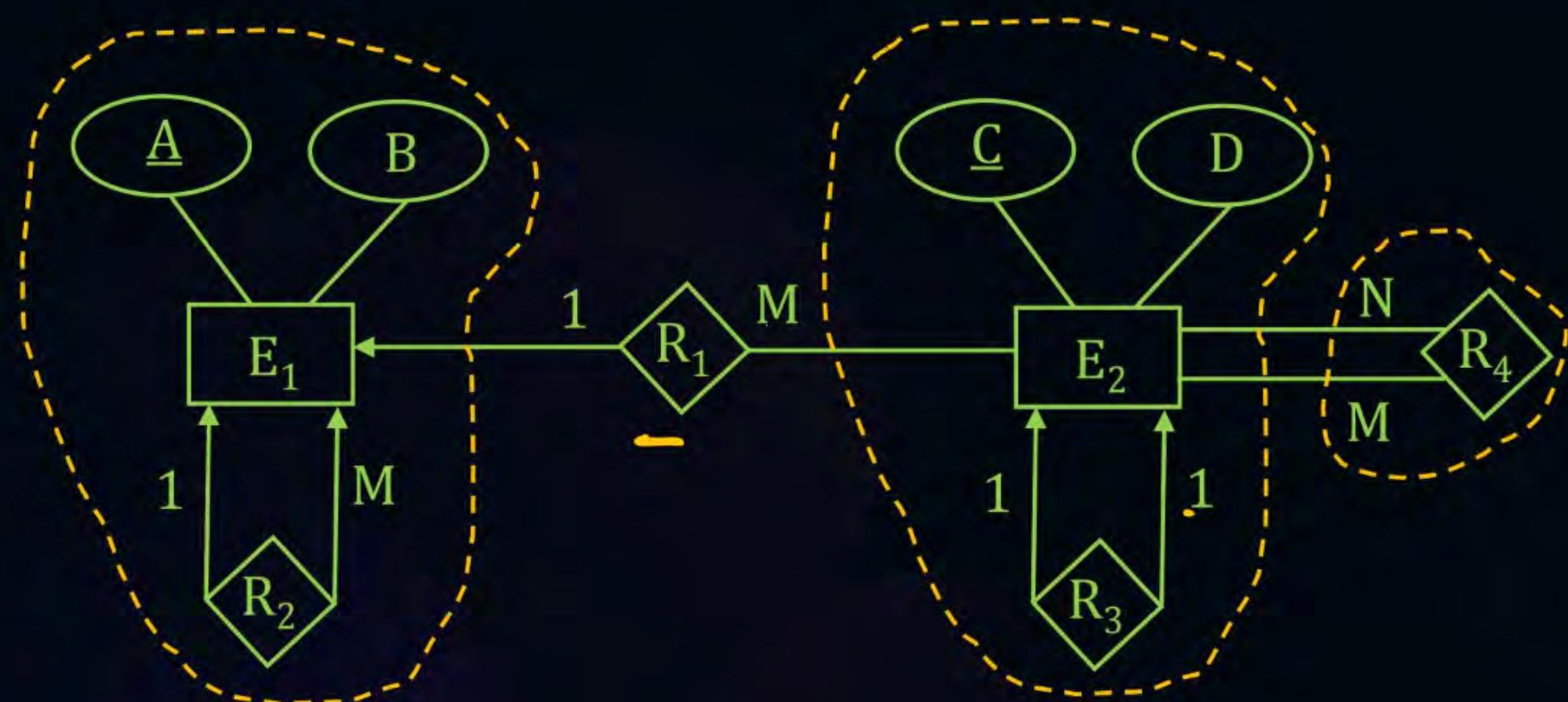


How many min RDBMS tables are required?



# Topic: Self Referential Relationship Set

#Q. Binary one-to-many mapping



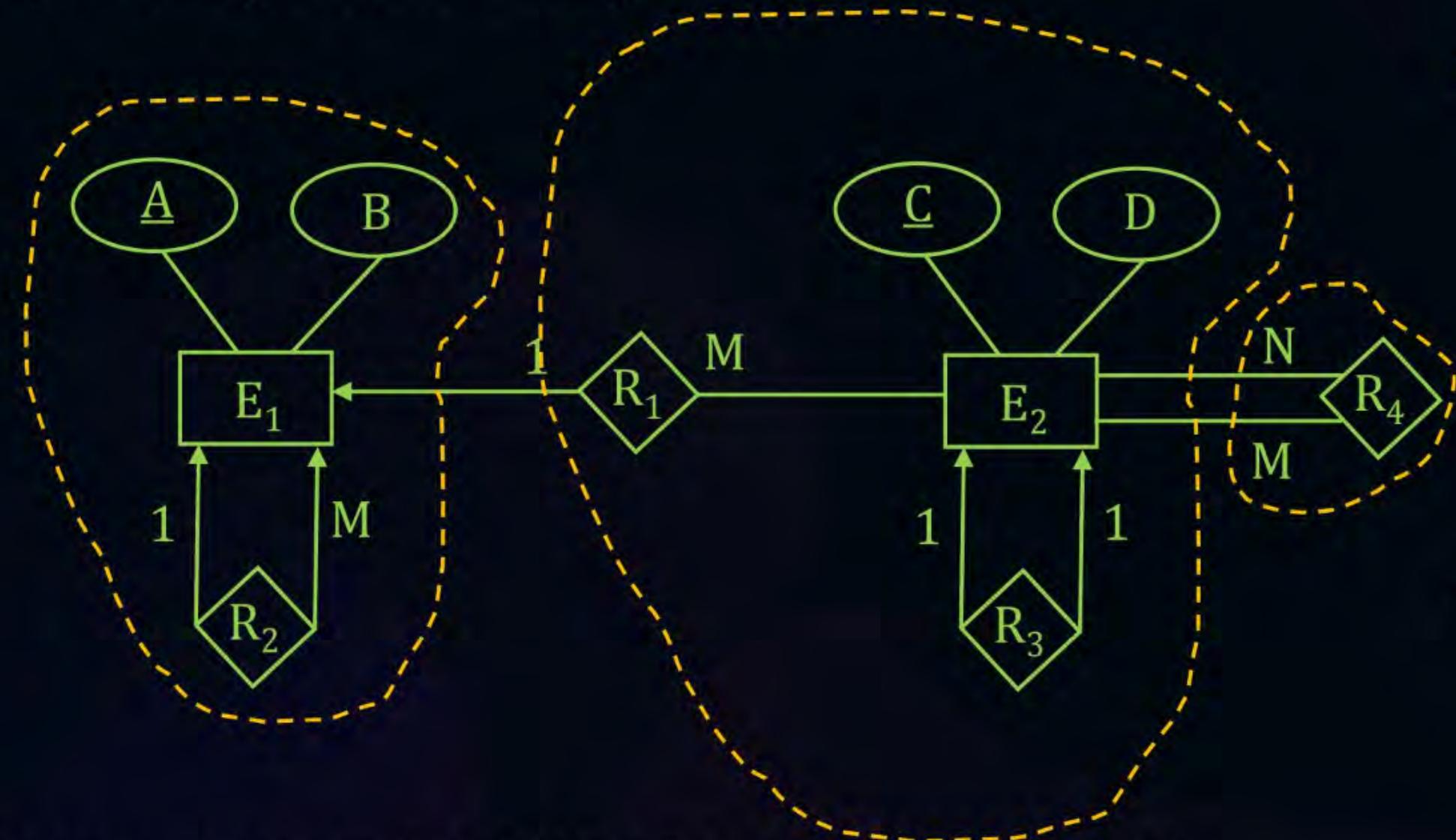
How many min RDBMS tables are required?



# Topic: Self Referential Relationship Set



#Q. Binary one-to-many mapping



Sol:

1. Self referential one to many can be combined into  $E_1$
2. Binary one-to many can combine into  $E_2$  and Self referential one-to - one can also be combined into  $E_2$ .
3. Self referential Many-to-many will be a separate table.  
⇒ Total 3 table required.

How many min RDBMS tables are required?



## Topic: Weak Entity Set

**Weak Entity Set:** ✓

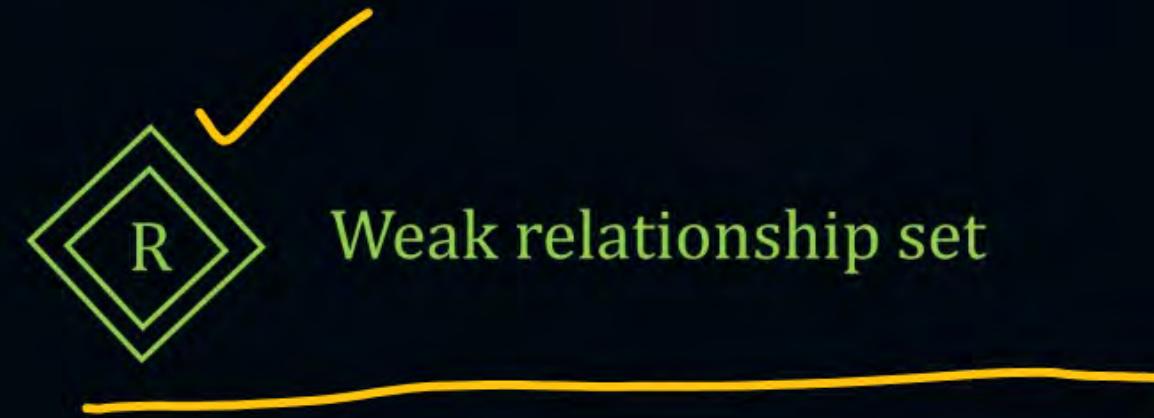
- Entity set with no Key.
- Attributes of weak entity set are not sufficient to differentiate records uniquely.
- Represented as A diagram showing a rectangle with a double border, representing the symbol for a weak entity set in entity-relationship modeling.
- Weak entities depends on other strong entity set called Identifier/Owner



## Topic: Weak Entity Set

### Example:

-  Partial key





## Topic: Weak Entity Set



### Example:

- Partial key
- Weak relationship set

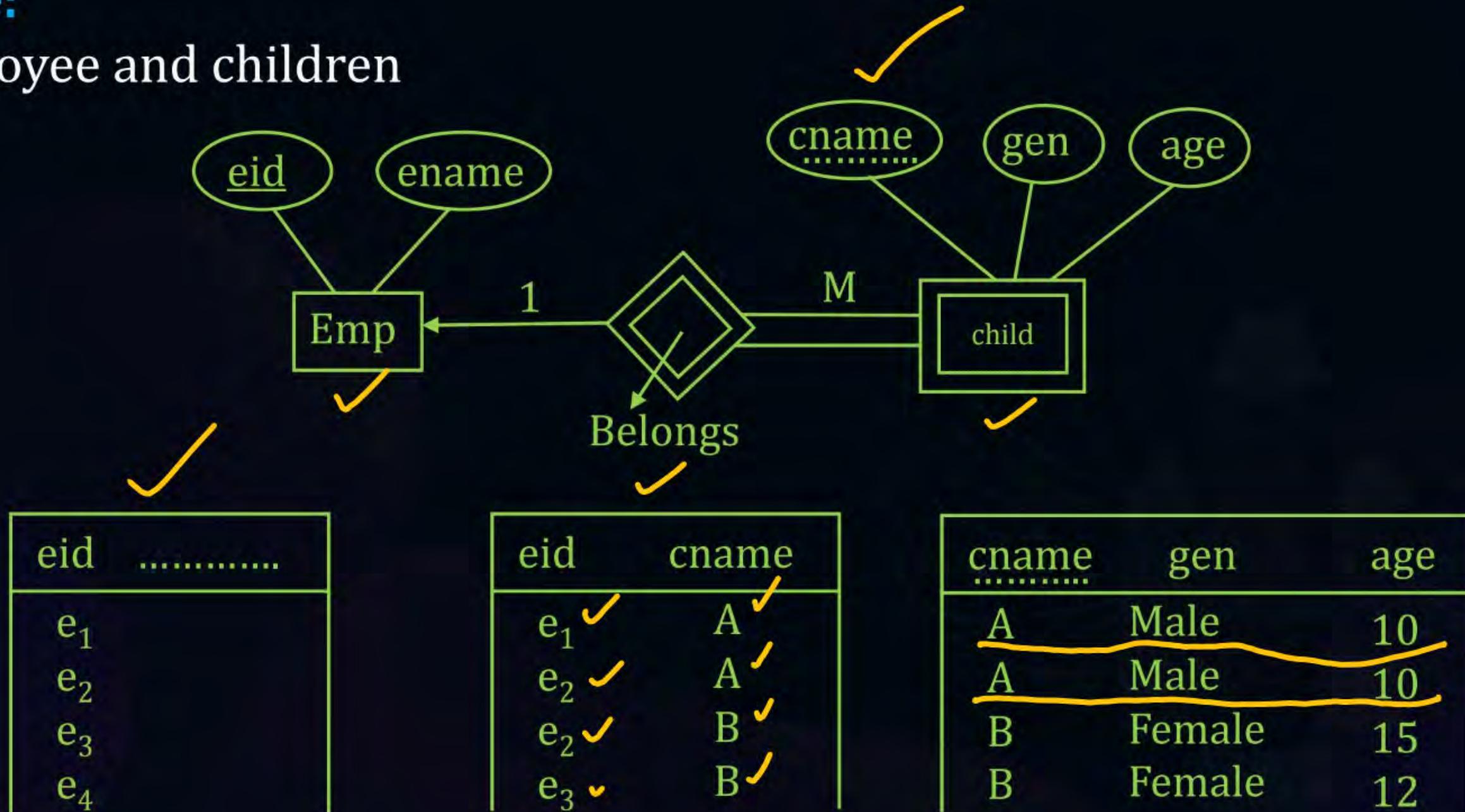
When there is a relationship between a strong entity set, and a weak entity set, the relationship is also weak relationship. One might be strong candidate key, other will be weak partial key.



# Topic: Weak Entity Set

## Example:

Employee and children





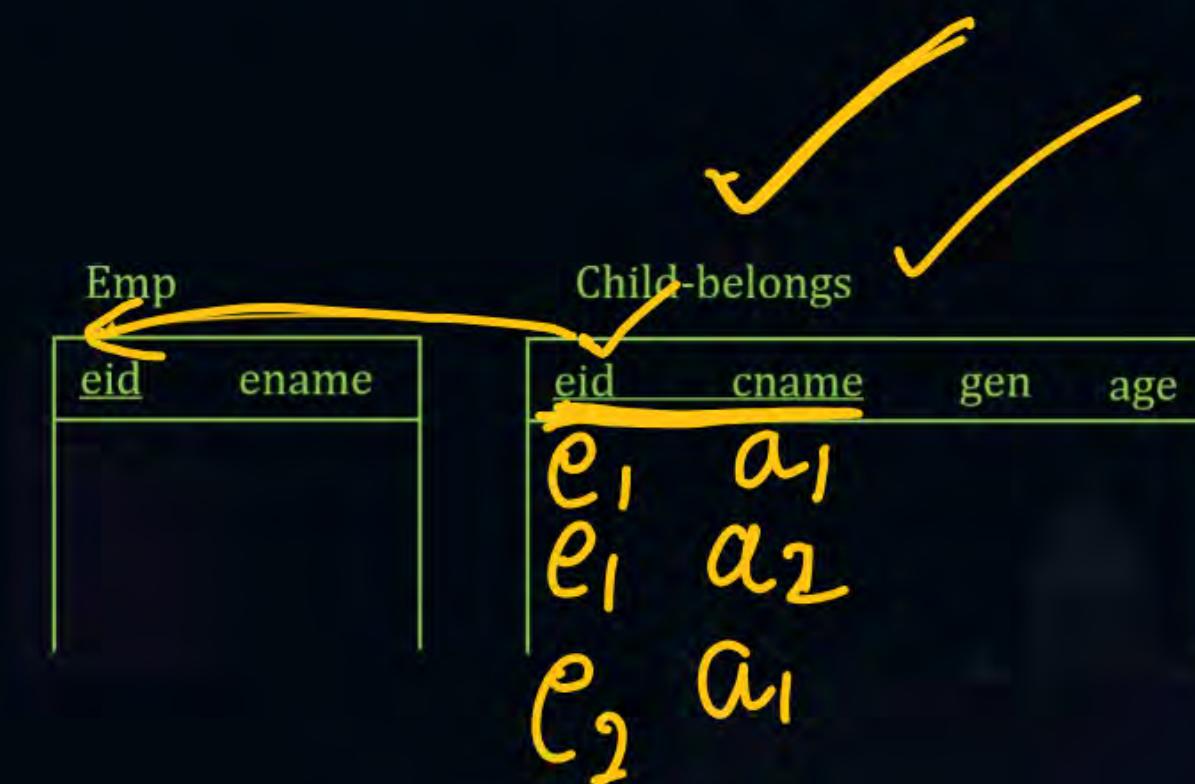
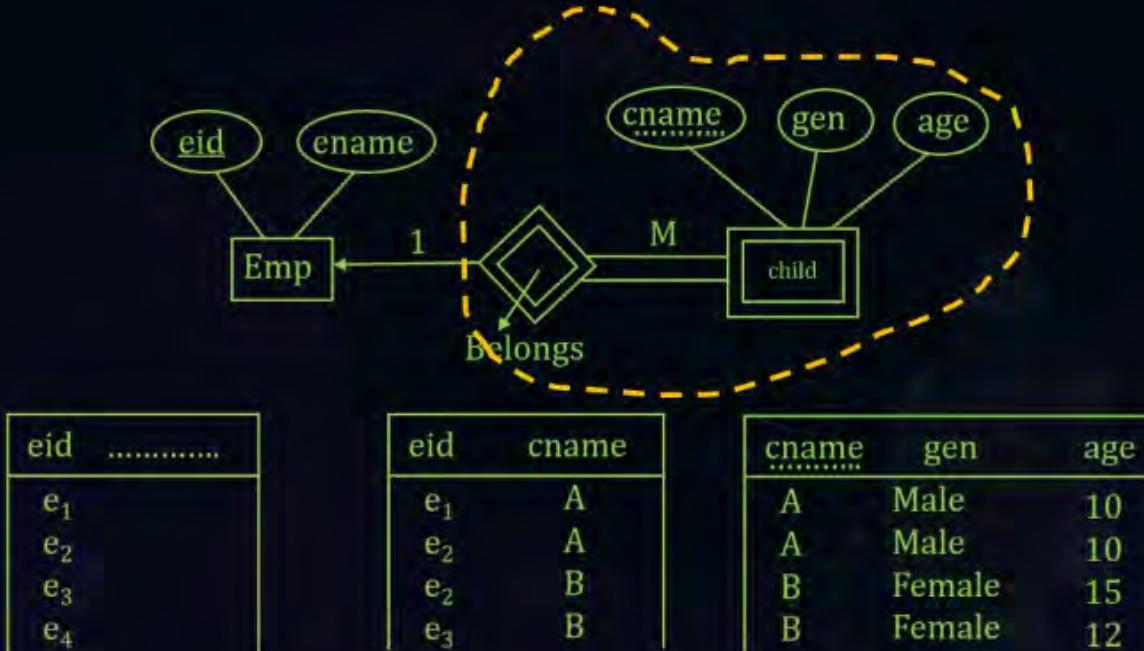
## Topic: Weak Entity Set

- Participation of WEAK entity should be TOTAL
- Mapping between strong entity set and weak entity set will be 1:M
- Weak Relationship set is uniquely identified by a combination of partial Key of Weak entity and PK of strong entity



# Topic: Weak Entity Set

- When talking about its RDBMS design, then there is no other alternative. Only one design, i.e. the relationship and weak entities should be together.

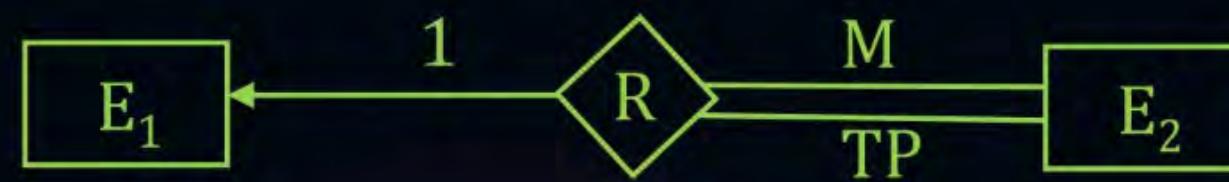


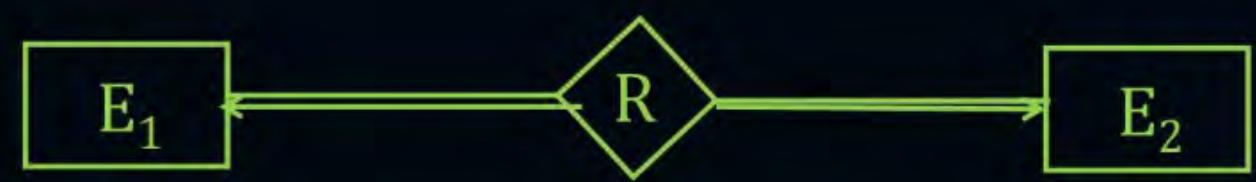
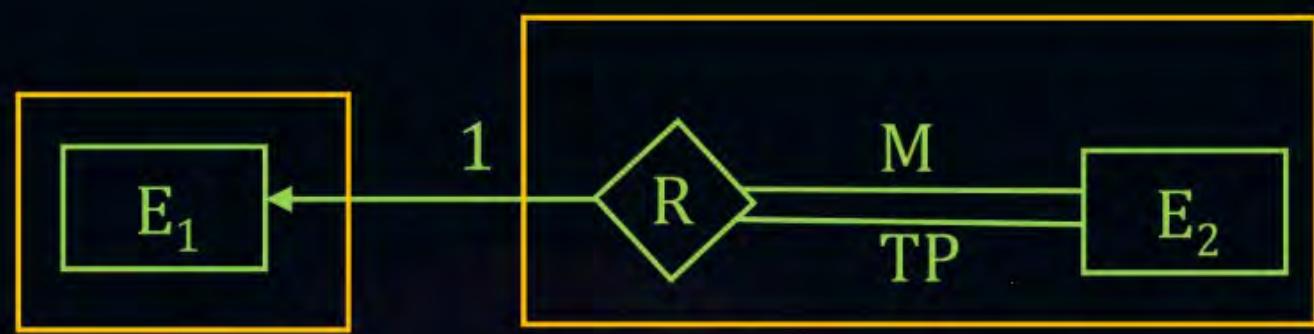
Key for belongs relationship {eidcname}

Weak Relationship is uniquely identified by a combination of partialKey and identifier Key.

Minimum 2 relational table required.

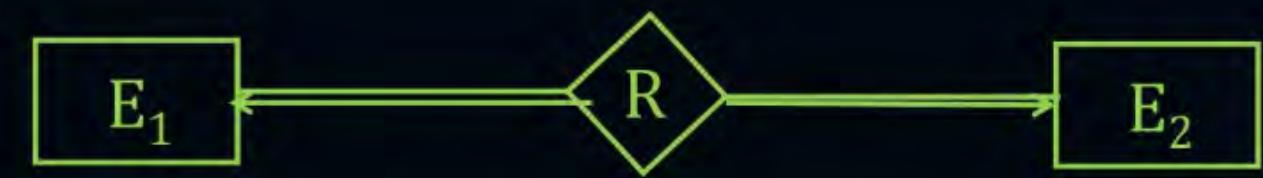
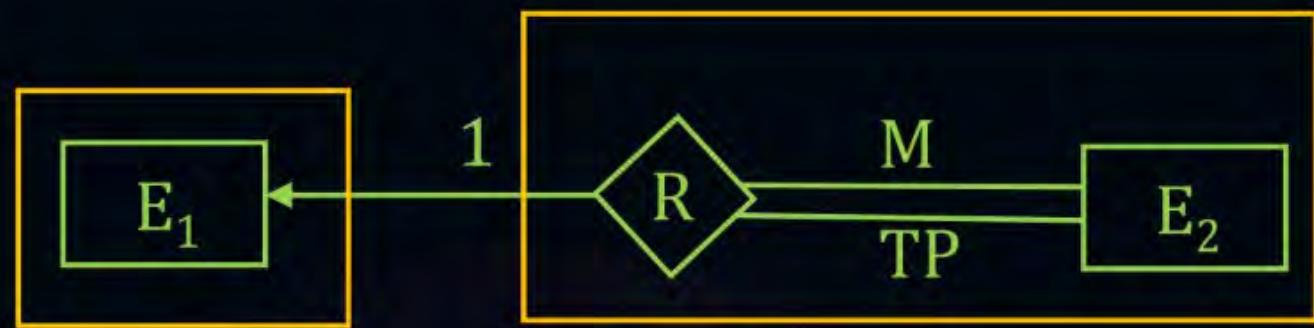


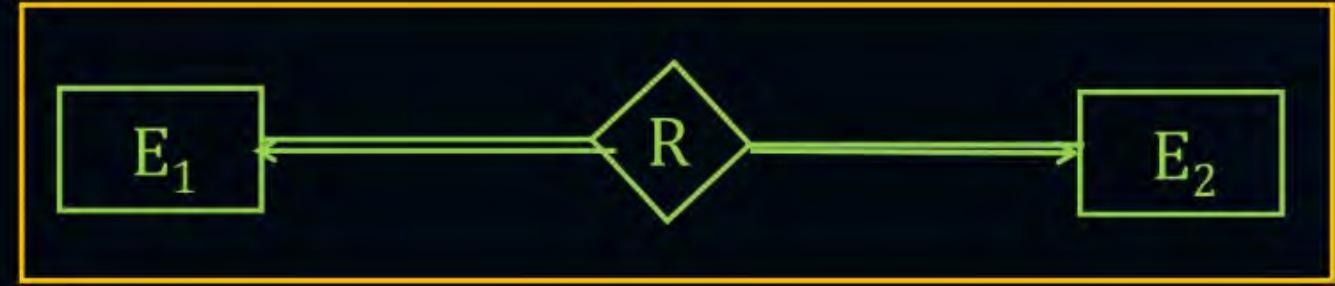


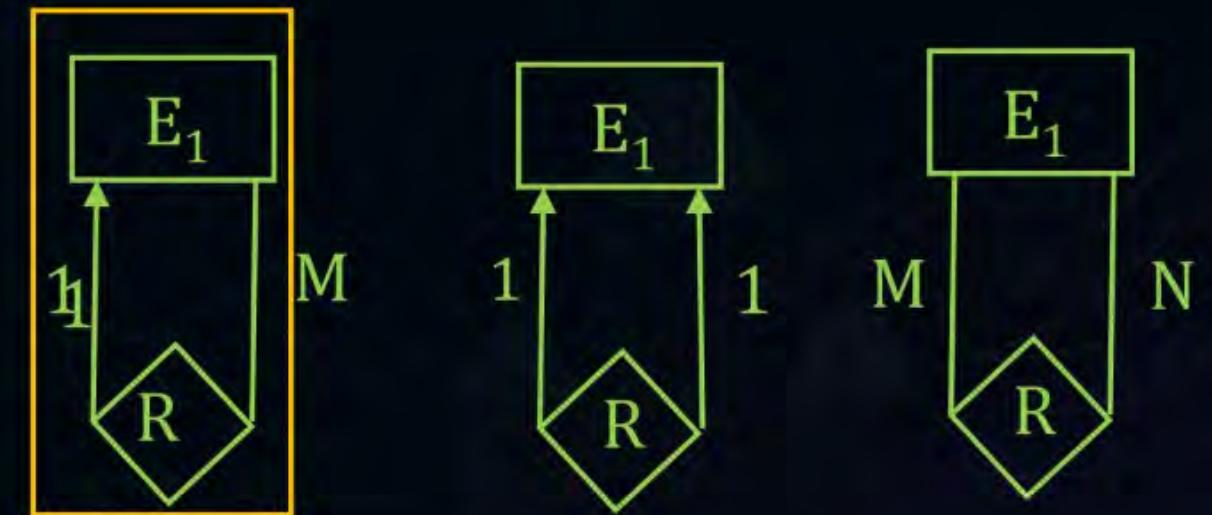
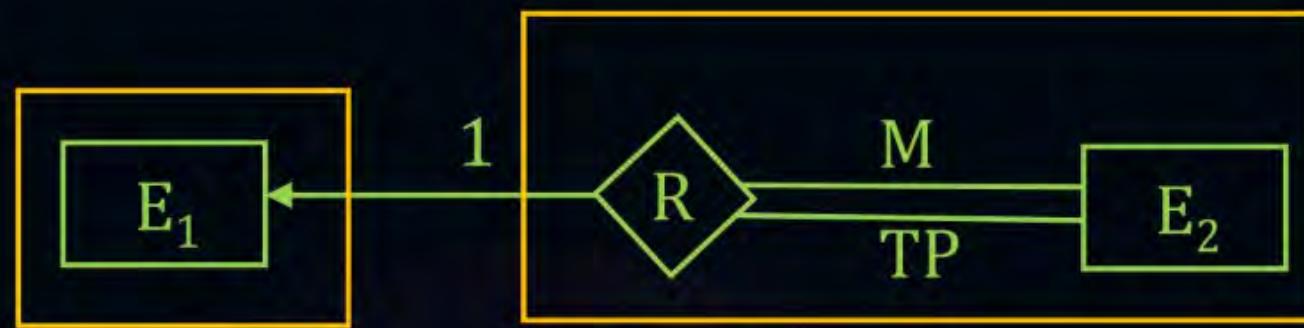




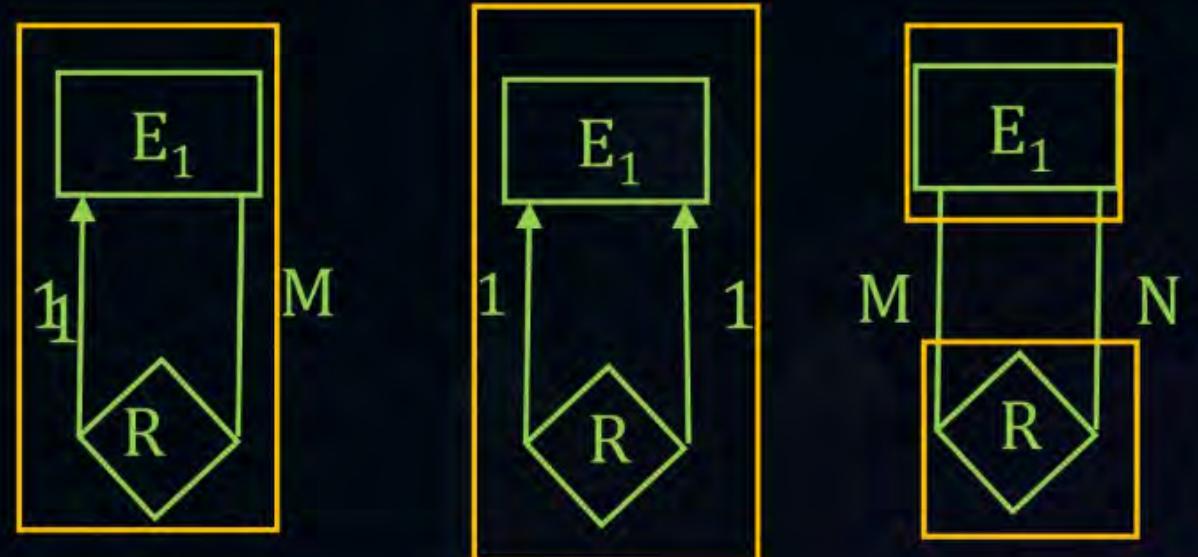


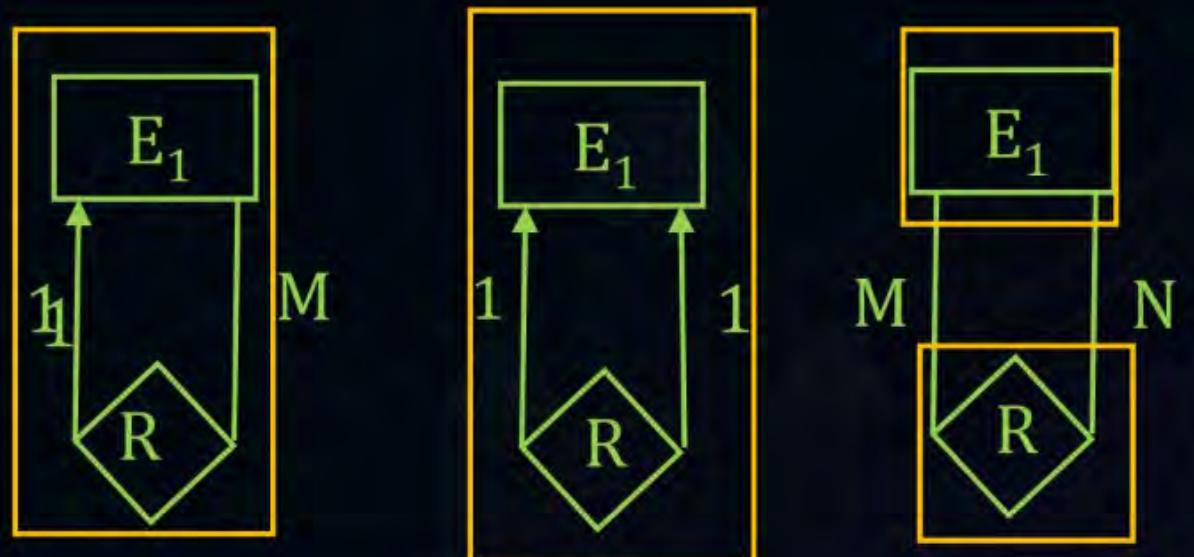
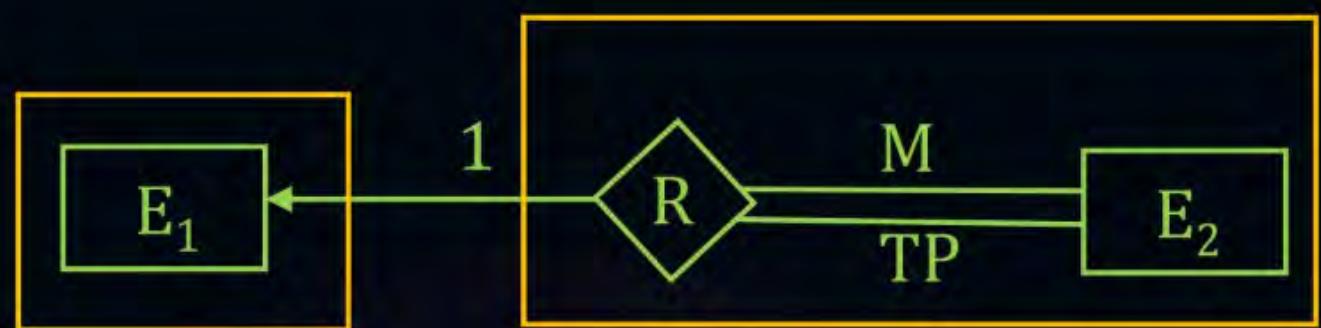














## Topic: Question

#Q.  $E_1 E_2$  entity set

$R_1 R_2$  relationship set related between  $E_1 & E_2$  with 1 : M & M : N respectively.

How many minimum relationship tables are required in RDBMS?

Gate



## Topic: Question

#Q.  $E_1 E_2$  entity set

$R_1 R_2$  relationship set related between  $E_1$  &  $E_2$  with 1 : M & M : N respectively.  
How many minimum relationship tables are required in RDBMS?

Sol:



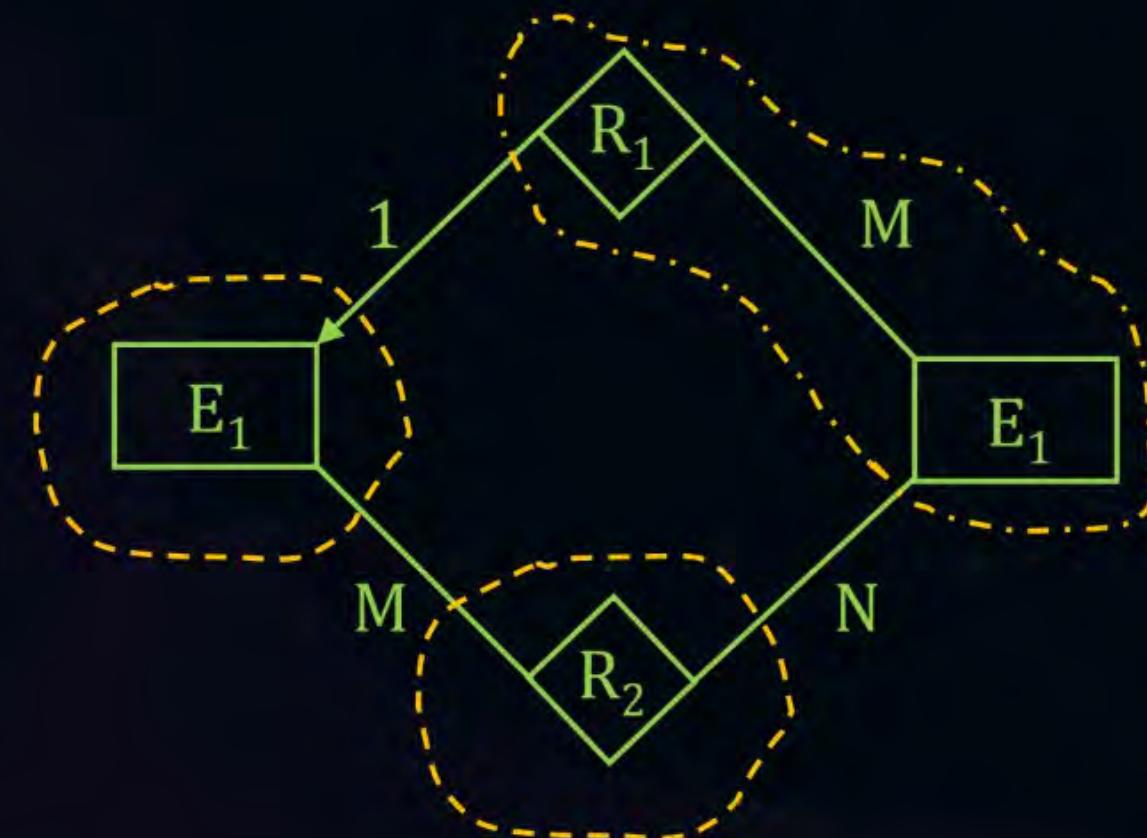


## Topic: Question

#Q.  $E_1 E_2$  entity set

$R_1 R_2$  relationship set related between  $E_1 & E_2$  with 1 : M & M : N respectively.  
How many minimum relationship tables are required in RDBMS?

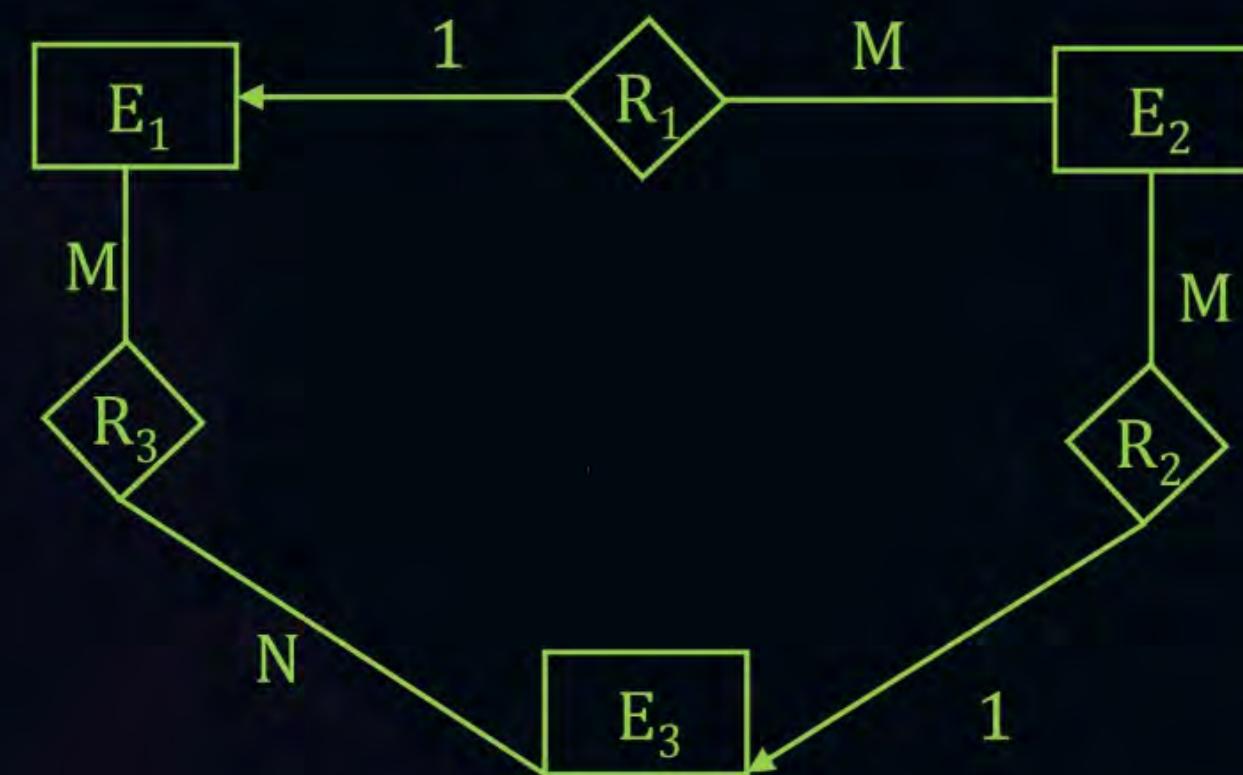
Sol:



#Q.  $E_1, E_2, E_3$  Entity Sets       $R_1, R_2, R_3$  Relationship Sets.  
 $R_1$  related between  $E_1$  &  $E_2$  with 1 : M     $R_2$  related between  $E_2$  &  $E_3$  with M : 1  
 $R_3$  related between  $E_1$  &  $E_3$  with M : N    How many min relationship tables?

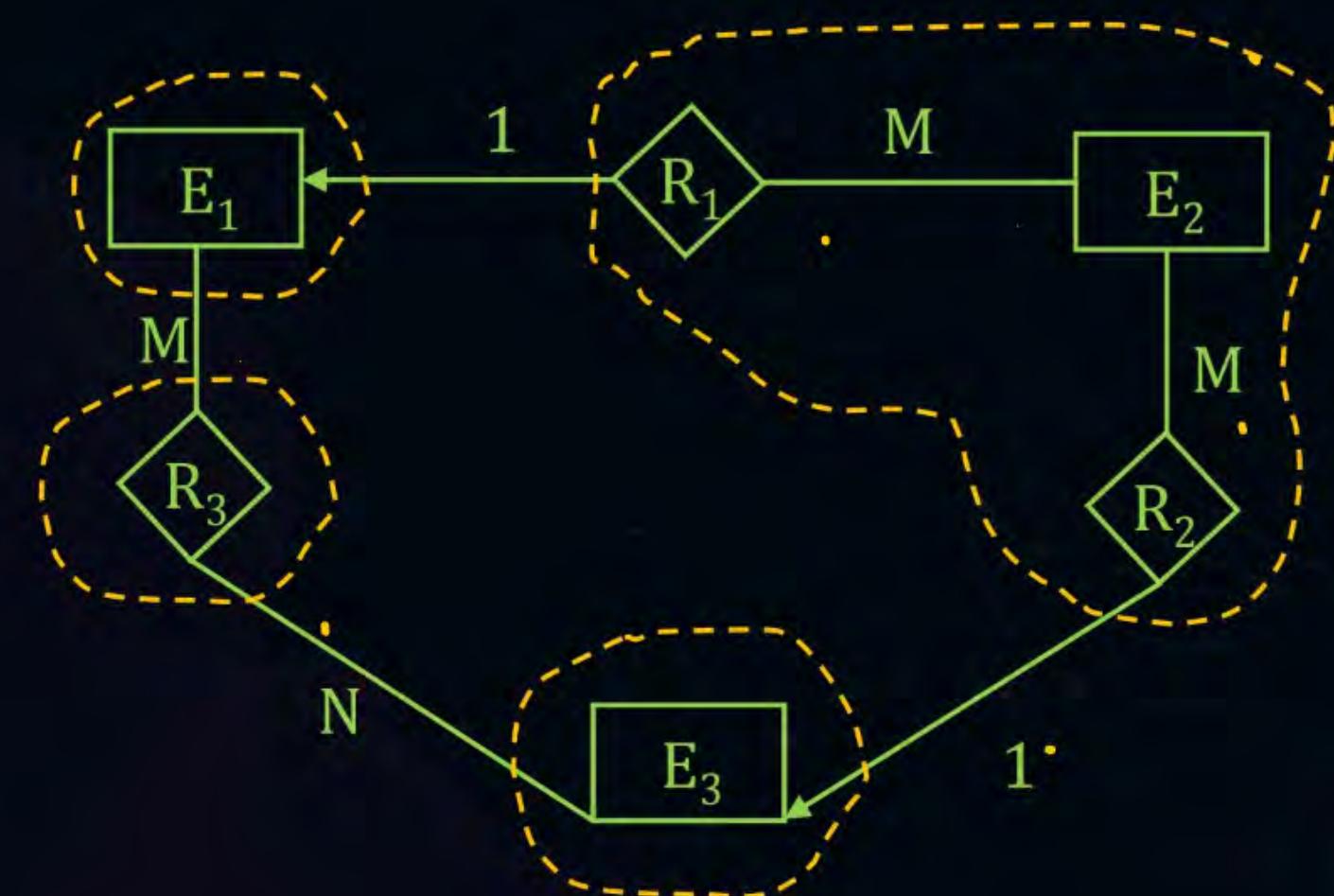
#Q.  $E_1, E_2, E_3$  Entity Sets       $R_1, R_2, R_3$  Relationship Sets.  
 $R_1$  related between  $E_1$  &  $E_2$  with 1 : M     $R_2$  related between  $E_2$  &  $E_3$  with M : 1  
 $R_3$  related between  $E_1$  &  $E_3$  with M : N    How many min relationship tables?

Sol:



#Q.  $E_1, E_2, E_3$  Entity Sets       $R_1, R_2, R_3$  Relationship Sets.  
 $R_1$  related between  $E_1$  &  $E_2$  with 1 : M     $R_2$  related between  $E_2$  &  $E_3$  with M : 1  
 $R_3$  related between  $E_1$  &  $E_3$  with M : N    How many min relationship tables?

Sol:





THANK - YOU