# COMPUTER SCIENCE

Database Management System

# ER Model



Lecture_2

Vijay Agarwal sir


TOPICS TO BE COVERED
01
Foreign Key Concepts
02
ER to RDBMS Conversions

# ER MODEL (Conceptual Design)

Entity

Entity set 

Relationship

Relationship set

Attribute &

Attribute type

① Simple & Composite Attribute

② Single Valued & Multivalued Attribute

③ Stored & Derived Attribute

④ Descriptive Attribute

⑤ key Attribute

⑥ Complex Attribute (Composite + multivalued)

Degree of Relationship set : # Entity Set

Unary
Binary
Ternary
N-ary

Participation Constraints ( Total (Full) or 100% (Double line)
Partial Participation

# Cardinality Mapping : Entities

1. One to one (1:1)


2. One to Many (1:M)


3. Many to One (M:1)


4. Many to many (M:N)


Strong Entity Set & Weak Entity Set

# Foreign key : Foreign key is a Set of Attribute Reference to the Primary key (or) Alternate key of Same table (or) Some other Table.

# Referential Integrity Constraints

## Foreign key

Foreign key is a set of attributes that references primary key or alternative key of the same relation or other relation.

USP

1. CNC
2. Dummy (View)
3. Arrow (1 to M) & (M to 1)
4. Referencing Relation
   Refferenced Relation

If $X_{Attribute} \rightarrow$ [Prime Attribute]

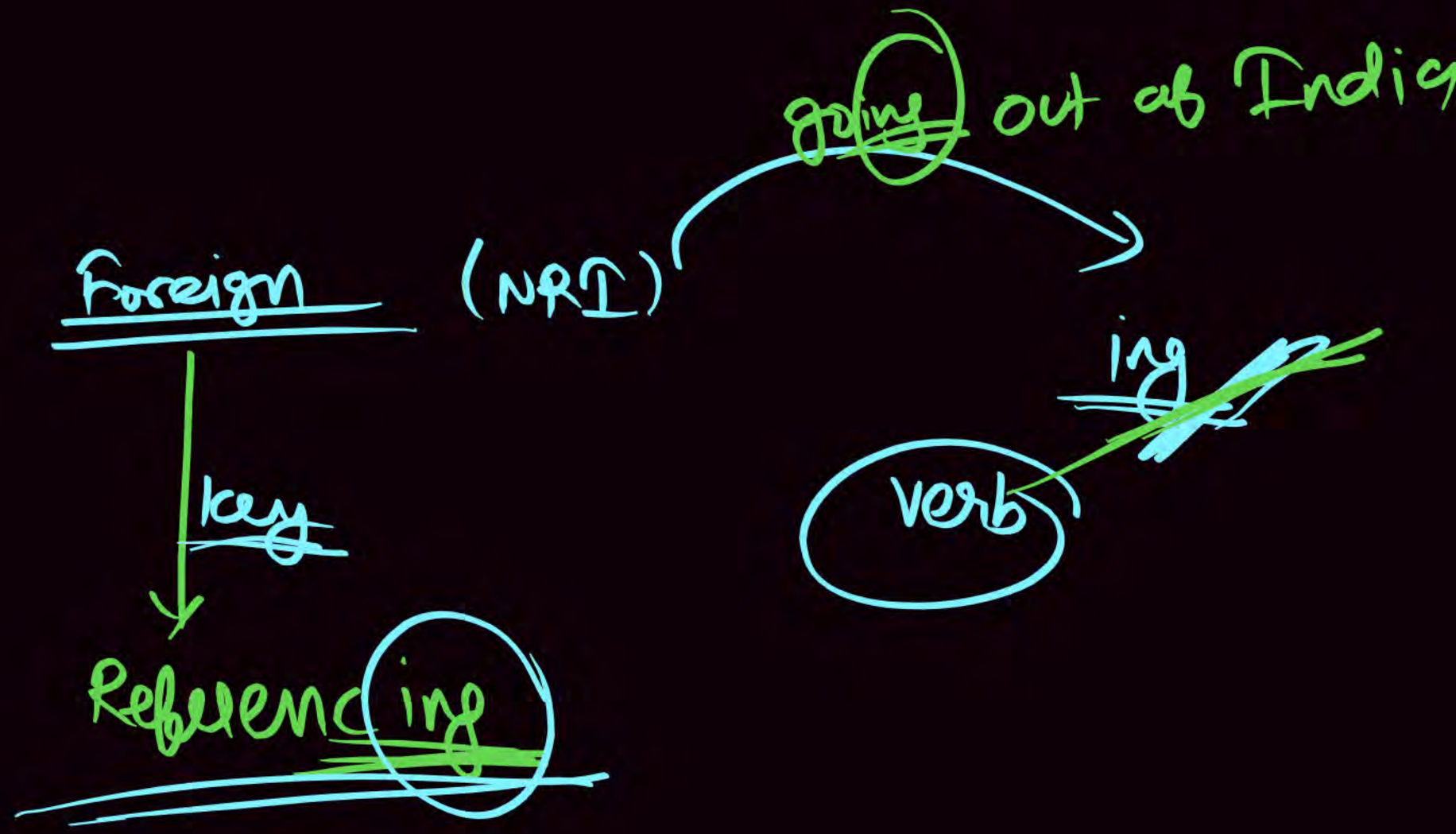
## Partial Dependency

AB → C is Partial

if A → C

or

B → C


going out of India
Foreign (NRI)
ing
verb
key
Referencing

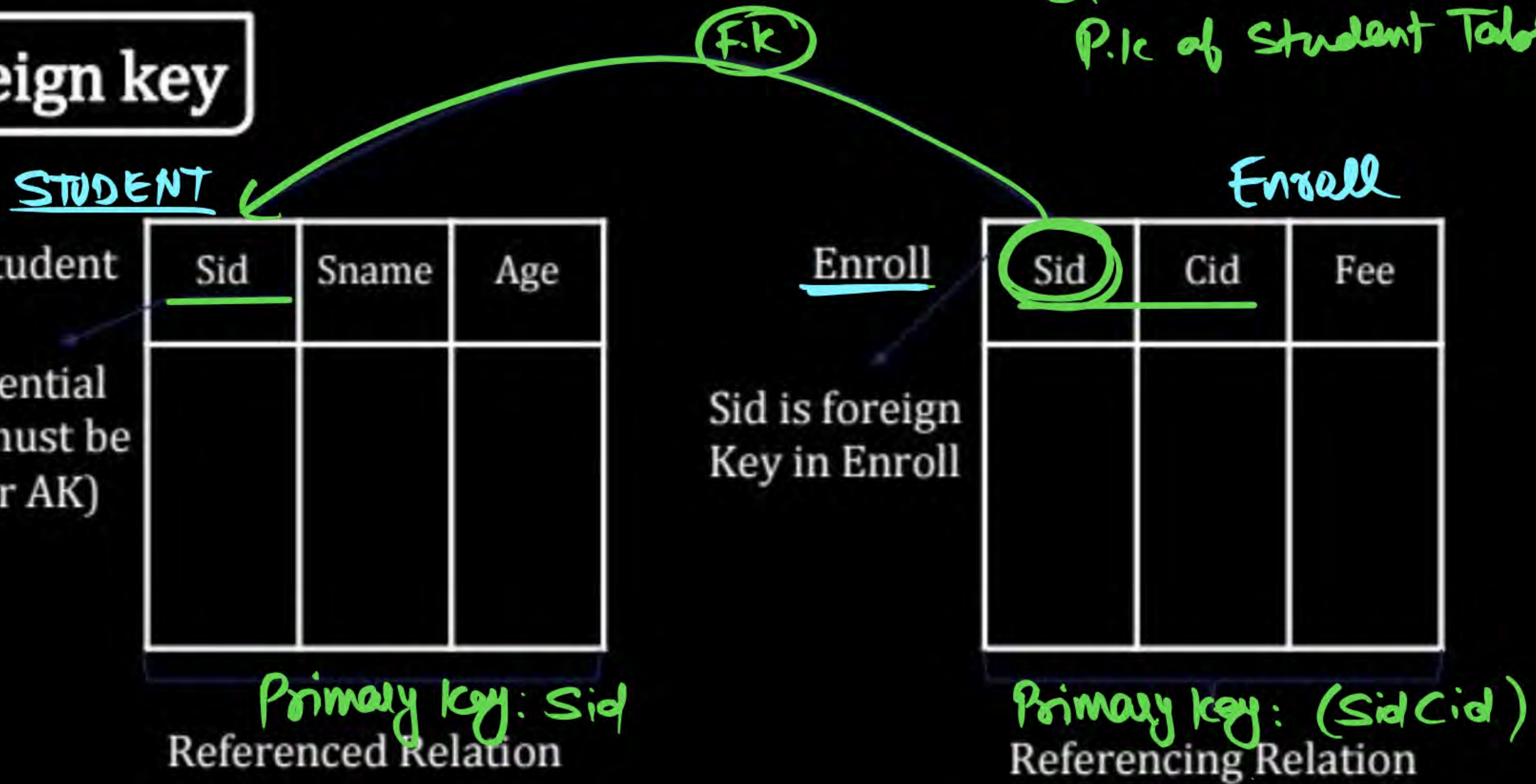
## Referenced Relation : The Table which is Referenced by foreign key is known as Referenced Relation (Parent Table | Relation).

## Referencing Relation : The Table which Contain the foreign key is known as Referencing Relation (CHILD Table/Relation).

# Referential Integrity Constraints

## Foreign key

Sid is F.k Reference to P.k of Student Table.



# Foreign Key

**Foreign Key:** is a set of Attribute reference to the primary key or alterative key of the same table or same other table.

Same or Different table [Primary key or alternative key] ← Reference — [Foreign key]

It is used to relate or relation (table) with other or same relation (table)

1. **Referencing Relation:** Table which contain the foreign key is known as Referencing Relation [CHILD Relation].
2. **Referenced Relation:** Table which is referenced by foreign key is referenced relation.

Referenced Relation

[Parent Table]

Referencing Relation

[CHILD Table]

which contain Foreign key.

# Foreign Key Constraint
# [Referential Integrity Constraint]

| STUDENT | | |
|---|---|---|
| **Sid** | **Sname** | **Login** |
| $S_1$ | A | -@ |
| $S_2$ | A | --@ |
| $S_3$ | B | ---@ |
| $S_4$ | $C_1$ | ----@ |

[Sid: Primary Key]

Referenced Relation (Parent)

| Enrolled | | |
|---|---|---|
| **Sid** | **Cid** | **Fees** |
| $S_1$ | $C_1$ | 5K |
| $S_1$ | $C_2$ | 6K |
| $S_2$ | $C_1$ | 7K |
| $S_3$ | $C_2$ | 8K |

[Sid, Cid: Primary Key]

Referencing Relation (CHILD)

Sid of Enrolled table is the foreign key referencing to the primary key of student table.

# Foreign Key Constraint
# [Referential Integrity Constraint]

F.K

F.K

| STUDENT | | |
|---|---|---|
| **Sid** | **Sname** | **Login** |
| $S_1$ | A | -@ |
| $S_2$ | A | --@ |
| $S_3$ | B | ---@ |
| $S_4$ | $C_1$ | ----@ |

| Enrolled | | |
|---|---|---|
| **Sid** | **Cid** | **Fees** |
| $S_1$ | $C_1$ | 5K |
| $S_1$ | $C_2$ | 6K |
| $S_2$ | $C_1$ | 7K |
| $S_3$ | $C_2$ | 8K |

Sid of Enrolled table is the foreign key referencing to the primary key of student table.

[Sid: Primary Key]

Referenced Relation (Parent)

[Sid, Cid: Primary Key]

Referencing Relation (CHILD)

Foreign key Reference to the Primary key

(or) Alternate key of the same table

(or) Some other Table.

Note

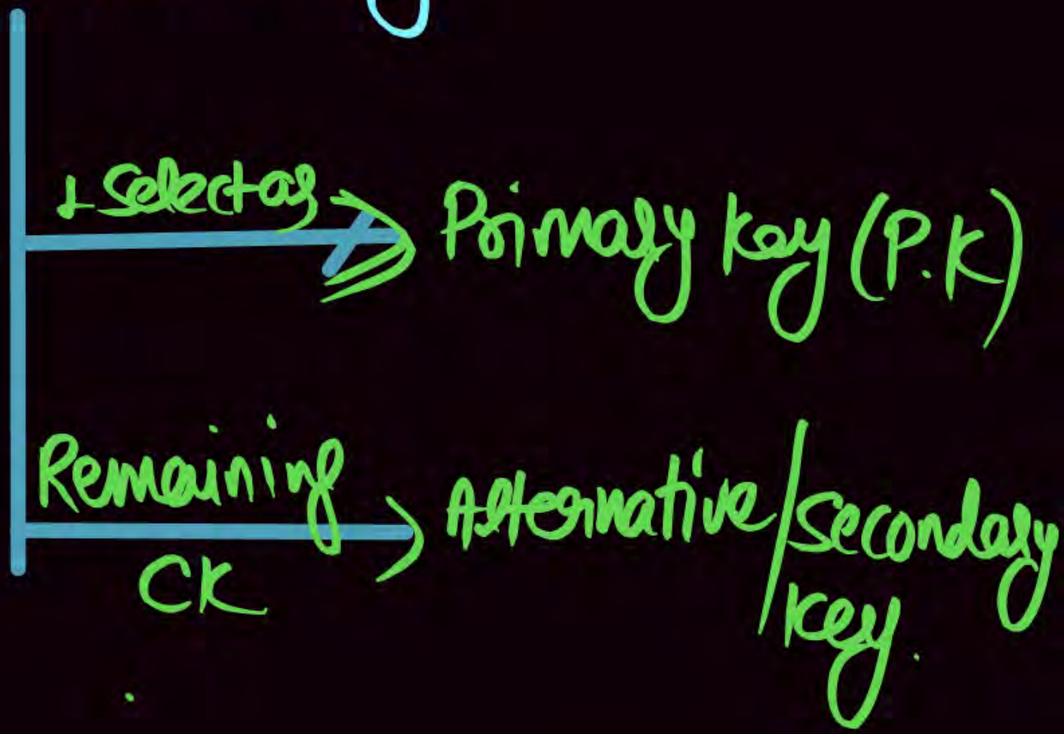
At Most one Primary key per Relation (Table)

Primary key

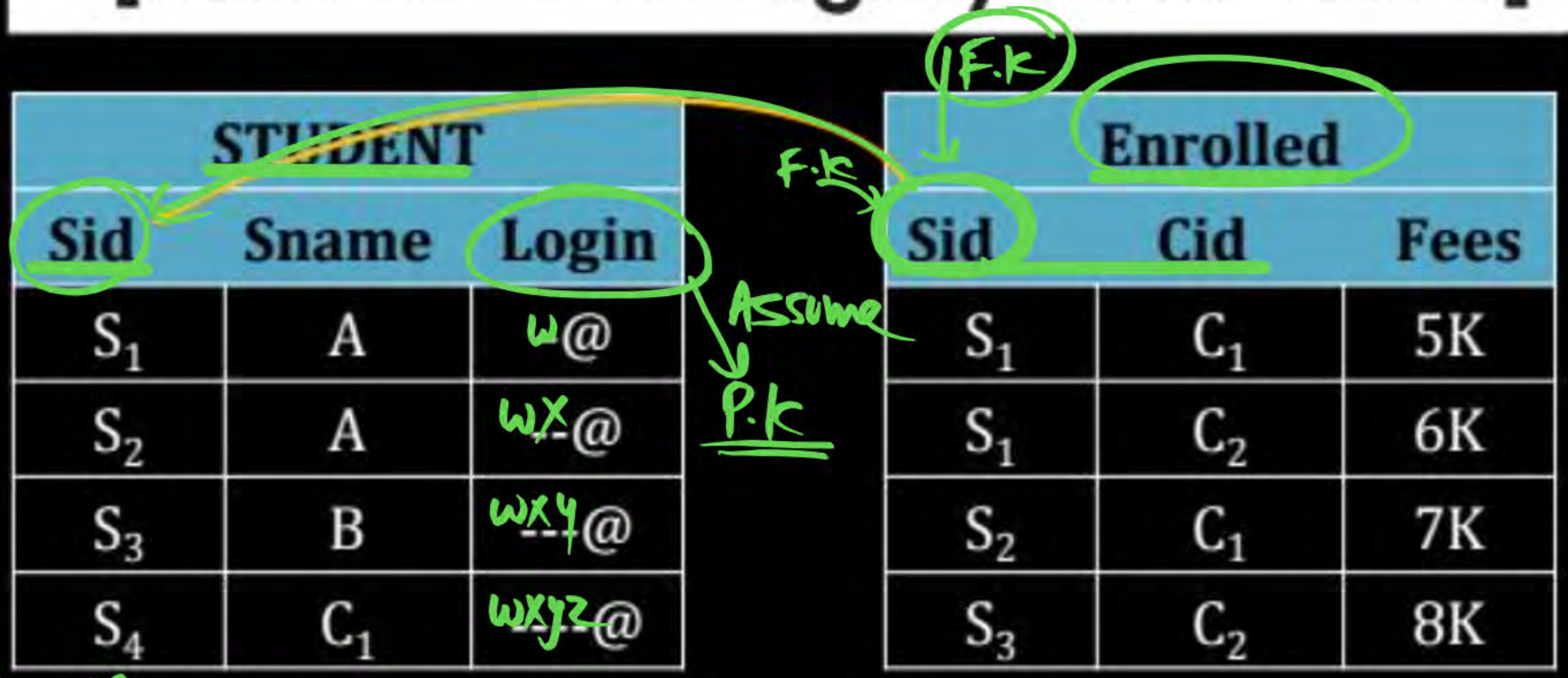
⇓

(Unique + NOT NULL)

Assume 6 Candidate key



# Foreign Key Constraint
# [Referential Integrity Constraint]




**STUDENT**

| Sid | Sname | Login |
|---|---|---|
| $S_1$ | A | w@ |
| $S_2$ | A | wx@ |
| $S_3$ | B | wxy@ |
| $S_4$ | $C_1$ | wxyz@ |

**Enrolled**

| Sid | Cid | Fees |
|---|---|---|
| $S_1$ | $C_1$ | 5K |
| $S_1$ | $C_2$ | 6K |
| $S_2$ | $C_1$ | 7K |
| $S_3$ | $C_2$ | 8K |

Sid of Enrolled table is the foreign key referencing to the primary key of student table.

Assume: Alternative key

[Sid: Primary Key]

Referenced Relation (Parent)

[Sid, Cid: Primary Key]

Referencing Relation (CHILD)

Note By Default foreign key Reference to the Primary key of Referenced Relation

Note At Most One Primary Per Table (Relation) → so Directly write Name of table.

```
Foreign key (Sid) Reference Student
```

---

Note Assume Login id is Primary key & Sid is Alternative key of Student Table.

```
Foreign key (Sid) Reference STUDENT(Sid)
```

when sid is Alternative key
∵ many Ak Possible
So Write Attribute Name

## CREATE TABLE ENROLLED

Sid Varchar (10)

Cid Varchar (10)

Fees Integer (11)

Primary key (Sid Cid)

Foreign Key (Sid) Reference Student → By Default foreign key Reference to Primary key of Referenced Relation.

When Sid is the primary key of Student

---

Let login is primary key & Sid is alternative key then

Foreign Key (Sid) Reference Student (Sid) → When Sid is not primary key.

# RDBMS Constraints

Legal Instance

- Domain Constraint (Atomic)
- key Constraint (Uniqueness)
- Entity Integrity Constraint (P.K Not NULL)
- Referential Integrity Constraint (foreign key Concept.)

Referenced Relation
(Parent Table)

Referencing Relation
(Child Table) ← foreign key

Note

Some times Same Table Contain Primary key & foreign key Both (Different Name also) sometimes

Note

The Value Present in the foreign key Must be Present in the Primary key of Referenced Relation.

Foreign key May Contain Duplicates & NULL Values.

Q Consider a Relation R With Attribute A, B where A is the primary & B is the foreign key, Reference to Primary key 'A' on Same table. Which of the following is correct According to Constraints?

Which of the following Row (or) sequence Can be Successfully inserted into R? (legal Instance)

(a) (1,2) (2,3) (3,4) (4,5) (5,6) (6,7) (6,8) → 8 Not in P.K present

(b) (1,2) (2,3) (3,9) (4,5) (5,6) (6,7) (7,8) 9 & 8 Not Present in P.K

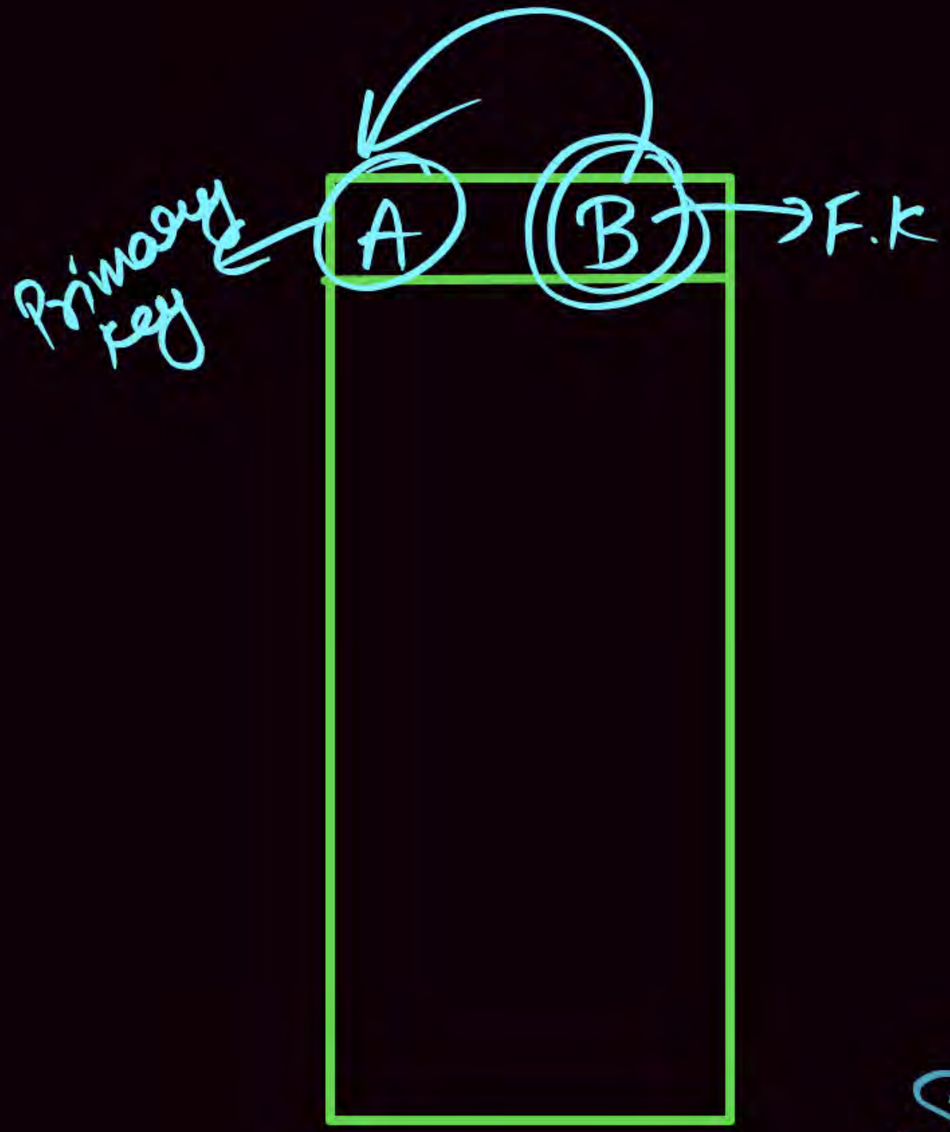
(c) (1,2) (2,3) (3,4) (NULL,2) (5,6) (6,7) (7,3) P.K Not NULL.

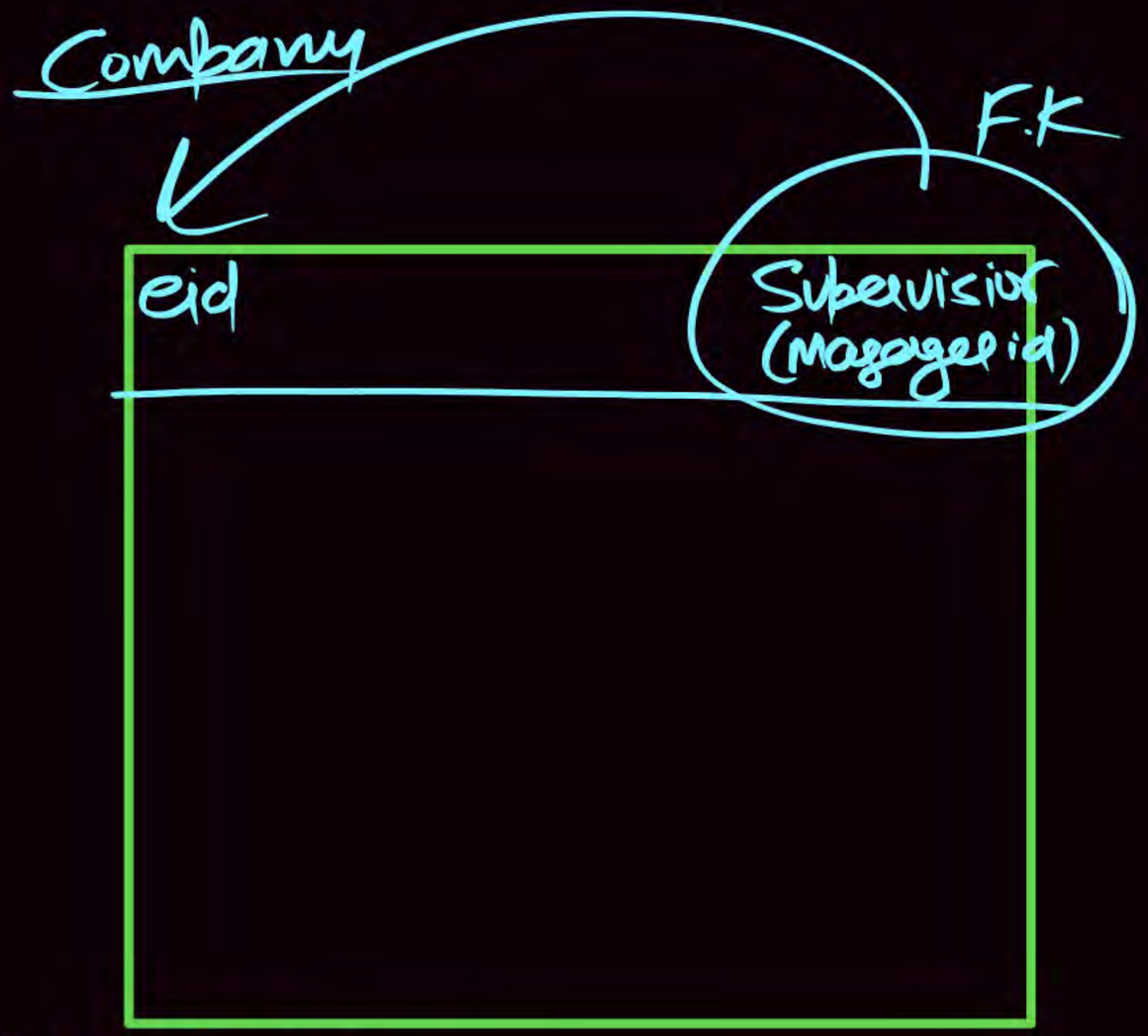
(d) (1,2) (2,3) (3,4) (4,5) (5,6) (2,7) (6,1) P.K unique & 7 Not Present in P.K

(e) (1,2) (2,3) (3,NULL) (4,2) (5,3) (6,5) (7,4)


## Constraints

- PK Unique & NOT NULL
- P.K Not Contain NULL Value
- The Value Present in F.K Must be Present in Primary key of Referenced Relation
- F.K May Contain Duplicates & NULL Value.




Same table Wallah with Diffeerent Name Concept

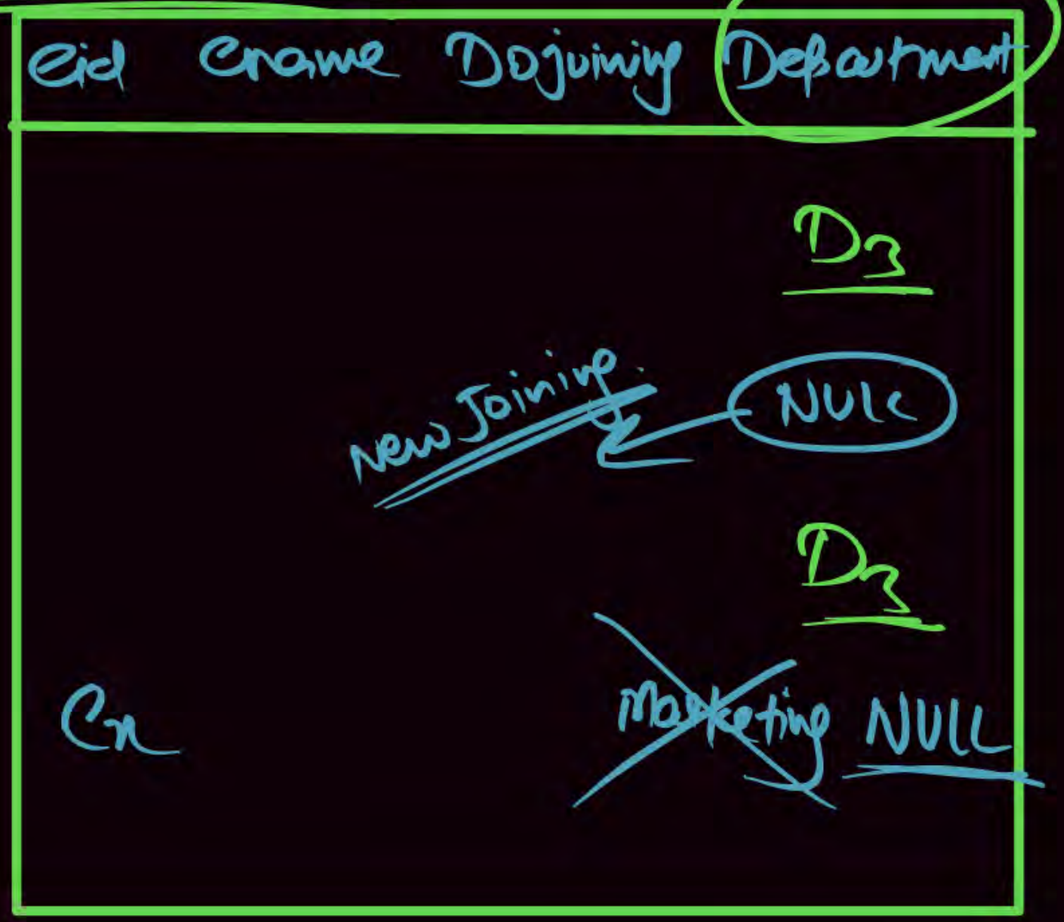
Note Concept: The value present in foreign key must be present in Primary key of Referenced Relation.

Company

(1) Employee Detail

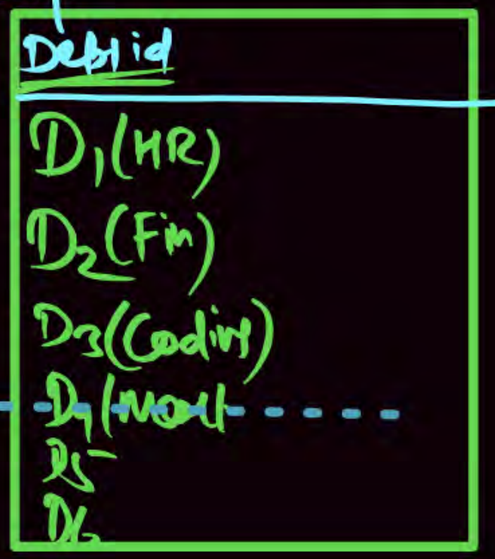
(2) Department

eg.




P.K

Referenced Relation [NULL ← Duplicate]



# Foreign Key Constraint
# [Referential Integrity Constraint]

| STUDENT | | |
|---|---|---|
| Roll No | Name | Branch |
| 1 | A | CSE |
| 2 | B | IT |
| 3 | C | CSE |

Pk →

Referenced Relation
(Parent)

| Registration | | |
|---|---|---|
| CNo | Cname | Roll No |
| 101 | DBMS | 1 |
| 102 | OS | 1 |
| 103 | CD | 3 |
| 104 | TOC | - |

← F.K

Referencing Relation
(CHILD)

STUDENT Table

↓

Referenced (Parent) Table

CHILD Table

↓ foreign key

Referencing (CHILD) Table

✓ Insert <4, D, IT>

✗ Delete <1, A, CSE>

✗ Insert <105 CE 6>

✓ Delete <103, CD 3>

**Note:** The value present in Foreign key must be Present in Primary key of Referenced relation

Foreign key may contain duplicate & NULL values.

| Parent table | CHILD table |
| --- | --- |
| Referenced table | Referencing Relation |
| ✓ Insert < 4 D ECE> | ✗ Insert < 105 DSA 67> |
| ✗ Delete < 1 A CSE> | ✓ Delete < 103 CD 3> |

**Note:** Deletion from the Referenced Relation and Insertion into Referencing Relation may violate Foreign key constraint.

**Note:** A Relation can Act as Parent & CHILD i.e. Relation may contain a primary key & a Foreign key that Refer to the same Relation.

# Referential Integrity Constraint

## (1) Referenced Relation

(i) Insertion : No Violation

(ii) Deletion : May cause Violation if Primary key is used by referencing relation

I. ON DELETE NO ACTION.

II. ON DELETE CASCADE.

III. ON DELETE SET NULL.

V. Imp

Foreign key (Sid) Reference STUDENT.

OR

Foreign key (Sid) Reference STUDENT (Sid)
ON DELETE NO ACTION.

Foreign key (Sid) Reference STUDENT (Sid)
ON DELETE CASCADE.

Foreign key (Sid) Reference STUDENT (Sid)
ON DELETE SET NULL.

# Referenced Relation

1. ON DELETE NO ACTION
2. ON DELETE CASCADE
3. ON DELETE Set NULL

Note Concept: The value present in foreign key must be present in Primary key of Referenced Relation.

(1) Employee Detail

(eg.)

Company

F.K

| Eid | Ename | DOjoining | Department |
|---|---|---|---|
| | | | $D_3$ |
| | | New Joining: | NULL |
| | | | $D_2$ |
| $C_n$ | | | ~~Marketing~~ NULL |

(2) Department

P.K

Referenced Relation [NULL ← Duplicate]

| Dept id |
|---|
| $D_1$ (HR) |
| $D_2$ (Fin) |
| $D_3$ (Coding) |
| $D_4$ (Mgmt) |
| $D_5$ |
| $D_6$ |

# Foreign key

Referenced Relation

1. Insertion : No violation
2. Deletion : [May cause violation]

(a) On delete no action : Means if it cause problem on delete then deletion is not allowed on table.

(b) On delete cascade : If we want to delete primary key value from referenced table then it will delete that value from referencing table also.

(c) On delete set null : If we want to delete primary key value from referenced table then it will try to set the null values in place of that value in referencing table.

**NOTE:**

If foreign key field is not null attribute then “On delete set null” is same as “on delete no action.”

**ON DELETE CASCADE** : Whenever Primary key Deleted then Corresponding Tuple (value) from the Referencing Relation Deleted Cascadely.

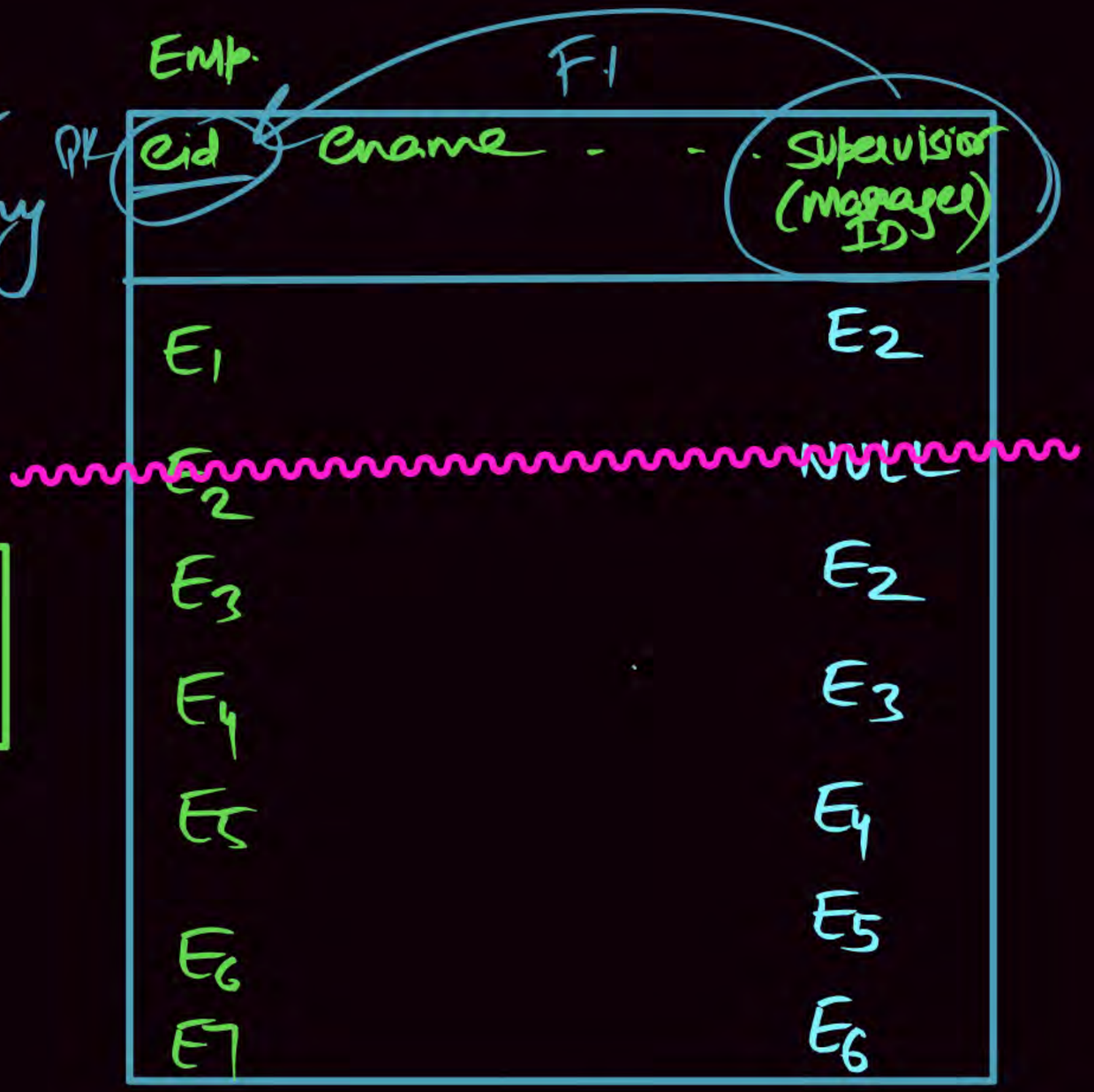
Note for good DB Design Sometimes On Delete Cascade Not Suggestable

Bcz like in Previous example Deletion of One Tuple (on Delete Cascade) Result leads to Deletion of Complete Table.

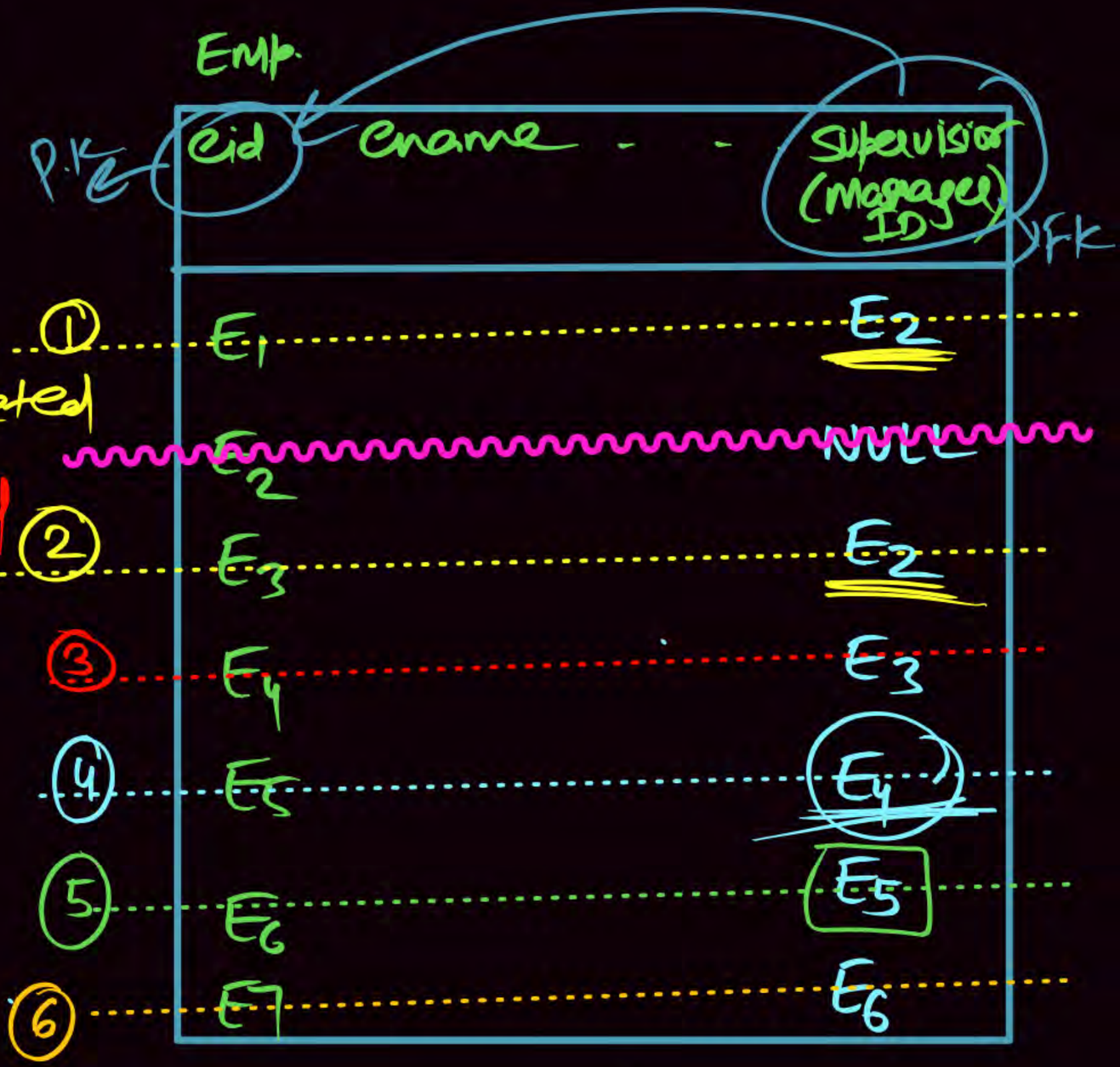
If we want to Delect $(E_2, NULL)$ with On Delete Cascade then How Many Number of additional Deleted to Preserve Referential Integrity ?

```
foreign key (mgr id) Reference Emp
ON DELETE CASCADE
```

Sol^n ALL Tuple Deleted



# Foreign key (Mgrid) Reference Emp ON DELETE CASCADE




(Q) ($E_2$ NULL) P.k $E_2$ Deleted

Due to P.k $E_2 \Rightarrow$ [$E_1$ $E_2$] [$E_3$ $E_2$] Deleted

Now Primary key $E_1$ & $E_3$ Deleted

Due to P.k $E_3 \Rightarrow$ [$E_4$, $E_3$] Deleted

Now Primary key $E_4$ Deleted

Due to P.k $E_4 \Rightarrow$ [$E_5$, $E_4$] Deleted

Now Primary key $E_5$ Deleted

Due to P.k $E_5 \Rightarrow$ [$E_6$, $E_5$] Deleted

Now Primary key $E_6$ Deleted

Due to PK $E_6 \Rightarrow$ [$E_7$, $E_6$] Deleted

# Foreign key

3. Updation : [May cause violation]
   (a) On update no action
   (b) On update cascade
   (c) On update set null

Referencing Relation

1. Insertion : [May cause violation]
2. Deletion : No violation
3. Updation : [May cause violation]

**NOTE:**

If integrity violation occurs because of insertion or updation in referencing table then restrict insertion and updation.

# Example

P.K F.K

| A | B |
|---|---|
| 2 | 4 |
| 3 | 4 |
| 4 | 5 |
| 5 | 4 |
| 6 | 2 |

B is foreign key
Referencing A,
Delete (2, 4) and
on delete cascade

| A | B |
|---|---|
| 3 | 4 |
| 4 | 5 |
| 5 | 4 |

Result

So, If we delete (2, 4) then PK "2". gets deleted from the table and all the tuples in which B is referencing PK.2" also gets deleted.

Q.

The following table has two attributes A and C where A is the primary key and C is the foreign key referencing A with on-delete cascade.

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple (2, 4) is deleted is:

A. (3 4) and (6, 4)

B. (5, 2) and (7, 2)

C. (5, 2), (7, 2) and (9, 5)

D. 1

| A | C |
|---|---|
| 2 | 4 |
| 3 | 4 |
| 4 | 3 |
| 5 | 2 |
| 7 | 2 |
| 9 | 5 |
| 6 | 4 |

P.k : 2 Deleted

Due to Pk 2

(5,2) (7,2) Deleted

Now Primary key 5,7 Delete

Due to P.k 5 ⇒ (9,5) Deleted

Now Pk '9' also Deleted

But P.k : 7 & 9 Not in F.k

So 3 Tuple Deleted

Q.

Let R (a, b, c) and s(d, e, f) be two relation in which d is the foreign key of S that refers to the primary key of R. Consider the following four operations on R and S.

(i) Insert into R

(ii) Insert into S

(iii) Delete from R

(iv) Delete from S

Which of the following is true about the referential integrity constraint above?

A. None of (i), (ii), (iii), or (iv) can cause its violation

B. All of (i), (ii), (iii), and (iv) can cause its violation

C. Both (i) and (iv) can cause its violation

D. Both (ii) and (iii) can cause its violation

# MCQ

Suppose $R_1$(A, B) and $R_2$(C, D) are two relation schemes. Let $r_1$ and $r_2$ be the corresponding relation instances. B is a foreign key that refers to C in $R_2$. If data in $r_1$ and $r_2$ satisfy referential integrity constraints, which of the following is ALWAYS TRUE?

[GATE-2013-CS: 2M]

A $\Pi_B(r_1)-\Pi_C(r_2)=\phi$

B $\Pi_C(r_2)-\Pi_B(r_1)=\phi$

C $\Pi_B(r_1)=\Pi_C(r_2)$

D $\Pi_B(r_1)-\Pi_C(r_2)\neq\phi$

# NAT

Consider the following tables T1 and T2.

In table T1, **P** is the primary key and **Q** is the foreign key referencing **R** in table T2 with on-delete cascade and on-update cascade. In table $T_2$, **R** is the primary key and **S** is the foreign key referencing **P** in table T1 with on-delete set NULL and on-update cascade. In order to delete record ⟨3,8⟩ from table T1, the number of additional records that need to be deleted from table T1 is

| $T_1$ | |
|---|---|
| **P** | **Q** |
| 2 | 2 |
| 3 | 8 |
| 7 | 3 |
| 5 | 8 |
| 6 | 9 |
| 8 | 5 |
| 9 | 8 |

| $T_2$ | |
|---|---|
| **R** | **S** |
| 2 | 2 |
| 8 | 3 |
| 3 | 2 |
| 9 | 7 |
| 5 | 7 |
| 7 | 2 |

[GATE-2017-CS: 1M]

## MCQ

Given the basic ER and relational models, which of the following is INCORRECT?

[GATE-2012-CS: 1M]

A. An attribute of an entity can have more than one value.

B. An attribute of an entity can be composite.

C. In a row of a relational table, an attribute can have more than one value.

D. In a row of a relational table, an attribute can have exactly one value or a NULL value.

Q.

The term in list A have been mapped to list B so that is corresponds to the mapping process of ER MODEL into relational. Which of the following represent the mapping process? [MCQ]

| List-A | List-B |
|---|---|
| A. Entity type | 1. Primary key (or alternate key) |
| B. Key attributes | 2. Child table |
| C. Composite attribute | 3. Set of simple component attributes |
| D. Multivalued attribute | 4. Relation |

A. A-3, B-1, C-4, D-2

B. A-4, B-1, C-3, D-2

C. A-3, B-2, C-2, D-4

D. A-4, B-1, C-2, D-3

# Mapping [Cardinality constraints of relationship set]

(For binary relationship)

## Partial participation on both side of binary relationship

- One to Many : Merge relationship set towards many side. So, 2 relational tables.
- Many to one : Merge relationship set towards many side. So, 2 relational tables.
- One to one : Merge relationship set any one side. So, 2 relational tables.
- Many to Many : Separate table for each entity set and relationship set. so, 3 relational tables.

# Mapping [Cardinality constraints of relationship set]

(For binary relationship)

## Full participation on “one” side of many to one relationship

Merge the entities and relationship set into single relational table. So, 1 table.

# Mapping [Cardinality constraints of relationship set]

(For binary relationship)

## Full participation on “Many” side of Many-to-one relationship

Merge relationship set towards many side. So, 2 relational tables.

# Mapping [Cardinality constraints of relationship set]

(For binary relationship)

## Full participation on any “one” side in one-to-one relationship

Merge the entity sets and relationship set into single table. So, 1 table.

# Mapping [Cardinality constraints of relationship set]

(For binary relationship)

## Full participation on any "Many" side of Many-to-Many relationship

Merge relationship set towards any "Many" side of relationship. So, 2 table.

# Mapping [Cardinality constraints of relationship set]

(For binary relationship)

## Full participation on both side of relationship

1 : 1
1 : M
M : 1
M : N

Merge the entity sets and Relationship into single Relational table so, 1 relational table.

# Self-Referential Relationship Set

(Recursive entity set)

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


Recursive
Entity set

Self-referential relationship set

- Binary relationship set
- 1: M, M:1, 1:1, M:N can be any of the relationship

Q.

Consider the following entity relationship diagram(ERD), where two entities E1 and E2 have a relation R of cardinality 1:m


The attributes of E1 are A11, A12 and A13 where A11 is the key attribute. The attributes of E2 are A21, A22, A23 where A21 is the key attribute and A23 is a multi-valued attribute. Relation R does not have any attribute. A relational database containing minimum number of tables with each tables satisfying the requirements of the third normal from (3NF) is designed form the above ERD. The number of tables in the database is

[GATE-2004 : 2 Marks]

A 2

B 3

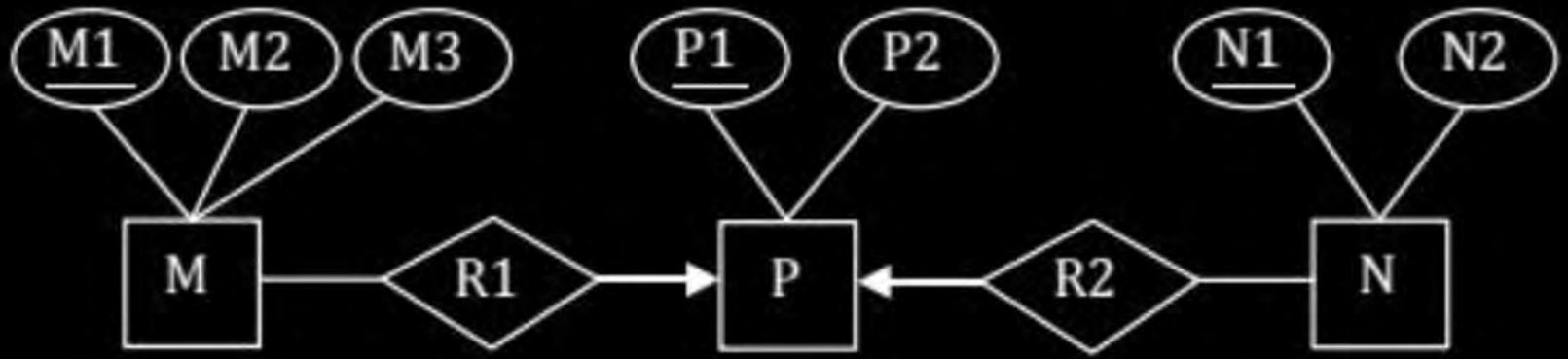
C 5

D 4

Q.

## Common Data for Question

Consider the following ER Diagram




(i) The minimum number of tables needed to represent M, N P, R1, R2 is [GATE-2008 : 2 Marks]

A 2

B 3

C 4

D 5

(ii) Which of the following is a correct attribute set for one of the table for the correct answer to the above question?

GATE-2008 : 2 Marks]

A {M1, M2, M3, P1}

B {M1, P1, N1, N2}

C {M1, P1, N1}

D {M1, P1}

## E-R Model:

Entity Relationship Diagrams used to represent Diagrammatic design [High Level Design] of Databases.

## DB Design Steps:

1. Requirement: What type of data stored and what operation required etc.
2. Conceptual and Logical: [ER Diagram]
3. ER Diagram to RDBMS table design and apply normalization.
4. Physical DB Design (Indexing Design)
5. User interface Design && Security Design

High Level Design

Low Level Design

# Main Components in ERD:

1. Attributes
2. Entity Sets
3. Relationship Sets

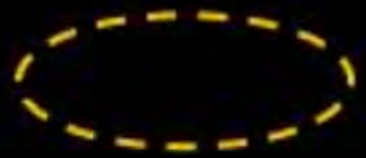
# Attribute

Attributes:

Key Attributes:

Multi valued Attribute:

Derived Attribute:

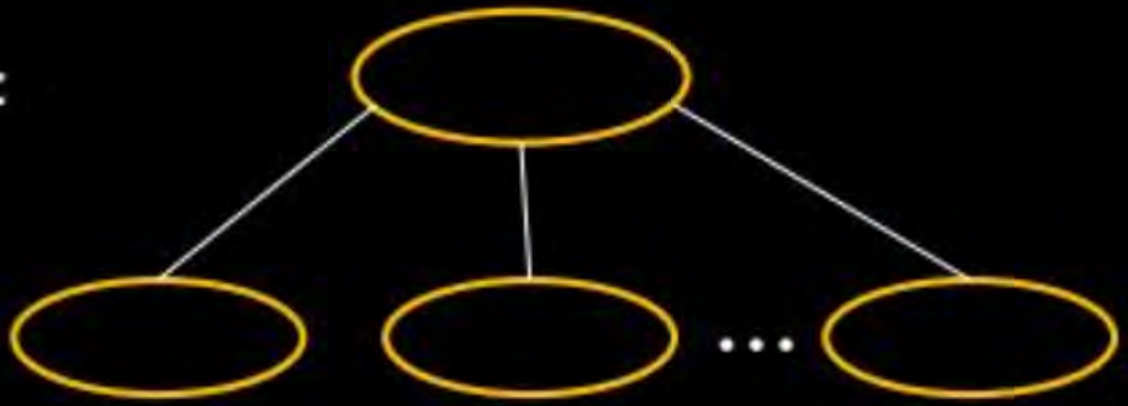


Value derived from
Other stored value

Example:

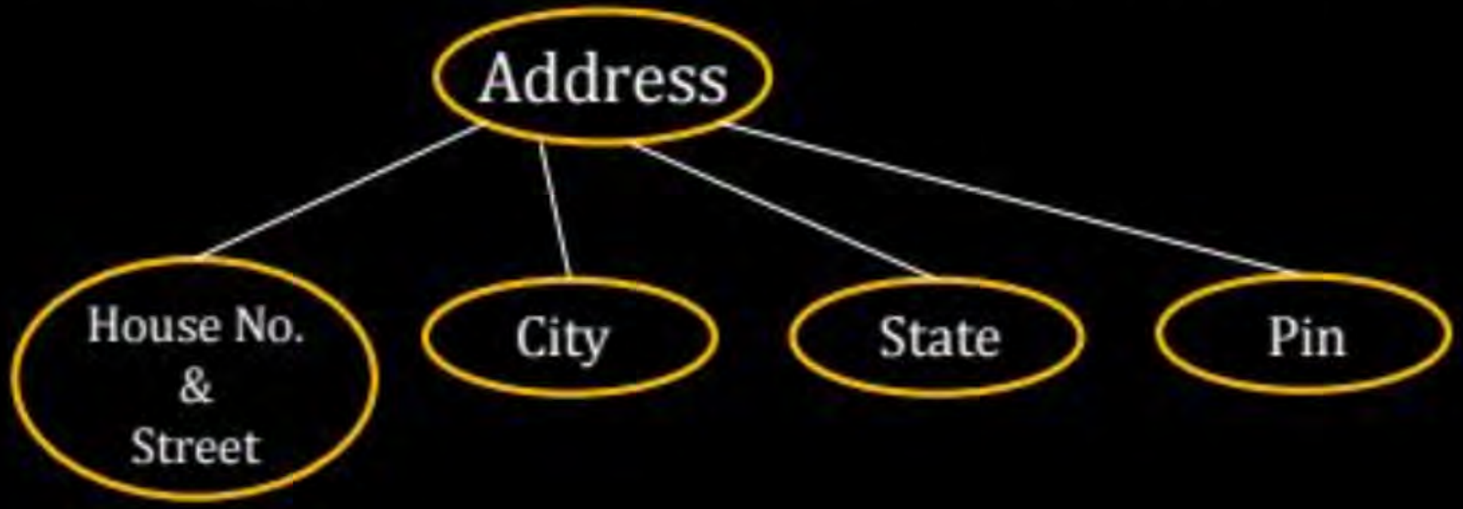
# Attribute

Composite Attribute:



Attribute Which can represent as two or more attributes

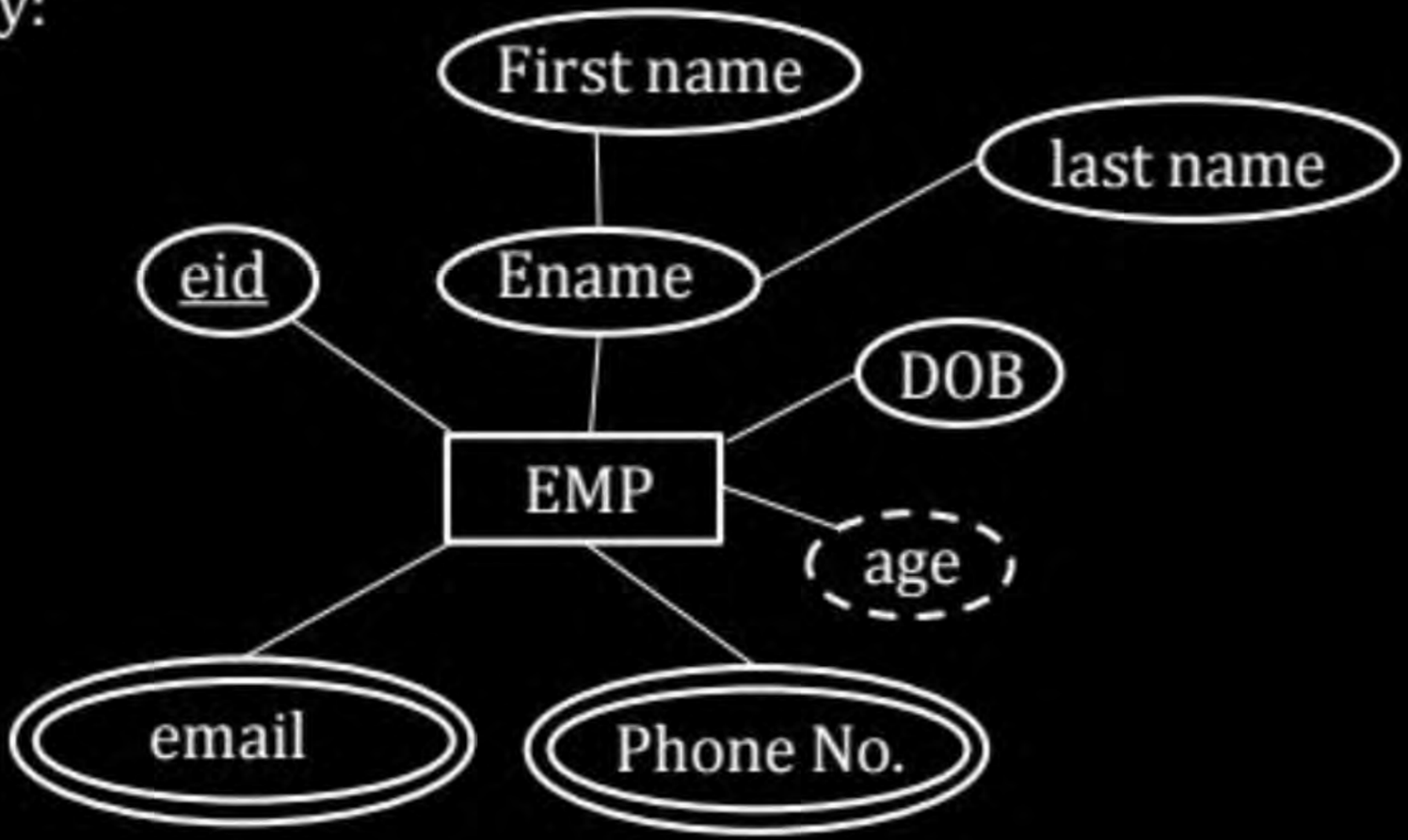
Example:



# Entity Set:

Set of similar entities (recorded/objects).

Represented by:



# Relationship Set:

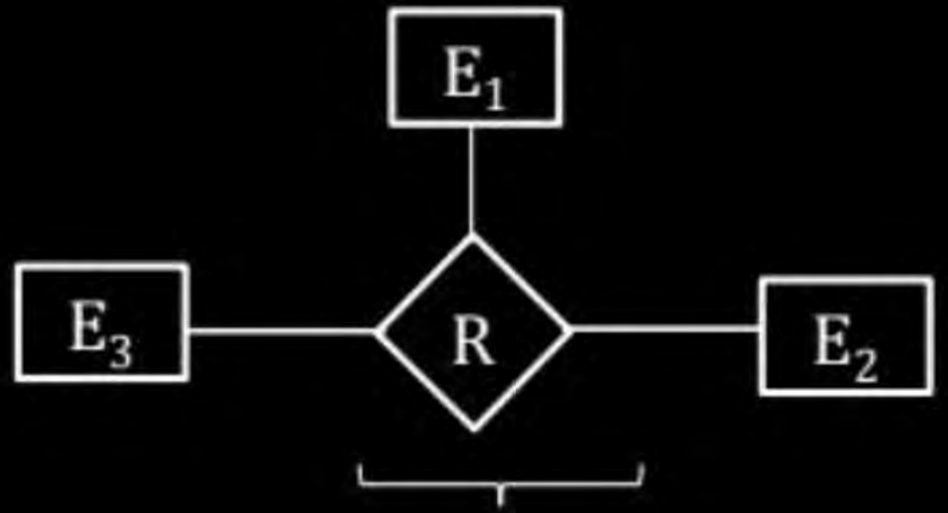
Used to relate two or more entity set.

Represented by:

1.


Binary Relationship

2.



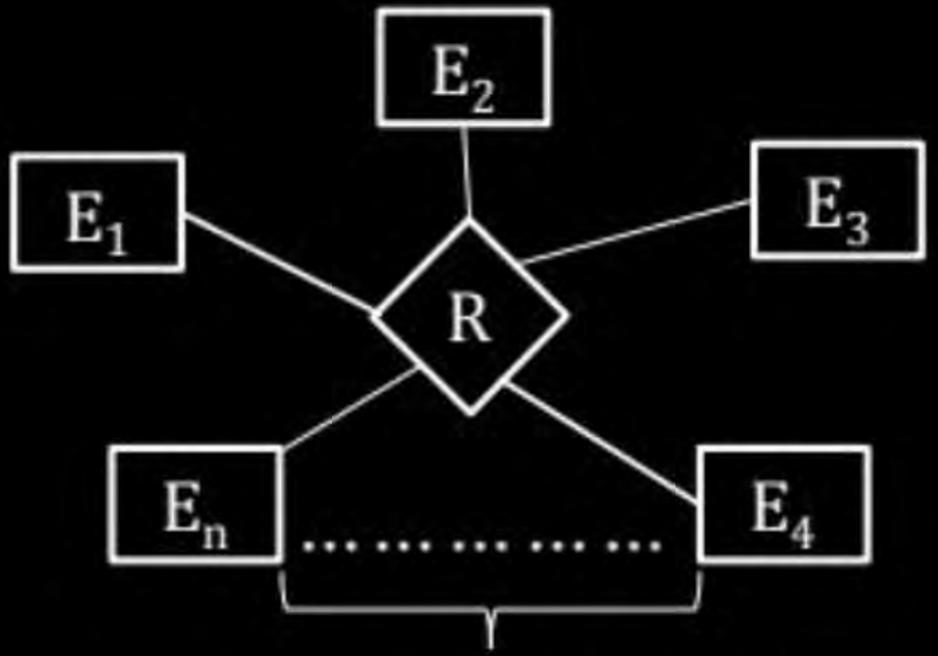

Ternary Relationship

# Relationship Set:

Used to relate two or more entity set.

Represented by:

3.




N-ary Relationship

# Participation:

If every entity of entity set (E1) must participate with relationship set than Total Participation.

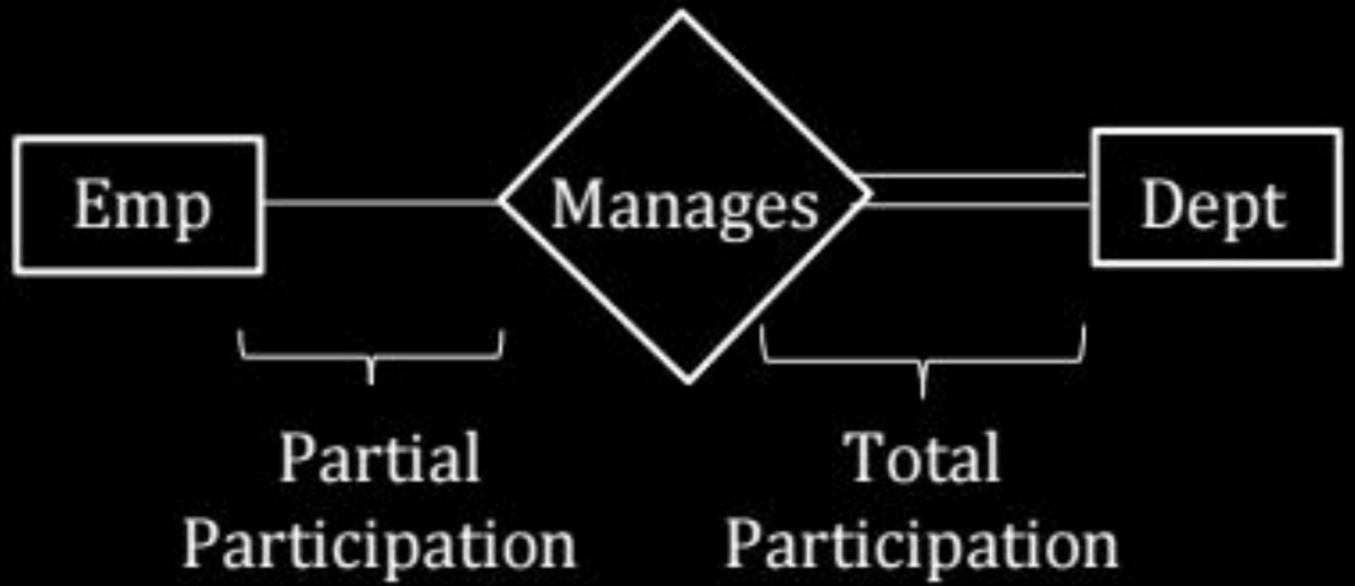
[must be 100% participation]

Otherwise:

Partial Participation:

May or may not 100% participation

Example: "Emp" & "Dept" Entity Set.
"Manages" Relationship Set such that each dept there must be a manager.



# Mapping:

Mapping [cardinality of Relationship set]:

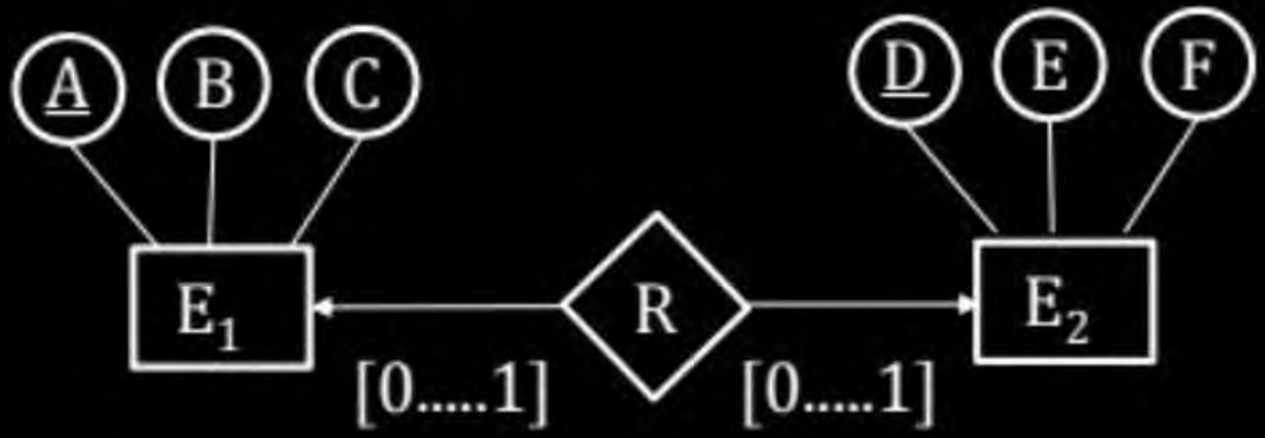
One: [At most one] [0.....1]
Many: [0 or more] [0.....M]

Possible Mapping of binary relationship sets are:

1) One : One
2) One : Many
3) Many : One
4) Many : Many

Candidate key's of Relationship set is Based on mapping

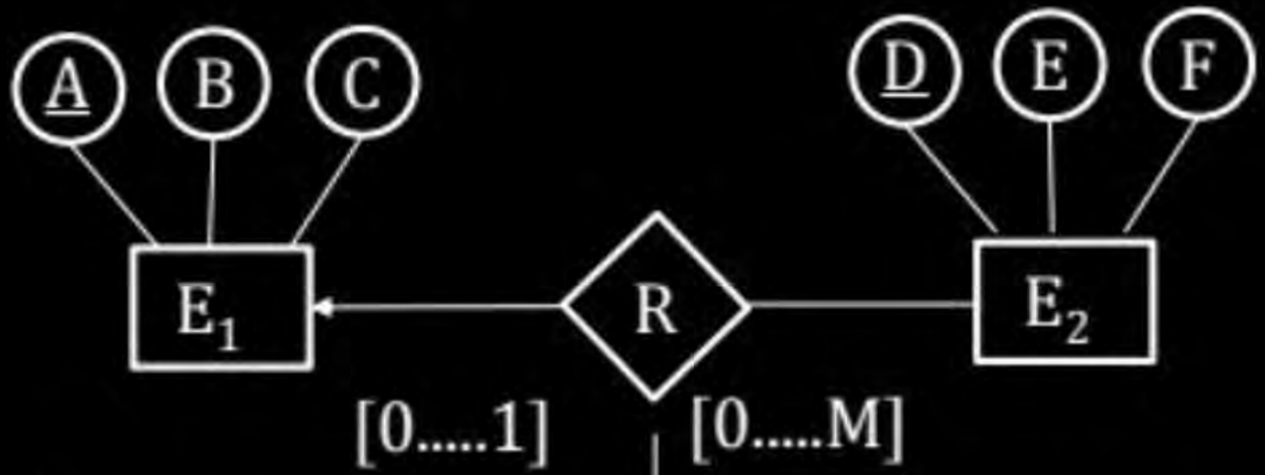
## One to One:




R (A D) ≡ {A, D}

| | | |
|---|---|---|
| $a_1$ | $d_1$ | 1 : 1 mapping |
| $a_2$ | $d_2$ | candidate key of |
| $a_3$ | $d_5$ | Relationship set |

**One: Many:**



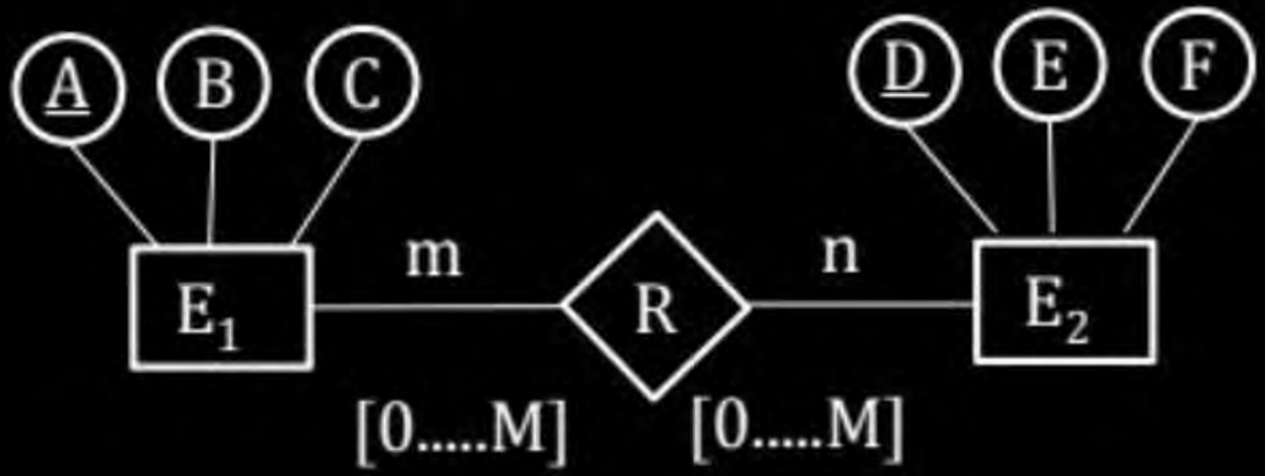

Each object of $E_2$ allowed to pair by at most one entity of $E_1$

Each entity of $E_1$ can pair with many $E_2$ entity.

Candidate key of R (A D) ≡ { D }

| | | |
|---|---|---|
| $a_1$ | $d_1$ | : for 1: M |
| $a_1$ | $d_2$ | Relationship |
| $a_2$ | $d_3$ | |

## Many to Many:

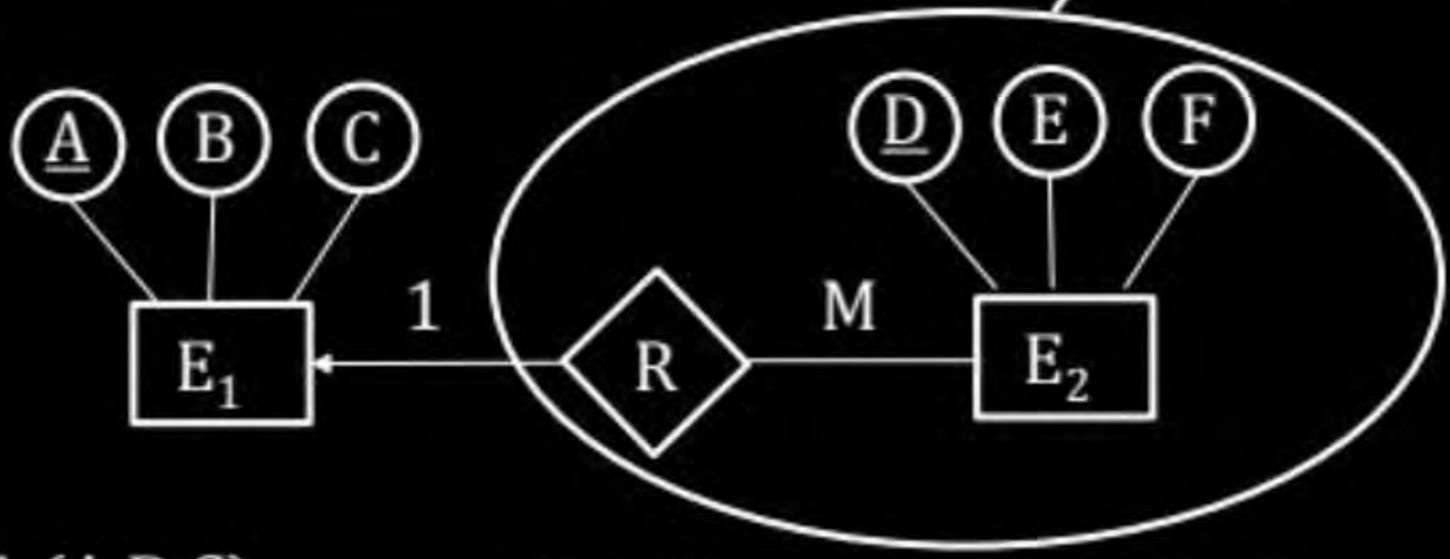



R (A D) ≡ {AD}

| | | |
|---|---|---|
| $a_1$ | $d_1$ | M:N ⇒ |
| $a_1$ | $d_2$ | for many to many mapping |
| $a_2$ | $d_2$ | Candidate Key for |
| $a_3$ | $d_3$ | Relationship Set is |
| | | {AD} |

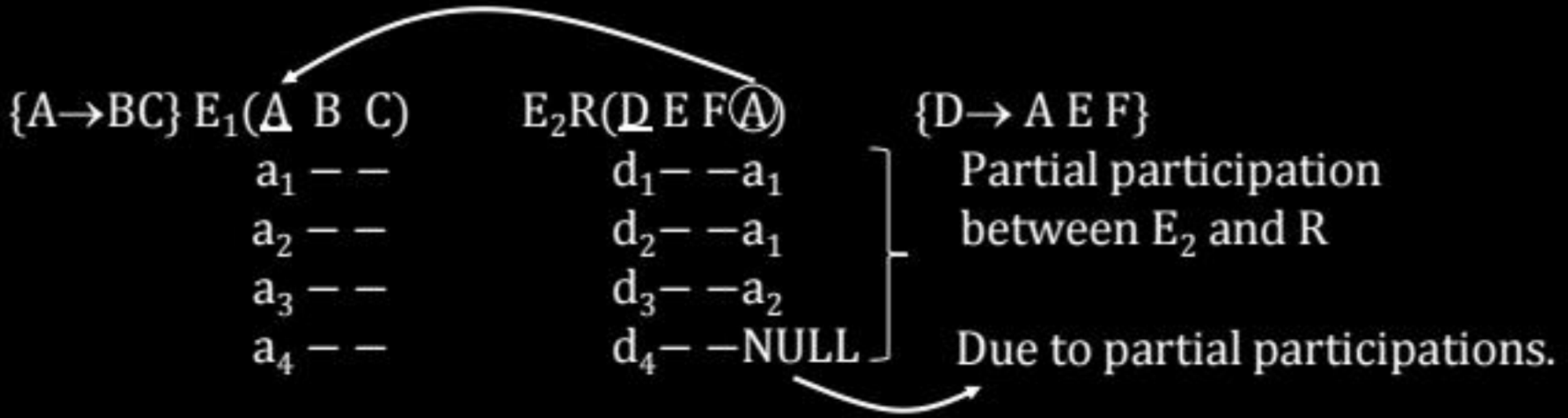
# RDBMS Design of Given ERD:

1 : M Relationship:



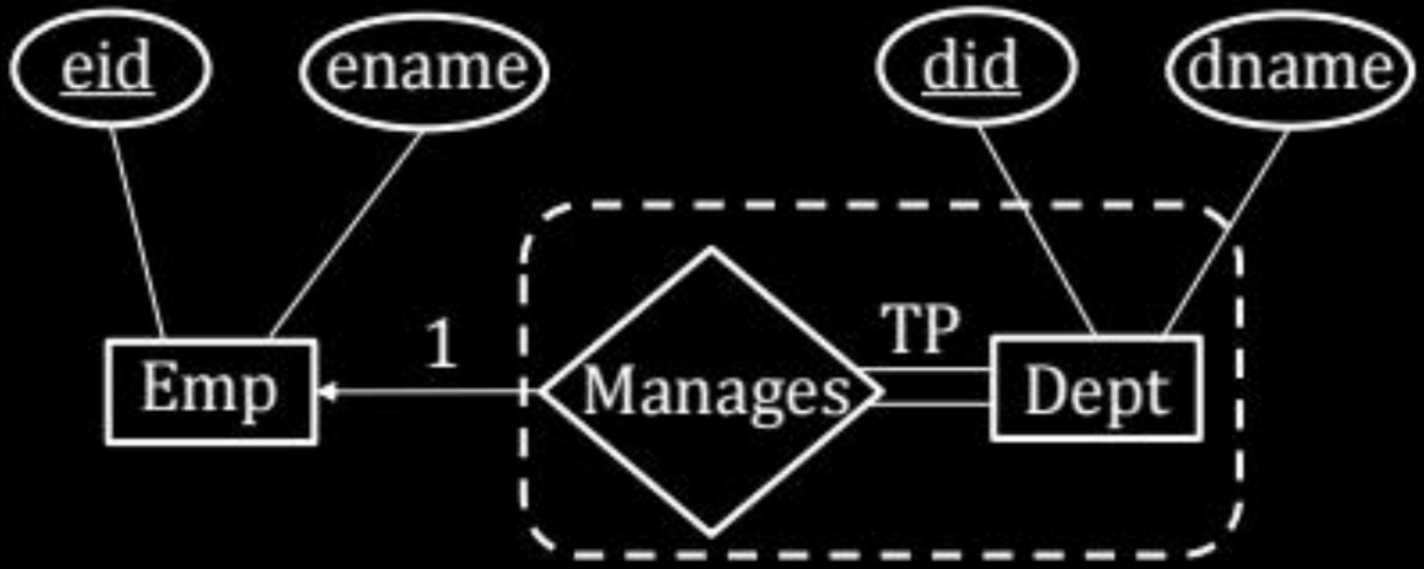

$E_1(\underline{A}\ B\ C)$ $R(A\ \underline{D})$ $E_2(\underline{D}\ E\ F)$

| | | |
|---|---|---|
| $a_1$ – – | $a_1\ d_1$ | $d_1$ – – |
| $a_2$ – – | $a_1\ d_2$ | $d_2$ – – |
| $a_3$ – – | $a_2\ d_3$ | $d_3$ – – |
| $a_4$ – – | | $d_4$ – – |
| $A \rightarrow BC$ | $D \rightarrow A$ | $D \rightarrow EF$ |
| $\{\underline{A}\}$ | $\{\underline{D}\}$ | $\{\underline{D}\}$ |




∴ minimum 2 relational Table Required and 1 Foreign Key required for given ERD

Given ERD
Example:



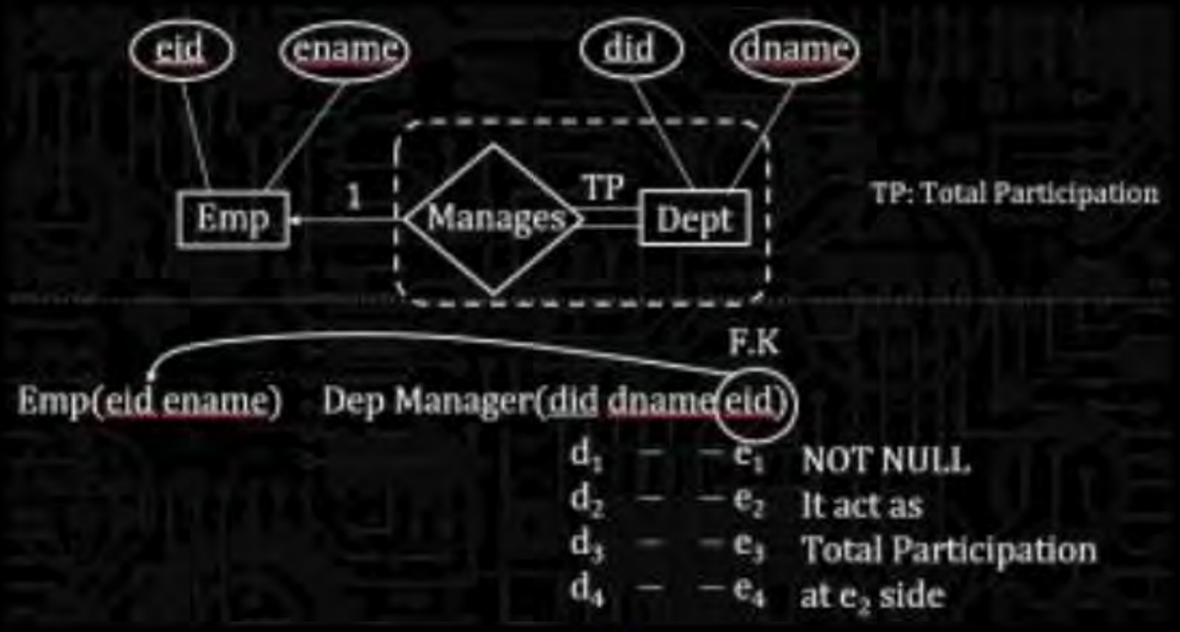

TP: Total Participation

F.K

Emp(eid ename)  Dep Manager(did dname eid)

| | | | |
|---|---|---|---|
| $d_1$ | – | – $e_1$ | NOT NULL |
| $d_2$ | – | – $e_2$ | It act as |
| $d_3$ | – | – $e_3$ | Total Participation |
| $d_4$ | – | – $e_4$ | at $e_2$ side |

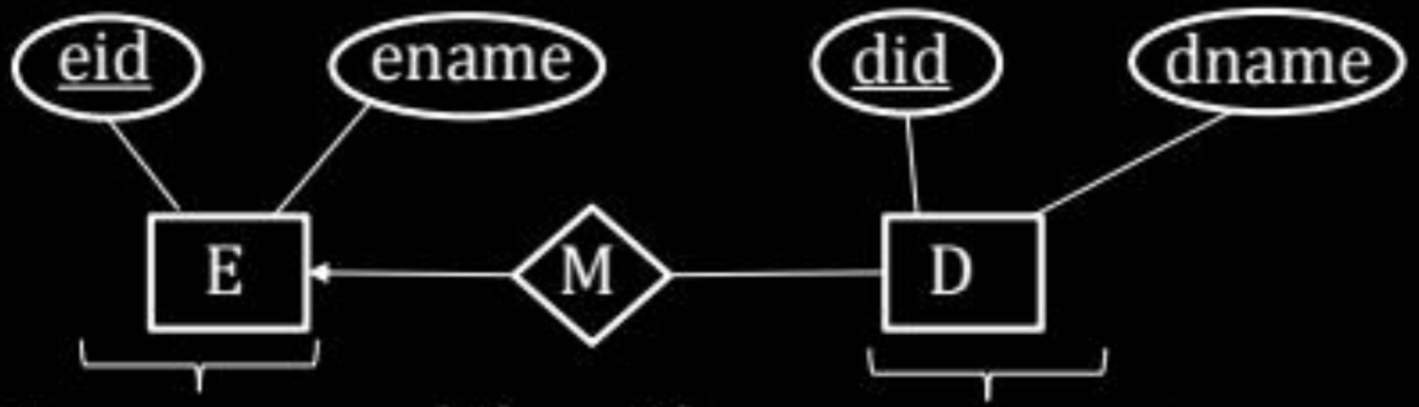
Given ERD




Emp_Dept_Manages
(did dname eid ename)
did → dname
did → eid
(_____ $e_5$ A)
Not allowed because $e_5$ not in relationship

CREATE TABLE Dept_manages
(did varchar(10) primary key,
dname varchar(30),
eid varchar(10) NOT NULL,
FOREIGN KEY(eid)
REFERENCES Emp);

If 1 : M relationship set merges into left side Entity set:




eid → ename  did → eid  did → dname

E_M(eid ename did)

D(did dname)

| | | |
|---|---|---|
| $e_1$ | A | $d_1$ |
| $e_1$ | A | $d_2$ |
| $e_1$ | B | $d_2$ |
| $e_1$ | C | $d_3$ |
| $e_1$ | D | Null |

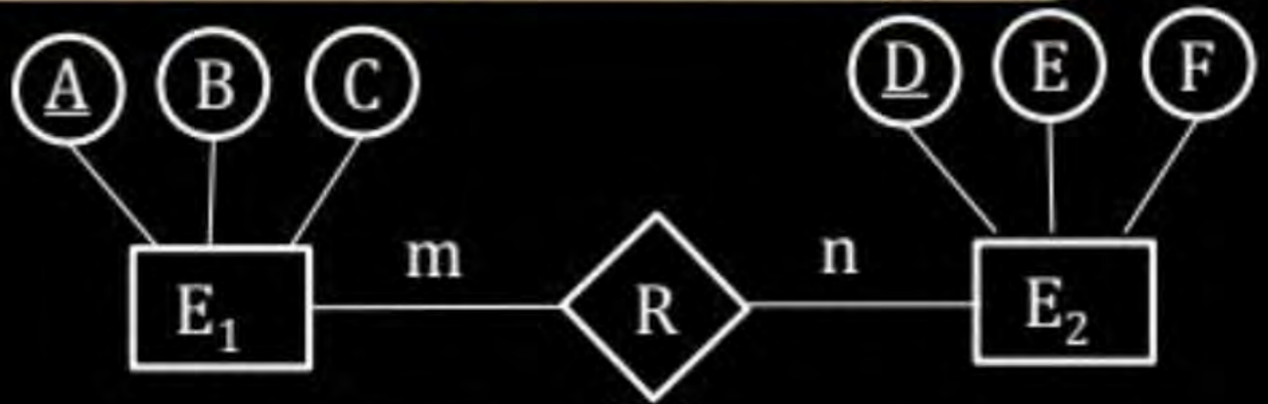
Not valid as did is key of E_M
(Partial Participation)
[lost at E side]

## Disadvantage:

1) Data Redundancy Occurs
   eid → ename
   Not Super Key(S.K)

2) Partial Participation will be lost
   [Not possible to insert employees who are not manager of dept.]

# Many : Many Relationship:




$E_1$ (A B C)

| A |
|---|
| $a_1$ |
| $a_2$ |
| $a_3$ |
| $a_4$ |
| ⇑ |
| key |

R (A D)

| A | D |
|---|---|
| $a_1$ | $d_1$ |
| $a_2$ | $d_2$ |
| $a_2$ | $d_2$ |
| ⇑ | ⇑ |
| Not key | |

$E_2$ (D E F)

| D |
|---|
| $d_1$ |
| $d_2$ |
| $d_3$ |
| ⇑ |
| key |

Minimum 3 relational table required and 2 foreign key required.

M:N relationship set not allowed to merge with any entity set.

# One : One Relationship:


$E_1$, $E_2$ entity sets & R relationship set related between $E_1$ and $E_2$ with 1:1 cardinality and partial participation(pp) at both side.

$E_1$ (A B C)

$a_1$
$a_2$
$a_3$
$a_4$
⇑
key
[A→BC]

R (A D)

$a_1$ $d_2$
$a_2$ $d_3$
$a_3$ $d_4$
⇑ ⇑
key key
[A→D
D→A]

$E_2$ (D E F)

$d_1$
$d_2$
$d_3$
$d_4$
⇑
key
[D→EF]

## If all merge with single Table:

$E_1RE_2$

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| $a_1$ | - | - | $d_2$ | - | - |
| $a_2$ | - | - | $d_3$ | - | - |
| $a_3$ | - | - | $d_4$ | - | - |
| $a_4$ | - | - | Null | Null | Null |
| Null | NULL | | $d_1$ | - | - |

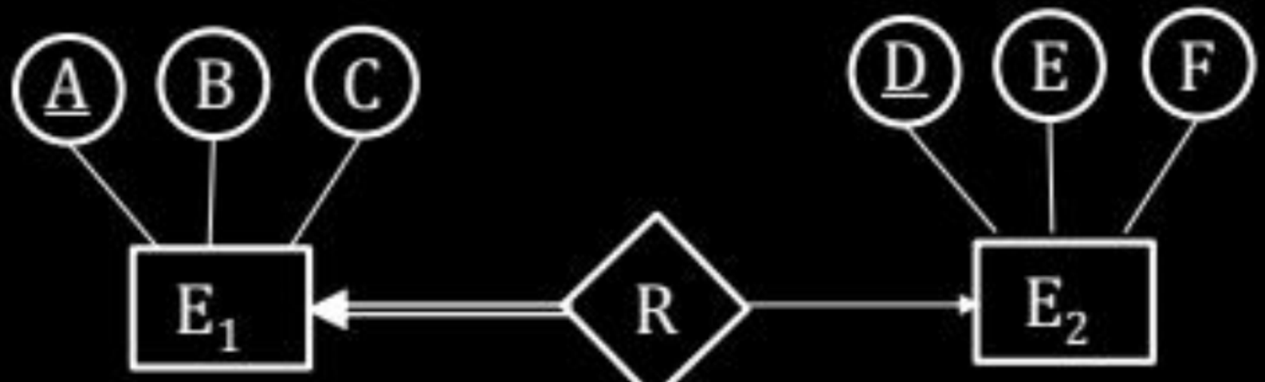
Candidate key = {A, D}

No Candidate key is present, & No primary key

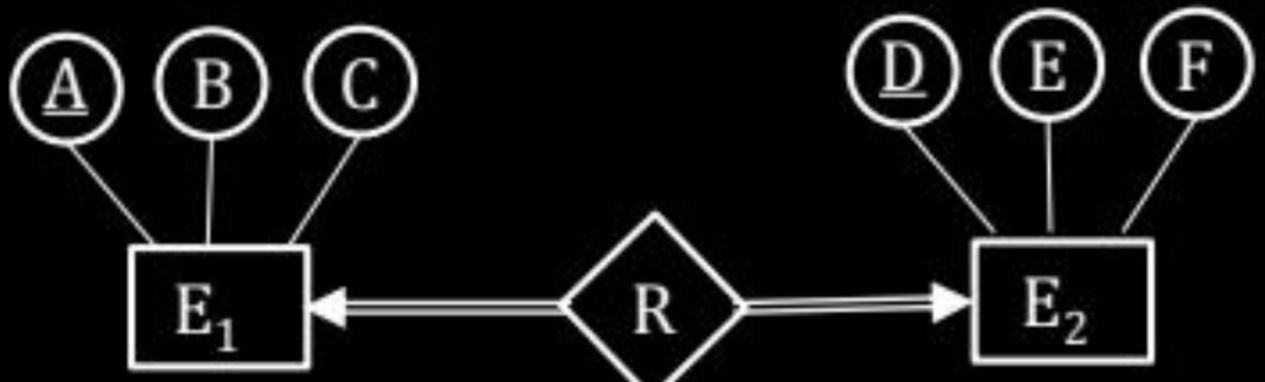
A relational table in which no attribute having "NOT NULL" values are "NOT" allowed in RDBMS . So,

- minimum 2 Relation Table Required and 1 Foreign key.

$E_1$ $E_2$ Entity set R Rel set between $E_1E_2$
With 1:1 mapping and at least one end having total participation.




$E_1$ R $E_2$ (A B C D E F)

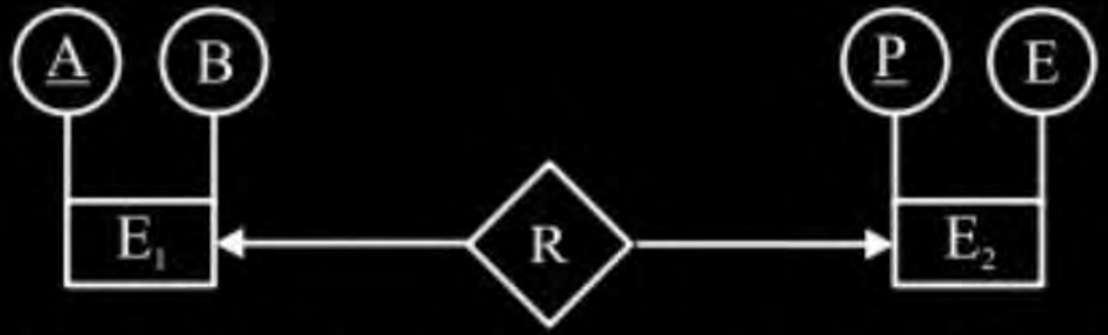



$E_1$ R $E_2$ (A B C D E F)

Primary Key

UNIQUE & NOT NULL (Alternative Key)

Q.




70% participation at $E_1$ end & 40% participation at $E_2$ end.
Which is best possible design?

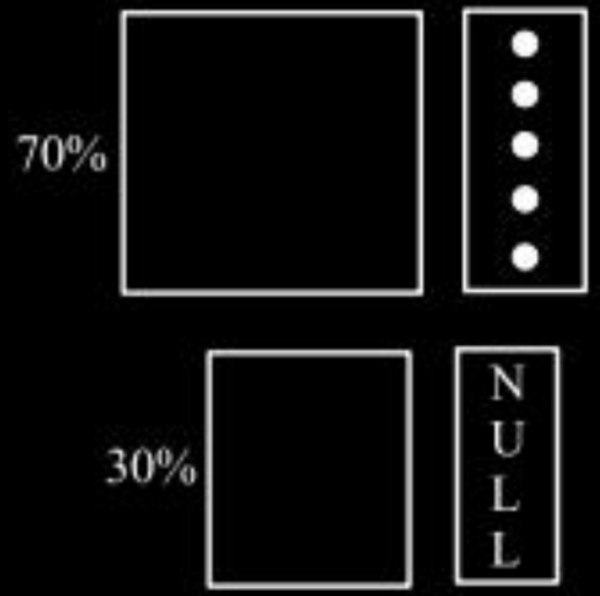
A. $E_1$ $E_2$ entity set kept separate with F.K at $E_1$

B. $E_1$ $E_2$ entity set kept separate with F.K at $E_2$

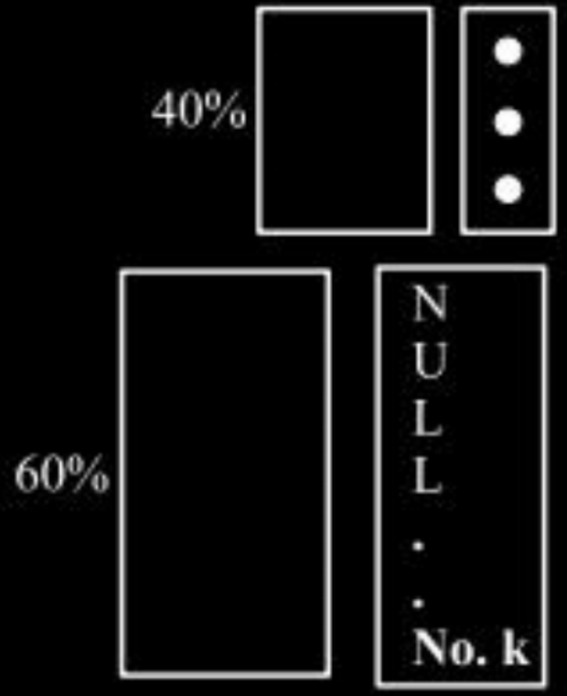
C. $E_1$ $E_2$ entity set kept separate with F.K at both $E_1$ $E_2$

D. $E_1$ $E_2$ merges into single table with No F.K.

(a) $E_1$ R (A B D) $E_2$ (D E)

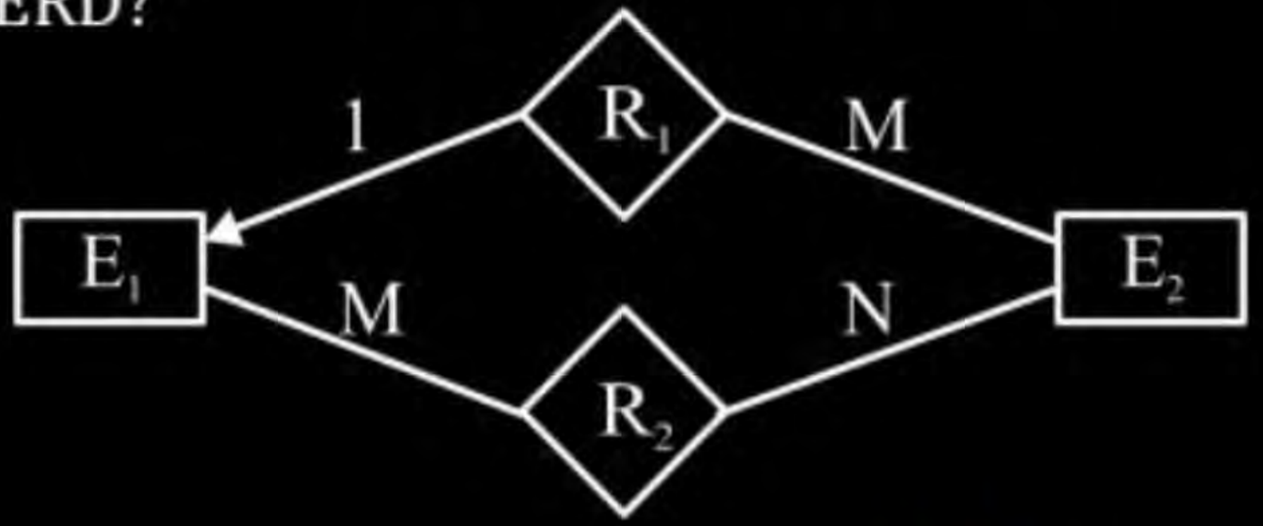



(b) E1(A, B) $E_2$ (D E A)



Q.

$E_1$ $E_2$ entity set $R_1$ $R_2$ Relationship set related between $E_1$ and $E_2$ with 1:M and M : N mapping min Relational Table required in ERD?




A. 2

B. 3

C. 4

D. 5

Q.

$E_1E_2E_3$ entity set $R_1$ Relation between $E_1$ $E_2$ with 1 : M, $R_2$ Relations between $E_2$ $E_3$ with M : 1, $R_3$ Relationship between $E_1$ & $E_3$ with M : N

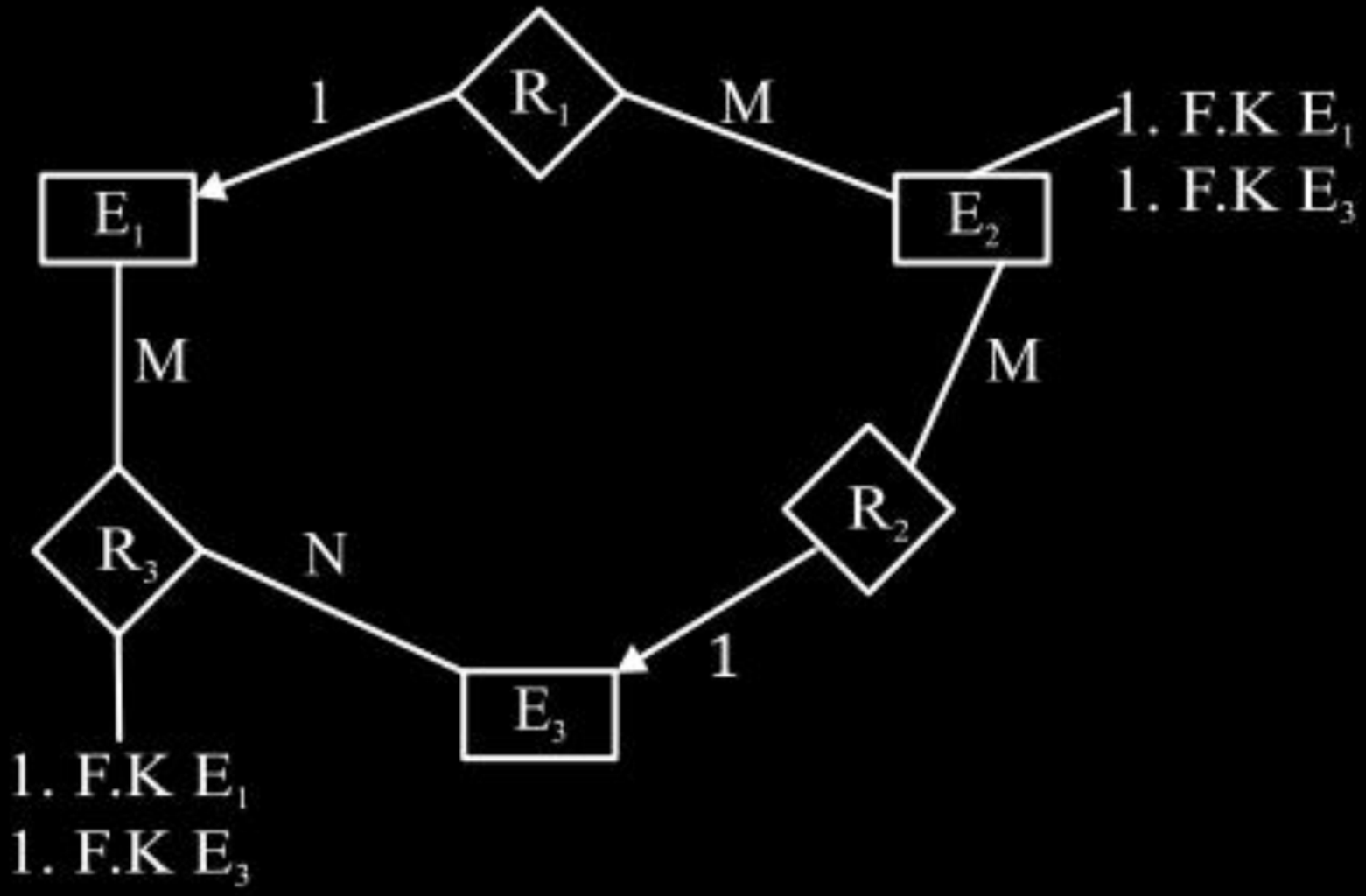
How many minimum relational tables required for given ERD?

A. 2

B. 3

C. 4

D. 5



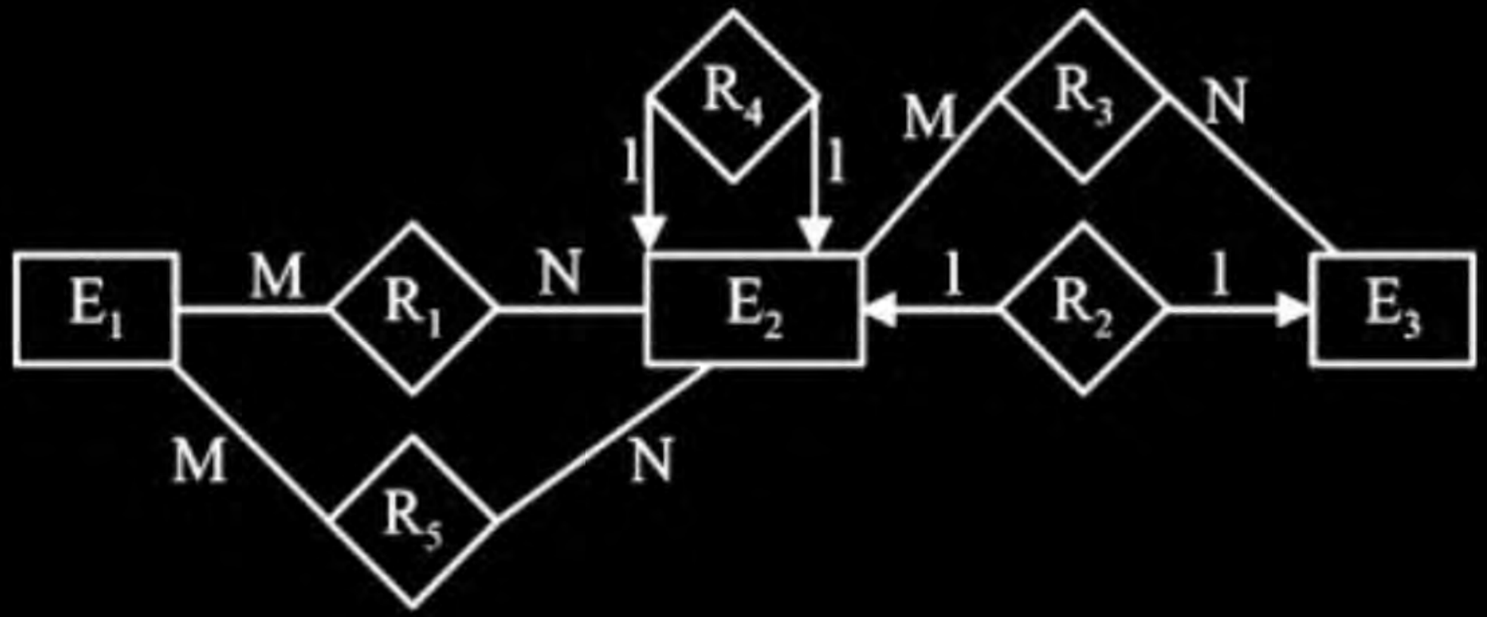

Min 4 Rel Table and 4 Foreign key.

Q.

[NAT]

Consider the following ER diagram

Total number of RDBMS table in the above diagram?



# Self Referential Relationship set:

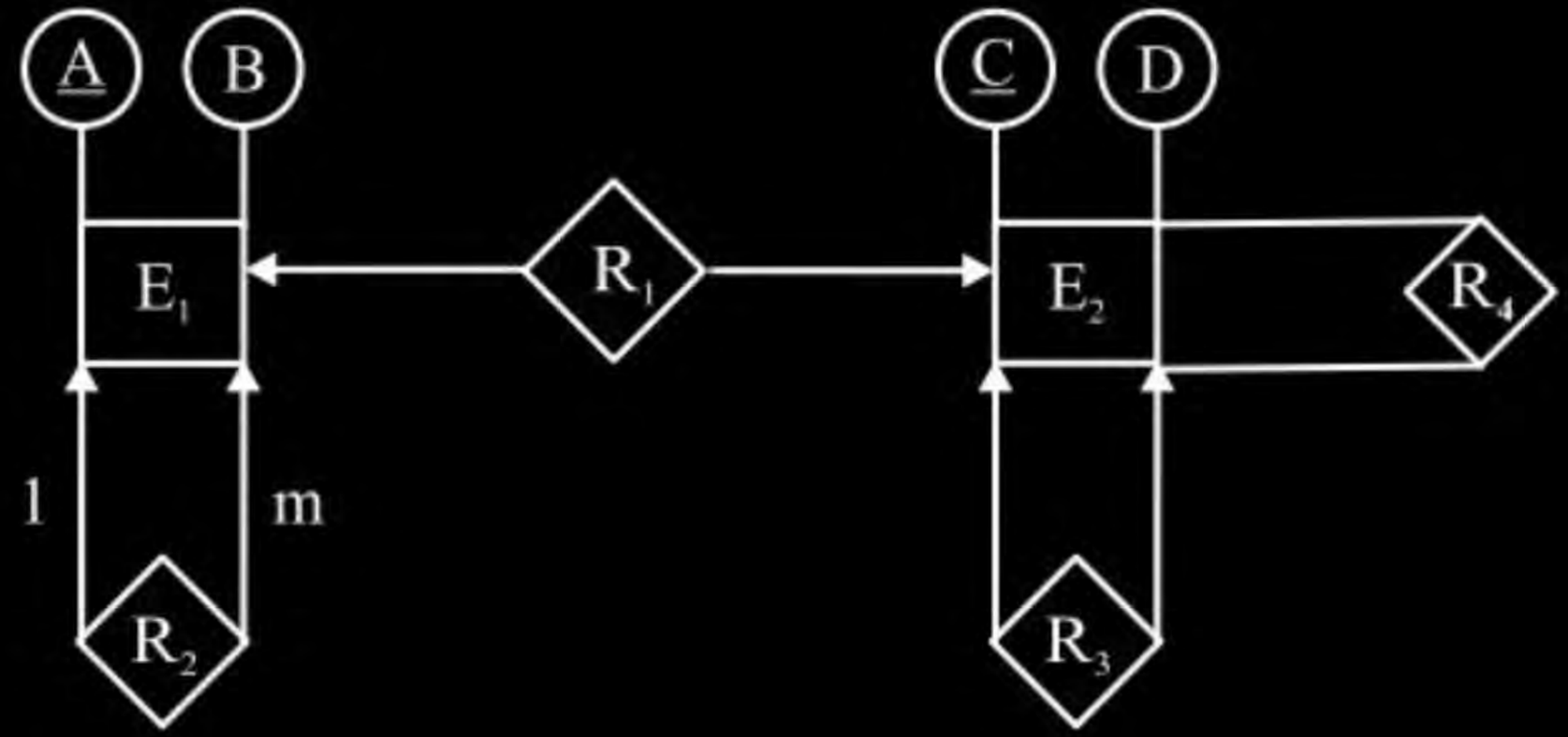
[ Recursive entity set]
Entities of entity set (E) related to some other entity of same entity set (E)


Recursive Entity set

Self Referential Relationship set.

- Also Binary Relationship set
- 1:M, M :1, M :1, 1:1, N: N be any of the relationship

Q.



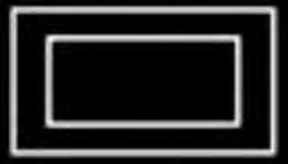

(I) $E_1\ R_2\ (\underline{A}\quad B\quad A)$

(II) $E_2\ R_1\ R_3\ (\underline{C}\quad D\quad A\quad C)$

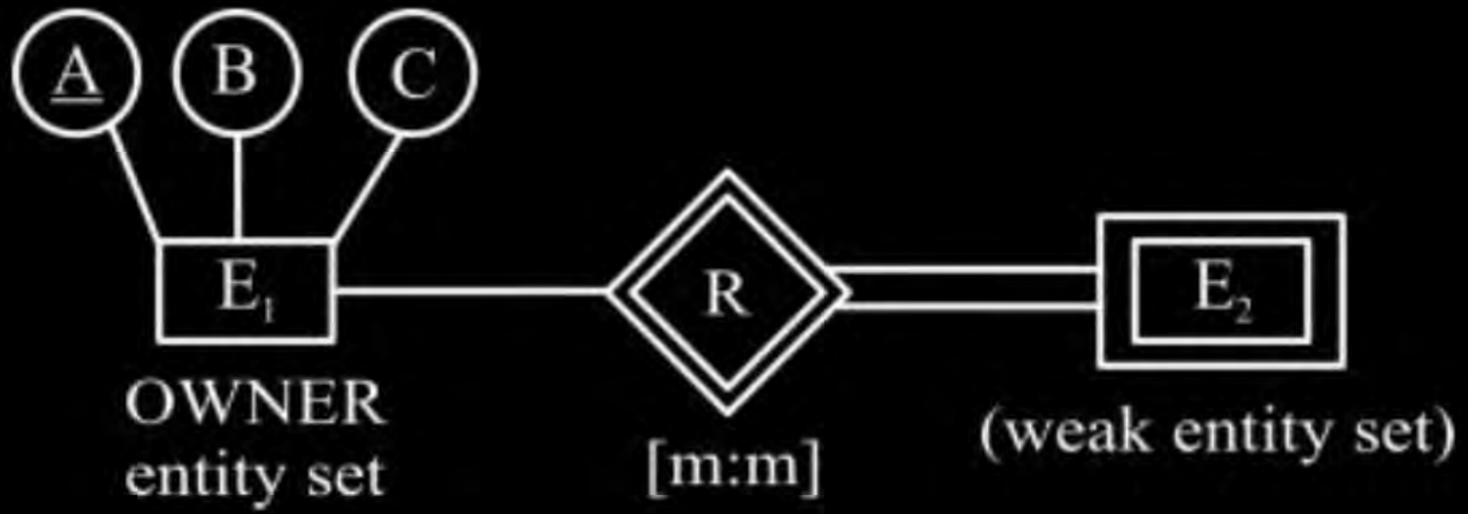
(III) $R_4\ C_3\ C_4$

# WEAK ENTITY SET:

⇒ Entity set with no key.
[Attributes of weak entity set not sufficient to differentiate entities uniquely]

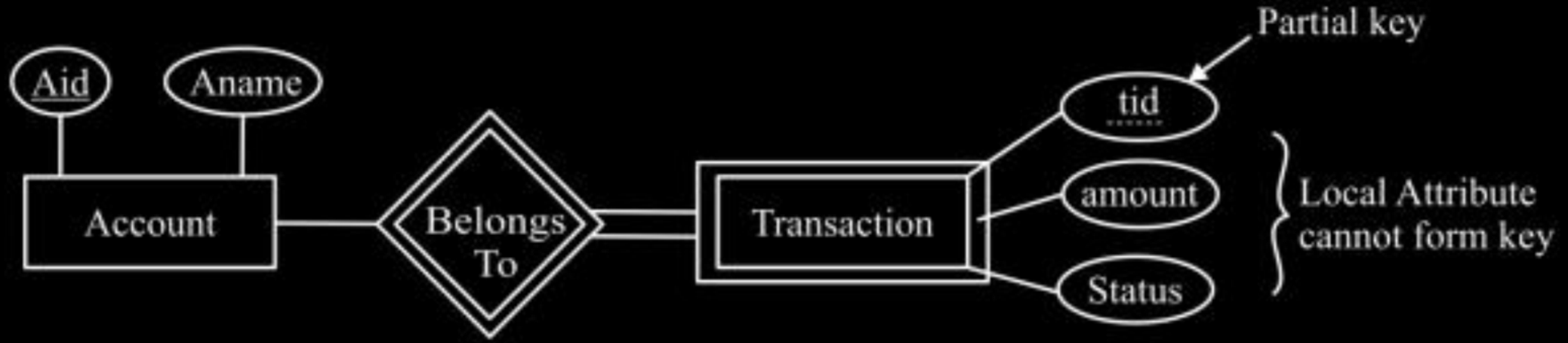
⇒ 

⇒ For each weak entity set there must be OWNER entity set which is strong entity.



⇒ Entities of weak entity set are depending entities.

Ex.




Aid tid

| Aid | tid |
|---|---|
| $A_1$ | $t_1$ |
| $A_2$ | $t_1$ |
| $A_2$ | $t_2$ |
| $A_3$ | $t_2$ |

| Aid | Aname |
|---|---|
| $A_1$ | |
| $A_2$ | |
| $A_3$ | |
| $A_4$ | |

Transaction

| tid | amount | Status |
|---|---|---|
| $t_1$ | 5000 | Debit |
| $t_1$ | 4000 | Credit |
| $t_2$ | 5000 | Debit |
| $t_2$ | 5000 | Debit |

⇑

[1: m]

Ambiguity

- Relationship set between weak entity set and identifier entity is also weak relationship set.

- Participation towards weak entity set end must be TOTAL PARTICIPATION.
- Mapping between identifier set must be one: many (1: m )
- RDBMS Design:
  Transaction belongs (Aid tid amount state)
  Account (Aid Aname)

**NOTE:** Weak entity set and multivalued attributes allowed to represent in ERD, but NOT allowed in RDBMS Table.

# THANK YOU!