Dr.Hrudaya Kumar Tripathy,

School of Computer Engineering, KIIT

Converting ER Diagrams to Tables

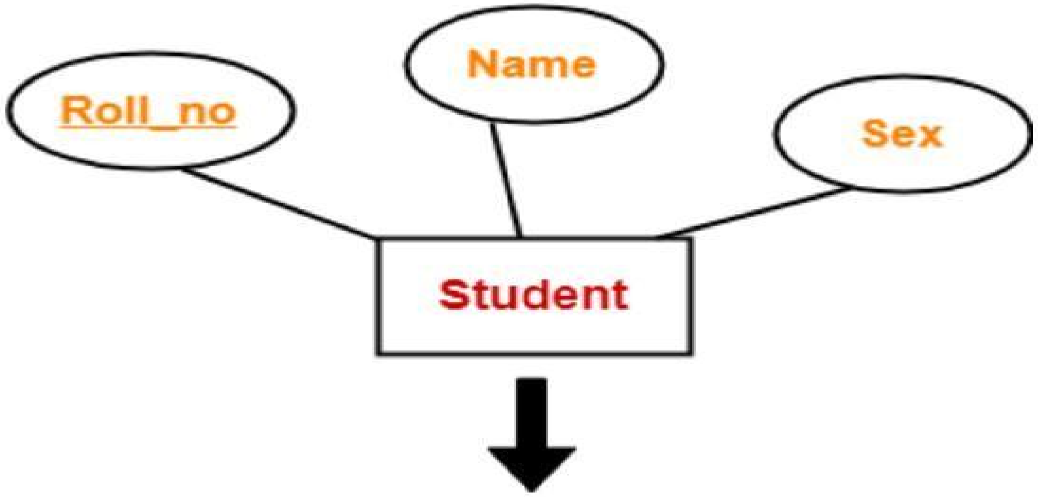
ER diagram is converted into the tables in relational model.

This is because relational models can be easily implemented by RDBMS like MySQL , Oracle etc.

Following rules are used for converting an ER diagram into the tables-

Rule-01: For Strong Entity Set With Only Simple Attributes-

- A strong entity set with only simple attributes will require only one table in relational model.
- Attributes of the table will be the attributes of the entity set.
- The primary key of the table will be the key attribute of the entity set.

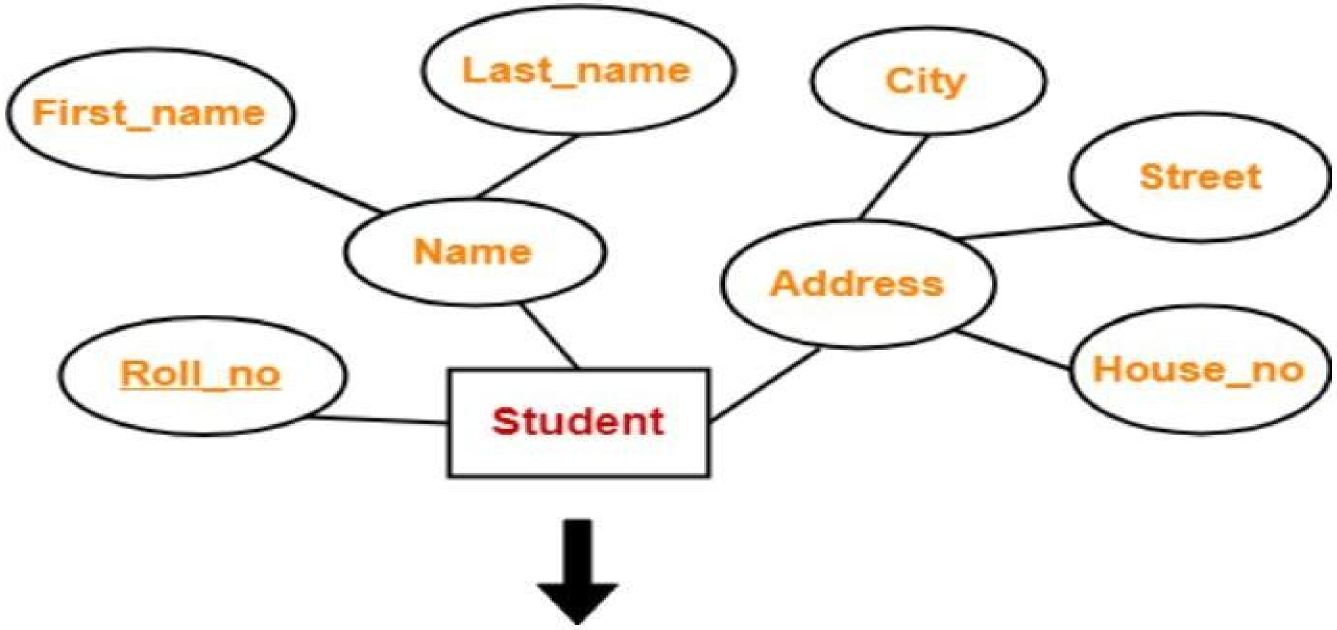


Roll_no	Name	Sex

Schema : Student (Roll_no , Name , Sex)

Rule-02: For Strong Entity Set With Composite Attributes-

- A strong entity set with any number of composite attributes will require only one table in relational model.
- While conversion, simple attributes of the composite attributes are taken into account and not the composite attribute itself.



Roll_no	First_name	Last_name	House_no	Street	City

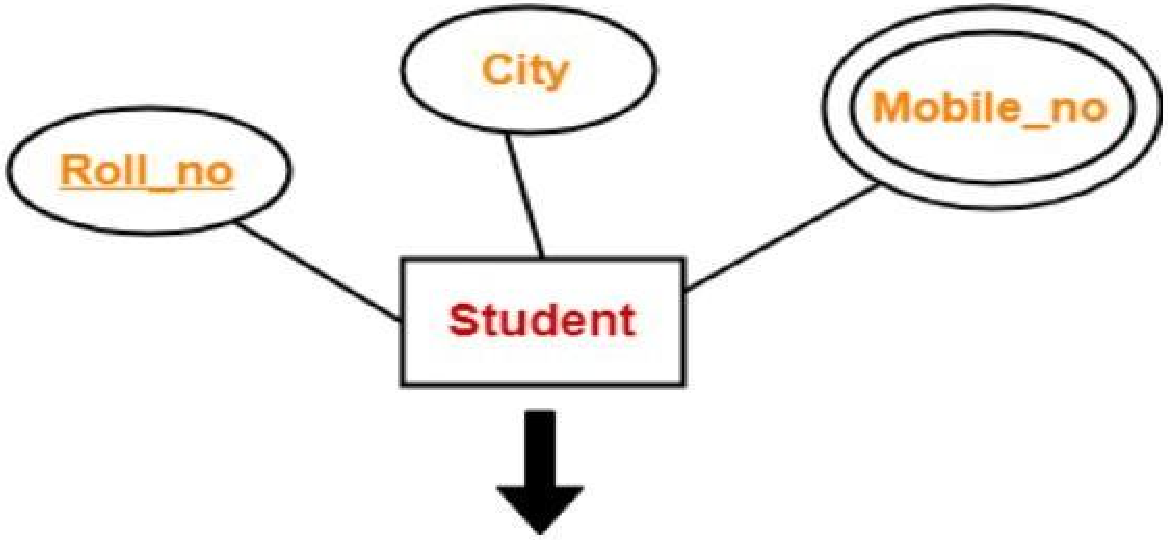
Schema : Student (Roll_no , First_name , Last_name , House_no , Street , City)

Rule-03: For Strong Entity Set With Multi Valued Attributes-

- A strong entity set with any number of multi valued attributes will require two tables in relational model.

One table will contain all the simple attributes with the primary key.

Other table will contain the primary key and all the multi valued attributes.



Roll_no	City

Roll_no	Mobile_no

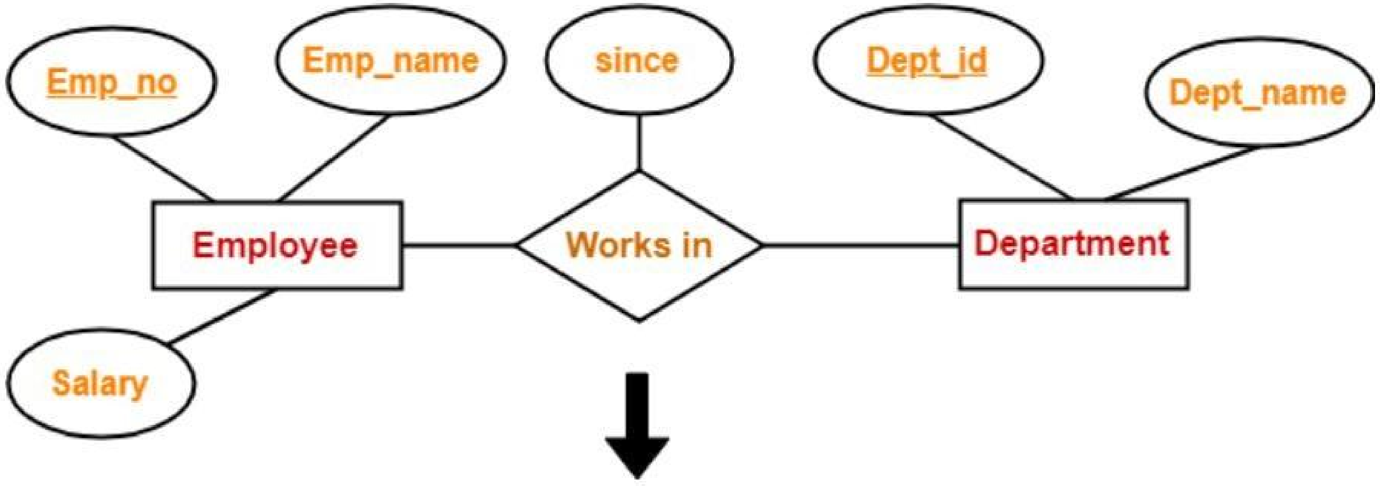
Rule-04: Translating Relationship Set into a Table-

- Attributes of the table are-

Primary key attributes of the participating entity sets

Its own descriptive attributes if any.

- Set of non-descriptive attributes will be the primary key.



Emp_no	Dept_id	since

Schema : Works in (Emp_no , Dept_id , since)

If we consider the overall ER diagram, three tables will be required in relational model-

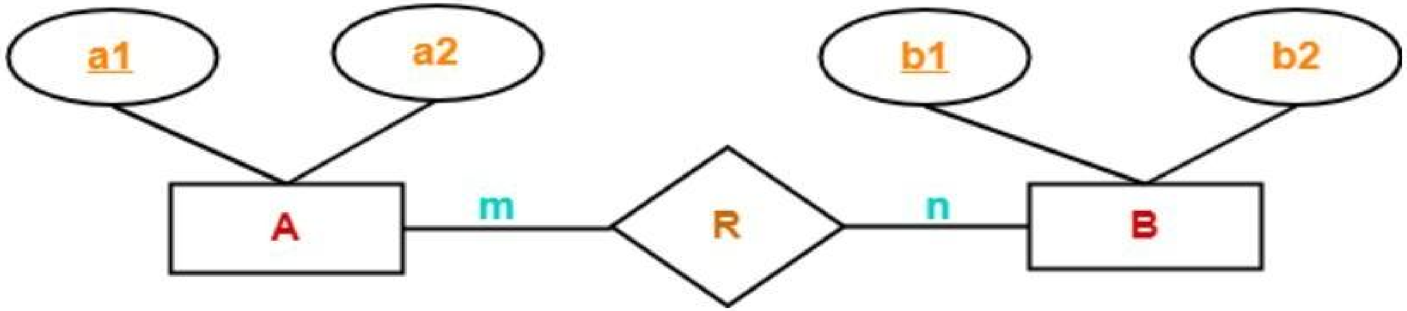
One table for the entity set “Employee”
One table for the entity set “Department”
One table for the relationship set “Works in”

Rule-05: For Binary Relationships With Cardinality Ratios-

The following four cases are possible-

- Case-01: Binary relationship with cardinality ratio m:n
- Case-02: Binary relationship with cardinality ratio 1:n
- Case-03: Binary relationship with cardinality ratio m:1
- Case-04: Binary relationship with cardinality ratio 1:1

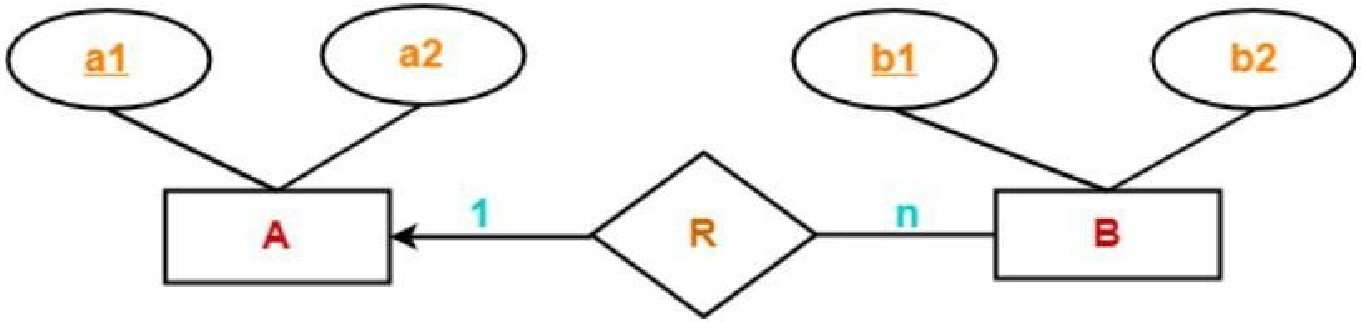
Rule-05:Case-01: For Binary Relationship With Cardinality Ratio m:n



Here, three tables will be required-

1. A (a1 , a2)
2. R (a1 , b1)
3. B (b1 , b2)

Rule-05:Case-02: For Binary Relationship With Cardinality Ratio 1:n

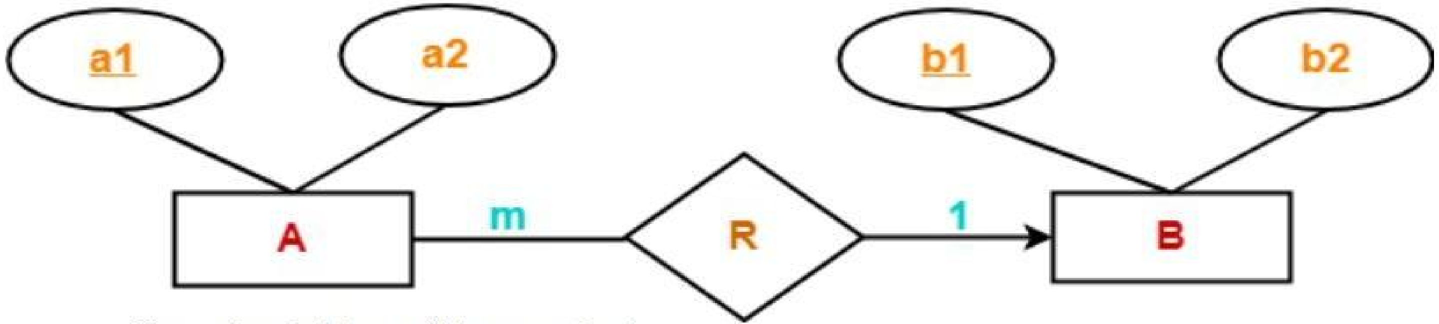


Here, two tables will be required-

1. A (a1 , a2)
2. BR (a1 , b1 , b2)

NOTE- Here, combined table will be drawn for the entity set B and relationship set R.

Rule-05:Case-03: For Binary Relationship With Cardinality Ratio m:1

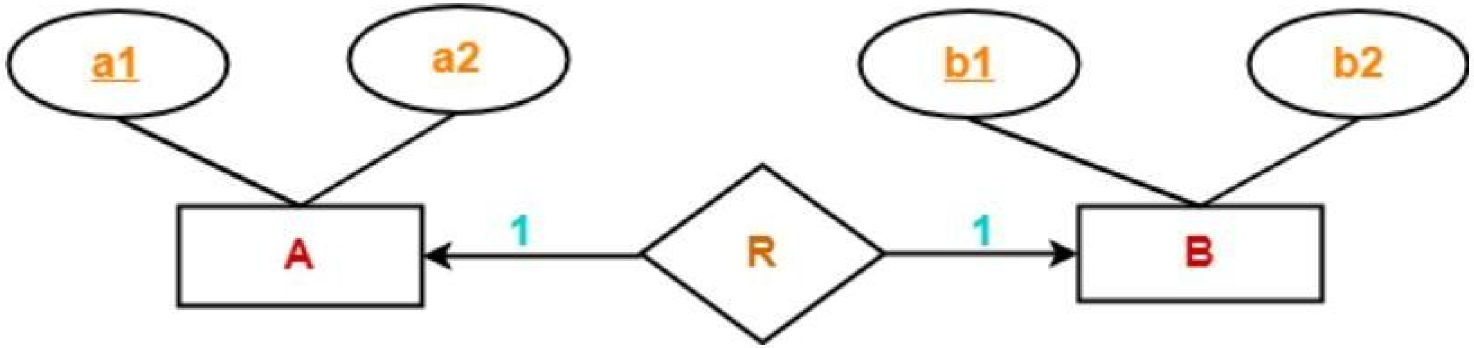


Here, two tables will be required-

1. AR (a1 , a2 , b1)
2. B (b1 , b2)

NOTE- Here, combined table will be drawn for the entity set A and relationship set R.

Rule-05:Case-04: For Binary Relationship With Cardinality Ratio 1:1



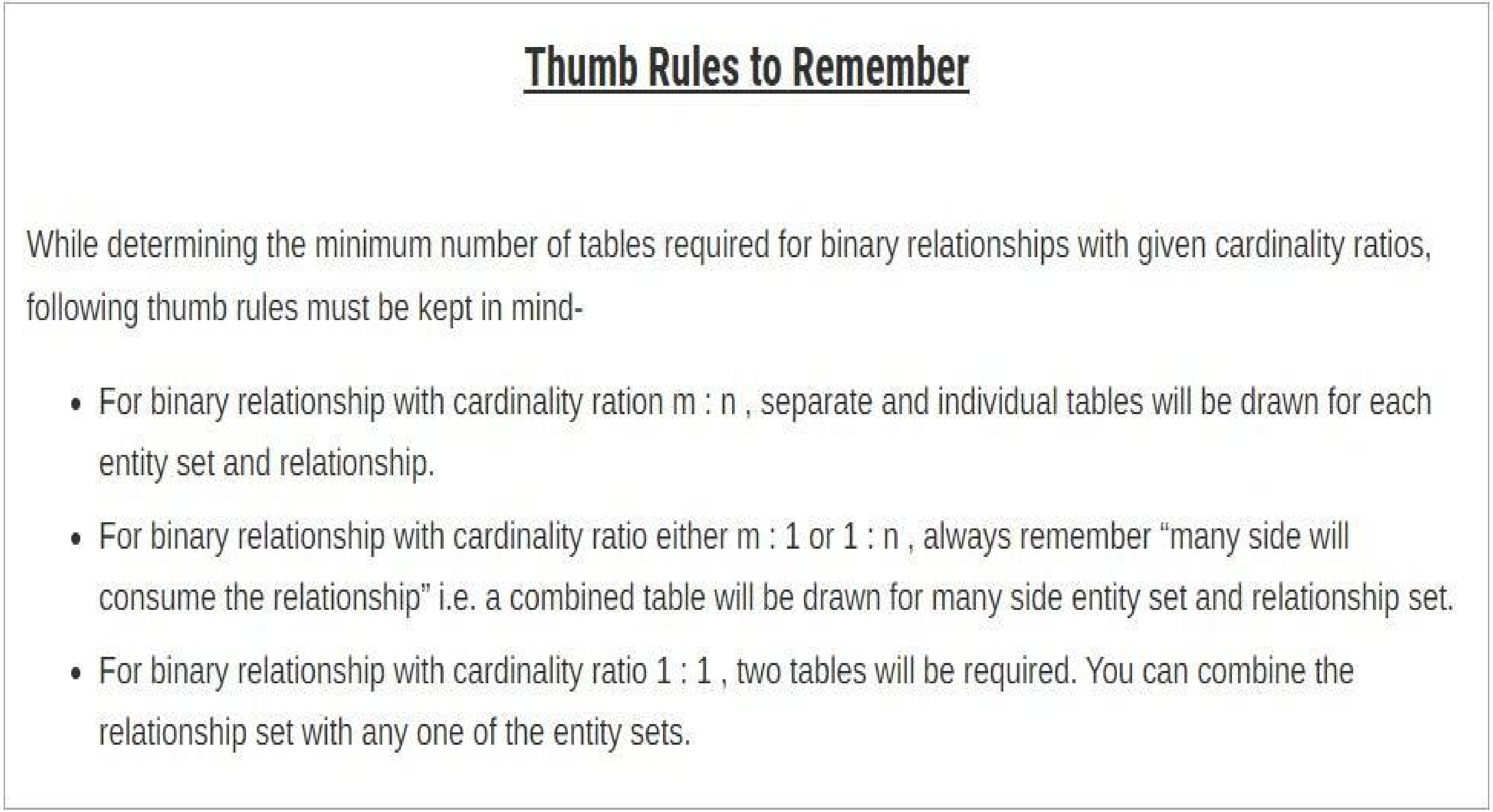
Here, two tables will be required. Either combine ‘R’ with ‘A’ or ‘B’

Way-01:

1. AR (a1 , a2 , b1)
2. B (b1 , b2)

Way-02:

1. A (a1 , a2)
2. BR (a1 , b1 , b2)



Thumb Rules to Remember

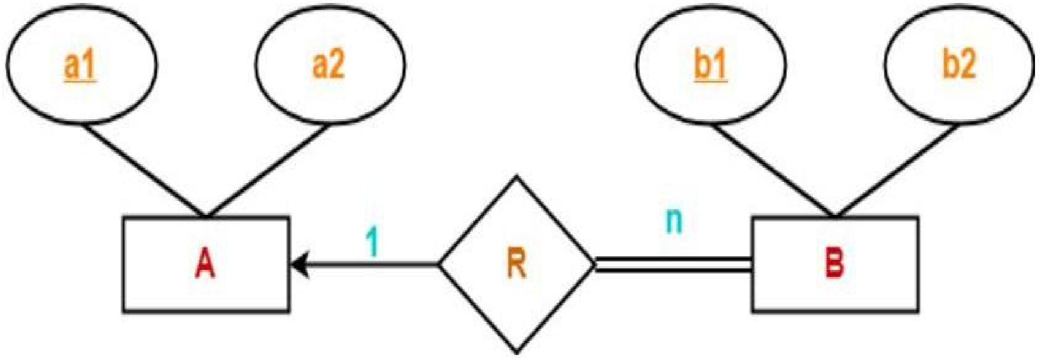
While determining the minimum number of tables required for binary relationships with given cardinality ratios, following thumb rules must be kept in mind-

- For binary relationship with cardinality ration m : n , separate and individual tables will be drawn for each entity set and relationship.
- For binary relationship with cardinality ratio either m : 1 or 1 : n , always remember “many side will consume the relationship” i.e. a combined table will be drawn for many side entity set and relationship set.
- For binary relationship with cardinality ratio 1 : 1 , two tables will be required. You can combine the relationship set with any one of the entity sets.

Rule-06: For Binary Relationship With Both Cardinality Constraints and Participation Constraints-

- Cardinality constraints will be implemented as discussed in Rule-05.
- Because of the total participation constraint, foreign key acquires NOT NULL constraint i.e. now foreign key can not be null.

Case-01: For Binary Relationship With Cardinality Constraint and Total Participation Constraint From One Side-



Because cardinality ratio = 1 : n , so we will combine the entity set B and relationship set R.

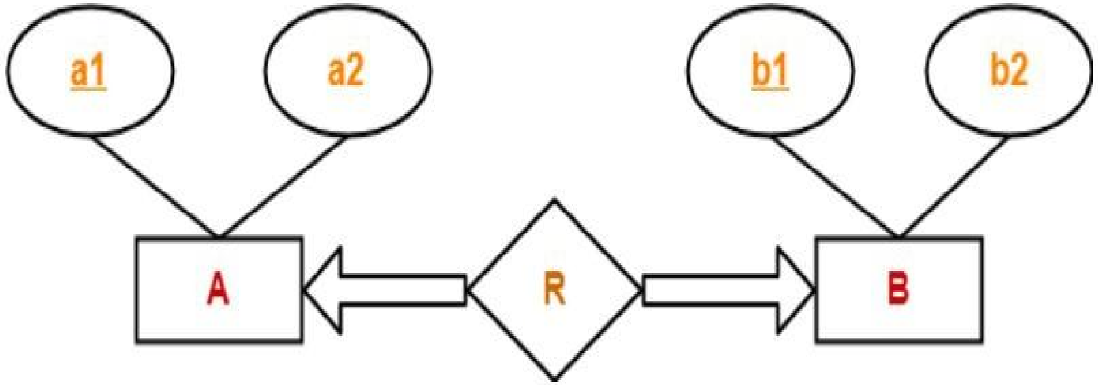
Then, two tables will be required-

1. A (a1 , a2)
2. BR (a1 , b1 , b2)

Because of total participation, foreign key a1 has acquired NOT NULL constraint, so it can't be null now.

Case-02: For Binary Relationship With Cardinality Constraint and Total Participation Constraint From Both Sides-

If there is a key constraint from both the sides of an entity set with total participation, then that binary relationship is represented using only single table.

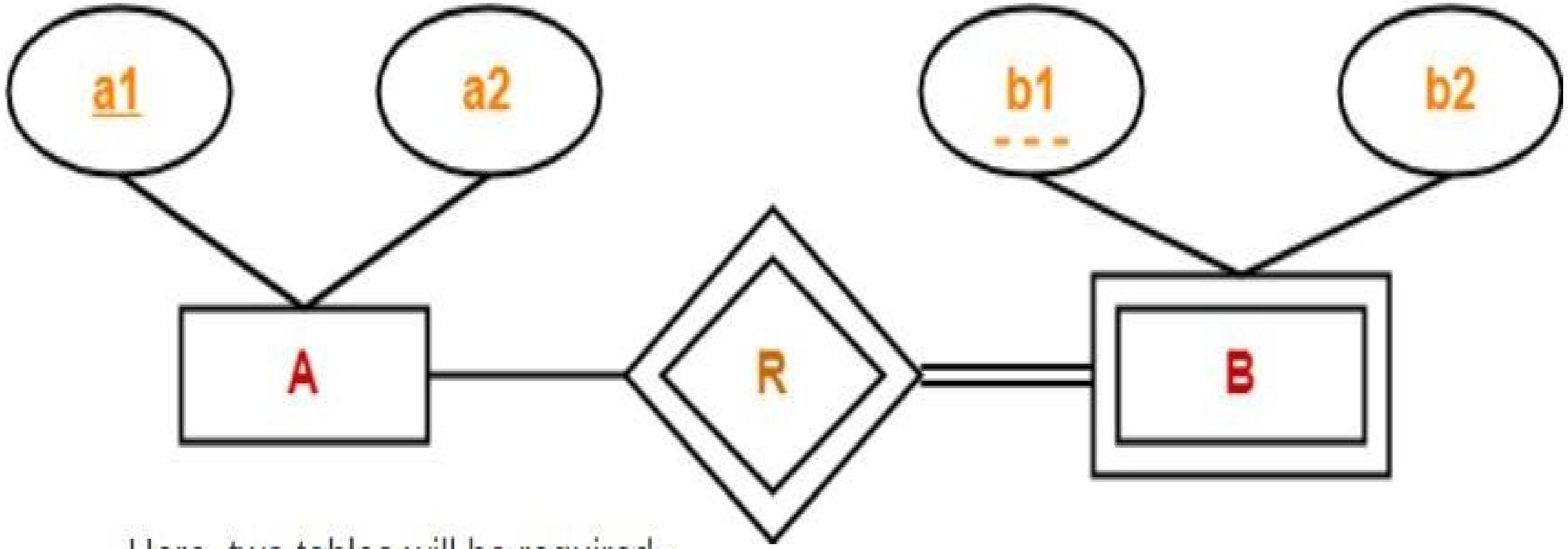


Here, Only one table is required.

- ARB (a1 , a2 , b1 , b2)

Rule-07: For Binary Relationship With Weak Entity Set-

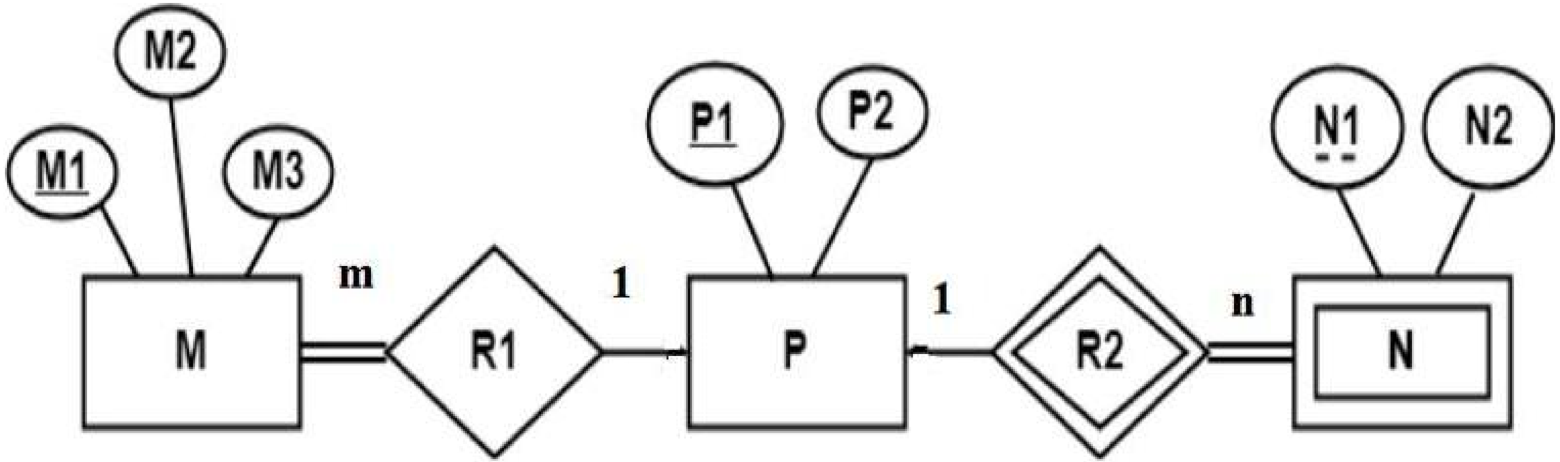
- **Weak entity set always appears in association with identifying relationship with total participation constraint.**



Here, two tables will be required-

1. A (a1 , a2)
2. BR (a1 , b1 , b2)

Find the minimum number of tables required for the following ER diagrams in relational model-

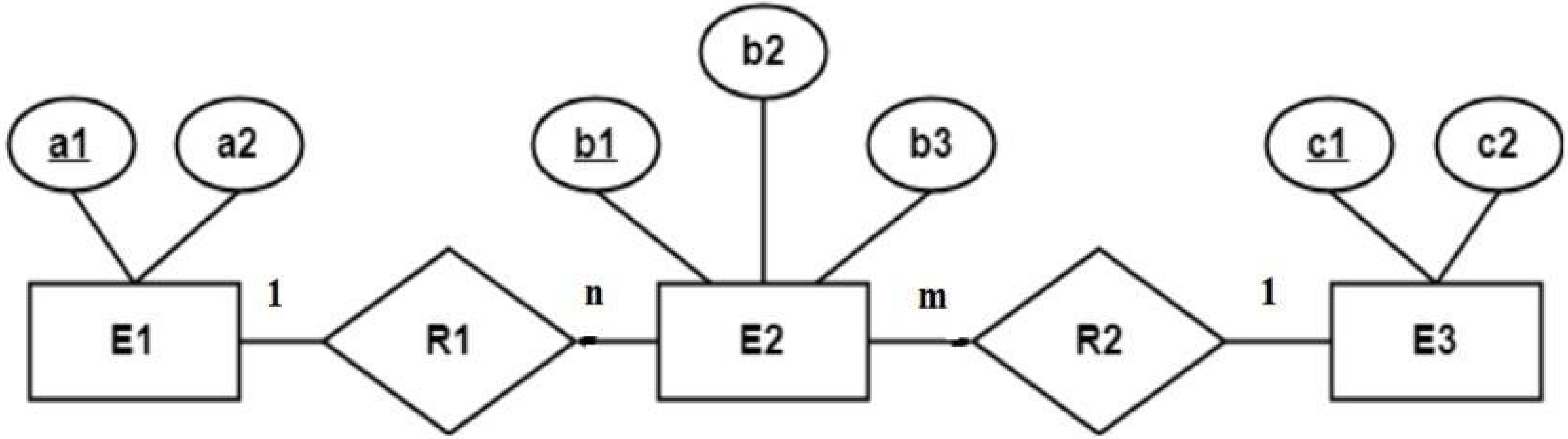


Applying the rules, minimum 3 tables will be required-

MR1 (<u>M1</u> , M2 , M3 , P1)

P (<u>P1</u> , P2)

NR2 (<u>P1 , N1</u> , N2)

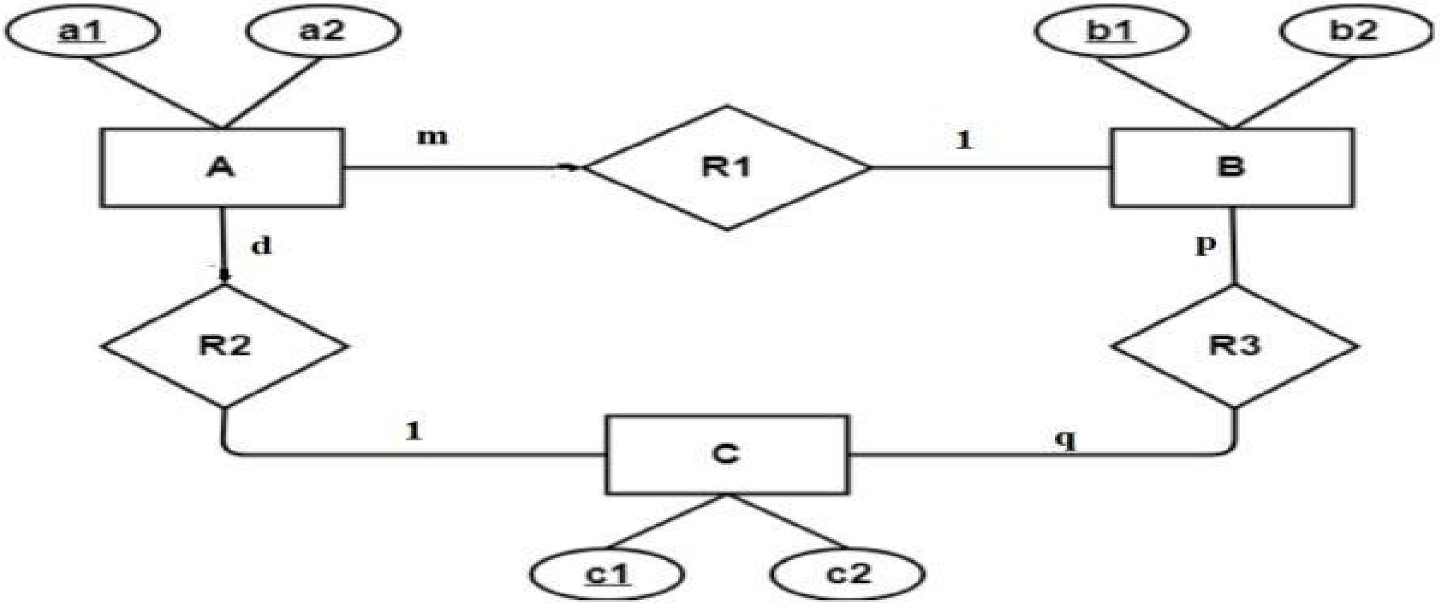


Applying the rules, minimum 3 tables will be required-

E1 (<u>a1</u> , a2)

E2R1R2 (<u>b1</u> , b2 , a1 , c1 , b3)

E3 (<u>c1</u> , c2)



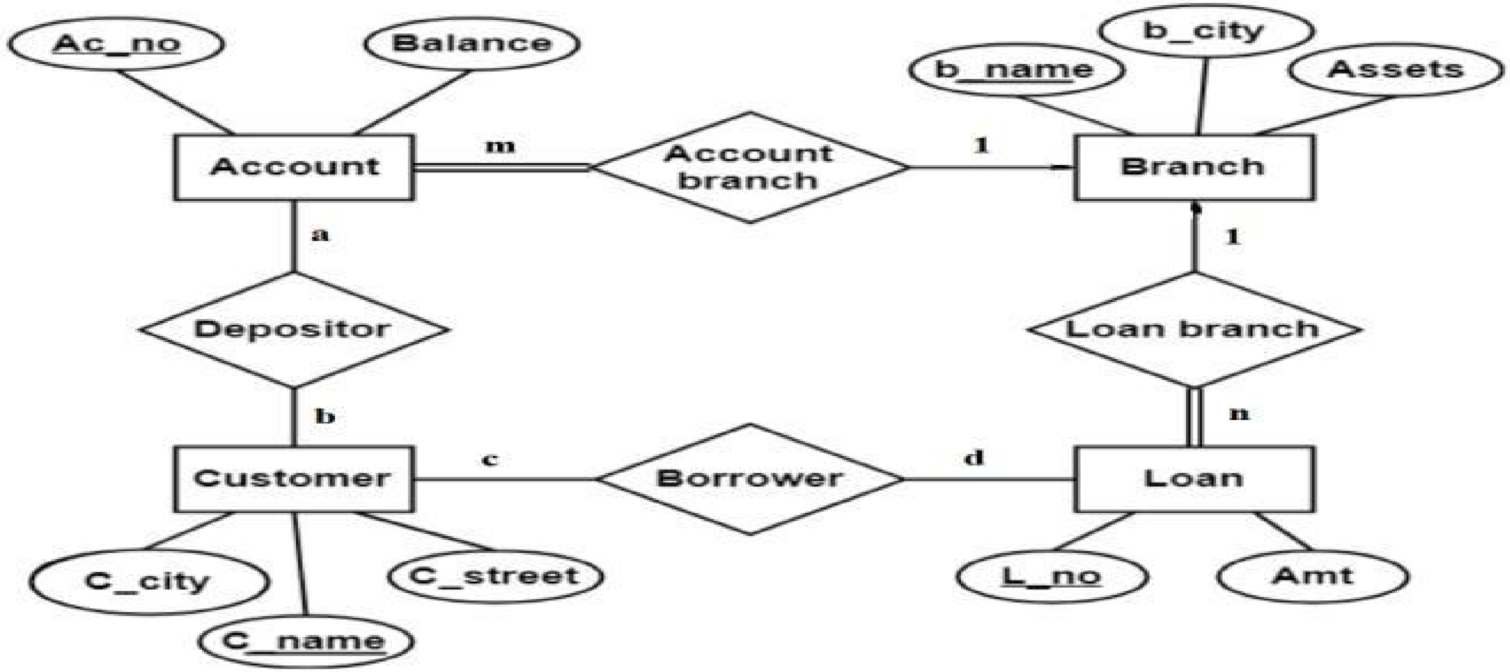
Applying the rules, minimum 4 tables will be required-

AR1R2 (<u>a1</u> , a2 , b1 , c1)

B (<u>b1</u> , b2)

C (<u>c1</u> , c2)

R3 (<u>b1 , c1</u>)



Applying the rules that we have learnt, minimum 6 tables will be required-

Account_Account branch (Ac_no , Balance , b_name)

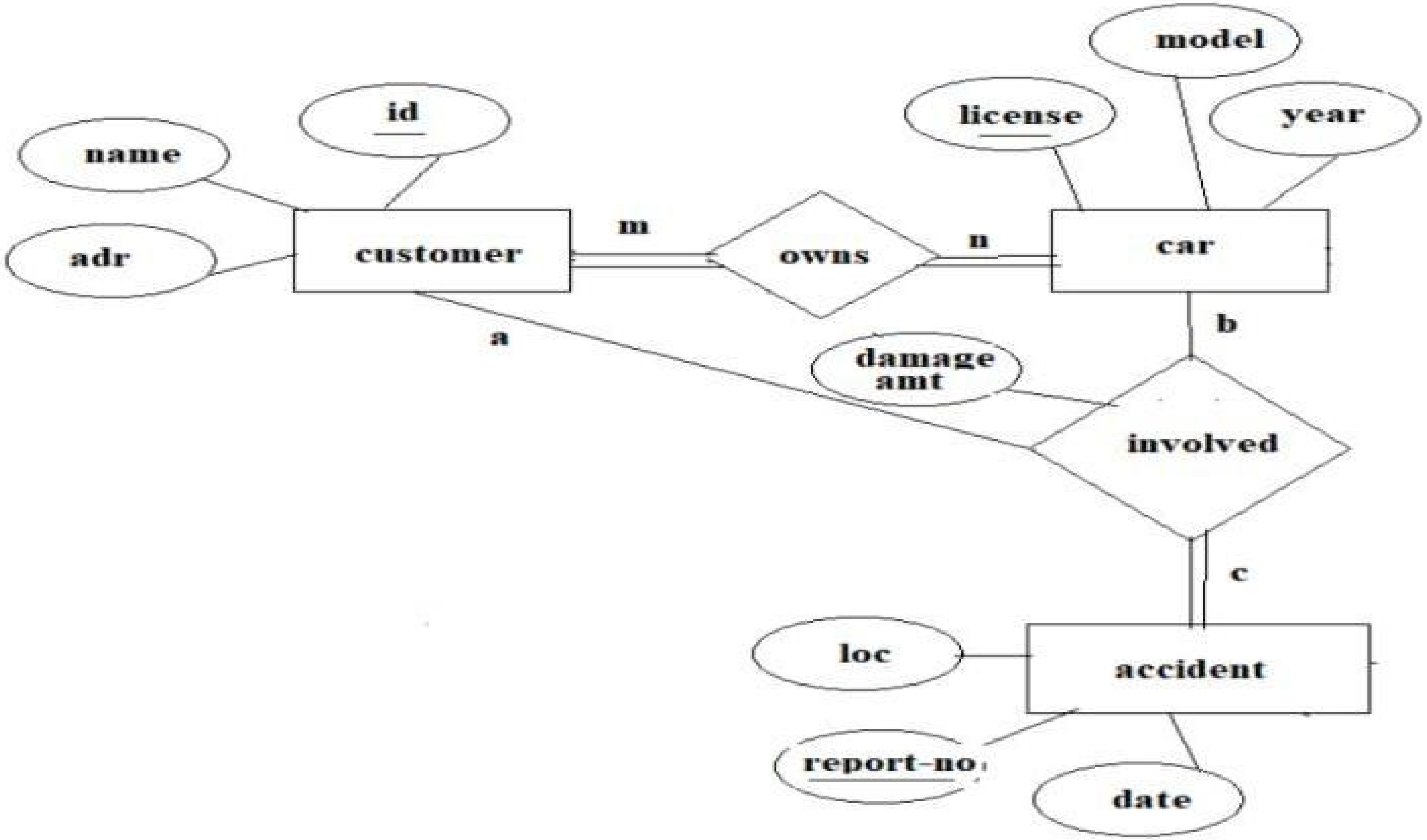
Branch (b_name , b_city , Assets)

Loan_loan branch (L_no , Amt , b_name)

Borrower (C_name , L_no)

Customer (C_name , C_street , C_city)

Depositor (C_name , Ac_no)



Customer_owns_car (id, adr, name, license, model, year)

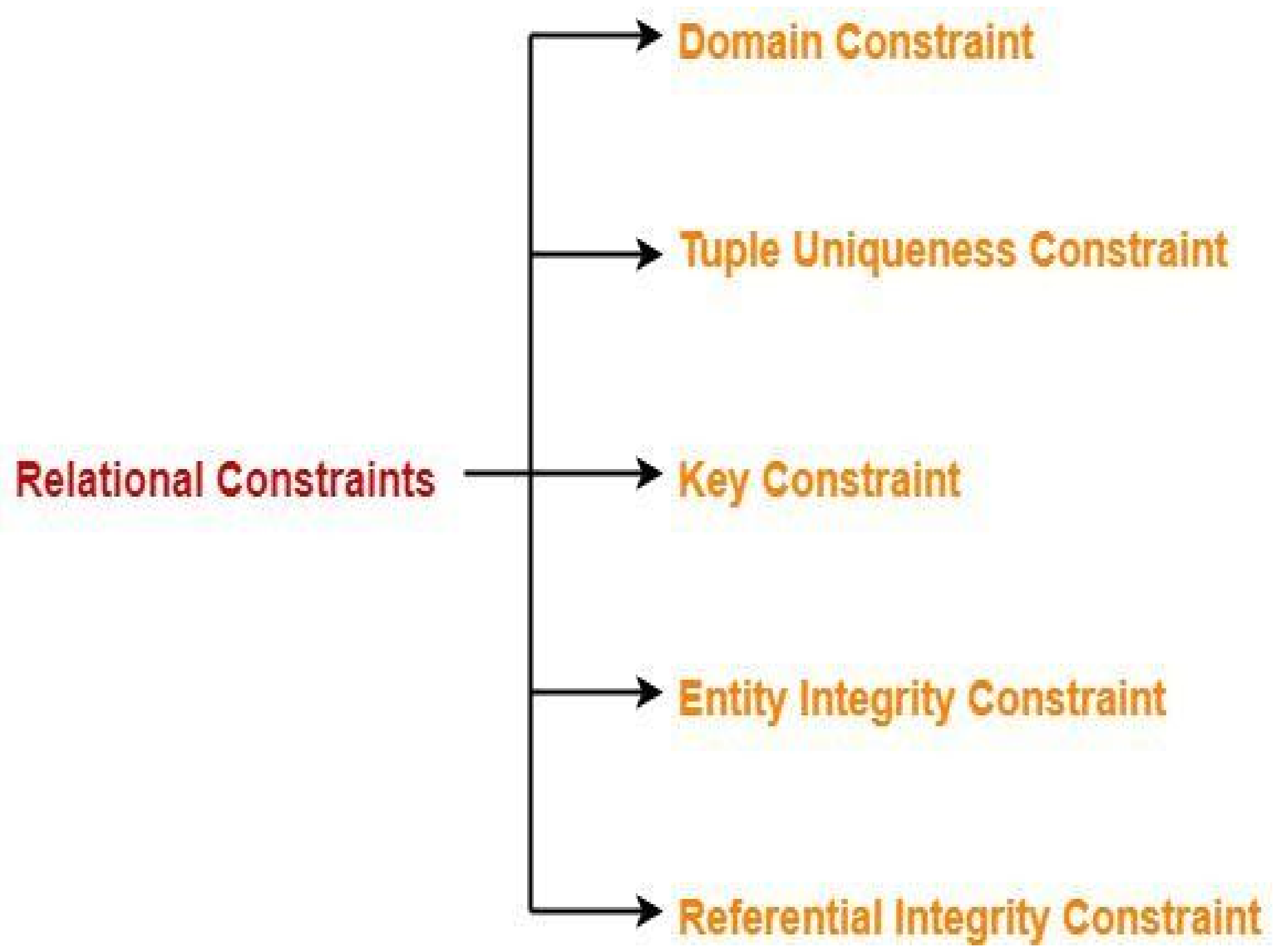
Involved (license, damage amt, report-no)

Accident (report-no, loc, date)

Constraints in DBMS

- Relational constraints are the restrictions imposed on the database contents and operations.

- They ensure the correctness of data in the database.

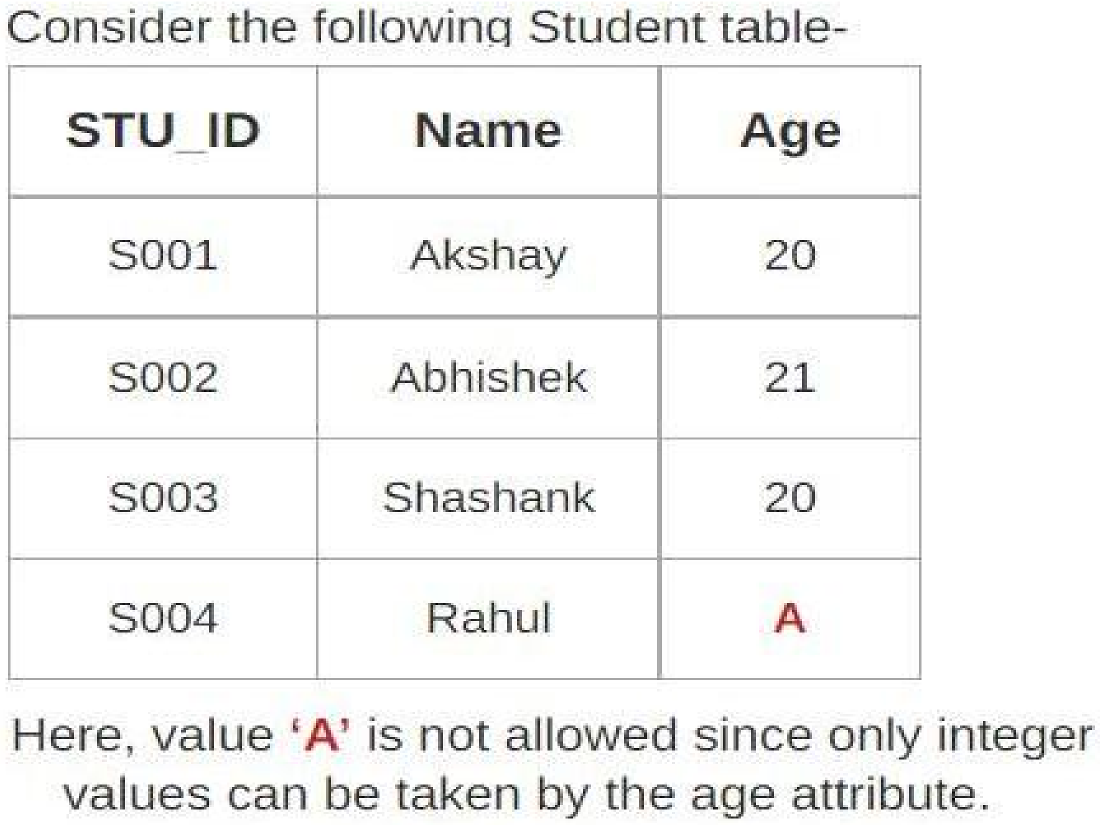


Relational Constraints

- Domain Constraint
- Tuple Uniqueness Constraint
- Key Constraint
- Entity Integrity Constraint
- Referential Integrity Constraint

Domain Constraint-

- ❑ Domain constraint defines the domain or set of values for an attribute.
- ❑ It specifies that the value taken by the attribute must be the atomic value from its domain.



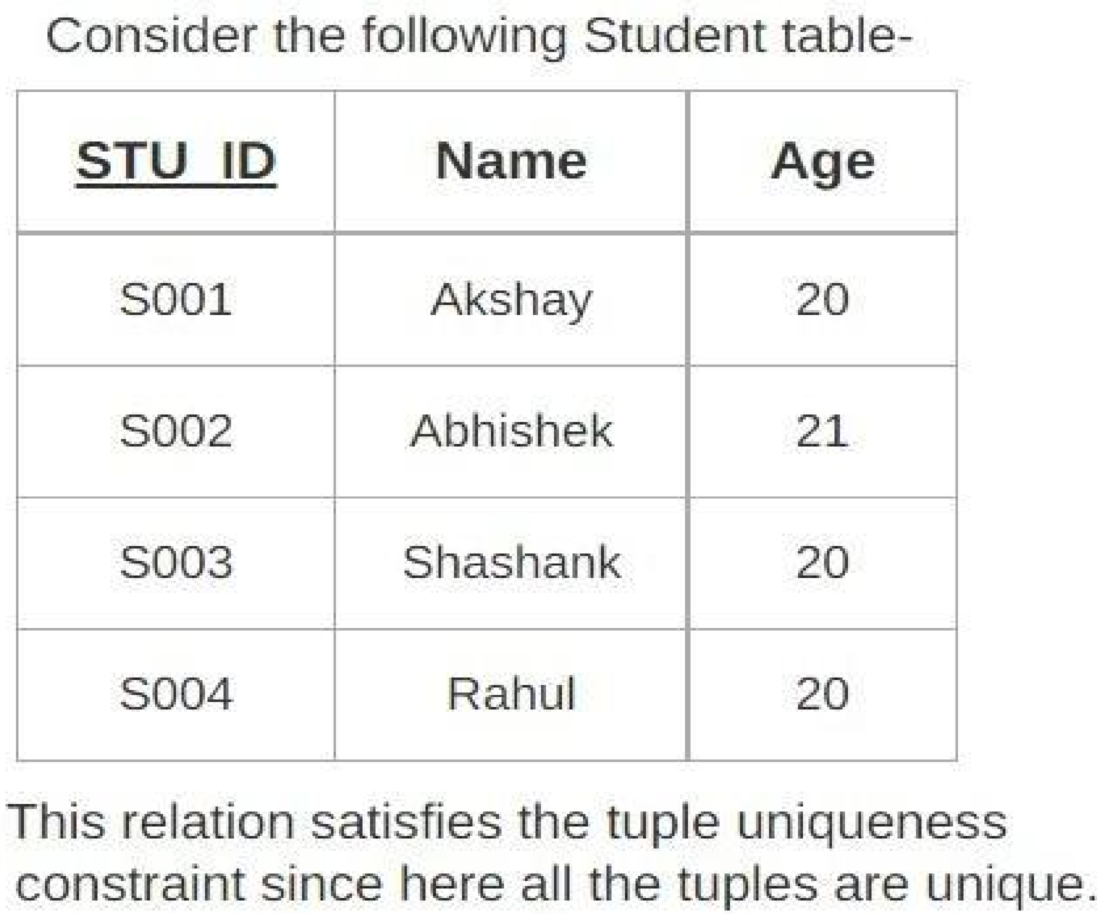
Consider the following Student table-

STU_ID	Name	Age
S001	Akshay	20
S002	Abhishek	21
S003	Shashank	20
S004	Rahul	A

Here, value 'A' is not allowed since only integer values can be taken by the age attribute.

Tuple Uniqueness Constraint-

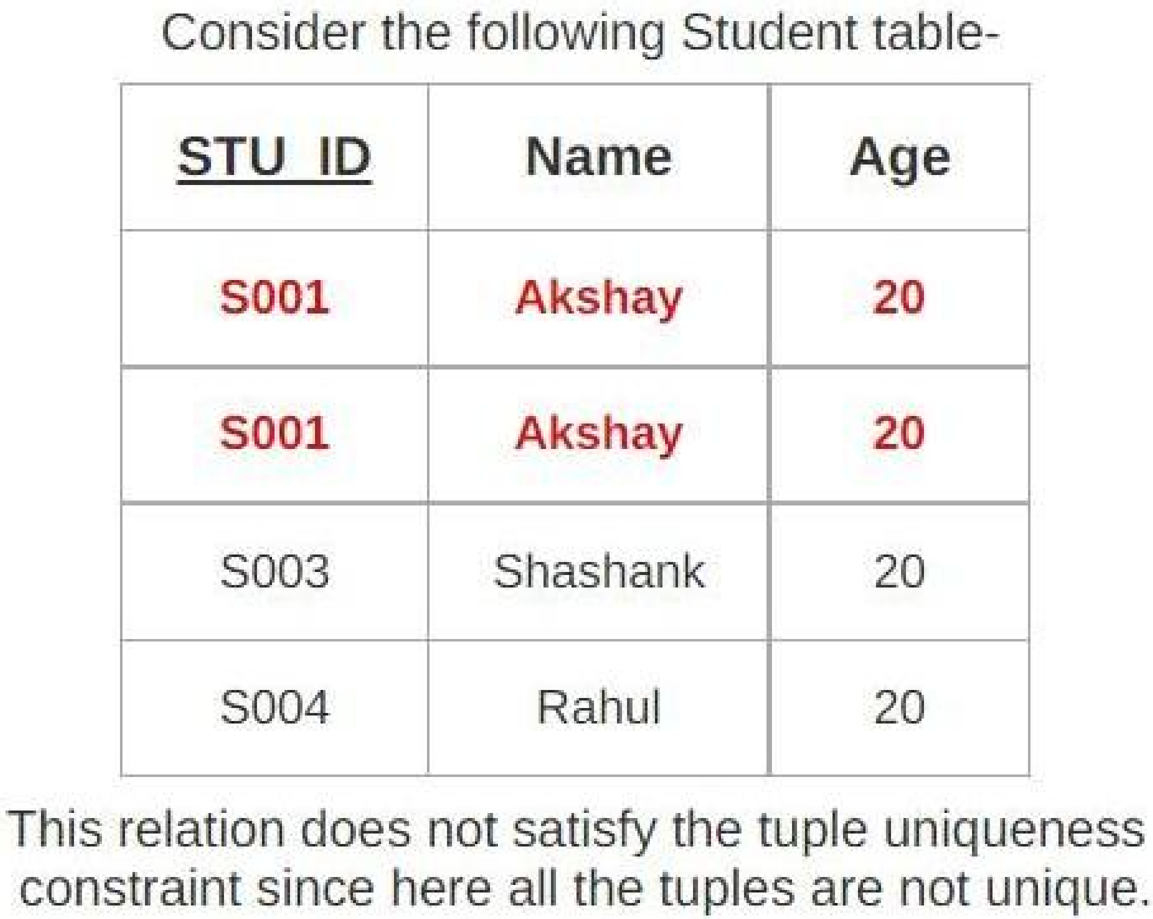
- ❑ **Tuple Uniqueness constraint specifies that all the tuples must be necessarily unique in any relation.**



Consider the following Student table-

STU_ID	Name	Age
S001	Akshay	20
S002	Abhishek	21
S003	Shashank	20
S004	Rahul	20

This relation satisfies the tuple uniqueness constraint since here all the tuples are unique.



Consider the following Student table-

STU_ID	Name	Age
S001	**Akshay**	**20**
S001	**Akshay**	**20**
S003	Shashank	20
S004	Rahul	20

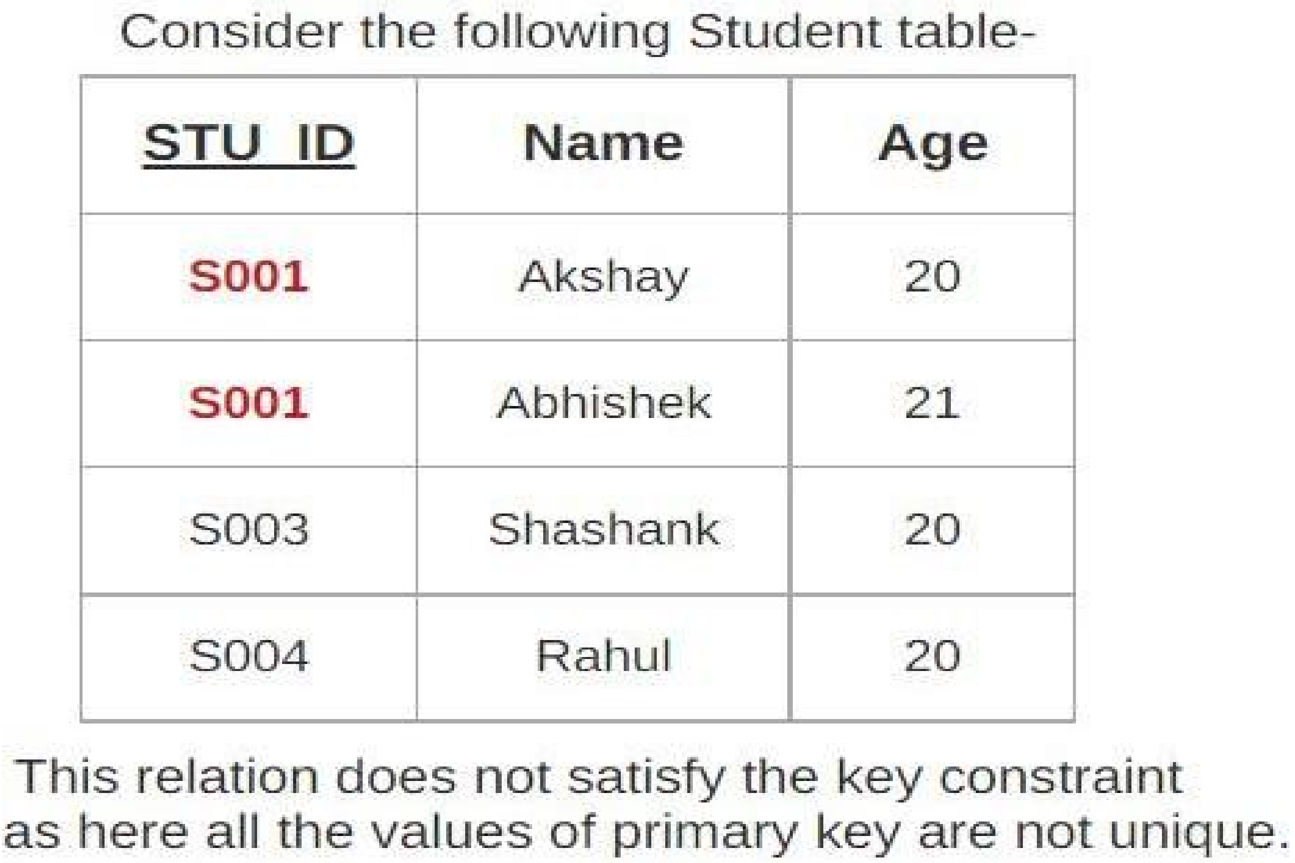
This relation does not satisfy the tuple uniqueness constraint since here all the tuples are not unique.

Key Constraint-

- Key constraint specifies that in any relation-

  All the values of primary key must be unique.

  The value of primary key must not be null.



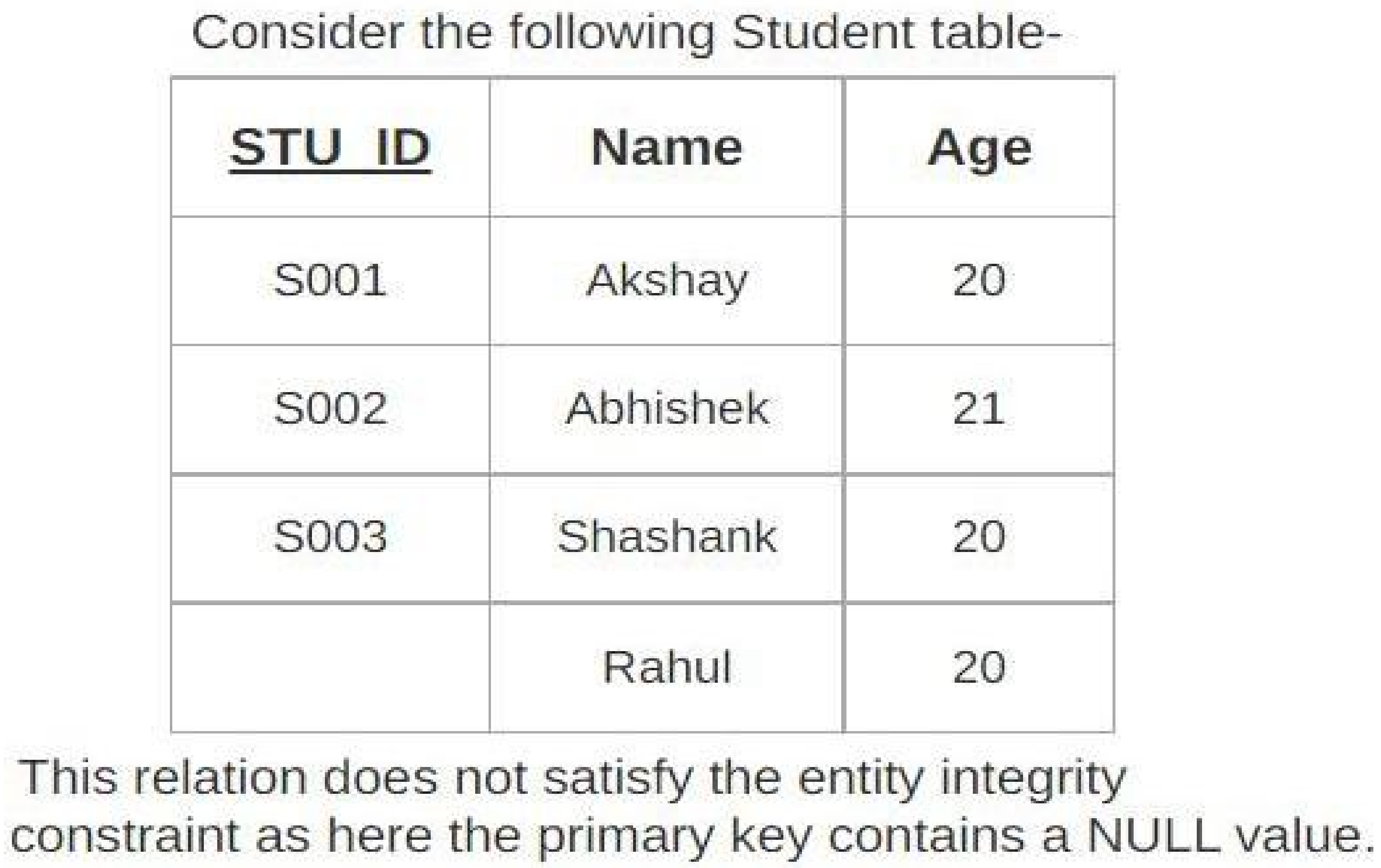
Consider the following Student table-

STU_ID	Name	Age
S001	Akshay	20
S001	Abhishek	21
S003	Shashank	20
S004	Rahul	20

This relation does not satisfy the key constraint
as here all the values of primary key are not unique.

Entity Integrity Constraint-

- ❑ Entity integrity constraint specifies that no attribute of primary key must contain a null value in any relation.
- ❑ This is because the presence of null value in the primary key violates the uniqueness property.



Consider the following Student table-

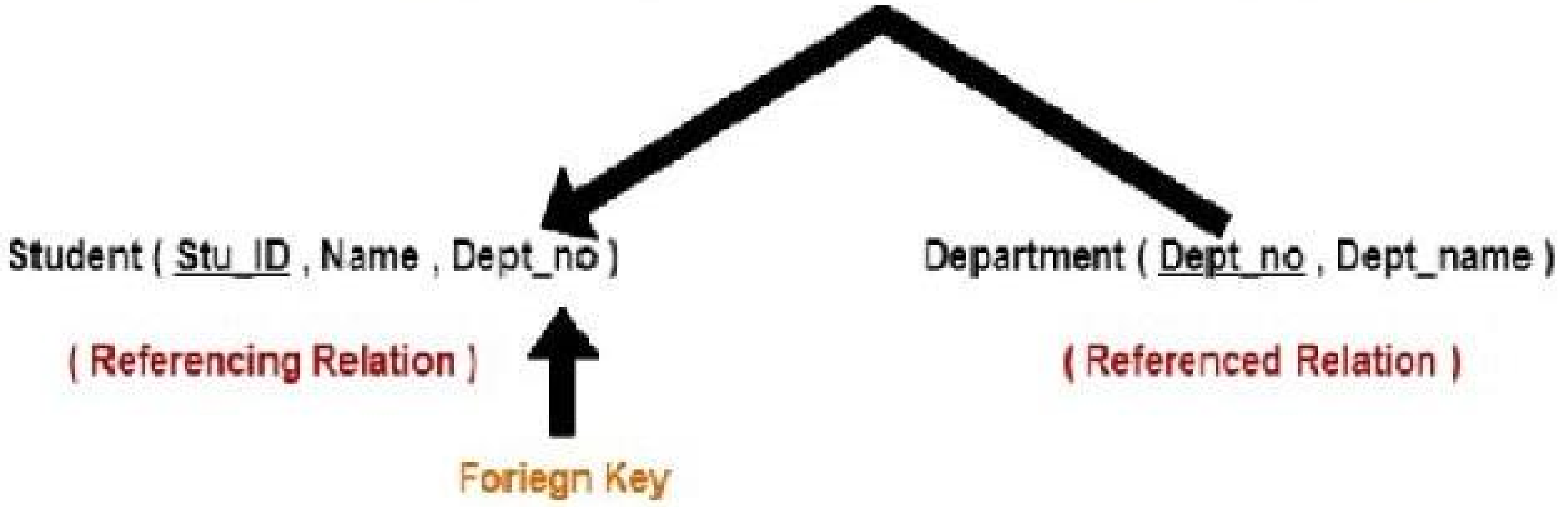
STU_ID	Name	Age
S001	Akshay	20
S002	Abhishek	21
S003	Shashank	20
	Rahul	20

This relation does not satisfy the entity integrity constraint as here the primary key contains a NULL value.

Referential Integrity Constraint-

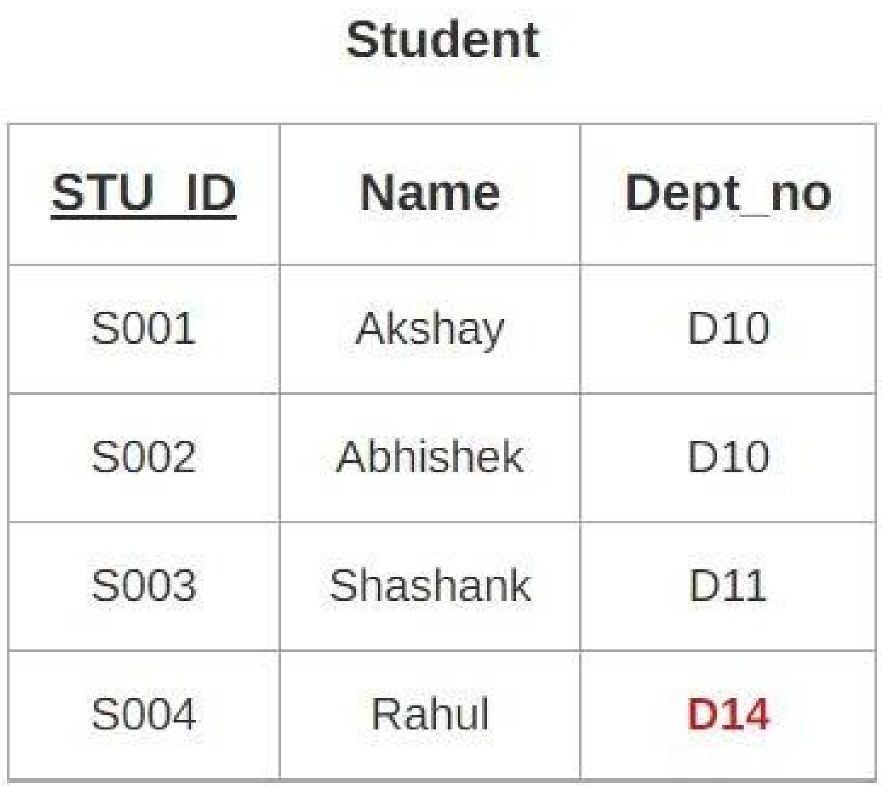
- This constraint is enforced when a foreign key references the primary key of a relation.
- It specifies that all values taken by the foreign key must either be available in the relation of the primary key or be null.
- We can not insert a record into a referencing relation if the corresponding record does not exist in the referenced relation.
- We can not delete or update a record of the referenced relation if the corresponding record exists in referencing relation.

Consider the following two relations- ‘Student’ and ‘Department’.
Here, relation ‘Student’ references the relation ‘Department’.



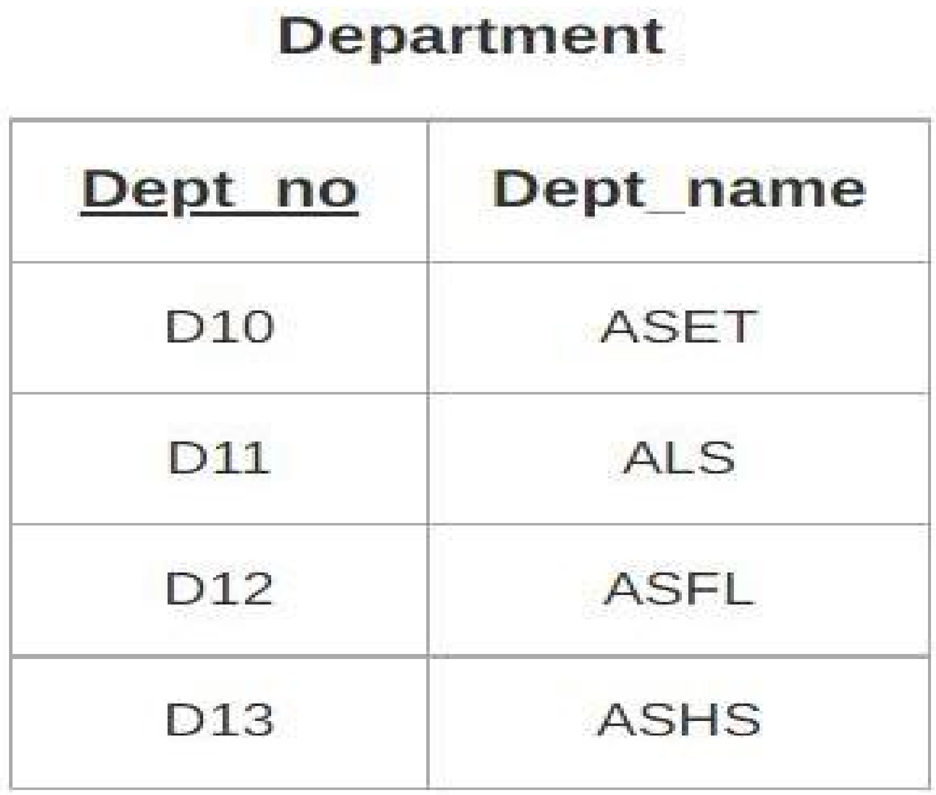
Student (Stu_ID , Name , Dept_no)

Department (Dept_no , Dept_name)



Student

STU_ID	Name	Dept_no
S001	Akshay	D10
S002	Abhishek	D10
S003	Shashank	D11
S004	Rahul	D14



Department

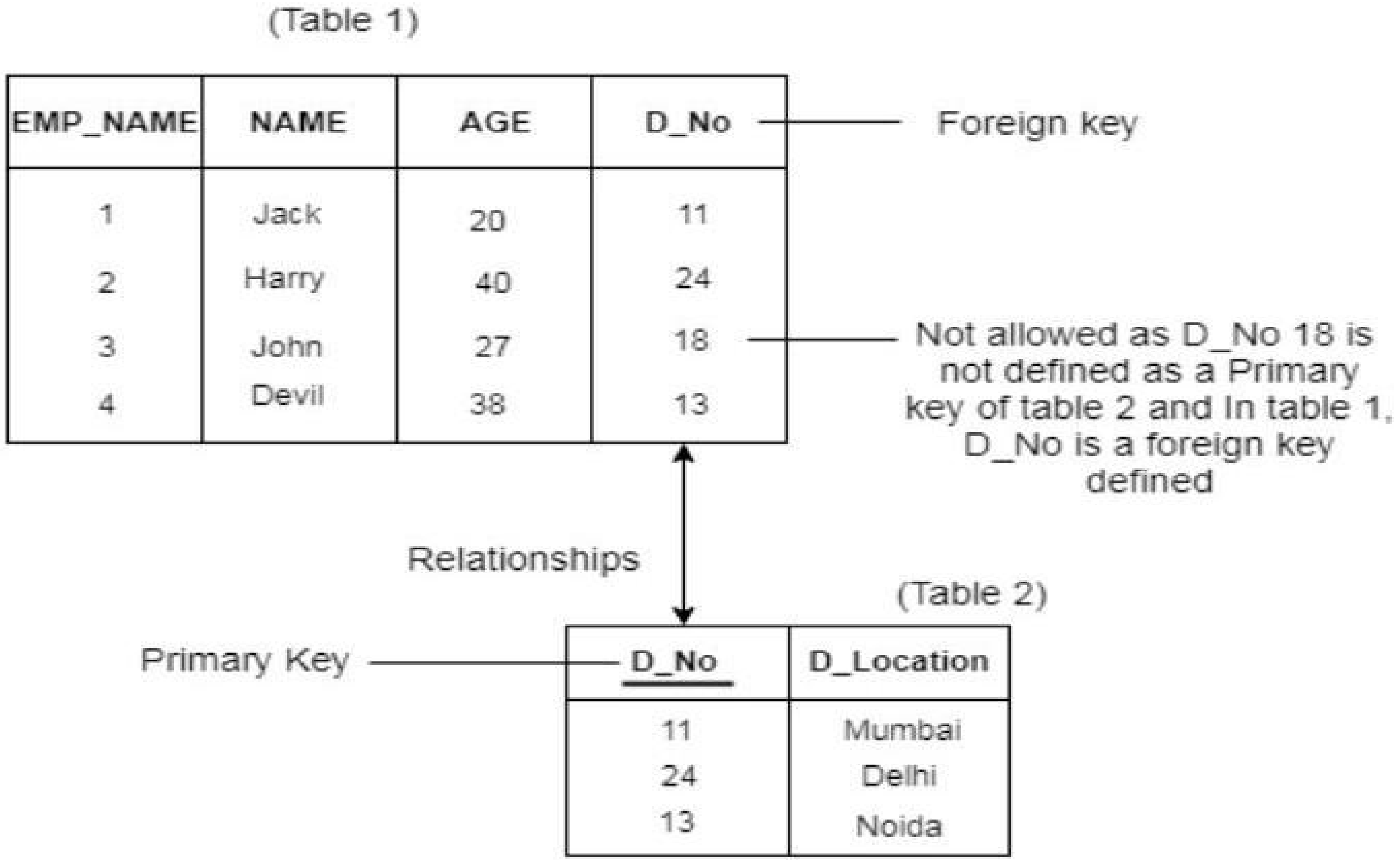
Dept_no	Dept_name
D10	ASET
D11	ALS
D12	ASFL
D13	ASHS

Here,

The relation ‘Student’ does not satisfy the referential integrity constraint.
This is because in relation ‘Department’, no value of primary key specifies department no. 14.
Thus, referential integrity constraint is violated.

- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.

Example:



(Table 1)

EMP_NAME	NAME	AGE	D_No
1	Jack	20	11
2	Harry	40	24
3	John	27	18
4	Devil	38	13

Foreign key

Not allowed as D_No 18 is not defined as a Primary key of table 2 and In table 1, D_No is a foreign key defined

Relationships

(Table 2)

Primary Key

D_No	D_Location
11	Mumbai
24	Delhi
13	Noida

The End