Mapping ER Diagrams To Relation Data Model (ER2RDM Mapping)

CE384: Database Design
Maryam Ramezani
Sharif University of Technology
maryam.ramezani@sharif.edu

01

Introduction

Translating Relationship Set into Tables

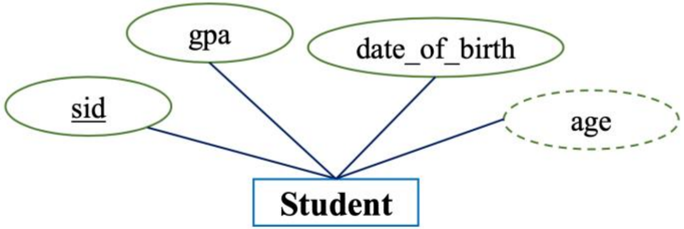
- Conceptual ER-models allow you to more accurately represent the subject area than logical models (relational, network, etc.).
- But, there are no DBMSs that support ER models.
- So, ER diagram is converted into the tables in Relational Data Model (RDM or RM).
- Relational models can be easily implemented by RDBMS like mySQL, MS SQL, PostgreSQL, Oracle etc.

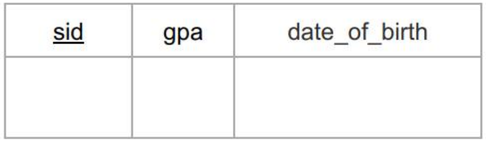
Translating Relationship Set into Tables

- The **ER2RDM mapping method** is based on the formation of a set of initial relation tables from ER-diagrams (initial logical model) and based on this factors – atomic and multivalued of attribute, cardinality (max=one-many) and obligation (min=optional-mandatory) of relationship.
- At the next stage, the initial logical RD model are optimized (**normalized**).

3 Simple Rules for Entities

- A table is created for each entity:
  - 01) Each simple entity attribute corresponds to a current table column, derived entity attribute removed from a current table.
    - The primary key of the table will be the key attribute of the entity set.



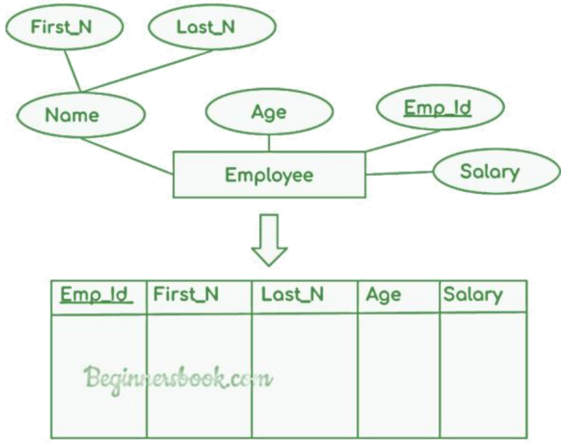


sid	gpa	date_of_birth

Schema: Student(sid, gpa, date_of_birth)

3 Simple Rules for Entities

- 02) Each element of composite attribute corresponds to a current table column. While conversion, simple attributes of the composite attributes are taken into account and not the composite attribute itself.

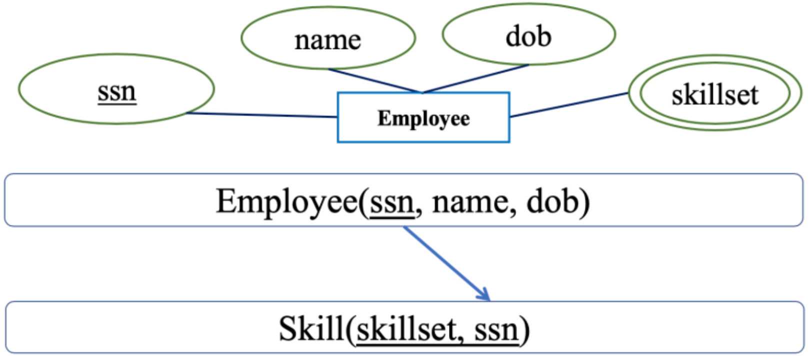


Emp_Id	First_N	Last_N	Age	Salary

Table Schema: (Emp_id, First_N, Last_N, Age, Salary)

3 Simple Rules

- 03) A strong entity set with any number of multivalued attributes will require 2 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 multivalued attributes.

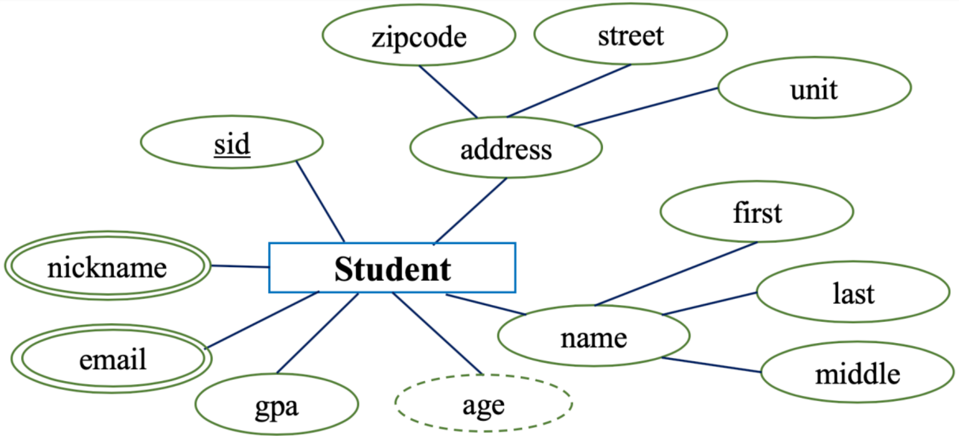


Employee(ssn, name, dob)

Skill(skillset, ssn)

Key Attribute Migration (tranzition).

Let’s Practice



Student(sid, zipcodeAddr, streetAddr, unitAddr, firstName, middleName, lastName, gpa, dob)

Nickname(sid, nickname)

Email(sid, email)

Some Notations

- We represent the primary key with a continuous underline.

Empolyee(Emp_no,Emp_name,Salary)

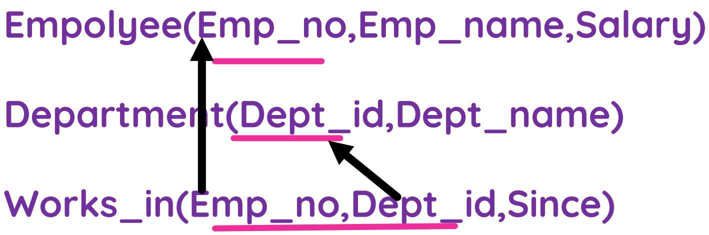
- We represent the foreign key with a dotted underline.

Empolyee(Emp_no,Emp_name,Salary)

Department(Dept_id,Dept_name)

Works_in(Emp_no,Dept_id,Since)

- The attribute name of a relationship that has a foreign key to another relation is either kept the same (to make the reference clear), or we draw an arrow between them.



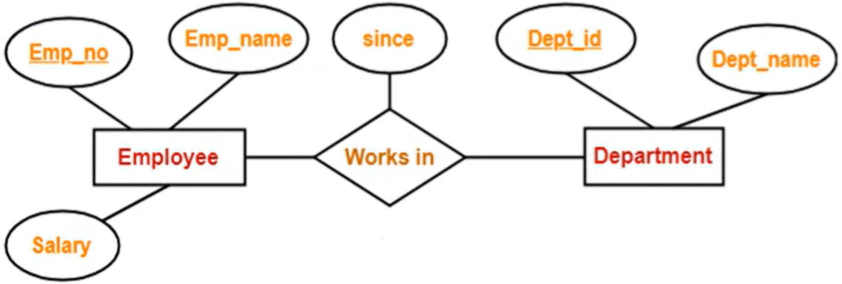
Empolyee(Emp_no,Emp_name,Salary)

Department(Dept_id,Dept_name)

Works_in(Emp_no,Dept_id,Since)

Relationship to Table

- One table for each entity type.
- One table for relationship type with:
  - Attributes are:
    - Primary key of participating entity sets.
    - Its own descriptive attributes if any.
  - Primary key:
    - Set of non-descriptive attributes



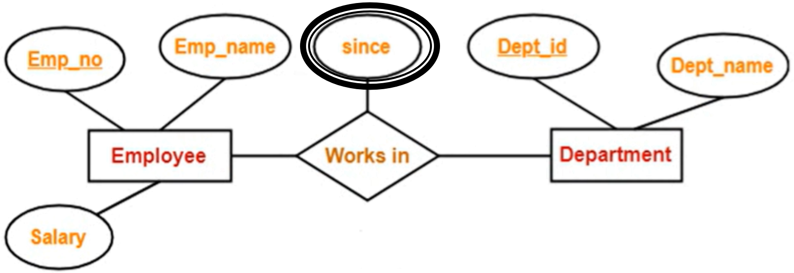
Empolyee(Emp_no,Emp_name,Salary)

Department(Dept_id,Dept_name)

Works_in(Emp_no,Dept_id,Since)

Relationship to Table

- If the relationship is unique by “since” attribute. “since” is a multivalued attribute, then its in the primary key of “Works_in” table.



Empolyee(Emp_no,Emp_name,Salary)

Department(Dept_id,Dept_name)

Works_in(Emp_no,Dept_name,Since)

02

Binary Relationship with Cardinality 1:1

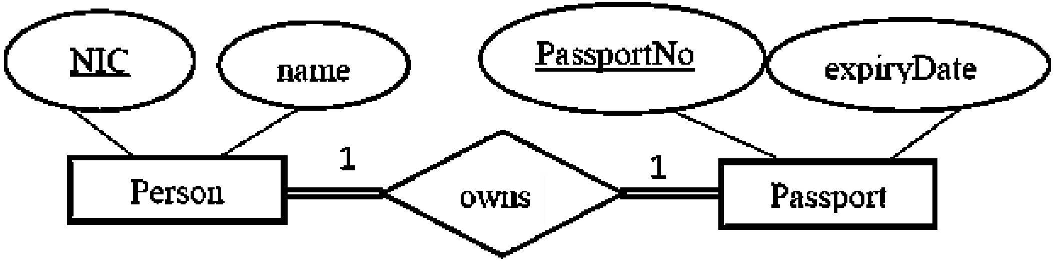
Three Possible Approaches

1) Merged Relation

2) Foreign key

3) Cross-reference

1) Both sides Total Participation



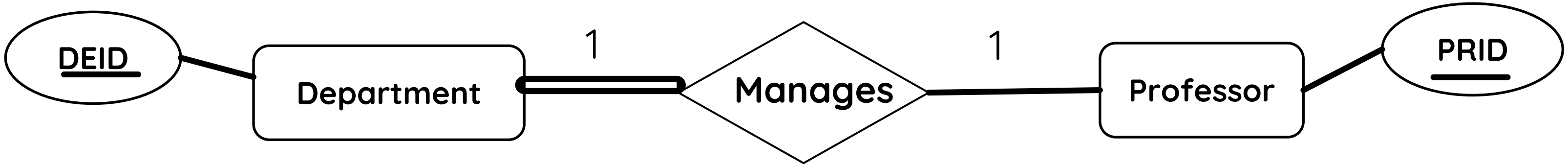
- Merged relation approach
  - One table by combine both entities and relationship.
  - Assign one PK from any of the entity types.

Person_passport(NIC,name,PassportNo,expiryDate)

OR

Person_passport(NIC,name,PassportNo,expiryDate)

2) One side Total Participation

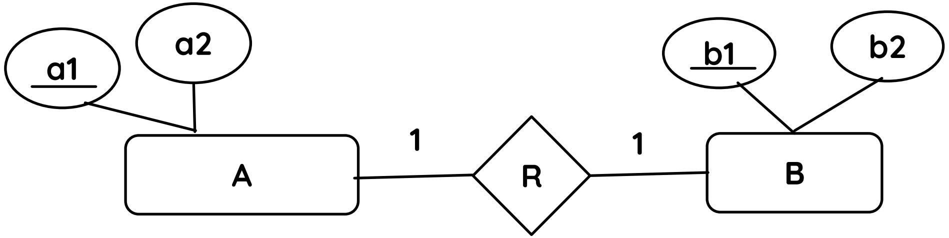


- Foreign key approach
  - Two table.
  - PK must to go Total Participation side as FK.

Department(DEID , PRID)

Professor(PRID)

3) Both side Partial Participation



- Merged relation approach
  - PK can go to either side.

A (a1 , a2)

BR (b1 , b2 , a1)

OR

AR (a1 , a2 , b1)

B (b1 , b2)

- Cross-reference approach
  - When number of participations are very low, maybe three table will be better to avoid null values:

A (a1 , a2)

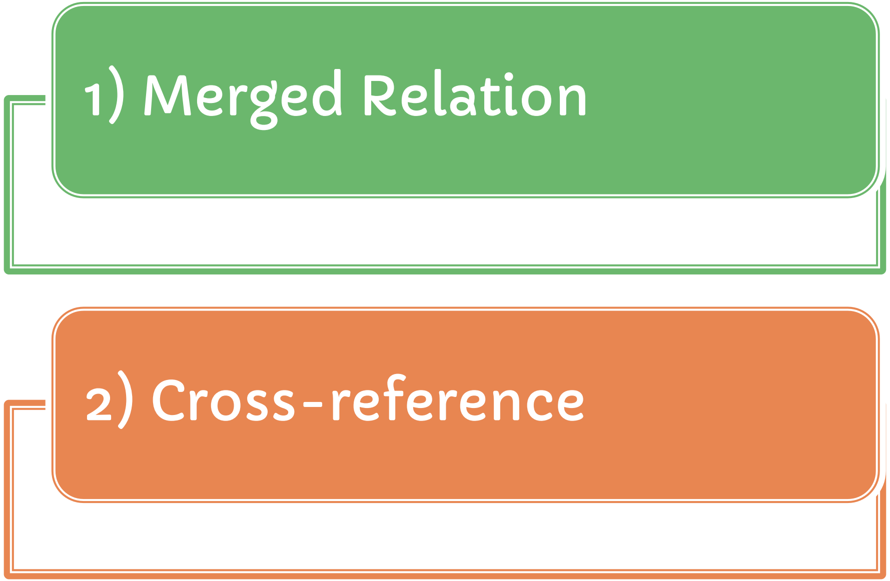
B (b1 , b2)

R (a1,b1)

03

Binary Relationship with Cardinality 1:N

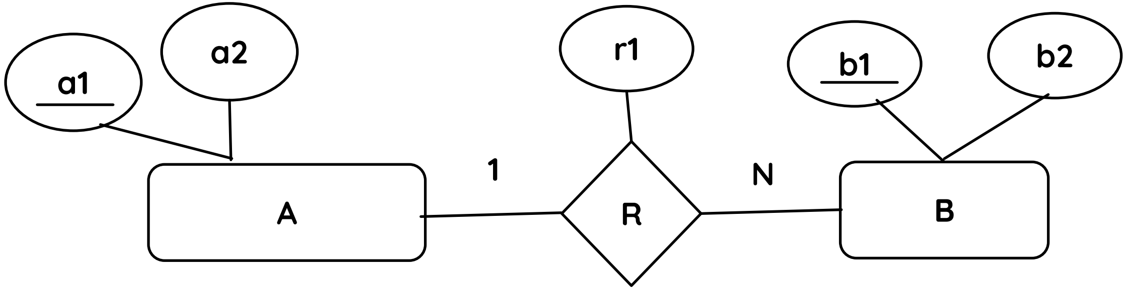
Two Possible Approaches



1) Merged Relation

2) Cross-reference

1) Strong Entity Types



- First solution: Merged relation approach
  - Two tables for two entities
  - PK of 1 side go to N side
  - **Note: If there are any descriptive attributes they also go to the N side (Wherever the FK goes, descriptive attributes goes there)**

A (a1 , a2)

BR (b1 , b2 , a1, r1)

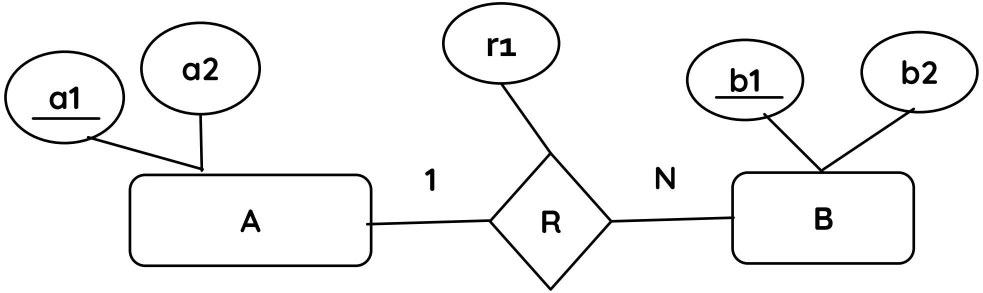
1) Strong Entity Types - Continue

- Second solution: Cross-reference approach
  - Three tables:

A (a1 , a2)

B (b1 , b2)

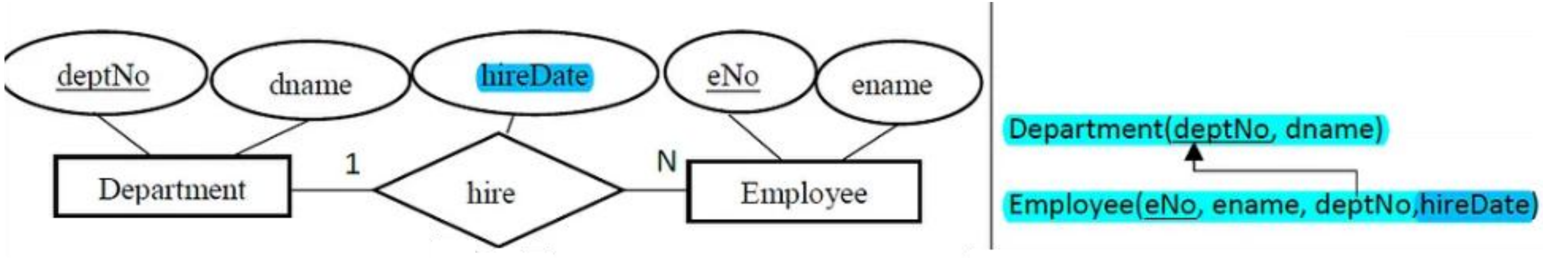
R (a1,b1,r1)



- When second solution is preferred?
  - To avoid null values in table “BR”: Number of “B” entity set not participated in “R” relationship is large (so, “B” must be partial participated).
  - The frequency of reference to the relation “R” is high while other attributes with a lower frequency are needed.
  - Attributes of “R” is too large, which leads to large columns for table “B”.

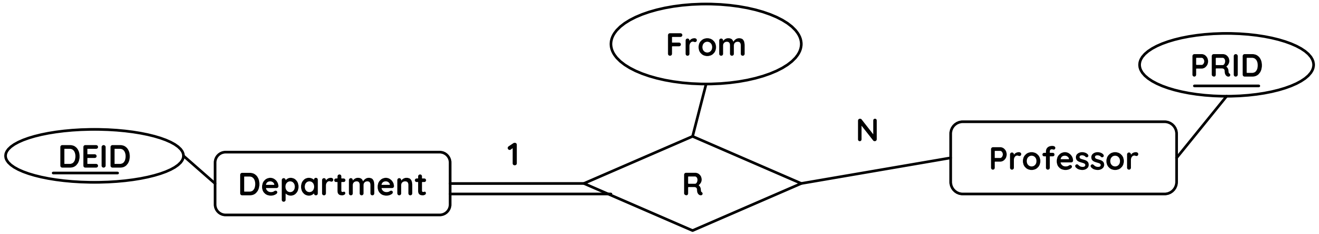
1) Strong Entity Types - Continue

- Examples



Department(deptNo, dname)

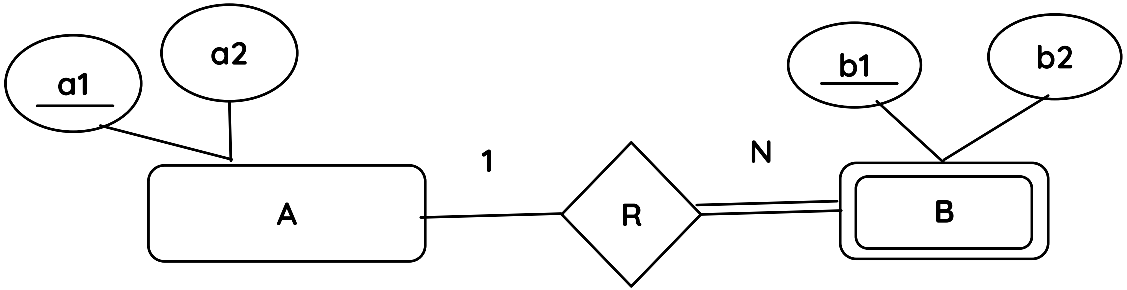
Employee(eNo, ename, deptNo,hireDate)



DEP (DEID)

PRO (PRID , DEID, From)

3) Strong Entity Type and Weak Entity Type



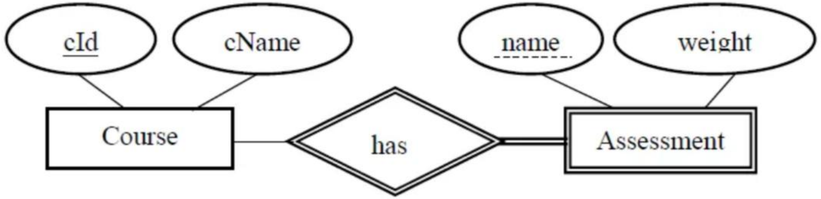
- PK of Owner Entity goes to combine with the Partial Key of the Weak Entity to form the PK.

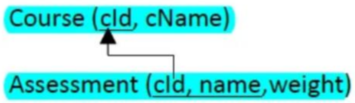
A (a1 , a2)

B (b2 , b1 , a1)

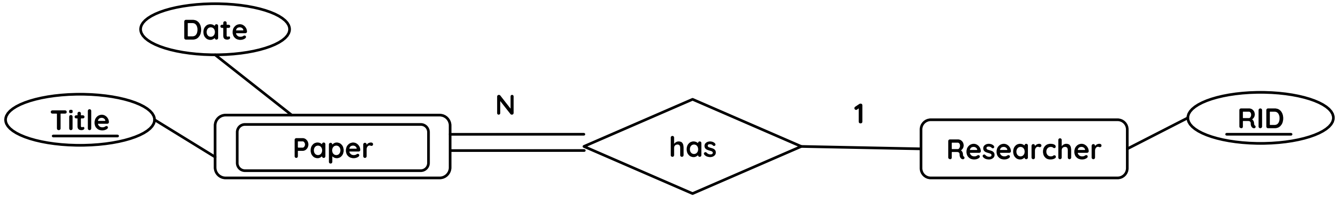
3) Strong Entity Type and Weak Entity Type

- Examples




Course (cId, cName)

Assessment (cId, name,weight)



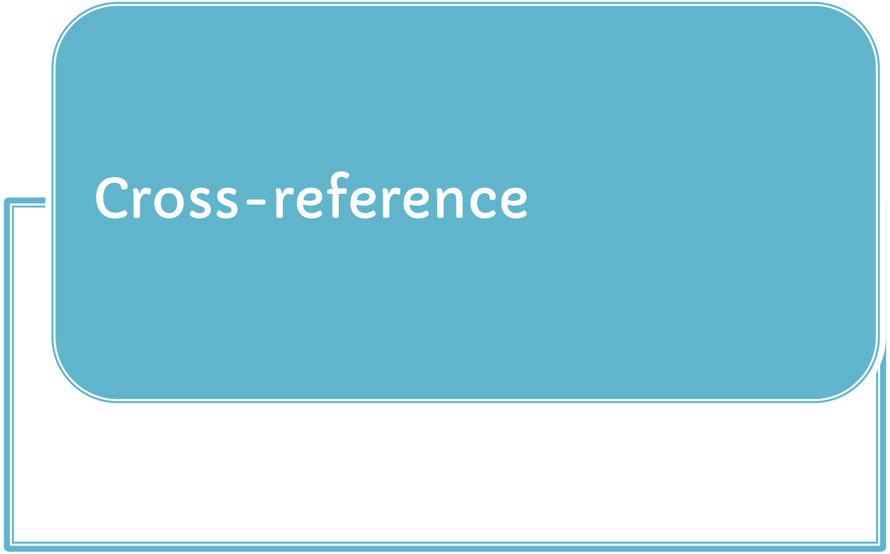
Paper (PRID,Title,Date)

Researcher (RID)

04

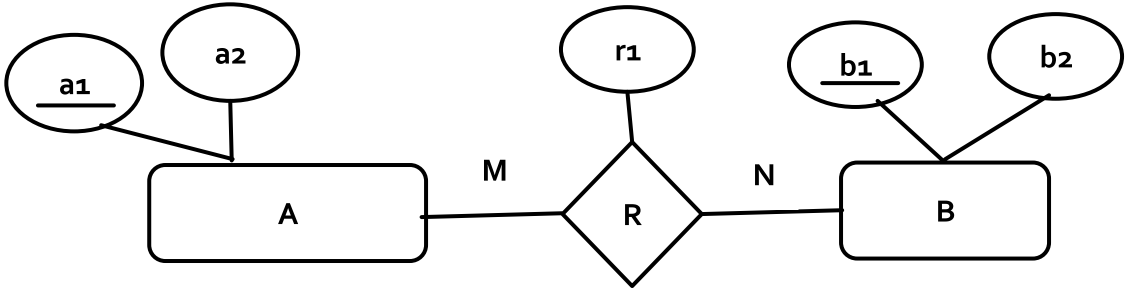
Binary Relationship with Cardinality M:N

One Possible Approaches



Cross-reference

1) Single Attribute for Relationship



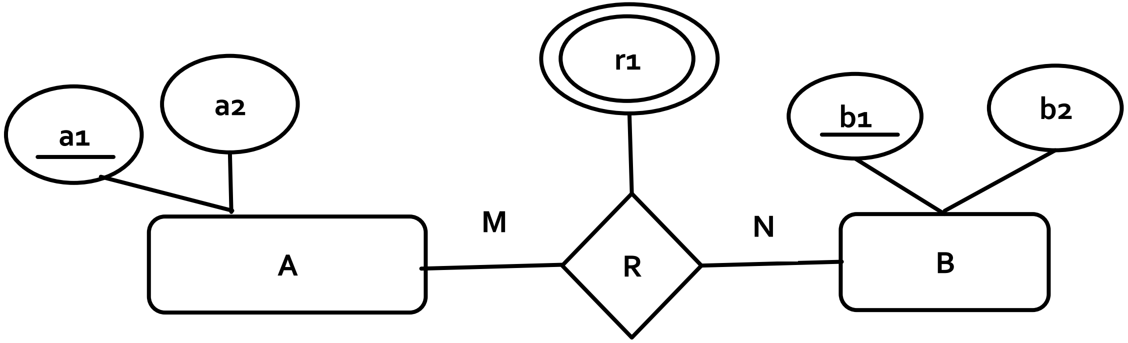
- A table/relation for the Relationship is created including the PK’s of the participating entities and descriptive attributes, if any.

$A\ (\underline{a1}\ ,\ a2)$

$B\ (\underline{b1}\ ,\ b2)$

$R\ (\underline{\underline{a1}\ ,\ \underline{b1}}\ ,\ r1)$

2) Multivalued Attribute for Relationship

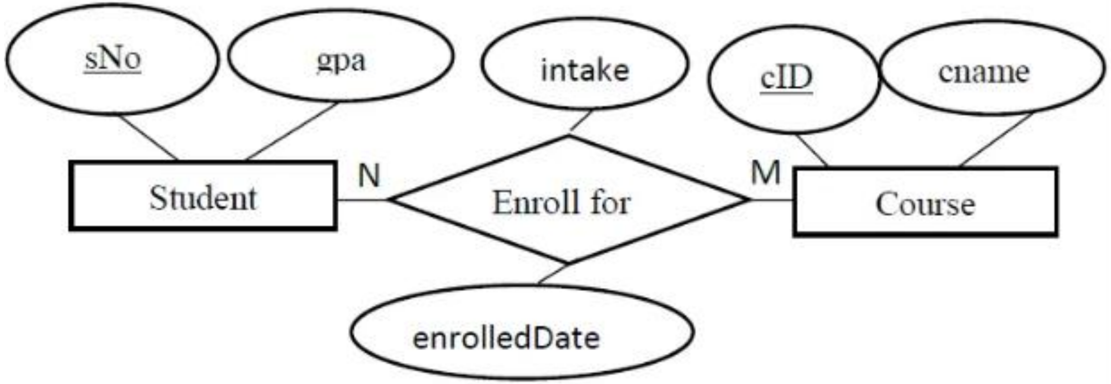


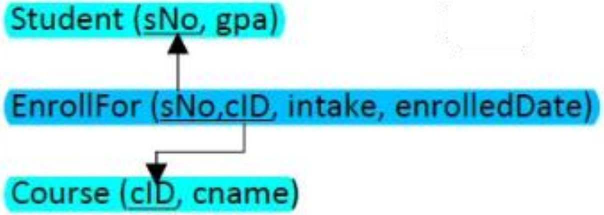
A (a1 , a2)

B (b1 , b2)

R (a1 , b1 , r1)

Examples





Student (sNo, gpa)

EnrollFor (sNo,cID, intake, enrolledDate)

Course (cID, cname)

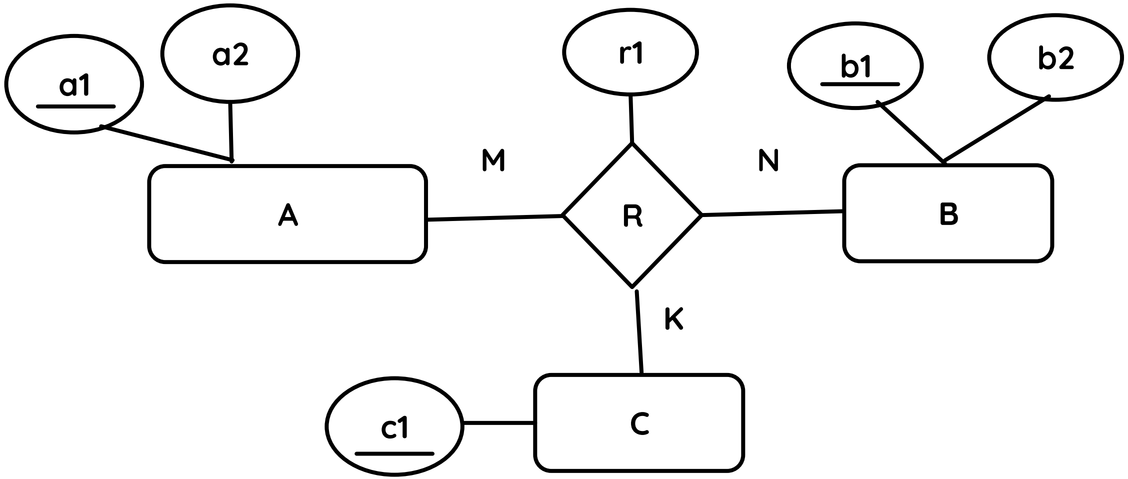
05

Ternary Relationship with Cardinality M:N:K

One Possible Approaches

Cross-reference

1) Single Attribute for Relationship



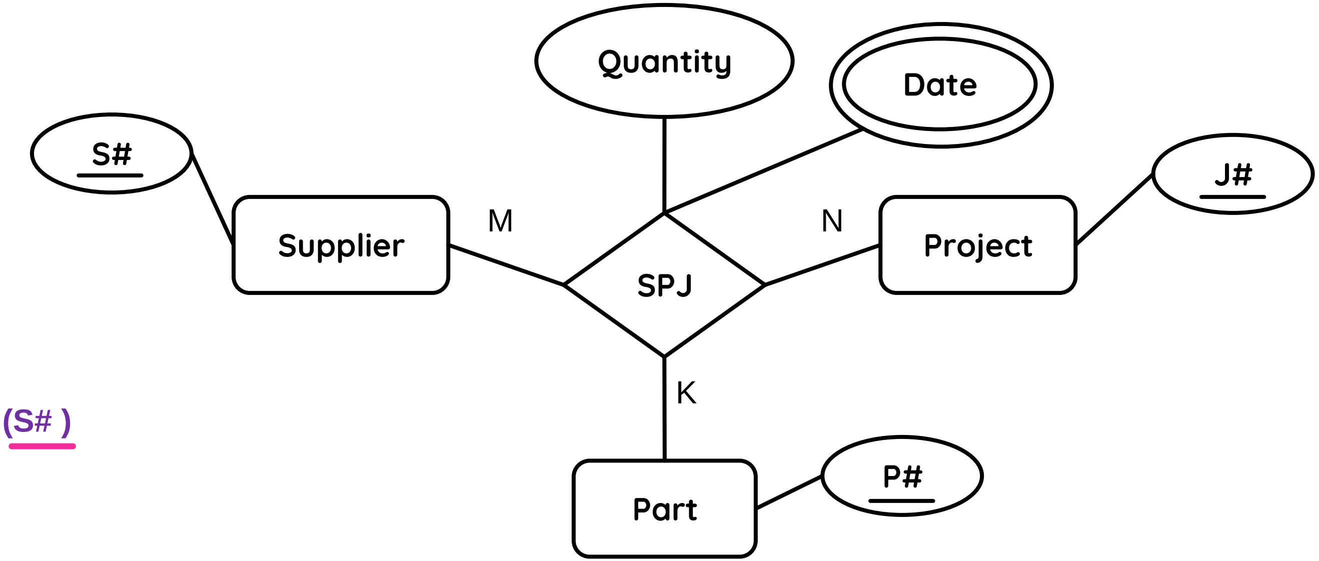
A ($\underline{\text{a1}}$, a2)

B ($\underline{\text{b1}}$, b2)

C ($\underline{\text{c1}}$)

R ($\underline{\text{a1 , b1 , c1}}$, r1)

Example: Multivalued Attribute for Relationship



Supplier (S#)

Part (P#)

Project (J#)

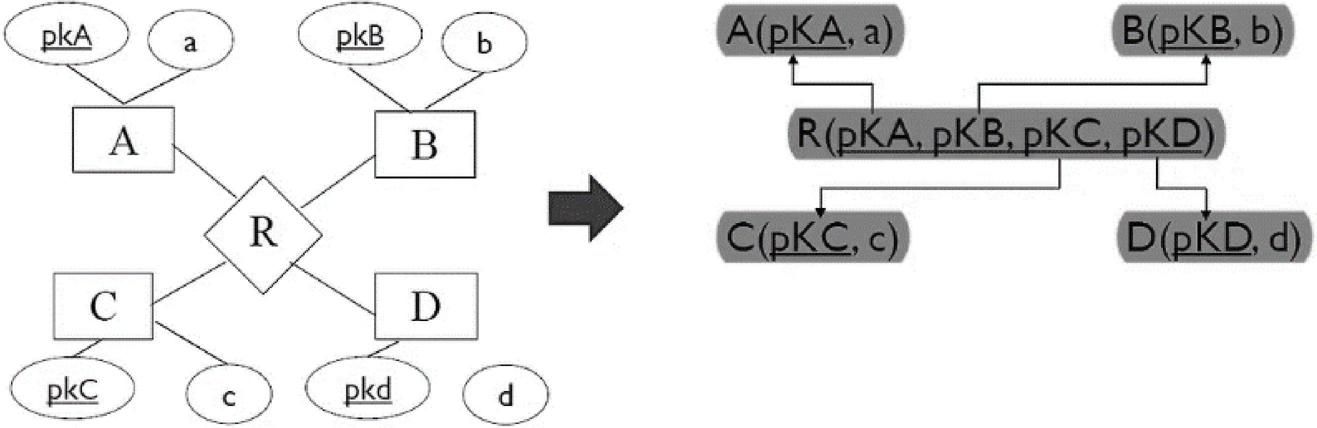
SPJ (S# , J# , P# , Date, Quantity)

06

N-ARY Relationship

N-ARY Relationship

- N-ary relationship is mapped in to a “Relationship” relation and foreign keys.
  - “N” means Degree greater than 2
  - Review: Degree = No of Entities attached to the relationship.



A(pKA, a)

B(pKB, b)

R(pKA, pKB, pKC, pKD)

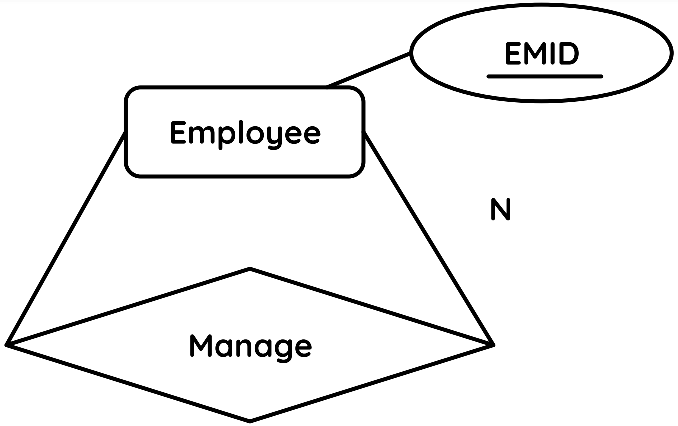
C(pKC, c)

D(pKD, d)

07

Unitary Relationship with Cardinality 1:N

1:N Unitary Relationship



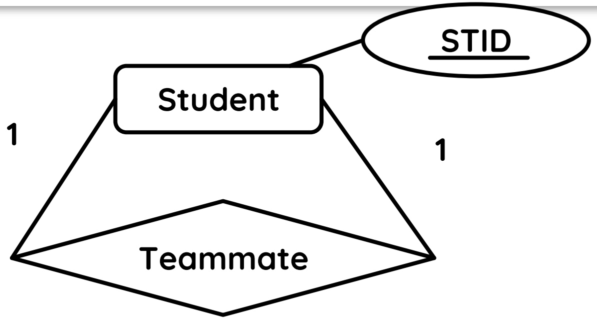
- One table:

EMPL (<u>EMID</u> , EMGRID)

08

Unitary Relationship with Cardinality 1:1

1:1 Unitary Relationship



- Solution for when there are not many people without group members.

EMPL (EMID , EMGRID)

Unique

- Solution for when there are many people without group members.

EMPL (EMID)

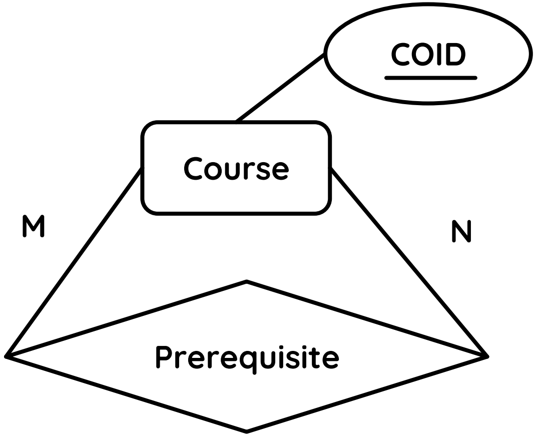
TEAMMATE (EMID , EMGRID)

Unique

09

Unitary Relationship with Cardinality M:N

M:N Unitary Relationship

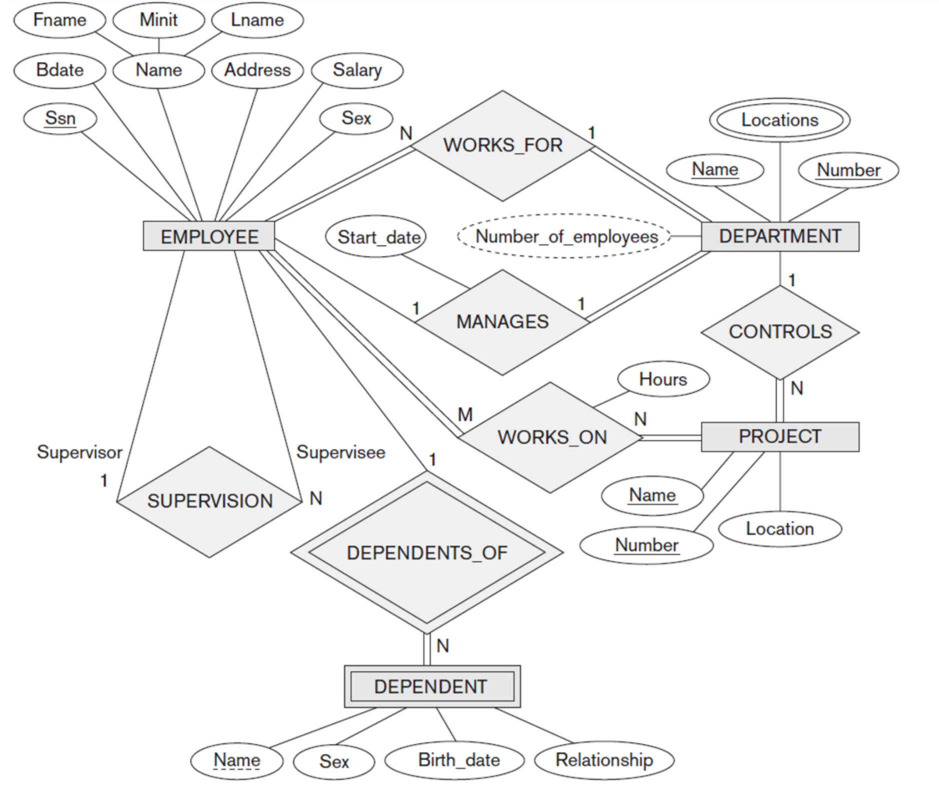


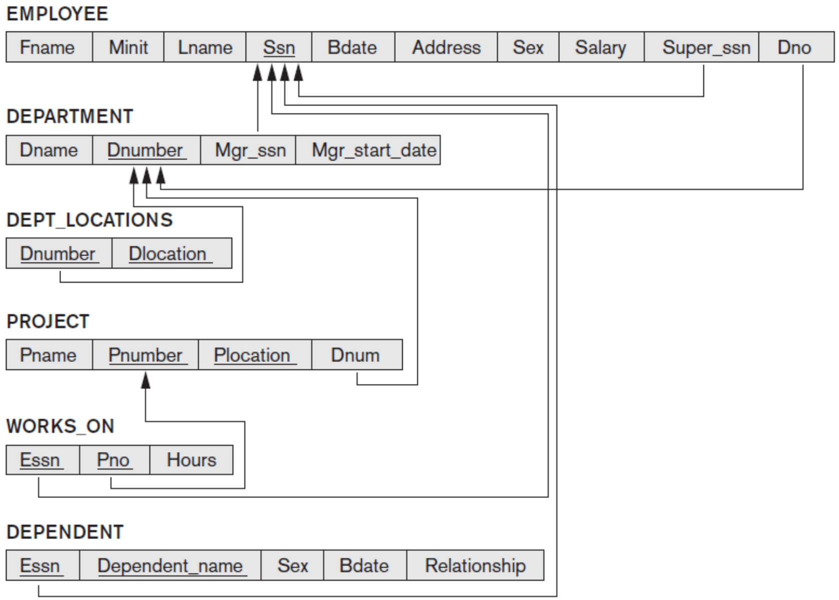
- Two tables:

Prerequisite (COID , PRECOID)

Course (COID)

Example





EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

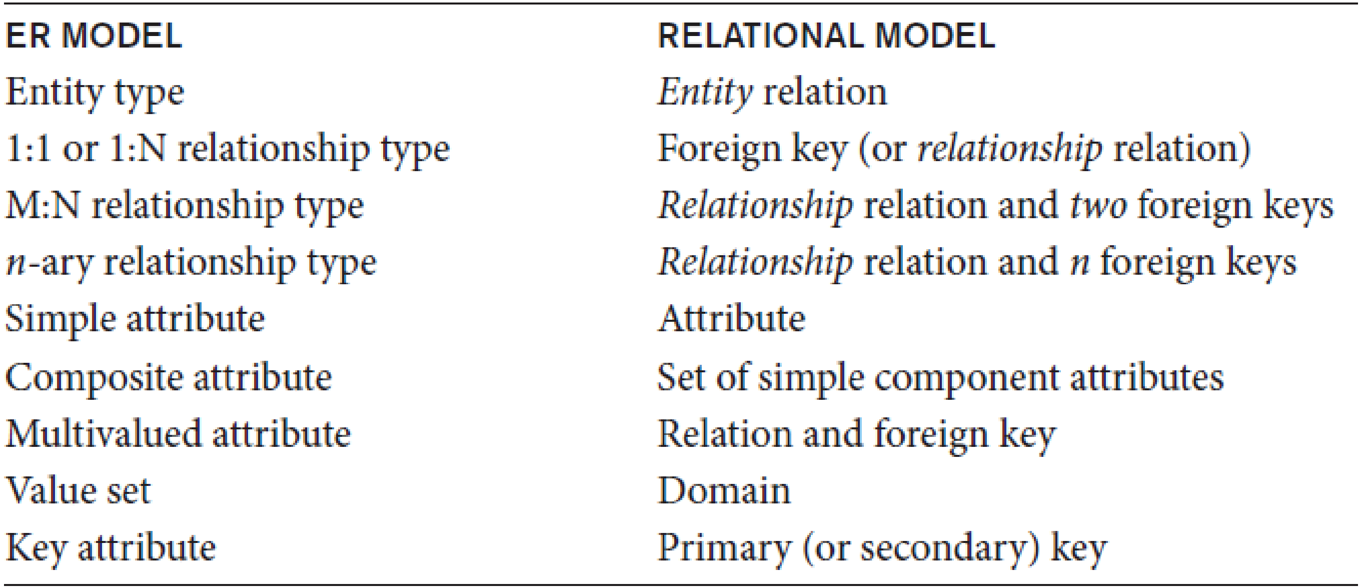
Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Conclusion

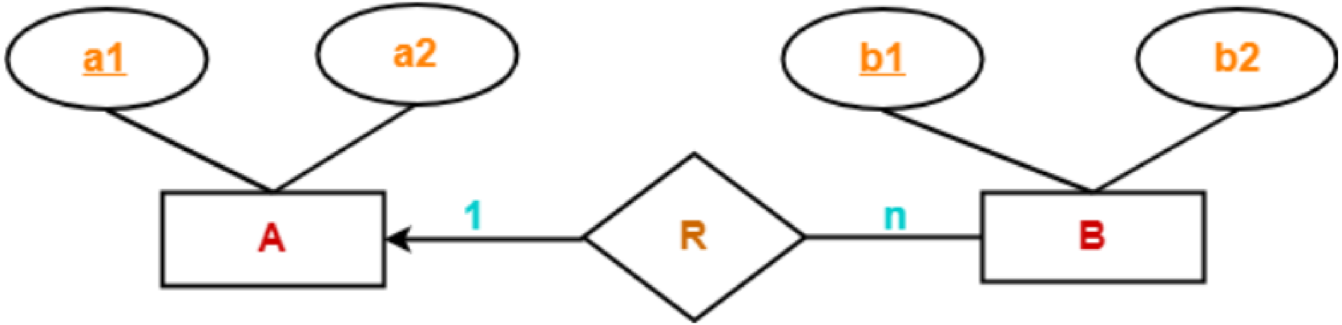
- Correspondence between ER and Relational Models



ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

Notes

- In some references, the ER notation is as the follows, where the **arrow** indicates cardinality **1** and **line** indicates cardinality.



10

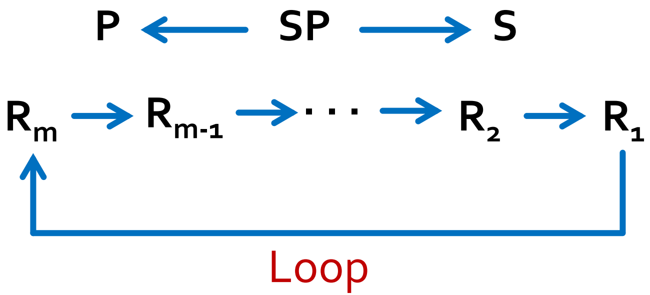
Referential Integrity

Referential Integrity Rules

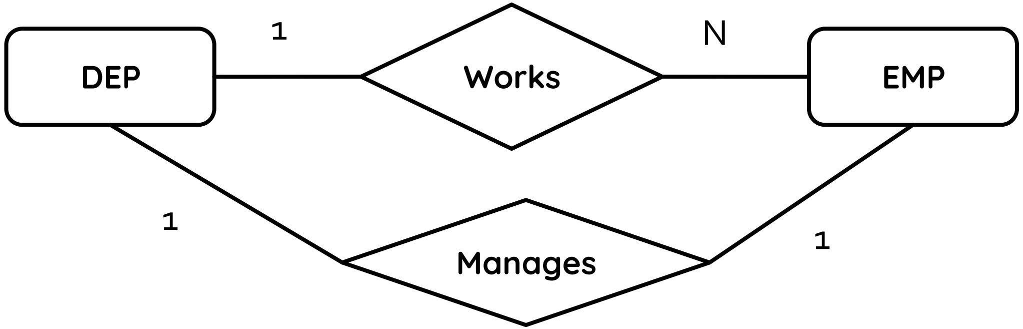
- The main rules of referential integrity are to ensure that:
  - A foreign key value in one table corresponds to an existing primary key value in another table.
  - When a primary key value is deleted, all foreign key values that reference it are also deleted or set to null.
  - When a primary key value is updated, all foreign key values that reference it are also updated.
- Referential integrity is enforced by creating relationships between tables and enforcing integrity constraints, such as the use of foreign key constraints, which ensure that referential integrity is maintained in the database.

Referential Integrity Graph

- We can diagrammatically display referential integrity constraints by drawing a directed arc from each foreign key to the relation it references.
- For clarity, the arrowhead may point to the primary key of the referenced relation.



Loop-Referencing with two relationships



E#: Employee ID of the manager

D#: Department ID

DEPT (D#, DTITLE, … , E#)

Unique

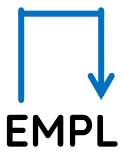
EMPL (E#, ENAME, …, D#)

DEPT ⇄ EMPL

Self-Referencing (Loop-Referencing with one relationship)

Manager ID with renamed name

EMPL (E#, ENAME, ELASTNAME, …, EPHONE, EMANAGER#)



EMPL

Loop-Referencing with three relationships

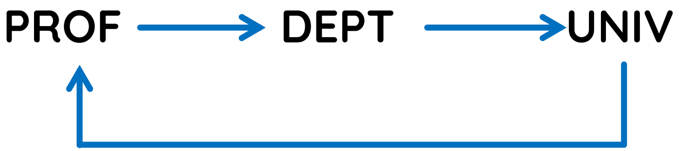
Dep of prof

PROF (PRID, PRNAME, …, DEID)

DEPT(DEID, DTITLE, …., UNID)

UNIV(UNID, UNAME, …, UNPRESNUM)

PRID of manager of university



PROF DEPT UNIV

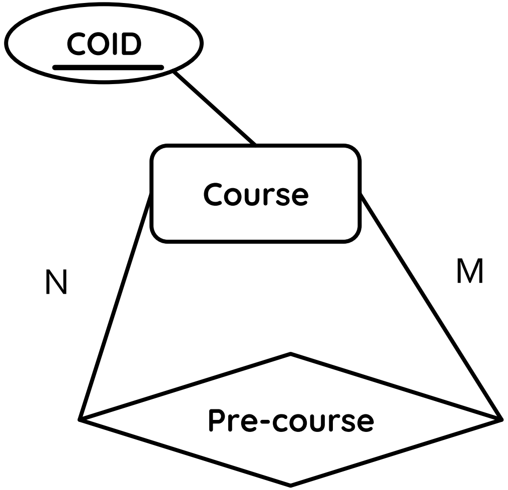
- Draw the ER of this logical model!

Note

- Does the loop in ER necessarily make a loop-referencing?
  - No!!! Look at the cardinality of relationships! Example:

COT (COID, …)

COPRECO(COID, PRECO)

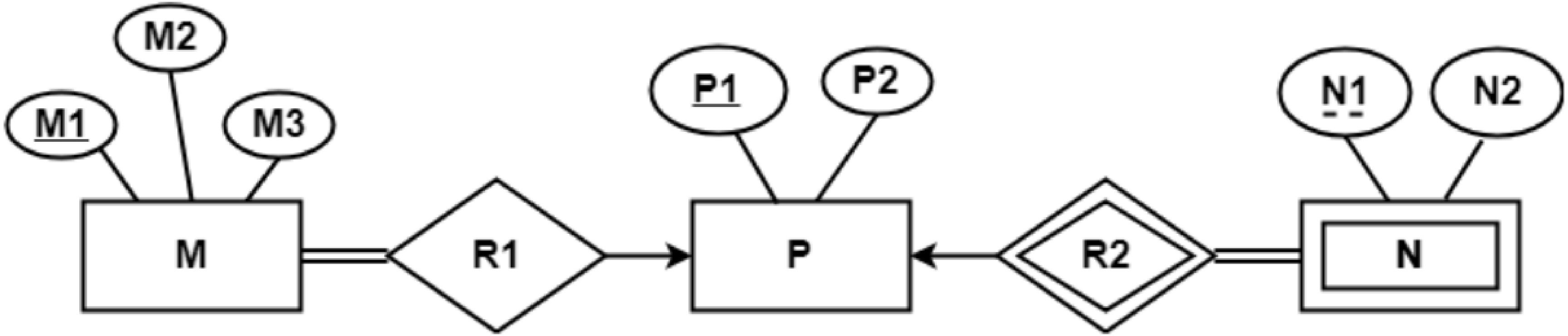


11

Practice Problems

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

Example #1



Solution:

MR1(M1, M2, M3, P1)

P(P1, P2)

NR2(P1, N1, N2)

Example #2

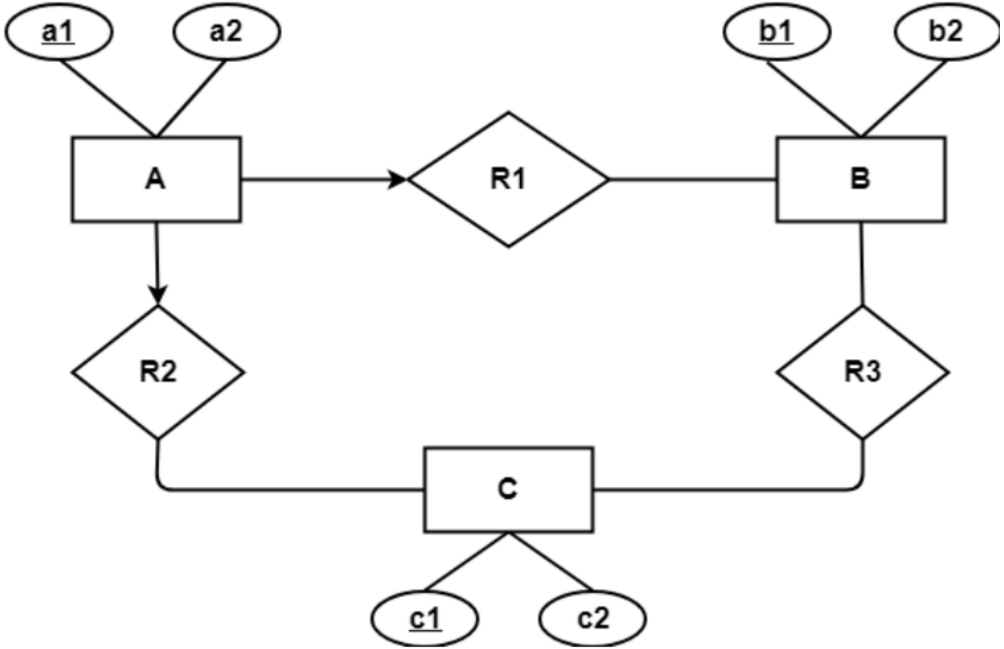
Solution:

AR1R2($\underline{\text{a1, b1, c1}}$, a2)

B($\underline{\text{b1}}$, b2)

C($\underline{\text{c1}}$, c2)

R3($\underline{\text{b1, c1}}$)



References

- Chapter 9 of FUNDAMENTALS OF Database Systems, SEVENTH EDITION
- Chapter 6 Part 7 of DATABASE SYSTEM CONCEPTS, SIXTH EDITION.
- Chapter 4 of Database Systems A Practical Approach to Design, Implementation, and Management, SIXth edition