Reducing ER Diagrams to Relational Schemas

Database Design

Department of Computer Engineering

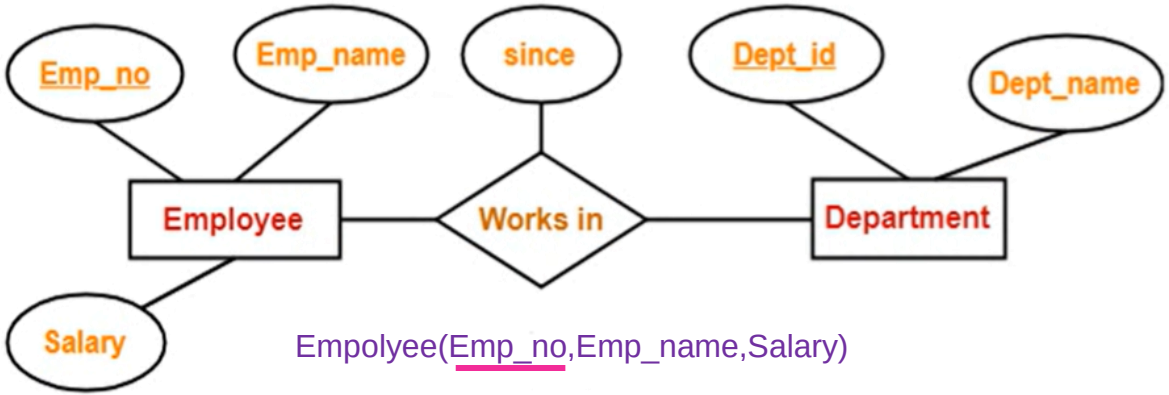
Sharif University of Technology

Maryam Ramezani maryam.ramezani@sharif.edu

Translating Relationship Set into a Table

Relationship to Table

- One table for each entity type.
- One table for relationship type with:
  - Primary key of participating entity sets.
  - Its own descriptive attributes if any.
  - Set of non-descriptive attributes will be the primary key of this table.



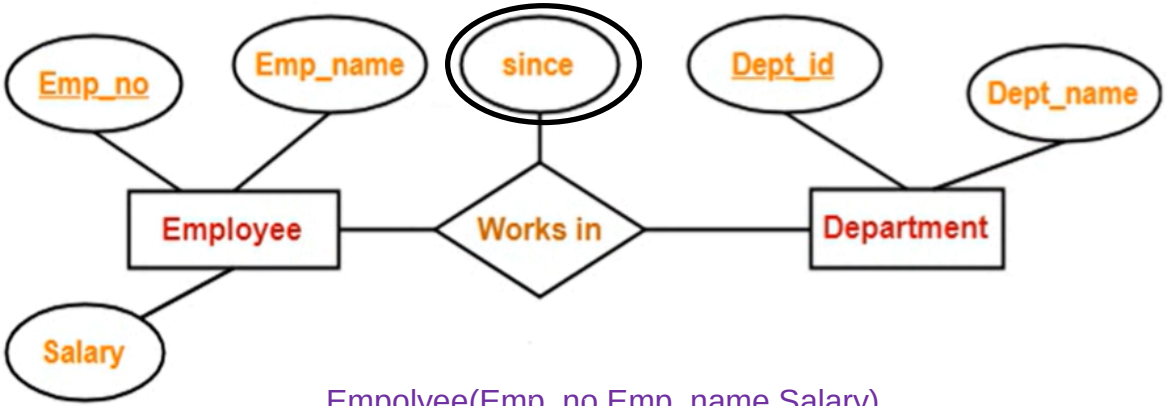
Empolyee(Emp_no,Emp_name,Salary)

Department(Dept_id,Dept_name)

Works_in(Emp_no,Dept_name,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.



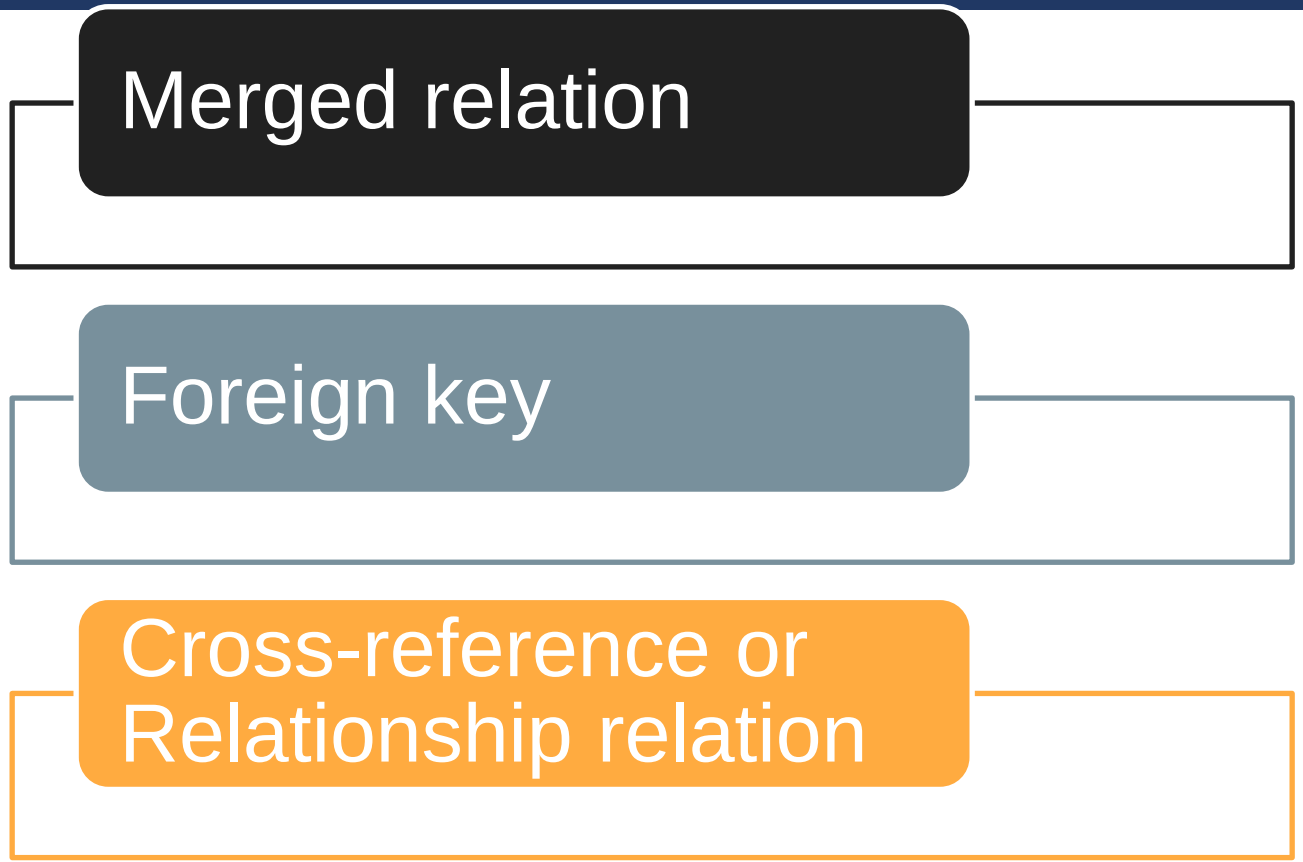
Empolyee(Emp_no,Emp_name,Salary)

Department(Dept_id,Dept_name)

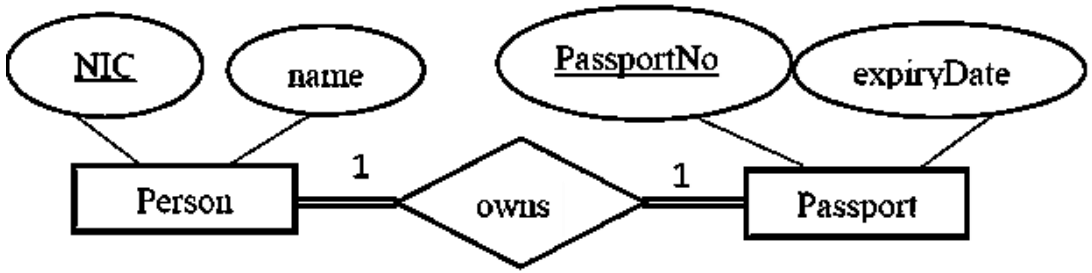
Works_in(Emp_no,Dept_name,Since)

1) Binary Relationship with Cardinality 1:1

Three Possible Approaches



1-1) Both sides Total Participation



Merged relation approach

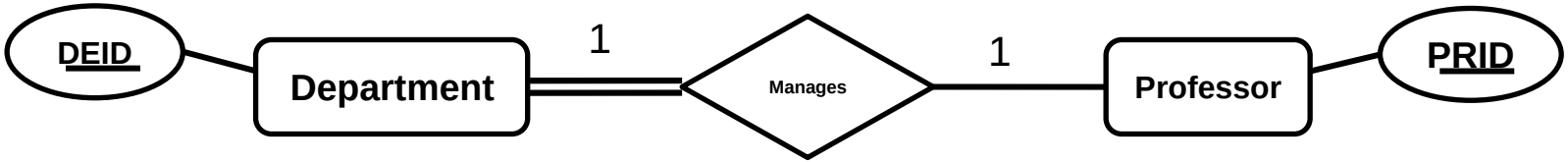
- One table by combine both entities and relationship.
- Assign one PK from any of the entity types.

Person_passport(NIC,name,PassportNo,expiryDate)

OR

Person_passport(NIC,name,PassportNo,expiryDate)

1-2) One side Total Participation



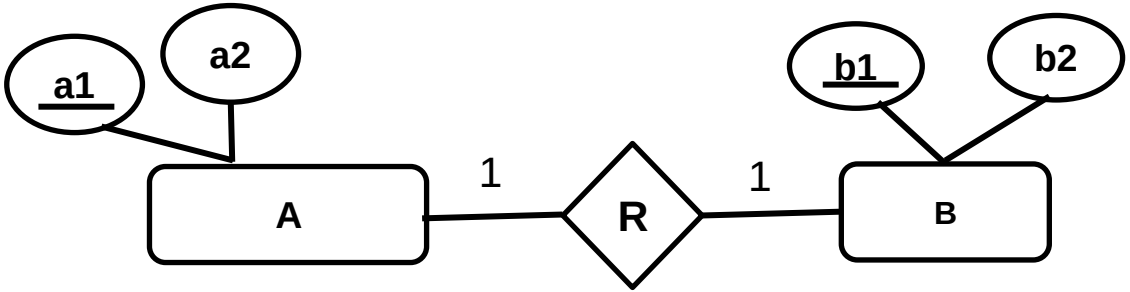
Foreign key approach

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

Department(DEID , PRID)

Professor(PRID)

1-3) 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:

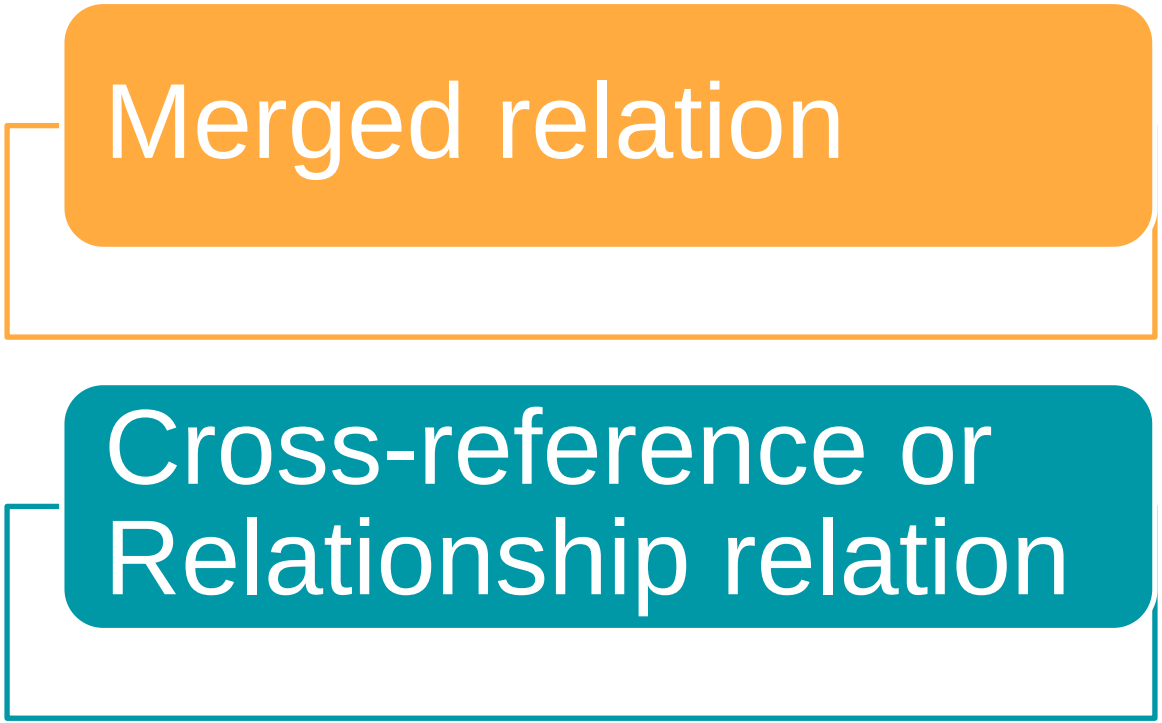
A (a1 , a2)

B (b1 , b2)

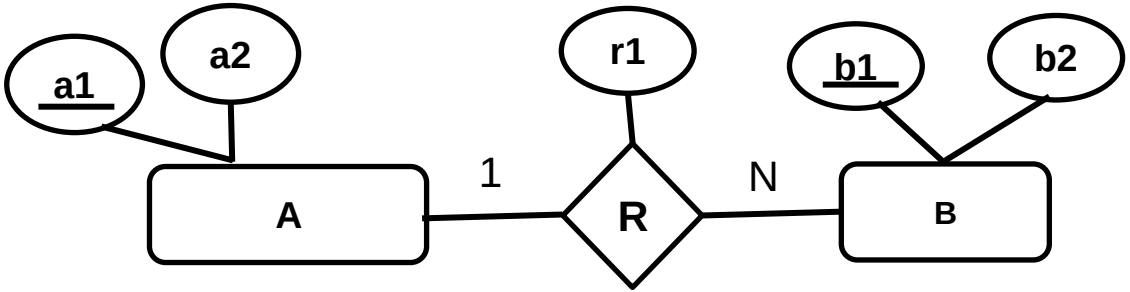
R (a1,b1)

2) Binary Relationship with Cardinality 1:N

Two Possible Approaches



2-1) Strong Entity Types

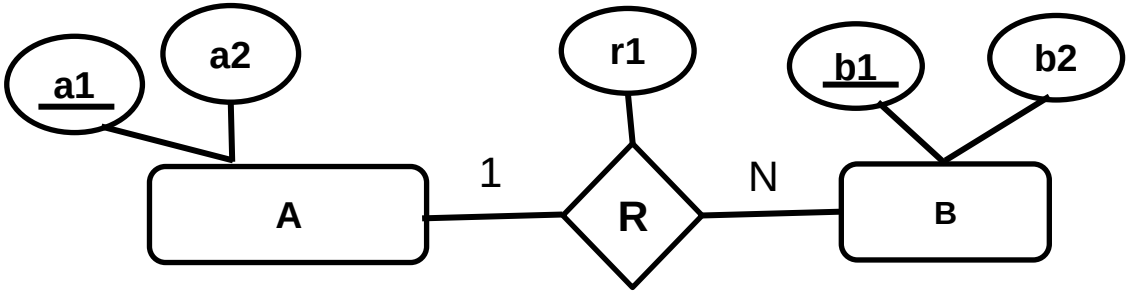


- First solution: Merged relation approach
  - PK of 1 side go to N side
  - Note: If there are any descriptive attributes they also go to the N side (Wherever the FK goes, descriptive attributes goes there)

A (a1 , a2)

BR (b1 , b2 , a1, r1)

2-1) Strong Entity Types



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

    A (a1 , a2)

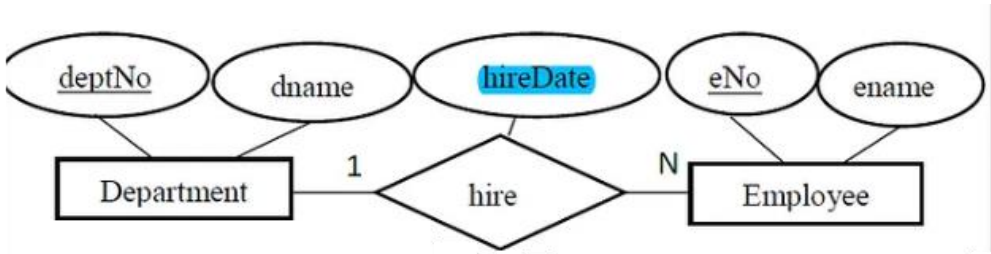
    B (b1 , b2)

    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.
  - The frequency of reference to the relation "R" is high while other attributes with a lower frequency are needed.
  - Attributes of entity type B is too large.

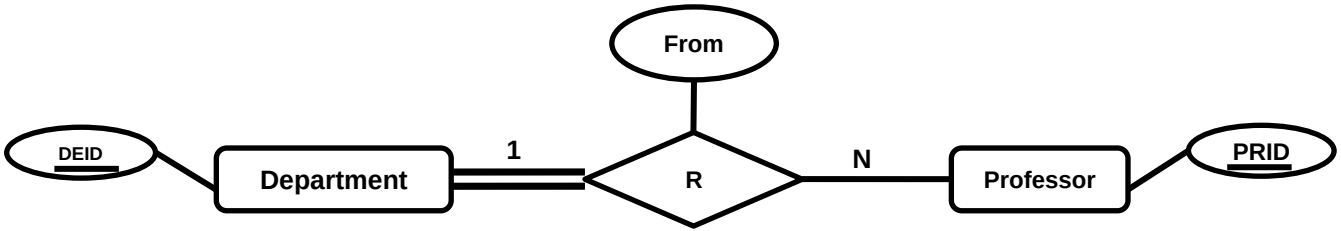
2-1) Strong Entity Types

- Examples



Department(deptNo, dname)

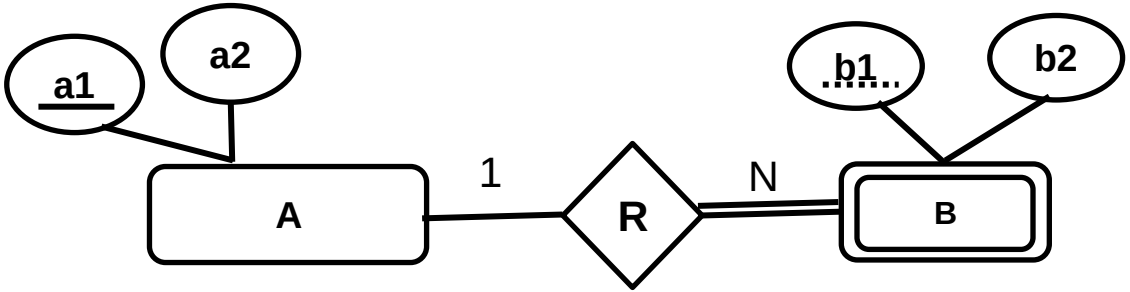
Employee(eNo, ename, deptNo,hireDate)



DEP (DEID)

PRO (PRID , DEID, From)

2-2) Strong Entity Type and Weak Entity Type



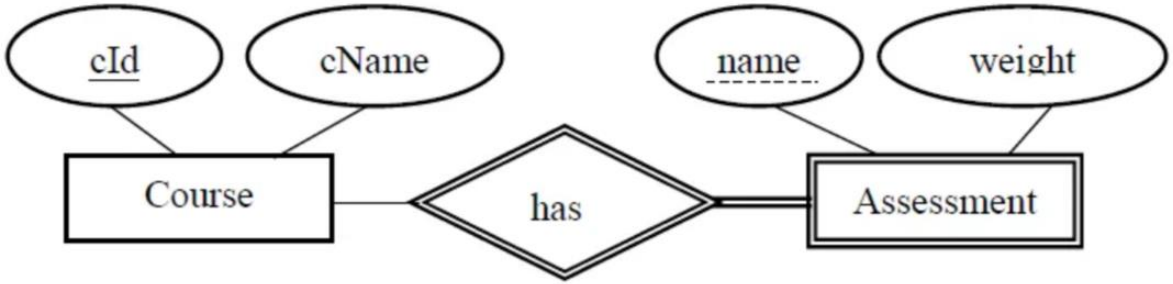
- PK of Owner Entity goes to combine with the Partial Key of the Weak Entity to form the PK.

A (a1 , a2)

B (b2 , b1 , a1)

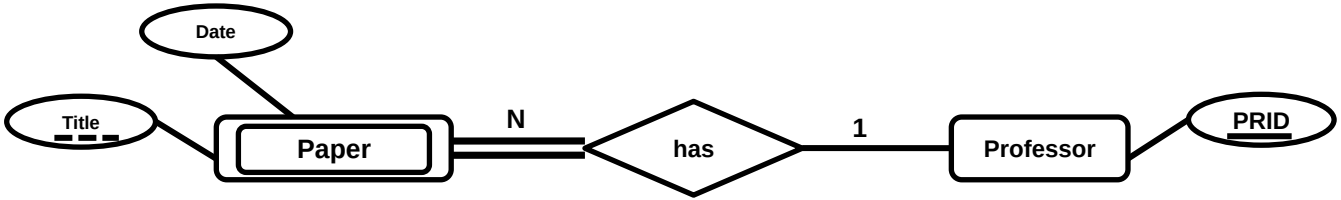
2-2) Strong Entity Type and Weak Entity Type

- Examples



Course (cId, cName)

Assessment (cId, name,weight)

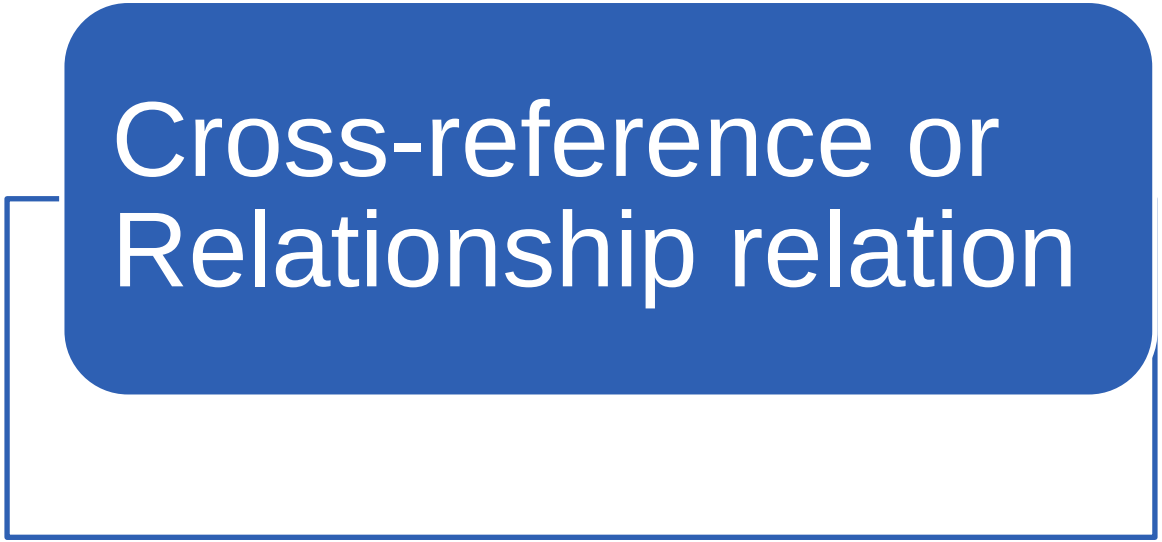


Paper (PRID,Title,Date)

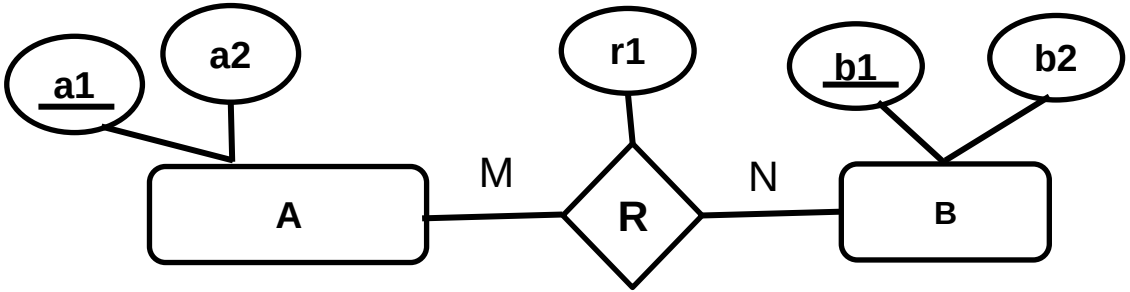
Professor (PRID)

3) Binary Relationship with Cardinality M:N

One Possible Approaches



3-1) Single Attribute for Relationship



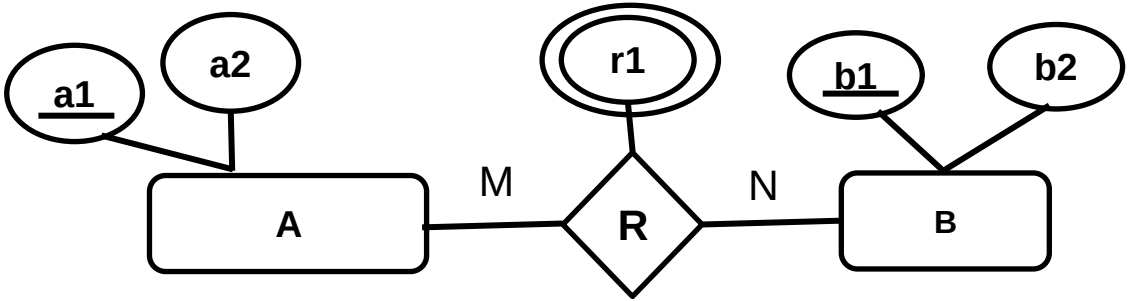
- A table/relation for the Relationship is created including the PK's of the participating entities and descriptive attributes, if any.

A (a1 , a2)

B (b1 , b2)

R (a1 , b1 , r1)

3-1) Multivalued Attribute for Relationship

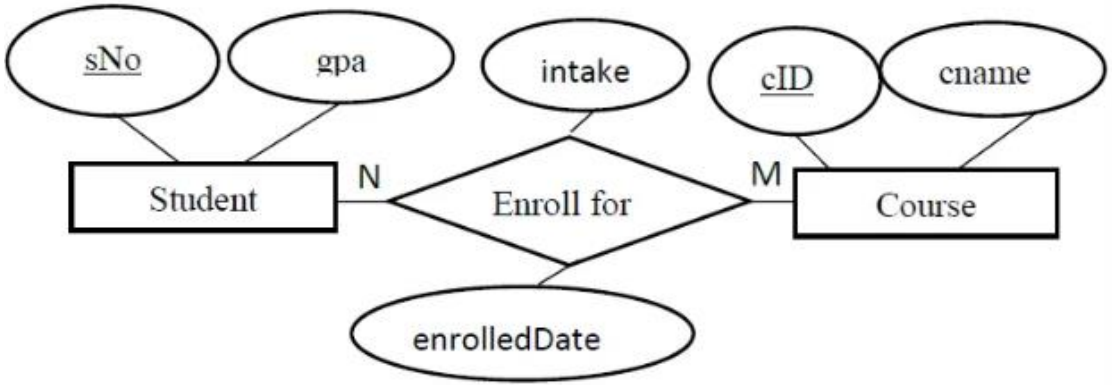


A (a1 , a2)

B (b1 , b2)

R (a1 , b1 , r1)

Examples



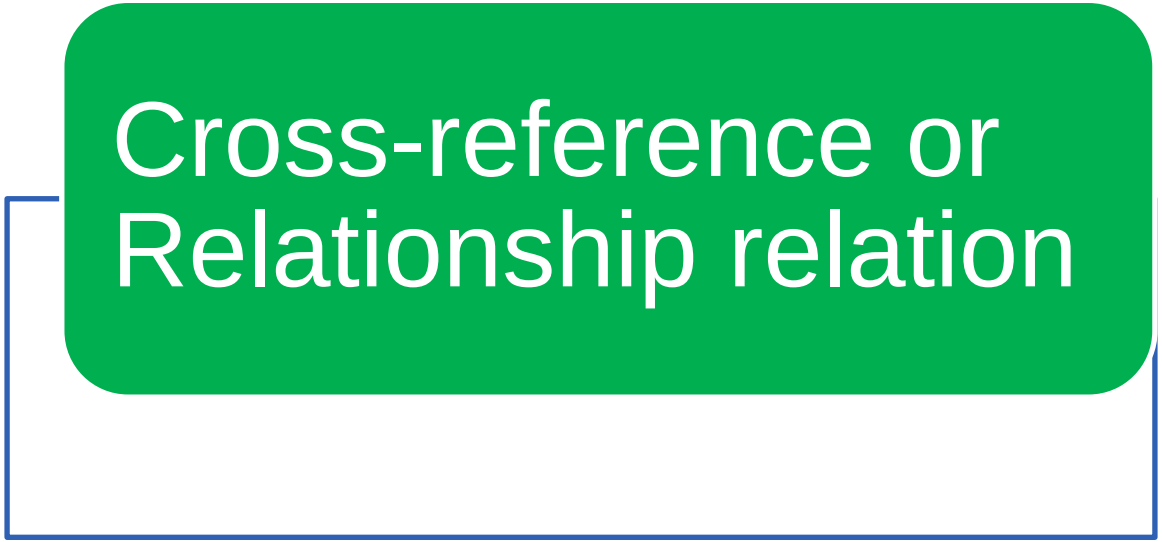
Student (sNo, gpa)

EnrollFor (sNo,cID, intake, enrolledDate)

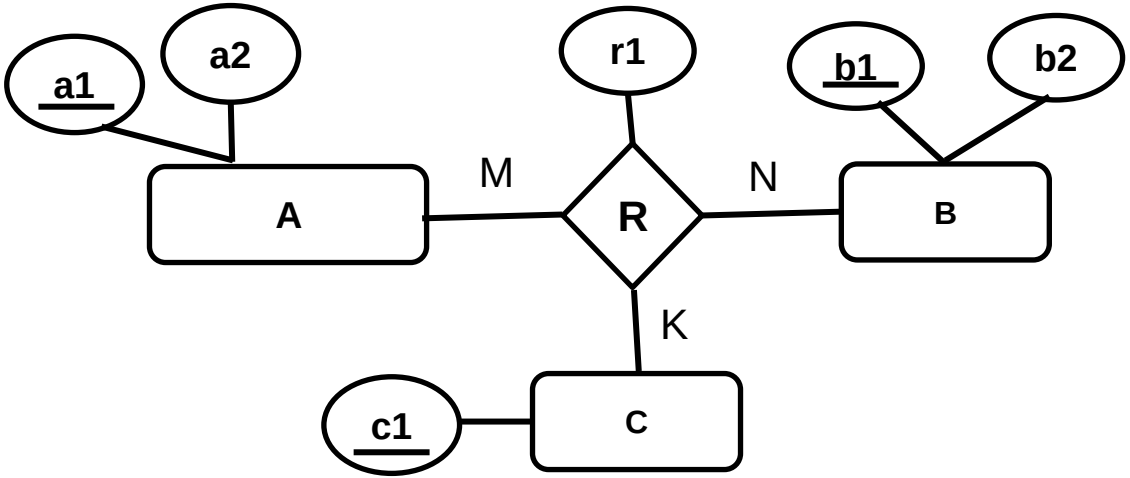
Course (cID, cname)

4) Ternary Relationship with Cardinality M:N:K

One Possible Approaches



4-1) Single Attribute for Relationship



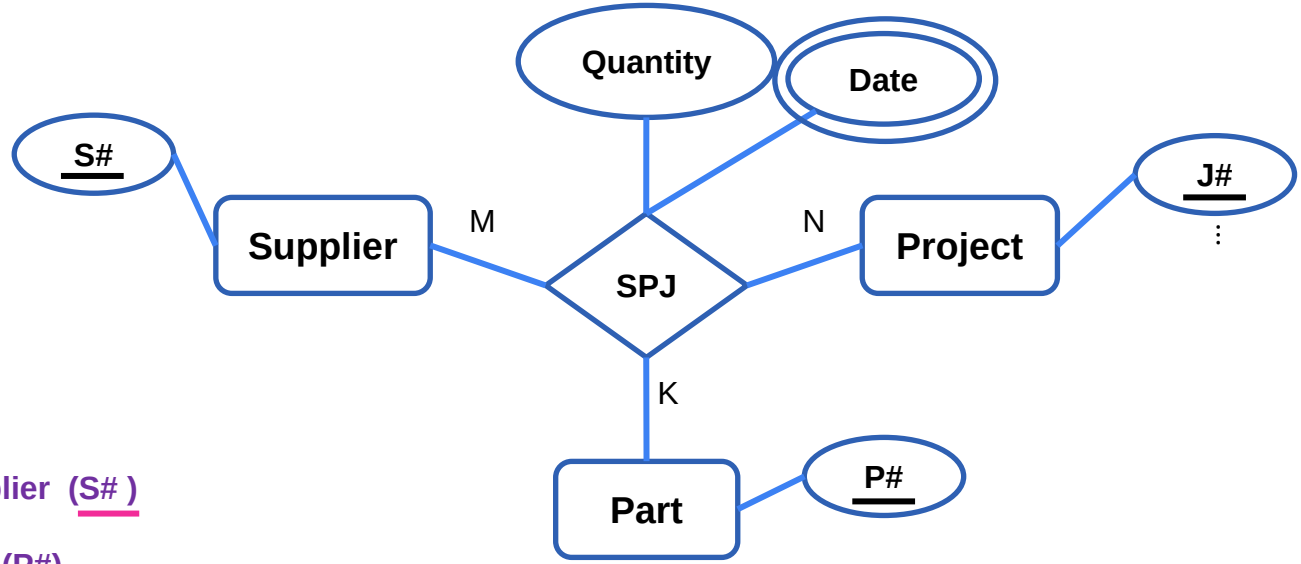
A (a1 , a2)

B (b1 , b2)

C (c1)

R (a1 , b1 , c1 , r1)

Example: Multivalued Attribute for Relationship



Supplier (S#)

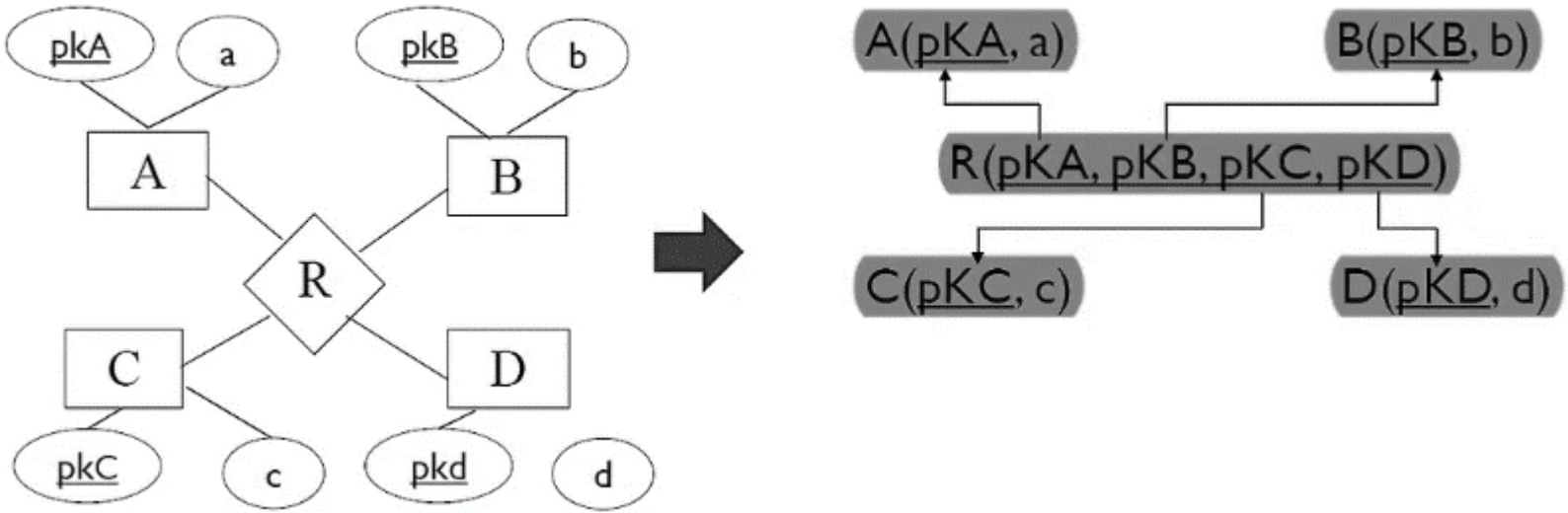
Part (P#)

Project (J#)

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

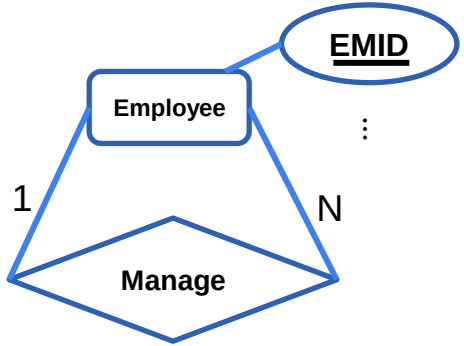
5) N-ARY Relationship

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



6) Unitary Relationship with Cardinality 1:N

1:N Unitary Relationship

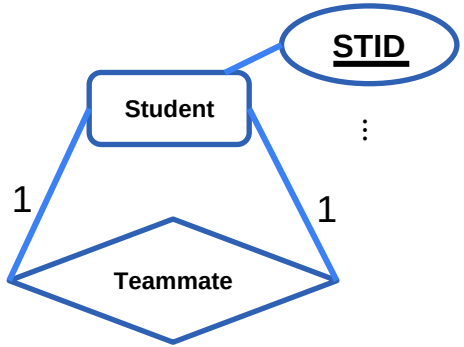


- One table:

EMPL (EMID , EMGRID)

7) 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

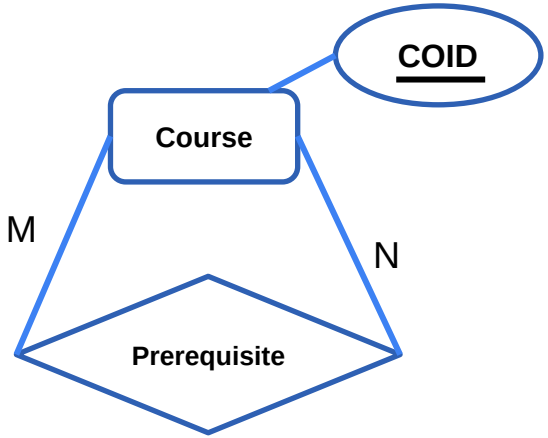
- Solution for when there are many people without group members.

EMPL (EMID)

TEAMMATE (EMID , EMGRID) — Unique

8) Unitary Relationship with Cardinality M:N

M:N Unitary Relationship

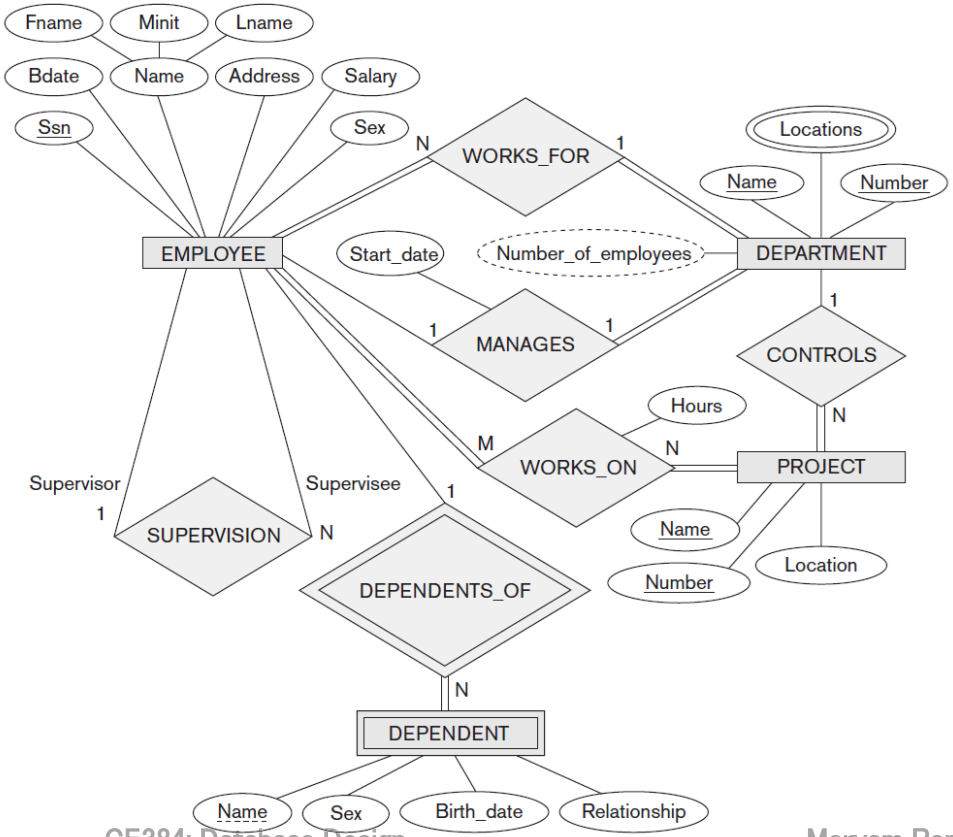


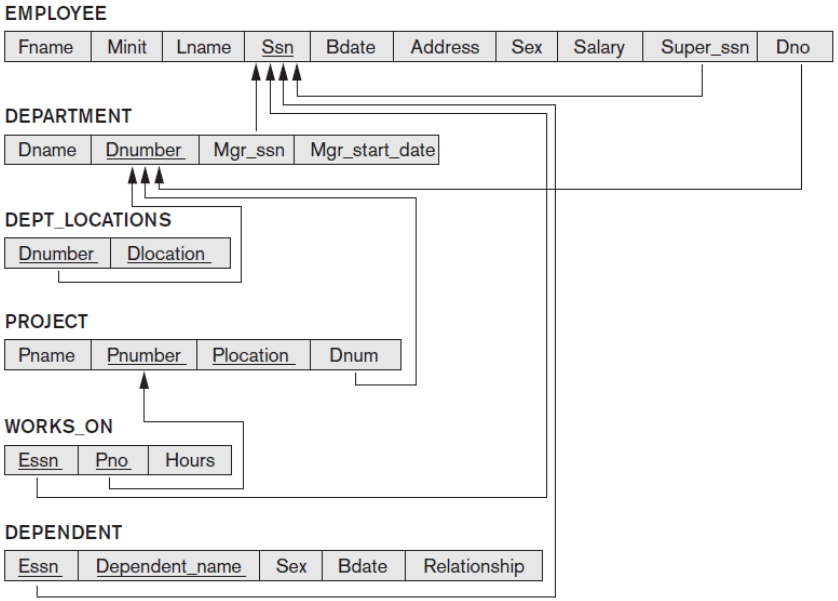
- Two tables:

Prerequisite (COID , PRECOID)

Course (COID)

Example


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
N
Hours
M
WORKS_ON
N
PROJECT
Supervisor
1
Supervisee
1
SUPERVISION
N
DEPENDENTS_OF
Name
Number
Location
N
DEPENDENT
Name
Sex
Birth_date
Relationship


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

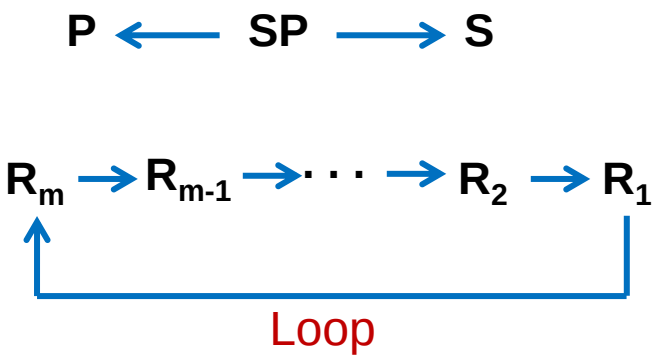
Referential Integrity

Referential Integrity Rules

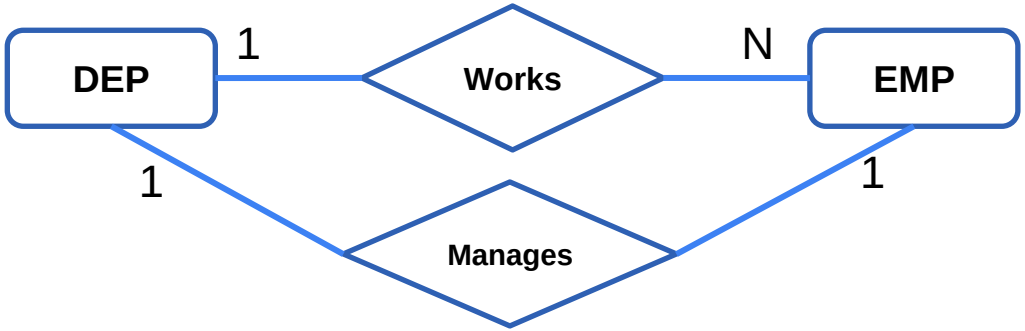
- The main rules of referential integrity are to ensure that:
  - A foreign key value in one table corresponds to an existing primary key value in another table.
  - When a primary key value is deleted, all foreign key values that reference it are also deleted or set to null.
  - When a primary key value 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



Employee ID of the manager

DEPT (D#, DTITLE, … , E#)

Unique

Department ID

EMPL (E#, ENAME, …, D#)

DEPT ⇄ **EMPL**

Self-Referencing (Loop-Referencing with one relationship)

Manager ID with renamed name

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

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

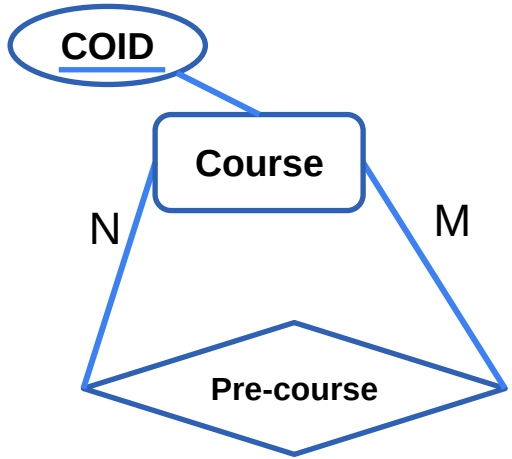
- Draw the ER of this logical model!

NOTE

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

COT (COID, …)

COPRECO(COID, PRECO)



References

- Chapter 9 of FUNDAMENTALS OF Database Systems, SEVENTH EDITION
- Chapter 6 Part 7 of DATABASE SYSTEM CONCEPTS, S I XTH E D I T I ON.
- Chapter 4 of Database Systems A Practical Approach to Design, Implementation, and Management, SIXth edition