Mapping EER Diagrams To Relation Data Model (EER2RDM Mapping)

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

01

ISA Relationships

ISA Relationship to Table

- Multi-relation options 1 & 2
  - Resulting more than 1 Table
- Single-relation options 3 & 4
  - Resulting ONLY 1 Table

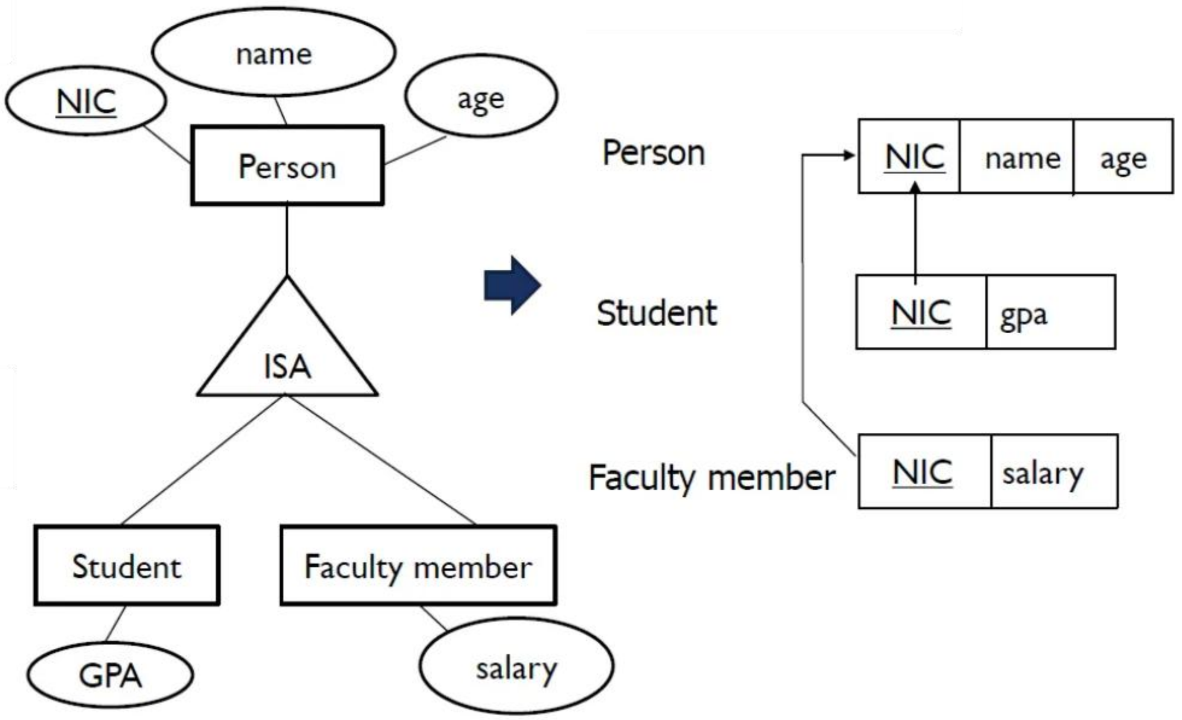
Each option is suitable for specific situations.

Option 1

- Create a relation for the superclass with its attributes.
- Primary key of the superclass becomes the primary key of the relation.
- Create separate relations for the sub classes with their attributes.
- Primary key of the superclass is also primary key of each subclass.
- They are also foreign keys referring to the primary key of the relation created for the super class.
- Option 1 works for all constraints disjoint, overlapping, total and partial.

Option 1

- Example



Option 1

- It works for all constraints disjoint, overlapping, total and partial.
  - Overlapping: There is a Person who is a Student and a Faculty member. All 3 tables can be utilized.
  - Disjoint: There is a Person who is a Student but not a Faculty member and vice versa. 2 tables can be utilized.
  - Total: Other than Student and Faculty member, there are no other persons. 2 tables can be utilized.
  - Partial: Other than Student and Faculty member, there are other persons as well. 1 table can be utilized.

Disadvantage of Option 1

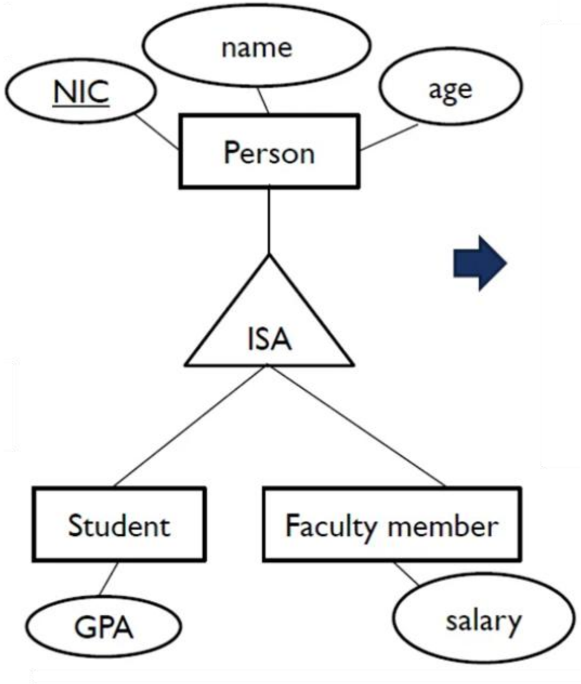
- Let's say there are 10,000 Students and you want to get all their information (NIC, name, age, GPA)
- So what do you have to do to get these information when you are querying?
- You have to join Person and Student Table, so for 10,000 students joining tables might take a little time.
- Basically, you have lot of tables, that is a disadvantage of option 1.

Option 2

- Create separate relations for all the subclasses with their own attributes and the attributes of the superclass.
- Primary key of the superclass becomes primary key of the subclasses.
- The ISA relationship must be total (i.e. subclasses must cover the super class)

Option 2

- Example



Student

<u>NIC</u>	name	age	gpa

Faculty member

<u>NIC</u>	name	age	salary

Option 2

- It works for all constraints disjoint, overlapping, total.
  - Overlapping: There is a Person who is a Student and a Faculty member. All 2 tables can be utilized (Only thing is some redundant data might be created)
  - Disjoint: There is a Person who is a Student but not a Faculty member and vice versa. 1 table can be utilized.
  - Total: Other than Student and Faculty member, there are no other persons. All 2 tables can be utilized.
  - Partial: Other than Student and Faculty member, there are other persons as well. Not supported because there is no person table here.

Disadvantage of Option 2

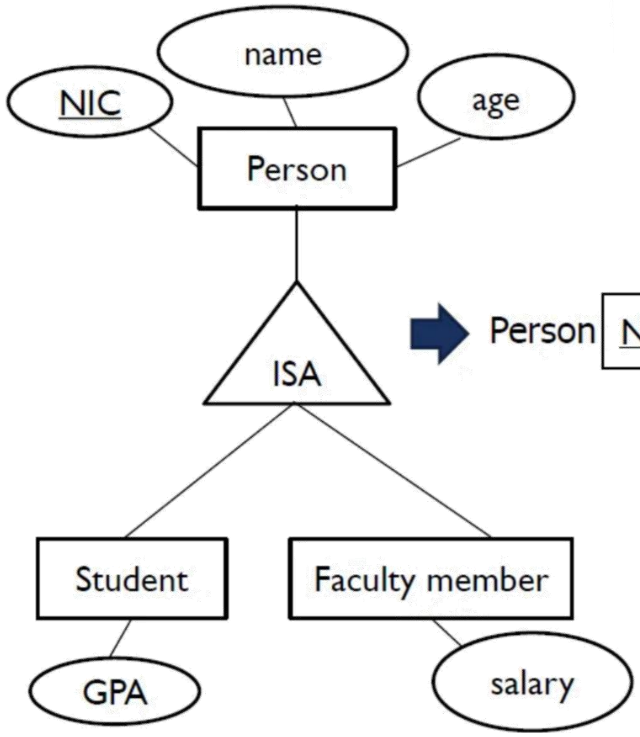
- Partial is not supported. We can not store information of other persons than Student and Faculty member.
- When the person is both student and faculty member. data duplication will happen.

Option 3

- Create a single relation including attributes of the superclass as well as attributes of all sub classes.
- Include an attribute named type for specifying which subclass the entity belongs if any.
- Primary key of the superclass becomes primary key of the relation.
- The specialization/generalization relationship must be disjoint.
- Good if subclasses have few attributes.

Option 3

- Example



Person

NIC	name	age	gpa	salary	type

"type" can store only 3 Values (Including NULL), Values are:
Student , Faculty member or NULL

Option 3

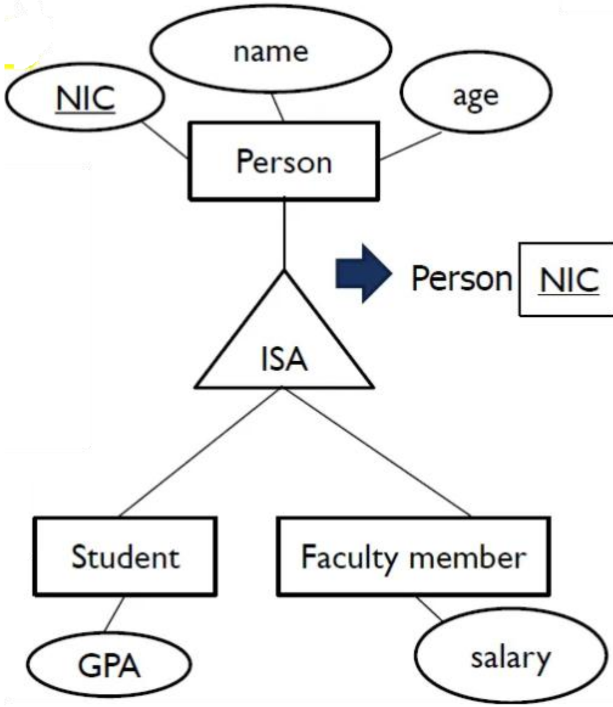
- It works for all constraints disjoint, total, partial.
  - Overlapping: There is a Person who is a Student and a Faculty member. Not supported because in type you can choose either Student OR Faculty member OR NULL. There is no option called BOTH.
  - Disjoint: There is a Person who is a Student but not a Faculty member and vice versa. Type = Student OR Faculty Member.
  - Total: Other than Student and Faculty member, there are no other persons. Supported.
  - Partial: Other than Student and Faculty member, there are other persons as well. Supported.

Option 4

- Create a single relation including attributes of the superclass as well as attributes of all sub classes.
- Include a Boolean attribute to indicate which subclass each tuple belongs to.
- Primary key of the superclass becomes primary key of the relation.
- This relation allows overlapping constraints for specialization /generalization relationship.

Option 4

- Example



Person	<u>NIC</u>	name	age	gpa	salary	student	faculty

Boolean attributes

Boolean means, you can store Boolean combinations.

Option 4

- It works for all constraints overlapping, disjoint, total, partial.
  - Overlapping: There is a Person who is a Student and a Faculty member. Student | FM : True | True.
  - Disjoint: There is a Person who is a Student but not a Faculty member and vice versa. Student | FM : True | False or False | True.
  - Total: Other than Student and Faculty member, there are no other persons. Student | FM : True | True or False | True or True|False
  - Partial: Other than Student and Faculty member, there are other persons as well. Student | FM : False|False

Notes on Options 3 & 4

- Let’s say Student has 8 attributes and Faculty member has 10 attributes.
- Let’s say you have 5000 students. When storing information for all that 5000 students the Faculty member related attribute will remain NULL and Vice versa.
    - Therefore, it is NOT RECOMMENDED to use OPTION 3 and OPTION 4 when you have lot of attributes in the sub classes. Why? because you will end up with a lot of NULL Values. (Storing empty values is not recommended, it wasting space).
    - It is also NOT RECOMMENDED to use OPTION 3 and OPTION 4 when you have RELATIONSHIPS in the SUB-CLASSES. Why? because then all these data are going to be mapped into the Person Table.
- Let’s say faculty member has the relationship. Now since there is ONLY ONE TABLE (That is the Person table) and you are having one FK, so it implies that it is applicable to Person, where in-fact it is not applicable to the Person because it’s (relationship) with Faculty member ONLY.

Conclusion

	Overlapping	Disjoint	Total	Partial
Option1	✅	✅	✅	✅
Option2	✅	✅	✅	
Option3		✅	✅	✅
Option4	✅	✅	✅	✅

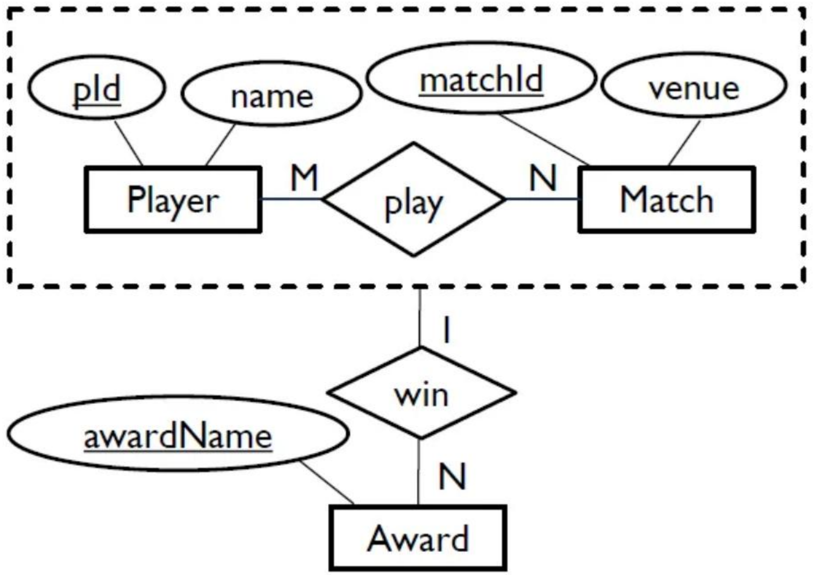
02

Aggregation Relationships

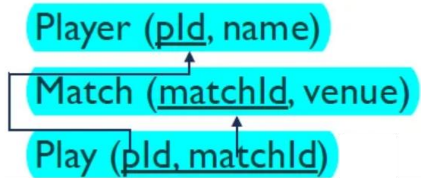
Mapping to Logical Table

- Aggregation mapping could be performed in two steps.
  - Step 1: First map the aggregation relationship R.
  - Step 2:To map relationship set involving aggregation of R, treat the aggregation like an entity set whose primary key is the primary key of the table for R.
- There are actually many combinations for Aggregation Mapping.
- That is mainly due to the different Cardinality Ratios.
- Get the idea of the Process, then you will be able to tackle any combination.

Example

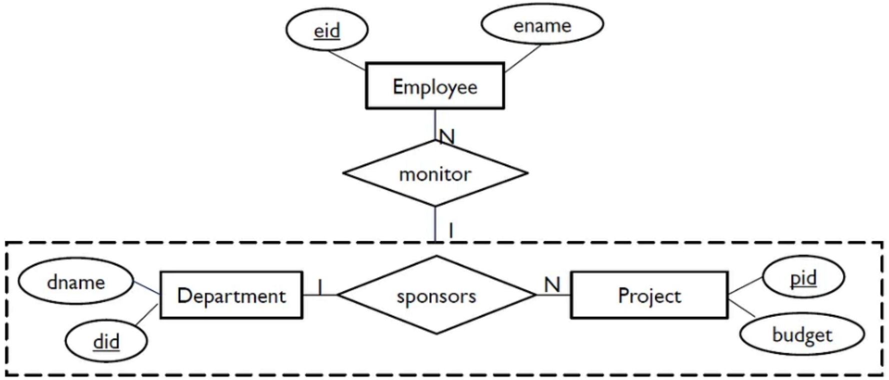


Step 1: First map the aggregation relationship R.
→Firstly map the diagram Inside the BOX.



Step 2 :To map relationship set involving aggregation of R, treat the aggregation like an entity set whose primary key is the primary key of the table for R.
Table for aggregation is ‘Play’. Thus, the relationship win should be mapped considering 1:N relationship between play (primary key : pId, matchId) and award,
Award (awardName, pId, matchId) →Since 1:N, PK of 1 side come to N side.

Example



Step 01 : Map inside the Box

Department(did, dname)

Project(pid, budget, did)

Step 02 : What table did you get OR edit (that means added a FK OR did some kind of change) when you mapped Sponsors relationship ?

Project Table. We added ‘did’ as a FK.(we edited Project schema and added did)

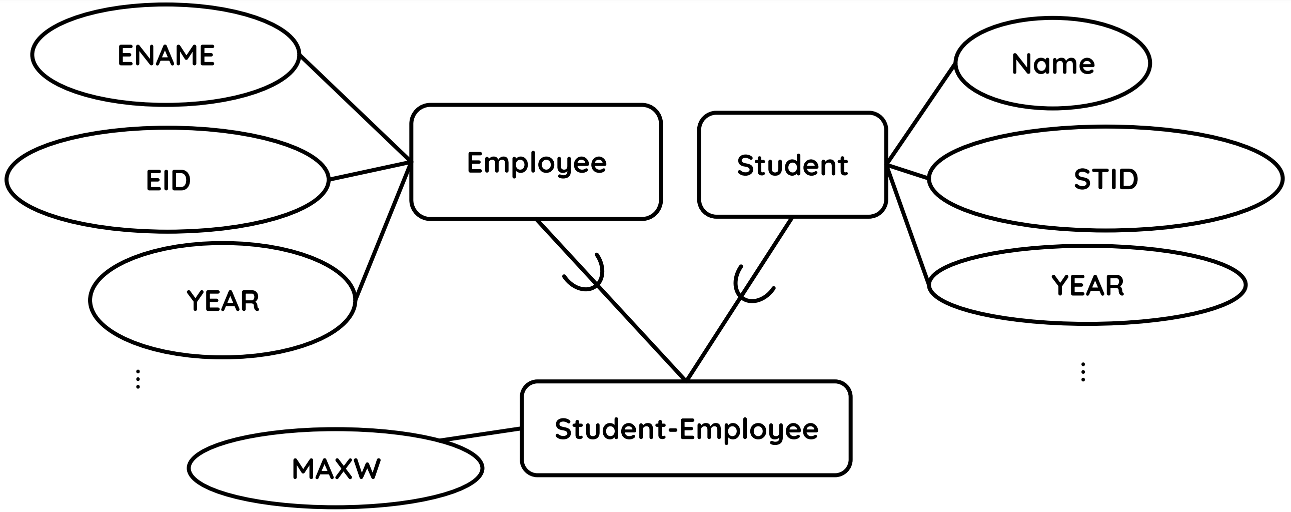
Finally : 1:N, 1 side you have Project table, N side you have Employee.

Employee(eid, ename, pid).

03

Multiple Inheritance (Shared Subclasses) Relationship

Example



STUD (STID, STNAME, …)

EMPL (EID, ENAME, …)

STEM (STID, EID, MAXW)

04

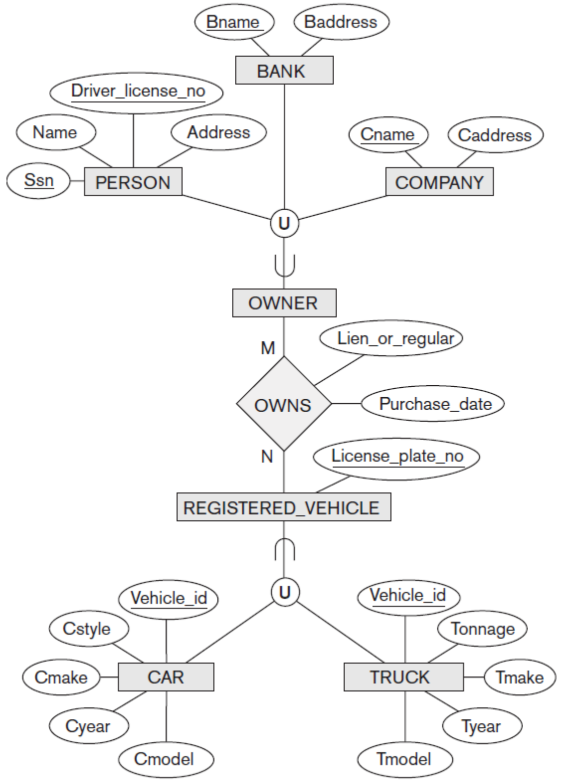
U-Type Relationship

Mapping U-Type

- If the supertype identifiers are from different domains, the subtype display table FKs the supertype display tables, outside the key.

- If the identifier of the supertypes is from the same domain (and it is the same identifier in all instances of the supertypes), the key of the subtype table is the same as the key of the display tables of the supertypes.

Example



Because the scope of the keys of the supertypes is not the same, we put a dummy key ourselves.

PERS (PID, …., OID)
p.k.

COMP (CID, …., OID)
p.k.

BANK (BID, …., OID)
p.k.

OWNER (OID,….)
p.k.

VEHIC (VID,….)
p.k.

OWNS (OID, VID, F, T, ….)
p.k.

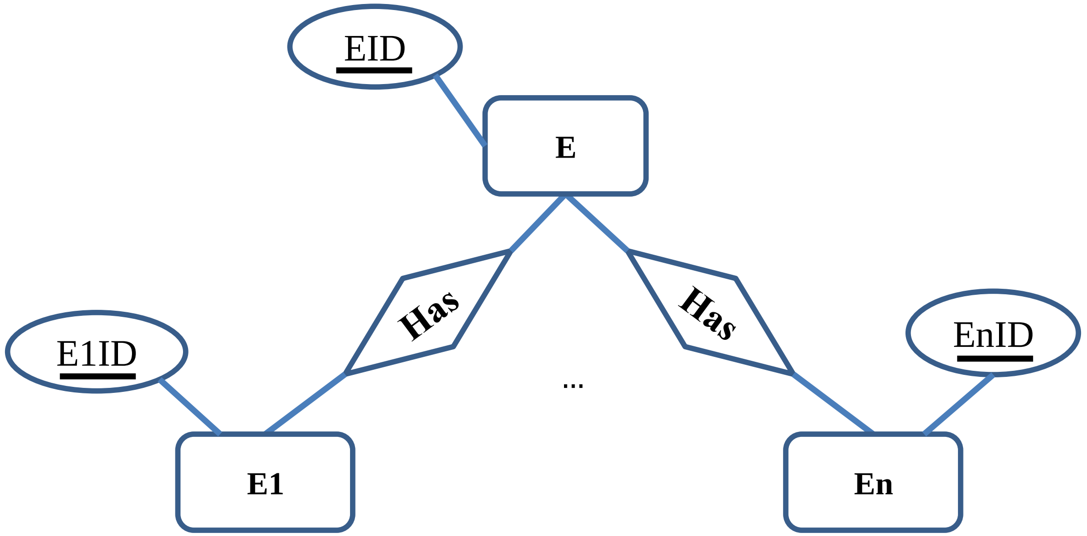
SAVARY (VID, N, ….)
p.k.

BARY (VID, T, ….)
p.k.

05

IS-A-PART-OF Relationship

Example



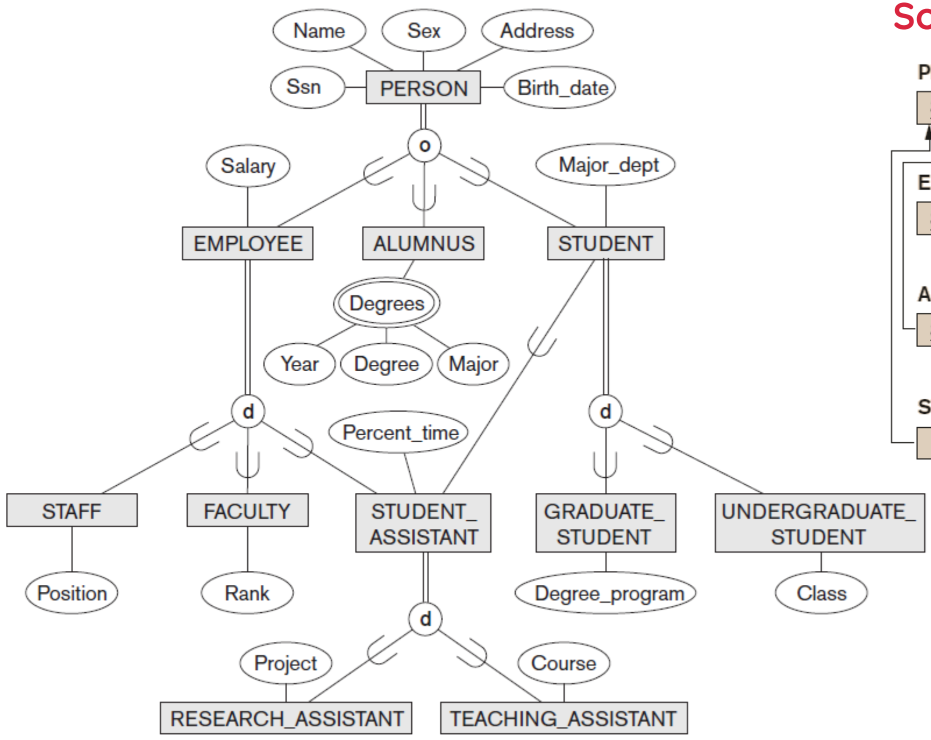
E (EID, ….)
p.k.

E1 (E1ID, EID, ….)
p.k.

….

En (EnID, EID, ….)
p.k.

Example



Solution

PERSON

<u>Ssn</u>	Name	Birth_date	Sex	Address

EMPLOYEE

<u>Ssn</u>	Salary	Employee_type	Position	Rank	Percent_time	Ra_flag	Ta_flag	Project	Course

ALUMNUS

<u>Ssn</u>

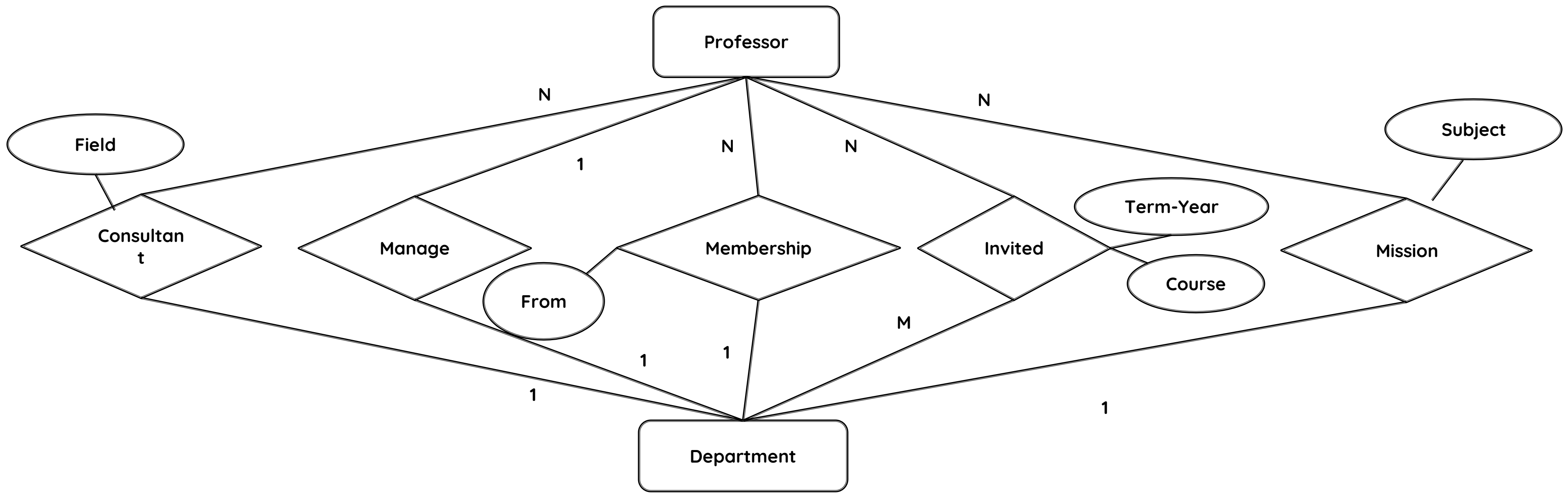
ALUMNUS_DEGREES

<u>Ssn</u>	<u>Year</u>	<u>Degree</u>	<u>Major</u>

STUDENT

<u>Ssn</u>	Major_dept	Grad_flag	Undergrad_flag	Degree_program	Class	Student_assist_flag

Example


Professor
N
N
N
N
1
Field
Subject
Consultan t
Manage
Membership
Invited
Term-Year
Mission
From
Course
M
1
1
1
1
Department

Example Solution

DEPT (DEID, ..., DPHONE, PRID)

p.k. p.k.

PROF (PRID, ..., PRRANK, MDEID, FIELD, MEMDEID, FROM, CDEID, INT)

p.k.

3 Foreign keys from same domain

INVITED (DEID, PRID, YR, TR)

p.k.

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