DATABASE MANAGEMENT SYSTEMS (IT 2040)

LECTURE 02- ER AND EER TO RELATIONAL MODEL MAPPING

LECTURE CONTENT

- Logical database design
- Relational model and its components
- ER to relational mapping

LEARNING OUTCOMES

- Explain the process of logical database design
- Explain relational model and its components
- Convert a complex ER model to the relational model

LOGICAL DATABASE DESIGN

- Once we finished the step of conceptual database design we next select a DBMS to implement our database design.
- We then convert the conceptual database design into a database schema in the data model of the chosen DBMS.
- Before 1970 most database systems were based on two older data models namely, hierarchical model and network model.
- Leading DBMS products nowadays are based on the relational model which introduced by Codd in 1970.

THE RELATIONAL MODEL

- The major advantage of relational model is its simplicity in data representation.
- A relational database is a collection of relations with distinct relation names.
- The main construct representing data in the relational model is the relation.
- A relation consists of a relational schema and a relational instance.

THE RELATIONAL MODEL (CONTD.)

- The relational schema describes the columns for a relation.
  - Ex: Students(*sid:* string, *name:* string, *login:* string, *age:* integer, *gpa:* real)
  - The schema species the relation's name, the name of each **field** and the **domain** of each field.
- An **instance** of a relation is a set of **tuples**, also called **records**, in which each tuple has the same number of fields as the relation schema.
  - A relation instance can be thought of as a *table* in which each tuple is a *row*, and all rows have the same number of fields.

sid	*name*	*login*	*age*	*gpa*
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8

INTEGRITY CONSTRAINTS

- An integrity constraints (IC) is a condition specified on a database schema and restricts the data that could be stored in an instance of the database.
- If a database instance satisfies all the integrity constraints specified on the database schema, it is a **legal** instance.
- There are several types of integrity constraints
  - Domain constraints
  - Referential integrity constraints
  - Key constraints
  - Other constraints

INTEGRITY CONSTRAINTS (CONTD.)

- Domain Constraint
  - Domain constraint specifies that the values that appear in a column must be drawn from the domain associated with that column.
  - Relational database provides data types to specify valid domains.
- Key Constraint
  - The minimal set of attributes that uniquely identify a tuple is called the **key** of a relation
  - A set of fields that uniquely identifies a tuple according to a key constraint is called a candidate key for the relation; we often abbreviate this to just key.
  - One of the candidate keys is designated as the primary key.

INTEGRITY CONSTRAINTS (CONTD.)

- Referential integrity Constraint
  - The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples in the two relations.
  - Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation.
  - Foreign keys enforce referential integrity constraints
  - Foreign key attributes in R1 referring to R2 have the following rules:
    - The FK attributes in R_1 have the *same domain(s)* as the primary key attributes if R_2
    - The value of FK in tuple t_1 in R_1 must reference an *existing* PK value in tuple t_2 of R_2
  - We can diagrammatically display the foreign keys by drawing an arrow from the foreign key to the primary key

INTEGRITY CONSTRAINTS (CONTD.)

- Other constraints
  - **Table Constraints** (constraints within tables)
    - Example : Balance of the account should be greater than 0
    - In SQL, CHECK constraint can be used
  - **Assertions** (constraints between multiple tables)
    - For example: consider the following schema

      Emp(eid, ename, salary, did)

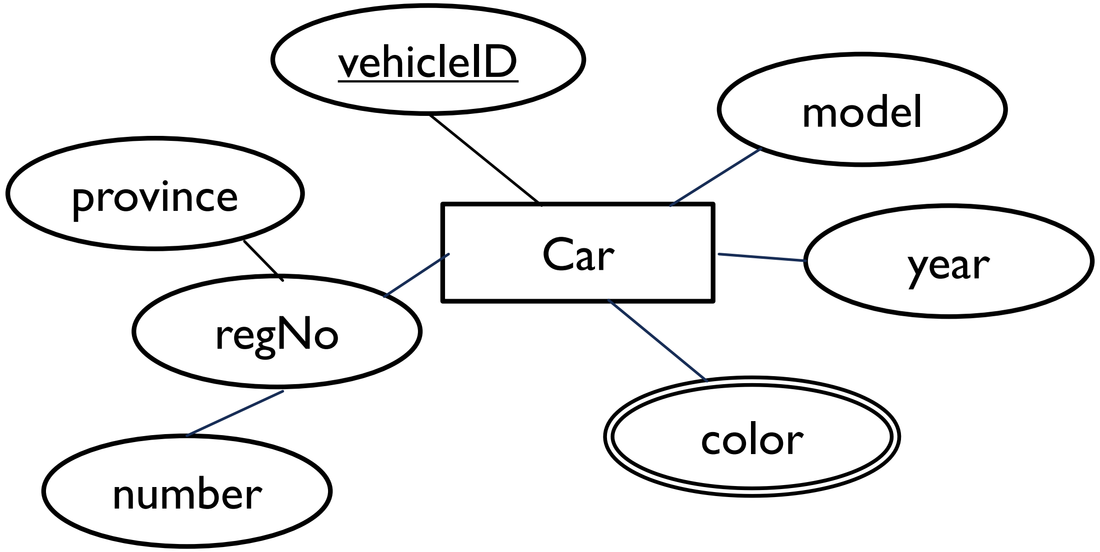
      Department (did, dname, mgr)
    - Suppose that an employee's salary should not exceed his/her manager's salary
    - Such constraints can be using specific language features such as triggers and assertions

ACTIVITY

- In two minutes summarize the integrity constraints discussed in today's lecture on a blank paper.
- Exchange what you have written with your peer and correct the answer.
- Have you understood each constraint well?

MAPPING ENTITIES AND ATTRIBUTES

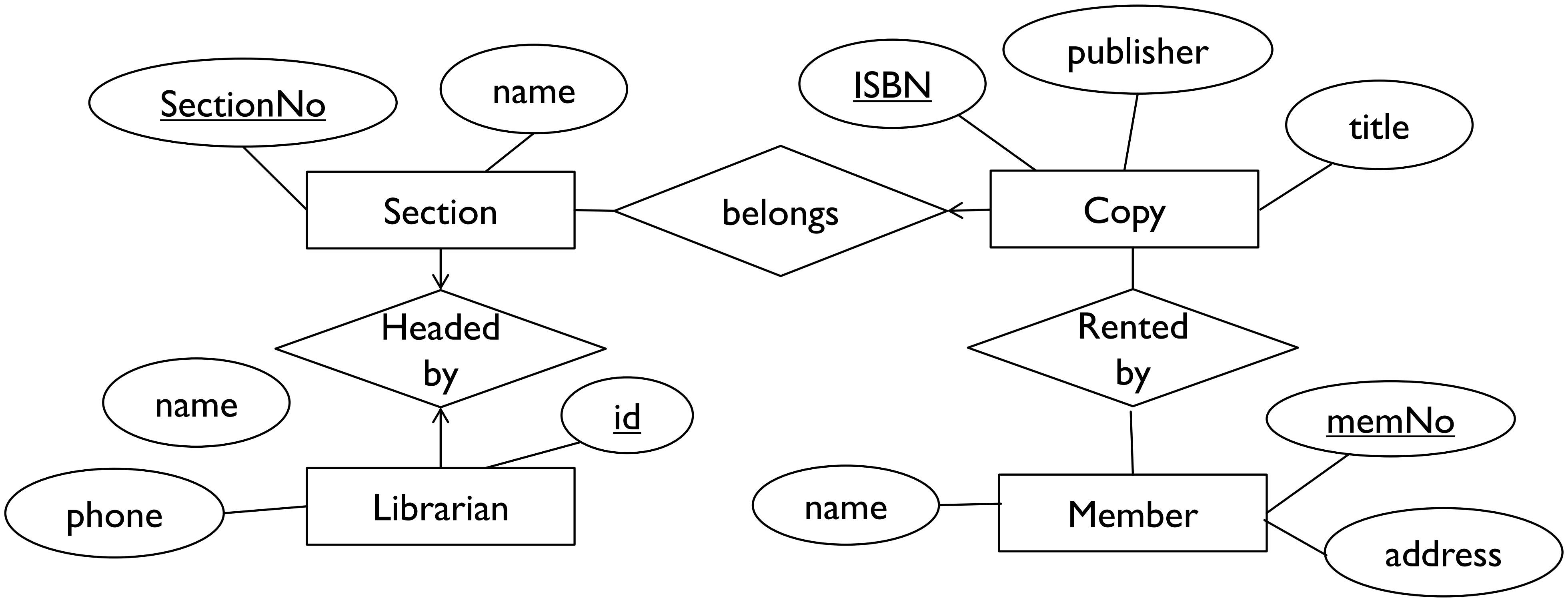
- Go through page 1 of the handout 2.
- Map the entity below to relational model.



- Compare what you have written with your peer.

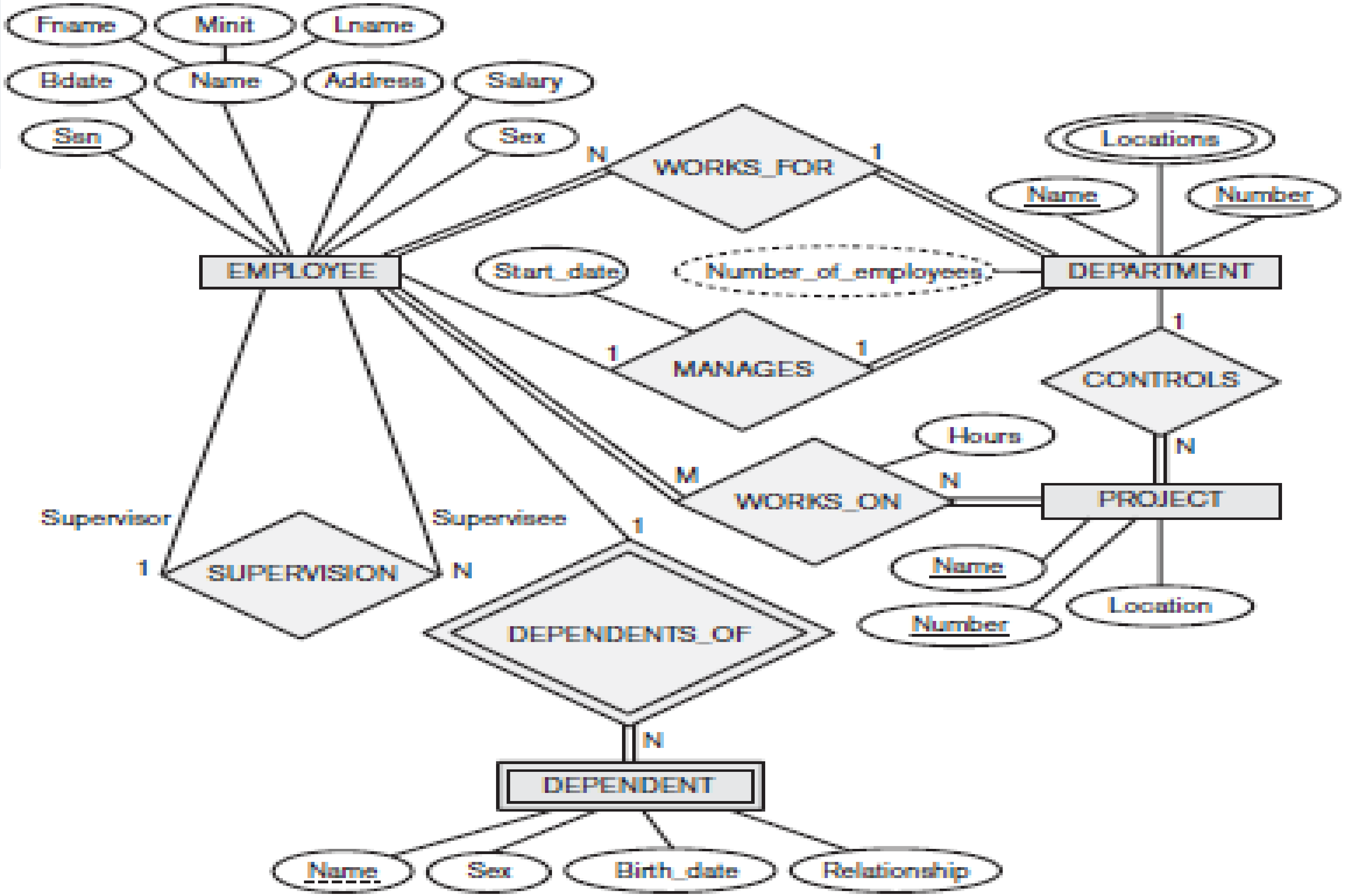
MAPPING BINARY RELATIONSHIPS

- Go through page 1 of the handout 2.
- Now lets map the following ER diagram to relational model



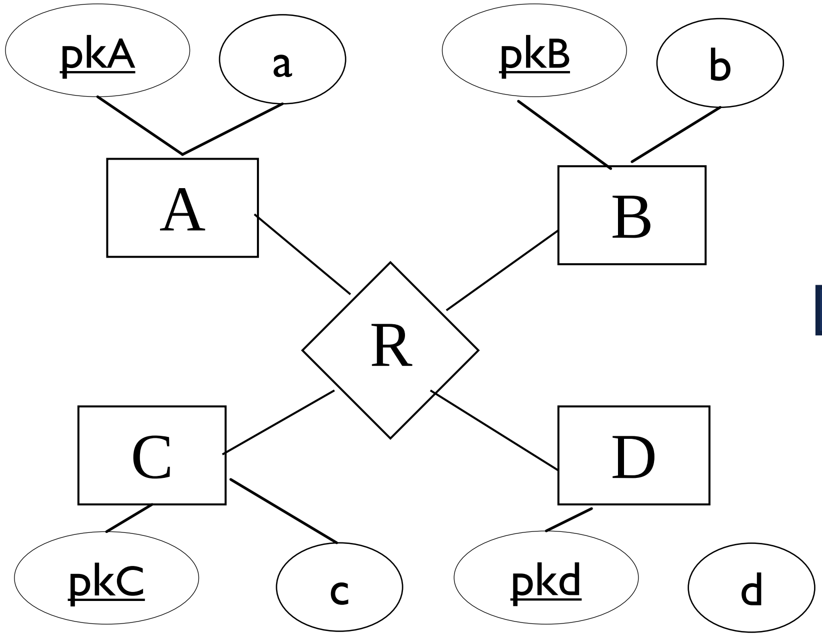
ACTIVITY

- Map the ER diagram to relational model



MAPPING N-ARY RELATIONSHIPS

- N-ary relationship is mapped in to a “Relationship” relation and foreign keys



A(pKA, a)

B(pKB, b)

R(pKA, pKB, pKC, pKD)

C(pKC, c)

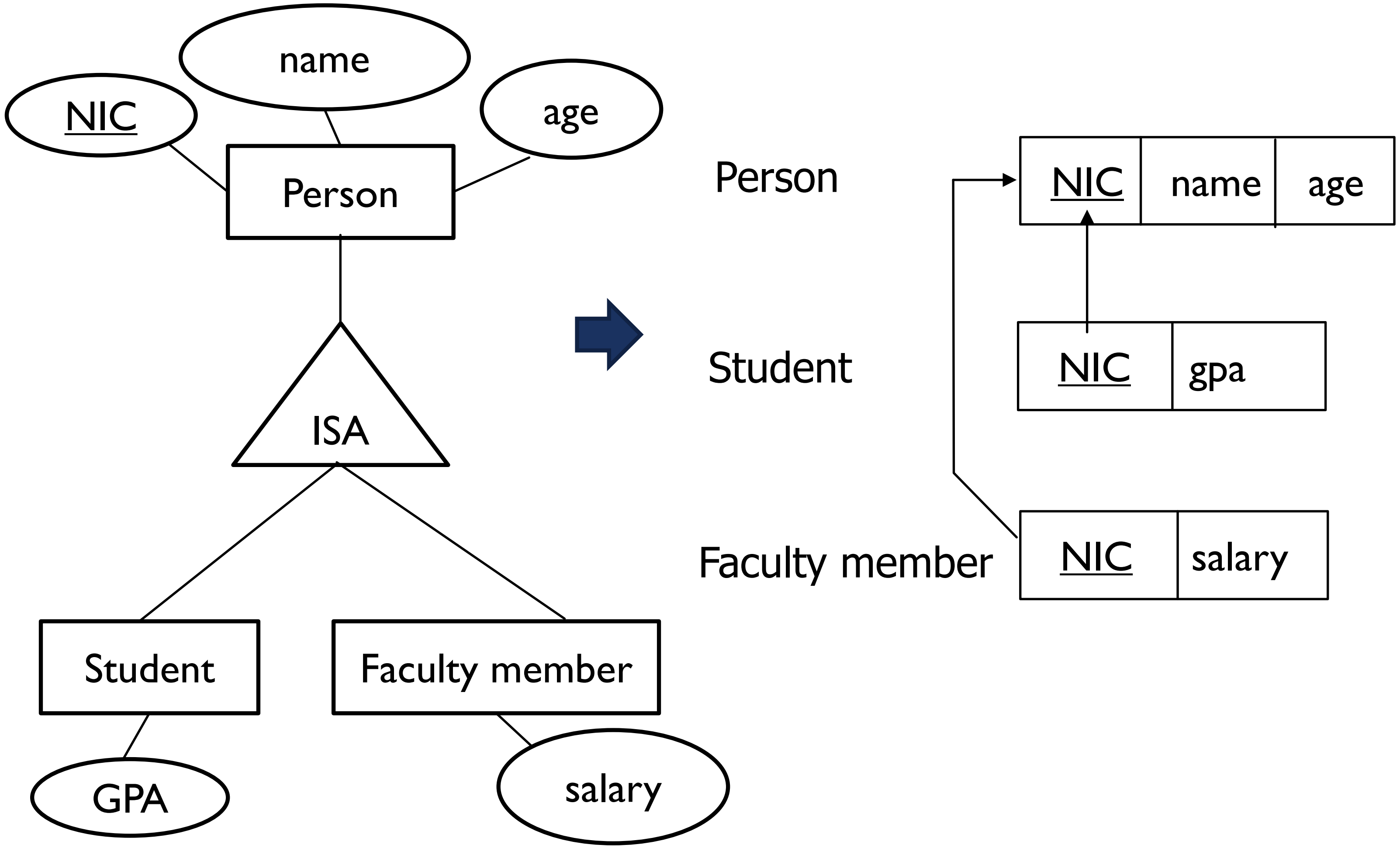
D(pKD, d)

MAPPING ISA-RELATIONSHIPS

- There are four different options for mapping ISA relationships.
  - Multi-relation options : option 1 & option 2
  - Single-relation options : option 3 & option 4
- Each option is suitable for specific situations.

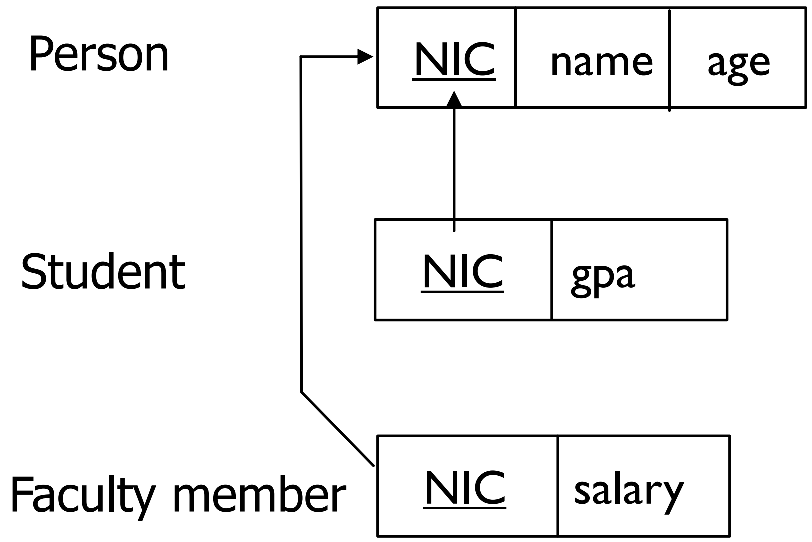
ISA MAPPING – OPTION I

- 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 I works for all constraints disjoint, overlapping, total and partial**



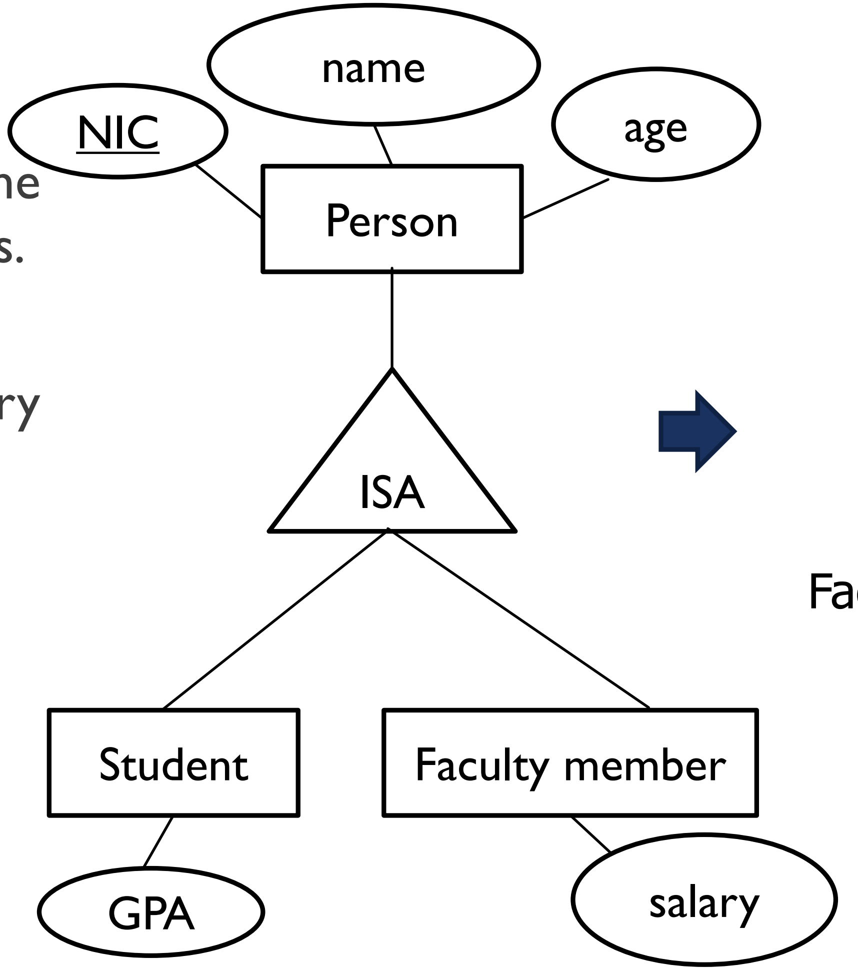
ISA MAPPING – OPTION I (CONTD.)

- Now think how you would store information of following people
  - A person who is not a student or a faculty member(i.e when the ISA relationship is parial)
  - A person who is a student but not a faculty member (i.e disjoint)
  - A person who is both a student and a faculty member (i.e: overlapping classes)



ISA MAPPING – 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)**



Student	<u>NIC</u>	name	age	gpa

Faculty member	<u>NIC</u>	name	age	salary

ISA MAPPING – OPTION 2 (CONTD.)

- Now think how you would store information of following people
  - A person who is not a student or a faculty member(i.e when the ISA relationship is parial)
  - A person who is a student but not a faculty member (i.e disjoint)
  - A person who is both a student and a faculty member (i.e: overlapping classes)

Student

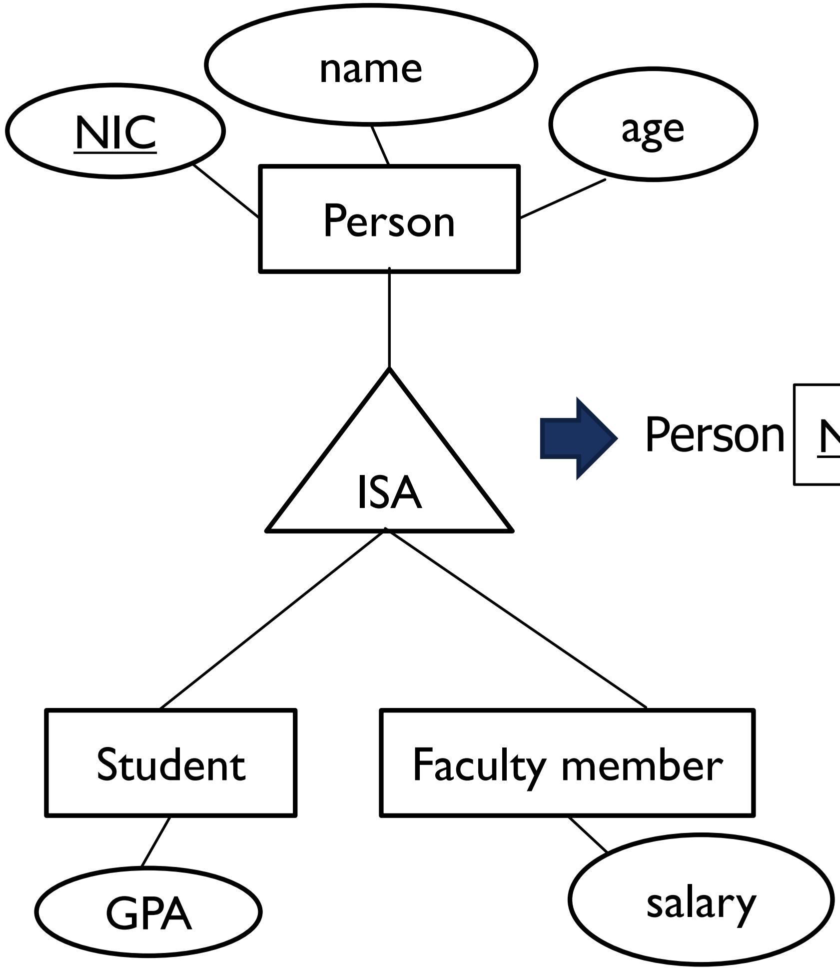
NIC	name	age	gpa

Faculty member

NIC	name	age	salary

ISA MAPPING – 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**



Person	<u>NIC</u>	name	age	gpa	salary	type

ISA MAPPING – OPTION 3 (CONTD.)

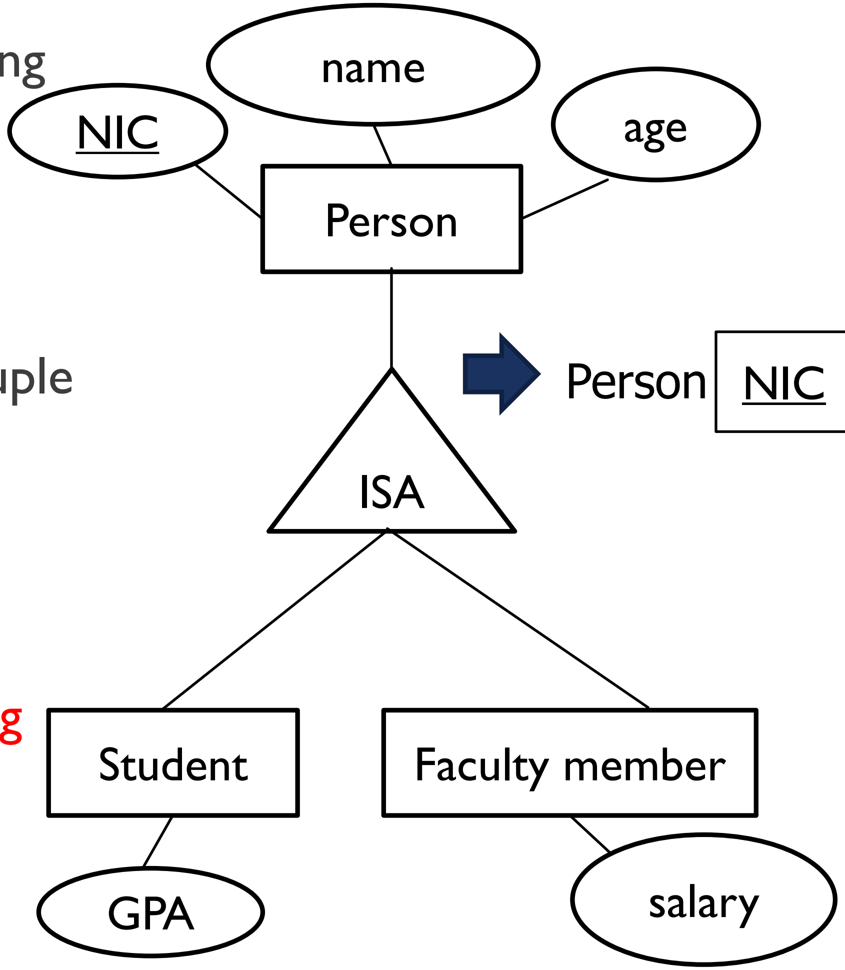
Person

<u>NIC</u>	name	age	gpa	salary	type

- Now think how you would store information of following people
  - A person who is not a student or a faculty member(i.e when the ISA relationship is parial)
  - A person who is a student but not a faculty member (i.e disjoint)
  - A person who is both a student and a faculty member (i.e: overlapping classes)

ISA MAPPING – 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



Person

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

Boolean attributes

ISA MAPPING – OPTION 4 (CONTD.)

Person

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

- Now think how you would store information of following people
  - A person who is not a student or a faculty member(i.e when the ISA relationship is parial)
  - A person who is a student but not a faculty member (i.e disjoint)
  - A person who is both a student and a faculty member (i.e: overlapping classes)

ACTIVITY

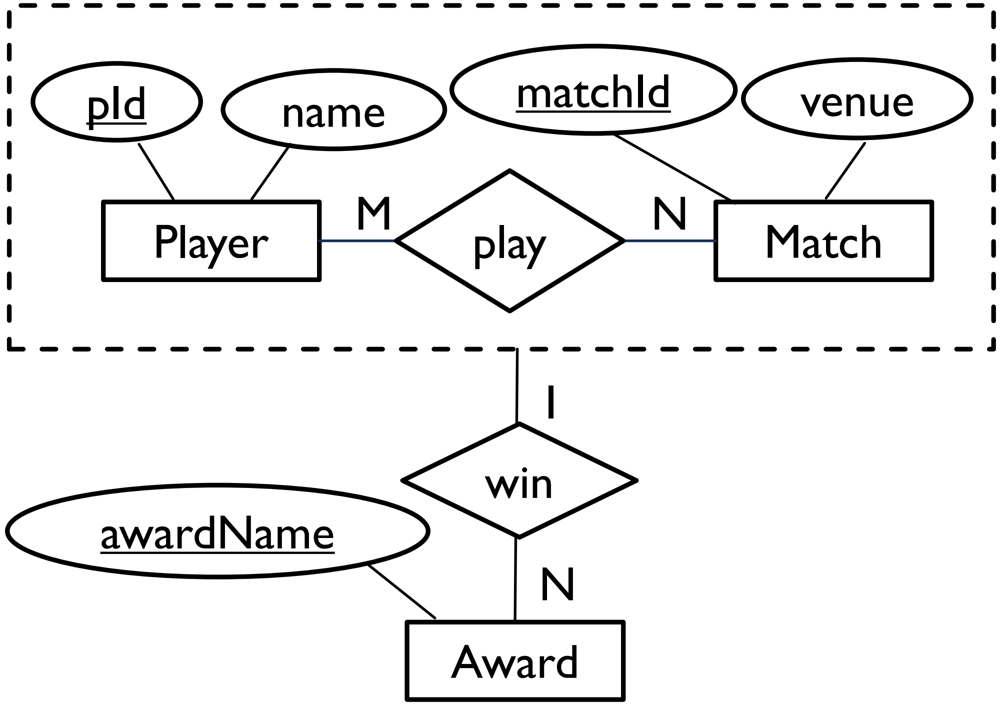
- Fill the table below indicating the which constraints work with which option.

	Overlapping	Disjoint	Total	Partial
Option 1	X	X	X	X
Option 2	X	X	X	
Option 3		X	X	X
Option 4	X	X	X	X

MAPPING AGGREGATION RELATIONSHIPS

- 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

MAPPING AGGREGATION RELATIONSHIPS (CONTD.)



- Step 1: First map the aggregation relationship R.

  Player (<u>pId</u>, name)

  Match (<u>matchId</u>, venue)

  Play (<u>pId, matchId</u>)

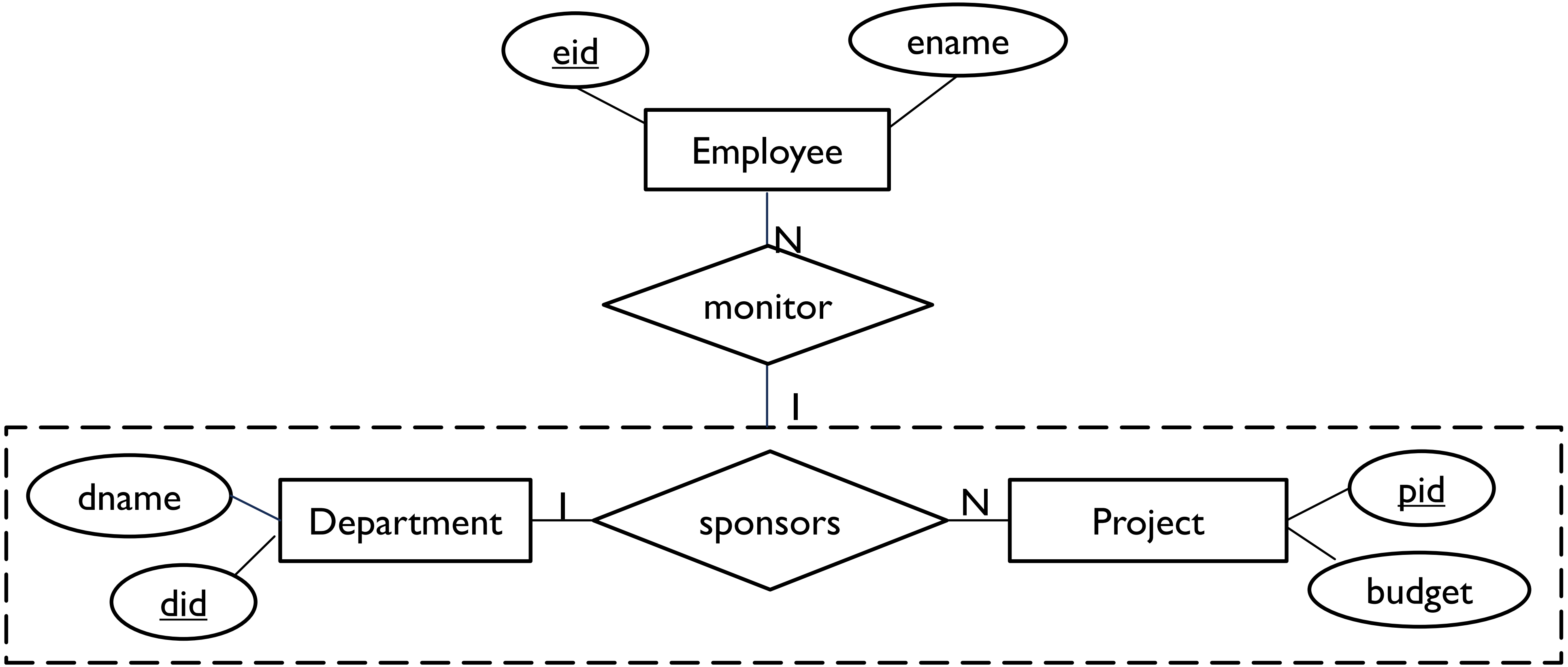
- Step 2 :To map relationship set involving aggregation of R, treat the aggregation like an entity set whose primary key is the primary key of the table for R.

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 (<u>awardName, pId, matchId</u>)

MAPPING AGGREGATION RELATIONSHIPS (CONTD.)

- Map the following EER diagram to relational model.



WHAT YOU HAVE TO DO BY NEXT WEEK

- Try out the self-test questions on the course web.
- Complete the tutorial.
- Complete the lab sheet