Mapping ER-EER to Relational Model

Relational Model Review - Concepts

Relational Model is made up of tables

- A row of table = a relational instance/tuple
- A column of table = an attribute
- A table = a schema/relation
- Cardinality = number of rows
- Degree = number of columns

Review - Example

Attribute

tuple/relational instance

SID	Name	Major	GPA
1234	John	CS	2.8
5678	Mary	EE	3.6

Cardinality = 2

4 Degree

A Schema / Relation

ER to Relational Mapping

How do we convert an ER diagram into a table?? Simple!!

Basic Ideas:

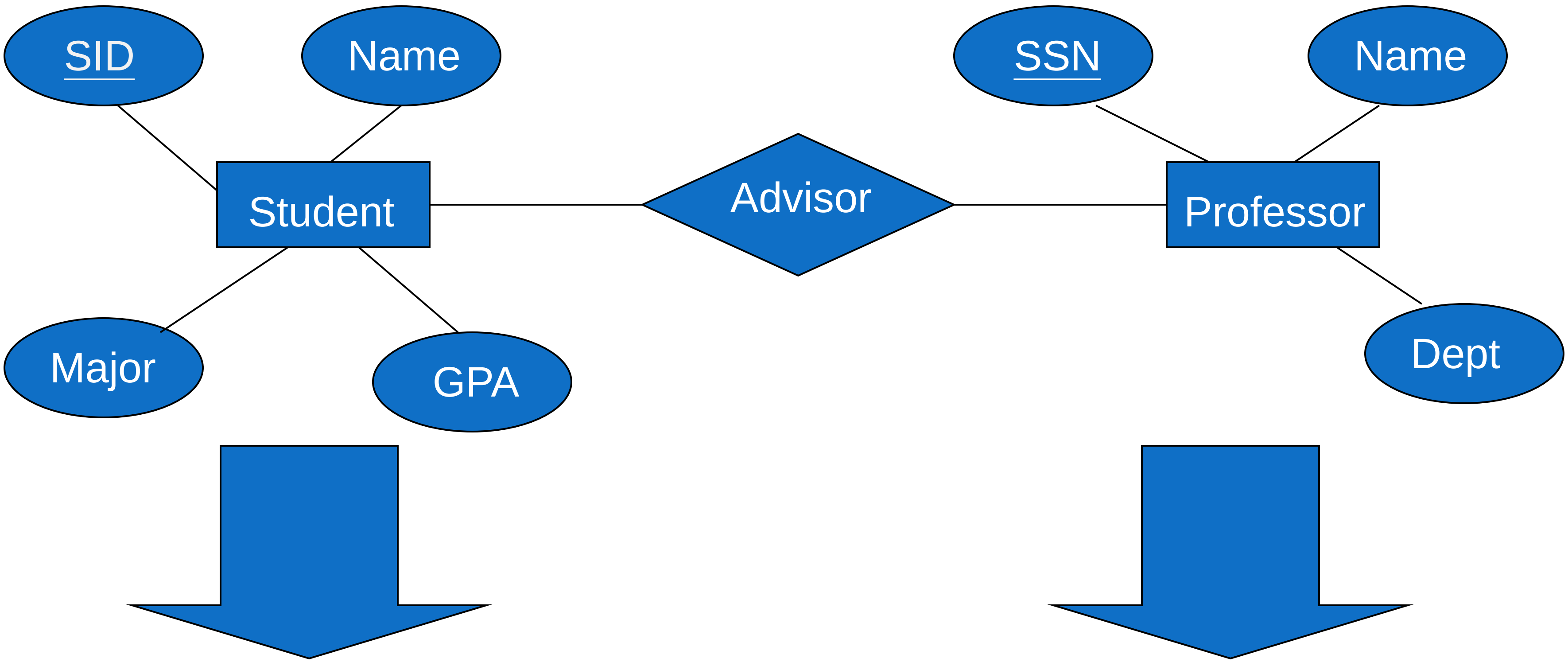
- Build a table for each entity set
- Build a table for each relationship set if necessary (more on this later)
- Make a column in the table for each attribute in the entity set
- Composite and Multivalue Attributes
- Primary Key

ER\EER to Relational Mapping

- **ER-to-Relational Mapping Algorithm**
  - Step 1: Mapping of Regular Entity Types
  - Step 2: Mapping of Weak Entity Types
  - Step 3: Mapping of Binary 1:1 Relationship Types
  - Step 4: Mapping of Binary 1:N Relationship Types
  - Step 5: Mapping of Binary M:N Relationship Types
  - Step 6: Mapping of Multivalued attributes
  - Step 7: Mapping of N-ary Relationship Types

- **Mapping EER Model Constructs to Relations**
  - Step 8: Mapping of Specialization or Generalization
  - Step 9: Mapping of Union Types (Categories)

Mapping – Strong Entity Set



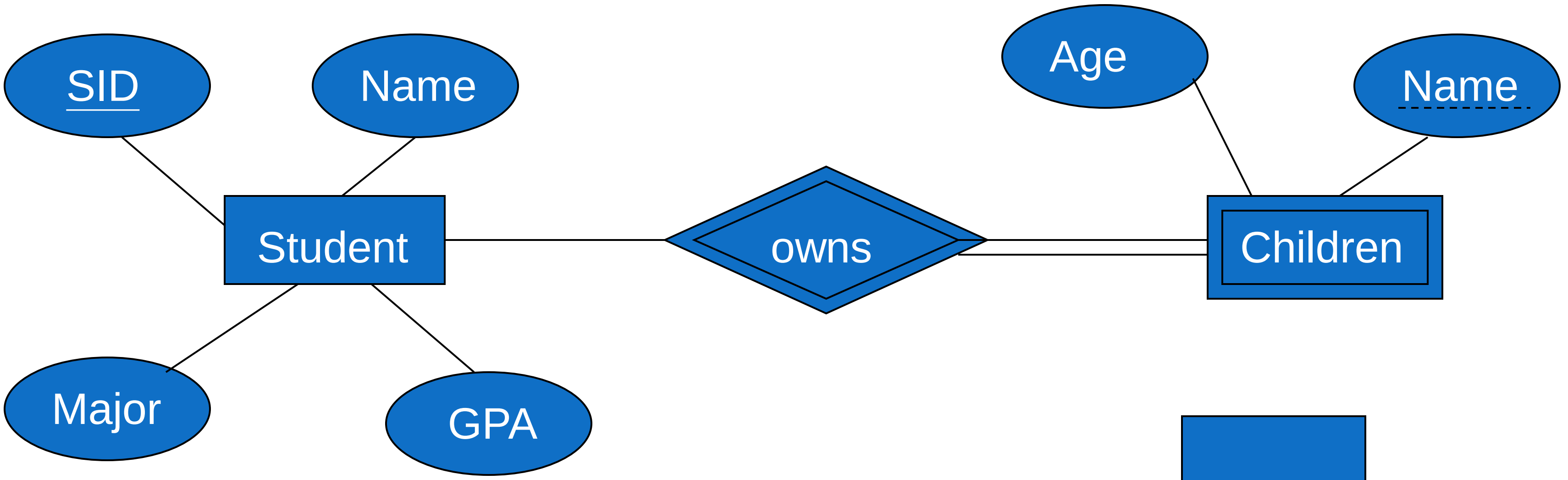
SID	Name	Major	GPA
1234	John	CS	2.8
5678	Mary	EE	3.6

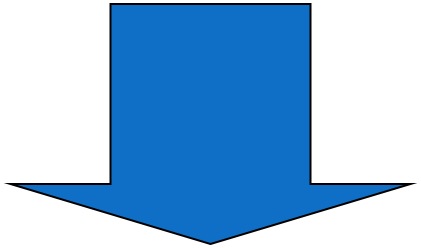
SSN	Name	Dept
9999	Smith	Math
8888	Lee	CS

Mapping of Weak Entity

- Weak Entity Set cannot exists alone
- To build a table/schema for weak entity set
  - Construct a table with one column for each attribute in the weak entity
  - Add a column for the primary key of the Owner of the Weak Entity
  - Primary Key of the weak entity = Discriminator + foreign key

Mapping - Weak Entity Set





Mapping Rule

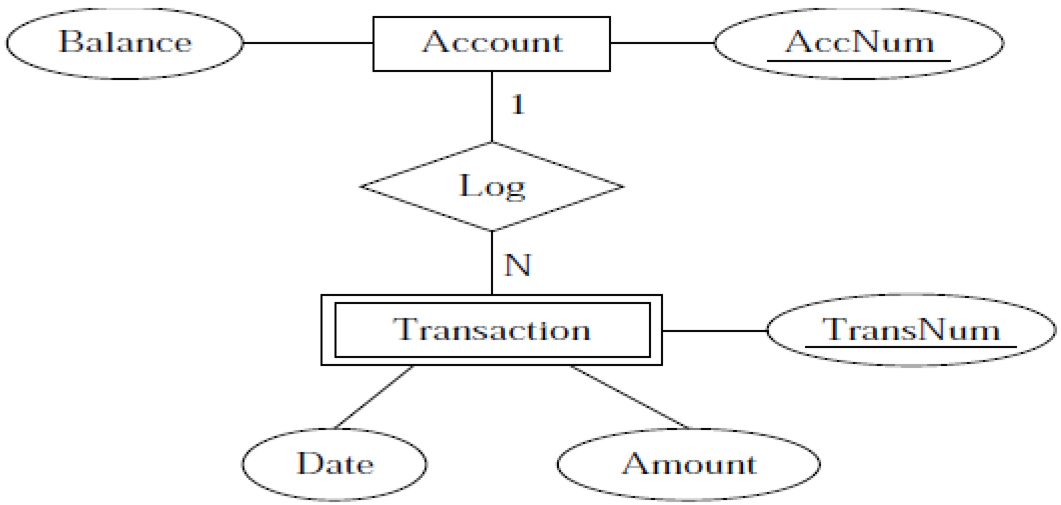
- Construct a table with one column for each attribute in the weak entity
- Add primary key of the Owner Entity in the table

Age	Name	SID
10	Bart	1234
8	Lisa	5678

* Primary key of *Children* is *Parent_SID + Name*

Mapping - Weak Entity Set

Example:



Account

AccNum	Balance

Transaction

TransNum	AccNum	Date	Amount

MAPPING OF RELATIONSHIPS

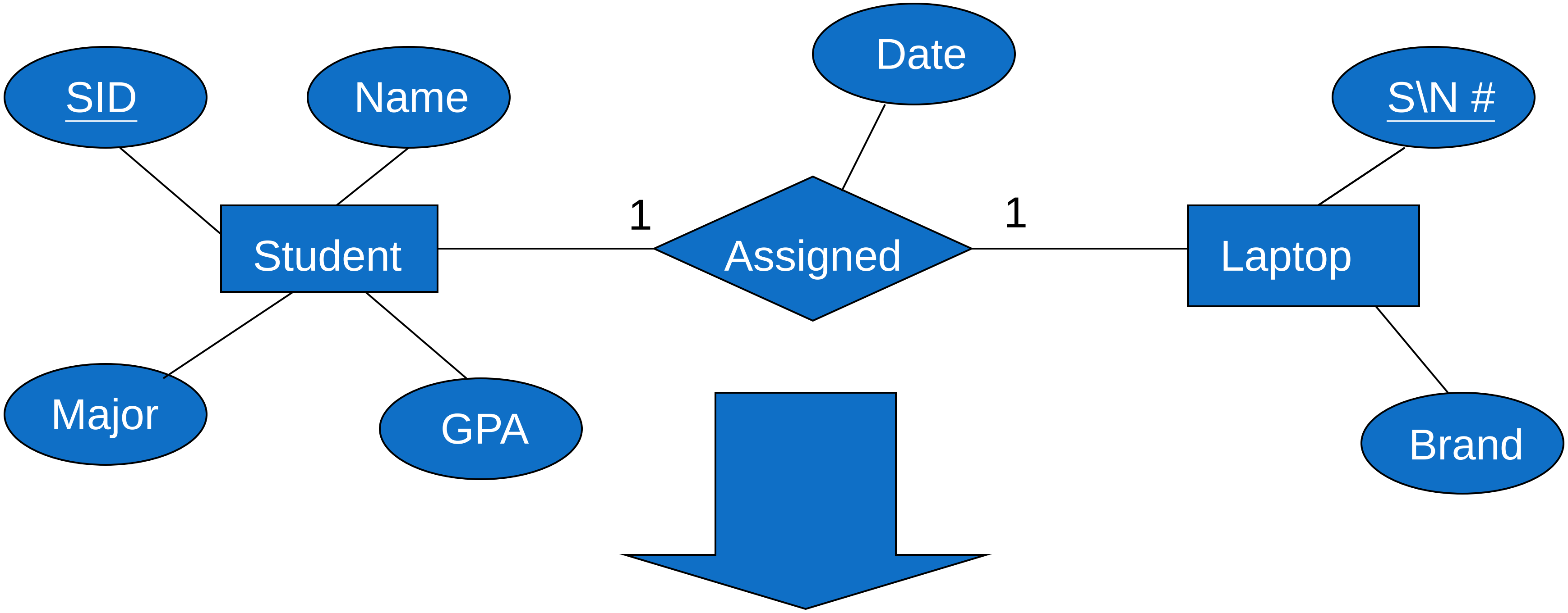
--This is a little complicated—

- Unary/Binary Relationship set
  - Depends on the cardinality and participation constraints
- N-ary (multiple) Relationship set
- Identifying Relationship

Mapping Relationship Set
Unary/Binary Relationship

- 1-1 relationship without total participation
  - **Relationship relation:** Build a table and add columns for each participating entity's primary key. Also add the attributes of the relationship. ***(cross-reference)***
- 1-1 relationship with one total participation
  - **Foreign key approach**: Add primary key of the entity without total participation in the table of the entity with total participation.
- **Merged relation (**alternate mapping): merge the two entities and the relationship into a single relation *(used when both participations are total)*.

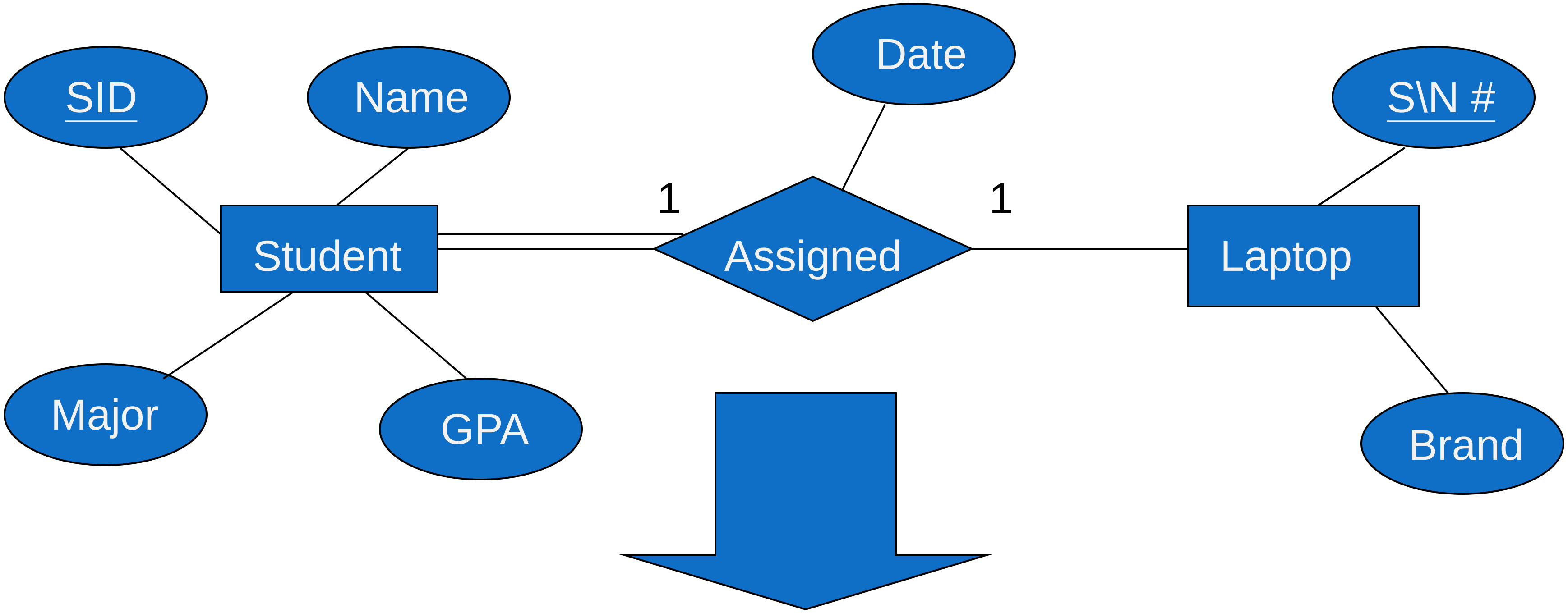
Example: Relationship Relation



SID	S\N#	Date
9999	07	12-08-09
8888	05	15-07-10

* **Primary key can be either *SID* or S\N#**

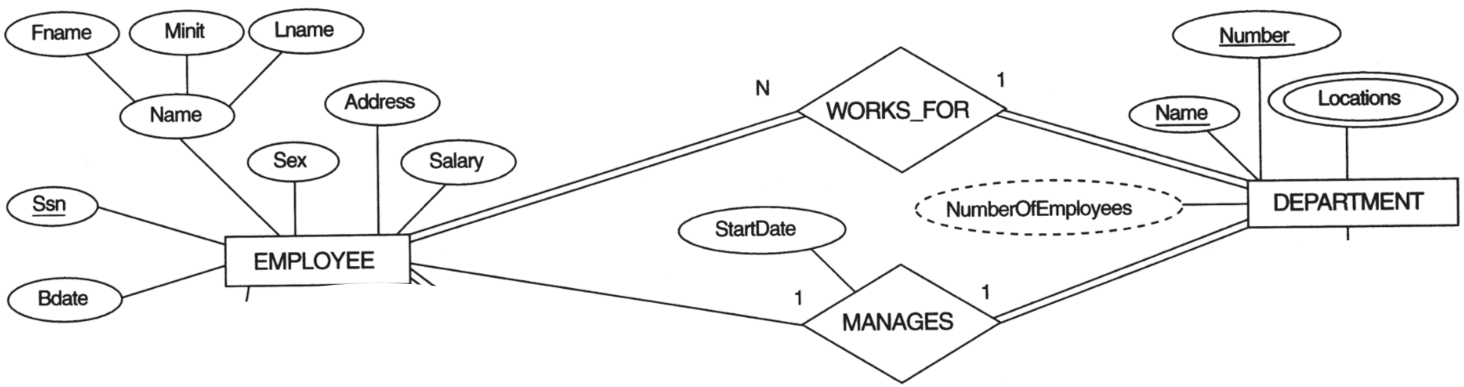
Example: Foreign Key Approach



SID	Name	GPA	Major	S\N#	Date
9999	Bart	3.2	1	11289	12-09-09
8888	Lisa	4.0	2	12345	14-02-10

FIGURE 7.1

THE ER CONCEPTUAL SCHEMA DIAGRAM FOR THE COMPANY DATABASE.

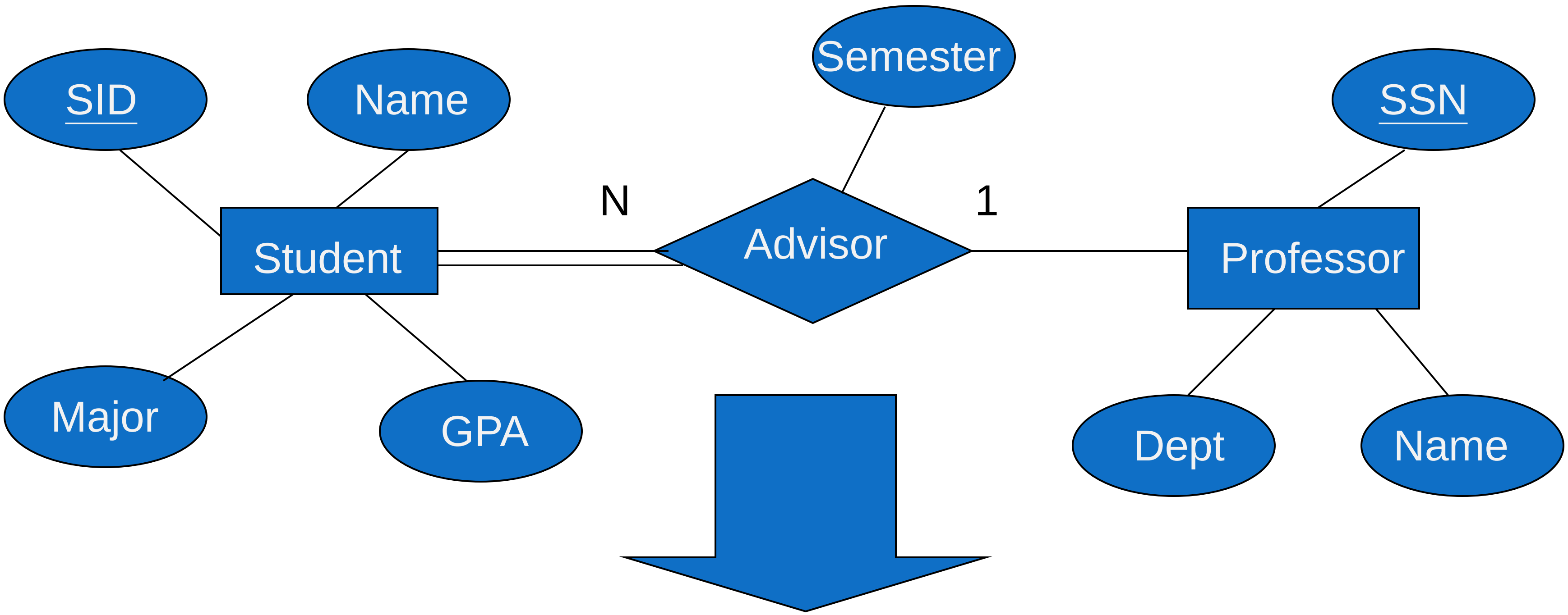


Representing Relationship Set
Unary/Binary Relationship

- 1-N relationship without total participation
  - Same as 1-1 relationship
  - **Relationship relation:** Build a table and add columns for each participating entity's primary key. Also add the attributes of the relationship. ***(cross-reference)***

- 1-N with total participation on **N** side
  - **Foreign key approach :** Add a column in the table of the entity on the **N** side, put in there the primary key of the entity on the **1** side.

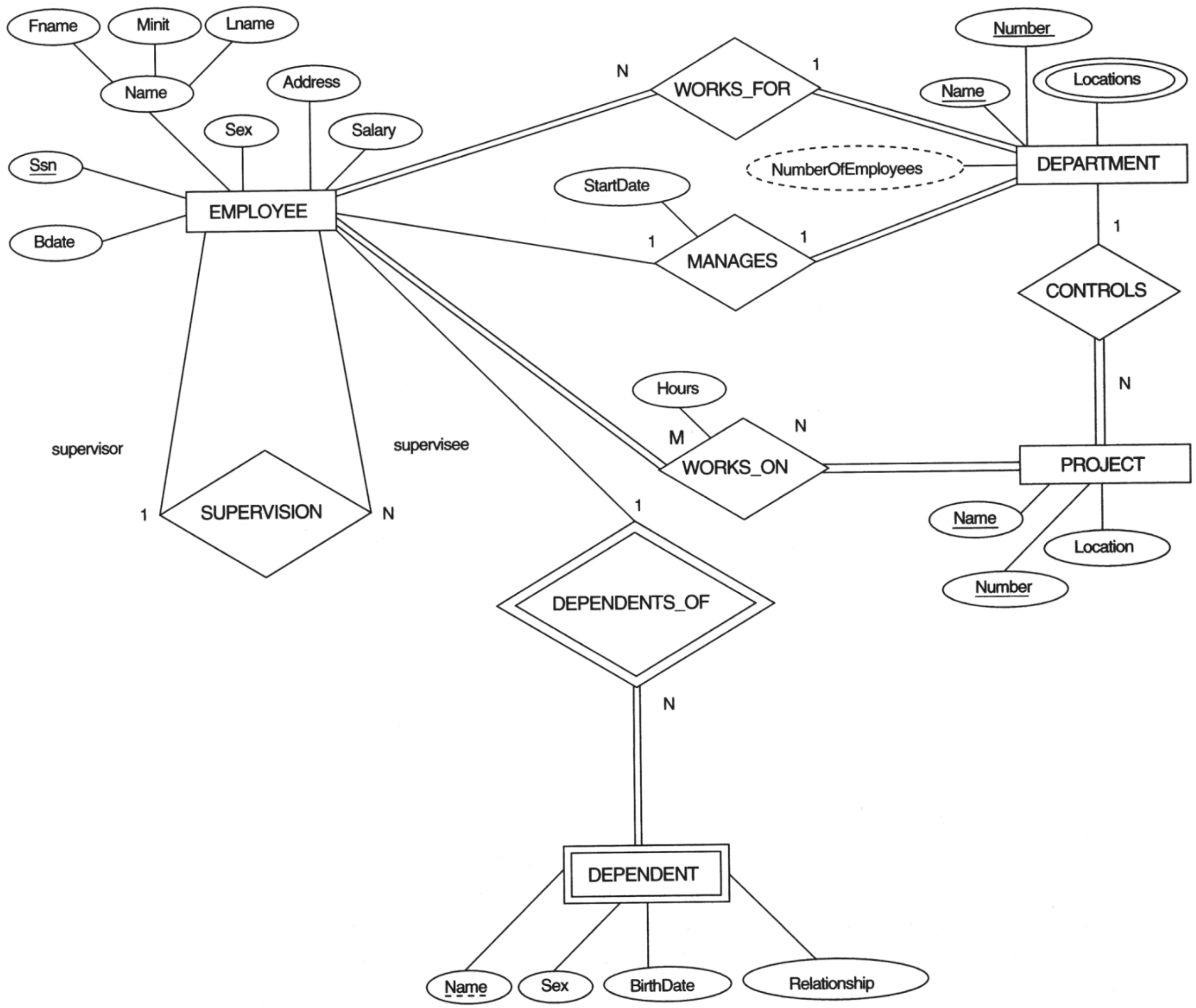
Example – 1:N Relationship Set



SID	Name	Major	GPA	Pro_SSN	Ad_Sem
9999	Ali	EE	3.0	123-456	Fall 2009
8888	Aliya	CS	3.8	567-890	Fall 2008

* Primary key of this table is *SID*

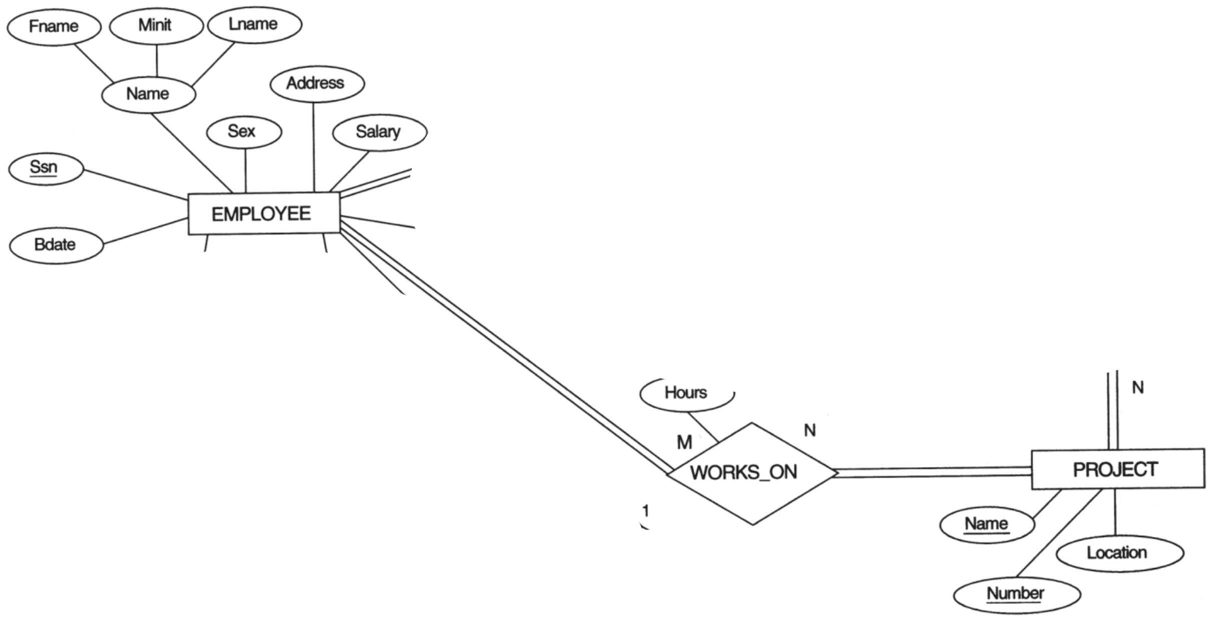
FIGURE 7.1
THE ER CONCEPTUAL SCHEMA DIAGRAM FOR THE COMPANY DATABASE.



Representing Relationship Set
Unary/Binary Relationship

- N:M relationship
  - **Relationship relation:** Build a table and add columns for each participating entity's primary key. Also add the attributes of the relationship. ***(cross-reference)***
  - Primary key of this new table is the union of the foreign keys of both entity sets.
  - Note No Foreign Key approach is possible…

FIGURE 7.1
THE ER CONCEPTUAL SCHEMA DIAGRAM FOR THE COMPANY DATABASE.

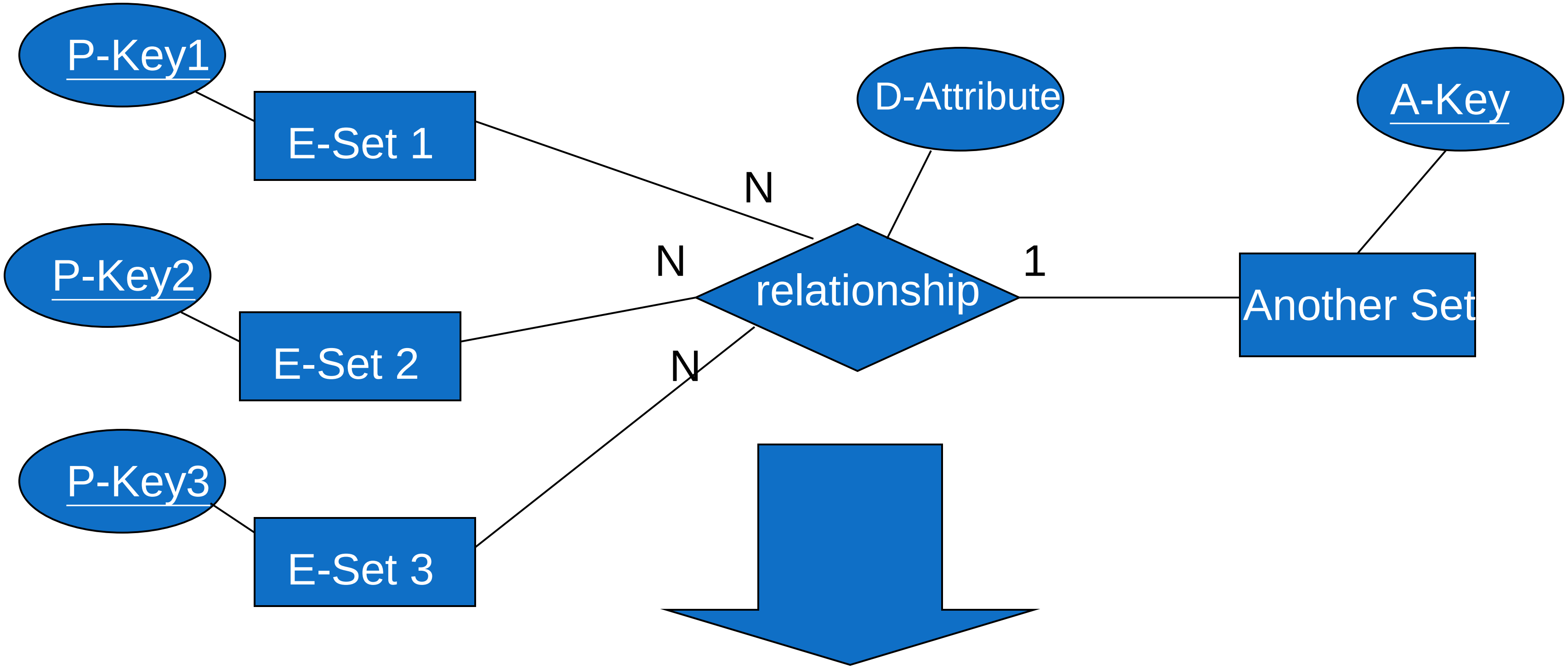


Representing Relationship Set
N-ary Relationship

- Intuitively Simple
  - Build a new table, add primary keys of all participating entity sets.
  - Add attributes of the relationship set
  - The primary key of this new table is the union of all primary keys of entities that are on **N** side

  - That is it, we are done.

Example – N-ary Relationship Set



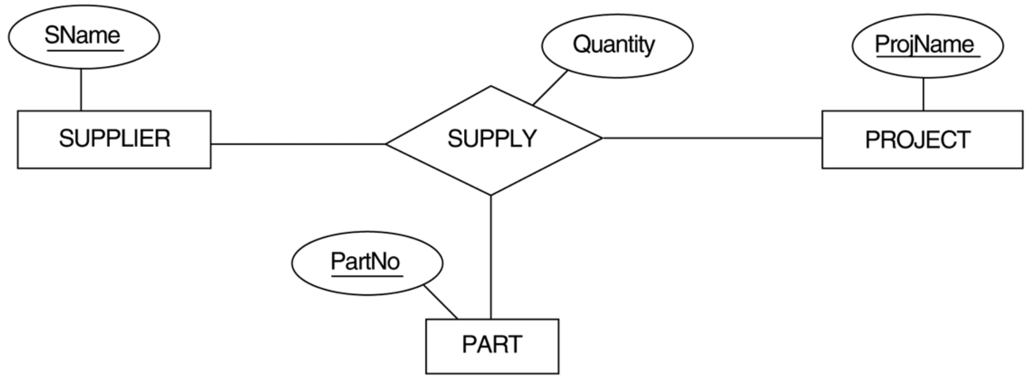
P-Key1	P-Key2	P-Key3	A-Key	D-Attribute
9999	8888	7777	6666	Yes
1234	5678	9012	3456	No

* Primary key of this table is *P-Key1 + P-Key2 + P-Key3*

FIGURE 4.11

Ternary relationship types. (a) The SUPPLY relationship.

(a)



SUPPLIER

SNAME	. . .

PROJECT

PROJNAME	. . .

PART

PARTNO	. . .

SUPPLY

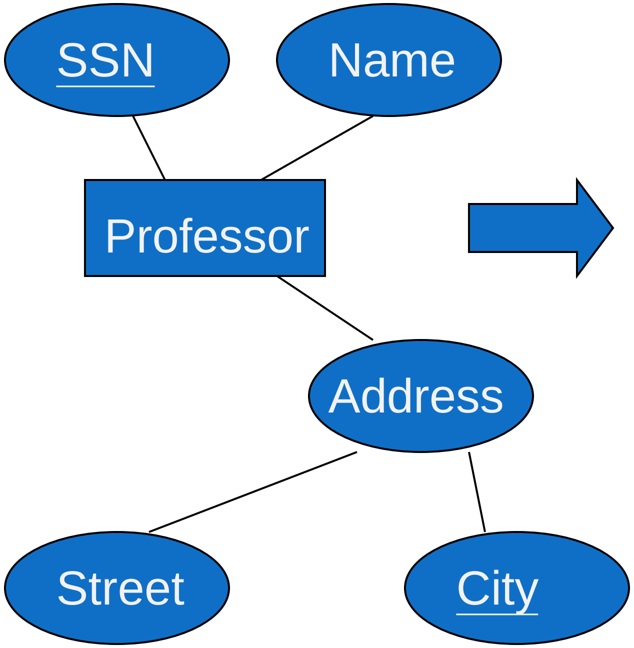
SNAME	PROJNAME	PARTNO	QUANTITY

Representing Relationship Set
Identifying Relationship

- Don't create a table for the identifying relationship
- As we have built a table for the corresponding weak entity
  - Reason:
    - A special case of 1:N with total participation
    - Reduce Redundancy

Representing Composite Attribute

- One column for each component attribute
- NO column for the composite attribute itself

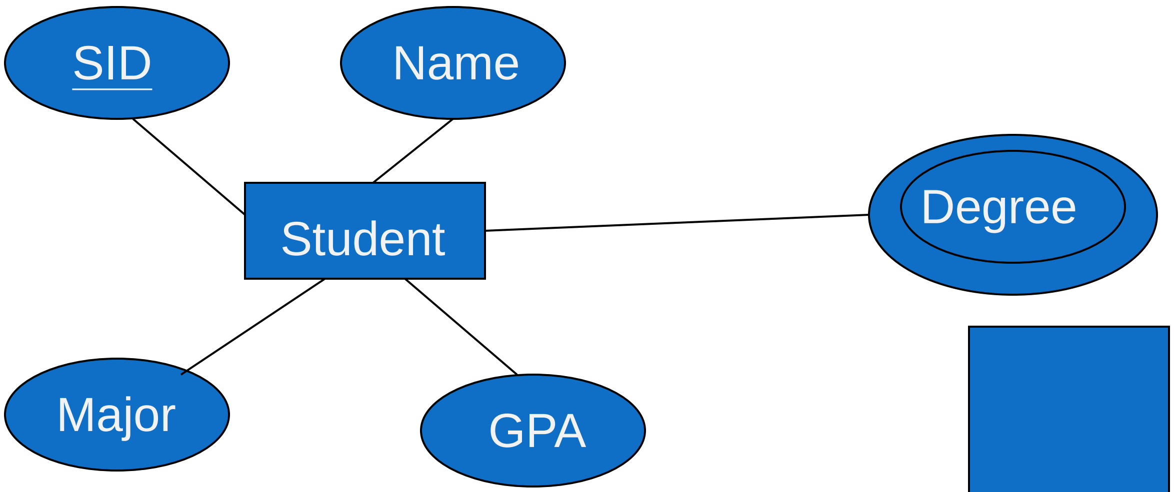


SSN	Name	Street	City
9999	Dr. Smith	50 1st St.	Fake City
8888	Dr. Lee	1 B St.	San Jose

Representing Multivalue Attribute

- Build a new relation schema with two columns
- Add the primary keys of the entity/relationship that has the multivalue attribute
- Add the multivalue attribute.
  - Each cell of this column holds only one value. So each value is represented as an unique tuple
- Primary key for this schema is the union of all attributes

Example – Multivalue attribute

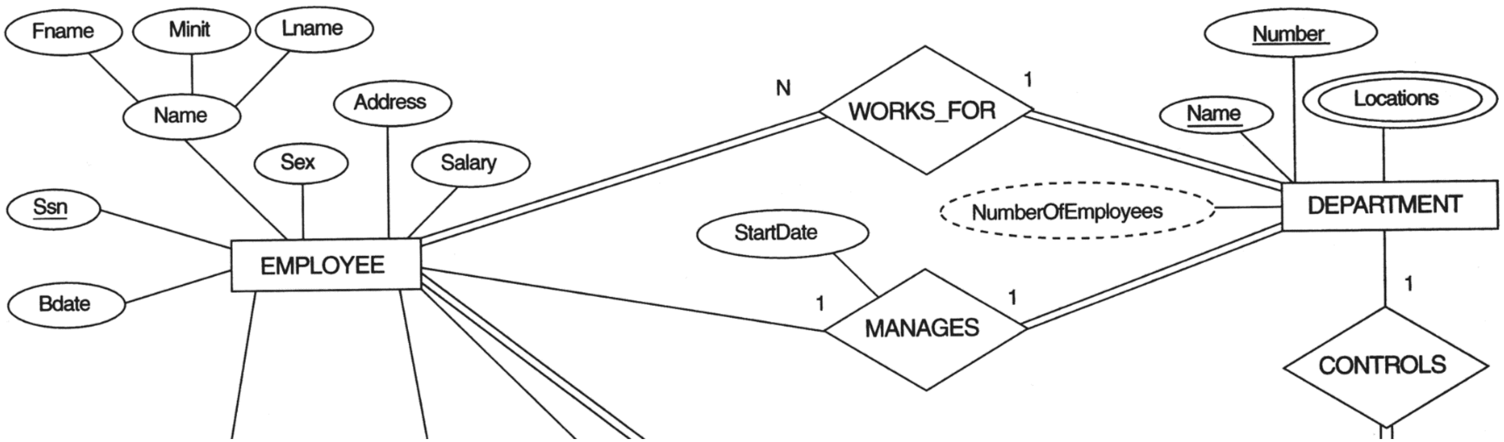


The primary key for this table is Student_SID + Degree, the union of all attributes

SID	Name	Major	GPA
1234	Javed	CS	2.8
5678	Saif	EE	3.6

Stud_SID	Degree
1234	FSC
1234	BS
5678	BS
5678	MS
5678	FA

Example – Multivalue attribute



DEPT_LOCATIONS

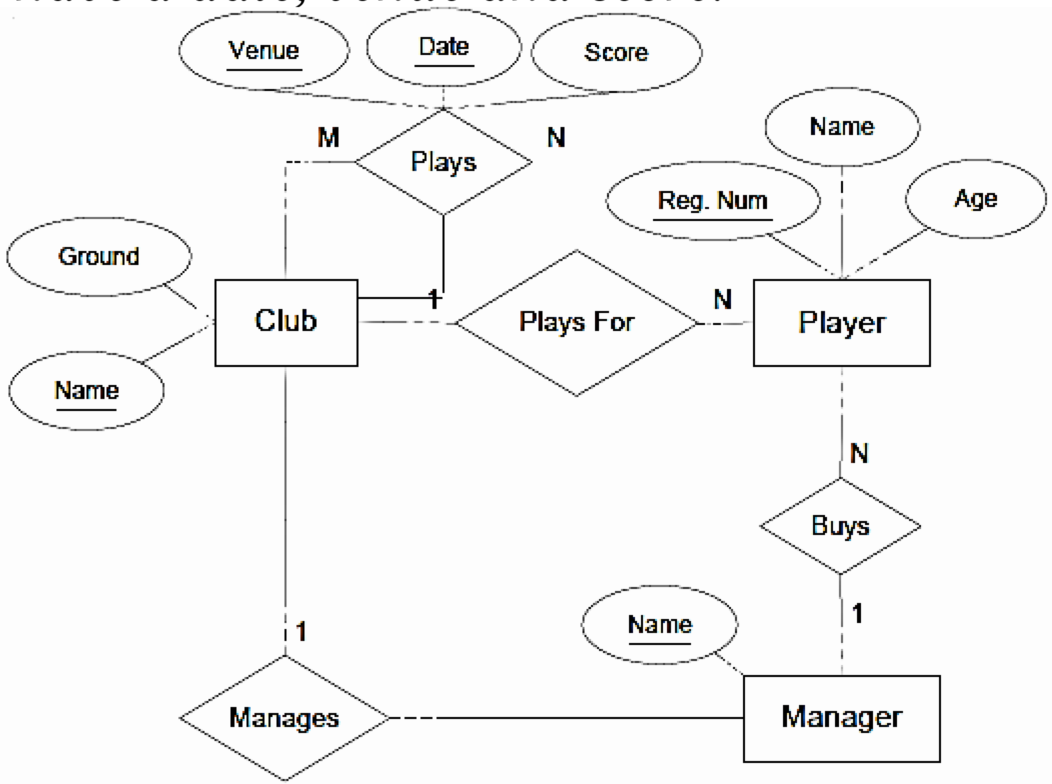
Dnumber	Dlocation

Correspondence between ER Model & Relational Model

ER Model	Relational Model
1. Entity type	1. Entity relation
2. 1:1 or 1:N relationship type	2. Foreign key (or relationship relation)
3. M:N relationship type	3. Relationship relation and two foreign keys
4. *n*-ary relationship type	4. Relationship relation and n foreign keys
5. Simple attribute	5. Attribute
6. Composite attribute	6. Set of simple component attributes
7. Multivalued attribute	7. Relation and foreign key
8. Value set	8. Domain
9. Key attribute	9. Primary (or secondary) key

ER to Relational: Example Football Club

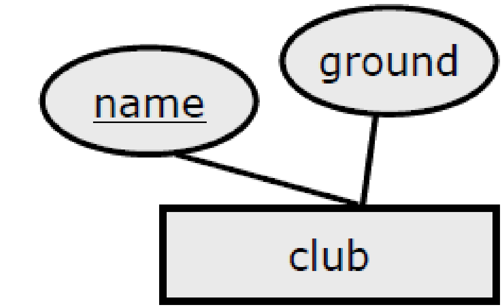
"A football club has a name and a ground and is made up of players. A player can play for only one club. A manager, represented by his name manages a club. A footballer has a registration number, name and age. A club manager also buys players. Each club plays against other clubs in the league and matches have a date, venue and score."



Dark lines represent total participation

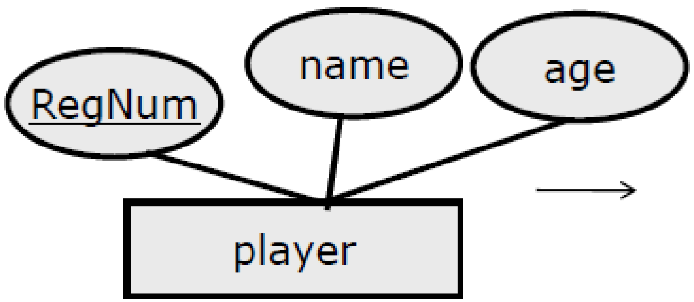
Example Step1

- Create a relation for each strong entity type:



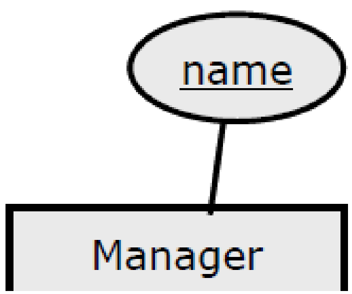
1. CLUB

<u>name</u>	ground



1. PLAYER

<u>regnum</u>	name	age

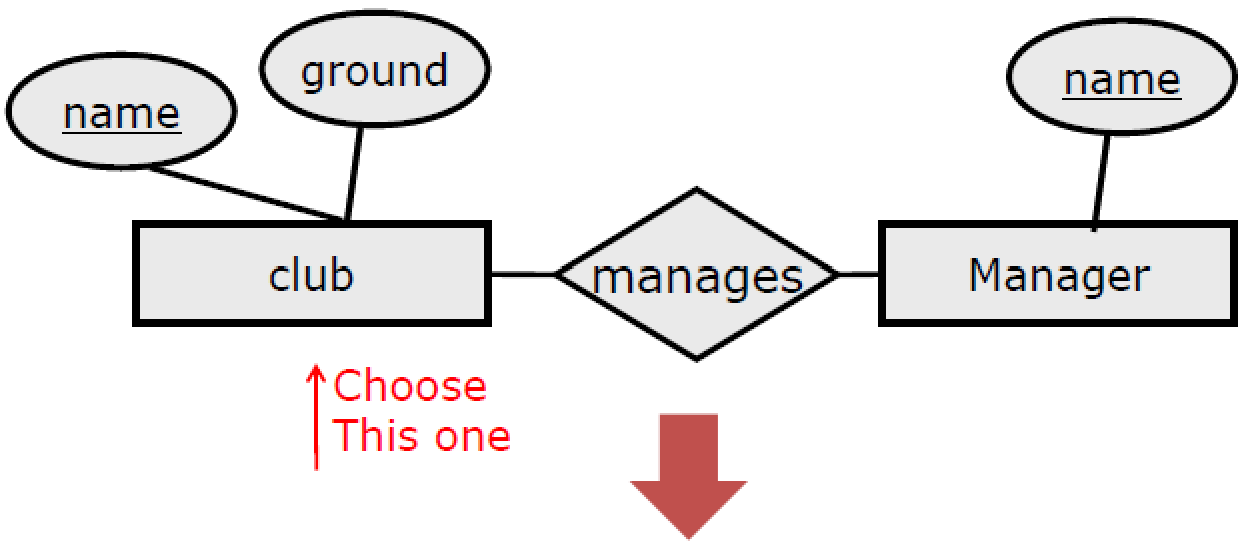


1. MANAGER

<u>mgrname</u>

EXAMPLE STEP 2 &3

- No weak entity types (Step 2), so move on to next step
- For each binary 1:1 relationship choose an entity and include the other's PK as a FK



1. 3 1.

CLUB

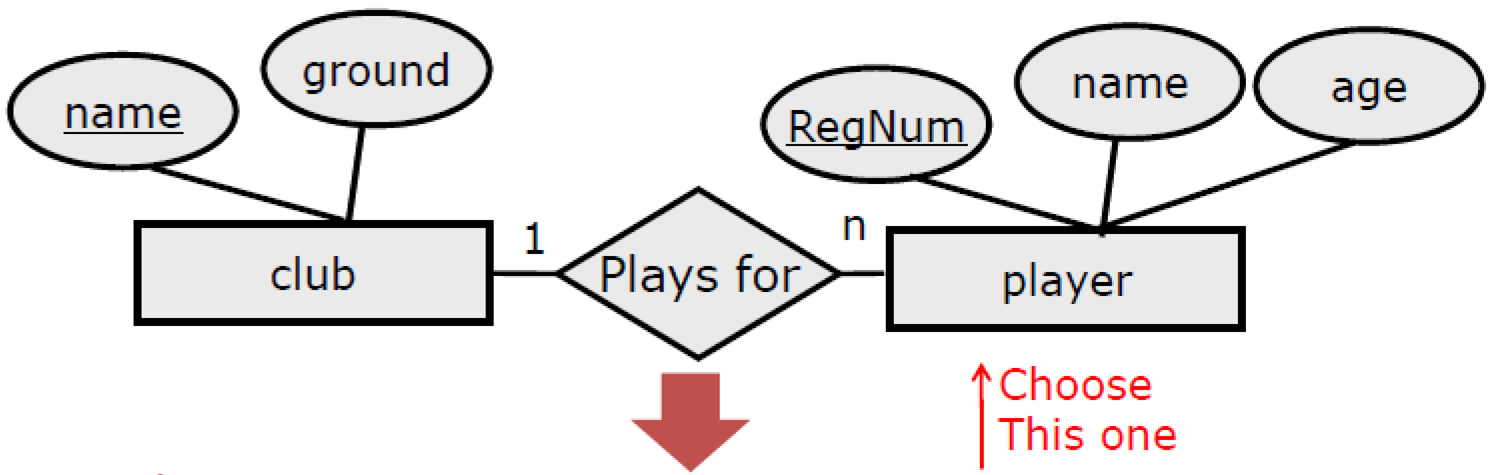
name	ground	mgrname

MANAGER

mgrname

Example Step 4

- For each binary 1:n relationship choose the n-side entity and include a FK with respect to the other entity
- (a) Player *plays for* Club



1.

3

CLUB

name	ground	mgrname

MANAGER

mgrname

1.

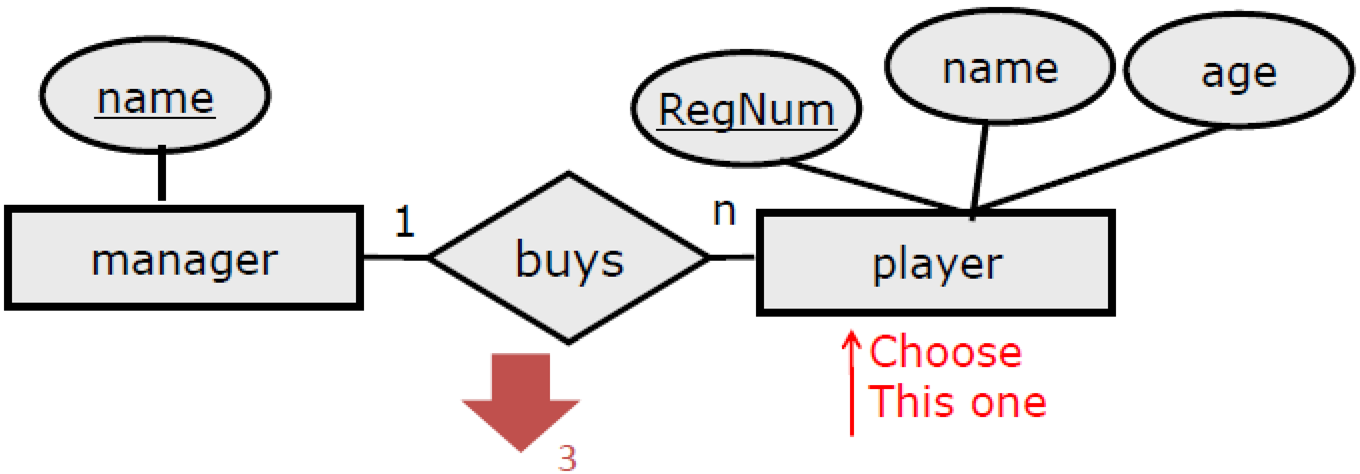
PLAYER

regnum	name	age	clubname

4.

Example Step 4

- For each binary 1:n relationship choose the n-side entity and include a FK with respect to the other entity
- (b) Manager *buys* Player



1.

CLUB

name	ground	mgrname

1.

MANAGER

mgrname

1.

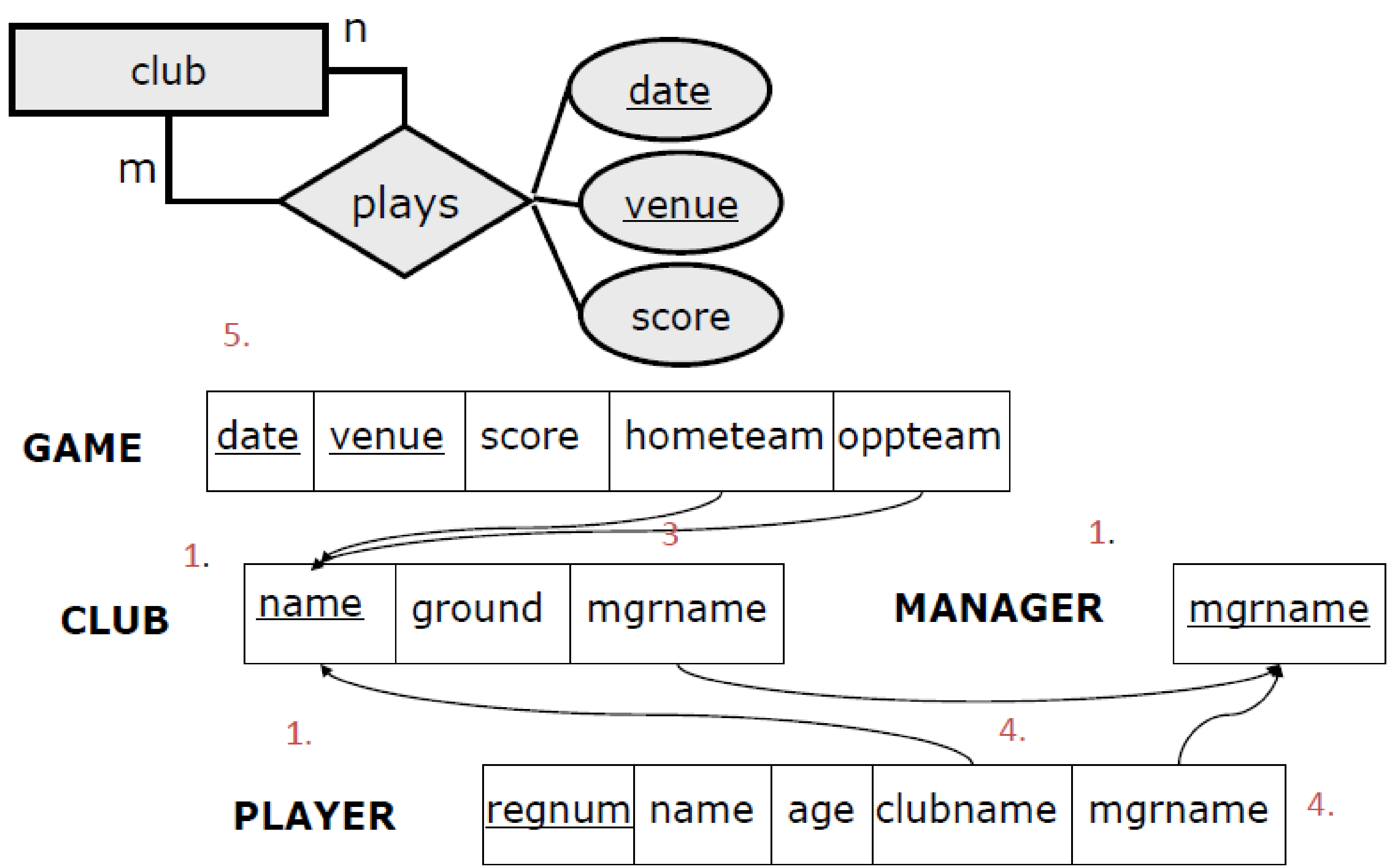
PLAYER

regnum	name	age	clubname	mgrname

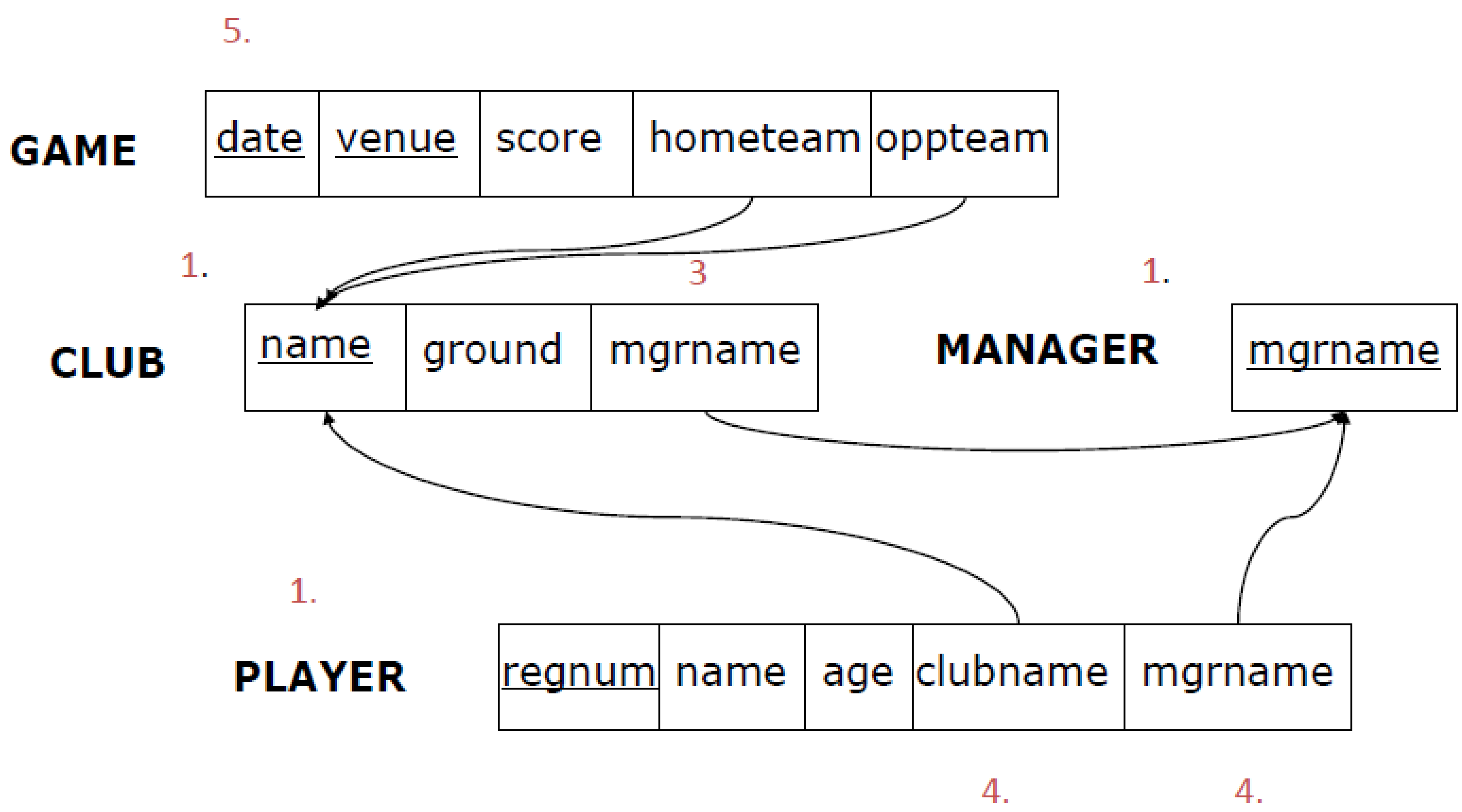
4. 4.

Example Step 5

- For each binary n:m relationship, create a relation for the relationship:



EXAMPLE – FINAL RELATION



Correspondence between ER Model & Relational Model

ER Model	Relational Model
1. Entity type	1. Entity relation
2. 1:1 or 1:N relationship type	2. Foreign key (or relationship relation)
3. M:N relationship type	3. Relationship relation and two foreign keys
4. *n*-ary relationship type	4. Relationship relation and n foreign keys
5. Simple attribute	5. Attribute
6. Composite attribute	6. Set of simple component attributes
7. Multivalued attribute	7. Relation and foreign key
8. Value set	8. Domain
9. Key attribute	9. Primary (or secondary) key

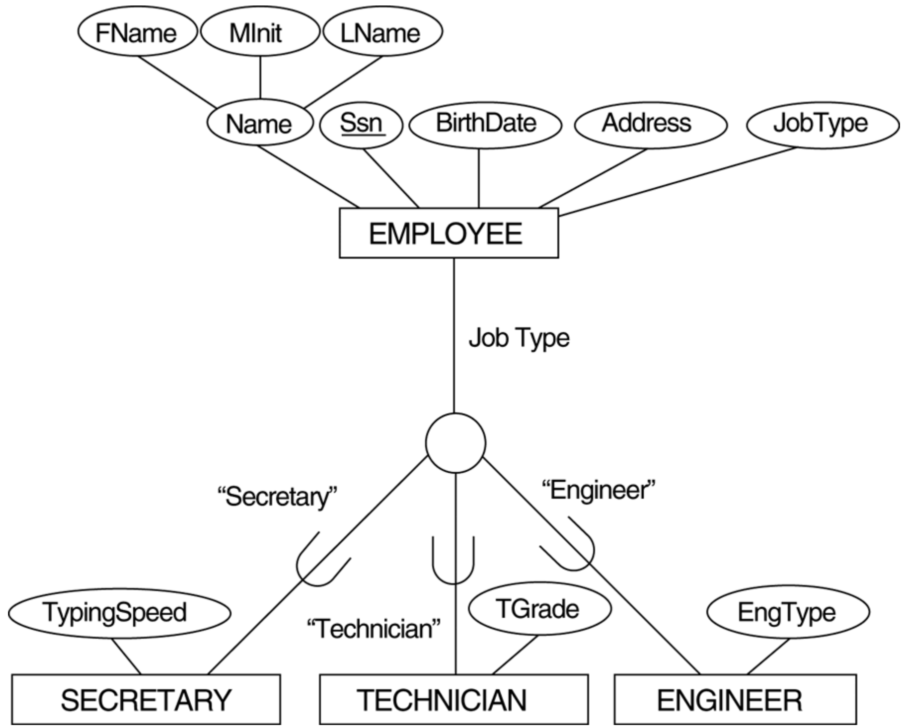
Mapping EER Model Constructs to Relations

- For Mapping Specialization or Generalization we have four options:
  - Multiple relations-Superclass and subclasses
  - Multiple relations-Subclass relations only
  - Single relation with one type attribute
  - Single relation with multiple type attributes

Mapping EER Model Constructs to Relations

- **Multiple relations- Superclass and Subclasses**
  - Create a relation for the Superclass
  - Create a relation for each subclass and also include the primary key of the Superclass
  - This option works for any specialization (total or partial, disjoint or over-lapping).

ATTRIBUTE-DEFINED SPECIALIZATION ON JOBTYPE



(a) EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType

SECRETARY

SSN	TypingSpeed

TECHNICIAN

SSN	TGrade

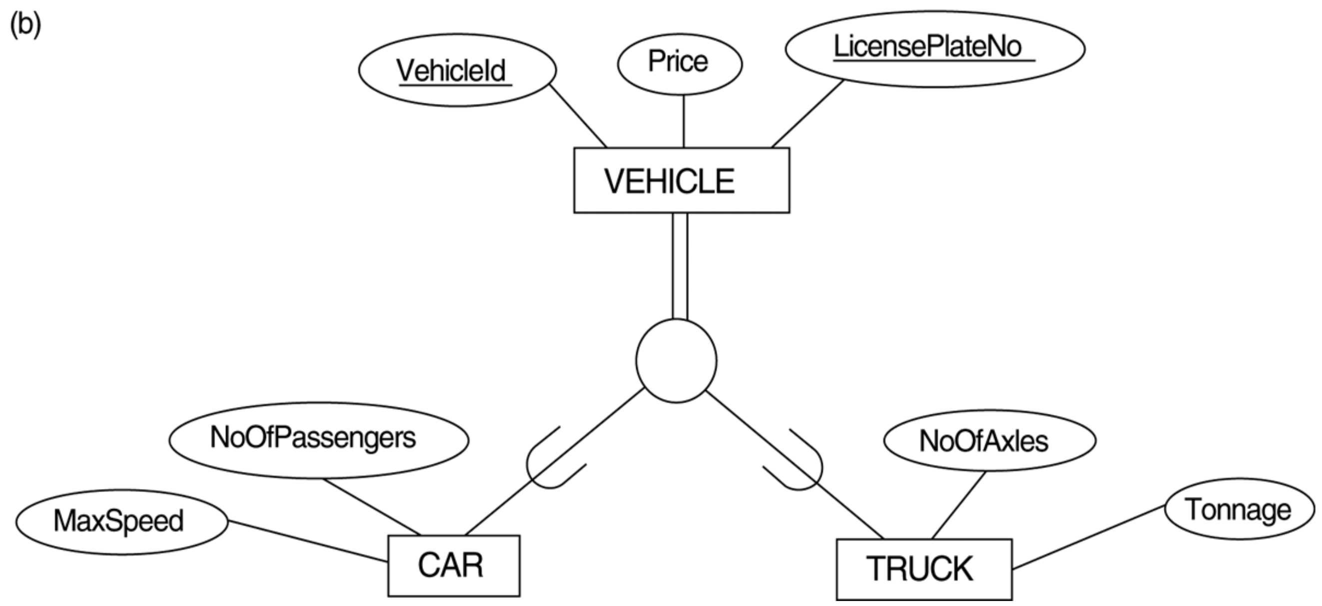
ENGINEER

SSN	EngType

Mapping EER Model Constructs to Relations

- **Multiple Relations-Subclass relations only**
  - Create a relation for each subclass and include the attributes of the superclass in each subclass relation
  - This option only works for a specialization whose subclasses are total
    - Every entity in the superclass must belong to at least one of the subclasses.
  - It is preferred that subclasses are disjoint (to avoid redundancy)
  - Need Outer join (or full outer join) to get all entities

Generalizing CAR and TRUCK into the superclass VEHICLE.



(b) CAR

VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

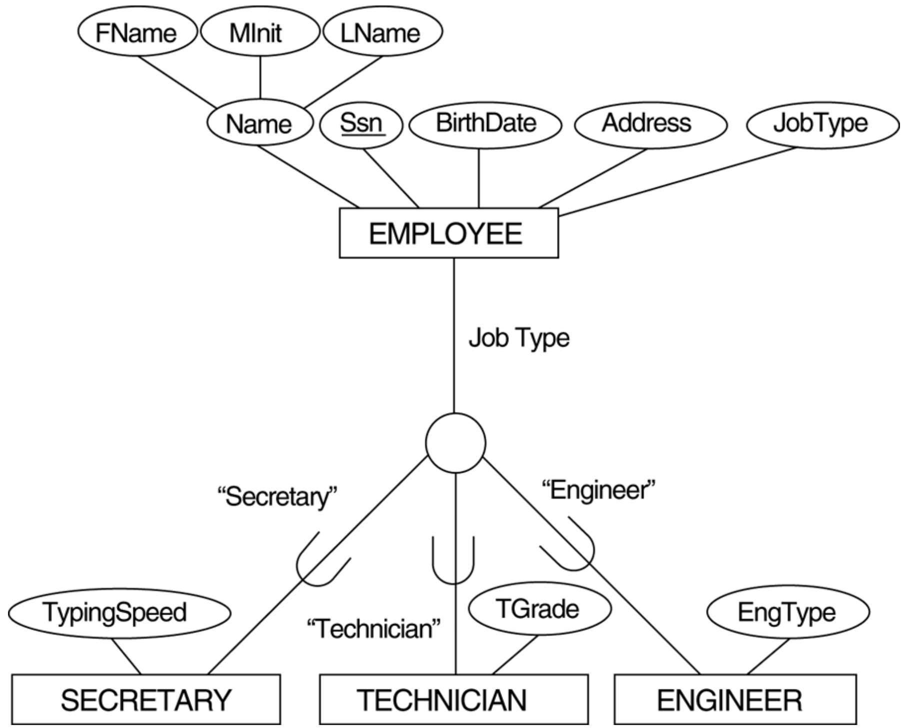
TRUCK

VehicleId	LicensePlateNo	Price	NoOfAxles	Tonnage

Mapping EER Model Constructs to Relations

- **Single relation with one type attribute**
  - Create a single relation for superclass and all of the subclasses
  - The new relation includes the attributes of superclass and all the attributes of each subclass
  - The relation also includes an attribute that indicates the subclass to which each tuple belongs
  - Not recommended if subclasses have many attributes
  - This option works only for a specialization whose subclasses are *disjoint,*

ATTRIBUTE-DEFINED SPECIALIZATION ON JOBTYPE



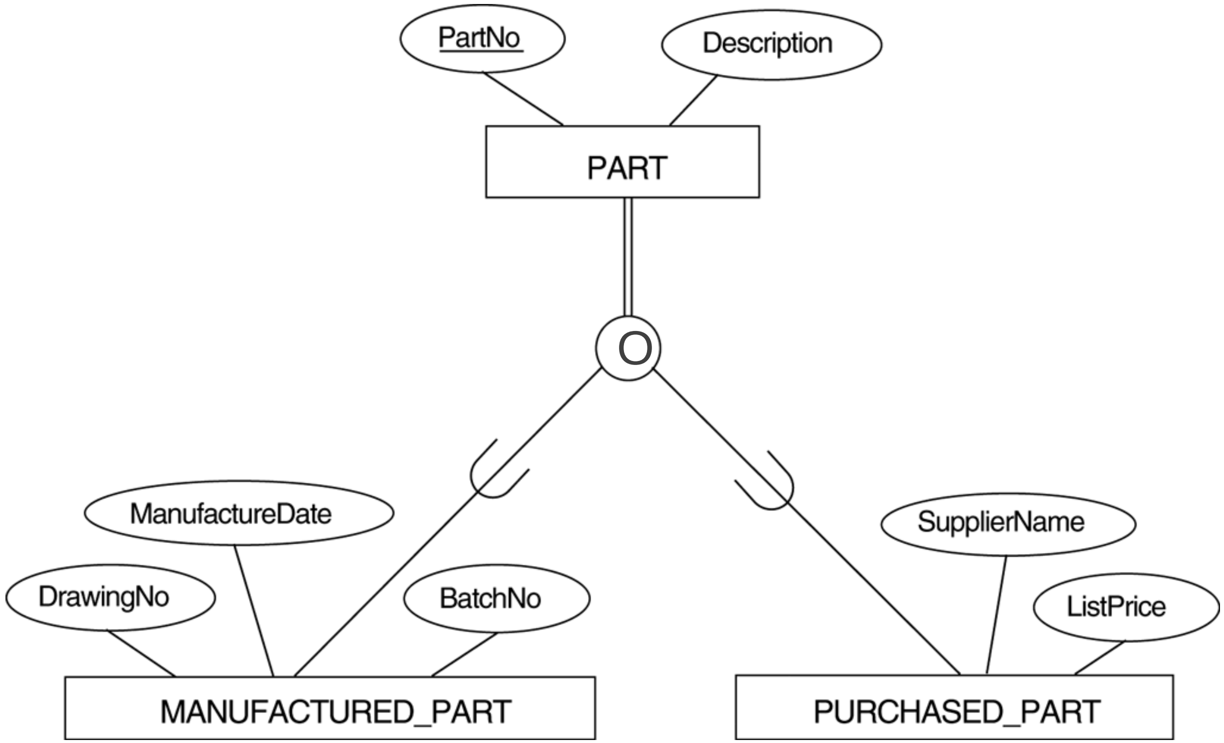
(c) EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	EngType

Mapping EER Model Constructs to Relations

- **Single relation with multiple type attributes**
  - Create a single relation for superclass and all of the subclasses
  - The new relation includes the attributes of superclass and all the attributes of each subclass
  - The relation also includes m type attributes, that is $\{t_1, t_2, \ldots, t_m\}$, where m is the no of subclasses.
  - Each t_i, $1 < i < m$, is a Boolean type attribute indicating whether a tuple belongs to the i^{th} subclass.
  - This option is for overlapping subclasses (but will work for a disjoint subclasses).

FIGURE 4.5
EER DIAGRAM NOTATION FOR AN OVERLAPPING SPECIALIZATION.



(d) PART

PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

Mapping of Shared Subclasses (Multiple Inheritance)

- A shared subclass, is a subclass of several classes, indicating multiple inheritance.
- These classes must all have the same key attribute. WHY ?
  - Otherwise, the shared subclass would be modeled as a category.
- We can apply any of the options discussed before for Specialization\Generalization to a shared subclass, subject to the restriction.

FIGURE 4.7
A SPECIALIZATION LATTICE WITH MULTIPLE INHERITANCE FOR A UNIVERSITY DATABASE.

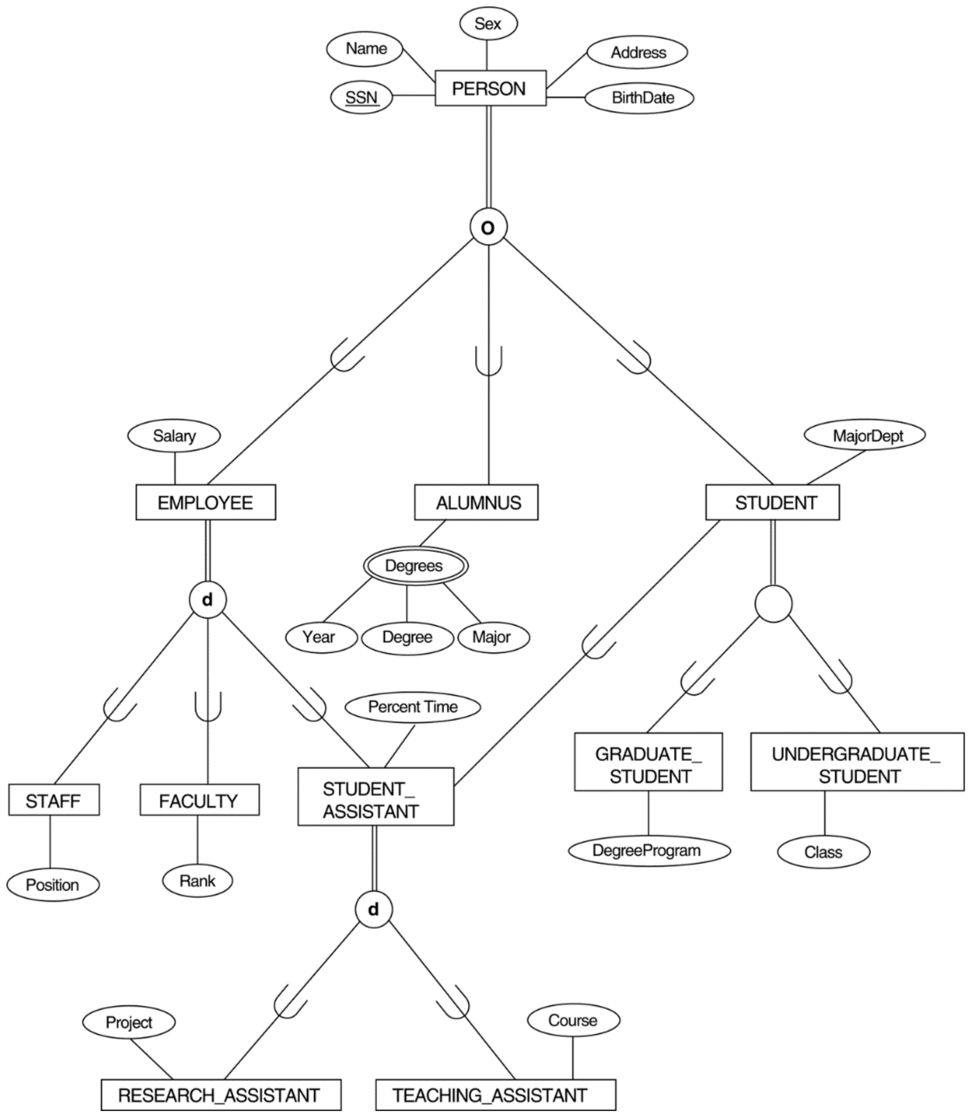


FIGURE 7.5

MAPPING THE EER SPECIALIZATION LATTICE IN FIGURE 4.6 USING MULTIPLE OPTIONS.

PERSON

SSN	Name	BirthDate	Sex	Address

EMPLOYEE

SSN	Salary	EmployeeType	Position	Rank	PercentTime	RAFlag	TAFlag	Project	

ALUMNUS

SSN

ALUMNUS_DEGREES

SSN	Year	Degree	

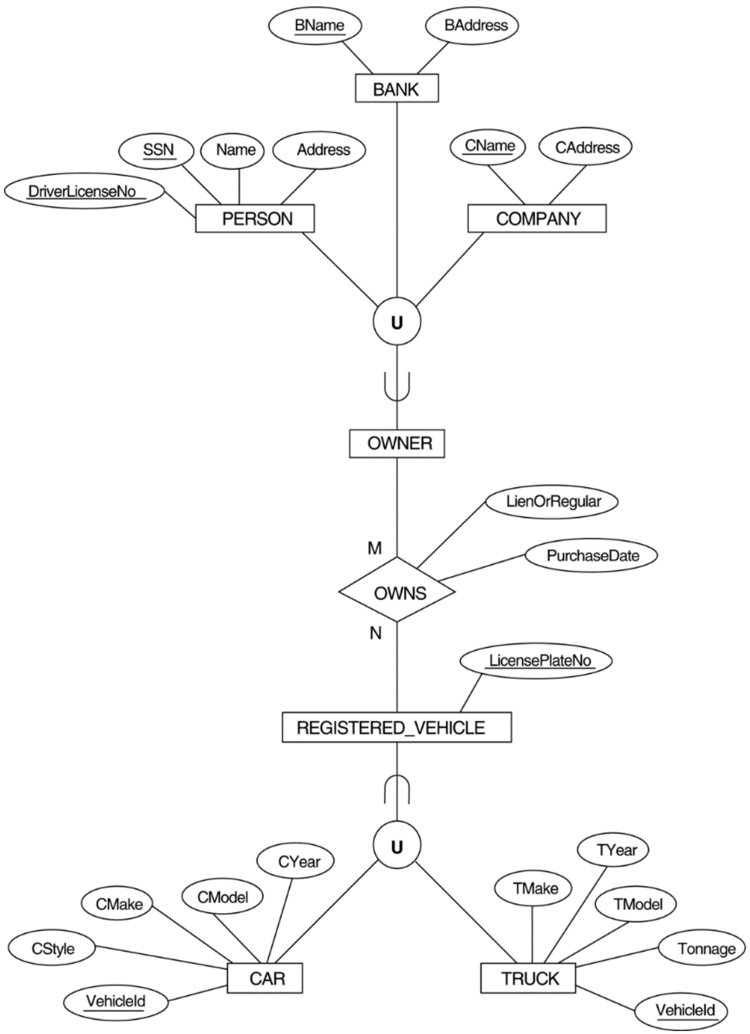
STUDENT

SSN	MajorDept	GradFlag	UndergradFlag	DegreeProgram	Class	StudAssistFlag

Mapping of Union Types (Categories)

- For mapping a category whose superclass have different keys, we specify a new key attribute, called a surrogate key.

FIGURE 4.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.



PERSON

SSN	DriverLicenseNo	Name	Address	OwnerId

BANK

BName	BAddress	OwnerId

COMPANY

CName	CAddress	OwnerId

OWNER

OwnerId

REGISTERED_VEHICLE

VehicleId	LicensePlateNumber

CAR

VehicleId	CStyle	CMake	CModel	

TRUCK

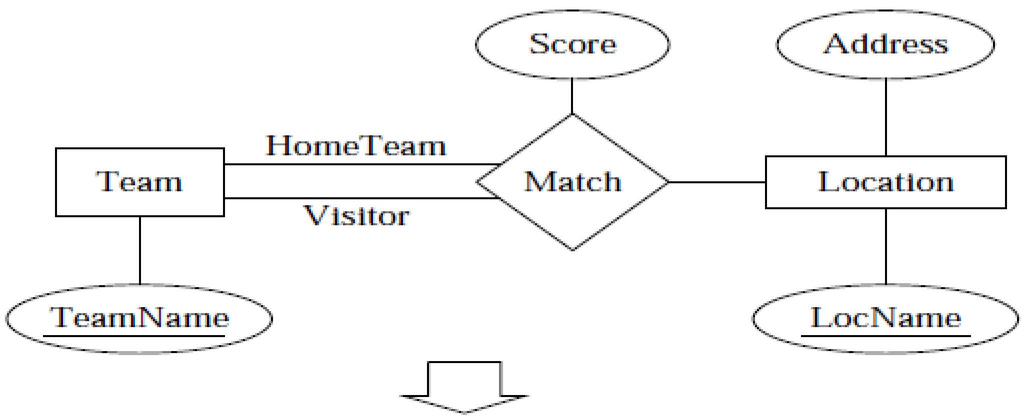
VehicleId	TMake	TModel	Tonnage	TYear

OWNS

OwnerId	VehicleId	PurchaseDate	LienOrRegular

Mapping Exercise 1

Example:



Team

TeamName

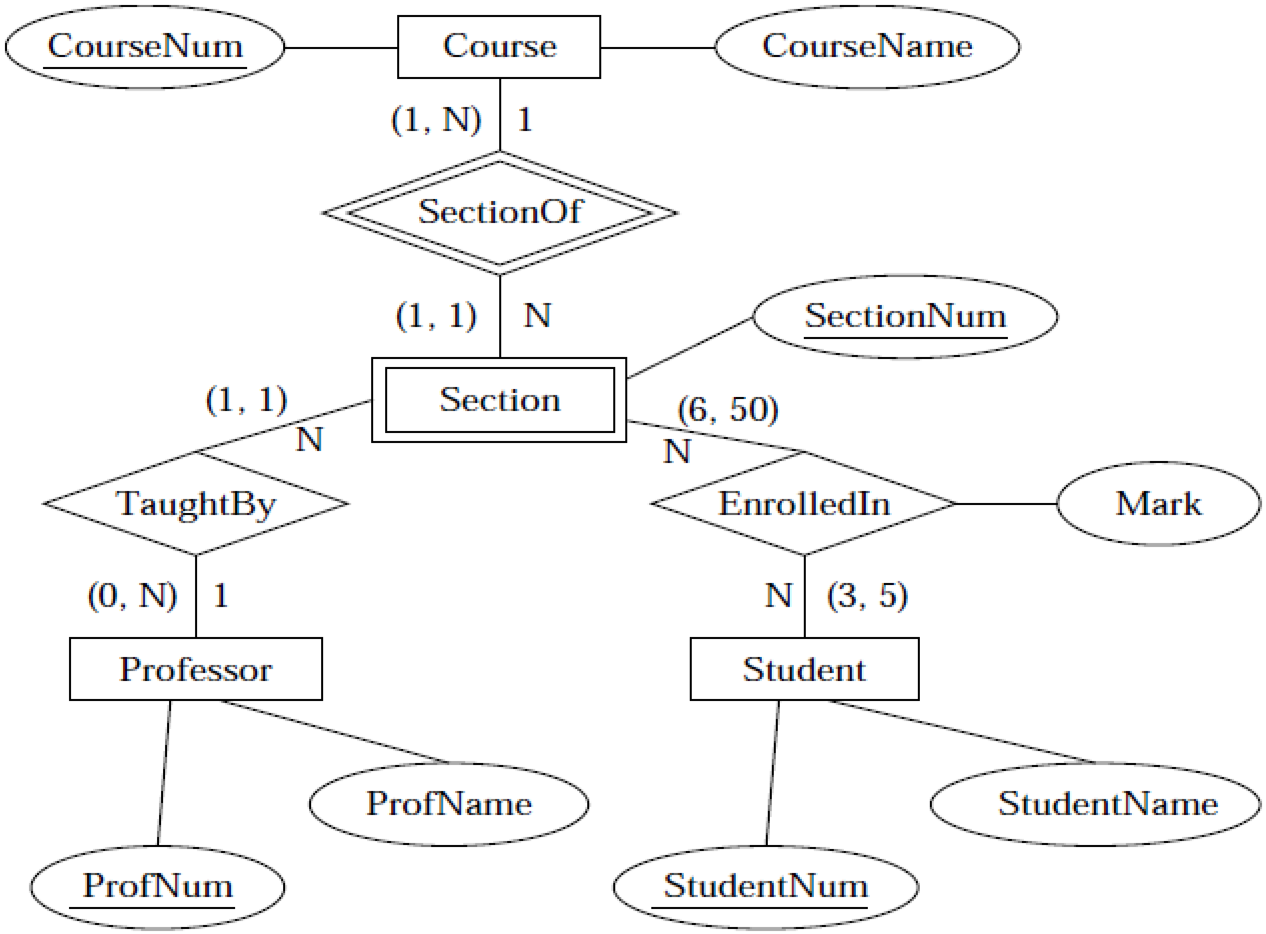
Location

LocName	Address

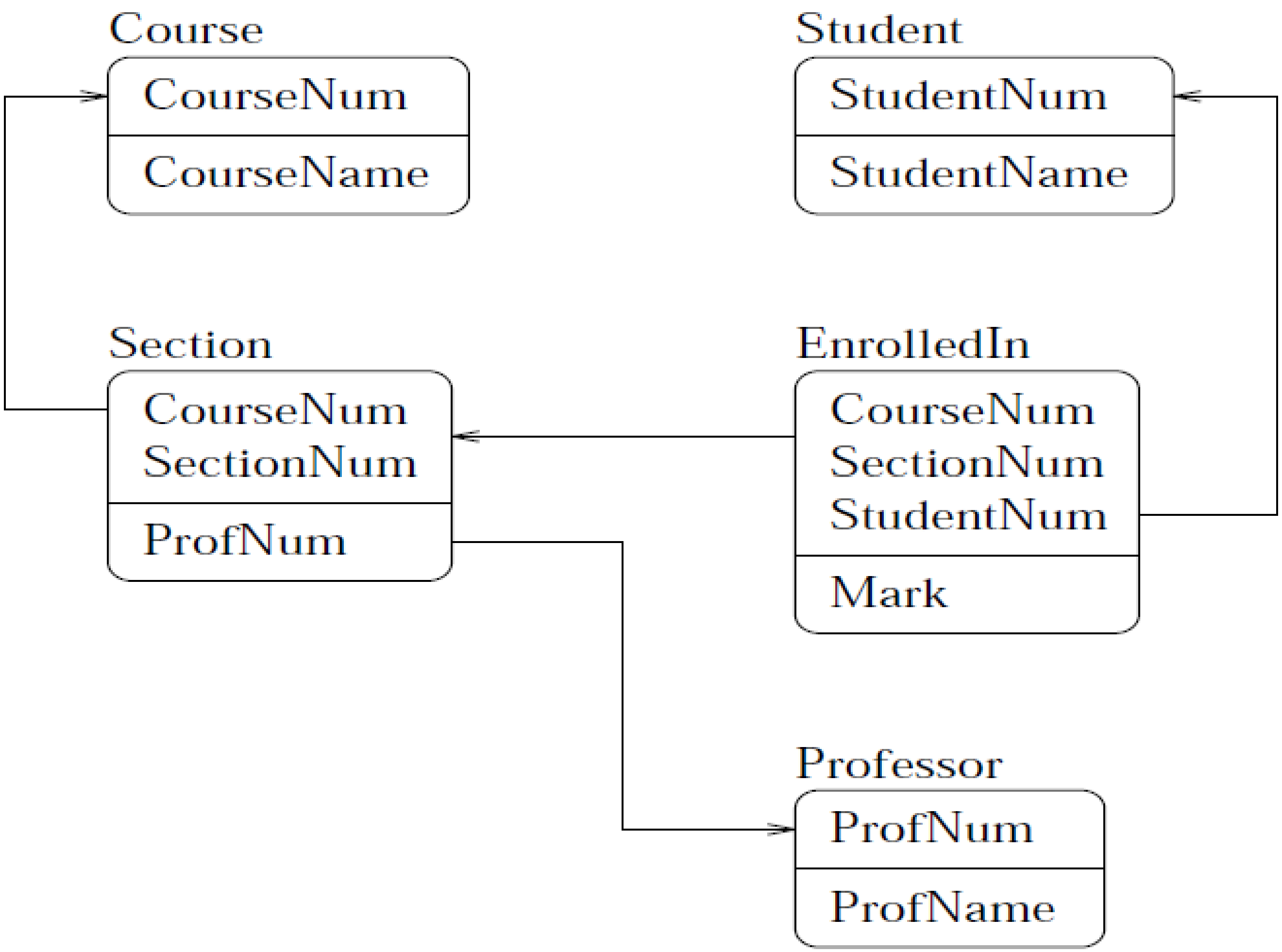
Match

HomeTeamName	VisitorTeamName	LocName	Score

Mapping Exercise 2

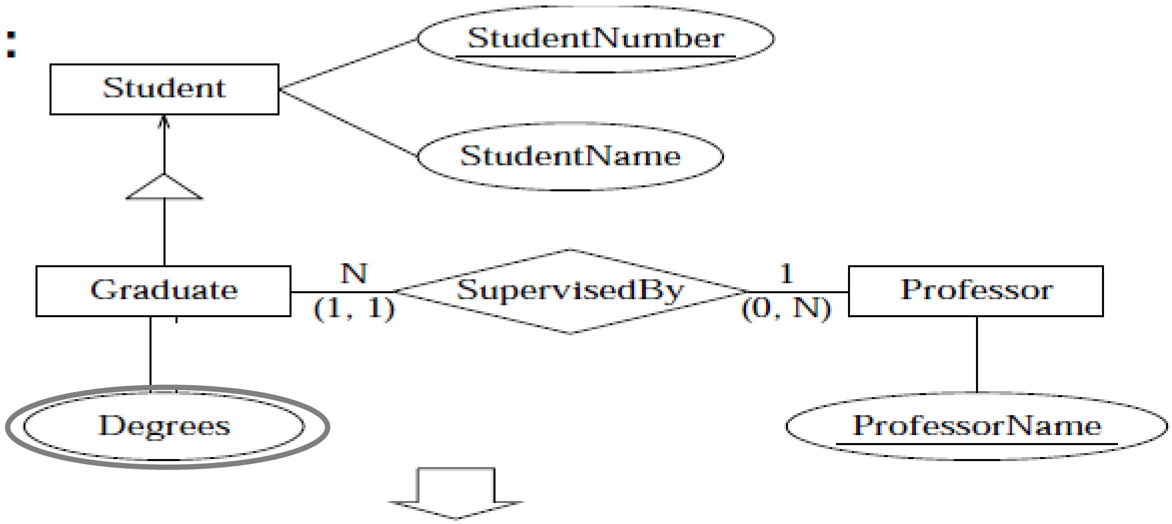


MAPPING EXERCISE 2 -RESULT



Specialization Mapping Example

Example:



Student

StudentNumber	StudentName

Professor

ProfessorName

Graduate

StudentNumber	ProfessorName

Degree

StudentNumber	Degree