

# CSC343 Worksheet 15 Solution (Final)

July 15, 2020

1.
  - E/R Diagram



- UML



### Notes:

- UML
  - Was developed originally as a graphical notation for describing software designs in an object-oriented style
  - Offers the same as E/R model, with the exception of multiway relationship

UML	E/R Model
Class	Entity set
Association	Binary relationship
Association Class	Attributes on a relationship
Subclass	Isa hierarchy
Aggregation	Many-one relationship
Composition	Many-one relationship with referential integrity

- UML Class



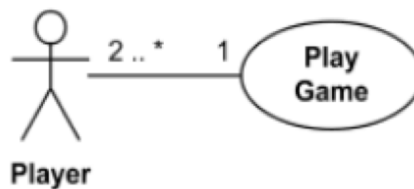
- Associations



### Multiplicity in UML

Multiplicity	Option	Cardinality
0..0	0	Collection must be empty
0..1		No instances or one instance
1..1	1	Exactly one instance
0..*	*	Zero or more instance
5..5	5	Exactly 5 instances
$m..n$		At least $m$ but no more than $n$ instances

### Example:



Two or more Player actors are required to initiate **one** Play Game use case.

### References:

- 1) uml-diagrams, UML Multiplicity and Collections, link
- Referential Integrity
    - Means that a value appearing in one context must also appear in another



Means studio can have **at most one** president  
but it could not have a president at some time



Means one studio can have **at least one** movies  
and there could have many more (to infinity!)

- Self-Associations

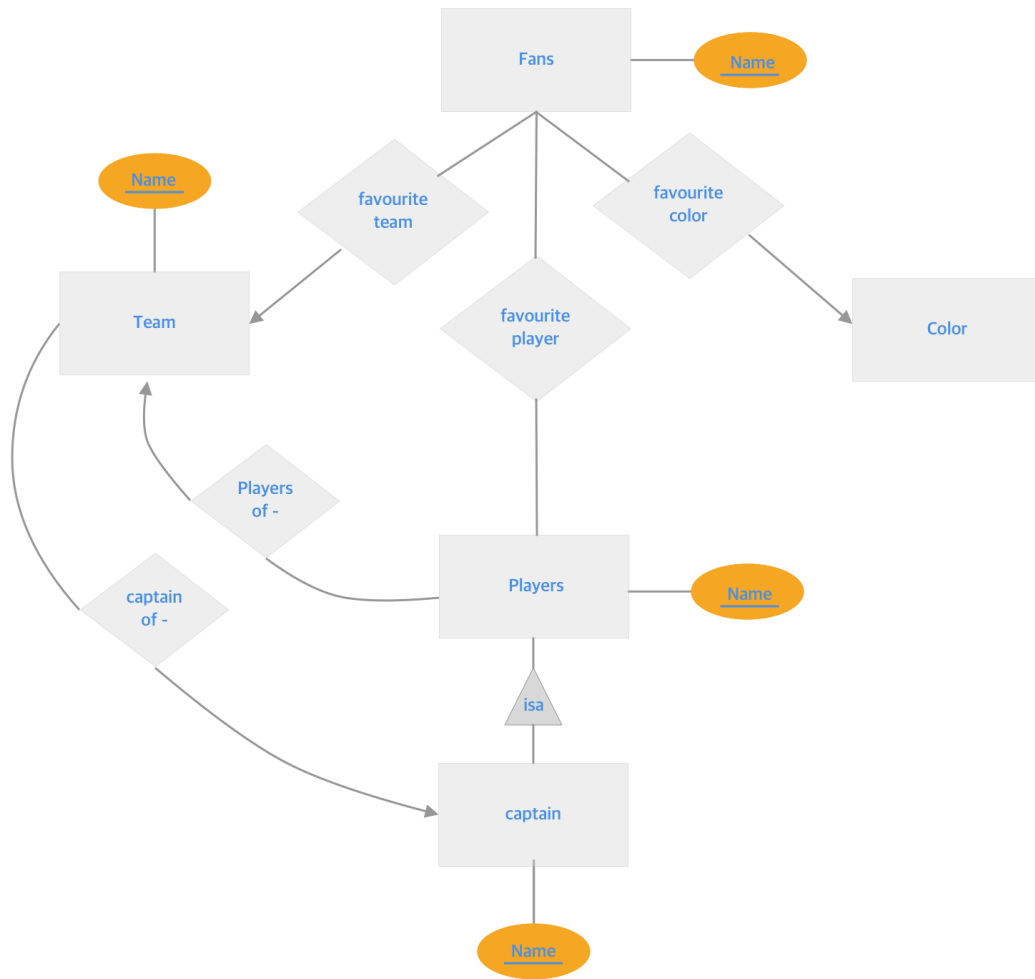




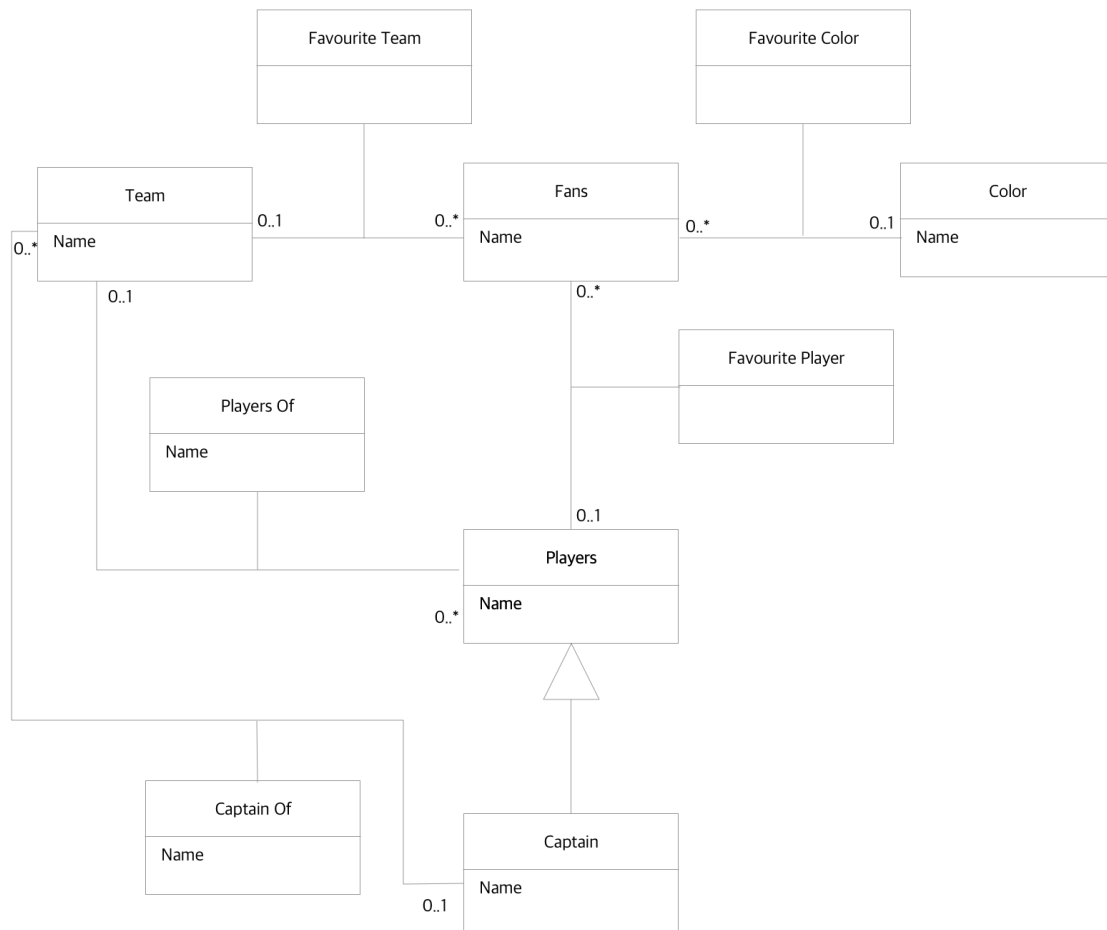
b) Solution:



3. • E/R Diagram

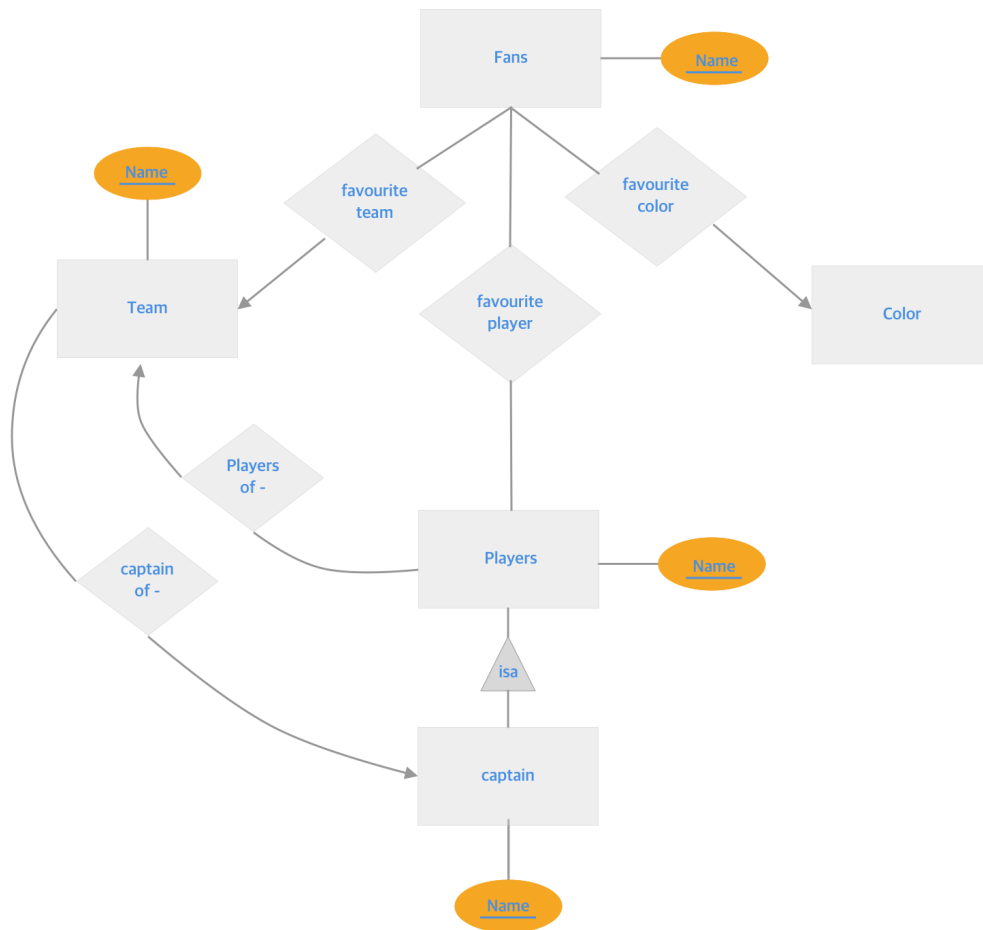


- UML

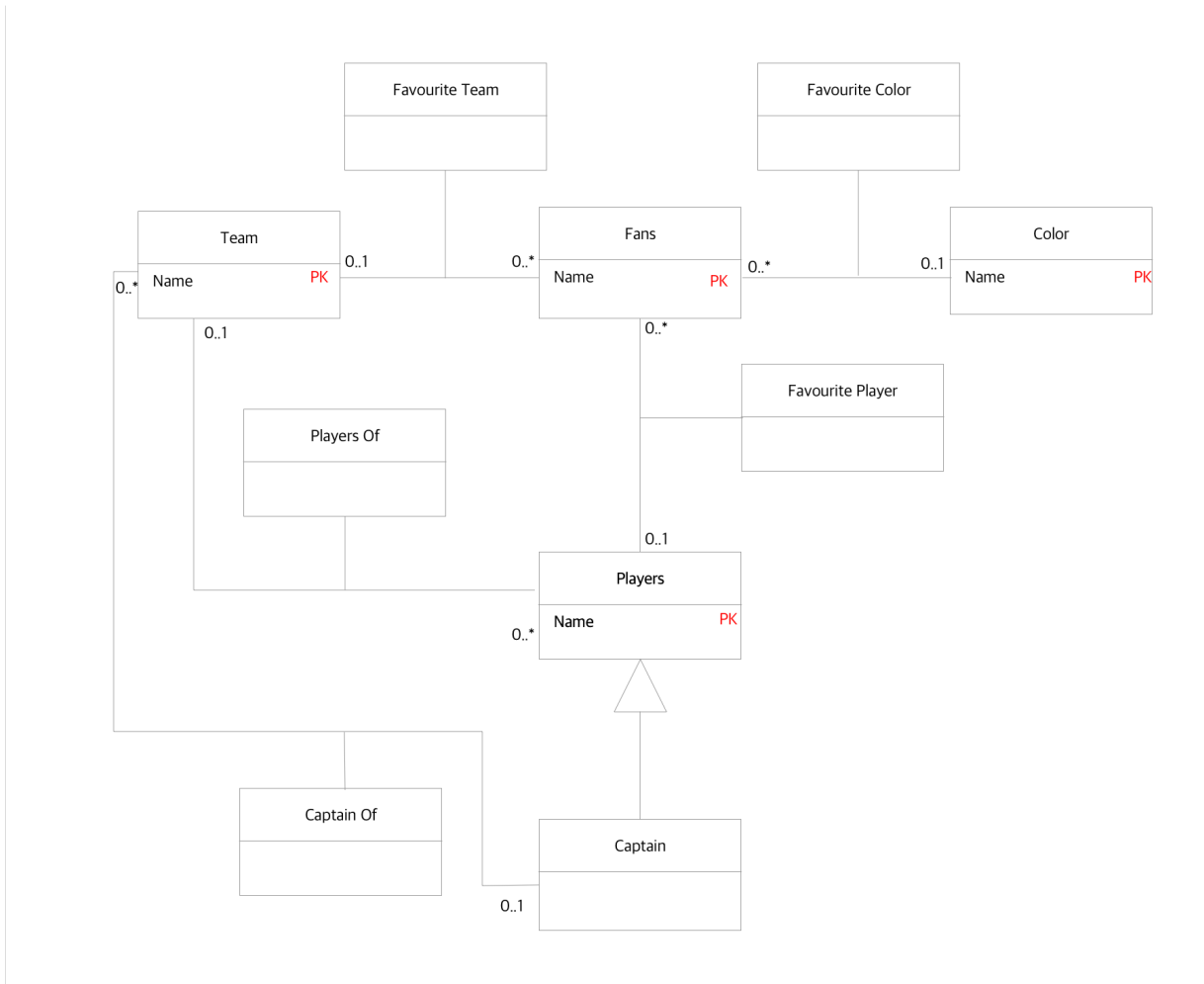
**Correct Solution:**

- E/R Diagram

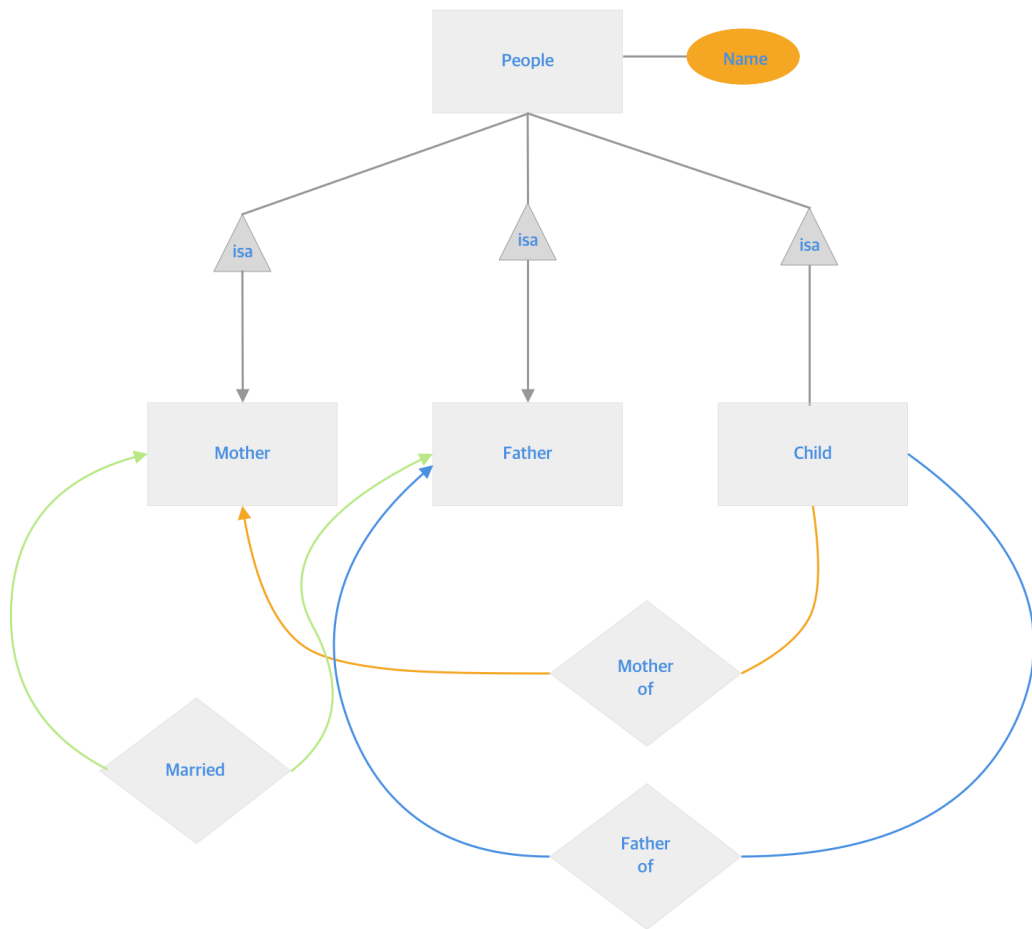




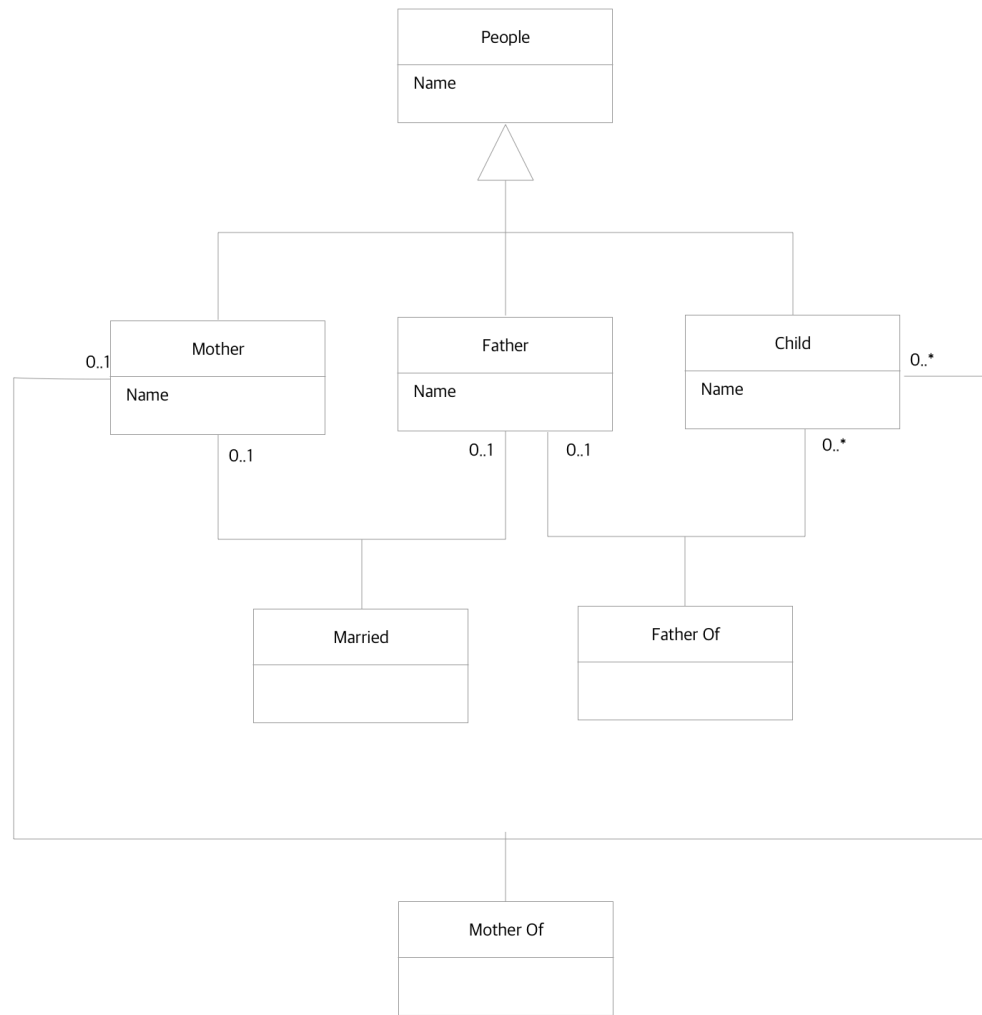
- UML



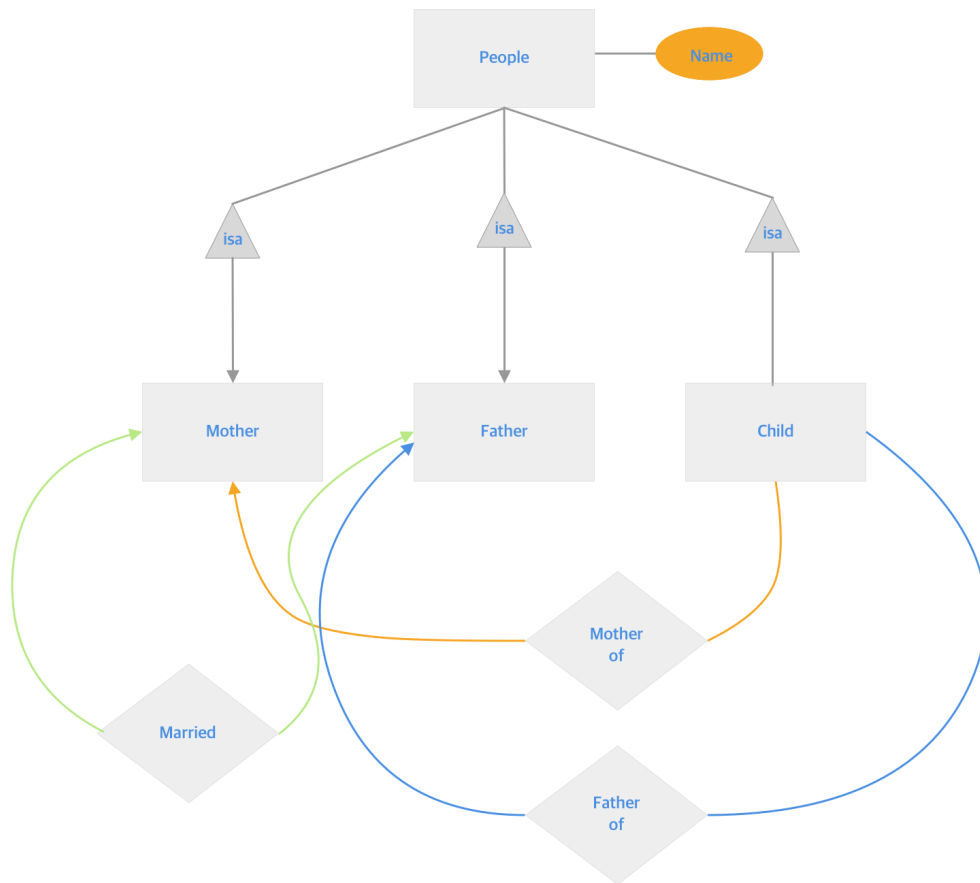
4. • E/R Diagram



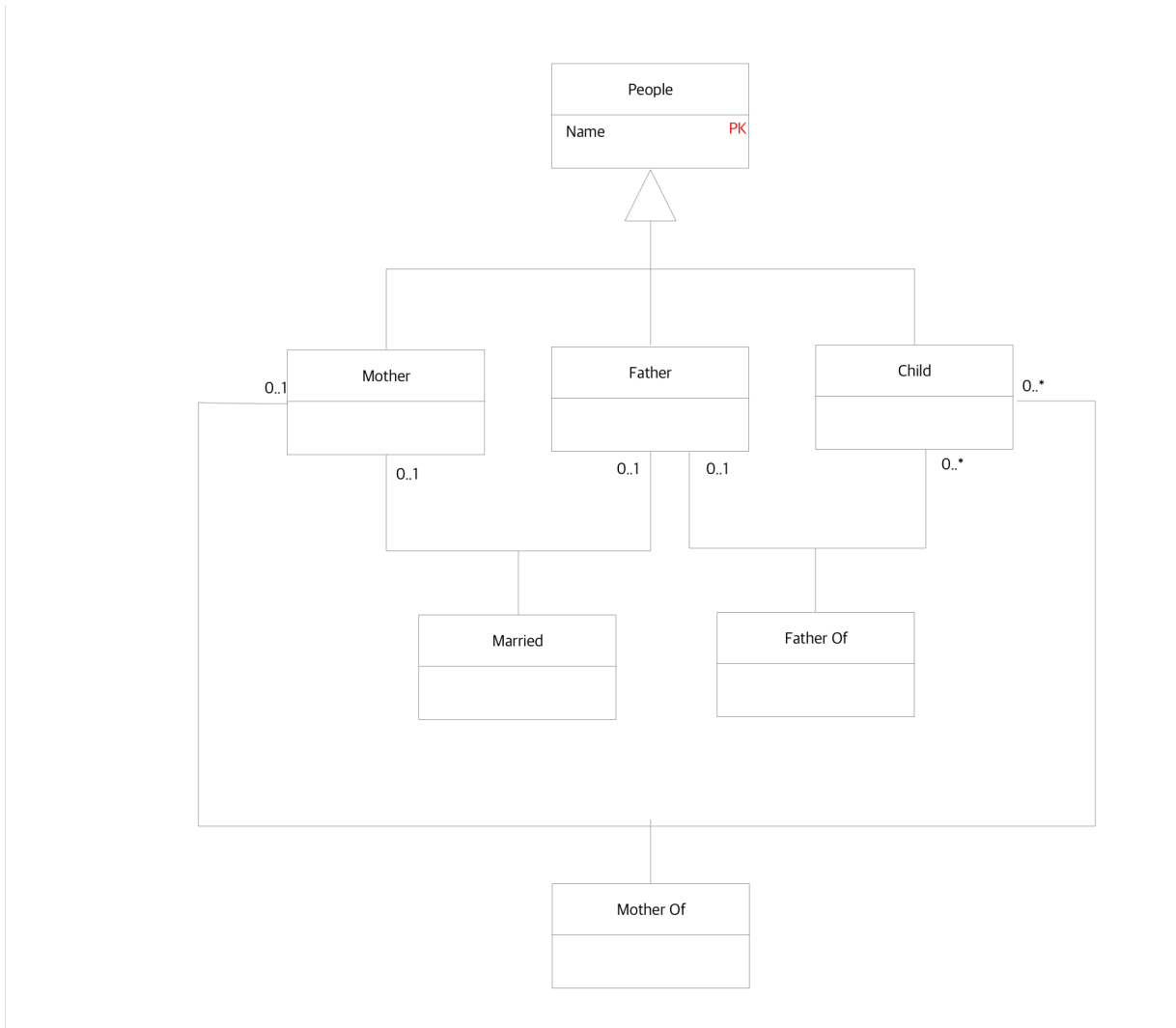
- UML

**Correct Solution:**

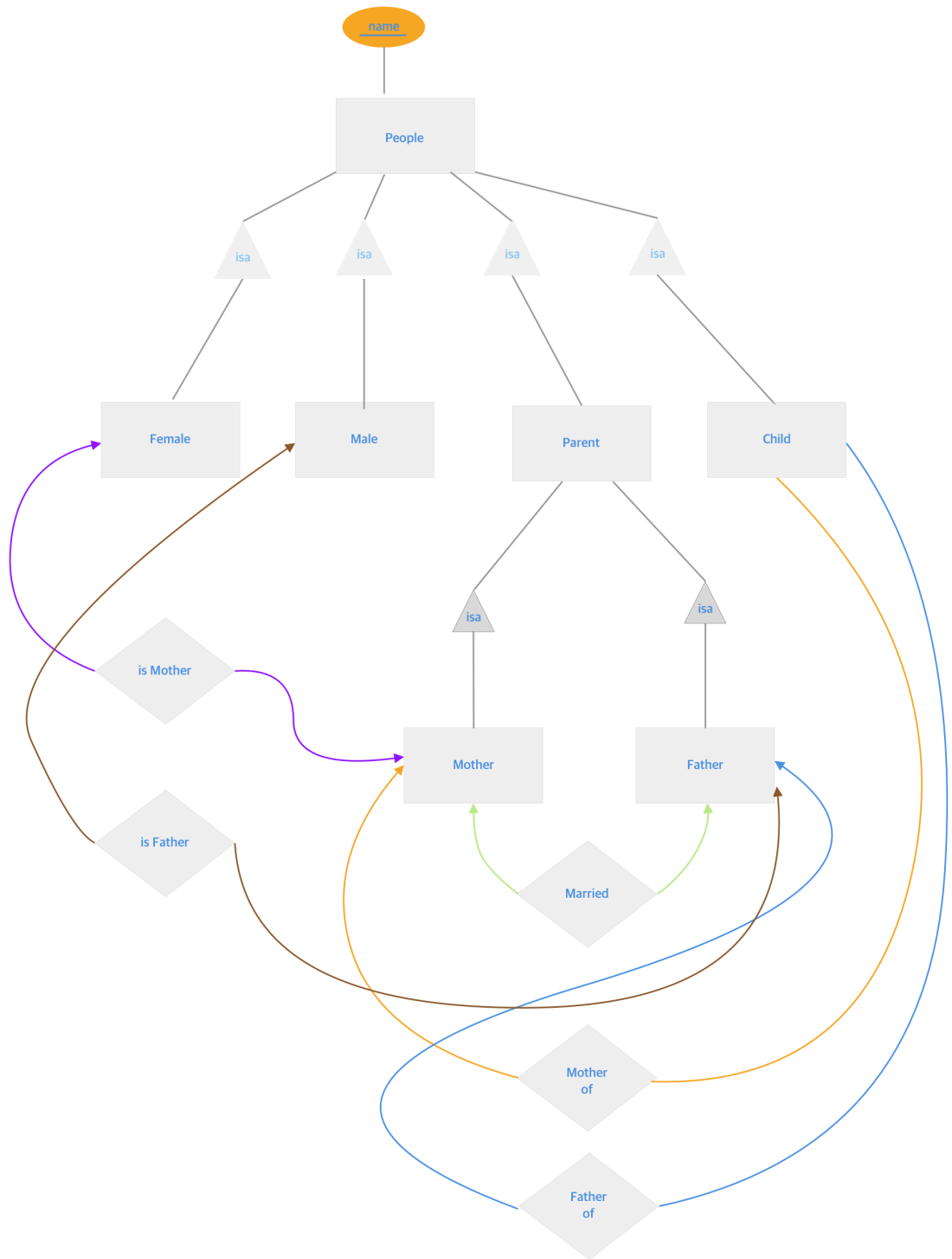
- E/R Diagram



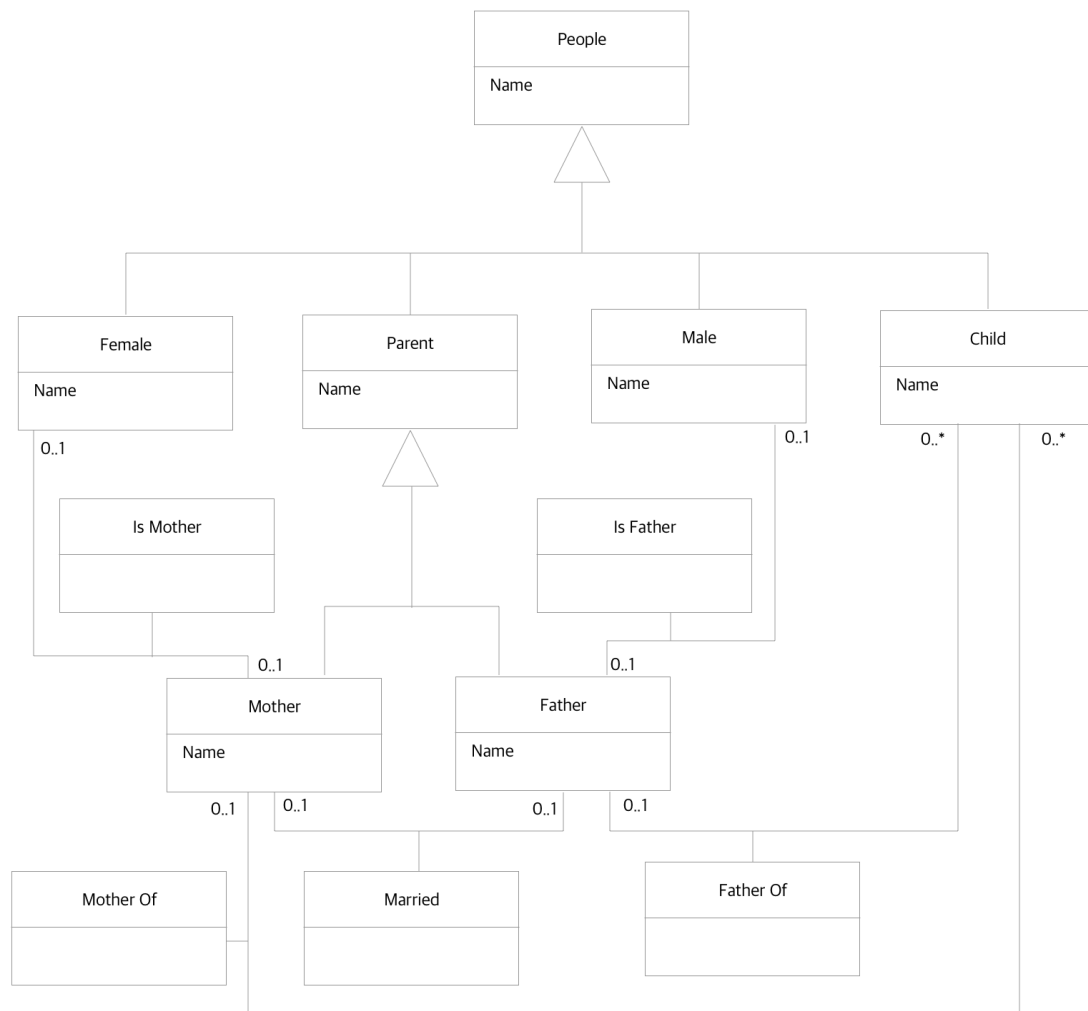
- UML



5. • E/R Diagram

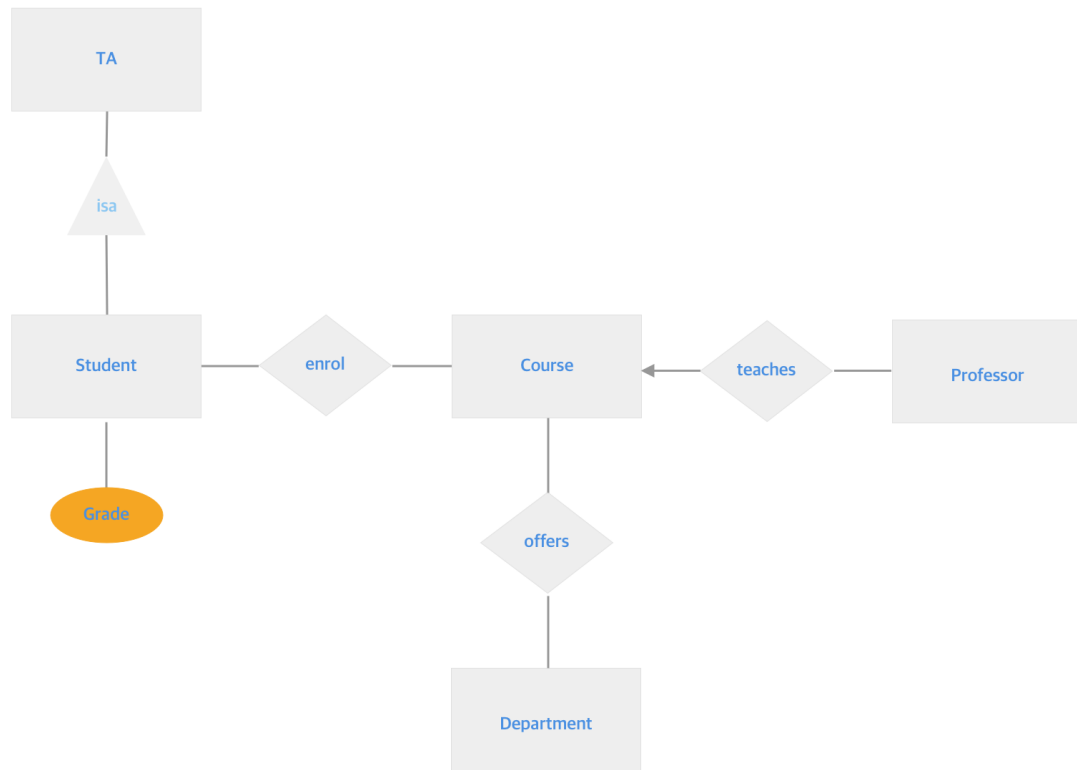


- UML

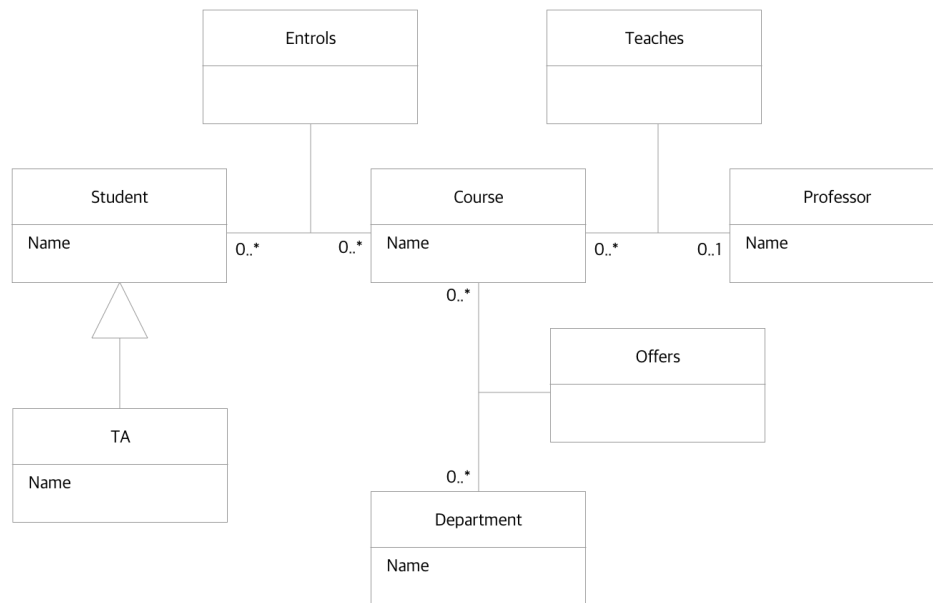


6. • E/R Diagram

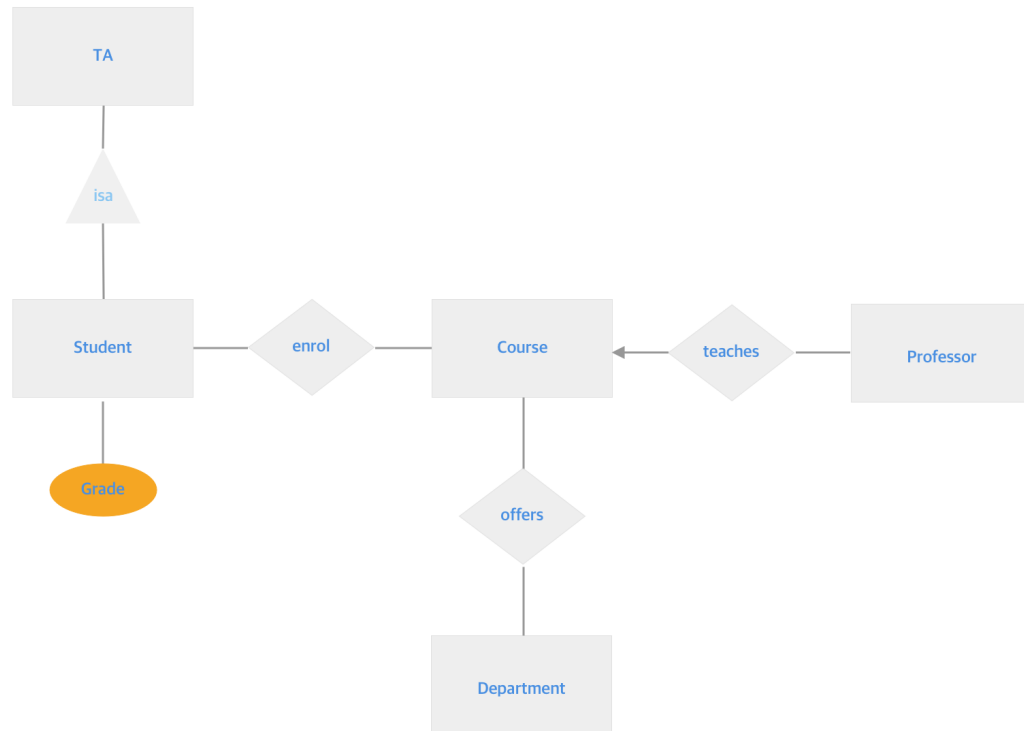




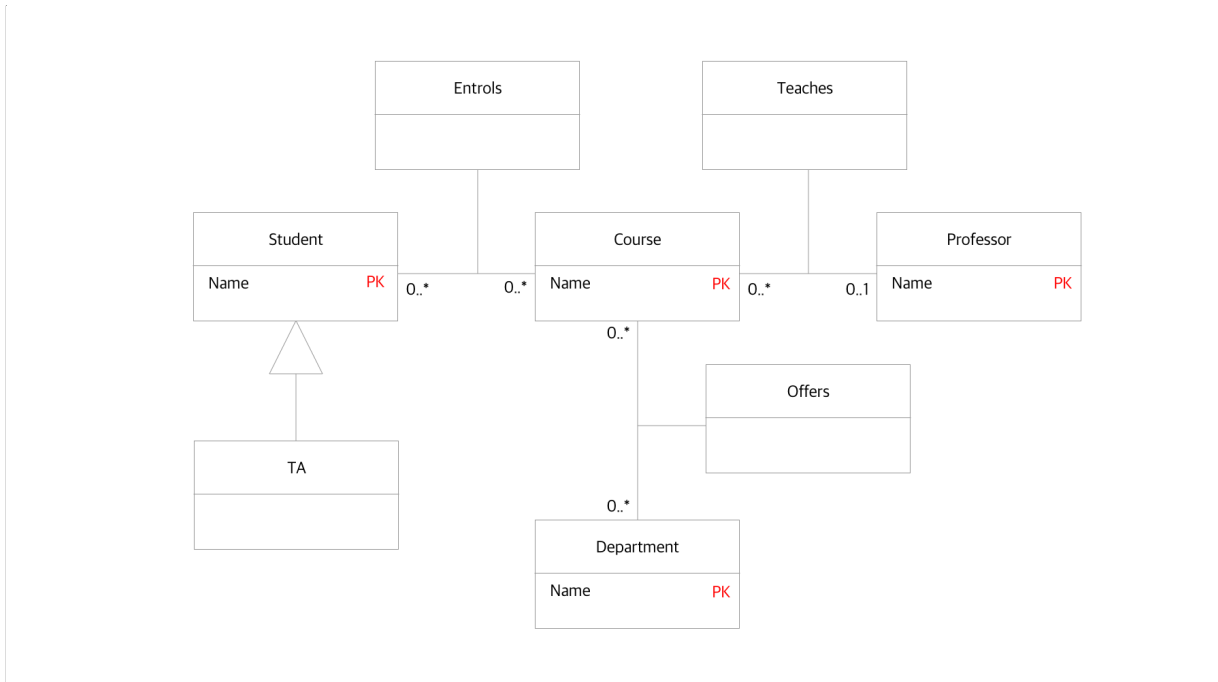
- UML

**Correct Solution:**

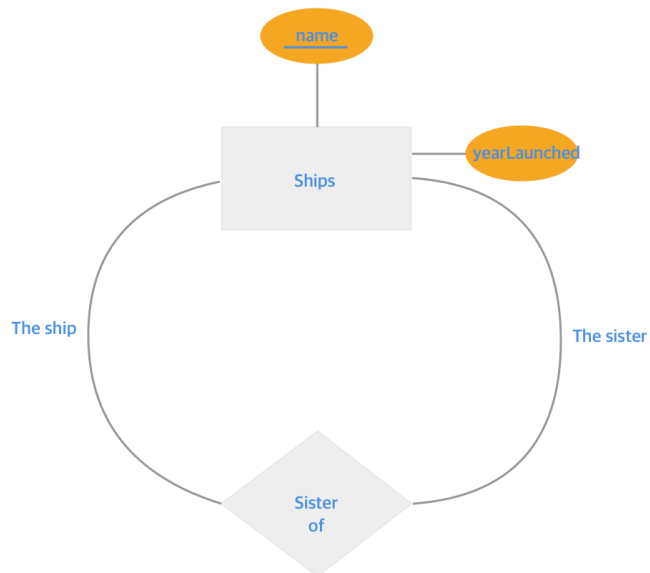
- E/R Diagram



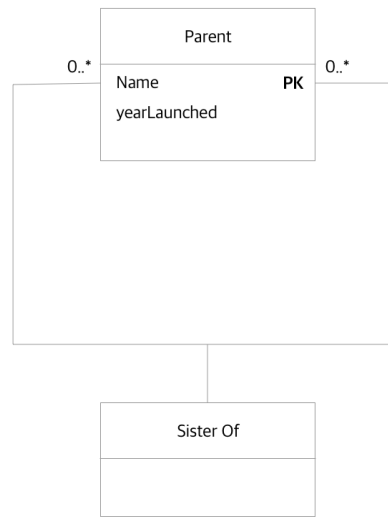
- UML



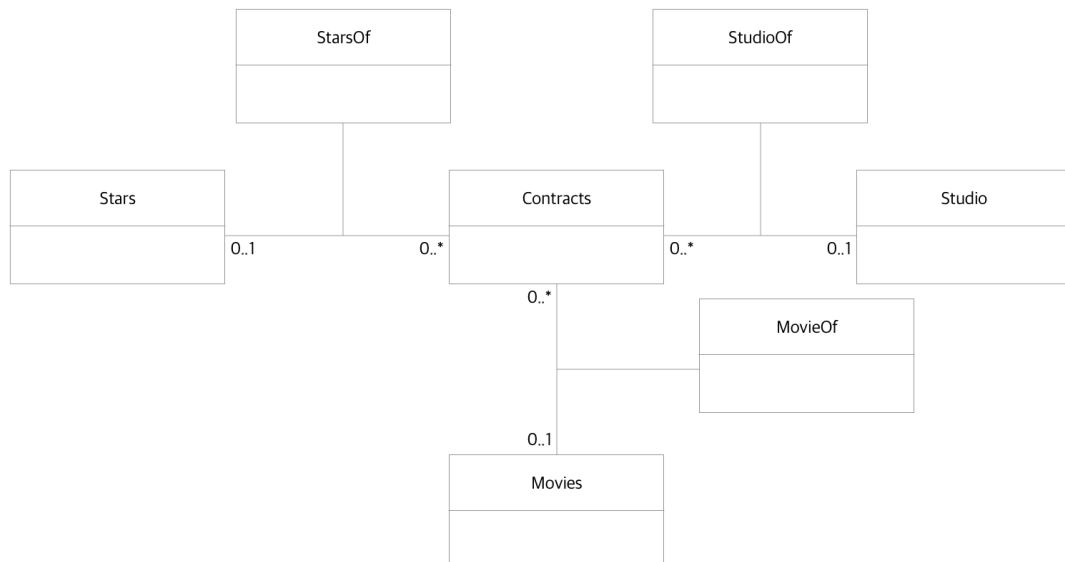
## 7. • E/R Diagram

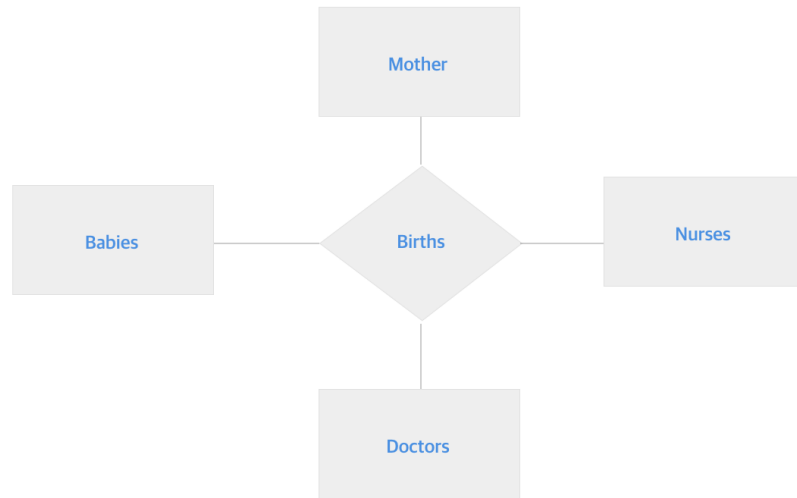


## • UML



8. • UML

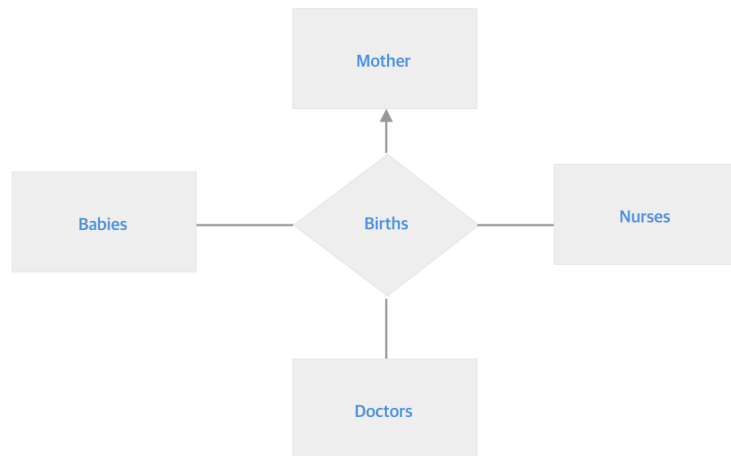




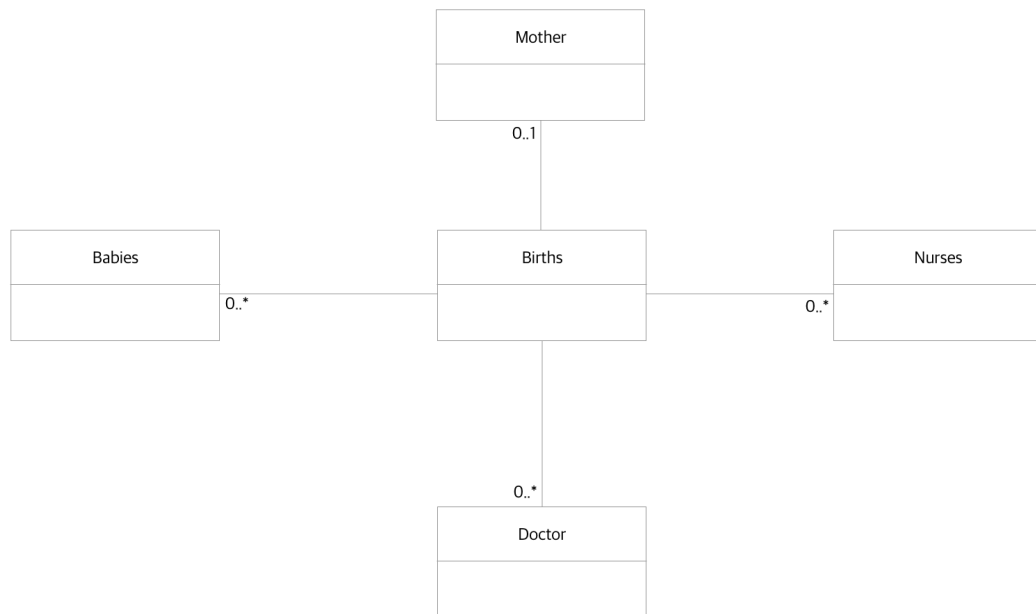
9.

a) **Solution:**

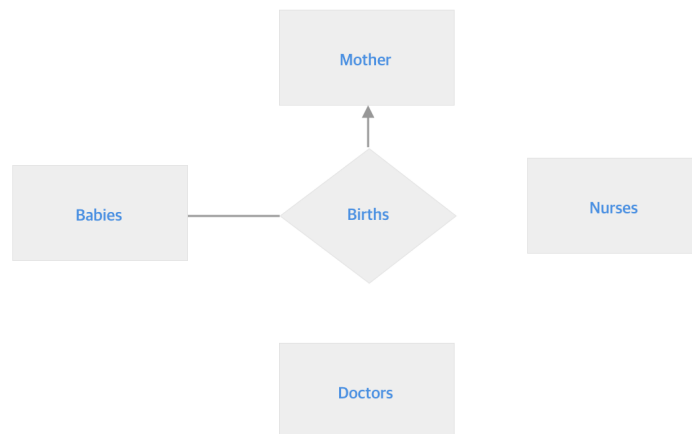
- E/R Model



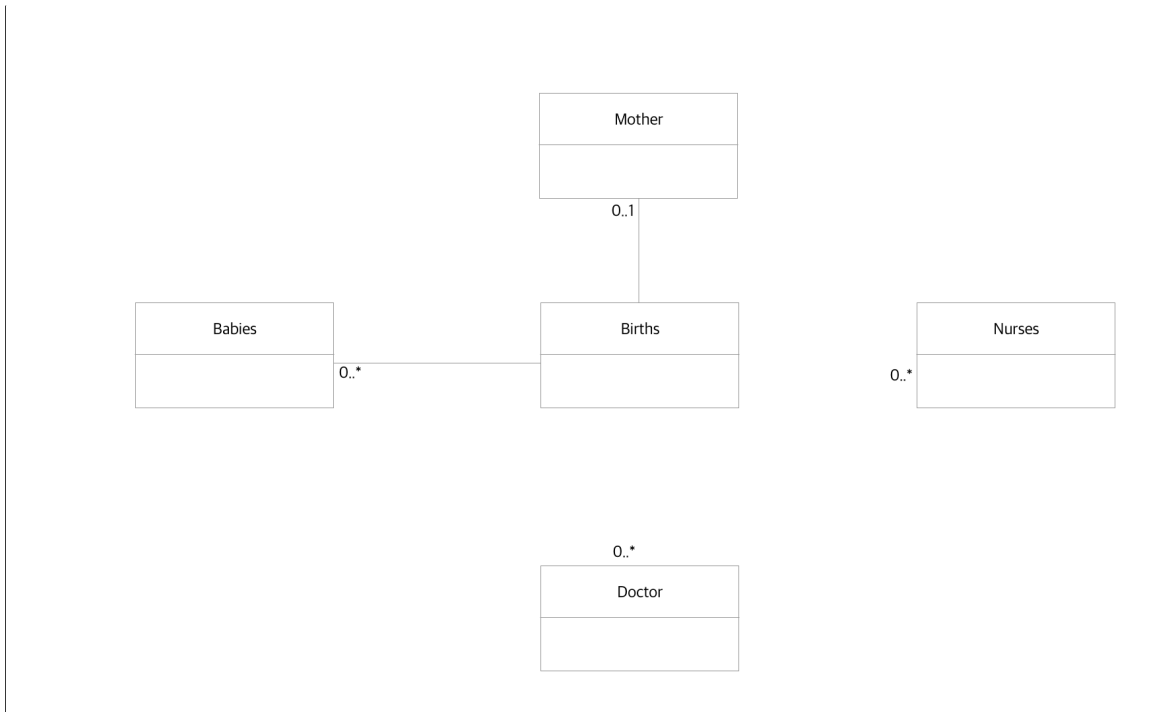
- UML

**Correct Solution:**

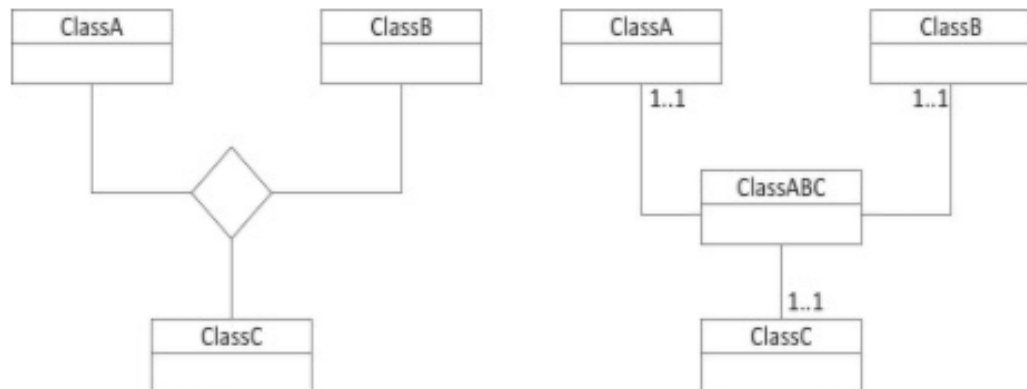
- E/R Model



- UML

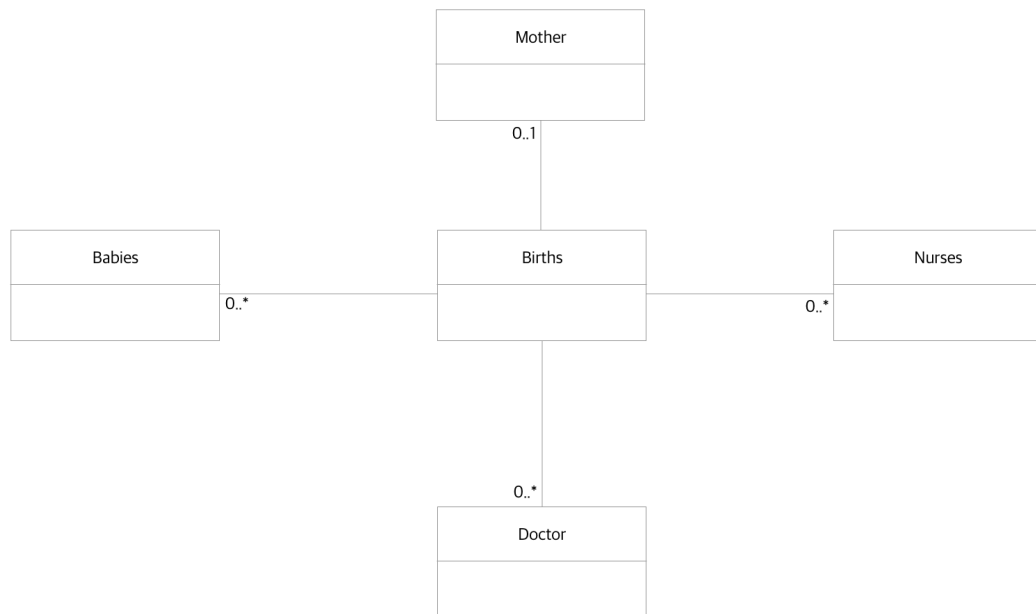
**Notes:**

- An N-ary association is equivalent to one “central” class and N binary associations connecting the central class to the participant classes of the N-ary association

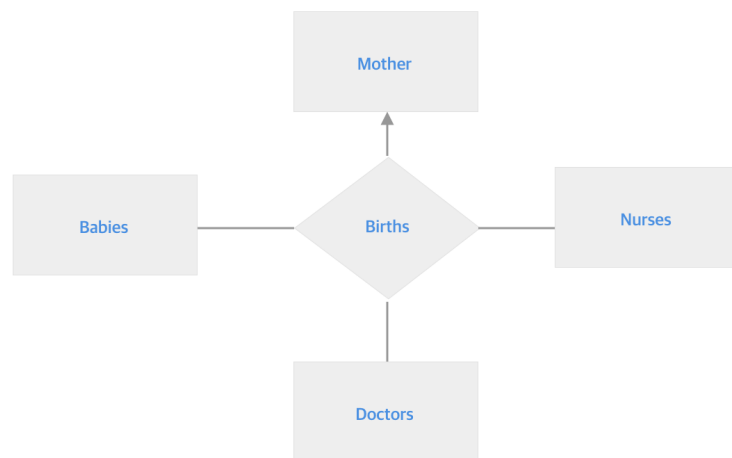
**b) Solution:**

- E/R Model



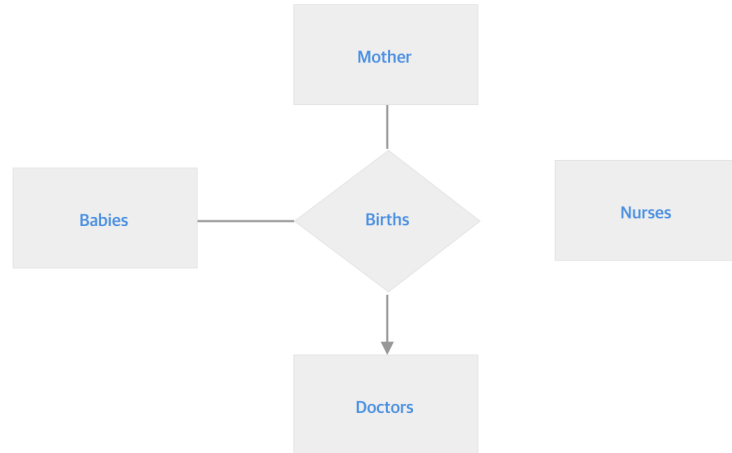


- UML

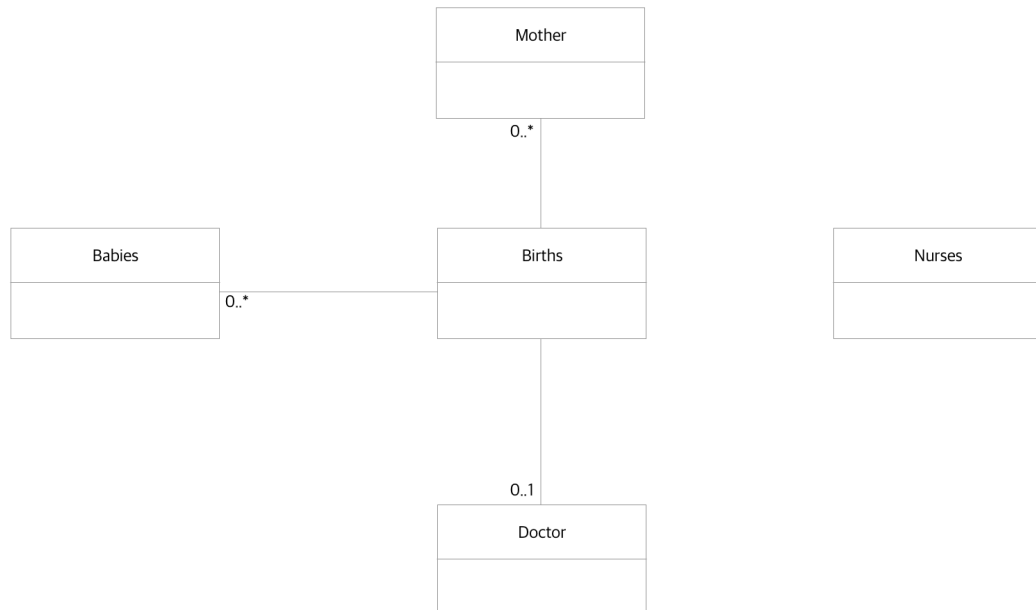


c) **Solution:**

- E/R Model



- UML



10. 

- Bookings

Bookings(SSNo, number, day, row, seat)

- Customers

Customers(SSNo, name, addr, phone)

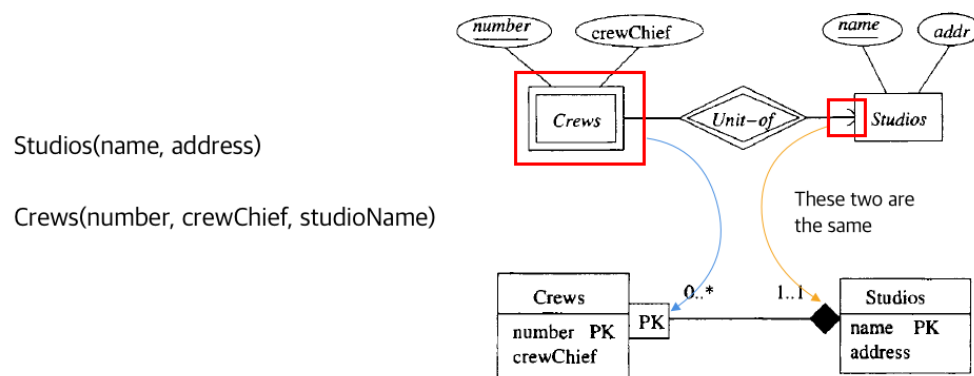
- Flights

Flights(number, day, aircraft)

### Notes:

- Weak Entity sets in UML

### Example:



11. a) • Movies

Movies(title, year, length, genre)

- Studios

Studios(name, address)

- Presidents

Presidents (cert#, name, address)

- Owns

Owns(title, year, name)

- Runs

Runs (cert#, name)

- b) • Movies

Movies(title, year, length, genre)

- Muder-Mysteries

Muder-Mysteries(title, year, weapon)

- Cartoons

Cartoontitle, year)

- Cartoon-Murder-Mysteries

Cartoon-Murder-Mysteries(title, year, weapon)

- Voices (for Cartoons and Cartoon-Murder-Mysteries)

Voices (starName, title, year)

- c) • Customer

Customer(Phone, Address, SSN, Name)

- Account

Account(Number, types, Balance)

- Account Record

AccountRecord(Phone, Address, SSN, Number)

- d) • Team

Team(Name)

- Fans

Fans(Name)

- Color(Name)

Color(Name)

- Players

Players(Name)

- Captain

Captain(Name)

- CaptainOf

CaptainOf(CaptainName, TeamName)

- PlayersOf

PlayersOf(PlayerName, TeamName)

- FavouriteTeam

FavouriteTeam(FanName, TeamName)

- FavouriteColor

FavouriteColor(FanName, ColorName)

- FavouritePlayer

FavouritePlayer(FanName, PlayerName)

- e)
  - People

People(Name)

- Mother

Mother(Name)

- Father

Father(Name)

- Child

Child(Name)

- MotherOf

MotherOf(MotherName, ChildName)

- FatherOf

FatherOf(MotherName, ChildName)

- Married

Married(MotherName, FatherName)

- f)
  - People

People(Name)

- Mother

Mother(Name)

- Father

Father(Name)

- Child

Child(Name)

- MotherOf

MotherOf(MotherName, ChildName)

- FatherOf

FatherOf(MotherName, ChildName)

- Married

Married(MotherName, FatherName)

- g)
  - Student

Student(Name)

- Course

Course(Name)

- Professor

Professor(Name)

- TA

TA(Name)

- Department

Department(Name)

- Enrols

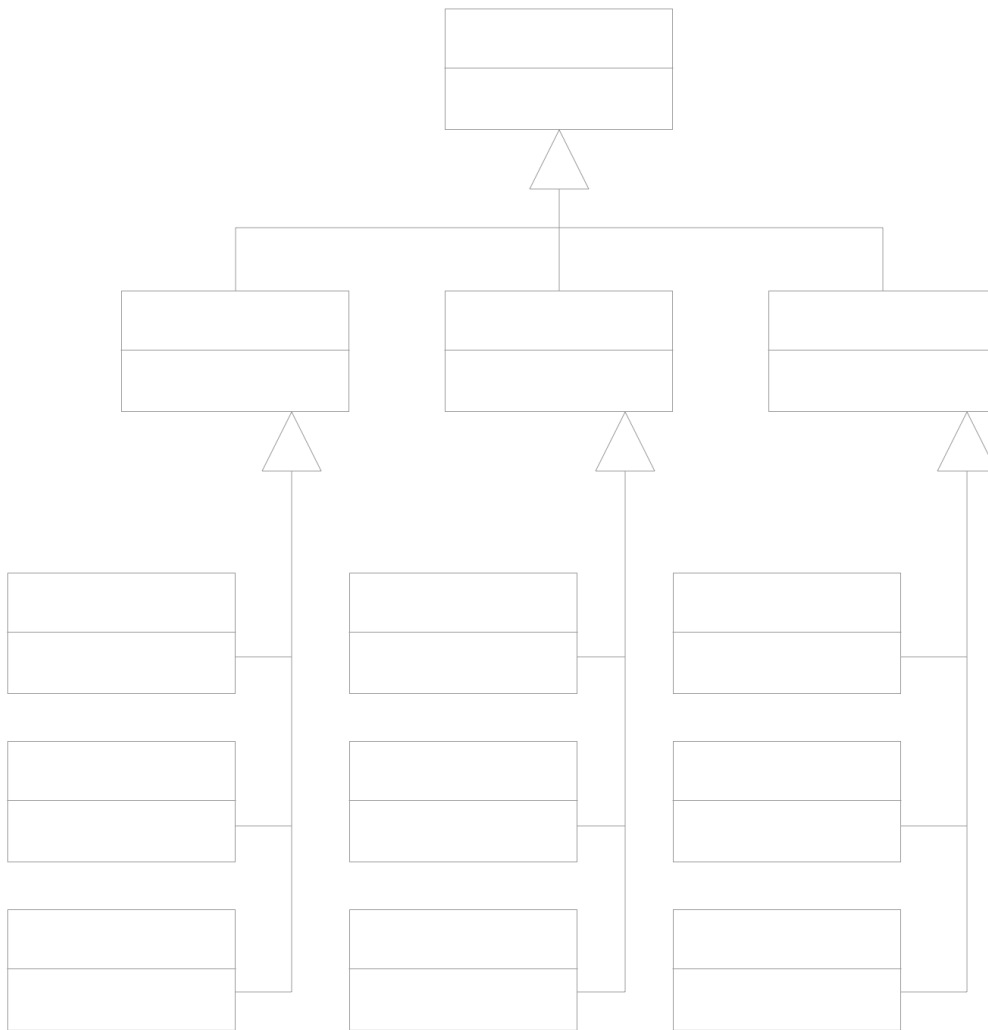
Enrols(StudentName, CourseName)

- Teaches

Teaches(ProfessorName, CourseName)

- Offers

Offers(DepartmentName, CourseName)



12.

a) Total of 13 relations need to be created

**Correct Solution:**

Total of 9 relations need to be created

**Notes:**



- Complete
  - Union of subclasses equals the superclass (parent)
- Disjoint
  - None of the tuples in subclasses are overlapping

**Example:**

The following example is both disjoint and complete :).

