

COMP 3311

DATABASE MANAGEMENT

SYSTEMS

LECTURE 2 EXERCISES

ENTITY-RELATIONSHIP (E-R) MODEL

AND DATA BASE DESIGN

EXERCISE 1: UNIVERSITY APPLICATION

We want to record information about students, departments, courses and course teaching teams.

- For each student we store the student id, name and majors.
- For each department we store a unique code and name.
- For each course we store a unique course id, name, department and prerequisites.
- For each offering of a course we store the section, semester and year.
- Each student must enroll in one to five course offerings.
- Each course offering can enroll zero to sixty students.
- For each course offering that a student takes we store the grade.
- Each course offering's teaching team has one or more staff, who is either an instructor or a TA.
- For each staff assigned to a course offering's teaching team we store the hkid, name, department and office number.
- For each instructor we store their academic title (e.g., professor).

Construct an E-R diagram for the university application.

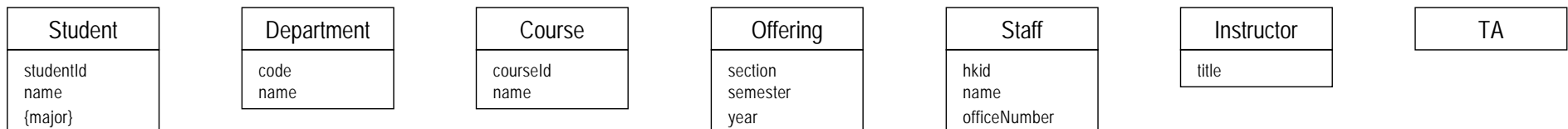
EXERCISE 1: UNIVERSITY APPLICATION— ENTITIES

- For each **student** we store the student id, name and majors.
- For each **department** we store a unique code and name.
- For each **course** we store a unique course id, name, department and prerequisites.
- For each **offering** of a course we store the section, semester and year.
- Each student must enroll in one to five course offerings.
- Each course offering can enroll zero to sixty students.
- For each course offering that a student takes we store the grade.
- Each course offering's teaching team has one or more **staff**, who is either an **instructor** or a **TA**.
- For each staff assigned to a course offering's teaching team we store the hkid, name, department and office number.
- For each instructor we store their academic title (e.g., professor).



EXERCISE 1: UNIVERSITY APPLICATION— ENTITY ATTRIBUTES

- For each **student** we store the **student id**, **name** and **majors**.
- For each **department** we store a unique **code** and **name**.
- For each **course** we store a unique **course id**, **name**, department and prerequisites.
- For each **offering** of a course we store the **section**, **semester** and **year**.
- Each student must enroll in one to five course offerings.
- Each course offering can enroll zero to sixty students.
- For each course offering that a student takes we store the grade.
- Each course offering's teaching team has one or more staff, who is either an instructor or a TA.
- For each **staff** assigned to a course offering's teaching team we store the **hkid**, **name**, department and **office number**.
- For each **instructor** we store their academic **title** (e.g., professor).

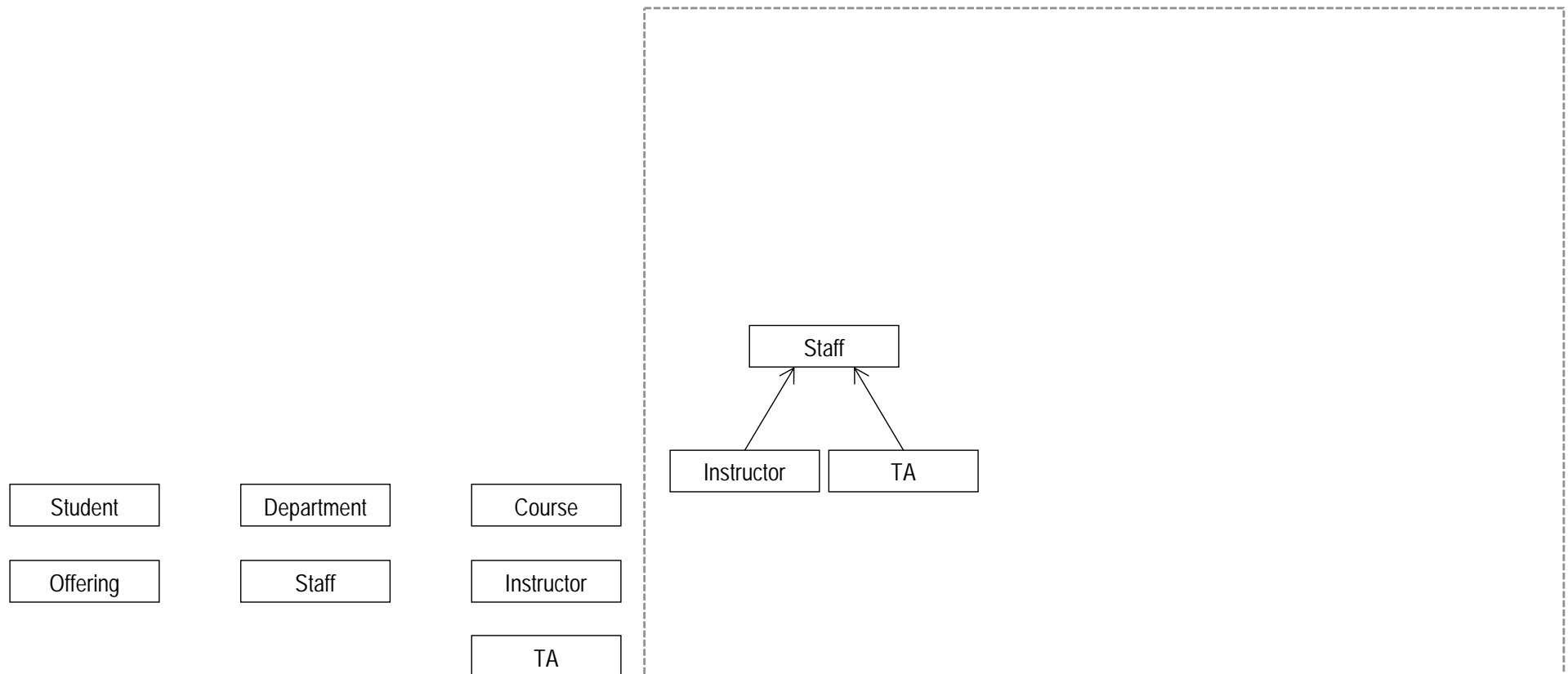


EXERCISE 1: UNIVERSITY APPLICATION— ENTITY GENERALIZATION

- Each course offering's teaching team has one or more **staff**, who is either an instructor or a TA.

What should be the generalization?

⇒ Staff superclass; Instructor, TA subclasses.

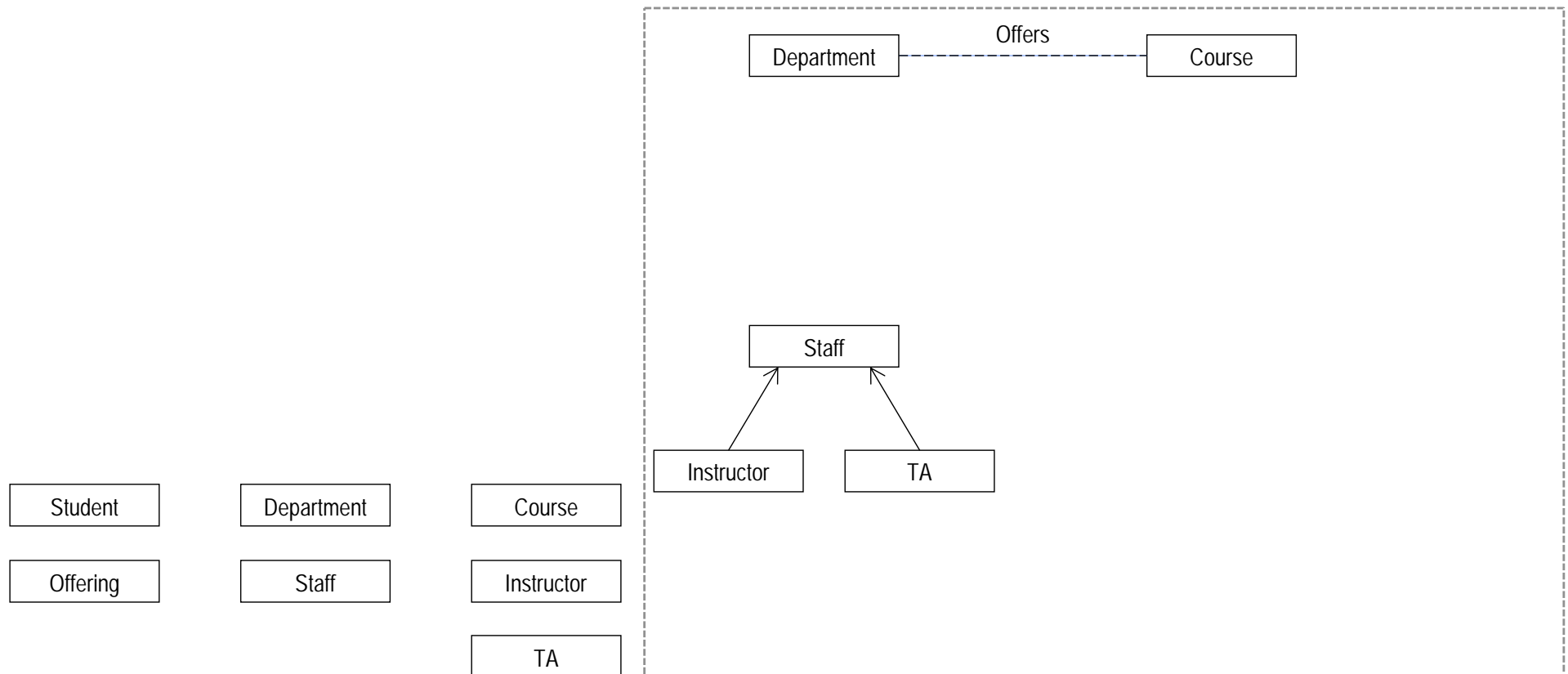


EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIPS

- For each **course** we store a unique course id, name, **department** and prerequisites.

What should be related?

⇒ Course related to Department.

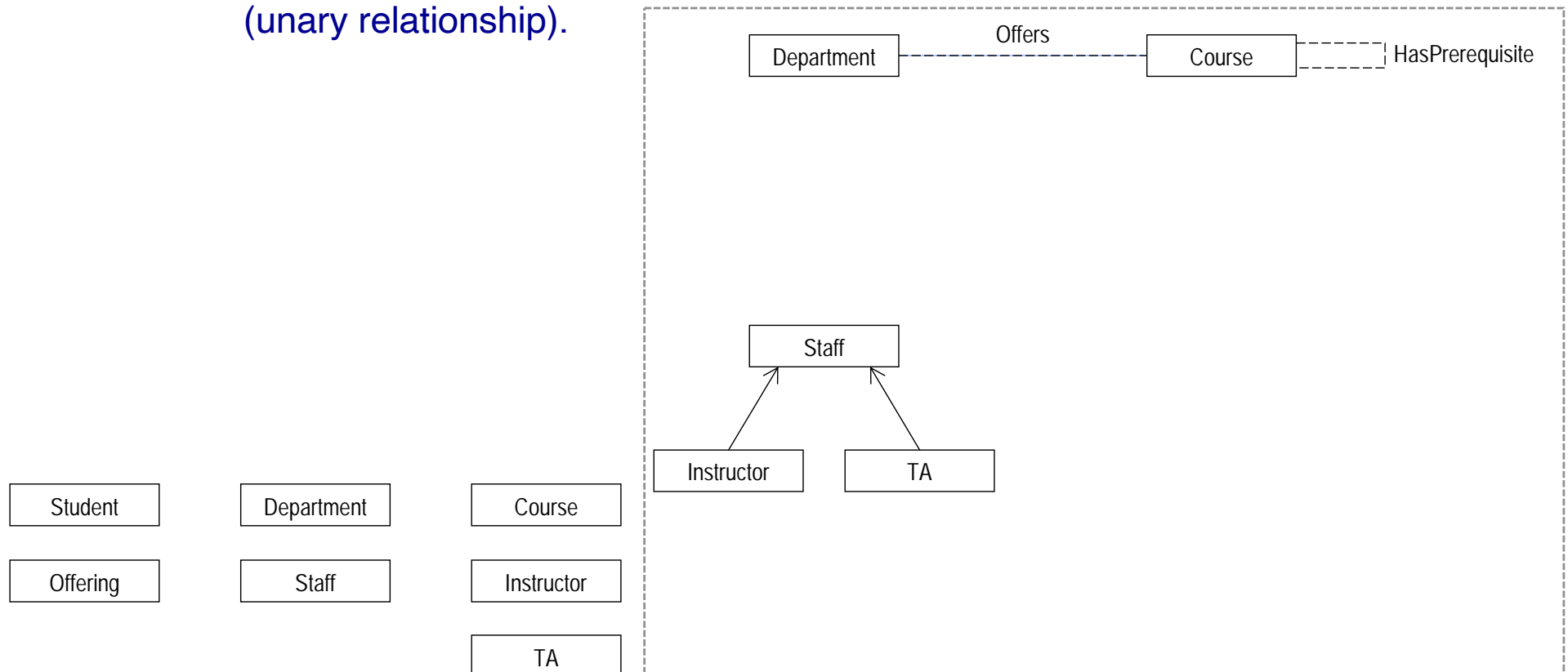


EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIPS

- For each **course** we store a unique course id, name, department and prerequisites.

What should be related?

⇒ Course related to Course
(unary relationship).

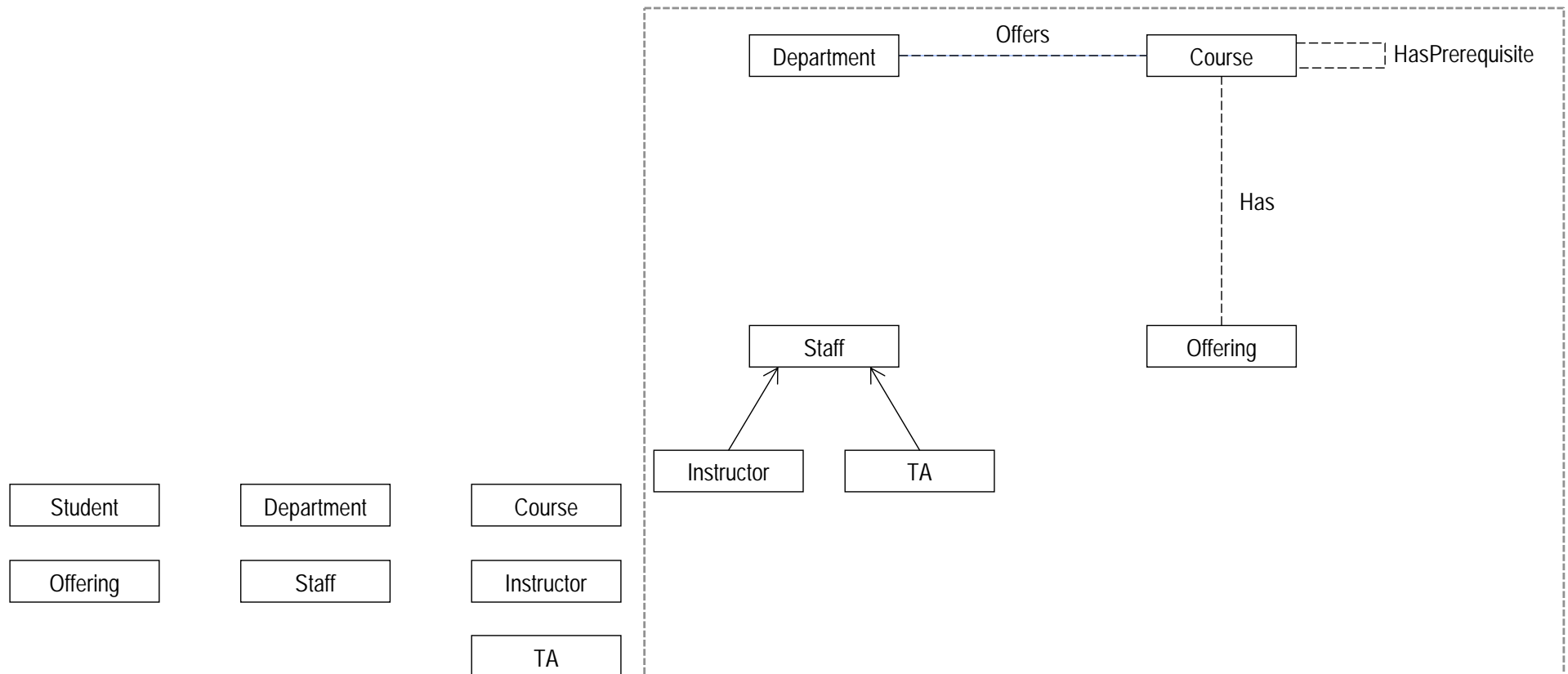


EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIPS

- For each **offering** of a **course** we store the section, semester and year.

What should be related?

⇒ Offering related to Course.

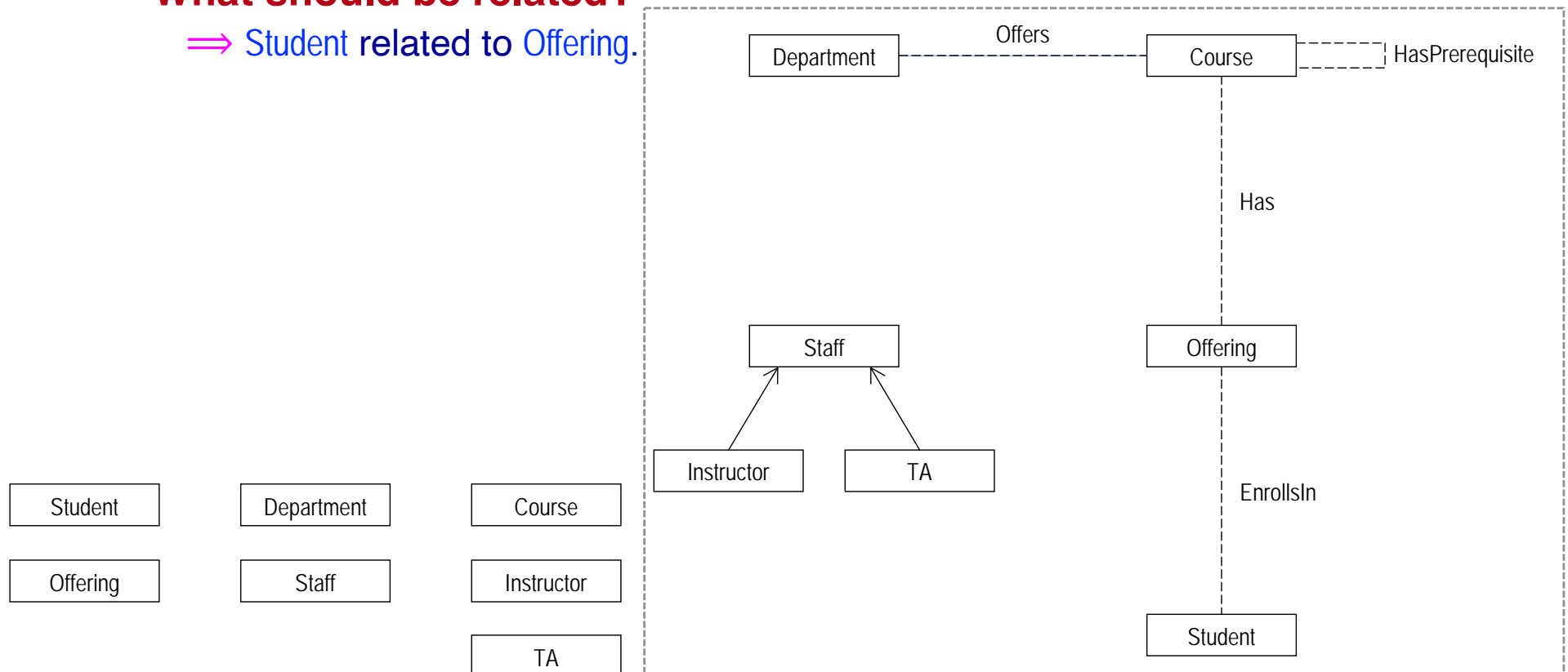


EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIPS

- Each **student** must enroll in one to five course **offerings**.
- Each course offering can enroll zero to sixty students.
- For each course that a student takes we store the grade.

What should be related?

⇒ Student related to Offering.

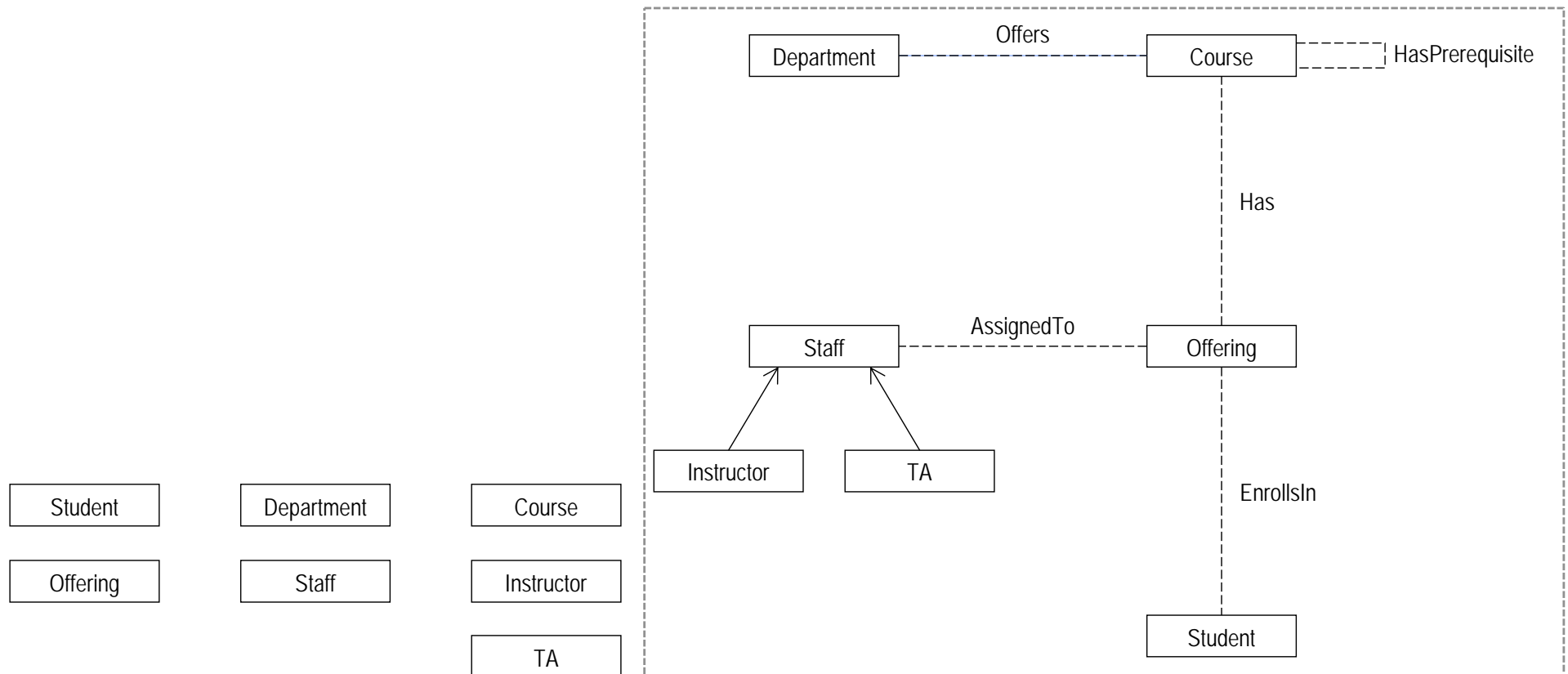


EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIPS

- For each **staff** assigned to a course **offering**'s teaching team we store the hkid, name, department and office number.

What should be related?

⇒ Staff related to Offering.

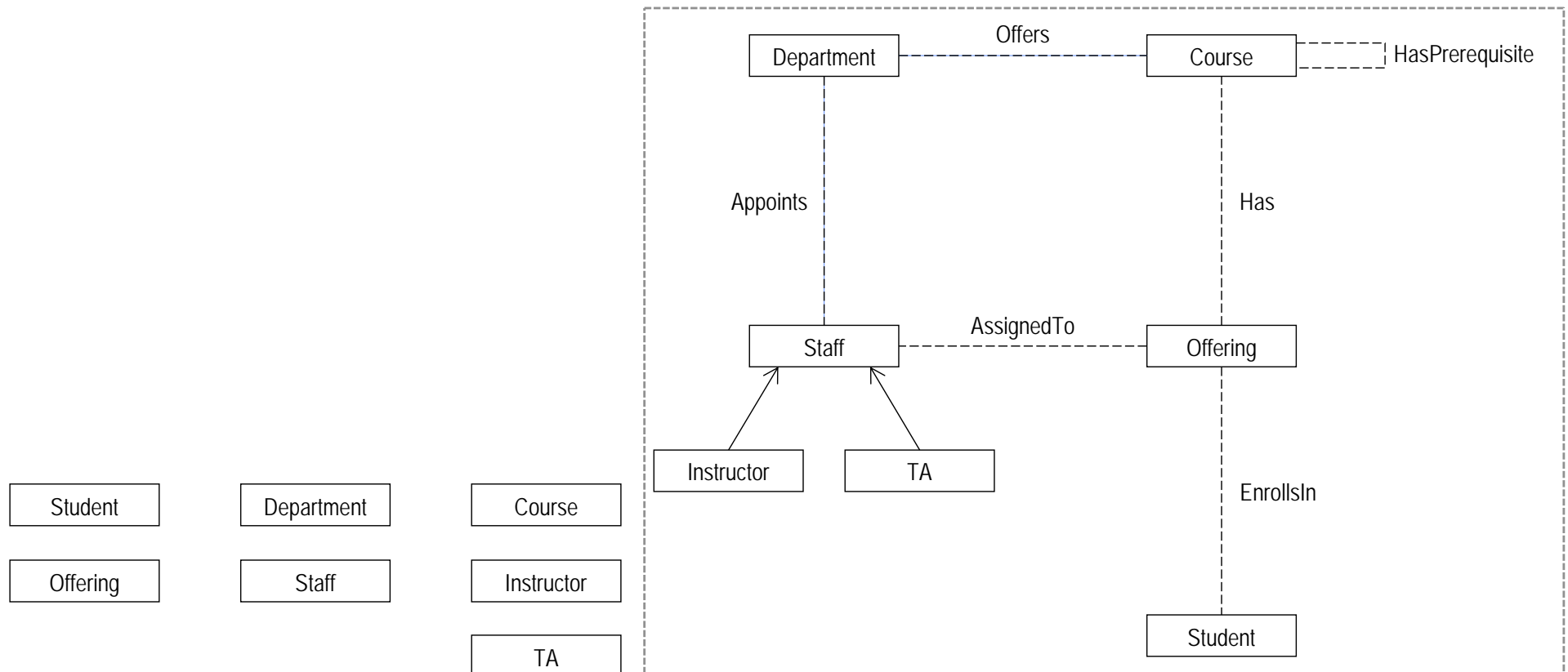


EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIPS

- For each **staff** assigned to a course offering's teaching team we store the **hkid**, name, **department** and office number.

What should be related?

⇒ Staff related to Department.



EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIP ATTRIBUTES

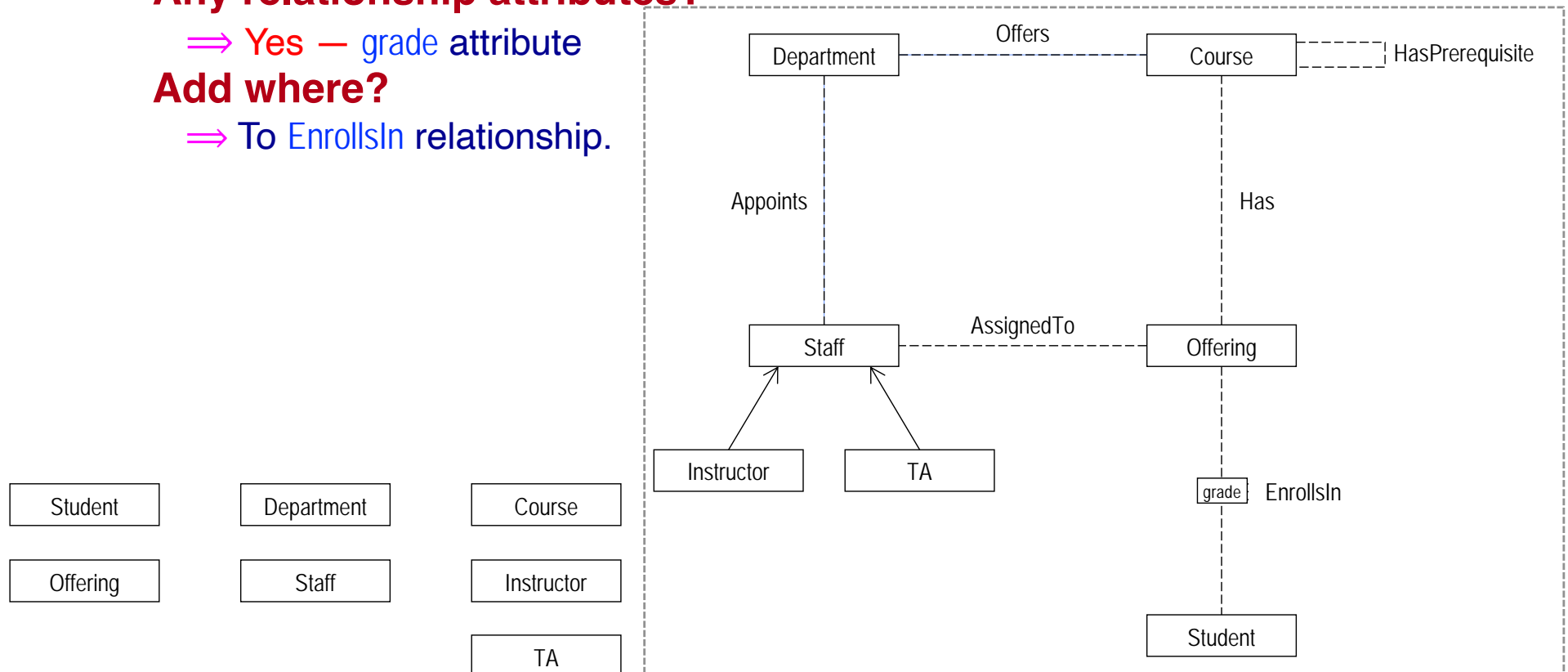
- Each student must enroll in one to five course offerings.
- Each course offering can enroll zero to sixty students.
- For each course that a student takes we store the **grade**.

Any relationship attributes?

⇒ Yes — **grade** attribute

Add where?

⇒ To **EnrollsIn** relationship.

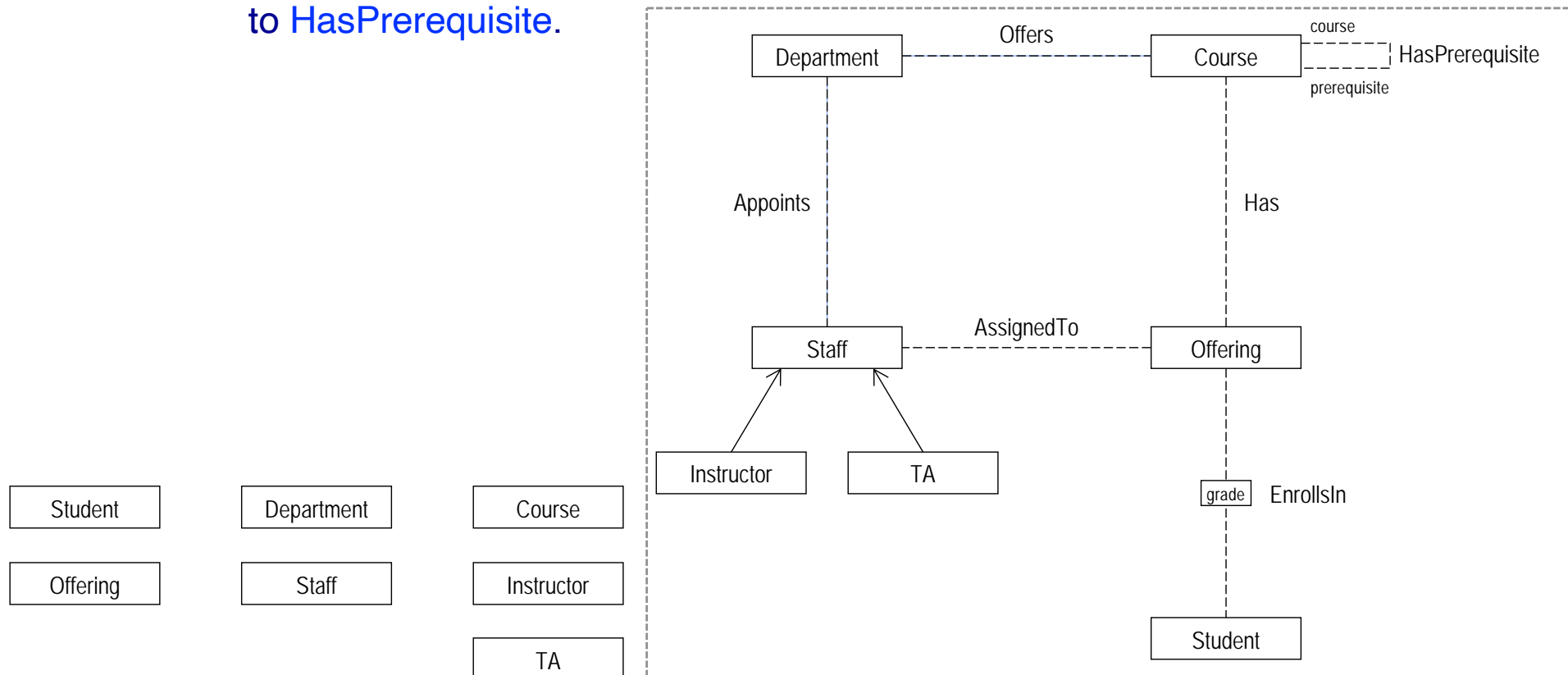


EXERCISE 1: UNIVERSITY APPLICATION— RELATIONSHIP ROLE NAMES

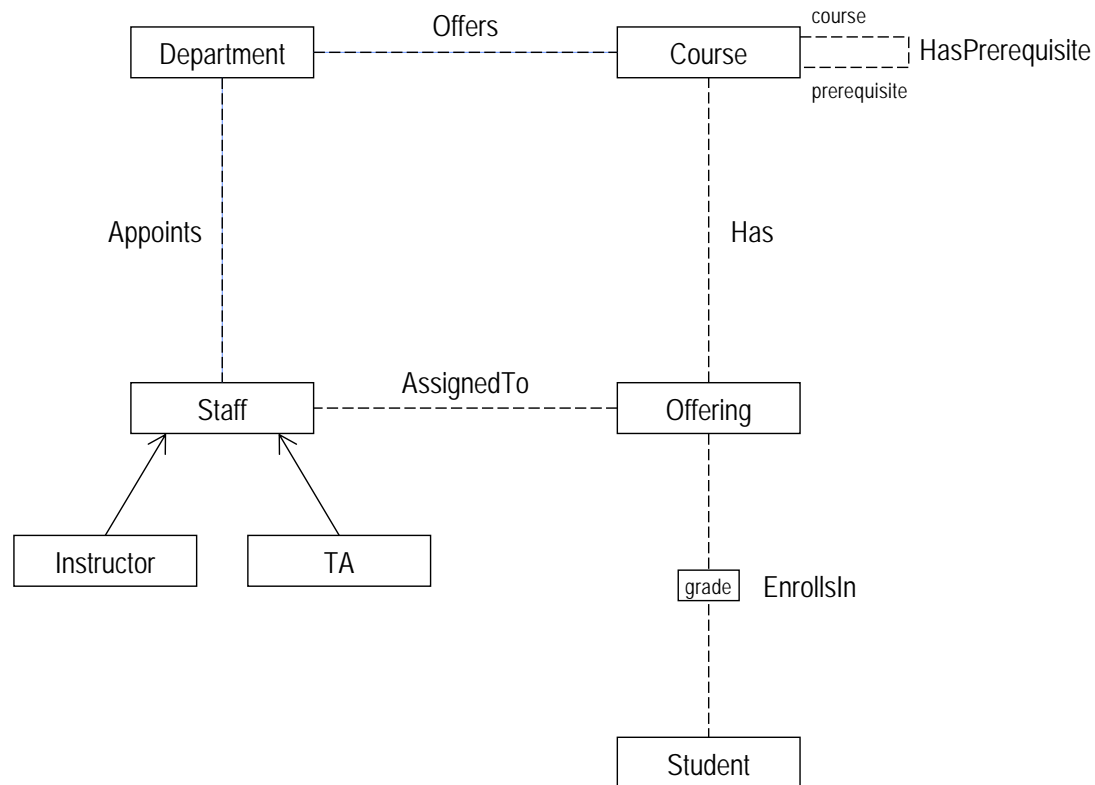
- For each course we store a unique course id, name, department and prerequisites.

Any role names?

⇒ Yes — add role names
to HasPrerequisite.



EXERCISE 1: UNIVERSITY APPLICATION— E-R DIAGRAM



Student
studentId name {major}

Department
code name

Course
courseId name

Offering
section semester year

Staff
hkid name officeNumber

Instructor
title

TA

EXERCISE 2: BUS COMPANY

We want to keep track of bus routes and schedules for a bus company.

- Each bus route has a unique route number, a departure station and a destination station.
- For each bus route, there is a schedule, which records the departure times of buses.
- For each departure time of each route, a driver and a bus can be assigned; however, information about the driver or the bus may sometimes be missing.
- A driver has a unique employee id, a name and a phone number.
- A bus is identified by its license number and has a maximum seating capacity.

Construct an E-R diagram for the bus company application.

EXERCISE 2: BUS COMPANY—ENTITIES

We want to keep track of bus routes and schedules for a bus company.

- Each bus **route** has a unique route number, a departure station and a destination station.
- For each bus route, there is a **schedule**, which records the departure times of buses.
- For each departure time of each route, a **driver** and a **bus** can be assigned; however, information about the driver or the bus may sometimes be missing.
- A driver has a unique employee id, a name and a phone number.
- A bus is identified by its license number and has a maximum seating capacity.

Route

Schedule

Driver

Bus

EXERCISE 2: BUS COMPANY— ATTRIBUTES OF ENTITIES

- Each bus **route** has a unique **route number**, a **departure station** and a **destination station**.
- For each bus route, there is a **schedule**, which records the **departure times** of buses.
- A **driver** has a unique **employee id**, a **name** and a **phone number**.
- A **bus** is identified by its **license number** and has a **maximum seating capacity**.

Route
routeNo departureStation destinationStation

Schedule
departureTime

Driver
empld name phoneNo

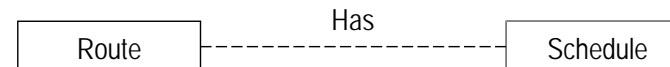
Bus
licenseNo maxSeating

EXERCISE 2: BUS COMPANY— RELATIONSHIPS (ROUTE, SCHEDULE)

- Each bus route has a unique route number, a departure station and a destination station.
- For each bus route, there is a schedule, which records the departure times of buses.

What should be related?

⇒ Route related to Schedule.



EXERCISE 2: BUS COMPANY— RELATIONSHIPS (DRIVER, BUS)

- For each departure time of each route, a driver and a bus can be assigned; however, information about the driver or the bus may sometimes be missing.

What should be related?

⇒ Driver related to Bus.

Driver

Bus

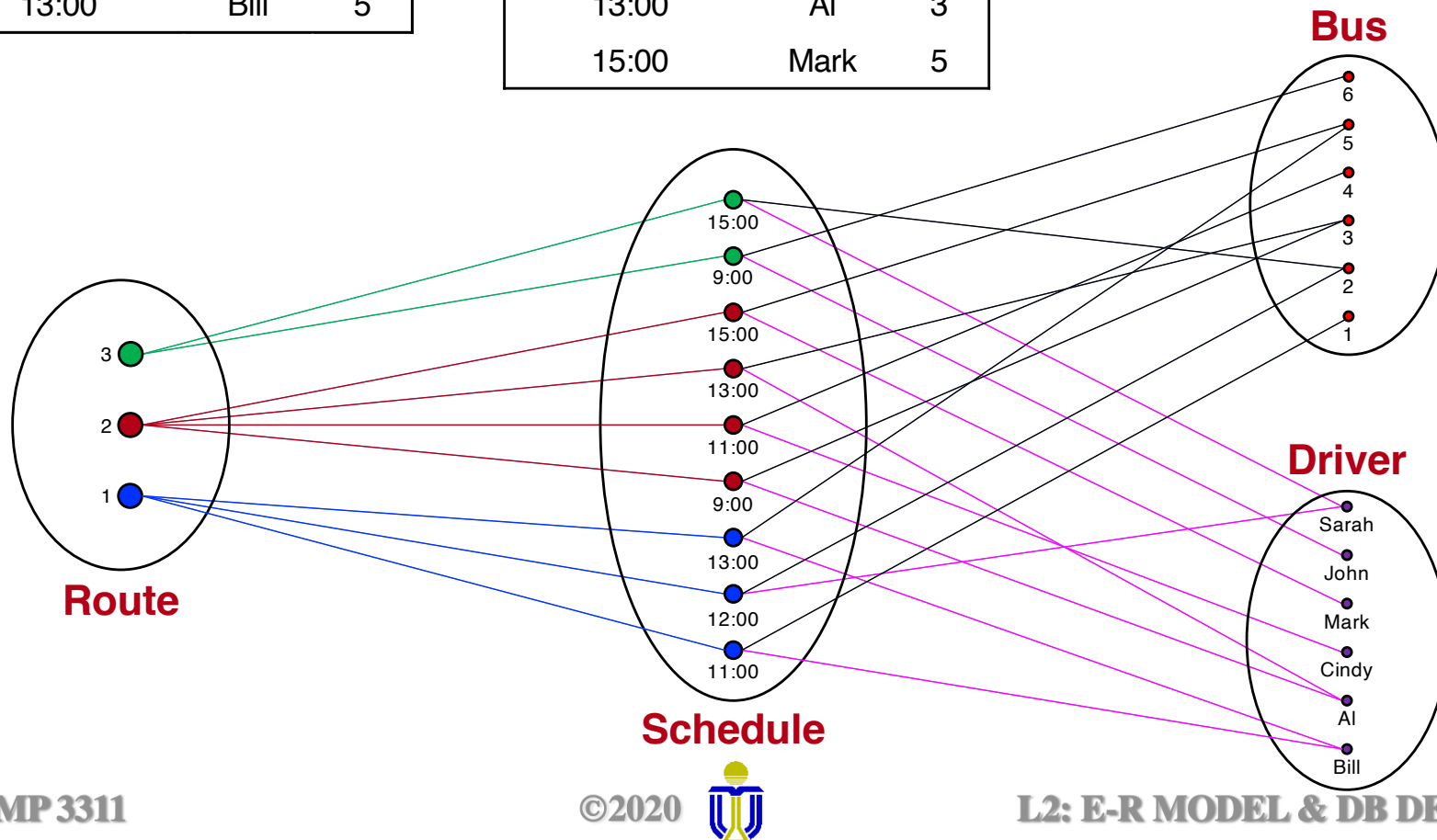
How should they be related?

EXERCISE 2: WHAT IS A SCHEDULE?

Route 1		
Departure time	Driver	Bus
11:00	Bill	1
12:00	Sarah	2
13:00	Bill	5

Route 2		
Departure time	Driver	Bus
9:00	Al	3
11:00	Cindy	4
13:00	Al	3
15:00	Mark	5

Route 3		
Departure time	Driver	Bus
9:00	John	6
15:00	Sarah	2



EXERCISE 2: BUS COMPANY— RELATIONSHIPS (DRIVER, BUS)

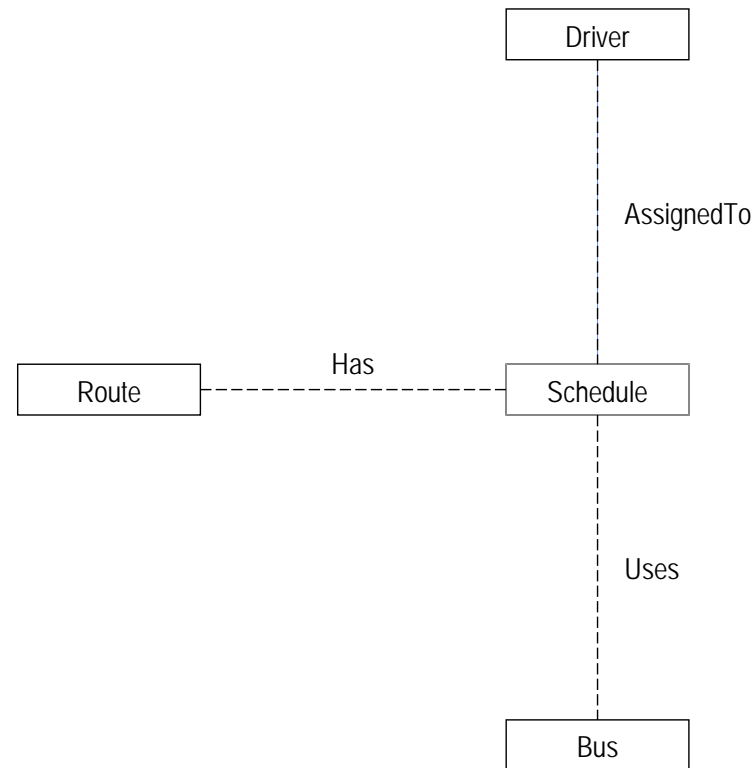
- For each departure time of each route, a driver and a bus can be assigned; however, information about the driver or the bus may sometimes be missing.

What should be related?

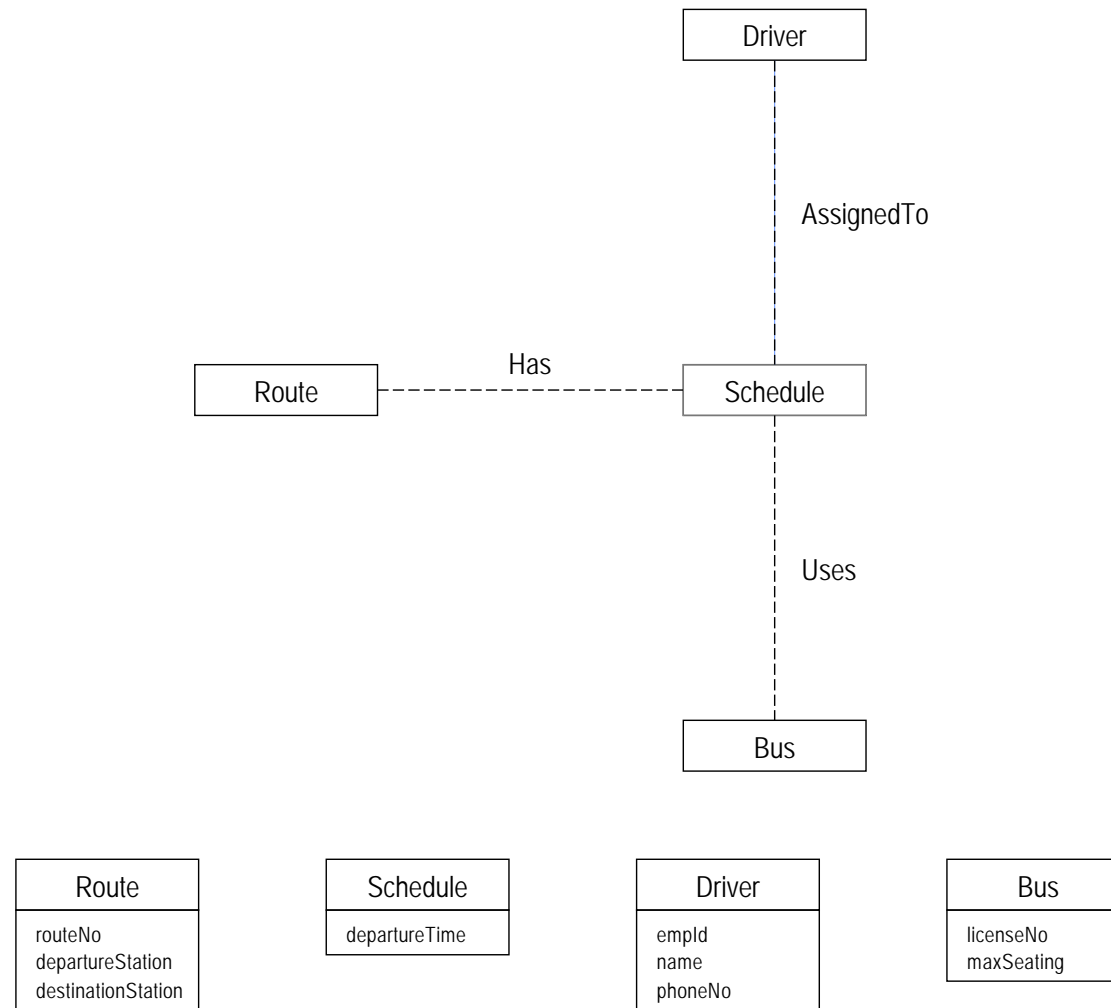
⇒ Driver related to Bus.

How should they be related?

⇒ Through the Schedule entity.



EXERCISE 2: BUS COMPANY—E-R DIAGRAM



EXERCISE 2: BUS COMPANY—E-R DIAGRAM

POSSIBLE REFINEMENT

