

# CS 411: Database Systems

Fall 2018

## Homework 2 (Due by 23:59 CDT on October 15)

### Logistics

1. The homework is due on Oct 15 23:59. **We DO NOT accept late homework submissions.**
2. You will be using Gradescope to submit your solutions. Answer each question “Part” on a new page and submit your solution as a single PDF file on Gradescope. The link to submission on GradeScope will be updated HERE soon.
3. Please write all answers electronically. We won't grade hand written and drawn versions. If you are looking for tools to create ER-diagrams etc, consider <https://www.draw.io> or [GraphViz](https://graphviz.org/).
4. Feel free to talk to other members of the class in doing the homework. You should, however, write down your solutions yourself. If you're finding yourself drawing ER-diagrams on a paper/board with a classmate, you're going too far. List the names of everyone you worked with at the top of your submission.
5. Keep your solutions brief and clear.
6. Please use Piazza if you have questions about the homework but do not post answers. Feel free to use private posts or come to the office hours.

### Rubric

1. Always underline primary keys in ER/UML/Relations.
2. In the ER/UML, address as many constraints implied in the problem description as possible. Explicitly state any extra assumptions you assume that cannot be derived from the description.
3. When drawing ER or converting ER to relations, have the design principles in mind.
  - a. Try not to create unnecessary entities.
  - b. Try not to create tables that might suffer redundancy.

## Section 1. Concepts [20 pts]

### Part 1 [6 points]

Given two *entity sets*:  $EA = \{(a, 10), (b, 12), (c, 14), (d, 24)\}$  and  $EB = \{U, V, W, X, Y, Z\}$ . We create a *relationship set*  $R = \{((a, 10), W), ((c, 14), U), ((d, 24), V)\}$ . Note we often refer to “relationship set” simply as “relationship”. Answer the following questions

1. Is  $R$  a one-one relationship (set)? Briefly explain your reasoning.
2. Is  $R$  a many-one relationship (set) from  $EA$  to  $EB$ ? Briefly explain your reasoning.
3. Write a many-one relationship (set)  $R'$  from  $EB$  to  $EA$  that contains as many relationships as possible.
4. What is the maximum number of relationships possible in a many-many relationship (set)  $R$ ?

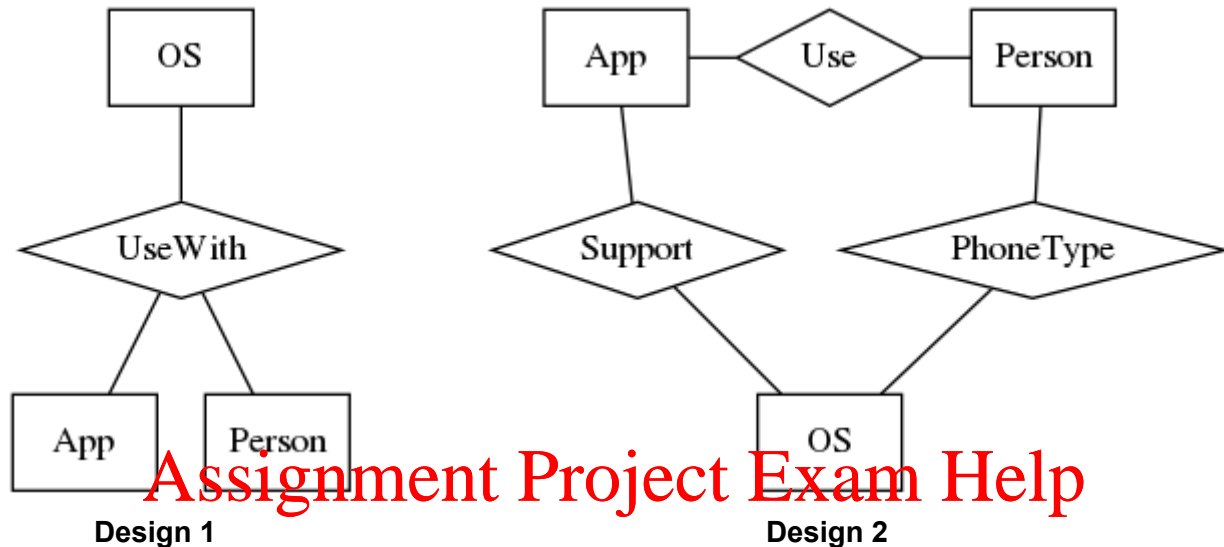
Solutions:

1. Yes. at most one. True for both sides
2. Yes. at most one. For  $EA \rightarrow EN$  side.
3.  $R' = \{(U, (a, 10)), (V, (a, 10)), (W, (a, 10)), (X, (a, 10)), (Y, (a, 10)), (Z, (a, 10)), \}$ .
4.  $4 \times 6 = 24$

Note: for Q1 and Q2, we also accept other answers as long as the reasoning is showing the student understands the difference between them. The question should be better formed such as “Is it possible that  $R$  is a one-one/many-one relationship ...”. Then the answer will be Yes for both. “One-one relationship” and “Many-one relationship” are *constraints* on the relation, not some conclusion drawn from a sample table. This is very similar to property of functional dependency.

## Part 2 [6 pts]

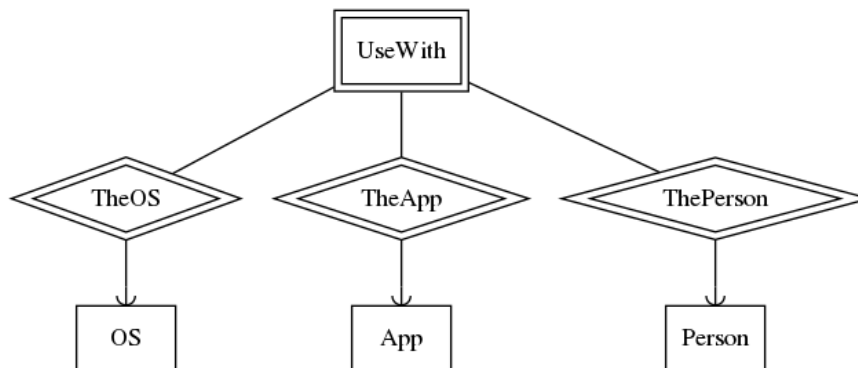
Given the below two ER diagrams that capture relationships among smartphone operating systems (OS), the applications (App) they support, and the users (Person) who use such OSs, answer the following questions:



1. Are design 1 and 2 equivalent? Explain your answers. If you think the two designs are not equivalent, then convert the multiway ER (design 1) into a binary one
2. Describe a scenario in which design 1 is better than design 2
3. Describe a scenario in which design 2 is better than design 1

Solution:

1. No. D1 captures 3-way relations while D2 can only capture relations between any two of them.



It's also OK to use "at most one" relationship instead of weak entity set.

2. To keep record of each user's current use status. Need to query "Whether a user X is using app Y on OS Z"

3. To compose queries relating each two entities at a time. For example, to query “Does the App Y support any OS that a user X is currently using”

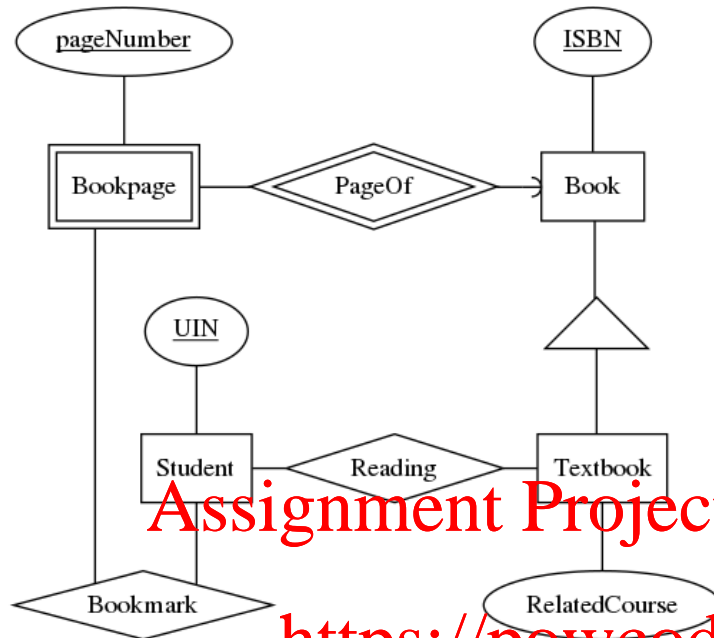
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

### Part 3 [8 pts]

Convert the following ER diagram into a relational model using DDL.



Solution:

Textbook(ISBN, RelatedCourse)

Book(ISBN)

Bookpage(pageNumber, ISBN)

Student(UIN)

Reading(UIN, ISBN)

Bookmark(UIN, pageNumber, ISBN)

```
CREATE TABLE Textbook(  
    ISBN    VARCHAR(30) NOT NULL PRIMARY KEY  
    ,RelatedCourse VARCHAR(16) NOT NULL  
    ,FOREIGN KEY (ISBN) REFERENCES Book(ISBN)  
);
```

```
CREATE TABLE Book(  
    ISBN VARCHAR(30) NOT NULL PRIMARY KEY  
);
```

```
CREATE TABLE Bookpage(  
    pageNumber INTEGER NOT NULL PRIMARY KEY
```

```
,ISBN    VARCHAR(30) NOT NULL PRIMARY KEY
,FOREIGN KEY (ISBN) REFERENCES Book(ISBN)
);
```

```
CREATE TABLE Student(
    UIN VARCHAR(30) NOT NULL PRIMARY KEY
);
```

```
CREATE TABLE Reading(
    UIN VARCHAR(30) NOT NULL PRIMARY KEY
    ,ISBN    VARCHAR(30) NOT NULL PRIMARY KEY
    ,FOREIGN KEY (ISBN) REFERENCES Book(ISBN)
    ,FOREIGN KEY (UIN) REFERENCES Student(UIN)
);
```

```
CREATE TABLE Bookmark(
    UIN    VARCHAR(30) NOT NULL PRIMARY KEY
    ,pageNumber INTEGER NOT NULL PRIMARY KEY
    ,ISBN    VARCHAR(30) NOT NULL PRIMARY KEY
    ,FOREIGN KEY (ISBN) REFERENCES Book(ISBN)
    ,FOREIGN KEY (UIN) REFERENCES Student(UIN)
    ,FOREIGN KEY (pageNumber) REFERENCES Bookpage(pageNumber)
);
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

## Section 2. Database schema design [55 points]

Throughout the 4 parts of this problem, we will **incrementally** design a database schema that contains information about a university—its departments, people, course registration, and semester information. We might need to do minor adjustment to the structures between consecutive parts. Keep the assumptions stated in the previous parts unless explicitly specified.

Notes:

- A. You need to underline any primary key in an ER/UML/relation.
- B. If you come across any ambiguity (say several alternative designs) when drawing an ER diagram/UML/table, please make an active design decision by choosing one possibility, and clearly stating your assumptions (if any).

### Part 1. [15 points]

Entity set: Department, People

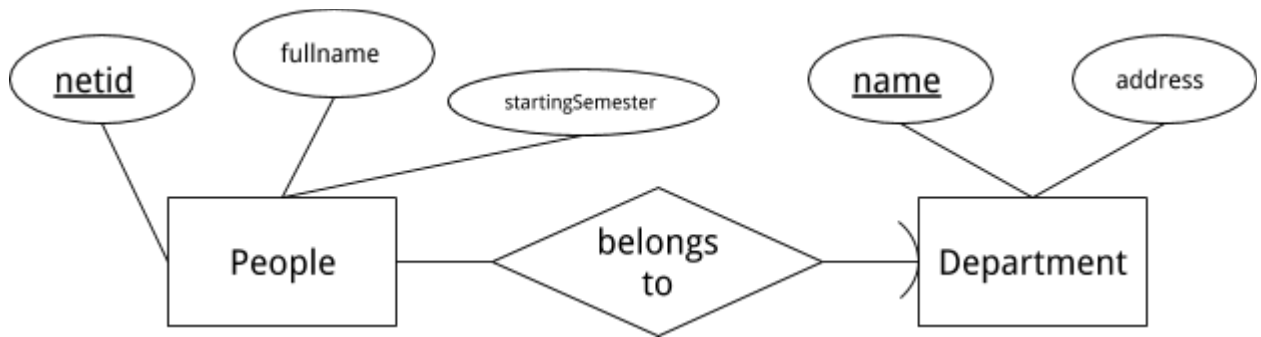
Department information includes name, and address. Person information includes netID, fullName, and startingSemester.

1. [5 points] Given the assumption that each person belongs to *exactly one* department, draw the ER diagram. Write down the relational tables of entities and relationships in your diagram, e.g., Department(name,...),...
2. [5 points] Since a person (e.g. a professor) can be affiliated to multiple departments, to improve the design, assume that each person must have *at least one* department association. Based on this assumption, draw the ER diagram, UML and write down the relational tables of entities and relationships in your diagram. Use this assumption in the following parts as well.
3. [5 points] Someone comes up with a controversial design and use a single table to represent the current scenario:  
TableOfEverything(netID, fullName, startingSemester, departmentName, departmentAddress)
  - a. What are the primary key(s) of the above table.
  - b. Evaluate this representation in terms of the design principles of schema.

Solution:

1. Tables:  
People(netID, fullName, startingSemester, departmentName),

Department(name, address)

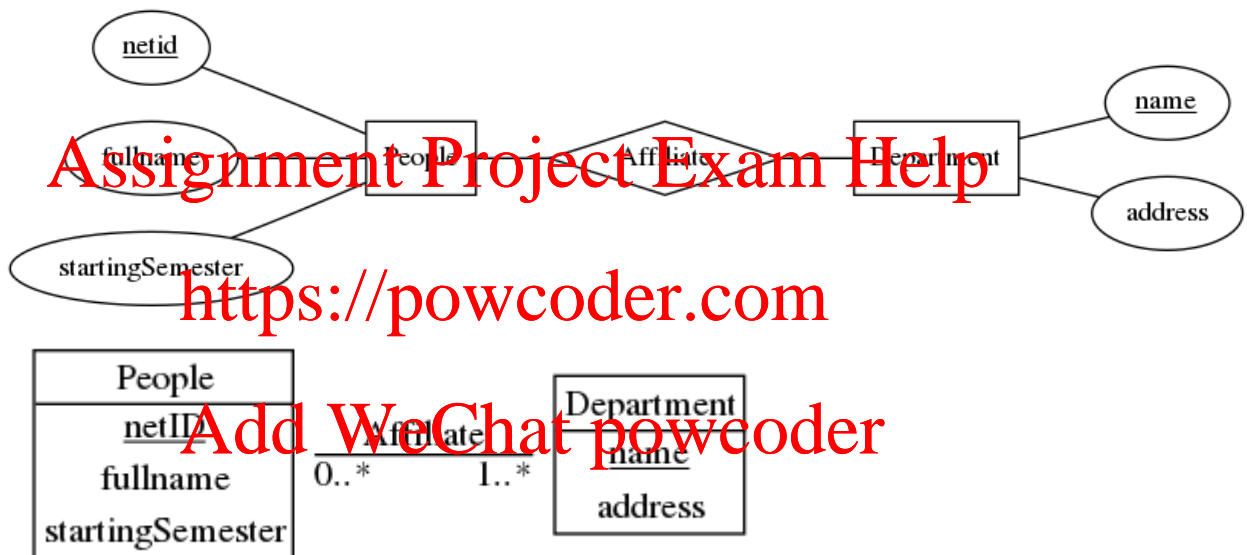


2. Tables:

People(netID, fullName, startingSemester),

Department(name, address)

Affiliate(netID, DepartmentName)



3.

a. NetID and DepartmentName are the primary keys.

b. It violates the principle of avoiding redundancy.

DepartmentAddress/fullname/startingSemester will be repeated multiple times in this table.

## Part 2. [20 points]

Continue from Part 1 and keep all existing entities and relationships.

Let's say within the university, people may have different roles, such as student, professor and staff. We're particularly interested in professors and students. In addition to the attributes in People entity, each professor also has an office room and a salary amount; and each student



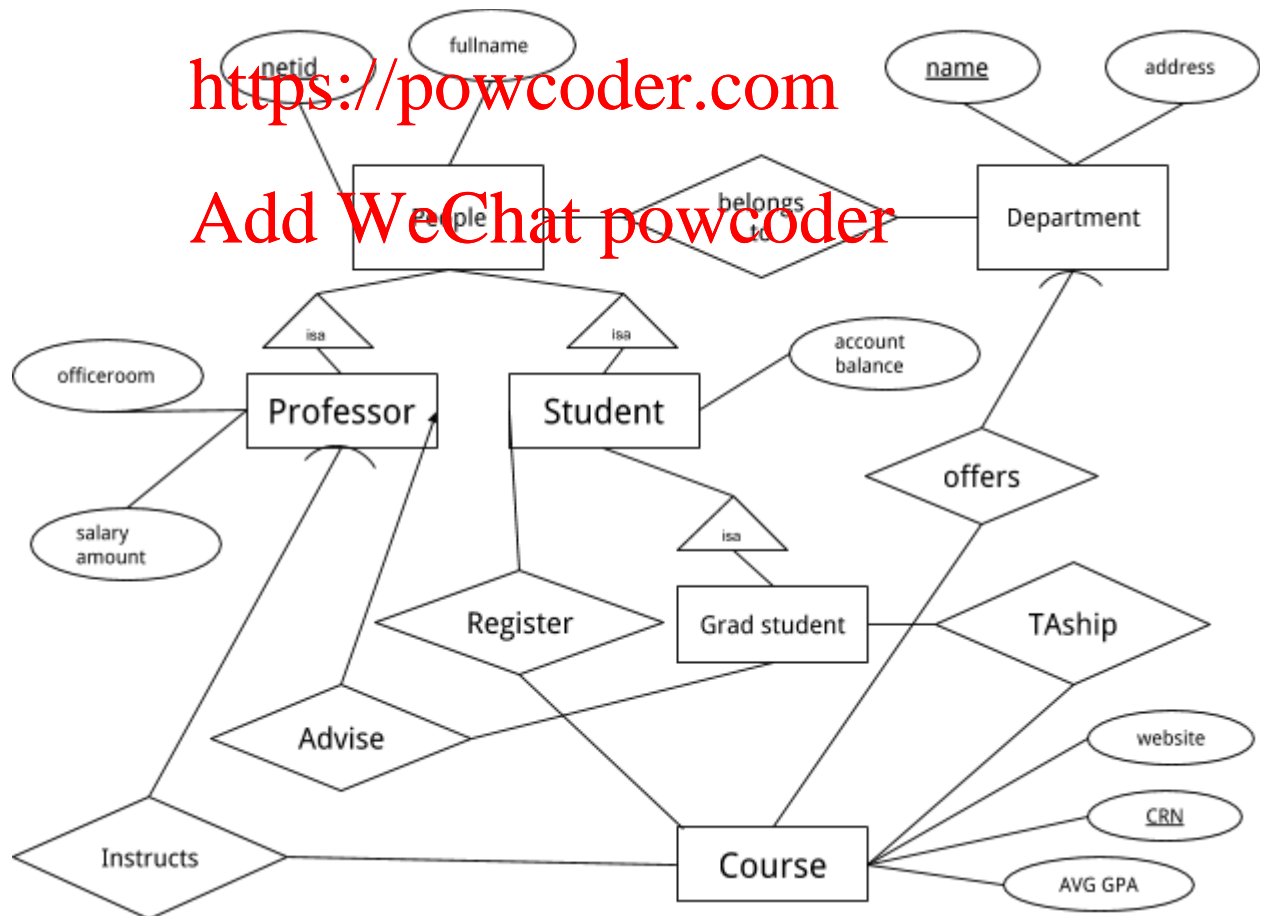
has an account balance attribute. We also want to keep records of *course registrations for all students*.

Among the students, we are further interested in graduate students who might hold TA positions. Each graduate student has at most one main advisor, who must be a professor. A course is related to attributes CRN, one professor as the instructor, 0 or any number of TAs, an average GPA of the class, and a course website address. A course is offered from one unique department.

1. Explain how you model the relationship between “People” and “Professor” entities here. Explain why or why not you use subclass or weak entity set to achieve this. [3 points]
2. Draw the ER diagram. [10 points]
3. If there’s a multi-way relationship in your ER diagram, convert it to binary relationship representation. If there’s no multiway relationship in your ER diagram, try to create one. Evaluate the versions before/after the change in terms of the design principles of ER. [2 points]
4. Draw the UML. [5 points]

Solution:

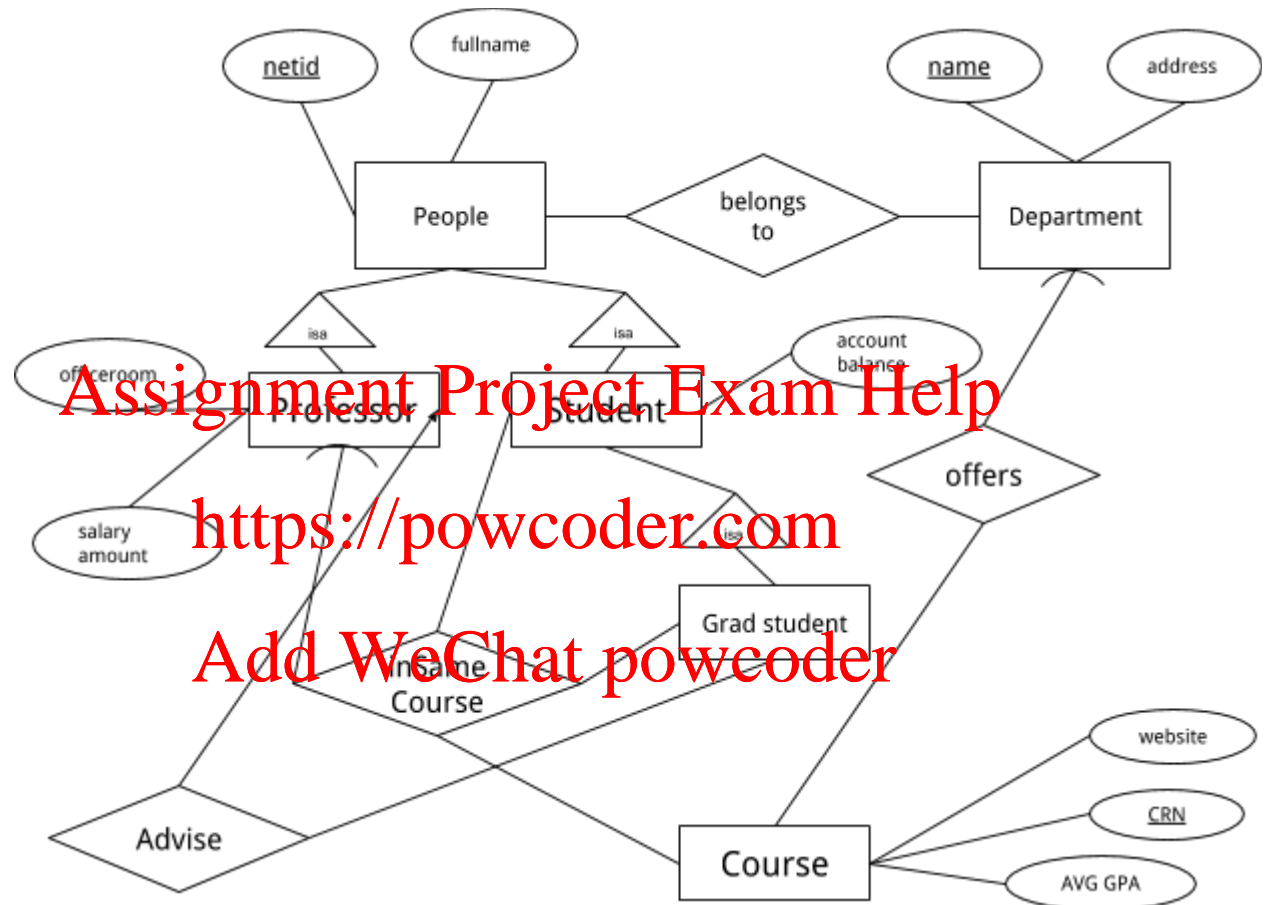
1. I use subclass. Professor has all attributes as people. I don’t use weak entity because I don’t need another person to distinguish a unique professor.



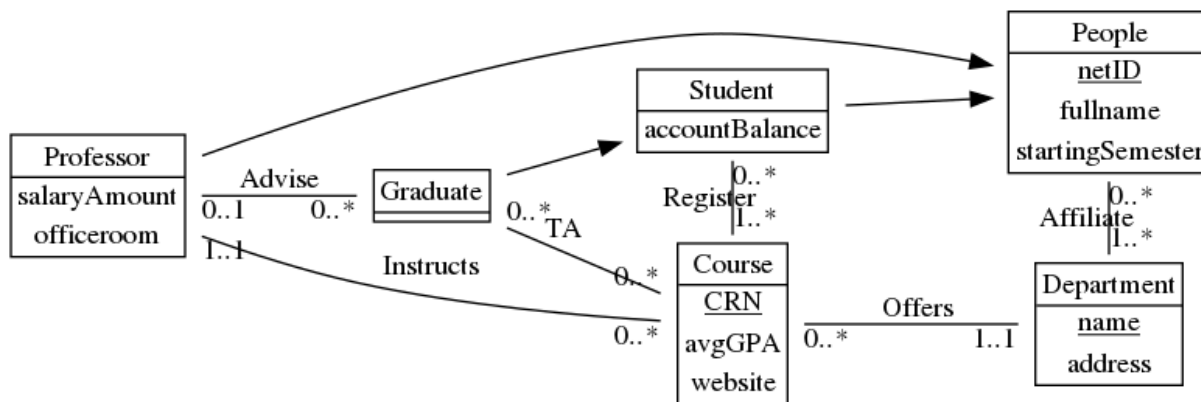
3. Instead of “Course”-“Prof”, “Course”-“Grad Student”, “Course”-“Student”, create a 4-way relationship “inSameCourse” relating Prof, Student, Grad and Course.

Table will be like: InSameCourse(Prof\_netid, Student\_netid, TA\_netid, CRN). This design introduces redundancy.

Why? Because if (\*, Stu1, \*, CRN1) exists in the table, and we know there's a single entry (?, Stu2, ?, CRN1), then all the entries (\*, Stu2, \*, CRN1) must also be in the table! Here, two students went to the same course meaning they met the same Professor & TAs.



4. UML:



### Part 3. [10 points]

Continue from Part 2 and keep all existing entities and relationships.

By the end of the semester when we deploy our database with the schema in Part 2, everyone likes our system and we are told our system will be used in the following semester as well.

However, our current design doesn't support the same course in multiple semesters. Therefore, we have to improve our design again.

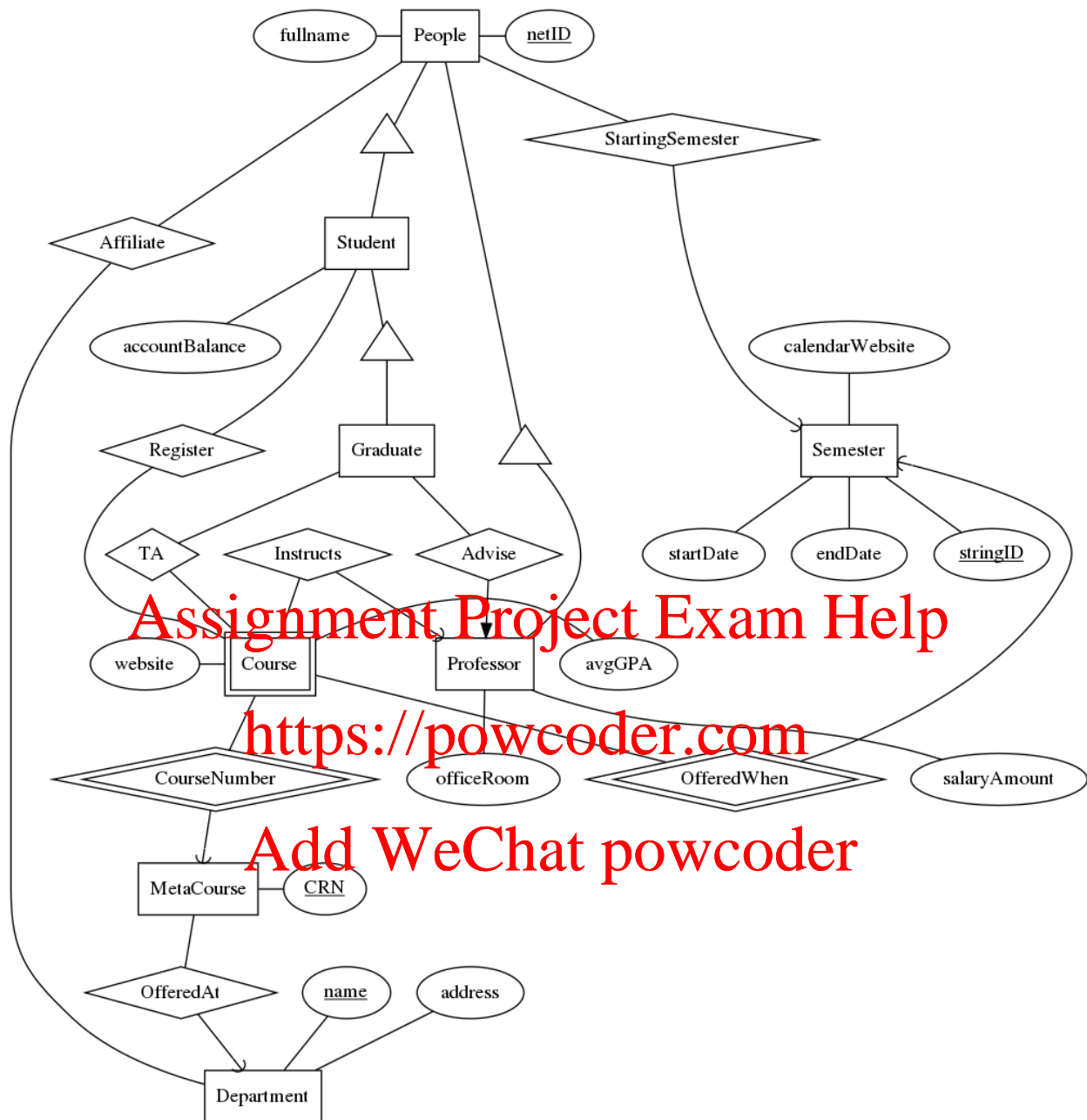
We augment the concept of semester in our schema. Things related to semester include start date and end date, university calendar website of the semester, and a string\_id (e.g. FA18).

We elevate the concept of "course" in part 2 to a new one called "meta-course". A meta-course has a unique CRN and can be offered in multiple semesters. *A student can't register a meta-course, but instead, its actual "course" in a specific semester.* Think about the relations and attributes of "course" in part 2, which ones of them now actually belong to meta-course, while the rest still remain specific to an actual course?

1. Explain how you distinguish the "meta-course" vs. "courses" here. Explain why or why not you use subclass or weak entity set to achieve this. [4 points]
2. Draw the ER diagram. [6 points]

Solution:

1. I use weak entity to distinguish them. Because in order to distinguish a "course" I have to know its "meta-course". "Course" is determined by "meta-course" and a "semester". I don't use subclass because "course" doesn't share the same attribute as "meta-course".
- 2.



#### Part 4. Make things real [10 points]

Continue from Part 3 and keep all existing entities and relationships.

1. List all the relations (tables) that would be resulted from the ER diagram in Part 3. This should enable us to readily create this schema in a database in DDL very easily. [2 points]

Example: Department(address, ...), Enrollment(...)

2. Translate any 5 relations you created, each of which has to contain at least 4 attributes into a relational schema by specifying the relations, their attributes, keys and foreign keys using SQL DDL. If there's any weak entity in the ER diagram, make sure one is among the 5 relations you choose. [8 points]

#### Solution

1. Department(name, address)  
People(netID, fullname, startingSemester)  
Student(netID, fullname, startingSemester, accountBalance)  
Graduate(netID, fullname, startingSemester, accountBalance, Professor.netID)  
// It's also OK to make Advise(Graduate.netID, Professor.netID) an individual table.  
Professor(netID, fullname, startingSemester, officeRoom, salaryAmount)  
Semester(stringID, startDate, endDate, calendarWebsite)  
MetaCourse(CRN, Department.name)  
Course(MetaCourse.CRN, Semester.stringID, website, avgGPA, Professor.netID)

Affiliate(People.netID, Department.name)

Register(Student.netID, CRN, stringID)

TA(Graduate.netID, CRN, stringID)

2. Student(netID, fullname, startingSemester, accountBalance)

```
CREATE TABLE `Student` (  
  `netID` varchar(30) NOT NULL PRIMARY KEY,  
  `fullname` varchar(30) NOT NULL,  
  `startingSemester` varchar(30) NOT NULL,  
  `accountBalance` varchar(30) NOT NULL  
);
```

Graduate(netID, fullname, startingSemester, accountBalance, Professor.netID)

```
CREATE TABLE `Graduate` (  
  `netID` varchar(30) NOT NULL PRIMARY KEY,  
  `fullname` varchar(30) NOT NULL,  
  `startingSemester` varchar(30) NOT NULL,  
  `accountBalance` varchar(30) NOT NULL,  
  `ProfessorNetID` varchar(30),  
  FOREIGN KEY (ProfessorNetID) REFERENCES Professor(netID)  
);
```

Professor(netID, fullname, startingSemester, officeRoom, salaryAmount)

```
CREATE TABLE `Professor` (  
  `netID` varchar(30) NOT NULL PRIMARY KEY,  
  `fullname` varchar(30) NOT NULL,  
  `startingSemester` varchar(30) NOT NULL,  
  `officeRoom` varchar(30) NOT NULL,  
  `salaryAmount` varchar(30) NOT NULL  
);
```

```

`netID` varchar(30) NOT NULL PRIMARY KEY,
`fullName` varchar(30) NOT NULL,
`startingSemester` varchar(30) NOT NULL,
`officeRoom` varchar(30) NOT NULL,
`salaryAmount` varchar(30) NOT NULL
);

```

Semester(stringID, startDate, endDate, calendarWebsite)

```

CREATE TABLE `Semester` (
  `stringID` varchar(30) NOT NULL PRIMARY KEY,
  `startDate` varchar(30) NOT NULL,
  `endDate` varchar(30) NOT NULL,
  `calendarWebsite` varchar(30) NOT NULL
);

```

Course(MetaCourse.CRN, Semester.stringID, website, avgGPA, Professor.netID)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

```

CREATE TABLE `Course` (
  `MetaCourseCRN` varchar(30) NOT NULL PRIMARY KEY,
  `SemesterStringID` varchar(30) NOT NULL PRIMARY KEY,
  `website` varchar(30) NOT NULL,
  `avgGPA` varchar(30) NOT NULL,
  `ProfessorNetID` varchar(30) NOT NULL,
  FOREIGN KEY (MetaCourseCRN) REFERENCES MetaCourse(CRN),
  FOREIGN KEY (SemesterStringID) REFERENCES Semester(stringID),
  FOREIGN KEY (ProfessorNetID) REFERENCES Professor(netID)
);

```

## Section 3. Functional dependency [20 pts]

For questions on functional dependency, you need to show the intermediate steps for getting full score.

1. Suppose a relation with attributes A, B, C, D, E, and F satisfies the following FDs.

$C \rightarrow A$

$BE \rightarrow F$

$BF \rightarrow CD$

$DF \rightarrow B$

Now, compute closure for the following sets: (i) {D, E}, (ii) {D, F}, (iii) {B, E, F}.

[6 points]

Solution:

(i)

DE+

= DE

**The closure is {D, E}**

(ii)

DF+

= DF

= BDF [DF  $\rightarrow$  B]

= BCDF [BF  $\rightarrow$  CD]

= ABCDF [C  $\rightarrow$  A]

**The closure is {A, B, C, D, F}**

(iii)

BEF+

= BCDEF [BF  $\rightarrow$  CD]

= ABCDEF [C  $\rightarrow$  A]

**The closure is {A, B, C, D, E, F}**

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

2. Assume that there are 4 attributes A, B, C, D, and the set of FDs is  $S = \{C \rightarrow B, A \rightarrow C\}$ . Now, compute the closure for the FDs in S. [6 points]

Solution:

A+ = ABC

B+ = B

C+ = BC

D+ = D

AB+ = ABC

AC+ = ABC

AD+ = ABCD

BC+ = BC

BD+ = BD

CD+ = BCD

ABC+ = ABC

ACD+ = ABCD

$BCD^+ = BCD$   
 $ABCD^+ = ABCD$

$A \rightarrow A, A \rightarrow C, A \rightarrow B,$   
 $B \rightarrow B,$   
 $C \rightarrow C, C \rightarrow B,$   
 $D \rightarrow D,$   
 $AB \rightarrow A, AB \rightarrow B, AB \rightarrow C,$   
 $AC \rightarrow A, AC \rightarrow B, AC \rightarrow C,$   
 $AD \rightarrow A, AD \rightarrow B, AD \rightarrow C, AD \rightarrow D,$   
 $BC \rightarrow C, BC \rightarrow B,$   
 $BD \rightarrow B, BD \rightarrow D,$   
 $CD \rightarrow B, CD \rightarrow C, CD \rightarrow D,$   
 $ABC \rightarrow A, ABC \rightarrow B, ABC \rightarrow C,$   
 $ACD \rightarrow A, ACD \rightarrow B, ACD \rightarrow C, ACD \rightarrow D,$   
 $BCD \rightarrow B, BCD \rightarrow C, BCD \rightarrow D,$   
 $ABCD \rightarrow A, ABCD \rightarrow C, ABCD \rightarrow B, ABCD \rightarrow D$

## Assignment Project Exam Help

3. Consider a relation  $R(A, B, C, D, E, F)$ . You are given the following dependencies:

$C \rightarrow AB$

$FD \rightarrow C$

$B \rightarrow FE$

List all the keys for  $R$  (make sure they are minimal, i.e. not a superset of some other key). [4 points]

<https://powcoder.com>

Add WeChat powcoder

Solution:

$CD, BD, FD$

$C^+$

$= C$

$= ABC [C \rightarrow AB]$

$= ABCEF [B \rightarrow FE]$

**$CD$  is a candidate Key**

$FD^+ = FD$

$= DF$

$= CDF [FD \rightarrow C]$

$= ABCDF [C \rightarrow AB]$

$= ABCDEF [B \rightarrow FE]$

**$FD$  is a candidate Key**



BD+  
= BD  
= BDEF [B  $\rightarrow$  FE]  
= BCDEF [FD  $\rightarrow$  C]  
= ABCDEF [C  $\rightarrow$  AB]  
**BD is a candidate Key**

4. Consider two sets of FDs for relation R(A, B, C, D): FD1 = {B $\rightarrow$ D, D $\rightarrow$ A, B $\rightarrow$ A} and FD2 = {B $\rightarrow$ D, D $\rightarrow$ A, B $\rightarrow$ C}. Are these two set of FDs equivalent? [4 points]

Solution:

No. They are not equivalent.

B $\rightarrow$ C is FD2 cannot be derived from FD1

**Assignment Project Exam Help**

**<https://powcoder.com>**

**Add WeChat powcoder**