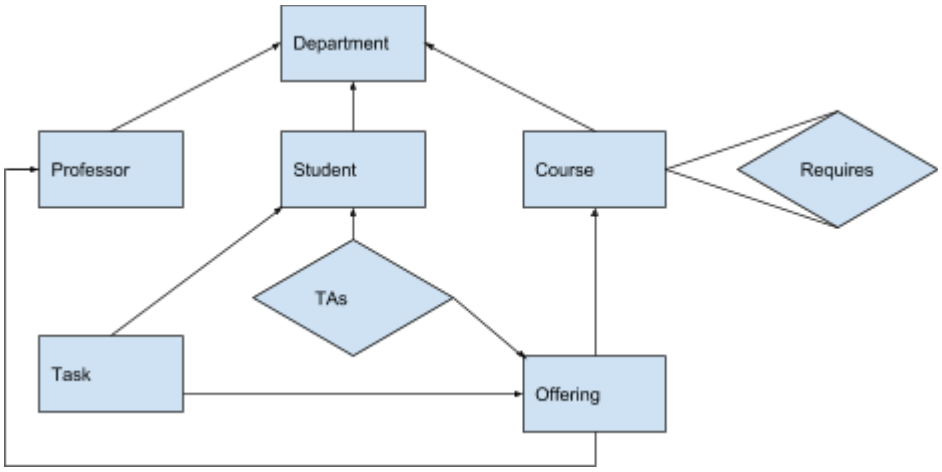
Christian Zinck

HW3

1.

1.



2.

Entities:

Department: Department name, address, phone number, website, chair (Faculty ID)

Professor: Faculty ID, Department name, name, email address, phone number, title

Student: Student ID, Department name, name

Course: Course ID, Department name, title, description

Offering: Class ID, Course ID, professor (Faculty ID)

Task: Task ID, Class ID, Student ID, weight, grade

Relations:

TAs: Class ID, Student ID

Requires: class (Class ID), prereq (Class ID)

2.

1.

```
drop table Department;
drop table Professor;
drop table Student;
drop table Course;
drop table Offering;
drop table Task;
drop table TAs;
drop table Requires;
```

```
create table Department(
        name varchar not null primary key,
        address varchar not null,
        phone_number int not null,
        website varchar not null,
        chair int not null references Professor);

create table Professor(
        id int not null primary key,
        department varchar not null references Department,
        name varchar not null,
        email_address varchar not null,
        phone_number int not null,
        title varchar not null);

create table Student(
        id int not null primary key,
        department varchar not null references Department,
        name varchar not null);

create table Course(
        id varchar not null primary key,
        department varchar not null references Department,
        title varchar not null,
        description varchar not null);

create table Offering(
        id int not null primary key,
        course varchar not null references Course,
        professor int not null references Professor);

create table Task(
        id int not null primary key,
        student int not null references Student,
        class int not null references Offering,
        weight int not null,
        grade int not null);
```

```
create table TAs(
        student int not null references Student,
        class int not null references Offering,
        primary key(student, class));

create table Requires(
        class varchar not null references Course,
        prereq varchar not null references Course,
        primary key(class, prereq));
```

2. In general, if there was a natural candidate key in the table, I used it as the primary key. For example, I used department name and Course ID as keys of their respective tables. For students, professors, offerings and tasks, artificial keys are necessary because there are no natural keys in the table. The relations can be only uniquely determined by the IDs of the two entities involved in the relation.

3.

1.
X -> Y
Y -> X
X -> Z
Z -> X
Y -> Z
Z -> Y
X, Y -> Z
X, Z -> Y
Z, Y -> X
X -> Y, Z
Y -> X, Z
Z -> X, Y

2.
All but Y -> Z and X, Y -> Z

3.
Only X -> Y and X -> Z

4.
None. In order to rule out an FD at least 2 rows are needed.

4.

1.

The only candidate key of the relation is V because it is the only column whose closure includes the entire relation. Since X is not a candidate key and the FD X -> Y, Z exists, the relation is not in BCNF. The closure of this FD is X, Y and Z, so these columns must be in their own table to achieve BCNF. After this decomposition, we get the two relations VWX and XYZ with V and X as primary keys respectively.

2.

The candidate keys of the relation are W, X and X, Y. The FDs X -> Z and Y -> W both prevent the relation from being in BCNF because neither X nor Y is a candidate key. The closures of X and Y are X, Z and Y, W respectively. We can therefore determine that the columns X, Z and columns Y, W must be in their own tables. The resulting decomposition is XY, XZ, and WY with primary keys XY, X, and Y respectively.