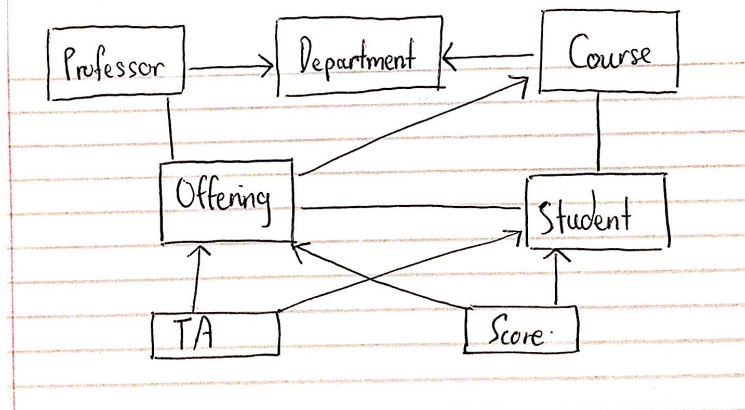


# Assignment 3

## **Part 1.1:**

Entity/Relationship diagram:



## **Part 1.2:**

### **Professor:**

- University ID
- Department
- Chair (bool)
- Name
- Email address
- Phone number
- Website link

### **Department:**

- Department name
- Department chair
- Address
- Phone number
- Website link

### **Course:**

- Department
- Catalog number
- Title
- Description
- Prerequisite

### **Student:**

- Name
- University ID
- Courses taken

### **Offering:**

- Course
- Semester
- Professor
- TAs
- Task

### **Score:**

- Offering
- Student ID
- Task
- Score (Weight is incorporated here)

### **TA:**

- Student ID
- Offering

### Part 2.1:

```
create table Professor (  
    p_id int(8) not null primary key,  
    dept_name varchar not null references Department,  
    chair bool not null;  
    prof_name varchar not null,  
    email varchar not null,  
    phone int(10) not null,  
    website varchar);  
  
create table Department (  
    dept_name varchar not null primary key,  
    prof_name varchar not null references Professor // Used for selective dept chair name  
    address varchar not null,  
    phone int(10) not null,  
    website varchar not null);  
  
create table Course (  
    c_id varchar not null primary key,  
    dept_name varchar not null references Department,  
    title varchar not null,  
    description varchar not null,  
    prerequisite varchar);  
  
create table Student (  
    name varchar not null,  
    s_id(8) not null primary key,  
    c_id varchar not null references Course); // To reference courses taken  
  
create table Offering (  
    c_id varchar not null references Course,  
    p_id varchar not null references Professor,  
    semester varchar not null,  
    task varchar not null, // To denote evaluation tasks.  
  
    Primary key (c_id, p_id, semester));  
  
Create table TA (  
    c_id varchar not null references Course,  
    p_id varchar not null references Professor,  
    semester varchar not null,  
    s_id int(8) not null references Student, // To reference TAs  
  
    Primary key (c_id, p_id, semester, s_id));
```

```

create table Score (
    c_id varchar not null references Course,
    p_id varchar not null references Professor,
    semester varchar not null,
    s_id not null references Student,           //To reference students enrolled
    Task not null references Offering,
    score int(3),    // Note that score incorporates weight as max score for a task is weight

    Primary key (c_id, p_id, semester, s_id, task);

```

## **Part 2.2:**

For my schema, we know that there are 4 main entities that have natural keys which are Courses, Students, Professor, and Department. The entities Offering, TA and Score arise from relationships between courses, students and professors. For this schema, I tried to not use surrogate keys as due to the nature of the 3-entity relationship, it is possible to use a combination of the natural keys p\_id, c\_id and a semester string (i.e FALL2018) to uniquely identify an Offering. Within an offering is TA and Score. The primary key of a TA relationship between an offering and a student can be modelled by incorporating the key used for an offering and the natural key for Student (s\_id). Similarly, Score can be identified based off of the primary used for offerings, in addition with an s\_id of a student taking the course, followed by the task that is being evaluated.

### **Part 3.1:**

List of possible FDs is:

$X \rightarrow Y$

$X \rightarrow Z$

$Y \rightarrow X$

$Y \rightarrow Z$

$Z \rightarrow Y$

$Z \rightarrow X$

$X \rightarrow Y, Z$

$Y \rightarrow X, Z$

$Z \rightarrow X, Y$

$X, Y \rightarrow Z$

$X, Z \rightarrow Y$

$Y, Z \rightarrow X$

### **Part 3.2:**

FDs that can be ruled out are:

$X \rightarrow Y$

$X \rightarrow Z$

$Y \rightarrow X$

$Z \rightarrow X$

$Z \rightarrow Y$

$X \rightarrow Y, Z$

$Y \rightarrow X, Z$

$Z \rightarrow X, Y$

$Y, Z \rightarrow X$

$X, Z \rightarrow Y$

### **Part 3.3:**

FDs that can be ruled out are:

$X \rightarrow Y$

$X \rightarrow Z$

$X \rightarrow Y, Z$

### **Part 3.4:**

No FDs can be ruled out.

#### **Part 4.1:**

Candidate keys are  $\{V\}$ ,  $\{X\}$

FD that violates BCNF:  $X \rightarrow Y, Z$

Closure of dependency:  $\{X, Y, Z\}$

Decompose  $(V, W, X, Y, Z)$  using  $\{X\}$

$C = \{X, Y, Z\}$

$S = \{X, Y, Z\}$

$T = \{V, W, X\}$

Now we have both S and T in BCNF.

#### **Part 4.2:**

Candidate keys are  $\{W, X\}$ ,  $\{X\}$ ,  $\{Y\}$

##### **Step 1:**

FD that violates BCNF:  $X \rightarrow Z$

Closure of dependency:  $\{X, Z\}$

Decompose  $(W, X, Y, Z)$  using  $\{X\}$

$C = \{X, Z\}$

$S = \{X, Z\}$

$T = \{W, X, Y\}$

Now we have both S and T in BCNF. But we are not finished yet

##### **Step 2:**

FD that violates BCNF:  $Y \rightarrow W$

Closure of dependency:  $\{Y, W\}$

Decompose  $(W, X, Y)$  using  $\{W\}$

$C = \{W, Y\}$

$S = \{W, Y\}$

$T_2 = \{X, W\}$

Now we have both S and T and  $T_2$  in BCNF.