INTRODUCTION TO DATABASES Project 1.

cl4335, xc2602

1. The Revised Version of the Document

○ **Description of our application**

For a university academic administrator, information about students, professors and courses is often abundant and confusing, and there are many relationships between them. In this case, it is very necessary to build a relevant database so that people can **easily store and access** the information they want.

We will build a database of the student course selection system, which aims to store information about students and courses, and allow students to use the system to query course rating, corresponding professor, credit, etc.

Due to the nature of our topic, our entities and relations are **highly correlated**, almost always forming a closed cycle between entities. This will be the top challenge for us when implementing the database for our application.

○ **Data plan**

Since it is hard to find real data about students and professors on the Internet, we decided to use fabricated data for the personnel information. And for the course information, we will try to use Columbia's department and course information.

○ **Part 3. options**

We will follow the Web Front-End Option for Part 3.

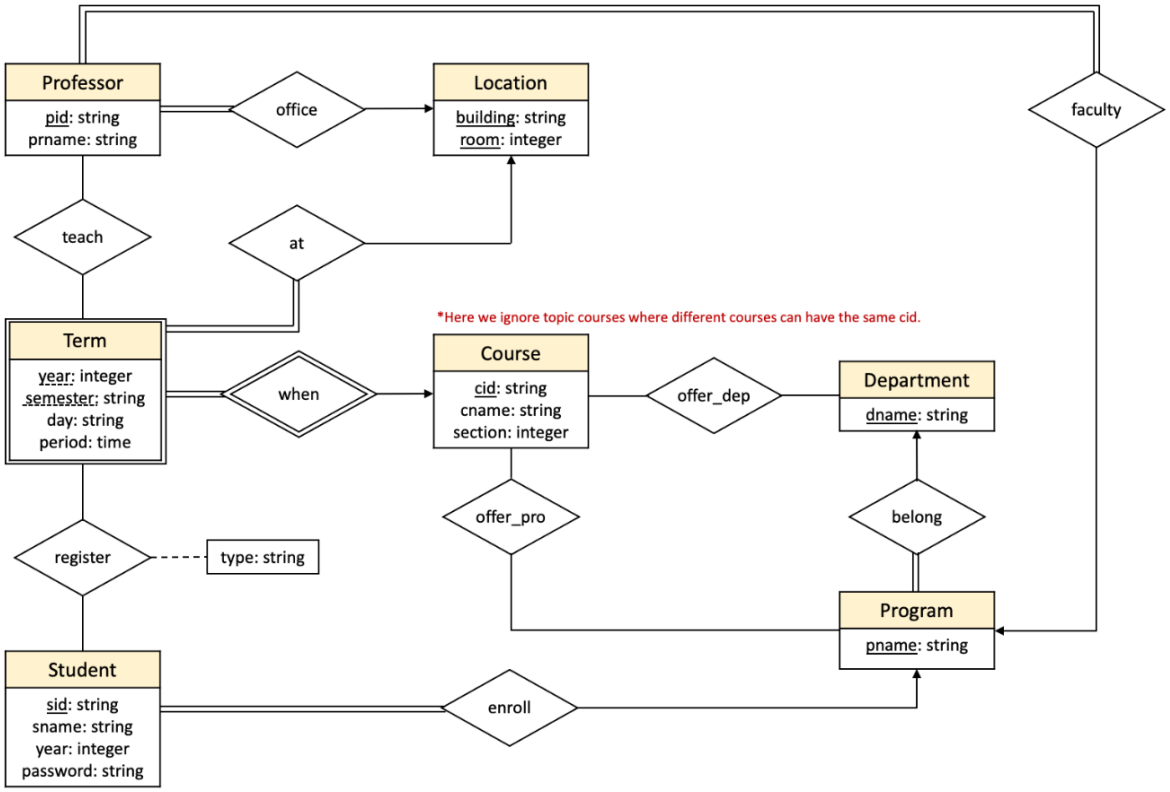
○ **Description of user interaction plans**

- Students can add courses with specific terms to the (wait, wish registered) list.
- Given sid, password, Students can log in to their accounts.
- Given sid, the type of list (wait, wish registered),return the courses list
- Given the pid, return the courses info list
- Given the lid, return the courses info list
- Given the lid, return professor info list
- Given the department name, return the courses info list
- Given the department name, return the program list
- Given the sid, return their program info (courses offered by the program, department it belongs to)
- Given cid, term, return courses info (location, professor, department)

2. The E/R diagram

We have
7 entities: *Student, Course, Professor, Location, Term, Department, Program.*
10 relationship sets: *register, teach, enroll, when, at, offer_dep, offer_pro, belong, office, faculty*.

Our E-R diagram is as follows:



Constraints:
Courses can be offered by departments or programs, which is why we have 2 relationship sets to separately represent 2 possibilities. Normally, courses offered by programs can also be categorized into offered by departments. However, courses offered by departments include all programs that belong to it.

3. SQL schema

```
CREATE TABLE Course(
   cid INTEGER,
   cname CHAR(30),
   section INTEGER,
   PRIMARY KEY(cid))

CREATE TABLE Department(
   dname CHAR(20),
   PRIMARY KEY(dname))

CREATE TABLE offer_dep(
   cid INTEGER,
   dname CHAR(20),
   PRIMARY KEY(cid, dname),
   FOREIGN KEY (cid) REFERENCES Course,
   FOREIGN KEY (dname) REFERENCES Department)

CREATE TABLE Prog_blg(
   pname CHAR(20),
   dname CHAR(20) NOT NULL,
   PRIMARY KEY(pname),
   FOREIGN KEY (dname) REFERENCES Department)

CREATE TABLE offer_pro(
   cid INTEGER,
   pname CHAR(20),
   PRIMARY KEY(cid, pname),
   FOREIGN KEY (cid) REFERENCES Course,
   FOREIGN KEY (pname) REFERENCES Prog_blg)

CREATE TABLE Stu_enrl(
   sid CHAR(10),
   sname CHAR(20),
   year INTEGER,
   password CHAR(20),
   pname CHAR(20) NOT NULL,
   PRIMARY KEY(sid),
   FOREIGN KEY (pname) REFERENCES Prog_blg)
```

```
CREATE TABLE Term_when(
   year INTEGER,
   semester CHAR(10),
   day CHAR(10),
   period TIME,
   cid INTEGER,
   PRIMARY KEY(cid, year, semester),
   FOREIGN KEY (cid) REFERENCES Course
      ON DELETE CASCADE)

CREATE TABLE register(
   sid CHAR(10),
   cid INTEGER,
   year INTEGER,
   semester CHAR(10),
   type CHAR(10),
   PRIMARY KEY(sid, cid, year, semester),
   FOREIGN KEY (sid) REFERENCES Stu_enrl,
   FOREIGN KEY (cid, year, semester) REFERENCES Term_when (cid, year, semester))

CREATE TABLE Location(
   building CHAR(20),
   room INTEGER,
   PRIMARY KEY(building, room))

CREATE TABLE Term_at(
   building CHAR(20),
   room INTEGER,
   cid INTEGER,
   year INTEGER,
   semester CHAR(10),
   day CHAR(10),
   period TIME,
   PRIMARY KEY(building, room, cid, year, semester),
   FOREIGN KEY (building, room) REFERENCES Location (building, room),
   FOREIGN KEY (cid, year, semester) REFERENCES Term_when (cid, year, semester))
```

```
CREATE TABLE Prof_off(
   pid CHAR(10),
   prname CHAR(20),
   building CHAR(20) NOT NULL,
   room INTEGER NOT NULL,
   PRIMARY KEY(pid),
   FOREIGN KEY (building, room) REFERENCES Location (building, room))

CREATE TABLE teach(
   pid CHAR(10),
   cid INTEGER,
   year INTEGER,
   semester CHAR(10),
   PRIMARY KEY(pid, cid, year, semester),
   FOREIGN KEY (pid) REFERENCES Prof_off,
   FOREIGN KEY (cid, year, semester) REFERENCES Term_when (cid, year, semester))

CREATE TABLE Prof_fac(
   pid CHAR(10),
   prname CHAR(20),
   pname CHAR(20),
   PRIMARY KEY(pid),
   FOREIGN KEY (pid) REFERENCES Prof_off,
   FOREIGN KEY (pname) REFERENCES Prog_blg)
```