Progress Report:

Our application is a scheduler that allows students to build a schedule of Duke classes with the offerings for the next semester. The classes include professors, corequisites such as labs or recitations, attributes, and departments. The class and professor combination also has a rating based off of previous Duke students ratings for the class and professor from an external source. This allows students to choose their class based off of ratings of professors and the class, especially if they are looking for a general requirement to be fulfilled. It will also allow the users to save up to a certain number of schedules for future use.
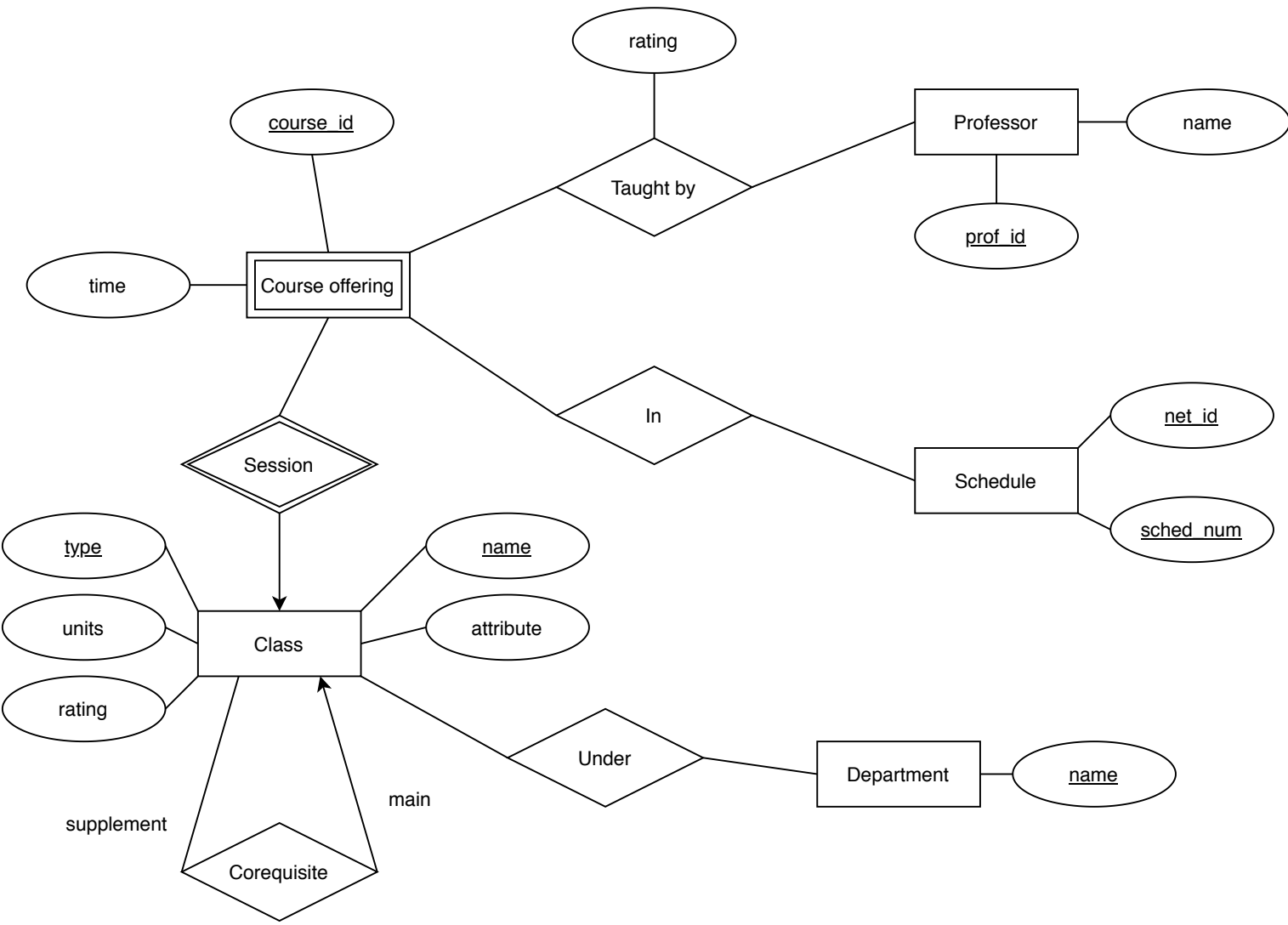
We plan to obtain our data from DukeHub for the different classes, requirements, attributes, departments, times, and professors. We also plan to utilize a scraper to obtain the data for the ratings. The sample data is in the source code with Insert statements for SQL.

We assume that the class timings will not change from now until the next semester, that classes won't change between now and the next semester, and that sections won't be added/dropped between now and the next semester. We also assume DukeHub has the correct information for all classes.

The E/R diagram and tables are on the next page.

The web interface is basically a split screen with one part of the screen being the schedule as it is updated and the other part being a list of all classes with professors, departments, attributes, times, and the ratings. You can search through the list of classes and then choose a class and add it to the schedule on the other side. On the bottom of the schedule, it will provide an option to save the schedule.

Note: There is no rounded arrow in draw.io, so we are using the barbed arrow to represent exactly one.

Class(name, type, attribute, units, rating)

Corequisite(main_name, main_type, supplement_name, supplement_type)

Department(name)

Professor(prof_id, name)

CourseDept(course_name, course_type, dept_name)

CourseOff(course_name, course_type, course_id, time, attribute, units, rating)

CourseProf(course_name, course_type, course_id, prof_id)

Schedule(net_id, sched_num, course_name, course_type, course_id)


CourseDept = "Under"

CourseOff = "Course offering"

CourseProf = "Taught by"

Schedule = "In"

Simplified schedule in SQL schema: no need for separate "Schedule" table, want to enforce limit of 6 classes in "In".