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. Our scheduler design offers substantially improved functionality compared to DukeHub's preexisting scheduler, which does not directly allow users to search for courses by attribute and does not provide ratings for specific professors per course.

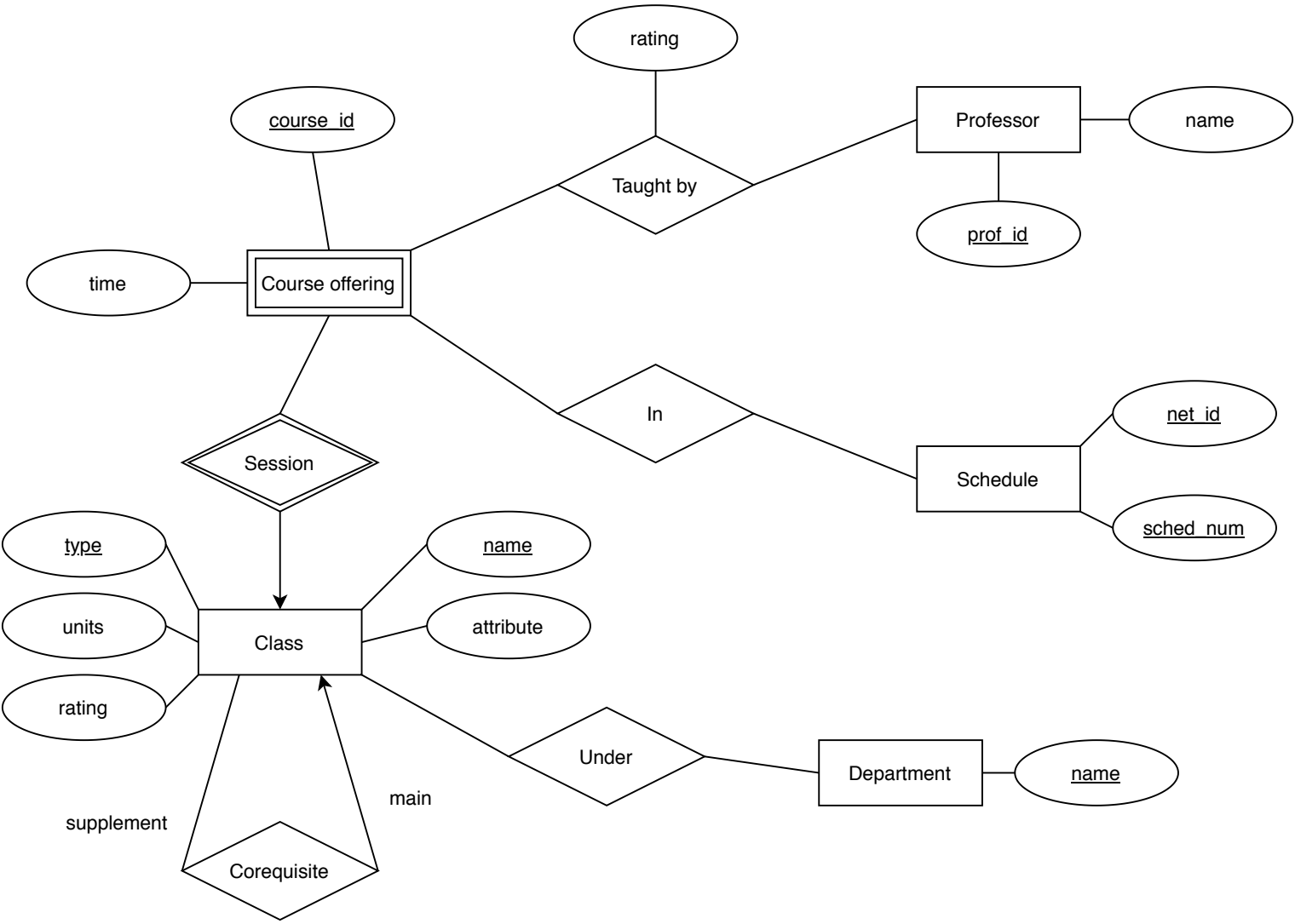
We plan to obtain our data from DukeHub for the different classes, requirements, attributes, departments, times, and professors. We also plan to utilize a scraper to obtain the data for the ratings. The sample data is in the source code with Insert statements for SQL.

We assume that class names are unique for each class, and that different sections for the same class (such as labs or recitations) are equivalent. We also disregard the course unit restrictions when forming schedules, since users may want to examine fewer than 4 units of courses (if a schedule is not finalized) or more than 6 units (if the user is examining various potential classes). Finally, we assume that DukeHub has the correct information about course times, class listings, and all sections.

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.



Database tables:

Class(name, type, units, attribute, 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)

CourseProf(course_name, course_type, course_id, prof_id, rating)

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

Relations between E/R diagram and SQL tables:

CourseDept = “Under”

CourseOff = “Course offering”

CourseProf = “Taught by”

Schedule = “In”

There is no need to reference a separate “Schedule” table for our application, so we did not include this table in our SQL database design.