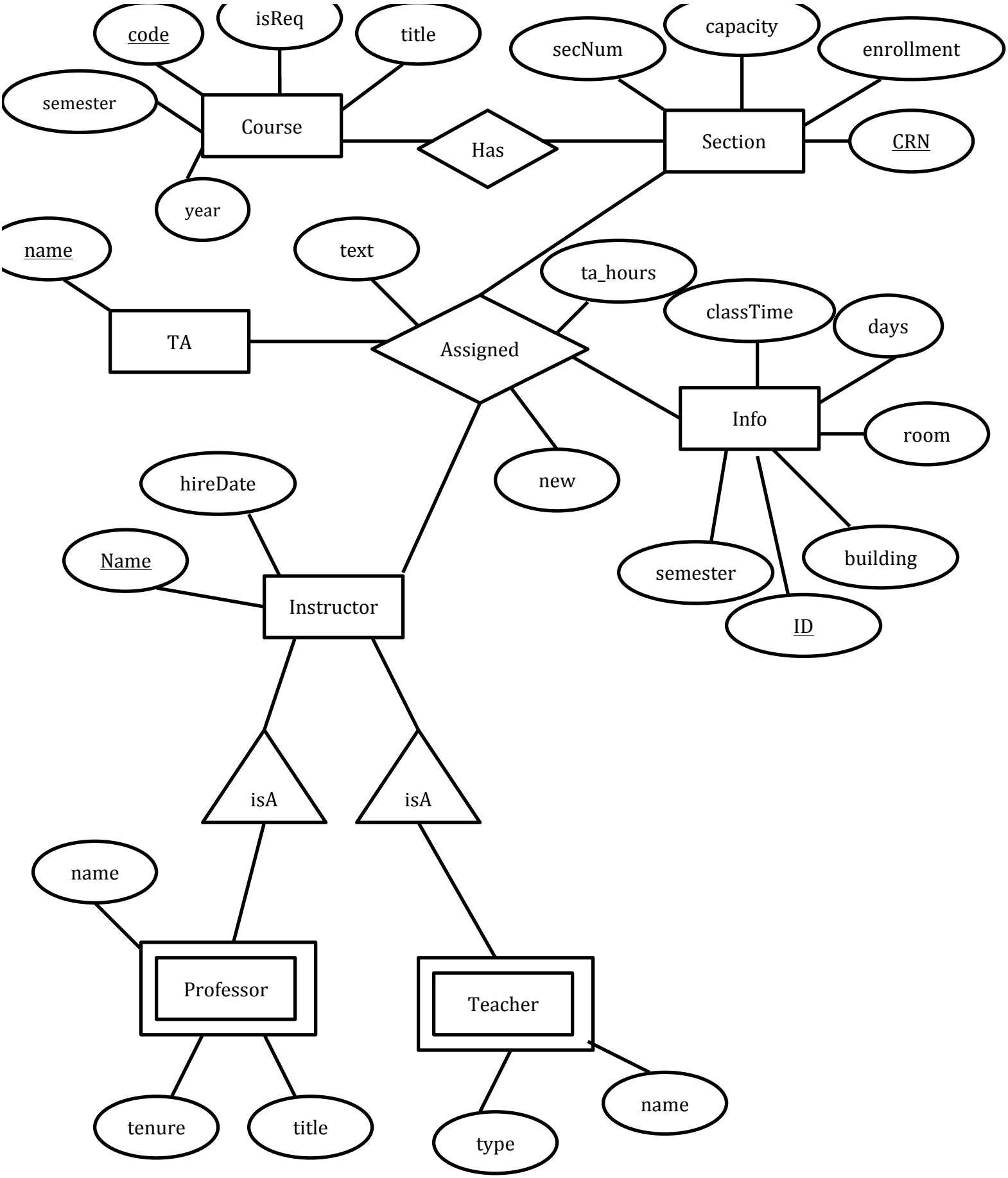
Intermediate Report

Database Project

Cole Troutman
Patrick Braud
Ryan Daily


code
isReq
title
semester
Course
Has
secNum
capacity
enrollment
Section
CRN
year
name
text
ta_hours
classTime
days
TA
Assigned
Info
room
new
hireDate
Name
Instructor
semester
building
ID
isA
isA
name
Professor
Teacher
tenure
title
type
name

ER Diagram Explanation:
We broke up the ER model into different entity sets which are; course, section, info, instructor, and TA. Instructor is broken up into weak entity sets, professor and teacher, so that we can only have the tenured attribute for professors instead of for both professors and teachers since teachers will not be tenured. All of our entity sets go through the assign relation which has the additional text (textbook) and new (is this the professors first time teaching the course). We make the info entity set so that the assign relation wouldn't have to have so many attributes attached to it.

Schemas:
assign(crn, info_id, instr_name, new, ta_hours, ta_name, textbook)
course(code, isReq, semester, title, year)
has(code, crn)
info(building, classTime, days, id, room_num, semester)
instructor(dateHired, name)
professor(name, tenure, title)
section(capacity, crn, enrollment, section_num)
ta(name)
teacher(name, type)

SQL Pseudo Code

Faculty/Staff Queries

a) Adding New Professor Information:
   Insert the values entered by the user into the instructor table.
   Code:
   INSERT INTO instructor(name, dateHired) Values('$name', '$dateHired')

b) Adding a Section
   Not yet implemented
c) Input the TA/Grader Information
   Update the assign table with the information on the TA that was input by the user. Show that the information has been added.
   Code:
   UPDATE assign
   SET ta_name = '$name', ta_hours = '$hours'
   WHERE assign.crn = '$crn';

   SELECT has.code, section.section_num\n
   FROM has, section\n
   WHERE section.crn = '$crn' and has.crn = '$crn';

   First query is for updating the TA information. The second query is for retrieving the information to display it to the user
d) Look Up Course in a certain catalog
   Get the code and title from the course table in the database where the year is the same year that the user had input.
   Code:
   SELECT code, title
   FROM course
   WHERE year = '$year'

Professor Queries
   Not yet implemented

Business Manager
   Not yet implemented

Design/pseudo code of PHP/MySQL program

All of our files follow the same format:

- The top is dedicated to the database connections and query implementation
- The next part is where we layout our page forms
- The bottom section is where we may or may not send feedback to the user based on the forms that they submit

User Interface Design

Our UI is pretty simple. We have text at the top of our page showing the different accounts instead of doing a log in for the different account. Clicking on these account names will bring you to a list of actions you can do from that account. Clicking on those actions will take you to the form where you actually fill out the information and submit the query. At any point you can always navigate backwards to the previous page.