Introduction to SQL exercise

Choose one of the tracks below based on your current SQL experience and interests. If you have substantial SQL experience you may install Postgres (if you have not already done so) or continue working on Homework 5. If you complete one track you may begin the next.

If you have Postgres installed, follow the setup directions in `setup-postgres.md`. Alternatively, go to https://www.db-fiddle.com/f/9U2vbN4X1w12LeiGna5UwB/0#&togetherjs=eVFEPI9ajG. The linked site allows you to do write your SQL queries online, but your changes will NOT be persistent, so you may need to keep previous queries in the editor pane to see their changes (or alternatively you could update the schema.)

For each track, write SQL queries to accomplish each task. You do not need to (and should not) run the actual student registration application for this exercise.

Track 1: Basic SQL queries

1. Insert two new students into the database.
2. Get all student information for all students.
3. Get the course names (and just the course names) for all courses.
4. Get all students with the last name Lee.
5. Order the results of the previous query by the students' first names.
6. Get all students whose first name contains `ha`.
7. Get the number of students whose first name contains `ha`.
8. Get the number of students in each course. (This is a single query.)

Track 2: SQL data definition language and join queries

1. For each course, get the course name, student's first name, and student's last name for each student in the course. (This is a single query.) This query will use the course number and student Andrew ID, but do not get the course numbers and Andrew IDs.
2. Add a column to record each student's grade in a course to the `sio_course_students` table.
3. Add some student grades to the database.
4. For each course, get the course name (not the course number) and the average grade for the course. (This is a single query.)

Track 3: SQL data definition language and relational structure

1. Alter the database structure so that it allows an arbitrary number of instructors for each course. (It's OK if your change here breaks the sample Django application; it assumes each course has a single instructor.)