

**PS4: SQL**

## 1 SQL Queries (60 points)

Import the university schema into your MySQL database by running the following scripts, available on the course's Canvas page:

- `University_DDL.sql` creates the necessary tables.
- `University_DML.sql` populates the tables with sample data.

Write and execute the following SQL queries using the university schema. For each query, show your SQL statement and the results returned by the database (you can include a screenshot of the query results). Note that some queries may return empty results.

1. For each department, find the maximum salary of instructors in that department.
2. Find the IDs of all students who were taught by an instructor named Katz; make sure there are no duplicates in the result.
3. Find the ID and title of each course in Comp. Sci. that has had at least one section with afternoon hours (i.e., ends at or after 12:00). (You should eliminate duplicates if any.)
4. Find the IDs and titles of all the courses that are (direct) prerequisite to the Robotics course.
5. Find the IDs and names of all instructors earning the highest salary (there may be more than one with the same salary).
6. Find the enrollment (number of students) in each section that was offered in Spring 2017. The result columns should be `course_id`, `section_id`, `students_num`. You do not need to output sections with 0 students.
7. Rewrite the preceding query, but also output sections with 0 students.
8. Find the IDs and names of all instructors who have taught at least 3 different courses.
9. Find the ID and name of the student with the highest number of 'A' grades (there may be more than one such student).
10. Find the ID and name of each History student who has *not* taken any Music courses.
11. Find the ID and name of each instructor who has never given an 'A' grade in any course she or he has taught. (Instructors who have never taught a course trivially satisfy this condition.)

12. Rewrite the preceding query, but also ensure that you include only instructors who have given at least one other non-null grade in some course.
13. For each student who have retaken a course at least once (i.e., the student has taken the course at least twice), show the student's ID, name and the course ID.
14. Find the IDs of those students who have retaken at least three distinct courses at least once (i.e., the student has taken the course at least two times).
15. Find the IDs and names of those instructors who have taught every course in their department.

## 2 SQL DML (10 points)

Write the SQL statements to perform the following operations on the university database:

1. Create a new course "CS-001" in the Comp. Sci. department, titled "Weekly Seminar", with 2 credits.
2. Create a section of this course in Spring 2022, with *sec\_id* of 1, and with the location of this section not yet specified.
3. Enroll every student in the Comp. Sci. department in the above section (*Hint*: use INSERT INTO SELECT statement).
4. Delete enrollments in the above section where the student's ID is 12345.
5. Delete the course CS-001. What happened to the section and enrollments of this course?

## 3 Stored Procedures (15 points)

Create a stored procedure that enrolls a student into a section of a course, but only if the student has completed the prerequisites for that course. If the prerequisites are not met, the procedure should return a message indicating that the student cannot be enrolled.

Call the procedure for a student who meets the prerequisites and a student who does not meet the prerequisites, and show the output in each case.

## 4 SQL DDL (15 points)

Write SQL DDL commands to create a database for an insurance company according to the following schemas:

```

person(driver_id, name, address)
car(license_plate, model, year)
owns(driver_id, license_plate)
accident(report_number, date, location)
participated(report_number, license_plate, driver_id, damage_amount)

```

Make any reasonable assumptions about data types, and be sure to declare primary and foreign keys.