Lab 12: SQL lab12.zip (lab12.zip)

Due by 11:59pm on Wednesday, April 24.

Starter Files

Download lab12.zip (lab12.zip). Inside the archive, you will find starter files for the questions in this lab, along with a copy of the Ok (ok) autograder.

Required Questions

SQL

A `SELECT` statement describes an output table based on input rows. To write one:

1. Describe the **input rows** using `FROM` and `WHERE` clauses.
2. **Group** those rows and determine which groups should appear as output rows using `GROUP BY` and `HAVING` clauses.
3. Format and order the **output rows** and columns using `SELECT` and `ORDER BY` clauses.

`SELECT` *(Step 3)* `FROM` *(Step 1)* `WHERE` *(Step 1)* `GROUP BY` *(Step 2)* `HAVING` *(Step 2)* `ORDER BY` *(Step 3)*;

Step 1 may involve joining tables (using commas) to form input rows that consist of two or more rows from existing tables.

The `WHERE` , `GROUP BY` , `HAVING` , and `ORDER BY` clauses are optional.

Consult the drop-down for a refresher on SQL. It's okay to skip directly to the questions and refer back here should you get stuck.

SQL

Final Exam Rooms

The `finals` table has columns `hall` (strings) and `course` (strings), and has rows for the lecture halls in which a course is holding its final exam.

The `sizes` table has columns `room` (strings) and `seats` (numbers), and has one row per unique room on campus containing the number of seats in that room. All lecture halls are rooms.

```
CREATE TABLE finals AS
  SELECT "RSF" AS hall, "61A" as course UNION
  SELECT "Wheeler"    , "61A"           UNION
  SELECT "Pimentel"   , "61A"           UNION
  SELECT "Li Ka Shing", "61A"           UNION
  SELECT "Stanley"    , "61A"           UNION
  SELECT "RSF"        , "61B"           UNION
  SELECT "Wheeler"    , "61B"           UNION
  SELECT "Morgan"     , "61B"           UNION
  SELECT "Wheeler"    , "61C"           UNION
  SELECT "Pimentel"   , "61C"           UNION
  SELECT "Soda 310"   , "61C"           UNION
  SELECT "Soda 306"   , "10"            UNION
  SELECT "RSF"        , "70";

CREATE TABLE sizes AS
  SELECT "RSF" AS room, 900 as seats    UNION
  SELECT "Wheeler"    , 700             UNION
  SELECT "Pimentel"   , 500             UNION
  SELECT "Li Ka Shing", 300             UNION
  SELECT "Stanley"    , 300             UNION
  SELECT "Morgan"     , 100             UNION
  SELECT "Soda 306"   , 80              UNION
  SELECT "Soda 310"   , 40              UNION
  SELECT "Soda 320"   , 30;
```

Q1: Big Courses

Create a `big` table with one column `course` (strings) containing the names of the courses (one per row) that have at least 1,000 seats in their final exam.

Your query should work correctly for any data that might appear in the `finals` and `sizes` table, but for the example above the result should be:

```
61A
61B
61C
```

Hint: Template

```
CREATE TABLE big AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

Use Ok to test your code:

```
python3 ok -q big
```

Q2: Seats Remaining

Create a `remaining` table with two columns, `course` (strings) and `remaining` (numbers), that has a row for each course. Each row contains the name of the course and the total number of seats in **all final rooms for that course except the largest one**.

Your query should work correctly for any data that might appear in the `finals` and `sizes` table, but for the example above the result should be:

```
10|0
61A|1800
61B|800
61C|540
70|0
```

Hint: Template

```
CREATE TABLE remaining AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

Use Ok to test your code:

```
python3 ok -q remaining
```

Q3: Room Sharing

Create a `sharing` table with two columns, `course` (strings) and `shared` (numbers), that has a row for **each course using at least one room that is also used by another course**. Each row contains the name of the course and the total number of rooms for that course which are also used by another course.

Reminder: `COUNT(DISTINCT x)` evaluates to the number of distinct values that appear in column `x` for a group.

Your query should work correctly for any data that might appear in the `finals` and `sizes` table, but for the example above the result should be:

```
61A|3
61B|2
61C|2
70|1
```

Hint: Template

```
CREATE TABLE sharing AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

Use Ok to test your code:

```
python3 ok -q sharing
```

Q4: Two Rooms

Create a `pairs` table with one column `rooms` (strings) that contains sentences describing pairs of rooms that together have at least 1,000 seats, along with the number of seats they have. The room names should appear in alphabetical order. Rows should appear in decreasing order of the total seats in the pair of rooms.

Your query should work correctly for any data that might appear in the `finals` and `sizes` table, but for the example above the result should be:

Hint: When adding numbers and including the result in a string, put parentheses around the arithmetic: `"1 + 2 = " || (1 + 2)`

```
RSF and Wheeler together have 1600 seats
Pimentel and RSF together have 1400 seats
Li Ka Shing and RSF together have 1200 seats
Pimentel and Wheeler together have 1200 seats
RSF and Stanley together have 1200 seats
Li Ka Shing and Wheeler together have 1000 seats
Morgan and RSF together have 1000 seats
Stanley and Wheeler together have 1000 seats
```

Hint: Template

```
CREATE TABLE pairs AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

Use Ok to test your code:

```
python3 ok -q pairs
```

Check Your Score Locally

You can locally check your score on each question of this assignment by running

```
python3 ok --score
```

This does NOT submit the assignment! When you are satisfied with your score, submit the assignment to Gradescope to receive credit for it.

Submit

Submit this assignment by uploading any files you've edited **to the appropriate Gradescope assignment.** Lab 00 (https://cs61a.org/lab/lab00/#submit-with-gradescope) has detailed instructions.

In addition, all students who are **not** in the mega lab must complete this attendance form (https://go.cs61a.org/lab-att). Submit this form each week, whether you attend lab or missed it for a good reason. The attendance form is not required for mega section students.