

---

## COMPUTER SCIENCE MENTORS 61A

April 28 – May 2, 2025

---

### 1 SQL

---

Last week we went over SQL and all the main keywords in SQL. This week we are going to go over more SQL problems.

competitor

| Clause              | Function  |
|---------------------|---|
| <b>SELECT</b>       | Specifies the columns to retrieve or calculate or used in creation of a new row |
| <b>UNION</b>        | Joins rows together   |
| <b>CREATE TABLE</b> | Creates a new named table   |
| <b>FROM</b>         | Specifies the table from which to retrieve data                                 |
| <b>WHERE</b>        | Chooses the rows that meet the condition  |
| <b>GROUP BY</b>     | Groups rows that share a property   |
| <b>HAVING</b>       | Filters groups based on conditions, applied after grouping                      |
| <b>ORDER BY</b>     | Sorts the result set by specified columns                                       |
| <b>LIMIT</b>        | Specifies the maximum number of rows to return                                  |

In SQL, the order of execution is as follows:

1. FROM
  2. WHERE
  3. GROUP BY
  4. HAVING
  5. SELECT
  6. ORDER BY
  7. LIMIT
1. After more than 100 years of operation, the Ringling Bros. circus is closing. A victory for animal rights advocates, the circus' closure poses a challenge for the zoologists tasked with moving the circus' animals to more suitable habitats.

The zoologists must first take the animals in a freight elevator with a weight limit of 2000. In order to speed up the process, the zoologists prefer to take groups of animals of the same species in the elevator, rather than one animal at a time.

Assume the zoologists will only put all of the animals of a particular species in the elevator, or take animals of that particular species one at a time.

You have access to the table `animals`, with columns containing the animals' names, heights, weights, and species.

| Name           | Height | Weight | Species |
|----------------|--------|--------|---------|
| Wilbur         | 4.1    | 150    | pig     |
| Tigress        | 4.4    | 700    | tiger   |
| Phil           | 3.3    | 79     | pig     |
| Dug            | 3.5    | 40     | dog     |
| Buddy          | 4      | 51     | dog     |
| Marty          | 4.9    | 300    | zebra   |
| Richard Parker | 5.2    | 918    | tiger   |

- (a) Write a query that returns the collective weight and species of animals in a group where there is more than one animal of a particular species in a group, and the collective weight of the animals in the group is less than 2000.

Your query should yield the following result.

```
91 dog
229 pig
1618 tiger
```

(b) To take the animals to their new habitats, the zoologists load the animals into trucks. The zoologists would like to reduce the number of trips taken by pairing up the animals for the trip (species does not matter). However, the trucks have two restrictions:

1. The maximum height of any animal is 5.0.
2. The total weight of both animals cannot exceed 300.

Write a query which will yield all possible pairings in a table (which is basically the table below). To match the table below, make sure that the left column animals' names come earlier in the alphabet than their partners.

|       |        |
|-------|--------|
| Buddy | Dug    |
| Buddy | Phil   |
| Buddy | Wilbur |
| Dug   | Phil   |
| Dug   | Wilbur |
| Phil  | Wilbur |

2. CS 61A wants to start a fish hatchery, and we need your help to analyze the data we've collected for the fish populations! Running a hatchery is expensive—we'd like to make some money on the side by selling some seafood (only older fish of course) to make delicious sushi.

The table `fish` contains a subset of the data that has been collected. The SQL column names are listed in brackets.

`fish`

| Species<br>[species] | Population<br>[pop] | Breeding Rate<br>[rate] | \$/piece<br>[price] | # of pieces per fish<br>[pieces] |
|----------------------|---------------------|-------------------------|---------------------|----------------------------------|
| Salmon               | 500                 | 3.3                     | 4                   | 30                               |
| Eel                  | 100                 | 1.3                     | 4                   | 15                               |
| Yellowtail           | 700                 | 2.0                     | 3                   | 30                               |
| Tuna                 | 600                 | 1.1                     | 3                   | 20                               |

`competitor`

| Species<br>[species] | \$/piece<br>[price] |
|----------------------|---------------------|
| Salmon               | 2                   |
| Eel                  | 3.4                 |
| Yellowtail           | 3.2                 |
| Tuna                 | 2.6                 |

- (a) Write a query to find the three most populated fish species.
- (b) Write a query to find the total number of fish in the ocean. Additionally, include the number of species we summed. Your output should have the number of species and the total population and it should yield in a table with a single row.
- (c) Profit is good, but more profit is better. Write a query to select the species that yields the greatest number of pieces for each price point. For example, if "Trout" and "Catfish" both cost 100 dollars/-piece, you should output the species that has the greater number of pieces. Your output should include the species, price, and number of pieces per fish.

- (d) Write a query that returns, for each species, the difference between our hatchery's revenue versus the competitor's revenue for one whole fish. Assume that number of pieces per one whole fish of a certain species of fish stays the same across both our hatchery and our comptitor's hatchery.

3. In this question, you have access to two tables.

**Grades**, which contains three columns: **day**, **class**, and **score**. Each row represents the **score** you got on a midterm for some **class** that you took on some **day**.

**Outfits**, which contains two columns: **day** and **color**. Each row represents the **color** of the shirt you wore on some **day**. Assume you have a row for each possible day.

grades

| Day   | Class    | Score |
|-------|----------|-------|
| 10/31 | Music 70 | 88    |
| 9/20  | Math 1A  | 72    |

outfits

| Day   | Color  |
|-------|--------|
| 11/5  | Blue   |
| 9/13  | Red    |
| 10/31 | Orange |

- (a) Instead of actually studying for your finals, you decide it would be the best use of your time to determine what your "lucky shirt" is. Suppose you're pretty happy with your exam scores this semester, so you define your lucky shirt as the shirt you wore to the most exams.

Write a query that will output the color of your lucky shirt and how many times you wore it.

- (b) You want to find out which classes you need to prepare for the most by determining how many points you have so far. However, you only want to do so for classes where you did relatively poorly.

Write a query that will output the sum of your midterm scores for each class along with the corresponding class, but only for classes in which you scored less than 80 points on at least one midterm. List the output from highest to lowest total score.