CS 61A Fall 2020

Structure and Interpretation of Computer Programs

Quiz 10 Solutions

INSTRUCTIONS

- Please review this worksheet before the exam prep session. Coming prepared will help greatly, as the TA will be live solving without allocating much time for individual work.
- Either Sean or Derek will be on video live solving these questions. The other TA will be answering questions in the chat. It is in your best interest to come prepared with **specific** questions.
- This is not graded, and you do not need to turn this in to anyone.
- Fall 2020 students: the boxes below are an artifact from more typical semesters to simulate exam environments. Obviously this doesn't apply to this semester's exams, but we just kept the fields to keep our materials looking professional :) Feel free to ignore them.
- For multiple choice questions, fill in each option or choice completely.
  - ☐ means mark **all options** that apply
  - ◯ means mark a **single choice**

Last name	
First name	
Student ID number	
CalCentral email (_@berkeley.edu)	
Discussion Section	___ ___ ___
All the work on this exam is my own. **(please sign)**	

1. **The Gram**

   Suppose Instagram stores a table named `follow` with two columns, followee and follower. As the name suggests, a follower in the table follows the followee. Write a SQL query to get the number of followers each *follower* has. As a specific example, say our table looks like this

followee	follower
A	B
B	C
B	D
D	E

   Then the output should look like this

Popular person	Followers
B	2
D	1

   There is no row for A because A does not follow anyone. There is no row for C because no one follows C :(Order does not matter

```
SELECT f1.follower, COUNT(DISTINCT f2.follower)
FROM follow AS f1, follow AS f2
WHERE f1.follower = f2.followee
GROUP BY f1.follower;
```

2. **Prime Time**

   Assume you have a table called `ns` that contains numbers from 0 to 10,000 in a column called `n`. Select all prime numbers from this table. Remember that 2 is the smallest prime number.

```
SELECT a.n

FROM ns AS a, ns AS b

WHERE (b.n < a.n AND b.n > 1 AND a.n % b.n != 0) OR (a.n = 2)

GROUP BY a.n

HAVING (COUNT(*) = a.n - 2) or a.n = 2;
```

3. **Accounting**

Say you have a table called `employees` holding columns `Id`, `Month`, and `Salary`. Each `Id` corresponds to some employee. Write a query that outputs the cumulative sum of each employee's salary in the last 3 months, including the current month. For example, if we have a table like this

Id	Month	Salary
1	1	20
2	1	20
1	2	30
2	2	30
3	2	40
1	3	40
3	3	60
1	4	60
3	4	70

Then the output should be

Id	Month	Salary
1	4	130
1	3	90
1	2	50
1	1	20
2	1	20
2	2	50
3	2	40
3	3	100
3	4	170

For the first row in the output, the cumulative salary for employee 1 for the last 3 months relative to month 4 (months 4, 3, and 2) is $60 + 40 + 30$. In the second row, the cumulative salary from the last 3 months for employee 1 relative to month 3 (months 3, 2, and 1) is $40 + 30 + 20$. Order does not matter

```
SELECT a.id, a.month, SUM(b.salary)

FROM employees as a, employees as b

WHERE a.id = b.id AND a.month - b.month >= 0 AND a.month - b.month < 3

GROUP BY a.id, a.month;
```