Mo'SQL

W4D1

vs.

ALL OF THE STUDENTS

AND

THEIR COHORT NAME

JS

```
/* FROM students */
const result = input.students.map(student => {
  /* JOIN cohorts ON cohort.id = students.cohort_id */
  const cohort = input.cohorts.find(cohort => {
    return cohort.id === student.cohort_id
  })

  /* SELECT
     students.id,
     students.name AS student_name,
     cohorts.name AS cohort_name
  */
  return {
    id: student.id,
    student_name: student.name,
    cohort_name: cohort.name
  }
})
```

JS 27ms

```
/* FROM students */
const result = input.students.map(student => {
  /* JOIN cohorts ON cohort.id = students.cohort_id */
  const cohort = input.cohorts.find(cohort => {
    return cohort.id === student.cohort_id
  })

  /* SELECT
     students.id,
     students.name AS student_name,
     cohorts.name AS cohort_name
  */
  return {
    id: student.id,
    student_name: student.name,
    cohort_name: cohort.name
  }
})
```

SQL 7ms

```
SELECT
  students.id,
  students.name AS student_name,
  cohorts.name AS cohort_name
FROM students
JOIN cohorts
ON cohorts.id = students.cohort_id;
```

12 Cohorts

72725 Assignment Submissions

ALL OF THE STUDENTS

AND

THE NUMBER OF COMPLETED ASSIGNMENTS

JS

```
/* FROM students */
const result = input.students.map(student => {
  /* JOIN assignment_submissions
     ON assignment_submissions.student_id = students.id */
  const submissions = input.assignment_submissions
  .filter(submission => {
    return submission.student_id === student.id
  }).length

  /* SELECT
     students.id,
     students.name,
     count(assignment_submissions.id) AS assignments_complete
  */
  return {
    id: student.id,
    student_name: student.name,
    assignments_complete: submissions
  }
})
```

JS 1122ms

```
/* FROM students */
const result = input.students.map(student => {
  /* JOIN assignment_submissions
     ON assignment_submissions.student_id = students.id */
  const submissions = input.assignment_submissions
  .filter(submission => {
    return submission.student_id === student.id
  }).length

  /* SELECT
     students.id,
     students.name,
     count(assignment_submissions.id) AS assignments_complete
  */
  return {
    id: student.id,
    student_name: student.name,
    assignments_complete: submissions
  }
})
```

SQL 42ms

```
SELECT
  students.id,
  students.name,
  count(assignment_submissions.id)
  AS assignments_complete
FROM students
JOIN assignment_submissions
ON assignment_submissions.student_id = students.id
GROUP BY students.id;
```

INDEX THE COUNT BY STUDENT ID

712ms

JS

42ms

SQL

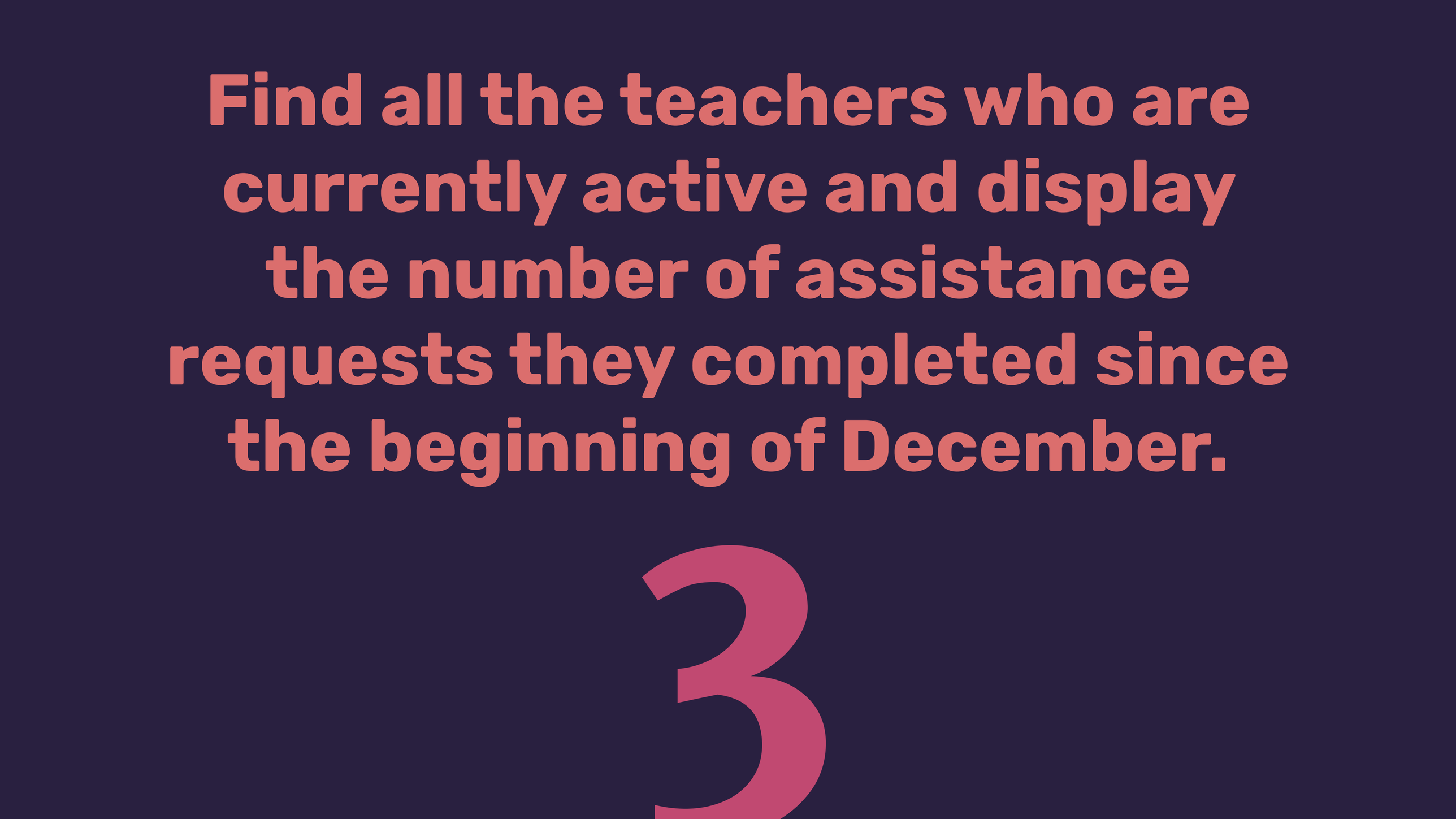
SCENARIOS

Hey Developer, I need to know...

Find all inactive cohorts and identify how many assistance requests were made per student on average.

Find all the students who are currently active and display the number of complete and the total assignments for the program.

Find all the teachers who are currently active and display the number of assistance requests they completed since the beginning of December.



Find all active students that have a rating of less than 2.

Find all active students and display their requests per day and average rating.

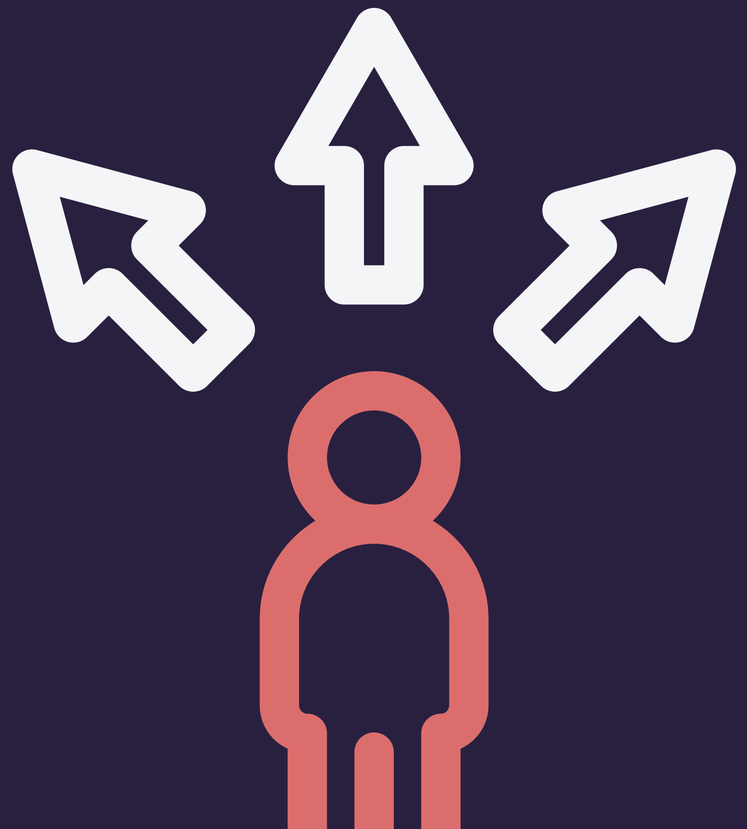
5

Find out how many assistance requests have been created for each assignment.

Find out which assignments a specific student hasn't finished.

7

DECISIONS DECISIONS DECISIONS



SHORT

FEW

MANY

LONG

PERFORMANCE

Questions?