⌂ Course / Indexes and Transactions / Indexes Quiz

Previous          ✎ ✓          Next >

# Indexes Quiz

⊓ Bookmark this page

✎ Hide Notes

Each multiple-choice quiz problem is based on a "root question," from which the system generates different correct and incorrect choices each time you take the quiz. Thus, you can test yourself on the same material multiple times. We strongly urge you to continue testing on each topic until you complete the quiz with a perfect score at least once. Simply click the "Reset" button at the bottom of the page for a new variant of the quiz.

After submitting your selections, the system will score your quiz, and for incorrect answers will provide an "explanation" (sometimes for correct ones too). These explanations should help you get the right answer the next time around. To prevent rapid-fire guessing, the system enforces a minimum of 10 minutes between each submission of solutions.

## Q1

1.0/1.0 point (graded)
[Q1] Consider the following relational schema:

```
Course(courseName unique, department, instrID)
Instructor(instrID unique, office)
Student(studentID unique, major)
Enroll(studentID, courseName, unique
(studentID,courseName))
```

Suppose there are five types of queries commonly asked on this schema:

- Given a course name, find the department offering that course.

- List all studentIDs together with all of the departments they are taking courses in.

- Given a studentID, find the names of all courses the student is enrolled in.

- List the offices of instructors teaching at least one course.

- Given a major, return the studentIDs of students in that major.

Which of the following indexes could NOT be useful in speeding up execution of one or more of the above queries?

- ○ Index on Enroll.studentID

- ● Index on Instructor.office

- ○ Index on Student.major

- ○ Index on Instructor.instrID

✔

Submit

## Q2

1/1 point (graded)

[Q2] Consider a table storing temperature readings taken by sensors:

```
Temps(sensorID, time, temp)
```

Assume the pair of attributes [sensorID,time] is a key. Consider the following query:

```
select * from Temps
where sensorID = 'sensor541'
and time = '05:11:02'
```

Consider the following scenarios:

A - No index is present on any attribute of Temps

B - An index is present on attribute sensorID only

C - An index is present on attribute time only

D - Separate indexes are present on attributes sensorID and time

E - A multi-attribute index is present on (sensorID,time)

Suppose table Temps has 50 unique sensorIDs and each sensorID has exactly 20 readings. Furthermore there are exactly 10 readings for every unique time in Temps.

For each scenario A-E, determine the maximum number of tuples that might be accessed to answer the query, assuming one "best" index is used whenever possible. (Don't count the number of index accesses.) Which of the following combinations of values is correct?

- ◉ A:1000, D:10, E:1
- ○ B:20, D:10, E:10
- ○ A:1000, D:20, E:10
- ○ B:10, C:10, E:10

✔

**Problem Explanation**
The query has an equality condition on both the sensorID and time attributes of Temps. Since [sensorID,time] is a key, the query result contains either 0 or 1 tuples.

Scenario A: Since there are no indexes, all tuples of the table may need to be accessed to look for 'sensor541' and '05:11:02'. The number of tuples in Temps is 50 (unique sensorIDs) * 20 (number of readings per sensor) = 1000.

Scenario B: Using the index on sensorID, 20 readings will match

the given sensorID, and all 20 tuples may need to be accessed to look for a matching time.

Previous

Next >

# edX

About

Affiliates

edX for Business

Open edX

Careers

News

# Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

Sitemap

# Connect

Blog

Contact Us

Help Center

Media Kit

Hide Notes