

CS348: Introduction to Database Systems (S'17)

Assignment 4 (due by 10am on Monday, July 24, 2017)

1 Overview

Here is a database schema for the first two questions:

```
Student(Sid integer,      /* student identification number */
        Name char(50))   /* student name */
Course(Cid integer,      /* class identification number */
        Name char(20),   /* class name */
        Semester integer) /* suggested semester (numbered 1-12) */
Prereq(Cid integer,      /* class identification number */
        Required integer) /* prerequisite class number */
Enrolled(Sid integer,    /* student identification number */
          Cid integer,    /* class identification number */
          Semester integer) /* actual semester (numbered 1-12) */
```

1. Relational Algebra:

Write the following two queries in Relational Algebra:

Query 1: Find all students who took 2nd year classes (semesters 4-6) in their first year (semesters 1-3). List their name and student id.

Query 2: List all students who (at any time during their studies) have taken a class but not taken all its prerequisites before. List student id and name.

2. Query Plans:

For the above queries list a possible access plan, including the choice of physical operators. Estimate its cost. Try to pick the best plan you can think of.

Use the following stats for your estimates:

	# of records	indices
Student	10000	B+ tree on Sid
Course	5000	B+ tree on Cid
Enrolled	50000	B+ tree on Sid and B+ tree on Cid
Prereq	10000	B+ tree on Cid

and assume the block size is 256 bytes. Indices are clustered except the second B+ tree on `Enrolled(Cid)`. All the student ids and class ids are distinct.

If you need to make any additional assumptions, remember to list them with your solution.

1. Schedules

Consider the following (incomplete) schedule S (containing operations issued by transactions T_1 , T_2 , and T_3):

`r1[x] r1[y] w1[x] r2[y] w3[y] w1[x] r2[y]`

Assume that all the three transactions eventually commit. Is this schedule serializable?

Add the smallest number of operations to S that guarantee that the extended schedule is

- recoverable
- conflict-serializable

solve each case separately.

2. Locking Schedulers

Show that the schedules produced using the two-phase locking (2PL) protocol are always serializable (hint: show that the serialization graph cannot contain a cycle if 2PL is used).

For extra credit show, that violating the 2PL rule (namely, acquiring a lock after another lock was released by a transaction) may lead to non-serializable schedules.

2 Assignment Submission

You should submit:

1. a PDF file `a4.pdf` containing solutions to the above questions (hand-written/drawn and scanned solutions, perhaps with additional assumptions stated in English).