

CPSC-608 Database Systems

Fall 2009

Instructor: Dr. Jianer Chen

Office: HRBB 309B

Phone: 845-4259

Email: chen@cs.tamu.edu

Office Hours: MWF 11:30am-12:20pm

Teaching Assistant: Mu-Fen Hsieh

Office: HRBB 328A

Phone: 845-5457

Email: mufen@cse.tamu.edu

Office Hours: TTh 2:00pm-3:30pm (or by appointment)

COURSE PROJECT TESTING

The score for the implementation of Tiny-SQL interpreter is 87 points, and the project report is 20 points. If your report experimentally shows the relation between table sizes and disk I/Os for your algorithm, 3 extra points will be given.

Your interpreter will be tested using SQL queries and is expected to output results for each query. You should output number of disk I/O's used in executing a query to show any optimization technique you use. Similar queries will be used to test your interpreter as follows. The queries in boldface are basic requirements. Each required query is given 2 points, and each other query is given 0.5 point.

1. Data update (11 points)

- **(Creation)** CREATE TABLE course (sid INT, homework INT, project INT, exam INT, grade STR20)
- **(Insertion)** INSERT INTO course (sid, homework, project, exam, grade) VALUES (1, 100, 100, 100, "A")
- (Insertion) INSERT INTO course (sid, grade, exam, project, homework) VALUES (2, "B", 60, 70, 80)
- (Insertion) INSERT INTO course (sid, grade, exam, project, homework) SELECT * FROM course

- **(Deletion)** DELETE FROM course WHERE grade="A"
- **(Deletion)** DELETE FROM course
- **(Drop)** DROP table course

2. Single table queries (18.5 points)

Assume the table has 10 tuples.

- **(Selection)** SELECT * FROM course
- **(Projection)** SELECT sid, course.grade, FROM course
- **(Duplicate elimination)** SELECT DISTINCT grade FROM course
- (Duplicate elimination) SELECT DISTINCT * FROM course
- **(Sorting)** SELECT * FROM course ORDER BY exam
- **(Condition)** SELECT * FROM course WHERE exam = 100
- **(Condition)** SELECT * FROM course WHERE grade = "A"
- **(Condition)** SELECT * FROM course WHERE exam = 100 AND project = 100
- SELECT * FROM course WHERE exam = 100 OR exam = 99
- SELECT * FROM course WHERE NOT exam = 100
- SELECT * FROM course WHERE exam > 70
- SELECT * FROM course WHERE exam = 100 OR homework = 100 AND project = 100
- SELECT * FROM course WHERE (exam = 100 OR homework = 100) AND project = 100
- SELECT * FROM course WHERE exam + homework = 200
- SELECT * FROM course WHERE (exam * 30 + homework * 20 + project * 50) / 100 = 100
- SELECT * FROM course WHERE grade = "C" AND (exam > 70 OR project > 70) AND NOT (exam * 30 + homework * 20 + project * 50) / 100 < 60

3. Single table queries (8 points)

Assume the table has 250 tuples.

- **(Selection)** SELECT * FROM course
- **(Duplicate elimination)** SELECT DISTINCT grade FROM course
- **(Sorting)** SELECT * FROM course ORDER BY exam

- **(Condition)** SELECT * FROM course WHERE exam = 100

4. Two table queries (12.5 points)

Suppose there are two tables course(sid INT, homework INT, project INT, exam INT, grade STR20) and course2(sid INT, exam INT, grade STR20), each has 10 tuples and 8 tuples respectively.

- **(Product or natural join depending on your assumption)** SELECT * FROM course, course2
- **(Natural join and projection)** SELECT course.sid, course.grade, course2.grade FROM course, course2 WHERE course.sid = course2.sid
- **(Natural join and duplicate elimination)** SELECT DISTINCT course.grade, course2.grade FROM course, course2 WHERE course.sid = course2.sid
- **(Natural join and sorting)** SELECT * FROM course, course2 WHERE course.sid = course2.sid ORDER BY course.exam
- **(Natural join and condition)** SELECT * FROM course, course2 WHERE course.sid = course2.sid AND course.exam = 100 AND course2.exam = 100
- **(Natural join and condition)** SELECT * FROM course, course2 WHERE course.sid = course2.sid AND (course.exam = 100 OR course2.exam = 100)
- **(Natural join and condition)** SELECT * FROM course, course2 WHERE course.sid = course2.sid AND course.exam > course2.exam
- SELECT * FROM course, course2 WHERE course.sid = course2.sid AND course.exam > course2.exam AND course.homework = 100
- SELECT * FROM course, course2 WHERE course.sid = course2.sid AND (course.exam > course2.exam OR course.homework = 100)
- **(Putting together)** SELECT DISTINCT course.grade, course2.grade FROM course, course2 WHERE course.sid = course2.sid AND (course.exam > course2.exam OR course.grade = "A" AND course2.grade = "A") ORDER BY course.exam

5. Two table queries (12.5+4 points)

Repeat the tests in 4. Assume the two tables has 50 and 250 tuples respectively. Implementation of condition pushing down will be given 4 points.

6. Two table queries (12.5 points)

Repeat the tests in 4. Assume the two tables has 250 and 2500 tuples respectively.

7. Multiple table queries (4+4 points)

Implementation of join/product tree optimization will be given 4 points.

- **(Natural join)** Suppose there are three tables: $r(a,b)$, $s(b,c)$ and $t(a,c)$.
`SELECT * FROM r, s, t WHERE r.a = t.a AND r.b = s.b AND s.c = t.c`
- **(Product)** Suppose there are six tables: $t1(c1)$, $t2(c2)$, $t3(c3)$, $t4(c4)$, $t5(c5)$, $t6(c6)$.
`SELECT * FROM t1, t2, t3, t4, t5, t6`