UCLA
Computer Science Department

Instructor: C. Zaniolo
TA: Hetal Thakkar

Student Name and ID: [redacted]

CS143–Fall 2007—MIDTERM EXAM: Closed Book, 110 Minutes

Attach extra pages as needed. Write your name and ID on any extra page that you attach. Please, write neatly.

Problem	Score	
A	(22%)	22
B.1	(22%)	10
B.2	(8%)	8
B.3	(8%)	8
C	(40%)	16
Total	(100%)	64

Extra Credit: 4

Midterm Score: 68

22

Problem A. Relational Algebra [22 points]

Consider the relation `taken(StID, Course, Quarter, Year)` whose tuples record that a student took a given class in a given quarter. You can assume that `StID`, `Quarter`, and `Year` are integers.

Write a relational algebra expression (no aggregates!) to find the `StID` of every student who, in some quarter of year 2006, took **exactly two courses.**

SELECT * FROM taken AS A INNER JOIN Taken AS B ON A.Course <> B.Course
WHERE Year = 2006 AND A.Quarter = B.Quarter

✓

~~$\sigma_{a.Course \neq b.Course} (\rho_a(taken) \times \rho_b(taken))$~~
~~a.Quarter = b.Quarter~~
~~a.Year = 2006~~
~~b.Year = 2006~~

StID

$$\left(\sigma_{\substack{a.Course \neq b.Course \wedge \\ a.Quarter = b.Quarter \wedge \\ a.Year = 2006 \wedge \\ b.Year = 2006 \wedge \\ a.StID = b.StID}} (\rho_a(taken) \times \rho_b(taken)) \right) -$$

$$\Pi_{StID} \left(\sigma_{\substack{c.Course \neq d.Course \wedge \\ e.Course \neq d.Course \wedge \\ e.Course \neq c.Course \wedge \\ c.Year = 2006 \wedge \\ d.Year = 2006 \wedge \\ e.Year = 2006 \wedge \\ c.Quarter = d.Quarter \wedge \\ c.Quarter = e.Quarter \wedge \\ c.StID = d.StID \wedge \\ c.StID = e.StID}} (\rho_c(taken) \times \rho_d(taken) \times \rho_e(taken)) \right)$$

Problem B. SQL

Consider the relation `warehouse(PartNo, SupplierNo, Price, Days)` which describes the suppliers for each part, along with the price they charge and the number of days needed to deliver the part to the warehouse.

To cut costs, the manager of our warehouse wants to eliminate redundant suppliers. A supplier is **redundant** when **all** the parts he/she supplies are also supplied by other suppliers in the **same number of days or in fewer days.**

B.1 [22 points] Write an SQL query to find all redundant suppliers.

B.2 [8 points] Write an SQL statement to delete all redundant suppliers.

B.3 [8 points] In reality, the objective of our manager was to prune out unneeded suppliers **while preserving at least one supplier for each part in the original relation.** Will the manager's objective be realized by the execution of B.2? Explain your answer using a simple example.

10

B1 SELECT A.supplierNo FROM warehouse AS A, warehouse AS B WHERE B.supplierNo <> A.supplierNo AND B.PartNo = A.PartNo AND B.Days <= A.Days GROUP BY A.supplierNo

-12

is selects suppliers. thout the "=" it selects all non-fastest upplier for each part, nich is not what is asked ← not needed

B.2 DELETE FROM warehouse WHERE supplierNo IN (SELECT A.SupplierNo FROM warehouse AS A, warehouse AS B WHERE B.supplierNo <> A.supplierNo AND B.Days <= A.Days GROUP BY A.SupplierNo)

subquery wrong from above

8

B.3 No, this will not always work because if 2 suppliers have identical parts and also have equal days for each part than both suppliers would be treated as redundant by this query and removed, leaving no supplier.

long as no suppliers same days ive minimum will be, a

16 **Problem C. Files and Storage [40 points]**

The relation: `took(StudentID, CourseNo, Quarter, Year, Units, Grade)`

contains the grades for the courses completed by UCLA students during the last 20 years. For simplicity, assume that there are 25,000 students enrolled each quarter, and that each student takes four courses per quarter, and that there are four quarters each year. Then we get a total of 8,000,000 records. If 10,000 new students enter UCLA every year, we can assume that in `took` there are 200,000 different students, each identified by its `StudentID`. On the average, a student took 40 different courses.

1. If the file blocks hold 2048 bytes and each `took` tuple requires 100 bytes. How many blocks will then be needed to store the unspanned tuples of this relation ?
2. The table has a **sparse index** on `StudentID`. Say that `StudentID` requires 10 bytes and the pointers to the blocks also require 10 bytes. Is this a primary index? [Explain your answer]
3. Compute the levels and the blocks at each level required by this index implemented as a B+ tree. Assume a best-case scenario.
4. Compute the levels and the blocks at each level required for the B+ tree assuming the worst-case scenario.
5. How many blocks from the B+ tree and the file will the DBMS retrieve from disk to answer the following query: find the GPA (average grade weighted by units) for a given student. Report your results first assuming the best-case scenario and then the worst-case scenario. In both cases assume that the main-memory buffers are initially empty.

1. $\frac{2048}{100} = 20$

$\frac{8,000,000}{20} =$ 400,000 blocks

2. Yes, it is a primary index because it is a sparse index which can only lead to sets based on student ID

3. $\log_2$ (200,000) levels

4. 4 levels

5. $\log_2(200{,}000) + \frac{30}{2048} + 40$