**NAME:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**COM S 363 Final**

|  |  |  |
| --- | --- | --- |
| **Problem** | **Max Points** | **Points** |
| **1** | **8** |  |
| **2** | **7** |  |
| **3** | **18** |  |
| **4** | **15** |  |
| **5** | **15** |  |
| **6** | **8** |  |
| **7** | **9** |  |
| **8** | **20** |  |
| **Total** | **100** |  |

1. (8 points) Construct an ER diagram for a simple database that records the following information for the United Soccer League (USL).
   1. There are many teams in the league.
   2. Each team has a name, a city, a coach, a captain, and a set of players
   3. Each player belongs to only one team.
   4. Each player has a name, a position (such as left wing striker or goalie), a skill level.
   5. A player may be injured. Each injury has an ID, a description, and the game when it happens.
   6. A team captain is also a player.
   7. A game is played between two teams (referred to as host\_team and guest\_team) and has a date (such as May 4, 2020) and a score (such as 4 to 2)
2. (7 points) Consider a relation R with attributes A, B, C, D, E, and G, and the set of dependencies F = {AB → C, AC → B, AD → E, B → D, BC → A, E → G}. Answer the following questions.
3. (3 points) Find all keys that contain only two attributes
4. (4 points) Suppose R is decomposed into R1(ABC), R2(ACDE), and R3(ADG). Is this a lossless join decomposition? Explain your answer in details.
5. (18 points) query processing. We have the following relations.

* **students**

|  |  |  |
| --- | --- | --- |
| snum | name | gender |
| 1001 | Randy | M |
| 1005 | Nicole | F |

* **departments**

|  |  |  |
| --- | --- | --- |
| **code** | **name** | **college** |
| 401 | Computer Science | LAS |
| 402 | Mathematics | LAS |
| 403 | Chemical Engineering | Engineering |
| 404 | Landscape Architect | Design |

* **degrees**

|  |  |  |
| --- | --- | --- |
| **name** | **level** | **department\_code** |
| Computer Science | BS | 401 |
| Software Engineering | BS | 401 |
| Computer Science | MS | 401 |
| Computer Science | PhD | 401 |
| Applied Mathematics | MS | 402 |
| Chemical Engineering | BS | 403 |
| Landscape Architect | BS | 404 |

* **major**

|  |  |  |
| --- | --- | --- |
| **snum** | **name** | **level** |
| 1001 | Computer Science | BS |
| 1005 | Applied Mathematics | MS |

* **minor**

|  |  |  |
| --- | --- | --- |
| **snum** | **name** | **level** |
| 1005 | Computer Science | MS |
| 1001 | Software Engineering | BS |

* **courses**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **number** | **name** | **description** | **credithours** | **level** | **department\_code** |
| 113 | Spreadsheet | Microsoft Excel and Access | 3 | Undergraduate | 401 |
| 311 | Algorithm | Design and Analysis | 3 | Undergraduate | 401 |
| 531 | Theory of Computation | Theorem and Probability | 3 | Graduate | 401 |
| 363 | Database | Design Principle | 3 | Undergraduate | 401 |
| 412 | Water Management | Water Management | 3 | Undergraduate | 404 |
| 228 | Special Topics | Interesting Topics about CE | 3 | Undergraduate | 403 |
| 114 | Calculus | Limit and Derivative | 4 | Undergraduate | 402 |

* **register**

|  |  |  |  |
| --- | --- | --- | --- |
| **snum** | **course\_number** | **regtime** | **grade** |
| 1001 | 363 | Fall2015 | 3 |
| 1005 | 412 | Spring2015 | 4 |

1. (10 points) Please show the results for the following relational algebra expressions
2. (8 points) Please write the relational algebra expressions and sql code for the following queries.
   1. The course numbers and names of all courses offered by Randy’s home department
   2. The college(s) that have student registered.
3. (15 points) For each of the following schedules, determine what schedule(s) it is. Please explain briefly
4. S3 is a serial schedule. Yes[ ] No[ ]
5. S3 is a strict schedule. Yes[ ] No[ ]
6. S3 is a serializable schedule. Yes[ ] No[ ]

T2

R (B)

commit

T1

R(A)

R(B)

W(A)

commit

S3

T3

R(A)

commit

1. S2 is a serial schedule. Yes[ ] No[ ]
2. S2 is a strict schedule. Yes[ ] No[ ]
3. S2 is a serializable schedule. Yes[ ] No[ ]

T2

R(A)

W(B)

commit

T1

R(A)

W(A)

commit

S2

1. S1 is a serial schedule. Yes[ ] No[ ]
2. S1 is a strict schedule. Yes[ ] No[ ]
3. S1 is a serializable schedule. Yes[ ] No[ ]

T2

W(B)

commit

T1

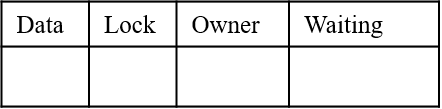
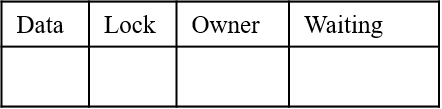
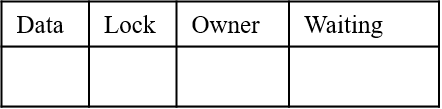
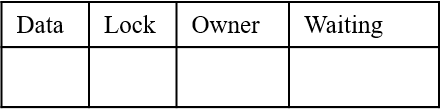
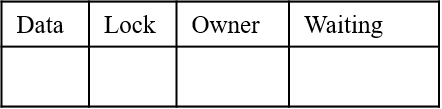
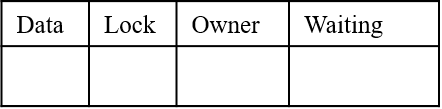
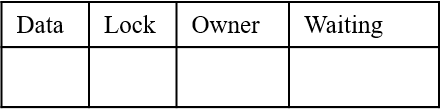
R(A)

W(A)

commit

S1

1. (15 points) Strict 2PL uses a lock table to track 1) which data object is locked; 2) which transaction owns which lock; 3) which transaction is waiting for a lock on which object. Consider the following two transactions. For each step of their execution, write down the status of the corresponding lock table. An execution can result in deadlock. When this happens, you write own “deadlock” and ignore the rest steps.



1) T1.X(A)

7) T1.commit

6) T2.R(A)

5) T1.commit

4) T1.W(A)

3) T2.S(A)

2) T1.R(A)

T2

S(A)

R(A)

commit

T1

X(A)

R(A)

W(A)

commit

S1

1. (8 points) Draw the Precedence Graph for the following schedules. Are they conflict serializable?

T2

R (B)

commit

T1

R(A)

R(B)

W(A)

commit

S1

T3

R(A)

commit

T2

R (B)

W(B)

commit

T1

R(A)

R(B)

W(A)

commit

S2

T3

R(A)

W(A)

commit

1. (9 points) For better current execution, multiple-granularity locking (MGL) introduces three new types of locks, IS, IX, and SIX. Consider the following tree of objects, where each node contains all its children.
2. (3 points) Transaction T1 needs to read object r1. What lock(s) on which node(s) does it need to acquire?
3. (3 points) Transaction T2 needs to write object r4 and r5. What lock(s) on which node(s) does it need to acquire?
4. (3 points) Transaction T3 needs to read r6 and r7. What lock(s) should it acquire on object p3?
5. (20 points) Database for COVID-19

We are in need of a database for COVID-19 reports that can answer the following questions

For each day, in each county in IA

1. How many tests are conducted?
2. How many additional confirmed cases?
3. How many additional hospitalized patients?
4. How many additional death?
5. How many additional recovered patients?
6. How many additional discharged patients?

We also need to report the demographic information of those patients including age group, gender, ethnicity, and race. As outbreaks occurred in long-term care facilities, it’s better to know which patients are related with LTC (staff/resident) as well.

This database will have three parties of users: test labs, hospitals, and the government. The test labs will handle all COVID-19 tests. They will enter the test results for every test and the basic demographic information of the testees. A person can get multiple tests. If a person is tested positive, then it is a confirmed case. If a confirmed patient is later tested negative, then this patient is recovered. The hospitals will update their patients’ status (hospitalized/discharged/death). Hospitals also need to query the test labs for the patients’ test results. The government will query the database to update the daily reports: for example <https://coronavirus.iowa.gov/>.

Q1 (4 pts). Please design a database and explain your design (also any additional assumptions). You can get extra credits if your design is more comprehensive than the basic needs described above. Your design should not require test labs or hospitals to provide summarized numbers everyday so that they can focus on their jobs.

Q2 (4 pts). Please provide examples SQL codes to the test labs and hospitals of how to insert and update records in your designed database.

Q3 (4 pts). Please provide SQL codes for finding the county in Iowa which has the highest number of confirmed patients who are elderly and LTC residents based on your database.

Q4 (4 pts). The hospitals will submit many queries for the test results of their patients, and test labs will update the test results very frequently. Please explain what you plan to choose for file format and indexing to improve the query responding time?

Q5 (4 pts). Different users may access your database concurrently. Please explain how will you handle concurrency?