CSCI-585 - Database Systems
Exam #2 - SOLUTIONS
12/5/14, 6-8 pm

Name:

Student ID:

Please read each question carefully before answering. The space provided for the answers should be adequate.

This is a closed-book, closed-notes, closed-devices, closed-peeking (but open mind!) test. If you are caught cheating or discovered to have cheated in any way, your score for the entire test will be 0.

Good luck, have fun!

Your score

Q1:	/10
Q2:	/10
Q3:	/10
Q4:	/10
Q5:	/10
Q6:	/10
Q7:	/10
Q8:	/10
Q9:	/10
Q10:	/10
Bonus:	/5
Total:	/105

Q1, SQL (5+5=10 points).

Assume that we have a database containing data about students, their transcripts, and courses offered by the CS department:
Student (Id, SName, Address, GPA)
Transcript (StudId, CrsId, Year, Grade)
CSCourse (CrsId, CrsName, CreditHours)

a. How many students have the max GPA recorded in the database?

```
SELECT COUNT(*) FROM Student
WHERE GPA= ALL (SELECT max(GPA) FROM Student);
```

b. How would you get a list of students who have taken all the CS courses? Note - all the fields of such students need to be shown.

```
SELECT S.* FROM student S WHERE
(SELECT CrsId FROM Course) IN (SELECT CrsId FROM transcript t WHERE S.id = t.StudId) ;
```

Q2, Spatial DBs (5*2=10 points). Circle T or F below, for true/false.

a. During searching for an object in an R-tree, one might need to search the entire database. **T/F - FALSE**

b. Z-order and kd-trees are two examples of index structures suitable for spatial data. **T/F - TRUE**

c. Quad trees have an advantage over kd-trees in that they keep the tree balanced therefore keeping the search efficient at all times. **T/F - FALSE**

d. kd-trees are only suitable for indexing of data in two dimensional space. **T/F - FALSE**

e. Quad-trees are only suitable for indexing of data in two dimensional space. **T/F - TRUE**

Q3, Spatial DBs (10 points). Describe the main purpose of using a minimum bounding rectangle (MBR) in spatial indexing structures such as R trees and R+ trees?

The minimum bounding box is most predominantly used as an **approximation object** in spatial indexes like the R-Tree and its variants. It is also used quite frequently in the areas of **object intersection and collision detection**. Minimum bounding boxes are also designed to be a general, intuitive approach to approximating a complex object.

Q4, XML (4*2.5= points). List FOUR differences between a DTD and an XML Schema ("xsd")?

1. XML schemas are written in XML while DTD are derived from SGML syntax.
2. XML schemas define data types for elements and attributes while DTD doesn't support data types.
3. XML schemas support restriction to be specified for element while DTD does not.
4. XML schemas allow support for namespaces while DTD does not.
5. XML schemas define number and order of child elements, while DTD does not.
6. XML schemas are extensible while DTD is not extensible.
7. XML Schemas are richer and more powerful than DTDs

Q5, XQuery (10 points)

Consider the following XML db:

```
<airport_management>
    <airport>
        <airportId>LAX</airportId>
        <airportName>Los Angeles</airportName>
    </airport>
    <airport>
        <airportId>LHR</airportId>
        <airportName>London Heathrow</airportName>
    </airport>
    <airport>
        <airportId>AMS</airportId>
        <airportName>Amsterdam</airportName>
    </airport>
    <airport>
        <airportId>JFK</airportId>
        <airportName>New York</airportName>
    </airport>

    <flight>
        <flightId>LA123</flightId>
        <seats>80</seats>
        <departureDate>12-24-2014</departureDate>
        <departureTime>12:00</departureTime>
        <arrivalDate>12-24-2014</arrivalDate>
        <arrivalTime>14:30</arrivalTime>
        <source>LHR</source>
        <destination>AMS</destination>
    </flight>
    <flight>
        <flightId>LL123</flightId>
        <seats>200</seats>
        <departureDate>12-24-2014</departureDate>
        <departureTime>13:00</departureTime>
        <arrivalDate>12-25-2014</arrivalDate>
        <arrivalTime>10:30</arrivalTime>
        <source>LAX</source>
        <destination>LHR</destination>
    </flight>
    <flight>
        <flightId>AL123</flightId>
        <seats>150</seats>
        <departureDate>12-24-2014</departureDate>
        <departureTime>12:00</departureTime>
        <arrivalDate>12-24-2014</arrivalDate>
        <arrivalTime>16:30</arrivalTime>
        <source>AMS</source>
        <destination>LAX</destination>
    </flight>
    <flight>
        <flightId>JL123</flightId>
        <seats>100</seats>
        <departureDate>12-24-2014</departureDate>
        <departureTime>08:00</departureTime>
        <arrivalDate>12-24-2014</arrivalDate>
```

```
        <arrivalTime>15:00</arrivalTime>
        <source>JFK</source>
        <destination>LAX</destination>
    </flight>
    <flight>
        <flightId>LJ123</flightId>
        <seats>120</seats>
        <departureDate>12-24-2014</departureDate>
        <departureTime>06:45</departureTime>
        <arrivalDate>12-24-2014</arrivalDate>
        <arrivalTime>15:50</arrivalTime>
        <source>LAX</source>
        <destination>JFK</destination>
    </flight>
</airport_management>
```

What would be the output of the following XQuery [that is applied to the db shown above]?

```
<result>
{
let $xmlFilePath := "airport.xml"
    let $results := for $a in doc($xmlFilePath)/airport_management/airport
        let $c := count(doc($xmlFilePath)/airport_management/flight[source = $a/airportId or destination = $a/airportId])
        order by $c descending
        return $a
    for $r in $results [position() = 1]
    return $r/airportId/text()
}
</result>
```

<result>LAX</result>
(The query is finding the busiest airport based on the number of departures and arrivals)

Q6, Normalization (3+3+4=10 points).
Assume that a video library maintains a database of movies rented out, in the following format - all information is stored in a single table!

Name	Address	Rented Movies	Salutation	Genre

Jan Martinez	456 Third St.	Forgetting Sarah Marshal, Daddy’s Little Girl	Dr.	Romance, Romance
Karl Philips	784 Torrance Blvd.	Pirates of The Caribbean, Clash of Titans	Mr.	Action, Action
Karl Philips	34 Forth St.	Clash of Titans, Frozen	Dr.	Action, Animated

Normalize the above table using 1NF, 2NF, and 3NF - **show** the resulting tables.

1NF :

Name	Address	Rented Movies	Salutation	Genre
Jan Martinez	456 Third St.	Forgetting Sarah Marshal	Dr.	Romance
Jan Martinez	456 Third St.	Daddy’s Little Girl	Dr.	Romance
Karl Philips	784 Torrance Blvd.	Pirates of Caribbean	Mr.	Action
Karl Philips	784 Torrance Blvd.	Clash of Titans	Mr.	Action
Karl Philips	34 Forth St.	Clash of Titans	Dr.	Action
Karl Philips	34 Forth St.	Frozen	Dr.	Animated

2NF:

Table 1

ID	Name	Address	Salutation
101	Jan Martinez	456 Third St.	Dr.
102	Karl Philips	784 Torrance Blvd.	Mr.
103	Karl Philips	34 Forth St.	Dr.

Table 2

ID	Rented Movies
101	Daddy’s Little Girl
101	Forgetting Sarah Marshal
102	Pirates of Caribbean
102	Clash of Titans
103	Clash of Titans
103	Frozen

3NF:

Table 1

ID	Name	Address	Salutation Id
101	Jan Martinez	456 Third St.	1
102	Karl Philips	784 Torrance Blvd.	2
103	Karl Philips	34 Forth St.	3

Table 2

ID	Rented Movies
101	Daddy’s Little Girl
101	Forgetting Sarah Marshal
102	Pirates of Caribbean
102	Clash of Titans
103	Clash of Titans
103	Frozen

Table 3

Salutation Id	Salutation
1	Dr.
2	Mr.
3	Ms.
4	Mrs.

Q7, Transaction Mgmt (5+5=10 points).

MyViterbi 'D-clearance' systems will have large amount of client requests when it comes to choosing courses. Consider a schedule based on the following log:

Operation Log	Operation
T1 STARTS T1 reads item CSCI585 T1 writes CSCI585: old value 55 => new value 56 T2 STARTS T2 reads item CSCI585 T2 writes CSCI585: old value 56 => new value 57 T3 STARTS T3 reads item CSCI570 T3 writes CSCI570: old value 42 => new value 43 T2 reads item CSCI570 T2 writes CSCI570: old value 43 => new value 44 T2 COMMITS T1 reads CSCI572 T1 writes CSCI572: old value 17 => new value 18 T3 COMMITS T1 COMMITS	

a. What serial schedule is this equivalent to?

T1->T2<-T3

b. Is this schedule consistent with two phase locking ("2PL")? If yes, **add** the fewest number of operations to the Operation column that will break the two phase locking. If not, **remove** operations from the Operation Log to make it consistent with two phase locking.

If we assume that all the transactions get the locks exactly before the operation and release them afterwards, it is not consistent with two phase locking. This is because T1 releases its lock on CSCI585 after its second operation while acquiring a lock on CSCI572 at its last two operations. By removing the last 2 operations of T1, it becomes 2PL

If we assume that the transactions get all locks they need at the beginning of the transaction and release them after the finish the operation, this schedule will be 2PL. The minimum operations that could be added to the schedule will be “T1 reads item CSCI570”. In this case, T1 has to acquire the lock on A again after releasing its lock on A after its first write (either is fine)

Q8, Query optimization (5*2=10 points)

Indexing is the not the only way to improve the 'search' performance of databases - another way is to tune SQL conditional expressions. List 5 ways of writing efficient conditional expressions.

1). Use simple columns or literals as operands
2). Numeric field comparisons are faster than character, date, and NULL comparisons
3). Equality comparisons are faster than inequality comparisons
4). Transform conditional expressions to use literals
5). Write equality conditions first when using multiple conditional expressions
6). When using multiple AND conditions, write the condition most likely to be false first
7). When using multiple OR conditions, put the condition most likely to be true first
8). Avoid the use of NOT logical operator

Q9, Distributed DBs (4*2.5=10 points)

In a distributed DB, there are three sites, A, B, C as shown below:

Note that there are communication lines between A and B and between B and C., but there is no direct line between A and C. The three sites cooperate to perform distributed transactions, using the two phase commit protocol. A is the coordinator.

Answer the following, with 'True' or 'False'. If true, just write 'True'; if false, write 'False' and **also explain why.**

a. If the communication is lost between A and B, A waits indefinitely in a blocked state until communication is restored.

False. A is the coordinator. If the communication stops before it receives the ‘vote to commit’ by all sites, A waits for a while but then it aborts and instructs all the participants to do the same.

b. If B receives “commit T” from A but cannot forward such instruction to C [eg. because C has crashed or communication between B and C is down], then B waits in a blocked state till everything Is restored after which C can be sent the instruction to commit.

False. B will just commit. Sites act on the instruction they received, not the instruction they send.

c. Assume that right after B receives “commit T” from A, the connections between A, B and B,C both go down. After a while, the B-C connection is restored but the A-B link is still down. B will remain in a blocked state until line with A is restored and will then ask A for instructions.

False. B commits for the same reason as 'b' above

d. Assume that B has finished “commit T” but C has not – at this point, both B and C crash. After B and C get restored, all the nodes are instructed to roll back transactions in order to avoid inconsistency.

False. Since B has already committed, what happens is that when the crashed nodes come back, the coordinator tell them to commit again.

Q10, “Big Data” (4*2.5 = 10 points)

What are the four characteristics of Big Data that we discussed in class? Describe each briefly.

a. Volume: Large volumes of data
b. Velocity: Quickly moving data
c. Variety: structured, unstructured, images, etc.
d. Value: the data must be useful.
e. Veracity: Trust and integrity is a challenge and a must and is important for big data just as for traditional relational DBs

Bonus (5 points)

Here is a folded piece of paper:

What letters can you form, by unfolding the above?

L and F [2.5 points for each]