

# CSCI585 Midterm exam

2017-03-03

Duration: 1 hour

Last Name: \_\_\_\_\_

First Name: \_\_\_\_\_

Student ID: \_\_\_\_\_

Email: \_\_\_\_\_

Hi there! There are 9 questions below (8 plus a bonus), one question per page. Please read each question carefully before answering. There's need to elaborate on anything, so you shouldn't need extra sheets.

The exam is **CLOSED** book/notes/devices/neighbors(!) but 'open mind' :) If you are observed cheating, or later discovered to have cheated in any manner, you will get a 0 on the test and also be reported to SJACS - so please don't! **DO YOUR OWN WORK.**

When we announce that the time is up, you NEED to stop writing immediately, and turn in what you have; if you continue working on the exam, we will not grade it (ie. you will get a 0). So please stick to the limit of one hour, use time wisely!

**Have fun, and good luck - hope you do well!**

Saty

**Q1 (4 points).**

Suppose an online vendor maintains its customer list like so:

| firstName | lastName | address        | city          | state | ZIP   | phoneNumber  | SkypeID     | emailAddress        |
|-----------|----------|----------------|---------------|-------|-------|--------------|-------------|---------------------|
| A         | B        | 123 Main St    | Los Angeles   | CA    | 90089 | 213-543-6543 |             | AB@mail.com         |
| Fam       | Act      | 222 Burton Way | Beverly Hills | CA    | 90210 |              | RichNFamous | RNF@imdb.com        |
| MoreFam   | Act      | 108 Roxbury St | Beverly Hills | CA    | 90210 | 323-654-1002 |             | TheBest@BevHills.us |
| Grad      | Student  | 154 Adams St   | Los Angeles   | CA    | 90089 |              |             | DontBugMe@usc.edu   |

**What two problems do you see with the above scheme, and how would you fix them?** Your answer can be in the form of E-R (using any notation), or in table format (like above) or even SQL. And, feel free to create any new attributes that might be necessary.

**Repetition of data, with city and state names (so make a separate table of these, with ZIP as the PK); missing (NULL) values for contact info (so, make a separate ContactInfo table with (ContactID, ContactType, ContactValue) rows and move the contact data there (no NULLS will exist because we will have a new row for each contact type a person has).**

**Q2 (4 points).** Parents in a wealthy family want to create a DB of all their assets. For each asset, they would like to name benefactors - some or all of their five children who would get the asset. Each asset has a financial value associated with it, and a maturity date (when the kid(s) can cash in). They'd like to track the following diverse set of assets they own: bank accounts, real estate, stocks, jewelry, life insurance. **What would be a good design (using an ER diagram) for this?** You can make any assumptions you want about the assets, create whatever descriptors (columns) you need, etc.

**Make a superclass entity Assets, and a Benefactors one, link them as 1:M. Under Assets, create BankAccounts etc. as subclass entities.**

**Q3 (3 points).** A reality company keeps track of its home sales like so:

| Seller | Buyer | LendingBank |
|--------|-------|-------------|
| S1     | B1    | BofA        |
| S2     | B1    | Chase       |
| S1     | B2    | Chase       |

Things seem fine (redundancy and all), until they hire you to 'clean up' their table. After analysis, you come up with these three separate tables [all linked properly with FK/PK], which makes for good design:

Table 'SellerBuyer', with rows such as (S1,B1).

Table 'BuyerBank', with (B2,Chase) as a sample row.

Table 'SellerBank', eg. with (S2,Chase) as a row.

You write the following three-way 'join' query just for fun, to see if you can recreate the original triplets (eg. S1,B1,BofA):

```
SELECT SB.Buyer, SN.Seller, BN.LendingBank
FROM SellerBuyer as SB, SellerBank as SN, BuyerBank as BN
WHERE BN.Buyer=SB.Buyer
AND BN.LendingBank=SN.LendingBank
AND SN.Seller=SB.Seller
```

**Question: what, if any, is the problem with the above query?**

**The query will result in correct triples such as (S1,B1,BofA) etc, but ALSO wrong ones such as (S1,B1,Chase) [because it will multiply all three tables].**

**Q4 (1+1=2 points).** You pull out your smartphone, log on to your banking app, and proceed to transfer \$7200 (to pay for a 4-unit 'SC course!) from your savings account into your checking account. Prior to the transfer, you had \$20,000 in savings and \$800 in checking. While you are in the middle of doing this, due to poor DB design, a report generator (that would produce a monthly statement to email you) runs on the bank's server. **What could go wrong, and what is such a scenario called?**

**If the report generator grabs the 'After' value of saving (\$12,800) and 'Before' value of checking (\$800), it will show our balance incorrectly as \$13,600 [instead of \$20,800]. This is an 'Inconsistent Retrieval'.**

**Q5 (2+2=4 points). How would you optimize (by rewriting) the following two queries?**

a. `SELECT * FROM TBL WHERE substr(STATE,1,1)='C'`

[we want to select all rows containing states CA, CO, or CT;  
substr(<string>,1,1) returns the first character of a string]

**WHERE STATE IN (CA,CO,CT) [or, can use OR]**

b. `SELECT * FROM TBL WHERE AGE>21`

[the AGE column stores ages as 0..99 integers; assume it has been indexed]

**WHERE AGE >=22 [the = will result in the index being used to fetch all entries that are >=22, no row-by-row comparison in the main table needed!]**

**Q6 (4 points).** In the world of relational DBs, the 'ACID' properties ensure that a DB always preserves data integrity. In the newer world of Internet-enabled, distributed DBs, there is instead 'BASE'. **What two essential features of a DB are traded off, in BASE?** Explain using an example (or two).

**Consistency (all copies of a fragment need to contain identical data), and Availability (a transaction should always be achievable, without 'downtime').**

**Q7 (4 points). What operation does the following SQL query implement?**

```
SELECT DISTINCT c
FROM TABLE_A as t1
WHERE EXISTS (SELECT *
              FROM TABLE_B as t2
              WHERE t1.c = t2.c);
```

**Finds the INTERSECTION of t1 and t2.**



**Q8 (5 points).**

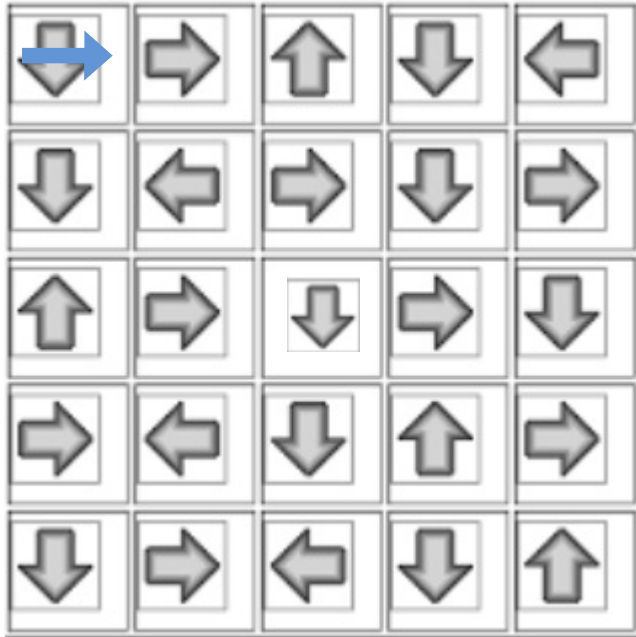
**What does the following query do** (:Name is simply a local variable)?

```
SELECT :Name, MAX(P1.reviewDate), P2.reviewDate
FROM EmpDB as P1, EmpDB as P2
WHERE P1.reviewDate<P2.reviewDate
      AND P1.EmpName=:Name
      AND P2.reviewDate = (SELECT(MAX(reviewDate) FROM
EmpDB)
GROUP BY P2.reviewDate;
```

**Finds the 2 latest reviews for an employee.**

**Bonus question (1 point).**

Complete the puzzle below..



**Trace a clockwise spiral from the top-left**, observe the sequence: down,right,up,down,left,right.. Repeat the sequence along our spiral path :) That makes the central square have a 'down' arrow.