CSE 414 Final Examination

Name: __________Solution__________

	Topic	Points	Max
I	Relational Data		30
II	DB Applications		36
III	Semistructured Data		12
IV	DBMS Implementation		48
V	Big Data		26
	Total		152

- The exam is closed book.
- No electronic devices may be used.
- You are allowed two letter-sized pages of notes (both sides, 8+ pt font).

- All questions have **one answer unless stated otherwise**.
- Questions with one answer are worth 2 points.
- Questions with many answers are worth 1 point per available option (4-6 points total).

- You have **1 hour and 50 minutes** to complete the test.
- If you are not done after 1 hour and 30 minutes, please **stay in your seat** until the end.

I. Relational Data

Consider the following schema for a simple social network. It maintains a collection of users, identified by unique “handles” (short names).

```
User(handle, name, home_city, bio)
Friend(handle1, handle2)
Message(handle, text, from_city)
```

The next few problems will consider queries for answering the following questions:

A. Find the handles of users who wrote messages while in cities other than the one in which they live.

B. Find the handles of users who are friends with someone who is friends with them.

C. Find the names of users who either live in Kansas City or have written a message from Kansas City.

D. Find the names of users who both live in Kansas City and have written a message from Kansas City.

After you have read these questions, turn the page….

(There are no problems on this page.)

For each **extended** relational algebra expression below, indicate which of the questions above (A – D), **if any**, it answers. If no question matches, then leave the space blank.

1. $\delta(\pi_{\text{handle1}}\ \sigma_{\text{handle1 = handle3}}\ [\text{Friend} \bowtie (\rho_{\text{handle2, handle3}}\ \text{Friend})])$ ________B_______

   (Recall that the ρ operator simply *renames* the columns of a table.)

2. $\delta(\pi_{\text{handle}}\ [\delta(\pi_{\text{handle, from\_city}}\ \text{Message}) - \pi_{\text{handle, home\_city}}\ \text{User}])$ ________A_______

3. $\delta(\pi_{\text{name}}\ [([\pi_{\text{handle}}\ \sigma_{\text{home\_city = 'Kansas City'}}\ \text{User}] \cup [\pi_{\text{handle}}\ \sigma_{\text{from\_city = 'Kansas City'}}\ \text{Message}]) \bowtie \text{User}])$ ________C_______

4. $\delta[\pi_{\text{handle}}\ \sigma_{\text{from\_city <> home\_city}}\ (\text{User} \bowtie \text{Message})]$ ________A_______

For each SQL query below, indicate which of the questions from before (A – D), **if any**, it answers. If no question matches, then leave the space blank.

5\.

```
SELECT name
FROM User U
WHERE home_city = ‘Kansas City’ OR
      EXISTS (SELECT name FROM Message
              WHERE handle = U.handle AND
                    from_city = ‘Kansas City’)
```

______C______

6\.

```
(SELECT name FROM User
 WHERE home_city = ‘Kansas City’)
INTERSECT
(SELECT name FROM User U, Message M
 WHERE U.handle = M.handle AND
       from_city = ‘Kansas City’)
```

______D______

7\.

```
SELECT DISTINCT U.handle
FROM User U, Message M
WHERE U.handle = Message.handle AND
      from_city <> home_city
```

______A______

8\.

```
SELECT F1.handle1
FROM Friend F1, Friend F2
WHERE F1.handle2 = F2.handle2 AND
      F1.handle1 = F2.handle1
GROUP BY F1.handle1
```

____

For each Datalog query below, indicate which of the questions from earlier (A –D), **if any**, it answers. If no question matches, then leave the space blank.

In each item, the rule of the form QN, where N is a number, defines the query in question.

9.
```
Q1(h) :- User(h, _, hc, _),
         Message(h, _, fc), hc <> fc.
```
_______A______

10.
```
Q2(h) :- User(h, _, "Kansas City", _).
Q2(h) :- User(h, _, hc, _),
         Message(h, _, fc), hc = "Kansas City".
```
____

11.
```
Q3(h) :- User(h, _, "Kansas City", _),
         Message(h, _, "Kansas City").
```
____

12.
```
FoF(h1, h2, 1) :- Friend(h1, h2).
FoF(h1, h2, d) := FoF(h1, h3, d-1),
                  Friend(h3, h2).
Q4(h) :- FoF(h, h, 2).
```
_______B______

13. Consider the following SQL query:

```
SELECT handle, bio, min(home_city)
FROM User NATURAL JOIN Message
GROUP BY handle, from_city
```

Which section of the query is illegal?

`handle` [✓ `bio`]

`min(home_city)` `from_city`

14. Consider the following expression in **standard** relational algebra:

$\pi_{\text{from\_city}}\,[\,\text{Message} \bowtie (\,\sigma_{\text{handle} > \text{‘c’}}\,\text{User} - \sigma_{\text{handle} > \text{‘d’}}\,\text{User}\,)]$

Can the same result be computed using a simple select-from-where query (i.e., one with no subqueries, grouping, or aggregation)?

[✓ Yes] No

15. Which of the following is an advantage of using an auto-generated, unique integer ID as the primary key for `User` instead of the handle? (Choose the <u>best</u> answer.)

uses much less disk space

makes searches by primary key much more efficient

[✓ makes handle changes much more efficient]

makes hash indexes an option instead of only B+ tree indexes

II. DB Applications

1. Which of the following are tuple constraints that can be automatically checked by a DBMS when tuples are changed or added to the table? Circle all that apply.

[✓ value in column is a string]	[✓ value in column is the key of a tuple in table T]
[✓ value in column is not null]	[✓ value in column is a positive integer]
[✓ value in column is unique in that column of the table]	[✓ values in two columns are different]

2. Suppose that `R(a,b,c)` and `S(c,d)` are relations. If R has a dependency `b` → `c`, then the same dependency **must** exist in:

$R \times S$:	[✓ Yes]	No
$R \bowtie S$:	[✓ Yes]	No

3. For each JDBC statement type below, to which method do you pass the string describing your SQL query? (Recall that `Statements` can execute multiple queries.)

`Statement:`	`createStatement`	[✓ `executeQuery`]
`PreparedStatement:`	[✓ `prepareStatement`]	`executeQuery`

4. Using SQL Server, if **one** JDBC `Connection` sets its isolation level to *read uncommitted*, only that connection may see behavior that is not serializable?

[✓ Yes] No

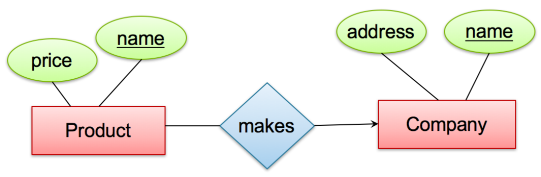
5. Which tier of a 3-tiered architecture is most likely to become the performance bottleneck as you scale to a very large number of users?

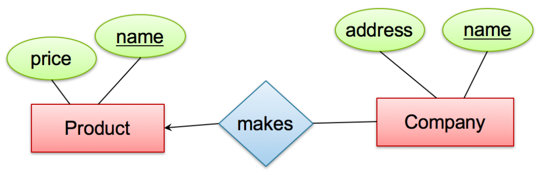
browsers	hard disks
web servers	[✓ DB server]

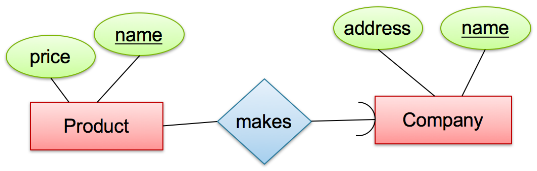
Consider the following SQL schema:

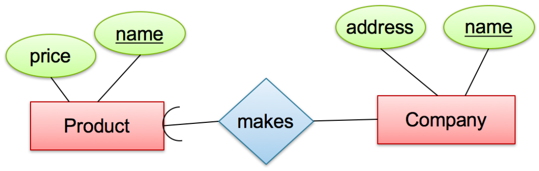
```
CREATE TABLE Company(
  name VARCHAR(100) PRIMARY KEY,
  address VARCHAR(200));

CREATE TABLE Product(
  name VARCHAR(100) PRIMARY KEY,
  price FLOAT,
  made_by VARCHAR(100) FOREIGN KEY REFERENCES Company);
```

6. Which of the following E/R diagrams could produce that schema?

[✓ ]







7. Which of the following describes the relationship between Products and Companies?

one to one | one to many

[✓ many to one] | many to many

8. Which of the following types of relationships between Products and Companies **could** be represented in SQL without an additional table? <u>Circle all that apply</u>.

[✓ one to one] | [✓ one to many]

[✓ many to one] | many to many

The next two questions refer to the schema for `User` and `Message` that we saw in Part I.

9. Which of the following changes to the data for a **single user** would, most likely, require changing more rows of the database?

   [✓ changing handle]   changing name

10. Which of the following would be the best way to reduce the number of rows that need to be changed the case above?

    add an index on `Message(handle)`   use `(handle, name)` as the primary key of `User`

    add an index on `Message(text)`   [✓ use an auto-generated, unique ID as the primary key of `User`]

Suppose that we changed the schema from earlier so that (1) the author's name is included (in addition to their handle) and (2) the city's zip code is included:

```
Message(handle, name, text, from_city, city_zip)
```

11. Suppose that we allowed users to change both their names and handles. Each item below gives one choice of the many names and handles a user has had that we could store with the message. In which situations, would `handle` → `name` in `Message` **still** be a functional dependency? Circle all that apply.

    current handle / name at time written   [✓ current handle / current name]

    handle at time written / name at time written   first handle of author / current name

12. Which of the following could result from a BCNF decomposition of this table? Circle all that apply.

    `City(name, zip)`   `Message(name, text, from_city)`

    [✓ `Author(handle, name)`]   [✓ `Message(handle, text, from_city)`]

13. Which of these would make sense to add to the schema? Circle all that apply.

    City   Author

III. Semistructured Data

Suppose that we store all the data for our social network in a single dataset of Users:

```
[{“handle”: “biebs”, “name”: “Justin Bieber”,
  “home_city”: “Somewhere, Canada”, “bio”: “…”,
  “friends”: [“kimkardashian”, “shaq”, …],
  “messages”: [
    {“text”: “:-* :-* :-* :-*”, “from_city”: “Los Angeles, CA”},
    {“text”: “New. Music. Friday.”, “from_city”: “Los Angeles, CA”},
    … ]}
  … ]
```

Consider the following SQL++ query:

```
SELECT DISTINCT m.from_city, n.from_city
FROM Users u, u.messages m, Users v, v.messages n
WHERE u.handle <> v.handle AND m.text = n.text
```

1. Which of the SQL queries below return the **same** result as the SQL++ query above? (Each SQL query uses the same schema for User and Message as we saw in Part I.) Circle all that apply.

```
SELECT DISTINCT m.from_city, n.from_city
FROM User u, Message m, User v, Message n
WHERE u.handle = m.handle AND v.handle = m.handle AND
      u.handle <> v.handle AND m.text = n.text
```

```
SELECT DISTINCT m.from_city, n.from_city
FROM Message m, Message n
WHERE m.handle <> n.handle AND m.text <> n.text
```

```
SELECT m.from_city, n.from_city
FROM Message m, Message n
WHERE m.handle <> n.handle AND m.text <> n.text
GROUP BY m.from_city, n.from_city
```

```
SELECT DISTINCT m.from_city, n.from_city
FROM Message m, Message n, User v
WHERE m.handle <> v.handle AND m.text = n.text
```

Consider the following SQL++ query:

```
SELECT max(coll_count(m))
FROM User u, u.messages m
WHERE m.text > 3 AND u.handle = “biebs”
ORDER BY u.name
```

2. Which parts of the query above are illegal? <u>Circle all that apply</u>.

    [✓ `coll_count(m)`]   `ORDER BY u.name`

    `u.handle = “biebs”`   [✓ `m.text > 3`]

Consider the following SQL++ query:

```
SELECT u.handle, count(*)
FROM User u, u.messages m
WHERE u.home_city = ‘Los Angeles, CA’
GROUP BY u.handle
```

3. Which of the following SQL++ queries return the **same** result? <u>Circle all that apply</u>.

```
SELECT u.handle, coll_count(u.messages)
FROM User u WHERE u.home_city = ‘Los Angeles, CA’
```

```
SELECT u.handle, coll_count((SELECT * FROM u.messages))
FROM User u WHERE u.home_city = ‘Los Angeles, CA’
```

4. Which of the following SQL queries return the **same** result? <u>Circle all that apply</u>.

[✓
```
SELECT u.handle, count(*)
FROM User u, Message m
WHERE u.handle = m.handle AND u.home_city = ‘Los Angeles, CA’
GROUP BY u.handle
```
]

[✓
```
SELECT u.handle, count(*)
FROM User u NATURAL JOIN Message
WHERE u.home_city = ‘Los Angeles, CA'
GROUP BY u.handle
```
]

IV. DBMS Implementation

1. Suppose that A = 100, transaction T1 wants to increase A by 100, and T2 wants to double A. Consider the schedule $R_1(A)$, $R_2(A)$, $W_1(A)$, $W_2(A)$. Which of the following properties hold for this schedule? Circle all that apply.

   [✓ transactions are isolated] [✓ serializable]

   follows strict 2PL locking conflict serializable

2. Suppose that T1 and T2 are accessing element B in a **globally-distributed** database from opposite sides of the planet. Which of the following is most difficult to support?

   $R_1(B)$ before $R_2(B)$ $R_1(B)$ before $W_2(B)$

   [✓ $W_1(B)$ before $R_2(B)$] $W_1(B)$ before $W_2(B)$

3. In the absence of rollbacks, which of the following ensures that locking result in a serializable schedule? Circle all that apply.

   [✓ two-phase locking] using read and write locks

   [✓ strict two-phase locking] using read, pending, reserved, and exclusive locks

4. Which of the ACID properties are most at risk of being violated if the DBMS crashes mid-way through committing results to disk. Circle two.

   [✓ atomicity] consistency

   isolation [✓ durability]

5. In the absence of rollbacks, which of the following can be used to determine whether a schedule is conflict serializable? Circle all that apply.

   [✓ check for cycles in precedence graph]

   [✓ simulate a 2PL scheme and check for deadlocks]

   [✓ simulate strict 2PL and check for deadlocks]

   see if you can to change the schedule into a serial order by swapping adjacent elements

6. If adding more clients is decreasing the rate of transactions per second, which of the following techniques will likely increase the rate with the least possibility of bugs?

read committed isolation level	add more web servers
read uncommitted isolation level	[✓ admission control]

7. Suppose we are computing R ⋈ S, with both tables large (1M+ rows). The estimated cost of a block nested loop join is approximately cut in half if (<u>circle all that apply</u>):

number of tuples of R halved with number of blocks unchanged	[✓ tuples per block of R is doubled with number of tuples unchanged]
number of tuples of S is halved with number of blocks unchanged	[✓ tuples per block of S is doubled with number of tuples unchanged]

8. Compared to regular (tuple-based) nested loop join, the estimated cost of a block nested loop join is:

always faster	never faster
sometimes faster, sometimes slower	[✓ never slower]

9. If each block contains only a single tuple, then compared to a block nested loop join, the estimated cost of an indexed join is:

always faster	never faster
sometimes faster, sometimes slower	[✓ never slower]

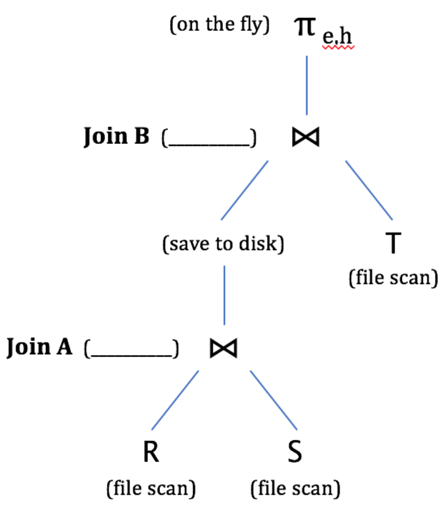
10. Suppose that the primary key of table is an auto-generated, UUID with seemingly random values. Compared to an unclustered index on the primary key, the clustered index created automatically by SQL server makes each query:

always faster	always slower
sometimes faster, sometimes slower	[✓ always the same]

In the next few problems, we will compute the cost of the query plan on the right, which finds the result of the expression $\pi_{e,h}$ (R ⋈ S ⋈ T) over the tables R(e,f), S(f,g), and T(g,h). (Both joins natural.)

The plan on the right is still *incomplete*, however, because the choice of algorithm for two parts, **Join A** and **Join B**, have not yet been filled in.

In the next few problems (starting on the next page), we will consider the cost of this plan with different choices of algorithms for these two points.



You can assume the following statistics about these tables:

Table	# tuples	# blocks
R	1,000	100
S	5,000	200
T	100,000	10,000
R ⋈ S	5,000	20

Furthermore, you can assume the following statistics about their columns:

Column	# distinct	Low	High
R.f	100	1	1000
S.f	1000	1	2000
S.g	5000	1	2,000
T.g	1000	1	10,000

Useful Formulas

Estimated cost of X join Y:

- Using a block nested loop, the cost is B(X) + B(X) B(Y), where B(X) is # blocks in X.
- Using a clustered index on Y(A), the cost is B(X) + T(X) B(Y) * E, where T(X) is # tuples in X, and E is the selectivity of the condition A = c.
- Using an unclustered index on Y(A), the cost is B(X) + T(X) T(Y) * E.

Estimated selectivity of conditions:

- For A = c, the selectivity is 1 / (# distinct values of A)
- For A < c, the selectivity is (c - lowest value of A) / (highest - lowest value of A)

11. What is the estimated cost of **Join B** in the plan if we implement it with a block nested loop join?

20 + 20 * 10,000 = **200,020**

12. What is the estimated cost of **Join B** the if we use an indexed join? Assume that we have a clustered index on `T(g)`.

20 + 5,000 * 10,000 / 1000 = **50,020**

13. What is the estimated cost of **Join A** in the plan if we implement it with a block nested loop join?

100 + 100 * 200 = **20,100**

14. What is the estimated cost of **Join A** if we use an indexed join? Assume that we have an unclustered index on `S(f)`.

100 + 1,000 * 5,000 / 1000 = **5,100**

15. What is the total cost of this plan if we use the best choice of join algorithm for A and B?

In addition to the two joins above, we need to write R ⋈ S to disk, which costs 20 IOs. The final projection has no cost. Thus, the total cost is 55,140.

You are implementing a DB application using SQL Server. At first, as you add more users, you see a nice increase in completed transactions per second. However, after reaching a high level of simultaneous users, you find that completed transactions per second starts *dropping* as you add more users.

You determine that the problem is SQL Server's **row-level locking**. To improve performance, you would like to run some queries using table-level locking instead.

Your idea is as follows. You create a special table:

```
    Tables(name VARCHAR(100), last_write_by INT);
```

It will contain one row for each table, indicating the name of that table and the number of the transaction that last wrote to it. Each transaction will start by reading the rows of `Tables` for the tables that its wants to read and writing the rows for the tables that it wants to write, which will require Sql Server to acquire read or write locks on those rows. In that manner, the row-level locks in SQL Server `Tables` for act as table-level locks.

16. After the transaction has acquired the appropriate table locks and it executes its operations against the actual tables, will SQL Server still acquire locks on the rows of those tables?

    [✓ Yes] No

17. To improve performance, you decide to use a second DB connection to execute the query once you have acquired the table locks using your primary DB connection. What is the **weakest** isolation level that you can safely use for this second connection if you still want your transactions to be ACID?

    [✓ read uncommitted] read committed

    repeatable read serializable

18. Which additional changes would make deadlock impossible in this system?

    none (already impossible) take write locks before read locks

    [✓ lock tables in alphabetical order] take read locks before write locks

19. Suppose that SQL Server used read/write locks where a read lock is *always granted* provided *no write lock has yet been acquired*. In **one sentence**, describe a problem with this approach that would be fixed by switching to SQLite's scheme with 4 lock types.

    Writers could be waiting forever for a time with no read locks.

V. Big Data Systems

1. Which features do parallel DBs provide that dataflow engines don't? Circle all that apply.

[✓ declarative queries]	scale to petabytes of data
[✓ transactions]	compute results of non-monotone queries

2. The purpose of a watermark in a streaming system is to control what?

[✓ which data can be ignored]	batch size
how quickly results must be updated when new data arrives	number of MapReduce jobs executed

3. The "shuffle sort" phase of MapReduce is most like which operation in SQL?

inner join	[✓ group by]
outer join	count (*)

4. The physical plans in Spark SQL add what element beyond what is in the logical plans?

where to acquire locks	[✓ choice of broadcast vs parallel join]
where to write to disk	choice of hash vs B+ tree indexes

5. A dataflow pipeline with T "reshuffle" operations can be executed with *at most* how many MapReduce jobs?

T-1	[✓ T]
T+1	2T

You are hired by a web company that makes a popular product with over 100 million users. To achieve that scale, they built their app on top of a NoSQL system. However, they found that it is hard to answer complex questions about the data stored in that system.

The CEO has now hired you to fix this problem. She tells you that you can hire a team of programmers to help you, but you cannot change away from the NoSQL system they are already using — that would be too costly.

Thankfully, the NoSQL system they use, an extensible record system (or "Big Table"), does support MapReduce. (Unfortunately, the system provides you with **no statistics** about the columns in the records.) After thinking it over, you decide to build a query engine that lets employees execute SQL queries over the data by turning SQL into MapReduce jobs.

6. Which of the following would your team **not** need to implement in the code that you write for this system? Circle all that apply.

parser for SQL	conversion from SQL to RA
[✓ estimate selectivity of conditions in where clause]	put RA operators into groups that require only one reshuffle
[✓ take locks on data elements according to two-phase locking]	[✓ start backup jobs for any stragglers]

7. Which of the following would be the best estimator of the cost of a physical plan in this system?

total CPU cycles	total CPU cycles per node
total network I/O	[✓ number of reshuffles]

8. Suppose that you wanted to maintain statistics about the columns. Your idea is to do that by adding a record, for each column, that stores its statistics. Which of the following features would you require of the NoSQL system? Circle all that apply.

[✓ row-level transactions]	[✓ general transactions]
two-phase locking	strict two-phase locking

9. Do some NoSQL systems offer that feature?

[✓ Yes] No