NAME: **SEAT NO.:**

STUDENT ID:

Midterm Exam #2 (Version A)
CS 122A
Winter 2017

Max. Points: 100
(Please read the instructions carefully)

Instructions:
- The total time for the exam is 50 minutes; be sure to *budget your time* accordingly.
- The exam is closed book and closed notes but "open cheat sheet".
- Read each question first, in its entirety, and then carefully answer each part of the question.
- If you don't understand something, ask one of the exam proctors for clarification.
- If you still find ambiguities in a question, note the interpretation you are taking.

QUESTION	***TOPIC***	***POINTS***	***SCORE***
1	**True or False?**	**20**	
2	**Query Matching**	**20**	
3	**Query Writing**	**20**	
4	**SQL Execution**	**30**	
5	**Short Answers**	**10**	
TOTAL	**All**	**100**	

Question 1: True or False? (20 points)

(2 pts each) For each of the following statements, indicate whether the statement is true (circle TRUE) or false (circle FALSE):

- Views can be queried just as if they were stored tables.

  TRUE FALSE

- Views based on joining two or more tables together are always updatable.

  TRUE **FALSE**

- SQL is a less powerful language than the relational algebra.

  TRUE **FALSE**

- SQL/PSM (the SQL standard's stored procedure extension) is more powerful than SQL.

  TRUE FALSE

- The relational algebra and the safe subset of the tuple relational calculus are equivalent in their expressive power.

  TRUE FALSE

- Given a relational schema, there is only one way to translate a given English query against that schema's data into a correct SQL query.

  TRUE **FALSE**

- Triggers in SQL are less powerful than foreign key constraints.

  TRUE **FALSE**

- To be union-compatible (e.g., combinable with UNION or UNION ALL), two SQL tables or sub-queries must have the same number of columns with the same data types and the same column names.

  TRUE **FALSE**

- The set operation INTERSECT is not actually necessary in SQL because it is always possible to express an equivalent query by appropriately using a JOIN operation instead.

  TRUE FALSE

- To express universal quantification (for all) queries, the SQL language includes a DIVISION operation modeled after the corresponding relational algebra operator.

  TRUE **FALSE**

SCORE: _____

Question 2: Query Matching (20 points)

(20 pts) Consider our old favorite Hoofers Sailing Club example database, the schema for which is sketched below:

Sailors(sid, sname, rating, age)
Reserves(sid, bid, date)
Boats(bid, bname, colo

Here is a list of query meanings to choose from when answering the following questions:

Q1 – Print the maximum age of sailors who have at least one boat reserved.

Q2 – Print the maximum age of sailors.

Q3 – Print the minimum age of sailors.

Q4 – Not a legal SQL query.

Q5 – None of the meanings listed above!

(4 pts each) For each of the following SQL queries, indicate the meaning of the query in the space to its right by choosing the appropriate answer from the list of meanings (one of Q1 through Q5) given above. Note that you can (and may need to) use the same answer multiple times!

SQL QUERY:	**MEANING:**
SELECT MAX(age) FROM Sailors S, Reserves R WHERE S.sid = R.sid;	____Q1____
SELECT MAX(age) FROM Sailors S LEFT OUTER JOIN Reserves R ON (S.sid = R.sid);	____Q2____
SELECT S.age FROM Sailors S ORDER BY S.age ASC LIMIT 1;	____Q3____
SELECT MAX(age) FROM Sailors S WHERE EXISTS (SELECT * FROM Boat);	____Q2____

SCORE: _____

```
SELECT S.age
FROM Sailors S
WHERE MIN(S.age);
```

____Q4____

Question 3: Query Writing (20 points)

(20 pts) Consider again the Hoofers Sailing Club database, and suppose (if needed or helpful) that the tables contain the data shown below.

Sailors(sid, sname, rating, age)
Reserves(sid, bid, date)
Boats(bid, bname, color)

sid	sname	rating	age
1	Bob	5	35
2	Sally	9	22
3	Zack	10	19
4	Abby	3	*null*
5	Joe	*null*	*35*

sid	bid	date
1	4	2017-03-15
1	5	2017-04-15
3	2	2017-04-15
4	4	2018-01-01
5	1	2017-12-25

bid	bname	color
1	Interlake	red
2	Sunfish	yellow
3	Clipper	green
4	Yacht	green
5	Sunfish	yellow
6	Clipper	red

Write the following queries in the indicated language:

(10 pts) ***Relational Algebra:***
Print the bids of Boats that are reserved by both a 35-year-old sailor with a rating of 5 or more and a sailor with a rating of less than 5.

π bid ((σ age = 35 $\wedge$ rating >= 5 (Sailor)) $\bowtie$ Reserves) $\cap$ π bid ((σ rating < 5 (Sailor)) $\bowtie$ Reserves))

(10 pts) ***SQL:***
For each popular boat color, print the color along with its total number of reservations and the number of *different* days on which boats of that color are reserved. (A color is said to be a popular boat color if 2 or more reservations exist for boats of that color.)

```
SELECT b.color, count(*) as cnt, count(DISTINCT r.date)
FROM Boats b, Reserves r
WHERE b.bid = r.bid
GROUP BY b.color
HAVING cnt >=2;
```

SCORE: _____

Question 4: SQL Execution (30 points)

Consider yet again the Hoofers Sailing Club database containing the data shown below:

Sailors(sid, sname, rating, age)
Reserves(sid, bid, date)
Boats(bid, bname, color)

sid	sname	rating	age
1	Bob	5	35
2	Sally	9	22
3	Zack	10	19
4	Abby	3	*null*
5	Joe	*null*	*35*

sid	bid	date
1	4	2017-03-15
1	5	2017-04-15
3	2	2017-04-15
4	4	2018-01-01
5	1	2017-12-25

bid	bname	color
1	Interlake	red
2	Sunfish	yellow
3	Clipper	green
4	Yacht	green
5	Sunfish	yellow
6	Clipper	red

In addition, assume that the following additional SQL goodies have already been defined:

```
DELIMITER $$
CREATE PROCEDURE AddSailor(sname varchar(20), age float)
BEGIN
   DECLARE new_sid INT(11);
   SET          new_sid = (SELECT MAX(sid)+100 FROM Sailors);
   INSERT INTO Sailors(sid, sname, rating, age)
        VALUES (new_sid, sname, 1, age);
END$$

CREATE TRIGGER someTrigger
AFTER DELETE ON Sailors FOR EACH ROW
BEGIN
    UPDATE Reserves
    SET sid = NULL
    WHERE sid = OLD.sid;
END$$
DELIMITER ;
```

(a) (10 pts) Show below the result of running the following query against the tables above:

```
SELECT S.sname, S.age, B.bname, B.color
FROM (Sailors S LEFT OUTER JOIN Reserves R ON (S.sid = R.sid))
              JOIN Boats B ON (R.bid = B.bid)
WHERE S.age = 35.0 AND S.rating >= 3;
```

3pts

(Bob, 35, Yacht, green)
(Bob, 35, Sunfish, yellow)

SCORE: _____

Question 4: SQL Execution (*continued*)

Here's all of the database information once again:

Sailors(sid, sname, rating, age)
Reserves(sid, bid, date)
Boats(bid, bname, color)

sid	sname	rating	age
1	Bob	5	35
2	Sally	9	22
3	Zack	10	19
4	Abby	3	*null*
5	Joe	*null*	*35*

sid	bid	date
1	4	2017-03-15
1	5	2017-04-15
3	2	2017-04-15
4	4	2018-01-01
5	1	2017-12-25

bid	bname	color
1	Interlake	red
2	Sunfish	yellow
3	Clipper	green
4	Yacht	green
5	Sunfish	yellow
6	Clipper	red

```
DELIMITER $$
CREATE PROCEDURE
   AddSailor(sname varchar(20), age float)
BEGIN
   DECLARE new_sid INT(11);
   SET  new_sid = (SELECT MAX(sid)+100 FROM Sailors);
   INSERT INTO Sailors(sid, sname, rating, age)
        VALUES (new_sid, sname, 1, age);
END$$
```

```
DELIMITER $$
CREATE TRIGGER someTrigger
AFTER DELETE ON Sailors FOR EACH ROW
BEGIN
   UPDATE Reserves
   SET sid = NULL
   WHERE sid = OLD.sid;
END$$
DELIMITER ;
```

(b) (10 pts) Show below the results of executing the following statement against the tables above. (I.e., show the contents of the above tables after execution; you may skip the tables that stay the same.)

```
CALL AddSailor('Sandy', 20.0);
```

3pts

(105, Sandy, 1, 20)

(c) (10 pts) Show below the results of executing the following statement against the initial tables above. (I.e., show the contents of the above tables after execution; you may skip the tables that stay the same.)

```
DELETE FROM Sailors S WHERE S.sname LIKE '%a%';
```

Sailor: 4 pts

1, Bob, …
4, Abby, …
5, Joe, …

SCORE: _____

Reserves: 6pts (3pts for finding the right record, 3 pts for null value)
1, 4, …
1, 5, …
null, 2, …
4, 4, …
5, 1, …

Question 5: Short Answers (10 points)

(a) (1 pt each) Given two relations **R** and **S**, where **R** has N_R tuples, **S** has N_S tuples, and $N_R > N_S > 0$, what are the minimum and maximum possible result cardinalities for the relational algebra queries **R** ∪ **S**, **R** ∩ **S**, and **R** □ **S** expressed in terms of N_R and N_S?

R ∪ S: min size: ____N_R max size: ____$N_R + N_S$

R ∩ S: min size: ____0 max size: ____N_s

R □ S: min size: ____Nr * Ns max size: ____$N_R * N_S$

(b) (4 points) Suppose that the table Emps(eid, name, salary, bonus) contains only two tuples:

(10, 'Joe', 1000.0, NULL)
(20, 'Sally', NULL, 2000.0)

For each of the following SQL queries, show the output that it would produce if run against Emps:

(i) (2 pts) SELECT salary + bonus AS answer FROM Emps WHERE name = 'Joe';

NULL

(ii) (2 pts) SELECT SUM(salary) AS answer FROM Emps;

1000

SCORE: _____