NAME:

STUDENT ID:

Midterm Exam #2
CSC 380
Fall 2018

Max. Points: 40

QUESTION	***TOPIC***	***POINTS***	***SCORE***
1	**True or False?**	**5**	
2	**Query Matching**	**5**	
3	**Query Writing**	**20**	
4	**SQL Execution**	**5**	
5	**Short Answers**	**5**	
TOTAL	**All**	**40**	

Question 1: True or False? (5 points)

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

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

  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

- The statement Select * from SP where PNO = 'P1' and PNO = 'P2'; is correct syntactically, but returns no record.

  TRUE FALSE

Question 2: Query Matching (5 points)

Consider the schema for which is sketched below:

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

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!

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 NATURAL JOIN Reserves R;	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……
SELECT S.age FROM Sailors S WHERE MIN(S.age);	……Q4……

Question 3: Query Writing (20 points)

Consider again the 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:

(5 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 ((π bid ((σ age = 35 ∧rating >= 5 (Sailor)) □ Reserves) ∩ π bid ((σ rating < 5 (Sailor)) □ Reserves)) □ Boats)

(5 pts) ***Relational Algebra:***
Find the names of sailors who have not reserved a red boat.

π sname([(π sid (s) - π sid (σ color = 'red'(b) ⍁ Reserves)]⍁ Sailor)

(5 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
```

(5 pts) ***SQL:***
Find the age of the youngest sailor with age >= 18 for each rating level with at least 2 such sailors

```
SELECT s.rating, MIN(S.age)
 FROM Sailors s
WHERE s.age >= 18
GROUP BY S.rating
HAVING COUNT(*) >1
```

Question 4: SQL Execution (5 points)
Consider yet again the 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

Show below the result of running the following queries against the tables above:

- SELECT S.sname, S.age, B.bname, B.color
  FROM Sailors S, Reserves R, Boats B
  WHERE S.sid = R.sid AND S.age = 35.0 AND S.rating >= 3 AND R.bid = B.bid;

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

- SELECTS.rating, MIN(S.age)
  FROM Sailors S WHERE S.age>= 18
  GROUP BY S.rating
  Rating Min_Age
  5 35
  9 22
  10 19

Question 5: Short Answers (5 points)

(a) 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 x 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 x S: min size: $N_R * N_S$ max size: … $N_R * N_S$ ……..

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

……NULL……

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

…… NULL ……