Exercises on SQL

Try to solve as many problems as possible. You are not expected to complete this in one tutorial. Go over these exercises before you attend your tutorial. For each exercise, try it first and discuss with your TA any problems you may have.
Note on Solution: For most questions, more than one solution is possible.

Question

Consider the Sailors-Boats-Reserves DB described in the text.
s (sid, sname, rating, age)
b (bid, bname, color)
r (sid, bid, date)
Write each of the following queries in SQL.

1. Find the colors of boats reserved by Albert.

```
SELECT color
FROM    s, b, r
WHERE r.sid=s.sid AND r.bid=b.bid AND
            sname=`Albert'
```

2. Find all sailor id's of sailors who have a rating of at least 8 or reserved boat 103.

```
(SELECT sid
FROM     s
WHERE  rating>=8)
UNION
(SELECT  sid
FROM     r
WHERE  bid=103)
```

3. Find the names of sailors who have not reserved a boat whose name contains the string "storm". Order the names in ascending order.

```
SELECT  sname
FROM     s s1
WHERE  sid NOT IN
   (SELECT  sid
    FROM      r, s
    WHERE r.sid=s.sid AND sname LIKE `%storm%')
ORDER BY s1.sname
```

4. Find the sailor id's of sailors with age over 20 who have not reserved a boat whose name includes the string "thunder".

```
SELECT  sid
FROM    s
WHERE  age>20 AND sid NOT IN
   (SELECT sid
    FROM r, b
    WHERE r.bid=b.bid AND bname LIKE `%thunder%')
```

5. Find the names of sailors who have reserved at least two boats.

```
SELECT sname
FROM    s, r r1, r r2
WHERE s.sid=r1.sid AND s.sid=r2.sid AND
            r1.bid<>r2.bid
```

Note: If we want to eliminate duplicates, we SELECT DISTINCT sname. Alternatively, we could simply change the condition in the WHERE clause from r1.bid<>r2.bid to, say r1.bid<r2.bid. (Why would this have the same effect?)

6. Find the names of sailors who have reserved all boats.

```
SELECT sname
FROM    s
WHERE NOT EXISTS (
   SELECT *
   FROM b
  WHERE NOT EXISTS (
     SELECT *
     FROM    r
    WHERE r.sid=s.sid AND r.bid=b.bid))
```

Note: A general remark that applies to most queries involving EXISTS or NOT EXISTS is that you can often "not care" about which attributes are selected in the SELECT statement. The reason is using the above constructs you merely tests the non-emptiness or emptiness of a certain collection of tuples. In other words, we can just do a SELECT *. On the other hand, if you are taking the difference between two collections (e.g., using EXCEPT) then you **must** care about the attributes being selected, since difference is only defined between two union-compatible relations.

7. Find the names of sailors who have reserved all boats whose name starts with “typhoon”.

```
SELECT sname
FROM    s
WHERE NOT EXISTS (
   SELECT *
   FROM b
  WHERE bname LIKE `typhoon%’ AND NOT EXISTS (
     SELECT *
     FROM     r
    WHERE r.sid=s.sid AND r.bid=b.bid))
```

8. Find the sailor id’s of sailors whose rating is better than some sailor called Bob.

```
SELECT  s1.sid
FROM     s s1, s s2
WHERE s1.rating>s2.rating AND s2.sname=`Bob’
```

Alternatively:

```
SELECT sid
FROM    s
WHERE rating > any (
  SELECT rating
  FROM    s s2
  WHERE s2.sname=`Bob’))
```

9. Find the sailor id’s of sailors whose rating is better than every sailor called Bob.

```
SELECT sid
FROM    s
WHERE rating > all (
  SELECT rating
  FROM    s s2
  WHERE s2.sname=`Bob’)
```

10. Find the sailor id’s of sailors with the highest rating.

```
SELECT sid
FROM    s s1
WHERE s1.rating >= all (
  SELECT rating
  FROM    s)
```

11. Find the name and age of the oldest sailor.

```
SELECT s1.sname, s1.age
FROM    s s1
WHERE NOT EXISTS (
  SELECT *
  FROM     s s2
  WHERE s2.age>s1.age)
```

Alternatively:

```
SELECT s1.sname, s1.age
FROM    s s1
WHERE s1.age >= all (
  SELECT age
  FROM    s)
```

12. Find the names of sailors who have reserved every boat reserved by those with a lower rating.

```
SELECT s1.sname
FROM    s s1
WHERE NOT EXISTS (
  SELECT *
  FROM     b, r, s s2
  WHERE  s2.sid=r.sid AND s2.bid=b.bid AND s2.rating<s1.rating AND
              NOT EXISTS (
    SELECT *
    FROM    r r2
    WHERE s1.sid=r2.sid AND r2.bid=b.bid))
```

Alternatively:

```
SELECT s1.sname
FROM    s s1
WHERE NOT EXISTS (
  (SELECT b1.bid
   FROM     b b1, r, s s2
  WHERE   s2.rating<s1.rating AND s2.sid=r.sid)
    EXCEPT
  (SELECT b2.bid
   FROM     b b2, r r2
   WHERE r2.bid=b.bid AND r2.sid=s1.sid))
```

Note: Recall my note above regarding the use of EXISTS/NOT EXISTS in conjunction with EXCEPT. Make sure you understand why in the previous queries we could just SELECT * but in this query, we pay attention to the attributes being selected.

13. For each rating, find the average age of sailors at that level of rating.

```
SELECT  rating, AVG(age)
FROM    s
GROUP BY rating
```

14. For each boat which was reserved by at least 5 distinct sailors, find the boat id and the average age of sailors who reserved it.

```
SELECT  b.bid, AVG(age)
FROM    b, r, s
WHERE  b.bid=r.bid AND r.sid=s.sid
GROUP BY bid
HAVING 5<=COUNT(DISTINCT r.sid)
```

This is equivalent to the following more complex query, which many optimizers may not optimize well and hence the DBMS may take longer to evaluate the following query.

```
SELECT b1.bid, AVG(s.age)
FROM    r, s (SELECT bid
                FROM    b, r r2
                WHERE b.bid=r2.bid
                GROUP BY bid
                HAVING 5<=COUNT(DISTINCT r2.sid)) b1
WHERE b1.bid=r.bid AND r.sid=s.sid
GROUP BY b1.bid
```

15. For each boat which was reserved by at least 5 sailors with age >= 40, find the boat id and the average age of such sailors.

```
SELECT bid, AVG(age)
FROM    b, r, s
WHERE s.age>=40 AND b.bid=r.bid AND s.sid=r.sid
GROUP BY bid
HAVING 5<=COUNT(DISTINCT s.sid)
```

16. For each boat which was reserved by at least 5 sailors with age >= 40, find the boat id and the average age of all sailors who reserved the boat.

```
SELECT b1.bid, AVG(s.age)
FROM    r, s (SELECT bid
              FROM    b, r r2, s s2
              WHERE b.bid=r2.bid AND s2.sid=r.sid AND s.age>=40
              GROUP BY bid
              HAVING 5<=COUNT(DISTINCT r2.sid)) b1
WHERE b1.bid=r.bid AND r.sid=s.sid
GROUP BY b1.bid
```

Unlike the previous query, this query cannot be “flattened”, i.e., cannot be expressed without a nested subquery. It is of course possible to express this query with the WITH construct for defining a temporary table. Similarly, it can also be done by defining a view. Practice writing those queries on your own.