

CSE 132A
Winter 2019
Assignment #1 (SQL)
Due February 4, 11:59pm

This is an individual assignment. Please review the academic integrity policy and contact the instructor or TA if you have any questions.

The boat reservations database has the following schema:

sailor: *sname* (string), *rating* (integer)
boat: *bname* (string), *color* (string), *rating* (integer)
reservation: *sname* (string), *bname* (string), *day* (string)

The rating attribute for boats indicates the minimum rating required of a sailor reserving the boat. Assume all values are non-null and the database contains no duplicate tuples. In addition, the following hold:

- *sname* is the primary key of *sailor* (so every sailor has just one rating);
- *bname* is the primary key of *boat* (so every boat has just one color and one rating);
- *bname* is a foreign key in the *reservation* relation, referencing relation *boat* (so every *bname* in *reservation* occurs in *boat*, but the converse need not be true)
- *sname* is a foreign key in the *reservation* relation, referencing relation *sailor* (so every *sname* in *reservation* occurs in *sailor*, but the converse need not be true)

This schema is set up by the posted script `schema.sql`. The script additionally sets up a table *weekday* with a single string attribute *day*, containing the days of the week. You may use this table in your queries.

Here is one example instance over the above schema:

<i>sailor</i>	<i>sname</i>	<i>rating</i>	<i>boat</i>	<i>bname</i>	<i>color</i>	<i>rating</i>
	Brutus	1		SpeedQueen	white	9
	Andy	8		Interlake	red	8
	Horatio	7		Marine	blue	7
	Rusty	8		Bay	red	3
	Bob	1				

<i>reservation</i>	<i>sname</i>	<i>bname</i>	<i>day</i>
	Andy	Interlake	Monday
	Andy	Bay	Wednesday
	Andy	Marine	Saturday
	Rusty	Bay	Sunday
	Rusty	Interlake	Wednesday
	Rusty	Marine	Wednesday
	Bob	Bay	Monday

(A) Write the following queries in SQL (of course, the queries must work on all data, not just the sample one). You are free to use multiple commands and define views as convenient.

1. List all boats reserved on Wednesday and their color.
2. List all pairs of sailors who have reserved boats on the same day (avoid listings of the form $\langle a, a \rangle$, or listing both $\langle a, b \rangle$ and $\langle b, a \rangle$).
3. For each day, list the number of red boats reserved on that day. If there is no red boat reserved on a given day, the number shown should be zero. In particular, if a day of the week does not appear at all in the *reservation* relation, it should appear in the answer with zero red boat reservations.

Hint: you may use the *weekday* table containing all days of the week.

4. List the days appearing in the *reservation* table for which only red boats are reserved. Provide three SQL queries, using nested sub-queries in different ways:
 - with only NOT IN tests;
 - with only NOT EXISTS tests;
 - with only COUNT aggregate functions.

5. For each day of the week occurring in the *reservation* relation, list the average rating of sailors having reserved boats that day. **Caution:** watch for dangerous duplicate tuples!. You may write this as a sequence of two or more queries (including CREATE VIEW if useful). Make sure the average is not truncated to an integer

(you may convert an integer to a real by multiplying the integer with 1.0).

6. List the busiest days, i.e. the days with the largest number of reservations. Note: if the *reservations* table is empty, all days of the week should be in the answer.
7. Formulate a query to verify that all sailors having reservations are qualified to sail the boats they reserved. Specifically, the query should look for violations. It should return the sailor/boat pairs that violate the constraint together with the rating of the sailor (named *srating*), the rating of the boat (named *brating*), and the day of the reservation. The answer should be empty if there is no violation.

(B) Write the following updates in SQL.

1. Switch all Wednesday and Monday reservations, without explicitly naming the boats involved. **Hint:** You may use several commands to do this.
2. Delete all sailors who are not qualified to sail any boat, together with their reservations (this has to be done carefully, to avoid violations of the referential integrity constraint from *reservation* to *sailor*).

Instructions on turning in the assignment.

1. Create separate files containing your answer to each of the questions A1 - A7 and B1 - B2. The files should be named A1.sql, A2.sql, etc. Do not include any comments in the files.
2. Create a zip file containing all of the above and upload on Gradescope