

B561 Assignment 1
Relational Databases and Basic SQL queries
Due: Thursday January 17, 2019 by 11:59 pm

The goal of this assignment is to become familiar with the PostgreSQL system, to create a relational database, and to write and evaluate some simple SQL statements and queries in that system.

You need to upload a single file with name `assignment1.sql` file to Canvas which contains the necessary SQL statements that solve the problems in this assignment. The `assignment1.sql` file must be so that the AI's can run it in their PostgreSQL shell. (We will post a sample .sql file that gives the template for your `assignment1.sql` file.)

Before you can solve this assignment, you will need download PostgreSQL (version 10) and install it on your computer.

Consider the following relation schemas for a database that maintains sailors, boats, and reservations of boats by sailors.

Sailor(Sid INTEGER, Sname TEXT, Rating INTEGER, Age INTEGER)
Boat(Bid INTEGER, Bname TEXT, Color TEXT)
Reserves(Sid INTEGER, Bid INTEGER, Day TEXT)

You should assume that Sid in Reserves is a foreign key that references the primary key Sid in Sailor, and that Bid in Reserves is a foreign key that references the primary key Bid in Boat.

Note the files sailor.sql, boat.sql, and reserves.sql that contain the relation instances for the Sailor, Boat, and Reserves relations, respectively.

1. Create a database with name `assignment1` in PostgreSQL that stores these relations. Make sure to specify primary and foreign keys. Then write SQL queries that return each of the relation instances Sailor, Boat, and Reserves.
2. Provide 6 examples that illustrate how the presence or absence of primary and foreign keys affects insert and deletes in these relations. To solve this problem, you will need to experiment with the relation schemas. For example, you should consider altering primary keys and foreign key constraints and then consider various sequences of insert and delete operations. Certain inserts and deletes should succeed but other should create error conditions. (Consider the lecture notes about keys, foreign keys, and inserts and deletes as a guide to solve this problem.)

3. Write SQL statements for the following queries.

Make sure that each of your queries returns a set but not a bag. In other words, make appropriate use of the `DISTINCT` clause where necessary.

- (a) Find the name of each boat.
- (b) Find the sid and rating of each sailor.
- (c) Find the sid, name, and rating of each sailor whose rating is not in the range $[7, 9]$.
- (d) Find the bid and name of each red boat that was reserved by some sailor whose rating is less than 9.
- (e) Find the bid and bname of each boat that was reserved by a sailor on a Monday and by possibly a different sailor on a Tuesday.
- (f) Find the sid of each sailor who did not reserve any boats on a Monday, Wednesday, or a Thursday.
- (g) Find the bid and name of each boat that was reserved by more than one sailor (i.e., reserved by at least two sailors).
- (h) Find the pairs of bids (b_1, b_2) of different boats that were both reserved by a same sailor.
- (i) Find the pairs (s, b) such that the sailor with sid s reserved the boat with bid b , provided that the sailor s has a rating above 7 and the color of boat b is not red.
- (j) Find the sids of sailors who reserved exactly one red boat. (You should write this query without using the `COUNT` aggregate function.)
- (k) Find the bids of boats that were reserved by fewer than two sailors. (You should write this query without using the `COUNT` aggregate function.)