Zhihao Wang

ECE-464 Assignment 1

Prof. Sokolov

Part 1

Download the dataset and schema of sailors and boats from our in class discussion. Write SQL queries to answer the following questions. Include your query (and its output from your terminal) in your submissions.

1. Select, for each boat, the sailor who made the highest number of reservations for that boat.

SELECT DISTINCT b.bname, s.sname, count(*) FROM boats AS b JOIN reserves AS r ON b.bid = r.bid Join sailors AS s ON s.sid = r.sid GROUP BY b.bid, b.bname, s.sid, s.sname HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM reserves ra WHERE ra.bid = b.bid GROUP BY ra.sid);

```
mysql> SELECT DISTINCT b.bname, s.sname, count(*) FROM boats AS b JOIN reserves AS r ON b.bid = r.bid Join s
ailors AS s ON s.sid = r.sid GROUP BY b.bid, b.bname, s.sid, s.sname HAVING COUNT(*) >= ALL (SELECT COUNT(*)
 FROM reserves ra WHERE ra.bid = b.bid GROUP BY ra.sid);
+-----------+----------+----------+
| bname     | sname    | count(*) |
+-----------+----------+----------+
| Interlake | dusting  |        1 |
| Clipper   | dusting  |        1 |
| Clipper   | emilio   |        1 |
| Marine    | emilio   |        1 |
| Clipper   | scruntus |        1 |
| Interlake | lubber   |        1 |
| Clipper   | lubber   |        1 |
| Clipper   | figaro   |        1 |
| Marine    | figaro   |        1 |
| Marine    | stum     |        1 |
| Driftwood | stum     |        1 |
| Marine    | jit      |        2 |
| Driftwood | jit      |        1 |
| Sooney    | ossola   |        1 |
| Interlake | horatio  |        1 |
| Clipper   | horatio  |        1 |
| Marine    | dan      |        1 |
| Klapser   | dan      |        2 |
| Sooney    | dan      |        1 |
| Driftwood | dye      |        1 |
| Driftwood | vin      |        1 |
+-----------+----------+----------+
21 rows in set (0.02 sec)
```

2. List, for every boat, the number of times it has been reserved, excluding those boats that have never been reserved (list the id and the name).

SELECT b.bid, b.bname, COUNT(*) FROM reserves AS r, boats AS b WHERE r.bid = b.bid GROUP BY r.bid;

```
mysql> SELECT b.bid, b.bname, COUNT(*) FROM reserves AS r, boats AS b WHERE r.bid = b.bid GROUP
 BY r.bid;
+-----+-----------+----------+
| bid | bname     | COUNT(*) |
+-----+-----------+----------+
| 101 | Interlake |        2 |
| 102 | Interlake |        3 |
| 103 | Clipper   |        3 |
| 104 | Clipper   |        5 |
| 105 | Marine    |        3 |
| 106 | Marine    |        3 |
| 109 | Driftwood |        4 |
| 112 | Sooney    |        1 |
| 110 | Klapser   |        3 |
| 107 | Marine    |        1 |
| 111 | Sooney    |        1 |
| 108 | Driftwood |        1 |
+-----+-----------+----------+
12 rows in set (0.00 sec)
```

3. List those sailors who have reserved every red boat (list the id and the name).

SELECT s.sid, s.sname FROM sailors AS s WHERE NOT EXISTS (SELECT * FROM boats AS b WHERE b.color = 'red' AND NOT EXISTS (SELECT * FROM reserves AS r WHERE r.sid = s.sid AND r.bid = b.bid));

```
mysql> SELECT s.sid,  s.sname FROM sailors AS s WHERE NOT EXISTS (SELECT * FROM boats AS b WHERE b.color = 'red'
 AND NOT EXISTS (SELECT * FROM reserves AS r WHERE r.sid = s.sid AND r.bid = b.bid));
 Empty set (0.00 sec)
```

4. List those sailors who have reserved only red boats.

SELECT DISTINCT s.sid, s.sname FROM sailors AS s, reserves AS r WHERE s.sid = r.sid AND s.sid NOT IN (SELECT ra.sid FROM reserves AS ra, boats AS b WHERE ra.bid = b.bid AND b.color <> 'red');

```
mysql> SELECT DISTINCT s.sid, s.sname FROM sailors AS s, reserves AS r WHERE s.sid = r.sid AND s.sid NOT IN (SELECT
 ra.sid  FROM reserves AS ra, boats as b where ra.bid = b.bid and b.color <> 'red');
 +-----+----------+
 | sid | sname    |
 +-----+----------+
 |  23 | emilio   |
 |  24 | scruntus |
 |  35 | figaro   |
 |  61 | ossola   |
 |  62 | shaun    |
 +-----+----------+
 5 rows in set (0.00 sec)
```

5. For which boat are there the most reservations?

SELECT r.bid, b.bname, COUNT(*) FROM reserves AS r, boats AS b WHERE r.bid = b.bid GROUP BY r.bid ORDER BY COUNT(*) DESC LIMIT 1;

```
mysql> select r.bid, b.bname, count(*) from reserves as r, boats as b where r.bid = b.bid group
 by r.bid order by count(*) desc limit 1;
+-----+---------+----------+
| bid | bname   | count(*) |
+-----+---------+----------+
| 104 | Clipper |        5 |
+-----+---------+----------+
1 row in set (0.01 sec)
```

6. Select all sailors who have never reserved a red boat.

SELECT s.sid, s.sname FROM sailors AS s WHERE s.sid NOT IN (SELECT r.sid FROM reserves AS r JOIN boats AS b ON r.bid = b.bid WHERE b.color = 'red');

```
mysql> SELECT s.sid, s.sname FROM sailors AS s WHERE s.sid NOT IN (SELECT r.sid FROM reserves AS r JOIN boats AS b
ON r.bid = b.bid WHERE b.color = 'red');
+-----+---------+
| sid | sname   |
+-----+---------+
|  29 | brutus  |
|  32 | andy    |
|  58 | rusty   |
|  60 | jit     |
|  71 | zorba   |
|  74 | horatio |
|  85 | art     |
|  90 | vin     |
|  95 | bob     |
+-----+---------+
9 rows in set (0.00 sec)
```

7. Find the average age of sailors with a rating of 10.

SELECT AVG(s.age) FROM sailors AS s WHERE s.rating=10;

```
mysql> select avg(s.age) from sailors as s where s.rating=10;
+-------------+
| avg(s.age) |
+-------------+
|    35.0000 |
+-------------+
1 row in set (0.00 sec)
```

Part 2

Represent the sailors and boats schema using an ORM. Show that it is fully functional by writing tests using the data from part 1(writing the queries for the questions in Part 1)

Created in different python files.

Part 3

Students are hired as software consults for a small business boat rental that is experiencing a heavy influx of tourism in its area. This increase is hindering operations of the mom/pop shop that uses paper/pen for most tasks. Students should explore “inefficient processes” the business may have and propose ideas for improvements - in the form of a brief write-up. Expand the codebase from part 2 to include a few jobs, reports, integrity checks, and/or other processes that would be beneficial to the business. Use the data provided in part 1 and expand it to conduct tests and show functionality.

Proposal for Improvement:

Expanding from the original data in part 1, in order to keep track of the payment and cost, we need to add a column ‘wage’ to the sailors for paying the salary of the sailors.

The current system lacks essential mechanism to record any financial transactions.

In addition, we need to add a new column for recording profit to the reserves table to log the earning for each reservation. We also need a relation table to record the actual working time and date of the sailors.

Then the tables became structures like this.

```
mysql> select * from reserves;
+-----+-----+------------+---------+
| sid | bid | day        | earning |
+-----+-----+------------+---------+
|  22 |  22 | 1998-10-10 |      10 |
|  23 |  23 | 1998-10-12 |      18 |
|  32 |  32 | 1998-10-15 |      22 |
|  34 |  33 | 1998-10-18 |      15 |
|  56 |  44 | 1998-10-20 |      17 |
+-----+-----+------------+---------+
5 rows in set (0.00 sec)

mysql> select * from work_record;
+------------+-----+-------+
| day        | sid | hours |
+------------+-----+-------+
| 1998-10-10 |  22 |     5 |
| 1998-10-12 |  23 |     5 |
| 1998-10-15 |  32 |     5 |
| 1998-10-18 |  34 |     5 |
| 1998-10-20 |  56 |     5 |
+------------+-----+-------+
5 rows in set (0.00 sec)

mysql> select * from work_record;
+------------+-----+-------+
| day        | sid | hours |
+------------+-----+-------+
| 1998-10-10 |  22 |     5 |
| 1998-10-12 |  23 |     5 |
| 1998-10-15 |  32 |     5 |
| 1998-10-18 |  34 |     5 |
| 1998-10-20 |  56 |     5 |
+------------+-----+-------+
5 rows in set (0.00 sec)
```

Now it is possible to realize these two functions:

- Bi weekly payment query

select sailors.sname, sailors.salary*work_record.hours AS wages From work_record join sailors on sailors.sid = work_record.sid where day >= '2019/10/10' AND day <= '2019/10/24';

```
mysql> select sailors.sname, sailors.salary*work_record.hours AS wages From work_record join sailors on sailors.sid
 = work_record.sid where day >= '2019/10/10' AND day <= '2019/10/24';
+-------+-------+
| sname | wages |
+-------+-------+
| abc   |    33 |
| def   |    84 |
| ghi   |   169 |
| jkl   |    70 |
| mnq   |   120 |
+-------+-------+
5 rows in set (0.00 sec)
```

- Monthly accounting manager

select((select sum(reserves.earning) from reserves where reserves.day >= '2019/10/10' and reserves.day <= '2019/11/10') - (select SUM(sailors.salary*work_record.hours) AS wages From work_record join sailors on sailors.sid = work_record.sid where day >= '2019/10/10' AND day <= '2019/11/10')) as october_profit;

```
mysql> select((select sum(reserves.earning) from reserves where reserves.day >= '2019/10/10' and reserves.day
 <= '2019/11/10') - (select SUM(sailors.salary*work_record.hours) AS wages From work_record join sailors on s
ailors.sid = work_record.sid where day >= '2019/10/10' AND day <= '2019/11/10')) as october_profit;
+----------------+
| october_profit |
+----------------+
|            344 |
+----------------+
1 row in set (0.00 sec)
```