

Testing

You can run your answers through SQLite directly by running `sqlite3 lahman.db` to open the database and then entering `.read proj1.sql`

```
$ sqlite3 lahman.db
SQLite version 3.33.0 2020-08-14 13:23:32
Enter ".help" for usage hints.
sqlite> .read proj1.sql
```

This can help you catch any syntax errors in your SQLite.

To help debug your logic, we've provided output from each of the views you need to define in questions 1-4 for the data set you've been given. Your views should match ours, but note that your SQL queries should work on ANY data set. **We will test your queries on a (set of) different database(s), so it is *NOT* sufficient to simply return these results in all cases!** Please also note that queries that join on extra, unnecessary tables will slow down queries and not receive full credit on the hidden tests.

To run the test, from within the `sp24-proj1-yourname` directory:

```
$ python3 test.py
$ python3 test.py -q 4ii # This would run tests for only q4ii
```

Become familiar with the UNIX [diff](#) format, if you're not already, because our tests saves a simplified diff for any query executions that don't match in `diffs/`. As an example, the following output for `diffs/q1i.txt`:

```
- 1|1|1
+ Jumbo|Diaz|1984
+ Walter|Young|1980
```

indicates that your output has an extra `1|1|1` (the `-` at the beginning means the expected output *doesn't* include this line but your output has it) and is missing the lines `Jumbo|Diaz|1984` and `Walter|Young|1980` (the plus at the beginning means the expected output *does* include

those lines but your output is missing it). If there is neither a `+` nor `-` at the beginning then it means that the line is in both your output and the expected output (your output is correct for that line).

If you care to look at the query outputs directly, ours are located in the `expected_output` directory. Your view output should be located in your solution's `your_output` directory once you run the tests.

Note: For queries where we don't specify the order, it doesn't matter how you sort your results; we will reorder before comparing. Note, however, that our test query output is sorted for these cases, so if you're trying to compare yours and ours manually line-by-line, make sure you use the proper `ORDER BY` clause (you can determine this by looking in `test.py`). Different versions of SQLite handle floating points slightly differently so we also round certain floating point values in our own queries. A full list is specified here for convenience:

```
SELECT * FROM q0;
SELECT * FROM q1i ORDER BY namefirst, namelast, birthyear;
SELECT * FROM q1ii ORDER BY namefirst, namelast, birthyear;
SELECT birthyear, ROUND(avgheight, 4), count FROM q1iii;
SELECT birthyear, ROUND(avgheight, 4), count FROM q1iv;
SELECT * FROM q2i;
SELECT * FROM q2ii;
SELECT * FROM q2iii;
SELECT playerid, namefirst, namelast, yearid, ROUND(slg, 4) FROM q3i;
SELECT playerid, namefirst, namelast, ROUND(lslg, 4) FROM q3ii;
SELECT namefirst, namelast, ROUND(lslg, 4) FROM q3iii ORDER BY namefirst, namela
SELECT yearid, min, max, ROUND(avg, 4) FROM q4i;
SELECT * FROM q4ii WHERE binid <> 9;
WITH max_salary AS (SELECT MAX(salary) AS salary FROM salaries)
    SELECT binid, low,
        ((CASE WHEN high >= salary THEN ' ' ELSE 'not ' END) ||
         'at least ' || salary) AS high, count
    FROM q4ii, max_salary WHERE binid = 9;
SELECT yearid, mindiff, maxdiff, ROUND(avgdiff, 4) FROM q4iii;
SELECT * FROM q4iv ORDER BY yearid, playerid;
SELECT team, ROUND(diffAvg, 4) FROM q4v ORDER BY team;
```

[Previous](#)
[Your Tasks](#)

[Next](#)
[Submitting the Assignment](#)

Last updated 6 months ago