QUERY PERFORMANCE COMPARISON BETWEEN POSTGRESQL AND SQLIGHT

Rayan Jaipuriyar
March 5, 2024

Abstract:

This study measures the performance of PostgreSQL and SQLight database systems. Queries that utilize JOIN, GROUP BY, WHERE, and HAVING clauses were run on a light relational database structure loaded into both systems. Execution times were collected and averaged over 10 groupings per query. The results indicate that PostgreSQL handles JOIN and WHERE queries better than SQLight with an average execution time of 0.1403 ±0.0243 to 0.3446 ±0.0546 at 95% confidence (p-value <0.001) and SQLight handles GROUP BY and HAVING queries better than PostgreSQL with an average execution time of 0.1837 ±0.0267 to 0.1301 ±0.0415 at 95% confidence (p-value <0.001). Additionally, performance degradation when running successive queries over short time intervals was observed in SQLight.

Introduction:

The aim of this term project is to conduct a comprehensive performance benchmark study comparing PostgreSQL and SQLite, two popular relational database management systems. The focus will be on evaluating their performance on a series of SQL queries. Both databases are known for their reliability, robustness, and versatility, but they serve different use cases and exhibit distinct architectures. Postgres is designed to handle large volumes of data and complex transactions while providing advanced features such as indexing, foreign keys, and stored procedures. It is frequently utilized in enterprise-level applications, data warehousing, and situations demanding scalability. SQLite, on the other hand, is renowned for its simplicity, portability, and ease of integration. Unlike PostgreSQL, SQLite operates in a file-based manner and is an ideal solution for scenarios where a standalone, zero-configuration database is sufficient. Businesses will often have to pick between the two systems and having a clear understanding of benchmark performance metrics across both platforms will allow for more holistic and informed decision making.

Literature Review:

Websites comparing the two platforms can be found all over the internet. However, these studies do not take place in an academic setting and contain a fair amount of hearsay. Additionally the studies often avoid publishing the raw, numeric results from these tests. Some examples of these studies can be found in the bibliography of this paper. In light of this, it seems prudent to conduct a numerically oriented study that compares execution time for the database systems.

Method:

The Database:

I will be using a sample employee record database containing 40 employee entries. I procured this database from sqltutorial.org (SQL Tutorials 2009). This website very helpfully provided me with build and load SQL files for both PostgreSQL and SQLight, allowing me to bypass having to construct the files myself. The Entity Relationship Diagram for the database can be found in the Appendix.

The SQL Queries:

This study focuses on queries that utilize JOIN, GROUP BY, WHERE, and HAVING clauses. The following queries were tested for this study:

- select e.employee_id, e.first_name, e.last_name, j.job_title from employees e JOIN jobs j on e.job_id = j.job_id;

- select count(e.employee_id) as employees, j.job_title from employees e join jobs j on e.job_id = j.job_id GROUP BY j.job_title;
- select * from employees e WHERE e.salary >= 10000;
- select count(e.employee_id) as total_employees, j.job_title from employees e join jobs j on e.job_id = j.job_id group by j.job_title HAVING count(e.employee_id) > 1;

Each of the queries will be run 10 times and their average execution times will be collected. All other processes will be terminated for the duration of the test. The execution time is displayed using the EXPLAIN ANALYZE precursor to an SQL query in Postgres, and the .timer on function can be used to display execution time on SQLight.

Results:

The findings are summarized in the following table:

Query Type	**PostgreSQL Execution Time (ms)**	**SQLight Execution Time (ms)**
JOIN	**0.1403 ±0.0243 (±17.35%)**	**0.3446 ±0.0546 (±15.85%)**
GROUP BY	**0.1837 ±0.0267 (±14.52%)**	**0.1301 ±0.0415 (±31.93%)**
WHERE	**0.0513 ±0.00587 (±11.45%)**	**0.1283 ±0.0171 (±13.31%)**
HAVING	**0.1826 ±0.0285 (±15.62%)**	**0.0633 ±0.00775 (±12.25%)**

The raw data for every run can be found in the Appendix.

Conclusions:

From my testing it is apparent that PostgreSQL handles the JOIN and WHERE queries better than SQLight (p-value <0.001). On the flip side, GROUP BY and HAVING are better handled by SQLight (p-value <0.001). Taking a closer look, it is noticeable that these 2 queries both execute a COUNT and are likely to require more processing. Interestingly, it seems as if performance with SQLight degrades over successive query runs. Execution times trend upwards for all executed queries. No such trend is evident in the PostgreSQL run times, with performance remaining more consistent. Repeating the study with a larger database will be a prudent next step.

Appendix A

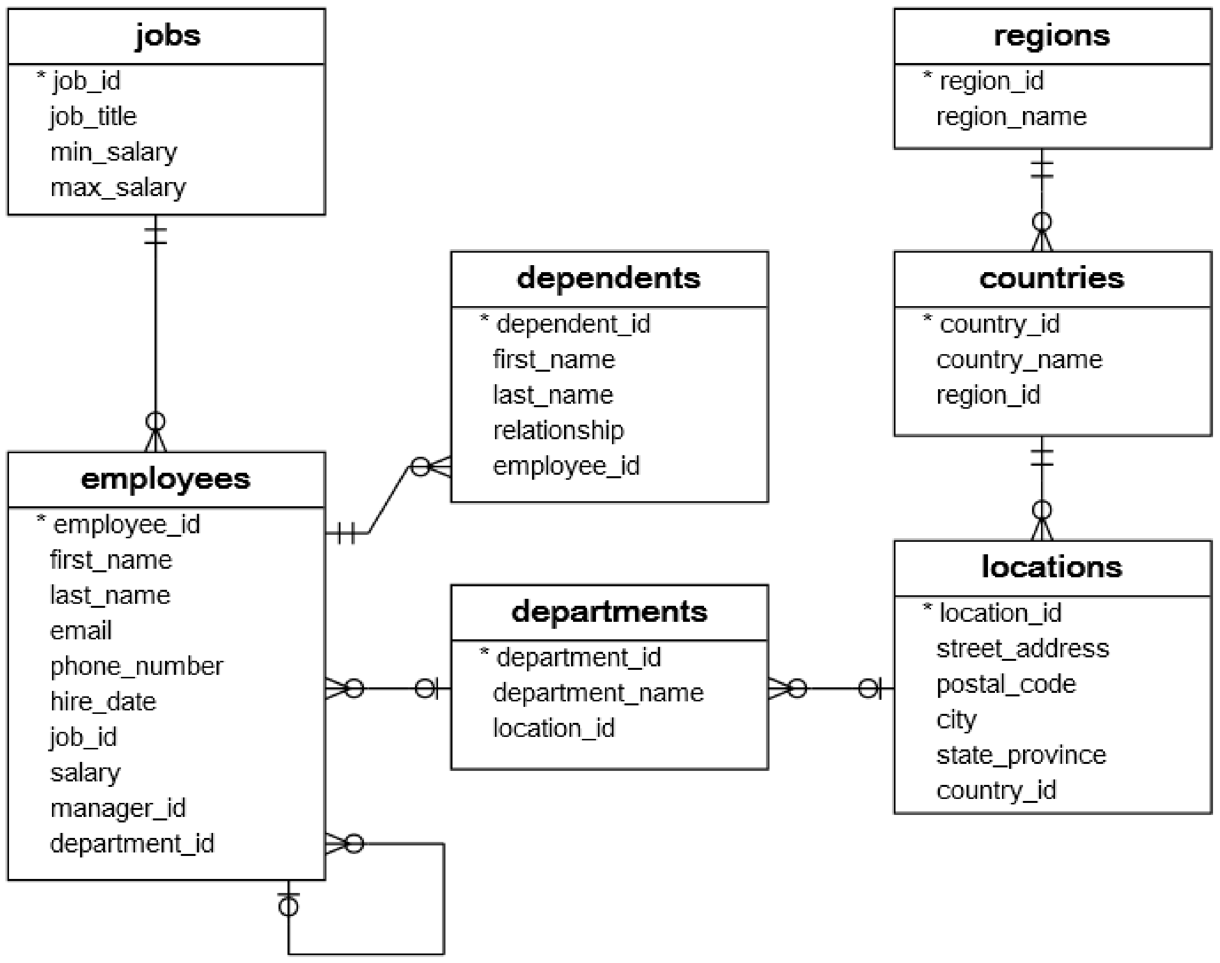
Raw Data:

Query Type	PostgreSQL Execution Time (ms)	SQLight Execution Time (ms)
JOIN	0.148	0.121
	0.111	0.243
	0.119	0.376
	0.118	0.363
	0.241	0.354
	0.154	0.381
	0.172	0.402
	0.111	0.419
	0.110	0.371
	0.119	0.416
Average at 95% confidence	**0.1403 ±0.0243 (±17.35%)**	**0.3446 ±0.0546 (±15.85%)**
GROUP BY	0.274	0.095
	0.237	0.009
	0.186	0.036
	0.149	0.074
	0.158	0.173
	0.148	0.189
	0.160	0.174
	0.224	0.175
	0.151	0.211
	0.150	0.165
Average at 95% confidence	**0.1837 ±0.0267 (±14.52%)**	**0.1301 ±0.0415 (±31.93%)**

WHERE	0.057	0.109
	0.078	0.067
	0.046	0.143
	0.050	0.134
	0.046	0.156
	0.046	0.142
	0.045	0.158
	0.048	0.137
	0.048	0.142
	0.049	0.095
Average at 95% confidence	**0.0513 ±0.00587 (±11.45%)**	**0.1283 ±0.0171 (±13.31%)**
HAVING	0.261	0.064
	0.154	0.053
	0.158	0.057
	0.212	0.066
	0.151	0.073
	0.159	0.085
	0.150	0.047
	0.160	0.080
	0.147	0.062
	0.274	0.046
Average at 95% confidence	**0.1826 ±0.0285 (±15.62%)**	**0.0633 ±0.00775 (±12.25%)**

Appendix B

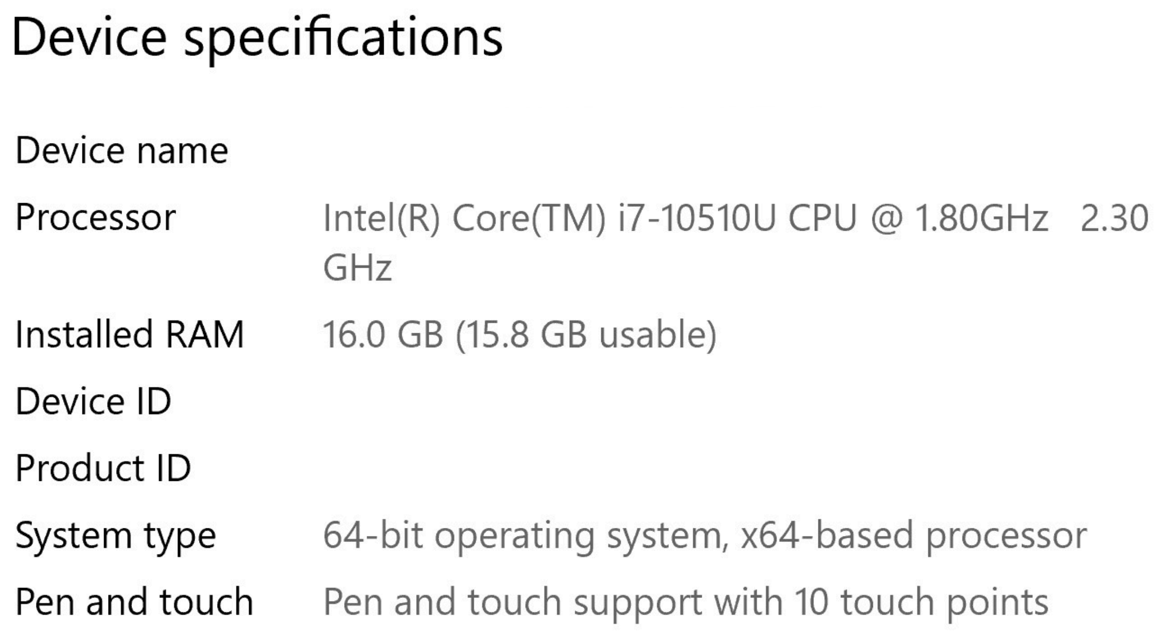
Database Entity Relationship Diagram:



Source: https://www.sqltutorial.org/sql-sample-database/

Appendix C

Local device information:



Device specifications

Device name	
Processor	Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz
Installed RAM	16.0 GB (15.8 GB usable)
Device ID	
Product ID	
System type	64-bit operating system, x64-based processor
Pen and touch	Pen and touch support with 10 touch points

References

Astera Analytics Team. 2024. “PostgreSQL vs SQLite: The Ultimate Database Showdown.” Astera Software. https://www.astera.com/knowledge-center/postgresql-vs-sqlite/.

Boltic. n.d. “PostgreSQL vs SQLite A Guide to Choosing Right 1.” Boltic. Accessed March 6, 2024. https://www.boltic.io/blog/postgresql-vs-sqlite.

SQL Tutorials. 2009. “SQL Sample Database.” SQL Tutorial. https://www.sqltutorial.org/sql-sample-database/.