Small Web-Based Denormalized Database Effectiveness Threshold

Min Young Lee
Computer Science Department
Advisors Dr. Zhang & Dr. Iyengar

WINONA STATE UNIVERSITY
Computer Science Department
http://cs.winona.edu

Introduction

The effectiveness of database organization influences database's performance. As the size of a database increases, data is organized in smaller sets in order to be cost-efficient. The process of organizing the dataset to achieve least amount of redundancy and dependency within the database is called **normalization**. The **denormalization** is a process to revert parts of the normalization in order to increase a read effectiveness. Denormalization is a strategy used to increase readability at the cost of a lower writability and more redundancy within the database . The sets of databases will be created from a completely normalized database then systematically adding redundancy until the database only has one table of data. Each sets of the database will be tested on readability and writability.

Question

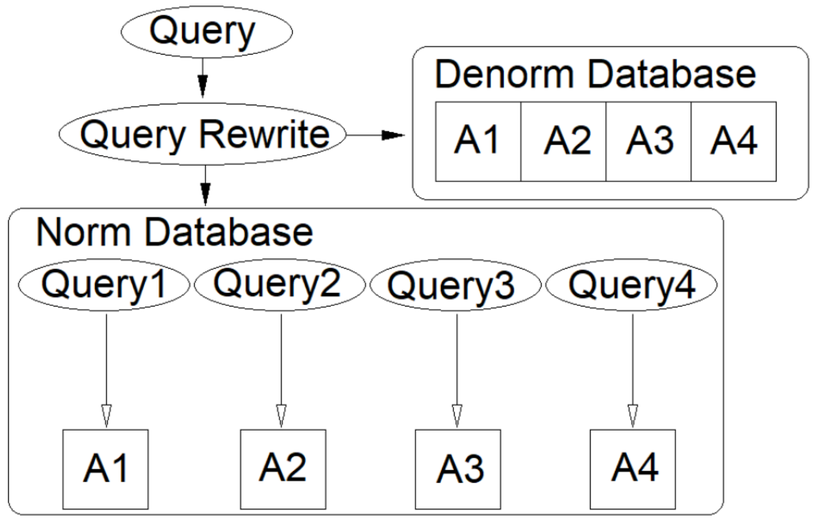
For small databases, how necessary is a database normalization in context of both readability and writability performance?

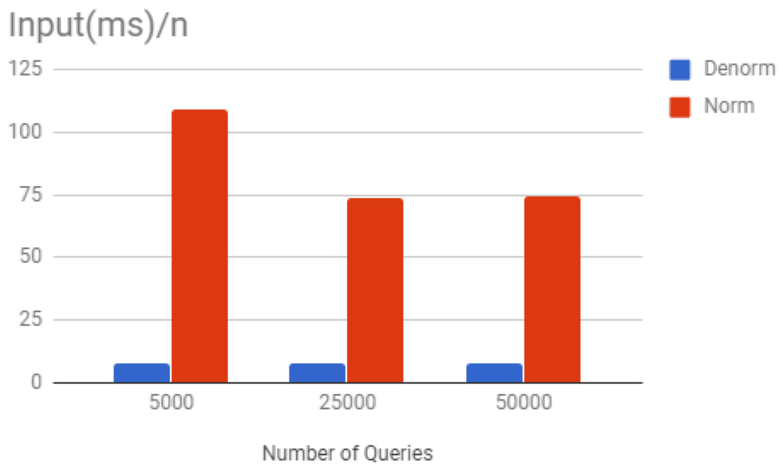
Method

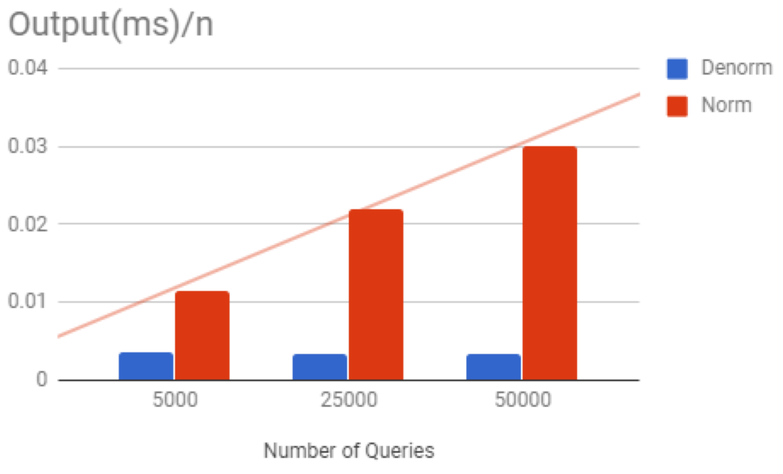
Sets of database are made with ten attributes. From one end of a spectrum (normalized), there is a completely normalized database with each attribute having its own table. From another end of the spectrum (denormalized), there is a completely denormalized database with every attribute on a single table. Each attribute consists of 6 digit random integer.

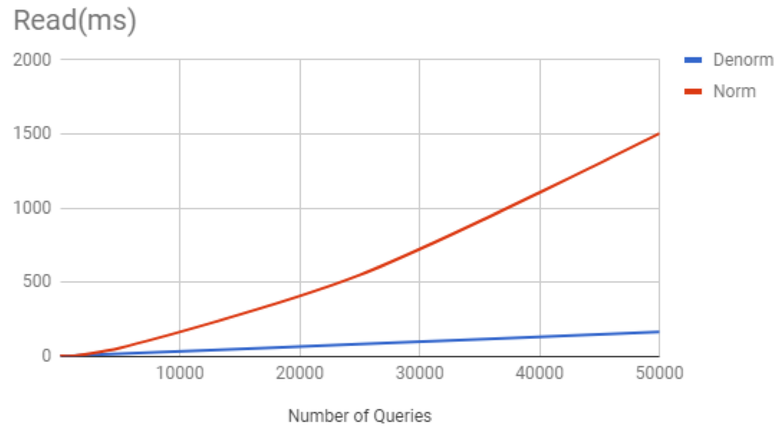
Sets of 1, 10, 100, 1000, 5000, 25,000, 50,000 INSERT/SELECT queries are executed for each databases. The query processing times are measured.

The time it took to execute the query to interact with databases and the browser were measured in milliseconds(ms) using Unix time.









Results

The raw data is gathered then calculated for the average for each set of query sizes. Each set is divided by the number of queries in the set to estimate the average the average processing time.

Overall, the normalized database took more time (ms) for both input and output. However, the trajectory of normalized database suggests a decrease in time-cost for inputs as the number of queries increases.

For input, the normalized databases seem to increase in efficiency as the number of queries increases. The denormalized databases seem to be almost unaffected by the change in number of input requests.

For output, the normalized databases seem to be greatly losing efficiency as the number of queries increases. The denormalized databases seem to be almost unaffected by the change in number of read requests.

Conclusion

It appears that for a small database, the degree of normalization is not impactful enough to be a concern in the context of readability and writability.

In the context of a small database, a degree of denormalization improves the time-cost performance of read queries.