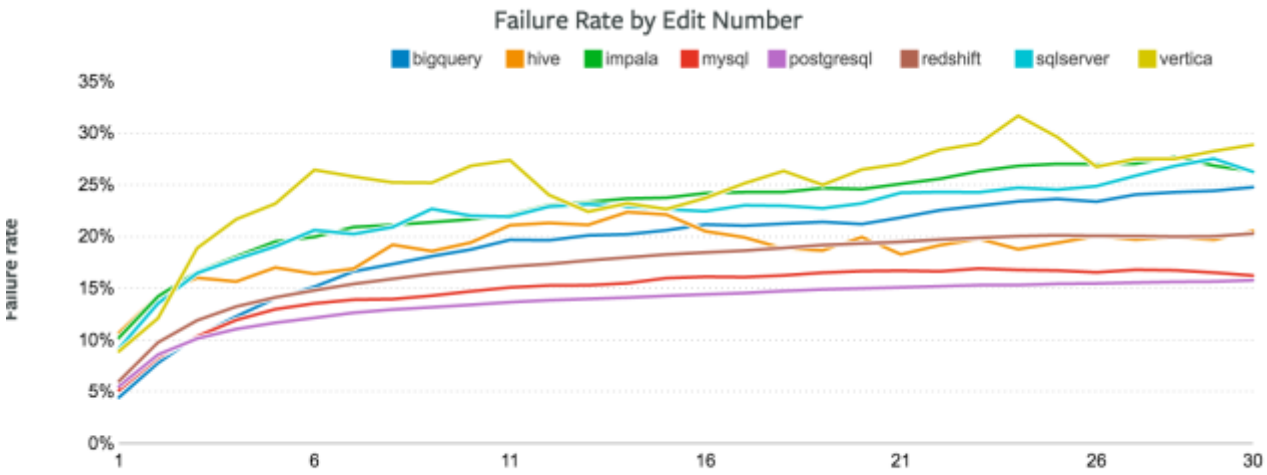
Mode Blog

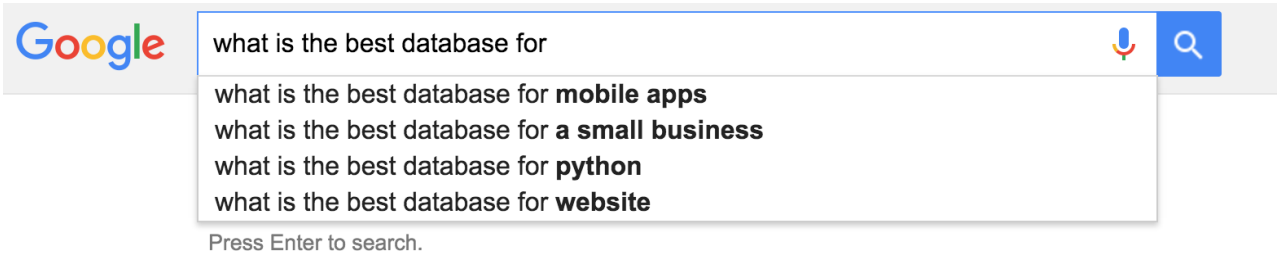


General
December 9, 2015 • 6 minute read

What's the best database for an analyst?

BENN STANCIL
CHIEF ANALYST

Which database is best? The question, obviously, depends on what you want to use it for.



I, like most analysts, want to use a database to warehouse, process, and manipulate data—and there's no shortage of thoughtful commentary outlining the types of databases I should prefer. But these evaluations, which typically discuss

Share on

other key consideration: how hard is it for analysts to write queries against these databases?

Relative to factors like processing speed and scalability, the particulars of a database's query language may seem trivial. Regardless of how hard it is to get through a race car's door, won't it always be faster than a Prius?

For many analysts, however, we aren't always driving 500 miles as fast as we can—we're making lots of short trips to the grocery store. When you work with a database day in and day out, the annoyances that hinder every quick project—how do I get the current time in Redshift? `NOW()`? `CURDATE()`? `CURDATE`? `SYSDATE`? `WHATDAYISIT`?—often slow you down more than a lower top speed.

With this in mind, I decided to approach the "which database is best?" question from a different angle. I wanted to find the easiest database to query.

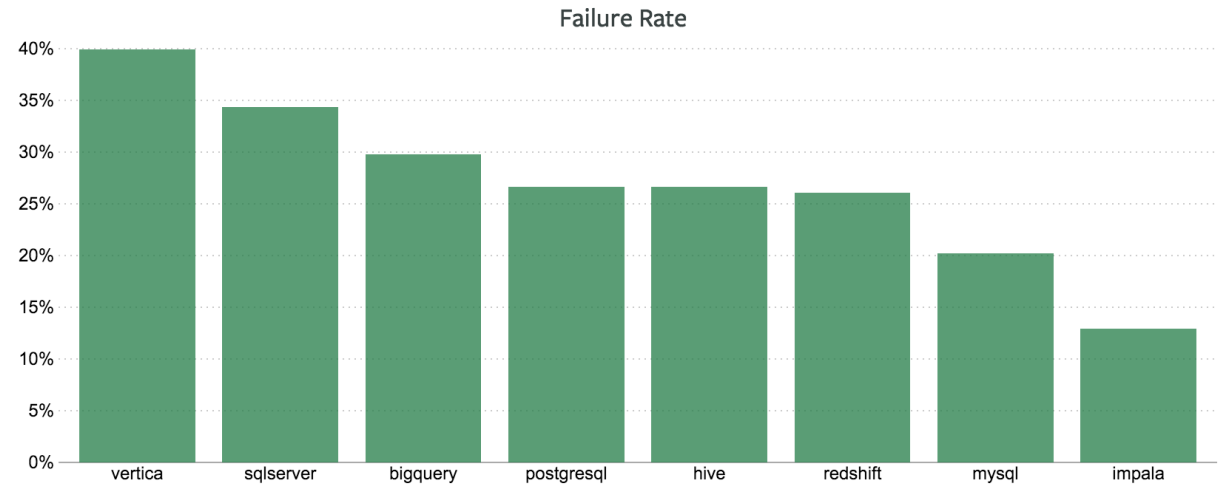
I looked into the question as any analyst would—by using data. Analysts write thousands of queries in a number of different languages in Mode every day. Though Mode supports 11 types of databases, my analysis focused on the eight most popular: MySQL, PostgreSQL, Redshift, SQL Server, BigQuery, Vertica, Hive, and Impala. I looked at millions of queries run in Mode's editor, which excludes all scheduled runs, reports run in lists, and reports run with parameters by people other than the query's authors.

So which databases, despite how fast they go and how much they cost, have doors that are just too hard to get in and out of?

A basic measure of difficulty

The most basic indicator that an analyst is having trouble with a query is when it fails. These error messages, (constantly) rejecting bad syntax, misnamed functions, or a misplaced comma, probably provide the truest indication of how much a language frustrates an analyst.

I started simple, looking at how often queries fail. As it turns out, Vertica and SQL Server have the highest error rates and MySQL and Impala the lowest. The chart below shows the error rates for each database.



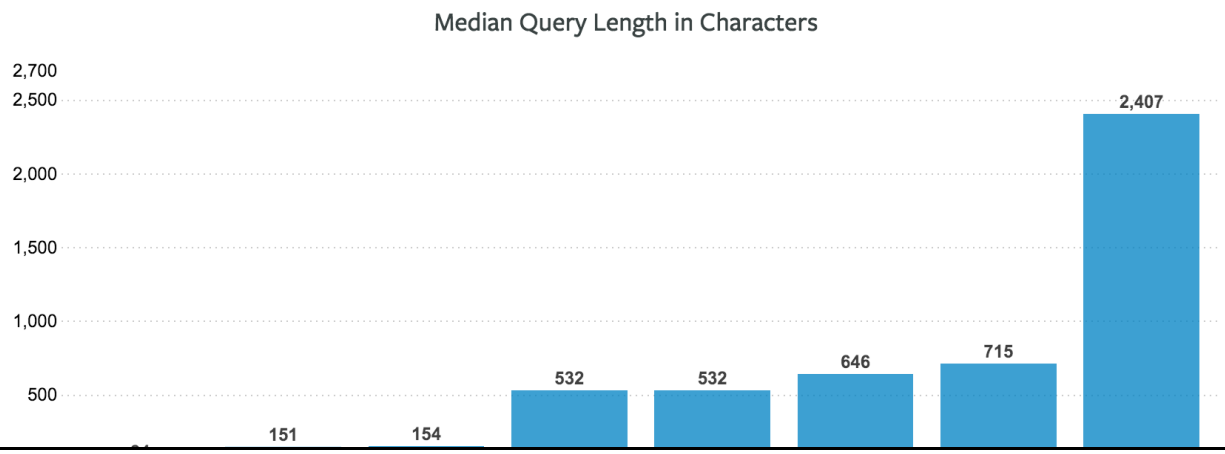
Unfortunately for our wallets (Impala, MySQL, and Hive are all open-source and free, while Vertica, SQL Server, and BigQuery are decidedly not), rates like these are probably too crude to be conclusive.

People use databases for different things. Vertica and SQL Server are proprietary databases provided by major vendors, and most likely used by large businesses with deeper analytical budgets. The high error rates from these languages may come from a more ambitious use of the language rather than the language being "harder."

Controlling for query complexity

Can we then adjust for how complex a query is? Unfortunately, controlling for query complexity is hard.

Query length could be a decent proxy, but it's not perfect.

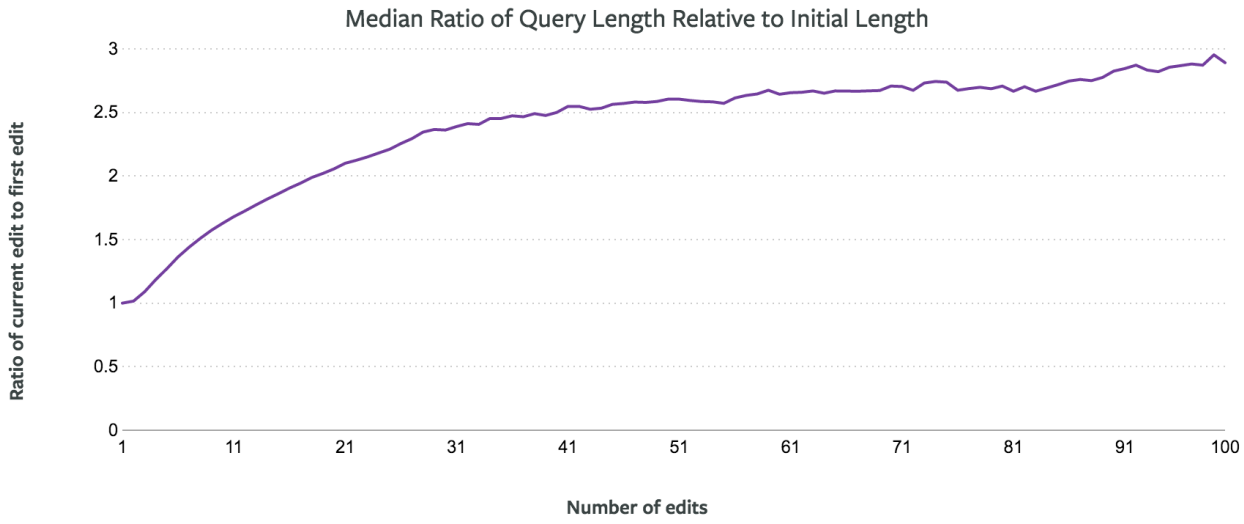


Share on

An easy language may be easy *because* it's concise. Or, as anyone who's attempted to parse a string of seemingly random brackets, backslashes, and periods in a regular expression will tell you, a language may be *hard* because it's concise.

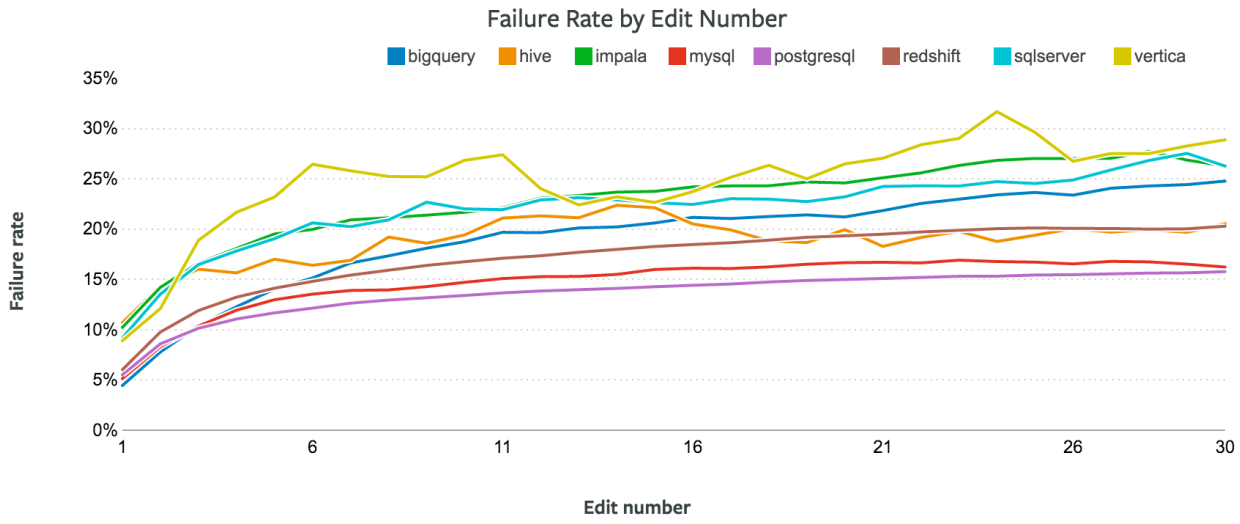
While there are clear differences in query lengths across different languages, the relationships between query length, query complexity, and language difficulty are all intertwined. Figuring out these relationships sounds even more daunting than parsing regex.

But we may be able to control for complexity in other ways. Queries often evolve over the course of an analysis. They start as simple explorations and become more complex as analysts add layers. You can see this evolution in the chart below, which shows how the median query doubles in length after 20 or so edits, and triples after 100 edits.



Rather than comparing queries of similar lengths, we could instead compare queries at the same stage in the analytical process. How often does the first query run result in an error? The fifth? The 20th?

The chart below shows the error rates for queries by the number of times analysts have edited them. After five or so runs, a few clear patterns emerge. PostgreSQL, MySQL, and Redshift have consistently low error rates. Impala, BigQuery, and SQL Server have high error rates. And as before, Vertica consistently outpaces the rest with the highest error rate.



This suggests that more traditional versions of SQL—PostgreSQL and MySQL—are the easiest SQL languages to use. Among analytical databases, Redshift takes a clear lead over languages like Vertica and SQL Server.

Head-to-head comparisons

Query complexity, however, isn't the only factor affecting error rates. If you've seen me bulldoze through 15 syntax errors in a row, you know the skill of the analyst also matters.

Twenty percent of analysts using Mode write queries against more than one type of database. Personally, I regularly use PostgreSQL and Redshift, and sometimes MySQL and BigQuery.

These multi-lingual analysts offer us an opportunity. Among people who use different languages, which are they most comfortable with? Does an analyst who uses PostgreSQL and BigQuery tend to have higher error rates in one language or another? If we could pit SQL languages against each other (in what would surely be the nerdiest round robin tournament ever), which one would win?

I used a method of pairwise comparisons to aggregate together these head-to-head matchups:

1. I found all the analysts who've run a minimum of 10 queries per database for multiple databases.

3. I averaged the differences in error rates for every database pair to construct the matrix below.

The matrix shows the difference in error rates of the database on the top row compared to the database on the left. Here, a higher number is worse than a lower number. For example, the “20.2” at the intersection of Hive and BigQuery indicates that, among analysts who use both of those databases, the error rate tends to be 20.2% higher for Hive than BigQuery.

		How error rates for this database...							
		BigQuery	Hive	Impala	MySQL	PostgreSQL	Redshift	SQL Server	Vertica
	BigQuery		20.2	-3.4	-9.6	-6.6	2.0	19.3	8.3
	Hive	-20.2		-15.4	-11.0	-29.7	-34.5	-2.1	-19.1
..compare to	Impala	3.4	15.4		0.7	1.2	0.4	13.3	-3.7
error rates	MySQL	9.6	11.0	-0.7		7.7	8.0	9.3	-3.6
for this one	PostgreSQL	6.6	29.7	-1.2	-7.7		-4.0	5.4	6.3
	Redshift	-2.0	34.5	-0.4	-8.0	4.0		18.7	-6.1
	SQL Server	-19.3	2.1	-13.3	-9.3	-5.4	-18.7		-16.0
	Vertica	-8.3	19.1	3.7	3.6	-6.3	6.1	16.0	
	Total	-30.2	131.9	-30.6	-41.2	-35.1	-40.6	79.9	-34.0

The total score line on the bottom sums the differences for each database. The result provides a similar conclusion to the error-by-run analysis: MySQL and PostgreSQL are the easiest versions of SQL to write. Redshift also jumps up a couple spots, from the fourth easiest to the second easiest.

Vertica gains the most ground. It moves from being the most difficult language to somewhere near the middle of the pack, beating out SQL Server and Hive. This suggests that Vertica's high error rate may be more indicative of the type of analyst that uses it than it is of the language itself.

The winners

Overall, these numbers point to MySQL and PostgreSQL as the easiest versions of

uses PostgreSQL). Unfortunately for analysts, they're also poorer in features—and often slower—than languages like Vertica and SQL Server.

For analysts looking for ease of use without sacrificing too much speed—driving to the grocery store is faster than walking, after all—Redshift is the clear winner. Add the collective vote of analysts using Mode to the growing pile of recommendations.

Related Articles

General
October 10, 2016 • 2 minute read

Analytics Dispatch 044: Studying The Simpsons

General
August 6, 2020 • 2 minute read

Announcing Our Series D

General
March 25, 2020 • 12 minute read

An Analyst's Perspective on COVID-19's Economic Impact on Businesses

See all articles >

Subscribe

Contact

Request a demo
hi@modeanalytics.com

208 Utah Street, Suite 400
San Francisco CA 94103

Product

Compare plans
SQL Editor
Notebooks
Reports & Dashboards
Embedded Analytics
Mode for Slack
Customers
Integrations
Security

Resources

Resource center
Learn SQL
Learn Python
Data news
Help & support
Developer
Gallery
Blog

Company

About us
Team
Values

Share on

Share on