MySQL vs PostgreSQL for Web Applications [closed]

Asked 16 years, 3 months ago Modified 4 years, 4 months ago

Viewed 75k times

139

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 11 years ago.

I am working on a web application using Python (Django) and would like to know whether MySQL or PostgreSQL would be more suitable when deploying for production.

In one podcast Joel said that he had some problems with MySQL and the data wasn't consistent.

I would like to know whether someone had any such problems. Also when it comes to performance which can be easily tweaked?

mysql postgresql

Share

Improve this question

Follow

edited Jan 18, 2021 at 12:38

Community Bot
1 ● 1

asked Aug 26, 2008 at 5:05

cnu
37.1k ● 23 ● 66 ● 63

10 Answers

Sorted by: Highest score (default)

▲

133

▼

✓

A note to future readers: The text below was last edited in August 2008. That's nearly 11 years ago as of this edit. Software can change rapidly from version to version, so before you go choosing a DBMS based on the advice below, do some research to see if it's still accurate. Check for newer answers below.

Better?

MySQL is much more commonly provided by web hosts.

PostgreSQL is a much more mature product.

There's this discussion addressing your "better" question

Apparently, according to this web page, MySQL is fast when concurrent access levels are low, and when there are many more reads than writes. On the other hand, it exhibits low scalability with increasing loads and write/read ratios. PostgreSQL is relatively slow at low concurrency levels, but scales well with increasing load levels, while providing enough isolation between concurrent accesses to avoid slowdowns at high write/read ratios. It goes on to link to a number of performance comparisons, because these things are very... sensitive to conditions.

So if your decision factor is, "**which is faster?**" Then the answer is "it depends. **If it really matters, test your application against both.**" And if you really, really care, you get in two DBAs (one who specializes in each database) and get them to tune the crap out of the databases, and then choose. It's astonishing how expensive *good* DBAs are; and *they are worth every cent*.

When it matters.

Which it probably doesn't, so just pick whichever database you like the sound of and go with it; better performance can be bought with more RAM and CPU, and more appropriate database design, and clever stored procedure tricks and so on - and all of that is cheaper and easier for random-website-X than agonizing over which to pick, MySQL or PostgreSQL, and specialist tuning from expensive DBAs.

Joel also said in that podcast that comment would come back to bite him because people would be saying that MySQL was a piece of crap - Joel couldn't get a `count` of rows back. The plural of anecdote is not data. He said:

> MySQL is the only database I've ever programmed against in my career that has had data integrity problems, where you do queries and you get nonsense answers back, that are incorrect.

and he also said:

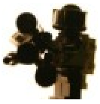
> It's just an anecdote. And that's one of the things that frustrates me, actually, about blogging or just the Internet in general. [...] There's just a weird tendency to make anecdotes into truths and I actually as a blogger I'm starting to feel a little bit guilty about this

Share Improve this answer Follow

edited Sep 3, 2019 at 11:08

Ankur Pandey
57 ● 9

answered Aug 26, 2008 at 5:14



Josh
8,016 ● 5 ● 43 ● 63

15 And another point is that, if you use a framework, you may be able to abstract your code from the database, so you could switch easily. – Don Jones Dec 7, 2008 at 23:38

"which is faster?" . Depends what you want, If you want ACID compliant Database, then read this: randombugs.com/linux/mysql-postgresql-benchmarks.html – Sacx Feb 25, 2011 at 8:42

49 Also worthy of note. MySQL is *owned* by Oracle. Therefore, it wouldn't be allowed to ever compete with their paid products. – unixman83 Mar 25, 2012 at 15:39

15

Just chiming in many months later.

The geographical capabilities of the two databases are very, very different. PostgreSQL has the exceptional PostGIS extension. MySQL's geographical functionality is practically zero in comparison.

If your web service has a location component, choose PostgreSQL.

Share Improve this answer Follow

answered Mar 12, 2010 at 23:39

Summer
2,498 ● 3 ● 23 ● 32

14

I haven't used Django, but I have used both MySQL and PostgreSQL. If you'll be using your database only as a backend for Django, it doesn't matter much, because it will abstract away most of the differences. PostgreSQL is a little more scalable because it doesn't hit the brick wall as fast as MySQL as data-size/client-count increase.

The real difference comes in if you are doing a new system. Then I'd recommend PostgreSQL hands down, because it has a lot more features which make your DB layer much more customizable, so that you can fine-tune it to any requirements you might have.

Share Improve this answer

Follow

edited Aug 1, 2020 at 21:14

Xbox One

303 ● 3 ● 15

answered Aug 26, 2008 at 5:24

Grey Panther

13.1k ● 7 ● 49 ● 67

7

Although it's a bit out of date, it would be worth reading the MySQL Gotchas page. Many of the items listed there are still true, to the best of my knowledge.

I use PostgreSQL.

Share Improve this answer

Follow

answered Aug 26, 2008 at 8:29

Greg Hewgill

990k ● 191 ● 1.2k ● 1.3k

Is this still the case? – Orkhan Alikhanov Feb 11, 2021 at 17:53

7

I use both extensively. My choice for a particular project boils down to:

- Licensing - Are you going to distribute your app (IANAL)
- Existing Infrastructure and Knowledge Base
- Any *special sauce* you have to have.

By *special sauce* I mean things like:

- Easy/cheap replication = MySQL
- Huge dataset problems with small results = PostgreSQL. Use the language extensions, and have very efficient data operations. (PL/Python, PL/TCL, PL/Perl, etc)
- Interface with R Statistical Libraries = PostgreSQL PL/R *available in debian/ubuntu*

Share Improve this answer Follow

edited Aug 1, 2020 at 20:35
Xbox One
303 ● 3 ● 15

answered Aug 5, 2009 at 0:15
Lance Rushing
7,620 ● 4 ● 30 ● 34

3 Postgres also has easy/cheap replication. It's just not *built in*, because there a half dozen to choose from, each with their own good and bad parts. So you can choose the *one* mediocre replication for MySQL, or from one of many for PostgreSQL. Your choice. – Randal Schwartz Mar 13, 2010 at 0:11

6 PostgreSQL 9.0+ has built-in binary WAL replication that is very easy to setup. – Sean Jul 5, 2011 at 23:59

2

Well, I don't think you should be using a different database brand in anything past development (build, staging, prod) as that will come back to bite you.

From how I understand it PostgreSQL is a more 'correct' database implementation while mySQl is less correct (less compliant) but faster.

So if you are pretty much writing a CRUD application mySQL is the way to go. If you require certain features out of your database (if you're not sure then you don't) then you may want to look into postgreSQL.

Share Improve this answer

Follow

answered Aug 26, 2008 at 5:14

SCdF
59.3k ● 24 ● 79 ● 114

1

If you are writing an application which may get distributed quite a bit on different servers, MySQL carries a lot of weight over PostgreSQL because of the portability. PostgreSQL is difficult to find on less than satisfactory web hosts, albeit there are a few. In most regards, PostgreSQL is slower than MySQL, especially when it comes to fine tuning in the end. All in all, I'd say to give PostgreSQL a shot for a short amount of time, that way you aren't completely avoiding it, and then make a judgement.

Share Improve this answer

Follow

edited Aug 1, 2020 at 10:25

Ryan M ♦
20k ● 34 ● 73 ● 82

answered Aug 26, 2008 at 5:23

tslocum
3,422 ● 5 ● 32 ● 33

1 I'll agree with you that more low-quality web-hosts should offer a choice instead of shoving MySQL in our faces. However, when going with shared hosting you're at the mercy of them anyway. So a DBM choice doesn't matter as much. – unixman83 Mar 25, 2012 at 15:54

0

Thank you. I've used Django with MySQL and it's fine. Choose your database on the features you need. Hard to compare MySQL and Postgres. Better to compare Postgress to SQl Server.

Share Improve this answer

Follow

answered Aug 26, 2008 at 7:51

Stephen Cox

3,655 ● 3 ● 23 ● 16

0

@WolfmanDragon

PostgreSQL has (tiny) support for objects, but it is, by nature, a relational database. From its about page:

> PostgreSQL is a powerful, open source **relational** database system.

Share Improve this answer

Follow

answered Oct 10, 2008 at 2:34

marcospereira

12.2k ● 3 ● 48 ● 53

2 a Object-Relational Database Management System ISA Relational Database Management System. If you read farther

into PostgSQL documentation, you will find that PostgSql is an ORDBMS. Perhaps you are confusing ORDBMS with ODBMS such as Objectivity/DB. – WolfmanDragon Oct 13, 2008 at 18:54

-3

MySQL is a relational database management system while PostgreSQL is an **object**-relational database management system. PostgreSQL is suited well for C++ or Java developers, as it gives us more control over how queries are written. ORDBMS also gives us Objects and User Defined Types. The SQL queries themselves are much closer to the ISO standards than MySQL.
Do you need an ORDBMS or a RDBMS? That will better answer your question.

Share Improve this answer Follow

answered Sep 19, 2008 at 16:11
WolfmanDragon
7,952 ● 14 ● 51 ● 61

That's kinda of an oversimplification; just because you don't need an ORDBMS doesn't necessarily mean you must go with MySQL. There are other more significant differences, no? – Don Jones Dec 7, 2008 at 23:42

1 You are correct. I'm trying not to sound like a PostgreSQL pusher. I see no real reason to use anything other than PostgreSQL unless the design calls for a true object database. I see no point in MySQL other than more people are familiar with it. – WolfmanDragon Dec 9, 2008 at 1:29