10 Deadly PostgreSQL Mistakes

Every company is different. The missteps they make when designing their PostgreSQL database, however, are often not. Below, we've compiled our years of helping companies successfully use PostgreSQL.

#1 Tool Selection
Tools are supposed to help us do its job. For example, one tool is covering another, or the wrong tool is being used. To avoid this, we recommend:

- Check default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#2 Not Tuning the Workload
You may be using PostgreSQL on a larger scale than is needed for longer than expected.

- Remember to check your WAL settings, use automated management, and use tools to determine if and when you can have the opposite effect.
- Check the default settings on your database and use case.
- Use proper security practices. Common mistakes include not using effective roles, and storing everything in the public schema.
- Implement database security best practices. Common mistakes include not using SSL to encrypt traffic, over-granting permissions or misusing roles.
- Audit your activity and check that security best practices are followed.
- Getting database security right is crucial. Regularly audit your activity and check that security best practices are followed.

#3 Poor Database Design
A complicated database design doesn’t mean an optimized database. Understand how you need your database to work so you can choose the right tool.

- Check default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#4 Under-Indexing
Extensions are wonderful, but additional work for the database.

- Check the default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#5 Over-Under-Indexing
The right index speeds up performance. But over- and under-indexing can create additional work for the database.

- Check the default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#6 Improper Use of Vacuum
Vacuuming compacts and optimizes dead space, which can greatly speed up performance. But when a vacuum is running, it can often cause a performance hit.

- Check the default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#7 Poor Connection Management
Every connection is a new process, and there is a high overhead for new connections.

- Check the default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#8 Over-/Under-indexing
Under-indexing can create additional work for the database.

- Check the default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#9 Lax Security Practices
A large number of open connections can lead to severe performance degradation over time.

- Check the default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

#10 Mistakes
The right index speeds up performance. But over- and under-indexing can create additional work for the database.

- Check the default settings on your database and use case.
- Understand how you need your database to work so you can choose the right tool.
- Make sure you understand the extensions you want to add to your implementation. Stick with reliable extensions.
- Check the tool versions you are using; an old tool can hamper productivity.
- Make sure you are using the right tools for the job.

WEBINAR: Using Vacuum to Clean Up PostgreSQL for Performance
https://learn.percona.com/using-vacuum-to-clean-up-postgres-for-performance

WEBINAR: PostgreSQL Security Missteps and Tips

WEBINAR: PostgreSQL Database Design Best Practices
https://www.percona.com/resources/webinars/postgresql-database-design-best-practices