

When a WordPress site goes down, you might expect a plain “500 Internal Server Error.” Instead, the browser may show “Error establishing a database connection” or a similar message. That can seem confusing: isn’t a database failure a server failure?

It is—but the message you see depends on which part of the request failed and how the web server and WordPress handled it. A database connection error is a specific application-level failure. A 500 status is a general HTTP response meaning the server could not complete the request. They describe different things, and a site can show a database error page while returning a 500 status—or, depending on the configuration, a different status.

This guide explains the difference, how to triage the outage quickly, and how to check WordPress database settings without making a risky change in the dark.

Database Error vs. 500 Error: What’s the Difference?

A 500 error is an HTTP status code. It tells the browser or monitoring service that something went wrong on the server, but it does not identify the cause. A PHP fatal error, a server misconfiguration, or an application failure might all lead to a 500 response.

A WordPress database connection error is a message generated when WordPress cannot establish or use its database connection. Before WordPress can show posts, pages, settings, and users, it needs to communicate with MySQL or MariaDB. If that connection fails, WordPress may stop processing the request and display a database-related message.

Think of the distinction this way:

- **Database error:** A message describing the failure WordPress encountered.
- **HTTP 500:** A status code describing the server’s response to the request.
- **Server response differences:** The page text and HTTP status are separate outputs. One does not reliably predict the other.

Depending on the WordPress version, PHP behavior, hosting setup, and server configuration, the database error page might be delivered with a 500 status, a 200 status, or another status. A custom error page, proxy, or cache can also affect what a visitor sees. The status is worth checking, but the message alone does not tell you what status the server returned.

What Happens During the Database Handshake?

On a typical page request, the web server passes the request to PHP. WordPress loads its core files and uses the wpdb class to interact with the database. The wpdb class reads connection settings, attempts to connect to the database server, and sends queries needed to load the site.

The initial database handshake can fail for several reasons: the database host cannot be reached, the database server is overloaded or stopped, the supplied username or password is wrong, or the user does not have permission to access the database. In some cases, the connection succeeds but a later query fails because of database damage or a server-side resource problem.

That distinction matters. A failed connection is not the same as a database that is reachable but returning errors. Start by confirming whether WordPress can reach the database service before attempting repairs.. Pretty simple.

Fast, Reversible Outage Triage

During an outage, begin with checks that do not change the site. Record the exact error text, the time it first appeared, and any recent changes. This creates a useful incident record and helps distinguish a site-wide outage from a problem affecting one browser, one page, or one region.

1. **Check the site from another connection.** Try a mobile connection or a separate device. If only one network fails, DNS, browser cache, or a local network issue may be involved.
2. **Check more than one URL.** Test the homepage and a known WordPress page. Note whether the WordPress admin area shows the same error.
3. **Check the HTTP response status.** Use browser developer tools or run `curl -I https://example.com` from a terminal, replacing the address with your site. Record the status code and any useful response headers. Do not assume the visible error message is the status code.
4. **Check recent changes.** Look for a database password rotation, hosting migration, DNS change, WordPress update, plugin update, or server maintenance around the time of the first failure.
5. **Capture exact evidence.** Save the message, timestamp and timezone, affected URLs, HTTP status, and relevant PHP or database log entries. Redact passwords and other secrets before sharing the records.

If the site is business-critical, preserve a copy of the current files and database before editing configuration or running repair commands. If backups are managed by the host, verify the restore point and its date rather than assuming a backup exists.

Observation What it may suggest First safe check Database error on every page Connection settings, database service, or hosting issue Check server health, credentials, and database logs Only one page fails A page-specific query, plugin, or theme issue Compare the failing page with the homepage and review logs Intermittent database errors Load, connection limits, network interruptions, or resource pressure Record timestamps and compare them with database and hosting metrics HTTP 500 with a PHP fatal error PHP execution failure, which may or may not involve the database Check the PHP error log for the first relevant fatal message

Check Hosting and Server Health

Before editing WordPress, determine whether the database server is available. A correct password cannot help if MySQL or MariaDB is stopped, unreachable, or refusing connections because it has reached a limit.



Want to know something interesting? check your hosting control panel for database service alerts, account resource usage, disk space, and scheduled maintenance. If you have authorized server access, inspect the database service status and its logs. Look for messages about failed starts, “too many connections,” exhausted disk space, memory pressure, or network errors around the recorded outage time.

Also check whether the web server and PHP are healthy. If a PHP process cannot run, WordPress may return a generic 500 error without reaching the database connection stage. If a reverse proxy or content delivery network sits in front of the site, compare its response with the origin server only if you are authorized and know the correct origin address. Avoid changing DNS during triage: it can make diagnosis harder and may prolong an outage.

Verify the Database Credentials in wp-config.php

WordPress normally stores its database settings in the wp-config.php file in the site’s WordPress directory. The key definitions are commonly shown in this form:

```
define( 'DB_NAME', 'database_name' ); define( 'DB_USER', 'database_user' ); define( 'DB_PASSWORD',  
'database_password' ); define( 'DB_HOST', 'localhost' );
```

The actual values are specific to your hosting account. Do not copy example values into a live site. A correct database name and password can still fail if DB_HOST is wrong. Hosts may use localhost, a hostname, or a host-and-port value; some use a Unix socket or a database server on a separate machine.

Safe credential-check procedure

1. Locate the correct wp-config.php for the site that is failing. Multiple installations can have similar-looking directories.
2. Make a private copy of the current file before editing it. Keep the copy outside the public web directory if possible.
3. Compare the database name, username, password, and host with the values in the hosting control panel or the database account’s source-of-truth records.
4. Check whether the database user is assigned to the correct database and has the required privileges. Having a valid account is not enough if it cannot access that database.

5. If a value is wrong, change only that value, save the file, and immediately test the homepage and admin area.

Be careful with quotation marks, semicolons, and special characters when editing PHP. A missing quote or semicolon can create a PHP error and turn one outage into another. Do not paste credentials into a public support ticket or an unprotected chat. If you must share configuration for review, replace all secret values first.

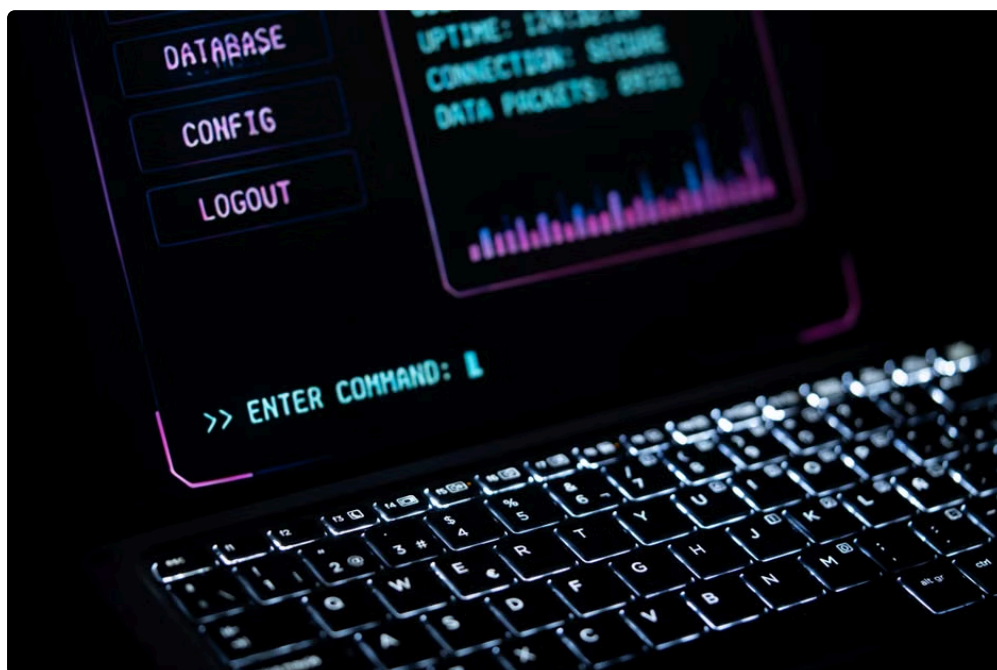
What to record for a credential change

- **What changed:** The specific database setting you corrected, such as DB_HOST or DB_PASSWORD.
- **How to roll back:** Restore the saved original wp-config.php if the edit causes a new error. If a password was rotated, coordinate the rollback with the database account so the WordPress value and database account remain in sync.

Other Common Causes of a WordPress Database Error

Database service unavailable or overloaded

The database may be stopped, restarting, overloaded, or unable to accept more connections. Repeatedly refreshing the site will not fix a service issue and may add load. Check service status and timestamps first. If you have server access, look at the database service logs and resource metrics rather than restarting services without evidence.



Wrong database host or network path

A site migration or hosting change can leave the old database host in wp-config.php. If the database is remote, network rules or a changed hostname can prevent the handshake. Confirm the host value with the hosting configuration and check whether the site server is allowed to connect from its current address.

Insufficient privileges

The database user may exist but lack access to the WordPress database. In the hosting panel, verify the user-to-database assignment and required privileges. Avoid granting broad access to unrelated databases as a shortcut.

Database damage or query failures

If logs show that WordPress connects successfully but a table or query fails, the issue may involve database integrity, a plugin, or a specific schema problem rather than the initial handshake. Preserve a backup before attempting repairs. WordPress includes a database repair mechanism, but enabling or running it should be done deliberately and removed or disabled afterward if it is no longer needed.

Why a 500 Can Appear Instead—or Alongside the Database Message

WordPress, PHP, and the web server each have an opportunity to handle a failure. If PHP crashes before WordPress can display its database message, the server may show a generic 500 page. If WordPress catches the connection problem and renders its own message, visitors may see that message. A proxy or host-provided error handler may replace the page, and the status code may differ from what you expect.

For accurate diagnosis, check both the content and the response: note the page text, HTTP status, and matching PHP and database log entries. A 500 status alone does not prove the database is broken, and a database error page alone does not prove the server returned HTTP 500.

When to Escalate—and What to Send

Escalate when the database service is unavailable, credentials are confirmed but the connection still fails, or the logs show resource limits or database errors you cannot safely resolve. Instead of sending a vague request for help, provide a concise incident report. Use this template:

Site: [domain] First observed: [date, time, timezone] Affected URLs: [URLs] Exact message: [copy of message] HTTP status: [status from curl or browser tools] Recent changes: [updates, migration, password rotation, or none known] Checks completed: [wp-config values verified without sharing secrets; service status; logs] Relevant log excerpt: [redacted excerpt and timestamp] Backups: [available restore point and date]

Ask the administrator or hosting support to confirm whether the database service was reachable from the web server at the reported time, whether connection limits or disk space were exhausted, and whether the **fantasticlawyers** database logs show authentication or network failures. Do not include the database password in the report.

Keep the Fix Controlled

During an outage, avoid changing several things at once, installing troubleshooting plugins, or making unverified edits. Those actions can obscure the original cause and introduce new failures. Make one evidence-based change at a time, test immediately, and retain a way to reverse it.

In short, WordPress shows a database connection message because its database access failed at the application level. A 500 error is a separate HTTP response, and the two can overlap. Check the response status, capture exact timestamps, verify hosting health, and carefully confirm the wp-config.php values before changing anything. That sequence is faster, safer, and more likely to reveal the real cause than treating every WordPress outage as the same generic server error.