

A WordPress database connection error means the site's PHP code could not connect to the database it needs to load pages. On Windows hosting, the cause may be incorrect database details, a database service problem, or an IIS/PHP configuration issue. The message does not, by itself, tell you which one.

Start with the fastest reversible checks. Record the exact error and the time it began, then check whether the site is down for everyone and whether the hosting control panel reports a database or server issue. Do not start by installing plugins, changing several settings at once, or editing files without a backup. Those steps can make the cause harder to identify and complicate rollback.

1. Triage the outage before changing anything

First, capture what you can see. Note the exact message, the first time you observed it, and whether it appears on every page or only in the WordPress admin area. Save a screenshot if possible. If the wording changes—for example, to “Error establishing a database connection”—record that too.

Next, test the site from a private browser window or a different network. This helps rule out a local browser or cached-page issue. Check both the homepage and a known inner page. If the site loads for some visitors but not others, note that detail; it may point to an intermittent server or network problem rather than a permanent credential mistake.

- Check whether the hosting control panel is reachable.
- Look for a database status page, outage notice, or recent maintenance message.
- Ask a colleague or use a separate connection to confirm whether the failure is public.
- Record any recent change, such as a password update, hosting migration, PHP version change, or domain move.

If you have access to logs, save a copy before making changes. Useful sources may include the hosting control panel's PHP error log, IIS logs, or WordPress debug log if it was already enabled. Do not turn on public error display on a live site: database errors can expose technical details to visitors.

Keep a small incident scratchpad

A short timeline prevents guesswork, especially when multiple people are investigating. Use a private note, not a public support ticket containing passwords.

RecordExample Time first observed2026-10-15, 09:20 local time Exact visitor message“Error establishing a database connection” Recent changesDatabase password changed at 08:55 Checks performedControl panel reachable; database status unknown Change and resultUpdated DB_PASSWORD; homepage recovered at 09:42

2. Confirm the database type and hostname

Before changing credentials, confirm what kind of database the hosting account provides. Standard WordPress installations use MySQL or a compatible service such as MariaDB. WordPress does not normally connect to Microsoft SQL Server as its database. The fact that a website runs on Windows or IIS does not mean that its database is SQL Server.

SQL Server vs MySQL: why the distinction matters

SQL Server and MySQL are different database products. They use different drivers and connection methods. If the host's control panel lists a MySQL or MariaDB database, use those database details in WordPress. If it lists only Microsoft SQL Server, verify that the hosting plan supports a standard WordPress installation before attempting a repair. Do not put a SQL Server name into DB_HOST and expect a typical WordPress setup to connect.

Find the database name, username, password, and database hostname in the hosting control panel. The database hostname is the server address WordPress must contact. It may be localhost, but on shared Windows hosting it can instead be a named server, a network address, or a hostname with a port. Use the exact value provided by the host; guessing localhost is a common cause of failure.

Also distinguish the website's hosting account from the database account. The FTP or control-panel password is not necessarily the database password. A database name or username may include an account prefix, such as accountname_db1. Copy the full value, including the prefix, and watch for accidental spaces when pasting.

3. Check WordPress database settings safely

WordPress stores its database connection settings in wp-config.php, typically in the site's document root. On Windows hosting, you may access it through the hosting file manager, FTP/SFTP, or a deployment tool. Before editing it, download a copy of the current file and save it somewhere secure. That gives you a simple rollback path if the edit introduces a syntax error.





The core settings usually look like this:

```
define( 'DB_NAME', 'your_database_name' ); define( 'DB_USER', 'your_database_user' ); define( 'DB_PASSWORD',  
'your_database_password' ); define( 'DB_HOST', 'your_database_hostname' ); define( 'DB_CHARSET', 'utf8mb4' );  
define( 'DB_COLLATE', '' );
```

Replace the example values with the exact details from the control panel. Keep the quotes and semicolons intact. Do not post this file, its contents, or a screenshot showing credentials in a public forum or an unsecured support ticket.

Compare, do not guess

1. Open the database details in the hosting control panel.
2. Compare each value with the matching constant in wp-config.php.
3. Check that the hostname includes any required port or instance information, exactly as supplied.
4. Confirm the database user has permission to access that database.
5. Save only if you identify a specific mismatch, then test the site immediately.

If the database password was recently changed, the value in wp-config.php must match the new password. If you are unsure whether a password is correct, use the control panel's database tools or ask the provider to confirm or reset it using its secure process. Avoid repeatedly guessing passwords; failed attempts can trigger account protections on some systems.

What changed: For a credential repair, change only the incorrect constant or constants, such as DB_PASSWORD or DB_HOST. **How to roll back:** Restore the saved original wp-config.php if the edit causes a new error. Keep a note of the exact value changed, without recording the password in an insecure log.

4. Test database availability and permissions

If the values match the control panel, check whether the database itself is available. Use the provider's database-management tool, such as phpMyAdmin where offered, or a connection test provided in the control panel. If the tool cannot reach the same database, the issue may be service availability, network access, or account-level configuration rather than WordPress.

If the database tool works but WordPress does not, focus on whether the WordPress database user is correctly assigned to the database and has the required permissions. A database can exist and be online while the application user lacks access. Do not grant broad permissions to every database user as a workaround. Use the provider's documented WordPress permissions or request a targeted correction.

Check database storage limits and account quotas as well. A full database quota, suspended database, or exceeded hosting limit can prevent normal operations. If the failure began suddenly and no credentials changed, service health and resource limits deserve attention before editing configuration.

5. Check hosting and server health

On Windows hosting, WordPress runs as PHP under IIS and connects from the web server to a MySQL-compatible database. The relevant parts can fail independently: IIS may be running while PHP is misconfigured, or PHP may run while the database service is unavailable.

- Check the hosting dashboard for database service status, maintenance notices, and resource-limit alerts.
- Look for recent changes to the PHP version, IIS application pool, site binding, or hosting plan.
- Review PHP and IIS logs around the recorded outage time for connection, timeout, or extension errors.
- Check whether the database hostname resolves or is reachable using tools the provider makes available.
- Ask whether the web server is allowed to connect to the database server, especially after a migration or firewall change.

A timeout or connection-refused message may indicate a server-side problem, while an "access denied" message more often points to a username, password, or permission mismatch. These clues are not definitive, but they can make a support request far more useful. Share the timestamp, exact error, affected domain, and checks already completed. Never include the database password in an ordinary email or ticket.

6. Verify IIS PHP settings and required extensions

WordPress needs a compatible PHP environment, and the PHP configuration used by IIS must be the one actually serving the site. A command-line PHP installation may have different settings from the PHP version attached to the IIS site or application pool. Confirm the active PHP version in the hosting control panel or with the provider's site-specific diagnostic tools.

For a MySQL-compatible database, WordPress needs the appropriate MySQLi support in PHP. If the required extension is missing, disabled, or unavailable after a PHP update, WordPress may be unable to connect even when the database credentials are correct. Check the [error establishing database connection](#) PHP error log for messages about mysqli or a missing database driver.

Other IIS PHP settings can matter indirectly. A PHP version change may alter extension availability or configuration. An application pool identity or hosting restriction may affect access to files or network services. Do not change IIS configuration or recycle application pools blindly on a production site. First identify a relevant log message or confirm the setting with the hosting control panel.

If you have a documented, reversible PHP setting change to test, record the original version or value, make one change, and test immediately. If it does not help, restore the original setting. Avoid switching PHP versions repeatedly: each change adds uncertainty and can create compatibility problems beyond the database connection.

7. Use a safe order of operations

During an outage, the goal is to restore service without losing evidence or making unrelated changes. Follow this sequence:

1. Record the timestamp and exact error; check the site from another browser or network.
2. Check hosting and database service status before editing files.
3. Confirm the database type, database hostname, name, user, password, and permissions in the control panel.
4. Back up wp-config.php and compare its values with the confirmed details.
5. Make one targeted change only if you have identified a mismatch.
6. Test the homepage and admin area, then check logs for new errors.
7. If the site remains down, provide the provider with the evidence and a clear request for specific checks.

For a provider support request, use a concise script such as:

“Our WordPress site at [domain] began showing ‘[exact error]’ at [time and time zone]. We confirmed the database type is [MySQL/MariaDB], and the control panel reports the database hostname as [hostname—omit credentials]. The settings in wp-config.php match the control panel, and [database tool test result]. Please check database service health, user permissions, connection reachability from the IIS web server, account quotas, and any recent PHP/IIS or firewall changes. Please do not reset credentials without notifying us.”

8. What not to do during a database outage

- Do not install a “database repair” plugin; WordPress may not be able to load plugins while disconnected.
- Do not delete or recreate the database unless you have a verified, restorable backup and a clear recovery plan.
- Do not copy configuration from another site without checking its database details.
- Do not expose credentials in public logs, screenshots, or support requests.
- Do not change the database hostname, PHP version, permissions, and plugins all at once.

If the database is damaged rather than unreachable, use a verified backup and the host’s recommended recovery process. Repair operations can alter data, so take a database backup first whenever the database is accessible. A connection error alone is not proof that the database needs repair.

After the site is back online

Confirm that both the homepage and WordPress admin area load, and test a few important pages or forms. Check the logs again to make sure the original error has stopped. Record what fixed the issue, when it was changed, and how to reverse the change. If credentials were exposed during troubleshooting, rotate them securely and update wp-config.php in a controlled change.

The quickest reliable recovery is usually a careful comparison of the database hostname and credentials, followed by a check of database health and IIS/PHP configuration. Keep the change set small, preserve a backup before editing, and use timestamps and exact error messages to separate a WordPress configuration problem from a hosting service issue.