MySQL (/sql/docs/mysql/manage-connections) | **PostgreSQL** | SQL Server (/sql/docs/sqlserver/manage-connections)

This page provides best practices and language-specific code samples to help you create applications that use Cloud SQL database connections effectively.

These samples are excerpts from a complete App Engine application available to you on GitHub. Learn more (#app-links).

A connection pool is a cache of database connections that are shared and reused to improve connection latency and performance. When your application needs a database connection, it borrows one from its pool temporarily; when the application is finished with the connection, it returns the connection to the pool, where it can be reused the next time the application needs a database connection.

When you use a connection pool, you must open and close connections properly, so that your connections are always returned to the pool when you are done with them. Unreturned or "leaked" connections are not reused, which wastes resources and can cause performance bottlenecks for your application.

Every database connection uses client and server-side resources. In addition, Cloud SQL imposes overall connection limits that cannot be exceeded. Creating and using fewer connections reduces overhead and helps you stay under the connection limit.

If your application attempts to connect to the database and does not succeed, the database could be temporarily unavailable. In this case, sending too many simultaneous connection requests might waste additional database resources and increase the time needed to recover. Using exponential backoff prevents your application from sending an unhealthy number of connection requests when it can't connect to the database.

There are many reasons why a connection attempt might not succeed. Network communication is never guaranteed, and the database might be temporarily unable to respond. Your application should handle broken or unsuccessful connections gracefully.

Limiting a connection's lifetime can help prevent abandoned connections from accumulating. You can use the connection pool to limit your connection lifetimes.

To see the complete application, click the link below.

- Learn more about Private IP (/sql/docs/postgres/private-ip).
- Learn about quotas and limits (/sql/docs/postgres/quotas) for Cloud SQL and App Engine.
- Learn about best practices (/sql/docs/postgres/best-practices) for working with Cloud SQL.
- Learn more about connecting from an external application (/sql/docs/postgres/connect-external-app).