

¿Qué es una bitácora (log)?

Se usa el término log, historial de log o registro a la grabación secuencial en un archivo o en una base de datos de todos los acontecimientos (eventos o acciones) que afectan a un proceso particular (aplicación, actividad de una red informática, etc.). De esta forma constituye una evidencia del comportamiento del sistema.

Por derivación, el proceso de generación del log se le suele llamar guardar, registrar o loguear y al proceso o sistema que realiza la grabación en el log se le suele llamar logger o registrador.

¿Qué bitácoras permite utilizar el gestor?

-MySQL

Error log.- Problems encountered starting, running, or stopping mysqld

General query log.- Established client connections and statements received from clients

Binary log.- Statements that change data (also used for replication)

Relay log.- Data changes received from a replication master server

Slow query log.- Queries that took more than long_query_time seconds to execute

DDL log (metadata log).- Metadata operations performed by DDL statements

-SQLite

The Error And Warning Log

¿Qué información se guarda en dichas bitácoras?

Mysql

Nombre de la transacción: Nombre de la transacción que realizó la operación de escritura.

Nombre del dato: El nombre único del dato escrito.

Valor antiguo: El valor del dato antes de la escritura.

Valor nuevo: El valor que tendrá el dato después de la escritura.

SQLite

Mensajes de Error

Comandos de administración de bitácoras en el gestor

MySQL

Modificar archivo my.cnf

```
[mysqld_safe]
log_error=/var/log/mysql/mysql_error.log

[mysqld]
log_error=/var/log/mysql/mysql_error.log

general_log_file      = /var/log/mysql/mysql.log
general_log           = 1

log_slow_queries      = /var/log/mysql/mysql-slow.log
long_query_time = 2
log-queries-not-using-indexes
```

SQLite

Solo puede haber una única devolución de llamada de registro de errores por proceso. La devolución de llamada de registro de errores se registra al inicio utilizando un código C similar al siguiente:

```
sqlite3_config ( SQLITE_CONFIG_LOG , errorLogCallback, pData);
```

La función de devolución de llamada del registrador de errores podría tener este aspecto:

```
void errorLogCallback (void * pArg, int iErrCode, const char * zMsg) {
    fprintf (stderr, "(% d)% s \ n", iErrCode, zMsg);
}
```