
See Also [ACID](#), [concurrency](#), [isolation level](#), [locking](#), [transaction](#).

locking read

A `SELECT` statement that also performs a **locking** operation on an `InnoDB` table. Either `SELECT ... FOR UPDATE` or `SELECT ... LOCK IN SHARE MODE`. It has the potential to produce a **deadlock**, depending on the **isolation level** of the transaction. The opposite of a **non-locking read**. Not allowed for global tables in a **read-only transaction**.

`SELECT ... FOR SHARE` replaces `SELECT ... LOCK IN SHARE MODE` in MySQL 8.0.1, but `LOCK IN SHARE MODE` remains available for backward compatibility.

See [Section 15.7.2.4, “Locking Reads”](#).

See Also [deadlock](#), [isolation level](#), [locking](#), [non-locking read](#), [read-only transaction](#).

log

In the `InnoDB` context, “log” or “log files” typically refers to the **redo log** represented by the `ib_logfileN` files. Another type of `InnoDB` log is the **undo log**, which is a storage area that holds copies of data modified by active transactions.

Other kinds of logs that are important in MySQL are the **error log** (for diagnosing startup and runtime problems), **binary log** (for working with replication and performing point-in-time restores), the **general query log** (for diagnosing application problems), and the **slow query log** (for diagnosing performance problems).

See Also [binary log](#), [error log](#), [general query log](#), [ib_logfile](#), [redo log](#), [slow query log](#), [undo log](#).

log buffer

The memory area that holds data to be written to the **log files** that make up the **redo log**. It is controlled by the `innodb_log_buffer_size` configuration option.

See Also [log file](#), [redo log](#).

log file

One of the `ib_logfileN` files that make up the **redo log**. Data is written to these files from the **log buffer** memory area.

See Also [ib_logfile](#), [log buffer](#), [redo log](#).

log group

The set of files that make up the **redo log**, typically named `ib_logfile0` and `ib_logfile1`. (For that reason, sometimes referred to collectively as **ib_logfile**.)

See Also [ib_logfile](#), [redo log](#).

logical

A type of operation that involves high-level, abstract aspects such as tables, queries, indexes, and other SQL concepts. Typically, logical aspects are important to make database administration and application development convenient and usable. Contrast with **physical**.

See Also [logical backup](#), [physical](#).

logical backup

A **backup** that reproduces table structure and data, without copying the actual data files. For example, the `mysqldump` command produces a logical backup, because its output contains statements such as `CREATE TABLE` and `INSERT` that can re-create the data. Contrast with **physical backup**. A logical backup offers flexibility (for example, you could edit table definitions or insert statements before restoring), but can take substantially longer to **restore** than a physical backup.

See Also [backup](#), [mysqldump](#), [physical backup](#), [restore](#).

loose_

A prefix added to `InnoDB` configuration options after server **startup**, so any new configuration options not recognized by the current level of MySQL do not cause a startup failure. MySQL processes configuration options