MySQL

MySQL Knowledge

- Percona Tools
- Alter Table

Percona Tools

Percona Tools can help perform schema updates on a busy DB without needing to take it offline. You will need enough performance overhead to perform the migration and keep up with normal traffic at the same time.

> **WARNING:** This still requires getting a table metadata lock at the start so it can be blocked by long transactions. You will need to kill those for this to continue. Sometimes this means using native MySQL is just as good as Percona.

> **WARNING:** Altering a busy DB table with lots of incoming foreign keys can cause hours of downtime due to all incoming foreign PKs locking their tables during rebuild. Options are: 1) Drop these FKs. 2) Use drop_swap option (dangerous) 3) take down application & DB during update

Tool name: pt-online-schema-change

Use this tool if a standard alter MySQL command fails (eg. due to duplicate key)

The below `pt-online-schema-change` sample is a good starting place, but you should become familiar with all the options. This avoids the potentially dangerous `drop_swap` option for foreign keys. While it can be a lot faster, it can also cause damage. When possible, avoid foreign keys.

Sample Usage:

1. First collect the **ALTER** commands you plan to perform, excluding the DB and table name. This should be MySQL's native alter format. You can use Workbench to help build your alter query.
    1. eg: `ADD COLUMN qty_test INT(11) NOT NULL DEFAULT 0`
    2. **DO NOT** include tick marks around the field names. If you must because of special characters in field/index names, ensure the alter command is surrounded by single quotes, not double
    3. If you have a `CREATE INDEX` command, reformat it to look like this:
        1. `ADD [UNIQUE] INDEX idx_name (col_1, col_2, ...) USING BTREE`
    4. If you have multiple commands, separate them with a comma
2. Build your command like so:
    1. pt-online-schema-change --alter "**ALTER_COMMAND**" --alter-foreign-keys-method rebuild_constraints 'h=**MYSQL_HOST**,P=3306,u=**DB_USER**,p=**DB_PASS**,D=**SCHEMA**,t=**TABLE_NAME**' --max-load Threads_running=300 --critical-load Threads_running=450 --set-vars

innodb_lock_wait_timeout=10,foreign_key_checks=0
2. Fill in your **MYSQL_HOST** and **SCHEMA** name
3. Fill in your credentials in **DB_USER** and **DB_PASS**
    1. NOTE: You must user a user with the following privileges: DROP, REPLICATION SLAVE (If replication is enabled)
4. In **ALTER_COMMAND** - put your alter command to be run
5. in **TABLE_NAME** put the table to alter
6. You must add either **--dry-run** or **--execute** at the end.
7. Other options are available as well:
    1. **--max-load Threads_running=300 --critical-load Threads_running=450**
        1. This allows the script to run under busier conditions and pause if taking too many resources instead of exiting early. Max Load specifies when the script pauses, Critical load specifies when the script terminates.
        2. You can leave these off, or tune them to your performance needs
    2. **--set-vars innodb_lock_wait_timeout=10**
        1. We prevent lock wait timeout errors by increasing the innodb lock wait timeout from the default of 1 second

        Percona Toolkit Documentation

1. A Dry Run is recommended first!
2. This may take a while, to prevent it from halting early if your ssh session ends use the following: `nohup COMMAND &`

Side effects: Foreign keys will be renamed. Usually by adding an underscore at the beginning

Find Duplicate & Unneeded Indexes

This tool will find and output a list of DB indexes that are unnecessary. Review each one manually and determine if it really should be removed.

There can be a lot of output, so it's recommended to pipe this into a file like the example below does.

```
pt-duplicate-key-checker --host MYSQL_HOST --user DB_USER --password DB_PASS --databases SCHEMA > duplicate_indexes.txt
```

Alter Table

These are native MySQL Alter Table recommendations for 8.0 and above production systems. If they don't fit your needs, check out Percona Tools as well.

The advice below will always suggest the least-blocking option. In general:

- `ALGORITHM=INPLACE` - Requires a metadata table lock when the transaction starts and finishes. Otherwise will not block the table. If you have active long-running transactions you will need to kill them at the start and end of processing.
- `ALGORITHM=INSTANT` - Does not require any locking. This is always the best option but only supported in a few scenarios.
- `ALGORITHM=COPY` - Blocks the table during rebuild. Consider using pt-online-schema-change if you don't have (many) foreign keys. Cannot use LOCK=NONE with this algorithm.

Add Index

This alter command allows adding an index to an active table in production.

Uses `ALGORITHM=INPLACE`

```
ALTER TABLE table_name
  ALGORITHM=INPLACE,
  LOCK=NONE,
  ADD INDEX some_index_name (field,list,of,index,cols)
```

Modify Column - Non FK

This alter command allows modifying a column type in production.

Generally you can only use INPLACE for things like slightly extending a varchar type, or other scenarios where the column width/type doesn't change. You can always try INPLACE without fear and you will get an error if it's not allowed.

> Warning: This option won't work for columns that are part of a foreign key.

Uses `ALGORITHM=INPLACE` or `COPY`

```
ALTER TABLE my_table
  ALGORITHM=INPLACE,
  LOCK=NONE,
  MODIFY COLUMN user_name VARCHAR(45) NOT NULL
```

> Some types of updates that extend the data length may trigger the following error. In this case you can either use COPY which will block the table, or use pt-online-schema-change.
>
> SQL Error [1846] [0A000]: ALGORITHM=INPLACE is not supported. Reason: Cannot change column type INPLACE. Try ALGORITHM=COPY.