# 9.2.2 Identifier Case Sensitivity

**Section Navigation** [Toggle]

- 9.2 Schema Object Names
- 9.2.1 Identifier Qualifiers
- 9.2.2 Identifier Case Sensitivity
- 9.2.3 Mapping of Identifiers to File Names
- 9.2.4 Function Name Parsing and Resolution

In MySQL, databases correspond to directories within the data directory. Each table within a database corresponds to at least one file within the database directory (and possibly more, depending on the storage engine). Triggers also correspond to files. Consequently, the case sensitivity of the underlying operating system plays a part in the case sensitivity of database, table, and trigger names. This means such names are not case sensitive in Windows, but are case sensitive in most varieties of Unix. One notable exception is Mac OS X, which is Unix-based but uses a default file system type (HFS+) that is not case sensitive. However, Mac OS X also supports UFS volumes, which are case sensitive just as on any Unix. See Section 1.8.1, “MySQL Extensions to Standard SQL”. The `lower_case_table_names` system variable also affects how the server handles identifier case sensitivity, as described later in this section.

> **Note**
>
> Although database, table, and trigger names are not case sensitive on some platforms, you should not refer to one of these using different cases within the same statement. The following statement would not work because it refers to a table both as `my_table` and as `MY_TABLE`:
>
> ```
> mysql> SELECT * FROM my_table WHERE MY_TABLE.col=1;
> ```

Column, index, stored routine, and event names are not case sensitive on any platform, nor are column aliases.

However, names of logfile groups are case sensitive. This differs from standard SQL.

By default, table aliases are case sensitive on Unix, but not so on Windows or Mac OS X. The following statement would not work on Unix, because it refers to the alias both as `a` and as `A`:

```
mysql> SELECT col_name FROM tbl_name AS a
    -> WHERE a.col_name = 1 OR A.col_name = 2;
```

However, this same statement is permitted on Windows. To avoid problems caused by such differences, it is best to adopt a consistent convention, such as always creating and referring to databases and tables using lowercase names. This convention is recommended for maximum portability and ease of use.

How table and database names are stored on disk and used in MySQL is affected by the `lower_case_table_names` system variable, which you can set when starting **mysqld**. `lower_case_table_names` can take the values shown in the following table. This variable does *not* affect case sensitivity of trigger identifiers. On Unix, the default value of `lower_case_table_names` is 0. On Windows the default value is 1. On Mac OS X, the default value is 2.

| Value | Meaning |
|---|---|
| 0 | Table and database names are stored on disk using the lettercase specified in the `CREATE TABLE` or `CREATE DATABASE` statement. Name comparisons are case sensitive. You should *not* set this variable to 0 if you are running MySQL on a system that has case-insensitive file names (such as Windows or Mac OS X). If you force this variable to 0 with `--lower-case-table-names=0` on a case-insensitive file system and access `MyISAM` tablenames using different lettercases, index |

| | |
|---|---|
| | corruption may result. |
| 1 | Table names are stored in lowercase on disk and name comparisons are not case sensitive. MySQL converts all table names to lowercase on storage and lookup. This behavior also applies to database names and table aliases. |
| 2 | Table and database names are stored on disk using the lettercase specified in the `CREATE TABLE` or `CREATE DATABASE` statement, but MySQL converts them to lowercase on lookup. Name comparisons are not case sensitive. This works *only* on file systems that are not case sensitive! `InnoDB` table names are stored in lowercase, as for `lower_case_table_names=1`. |

If you are using MySQL on only one platform, you do not normally have to change the `lower_case_table_names` variable from its default value. However, you may encounter difficulties if you want to transfer tables between platforms that differ in file system case sensitivity. For example, on Unix, you can have two different tables named `my_table` and `MY_TABLE`, but on Windows these two names are considered identical. To avoid data transfer problems arising from lettercase of database or table names, you have two options:

- Use `lower_case_table_names=1` on all systems. The main disadvantage with this is that when you use `SHOW TABLES` or `SHOW DATABASES`, you do not see the names in their original lettercase.
- Use `lower_case_table_names=0` on Unix and `lower_case_table_names=2` on Windows. This preserves the lettercase of database and table names. The disadvantage of this is that you must ensure that your statements always refer to your database and table names with the correct lettercase on Windows. If you transfer your statements to Unix, where lettercase is significant, they do not work if the lettercase is incorrect.

  ***Exception***: If you are using `InnoDB` tables and you are trying to avoid these data transfer problems, you should set `lower_case_table_names` to 1 on all platforms to force names to be converted to lowercase.

If you plan to set the `lower_case_table_names` system variable to 1 on Unix, you must first convert your old database and table names to lowercase before stopping **mysqld** and restarting it with the new variable setting. To do this for an individual table, use `RENAME TABLE`:

```
RENAME TABLE T1 TO t1;
```

To convert one or more entire databases, dump them before setting `lower_case_table_names`, then drop the databases, and reload them after setting `lower_case_table_names`:

1. Use **mysqldump** to dump each database:

   ```
   mysqldump --databases db1 > db1.sql
   mysqldump --databases db2 > db2.sql
   ...
   ```

   Do this for each database that must be recreated.

2. Use `DROP DATABASE` to drop each database.

3. Stop the server, set `lower_case_table_names`, and restart the server.

4. Reload the dump file for each database. Because `lower_case_table_names` is set, each database and table name will be converted to lowercase as it is recreated:

   ```
   mysql < db1.sql
   mysql < db2.sql
   ...
   ```

Object names may be considered duplicates if their uppercase forms are equal according to a binary

collation. That is true for names of cursors, conditions, procedures, functions, savepoints, stored routine parameters, stored program local variables, and plugins. It is not true for names of columns, constraints, databases, partitions, statements prepared with **PREPARE**, tables, triggers, users, and user-defined variables.

File system case sensitivity can affect searches in string columns of **INFORMATION_SCHEMA** tables. For more information, see Section 10.1.7.9, "Collation and **INFORMATION_SCHEMA** Searches".

Previous / Next / Up / Table of Contents

# User Comments

| Posted by Anders Eriksson on January 20 2009 9:31am | [Delete] [Edit] |
|---|---|

How to produce a SQL script that renames all tables in a schema to its lower case form:

select concat('rename table ', table_name, ' to ' , lower(table_name) , ';') from information_schema.tables where table_schema = 'your_schema_name';

/Anders Eriksson

| Posted by Lone Wolf on December 6 2009 3:32am | [Delete] [Edit] |
|---|---|

Another way, via the command prompt:

cd /var/lib/mysql
for i in */*.frm; do DBASE="`dirname $i`"; TBL="`basename $i .frm`"; TBLI="`echo $TBL | tr A-Z a-z`"; if [[ "$TBL" != "$TBLI" ]] ; then echo "RENAME TABLE \`$DBASE\`.\`$TBL\` TO \`$DBASE\`.\`$TBLI\`" | mysql; fi; done

| Posted by Mircea LUTIC on February 16 2012 1:16pm | [Delete] [Edit] |
|---|---|

##lower_case_table_names :
##Cmd-Line=Yes
##my.ini=Yes
##System =Yes
##Scope =Global
##Dynamic=No

## 0 store=lowercase ; compare=sensitive (works only on case sensitive file systems )
## 1 store=lowercase ; compare=insensitive
## 2 store=exact ; compare=insensitive (works only on case INsensitive file systems )
#default is 0/Linux ; 1/Windows

Add your own comment

Top / Previous / Next / Up / Table of Contents