clickhouse-client ClickHouse SQL shell

CONCEPTS

Interactive mode runs a query on Enter without a semicolon; batch mode (-q, stdin, --queries-file) runs and exits. The client speaks the native TCP protocol, not HTTP.

Connect flags, string, config
↓ query
Query REPL, -q, stdin, file
↓ format
Result Pretty or --format
↓ pipe
Output terminal, pager, file

Defaults differ by mode: PrettyCompact in the REPL, TabSeparated in batch mode. Set --format or a FORMAT clause when scripting.

Missing credentials fall back to CLICKHOUSE_USER, CLICKHOUSE_PASSWORD and CLICKHOUSE_HOST; flags and a connection string win over them.

CONNECT

clickhouse-client
Connect to localhost:9000 as default

clickhouse-client -h *<host>* **--port** *<port>* **-u** *<user>* **--password** *<pw>*
Connect with explicit options

clickhouse-client clickhouse://*<user>***@***<host>***:***<port>***/***<db>*
Connect with a connection string

clickhouse-client -s --port 9440
Force a TLS connection

clickhouse-client -d *<db>*
Start in a database

clickhouse-client --connection *<name>*
Use a saved connection from config

clickhouse-client -V
Print the client version

clickhouse-client --help --verbose
List all options including settings

INSPECT

SHOW DATABASES
List databases

SHOW TABLES
List tables in the current database

DESCRIBE TABLE *<table>*
Show a table's columns

USE *<db>*
Switch database

SHOW CREATE TABLE *<table>*
Show the table DDL

help *<name>*
Docs for a function, engine or setting

SELECT currentDatabase()
Which database am I in?

SELECT version()
Server version

SELECT * FROM system.tables WHERE database = '*<db>***'**
Table metadata

SELECT * FROM system.processes
Running queries

RUN QUERIES

clickhouse-client -q "*<sql>***"**
Run a query and exit

clickhouse-client "*<sql>***"**
Query without --query

echo "*<sql>***" | clickhouse-client**
Query from stdin

clickhouse-client --queries-file *<file.sql>*
Run queries from a file

clickhouse-client -q "*<sql1>***; ***<sql2>***;"**
Several queries at once

clickhouse-client -q "*<sql>***" -q "***<sql2>***"**
Repeat --query

clickhouse-client --ignore-error --queries-file *<file>*
Keep going after a failed query

clickhouse-client -m
Multiline: send only at ;

*****<Enter>
Continue a query on the next line

PARAMETERS

--param_*<name>***=***<value>*
Set a query parameter

{*<name>***:***<Type>***}**
Typed placeholder in SQL

{*<name>***:Identifier}**
Parameterise table or column names

OUTPUT

--format *<format>*
Choose the output format

--format CSVWithNames
CSV with a header row

--format JSONEachRow
One JSON object per line

--format Vertical
One value per line

... FORMAT *<format>*
Set the format inside the query

<query> **\G**
Vertical output for one query

IMPORT & EXPORT

clickhouse-client -q "INSERT INTO *<table>* **FORMAT CSV" <** *<file>*
Load a CSV file

clickhouse-client -q "INSERT INTO *<table>* **FORMAT CSVWithNames" <** *<file>*
Load CSV with a header

clickhouse-client -q "INSERT INTO *<table>* **FORMAT JSONEachRow" <** *<file>*
Load NDJSON

<cmd> **| clickhouse-client -q "INSERT INTO** *<table>* **FORMAT** *<f>***"**
Stream data in from a pipe

clickhouse-client -q "*<select>***" --format CSVWithNames > ***<file>***
Export a query to CSV

INTERACTIVE

Ctrl+D
Quit the client

Ctrl+C
Cancel the running query

Ctrl+R
Fuzzy history search