MySQL® Compatibility

Robert Schulze (Core Team)
Serge Klochkov (Integrations Team)

31 October 2023

ClickHouse

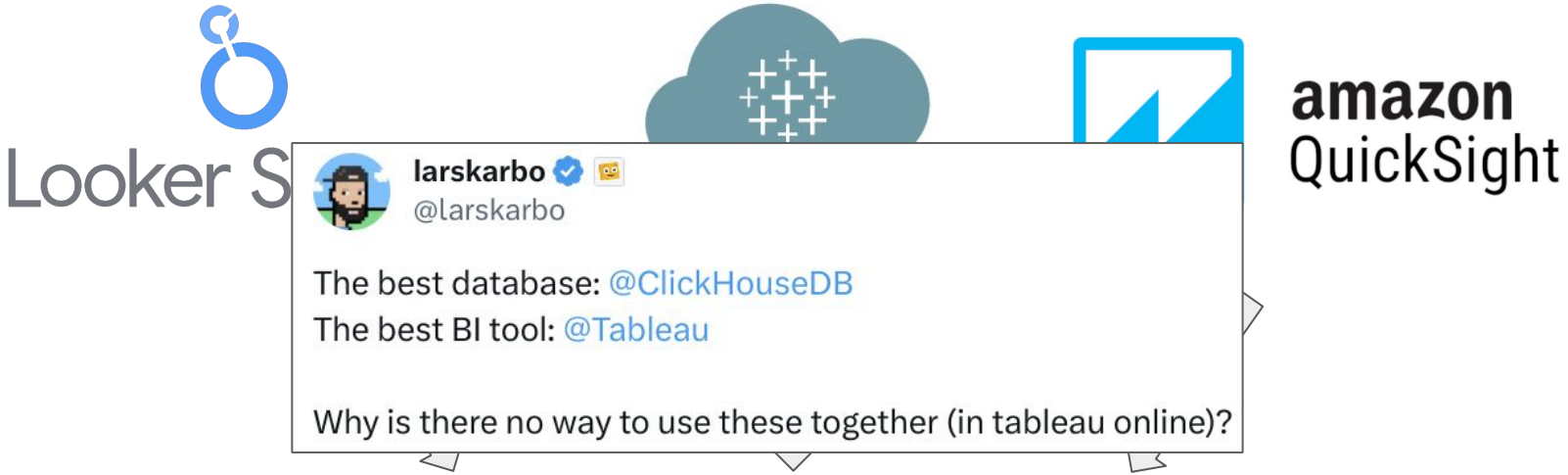
Agenda

01 Motivation and Context

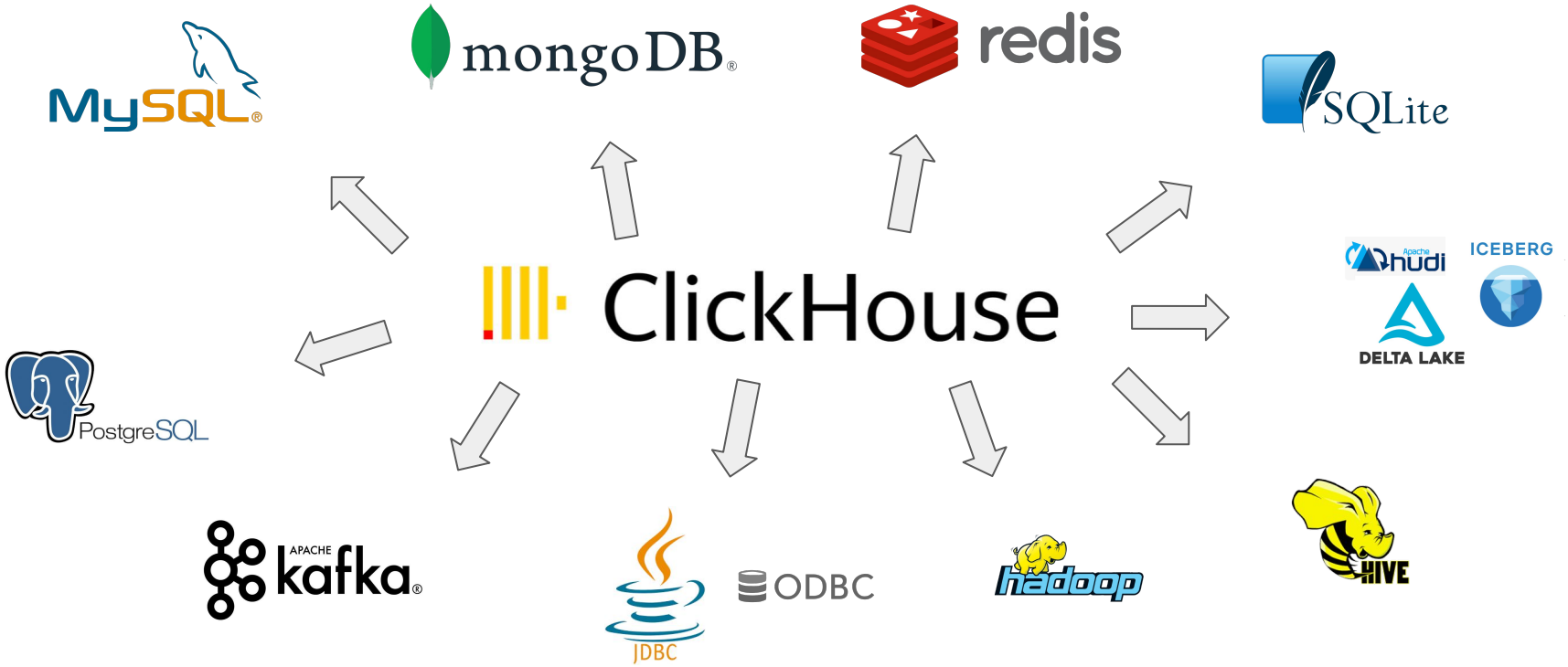
02 MySQL Wire Protocol Compatibility

03 MySQL SQL Dialect Compatibility

Motivation


Looker S
amazon
QuickSight
larskarbo
@larskarbo
The best database: @ClickHouseDB
The best BI tool: @Tableau
Why is there no way to use these together (in tableau online)?
ClickHouse

ClickHouse and Other Databases



Setting up ClickHouse for Connection from MySQL

- Adding a few lines to config.xml:

```
<clickhouse>
  <mysql_port>9004</mysql_port>
  <custom_settings_prefixes>SQL_,custom_</custom_settings_prefixes>
</clickhouse>
```

Don't fail on commands such as `SET SQL_AUTO_IS_NULL = 0`

- And (quite) a few more to users.xml:

```
<clickhouse>
  <profiles>
    <default>
      <use_mysql_types_in_show_columns>1</use_mysql_types_in_show_columns>
      <mysql_map_string_to_text_in_show_columns>1</mysql_remap_string_as_text_in_show_columns>
      <mysql_map_fixed_string_to_text_in_show_columns>1</mysql_remap_fixed_string_as_text_in_show_columns>
    </default>
  </profiles>
  <users>
    <mysql_user>
        <password_double_sha1_hex>...</password_double_sha1_hex>
    </mysql_user>
  </users>
</clickhouse>
```

Enabling ClickHouse -> MySQL types translation in `SHOW [FULL] COLUMNS` queries

By default, String/FixedString types are reported as BLOB MySQL type. QuickSight does not like that.

MySQL user's password has to be encrypted with Double SHA1

Full documentation: https://clickhouse.com/docs/en/integrations/lookerstudio#on-premise-clickhouse-server-setup

What does “MySQL Wire Protocol” mean?

- TCP Connection
- MySQL authentication support
- Incoming command packets handler *
- Result set protocols - text and binary

* https://dev.mysql.com/doc/dev/mysql-server/latest/PAGE_PROTOCOL.html

How it started …

QuickSight

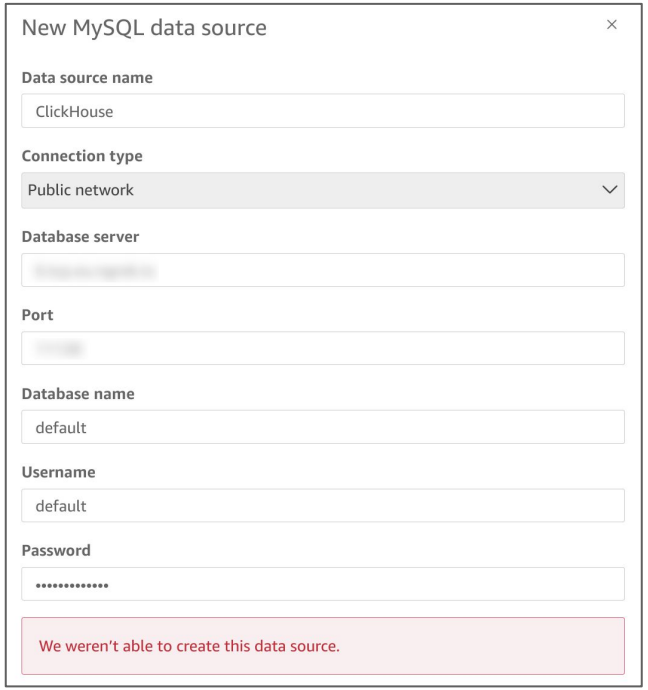
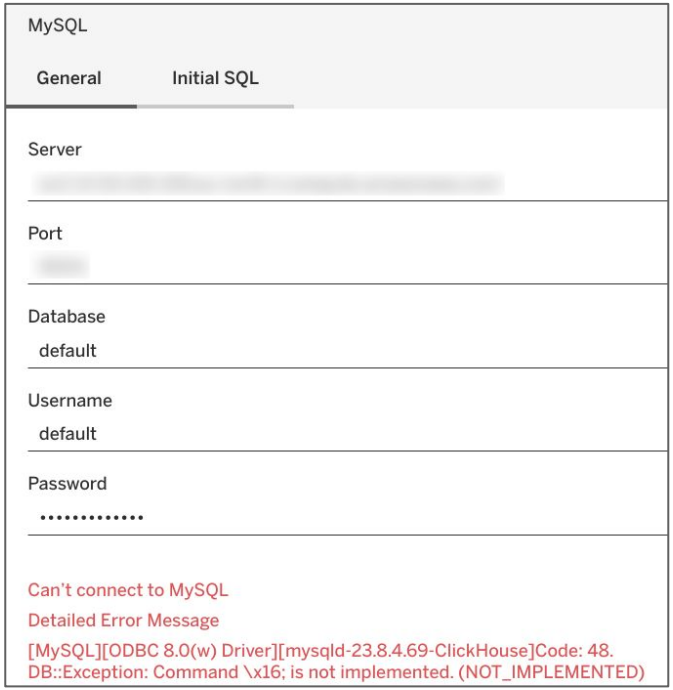


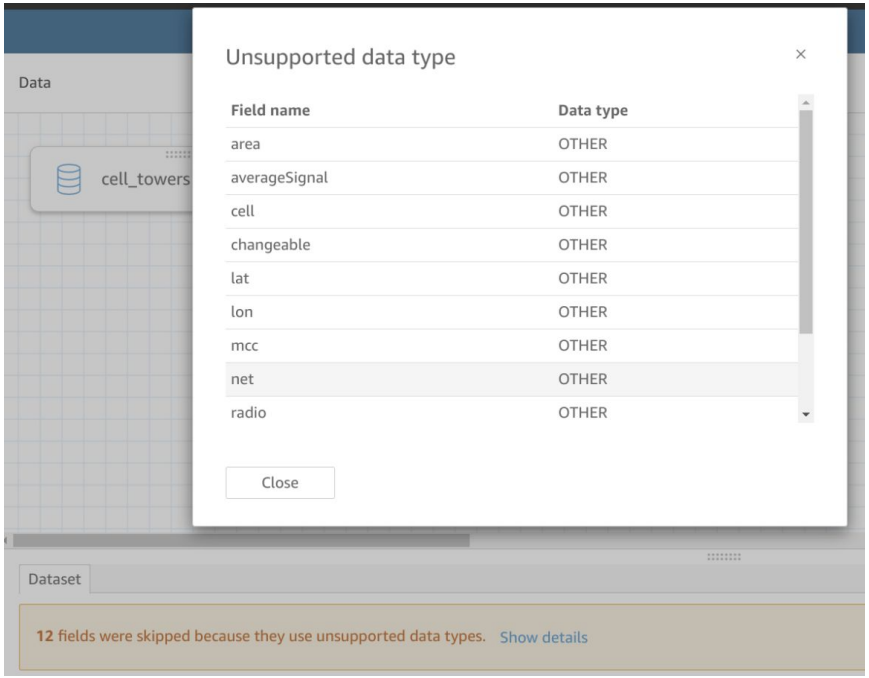
Tableau Online



What is “command \x16”? 🤔

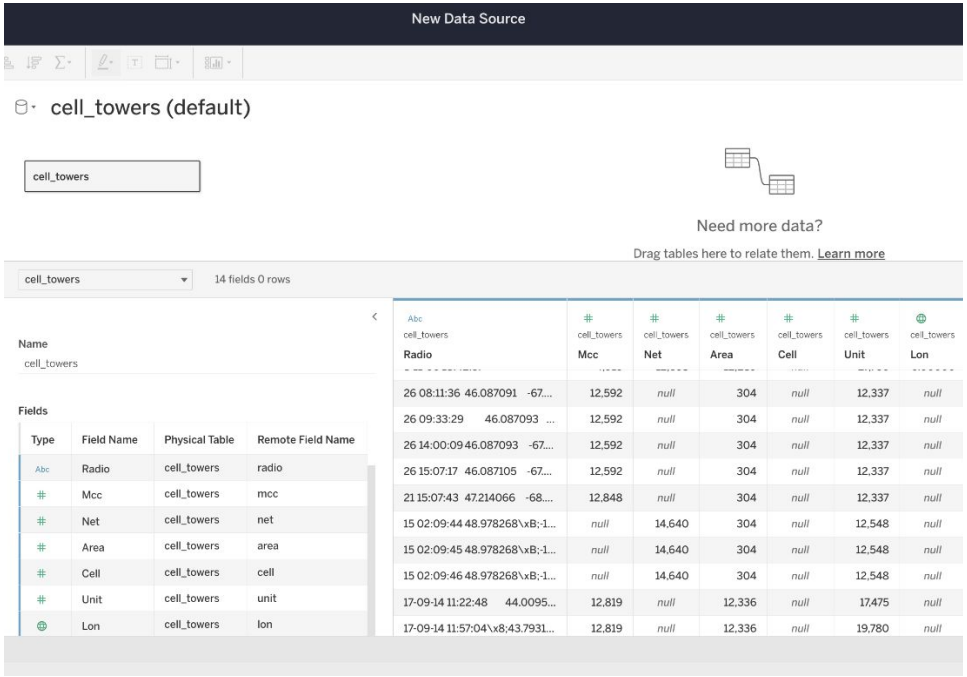
How it started …

QuickSight



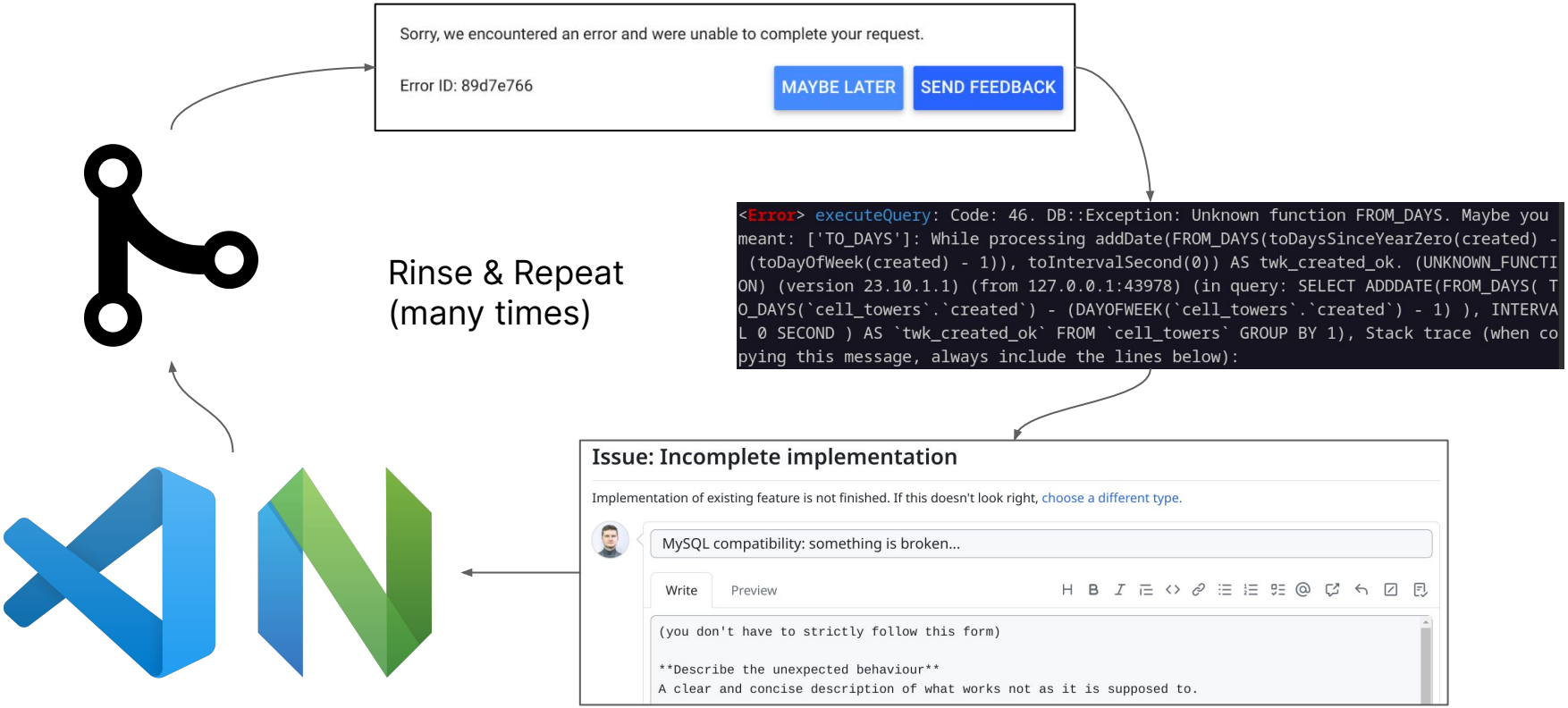
Missing ClickHouse → MySQL types translation

Tableau Online



Too many nulls… And what's up with that "Radio" column?

Testing a Particular BI tool for Compatibility



MySQL Binary Protocol (#51103)

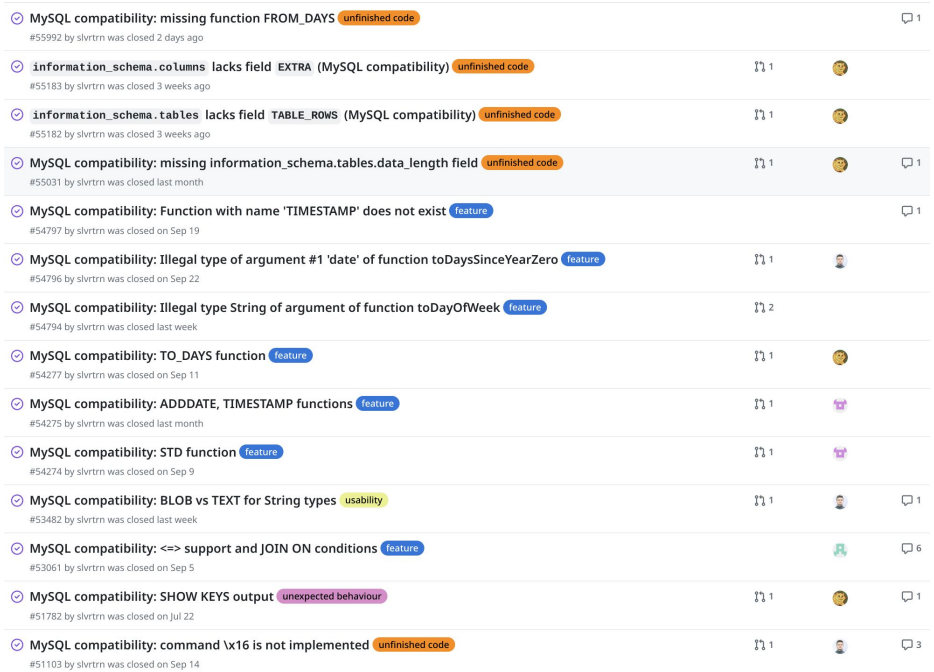
- Remember “command \x16” error in Tableau?
- Missing `COM_STMT_PREPARE`, `COM_STMT_EXECUTE`, `COM_STMT_CLOSE` handlers
- Unlike “standard” queries (`COM_QUERY`) which return just a plain text result set, `COM_STMT_EXECUTE` responds with binary protocol result set.
- Missing the entire binary protocol implementation.
- Added in 23.9, allowing to connect Tableau Online to ClickHouse.

Generated SQL Queries (#53066)

```
SELECT `87605426-78cf-432d-ad42-311766be5478.created_tg`, `count`
FROM (SELECT *
      FROM (SELECT `87605426-78cf-432d-ad42-311766be5478.created_tg`,
                   COUNT(*)    AS `count`,
                   MAX(`$f16`) AS `$PERCENTILECONT_1`,
                   MAX(`$f17`) AS `$PERCENTILECONT_2`,
                   MAX(`$f18`) AS `$PERCENTILECONT_3`
            FROM (SELECT `87605426-78cf-432d-ad42-311766be5478.created_tg`,
                         (MAX(`$f18`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)) +
                         ((`$f16` - 1) * `$f15` -
                          (MAX(`$f17`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`))) *
                         ((MAX(`$f20`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)) -
                          (MAX(`$f18`)
                               OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)))         AS `$f16`,
                         (MAX(`$f23`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)) +
                         ((`$f21` - 1) * `$f15` -
                          (MAX(`$f22`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`))) *
                         ((MAX(`$f25`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)) -
                          (MAX(`$f23`)
                               OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)))         AS `$f17`,
                         (MAX(`$f28`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)) +
                         ((`$f26` - 1) * `$f15` -
                          (MAX(`$f27`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`))) *
                         ((MAX(`$f30`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`)) -
                          (MAX(`$f28`) OVER (PARTITION BY `87605426-78cf-432d-ad42-311766be5478.created_tg`))) AS `$f18`

                         [... hundred more lines like that ...]
```

Tracking Progress …



Fall into categories:

- Functions and function aliases
- SHOW statements
- information_schema

All findings:
https://github.com/ClickHouse/ClickHouse/issues?q=is%3Aissue+author%3Aslvrtrn

Functions and Function Aliases (#54795)

```
SELECT
    ADDDATE(
        DATE_FORMAT(
            cell_towers.updated,
            '%Y-01-01 00:00:00'
        ),
        INTERVAL 0 SECOND)
FROM cell_towers;
```

- **Missing string arguments**: `TO_DAYS()`, `ADDDATE()`, `TODAYOFWEEK()`, ...
- **Missing functions**: `FROM_DAYS()`, `TIMESTAMP()`, `MAKEDATE()`, `REGEXP` operator, ...
- **Missing aliases**: `INSTR()` for `positionCaseInsensitive()`, `STR_TO_DATE()` for `parseDateTimeOrNull()`, `STD()` for `stddevPop()`, ...

Different convenience vs. type-safety philosophies in ClickHouse / MySQL.

- String-formatted dates: '2023-10-31'
- Relaxed mode (e.g. '23:10:31'), invalid dates (e.g. '2023-11-31`)

Carefully consider what ClickHouse can support and what not.

SHOW Statements (#49140)

MySQL MYSQL.COM DOW

MySQL Server MySQL Enterprise Workbench

MySQL 8.2 Reference Manual / ... / SHOW Statements

13.7.7 SHOW Statements

13.7.7.1 SHOW BINARY LOGS Statement
13.7.7.2 SHOW BINARY LOG STATUS Statement
13.7.7.3 SHOW BINLOG EVENTS Statement
13.7.7.4 SHOW CHARACTER SET Statement
13.7.7.5 SHOW COLLATION Statement
13.7.7.6 SHOW COLUMNS Statement
13.7.7.7 SHOW CREATE DATABASE Statement
13.7.7.8 SHOW CREATE EVENT Statement
13.7.7.9 SHOW CREATE FUNCTION Statement
13.7.7.10 SHOW CREATE PROCEDURE Statement
13.7.7.11 SHOW CREATE TABLE Statement
13.7.7.12 SHOW CREATE TRIGGER Statement
13.7.7.13 SHOW CREATE USER Statement
13.7.7.14 SHOW CREATE VIEW Statement
13.7.7.15 SHOW DATABASES Statement
13.7.7.16 SHOW ENGINE Statement
13.7.7.17 SHOW ENGINES Statement
13.7.7.18 SHOW ERRORS Statement
13.7.7.19 SHOW EVENTS Statement
13.7.7.20 SHOW FUNCTION CODE Statement
13.7.7.21 SHOW FUNCTION STATUS Statement
13.7.7.22 SHOW GRANTS Statement
13.7.7.23 SHOW INDEX Statement
13.7.7.24 SHOW MASTER STATUS Statement
13.7.7.25 SHOW OPEN TABLES Statement
13.7.7.26 SHOW PARSE_TREE Statement
13.7.7.27 SHOW PLUGINS Statement
13.7.7.28 SHOW PRIVILEGES Statement
13.7.7.29 SHOW PROCEDURE CODE Statement
13.7.7.30 SHOW PROCEDURE STATUS Statement
13.7.7.31 SHOW PROCESSLIST Statement
13.7.7.32 SHOW PROFILE Statement
13.7.7.33 SHOW PROFILES Statement
13.7.7.34 SHOW RELAYLOG EVENTS Statement
13.7.7.35 SHOW REPLICAS Statement
13.7.7.36 SHOW SLAVE HOSTS Statement
13.7.7.37 SHOW REPLICA STATUS Statement
13.7.7.38 SHOW SLAVE STATUS Statement
13.7.7.39 SHOW STATUS Statement
13.7.7.40 SHOW TABLE STATUS Statement
13.7.7.41 SHOW TABLES Statement
13.7.7.42 SHOW TRIGGERS Statement
13.7.7.43 SHOW VARIABLES Statement
13.7.7.44 SHOW WARNINGS Statement

SHOW has many forms that provide information about dat

13.7.7.23 SHOW INDEX Statement

```
SHOW [EXTENDED] {INDEX | INDEXES | KEYS}
    {FROM | IN} tbl_name
    [{FROM | IN} db_name]
    [WHERE expr]
```

`SHOW INDEX` returns the following fields:

- `Table`

  The name of the table.

- `Non_unique`

  0 if the index cannot contain duplicates, 1 if it can.

- `Key_name`

  The name of the index. If the index is the primary key, the name is always `PRIMARY`.

- `Seq_in_index`

  The column sequence number in the index, starting with 1.

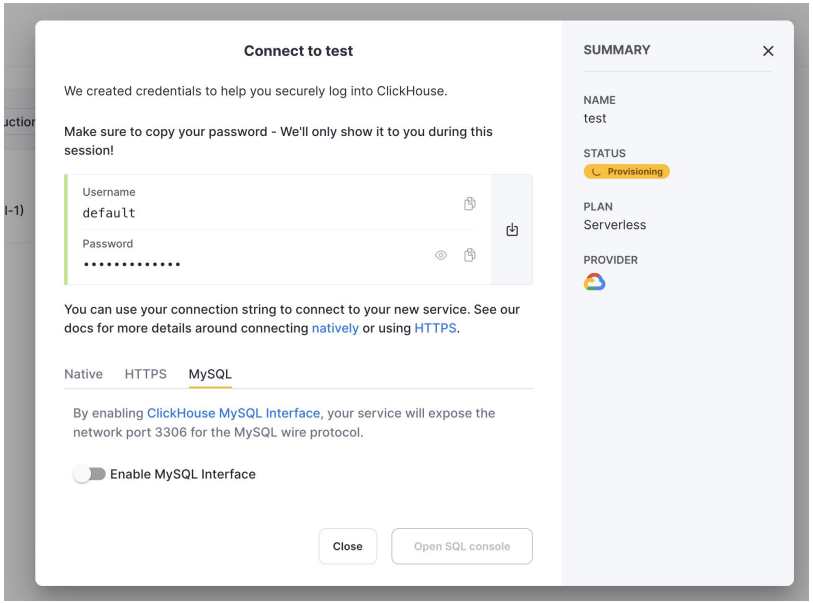
INFORMATION_SCHEMA (#55183)

26.2 INFORMATION_SCHEMA Table Reference

The following table summarizes all available `INFORMATION_SCHEMA` tables. For greater detail, see the individual table descriptions.

Table 26.1 INFORMATION_SCHEMA Tables

Table Name	Description	Deprecated
`ADMINISTRABLE_ROLE_AUTHORIZATIONS`	Grantable users or roles for current user or role	
`APPLICABLE_ROLES`	Applicable roles for current user	
`CHARACTER_SETS`	Available character sets	
`CHECK_CONSTRAINTS`	Table and column CHECK constraints	
`COLLATION_CHARACTER_SET_APPLICABILITY`	Character set applicable to each collation	
`COLLATIONS`	Collations for each character set	
`COLUMN_PRIVILEGES`	Privileges defined on columns	
`COLUMN_STATISTICS`	Histogram statistics for column values	
`COLUMNS`	Columns in each table	
`COLUMNS_EXTENSIONS`	Column attributes for primary and secondary storage engines	
`CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS`	Current number of consecutive failed connection attempts per account	

Conclusions

- v23.10: most fixes included, 90% of functionality in Looker Studio, Tableau Online and QuickSight work
- v23.11: Remaining fixes included
- In ClickHouse Cloud, enable the MySQL interface and be done.
- In parallel, we are working with BI tools vendors to develop official ClickHouse support (long-term effort).