MySQL Auto-Reset Sequence Plugin Installation & Usage Guide

Prerequisites

1. MySQL development headers and libraries
2. GCC compiler
3. MySQL server with plugin support enabled

Install Development Tools (Ubuntu/Debian)

```bash
sudo apt-get update
sudo apt-get install build-essential libmysqlclient-dev mysql-server
```

Install Development Tools (CentOS/RHEL)

```bash
sudo yum install gcc gcc-c++ mysql-devel mysql-server
```

Compilation

1. Save the plugin source code as `sequence_plugin.cpp`
2. Check your MySQL version:

```bash
mysql --version
```

3. Compile the plugin:

For MySQL 5.7 and earlier:

```bash
gcc -shared -fPIC -o sequence_plugin.so sequence_plugin.cpp \
    `mysql_config --cflags` -lmysqlservices -lpthread
```

For MySQL 8.0 and later:

bash

```
gcc -shared -fPIC -o sequence_plugin.so sequence_plugin.cpp \
    `mysql_config --cflags` -lmysqlservices -lpthread -std=c++11
```

Alternative if mysql_config is not available:

bash

```
# Find MySQL include directory
find /usr -name mysql.h 2>/dev/null

# Compile with explicit paths (adjust as needed)
gcc -shared -fPIC -o sequence_plugin.so sequence_plugin.cpp \
    -I/usr/include/mysql -L/usr/lib/mysql \
    -lmysqlservices -lpthread -std=c++11
```

4. Copy the compiled plugin to MySQL plugin directory:

bash

```
# Find plugin directory
mysql -e "SHOW VARIABLES LIKE 'plugin_dir';"

# Copy plugin (adjust path as needed)
sudo cp sequence_plugin.so /usr/lib/mysql/plugin/
```

Installation in MySQL

1. Connect to MySQL as root:

bash

```
mysql -u root -p
```

2. Install the UDF functions:

sql

```
CREATE FUNCTION auto_sequence RETURNS INTEGER SONAME 'sequence_plugin.so';
CREATE FUNCTION row_sequence RETURNS INTEGER SONAME 'sequence_plugin.so';
```

3. Verify installation:

```sql
SELECT * FROM mysql.func WHERE name LIKE 'sequence%';
```

Usage Examples

Basic Usage - Automatic Reset to 1

```sql
-- Each SELECT starts from 1 automatically
SELECT
    auto_sequence() AS row_num,
    id,
    name,
    email
FROM users
ORDER BY name;

-- Running another SELECT will start from 1 again
SELECT
    auto_sequence() AS row_num,
    product_name,
    price
FROM products
WHERE category = 'Electronics';
```

Custom Start Value and Increment

```sql
-- Start from 100, increment by 10
SELECT
    auto_sequence(100, 10) AS row_num,
    department,
    employee_count
FROM departments;

-- Start from 1000, increment by 1
SELECT
    auto_sequence(1000) AS customer_code,
    customer_name,
    registration_date
FROM customers
LIMIT 10;
```

Using ROW_SEQUENCE (Alternative Implementation)

```sql
-- Time-based reset detection
SELECT
    row_sequence() AS row_num,
    order_id,
    order_date,
    total_amount
FROM orders
WHERE status = 'completed';
```

Practical Examples

Example 1: Numbered Report

```sql
-- Sales report with auto-numbering
SELECT
    auto_sequence() AS line_no,
    DATE_FORMAT(order_date, '%Y-%m') AS month,
    COUNT(*) AS order_count,
    SUM(total_amount) AS revenue
FROM orders
GROUP BY DATE_FORMAT(order_date, '%Y-%m')
ORDER BY month DESC;
```

Example 2: Pagination with Row Numbers

```sql
-- First page (automatically starts from 1)
SELECT
    auto_sequence() AS row_num,
    product_id,
    product_name,
    price
FROM products
ORDER BY price DESC
LIMIT 20;

-- Second page (also starts from 1)
SELECT
    auto_sequence() AS row_num,
    product_id,
    product_name,
    price
FROM products
ORDER BY price DESC
LIMIT 20 OFFSET 20;
```

Example 3: Ranking Within Groups

```sql
-- Each query gets its own numbering
SELECT
    category,
    auto_sequence() AS rank_in_category,
    product_name,
    sales_amount
FROM (
    SELECT category, product_name, sales_amount
    FROM products
    ORDER BY category, sales_amount DESC
) AS sorted_products;
```

Example 4: Multiple Sequences in One Query

```sql
-- Different sequences for different purposes
SELECT
    auto_sequence() AS display_row,
    auto_sequence(1000, 1) AS internal_id,
    customer_name,
    registration_date
FROM customers
WHERE status = 'active';
```

How It Works

The plugin provides two functions that automatically reset to 1 for each new SELECT:

1. **AUTO_SEQUENCE()**: Uses query tracking to detect new SELECT statements
2. **ROW_SEQUENCE()**: Uses time-based detection (resets if more than 1 second between calls)

Both functions automatically:

- Start from 1 for each new SELECT statement
- Increment by 1 for each row (customizable)
- Work independently for each query execution

Advanced Usage

Behavior in Subqueries

```sql
-- Each subquery gets its own sequence
SELECT
    main.category,
    main.row_num,
    main.product_name
FROM (
    SELECT
        auto_sequence() AS row_num,
        category,
        product_name
    FROM products
) AS main;
```

Stored Procedures

```sql
DELIMITER //
CREATE PROCEDURE numbered_report()
BEGIN
    -- Each execution starts from 1
    SELECT
        auto_sequence() AS line_no,
        report_data
    FROM report_table;
END//
DELIMITER ;
```

Troubleshooting

Compilation Errors

"my_bool does not name a type"

This error occurs in MySQL 8.0+. The plugin code already handles this with version detection, but ensure you're using the `-std=c++11` flag when compiling.

Missing mysql.h

```bash
# Ubuntu/Debian
sudo apt-get install libmysqlclient-dev

# CentOS/RHEL/Fedora
sudo yum install mysql-devel
# or
sudo dnf install mysql-devel

# macOS with Homebrew
brew install mysql
```

Library linking errors

If you get linking errors, try:

```bash
# Find the MySQL library path
find /usr -name "libmysqlclient*" 2>/dev/null

# Compile with explicit library path
gcc -shared -fPIC -o sequence_plugin.so sequence_plugin.cpp \
    -I/usr/include/mysql -L/usr/lib64/mysql \
    -lmysqlclient -lpthread -std=c++11
```

Plugin Not Loading

```sql
-- Check plugin directory
SHOW VARIABLES LIKE 'plugin_dir';

-- Check if functions are installed
SHOW FUNCTION STATUS WHERE Name LIKE 'sequence%';

-- Check MySQL error log
SHOW VARIABLES LIKE 'log_error';
```

Permission Issues

```sql
-- Grant execute permission to users
GRANT EXECUTE ON FUNCTION sequence_next TO 'username'@'hostname';
GRANT EXECUTE ON FUNCTION sequence_reset TO 'username'@'hostname';
GRANT EXECUTE ON FUNCTION sequence_set TO 'username'@'hostname';
```

Uninstallation

To remove the plugin:

```sql
DROP FUNCTION IF EXISTS sequence_next;
DROP FUNCTION IF EXISTS sequence_reset;
DROP FUNCTION IF EXISTS sequence_set;
```

Then remove the .so file from the plugin directory:

```bash
sudo rm /usr/lib/mysql/plugin/sequence_plugin.so
```

Limitations

1. Sequence is global across all connections
2. Not persistent across MySQL restarts
3. No support for named sequences
4. Maximum value limited by BIGINT (9,223,372,036,854,775,807)

Alternative: Per-Session Sequence

If you need per-session sequences, you can use MySQL variables:

```sql
SET @row_num = 0;
SELECT
    (@row_num:=@row_num + 1) AS row_num,
    *
FROM your_table;
```