Migration of OpenRIMS database from MySQL 5.7 to MySQL 8.

(2024-08-14)

Contents

[Motivation 1](#_Toc174551470)

[Complexity 1](#_Toc174551471)

[The IT environment 1](#_Toc174551472)

[The migration checklist 2](#_Toc174551473)

[Annex 1 Compatibility diagnostic 4](#_Toc174551474)

[Troubleshooting 5](#_Toc174551475)

[ERROR 1231 (42000) at line 3827: Variable 'sql\_mode' can't be set to the value of 'NO\_AUTO\_CREATE\_USER' 5](#_Toc174551476)

# Motivation

Oracle America Inc. doesn’t support MySQL 5.x. The usage of not supported software produces a high risk of data loss, because of the incompatibility with the newest system software. Additionally:

* The installation of the not-supported software may be disabled because of the security and dependency issues
* Third-party software tools like Business Intelligence, database drivers, etc. may stop supporting the outdated software
* The new database improvements related to GIS, JSON, connection pooling, etc., can’t be used

From August 2024 release, the OpenRIMS will support only MySQL 8 database. Thus, migration to MySQL 8 is required. For successful migration, a simple IT environment for migration should be prepared. The software necessary for migration is available on OpenRIMS GitHub.

# Complexity

The basic IT ability is required. The complexity of this process is medium. The estimated completion time is one business day.

# The IT environment

1. A server, on which OpenRIMS and MySQL 5 are running :
   1. It is available using the TCP/IP protocol
   2. A user with DBA administrative rights has the allowance to connect remotely[[1]](#footnote-1)
2. A workstation Windows 10 or Ubuntu Linux on which should be installed the latest
   1. MySQL Workbench Community Edition
   2. MySQL Shell

# The migration checklist

The migration should be provided by an IT expert. A system administrator is the best option.

It is possible the use database dumps created from the 5.7 database for migration to the 8.0.x server. The recommended checklist is below.

| **#** | **Action** | **Result** |
| --- | --- | --- |
|  | Stop OpenRIMS on the server. | It will be possible to upgrade the MySQL 5.7 database structure and OpenRIMS software. |
|  | Create a dump of the server’s database and a copy of the OpenRIMS binary | To allow usage of the previous OpenRIMS release in case of hard mistakes |
|  | Apply openrims57final.mdb model provided to ensure that the latest structure is and fix the most of incompatibilities. | The model contains the latest 5.7. database structure along with compatibility fixes. |
|  | Run compatibility diagnostic from the workstation. See Annex 1 Compatibility diagnostic for details. | To ensure that the database can be transferred to the MySQL 8 server. |
|  | Create a dump of the pdx2 database from the server using MySQL Workbench from the workstation. | The OpenRIMS database is ready to migrate. Please note, that if the server hosts databases for other applications, dumps should be created for these databases. Afterward, these applications may be left on the new database server or moved to another server. |
|  | Stop MySQL 5.7 server | Sometimes it is necessary |
|  | Uninstall the MySQL 5.7 from the server. It would be appreciated if to remove data directories as well. | MySQL 5.7 is no longer in action. |
|  | Install the latest MySQL 8.0.x on the server. (for current it is 8.0.39). While installing following users should be defined:   * root * remote access * OpenRIMS application user from application.properties | MySQL 8.0.x is ready. |
|  | Using MySQL Workbench from the workstation,   * Execute GRANT SYSTEM\_USER ON \*.\* TO *remote access user*; * Create pdx2 schema * restore the dump of the pdx2 database. | The database is migrated. |
|  | Apply the openrims8Start.mdb model provided to ensure the latest structure. | To ensure the latest database MySQL 8 structure. |
|  | Install Oracle Java JDK 21 on the server. On Linux we’ve tested OpenRIMS using OpenJDK | Despite the OpenRIMS is compatible with Java 8, the usage of Java 21 is preferable. |
|  | Install the latest OpenRIMS on the server | The server is ready for testing |
|  | Revise the application.properties parameters[[2]](#footnote-2) in  spring.datasource.url   * useSSL=false * allowPublicKeyRetrieval=true |  |
|  | Run the OpenRIMS and test the installation | The server is migrated |
|  | Check Data Sources in Google Looker reports. | All reports are functional |

# Annex 1 Compatibility diagnostic

The MySQL vendor provides an upgrade checker utility that enables you to verify whether MySQL server instances are ready for upgrade[[3]](#footnote-3). This tool verifies the whole 5.7 server installation.

This tool is running from the workstation. An example of shell scripts:

For Windows

C:\Program Files\MySQL\MySQL Shell 8.4\bin>mysqlsh.exe -- util check-for-server-upgrade swinomatkoff@152.109.10.10 --target-version=8.0

For Linux

mysqlsh -- util check-for-server-upgrade swinomatkoff@152.109.10.10 --target-version=8.0

The result of the upgrade check will be on the screen or can be redirected into a text file.

For Linux, the upgrade check returns in the last lines

**Replace** [**swinomatkoff@152.109.10.10**](mailto:swinomatkoff@152.109.10.10) **with the real user and address**

**Errors: 0**

Unfortunately, for Windows the upgrade check returns in the last lines

**Errors: 1**

We believe that it is a false error. The source of this message is

2) MySQL syntax check for routine-like objects (routineSyntax)

The following objects did not pass a syntax check with the latest MySQL

grammar. A common reason is that they reference names that conflict with new

reserved keywords. You must update these routine definitions and `quote` any

such references before upgrading.

**pdx2.dwh\_update - at line 307,1: unexpected token 'join'**

We carefully checked the syntax and didn’t find anything wrong. This SQL works properly on MySQL 8. Thus, the error is false and the database is ready to upgrade.

# Troubleshooting

## ERROR 1231 (42000) at line 3827: Variable 'sql\_mode' can't be set to the value of 'NO\_AUTO\_CREATE\_USER'

It is a kind of incompatibility MySQL 5.7 and MySQL 8 that is not covered by the import/export procedure. The dump from 5.7 should be fixed.

For MS Windows workstation we provide a script fix57dump.cmd. This script removes NO\_AUTO\_CREATE\_USER from the database dump.

To use this script:

1. Run Command Prompt
2. Set a disk folder that contains the dump as a current
3. Copy to it fix57dump.cmd
4. Run fis57dump.cmd with a dump file name as a parameter, e.g.   
   **fix57dump.cmd dump20240814Anarme.sql**
5. The result will be in a file **fixed\_dump.sql**
6. Import fixed\_dump.sql instead of the original dump

For a Linux workstation, you should create a shell script for ‘sed’ or similar.

1. In general, it is not a good idea to have such kind of user. However, for migration purpose it should be created [↑](#footnote-ref-1)
2. An example is spring.datasource.url = jdbc:mysql://localhost/pdx2?**useSSL=false&allowPublicKeyRetrieval=true**&useUnicode=yes&characterEncoding=UTF-8&characterSetResults=UTF-8&serverTimezone=Europe/Kiev [↑](#footnote-ref-2)
3. https://dev.mysql.com/doc/mysql-shell/8.4/en/mysql-shell-utilities-upgrade.html [↑](#footnote-ref-3)