Database Migration from MySQL to RDM Server

A Raima Inc. Embedded Database Migration Guide

Published: May, 2009
Author: Daigoro F. Toyama
Senior Software Engineer
daigoro.toyama@raima.com

Abstract:

This document is designed to act as a quick guide to the database migration from MySQL to RDM Server by exposing the differences between the two products with regards to the data types, features and SQL statement syntaxes supported differently between them.

This article is relative to the following versions of RDM:

In this document, MySQL refers to MySQL 6.0. Likewise, RDM Server (or RDMs) refers to RDM Server 8.2.

Contents:

Overview

This document gives an overview of the key differences in the SQL implementations between MySQL and RDM Server. It is intended to be a guide for developers to obtain a quick overview of the items to take into consideration when moving an application from MySQL to RDM Server. The first two sections of this document outline the differences in the data types and SQL statements that exist between the two database solutions. However, it does not outline any of the differences in the native APIs of each product. As long as developers use the standard SQL-API, ODBC and/or JDBC interfaces the match between the two products should be nearly identical. If developers are using the MySQL native C-API a translation is required to RDM Servers-supported SAG CLI-API. Information on how to do this translation can be found at: http://en.wikipedia.org/wiki/Call_Level_Interface.

Data Types

The following table shows the data types supported by MySQL in its data definition language and the corresponding data types supported by RDM Server when applicable.

Data Type	*MySQL*	*RDM Server*
8-bit integer	BIT	
16-bit integer	SMALLINT	SMALLINT
24-bit integer	MEDIUMINT	
32-bit integer	INT/INTEGER	INTEGER
64-bit integer	BIGINT	BIGINT
32-bit floating point	FLOAT	
64-bit floating point	DOUBLE / REAL	
Variable floating point	DECIMAL / NUMERIC	DECIMAL / NUMERIC
Date	DATE	DATE
Time	TIME	TIME
Date and Time	DATETIME / TIMESTAMP	TIMESTAMP
Year	YEAR	
Character String	CHAR / VARCHAR	CHAR / VARCHAR
Binary	BINARY / VARBINARY	BINARY / VARBINARY

Data Type	*MySQL*	*RDM Server*
BLOB (binary)	TINYBLOB / BLOB / MEDIUMBLOB / LONGBLOB	LONG VARBINARY
BLOB (character)	TINYTEXT / TEXT / MEDIUMTEXT / LONGTEXT	LONG VARCHAR
Character string (enum)	ENUM	
Character string (set)	SET	

Data Definition Language Statements

Here is the list of the DDL statements supported by MySQL.

ALTER DATABASE / SCHEMA

The ALTER DATABASE statement in MySQL is used to change the overall characteristics of a database. It differs from that of RDM Server where the statement is used to change the contents of a database (such as adding/dropping tables/indices). RDM Server currently does not support this functionality, but the statement is used mainly for upgrading a MySQL 5.1 database to 6.0.

Supported: No
Necessary: No

ALTER EVENT

The ALTER EVENT statement in MySQL is used to change the characteristics of an existing event. Since RDM Server does not support events, it naturally does not support this statement.

Supported: No
Necessary:

ALTER FUNCTION / PROCEDURE

The ALTER FUNCTION / PROCEDURE statement in MySQL is used to change the characteristics of a stored function or stored procedure. RDM Server supports functions and procedures but does not allow the alternation of their characteristics.

Supported: No
Necessary:

ALTER SERVER

The ALTER SERVER statement in MySQL is used to change the information on the specified server. RDM Server does not support the concept of creating/altering servers.

Supported: No
Necessary: No

ALTER TABLE

The ALTER TABLE statement in MySQL is used to change the structure of an existing table. RDM Server supports most of the core options, such as adding/dropping indices, adding/dropping/altering columns, and renaming columns. Those we do not support are mostly for the features that we do not support.

Supported: Yes
Necessary: Yes

CREATE DATABASE

The CREATE DATABASE statement in MySQL creates a new database. MySQL and RDM Server supports different options as they support different database features. For instance, MySQL allows the user to set the default database character set and collation name.

Supported: Yes
Necessary; Yes

CREATE EVENT

The CREATE EVENT statement in MySQL creates and schedules a new event. RDM Server does not support events; therefore not CREATE EVENT.

Supported: No
Necessary: No

CREATE INDEX

The CREATE INDEX statement in MySQL is used to create an index. RDM Server supports it except for the USING [BTREE | HASH | RTREE] clause that specifies the indexing algorithm (we only use BTREE) and the WITH PARSER clause.

Supported: Yes
Necessary: Yes

CREATE FUNCTION / PROCEDURE

The CREATE FUNCTION / PROCEDURE statement in MySQL is used to create a function or a stored procedure. RDM Server supports both, though the details differ. MySQL lets you create both functions and procedures in SQL; RDM Server requires that functions and user-defined procedures be written as shared libraries

Supported: Yes
Necessary: Yes

CREATE SERVER

The CREATE SERVER statement in MySQL creates the definition of a server. RDM Server does not support it as our concept of a server is different than that of MySQL.

Supported: No
Necessary: No

CREATE TABLE

The CREATE TABLE statement in MySQL creates a new table. The IF NOT EXISTS clause is not supported by RDM Server. Detailed additional clauses for the CREATE TABLE statement must be fully investigated in order to figure out exactly how RDM Server supports (or offers alternatives to) them. As for the data types supported, see the **Data Types** section above.

Supported: Yes
Necessary: Yes

CREATE TRIGGER

The CREATE TRIGGER statement in MySQL creates a new trigger. A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. RDM Server supports triggers in the form of UDF, but not in this ANSI format.

Supported: No
Necessary: Probably

CREATE VIEW

The CREATE VIEW statement in MySQL creates a new view. It can also replace an existing view by specifying the OR REPLACE clause, in which case it behaves just like ALTER VIEW. RDM Server supports CREATE VIEW, but not the OR REPLACE feature.

Supported: Yes
Necessary: Yes

DROP DATABASE / SCHEMA

The DROP DATABASE statement in MySQL drops all the tables in the database and deletes the database. RDM Server supports it, except for the IF EXISTS extension.

Supported: Yes
Necessary: Yes

DROP EVENT

The DROP EVENT statement in MySQL drops an event. RDM Server does not support events, hence not the DROP EVENT statement.

Supported: No
Necessary: No

DROP INDEX

The DROP INDEX statement in MySQL drops an index. RDM Server fully supports it.

Supported: Yes
Necessary: Yes

DROP PROCEDURE / FUNCTION

The DROP PROCEDURE / FUNCTION statement in MySQL drops a stored procedure or function (user-defined function or stored function). RDM Server supports it except for the IF EXISTS extension.

DROP SERVER

The DROP SERVER statement in MySQL drops a server definition. RDM Server does not support it as our concept of a server is different than that of MySQL.

DROP TABLE

The DROP TABLE statement in MySQL removes one or more tables. Triggers for the dropped table(s) will also be dropped. RDM Server supports DROP TABLE, though only one table can be removed at a time. RDM Server does not support the IF EXISTS extension, either.

Supported: Yes
Necessary; Yes

DROP TRIGGER

The DROP TRIGGER statement in MySQL drops a trigger. RDM Server supports triggers in the form of UDF, but not in this ANSI format.

Supported: No
Necessary: Probably

DROP VIEW

The DROP VIEW statement in MySQL drops one or more views. RDM Server supports DROP VIEW, though only one view can be removed at a time. RDM Server does not support the IF EXISTS extension, either.

Supported: Yes, basically
Necessary: Yes

RENAME TABLE

The RENAME TABLE statement in MySQL renames one or more tables. RDM Server supports RENAME TABLE with a slightly different syntax. It also only supports renaming one table at a time.

Supported: Yes, basically
Necessary: Yes

Data Manipulation Language Statements

Here is the list of the DML statements supported by MySQL.

CALL

The CALL statement in MySQL invokes a stored procedure. The RDM Server equivalent is EXECUTE.

Supported: Yes
Necessary: Yes

DELETE

The DELETE statement in MySQL deletes rows from the specified table and returns a count of the number of deleted rows. MySQL extends the statement with keywords such as QUICK (where the index leaves are not merged during DELETE) and LIMIT (which limits the number of rows to delete). RDM Server supports the core DELETE statement, but not those MySQL extensions.

Supported: Yes
Necessary: Yes

DO

The DO statement in MySQL executes the expressions but does not return any results. RDM Server does not support it.

Supported: No
Necessary: No

HANDLER

The HANDLER statement in MySQL provides direct access to table storage engine interfaces. It is available for MyISAM and InnoDB tables. Naturally, RDM Server does not support it.

Supported: No
Necessary: No

INSERT

The INSERT statement in MySQL inserts new rows into an existing table. The INSERT … VALUES statement works identically to that of RDM Server. The INSERT … SET statement can be replaced easily with INSERT … VALUES with specific column names. RDM Server does not support MySQL extensions such as INSERT DELAYED and INSERT IGNORE.

Supported: Yes
Necessary; Yes

LOAD DATA INFILE

The LOAD DATA INFILE statement in MySQL reads rows from a text file into a table. The RDM Server equivalent is INSERT … FROM (ASCII) FILE. The LOAD DATA INFILE statement allows for more flexibility than our INSERT … FROM FILE when selecting particular sections of the file to be read and inserted.

Supported: Yes
Necessary: Yes

LOAD XML INFILE

The LOAD XML INFILE statement in MySQL reads data from an XML file into a table. The RDM Server equivalent is INSERT … FROM XML FILE. The same issue with flexibility found in LOAD DATA INFILE exists here.

Supported: Yes
Necessary; Yes

REPLACE

The REPLACE statement in MySQL works exactly like INSERT, except that if an old row has the same value as a new row for a PRIMARY / UNIQUE KEY index, the old row is deleted before the new row is inserted.

It means the REPLACE statement is a combination of INSERT and UPDATE. RDM Server does not support this specific syntax, but the same result can be achieved by combining SELECT, INSERT and UPDATE. A UDP would be an option, too.

Supported: No, but it can be worked around
Necessary: No

SELECT

The SELECT statement in MySQL is used to retrieve rows from one or more tables. MySQL allows UNION, JOIN and sub queries. RDM Server supports it except for a few fine-tuning parameters and UNION. The UNION syntax will be supported in the next version of RDM Server. MySQL also allows index hints to be specified for manual query optimization. RDM Server supports the same concept through a different syntax.

Supported: Yes, except for UNION
Necessary: Yes

TRUNCATE

The TRUNCATE [TABLE] statement in MySQL empties a table completely. RDM Server does not support this, although DELETE FROM TABLE syntactically serves the same purpose. The TRUNCATE statement in MySQL exists primarily for performance purposes.

Supported: No
Necessary: No

UPDATE

The UPDATE statement in MySQL updates rows in one or more tables with the specified new values. RDM Server supports it except for the IGNORE and LIMIT extensions.

Supported: Yes, basically
Necessary: Yes

Transactional and Locking Statements

Here is the list of the transactional and locking statements supported by MySQL.

START TRANSACTION / BEGIN

The START TRANSACTION or BEGIN statement in MySQL begins a new transaction. RDM Server supports the operation with the BEGIN [WORK] statement.

COMMIT / ROLLBACK

The COMMIT / ROLLBACK statement in MySQL commits / rolls back the current transaction. RDM Server supports the operations with the COMMIT / ROLLBACK statements.

SET AUTOCOMMIT

The SET AUTOCOMMIT statement in MySQL turns on or off the automatic commit mode. The automatic commit mode is ON by default in MySQL. You run the statement in the following way to disable auto commit.

```
SET autocommit = 0;
```

RDM Server does not support the SET AUTOCOMMIT statement directly, though it should not be hard to add support for it. RDM Server also supports the INSERT statement with the WITH AUTO COMMIT option as follows.

```
INSERT WITH AUTO COMMIT INTO table_name VALUES( … );
```

SAVEPOINT

The SAVEPOINT statement in MySQL, when used with InnoDB or Falcon, sets a named transaction save point with a name. If the current transaction has a save point with the same name, the old save point is deleted and a new one is set. RDM Server supports this feature with the MARK statement. The behaviour is identical.

ROLLBACK TO SAVEPOINT

The ROLLBACK TO SAVEPOINT statement in MySQL rolls back a transaction to the named save point without terminating the transaction. The locks applied after the save point will be freed under Falcon but not under InnoDB. RDM Server supports this feature with the same syntax. Its behaviour regarding the locks is identical to that of InnoDB.

LOCK / UNLOCK TABLES

The LOCK / UNLOCK TABLES statements in MySQL acquires / releases non-transactional and transactional table locks. The LOCK / UNLOCK TABLE statements can be used on views as well as tables. RDM Server supports this feature with the LOCK / UNLOCK TABLE statements, though it does not support locking / unlocking views.

SET TRANSACTION ISOLATION

The SET TRANSACTION ISOLATION statement in MySQL sets the transaction isolation level globally, for the current session or for the next transaction. All modes – READ COMMITTED, READ UNCOMMITTED, REPEATABLE READ and SERIALIZE are supported. RDM Server supports this feature as well, sans SERIALIZE. RDM Server also only supports the transaction isolation setting on the current session.

Conclusion

With RDM Server's support for SQL being nearly the same as that of MySQL's and RDM Server's ability to run as a embedded library like MySQL, moving an embedded application over to RDM Server is relatively easy. So why not take advantage of the additional benefits in RDM Server's performance, full transactional support, and competitive licensing options.

Contact Information

Website: http://www.raima.com

WORLDWIDE

Raima Inc.
720 Third Avenue, Suite 1100
Seattle, WA 98104
Telephone: +1 206 748 5300
Fax: +1 206 748 5200
E-mail: sales@raima.com

EUROPE

Raima Inc.
Stubbings House, Henley Road
Maidenhead SL6 6QL United Kingdom
Telephone: +44 1628 826 800
Fax: +44 1628 825 343
E-mail: sales@raima.com