

Database - Feature #6348

implement support for MariaDB

05/10/2022 09:53 PM - Greg Shah

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Ovidiu Maxiniuc	% Done:	80%
Category:		Estimated time:	0.00 hour
Target version:		version:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to Database - Feature #6441: compatible collation for UTF-8 databases			New
Related to Database - Feature #6507: cross-database joins			New
Related to Database - Feature #6418: represent extent fields as arrays			WIP
Related to Database - Feature #6628: port PostgreSQL native user defined func...			WIP

History

#1 - 05/23/2022 11:44 AM - Ovidiu Maxiniuc

I am interested in taking this as a background task. I worked previously with MySQL so the syntax should be familiar.

#2 - 05/26/2022 09:05 PM - Ovidiu Maxiniuc

Subqueries / Subselects	Full support	https://mariadb.com/kb/en/subqueries/
JOINS (inner and outer)	Full support	https://mariadb.com/kb/en/join-syntax/
LOBs	Full support	https://mariadb.com/docs/reference/mdb/data-types/CLOB/ https://mariadb.com/kb/en/blob-and-text-data-types/
Custom collations	Supported. Scripts could be created similar to H2/PSQL	https://dev.mysql.com/doc/refman/5.7/en/ldml-collation-example.html
Sort nulls high (or low)	Not directly, the sorts need a bit of 'decorations'	https://mariadb.com/kb/en/null-values/ https://www.designcise.com/web/tutorial/how-to-order-null-values-first-or-last-in-mysql
Functional Index/Key Parts Functional Indexes Generated (Virtual and Persistent/Stored) Columns	Not sure if it is as powerful as psql but virtual columns (like mssql) are possible	https://dev.mysql.com/doc/refman/8.0/en/create-index.html#create-index-functional-key-parts https://dasini.net/blog/2019/03/14/mysql-functional-indexes/ https://mariadb.com/kb/en/generated-columns/
Sequences	Full support	https://mariadb.com/kb/en/sequence-overview/ https://mariadb.com/kb/en/sequence-functions/
PL/SQL UDF functions	It has a PL language somewhat similar to PSQL. Alternatively, there is support native code (C/C++) from a plugin. Java is probably not supported	https://mariadb.com/kb/en/create-function/

#3 - 06/08/2022 09:42 PM - Greg Shah

UTF8MB4 support will be needed.

#4 - 06/15/2022 09:37 AM - Eric Faulhaber

- Related to Feature #6441: compatible collation for UTF-8 databases added

#5 - 07/04/2022 04:37 PM - Eric Faulhaber

- Related to Feature #6507: cross-database joins added

#6 - 07/05/2022 04:37 PM - Eric Faulhaber

- Assignee set to Ovidiu Maxiniuc

#7 - 07/07/2022 11:24 AM - Eric Faulhaber

Proposed data type mappings:

4GL	MariaDB	FWD
blob	blob	blob
character	varchar(#) ¹	character
clob	mediumtext	clob
date	date	date
datetime	datetime	datetime
datetime-tz	timestamp	datetimetz
decimal	decimal	decimal
integer	integer	integer
int64	bigint	int64
logical	boolean	logical
recid	bigint	recid
rowid	bigint ²	rowid
Extents	JSON ³	N/A, any

Notes:

1. This mapping was suggested by a customer, but note that it is not strictly compatible, as varchar will strip trailing spaces (see <https://www.oreilly.com/library/view/mysql-cookbook/0596001452/ch04s03.html>). It also is case-insensitive by default (see <https://www.oreilly.com/library/view/mysql-cookbook/0596001452/ch04s10.html>). Although it is very desirable to eliminate the requirement to wrap text columns in upper(trim()) in queries and indices, we need to take these differences into account. The case-insensitivity matches the common case in the 4GL, where it is uncommon to see a field declared case-sensitive, so this aspect seems to be good default behavior. The trailing whitespace trimming, while probably the common, desired behavior, represents a departure from 4GL compatibility, which stores the whitespace, but ignores it on comparisons. We should provide a strictly compatibility type as either an option or as the default, with overrides available via schema hint.
2. There is a customer request to use UUID instead of a 64-bit integer, but this has not been decided yet, and is not part of this initial implementation, as it has impacts beyond support for any particular database dialect.
3. Extent fields require special handling (see #6418), though this is not in scope for the first phase of MariaDB support. We will try to implement this as soon as possible, but the top priority is to get the basics working to stand up MariaDB as a back end for a converted application.

#8 - 07/07/2022 11:32 AM - Eric Faulhaber

The initial target character set / collation is UTF8MB4. We won't worry about getting an exact, backward compatible collation (i.e., [#6441](#)) working immediately; this can be deferred a bit.

I understand the configuration to use UTF8MB4 will look like this:

```
[mysqld]

collation-server = utf8mb4_unicode_ci
init-connect='SET NAMES utf8mb4'
character-set-server = utf8mb4

[mysql]

default-character-set=utf8mb4

[client]

default-character-set=utf8mb4
```

#9 - 07/07/2022 02:29 PM - Eric Faulhaber

See [Adding Support For a New Database](#), but note it is unfinished at the time of this writing.

The priority is to get a working, if minimal, implementation of FWD with MariaDB as a back-end. The basics include:

Server Setup

- Plan the backend architecture, allowing for multiple databases in the same server (so as to not preclude the cross-database join requirement in [#6507](#)).
- Create a MariaDbDialect subclass of Dialect and implementing the minimal methods needed.
- Data type mapping (see [#6348-7](#) as a starting point).
- DDL generation.
- Index definitions (will need to vary depending on use of varchar(#) vs. another text data type).
- Ensuring import works.
- Port native UDFs from PostgreSQL.
 - This should be worked in parallel with other tasks; it is somewhat independent and not strictly necessary for basic support to work, but it is still high priority, because an application of any complexity will use these.
 - In PostgreSQL, we have a separate udf schema for the UDFs. What does this need to look like in MariaDB?
 - If needed, update ScriptRunner to install the UDFs (see also [#6574](#)).

FWD Runtime

- Integrating any dialect-specifics into:
 - FQL preprocessing (e.g., don't inject upper(rtrim())) if using varchar(#); inject BINARY if case-sensitive string comparison with varchar(#) is needed (or is this better injected in the FQL -> SQL conversion?).
 - FQL -> SQL conversion;
 - Legacy sequence support.
 - any other, dialect-specific runtime areas.
- Connection pool integration (are any changes needed here? it's just JDBC).

Lower Priority Functional

- Non-UTF8MB4 character sets
- Matching legacy collation

Documentation

- Database server setup needs to be updated for this dialect.

Ovidiu, when you have time, please post any additional items I may have missed, and we can prioritize accordingly. I think initially you need to take on the items that are very tightly coupled with FWD. Perhaps Boris initially can make the biggest impact with items which are less intertwined (e.g., the UDF port?).

#10 - 07/12/2022 11:05 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

It has a PL language somewhat similar to PSQL.

I followed the CREATE FUNCTION doc link you posted, and found this statement in the context of the CREATE FUNCTION syntax documentation:

"LANGUAGE SQL is a standard SQL clause, and it can be used in MariaDB for portability. However that clause has no meaning, because SQL is the only supported language for stored functions."

If we are truly limited to SQL, this is concerning, as many of the UDFs we have are too complex to be expressed as simple SQL.

There is also a way to use a kind of Oracle compatibility mode (https://mariadb.com/kb/en/sql_modeoracle/), which apparently offers a subset of Oracle's PL/SQL syntax.

In order to port our PostgreSQL UDFs to Maria, we need a better understanding of the facilities we have in this area, and the limitations we will face, w.r.t. the types of things we are doing in our more complex (i.e., non-SQL) UDFs today.

#11 - 07/16/2022 05:49 PM - Ovidiu Maxiniuc

- % Done changed from 0 to 10
- Status changed from New to WIP

Started adding support for the new database by implementation of the MariaDbDialect.java. On Monday I will continue to DDL generation.

#12 - 07/19/2022 01:57 PM - Eric Faulhaber

- Related to Feature #6418: represent extent fields as arrays added

#13 - 07/19/2022 08:20 PM - Ovidiu Maxiniuc

Some observations on data types resulted from the attempt to create a (partial) database from generated DDL for MariaDb:

- `varchar(M)`. The size M is mandatory. Also, it is limited to 16383 (see error \$1074 - Column length too big for column '<column-name>' (max = 16383); use BLOB or TEXT instead). Since I am unable to predict the data size of the imported database I created all columns with maximum size;
- There a limitation on the size of the record (Error \$1118 - Row size too large. The maximum row size for the used table type, not counting BLOBs, is 65535. This includes storage overhead, check the manual. You have to change some columns to TEXT or BLOBs). My test table has more than 4 character columns which will make record overflow;
I am thinking of using a really small value for M or using TEXT as an alternative.
- There are some unmapped data types: `raw` and `rowid`. I propose `bigint` for `rowid` (and maybe downgrade `recid` to integer as it is only 4 byte long?), and `varbinary` for `raw`;
- Using the `utf8mb4` as CHARACTER SET for the database would cover all Unicode characters but have some downsides. Each character uses 4 bytes to store so the MariaDb limits will be rapidly reached. There is a limit on the index size (Error \$1071 - Specified key was too long; max key length is 3072 bytes). If we convert to characters we get maximum of 768 of them and, if an index has 3 string components, they are limited to 256 characters. It matters the definition size, not actual data stored.

#14 - 07/20/2022 07:53 PM - Ovidiu Maxiniuc

A first revision of FWD with partial support for MariaDb dialect was committed in 14095:

- the build script will fetch the required JDBC driver;
- do not forget to add `mariadb` in the list of your `namespace/ddl-dialects` list in `p2j.cfg.xml` to enable generation of DDLs;
- the mapping datatypes from note-7 with the additions from note-13;
- the DDL for testcase project should be fully compatible with MariaDb server. Both the table and index definitions were installed, but with the constraint of `MAX_VARCHAR_SIZE = 100` because of the issue from note-13.

#15 - 07/22/2022 04:00 AM - Constantin Asofiei

With 3821c/14095 change of `build.gradle`, this was added:

```
fwdServer group: 'org.mariadb.jdbc', name: 'mariadb-java-client', version: '2.7.3'
```

Greg: I know Roger was keeping the documentation/license project up-to-date, but is there a wiki with the steps needed to do when a new jar is added or a jar version is changed?

#16 - 07/22/2022 07:50 AM - Greg Shah

Greg: I know Roger was keeping the documentation/license project up-to-date, but is there a wiki with the steps needed to do when a new jar is added or a jar version is changed?

Roger has been helping out but indeed we have planned for everybody to maintain it themselves in combination with their dependency changes/additions/deletions.

See [Maintaining the List of Software Dependencies and Licenses](#). If anything is unclear or incomplete, let's discuss and update the documentation to make it better.

#17 - 07/22/2022 11:50 AM - Eric Faulhaber

- Related to Feature #6628: port PostgreSQL native user defined functions to MariaDB added

#18 - 07/22/2022 12:07 PM - Ovidiu Maxiniuc

Greg Shah wrote:

Greg: I know Roger was keeping the documentation/license project up-to-date, but is there a wiki with the steps needed to do when a new jar is added or a jar version is changed?

Roger has been helping out but indeed we have planned for everybody to maintain it themselves in combination with their dependency changes/additions/deletions.

See [Maintaining the List of Software Dependencies and Licenses](#). If anything is unclear or incomplete, let's discuss and update the documentation to make it better.

The JDBC driver is only required at runtime. I have not reached that point yet. I never started an import on this dialect nor the server. I just added it to save my research work in this regard. I do not know if it is actual the right thing (there are a few versions, including a 3.0 in Beta stage). I will add the documentation as soon as I see it in action and test it a little bit.

#19 - 08/03/2022 02:56 PM - Igor Skornyakov

Please note that CONTAINS (word tables') support for MariaDB will also need some some efforts (we need MariaDB-specific words functions and table triggers). Hopefully the SQL re-writing will require minimal re-work. However supporting word indexes for non-normalized extent fields can be not straightforward.

#20 - 08/03/2022 03:52 PM - Ovidiu Maxiniuc

Yes, I noticed that WordTable class and some methods of DDLGeneratorWorker are not fully dialect independent. These have to be refactored so that the SQL constructs will be generated by the dialect received as parameter.

Luckily, the first customer which intends to use MariaDb as backing database is not using word indexes so this effort can be delayed a bit. At least this is what I understood. Eric, please confirm that this is true.

#21 - 08/03/2022 10:59 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

Yes, I noticed that WordTable class and some methods of DDLGeneratorWorker are not fully dialect independent. These have to be refactored so that the SQL constructs will be generated by the dialect received as parameter.

Agreed, we should not have logic that begins if (dialect instanceof *Dialect)

Luckily, the first customer which intends to use MariaDb as backing database is not using word indexes so this effort can be delayed a bit. At least this is what I understood. Eric, please confirm that this is true.

Yes, this is true. We eventually will need to support word indices, but that is a secondary priority right now.

#22 - 08/04/2022 12:38 AM - Igor Skornyakov

Eric Faulhaber wrote:

Agreed, we should not have logic that begins if (dialect instanceof *Dialect)

I'm thinking about refactoring the word tables' code generator logic by introducing a corresponding abstract class. The Dialect should contain a factory method to create a dialect-specific instance of this class. A similar refactoring can improve the ScriptRunning code.

#23 - 08/04/2022 05:50 AM - Greg Shah

However supporting word indexes for non-normalized extent fields can be not straightforward.

Is it possible to use word indexes on an extent field? I thought extent fields could not be indexed in the 4GL.

#24 - 08/04/2022 06:18 AM - Igor Skornyakov

Greg Shah wrote:

However supporting word indexes for non-normalized extent fields can be not straightforward.

Is it possible to use word indexes on an extent field? I thought extent fields could not be indexed in the 4GL.

Yes, it is possible. One of the words functions in the words-udfs-sql.sql is for such fields.

#25 - 08/12/2022 01:33 PM - Ovidiu Maxiniuc

I did a basic write/read from database test. It seems to be working, but I did not use any fancy queries.

I used the following table (note that it has fields of all all data types, and a few extents, too):

```
ADD TABLE "tm-all"
  AREA "Schema Area"
  DUMP-NAME "tm-all"

ADD FIELD "f-1-char" OF "tm-all" AS character
  FORMAT "X(8)"
  INITIAL "1111111111"
  POSITION 2
  MAX-WIDTH 16
  ORDER 10

ADD FIELD "f-1-char-sens" OF "tm-all" AS character
  CASE-SENSITIVE
  FORMAT "X(8)"
  INITIAL "1111111111"
  POSITION 2
  MAX-WIDTH 16
  ORDER 12

ADD FIELD "f-2-date" OF "tm-all" AS date
  FORMAT "99/99/9999"
  INITIAL "2/22/22"
  POSITION 3
  MAX-WIDTH 4
  ORDER 20

ADD FIELD "f-3-decimal" OF "tm-all" AS decimal
  FORMAT "$*,**9.99+"
  INITIAL "333.333"
  POSITION 4
  MAX-WIDTH 17
```



```

DECIMALS 2
ORDER 30

ADD FIELD "f-4-integer" OF "tm-all" AS integer
  FORMAT "+>99"
  INITIAL "4"
  POSITION 5
  MAX-WIDTH 4
  ORDER 40

ADD FIELD "f-5-int64" OF "tm-all" AS int64
  FORMAT "+>9999"
  INITIAL "55555"
  POSITION 6
  MAX-WIDTH 8
  ORDER 50

ADD FIELD "f-6-logical" OF "tm-all" AS logical
  FORMAT "male/female"
  INITIAL "no"
  POSITION 7
  MAX-WIDTH 1
  ORDER 60

ADD FIELD "f-7-datetime" OF "tm-all" AS datetime
  FORMAT "99-99-9999 HH:MM:SS.SSS AM"
  INITIAL "7/7/77 07:07:07.777"
  POSITION 8
  MAX-WIDTH 8
  ORDER 70

ADD FIELD "f-8-datetimez" OF "tm-all" AS datetime-tz
  FORMAT "99/99/9999 HH:MM:SS.SSS AM +HH:MM"
  INITIAL "8/8/88 8:8:8.888+8:0"
  POSITION 9
  MAX-WIDTH 12
  ORDER 80

ADD FIELD "f-9-blob" OF "tm-all" AS blob
  FORMAT "x(8)"
  INITIAL ?
  POSITION 10
  LOB-AREA "Schema Area"
  LOB-BYTES 104857600
  LOB-SIZE 100M
  ORDER 90

ADD FIELD "f-10-clob" OF "tm-all" AS clob
  FORMAT "x(8)"
  INITIAL ?
  POSITION 11
  LOB-AREA "Schema Area"
  LOB-BYTES 104857600
  LOB-SIZE 100M
  CLOB-CODEPAGE "iso8859-1"
  CLOB-COLLATION "basic"
  CLOB-TYPE 1
  ORDER 100

ADD FIELD "f-11-raw" OF "tm-all" AS raw
  FORMAT "x(8)"
  INITIAL ""
  POSITION 12
  MAX-WIDTH 16
  ORDER 110

ADD FIELD "f-12-recid" OF "tm-all" AS recid
  FORMAT ">>>>>>9"
  INITIAL "121212"
  POSITION 13
  MAX-WIDTH 8
  ORDER 120

ADD FIELD "e-1-char" OF "tm-all" AS character
  FORMAT "X(8)"

```

```

INITIAL "111111111"
POSITION 2
EXTENT 5
MAX-WIDTH 16
ORDER 15

ADD FIELD "e-2-date" OF "tm-all" AS date
FORMAT "99/99/9999"
INITIAL "2/22/22"
EXTENT 5
POSITION 3
MAX-WIDTH 4
ORDER 25

ADD FIELD "e-3-decimal" OF "tm-all" AS decimal
FORMAT "$*,**9.99+"
INITIAL "333.333"
EXTENT 3
POSITION 4
MAX-WIDTH 17
DECIMALS 2
ORDER 35

ADD FIELD "e-4-integer" OF "tm-all" AS integer
FORMAT "+>99"
INITIAL "4"
EXTENT 3
POSITION 5
MAX-WIDTH 4
ORDER 45

ADD INDEX "multi-index" ON "tm-all"
AREA "Schema Area"
UNIQUE
PRIMARY
INDEX-FIELD "f-1-char" ASCENDING
INDEX-FIELD "f-2-date" DESCENDING
INDEX-FIELD "f-3-decimal" ASCENDING
INDEX-FIELD "f-4-integer" DESCENDING
INDEX-FIELD "f-5-int64" ASCENDING
INDEX-FIELD "f-6-logical" DESCENDING
INDEX-FIELD "f-7-datetime" ASCENDING
INDEX-FIELD "f-8-datetimeztz" DESCENDING
INDEX-FIELD "f-12-recid" DESCENDING

```

I also wrote a short procedure in a hello-world style to create a record (two, in fact) and retrieve it back from database:

```

CREATE tm-all.

f-1-char = "AbcDef". // character
f-1-char-sens = "AbcDef". // character CASE-SENSITIVE
f-2-date = TODAY. // date
f-3-decimal = 3.14159266. // decimal
f-4-integer = 100. // integer
f-5-int64 = 123456789012. // int64
f-6-logical = YES. // logical
f-7-datetime = NOW. // datetime
f-8-datetimeztz = NOW. // datetime-tz
f-9-blob = ?. // blob
f-10-clob = ?. // clob
f-11-raw = ?. // raw
f-12-recid = RECID(tm-all). // recid
e-1-char[1] = "xYz 1". // character EXTENT 5
e-1-char[2] = ?. // character EXTENT 5
e-1-char[3] = ". // character EXTENT 5
e-1-char[4] = "xYz 4". // character EXTENT 5
e-1-char[5] = "xYz 5". // character EXTENT 5
e-2-date[1] = TODAY. // date EXTENT 5
e-2-date[2] = ?. // date EXTENT 5
e-2-date[3] = 10/10/2020. // date EXTENT 5
e-2-date[4] = 08/30/2022. // date EXTENT 5

```

```

e-2-date[5] = 12/12/12.          // date    EXTENT 5
e-3-decimal[1] = 0.              // decimal  EXTENT 3
e-3-decimal[2] = ?.              // decimal  EXTENT 3
e-3-decimal[3] = -1000.00001.    // decimal  EXTENT 3
e-4-integer[1] = 0.              // integer  EXTENT 3
e-4-integer[2] = ?.              // integer  EXTENT 3
e-4-integer[3] = -999.           // integer  EXTENT 3

```

```

CREATE tm-all.
f-1-char = "nu-ll".

```

```

FIND FIRST tm-all WHERE f-1-char NE "nu-ll" .

```

```

MESSAGE "f-1-char:"          f-1-char.
MESSAGE "f-1-char-sens:"     f-1-char-sens.
MESSAGE "f-2-date:"          f-2-date.
MESSAGE "f-3-decimal:"       f-3-decimal.
MESSAGE "f-4-integer:"       f-4-integer.
MESSAGE "f-5-int64:"         f-5-int64.
MESSAGE "f-6-logical:"       f-6-logical.
MESSAGE "f-7-datetime:"      f-7-datetime.
MESSAGE "f-8-datetimeetz:"   f-8-datetimeetz.
MESSAGE "f-9-blob:"          f-9-blob.
MESSAGE "f-10-clob:"         f-10-clob.
MESSAGE "f-11-raw:"          f-11-raw.
MESSAGE "f-12-recid: "       f-12-recid.
MESSAGE "e-1-char[1]:"       e-1-char[1].
MESSAGE "e-1-char[2]:"       e-1-char[2].
MESSAGE "e-1-char[3]:"       e-1-char[3].
MESSAGE "e-1-char[4]:"       e-1-char[4].
MESSAGE "e-1-char[5]:"       e-1-char[5].
MESSAGE "e-2-date[1]:"       e-2-date[1].
MESSAGE "e-2-date[2]:"       e-2-date[2].
MESSAGE "e-2-date[3]:"       e-2-date[3].
MESSAGE "e-2-date[4]:"       e-2-date[4].
MESSAGE "e-2-date[5]:"       e-2-date[5].
MESSAGE "e-3-decimal:"       e-3-decimal[1].
MESSAGE "e-3-decimal:"       e-3-decimal[2].
MESSAGE "e-3-decimal:"       e-3-decimal[3].
MESSAGE "e-4-integer:"       e-4-integer[1].
MESSAGE "e-4-integer:"       e-4-integer[2].
MESSAGE "e-4-integer:"       e-4-integer[3].

```

If the project is configured correctly, the converted procedure will output:

```

f-1-char: ABCDef
f-1-char-sens: ABCDef
f-2-date: 08/12/22
f-3-decimal: 3.14
f-4-integer: 100
f-5-int64: 123456789012
f-6-logical: yes
f-7-datetime: 08/12/2022 04:31:26.297
f-8-datetimeetz: 08/12/2022 04:31:26.297+03:00
f-9-blob: ?
f-10-clob: null
f-11-raw: ?
f-12-recid: 11
e-1-char[1]: xYz 1
e-1-char[2]: ?
e-1-char[3]:
e-1-char[4]: xYz 4
e-1-char[5]: xYz 5
e-2-date[1]: 08/12/22
e-2-date[2]: ?
e-2-date[3]: 10/10/20
e-2-date[4]: 08/30/22
e-2-date[5]: 12/12/12
e-3-decimal: 0
e-3-decimal: ?
e-3-decimal: -1000
e-4-integer: 0

```

e-4-integer: ?
e-4-integer: -999

After the execution you can check the database using command maria line:

```
MariaDB [fwd]> select * from tm_all;
+-----+-----+-----+-----+-----+-----+-----+-----+
+
| recid | f_1_char | f_1_char_sens | f_2_date   | f_3_decimal | f_4_integer | f_5_int64   | f_6_logical | f_7
_datetime | f_8_datetimetz | f_8_datetimetz_offset | f_9_blob | f_10_clob | f_11_raw | f_12_recid |
+-----+-----+-----+-----+-----+-----+-----+-----+
+
|      11 | AbcDef   | AbcDef         | 2022-08-12 |          3 |          100 | 123456789012 |          1 | 202
2-08-12 04:31:26 | 2022-08-12 04:31:26 |          180 | NULL      | NULL         | NULL         |          11 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+
|      12 | nu-11    | 1111111111    | 2022-02-22 |        333 |           4 |          55555 |          0 | 197
7-07-07 07:07:07 | 1988-08-08 03:08:08 |          480 | NULL      | NULL         |              |        121212 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+
2 rows in set (0.000 sec)
```

```
MariaDB [fwd]> select * from tm_all__3;
+-----+-----+-----+-----+
| parent__id | e_3_decimal | e_4_integer | list__index |
+-----+-----+-----+-----+
|          11 |          0 |          0 |          0 |
|          11 |        NULL |        NULL |          1 |
|          11 |       -1000 |       -999 |          2 |
|          12 |         333 |          4 |          0 |
|          12 |         333 |          4 |          1 |
|          12 |         333 |          4 |          2 |
+-----+-----+-----+-----+
6 rows in set (0.000 sec)
```

```
MariaDB [fwd]> select * from tm_all__5;
+-----+-----+-----+-----+
| parent__id | e_1_char   | e_2_date   | list__index |
+-----+-----+-----+-----+
|          11 | xYz 1     | 2022-08-12 |          0 |
|          11 | NULL      | NULL       |          1 |
|          11 |           | 2020-10-10 |          2 |
|          11 | xYz 4     | 2022-08-30 |          3 |
|          11 | xYz 5     | 2012-12-12 |          4 |
|          12 | 1111111111 | 2022-02-22 |          0 |
|          12 | 1111111111 | 2022-02-22 |          1 |
|          12 | 1111111111 | 2022-02-22 |          2 |
|          12 | 1111111111 | 2022-02-22 |          3 |
|          12 | 1111111111 | 2022-02-22 |          4 |
+-----+-----+-----+-----+
10 rows in set (0.000 sec)
```

Everything seems fine, nothing fancy. ~~There is a catch: in my previous commit there is a typo bug so the conversion configuration requires mariadb as dialect. I will fix that with the next commit.~~ LE: fixed in r14156.

Regarding the hotel application. For the moment I have only the project converted and DDLs for mariadb generated. I checked and they (both datable and indices definitions) are compatible with the database. I do not have an import yet because of the lack of the script support, but I am also working on that.

How to configure your test project to work with MariaDb:

- update your `cfg/p2j.cfg.xml` to add generation of DDL for this dialect. For each namespace add `mariabd` (this is incorrect but slipped into FWD repository. I will fix this with next commit and update this note) like this:

```
<parameter name="ddl-dialects" value="postgresql,h2,sqlserver2012,mariadb" />
```

- the FWD build script already grabs the `mariadb-java-client-2.7.3.jar` from gradle repository, you need to add it to classpath of your project if not not done automatically;
- create a database and roles for accessing it. I used `fwd` name, running on standard port 3306. Access is done using `fwd_user` / `fwd_password`.
- update `deploy/server/registry.xml` with following settings (I replaced an old PostgreSQL configuration):

```
<node class="container" name="database">
  <node class="container" name="fwd">
    <node class="container" name="orm">
      <node class="string" name="dialect">
        <node-attribute name="value" value="com.goldencode.p2j.persist.dialect.MariaDbDialect"/>
      </node>
    <node class="container" name="connection">
      <node class="string" name="driver_class">
        <node-attribute name="value" value="org.mariadb.jdbc.Driver"/>
      </node>
      <node class="string" name="url">
        <node-attribute name="value" value="jdbc:mysql://localhost:3306/fwd"/>
      </node>
      <node class="string" name="username">
        <node-attribute name="value" value="fwd_user"/>
      </node>
      <node class="string" name="password">
        <node-attribute name="value" value="fwd_password"/>
      </node>
    </node>
  </node>
</node>
```

[...]

- start the server normally, connect with a standard client

#27 - 08/12/2022 09:31 PM - Ovidiu Maxiniuc

- % Done changed from 10 to 20

The revision 14156 brings several improvements for the dialect:

- I noticed they released a new GA JDBC driver (3.0.7) last week and updated the project to grab and use it;
- I can confirm the char r-trimming is provided out-of-the-box and char-indexes seems to work as expected;
- The decimal column were incorrectly generated with 0 decimals;
- Also, the support for datetime-tz works as expected. MariaDb does not support time-zoned data-types, but the JDBC driver will convert the values to local TZ. Luckily we already have the support for restoring the right TZ using the additional column injected at conversion time;
- I started writing queries with increasingly complex where predicates and I have not encountered issues yet.

#28 - 08/15/2022 08:17 PM - Ovidiu Maxiniuc

Please upgrade to FWD r14164 if you want to convert, import and start the hotel_gui test application. However, connecting with a client is not working for the moment in the absence of the native UDFs. The client is kicked out with the following exception:

```
(com.goldencode.p2j.persist.Persistence:SEVERE) [00000002:0000000D:bogus-->local/hotel/primary] error executing query [
select room, roomType
from Room__Impl__ as room, RoomType__Impl__ as roomType
where (room.active = true and
      true and
      lookup(toString(room.roomNum), ?0) = 0 and
      lookup(toString(room.roomNum), ?1) = 0) and
      (roomType.roomType = room.roomType)
order by room.roomNum asc, roomType.roomType asc]
com.goldencode.p2j.persist.PersistenceException: Error while processing the SQL list
Caused by: java.sql.SQLException: (conn=1834) FUNCTION hotel.lookup does not exist
    at org.mariadb.jdbc.internal.util.exceptions.ExceptionFactory.createException(ExceptionFactory.java:62)
)
    at org.mariadb.jdbc.internal.util.exceptions.ExceptionFactory.create(ExceptionFactory.java:155)
    at org.mariadb.jdbc.MariaDbStatement.executeExceptionEpilogue(MariaDbStatement.java:274)
    at org.mariadb.jdbc.ClientSidePreparedStatement.executeInternal(ClientSidePreparedStatement.java:229)
    at org.mariadb.jdbc.ClientSidePreparedStatement.execute(ClientSidePreparedStatement.java:149)
    at org.mariadb.jdbc.ClientSidePreparedStatement.executeQuery(ClientSidePreparedStatement.java:163)
    at com.mchange.v2.c3p0.impl.NewProxyPreparedStatement.executeQuery(NewProxyPreparedStatement.java:353)
    at com.goldencode.p2j.persist.orm.SQLQuery.list(SQLQuery.java:508)
    at com.goldencode.p2j.persist.orm.Query.list(Query.java:287)
    at com.goldencode.p2j.persist.Persistence.list(Persistence.java:1502)
    at com.goldencode.p2j.persist.ProgressiveResults.getResults(ProgressiveResults.java:1111)
    at com.goldencode.p2j.persist.ProgressiveResults.getResults(ProgressiveResults.java:1067)
    at com.goldencode.p2j.persist.ProgressiveResults.moveTo(ProgressiveResults.java:918)
    at com.goldencode.p2j.persist.ProgressiveResults.moveTo(ProgressiveResults.java:800)
    at com.goldencode.p2j.persist.ProgressiveResults.next(ProgressiveResults.java:414)
    at com.goldencode.p2j.persist.ResultsAdapter.next(ResultsAdapter.java:161)
    at com.goldencode.p2j.persist.AdaptiveQuery.next(AdaptiveQuery.java:1652)
    at com.goldencode.p2j.persist.CompoundQuery.processComponent(CompoundQuery.java:3104)
    at com.goldencode.p2j.persist.CompoundQuery.retrieveImpl(CompoundQuery.java:2705)
    at com.goldencode.p2j.persist.CompoundQuery.retrieve(CompoundQuery.java:2116)
    at com.goldencode.p2j.persist.CompoundQuery.retrieve(CompoundQuery.java:2001)
    at com.goldencode.p2j.persist.CompoundQuery.next(CompoundQuery.java:1154)
    at com.goldencode.p2j.persist.CompoundQuery.next(CompoundQuery.java:1045)
    at com.goldencode.p2j.persist.AbstractQuery.afterReposition(AbstractQuery.java:1369)
    at com.goldencode.p2j.persist.CompoundQuery.afterReposition(CompoundQuery.java:1876)
    at com.goldencode.p2j.persist.AbstractQuery.afterReposition(AbstractQuery.java:1338)
    at com.goldencode.p2j.persist.DynamicQuery.reposition(DynamicQuery.java:328)
    at com.goldencode.p2j.persist.CompoundQuery.reposition(CompoundQuery.java:374)
    at com.goldencode.p2j.persist.AbstractQuery.open(AbstractQuery.java:838)
    at com.goldencode.p2j.persist.DynamicQuery.open(DynamicQuery.java:578)
    at com.goldencode.p2j.persist.CompoundQuery.open(CompoundQuery.java:768)
    at com.goldencode.p2j.persist.QueryWrapper.open(QueryWrapper.java:785)
    at com.goldencode.hotel.AvailRoomsFrame.lambda$refreshAvailRooms$52(AvailRoomsFrame.java:1176)
    [...]
```

The hotel_gui was updated, after the checkout you only need to configure the build.properties.

I wrote a brief wiki for installing and basic configuration of the MariaDb server, available at:
https://proj.goldencode.com/projects/p2j/wiki/Database_Server_Setup_for_MariaDb_on_Linux.

#29 - 08/16/2022 10:50 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

I can confirm the char r-trimming is provided out-of-the-box and char-indexes seems to work as expected;

upper also should be unnecessary for columns representing case-insensitive fields, when using varchar(#). I assume we will need a different type mapping for case-sensitive fields. Our type mapping in the Dialect hierarchy may need to be more flexible to accommodate various type mappings in MariaDB for the same legacy type (based on configuration or traits in the original field, like case-sensitivity).

#30 - 08/16/2022 08:14 PM - Ovidiu Maxiniuc

With hotel_gui/r248 and 3821c/14168, I am able to start the client and authenticate. The Guests, Rates and Rooms tabs are populated but Available Rooms and Reservations are empty for the moment. I am investigating why (there are some logs related to the new dialect, which may point to the causes).

PS:

My attempt to add a new guest caused a Guest Accommodation Report with 225 entries (even if the respective tab was empty), to show in from of the Add dialog. I think this is a keystroke/shortcut handling issue unrelated to this task.

#31 - 08/19/2022 09:49 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

With hotel_gui/r248 and 3821c/14168, I am able to start the client and authenticate. The Guests, Rates and Rooms tabs are populated but Available Rooms and Reservations are empty for the moment. I am investigating why (there are some logs related to the new dialect, which may point to the causes).

Any update on the status of running Hotel GUI with MariaDB?

Is import working well?

What is left to do from [#6348-9](#), from the "Server Setup" and "FWD Runtime" categories (besides the UDF work)?

#32 - 08/19/2022 11:37 AM - Eric Faulhaber

Now that you are working with Hotel GUI (i.e., something that, while not a "real" application, is more than a simple test table), what is the practical impact of the limitations you outlined in [#6348-13](#)? Have you had to make further changes to deal with them?

#33 - 08/19/2022 03:00 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

My attempt to add a new guest caused a Guest Accommodation Report with 225 entries (even if the respective tab was empty), to show in from of the Add dialog. I think this is a keystroke/shortcut handling issue unrelated to this task.

IIRC, there is a certain key in one of these screens ("H", maybe?) which is hooked to generate a report. I think this was intended to show a feature, rather than being a bug. Maybe this is what you are referring to?

#34 - 08/19/2022 09:22 PM - Ovidiu Maxiniuc

- % Done changed from 20 to 60

Committed revision 14182.

Eric Faulhaber wrote:

Any update on the status of running Hotel GUI with MariaDB?

The status is unchanged. The reason for the pages not being loaded are the missing implementation of some UDFs.

Is import working well?

Yes, it is.

What is left to do from [#6348-9](#), from the "Server Setup" and "FWD Runtime" categories (besides the UDF work)?

- the sequences were not tested enough;
- injection of binary for CS compare is not done yet. My attempt to do it in FQLPreprocessor were not cleanly handled by FQL2SQL Converter so I decide to postpone this a bit;
- known character issue: a ~n character at the end of a string is not correctly trimmed when comparing strings;
- additional tests with character data;
- legacy collation. I did not put effort in this direction, mostly because the initial customer already picked the collation to use;
- the extents are working well (only normalized, but I do not expect issues with denormalized fields), but there is no change yet for JSON extents yet.

Now that you are working with Hotel GUI (i.e., something that, while not a "real" application, is more than a simple test table), what is the practical impact of the limitations you outlined in [#6348-13](#)? Have you had to make further changes to deal with them?

The dialect generates now string columns as varchar(100). I did not push the system intentionally, but with the screens which are working in Hotel application I have not seen any issue. I guess all the texts saved in DB are shorted than this limit. I think we can compute the N for each table dynamically (based on how many char columns are involved), but some customer hints might be better.

IIRC, there is a certain key in one of these screens ("H", maybe?) which is hooked to generate a report. I think this was intended to show a feature, rather than being a bug. Maybe this is what you are referring to?

Most likely. IIRC I tried to type "John Doe" as guest name, but only the couple of two letters appeared on screen. I did not know about this feature.

#35 - 08/22/2022 08:40 AM - Greg Shah

The dialect generates now string columns as varchar(100). I did not push the system intentionally, but with the screens which are working in Hotel application I have not seen any issue. I guess all the texts saved in DB are shorted than this limit. I think we can compute the N for each table dynamically (based on how many char columns are involved), but some customer hints might be better.

Basing this off actual data is a problem. I don't think customers will accept that the schema of each database instance may be different, even if it is just data sizes. The application will respond differently based on any different schema values such as this. For example, the same data value used in one instance might cause an error while in another database instance it will not cause the error.

#36 - 08/24/2022 10:24 AM - Constantin Asofiei

The Hotel GUI project is broken, when running ./prepare_hotel.sh on a fresh checkout (interestingly, running again a second time prepared directory.xml OK).

The problem is that all the maria DB configurations should be in the .template files or directory_db_mariadb.xml.template, not directly in build.properties, directory.xml, etc.

#37 - 09/09/2022 04:51 AM - Igor Skorniyakov

It looks that we need to use timestamp(3)/datetime(3) instead of timestamp/datetime in the tables' DDL for MariaDB. Otherwise the millisecond part will be truncated.

#39 - 09/12/2022 06:11 AM - Ovidiu Maxiniuc

Igor Skorniyakov wrote:

It looks that we need to use timestamp(3)/datetime(3) instead of timestamp/datetime in the tables' DDL for MariaDB. Otherwise the millisecond part will be truncated.

Igor,
Nice observation, thank you.
My testcase above failed to find it because of the heavy caching :((AKA: how to shot yourself in the foot). Both 3821c(14237) and 6129a(14395) were updated with correct datatypes.

Constantin, please use the new revision(s) in your customer environment.

#40 - 09/12/2022 10:10 AM - Igor Skorniyakov

I've found the following tool which can be useful - <https://www.linuxbabe.com/ubuntu/mysql-mariadb-database-performance-monitoring-percona>

#41 - 09/16/2022 09:40 PM - Ovidiu Maxiniuc

I have committed r14399/6129a. It contains improved compatibility for sorting NULL high for MariaDb.

Note that, MariaDb does not support indexes composed of expressions, only columns, but the columns can be computed / dynamically generated, so no additional disk space is used by the record.

Eric, please review. The Dialect's API is a bit changed to accommodate the new needs of this dialect.

Thank you!

#42 - 10/14/2022 06:22 AM - Constantin Asofiei

6129a/14487 contains some fixes related to sequences in MariaDB:

- fixes related to sequence management in MariaDB - for this dialect, the table and sequence names are case-sensitive.
- in MariaDB, current value is dependent on the connection, not the db server, so getting current value requires getting next value and adjusting back the sequence's value.

Ovidiu, the main issue is the case-sensitivity of sequence names (and tables, btw) in MariaDB. For sequences, please review these changes.

#43 - 10/14/2022 11:01 AM - Constantin Asofiei

Ovidiu, with an app I'm seeing lots of MISSING: MariaDbDialect.isQueryRangeParameterInlined() - is this a TODO related to performance or something critical? If it is related to performance, please remove or otherwise replace this logging.

#44 - 10/14/2022 12:17 PM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

Ovidiu, the main issue is the case-sensitivity of sequence names (and tables, btw) in MariaDB. For sequences, please review these changes.

I have reviewed my isolated testcases from when I worked on MariaDb sequences. In all my testcases the sequence names are converted to lowercase for SQL. For example:

```
ADD SEQUENCE "neXt-serial"  
  INITIAL 0  
  INCREMENT 1  
  CYCLE-ON-LIMIT no  
  MIN-VAL 0
```

is converted to

```
@Sequence(name = "ne_xt_serial", legacy = "neXt-serial", start = 0L, increment = 1L)  
ne_xt_serial,
```

How are you able to keep then in various casing?

After further investigations, it seems that MariaDb implementation is really strange. For example, you **cannot set a value which is less than the current value**. You need to recreate the whole sequence for that:

```
MariaDB [fwd]> CREATE OR REPLACE SEQUENCE s1 START WITH 1 MINVALUE 1 MAXVALUE 99 INCREMENT BY 1;
Query OK, 0 rows affected (0.039 sec)
```

```
MariaDB [fwd]> select nextval(s1);
+-----+
| nextval(s1) |
+-----+
|          1 |
+-----+
```

```
MariaDB [fwd]> select setval(s1, 20);
+-----+
| setval(s1, 20) |
+-----+
|          20 |
+-----+
```

```
MariaDB [fwd]> select setval(s1, 10);
+-----+
| setval(s1, 10) |
+-----+
|          NULL |
+-----+
```

```
MariaDB [fwd]> select lastval(s1);
+-----+
| lastval(s1) |
+-----+
|          NULL |
+-----+
```

```
MariaDB [fwd]> select nextval(s1);
+-----+
| nextval(s1) |
+-----+
|          21 |
+-----+
```

Also, as you already noticed, the current value, which in MariaDb is obtained using **lastval** function will return **NULL** immediately after set of create (that is, without calling nextval first).

#45 - 10/14/2022 12:23 PM - Constantin Asofiei

Ovidiu Maxiniuc wrote:

How are you able to keep then in various casing?

Set `conversion="minimal"` in `p2j.cfg.xml` for the namespace node.

After further investigations, it seems that MariaDb implementation is really strange. For example, you cannot set a value which is less than the current value. You need to recreate the whole sequence for that:

That's unfortunate - this means there is no 'current value' support in MariaDB.

Also, as you already noticed, the current value, which is obtained using `lastval` function in MariaDb will return NULL immediately after set of create (that is, without calling `nextval` first).

`lastval` will always return null for a new connection, if `nextval` is not invoked first. This is dependent on the i.e. JDBC connection.

#46 - 10/14/2022 12:23 PM - Constantin Asofiei

Regarding case-sensitivity in MariaDB: this is OS-dependent, it will be case-insensitive on Windows.

#47 - 10/14/2022 12:24 PM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

Regarding case-sensitivity in MariaDB: this is OS-dependent, it will be case-insensitive on Windows.

This makes sense, as tables (hence the sequences) in `mysql/mariadb` are files.

#48 - 10/14/2022 12:51 PM - Tijs Wickardt

Fun fact (nothing more): you can set NTFS to case sensitive in Windows.
That will OE classes in Windows to become case sensitive as well.
I heard of a development team that used that for strictness :).

#50 - 10/14/2022 09:33 PM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

6129a/14487 contains some fixes related to sequences in MariaDB:

- fixes related to sequence management in MariaDB - for this dialect, the table and sequence names are case-sensitive.
- in MariaDB, current value is dependent on the connection, not the db server, so getting current value requires getting next value and adjusting back the sequence's value.

Ovidiu, the main issue is the case-sensitivity of sequence names (and tables, btw) in MariaDB. For sequences, please review these changes.

If the tables and sequences are case sensitive (at least for Linux) then we should not use 'to-lower' on these names. However, it is mandatory to normalize the ABL names in order to map them correctly. I have some pending changes to your fixes in this regard.

With respect to the second issue: your patch will not hold. As noted above, in MariaDb sequences are really strict. They cannot be 'turned back' to a value less than the current one. So the trick with burning a value and reset it back will not work. The nasty part is that no error is generated, only null is returned instead of desired value, as returned when everything is fine.

Also, there is one more issue with the fix: you started with the assumption that all sequences are increment 1. This is true for this poc project, but not generally. A temporary fix would be to replace
`long currentValue = nextVal.longValue() - 1;`
with
`long currentValue = nextVal.longValue() - seq.getIncrement();`
Anyway, this will also not work for cycling sequences if invoking the next-value rolls the sequence around.

I am trying to obtain this information (current-value) from mariadb using some metadata information. It might be possible, but a long shot and admin role could be necessary.

#51 - 10/15/2022 03:29 AM - Constantin Asofiei

There is another issue in MariaDB - sequences and tables are in the same namespace. In OE, IIRC they are not in the same namespace. So you can't have a sequence and a table with the same SQL name in MariaDB.

#52 - 10/25/2022 09:27 PM - Ovidiu Maxiniuc

Further information on sequences.

It seems that we cannot rely on `lastval()` method at all to detect the current value of the sequence. As noted above, it will return null if no `nextval()` was used in current connection. But things are stranger. If a sequence is reset to a new value after the `nextval()` was used, `lastval()` will continue to return the same value and not even null. An example:

```
setval(seq, 1, false) => 1
lastval(seq)          => null
nextval(seq)          => 1 // this is OK
lastval(seq)          => 1 // this is OK
setval(seq, 10, false) => 10
lastval(seq)          => 1 // this is NOT OK
```

However, the `setval()` itself is not very useful: the attempt to set the sequence back will fail. To continue the example above:

```
setval(seq, 5, false) => null
```

We need to `ALTER SEQUENCE seq RESTART WITH 5` to reset the sequence. This is more time consuming because a lock is needed internally in MariaDb and possible, a high-level role (admin?).

#53 - 10/26/2022 06:48 AM - Greg Shah

It seems like sequences are not implemented in MariaDB for general usage. There must be some kind of use case that actually works and makes sense. What is that normal use case (for how people are "supposed to" use sequences in MariaDB)? Perhaps if we understand that, we will see the way things can be mapped for our usage.

Or perhaps it is just my ignorance and sequences really just don't work in MariaDB. :(

#54 - 10/26/2022 03:22 PM - Ovidiu Maxiniuc

I have an update which passes my (rather simple) testcase (unlike the current implementation from 6129 branches). I really do not like it because:

- the `set-value()` uses two statements:
 `ALTER SEQUENCE seq RESTART WITH n`
 followed by a
 `SELECT SETVAL(seq, n, FALSE)`

- The second statement is still necessary because otherwise the first value obtained will be $n + \text{increment}$ instead of just n ;
- the current-value is not looking at `lastval()` function at all. Even if it `nextval()` was called in current session, there is a chance that it have been also called on other session and the value is expired. So it is implemented as Constantin suggested:
`SELECT NEXTVAL(seq)`
followed by the `set-value()` back, as described above. Not very optimal :(.
• only the `next-value()` seems to work as expected.

Note: I even thought of implementing the sequences as a table, but this will not work for multiple reasons. The primary reason would be the transactions: the values of sequences will be restored in the event of a rollback.

#55 - 01/24/2023 10:40 AM - Eric Faulhaber

Ovidiu, as of 3821c/14182, we have this issue marked as 60% done. Many changes have been made since then, including lenient mode. What would you say is the correct % done at this point? What is left to implement? Thanks.

#56 - 01/26/2023 10:58 PM - Ovidiu Maxiniuc

- % Done changed from 60 to 80

Indeed, there were several updates to the dialects. But there are some unfinished features:

- what I know for sure is not implemented is the case-sensitive query support;
- I noticed that the `schema_word_tables` are not yet generated. Indeed, the `useWordTables()` returns false and `generateWordTablesDDLImpl()` returns just the EOLN. I do not know if the project uses them;
- beside these, there are a few methods which still use `UnimplementedFeature` to report missing features (`createSequenceHandler`, `getSyntheticIndexSize`), but I think they should be cleaned-up.

Overall, I roughly estimate the completeness to 80, maybe 90%.

#57 - 01/31/2023 01:35 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

... there are a few methods which still use `UnimplementedFeature` to report missing features (`createSequenceHandler`, `getSyntheticIndexSize`), but I think they should be cleaned-up.

Please go ahead and do this.

#58 - 01/31/2023 04:12 PM - Ovidiu Maxiniuc

Committed revision 14396 / 6129b.

#59 - 01/31/2023 05:11 PM - Eric Faulhaber

Ovidiu, are all the sequence features working now? Based on [#6348-54](#) and the entries leading up to it, it seemed this might not be finished, or perhaps not committed to 6129b.

#60 - 02/01/2023 04:28 PM - Ovidiu Maxiniuc

The update I was talking in [#6348-54](#) was committed the next day in revision 14308 (see MariaDbSequenceHandler.java). It is not the perfect match with 4GL but I think it's the optimal solution. Note that, because of the peculiarities of how 4GL we might have compatibility issues with the other dialects, too, for specific edge-cases.

#61 - 02/03/2023 10:50 AM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

Committed revision 14396 / 6129b.

The change looks good. Can we close this task?