

w: www.mariadb.com

e: info@mariadb.com

MariaDB CONNECT Storage Engine

Introduction

IT organizations produce a lot of data. Currently most of the data is in flat files formats. To access this data for a Business Intelligence purpose a lot of effort has to be done. This effort involves expertise and complex processes with ETL and database loading. Not everyone can afford this effort just to query day to day data. In many cases this data remains unused or is not queried in a consolidated way.

The main use case of the CONNECT Storage engine is to ease access to this category of data. It offers access to various flat files format, to ODBC data sources and to remote MySQL databases.

How non relational data is usually treated in BI

Most of the time before being accessed the data need to go through an ETL system to be put in the right format. It is then loaded into a database (relational or not) to be efficiently queried.

This lead to a complex and lengthy processing. The data that is queried is not up to date. Request are considered as fast when counting only the query time and not counting the transformation/loading process. Most of these architectures are quite rigid and requires IT resources for any changes.

Business Intelligence Users expectations

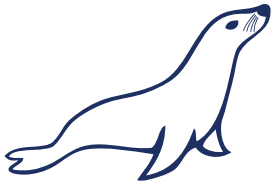
Many business intelligence user would like to have directly access to real time data. This is very critical in many business areas.

In many cases the data is not ready to be loaded. The users that is going to query it does not have the expertise to do it himself. Being able to access this data directly through the CONNECT storage engine brings a lot of value to users.

How and why was MariaDB CONNECT Storage Engine created ?

The contributor for this storage engine is Olivier Bertrand who has a 50 years experience in database development. As an ex IBM database researcher Olivier has a very extensive experience of relational databases technologies. Olivier was very interested in adding to MariaDB the result of his experience regarding BI data access. He wanted a more versatile way to access external data sources in various formats. The CONNECT storage engine enables MySQL and MariaDB to directly use external data as if it were tables without being obliged to load them into the physical database.

Olivier has been welcomed by the MariaDB team and has been working with them to integrate his work, build test cases and fix remaining bugs. The CONNECT storage engine is 100% open source and released under the GPL license. It really fulfills a gap in some area of BI. The integration to MariaDB shows that MariaDB is a real ground of innovation and very friendly to third party storage engines developers. The "MariaDB CONNECT Storage Engine" adds one more innovative differentiation to the MariaDB ecosystem (Cassandra and LevelDB are two other nice examples). All these extra features are under GPL and they come with the standard MariaDB release.



Why a storage engine for MariaDB ?

MySQL has an open architecture with a well specified API allowing to add plugins. There are various kind of plugins ; authentication, information schema, audit and Storage Engine plugins. The storage engine API has brought a lot of innovation to MySQL.

The CONNECT storage engine use extensively all the capabilities of the storage engine API and some useful MariaDB extension that allow better integration of a storage engine. The MariaDB server is now considered as a database platform federating various unique technologies.

Why not use the CSV, Federated & Merge Storage Engines ?

There was since MySQL 5.1 a CSV storage engine offering a very limited coverage access to a particular format of flat files. This storage engine has many limitations such as dealing with only one file format, CSV, limited performance (no block reading and no indexing).

There is also a federated(X) storage engine. This storage engine allows access to a remote MySQL database. Unfortunately it has many restriction that considerably limit its usage.

Another interesting is the Merge storage Engine. This storage engine allows access to a set of tables as if they were a single table. Unfortunately it has many restriction that considerably limit its usage. All the tables need to be MyISAM and they need to have a strictly identical structure.

The MariaDB CONNECT Storage Engine covers all the functionalities of the CSV + FEDERATED + MERGE storage engines. It has none of their limitations . It offers much broader use cases (XML, dbase, ODBC...). The implementation is smart and uses advanced features (indexing, compression, condition push down). The MariaDB CONNECT Storage Engine targets BI. It is not designed for OLTP.

Key features of the MariaDB CONNECT Storage Engine

The MariaDB CONNECT Storage Engine takes into account many BI related feature. As it is not be targeted as an OLTP a lot of costly internal mechanism are not implemented (transaction, Multiple view consistency ...). This allow this storage engine to be faster for read only activities.

Here are some of the key features of the MariaDB CONNECT storage engine :

Multiple File Tables

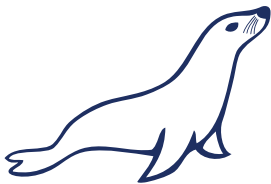
A multiple file table is one that is physically contained in several files of the same type instead of just one. These files are processed sequentially during the process of a query and the result is the same as if all the table files were merged into one. This is great to process files coming from different sources (such as cash register log files) or made at different time periods (such as bank monthly reports) regarded as one table. Note that the operations on such files are restricted to sequential Select and

Update.

The FILEID special column, described later in this document, allows filtering the file list or doing some grouping on the files that make a multiple table.

Big File tables

Because all files are handled by the standard input/output functions of the operating system, their size is limited to 2GB, the maximum size handled by standard functions. For some table types, CONNECT can deal with files



that are larger than 2GB, or prone to become larger than this limit. These are the FIX, BIN and VEC types. To tell CONNECT to use input/output functions dealing with big files, specify 'huge=1' in the OPTION_LIST or COMMENT string for that table. Note however that CONNECT does not support tables having more than 2G records.

Compressed Tables

CONNECT can make and processed some tables whose data file is compressed. The only supported compression format is the gzlib format. Zip and zlib formats are not supported. The table types that can be compressed are DOS, FIX, BIN, CSV and FMT. This can save some disk space at the cost of a somewhat longer processing time.

ODBC tables

The CONNECT storage engine offer the capability to access table through an ODBC driver. Through ODBC it is possible to access local or remote tables. The column can be partially used and reordered. The conditions included in the where clause are pushed to the foreign data source. This functionality give access to a wide range of data sources : MS Access, Excel, Firebird, SQLServer, DB2, Oracle ...

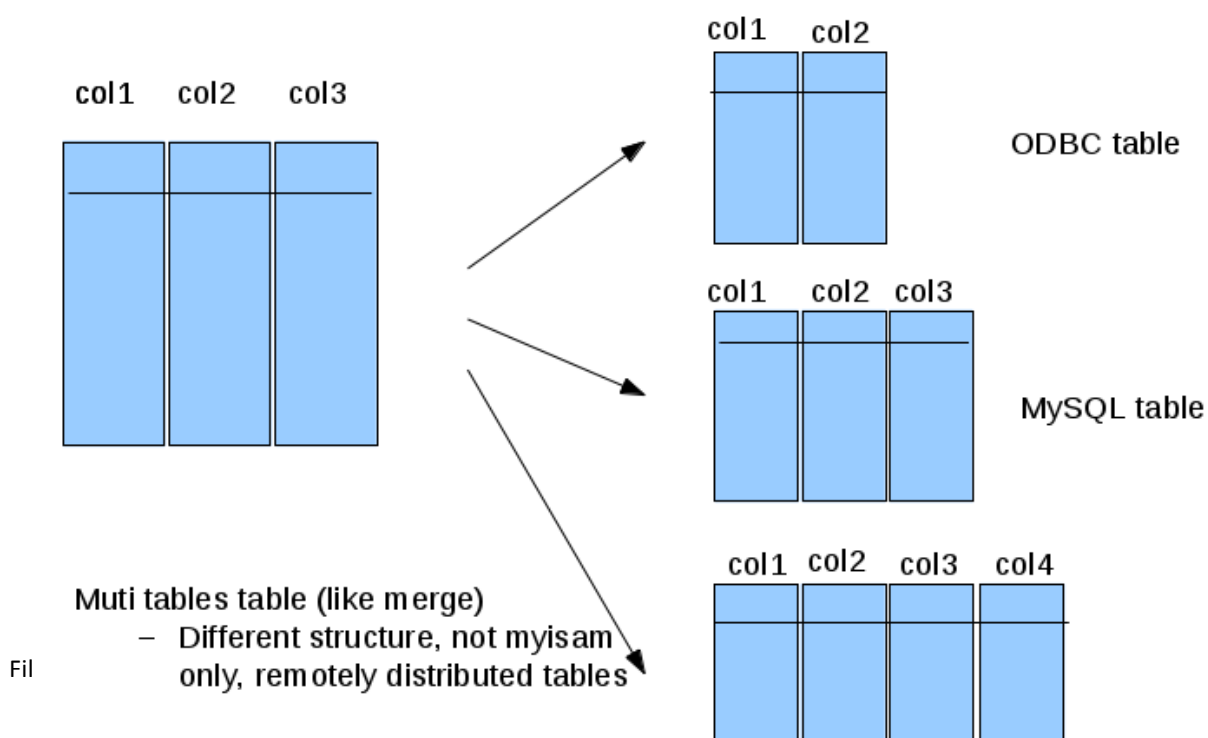
For ODBC tables related to files it is possible to have a table consolidating multiple files. For example this is useful to have a consolidated vision of multiple monthly excel files.

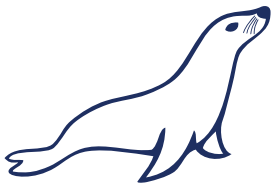
MySQL tables

The CONNECT storage engine offer the capability to access MySQL tables located on a local or remote server. Contrary to federated columns can be a subset of the target and type conversion can take place. Compared to federated this implementation is more complete. It handles condition push down and limit clause.

Table list table (TBL)

The CONNECT storage engine allow to define a table which is build of multiple tables. This closed to the concept of merge table that already exists. This e implementation allows to use CONNECT table as member of the list. This allow to have a distributed table made of MySQL or ODBC table.

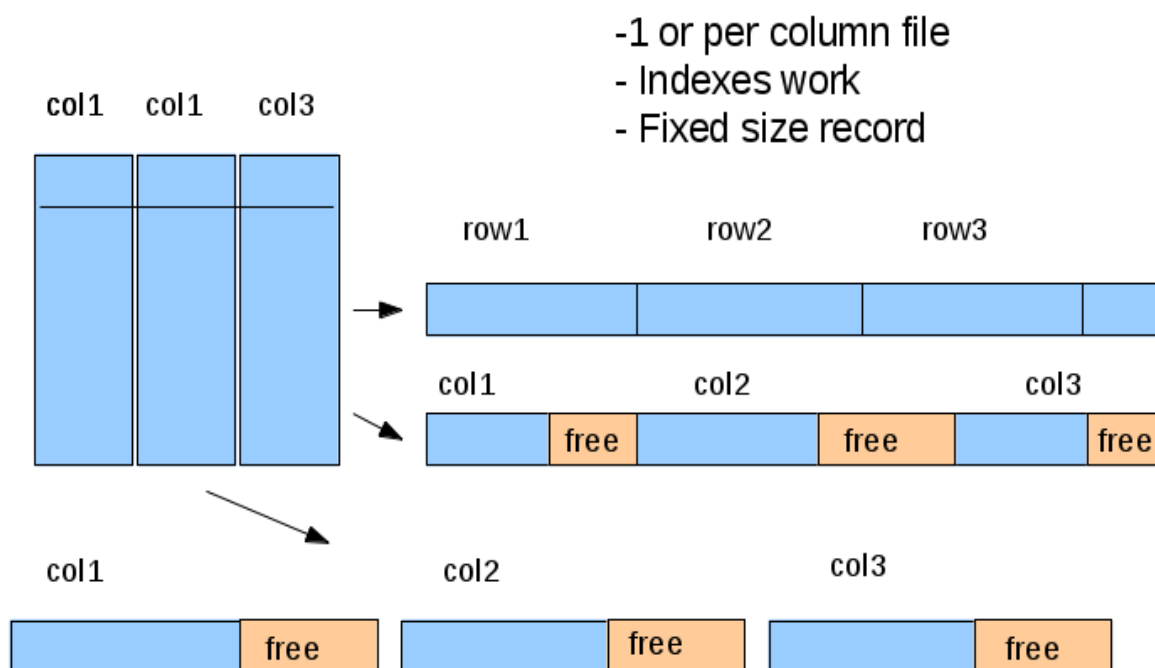




All the limitations of the MERGE Storage Engine have been lifted. The tables are not restricted to MyISAM only. The tables are not required to have the same structure. The final table can reorder the columns and use a subset of them.

Column Store / VEC Tables

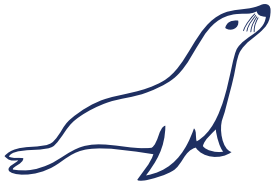
The CONNECT Storage Engine allows to define a table where data is store on a per column basis. This is very optimal for tables having many columns. This is often the case in a BI context where the data model is denormalized. Many of the request only address a small subset of all of the columns.



Tables of type **vec** are binary files that are designed to provide extremely good performance on read-intensive query workloads. CONNECT organizes their data on disk as columns of values from the same attribute, as opposed to storing it as rows of tabular records. This organization means that when a query needs to access only a few columns of a particular table, only those columns need to be read from disk. Conversely, in a row-oriented table, all values in a table are typically read from disk, and wasting I/O bandwidth.

All this can be in one physical file or each column data can be in a separate file. The "Split" format with separate files is a much safer solution when the table must be frequently inserted or shared among several users.

Like for **BIN** tables, numeric values are stored using platform internal layout, the correspondence between column types and internal format being the same than the default ones given above for **BIN**. However, field formats are not available for **vec** tables.



w: www.mariadb.com

e: info@mariadb.com

XML tables

CONNECT supports tables represented by XML files. This is a powerful feature as any single column can be associated with an Xpath query. For these tables, the standard input/output functions of the operating system are not used but the parsing and processing of the file is delegated to a specialized library.

XML is a very general language used to encode data having any structure. In particular, the tag hierarchy in an XML file describes a tree structure of the data.

When we query an XML table CONNECT is able to present multi-nodes value either as multiple rows or as a column with coma separated values.

MariaDB / MySQL features fitted for BI

Optimizer improvement

The MySQL optimizer has made great progress. This allow MySQL to execute much more efficiently decision support type queries. This queries most of the time include subqueries and requires IOs to be minimized. Benchmarks like dbt3 that have been run with latest versions have shown considerable improvement in query response time.

Some of the important optimizations :

- Subquery optimizations
- Optimizations for derived tables and views
- Disk access optimization: Index Condition Pushdown & Multi-Range-Read optimization
- Many Join optimizations including : Block Hash Join and Block Index Join (Batch Key Access Join)

MySQL partitioning

MySQL partitioning allow to partition a tables according to a partition key/strategy. This allow to speed up some queries by using some of the WHERE clause criteria to eliminate one or more partitions(*partition pruning*).

Dynamic columns

MariaDB Dynamic columns allows one to store different sets of columns for each row in a table. It works by storing a set of columns in a blob and having a small set of functions to manipulate it.

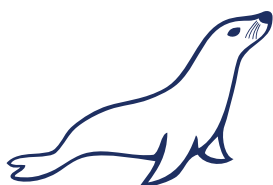
virtual columns

MariaDB allows to define a column based on any legal deterministic expression which can be calculated. It can be persitent or not.

Cassandra storage Engine

MariaDB allow to access through SQL a Cassandra database. The Cassandra Storage Engine (SE) allows access to Cassandra databases from MariaDB, and to provide data integration between the SQL and NoSQL worlds. This allows to read and write in Cassandra.

Cassandra belongs to the distributed column store families. You can perform joins on Cassandra data, or against data in Cassandra and data in MariaDB.



w: www.mariadb.com

e: info@mariadb.com

Multi-source replication

MariaDB has enhanced MySQL replication to support Multi-source replication. This allows multiple MySQL servers to simultaneously replicate onto a single MariaDB Server. One of the typical use cases for multi-source replication is to partitioning your data over many masters and then get it all together on one machine to do analytical queries on all data.

Conclusion

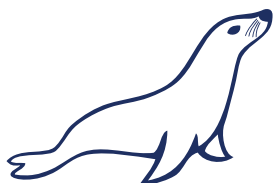
The MariaDB CONNECT Storage Engine brings a lot of value to users interested in accessing various external data sources. It is well complemented by the set of optimizations and specific functionalities of MariaDB.

References

MariaDB website : <http://www.mariadb.com>

Appendix A: File type supported by CONNECT Storage Engine

Type	Description
DOS	The table is contained in one or several files. The file format can be refined by some other options of the command or more often using a specific type as many of those described below. Otherwise, it is a flat text file where columns are placed at a fixed offset within each record, the last column being of variable length.
FIX	Text file arranged like DOS but with fixed length records.
BIN	Binary file with numeric values in platform representation, also with columns at fixed offset within records.
VEC	File in which each record contains the column values in a non-standard format (the same for each record) .This format is specified in the column definition.
DBF	File having the dBASE format.
CSV	“Comma Separated Values” file in which each variable length record contains column values separated by a specific character (defaulting to the comma)
FMT	File in which each record contains the column values in a non-standard format (the same for each record) This format is specified in the column definition.
INI	File having the format of the initialization or configuration files used by many applications.
XML	File having the XML or HTML format.



w: www.mariadb.com

e: info@mariadb.com

ODBC	able extracted from an application accessible via ODBC. For example from another DBMS or from an Excel spreadsheet. (Windows only)
DIR	Virtual table that returns a file list like the DOS dir command.
WMI	Windows Management Instrumentation table displaying information coming from a WMI provider. This type enables to get in tabular format all sort of information about the machine hardware and operating system (Windows only)
MAC	Virtual table returning information about the machine and network cards (Windows only)
OEM	Table of any other formats not directly handled by CONNECT but whose access is implemented by an external DLL (or Shared Library).