Playing with CONNECT

Federico Razzoli
$ whoami

Hi, I’m Federico Razzoli from Vettabase Ltd

Database consultant, open source supporter, long time MariaDB and MySQL user

- vettatable.com
- Federico-Razzoli.com
What is CONNECT?
What is a Storage Engine?

- MariaDB knows nothing about...
  - Writing / reading data
  - Writing / reading indexes
  - Caching data and indexes
  - Transactions
  - ...
- These functionalities are implemented in special plugins called storage engines
- InnoDB is the default storage engine
Some storage engines do strange things...

- BLACKHOLE
- SEQUENCE
- SPIDER
- CSV
What is a Storage Engine?

The list can vary depending on MariaDB version and distribution

MariaDB [(none)]> SELECT * FROM information_schema.ENGINES
                WHERE ENGINE = 'CONNECT' \G

*************************** 1. row ***************************
  ENGINE: CONNECT
  SUPPORT: YES
  COMMENT: Management of External Data (SQL/NOSQL/MED), including Rest query results
  TRANSACTIONS: NO
    XA: NO
  SAVEPOINTS: NO
1 row in set (0.000 sec)
CONNECT

- CONNECT is designed for MED (Management of External Data)
- It connects MariaDB to data stored in another form
- It depending on the TABLE_TYPE it can do many things:
  - Use data from remote DBMSs
  - Use data from files in various formats
  - Special data sources
  - Data transformation
File-Based Tables
### Inward / Outward

A file-based table can be inward or outward.

- If `FILE_NAME` is specified, the table is outward.
- Outward tables are assumed to be “holy”.

```sql
CREATE TABLE file_table (
    a INT,
    b INT
)

ENGINE = CONNECT,
TABLE_TYPE = CSV,
FILE_NAME = 'data.csv';
```
ALTER TABLE on File-Based tables

ALTER TABLE file_table DROP COLUMN a;

- If the table is Outward:
  - A column disappears from the table;
  - But the underlying file remains unchanged.
- If the table is Inward, the underlying file is modified
Inward Tables

CREATE TABLE csv_data ( ... ) ENGINE = CONNECT, TABLE_TYPE = CSV;

- The CSV file will be located in the database directory
- In this case, the file name will be csv_data.csv
- To know the exact name:
  - SHOW WARNINGS;
  - Regexp to get the filename: \s(\w+)$
Import + Modify + Export

- An interesting use case for CONNECT is:
  - Receive data in a certain understood format
  - Make some changes that are easier in SQL
    - `SELECT column_list FROM table`
    - `SELECT a, AVG(b) FROM table GROUP BY a`
  - Export the data in the same format
Import + Modify + Export

- This can be done:
  - Create an Outward table
  - `CREATE TABLE exported_data SELECT ...`
  - Copy the table elsewhere and DROP it

- Or, for more complex transformations:
  - Create an Outward table
  - `CREATE TABLE intermediate_data ... ENGINE InnoDB;`
  - Add indexes as needed
  - Make some data transformation
  - `CREATE TABLE exported_data
    ENGINE=CONNECT, TABLE_TYPE=CSV, SEP_CHAR='\t', HEADER=1
    SELECT * FROM intermediate_data`
  - Copy the table elsewhere and DROP it
Exporting data

```sql
ALTER TABLE numbers
    ENGINE = CONNECT,
    TABLE_TYPE = CSV,
    SEP_CHAR = '\t';
```

- This is the most efficient way to transform a table that you don’t need anymore
- But the file will be created in MariaDB datadir, you cannot specify a different path for Inward tables
Let’s try reading Apache logs
Sample

- A small sample of vettabase.com Apache error log
- IPs are scrambled
40.88.21.225 - - [07/Sep/2020:17:11:22 +0000] "GET / HTTP/1.1" 302 - "http://vettabase.com/" "Mozilla/5.0 (compatible; DuckDuckGo-Favicons-Bot/1.0; +http://duckduckgo.com)"
198.100.126.179 - - [07/Sep/2020:17:14:34 +0000] "GET /admin/ HTTP/1.1" 404 - "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0"
120.26.50.46 - - [07/Sep/2020:18:33:24 +0000] "HEAD /caiyuan/login.php HTTP/1.1" 404 - "-" "-"
120.27.51.66 - - [07/Sep/2020:18:33:24 +0000] "HEAD /guanli/login.php HTTP/1.1" 404 - "-" "-"
120.27.51.66 - - [07/Sep/2020:18:33:24 +0000] "HEAD /admin/login.php HTTP/1.1" 404 - "-" "-"
Mmm... 

- A precise machine-readable format is used
- But it’s a bit irregular - a bit less machine readable than CSV or JSON
- We’ll have to define a way to parse the data we need
- We only care about some columns
40.88.21.225 - - [07/Sep/2020:17:11:22 +0100] "GET / HTTP/1.1" 302 - "http://vettabase.com/" "Mozilla/5.0 (compatible; DuckDuckGo-Favicons-Bot/1.0; +http://duckduckgo.com)"

- ip: 40.88.21.225
- time: 07/Sep/2020:17:11:22
- timezone: +0100
- request_type: GET
- path: /
- protocol: HTTP/1.1
- http_response_code: 302
CREATE OR REPLACE TABLE web_log (  
ip VARCHAR(15) NOT NULL FIELD_FORMAT = '%n%s%- -' ,  
time VARCHAR(100) NOT NULL FIELD_FORMAT = '%n%s' ,  
timezone VARCHAR(50) NOT NULL FIELD_FORMAT = ' %n%s "' ,  
request_type VARCHAR(5) NOT NULL FIELD_FORMAT = '%n%s' ,  
path VARCHAR(200) NOT NULL FIELD_FORMAT = ' %n%s' ,  
protocol VARCHAR(10) NOT NULL FIELD_FORMAT = ' %n%s ' ,  
http_response_code SMALLINT UNSIGNED NOT NULL FIELD_FORMAT = '%n%d%n' )  
ENGINE = CONNECT ,  
TABLE_TYPE = 'FMT' ,  
FILE_NAME= '/var/shared/apache.log' ;
-- If you forget to specify the full path

Warning (Code 1105): Open(rb) error 2 on
/usr/local/mariadb/data/../test/apache.log: No such file or directory

-- If a FIELD_FORMAT is not correct or the file has lines in an
-- inconsistent format

ERROR 1296 (HY000): Got error 122 'Bad format line 1 field 3 of web_log'
from CONNECT
MariaDB [test]> SELECT * FROM web_log LIMIT 1 \G
*************************** 1. row ***************************
ip: 114.119.159.128
    time: [07/Sep/2020:13:17:13
    timezone: +0100]
request_type: GET
    path: /robots.txt
    protocol: HTTP/1.1"
http_response_code: 404
CREATE OR REPLACE TABLE web_log (  
  ...  
  , time VARCHAR(100)  
    GENERATED ALWAYS AS (SUBSTRING(raw_time FROM 2))  
  , timezone VARCHAR(5)  
    GENERATED ALWAYS AS (SUBSTRING(raw_timezone FROM 1 FOR CHAR_LENGTH(raw_timezone) - 1))  
  , protocol VARCHAR(10)  
    GENERATED ALWAYS AS (SUBSTRING(raw_protocol FROM 1 FOR CHAR_LENGTH(raw_protocol) - 1))  
)  
  ...  
;
MariaDB [test]> SELECT ip, time, timezone, protocol FROM web_log LIMIT 1
\G
*************************** 1. row ***************************
   ip: 114.119.159.128
      time: 07/Sep/2020:13:17:13
   timezone: +0100
dump protocol: HTTP/1.1
1 row in set (0.002 sec)
Let’s do some analyses
Analyses on a log

MariaDB [test]> SELECT http_response_code, COUNT(*) FROM web_log GROUP BY http_response_code;
+--------------------+----------+
| http_response_code | COUNT(*) |
+--------------------+----------+
| 302                | 11       |
| 404                | 61       |

MariaDB [test]> SELECT request_type, COUNT(*) FROM web_log GROUP BY http_response_code;
+--------------+----------+
| request_type | COUNT(*) |
+--------------+----------+
| GET          | 11       |
| GET          | 61       |
PIVOTing a table

- Doing some analyses on logs is cool
- But we’d like to pivot a table, and MariaDB doesn’t support the PIVOT syntax
- But we can use CONNECT’s PIVOT table type
CONNECT user

- CONNECT tables that transform data from other tables need to establish a connection to MariaDB and run queries
- In order to do that, they need to use an account
- It is a good practice (and default) to have a `mysql@localhost` account
Creating a PIVOT table

- Note that the table definition contains CONNECT’s user
- SHOW CREATE TABLE shows this info
- This is why it is best to use *unix_socket* authorisation plugin

```sql
CREATE OR REPLACE TABLE requests_by_response_and_type
    ENGINE = CONNECT,
    TABLE_TYPE = PIVOT,
    TABNAME = 'web_log',
    OPTION_LIST = 'user=mysql,host=localhost,
                    PivotCol=request_type,Function=count'
;
```
Reading our PIVOT table

```sql
MariaDB [test]> SELECT * FROM requests_by_response_and_type ;
+-----------------+-----------------------+--------------+------------------
| ip              | raw_time              | raw_timezone | path              |
| raw_protocol    | GET                   | HEAD         |
+-----------------+-----------------------+--------------+------------------
| HTTP/1.1"       | 0                     | 1            | 112.124.0.114    | [07/Sep/2020:16:44:25 | +0100]   |
|                 |                       |              | /dedea/login.php |
HTTP/1.1"       | 0                     | 1            |
...```
Reading our PIVOT table

MariaDB [test]> SELECT request_type, `GET`, `HEAD` FROM requests_by_response_and_type ;

ERROR 1054 (42S22): Unknown column 'request_type' in 'field list'
PIVOTing a query

```sql
OPTION_LIST = 'user=mysql,host=localhost',
SrcDef = 'SELECT request_type, COUNT(*) FROM web_log
        GROUP BY request_type'
```

```
MariaDB [test]> SELECT * FROM requests_by_response_and_type ;
+-----+------+
| GET | HEAD |
+-----+------+
|  23 |   49 |
+-----+------+
```
Other transformations?

- OCCUR unpivots columns
- XCOL turns lists into multiple rows
- TBL allows to treat a set of tables as a single table
What did we leave out?

- Almost everything! We’ve just played a bit with an Apache log!
- Other file formats (JSON, XML, HTML tables, ini, fixed-length, ...)
- Compressed files
- More magic with custom formats
- Connections via MySQL format, ODBC, JDBC, MongoDB
- Querying remote REST API’s
- ...and more

Final hints:
  build proper indexes where possible
  increase connect_work_size if your files are big
Thanks for attending!
Question time :-)