

MariaDB Feature Plan - Replication & DDL

Lixun Peng

Agenda

- Multi-Source Replication: Adding For Channel Syntax
- Recycle Bin for DDL
- Flashback for DDL
- Purging Large Table Asynchronously

Syntax Difference for Multi-Source

- **MariaDB:** ... ['connection_name'] ...
- **MySQL:** ... FOR CHANNEL 'channel' / 'channel': ...
- **Purpose:** Adding MySQL 'for channel' syntax for MariaDB multi-source replication and keep the original MariaDB 'connection_name' syntax in the same time.

MariaDB Syntax Now

- CHANGE MASTER [*'connection_name'*] TO
- FLUSH RELAY LOGS [*'connection_name'*]
- MASTER_POS_WAIT(.....,*'connection_name'*)
- RESET SLAVE [*'connection_name'*] [ALL].
- SHOW RELAYLOG [*'connection_name'*] EVENTS
- SHOW SLAVE [*'connection_name'*] STATUS
- START/STOP SLAVE [*'connection_name'...*]
- START/STOP ALL SLAVES
- SHOW ALL SLAVES STATUS
- set @@**default_master_connection**= [*'connection_name'*];
- --*'connection_name'*.**replicate_do_db** / **replicate_do_table** ... =

MySQL Syntax Now

- CHANGE MASTER TO ***[FOR CHANNEL 'channel_name']***
- START SLAVE ***[FOR CHANNEL 'channel_name']***
- STOP SLAVE ***[FOR CHANNEL 'channel_name']***
- SHOW RELAYLOG EVENTS ***[FOR CHANNEL 'channel_name']***
- FLUSH RELAY LOGS ***[FOR CHANNEL 'channel_name']***
- SHOW SLAVE STATUS ***[FOR CHANNEL 'channel_name']***
- RESET SLAVE ***[FOR CHANNEL 'channel_name']***
- --replicate-do-db=**channel:database_id**
- --replicate-ignore-db=**channel:database_id**

Recycle Bin for DDL

- The same with Alibaba's Implement:
https://help.aliyun.com/document_detail/130152.html
- **Purpose:** Because DDL statements cannot be rolled back, developers or DBAs may cause data loss if they are mis-operated database(for example, DROP TABLE). Recycle Bin function can transfer the deleted table to the recycle bin temporarily, and setting the retention time so that you can retrieve the data.

Variables For Recycle Bin

- **recycle_bin**: If enable Recycle Bin feature, [SESSION | GLOBAL].
- **recycle_bin_retention**: Recycle bin retention time in seconds. The default is 604800, which is one week.
- **recycle_scheduler**: If enable the asynchronous purge thread for the recycle bin.
- **recycle_scheduler_interval**: The polling interval of the recycle bin purge thread, in seconds. The default is 30.
- **recycle_scheduler_purge_table_print**: If printing the details log of asynchronous cleanup thread works.

Design – Recycling

- Add a system database ‘__recycle_bin__’ for recycle bin.
- When DROP TABLE/DATABASE is executing, we keep only the table-related objects and move to the recycle bin dictionary.
- But If the objects are dependent for the other tables and the table data may be modified by the objects, then delete them, such as Trigger and Foreign key.
- And disable View.

Design – Clean Up

- Recycle Bin can start a background thread to clean up the table objects which are over the time of `recycle_bin_retention`.
- If purging large tables, we can start another project – background purge thread

Design - Management

• `mysql> call dbms_recycle.show_tables();`

SCHEMA	TABLE	ORIGIN_SCHEMA	ORIGIN_TABLE	RECYCLED_TIME	PURGE_TIME
__recycle_bin__	__innodb_1063	product_db	t1	2019-08-08 11:01:46	2019-08-15 11:01:46
__recycle_bin__	__innodb_1064	product_db	t2	2019-08-08 11:01:46	2019-08-15 11:01:46
__recycle_bin__	__innodb_1065	product_db	parent	2019-08-08 11:01:46	2019-08-15 11:01:46
__recycle_bin__	__innodb_1066	product_db	child	2019-08-08 11:01:46	2019-08-15 11:01:46

4 rows in set (0.00 sec)

`mysql> call dbms_recycle.purge_table('__innodb_1063');`

Query OK, 0 rows affected (0.01 sec)

Flashback DDL

- Based on Recycle Bin for DDL
- Flashback can move the deleted table or database to original position.
- For ALTER TABLE, the easier way is always copy original table to Recycle Bin database, but it will waste lot of space.
- Optimization: Add a DDL history table
 - For ADD INDEX / ADD COLUMN, Flashback is DROPPING indexes/columns.
 - For DROP INDEX, Flashback is ADDING index back.
 - For DROP COLUMN, only copy the column values to Recycle Bin.

Purging Large Table Asynchronously

- Adding InnoDB File Purge Thread
 - When DROP TABLE, move the datafile to another name, and filename add into the purge queue.
 - Purge Thread pop a file to call ftruncate().
 - If the file is already small enough, remove file completely and unlink from queue.
 - If the file is still big, then continue to next loop.