until the transaction has been committed (or rolled back). When the client has finished processing the transaction, the variable is cleared. If gtid_next=AUTOMATIC is used for the session, gtid_owned is populated only briefly during the execution of the commit statement for the transaction, so it cannot be observed from the session concerned, although it is listed if @@global.gtid_owned is read at the right point. If you have a requirement to track the GTIDs that are handled by a client in a session, you can enable the session state tracker controlled by the session_track_gtids system variable.

- gtid_purged

System Variable	gtid_purged
Scope	Global
Dynamic	Yes
SET_VAR Hint Applies	No
Type	String
Unit	set of GTIDs

The global value of the gtid_purged system variable (@@GLOBAL.gtid_purged) is a GTID set consisting of the GTIDs of all the transactions that have been committed on the server, but do not exist in any binary log file on the server. gtid_purged is a subset of gtid_executed. The following categories of GTIDs are in gtid_purged:

- GTIDs of replicated transactions that were committed with binary logging disabled on the replica.
- GTIDs of transactions that were written to a binary log file that has now been purged.
- GTIDs that were added explicitly to the set by the statement SET @@GLOBAL.gtid_purged.

When the server starts, the global value of gtid_purged is initialized to a set of GTIDs. For information on how this GTID set is computed, see The gtid_purged System Variable. If binary logs from MySQL 5.7.7 or older are present on the server, you might need to set binlog_gtid_simple_recovery=FALSE in the server's configuration file to produce the correct computation. See the description for binlog_gtid_simple_recovery for details of the situations in which this setting is needed.

Issuing RESET MASTER causes the value of gtid_purged to be reset to an empty string.

You can set the value of gtid_purged in order to record on the server that the transactions in a certain GTID set have been applied, although they do not exist in any binary log on the server. An example use case for this action is when you are restoring a backup of one or more databases on a server, but you do not have the relevant binary logs containing the transactions on the server.

> **Important**
>
> GTIDs are only available on a server instance up to the number of non-negative values for a signed 64-bit integer (2 to the power of 63, minus 1). If you set the value of gtid_purged to a number that approaches this limit, subsequent commits can cause the server to run out of GTIDs and take the action specified by binlog_error_action. From MySQL 8.0.23, a warning message is issued when the server instance is approaching the limit.

From MySQL 8.0, there are two ways to set the value of gtid_purged. You can either replace the value of gtid_purged with your specified GTID set, or you can append your specified GTID set to the GTID set that is already held by gtid_purged. If the server has no existing GTIDs, for example an