The events_transactions_current table has these columns:

- THREAD_ID, EVENT_ID

  The thread associated with the event and the thread current event number when the event starts. The THREAD_ID and EVENT_ID values taken together uniquely identify the row. No two rows have the same pair of values.

- END_EVENT_ID

  This column is set to NULL when the event starts and updated to the thread current event number when the event ends.

- EVENT_NAME

  The name of the instrument from which the event was collected. This is a NAME value from the setup_instruments table. Instrument names may have multiple parts and form a hierarchy, as discussed in Section 27.6, “Performance Schema Instrument Naming Conventions”.

- STATE

  The current transaction state. The value is ACTIVE (after START TRANSACTION or BEGIN), COMMITTED (after COMMIT), or ROLLED BACK (after ROLLBACK).

- TRX_ID

  Unused.

- GTID

  The GTID column contains the value of gtid_next, which can be one of ANONYMOUS, AUTOMATIC, or a GTID using the format UUID:NUMBER. For transactions that use gtid_next=AUTOMATIC, which is all normal client transactions, the GTID column changes when the transaction commits and the actual GTID is assigned. If gtid_mode is either ON or ON_PERMISSIVE, the GTID column changes to the transaction's GTID. If gtid_mode is either OFF or OFF_PERMISSIVE, the GTID column changes to ANONYMOUS.

- XID_FORMAT_ID, XID_GTRID, and XID_BQUAL

  The elements of the XA transaction identifier. They have the format described in Section 13.3.8.1, “XA Transaction SQL Statements”.

- XA_STATE

  The state of the XA transaction. The value is ACTIVE (after XA START), IDLE (after XA END), PREPARED (after XA PREPARE), ROLLED BACK (after XA ROLLBACK), or COMMITTED (after XA COMMIT).

  On a replica, the same XA transaction can appear in the events_transactions_current table with different states on different threads. This is because immediately after the XA transaction is prepared, it is detached from the replica's applier thread, and can be committed or rolled back by any thread on the replica. The events_transactions_current table displays the current status of the most recent monitored transaction event on the thread, and does not update this status when the thread is idle. So the XA transaction can still be displayed in the PREPARED state for the original applier thread, after it has been processed by another thread. To positively identify XA transactions that are still in the PREPARED state and need to be recovered, use the XA RECOVER statement rather than the Performance Schema transaction tables.