

Triggers

The ClustrixDB implementation for triggers is intended to provide compatibility with the MySQL implementation of triggers. For general semantics and operation of triggers refer to the [MySQL documentation for triggers](#).

Caveats for triggers

- ClustrixDB does not perform create-time checks on triggers. If a trigger is defined with access errors (permissions, modifying table the trigger is defined on, etc.), MySQL will generate an error on the CREATE TRIGGER statement, however, ClustrixDB will generate an error when the trigger is executed (i.e. on INSERT/UPDATE/DELETE).
- NOT NULL columns within a BEFORE trigger do not retain the NOT NULL property while within the trigger. In MySQL, a NOT NULL column set to NULL within a trigger will immediately generate an error. In ClustrixDB, however, the column will be set to NULL. Unless changed to NOT NULL later in the trigger, an error will occur when the DML is executed.
 - NOT NULL AUTO_INCREMENT columns are an exception. In MySQL, NOT NULL AUTO_INCREMENT columns set to NULL within a trigger are immediately and silently converted to 0. In ClustrixDB the column will be set to NULL. After a trigger finishes executing, but before the subsequent DML operation, any NOT NULL AUTO_INCREMENT columns with NULL values will be silently converted to 0, allowing auto increment to work correctly.
- ClustrixDB does not support triggers that modify columns with the ENUM data type. No error message will be given when creating or executing triggers that modify ENUM types, but the results are undefined. For example, if a table has a ENUM column state, then the effects of the statement `NEW.state = NEW.state + 1` are undefined. Use explicit values instead, like `NEW.state = "RUNNING"`.
- ClustrixDB supports the TRIGGER permission, but the SUPER permission also grants all abilities associated with the trigger permission.
- Query logs from BEFORE and AFTER triggers will appear before the logs for the statement invoking the trigger.