

```

MariaDB [satish]> CREATE VIEW user_addresses AS
-> SELECT
->     u.id AS user_id,
->     u.username,
->     a.street_address,
->     a.city,
->     a.state,
->     a.postal_code
-> FROM
->     users u
-> JOIN
->     addresses a ON u.id = a.user_id;
Query OK, 0 rows affected (0.012 sec)

```

```

MariaDB [satish]> SELECT * FROM user_addresses;

```

user_id	username	street_address	city	state	postal_code
2	JaneDoe456	Street 2	Lalitpur	Bagmati	44700
3	AliceSmith789	Street 3	Bhaktapur	Bagmati	44800
4	ramg	Street 4	Pokhara	Gandaki	33700
5	Shyam	Street 5	Biratnagar	Koshi	56600
6	Hari	Street 6	Birgunj	Narayani	44300
7	Gita	Street 7	Dharan	Koshi	56700
8	Sita	Street 8	Bharatpur	Narayani	44200
9	Bimal	Street 9	Janakpur	Janakpur	45600
10	Sabina	Street 10	Dhangadhi	Seti	22400
1	JohnDoe123	123 Main Street	Kathmandu	Bagmati	44600

10 rows in set (0.001 sec)

```

MariaDB [satish]> DROP VIEW IF EXISTS user_addresses;
Query OK, 0 rows affected (0.001 sec)

```

```

MariaDB [satish]> CREATE TABLE user_logs (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     user_id INT,
->     action_type VARCHAR(20),
->     action_timestamp DATETIME
-> );
Query OK, 0 rows affected (0.018 sec)

```

```
MariaDB [satish]> DELIMITER //
```

```
MariaDB [satish]> CREATE TRIGGER before_insert_users
```

```
    -> BEFORE INSERT ON users
```

```
    -> FOR EACH ROW
```

```
    -> BEGIN
```

```
    ->     INSERT INTO user_logs (user_id, action_type, action_timestamp)
```

```
    ->     VALUES (NEW.id, 'INSERT', NOW());
```

```
    -> END//
```

Query OK, 0 rows affected (0.040 sec)

```
MariaDB [satish]> DELIMITER ;
```

```
MariaDB [satish]> DELIMITER //
```

```
MariaDB [satish]> CREATE TRIGGER after_insert_users
```

```
    -> AFTER INSERT ON users
```

```
    -> FOR EACH ROW
```

```
    -> BEGIN
```

```
    ->     INSERT INTO user_logs (user_id, action_type, action_timestamp)
```

```
    ->     VALUES (NEW.id, 'AFTER INSERT', NOW());
```

```
    -> END//
```

Query OK, 0 rows affected (0.015 sec)

```
MariaDB [satish]> DELIMITER ;
```

```
MariaDB [satish]> DELIMITER //
```

```
MariaDB [satish]> CREATE TRIGGER before_update_users
```

```
    -> BEFORE UPDATE ON users
```

```
    -> FOR EACH ROW
```

```
    -> BEGIN
```

```
    ->     INSERT INTO user_logs (user_id, action_type, action_timestamp)
```

```
    ->     VALUES (OLD.id, 'UPDATE', NOW());
```

```
    -> END//
```

Query OK, 0 rows affected (0.013 sec)

```
MariaDB [satish]> DELIMITER ;
```

```
MariaDB [satish]> DELIMITER //
```

```
MariaDB [satish]> CREATE TRIGGER after_update_users
```

```
    -> AFTER UPDATE ON users
```

```
    -> FOR EACH ROW
```

```
    -> BEGIN
```

```
    ->     INSERT INTO user_logs (user_id, action_type, action_timestamp)
```

```
    ->     VALUES (OLD.id, 'AFTER UPDATE', NOW());
```

```
    -> END//
```

Query OK, 0 rows affected (0.042 sec)

```
MariaDB [satish]> DELIMITER ;
```

```

MariaDB [satish]> DELIMITER //
MariaDB [satish]> CREATE TRIGGER before_delete_users
    -> BEFORE DELETE ON users
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO user_logs (user_id, action_type, action_timestamp)
    ->     VALUES (OLD.id, 'DELETE', NOW());
    -> END//

```

Query OK, 0 rows affected (0.014 sec)

```

MariaDB [satish]> DELIMITER ;

```

```

MariaDB [satish]> DELIMITER //
MariaDB [satish]> CREATE TRIGGER after_delete_users
    -> AFTER DELETE ON users
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO user_logs (user_id, action_type, action_timestamp)
    ->     VALUES (OLD.id, 'AFTER DELETE', NOW());
    -> END//

```

Query OK, 0 rows affected (0.014 sec)

```

MariaDB [satish]> DELIMITER ;

```

```

MariaDB [satish]> SHOW TRIGGERS;

```

Trigger	Event	Table	Statement
before_insert_users	INSERT	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (NEW.id, 'INSERT', NOW()); END
after_insert_users	INSERT	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (NEW.id, 'AFTER INSERT', NOW()); END
before_update_users	UPDATE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'UPDATE', NOW()); END
after_update_users	UPDATE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'AFTER UPDATE', NOW()); END
before_delete_users	DELETE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'DELETE', NOW()); END
after_delete_users	DELETE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'AFTER DELETE', NOW()); END

6 rows in set (0.002 sec)

```
MariaDB [satish]> SHOW TRIGGERS LIKE 'users';
```

Trigger	Event	Table	Statement
before_insert_users	INSERT	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (NEW.id, 'INSERT', NOW()); END
after_insert_users	INSERT	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (NEW.id, 'AFTER INSERT', NOW()); END
before_update_users	UPDATE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'UPDATE', NOW()); END
after_update_users	UPDATE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'AFTER UPDATE', NOW()); END
before_delete_users	DELETE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'DELETE', NOW()); END
after_delete_users	DELETE	users	BEGIN INSERT INTO user_logs (user_id, action_type, action_timestamp) VALUES (OLD.id, 'AFTER DELETE', NOW()); END

6 rows in set (0.002 sec)

```
MariaDB [satish]> INSERT INTO users (username, email, age)
-> VALUES (
-> 'Satish_Chaudhary',
-> 'sung20700@gmail.com',
-> 21
-> );
```

Query OK, 1 row affected (0.012 sec)

```
MariaDB [satish]> UPDATE users
-> SET age = 22
-> WHERE email = 'sung20700@gmail.com';
```

Query OK, 1 row affected (0.012 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
MariaDB [satish]> DELETE FROM users
-> WHERE email = 'sung20700@gmail.com';
```

Query OK, 1 row affected (0.012 sec)

```
MariaDB [satish]> SELECT * FROM user_logs;
```

id	user_id	action_type	action_timestamp
1	0	INSERT	2024-03-14 19:44:35
2	20	AFTER INSERT	2024-03-14 19:44:35
3	20	UPDATE	2024-03-14 19:44:44
4	20	AFTER UPDATE	2024-03-14 19:44:44
5	20	DELETE	2024-03-14 19:44:54
6	20	AFTER DELETE	2024-03-14 19:44:54

6 rows in set (0.000 sec)