NAME : ANUHYA BHEEMIREDDY

// Execute these triggers on the database we created for phase 2.

**Trigger : 1** The trigger inserts the old event location in a new table before updating the event location on the event table.

Create a new table using the below query.

**CREATE TABLE event\_loc\_change\_info (**

**ID int NOT NULL AUTO\_INCREMENT,**

**Event\_id int NOT NULL,**

**old\_location varchar(255),**

**changedat DATETIME DEFAULT NULL,**

**PRIMARY KEY (ID)**

**)**

Then create the trigger using below query.

**Delimiter$$**

[**CREATE**](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-trigger.html)[**TRIGGER**](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-trigger.html)**before\_event\_update**

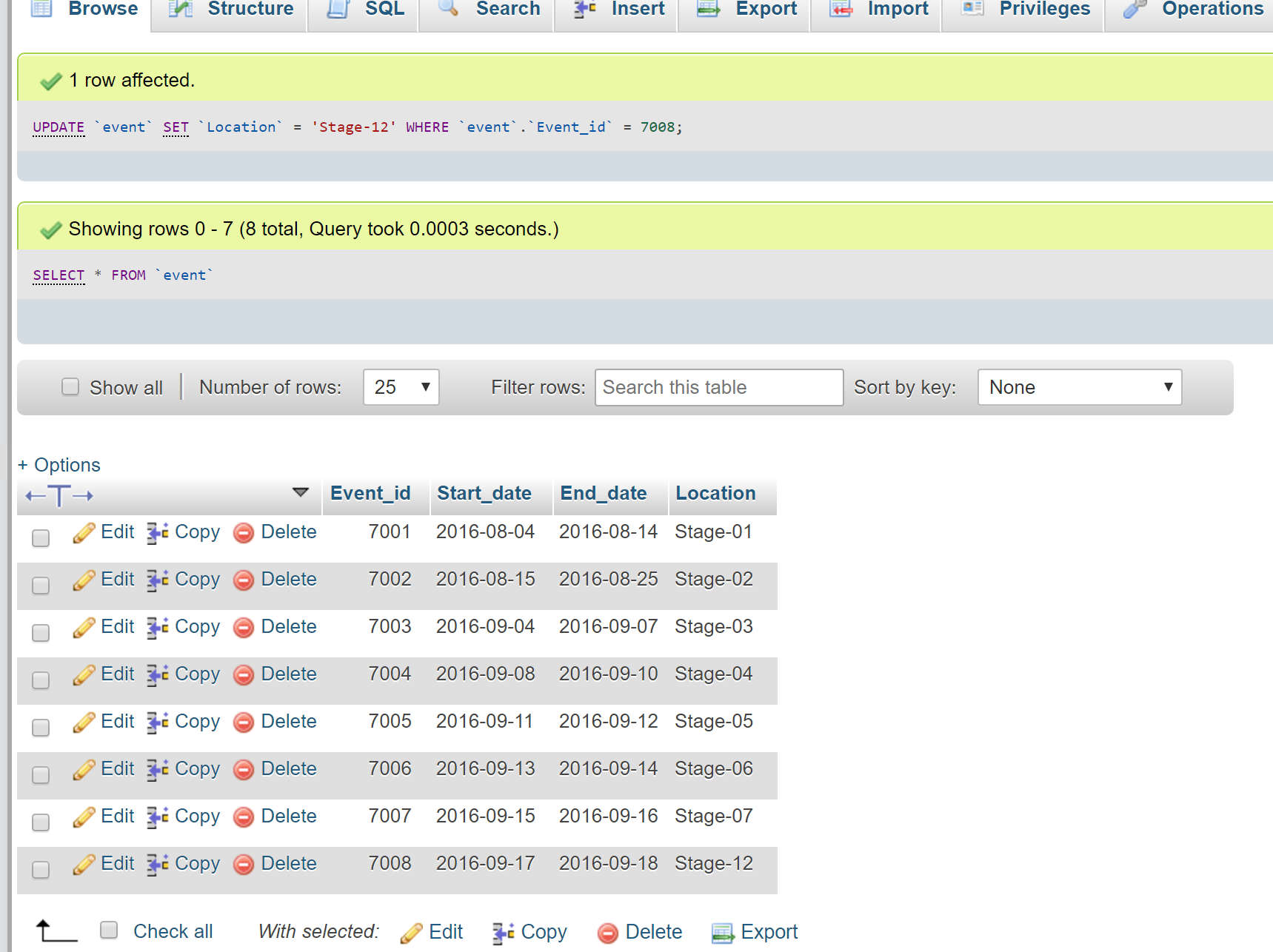
**BEFORE**[**UPDATE**](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html)**ON event**

**For each ROW**

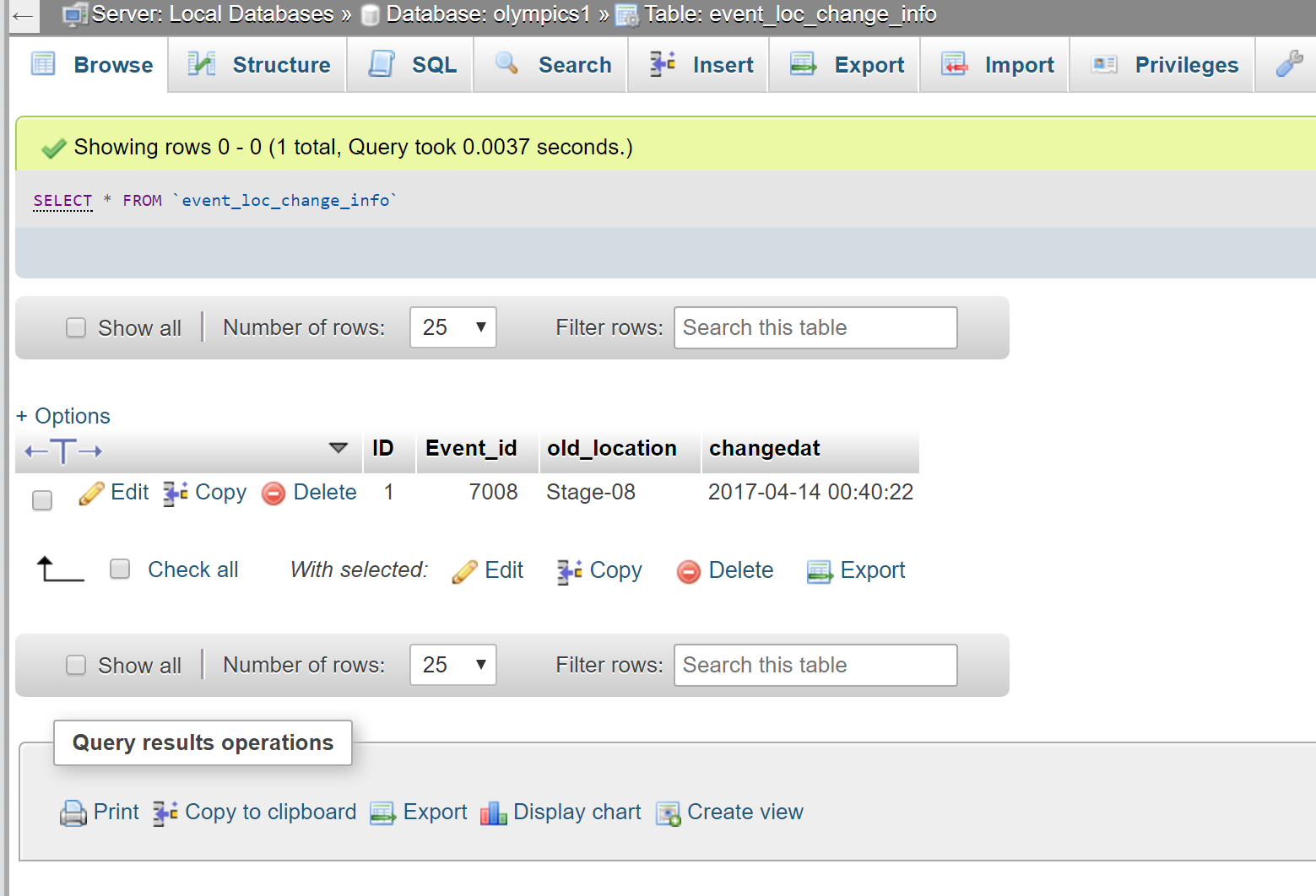
**BEGIN**[**insert**](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html)**INTO event\_loc\_change\_info**[**set**](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html)**Event\_id = OLD.Event\_id, old\_location = OLD.Location, changedat = NOW();**

**END $$**

**This is screen short when you update the location in event table .**



**This image is the execution of trigger i.e. whenever you update the event table location, event\_loc\_change\_info table is going to store the old location of the updated event\_id.**



**Trigger : 2**

This trigger is a delete trigger. When you try to delete event\_id in event table , it first deletes that event id in participate, event\_official and event\_loc\_change\_info tables and then deletes it from event table. This trigger avoids foreign key constraints while deleting an event\_id from event table.

**DELIMITER $$**

**CREATE TRIGGER before\_event\_delete**

**BEFORE DELETE ON event**

**For each ROW**

**BEGIN**

**DELETE FROM event\_loc\_change\_info**

**WHERE Event\_id = OLD.Event\_id;**

**DELETE FROM event\_official WHERE Event\_id = OLD.Event\_id;**

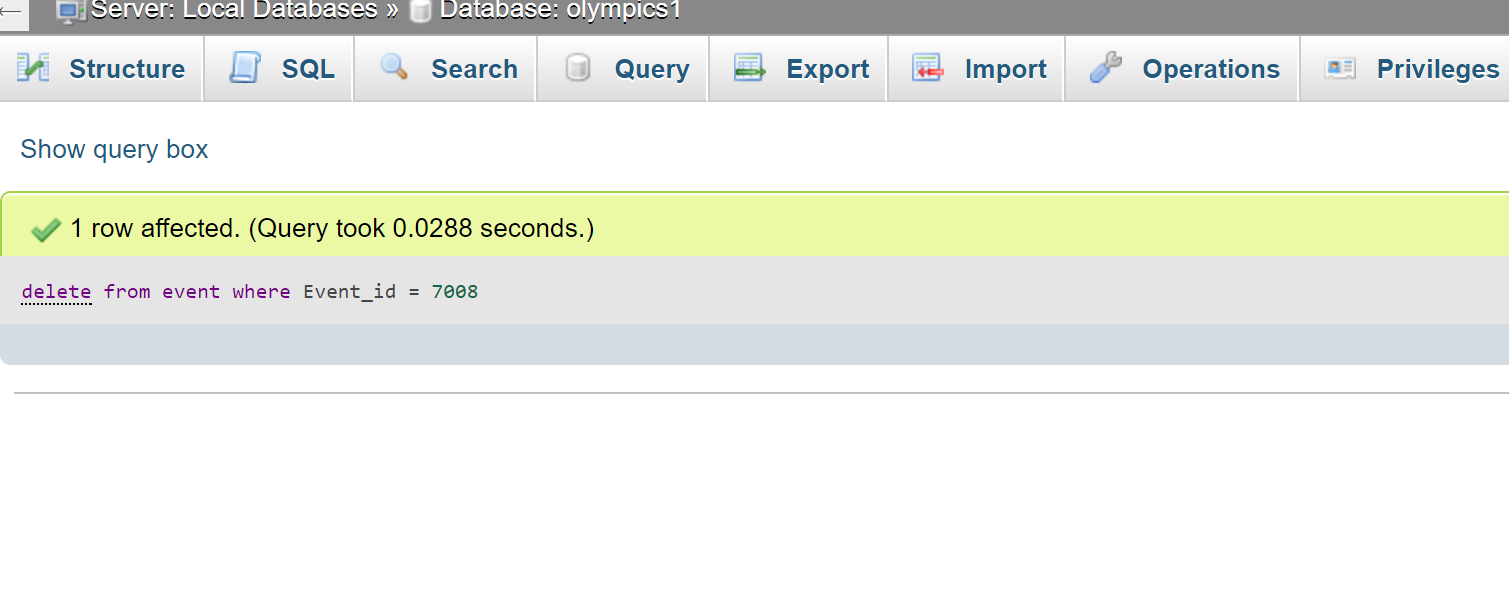
**DELETE FROM participate WHERE Event\_id = OLD.Event\_id;**

**END$$**

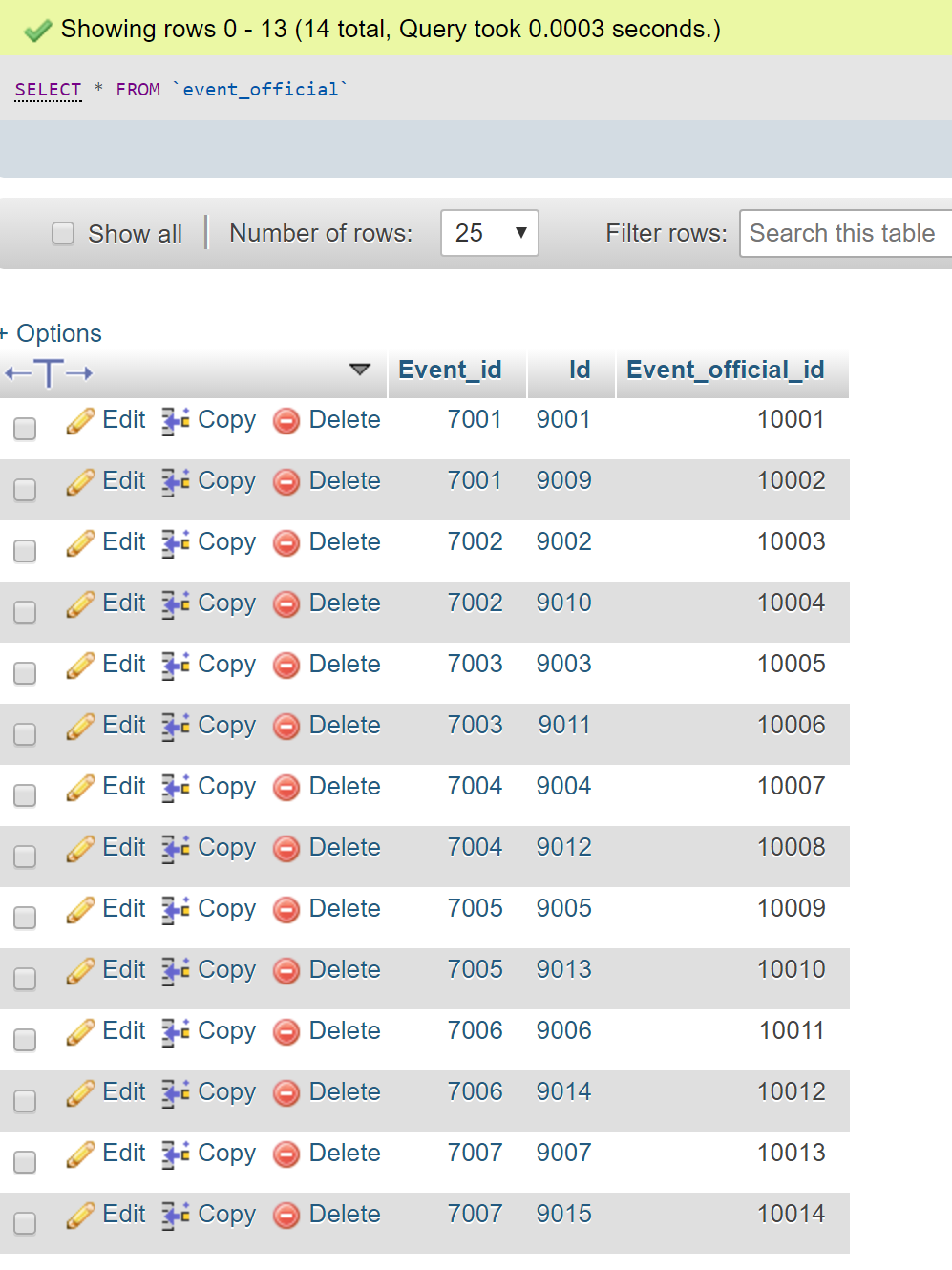
When you execute the query:

Delete from event where Event\_id =7008;

It deletes rows that contain Event\_id = 7008 from participate, event\_official and event\_loc\_change\_info tables and then deletes the Event\_id from event table to avoid foreign key constraints.



**The effected tables after the delete query is executed and you can see that none of the tables have event\_id = 7008.**

y:

