

1. Create a table named teachers with fields id, name, subject, experience and salary and insert 8 rows.

2. CREATE TABLE TEACHERS (
3. TEACHER_ID INT AUTO_INCREMENT PRIMARY KEY,
4. NAME VARCHAR(50),
5. SUBJECT VARCHAR(30),
6. EXPERIENCE INT,
7. SALARY DECIMAL(10, 2)
8.);

```
2 • use entri_d41;
3 • CREATE TABLE teachers (
4   teacher_id INT AUTO_INCREMENT PRIMARY KEY,
5   name VARCHAR(50),
6   subject VARCHAR(30),
7   experience int,
8   salary DECIMAL(10, 2)
9 );
```

Output

Action Output

#	Time	Action	Message
✓ 1	11:25:26	use entri_d41	0 row(s) affected
✓ 2	11:25:32	CREATE TABLE teachers (teacher_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(5...	0 row(s) affected

INSERT INTO TEACHERS (TEACHER_ID, NAME, SUBJECT, EXPERIENCE,SALARY) VALUES
(1, 'FAZIL', 'MATHAMATICS', 12,50000);

```
13 • INSERT INTO teachers (teacher_id, name, subject, experience,salary) VALUES
14 (1, 'Fazil', 'Mathamatics', 12,50000);
15 • select * from teachers;
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit: Export/Import: Wrap Cell Content:					
	teacher_id	name	subject	experience	salary
▶	1	Fazil	Mathamatics	12	50000.00
*	NULL	NULL	NULL	NULL	NULL

teachers 1 x				
Output				
Action Output				
#	Time	Action	Message	
✓ 1	11:25:26	use entri_d41	0 row(s) affected	
✓ 2	11:25:32	CREATE TABLE teachers (teacher_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(...	0 row(s) affected	
✓ 3	11:30:57	INSERT INTO teachers (teacher_id, name, subject, experience,salary) VALUES (1, 'Fazil', 'Mathamatic...	1 row(s) affected	

INSERT INTO TEACHERS (NAME, SUBJECT, EXPERIENCE,SALARY) VALUES
('SHAHID', 'PHYSICS', 10,45000),('KIRAN', 'CHEMISTRY', 15,65000),
('JONAH', 'BIOLOGY', 8,47000),('TESSA', 'HINDI', 11,48500),
('MARIA', 'MALAYALAM', 15,55000),('SURESH', 'ENGLISH', 20,75000),
('SUPRIYA', 'HISTORY', 25,88500);

```
17 • INSERT INTO teachers ( name, subject, experience,salary) VALUES
18 ( 'Shahid', 'physics', 10,45000),( 'Kiran', 'Chemistry', 15,65000),
19 ( 'Jonah', 'Biology', 8,47000),( 'Tessa', 'Hindi', 11,48500),
20 ( 'Maria', 'Malayalam', 15,55000),( 'Suresh', 'English', 20,75000),
21 ( 'Supriya', 'History', 25,88500);
22
23
24
25
26
27
```

Output				
Action Output				
#	Time	Action	Message	
✓ 1	11:25:26	use entri_d41	0 row(s) affected	
✓ 2	11:25:32	CREATE TABLE teachers (teacher_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(...	0 row(s) affected	
✓ 3	11:30:57	INSERT INTO teachers (teacher_id, name, subject, experience,salary) VALUES (1, 'Fazil', 'Mathamatic...	1 row(s) affected	
✗ 4	11:31:15	select * from teachers (2, 'Jane Smith', 'Finance', 65000, '2020-08-15'), (3, 'Mike Brown', 'IT', 7500...	Error Code: 1064. You have an error in your SQL syntax; c	
✓ 5	11:31:28	select * from teachers LIMIT 0, 50000	1 row(s) returned	
✓ 6	11:42:31	INSERT INTO teachers (name, subject, experience,salary) VALUES ('Shahid', 'physics', 10,45000),('...	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0	

2. Create a before insert trigger named before_insert_teacher that will raise an error “salary cannot be negative” if the salary inserted to the table is less than zero.

```
DELIMITER $$
CREATE TRIGGER BEFORE_INSERT_TEACHERS
BEFORE INSERT ON TEACHERS
FOR EACH ROW
BEGIN
IF NEW.SALARY<0
THEN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT= "SALARY CANNOT BE NEGATIVE";
END IF;
END $$
DELIMITER ;
```

```
23 DELIMITER $$
24 • CREATE TRIGGER before_insert_teachers
25 BEFORE INSERT ON teachers
26 FOR EACH ROW
27 BEGIN
28 if new.salary<0
29 then
30 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT= "salary cannot be negative";
31 end if;
32 END $$
33 DELIMITER ;
34
35 • INSERT INTO teachers ( name, subject, experience,salary) VALUES
36 ('Menon', 'History', 25,-2000);
```

Output

Action Output

#	Time	Action	Message
✓ 1	12:08:29	CREATE TRIGGER before_insert_teachers BEFORE INSERT ON teachers FOR EACH ROW BEGIN...	0 row(s) affected
✗ 2	12:10:51	INSERT INTO teachers (name, subject, experience,salary) VALUES ('Menon', 'History', 25,-2000)	Error Code: 1644. salary cannot be negative

3. Create an after insert trigger named after_insert_teacher that inserts a row with teacher_id, action, timestamp to a table called teacher_log when a new entry gets inserted to the teacher table. teacher_id -> column of teacher table, action -> the trigger action, timestamp -> time at which the new row has got inserted.

```
DELIMITER $$
CREATE TRIGGER AFTER_TEACHER_INSERT
AFTER INSERT ON TEACHERS
FOR EACH ROW
BEGIN
INSERT INTO TEACHERS_LOG(LOG_ID,TEACHER_ID,ACTION)
VALUES(LOG_ID,NEW.TEACHER_ID,'NEW TEACHER ENTRY');
END $$
DELIMITER ;
```

```
47 DELIMITER $$
48 • CREATE TRIGGER after_teacher_insert
49   after insert ON teachers
50   FOR EACH ROW
51   BEGIN
52     INSERT into teachers_log(log_id,teacher_id,action)
53     values(log_id,NEW.teacher_id,'NEW Teacher entry');
54   END $$
55   DELIMITER ;
56
57 • INSERT INTO teachers (name, subject, experience,salary) VALUES
58   ('mary','Computer',10,48500);
```

Output :

Action Output

	#	Time	Action	Message
✓	1	22:47:13	CREATE TRIGGER after_teacher_insert after insert ON teachers FOR EACH ROW BEGIN INSERT ...	0 row(s) affected
✓	2	22:47:28	INSERT INTO teachers (name, subject, experience,salary) VALUES ('mary','Computer',10,48500)	1 row(s) affected

```
CREATE TABLE TEACHERS_LOG (
LOG_ID INT AUTO_INCREMENT PRIMARY KEY,
TEACHER_ID INT,
ACTION VARCHAR(70),
ACTION_TIME TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
40 • CREATE TABLE teachers_log (
41   log_id INT AUTO_INCREMENT PRIMARY KEY,
42   teacher_id INT,
43   action VARCHAR(70),
44   action_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
45 );
46
```

Output

Action Output

#	Time	Action	Message
✓ 1	22:51:35	CREATE TABLE teachers_log (log_id INT AUTO_INCREMENT PRIMARY KEY, teacher_id INT, act...	0 row(s) affected

```
SELECT * FROM TEACHERS_LOG;
```

```
67 • select * from teachers_log;
```

log_id	teacher_id	action	action_time
1	18	NEW Teacher entry	2024-11-15 22:45:48
2	19	NEW Teacher entry	2024-11-15 22:47:28
*	NULL	NULL	NULL

teachers_log 4 x

Output

Action Output

#	Time	Action	Message
✓ 1	22:47:13	CREATE TRIGGER after_teacher_insert after insert ON teachers FOR EACH ROW BEGIN INSERT ...	0 row(s) affected
✓ 2	22:47:28	INSERT INTO teachers (name, subject, experience,salary) VALUES ('mary','Computer',10,48500)	1 row(s) affected
✓ 3	22:47:35	select *from teachers_log LIMIT 0, 50000	2 row(s) returned

4. Create a before delete trigger that will raise an error when you try to delete a row that has experience greater than 10 years.

```
DELIMITER $$
CREATE TRIGGER BEFORE_TEACHER_DELETE
BEFORE DELETE ON TEACHERS
FOR EACH ROW
BEGIN
IF OLD.EXPERIENCE>10
THEN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT= 'YOU DONT HAVE THE RIGHTS TO DELETE
THIS TEACHER';
END IF;
END $$
DELIMITER ;
```

```
70 DELIMITER $$
71 • CREATE TRIGGER before_teacher_delete
72 before delete ON teachers
73 FOR EACH ROW
74 BEGIN
75 if old.experience>10
76 then
77 SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT= 'you dont have the rights to delete this teacher';
78 end if;
79 END $$
80 DELIMITER ;
81
82 • delete from teachers where name ='maria';
```






Output

Action Output

#	Time	Action	Message
✓ 1	23:20:47	CREATE TRIGGER before_teacher_delete before delete ON teachers FOR EACH ROW BEGIN if ol...	0 row(s) affected
✓ 2	23:20:58	select * from teachers LIMIT 0, 50000	11 row(s) returned
✗ 3	23:21:30	delete from teachers where name ='maria'	Error Code: 1644, you dont have the rights to delete this teacher

SELECT * FROM TEACHERS;

62 • `select * from teachers;`

Result Grid   Filter Rows: Edit:   

	teacher_id	name	subject	experience	salary
▶	1	Fazil	Mathamatics	12	50000.00
	2	Shahid	physics	10	45000.00
	3	Kiran	Chemistry	15	65000.00
	4	Jonah	Biology	8	47000.00
	5	Tessa	Hindi	11	48500.00
	6	Maria	Malayalam	15	55000.00
	8	Supriya	History	25	88500.00
	16	jonah	physics	10	46000.00
	17	Roja	Computer	10	48500.00

5. Create an after delete trigger that will insert a row to teacher_log table when that row is deleted from teacher table.

```
DELIMITER $$
```

```
CREATE TRIGGER AFTER_TEACHER_DELETE
```

```
AFTER DELETE ON TEACHERS
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO TEACHERS_LOG(LOG_ID,ACTION) VALUES(OLD.TEACHER_ID,'RETIRED  
TEACHER');
```

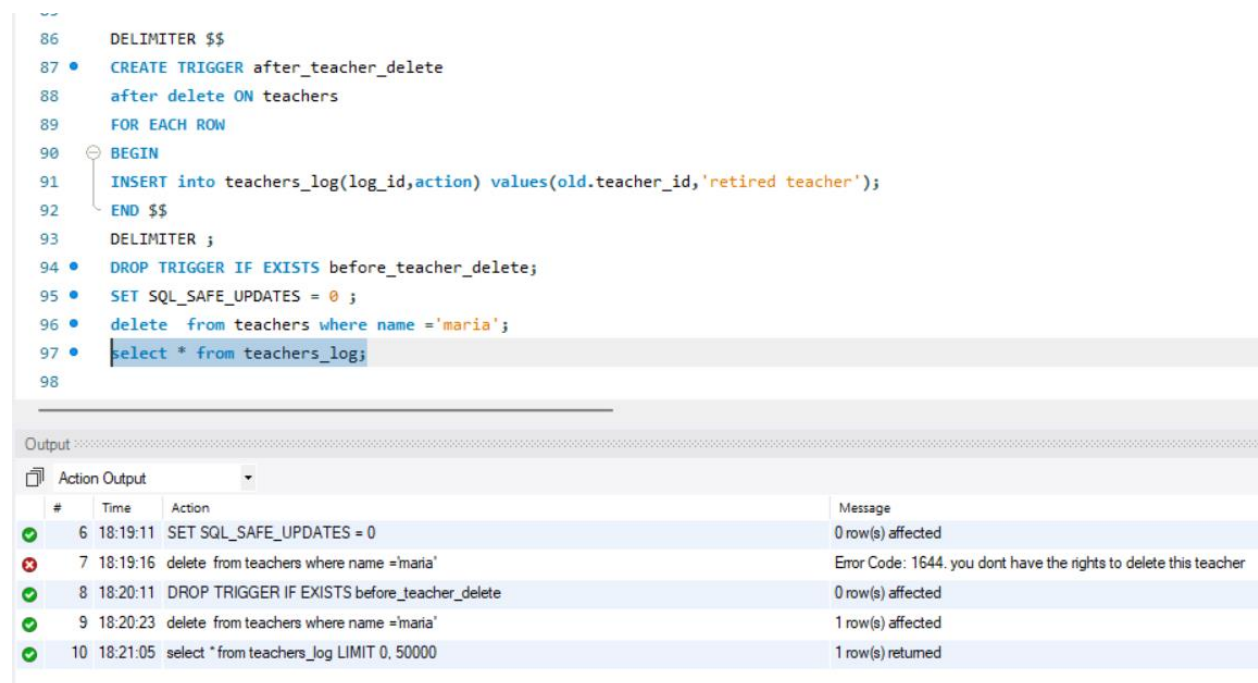
```
END $$
```

```
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS BEFORE_TEACHER_DELETE;
```

```
SET SQL_SAFE_UPDATES = 0 ;
```

```
DELETE FROM TEACHERS WHERE NAME ='MARIA';
```



The screenshot displays a SQL script in a text editor and its corresponding execution results in an 'Output' window. The script includes creating an after-delete trigger, dropping an existing trigger, disabling safe updates, deleting a row from the 'teachers' table, and querying the 'teachers_log' table. The output window shows the execution of these steps with status icons (green checkmarks for success, red X for error) and messages indicating the number of rows affected or returned.

```
86 DELIMITER $$
87 CREATE TRIGGER after_teacher_delete
88 after delete ON teachers
89 FOR EACH ROW
90 BEGIN
91 INSERT into teachers_log(log_id,action) values(old.teacher_id,'retired teacher');
92 END $$
93 DELIMITER ;
94 DROP TRIGGER IF EXISTS before_teacher_delete;
95 SET SQL_SAFE_UPDATES = 0 ;
96 delete from teachers where name ='maria';
97 select * from teachers_log;
98
```

Output




#	Time	Action	Message
6	18:19:11	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
7	18:19:16	delete from teachers where name ='maria'	Error Code: 1644. you dont have the rights to delete this teacher
8	18:20:11	DROP TRIGGER IF EXISTS before_teacher_delete	0 row(s) affected
9	18:20:23	delete from teachers where name ='maria'	1 row(s) affected
10	18:21:05	select * from teachers_log LIMIT 0, 50000	1 row(s) returned



SELECT * FROM TEACHERS_LOG;


97 • `select * from teachers_log;`

Result Grid

Filter Rows:

Edit:   

Export/Import:  

Wrap Cell Content: 

	log_id	teacher_id	action	action_time
▶	6	NULL	retired teacher	2024-11-16 18:20:23
*	NULL	NULL	NULL	NULL

teachers_log 3 x

Apply

Output

Action Output

#	Time	Action	Message
✓	6 18:19:11	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
✗	7 18:19:16	delete from teachers where name ='maria'	Error Code: 1644, you dont have the rights to delete this teacher
✓	8 18:20:11	DROP TRIGGER IF EXISTS before_teacher_delete	0 row(s) affected
✓	9 18:20:23	delete from teachers where name ='maria'	1 row(s) affected
✓	10 18:21:05	select * from teachers_log LIMIT 0, 50000	1 row(s) returned