

**Institute of Computer Technology**  
**B. Tech. Computer Science and Engineering**

**Semester: III**

**Sub: Database Management System**

**Course Code: 2CSE301**

## Practical Number:10

### Objective:

MySQL Triggers

### Queries:

1. Create a trigger that converts each newly inserted language name into uppercase
2. Create a trigger that displays a message “The record has been inserted successfully” whenever a record is inserted in the language table.

#### Code :

```
DELIMITER //
CREATE TRIGGER Upper_name
BEFORE INSERT ON language
FOR EACH ROW
BEGIN
    set new.name = upper(new.name);
END
//
DELIMITER ;

INSERT INTO language VALUES
(9,'Korean',CURRENT_TIMESTAMP);

SELECT * FROM language;
```

#### Output :

	language_id	name	last_update
▶	1	English	2006-02-15 05:02:19
	2	Italian	2006-02-15 05:02:19
	3	Japanese	2006-02-15 05:02:19
	4	Mandarin	2006-02-15 05:02:19
	5	French	2006-02-15 05:02:19
	6	German	2006-02-15 05:02:19
	7	Spanish	2024-11-18 20:26:00
	8	Hindi	2020-07-13 10:00:19
	9	gujarati	2024-12-05 13:49:49
•	NULL	NULL	NULL

3. Create a new table audit\_language\_update with the following schema.

For audit purpose, Create a trigger that inserts existing (old) record as well as updated record into above table whenever a record gets updated in 'language' table

Code :

```
CREATE TABLE audit_language_update (  
    language_id TINYINT(3),  
    name CHAR(20),  
    last_update TIMESTAMP,  
    log CHAR(20)  
);  
  
DELIMITER //  
  
CREATE TRIGGER LANUPDATE  
BEFORE UPDATE ON language  
FOR EACH ROW  
BEGIN  
    INSERT INTO audit_language_update VALUES  
    (old.language_id,old.name,old.last_update,"Before Update");  
    INSERT INTO audit_language_update VALUES  
    (new.language_id,new.name,new.last_update,"After Update");  
END
```

```
//  
DELIMITER ;  
  
UPDATE language  
SET name = 'gujarati'  
WHERE language_id = 09;  
  
SELECT * FROM audit_language_update;
```

**Output :**

	language_id	name	last_update	log
▶	9	Hindi	2020-07-13 10:00:19	Before Update
	9	gujarati	2024-12-05 13:49:49	After Update
	9	gujarati	2024-12-05 13:49:49	Before Update
	9	gujarati	2024-12-05 13:49:49	After Update

4. Create a new table `language_before_delete` with the following schema. Create a trigger that keeps backup of deleted records into above table whenever a record is deleted in 'language' table

**Code :**

```
CREATE TABLE language_before_delete (  
    language_id TINYINT(3),  
    name CHAR(20),  
    last_update TIMESTAMP,  
    log CHAR(20)  
);
```

```
DELIMITER //  
  
CREATE TRIGGER LANBEFDEL  
BEFORE DELETE ON language  
FOR EACH ROW  
BEGIN
```

```
INSERT INTO language_before_delete VALUES
(old.language_id,old.name,old.last_update,"Before Delete");

END

//

DELIMITER ;

DELETE FROM language

WHERE language_id = 9;

SELECT * FROM language_before_delete;
```

**Output :**

	language_id	name	last_update	log
▶	9	gujarati	2024-12-05 13:49:49	Before Delete

5. Create a new table `language_after_delete` with the following schema. Create a trigger that maintains the status of deleted records into above table whenever a record is deleted in 'language' table. Status should be displayed as: "Language [language\_name] with ID [language\_id] was deleted on [timestamp when the record was deleted]"

**Code :**

```
CREATE TABLE language_after_delete (
    language_id TINYINT(3),
    name CHAR(20),
    log CHAR(200)
);

DELIMITER //
CREATE TRIGGER LANAFTDEL
BEFORE DELETE ON language
FOR EACH ROW
BEGIN
    INSERT INTO language_after_delete VALUES (
        old.language_id,old.name,
        CONCAT(
            "LANGUAGE ",old.name,
            " WITH ID ",old.language_id,
            " WAS DELETED ON ",CURRENT_TIMESTAMP
        )
    );
END
```

```
//
DELIMITER ;

DELETE FROM language
WHERE language_id = 5;

SELECT * FROM language_after_delete;
```

**Output :**

	language_id	name	log
►	5	French	LANGUAGE French WITH ID 5 WAS DELETED O...

## 6. Display all the triggers in your database.

**Code :**

```
SHOW TRIGGERS;
```

**Output :**

	Trigger	Event	Table	Statement	Timing	Created	sql_mode
►	ins_film	INSERT	film	BEGIN INSERT INTO film_text (film_id, title, d...	AFTER	2024-11-18 11:05:45.47	STRICT_TRANS_TABLES,STRICT_ALL_T
	upd_film	UPDATE	film	BEGIN IF (old.title != new.title) or (old.descri...	AFTER	2024-11-18 11:05:45.54	STRICT_TRANS_TABLES,STRICT_ALL_T
	del_film	DELETE	film	BEGIN DELETE FROM film_text WHERE film_id...	AFTER	2024-11-18 11:05:45.57	STRICT_TRANS_TABLES,STRICT_ALL_T
	Upper_name	INSERT	language	BEGIN set new.name = upper(new.name); END	BEFORE	2024-12-05 13:48:47.32	ONLY_FULL_GROUP_BY,STRICT_TRANS
	LANUPDATE	UPDATE	language	BEGIN INSERT INTO audit_language_update VA...	BEFORE	2024-12-05 13:49:48.98	ONLY_FULL_GROUP_BY,STRICT_TRANS
	LANBEFDEL	DELETE	language	BEGIN INSERT INTO language_before_delete V...	BEFORE	2024-12-05 13:55:52.37	ONLY_FULL_GROUP_BY,STRICT_TRANS
	LANAFTDEL	DELETE	language	BEGIN INSERT INTO language_after_delete VA...	BEFORE	2024-12-05 13:56:41.16	ONLY_FULL_GROUP_BY,STRICT_TRANS