Name : Ayush Patel   Class B Batch 35   Enrolment No : 22162171038

Institute of Computer Technology

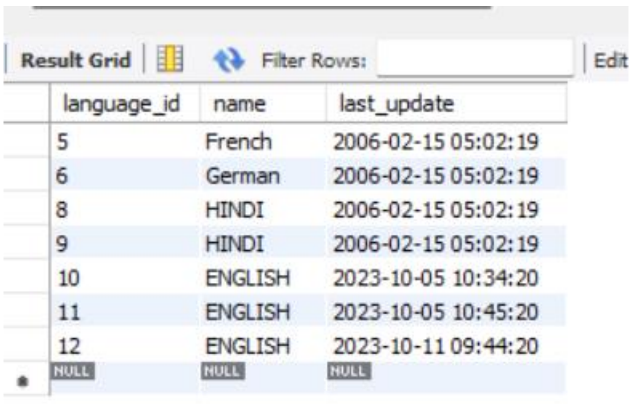
Ganpat University

(2CSE301) DATABASE MANAGEMENT SYSTEM

Practical 11 MySQL Triggers

1. Create a trigger that converts each newly inserted language name into uppercase

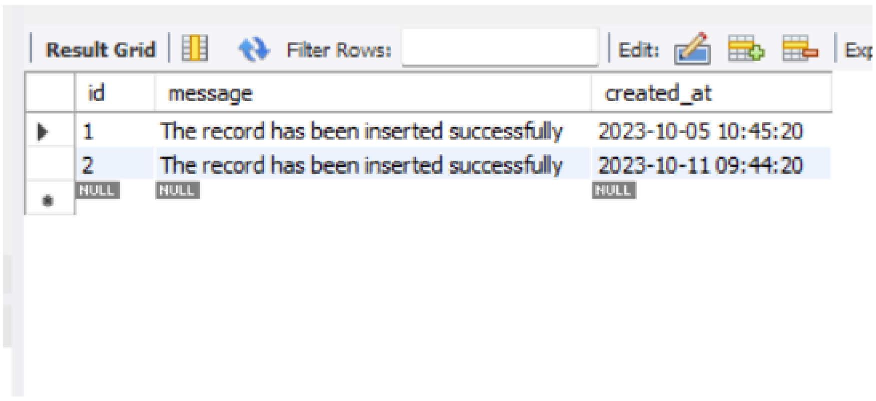
```
/*1*/

DELIMITER //
CREATE TRIGGER uppercase
BEFORE INSERT ON language
FOR EACH ROW
BEGIN
  SET NEW.name = UPPER(NEW.name);
END;
//
DELIMITER ;

INSERT INTO language (name) VALUES ('english');
select *from language;
-- show triggers like 'uppercase';
```



language_id	name	last_update
5	French	2006-02-15 05:02:19
6	German	2006-02-15 05:02:19
8	HINDI	2006-02-15 05:02:19
9	HINDI	2006-02-15 05:02:19
10	ENGLISH	2023-10-05 10:34:20
11	ENGLISH	2023-10-05 10:45:20
12	ENGLISH	2023-10-11 09:44:20
NULL	NULL	NULL

2. Create a trigger that displays a message **“The record has been inserted successfully”** whenever a record is inserted in the language table.

```
/*2*/

CREATE TABLE insertion_messages (
  id INT AUTO_INCREMENT PRIMARY KEY,
   message VARCHAR(255),
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

DELIMITER //
CREATE TRIGGER display
AFTER INSERT ON language
FOR EACH ROW
BEGIN
   INSERT INTO insertion_messages (message) VALUES ('The record has been
inserted successfully');
END;
//
DELIMITER ;

INSERT INTO language (name) VALUES ('english');
SELECT * FROM insertion_messages;
```



```
id    message                                    created_at
1     The record has been inserted successfully  2023-10-05 10:45:20
2     The record has been inserted successfully  2023-10-11 09:44:20
NULL  NULL                                       NULL
```

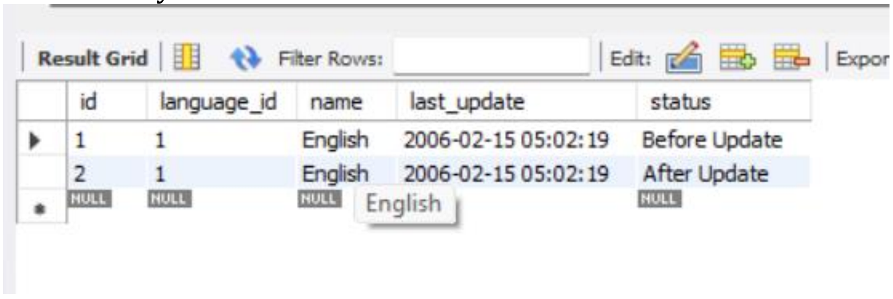
3. Create a new table **audit_language_update** with the following schema.

Column	Datatype and constraint
language_id	TINYINT(3)
name	CHAR(20)
last_update	TIMESTAMP

Status	CHAR(20) Should accept only “Before Update” and “After Update”

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

```
CREATE TABLE audit_language_update (
   id INT AUTO_INCREMENT PRIMARY KEY,
   language_id TINYINT(3),
   name CHAR(20),
   last_update TIMESTAMP,
   status CHAR(20)
);



DELIMITER //
CREATE TRIGGER language_audit_trigger
AFTER UPDATE ON language
FOR EACH ROW
BEGIN
   -- Insert old record (BEFORE UPDATE) into audit table
   INSERT INTO audit_language_update (language_id, name, last_update, status)
   VALUES (OLD.language_id, OLD.name, OLD.last_update, 'Before Update');

   -- Insert updated record (AFTER UPDATE) into audit table
   INSERT INTO audit_language_update (language_id, name, last_update, status)
   VALUES (NEW.language_id, NEW.name, NEW.last_update, 'After Update');
END;
//
DELIMITER ;

UPDATE language
SET name = 'English'
WHERE language_id = 1;

SELECT * FROM audit_language_update;
```



id	language_id	name	last_update	status
1	1	English	2006-02-15 05:02:19	Before Update
2	1	English	2006-02-15 05:02:19	After Update
NULL	NULL	NULL		NULL

4. Create a new table **language_before_delete** with the following schema.

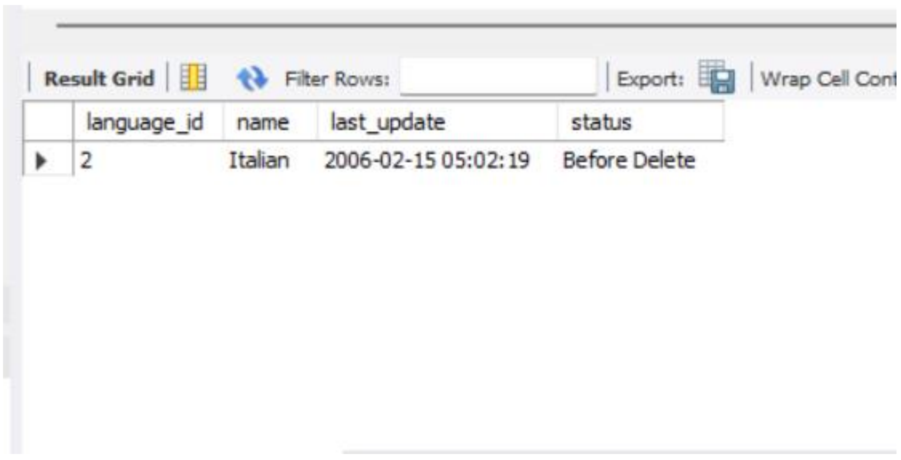
Column	Datatype and constraint
language_id	TINYINT(3)
name	CHAR(20)
last_update	TIMESTAMP
Status	CHAR(20) **Default:** “Before Delete”

Create a trigger that keeps backup of deleted records into above table whenever a record is deleted in ‘language’ table

```
CREATE TABLE language_before_delete (
   language_id TINYINT(3),
   name CHAR(20),
   last_update TIMESTAMP,
   status CHAR(20) DEFAULT 'Before Delete'
);


DELIMITER //
CREATE TRIGGER language_delete_trigger
AFTER DELETE ON language
FOR EACH ROW
BEGIN
   -- Insert deleted record into backup table
   INSERT INTO language_before_delete (language_id, name, last_update)
   VALUES (OLD.language_id, OLD.name, OLD.last_update);
END;
//
DELIMITER ;
```

DELETE FROM language WHERE language_id = 2;

SELECT * FROM language_before_delete;



```
language_id  name     last_update          status
2            Italian  2006-02-15 05:02:19  Before Delete
```

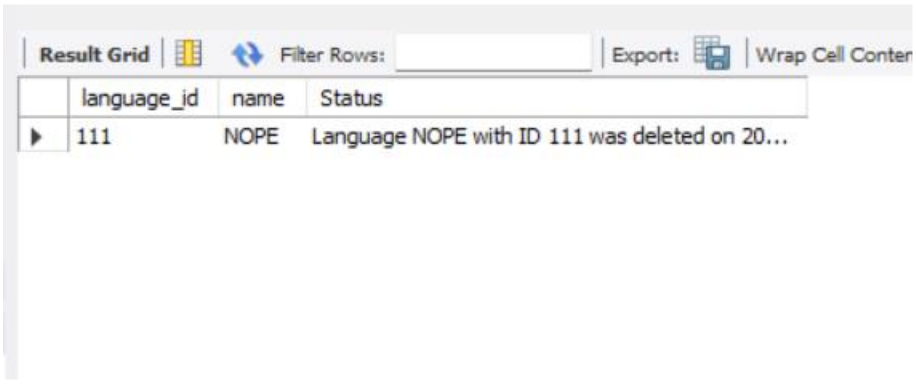
5. Create a new table **language_after_delete** with the following schema.

Column	**Datatype and constraint**
language_id	TINYINT(3)
name	CHAR(20)
Status	VARCHAR(200)

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]***

```
-- Create the language_after_delete table
CREATE TABLE language_after_delete (
   language_id TINYINT(3),
   name CHAR(20),
   Status VARCHAR(200)
);

-- Create the trigger to maintain deleted records in language_after_delete
table
DELIMITER //
```

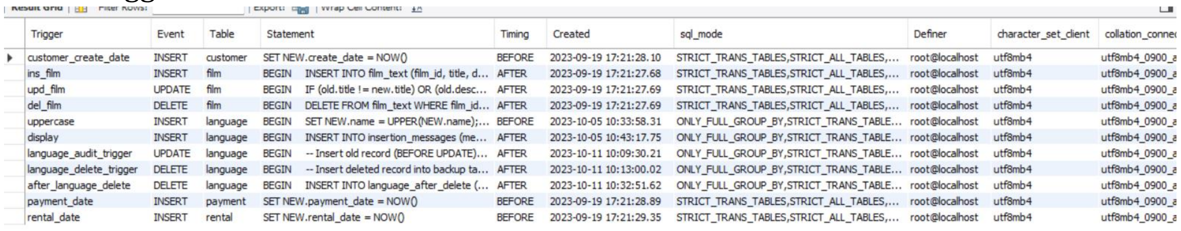
```
CREATE TRIGGER after_language_delete
AFTER DELETE ON language
FOR EACH ROW
BEGIN
   INSERT INTO language_after_delete (language_id, name, Status)
   VALUES (OLD.language_id, OLD.name,
   CONCAT('Language ', OLD.name, ' with ID ', OLD.language_id, ' was deleted
on ', NOW()));
END;
//
DELIMITER ;

-- Insert some sample data into the language table
INSERT INTO language (language_id, name) VALUES (111, 'nope ');
INSERT INTO language (language_id, name) VALUES (121, 'animal');
INSERT INTO language (language_id, name) VALUES (131, 'code');



DELETE FROM language WHERE language_id = 111;

SELECT * FROM language_after_delete;
```



language_id	name	Status
111	NOPE	Language NOPE with ID 111 was deleted on 20...

6. Display all the triggers in your database.

show triggers;



Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connec
customer_create_date	INSERT	customer	SET NEW.create_date = NOW()	BEFORE	2023-09-19 17:21:28.10	STRICT_TRANS_TABLES,STRICT_ALL_TABLES,...	root@localhost	utf8mb4	utf8mb4_0900_a
ins_film	INSERT	film	BEGIN INSERT INTO film_text (film_id, title, d...	AFTER	2023-09-19 17:21:27.68	STRICT_TRANS_TABLES,STRICT_ALL_TABLES,...	root@localhost	utf8mb4	utf8mb4_0900_a
upd_film	UPDATE	film	BEGIN IF (old.title != new.title) OR (old.desc...	AFTER	2023-09-19 17:21:27.69	STRICT_TRANS_TABLES,STRICT_ALL_TABLES,...	root@localhost	utf8mb4	utf8mb4_0900_a
del_film	DELETE	film	BEGIN DELETE FROM film_text WHERE film_id...	AFTER	2023-09-19 17:21:27.69	STRICT_TRANS_TABLES,STRICT_ALL_TABLES,...	root@localhost	utf8mb4	utf8mb4_0900_a
uppercase	INSERT	language	BEGIN SET NEW.name = UPPER(NEW.name);...	BEFORE	2023-10-05 10:33:58.31	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900_a
display	INSERT	language	BEGIN INSERT INTO insertion_messages (me...	AFTER	2023-10-05 10:43:17.75	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900_a
language_audit_trigger	UPDATE	language	BEGIN -- Insert old record (BEFORE UPDATE)...	AFTER	2023-10-11 10:09:30.21	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900_a
language_delete_trigger	DELETE	language	BEGIN -- Insert deleted record into backup ta...	AFTER	2023-10-11 10:13:00.02	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900_a
after_language_delete	DELETE	language	BEGIN INSERT INTO language_after_delete (...	AFTER	2023-10-11 10:32:51.62	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900_a
payment_date	INSERT	payment	SET NEW.payment_date = NOW()	BEFORE	2023-09-19 17:21:28.89	STRICT_TRANS_TABLES,STRICT_ALL_TABLES,...	root@localhost	utf8mb4	utf8mb4_0900_a
rental_date	INSERT	rental	SET NEW.rental_date = NOW()	BEFORE	2023-09-19 17:21:29.35	STRICT_TRANS_TABLES,STRICT_ALL_TABLES,...	root@localhost	utf8mb4	utf8mb4_0900_a