Here’s a **step-by-step guide** from the beginning, including the changes you made. This will help you **record everything** properly.

**📌 Steps to Set Up the Trigger for SMS Processing**

**1️⃣ Create the SMS Database (sms\_db)**

Since we don’t want to modify comcenter, we create a **separate database (sms\_db)** to store messages.

sql

CopyEdit

CREATE DATABASE IF NOT EXISTS sms\_db;

**2️⃣ Create the sms\_queue Table**

This table will store messages that need to be sent.

sql

CopyEdit

USE sms\_db;

CREATE TABLE IF NOT EXISTS sms\_queue (

id INT AUTO\_INCREMENT PRIMARY KEY,

message TEXT NOT NULL,

status ENUM('pending', 'sent', 'failed', 'inactive') DEFAULT 'pending',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**3️⃣ Verify the comcenter Database**

Make sure we have the necessary tables with the correct columns:

* **fed\_call\_log** (stores call logs)
* **fed\_emergency\_type** (stores emergency types)
* **barangay** (stores barangay names)

sql

CopyEdit

USE comcenter;

DESCRIBE fed\_call\_log;

DESCRIBE fed\_emergency\_type;

DESCRIBE barangay;

Ensure that:

* fed\_call\_log has: **CallLog\_ID, Date\_Log, Type\_Emergency\_ID, Brgy\_ID, Remarks**
* fed\_emergency\_type has: **Type\_Emergency\_ID, Type\_Emergency**
* barangay has: **Brgy\_ID, Barangay**

**3.3 in maria db console: USE comcenter (then step 4)**

**4️⃣ Create the Updated Trigger**

This trigger automatically **copies new call logs** from comcenter.fed\_call\_log into sms\_db.sms\_queue with the correct formatting.

sql

CopyEdit

DELIMITER $$

CREATE TRIGGER after\_insert\_fed\_call\_log

AFTER INSERT ON fed\_call\_log

FOR EACH ROW

BEGIN

DECLARE emergency\_type VARCHAR(255);

DECLARE barangay\_name VARCHAR(255);

-- Get the actual emergency type name

SELECT Type\_Emergency INTO emergency\_type

FROM fed\_emergency\_type

WHERE Type\_Emergency\_ID = NEW.Type\_Emergency\_ID;

-- Get the barangay name

SELECT Barangay INTO barangay\_name

FROM barangay

WHERE Brgy\_ID = NEW.Brgy\_ID;

-- Insert formatted message into sms\_queue

INSERT INTO sms\_db.sms\_queue (message, status)

VALUES (

CONCAT(

NEW.CallLog\_ID, '\n',

NEW.Date\_Log, '\n',

emergency\_type, '\n',

barangay\_name, '\n',

NEW.Remarks

),

'pending'

);

END$$

DELIMITER ;

✅ **This ensures that whenever a new log is inserted into fed\_call\_log, a corresponding SMS message is added to sms\_db.sms\_queue.**

**5️⃣ Verify the Trigger**

Check if the trigger is correctly created:

sql

CopyEdit

SHOW TRIGGERS FROM comcenter;

This should display after\_insert\_fed\_call\_log in the list.

**6️⃣ Test the Trigger**

Insert a test record into fed\_call\_log:

sql

CopyEdit

INSERT INTO fed\_call\_log (CallLog\_ID, Date\_Log, Type\_Emergency\_ID, Brgy\_ID, Remarks)

VALUES ('00001', '2025-03-12', 1, 10, 'Test Fire Incident');

Then, check if it created the correct message in sms\_db.sms\_queue:

sql

CopyEdit

SELECT \* FROM sms\_db.sms\_queue;

Expected output:

yaml

CopyEdit

00001

March 12, 2025

Fire

Bugo

Test Fire Incident

**📌 Summary of Changes**

1. Created a separate sms\_db to prevent modifying comcenter.
2. Created sms\_queue to store pending SMS messages.
3. Verified the necessary tables and columns in comcenter.
4. Created a trigger that formats the message correctly.
5. Tested the trigger to ensure it works safely.

🚀 **Everything is now recorded, and your production database (comcenter) is SAFE!**  
Let me know if you want any refinements!

4o

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*JAN wanted sms format:*

*i want the message to be like this user input [CallLog\_ID] next line user input [Date\_Log] next line user input[Type\_Emergency\_ID] user input[Brgy\_ID] user input [Remarks]*

*example:*

*00000*

*March 12, 2025*

*Fire*

*Bugo*

*Grass Fire as per caller*

1. *ALTER contact\_number in* ***sms\_db.sms\_queue so that it won’t affect the user log during inserting the data into database fed\_call\_log***
2. *Whenever there is new entry that will triggered the sms in the* ***sms\_db.sms\_queue*** *under contact\_number column the default value will be “Unknow”*

*MariaDB [comcenter]>* ***ALTER TABLE sms\_db.sms\_queue***

***-> MODIFY COLUMN contact\_number VARCHAR(255) NULL;***

*Query OK, 0 rows affected (0.043 sec)*

*Records: 0 Duplicates: 0 Warnings: 0*

*MariaDB [comcenter]>* ***ALTER TABLE sms\_db.sms\_queue***

***-> MODIFY COLUMN contact\_number VARCHAR(255) DEFAULT 'Unknown';***

*Query OK, 0 rows affected (0.030 sec)*

*Records: 0 Duplicates: 0 Warnings: 0*

### Steps to update your trigger:

1. **Ensure you're using the correct database:**

sql

CopyEdit

USE comcenter;

1. **Drop the existing trigger:**

sql

CopyEdit

DROP TRIGGER IF EXISTS after\_insert\_fed\_call\_log;

1. **Create the updated trigger with your changes:**

*Version2 use this one:*

*DELIMITER $$*

*CREATE TRIGGER after\_insert\_fed\_call\_log*

*AFTER INSERT ON fed\_call\_log*

*FOR EACH ROW*

*BEGIN*

*DECLARE emergency\_type VARCHAR(255);*

*DECLARE barangay\_name VARCHAR(255);*

*-- Get the actual emergency type name*

*SELECT Type\_Emergency INTO emergency\_type*

*FROM fed\_emergency\_type*

*WHERE Type\_Emergency\_ID = NEW.Type\_Emergency\_ID;*

*-- Get the actual barangay name*

*SELECT Barangay INTO barangay\_name*

*FROM barangay*

*WHERE Barangay\_ID = NEW.Brgy\_ID;*

*-- Insert formatted message into sms\_queue*

*INSERT INTO sms\_db.sms\_queue (message, status)*

*VALUES (*

*CONCAT(*

*NEW.CallLog\_ID, '\n',*

*NEW.Date\_Log, '\n',*

*emergency\_type, '\n',*

*barangay\_name, '\n',*

*NEW.Remarks*

*),*

*'pending'*

*);*

*END$$*

*DELIMITER ;*

*Use this Trigger IF only specific cases are needed to be recorded in the sms database*

*DELIMITER $$*

*CREATE TRIGGER after\_insert\_fed\_call\_log*

*AFTER INSERT ON fed\_call\_log*

*FOR EACH ROW*

*BEGIN*

*DECLARE emergency\_type VARCHAR(255);*

*DECLARE barangay\_name VARCHAR(255);*

*-- Check if the Type\_Emergency\_ID is in the allowed list*

*IF NEW.Type\_Emergency\_ID IN (68, 46, 57, 86, 159, 9, 161) THEN*

*-- Get the actual emergency type name*

*SELECT Type\_Emergency INTO emergency\_type*

*FROM fed\_emergency\_type*

*WHERE Type\_Emergency\_ID = NEW.Type\_Emergency\_ID;*

*-- Get the actual barangay name*

*SELECT Barangay INTO barangay\_name*

*FROM barangay*

*WHERE Barangay\_ID = NEW.Brgy\_ID;*

*-- Insert formatted message into sms\_queue*

*INSERT INTO sms\_db.sms\_queue (message, status)*

*VALUES (*

*CONCAT(*

*NEW.CallLog\_ID, '\n',*

*NEW.Date\_Log, '\n',*

*emergency\_type, '\n',*

*barangay\_name, '\n',*

*NEW.Remarks*

*),*

*'pending'*

*);*

*END IF;*

*END$$*

*DELIMITER ;*

### ****If You Only Want Today's Data (Within the Last Hour)****

sql

CopyEdit

SELECT \* FROM fed\_call\_log

WHERE Date\_Log >= NOW() - INTERVAL 1 HOUR

AND DATE(Date\_Log) = CURDATE();

Or

SELECT \* FROM comcenter.fed\_call\_log

ORDER BY CallLog\_ID DESC

LIMIT 1;

recipients = ["+639353767468", "+639268838842"]

RECIPIENT\_NUMBER = "+639353767468"

***Edited insert CallLog\_ID (use this as final trigger --- old)***

*DELIMITER $$*

*CREATE TRIGGER after\_insert\_fed\_call\_log*

*AFTER INSERT ON fed\_call\_log*

*FOR EACH ROW*

*BEGIN*

*DECLARE emergency\_type VARCHAR(255);*

*DECLARE barangay\_name VARCHAR(255);*

*-- Check if the Type\_Emergency\_ID is in the allowed list*

*IF NEW.Type\_Emergency\_ID IN (68, 44, 62, 46, 57, 86, 159, 9, 161) THEN*

*-- Get the actual emergency type name*

*SELECT Type\_Emergency INTO emergency\_type*

*FROM fed\_type\_emergency*

*WHERE Type\_Emergency\_ID = NEW.Type\_Emergency\_ID;*

*-- Get the actual barangay name*

*SELECT Barangay INTO barangay\_name*

*FROM barangay*

*WHERE Barangay\_ID = NEW.Brgy\_ID;*

*-- Insert CallLog\_ID and formatted message into sms\_queue*

*INSERT INTO sms\_db.sms\_queue (CallLog\_ID, message, status)*

*VALUES (*

*NEW.CallLog\_ID, -- ✅ Store CallLog\_ID directly*

*CONCAT(*

*NEW.CallLog\_ID, '\n',*

*NEW.Date\_Log, '\n',*

*emergency\_type, '\n',*

*barangay\_name, '\n',*

*NEW.Remarks*

*),*

*'pending'*

*);*

*END IF;*

*END$$*

*DELIMITER ;*

*This is the corrected version “Trigger”*

*First create the Trigger log to monitor if it is really fetching the desired criteria*

*USE sms\_db;*

*CREATE TABLE trigger\_log (*

*log\_id INT AUTO\_INCREMENT PRIMARY KEY,*

*CallLog\_ID INT,*

*emergency\_type VARCHAR(255),*

*barangay\_name VARCHAR(255),*

*message TEXT,*

*created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP*

*);*

***The corrected trigger:***

*DELIMITER $$*

*CREATE TRIGGER after\_insert\_fed\_call\_log*

*AFTER INSERT ON comcenter.fed\_call\_log*

*FOR EACH ROW*

*BEGIN*

*-- ✅ Declare all variables at the start*

*DECLARE emergency\_type VARCHAR(255);*

*DECLARE barangay\_name VARCHAR(255);*

*DECLARE log\_message TEXT;*

*-- Log BEFORE trying to insert into sms\_queue*

*INSERT INTO sms\_db.trigger\_log (CallLog\_ID, emergency\_type, barangay\_name, message)*

*VALUES (NEW.CallLog\_ID, 'DEBUG: Before Insert', 'Trigger is running', 'Trying to insert into sms\_queue');*

*-- Get Emergency Type Name*

*SELECT Type\_Emergency INTO emergency\_type*

*FROM fed\_type\_emergency*

*WHERE Type\_Emergency\_ID = CAST(NEW.Type\_Emergency\_ID AS UNSIGNED);*

*-- Get Barangay Name*

*SELECT Barangay INTO barangay\_name*

*FROM barangay*

*WHERE Barangay\_ID = NEW.Brgy\_ID;*

*-- Construct Message*

*SET log\_message = CONCAT(*

*NEW.CallLog\_ID, '\n',*

*NEW.Date\_Log, '\n',*

*emergency\_type, '\n',*

*barangay\_name, '\n',*

*NEW.Remarks*

*);*

*-- Insert into sms\_queue*

*INSERT INTO sms\_db.sms\_queue (CallLog\_ID, message, status)*

*VALUES (NEW.CallLog\_ID, log\_message, 'pending');*

*-- Log AFTER inserting into sms\_queue*

*INSERT INTO sms\_db.trigger\_log (CallLog\_ID, emergency\_type, barangay\_name, message)*

*VALUES (NEW.CallLog\_ID, emergency\_type, barangay\_name, 'DEBUG: After Insert into sms\_queue');*

*END$$*

*DELIMITER ;*

*Test insert Data manual:*

*INSERT INTO fed\_call\_log (CallLog\_ID, Date\_Log, Type\_Emergency\_ID, Brgy\_ID, Remarks)*

*VALUES (99999, NOW(), 9, 1, 'Test trigger log');*

*Emergency ID*

*(68, 44, 62, 46, 57, 86, 159, 9, 161)*

Fire, Robbery, Drowning, Shooting incident, Explosion, Structural Collapse, Missing Person, Dead body, Suicide

**Local Db fetch from comcenter.fed\_call\_log Trigger (currently using before Barangay update)**

**DELIMITER $$**

**CREATE TRIGGER after\_insert\_fed\_call\_log**

**AFTER INSERT ON sms\_db.fed\_call\_log**

**FOR EACH ROW**

**BEGIN**

**-- Declare all variables at the start**

**DECLARE emergency\_type VARCHAR(255);**

**DECLARE barangay\_name VARCHAR(255);**

**DECLARE log\_message TEXT;**

**DECLARE formatted\_date VARCHAR(50);**

**-- Get Emergency Type Name from sms\_db.fed\_type\_emergency**

**SELECT Type\_Emergency INTO emergency\_type**

**FROM sms\_db.fed\_type\_emergency**

**WHERE Type\_Emergency\_ID = CAST(NEW.Type\_Emergency\_ID AS UNSIGNED);**

**-- Get Barangay Name from sms\_db.barangay**

**SELECT Barangay INTO barangay\_name**

**FROM sms\_db.barangay**

**WHERE Barangay\_ID = NEW.Barangay;**

**-- Format the Date\_Log field (NEW.Date\_Log) to match your desired format**

**SET formatted\_date = DATE\_FORMAT(NEW.Date\_Log, '%M %d, %Y, %h:%i %p');**

**-- Construct Message (Concatenate all relevant fields into the log\_message)**

**SET log\_message = CONCAT(**

**NEW.CallLog\_ID, '\n',**

**formatted\_date, '\n', -- Use the dynamically formatted date**

**emergency\_type, '\n',**

**'Barangay: ', barangay\_name, '\n',**

**NEW.Remarks**

**);**

**-- Insert into sms\_db.sms\_queue**

**INSERT INTO sms\_db.sms\_queue (CallLog\_ID, message, status, created\_at)**

**VALUES (NEW.CallLog\_ID, log\_message, 'pending', NEW.Date\_Log);**

**END$$**

**DELIMITER ;**

**Local Db fetch from comcenter.fed\_call\_log Trigger**

**--update the message (Remarks)**

**--added timestamp and compare the old and new update fast**

**DELIMITER $$**

**CREATE TRIGGER after\_update\_fed\_call\_log**

**AFTER UPDATE ON sms\_db.fed\_call\_log**

**FOR EACH ROW**

**BEGIN**

**-- Declare necessary variables**

**DECLARE emergency\_type VARCHAR(255);**

**DECLARE barangay\_name VARCHAR(255);**

**DECLARE log\_message TEXT;**

**DECLARE formatted\_date VARCHAR(50);**

**DECLARE formatted\_time VARCHAR(20);**

**-- Check if the Remarks field was actually updated**

**IF OLD.Remarks <> NEW.Remarks THEN**

**-- Get Emergency Type Name from sms\_db.fed\_type\_emergency**

**SELECT Type\_Emergency INTO emergency\_type**

**FROM sms\_db.fed\_type\_emergency**

**WHERE Type\_Emergency\_ID = CAST(NEW.Type\_Emergency\_ID AS UNSIGNED);**

**-- Get Barangay Name from sms\_db.barangay**

**SELECT Barangay INTO barangay\_name**

**FROM sms\_db.barangay**

**WHERE Barangay\_ID = NEW.Barangay;**

**-- Format the Date\_Log field (NEW.Date\_Log) to match your desired format**

**SET formatted\_date = DATE\_FORMAT(NEW.Date\_Log, '%M %d, %Y, %h:%i %p');**

**-- Format the timestamp as "HH:MM, today"**

**SET formatted\_time = CONCAT(DATE\_FORMAT(CURRENT\_TIMESTAMP, '%H:%i'), ', today');**

**-- Construct the new message with updated remarks and new timestamp format**

**SET log\_message = CONCAT(**

**NEW.CallLog\_ID, '\n',**

**formatted\_date, '\n', -- Use the dynamically formatted date**

**emergency\_type, '\n',**

**'Brgy: ', barangay\_name, '\n',**

**NEW.Remarks, '\nAsof: ', formatted\_time**

**);**

**-- Update the existing entry in sms\_queue**

**UPDATE sms\_db.sms\_queue**

**SET message = log\_message,**

**status = 'pending',**

**created\_at = CURRENT\_TIMESTAMP**

**WHERE CallLog\_ID = NEW.CallLog\_ID;**

**END IF;**

**END$$**

**DELIMITER ;**

Sure! Here's a detailed explanation of the code that handles fetching data from comcenter.fed\_call\_log and inserting it into sms\_db.fed\_call\_log:

**1. Fetching Data from comcenter.fed\_call\_log:**

python

CopyEdit

def fetch\_and\_insert\_data():

try:

# Connect to comcenter database

db\_comcenter = mysql.connector.connect(\*\*DB\_CONFIG\_COMCENTER)

cursor\_comcenter = db\_comcenter.cursor()

# Fetch the latest 10 records from comcenter.fed\_call\_log

cursor\_comcenter.execute("""

SELECT CallLog\_ID, Type\_Emergency\_ID, Brgy\_ID, Remarks, Date\_Log

FROM comcenter.fed\_call\_log

WHERE Type\_Emergency\_ID IN (68, 44, 62, 46, 57, 86, 159, 9, 161)

ORDER BY Date\_Log DESC

LIMIT 10;

""")

records = cursor\_comcenter.fetchall()

if records:

print(f"Fetched {len(records)} records from comcenter.fed\_call\_log.")

else:

print("No new records found to fetch.")

db\_comcenter.close()

* **Database Connection**: First, the function connects to the comcenter database using the DB\_CONFIG\_COMCENTER dictionary, which contains the database credentials like host, user, password, etc.
* **SQL Query**: The function then runs a query to fetch the **latest 10 records** from the comcenter.fed\_call\_log table:
  + The SELECT statement retrieves the following columns: CallLog\_ID, Type\_Emergency\_ID, Brgy\_ID, Remarks, and Date\_Log.
  + It filters the records by Type\_Emergency\_ID (only includes certain emergency types defined in the IN clause).
  + The results are ordered by the Date\_Log field in descending order (ORDER BY Date\_Log DESC), meaning the most recent entries come first.
  + LIMIT 10 ensures that only the latest 10 records are fetched.
* **Fetching Records**: The query result is fetched using cursor\_comcenter.fetchall(), which returns all the records from the query as a list of tuples.
* **Checking Results**: The function checks if any records were fetched. If records are found, it prints the number of records. If no records are found, it prints "No new records found to fetch."
* **Closing Database Connection**: After fetching the records, the function closes the connection to the comcenter database using db\_comcenter.close().

**2. Inserting Data into sms\_db.fed\_call\_log:**

python

CopyEdit

# Insert records into sms\_db.fed\_call\_log

db\_sms = mysql.connector.connect(\*\*DB\_CONFIG\_SMS)

cursor\_sms = db\_sms.cursor()

for record in records:

cursor\_sms.execute("SELECT COUNT(\*) FROM sms\_db.fed\_call\_log WHERE CallLog\_ID = %s", (record[0],))

if cursor\_sms.fetchone()[0] == 0:

cursor\_sms.execute("""

INSERT INTO sms\_db.fed\_call\_log (CallLog\_ID, Type\_Emergency\_ID, Barangay, Remarks, Date\_Log, Status)

VALUES (%s, %s, %s, %s, %s, %s)

""", (record[0], record[1], record[2], record[3], record[4], 'pending'))

print(f"Inserted CallLog\_ID={record[0]} into sms\_db.fed\_call\_log.")

db\_sms.commit()

db\_sms.close()

* **Database Connection to sms\_db**: After fetching the data from comcenter, the function connects to the sms\_db database using the DB\_CONFIG\_SMS dictionary, which contains the credentials for the sms\_db.
* **Creating a Cursor**: A cursor (cursor\_sms) is created to interact with the sms\_db.
* **Loop Through Records**: The function then loops through the list of fetched records from comcenter.fed\_call\_log. Each record contains the following fields: CallLog\_ID, Type\_Emergency\_ID, Brgy\_ID, Remarks, and Date\_Log.
* **Check for Existing Records**: For each record, the function checks if the CallLog\_ID already exists in sms\_db.fed\_call\_log by executing the following query:

sql

CopyEdit

SELECT COUNT(\*) FROM sms\_db.fed\_call\_log WHERE CallLog\_ID = %s

This query returns the count of records with the same CallLog\_ID in the sms\_db.fed\_call\_log table. If the count is 0, it means that this CallLog\_ID doesn't exist in sms\_db.fed\_call\_log, and the function proceeds to insert it.

* **Insert New Record**: If the CallLog\_ID doesn't exist in sms\_db.fed\_call\_log, the function inserts the new record with the following SQL query:

sql

CopyEdit

INSERT INTO sms\_db.fed\_call\_log (CallLog\_ID, Type\_Emergency\_ID, Barangay, Remarks, Date\_Log, Status)

VALUES (%s, %s, %s, %s, %s, %s)

The values being inserted are:

* + CallLog\_ID, Type\_Emergency\_ID, Brgy\_ID, Remarks, Date\_Log, and 'pending' as the initial Status.
* **Commit and Close**: After inserting all the records, the function commits the transaction using db\_sms.commit() to make sure the changes are saved to the database. Then, the connection is closed with db\_sms.close().

**3. Final Integration in the Main Loop:**

The fetch\_and\_insert\_data() function is called in the main loop of your script to ensure that it runs repeatedly and fetches new records from comcenter.fed\_call\_log to insert them into sms\_db.fed\_call\_log:

python

CopyEdit

def main():

try:

while True:

fetch\_and\_insert\_data() # Fetch data from comcenter and insert into sms\_db

mark\_old\_messages\_inactive() # Mark outdated messages inactive

sms\_data = fetch\_pending\_sms() # Fetch messages that match your criteria

if sms\_data:

send\_sms(sms\_data["id"], sms\_data["message"])

else:

print("No pending SMS found.")

time.sleep(10) # Wait 10 seconds before checking again

except KeyboardInterrupt:

print("\nScript stopped by user. Exiting gracefully...")

In the main() function:

* fetch\_and\_insert\_data() is called to fetch and insert records into sms\_db.fed\_call\_log every time the loop runs.
* The script then continues to handle SMS processing by marking old messages as inactive, fetching pending messages, and sending SMS as necessary.

**Summary:**

1. **Fetch Data from comcenter.fed\_call\_log:**
   * Query the database for the latest 10 records based on specific Type\_Emergency\_ID values and the Date\_Log.
2. **Insert New Data into sms\_db.fed\_call\_log:**
   * Check if the CallLog\_ID from the fetched records already exists in sms\_db.fed\_call\_log.
   * If not, insert the record into sms\_db.fed\_call\_log with the initial pending status.
3. **Integration with the Main Loop:**
   * The entire process is integrated into the main loop to ensure continuous fetching and inserting of data.

By following these steps, the data from comcenter.fed\_call\_log is securely transferred and inserted into sms\_db.fed\_call\_log, ready for SMS processing.

Updated trigger no duplication on sms (in\_progress method\_\_current\_using):

DELIMITER $$

CREATE TRIGGER after\_update\_fed\_call\_log

AFTER UPDATE ON sms\_db.fed\_call\_log

FOR EACH ROW

BEGIN

-- Declare necessary variables

DECLARE emergency\_type VARCHAR(255);

DECLARE barangay\_name VARCHAR(255);

DECLARE log\_message TEXT;

DECLARE formatted\_date VARCHAR(50);

DECLARE formatted\_time VARCHAR(20);

-- Check if the Remarks field was actually updated

IF OLD.Remarks <> NEW.Remarks THEN

-- Get Emergency Type Name from sms\_db.fed\_type\_emergency

SELECT Type\_Emergency INTO emergency\_type

FROM sms\_db.fed\_type\_emergency

WHERE Type\_Emergency\_ID = CAST(NEW.Type\_Emergency\_ID AS UNSIGNED);

-- Get Barangay Name from sms\_db.barangay

SELECT Barangay INTO barangay\_name

FROM sms\_db.barangay

WHERE Barangay\_ID = NEW.Barangay;

-- Format the Date\_Log field (NEW.Date\_Log) to match your desired format

SET formatted\_date = DATE\_FORMAT(NEW.Date\_Log, '%M %d, %Y, %h:%i %p');

-- Format the timestamp as "HH:MM, today"

SET formatted\_time = CONCAT(DATE\_FORMAT(CURRENT\_TIMESTAMP, '%H:%i'), ', today');

-- Construct the new message with updated remarks and new timestamp format

SET log\_message = CONCAT(

NEW.CallLog\_ID, '\n',

formatted\_date, '\n', -- Use the dynamically formatted date

emergency\_type, '\n',

'Brgy: ', barangay\_name, '\n',

NEW.Remarks, '\nAsof: ', formatted\_time

);

-- Update the existing entry in sms\_queue with 'pending' status

UPDATE sms\_db.sms\_queue

SET message = log\_message,

status = 'pending', -- Set to pending so it can be sent next

created\_at = CURRENT\_TIMESTAMP

WHERE CallLog\_ID = NEW.CallLog\_ID;

-- Mark the record as 'in\_progress' immediately after updating remarks

UPDATE sms\_db.sms\_queue

SET status = 'in\_progress'

WHERE CallLog\_ID = NEW.CallLog\_ID AND status = 'pending';

END IF;

END$$

DELIMITER ;

Trigger Update Barangay name and REMARKS

DELIMITER $$

CREATE TRIGGER after\_update\_fed\_call\_log

AFTER UPDATE ON sms\_db.fed\_call\_log

FOR EACH ROW

BEGIN

-- Declare necessary variables

DECLARE emergency\_type VARCHAR(255);

DECLARE barangay\_name VARCHAR(255);

DECLARE log\_message TEXT;

DECLARE formatted\_date VARCHAR(50);

DECLARE formatted\_time VARCHAR(20);

-- Check if Remarks or Brgy\_ID was updated

IF OLD.Remarks <> NEW.Remarks OR OLD.Barangay <> NEW.Barangay THEN

-- Get Emergency Type Name from sms\_db.fed\_type\_emergency

SELECT Type\_Emergency INTO emergency\_type

FROM sms\_db.fed\_type\_emergency

WHERE Type\_Emergency\_ID = CAST(NEW.Type\_Emergency\_ID AS UNSIGNED);

-- Get Barangay Name from sms\_db.barangay

SELECT Barangay INTO barangay\_name

FROM sms\_db.barangay

WHERE Barangay\_ID = NEW.Barangay;

-- Format the Date\_Log field

SET formatted\_date = DATE\_FORMAT(NEW.Date\_Log, '%M %d, %Y, %h:%i %p');

-- Format the timestamp as "HH:MM, today"

SET formatted\_time = CONCAT(DATE\_FORMAT(CURRENT\_TIMESTAMP, '%H:%i'), ', today');

-- Construct the updated message

SET log\_message = CONCAT(

NEW.CallLog\_ID, '\n',

formatted\_date, '\n',

emergency\_type, '\n',

'Brgy: ', barangay\_name, '\n',

NEW.Remarks, '\nAsof: ', formatted\_time

);

-- Update the sms\_queue entry

UPDATE sms\_db.sms\_queue

SET message = log\_message,

status = 'pending',

created\_at = CURRENT\_TIMESTAMP

WHERE CallLog\_ID = NEW.CallLog\_ID;

-- Mark as in\_progress immediately

UPDATE sms\_db.sms\_queue

SET status = 'in\_progress'

WHERE CallLog\_ID = NEW.CallLog\_ID AND status = 'pending';

END IF;

END$$

DELIMITER ;

Add\_recipient query

INSERT INTO sms\_db.recipient\_list (name, phone\_number)

VALUES ('Jan', '+639353767468');

After insert (current using 4.11.2025)

DELIMITER $$

CREATE TRIGGER after\_insert\_fed\_call\_log

AFTER INSERT ON sms\_db.fed\_call\_log

FOR EACH ROW

BEGIN

-- Declare variables

DECLARE emergency\_type VARCHAR(255);

DECLARE barangay\_name VARCHAR(255);

DECLARE log\_message TEXT;

DECLARE formatted\_date VARCHAR(50);

DECLARE formatted\_time VARCHAR(20);

-- Get Emergency Type Name

SELECT Type\_Emergency INTO emergency\_type

FROM sms\_db.fed\_type\_emergency

WHERE Type\_Emergency\_ID = CAST(NEW.Type\_Emergency\_ID AS UNSIGNED);

-- Get Barangay Name

SELECT Barangay INTO barangay\_name

FROM sms\_db.barangay

WHERE Barangay\_ID = NEW.Barangay;

-- Format the Date\_Log field

SET formatted\_date = DATE\_FORMAT(NEW.Date\_Log, '%M %d, %Y, %h:%i %p');

SET formatted\_time = CONCAT(DATE\_FORMAT(CURRENT\_TIMESTAMP, '%H:%i'), ', today');

-- Build the message

SET log\_message = CONCAT(

NEW.CallLog\_ID, '\n',

formatted\_date, '\n',

emergency\_type, '\n',

'Brgy: ', barangay\_name, '\n',

NEW.Remarks, '\nAsof: ', formatted\_time

);

-- Insert directly into sms\_queue

INSERT INTO sms\_db.sms\_queue (CallLog\_ID, message, status, created\_at)

VALUES (NEW.CallLog\_ID, log\_message, 'pending', CURRENT\_TIMESTAMP);

END$$

DELIMITER ;