

Se crearon las siguientes tablas:

```
CREATE DATABASE HOMESALES;  
  
CREATE TABLE CLIENTES(  
CLIENTE_ID INT NOT NULL AUTO_INCREMENT,  
CLIENTE_NOMBRE VARCHAR(100) NOT NULL,  
CLIENTE_TEL VARCHAR(30),  
CLIENTE_EMAIL VARCHAR(50),  
PRIMARY KEY(CLIENTE_ID)  
);
```

```
DROP TABLE IF EXISTS COMENTARIOS;  
CREATE TABLE COMENTARIOS(  
COMENTARIO_ID INT NOT NULL,  
CLIENTE_ID INT NOT NULL,  
COMENTARIO_FECHA TIMESTAMP NOT NULL,  
COMENTARIO TEXT,  
PRIMARY KEY(COMENTARIO_ID, CLIENTE_ID),  
FOREIGN KEY(CLIENTE_ID) REFERENCES CLIENTES(CLIENTE_ID)  
);
```

```
DROP TABLE IF EXISTS CLASIFICACION;  
CREATE TABLE CLASIFICACION(  
CLASIFICACION_ID INT NOT NULL AUTO_INCREMENT,  
CLIENTE_ID INT NOT NULL,  
CLASIFICACION VARCHAR(30) NOT NULL,  
  
PRIMARY KEY(CLASIFICACION_ID),  
FOREIGN KEY(CLIENTE_ID) REFERENCES COMENTARIOS(CLIENTE_ID)  
);
```

Nota: hice algunas modificaciones ya que noté deficiencias en el otro modelo sobre todo cuando se borra un comentario para que el procedimiento se pueda usar n veces

### Creación de la vista

```
DROP VIEW IF EXISTS MOSTRAR_CLASS;  
CREATE VIEW MOSTRAR_CLASS AS  
SELECT CLASIFICACION.CLIENTE_ID,CLIENTES.CLIENTE_NOMBRE,ULTIMO_COMENTARIO, CLASIFICACION  
FROM CLASIFICACION  
INNER JOIN CLIENTES  
ON CLASIFICACION.CLIENTE_ID=CLIENTES.CLIENTE_ID;
```

## Creación del procedimiento:

```
DROP PROCEDURE IF EXISTS SP_BITACORA;
DELIMITER //
CREATE PROCEDURE SP_BITACORA()
BEGIN
    --declaracion de variables
    DECLARE done INT DEFAULT FALSE;
    DECLARE VAL_COM TEXT(100);
    DECLARE VAL_IDCLI INT;
    DECLARE ESTADO VARCHAR(50);

    DECLARE CUR CURSOR FOR
        SELECT CLIENTE_ID
        FROM COMENTARIOS group by cliente_ID;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN CUR;
    --aquí hice una pequeña modificación del repeat por loop ya que el repeat al momento de instertar
    --en la tabla comentarios me insertaba dos veces el ultimo cliente

    --aquí inicia el loop

    read_loop: LOOP
    FETCH CUR INTO VAL_IDCLI;

    IF done THEN
        LEAVE read_loop;
    END IF;

    --al principio habia planteado controlar el ultimo mensaje asignando un id al comentario, pero detecte que
    --borraba un comentario el conteo se hacia mal y las modificaciones tambien
    --aquí solo selecciono el ulimo mensaje con la ayuda de variable fecha utilizando el id del cliente

    SET VAL_COM = (SELECT COMENTARIO
        FROM COMENTARIOS WHERE CLIENTE_ID = VAL_IDCLI
        GROUP BY COMENTARIO_FECHA
        DESC LIMIT 1);
    SET VAL_COM = LOWER(VAL_COM);

    --en estos if le doy un estado al ultimo comentario del cliente
    IF (VAL_COM LIKE 'quiere conocer el fraccionamiento' OR VAL_COM LIKE 'quiere visitar la casa modelo') THEN
        SET ESTADO = 'VERDE';
    ELSEIF (VAL_COM LIKE 'quiere que le llamen' OR VAL_COM LIKE 'quiere saber si califica') THEN
        SET ESTADO = 'AMARILLO';
    ELSEIF (VAL_COM LIKE 'nos llama despues' OR VAL_COM LIKE 'preguntara a su esposa' ) THEN
        SET ESTADO = 'ROJO';
    ELSE
        SET ESTADO = 'BLANCO';
    END IF;

    --Aquí lo que hago es ver si se repiten datos en una columna en caso de ser si solo hago update, y en caso de
    -- ser no hago la inserccion, esta es otra de las moficiaciones que hice
    --con el fin de que el procedimiento se pudiera usar todas las veces que se quiera, con el otro tenia que
    --borrar todos los registros y resetear el auto increment para usar el procedimiento sin que se repitan
    --columnas
    IF(EXISTS(SELECT CLASIFICACION_ID FROM CLASIFICACION WHERE CLIENTE_ID = VAL_IDCLI)) THEN
        UPDATE CLASIFICACION SET CLASIFICACION = ESTADO, ULTIMO_COMENTARIO = VAL_COM WHERE CLIENTE_ID = VAL_IDCLI;
    ELSE
        INSERT INTO CLASIFICACION (CLIENTE_ID, CLASIFICACION, ultimo_comentario) VALUES (VAL_IDCLI,ESTADO, VAL_COM);
    END IF;
    END LOOP;
    CLOSE CUR;
    END
    //
    DELIMITER ;
```

## Creación del trigger

```
DROP TRIGGER IF EXISTS TG_BITACORA_AI;
DELIMITER //
CREATE TRIGGER TG_BITACORA_AI
AFTER INSERT ON COMENTARIOS
FOR EACH ROW
BEGIN
    --declaracion de variables
    DECLARE VAL_COM TEXT;
    DECLARE IDCLI INT;
    DECLARE VAL_IDMEN INT;
    DECLARE LASTID INT;
    DECLARE ESTADO VARCHAR(50);

    --obtengo el ultimo comentario insertado del cliente
    SET VAL_COM = (SELECT COMENTARIO FROM COMENTARIOS
    WHERE COMENTARIO_ID = NEW.COMENTARIO_ID
    AND CLIENTE_ID=NEW.CLIENTE_ID);

    SET VAL_COM = LOWER(VAL_COM);

    --obtengo el id del cliente
    SET IDCLI = (SELECT CLIENTE_ID FROM COMENTARIOS
    WHERE COMENTARIO_ID = NEW.COMENTARIO_ID
    AND CLIENTE_ID=NEW.CLIENTE_ID);

    --en estos if le doy un estado al ultimo comentario del cliente
    IF (VAL_COM LIKE 'quiere conocer el fraccionamiento' OR VAL_COM LIKE 'quiere visitar la casa modelo') THEN
        SET ESTADO = 'VERDE';
    ELSEIF (VAL_COM LIKE 'quiere que llamen' OR VAL_COM LIKE 'quiere saber si califica') THEN
        SET ESTADO = 'AMARILLO';
    ELSEIF (VAL_COM LIKE 'nos llama despues' OR VAL_COM LIKE 'preguntara a su esposa' ) THEN
        SET ESTADO = 'ROJO';
    ELSE
        SET ESTADO = 'BLANCO';
    END IF;

    --Aquí lo que hago es ver si se repiten datos en una columna en caso de ser si solo hago update, y en caso de
    -- ser no hago la inserccion, esta es otra de las moficiaciones que hice
    --con el fin de que el procedimiento se pudiera usar todas las veces que se quiera, con el otro tenia que
    --borrar todos los registros y resetear el auto increment para usar el procedimiento sin que se repitan
    --columnas
    IF(EXISTS(SELECT CLASIFICACION_ID FROM CLASIFICACION WHERE CLIENTE_ID = IDCLI)) THEN
        UPDATE CLASIFICACION SET CLASIFICACION = ESTADO, ULTIMO_COMENTARIO = NEW.COMENTARIO WHERE CLIENTE_ID = IDCLI;
    ELSE
        INSERT INTO CLASIFICACION (CLIENTE_ID, CLASIFICACION, ultimo_comentario) VALUES (NEW.CLIENTE_ID,ESTADO, NEW.COMENTARIO);
    END IF;
END
//
DELIMITER ;
```

Esta es la tabla de clientes con datos:

```
mysql> select * from clientes;
```

CLIENTE_ID	CLIENTE_NOMBRE	CLIENTE_TEL	CLIENTE_EMAIL
1	ALAN ACOSTA BAEZ	6623351163	bryanbaez2504@gmail.com
2	JUAN TREVINO ROBLES	6623363355	heyjhonny@gmail.com
3	WILLIAM PENA VALENZUELA	6622334334	willpv@gmail.com
4	CARLOS SALINAS SAAVEDRA	6623918832	csalsa95@gmail.com
5	LUIS CUBEDO INIGUEZ	6622222222	luiscepillo@gmail.com

```
5 rows in set (0.00 sec)
```

Esta es la tabla comentarios con datos:

```
mysql> select * from comentarios;
```

COMENTARIO_ID	CLIENTE_ID	COMENTARIO_FECHA	COMENTARIO
1	1	2019-02-21 22:16:17	BUSCA INFORMACION
2	1	2019-02-21 22:20:04	QUIERE SABER SI CALIFICA
3	1	2019-02-21 22:21:26	LE PREGUNTARA A SU ESPOSA
4	1	2019-02-21 22:22:21	PREGUNTARA A SU ESPOSA
1	2	2019-02-21 22:26:00	PREGUNTARA A SU ESPOSA
2	2	2019-02-21 22:26:35	QUIERE SABER SI CALIFICA
1	3	2019-02-21 22:37:50	busca informacion
2	3	2019-02-21 22:38:17	quiere saber si califica

```
8 rows in set (0.00 sec)
```

Tabla de clasificación después de activar el procedimiento:

```
mysql> call sp_bitacora;
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> select * from clasificacion;
```

CLASIFICACION_ID	CLIENTE_ID	CLASIFICACION	ultimo_comentario
1	1	ROJO	preguntara a su esposa
2	2	AMARILLO	quiere saber si califica
3	3	AMARILLO	quiere saber si califica

```
3 rows in set (0.00 sec)
```

## Insert manual para activar el trigger

```
mysql> INSERT INTO COMENTARIOS(COMENTARIO_ID, CLIENTE_ID, COMENTARIO_FECHA, COMENTARIO) VALUES (3,3,NOW(), 'quiere visitar la casa modelo');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from clasificacion;
```

CLASIFICACION_ID	CLIENTE_ID	CLASIFICACION	ultimo_comentario
1	1	ROJO	preguntara a su esposa
2	2	AMARILLO	quiere saber si califica
3	3	VERDE	quiere visitar la casa modelo

3 rows in set (0.00 sec)

```
mysql> INSERT INTO COMENTARIOS(COMENTARIO_ID, CLIENTE_ID, COMENTARIO_FECHA, COMENTARIO) VALUES (1,4,NOW(), 'SIN INTERES');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM CLASIFICACION;
```

CLASIFICACION_ID	CLIENTE_ID	CLASIFICACION	ultimo_comentario
1	1	ROJO	preguntara a su esposa
2	2	AMARILLO	quiere saber si califica
3	3	VERDE	quiere visitar la casa modelo
4	4	BLANCO	SIN INTERES

4 rows in set (0.00 sec)

## Vista

```
mysql> select * from mostrar_class;
```

CLIENTE_ID	CLIENTE_NOMBRE	ULTIMO_COMENTARIO	CLASIFICACION
1	ALAN ACOSTA BAEZ	preguntara a su esposa	ROJO
2	JUAN TREVINO ROBLES	quiere saber si califica	AMARILLO
3	WILLIAM PENA VALENZUELA	quiere visitar la casa modelo	VERDE

3 rows in set (0.00 sec)