

GRAMMYS LATINOS

Genera los siguientes TRIGGERS:

1. Crear un TRIGGER para registrar la fecha de lanzamiento de la canción como la fecha actual si no se especifica en la inserción.

DROP TRIGGER IF EXISTS Before_Insert_Canciones;

DELIMITER //

```
CREATE TRIGGER Before_Insert_Canciones
BEFORE INSERT ON Canciones
FOR EACH ROW
BEGIN
    IF NEW.Anio_lanzamiento IS NULL OR NEW.Anio_lanzamiento = 0 THEN
        SET NEW.Anio_lanzamiento = YEAR(CURRENT_DATE);
    END IF;
END//
```

DELIMITER ;

9	Bidi Bidi Bom Bom	9	00:03:31	1994	Amor Prohibido
10	Conga	10	00:04:15	1985	Primitive Love
11	Javier I el Grande	11	00:04:35	2002	Javier I el Grande
12	Poquito a poco	12	00:03:45	2001	Poquito a poco
13	Cancion de prueba	1	00:04:00	2024	Álbum de prueba
*	NULL	NULL	NULL	NULL	NULL

2. Crear un TRIGGER que al insertar una canción, si no localiza al artista en la tabla de artistas, lo inserta como nuevo artista en dicha tabla con el Nombre = 'Artista creador del tema nombre_cancion'.

```
DROP TRIGGER IF EXISTS Before_Insert_Canciones_Artista;
```

```
-- Definir el nuevo trigger
```

```
DELIMITER //
```

```
CREATE TRIGGER Before_Insert_Canciones_Artista
```

```
BEFORE INSERT ON Canciones
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE artist_id INT;
```

```
-- Buscar si el artista ya existe en la tabla de artistas
```

```
    SELECT ID INTO artist_id FROM Artistas WHERE Nombre = NEW.Nombre_cancion;
```

```
-- Si el artista no existe, insertarlo como nuevo artista
```

```
    IF artist_id IS NULL THEN
```

```
        INSERT INTO Artistas (Nombre, Discografica)
```

```
        VALUES (CONCAT('Artista creador del tema ', NEW.Nombre_cancion), 'Desconocida');
```

```
        SET artist_id = LAST_INSERT_ID();
```

```
    END IF;
```

```
-- Asignar el ID del artista a la nueva canción
```

```
    SET NEW.Artista_ID = artist_id;
```

```
END//
```

```
DELIMITER ;
```

11	Kase.O	Español	1978	BOA	11
12	El Arrebato	Español	1970	EMI Music	4
13	Artista creador del tema Cancion de prueba 2	NULL	NULL	Desconocida	NULL
*	NULL	NULL	NULL	NULL	NULL

3. Crea un TRIGGER que al insertar una línea en la lista de reproducciones actualice el número de reproducciones para una misma canción.

```
-- Eliminar el trigger si existe
```

```
DROP TRIGGER IF EXISTS After_Insert_Reproducciones;
```

```
DELIMITER //
```

```
CREATE TRIGGER After_Insert_Reproducciones
```

```
AFTER INSERT ON Reproducciones
```

```
FOR EACH ROW
```

```
BEGIN
```

```

DECLARE total_reproducciones INT;

-- Obtener el total de reproducciones para la canción
SELECT SUM(Cantidad_reproducciones) INTO total_reproducciones
FROM Reproducciones
WHERE Cancion_ID = NEW.Cancion_ID;

-- Actualizar el número de reproducciones en la tabla de canciones
UPDATE Canciones
SET Canciones.Cantidad_reproducciones = total_reproducciones
WHERE Canciones.ID = NEW.Cancion_ID;
END//
DELIMITER ;

-- Comprobamos trigger 3

-- Insertar una nueva fila en la tabla de Reproducciones
INSERT INTO Reproducciones (Cancion_ID, Cantidad_reproducciones) VALUES (1, 50);

-- Verificar si el número de reproducciones de la canción correspondiente se actualizó
correctamente

SELECT * FROM Reproducciones where Cancion_ID = 1;

```

ID	Cancion_ID	Fecha_reproduccion	Cantidad_reproducciones
1	1	2024-04-30 14:07:21	100
15	1	2024-04-30 14:13:59	50
NULL	NULL	NULL	NULL

select*from reproducciones;

ID	Cancion_ID	Fecha_reproduccion	Cantidad_reproducciones
1	1	2024-04-30 14:07:21	100
2	2	2024-04-30 14:07:21	150
3	3	2024-04-30 14:07:21	200
4	4	2024-04-30 14:07:21	300
5	5	2024-04-30 14:07:21	250
6	6	2024-04-30 14:07:21	400
7	7	2024-04-30 14:07:21	500
8	8	2024-04-30 14:07:21	350
9	9	2024-04-30 14:07:21	200
10	10	2024-04-30 14:07:21	150
11	11	2024-04-30 14:07:21	50
15	1	2024-04-30 14:13:59	50
NULL	NULL	NULL	NULL