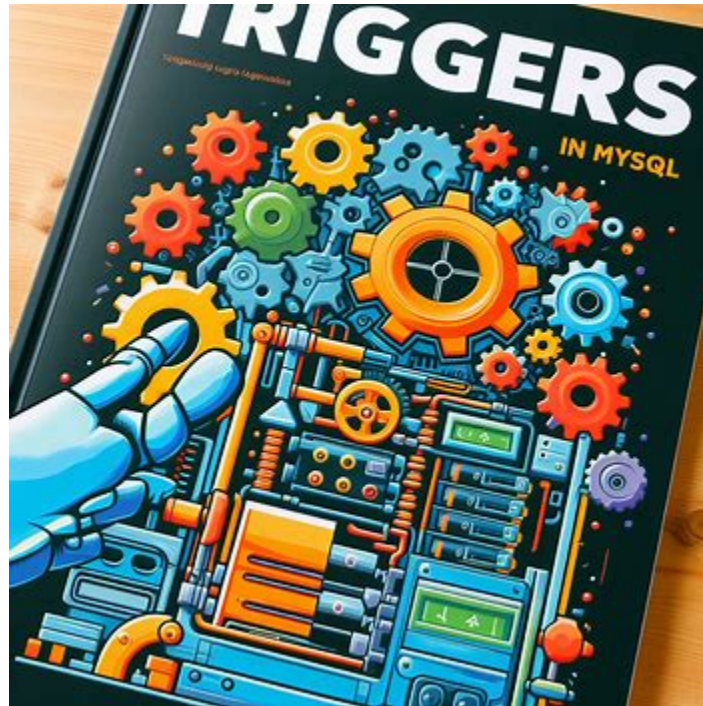


# TRIGGERS



IVÁN DE LA FUENTE VARONA

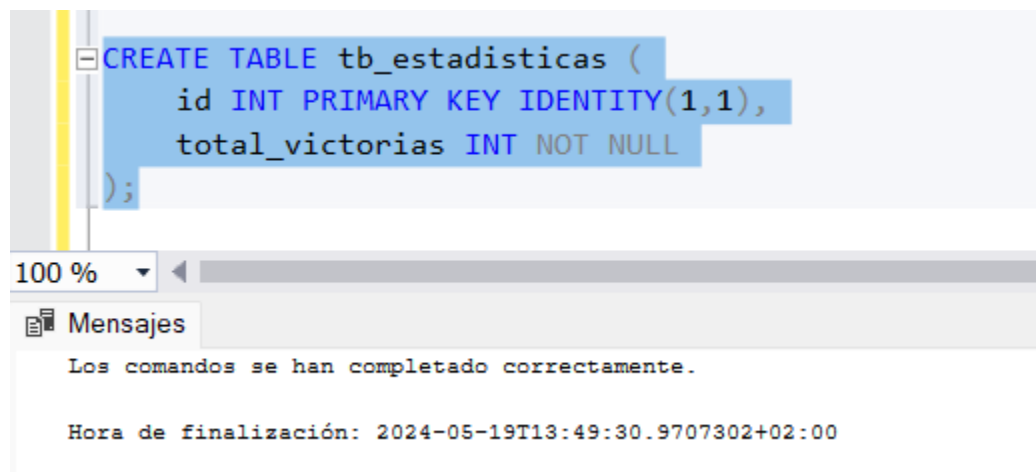
## ENUNCIADO TRIGGER 1

Este disparador asegurará que no se pueda insertar un piloto con el mismo `nombre`, `apellido1`, y `apellido2` más de una vez en la tabla `tb_pilotos`. Esta restricción no se puede aplicar con un `CHECK` en MySQL, por lo tanto, usamos un disparador `BEFORE INSERT`.

He decidido realizar este trigger por los siguientes motivos:

1. Evitar Duplicados: La existencia de pilotos con la misma combinación de nombre y apellidos puede llevar a confusión y errores en la gestión de los datos, como la contabilización de estadísticas de carrera o la administración de licencias.
2. Integridad de Datos: Al prevenir duplicados, se garantiza que cada piloto en la base de datos es único, lo cual es crucial para la integridad referencial en el caso de que otras tablas dependan de `tb_pilotos`.
3. Mantenimiento y Administración: Tener datos duplicados puede complicar tareas de mantenimiento y actualización de registros, por lo que evitar duplicados simplifica la administración de la base de datos.

## CREACIÓN



```
CREATE TABLE tb_estadisticas (  
  id INT PRIMARY KEY IDENTITY(1,1),  
  total_victorias INT NOT NULL  
);
```

100 %

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2024-05-19T13:49:30.9707302+02:00

Antes de nada deberemos crear una nueva tabla llamada estadísticas.

```
INSERT INTO tb_estadisticas (total_victorias) VALUES (0);
```

100 %

Mensajes

(1 fila afectada)

Hora de finalización: 2024-05-19T13:50:31.4365222+02:00

Le daremos como registro inicial el valor de 0.

```
CREATE TRIGGER trg_after_insert_pilotos  
ON tb_pilotos  
AFTER INSERT  
AS  
BEGIN  
    UPDATE tb_estadisticas  
    SET total_victorias = (  
        SELECT SUM(num_victorias)  
        FROM tb_pilotos  
    );  
END;
```

100 %

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2024-05-19T13:51:37.6685573+02:00

DISPARADOR

100 %

SQL

Mensajes

INSERT INTO tb\_pilotos (superlicencia, nombre\_piloto, apellido1\_piloto, apellido2\_piloto, peso, altura\_cm, num\_carreras, num\_victorias)

VALUES (23460, 'New', 'Pilot', NULL, 70.0, 180, 50, 5);

(1 fila afectada)

(1 fila afectada)

(1 fila afectada)

Hora de finalización: 2024-05-19T13:53:16.5720897+02:00

FUNCIONAMIENTO CORRECTO

83 %

SQL

Resultados

Mensajes

SELECT \* FROM tb\_pilotos;

SELECT \* FROM tb\_estadísticas;

	superlicencia	nombre_piloto	apellido1_piloto	apellido2_piloto	peso	altura_cm	num_carreras	num_victorias
6	23458	Lance	Stroll	NULL	72.0	185	65	5
7	23459	Fernando	Alonso	NULL	69.0	179	110	30
8	23460	New	Pilot	NULL	70.0	180	50	5
9	34567	Max	Verstappen	NULL	67.8	183	90	30
10	34568	Sergio	Pérez	NULL	69.4	180	70	8
11	45678	Valtteri	Bottas	NULL	72.3	178	95	20
12	45679	Nicholas	Latifi	NULL	70.1	183	50	2
13	56780	George	Russell	NULL	67.0	176	40	1

id	total_victorias
1	374

Consulta ejecutada correctamente.

IVAN-LP\SQLSERVER (15.0 RTM) | sa (51) | master | 00:00:00 | 23 filas

## ENUNCIADO TRIGGER 2

Este disparador asegurará que no se pueda insertar un piloto con el mismo `nombre`, `apellido1`, y `apellido2` más de una vez en la tabla `tb_pilotos`. Esta restricción no se puede aplicar con un `CHECK` en MySQL, por lo tanto, usamos un disparador `BEFORE INSERT`.

He decidido realizar este trigger por los siguientes motivos:

1. **Agregación de Datos:** Mantener un total agregado de victorias en una tabla separada facilita el acceso rápido a estadísticas globales sin necesidad de recalcular este valor cada vez que se necesita, mejorando el rendimiento de las consultas.
2. **Simplificación de Consultas:** Al tener el total de victorias pre-calculado, las consultas sobre estadísticas generales son más simples y eficientes, evitando la necesidad de realizar cálculos en tiempo real sobre un gran conjunto de datos.
3. **Actualización Automática:** Asegurar que las estadísticas se actualicen automáticamente en respuesta a cambios en la tabla `tb_pilotos` reduce el riesgo de inconsistencias y errores que podrían surgir si este cálculo se hiciera manualmente.
4. **Análisis y Reportes:** Para tareas de análisis y generación de reportes, tener estadísticas pre-calculadas facilita el proceso y permite obtener resultados rápidamente, lo cual es crucial en entornos donde el tiempo de respuesta es importante.

# CREACIÓN

```
CREATE TRIGGER trg_before_insert_pilotos
ON tb_pilotos
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM tb_pilotos p
        JOIN inserted i ON p.nombre_piloto = i.nombre_piloto
                        AND p.apellido1_piloto = i.apellido1_piloto
                        AND p.apellido2_piloto = i.apellido2_piloto
    )
    BEGIN
        RAISERROR('Piloto ya existe con el mismo nombre y apellidos', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO tb_pilotos (superlicencia, nombre_piloto, apellido1_piloto, apellido2_piloto, peso, altura_cm, num_carreras, num_victorias)
        SELECT superlicencia, nombre_piloto, apellido1_piloto, apellido2_piloto, peso, altura_cm, num_carreras, num_victorias
        FROM inserted;
    END
END;
```

0 %

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2024-05-19T13:36:16.1900692+02:00

# DISPARADOR

```
INSERT INTO tb_pilotos (superlicencia, nombre_piloto, apellido1_piloto, apellido2_piloto, peso, altura_cm, num_carreras, num_victorias)
VALUES (23459, 'Fernando', 'Alonso', NULL, 69.0, 179, 110, 30);
```

Mensajes

mensaje 2627, nivel 14, estado 1, procedimiento trg\_before\_insert\_pilotos, línea 19 [línea de inicio de lote 29]  
violación de la restricción PRIMARY KEY 'pk\_tbpii\_super'. No se puede insertar una clave duplicada en el objeto 'dbo.tb\_pilotos'. El valor de la clave duplicada es (23459).  
terminó la instrucción.

hora de finalización: 2024-05-19T13:38:47.6930573+02:00

# FUNCIONAMIENTO CORRECTO

```
SELECT * FROM tb_pilotos;
```

100 %

Resultados Mensajes

	superlicencia	nombre_piloto	apellido1_piloto	apellido2_piloto	peso	altura_cm	num_carreras	num_victorias
1	12345	Lewis	Hamilton	NULL	70.5	175	100	50
2	12346	Kimi	Räikkönen	NULL	70.8	182	130	40
3	12347	Fernando	Alonso	NULL	69.0	179	110	30
4	23456	Sebastian	Vettel	NULL	68.2	180	110	40
5	23457	Esteban	Ocon	NULL	68.5	177	60	10
6	23458	Lance	Stroll	NULL	72.0	185	65	5
7	23459	Fernando	Alonso	NULL	69.0	179	110	30
8	34567	Max	Verstappen	NULL	67.8	183	90	30
9	34568	Sergio	Pérez	NULL	69.4	180	70	8

Consulta ejecutada correctamente. IVAN-LP\SQLEXPRESS (15.0 RTM) | sa (51) | master | 00:00:00 | 21 filas