

Triggers

Con el disparador `asegurar_num_vueltas_positivo`, se pretende garantizar la integridad de los datos en la tabla `tb_circuitos`, asegurando que el campo `num_vueltas` siempre sea un valor positivo mayor que cero.

Esto ayuda a evitar la inserción o actualización de registros con valores que no son reales como 0 o negativo

```
CREATE TRIGGER asegurar_num_vueltas_positivo
ON tb_circuitos
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE num_vueltas <= 0)
    BEGIN
        RAISERROR ('El número de vueltas debe ser mayor que cero.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END
END;
```

as

```
INSERT INTO tb_circuitos (id_circuito, nombre_circuito, cod_direccion, num_vueltas, km_circuitos, vuelta_rapida, num_aforo)
VALUES ('CI-01', 'Nuevo Circuito', 1, 0, 4.5, '00:01:30.000', 50000);
```

Con esta sentencia comprobamos el funcionamiento del trigger queriendo meter el `num_vueltas` a 0

```
INSERT INTO tb_circuitos (id_circuito, nombre_circuito, cod_direccion, num_vueltas, km_circuitos, vuelta_rapida, num_aforo)
VALUES ('CI-01', 'Nuevo Circuito', 1, 0, 4.5, '00:01:30.000', 50000);
```

89 %

Messages

Msg 80000, Level 16, State 1, Procedure asegurar_num_vueltas_positivo, Line 8 [Batch Start Line 21]
El número de vueltas debe ser mayor que cero.
Msg 3609, Level 16, State 1, Line 22
La transacción terminó en el desencadenador. Se anuló el lote.

Al ejecutar el insert nos da el error y nos dice que tiene que ser mayor que 0

Este trigger se encarga de calcular y actualizar el campo tiempo_promedio_vuelta en la tabla tb_circuitos después de que se inserte o actualice un registro en dicha tabla.

Calcular este valor automáticamente después de cada inserción o actualización garantiza que los datos estén siempre actualizados , lo que facilita el análisis y la comparación de los circuitos.

```
ALTER TABLE tb_circuitos
ADD tiempo_promedio_vuelta TIME;
```

Para la creación de este trigger he tenido que añadir el campo tiempo_promedio_vuelta a la tabla

```
CREATE TRIGGER calcular_tiempo_promedio_vuelta
ON tb_circuitos
AFTER INSERT, UPDATE
AS
BEGIN
    UPDATE tb_circuitos
    SET tiempo_promedio_vuelta = CONVERT(TIME, DATEADD(SECOND, DATEDIFF(SECOND, 0, vuelta_rapida) / num_vueltas, 0))
END;
```

Esta es la sentencia

```
INSERT INTO tb_circuitos (id_circuito, nombre_circuito, cod_direccion, num_vueltas, km_circuitos, vuelta_rapida, num_aforo)
VALUES ('CI-01', 'Nuevo Circuito', 1, 66, 4.5, '00:01:30.000', 50000);
```

89 %

Messages

(16 rows affected)

(1 row affected)

Completion time: 2024-05-16T11:01:46.6924470+02:00

```
select * from tb_circuitos
SELECT * FROM tb_circuitos WHERE id_circuito = 'CI-01';
```

89 %

	id_circuito	nombre_circuito	cod_direccion	num_vueltas	km_circuitos	vuelta_rapida	num_aforo	tiempo_promedio_vuelta
1	CI-01	Nuevo Circuito	1	66	4.500	00:01:30.0000000	50000	00:00:01.0000000

Insertamos los datos para comprobar que el trigger ha funcionado correctamente