Ejercicio 1

Uno de estos triggers debe realizar una restricción que no se haya podido hacer con check en la creación de la tabla.

Para el enunciado de este primer ejercicio, pensé en que al tener que meter las id's de forma manual para los circuitos, podrían haber erratas a la hora de poner el guión, que no estuviera bien definido u otras cuestiones, para ello he querido mantener el orden en nuestra bd sin necesidad de tener que hacer un check.

Como en nuestra tabla estaban todos los check definidos, he decidido eliminar un check para poder realizar este trabajo.
El check que he quitado es la perteneciente a la tabla tb_circuitos, donde definimos con la constraint que la id_circuito queremos que tenga el formato [A-Z][A-Z]-[0-9][0-9]

```
create table tb_circuitos(
id_circuito varchar(5) not null,
nombre_circuito varchar(50) not null,
cod_direccion numeric(10) NOT NULL,
num_vueltas numeric(2) not null,
km_circuitos decimal(4,3) not null,
vuelta_rapida time not null,
num_aforo numeric(5)  not null,

constraint pk_tbcir_idcir primary key (id_circuito),
constraint uq_tbcir_idcir unique(id_circuito),
constraint fk_tbcir_tbdir_codcir foreign key (cod_direccion) references tb_direccion (cod_direccion),
constraint ck_tbcir_idcir CHECK(id_circuito LIKE ('[A-Z][A-Z]-[0-9][0-9]')),
```

Para ello, con el comando Alter table tb_circuitos drop constraint ck_tbcir_idcir; eliminamos la constraint y así podemos seguir con el ejercicio.
Además nos aseguramos que se ha borrado con el comando exec sp_helpconstraint [nombre_tabla].

```
exec sp_helpconstraint tb_circuitos;
```

100 %

Resultados | Mensajes

	Object Name
1	tb_circuitos

	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	FOREIGN KEY	fk_tbcir_tbdir_codcir	No Action	No Action	Enabled	Is_For_Replication	cod_direccion
2							REFERENCES bd_f1_sergio_ivan_raul.dbo.tb_direcci...
3	PRIMARY KEY (clustered)	pk_tbcir_idcir	(n/a)	(n/a)	(n/a)	(n/a)	id_circuito
4	UNIQUE (non-clustered)	uq_tbcir_idcir	(n/a)	(n/a)	(n/a)	(n/a)	id_circuito

	Table is referenced by foreign key
1	bd_f1_sergio_ivan_raul.dbo.tb_piloto_escuderia_...

Primeramente y lo que he hecho ha sido crear una función para decir cómo quiero que sea la id del circuito [subida a trigger1_sergio.txt]

```
CREATE FUNCTION dbo.validar_formato_id_circuito(@id_circuito VARCHAR(5))
RETURNS INT
AS
BEGIN
    DECLARE @validacion INT = 0;

    IF @id_circuito NOT LIKE '[A-Z][A-Z]-[0-9][0-9]'
    BEGIN
        SET @validacion = 1;
    END;

    RETURN @validacion;
END;
```

En resumidas cuentas, la función define que dbo… tomará id_circuito como parámetro de entrada y que será de tipo varchar(5) y queremos que nos retorne un int para verificar que se está cumpliendo o no con la condición, inicializando validación en 0 y entramos en el bucle IF.
Mientras la id_circuito no sea como queremos que sea, validacion cambiará su valor a 1 y se mantendrá en el bucle con el return, que al volver transformará de nuevo su valor a 0 y volverá a comprobar.
En caso de que id_circuito cumpla y sea [A-Z][A-Z]-[0-9][0-9] se mantendrá como 0 el valor de validación y por ende finalizará el programa.
Al ejecutar la función vemos que funciona correctamente y se ejecuta:

```sql
CREATE FUNCTION dbo.validar_formato_id_circuito(@id_circuito VARCHAR(5))
RETURNS INT
AS
BEGIN
    DECLARE @validacion INT = 0;

    IF @id_circuito NOT LIKE '[A-Z][A-Z]-[0-9][0-9]'
    BEGIN
        SET @validacion = 1;
    END;

    RETURN @validacion;
END;
```

0 %

Mensajes

```
Los comandos se han completado correctamente.

Hora de finalización: 2024-05-15T11:16:08.8939193+02:00
```

Ahora pasamos a los triggers

En un principio, cree únicamente uno ya que pensé que sería suficiente, pero al ejecutarlo vi que solo funcionaba cuando el campo que contiene el error estaba por delante del resto de inserts, si este estaba detrás de uno o el último, ejecutaba todos pese a que alguno tuviera un error.

Vamos con el primero:

El script está en el mismo documento de la función, por lo que no pasaré por aquí el script.

```sql
CREATE TRIGGER trigger_simple_validar_formato_id_circuito
ON tb_circuitos
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @id_circuito VARCHAR(5);
    SELECT @id_circuito = id_circuito FROM inserted;
    IF dbo.validar_formato_id_circuito(@id_circuito) = 1
    BEGIN
        RAISERROR ('El ID del circuito debe tener el formato correcto: XX-99', 16, 1);
        ROLLBACK TRANSACTION;
    END;
END;
```

100 %

Mensajes

```
Los comandos se han completado correctamente.

Hora de finalización: 2024-05-15T11:18:37.2351659+02:00
```

Se ejecuta sin ningún problema, ahora vamos a comprobar que realmente si está funcionando, para ello he ido a mi tabla de inserción y voy a borrar con delete from tb_circuitos todas las inserciones que tuviera hasta el momento.

```
select * from tb_circuitos
```

100 %

Resultados | Mensajes

id_circuito	nombre_circuito	cod_direccion	num_vueltas	km_circuitos	vuelta_rapida	num_aforo

```
INSERT INTO tb_circuitos
VALUES
    ('BR24', 'Brasil', 5555, 74, 2.80, '00:01:01.112', 9712),
    ('ES-24', 'Montmeló', 1111, 66, 4.67, '00:01:27.189', 50000),
    ('BE-24', 'Spa Francorchamps', 2222, 47, 7.23, '00:01:48.984', 45000),
    ('IT-24', 'Monza', 3333, 72, 2.67, '00:01:11.175', 76000),
    ('LV-24', 'Las Vegas', 5555, 71, 4.10, '00:01:50.202', 35000),
    ('AS-24', 'Albert Park', 4444, 53, 3.14, '00:01:34.112', 88711);
```

100 %

Mensajes

```
Mensaje 50000, nivel 16, estado 1, procedimiento trigger_simple_validar_formato_id_circuito, línea 11 [línea de inicio de lote 26]
El ID del circuito debe tener el formato correcto: XX-99
Mens. 3609, Nivel 16, Estado 1, Línea 27
La transacción terminó en el desencadenador. Se anuló el lote.

Hora de finalización: 2024-05-15T11:20:47.8206821+02:00
```

Al insertar como primer campo una ID errónea, nos salta el error con el mensaje de cómo debe ser el formato correcto y además no introduce ningún campo.
Aunque si lo movemos a otro lado y ejecutamos de nuevo, vemos que efectivamente se crean las inserciones:

```sql
INSERT INTO tb_circuitos
VALUES
    ('ES-24', 'Montmeló', 1111, 66, 4.67, '00:01:27.189', 50000),
    ('BE-24', 'Spa Francorchamps', 2222, 47, 7.23, '00:01:48.984', 45000),
    ('BR24', 'Brasil', 5555, 74, 2.80, '00:01:01.112', 9712),
    ('IT-24', 'Monza', 3333, 72, 2.67, '00:01:11.175', 76000),
    ('LV-24', 'Las Vegas', 5555, 71, 4.10, '00:01:50.202', 35000),
    ('AS-24', 'Albert Park', 4444, 53, 3.14, '00:01:34.112', 88711);

    select * from tb_circuitos
```

Resultados | Mensajes

id_circuito	nombre_circuito	cod_direccion	num_vueltas	km_circuitos	vuelta_rapida	num_aforo
AS-24	Albert Park	4444	53	3.140	00:01:34.1120000	88711
BE-24	Spa Francorchamps	2222	47	7.230	00:01:48.9840000	45000
BR24	Brasil	5555	74	2.800	00:01:01.1120000	9712
ES-24	Montmeló	1111	66	4.670	00:01:27.1890000	50000
IT-24	Monza	3333	72	2.670	00:01:11.1750000	76000
LV-24	Las Vegas	5555	71	4.100	00:01:50.2020000	35000

Para ello, he tenido que crear otro trigger el cual verifique dato a dato que realmente se está introduciendo los campos correctamente, también está subido al documento .txt.
Volvemos a borrar los inserts para que funcione correctamente.

```sql
CREATE OR ALTER TRIGGER trigger_validar_formato_id_circuito
ON tb_circuitos
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @id_circuito VARCHAR(5);
    DECLARE @error_count INT = 0;
    DECLARE @row_count INT = (SELECT COUNT(*) FROM inserted);
    DECLARE @current_row INT = 1;

    WHILE @current_row <= @row_count
    BEGIN
        SELECT @id_circuito = id_circuito FROM (
            SELECT *, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS row_num FROM inserted
        ) AS temp WHERE row_num = @current_row;
        IF dbo.validar_formato_id_circuito(@id_circuito) = 1
        BEGIN
            RAISERROR ('El ID del circuito debe tener el formato correcto: XX-99', 16, 1);
            SET @error_count = @error_count + 1;
        END;
        SET @current_row = @current_row + 1;
    END;
    IF @error_count > 0
    BEGIN
        ROLLBACK TRANSACTION;
    END;
END;
```

0 %

Mensajes

```
Los comandos se han completado correctamente.

Hora de finalización: 2024-05-15T11:24:18.8307339+02:00
```

Vemos que se ejecuta correctamente y ahora, si vamos de nuevo a la inserción y ejecutamos tal y como hemos dejado la inserción

```sql
INSERT INTO tb_circuitos
VALUES
    ('ES-24', 'Montmeló', 1111, 66, 4.67, '00:01:27.189', 50000),
    ('BE-24', 'Spa Francorchamps', 2222, 47, 7.23, '00:01:48.984', 45000),
    ('BR24', 'Brasil', 5555, 74, 2.80, '00:01:01.112', 9712),
    ('IT-24', 'Monza', 3333, 72, 2.67, '00:01:11.175', 76000),
    ('LV-24', 'Las Vegas', 5555, 71, 4.10, '00:01:50.202', 35000),
    ('AS-24', 'Albert Park', 4444, 53, 3.14, '00:01:34.112', 88711);
```

0 %

Mensajes

```
Mensaje 50000, nivel 16, estado 1, procedimiento trigger_validar_formato_id_circuito, línea 19 [línea de inicio de lote 26]
El ID del circuito debe tener el formato correcto: XX-99
Mens. 3609, Nivel 16, Estado 1, Línea 27
La transacción terminó en el desencadenador. Se anuló el lote.

Hora de finalización: 2024-05-15T11:24:54.9333183+02:00
```

Observamos que pese a que está en una posición más baja, al llegar y comprobar que está introducido de forma errónea, salta el mensaje de que hay un error y no se crea ninguna inserción.

```sql
INSERT INTO tb_circuitos
VALUES
    ('ES-24', 'Montmeló', 1111, 66, 4.67, '00:01:27.189', 50000),
    ('BE-24', 'Spa Francorchamps', 2222, 47, 7.23, '00:01:48.984', 45000),
    ('BR24', 'Brasil', 5555, 74, 2.80, '00:01:01.112', 9712),
    ('IT-24', 'Monza', 3333, 72, 2.67, '00:01:11.175', 76000),
    ('LV-24', 'Las Vegas', 5555, 71, 4.10, '00:01:50.202', 35000),
    ('AS-24', 'Albert Park', 4444, 53, 3.14, '00:01:34.112', 88711);

    select * from tb_circuitos
    delete from tb_circuitos
```

100 %

Resultados | Mensajes

id_circuito	nombre_circuito	cod_direccion	num_vueltas	km_circuitos	vuelta_rapida	num_aforo

Ejercicio 2

Rellenar un campo calculado o derivado (calculado a partir del valor de otros o de contar registros o valores de alguna tabla).

Para este ejercicio, he decidido que sería una buena opción calcular sobre el campo num_aforo de la tabla tb_circuitos cuál ha sido el porcentaje de ocupación de ese aforo, para saber a ciencia cierta el % de ocupación y saber que circuito ha sido al que más gente ha acudido.

Para ello primero debemos crear un nuevo campo que sea el numero de aforo que ha tenido el circuito de tipo numérico,

```sql
ALTER TABLE tb_circuitos
ALTER COLUMN num_aforo_ocupado NUMERIC(10);
```

100 %

Mensajes

```
Los comandos se han completado correctamente.

Hora de finalización: 2024-05-15T12:16:42.6680589+02:00
```

con alter table tb_circuitos add num_aforo_ocupado NUMERIC(10) lo crearemos.

Pero solo con eso no sirve, ya que necesitamos un tercer campo que muestre el porcentaje total. Con alter table tb_circuitos add porcentaje_ocupacion DECIMAL(5,2); creamos el nuevo campo que guardará el cálculo del número de aforo máximo, el aforo que ha acudido al circuito y el porcentaje de ocupación.

```sql
create table tb_circuitos(
id_circuito varchar(5) not null,
nombre_circuito varchar(50) not null,
cod_direccion numeric(10) NOT NULL,
num_vueltas numeric(2) not null,
km_circuitos decimal(4,3) not null,
vuelta_rapida time not null,
num_aforo numeric(5)  not null,

constraint pk_tbcir_idcir primary key (id_circuito),
constraint uq_tbcir_idcir unique(id_circuito),
constraint fk_tbcir_tbdir_codcir foreign key (cod_direccion) references tb_direccion (cod_direccion),
constraint ck_tbcir_idcir CHECK(id_circuito LIKE ('[A-Z][A-Z]-[0-9][0-9]')),
);

ALTER TABLE tb_circuitos
ADD porcentaje_ocupacion DECIMAL(5,2);
```

100 %

Mensajes

```
Los comandos se han completado correctamente.

Hora de finalización: 2024-05-15T11:58:24.5864065+02:00
```

Ahora creamos el trigger:

```sql
CREATE OR ALTER TRIGGER trigger_actualizar_porcentaje_ocupacion
ON tb_circuitos
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE c
    SET porcentaje_ocupacion = (CONVERT(DECIMAL(7,2), c.num_aforo_ocupado) / CONVERT(DECIMAL(7,2), c.num_aforo)) * 100
    FROM tb_circuitos c
    INNER JOIN inserted i ON c.id_circuito = i.id_circuito;
END;

drop trigger trigger_actualizar_porcentaje_ocupacion
```

100 %

Mensajes

```
Los comandos se han completado correctamente.

Hora de finalización: 2024-05-15T12:11:31.2633744+02:00
```

Se crea correctamente y ahora vamos a poner campos para comprobar que funciona correctamente.

Ahora vamos a ver que funciona correctamente, para ello vamos a hacer un update de algunos campos y añadir de forma manual el num_aforo_ocupado

```sql
INSERT INTO tb_circuitos
VALUES
    ('ES-24', 'Montmeló', 1111, 66, 4.67, '00:01:27.189', 50000),
    ('BE-24', 'Spa Francorchamps', 2222, 47, 7.23, '00:01:48.984', 45000),
    ('IT-24', 'Monza', 3333, 72, 2.67, '00:01:11.175', 76000),
    ('LV-24', 'Las Vegas', 4444, 71, 4.10, '00:01:50.202', 35000),
    ('AS-24', 'Albert Park', 4444, 53, 3.14, '00:01:34.112', 88711);

    update tb_circuitos
    set num_aforo_ocupado = 45000
    where id_circuito IN ('BE-24', 'ES-24', 'IT-24');
```

100 %

Mensajes

```
(3 filas afectadas)

Hora de finalización: 2024-05-15T12:13:50.1690813+02:00
```

He querido modificar los campos con ID BE-24 ES-24 e IT-24 para que el num de aforo ocupado sea de 45000 personas. Efectivamente se ejecuta al y se modifican las columnas.

Como resultado muestra lo siguiente:

```sql
select * from tb_circuitos
```

100 %

Resultados | Mensajes

	id_circuito	nombre_circuito	cod_direccion	num_vueltas	km_circuitos	vuelta_rapida	num_aforo	num_aforo_ocupado	porcentaje_ocupacion
1	AS-24	Albert Park	4444	53	3.140	00:01:34.1120000	88711	NULL	NULL
2	BE-24	Spa Francorchamps	2222	47	7.230	00:01:48.9840000	45000	45000	100.00
3	BR-24	Brasil	2222	74	2.800	00:01:01.1120000	9712	NULL	NULL
4	ES-24	Montmeló	1111	66	4.670	00:01:27.1890000	50000	45000	90.00
5	IT-24	Monza	3333	72	2.670	00:01:11.1750000	76000	45000	59.21
6	LV-24	Las Vegas	4444	71	4.100	00:01:50.2020000	35000	NULL	NULL

Como podemos comprobar, ahora calcula de forma automática el porcentaje de aforo que tenemos, por ejemplo, en Spa Francochamps vemos que el aforo ha sido del 100%, mientras que en Montmeló ha sido del 90% y en Monza del 59,21%.