

Taller 4

Fecha: Septiembre de 2025

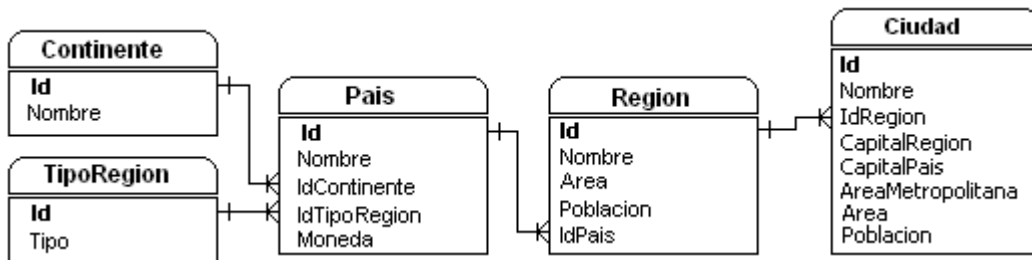
Sistema de habilidades a evaluar: Usar los conceptos del modelo relacional y de un lenguaje de programación de motor de bases de datos (*Transact-SQL*) para la programación en los motores de bases de datos.

NOTAS:

- Este taller se debe hacer con carácter evaluativo. Representa en total una calificación de 15%.
- Se entregan ejercicios resueltos como ejemplo para el desarrollo de los demás.

Basado en cada uno de los siguientes modelos relacionales, escribir los códigos necesarios para resolver los problemas se enuncian, utilizando el lenguaje de programación del motor *MS SQL SERVER*.

1. El modelo relacional de una base de datos que registra la división política mundial es el siguiente:



Además de las restricciones de las claves principales foráneas, se tienen las siguientes:

- a. Una región no puede tener sino una ciudad capital
- b. Un país no puede tener sino una ciudad capital que no necesariamente tiene que ser capital de región.

R/

- a. El código del TRIGGER que resuelve el problema sería el siguiente:

```
CREATE TRIGGER tActualizarCapitalRegion
ON Ciudad
FOR INSERT, UPDATE
AS BEGIN
    --Validar que se esta actualizando una capital de region
    IF EXISTS(SELECT * FROM Inserted WHERE CapitalRegion=1)
    BEGIN
        -- Verificar si la inserción o actualización causaría que
        haya más de una capital de región
        IF EXISTS(SELECT 1
                    FROM Inserted I
```

```

I.IdRegion=C.IdRegion JOIN Ciudad C ON
WHERE I.CapitalRegion=1 AND
C.CapitalRegion=1 AND C.Id<>I.Id
GROUP BY I.IdRegion
HAVING COUNT(*) > 1)
BEGIN
RAISERROR('No se acepta más de una capital por
región', 16, 1)
ROLLBACK TRANSACTION
END

-- Si se está estableciendo una ciudad como capital,
asegurarse de que las demás no lo sean
UPDATE Ciudad
SET CapitalRegion=0
FROM Ciudad C
JOIN Inserted I ON C.IdRegion=I.IdRegion
AND C.Id<>I.Id

END
END
GO

```

Tener en cuenta que el objeto **inserted** contiene la información de los registros que se están actualizando.

- b. El código del TRIGGER que resuelve el problema sería el siguiente:

```

CREATE TRIGGER tActualizarCapitalPais
ON Ciudad
FOR INSERT, UPDATE
AS
BEGIN
-- Evitar ejecución recursiva del trigger
IF TRIGGER_NESTLEVEL() > 1
RETURN;

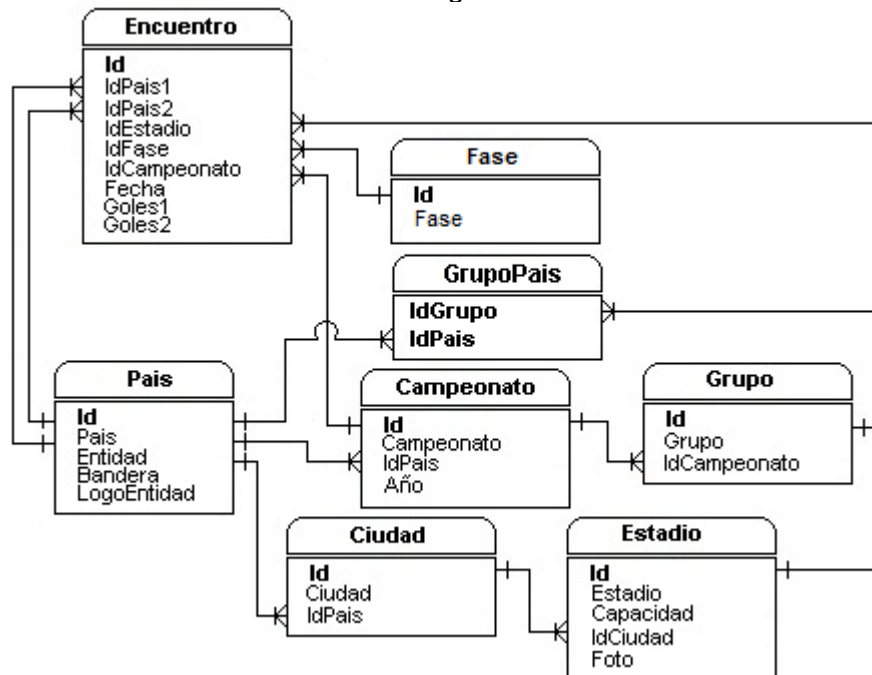
-- Asegurarnos de que solo quede una ciudad como CapitalPais = 1
por cada Pais
WITH UltimaCapital AS (SELECT C.Id, R.IdPais
FROM Inserted I
JOIN Ciudad C ON C.Id=I.Id
JOIN Region R ON
C.IdRegion=R.Id
WHERE I.CapitalPais=1)

```

```

UPDATE C
  SET C.CapitalPais=CASE WHEN C.Id=U.Id THEN 1
                        ELSE 0
                        END
FROM Ciudad C
      JOIN Region R ON C.IdRegion=R.Id
      JOIN UltimaCapital U ON U.IdPais = R.IdPais
END;
GO;
  
```

2. El modelo relacional de una base de datos para registro de campeonatos de fútbol de selecciones nacionales es el siguiente:



Crear un trigger que valide que un País no figure en más de un Grupo de un Campeonato.

R/

```

CREATE TRIGGER tActualizarGrupoPais
ON GrupoPais
FOR INSERT, UPDATE
AS
BEGIN
  SET NOCOUNT ON;

  -- Verificar si algún país de la inserción/actualización
  -- ya está en otro grupo del mismo campeonato
  IF EXISTS(
    SELECT 1
  
```

```
FROM Inserted I
JOIN Grupo GNuevo ON GNuevo.Id = I.IdGrupo
JOIN GrupoPais GP ON GP.IdPais = I.IdPais
JOIN Grupo GExistente ON GExistente.Id = GP.IdGrupo
WHERE GNuevo.IdCampeonato = GExistente.IdCampeonato
      AND GNuevo.Id <> GExistente.Id
)
BEGIN
    -- Cancelar la instrucción y lanzar error
    RAISERROR('Un país no puede pertenecer a más de un grupo
en el mismo campeonato.', 16, 1);
    ROLLBACK TRANSACTION;
    RETURN;
END
END
GO
```

3. Basado en la anterior base de datos, validar que, en un mismo campeonato y fase, un encuentro entre dos países no puede repetirse. (Ejemplo: si ya existe Brasil vs Alemania en fase de grupos, no se puede volver a insertar ese mismo partido en esa fase del mismo campeonato)