

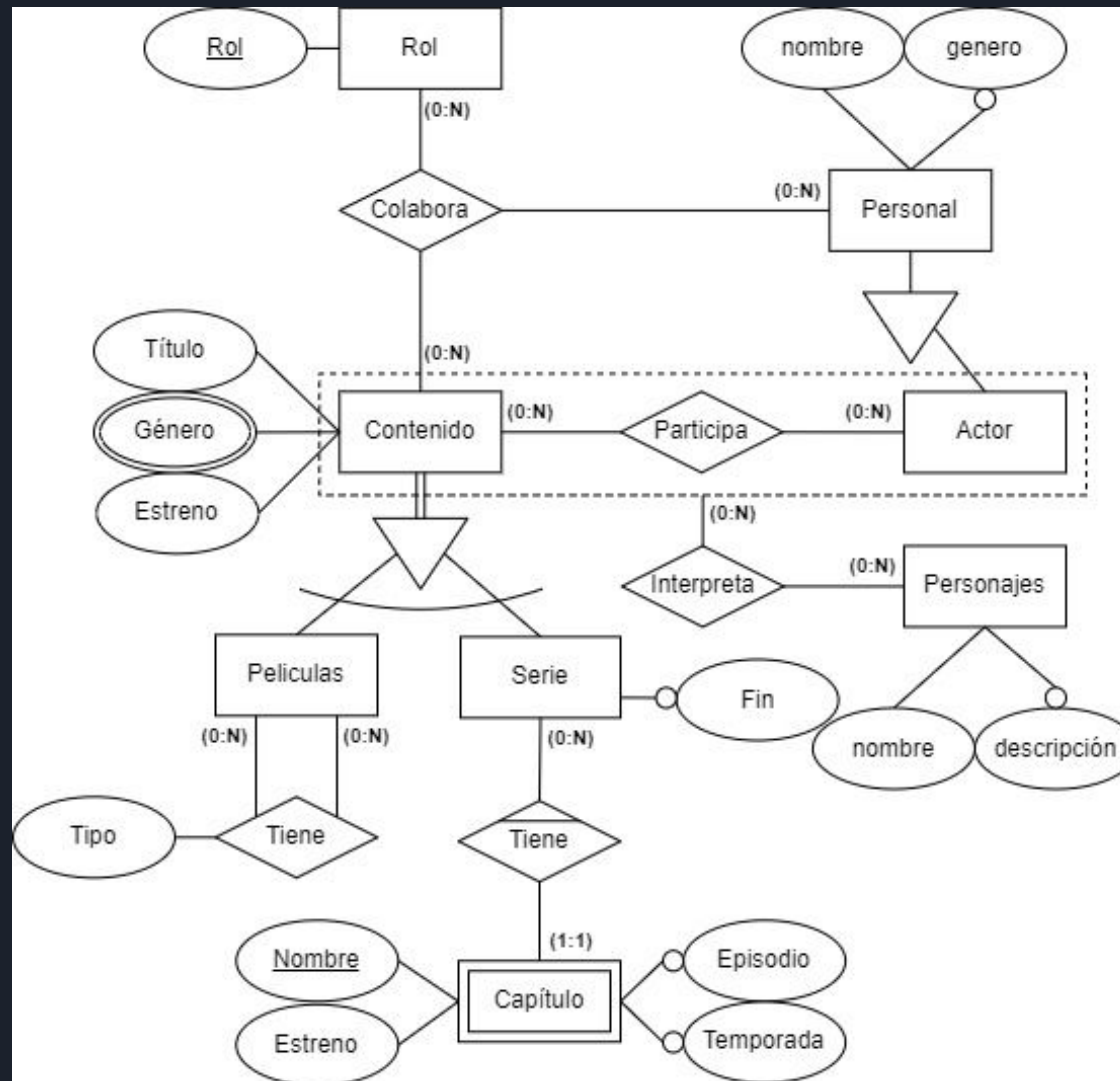


BBDD - Práctica 2

Héctor Toral
Francisco Javier Pizarro
Pablo López

1/04/2022

Modelo E/R





Alternativas a nuestro modelo:

- El atributo multivaluado género, podría haber sido modelado como una relación N:M con contenido
- Atributo multivaluado roles en vez de la ternaria colabora.
- Agregación contenido, actores en vez de otra agregación como contenido personajes o una ternaria.

Modelo relacional

Series = (id: entero; **fin**: entero);
verificar que fin >= 1850;

Capítulos = (id: entero; **nombre**: cadena(100); **estreno**: entero, no nulo; **temporada**: entero; **episodio**: entero);
verificar que: estreno >= 1850; temporada >= 1; episodio >= 1;

Películas = (id: entero);

Relacion_películas = (id_película_a: entero; id_película_b: entero; **tipo**: cadena(50), no nulo);

Géneros = (**nombre**: cadena(100); id_contenido: entero);

Contenido = (id: entero; **título**: cadena(150), no nulo; **estreno**: entero, no nulo);
verificar que estreno >= 1850;

Roles = (rol: cadena(100));

Colabora = (id_personal: entero; id_contenido: entero; rol: cadena(100));

Interpreta = (id_actor: entero; id_contenido: entero; id_personaje: entero);

Participa = (id_actor: entero; id_contenido: entero);

Personal = (id: entero; **nombre**: cadena(100) no nulo; **genero**: cadena(100));

Actores = (id: entero);

Personajes = (id: entero; **nombre**: cadena(100), no nulo; **descripción**: cadena(300));



Dudas pasando del modelo E/R al modelo relacional

- Tipo de traducción de la especialización/generalización

Porcentaje de películas con hasta 3 actores o actrices

```
SELECT ROUND(N.num_conteidos_limite_actores / D.num_contenidos * 100, 2)||'%'
  AS porcentaje
FROM (
  -- Número de contenidos con hasta 3 actores/actrices
  SELECT COUNT(*) AS num_conteidos_limite_actores
  FROM (
    -- Devuelve tuplas en las que el num_actores es <= 3; num_actores = count(*)
    SELECT id_contenido
    FROM participa
    GROUP BY id_contenido
    HAVING COUNT(*) <= 3
  )
) N, (
  -- Devuelve el número total de contenidos
  SELECT COUNT(*) AS num_contenidos
  FROM contenido
) D;
```

Título de las películas y series de terror en las que el mismo personaje ha sido interpretado por al menos 50 diferentes actores o actrices, junto con el nombre del personaje y el número de actores distintos que lo han interpretado

```
SELECT N.titulo, P.nombre, N.num_actores
FROM personajes P, (
  -- Devuelve tuplas (id, título, num_actores, id personaje) en las que el contenido es de terror y el número de veces que un personaje
  -- es interpretado dentro del mismo contenido es de al menos 50.
  SELECT A.id, A.titulo, B.num_actores, B.id_personaje
  FROM (
    -- Devuelve una lista de tuplas (id, título) de aquellos contenidos que
    -- pertenecen al género de terror.
    SELECT C.id, C.titulo
    FROM contenido C, generos G
    WHERE C.id = G.id_contenido AND G.nombre = 'terror'
  ) A, (
    -- Devuelve tuplas como (número actores, id_contenido, id_personaje) en las que el número de actores para interpretar
    -- un personaje dentro de un mismo contenido (película/serie) es al menos de 50.
    SELECT COUNT(*) AS num_actores, id_contenido, id_personaje
    FROM interpreta I
    GROUP BY id_contenido, id_personaje
    HAVING COUNT(*) > 49
  ) B
  WHERE A.id = B.id_contenido
) N
WHERE P.id = N.id_personaje;
```

```
SELECT nombre
```

```
FROM personal P, (
```

Actores que solo han participado en una película de la década de los 90 que forma parte de una saga de al menos tres películas

```
SELECT id_actor FROM participa P , (
```

```
-- Selecciona de la lista de películas que pertenecen a una saga de al menos 3 películas, las que se estrenaron en la década de los 90
```

```
SELECT DISTINCT S.id FROM contenido C, (
```

```
-- Saca la lista de sagas con al menos 3 películas
```

```
SELECT id_pelicula_a AS id
```

```
FROM (
```

```
SELECT * FROM relacion_peliculas WHERE tipo = 'follows'
```

```
) T1
```

```
GROUP BY id_pelicula_a
```

```
HAVING COUNT(*) >= 2
```

```
-- es 2 y no 3 ya que la primera película no se cuenta, se usa como "identificador" de saga
```

```
UNION ALL
```

```
-- Devuelve las películas que pertenecen a la lista de sagas siendo "saga" la primera película de la saga
```

```
SELECT id_pelicula_b AS id
```

```
FROM relacion_peliculas
```

```
WHERE id_pelicula_a IN (
```

```
SELECT id_pelicula_a FROM relacion_peliculas
```

```
WHERE tipo = 'follows'
```

```
)
```

```
) S
```

```
WHERE C.id = S.id AND 1990 <= estreno AND estreno < 2000
```

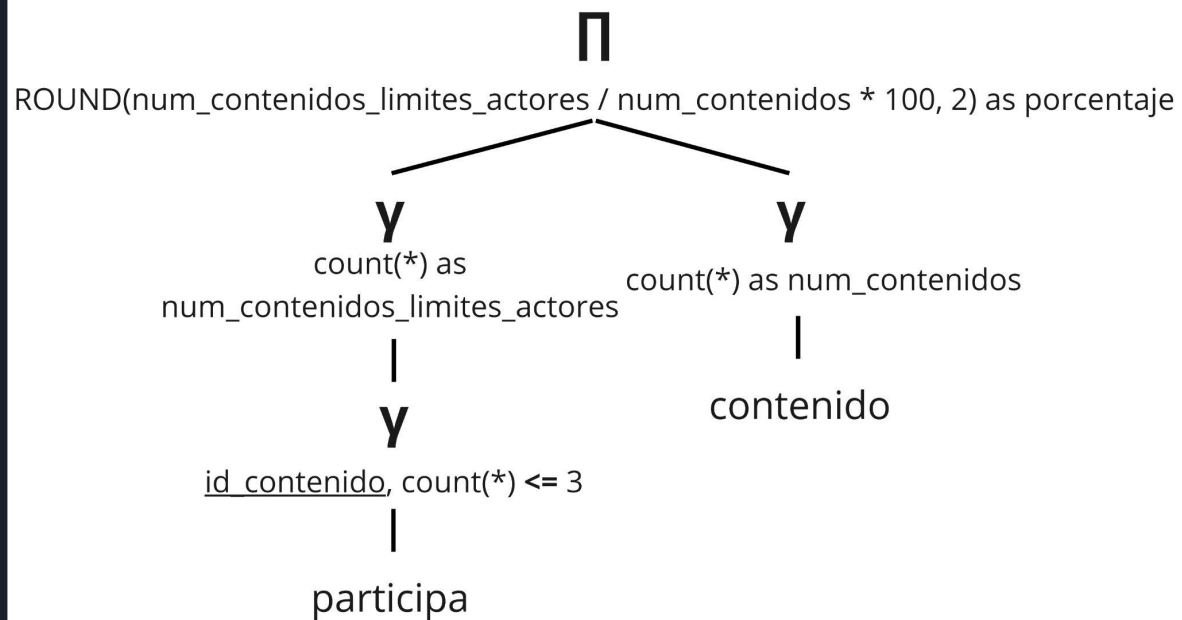
```
) S
```

```
WHERE P.id_contenido = S.id GROUP BY id_actor HAVING COUNT(*) = 1
```

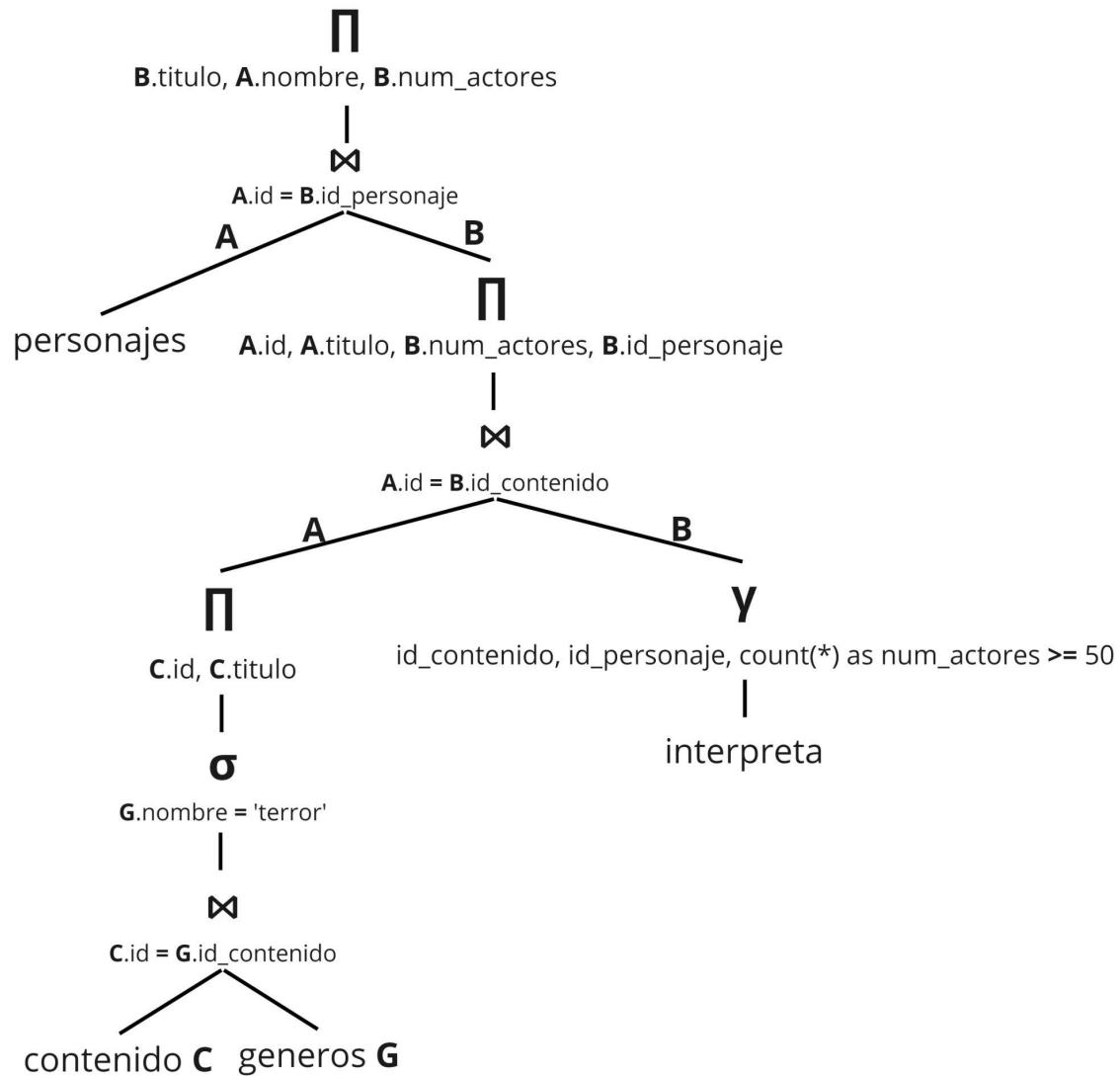
```
) A
```

```
WHERE P.id = A.id_actor;
```

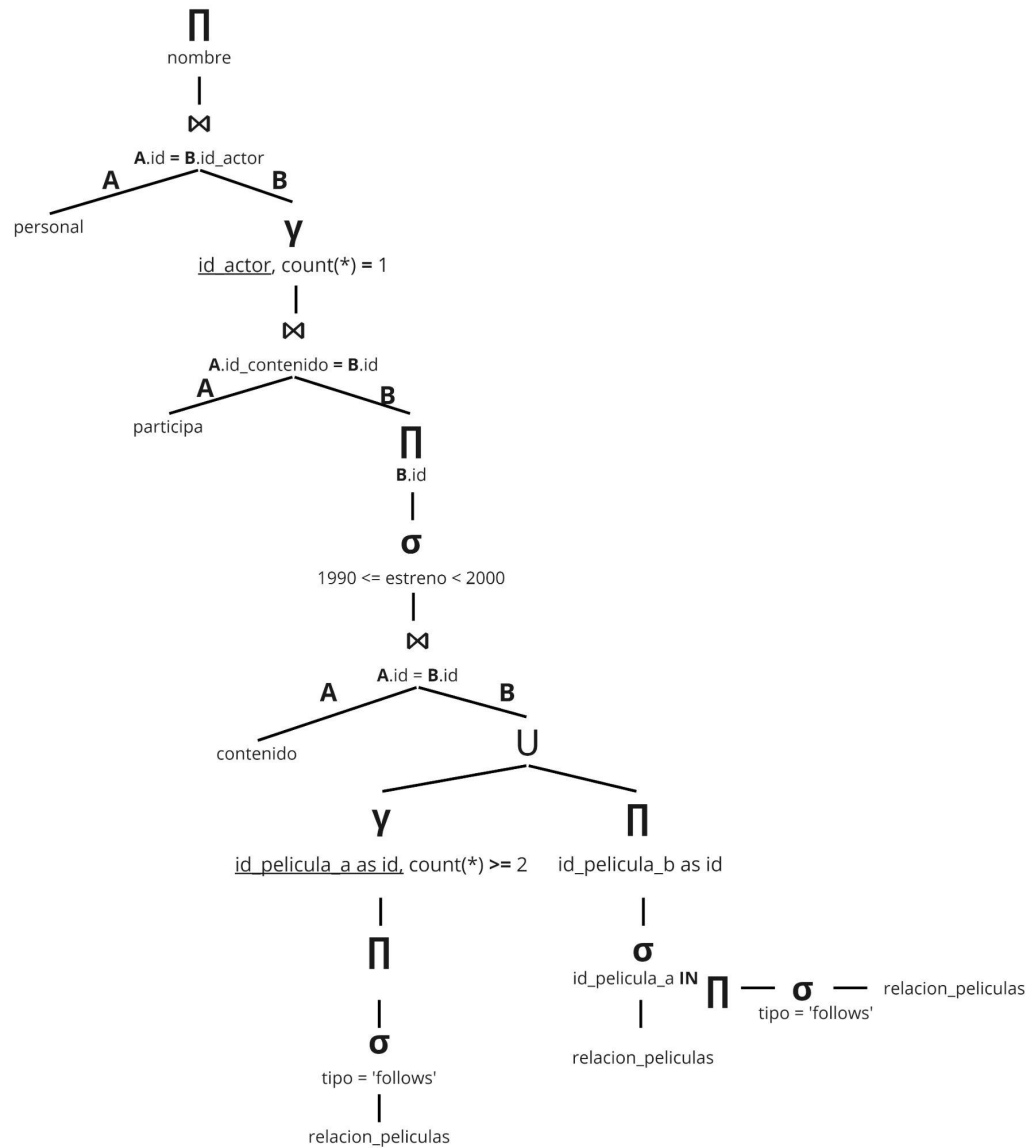

Árbol de la 1ª consulta

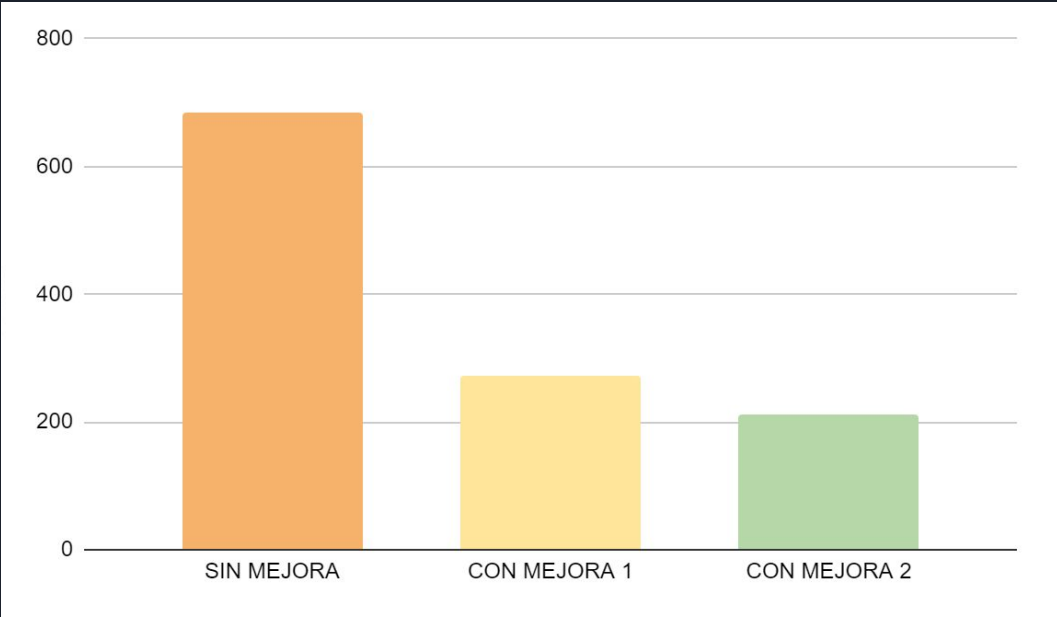
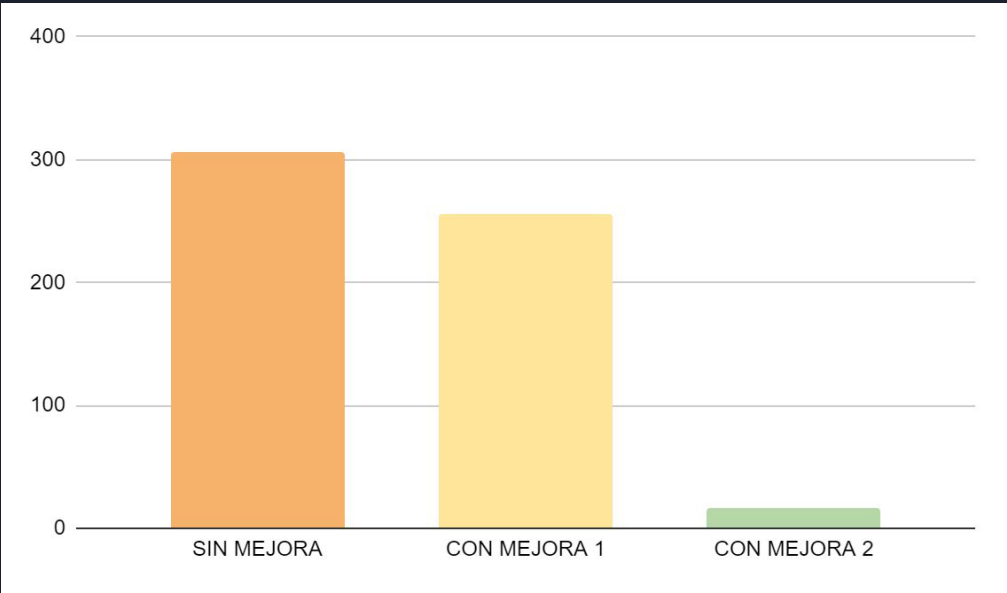
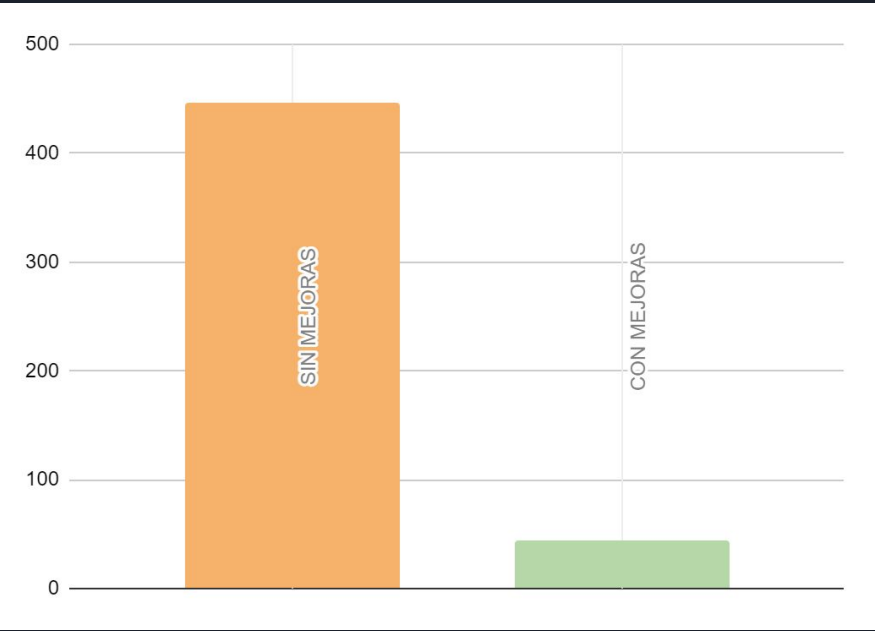


Árbol de la 2ª consulta



Árbol de la 3ª consulta





Trigger 1

```
CREATE OR REPLACE TRIGGER EXC_PELIS
BEFORE INSERT ON películas
FOR EACH ROW
DECLARE
    flag NUMBER;
BEGIN
    SELECT COUNT(*) INTO flag FROM series WHERE id = :NEW.id;
    IF flag >= 1 THEN
        RAISE_APPLICATION_ERROR (-20000, 'La fila insertada ya pertenece a
        la categoría de series.');
```

END IF;

END;

/

Trigger 2

```
CREATE OR REPLACE TRIGGER CLEAN_DATES
BEFORE INSERT ON series
FOR EACH ROW
DECLARE
    fechaEstr NUMBER;
BEGIN
    SELECT estreno INTO fechaEstr FROM contenido WHERE :NEW.id = id;
    IF :NEW.fin < fechaEstr THEN
        RAISE_APPLICATION_ERROR (-20002, 'La serie: ' || :NEW.id || ' tiene una
        fecha de finalización menor que la del estreno de la tabla contenido.
        Considera eliminar/corregir la inserción.');
```

END IF;

END;

/

Trigger 3

```
CREATE OR REPLACE TRIGGER INSERT_ACT
BEFORE INSERT ON actores
FOR EACH ROW
DECLARE
    flag NUMBER;
BEGIN
    SELECT COUNT(*) INTO flag
    FROM colabora
    WHERE id_personal = :NEW.id AND (rol = 'actor' or rol = 'actress');

    IF flag < 1 THEN
        RAISE_APPLICATION_ERROR (-20002, 'La persona insertada no es un actor.');
```

END IF;

```
END;
/
```