

BASES DE DATOS

2º Grado Ing. Informática

curso 2021 - 2022

PRÁCTICA 2 : Base de datos de cine

GRUPO 11

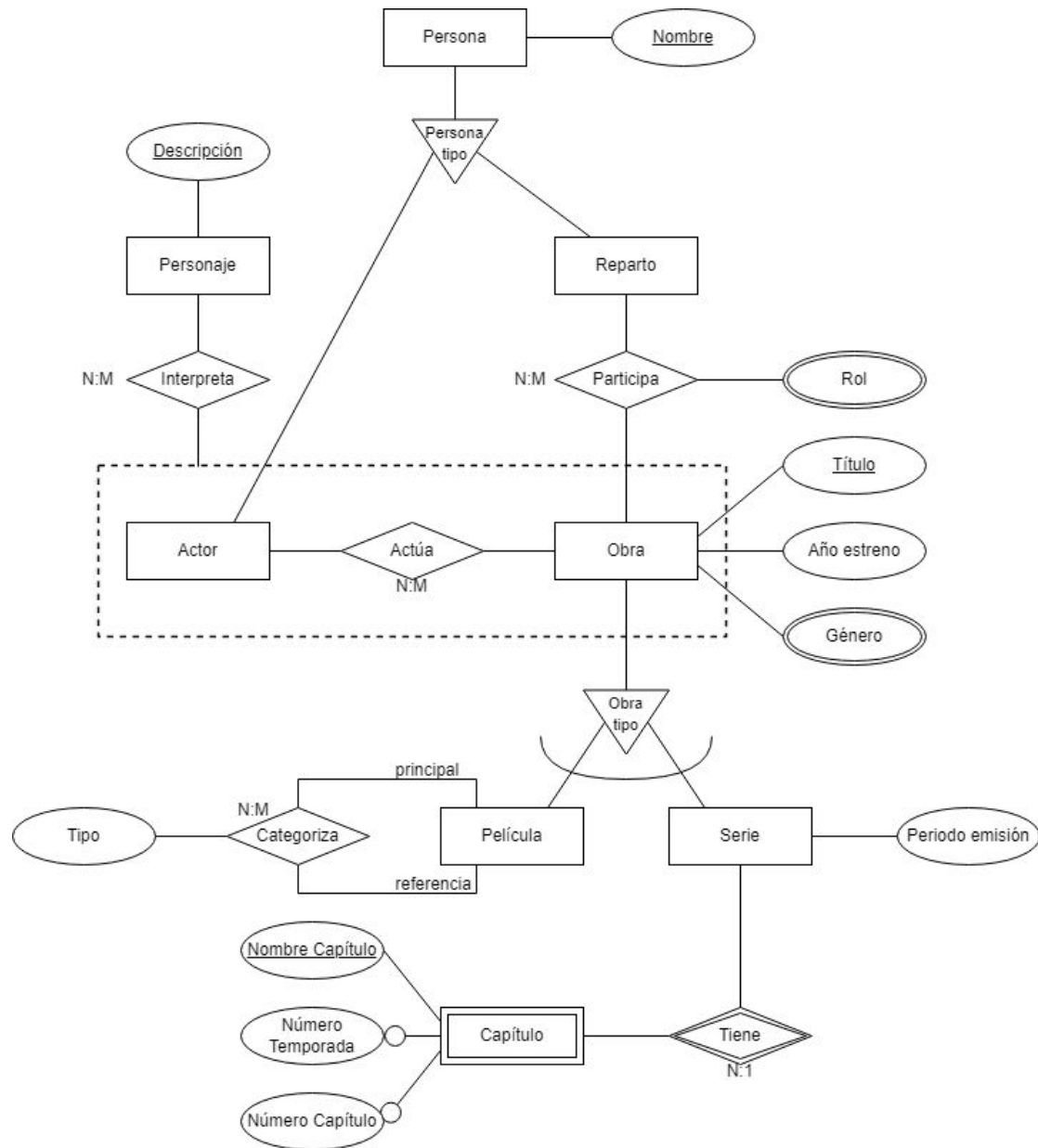
Cristian Andrei Selivanov Dobrisan [816456]

Dorian Boleslaw Wozniak [817570]

Álvaro Seral Gracia [819425]

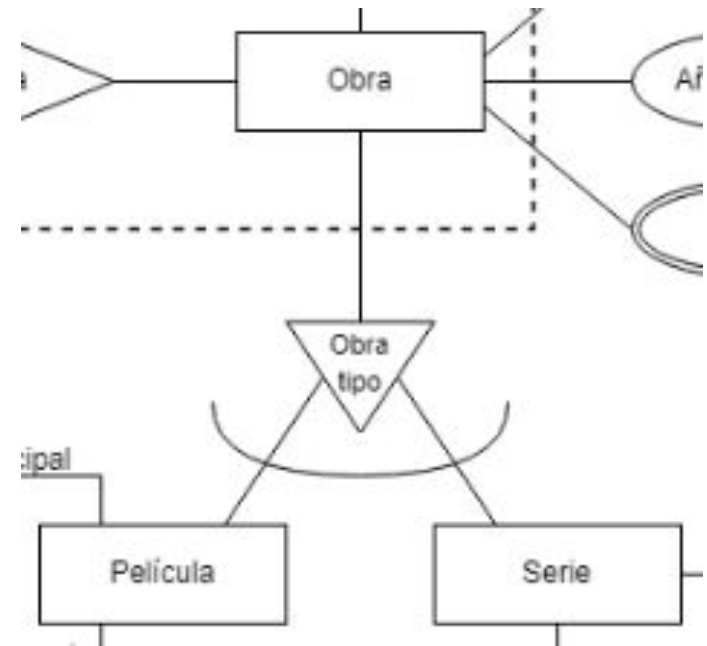
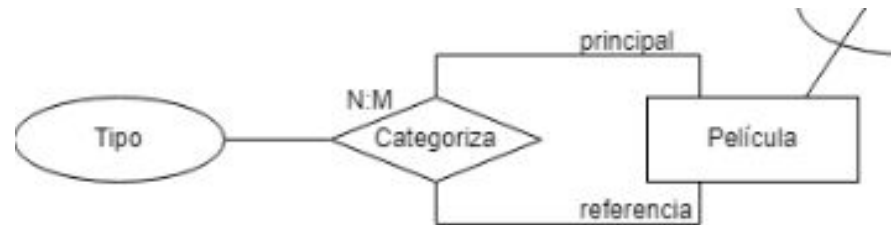
2 Mayo de 2022

DISEÑO ENTIDAD RELACIÓN



Consideraciones del modelo E/R

- El “Tipo” que categorizan las obras es excluyente.
(secuela, precuela...)
- No se dispone de participación total en entidades obra / persona.



MODELO RELACIONAL

Persona (

idPersona : tpNatural;
Nombre : tpNombre, NO NULO);

Personaje (

idPersonaje : tpNatural;
Descripción : tpNombre, NO NULO);

Obra (

idObra : tpNatural;
Título : tpObra, NO NULO;
Estreno : tpAño, NO NULO);

Tema (

Género : tpNombre;
clvObra : tpNatural, clave ajena de **Obra**);

Participa (

clvReparto : tpNatural, clave ajena de **Reparto**;
clvObra : tpNatural, clave ajena de **Obra**);

Labor (

Rol : tpNombre;
clvReparto, clvObra : tpNatural;
(clvReparto, clvObra) clave ajena de **Participa**);

Interpreta (

clvActor, clvObra : tpNatural;
clvPersonaje : tpNatural, clave ajena de **Personaje**;
(clvActor, clvObra) clave ajena de **Actúa**);

Reparto (

idReparto : tpNatural, clave ajena de **Persona**);

Actor (

idActor : tpNatural, clave ajena de **Persona**);

Actúa (

clvActor : tpNatural, clave ajena de **Actor**;
clvObra : tpNatural, clave ajena de **Obra**);

Película (

idPelicula : tpNatural, clave ajena de **Obra**);

Categoriza (

clvPrincipal, clvReferencia : clave ajena de **Película**;
Tipo : tpNombre, NO NULO);

Serie (

idSerie : tpNatural, clave ajena de **Obra**;
Periodo : tpAño, NO NULO);

Capítulo (

idCapitulo : tpNatural;
NombreC : tpNombre, NO NULO;
NumC, NumT : tpNatural;

Consideraciones del relacional

- Se han eliminado los atributos multivaluados para que cumpla la Primera Forma Normal.

Obra (

idObra : tpNatural;
Título : tpObra, NO NULO;
Estreno : tpAño, NO NULO;
Genero : tpNombre);

Obra (

idObra : tpNatural;
Título : tpObra, NO NULO;
Estreno : tpAño, NO NULO);

Tema (

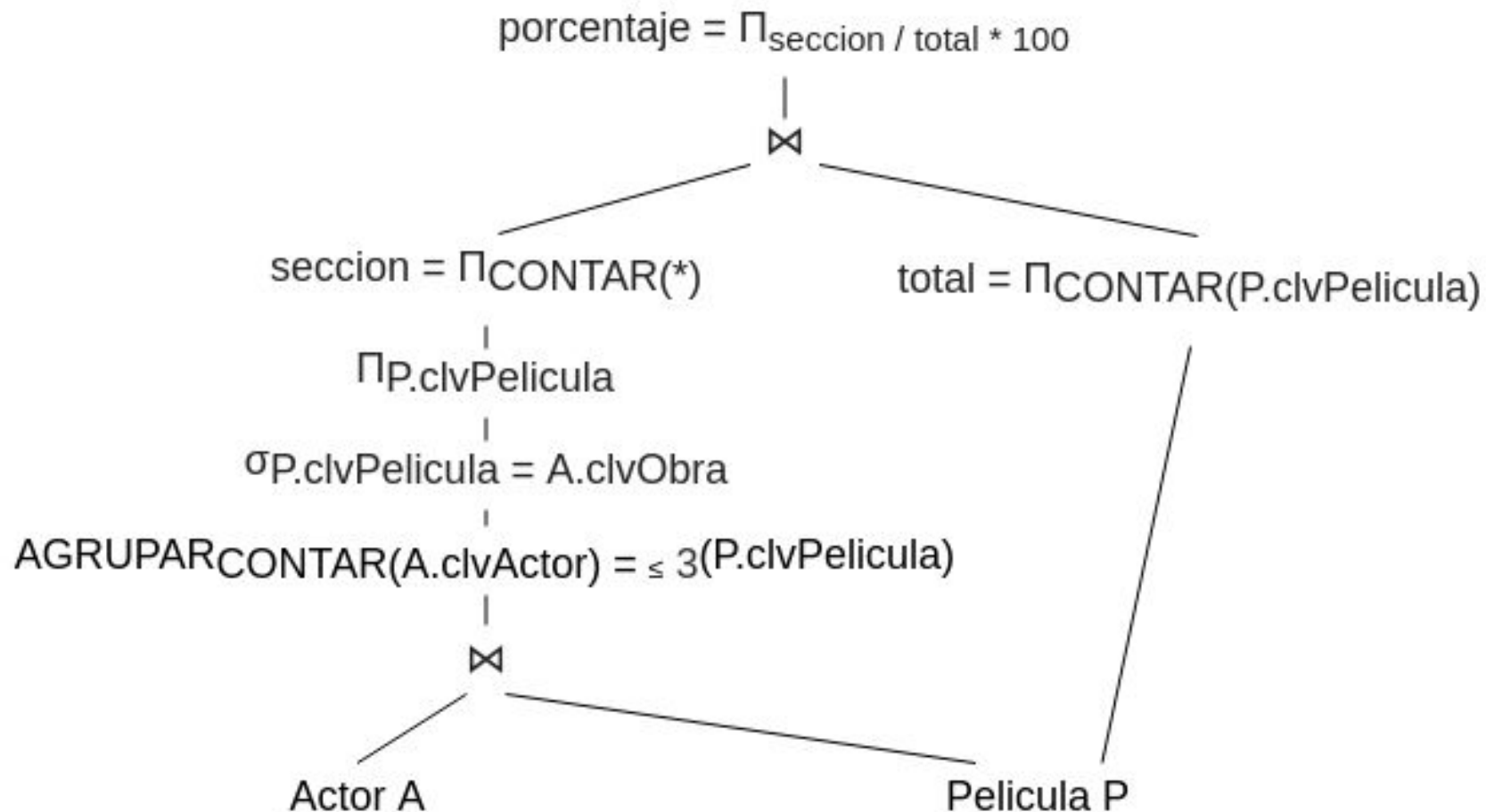
Género : tpNombre;
clvObra : tpNatural, clave ajena de **Obra**);

CONSULTA 1:

“Porcentaje de películas con hasta 3 actores o actrices”

```
SELECT seccion, total, (seccion/total) * '100' Porcentaje
FROM
(
    SELECT count(*)
    AS seccion
    FROM
    (
        SELECT DISTINCT P.clvPelicula
        FROM Pelicula P, Actua A
        WHERE P.clvPelicula=A.clvObra
        GROUP BY P.clvPelicula
        HAVING '3' >= count(A.clvActor)
    )
),
(
    SELECT count(DISTINCT P.clvPelicula)
    AS total
    FROM Pelicula P
);
```

CONSULTA 1: ÁRBOL SINTÁCTICO Y RESULTADO



SECCION	TOTAL	PORCENTAJE
583	3803	15,3300026

CONSULTA 2:

“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 DISTINCT

```
O.Titulo Titulo_Obra,  
Pj.Descripcion Descripcion_Personaje,  
count(I.clvActor) Numero_Actores
```

```
FROM Obra O, Personaje Pj, Interpreta I, Tema T
```

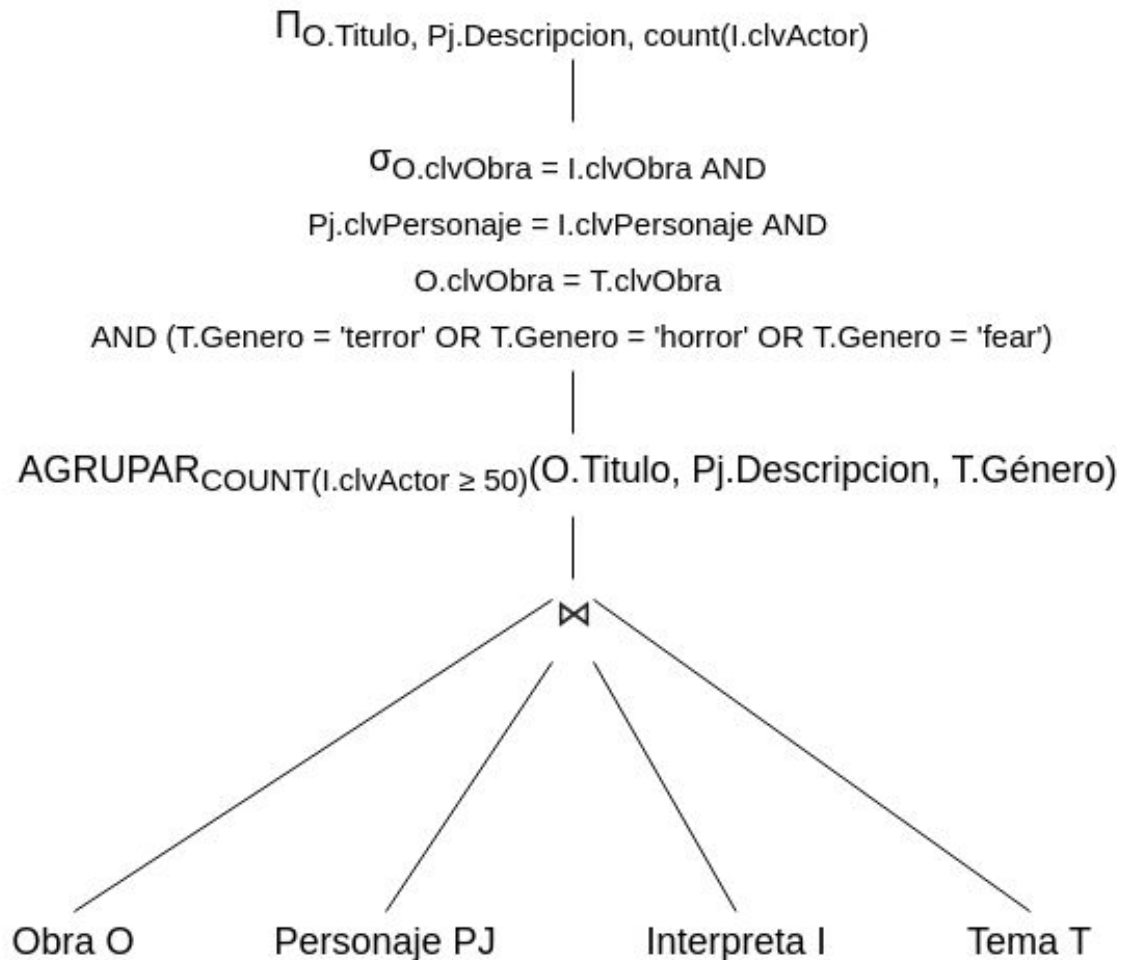
WHERE

```
O.clvObra = I.clvObra AND  
Pj.clvPersonaje = I.clvPersonaje AND  
O.clvObra = T.clvObra  
AND
```

```
(  
    T.Genero LIKE '%terror' OR  
    T.Genero LIKE '%horror%' OR  
    T.Genero LIKE '%fear%'  
)
```

```
GROUP BY O.Titulo, Pj.Descripcion, T.Genero  
HAVING '50' <= count(I.clvActor);
```


CONSULTA 2: ÁRBOL SINTÁCTICO Y RESULTADO



TITULO_OBRA	DESCRIPCION_PERSONAJE	NUMERO_ACTORES
Angustia	New cinema spectator	56

CONSULTA 3:

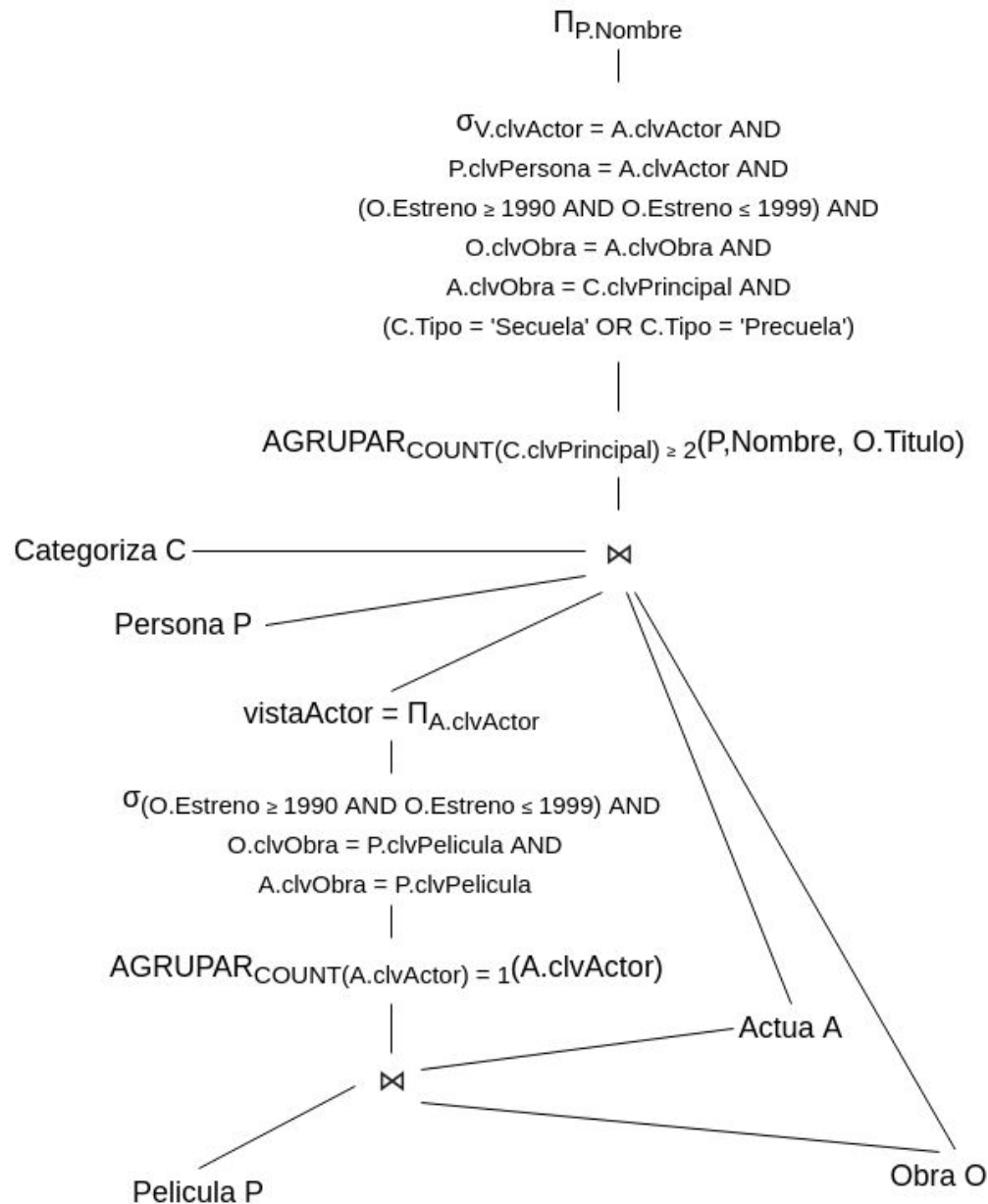
“Actores que solo han participado en una película de la década de los 90 que forma parte de una saga de al menos tres películas”

```
CREATE VIEW vistaActor AS
  SELECT A.clvActor
  FROM Pelicula P, Obra O, Actua A
  WHERE
    (O.Estreno BETWEEN '1990' AND '1999') AND
    O.clvObra = P.clvPelicula AND
    A.clvObra = P.clvPelicula
  GROUP BY A.clvActor
  HAVING '1' = count(A.clvActor);

SELECT P.Nombre
FROM Persona P, vistaActor V, Categoriza C, Actua A, Obra O
WHERE
  V.clvActor = A.clvActor AND
  P.clvPersona = A.clvActor AND
  (O.Estreno BETWEEN '1990' AND '1999') AND
  O.clvObra = A.clvObra AND
  A.clvObra = C.clvPrincipal AND
  C.Tipo IN ('follows', 'followed by')
GROUP BY P.Nombre, O.Titulo
HAVING '2' <= count(C.clvPrincipal);

DROP VIEW vistaPeli;
```

CONSULTA 3: ÁRBOL SINTÁCTICO Y RESULTADO



NOMBRE

 Lucas, Jorge R.
 Ortiz, Marcos
 Ponce, Ester
 Arraud, Mariano
 Bl??zquez, ??lvaro
 Gallo, Jes??s
 Mullen, Rory
 Ranz, Fernando
 Rodr??guez, Luis Miguel
 Garc??a, Palomo
 L??pez, Santos
 Dill, Marcus
 Pica, Antonio
 Mosquera, Carmen
 Guerrero, Francisco
 Ramo, Carlos
 Fontana, Rosa
 Quiroga, ??lvaro
 Molero Alfonso, Marta
 Urtado, Alberto

20 filas seleccionadas.

DISEÑO FÍSICO

CONSULTA 1

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	26	77 (6)	00:00:01

```
CREATE INDEX ind_act_clv0bra on Actua(clv0bra);
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	26	52 (8)	00:00:01

CONSULTA 3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		7422	934K	227 (1)	00:00:01

```
CREATE MATERIALIZED VIEW vistaActor
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1274	338K	157 (2)	00:00:01

TRIGGER 1:

Una serie no puede volver aparecer en la tabla de películas, y viceversa

```
-- Cualquier identificador de la tabla de Películas no puede estar como  
-- identificador en la tabla de Series
```

```
CREATE OR REPLACE TRIGGER trigger1P
```

```
BEFORE INSERT ON Pelicula
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    n NUMBER;
```

```
BEGIN
```

```
    -- n = número de veces que la clvPelicula nueva aparece
```

```
    -- en la tabla Serie
```

```
SELECT count(*)
```

```
INTO n
```

```
FROM Serie
```

```
WHERE clvSerie=:NEW.clvPelicula;
```

```
    -- Si n>0, ya existe esa la clvPelicula en Series,
```

```
    -- lanza un error y no ejecuta el insert
```

```
IF n > 0 THEN
```

```
        RAISE_APPLICATION_ERROR(-20001, 'Ya existe una serie con id');
```

```
END IF;
```

```
END trigger1P;
```

```
/
```

TRIGGER 2:

Una película que es remake de otra no puede tener como remake esa otra película. Lo mismo con precuelas y secuelas

-- Si una película 'Uno' es remake de una película 'Dos', entonces 'Dos' no puede ser remake de 'Uno'. Lo mismo con precuelas y secuelas.

```
CREATE OR REPLACE TRIGGER trigger2
```

```
BEFORE INSERT ON Categoriza
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    n NUMBER;
```

```
BEGIN
```

-- n = número de veces que la tupla de Películas opuesta a la nueva es del tipo nuevo

```
SELECT count(*)
```

```
INTO n
```

```
FROM Categoriza
```

```
WHERE
```

```
    :NEW.clvPrincipal=clvReferencia AND
```

```
    :NEW.clvReferencia=clvPrincipal AND
```

```
    :NEW.Tipo=Tipo;
```

-- Si n>1, ya existe una relación entre las dos películas nuevas del mismo tipo pero

-- con claves invertidas, lanza un error y no ejecuta el insert

```
IF n > 1 THEN
```

```
    RAISE_APPLICATION_ERROR(-20003, 'Ya existe el contrareciproco de esta  
insercion.');
```

```
END IF;
```

```
END trigger2;
```

```
/
```

TRIGGER 3:

Si una película es precuela de otra, la otra es secuela de la primera, y viceversa

-- Si una película 'Uno' es precuela de una película 'Dos', entonces 'Dos' es secuela de 'Uno';

-- y viceversa.

CREATE OR REPLACE TRIGGER trigger3

AFTER INSERT ON Categoriza

FOR EACH ROW

DECLARE

n NUMBER;

BEGIN

-- Si el tipo de inserción es una secuela con respecto a la referencia:

IF :NEW.Tipo = 'follows' **THEN**

-- n = número de veces que la tupla de Películas inversa aparece con el tipo precuela

SELECT count(*)

INTO n

FROM Categoriza

WHERE

:NEW.clvPrincipal=clvReferencia **AND**

:NEW.clvReferencia=clvPrincipal **AND**

Tipo='followed by';

-- Si n = 0, no hay ninguna aparición, se inserta automáticamente la nueva fila

-- correspondiente

IF n = 0 **THEN**

INSERT INTO Categoriza (clvPrincipal,clvReferencia,Tipo)

VALUES (:NEW.clvReferencia,:NEW.clvPrincipal,'followed by');

END IF;

END IF;

TRIGGER 3:

Si una película es precuela de otra, la otra es secuela de la primera, y viceversa

```
-- Si el tipo de inserción es una precuela con respecto a la referencia:
IF :NEW.Tipo = 'followed by' THEN
    -- n = número de veces que la tupla de Películas inversa aparece con el
    tipo secuela
    SELECT count(*)
    INTO n
    FROM Categoriza
    WHERE
        :NEW.clvPrincipal=clvReferencia AND
        :NEW.clvReferencia=clvPrincipal AND
        Tipo='follows';
    -- Si n = 0, no hay ninguna aparición, se inserta automáticamente la nueva
    fila
    -- correspondiente
    IF n = 0 THEN
        INSERT INTO Categoriza (clvPrincipal,clvReferencia,Tipo)
        VALUES (:NEW.clvReferencia,:NEW.clvPrincipal,'follows');
    END IF;
END IF;
END trigger3;
/
```