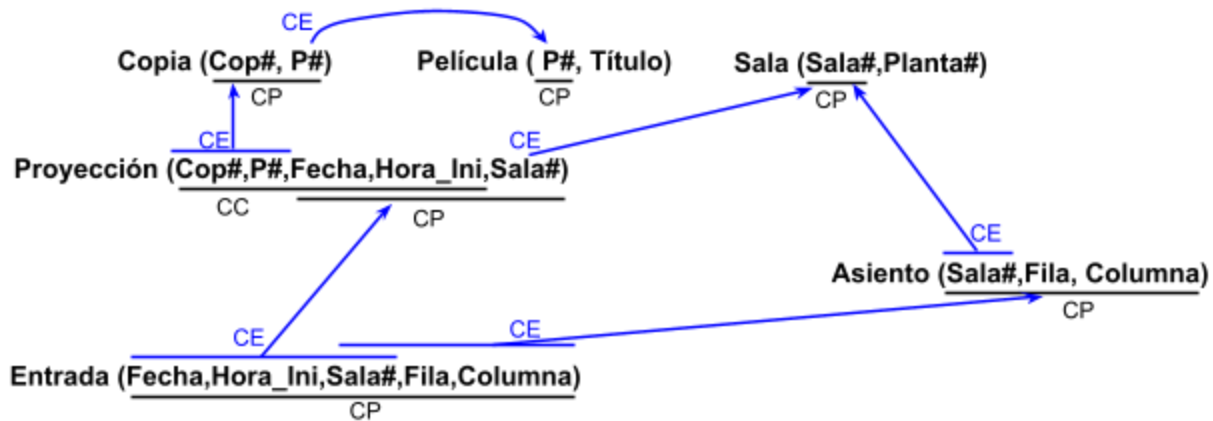


Parcial 2

Disponemos de la siguiente BD que gestiona la actividad de un cine, donde la tabla *Entrada* registra las entradas que se han vendido para cada proyección.



Se pide:

- Escribe las instrucciones en SQL para la creación de la tabla *Entrada* y la inserción de una tupla en dicha tabla. Para la creación de la tabla se presuponen creadas el resto de tablas. Además de las restricciones de integridad especificadas en el esquema, deben considerarse las siguientes:
 - Fecha es de tipo date, Hora_ini es de tipo numérico, se almacena en formato 24h, o sea de 0 a 23h y sólo se permiten valores enteros desde las 10h a las 01h de la madrugada, Sala# es entero de 0 a 19, Fila es una letra mayúscula del alfabeto y Columna un entero de 1 a 99. (1.25 pts.).

Res:

```

CREATE TABLE Entrada(
  Fecha date,
  Hora_ini number(2) CHECK ((Hora BETWEEN 10 AND 23) OR (Hora BETWEEN 0
AND 1)),
  Sala# number(2) CHECK (Sala# BETWEEN 0 AND 19),
  Fila varchar(1) CHECK (Fila BETWEEN 'A' and 'Z')),
  Columna number(2) CHECK (Columna BETWEEN 1 AND 99),
  FOREIGN KEY (Fecha, Hora_ini, Sala#)
    REFERENCES Proyeccion(Fecha, Hora_ini, Sala#),
  FOREIGN KEY (Sala#, Fila, Columna)
    REFERENCES Asiento(Sala#, Fila, Columna),
  PRIMARY KEY (Fecha, Hora_ini, Sala#, Fila, Columna),
);
  
```

```

INSERT INTO Proyeccion VALUES
  
```

```
('C12','P3',TO_DATE('10/06/2015','DD/MM/YYYY'),20,'S7');
```

B. Realiza las siguientes consultas::

- a. “Muestra proyecciones para las que no se han vendido ninguna entrada”.(en **AR 1.25 pts. y CRT 1.25 pts.**).

$$\pi_{Fecha,Hora_ini,Sala\#}(Proyeccion) - \pi_{Fecha,Hora_ini,Sala\#}(Entrada)$$

```
{P | proyeccion(P) and not (exists E)(entrada(E) and E.Fecha=P.Fecha
and E.Hora_ini=P.Hora_ini and E.Sala#=P.Sala#)};
```

```
RANGE P in Proyeccion
```

```
RANGE E in Entrada
```

```
Select P Where  $\nexists$ E (E.Fecha=P.Fecha  $\wedge$  E.Hora_ini=P.Hora_ini  $\wedge$ 
E.Sala#=P.Sala#)
```

- b. “Muestra el título de las películas de las que sólo existe una copia”. (en **AR 1.25pts. y SQL 1.25 pts.**)

$$\rho(Copia) = c1$$

$$\rho(Copia) = c2$$

$$\pi_{Titulo}(Pelicula \bowtie (\pi_{P\#}(Copia) - (\pi_{c1.P\#}(\sigma_{c1.P\#=c2.P\# \wedge c1.cop\# < c2.cop\#}(c1 \times c2)))))$$

```
{P.Titulo | pelicula(P) and (exists C1)(copia(C1) and C1.P#=P.P# and
not (exists C2) (copia(C2) and C2.P#=C1.P# and C2.Cop#<>C1.Cop#))};
```

```
RANGE P in Pelicula
```

```
RANGE C1,C2 in Copia
```

```
Select P.Titulo Where  $\exists$ C1(C1.P#=P.P#  $\wedge$   $\nexists$ C2 (C2.P#=C1.P#  $\wedge$ 
C2.Cop#<>C1.Cop#))
```

```
Select P.Titulo From Pelicula P, Copia C1 Where C1.P#=P.P# and
not exists( Select * from Copia C2 Where C2.P#=C1.P# and
C2.Cop#<>C1.Cop#);
```

```
Select P.Titulo From Pelicula P, Copia C Where C.P#=P.P#
```

```
Group by (P.Titulo,C.P#) Having couny(*) = 1;
```

- c. “Muestra el título de las películas que se han proyectado en todas las salas”. (en **CRT 1.25 pts. y SQL 1.25 pts.**).

```
{P.Titulo | pelicula(P) and not (exists S)(sala(S) and not (exists
Pr)(proyeccion(Pr) and Pr.P#=P.P# and Pr.sala#=S.sala#))};
```

```
RANGE P in Pelicula
```

```
RANGE S in Sala
```

```

RANGE Pr in Proyeccion
Select P.Titulo Where AS(Pr(Pr.P#=P.P# ^ Pr.sala#=S.sala#))

Select P.Titulo Where VS(Pr(Pr.P#=P.P# ^ Pr.sala#=S.sala#))

Select P.Titulo From pelicula P Where and not exists (
select * From sala S Where not exists(
Select * From proyeccion Pr Where Pr.P#=P.P# and Pr.sala#=S.sala#))
Select P.Titulo From pelicula P Where and not exists (
Select S.sala# From sala S
Minus
Select Pr.sala# From proyeccion Pr Where Pr.P#=P.P#
)

```

- C. Crea una vista que muestre la cantidad de entradas vendidas por cada proyección que haya vendido más de 40 entradas. (1.25 pt.).

```

Create View entradas_proyeccion AS
Select E.Fecha,E.Hora_ini,Sala#,Count(*) From Entrada E
Group by E.Fecha,E.Hora_ini,E.Sala# Having Count(*)> 40;

```

Tiempo de realización: **Un Parcial 1.5h, dos parciales 3h**