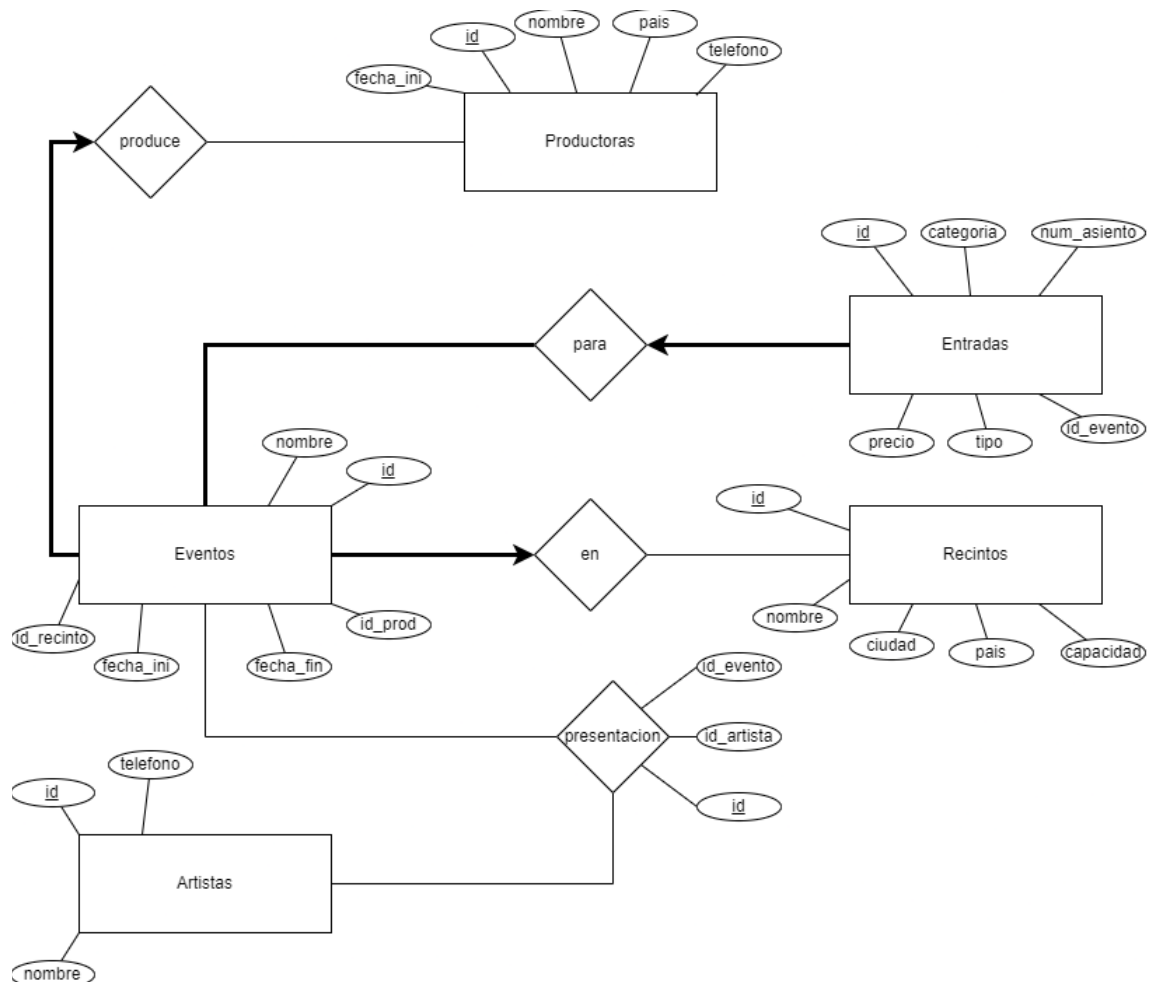


Entrega 2

Diagrama E/R:



Esquema Relacional:

Productoras(
id int primary key,
nombre varchar(70),
pais varchar(20),
fecha_ini date,
telefono bigint
)

```
Eventos(  
    id int primary key,  
    nombre varchar(30),  
    fecha_ini date,  
    fecha_fin date,  
    id_recinto int not null,  
    id_prod int not null,  
    foreign key(id_recinto) references Recintos(id) on delete cascade,  
    foreign key(id_prod) references Productoras(id) on delete cascade  
)
```

```
Recintos(  
    id int primary key,  
    nombre varchar(40),  
    ciudad varchar(20),  
    pais varchar(20),  
    capacidad int  
)
```

```
Entradas(  
    id int primary key,  
    categoria varchar(20),  
    precio float,  
    num_asiento varchar(5),  
    tipo varchar(15),  
    id_evento int not null,  
    foreign key(id_evento) references Eventos(id) on delete cascade  
)
```

```
Artistas(  
    id int primary key,  
    nombre varchar(30),  
    telefono int  
)
```

```
Presentacion(  
    id int primary key,  
    id_artista int not null,  
    id_evento int not null,  
    foreign key(id_artista) references Artistas(id) on delete cascade,  
    foreign key(id_evento) references Eventos(id) on delete cascade  
)
```

Justificación BCNF:

En la relación Productoras:

Productoras(id, nombre, pais, fecha_ini, telefono)

Llaves:

- id
- nombre
- telefono

Dependencias:

- $\text{id} \rightarrow \text{todo}$
- $\text{nombre} \rightarrow \text{todo}$
- $\text{telefono} \rightarrow \text{todo}$
- $\text{id}, \text{nombre} \rightarrow \text{todo}$
- $\text{id}, \text{telefono} \rightarrow \text{todo}$
- $\text{nombre}, \text{telefono} \rightarrow \text{todo}$
- $\text{id}, \text{nombre}, \text{telefono} \rightarrow \text{todo}$

En la relación Eventos:

Eventos(id, nombre, fecha_ini, fecha_fin, id_recinto, id_prod)

Llaves:

- id
- nombre
- fecha_fin

Dependencias:

- $\text{id} \rightarrow \text{todo}$
- $\text{nombre} \rightarrow \text{todo}$
- $\text{fecha_fin} \rightarrow \text{todo}$
- $\text{id}, \text{nombre} \rightarrow \text{todo}$
- $\text{id}, \text{fecha_fin} \rightarrow \text{todo}$
- $\text{nombre}, \text{fecha_fin} \rightarrow \text{todo}$
- $\text{id}, \text{nombre}, \text{fecha_fin} \rightarrow \text{todo}$

En la relación Recintos:

Recintos(id, nombre, ciudad, pais, capacidad)

Llaves:

- id
- nombre
- capacidad

Dependencias:

- $\text{id} \rightarrow \text{todo}$
- $\text{nombre} \rightarrow \text{todo}$
- $\text{capacidad} \rightarrow \text{todo}$
- $\text{id}, \text{nombre} \rightarrow \text{todo}$
- $\text{id}, \text{capacidad} \rightarrow \text{todo}$

- nombre, capacidad \rightarrow todo
- id, nombre, capacidad \rightarrow todo

En la relación Entradas:

Entradas(id, categoria, precio, num_asiento, tipo, id_evento)

Llaves:

- id

Dependencias:

- id \rightarrow todo

En la relación Artistas:

Artistas(id, nombre, telefono)

Llaves:

- id
- nombre
- telefono

Dependencias:

- id \rightarrow todo
- nombre \rightarrow todo
- telefono \rightarrow todo
- id, nombre \rightarrow todo
- id, telefono \rightarrow todo
- nombre, telefono \rightarrow todo
- id, nombre, telefono \rightarrow todo

En la relación Presentación:

Presentacion(id, id_artista, id_evento)

Llaves:

- id

Dependencias:

- id \rightarrow todo

Como todas las relaciones están en BCNF el esquema está en BCNF. Cabe mencionar que los atributos que son llaves, no fueron “forzados a serlo”. Es decir, al filtrar los datos no eliminamos a propósito los repetidos en esa columna (a excepción de los mencionados en los supuestos). Sin embargo, el id obviamente lo creamos para que sí sea llave (llave primaria).

Consultas SQL:

1. Entregue los nombres y números de contacto de todas las productoras de eventos

```
select nombre, telefono  
from Productoras;
```

2. Entregue el nombre de las productoras, junto a la cantidad de eventos que han producido

```
select Productoras.nombre, count(*)  
from Eventos, Productoras  
where Eventos.id_prod = Productoras.id  
group by Productoras.id;
```

3. Dado el nombre de una productora (ej: 'lets go company'), entregue los datos del último evento que ha producido

```
select id, nombre, fecha_ini, fecha_fin, id_recinto, id_prod  
from eventos  
where eventos.fecha_ini = (  
    select max(fecha_ini)  
    from eventos  
    where eventos.id_prod = (  
        select productoras.id  
        from productoras  
        where productoras.nombre = 'lets go company'))  
order by id;
```

4. Dada una productora (ej: 'lets go company'), liste a los artistas con los que ha trabajado dicha productora

```
select Artistas.nombre, Artistas.telefono  
from artistas, eventos, presentacion  
where eventos.id = presentacion.id_evento  
and presentacion.id_artista = artistas.id  
and eventos.id_prod = all (  
    select productoras.id  
    from productoras  
    where productoras.nombre = 'lets go company');
```

5. Dado un evento (ej: 'rio concert'), entregue la suma total de ingresos por ventas de entradas de dicho evento

```
select sum(entradas.precio)
from entradas
where entradas.id_evento = all (
  select eventos.id
  from eventos
  where eventos.nombre = 'rio concert');
```

6. Liste todos los eventos, junto al número total de artistas que se presentarán en cada uno

```
select eventos.id, eventos.nombre, eventos.fecha_ini, eventos.fecha_fin, eventos.id_recinto,
eventos.id_prod, count(*) as cantidad
from Eventos, Presentacion
where Eventos.id = Presentacion.id_evento
group by Eventos.id
order by cantidad desc;
```

7. Muestre el nombre del evento que tiene la mayor cantidad de entradas vendidas

```
select eventos.nombre, count(entradas.id_evento) as entr_vend
from eventos, entradas
where eventos.id = entradas.id_evento
group by eventos.id
order by entr_vend desc
limit 1;
```

Supuestos:

1. En las relaciones que existe el atributo nombre, se eliminaron los repetidos de esa columna.
2. Las tuplas con fechas inexistentes se eliminaron.
3. Se asume que las consultas que comienzan con “dado una productora” o “dado un evento”, se refiere al nombre de la productora o del evento respectivamente.