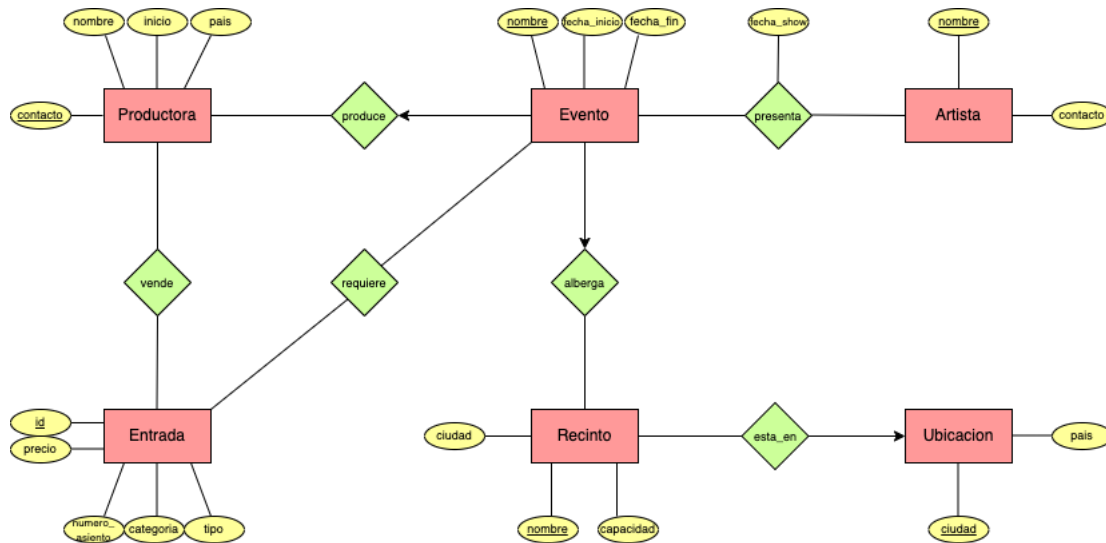


Reporte Entrega 2

1. Modelo de Entidad/Relación

A continuación se ilustra un modelo de Entidad/Relación que representa la situación planteada. Se ilustran las entidades, asociaciones y cardinalidades.



Productora.inicio = fecha de inicio de actividades de la productora.

2. Esquema Relacional

A continuación se muestra el esquema relacional, en el que se especifican los datatypes, las llaves primarias y las llaves foráneas como restricciones de integridad de las relaciones.

Productora(contacto VARCHAR PRIMARY KEY, nombre VARCHAR, pais VARCHAR, inicio DATE)

Evento(nombre VARCHAR PRIMARY KEY, fecha_inicio DATE, fecha_fin DATE)

Entrada(id INT PRIMARY KEY, tipo VARCHAR, precio INT, numero_asiento INT, categoria VARCHAR)

Presenta(nombre_artista VARCHAR FOREIGN KEY, nombre_evento VARCHAR FOREIGN KEY, fecha_show DATE)

Artista(nombre VARCHAR PRIMARY KEY, contacto VARCHAR)

Recinto(nombre VARCHAR PRIMARY KEY, ciudad VARCHAR, capacidad INT)

Ubicacion(ciudad VARCHAR PRIMARY KEY, pais VARCHAR)

Requiere(nombre_evento VARCHAR FOREIGN KEY, id_entrada INT FOREIGN KEY)

Vende(contacto_evento VARCHAR FOREIGN KEY, id_entrada INT FOREIGN KEY)

Produce(contacto_productora VARCHAR FOREIGN KEY, nombre_evento VARCHAR FOREIGN KEY)

Esta_en(nombre_recinto VARCHAR FOREIGN KEY, ciudad_recinto VARCHAR FOREIGN KEY)

3. Justificación del modelo

A continuación se muestra que el modelo está en BCNF. Se justifica esto mediante el listado de dependencias funcionales de cada tabla, al mostrar que está todo normalizado.

Productora(contacto, nombre, pais, inicio)

{contacto} → {nombre, pais, inicio}

Evento(nombre, fecha_inicio, fecha_fin)

{nombre} → {fecha_inicio, fecha_fin}

Artista(nombre, contacto)

{nombre} → {contacto}

{contacto} → {nombre}

Recinto(nombre, ciudad, capacidad)

{nombre} → {ciudad, capacidad}

Ubicacion(ciudad, pais)

{ciudad} → {pais}

Entrada(id, tipo, precio, numero_asiento, categoria)

{id} → {tipo, precio, numero_asiento, categoria}

Presenta(nombre_artista, nombre_evento, fecha_show)

{nombre_artista, nombre_evento} → {fecha_show}

En las siguientes relaciones, las dependencias funcionales son triviales:

Requiere(nombre_evento, id_entrada)

Vende(contacto_evento, id_entrada)

Produce(contacto_productora, nombre_evento)

Esta_en(nombre_recinto, ciudad_recinto)

4. Consultas en SQL

A continuación se muestran todas las consultas en SQL que se implementaron en la aplicación.

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

```
SELECT nombre, contacto
FROM productora
```
- b. Entregue el nombre de las productoras, junto a la cantidad de eventos que han producido

```
SELECT productora.nombre, COUNT(DISTINCT(produce.contacto_productora,
produce.nombre_evento)) AS cant_eventos
FROM productora, produce
WHERE productora.contacto = produce.contacto_productora
GROUP BY productora.nombre
```
- c. Dado el nombre de una productora, entregue los datos del último evento que ha producido

```
SELECT evento.nombre, evento.fecha_inicio, evento.fecha_fin
FROM evento, productora
WHERE productora.nombre LIKE '$nombre'
AND evento.fecha_fin >= ALL(
    SELECT fecha_fin
    FROM evento)
```
- d. Dada una productora, liste a los artistas con los que ha trabajado dicha productora

```
SELECT artista.nombre, artista.contacto
FROM artista, presenta
WHERE artista.nombre = presenta.nombre_artista
AND presenta.nombre_evento(
    SELECT nombre_evento
    FROM produce
    WHERE produce.contacto_productora LIKE '$contacto')
```
- e. Dado un evento, entregue la suma total de ingresos por ventas de entradas de dicho evento

```
SELECT SUM(entrada.precio)
FROM requiere, entrada
WHERE entrada.id = requiere.id_entrada
AND requiere.nombre_evento LIKE '$nombre'
```
- f. Liste todos los eventos, junto al número total de artistas que se presentaran en cada uno

```
SELECT evento.nombre, COUNT(DISTINCT(presrnta.nombre_artista,
presenta.nombre_evento)) AS cant_artistas
FROM evento, presenta
```

```
WHERE evento.nombre = presenta.nombre_evento  
GROUP BY evento.nombre
```

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

5. Supuestos

A continuación se describen los supuestos realizados sobre la entrega.

- a. Las consultas 1, 2, 4 y 6 funcionan todas perfectamente en la terminal, al entregar los datos solicitados usando las tablas y datos subidos a la carpeta de la Entrega 2. Sin embargo, solamente la consulta 1 funciona mediante la página web. Podemos demostrar esto y también es posible que cualquier ayudante lo
- b. No fuimos capaces de completar la consulta 7 en SQL. Por eso, no se incluye. La página muestra los datos de la tabla Vende. Decidimos que mostrar esto era mejor a no mostrar nada, aunque sabemos que es un error.
- c. Se eliminaron los eventos de mismo nombre que se daban en distinta fecha, ya que se consideraron duplicados (se siguió lo recomendado en las Issues del curso).