

Bases de datos SQL

HUGO GÓMEZ SABUCEDO

hugogomezsabucedo@gmail.com

Máster Big Data, Data Science & Inteligencia Artificial

Curso 2024-2025

Universidad Complutense de Madrid

1. Enunciado del problema

El enunciado del problema es el siguiente:

Tenemos una empresa dedicada a la organización de eventos culturales únicos “ArteVida Cultural”. Organizamos desde deslumbrantes conciertos de música clásica hasta exposiciones de arte vanguardista, pasando por apasionantes obras de teatro y cautivadoras conferencias, llevamos la cultura a todos los rincones de la comunidad.

Necesitamos gestionar la gran variedad de eventos y detalles, así como las ganancias que obtenemos. Para ello, es necesario llevar un registro adecuado de cada evento, de los artistas que los protagonizan, las ubicaciones donde tienen lugar, la venta de entradas y, por supuesto, el entusiasmo de los visitantes que asisten.

Hemos decidido diseñar e implementar una base de datos relacional que no solo simplifique la organización de eventos, sino que también permita analizar datos valiosos para tomar decisiones informadas.

En nuestra empresa ofrecemos una serie de actividades que tienen un nombre, un tipo: concierto de distintos tipos de música (clásica, pop, blues, soul, rock and roll, jazz, reggaeton, gospel, country, ...), exposiciones, obras de teatro y conferencias, aunque en un futuro estamos dispuestos a organizar otras actividades. Además, en cada actividad participa uno o varios artistas y un coste (suma del caché de los artistas).

El artista tiene un nombre, un caché que depende de la actividad en la que participe y una breve biografía.

La ubicación tendrá un nombre (Teatro Maria Guerrero, Estadio Santiago Bernabeu, ...), dirección, ciudad o pueblo, aforo, precio del alquiler y características.

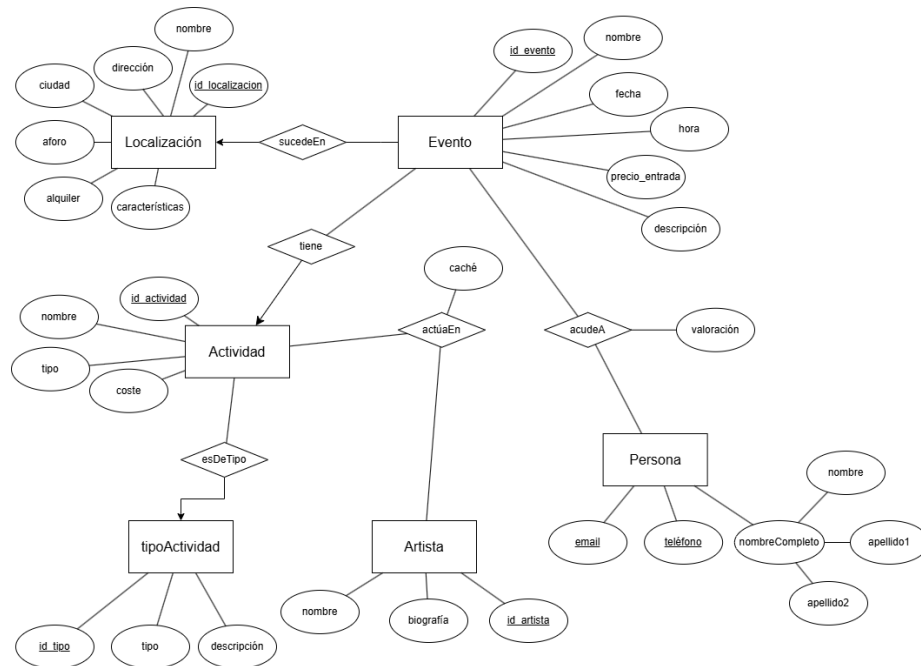
De cada evento tenemos que saber el nombre del evento (p.e. “VI festival de música clásica de Alcobendas”), la actividad, la ubicación, el precio de la entrada, la fecha y la hora, así como una breve descripción del mismo. En un evento sólo se realiza una actividad.

También tendremos en cuenta los asistentes a los eventos, de los que sabemos su nombre completo, sus teléfonos de contacto y su email. Una persona puede asistir a más de un evento y a un evento pueden asistir varias personas.

Nos interesará realizar consultas por el tipo de actividad, en que fecha se han realizado más eventos, en qué ciudad realizamos más eventos, ...

2. Diseño conceptual

El diagrama entidad-relación que se propone es el siguiente:



A continuación se explica más en profundidad dicho diagrama y las diferentes entidades y relaciones que lo componen.

En resumen, se han definido seis entidades (actividad, tipoActividad, artista, evento, localización y persona), cada una de las cuales se corresponde con los principales conceptos que se mencionan en el enunciado del ejercicio. Además, se han definido cinco relaciones (sucedeEn, tiene, actúaEn, acudeA y esDeTipo) entre las entidades anteriormente mencionadas.

En primer lugar tenemos la entidad **Evento**, que representa cada uno de los eventos que realiza la empresa. Esta entidad tendrá como atributos el nombre, la fecha y hora (que en este punto he considerado como dos atributos diferenciados, aunque a posteriori en el paso a tablas pueda definirse como un único atributo, por ejemplo, de tipo *timestamp*), el precio de la entrada y la descripción del mismo. Además, se ha añadido un atributo adicional, id_evento, el cual se usará como clave primaria de la entidad. Aunque inicialmente había pensado en utilizar como clave del evento su nombre y la fecha y hora, no me parece la mejor decisión, también en terminos de rendimiento.

Un evento tiene lugar en una ubicación en concreto. Es por eso que se define la entidad **Localización**, que representa las diferentes salas o lugares donde

puede tener lugar un evento. Como se indica, sus atributos son su nombre, la dirección, la ciudad, el aforo, el precio del alquiler y las características del mismo. A mayores, se ha definido un atributo adicional id_localizacion, el cual, al igual que en el caso anterior, usaremos como clave primaria de la entidad.

Entre estas dos entidades surge la relación **sucedeEn**, que es de 1 a muchos entre Evento y Localización, ya que un evento solamente puede tener lugar en una localización, pero una misma localización puede albergar varios eventos a lo largo del tiempo.

Por una parte, se define la entidad **Persona**, que representa los diferentes asistentes a los eventos. De éstas, únicamente se nos indica que nos interesan su nombre completo, su email y su teléfono. El nombre completo lo hemos definido como un atributo compuesto, el cual dividimos a su vez en nombre, apellido1 y apellido2. Por su parte, como clave primaria se ha considerado el email y el teléfono de la persona. En el enunciado del problema no se especifica que se vayan a recoger identificadores únicos de las personas, como podría ser el DNI, por lo que este no se puede añadir como atributo. Además, emplear un identificador autogenerado y autoincremental me parecía redundante respecto al resto de entidades, y poco eficiente. Por eso he considerado el email y teléfono como la clave primaria, ya que me parece que es una combinación que, en cualquier caso, debería ser única (no tendría mucho sentido que una persona se registrase varias veces pero con diferentes combinaciones de email y teléfono).

Esta entidad tiene únicamente una relación, **acudeA**, que la relaciona con el evento. Esta relación es de muchos a muchos, ya que se indica que una persona puede asistir a varios eventos, y que a un evento asisten varias personas. Además, en el enunciado se comenta que para los eventos se desea conocer "el entusiasmo de los visitantes que asisten". Es por ello que se ha definido un atributo de esta relación, valoración, que nos permitirá registrar las opiniones de los asistentes a los diferentes eventos.

Por otra parte, se define la entidad **Actividad**, que se corresponde con cada una de las diferentes actividades que tienen lugar en el evento. Como se indica, de una actividad nos interesa su nombre, su tipo y su coste. Adicionalmente, se define el atributo id_actividad, para establecerlo como clave primaria de la entidad. Respecto al coste de la actividad, se indica que el coste es la suma del caché de los artistas. Este coste, en el modelo entidad relación, se ha definido como un atributo diferenciado, pero es importante destacar que depende de la relación entre actividad y artista (un artista, por ejemplo, puede tener un caché diferente para diferentes actividades). En el paso a tablas se explicará como se abordará esta cuestión en nuestra base de datos. La actividad se relaciona con evento mediante la relación **tiene**, de 1 a muchos entre evento y actividad. Es decir, un evento sólo tiene una actividad, pero una actividad puede estar en varios eventos.

Respecto a la actividad, se indica que puede ser de diferentes tipos. A priori, podría parecer lógico definir el tipo de la actividad como un atributo de la

propia entidad. Sin embargo, parece más adecuado definir el tipo de actividad como una entidad diferenciada, ya que los tipos de actividad serán siempre los mismo (dentro de un conjunto), y esto nos permite además definir atributos adicionales como por ejemplo una breve descripción. Es por esto que nace la entidad **tipoActividad**, cuyos atributos son un id_tipo, así como el nombre o tipo y la descripción. Esta entidad se relaciona con Actividad mediante la relación **esDeTipo**, que es de cardinalidad varios a uno. Es decir, una actividad sólo es de un tipo en concreto, pero un tipo puede tener (y tiene) varias actividades.

Por último, se define la entidad **Artista**, que únicamente tiene como atributos el nombre del mismo y su biografía, así como un atributo adicional id_artista que se usará como clave primaria. El Artista se relaciona con la Actividad (ya que se indica en el enunciado que un artista participa en una actividad, y no en un evento), mediante la relación **actúaEn**. Esta relación tiene un atributo, caché, que representa el caché del artista. Como se comentó anteriormente, el caché de un artista no es una cantidad fija, sino que dependiendo de la actividad, un artista puede tener un caché diferente. Por eso, no puede considerarse un atributo de la entidad artista, ya que no es algo intrínseco a él, sino que se establece al crearse la actividad en cuestión. Es por eso que se ha decidido incluir como un atributo de la relación actúaEn. Además, esta relación es de muchos a muchos: se indica que en cada actividad participan uno o varios artistas y, naturalmente, un artista puede participar en múltiples actividades.

3. Diseño lógico

Una vez que tenemos definidas todas las entidades, podemos realizar el diseño lógico de la base de datos, mediante el paso a tablas del modelo entidad relación para obtener el modelo relacional.

En primer lugar, deberemos crear una tabla por cada una de las entidades que tenemos en el MER. Esto nos dará lugar a seis tablas, con sus correspondientes atributos: evento, localización, persona, actividad, tipoActividad y artista.

Por otra parte, aquellas relaciones que son de muchos a muchos generarán una tabla. De esta forma, se añaden dos nuevas tablas: *acudeA* y *actúaEn*. En cada una de estas tablas, guardaremos las claves primarias de cada una de las entidades participantes, así como los atributos, si los hubiese, de dicha relación. Para las que son de 1 a muchos, se pasará la clave primaria de la entidad con participación 1 a la entidad con participación n.

Como detalle, destacar que los atributos *fecha* y *hora* de la entidad Evento se han fusionado en uno solo, *fecha*, ya que a nivel de implementación en SQL se puede definir como un tipo timestamp, facilitando el trabajo con los datos.

De esta forma, el modelo relacional es el siguiente:

LOCALIZACION(id_localizacion, nombre, direccion, ciudad, aforo, alquiler, características)

EVENTO(id_evento, id_localizacion, fecha, precio_entrada, descripción)

ACTIVIDAD(id_actividad, id_evento, id_tipo, nombre, coste)

TIPOACTIVIDAD(id_tipo, tipo, descripción)

ARTISTA(id_artista, nombre, biografía)

PERSONA(email, telefono, nombre, apellido1, apellido2)

ACTUAEN(id_artista, id_actividad, cache)

ACUDEA(email, telefono, id_evento, valoracion)

4. Implementación y consultas

El script completo de la implementación de la base de datos se encuentra en el Anexo A. Sin embargo, aquí se explicará la elección de algunos tipos de los datos, así como un breve detalle de los triggers y vistas definidos y de las consultas propuestas, aclarando brevemente su objetivo y como se llega al mismo.

A. Anexo I: Script de SQL

aaaaaaa

```
1  /* -----
2  HUGO GOMEZ SABUCEDO
3  XXXXX NOMBRE DE LA BASE DE DATOS XXXXXXXXXXXX
4  ----- */
5  /* -----
6  Definicion de la estructura de la base de datos
7  -----
8  */
9  /* -----
10 Trigger
11 Insercion de datos
12 -----
13 */
14 /* -----
15 Consultas, modificaciones, borrados y vistas
16 -----
17 */
```