

Proyecto SQL – ArteVida Cultural



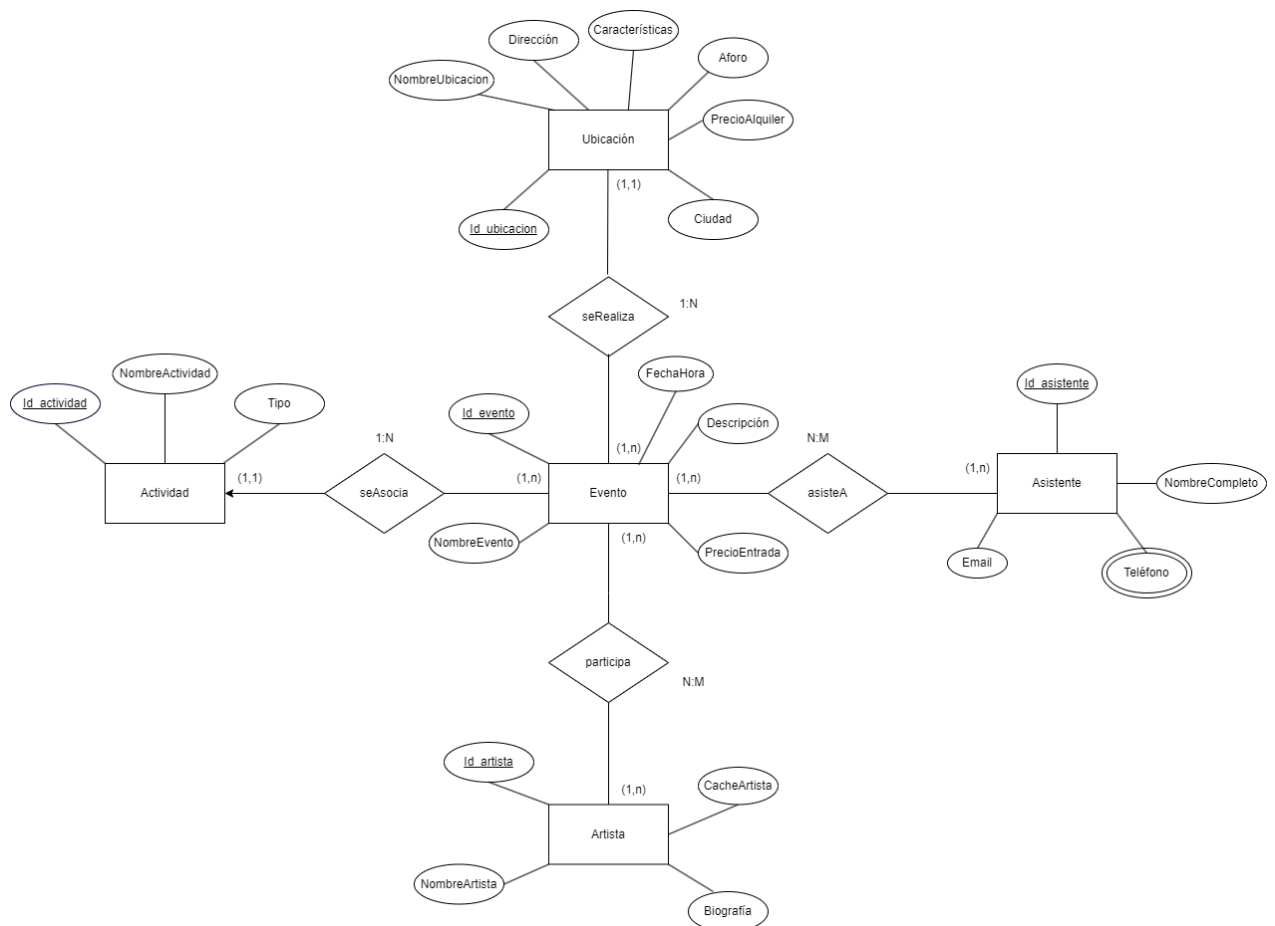
ArteVida Cultural

Objetivo

El objetivo de este proyecto es diseñar e implementar una base de datos relacional que simplifique la organización de eventos de la empresa “ArteVida Cultural” y que permita analizar sus datos para la futura toma de decisiones. Para ello, primero es necesario gestionar la gran variedad de eventos y detalles, así como las ganancias obtenidas.

Diseño conceptual

Modelo Entidad-Relación



Entidades:

- **Actividad:** Describe las diferentes actividades que se ofrecen (conciertos, exposiciones, obras de teatro y conferencias)
 - Id_actividad: clave primaria.
 - NombreActividad: simple, univalorado.
 - Tipo: simple, univalorado.

- **Artista:** Representa a los artistas que participan en las actividades.
 - Id_artista: clave primaria
 - NombreArtista: simple, univalorado.
 - CacheArtista: simple, univalorado.
 - Biografia: simple, univalorado.
- **Ubicación:** Describe los lugares donde se llevan a cabo los eventos.
 - Id_ubicacion: clave primaria
 - NombreUbicacion: simple, univalorado.
 - Direccion: compuesto (calle, número...)
 - Ciudad: simple, univalorado.
 - Aforo: simple, univalorado.
 - PrecioAlquiler: simple, univalorado.
 - Características: multivalorado (puede tener múltiples características)
- **Evento:** Representa los eventos organizados por la empresa. Cada evento está asociado a una actividad específica.
 - Id_evento: clave primaria.
 - NombreEvento: simple, univalorado.
 - FechaHora: compuesto.
 - PrecioEntrada: simple, univalorado.
 - Descripcion: simple, univalorado.
- **Asistente:** Representa a las personas que acuden a los eventos.
 - Id_asistente: clave primaria.
 - NombreCompleto: compuesto (Nombre, apellidos).
 - Teléfono: multivalorado. Se requerirá al menos un número de teléfono hasta un máximo de 3.
 - Email: simple, univalorado.

Relaciones:

- *participa*: relación binaria. Conecta los eventos con los artistas. Un evento puede tener uno o varios artistas y un artista puede participar en varios eventos (N:M).
- *seRealiza*: relación binaria. Conecta evento y ubicación. Un evento tiene lugar en una ubicación, pero una ubicación puede albergar varios eventos (1:N).
- *asisteA*: relación binaria. Conecta asistente y evento. Un asistente puede asistir a uno o varios eventos y un evento puede tener varios asistentes (N:M).
- *seAsocia*: relación binaria. Conecta evento y actividad. Un evento se clasifica según la actividad que representa. Un evento puede estar asociado a una actividad y una actividad puede tener varios eventos (1:N).

Otros conceptos:

1. Coste total del evento: deriva de la suma del caché de cada artista y del precio del alquiler de la ubicación.
2. Valoración de los asistentes: no está representada en el modelo E-R pero podría ser parte de la gestión financiera de la empresa.
3. Empleados: los encargados de organizar el evento no están incluidos, pero son una parte importante del proceso de organización.
4. Ganancias: no se especifica como se calculan las ganancias, pero podría derivar del coste total de cada evento y de las entradas vendidas.

Se creará una vista para poder consultar el coste total del evento, calculando para ello el coste de los artistas que participan en cada actividad, el precio del alquiler y las ganancias producidas de la venta de entradas.

Diseño lógico

Paso a tablas

Entidades:

Actividad (id_actividad, nombreActividad, tipo)

Artista (id_artista, nombreArtista, cacheArtista, biografia)

Ubicación (id_ubicacion, nombreUbicacion, dirección, ciudad, aforo, precioAlquiler, características)

Evento (id_evento, nombreEvento, fechaHora, precioEntrada, descripción, *id_actividad*, *id_ubicacion*)

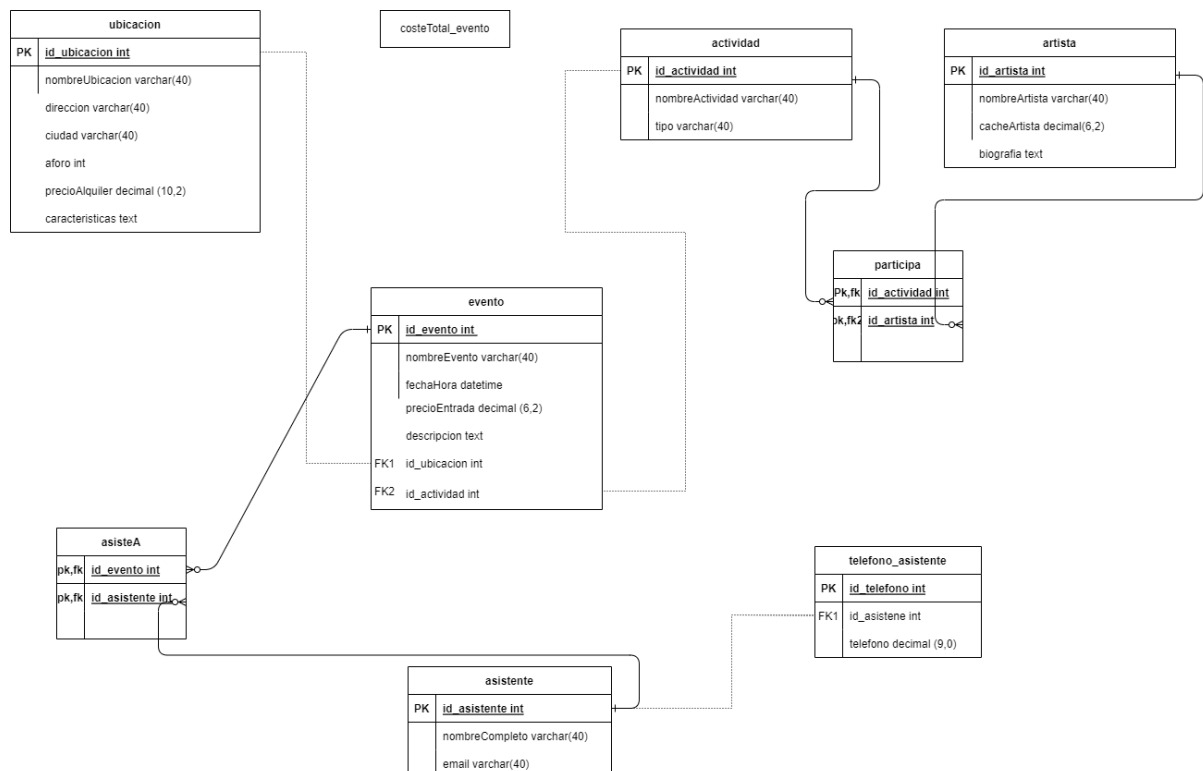
Asistente (id_asistente, nombreCompleto, teléfono, email)

Telefono_asistente(id_telefono, *id_asistente*, teléfono)

Relaciones:

Participa (*id_artista*, *id_evento*)

asisteA (*id_asistente*, *id_evento*)



Implementación

Código del script:

```
/*-----  
Nombre del autor: Coral García Cardeñas  
Nombre de la base de datos: arteVida_Coral  
-----*/  
  
DROP DATABASE IF EXISTS arteVida_Coral;  
  
CREATE DATABASE arteVida_Coral;  
USE arteVida_Coral;  
/*DROP TABLE IF EXISTS actividad;  
DROP TABLE IF EXISTS artista;  
DROP TABLE IF EXISTS ubicacion;  
DROP TABLE IF EXISTS evento;  
DROP TABLE IF EXISTS asistente;  
DROP TABLE IF EXISTS telefono_asistente;  
DROP TABLE IF EXISTS participa;  
DROP TABLE IF EXISTS asisteA;  
*/  
  
/*-----  
Definición de la estructura de la base de datos  
-----*/  
  
-- Creacion tabla actividad  
  
CREATE TABLE actividad (  
    id_actividad int auto_increment,  
    nombreActividad varchar(40),  
    tipo varchar(40),  
    PRIMARY KEY (id_actividad)  
);  
  
-- Creacion tabla artista  
  
CREATE TABLE artista (  
    id_artista int auto_increment,  
    nombreArtista varchar(40),  
    cacheArtista numeric(6,2),  
    biografia text,  
    PRIMARY KEY (id_artista)  
);  
  
-- Creacion tabla ubicacion  
  
CREATE TABLE ubicacion (  
    id_ubicacion int auto_increment,  
    nombreUbicacion varchar(40),  
    direccion varchar(40),  
    ciudad varchar(40),
```

```

aforo int,
precioAlquiler numeric(10, 2),
caracteristicas text,
PRIMARY KEY (id_ubicacion)
);

-- Creacion tabla evento

CREATE TABLE evento (
    id_evento int auto_increment,
    nombreEvento varchar(40),
    fechaHora datetime NOT NULL,
    precioEntrada numeric(6,2) NOT NULL,
    descripcion text,
    id_ubicacion int unique, #cada evento solo puede tener una unica ubicacion y una unica
    actividad
    id_actividad int unique,
    PRIMARY KEY (id_evento),
    FOREIGN KEY (id_ubicacion) REFERENCES ubicacion(id_ubicacion),
    FOREIGN KEY (id_actividad) REFERENCES actividad(id_actividad)
);

-- Creacion tabla asistente

CREATE TABLE asistente (
    id_asistente int auto_increment,
    nombreCompleto varchar(40),
    email varchar(40),
    PRIMARY KEY (id_asistente),
    CONSTRAINT email_unique UNIQUE (email) # emails unicos
);

-- Creacion tabla telefono_asistente
# cada asistente puede tener varios telefonos asociados

CREATE TABLE telefono_asistente (
    id_telefono int auto_increment,
    id_asistente int,
    telefono numeric(9,0),
    PRIMARY KEY (id_telefono),
    FOREIGN KEY (id_asistente) REFERENCES asistente(id_asistente)
);

-- Creacion tabla participa

CREATE TABLE participa (
    id_actividad int,
    id_artista int,
    PRIMARY KEY (id_actividad, id_artista),
    FOREIGN KEY (id_actividad) REFERENCES actividad(id_actividad),
    FOREIGN KEY (id_artista) REFERENCES artista(id_artista)
);

-- Creacion tabla asisteA

```

```

CREATE TABLE asisteA (
    id_evento int,
        id_asistente int,
    PRIMARY KEY (id_evento, id_asistente),
    FOREIGN KEY (id_evento) REFERENCES evento(id_evento),
    FOREIGN KEY (id_asistente) REFERENCES asistente(id_asistente)
);

/*
CREATE TABLE seRealiza (
    id_evento int,
        id_ubicacion int,
    PRIMARY KEY (id_evento, id_ubicacion),
    FOREIGN KEY (id_evento) REFERENCES evento(id_evento),
    FOREIGN KEY (id_ubicacion) REFERENCES ubicacion(id_ubicacion)
);
CREATE TABLE seAsocia (
    id_evento int,
        id_actividad int,
    PRIMARY KEY (id_evento, id_actividad),
    FOREIGN KEY (id_evento) REFERENCES evento(id_evento),
    FOREIGN KEY (id_actividad) REFERENCES actividad(id_actividad)
);
*/

/*-----
Trigger
Inserción de datos
-----*/

DELIMITER //

create trigger limite_telefonos # creamos trigger que se activa antes de insertar una fila en
telefono_asistente
before insert on telefono_asistente
for each row # el trigger se ejecuta una vez por cada fila
begin
    declare num_telefonos int;

    -- Contar el número de teléfonos registrados del asistente
    select count(*) into num_telefonos from telefono_asistente
    where id_asistente = new.id_asistente;

    -- Si el asistente ya tiene 3 teléfonos, cancela la inserción
    if num_telefonos >= 3 then
        signal sqlstate '45000' # instrucción para generar error
        set message_text = 'El asistente ya tiene tres números de teléfono asociados.';
    end if;
end;
//

DELIMITER ;

```

-- Insertamos datos en la tabla actividad

```
INSERT INTO actividad (nombreActividad, tipo) VALUES ('Concierto de música', 'Rock');
INSERT INTO actividad (nombreActividad, tipo) VALUES ('Concierto de musica', 'Electrónica');
INSERT INTO actividad (nombreActividad, tipo) VALUES ('Exposición', 'Arte vanguardista');
INSERT INTO actividad (nombreActividad, tipo) VALUES ('Obra de teatro', 'Musical');
INSERT INTO actividad (nombreActividad, tipo) VALUES ('Obra de teatro', 'Drama');
INSERT INTO actividad (nombreActividad, tipo) VALUES ('Conferencia', 'Tecnología');
INSERT INTO actividad (nombreActividad, tipo) VALUES ('Conferencia', 'Medio ambiente');
```

-- Insertamos datos en la tabla artista

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Blink-182', 2500.00, 'Blink-182 es una banda estadounidense de pop punk, formada el 2 de agosto de 1992 por Tom DeLonge, Mark Hoppus, y Scott Raynor en Poway, California.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Falling in Reverse', 2000.00, 'Falling In Reverse es una banda estadounidense formada a finales del año 2008 en Las Vegas, Nevada.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('David Guetta', 3500.00, 'Pierre David Guetta (París, 7 de noviembre de 1967) es un DJ, compositor electrónico y productor francés. Actualmente, ocupa el puesto #1 según la encuesta realizada por DJ Magazine.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Martin Garrix', 3200.00, 'Martin Garrix es un DJ y productor neerlandés. Garrix ha sido clasificado como uno de los mejores DJs del mundo por la revista DJ Mag.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Calvin Harris', 3800.00, 'Calvin Harris es un DJ, productor y cantante escocés. Harris ha ganado numerosos premios Grammy y ha colaborado con varios artistas internacionales.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Piet Mondrian', 1400.00, 'Piet Mondrian fue un pintor vanguardista neerlandés. Fue miembro de la corriente De Stijl y fundador del neoplasticismo.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Salvador Dalí', 1500.00, 'Salvador Dalí fue un pintor, escultor, grabador, escenógrafo y escritor español del siglo XX. Se le considera uno de los máximos representantes del surrealismo.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Pablo Picasso', 1600.00, 'Pablo Ruiz Picasso fue un pintor y escultor español, creador, junto con Georges Braque, del cubismo.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Cillian Murphy', 5000.00, 'Cillian Murphy (Cork, 25 de mayo de 1976) es un actor, actor de voz, músico y productor irlandés.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Emma Stone', 5100.00, 'Emily Jean Stone (Scottsdale, Arizona, 6 de noviembre de 1988),1 más conocida como Emma Stone, es una actriz, actriz de voz y productora de cine y televisión estadounidense.');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Jennifer Lawrence', 5300.00, 'Jennifer Shrader Lawrence (Indian Hills, Kentucky; 15 de agosto de 1990) es una actriz y productora de cine estadounidense');
```

```
INSERT INTO artista (nombreArtista, cacheArtista, biografia) VALUES ('Jeremy Allen White', 4800.00, 'Jeremy Allen White (Nueva York; 17 de febrero de 1991)1 es un actor de cine y televisión estadounidense.');
```

-- Insertamos datos en la tabla ubicacion

```
INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Cívitats metropolitan', 'Av. de Luis Aragonés, 4', 'Madrid', 60000, 45000.00, 'Estadio moderno con capacidad para grandes eventos deportivos y culturales.');
```



```

INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Teatro Real', 'Plaza de Isabel II', 'Madrid', 1500, 10000.00, 'El Teatro Real es un teatro de ópera situado en Madrid.');
```

```

INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Parque del Retiro', 'Plaza de la Independencia, 7', 'Madrid', 10000, 1000.00, 'El Parque del Retiro es uno de los principales parques de Madrid, lleno de áreas verdes, estatuas y un lago.');
```

```

INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Wizink Center', 'Av. Felipe II', 'Madrid', 17000, 25000.00, 'Wizink Center es un recinto multiusos para conciertos, eventos deportivos y espectáculos.');
```

```

INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Palau Sant Jordi', 'Passeig Olímpic, 5-7', 'Barcelona', 24000, 30000.00, 'El Palau Sant Jordi es un estadio cubierto multiusos que ha acogido conciertos de artistas internacionales y eventos deportivos.');
```

```

INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Estadio Olímpico de Sevilla', 'Av. de Eduardo Dato', 'Sevilla', 57000, 40000.00, 'El Estadio Olímpico de Sevilla es un estadio de fútbol que se utiliza para grandes conciertos y eventos.');
```

```

INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Centro de Convenciones de Barcelona', 'Plaça de Willy Brandt, 11-14', 'Barcelona', 3000, 7000.00, 'El Centro de Convenciones de Barcelona es un espacio amplio y bien equipado para conferencias y exposiciones.');
```

```

INSERT INTO ubicacion (nombreUbicacion, direccion, ciudad, aforo, precioAlquiler, características)
VALUES ('Auditorio de Barcelona', 'Carrer de Lepant, 150', 'Barcelona', 2200, 9000.00, 'El Auditorio de Barcelona es un espacio cultural polivalente para conciertos, espectáculos teatrales y eventos corporativos.');
```

-- Insertamos datos en la tabla evento

```

INSERT INTO evento (nombreEvento, fechaHora, precioEntrada, descripcion, id_ubicacion, id_actividad)
VALUES ('Concierto de rock', '2024-09-18 21:00:00', 35.00, 'Concierto de rock', 1, 1);
```

```

INSERT INTO evento (nombreEvento, fechaHora, precioEntrada, descripcion, id_ubicacion, id_actividad)
VALUES ('Festival de electronica', '2024-07-23 20:00:00', 30.00, 'Festival de electronica al aire libre con los mejores artistas mundiales', 3, 2);
```

```

INSERT INTO evento (nombreEvento, fechaHora, precioEntrada, descripcion, id_ubicacion, id_actividad)
VALUES ('Exposición de pinturas modernas', '2024-11-15 10:00:00', 10.00, 'Exposición de pinturas de artistas vanguardistas', 8, 3);
```

```

INSERT INTO evento (nombreEvento, fechaHora, precioEntrada, descripcion, id_ubicacion, id_actividad)
VALUES ('Drama en el Teatro Real', '2024-06-30 19:30:00', 25.00, 'Obra de teatro dramática dirigida por un director de renombre', 7, 5);
```

-- Insertamos datos en la tabla asistente

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Coral García', 'coral@ucm.es');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Adrian López', 'adrian@gmail.com');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Alejandro Jimenez', 'alejandro@outlook.es');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Carlos Rodriguez', 'carlos@hotmail.com');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Sara Sánchez', 'sara@gmail.com');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Laura Pérez', 'laura@outlook.es');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Daniel Yepes', 'daniel@hotmail.com');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Guillermo Pizarro', 'guille@gmail.com');
```

```

INSERT INTO asistente (nombreCompleto, email)
VALUES ('Enrique Palencia', 'kike@gmail.com');
```

-- Insertamos datos en la tabla telefono_asistente

```

INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (1, 345124590);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (1, 352654254);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (2, 345677876);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (3, 765345677);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (4, 876542345);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (5, 098765476);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (6, 456798765);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (6, 765345677);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (6, 876543234);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (7, 234568996);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (8, 234578999);
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (9, 113567888);

```

```

INSERT INTO participa VALUES (1,1),(1,2),
(2,3), (2,4), (2,5),
(3,6), (3,7),(3,8),
(5,9),(5,10),(5,12);

```

```

INSERT INTO asisteA VALUES (1,1), (1,5), (1,3), (2,7), (2,5), (3,9), (3,8), (4,2), (4,5), (4,1);

```

```

-- INSERT INTO seRealiza VALUES (1,1), (2,3), (3,8), (4,7);
-- INSERT INTO seAsocia VALUES (1,1), (2,2), (3,3), (4,5);

```

```

/*-----
Consultas, modificaciones, borrados y vistas con enunciado
-----*/

```

```

-- 1. Mostrar todos los eventos
select * from evento;

```

```

-- 2. Mostrar artistas cuyo caché sea superior a 3000
select * from artista where cacheArtista > 3000;

```

```

-- 3. Mostrar eventos que se realicen en Barcelona
select e.* from evento e
inner join ubicacion u on e.id_ubicacion = u.id_ubicacion
where u.ciudad = 'Barcelona';

```

```

-- 4. Mostrar eventos de Madrid ordenados por fecha
select e.* from evento e
inner join ubicacion u on e.id_ubicacion = u.id_ubicacion
where u.ciudad = 'Madrid'
order by e.fechaHora;

```

```

-- 5. Mostrar asistentes y sus numeros de telefono asociados
select a.nombreCompleto, group_concat(t.telefono order by t.telefono) as telefonos
from asistente a
inner join telefono_asistente t on a.id_asistente = t.id_asistente
group by a.nombreCompleto;

```

```

-- 6. Mostrar cantidad de eventos por tipo de actividad

```

```
select a.tipo, count(*) as cantidad_eventos
from actividad a
inner join evento e on a.id_actividad = e.id_actividad group by a.tipo;
```

```
-- 7. Mostrar los asistentes al festival de electronica
select a.* from asistente a
inner join asisteA aa on a.id_asistente = aa.id_asistente
inner join evento e on aa.id_evento = e.id_evento
where e.nombreEvento like '%electronica%'; # por si no sabemos como se llama exactamente, uso
like
```

```
-- 8. Mostrar eventos que se realizan en una ubicacion dada
select e.* from evento e
inner join ubicacion u on e.id_ubicacion = u.id_ubicacion
where u.nombreUbicacion like '%retiro%';
```

```
-- 9. Mostrar eventos junto con la cntidad de asistentes
select e.id_evento, e.nombreEvento, count(aa.id_asistente) as cantidad_asistentes
from evento e
left join asisteA aa on e.id_evento = aa.id_evento
group by e.id_evento;
```

```
-- 10. Mostrar dinero recaudado por cada evento
select e.id_evento, e.nombreEvento, precioEntrada, total_asistentes * e.precioEntrada as
ingresos_totales
from evento e
left join (
    select id_evento, count(id_asistente) as total_asistentes
    from asisteA
    group by id_evento
) as asistentes_por_evento on e.id_evento = asistentes_por_evento.id_evento;
```

```
-- 11. Mostrar la ubicacion mas cara y mas barata (alquiler)
select nombreUbicacion, precioAlquiler
from ubicacion
where precioAlquiler = (select max(precioAlquiler) from ubicacion)
or precioAlquiler = (select min(precioAlquiler) from ubicacion);
```

```
-- 12. Eliminar un artista
-- select * from artista;
-- set sql_safe_updates = 0;
delete from artista where nombreArtista = 'Jennifer Lawrence';
```

```
-- 13. Eliminar numero de telefono de un asistente
delete from telefono_asistente where telefono = 352654254;
-- select * from telefono_asistente;
```

```
-- 14. Modificar la fecha y hora de un evento
update evento set fechaHora = '2024-05-10 22:00:00' where nombreEvento = 'Concierto de rock';
```

```
-- 15. Mostrar asistentes que tengan 3 numeros de telefono asociados
select a.id_asistente, a.nombreCompleto, count(t.id_telefono) as num_telefonos
from asistente a
```

```
inner join telefono_asistente t on a.id_asistente = t.id_asistente
group by a.id_asistente
having num_telefonos = 3;
```

```
-- 16. Comprobamos que no podemos añadir otro numero de telefono si el asistente tiene 3:
/*
```

```
INSERT INTO telefono_asistente (id_asistente, telefono) VALUES (6, 765345677);
*/
```

```
-- 17. Vista para ver el coste total del evento
drop view if exists costeTotal_evento;
```

```
create view costeTotal_evento as
select e.id_evento,
       e.nombreEvento,
       sum(a.cacheArtista) as costeArtistas,
       e.precioEntrada * (select count(*) from asisteA aa where aa.id_evento = e.id_evento) as
gananciasEntradas,
       (sum(a.cacheArtista) + u.precioAlquiler - (e.precioEntrada * (select count(*) from asisteA aa
where aa.id_evento = e.id_evento))) as costeTotal
from evento e
inner join participa p on e.id_actividad = p.id_actividad
inner join artista a on p.id_artista = a.id_artista
inner join ubicacion u on e.id_ubicacion = u.id_ubicacion
group by e.id_evento, e.nombreEvento;
```

```
select * from costeTotal_evento;
# nos salen costes elevadísimos ya que hemos metido muy pocos asistentes
```