

Creación de una base de datos

Carlos Matías Sáez

Máster big data, data science & inteligencia artificial

2023-2024

Universidad Complutense de Madrid



Diseño Entidad-Relacion

El diseño Entidad-Relacion para nuestro caso se muestra en la Figura 1.

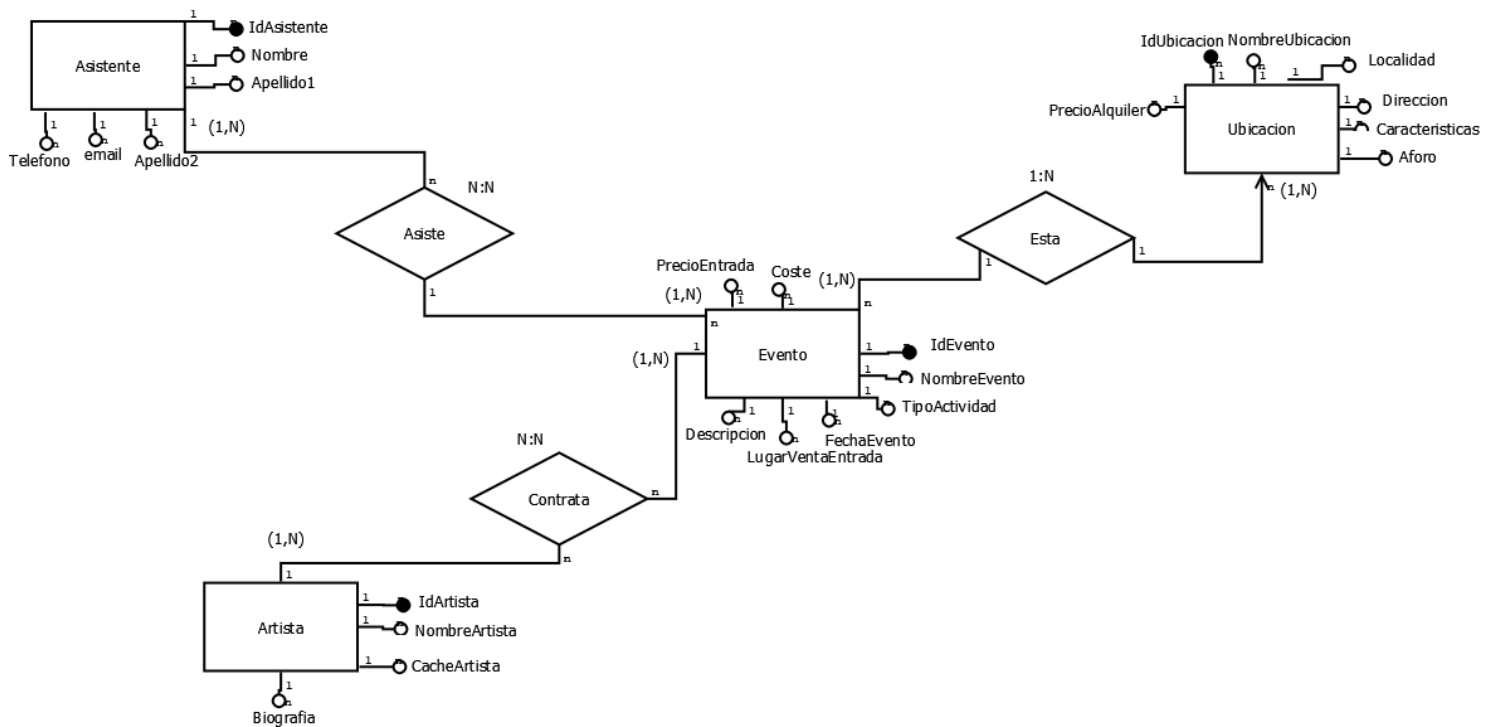


Figura 1. Diseño Entidad-Relacion

*Recaltar que Telefono es un atributo multivalorado y que debería representarse doblemente redondeado, pero no sabía cómo hacerlo con el programa DIA.

**Descripcion=DescripcionEvento.

Las entidades junto a sus respectivos atributos serán:

- **Asistente:**

Esta entidad almacena información sobre las personas que asisten a los eventos culturales. Los atributos incluyen información personal como nombre, apellidos, correo electrónico y teléfono de contacto. Se utiliza un identificador único (IdAsistente) como clave primaria para identificar a cada asistente.

- **Ubicacion:**

Esta entidad almacena información sobre las ubicaciones donde se realizan los eventos culturales. Los atributos incluyen el nombre de la ubicación, localidad, dirección, características de la ubicación, aforo (número máximo de asistentes permitidos) y

precio de alquiler de la ubicación. Se utiliza un identificador único (IdUbicacion) como clave primaria para identificar cada ubicación.

- **Artista:**

Esta entidad almacena información sobre los artistas que participan en los eventos culturales. Los atributos incluyen el nombre del artista, su tarifa (cache), y una breve biografía. Se utiliza un identificador único (IdArtista) como clave primaria para identificar a cada artista.

- **Evento:**

Esta entidad almacena información sobre los eventos culturales en sí. Los atributos incluyen el nombre del evento, el tipo de actividad, la fecha del evento, el lugar de venta de entradas, una descripción del evento, el precio de la entrada y el costo del evento. Se utiliza un identificador único (IdEvento) como clave primaria para identificar cada evento. Se establece una relación con la entidad Ubicacion a través del atributo CodUbicacion, que actúa como clave externa para indicar dónde se llevará a cabo el evento.

- **Asiste:**

Esta entidad es una tabla de relación que conecta asistentes con eventos. Se utiliza para registrar qué asistentes están inscritos para asistir a qué eventos. Las claves primarias combinadas (CodAsistente y CodEvento) actúan como clave primaria compuesta. Se establecen dos relaciones foráneas con las entidades Asistente y Evento para asegurar que los registros de asistencia estén relacionados con los registros de asistentes y eventos correspondientes.

- **Contrata:**

Esta entidad es una tabla de relación que conecta artistas con eventos. Se utiliza para registrar qué artistas han sido contratados para actuar en qué eventos. Las claves primarias combinadas (CodArtista y CodigoEvento) actúan como clave primaria compuesta. Se establecen dos relaciones foráneas con las entidades Artista y Evento para asegurar que los registros de contratación estén relacionados con los registros de artistas y eventos correspondientes.

Los atributos de cada entidad podemos observarlos en la figura 1, donde el círculo relleno negro representa la clave primaria de la entidad y la vacía blanca el resto de atributos.

Las relaciones serán:

- **Asiste:** Relación N:N, ya que un asistente puede ir a varios eventos (N) y a un evento pueden ir varios asistentes (N).
- **Contrata:** Relación N:N, ya que un artista puede ser contratado por varios eventos (N) y en un evento puede contratar a varios artistas (N).

- **Esta:** Relación 1:N, ya que en una ubicación puede haber varios eventos (N) pero un evento solo puede estar en una ubicación (1).

El modelo entidad-relación es una representación simplificada de una base de datos que se utiliza para definir la estructura de las entidades, sus atributos y las relaciones entre ellas. Sin embargo, hay ciertos conceptos o detalles que no pueden expresarse directamente en una modelización entidad-relación y que requerirían una descripción más detallada para ser comprendidos por completo. Algunos de estos conceptos incluyen:

- **Reglas de negocio complejas:** El modelo entidad-relación es estático y no permite expresar reglas de negocio complejas o lógica de aplicación. Las restricciones de negocio, como las reglas de validación o los flujos de trabajo específicos, generalmente se documentan por separado en documentos de diseño o en el código de la aplicación.
- **Secuenciación de eventos o actividades:** El modelo entidad-relación no describe la secuencia en la que ocurren los eventos o actividades. No puedes representar fácilmente la cronología de eventos o las dependencias temporales entre ellos en este tipo de modelo.
- **Valores predeterminados o comportamiento por defecto:** No puedes especificar valores predeterminados para atributos en el modelo entidad-relación. Los valores predeterminados y el comportamiento por defecto suelen ser implementados a nivel de la base de datos o en la lógica de la aplicación.
- **Seguridad y control de acceso:** El modelo entidad-relación no aborda cuestiones de seguridad y control de acceso. Las políticas de seguridad, como quién tiene acceso a qué datos y bajo qué condiciones, se gestionan a nivel de la base de datos y en la lógica de la aplicación.

Matrices personales que no se han tenido en cuenta según el enunciado:

1. Suponemos que un artista puede ir a varios eventos.
2. Suponemos que los eventos no son online y necesitan una ubicación.
3. Hemos usado identificadores para las entidades en vez de sus nombres para que no haya confusiones. Ej: Para la ubicación puede haber confusiones como en el caso de Córdoba (Argentina) y Córdoba (España).

Modelo Relacional

El modelo relacional podemos observarlo en la Figura 2. Hemos transformado el diagrama entidad-relación anterior en su correspondiente modelo relacional.

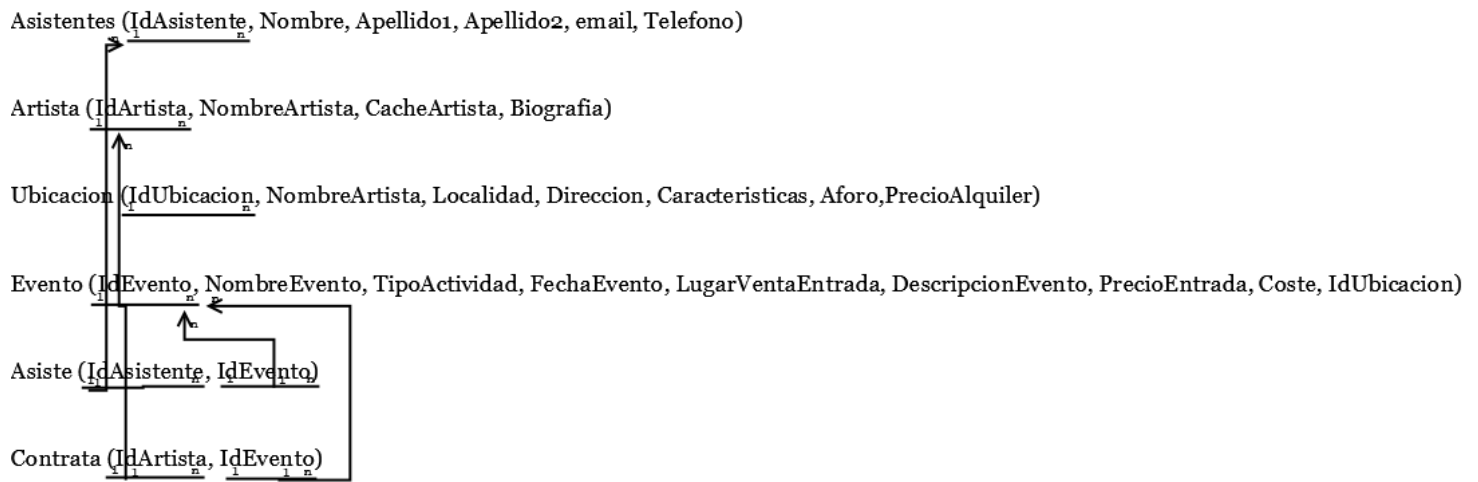


Figura 2. Modelo relacional

Script

```
DROP DATABASE IF EXISTS BDArteVidaCultural;
```

```
-- Creamos la base de datos
```

```
CREATE DATABASE BDArteVidaCultural;
```

```
USE BDArteVidaCultural;
```

```
-- Creamos las tablas
```

```
CREATE TABLE Asistente (
```

```
    IdAsistente char(6),
```

```
    Nombre varchar (40) not null,
```

```
    Apellido1 varchar(40) not null,
```

```
    Apellido2 varchar(40) not null,
```

```
    email varchar(40) not null,
```

```
    telefono numeric(9,0) not null,
```

```
    PRIMARY KEY (IdAsistente)
```

```
);
```

```
CREATE TABLE Ubicacion (
```

```
    IdUbicacion char(6),
```

```
    NombreUbicacion varchar(40) not null,
```

```
    Localidad varchar(40) not null,
```

```
    Direccion varchar(40) not null,
```

```
    Caracteristicas varchar(120) not null,
```

```
    Aforo integer not null,
```

```
    PrecioAlquiler numeric(8,2) not null,
```

```
    PRIMARY KEY (IdUbicacion)
```

```
);
```

```
CREATE TABLE Artista (  
    IdArtista char(6),  
    NombreArtista varchar(40) not null,  
    CacheArtista numeric(7,2) not null,  
    Biografia varchar(120) not null,  
    PRIMARY KEY (IdArtista)  
);
```

```
CREATE TABLE Evento (  
    IdEvento char(6),  
    NombreEvento varchar(40) not null,  
    TipoActividad varchar (40) not null,  
    FechaEvento date not null,  
    LugarVentaEntrada varchar(100) not null,  
    DescripcionEvento varchar(100) not null,  
    PrecioEntrada integer not null,  
    Coste decimal (10,2),  
    CodUbicacion char(6),  
    PRIMARY KEY (IdEvento),  
    FOREIGN KEY (CodUbicacion) REFERENCES Ubicacion(IdUbicacion) ON DELETE cascade  
);
```

```
CREATE TABLE Asiste (  
    CodAsistente char(6),  
    CodEvento char(6),  
    PRIMARY KEY (CodAsistente, CodEvento),  
    FOREIGN KEY (CodAsistente) REFERENCES Asistente(IdAsistente) ON DELETE cascade,  
    FOREIGN KEY (CodEvento) REFERENCES Evento(IdEvento) ON DELETE cascade  
);
```

```
CREATE TABLE Contrata (  
    CodArtista char(6),  
    CodigoEvento char(6),  
    PRIMARY KEY (CodArtista, CodigoEvento),  
    FOREIGN KEY (CodArtista) REFERENCES Artista(IdArtista) ON DELETE cascade,  
    FOREIGN KEY (CodigoEvento) REFERENCES Evento(IdEvento) ON DELETE cascade  
);
```

-- INTRODUCIMOS VALORES PARA LAS TABLAS

-- Datos para Asistente:

```
INSERT INTO Asistente VALUES ('000001', 'Gabriel', 'Garcia', 'Higuera', 'aleg23@gmail.com',  
935432859);
```

```
INSERT INTO Asistente VALUES ('000002', 'Juan', 'Saez', 'Cortes', 'ju12@gmail.com',  
937845648);
```

```
INSERT INTO Asistente VALUES ('000003', 'Pedro', 'Serrano', 'Rueda', 'a23@gmail.com',  
937844657);
```

```
INSERT INTO Asistente VALUES ('000004', 'Maria', 'Parejo', 'Casquet', 'mar22@gmail.com',  
937844658);
```

```
INSERT INTO Asistente VALUES ('000005', 'Clara', 'Encinas', 'Saez', 'a23@gmail.com',  
937876592);
```

```
INSERT INTO Asistente VALUES ('000006', 'Rocio', 'Ruiz', 'Matias', 'dsf34@gmail.com',  
937876523);
```

```
INSERT INTO Asistente VALUES ('000007', 'Ana', 'Encinas', 'Ruiz', 'dsf54@gmail.com',  
937876555);
```

```
INSERT INTO Asistente VALUES ('000008', 'Isabel', 'Garcia', 'Higuera', 'sdg65@gmail.com',  
937874492);
```

```
INSERT INTO Asistente VALUES ('000009', 'Lucia', 'Encinas', 'Parejo', 'gfh56@gmail.com',  
937874592);
```

```
INSERT INTO Asistente VALUES ('000010', 'Maria', 'Fernandez', 'Turus', 'tyu76@gmail.com',  
937812592);
```

```
INSERT INTO Asistente VALUES ('000011', 'Encarnacion', 'Cortes', 'Catre', 'gdh76@gmail.com',  
978876592);
```

```
INSERT INTO Asistente VALUES ('000012', 'Rosario', 'Gallego', 'Garcia', 'cvb66@gmail.com',  
937406592);
```


INSERT INTO Asistente VALUES ('000013', 'Africa', 'Fernandez', 'Ruiz', 'gfh999@gmail.com', 944476592);

INSERT INTO Asistente VALUES ('000014', 'Estrella', 'Rueda', 'Molina', 'ytu11@gmail.com', 937879834);

INSERT INTO Asistente VALUES ('000015', 'Rocio', 'Parra', 'Moreno', 'fgh22@gmail.com', 937886340);

INSERT INTO Asistente VALUES ('000016', 'Carmen', 'Palomo', 'Merced', 'poi44@gmail.com', 937230392);

INSERT INTO Asistente VALUES ('000017', 'Antonio', 'Cerezo', 'Angustias', 'iurt66@gmail.com', 937820092);

INSERT INTO Asistente VALUES ('000018', 'Carlos', 'Matias', 'Saez', 'suxarly22@gmail.com', 937893522);

-- Datos para Ubicacion:

INSERT INTO Ubicacion VALUES ('000001', 'Playa Cabria', 'Granada', 'Rio Lozoya 35', 'Amplio recinto cerca de la playa', 5000, 10000.00);

INSERT INTO Ubicacion VALUES ('000002', 'Marenostrum', 'Cordoba', 'Luna 11', 'Recinto mediano con vistas al monte', 1000, 5500.56);

INSERT INTO Ubicacion VALUES ('000003', 'Polideportivo Rocio', 'Huelva', 'Lucero 33', 'Polideportivo en el centro de la ciudad', 3000, 15000.44);

INSERT INTO Ubicacion VALUES ('000004', 'Plaza España', 'Bilbao', 'Perpetuo Socorro 12', 'Plaza pequeña para un escenario pequeño', 400, 20000.87);

INSERT INTO Ubicacion VALUES ('000005', 'Teatro de luz', 'Cadiz', 'Virgen Angustias 11', 'Amplio teatro en el centro de Cadiz', 2000, 12000.32);

-- Datos para Artista:

INSERT INTO Artista VALUES ('000001', 'Feid', '20000.00', 'Reggaetonero nacido en Puerto Rico');

INSERT INTO Artista VALUES ('000002', 'Los Planetas', '5000.00', 'Grupo de Rock Granadino');

INSERT INTO Artista VALUES ('000003', 'Shakira', '10000.00', 'Cantante de musica latina');

INSERT INTO Artista VALUES ('000004', 'Rosalia', '15000.00', 'Cantante catalana especializada en musica española');

INSERT INTO Artista VALUES ('000005', 'El Cigala', '3000.00', 'Cantaor de Flamenco español');

INSERT INTO Artista VALUES ('000006', 'Javier Santaolalla', '500.00', 'Divulgador de Fisica');

```
INSERT INTO Artista VALUES ('000007', 'Antonio Banderas', '700.00', 'Actor de teatro');
```

```
INSERT INTO Artista VALUES ('000008', 'Rels B', '800.00', 'Cantante HipHop');
```

-- Datos para el Evento:

```
INSERT INTO Evento (IdEvento, NombreEvento, TipoActividad, FechaEvento,  
LugarVentaEntrada, DescripcionEvento, PrecioEntrada, CodUbicacion)
```

```
VALUES ('000001', 'SuperFest', 'Concierto', '2023-03-22', 'Calle Gaona 12', 'Concierto musica  
Latina', 150, '000001');
```

```
INSERT INTO Evento (IdEvento, NombreEvento, TipoActividad, FechaEvento,  
LugarVentaEntrada, DescripcionEvento, PrecioEntrada, CodUbicacion)
```

```
VALUES ('000002', 'RockFlamenco', 'Concierto', '2023-05-11', 'Calle Real 22r', 'Concierto de  
musica fusion flamenco-rock', 200, '000002');
```

```
INSERT INTO Evento (IdEvento, NombreEvento, TipoActividad, FechaEvento,  
LugarVentaEntrada, DescripcionEvento, PrecioEntrada, CodUbicacion)
```

```
VALUES ('000003', 'LaBandera', 'Teatro', '2023-06-27', 'Calle Rivera 9', 'Teatro dirigido por  
Antonio Banderas', 178, '000003');
```

```
INSERT INTO Evento (IdEvento, NombreEvento, TipoActividad, FechaEvento,  
LugarVentaEntrada, DescripcionEvento, PrecioEntrada, CodUbicacion)
```

```
VALUES ('000004', 'ExpoScience', 'Exposicion', '2023-04-12', 'Calle Amargura 16', 'Exposicion de  
divulgacion cientifica', 30, '000004');
```

```
INSERT INTO Evento (IdEvento, NombreEvento, TipoActividad, FechaEvento,  
LugarVentaEntrada, DescripcionEvento, PrecioEntrada, CodUbicacion)
```

```
VALUES ('000005', 'RosaliaFest', 'Concierto', '2023-09-22', 'Calle Luz 44', 'Concierto exclusivo de  
Rosalia', 120, '000005');
```

-- Datos para Contrata:

-- Contratacion para SuperFest

```
INSERT INTO Contrata VALUES ('000001', '000001');
```

```
INSERT INTO Contrata VALUES ('000003', '000001');
```

-- Contratacion para RockFlamenco

```
INSERT INTO Contrata VALUES ('000002', '000002');
```

```
INSERT INTO Contrata VALUES ('000005', '000002');
```

-- Contratacion para LaBandera

```
INSERT INTO Contrata VALUES ('000007','000003');  
  
-- Contratacion para ExpoScience  
  
INSERT INTO Contrata VALUES ('000006','000004');  
  
-- Contratacion para RosaliaFest  
  
INSERT INTO Contrata VALUES ('000004','000005');
```

```
-- Datos para Asiste
```

```
-- Asisten a SuperFest
```

```
INSERT INTO Asiste VALUES ('000001','000001');  
INSERT INTO Asiste VALUES ('000002','000001');  
INSERT INTO Asiste VALUES ('000003','000001');  
INSERT INTO Asiste VALUES ('000004','000001');  
INSERT INTO Asiste VALUES ('000005','000001');  
INSERT INTO Asiste VALUES ('000006','000001');  
INSERT INTO Asiste VALUES ('000007','000001');  
INSERT INTO Asiste VALUES ('000008','000001');  
INSERT INTO Asiste VALUES ('000009','000001');  
INSERT INTO Asiste VALUES ('000010','000001');  
INSERT INTO Asiste VALUES ('000011','000001');  
INSERT INTO Asiste VALUES ('000012','000001');  
INSERT INTO Asiste VALUES ('000013','000001');  
INSERT INTO Asiste VALUES ('000014','000001');  
INSERT INTO Asiste VALUES ('000015','000001');
```

```
-- Asisten a RockFlamenco
```

```
INSERT INTO Asiste VALUES ('000001','000002');  
INSERT INTO Asiste VALUES ('000002','000002');  
INSERT INTO Asiste VALUES ('000003','000002');  
INSERT INTO Asiste VALUES ('000004','000002');  
INSERT INTO Asiste VALUES ('000005','000002');  
INSERT INTO Asiste VALUES ('000007','000002');  
INSERT INTO Asiste VALUES ('000009','000002');
```

INSERT INTO Asiste VALUES ('000010','000002');

-- Asisten a LaBandera

INSERT INTO Asiste VALUES ('000006','000003');

INSERT INTO Asiste VALUES ('000007','000003');

INSERT INTO Asiste VALUES ('000008','000003');

INSERT INTO Asiste VALUES ('000009','000003');

INSERT INTO Asiste VALUES ('000010','000003');

INSERT INTO Asiste VALUES ('000011','000003');

INSERT INTO Asiste VALUES ('000012','000003');

INSERT INTO Asiste VALUES ('000013','000003');

INSERT INTO Asiste VALUES ('000014','000003');

INSERT INTO Asiste VALUES ('000016','000003');

INSERT INTO Asiste VALUES ('000017','000003');

INSERT INTO Asiste VALUES ('000003','000003');

-- Asisten a ExpoScience

INSERT INTO Asiste VALUES ('000003','000004');

INSERT INTO Asiste VALUES ('000007','000004');

INSERT INTO Asiste VALUES ('000011','000004');

INSERT INTO Asiste VALUES ('000013','000004');

INSERT INTO Asiste VALUES ('000015','000004');

INSERT INTO Asiste VALUES ('000017','000004');

INSERT INTO Asiste VALUES ('000002','000004');

-- Asisten a RosaliaFest

INSERT INTO Asiste VALUES ('000001','000005');

INSERT INTO Asiste VALUES ('000002','000005');

INSERT INTO Asiste VALUES ('000004','000005');

INSERT INTO Asiste VALUES ('000006','000005');

INSERT INTO Asiste VALUES ('000007','000005');

INSERT INTO Asiste VALUES ('000008','000005');

INSERT INTO Asiste VALUES ('000009','000005');

INSERT INTO Asiste VALUES ('000010','000005');

```
INSERT INTO Asiste VALUES ('000011','000005');
INSERT INTO Asiste VALUES ('000012','000005');
INSERT INTO Asiste VALUES ('000013','000005');
INSERT INTO Asiste VALUES ('000014','000005');
INSERT INTO Asiste VALUES ('000016','000005');
INSERT INTO Asiste VALUES ('000017','000005');
```

-- CONSULTAS DE LA BASE DE DATOS

-- 1. Debido a la inflacion el precio de las entradas podría aumentar un 25%, por ello se calcula el precio estimado de

-- la entrada en ese caso:

```
select NombreEvento, PrecioEntrada, PrecioEntrada*1.25 as PrecioInflacion
from evento;
```

-- 2. Una persona tiene como maximo 50 euros para gastar en un evento, por tanto busca aquellos que pueda costearse:

```
select NombreEvento, PrecioEntrada
from Evento
where PrecioEntrada between 0 and 50;
```

-- 3. Queremos ver el nombre de los asistentes y los eventos a los que asisten

```
select IdAsistente, Nombre, Apellido1, Apellido2, CodEvento
from Asistente inner join Asiste on Asistente.IdAsistente=Asiste.CodAsistente;
```

-- 4. Con motivo de obtener las máximas ganancias posibles, queremos asegurarnos de que todas las personas

-- que tenemos registradas en Asistentes hayan asistido por lo menos a un evento, por ello vemos quienes no han asistido a ninguno para

-- poder ofrecerle un evento al cual pueda asistir.

```
select IdAsistente, Nombre, Apellido1, Apellido2
from Asistente left join Asiste on Asistente.IdAsistente=Asiste.CodAsistente
```

where CodAsistente is null;

-- VISTA: calcula el costo total de un evento con varios artistas

CREATE OR REPLACE VIEW VistaCostoEvento AS

SELECT

e.IdEvento,

e.NombreEvento,

(SUM(a.CacheArtista) + u.PrecioAlquiler) AS CostoTotalEvento

FROM

Evento e

INNER JOIN

Ubicacion u ON e.CodUbicacion = u.IdUbicacion

LEFT JOIN

Contrata c ON e.IdEvento = c.CodigoEvento

LEFT JOIN

Artista a ON c.CodArtista = a.IdArtista

GROUP BY

e.IdEvento, e.NombreEvento, u.PrecioAlquiler;

-- 5. Consulta para obtener el costo total de un evento específico el cual sera la suma del PrecioAlquiler mas CacheArtista. En este caso consultamos el precio del concierto "RockFest", con Id 000002:

SELECT * FROM VistaCostoEvento WHERE IdEvento = '000002';

-- 6. Queremos ver cuales son los asistentes que han asistido a más de un evento:

SELECT A1.Nombre, A1.Apellido1, A1.Apellido2

FROM Asistente A1

JOIN Asiste AS A2 ON A1.IdAsistente = A2.CodAsistente

GROUP BY A1.IdAsistente, A1.Nombre, A1.Apellido1, A1.Apellido2

HAVING COUNT(A2.CodEvento) > 1;

-- 7. Mostrar la ubicación de un evento específico:

```
SELECT NombreEvento, NombreUbicacion, Localidad, Direccion
FROM Ubicacion
INNER JOIN Evento ON IdUbicacion = CodUbicacion;
```

-- 8. Listar todos los artistas que han actuado en un evento específico. Vemos como el artista "Rels B" no ha asistido a ningún evento ya que no aparece en la consulta:

```
SELECT NombreArtista, NombreEvento
FROM Artista
INNER JOIN Contrata ON IdArtista = CodArtista
INNER JOIN Evento ON CodigoEvento = IdEvento;
```

-- 9. Queremos ver que número de cada tipo de eventos hay disponibles:

```
SELECT TipoActividad, COUNT(*) AS CantidadEventos
FROM Evento
GROUP BY TipoActividad;
```

-- 10. Muestra los artistas que han actuado en evento de tipo "concierto":

```
SELECT DISTINCT Ar.NombreArtista
FROM Artista Ar
INNER JOIN Contrata C ON Ar.IdArtista = C.CodArtista
INNER JOIN Evento E ON C.CodigoEvento = E.IdEvento
WHERE E.TipoActividad = 'Concierto';
```

-- TRIGGER: Para asegurarnos de que el precio de la entrada del evento sea una cantidad mayor o igual que 0 y no haya fallos a la hora de la venta de entradas, hacemos un trigger que compruebe esto:

```
DELIMITER //

CREATE TRIGGER VerificarPrecioEntradaTrigger BEFORE INSERT ON Evento
FOR EACH ROW
BEGIN
    IF NEW.PrecioEntrada <= 0 THEN
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'El precio de la entrada debe ser mayor que 0';  
    END IF;  
END;  
//  
DELIMITER ;
```