

Nombre: Jalevy Senar Trujillo

Profesor: Jhon Corredor

Institución:

Fecha:

Materia:

Curso:

Nota:

## 4: Eventos y Ticketing

DDL:

**CREATE DATABASE** Eventos  
**USE** Eventos;

**CREATE TABLE** lugar (

id-lugar INT PRIMARY KEY AUTO-INCREMENT,  
nombre-lugar VARCHAR (150) NOT NULL UNIQUE,  
direccion VARCHAR (255) NOT NULL,  
capacidad-maxima INT NOT NULL,  
tipo-lugar VARCHAR (50),  
telefono-contacto VARCHAR (20),  
email-contacto VARCHAR (100)

);

**CREATE TABLE** TipoEntrada (

id-tipo-entrada INT PRIMARY KEY AUTO-INCREMENT,  
nombre-tipo VARCHAR (50) NOT NULL,  
precio-base DECIMAL (10, 2) NOT NULL,  
descripcion TEXT,  
limite-por-compra INT

);

**CREATE TABLE** Evento (

id-evento INT PRIMARY KEY AUTO-INCREMENT,  
nombre-evento VARCHAR (255) NOT NULL,  
descripcion TEXT,  
fecha-hora-inicio DATETIME NOT NULL,  
fecha-hora-fin DATETIME,  
id-lugar INT NOT NULL,  
estado-evento VARCHAR (50) NOT NULL DEFAULT 'programado',  
FOREIGN KEY (id-lugar) REFERENCES lugar (id-lugar)

);

**CREATE TABLE** Artista (

id-artista INT PRIMARY KEY AUTO-INCREMENT,  
nombre-artista VARCHAR (150) NOT NULL,  
genero-musical-o-actuacion VARCHAR (100),  
nacionalidad VARCHAR (50),  
sitio-web VARCHAR (255)

);



## CREATE TABLE Usuario (

id-usuario INT PRIMARY KEY AUTO-INCREMENT,  
nombre-usuario VARCHAR(100) NOT NULL,  
apellido-usuario VARCHAR(100) NOT NULL,  
email VARCHAR(100) NOT NULL UNIQUE,  
telefono VARCHAR(20),  
fecha-registro DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP

);

## CREATE TABLE Participacion (

id-participacion INT PRIMARY KEY AUTO-INCREMENT,  
id-evento INT NOT NULL,  
id-artista INT NOT NULL,  
rol-en-evento VARCHAR(50),  
hora-actuacion TIME,  
FOREIGN KEY (id-evento) REFERENCES Evento (id-evento),  
FOREIGN KEY (id-artista) REFERENCES Artista (id-artista),  
UNIQUE (id-evento, id-artista)

);

## CREATE TABLE Ticket (

id-ticket INT PRIMARY KEY AUTO-INCREMENT,  
id-usuario INT NOT NULL,  
id-tipo-entrada INT NOT NULL,  
id-evento INT NOT NULL,  
fecha-compra DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
precio-pagado VARCHAR(10,2) NOT NULL,  
estado-ticket VARCHAR(50) NOT NULL DEFAULT 'Válido',  
codigo-qr VARCHAR(255) UNIQUE NOT NULL,  
FOREIGN KEY (id-usuario) REFERENCES Usuario (id-usuario),  
FOREIGN KEY (id-tipo-entrada) REFERENCES TipoEntrada (id-tipo-entrada),  
FOREIGN KEY (id-evento) REFERENCES Evento (id-evento)

);

## DML:

INSERT INTO lugar (nombre-lugar, direccion, capacidad-maxima, tipo-lugar, telefono-contacto, email-contacto) VALUES

('Coliseo Iguera', 'Carretera 10 # 5-10, Iguera', 5000, 'Estadio', '3101112233', 'Contacto@coliseo:iguera.com');



```
INSERT INTO TipoEntrada (nombre-tipo, precio-base, descripcion,  
lembete-por-compra) VALUES
```

```
('General', 50000.00, 'Acceso general al evento', 5);
```

```
INSERT INTO Evento (nombre-evento, descripcion, fecha-hora-  
inicio, fecha-hora-fin, id-lugar, estado-evento) VALUES
```

```
('Obra de teatro clásica', 'Representación de una obra  
dramática famosa.', '2025-09-05 20:00:00', '2025-09-05  
22:00:00', 2, 'Programado');
```

```
INSERT INTO Artista (nombre-artista, genero-musical-o-actuacion,  
nacionalidad, sitio-web) VALUES
```

```
('Teatro vivo', 'Drama clásico', 'Colombiana', 'www.teatrovivo.org');
```

```
INSERT INTO Usuario (nombre-usuario, apellido-usuario, email,  
telefono) VALUES
```

```
('María Pérez', 'maria.perez@example.com', '3101234567');
```

```
INSERT INTO ParticipacionArtista (id-evento, id-artista,  
rol-en-evento, hora-actuacion) VALUES
```

```
(2, 2, 'Principal', '20:00:00');
```

```
INSERT INTO Ticket (id-usuario, id-tipo-entrada, id-evento,  
precio-pagado, codigo-qr) VALUES
```

```
(2, 1, 2, 50000.00, 'QR-TEATRO-001-GHI');
```

## Consulta

### 1. SELECT

```
SELECT  
    nombre-lugar,  
    direccion  
FROM  
    lugar;
```

### 2. JOIN

```
SELECT  
    E.nombre-evento,  
    L.nombre-lugar,  
    L.direccion  
FROM  
    Evento E  
JOIN  
    lugar L ON E.id-lugar = L.id-lugar
```



### 3º GROUP BY y COUNT

```
SELECT
    TG.nombre_tipo,
    COUNT (T.id_ticket) AS total_tickets_vendidos
FROM
    Ticket T
JOIN
    TipoEntrada TE ON T.id_tipo_entrada = TE.id_tipo_entrada
GROUP BY
    TG.nombre_tipo
ORDER BY
    total_tickets_vendidos DESC;
```

### 4º HAVING y COUNT

```
SELECT
    U.nombre_usuario,
    U.apellido_usuario,
    COUNT (T.id_ticket) AS total_tickets_comprados
FROM
    Usuario U
JOIN
    Ticket T ON U.id_usuario = T.id_usuario
GROUP BY
    U.id_usuario, U.nombre_usuario, U.apellido_usuario
HAVING
    COUNT (T.id_ticket) > 1;
```

### 5º AVG

```
SELECT
    AVG (precio_pagado) AS precio_promedio_ticket
FROM
    Ticket;
```

### 6º DISTINCT

```
SELECT DISTINCT
    genero_musical - O - actuacion
FROM
    Artista;
```

### 7º subconsulta

```
SELECT
    A.nombre_artista
FROM
    Artista A
WHERE
    A.id_artista IN (
        SELECT PA.id_artista
        FROM ParticipacionArtista PA
        JOIN Evento E ON PA.id_evento = E.id_evento
        JOIN Lugar L ON E.id_lugar = L.id_lugar
        WHERE L.capacidad > 1000);
```



Nombre:

Profesor:

Institución:

Fecha:

Materia:

Curso:

Página:

## Sentencias

### 1. UPDATE

```
UPDATE Evento  
SET estado_evento = 'Cancelado'  
WHERE nombre_evento = 'obra de teatro clásica';
```

### 2. DELETE

```
DELETE FROM TipoEntrada WHERE nombre_tipo = 'platino';
```

## Procedimiento

```
CREATE PROCEDURE RegistrarCompreTicketSimple (  
  IN p_id_usuario INT,  
  IN p_id_tipo_entrada INT,  
  IN p_id_evento INT,  
  IN p_codigo_qr VARCHAR(255),  
  OUT p_mensaje VARCHAR(255)  
)
```

BEGIN

```
DECLARE v_usuario_existe INT;  
DECLARE v_tipo_entrada_existe INT;  
DECLARE v_evento_existe INT;
```

```
SELECT COUNT(*) INTO v_usuario_existe FROM Usuario WHERE  
id_usuario = p_id_usuario;  
SELECT COUNT(*) INTO v_tipo_entrada_existe FROM TipoEntrada  
WHERE id_tipo_entrada = p_id_tipo_entrada;  
SELECT COUNT(*) INTO v_evento_existe FROM Evento WHERE  
id_evento = p_id_evento;
```

```
IF v_usuario_existe = 0 THEN  
  SET p_mensaje = 'Error: El ID de usuario no existe.';  
ELSE IF v_tipo_entrada_existe = 0 THEN  
  SET p_mensaje = 'Error: El ID de tipo de entrada no existe.';  
ELSE IF v_evento_existe = 0 THEN  
  SET p_mensaje = 'Error: El ID del evento no existe.';
```

ELSE

```
INSERT INTO Ticket (id_usuario, id_tipo_entrada, id_evento,  
precio_pagado, codigo_qr, estado_ticket)
```

```
VALUES (p_id_usuario, p_id_tipo_entrada, p_id_evento,  
p_precio_pagado, p_codigo_qr, 'Valido');
```



SET p-message = 'Ticket registrado con éxito.';

END IF

END



Nombre de Tabla:

Tipo Entrada

Descripción:

Entidad

almacenada

la

información

tipo entrada

Fecha de Creación: 10/07/2025

Clave

Campo

Tipo de Dato

Tamaño

Descripción

PK

id-tipo-entrada

INT

nombre-tipo

VARCHAR

50

precio-base

DECIMAL

10,2

descripcion

TEXT

limite-por-compra

INT

Identificador unico de tipo entrada

Nombre de tipo entrada

Precio base

descripcion

Limite por Compra



Nombre de Tabla: Evento

Fecha de creación: 10/07/2025

Descripción: Entidad implementada la información del Evento

Clave	Campo	Tipo de dato	Ordenado	Descripción
PK	id - evento	INT		Identificador único del evento
	nombre - evento	VARCHAR	255	Nombre del evento
	descripción	TEXT		Descripción del evento
	fecha, hora, inicio	DATETIME		Fecha y hora de inicio del evento
	fecha, hora, fin	DATETIME		Fecha y hora final del evento
FK	id - lugar	INT		
	estado - evento	VARCHAR	30	Estado del evento



# Nombre de tabla: Artista

Fecha creación: 10/07/2025

## Descripción: Entidad

que almacenará la información del artista

Clave	Campo	Tipo de dato	Tamaño	Descripción
PK	id-artista	INT		Identificador único del artista
	nombre-artista	VARCHAR	150	Nombre del artista
	genero-musical	VARCHAR	100	Genero musical
	nacionalidad	VARCHAR	50	Nacionalidad del artista
	seteo-web	VARCHAR	255	seteo web



Nombre de Tabla: Usuario

Fecha de Creación: 10/07/2025

Descripción: En Tidad que almacenará la información del usuario

Clave	Campo	tipo de dato	Tamaño	Descripción
PK	id_usuario	INT		Identificador único del usuario
	nombre_usuario	VARCHAR	100	Nombre del usuario
	apellido_usuario	VARCHAR	100	Apellido del usuario
	email	VARCHAR	100	Email del usuario
	telefono	VARCHAR	20	Nº de telefono del usuario
	fecha_registro	DATETIME		Fecha registro



Nombre de Tabla: ParticipacionArtista

Fecha Creación: 10/07/2025

Descripción: Entidad que almacena la participación del artista

Clave Campo

Tipo de Dato

Tamaño

Descripción

PK

id-participacion

INT

Identificador único participación

FK

id-evento

INT

FK

id-artista

INT

rol-en-evento

VARCHAR

50

Rol en evento

hora-actuacion

TIME

Hora de actuación



Nombre de tabla: Ticket

Fecha de creación: 10/07/2025

Descripción: En la tabla se almacena la información del ticket

Clase	Campo	tipo de dato	Tamaño	Descripción
PK	id-ticket	INT		Identificador único del ticket
FK	id-usuario	INT		
FK	id-tipo-entrada	INT		
FK	id-evento	INT		
	fecha-compra	DATE TIME		Fecha y compra del ticket
	precio-pagado	NUMERIC	10,2	Precio pagado por el ticket
	estado-ticket	VARCHAR	50	Estado del ticket
	Código QR	VARCHAR	255	Código QR



Lugar
id_lugar (PK)
nombre_lugar
descripcion
capacidad_maxima
tipo_lugar
telefono_contacto
email_contacto

TipoEntrada
id_tipo_entrada (PK)
nombre_tipo
precio_base
descripcion
motor_pago_compra

Artista
id_artista (PK)
nombre_artista
genero_musical
o_actividad
nacionalidad
sitio_web

Evento
id_evento (PK)
nombre_evento
descripcion
fecha_hora_inicio
fecha_hora_fin
id_lugar (FK)
estado_evento

ParticipacionArtista
id_participacion (PK)
id_evento (FK)
id_artista (FK)
rol_en_evento
hora_activacion

Usuario
id_usuario (PK)
nombre_usuario
apellido_usuario
email
telefono
fecha_registro

Ticket
id_ticket (PK)
id_usuario (FK)
id_tipo_entrada (FK)
id_evento (FK)
fecha_compra
precio_pago
estado_ticket
codigo_qr

