CREATE TABLE EVENT(  
EVENT\_ID INT PRIMARY KEY AUTO\_INCREMENT,  
EVENT\_NAME VARCHAR(255),  
EVENT\_DATE DATE,

EVENT\_TIME TIME  
);

INSERT INTO EVENT (EVENT\_ID, EVENT\_NAME, EVENT\_DATE, EVENT\_TIME) VALUES

(1, 'Tech Conference', '2025-08-05', '09:00:00'),

(2, 'Art Expo', '2025-08-06', '10:30:00'),

(3, 'Music Festival', '2025-08-07', '15:00:00'),

(4, 'Startup Pitch', '2025-08-08', '14:00:00'),

(5, 'Book Fair', '2025-08-09', '11:15:00'),

(6, 'Science Exhibition', '2025-08-10', '13:30:00'),

(7, 'Food Carnival', '2025-08-11', '12:45:00'),

(8, 'Film Screening', '2025-08-12', '18:00:00'),

(9, 'Job Fair', '2025-08-13', '09:00:00'),

(10, 'Photography Workshop', '2025-08-14', '10:00:00'),

(11, 'Dance Show', '2025-08-15', '17:00:00'),

(12, 'Coding Bootcamp', '2025-08-16', '08:00:00'),

(13, 'Chess Tournament', '2025-08-17', '10:30:00'),

(14, 'Hackathon', '2025-08-18', '09:30:00'),

(15, 'Open Mic Night', '2025-08-19', '19:00:00'),

(16, 'Marathon', '2025-08-20', '06:00:00'),

(17, 'Charity Gala', '2025-08-21', '20:00:00'),

(18, 'Fashion Show', '2025-08-22', '17:30:00'),

(19, 'Product Launch', '2025-08-23', '14:30:00'),

(20, 'Yoga Retreat', '2025-08-24', '07:00:00');

CREATE TABLE ATTENDEE (

ATTENDEE\_ID INT AUTO\_INCREMENT PRIMARY KEY,

NAME VARCHAR(255),

PH\_NUMBER VARCHAR(20),

EVENT\_ID INT,

FOREIGN KEY (EVENT\_ID) REFERENCES EVENT(EVENT\_ID)

);

ALTER TABLE ATTENDEE MODIFY ATTENDEE\_ID INT AUTO\_INCREMENT;

ALTER TABLE ATTENDEE AUTO\_INCREMENT = 101;

INSERT INTO ATTENDEE (ATTENDEE\_ID, NAME, NUMBER, EVENT\_ID) VALUES

(101, 'Alice Johnson', '1234567890', 1),

(102, 'Bob Smith', '9876543210', 2),

(103, 'Charlie Lee', '5551234567', 3),

(104, 'Diana Evans', '4445678901', 4),

(105, 'Ethan Clark', '3214567890', 5),

(106, 'Fiona Davis', '6789012345', 6),

(107, 'George Brown', '8901234567', 7),

(108, 'Hannah Scott', '4567890123', 8),

(109, 'Ian Miller', '2345678901', 9),

(110, 'Julia Adams', '7654321098', 10),

(111, 'Kevin Wilson', '1230984567', 11),

(112, 'Laura Green', '5678901234', 12),

(113, 'Mike Turner', '6789123456', 13),

(114, 'Nina Carter', '7890123456', 14),

(115, 'Owen Hall', '8901345678', 15),

(116, 'Paula White', '9012345678', 16),

(117, 'Quinn Brooks', '3456789012', 17),

(118, 'Rachel Young', '4561237890', 18),

(119, 'Sam Harris', '5678912345', 19),

(120, 'Tina Lopez', '6789012456', 20);

INSERT INTO ATTENDEE (NAME, PH\_NUMBER, EVENT\_ID) VALUES

('Alice Johnson', '1234567890', 1),

('Bob Smith', '9876543210', 2),

('Charlie Lee', '5551234567', 1),

('Diana Evans', '4445678901', 3),

('Ethan Clark', '3214567890', 4),

('Fiona Davis', '6789012345', 5),

('George Brown', '8901234567', 1),

('Hannah Scott', '4567890123', 2),

('Ian Miller', '2345678901', 3),

('Julia Adams', '7654321098', 4),

('Kevin Wilson', '1230984567', 6),

('Laura Green', '5678901234', 7),

('Mike Turner', '6789123456', 8),

('Nina Carter', '7890123456', 9),

('Owen Hall', '8901345678', 10),

('Paula White', '9012345678', 15),

('Quinn Brooks', '3456789012', 11),

('Rachel Young', '4561237890', 12),

('Sam Harris', '5678912345', 13),

('Tina Lopez', '6789012456', 14);

🧠 Note:

The ATTENDEE\_ID will automatically start from 101 because of your AUTO\_INCREMENT = 101 setting.

All phone numbers are strings (VARCHAR) and valid formats for basic insert testing.

You can modify or extend these names and phone numbers as needed.

CREATE TABLE VENUE(  
VENUE\_ID INT PRIMARY KEY,  
VENUE\_NAME VARCHAR(255),  
CAPACITY INT  
);

INSERT INTO VENUE (VENUE\_ID, VENUE\_NAME, CAPACITY) VALUES

(1, 'Grand Hall', 500),

(2, 'City Auditorium', 800),

(3, 'Skyline Conference Center', 300),

(4, 'Downtown Expo Arena', 1000),

(5, 'Lakeside Pavilion', 200),

(6, 'Heritage Banquet Hall', 400),

(7, 'Summit Convention Hall', 750),

(8, 'Crystal Ballroom', 350),

(9, 'Open Air Theater', 1200),

(10, 'Cultural Center Hall', 600);

CREATE TABLE VENDOR(  
VENDOR\_ID INT PRIMARY KEY,  
NAME VARCHAR(255),  
AVAILABILITY VARCHAR(10)  
);

INSERT INTO VENDOR (VENDOR\_ID, NAME, AVAILABILITY) VALUES

(1, 'Elite Catering Co.', 'Yes'),

(2, 'SoundTech Solutions', 'No'),

(3, 'Floral Designs', 'Yes'),

(4, 'Stage Masters', 'Yes'),

(5, 'EventLights Inc.', 'No'),

(6, 'Gourmet Bites', 'Yes'),

(7, 'SecureNow Security', 'Yes'),

(8, 'Sparkle Decorators', 'No'),

(9, 'Live Beats Band', 'Yes'),

(10, 'Royal Rentals', 'Yes');

///// CREATE TABLE TASK(  
TASK\_ID INT PRIMARY KEY,  
NAME VARCHAR(255),  
EVENT\_ID INT FOREIGN KEY REFERENCES EVENT(EVENT\_ID)  
); ///////////////

CREATE TABLE SCHEDULE(

SCHEDULE\_ID INT PRIMARY KEY,

EVENT\_ID INT,

FOREIGN KEY (EVENT\_ID) REFERENCES EVENT(EVENT\_ID),

START\_DATE DATETIME,

END\_TIME DATETIME

);

INSERT INTO SCHEDULE (SCHEDULE\_ID, EVENT\_ID, START\_DATE, END\_TIME) VALUES

(1, 2, '2025-08-10 09:00:00', '2025-08-10 12:00:00'),

(2, 10, '2025-08-11 13:00:00', '2025-08-11 16:00:00'),

(3, 6, '2025-08-12 10:00:00', '2025-08-12 14:00:00'),

(4, 1, '2025-08-13 15:00:00', '2025-08-13 18:00:00'),

(5, 5, '2025-08-14 11:00:00', '2025-08-14 14:00:00'),

(6, 1, '2025-08-15 09:30:00', '2025-08-15 11:30:00'),

(7, 2, '2025-08-16 10:00:00', '2025-08-16 13:00:00'),

(8, 9, '2025-08-17 14:00:00', '2025-08-17 17:00:00'),

(9, 4, '2025-08-18 08:00:00', '2025-08-18 10:00:00'),

(10, 5, '2025-08-19 12:00:00', '2025-08-19 15:00:00'),

(11, 12, '2025-08-20 09:00:00', '2025-08-20 12:00:00'),

(12, 14, '2025-08-21 13:00:00', '2025-08-21 16:00:00'),

(13, 13, '2025-08-22 10:00:00', '2025-08-22 13:00:00'),

(14, 11, '2025-08-23 11:00:00', '2025-08-23 14:00:00'),

(15, 10, '2025-08-24 15:00:00', '2025-08-24 18:00:00');

CREATE TABLE BUDGET(

BUDGET\_ID INT,

EVENT\_ID INT,

FOREIGN KEY (EVENT\_ID) REFERENCES EVENT(EVENT\_ID),

TOTAL\_BUDGET DECIMAL(10,2)

);

INSERT INTO BUDGET (BUDGET\_ID, EVENT\_ID, TOTAL\_BUDGET) VALUES

(1, 1, 5000.00),

(2, 2, 7500.50),

(3, 13, 3200.75),

(4, 14, 10000.00),

(5, 5, 4500.25),

(6, 3, 5200.00),

(7, 4, 7800.00),

(8, 6, 3000.00),

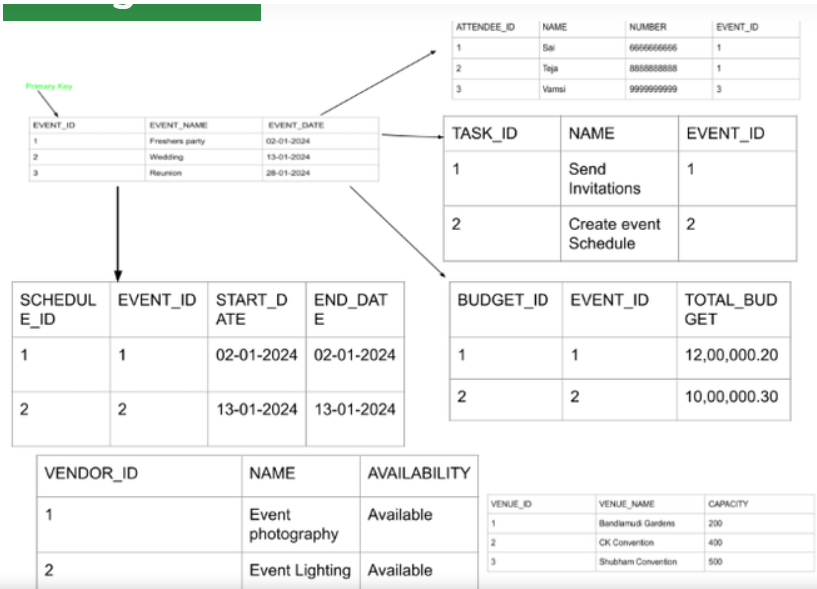
(9, 14, 11000.00),

(10, 7, 4800.00);

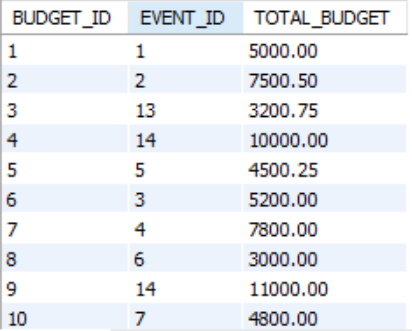
Notes:

Make sure the EVENT table has these EVENT\_IDs to satisfy foreign key constraints.

**Relational Database Model for Event Management**



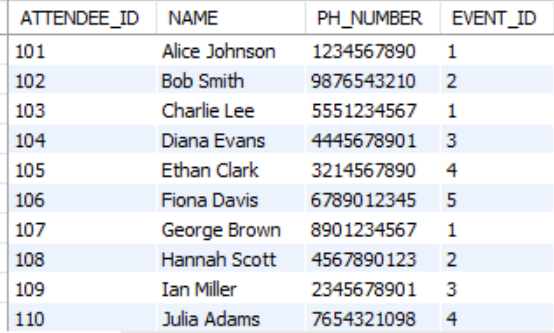
**Budget entity**



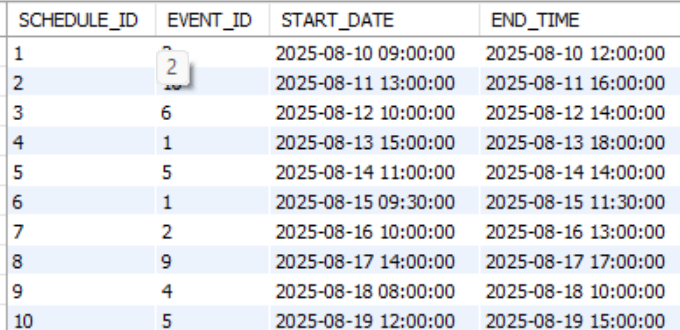
**Event entity**



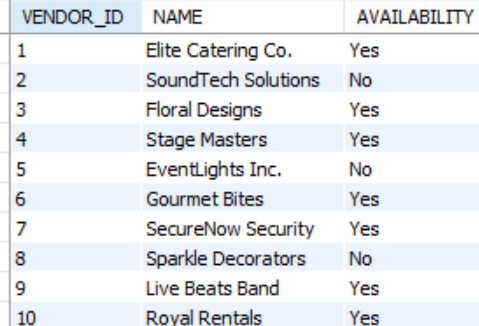
**Attendee entity**



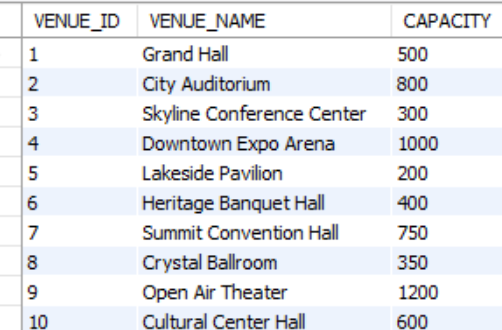
**schedule entity**



**vendor entity**



**Venue entity**



Sql queries

1. **1. Retrieve all upcoming events:**

SELECT \*

FROM Events

WHERE EventDate >= CURDATE()

ORDER BY EventDate;

***OR (TO DISPLAY IN DATE FORMAT)***

SELECT EVENT\_ID, EVENT\_NAME, DATE\_FORMAT(EVENT\_DATE, '%d-%m-%Y') AS EVENT\_DATE, EVENT\_TIME

FROM event

WHERE EVENT\_DATE >= CURDATE()

ORDER BY EVENT\_DATE ;

1. **Find attendees for a specific event:**

SELECT A.AttendeeName, A.AttendeeEmail  
FROM Attendees AS A  
JOIN Events AS E ON A.EventID = E.EventID  
WHERE E.EventName = 'Conference on AI';

Or

SELECT COUNT(A.ATTENDEE\_ID) AS attendee\_count, A.NAME, A.PH\_NUMBER

FROM attendee A

JOIN event E ON E.EVENT\_ID = A.EVENT\_ID

WHERE E.EVENT\_NAME = 'Film Screening'

GROUP BY A.NAME, A.PH\_NUMBER;

OR

SELECT A.ATTENDEE\_ID, A.NAME, A.PH\_NUMBER, E.EVENT\_DATE,E.EVENT\_TIME

FROM attendee A

JOIN event E ON E.EVENT\_ID = A.EVENT\_ID

WHERE E.EVENT\_NAME = 'Film Screening';

1. **Count the number of attendees for each event:**

SELECT E.EVENT\_NAME, COUNT(A.ATTENDEE\_ID) AS NumberOfAttendees

FROM event AS E

LEFT JOIN attendee AS A ON E.EVENT\_ID = A.EVENT\_ID

GROUP BY E.EVENT\_NAME;

**4. List all events held at a particular venue:**

**SELECT E.EVENT\_NAME, E.EVENT\_DATE, E.EVENT\_TIME**

**FROM event AS E**

**JOIN Venue AS V ON E.VENUE\_ID = V.VENUE\_ID**

**WHERE V.VENUE\_NAME = 'city auditorium';**

**5.Add a new event:**

**INSERT INTO Events (EventName, EventDate, VenueID, Description)  
VALUES ('Tech Expo 2025', '2025-11-15', 101, 'Showcasing the latest**

**6. Update an event's details:**

**UPDATE Events  
SET EventDate = '2025-12-01', Description = 'An updated description for the event.'  
WHERE EventName = 'Annual Gala Dinner';**

**7. Delete an event:**

**DELETE FROM Events  
WHERE EventName = 'Charity Run';**

**\---------------------------------**

**1. List all events**

**sql**

**CopyEdit**

**SELECT \* FROM EVENT;**

**👥 2. Show all attendees of a specific event (e.g., EVENT\_ID = 101)**

**SELECT \* FROM ATTENDEE WHERE EVENT\_ID = 101;**

**🧑‍💼 3. List all vendors who are available**

**SELECT \* FROM VENDOR WHERE AVAILABILITY = 'Yes';**

**📅 4. Show the schedule for a specific event**

**SELECT \* FROM SCHEDULE WHERE EVENT\_ID = 103;**

**💰 5. Find the total budget for each event**

**SELECT EVENT\_ID, SUM(TOTAL\_BUDGET) AS TOTAL FROM BUDGET GROUP BY EVENT\_ID ORDER BY TOTAL DESC;**

**🏟️ 6. Show all venues with capacity greater than 500**

**SELECT \* FROM VENUE WHERE CAPACITY > 500;**

**📈 7. List events with their schedule**

**SELECT E.EVENT\_ID, E.EVENT\_NAME, S.START\_DATE, S.END\_TIME**

**FROM EVENT E**

**JOIN SCHEDULE S ON E.EVENT\_ID = S.EVENT\_ID;**

**👥 8. Count number of attendees per event**

**SELECT EVENT\_ID, COUNT(\*) AS ATTENDEE\_COUNT**

**FROM ATTENDEE**

**GROUP BY EVENT\_ID;**

**💸 9. List events where budget exceeds 7000**

**SELECT \* FROM BUDGET WHERE TOTAL\_BUDGET > 7000;**

**🗂️ 10. List unique event names with budgets**

**SELECT DISTINCT E.EVENT\_NAME, B.TOTAL\_BUDGET**

**FROM EVENT E**

**JOIN BUDGET B ON E.EVENT\_ID = B.EVENT\_ID;**

**10) Goal: Get event details with venue name, schedule time, and total budget (via subquery)**

SELECT

e.event\_id,

e.EVENT\_name,

v.venue\_name,

s.start\_DATE,

s.end\_time,

b.total\_budget

FROM event e

LEFT JOIN venue v ON e.venue\_id = v.venue\_id

LEFT JOIN schedule s ON e.event\_id = s.event\_id

LEFT JOIN (

SELECT event\_id, SUM(TOTAL\_BUDGET) AS total\_budget

FROM budget

GROUP BY event\_id

) b ON e.event\_id = b.event\_id;