**Task 3**

**Project: Event Management System using PostgreSQL.**

Objective: To develop the application that allows users to create and manage events, track attendees, and handle event registrations efficiently. The project will include the following tasks:

**1. Database Creation**

Create a database named "EventsManagement." Create tables for Events, Attendees, and Registrations.

Events- Event\_Id, Event\_Name, Event\_Date, Event\_Location, Event\_Description

Attendees- Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City

Registrations-Registration\_id, Event\_Id, Attendee\_Id,Registration\_Date,Registration\_Amount.

The FOREIGN KEY constraint in the Registrations table references the Event\_Id column in the Events table and the Attendee\_Id column in the Attendees table.

CREATE DATABASE EventsManagement;

CREATE TABLE Events (

Event\_Id SERIAL PRIMARY KEY,

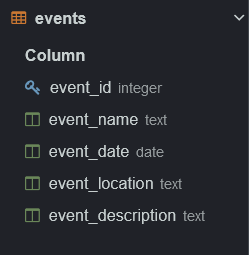
Event\_Name TEXT NOT NULL,

Event\_Date DATE NOT NULL,

Event\_Location TEXT NOT NULL,

Event\_Description TEXT

);



CREATE TABLE Attendees (

Attendee\_Id SERIAL PRIMARY KEY,

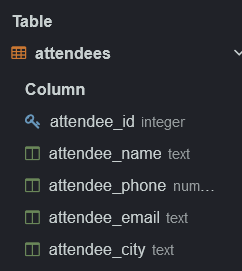
Attendee\_Name TEXT NOT NULL,

Attendee\_Phone NUMERIC NOT NULL,

Attendee\_Email TEXT NOT NULL,

Attendee\_City TEXT NOT NULL

);



CREATE TABLE Registrations (

Registration\_Id SERIAL PRIMARY KEY,

Event\_Id INT NOT NULL,

Attendee\_Id INT NOT NULL,

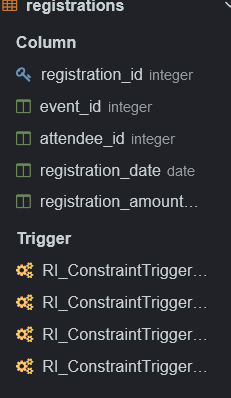
Registration\_Date DATE NOT NULL,

Registration\_Amount NUMERIC NOT NULL,

FOREIGN KEY (Event\_Id) REFERENCES Events(Event\_Id),

FOREIGN KEY (Attendee\_Id) REFERENCES Attendees(Attendee\_Id)

);



**2. Data Creation**

Insert some sample data for Events, Attendees, and Registrations tables with respective fields.

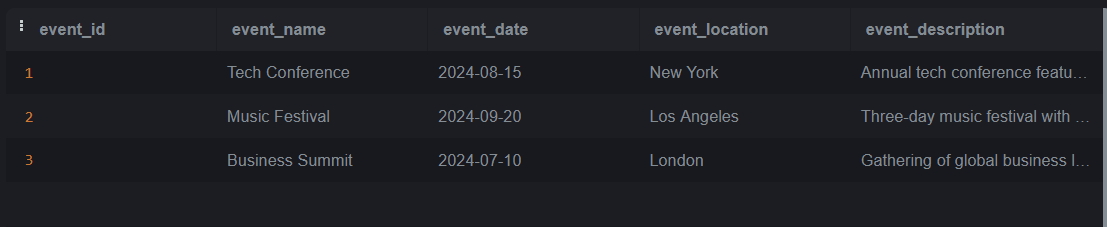
INSERT INTO Events (Event\_Id, Event\_Name, Event\_Date, Event\_Location, Event\_Description)

VALUES

(1, 'Tech Conference', '2024-08-15', 'New York', 'Annual tech conference featuring latest trends.'),

(2, 'Music Festival', '2024-09-20', 'Los Angeles', 'Three-day music festival with top artists.'),

(3, 'Business Summit', '2024-07-10', 'London', 'Gathering of global business leaders.');



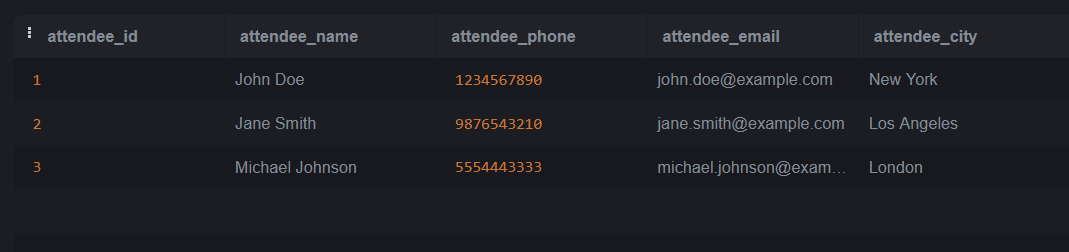
INSERT INTO Attendees (Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City)

VALUES

(1, 'John Doe', '1234567890', 'john.doe@example.com', 'New York'),

(2, 'Jane Smith', '9876543210', 'jane.smith@example.com', 'Los Angeles'),

(3, 'Michael Johnson', '5554443333', 'michael.johnson@example.com', 'London');



-- Inserting sample data into Registrations table

INSERT INTO Registrations (Event\_Id, Attendee\_Id, Registration\_Date, Registration\_Amount)

VALUES

(1, 1, '2024-08-01', 100.00),

(2, 2, '2024-09-01', 150.00),

(3, 3, '2024-07-01', 200.00);



**3. Manage Event Details**

a) Inserting a new event.

-- Inserting a new event without specifying Event\_Id (assuming Event\_Id is SERIAL)

INSERT INTO Events (Event\_Name, Event\_Date, Event\_Location, Event\_Description)

VALUES ('Fashion Show', '2024-10-15', 'Paris', 'Annual fashion event showcasing latest trends.');

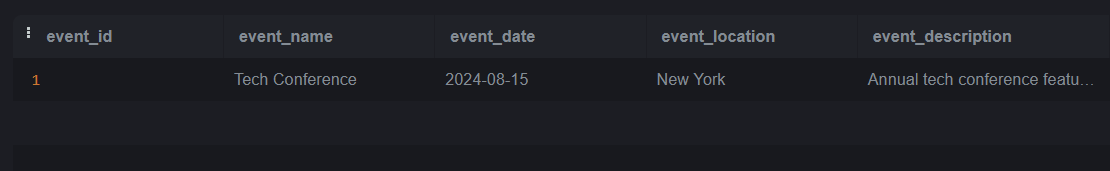
b) Updating an event's information.

UPDATE Events

SET Event\_Location = 'New York',

Event\_Description = 'Annual tech conference featuring latest technologies and speakers.'

WHERE Event\_Id = 1;



c) Deleting an event.

DELETE FROM Events

WHERE Event\_Id = 1;

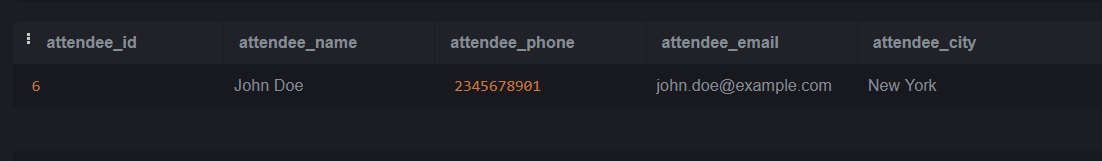
**4) Manage Track Attendees & Handle Events**

a) Inserting a new attendee

-- Inserting a new attendee with Attendee\_Id specified

INSERT INTO Attendees (Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City)

VALUES (6, 'John Doe', '+1-234-567-8901', 'john.doe@example.com', 'New York');

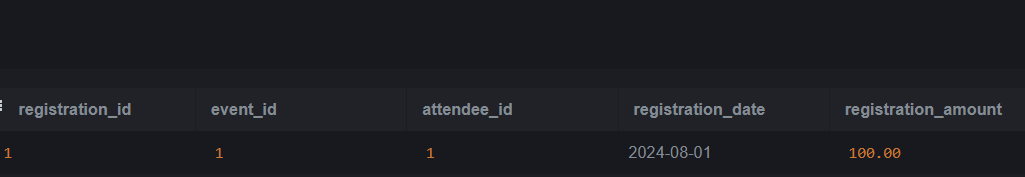


b) Registering an attendee for an event.

INSERT INTO Registrations (Event\_Id, Attendee\_Id, Registration\_Date, Registration\_Amount)

VALUES (1, 1, '2024-06-01', 100.00);

**Output:**



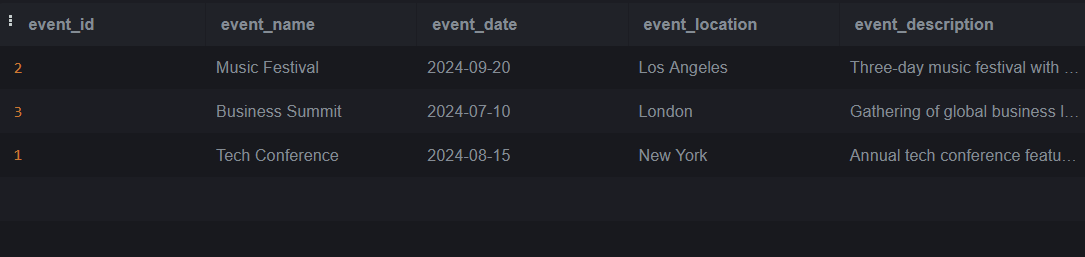
5.Develop queries to retrieve event information, generate attendee lists, and calculate event attendance statistics.

**Retrieve Event Information**

To retrieve all information about events from the Events table:

SELECT \* FROM Events;

Output:



**Generate Attendee Lists**

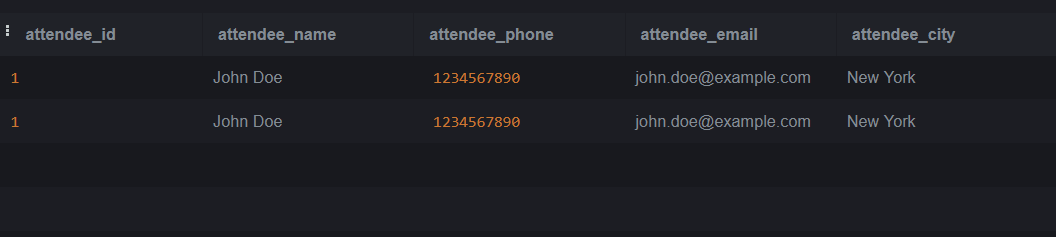
SELECT a.Attendee\_Id, a.Attendee\_Name, a.Attendee\_Phone, a.Attendee\_Email, a.Attendee\_City

FROM Attendees a

JOIN Registrations r ON a.Attendee\_Id = r.Attendee\_Id

WHERE r.Event\_Id = 1;

Output:



**Calculate Event Attendance Statistics**

SELECT e.Event\_Name, COUNT(r.Registration\_Id) AS Attendee\_Count

FROM Events e

LEFT JOIN Registrations r ON e.Event\_Id = r.Event\_Id

GROUP BY e.Event\_Name;

