# Capstone Task 3

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

#### **Database Creation**

**Create a database named "EventsManagement."**

CREATE DATABASE EventsManagement;

**Create tables for Events, Attendees, and Registrations.**

CREATE TABLE Events ( Event\_Id INT PRIMARY KEY, Event\_Name VARCHAR(255) NOT NULL, Event\_Date DATE NOT NULL, Event\_Location VARCHAR(255) NOT NULL, Event\_Description TEXT );

CREATE TABLE Attendees (Attendee\_Id INT PRIMARY KEY, Attendee\_Name VARCHAR(255) NOT NULL, Attendee\_Phone VARCHAR(15), Attendee\_Email VARCHAR(255), Attendee\_City VARCHAR(255) );

CREATE TABLE Registrations ( Registration\_Id INT PRIMARY KEY, Event\_Id INT, Attendee\_Id INT, Registration\_Date DATE NOT NULL, Registration\_Amount DECIMAL(10, 2), FOREIGN KEY (Event\_Id) REFERENCES Events(Event\_Id), FOREIGN KEY (Attendee\_Id) REFERENCES Attendees(Attendee\_Id) );

#### **2) Data Creation**

- Insert sample data into Events table

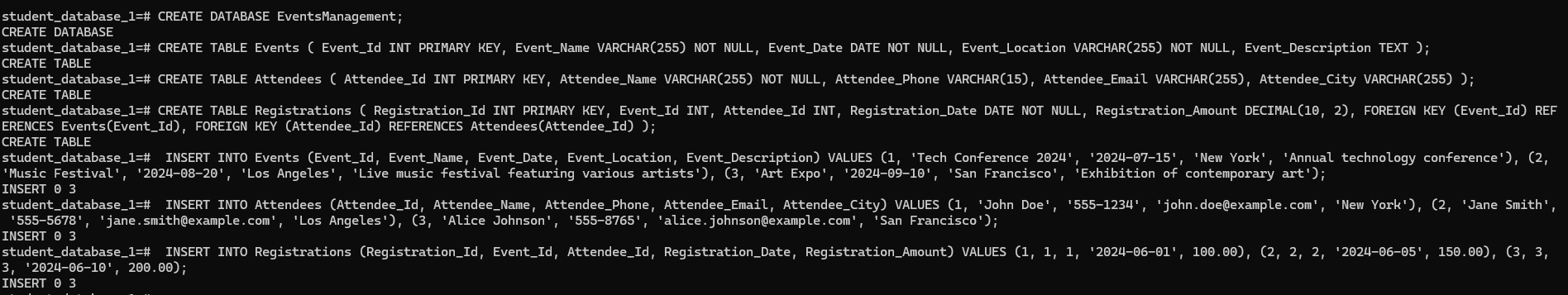
INSERT INTO Events (Event\_Id, Event\_Name, Event\_Date, Event\_Location, Event\_Description) VALUES (1, 'Tech Conference 2024', '2024-07-15', 'New York', 'Annual technology conference'), (2, 'Music Festival', '2024-08-20', 'Los Angeles', 'Live music festival featuring various artists'), (3, 'Art Expo', '2024-09-10', 'San Francisco', 'Exhibition of contemporary art');

-- Insert sample data into Attendees table

INSERT INTO Attendees (Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City) VALUES (1, 'John Doe', '555-1234', 'john.doe@example.com', 'New York'), (2, 'Jane Smith', '555-5678', 'jane.smith@example.com', 'Los Angeles'), (3, 'Alice Johnson', '555-8765', 'alice.johnson@example.com', 'San Francisco');

-- Insert sample data into Registrations table

INSERT INTO Registrations (Registration\_Id, Event\_Id, Attendee\_Id, Registration\_Date, Registration\_Amount) VALUES (1, 1, 1, '2024-06-01', 100.00), (2, 2, 2, '2024-06-05', 150.00), (3, 3, 3, '2024-06-10', 200.00);



#### **3) Manage Event Details**

1. Inserting a new event.

INSERT INTO Events (Event\_Id, Event\_Name, Event\_Date, Event\_Location, Event\_Description) VALUES (4, 'Health & Wellness Fair', '2024-10-05', 'Chicago', 'A fair promoting health and wellness');

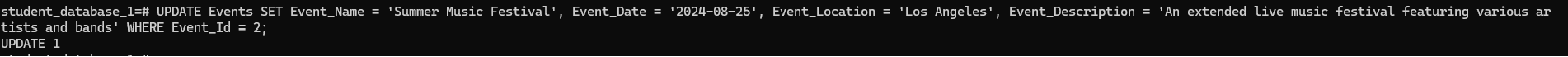


b) Updating an event's information.

-- Use the database USE EventsManagement;

-- Update the event information in the Events table

UPDATE Events SET Event\_Name = 'Summer Music Festival', Event\_Date = '2024-08-25', Event\_Location = 'Los Angeles', Event\_Description = 'An extended live music festival featuring various artists and bands' WHERE Event\_Id = 2;



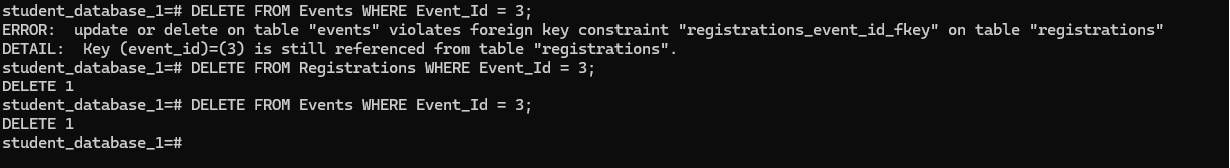
c) Deleting an event.

-- Delete related records in the Registrations table

DELETE FROM Registrations WHERE Event\_Id = 3;

-- Delete the event from the Events table

DELETE FROM Events WHERE Event\_Id = 3;



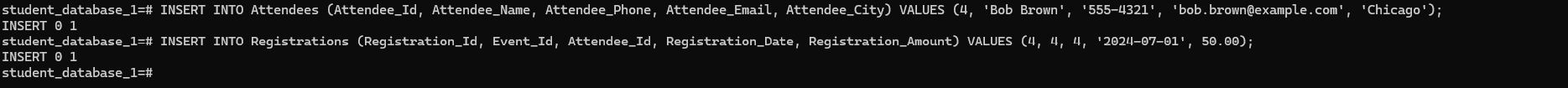
#### **4) Manage Track Attendees & Handle Events**

a) Inserting a new attendee.

INSERT INTO Attendees (Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City) VALUES (4, 'Bob Brown', '555-4321', 'bob.brown@example.com', 'Chicago');

b) Register the attendee for the event

INSERT INTO Registrations (Registration\_Id, Event\_Id, Attendee\_Id, Registration\_Date, Registration\_Amount) VALUES (4, 4, 4, '2024-07-01', 50.00);

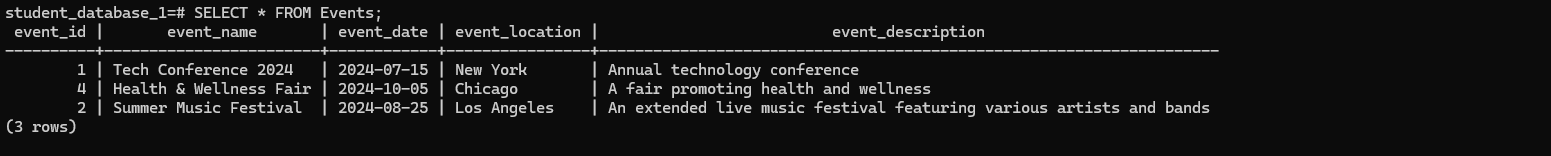


#### **5) Develop queries to retrieve event information, generate attendee lists, and calculate event attendance statistics.**

1. Retrieve Event Information

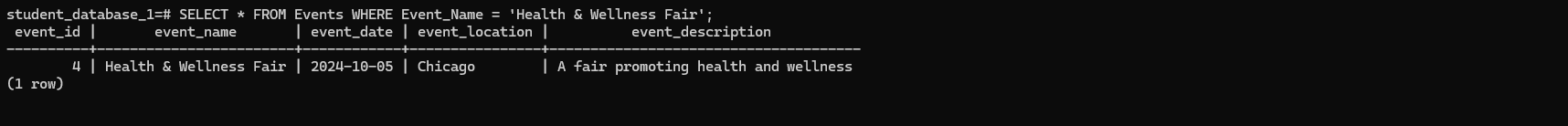
To retrieve all information about events

SELECT \* FROM Events;



To retrieve specific event information by event name:

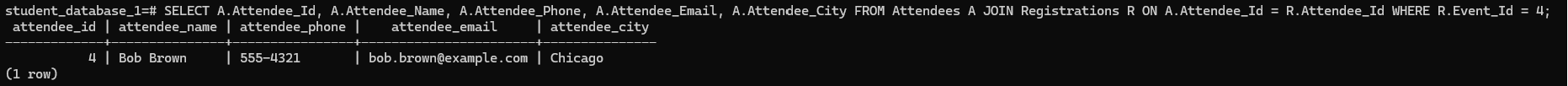
SELECT \* FROM Events WHERE Event\_Name = 'Health & Wellness Fair';



2. Generate Attendee Lists

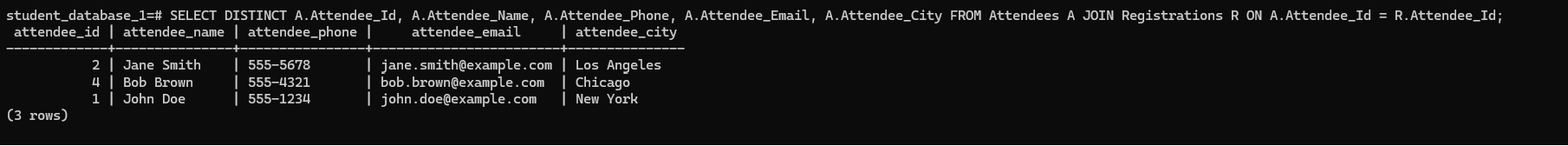
To generate a list of attendees for a specific event:

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 = 4;



To generate a list of all attendees across all events:

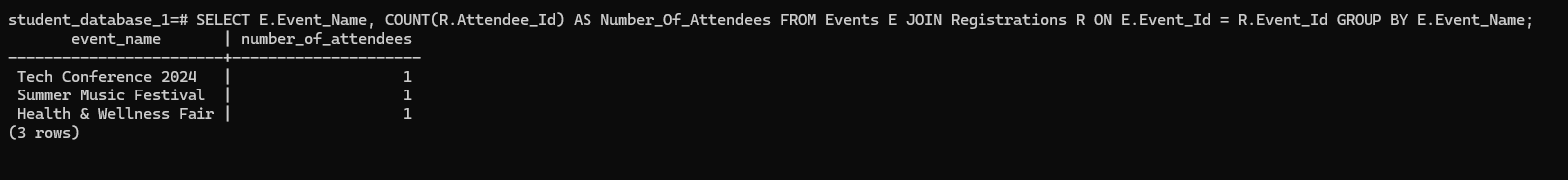
SELECT DISTINCT A.Attendee\_Id, A.Attendee\_Name, A.Attendee\_Phone, A.Attendee\_Email, A.Attendee\_City FROM Attendees A JOIN Registrations R ON A.Attendee\_Id = R.Attendee\_Id;



3. Calculate Event Attendance Statistics

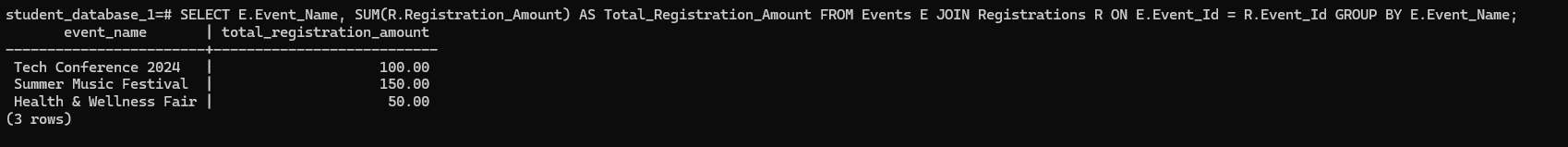
To calculate the number of attendees for each event

SELECT E.Event\_Name, COUNT(R.Attendee\_Id) AS Number\_Of\_Attendees FROM Events E JOIN Registrations R ON E.Event\_Id = R.Event\_Id GROUP BY E.Event\_Name;



To calculate the total registration amount collected for each event

SELECT E.Event\_Name, SUM(R.Registration\_Amount) AS Total\_Registration\_Amount FROM Events E JOIN Registrations R ON E.Event\_Id = R.Event\_Id GROUP BY E.Event\_Name;



To calculate the average registration amount for each event:

SELECT E.Event\_Name, AVG(R.Registration\_Amount) AS Average\_Registration\_Amount FROM Events E JOIN Registrations R ON E.Event\_Id = R.Event\_Id GROUP BY E.Event\_Name;

