**Task3**

**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 DATABASE EventsManagement;

Create tables for Events, Attendees, and Registrations.

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

CREATE TABLE Events (

Event\_Id SERIAL PRIMARY KEY,

Event\_Name VARCHAR(255) NOT NULL,

Event\_Date DATE NOT NULL,

Event\_Location VARCHAR(255),

Event\_Description TEXT

);

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

CREATE TABLE Attendees (

Attendee\_Id SERIAL PRIMARY KEY,

Attendee\_Name VARCHAR(255) NOT NULL,

Attendee\_Phone VARCHAR(15),

Attendee\_Email VARCHAR(255) UNIQUE NOT NULL,

Attendee\_City VARCHAR(255)

);

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

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 DECIMAL(10, 2),

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

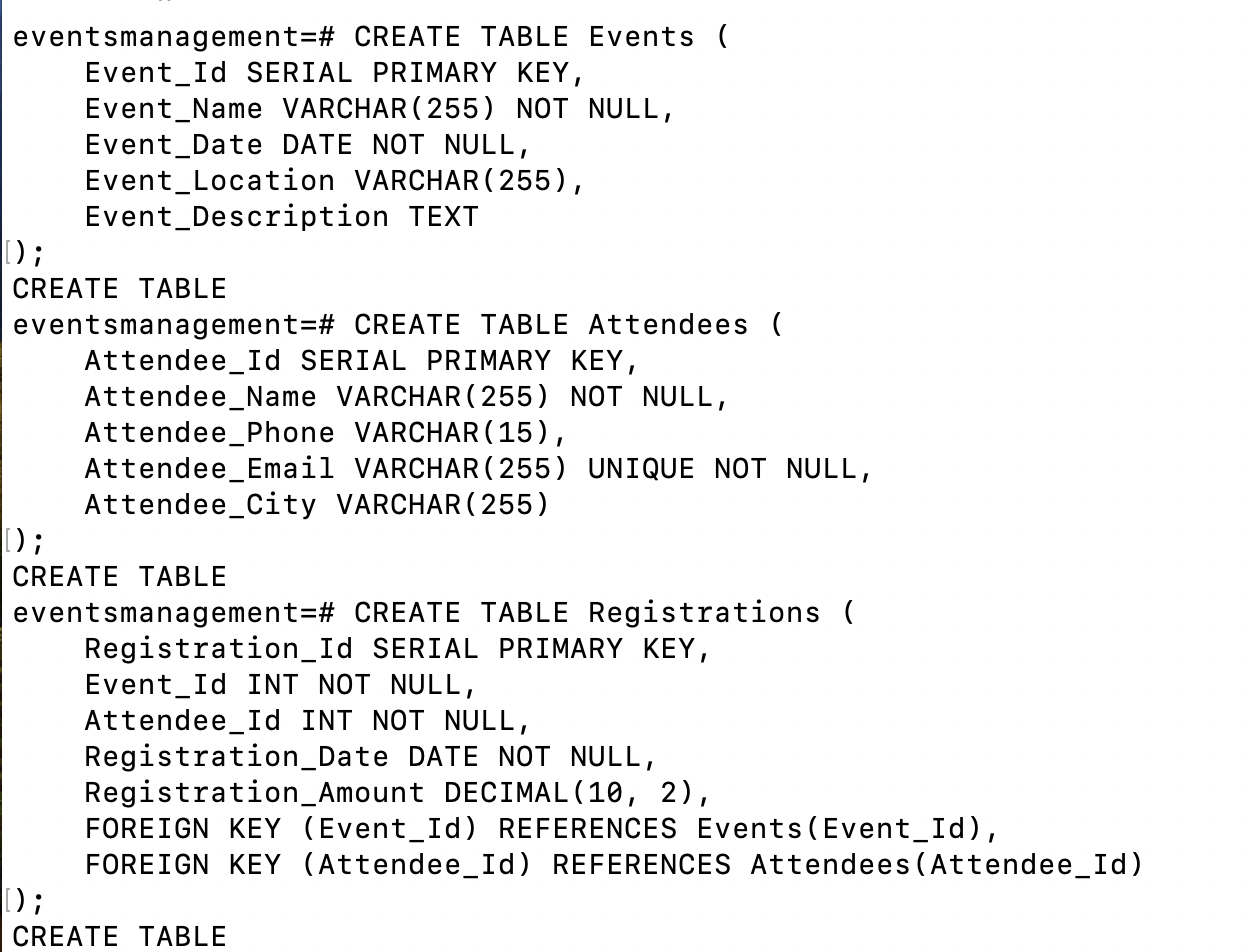
FOREIGN KEY (Attendee\_Id) REFERENCES Attendees(Attendee\_Id)

);

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.





2.Data Creation

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

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

('Tech Conference 2024', '2024-06-15', 'San Francisco', 'A conference about the latest in technology.'),

('Music Festival', '2024-07-20', 'Los Angeles', 'A festival featuring various music genres.'),

('Art Exhibition', '2024-08-10', 'New York', 'An exhibition showcasing contemporary art.');

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

('Alice Johnson', '555-1234', 'alice.johnson@example.com', 'San Francisco'),

('Bob Smith', '555-5678', 'bob.smith@example.com', 'Los Angeles'),

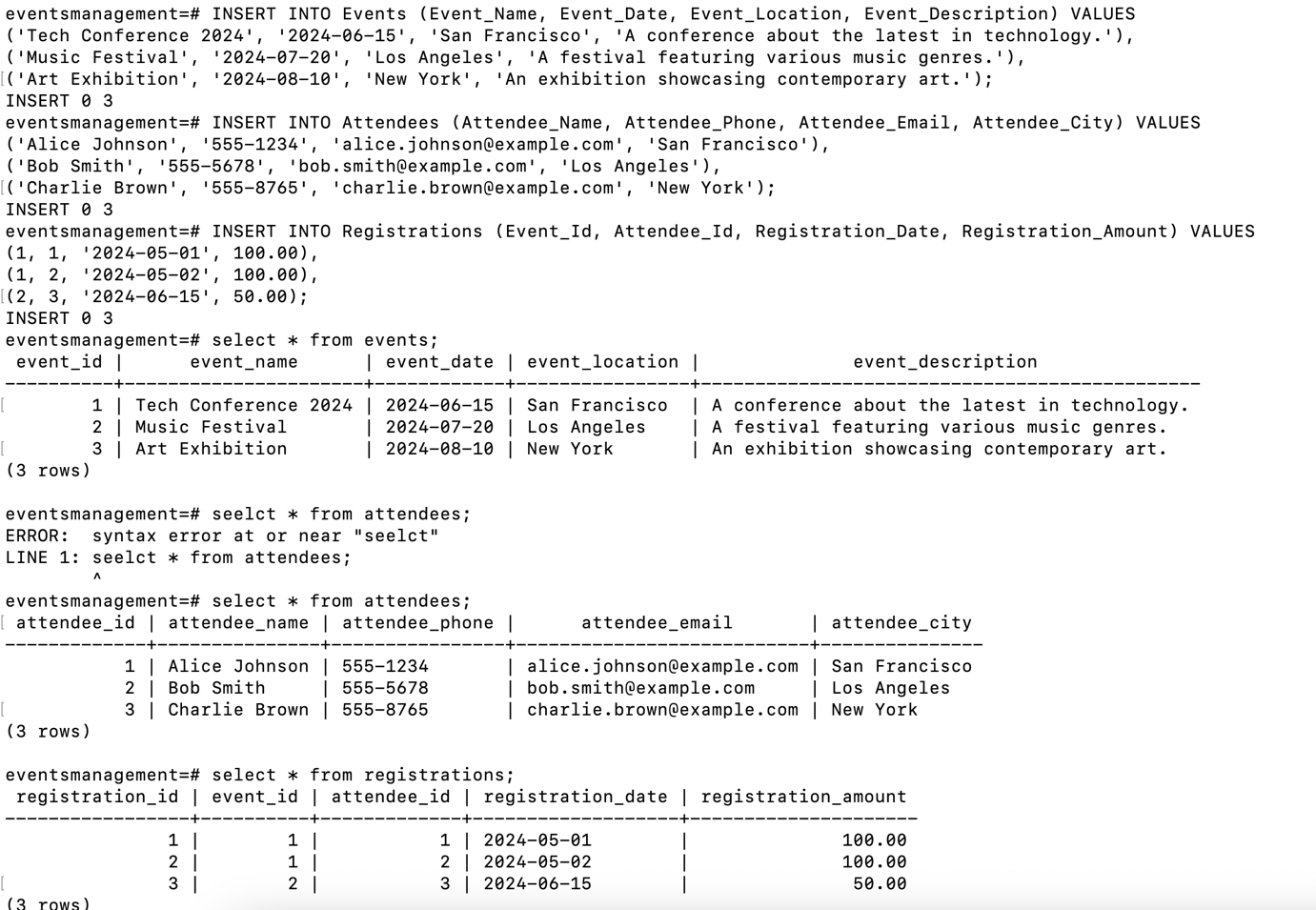
('Charlie Brown', '555-8765', 'charlie.brown@example.com', 'New York');

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

(1, 1, '2024-05-01', 100.00),

(1, 2, '2024-05-02', 100.00),

(2, 3, '2024-06-15', 50.00);



3. Manage Event Details

a) Inserting a new event.

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

('Networking Event', '2024-09-01', 'San Jose', 'An event to network with industry professionals.');

b) Updating an event's information.

UPDATE Events

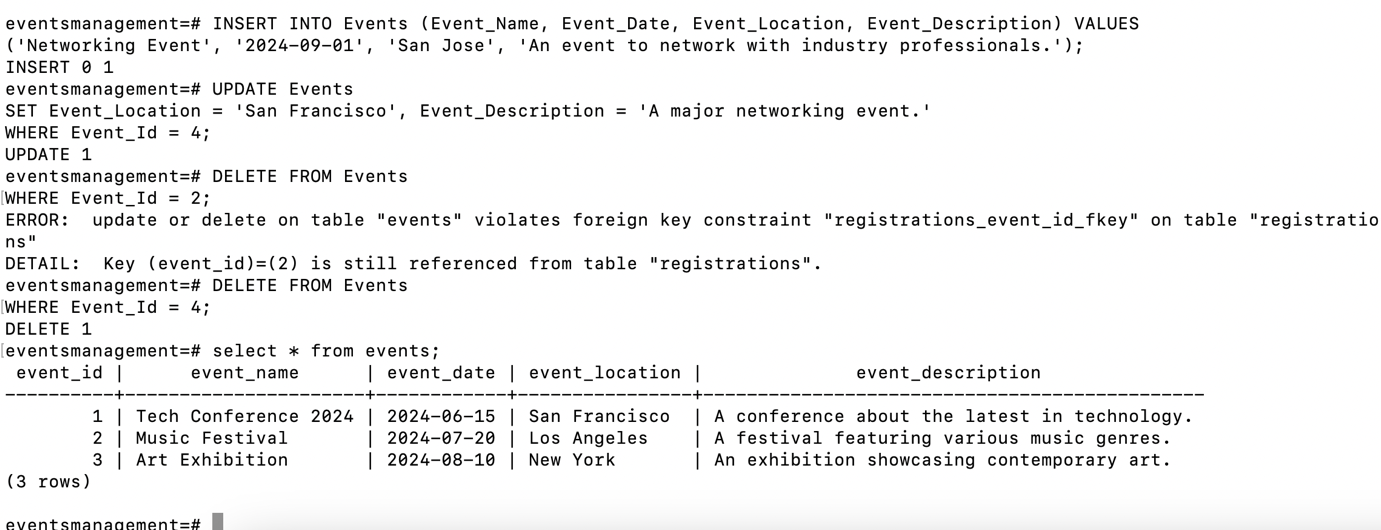
SET Event\_Location = 'San Francisco', Event\_Description = 'A major networking event.'

WHERE Event\_Id = 4;

c) Deleting an event.

DELETE FROM Events

WHERE Event\_Id = 4;



4) Manage Track Attendees & Handle Events

a)Inserting a new attendee.

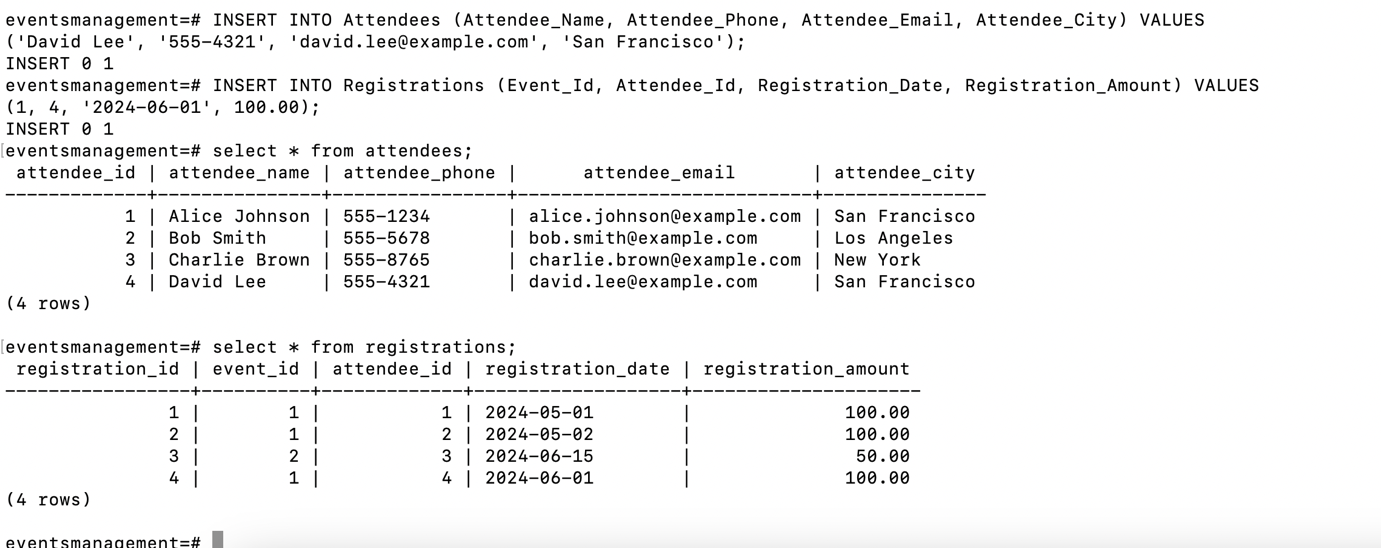
INSERT INTO Attendees (Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City) VALUES

('David Lee', '555-4321', 'david.lee@example.com', 'San Francisco');

b)Registering an attendee for an event.

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

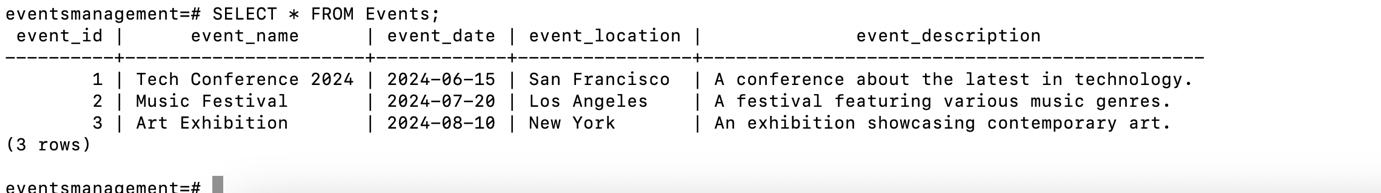
(1, 4, '2024-06-01', 100.00);



5.Develop queries to

1. retrieve event information

SELECT \* FROM Events;



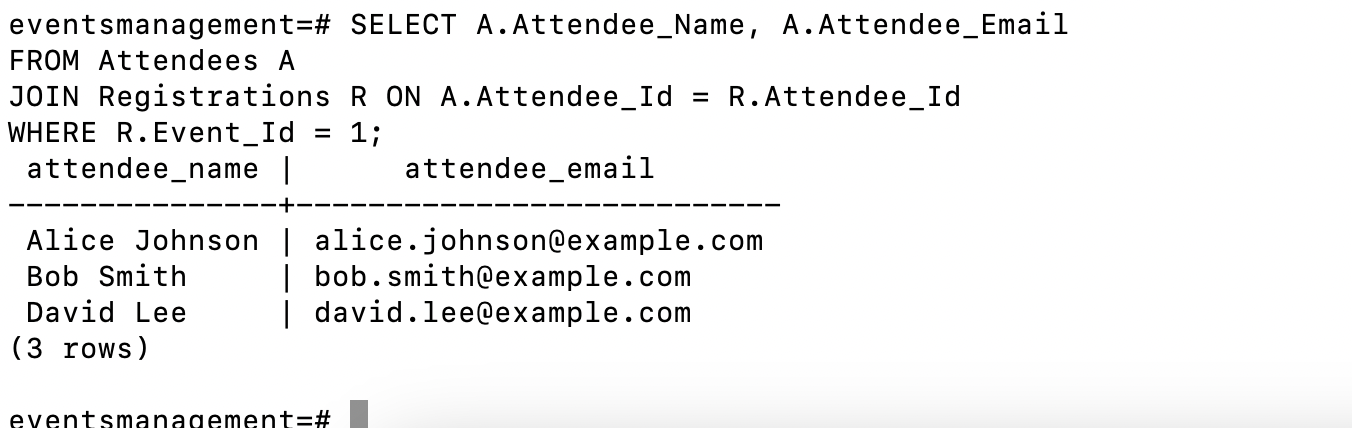
1. generate attendee lists

SELECT A.Attendee\_Name, A.Attendee\_Email

FROM Attendees A

JOIN Registrations R ON A.Attendee\_Id = R.Attendee\_Id

WHERE R.Event\_Id = 1;



1. calculate event attendance statistics.

SELECT E.Event\_Name, COUNT(R.Attendee\_Id) AS NumberOfAttendees

FROM Events E

LEFT JOIN Registrations R ON E.Event\_Id = R.Event\_Id

GROUP BY E.Event\_Name;

