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.A Database Creation**

CREATE DATABASE EventsManagement;

**B) Table Creation**

**Events**

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);

**Attendees**

CREATE TABLE Attendees (Attendee\_Id SERIAL PRIMARY KEY,  
Attendee\_Name TEXT NOT NULL,  
Attendee\_Phone TEXT NOT NULL,  
Attendee\_Email TEXT NOT NULL,  
Attendee\_City TEXT NOT NULL);

**Registration**

CREATE TABLE Registrations (Registration\_Id SERIAL PRIMARY KEY,  
Event\_Id INT REFERENCES Events(Event\_Id),  
Attendee\_Id INT REFERENCES Attendees(Attendee\_Id),  
Registration\_Date DATE NOT NULL,  
Registration\_Amount NUMERIC);

**2) Data Creation**

-- Insert sample rows into Events table

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

('Women in Tech Conference', '2024-07-10', 'San Francisco', 'A conference celebrating women in technology.'),

('Women Health Expo', '2024-08-15', 'New York', 'An expo focusing on women’s health and wellness.'),

('Women Entrepreneurs Summit', '2024-09-20', 'Chicago', 'A summit for women entrepreneurs to network and learn.'),

('Women in Arts Exhibition', '2024-10-05', 'Los Angeles', 'An exhibition showcasing art by women artists.'),

('Women in Sports Gala', '2024-11-12', 'Boston', 'A gala celebrating achievements of women in sports.'),

('Women Leadership Workshop', '2024-12-01', 'Seattle', 'A workshop on leadership skills for women.');

-- Insert sample rows into Attendees table

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

('Anita Roy', '1234567890', 'anita.roy@example.com', 'San Francisco'),

('Meena Patel', '0987654321', 'meena.patel@example.com', 'New York'),

('Priya Singh', '1122334455', 'priya.singh@example.com', 'Chicago'),

('Lakshmi Nair', '5566778899', 'lakshmi.nair@example.com', 'Los Angeles'),

('Sunita Das', '6677889900', 'sunita.das@example.com', 'Boston'),

('Radha Iyer', '7788990011', 'radha.iyer@example.com', 'Seattle');

-- Insert sample rows into Registrations table

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

(1, 1, '2024-06-01', 150.00),

(2, 2, '2024-06-05', 100.00),

(3, 3, '2024-06-10', 200.00),

(4, 4, '2024-06-15', 75.00),

(5, 5, '2024-06-20', 120.00),

(6, 6, '2024-06-25', 80.00);

**3) Manage Event Details**

**a) Inserting a new event**

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

VALUES ('Women in Science Symposium', '2024-10-18', 'Houston', 'A symposium highlighting contributions of women in science.');

**b) Updating an event's information.**

UPDATE Events

SET Event\_Location = 'Washington D.C.', Event\_Description = 'An updated expo focusing on women’s health and wellness.'

WHERE Event\_Id = 2;

**c) Deleting an event.**

DELETE FROM Events

WHERE Event\_Id = 3;

**4) Inserting a new attendee**

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

VALUES ('Neha Sharma', '9998887777', 'neha.sharma@example.com', 'Miami');

**b) Registering an attendee for an event.**

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

VALUES (1, 7, '2024-06-29', 150.00);

**5)** **Develop queries to retrieve event information, generate attendee lists, and calculate event**

**attendance statistics**

1. **Retrieve event information**

For entire list of events

SELECT \* FROM Events

For a specific event

SELECT \* FROM Events

WHERE Event\_Id = 1;

**B)** **Generate attendee lists for a specific event**

SELECT Attendees.Attendee\_Name, Attendees.Attendee\_Phone, Attendees.Attendee\_Email, Attendees.Attendee\_City

FROM Registrations

JOIN Attendees ON Registrations.Attendee\_Id = Attendees.Attendee\_Id

WHERE Registrations.Event\_Id = 1;

**For all events**

SELECT Events.Event\_Name, Attendees.Attendee\_Name, Attendees.Attendee\_Phone, Attendees.Attendee\_Email, Attendees.Attendee\_City

FROM Registrations

JOIN Attendees ON Registrations.Attendee\_Id = Attendees.Attendee\_Id

JOIN Events ON Registrations.Event\_Id = Events.Event\_Id

ORDER BY Events.Event\_Name;

**C) Calculate the Total Number of Attendees for Each Event**

SELECT e.Event\_Name, COUNT(r.Attendee\_Id) AS Total\_Attendees

FROM Events e

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

GROUP BY e.Event\_Name;

**D) Calculate the Total Registration Amount Collected for Each Event**

SELECT e.Event\_Name, SUM(r.Registration\_Amount) AS Total\_Registration\_Amount

FROM Events e

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

GROUP BY e.Event\_Name;

**E) Calculate the Maximum and Minimum Registration Amount Collected for Each Event**

SELECT e.Event\_Name,

MAX(r.Registration\_Amount) AS Max\_Registration\_Amount,

MIN(r.Registration\_Amount) AS Min\_Registration\_Amount

FROM Events e

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

GROUP BY e.Event\_Name;

**F) List Events and Their Corresponding Attendee Count Grouped by Event Location**

SELECT e.Event\_Location, e.Event\_Name, COUNT(r.Attendee\_Id) AS Total\_Attendees

FROM Events e

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

GROUP BY e.Event\_Location, e.Event\_Name;