-- Task 3: Event Management System using PostgreSQL

-- Step 1: Database Creation

-- Create a database named "EventsManagement"

-- Step 2: Create tables for Events, Attendees, and Registrations

-- Create the "Events" table with columns

CREATE TABLE Events (

Event\_Id SERIAL PRIMARY KEY,

Event\_Name VARCHAR(100),

Event\_Date DATE,

Event\_Location VARCHAR(100),

Event\_Description TEXT

);

-- Create the "Attendees" table with columns

CREATE TABLE Attendees (

Attendee\_id SERIAL PRIMARY KEY,

Attendee\_Name VARCHAR(100),

Attendee\_Phone NUMERIC,

Attendee\_Email TEXT,

Attendee\_City VARCHAR(50)

);

-- Create the "Registrations" table with columns

CREATE TABLE Registrations (

Registration\_id SERIAL PRIMARY KEY,

Event\_Id INT,

Attendee\_id INT,

Registration\_Date DATE,

Registration\_Amount NUMERIC

);

-- Add foreign key constraints

ALTER TABLE Registrations

ADD CONSTRAINT fk\_event

FOREIGN KEY (Event\_Id)

REFERENCES Events(Event\_Id);

ALTER TABLE Registrations

ADD CONSTRAINT fk\_attendee

FOREIGN KEY (Attendee\_id)

REFERENCES Attendees(Attendee\_id);

-- Step 3: Data Creation

-- Insert sample data for the "Events" table

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

VALUES

('Seminar on AI', '2023-03-15', 'Mumbai', 'Introduction to Artificial Intelligence'),

('Music Concert', '2023-04-20', 'Delhi', 'Live music performance'),

('Tech Conference', '2023-05-10', 'Bangalore', 'Latest tech trends and innovations');

-- Insert sample data for the "Attendees" table

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

VALUES

('Rajesh Kumar', 9876543210, 'rajesh@email.com', 'Delhi'),

('Asha Sharma', 8765432109, 'asha@email.com', 'Mumbai'),

('Kiran Singh', 7654321098, 'kiran@email.com', 'Bangalore');

-- Insert sample data for the "Registrations" table

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

VALUES

(1, 1, '2023-03-10', 500),

(1, 2, '2023-03-11', 500),

(2, 2, '2023-04-15', 300),

(3, 3, '2023-05-05', 600);

-- Step 4: Manage Event Details

-- a) Inserting a new event

-- Insert a new event named "Art Exhibition"

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

VALUES

('Art Exhibition', '2023-06-15', 'Chennai', 'Display of modern art creations');

-- b) Updating an event's information

-- Update event description for the "Seminar on AI"

UPDATE Events

SET Event\_Description = 'In-depth AI knowledge sharing'

WHERE Event\_Id = 1;

-- c) Deleting an event

-- Delete the event with Event\_Id = 3 (Tech Conference)

DELETE FROM Events

WHERE Event\_Id = 3;

-- Step 5: Manage Track Attendees & Handle Events

-- a) Inserting a new attendee

-- Insert a new attendee named "Neha Pandey"

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

VALUES

('Neha Pandey', 9876543211, 'neha@email.com', 'Chennai');

-- b) Registering an attendee for an event

-- Register Neha Pandey for the "Art Exhibition" event

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

SELECT

e.Event\_Id,

a.Attendee\_id,

'2023-06-10',

400

FROM

Events e

JOIN

Attendees a ON a.Attendee\_Name = 'Neha Pandey';

-- Step 6: Develop queries to retrieve event information, generate attendee lists, and calculate event attendance statistics

-- a) Retrieve event information

SELECT \* FROM Events;

-- b) Generate attendee lists for a specific event

SELECT

a.Attendee\_Name

FROM

Registrations r

JOIN

Attendees a ON r.Attendee\_id = a.Attendee\_id

WHERE

r.Event\_Id = 1; -- Specify the event ID

-- c) 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;

-- Step 7: Commit the transaction

COMMIT;