**Event Management DB**

To develop the application that allows users to create and manage events, track

attendees, and handle event registrations efficiently.

1. **Database Creation**

-- Create Database

CREATE DATABASE EventsManagement;

-- Create Events Table

CREATE TABLE Events (

Event\_Id INTEGER PRIMARY KEY,

Event\_Name TEXT NOT NULL,

Event\_Date DATE NOT NULL,

Event\_Location TEXT NOT NULL,

Event\_Description TEXT

);

-- Create Attendees Table

CREATE TABLE Attendees (

Attendee\_Id INTEGER PRIMARY KEY,

Attendee\_Name TEXT NOT NULL,

Attendee\_Phone NUMERIC(10),

Attendee\_Email TEXT UNIQUE,

Attendee\_City TEXT

);

-- Create Registrations Table with Foreign Keys

CREATE TABLE Registrations (

Registration\_id INTEGER PRIMARY KEY,

Event\_Id INTEGER REFERENCES Events(Event\_Id),

Attendee\_Id INTEGER REFERENCES Attendees(Attendee\_Id),

Registration\_Date DATE DEFAULT CURRENT\_DATE,

Registration\_Amount NUMERIC(10,2),

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

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

);

-- Create Indexes for better performance

CREATE INDEX idx\_event\_name ON Events(Event\_Name);

CREATE INDEX idx\_event\_date ON Events(Event\_Date);

CREATE INDEX idx\_attendee\_name ON Attendees(Attendee\_Name);

CREATE INDEX idx\_registration\_date ON Registrations(Registration\_Date);

1. **Data Creation**

-- Insert data into Events table

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

(1, 'Tech Summit 2024', '2024-07-15', 'Chennai Trade Centre', 'Annual technology conference with industry experts'),

(2, 'Music Festival', '2024-08-20', 'Marina Beach', 'Open-air music festival featuring local artists'),

(3, 'Food Expo', '2024-09-10', 'Chennai Convention Centre', 'International food and beverage exhibition'),

(4, 'Digital Marketing Workshop', '2024-07-25', 'ITC Grand Chola', 'Hands-on workshop on digital marketing strategies'),

(5, 'Startup Meet', '2024-08-05', 'Anna Centenary Library', 'Networking event for startups and investors');

-- Insert data into Attendees table

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

(101, 'Rajesh Kumar', 9876543210, 'rajesh.k@email.com', 'Chennai'),

(102, 'Priya Mehta', 9876543211, 'priya.m@email.com', 'Bangalore'),

(103, 'Arun Sharma', 9876543212, 'arun.s@email.com', 'Chennai'),

(104, 'Deepa Venkat', 9876543213, 'deepa.v@email.com', 'Coimbatore'),

(105, 'Karthik Raja', 9876543214, 'karthik.r@email.com', 'Chennai'),

(106, 'Meena Kumari', 9876543215, 'meena.k@email.com', 'Madurai'),

(107, 'Suresh Babu', 9876543216, 'suresh.b@email.com', 'Chennai'),

(108, 'Lakshmi Priya', 9876543217, 'lakshmi.p@email.com', 'Salem');

-- Insert data into Registrations table

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

(1001, 1, 101, '2024-06-01', 1500.00),

(1002, 1, 102, '2024-06-02', 1500.00),

(1003, 2, 103, '2024-06-05', 750.00),

(1004, 2, 104, '2024-06-05', 750.00),

(1005, 3, 105, '2024-06-10', 1000.00),

(1006, 3, 106, '2024-06-10', 1000.00),

(1007, 4, 107, '2024-06-15', 2000.00),

(1008, 4, 108, '2024-06-15', 2000.00),

(1009, 5, 101, '2024-06-20', 500.00),

(1010, 5, 103, '2024-06-20', 500.00),

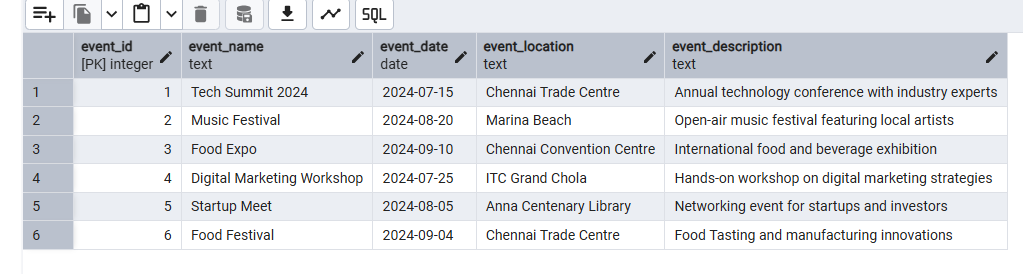
(1011, 1, 105, '2024-06-21', 1500.00),

(1012, 2, 107, '2024-06-22', 750.00);

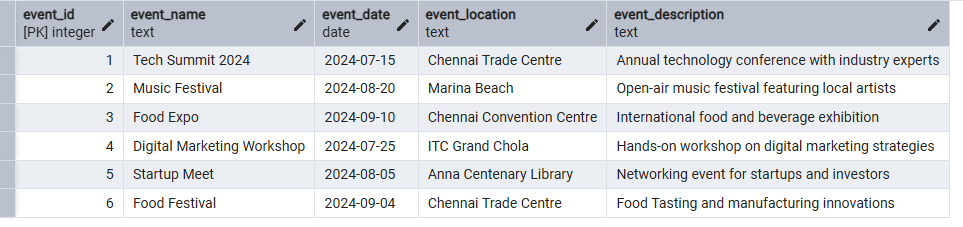
1. **Manage Event Details**
2. Inserting a new event.

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

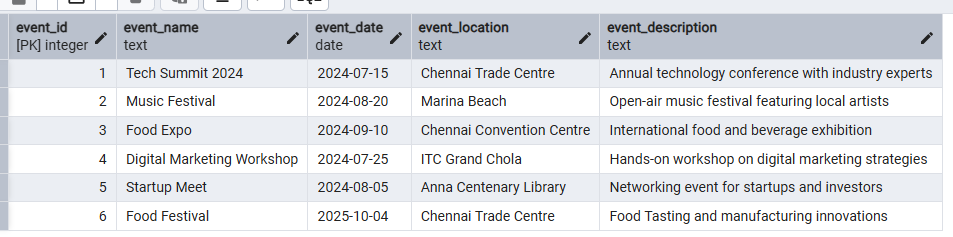
(6, 'Food Festival', '2024-09-04', 'Chennai Trade Centre', 'Food Tasting and manufacturing innovations');



1. Updating an event's information.

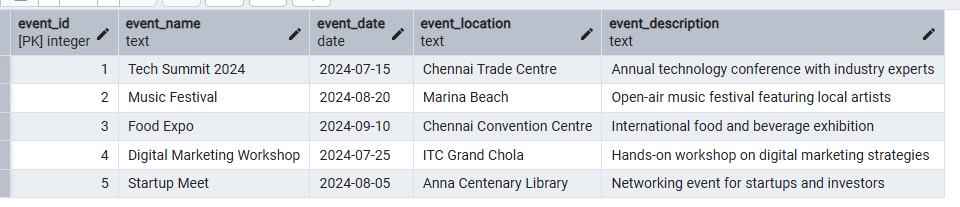


UPDATE events SET event\_date = '2025-10-04' WHERE event\_id = 6;



1. Deleting an event.

DELETE FROM Events WHERE Event\_Id = 6;



1. **Manage Track Attendees & Handle Events**

4) Manage Track Attendees & Handle Events

select

r.registration\_date

, r.registration\_amount

, e.event\_id

, e.event\_name

, e.event\_date

, e.event\_location

, e.event\_description

, a.attendee\_name

, a.attendee\_phone

, a.attendee\_email

, a.attendee\_city

from registrations as r

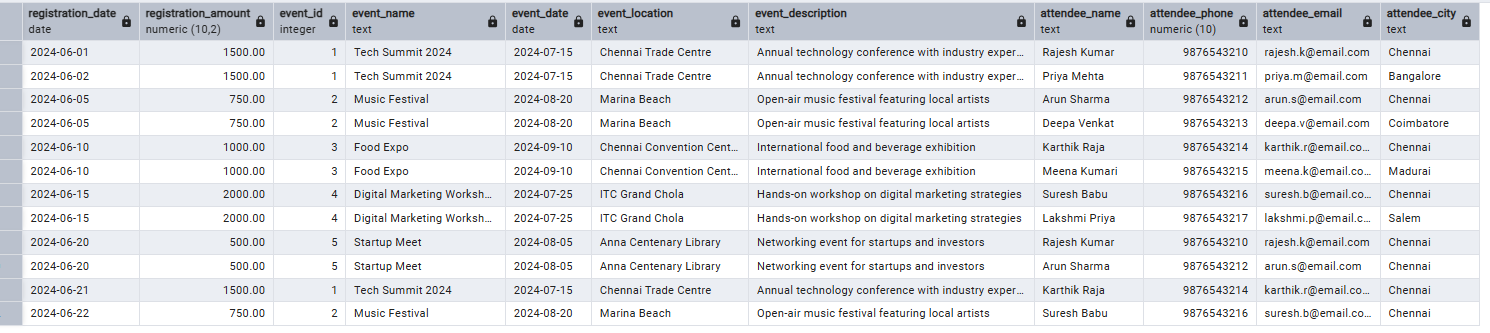
inner join events as e

on r.event\_id = e.event\_id

inner join attendees as a

on r.attendee\_id = a.attendee\_id

;



1. Inserting a new attendee.

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

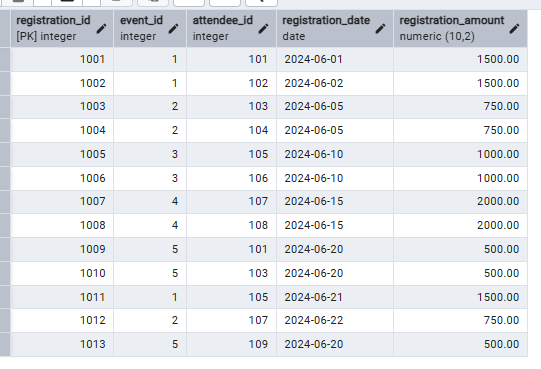
(109, 'Ram Kumar', '9785637218', 'ramk@email.com', 'Porur');



1. Registering an attendee for an event.

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

(1013, 5, 109, '2024-06-20', 500.00);



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

-- . Retrieve basic event information

SELECT

Event\_Id,

Event\_Name,

Event\_Date,

Event\_Location,

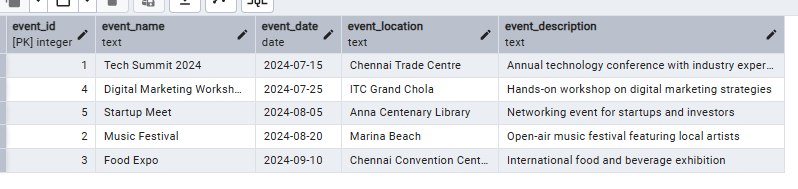
Event\_Description

FROM

Events

ORDER BY

Event\_Date;



-- . Generate attendee list for a specific event

SELECT

a.Attendee\_Name,

a.Attendee\_Email,

a.Attendee\_City,

r.Registration\_Date

FROM

Attendees a

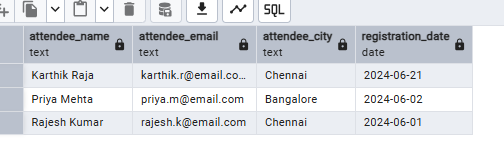
JOIN Registrations r ON a.Attendee\_Id = r.Attendee\_Id

WHERE

r.Event\_Id = 1

ORDER BY

a.Attendee\_Name;



-- . Calculate total attendees and revenue for each event

SELECT

e.Event\_Id,

e.Event\_Name,

COUNT(r.Attendee\_Id) as Total\_Attendees,

SUM(r.Registration\_Amount) as Total\_Revenue

FROM

Events e

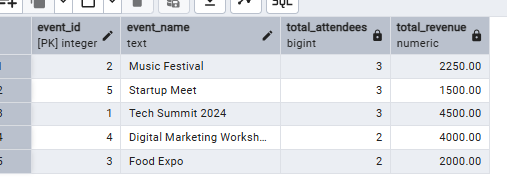
LEFT JOIN Registrations r ON e.Event\_Id = r.Event\_Id

GROUP BY

e.Event\_Id, e.Event\_Name

ORDER BY

Total\_Attendees DESC;



-- . Calculate attendance statistics by city

SELECT

a.Attendee\_City,

COUNT(DISTINCT r.Attendee\_Id) as Total\_Attendees,

COUNT(DISTINCT r.Event\_Id) as Events\_Attended

FROM

Attendees a

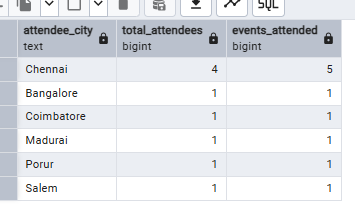
JOIN Registrations r ON a.Attendee\_Id = r.Attendee\_Id

GROUP BY

a.Attendee\_City

ORDER BY

Total\_Attendees DESC;



-- . Find the most popular events (by attendance)

SELECT

e.Event\_Name,

COUNT(r.Attendee\_Id) as Attendee\_Count

FROM

Events e

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

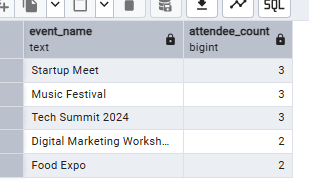
GROUP BY

e.Event\_Name

ORDER BY

Attendee\_Count DESC

LIMIT 5;



.-- Find attendees who have registered for multiple events

SELECT

a.Attendee\_Name,

COUNT(DISTINCT r.Event\_Id) as Events\_Registered

FROM

Attendees a

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

GROUP BY

a.Attendee\_Id, a.Attendee\_Name

HAVING

COUNT(DISTINCT r.Event\_Id) > 1

ORDER BY

Events\_Registered DESC;

