

## 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."

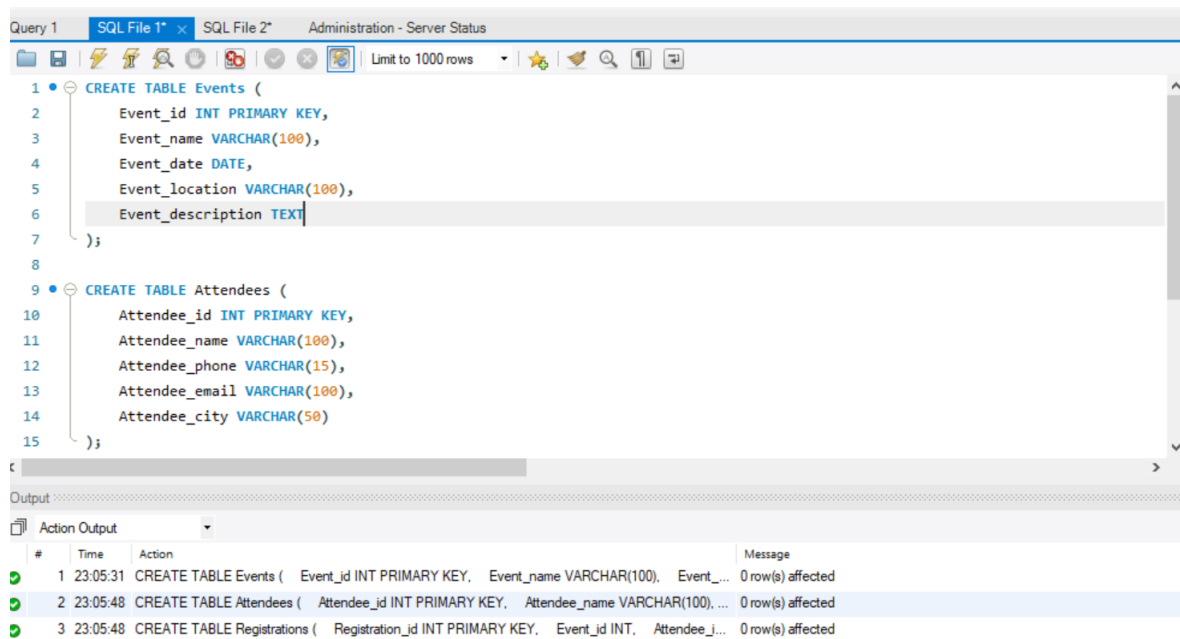
Create tables for Events, Attendees, and Registrations.

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

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

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

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.



```
1 CREATE TABLE Events (  
2     Event_id INT PRIMARY KEY,  
3     Event_name VARCHAR(100),  
4     Event_date DATE,  
5     Event_location VARCHAR(100),  
6     Event_description TEXT  
7 );  
8  
9 CREATE TABLE Attendees (  
10     Attendee_id INT PRIMARY KEY,  
11     Attendee_name VARCHAR(100),  
12     Attendee_phone VARCHAR(15),  
13     Attendee_email VARCHAR(100),  
14     Attendee_city VARCHAR(50)  
15 );
```

Output

#	Time	Action	Message
1	23:05:31	CREATE TABLE Events ( Event_id INT PRIMARY KEY, Event_name VARCHAR(100), Event_...	0 row(s) affected
2	23:05:48	CREATE TABLE Attendees ( Attendee_id INT PRIMARY KEY, Attendee_name VARCHAR(100), ...	0 row(s) affected
3	23:05:48	CREATE TABLE Registrations ( Registration_id INT PRIMARY KEY, Event_id INT, Attendee_i...	0 row(s) affected

### 2. Data Creation

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

Query 1 SQL File 1\* x SQL File 2\* Administration - Server Status

Limit to 1000 rows

```

10 (1, 'Bruce Wayne', '123-456-7890', 'bruce.wayne@waynecorp.com', 'Gotham'),
11 (2, 'Clark Kent', '987-654-3210', 'clark.kent@dailyplanet.com', 'Metropolis'),
12 (3, 'Diana Prince', '555-666-7777', 'diana.prince@themiscira.com', 'Themyscira'),
13 (4, 'Barry Allen', '333-444-5555', 'barry.allen@ccpd.com', 'Central City'),
14 (5, 'Arthur Curry', '222-333-4444', 'arthur.curry@atlantis.com', 'Atlantis');
15
16
17 INSERT INTO Registrations (Registration_id, Event_id, Attendee_id, Registration_date, Amount)
18 VALUES
19 (1, 1, 1, '2024-09-15', 100.00),
20 (2, 1, 2, '2024-09-16', 120.00),
21 (3, 2, 3, '2024-10-01', 150.00),
22 (4, 3, 4, '2024-11-20', 200.00),
23 (5, 3, 5, '2024-11-25', 250.00);
24

```

Output

Action Output

#	Time	Action	Message
✓ 1	23:06:59	INSERT INTO Events (Event_id, Event_name, Event_date, Event_location, Event_description) VALU...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
✓ 2	23:06:59	INSERT INTO Attendees (Attendee_id, Attendee_name, Attendee_phone, Attendee_email, Attendee_...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✓ 3	23:06:59	INSERT INTO Registrations (Registration_id, Event_id, Attendee_id, Registration_date, Amount) VALU...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

### 3. Manage Event Details

#### a) Inserting a new event.

Query 1 SQL File 1\* x SQL File 2\* Administration - Server Status

Limit to 1000 rows

```

1 • INSERT INTO Events (Event_id, Event_name, Event_date, Event_location, Event_description)
2 VALUES (4, 'Star City Charity Run', '2024-10-20', 'Star City', 'Annual marathon to raise funds for charity.');
```

Output

Action Output

#	Time	Action	Message
✓ 1	23:08:09	INSERT INTO Events (Event_id, Event_name, Event_date, Event_location, Event_description) VALU...	1 row(s) affected

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

Query 1 SQL File 1\* x SQL File 2\* Administration - Server Status

Limit to 1000 rows

```

1 • UPDATE Events
2 SET Event_location = 'Gotham City Plaza'
3 WHERE Event_id = 2;

```

Output

Action Output

#	Time	Action	Message
1	23:08:09	INSERT INTO Events (Event_id, Event_name, Event_date, Event_location, Event_description) VALU...	1 row(s) affected
2	23:08:44	UPDATE Events SET Event_location = 'Gotham City Plaza' WHERE Event_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

c) Deleting an event.

Query 1 SQL File 1\* x SQL File 2\* Administration - Server Status

Limit to 1000 rows

```

1 • DELETE FROM Events
2 WHERE Event_id = 4;

```

Output

Action Output

#	Time	Action	Message
1	23:08:09	INSERT INTO Events (Event_id, Event_name, Event_date, Event_location, Event_description) VALU...	1 row(s) affected
2	23:08:44	UPDATE Events SET Event_location = 'Gotham City Plaza' WHERE Event_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
3	23:09:16	DELETE FROM Events WHERE Event_id = 4	1 row(s) affected

### 3) Manage Track Attendees & Handle Events

a) Inserting a new attendee.

Query 1 SQL File 1\* SQL File 2\* Administration - Server Status

Limit to 1000 rows

```
1 • INSERT INTO Attendees (Attendee_id, Attendee_name, Attendee_phone, Attendee_email, Attendee_city)
2 VALUES (6, 'Kara Danvers', '444-333-2222', 'kara.danvers@deo.com', 'National City');
```

Output

Action Output

#	Time	Action	Message
✓ 1	23:10:08	INSERT INTO Attendees (Attendee_id, Attendee_name, Attendee_phone, Attendee_email, Attendee_...	1 row(s) affected

b) Registering an attendee for an event.

Query 1 SQL File 1\* SQL File 2\* Administration - Server Status

```
1 • INSERT INTO Registrations (Registration_id, Event_id, Attendee_id, Registration_date, Amount)
2 VALUES (6, 2, 6, '2024-08-27', 150.00);
```

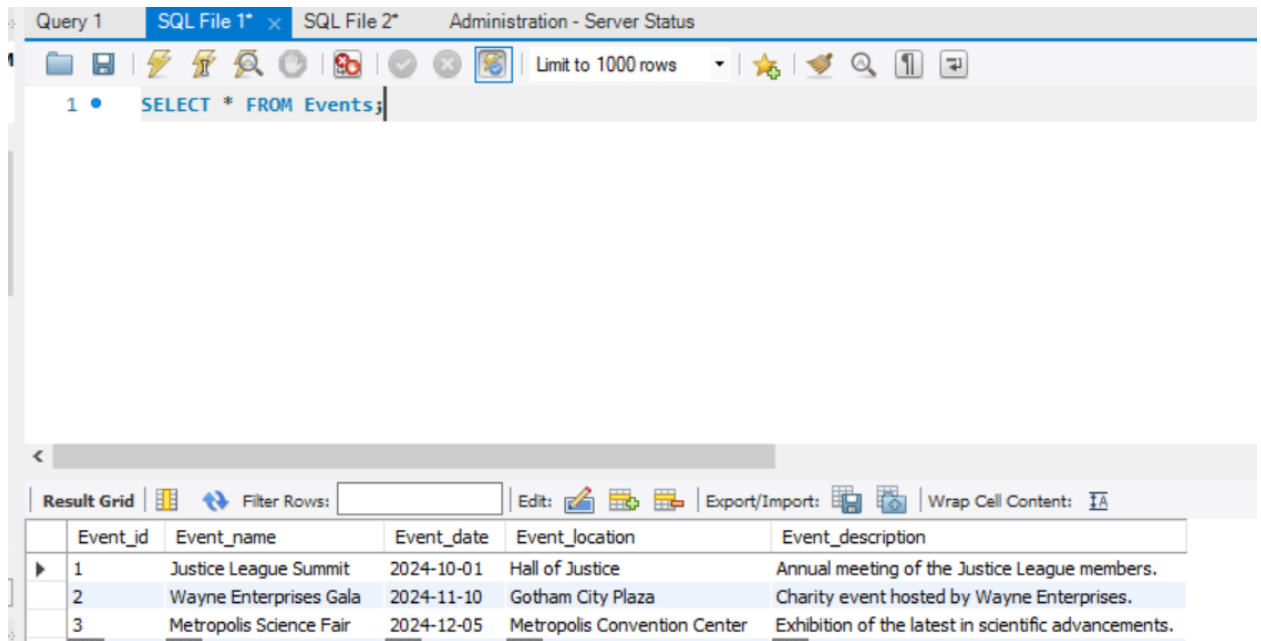
Output

Action Output

#	Time	Action	Message
✓ 1	23:10:08	INSERT INTO Attendees (Attendee_id, Attendee_name, Attendee_phone, Attendee_email, Attendee_...	1 row(s) affected
✓ 2	23:10:53	INSERT INTO Registrations (Registration_id, Event_id, Attendee_id, Registration_date, Amount) VALU...	1 row(s) affected

## 5. Develop queries to

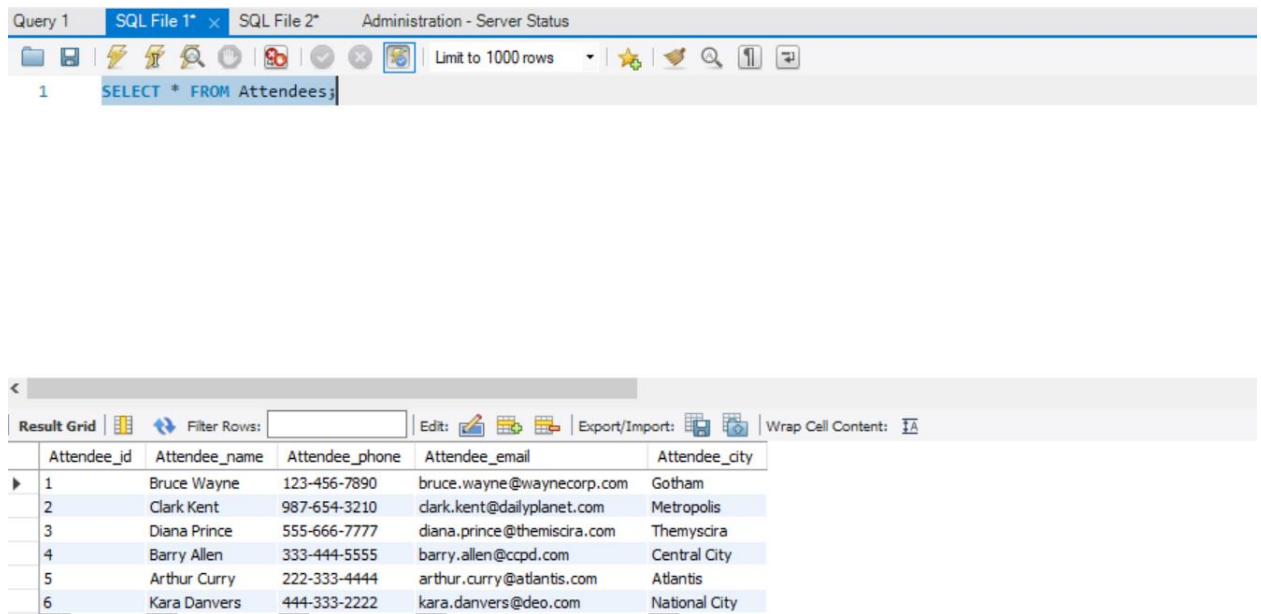
- retrieve event information,



The screenshot shows a database application interface. At the top, there are tabs for 'Query 1', 'SQL File 1\*', 'SQL File 2\*', and 'Administration - Server Status'. Below the tabs is a toolbar with various icons. The main area displays a SQL query: `SELECT * FROM Events;`. Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows:' input field and buttons for 'Edit', 'Export/Import', and 'Wrap Cell Content:'. The result grid contains a table with the following data:

Event_id	Event_name	Event_date	Event_location	Event_description
1	Justice League Summit	2024-10-01	Hall of Justice	Annual meeting of the Justice League members.
2	Wayne Enterprises Gala	2024-11-10	Gotham City Plaza	Charity event hosted by Wayne Enterprises.
3	Metropolis Science Fair	2024-12-05	Metropolis Convention Center	Exhibition of the latest in scientific advancements.

- generate attendee lists



The screenshot shows the same database application interface as above, but with a different query: `SELECT * FROM Attendees;`. The 'Result Grid' section displays a table with the following data:

Attendee_id	Attendee_name	Attendee_phone	Attendee_email	Attendee_city
1	Bruce Wayne	123-456-7890	bruce.wayne@waynecorp.com	Gotham
2	Clark Kent	987-654-3210	clark.kent@dailyplanet.com	Metropolis
3	Diana Prince	555-666-7777	diana.prince@themiscira.com	Themyscira
4	Barry Allen	333-444-5555	barry.allen@ccpd.com	Central City
5	Arthur Curry	222-333-4444	arthur.curry@atlantis.com	Atlantis
6	Kara Danvers	444-333-2222	kara.danvers@deo.com	National City

Attendee list for specific event

Query 1   SQL File 1\*   SQL File 2\*   Administration - Server Status

Limit to 1000 rows

```

1  SELECT A.Attendee_name, A.Attendee_email, A.Attendee_phone
2  FROM Attendees A
3  JOIN Registrations R ON A.Attendee_id = R.Attendee_id
4  WHERE R.Event_id = '1';

```

Result Grid   Filter Rows:   Export:   Wrap Cell Content: [IA](#)

	Attendee_name	Attendee_email	Attendee_phone
▶	Bruce Wayne	bruce.wayne@waynecorp.com	123-456-7890
	Clark Kent	clark.kent@dailyplanet.com	987-654-3210

- calculate event attendance statistics.

Query 1   SQL File 1\*   SQL File 2\*   Administration - Server Status

Limit to 1000 rows

```

1  SELECT
2      E.Event_name,
3      COUNT(R.Registration_id) AS Total_Attendees,
4      SUM(R.Amount) AS Total_Amount,
5      AVG(R.Amount) AS Average_Amount,
6      MIN(R.Amount) AS Minimum_Amount,
7      MAX(R.Amount) AS Maximum_Amount
8  FROM Events E
9  LEFT JOIN Registrations R ON E.Event_id = R.Event_id
10 GROUP BY E.Event_id, E.Event_name;

```

Result Grid   Filter Rows:   Export:   Wrap Cell Content: [IA](#)

	Event_name	Total_Attendees	Total_Amount	Average_Amount	Minimum_Amount	Maximum_Amount
▶	Justice League Summit	2	220.00	110.000000	100.00	120.00
	Wayne Enterprises Gala	2	300.00	150.000000	150.00	150.00
	Metropolis Science Fair	2	450.00	225.000000	200.00	250.00