QUERY SHEET

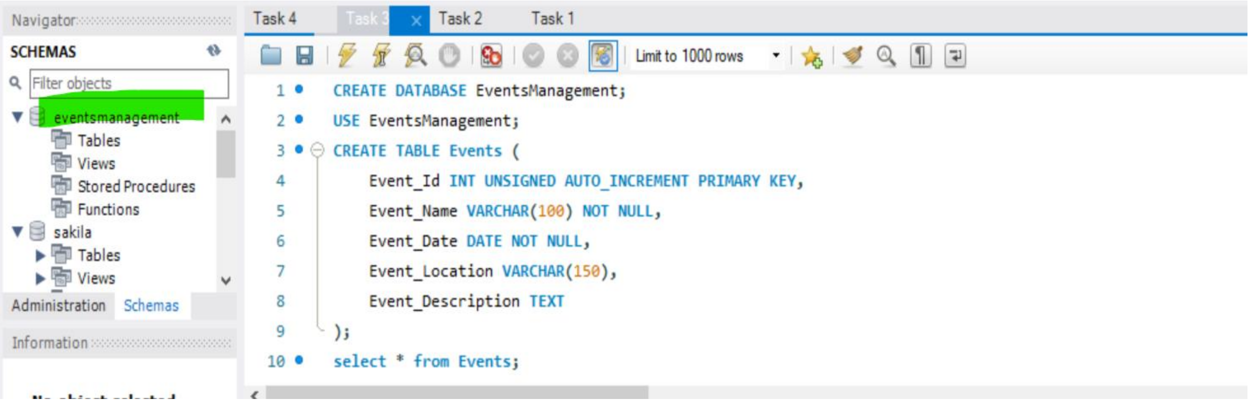
Task 3

Project: Event Management System using MySQL

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

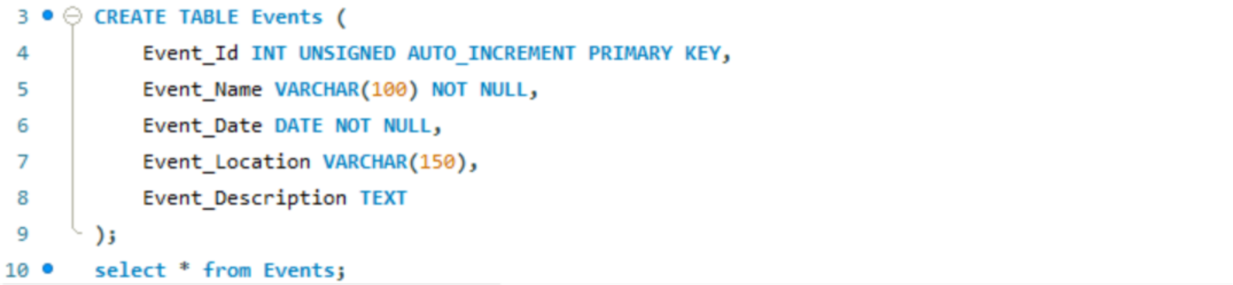
a. Create a database named "EventsManagement."



```sql
CREATE DATABASE EventsManagement;
USE EventsManagement;
CREATE TABLE Events (
    Event_Id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Event_Name VARCHAR(100) NOT NULL,
    Event_Date DATE NOT NULL,
    Event_Location VARCHAR(150),
    Event_Description TEXT
);
select * from Events;
```

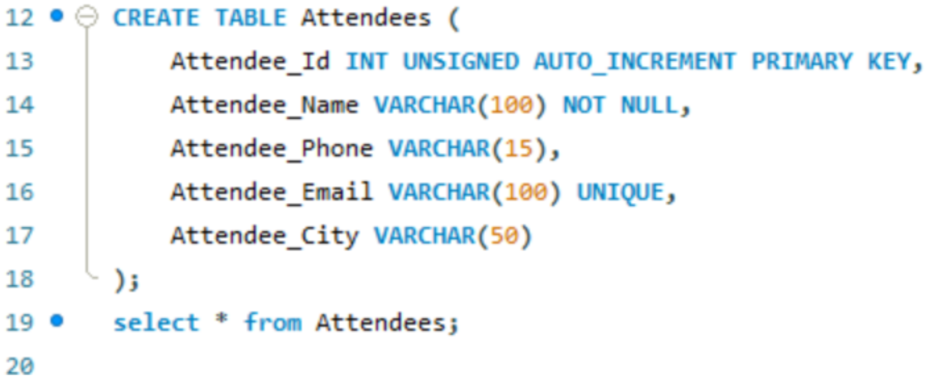
b. Create tables for Events, Attendees, and Registrations.

i. Events- Event_Id, Event_Name, Event_Date, Event_Location, Event_Description



```sql
CREATE TABLE Events (
    Event_Id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Event_Name VARCHAR(100) NOT NULL,
    Event_Date DATE NOT NULL,
    Event_Location VARCHAR(150),
    Event_Description TEXT
);
select * from Events;
```

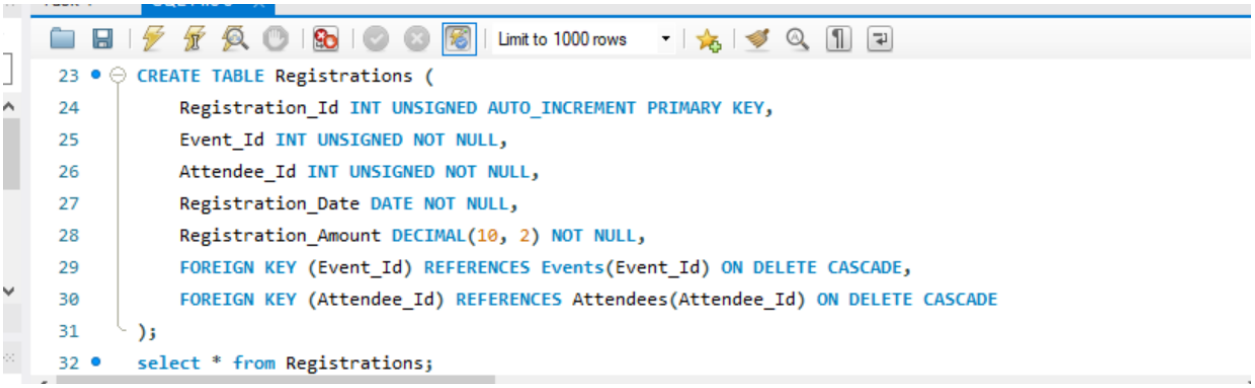
ii. Attendees- Attendee_Id, Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City



```sql
CREATE TABLE Attendees (
    Attendee_Id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Attendee_Name VARCHAR(100) NOT NULL,
    Attendee_Phone VARCHAR(15),
    Attendee_Email VARCHAR(100) UNIQUE,
    Attendee_City VARCHAR(50)
);
select * from Attendees;
```

iii. 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.

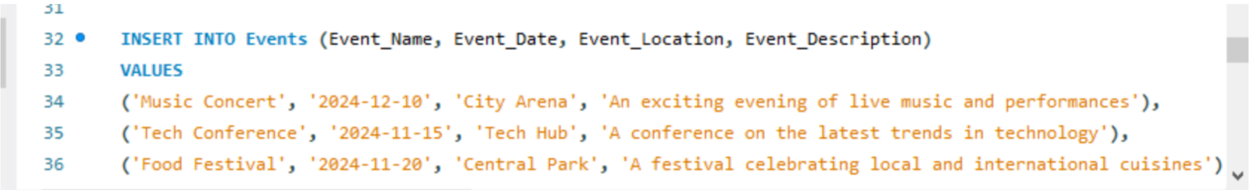


```sql
CREATE TABLE Registrations (
    Registration_Id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Event_Id INT UNSIGNED NOT NULL,
    Attendee_Id INT UNSIGNED NOT NULL,
    Registration_Date DATE NOT NULL,
    Registration_Amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Event_Id) REFERENCES Events(Event_Id) ON DELETE CASCADE,
    FOREIGN KEY (Attendee_Id) REFERENCES Attendees(Attendee_Id) ON DELETE CASCADE
);
select * from Registrations;
```

2. Data Creation

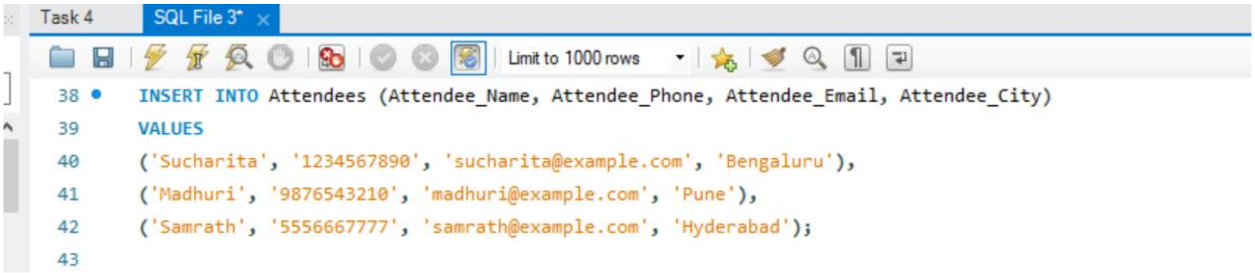
Insert some sample data for Events, Attendees, and Registrations tables with respective fields.

a. Insert data in Events Table



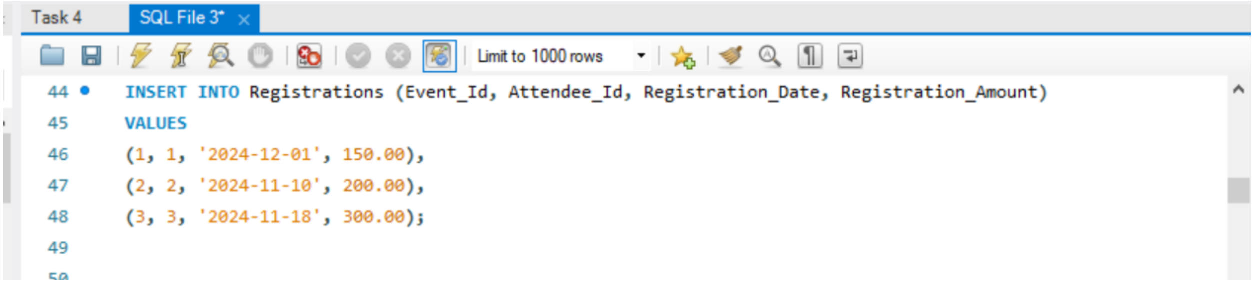
```sql
INSERT INTO Events (Event_Name, Event_Date, Event_Location, Event_Description)
VALUES
('Music Concert', '2024-12-10', 'City Arena', 'An exciting evening of live music and performances'),
('Tech Conference', '2024-11-15', 'Tech Hub', 'A conference on the latest trends in technology'),
('Food Festival', '2024-11-20', 'Central Park', 'A festival celebrating local and international cuisines')
```

b. Insert data in Attendees Table



```sql
INSERT INTO Attendees (Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City)
VALUES
('Sucharita', '1234567890', 'sucharita@example.com', 'Bengaluru'),
('Madhuri', '9876543210', 'madhuri@example.com', 'Pune'),
('Samrath', '5556667777', 'samrath@example.com', 'Hyderabad');
```

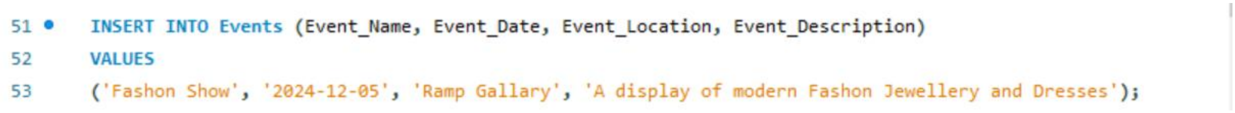
c. Insert data in Registrations Table



```sql
INSERT INTO Registrations (Event_Id, Attendee_Id, Registration_Date, Registration_Amount)
VALUES
(1, 1, '2024-12-01', 150.00),
(2, 2, '2024-11-10', 200.00),
(3, 3, '2024-11-18', 300.00);
```

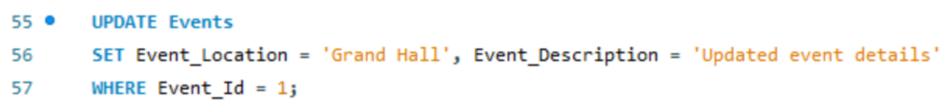
3. Manage Event Details

a) Inserting a new event.



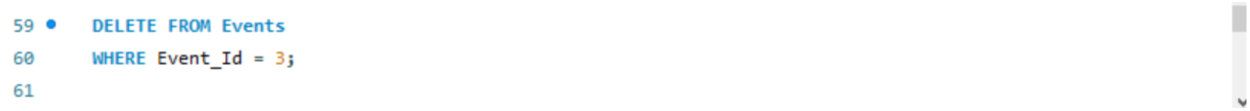
```sql
INSERT INTO Events (Event_Name, Event_Date, Event_Location, Event_Description)
VALUES
('Fashon Show', '2024-12-05', 'Ramp Gallary', 'A display of modern Fashon Jewellery and Dresses');
```

b) Updating an event's information.



```sql
UPDATE Events
SET Event_Location = 'Grand Hall', Event_Description = 'Updated event details'
WHERE Event_Id = 1;
```

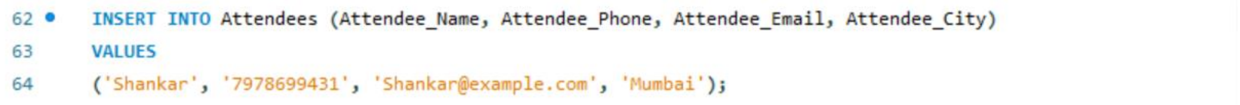
c) Deleting an event.



```sql
DELETE FROM Events
WHERE Event_Id = 3;
```

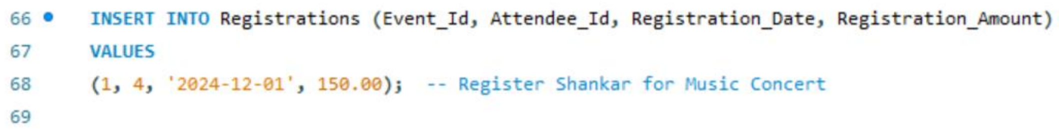
4) Manage Track Attendees & Handle Events

a) Inserting a new attendee.



```sql
INSERT INTO Attendees (Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City)
VALUES
('Shankar', '7978699431', 'Shankar@example.com', 'Mumbai');
```

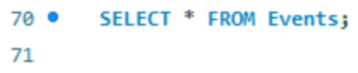
b) Registering an attendee for an event.



```sql
INSERT INTO Registrations (Event_Id, Attendee_Id, Registration_Date, Registration_Amount)
VALUES
(1, 4, '2024-12-01', 150.00);  -- Register Shankar for Music Concert
```

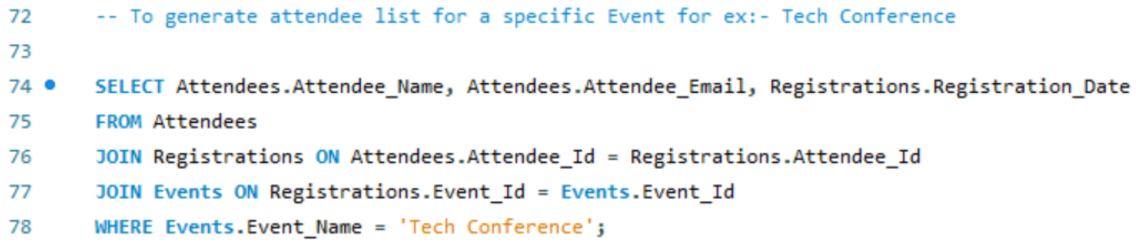
5.Develop queries to retrieve event information, generate attendee lists, and calculate event attendance statistics.

a) To retrieve event information



```sql
SELECT * FROM Events;
```

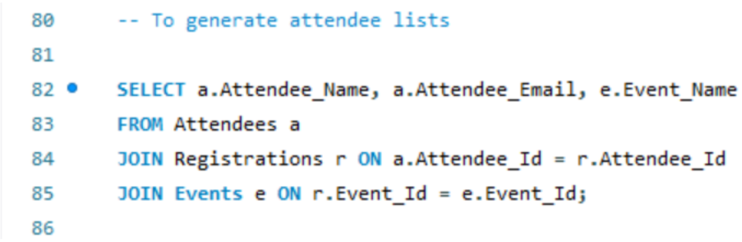
b) To generate attendee list for a specific Event (Tech Conference)



```
-- To generate attendee list for a specific Event for ex:- Tech Conference

SELECT Attendees.Attendee_Name, Attendees.Attendee_Email, Registrations.Registration_Date
FROM Attendees
JOIN Registrations ON Attendees.Attendee_Id = Registrations.Attendee_Id
JOIN Events ON Registrations.Event_Id = Events.Event_Id
WHERE Events.Event_Name = 'Tech Conference';
```

c) To generate attendee lists

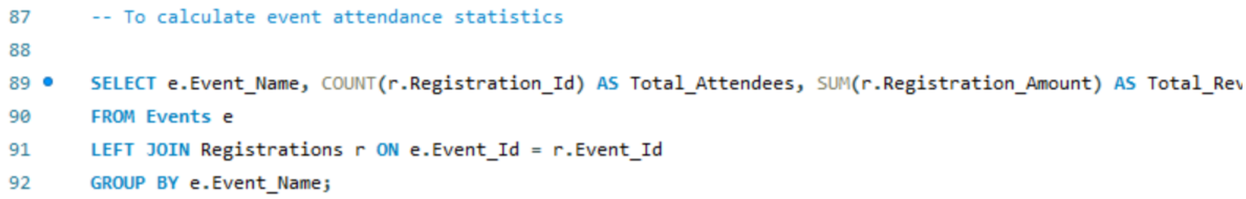


```
-- To generate attendee lists

SELECT a.Attendee_Name, a.Attendee_Email, e.Event_Name
FROM Attendees a
JOIN Registrations r ON a.Attendee_Id = r.Attendee_Id
JOIN Events e ON r.Event_Id = e.Event_Id;

```

d) To calculate event attendance statistics



```
-- To calculate event attendance statistics

SELECT e.Event_Name, COUNT(r.Registration_Id) AS Total_Attendees, SUM(r.Registration_Amount) AS Total_Rev
FROM Events e
LEFT JOIN Registrations r ON e.Event_Id = r.Event_Id
GROUP BY e.Event_Name;
```

Full Query

1. Database Creation

```
CREATE DATABASE EventsManagement;

USE EventsManagement;
```

2. Create Tables

a. Create the Events table

```
CREATE TABLE Events (
    Event_Id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Event_Name VARCHAR(100) NOT NULL,
    Event_Date DATE NOT NULL,
    Event_Location VARCHAR(150),
    Event_Description TEXT
);
select * from Events;
```

b. Create the Attendees table

```
CREATE TABLE Attendees (
    Attendee_Id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Attendee_Name VARCHAR(100) NOT NULL,
    Attendee_Phone VARCHAR(15),
    Attendee_Email VARCHAR(100) UNIQUE,
    Attendee_City VARCHAR(50)
);
select * from Attendees;
```

c. Create the Registrations table

```
CREATE TABLE Registrations (
    Registration_Id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    Event_Id INT UNSIGNED NOT NULL,
    Attendee_Id INT UNSIGNED NOT NULL,
    Registration_Date DATE NOT NULL,
    Registration_Amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Event_Id) REFERENCES Events(Event_Id) ON DELETE CASCADE,
    FOREIGN KEY (Attendee_Id) REFERENCES Attendees(Attendee_Id) ON DELETE CASCADE
);
select * from Registrations;
```

3. Data Creation

a. Inserting sample data for Events

```
INSERT INTO Events (Event_Name, Event_Date, Event_Location, Event_Description)
VALUES
```

```
('Music Concert', '2024-12-10', 'City Arena', 'An exciting evening of live music and performances'),

('Tech Conference', '2024-11-15', 'Tech Hub', 'A conference on the latest trends in technology'),

('Food Festival', '2024-11-20', 'Central Park', 'A festival celebrating local and international cuisines');
```

b. Inserting sample data for Attendees

```
INSERT INTO Attendees (Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City)

VALUES

('Sucharita', '1234567890', 'sucharita@example.com', 'Bengaluru'),

('Madhuri', '9876543210', 'madhuri@example.com', 'Pune'),

('Samrath', '5556667777', 'samrath@example.com', 'Hyderabad');
```

c. Inserting sample data for Registrations

```
INSERT INTO Registrations (Event_Id, Attendee_Id, Registration_Date, Registration_Amount)

VALUES

(1, 1, '2024-12-01', 150.00),

(2, 2, '2024-11-10', 200.00),

(3, 3, '2024-11-18', 300.00);
```

4. Manage Event Details

a) **Inserting a New Event**

```
INSERT INTO Events (Event_Name, Event_Date, Event_Location, Event_Description)

VALUES

('Fashon Show', '2024-12-05', 'Ramp Gallary', 'A display of modern Fashon Jewellery and Dresses');
```

b) **Updating an Event's Information**

```
UPDATE Events

SET Event_Location = 'Grand Hall', Event_Description = 'Updated event details'

WHERE Event_Id = 1;
```

c) **Deleting an Event**

```
DELETE FROM Events
```

```
WHERE Event_Id = 3;
```

5. Manage Track Attendees & Handle Events

a) **Inserting a New Attendee**

```
INSERT INTO Attendees (Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City)

VALUES

('Shankar', '7978699431', 'Shankar@example.com', 'Mumbai');
```

b) **Registering an Attendee for an Event**

```
INSERT INTO Registrations (Event_Id, Attendee_Id, Registration_Date, Registration_Amount)

VALUES

(1, 4, '2024-12-01', 150.00);  -- Register Shankar for Music Concert
```

6. Develop Queries to Retrieve Event Information and Attendance Statistics

a) **Retrieve All Event Information**

```
SELECT * FROM Events;
```

b) **Retrieve List of Attendees for a Specific Event**

To generate attendee list for a specific Event for ex:- Tech Conference

```
SELECT Attendees.Attendee_Name, Attendees.Attendee_Email,
Registrations.Registration_Date

FROM Attendees

JOIN Registrations ON Attendees.Attendee_Id = Registrations.Attendee_Id

JOIN Events ON Registrations.Event_Id = Events.Event_Id

WHERE Events.Event_Name = 'Tech Conference';
```

c) To generate attendee lists

```
SELECT a.Attendee_Name, a.Attendee_Email, e.Event_Name

FROM Attendees a

JOIN Registrations r ON a.Attendee_Id = r.Attendee_Id

JOIN Events e ON r.Event_Id = e.Event_Id;
```

d) To calculate event attendance statistics

```
SELECT e.Event_Name, COUNT(r.Registration_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_Name;
```