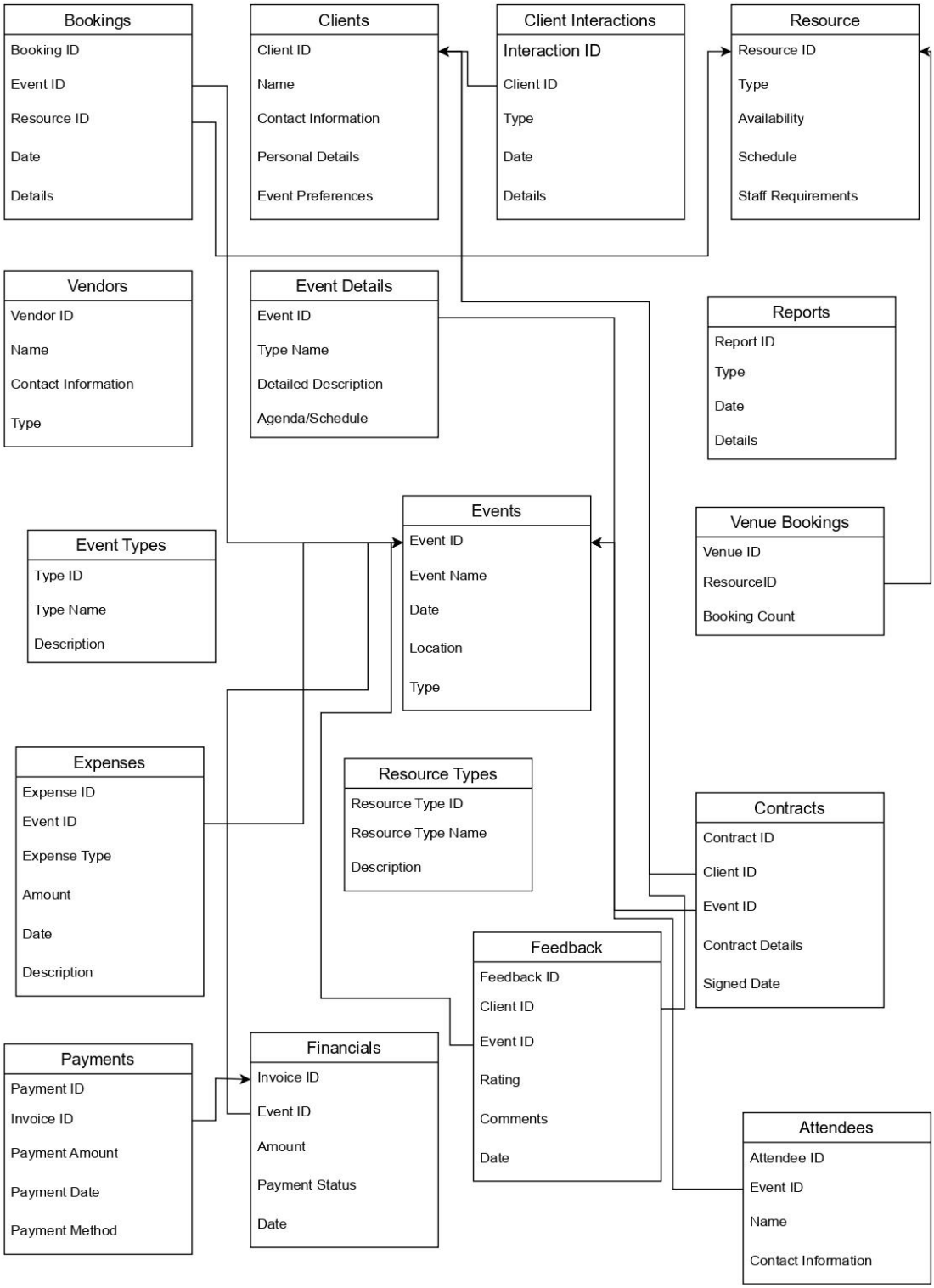
Report Name: Event Management System

Introduction:
This report focuses on the development of an event management system database, detailing the creation of entities and their relationships, along with SQL queries to retrieve key insights. The system handles events, clients, resources, vendors, and financials, ensuring effective management of event-related data.

Objective:
- Design a relational database schema for event management.
- Define relationships between entities such as Events, Clients, and Resources.
- Populate the database with sample data.
- Write SQL queries to analyze data and generate reports.

Project Work:
The event management system's schema was designed based on a relational model, with tables representing key entities like Events, Clients, Vendors, and Resources. SQL queries were written to perform tasks such as calculating event revenue, finding the most active clients, and managing attendee data.

Diagram:


Bookings
Booking ID
Event ID
Resource ID
Date
Details
Clients
Client ID
Name
Contact Information
Personal Details
Event Preferences
Client Interactions
Interaction ID
Client ID
Type
Date
Details
Resource
Resource ID
Type
Availability
Schedule
Staff Requirements
Vendors
Vendor ID
Name
Contact Information
Type
Event Details
Event ID
Type Name
Detailed Description
Agenda/Schedule
Reports
Report ID
Type
Date
Details
Events
Event ID
Event Name
Date
Location
Type
Event Types
Type ID
Type Name
Description
Venue Bookings
Venue ID
ResourceID
Booking Count
Expenses
Expense ID
Event ID
Expense Type
Amount
Date
Description
Resource Types
Resource Type ID
Resource Type Name
Description
Contracts
Contract ID
Client ID
Event ID
Contract Details
Signed Date
Feedback
Feedback ID
Client ID
Event ID
Rating
Comments
Date
Payments
Payment ID
Invoice ID
Payment Amount
Payment Date
Payment Method
Financials
Invoice ID
Event ID
Amount
Payment Status
Date
Attendees
Attendee ID
Event ID
Name
Contact Information

Codes:

```
-- Create Event Types table
CREATE TABLE EventTypes (
   TypeID INT PRIMARY KEY,
   TypeName VARCHAR(255),
   Description TEXT
);

-- Insert data into EventTypes
INSERT INTO EventTypes (TypeID, TypeName, Description) VALUES
(1, 'Conference', 'Business and academic conferences'),
(2, 'Wedding', 'Marriage ceremonies and receptions'),
(3, 'Concert', 'Live music performances');

-- Create Events table
CREATE TABLE Events (
   EventID INT PRIMARY KEY,
   EventName VARCHAR(255),
   Date DATE,
   Location VARCHAR(255),
   Type INT,
   FOREIGN KEY (Type) REFERENCES EventTypes(TypeID)
);

-- Insert data into Events
INSERT INTO Events (EventID, EventName, Date, Location, Type) VALUES
(101, 'Tech Expo 2024', '2024-09-15', 'San Francisco', 1),
(102, 'John and Jane Wedding', '2024-10-10', 'Los Angeles', 2),
(103, 'Rock Fest', '2024-11-20', 'New York', 3);

-- Create Clients table
CREATE TABLE Clients (
   ClientID INT PRIMARY KEY,
   Name VARCHAR(255),
   ContactInformation VARCHAR(255),
   PersonalDetails TEXT,
   EventPreferences TEXT
);

-- Insert data into Clients
INSERT INTO Clients (ClientID, Name, ContactInformation, PersonalDetails, EventPreferences) VALUES
(1, 'Alice Smith', 'alice.smith@example.com', 'VIP Client', 'Prefers outdoor venues'),
(2, 'John Doe', 'john.doe@example.com', 'Frequent Attendee', 'Prefers conference settings'),
(3, 'Emily White', 'emily.white@example.com', 'Music Enthusiast', 'Prefers large concerts');

-- Create Client Interactions table
```

```
CREATE TABLE ClientInteractions (
   InteractionID INT PRIMARY KEY,
   ClientID INT,
   Type VARCHAR(255),
   Date DATE,
   Details TEXT,
   FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)
);

-- Insert data into Client Interactions
INSERT INTO ClientInteractions (InteractionID, ClientID, Type, Date, Details) VALUES
(1, 1, 'Consultation', '2024-07-01', 'Discussed wedding plans'),
(2, 2, 'Feedback', '2024-08-05', 'Provided feedback on last conference'),
(3, 3, 'Contract', '2024-09-01', 'Signed concert event contract');

-- Create Resources table
CREATE TABLE Resources (
   ResourceID INT PRIMARY KEY,
   Type VARCHAR(255),
   Availability BOOLEAN,
   Schedule TEXT,
   StaffRequirements INT
);

-- Insert data into Resources
INSERT INTO Resources (ResourceID, Type, Availability, Schedule, StaffRequirements) VALUES
(1, 'venue', TRUE, '2024-09-15 10:00-18:00', 10),
(2, 'staff', TRUE, '2024-10-10 12:00-22:00', 15),
(3, 'equipment', TRUE, '2024-11-20 18:00-23:00', 5);

-- Create Vendors table
CREATE TABLE Vendors (
   VendorID INT PRIMARY KEY,
   Name VARCHAR(255),
   ContactInformation VARCHAR(255),
   Type VARCHAR(255)
);

-- Insert data into Vendors
INSERT INTO Vendors (VendorID, Name, ContactInformation, Type) VALUES
(1, 'SoundPro Inc.', 'contact@soundpro.com', 'equipment supplier'),
(2, 'Grand Venue Co.', 'info@grandvenue.com', 'venue provider'),
(3, 'Elite Catering', 'orders@elitecatering.com', 'catering services');

-- Create Bookings table
CREATE TABLE Bookings (
   BookingID INT PRIMARY KEY,
   EventID INT,
```

```
    ResourceID INT,
    Date DATE,
    Details TEXT,
    FOREIGN KEY (EventID) REFERENCES Events(EventID),
    FOREIGN KEY (ResourceID) REFERENCES Resources(ResourceID)
);

-- Insert data into Bookings
INSERT INTO Bookings (BookingID, EventID, ResourceID, Date, Details) VALUES
(1, 101, 1, '2024-09-15', 'Tech Expo venue booking'),
(2, 102, 2, '2024-10-10', 'Wedding venue booking'),
(3, 103, 3, '2024-11-20', 'Concert equipment booking');

-- Create Financials table
CREATE TABLE Financials (
    InvoiceID INT PRIMARY KEY,
    EventID INT,
    Amount DECIMAL(10, 2),
    PaymentStatus VARCHAR(50),
    Date DATE,
    FOREIGN KEY (EventID) REFERENCES Events(EventID)
);

-- Insert data into Financials
INSERT INTO Financials (InvoiceID, EventID, Amount, PaymentStatus, Date) VALUES
(1, 101, 5000.00, 'Paid', '2024-08-20'),
(2, 102, 15000.00, 'Pending', '2024-09-05'),
(3, 103, 25000.00, 'Paid', '2024-09-10');

-- Create Reports table
CREATE TABLE Reports (
    ReportID INT PRIMARY KEY,
    Type VARCHAR(255),
    Date DATE,
    Details TEXT
);

-- Insert data into Reports
INSERT INTO Reports (ReportID, Type, Date, Details) VALUES
(1, 'Performance', '2024-09-16', 'Tech Expo performance report'),
(2, 'Client Satisfaction', '2024-10-11', 'Wedding client feedback'),
(3, 'Resource Utilization', '2024-11-21', 'Concert resource usage');

-- Create EventDetails table
CREATE TABLE EventDetails (
    EventID INT PRIMARY KEY,
    TypeName VARCHAR(255),
    DetailedDescription TEXT,
```

```
    AgendaSchedule TEXT,
    FOREIGN KEY (EventID) REFERENCES Events(EventID)
);

-- Insert data into EventDetails
INSERT INTO EventDetails (EventID, TypeName, DetailedDescription, AgendaSchedule) VALUES
(101, 'Conference', 'A large technology expo with speakers and demos.', 'Opening ceremony, Panel discussions, Closing remarks'),
(102, 'Wedding', 'A formal wedding ceremony with a reception afterward.', 'Ceremony, Dinner, Dancing'),
(103, 'Concert', 'A live music concert featuring multiple bands.', 'Opening act, Main performance, Encore');

-- Create Contracts table
CREATE TABLE Contracts (
    ContractID INT PRIMARY KEY,
    ClientID INT,
    EventID INT,
    ContractDetails TEXT,
    SignedDate DATE,
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
    FOREIGN KEY (EventID) REFERENCES Events(EventID)
);

-- Insert data into Contracts
INSERT INTO Contracts (ContractID, ClientID, EventID, ContractDetails, SignedDate) VALUES
(1, 1, 101, 'Contract for Tech Expo event management.', '2024-07-01'),
(2, 2, 102, 'Contract for wedding planning services.', '2024-08-01'),
(3, 3, 103, 'Contract for concert organization.', '2024-09-01');

-- Create Payments table
CREATE TABLE Payments (
    PaymentID INT PRIMARY KEY,
    InvoiceID INT,
    PaymentAmount DECIMAL(10, 2),
    PaymentDate DATE,
    PaymentMethod VARCHAR(255),
    FOREIGN KEY (InvoiceID) REFERENCES Financials(InvoiceID)
);

-- Insert data into Payments
INSERT INTO Payments (PaymentID, InvoiceID, PaymentAmount, PaymentDate, PaymentMethod) VALUES
(1, 1, 5000.00, '2024-08-20', 'Credit Card'),
(2, 2, 15000.00, '2024-09-05', 'Bank Transfer'),
(3, 3, 25000.00, '2024-09-10', 'Credit Card');
```

```
-- Create Expenses table
CREATE TABLE Expenses (
   ExpenseID INT PRIMARY KEY,
   EventID INT,
   ExpenseType VARCHAR(255),
   Amount DECIMAL(10, 2),
   Date DATE,
   Description TEXT,
   FOREIGN KEY (EventID) REFERENCES Events(EventID)
);

-- Insert data into Expenses
INSERT INTO Expenses (ExpenseID, EventID, ExpenseType, Amount, Date, Description) VALUES
(1, 101, 'Venue rental', 3000.00, '2024-09-15', 'Rent for Tech Expo venue'),
(2, 102, 'Catering', 12000.00, '2024-10-10', 'Wedding catering services'),
(3, 103, 'Equipment rental', 10000.00, '2024-11-20', 'Concert sound equipment rental');

-- Create ResourceTypes table
CREATE TABLE ResourceTypes (
   ResourceTypeID INT PRIMARY KEY,
   ResourceTypeName VARCHAR(255),
   Description TEXT
);

-- Insert data into ResourceTypes
INSERT INTO ResourceTypes (ResourceTypeID, ResourceTypeName, Description) VALUES
(1, 'Venue', 'Locations for events'),
(2, 'Staff', 'Event support staff'),
(3, 'Equipment', 'Event equipment such as sound and lighting');

-- Create Feedback table
CREATE TABLE Feedback (
   FeedbackID INT PRIMARY KEY,
   ClientID INT,
   EventID INT,
   Rating INT,
   Comments TEXT,
   Date DATE,
   FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
   FOREIGN KEY (EventID) REFERENCES Events(EventID)
);

-- Insert data into Feedback
INSERT INTO Feedback (FeedbackID, ClientID, EventID, Rating, Comments, Date) VALUES
(1, 1, 101, 4, 'Great event, but more food options would be appreciated.', '2024-09-16'),
(2, 2, 102, 5, 'Perfect wedding, everything was amazing!', '2024-10-11'),
(3, 3, 103, 5, 'Best concert ever!', '2024-11-21');
```

```
-- Create Attendees table
CREATE TABLE Attendees (
   AttendeeID INT PRIMARY KEY,
   EventID INT,
   Name VARCHAR(255),
   ContactInformation VARCHAR(255),
   FOREIGN KEY (EventID) REFERENCES Events(EventID)
);

-- Insert data into Attendees
INSERT INTO Attendees (AttendeeID, EventID, Name, ContactInformation) VALUES
(1, 101, 'Michael Brown', 'michael.brown@example.com'),
(2, 102, 'Sarah Lee', 'sarah.lee@example.com'),
(3, 103, 'David Clark', 'david.clark@example.com');

-- Create VenueBookings table
CREATE TABLE VenueBookings (
   VenueID INT PRIMARY KEY,
   ResourceID INT,
   BookingCount INT,
   FOREIGN KEY (ResourceID) REFERENCES Resources(ResourceID)
);

-- Insert data into VenueBookings
INSERT INTO VenueBookings (VenueID, ResourceID, BookingCount) VALUES
(1, 1, 5),
(2, 2, 10),
(3, 3, 8);
```

Questions:

1. Find the event that generated the highest revenue.

`SELECT EventID, SUM(Amount) AS TotalRevenue FROM Financials GROUP BY EventID ORDER BY TotalRevenue DESC LIMIT 1;`

2. Calculate the average number of attendees per event type.

```
SELECT ET.TypeName, AVG(AttendeeCount) AS AverageAttendees
FROM (
   SELECT E.Type, COUNT(A.AttendeeID) AS AttendeeCount
   FROM Events E
   JOIN Attendees A ON E.EventID = A.EventID
   GROUP BY E.EventID
) AS EventAttendeeCounts
JOIN EventTypes ET ON EventAttendeeCounts.Type = ET.TypeID
GROUP BY ET.TypeName
LIMIT 0, 25;
```

3. Retrieve the minimum number of staff required for any event.

`SELECT MIN(StaffRequirements) AS MinStaffRequired FROM Resources WHERE Type = 'staff';`

4. Find the client who has hosted the most events.

```
SELECT C.ClientID, C.Name, COUNT(E.EventID) AS EventCount
FROM Clients C
JOIN Contracts Co ON C.ClientID = Co.ClientID
JOIN Events E ON Co.EventID = E.EventID
GROUP BY C.ClientID, C.Name
ORDER BY EventCount DESC
LIMIT 1;
```

5. List the top 5 most booked venues.

`SELECT ResourceID, COUNT(*) AS BookingCount FROM Bookings WHERE ResourceID IN (SELECT ResourceID FROM Resources WHERE Type = 'venue') GROUP BY ResourceID ORDER BY BookingCount DESC LIMIT 5;

Conclusion:

The event management system's schema and queries enable effective handling of events, clients, and financials. The system provides insights through well-structured queries, making it a reliable solution for managing and analyzing event-related data.