CIS-552 PROJECT REPORT CONFERENCE MANAGEMENT SYSTEM

REPORT BY: GROUP-14
NANDINI LOKESH REDDY - 02074628
VUTA SRIKANTH REDDY – 02074624

PROFESSOR:
YUKI LUO

CONTENT:

1. PROJECT OVERVIEW:

The Conference Management System is a comprehensive solution designed to streamline the organization and execution of conferences, ensuring a smooth and efficient experience for organizers, speakers, attendees, and sponsors. The system employs a relational database model to store and manage crucial information about conferences, including details about speakers, sessions, attendees, feedback, and more.

Key features of the system include the ability to create and manage conferences with relevant information such as name, date, venue, and associated sessions. Speakers play a pivotal role, with their profiles and sessions integrated seamlessly into the platform. Attendees can register for conferences and sessions, and the system facilitates the tracking of attendance. Session feedback mechanisms allow attendees to provide valuable input, contributing to the continuous improvement of future conferences.

The system prioritizes data integrity through the normalization and organization of data tables, ensuring optimal efficiency and flexibility. SQL queries are strategically crafted to extract insightful information, such as conference statistics, speaker lists, attendee counts, and top-rated sessions. Additionally, the system incorporates advanced features, such as identifying overlapping sessions and providing a platform for sponsors to associate with conferences.

With its robust database architecture and well-defined queries, the Conference Management System is poised to enhance the management of conferences, offering a user-friendly interface and empowering organizers to make informed decisions for the success of their events.

2. PROBLEM STATEMENT:

"Organizing and managing conferences is a complex task involving numerous interrelated elements, including conference details, speaker information, session scheduling, attendee registration, and feedback collection. Traditional methods of manual organization and tracking often lead to inefficiencies, errors, and a lack of data-driven decision-making. The absence of a centralized and automated system hinders the seamless execution of conferences, impacting organizers, speakers, sponsors, and attendees. The challenge is to develop a robust Conference Management System that leverages a relational database structure and SQL queries to streamline the entire conference lifecycle. This system should facilitate easy conference creation, speaker management, attendee registration, session tracking, and feedback analysis. Additionally, the system should offer advanced features such as identifying overlapping sessions and integrating sponsor associations. By addressing these challenges, the Conference Management System aims to enhance the efficiency, accuracy, and overall success of organizing and executing conferences."

3. DATABASE DESIGN:

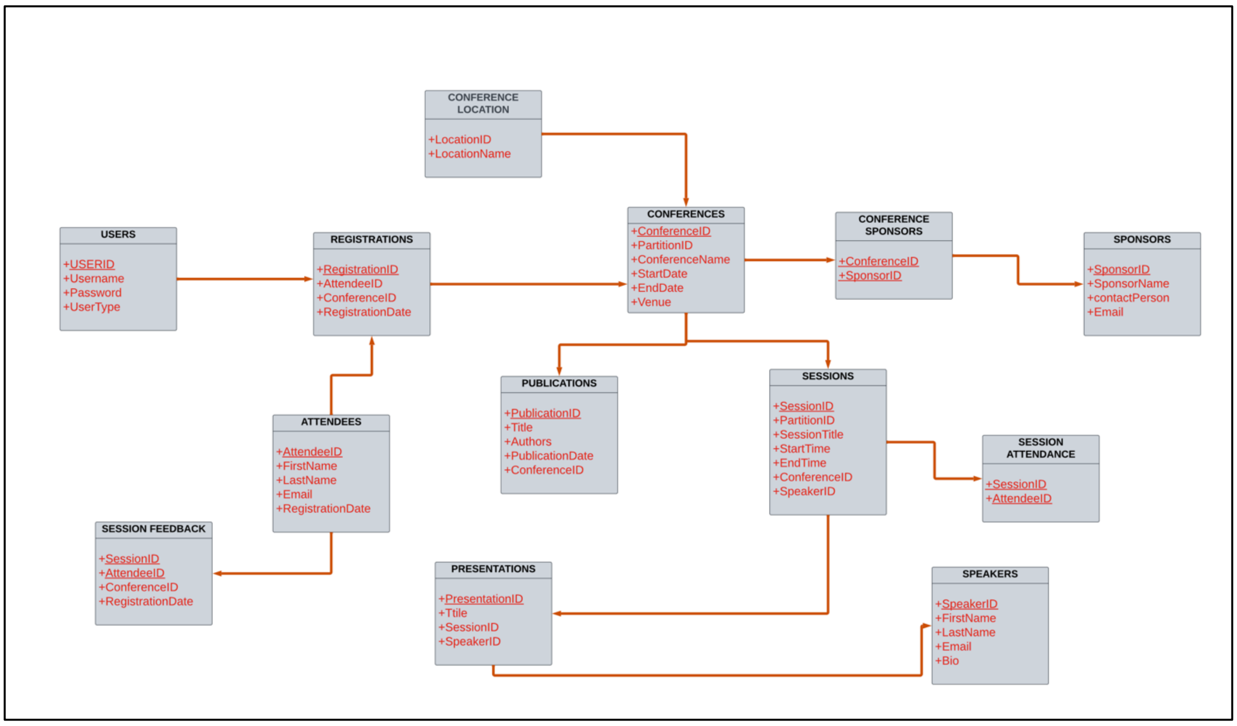


Fig-3.1 Conference Management UML Diagram

4. DATA COLLECTION:

1. Conferences Table:
    - Stores information about conferences.
    - Fields: ConferenceID, PartitionID, ConferenceName, StartDate, EndDate, Venue.
    - The ConferenceID is a unique identifier, and PartitionID may be used for partitioning.
    - Dates (StartDate and EndDate) indicate the duration of the conference.
2. Speakers Table:
    - Contains details about speakers participating in conferences.
    - Fields: SpeakerID, FirstName, LastName, Email, Bio.
    - SpeakerID serves as a unique identifier for each speaker.
3. Sessions Table:
    - Captures information about conference sessions.
    - Fields: SessionID, PartitionID, SessionTitle, StartTime, EndTime, ConferenceID, SpeakerID.
    - SessionID uniquely identifies each session.
    - ConferenceID and SpeakerID are foreign keys linking to the respective tables.
4. Attendees Table:
    - Stores data about attendees who register for conferences.
    - Fields: AttendeeID, FirstName, LastName, Email, RegistrationDate.
    - AttendeeID uniquely identifies each attendee.
5. SessionAttendance Table:
    - Tracks the attendance of attendees in sessions.
    - Fields: SessionID, AttendeeID.
    - SessionID and AttendeeID are foreign keys linked to the Sessions and Attendees tables.
6. Sponsors Table:
    - Contains information about sponsors supporting conferences.
    - Fields: SponsorID, SponsorName, ContactPerson, Email.
    - SponsorID uniquely identifies each sponsor.
7. ConferenceSponsors Table:
    - Associates sponsors with conferences.
    - Fields: ConferenceID, SponsorID.
    - Foreign keys link to the Conferences and Sponsors tables.

8. Publications Table:
    - Stores details about publications related to conferences.
    - Fields: PublicationID, Title, Authors, PublicationDate, ConferenceID.
    - PublicationID serves as a unique identifier, and ConferenceID is a foreign key.
9. Presentations Table:
    - Records information about presentations in sessions.
    - Fields: PresentationID, Title, SessionID, SpeakerID.
    - SessionID and SpeakerID are foreign keys associated with the Sessions and Speakers tables.
10. Users Table:
    - Stores information about system users (e.g., organizers, admins).
    - Fields: UserID, Username, Password, UserType.
    - UserID uniquely identifies each user.
11. Registrations Table:
    - Tracks registrations of attendees for conferences.
    - Fields: RegistrationID, AttendeeID, ConferenceID, RegistrationDate.
    - Foreign keys link to the Attendees and Conferences tables.
12. SessionFeedback Table:
    - Records feedback from attendees about sessions.
    - Fields: SessionID, AttendeeID, Rating.
    - SessionID and AttendeeID are foreign keys.
13. ConferenceLocations Table:
    - Contains information about conference rooms or locations.
    - Fields: LocationID, LocationName.
    - LocationID is a unique identifier.
14. ConferenceLocationMapping Table:
    - Associates locations with conferences.
    - Fields: ConferenceID, LocationID.
    - Foreign keys link to the Conferences and ConferenceLocations tables.

Table Name	Fields/Columns	Notes
Conferences	ConferenceID, PartitionID, ConferenceName, StartDate, EndDate, Venue	Unique identifier for conferences, partitioning support.
Speakers	SpeakerID, FirstName, LastName, Email, Bio	Details about speakers participating in conferences.
Sessions	SessionID, PartitionID, SessionTitle, StartTime, EndTime, ConferenceID, SpeakerID	Information about conference sessions.
Attendees	AttendeeID, FirstName, LastName, Email, RegistrationDate	Data about attendees who register for conferences.
SessionAttendance	SessionID, AttendeeID	Tracks attendance of attendees in sessions.
Sponsors	SponsorID, SponsorName, ContactPerson, Email	Information about sponsors supporting conferences.
ConferenceSponsors	ConferenceID, SponsorID	Associates sponsors with conferences.
Publications	PublicationID, Title, Authors, PublicationDate, ConferenceID	Details about publications related to conferences.
Presentations	PresentationID, Title, SessionID, SpeakerID	Records information about presentations in sessions.
Users	UserID, Username, Password, UserType	Information about system users (organizers, admins).
Registrations	RegistrationID, AttendeeID, ConferenceID, RegistrationDate	Tracks registrations of attendees for conferences.
SessionFeedback	SessionID, AttendeeID, Rating	Records feedback from attendees about sessions.
ConferenceLocations	LocationID, LocationName	Information about conference rooms or locations.
ConferenceLocationMapping	ConferenceID, LocationID	Associates locations with conferences.

Table 4.1.1 Conference Management Tables

4.1. Sequence of Actions

- Conference Organization:
  Organizers create a new conference by entering details such as the conference name, start and end dates, venue, etc. This information is stored in the Conferences table.
- Speaker Management:
  Organizers add information about speakers by creating records in the Speakers table. Each speaker has a unique identifier (SpeakerID), along with details like their name, email, and bio.
- Session Planning:
  Organizers plan conference sessions by creating records in the Sessions table. They link sessions to specific conferences and speakers. The Sessions table includes information about session title, start time, end time, and references to the associated conference and speaker.
- Publication Submission:
  Authors submit publications by providing details such as title, authors, publication date, and the associated conference. This information is stored in the Publications table.
- Presentation Submission:
  Speakers submit presentation proposals by providing details like title, session, and speaker information. This data is stored in the Presentations table.
- User Management:
  Administrators manage user accounts by creating records in the Users table. User types can include organizers and administrators. Usernames and passwords are stored securely.
- Attendee Registration:
  Attendees register for conferences by creating records in the Attendees table. Registration details, such as the attendee's name, email, and registration date, are stored.
- Session Attendance:
  Attendees attend specific sessions by creating records in the SessionAttendance table. This records which attendees participated in which sessions.
- Feedback Submission:
  Attendees provide feedback on sessions by creating records in the SessionFeedback table. This includes details such as the session ID, attendee ID, and a rating.

- Sponsorship Management:
  Organizers manage conference sponsors by creating records in the Sponsors table. They associate sponsors with conferences using the ConferenceSponsors table.
- Conference Location Management:
  Organizers manage conference locations by creating records in the ConferenceLocations table. They associate locations with conferences using the ConferenceLocationMapping table.
- Review and Analysis:
  Organizers and administrators can review data, analyze feedback, and make informed decisions for future conferences based on the information stored in the database.

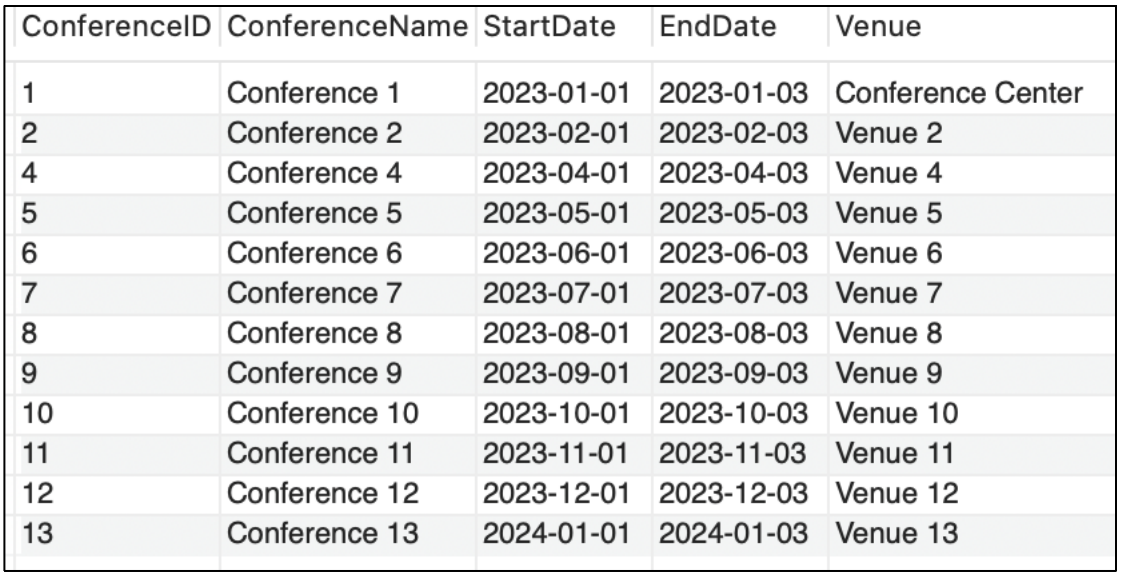
4.2. SAMPLE-DATA COLLECTION QUERIES:

1. Retrieve Conference Information:

Query:

```
SELECT ConferenceName, StartDate, EndDate, Venue
FROM Conferences;
```

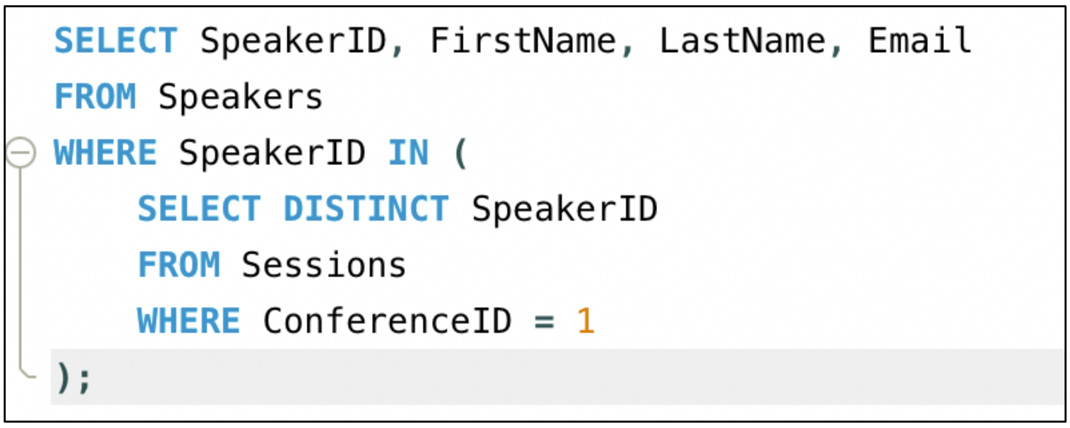
Output:



ConferenceID	ConferenceName	StartDate	EndDate	Venue
1	Conference 1	2023-01-01	2023-01-03	Conference Center
2	Conference 2	2023-02-01	2023-02-03	Venue 2
4	Conference 4	2023-04-01	2023-04-03	Venue 4
5	Conference 5	2023-05-01	2023-05-03	Venue 5
6	Conference 6	2023-06-01	2023-06-03	Venue 6
7	Conference 7	2023-07-01	2023-07-03	Venue 7
8	Conference 8	2023-08-01	2023-08-03	Venue 8
9	Conference 9	2023-09-01	2023-09-03	Venue 9
10	Conference 10	2023-10-01	2023-10-03	Venue 10
11	Conference 11	2023-11-01	2023-11-03	Venue 11
12	Conference 12	2023-12-01	2023-12-03	Venue 12
13	Conference 13	2024-01-01	2024-01-03	Venue 13

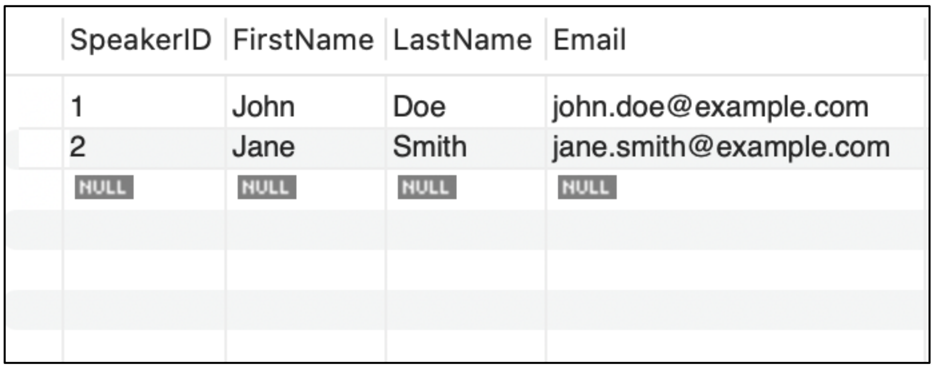
2. **List Speakers for a Specific Conference:**

Query:



```sql
SELECT SpeakerID, FirstName, LastName, Email
FROM Speakers
WHERE SpeakerID IN (
    SELECT DISTINCT SpeakerID
    FROM Sessions
    WHERE ConferenceID = 1
);
```

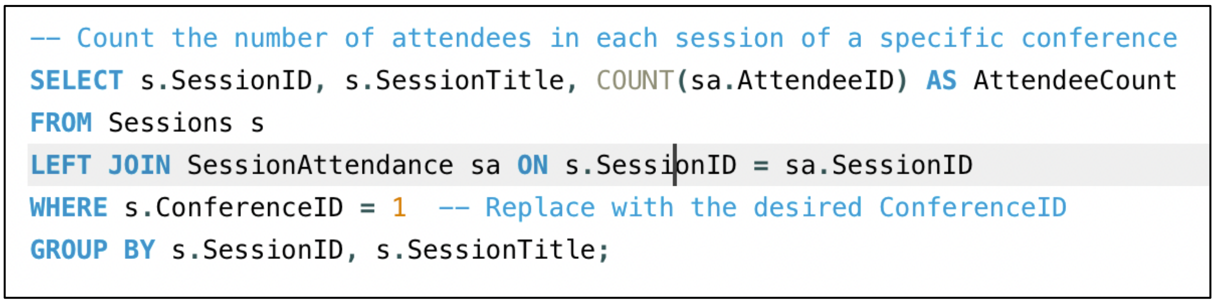
Output:



```
SpeakerID  FirstName  LastName  Email
1          John       Doe       john.doe@example.com
2          Jane       Smith     jane.smith@example.com
NULL       NULL       NULL      NULL
```

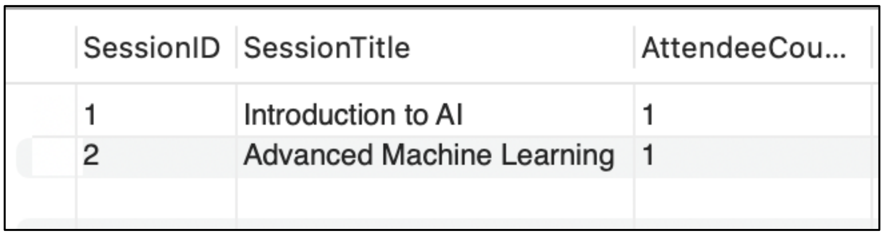
3. Count Attendees in Each Session:

Query:



```sql
-- Count the number of attendees in each session of a specific conference
SELECT s.SessionID, s.SessionTitle, COUNT(sa.AttendeeID) AS AttendeeCount
FROM Sessions s
LEFT JOIN SessionAttendance sa ON s.SessionID = sa.SessionID
WHERE s.ConferenceID = 1  -- Replace with the desired ConferenceID
GROUP BY s.SessionID, s.SessionTitle;
```

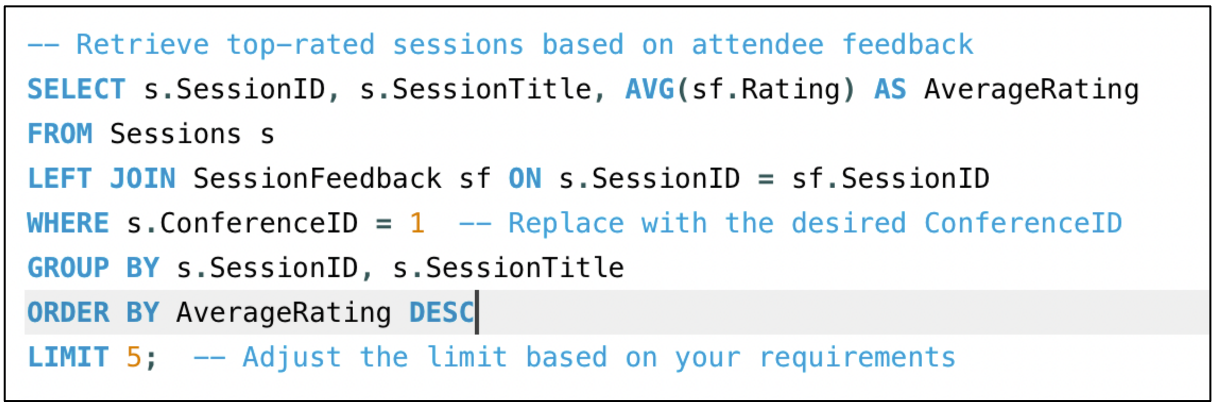
Output:



SessionID	SessionTitle	AttendeeCou...
1	Introduction to AI	1
2	Advanced Machine Learning	1

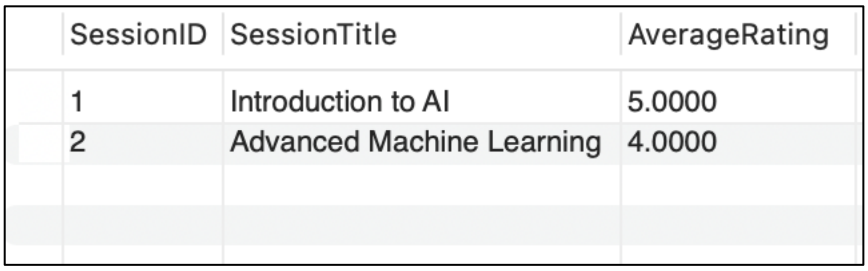
4. Top-Rated Sessions:

Query:



```
-- Retrieve top-rated sessions based on attendee feedback
SELECT s.SessionID, s.SessionTitle, AVG(sf.Rating) AS AverageRating
FROM Sessions s
LEFT JOIN SessionFeedback sf ON s.SessionID = sf.SessionID
WHERE s.ConferenceID = 1  -- Replace with the desired ConferenceID
GROUP BY s.SessionID, s.SessionTitle
ORDER BY AverageRating DESC
LIMIT 5;  -- Adjust the limit based on your requirements
```

Output:



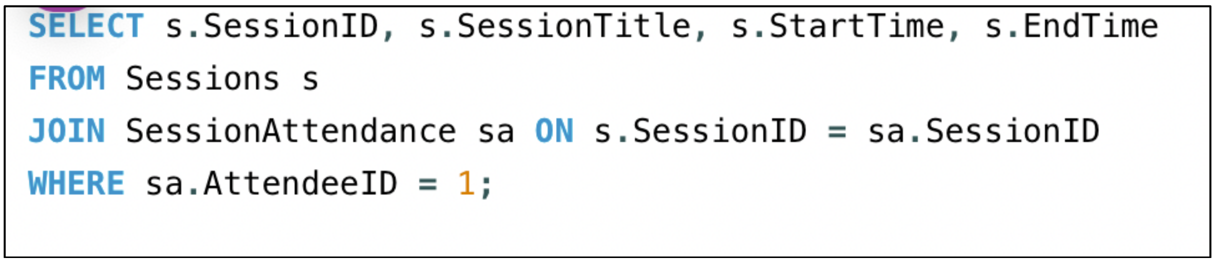
SessionID	SessionTitle	AverageRating
1	Introduction to AI	5.0000
2	Advanced Machine Learning	4.0000

5. SQL-QUERIES:

1. Show Sessions for an Attendee:

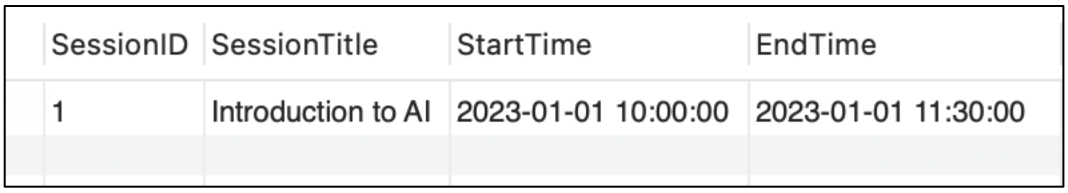
Helps attendees view the sessions they have registered for, aiding in managing their schedule and ensuring they don't miss sessions of interest.

Query:



```
SELECT s.SessionID, s.SessionTitle, s.StartTime, s.EndTime
FROM Sessions s
JOIN SessionAttendance sa ON s.SessionID = sa.SessionID
WHERE sa.AttendeeID = 1;
```

Output

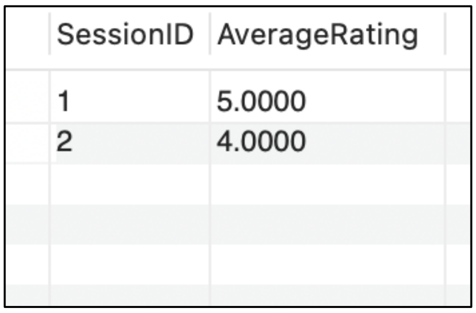


SessionID	SessionTitle	StartTime	EndTime
1	Introduction to AI	2023-01-01 10:00:00	2023-01-01 11:30:00

2. Conference Feedback Analysis:

Allows organizers to analyse attendee feedback on sessions, helping to identify popular sessions and areas for improvement in future conferences.

Query:



SessionID	AverageRating
1	5.0000
2	4.0000

Output:

```
SELECT SessionID, AVG(Rating) AS AverageRating
FROM SessionFeedback
GROUP BY SessionID
ORDER BY AverageRating DESC;
```

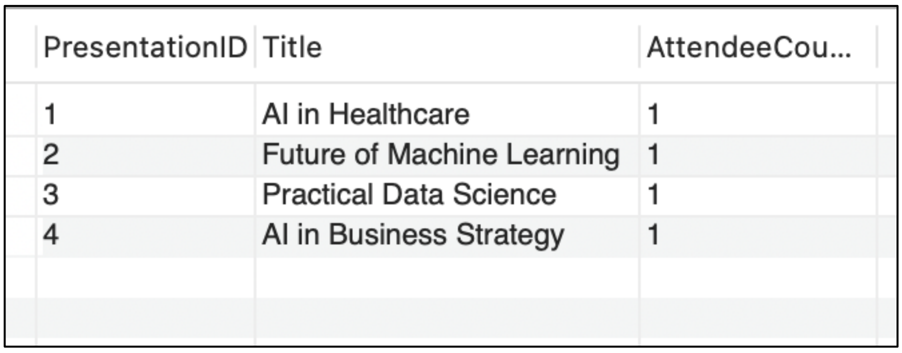
3. Identify Popular Presentations:

Provides insights into the popularity of individual presentations, assisting organizers in planning future events and allocating resources effectively.

Query:

```
SELECT PresentationID, Title, COUNT(*) AS AttendeeCount
FROM SessionAttendance
JOIN Presentations ON SessionAttendance.SessionID = Presentations.SessionID
GROUP BY PresentationID
ORDER BY AttendeeCount DESC;
```

Output:

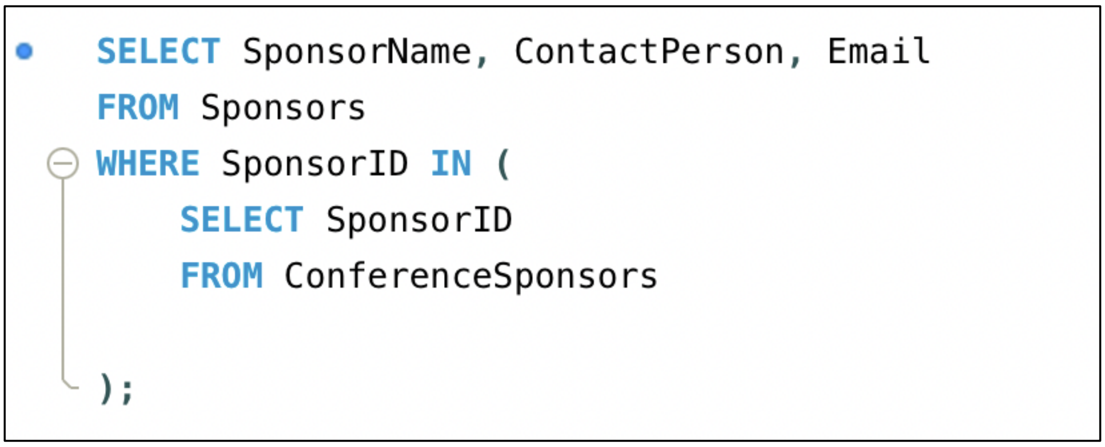


PresentationID	Title	AttendeeCou...
1	AI in Healthcare	1
2	Future of Machine Learning	1
3	Practical Data Science	1
4	AI in Business Strategy	1

4. List Sponsors for a Conference:

Useful for showcasing sponsors associated with a particular conference, which is crucial for acknowledging and promoting sponsors' contributions.

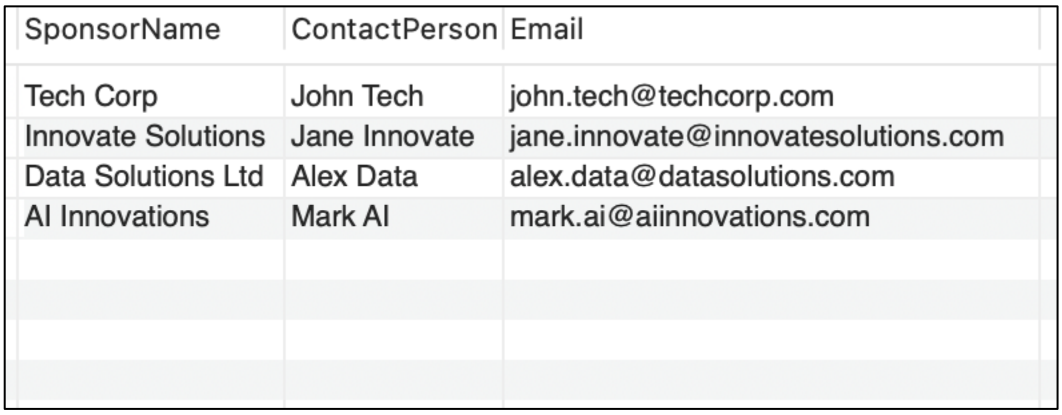
Query:



```
SELECT SponsorName, ContactPerson, Email
FROM Sponsors
WHERE SponsorID IN (
    SELECT SponsorID
    FROM ConferenceSponsors

);
```

Output:



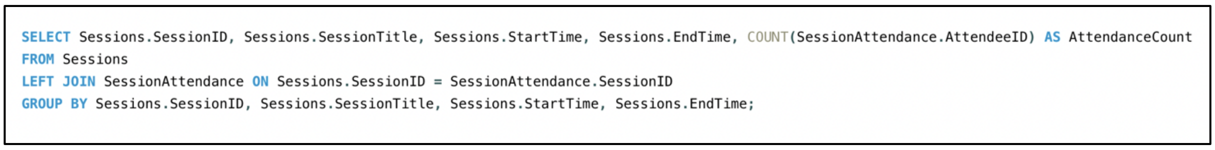
SponsorName	ContactPerson	Email
Tech Corp	John Tech	john.tech@techcorp.com
Innovate Solutions	Jane Innovate	jane.innovate@innovatesolutions.com
Data Solutions Ltd	Alex Data	alex.data@datasolutions.com
AI Innovations	Mark AI	mark.ai@aiinnovations.com

6. PERFORMANCE TUNING:

6.1. Optimization:

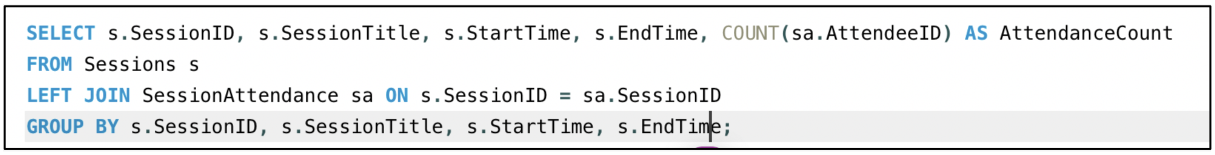
The optimized query uses shorter table aliases, making the query more readable, and it explicitly references columns with aliases to improve clarity.

Original Query:



```
SELECT Sessions.SessionID, Sessions.SessionTitle, Sessions.StartTime, Sessions.EndTime, COUNT(SessionAttendance.AttendeeID) AS AttendanceCount
FROM Sessions
LEFT JOIN SessionAttendance ON Sessions.SessionID = SessionAttendance.SessionID
GROUP BY Sessions.SessionID, Sessions.SessionTitle, Sessions.StartTime, Sessions.EndTime;
```

Optimized Query:



```
SELECT s.SessionID, s.SessionTitle, s.StartTime, s.EndTime, COUNT(sa.AttendeeID) AS AttendanceCount
FROM Sessions s
LEFT JOIN SessionAttendance sa ON s.SessionID = sa.SessionID
GROUP BY s.SessionID, s.SessionTitle, s.StartTime, s.EndTime;
```

6.2. Database indexing:

These indexes will help speed up queries that involve filtering, joining, or ordering based on the specified columns. Adjust the indexing strategy based on the specific queries and usage patterns in your conference management system.

-- Indexing on ConferenceID in Conferences table
CREATE INDEX idx_ConferenceID_Conferences ON Conferences(ConferenceID);

-- Indexing on SessionID in Sessions table
CREATE INDEX idx_SessionID_Sessions ON Sessions(SessionID);

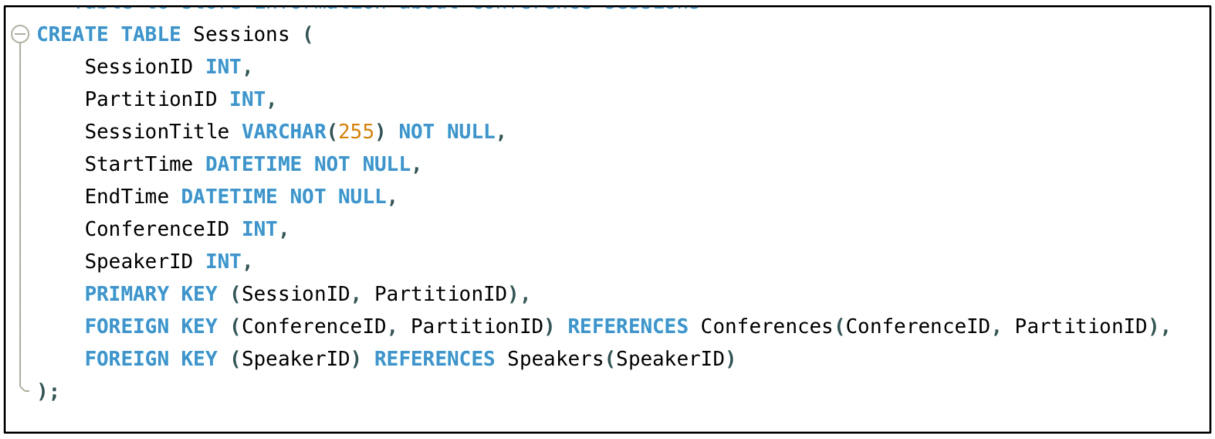
-- Indexing on AttendeeID in Attendees table
CREATE INDEX idx_AttendeeID_Attendees ON Attendees(AttendeeID);

-- Indexing on SpeakerID in Speakers table
CREATE INDEX idx_SpeakerID_Speakers ON Speakers(SpeakerID);

6.3. MANAGING LARGE AMOUT OF DATA: PARTITIONING

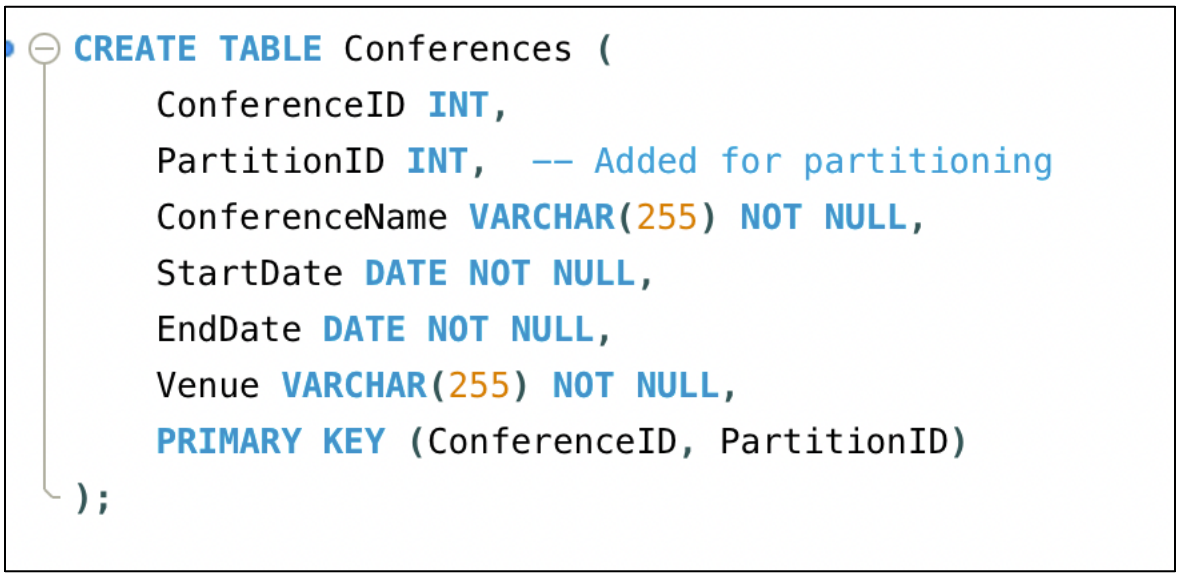
Horizontal partitioning based on the ConferenceID to distribute conferences across different partitions. This can be beneficial for scalability, especially if you expect a large number of conferences.

Session Table:



```
CREATE TABLE Sessions (
    SessionID INT,
    PartitionID INT,
    SessionTitle VARCHAR(255) NOT NULL,
    StartTime DATETIME NOT NULL,
    EndTime DATETIME NOT NULL,
    ConferenceID INT,
    SpeakerID INT,
    PRIMARY KEY (SessionID, PartitionID),
    FOREIGN KEY (ConferenceID, PartitionID) REFERENCES Conferences(ConferenceID, PartitionID),
    FOREIGN KEY (SpeakerID) REFERENCES Speakers(SpeakerID)
);
```

Conferences Table:

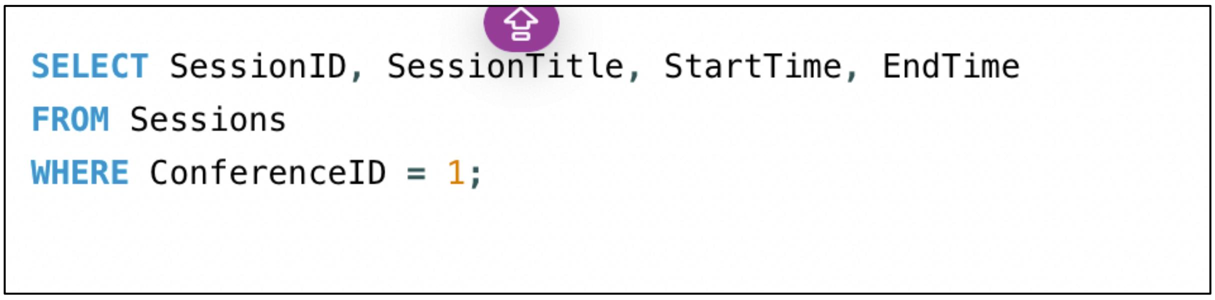


```
CREATE TABLE Conferences (
    ConferenceID INT,
    PartitionID INT,  -- Added for partitioning
    ConferenceName VARCHAR(255) NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    Venue VARCHAR(255) NOT NULL,
    PRIMARY KEY (ConferenceID, PartitionID)
);
```

7. QUESTIONS & SOLUTION:

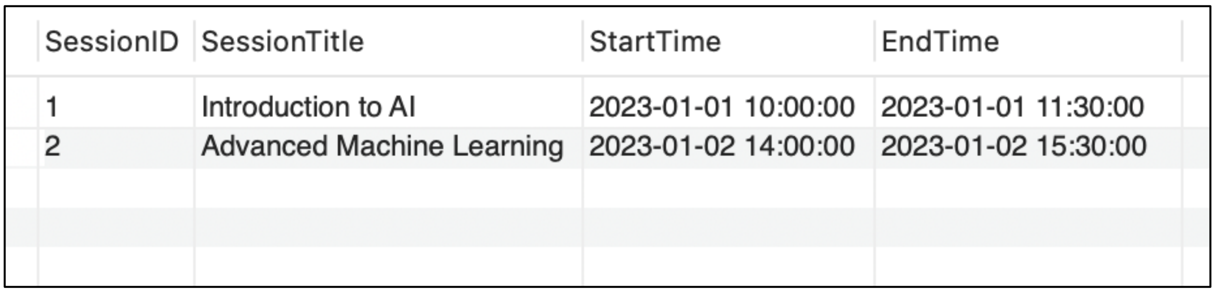
Question 1: Retrieve the list of sessions scheduled for a specific conference.

-- SQL Query



```sql
SELECT SessionID, SessionTitle, StartTime, EndTime
FROM Sessions
WHERE ConferenceID = 1;
```

OUTPUT:



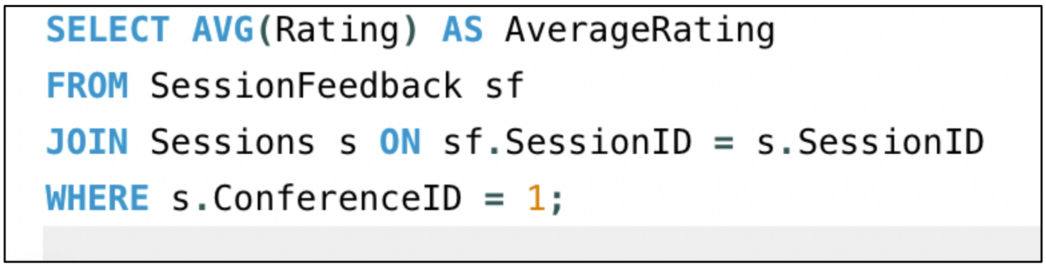
SessionID	SessionTitle	StartTime	EndTime
1	Introduction to AI	2023-01-01 10:00:00	2023-01-01 11:30:00
2	Advanced Machine Learning	2023-01-02 14:00:00	2023-01-02 15:30:00

QUERY-PLAN:

$$\begin{tikzcd}
\Pi \text{ (SessionID, SessionTitle, StartTime, EndTime)} \arrow[d, dash] \\
\bowtie \text{ (ConferenceID=1)} \arrow[d, dash] \\
\textbf{SESSIONS}
\end{tikzcd}$$

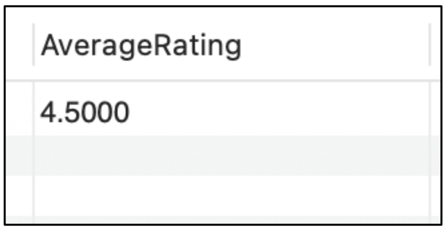
Question 2: Find the average rating of sessions for a given conference.

-- SQL Query



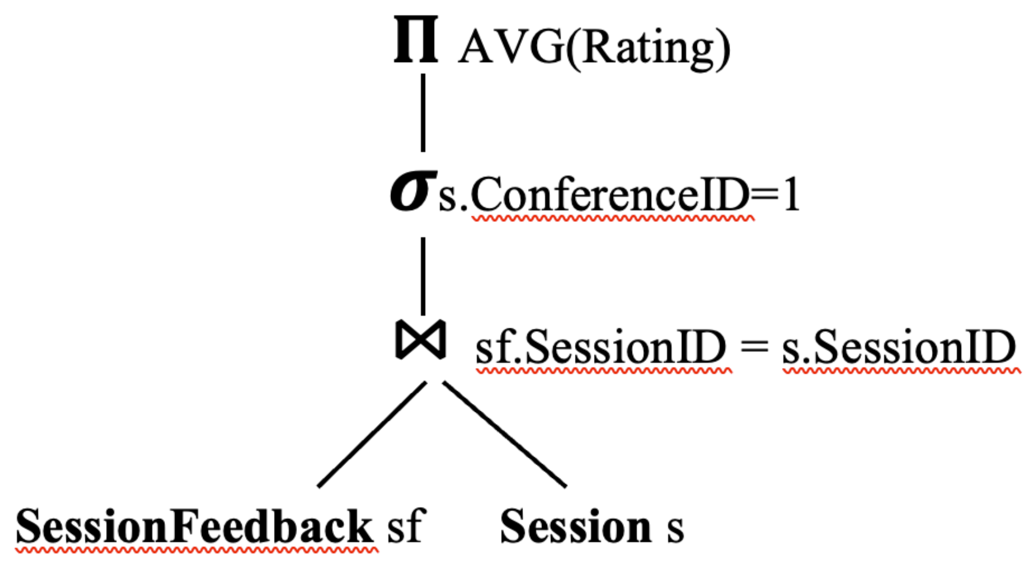
```
SELECT AVG(Rating) AS AverageRating
FROM SessionFeedback sf
JOIN Sessions s ON sf.SessionID = s.SessionID
WHERE s.ConferenceID = 1;
```

OUTPUT:



AverageRating
4.5000

QUERY-PLAN:

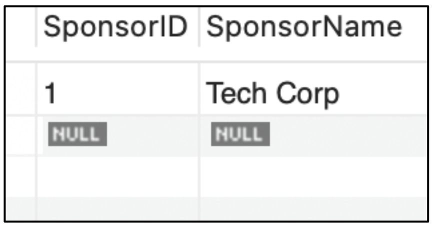


Question 3: List all sponsors for a specific conference.

-- SQL Query

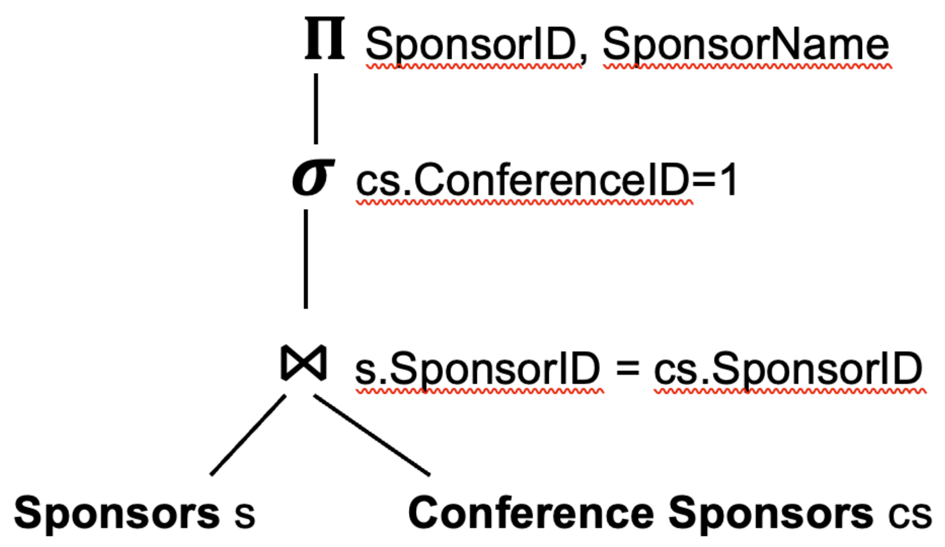
```
SELECT SponsorID, SponsorName
FROM Sponsors
WHERE SponsorID IN (SELECT SponsorID FROM ConferenceSponsors WHERE ConferenceID = 1);
```

OUTPUT:



SponsorID	SponsorName
1	Tech Corp
NULL	NULL

QUERY-PLAN:



8. LIMITATIONS:

8.1. Data Limitations:

- Sample Data Size: The provided sample data may not fully represent the scale and diversity of a real-world conference management system. The system's performance and behaviour with a larger dataset might differ.
- Real-time Data: The project lacks real-time data integration. In a live scenario, data such as attendee registrations, session feedback, and sponsorship details would be constantly updated.

8.2. Scope Constraints:

- Feature Limitations: The project focuses on core features such as session management, attendee registration, and sponsorship. Additional features, like advanced analytics, participant networking, or mobile app integration, are not explored in depth.
- Security Measures: The project assumes a basic level of security and does not delve into advanced security measures like encryption and secure user authentication, which are crucial for a production system.

8.3. Resource Constraints:

- Hardware and Infrastructure: The system's performance may vary based on the underlying hardware and infrastructure. A comprehensive performance analysis and optimization might be required for real-world deployment.
- User Scalability: The system's scalability to handle a large number of concurrent users has not been thoroughly tested in this project.