Event Management System

DR. NOUMAT KABBARA

Jihane Lagha, Abir Melhem, Jad Suleiman | Database | 09/12/2024

Event Management Table

ER Diagrams

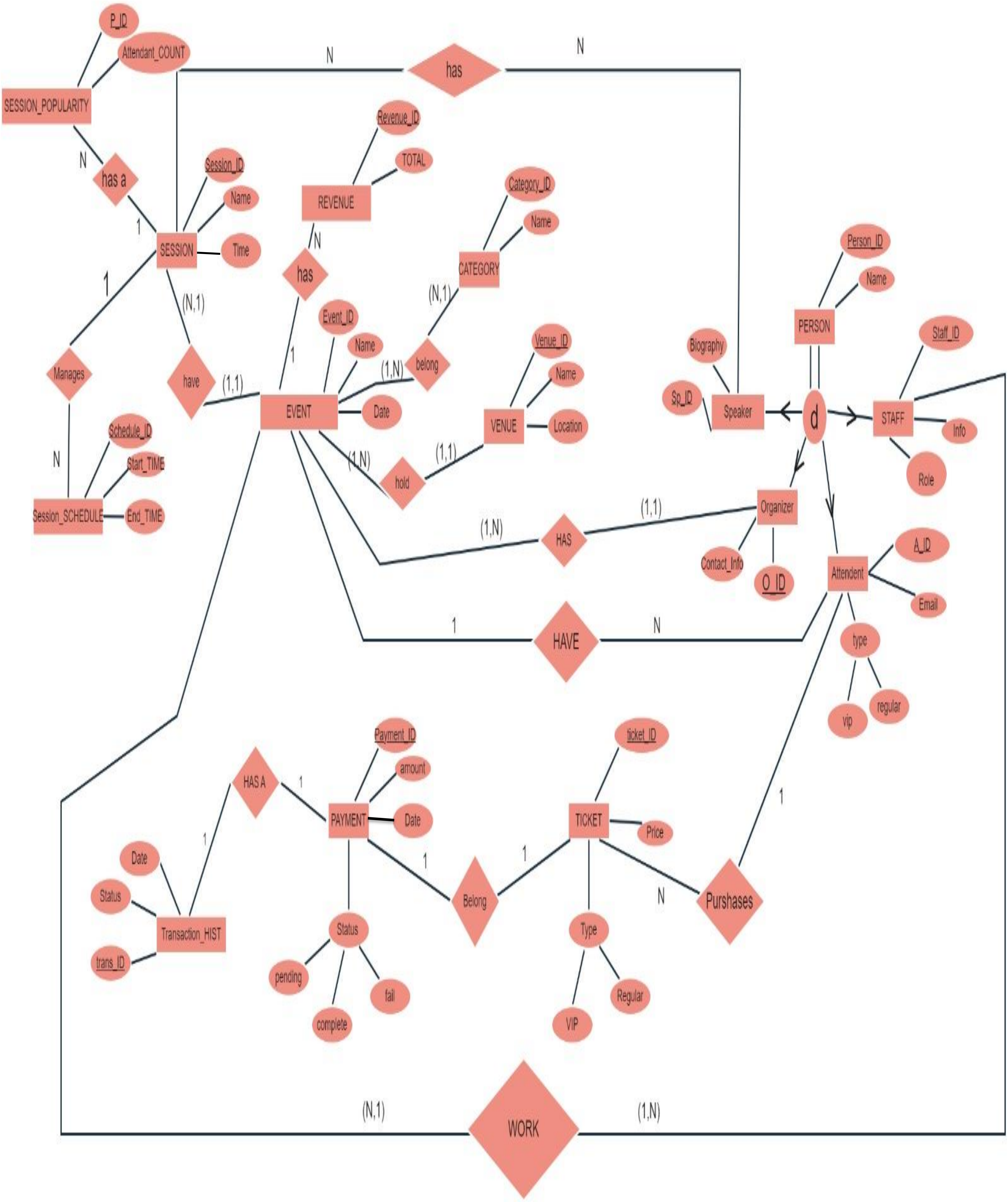
DDL – DML

Figures

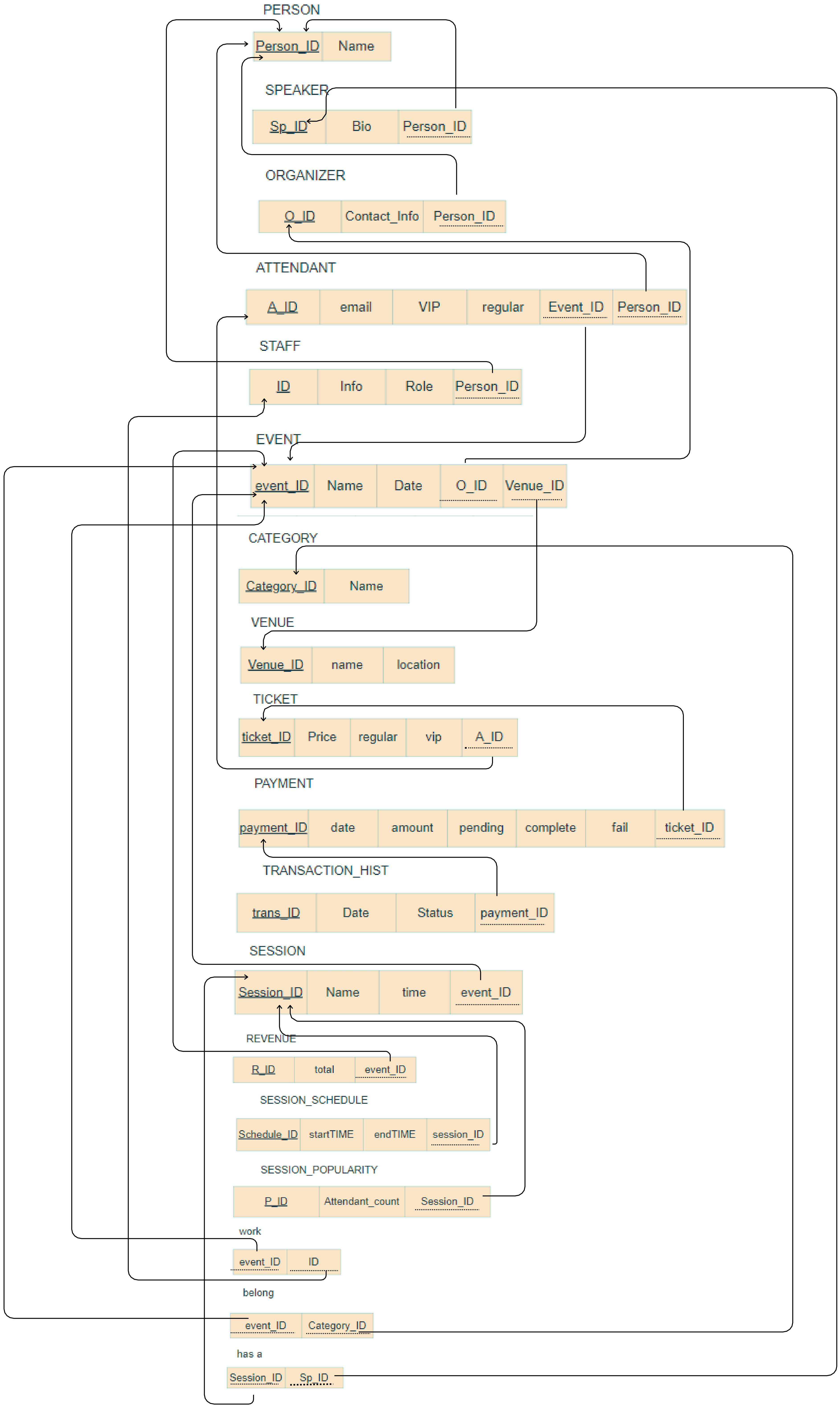
PROJECT SUMMARY REPORT

Name of Project	Event Management System
Brief Project Description	This project is a comprehensive platform designed to streamline the planning, execution, and tracking of events. It integrates key components such as event details, session scheduling, ticketing, attendee management, staff assignments, and revenue analytic into a centralized database. The system enables event organizers to manage event logistics, including assigning speakers to sessions, categorizing events, and tracking ticket sales and payments. Additionally, it provides insights into event performance by calculating total revenue, attendee counts, and session popularity. With features like customization ticket types, payment tracking, and session scheduling, this platform ensures that events run smoothly while offering valuable data to help organizers optimize their operations and enhance attendee experiences. The system also allows for scalability and future enhancements, such as feedback collection and dynamic pricing.
Database Overview	The purpose of this project was to design and implement a relational database system for events management. The database was designed to efficiently store information about speakers, staff, attendant, organizer, event, session, session schedule, ticket, payment, transactions history, venue, categories, revenue and session popularity. The database was implemented in Microsoft SQL Server using SQL queries. This report details all the technical aspects of the project, including the database design, implementation, and testing.
Database Design	The database was designed using entity-relationship modeling. The relationships between these entities were established, and foreign keys and constraints were added to enforce referential integrity. The design was optimized for efficient data retrieval and storage.
Database Implementation	In addition to the previously mentioned database design, we have implemented total disjoint specialization with cardinality. Total disjoint specialization means that each entity instance is a member of only one sub-entity set. We have implemented this by creating sub-entities of person such as speaker, attendant, staff and organizer. We have also implemented many-to-many relationships such as Session Speaker, which allows a session to have many speakers and a speaker can speak in many sessions. The cardinality of the relationships has been defined to ensure data consistency and integrity.

EER:


P_ID
Attendant_COUNT
SESSION_POPULARITY
N
has
N
has a
N
1
Revenue_ID
TOTAL
REVENUE
Session_ID
Name
Time
SESSION
1
(N,1)
N
has
Category_ID
Name
CATEGORY
(N,1)
Person_ID
Name
PERSON
Staff_ID
Manages
have
(1,1)
1
Event_ID
Name
(1,N)
belong
Venue_ID
Name
Biography
Sp_ID
Speaker
d
STAFF
Info
Role
EVENT
Date
VENUE
Location
N
(1,N)
(1,1)
hold
Schedule_ID
Start_TIME
Session_SCHEDULE
End_TIME
Organizer
(1,1)
(1,N)
HAS
Contact_Info
O_ID
Attendent
A_ID
Email
1
HAVE
N
type
vip
regular
Payment_ID
amount
HAS A
1
PAYMENT
Date
ticket_ID
TICKET
Price
1
Date
1
Status
Transaction_HIST
trans_ID
1
Belong
1
N
Purshases
Status
pending
complete
fail
Type
VIP
Regular
(N,1)
WORK
(1,N)

Relational Schema:



PERSON

Person_ID	Name

SPEAKER

Sp_ID	Bio	Person_ID

ORGANIZER

O_ID	Contact_Info	Person_ID

ATTENDANT

A_ID	email	VIP	regular	Event_ID	Person_ID

STAFF

ID	Info	Role	Person_ID

EVENT

event_ID	Name	Date	O_ID	Venue_ID

CATEGORY

Category_ID	Name

VENUE

Venue_ID	name	location

TICKET

ticket_ID	Price	regular	vip	A_ID

PAYMENT

payment_ID	date	amount	pending	complete	fail	ticket_ID

TRANSACTION_HIST

trans_ID	Date	Status	payment_ID

SESSION

Session_ID	Name	time	event_ID

REVENUE

R_ID	total	event_ID

SESSION_SCHEDULE

Schedule_ID	startTIME	endTIME	session_ID

SESSION_POPULARITY

P_ID	Attendant_count	Session_ID

work

event_ID	ID

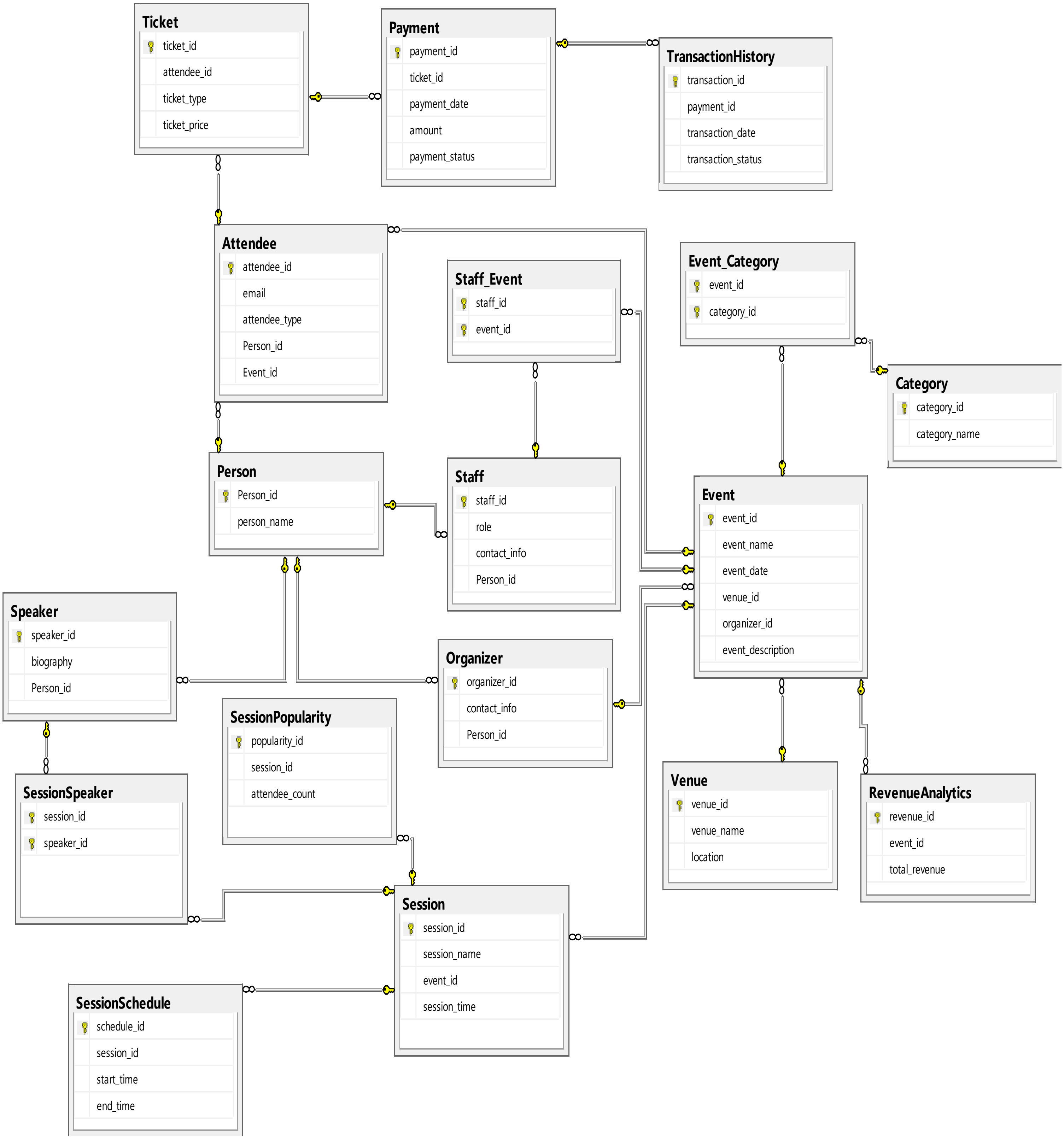
belong

event_ID	Category_ID

has a

Session_ID	Sp_ID

Auto-Generated ER Diagram:



DDL – DML

Database Creation Queries:

```
CREATE DATABASE Event_Management;

CREATE TABLE Venue (
   venue_id INT IDENTITY(1,1) PRIMARY KEY,
   venue_name VARCHAR(100),
   location VARCHAR(150)
);

CREATE TABLE Person (
  Person_id INT IDENTITY(1,1) PRIMARY KEY,
  person_name VARCHAR(100)
);

CREATE TABLE Organizer (
   organizer_id INT IDENTITY(1,1) PRIMARY KEY,
   contact_info VARCHAR(100),
   Person_id INT,
   FOREIGN KEY (Person_id) REFERENCES Person (Person_id),
);

CREATE TABLE Attendee (
   attendee_id INT IDENTITY(1,1) PRIMARY KEY,
   email VARCHAR(100) UNIQUE,
   attendee_type VARCHAR(10),
   Person_id INT ,
   Event_id INT ,
   FOREIGN KEY (event_id) REFERENCES Event (event_id),
   FOREIGN KEY (Person_id) REFERENCES Person (Person_id),
);

CREATE TABLE Staff (
   staff_id INT IDENTITY(1,1) PRIMARY KEY,
   role VARCHAR(50),
   contact_info VARCHAR (100),
   Person_id INT ,
   FOREIGN KEY (Person_id) REFERENCES Person (Person_id),
);

CREATE TABLE Speaker (
   speaker_id INT IDENTITY(1,1) PRIMARY KEY,
   biography TEXT,
   Person_id INT ,
   FOREIGN KEY (Person_id) REFERENCES Person (Person_id),
);


CREATE TABLE Event (
   event_id INT IDENTITY(1,1) PRIMARY KEY,
   event_name VARCHAR(100),
   event_date DATE,
   venue_id INT,
   organizer_id INT,
   FOREIGN KEY (venue_id) REFERENCES Venue(venue_id),
   FOREIGN KEY (organizer_id) REFERENCES Organizer(organizer_id)
);

CREATE TABLE Ticket (
   ticket_id INT IDENTITY(1,1) PRIMARY KEY,
   attendee_id INT,
   ticket_type VARCHAR(10),
   ticket_price DECIMAL(10, 2),
   FOREIGN KEY (attendee_id) REFERENCES Attendee (attendee_id)
);

CREATE TABLE Session (
   session_id INT IDENTITY(1,1) PRIMARY KEY,
   session_name VARCHAR(100),
   event_id INT,
   session_time TIMESTAMP,
   FOREIGN KEY (event_id) REFERENCES Event (event_id)
);

CREATE TABLE Staff_Event (
   staff_id INT,
   event_id INT,
   PRIMARY KEY (staff_id, event_id),
   FOREIGN KEY (staff_id) REFERENCES Staff (staff_id),
   FOREIGN KEY (event_id) REFERENCES Event (event_id)
);

CREATE TABLE Category (
   category_id INT IDENTITY(1,1) PRIMARY KEY,
   category_name VARCHAR(50)
);

CREATE TABLE Event_Category (
   event_id INT,
   category_id INT,
   PRIMARY KEY (event_id, category_id),
   FOREIGN KEY (event_id) REFERENCES Event (event_id),
   FOREIGN KEY (category_id) REFERENCES Category (category_id)
);
```

```
CREATE TABLE SessionSchedule (
    schedule_id INT IDENTITY(1,1) PRIMARY KEY,
    session_id INT,
    start_time DATETIME2,
    end_time DATETIME2,
    FOREIGN KEY (session_id) REFERENCES Session (session_id)
);


CREATE TABLE SessionSpeaker (
    session_id INT,
    speaker_id INT,
    PRIMARY KEY (session_id, speaker_id),
    FOREIGN KEY (session_id) REFERENCES Session (session_id),
    FOREIGN KEY (speaker_id) REFERENCES Speaker (speaker_id)
);

CREATE TABLE Payment (
    payment_id INT IDENTITY(1,1) PRIMARY KEY,
    ticket_id INT,
    payment_date DATETIME2 DEFAULT CURRENT_TIMESTAMP,
    amount DECIMAL(10, 2),
    payment_status VARCHAR(15),
    FOREIGN KEY (ticket_id) REFERENCES Ticket (ticket_id)
);

CREATE TABLE TransactionHistory (
    transaction_id INT IDENTITY(1,1) PRIMARY KEY,
    payment_id INT,
    transaction_date DATETIME2 DEFAULT CURRENT_TIMESTAMP,
    transaction_status VARCHAR(20),
    FOREIGN KEY (payment_id) REFERENCES Payment (payment_id)
);

CREATE TABLE RevenueAnalytics (
    revenue_id INT IDENTITY(1,1) PRIMARY KEY,
    event_id INT,
    total_revenue DECIMAL(10, 2),
    FOREIGN KEY (event_id) REFERENCES Event (event_id)
);

CREATE TABLE SessionPopularity (
    popularity_id INT IDENTITY(1,1) PRIMARY KEY,
    session_id INT,
    attendee_count INT,
    FOREIGN KEY (session_id) REFERENCES Session (session_id)
);
```

Database Sample Data Insertion:

```
INSERT INTO Venue (venue_name, location)
VALUES
   ('Tech Conference Center', 'New York, NY'),
   ('Global Expo Hall', 'Los Angeles, CA'),
   ('Innovation Arena', 'San Francisco, CA');

INSERT INTO Person (person_name)
VALUES
   ('John Doe'),
   ('Jane Smith'),
   ('Alice Johnson'),
   ('Bob Brown');

INSERT INTO Organizer (contact_info, Person_id)
VALUES
   ('john.doe@email.com', 1), -- Assuming John Doe has Person_id 1
   ('jane.smith@email.com', 2); -- Assuming Jane Smith has Person_id 2

INSERT INTO Attendee (email, attendee_type, Person_id,event_id)
VALUES
   ('attendee1@email.com', 'VIP', 3, 1), -- Alice Johnson, VIP, Tech Conference 2024
   ('attendee2@email.com', 'Regular', 4, 2); -- Bob Brown, Regular, Global Expo 2024

INSERT INTO Staff (role, contact_info, Person_id)
VALUES
   ('Security', 'security@email.com', 1), -- John Doe
   ('Technician', 'tech@email.com', 2); -- Jane Smith

INSERT INTO Speaker (biography, Person_id)
VALUES
   ('Tech enthusiast with 20 years of experience.', 3), -- Alice Johnson
   ('Cybersecurity expert with a passion for innovation.', 4); -- Bob Brown

INSERT INTO Event (event_name, event_date, venue_id, organizer_id)
VALUES
   ('Tech Conference 2024', '2024-06-15', 1, 1), -- Venue 1, Organizer 1
   ('Global Expo 2024', '2024-08-20', 2, 2); -- Venue 2, Organizer 2

INSERT INTO Ticket (attendee_id, ticket_type, ticket_price)
VALUES
   (2, 'VIP', 100.00), -- For the attendee with attendee_id = 2
   (3, 'Regular', 50.00); -- For the attendee with attendee_id = 3

ALTER TABLE Session DROP COLUMN session_time;

ALTER TABLE Session ADD session_time DATETIME2;
```

```
INSERT INTO Session (session_name, event_id, session_time)
VALUES
   ('Keynote: Future of Tech', 1, '2024-06-15 09:00:00'), -- Tech Conference 2024
   ('Innovation in AI', 2, '2024-08-20 10:00:00'); -- Global Expo 2024

INSERT INTO SessionSchedule (session_id, start_time, end_time)
VALUES
   (1, '2024-06-15 09:00:00', '2024-06-15 10:00:00'),  -- Keynote session
   (2, '2024-08-20 10:00:00', '2024-08-20 11:00:00');  -- AI session

INSERT INTO SessionSpeaker (session_id, speaker_id)
VALUES
   (1, 1), -- Keynote session, Alice Johnson
   (2, 2); -- Innovation in AI session, Bob Brown

INSERT INTO Payment (ticket_id, amount, payment_status)
VALUES
   (6, 150.00, 'Completed'), -- VIP ticket, payment completed
   (7, 50.00, 'Pending'); -- Regular ticket, payment pending

INSERT INTO TransactionHistory (payment_id, transaction_status)
VALUES
   (1, 'Success'), -- Payment ID 1, successful transaction
   (2, 'Failed'); -- Payment ID 2, failed transaction

INSERT INTO RevenueAnalytics (event_id, total_revenue)
VALUES
   (1, 150.00), -- Tech Conference 2024, total revenue from tickets
   (2, 50.00); -- Global Expo 2024, total revenue from tickets

INSERT INTO SessionPopularity (session_id, attendee_count)
VALUES
   (1, 100), -- Keynote session has 100 attendees
   (2, 50); -- Innovation in AI session has 50 attendees

INSERT INTO Staff_Event (staff_id, event_id)
VALUES
    (1, 1), -- John Doe (staff_id 1) assigned to Tech Conference 2024 (event_id 1)
    (2, 2); -- Jane Smith (staff_id 2) assigned to Global Expo 2024 (event_id 2)

INSERT INTO Category (category_name)
VALUES
    ('Technology'), -- Category for Tech Conference 2024
    ('Innovation'); -- Category for Global Expo 2024

INSERT INTO Event_Category (event_id, category_id)
VALUES
    (1, 1),--Tech Conference 2024(event_id 1) belongs to Technology(category_id 1)
    (2, 2); -- Global Expo 2024 (event_id 2) belongs to Innovation (category_id 2)
```

Database Sample Data Update, Delete, Alter:

```
UPDATE Ticket
SET ticket_type = 'Regular'
WHERE ticket_id = 1;

ALTER TABLE Event
ADD event_description TEXT;

UPDATE Payment
SET payment_status = 'Completed'
WHERE payment_id = 1;

DELETE FROM TransactionHistory
WHERE payment_id IN (
          SELECT payment_id
          FROM Payment
          WHERE ticket_id = 1);
```

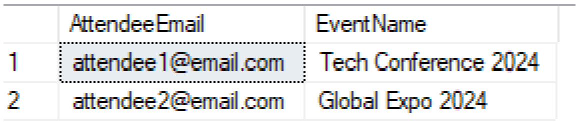
Database Questions to Answer:

1.What are the email addresses of attendees and the corresponding events they attended?

```
SELECT A.email, E.event_name
FROM Attendee A
INNER JOIN Event E ON A.event_id = E.event_id;
```

2.What are the email addresses of attendees and the events they attended, with the email labeled as 'AttendeeEmail' and the event name as 'EventName'?

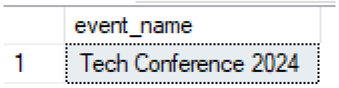
```
SELECT A.email AS AttendeeEmail, E.event_name AS EventName
FROM Attendee A
INNER JOIN Event E ON A.event_id = E.event_id;
```



	AttendeeEmail	EventName
1	attendee1@email.com	Tech Conference 2024
2	attendee2@email.com	Global Expo 2024

3.What are the names of events that include 'Tech' in their title?

```
SELECT event_name
FROM Event
WHERE LOWER(event_name) LIKE '%tech%';
```



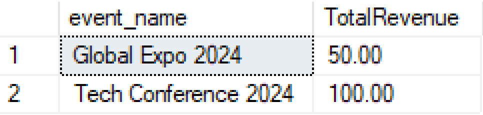
	event_name
1	Tech Conference 2024

4.What are the unique email addresses of all attendees?

```
SELECT DISTINCT email

FROM Attendee;
```

5.What is the total revenue for each event based on ticket sales?

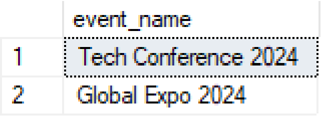
```
SELECT E.event_name, SUM(T.ticket_price) AS TotalRevenue
FROM Event E
JOIN Attendee A ON E.event_id = A.event_id
JOIN Ticket T ON A.attendee_id = T.attendee_id
GROUP BY E.event_name;
```



	event_name	TotalRevenue
1	Global Expo 2024	50.00
2	Tech Conference 2024	100.00

6.What are the names of events that do not have a description?

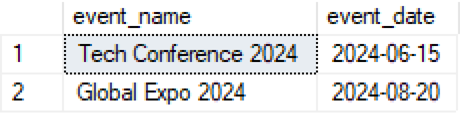
```
SELECT event_name

FROM Event

WHERE event_description IS NULL;
```



	event_name
1	Tech Conference 2024
2	Global Expo 2024

7.What are the names and dates of events, sorted by date in ascending order?

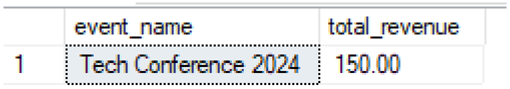
```
SELECT event_name, event_date

FROM Event

ORDER BY event_date ASC;
```



	event_name	event_date
1	Tech Conference 2024	2024-06-15
2	Global Expo 2024	2024-08-20

8.What are the names of the events and their corresponding total revenues for events that have more than 50 attendees in any of their sessions?

```
SELECT E.event_name, RA.total_revenue

FROM RevenueAnalytics RA

JOIN Event E ON RA.event_id = E.event_id

WHERE RA.event_id IN (

    SELECT SE.event_id

    FROM SessionPopularity SP

    JOIN Session SE ON SP.session_id = SE.session_id

    WHERE SP.attendee_count > 50);
```



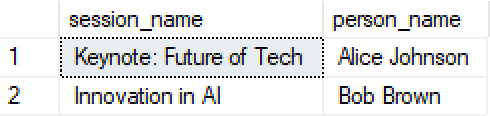
	event_name	total_revenue
1	Tech Conference 2024	150.00

9.What are the names of the events and their total revenues for events that have a total revenue greater than 100?

```
SELECT E.event_name, RA.total_revenue
FROM RevenueAnalytics RA
JOIN Event E ON RA.event_id = E.event_id
WHERE RA.total_revenue > 100 AND RA.total_revenue IS NOT NULL;
```

10.What are the names of the sessions and the corresponding speaker names for sessions scheduled after June 15, 2024?

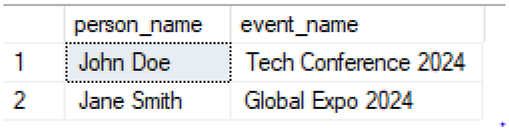
```
SELECT S.session_name, P.person_name
FROM SessionSpeaker SS
JOIN Session S ON SS.session_id = S.session_id
JOIN Speaker SP ON SS.speaker_id = SP.speaker_id
JOIN Person P ON SP.Person_id = P.Person_id
WHERE S.session_time > '2024-06-15 00:00:00';
```



	session_name	person_name
1	Keynote: Future of Tech	Alice Johnson
2	Innovation in AI	Bob Brown

11. Get All Staff Members and the Events They Are Assigned To

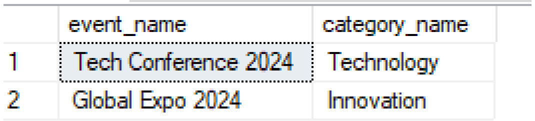
```
SELECT P.person_name, E.event_name
FROM Staff_Event SE
JOIN Staff S ON SE.staff_id = S.staff_id
JOIN Person P ON S.Person_id = P.Person_id
JOIN Event E ON SE.event_id = E.event_id;
```



	person_name	event_name
1	John Doe	Tech Conference 2024
2	Jane Smith	Global Expo 2024

12. Get Categories for Each Event

```
SELECT E.event_name, C.category_name
FROM Event E
JOIN Event_Category EC ON E.event_id = EC.event_id
JOIN Category C ON EC.category_id = C.category_id;
```



	event_name	category_name
1	Tech Conference 2024	Technology
2	Global Expo 2024	Innovation

13. Get Events that Belong to a Specific Category

```
SELECT E.event_name
FROM Event E
JOIN Event_Category EC ON E.event_id = EC.event_id
JOIN Category C ON EC.category_id = C.category_id
WHERE C.category_name = 'Technology';
```

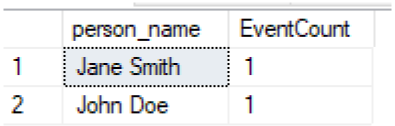
14. Get Events that Belong to ONE OR Multiple Categories

```
SELECT E.event_name
FROM Event E
```

```
JOIN Event_Category EC ON E.event_id = EC.event_id
GROUP BY E.event_name
HAVING COUNT(EC.category_id) >= 1;
```

15. Get Staff Assigned to ONE OR Multiple Events

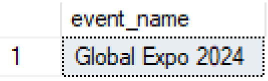
```
SELECT P.person_name, COUNT(SE.event_id) AS EventCount
FROM Staff_Event SE
JOIN Staff S ON SE.staff_id = S.staff_id
JOIN Person P ON S.Person_id = P.Person_id
GROUP BY P.person_name
HAVING COUNT(SE.event_id) >= 1;
```



	person_name	EventCount
1	Jane Smith	1
2	John Doe	1

16. Find Events that Have a Specific Category

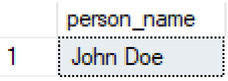
```
SELECT E.event_name
FROM Event E
JOIN Event_Category EC ON E.event_id = EC.event_id
JOIN Category C ON EC.category_id = C.category_id
WHERE C.category_name = 'Innovation';
```



	event_name
1	Global Expo 2024

17. Find Staff Assigned to the Tech Conference 2024

```
SELECT P.person_name
FROM Staff_Event SE
JOIN Staff S ON SE.staff_id = S.staff_id
JOIN Person P ON S.Person_id = P.Person_id
JOIN Event E ON SE.event_id = E.event_id
WHERE E.event_name = 'Tech Conference 2024';
```

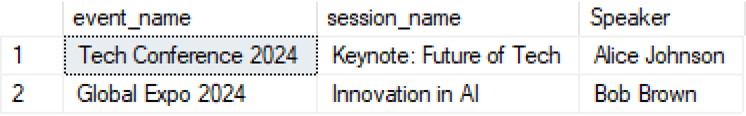


	person_name
1	John Doe

18.What are the event names, session names, and the names of the speakers for each session at the event?

```
SELECT E.event_name, S.session_name, P.person_name AS Speaker
```

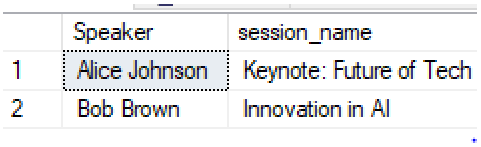
```
FROM Event E

LEFT JOIN Session S ON E.event_id = S.event_id

LEFT JOIN SessionSpeaker SS ON S.session_id = SS.session_id

LEFT JOIN Speaker SP ON SS.speaker_id = SP.speaker_id

LEFT JOIN Person P ON SP.Person_id = P.Person_id;
```



	event_name	session_name	Speaker
1	Tech Conference 2024	Keynote: Future of Tech	Alice Johnson
2	Global Expo 2024	Innovation in AI	Bob Brown

19.What are the names of the speakers and the session names they are associated with?

```
SELECT P.person_name AS Speaker, S.session_name
FROM Session S
RIGHT JOIN SessionSpeaker SS ON S.session_id = SS.session_id
RIGHT JOIN Speaker SP ON SS.speaker_id = SP.speaker_id
JOIN Person P ON SP.Person_id = P.Person_id;
```



	Speaker	session_name
1	Alice Johnson	Keynote: Future of Tech
2	Bob Brown	Innovation in AI

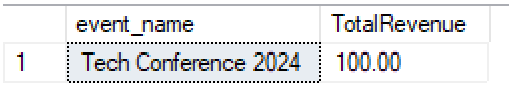
20.What are the session names and the names of the speakers, including sessions that may not have speakers and speakers who may not be assigned to sessions?

```
SELECT S.session_name, P.person_name AS Speaker
FROM Session S
FULL JOIN SessionSpeaker SS ON S.session_id = SS.session_id
FULL JOIN Speaker SP ON SS.speaker_id = SP.speaker_id
LEFT JOIN Person P ON SP.Person_id = P.Person_id;
```

21.What are the event names and the total revenue generated from ticket sales for each event, where the total revenue exceeds 50?

```
SELECT E.event_name, SUM(T.ticket_price) AS TotalRevenue
```

```
FROM Event E
JOIN Attendee A ON E.event_id = A.event_id
JOIN Ticket T ON A.attendee_id = T.attendee_id
GROUP BY E.event_name
HAVING SUM(T.ticket_price) > 50;
```



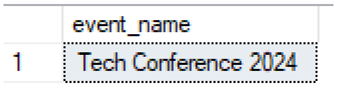
	event_name	TotalRevenue
1	Tech Conference 2024	100.00

22.What are the event names that belong to the categories 'Technology' or 'Innovation'?

```
SELECT E.event_name
FROM Event E
JOIN Event_Category EC ON E.event_id = EC.event_id
JOIN Category C ON EC.category_id = C.category_id
WHERE C.category_name IN ('Technology', 'Innovation');
```

23.What are the event names that have at least one VIP ticket purchased?

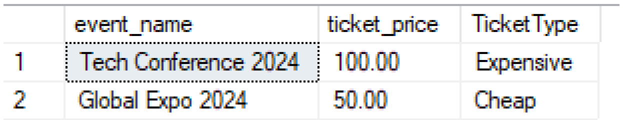
```
SELECT E.event_name
FROM Event E
WHERE EXISTS (
    SELECT 1
    FROM Ticket T
    JOIN Attendee A ON T.attendee_id = A.attendee_id
    WHERE T.ticket_type = 'VIP' AND A.event_id = E.event_id
);
```



	event_name
1	Tech Conference 2024

24.What are the event names, ticket prices, and the classification of each ticket as either 'Expensive' or 'Cheap' based on the ticket price?

```
SELECT E.event_name, T.ticket_price,
       CASE
          WHEN T.ticket_price >= 100 THEN 'Expensive'
          ELSE 'Cheap'
       END AS TicketType
FROM Ticket T
JOIN Attendee A ON T.attendee_id = A.attendee_id
JOIN Event E ON A.event_id = E.event_id;
```



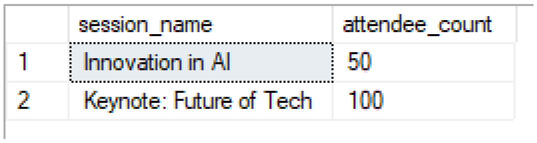
	event_name	ticket_price	TicketType
1	Tech Conference 2024	100.00	Expensive
2	Global Expo 2024	50.00	Cheap

25.What are the session names and attendee counts, including all sessions regardless of whether their attendee count is greater than or less than 50?

```
SELECT S.session_name, SP.attendee_count
FROM Session S
JOIN SessionPopularity SP ON S.session_id = SP.session_id;
```

OR

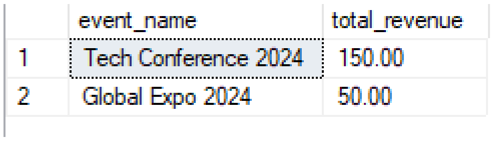
```
SELECT S.session_name, SP.attendee_count

FROM Session S

JOIN SessionPopularity SP ON S.session_id = SP.session_id

WHERE SP.attendee_count > 50

UNION

SELECT S.session_name, SP.attendee_count

FROM Session S

JOIN SessionPopularity SP ON S.session_id = SP.session_id

WHERE SP.attendee_count <= 50;
```



	session_name	attendee_count
1	Innovation in AI	50
2	Keynote: Future of Tech	100

26.What are the top 3 event names with the highest total revenue?

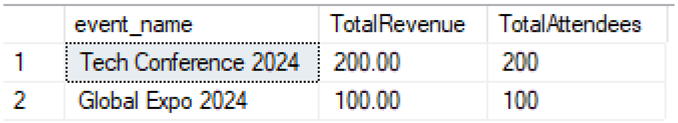
```
SELECT TOP 3 E.event_name, RA.total_revenue

FROM RevenueAnalytics RA

JOIN Event E ON RA.event_id = E.event_id

ORDER BY RA.total_revenue DESC;
```



	event_name	total_revenue
1	Tech Conference 2024	150.00
2	Global Expo 2024	50.00

27.What are the event names, total revenue from ticket sales, and total number of attendees for each event, ordered by total revenue in descending order?

```
SELECT E.event_name, SUM(T.ticket_price) AS TotalRevenue, SUM(SP.attendee_count) AS
TotalAttendees
FROM Event E
JOIN Attendee A ON E.event_id = A.event_id
JOIN Ticket T ON A.attendee_id = T.attendee_id
JOIN Session S ON E.event_id = S.event_id
JOIN SessionPopularity SP ON S.session_id = SP.session_id
GROUP BY E.event_name
ORDER BY TotalRevenue DESC;
```



	event_name	TotalRevenue	TotalAttendees
1	Tech Conference 2024	200.00	200
2	Global Expo 2024	100.00	100