

1. User Upcoming Events

```
SELECT full_name, title, city, start_date FROM Users, Events, Registrations
WHERE Users.user_id = Registrations.user_id
AND Events.event_id = Registrations.event_id
AND Events.status = 'upcoming'
AND Users.city = Events.city
ORDER BY start_date;
```

2. Top Rated Events

```
SELECT title, AVG(rating) AS avg_rating, COUNT(*) AS feedback_count
FROM Events, Feedback
WHERE Events.event_id = Feedback.event_id
GROUP BY Events.event_id
HAVING feedback_count >= 10
ORDER BY avg_rating DESC;
```

3. Inactive Users

```
SELECT full_name, email
FROM Users
WHERE user_id NOT IN (
    SELECT user_id FROM Registrations
    WHERE registration_date >= CURDATE() - INTERVAL 90 DAY
);
```

4. Peak Session Hours

```
SELECT title, COUNT(*) AS session_count
FROM Events, Sessions
WHERE Events.event_id = Sessions.event_id
AND TIME(start_time) >= '10:00:00'
AND TIME(start_time) < '12:00:00'
GROUP BY Events.event_id;
```

5. Most Active Cities

```
SELECT city, COUNT(DISTINCT user_id) AS total_users
FROM Users, Registrations
WHERE Users.user_id = Registrations.user_id
GROUP BY city
ORDER BY total_users DESC
LIMIT 5;
```

6. Event Resource Summary

```
SELECT title,
SUM(resource_type = 'pdf') AS pdfs,
SUM(resource_type = 'image') AS images,
SUM(resource_type = 'link') AS links
FROM Events, Resources
WHERE Events.event_id = Resources.event_id
GROUP BY Events.event_id;
```

7. Low Feedback Alerts

```
SELECT full_name, title, rating, comments
FROM Users, Events, Feedback
WHERE Users.user_id = Feedback.user_id
AND Events.event_id = Feedback.event_id
AND rating < 3;
```

8. Sessions per Upcoming Event

```
SELECT title, COUNT(*) AS session_count
FROM Events, Sessions
WHERE Events.event_id = Sessions.event_id
AND status = 'upcoming' GROUP BY Events.event_id;
```

9.Organizer Event Summary

```
SELECT full_name, status, COUNT(*) AS total_events
FROM Users, Events
WHERE Users.user_id = Events.organizer_id
GROUP BY Users.user_id, status;
```

10. Feedback Gap

```
SELECT DISTINCT title
FROM Events, Registrations
WHERE Events.event_id = Registrations.event_id
AND Events.event_id NOT IN (
    SELECT event_id FROM Feedback
);
```

11. Daily New User Count

```
SELECT registration_date, COUNT(*) AS new_users
FROM Users
WHERE registration_date >= CURDATE() - INTERVAL 7 DAY
GROUP BY registration_date
ORDER BY registration_date;
```

12. Event with Maximum Sessions

```
SELECT event_id, COUNT(*) AS session_count
FROM Sessions
GROUP BY event_id
HAVING session_count = (
    SELECT MAX(session_total)
    FROM (
        SELECT COUNT(*) AS session_total
        FROM Sessions
        GROUP BY event_id
    ) AS temp);
```

13. Average Rating per City

```
SELECT E.city, AVG(F.rating) AS avg_rating
FROM Feedback F
JOIN Events E ON F.event_id = E.event_id
GROUP BY E.city;
```

14. Most Registered Events

```
SELECT E.title, COUNT(*) AS total_registrations
FROM Registrations R
JOIN Events E ON R.event_id = E.event_id
GROUP BY R.event_id
ORDER BY total_registrations DESC
LIMIT 3;
```

15. Event Session Time Conflict

```
SELECT S1.event_id, S1.session_id AS session1, S2.session_id AS session2
FROM Sessions S1
JOIN Sessions S2
    ON S1.event_id = S2.event_id
    AND S1.session_id < S2.session_id
    AND S1.start_time < S2.end_time
    AND S1.end_time > S2.start_time;
```

16. Unregistered Active Users

```
SELECT *
FROM Users U
WHERE registration_date >= CURDATE() - INTERVAL 30 DAY
AND NOT EXISTS (
    SELECT 1
    FROM Registrations R
    WHERE R.user_id = U.user_id
);
```

17. Multi-Session Speakers

```
SELECT speaker_name, COUNT(*) AS session_count
FROM Sessions
GROUP BY speaker_name
HAVING session_count > 1;
```

18. Resource Availability Check

```
SELECT E.event_id, E.title
FROM Events E
LEFT JOIN Resources R ON E.event_id = R.event_id
WHERE R.event_id IS NULL;
```

19. Completed Events with Feedback Summary

```
SELECT E.title, COUNT(DISTINCT R.user_id) AS total_registrations,
       AVG(F.rating) AS avg_rating
FROM Events E
LEFT JOIN Registrations R ON E.event_id = R.event_id
LEFT JOIN Feedback F ON E.event_id = F.event_id
WHERE E.status = 'completed'
GROUP BY E.event_id;
```

20. User Engagement Index

```
SELECT U.full_name,
       COUNT(DISTINCT R.event_id) AS events_attended,
       COUNT(DISTINCT F.feedback_id) AS feedbacks_given
FROM Users U
LEFT JOIN Registrations R ON U.user_id = R.user_id
LEFT JOIN Feedback F ON U.user_id = F.user_id
GROUP BY U.user_id;
```

21. Top Feedback Providers

```
SELECT U.full_name, COUNT(*) AS feedback_count
FROM Feedback F
JOIN Users U ON F.user_id = U.user_id
GROUP BY F.user_id
ORDER BY feedback_count DESC
LIMIT 5;
```

22. Duplicate Registrations Check

```
SELECT user_id, event_id, COUNT(*) AS duplicate_count
FROM Registrations
GROUP BY user_id, event_id
HAVING duplicate_count > 1;
```

23. Registration Trends

```
SELECT DATE_FORMAT(registration_date, '%Y-%m') AS month,
       COUNT(*) AS registration_count
FROM Registrations
WHERE registration_date >= CURDATE() - INTERVAL 12 MONTH
GROUP BY month
ORDER BY month;
```

24. Average Session Duration per Event

```
SELECT event_id,
       AVG(TIMESTAMPDIFF(MINUTE, start_time, end_time)) AS avg_duration_minutes
FROM Sessions
GROUP BY event_id;
```

25. Events Without Sessions

```
SELECT E.event_id, E.title
FROM Events E
LEFT JOIN Sessions S ON E.event_id = S.event_id WHERE S.event_id IS NULL;
```