**1. User Upcoming Events**

SELECT e.title, e.city, e.start\_date

FROM Events e

JOIN Registrations r ON e.event\_id = r.event\_id

JOIN Users u ON r.user\_id = u.user\_id

WHERE e.status = 'upcoming' AND e.city = u.city

ORDER BY e.start\_date;

**2. Top Rated Event**

SELECT e.title, AVG(f.rating) AS avg\_rating, COUNT(\*) AS feedback\_count

FROM Feedback f

JOIN Events e ON f.event\_id = e.event\_id

GROUP BY f.event\_id

HAVING COUNT(\*) >= 10

ORDER BY avg\_rating DESC;

**3. Inactive Users**

SELECT u.\*

FROM Users u

LEFT JOIN Registrations r ON u.user\_id = r.user\_id

WHERE r.registration\_date IS NULL

OR r.registration\_date < CURDATE() - INTERVAL 90 DAY;

**4. Peak Session Hours**

SELECT e.title, COUNT(\*) AS morning\_sessions

FROM Sessions s

JOIN Events e ON s.event\_id = e.event\_id

WHERE TIME(s.start\_time) BETWEEN '10:00:00' AND '11:59:59'

GROUP BY e.event\_id;

**5. Most Active Cities**

SELECT u.city, COUNT(DISTINCT r.user\_id) AS total\_users

FROM Registrations r

JOIN Users u ON r.user\_id = u.user\_id

GROUP BY u.city

ORDER BY total\_users DESC

LIMIT 5;

**6. Event Resource Summary**

SELECT e.title,

SUM(resource\_type = 'pdf') AS pdfs,

SUM(resource\_type = 'image') AS images,

SUM(resource\_type = 'link') AS links

FROM Resources r

JOIN Events e ON r.event\_id = e.event\_id

GROUP BY r.event\_id;

**7. Low Feedback Alerts**

SELECT u.full\_name, f.rating, f.comments, e.title

FROM Feedback f

JOIN Users u ON f.user\_id = u.user\_id

JOIN Events e ON f.event\_id = e.event\_id

WHERE f.rating < 3;

**8. Sessions per Upcoming Event**

SELECT e.title, COUNT(s.session\_id) AS session\_count

FROM Events e

LEFT JOIN Sessions s ON e.event\_id = s.event\_id

WHERE e.status = 'upcoming'

GROUP BY e.event\_id;

**9. Organizer Event Summary**

SELECT u.full\_name, e.status, COUNT(\*) AS event\_count

FROM Events e

JOIN Users u ON e.organizer\_id = u.user\_id

GROUP BY u.user\_id, e.status;

**10. Feedback Gap**

SELECT e.title

FROM Events e

JOIN Registrations r ON e.event\_id = r.event\_id

LEFT JOIN Feedback f ON e.event\_id = f.event\_id

WHERE f.event\_id IS NULL

GROUP BY e.event\_id;

**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;

**12. Event with Maximum Sessions**

SELECT e.title, COUNT(s.session\_id) AS session\_count

FROM Events e

JOIN Sessions s ON e.event\_id = s.event\_id

GROUP BY e.event\_id

ORDER BY session\_count DESC

LIMIT 1;

**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(r.user\_id) AS registrations

FROM Events e

JOIN Registrations r ON e.event\_id = r.event\_id

GROUP BY e.event\_id

ORDER BY registrations DESC

LIMIT 3;

**15. Event Session Time Conflict**

SELECT s1.event\_id, s1.title AS session1, s2.title AS session2

FROM Sessions s1

JOIN Sessions s2 ON s1.event\_id = s2.event\_id AND s1.session\_id < s2.session\_id

WHERE s1.start\_time < s2.end\_time AND s2.start\_time < s1.end\_time;

**16. Unregistered Active Users**

SELECT u.\*

FROM Users u

LEFT JOIN Registrations r ON u.user\_id = r.user\_id

WHERE u.registration\_date >= CURDATE() - INTERVAL 30 DAY

AND r.user\_id IS NULL;

**17. Multi-Session Speakers**

SELECT speaker\_name, COUNT(\*) AS session\_count

FROM Sessions

GROUP BY speaker\_name

HAVING COUNT(\*) > 1;

**18. Resource Availability Check**

SELECT e.title

FROM Events e

LEFT JOIN Resources r ON e.event\_id = r.event\_id

WHERE r.resource\_id IS NULL;

**19. Completed Events with Feedback Summary**

SELECT e.title, COUNT(DISTINCT r.user\_id) AS 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 dup\_count

FROM Registrations

GROUP BY user\_id, event\_id

HAVING COUNT(\*) > 1;

**23. Registration Trends**

SELECT DATE\_FORMAT(registration\_date, '%Y-%m') AS reg\_month,

COUNT(\*) AS total\_registrations

FROM Registrations

WHERE registration\_date >= CURDATE() - INTERVAL 12 MONTH

GROUP BY reg\_month

ORDER BY reg\_month;

**24. Average Session Duration per Event**

SELECT e.title,

AVG(TIMESTAMPDIFF(MINUTE, s.start\_time, s.end\_time)) AS avg\_duration\_mins

FROM Sessions s

JOIN Events e ON s.event\_id = e.event\_id

GROUP BY s.event\_id;

**25. Events Without Sessions**

SELECT e.title

FROM Events e

LEFT JOIN Sessions s ON e.event\_id = s.event\_id

WHERE s.session\_id IS NULL;