**SQL Exercise Answers**

-- 1. User Upcoming Events

SELECT u.full\_name, e.title, e.city, e.start\_date

FROM Users u

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

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

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

ORDER BY e.start\_date;

-- 2. Top Rated Events

SELECT e.title, AVG(f.rating) AS avg\_rating

FROM Events e

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

GROUP BY e.event\_id

HAVING COUNT(f.feedback\_id) >= 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 AND r.registration\_date >= CURDATE() - INTERVAL 90 DAY

WHERE r.registration\_id IS NULL;

-- 4. Peak Session Hours

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

FROM Events e

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

WHERE TIME(s.start\_time) BETWEEN '10:00:00' AND '12:00:00'

GROUP BY e.title;

-- 5. Most Active Cities

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

FROM Users u

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

GROUP BY u.city

ORDER BY user\_count DESC

LIMIT 5;

-- 6. Event Resource Summary

SELECT e.title,

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

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

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

FROM Events e

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

GROUP BY e.title;

-- 7. Low Feedback Alerts

SELECT u.full\_name, 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 total\_events

FROM Events e

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

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

-- 10. Feedback Gap

SELECT DISTINCT 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.feedback\_id IS NULL;

-- 11. Daily New User Count

SELECT registration\_date, COUNT(\*) AS user\_count

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 Events e

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

GROUP BY e.city;

-- 14. Most Registered Events

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

FROM Events e

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

GROUP BY e.event\_id

ORDER BY registration\_count 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 \*

FROM Users u

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

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

AND r.registration\_id IS NULL;

-- 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.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.registration\_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(f.feedback\_id) AS feedback\_count

FROM Users u

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

GROUP BY u.user\_id

ORDER BY feedback\_count DESC

LIMIT 5;

-- 22. Duplicate Registrations Check

SELECT user\_id, event\_id, COUNT(\*) AS registration\_count

FROM Registrations

GROUP BY user\_id, event\_id

HAVING registration\_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 e.title, AVG(TIMESTAMPDIFF(MINUTE, s.start\_time, s.end\_time)) AS avg\_duration\_minutes

FROM Events e

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

GROUP BY e.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;