# ANSI SQL Using MySQL – 25 Practical Exercises

## 1. User Upcoming Events

List all upcoming events a user is registered for in their city.

SELECT e.\*

FROM Events e

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

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

WHERE r.user\_id = ?

AND e.status = 'upcoming'

AND e.city = u.city

ORDER BY e.start\_date;

## 2. Top Rated Events

Events with highest average rating (min 10 feedbacks).

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

FROM Events e

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

GROUP BY e.event\_id, e.title

HAVING COUNT(\*) >= 10

ORDER BY avg\_rating DESC;

## 3. Inactive Users

Users without any event registrations in last 90 days.

SELECT u.\*

FROM Users u

LEFT JOIN Registrations r

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

AND r.registration\_date >= CURDATE() - INTERVAL 90 DAY

WHERE r.registration\_id IS NULL;

## 4. Peak Session Hours

Count of sessions scheduled between 10 AM and 12 PM.

SELECT e.event\_id, e.title, COUNT(\*) AS sessions\_10\_12

FROM Events e

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

WHERE TIME(s.start\_time) >= '10:00:00'

AND TIME(s.start\_time) < '12:00:00'

GROUP BY e.event\_id, e.title;

## 5. Most Active Cities

Top 5 cities with most unique user registrations.

SELECT e.city, COUNT(DISTINCT r.user\_id) AS distinct\_registrations

FROM Events e

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

GROUP BY e.city

ORDER BY distinct\_registrations DESC

LIMIT 5;

## 6. Event Resource Summary

Count of PDFs, images, and links uploaded per event.

SELECT e.event\_id,

e.title,

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

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

SUM(r.resource\_type = 'link') AS links,

COUNT(r.resource\_id) AS total\_resources

FROM Events e

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

GROUP BY e.event\_id, e.title;

## 7. Low Feedback Alerts

Users giving feedback < 3 along with comments and event names.

SELECT u.full\_name,

e.title AS event\_title,

f.rating,

f.comments

FROM Feedback f

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

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

WHERE f.rating < 3

ORDER BY f.feedback\_date;

## 8. Sessions per Upcoming Event

Upcoming events with their session counts.

SELECT e.event\_id,

e.title,

COUNT(s.session\_id) AS session\_count

FROM Events e

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

WHERE e.status = 'upcoming'

GROUP BY e.event\_id, e.title;

## 9. Organizer Event Summary

Number of events per status for each organizer.

SELECT u.user\_id AS organizer\_id,

u.full\_name,

SUM(e.status = 'upcoming') AS upcoming\_cnt,

SUM(e.status = 'completed') AS completed\_cnt,

SUM(e.status = 'cancelled') AS cancelled\_cnt,

COUNT(\*) AS total\_events

FROM Users u

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

GROUP BY u.user\_id, u.full\_name;

## 10. Feedback Gap

Events that have registrations but no feedback.

SELECT e.event\_id,

e.title

FROM Events e

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

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

WHERE f.feedback\_id IS NULL

GROUP BY e.event\_id, e.title;

## 11. Daily New User Count

Number of users who registered each day in last 7 days.

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

Event(s) that have the highest number of sessions.

WITH cnt AS (

SELECT event\_id, COUNT(\*) AS session\_cnt

FROM Sessions

GROUP BY event\_id

)

SELECT e.event\_id, e.title, cnt.session\_cnt

FROM cnt

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

WHERE cnt.session\_cnt = (SELECT MAX(session\_cnt) FROM cnt);

## 13. Average Rating per City

Average feedback rating of events conducted in each city.

SELECT e.city,

ROUND(AVG(f.rating), 2) AS avg\_rating

FROM Events e

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

GROUP BY e.city;

## 14. Most Registered Events

Top 3 events based on total user registrations.

SELECT e.event\_id,

e.title,

COUNT(r.user\_id) AS registrations

FROM Events e

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

GROUP BY e.event\_id, e.title

ORDER BY registrations DESC

LIMIT 3;

## 15. Event Session Time Conflict

Identify overlapping sessions within the same event.

SELECT s1.event\_id,

s1.session\_id AS session\_a,

s2.session\_id AS session\_b

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

## 16. Unregistered Active Users

Users who created an account in the last 30 days but haven’t registered for any events.

SELECT u.\*

FROM Users u

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

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

AND r.registration\_id IS NULL;

## 17. Multi-Session Speakers

Speakers handling more than one session across all events.

SELECT speaker\_name,

COUNT(\*) AS session\_count

FROM Sessions

GROUP BY speaker\_name

HAVING COUNT(\*) > 1;

## 18. Resource Availability Check

Events that do not have any resources uploaded.

SELECT e.event\_id,

e.title

FROM Events e

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

WHERE r.resource\_id IS NULL;

## 19. Completed Events with Feedback Summary

For completed events, show total registrations and average feedback rating.

SELECT e.event\_id,

e.title,

COUNT(DISTINCT r.user\_id) AS registrations,

ROUND(AVG(f.rating), 2) AS avg\_rating

FROM Events e

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

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

WHERE e.status = 'completed'

GROUP BY e.event\_id, e.title;

## 20. User Engagement Index

Calculate how many events each user attended and how many feedbacks they submitted.

SELECT u.user\_id,

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 r.user\_id = u.user\_id

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

GROUP BY u.user\_id, u.full\_name;

## 21. Top Feedback Providers

Top 5 users who have submitted the most feedback entries.

SELECT u.user\_id,

u.full\_name,

COUNT(\*) AS feedbacks

FROM Users u

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

GROUP BY u.user\_id, u.full\_name

ORDER BY feedbacks DESC

LIMIT 5;

## 22. Duplicate Registrations Check

Detect if a user has been registered more than once for the same event.

SELECT user\_id,

event\_id,

COUNT(\*) AS dup\_count

FROM Registrations

GROUP BY user\_id, event\_id

HAVING COUNT(\*) > 1;

## 23. Registration Trends

Month‑wise registration count trend over the past 12 months.

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

COUNT(\*) AS registrations

FROM Registrations

WHERE registration\_date >= DATE\_FORMAT(CURDATE() - INTERVAL 11 MONTH, '%Y-%m-01')

GROUP BY year\_month

ORDER BY year\_month;

## 24. Average Session Duration per Event

Compute the average duration (in minutes) of sessions in each event.

SELECT e.event\_id,

e.title,

ROUND(AVG(TIMESTAMPDIFF(MINUTE, s.start\_time, s.end\_time)), 2) AS avg\_minutes

FROM Events e

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

GROUP BY e.event\_id, e.title;

## 25. Events Without Sessions

List all events that currently have no sessions scheduled under them.

SELECT e.event\_id,

e.title

FROM Events e

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

WHERE s.session\_id IS NULL;