SQL Exercises Answers

1. User Upcoming Events

```
SELECT u.full_name, e.title, 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 average_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 average_rating DESC;
```

3. Inactive Users

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

4. Peak Session Hours (10 AM to 12 PM)

```
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.event_id;
```

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.event_id;
```

7. Low Feedback Alerts

```
SELECT u.full_name, e.title, f.rating, f.comments
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 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 (Last 7 Days)

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

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.registration_id) AS total_registrations
FROM Events e
JOIN Registrations r ON e.event_id = r.event_id
GROUP BY e.event_id
ORDER BY total_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 s1.end_time > s2.start_time;
```

16. Unregistered Active Users

```
SELECT *
FROM Users
WHERE registration_date >= CURDATE() - INTERVAL 30 DAY
AND user_id NOT IN (
  SELECT DISTINCT user_id FROM Registrations
);
```

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.user_id) AS total_registrations,
       AVG(f.rating) AS average_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 (Monthly)

```
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;
```