1. User Upcoming Events

```
SELECT u.full_name, e.title, e.start_date, e.city 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, COUNT(*) AS feedback_count FROM Events e
JOIN Feedback f ON e.event_id = f.event_id GROUP BY e.event_id
HAVING COUNT(*) >= 10
ORDER BY avg_rating DESC;
```

3. Inactive Users

```
SELECT *
FROM Users u
WHERE NOT EXISTS (
    SELECT 1 FROM Registrations r WHERE
    r.user_id = u.user_id
    AND r.registration_date >= CURDATE() - INTERVAL 90 DAY
);
```

4. Peak Session Hours

```
SELECT e.title, COUNT(*) AS session_count 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 s.event_id;
```

5. Most Active Cities

```
SELECT u.city, COUNT(DISTINCT r.user_id) AS total_users FROM Users u
JOIN Registrations r ON u.user_id = r.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 pdf_count, SUM(resource_type =
    'image') AS image_count, SUM(resource_type = 'link') AS link_count
```

```
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, 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, COUNT(e.event_id) AS
    total_events,
    SUM(e.status = 'upcoming') AS upcoming, SUM(e.status =
    'completed') AS completed, SUM(e.status = 'cancelled') AS cancelled
FROM Events e
JOIN Users u ON e.organizer_id = u.user_id GROUP BY u.user_id;
```

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.feedback_id IS
NULL
GROUP BY e.event_id;
```