**Query 1: Join Query with Aggregate Function to Retrieve All Events Along with**

**Their Organizers:**

SELECT e.event\_name, e.start\_date, e.end\_date, e.location, o.organizer\_name

FROM Event e

INNER JOIN Organizer o ON e.organizer\_id = o.organizer\_id;

**Query 2: Join Query to Retrieve Session Performances with Speaker Performer**

**and Event Details:**

SELECT s.title, s.start\_time, s.end\_time, sp.first\_name, sp.last\_name, e.event\_name

FROM Session\_Performance s

INNER JOIN Speaker\_Performer sp ON s.speaker\_performer\_id =

sp.speaker\_performer\_id

INNER JOIN Event e ON s.event\_id = e.event\_id;

**Query 3: Join Query with Aggregate Function to Calculate Total Cost per**

**Event:**

SELECT e.event\_name, SUM(s.cost) AS total\_cost

FROM Event e

LEFT JOIN Vendor v ON e.organizer\_id = v.vendor\_id

LEFT JOIN Service s ON v.vendor\_id = s.vendor\_id

GROUP BY e.event\_name;

**Query 4: Join Query to Retrieve Top 5 Expensive Services with Vendor**

**Information:**

SELECT v.vendor\_name, s.service\_type, s.cost

FROM Service s

INNER JOIN Vendor v ON s.vendor\_id = v.vendor\_id

ORDER BY s.cost DESC

LIMIT 5;

**Query 5: Join Query to Retrieve Events with Expenses Exceeding 50000:**

SELECT e.event\_name, ex.description, ex.amount

FROM Event e

INNER JOIN Expense ex ON e.event\_id = ex.event\_id

WHERE ex.amount > 50000;

**Query 6: Join Query to Retrieve Vendors Associated with Active Events:**

SELECT DISTINCT v.vendor\_name

FROM Vendor v

INNER JOIN Service s ON v.vendor\_id = s.vendor\_id

WHERE EXISTS (

SELECT 1

FROM Event e

WHERE e.status = 'Active'

);

**Query 7: Join Query with Aggregate Function to Calculate Average Cost per**

**Vendor:**

SELECT v.vendor\_name, AVG(s.cost) AS average\_cost FROM Vendor v

LEFT JOIN Service s ON v.vendor\_id = s.vendor\_id

GROUP BY v.vendor\_name;

**Query 8: Join Query to Retrieve Event Details with Session Performances by**

**Speaker Amit Sharma:**

SELECT e.event\_name, s.title, sp.first\_name, sp.last\_name FROM Event e

INNER JOIN Session\_Performance s ON e.event\_id = s.event\_id

INNER JOIN Speaker\_Performer sp ON s.speaker\_performer\_id =

sp.speaker\_performer\_id

WHERE sp.first\_name = 'Amit' AND sp.last\_name = 'Sharma';

**Query 9: Join Query to Retrieve Event Details with Venue Information,**

**Ordered**

**by Venue Price (Descending):**

SELECT e.event\_name, v.venue\_name, v.venue\_address, v.venue\_price FROM Event e

27INNER JOIN Venue v ON e.event\_id = v.event\_id

ORDER BY v.venue\_price DESC;

**Query 10: Join Query with Nested Subquery and Aggregate Function to**

**Retrieve**

**Events with Expenses Exceeding the Average Expense Amount:**

SELECT e.event\_name, ex.description, ex.amount

FROM Event e INNER JOIN Expense ex ON e.event\_id = ex.event\_id

WHERE ex.amount > (

SELECT AVG(amount) FROM Expense

);

**Query 11: Join Query with Nested Subquery to Retrieve Events with Start**

**Date**

**Equal to the Earliest Start Date and End Date Equal to the Latest End Date**

**for**

**Each Organizer:**

SELECT e.event\_name, o.organizer\_name, e.start\_date, e.end\_date

FROM Event e

INNER JOIN Organizer o ON e.organizer\_id = o.organizer\_id

WHERE e.start\_date <= ALL (

SELECT start\_date

FROM Event

WHERE organizer\_id = e.organizer\_id

) AND e.end\_date >= ALL (

SELECT end\_date

FROM Event

WHERE organizer\_id = e.organizer\_id

);

**Query 12: Join Query with Nested Subquery and Aggregate Function to**

**Retrieve**

**Event Details with Organizer Information and Average Service Cost per**

**Event:**

SELECT e.event\_name,

o.organizer\_name,

(SELECT AVG(cost)

FROM Service

WHERE vendor\_id IN (

SELECT vendor\_id

FROM Vendor

WHERE event\_id = e.event\_id

)

) AS average\_service\_cost

FROM Event e

INNER JOIN Organizer o ON e.organizer\_id = o.organizer\_id;

**Query 13: Join Query with Nested Subquery and Aggregate Function to**

**Retrieve**

**Event Details with Organizer Information and Count of Sessions with Speaker**

**Name Starting with 'A':**

SELECT e.event\_name,

o.organizer\_name,

(SELECT COUNT(\*)

FROM Session\_Performance

30WHERE event\_id = e.event\_id AND speaker\_performer\_id IN (

SELECT speaker\_performer\_id

FROM Speaker\_Performer

WHERE first\_name LIKE 'A%'

)

) AS sessions\_with\_a\_speaker

FROM Event e

INNER JOIN Organizer o ON e.organizer\_id = o.organizer\_id;

**Query 14: Join Query with Aggregate Function to Calculate Total Expenses**

**per**

**Organizer and Retrieve Top 5 Organizers by Total Expenses:**

SELECT o.organizer\_name, SUM(e.amount) AS total\_expenses

FROM Organizer o

JOIN Event ev ON o.organizer\_id = ev.organizer\_id

JOIN Expense e ON ev.event\_id = e.event\_id

GROUP BY o.organizer\_name

ORDER BY total\_expenses DESC

LIMIT 5;