### **Task – 3**

**Aggregate functions, Having, Order By, GroupBy and Joins:**

1. **Write a SQL query to List Events and Their Average Ticket Prices.**

select event\_name , avg(ticket\_price) from Event\_Table group by event\_name;

+-----------------------+-------------------+

| event\_name | avg(ticket\_price) |

+-----------------------+-------------------+

| Avengers The End Game | 1000.000000 |

| Interstellar | 2000.000000 |

| Fault in our stars | 4000.000000 |

| Folk | 6000.000000 |

| Melody Evening | 1500.000000 |

| Rythm Party | 3000.000000 |

| Basket Ball | 2500.000000 |

| Fun Run Challenge | 3500.000000 |

| Cricket | 500.000000 |

| Football | 4500.000000 |

+-----------------------+-------

1. **Write a SQL query to Calculate the Total Revenue Generated by Events.**

select event\_id , event\_name , sum(ticket\_price) from Event\_Table GROUP BY event\_id , event\_name;

+----------+-----------------------+-------------------+

| event\_id | event\_name | sum(ticket\_price) |

+----------+-----------------------+-------------------+

| 201 | Avengers The End Game | 1000.00 |

| 202 | Interstellar | 2000.00 |

| 203 | Fault in our stars | 4000.00 |

| 204 | Folk | 6000.00 |

| 205 | Melody Evening | 1500.00 |

| 206 | Rythm Party | 3000.00 |

| 207 | Basket Ball | 2500.00 |

| 208 | Fun Run Challenge | 3500.00 |

| 209 | Cricket | 500.00 |

| 210 | Football | 4500.00 |

+----------+-----------------------+---

**3. Write a SQL query to find the event with the highest ticket sales.**

select event\_id , event\_name , sum(ticket\_price) AS highestsales from Event\_Table GROUP BY event\_id , event\_name ORDER BY highestsales DESC LIMIT 1;

+----------+------------+--------------+

| event\_id | event\_name | highestsales |

+----------+------------+--------------+

| 204 | Folk | 6000.00 |

+----------+------------+--------------+

1. **Write a SQL query to Calculate the Total Number of Tickets Sold for Each Event.**

select event\_id , event\_name , count(\*) from Event\_Table GROUP BY event\_name , event\_id;

+----------+-----------------------+----------+

| event\_id | event\_name | count(\*) |

+----------+-----------------------+----------+

| 201 | Avengers The End Game | 1 |

| 202 | Interstellar | 1 |

| 203 | Fault in our stars | 1 |

| 204 | Folk | 1 |

| 205 | Melody Evening | 1 |

| 206 | Rythm Party | 1 |

| 207 | Basket Ball | 1 |

| 208 | Fun Run Challenge | 1 |

| 209 | Cricket | 1 |

| 210 | Football | 1 |

+----------+-----------------------+----------+

1. **Write a SQL query to Find Events with No Ticket Sales.**

SELECT

-> e.event\_name

-> FROM

-> Event\_Table e LEFT JOIN

-> Booking\_Table b ON e.event\_id = b.event\_id WHERE

-> b.booking\_id IS NULL;

**6. Write a SQL query to Find the User Who Has Booked the Most Tickets.**

select customer\_name c , sum(b.num\_tickets) from Customer\_Table c JOIN Booking\_Table b where c.booking\_id = b.booking\_id GROUP BY c.customer\_name ORDER BY sum(num\_tickets) DESC LIMIT 1;

+--------+--------------------+

| c | sum(b.num\_tickets) |

+--------+--------------------+

| harish | 5 |

+--------+--------------------+

**7. Write a SQL query to List Events and the total number of tickets sold for each month.**

select e.event\_name , sum(b.num\_tickets) , YEAR(b.booking\_date) AS year , MONTH(b.booking\_date) as month from Event\_Table e JOIN Booking\_Table b ON e.event\_id=b.event\_id GROUP BY event\_name , year , month;

+-----------------------+--------------------+------+-------+

| event\_name | sum(b.num\_tickets) | year | month |

+-----------------------+--------------------+------+-------+

| Avengers The End Game | 1 | 2025 | 2 |

| Interstellar | 2 | 2025 | 2 |

| Fault in our stars | 3 | 2025 | 2 |

| Folk | 1 | 2025 | 2 |

| Melody Evening | 4 | 2025 | 2 |

| Rythm Party | 2 | 2025 | 2 |

| Basket Ball | 1 | 2025 | 2 |

| Fun Run Challenge | 5 | 2025 | 2 |

| Cricket | 1 | 2025 | 2 |

| Football | 2 | 2025 | 2 |

+-----------------------+--------------------+------+-------+

**8. Write a SQL query to calculate the average Ticket Price for Events in Each Venue.**

select v.venue\_name , avg(e.ticket\_price) from Event\_Table e JOIN Venue\_Tables v ON e.venue\_id = v.venue\_id GROUP BY v.venue\_name;

+------------------+---------------------+

| venue\_name | avg(e.ticket\_price) |

+------------------+---------------------+

| Theatre Arena | 1000.000000 |

| Silver Screen | 2000.000000 |

| Movie magic | 4000.000000 |

| Music Venue | 6000.000000 |

| sound stage | 1500.000000 |

| Classic Fusion | 3000.000000 |

| Game Stadium | 2500.000000 |

| Sports Venue | 3500.000000 |

| playground Arena | 500.000000 |

| champion field | 4500.000000 |

+------------------+---------------------+

**9. Write a SQL query to calculate the total Number of Tickets Sold for Each Event Type.**

select e.event\_type , sum(b.num\_tickets) from Event\_Table e JOIN Booking\_Table b ON e.event\_id = b.event\_id GROUP BY e.event\_type;

+------------+--------------------+

| event\_type | sum(b.num\_tickets) |

+------------+--------------------+

| Movie | 6 |

| Concert | 7 |

| Sports | 9 |

+------------+--------------------+

**10. Write a SQL query to calculate the total Revenue Generated by Events in Each Year.**

select sum(b.total\_cost) , YEAR(b.booking\_date) as year from Booking\_Table b JOIN Event\_Table e ON e.event\_id = b.event\_id GROUP BY year;

+-------------------+------+

| sum(b.total\_cost) | year |

+-------------------+------+

| 64500.00 | 2025 |

+-------------------+------+

**11. Write a SQL query to list users who have booked tickets for multiple events.**

select c.customer\_name from Customer\_Table c JOIN Booking\_Table b ON c.customer\_id=b.customer\_id GROUP BY c.customer\_name HAVING count(DISTINCT b.event\_id) > 1;

Empty set (0.01 sec)

**12. Write a SQL query to calculate the Total Revenue Generated by Events for Each User.**

select c.customer\_name , sum(b.total\_cost) from Customer\_Table c JOIN Booking\_Table b ON b.customer\_id = c.customer\_id GROUP BY c.customer\_name;

+---------------+-------------------+

| customer\_name | sum(b.total\_cost) |

+---------------+-------------------+

| priya | 1000.00 |

| preethi | 4000.00 |

| ram | 12000.00 |

| raghul | 6000.00 |

| raj | 6000.00 |

| ramya | 6000.00 |

| anjali | 2500.00 |

| harish | 17500.00 |

| arjun | 500.00 |

| ramesh | 9000.00 |

+---------------+-------------------+

**13. Write a SQL query to calculate the Average Ticket Price for Events in Each Category and Venue.**

select e.event\_type , avg(e.ticket\_price) , v.venue\_name from Event\_Table e JOIN Venue\_Tables v ON e.venue\_id = v.venue\_id GROUP BY e.event\_type , v.venue\_name;

+------------+---------------------+------------------+

| event\_type | avg(e.ticket\_price) | venue\_name |

+------------+---------------------+------------------+

| Movie | 1000.000000 | Theatre Arena |

| Movie | 2000.000000 | Silver Screen |

| Movie | 4000.000000 | Movie magic |

| Concert | 6000.000000 | Music Venue |

| Concert | 1500.000000 | sound stage |

| Concert | 3000.000000 | Classic Fusion |

| Sports | 2500.000000 | Game Stadium |

| Sports | 3500.000000 | Sports Venue |

| Sports | 500.000000 | playground Arena |

| Sports | 4500.000000 | champion field |

+------------+---------------------+------------------+

**14. Write a SQL query to list Users and the Total Number of Tickets They've Purchased in the Last 30 Days.**

select c.customer\_name , count(b.num\_tickets) from Customer\_Table c JOIN Booking\_Table b ON b.customer\_id = c.customer\_id WHERE b.booking\_date >= CURDATE() - INTERVAL 30 DAY GROUP BY c.customer\_name;

+---------------+----------------------+

| customer\_name | count(b.num\_tickets) |

+---------------+----------------------+

| anjali | 1 |

| harish | 1 |

| arjun | 1 |

| ramesh | 1 |

+---------------+----------------------+