1.

SELECT

event\_name,

AVG(ticket\_price) AS average\_ticket\_price

FROM

events

GROUP BY

event\_name;

2.

SELECT

SUM(ticket\_price) AS total\_revenue

FROM

events;

3.

SELECT

event\_name,

num\_tickets\_sold

FROM

events

ORDER BY

num\_tickets\_sold DESC

LIMIT 1;

4. SELECT

event\_name,

SUM(num\_tickets\_sold) AS total\_tickets\_sold

FROM

events

GROUP BY

event\_name;

5.

SELECT

event\_name

FROM

events

WHERE

num\_tickets\_sold IS NULL OR num\_tickets\_sold = 0;

6.

SELECT

customer\_name,

SUM(num\_tickets) AS total\_tickets\_booked

FROM

bookings

GROUP BY

customer\_name

ORDER BY

total\_tickets\_booked DESC

LIMIT 1;

7.

SELECT

EXTRACT(MONTH FROM booking\_date) AS month,

EXTRACT(YEAR FROM booking\_date) AS year,

event\_name,

SUM(num\_tickets) AS total\_tickets\_sold

FROM

bookings

GROUP BY

year,

month,

event\_name

ORDER BY

year,

month;

8.

SELECT

venue\_name,

AVG(ticket\_price) AS average\_ticket\_price

FROM

Events

GROUP BY

venue\_name;

9. SELECT

event\_type,

SUM(num\_tickets) AS total\_tickets\_sold

FROM

bookings

GROUP BY

event\_type;

10.

SELECT

YEAR(booking\_date) AS event\_year,

SUM(total\_cost) AS total\_revenue

FROM

bookings

GROUP BY

event\_year;

11.

SELECT

customer\_name,

COUNT(DISTINCT event\_id) AS num\_events\_booked

FROM

bookings

GROUP BY

customer\_name

HAVING

num\_events\_booked > 1;

12.

SELECT

customer\_name,

SUM(total\_cost) AS total\_revenue

FROM

bookings

GROUP BY

customer\_name;

13.

SELECT

venue\_name,

event\_type,

AVG(ticket\_price) AS average\_ticket\_price

FROM

events

GROUP BY

venue\_name, event\_type;

14.

SELECT

u.user\_id,

u.user\_name,

COUNT(b.num\_tickets) AS total\_tickets\_purchased

FROM

users u

JOIN

bookings b ON u.user\_id = b.user\_id

WHERE

b.booking\_date >= CURRENT\_DATE - INTERVAL 30 DAY

GROUP BY

u.user\_id, u.user\_name;