#1. Show film which dont have any screening

SELECT DISTINCT name

FROM screening;

SELECT \*

FROM film

WHERE film.id NOT IN(select distinct film\_id FROM screening);

#2. Who book more than 1 seat in 1 booking

SELECT COUNT(seat\_id) as number\_of\_seat, booking\_id

FROM reserved\_seat

GROUP BY booking\_id

HAVING COUNT(seat\_id) >1;

SELECT \*

FROM customer

JOIN booking ON customer.id = booking.customer\_id

WHERE booking.id in(SELECT booking\_id FROM(

SELECT COUNT(seat\_id) as number\_of\_seat, booking\_id

FROM reserved\_seat

GROUP BY booking\_id

HAVING COUNT(seat\_id) >1

) as T);

#3. Show room show more than 2 film in one day

SELECT COUNT(T.id) as film, T.date, T.room\_id, room.name

FROM

(SELECT distinct f.name,f.id, date(s.start\_time) as date, s.room\_id

FROM film as f JOIN screening as s

ON f.id = s.film\_id) as T

JOIN room on T.room\_id = room.id

GROUP BY T.date, T.room\_id

HAVING COUNT(T.id) >2;

# 4.which room show the least film ?

SELECT COUNT(s.id) as number\_of\_screening, s.room\_id

FROM screening as s

GROUP BY s.room\_id

order by number\_of\_screening

;

#5. Query to find films that don't have any bookings:

SELECT film.id, film.name

FROM film

LEFT JOIN screening ON film.id = screening.film\_id

LEFT JOIN booking ON screening.id = booking.screening\_id

WHERE booking.id IS NULL;

#6. Query to find the film that has been shown in the most number of rooms:

SELECT film.id, film.name, COUNT(DISTINCT screening.room\_id) AS room\_count

FROM film

JOIN screening ON film.id = screening.film\_id

GROUP BY film.id

ORDER BY room\_count DESC

LIMIT 1;

#7. Query to show the number of films shown each day of the week, ordered in descending order:

SELECT DAYNAME(screening.start\_time) AS day\_of\_week, COUNT(DISTINCT screening.film\_id) AS film\_count

FROM screening

GROUP BY day\_of\_week

ORDER BY film\_count DESC;

#8. Query to show the total length of each film shown on 28/5/2022:

SELECT film.id, film.name, SUM(film.length\_min) AS total\_length

FROM film

JOIN screening ON film.id = screening.film\_id

WHERE DATE(screening.start\_time) = '2022-05-28'

GROUP BY film.id;

#9. Query to find films with showing times above and below the average showing time of all films:

WITH avg\_time AS (

SELECT AVG(film.length\_min) AS average\_time

FROM film

)

SELECT film.id, film.name, film.length\_min,

CASE

WHEN film.length\_min > (SELECT average\_time FROM avg\_time) THEN 'Above average'

ELSE 'Below average'

END AS comparison\_to\_average

FROM film;

#10. Query to find the room with the least number of seats:

SELECT room.id, room.name, room.no\_seats

FROM room

ORDER BY room.no\_seats ASC

LIMIT 1;

#11. Query to find rooms with a number of seats greater than the average number of seats in all rooms:

SELECT room.id, room.name, room.no\_seats

FROM room

WHERE room.no\_seats > (SELECT AVG(room.no\_seats) FROM room);

#13. Query to show films with total screenings, ordered by total screenings, but only showing data for films with more than 10 total screenings:

SELECT film.id, film.name, COUNT(screening.id) AS total\_screenings

FROM film

JOIN screening ON film.id = screening.film\_id

GROUP BY film.id

HAVING total\_screenings > 10

ORDER BY total\_screenings DESC;