

-- 1. **find out how many seats are booked in a Rajhansh theatre on a particular date for specific movies**

```
SELECT S.Screen_No, COUNT(*) AS Booked_Seats
FROM Screen S
INNER JOIN "Show" SH ON S.Screen_No = SH.Screen_No AND S.Theatre_Id =
SH.Theatre_Id
LEFT JOIN Booking B ON SH.Show_Id = B.Show_Id AND S.Screen_No = B.Screen_No AND
S.Theatre_Id = B.Theatre_Id
WHERE SH.Theatre_Id = (SELECT Theatre_Id FROM Theatre WHERE Theatre_Name =
'Rajhansh')
AND SH.Show_Date = '2024-04-18'
AND SH.Movie_Name IN ('Dangal', 'Darr')
GROUP BY S.Screen_No;
```

-- 2. **find the movie for which all tickets for the show is booked for at least one show**

```
SELECT DISTINCT S.Movie_Name, S.Release_Date
FROM (
    SELECT B.Show_Id, COUNT(*) AS Booked_Tickets
    FROM Booking B
    GROUP BY B.Show_Id
) AS BookingCount
JOIN "Show" S ON BookingCount.Show_Id = S.Show_Id
JOIN Screen SC ON S.Screen_No = SC.Screen_No AND S.Theatre_Id = SC.Theatre_Id
GROUP BY S.Show_Id, S.Movie_Name, S.Release_Date
HAVING SUM(BookingCount.Booked_Tickets) = SC.Total_No_Of_Silver_Seats +
SC.Total_No_Of_Gold_Seats + SC.Total_No_Of_Diamond_Seats;
```

-- 3. **Retrieve the names of the artists who have worked in at least one movie of each genre**

```
SELECT A."Name"
FROM Artist A
JOIN Works_On W ON A.Artist_Id = W.Artist_Id
JOIN Genre G ON W.Movie_Name = G.Movie_Name AND W.Release_Date =
G.Release_Date
GROUP BY A.Artist_Id, A."Name"
HAVING COUNT(DISTINCT G.Genre_Type) = (
    SELECT COUNT(DISTINCT Genre_Type)
    FROM Genre
);
```

-- 4. find the user who have not booked diamond seat but booked gold or silver seat anytime.

```
SELECT DISTINCT U."Name",U.User_Id
FROM "User" U
JOIN Booking B ON U.User_Id = B.User_Id
JOIN Seat ST ON B.Screen_No = ST.Screen_No AND B.Theatre_Id = ST.Theatre_Id AND
B.Seat_No = ST.Seat_No
WHERE ST.Type_Of_Seat IN ('Silver', 'Gold')
AND NOT EXISTS (
    SELECT B2.User_Id
    FROM Seat SD
```

```
WHERE B.Screen_No = SD.Screen_No AND B.Theatre_Id = SD.Theatre_Id AND  
B.Seat_No = SD.Seat_No  
  
AND SD.Type_Of_Seat = 'Diamond'  
  
);
```

-- 5. Retrieve the names of users who have booked the same seat for multiple shows

```
SELECT DISTINCT U."Name"  
  
FROM Booking B1  
  
JOIN Booking B2 ON B1.Seat_No = B2.Seat_No AND B1.Show_Id <> B2.Show_Id  
  
JOIN "User" U ON B1.User_Id = U.User_Id  
  
GROUP BY U.User_Id, U."Name", B1.Seat_No  
  
HAVING COUNT(DISTINCT B1.Show_Id) > 1;
```

-- 6. retrieve the details of the best-rated movie

```
SELECT m.Movie_Name, m.Release_Date, AVG(r.Rating) AS Average_Rating  
  
FROM Movies m  
  
JOIN Reviews r ON m.Movie_Name = r.Movie_Name AND m.Release_Date =  
r.Release_Date  
  
GROUP BY m.Movie_Name, m.Release_Date  
  
ORDER BY Average_Rating DESC  
  
LIMIT 1;
```

-- 7. Retrieve the name of users who have booked a particular movie more than once

```
SELECT "Name"
FROM "User"
WHERE User_Id IN (
    SELECT User_Id
    FROM Booking
    WHERE Movie_Name = 'YourMovieName'
    GROUP BY User_Id
    HAVING COUNT(*) > 1
);
```

-- 8. retrieve the names of artists who worked in movies with a budget higher than the average budget of all movies

```
SELECT DISTINCT A.Name
FROM Artist A
JOIN Works_On W ON A.Artist_Id = W.Artist_Id
JOIN Movies M ON W.Movie_Name = M.Movie_Name AND W.Release_Date =
M.Release_Date
WHERE M.Budget > (
    SELECT AVG(Budget)
    FROM Movies
);
```

--9 find movie-wise artists who work in more than one role for a movie

```
SELECT W.Movie_Name, W.Release_Date, A."Name",A.Artist_Id, COUNT(*) AS Role_Count
FROM Works_On W
JOIN Artist A ON W.Artist_Id = A.Artist_Id
GROUP BY W.Movie_Name, W.Release_Date, A."Name",A.Artist_Id
HAVING COUNT(*) > 1;
```

--10 retrieve the name of the movie that has been houseful most of the time

```
SELECT m.Movie_Name, m.Release_Date
FROM Movies m
INNER JOIN Booking b ON m.Movie_Name = b.Movie_Name AND m.Release_Date =
b.Release_Date
INNER JOIN Show s ON b.Show_Id = s.Show_Id
INNER JOIN Screen sc ON s.Screen_No = sc.Screen_No AND s.Theatre_Id = sc.Theatre_Id
GROUP BY m.Movie_Name, m.Release_Date, s.Show_Id
HAVING COUNT(b.Show_Id) = (
    SELECT SUM(sc.Total_No_Of_Gold_Seats + sc.Total_No_Of_Silver_Seats +
sc.Total_No_Of_Diamond_Seats)
    FROM Screen sc
    WHERE sc.Screen_No = s.Screen_No AND sc.Theatre_Id = s.Theatre_Id
)
ORDER BY COUNT(b.Show_Id) DESC;
```

--11 find the user which comments maximum time

```
SELECT User_Id, COUNT(*) AS Comment_Count
```

```
FROM Reviews
```

```
GROUP BY User_Id
```

```
ORDER BY Comment_Count DESC
```

```
LIMIT 1;
```

--12 List all the theaters along with the number of screens they have, ordered by the number of screens in descending order

```
SELECT Theatre_Name, No_Of_Screens
```

```
FROM Theatre
```

```
ORDER BY No_Of_Screens DESC;
```

--13 find the actor wise no of movies in descending order

```
SELECT A.Artist_Id , A.Name AS Artist_Name, COUNT(W.Movie_Name) AS No_Of_Movies
```

```
FROM Artist A
```

```
JOIN Works_On W ON A.Artist_Id = W.Artist_Id
```

```
GROUP BY A.Artist_Id, A.Name
```

```
ORDER BY No_Of_Movies DESC;
```

--14 Retrieve the name of the artist who worked on most genre action movies

```
SELECT A."Name", A.Artist_Id, No_Of_Movie
FROM Artist A
JOIN Role R ON A.Artist_Id = R.Artist_Id
JOIN Genre G ON R.Movie_Name = G.Movie_Name AND R.Release_Date = G.Release_Date
WHERE G.Genre_Type = 'Action'
GROUP BY A.Artist_Id, A."Name", COUNT(*) AS No_Of_Movie
ORDER BY No_Of_Movie DESC
LIMIT 1;
```

--15 retrieve the names of actors who have both producer and actor roles in a movie

```
SELECT DISTINCT A."Name"
FROM Artist A
INNER JOIN Role R ON A.Artist_Id = R.Artist_Id
WHERE R.Role_Name IN ('Producer', 'Actor')
GROUP BY A."Name", R.Movie_Name
HAVING COUNT(DISTINCT R.Role_Name) = 2;
```

--16 List Users who Booked a Seat but Did Not Leave a Review

```
SELECT DISTINCT U."Name"
FROM "User" U
```

```
JOIN Booking B ON U.User_Id = B.User_Id

LEFT JOIN Reviews R ON B.Movie_Name = R.Movie_Name AND B.Release_Date =
R.Release_Date AND U.User_Id = R.User_Id

WHERE R.User_Id IS NULL;
```

--17 Retrieve Users who Booked the Most Expensive Seat.

```
SELECT DISTINCT U."Name", S.Cost_Of_Diamond_Class
FROM "User" U
JOIN Booking B ON U.User_Id = B.User_Id
JOIN "Show" S ON B.Show_Id = S.Show_Id
WHERE (B.Theatre_Id, B.Screen_No, B.Seat_No) = (
    SELECT TOP 1 WITH TIES S2.Theatre_Id, S2.Screen_No, S2.Seat_No
    FROM Booking B2
    JOIN "Show" S2 ON B2.Show_Id = S2.Show_Id
    ORDER BY S2.Cost_Of_Diamond_Class DESC
);
```

--18 find the longest-running movie ever from Release date in theatre

```
SELECT m.Movie_Name, m.Release_Date,
    MAX(s.Show_Date) - MIN(s.Show_Date) AS Duration
FROM Movies m
GROUP BY m.Movie_Name, m.Release_Date
```


ORDER BY Duration DESC

LIMIT 1;

--19 Retrieve Movies that Were Released in All Available Languages

SELECT Movie_Name

FROM Language

GROUP BY Movie_Name

HAVING COUNT(DISTINCT Language_Name) = (SELECT COUNT(DISTINCT
Language_Name) FROM Language);

--20 Find the Gender Distribution of Artists Who Have Worked in Movies Released After
2022

SELECT Gender, COUNT(*) AS Artist_Count

FROM Artist A

JOIN Works_On W ON A.Artist_Id = W.Artist_Id

JOIN Movies M ON W.Movie_Name = M.Movie_Name AND W.Release_Date =
M.Release_Date

WHERE M.Release_Date > '2022-12-31'

GROUP BY Gender;