Task 1

In the given dataset, the dir (director) and ctry (country) columns have missing values that contain meaningful information. To make the data more user friendly, write a single SELECT statement that:

a. replaces “-” with not given for both columns.

b. Renames dir as director and ctry as country. Make each word in these columns start with a capital letter.

SQL code:

update movies

set ctry = 'not given'

where ctry = '-'

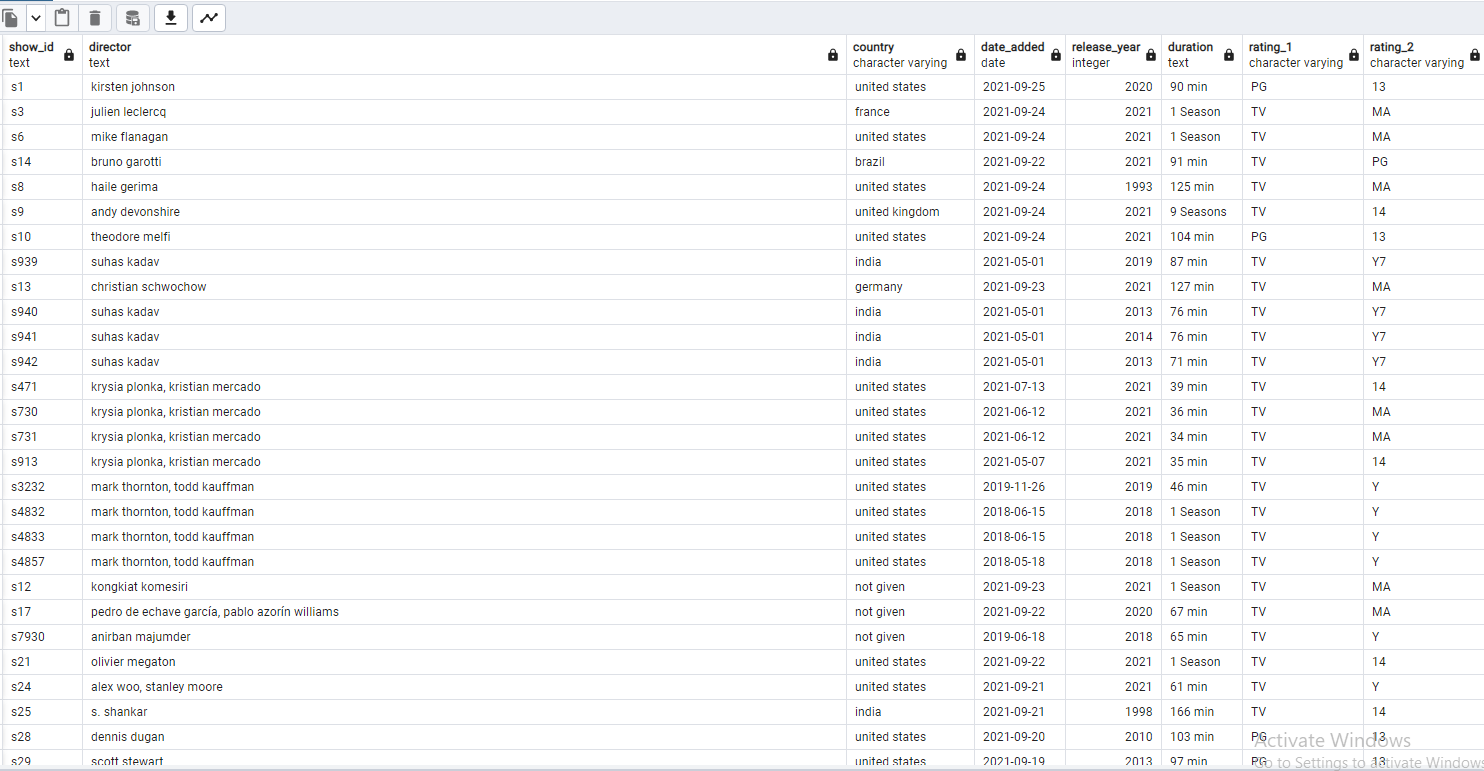
alter table movies

rename column dir to director;

alter table movies

rename column ctry to country;

Output:



Task 2

The management wanted to see the months when the shows were released. However, we do not have this information in our table explicitly. Write an SQL statement to create a column “month\_added“ containing the months extracted from the “date\_added.“ Make it reading friendly using conditional case structure so that “1“ turns   into “Jan,“ and “2“ turns into “Feb,“ and so on.

SQL code:

select extract (month from date\_added) from movies

alter table movies

add column month\_added varchar (10)

update movies

set month\_added = extract (month from date\_added)

select month\_added,

case

when month\_added = '2' then 'Feb'

when month\_added = '3' then 'Mar'

when month\_added = '4' then 'Apr'

when month\_added = '5' then 'May'

when month\_added = '6' then 'Jun'

when month\_added = '7' then 'Jul'

when month\_added = '8' then 'Aug'

when month\_added = '9' then 'Sep'

when month\_added = '10' then 'Oct'

when month\_added = '11' then 'Nov'

when month\_added = '12' then 'Dec'

end

from movies

update movies

set month\_added = case

when month\_added = '1' then 'Jan'

when month\_added = '2' then 'Feb'

when month\_added = '3' then 'Mar'

when month\_added = '4' then 'Apr'

when month\_added = '5' then 'May'

when month\_added = '6' then 'Jun'

when month\_added = '7' then 'Jul'

when month\_added = '8' then 'Aug'

when month\_added = '9' then 'Sep'

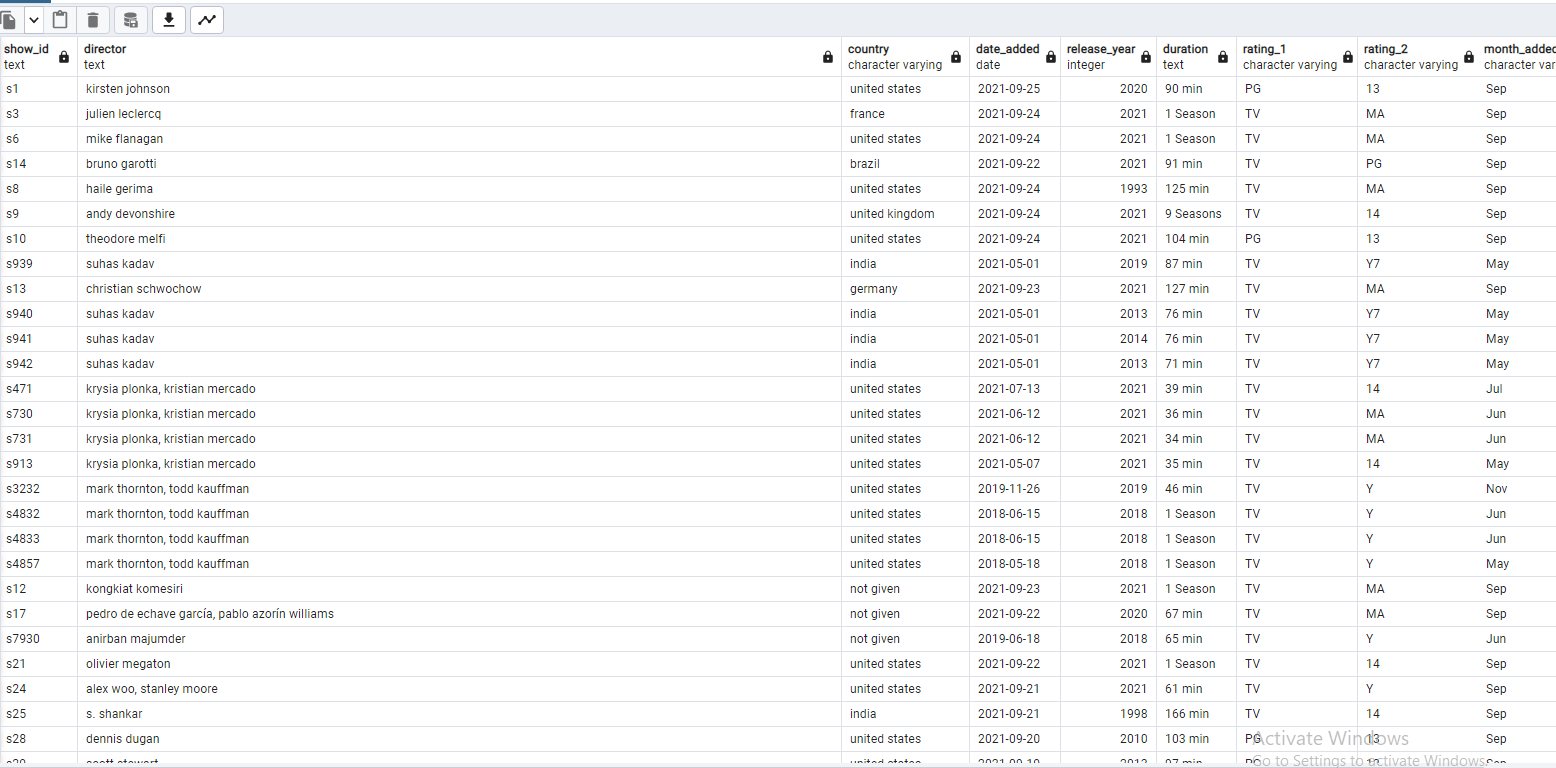
when month\_added = '10' then 'Oct'

when month\_added = '11' then 'Nov'

when month\_added = '12' then 'Dec'

end

Output:



Task 3

There is no explicit information if the title is a movie or a TV series. However, the duration column has either minutes or the number of seasons in it. Create a select statement that has two columns: “duration“ and a conditional one named “title\_type“ with a “TV series“ value if the corresponding duration has “Season“ in it and “Movie“ otherwise.

SQL code:

select duration,

case

when duration like '%min%' then 'Movie'

when duration like '%Sea%' then 'TV Series'

end as title\_type

from movies

alter table movies

add column title\_type varchar(20)

update movies

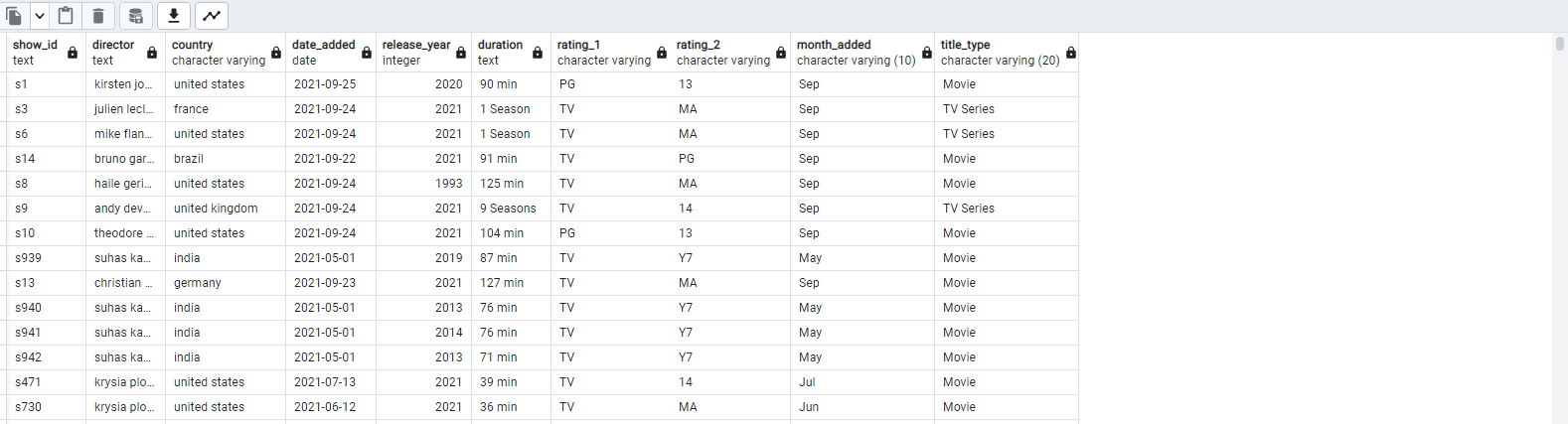
set title\_type = case

when duration like '%min%' then 'Movie'

when duration like '%Sea%' then 'TV Series'

end

output:



Task 4

Write an SQL statement that concatenates “rating\_1“ and “rating\_2“ with “-” between the values into a single column and name it “rating.“

SQL code:

select concat(rating\_1,'-',rating\_2) from movies

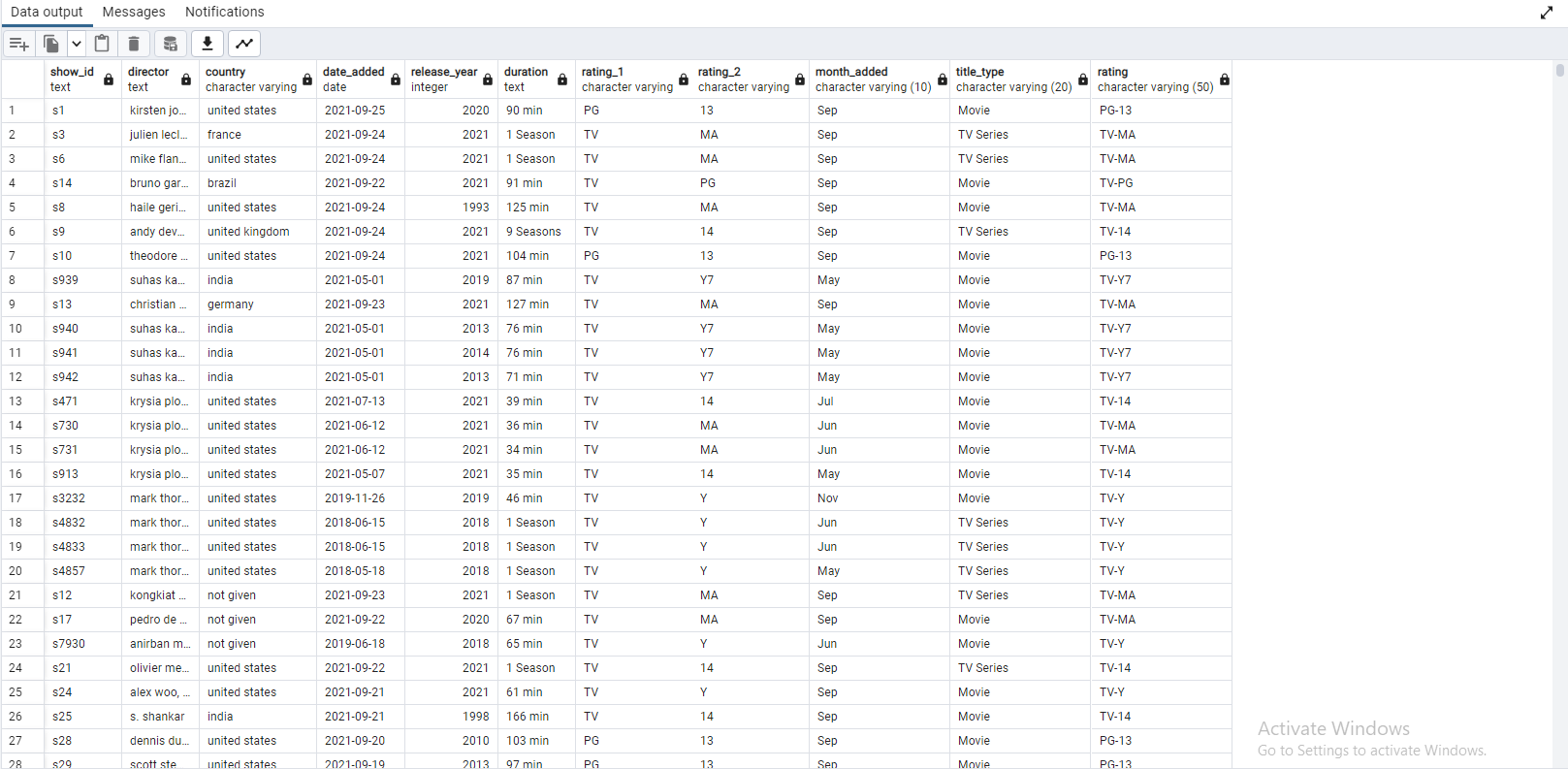
alter table movies

add column rating varchar (50)

update movies

set rating = concat(rating\_1,'-',rating\_2)

Output:



Task 5

Write a single SQL statement that combines the processing and cleaning steps from #1 through #4 above.  a The output data should have the following columns: “show\_id,“ director,“ “country,“ “month\_added,“ “date\_added,“ “release\_year,“ title\_type,“  “rating“ and “duration.“

SQL code:

select show\_id, director,country,month\_added,date\_added, release\_year,title\_type,rating,duration from movies;

Output:

