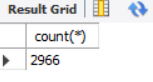
**--Loading the table data**

SELECT \* FROM db1.netflix\_data\_1;



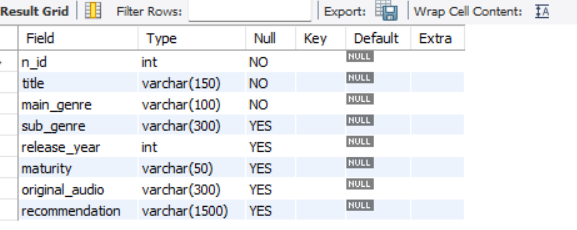
**--Count of total entries**

select count(\*) from netflix\_data\_1;

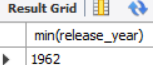
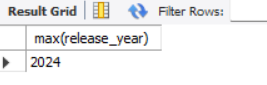


**--Describing the type of data present**

describe netflix\_data\_1;



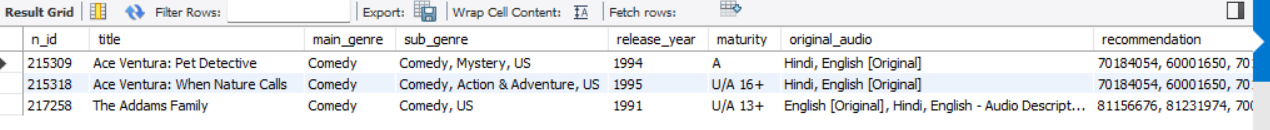
**--Displaying the minimum and maximum release year found in the table data**

select max(release\_year) from netflix\_data\_1;

select min(release\_year) from netflix\_data\_1;

select \* from netflix\_data\_1 where main\_genre = 'comedy' limit 3 ;



**-- Content over time**

SELECT

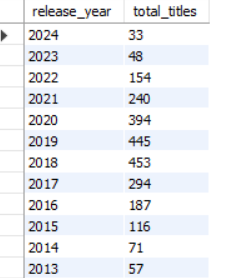
release\_year, COUNT(\*) AS total\_titles

FROM

netflix\_data\_1

GROUP BY release\_year

ORDER BY release\_year DESC;

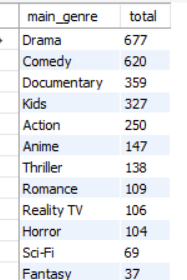


SELECT main\_genre, COUNT(\*) AS total

FROM netflix\_data\_1

GROUP BY main\_genre

ORDER BY total DESC;



SELECT sub\_genre, COUNT(\*) AS total

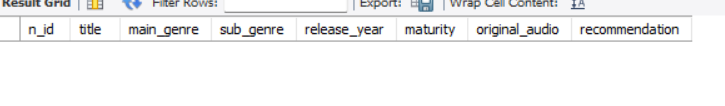
FROM netflix\_data\_1

GROUP BY sub\_genre ORDER BY total DESC limit 10;

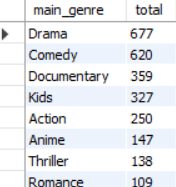


**-- Checking if subgenre is empty or having any missing values**

select \* from netflix\_data\_1 where sub\_genre = 'null' or sub\_genre='';

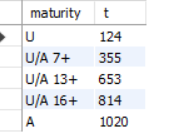


select main\_genre, count(\*) as total from netflix\_data\_1 group by main\_genre order by total desc limit 10;



**-- Maturity rating distribution**

select maturity, count(\*) as t from netflix\_data\_1 group by maturity order by t;



**-- Count of titles with "us" in subgenre**

select count(\*) as us\_titles from netflix\_data\_1 where sub\_genre like '%Us%';



**-- Most frequent original audio languages**

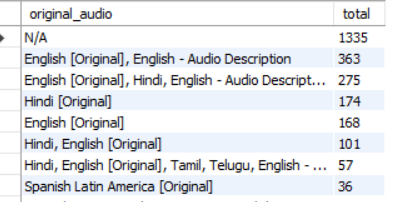
SELECT original\_audio, COUNT(\*) AS total

FROM netflix\_data\_1

GROUP BY original\_audio

ORDER BY total DESC

LIMIT 10;



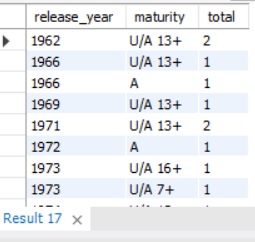
**-- Count of titles per maturity per year**

SELECT release\_year, maturity, COUNT(\*) AS total

FROM netflix\_data\_1

GROUP BY release\_year, maturity

ORDER BY release\_year, total DESC limit 20;



**-- Top 5 genre with most recommendations**

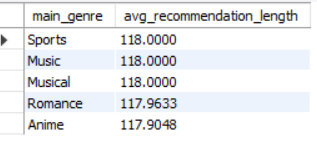
SELECT main\_genre, AVG(LENGTH(recommendation)) AS avg\_recommendation\_length

FROM netflix\_data\_1

GROUP BY main\_genre

ORDER BY avg\_recommendation\_length DESC

LIMIT 5;



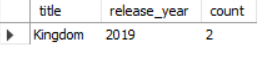
**-- Finding duplicate titles**

SELECT title, release\_year, COUNT(\*) AS count

FROM netflix\_data\_1

GROUP BY title, release\_year

HAVING COUNT(\*) > 1;



**-- Finding which rows are duplicated using CTE**

WITH duplicates AS (

SELECT title, release\_year

FROM netflix\_data\_1

GROUP BY title, release\_year

HAVING COUNT(\*) > 1

)

SELECT \*

FROM netflix\_data\_1

WHERE (title, release\_year) IN (

SELECT title, release\_year FROM duplicates

);

