**SQL CLASS ASSIGNMENT**

Question #1: What is the movie with the highest average\_rating of the year 2022- in case of same average\_rating give the one with more num\_votes - with more than 100,000 num\_votes ?

SELECT

primary\_title,

num\_votes,

MAX(average\_rating) as avg\_rating

FROM

`awesome-chess-369408.imdb.title\_basics` tb

LEFT JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tr.tconst=tb.tconst

WHERE

(tr.num\_votes>100000)

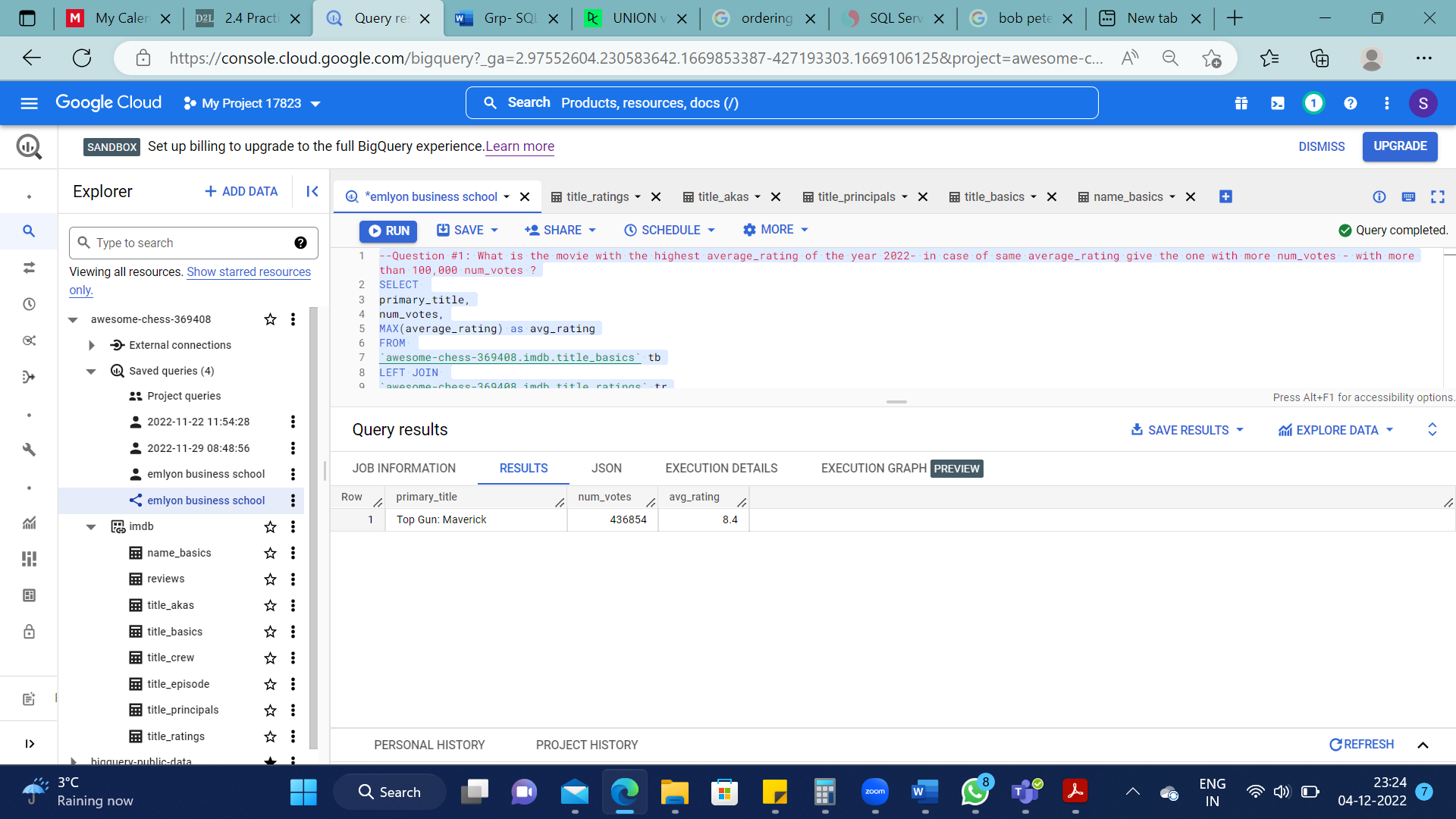
AND (tb.start\_year = 2022)

AND (tb.title\_type= 'movie')

GROUP BY primary\_title, num\_votes

ORDER BY MAX(average\_rating) DESC, num\_votes DESC

LIMIT 1



Question #2: In order to have 1 metric to find the best movies, we will use average\_rating \* num\_votes that we will call rating\_score - What are the top 5 movies with the highest rating\_score ever ?

SELECT

primary\_title, (average\_rating \*num\_votes) as rating\_score

FROM

`awesome-chess-369408.imdb.title\_basics` tb

LEFT JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

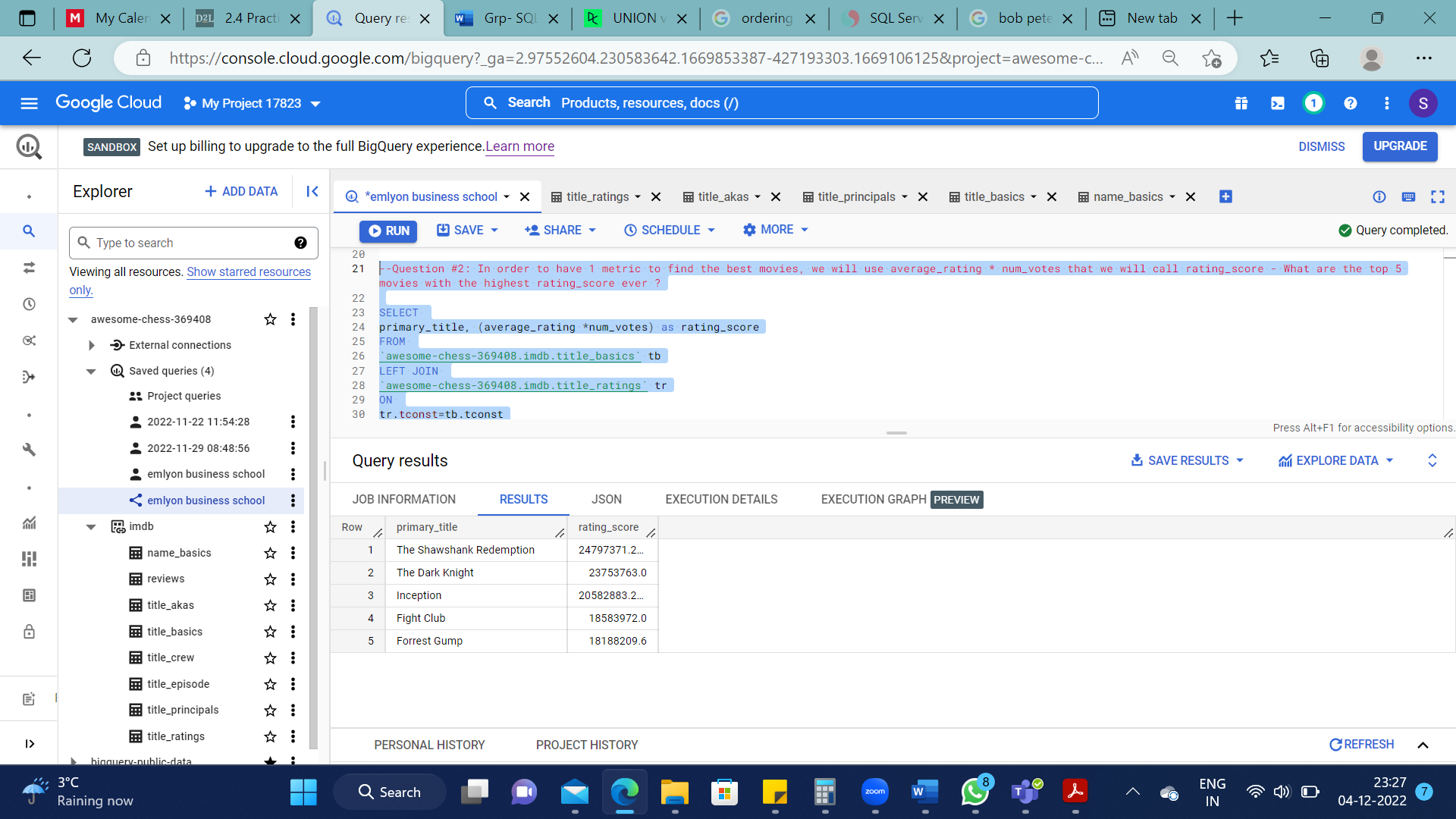
ON

tr.tconst=tb.tconst

WHERE title\_type = 'movie'

ORDER BY rating\_score DESC

LIMIT 5



Question #3: Get a query to have for each year, the number of movies released, the highest rating\_score for the year and the average of average\_rating for the year as well as the sum of num\_votes - order by year descending

SELECT

start\_year,

COUNT(primary\_title) as number\_of\_movies,

MAX(average\_rating \*num\_votes) as highest\_rating\_score,

AVG(average\_rating) as avg\_rating,

SUM(num\_votes) as sum\_of\_votes

FROM

`awesome-chess-369408.imdb.title\_basics` tb

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

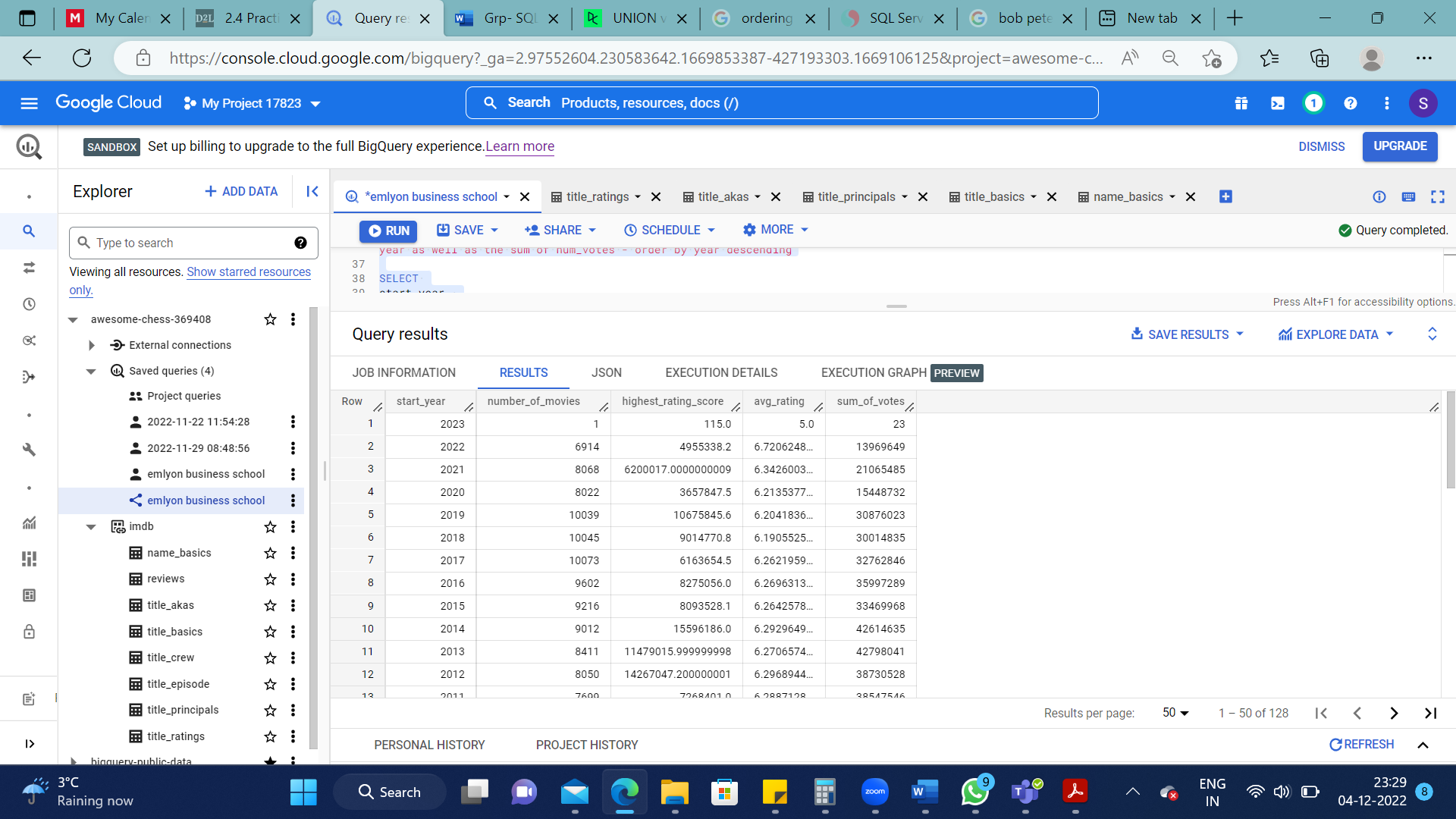
ON

tr.tconst=tb.tconst

WHERE title\_type= 'movie'

GROUP BY start\_year

ORDER BY start\_year DESC



Question #4: Who is the actor who played in movies that has the biggest sum of rating\_score - provide also his average of average\_rating and his number of movies ??

SELECT

primary\_name, COUNT (primary\_title) as count\_titles, SUM(num\_votes\*average\_rating) as rating\_score, AVG(average\_rating) as avg\_of\_avg

FROM

`awesome-chess-369408.imdb.name\_basics` nb

JOIN

`awesome-chess-369408.imdb.title\_principals` tp

ON

nb.nconst = tp.nconst

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tp.tconst = tr.tconst

JOIN

`awesome-chess-369408.imdb.title\_basics` tb

ON

tb.tconst = tr.tconst

WHERE

category = 'actor'

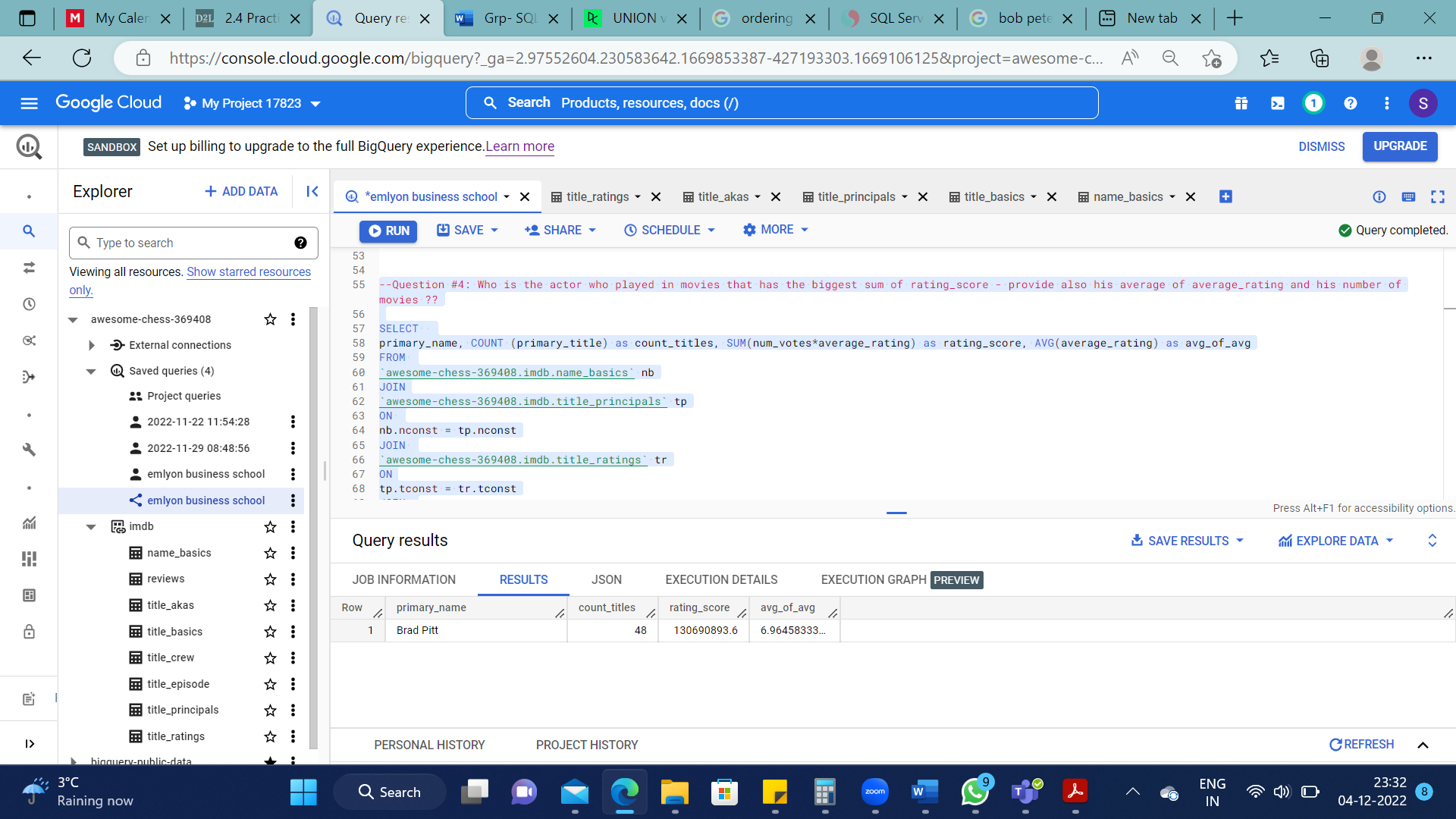
AND

title\_type = 'movie'

GROUP BY primary\_name

ORDER BY SUM(num\_votes\*average\_rating) DESC

LIMIT 1



Question #5: What are the top 3 movies with highest rating\_score for the actor found above

SELECT

primary\_title, (num\_votes\*average\_rating) as rating\_score

FROM

`awesome-chess-369408.imdb.name\_basics` nb

LEFT JOIN

`awesome-chess-369408.imdb.title\_principals` tp

ON

nb.nconst = tp.nconst

LEFT JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tp.tconst = tr.tconst

LEFT JOIN

`awesome-chess-369408.imdb.title\_basics` tb

ON

tb.tconst = tr.tconst

WHERE

primary\_name = 'Brad Pitt'

AND

title\_type = 'movie'

ORDER BY (num\_votes\*average\_rating) DESC

LIMIT 3

--Complex version

SELECT

primary\_title

FROM

`awesome-chess-369408.imdb.name\_basics` nb

JOIN

`awesome-chess-369408.imdb.title\_principals` tp

ON nb.nconst= tp.nconst

JOIN

`awesome-chess-369408.imdb.title\_basics` tb

ON tb.tconst = tp.tconst

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON tr.tconst = tb.tconst

WHERE

primary\_name = (SELECT

primary\_name

FROM

`awesome-chess-369408.imdb.name\_basics` nb

JOIN

`awesome-chess-369408.imdb.title\_principals` tp

ON

nb.nconst = tp.nconst

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tp.tconst = tr.tconst

JOIN

`awesome-chess-369408.imdb.title\_basics` tb

ON

tb.tconst = tr.tconst

WHERE

category = 'actor'

AND

title\_type = 'movie'

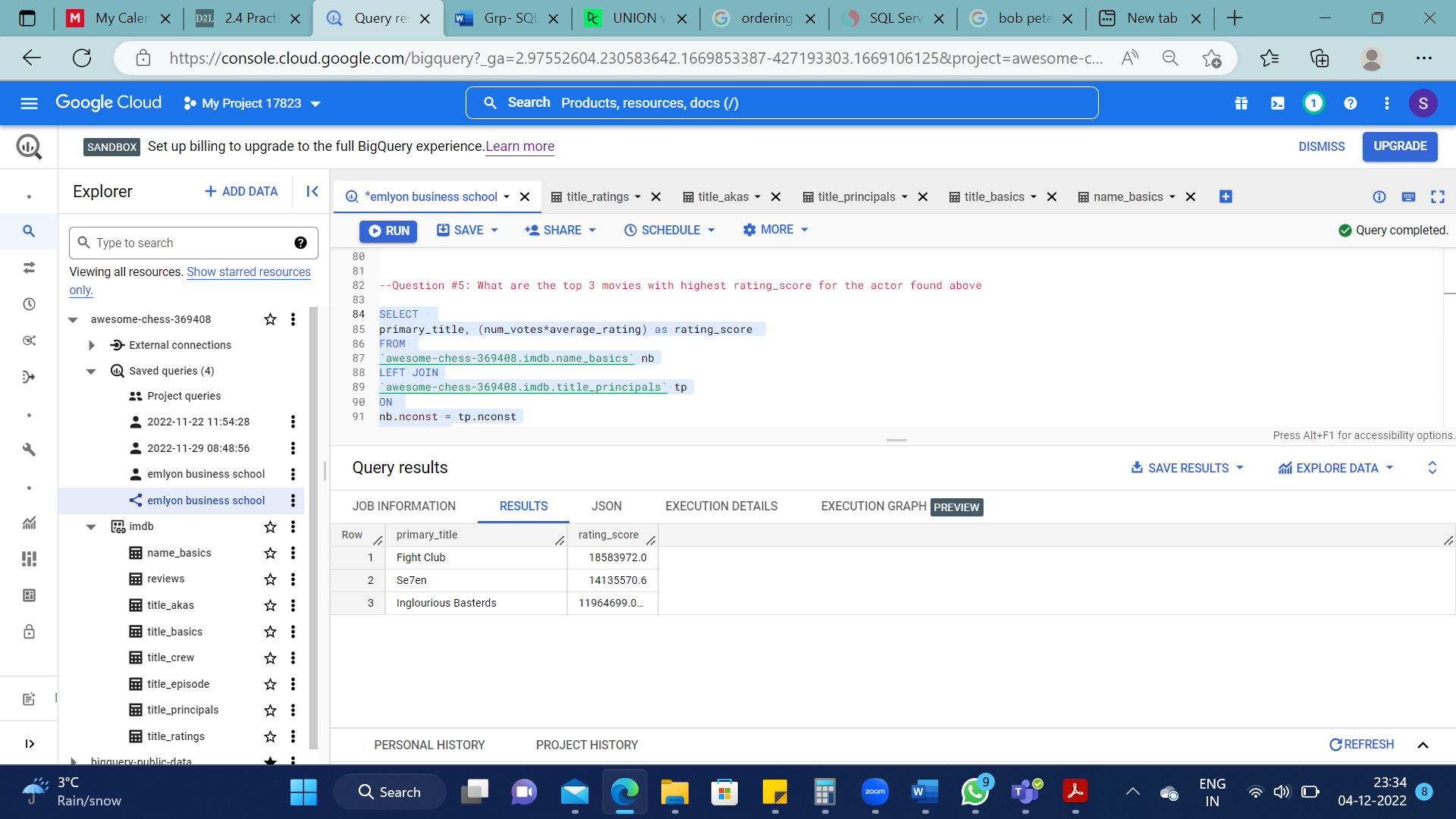
GROUP BY primary\_name

ORDER BY SUM(num\_votes\*average\_rating) DESC

LIMIT 1)

ORDER BY (num\_votes\*average\_rating) DESC

LIMIT 3



Question #6: Who is the actor who played in at least 5 movies with the highest average rating\_score per movie (what is his average rating\_score )?

SELECT

primary\_name, COUNT(DISTINCT primary\_title) AS title\_count, AVG(num\_votes\*average\_rating)

AS avg\_rating\_score

FROM

`awesome-chess-369408.imdb.name\_basics` nb

JOIN

`awesome-chess-369408.imdb.title\_principals` tp

ON

nb.nconst = tp.nconst

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tp.tconst = tr.tconst

JOIN

`awesome-chess-369408.imdb.title\_basics` tb

ON

tb.tconst = tr.tconst

WHERE

title\_type = 'movie'

AND

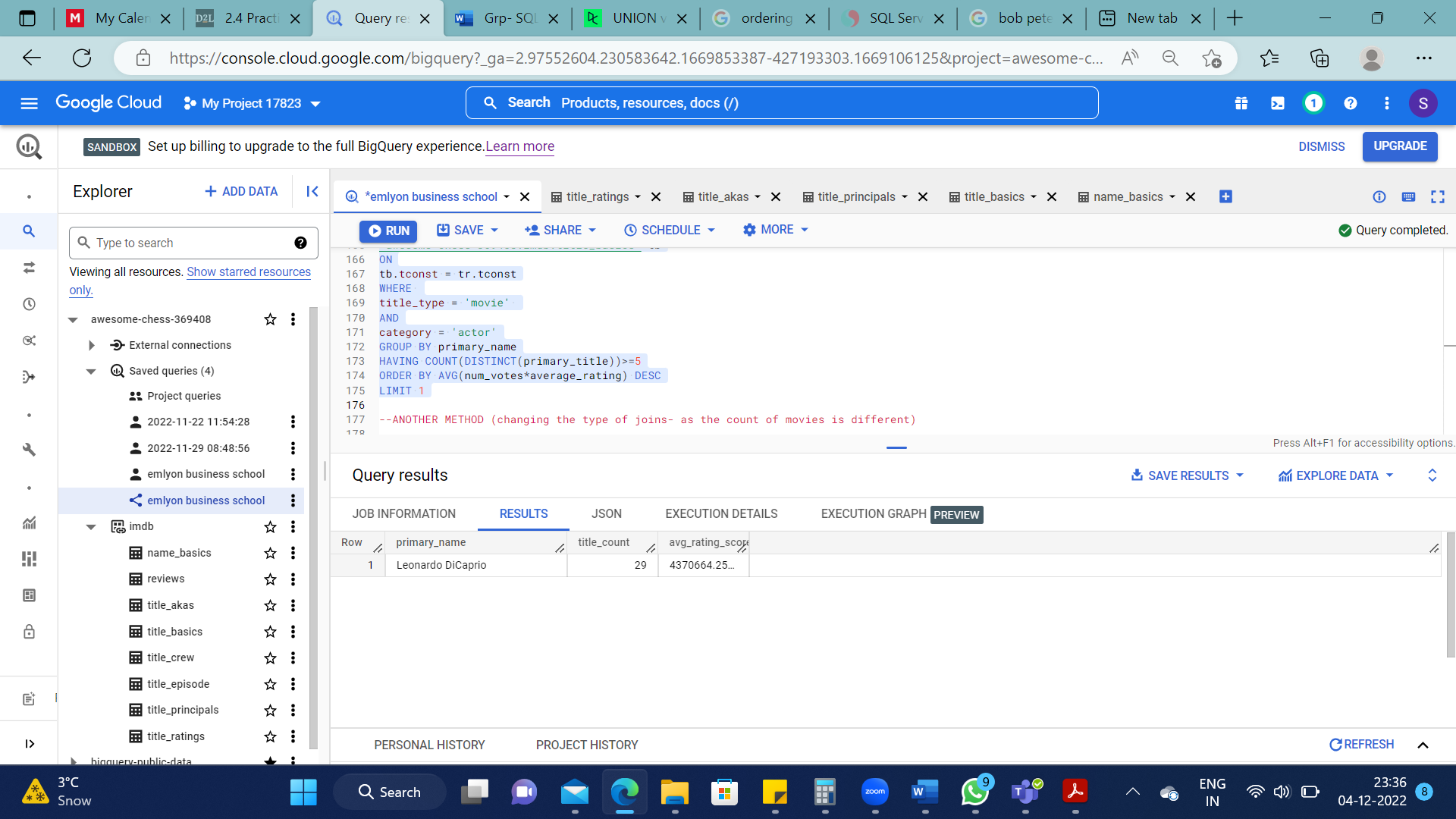
category = 'actor'

GROUP BY primary\_name

HAVING COUNT(DISTINCT(primary\_title))>=5

ORDER BY AVG(num\_votes\*average\_rating) DESC

LIMIT 1



ANOTHER METHOD (changing the type of joins- as the count of movies is different)

SELECT

primary\_name, COUNT(DISTINCT primary\_title) AS title\_count, AVG(num\_votes\*average\_rating) as avg\_rating\_score

FROM

`awesome-chess-369408.imdb.title\_basics` tb

LEFT JOIN

`awesome-chess-369408.imdb.title\_principals` tp

ON

tb.tconst = tp.tconst

LEFT JOIN

`awesome-chess-369408.imdb.name\_basics` nb

ON

nb.nconst = tp.nconst

LEFT JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tp.tconst = tr.tconst

WHERE

title\_type = 'movie'

AND

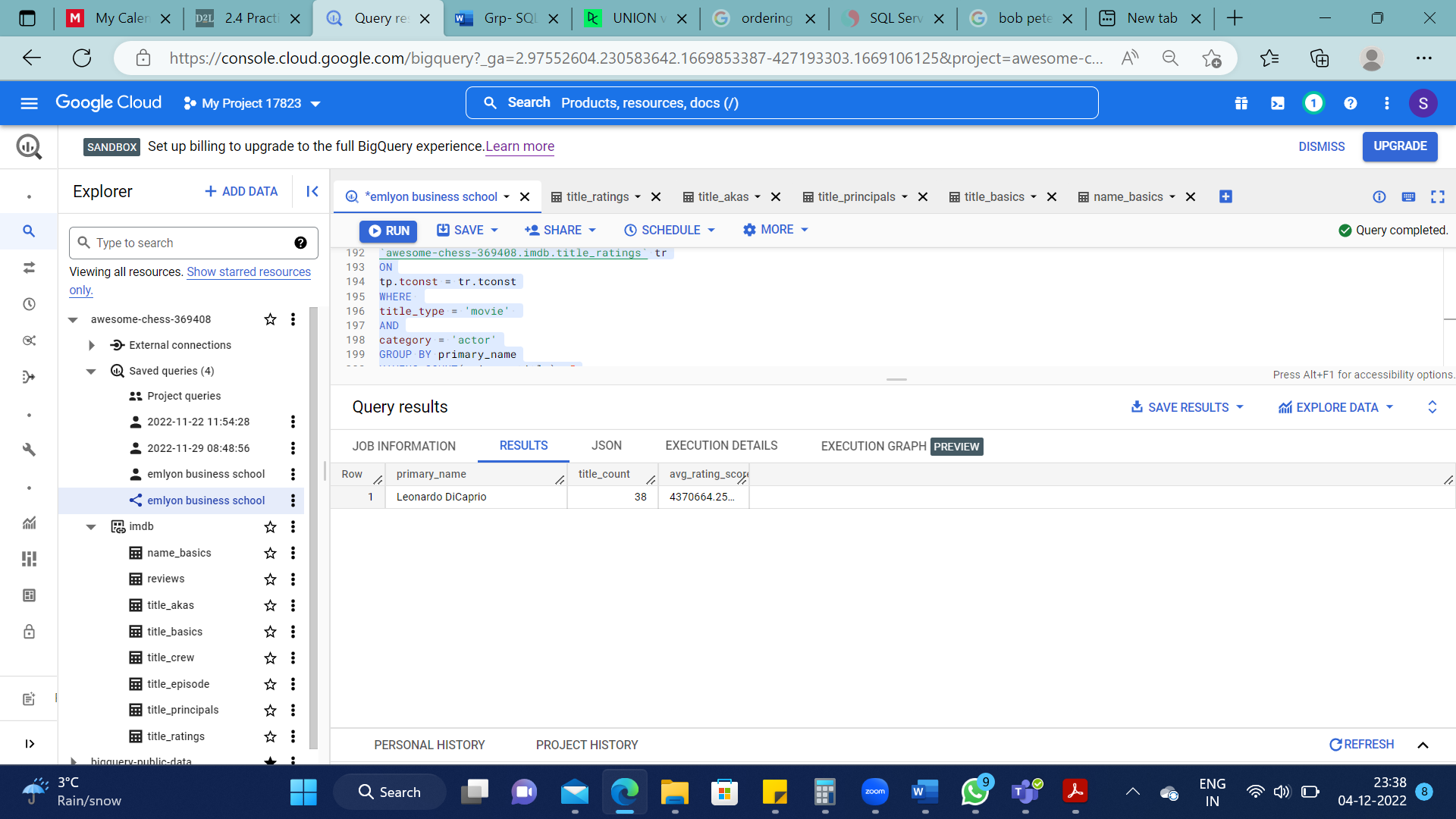
category = 'actor'

GROUP BY primary\_name

HAVING COUNT(primary\_title)>=5

ORDER BY AVG(num\_votes\*average\_rating) DESC

LIMIT 1



Question #7: Create a Query to get the top movie (highest rating\_score) for each year

SELECT

primary\_title,

(average\_rating\*num\_votes) AS rating\_score,

X.start\_year

FROM

`awesome-chess-369408.imdb.title\_basics` tb

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON tb.tconst = tr.tconst

JOIN

(SELECT

start\_year,

MAX(average\_rating \*num\_votes) as highest\_rating\_score

FROM

`awesome-chess-369408.imdb.title\_basics` tb

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tr.tconst=tb.tconst

WHERE title\_type ='movie'

GROUP BY start\_year

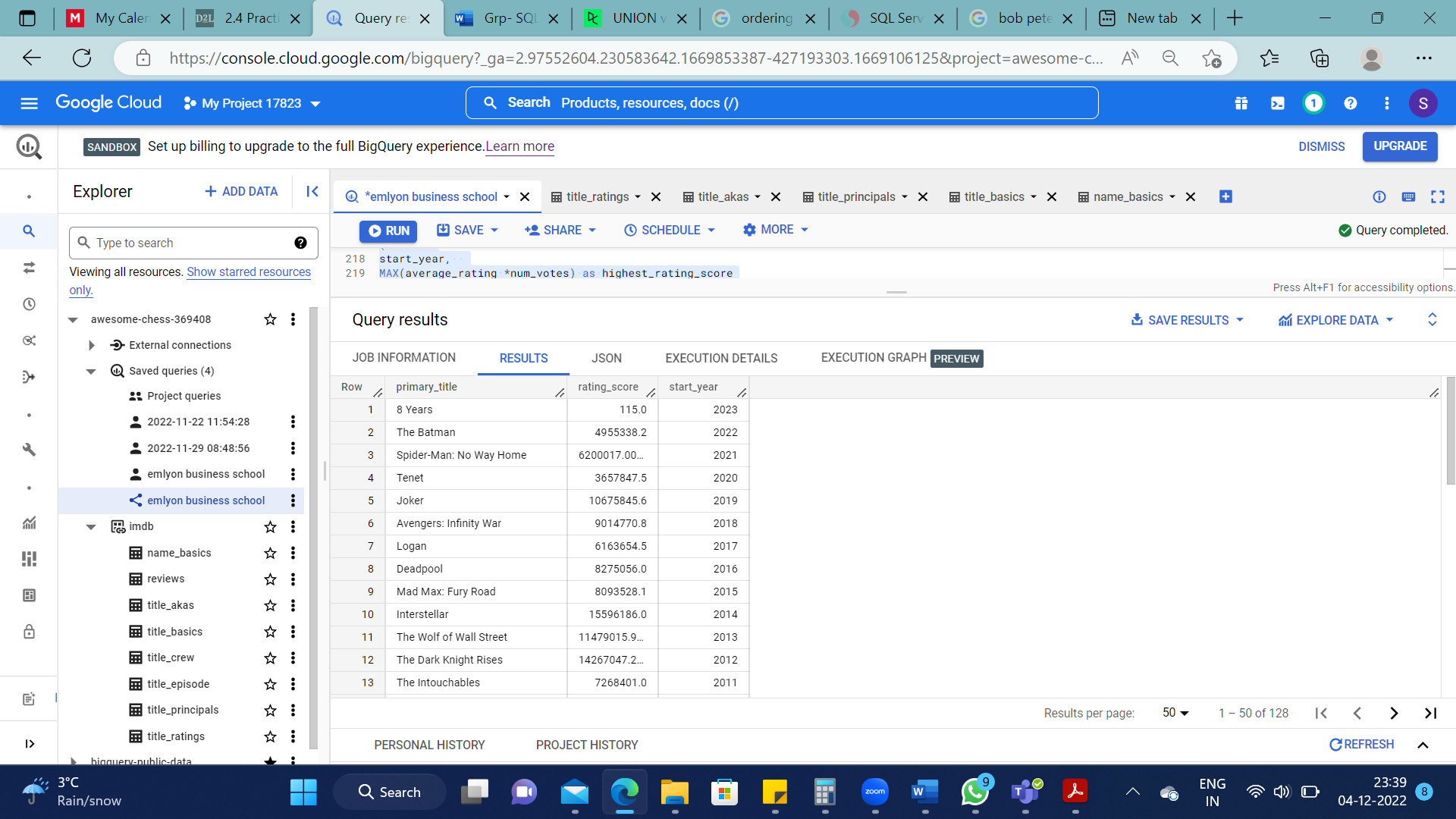
ORDER BY start\_year DESC) X

ON tb.start\_year = X.start\_year

WHERE

(average\_rating\*num\_votes) = X.highest\_rating\_score

ORDER BY 3 DESC



Question #8: For each Movie Genre, for the release since 2000, give the movie title with the highest rating\_score

SELECT

tb.genres,

primary\_title,

start\_year,

(average\_rating\*num\_votes) AS rating\_score

FROM

`awesome-chess-369408.imdb.title\_basics` tb

LEFT JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tb.tconst=tr.tconst

LEFT JOIN

(SELECT

genres,

MAX(average\_rating \*num\_votes) as highest\_rating\_score

FROM

`awesome-chess-369408.imdb.title\_basics` tb

LEFT JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

ON

tr.tconst=tb.tconst

GROUP BY genres

ORDER BY genres) X

ON tb.genres = X.genres

WHERE

(average\_rating\*num\_votes) = X.highest\_rating\_score

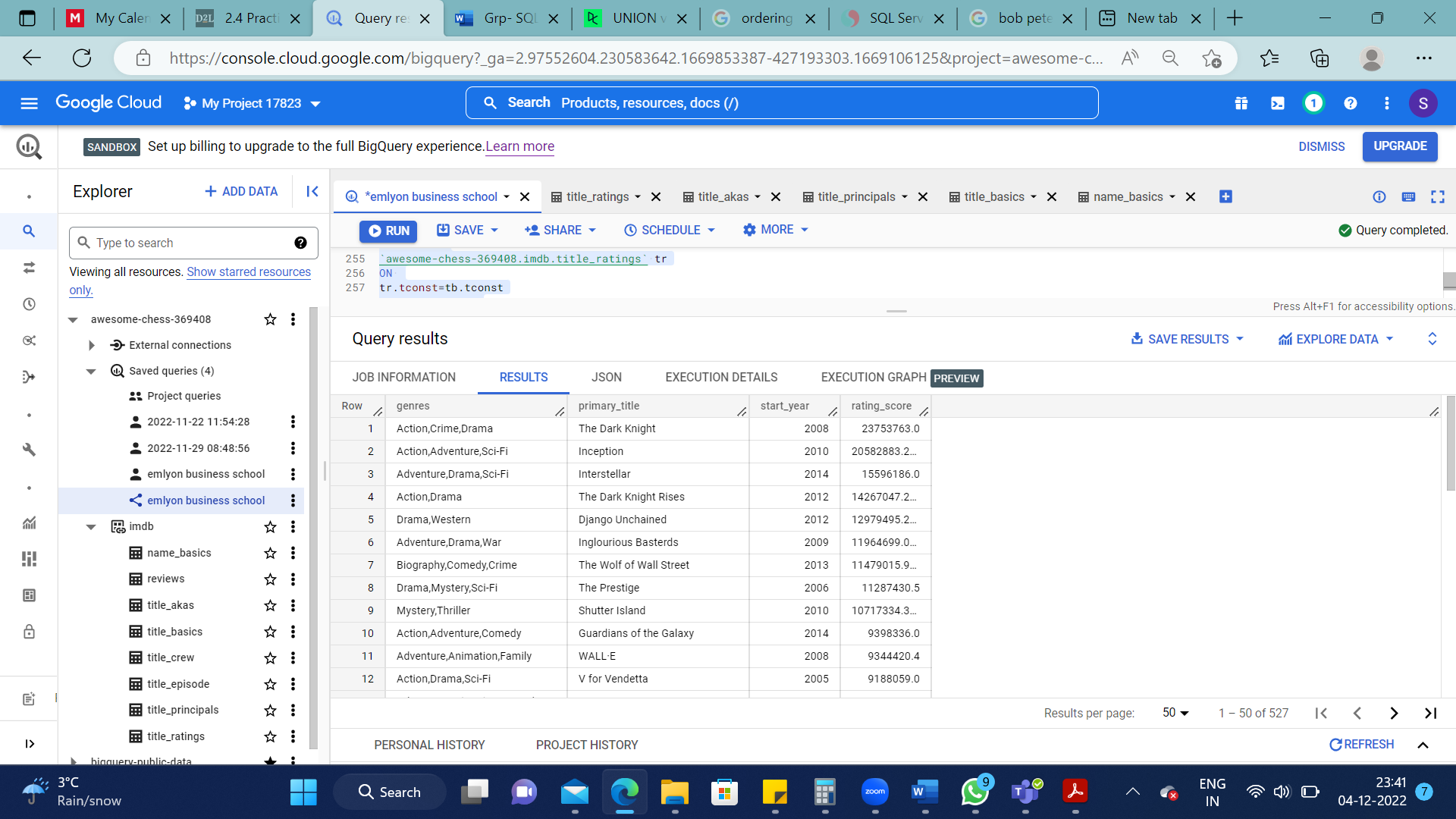
AND

start\_year>=2000

AND

title\_type = 'movie'

ORDER BY X.highest\_rating\_score DESC



Question #9: Duo? Find the actors duo that get the highest average rating\_score per movie together in ordering 1 or 2, with at least 4 movies together

WITH actor\_duo AS

(SELECT

primary\_name, tb.primary\_title, tp.tconst

FROM

`awesome-chess-369408.imdb.name\_basics` nb

JOIN

`awesome-chess-369408.imdb.title\_principals` tp

ON nb.nconst = tp.nconst

JOIN

`awesome-chess-369408.imdb.title\_basics` tb

ON tp.tconst = tb.tconst

WHERE tp.ordering<=2

AND title\_type LIKE '%movie%'

AND category = 'actor')

SELECT

ad.primary\_name,

ad1.primary\_name,

AVG(average\_rating\*num\_votes) AS rating\_score,

COUNT(ad.primary\_title) AS movie\_count

FROM actor\_duo ad

JOIN actor\_duo ad1

ON ad.primary\_name > ad1.primary\_name

AND

ad.primary\_title = ad1.primary\_title

JOIN

`awesome-chess-369408.imdb.title\_ratings` tr

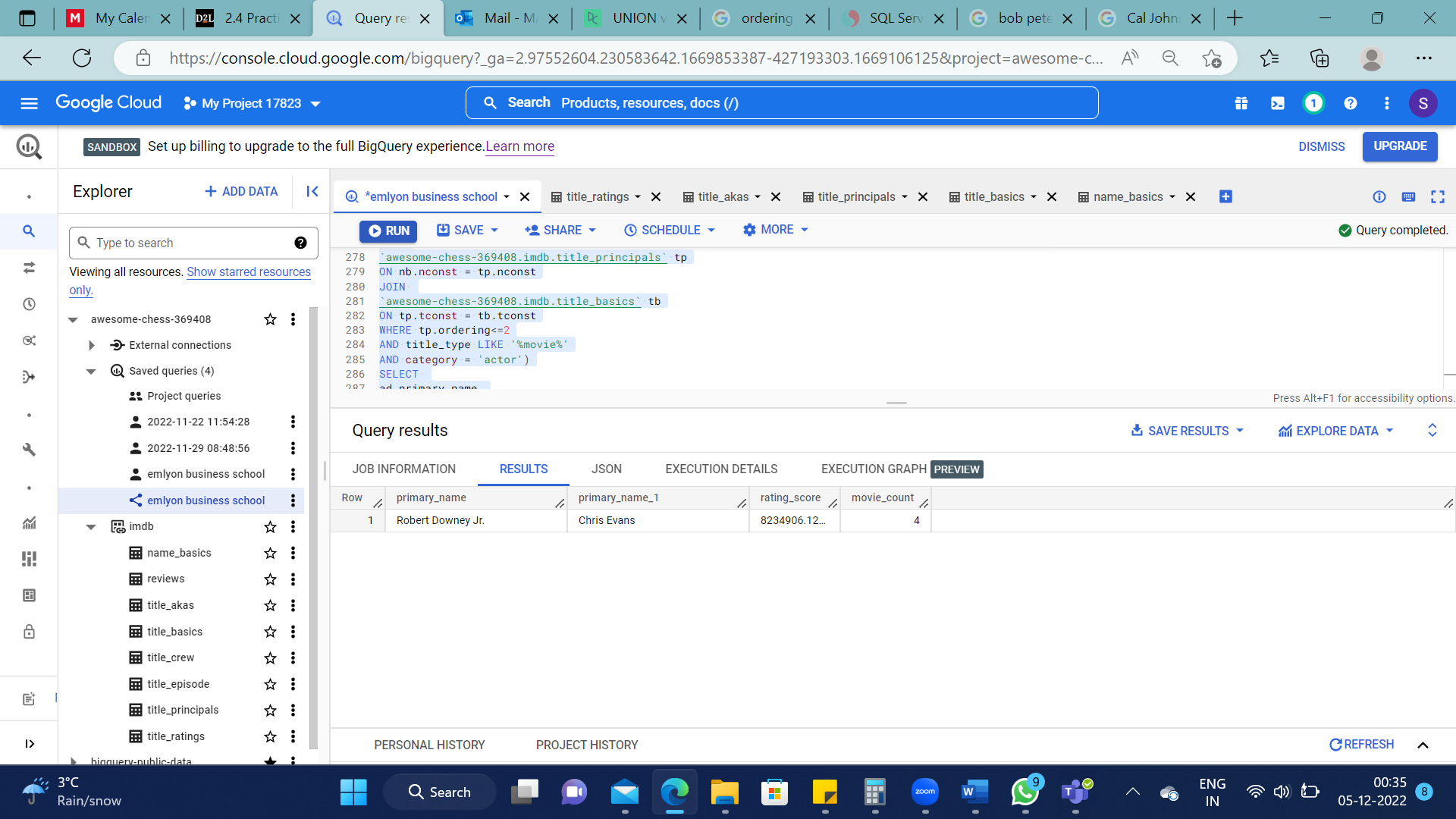
ON ad.tconst = tr.tconst

GROUP BY 1,2

HAVING COUNT(ad.primary\_title) >=4

ORDER BY rating\_score DESC

LIMIT 1



Question #10: Extra - Bonus question if you manage a smart way to show 2 example of their movies together and then to limit the cases where the movies titles very similar looking like they belong to a sequel…

WITH actor\_duo AS

(SELECT primary\_name,

primary\_title

FROM `awesome-chess-369408.imdb.name\_basics` nb

JOIN `awesome-chess-369408.imdb.title\_principals` tp

ON nb.nconst = tp.nconst

JOIN `awesome-chess-369408.imdb.title\_basics` tb

ON tp.tconst = tb.tconst

WHERE

title\_type LIKE '%movie%'

AND

category = 'actor')

SELECT

ad.primary\_title,

FROM actor\_duo ad

JOIN actor\_duo ad1

ON ad.primary\_title = ad1.primary\_title

WHERE

ad.primary\_name = 'Robert Downey Jr.'

AND

ad1.primary\_name = 'Chris Evans'

AND

ad.primary\_title LIKE '%Avengers%'

LIMIT 2

