1. **Investigación:**
   1. **Null:**
      1. **¿Qué significa?**

**Null** (nulo) es un marcador especial usado en el lenguaje de consulta estructurado (SQL) para indicar que no existe un valor dentro de una base de datos. pueden representar información “desconocida” o “no aplicable”.

* + 1. **¿Resultado de operarlo con los diferentes tipos de operadores:aritméticos, lógicos y de comparación?**

Cuando el null es operado con **aritmetica** el valor que retorna es null debido a que null es diferente de 0 y es un valor desconocido.

Para operaciones logicas el resultado depende de la operación realizada con OR cuando sea null or false es igual a null, cuando sea null or true por identidad es true cuando sea null or null el resultado es null por idempotencia.para el caso con el and cuando es null and false el resultado es false null and true es null y null and null es null.

* 1. **Junta:**
     1. **¿ Cuáles son las diferencias entre la junta interna y la junta externa?**

La junta inetrna nos muestra los valores de dos tablas que tengan relacion y la junta interna nos muestra todos los valores de cada tabla pero si es una junta externa completa los valores que no tienen relacion con la otra tabla se vuelven nulos.

* + 1. ¿Qué opciones se tienen para la junta externa?

Se tienen tres opciones la primera es el left join que une la primera tabla con la segunda y los valores de la segunda que no esten en la primera se vuelven nulos;para el rigth join es de la segunda tabla a la primera y el full join vuelve nulos los valores que no se relacionen de primera a segunda y de segunda a tercera.

* + 1. ¿Qué opciones se tienen para la junta interna?

Para la junta interna se tienen tres opciones el inner join que une la pk de una tabla con la fk de otra, el natural join para datos ambiguos los combina y queda una sola tabla, using clause que es mas eficiente que el natural join se escribe el atributo que se repite en las dos tablas.

1. **PRACTICA**
   1. **JOIN**
      1. SELECT matchid,player FROM goal

WHERE teamid='GER'

* + 1. SELECT distinct id,stadium,team1,team2

FROM game join goal on matchid=id

where id='1012'

* + 1. SELECT player,teamid,stadium,mdate

FROM game JOIN goal ON (id=matchid)

where teamid='GER'

* + 1. select team1,team2,player

from game join goal on (id=matchid)

where player like 'Mario%'

* + 1. SELECT player, teamid,coach,gtime

FROM goal join eteam on teamid=id

WHERE gtime<=10

* + 1. select mdate,teamname

from game JOIN eteam ON (team1=eteam.id)

where coach='Fernando Santos'

* + 1. select player from game join goal on id=matchid

where stadium= 'National Stadium, Warsaw'

* + 1. SELECT distinct player

FROM game JOIN goal ON matchid = id

where teamid!='GER' and (team1='GER' or team2='GER')

* + 1. SELECT teamname,count(gtime)

FROM eteam JOIN goal ON id=teamid

group by teamname

* + 1. select stadium,count(gtime)

from game join goal on id=matchid

group by stadium

* + 1. SELECT matchid,mdate,count(gtime)

FROM game JOIN goal ON matchid = id

WHERE (team1 = 'POL' OR team2 = 'POL')

group by mdate,matched

* + 1. select matchid,mdate,count(gtime)

from game join goal on id=matchid

where teamid='GER'

group by mdate,matched

* + 1. pendiente
  1. **Other join**
     1. SELECT id, title

FROM movie

WHERE yr=1962

* + 1. select yr from movie

where title='Citizen Kane'

* + 1. select id,title,yr from movie

where title like 'Star Trek%'

order by yr

* + 1. select id from actor

where name='Glenn Close'

* + 1. select distinct movieid from casting join movie on movieid=id

where title='Casablanca'

* + 1. select name from actor join casting on id=actorid

where movieid=11768

* + 1. select name from (movie join casting on movie.id=movieid) join actor on actorid=actor.id

where title='Alien'

* + 1. select title from (movie join casting on movie.id=movieid) join actor on actorid=actor.id

where name='Harrison Ford'

* + 1. select title from (movie join casting on movie.id=movieid) join actor on actorid=actor.id

where name='Harrison Ford' and ord!=1

* + 1. select distinct title,name from (movie join casting on movie.id=movieid) join actor on actorid=actor.id

where yr=1962 and ord=1

* + 1. SELECT yr,COUNT(title) FROM

movie JOIN casting ON movie.id=movieid

JOIN actor ON actorid=actor.id

WHERE name='Rock Hudson'

GROUP BY yr

HAVING COUNT(title) > 2

* + 1. SELECT title, actor.name FROM actor

JOIN casting ON (id = actorid)

JOIN movie ON (movieid = movie.id)

WHERE movieid IN

(SELECT movieid FROM casting

WHERE actorid = 179)

AND casting.ord = 1

* + 1. select name from (movie join casting on movie.id=movieid) join actor on actorid=actor.id

where ord=1

group by name

having count(ord)>=30

* + 1. select title,count(actorid) as perra from (movie join casting on movie.id=movieid) join actor on actorid=actor.id

where yr=1978

group by title

order by perra desc,title

* + 1. select name from (movie join casting on movie.id=movieid) join actor on actorid=actor.id

where name!='Art Garfunkel' and (movieid=10095 or movieid=11434 or movieid=13630)

* 1. **NULL**
     1. select name from teacher

where dept is null

* + 1. SELECT teacher.name, dept.name

FROM teacher INNER JOIN dept

ON (teacher.dept=dept.id)

* + 1. select teacher.name,dept.name from teacher left join dept on dept=dept.id
    2. select teacher.name,dept.name from teacher right join dept on dept=dept.id
    3. select name, coalesce(mobile,'07986 444 2266') from teacher
    4. select teacher.name,coalesce(dept.name,'None') from teacher left join dept on dept=dept.id
    5. select count(teacher.id),count(mobile) from teacher}
    6. select dept.name,count(teacher.id)from teacher right join dept on dept=dept.id

group by dept.name

* + 1. select name,Case when dept=1 then 'Sci'

when dept=2 then 'Sci'

else 'Art'

END

from teacher

* + 1. select name,Case when dept=1 then 'Sci'

when dept=2 then 'Sci'

when dept=3 then 'Art'

else 'None'

END

from teacher

* 1. **NUMERIC EXAMPLES**
     1. Select A\_STRONGLY\_AGREE

FROM nss

WHERE question='Q01'

AND institution='Edinburgh Napier University'

AND subject='(8) Computer Science'

* + 1. SELECT institution,subject

FROM nss

WHERE question='Q15' and score>=100

* + 1. SELECT institution,score

FROM nss

WHERE question='Q15'

and score<50

AND subject='(8) Computer Science'

* + 1. SELECT subject,sum(response)

FROM nss

WHERE question='Q22'

AND (subject='(8) Computer Science' or subject='(H) Creative Arts and Design')

group by subject

* + 1. SELECT subject,sum(response\*A\_STRONGLY\_AGREE/100)

FROM nss

WHERE question='Q22'

AND (subject='(8) Computer Science' or subject='(H) Creative Arts and Design')

group by subject

* + 1. SELECTsubject,round(sum(response\*A\_STRONGLY\_AGREE)/sum(response),0)

FROM nss

WHERE question='Q22'

AND (subject='(8) Computer Science' or subject='(H) Creative Arts and Design')

group by subject

* + 1. SELECT institution,round(sum(score\*response)/sum(response),0)

FROM nss

WHERE question='Q22'

AND (institution LIKE '%Manchester%')

group by institution

ORDER BY institution

* + 1. SELECT institution,sum(sample),sum(case when subject='(8) Computer Science' then sample else 0 end)

FROM nss

WHERE question='Q01'

AND (institution LIKE '%Manchester%')

group by institution

* 1. **SELF JOIN**
     1. select count(id) from stops
     2. select id from stops

where name='Craiglockhart'

* + 1. select id,name from stops join route on id=stop

where num='4' and company='LRT'

* + 1. SELECT company, num, COUNT(\*)

FROM route WHERE stop=149 OR stop=53

GROUP BY company, num

having count(\*)=2

* + 1. SELECT a.company, a.num, a.stop, b.stop

FROM route a JOIN route b ON

(a.company=b.company AND a.num=b.num)

WHERE a.stop=53 and b.stop=149

* + 1. SELECT a.company, a.num, stopa.name, stopb.name

FROM route a JOIN route b ON

(a.company=b.company AND a.num=b.num)

JOIN stops stopa ON (a.stop=stopa.id)

JOIN stops stopb ON (b.stop=stopb.id)

WHERE stopa.name='Craiglockhart' and stopb.name='London Road'

* + 1. SELECT distinct a.company, a.num

FROM route a JOIN route b ON

(a.company=b.company AND a.num=b.num)

JOIN stops stopa ON (a.stop=stopa.id)

JOIN stops stopb ON (b.stop=stopb.id)

WHERE a.stop=115 and b.stop=137

* + 1. SELECT distinct a.company, a.num

FROM route a JOIN route b ON

(a.company=b.company AND a.num=b.num)

JOIN stops stopa ON (a.stop=stopa.id)

JOIN stops stopb ON (b.stop=stopb.id)

WHERE stopa.name='Craiglockhart' and stopb.name='Tollcross'

* + 1. SELECT name,company, num

FROM (SELECT X.num,stop,company FROM route JOIN

(SELECT num FROM stops JOIN route ON id= stop

WHERE company= 'LRT' AND name= 'Craiglockhart')

AS X ON X.num= route.num ) AS Z JOIN stops

WHERE id= stop AND company= 'LRT'

* + 1. SELECT DISTINCT a.num, a.company,stopb.name , c.num,

c.company

FROM route a JOIN route b ON (a.company = b.company AND

a.num = b.num)

JOIN ( route c JOIN route d ON (c.company = d.company AND

c.num= d.num))

JOIN stops stopa ON (a.stop = stopa.id) JOIN stops stopb ON

(b.stop = stopb.id)

JOIN stops stopc ON (c.stop = stopc.id) JOIN stops stopd ON

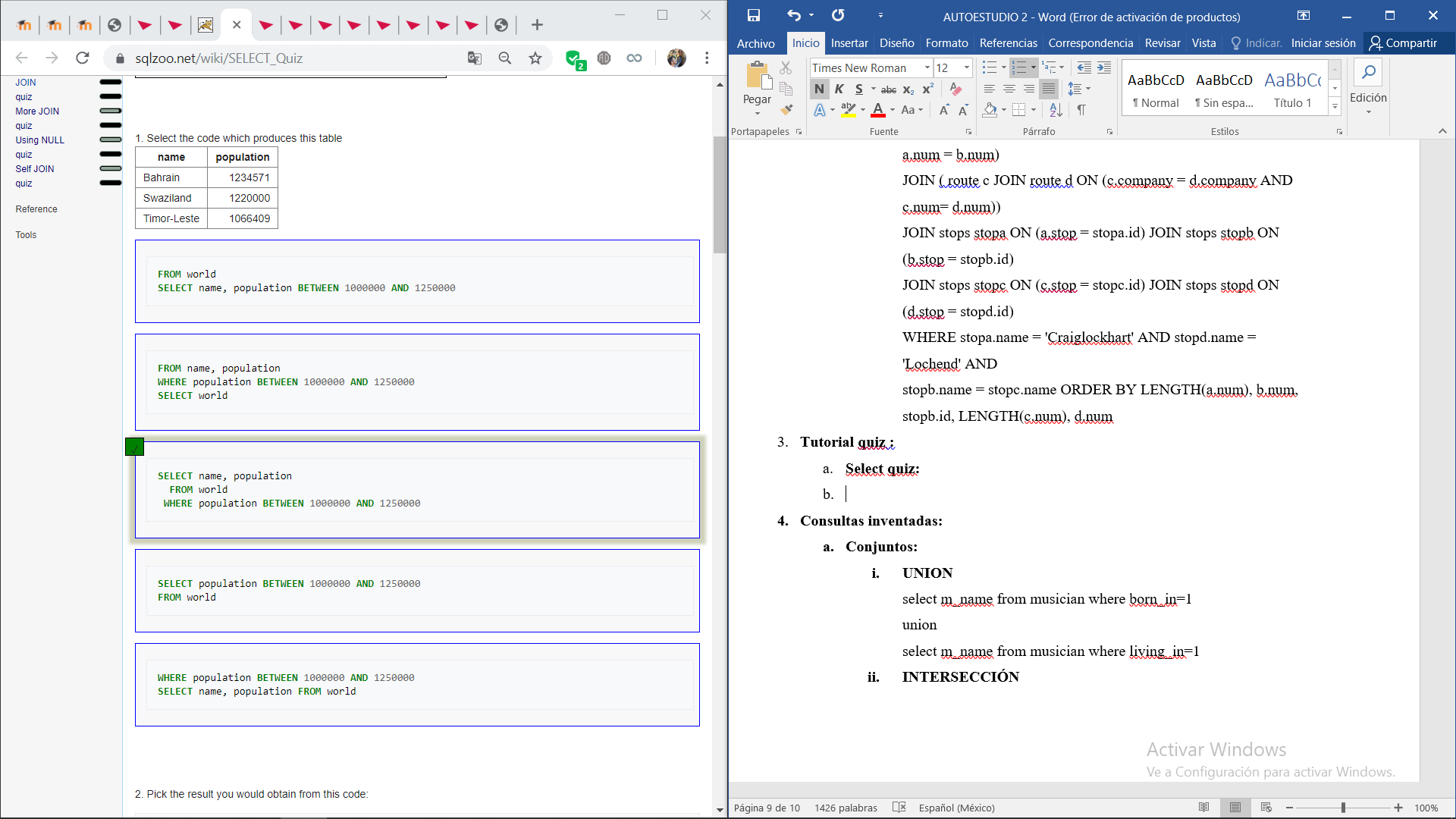
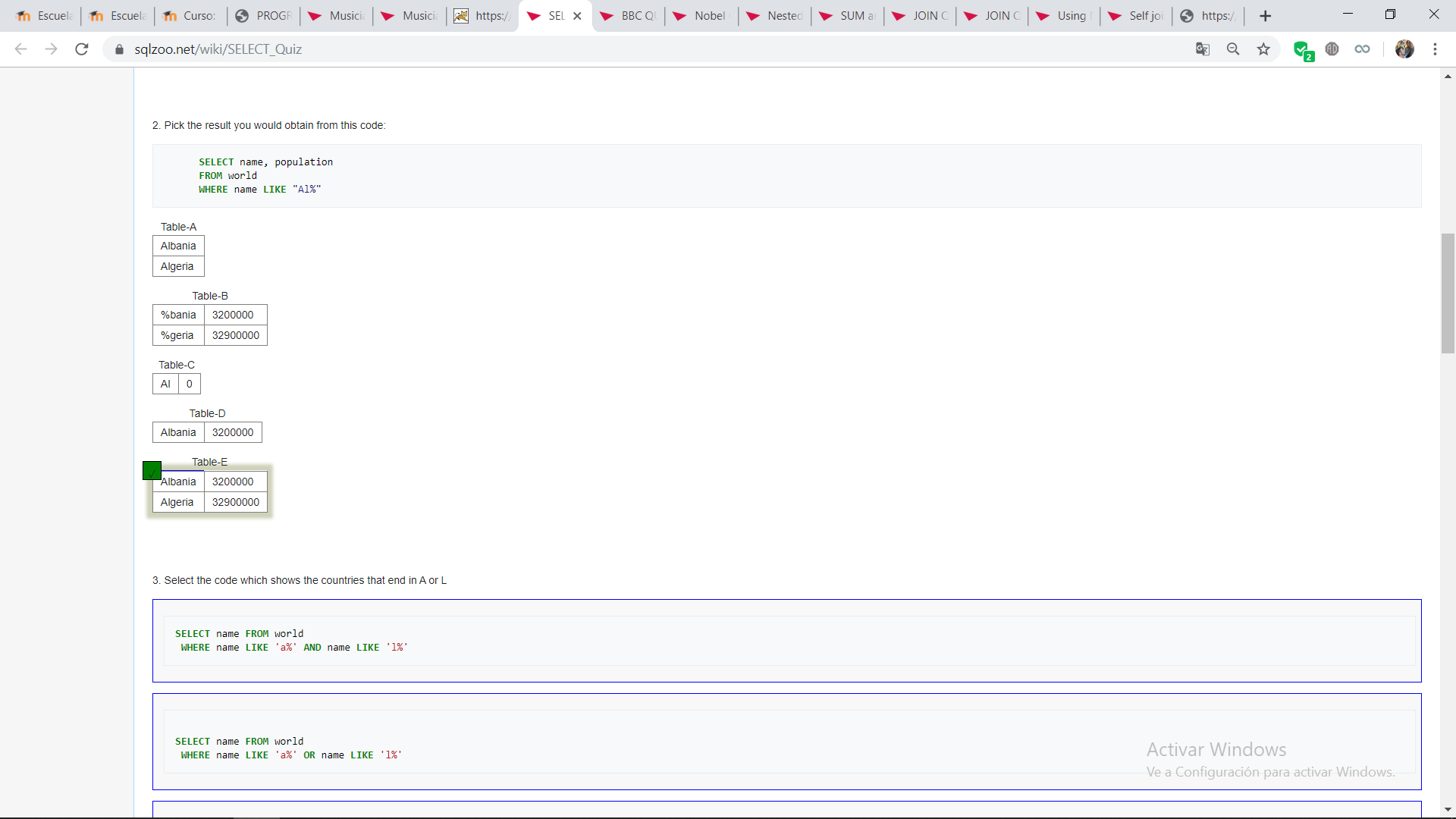
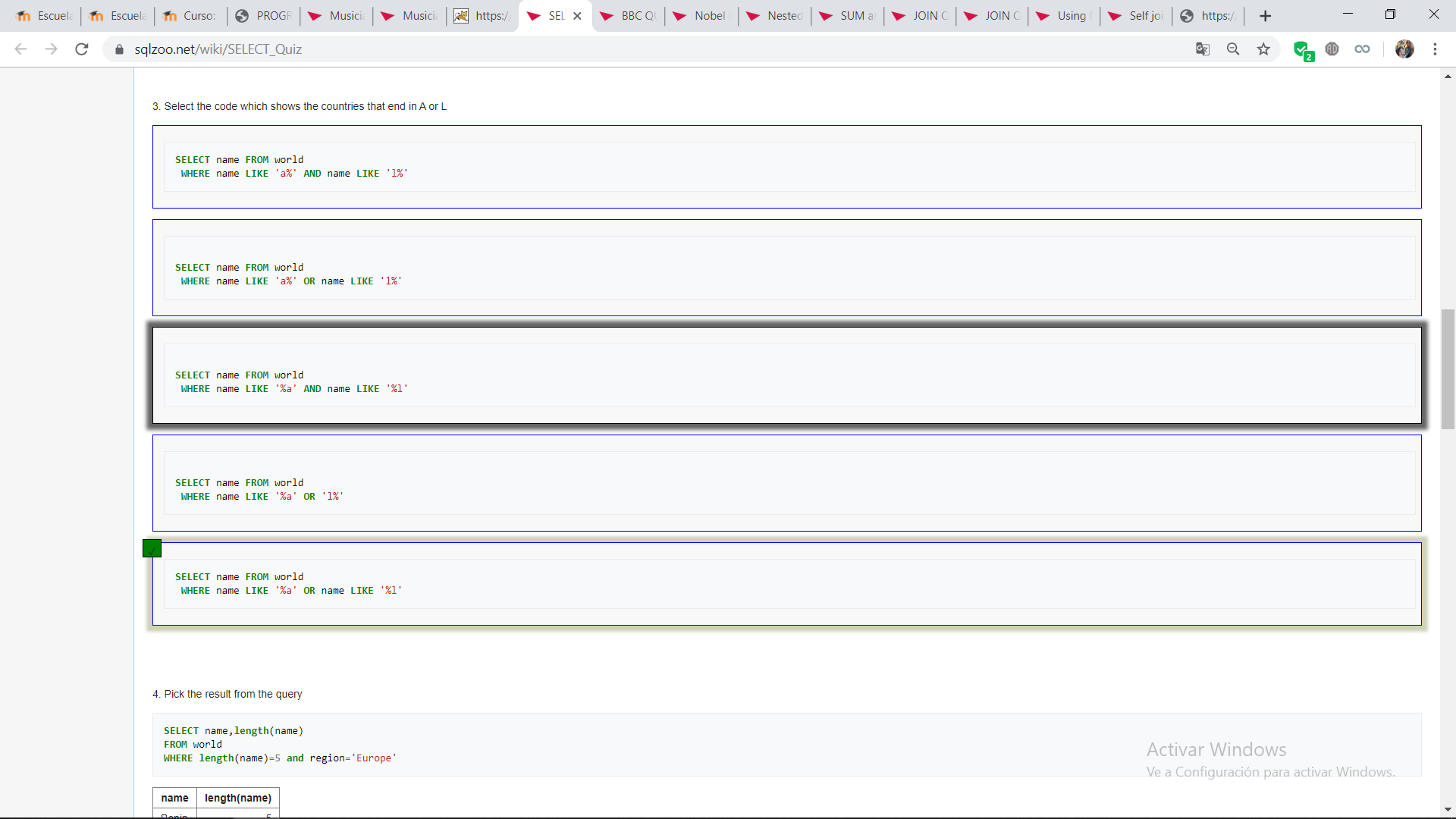
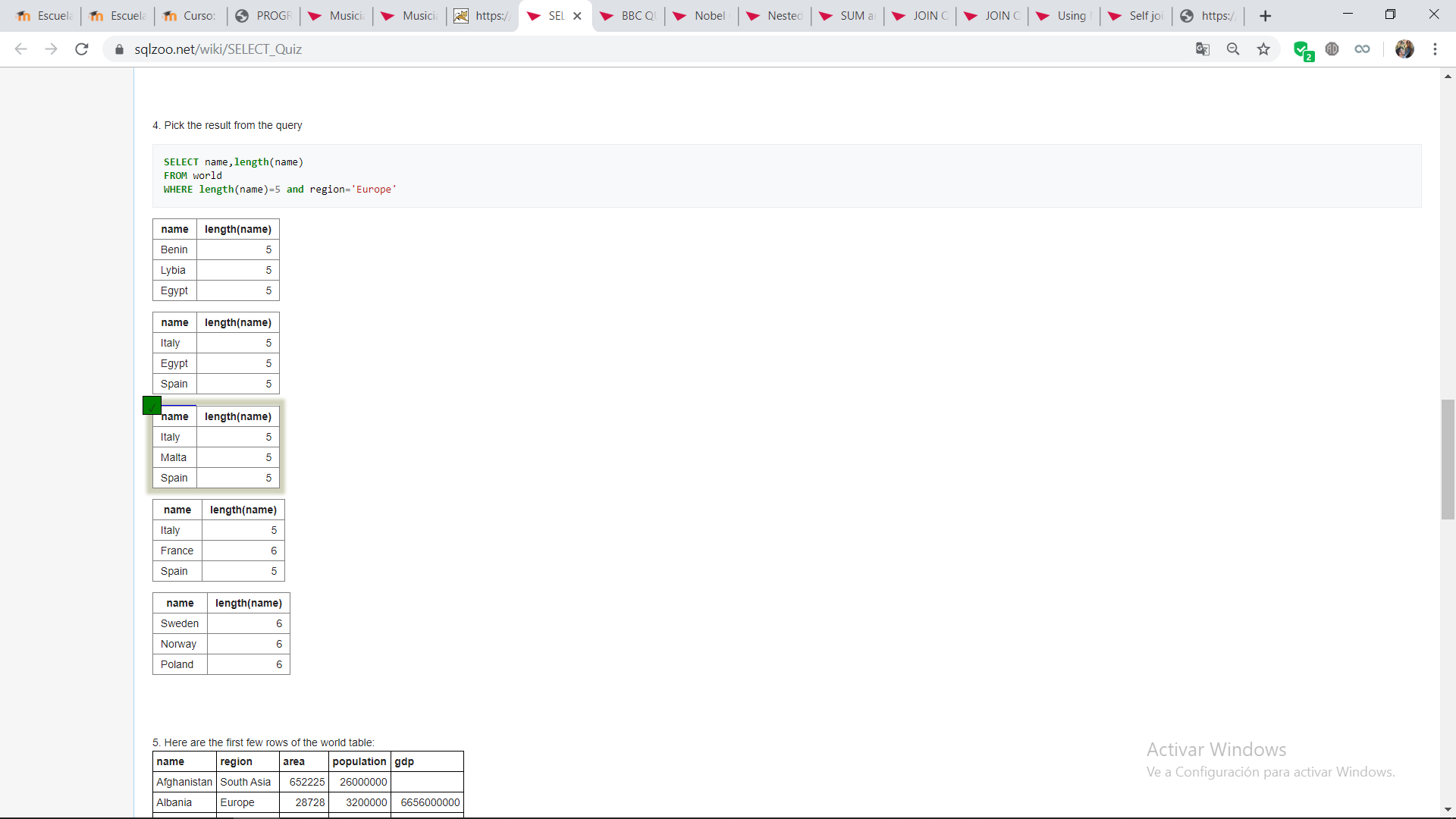
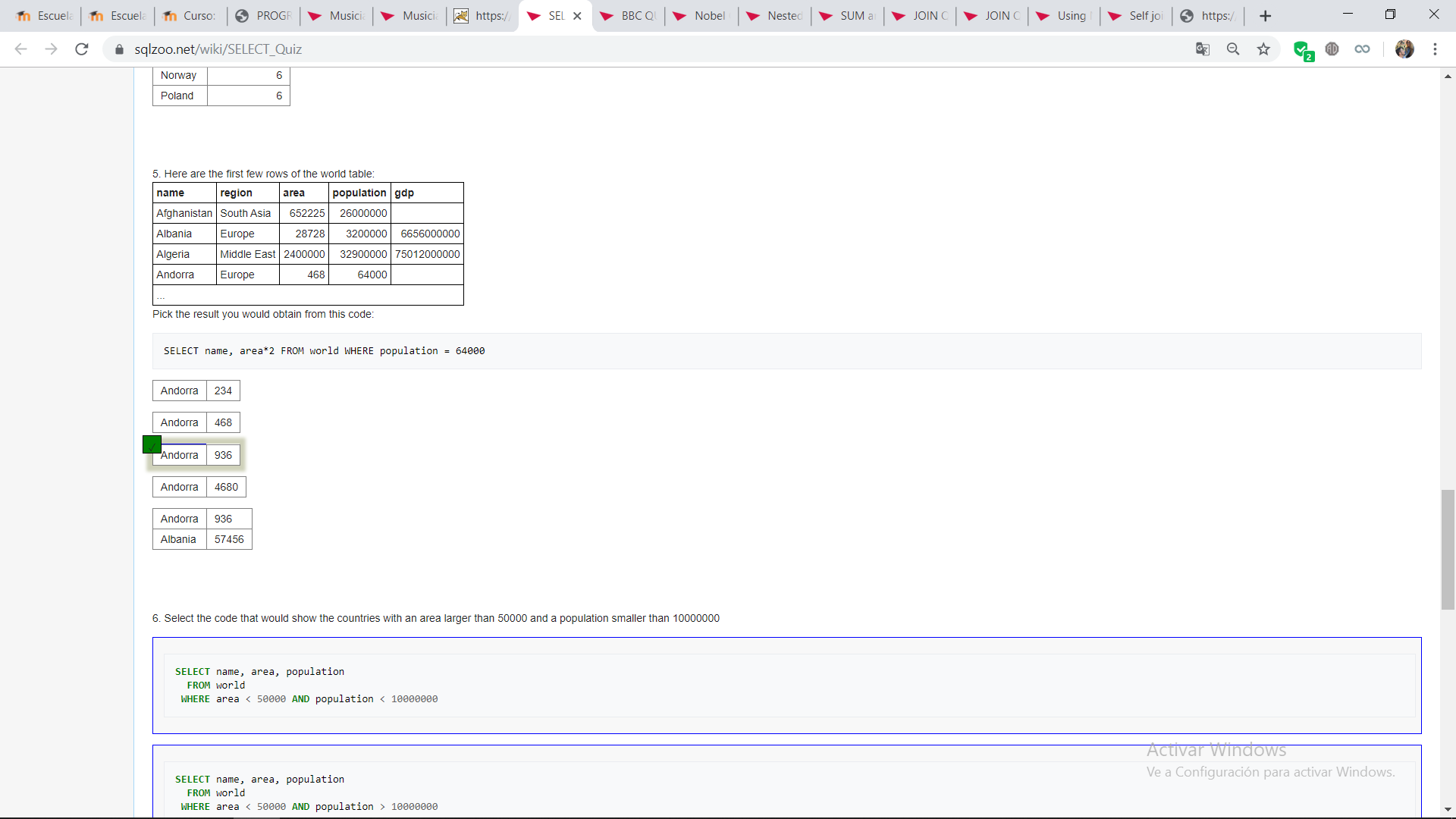
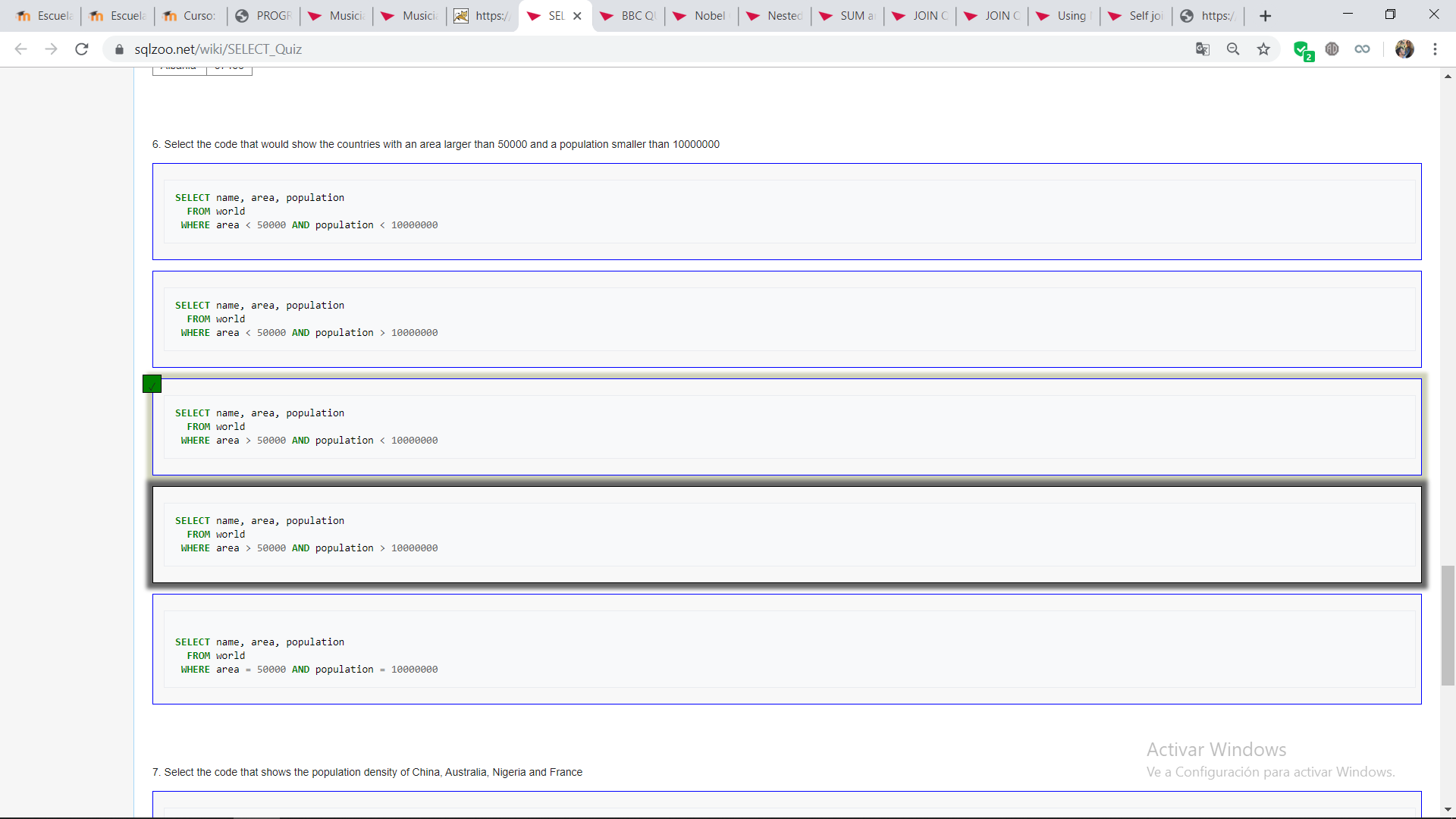
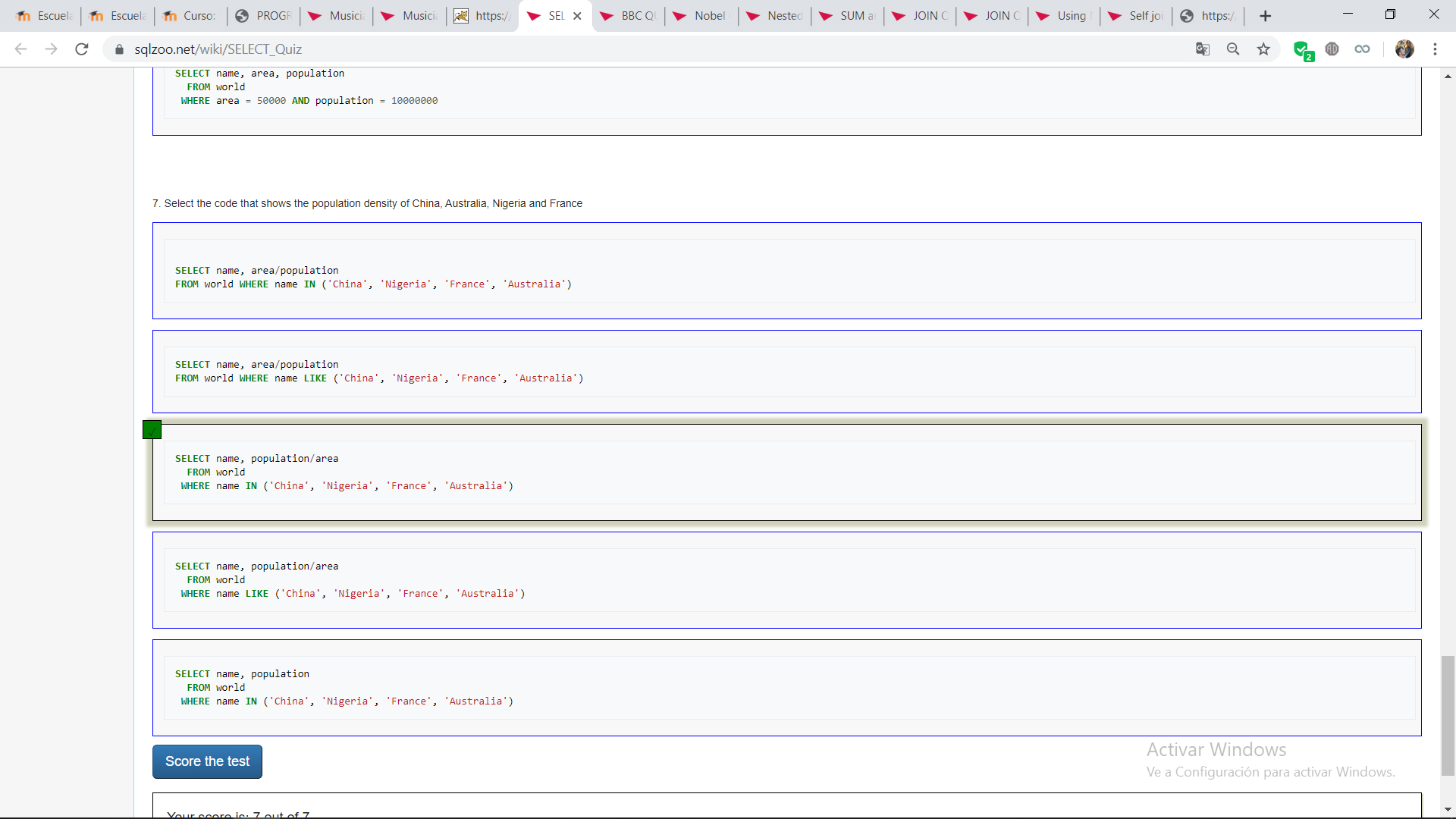
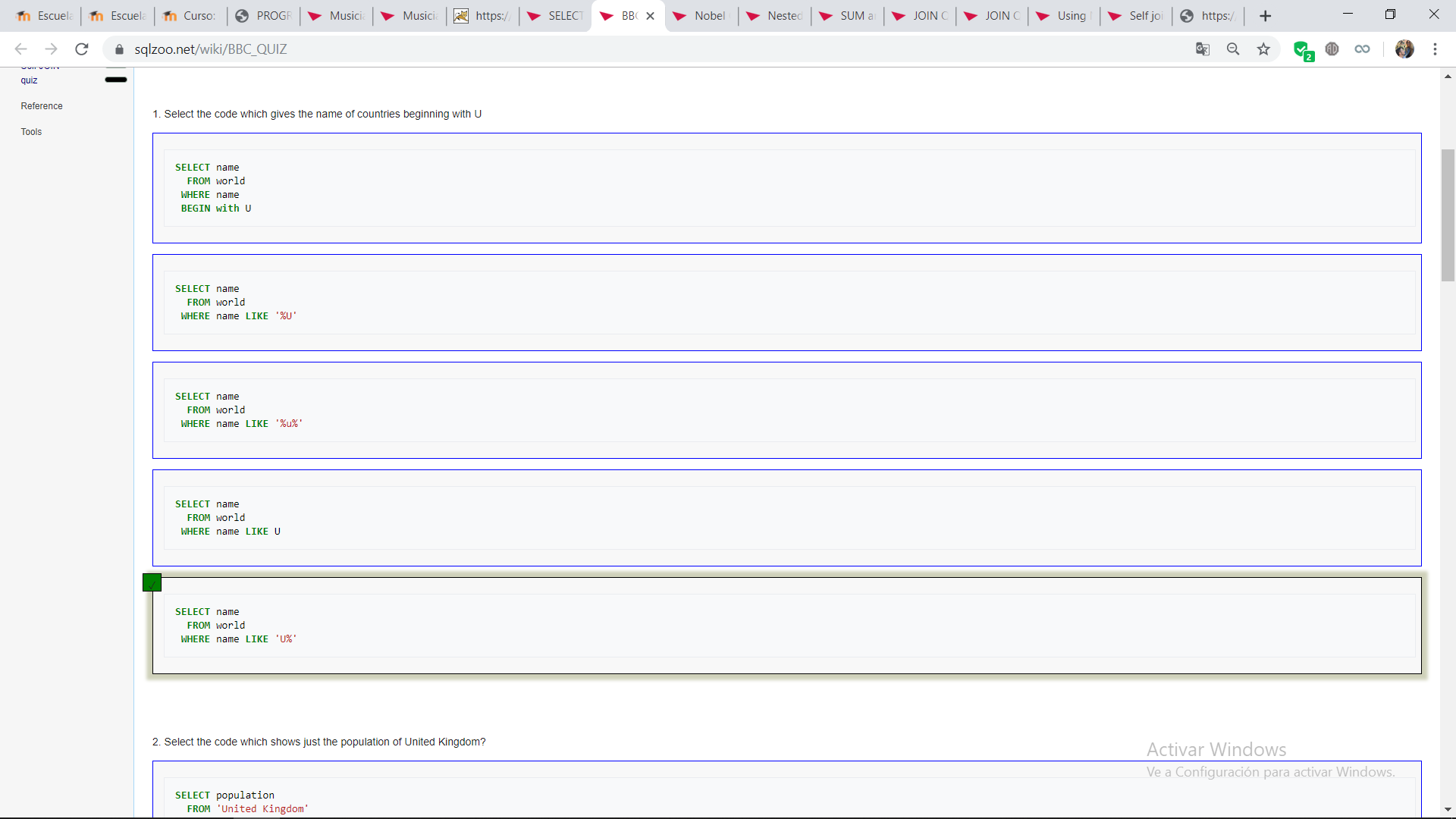
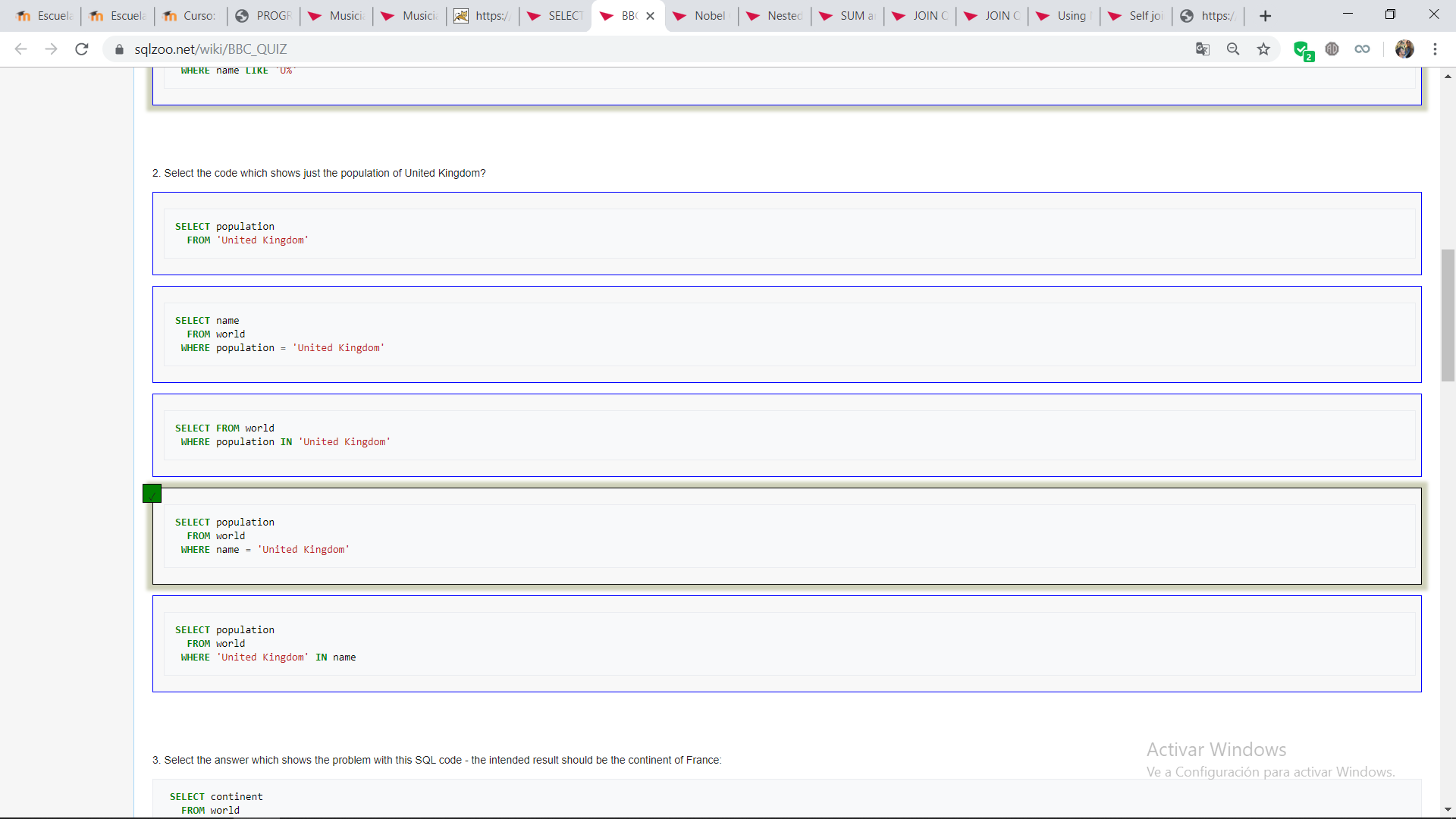
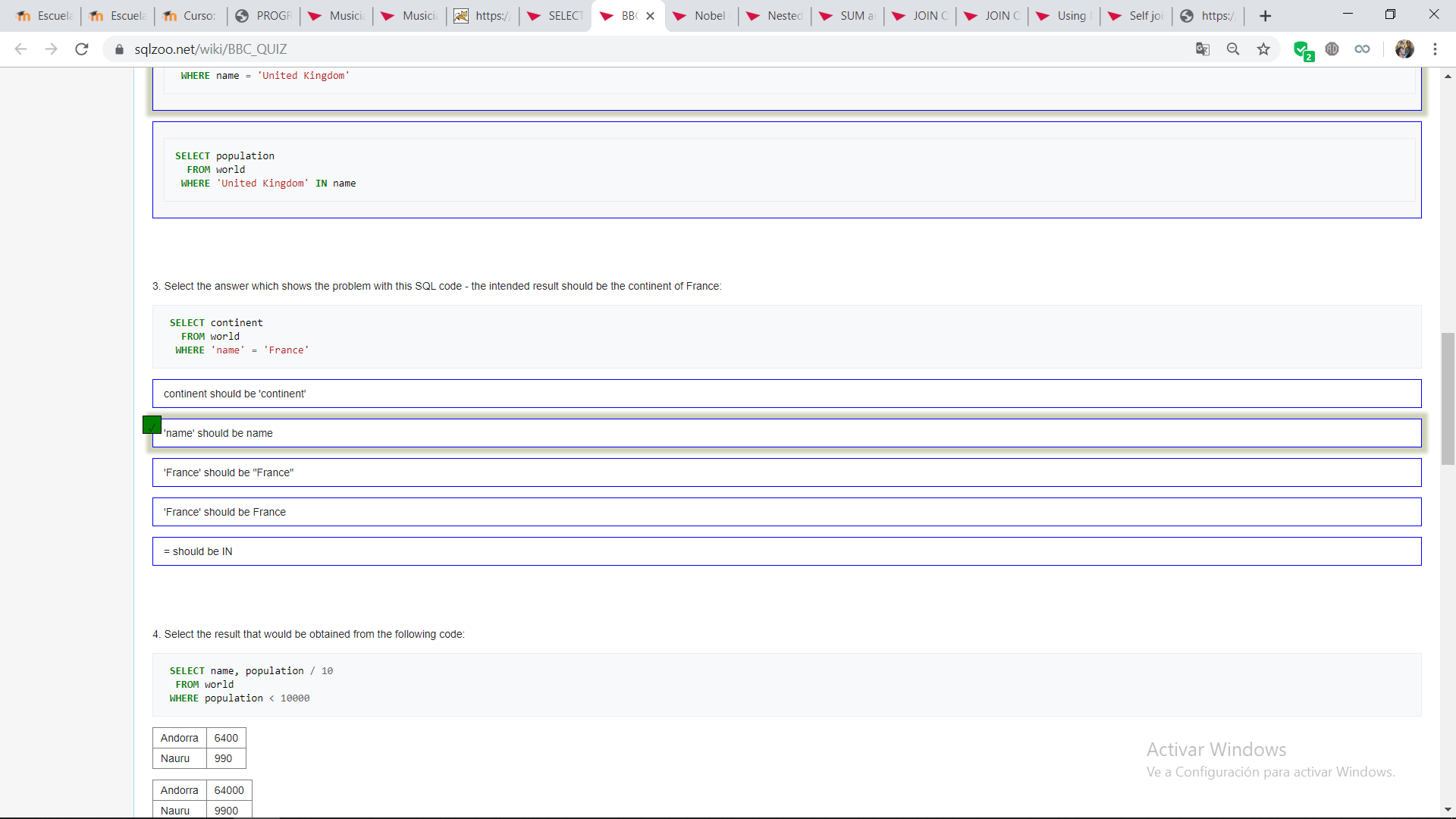
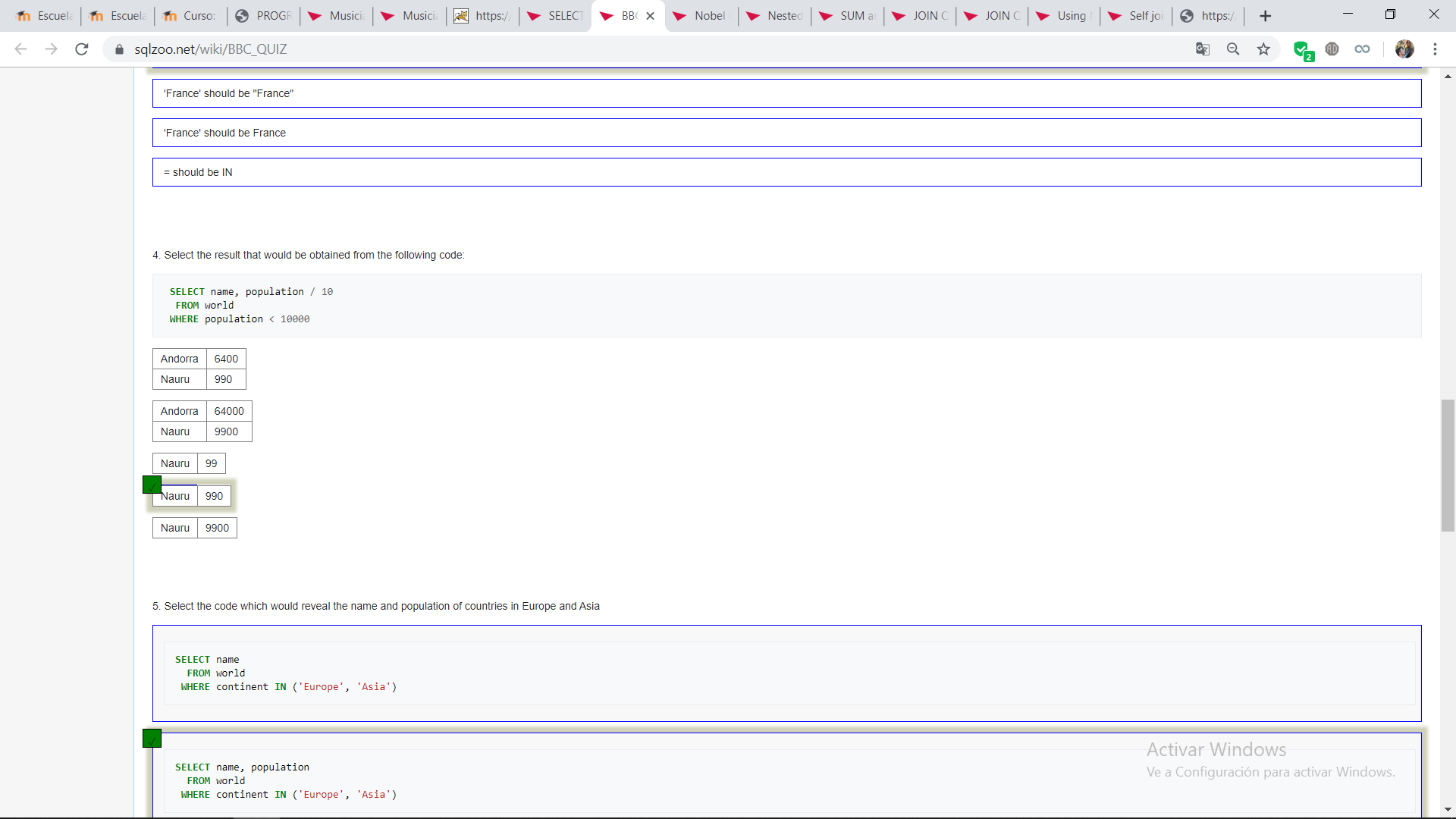
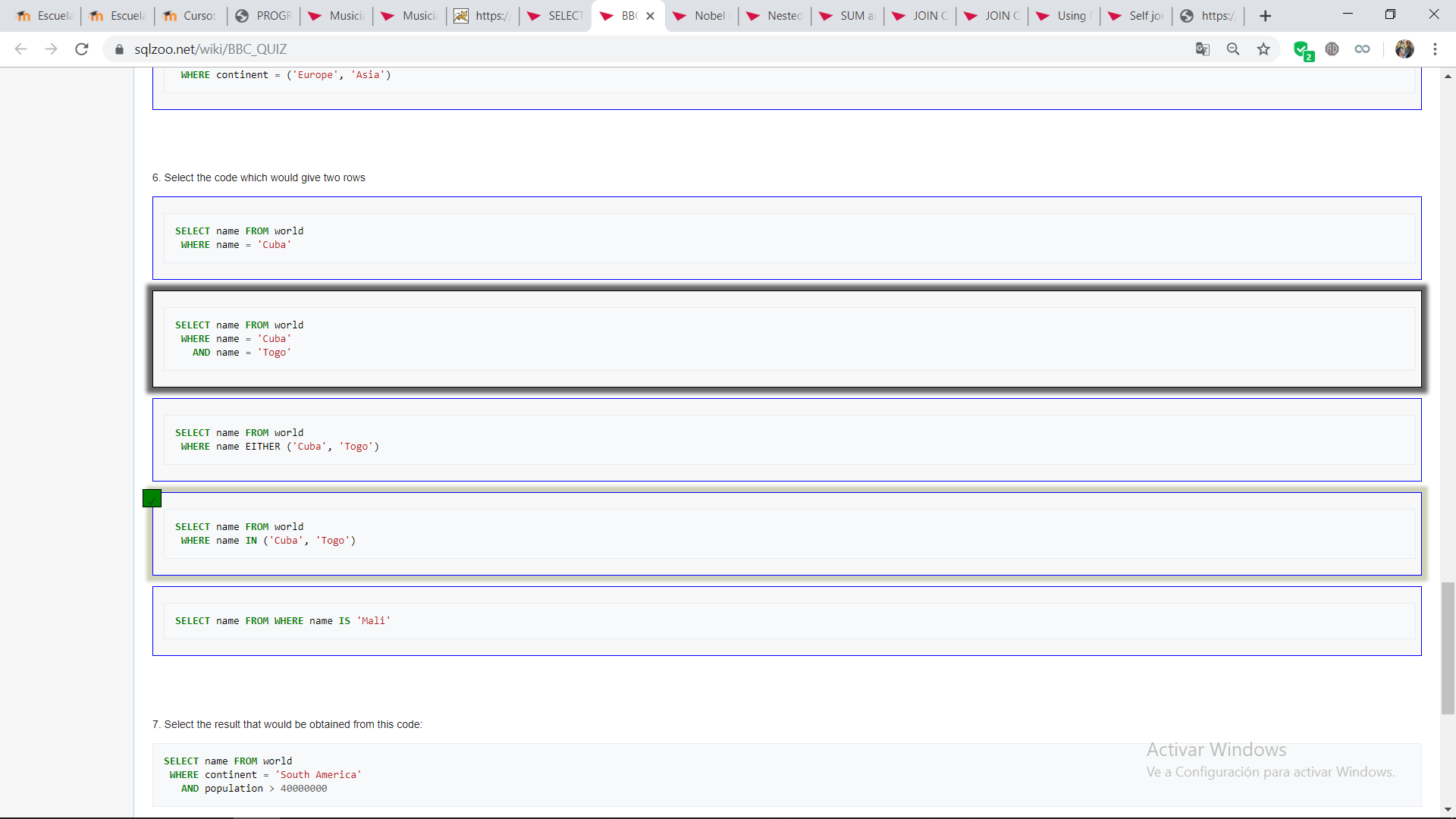
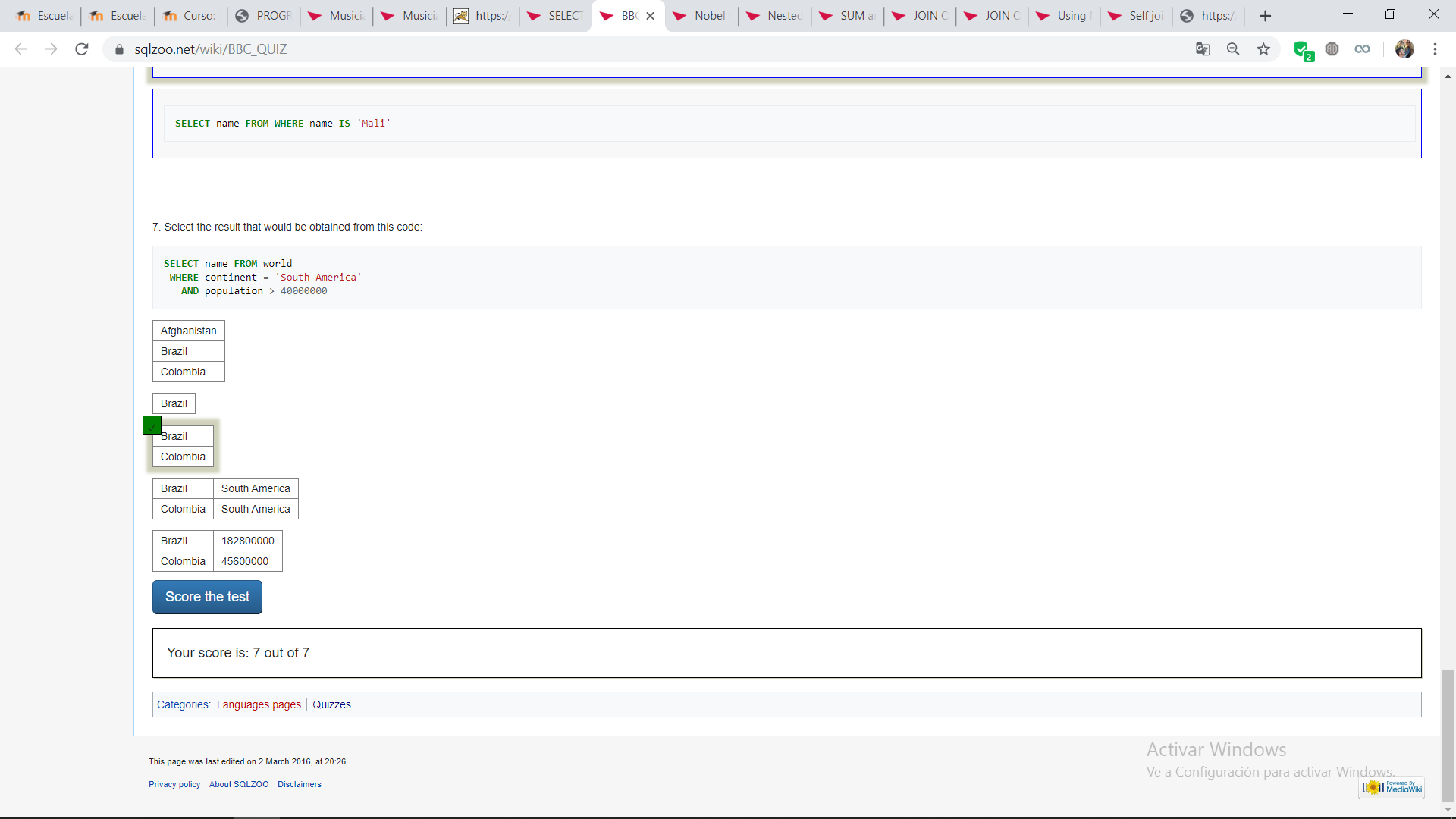
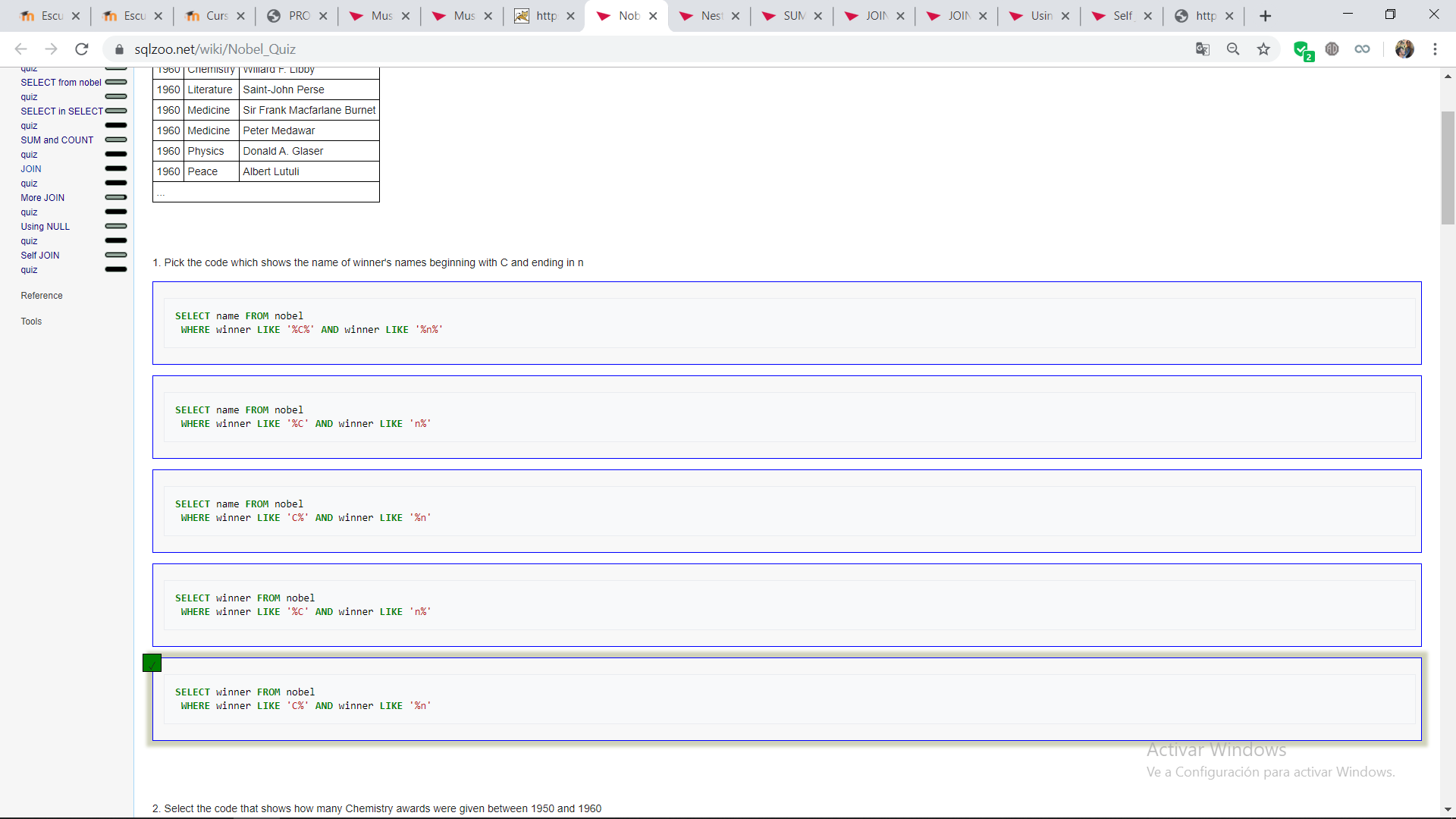
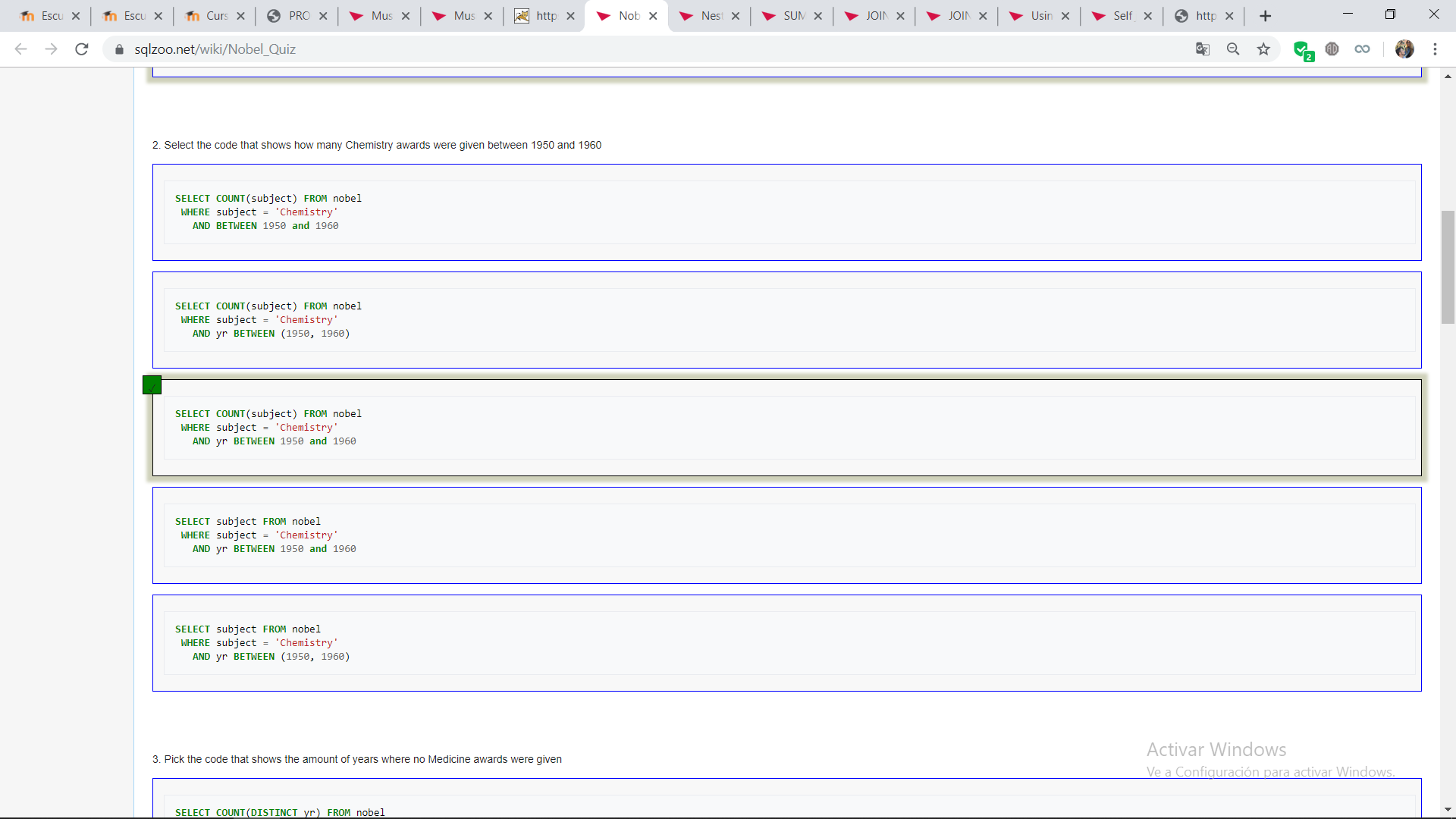
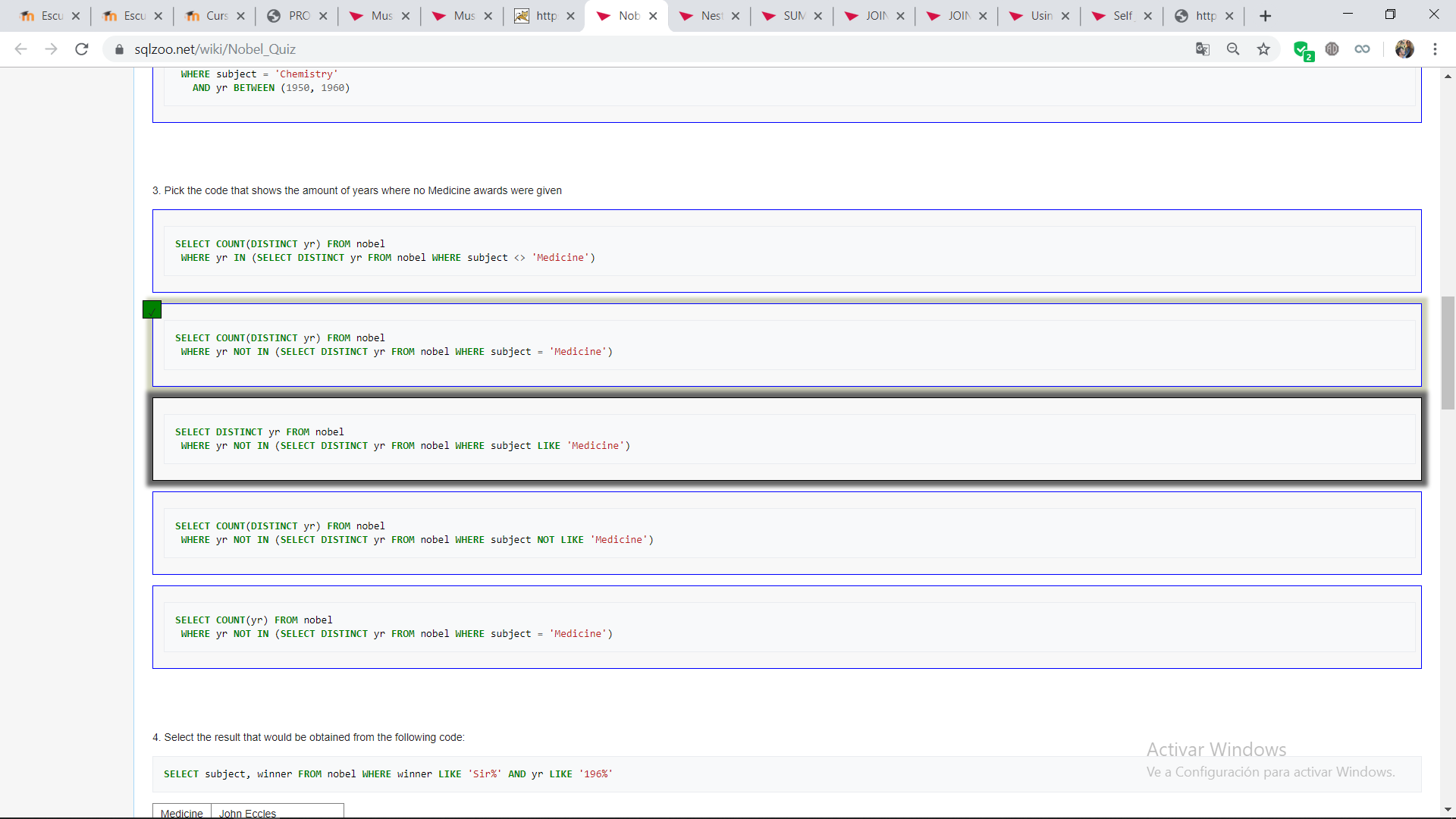
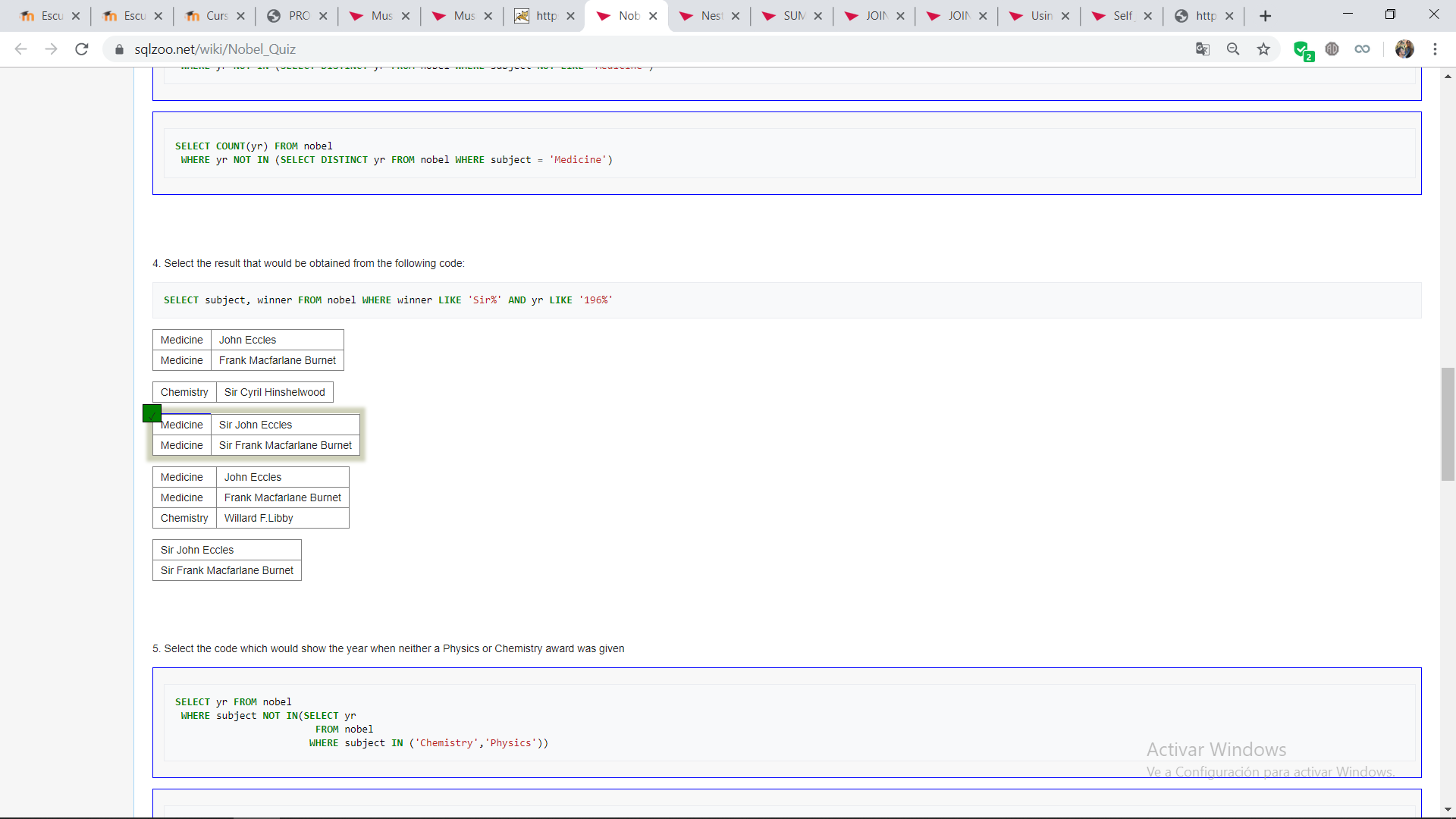
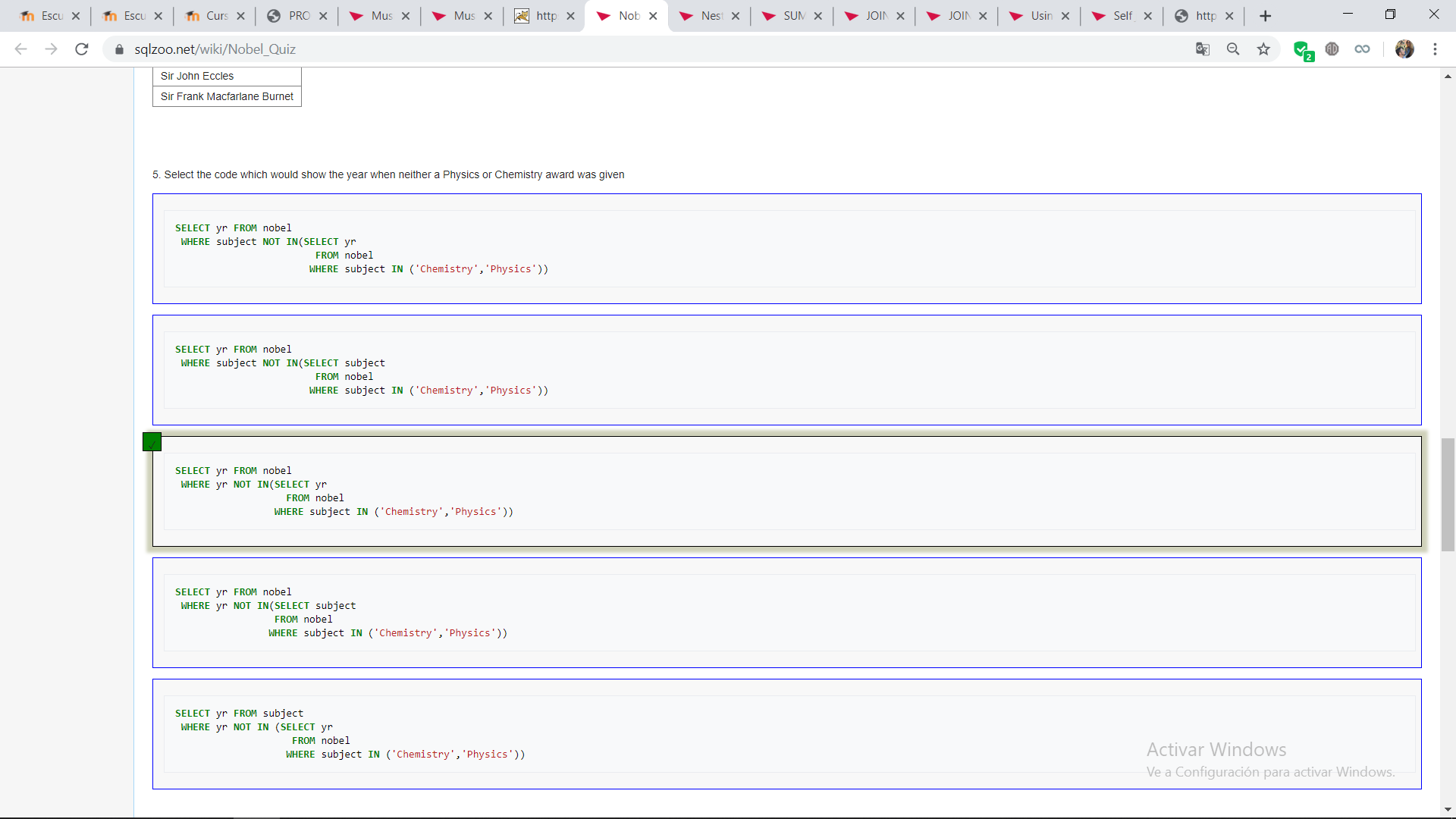
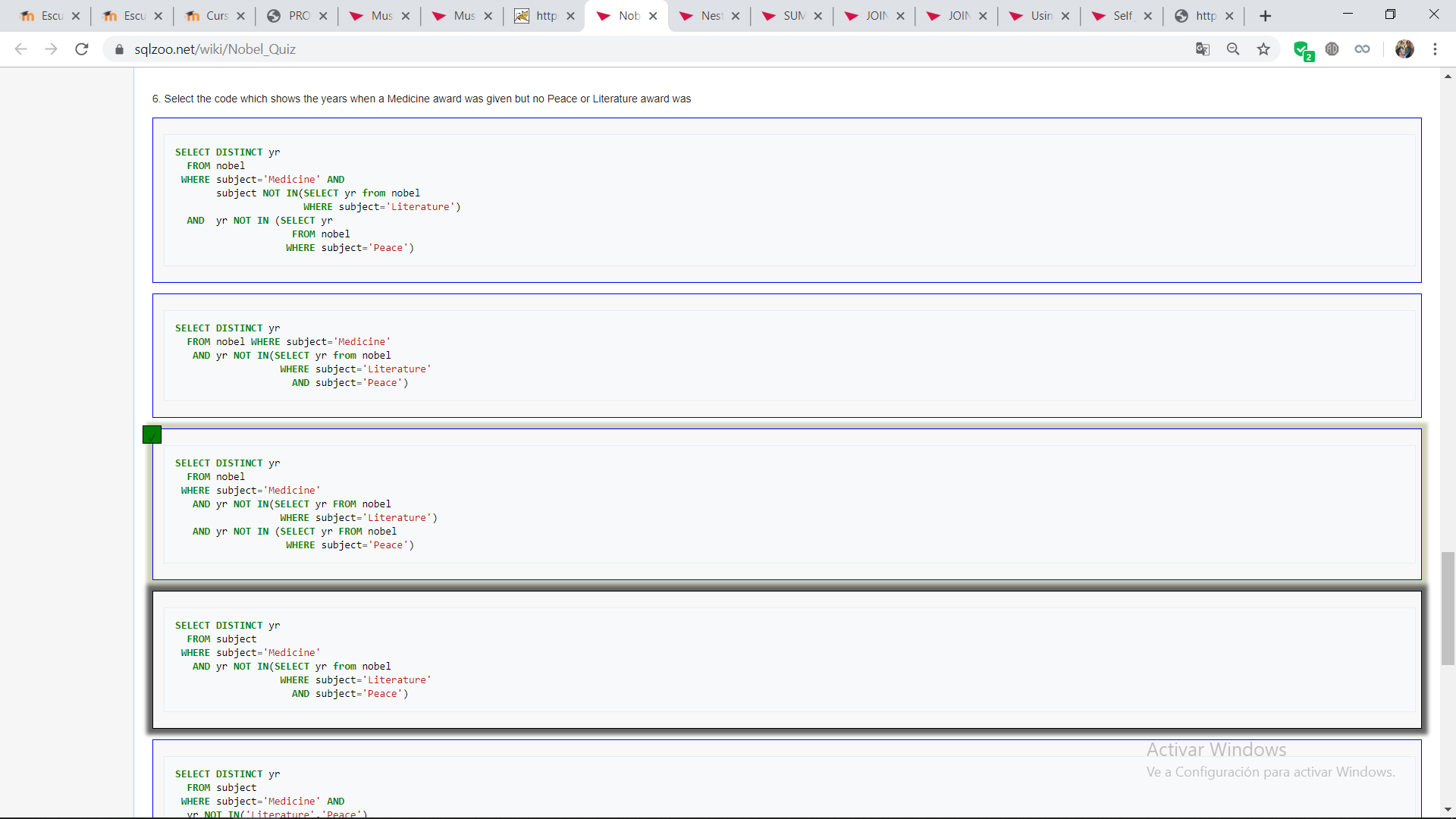
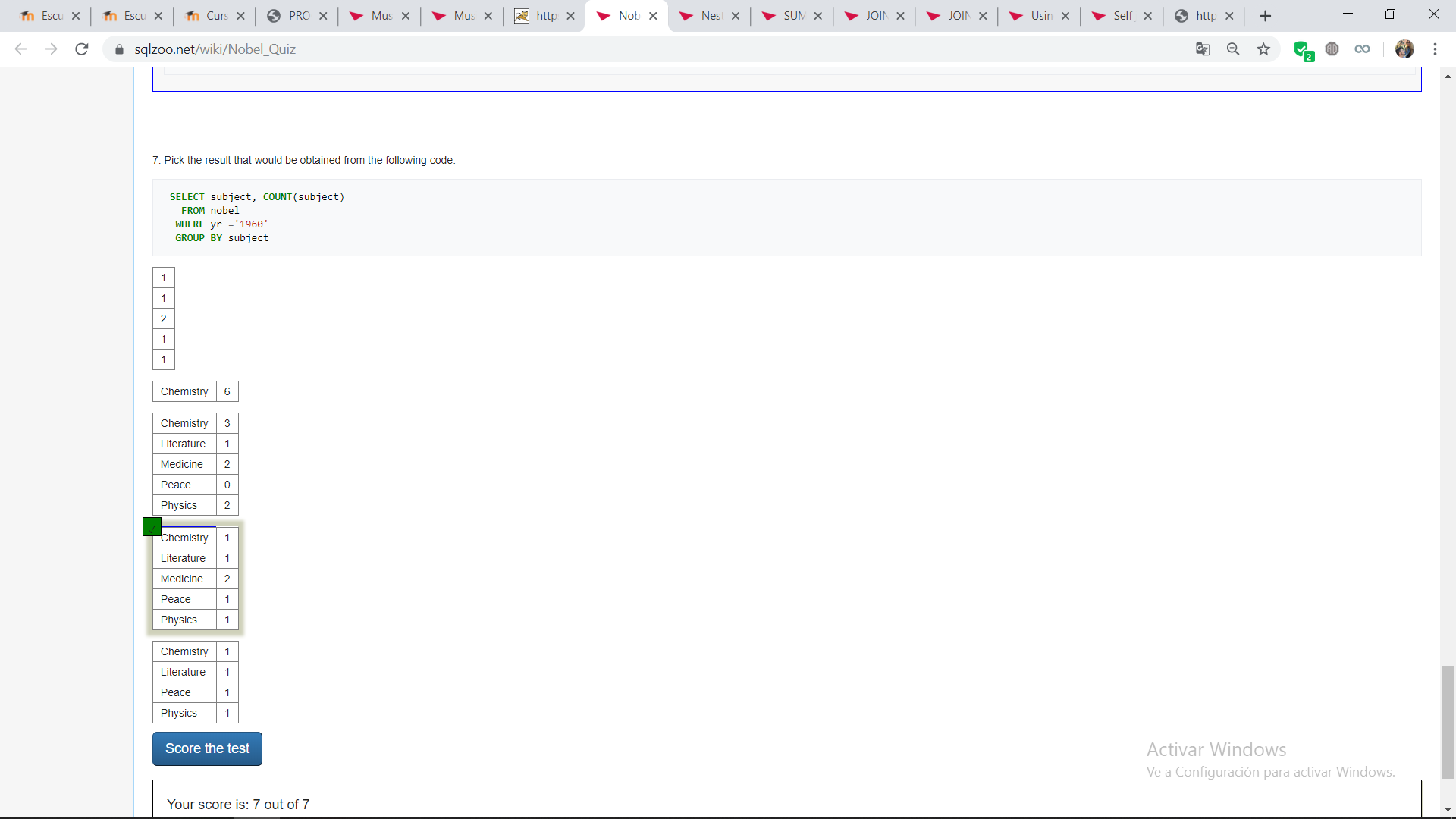
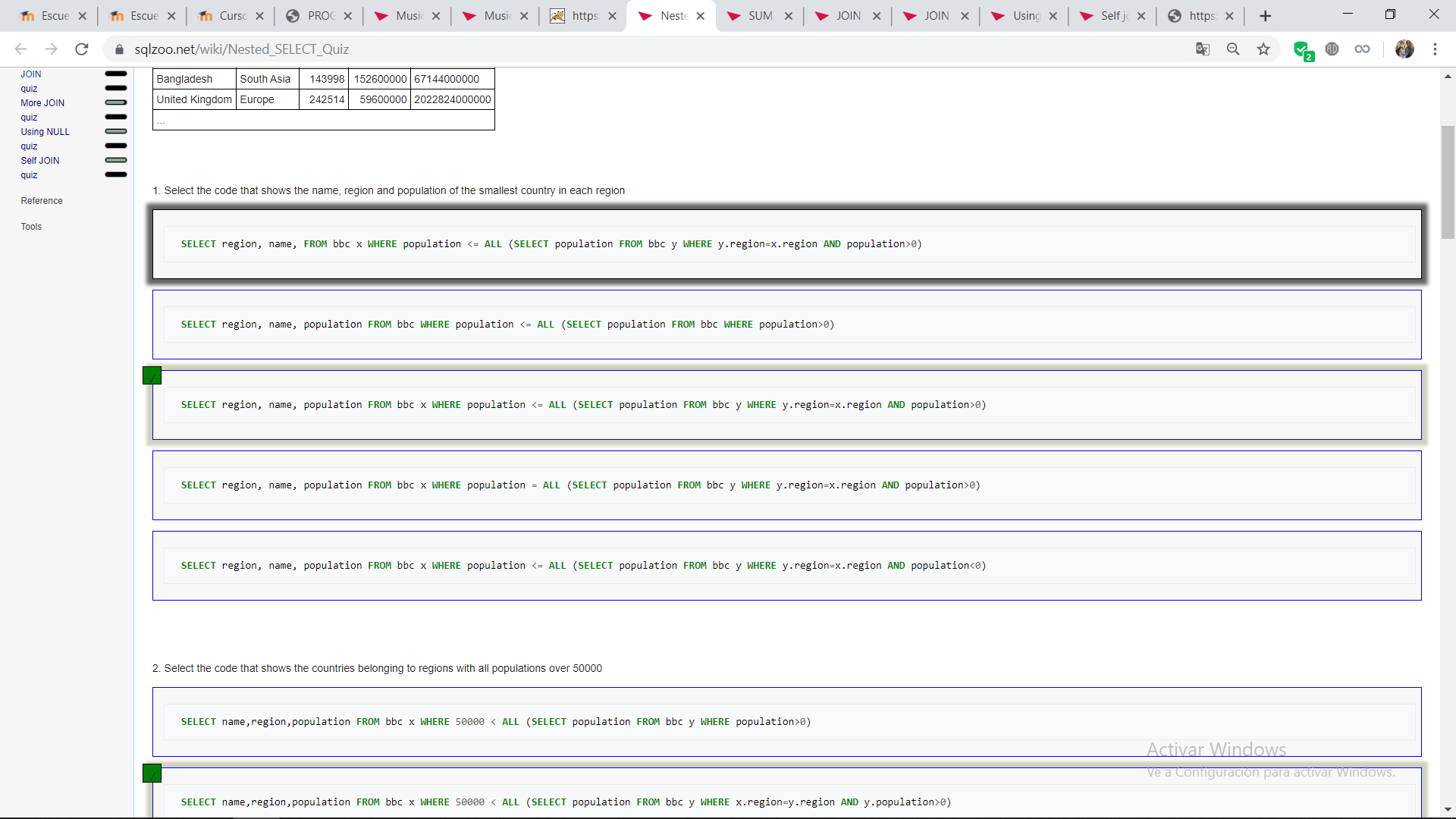
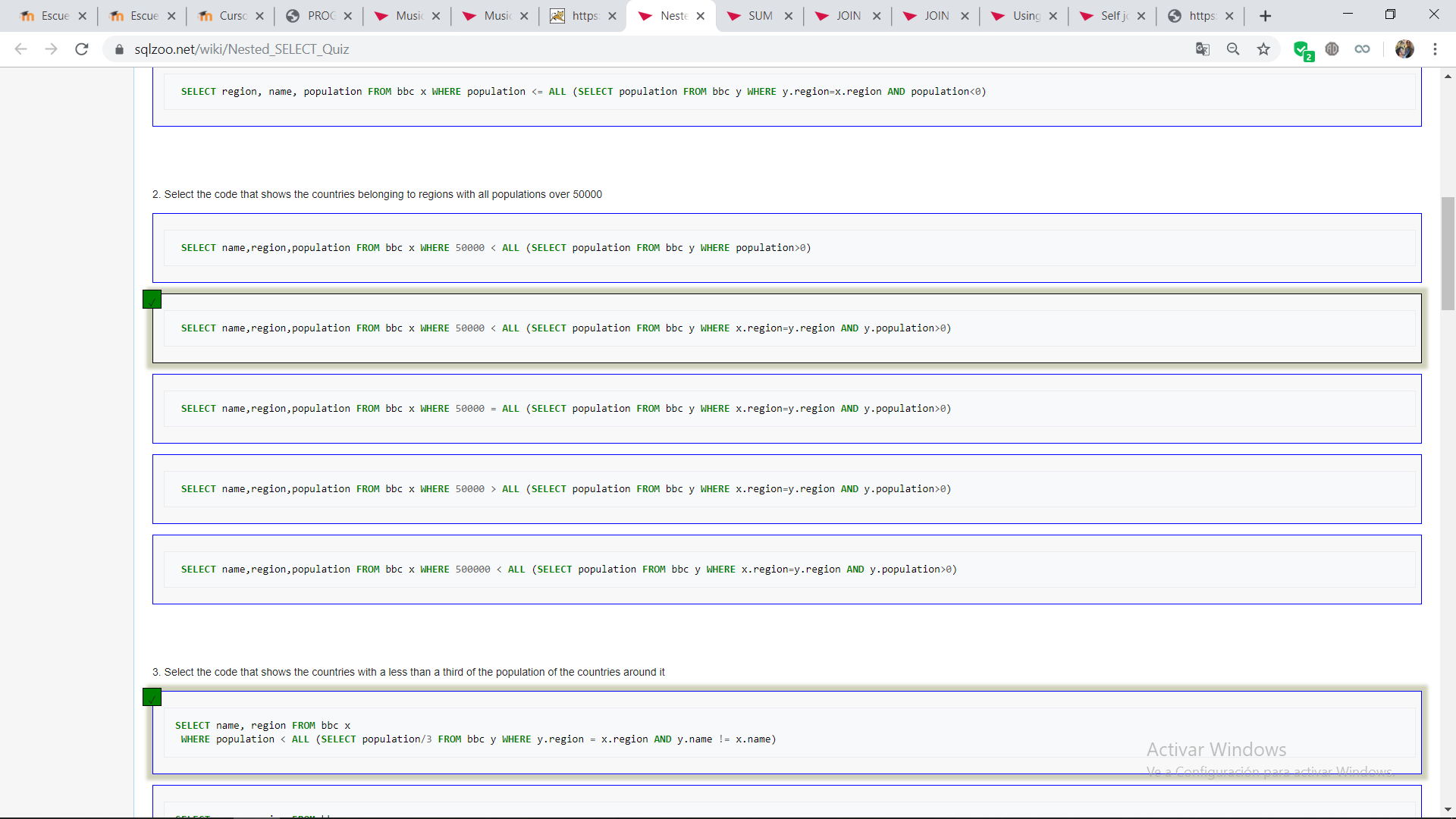
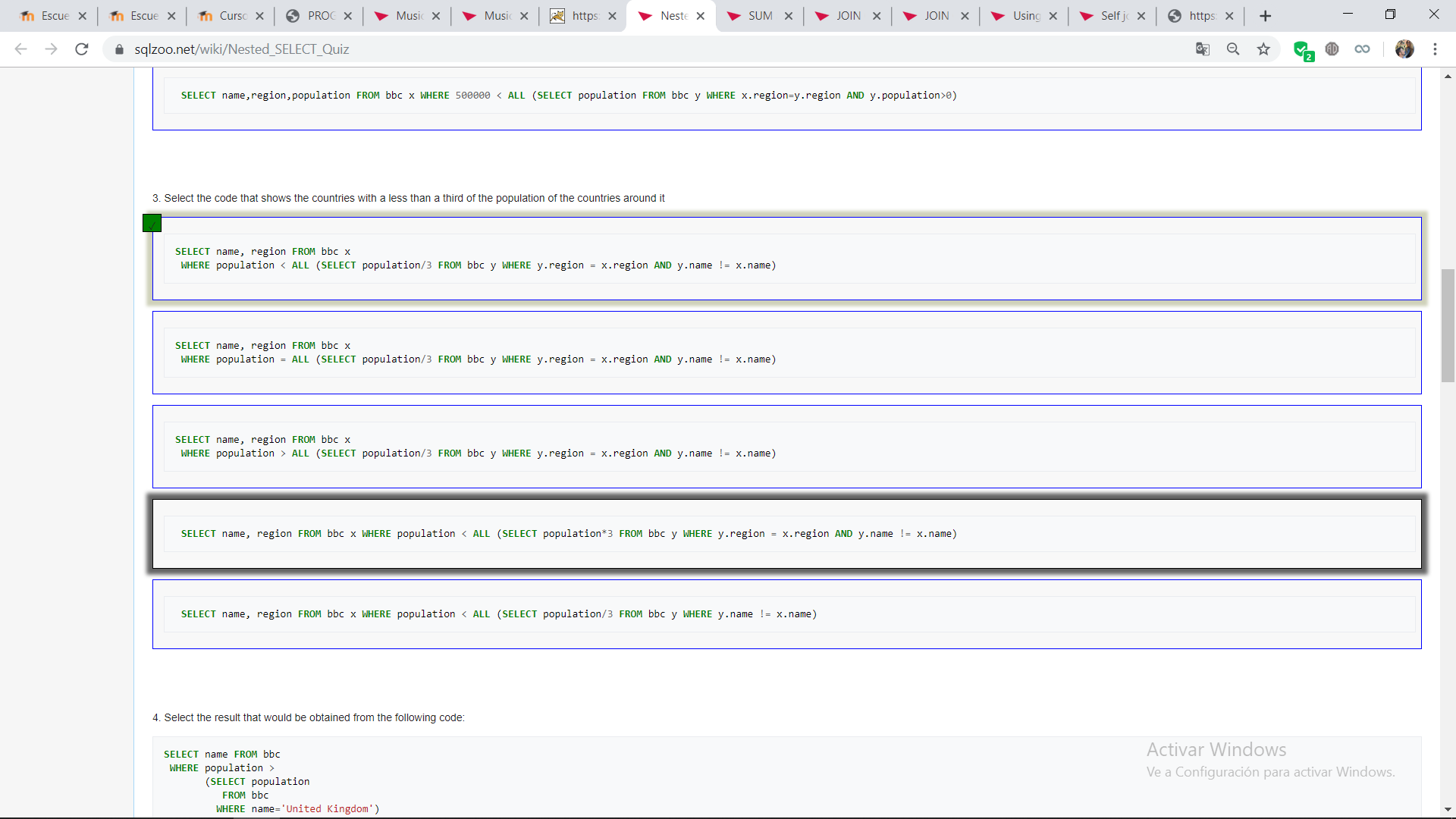
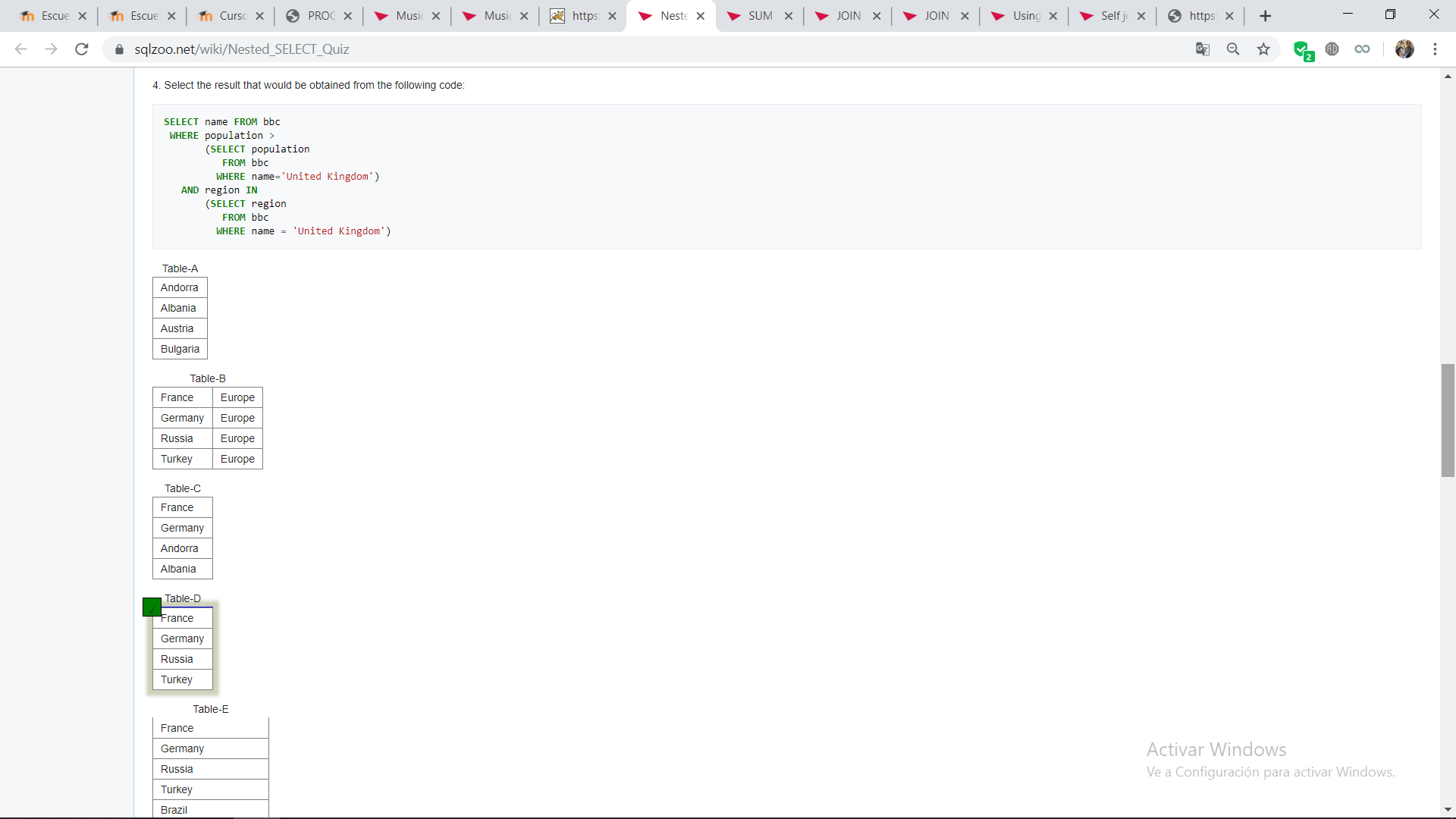
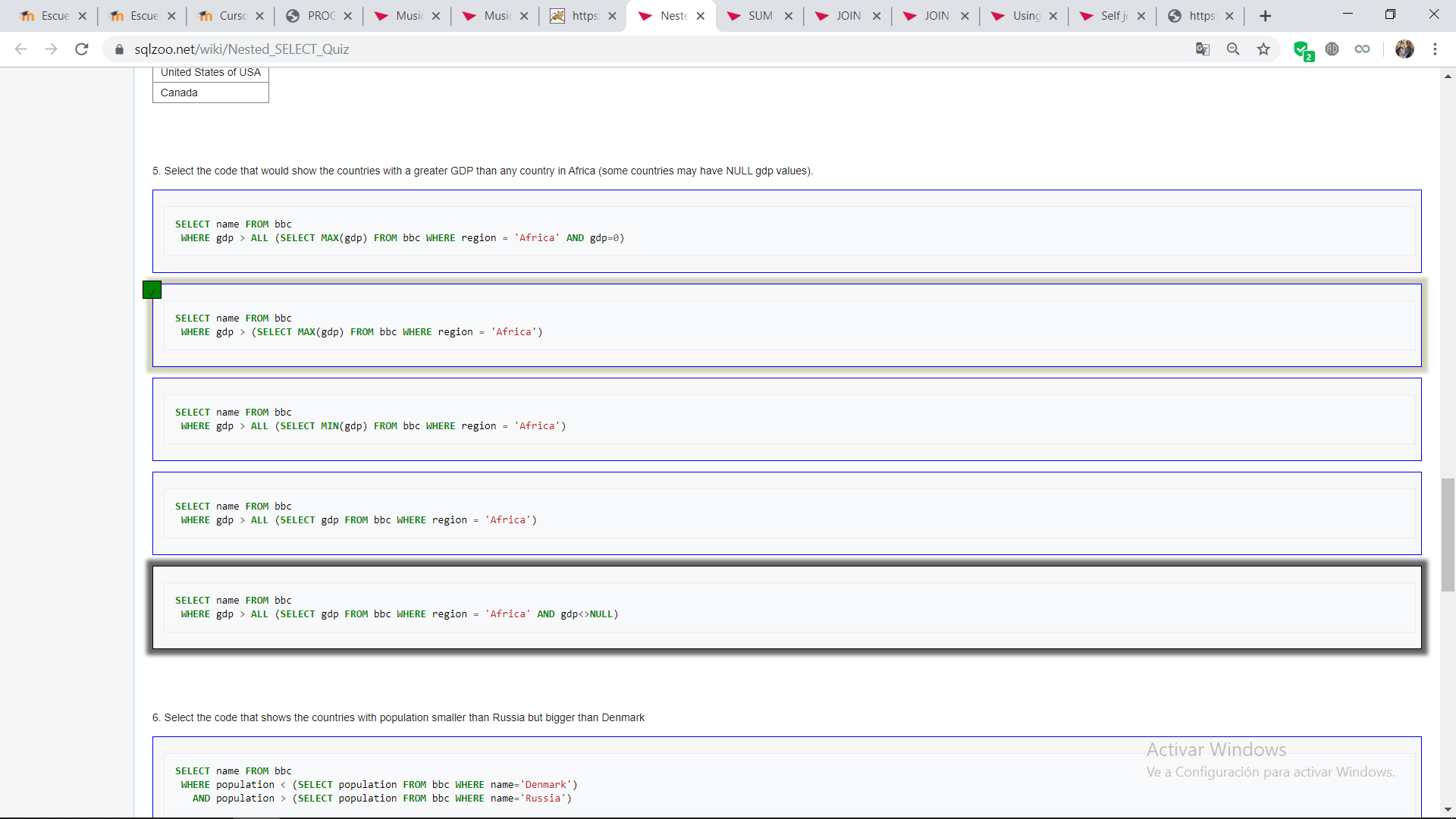
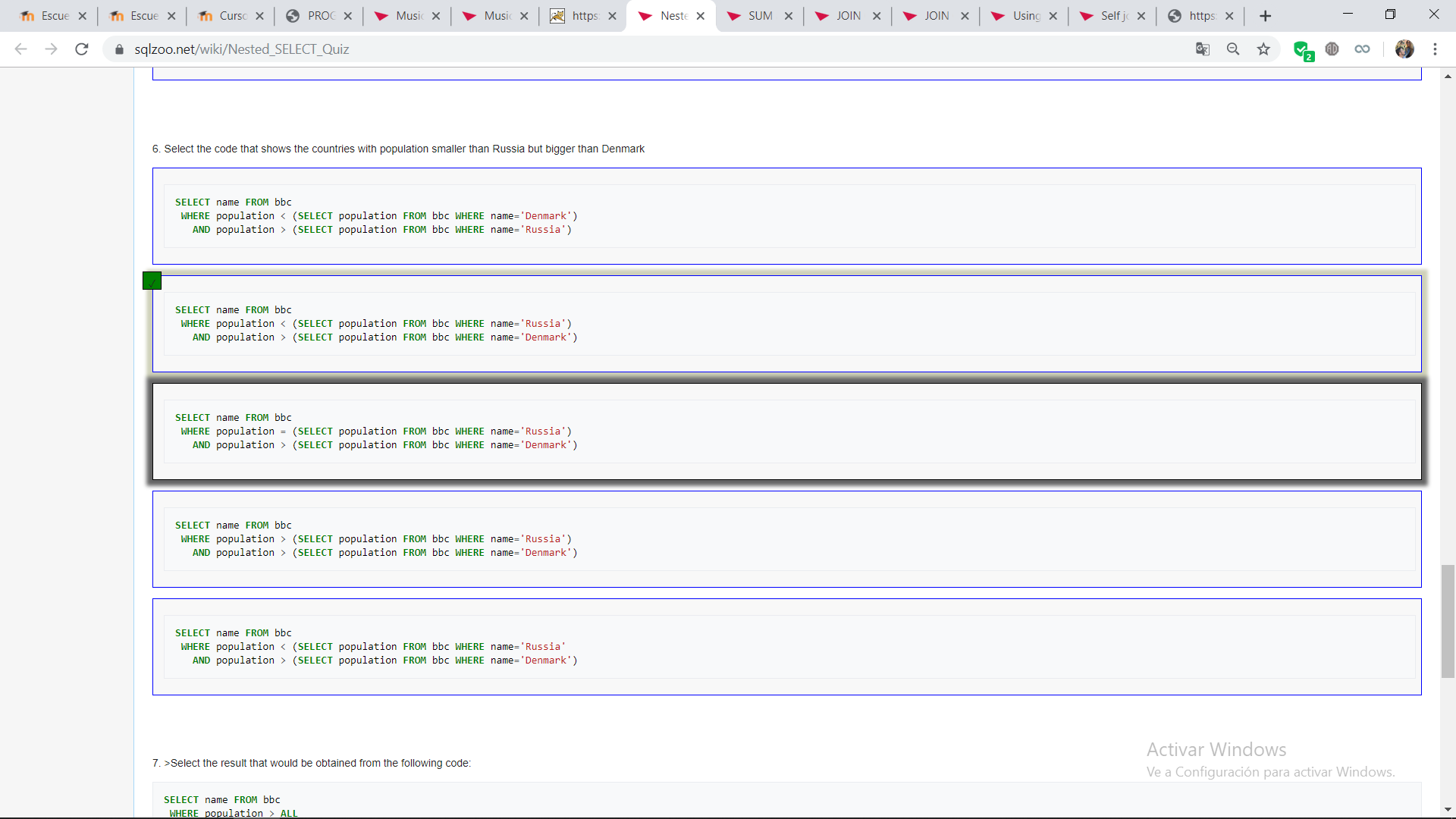
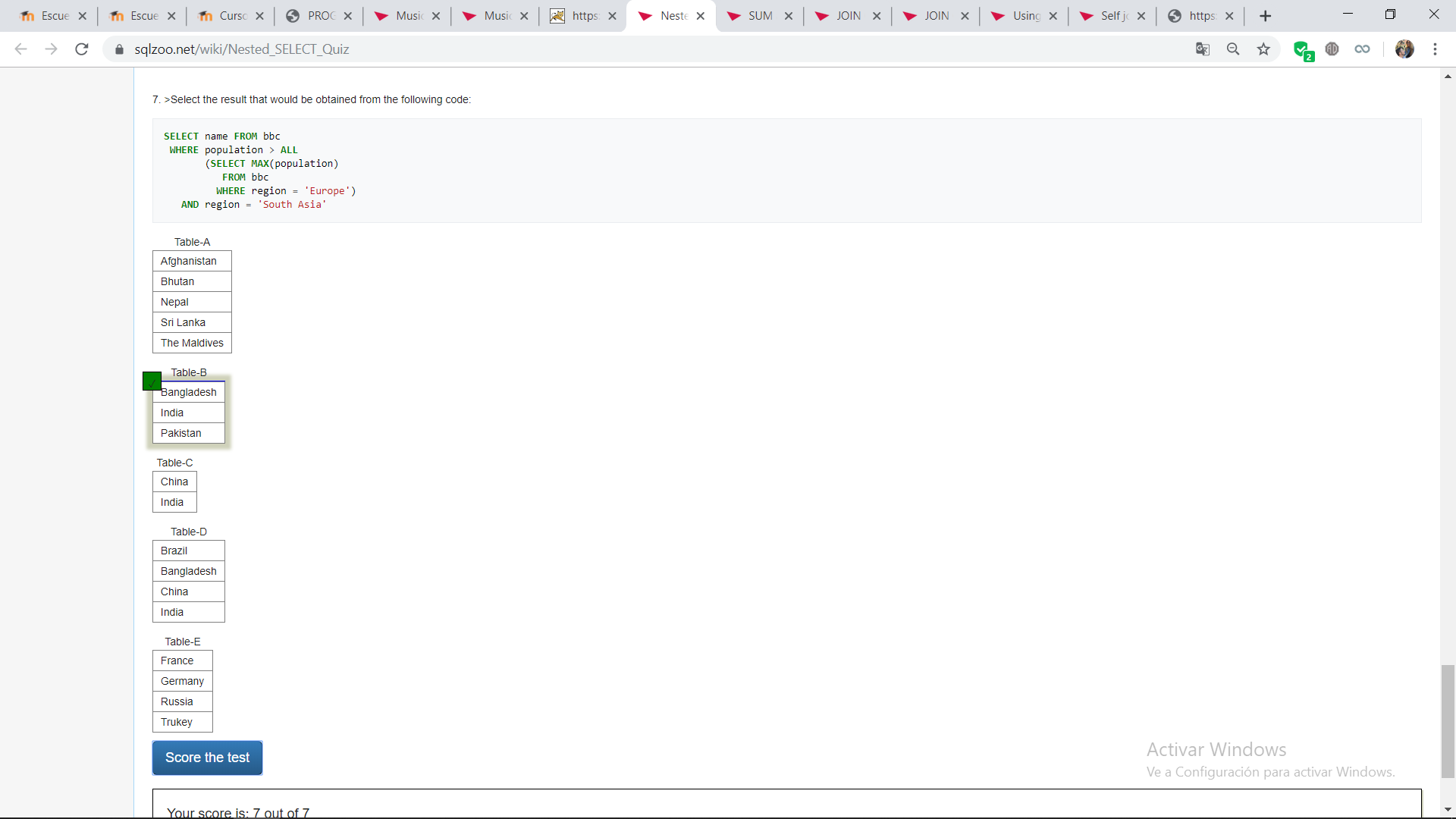
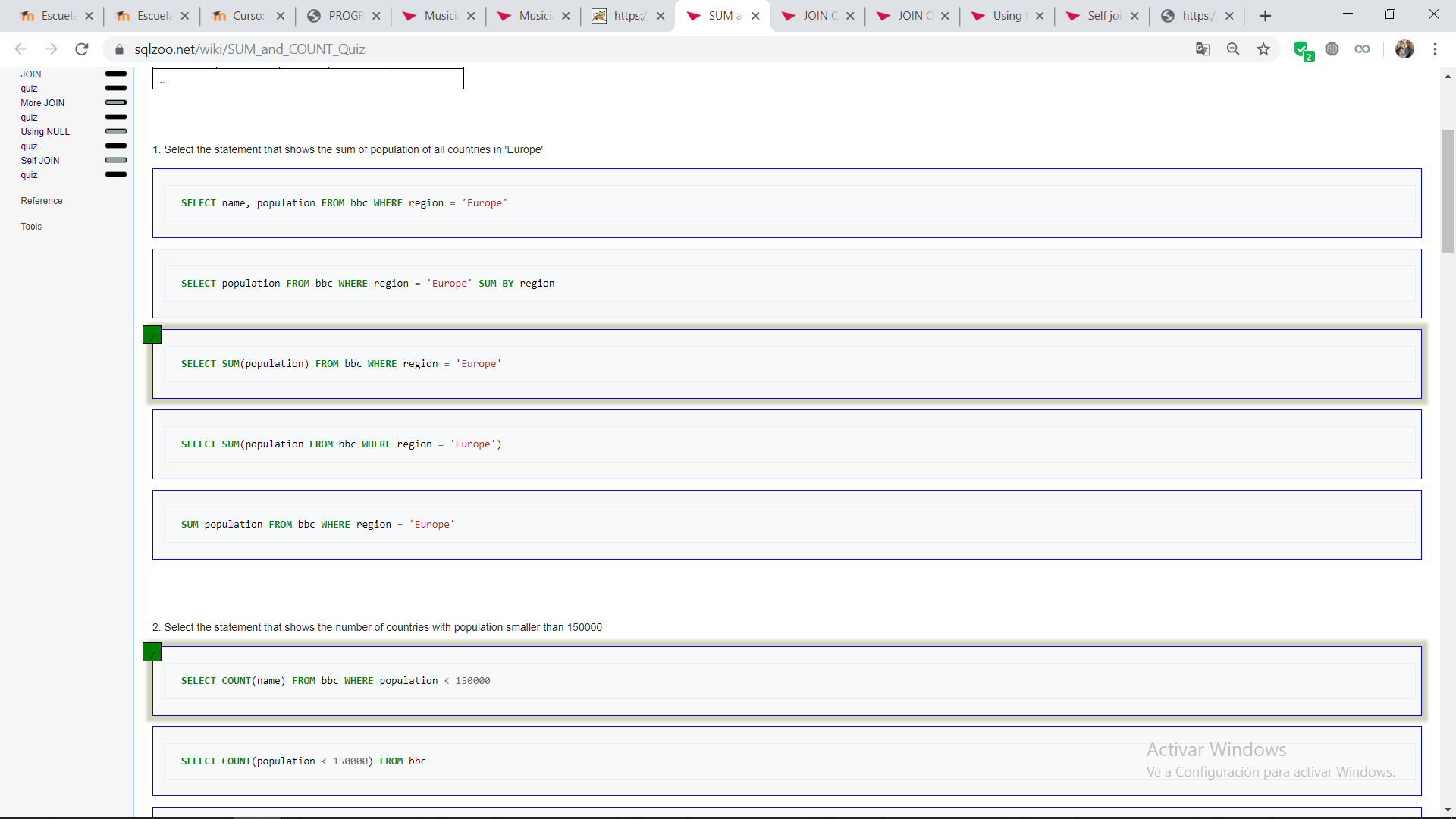
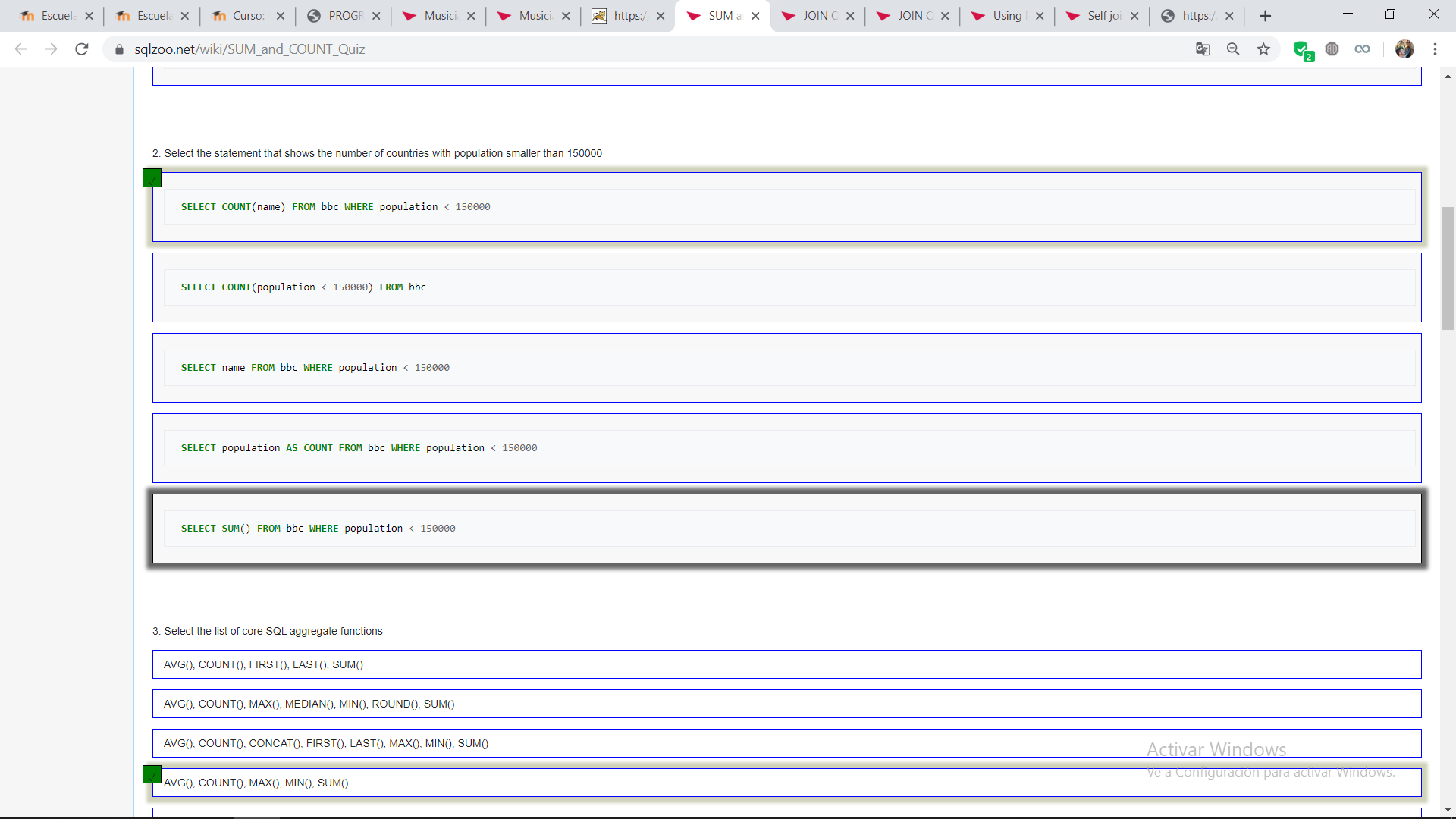
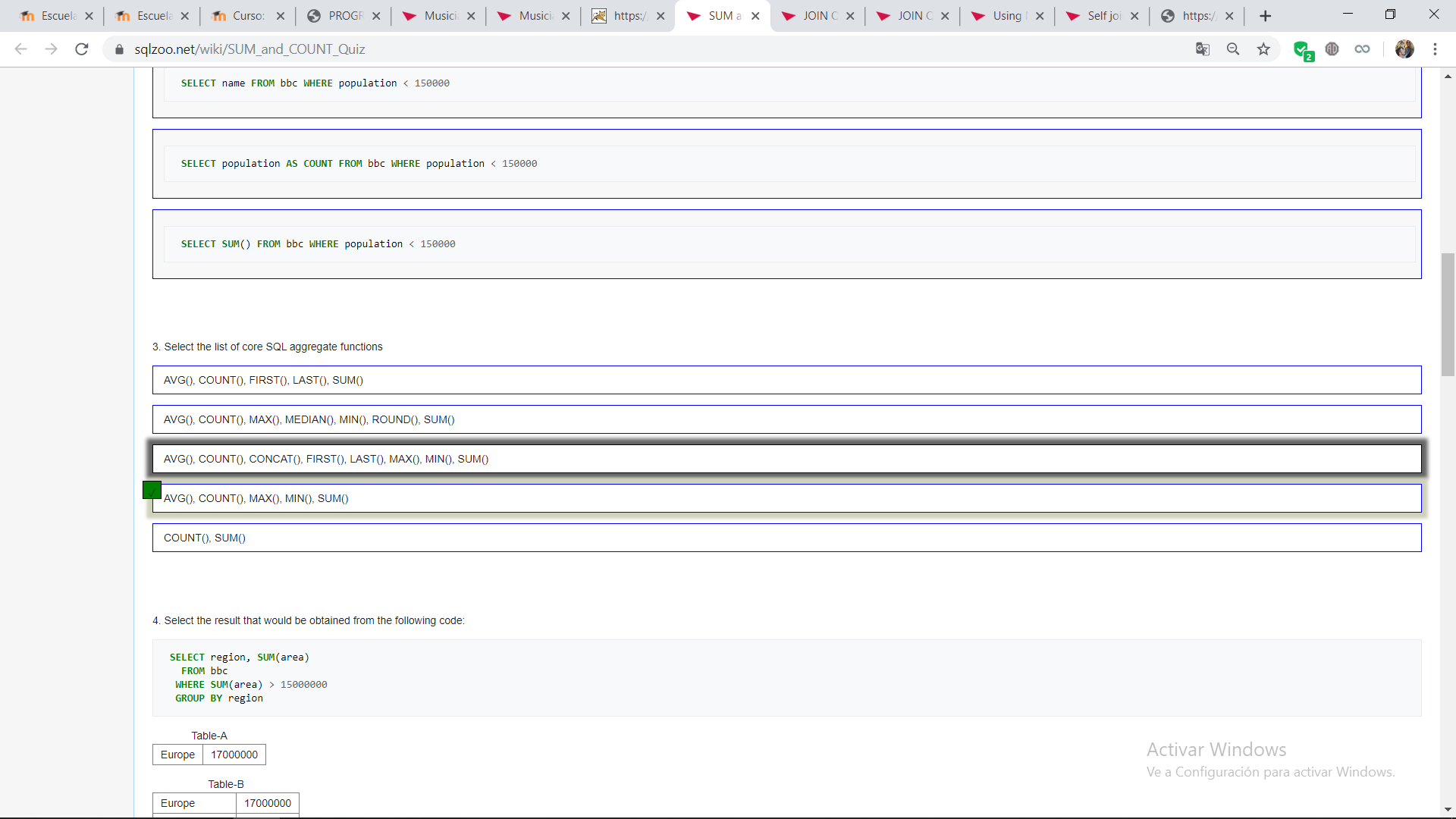
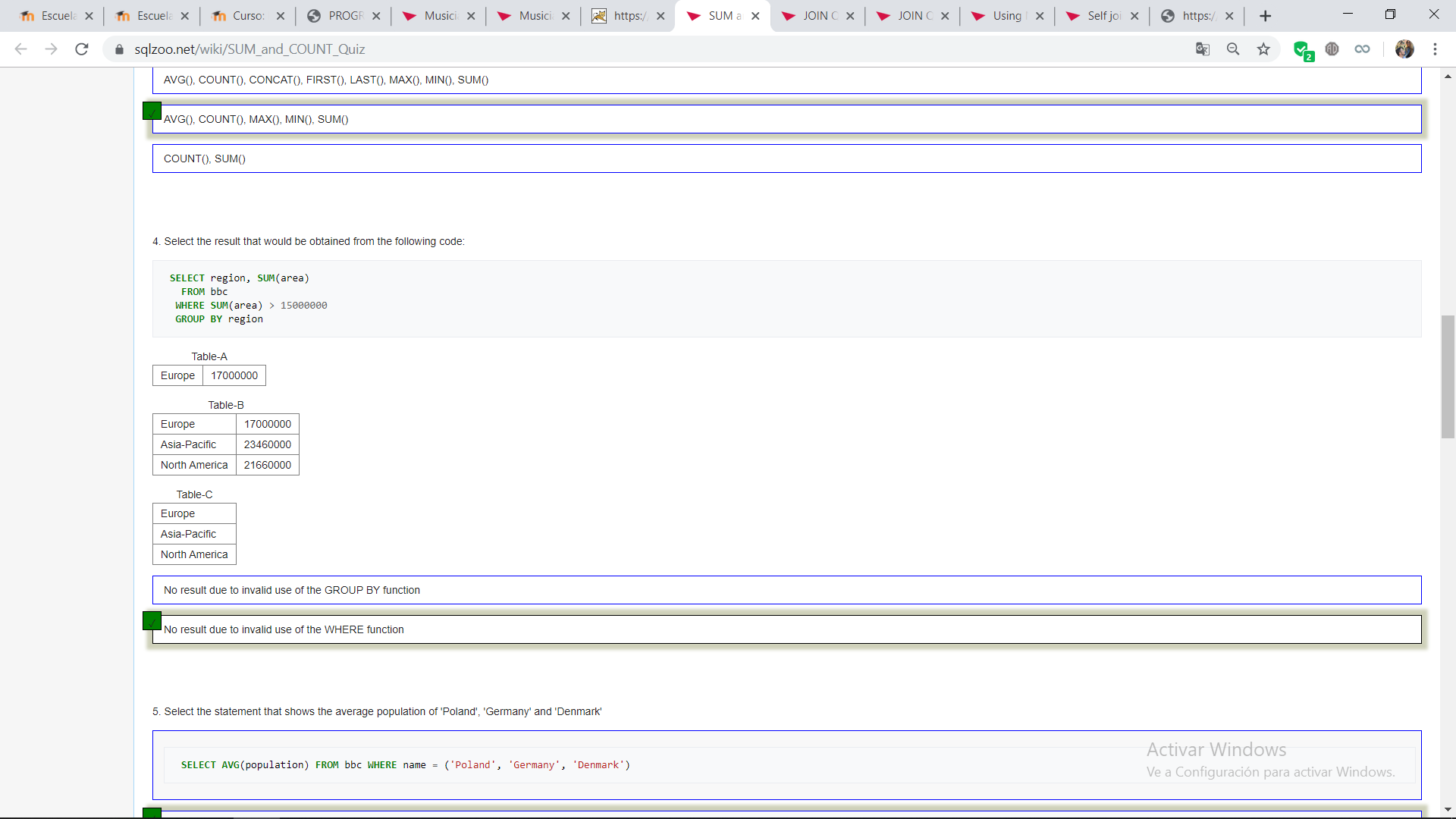
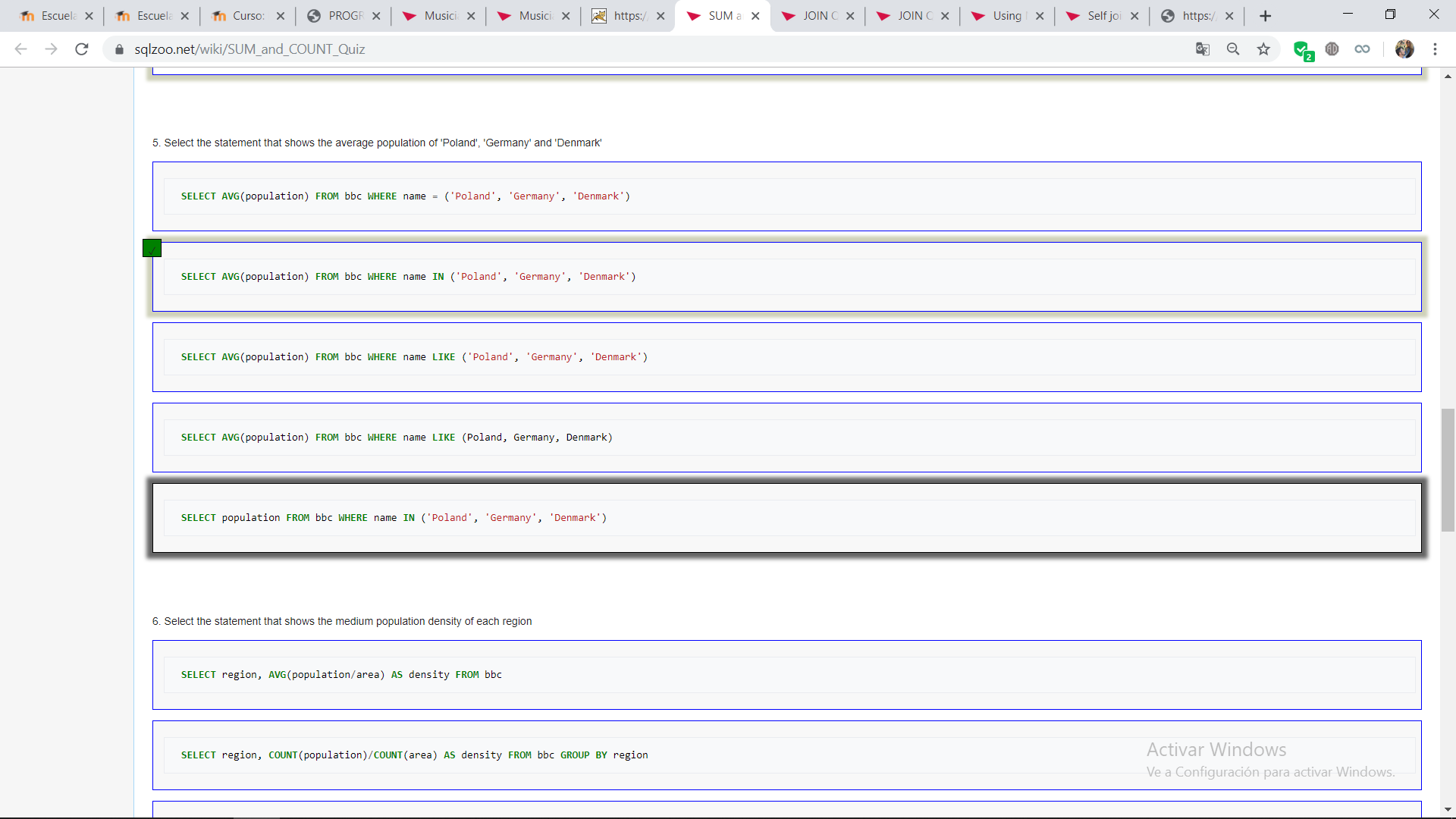
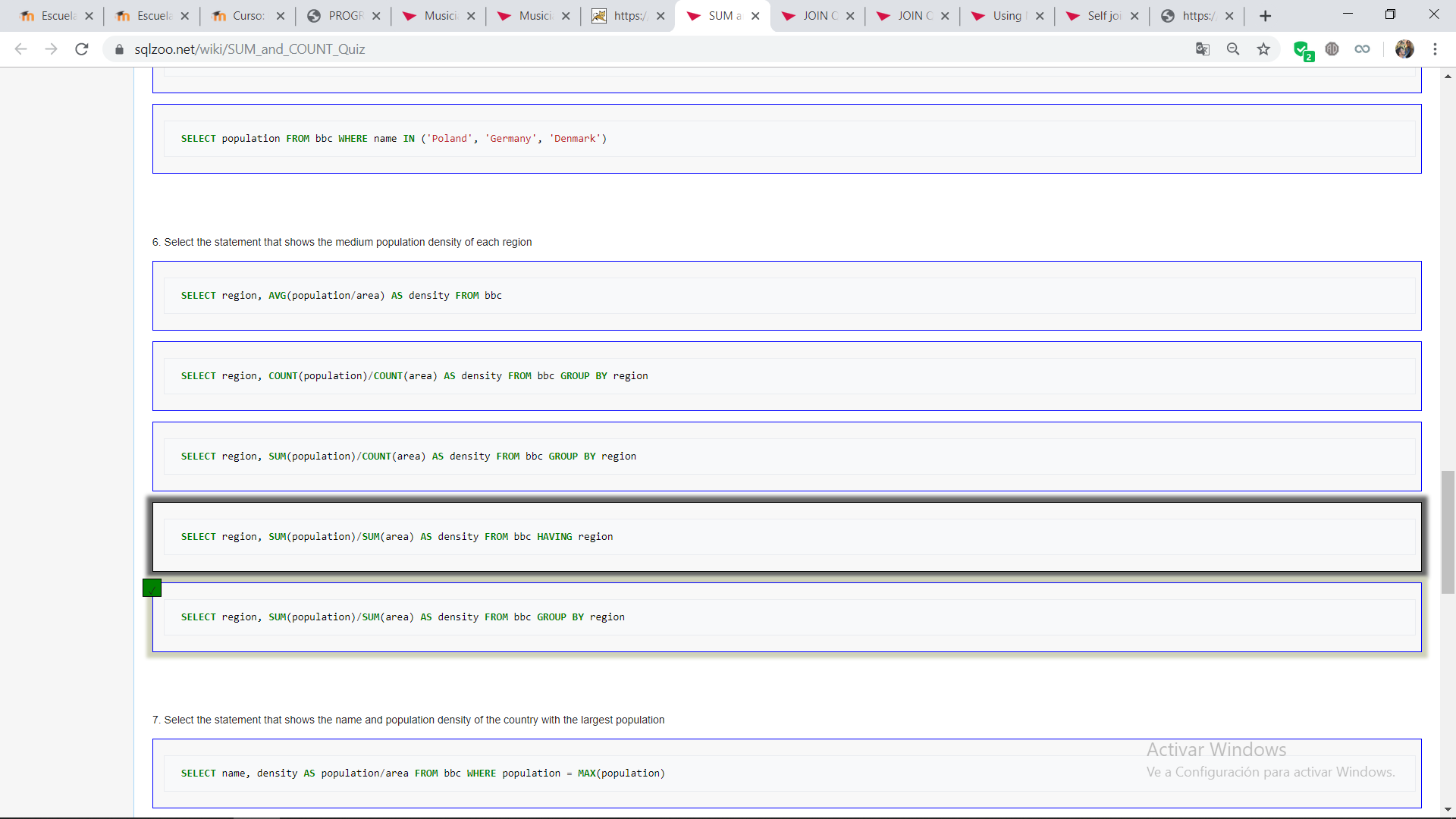
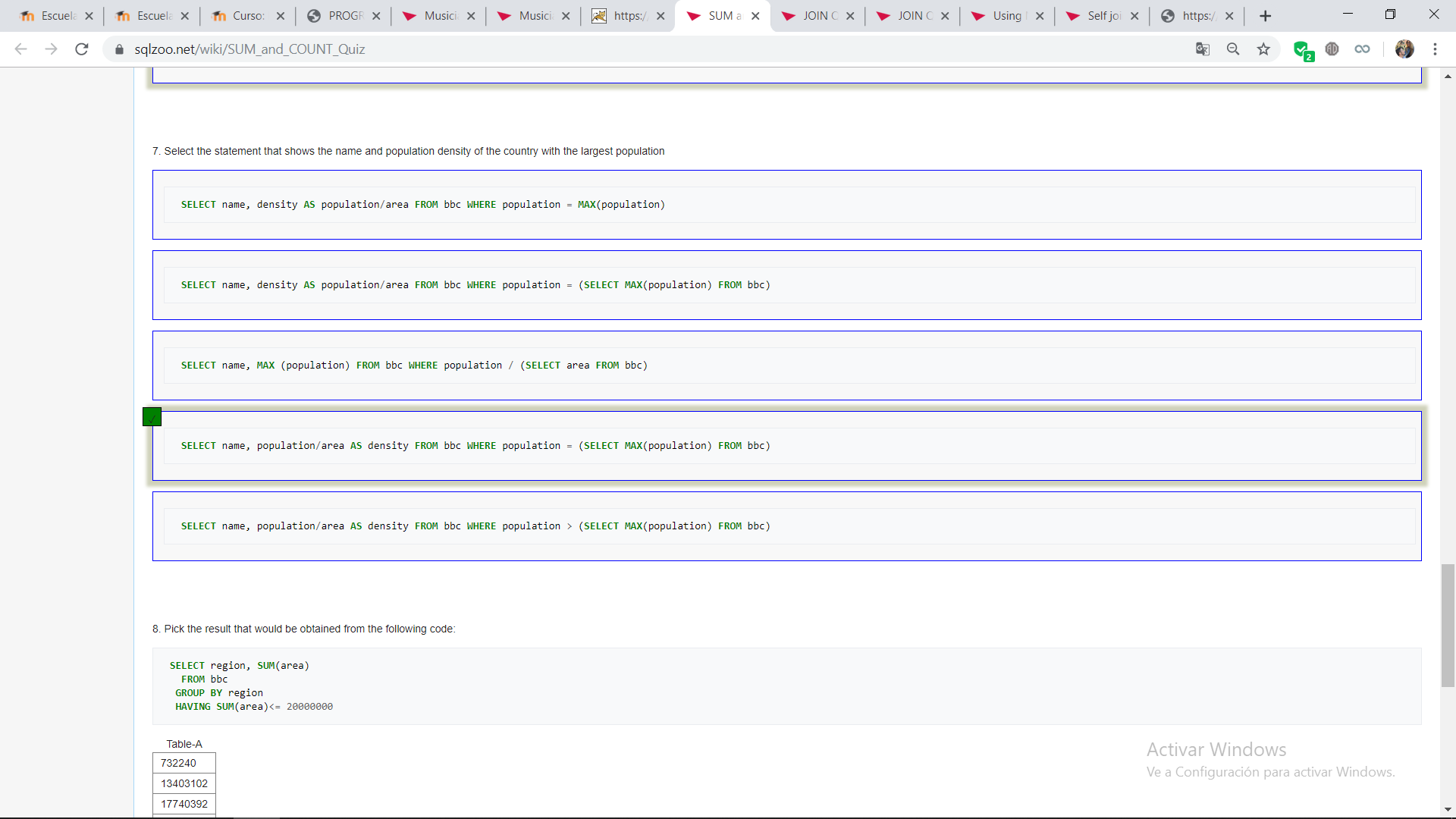
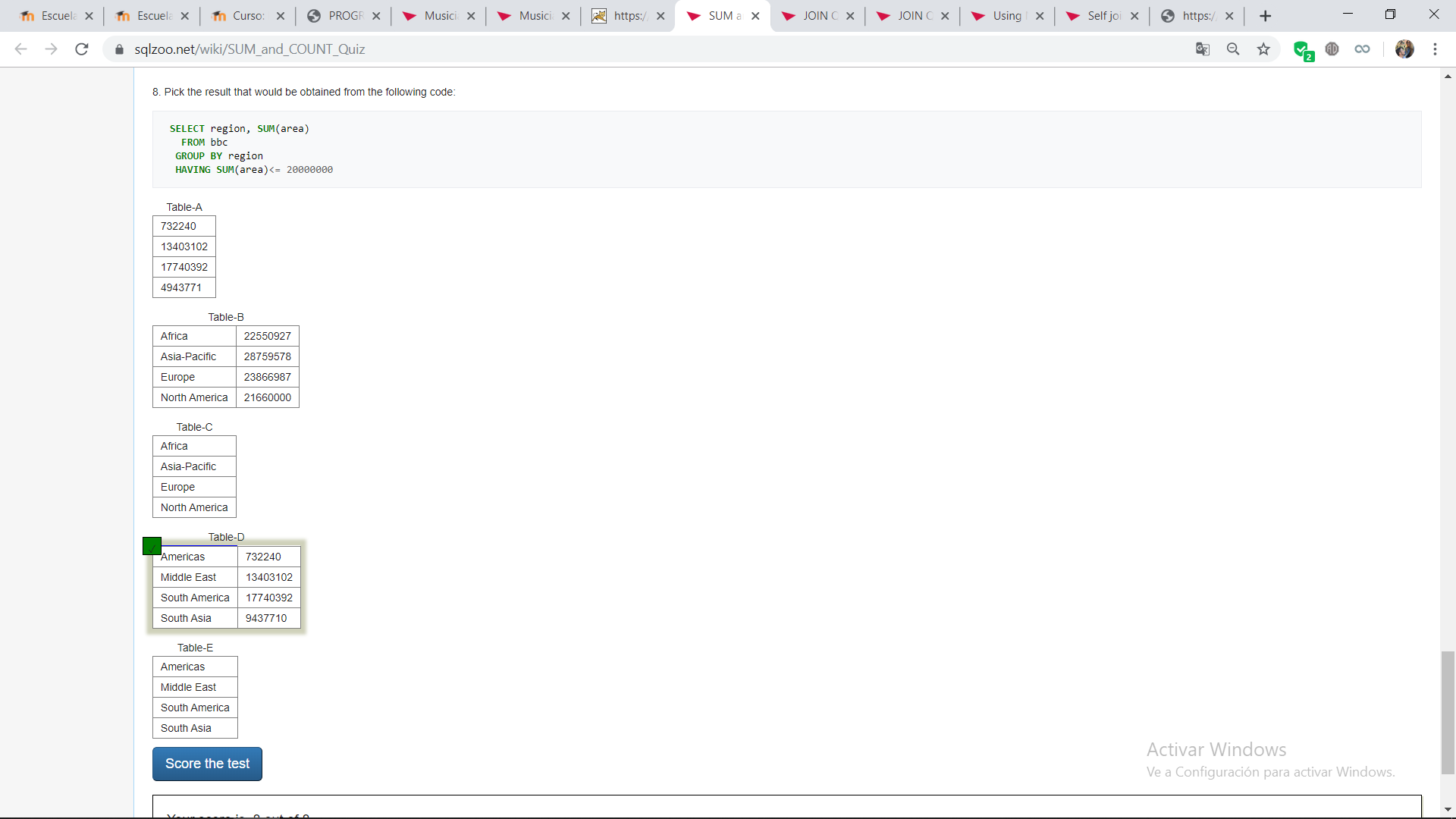
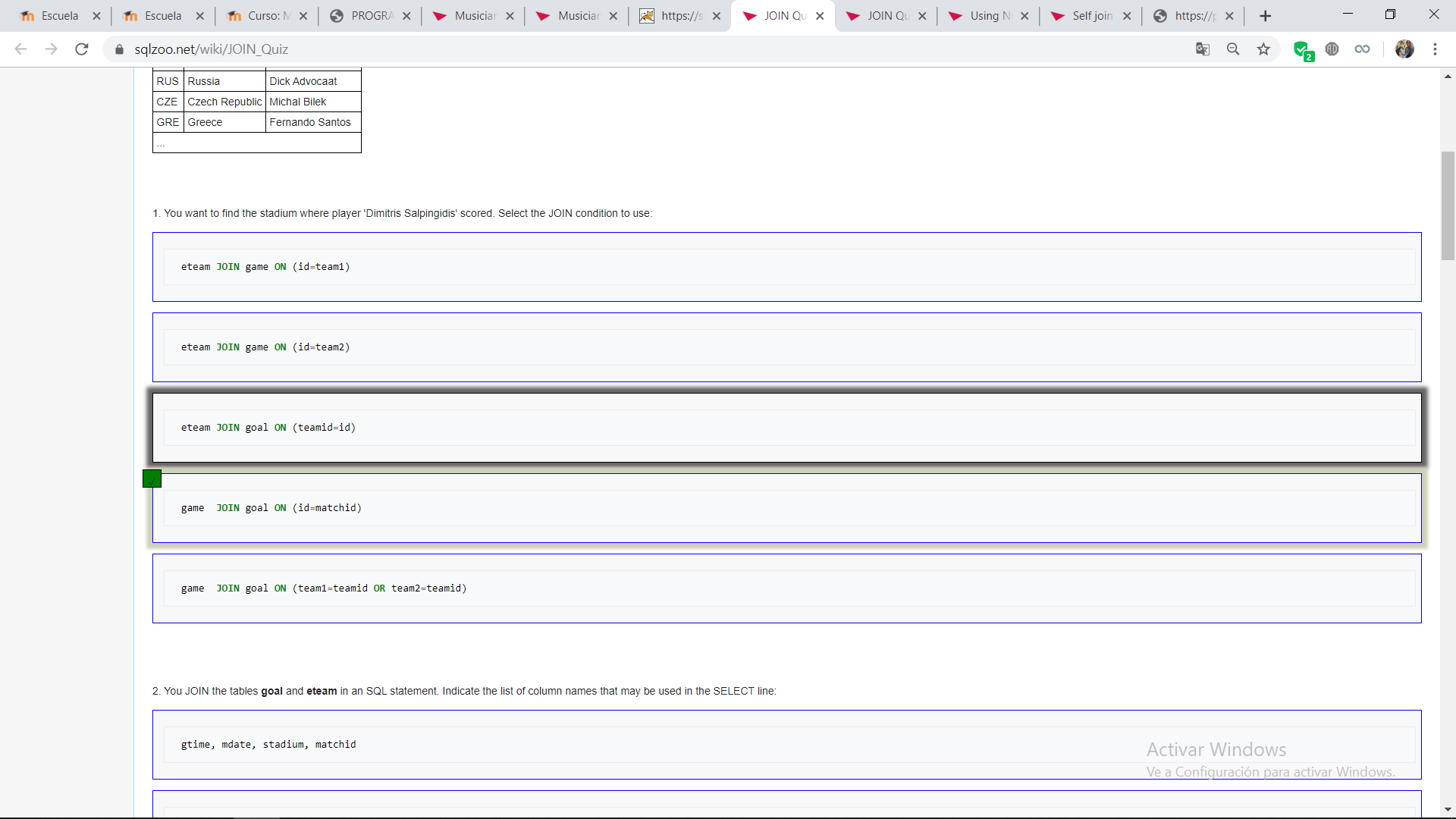
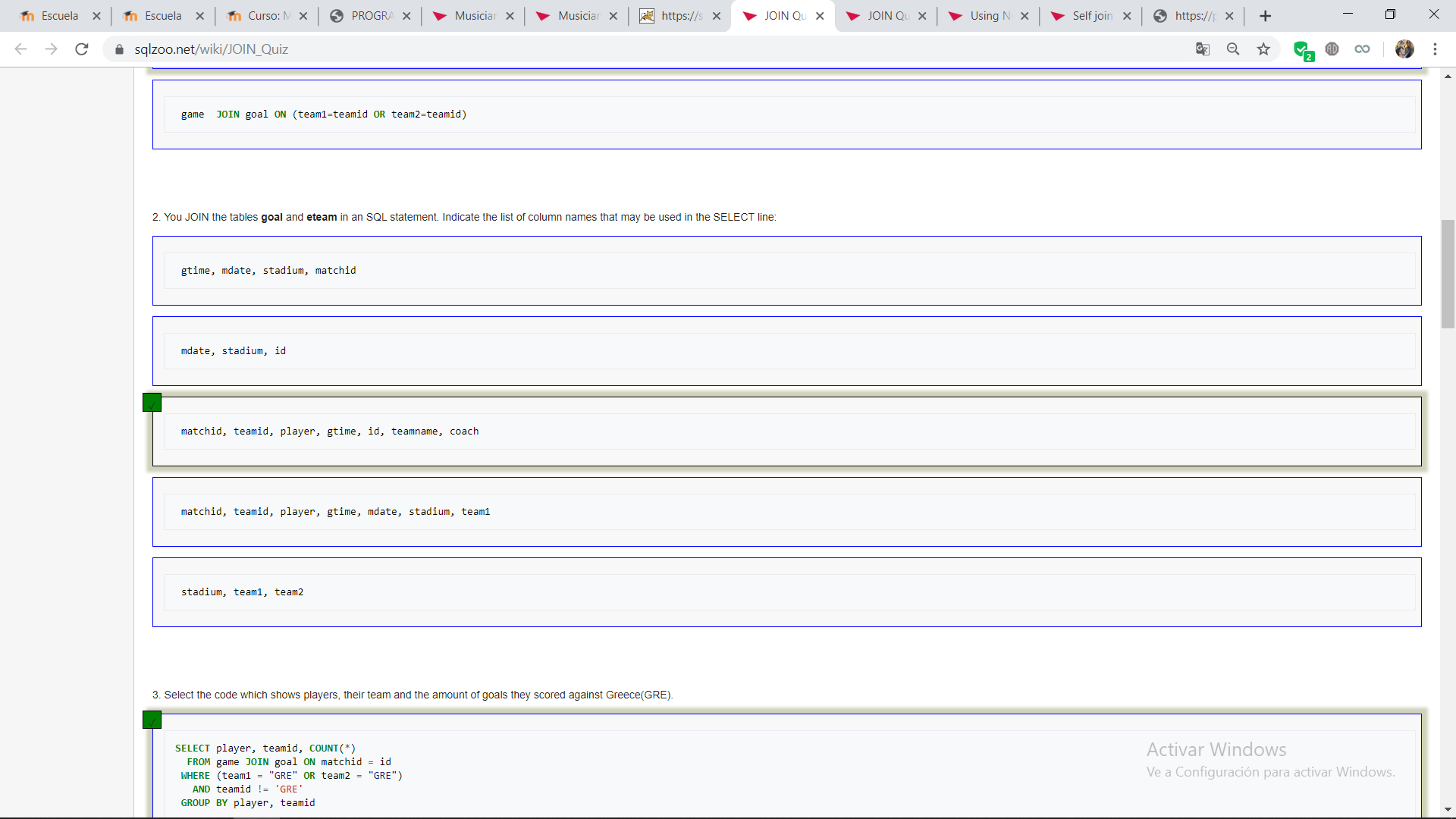
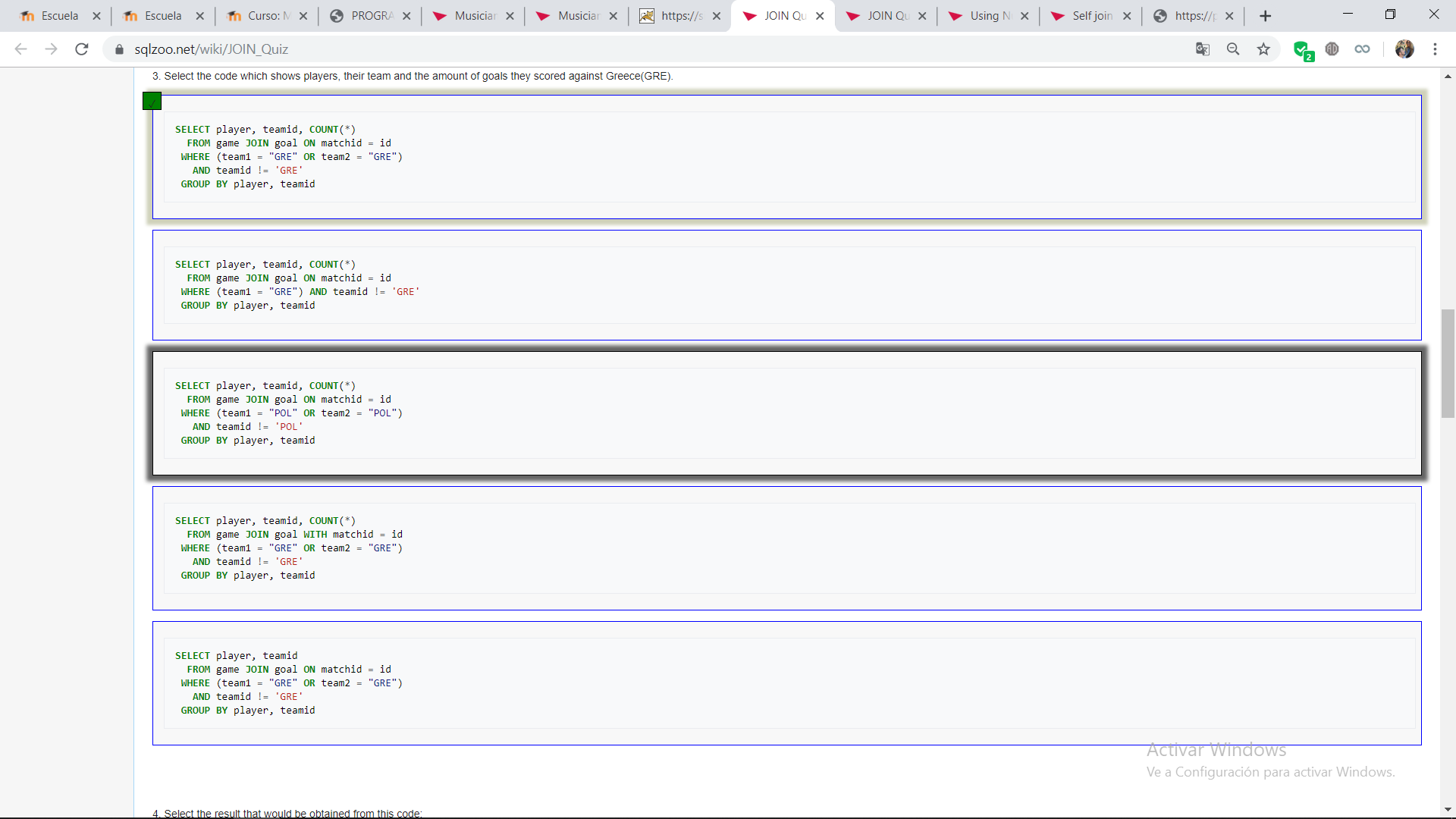
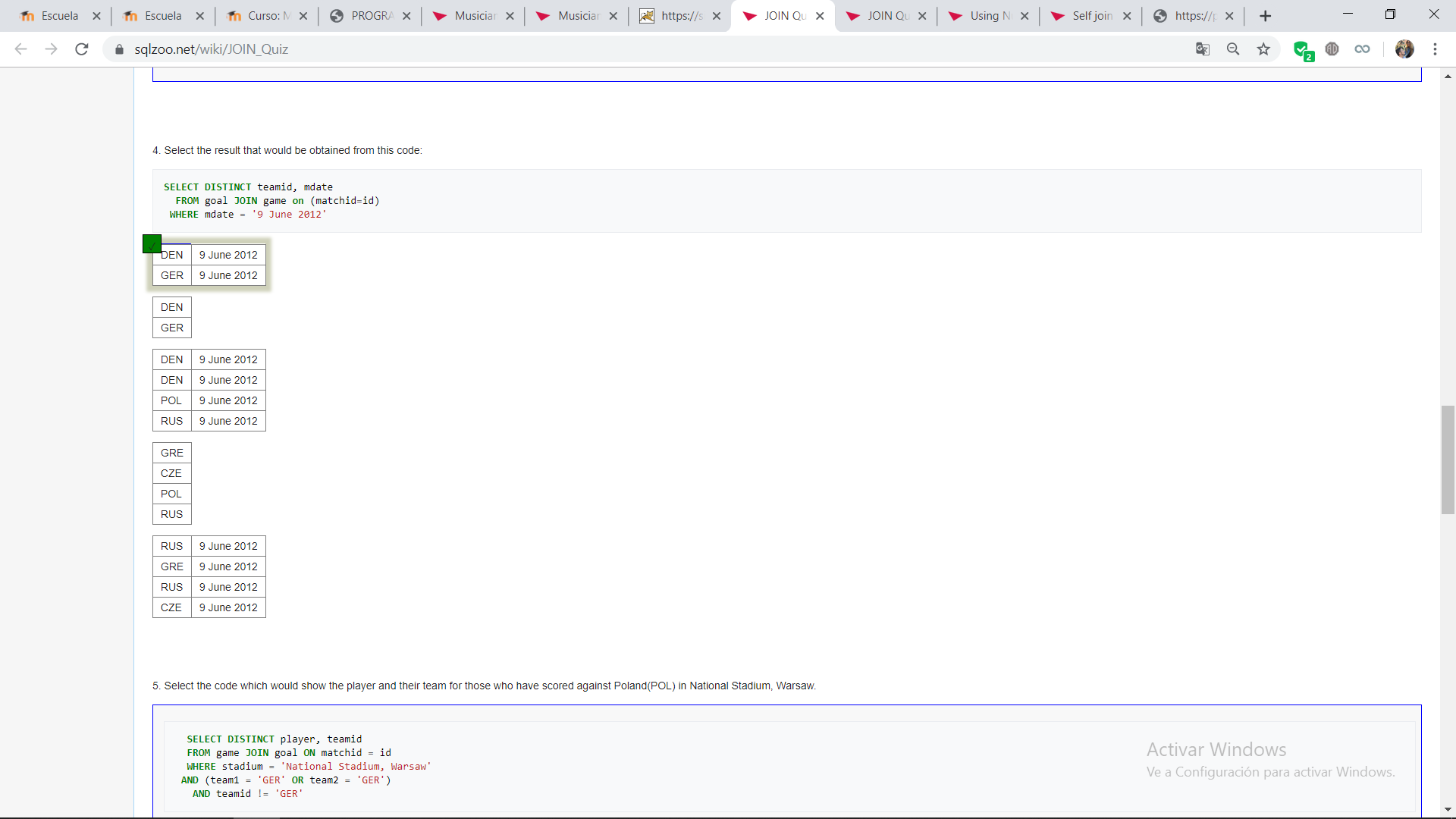
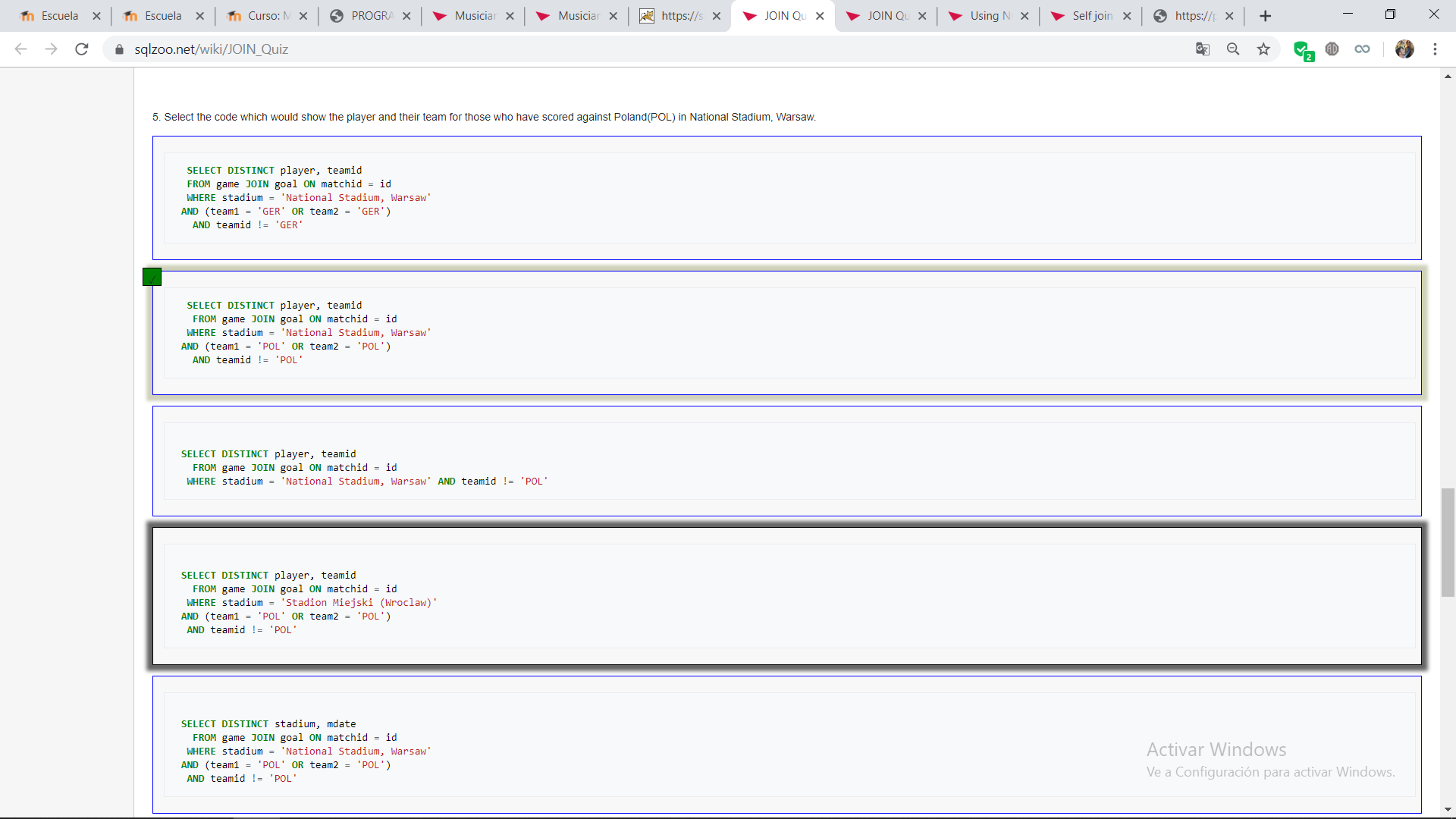
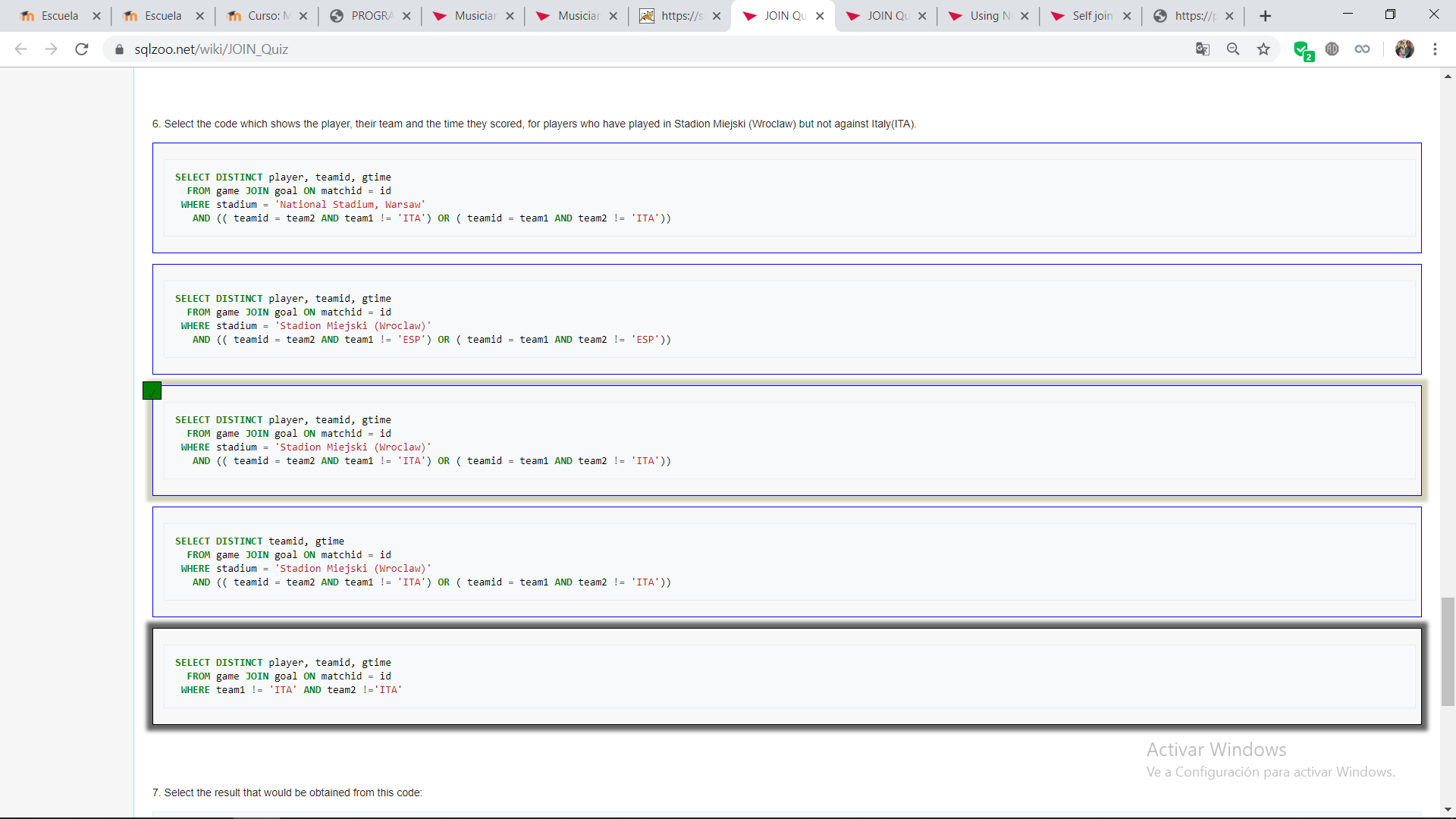
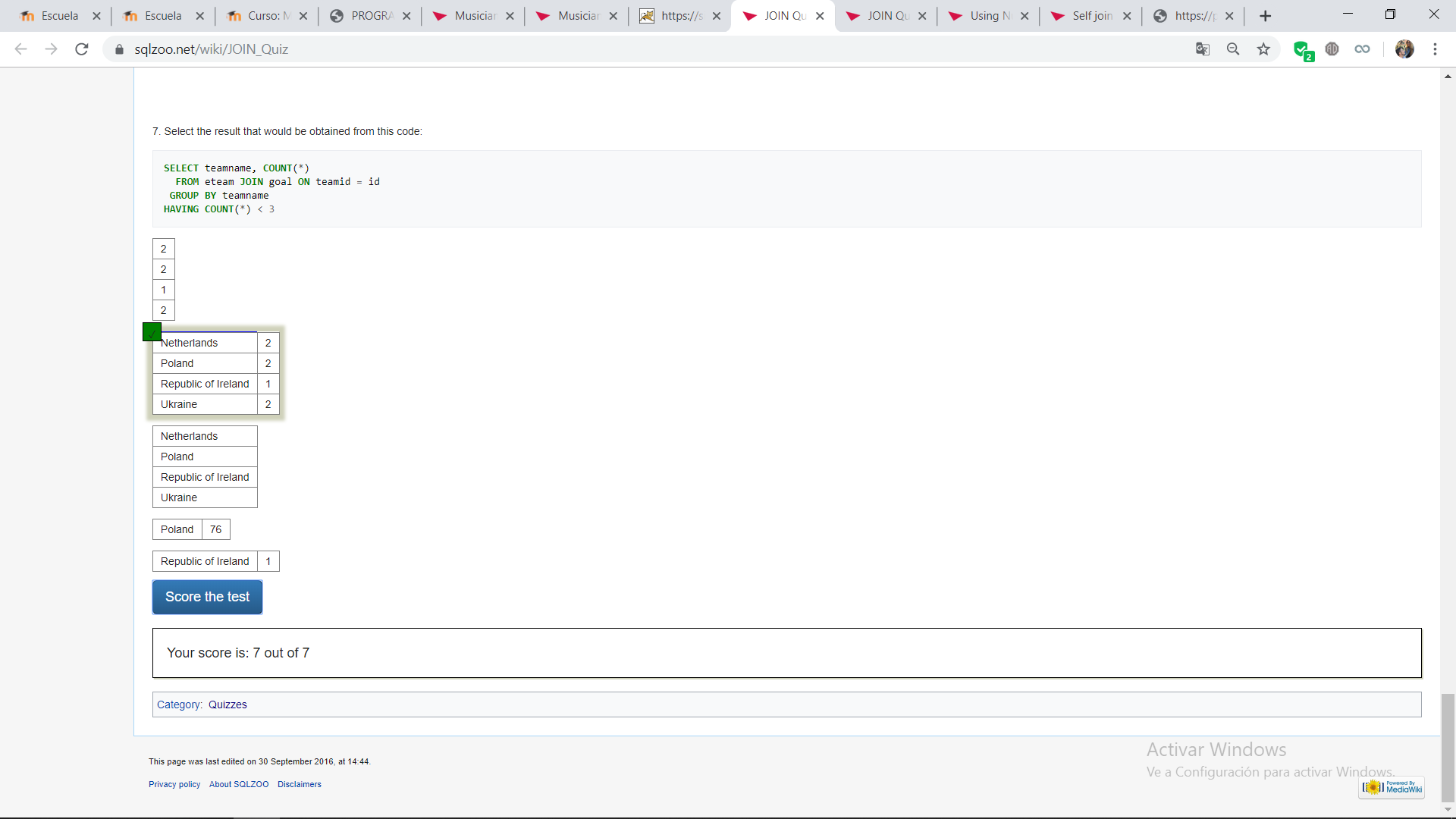
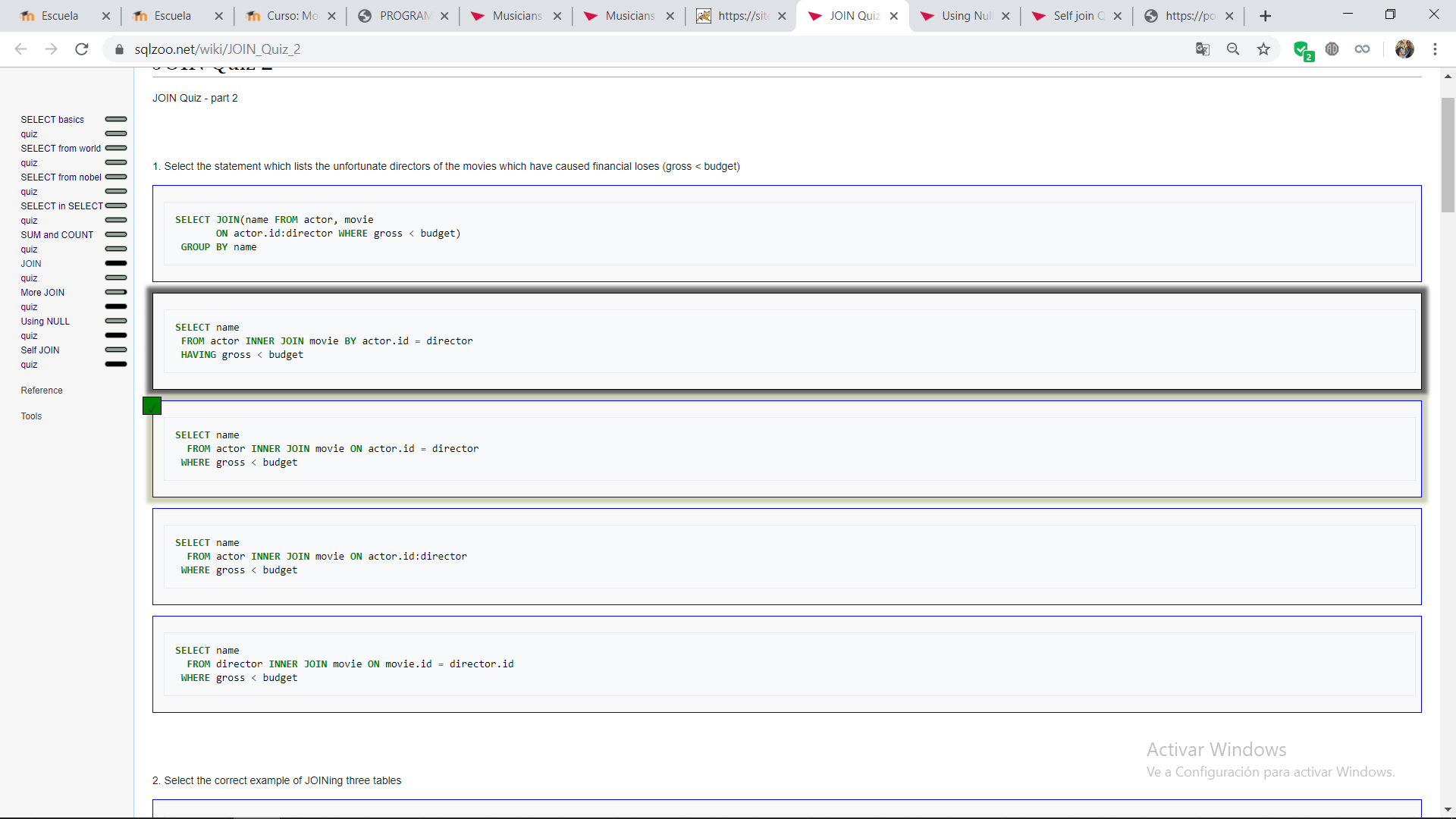
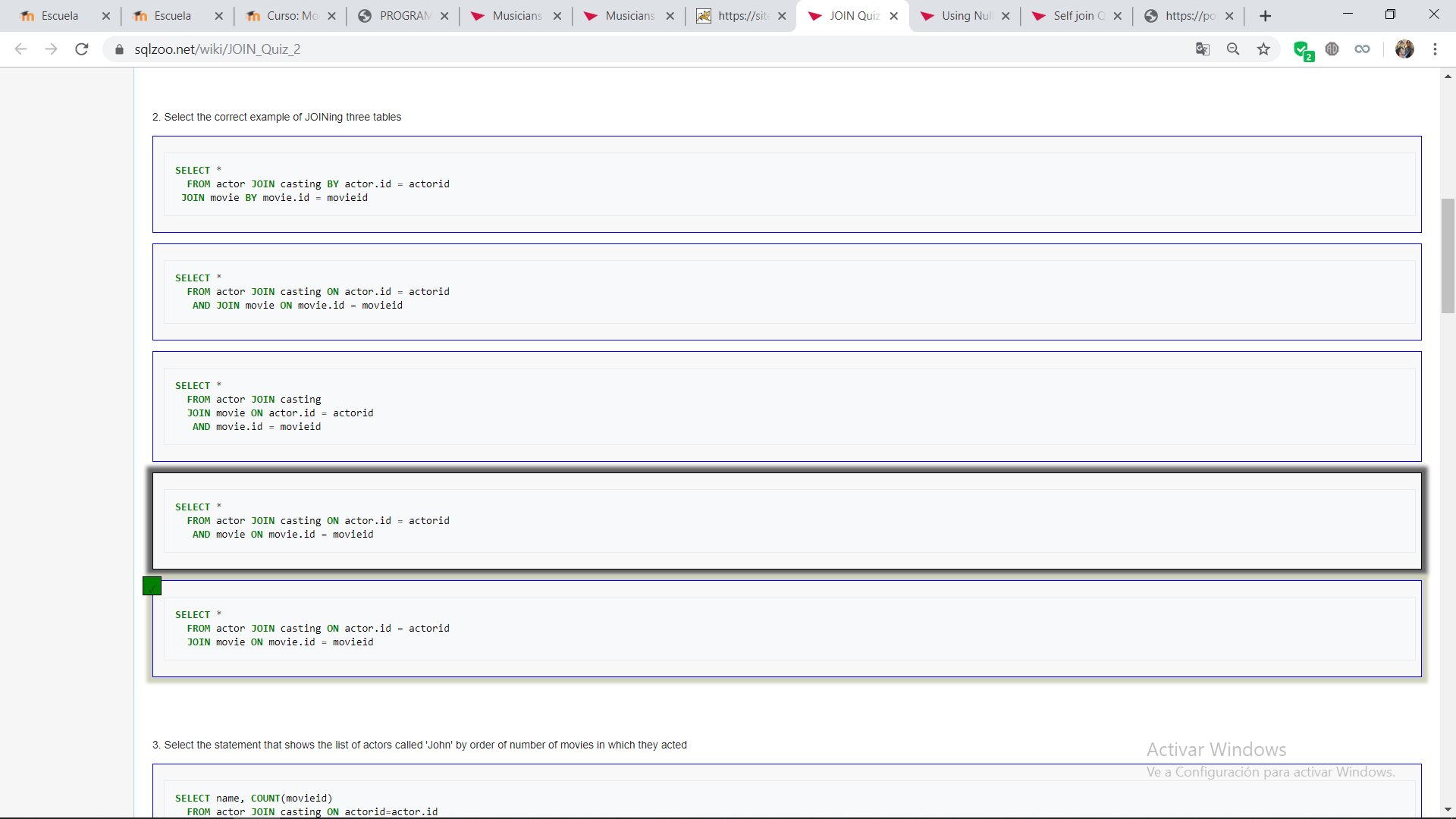
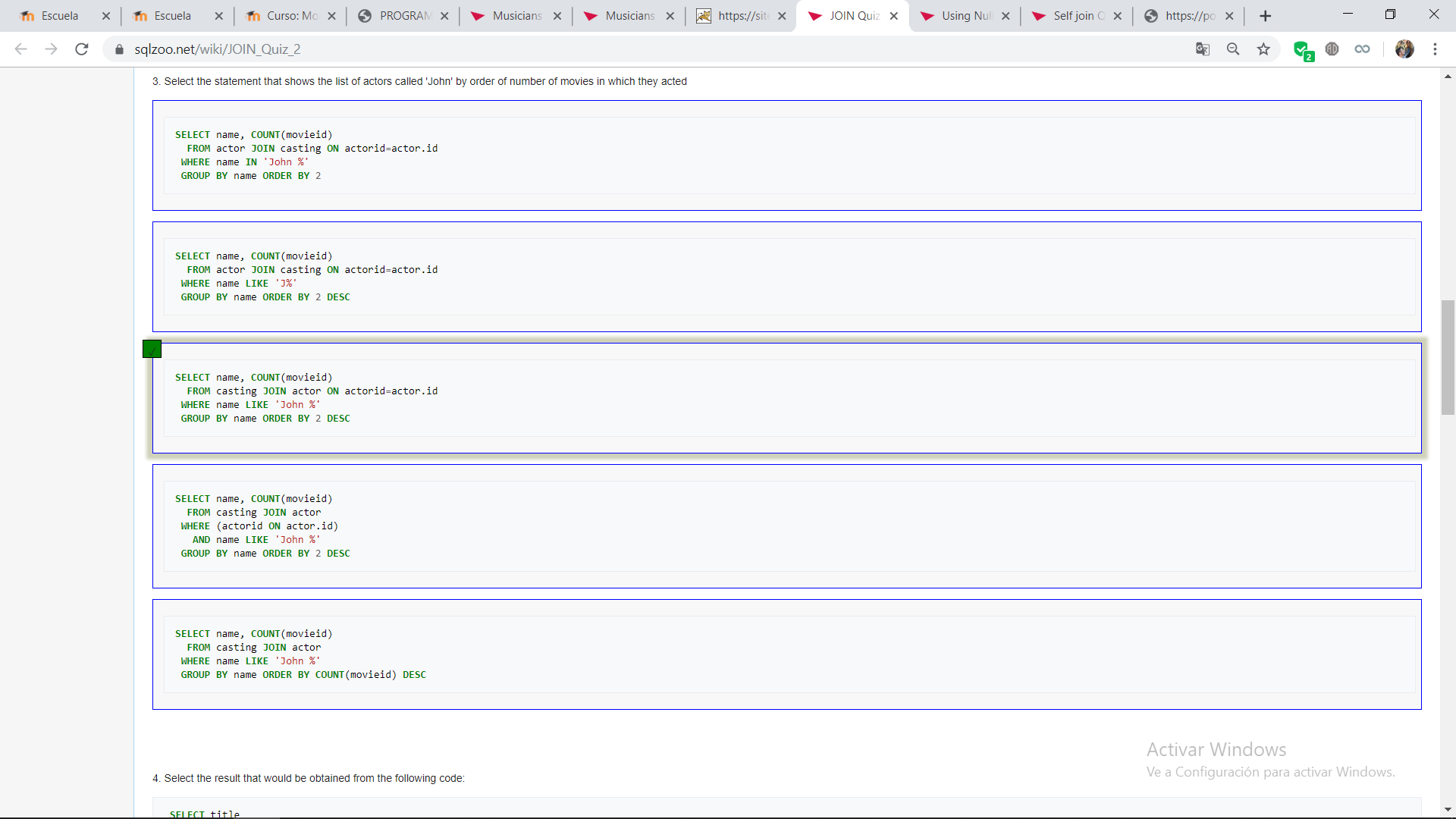
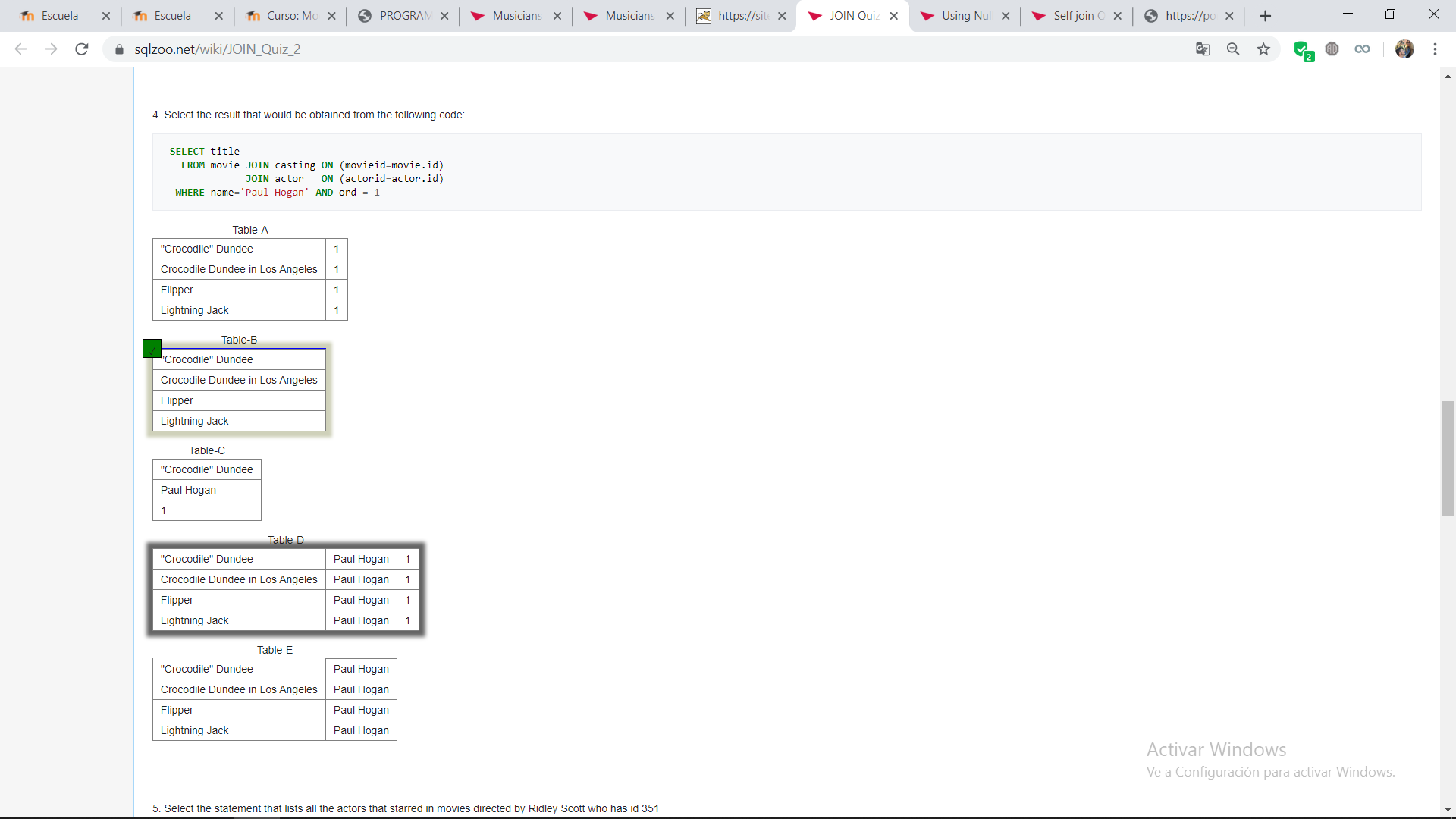
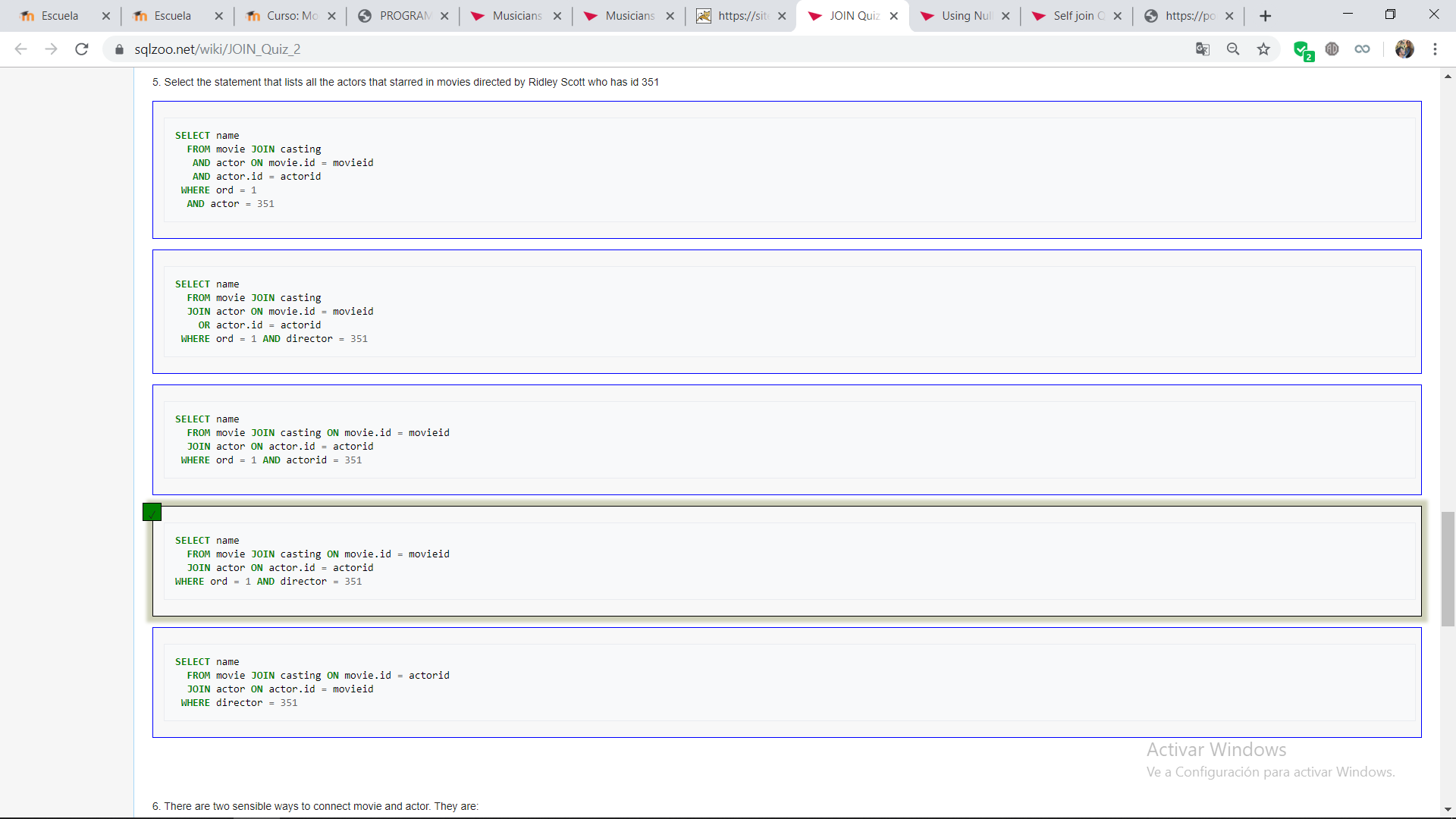
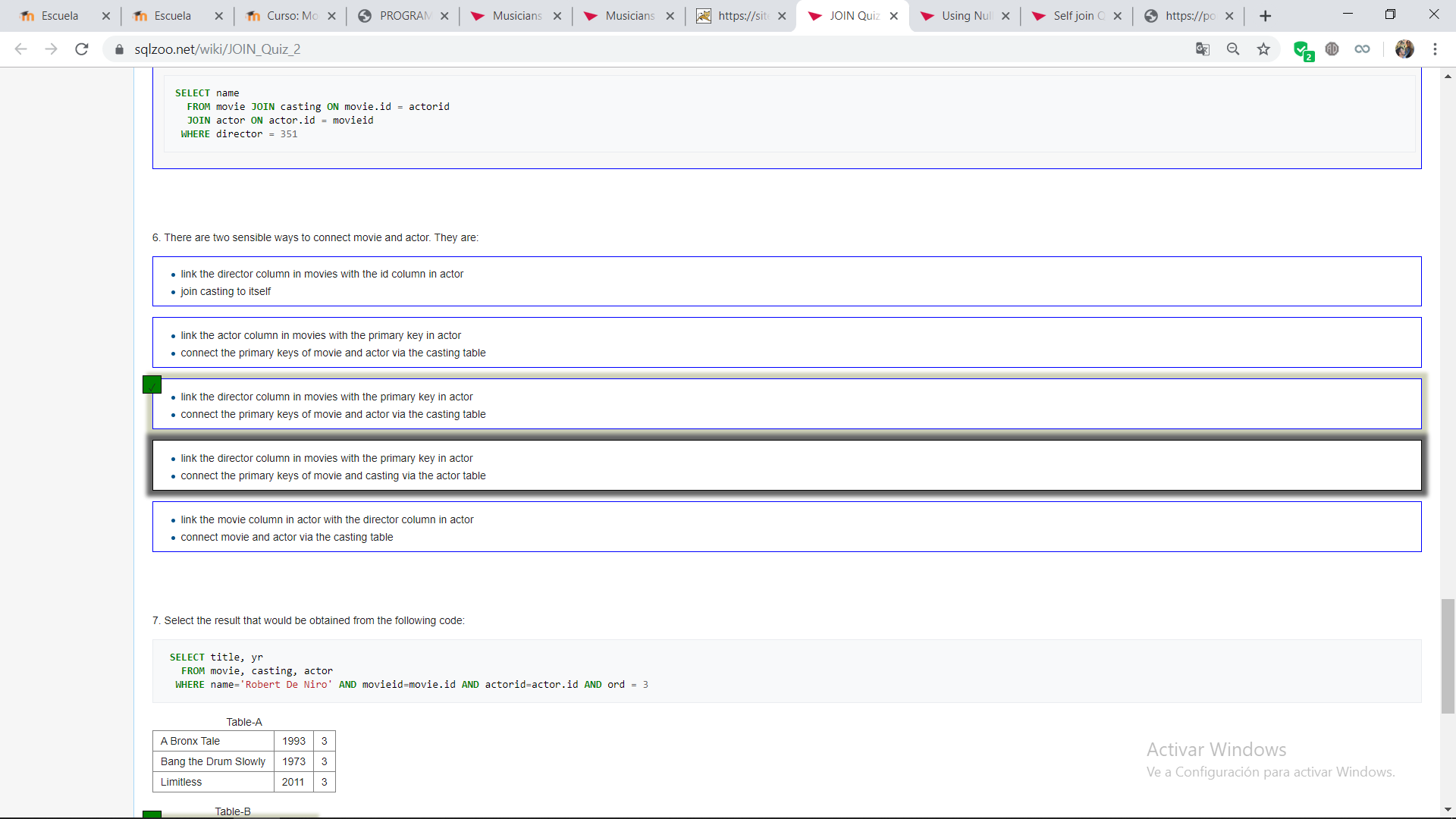
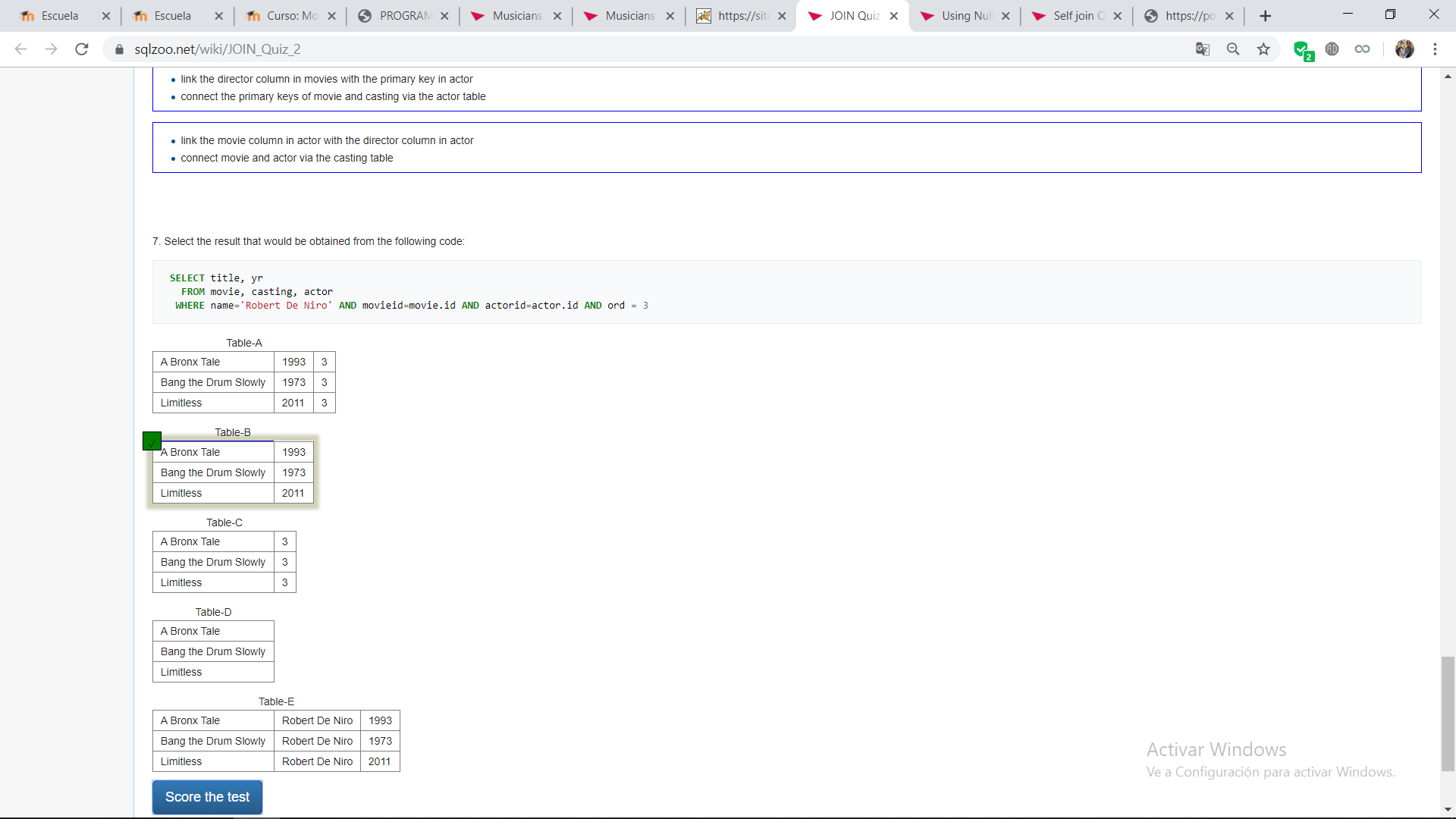
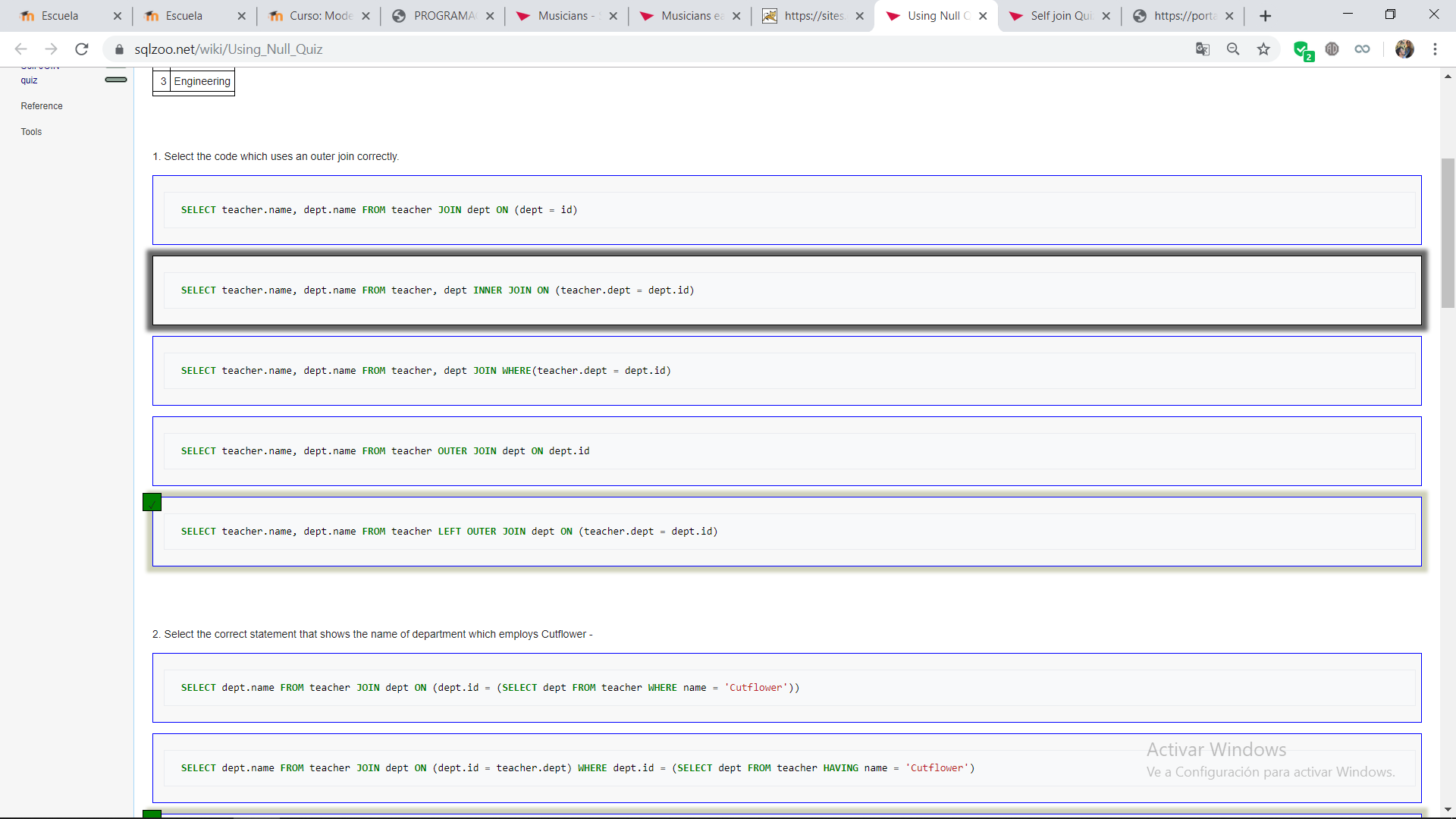
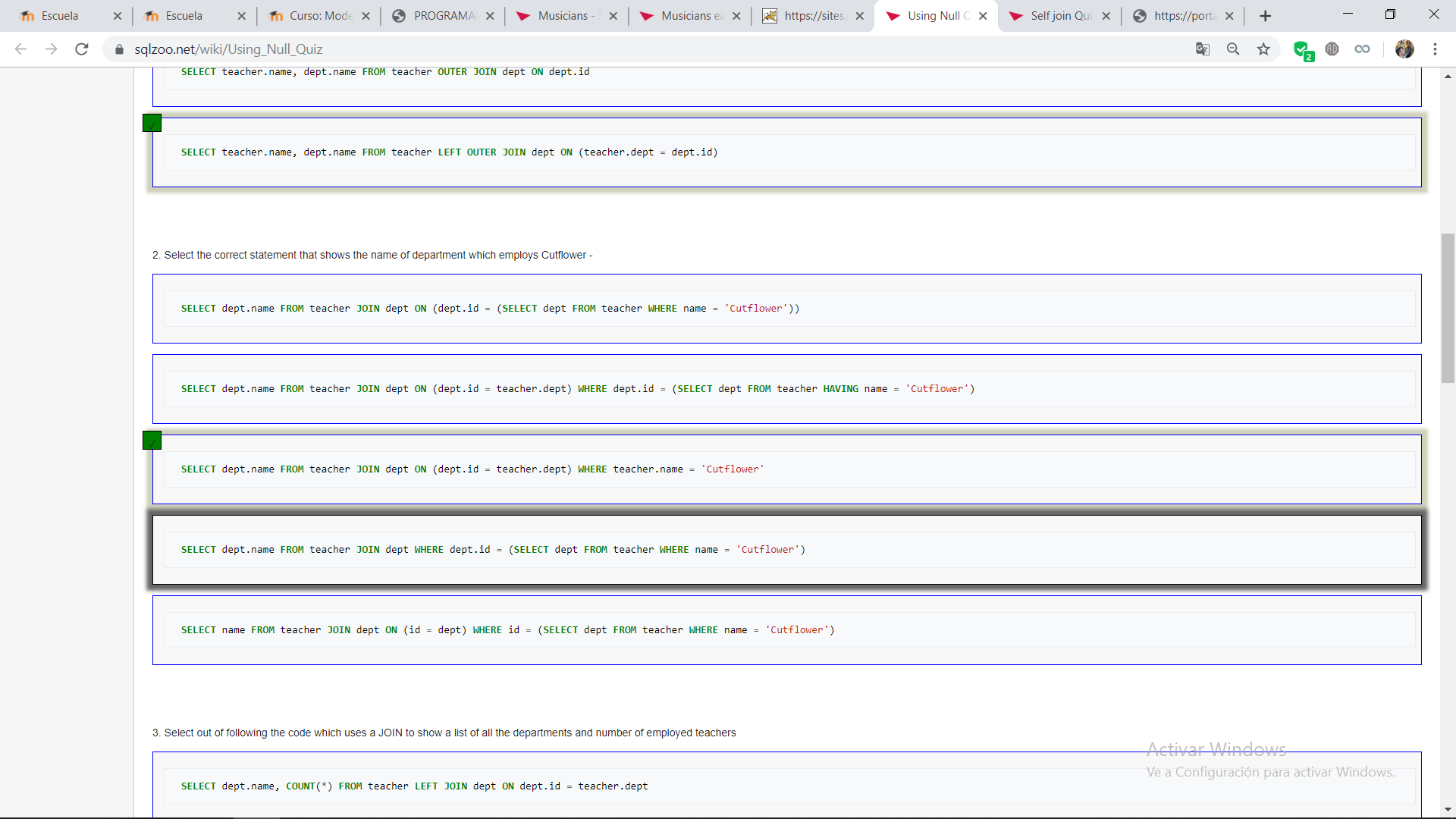
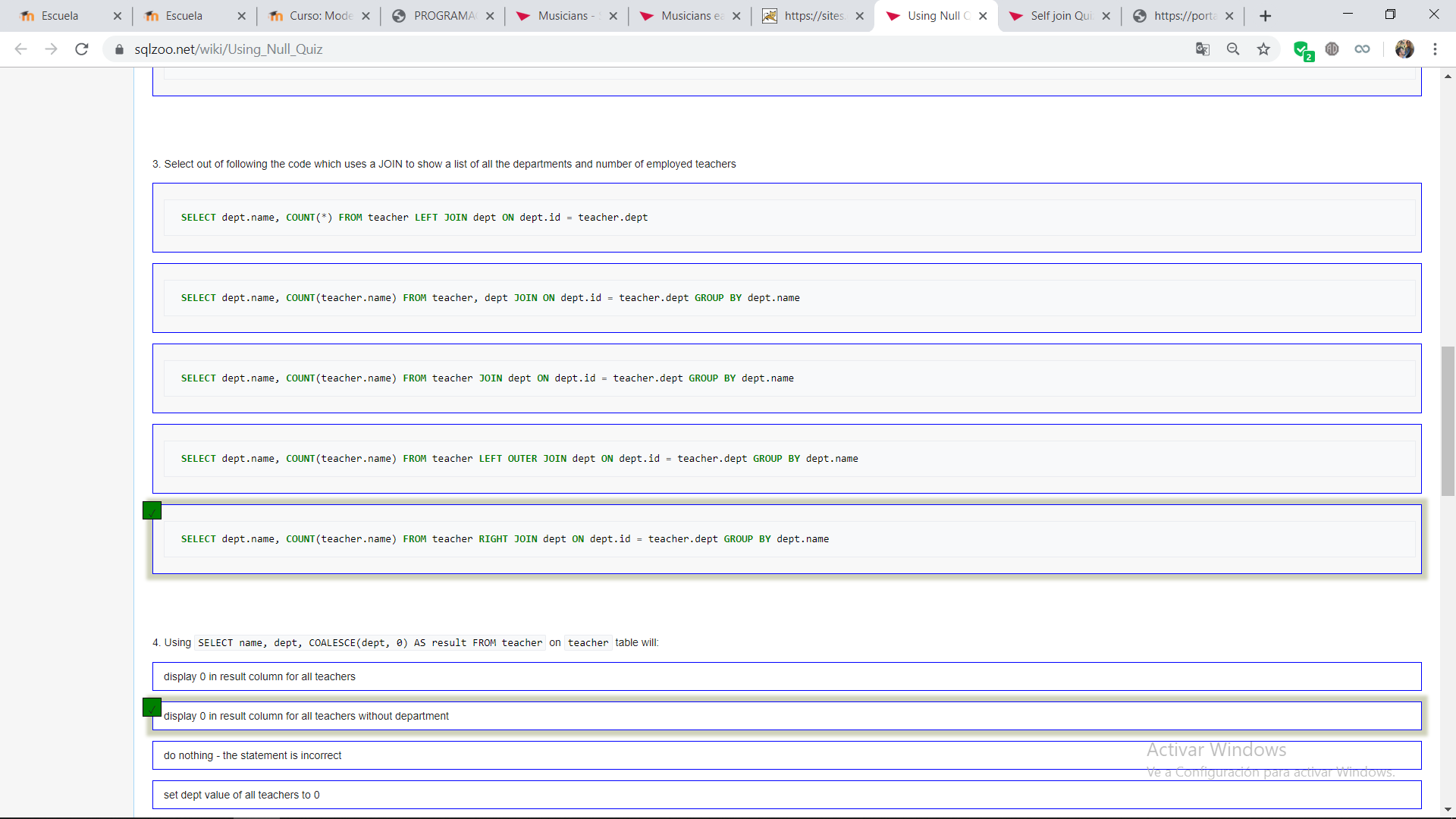
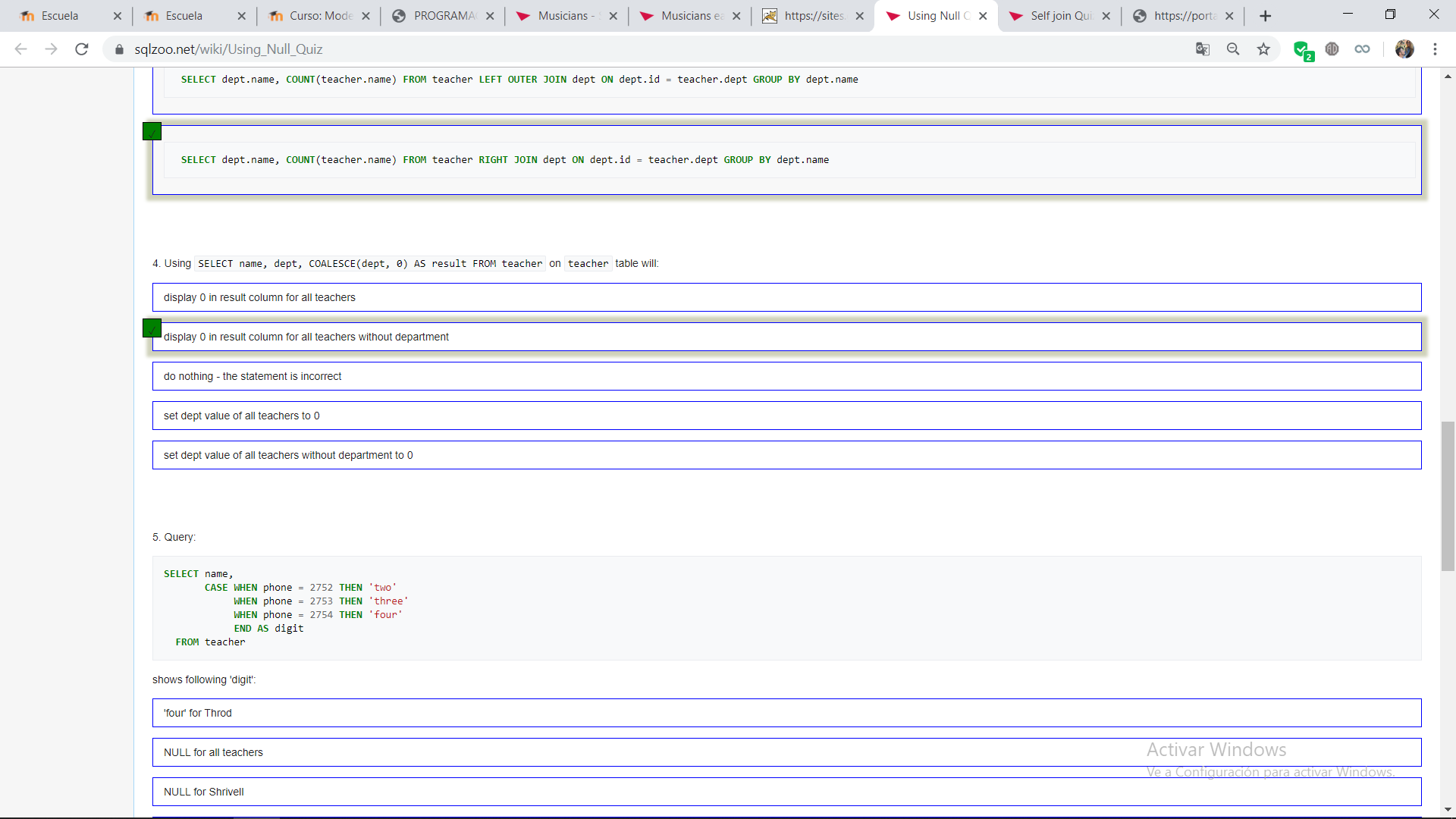
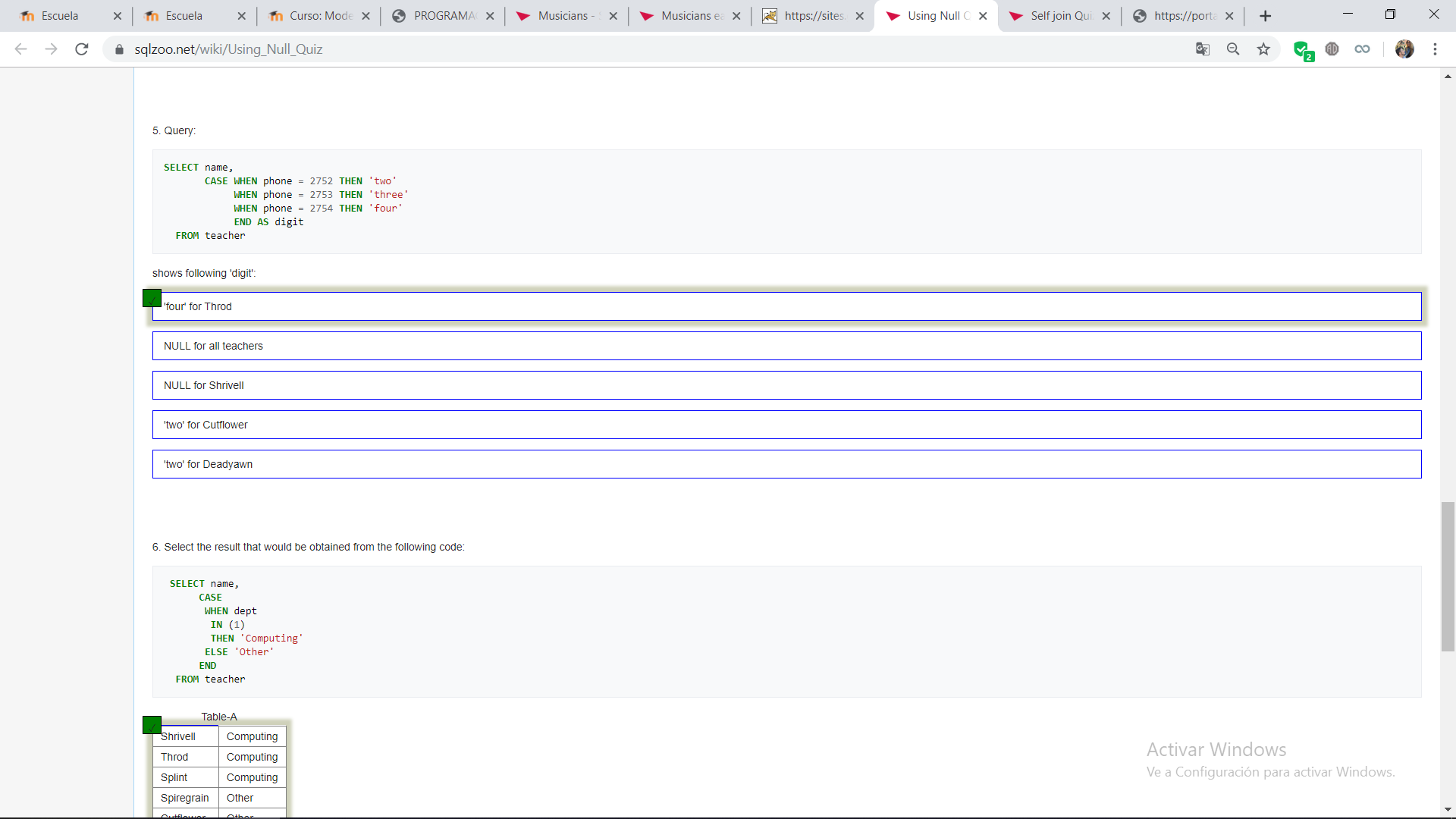
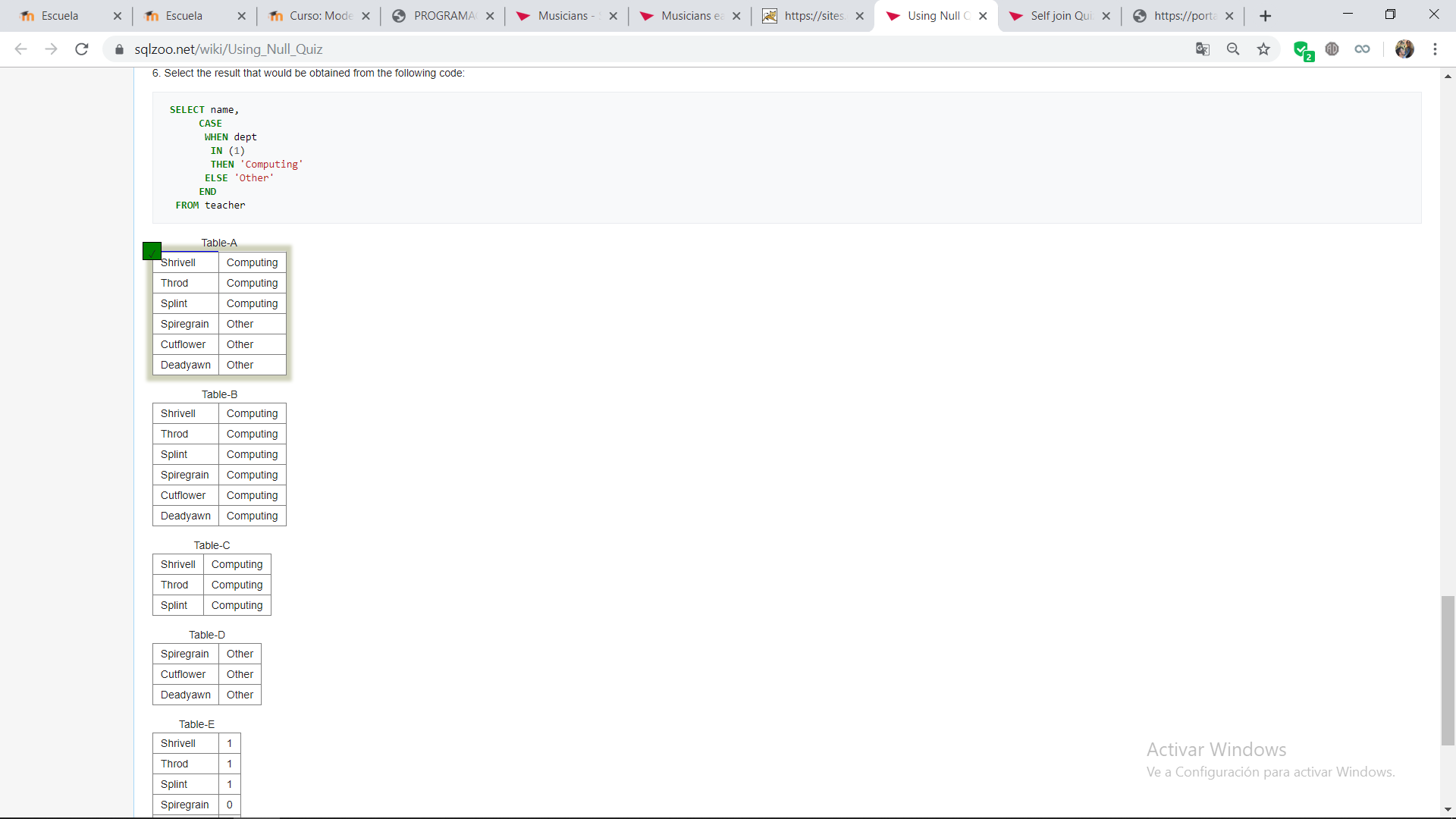
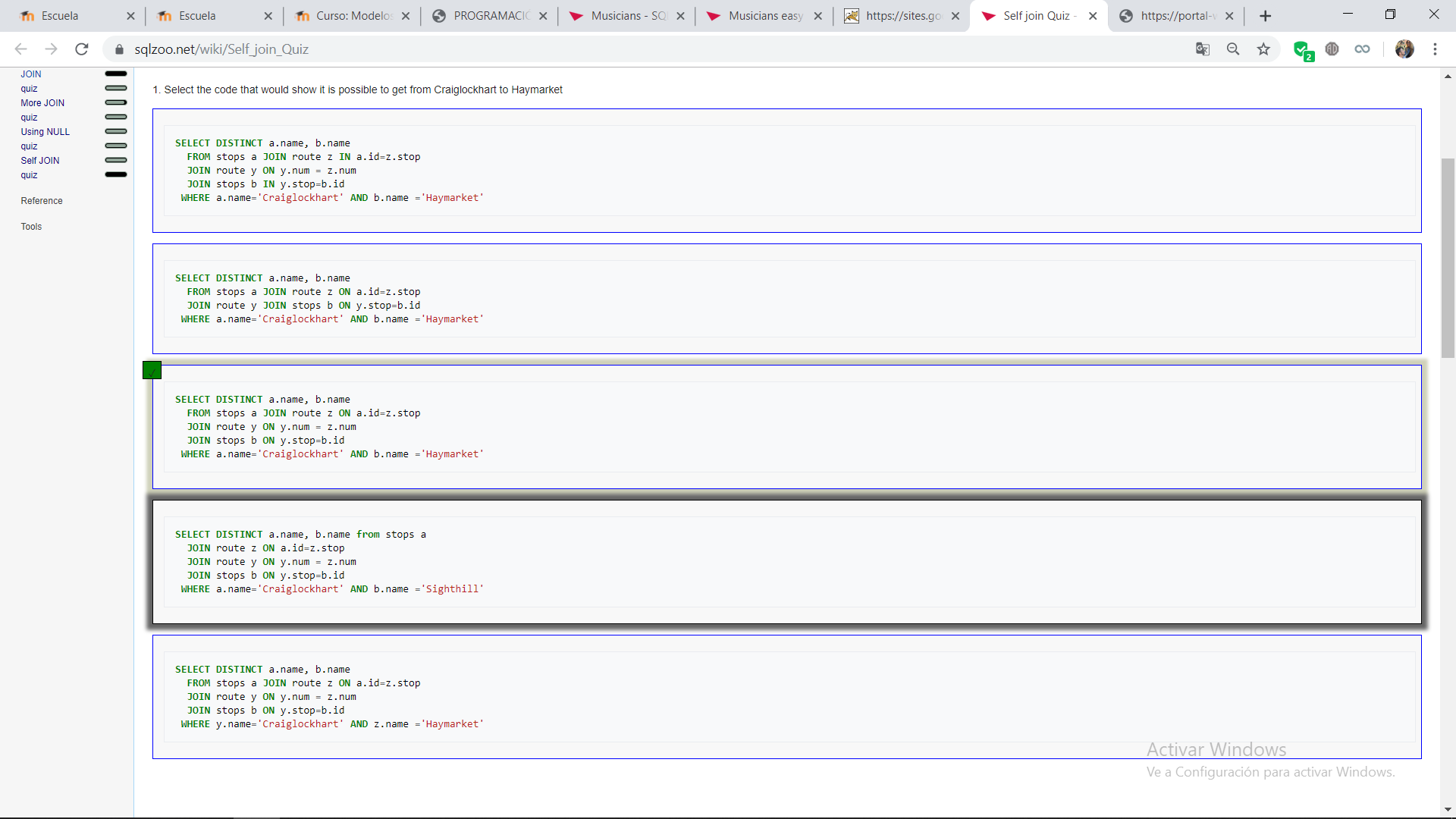
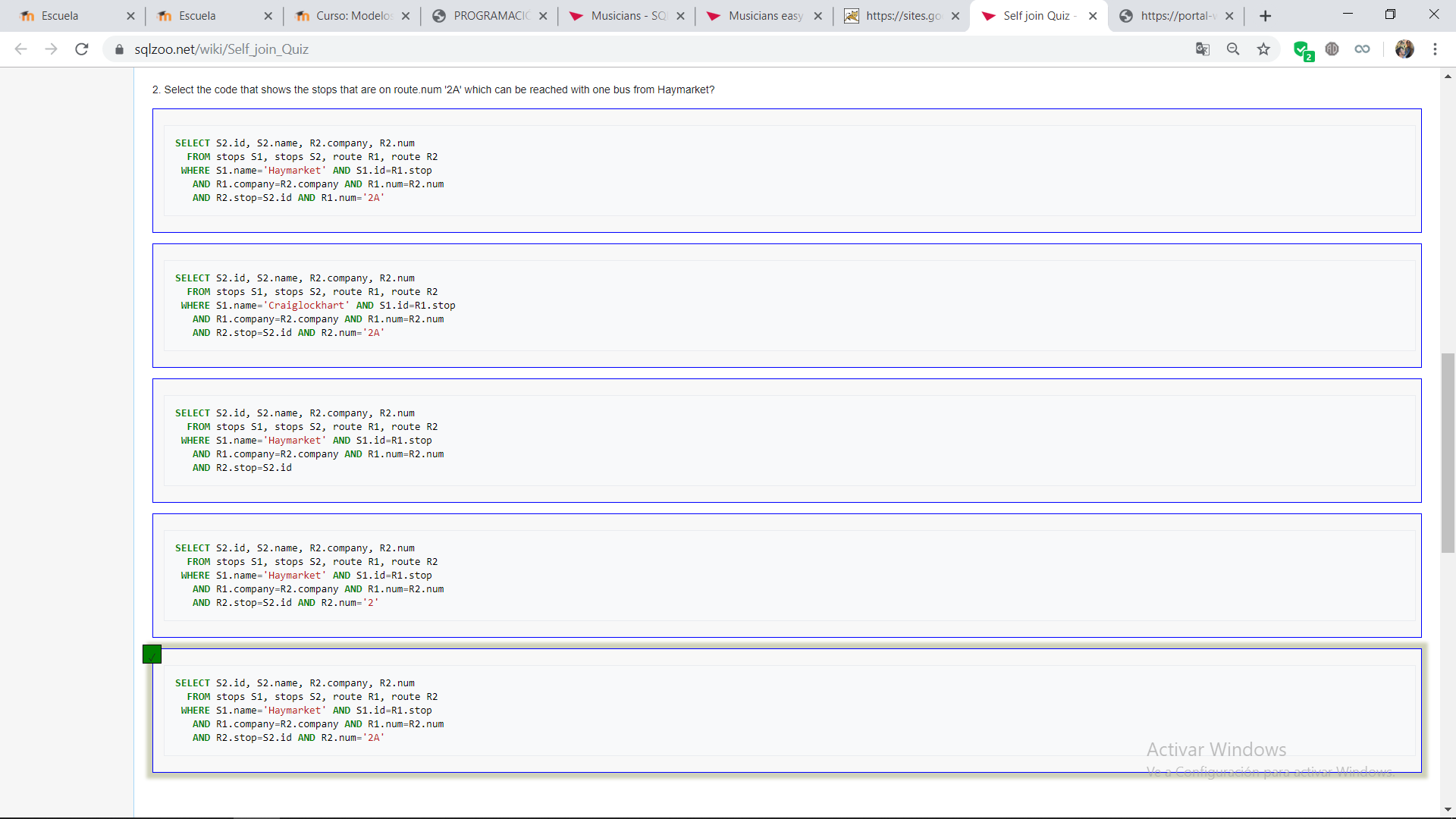
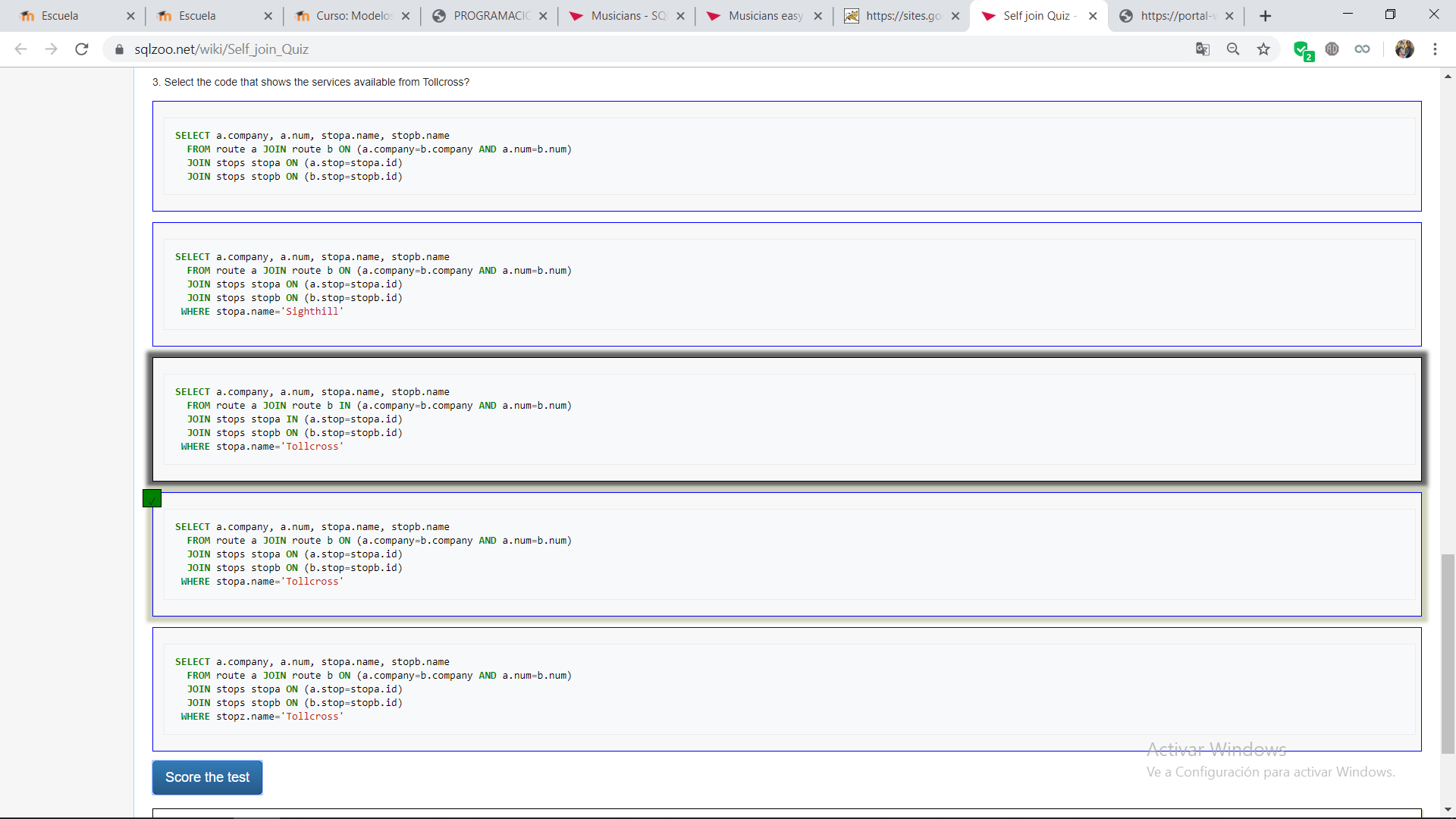
(d.stop = stopd.id)

WHERE stopa.name = 'Craiglockhart' AND stopd.name =

'Lochend' AND

stopb.name = stopc.name ORDER BY LENGTH(a.num), b.num,

stopb.id, LENGTH(c.num), d.num

1. **Tutorial quiz :** 
   1. **Select quiz:**
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
   2. **BBC QUIZ**
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
   3. **NOBEL QUIZ**
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
   4. **NESTED SELECT QUIZ**
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
   5. **SUM AND COUNT QUIZ**
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
      8. 
   6. **JOIN QUIZ** 
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
   7. **JOIN QUIZ 2**
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
   8. **USING NULL**
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
   9. **SELF JOIN QUIZ**
      1. 
      2. 
      3. 
2. **Consultas inventadas:**
   1. **Conjuntos:**
      1. **UNION**

select m\_name from musician where born\_in=1

union

select m\_name from musician where living\_in=1

* + 1. **INTERSECCIÓN**

select distinct m\_name from musician join performer on m\_no=perf\_is

* + 1. **DIFERENCIA**

select m\_name from musician where m\_no>5 and m\_name not in(select m\_name from musician where m\_name like 'A%')

* + 1. **CONCATENACION NATURAL**

select m\_name,instrument from musician natural join performer

* + 1. **EFICIENCIA**

select m\_name from musician where exists(select \* from performer where m\_no=perf\_is)

* 1. **JUNTA INTERNA**
     1. select place\_country,m\_name from place inner join musician on place\_no=born\_in
     2. select instrument,m\_name from musician inner join performer on m\_no=perf\_is
  2. **JUNTA EXTERNA** 
     1. select m\_name,instrument,died from musician left outer join performer on m\_no=perf\_is
     2. select band\_name,place\_town,place\_country from band right outer join place on band\_home=place\_no
  3. NULL