**Modelos y bases de datos**

**2019-2**

**Autoestudio 2**

**NOMBRES:**

Ramírez Pinto Fabian Mauricio

Rincón Saavedra Iván Camilo

**INVESTIGACIÓN**

**A) \_ NULL 1.**

**¿Qué significa?**

Null es un campo sin valor. Este es diferente a un valor cero o un campo que contiene espacios. El campo NULL es un campo que se ha dejado en blanco durante la creación de un registro.

El valor NULL no se puede probar con operadores de comparación, como =, <o <>. Se tendrá que usar una operación llamada IS NULL o IS NOT NULL para poder completar dicha operación.

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

* + Operadores aritméticos no se pueden realizar operaciones y por tanto serian un error.
  + Operadores lógicos si se podrían realizar operaciones como AND o OR o CASE entre otras.
  + Con operadores como lo son de comparación se podrían realizar ciertas operaciones como lo son = or <>, mas no se podrían hacer operaciones como lo son < o > o <= o >=.

**B. JUNTA 1.**

**¿Cuáles son las diferencias entre junta interna y externa?**

* + La junta interna está diseñada precisamente para reunir registros de varias tablas, entre ellas estaria las claves primarias y foráneas.
  + La junta externa puede verse como una reunión interna donde no es necesario que el registro hijo tenga informada la clave foránea para ser mostrado.

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

Se tiene las siguientes operaciones como lo son:

* + JOIN
  + NATURAL JOIN
  + CROSS JOIN

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

Las operaciones que se tienen son las siguientes:

* + LEFT JOIN
  + RIGHT JOIN
  + FULL JOIN

**PRACTICA**

**A.)**

**6 JOIN**

SELECT matchid, player FROM goal

WHERE teamiD = 'GER'

**2.**

SELECT id, stadium, team1, team2

FROM game

WHERE id = '1012'

**3.**

SELECT goal.player, goal.teamid, game.stadium, game.mdate

FROM goal INNER JOIN game

ON goal.matchid = game.id WHERE goal.teamid = 'GER'

**4.**

SELECT game.team1, game.team2, goal.player

FROM goal INNER JOIN game

ON goal.matchid = game.id WHERE goal.player LIKE 'Mario%'

**5.**

SELECT goal.player, goal.teamid, eteam.coach, goal.gtime

FROM goal INNER JOIN eteam

ON goal.teamid = eteam.id

WHERE goal.gtime <= 10

**6.**

SELECT game.mdate, eteam.teamname

FROM game INNER JOIN eteam

ON game.team1 = eteam.id

WHERE eteam.coach = 'Fernando Santos'

**7.**

SELECT goal.player

FROM goal INNER JOIN game

ON goal.matchid = game.id

WHERE game.stadium = 'National Stadium, Warsaw'

**8.**

SELECT DISTINCT player

FROM goal INNER JOIN game

ON id = matchid

WHERE teamid <> 'GER' and ( team1 = 'GER' or team2 = 'GER' )

**9.**

SELECT teamname, COUNT( PLAYER )

FROM eteam INNER JOIN goal ON id = teamid

GROUP BY teamname

**10.**

SELECT stadium, COUNT( player )

FROM goal INNER JOIN game ON matchid = id

GROUP BY stadium

**11.**

SELECT matchid, mdate, COUNT( player )

FROM game INNER JOIN goal ON id = matchid

WHERE team1 = 'POL' or team2 = 'POl'

GROUP BY matchid, mdate

**12.**

SELECT matchid, mdate, COUNT( player )

FROM goal INNER JOIN game ON id = matchid

WHERE teamid = 'GER'

GROUP BY matchid, mdate

**13.**

SELECT mdate, team1,

SUM( CASE WHEN teamid = team1 THEN 1 ELSE 0 END ) score1,

team2,

SUM( CASE WHEN teamid = team2 THEN 1 ELSE 0 END ) score2

FROM game LEFT JOIN goal ON id = matchid

GROUP BY mdate, team1, team2

ORDER BY mdate, matchid, team1, team2

[**7 More JOIN operations**](https://sqlzoo.net/wiki/More_JOIN_operations)

**1.**

SELECT id, title FROM movie

WHERE yr = 1962

**2.**

SELECT yr FROM movie

WHERE title = 'Citizen Kane'

**3.**

SELECT id,title,yr FROM movie

WHERE title LIKE '%Star Trek%'

**4.**

SELECT id FROM actor

WHERE name = 'Glenn Close'

**5.**

SELECT id FROM movie

WHERE title = 'Casablanca'

**6.**

SELECT name FROM casting JOIN actor ON id = actorid

WHERE movieid = 11768

**7.**

SELECT name FROM casting JOIN actor ON id = actorid

WHERE movieid IN (SELECT id FROM movie

WHERE title = 'Alien' )

8.

SELECT title FROM movie JOIN casting ON movieid = id

WHERE actorid IN

(SELECT id FROM actor

WHERE name = 'Harrison Ford')

9.

SELECT title FROM movie JOIN casting ON movieid = id

WHERE actorid IN

(SELECT id FROM actor

WHERE name = 'Harrison Ford') AND ord <> 1

10.

SELECT title, name

FROM movie AS M JOIN casting AS C JOIN actor AS A

ON M.id = C.movieid AND C.actorid = A.id

WHERE M.yr = 1962 AND C.ord = 1

11.

SELECT yr,COUNT( title ) Times 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

12.

SELECT M.title, A.name FROM movie M JOIN casting C JOIN actor A ON

M.id = C.movieid AND C.actorid = A.id

WHERE C.movieid IN

(SELECT movieid FROM casting

WHERE actorid IN (

SELECT id FROM actor

WHERE name='Julie Andrews')

) AND C.ord = 1

13.

SELECT name C FROM actor LEFT JOIN casting ON id = actorid

WHERE ord = 1

GROUP BY name

HAVING COUNT( ord ) >= 30

ORDER BY name

14.

SELECT title, COUNT( actorid ) T FROM

( SELECT id, title FROM movie

WHERE yr = 1978 ) M JOIN casting ON M.id = movieid

GROUP BY title

ORDER BY T DESC, title

15.

SELECT name FROM casting JOIN actor ON id = actorid

WHERE movieid IN

(SELECT movieid FROM (SELECT id FROM actor

WHERE name = 'Art Garfunkel') N JOIN casting ON N.id = actorid

) AND name <> 'Art Garfunkel'

**8 Using Null**

**1.**

SELECT name FROM teacher

WHERE dept IS NULL

**2.**

SELECT T.name, D.name FROM teacher T INNER JOIN dept D

ON T.dept = D.id

**3.**

SELECT T.name, D.name FROM teacher T LEFT JOIN dept D

ON T.dept = D.id

**4.**

SELECT T.name, D.name FROM teacher T RIGHT JOIN dept D

ON T.dept = D.id

**5.**

SELECT name, COALESCE( mobile, '07986 444 2266') FROM teacher

**6.**

SELECT T.name, COALESCE( D.name, 'None' ) FROM

teacher T LEFT JOIN dept D ON T.dept = D.id

**7.**

SELECT COUNT( name ),COUNT( mobile ) FROM teacher

**8.**

SELECT D.name, COUNT( T.NAME )

FROM dept D LEFT JOIN teacher T ON T.dept = D.id

GROUP BY D.name

**9.**

SELECT T.name,

CASE

WHEN T.dept IN (1,2) THEN 'Sci'

ELSE 'Art'

END

FROM teacher T LEFT JOIN dept D ON T.dept = D.id;

**10.**

SELECT T.name,

CASE

WHEN T.dept IN (1,2) THEN 'Sci'

WHEN T.dept = 3 THEN 'Art'

ELSE 'None'

END

FROM teacher T LEFT JOIN dept D ON T.dept = D.id;

**8+ Numeric Examples**

**1.**

SELECT A\_STRONGLY\_AGREE

FROM nss

WHERE question = 'Q01' AND institution ='Edinburgh Napier University'

AND subject ='(8) Computer Science'

2.

SELECT institution, subject FROM nss

WHERE score >= 100 AND question ='Q15'

3.

SELECT institution,score

FROM nss

WHERE question='Q15'

AND subject='(8) Computer Science'

AND score < 50

4.

SELECT subject, SUM ( response )

FROM nss

WHERE question='Q22'

AND subject IN ('(8) Computer Science',

'(H) Creative Arts and Design')

GROUP BY subject

5

SELECT subject, SUM(response \* A\_STRONGLY\_AGREE /100 ) FROM nss

WHERE subject IN

('(8) Computer Science','(H) Creative Arts and Design')

AND question ='Q22'

GROUP BY subject

.

6.

SELECT subject, ROUND(Num/Tot \* 100 ) Rta FROM

(SELECT subject, SUM ( response \* A\_STRONGLY\_AGREE /100 ) Num,

SUM( response ) Tot

FROM nss

WHERE question='Q22'

AND subject IN ('(8) Computer Science',

'(H) Creative Arts and Design')

GROUP BY subject

) N

7.

SELECT institution,ROUND( P/T \*100 )

FROM

(

SELECT institution, SUM( score \* response / 100 ) P, SUM( response ) T FROM nss

WHERE question = 'Q22' AND institution LIKE '%Manchester%'

GROUP BY institution ) N

8.

SELECT I, S, R

FROM ( SELECT institution I,SUM( sample ) S FROM nss

WHERE question='Q22'

AND (institution LIKE '%Manchester%')

GROUP BY institution

)N

JOIN (SELECT institution A, SUM( sample ) R FROM nss

WHERE question='Q22'

AND institution LIKE '%Manchester%' AND subject = '(8) Computer Science'

GROUP BY institution

) Y ON I = A

**9 Self join**

1.

SELECT COUNT( 1 ) FROM stops

2.

SELECT id FROM stops

WHERE name = 'Craiglockhart'

3.

SELECT id, name FROM route INNER JOIN stops

ON id= stop WHERE company = 'LRT' AND num = 4

4.

SELECT company, num, COUNT(\*)

FROM route WHERE stop=149 OR stop=53

GROUP BY company, num

HAVING COUNT( \* ) >=2

5.

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

6.

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'

7.

SELECT DISTINCT b.company, b.num FROM route a INNER JOIN route b

ON a.company = b.company AND a.num = b.num

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

8.

SELECT DISTINCT b.company, b.num FROM

route a JOIN route b ON a.company = b.company AND a.num = b.num

JOIN stops stopa ON stopa.id = a.stop

JOIN stops stopb ON stopb.id = b.stop

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

9.

SELECT stopb.name,a.company , b.num FROM

route a JOIN route b ON a.num = b.num AND a.company = b.company

JOIN stops stopa ON stopa.id = a.stop

JOIN stops stopb ON stopb.id = b.stop

WHERE a.company = 'LRT' AND stopa.name = 'Craiglockhart'

10.

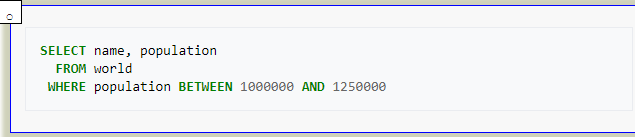
**FALTA HACERLO**

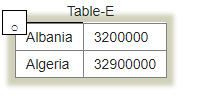
**B.)**

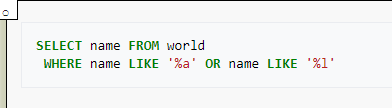
Tutorials: Learn SQL in stages

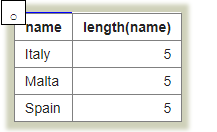
**10 Tutorial Quizzes**

**SELECT QUIZ**

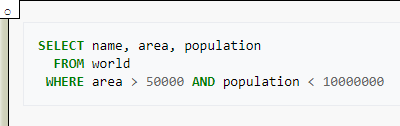
**1.** 

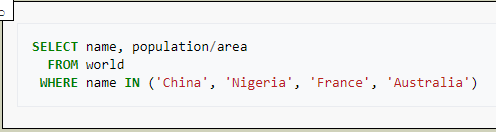
**2.** 

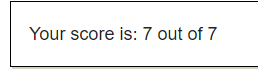
**3.** 

**4.** 

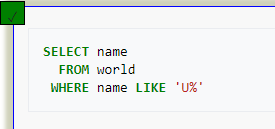
**5.** 

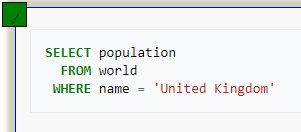
**6.** 

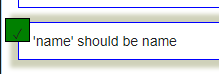
**7.** 

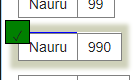
**Score** 

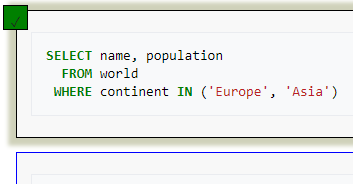
[**BBC QUIZ**](https://sqlzoo.net/wiki/BBC_QUIZ)

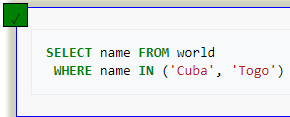
**1.** 

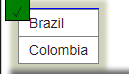
**2.** 

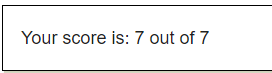
**3.** 

**4.** 

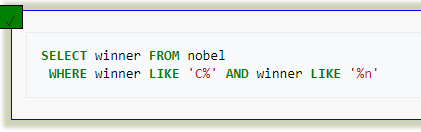
**5.** 

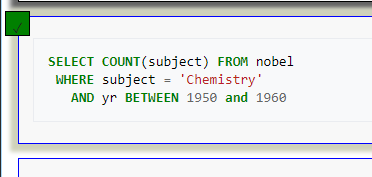
**6.** 

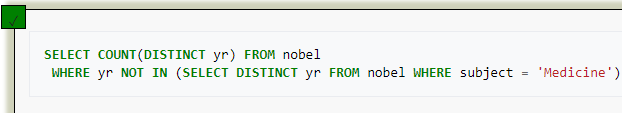
**7.**

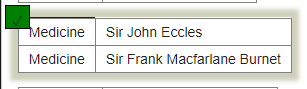
**Score** 

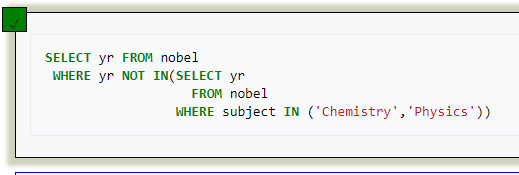
[**Nobel Quiz**](https://sqlzoo.net/wiki/Nobel_Quiz)

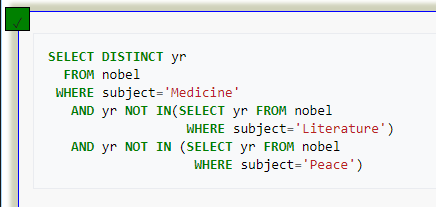
**1.** 

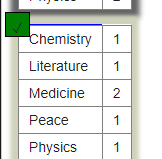
**2.** 

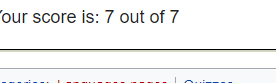
**3.** 

**4.** 

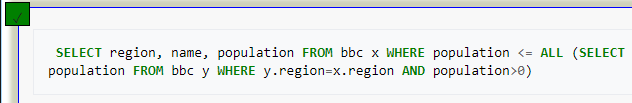
**5.** 

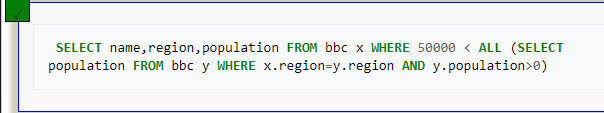
**6.** 

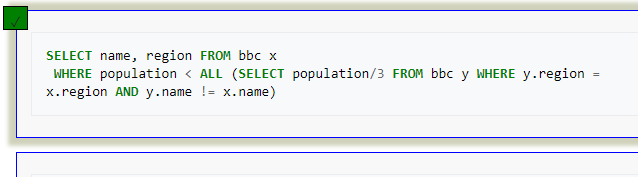
**7.** 

**Score**

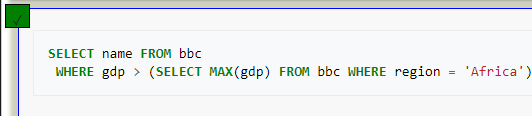
[**Nested SELECT Quiz**](https://sqlzoo.net/wiki/Nested_SELECT_Quiz)

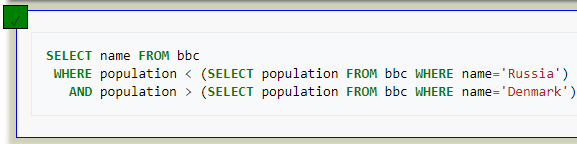
**1.** 

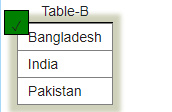
**2.** 

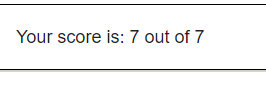
**3.** 

**4.** 

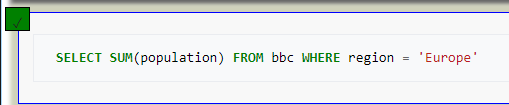
**5.** 

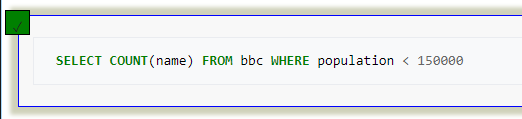
**6.** 

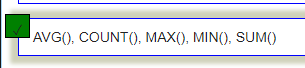
**7.** 

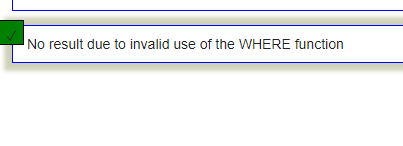
**Score**

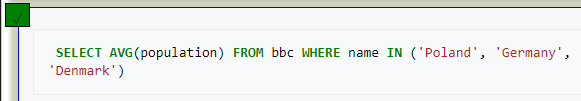
[SUM and COUNT Quiz](https://sqlzoo.net/wiki/SUM_and_COUNT_Quiz)

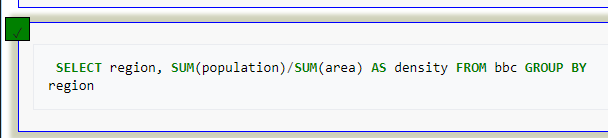
**1.** 

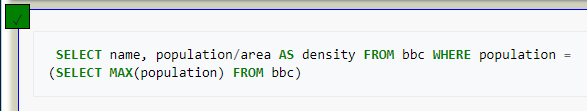
**2.** 

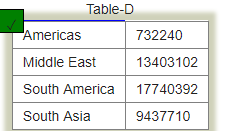
**3.** 

**4.** 

**5.** 

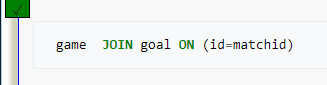
**6.** 

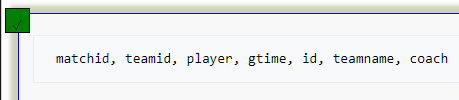
**7.** 

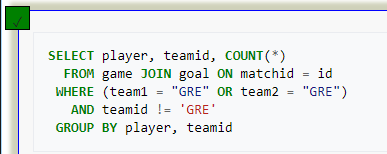
**8.** 

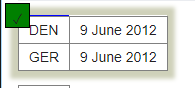
**Score**

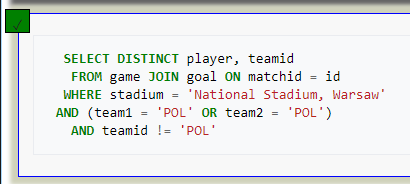
[**JOIN Quiz**](https://sqlzoo.net/wiki/JOIN_Quiz)

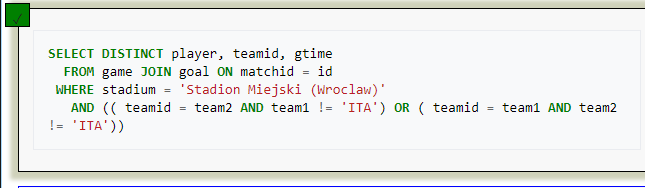
**1.** 

**2.** 

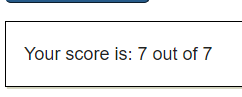
**3.** 

**4.** 

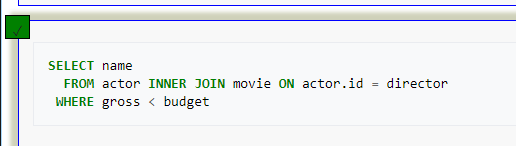
**5.** 

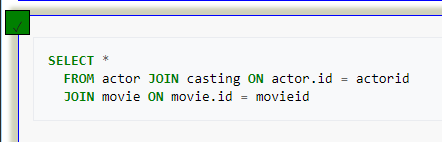
**6.** 

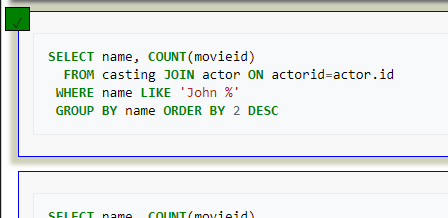
**7.** 

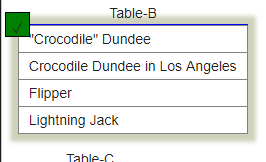
**Score**

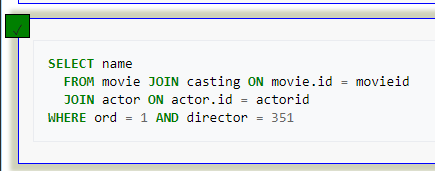
[**JOIN Quiz**](https://sqlzoo.net/wiki/JOIN_Quiz) **2**

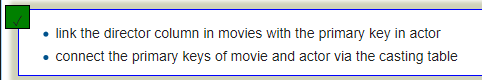
**1.** 

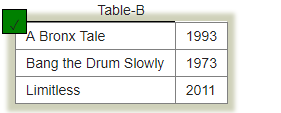
**2.** 

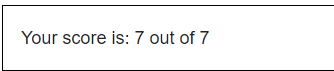
**3.** 

**4.** 

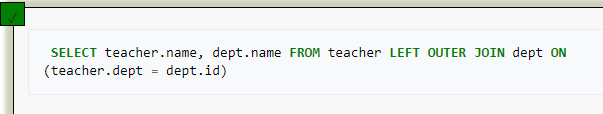
**5.** 

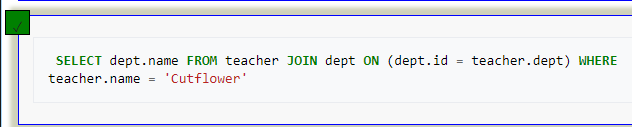
**6.** 

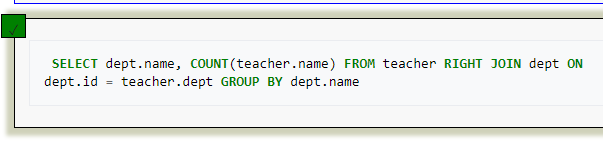
**7.** 

**Score**

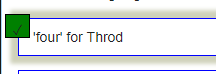
[**Using Null Quiz**](https://sqlzoo.net/wiki/Using_Null_Quiz)

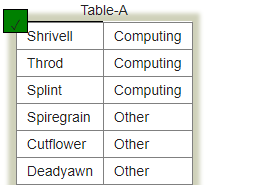
**1.** 

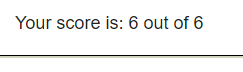
**2.** 

**3.** 

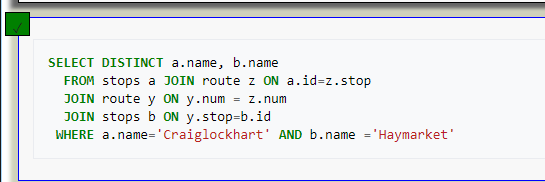
**4.** 

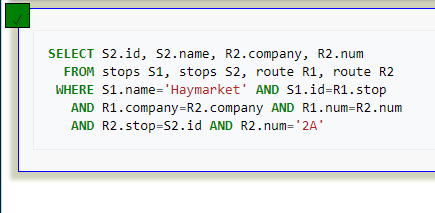
**5.** 

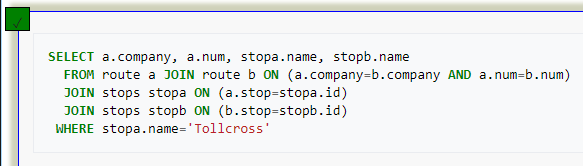
**6.** 

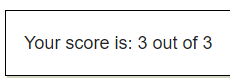
**Score**

[**Self-join Quiz**](https://sqlzoo.net/wiki/Self_join_Quiz)

**1.** 

**2.** 

**3.** 

**Score**

**CONSULTAS**

**C**. Propongan preguntas que cumplan los siguientes requerimientos. (15 puntos) Usen la base de datos musicians Escoja el motor que prefiera. Justifique la elección.

**Operadora de conjuntos**

* 5 consultas: una para cada operador de conjuntos

**UNION**

1. **Liste el nombre de las bandas y el nombre de los musicos**

SELECT band\_name FROM band

UNION

SELECT m\_name FROM musician

**UNION ALL**

1. **Liste el nombre de los musicos dos veces**

SELECT m\_name FROM musician

UNION ALL

SELECT m\_name FROM musician

**INTERSECT**

1. **Liste los nombre de musicos, que sean musicos y compositores**

SELECT m\_name

FROM ( SELECT m\_no FROM musician

INTERSECT

SELECT c\_no FROM composer

)New

INNER JOIN musician ON New.m\_no = musician.m\_no

**EXTRACT**

**4.liste el nombre de los donde su mes de nacimiento sea octubre**

SELECT m\_name FROM musician

WHERE EXTRACT(MONTH FROM born ) = '10'

**IN**

**5.Muestre los musicos que nacieron en USA o Autralia o Inglaterra .**

SELECT m\_name

FROM musician INNER JOIN place ON place\_no = born\_in

WHERE place\_country IN ('England','Austria','USA')

**Junta Interna y Junta Externa**

* **4 consultas: dos por cada junta Interna y dos para junta externa**

**Junta Interna**

**JOIN**

**1.Muestre los nombres de los compositores que viven en USA**

SELECT m\_name

FROM musician JOIN composer JOIN place ON m\_no = comp\_no AND place\_no = living\_in

WHERE place\_country = 'USA'

**NATURAL JOIN**

**2.Muestre los nombres de los intrepetes que viven en USA**

SELECT \* FROM

(SELECT m\_no T, m\_name FROM musician ) A NATURAL JOIN ( SELECT DISTINCT perf\_is T FROM performer ) N

**Junta Externa**

**LEFT JOIN**

**1.Muestre el nombre y la composicion de todos los musicos.**

SELECT m\_name, c\_title

FROM musician LEFT JOIN has\_composed ON m\_no = cmpr\_no

LEFT JOIN composition ON cmpn\_no = c\_no

**RIGHT JOIN**

**2.Muestre el nombre e instrumento de que toca cada musico**

SELECT m\_name, instrument

FROM performer RIGHT JOIN musician ON m\_no = perf\_is

**Operadores Desconocido**

* **2 consultas: una para cada operador desconocido.**

**ISNULL**

**1.Muestre los musicos y su fecha de fallecimiento si no existe, colocar ‘None’**

SELECT m\_name,

CASE

WHEN ISNULL ( died ) = 0

THEN died

ELSE 'None'

END

FROM musician

**COALESCE**

**2.Muestre los nombres de las bandas y su b\_date si este no existe, muestre el tipo de banda.**

SELECT band\_name, COALESCE (b\_date , band\_type) FROM band

**Operadores logicos**

* 3 consultas: una para cada uno de los tipos de operadores logicos.

**EXISTS**

**1. Muestre los interpretes que nacieron en USA o Autralia o Inglaterra**

SELECT m\_name FROM musician

WHERE EXISTS ( SELECT place\_no FROM place

WHERE place\_country IN ('England','Austria','USA') )

**ANY**

**2 .Muestre los musicos que nacieron el mismo dia.**

SELECT m\_name FROM musician x

WHERE x.born = ANY (SELECT y.born FROM musician y

WHERE x.m\_name <> y.m\_name )

**ALL**

**Muestre el musico cuyo nombre se nombre sea alfabeticamente mayor a todos.**

SELECT x.m\_name FROM musician x

WHERE x.m\_name >= ALL (SELECT y.m\_name FROM musician y)

**3.**

**Operador**

* 1 consulta: para operador CASE

**CASE**

Cuente cuantos musicos son interpretes

SELECT

SUM( CASE WHEN m\_no IN (SELECT DISTINCT perf\_is FROM performer ) THEN 1

ELSE 0

END

) FROM musician