**INVESTIGACIÓN**

2. Sirve para indicar que no existe un valor dentro de una base de datos.
3. Las diferentes comparaciones de NULL pueden resultar ciertas o falsas y no se puede comparar con los operadores de (>,<,<>,=) y en su lugar se usa IS NULL Y IS NOT NULL.
4. 1. **Junta interna:** Con esta operación cada registro en la tabla A es combinado con los correspondientes de la tabla B que satisfagan las condiciones que se especifiquen en el predicado del *JOIN*. Cualquier registro de la tabla A o de la tabla B que no tenga uno correspondiente en la otra tabla es excluido, y solo aparecerán los que tengan correspondencia en la otra tabla.

**Junta externa:** Mediante esta operación no se requiere que un registro en una tabla tenga un registro relacionado en la otra tabla. El registro es mantenido en la tabla combinada, aunque no exista el correspondiente en la otra tabla.

2.

* + Join
  + Natural Join
  + Cross Join

3.

* + Left Join
  + Right Join
  + Full Join

**PRÁCTICA**



6 JOIN:

SELECT matchid,

player

FROM goal

WHERE teamid = 'GER'



SELECT id,

stadium,

team1,

team2

FROM game

WHERE id = 1012



SELECT player,

teamid,

stadium,

mdate

FROM game JOIN goal ON (id=matchid)

WHERE teamid = 'GER'



SELECT team1,

team2,

player

FROM game JOIN goal ON (id = matchid)

WHERE player LIKE 'Mario%'



SELECT player,

teamid,

coach,

gtime

FROM goal JOIN eteam ON (teamid = id)

WHERE gtime<=10



SELECT mdate,

teamname

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

WHERE coach = 'Fernando Santos'



SELECT player

FROM game JOIN goal on (id = matchid)

WHERE stadium = 'National Stadium, Warsaw'



SELECT DISTINCT player

FROM game JOIN goal ON (matchid = id)

WHERE (team1='GER' AND team2=teamid)

OR (team2='GER' AND team1=teamid)



SELECT teamname,

COUNT(teamid)

FROM eteam JOIN goal ON id=teamid

GROUP BY teamname



SELECT stadium,

COUNT(stadium)

FROM game JOIN goal ON id=matchid

GROUP BY stadium



SELECT matchid,

mdate,

COUNT(teamid)

FROM game JOIN goal ON matchid = id

WHERE team1 = 'POL'

OR team2 = 'POL'

GROUP BY matchid, mdate

SELECT matchid,

mdate,

COUNT(teamid)

FROM game JOIN goal ON matchid = id

WHERE (teamid = 'GER')

GROUP BY matchid, mdate



SELECT mdate,

team1,

SUM(score1) AS score1,

team2,

SUM(score2) AS score2

FROM (

SELECT mdate,

team1,

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

team2,

CASE WHEN teamid=team2 THEN 1 ELSE 0 END score2

FROM game LEFT JOIN goal ON matchid = id) AS a

GROUP BY mdate,

team1,

team2

ORDER BY mdate,

team1,

team2

7 More JOIN operations:



SELECT id,

title

FROM movie

WHERE yr=1962



SELECT yr

FROM movie

WHERE title = 'Citizen Kane'



SELECT id,

title,

yr

FROM movie

WHERE title LIKE '%Star Trek%'

ORDER BY yr



SELECT id

FROM actor

WHERE name = 'Glenn Close'



SELECT id

FROM movie

WHERE title = 'Casablanca'



SELECT name

FROM movie JOIN casting ON movie.id = movieid

JOIN actor ON actor.id = actorid

WHERE title = 'Casablanca'



SELECT name

FROM movie JOIN casting ON movie.id = movieid

JOIN actor ON actor.id = actorid

WHERE title = 'Alien'



SELECT title

FROM movie JOIN casting ON movie.id = movieid

JOIN actor ON actor.id = actorid

WHERE name = 'Harrison Ford'



SELECT title

FROM movie JOIN casting ON movie.id = movieid

JOIN actor ON actor.id = actorid

WHERE name = 'Harrison Ford'

AND ord != 1



SELECT title,

name

FROM movie JOIN casting ON movie.id = movieid

JOIN actor ON actor.id = actorid

WHERE ord = 1

AND yr = 1962



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



SELECT title,

name

FROM movie JOIN casting ON movie.id = movieid

JOIN actor ON actorid = actor.id

WHERE movieid IN (

SELECT movieid

FROM casting

WHERE actorid IN (

SELECT id

FROM actor

WHERE name='Julie Andrews'))

AND ord = 1



SELECT name

FROM (

SELECT name,

COUNT(name) AS size

FROM casting JOIN actor ON actorid = id

WHERE ord = 1

GROUP BY name

HAVING size >= 30) AS base



SELECT title,

COUNT(actorid) AS nActors

FROM movie JOIN casting ON id = movieid

WHERE yr = 1978

GROUP BY title

ORDER BY nActors DESC,

Title



SELECT name

FROM casting JOIN actor ON actorid = id

WHERE movieid IN(

SELECT movieid

FROM casting JOIN actor ON actorid = id

WHERE name = 'Art Garfunkel')

AND name != 'Art Garfunkel'

8 Using Null:



SELECT name

FROM teacher

WHERE dept IS NULL



SELECT teacher.name,

dept.name

FROM teacher INNER JOIN dept ON (teacher.dept=dept.id)



SELECT teacher.name,

dept.name

FROM teacher LEFT JOIN dept ON (teacher.dept=dept.id)



SELECT teacher.name,

dept.name

FROM teacher RIGHT JOIN dept ON (teacher.dept=dept.id)



SELECT name,

COALESCE (mobile, '07986 444 2266')

FROM teacher



SELECT t.name,

COALESCE(d.name, 'None')

FROM teacher AS t LEFT JOIN dept AS d ON (t.dept = d.id)



SELECT COUNT(name),

COUNT(mobile)

FROM teacher



SELECT d.name,

COUNT(t.name)

FROM teacher AS t RIGHT JOIN dept AS d ON (t.dept = d.id)

GROUP BY d.name



SELECT name,

CASE WHEN dept = 1 OR dept = 2 THEN 'Sci'

ELSE 'Art'

END

FROM teacher



SELECT name,

CASE WHEN dept = 1 OR dept = 2 THEN 'Sci'

WHEN dept = 3 THEN 'Art'

ELSE 'None'

END

FROM teacher

8+ Numeric Examples:



SELECT A\_STRONGLY\_AGREE

FROM nss

WHERE question='Q01'

AND institution='Edinburgh Napier University'

AND subject='(8) Computer Science'



SELECT institution,

subject

FROM nss

WHERE question='Q15'

AND score >= 100



SELECT institution,

score

FROM nss

WHERE question='Q15'

AND subject='(8) Computer Science'

AND score < 50



SELECT subject,

SUM(response)

FROM nss

WHERE question='Q22'

AND (subject='(8) Computer Science'

OR subject = '(H) Creative Arts and Design')

GROUP BY subject



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



SELECT subject,

ROUND(SUM(response\*A\_STRONGLY\_AGREE/100)/SUM(response) \* 100) AS b

FROM nss

WHERE question='Q22'

AND (subject='(8) Computer Science'

OR subject = '(H) Creative Arts and Design')

GROUP BY subject



SELECT institution,

ROUND(AVG(score))

FROM nss

WHERE question='Q22'

AND (institution LIKE '%Manchester%')

GROUP BY institution

ORDER BY institution



SELECT institution,

SUM(sample),

SUM(response)

FROM nss

WHERE question='Q01'

AND (institution LIKE '%Manchester%')

GROUP BY institution

9 Self join:



SELECT COUNT(id)

FROM stops



SELECT id

FROM stops

WHERE name = 'Craiglockhart'



select id,

name

FROM stops JOIN route ON (id = stop)

WHERE num = '4'

AND company = 'LRT'



SELECT company,

num,

COUNT(\*) AS a

FROM route

WHERE stop=149

OR stop=53

GROUP BY company,

num

HAVING a = 2



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



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'



SELECT DISTINCT a.company,

a.num

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

AND a.num=b.num)

WHERE a.stop=115

AND b.stop=137



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'



SELECT s2.name,

r1.company,

r1.num

FROM route r1 JOIN route r2 ON (r1.num=r2.num

AND r1.company=r2.company)

JOIN stops s1 ON r1.stop = s1.id

JOIN stops s2 ON r2.stop = s2.id

WHERE s1.name = 'Craiglockhart'



SELECT r1.num,

r1.company,

s2.name,

r4.num,

r4.company

FROM (route r1 JOIN route r2 ON (r1.num=r2.num

AND r1.company=r2.company)

JOIN stops s1 ON (r1.stop = s1.id)),

(route r3 JOIN route r4 ON (r3.num=r4.num

AND r3.company=r4.company)

JOIN stops s2 ON (r3.stop = s2.id)

JOIN stops s3 ON (r4.stop = s3.id))

WHERE s1.name = 'Craiglockhart'

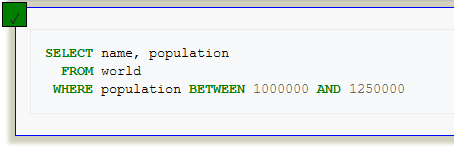
AND s3.name = 'Lochend'

AND r2.stop = r3.stop

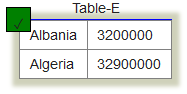


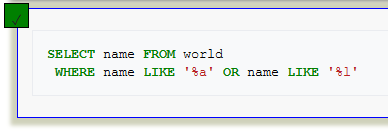
SELECT Quiz:



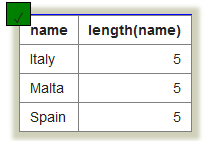








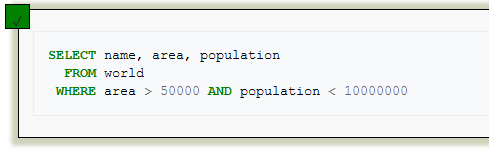


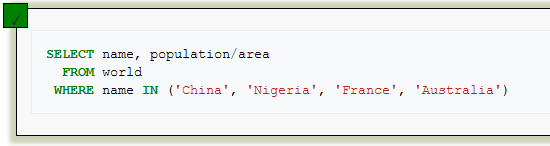






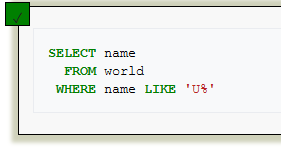




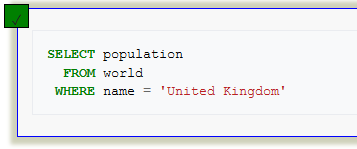


BBC Quiz:









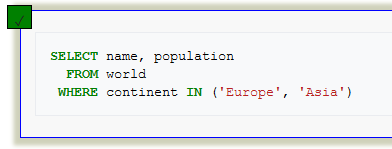




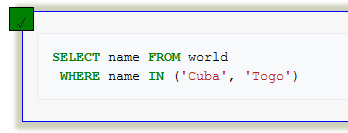










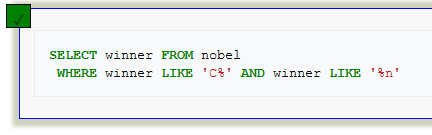




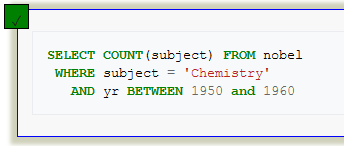


Nobel Quiz:

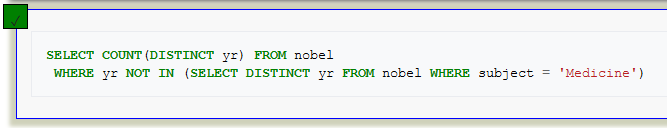




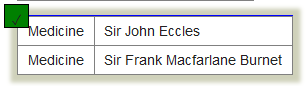




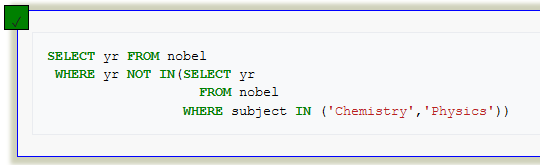




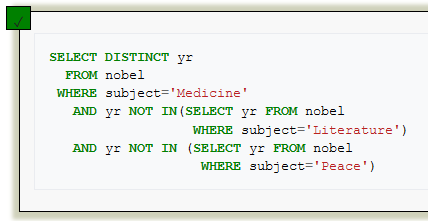




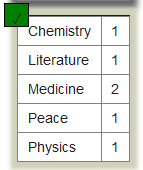






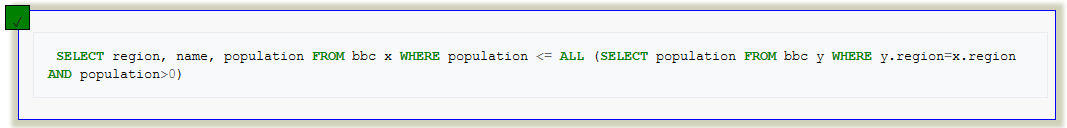




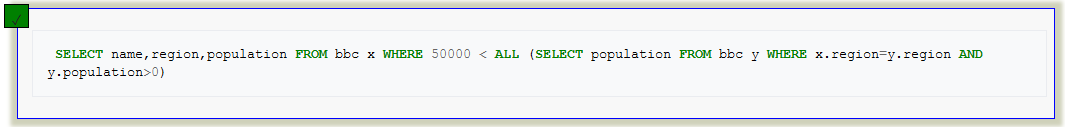


Nested SELECT Quiz:

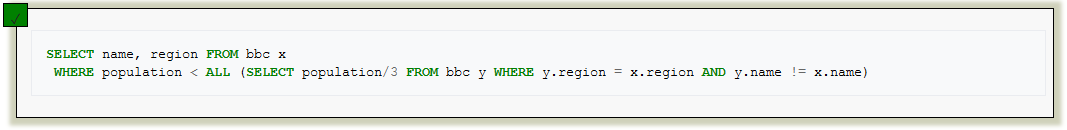








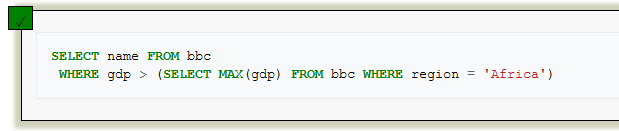




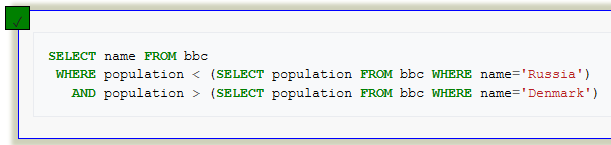




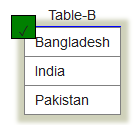






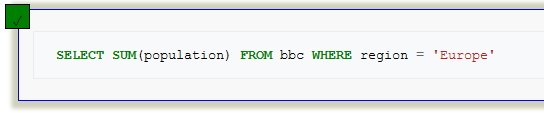




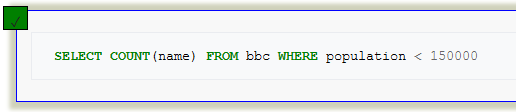


SUM and COUNT Quiz:









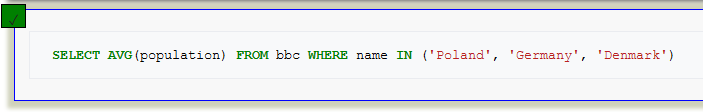




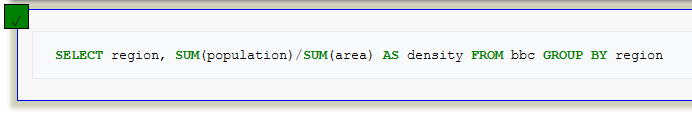




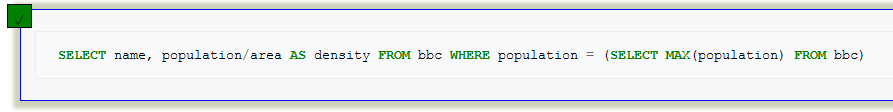




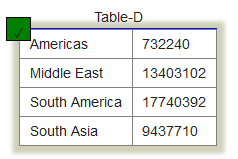






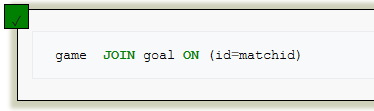




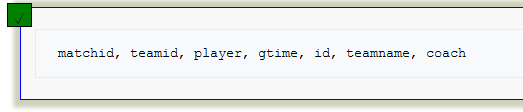


JOIN Quiz:

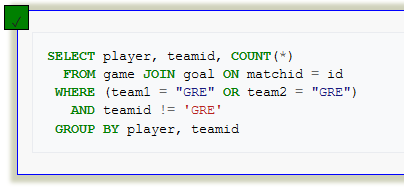








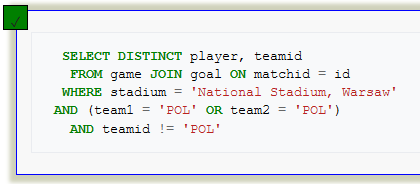




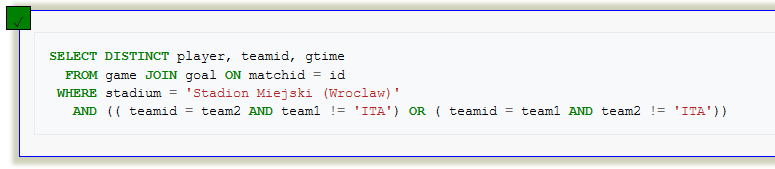




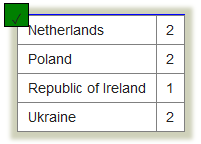






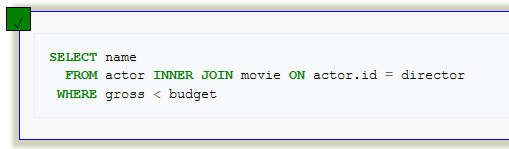




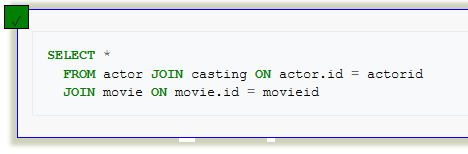


JOIN Quiz 2:

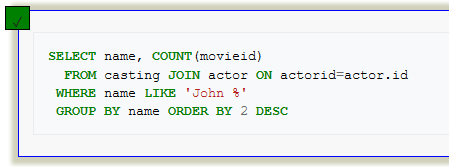




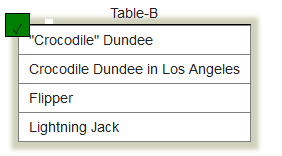




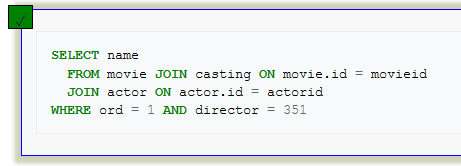




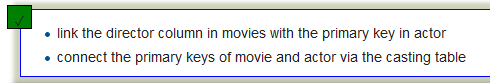




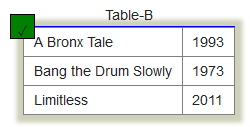






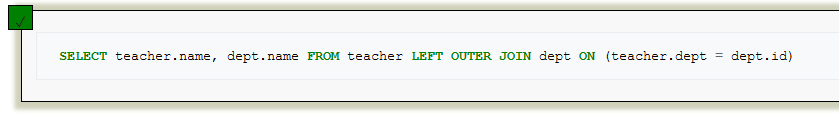




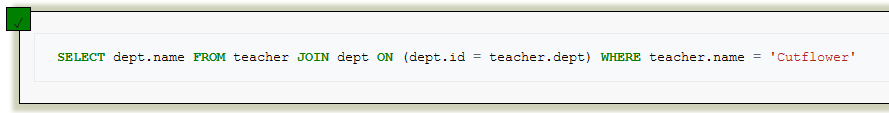


Using Null Quiz:

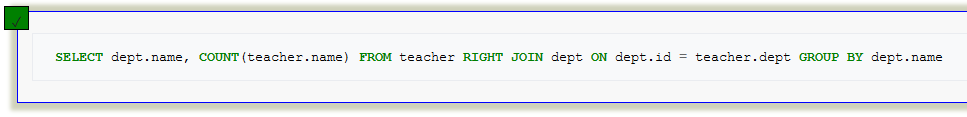












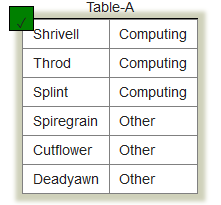






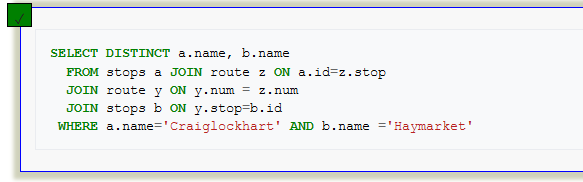




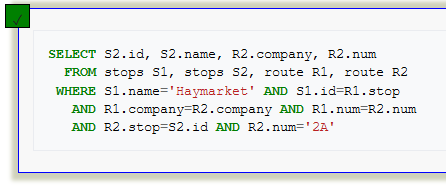


Self JOIN Quiz:

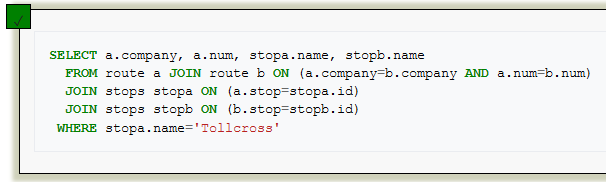














* 5 consultas: una para cada operador de conjuntos

1. Diga el número de músicos, numero de compositores y número de bandas.

SELECT COUNT(m\_no)

FROM musician

UNION

SELECT COUNT(comp\_no)

FROM composer

UNION

SELECT COUNT(band\_no)

FROM band

1. Escriba el nombre de los músicos que son compositores.

SELECT m\_name

FROM musician

INTERSECT

SELECT m\_name

FROM musician JOIN composer ON (m\_no = comp\_is)

1. Escriba el nombre de los músicos que están en una banda.

SELECT m\_name

FROM musician

WHERE m\_no in (SELECT m\_no

FROM musician, performer, plays\_in, band

WHERE band\_no = band\_id

AND player = perf\_no

AND perf\_is = m\_no)

* 4 consultas: dos para junta interna y dos para junta externa

1. Escriba el nombre de los músicos junto a su lugar de nacimiento.

SELECT m\_name,

place\_town

FROM musician JOIN place ON (born\_in = place\_no)

1. Escriba el nombre de todos los intérpretes y su fecha de muerte.

SELECT m\_name, died

FROM performer LEFT JOIN musician ON (m\_no = perf\_is)

1. Escriba el nombre de los compositores y su fecha de muerte.

SELECT m\_name, died

FROM musician RIGHT JOIN composer ON (m\_no = comp\_is)

* 2 consultas: una para cada operador de desconocido

1. Escriba el nombre de los músicos que fallecieron

SELECT m\_name

FROM musician

WHERE died IS NOT NULL

1. Escriba todos los músicos, si falleció, escriba la fecha, si no, escriba ‘VIVO’

SELECT m\_name, COALESCE (died, 'VIVO')

FROM musician

* 1 consulta: para el operador CASE

1. Escriba las bandas, y su b\_date, si tiene, escriba ‘Tiene’, si no tiene, escriba ‘No tiene’

SELECT band\_name,

CASE WHEN b\_date IS NULL THEN 'No tiene'

WHEN b\_date IS NOT NULL THEN 'Tiene'

END

FROM band

**BIBLIOGRAFÍA**

* <https://es.wikipedia.org/wiki/Null_(SQL)>
* <https://es.wikipedia.org/wiki/Join#Combinaci%C3%B3n_externa_(OUTER_JOIN)>