# **Investigación**

A. NULL

¿Qué significa? ¿Resultado de operarlo con los diferentes tipos de operadores: aritméticos, lógicos y de comparación?

Se utiliza un valor NULL en una base de datos relacional cuando el valor de una columna es desconocido o falta. Un valor NULL no es una cadena vacía (para tipos de datos de caracteres o de fecha y hora) ni un valor cero (para tipos de datos numéricos).

Al permitir valores NULL en definiciones de columna, se introduce la lógica de tres valores en la aplicación. Una comparación puede evaluarse en una de estas tres condiciones:

* True, False,Desconocido

Cualquier operación lógica que implique un NULL da como resultado un valor desconocido, excepto TRUE OR NULL y FALSE AND NULL.

Dado que se considera que NULL es desconocido, dos valores NULL comparados entre sí no se consideran iguales, no devuelven TRUE ni FALSE, sino UNKNOWN. Usa IS NULL y IS NOT NULL para verificar la presencia de NULL. En las expresiones que usan operadores aritméticos, si alguno de los operandos es NULL, el resultado es NULL también.

B. JUNTA

1. ¿Cuáles son las diferencias entre junta interna y externa?

Uniones interiores: Una unión interna se utiliza para seleccionar los registros que tienen correspondencia entre dos archivos unidos. Por ejemplo, para listar todos los clientes que han realizado pedidos, el fichero "Cliente" debe ser relacionado al fichero "Pedidos" a través de una unión interna. La consulta seleccionará a los clientes relacionado con al menos un número de pedido. Los clientes que no hayan hecho ningún pedido no aparecerán en la lista.

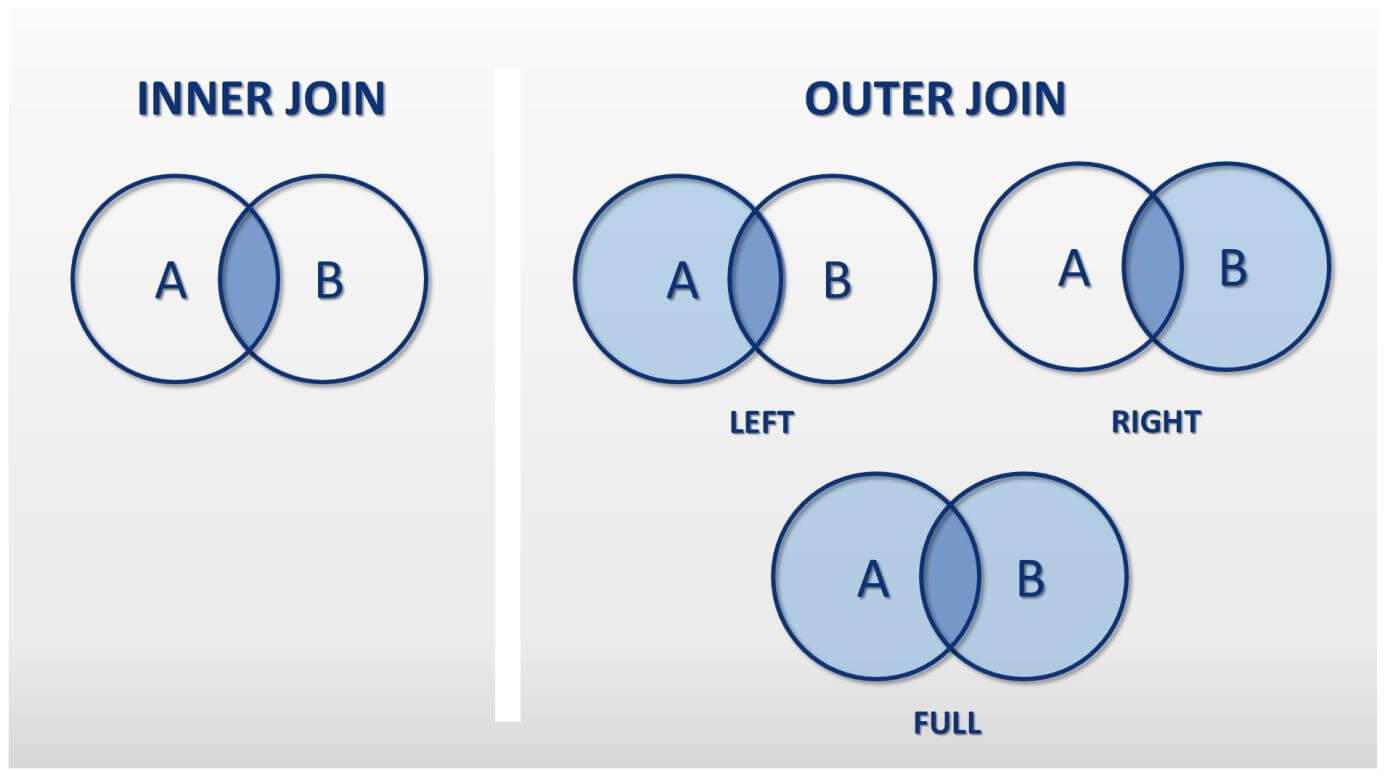
Uniones externas:

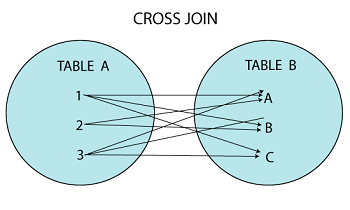
Una junta exterior se utiliza para seleccionar ambos:

* los registros que tienen correspondencia entre los dos archivos de relacionado.
* los registros que no tienen correspondencia en el primer fichero, en el segundo fichero o en todos los ficheros de relacionado.

Por ejemplo, para saber la cantidad total gastada por cada cliente: Al utilizar una unión interna, sólo se seleccionarán los clientes que ya han realizado un pedido (cuyo importe de gasto difiere de 0). Al utilizar una junta externa, se seleccionarán todos los clientes, incluso los que nunca han hecho un pedido.

¿Qué opciones se tienen para la junta interna ? ¿ Qué opciones se tienen para la junta externa?

Representación de INNER JOIN y OUTER JOIN con diagramas de conjuntos: las áreas coloreadas representan el conjunto de resultados del JOIN a partir de los conjuntos A y B. El INNER JOIN muestra como resultado la intersección de A y B (azul oscuro). El conjunto de resultados de un OUTER JOIN incluye la intersección entre A y B (azul oscuro) y el resto (azul claro) de A (LEFT OUTER JOIN), B (RIGHT OUTER JOIN) o A y B (FULL OUTER JOIN).

Aunque tambien hablamos de un CROSS JOIN que devuelve el producto cartesiano de las dos tablas, es decir, combina cada fila de la primera tabla con cada fila de la segunda tabla.

Bibliografía

1. **Microsoft Learn. (n.d.).** *Handle null values*. Retrieved September 11, 2024, from <https://learn.microsoft.com/es-es/sql/connect/ado-net/sql/handle-null-values?view=sql-server-ver16>
2. **Aprender SQL. (2022, September 15).** *Cómo utilizar los operadores de comparación con NULL en SQL*. Retrieved September 11, 2024, from <https://learnsql.es/blog/como-utilizar-los-operadores-de-comparacion-con-null-en-sql/>
3. **WinDev. (n.d.).** *Help*. Retrieved September 11, 2024, from <https://help.windev.com/es-ES/?2034007>
4. **IONOS. (2022, March 4).** *SQL OUTER JOIN: ¿qué es y cómo usarlo?*. Retrieved September 11, 2024, from <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/sql-outer-join/>

# **Practica**

A.

JOIN

1.

SELECT matchid, player

FROM goal

WHERE teamid ='GER'

2.

SELECT id,stadium,team1,team2

FROM game

WHERE id = 1012

3.

SELECT player,teamid, stadium , mdate

FROM game JOIN goal ON (game.id=goal.matchid)

WHERE teamid = 'GER'

4.

SELECT team1, team2, player

FROM game

JOIN goal ON (game.id=goal.matchid)

WHERE player LIKE 'Mario%'

5.

SELECT player, teamid, coach, gtime

FROM goal

JOIN eteam ON (goal.teamid=eteam.id)

WHERE gtime<=10

6.

SELECT mdate, teamname

FROM game

JOIN eteam ON (team1=eteam.id)

WHERE coach = 'Fernando Santos'

7.

SELECT player

FROM goal

JOIN game ON (game.id = goal.matchid)

WHERE stadium = 'National Stadium, Warsaw'

8.

SELECT DISTINCT player

FROM game

JOIN goal ON (goal.matchid = game.id)

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

9.

SELECT teamname,COUNT(teamname)

FROM eteam

JOIN goal ON (eteam.id=teamid)

GROUP BY teamname

10.

SELECT stadium , COUNT(stadium)

FROM goal

JOIN game ON (goal.matchid = game.id)

GROUP BY stadium

11.

SELECT matchid,mdate, COUNT(matchid)

FROM game

JOIN goal ON matchid = id

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

GROUP BY matchid

12.

SELECT matchid, mdate, COUNT(matchid)

FROM game

JOIN goal ON goal.matchid = game.id

WHERE teamid = 'GER'

GROUP BY matchid

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 (goal.matchid = game.id)

GROUP BY mdate, team1, team2

ORDER BY mdate, matchid, team1, team2

QUIZ JOIN



More JOIN

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%'

ORDER BY yr

4.

SELECT id

FROM actor

WHERE name = 'Glenn Close'

5.

SELECT id

FROM movie

WHERE title = 'Casablanca'

6.

SELECT name

FROM casting c

JOIN actor a ON a.id = c.actorid

WHERE movieid = 11768

7.

SELECT name

FROM casting c

JOIN actor a ON a.id = c.actorid

WHERE movieid = (SELECT id

FROM movie

WHERE title = 'Alien')

8.

SELECT title

FROM movie m

JOIN casting c ON c.movieid = m.id

JOIN actor a ON a.id = c.actorid

WHERE a.name = 'Harrison Ford'

9.

SELECT title

FROM movie m

JOIN casting c ON c.movieid = m.id

JOIN actor a ON a.id = c.actorid

WHERE a.name = 'Harrison Ford' and c.ord != 1

10.

SELECT title, name

FROM movie m

JOIN casting c ON c.movieid = m.id

JOIN actor a ON a.id = c.actorid

WHERE m.yr = 1962 AND c.ord = 1

11.

SELECT yr, COUNT(title) movie\_count

FROM movie m

JOIN casting c ON m.id = c.movieid

JOIN actor a ON c.actorid = a.id

WHERE name = 'Rock Hudson'

GROUP BY yr

HAVING COUNT(title) > 2

12.

SELECT title, name

FROM movie m

JOIN casting c ON m.id = c.movieid

JOIN actor a ON c.actorid = a.id

WHERE (ord = 1) AND (movieid IN (SELECT movieid

FROM casting c

JOIN actor a ON a.id = c.actorid

WHERE name = 'Julie Andrews'))

13.

SELECT name

FROM casting c

JOIN actor a ON a.id = c.actorid

WHERE c.ord = 1

GROUP BY a.name

HAVING COUNT(name) >= 15

ORDER by a.name

14.

SELECT title, COUNT(actorid)

FROM movie m

JOIN casting c ON m.id = c.movieid

WHERE yr = '1978'

GROUP BY title

ORDER BY COUNT(actorid) DESC, title

15.

SELECT a.name

FROM casting c

JOIN actor a ON c.actorid = a.id

WHERE (a.name != 'Art Garfunkel') AND (c.movieid IN (SELECT c.movieid

FROM casting c

JOIN actor a ON c.actorid = a.id

WHERE a.name = 'Art Garfunkel'))

QUIZ More JOIN



Using NULL

1.

SELECT name

FROM teacher

WHERE dept IS NULL

2.

SELECT tc.name, dept.name

FROM teacher tc

JOIN dept ON tc.dept = dept.id

3.

SELECT tc.name, dept.name

FROM teacher tc

LEFT JOIN dept ON tc.dept = dept.id

4.

SELECT tc.name, dept.name

FROM teacher tc

RIGHT JOIN dept ON tc.dept = dept.id

5.

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

FROM teacher

6.

SELECT tc.name, COALESCE(dept.NAME,'None')

FROM teacher tc

LEFT JOIN dept ON tc.dept = dept.id

7.

SELECT COUNT(name), COUNT(mobile)

FROM teacher

8.

SELECT dept.name, COUNT(tc.name)

FROM teacher tc

RIGHT JOIN dept ON tc.dept = dept.id

GROUP BY dept.name

9.

SELECT tc.name, CASE WHEN tc.dept = 1 OR tc.dept = 2

THEN 'Sci'

ELSE 'Art'

END

FROM teacher tc

LEFT JOIN dept ON tc.dept = dept.id

10.

SELECT tc.name, CASE WHEN tc.dept = 1 OR tc.dept = 2

THEN 'Sci'

WHEN tc.dept = 3

THEN 'Art'

ELSE 'None'

END

FROM teacher tc

LEFT JOIN dept ON tc.dept = dept.id

QUIZ Using NULL



Self JOIN

1.

SELECT COUNT(DISTINCT stop)

FROM route

2.

SELECT id

FROM stops

WHERE name = 'Craiglockhart'

3.

SELECT id, name

FROM stops

JOIN route ON stops.id = route.stop

WHERE num = '4' and company = 'LRT'

4.

SELECT company, num, COUNT(num)

FROM route

WHERE stop = 149 OR stop = 53

GROUP BY company, num

HAVING COUNT(num) > 1

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 IN (SELECT id

FROM stops

WHERE name = 'London Road' ))

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 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)

8.

SELECT a.company, a.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, 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'

10.

SELECT r1.num, r1.company, s1.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 r2.stop = s1.id

JOIN route r3 ON s1.id = r3.stop

JOIN route r4 ON (r3.num = r4.num) AND (r3.company = r4.company)

WHERE (r1.stop = (SELECT id

FROM stops

WHERE name = 'Craiglockhart')) AND

(r4.stop = (SELECT id

FROM stops

WHERE name = 'Lochend'))

ORDER BY r1.num, s1.name, r4.num

QUIZ Self JOIN



B.

Operadores de conjuntos:

UNION:

Natural:

Mostrar el nombre de los músicos que sean interpretes junto al nombre de los musicos que hayan dirigido una interpretación poniendo junto al nombre “performer” o “director” dependiendo de lo que este registrado en la base de datos

SQL:

SELECT 'performer' as 'Type', m\_name as 'Name'

FROM musician m

JOIN performer pr ON pr.perf\_is = m.m\_no

UNION

SELECT 'director', m\_name

FROM musician m

JOIN performance prc ON prc.conducted\_by = m.m\_no

UNION ALL:

Natural:

Mostrar el nombre de los cantantes, ciudad y país de origen junto al nombre de las bandas, ciudad y país de origen esto junto a un indicador de “Musician” o “Band” dependiendo la fila que sea.

SQL:

SELECT 'Musician' as 'Type', m\_name as 'Name', place\_town as 'from town', place\_country as 'from country'

FROM musician m

JOIN place pl ON pl.place\_no = m.born\_in

UNION ALL

SELECT 'Band', band\_name, place\_town, place\_country

FROM band b

JOIN place pl ON pl.place\_no = b.band\_home

INTERSECT:

Natural:

Mostrar el nombre de los musicos que han sido intérpretes y directores de interpretaciones

SQL:

SELECT m\_name as 'Performer and director'

FROM musician m

JOIN performer pr ON pr.perf\_is = m.m\_no

INTERSECT

SELECT m\_name

FROM musician m

JOIN performance prc ON prc.conducted\_by = m.m\_no

EXTRACT:

Natural:

Mostrar el nombre de los músicos y sus años de nacimiento

SQL:

SELECT m\_name as 'musician name', EXTRACT(YEAR FROM born) as 'birth year'

FROM musician

IN:

Natural:

Mostrar el titulo y fecha de composición de la composición más antigua de la que se tenga registro

SQL:

SELECT c\_title as 'composition title', DATE\_FORMAT(min(comp\_date), '%d/%m/%y') as 'composition date'

FROM composition

WHERE comp\_date IN (SELECT comp\_date

FROM composition

WHERE comp\_date IS NOT NULL)

Operadores de Junta:

JOIN:

Natural:

Mostrar el nombre de los músicos que sean organizadores y compositores y que vivan en Inglaterra o escocia

SQL:

SELECT DISTINCT m\_name

FROM musician m

JOIN performance p ON m.m\_no = p.conducted\_by

JOIN composer c ON m.m\_no = c.comp\_is

JOIN place pl ON m.living\_in = pl.place\_no

WHERE (place\_country = 'England') OR (place\_country = 'Scotland')

NATURAL JOIN:

Natural:

Mostrar el nombre del musico junto a su ciudad y país de nacimiento y residencia

SQL:

SELECT m\_name as 'musician name', town\_of\_birth, country\_of\_birth, town\_of\_residence, country\_of\_residence

FROM musician

NATURAL JOIN (SELECT place\_no as 'born\_in', place\_town as 'town\_of\_birth', place\_country as 'country\_of\_birth'

FROM place)x

NATURAL JOIN (SELECT place\_no as 'living\_in', place\_town as 'town\_of\_residence', place\_country as 'country\_of\_residence'

FROM place)y

CROSS JOIN:

Natural:

Mostrar todas las posibles bandas a las que pertensca cada musico mostrando el número de cada musico junto al número de cada banda

SQL:

SELECT m\_no as ‘player’, band.no as ‘band\_id’

FROM musician

CROSS JOIN band

LEFT JOIN:

Natural:

Mostrar el nombre de los músicos y las interpretaciones dirigidas por ellos si este no a interpretado igual mostrar el espacio vacío junto a su nombre

SQL:

SELECT m\_name as 'musician name', pfrmnc\_no as 'directed performance'

FROM musician m

LEFT JOIN performance pr ON pr.conducted\_by = m.m\_no

RIGHT JOIN:

Natural:

Mostrar el nombre de las bandas y las interpretaciones dadas por ellos si la banda no ha interpretado igual mostrar el espacio vacío junto a su nombre

SQL:

SELECT band\_name as 'band name', pfrmnc\_no as 'directed performance'

FROM performance pr

RIGHT JOIN band bd ON pr.gave = bd.band\_no

ORDER BY band\_name, pfrmnc\_no

RIGHT JOIN:

Natural:

Mostrar el nombre de las bandas y las interpretaciones dadas por ellos si la banda no ha interpretado igual mostrar el espacio vacío junto a su nombre

SQL:

SELECT band\_no, band\_name as 'band name', pfrmnc\_no as 'directed performance'

FROM performance pr

RIGHT JOIN band bd ON pr.gave = bd.band\_no

ORDER BY band\_no, pfrmnc\_no

FULL JOIN:

Natural:

Mostrar el nombre de los organizadores del concierto dados en el lugar “Assembly Rooms” y realizados después del primero de febrero de 1997

SQL:

SELECT m\_name

FROM concert

FULL JOIN musician ON concert\_orgniser = m\_no

WHERE (concert\_venue = 'Assembly Rooms') AND (con\_date > '01/02/1997')

Operadores de desconocido:

ISNULL/IFNULL:

Natural:

Mostrar el nombre de los músicos junto a ‘has not died’ si este no ha muerto o la fecha si este si tiene fecha de muerte registrada

SQL:

SELECT m\_name, IFNULL(died,'has not died')

FROM musician

COALESCE:

Natural:

Mostrar el nombre de los músicos junto a ‘has not died’ si este no ha muerto o la fecha si este si tiene fecha de muerte registrada

SQL:

SELECT m\_name, COALESCE(died,'has not died')

FROM musician

Operadores logicos:

EXISTS:

Natural:

Mostrar el nombre de los músicos que tocan violín

SQL:

SELECT m\_name

FROM musician

WHERE EXISTS (SELECT instrument

FROM performer

WHERE musician.m\_no = performer.perf\_is AND instrument = 'violin')

ORDER BY m\_name

ANY:

Natural:

Mostrar el nombre de los músicos que viven en USA

SQL:

SELECT m\_name

FROM musician

JOIN place ON place.place\_no = musician.living\_in

WHERE place\_country = ANY(SELECT place\_country

FROM place

WHERE place\_country = 'USA')

ALL:

Natural:

Mostrar el nombre de todas las bandas

SQL:

SELECT ALL band\_name

FROM band

WHERE TRUE

Operadores condicionales:

CASE:

Natural:

Mostrar el nombre de los musicos junto a los instrumentos que tocan, pero si estos no tocan un instrumento escribir “does not play an instrument”

SQL:

SELECT m\_name, CASE WHEN instrument IS NULL

THEN 'does not play an instrument'

ELSE instrumeNT

END

FROM musician m

LEFT JOIN performer pr ON pr.perf\_is= m.m\_no

ORDER BY m\_name