**MODELOS Y BASES DE DATOS**

**SQL Básico**

**2020-2 Guía autoestudio 2/6**

Johann Cepeda - Andrés Giraldo

**Investigación**

**A. NULL**

**1. ¿Qué significa?**

Un campo con un valor NULL es un campo sin valor. Un valor NULL es diferente de un valor cero o un campo que contiene espacios. Un campo con un valor NULL es uno que se ha dejado vacío durante la creación del registro

**2. ¿Resultado de operarlo con los diferentes tipos de operadores: aritméticos,**

**lógicos y de comparación?**

No es posible probar valores NULL con operadores de comparación, como =, <o <>. Tendremos que usar los operadores IS NULL y IS NOT NULL en su lugar.

**Operadores aritméticos**

Como Null no es un dato, sino un marcador para un valor ausente, usando operadores matemáticos Null da un resultado desconocido, que está representado por Null.

En el ejemplo siguiente, multiplicando 10 por resultados Null en Null:

10 \* NULL -- Result is NULL

**Operadores lógicos y de comparación**

Ya que Null no es un miembro de un dominio de datos, no se considera un «valor», sino más bien un marcador (o marcador de posición) indicando la ausencia de valor. Debido a esto, las comparaciones con Null nunca resulta verdadero o falso, pero siempre en un tercer resultado lógico, desconocido. El resultado lógico de la expresión siguiente, que compara el valor 10 a Null, es desconocido:

SELECT 10 = NULL -- Results in Unknown

Sin embargo, ciertas operaciones en Null pueden devolver valores si el valor ausente no es relevante para el resultado de la operación. Considere el siguiente ejemplo:

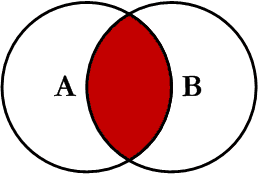
SELECT NULL OR TRUE -- Results in True

En este caso, el hecho de que el valor de la izquierda o es incognoscible es irrelevante, porque el resultado de la operación sería verdadero independientemente del valor de la izquierda.

**B. JUNTA**

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

Junta interna



Esta es la Unión más simple, más entendida y es la más común. Esta consulta devolverá todos los registros coincidentes entre dos tablas A Y B. Esta Unión se escribe como sigue:

SQL

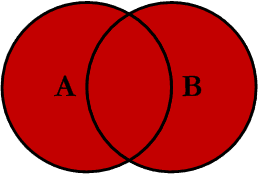
SELECT <select\_list>

FROM Table\_A A

INNER JOIN Table\_B B

ON A.Key = B.Key

Junta externa



Esta Unión también se puede referir como una Unión exterior completa o una Unión completa. Esta consulta devolverá todos los registros de ambas tablas, uniendo los registros de la tabla izquierda (tabla A) que coinciden con los registros de la tabla derecha (tabla B). Esta Unión se escribe como sigue:

SQL

SELECT <select\_list>

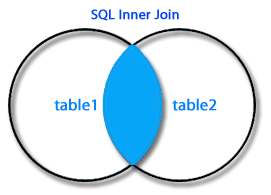
FROM Table\_A A

FULL OUTER JOIN Table\_B B

ON A.Key = B.Key

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

**INNER JOIN O JOIN:** Devuelve todas las filas cuando hay al menos una coincidencia en ambas tablas.

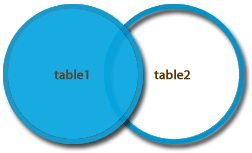


**JOIN NATURAL:** Es un tipo de JOIN estructurado de tal manera que, columnas con el mismo nombre de tablas asociadas aparecen sólo una vez. Se diferencia del INNER JOIN por el número de columnas que devuelve.

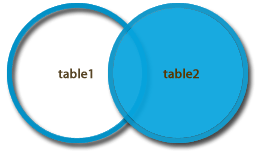
**CROSS JOIN:** devuelve el *producto cartesiano* de filas de tablas en el join. En otras palabras, va a producir filas que combinan cada fila de la primera tabla con cada fila de la segunda tabla.

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

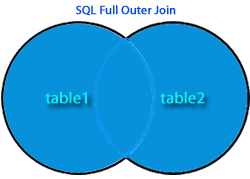
LEFT JOIN: Devuelve todas las filas de la tabla de la izquierda, y las filas coincidentes de la tabla de la derecha.



RIGHT JOIN: Devuelve todas las filas de la tabla de la derecha, y las filas coincidentes de la tabla de la izquierda.



FULL JOIN: Devuelve todas las filas de las dos tablas, la izquierda y la derecha. También se llama FULL OUTER JOIN.



**BIBLIOGRAFÍA**

Anónimo. Null(SQL). Obtenido de: https://en.wikipedia.org/wiki/Null\_(SQL)

Anónimo. Three-valued logic. Obtenido de: <https://en.wikipedia.org/wiki/Three-valued_logic>

[C.L. Moffatt](https://www.codeproject.com/script/Membership/View.aspx?mid=5909363),2009.Visual Representation of SQL Joins. Obtenido de: <https://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins>

Lázaro,2018. Principales tipos de JOINS en SQL. Obtenido de: https://diego.com.es/principales-tipos-de-joins-en-sql

Stackoverflow. CRUZADAS VS INNER JOIN. Obtenido de: <https://stackoverflow.com/questions/17759687/cross-join-vs-inner-join-in-sql-server-2008>

**Practica**

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

WHERE teamid = 'GER'

* 1. SELECT id,stadium,team1,team2

FROM game

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

* 1. SELECT teamname, COUNT(teamid)

FROM eteam JOIN goal ON teamid = id

GROUP BY teamname

* 1. SELECT stadium, COUNT(matchid)

FROM game JOIN goal ON matchid = id

GROUP BY stadium

* 1. SELECT matchid,mdate, COUNT(matchid)

FROM game JOIN goal ON matchid = id

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

GROUP BY matchid, mdate

* 1. SELECT matchid, mdate, COUNT(teamid)

FROM game JOIN goal ON matchid = id

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

GROUP BY matchid, mdate

* 1. SELECT mdate, team1, SUM(CASE WHEN teamid = team1 THEN 1 ELSE 0 END), team2, SUM(CASE WHEN teamid = team2 THEN 1 ELSE 0 END)

FROM game LEFT JOIN goal ON (id = matchid)

GROUP BY mdate,team1,team2

ORDER BY mdate, matchid, team1, team2

* **More 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 id

from movie

where title = 'Casablanca'

* 1. select name

from casting

join actor on (casting.actorid = actor.id)

where movieid = 11768

* 1. select name

from casting

join movie on(movie.id = casting.movieid)

join actor on(casting.actorid = actor.id)

where title = 'Alien'

* 1. select title

from movie

join casting on(casting.movieid = movie.id)

join actor on(casting.actorid = actor.id)

where name = 'Harrison Ford'

* 1. select title

from movie

join casting on(casting.movieid = movie.id)

join actor on(casting.actorid = actor.id)

where name = 'Harrison Ford' and casting.ord <> 1

* 1. select title, name

from casting

join actor on (casting.actorid = actor.id)

join movie on (casting.movieid = movie.id)

where ord = 1 and yr = 1962

* 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, name

from movie

join casting on (movie.id = casting.movieid and ord = 1)

join actor on (actor.id = casting.actorid)

where movie.id in (

select movieid

from casting

where actorid in(

select id from actor

where name = 'Julie Andrews'

)

)

* 1. SELECT name FROM movie

JOIN casting ON (movieid = movie.id and ord = 1)

JOIN actor ON actor.id = actorid

GROUP BY name asc

HAVING COUNT(\*) >= 15

* **SELF JOIN**
  1. select count(id)

from stops

* 1. select id

from stops

where name = 'Craiglockhart'

* 1. select id, name

from stops

join route on (stops.id = route.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 stopa.name='Haymarket' and stopb.name = 'Leith'

* 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 DISTINCT stopb.name, 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 a.company = 'LRT'

* 1. No se realize
* **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 (teacher.dept=dept.id)

* 1. SELECT teacher.name, dept.name

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

* 1. SELECT name, COALESCE(mobile,'07986 444 2266')

FROM teacher

* 1. SELECT COALESCE(teacher.name,'NONE'), COALESCE(dept.name,'None')

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

* 1. SELECT COUNT(name), COUNT(mobile)

FROM teacher

* 1. SELECT dept.name, COUNT(teacher.name)

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

GROUP BY dept.name

* 1. SELECT teacher.name,

CASE WHEN dept.id = 1

THEN 'Sci'

WHEN dept.id = 2

THEN 'Sci'

ELSE 'Art' END

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

* 1. SELECT teacher.name,

CASE

WHEN dept.id = 1

THEN 'Sci'

WHEN dept.id = 2

THEN 'Sci'

WHEN dept.id = 3

THEN 'Art'

ELSE 'None'

END

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

* **Consultas operadores de conjuntos**
  1. UNION, todos los trabajadores y si están con una consulta abierta.

SELECT Taken\_by, Status

FROM Issue

WHERE Status = 'Open'

UNION

SELECT Staff\_code, COALESCE('None')

FROM Staff

* 1. UNION ALL, muestra la referencia y el contacto o el nombre de la empresa

SELECT Company\_ref, First\_name

FROM Caller

UNION ALL

SELECT Company\_ref, Company\_name

FROM Customer

* 1. INTERSECT, las consultas cerradas asignadas a JE1

SELECT \* FROM Issue

WHERE status = 'Closed'

INTERSECT

SELECT \* FROM Issue

WHERE Assigned\_to = 'JE1'

* 1. EXCEPT, los operadores que no tienen consultas.

SELECT Staff\_code FROM Staff

EXCEPT

SELECT Taken\_by FROM Issue

* **Consultas junta interna y externa**
  1. select distinct Company\_name

from Customer

inner join Caller on(Customer.Company\_ref = Caller.Company\_ref)

* 1. select distinct First\_name

from Staff

inner join Level on(Staff.Level\_code = Level.Level\_code)

* 1. select Staff.First\_name, Level.Manager

from Staff

FULL OUTER JOIN Level

on(Staff.Level\_code = Level.Level\_code)

order by First\_name

* 1. select Staff.First\_name

from Staff

RIGHT JOIN Level

on(Staff.Level\_code = Level.Level\_code)

order by First\_name

* **Consultas para operador desconocido** 
  1. COALESCE, nos muestra el nombre, la dirección 1, y la dirección si la tiene y si no ‘Doesnt have’

SELECT Company\_name,Address\_1, COALESCE(Address\_2, 'Doesnt have')

FROM Customer

* 1. ISNULL, si el código no permite ejercer como manager, colocar ‘Cant manage’ de lo contrario colocar Y

SELECT Level\_code,ISNULL(Manager,'Cant manage')

FROM Level

* **Consultas operadores logicos**
  1. select Company\_name

from Customer

where exists(

select Telephone

from Customer

where Company\_name like '%Co.%'

)

* 1. select Call\_ref, Detail

from Issue

where Call\_date >= ALL(select Call\_date from Issue order by Call\_date)

* 1. select Call\_ref, Detail

from Issue

where Call\_date < ANY(select Call\_date from Issue order by Call\_date)

* **Consulta operador CASE**
  1. SELECT Call\_ref, Detail,

CASE

WHEN Call\_date <> '2017-08-04' THEN 'La fecha es diferente a 2017-08-04'

ELSE 'The quantity is under 30'

end

FROM Issue;