MODELOS Y BASES DE DATOS

SQL Básico

2019-02

Guía autoestudio 2/6

**Investigación.**

1. **NULL**
2. Significa que un valor en el campo de una tabla es vacío.
3. **Operadores aritméticos:** El resultado que se obtendrá al operar NULL con un valor numérico será NULL.

**Operadores lógicos y de comparación:** No es posible probar NULL con operadores de este tipo.

1. **JUNTA**

Junta interna (**INNER JOIN**)

Junta externa (**OUTER JOIN**)

1. La diferente es que el INNER JOIN se utiliza para unir los datos de dos o más tablas que tienen relación entre ellas, a contrario el OUTER JOIN, también junta los datos que no tienen relación entre dos tablas o más.
2. Un inner join de A y B entregará el resultado de la intersección de los conjuntos A y B. En otras palabras, la parte interna –intersección– en un diagrama de Venn.



1. Un outer join se puede trabajar de las siguientes maneras:

**Full outer join:** la palabra clave FULL OUTER JOIN devuelve todos los registros cuando hay una coincidencia en los registros de la tabla izquierda (tabla1) o derecha (tabla2).



**Right outer join:** devuelve todos los registros de la tabla derecha (tabla2) y los registros coincidentes de la tabla izquierda (tabla1). El resultado es NULL desde el lado izquierdo, cuando no hay coincidencia.



**Left outer join:** devuelve todos los registros de la tabla izquierda (tabla1) y los registros coincidentes de la tabla derecha (tabla2). El resultado es NULL desde el lado derecho, si no hay coincidencia.



**Práctica.**

**JOIN**

1. SELECT matchid, player FROM goal

JOIN game ON id = matchid

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 (game.id=goal.matchid)

WHERE teamid = "GER"

1. SELECT team1, team2, player FROM game

JOIN goal ON (game.id=goal.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 LIKE "Fernando Santos"

1. SELECT player FROM goal

JOIN game ON id=matchid

WHERE stadium LIKE "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(player)

FROM eteam

JOIN goal ON id=teamid

GROUP BY teamname

1. SELECT stadium, COUNT(player)

FROM game

JOIN goal ON (id=matchid)

GROUP BY stadium

1. SELECT matchid, mdate, COUNT(player) AS goals

FROM game

JOIN goal ON (matchid=id AND (team1 = 'POL' OR team2 = 'POL')) GROUP BY matchid, mdate

1. SELECT matchid, mdate, COUNT(player) FROM game

JOIN goal ON id=matchid

WHERE teamid = "GER"

GROUP BY matchid, mdate

1. SELECT mdate, team1,

SUM(CASE WHEN teamid = team1 THEN 1 ELSE 0 END) AS score1, team2, SUM(CASE WHEN teamid = team2 THEN 1 ELSE 0 END) AS score2 FROM game LEFT JOIN goal ON (id = matchid)

GROUP BY mdate,team1,team2

ORDER BY mdate, matchid, team1, team2

**MORE JOIN OPERATION**

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%'

1. SELECT actor.id FROM movie

JOIN actor

ON actor.id = movie.id

WHERE actor.name = 'Glenn Close'

1. SELECT id FROM movie

WHERE title = 'Casablanca'

1. SELECT name FROM actor

JOIN casting

ON casting.actorid = actor.id

JOIN movie

ON movieid=movie.id

WHERE title = 'Casablanca'

1. SELECT name FROM actor

JOIN casting

ON casting.actorid = actor.id

JOIN movie

ON movieid=movie.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 actor

JOIN casting

ON actorid = actor.id

JOIN movie

ON movieid = movie.id

WHERE name = 'Harrison Ford'AND ord!=1

1. SELECT title ,name FROM actor

JOIN casting ON actorid = actor.id

JOIN movie ON movie.id =movieid

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='Doris Day'

GROUP BY yr

HAVING COUNT(title) > 1

1. SELECT title, name FROM movie

JOIN casting ON movie.id = movieid

JOIN actor ON actor.id = actorid

WHERE name = 'Julie Andrews' AND ord = 1

1. . SELECT name FROM actor

JOIN casting ON (id = actorid AND (SELECT COUNT(ord) FROM casting WHERE actorid = actor.id AND ord=1)>=15)

GROUP BY name

1. SELECT title, COUNT(actorid) as cast FROM movie JOIN casting on id=movieid

WHERE yr = 1978

GROUP BY title

ORDER BY cast DESC

**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

RIGHT JOIN dept ON dept.id = teacher.dept

1. SELECT teacher.name, dept.name FROM teacher

LEFT JOIN dept ON dept.id = teacher.dept

1. SELECT name, COALESCE(mobile, "07986 444 2266") FROM teacher
2. SELECT teacher.name, COALESCE (dept.name, "None") FROM teacher

LEFT JOIN dept ON dept.id = teacher.dept

1. SELECT COUNT(name), COUNT(mobile) FROM teacher
2. SELECT dept.name, COUNT(teacher.name) FROM teacher

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

GROUP BY dept.name

1. SELECT teacher.name, CASE WHEN(teacher.dept = 1 OR teacher.dept=2) AND teacher.dept IS NOT NULL THEN "Sci" ELSE "Art" END

FROM teacher

1. SELECT teacher.name, CASE WHEN(teacher.dept = 1 OR teacher.dept=2) THEN "Sci"WHEN dept = 2 THEN "Art" ELSE "None"

END

FROM teacher

**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 company = 'LRT' AND num=4

1. SELECT company, num, COUNT(\*) AS visits

FROM route WHERE stop=149 OR stop=53

GROUP BY company, num

HAVING visits =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

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

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

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 stopa.name, a.company, a.num

FROM route a

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

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

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

WHERE stopb.name = 'Craiglockhart'

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 = 'Sighthill'

AND stopb.name = stopc.name

ORDER BY LENGTH(a.num), b.num, stopb.id, LENGTH(c.num), d.num

**Adventure Works**

**Operador de conjuntos**

1. **UNION**

**Mostrar el nombre de los productos que su precio unitario es menor que cien y mayor que 500**

SELECT Name FROM Product

JOIN SalesOrderDetail ON (SalesOrderDetail.ProductID = Product.ProductID)

WHERE UnitPrice >= "500"

UNION

SELECT Name FROM Product

JOIN SalesOrderDetail ON (SalesOrderDetail.ProductID = Product.ProductID)

WHERE UnitPrice <= "100"

1. **UNION ALL**

**Mostar el nombre y color de todos los productos donde el peso es menor que 20gr.**

SELECT Name FROM Product

WHERE Weight < 20

UNION ALL

SELECT Color FROM Product

WHERE Weight <20

1. **INTERSECT**

**Mostrar los productos que el precio por unidad es mayor que 200 y el modelo de producto igual a “bike”**

SELECT Name FROM Product

JOIN SalesOrderDetail ON (SalesOrderDetail.ProductID = Product.ProductID)

WHERE UnitPrice > “200"

INTERSECT

SELECT Name FROM Product

JOIN ProductModel ON(ProductModel.ProductModelID = Product.ProductID)

WHERE ProductModelID = “Bike”

1. **EXTRACT**

**Muestra la semana de la fecha del pedido**

SELECT EXTRACT(WEEK FROM OrderDate) FROM SalesOrderHeader

**5. IN**

**Muestra todos la table de productos que sean rojo o azul**

SELECT \* FROM Product

WHERE Color IN ('blue','red')

**Consulta interna y consulta externa**

**Counsulta interna**

**NATURAL JOIN**

**Nos unira la table Customer y CustomerAddress completamente ya que tiene en comun el CustomerID**

SELECT \* FROM Customer

NATURAL JOIN CustomerAddress

SELECT \* FROM Customer

NATURAL JOIN CustomerAddress

**CROSS JOIN**

SELECT Customer.FirstName, Customer.MiddleName , Address.City, Address.Stateprovince FROM Customer

CROSS JOIN Address

**Consulta externa**

**LEFT JOIN**

SELECT LastName, OrderDate FROM Customer

LEFT JOIN SalesOrderHeader

ON Customer.CustomerID = SalesOrderHeader.CustomerID

**RIGHT JOIN**

SELECT LastName, OrderDate FROM Customer

RIGHT JOIN SalesOrderHeader

ON Customer.CustomerID = SalesOrderHeader.CustomerID

**Operador desconocido**

1. **NULL**

SELECT name FROM Product

WHERE Weight IS NOT NULL

1. **COALESCE**

SELECT name , COALESCE (Color, “black”)

FROM Product

**Operadore lógicos**

**EXISTS**

SELECT City FROM Address

WHERE EXISTS(SELECT\* FROM SalesOrderHeader WHERE TaxAmt>1000)

**Operador CASE**

1. **CASE**

SELECT name

CASE WHEN color = “red” THEN “black” END AS color

FROM Proddcut

**Bibliografía:**

<https://www.w3schools.com/sql/default.asp>

<https://es.stackoverflow.com/questions/36/cu%C3%A1l-es-la-diferencia-entre-un-inner-y-un-outer-join>