Investigación

A

- ✓ En SQL, NULL no es un valor. Es un estado que indica que el valor de ese item es desconocido o no existente. No es cero o blanco o una “cadena vacía” y no se comporta como ninguno de esos valores. Pocas cosas en SQL ,llevan a tanta confusión como NULL, y será difícil de entender mientras no entiendas la siguiente simple
- ✓ El resultado al hacer uso del NULL con operadores aritméticos o lógicos es un error, pero muchas veces es usado con operadores de comparación para revisar si alguna fila de la tabla tiene valor nulo o no.

B

- ✓ En la junta interna cada registro de la tabla A es combinado con los correspondientes en la tabla B que satisfagan la condición que se especifique en la junta. Por otro lado, en la junta externa 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.
- ✓ Junta interna:
  - JOIN
  - NATURAL JOIN
  - CROSS JOIN
- ✓ Junta externa:
  - lEFT JOIN
  - RIGHT JOIN
  - FULL JOIN

Practica

A

- **JOIN**

1.

```
SELECT matchid,player
FROM goal
WHERE teamid= 'GER'
```

1.

```
SELECT id, stadium, team1, team2
FROM game
WHERE id= 1012
```

2.
```
SELECT player,teamid,stadium,mdate
 FROM game JOIN goal
 ON (id=matchid) WHERE teamid= 'GER'
```
3.
```
SELECT team1, team2, player
 FROM goal JOIN game ON (id= matchid)
 WHERE player LIKE 'Mario%'
```
4.
```
SELECT player, teamid, coach,gtime
 FROM goal JOIN eteam ON (id= teamid)
 WHERE gtime<=10
```
5.
```
SELECT mdate,teamname
 FROM game JOIN eteam ON team1= eteam.id
 WHERE coach= 'Fernando Santos'
```
6.
```
SELECT player
 FROM game JOIN goal ON id= matchid
 WHERE stadium= 'National Stadium, Warsaw'
```
8.
```
SELECT DISTINCT player
FROM game JOIN goal ON id= matchid
WHERE teamid!= 'GER' AND (team1= 'GER' OR  team2= 'GER')
```
9.
```
SELECT teamname, count(gtime)
FROM eteam JOIN goal ON id=teamid
GROUP BY teamname
```
10.
```
SELECT stadium, count(gtime)
FROM game JOIN goal ON id= matchid
GROUP BY stadium
```
11.

SELECT matchid,mdate,z **FROM**

(select matchid,count(teamid) **as** z

FROM game **JOIN** goal **ON** id= matchid

WHERE team1= 'POL' **or** team2= 'POL'

GROUP BY matchid) AS X

JOIN game **ON** id= matchid

12.

SELECT matchid,mdate, z **FROM**

(**SELECT** matchid,**count**(teamid) **AS** z

FROM game **JOIN** goal **ON** Id= matchid

WHERE (team1= 'GER' **or** team2= 'GER') **and** teamid= 'GER'

GROUP BY matchid) **AS** x **JOIN** game **ON** id= matchid

13.

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** matchid = id **GROUP BY** id,mdate,matchid,team1,team2 **ORDER BY** mdate,matchid, team1,team2

- **More JOIN operation**

1.

SELECT id, title

FROM movie

WHERE yr=1962

2.

SELECT yr

FROM movie

WHERE title = 'Citizen Kane'

3.

SELECT id,title,yr movie

Where title **LIKE** 'Star Trek%'

ORDER BY title **DESC**

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** (casting.actorid=actor.id)

WHERE movieid=11768

7.

SELECT name

FROM casting **JOIN** actor **ON** (casting.actorid=actor.id)

WHERE movieid=(**SELECT** id **FROM** movie **WHERE** title='Alien')

8.

SELECT title

FROM movie **JOIN** casting **ON**(movie.id=casting.movieid) **JOIN** actor **ON** (casting.actorid = actor.id)

WHERE actor.name = 'Harrison Ford'

9.

SELECT title

FROM movie **JOIN** casting **ON**(movie.id=casting.movieid) **JOIN** actor **ON** (casting.actorid = actor.id)

WHERE actor.name = 'Harrison Ford' **and** ord<>1

10.

SELECT title,name

FROM movie **JOIN** casting **ON**(movie.id=casting.movieid) JOIN actor **ON** (casting.actorid = actor.id)

WHERE ord=1 **and** yr=1962

11.

SELECT yr,**COUNT**(title) **FROM**

```
movie JOIN casting ON movie.id=movieid
JOIN actor   ON actorid=actor.id
where name='John Travolta'
GROUP BY yr
HAVING COUNT(title)=(SELECT MAX(c) FROM
(SELECT yr,COUNT(title) AS c FROM
 movie JOIN casting ON movie.id=movieid
 JOIN actor   ON actorid=actor.id
where name='John Travolta'
GROUP BY yr) AS t)
```

12.

```
SELECT title,name
FROM movie JOIN casting ON (movie.id = casting.movieid and
ord=1) JOIN actor ON(casting.actorid = actor.id)
WHERE movie.id IN (SELECT movieid FROM casting
WHERE actorid IN (
SELECT id FROM actor
WHERE name='Julie Andrews'))
```

13.

```
SELECT actor.name
FROM actor
JOIN casting ON (casting.actorid = actor.id)
WHERE ord = 1
GROUP BY name
HAVING COUNT(*) >= 30
```

14.

```
SELECT movie.title,count(actorid)
FROM movie JOIN casting ON(movie.id=casting.movieid)
WHERE movie.yr= 1978
GROUP BY movie.id
ORDER BY COUNT(actorid) DESC
```

- **Using NULL**

1.

```
SELECT name FROM teacher WHERE dept IS NULL
```

2.

```
SELECT teacher.name, dept.name
FROM teacher INNER JOIN dept
ON (dept.id= teacher.dept)
```

3.

```
SELECT teacher.name,dept.name
FROM teacher LEFT JOIN dept ON dept.id= teacher.dept
```

4.

```
SELECT teacher.name,dept.name
 FROM teacher RIGHT JOIN dept ON dept.id= teacher.dept
```

5.

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

6.

```
SELECT teacher.name,COALESCE(dept.name,'None') AS dept
 FROM teacher LEFT JOIN dept ON dept.id= teacher.dept
```

7.

```
SELECT COUNT(name) AS cant_name,COUNT(mobile)
 AS cant_mobile FROM teacher
```

8.

```
SELECT dept.name, COUNT(teacher.name) AS Staff
 FROM teacher RIGHT JOIN dept ON teacher.dept= dept.id
 GROUP BY dept.name
```

9.

```
SELECT name,
CASE
WHEN dept= 1 OR dept= 2
```

THEN 'Sci'

ELSE 'Art' **END** AS dept **FROM** teacher

10.

SELECT name,

CASE

WHEN dept= 1 or dept= 2

THEN 'Sci'

WHEN dept= 3 **THEN** 'Art'

ELSE 'None' **END AS** dept **FROM** teacher

- **Numeric example**

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 question='Q15'

AND score>=100

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 = '(H) Creative Arts and Design'

OR subject='(8) Computer Science')

GROUP BY subject

5.

SELECT subject,**SUM**(response * A_STRONGLY_AGREE/100)

FROM nss

WHERE question='Q22' **AND**(subject = '(H) Creative Arts and Design'

OR subject='(8) Computer Science')

GROUP BY subject

6.

SELECT subject,**ROUND**(**SUM**(response* A_STRONGLY_AGREE) /**SUM**(response))

FROM nss

WHERE question='Q22' **AND** (subject = '(H) Creative Arts and Design'

OR subject='(8) Computer Science')

GROUP BY subject

7.

SELECT institution,**ROUND**(**SUM**(score * response)/**SUM**(response))

FROM nss

WHERE question='Q22'

AND (institution **LIKE** '%Manchester%')

GROUP BY institution

ORDER BY institution

7.

SELECT institution,**SUM**(sample),**SUM**(**CASE WHEN** subject = '(8) Computer Science' **THEN** sample **ELSE** 0 **END**)

FROM nss

WHERE question='Q01'

AND (institution **LIKE** '%Manchester%')

GROUP BY institution

- **Self JOIN**

1.

SELECT COUNT(name) **AS** stops **FROM** stops

2.

SELECT id **FROM** stops **WHERE** name= 'Craiglockhart'

3.

SELECT id,name **FROM** route **INNER JOIN** stops

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

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(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** (a.stop=stopa.id)

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

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

9.

SELECT name,company, num

FROM (**SELECT** X.num,stop,company **FROM** route **JOIN**

(**SELECT** num **FROM** stops **JOIN** route **ON** id= stop

WHERE company= 'LRT' **AND** name= 'Craiglockhart')

AS X **ON** X.num= route.num) **AS** Z **JOIN** stops

WHERE id= stop **AND** company= 'LRT'

10.

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

- **Tutorial Quizzes se encuentra en la carpeta que esta comprimida en el trabajo**

B)

1)

UNION ALL

SELECT 'BORN IN', m_name **FROM** musician,place **WHERE** (born_in = place_no) **AND** place_town= 'Glasgow' **UNION ALL**

SELECT 'LIVES IN',m_name FROM musician,place WHERE (living_in= place_no) AND place_town= 'Glasgow'

UNION ALL

SELECT 'IN_BAND_IN',m_name **FROM** band,place,musician,performer,plays_in **WHERE** band_home= place_no **AND** place_town= 'Glasgow'

AND band_id= band_no **AND** player= perf_no **AND** perf_is=m_no **ORDER BY** m_name

UNION

(**SELECT** place_town **AS** ciudad_pais

FROM musician **JOIN** place **ON** place_no= born_in)

UNION

(**SELECT** place_country **AS** Country

FROM musician **JOIN** place **ON** place_no= living_in)

INTERSECT

No lo pudimos realizar,ya que se tuvo un error de sintaxis que no se pudo solucionar

EXTRACT

SELECT EXTRACT(**YEAR_MONTH**

FROM (**SELECT** died **FROM** musician

WHERE m_name= 'Harriet Smithson'))

2)

JOIN

SELECT concert_venue **AS** Venue, place_country AS Country

FROM place **JOIN** concert **ON** concert_in= place_no

CROSS JOIN

SELECT m_name **AS** Name, place_country **AS** Country

FROM place **CROSS JOIN** musician **ON** born_in = place_no

WHERE place_country= 'USA'

LEFT JOIN

SELECT m_name AS name, compositions,**COALESCE**(instrument, 0) **AS** instrument

FROM(**SELECT** m_name,**COALESCE**(compositions, 0) **AS** compositions

FROM (**SELECT DISTINCT** m_name from musician,place **WHERE** born_in= place_no **AND** (place_country= 'England' **OR** place_country= 'Scotland')) **AS** PRINCIPAL

LEFT JOIN

(**SELECT COUNT**(cmpr_no) **AS** compositions, m_name **AS** name **FROM** musician,composer,has_composed **WHERE** m_no= comp_is AND comp_no= cmpr_no **GROUP BY** (name)) **AS** COMPOSICIONES **ON** name= m_name) **AS** UNO

LEFT JOIN

(**SELECT COUNT**(perf_is) **AS** instrument , m_name AS name2 **FROM** musician,performer **WHERE** perf_is= m_no **GROUP BY** (name2)) **AS** DOS

ON m_name= name2

RIGHT JOIN

SELECT re **AS** Name,band_name **AS** 'Jeff"s band' ,n **AS** 'Sue"s band'

FROM (**SELECT DISTINCT** m_name **AS** re,band_name **FROM** band,place,musician,performer,plays_in **WHERE** perf_is=m_no **AND** player= perf_no **AND** band_id= band_no **AND** band_name **IN** (**SELECT**

band_name **FROM**
band,place,musician,performer,plays_in **WHERE**
perf_is=m_no **AND** player= perf_no **AND** band_id=
band_no **AND** m_name = 'Jeff Dawn')) **AS** X

RIGHT OUTER JOIN

(**SELECT DISTINCT** m_name,band_name **AS** n
FROM band,place,musician,performer,plays_in
WHERE perf_is=m_no **AND** player= perf_no **AND**
band_id= band_no **AND** band_name **IN** (**SELECT**
band_name **FROM**
band,place,musician,performer,plays_in **WHERE**
perf_is=m_no **AND** player= perf_no **AND** band_id=
band_no **AND** m_name = 'Sue Little')) **AS** Y

ON re=m_name

WHERE re **IS NOT NULL**

3)

ISNULL:

SELECT m_name, **ISNULL**(died) **FROM** musician

COALESCE:

SELECT m_name **AS** name,
compositions,**COALESCE**(instrument, 0)
AS instrument **FROM**

(**SELECT** m_name,COALESCE(compositions, 0) **AS**
compositions **FROM** (**SELECT DISTINCT** m_name
FROM musician,place **WHERE** born_in = place_no
AND (place_country= 'England' **OR** place_country=
'Scotland')) AS PRINCIPAL

LEFT JOIN

(**SELECT COUNT**(cmpr_no) **AS** compositions,
m_name **AS** name **FROM**
musician,composer,has_composed **WHERE** m_no=
comp_is **AND** comp_no= cmpr_no **GROUP BY**
(name)) **AS** COMPOSICIONES **ON** name=
m_name)**AS** UNO

LEFT JOIN

(**SELECT COUNT**(perf_is) **AS** instrument ,
m_name **AS** name2 **FROM** musician,performer

WHERE perf_is=m_no **GROUP BY** (name2)) **AS** DOS

ON m_name= name2

4)

ALL:

SELECT instrument

FROM(**SELECT** count(perf_is) as n,instrument **FROM** performer **GROUP BY** instrument) **AS** NN

WHERE n <= **ALL**(**SELECT COUNT**(perf_no) **FROM** performer **GROUP BY** instrument)

ANY

SELECT m_name

FROM musician

WHERE m_name **<> ANY**(**SELECT** m_name **FROM** musician **WHERE** m_name='Fred Bloggs')

EXISTS

SELECT m_name **FROM** musician

WHERE EXISTS (**SELECT** m_name **FROM** musician **WHERE** m_name= 'Ana')

SELECT m_name **FROM** musician

WHERE EXISTS (**SELECT** m_name **FROM** musician **WHERE** died **IS NULL**)

5)

CASE

SELECT m_name **AS** name,

CASE

WHEN died **IS NULL THEN** 'Vive'

ELSE 'Muerto'

END AS X

FROM musician