**INVESTIGACIÓN**

A. NULL

1. ¿Qué significa?

En SQL, NULL no es un valor. Es un estado que indica que el valor que ese atributo es un valor desconocido o que no existe. No es creo o blanco o una “concatenación vacía” y no se comporta como ninguno de esos valores. Pocas cosas en SQ, llevan a tanta confusión como NULL y será difícil de comprender mientras no se entienda la siguiente simple. (Firebird, s.f.)

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

comparación?

El resultado al hacer uso del NULL con operadores aritméticos o lógicos es un error, pero la mayoría de las veces es implementada con operadores de comparación para hacer una revisión si en alguna fila de la tabla tiene un valor nulo o no.

B. JUNTA

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

En la junta interna cada registro de la tabla A es combinado con los correspondientes de la tabla B que satisfacen las condiciones que se especifiquen en la junta. Partiendo de otro punto de vista, la junta externa no se necesita 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. (Tutorial, s.f.)

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

- join

- NATURAL join

- CROSS join

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

- LEFT join

- RIGHT join

- FULL join

**PRACTICA**

**A)**

JOIN

1.Modify it to show the matchid and player name for all goals scored by Germany. To identify German players, check for: teamid = 'GER'

SELECT matchid, player

FROM goal

WHERE teamid = 'GER'

2.Show id, stadium, team1, team2 for just game 1012

SELECT id,stadium,team1,team2

FROM game

WHERE id = 1012

3.Modify it to show the player, teamid, stadium and mdate for every German goal.

SELECT player,teamid,stadium,mdate

FROM game JOIN goal ON (id=matchid)

WHERE teamid = 'GER'

4.Show the team1, team2 and player for every goal scored by a player called Mario player LIKE 'Mario%'

SELECT team1, team2, player

FROM game JOIN goal ON (id=matchid)

WHERE player LIKE 'Mario%'

5. Show player, teamid, coach, gtime for all goals scored in the first 10 minutes gtime<=10

SELECT player, teamid, coach, gtime

FROM goal JOIN eteam on teamid=id

WHERE gtime<=10

6.List the the dates of the matches and the name of the team in which 'Fernando Santos' was the team1 coach.

SELECT mdate, teamname

FROM game JOIN eteam on game.team1=eteam.id

WHERE coach = 'Fernando Santos'

7.List the player for every goal scored in a game where the stadium was 'National Stadium, Warsaw'

SELECT player

FROM game JOIN goal on id=matchid

WHERE stadium ='National Stadium, Warsaw'

8. Instead show the name of all players who scored a goal against Germany.

SELECT DISTINCT player

FROM game JOIN goal ON matchid = id

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

9. Show teamname and the total number of goals scored.

SELECT teamname, COUNT(player)

FROM eteam JOIN goal ON eteam.id=goal.teamid

GROUP BY teamname

10. Show the stadium and the number of goals scored in each stadium.

SELECT stadium, COUNT(player)

FROM game JOIN goal ON id=matchid

GROUP BY stadium

11. For every match involving 'POL', show the matchid, date and the number of goals scored.

SELECT matchid,mdate, COUNT(teamid)

FROM game JOIN goal ON matchid = id

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

GROUP BY matchid

12. For every match where 'GER' scored, show matchid, match date and the number of goals scored by 'GER'

SELECT matchid,mdate, COUNT(player)

FROM game JOIN goal ON matchid = id

WHERE teamid = 'GER'

GROUP BY matchid

13. List every match with the goals scored by each team as shown. This will use "CASE WHEN" which has not been explained in any previous exercises.

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 JOIN goal ON matchid = id

GROUP BY id

ORDER BY mdate,matchid,team1,team2

MORE JOIN

1.List the films where the yr is 1962 [Show id, title]

SELECT id, title

FROM movie

WHERE yr=1962

2.When was Citizen Kane released?

SELECT yr

FROM movie

WHERE title = 'Citizen Kane'

3.Star Trek movies

SELECT id,title,yr

FROM movie

WHERE title LIKE 'Star Trek%'

ORDER BY yr

4.id for actor Glenn Close

SELECT id

FROM actor

WHERE name = 'Glenn Close'

5.What is the id of the film 'Casablanca'

SELECT id

FROM movie

WHERE title = 'Casablanca'

6.Cast list for Casablanca

SELECT name

FROM movie, casting, actor

WHERE movie.id = casting.movieid AND casting.actorid = actor.id AND movie.id = 11768

7. Obtain the cast list for the film 'Alien'

SELECT name

FROM movie JOIN casting ON movie.id=casting.movieid JOIN actor ON casting.actorid=actor.id

WHERE movie.title='Alien'

8.List the films in which 'Harrison Ford' has appeared

SELECT title

FROM movie JOIN casting ON movie.id=casting.movieid JOIN actor ON casting.actorid=actor.id

WHERE actor.name = 'Harrison Ford'

9.List the films where 'Harrison Ford' has appeared - but not in the starring role.

SELECT title

FROM movie JOIN casting ON movie.id=casting.movieid JOIN actor ON casting.actorid=actor.id

WHERE actor.name = 'Harrison Ford' AND casting.ord <> 1

10.List the films together with the leading star for all 1962 films.

SELECT title, name

FROM movie JOIN casting ON movie.id=casting.movieid JOIN actor ON casting.actorid=actor.id

WHERE casting.ord = 1 AND movie.yr = 1962

11.Which were the busiest years for 'Rock Hudson', show the year and the number of movies he made each year for any year in which he made more than 2 movies.

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

12.List the film title and the leading actor for all of the films 'Julie Andrews' played in.

SELECT title, name

FROM movie, casting, actor

WHERE movieid=movie.id AND actorid=actor.id AND ord=1 AND movieid IN

(SELECT movieid FROM casting, actor

WHERE actorid=actor.id

AND name='Julie Andrews')

13. Obtain a list, in alphabetical order, of actors who've had at least 15 starring roles.

SELECT name

FROM casting JOIN actor ON casting.actorid=actor.id

WHERE casting.ord = 1

GROUP BY name

HAVING COUNT(casting.movieid) >= 15

14.List the films released in the year 1978 ordered by the number of actors in the cast, then by title.

SELECT title, COUNT(actorid)

FROM movie JOIN casting ON movie.id=casting.movieid

WHERE movie.yr = 1978

GROUP BY title

ORDER BY 2 DESC

15.List all the people who have worked with 'Art Garfunkel'.

SELECT name

FROM movie JOIN casting ON id = movieid JOIN actor ON actor.id=actorid

WHERE title IN (SELECT title FROM movie JOIN casting ON id = movieid JOIN actor ON actor.id=actorid WHERE name = 'Art Garfunkel') AND name!='Art Garfunkel'

USING NULL

1.List the teachers who have NULL for their department.

SELECT name

FROM teacher

WHERE dept IS null

2.Note the INNER JOIN misses the teachers with no department and the departments with no teacher.

SELECT teacher.name, dept.name

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

3.Use a different JOIN so that all teachers are listed.

SELECT teacher.name, dept.name

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

4.Use a different JOIN so that all departments are listed.

SELECT teacher.name, dept.name

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

5.Show teacher name and mobile number or '07986 444 2266'

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

FROM teacher

6.Use the COALESCE function and a LEFT JOIN to print the teacher name and department name.

SELECT teacher.name, COALESCE(dept.name,'None')

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

7.Use COUNT to show the number of teachers and the number of mobile phones.

SELECT COUNT(teacher.name), COUNT(mobile)

FROM teacher

8.Use COUNT and GROUP BY dept.name to show each department and the number of staff.

SELECT dept.name, COUNT(teacher.name)

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

GROUP BY dept.name

9.Use CASE to show the name of each teacher followed by 'Sci' if the teacher is in dept 1 or 2 and 'Art' otherwise.

SELECT name, CASE WHEN dept IN (1,2) THEN 'Sci' ELSE 'Art' END

FROM teacher

10. Use CASE to show the name of each teacher followed by 'Sci' if the teacher is in dept 1 or 2, show 'Art' if the teacher's dept is 3 and 'None' otherwise.

SELECT name, CASE WHEN dept IN (1,2)

THEN 'Sci'

WHEN dept=3 THEN 'Art'

ELSE 'None' END

FROM teacher

SELF JOIN

1.How many stops are in the database.

SELECT COUNT(\*)

FROM stops

2.Find the id value for the stop 'Craiglockhart'

SELECT id

FROM stops

WHERE name = 'Craiglockhart'

3.Give the id and the name for the stops on the '4' 'LRT' service.

SELECT id, name

FROM stops, route

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

4.The query shown gives the number of routes that visit either London Road (149) or Craiglockhart (53). Run the query and notice the two services that link these stops have a count of 2. Add a HAVING clause to restrict the output to these two routes.

SELECT company, num, COUNT(\*)

FROM route WHERE stop=149 OR stop=53

GROUP BY company, num

HAVING COUNT(\*) = 2

5.Execute the self join shown and observe that b.stop gives all the places you can get to from Craiglockhart, without changing routes. Change the query so that it shows the services from Craiglockhart to London Road.

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.The query shown is similar to the previous one, however by joining two copies of the stops table we can refer to stops by name rather than by number. Change the query so that the services between 'Craiglockhart' and 'London Road' are shown.

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.Give a list of all the services which connect stops 115 and 137 ('Haymarket' and 'Leith')

SELECT DISTINCT R1.company, R1.num

FROM route R1 JOIN route R2 ON R1.num=R2.num AND R1.company=R2.company

WHERE R1.stop=115 AND R2.stop=137

8.Give a list of the services which connect the stops 'Craiglockhart' and 'Tollcross'

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'

9.Give a distinct list of the stops which may be reached from 'Craiglockhart' by taking one bus, including 'Craiglockhart' itself, offered by the LRT company. Include the company and bus no. of the relevant services.

SELECT DISTINCT S2.name, R2.company, R2.num

FROM stops S1, stops S2, route R1, route R2

WHERE S1.name='Craiglockhart' AND S1.id=R1.stop AND R1.company=R2.company AND R1.num=R2.num AND R2.stop=S2.id

10.Find the routes involving two buses that can go from Craiglockhart to Lochend.Show the bus no. and company for the first bus, the name of the stop for the transfer,and the bus no. and company for the second bus.

**QUICES**

**1.** **JOIN Quiz**

**Imagen que contiene captura de pantalla

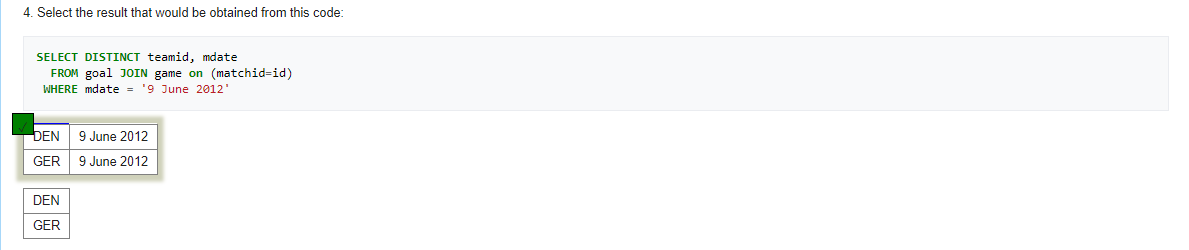
Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

****

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**2.** **JOIN Quiz 2**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**3.** **Using Null Quiz**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**4.** **Self join Quiz**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**Imagen que contiene captura de pantalla

Descripción generada automáticamente**

**B)**

-5 consultas: una para cada operador de conjuntos

1.UNION

(SELECT ProductID FROM Product WHERE Color = 'Red')

UNION

(SELECT ProductID FROM SalesOrderDetail)

2.UNION ALL

(SELECT CompanyName FROM Customer)

UNION ALL

(SELECT CustomerID FROM Customer)

UNION ALL

(SELECT AddressID FROM Address)

3.INTERSECT

(SELECT ProductID FROM Product WHERE Color = 'Red')

INTERSECT

(SELECT ProductID FROM SalesOrderDetail)

4.EXTRACT

SELECT EXTRACT(MONTH FROM "1999-09-08")

5.IN

SELECT City FROM Address WHERE CountryRegion IN ('Canada','United Kingdom')

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

1.JOIN

SELECT DISTINCT ProductCategory.Name

FROM Product

JOIN

ProductCategory ON Product.ProductCategoryID = ProductCategory.ProductCategoryID

WHERE Product.Color IN ('Red','Gold')

2.CROSS JOIN

SELECT \*

FROM Customer CROSS JOIN Address

3.LEFT JOIN

SELECT SubTotal

FROM SalesOrderHeader

JOIN LEFT

SalesOrderDetail ON SalesOrderHeader.SalesOrderID = SalesOrderDetail.SalesOrderID

4.RIGHT JOIN

SELECT OrderDate

FROM SalesOrderHeader

JOIN RIGHT

SalesOrderDetail ON SalesOrderHeader.SalesOrderID = SalesOrderDetail.SalesOrderID

-2 consultas: una para cada operador de desconocido

1.IS NULL

SELECT ISNULL("NULL")

2.COALESCE

SELECT COALESCE(NULL, 890, 350, 'Silver')

-3 consultas: una para cada uno de los tipos de operadores lógicos

1. EXISTS

SELECT DISTINCT City

FROM Address

WHERE EXISTS (SELECT \* FROM CustomerAddress WHERE CustomerID LIKE '1%')

2.ANY

SELECT FirstName

FROM Customer, SalesOrderHeader

WHERE SalesOrderHeader.CustomerID = Customer.CustomerID AND CompanyName = ANY (SELECT CompanyName FROM Customer WHERE LastName Like 'C%')

3.ALL

SELECT CustomerID

FROM CustomerAddress

WHERE AddressType = 'Main Office' AND AddressID = all (SELECT e.CustomerID FROM Customer as e WHERE e.LastName LIKE 'P%')

-1 consulta: para el operador CASE

SELECT SalesOrderID, CASE WHEN SubTotal > 10000 THEN 'OrderBT' ELSE 'Medium' END

FROM SalesOrderHeader

# Bibliografía

Firebird. (s.f.). Obtenido de https://www.firebirdsql.org/manual/es/nullguide-es-queesnull.html

Tutorial, R. (s.f.). Obtenido de https://riptutorial.com/es/sql/example/14930/diferencias-entre-uniones-internas---externas