AUTOGUIA 2

MODELOS Y BASES DE DATOS

ESTUDIANTES:

JUAN SEBASTIAN PUENTES JULIO

CHRISTIAN ALFONSO ROMERO

DOCENTE:

MARIA IRMA ROZO

INVESTIGACIÓN

A. NULL

- 1. ¿Qué significa?
 - NULL es un literal que representa una referencia nula que no hace referencia a ningún objeto.
- 2. ¿Resultado de operarlo con los diferentes tipos de operadores: aritméticos, lógicos y de comparación?
 - Comparación: IS NOT NULL devuelve una tabla con los registros en donde no haya un dato en blanco o no existente, mientras que, IS NULL devuelve la misma con datos desconocidos.
 - Aritmeticos: Cualquier operación con NULL resulta en NULL.
 - Lógico: El resultado de una operación lógica es True o False o Null, ya que lo que hace es la validación de si se cumple o no.

B. JUNTA

- 1. ¿ Cuáles son las diferencias entre junta interna y externa?
 - La junta interna devuelve solo los registros que coinciden en una tabla A y una tabla B, mientras que, la junta externa devuelve todas las filas cuando hay una coincidencia en una u otra tabla.
- 2. ¿ Qué opciones se tienen para la junta interna?
 - Se tiene tres opciones:

JOIN (INNER JOIN): Devuelve solo los registros que coinciden en una tabla A y una tabla B.

NATURAL JOIN: Se combinan columnas que tienen el mismo nombre en ambas tablas. CROSS JOIN (Producto Cartesiano): Combina ambas tablas sin importar si cumplen una condición o no. Este join toma cada fila de una tabla y la une a cada fila de otra tabla.

- 3. ¿ Qué opciones se tienen para la junta externa?
 - Se tienen tres opciones:

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

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

FULL JOIN: Devuelve todas las filas cuando hay una coincidencia en una u otra tabla

PRÁCTICA

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 AND teamid = "GER")
```

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

```
SELECT team1,team2,player
FROM goal JOIN game ON (player LIKE "Mario%" AND matchid = id)
```

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 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** (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 (matchid = id)

WHERE stadium = 'National Stadium, Warsaw'

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

SELECT DISTINCT player

FROM goal **JOIN** game **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**(gtime)

FROM eteam JOIN goal ON id=teamid

GROUP BY teamname

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

SELECT stadium, **COUNT**(gtime)

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**(gtime)

FROM game JOIN goal ON matchid = id

WHERE (team1 = 'POL' \mathbf{OR} team2 = 'POL')

GROUP BY matchid

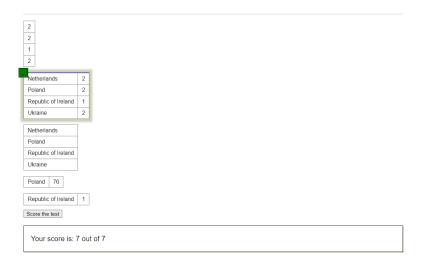
12. For every match where 'GER' scored, show matchid, match date and the number of goals scored by 'GER'

SELECT matchid, mdate, COUNT(gtime)
FROM goal JOIN game 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 LEFT JOIN goal ON matchid = id
GROUP BY mdate, team1, team2
ORDER BY mdate, matchid, team1, team2

JOIN QUIZ



MORE JOIN

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

SELECT id, title **FROM** movie **WHERE** yr=1962

2. Give the year of 'Citizen Kane'.

SELECT yr
FROM movie
WHERE title = 'Citizen Kane'

3. List all of the Star Trek movies, including the id, title and yr (all of these movies include the words Star Trek in the title). Order results by year.

SELECT id, title, yr
FROM movie
WHERE title LIKE "Star Trek%"
ORDER BY yr

4. What id number does the actor 'Glenn Close' have?

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. Obtain the cast list for 'Casablanca'.

SELECT name FROM casting JOIN actor ON (actorid = id) WHERE movieid = 11768

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

SELECT name
FROM movie JOIN casting ON (id = movieid) JOIN actor ON
(casting.actorid = actor.id)
WHERE title = 'Alien'

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

SELECT title

FROM movie **JOIN** casting **ON** (id = 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. [Note: the ord field of casting gives the position of the actor. If ord=1 then this actor is in the starring role]

SELECT title

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

WHERE actor.name = "Harrison Ford" AND NOT ord = 1

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

SELECT title, actor.name

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

WHERE ord = 1 AND 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 **JOIN** casting **ON** (movie.id = movieid **AND** ord = 1)

JOIN actor **ON** actorid = actor.id

WHERE movie.id IN (SELECT movieid FROM casting WHERE actorid IN (SELECT id FROM actor WHERE name = "Julie Andrews"))

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

SELECT name

FROM actor **JOIN** casting **ON** (actor.id = actorid AND ord = 1)

GROUP BY name

HAVING COUNT(ord) >= 15

ORDER BY name

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

SELECT title, **COUNT**(title)

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

WHERE yr = 1978

GROUP BY title

ORDER BY COUNT(title) DESC, title

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

SELECT name

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

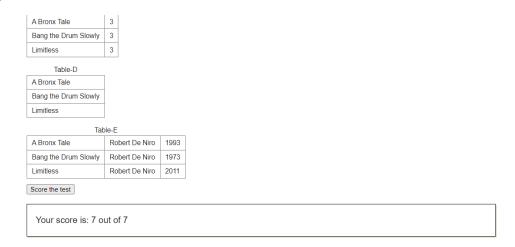
WHERE name != "Art Garfunkel" AND (movieid IN (

SELECT movieid

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

WHERE name = 'Art Garfunkel'))

JOIN QUIZ 2



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 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 dept = dept.id

5. Use COALESCE to print the mobile number. Use the number '07986 444 2266' if there is no number given. 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. Use the string 'None' where there is no department.

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

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

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

SELECT COUNT(name),**COUNT**(mobile) **FROM** teacher

8. Use COUNT and GROUP BY dept.name to show each department and the number of staff. Use a RIGHT JOIN to ensure that the Engineering department is listed.

SELECT dept.name, COUNT(teacher.name)

FROM teacher RIGHT JOIN dept ON (dept.id = 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 = 1 OR dept = 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 = 1 OR dept = 2 THEN "Sci"

WHEN dept = 3 THEN "Art"

ELSE "None" END

FROM teacher

NULL QUIZ



SELF JOIN

1. How many stops are in the database.

```
SELECT COUNT(name) 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 JOIN route ON id = stop

WHERE 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. If you are tired of these places try 'Fairmilehead' against 'Tollcross'

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

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

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

WHERE stopa.name = 'Craiglockhart' AND a.company= "LRT"
```

- 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.
 - No lo pudimos realizar debido a la dificultad del JOIN.

SELF JOIN QUIZ

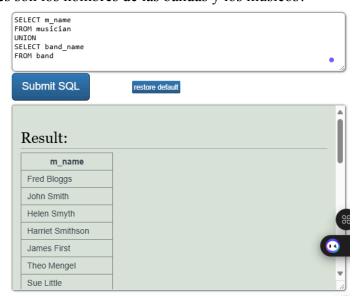
```
SELECT a.company, a.num, stopa.name, stopb.name
FROM route a JOIN route b ON (a.company=b.company AND a.num=b.num)
JOIN stops stopb ON (b.stop=stopb.id)
WHERE stopa.name='Tollcross'

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 stopz.name='Tollcross'

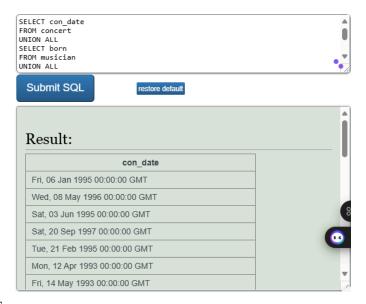
Score the test

Your score is: 3 out of 3
```

- C. Propongan preguntas que cumplan los siguientes requerimientos.
 - 5 consultas: una para cada operador de conjuntos
 - o UNION
 - ¿Cuáles son los nombres de las bandas y los músicos?

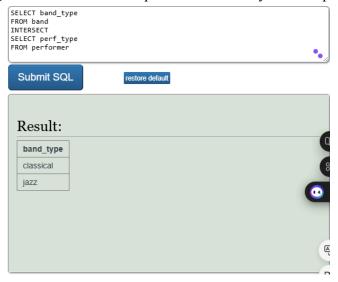


- UNION ALL
 - ¿Cuáles son las fechas relevantes en la música?



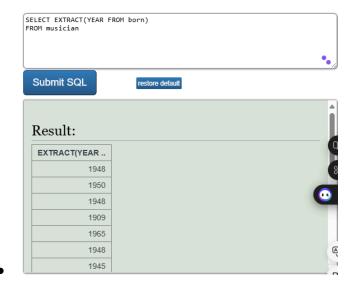
INTERSECT

■ ¿Qué géneros de música comparten las bandas y los intérpretes?



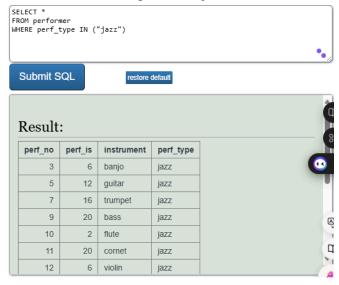
EXTRACT

■ ¿En qué años nacieron los músicos?



o IN

• ¿Intérpretes e instrumentos que hacen jazz?



6 consultas: una para cada operador de junta

- JOIN
 - o ¿En qué ciudad y país viven los músicos?

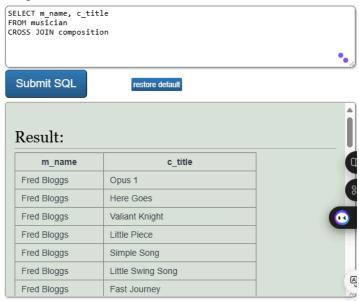


NATURAL JOIN

 No se puede hacer el ejercicio porque en la base de datos de musician's no contamos con 2 tabla que tengan el mismo nombre de columna para poder efectuar la búsqueda

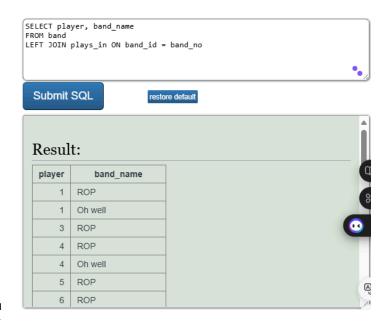
CROSS JOIN

o ¿Todos los posibles cantantes de cada canción?



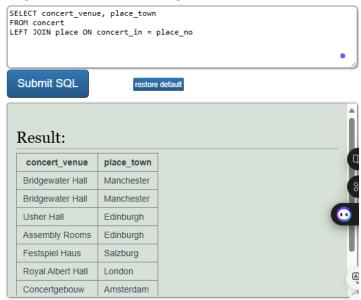
LEFT JOIN

o ¿A qué bandas pertenece cada músico?



RIGHT JOIN

o ¿En donde quedan los salones en los que se hicieron los conciertos?



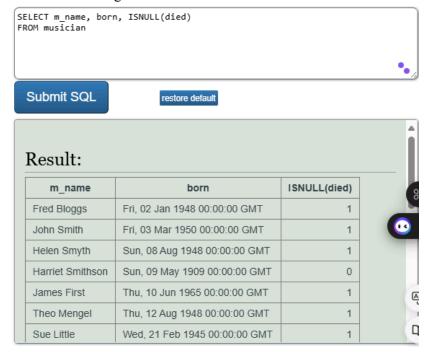
• FULL JOIN

¿Qué compositor compuso cada canción?

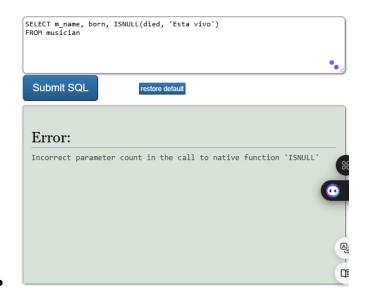


2 consultas: una para cada operador de desconocido

- ISNULL
 - o Dime cuando el músico sigue vivo

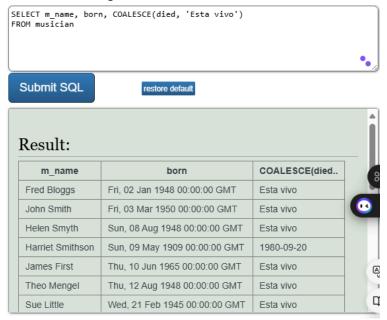


Tenemos un error al intentar usar el segundo parámetro salimos de los parámetros permitidos.



• COALESCE

Dime cuando el músico sigue vivo.



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

- EXISTS
 - o ¿Artistas que hacen parte de una banda?



ANY

• ¿Qué canciones tienen más reproducciones que el compositor con menos canciones?



• ALL

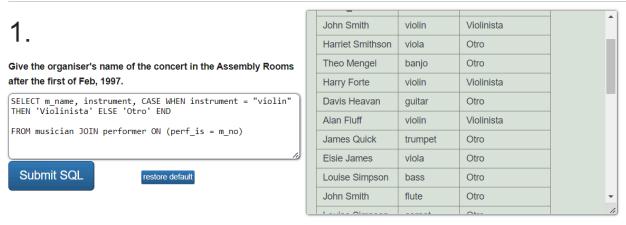
o ¿Qué músicos nacieron después del último músico fallecido?



• CASE:

Clasificar si un músico usa el violín, si no lo es, entonces llamarlo "Otro" :

Musicians: easy questions



REFERENCIAS:

https://www.w3schools.com/Sql/sql_join_left.asp

https://www.w3schools.com/sql/sql_null_values.asp

https://www.youtube.com/watch?v=jYYMMWlyFEI

https://www.freecodecamp.org/espanol/news/tutorial-de-uniones-en-sql/