**MODELOS Y BASES DE DATOS**

**SQL Básico**

**2019-02**

**Guia autoestudio 2/6**

# SQL- Detalle

* Dar nuevos nombres a tablas AS
* El valor NULL (DESCONOCIDO)
* Consultas que implican operaciones de conjuntos UNION, UNION ALL, INTERSECT, EXTRACT, IN
* Consultas que con junta explícita:
* Junta interna: de equivalencia, natural, cruzada JOIN, NATURAL JOIN, CROSS JOIN
* Junta externa: tabla izquierda, tabla derecha, completa LEFT JOIN, RIGTH JOIN, FULL JOIN
* **Operadores**
* Desconocido : ISNULL, COALESCE
* Lógicos : EXISTS, Comparación ANY, Comparación ALL,
* Condicionales: CASE

# INVESTIGACIÓN

# A. NULL

No olviden incluir la bibliografía.

1. ¿Qué significa?

**R/:** Se usa para indicar que no existe un valor dentro de una base de datos. Si un campo en una tabla es opcional, es posible insertar un nuevo registro o actualizar un registro sin agregar un valor a este campo. Luego, el campo se guardará con un valor NULL.

1. ¿Resultado de operarlo con los diferentes tipos de operadores:

aritméticos, lógicos y de comparación?

**R/:** El resultado de usar NULL con operadores aritméticos es un error, con operadores lógicos retorna un Booleano, y con los de comparación arroja error si son > o <, arrojan un Booleano en caso de que sea <> o =.

# B. JUNTA

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

**R/:** La junta interna es cuando se juntan las tablas A y B con los registros correspondientes, sólo cuando cumplen con una condición específica.

La junta externa no necesita que las tablas tengan datos relacionados, los registros se mantienen en la tabla combinada aunque no se tenga un correspondiente en una de las tablas.

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

**R/:** JOIN, NATURAL JOIN y CROSS JOIN.

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

**R/:** LEFT JOIN, RIGHT JOIN y FULL JOIN.

**PRACTICA**

# Usando SQLzoo.net [<http://sqlzoo.net/>]

**[En auto02.doc]**

**A.** Realicen los ejercicios propuestos en los siguientes tutoriales. (56 puntos) Utilice el motor My SQL 5. [Escriban la sentencia en SQL en auto02.doc y ejecuten la sentencia SQL en sqlzoo . Si no lograron escribir alguna sentencia indiquen el punto de problema]

1. [JOIN](http://sqlzoo.net/wiki/The_JOIN_operation)

In which we join two tables; game and goals. [previously music tutorial](http://sqlzoo.net/wiki/Music_Tutorial)

1.The first example shows the goal scored by a player with the last name 'Bender'. The \* says to list all the columns in the table - a shorter way of saying matchid, teamid, player, gtime

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

R/: SELECT matchid, player FROM goal

JOIN eteam ON ( id = teamid )

WHERE teamid = 'GER'

2.From the previous query you can see that Lars Bender's scored a goal in game 1012. Now we want to know what teams were playing in that match.

Notice in the that the column matchid in the goal table corresponds to the id column in the game table. We can look up information about game 1012 by finding that row in the **game** table.

**Show id, stadium, team1, team2 for just game 1012**

R/: SELECT DISTINCT(id),stadium,team1,team2 FROM game

JOIN goal ON ( id = matchid )

WHERE id = 1012

3.You can combine the two steps into a single query with a JOIN.

SELECT \*

FROM game JOIN goal ON (id=matchid)

The **FROM** clause says to merge data from the goal table with that from the game table. The **ON** says how to figure out which rows in **game** go with which rows in **goal** - the **matchid** from **goal** must match **id** from **game**. (If we wanted to be more clear/specific we could say

ON (game.id=goal.matchid)

The code below shows the player (from the goal) and stadium name (from the game table) for every goal scored.

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

R/: SELECT player,teamid,stadium, mdate FROM game

JOIN goal ON (id=matchid)

WHERE teamid LIKE 'GER'

4.Use the same JOIN as in the previous question.

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

R/: SELECT team1, team2, player FROM game

JOIN goal ON (id=matchid)

WHERE player LIKE 'Mario%'

5.The table eteam gives details of every national team including the coach. You can JOIN goal to eteam using the phrase goal JOIN eteam on teamid=id

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

R/: SELECT player, teamid, coach, gtime FROM goal

JOIN eteam ON ( teamid = id )

WHERE gtime<=10

6.To JOIN game with eteam you could use either

game JOIN eteam ON (team1=eteam.id) or game JOIN eteam ON (team2=eteam.id)

Notice that because id is a column name in both game and eteam you must specify eteam.id instead of just id

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

**R/:** SELECT mdate, teamname FROM game

JOIN eteam ON ( team1 = eteam.id )

WHERE coach LIKE 'Fernando Santos'

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

**R/:** SELECT player FROM game

JOIN goal ON (id=matchid)

WHERE stadium LIKE '%Warsaw'

8. The example query shows all goals scored in the Germany-Greece quarterfinal.

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

**R/:** 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.**

***COUNT and GROUP BY***

**R/:** SELECT teamname, COUNT(teamid) FROM goal

JOIN eteam ON (id = teamid)

GROUP BY teamname

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

**R/:** SELECT stadium, COUNT(teamid) FROM goal

JOIN game ON (id = matchid)

GROUP BY stadium

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

**R/:** SELECT matchid, mdate, COUNT( mdate ) FROM game

JOIN goal ON matchid = id

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

GROUP BY mdate, matchid

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

**R/:** SELECT matchid, mdate, COUNT( teamid ) FROM game

JOIN goal ON (id=matchid)

WHERE teamid LIKE 'GER'

GROUP BY matchid, mdate

13.

1. [More JOIN operations](http://sqlzoo.net/wiki/More_JOIN_operations)

In which we join actors to movies in the Movie Database.

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

R/: SELECT id, title

FROM movie

WHERE yr=1962

2.Give year of 'Citizen Kane'.

R/: SELECT yr FROM movie

WHERE title = 'Citizen Kane'

3. List all of the Star Trek movies, include 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%'

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 actor

JOIN casting ON movieid=11768 AND id = actorid

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

SELECT name FROM actor

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

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

SELECT title FROM movie a

JOIN casting ON movieid = a.id

JOIN actor b ON actorid=b.id

WHERE b.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 a

JOIN casting ON movieid = a.id

JOIN actor b ON actorid=b.id

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

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

SELECT title, name FROM movie a

JOIN casting ON movieid = a.id

JOIN actor b ON actorid=b.id

WHERE a.yr= 1962 AND ord = 1

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 (movieid=movie.id 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 30 starring roles.

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

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

1. [Using Null](http://sqlzoo.net/wiki/Using_Null)

In which we look at teachers in departments. [previously Scottish Parliament](http://sqlzoo.net/wiki/Scottish_Parliament)

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

SELECT name FROM teacher

WHERE dept IS NULL

1. 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)

1. 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

1. 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

1. 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

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

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

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

SELECT COUNT(name), COUNT(mobile)

FROM teacher

1. 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 (teacher.dept=dept.id)

GROUP BY dept.name

1. 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 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. 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 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 (teacher.dept=dept.id)

[8+ Numeric Examples](http://sqlzoo.net/wiki/NSS_Tutorial)

In which we look at a survey and deal with some more complex calculations.

The example shows the number who responded for:

* question 1
* at 'Edinburgh Napier University'
* studying '(8) Computer Science'

**Show the the percentage who STRONGLY AGREE**

1. SELECT A\_STRONGLY\_AGREE

FROM nss

WHERE question='Q01'

AND institution='Edinburgh Napier University'

AND subject='(8) Computer Science'

1. **Show the institution and subject where the score is at least 100 for question 15.**

**R/:** SELECT institution, subject FROM nss

WHERE question = 'Q15' AND score >= 100

1. **Show the institution and score where the score for '(8) Computer Science' is less than 50 for question 'Q15'**

**R/:** SELECT institution, score FROM nss

WHERE question='Q15'

AND subject='(8) Computer Science'

AND score < 50

1. **Show the subject and total number of students who responded to question 22 for each of the subjects '(8) Computer Science' and '(H) Creative Arts and Design'.**

**R/:** SELECT subject, SUM(response) FROM nss

WHERE question='Q22'

AND subject IN ('(H) Creative Arts and Design','(8) Computer Science')

GROUP BY subject

1. **Show the subject and total number of students who A\_STRONGLY\_AGREE to question 22 for each of the subjects '(8) Computer Science' and '(H) Creative Arts and Design'.**

**R/:** SELECT subject, SUM(response\*A\_STRONGLY\_AGREE/100)FROM nss

WHERE question='Q22'

AND subject IN ('(8) Computer Science','(H) Creative Arts and Design')

GROUP BY subject

1. **Show the percentage of students who A\_STRONGLY\_AGREE to question 22 for the subject '(8) Computer Science' show the same figure for the subject '(H) Creative Arts and Design'.**

**R/:** SELECT subject, ROUND(SUM(A\_STRONGLY\_AGREE\*response/100)/SUM(response)\*100) FROM nss

WHERE question='Q22'

AND subject IN ('(8) Computer Science','(H) Creative Arts and Design')

GROUP BY subject

1. **Show the average scores for question 'Q22' for each institution that include 'Manchester' in the name.**

The column **score** is a percentage - you must use the method outlined above to multiply the percentage by the **response** and divide by the total response. Give your answer rounded to the nearest whole number.

**R/:** SELECT institution,ROUND(SUM(score\*response/100)/SUM(response)\*100) FROM nss

WHERE question='Q22'

AND (institution LIKE '%Manchester%')

GROUP BY institution

1. [Self join](http://sqlzoo.net/wiki/Self_join)
2. How many **stops** are in the database.

SELECT COUNT(id) FROM stops

1. Find the **id** value for the stop 'Craiglockhart'

SELECT id FROM stops

WHERE name= 'Craiglockhart'

1. 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 company='LRT' AND num=4

1. 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(\*) AS a

FROM route WHERE stop=149 OR stop=53

GROUP BY company, num

HAVING a=2

1. 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

1. 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'

1. 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)

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. 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'

1. 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 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'

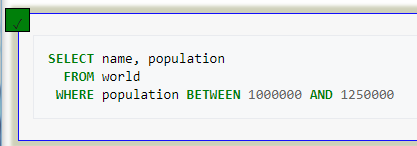
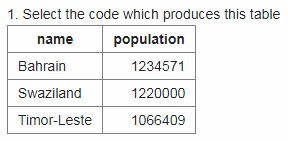
1. No pudimos

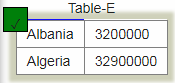
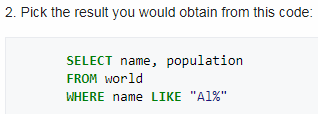
**B.**  Realicen los ejercicios propuestos en el siguiente quiz (59 puntos)

# Tutorials: Learn SQL in stages

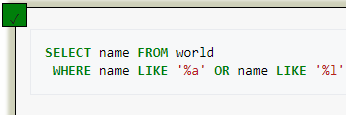
[10 Tutorial Quizzes](http://sqlzoo.net/wiki/Tutorial_Quizzes)

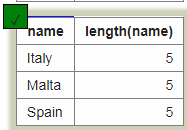
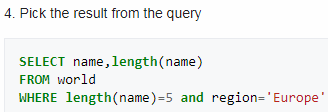
Test your knowledge with multiple choice quizzes

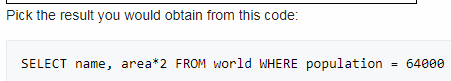




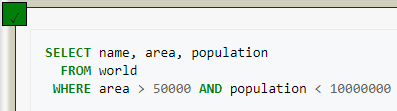




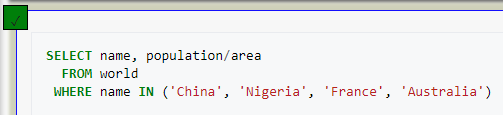




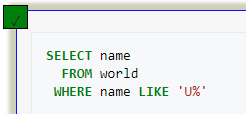




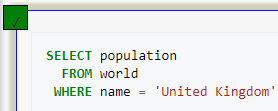










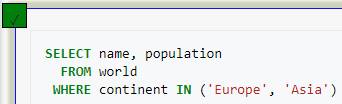




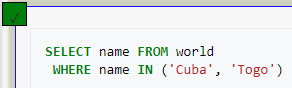


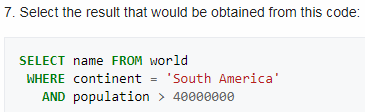


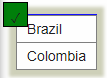




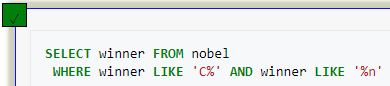




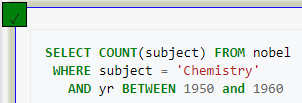




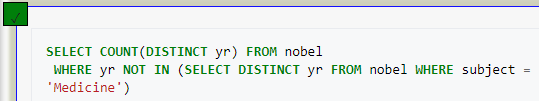


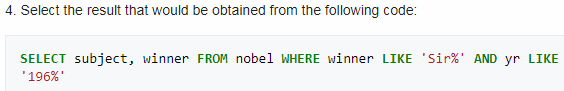


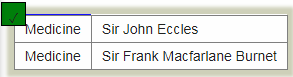




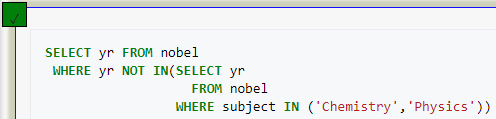




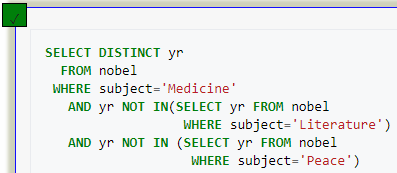


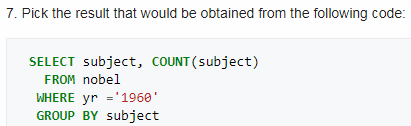


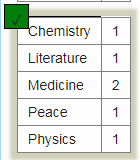




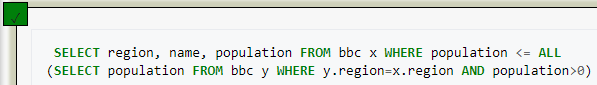




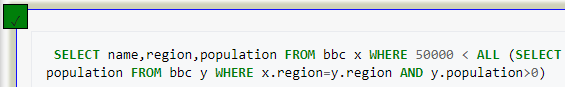




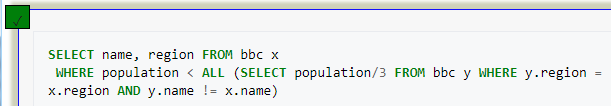








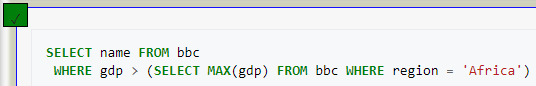




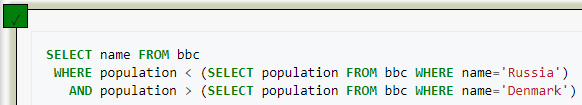




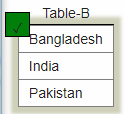




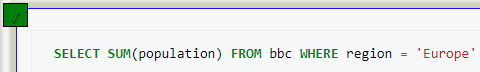


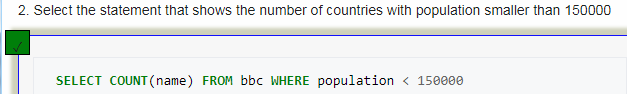






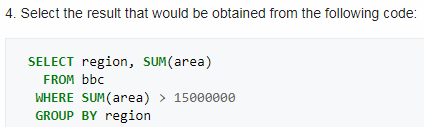






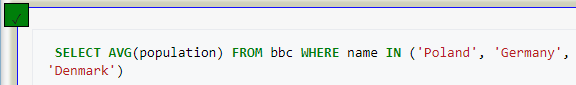




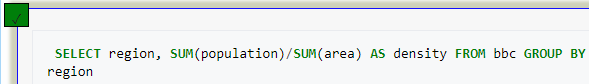




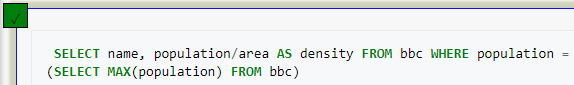




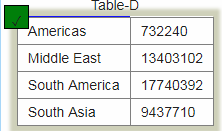




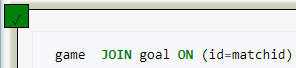




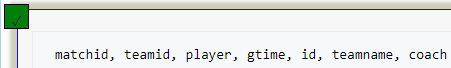


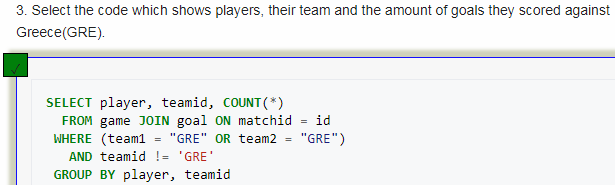


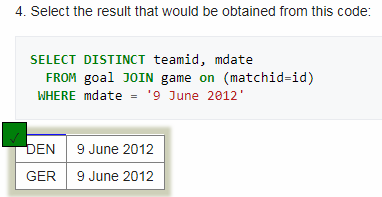




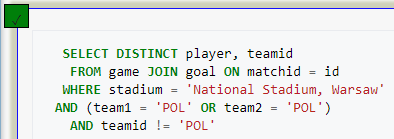




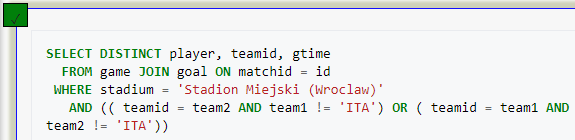


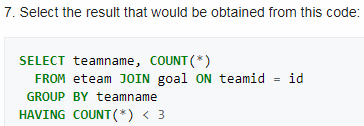






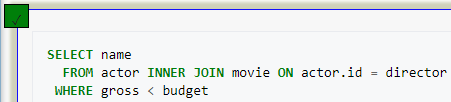




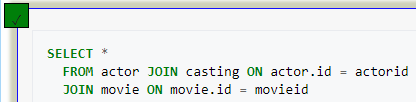




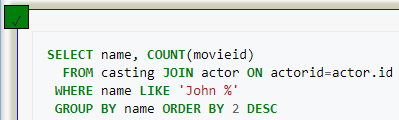


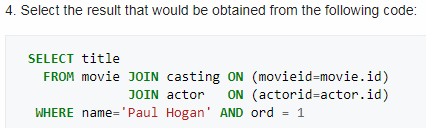


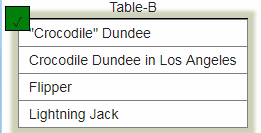




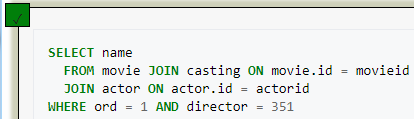




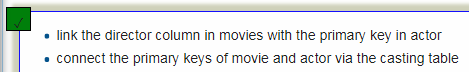


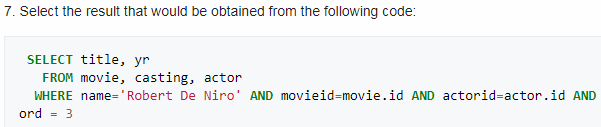


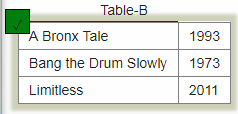




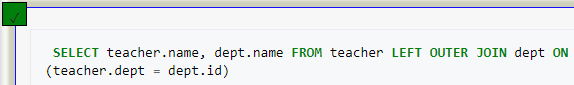




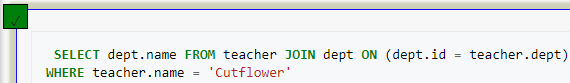




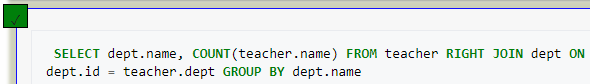


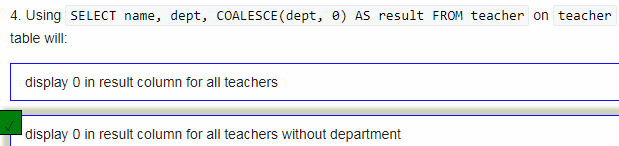


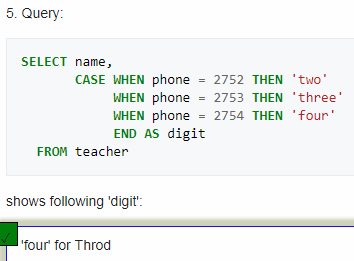


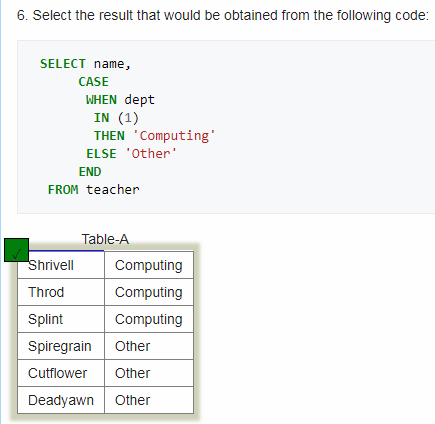




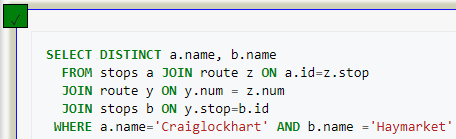




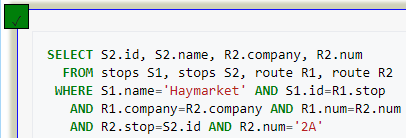




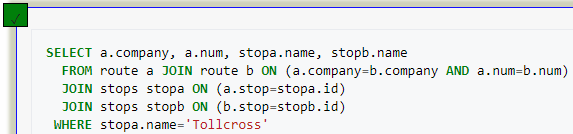












**C.** Propongan preguntas que cumplan los siguientes requerimientos. (15 puntos)

Usen la base de datos **musicians**

Escoja el motor que prefiera. Justifique la elección.

[Escriban la consulta en lenguaje natural y la sentencia en SQL en auto02..doc y ejecuten la sentencia SQL en sqlzoo . Si no lograron escribir alguna sentencia indiquen el punto de problema]

* 5 consultas: una para cada operador de conjuntos

1. ¿Qué artistas siguen viviendo en donde nacieron?

SELECT m\_name, place\_country FROM musician

INNER JOIN place ON (born\_in = living\_in AND place\_no = born\_in)

1. ¿Cuántos conciertos se han hecho en cada lugar?

SELECT place\_town, COUNT( concert\_in ) FROM place

LEFT JOIN concert ON ( concert\_in = place\_no )

GROUP BY place\_town

1. ¿Cuántas bandas han salido de cuál barrio?

SELECT place\_town, COUNT( band\_home) FROM place

RIGHT JOIN band ON ( band\_home = place\_no)

GROUP BY place\_town

1. ¿Cuántos artistas componen cierto tipo de musica?

SELECT perf\_type, COUNT(m\_name) FROM performer

FULL JOIN musician ON ( perf\_is = m\_no )

GROUP BY perf\_type

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

1. Cuantos musicos hay por cada pais

SELECT place\_country, COUNT(m\_no) AS Cantidad FROM musician

JOIN place ON (place\_no = born\_in)

GROUP BY place\_country

1. Cuales musicos tocan el violin

SELECT m\_name FROM musician

NATURAL JOIN performer

WHERE perf\_is=m\_no AND instrument = 'violin'

ORDER BY m\_name

1. De el nombre y los instrumentos que toca cada musico.

SELECT m\_name, instrument FROM musician

left JOIN performer ON ( perf\_is=m\_no )

WHERE instrument IS NOT NULL

ORDER BY m\_name

1. Cuantos musicos hay en cada banda

SELECT band\_name, COUNT(band\_name) FROM band

FULL JOIN plays\_in ON ( band\_no=band\_id )

GROUP BY band\_name

* 2 consultas: una para cada operador de desconocido

1. Muestre el nombre de los musicos que no tienen un tipo de performance definido.

SELECT m\_name FROM musician

JOIN performer ON ( perf\_is=m\_no )

WHERE perf\_type IS NULL

1. Mostrar el nombre y la fecha de muerte de los musicos, si aun esta vivo ponga ’Vive’

SELECT m\_name, COALESCE(died, 'Vive') AS Muerte

FROM musician

**WEBGRAFÍA:**

* <https://es.wikipedia.org/wiki/Null_(SQL)>
* <https://www.w3schools.com/sql/sql_null_values.asp>