

AUTOESTUDIO 14/02/19

AUTORES

Laura Izquierdo, Yarit Villalobos

INVESTIGACIÓN

A. NULL

- ¿Qué significa?

Es un marcador especial utilizado en el lenguaje de consulta estructurado para indicar que no existe un valor de un dato en la base de datos.

- ¿Resultado de operarlo con los diferentes tipos de operadores: aritméticos, lógicos y de comparación?

Operadores aritméticos

Como Null no es un dato, sino un marcador para un valor ausente, usando *operadores matemáticos* Null da un resultado desconocido, que está representado por Null.

En el ejemplo siguiente, multiplicando 10 por resultados Null en Null:

```
10 * NULL          -- Result is NULL
```

Operadores lógicos y de comparación

Ya que Null no es un miembro de un dominio de datos, no se considera un «valor», sino más bien un marcador (o marcador de posición) indicando la ausencia de valor. Debido a esto, las comparaciones con Null nunca resulta verdadero o falso, pero siempre en un *tercer resultado lógico*, desconocido. El resultado lógico de la expresión siguiente, que compara el valor 10 a Null, es desconocido:

```
SELECT 10 = NULL      -- Results in Unknown
```

Sin embargo, ciertas operaciones en Null pueden devolver valores si el valor ausente no es relevante para el resultado de la operación. Considere el siguiente ejemplo:

```
SELECT NULL OR TRUE   -- Results in True
```

En este caso, el hecho de que el valor de la izquierda o es incognoscible es irrelevante, porque el resultado de la operación sería verdadera independientemente del valor de la izquierda.

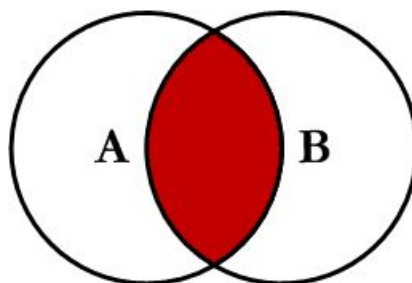
Nota:

En lógica, una **lógica de tres valores** (también **lógica trinario**, **trivalente**, **ternario**, o **trilean**, a veces es abreviado **3VL**) es cualquiera de los muchos sistemas de valores lógicos donde existen tres valores de verdad indicando *verdadero*, *falso* y algún valor tercer indeterminado

B. JUNTA

- ¿Cuáles son las diferencias entre junta interna y externa?

Junta interna



Esta es la Unión más simple, más entendida y es la más común. Esta consulta devolverá todos los registros coincidentes entre dos tablas A Y B. Esta Unión se escribe como sigue:

SQL

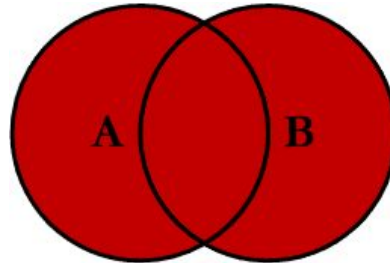
```
SELECT <select_list>
```

```
FROM Table_A A
```

```
INNER JOIN Table_B B
```

```
ON A.Key = B.Key
```

Junta externa



Esta Unión también se puede referir como una Unión exterior completa o una Unión completa. Esta consulta devolverá todos los registros de ambas tablas, uniendo los registros de la tabla izquierda (tabla A) que coinciden con los registros de la tabla derecha (tabla B). Esta Unión se escribe como sigue:

SQL

```
SELECT <select_list>
```

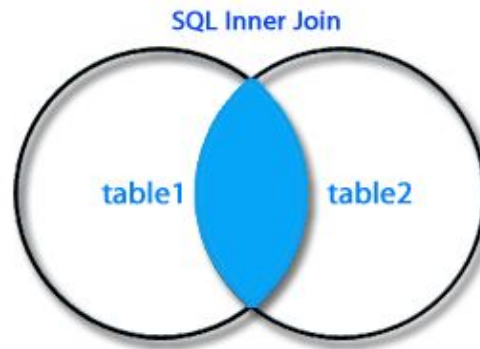
```
FROM Table_A A
```

```
FULL OUTER JOIN Table_B B
```

```
ON A.Key = B.Key
```

- **¿ Qué opciones se tienen para la junta interna ?**

INNER JOIN O JOIN: Devuelve **todas las filas** cuando hay al menos **una coincidencia** en **ambas** tablas.

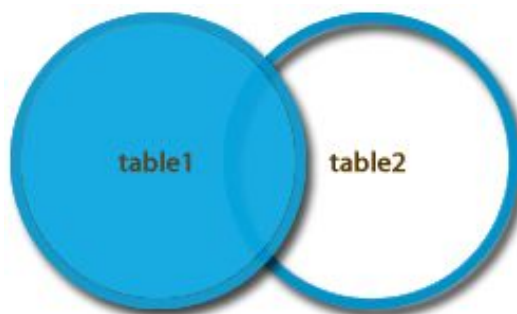


JOIN NATURAL: Es un tipo de JOIN estructurado de tal manera que, columnas con el mismo nombre de tablas asociadas aparecen sólo una vez. Se diferencia del INNER JOIN por el número de columnas que devuelve.

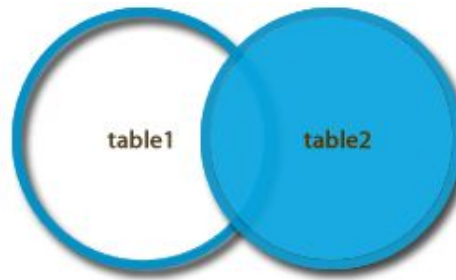
CROSS JOIN: devuelve el *producto cartesiano* de filas de tablas en el join. En otras palabras, va a producir filas que combinan cada fila de la primera tabla con cada fila de la segunda tabla.

- ¿ Qué opciones se tienen para la junta externa?

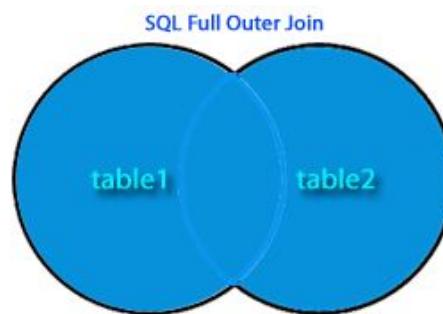
LEFT JOIN: Devuelve todas las filas de la tabla de la **izquierda**, y las filas coincidentes de la tabla de la **derecha**.



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



FULL JOIN: Devuelve **todas las filas** de las dos tablas, la izquierda y la derecha. También se llama FULL OUTER JOIN.



BIBLIOGRAFÍA

Anónimo. Null(SQL). Obtenido de: [https://en.wikipedia.org/wiki/Null_\(SQL\)](https://en.wikipedia.org/wiki/Null_(SQL))

Anónimo. Three-valued logic. Obtenido de: https://en.wikipedia.org/wiki/Three-valued_logic

C.L. Moffatt,2009.Visual Representation of SQL Joins. Obtenido de: <https://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins>

Lázaro,2018. Principales tipos de JOINS en SQL. Obtenido de: <https://diego.com.es/principales-tipos-de-joins-en-sql>

Stackoverflow. CRUZADAS VS INNER JOIN. Obtenido de: <https://stackoverflow.com/questions/17759687/cross-join-vs-inner-join-in-sql-server-2008>

PRÁCTICA

A. Tutorials: Learn SQL in stages - CONSULTAS

6 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 DISTINCT id,stadium,team1,team2  
  
FROM game ga  
  
JOIN goal go  
  
ON ga.id=go.matchid and 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 id=teamid  
  
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 (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(teamid)  
  
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(matchid)
```

```
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(mdate)
```

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

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

```
GROUP BY matchid,mdate
```

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

```
SELECT matchid,mdate,COUNT(teamid)
```

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

```
WHERE (teamid='GER')
```

```
GROUP BY matchid,mdate
```

- 13. Sort your result by mdate, matchid, team1 and team2.**

```
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 mdate,team1,team2
```

```
ORDER BY mdate, matchid, team1 and team2
```

7 MORE JOIN OPERATIONS

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

```
SELECT id, title
```



```
FROM movie
```

```
WHERE yr=1962
```

2. Give year of 'Citizen Kane'.

```
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' *what is a cast list?*, Use movieid=11768

```
SELECT name FROM actor JOIN (casting JOIN movie
```

```
ON casting.movieid=movie.id) ON casting.actorid=actor.id
```

```
WHERE title='Casablanca'
```

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

```

SELECT name

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

WHERE title='Alien'

```

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

```

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

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

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

```

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

```

SELECT title, name

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

WHERE yr='1962' AND ord='1'

```

11. Which were the busiest years for 'John Travolta', 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='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.List the film title and the leading actor for all of the films 'Julie Andrews' played in.

```

SELECT title, name

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

(movieid=movie.id AND ord=1)

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.

```

SELECT name

FROM casting JOIN actor ON actorid=id

WHERE ord=1

GROUP BY name

HAVING count(name)>=30

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(actorid) AS actors FROM movie

```

```
JOIN casting ON id = movieid
```

```
WHERE yr = 1978
```

```
GROUP BY title
```

```
ORDER BY actors DESC,title
```

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

```
SELECT DISTINCT name
```

```
FROM (SELECT movieid FROM casting JOIN actor ON actor.id=actorid
```

```
WHERE name='Art Garfunkel') tab
```

```
JOIN casting ON casting.movieid=tab.movieid
```

```
JOIN actor ON id=actorid
```

```
WHERE name!='Art Garfunkel'
```

8 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 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 teacher.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 teacher.dept=dept.id  
  
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 AS dep  
  
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
```

8+ NUMERIC EXAMPLES

- 1. Show the the percentage who STRONGLY AGREE**

```
SELECT A_STRONGLY_AGREE  
  
FROM nss  
  
WHERE question='Q01'  
  
AND institution='Edinburgh Napier University'  
  
AND subject='(8) Computer Science'
```

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

```
SELECT institution, subject  
  
FROM nss  
  
WHERE score >= 100
```

AND question='Q15'

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

SELECT institution,score

FROM nss

WHERE question='Q15'

AND subject='(8) Computer Science'

AND score<50

- 4. 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'.**

SELECT subject, SUM(response)

FROM nss

WHERE question='Q22'

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

GROUP BY subject

- 5. 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'.**

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

- 6. 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'.**

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

FROM nss

WHERE question='Q22'

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

GROUP BY subject
```

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

```
SELECT institution, ROUND(SUM(response*score)/SUM(response),0)
AS scores

FROM nss

WHERE question='Q22'

AND (institution LIKE '%Manchester%')

GROUP BY institution
```

- 8. Show the institution, the total sample size and the number of computing students for institutions in Manchester for 'Q01'.**

```
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
```

9 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 JOIN route ON id=stop  
  
WHERE 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= (SELECT DISTINCT stop FROM stops
JOIN route ON id=stop WHERE name='London Road')
```

- 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,b.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) JOIN (stops sa JOIN stops sb)
ON a.company=b.company AND a.num=b.num AND a.stop=sa.id AND
b.stop=sb.id
WHERE sa.name='Craiglockhart' AND sb.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 name,b.company,b.num
```

```
FROM route a JOIN route b ON a.num=b.num AND  
a.company=b.company
```

```
JOIN stops ON id=b.stop
```

```
WHERE a.stop= (SELECT DISTINCT stop FROM stops JOIN route ON  
id=stop WHERE name='Craiglockhart')
```

Nota: No se pudo realizar el 10

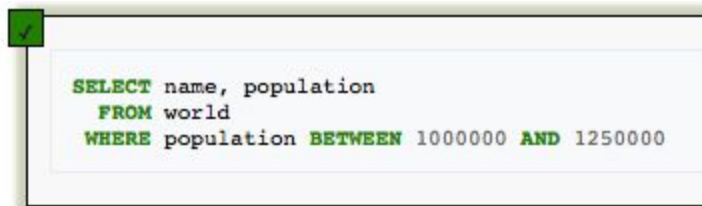
B. Tutorials: Learn SQL in stages - QUICES

10 TUTORIAL QUIZZES

1.SELECT

1. Select the code which produces this table

name	population
Bahrain	1234571
Swaziland	1220000
Timor-Leste	1066409



2. Pick the result you would obtain from this code:

```
SELECT name, population  
FROM world  
WHERE name LIKE "A%"
```

Table-E

Albania	3200000
Algeria	32900000

3. Select the code which shows the countries that end in A or L

```
SELECT name FROM world
WHERE name LIKE '%a' OR name LIKE '%l'
```

4. Pick the result from the query

```
SELECT name, length(name)
FROM world
WHERE length(name)=5 and region='Europe'
```

name	length(name)
Italy	5
Malta	5
Spain	5

5. Here are the first few rows of the world table:


name	region	area	population	gdp
Afghanistan	South Asia	652225	26000000	
Albania	Europe	28728	3200000	6656000000
Algeria	Middle East	2400000	32900000	75012000000
Andorra	Europe	468	64000	
...				

Pick the result you would obtain from this code:

```
SELECT name, area*2 FROM world WHERE population = 64000
```


Andorra	936
---------	-----

6. Select the code that would show the countries with an area larger than 50000 and a population smaller than 10000000



```
SELECT name, area, population
FROM world
WHERE area > 50000 AND population < 10000000
```


7. Select the code that shows the population density of China, Australia, Nigeria and France



```
SELECT name, population/area
FROM world
WHERE name IN ('China', 'Nigeria', 'France', 'Australia')
```


2. BBC

1. Select the code which gives the name of countries beginning with U



```
SELECT name
FROM world
WHERE name LIKE 'U%'
```

2. Select the code which shows just the population of United Kingdom?



```
SELECT population
FROM world
WHERE name = 'United Kingdom'
```

3. Select the answer which shows the problem with this SQL code - the intended result should be the continent of France:

```
SELECT continent
FROM world
WHERE 'name' = 'France'
```

☒ 'name' should be name

4. Select the result that would be obtained from the following code:

```
SELECT name, population / 10
FROM world
WHERE population < 10000
```

☒

Nauru	990
-------	-----

5. Select the code which would reveal the name and population of countries in Europe and Asia

☒

```
SELECT name, population
FROM world
WHERE continent IN ('Europe', 'Asia')
```

6. Select the code which would give two rows

☒

```
SELECT name FROM world
WHERE name IN ('Cuba', 'Togo')
```

7. Select the result that would be obtained from this code:

```
SELECT name FROM world
WHERE continent = 'South America'
AND population > 40000000
```

<input checked="" type="checkbox"/>	Brazil
<input type="checkbox"/>	Colombia

3. NOBEL

1. Pick the code which shows the name of winner's names beginning with C and ending in n

☒

```

SELECT winner FROM nobel
WHERE winner LIKE 'C%' AND winner LIKE '%n'

```

2. Select the code that shows how many Chemistry awards were given between 1950 and 1960

☒

```

SELECT COUNT(subject) FROM nobel
WHERE subject = 'Chemistry'
AND yr BETWEEN 1950 and 1960

```

3. Pick the code that shows the amount of years where no Medicine awards were given

☒

```

SELECT COUNT(DISTINCT yr) FROM nobel
WHERE yr NOT IN (SELECT DISTINCT yr FROM nobel WHERE subject = 'Medicine')

```

4. Select the result that would be obtained from the following code:

```

SELECT subject, winner FROM nobel WHERE winner LIKE 'Sir%' AND yr LIKE '196%'

```

<input checked="" type="checkbox"/>	Medicine	Sir John Eccles
<input type="checkbox"/>	Medicine	Sir Frank Macfarlane Burnet

5. Select the code which would show the year when neither a Physics or Chemistry award was given

```
SELECT yr FROM nobel
WHERE yr NOT IN(SELECT yr
                  FROM nobel
                  WHERE subject IN ('Chemistry','Physics'))
```

6. Select the code which shows the years when a Medicine award was given but no Peace or Literature award was

```
SELECT DISTINCT yr
FROM nobel
WHERE subject='Medicine'
AND yr NOT IN(SELECT yr FROM nobel
               WHERE subject='Literature')
AND yr NOT IN (SELECT yr FROM nobel
               WHERE subject='Peace')
```

7. Pick the result that would be obtained from the following code:

```
SELECT subject, COUNT(subject)
FROM nobel
WHERE yr = '1960'
GROUP BY subject
```

Chemistry	1
Literature	1
Medicine	2
Peace	1
Physics	1

4.NESTED SELECT

1. Select the code that shows the name, region and population of the smallest country in each region

`SELECT region, name, population FROM bbc x WHERE population <= ALL (SELECT population FROM bbc y WHERE y.region=x.region AND population>0)`

2. Select the code that shows the countries belonging to regions with all populations over 50000

`SELECT name,region,population FROM bbc x WHERE 50000 < ALL (SELECT population FROM bbc y WHERE x.region=y.region AND y.population>0)`

3. Select the code that shows the countries with a less than a third of the population of the countries around it

`SELECT name, region FROM bbc x
WHERE population < ALL (SELECT population/3 FROM bbc y WHERE y.region = x.region AND y.name != x.name)`

4. Select the result that would be obtained from the following code:

```
SELECT name FROM bbc  
WHERE population >  
      (SELECT population  
        FROM bbc  
        WHERE name='United Kingdom')  
AND region IN  
      (SELECT region  
        FROM bbc  
        WHERE name = 'United Kingdom')
```

Table-D

France
Germany
Russia
Turkey

5. Select the code that would show the countries with a greater GDP than any country in Africa (some countries may have NULL gdp values).

`SELECT name FROM bbc
WHERE gdp > (SELECT MAX(gdp) FROM bbc WHERE region = 'Africa')`

6. Select the code that shows the countries with population smaller than Russia but bigger than Denmark



```
SELECT name FROM bbc
WHERE population < (SELECT population FROM bbc WHERE name='Russia')
AND population > (SELECT population FROM bbc WHERE name='Denmark')
```

7. >Select the result that would be obtained from the following code:

```
SELECT name FROM bbc
WHERE population > ALL
  (SELECT MAX(population)
   FROM bbc
   WHERE region = 'Europe')
AND region = 'South Asia'
```



Table-B

Bangladesh
India
Pakistan

5. SUM AND COUNT

1. Select the statement that shows the sum of population of all countries in 'Europe'

```
SELECT SUM(population) FROM bbc WHERE region = 'Europe'
```

2. Select the statement that shows the number of countries with population smaller than 150000

```
SELECT COUNT(name) FROM bbc WHERE population < 150000
```

3. Select the list of core SQL aggregate functions

```
AVG(), COUNT(), MAX(), MIN(), SUM()
```

4. Select the result that would be obtained from the following code:

No result due to invalid use of the WHERE function

5. Select the statement that shows the average population of 'Poland', 'Germany' and 'Denmark'

```
SELECT AVG(population) FROM bbc WHERE name IN ('Poland', 'Germany', 'Denmark')
```

6. Select the statement that shows the medium population density of each region

```
SELECT region, SUM(population)/SUM(area) AS density FROM bbc GROUP BY region
```

7. Select the statement that shows the name and population density of the country with the largest population

```
SELECT name, population/area AS density FROM bbc WHERE population = (SELECT MAX(population) FROM bbc)
```

8. Pick the result that would be obtained from the following code:

Table-D	
Americas	732240
Middle East	13403102
South America	17740392
South Asia	9437710

6. JOIN QUIZ

1. You want to find the stadium where player 'Dimitris Salpingidis' scored. Select the JOIN condition to use:

```
game JOIN goal ON (id=matchid)
```

2. You JOIN the tables goal and eteam in an SQL statement. Indicate the list of column names that may be used in the SELECT line:

```
matchid, teamid, player, gtime, id, teamname, coach
```

3. Select the code which shows players, their team and the amount of goals they scored against Greece(GRE).

```
SELECT player, teamid, COUNT(*)  
FROM game JOIN goal ON matchid = id  
WHERE (team1 = "GRE" OR team2 = "GRE")  
AND teamid != 'GRE'  
GROUP BY player, teamid
```

4. Select the result that would be obtained from this code:

DEN	9 June 2012
GER	9 June 2012

5. Select the code which would show the player and their team for those who have scored against Poland(POL) in National Stadium, Warsaw.

```

SELECT DISTINCT player, teamid
  FROM game JOIN goal ON matchid = id
 WHERE stadium = 'National Stadium, Warsaw'
 AND (team1 = 'POL' OR team2 = 'POL')
 AND teamid != 'POL'

```

6. Select the code which shows the player, their team and the time they scored, for players who have played in Stadion Miejski (Wroclaw) but not against Italy(ITA).

```

SELECT DISTINCT player, teamid, gtime
  FROM game JOIN goal ON matchid = id
 WHERE stadium = 'Stadion Miejski (Wroclaw)'
 AND (( teamid = team2 AND team1 != 'ITA') OR ( teamid = team1 AND team2
 != 'ITA'))

```

7. Select the result that would be obtained from this code:

Netherlands	2
Poland	2
Republic of Ireland	1
Ukraine	2

7.JOIN QUIZ2

1. Select the statement which lists the unfortunate directors of the movies which have caused financial loses (gross < budget)

```

SELECT name
  FROM actor INNER JOIN movie ON actor.id = director
 WHERE gross < budget

```

2. Select the correct example of JOINing three tables

```
SELECT *  
FROM actor JOIN casting ON actor.id = actorid  
JOIN movie ON movie.id = movieid
```

3. Select the statement that shows the list of actors called 'John' by order of number of movies in which they acted

```
SELECT name, COUNT(movieid)  
FROM casting JOIN actor ON actorid=actor.id  
WHERE name LIKE 'John %'  
GROUP BY name ORDER BY 2 DESC
```

4. Select the result that would be obtained from the following code:

"Crocodile" Dundee
Crocodile Dundee in Los Angeles
Flipper
Lightning Jack

5. Select the statement that lists all the actors that starred in movies directed by Ridley Scott who has id 351

```
SELECT name  
FROM movie JOIN casting ON movie.id = movieid  
JOIN actor ON actor.id = actorid  
WHERE ord = 1 AND director = 351
```

6. There are two sensible ways to connect movie and actor. They are:



- link the director column in movies with the primary key in actor
- connect the primary keys of movie and actor via the casting table

7. Select the result that would be obtained from the following code:



A Bronx Tale	1993
Bang the Drum Slowly	1973
Limitless	2011

8.USING NULL QUIZZ

1. Select the code which uses an outer join correctly.



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

2. Select the correct statement that shows the name of department which employs Cutflower -



```
SELECT dept.name FROM teacher JOIN dept ON (dept.id = teacher.dept) WHERE  
teacher.name = 'Cutflower'
```

3. Select out of following the code which uses a JOIN to show a list of all the departments and number of employed teachers


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

4. Using `SELECT name, dept, COALESCE(dept, 0) AS result FROM teacher` on `teachertable` will:

display 0 in result column for all teachers without department

5. Query:

```
SELECT name,
       CASE WHEN phone = 2752 THEN 'two'
            WHEN phone = 2753 THEN 'three'
            WHEN phone = 2754 THEN 'four'
            END AS digit
FROM teacher
```

'four' for Throd

6. Select the result that would be obtained from the following code:

```
SELECT name,
       CASE
         WHEN dept
           IN (1)
         THEN 'Computing'
         ELSE 'Other'
       END
FROM teacher
```


table-A

Shrivell	Computing
Throd	Computing
Splint	Computing
Spiregrain	Other
Cutflower	Other
Deadyawn	Other

9.SELF JOIN

1. Select the code that would show it is possible to get from Craiglockhart to Haymarket

```
SELECT DISTINCT a.name, b.name
FROM stops a JOIN route z ON a.id=z.stop
JOIN route y ON y.num = z.num
JOIN stops b ON y.stop=b.id
WHERE a.name='Craiglockhart' AND b.name = 'Haymarket'
```

2. Select the code that shows the stops that are on route.num '2A' which can be reached with one bus from Haymarket?

```
SELECT S2.id, S2.name, R2.company, R2.num
FROM stops S1, stops S2, route R1, route R2
WHERE S1.name='Haymarket' AND S1.id=R1.stop
AND R1.company=R2.company AND R1.num=R2.num
AND R2.stop=S2.id AND R2.num='2A'
```

3. Select the code that shows the services available from 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='Tollcross'

```

C. Consultas Propuestas - Usando la base de datos *Guest House*

- 4 consultas: una para cada operador de conjuntos

UNION

Muestre quien y en que fechas hicieron reserva para el cuarto 102 y 103.

```

SELECT first_name,last_name,booking_date,room_no
FROM booking JOIN guest ON guest_id = guest.id
WHERE room_no=102

```

UNION

```

SELECT first_name,last_name,booking_date,room_no
FROM booking JOIN guest ON guest_id = guest.id
WHERE room_no=103

```

UNION ALL

Muestre el primer nombre de los huéspedes con sus respectivos id que llegaron el '2016-11-03'

```

SELECT first_name
FROM guest JOIN booking ON (guest.id=booking.guest_id)
WHERE booking_date='2016-11-03'

```

UNION ALL

```
SELECT guest_id AS first_name  
  
FROM booking  
  
WHERE booking_date='2016-11-03'
```

INTERSECT

Muestre los huéspedes que reservaron un cuarto doble y reservaron el '2016-11-03'

```
SELECT first_name  
  
FROM guest JOIN booking ON (guest.id=booking.guest_id)  
  
WHERE room_type_requested='double'  
  
INTERSECT
```

```
SELECT first_name  
  
FROM guest JOIN booking ON (guest.id=booking.guest_id)  
  
WHERE booking_date='2016-11-03'
```

Faltó EXTRACT

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

JOIN

Mencione el nombre y el apellido de la persona que se hospeda en la habitación 101, el 3 de Noviembre del 2016.

```
SELECT first_name,last_name  
  
FROM guest JOIN booking ON guest_id=id  
  
WHERE room_no=101 AND booking_date='2016-11-03'
```

CROSS JOIN

Menciona Todas las personas que compartan el apellido y diferentes nombres, y muestra el apellido con los nombres.

```

SELECT a.last_name,a.first_name,c.first_name

FROM guest c CROSS JOIN guest a

WHERE a.first_name!=c.first_name AND a.last_name=c.last_name

```

LEFT JOIN

Mencione los cuartos que se ocuparon ese día '2016-11-08', junto las descripciones de los extra que pidió cada cuarto.

```

SELECT room_no,description

FROM booking LEFT JOIN extra ON

booking.booking_id=extra.booking_id

WHERE booking_date ='2016-11-08'

```

Faltó RIGHT JOIN

- **2 consultas: una para cada operador de desconocido**

COALESCE

Mencione los cuartos que llegaron el primero de diciembre del 2016 y cuánto se gastó cada uno en extra.

```

SELECT room_no,COALESCE(amount,0)

FROM booking LEFT JOIN extra ON

booking.booking_id=extra.booking_id

WHERE booking_date ='20161201'

```

Faltó IS NULL

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

EXISTS

Muestre todos los cuartos, si existe algún cuarto que se ocupe el 12 de diciembre del 2016 y que lleve entre las 14:00 y 15:00

```

SELECT DISTINCT room_no

```

```
FROM booking
```

```
WHERE EXISTS
```

```
(SELECT arrival_time FROM booking WHERE arrival_time BETWEEN  
'14:00' AND '15:00')
```

```
AND booking_date='20161212')
```

Faltó ANY, ALL

- **1 consulta: para el operador CASE**

Menciona todas las personas que lleguen el 2016-12-10 y valorarlo como especial si se queda más de 3 noches y como no especial si no.

```
SELECT first_name,
```

```
CASE WHEN nights>3 THEN 'Frecuente' else 'No frecuente' END as  
valorado
```

```
FROM booking JOIN guest ON id=guest_id
```

```
WHERE booking_date='20161210'
```