Auto estudio 2

Jonatan Palomares Castañeda - Juan Diego Patiño Muñoz

Investigación

NULL

1. Significado

Un valor desconocido o no valor en sí mismo.

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

- aritméticos: También dará NULL
- lógicos: puede dar TRUE (TRUE OR NULL), FALSE (FALSE AND NULL) o NULL
- comparación: siempre NULL

JUNTA

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

- INNER: devuelve solo las filas donde haya matches
- LEFT: devuelve todas las filas de left y añade NULL para indicar las filas que no tuvieron match
- RIGHT: devuelve todas las filas de right y añade NULL para indicar las filas que no tuvieron match
- OUTER: devuelve todas las filas de ambas tablas y añade NULL donde no haya match

Práctica

A. Estudien las secciones SQL Joins, SQL Inner Join, SQL Left Join, SQL Full Join, SQL Self Join, SQL Union, SQL Exists, SQL Any, All, SQL Case, SQL Null Functions

I) Query: SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate FROM Orders INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

Calculus:
$\{o.OrderID,\ c.CustomerName,\ o.OrderDate \mid Orders(o) \wedge Customer(c) \wedge o.CustomerID =$

II) Query: SELECT Customers.CustomerName, Orders.OrderID FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID ORDER BY Customers.CustomerName;

Calculus:
$\{c.CustomerName,\ o.OrderID \mid Customer(c) \wedge Orders(o) \wedge (c.CustomerID = o.CustomerID$

III) Query: SELECT City FROM Customers UNION SELECT City FROM Suppliers ORDER BY City;

Calculus: $\{c.City \mid Customer(c)\} \cup \{s.City \mid Suppliers(s)\}$

IV) Query: SELECT ProductName FROM Products WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);

Calculus:
$\{p.ProductName \mid Products(p) \wedge (\exists_x \mid x \in \{od.ProductID \mid OrderDetails(od) \wedge od.Quantity =$

V) Query: SELECT ProductName, UnitPrice * (UnitsInStock + IFNULL(UnitsOnOrder, 0)) FROM Products;

Calculus:
$\{p.ProductName,\ calc \mid Products(p) \wedge ((p.UnitsOnOrder = 0 \wedge calc = p.UnitPrice * p.Units$

JOIN

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'`

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

Submit SQL restore default

Correct answer

matchid	player
1008	Mario Gómez
1010	Mario Gómez
1010	Mario Gómez
1012	Lukas Podolski
1012	Lars Bender
1026	Philipp Lahm
1026	Sami Khedira

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

```
SELECT DISTINCT id,stadium,team1,team2
FROM goal
JOIN game
ON game.id = 1012 AND game.id = goal.matchid
```

Submit SQL restore default

Correct answer

id	stadium	team1	team2
1012	Arena Lviv	DEN	GER

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.

```
SELECT player, teamid, stadium, mdate
FROM game
  JOIN goal ON (id=matchid AND teamid='GER')
```

Submit SQL restore default

Correct answer

player	teamid	stadium	mdate
Mario Gómez	GER	Arena Lviv	9 June 2012
Mario Gómez	GER	Metalist Stadium	13 June 2012
Mario Gómez	GER	Metalist Stadium	13 June 2012
Lukas Podolski	GER	Arena Lviv	17 June 2012
Lars Bender	GER	Arena Lviv	17 June 2012
Philipp Lahm	GER	PGE Arena Gdansk	22 June 2012
Sami Khedira	GER	PGE Arena Gdansk	22 June 2012

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%'`

```
SELECT team1, team2, player
FROM game
JOIN goal
ON (id = matchid AND player LIKE 'MarioX')
```

Submit SQL restore default

Correct answer

team1	team2	player
GER	POR	Mario Gómez
NED	GER	Mario Gómez
NED	GER	Mario Gómez
IRL	CRO	Mario Mandžukic
IRL	CRO	Mario Mandžukic
ITA	CRO	Mario Mandžukic
ITA	IRL	Mario Balotelli

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`

```
SELECT player, teamid, coach, gtime
FROM goal
JOIN eteam
ON (id = teamid AND gtime <= 10)
```

Submit SQL restore default

Correct answer

player	teamid	coach	gtime
Petr Jirácek	CZE	Michal Bílek	3
Václav Pilar	CZE	Michal Bílek	6
Mario Mandžukic	CRO	Slaven Bilic	3
Fernando Torres	ESP	Vicente del Bosque	4

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 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 (coach='Fernando Santos' AND team1=eteam.id)
```

Submit SQL restore default

Correct answer

mdate	teamname
12 June 2012	Greece
16 June 2012	Greece

7.

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

```
SELECT player
FROM goal
JOIN game
ON matchid = id AND stadium='National Stadium, Warsaw'
```

Submit SQL restore default

Correct answer

player
Robert Lewandowski
Dimitris Salpingidis
Alan Dzagoev
Jakub Blaszczykowski
Giorgos Karagounis
Cristiano Ronaldo
Mario Balotelli

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.

HINT

```
SELECT DISTINCT player
FROM goal
JOIN game
ON (id = matchid AND teamid <> 'GER')
WHERE (team1 = 'GER' OR team2 = 'GER')
```

Submit SQL restore default

Correct answer

player
Robin van Persie
Michael Krohn-Dehli
Georgios Samaras
Dimitris Salpingidis
Mario Balotelli

9.

Show teamname and the total number of goals scored.

COUNT and GROUP BY

```
SELECT T.teamname, COUNT(*)
FROM goal
JOIN eteam AS T
ON teamid = id
GROUP BY T.teamname
```

Submit SQL restore default

Germany	10
Greece	5
Italy	6
Netherlands	2
Poland	2
Portugal	6
Republic of Ireland	1
Russia	5
Spain	12
Sweden	5
Ukraine	2

10.

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

```
SELECT stadium, COUNT(*)
FROM game
JOIN goal
ON matchid = id
GROUP BY stadium
```

Submit SQL restore default

Correct answer

stadium	COUNT(*)
Arena Lviv	9
Donbass Arena	7
Metalist Stadium	7
National Stadium, Warsaw	9
Olimpiyskiy National Sports Complex	14
PGE Arena Gdansk	13
Stadion Miejski (Poznan)	8
Stadion Miejski (Wroclaw)	9

11.

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

```
SELECT id, mdate, COUNT(*)
FROM game
JOIN goal AS G
ON (id = matchid)
WHERE (team1 = 'POL' OR team2 = 'POL')
GROUP BY id, mdate
```

Submit SQL restore default

Correct answer

id	mdate	COUNT(*)
1001	8 June 2012	2
1004	12 June 2012	2
1005	16 June 2012	1

12.

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

```
SELECT id, mdate, COUNT(*)
FROM game
JOIN goal
ON id = matchid
WHERE goal.teamid = 'GER'
GROUP BY id, mdate
```

Submit SQL restore default

Correct answer

id	mdate	COUNT(*)
1008	9 June 2012	1
1010	13 June 2012	2
1012	17 June 2012	2
1026	22 June 2012	4
1030	28 June 2012	1

13.

.ist 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.

mdate	team1	score1	team2	score2
1 July 2012	ESP	4	ITA	0
10 June 2012	ESP	1	ITA	1
10 June 2012	IRL	1	CRO	3
...				

Notice in the query given every goal is listed. If it was a team1 goal then a 1 appears n score1, otherwise there is a 0. You could SUM this column to get a count of the goals scored by team1. **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 (id = matchid)
   GROUP BY mdate,team1,team2
   ORDER BY mdate, matchid, team1, team2
```

Submit SQL restore default

Correct answer

mdate	team1	score1	team2	score2
1 July 2012	ESP	4	ITA	0
10 June 2012	ESP	1	ITA	1
10 June 2012	IRL	1	CRO	3
11 June 2012	FRA	1	ENG	1
11 June 2012	UKR	2	SWE	1
12 June 2012	GRE	1	CZE	2
12 June 2012	POL	1	RUS	1

Quiz JOIN

Your score is: 7 out of 7

More JOIN

1.

List the films where the **yr** is 1962 and the budget is over 2000000 [Show **id**, **title**]

```
SELECT id, title
 FROM movie
 WHERE yr=1962 AND budget>2000000
```

Submit SQL restore default

id	title
59084	Mutiny on the Bounty
228186	Lawrence of Arabia
468484	The Longest Day
498150	Taras Bulba
521387	The Manchurian Candidate
555687	The Man Who Shot Liberty Valance
1217650	The Wonderful World of the Brothers Grimm
2633455	Gorath
44871835	Post restant

When was Citizen Kane released?

2.

Give year of 'Citizen Kane'.

```
SELECT yr
FROM movie
WHERE title='Citizen Kane'
```

Submit SQL restore default

Correct answer

yr
1941

Star Trek movies

3.

List all of the Star Trek movies, include the **id**, **title** and **yr** (all of these movies start with 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
```

Submit SQL restore default

Correct answer

id	title	yr
107940	Star Trek: The Motion Picture	1979
465478	Star Trek II: The Wrath of Khan	1982
221305	Star Trek III: The Search for Spock	1984
471842	Star Trek IV: The Voyage Home	1986
241218	Star Trek V: The Final Frontier	1989
579757	Star Trek VI: The Undiscovered Country	1991
723670	Star Trek: Generations	1994

4.

What **id** number does the actor 'Glenn Close' have?

```
SELECT id
FROM actor
WHERE name = 'Glenn Close'
```

Submit SQL restore default

Correct answer

id
372311

id for Casablanca

5.

What is the **id** of the 1942 film 'Casablanca'

```
SELECT id
FROM   movie
WHERE  title LIKE 'Casablanca' AND yr = 1942
```

Submit SQL restore default

Correct answer

id
132689

Get to the point

Cast list for Casablanca

6.

Obtain the cast list for 1942's 'Casablanca'.

what is a cast list?

Use **movieid=11768**, (or whatever value you got from the previous question)

```
SELECT name
FROM actor
WHERE id in (

SELECT actorid
FROM casting
WHERE movieid=132689)
```

Submit SQL restore default

Correct answer

name
Humphrey Bogart
Ingrid Bergman
Conrad Veidt
Curt Bois
Paul Panzer
Peter Lorre

7.

Obtain the cast list for the film 'Alien'

```
SELECT name
FROM actor
WHERE id in (
SELECT actorid
FROM casting
WHERE movieid = (SELECT
id
FROM movie
```

Submit SQL restore default

Correct answer

name
Sigourney Weaver
John Hurt
Ian Holm
unknown actor
unknown actor
Harry Dean Stanton

Harrison Ford movies

8.

List the films in which 'Harrison Ford' has appeared

```
SELECT a.title
FROM   movie a
       JOIN casting b
         ON a.id = b.movieid
WHERE  b.actorid IN (SELECT id
                     FROM   actor
                     WHERE  NAME = 'Harrison Ford')
```

Submit SQL restore default

Correct answer

title
Star Wars: Episode VII – The Force Awakens
Star Wars: Episode IV – A New Hope
Working Girl
Frantic
Zabriskie Point
The Fugitive

Harrison Ford as a supporting actor

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 a.title
FROM   movie a
       JOIN casting b
         ON a.id = b.movieid
WHERE  b.actorid IN (SELECT id
                     FROM   actor
                     WHERE  NAME = 'Harrison Ford') AND b.ord <> 1
```

Submit SQL restore default

Correct answer

title
Star Wars: Episode VII – The Force Awakens
Working Girl
Zabriskie Point
Apocalypse Now
Indiana Jones and the Last Crusade
American Graffiti

10.

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

```
SELECT a.title,
       b.NAME
FROM   movie a
       JOIN casting c
         ON a.id = c.movieid
       JOIN actor b
         ON b.id = c.actorid
WHERE  c.ord = 1
       AND a.yr = 1962
```

Submit SQL restore default

Correct answer

title	NAME
Julius Caesar Against The Pirates	Abbe Lane
The Sins	Abdelhalim Hafez
The Brainiac	Abel Salazar
Viyarppinte Vila	Adoor Bhasi
Les culottes rouges	Adrien Cayla-Legrand
Chûshingura: Hana no Maki, Yuki no Maki	Akira Takarada
Aradhana	Akkineni Nageswara Rao

Harder Questions

Busy years for Rock Hudson

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 a.yr,
       Count(b.NAME)
FROM   movie a
       JOIN casting c
         ON a.id = c.movieid
       JOIN actor b
         ON b.id = c.actorid
WHERE  b.NAME = 'Rock Hudson'
GROUP  BY a.yr
```

Submit SQL restore default

Correct answer

yr	Count(b.NAME)
1950	6
1951	5
1952	5
1953	7
1954	3
1955	3
1956	3

Lead actor in Julie Andrews movies

12.

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

Did you get "Little Miss Marker twice"?

```
SELECT title, n LeadingActor
FROM (
    SELECT title, id
    FROM movie JOIN casting ON movieid=movie.id
    WHERE actorid IN (
        SELECT id FROM actor WHERE name ="Julie
Andrews")
    ) m JOIN (
        SELECT actor.name n, movieid
        FROM casting JOIN actor ON actorid=actor.id
        WHERE ord=1
    ) act ON movieid=id
```

Submit SQL restore default

Correct answer

title	LeadingActor
The Sound of Music	Julie Andrews
The Man Who Loved Women	Burt Reynolds
Tooth Fairy	Dwayne Johnson
10	Dudley Moore
Mary Poppins	Julie Andrews
Torn Curtain	Paul Newman

Actors with 15 leading roles

13.

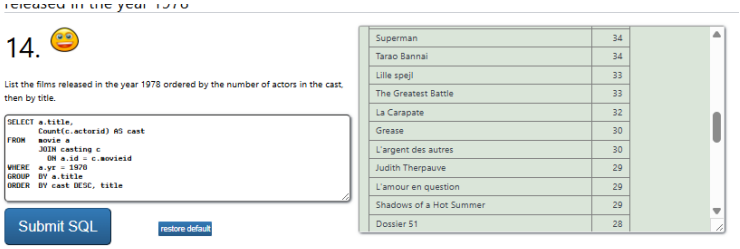
Obtain a list, in alphabetical order, of actors who've had at least 15 **starring** roles.

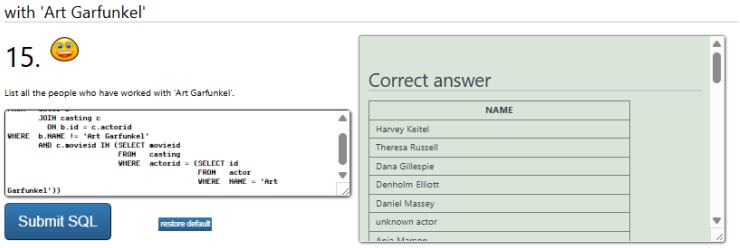
```
SELECT name
FROM casting JOIN actor ON actorid=id
WHERE ord=1
GROUP BY name
HAVING count(*) >= 15
ORDER BY name
```

Submit SQL restore default

Correct answer

name
A. V. M. Rajan
Aamir Khan
Aaron Eckhart
Aaron Kwok
Abhishek Bachchan
Abir Chatterjee
Adam Sandler





Quiz More JOIN

Your score is: 7 out of 7

Using NULL

NULL, INNER JOIN, LEFT JOIN, RIGHT JOIN

1.

List the teachers who have NULL for their department.

Why we cannot use =

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

Submit SQL restore default

Correct answer

name
Spiregrain
Deadyawn

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)
```

Submit SQL restore default

Correct answer

name	name
Shrivell	Computing
Throd	Computing
Splint	Computing
Cutflower	Design

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)
```

Submit SQL restore default

Correct answer

name	name
Shrivell	Computing
Throd	Computing
Splint	Computing
Spiregrain	null
Cutflower	Design
Deadyawn	null

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)
```

Submit SQL restore default

Correct answer

name	name
Shrivell	Computing
Throd	Computing
Splint	Computing
Cutflower	Design
null	Engineering

Using the COALESCE function

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
```

Submit SQL restore default

Correct answer

name	COALESCE(mobile, '07986 444 2266')
Shrivell	07986 555 1234
Throd	07122 555 1920
Splint	07986 444 2266
Spiregrain	07986 444 2266
Cutflower	07996 555 6574
Deadyawn	07986 444 2266

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

Submit SQL restore default

Correct answer

COALESCE(teacher.name, 'NONE')	COALESCE(dept.name, 'None')
Shrivell	Computing
Throd	Computing
Splint	Computing
Spiregrain	None
Cutflower	Design
Deadyawn	None

7.

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

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

Submit SQL restore default

Correct answer

COUNT(name)	COUNT(mobile)
6	3

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
```

Submit SQL restore default

Correct answer

name	COUNT(teacher.name)
Computing	3
Design	1
Engineering	0

Using CASE

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
```

Submit SQL restore default

Correct answer

name	CASE WHEN (dept=1 OR dept=2) THEN 'Sci' ELSE 'Art' END
Shrivell	Sci
Throd	Sci
Splint	Sci
Spiregrain	Art
Cutflower	Sci
Deadyawn	Art

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 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 (dept.id=teacher.dept)
```

Submit SQL restore default

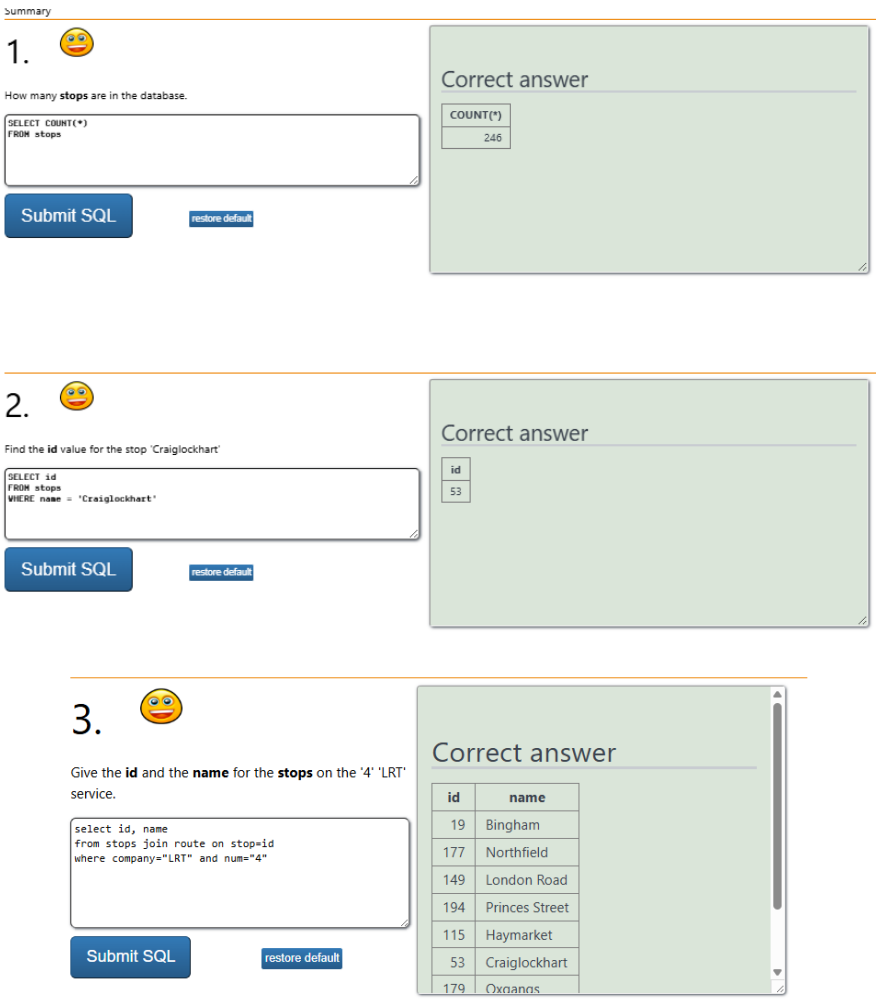
Correct answer

name	CASE WHEN dept.id = 1 THEN 'Sci' WHEN dept.id = 2 THEN 'Sci' WHEN dept.id = 3 THEN 'Art' ELSE 'None' END
Shrivell	Sci
Throd	Sci
Splint	Sci
Spiregrain	None
Cutflower	Sci
Deadyawn	None

Quiz Using NULL

Your score is: 6 out of 6

Self join


Summary
1.
How many **stops** are in the database.
SELECT COUNT(*)
FROM stops
Submit SQL
restore default
Correct answer
COUNT(*)
246
2.
Find the **id** value for the stop 'Craiglockhart'
SELECT id
FROM stops
WHERE name = 'Craiglockhart'
Submit SQL
restore default
Correct answer
id
53
3.
Give the **id** and the **name** for the **stops** on the '4' 'LRT' service.
select id, name
from stops join route on stop=id
where company="LRT" and num="4"
Submit SQL
restore default
Correct answer
id
name
19
Bingham
177
Northfield
149
London Road
194
Princes Street
115
Haymarket
53
Craiglockhart
179
Oxgangs

Routes and stops

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(*) AS visits
FROM route WHERE stop=149 OR stop=53
GROUP BY company, num
HAVING visits=2
```

Submit SQL restore default

Correct answer

company	num	visits
LRT	4	2
LRT	45	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
```

Submit SQL restore default

Correct answer

company	num	stop	stop
LRT	4	53	149
LRT	45	53	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'
```

Submit SQL restore default

Correct answer

company	num	name	name
LRT	4	Craiglockhart	London Road
LRT	45	Craiglockhart	London Road

Using a self join

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)
  JOIN stops stopa ON (a.stop=stopa.id)
  JOIN stops stopb ON (b.stop=stopb.id)
WHERE stopa.name='Haymarket' AND stopb.name='Leith'
```

Submit SQL restore default

Correct answer

company	num
LRT	12
LRT	2
LRT	22
LRT	25
LRT	2A
SMT	C5

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'
```

Submit SQL restore default

Correct answer

company	num
LRT	10
LRT	27
LRT	45
LRT	47

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 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'
```

Submit SQL restore default

Wrong answer. Some of the values are incorrect.

name	company	num
Balerno	LRT	47
Balerno Church	LRT	47
Bingham	LRT	4
Brunstane	LRT	45
Canonmills	LRT	27

Quiz Self join

Your score is: 3 out of 3

Consultas usando operadores

1. Operadores de conjuntos

I) UNION: Liste los nombres y apellidos de aquellas personas que sean parte del personal o sean personas que llaman.

```
SELECT First_name, Last_name
FROM Staff
UNION
SELECT First_name, Last_name
FROM Caller
```

II) UNION ALL: ¿Qué miembros del personal tomaron o se les fue asignada una llamada, y cuál llamada fue? Incluya duplicados en caso de que una persona haya tomado y se le haya asignado una llamada.

```
SELECT Taken_by Staff_code, Call_ref
FROM Issue
UNION ALL
SELECT Assigned_to Staff_code, Call_ref
FROM Issue
```

III) INTERSECT: ¿Cuántos nombres completos pertenecen tanto a una persona del personal como a una persona que llama?

```
SELECT count(*)
FROM (
    SELECT First_name, Last_name
```

```
    FROM Staff
    INTERSECT
    SELECT First_name, Last_name
    FROM Caller
) subquery
```

IV) EXTRACT: ¿Qué miembros del personal nunca han atendido una llamada?

```
SELECT Staff_code
FROM Staff
EXCEPT
SELECT Taken_by
FROM Issue
```

V) IN: ¿Qué miembros del personal han tomado por lo menos una llamada?

```
SELECT s.Staff_code, s.First_name, s.Last_name
FROM Staff s
WHERE s.Staff_code IN (
   SELECT Taken_by
   FROM Issue
)
```

2. Operadores de junta

I) JOIN: Liste todos los problemas junto con el nombre y apellido de quien los atendió

```
SELECT i.Call_ref, s.First_name, s.Last_name
FROM Issue i
INNER JOIN Staff s ON i.Taken_by = s.Staff_code
```

II) NATURAL JOIN: ¿Cuántas distintas llamadas se les han asignado a cada uno de los miembros del personal?

```
SELECT s.Staff_code, s.First_name, s.Last_name,
  count(DISTINCT c.Caller_id) distinct_callers
```

```
FROM (
  SELECT Assigned_to Staff_code, Caller_id
  FROM Issue
) i
NATURAL JOIN Staff s
NATURAL JOIN (SELECT Caller_id from Caller) c
GROUP BY s.Staff_code, s.First_name, s.Last_name
ORDER BY distinct_callers DESC
```

III) CROSS JOIN: Liste todas las posibles parejas personal-persona que llama

```
SELECT s.Staff_code, s.First_name, s.Last_name, c.Caller_id,
c.First_name, c.Last_name
FROM Staff s
CROSS JOIN Caller c
ORDER BY s.Last_name, s.First_name, c.Last_name,
c.First_name
```

IV) LEFT JOIN: ¿Cuántos problemas ha reportado cada una de las personas que llaman?

```
SELECT c.Caller_id, c.First_name, c.Last_name,
count(i.Call_ref) issue_count
FROM Caller c
LEFT JOIN Issue i ON c.Caller_id = i.Caller_id
GROUP BY c.Caller_id, c.First_name, c.Last_name
ORDER BY issue_count
```

V) RIGHT JOIN: ¿Para cuántas llamadas ha sido asignado cada miembro del personal?

```
SELECT s.Staff_code,s.First_name,s.Last_name,
count(i.Call_ref) assigned_count
FROM Issue i
RIGHT JOIN Staff s ON i.Assigned_to = s.Staff_code
GROUP BY s.Staff_code, s.First_name, s.Last_name
```

ORDER BY assigned_count

3. Operadores de desconocido

I) COALESCE: ¿Cuáles son las distintas direcciones alternativas de cada compañía?

```
SELECT DISTINCT COALESCE(Address_2, "None")
SecondAddresses
FROM Customer
```

II) ISNULL: ¿Qué compañías no tienen una segunda dirección?

```
SELECT Company_name
FROM Customer
WHERE ISNULL(Address_2)
```

4. Operadores lógicos:

I) EXISTS: ¿Qué miembros del personal nunca han tomado una llamada?

```
SELECT s.Staff_code, s.First_name, s.Last_name
FROM Staff s
WHERE NOT EXISTS (
  SELECT Call_ref
  FROM Issue i
  WHERE i.Taken_by = s.Staff_code
)
```

II) ANY:

```
SELECT s.Staff_code, s.First_name,
s.Last_name,count(i.Call_ref) c
FROM Staff s
LEFT JOIN Issue i ON i.Taken_by = s.Staff_code
GROUP BY s.Staff_code, s.First_name, s.Last_name
HAVING count(i.Call_ref) > ANY (
  SELECT count(*)
  FROM Issue
```

```
  GROUP BY Taken_by
)
```

III) ALL: ¿A qué miembros del personal que han recibido llamadas nunca les han asignado una?

```
SELECT s.Staff_code, s.First_name, s.Last_name
FROM Staff s
WHERE EXISTS (
  SELECT i.Call_ref
  FROM Issue i
  WHERE i.Taken_by = s.Staff_code
)
AND s.Staff_code <> ALL (
  SELECT Assigned_to
  FROM Issue
  WHERE Assigned_to IS NOT NULL
)
```

5. Operador CASE: Clasifique las llamadas en asignadas y no asignadas

```
SELECT Call_ref, CASE WHEN Assigned_to IS NULL THEN
"No asignado" ELSE "Asignado" END AS Status
FROM Issue
```

Bibliografía

W3Schools.com. (n.d.). https://www.w3schools.com/sql/default.asp