



Universidad
Zaragoza

Mario Labrador - 869022

Práctica 3

Bases de Datos

Mario Labrador
869022



Universidad Zaragoza

unizar.es

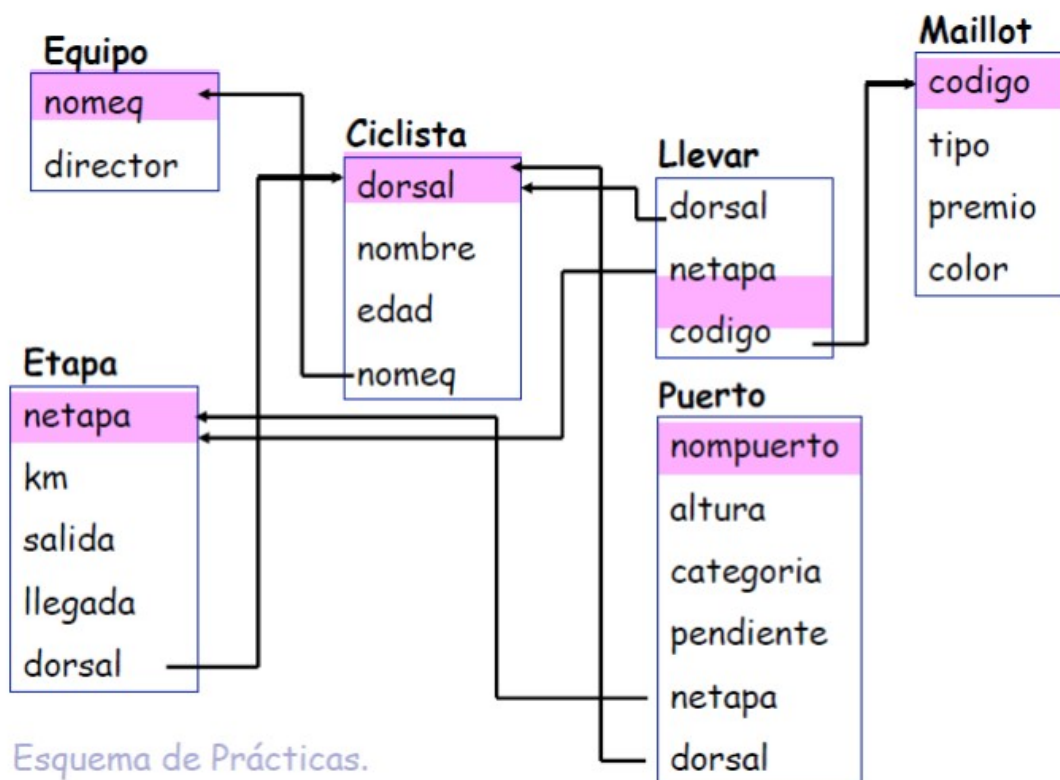
Contenido

Introducción	3
Diseño entidad-relación	4
Diseño lógico	5
Consultas SQL	6
Tareas de administración	12
Conclusiones	12
Referencias bibliográficas	13

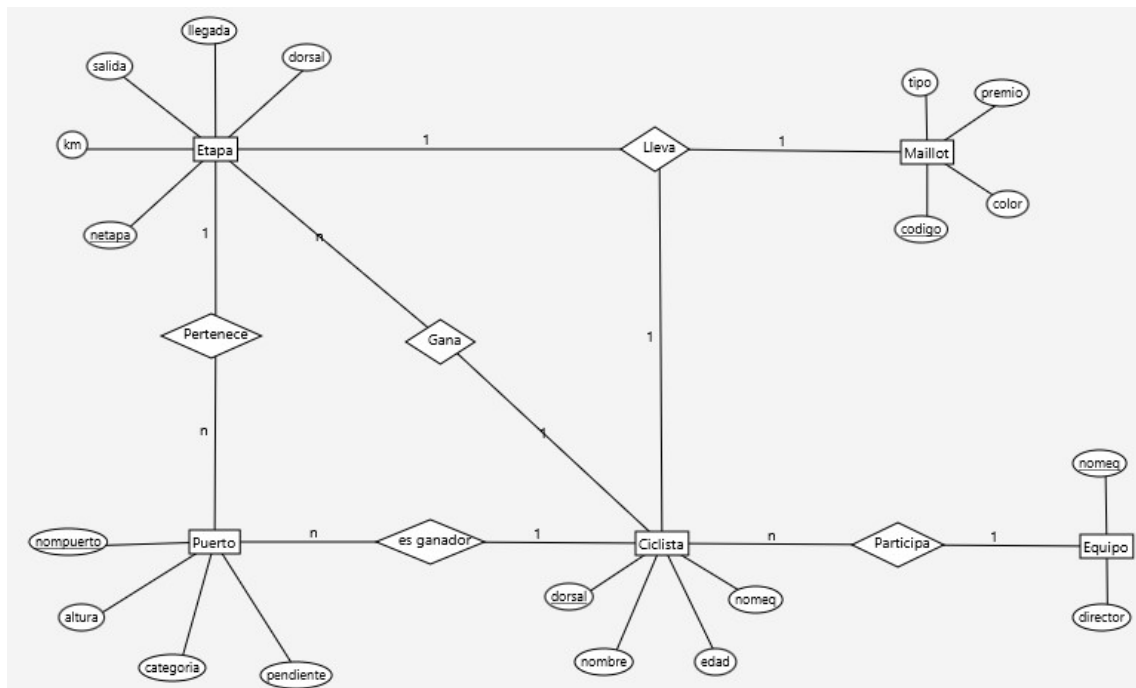


Introducción

En esta práctica realizaremos una lista de consultas SQL utilizando el SGBDR ANACON. Como este SGBDR tiene una estructura Cliente-Servidor, utilizaremos el archivo 'conexion.udl', que nos permitirá acceder al servidor que contiene la base de datos sobre la que vamos a trabajar. Esta base de datos contiene información sobre una vuelta ciclista. Su diseño físico es el siguiente:



Diseño entidad-relación



Diseño lógico

Equipo(nomeq: dom_nomeq, director: dom_director)

CP{nomeq}

Ciclista(dorsal: dom_dorsal, nombre: dom_nombre, edad: dom_edad, nomeq: dom_nomeq)

CP: {dorsal}

C.Aj: {nomeq}→equipo

Etapas(netapa: dom_netapa, km: dom_km, salida: dom_salida, llegada: dom_llegada, dorsal: dom_dorsal)

CP: {netapa}

C.Aj: {dorsal}→ciclista

Puerto(nompuerto: dom_nompuerto, altura: dom_altura, categoria: dom_categoria, pendiente: dom_pendiente, netapa: dom_netapa, dorsal: dom_dorsal)

CP: {nompuerto}

C.Aj: {netapa}→etapa

C.Aj: {dorsal}→ciclista

Maillot(codigo: dom_codigo, tipo: dom_tipo, color: dom_color, premio: dom_premio)

CP: {dom_premio}

Llevar(netapa: dom_netapa, codigo: dom_codigo, dorsal: dom_dorsal)

CP: {netapa, codigo}

Único: {netapa, codigo}

VNN: {dorsal}

C.Aj: {netapa}→etapa

C.Aj: {codigo}→maillot

C.Aj: {dorsal}→ciclista

Consultas SQL

1. Obtener el número de etapa, ciudad de salida y ciudad de llegada de las etapas con longitud mayor de 200 km.

```
SELECT netapa, salida, llegada
FROM etapa
WHERE km > 200
```

netapa	salida	llegada
3	Salamanca	Caceres
4	Almendralejo	Córdoba
7	Baza	Alicante
12	Benasque	Zaragoza
15	Santo Domingo de la Calzada	Santander

2. Obtener el nombre y la altura de los puertos de 1ª categoría.

```
SELECT nompuerto, altura
FROM puerto
WHERE categoria LIKE '1'
```

nompuerto	altura
Alto del Naranco	565
Coll de la Comella	1362
Navacerrada	1860
Puerto de Alisas	672
Puerto de Mijares	1525
Puerto de Pedro Bernardo	1250

3. Obtener la lista de equipos, incluyendo nombre y director, por orden alfabético de nombre de equipo.

```
SELECT nomeq, director
FROM equipo
ORDER BY nomeq ASC
```

nomeq	director
Artiach	Jean Philip
Banesto	José Pérez
Brescial-Prefin	Miguel Echevarria
Carrera	Pietro Amiani
Euskadi	Luigi Petroni
Gatorade	Pedro Txucaru
Gewiss	Gian Luca Paccelli
Jolly Club	Moreno Argentin
Kelme	Johan Richard
Lexus Festina	Alvaro Pino
Mapei-Clas	Suarez Cuevas
Mercatone Uno	Juan Fernandez
Motorola	Ettore Romano
Navigare	John Fiddell
ONCE	Lorenzo Sciacchi
PDM	Manuel Sainz
Seguros Amaya	Piet Van Der Knaap
Telecom	Minguez
TVM	Morgan Reikard
whi	Stevens Henk
Wordperfect	Ricardo Padacci
	Bill Gates

4. Obtener las diferentes categorías de puertos existentes.

```
SELECT DISTINCT(categoria)
FROM puerto
```

categoria
1
2
E

5. Obtener los nombres y edades de los ciclistas con edad comprendida entre 20 y 25 años (ambos incluidos).

```
SELECT nombre, edad
FROM ciclista
WHERE edad BETWEEN 20 AND 25
```

nombre	edad
Javier Palacin	25
Rolf Aldag	25
Agustin Sagasti	24
Eugeni Berzin	23
Enrico Zaina	24
Eleuterio Anguita	25

6. Obtener el nombre de los ciclistas que empiezan por "F".

```
SELECT nombre
FROM ciclista
WHERE nombre LIKE 'F%'
```

nombre
Flavio Giupponi
Francesco Casagrande
Fernando Escartin
Federico Echave
Fernando Mota
Federico Garcia

7. Obtener el número identificador de las etapas donde el nombre de la ciudad de llegada tenga por segunda letra una “E”, o donde el nombre de la ciudad de salida lleve dos o más “A”s.

```
SELECT netapa
FROM etapa
WHERE llegada LIKE '_e%' OR salida
LIKE '%a%a%'
```

netapa
1
2
3
4
6
7
8
10
11
13
14
15
16
17
19
20

8. Obtener todos los datos de los ciclistas de los que se desconoce su edad.

```
SELECT *
FROM ciclista
WHERE edad IS NULL
```

dorsal	nombre	edad	nomeq

9. Obtener el nombre y director de los equipos cuyo nombre contenga la cadena “100%”.

```
SELECT nomeq, director
FROM equipo
WHERE nomeq LIKE '%100#%'
```

nomeq	director

10. Obtener el número y longitud de las etapas que salen de Valladolid, Zaragoza o Segovia.

```
SELECT netapa, km
FROM etapa
WHERE salida IN ('Valladolid',
'Zaragoza', 'Segovia')
```

netapa	km
1	9
2	180
13	200
20	52

11. Obtener el nombre de cada ciclista y el director del equipo al que pertenece.

```
SELECT c.nombre, e.director
FROM ciclista c, equipo e
WHERE c.nomeq = e.nomeq
```

nombre	director
Alfonso Gutiérrez	José Pérez
Erwin Nijboer	José Pérez
Alberto Elli	José Pérez
Fernando Mola	José Pérez
Ivan Ivanov	José Pérez
Federico Garcia	José Pérez
Eleuterio Anguita	José Pérez
Miguel Indurán	Miguel Echevarria
Pedro Delgado	Miguel Echevarria
Julian Gorospe	Miguel Echevarria
Mikel Zarrabellia	Miguel Echevarria
Jesus Montoya	Miguel Echevarria
Melchor Mauri	Miguel Echevarria
Vicente Aparicio	Miguel Echevarria
Andrew Hampsten	Miguel Echevarria
Sandro Heulot	Miguel Echevarria
Prudencio Indurán	Miguel Echevarria
Marino Alonso	Miguel Echevarria
Flavio Giupponi	Pietro Armani
Bruno Leali	Pietro Armani
Giovanni Lombardi	Pietro Armani
Stefano Colage	Pietro Armani
Claudio Chiappucci	Luigi Pieltoni
Dimitri Abdouaparov	Luigi Pieltoni
Stefan Roche	Luigi Pieltoni
Javier Palacin	Pedro Txucaru
Agustin Sagasti	Pedro Txucaru
Gianni Bugno	Gian Luca Paccelli
Luca Gelli	Gian Luca Paccelli

12. Obtener el nombre y número de ciclistas de cada equipo.

```
SELECT c.nomeq, COUNT(c.nomeq)
FROM ciclista c, equipo e
WHERE c.nomeq = e.nomeq
GROUP BY c.nomeq
```

nomeq	COLUMN1
Artiach	7
Banesto	11
Brescial-Refin	4
Carrera	3
Euskadi	2
Gatorade	4
Gewiss	8
Jolly Club	2
Kelme	7
Lotus Festina	3
Mapei-Clas	7
Mercatone Uno	8
Motorola	3
Navigare	5
ONCE	5
Seguros Amaya	3
Telecom	4
TVM	6
Wordperfect	3

OPCIONAL: Conseguir que aparezcan también los equipos con 0 ciclistas.

```
SELECT e.nomeq, (SELECT COUNT(*)
FROM ciclista c
WHERE c.nomeq = e.nomeq)
FROM equipo e
GROUP BY e.nomeq;
```

nomeq	COLUMN1
Artiach	0
Banesto	7
Brescial-Refin	11
Carrera	4
Euskadi	3
Gatorade	2
Gewiss	4
Jolly Club	8
Kelme	2
Lotus Festina	7
Mapei-Clas	3
Mercatone Uno	7
Motorola	8
Navigare	3
ONCE	5
PDM	0
Seguros Amaya	3
Telecom	4
TVM	6
wh	0
Wordperfect	3

13. Obtener el número total de etapas ganadas por cada equipo (es decir, por ciclistas de cada equipo), siempre que ese equipo haya ganado más de 2 etapas.

```
SELECT eq.nomeq, COUNT(c.dorsal)
FROM etapa e, ciclista c, equipo eq
WHERE e.dorsal = c.dorsal AND
c.nomeq = eq.nomeq
GROUP BY eq.nomeq
HAVING COUNT(c.dorsal) > 2
```

nomeq	COLUMN1
Banesto	5

14. Obtener el nombre y edad del ciclista más joven.

```
SELECT nombre, edad
FROM ciclista
WHERE edad = (SELECT MIN(edad)
FROM ciclista)
```

nombre	edad
Eugeni Berzin	23

15. Obtener la longitud media de las etapas que tienen más de 2 puertos de montaña.

```
SELECT AVG(e.km)
FROM etapa e
WHERE e.netapa IN (SELECT
p.netapa
FROM puerto p
GROUP BY p.netapa
HAVING COUNT (p.nompuerto) > 2)
```

COLUMN1
197


```
SELECT AVG(e.km)
FROM etapa e, puerto p
WHERE e.netapa = p.netapa
GROUP BY e.netapa
HAVING COUNT (p.nompuerto) > 2
```

COLUMN1
200
195

16. Obtener el nombre de los ciclistas que no han ganado ninguna etapa.

```
SELECT nombre
FROM ciclista
WHERE dorsal NOT IN (SELECT dorsal
FROM etapa)
```

nombre
Alex Zülle
Adriano Baffi
Massimiliano Lelli
Massimo Podenzana
Flavio Giupponi
Lale Cubino
Roberto Pagnin
Jesper Skibby
Dimitri Konishev
Bruno Leali
Robert Millar
Julian Gorospe
Alfonso Gutiérrez
Erwin Nijboer
Lance Armstrong
Claudio Chiappucci
Gianni Bugno
Mikel Zarrabeitia
Jesus Montoya
Angel Edo
Melchor Mauri
Vicente Aparicio
Laurent Dufaux
Stefano della Santa
Angel Yesid Camargo
Scott Sunderland
Javier Palacin
Rudy Verdonck
Viatceslav Ekinov

17. Obtener toda la información sobre las etapas que contienen el puerto (o los puertos) de montaña más altos.

```
SELECT e.*
FROM etapa e, puerto p
WHERE e.netapa = p.netapa AND
p.altura = (SELECT MAX(altura)
FROM puerto)
```

netapa	km	salida	llegada	dorsal
2	180	Valldolid	Salamanca	36
11	195	Andorra	Estación de Cerler	65

18. Obtener el dorsal y nombre de los ciclistas que llevaban cada uno de los diferentes maillots en la última etapa.

```
SELECT c.dorsal, c.nombre
FROM ciclista c, llevar l
WHERE c.dorsal = l.dorsal AND
l.netapa IN (SELECT MAX(netapa)
FROM etapa)
```

dorsal	nombre
1	Miguel Induráin
2	Pedro Delgado
20	Alfonso Gutiérrez
22	Giorgio Furlan
42	Davide Cassani

19. Obtener los equipos con algún ciclista que haya llevado el maillot amarillo en algún momento de la vuelta.

```
SELECT e.nomeq
FROM equipo e, ciclista c, llevar l,
maillot m
WHERE e.nomeq = c.nomeq AND
c.dorsal = l.dorsal AND l.codigo =
m.codigo AND m.color LIKE 'amarillo'
GROUP BY e.nomeq
```

nomeq
Banesto
Mapei-Clas
ONCE

20. Obtener la lista de ciclistas, dorsal y nombre, ordenada por el número de etapas que ha ganado cada uno (de más a menos).

```
SELECT c.nombre, c.dorsal, c.edad,
c.nomeq, COUNT(c.dorsal)
FROM ciclista c, etapa e
WHERE c.dorsal = e.dorsal
GROUP BY c.nombre, c.dorsal,
c.edad, c.nomeq
ORDER BY COUNT(c.dorsal) DESC,
c.nombre ASC
```

nombre	dorsal	edad	nomeq	COLUMN1
Pedro Delgado	2	35	Banesto	3
Alessio Di Biasio	12	31	Amore Vita	2
Laurent Jalabert	27	28	ONCE	2
Miguel Indurain	1	32	Banesto	2
Bo Hamburger	93	29	TVM	1
Erik Dekker	35	28	Wordperfect	1
Gert-Jan Theunisse	5	32	TVM	1
Gian Matteo Fagnini	36	32	Mercatone Uno	1
Giorgio Furlan	22	32	Gewiss	1
Herman Buenaflora	83	32	Kelme	1
Jean Van Poppel	8	33	Lotus Festina	1
Juan Martinez Oliver	86	32	Kelme	1
Mario Cipollini	10	28	Mercatone Uno	1
Pascal Lino	65	29	Amore Vita	1
Tony Rominger	4	30	Mapei-Clas	1
Vladislav Bobik	52	26	Gewiss	1

OPCIONAL: Conseguir que aparezcan también los ciclistas que no han ganado ninguna etapa, y

en el caso de que varios ciclistas hayan ganado el mismo número de etapas, deben aparecer ordenados alfabéticamente por nombre.

```
SELECT c.nombre, c.dorsal, c.edad,
c.nomeq, (SELECT COUNT(*)
FROM etapa e
WHERE e.dorsal = c.dorsal) AS
victorias
FROM ciclista c
GROUP BY c.nombre, c.dorsal,
c.edad, c.nomeq
ORDER BY victorias DESC, c.nombre
ASC
```

nombre	dorsal	edad	nomeq	victorias
Pedro Delgado	2	35	Banesto	3
Alessio Di Biasio	12	31	Amore Vita	2
Laurent Jalabert	27	28	ONCE	2
Miguel Indurain	1	32	Banesto	2
Bo Hamburger	93	29	TVM	1
Erik Dekker	35	28	Wordperfect	1
Gert-Jan Theunisse	5	32	TVM	1
Gian Matteo Fagnini	36	32	Mercatone Uno	1
Giorgio Furlan	22	32	Gewiss	1
Herman Buenaflora	83	32	Kelme	1
Jean Van Poppel	8	33	Lotus Festina	1
Juan Martinez Oliver	86	32	Kelme	1
Mario Cipollini	10	28	Mercatone Uno	1
Pascal Lino	65	29	Amore Vita	1
Tony Rominger	4	30	Mapei-Clas	1
Vladislav Bobik	52	26	Gewiss	1
Adriano Baffi	6	33	Mercatone Uno	0
Agustín Sagasti	46	24	Euskadi	0
Alberto Elli	45	26	Artiach	0
Alex Zülle	3	27	ONCE	0
Alfonso Gutiérrez	20	29	Artiach	0
Alvaro Mejia	63	32	Motorola	0
Andrew Hampsten	56	29	Banesto	0
Angel Camanillo	88	28	Mapei-Clas	0
Angel Citaoca	68	28	Navigare	0
Angel Edo	29	28	Kelme	0
Angel Yessid Canavero	34	30	Kelme	0
Antonio Esparza	78	35	Kelme	0
Armand de las Cuevas	67	28	Castorama	0

21. Obtener el color de aquellos maillots que sólo han sido llevados por ciclistas de un mismo equipo.

```
SELECT m.color
FROM maillot m, llevar l, ciclista c,
equipo e
WHERE m.codigo = l.codigo AND
l.dorsal = c.dorsal AND c.nomeq =
e.nomeq
GROUP BY m.color
HAVING COUNT(DISTINCT e.nomeq)
= 1
```

color

22. Obtener el nombre de los ciclistas que han ganado todos los puertos de montaña y además han ganado esa misma etapa. Obtener junto con el nombre del ciclista el número de la etapa en cuestión.

```
SELECT c.nombre, e.netapa
FROM ciclista c, etapa e, puerto p
WHERE e.netapa = p.netapa AND
p.dorsal = c.dorsal
GROUP BY c.nombre, e.netapa
HAVING COUNT(DISTINCT p.dorsal) =
1
```

nombre	netapa
► Mikel Zarrabeitia	2
Massimiliano Lelli	10
Melchor Mauri	10
Pedro Delgado	10
Tony Rominger	10
Alfonso Gutiérrez	11
Massimo Podenzana	11
Miguel Induráin	15
Davide Cassani	16
Claudio Chiappucci	18
Gianni Bugno	18
Pedro Delgado	18
Pedro Delgado	19

23. Obtener el nombre y director del equipo con mayor media de edad de sus ciclistas.

```
SELECT nomeq, director
FROM equipo e
WHERE EXISTS (SELECT 1
FROM ciclista c
WHERE c.nomeq = e.nomeq
GROUP BY c.nomeq
HAVING AVG(c.edad) = (SELECT
MAX(media_edad)
FROM (SELECT AVG(edad) AS
media_edad
FROM ciclista
GROUP BY nomeq) AS subconsulta))
```

nomeq	director
► Gatorade	Gian Luca Paccelli

Tareas de administración

Para poder trabajar con la base de datos, he tenido que acceder a *nas-ter2*, copiar los archivos ANACON.exe (el SGBDR) y el archivo 'conexion.udl', el cual me permite conectarme al servidor que contiene la base de datos de una vuelta ciclista.

El software ANACON se distribuye de forma libre bajo licencia GNU Public License (GPL). La conexión que utiliza es una conexión Cliente-Servidor por lo que solo podemos acceder a la base de datos cuando estamos conectados a la misma red que el servidor, en este caso, *eduroam* (*también mediante la VPN FortiClient*).

No he tenido ningún problema a la hora de hacer funcionar el SGBD.

Conclusiones

Con esta práctica he podido comprobar la funcionalidad de las consultas SQL mediante el programa ANACON, un SGBDR (sistema gestor de base de datos relacional) que permite hacer consultas de manera rápida y sencilla.

Comencé realizando el diagrama Entidad-Relación de la base de datos de una vuelta ciclista y a continuación descargué el software necesario y realicé las consultas SQL.

En un inicio, las consultas son sencillas por lo que solo tengo que utilizar SELECT y FROM para mostrar información de alguna entidad sin filtrar los resultados obtenidos. En las siguientes consultas ya se pide que filtren los resultados siguiendo determinados requisitos que deben cumplir las consultas por lo que es necesario utilizar las sentencias WHERE, con la cual indicamos las condiciones que deben cumplir los datos y ORDER BY para ordenar los resultados obtenidos, además de funciones agregadas como MIN, MAX, COUNT...

También utilizo predicados como BETWEEN para comprobar que valores están en un rango, LIKE para comprobar si contienen la secuencia dada. Además, utilizo IN, que nos dice si se encuentra el valor dado, DISTINCT que elimina los resultados repetidos e IS NULL para comprobar valores nulos.

Por último, en consultas más complejas, es necesario utilizar GROUP BY que permite agrupar consultas y obtener información global y HAVING, que sirve para añadir condiciones a las consultas agrupadas con GROUP BY. Además, he utilizado subconsultas, lo cual me ha permitido descomponer y estructurar las condiciones para que así la consulta sea más sencilla.

Referencias bibliográficas

1. Garrido, P. *Tema 4.El lenguaje SQL*. Teruel: Universidad de Zaragoza, 2025.
2. Velilla, S. Tema III: el enfoque Relacional. Zaragoza: Univ. De Zaragoza, 2011/2012.
3. Garrido, P. *Práctica 3: Structured Query Language (SQL)*. Teruel: Universidad de Zaragoza, 2025.

Este trabajo tiene licencia CC BY 4.0. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0>