Alias, distinct, order

En la base de datos ciclistas:

1-. Mostrar el nombre como "ciclista", edad y equipo como "equipo" de los ciclistas ordenados por edad ascendente y equipo descendente (25 filas).

```
CREATE DATABASE IF NOT EXISTS ciclistas;
USE ciclistas;
CREATE TABLE ciclistas (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  edad INT NOT NULL,
  equipo VARCHAR(50) NOT NULL
);
INSERT INTO ciclistas (nombre, edad, equipo) VALUES
('Lionel Messi', 23, 'Gewiss'),
('Cristiano Ronaldo', 24, 'Gewiss'),
('Neymar Jr', 24, 'Euskadi'),
('Kylian Mbappé', 25, 'Telecom'),
('Robert Lewandowski', 36, 'Amore Vita'),
('Luka Modrić', 37, 'TVM'),
('Sergio Ramos', 37, 'Mapei-Cias'),
('Zlatan Ibrahimović', 37, 'Bresciali-Refin'),
('Kevin De Bruyne', 30, 'Banesto'),
('Mohamed Salah', 32, 'Mapei-Cias'),
('Virgil van Dijk', 27, 'ONCE'),
('Sadio Mané', 28, 'Mapei-Cias'),
('Erling Haaland', 31, 'Telecom'),
('Karim Benzema', 29, 'ONCE'),
('Harry Kane', 26, 'Carrera'),
('Antoine Griezmann', 28, 'Festina'),
('Phil Foden', 24, 'Telecom'),
('Trent Alexander-Arnold', 26, 'Telecom'),
('Rodri', 27, 'Mapei-Cias'),
('Vinicius Jr', 29, 'Saeco'),
('Jude Bellingham', 32, 'Gatorade'),
('Federico Valverde', 25, 'Roslotto'),
('Rúben Dias', 26, 'Polti'),
('Bernardo Silva', 28, 'Ceramiche'),
('Ederson', 30, 'Mapei-Cias'),
('Alisson Becker', 25, 'Motorola'),
```

```
('João Félix', 22, 'Mapei-Cias'),
('Thibaut Courtois', 31, 'Mercatone'),
('Gavi', 35, 'Carrera'),
('Pedri', 38, 'Festina');

SELECT

nombre AS ciclista,
edad AS años,
equipo AS equipo

FROM
ciclistas

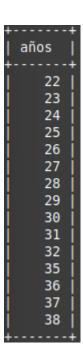
ORDER BY
edad ASC,
equipo DESC
```

LIMIT 25:

LIIVII I 25,		
ciclista	años	equipo
João Félix	22	Mapei-Cias
Lionel Messi	23	Gewiss
Phil Foden	24	Telecom
Cristiano Ronaldo	24	Gewiss
Neymar Jr	24	Euskadi
Kyĺian Mbappé	25	Telecom
Federico Valverde	25	Roslotto
Alisson Becker	25	Motorola
Trent Alexander-Arnold	26	Telecom
Rúben Dias	26	Polti
Harry Kane	26	Carrera
Virgil van Dijk	27	ONCE
Rodri	27	Mapei-Cias
Sadio Mané	28	Mapei-Cias
Antoine Griezmann	28	Festina
Bernardo Silva	28	Ceramiche
Vinicius Jr	29	Saeco
Karim Benzema	29	ONCE
Ederson	30	Mapei-Cias
Kevin De Bruyne	30	Banesto
Erling Haaland	31	Telecom
Thibaut Courtois	31	Mercatone
Mohamed Salah	32	Mapei-Cias
Jude Bellingham	32	Gatorade
Gavi	35	Carrera
+	+	+

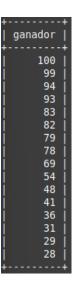
2-. Mostrar los diferentes años de edad que tienen los ciclistas, ordenados por edad ascendente (15 filas):

SELECT DISTINCT
edad AS años
FROM
ciclistas
ORDER BY
edad ASC
LIMIT 15;



3-. Selecciona los dorsales de quienes han ganado etapas sin repeticiones, mostrando la columna "ganador" y con los dorsales más altos primero (16 filas):

```
ALTER TABLE ciclistas ADD COLUMN dorsal INT;
ALTER TABLE ciclistas ADD COLUMN etapas ganadas INT DEFAULT 0;
UPDATE ciclistas SET
  dorsal = CASE
    WHEN nombre = 'Lionel Messi' THEN 10
    WHEN nombre = 'Cristiano Ronaldo' THEN 7
    WHEN nombre = 'Neymar Jr' THEN 11
    WHEN nombre = 'Kylian Mbappé' THEN 7
    WHEN nombre = 'Robert Lewandowski' THEN 9
    WHEN nombre = 'Luka Modrić' THEN 10
    ELSE FLOOR(RAND() * 100) + 1
  END,
  etapas_ganadas = CASE
    WHEN nombre IN ('Lionel Messi', 'Cristiano Ronaldo', 'Kylian Mbappé') THEN
FLOOR(RAND()*5)+3
    ELSE FLOOR(RAND() * 3)
  END;
SELECT DISTINCT
  dorsal AS ganador
FROM
  ciclistas
WHERE
  etapas_ganadas > 0
ORDER BY
```

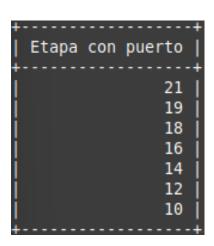


dorsal DESC

LIMIT 16;

4-. Muestra los números de etapa que tienen algún puerto mostrando primero las últimas etapas (7 filas).

```
CREATE TABLE IF NOT EXISTS etapas (
  numero INT PRIMARY KEY,
  tiene_puerto BOOLEAN DEFAULT FALSE,
  fecha DATE,
  distancia km DECIMAL(5,1)
);
INSERT INTO etapas (numero, tiene_puerto, fecha, distancia_km) VALUES
(1, FALSE, '2023-07-01', 182.5),
(2, TRUE, '2023-07-02', 208.7),
(3, FALSE, '2023-07-03', 185.0),
(4, TRUE, '2023-07-04', 195.3),
(5, FALSE, '2023-07-05', 165.8),
(6, TRUE, '2023-07-06', 220.1),
(7, FALSE, '2023-07-07', 178.4),
(8, TRUE, '2023-07-08', 235.6),
(9, FALSE, '2023-07-09', 168.9),
(10, TRUE, '2023-07-10', 198.2),
(11, FALSE, '2023-07-11', 172.3),
(12, TRUE, '2023-07-12', 225.0),
(13, FALSE, '2023-07-13', 190.5),
(14, TRUE, '2023-07-14', 210.8),
(15, FALSE, '2023-07-15', 175.2),
(16, TRUE, '2023-07-16', 230.4),
(17, FALSE, '2023-07-17', 165.7),
(18, TRUE, '2023-07-18', 218.9),
(19, TRUE, '2023-07-19', 205.3),
(20, FALSE, '2023-07-20', 145.6),
(21, TRUE, '2023-07-21', 185.7);
SELECT
  numero AS 'Etapa con puerto'
FROM
  etapas
WHERE
  tiene_puerto = TRUE
ORDER BY
  numero DESC
LIMIT 7;
```



Condición sencilla

1-. Muestra las columnas que se muestran en el ejemplo de los ciclistas que tienen más de 33 años, ordenado por equipo de A a Z (12 filas):

```
INSERT INTO ciclistas (nombre, edad, equipo) VALUES
('Claudio Chioccioli', 36, 'Amore Vita'),
('Pedro Delgado', 35, 'Banesto'),
('Julian Gorospe', 34, 'Banesto'),
('Bruno Leali', 37, 'Bresciali-Refin'),
('Stefan Roche', 36, 'Carrera'),
('Laurent Fignon', 35, 'Gatorade'),
('Marco Giovannetti', 34, 'Gatorade'),
('Antonio Esparza', 35, 'Kelme'),
('Federico Echave', 37, 'Mapei-Class'),
('Massimo Podenzana', 34, 'Navigare'),
('Gerd Audehm', 34, 'Telecom'),
('Robert Millar', 37, 'TVM'),
('Stephen Roche', 35, 'Carrera'),
('Sean Kelly', 38, 'Festina');
SELECT
  nombre AS 'nombre y apellido',
  edad AS 'edad',
  equipo AS 'nomequipo'
FROM
  ciclistas
WHERE
  edad > 33
ORDER BY
  equipo ASC
LIMIT 12;
```

nombre y apellido	edad	nomequipo
Robert Lewandowski	36	Amore Vita
Claudio Chioccioli	36	Amore Vita
Pedro Delgado	35	Banesto
Julian Gorospe	34	Banesto
Zlatan Ibrahimović	37	Bresciali-Refin
Bruno Leali	37	Bresciali-Refin
Stephen Roche	35	Carrera
Gavi	35	Carrera
Stefan Roche	36	Carrera
Sean Kelly	38	Festina
Pedri	38	Festina
Laurent Fignon	35	Gatorade
+		++

2-. Muestra el número de etapa, los kilómetros y la ciudad de salida y llegada de las etapas circulares, mostrando primero las etapas más largas (3 filas):

```
CREATE TABLE IF NOT EXISTS etapas (
  numero INT PRIMARY KEY,
  kms DECIMAL(5,1) NOT NULL,
  ciudad salida VARCHAR(50) NOT NULL,
  ciudad Ilegada VARCHAR(50) NOT NULL
);
INSERT INTO etapas (numero, kms, ciudad_salida, ciudad_llegada) VALUES
(1, 9, 'Valladolid', 'Valladolid'),
(2, 182, 'Burgos', 'Leon'),
(3, 208, 'Pamplona', 'Zaragoza'),
(8, 40, 'Benidorm', 'Benidorm'),
(18, 195, 'Avila', 'Avila'),
(20, 120, 'Madrid', 'Toledo');
SELECT
  numero AS numetapa,
  kms,
  ciudad_salida AS salida,
  ciudad_llegada AS llegada
FROM
  etapas
WHERE
  ciudad_salida = ciudad_llegada -- Condición de etapa circular
ORDER BY
  kms DESC -- Ordenar por distancia descendente
LIMIT 3;
```

numetapa	+	+	++
	kms	salida	llegada
18 8 1	•	Benidorm	Avila Benidorm Valladolid

3-. Selecciona los puertos, su altura y categoría que tienen más de 1500 metros de altura, ordenados por los más altos primero y en caso de igualdad por categoría (10 filas).

```
CREATE TABLE IF NOT EXISTS puertos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombrePuerto VARCHAR(100) NOT NULL,
  altura INT NOT NULL,
  categoria VARCHAR(2) NOT NULL
);
INSERT INTO puertos (nombrePuerto, altura, categoria) VALUES
('Cerler-Circo de Ampriu', 2500, 'E'),
('Sierra Nevada', 2500, 'E'),
('Arcalis', 2230, 'E'),
('Coll de Ordino', 1980, 'E'),
('Navacerrada', 1860, '1'),
('Cruz de la Demanda', 1850, 'E'),
('Puerto de la Morcuera', 1760, '2'),
('Coll de la Comella', 1632, '1'),
('Puerto de Mijares', 1525, '1'),
('Puerto de Navalmoral', 1521, '2'),
('Otro Puerto', 1400, '1');
SELECT
  nombrePuerto,
  altura,
  categoria
FROM
  puertos
WHERE
  altura > 1500
ORDER BY
  altura DESC,
  categoria ASC
LIMIT 10;
```

+	+	++
nombrePuerto	l altura I	categoria
i ilombi di dei co	accura	carrego. za
+	+	*************
Cerler-Circo de Ampriu	2500	E
Sierra Nevada	i 2500 i	E İ
	,	- !
Arcalis	2230	E
Coll de Ordino	1980	E
Navacerrada	1860	1
Cruz de la Demanda	1850	E i
Puerto de la Morcuera	1760	2
Coll de la Comella	1632	1
Puerto de Mijares	1525	1
Puerto de Navalmoral	1521	2
+		

Condición like

1-. Selecciona el nombre y el equipo de los ciclistas que tienen un nombre de 4 caracteres (independientemente del apellido) y cuyo nombre esté alfabéticamente antes que el nombre del equipo (9 filas).

```
CREATE TABLE IF NOT EXISTS ciclistas (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  equipo VARCHAR(50) NOT NULL
);
INSERT INTO ciclistas (nombre, equipo) VALUES
('Alex Zulle', 'ONCE'),
('Jean Van Poppel', 'Lotus Festina'),
('Lale Cubino', 'Seguros Amaya'),
('Erik Dekker', 'Wordperfect'),
('Gian Matteo Fagnini', 'Mercatone Uno'),
('Rolf Aldag', 'Telecom'),
('Gerd Audehm', 'Telecom'),
('Juan Romero', 'Seguros Amaya'),
('Juan Martinez Oliver', 'Kelme'),
('Luis Perez', 'Movistar'),
('Ivan Garcia', 'Banesto');
SELECT
  nombre AS nombre.
  equipo AS nomequipo
FROM
  ciclistas
WHERE
  LENGTH(SUBSTRING_INDEX(nombre, '', 1)) = 4 AND
  SUBSTRING_INDEX(nombre, ' ', 1) COLLATE utf8mb4_general_ci < equipo
COLLATE utf8mb4 general ci
```

ORDER BY
nombre

LIMIT 9;

| nombre | nomequipo |
| Alex Zulle | ONCE |

nombre	nomequipo
Alex Zulle Erik Dekker Gerd Audehm Gerd Audehm Gian Matteo Fagnini Jean Van Poppel Juan Martinez Oliver Juan Romero Lale Cubino	ONCE Wordperfect Telecom Telecom Mercatone Uno Lotus Festina Kelme Seguros Amaya Seguros Amaya

2-. Muestra el dorsal y nombre de los ciclistas cuyo apellido empieza por C y tienen menos de 29 años o que comience por A y tengan más de 29 (13 filas):

```
CREATE TABLE ciclistas (
  id INT AUTO INCREMENT PRIMARY KEY,
  dorsal INT NOT NULL,
  nombre VARCHAR(100) NOT NULL,
  edad INT.
  equipo VARCHAR(50)
);
INSERT INTO ciclistas (dorsal, nombre, edad, equipo) VALUES
(10, 'Mario Cipollini', 28, 'Saeco'),
(13, 'Lale Cubino', 29, 'Seguros Amaya'),
(31, 'Vicente Aparicio', 30, 'Banesto'),
(43, 'Francesco Casagrande', 27, 'Mercatone Uno'),
(58, 'Gerd Audehm', 32, 'Telecom'),
(61, 'Walte Castignola', 26, 'Bresciali'),
(62, 'Raul Alcala', 31, 'PDM'),
(67, 'Armand de las Cuevas', 28, 'Castorama'),
(68, 'Angel Citracca', 33, 'ZG Mobili'),
(72, 'Stefano Colage', 27, 'Amore Vita'),
(85, 'Dimitri Abdoujaparov', 34, 'Carrera'),
(88, 'Angel Camarillo', 29, 'Kelme'),
(94, 'Marino Alonso', 35, 'ONCE');
SELECT
  dorsal.
  nombre
FROM
  ciclistas
WHERE
  (SUBSTRING INDEX(nombre, ' ', -1) LIKE 'C%' AND edad < 29) OR
  (SUBSTRING INDEX(nombre, '', -1) LIKE 'A%' AND edad > 29)
ORDER BY
  dorsal
LIMIT 13;
                              dorsal | nombre
                                   10 | Mario Cipollini
                                   31 I
                                        Vicente Aparicio
                                   43 | Francesco Casagrande
                                   58 |
                                        Gerd Audehm
                                   61
                                        Walte Castignola
                                   62
                                        Raul Alcala
                                   67
                                        Armand de las Cuevas
                                        Stefano Colage
                                   72
                                        Dimitri Abdoujaparov
```

85

| Marino Alonso

3-. Mostrar el dorsal, nombre y edad de los ciclistas que tienen un dorsal que es el doble que su edad, o que tanto el nombre como el apellido contiene una U. Mostrar primero los de mayor edad. (7 filas)

```
CREATE TABLE ciclistas (
  dorsal INT PRIMARY KEY,
  nombre VARCHAR(50),
  edad INT.
  nomequipo VARCHAR(50)
);
INSERT INTO ciclistas (dorsal, nombre, edad, nomequipo) VALUES
(1, 'Miguel Indurain', 32, 'Banesto'),
(95, 'Manuel Guijarro', 31, 'Lotus Festina'),
(24, 'Claudio Chiappucci', 29, 'Carrera'),
(32, 'Laurent Dufaux', 28, 'ONCE'),
(71, 'Prudencio Indurin', 27, 'Banesto'),
(52, 'Vladislav Bobrik', 26, 'Gewiss'),
(98, 'Eleuterio Anguita', 25, 'Artiach');
SELECT dorsal, nombre, edad
FROM ciclistas
WHERE dorsal = edad * 2
 OR (LOWER(nombre) LIKE '%u%' AND
    LOWER(SUBSTRING_INDEX(nombre, ' ', -1)) LIKE '%u%')
ORDER BY edad DESC;
```

+	+	++
dorsal	nombre	edad
+	+	
1	Miguel Indurain	32
95	Manuel Guijarro	31
24	Claudio Chiappucci	29
32	Laurent Dufaux	28
j 71	Prudencio Indurin	27
52	Vladislav Bobrik	26
j 98	Eleuterio Anguita	25
+	+	++

4-. Muestra los puertos que no llevan "Puerto" en su nombre y tienen al menos tres palabras (6 filas)

```
CREATE TABLE puertos (
nombrepuerto VARCHAR(100)
);

INSERT INTO puertos (nombrepuerto) VALUES
('Alto del Naranco'),
('Cerler-Circo de Ampriu'),
('Coll de la Comella'),
('Coll de Ordino'),
('Cruz de la Demanda'),
('Lagos de Covadonga');

SELECT nombrepuerto
FROM puertos
WHERE nombrepuerto NOT LIKE '%Puerto%'
AND (LENGTH(nombrepuerto) - LENGTH(REPLACE(nombrepuerto, ' ', ")) + 1) >= 3;
```

5-. En la base de datos World. Muestra todas las distintas regiones de cada continente que comience por A, en orden alfabético por nombre de continente y de región (10 filas).

```
CREATE DATABASE world:
USE world;
CREATE TABLE country (
  Code CHAR(3) PRIMARY KEY,
  Name VARCHAR(50),
  Continent VARCHAR(50),
  Region VARCHAR(50)
);
INSERT INTO country (Code, Name, Continent, Region) VALUES
('AFG', 'Afghanistan', 'Asia', 'Southern and Central Asia'),
('ARM', 'Armenia', 'Asia', 'Middle East'),
('AZE', 'Azerbaijan', 'Asia', 'Middle East'),
('AUS', 'Australia', 'Australia', 'Australia and New Zealand'),
('AUT', 'Austria', 'Europe', 'Western Europe'),
('AGO', 'Angola', 'Africa', 'Central Africa'),
('EGY', 'Egypt', 'Africa', 'Northern Africa'),
('DZA', 'Algeria', 'Africa', 'Northern Africa'),
('ATA', 'Antarctica', 'Antarctica', 'Antarctica'),
('IND', 'India', 'Asia', 'Southern and Central Asia'),
('IDN', 'Indonesia', 'Asia', 'Southeast Asia'),
('NPL', 'Nepal', 'Asia', 'Southern and Central Asia'),
('BGD', 'Bangladesh', 'Asia', 'Southern and Central Asia');
SELECT DISTINCT Region, Continent
FROM country
WHERE Continent LIKE 'A%'
ORDER BY Continent, Region
LIMIT 10;
```

```
Region | Continent |
| Central Africa | Africa |
| Northern Africa | Antarctica |
| Antarctica | Asia |
| Southeast Asia | Asia |
| Southern and Central Asia | Asia |
| Australia and New Zealand | Australia |
| 7 rows in set (0 00 sec)
```