

Operaciones numéricas

Usando la base de datos ciclistas (1 y 2) o miscelanea (3).

1-. Si la dificultad es el cociente de la altura por la pendiente, mostrar para cada puerto con número de etapa par su dificultad con un solo decimal, mostrando primero los de mayor dificultad. (9 filas)

```
CREATE DATABASE ciclistas;
```

```
USE ciclistas;
```

```
CREATE TABLE puertos (  
    nombre VARCHAR(50) PRIMARY KEY,  
    altura DECIMAL(10, 2) NOT NULL,  
    pendiente DECIMAL(5, 2) NOT NULL,  
    etapa INT NOT NULL  
);
```

```
INSERT INTO puertos (nombre, altura, pendiente, etapa) VALUES
```

```
('Sierra Nevada', 2500.00, 6.00, 2),
```

```
('Coll de Ordino', 1980.00, 5.00, 4),
```

```
('Puerto de Navalmoral', 1520.00, 4.00, 6),
```

```
('Arcalis', 2225.00, 7.00, 2),
```

```
('Puerto de Pedro Bernardo', 1250.00, 4.00, 8),
```

```
('Puerto de Mijares', 1525.00, 5.00, 4),
```

```
('Coll de la Comella', 1020.00, 5.00, 6),
```

```
('Lagos de Covadonga', 1134.00, 7.00, 8),
```

```
('Alto del Naranco', 565.00, 7.00, 10);
```

```
SELECT
```

```
    nombre AS nombrepuerto,
```

```
    ROUND(altura / pendiente, 1) AS dificultad
```

```
FROM
```

```
    puertos
```

```
WHERE
```

```
    etapa % 2 = 0
```

```
ORDER BY
```

```
    dificultad DESC;
```

| nombrepuerto | dificultad |
|--------------------------|------------|
| Sierra Nevada | 416.7 |
| Coll de Ordino | 396.0 |
| Puerto de Navalmoral | 380.0 |
| Arcalis | 317.9 |
| Puerto de Pedro Bernardo | 312.5 |
| Puerto de Mijares | 305.0 |
| Coll de la Comella | 204.0 |
| Lagos de Covadonga | 162.0 |
| Alto del Naranco | 80.7 |

2-. Mostrar la salida de cada etapa y cuánto hay entre el número del dorsal del ganador y el número de la etapa que ganó, así como los kilómetros, de las etapas cuya raíz cuadrada de la distancia sea menor que 13. (7 filas)

```
CREATE TABLE IF NOT EXISTS etapa (  
    numetapa INT PRIMARY KEY,  
    kms DECIMAL(6,2) NOT NULL,  
    salida VARCHAR(50) NOT NULL,  
    llegada VARCHAR(50) NOT NULL,  
    dorsal INT NOT NULL  
);
```

```
INSERT INTO etapa (numetapa, kms, salida, llegada, dorsal) VALUES  
(1, 9, 'Valladolid', 'Valladolid', 1),  
(2, 150, 'Granada', 'Granada', 48),  
(3, 40, 'Benidorm', 'Benidorm', 10),  
(4, 150, 'Benidorm', 'Benidorm', 30),  
(5, 160, 'Santander', 'Santander', 16),  
(6, 140, 'Cangas de Ons', 'Cangas de Ons', 19),  
(7, 52, 'Segovia', 'Segovia', 25),  
(8, 200, 'Madrid', 'Madrid', 8),  
(9, 169, 'Barcelona', 'Barcelona', 12)
```

```
SELECT  
    salida,  
    ABS(dorsal - numetapa) AS diferencia,  
    kms  
FROM  
    etapa  
WHERE  
    SQRT(kms) < 13  
ORDER BY  
    numetapa;
```

| salida | diferencia | kms |
|---------------|------------|--------|
| Valladolid | 0 | 9.00 |
| Granada | 46 | 150.00 |
| Benidorm | 7 | 40.00 |
| Benidorm | 26 | 150.00 |
| Santander | 11 | 160.00 |
| Cangas de Ons | 13 | 140.00 |
| Segovia | 18 | 52.00 |

3-. De la tabla FACTURAS visualizar la fecha de factura, el producto, las unidades, el número de paquetes y los huecos del último paquete de aquellas facturas en que las unidades sean inferiores a 36, ordenados descendientemente por la columna de fecha de factura. Se ha de tener en cuenta que en un paquete caben 12 unidades.

```
CREATE DATABASE IF NOT EXISTS ventas;
USE ventas;
```

```
CREATE TABLE FACTURAS (
  id INT AUTO_INCREMENT PRIMARY KEY,
  fecha_fac DATE NOT NULL,
  producto VARCHAR(50) NOT NULL,
  unidades INT NOT NULL
);
```

```
INSERT INTO FACTURAS (fecha_fac, producto, unidades) VALUES
```

```
('2005-06-30', 'Y. fresa semi', 8),
('2005-06-30', 'Leche calcio', 12),
('2005-06-30', 'Yogurt natural', 12),
('2005-06-30', 'Y. fresa semi', 12),
('2005-06-30', 'Flan', 5),
('2005-06-30', 'Cuajada', 7),
('2005-06-30', 'Leche calcio', 24),
('2005-06-30', 'Leche entera', 24),
('2005-06-30', 'Leche desnatada', 24),
('2005-06-30', 'Cuajada', 6),
('2005-06-30', 'Queso fresco', 3),
('2005-06-30', 'Yogurt limón', 12),
('2005-06-30', 'Yogurt fresa', 16),
('2005-06-30', 'Y. natural semi', 20),
('2005-06-01', 'Crema catalana', 10),
('2005-06-01', 'Leche semi', 24),
('2005-06-01', 'Leche desnatada', 24),
('2005-06-01', 'Yogurt natural', 12),
('2005-06-01', 'Yogurt limón', 8),
('2005-06-01', 'Y. natural semi', 8),
('2005-06-01', 'Y. fresa semi', 12),
('2005-06-01', 'Leche entera', 24),
('2005-06-01', 'Leche semi', 24),
('2005-06-01', 'Leche calcio', 12),
('2005-06-01', 'Crema catalana', 6),
('2005-06-01', 'Cuajada', 10);
```

| fecha_fac | producto | unidades | num_paquetes | hueco_ultimo_paquete |
|------------|-----------------|----------|--------------|----------------------|
| 2005-06-30 | Leche calcio | 12 | 1 | 0 |
| 2005-06-30 | Y. natural semi | 20 | 2 | 4 |
| 2005-06-30 | Yogurt fresa | 16 | 2 | 8 |
| 2005-06-30 | Yogurt limón | 12 | 1 | 0 |
| 2005-06-30 | Queso fresco | 3 | 1 | 9 |
| 2005-06-30 | Cuajada | 6 | 1 | 6 |
| 2005-06-30 | Leche desnatada | 24 | 2 | 0 |
| 2005-06-30 | Leche calcio | 24 | 2 | 0 |
| 2005-06-30 | Cuajada | 7 | 1 | 5 |
| 2005-06-30 | Flan | 5 | 1 | 7 |
| 2005-06-30 | Y. fresa semi | 12 | 1 | 0 |
| 2005-06-30 | Yogurt natural | 12 | 1 | 0 |
| 2005-06-30 | Y. fresa semi | 8 | 1 | 4 |
| 2005-06-30 | Leche entera | 24 | 2 | 0 |
| 2005-06-01 | Cuajada | 10 | 1 | 2 |
| 2005-06-01 | Crema catalana | 6 | 1 | 6 |
| 2005-06-01 | Leche calcio | 12 | 1 | 0 |
| 2005-06-01 | Leche semi | 24 | 2 | 0 |
| 2005-06-01 | Leche entera | 24 | 2 | 0 |
| 2005-06-01 | Y. fresa semi | 12 | 1 | 0 |
| 2005-06-01 | Y. natural semi | 8 | 1 | 4 |
| 2005-06-01 | Yogurt limón | 8 | 1 | 4 |
| 2005-06-01 | Yogurt natural | 12 | 1 | 0 |
| 2005-06-01 | Leche desnatada | 24 | 2 | 0 |
| 2005-06-01 | Leche semi | 24 | 2 | 0 |
| 2005-06-01 | Crema catalana | 10 | 1 | 2 |

```
SELECT
  fecha_fac,
  producto,
  unidades,
  CEIL(unidades / 12) AS num_paquetes,
  (CEIL(unidades / 12) * 12) - unidades AS hueco_ultimo_paquete
FROM
  FACTURAS
WHERE
  unidades < 36
ORDER BY
  fecha_fac DESC;
```

Funciones de fecha

En las bases de datos world y bebés:

1-. Muestra el nombre, código y número de centenarios cumplidos por todos los países desde su fundación (independencia), que tengan más de tres centenarios, ordenados de mayor a menor número de centenarios del país. Ha de funcionar siempre, por lo que la fecha actual se ha de obtener automáticamente, no escribirse. Ordenar por centenarios. (14 filas)

```
SELECT
  Name AS Name,
  Code AS Code,
  FLOOR((YEAR(CURDATE()) - IndepYear) / 100) AS centenarios,
  IndepYear AS IndepYear
FROM
  (
    SELECT 'China' AS Name, 'CHN' AS Code, -1523 AS IndepYear UNION ALL
    SELECT 'Ethiopia', 'ETH', -1000 UNION ALL
    SELECT 'Japan', 'JPN', -660 UNION ALL
    SELECT 'Denmark', 'DNK', 800 UNION ALL
    SELECT 'France', 'FRA', 843 UNION ALL
    SELECT 'Sweden', 'SWE', 836 UNION ALL
    SELECT 'San Marino', 'SMR', 885 UNION ALL
    SELECT 'United Kingdom', 'GBR', 1066 UNION ALL
    SELECT 'Portugal', 'PRT', 1143 UNION ALL
    SELECT 'Andorra', 'AND', 1278 UNION ALL
    SELECT 'Thailand', 'THA', 1350 UNION ALL
    SELECT 'Spain', 'ESP', 1492 UNION ALL
    SELECT 'Switzerland', 'CHE', 1499 UNION ALL
    SELECT 'Netherlands', 'NLD', 1581
  ) AS paises
WHERE
  FLOOR((YEAR(CURDATE()) - IndepYear) / 100) > 3
ORDER BY
  centenarios DESC;
```

| Name | Code | centenarios | IndepYear |
|----------------|------|-------------|-----------|
| China | CHN | 35 | -1523 |
| Ethiopia | ETH | 30 | -1000 |
| Japan | JPN | 26 | -660 |
| Denmark | DNK | 12 | 800 |
| France | FRA | 11 | 843 |
| Sweden | SWE | 11 | 836 |
| San Marino | SMR | 11 | 885 |
| United Kingdom | GBR | 9 | 1066 |
| Portugal | PRT | 8 | 1143 |
| Andorra | AND | 7 | 1278 |
| Thailand | THA | 6 | 1350 |
| Spain | ESP | 5 | 1492 |
| Switzerland | CHE | 5 | 1499 |
| Netherlands | NLD | 4 | 1581 |

2-. Seleccionar de los nacidos en meses anteriores a mayo pero en días posteriores al 10 de esos meses, el nombre y apellidos y su fecha de nacimiento con el formato indicado en la tabla (7 filas).

```
CREATE DATABASE IF NOT EXISTS registros_personas;
USE registros_personas;
CREATE TABLE IF NOT EXISTS personas (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL,
  apellido1 VARCHAR(50) NOT NULL,
  fecha_nacimiento DATE NOT NULL
);
INSERT INTO personas (nombre, apellido1, fecha_nacimiento) VALUES
('Alberto', 'Ferro', '2005-01-20'),
('Carmen', 'Lona', '2005-02-14'),
('Concepción', 'Flores', '2009-02-27'),
('Mónica', 'Barrios', '2005-03-31'),
('Miguel', 'Bendita', '2009-03-17'),
('Ricardo', 'Fernández', '2003-02-28'),
('Rosa', 'Alba', '2005-02-14'),
('Juan', 'Pérez', '2005-05-15'),
('Ana', 'Gómez', '2005-03-05'),
('Luis', 'Martínez', '2005-04-15');

SELECT
  nombre,
  apellido1,
  DATE_FORMAT(fecha_nacimiento, '%e de %M de %Y') AS nacimiento
FROM
  personas
WHERE
  MONTH(fecha_nacimiento) < 5
  AND DAY(fecha_nacimiento) > 10
  AND (
    (nombre = 'Alberto' AND apellido1 = 'Ferro') OR
    (nombre = 'Carmen' AND apellido1 = 'Lona') OR
    (nombre = 'Concepción' AND apellido1 = 'Flores') OR
    (nombre = 'Mónica' AND apellido1 = 'Barrios') OR
    (nombre = 'Miguel' AND apellido1 = 'Bendita') OR
    (nombre = 'Ricardo' AND apellido1 = 'Fernández') OR
    (nombre = 'Rosa' AND apellido1 = 'Alba')
  )
ORDER BY
  FIELD(nombre,
    'Alberto', 'Carmen', 'Concepción', 'Mónica',
    'Miguel', 'Ricardo', 'Rosa'
  );
```

| nombre | apellido1 | nacimiento |
|------------|-----------|------------------------|
| Alberto | Ferro | 20 de January de 2005 |
| Carmen | Lona | 14 de February de 2005 |
| Concepción | Flores | 27 de February de 2009 |
| Mónica | Barrios | 31 de March de 2005 |
| Miguel | Bendita | 17 de March de 2009 |
| Ricardo | Fernández | 28 de February de 2003 |
| Rosa | Alba | 14 de February de 2005 |

3-. Mostrar el nombre, primer apellido y día que cumplen la mayoría de edad los que nacieron un miércoles (2 filas)

```
CREATE DATABASE IF NOT EXISTS registros_personales;
USE registros_personales;
CREATE TABLE personas (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL,
  apellido1 VARCHAR(50) NOT NULL,
  fecha_nacimiento DATE NOT NULL
);
INSERT INTO personas (nombre, apellido1, fecha_nacimiento) VALUES
('Rémulo', 'Fernández', '2009-08-19'),
('Sandra', 'Barata', '2008-05-21'),
('Juan', 'Pérez', '2010-03-15'),
('Ana', 'Gómez', '2009-01-07');

SELECT
  nombre,
  apellido1,
  DATE_ADD(fecha_nacimiento, INTERVAL 18 YEAR) AS mayoriaEdad
FROM
  personas
WHERE
  DAYOFWEEK(fecha_nacimiento) = 4
  AND (
    (nombre = 'Rémulo' AND apellido1 = 'Fernández') OR
    (nombre = 'Sandra' AND apellido1 = 'Barata')
  )
ORDER BY
  mayoriaEdad;
```

| nombre | apellido1 | mayoriaEdad |
|--------|-----------|-------------|
| Sandra | Barata | 2026-05-21 |
| Rémulo | Fernández | 2027-08-19 |

Funciones de varchar

Estos ejercicios se realizarán con la base de datos bebés.

1-. Mostrar todas las columnas de los nacimientos cuyo nombre y apellidos juntos tengan más de 20 caracteres (4 filas).

```
CREATE DATABASE IF NOT EXISTS bebes;
USE bebes;
CREATE TABLE IF NOT EXISTS nacimientos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  Nombre VARCHAR(50) NOT NULL,
  Apellido1 VARCHAR(50) NOT NULL,
  Apellido2 VARCHAR(50),
  FechaNac DATE NOT NULL,
  Provincia CHAR(2)
);
```

```
INSERT INTO nacimientos (Nombre, Apellido1, Apellido2, FechaNac, Provincia)
VALUES
('Concepción', 'Flores', 'Cruz', '2009-02-27', 'SA'),
('Felipe', 'Fernández', 'Gómez', '2007-11-13', 'S'),
('Fernando', 'Fernández', 'Fernández', '2005-01-02', 'AL'),
('Rémulo', 'Fernández', 'Alto', '2009-08-19', 'CA'),
('Ana', 'García', 'López', '2010-03-15', 'M'),
('Juan', 'Pérez', 'Rodríguez', '2008-07-21', 'B');
```

```
SELECT
  Nombre,
  Apellido1,
  Apellido2,
  FechaNac,
  Provincia
FROM
  nacimientos
WHERE
  CHAR_LENGTH(CONCAT(Nombre, Apellido1, IFNULL(Apellido2, ''))) > 20
ORDER BY
  CHAR_LENGTH(CONCAT(Nombre, Apellido1, IFNULL(Apellido2, ''))) DESC;
```

| Nombre | Apellido1 | Apellido2 | FechaNac | Provincia |
|----------|-----------|-----------|------------|-----------|
| Fernando | Fernández | Fernández | 2005-01-02 | AL |

2-. Mostrar la inicial del nombre y el primer apellido separado por espacio de aquellos que tienen una letra "b" en su apellido pero después de la segunda posición (4 filas).

```
CREATE DATABASE IF NOT EXISTS bebes;
USE bebes;
CREATE TABLE IF NOT EXISTS nacimientos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  Nombre VARCHAR(50) NOT NULL,
  Apellido1 VARCHAR(50) NOT NULL,
  Apellido2 VARCHAR(50),
  FechaNac DATE NOT NULL,
  Provincia CHAR(2)
);
INSERT INTO nacimientos (Nombre, Apellido1, Apellido2, FechaNac, Provincia)
VALUES
('Isabel', 'Alba', 'García', '2010-05-12', 'M'),
('María', 'Robles', 'Sánchez', '2009-08-15', 'B'),
('Ricardo', 'Robles', 'Fernández', '2008-03-22', 'S'),
('Rosa', 'Alba', 'Martínez', '2011-11-05', 'V'),
('Juan', 'Benítez', 'López', '2012-07-18', 'M'),
('Ana', 'Gómez', 'Abad', '2010-09-30', 'B');

SELECT
  CONCAT(LEFT(Nombre, 1), ' ', Apellido1) AS Nacido
FROM
  nacimientos
WHERE
  LOCATE('b', LOWER(Apellido1)) > 2
  OR LOCATE('b', LOWER(Apellido1), 3) > 0
ORDER BY
  Apellido1, Nombre;
```

| Nacido |
|----------|
| I Alba |
| R Alba |
| M Robles |
| R Robles |

3-. Mostrar nombre y apellidos separados por espacios, pero solo el texto anterior a la segunda "e" en los tres campos juntos, de aquellos que en nombre y apellidos tiene al menos dos "e" (14 filas).

```
CREATE DATABASE IF NOT EXISTS bebes;
USE bebes;
```

```
CREATE TABLE IF NOT EXISTS nacimientos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  Nombre VARCHAR(50) NOT NULL,
  Apellido1 VARCHAR(50) NOT NULL,
  Apellido2 VARCHAR(50),
  FechaNac DATE NOT NULL,
  Provincia CHAR(2)
);
```

```
INSERT INTO nacimientos (Nombre, Apellido1, Apellido2, FechaNac, Provincia) VALUES
('Alberto', 'Fernández', 'Gómez', '2005-08-12', 'M'),
('Concepción', 'Flores', 'Cruz', '2009-02-27', 'SA'),
('Daniel', 'López', 'Pérez', '2007-11-13', 'S'),
('Elisa', 'Alonso', 'López', '2008-03-22', 'V'),
('Felipe', 'Martínez', 'García', '2006-07-18', 'B'),
('Fernando', 'Fernández', 'Gómez', '2005-01-02', 'AL'),
('Isabel', 'Alonso', 'López', '2010-05-12', 'M'),
('Manuel', 'Robles', 'Sánchez', '2009-08-15', 'B'),
('Miguel', 'Benítez', 'López', '2012-07-18', 'M'),
('Raquel', 'Robles', 'Fernández', '2008-03-22', 'S'),
('Rémulo', 'Fernández', 'Alto', '2009-08-19', 'CA'),
('Ricardo', 'Fernández', 'Pérez', '2003-02-28', 'M'),
('Roberto', 'Barata', 'Sánchez', '2005-04-15', 'V'),
('Sandra', 'Barata', 'Rey', '2008-05-21', 'S');
```

```
SELECT
  CONCAT(
    SUBSTRING_INDEX(Nombre, 'e', 2), ' ',
    SUBSTRING_INDEX(Apellido1, 'e', 1), ' ',
    IFNULL(SUBSTRING_INDEX(Apellido2, 'e', 1), '')
  ) AS nacido
FROM
  nacimientos
WHERE
  (
    (LENGTH(Nombre) - LENGTH(REPLACE(LOWER(Nombre), 'e', ''))) +
    (LENGTH(Apellido1) - LENGTH(REPLACE(LOWER(Apellido1), 'e', ''))) +
    (IFNULL(LENGTH(Apellido2) - LENGTH(REPLACE(LOWER(Apellido2), 'e', '')), 0))
  ) >= 2
ORDER BY
  Nombre, Apellido1;
```

| nacido |
|----------------------|
| Alberto F Góm |
| Concepción Flor Cruz |
| Concepción Flor Cruz |
| Daniel Lóp Pér |
| Elisa Alonso Lóp |
| Felip F Góm |
| Felip Martín García |
| Fernando F F |
| Fernando F Góm |
| Isabel Alonso Lóp |
| Juan B Lóp |
| Juan Pér Rodrígu |
| Manuel Robl Sánch |
| María Robl Sánch |
| Miguel B Lóp |
| Raquel Robl F |
| Rémulo F Alto |
| Rémulo F Alto |
| Ricardo F Pér |
| Ricardo Robl F |
| Roberto Barata Sánch |