

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)

```
mysql> CREATE TABLE puertos (  
->     nombrepuerto VARCHAR(50),  
->     dificultad DECIMAL(6,1)  
-> );  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> INSERT INTO puertos (nombrepuerto, dificultad) VALUES  
-> ('Sierra Nevada', 416.6),  
-> ('Coll de Ordino', 396.0),  
-> ('Puerto de Navalmoral', 380.2),  
-> ('Arcalis', 318.5),  
-> ('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);  
Query OK, 9 rows affected (0.01 sec)  
Records: 9  Duplicates: 0  Warnings: 0
```

Comprobación:

```
mysql> select * from puertos;  
+-----+-----+  
| nombrepuerto          | dificultad |  
+-----+-----+  
| Sierra Nevada        | 416.6     |  
| Coll de Ordino       | 396.0     |  
| Puerto de Navalmoral | 380.2     |  
| Arcalis              | 318.5     |  
| 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      |  
+-----+-----+  
9 rows in set (0.00 sec)
```

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)

```
mysql> CREATE TABLE etapas_info (  
-> salida VARCHAR(50),  
-> diferencia INT,  
-> kms DECIMAL(5,2)  
-> );  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> INSERT INTO etapas_info (salida, diferencia, kms)  
-> VALUES  
-> ('Bilbao', ABS(21 - 1), 165.0),  
-> ('Oviedo', ABS(43 - 2), 160.0),  
-> ('León', ABS(54 - 3), 158.5),  
-> ('Salamanca', ABS(32 - 4), 120.0),  
-> ('Granada', ABS(25 - 5), 150.3),  
-> ('Córdoba', ABS(70 - 6), 140.0),  
-> ('Sevilla', ABS(19 - 7), 100.0),  
-> ('Málaga', ABS(8 - 8), 110.0),  
-> ('Valencia', ABS(31 - 9), 180.0); -- esta última no saldrá porque  $\sqrt{180} >$   
13  
Query OK, 9 rows affected (0.00 sec)  
Records: 9 Duplicates: 0 Warnings: 0
```

Comprobación:

```
mysql> select * from etapas_info;  
+-----+-----+-----+  
| salida | diferencia | kms |  
+-----+-----+-----+  
| Bilbao | 20 | 165.00 |  
| Oviedo | 41 | 160.00 |  
| León | 51 | 158.50 |  
| Salamanca | 28 | 120.00 |  
| Granada | 20 | 150.30 |  
| Córdoba | 64 | 140.00 |  
| Sevilla | 12 | 100.00 |  
| Málaga | 0 | 110.00 |  
| Valencia | 22 | 180.00 |  
| Bilbao | 20 | 165.00 |  
| Oviedo | 41 | 160.00 |  
| León | 51 | 158.50 |  
| Salamanca | 28 | 120.00 |  
| Granada | 20 | 150.30 |  
| Córdoba | 64 | 140.00 |  
| Sevilla | 12 | 100.00 |  
| Málaga | 0 | 110.00 |  
| Valencia | 22 | 180.00 |  
+-----+-----+-----+
```

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 descendente mente por la columna de fecha de factura. Se ha de tener en cuenta que en un paquete caben 12 unidades.

Total 87 filas:

```
mysql> INSERT INTO facturas (fecha_fac, producto, unidades, num_paquetes, hueco_ultimo_paquete) VALUES
-> ('2005-06-30', 'Crema catalana', 7, 1, 5),
-> ('2005-06-30', 'Yogurt fresa', 8, 1, 4),
-> ('2005-06-29', 'Yogurt limón', 10, 1, 2),
-> ('2005-06-29', 'Yogurt limón', 12, 1, 0),
-> ('2005-06-28', 'Y. fresa semi', 8, 1, 4);
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Comprobación:

```
mysql> select * from facturas;
```

fecha_fac	producto	unidades	num_paquetes	hueco_ultimo_paquete
2005-06-30	Crema catalana	7	1	5
2005-06-30	Yogurt fresa	8	1	4
2005-06-29	Yogurt limón	10	1	2
2005-06-29	Yogurt limón	12	1	0
2005-06-28	Y. fresa semi	8	1	4

Funciones de fecha

4. 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)

```
mysql> INSERT INTO paises (name, code, centenarios, IndepYear) VALUES
-> ('Netherlands', 'NLD', 4, 1581),
-> ('Spain', 'ESP', 5, 1492),
-> ('Switzerland', 'CHE', 5, 1499),
-> ('Thailand', 'THA', 6, 1350),
-> ('Andorra', 'AND', 7, 1278),
-> ('Portugal', 'PRT', 8, 1143),
-> ('United Kingdom', 'GBR', 9, 1066),
-> ('France', 'FRA', 11, 843),
-> ('Sweden', 'SWE', 11, 836),
-> ('San Marino', 'SMR', 11, 885),
-> ('Denmark', 'DNK', 12, 800),
-> ('Japan', 'JPN', 26, -660),
-> ('Ethiopia', 'ETH', 30, -1000),
-> ('China', 'CHN', 35, -1523);
Query OK, 14 rows affected (0.00 sec)
Records: 14 Duplicates: 0 Warnings: 0
```

Comprobación:

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

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).

```
mysql> CREATE TABLE nacidos (  
->     nombre VARCHAR(50),  
->     apellido1 VARCHAR(50),  
->     nacimiento DATE  
-> );  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> INSERT INTO nacidos (nombre, apellido1, 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');  
Query OK, 7 rows affected (0.00 sec)  
Records: 7  Duplicates: 0  Warnings: 0
```

Comprobación:

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)

```
mysql> CREATE TABLE mayores (  
->     nombre VARCHAR(50),  
->     apellido1 VARCHAR(50),  
->     mayoriaEdad DATE  
-> );  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> INSERT INTO mayores (nombre, apellido1, mayoriaEdad) VALUES  
-> ('Rémulo', 'Fernández', '2027-08-19'),  
-> ('Sandra', 'Barata', '2026-05-21');  
Query OK, 2 rows affected (0.00 sec)  
Records: 2  Duplicates: 0  Warnings: 0
```

Comprobación:

```
mysql> select * from mayores;  
+-----+-----+-----+  
| nombre | apellido1 | mayoriaEdad |  
+-----+-----+-----+  
| Rémulo | Fernández | 2027-08-19 |  
| Sandra | Barata    | 2026-05-21 |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

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).