Ejercicio 1

Vamos a aplicar buena parte de lo que conocemos para hacer un ejercicio de repaso que haga distintas manipulaciones a una única tabla. Será una tabla que contenga datos de productos: código, nombre, precio y fecha de alta, para que podamos trabajar con datos de texto, numéricos y de tipo fecha.

Los pasos que realizaremos (por si alguien se atreve a intentarlo antes de ver la solución) serán:

- Crear la base de datos
- Comenzar a usarla
- Introducir 3 datos de ejemplo
- Mostrar todos los datos
- Mostrar los datos que tienen un cierto nombre
- Mostrar los datos que comienzan por una cierta inicial
- Ver sólo el nombre y el precio de los que cumplen una condición (precio > 22)
- Ver el precio medio de aquellos cuyo nombre comienza con "Silla"
- Modificar la estructura de la tabla para añadir un nuevo campo: "categoría"
- Dar el valor "utensilio" a la categoría de todos los productos existentes
- Modificar los productos que comienza por la palabra "Silla", para que su categoría sea "silla"
- Ver la lista categorías (sin que aparezcan datos duplicados)
- Ver la cantidad de productos que tenemos en cada categoría

```
CREATE DATABASE productos1;
```

Y comenzar a usarla:

```
USE productos1;
```

Para crear la tabla haríamos:

```
CREATE TABLE productos (
  codigo varchar(3),
  nombre varchar(30),
  precio decimal(6,2),
  fechaalta date,
  PRIMARY KEY (codigo)
);
```

Para introducir varios datos de ejemplo:

```
INSERT INTO productos VALUES ('a01','Afilador', 2.50, '2017-06-
02');
INSERT INTO productos VALUES ('s01','Silla mod. ZAZ', 20, '2017-06-
03');
INSERT INTO productos VALUES ('s02','Silla mod. XAX', 25, '2017-06-
03');
```

Podemos ver todos los datos para comprobar que son correctos:

```
SELECT * FROM productos;
```

y deberíamos obtener

```
+--------+----------------+--------+------------+
| codigo | nombre         | precio | fechaalta  |
+--------+----------------+--------+------------+
| a01    | Afilador       |   2.50 | 2017-06-02 |
| s01    | Silla mod. ZAZ |  20.00 | 2017-06-03 |
| s02    | Silla mod. XAX |  25.00 | 2017-06-03 |
+--------+----------------+--------+------------+
```

Para ver qué productos se llaman "Afilador":

```
SELECT * FROM productos WHERE nombre='Afilador';
+--------+----------+--------+------------+
| codigo | nombre   | precio | fechaalta  |
+--------+----------+--------+------------+
| a01    | Afilador |   2.50 | 2017-06-02 |
+--------+----------+--------+------------+
```

Si queremos saber cuáles comienzan por S:

```
SELECT * FROM productos WHERE nombre LIKE 'S%';
+--------+----------------+--------+------------+
| codigo | nombre         | precio | fechaalta  |
+--------+----------------+--------+------------+
| s01    | Silla mod. ZAZ |  20.00 | 2017-06-03 |
| s02    | Silla mod. XAX |  25.00 | 2017-06-03 |
```

```
+--------+----------------+--------+------------+
```

Si queremos ver cuales tienen un precio superior a 22, y además no deseamos ver todos los campos, sino sólo el nombre y el precio:

```
SELECT nombre, precio FROM productos WHERE precio > 22;
+----------------+--------+
| nombre         | precio |
+----------------+--------+
| Silla mod. XAX |  25.00 |
+----------------+--------+
```

Precio medio de las sillas:

```
SELECT avg(precio) FROM productos WHERE LEFT(nombre,5) = 'Silla';
+-------------+
| avg(precio) |
+-------------+
|   22.500000 |
+-------------+
```

Esto de mirar las primeras letras para saber si es una silla o no... quizá no sea la mejor opción. Parece más razonable añadir un nuevo dato: la "categoría". Vamos a modificar la estructura de la tabla para hacerlo:

```
ALTER TABLE productos ADD categoria varchar(10);
```

Comprobamos qué ha ocurrido con un "select" que muestre todos los datos:

```
SELECT * FROM productos;
+--------+----------------+--------+------------+-----------+
| codigo | nombre         | precio | fechaalta  | categoria |
+--------+----------------+--------+------------+-----------+
| a01    | Afilador       |   2.50 | 2017-06-02 | NULL      |
| s01    | Silla mod. ZAZ |  20.00 | 2017-06-03 | NULL      |
| s02    | Silla mod. XAX |  25.00 | 2017-06-03 | NULL      |
+--------+----------------+--------+------------+-----------+
```

Ahora mismo, todas las categorías tienen el valor NULL, y eso no es muy útil. Vamos a dar el valor "utensilio" a la categoría de todos los productos existentes

```
UPDATE productos SET categoria='utensilio';
```

Y ya que estamos, modificaremos los productos que comienza por la palabra "Silla", para que su categoría sea "silla"

```
UPDATE productos SET categoria='silla' WHERE LEFT(nombre,5) =
'Silla';
+--------+----------------+--------+------------+-----------+
| codigo | nombre         | precio | fechaalta  | categoria |
+--------+----------------+--------+------------+-----------+
| a01    | Afilador       |   2.50 | 2017-06-02 | utensilio |
| s01    | Silla mod. ZAZ |  20.00 | 2017-06-03 | silla     |
| s02    | Silla mod. XAX |  25.00 | 2017-06-03 | silla     |
+--------+----------------+--------+------------+-----------+
```

Para ver la lista categorías (sin que aparezcan datos duplicados), deberemos usar la palabra "distinct"

```
SELECT DISTINCT categoria FROM productos;
+-----------+
| categoria |
+-----------+
| utensilio |
| silla     |
+-----------+
```

Finalmente, para ver la cantidad de productos que tenemos en cada categoría, deberemos usar "count" y agrupar los datos con "group by", así:

```
SELECT categoria, count(*) FROM productos GROUP BY categoria;
+-----------+----------+
| categoria | count(*) |
+-----------+----------+
| silla     |        2 |
| utensilio |        1 |
+-----------+----------+
```