
	Bases de datos Tema 6		Práctica
	El lenguaje SQL II – Tratamiento de datos		1

Tarea 1: Creación base de datos.

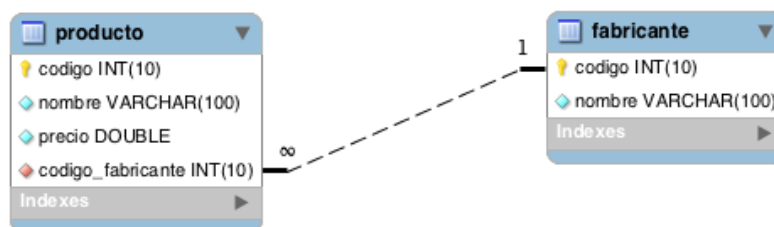
Crear una base de datos llamada **tema6practica1**

A continuación crea la estructura de las tablas de la base de datos según los siguientes datos:

fabricante, motor InnoDB	
codigo	Valor entero sin signo, autoincremental y clave primaria
nombre	Cadena de caracteres, tamaño 100, no nulo

producto, motor InnoDB	
codigo	Valor entero sin signo, autoincremental y clave primaria
Nombre	Cadena de caracteres, tamaño 100, no nulo
Precio	Doble no nulo
codigo_fabricante	Valor entero sin signo no nulo y clave foránea del campo codigo de la tabla fabricante



El diagrama es el siguiente:



Especificar la correspondiente sentencia SQL necesaria para realizar cada uno de los siguientes apartados (usar la fuente **Monaco**). Obtener una captura de al menos las 10 primeras filas de cada una de las consultas realizadas.

Ejemplo:

```
SELECT * FROM Personas;
```

	Bases de datos Tema 6		Práctica
	El lenguaje SQL II – Tratamiento de datos		1

Comandos usados para la creación de la base de datos y tablas:



```
create database tema6practical
Use tema6practical
create table fabricante (
    -> codigo int unsigned auto_increment primary key,
    -> nombre varchar(100) not null);
```

Tarea 2: Importar datos.

Crear un fichero **.sql**, llamado *p1t6.sql* e inserta las siguientes líneas. Luego ejecuta el script.

```
INSERT INTO fabricante VALUES(1, 'Asus');
INSERT INTO fabricante VALUES(2, 'Lenovo');
INSERT INTO fabricante VALUES(3, 'Hewlett-Packard');
INSERT INTO fabricante VALUES(4, 'Samsung');
INSERT INTO fabricante VALUES(5, 'Seagate');
INSERT INTO fabricante VALUES(6, 'Crucial');
INSERT INTO fabricante VALUES(7, 'Gigabyte');
INSERT INTO fabricante VALUES(8, 'Huawei');
INSERT INTO fabricante VALUES(9, 'Xiaomi');

INSERT INTO producto VALUES(1, 'Disco duro SATA3 1TB', 86.99, 5);
INSERT INTO producto VALUES(2, 'Memoria RAM DDR4 8GB', 120, 6);
INSERT INTO producto VALUES(3, 'Disco SSD 1 TB', 150.99, 4);
INSERT INTO producto VALUES(4, 'GeForce GTX 1050Ti', 185, 7);
INSERT INTO producto VALUES(5, 'GeForce GTX 1080 Xtreme', 755, 6);
INSERT INTO producto VALUES(6, 'Monitor 24 LED Full HD', 202, 1);
INSERT INTO producto VALUES(7, 'Monitor 27 LED Full HD', 245.99, 1);
INSERT INTO producto VALUES(8, 'Portátil Yoga 520', 559, 2);
INSERT INTO producto VALUES(9, 'Portátil Ideapd 320', 444, 2);
INSERT INTO producto VALUES(10, 'Impresora HP Deskjet 3720', 59.99, 3);
INSERT INTO producto VALUES(11, 'Impresora HP Laserjet Pro M26nw', 180, 3);
```



 salesianos COLEGIO SAN JUAN BOSCO LA CUESTA	Bases de datos Tema 6		 UNIVERSIDAD DE BURGOS	Práctica
	El lenguaje SQL II – Tratamiento de datos			1

1. Mostrar los datos introducidos en la tabla fabricante.

	codigo	nombre
▶	1	Asus
	2	Lenovo
	3	Hewlett-Packard
	4	Samsung
	5	Seagate
	6	Crucial
	7	Gigabyte
	8	Huawei
	9	Xiaomi
✱	NULL	NULL

2. Mostrar los datos introducidos en la tabla producto.

	codigo	nombre	precio	codigo_fabricante
▶	1	Disco duro SATA3 1TB	86.99	5
	2	Memoria RAM DDR4 8GB	120	6
	3	Disco SSD 1 TB	150.99	4
	4	GeForce GTX 1050Ti	185	7
	5	GeForce GTX 1080 Xtreme	755	6
	6	Monitor 24 LED Full HD	202	1
	7	Monitor 27 LED Full HD	245.99	1
	8	Portátil Yoga 520	559	2
	9	Portátil Ideapd 320	444	2
	10	Impresora HP Deskjet 3720	59.99	3
	11	Impresora HP Laserjet Pro...	180	3
✱	NULL	NULL	NULL	NULL

 salesianos COLEGIO SAN JUAN BOSCO LA CUESTA	Bases de datos Tema 6	 UNIVERSIDAD DE BURGOS	Práctica
	El lenguaje SQL II – Tratamiento de datos		1

Tarea 2: Tratamiento de datos.

3. Insertar un nuevo fabricante indicando su código y su nombre.

```
INSERT INTO fabricante VALUES(10, 'Logitech');
```

4. Insertar un nuevo fabricante indicando solamente su nombre.

```
INSERT INTO fabricante (nombre) VALUES( 'LG');
```

5. Insertar un nuevo producto asociado a uno de los nuevos fabricantes. La sentencia de inserción debe incluir: codigo, nombre, precio y codigo_fabricante

```
INSERT producto (codigo, nombre, precio, codigo_fabricante)
VALUES(12, 'ratón G705', 94, 10);
```



6. Insertar un nuevo producto asociado a uno de los nuevo fabricantes. La sentencia de inserción debe incluir: nombre, precio y codigo_fabricante.

```
INSERT producto (nombre, precio, codigo_fabricante)
VALUES('TV 4K OLED', 1601.85, 11);
```

7. Crear una nueva tabla con el nombre fabricante_productos que tenga las siguientes columnas: nombre_fabricante, nombre_producto y precio. Una vez creada la tabla inserta todos los registros de la base de datos tema6practica1 en esta tabla haciendo uso de única operación de inserción.

```
CREATE TABLE fabricante_productos(
-> nombre_fabricante VARCHAR(100) not null,
-> nombre_producto VARCHAR(100) not null,
-> precio DOUBLE not null);

INSERT fabricante_productos (nombre_fabricante,
nombre_producto, precio)
-> SELECT f.nombre,p.nombre, p.precio
-> FROM fabricante f
-> INNER JOIN producto p
-> ON f.codigo = p.codigo_fabricante;
```

 salesianos COLEGIO SAN JUAN BOSCO LA CUESTA	Bases de datos Tema 6		 UNIVERSIDAD DE BURGOS	Práctica
	El lenguaje SQL II – Tratamiento de datos			1

8. Crea una vista con el nombre *vista_fabricante_productos* que tenga las siguientes columnas: nombre_fabricante, nombre_producto y precio.

```
CREATE VIEW vista_fabricante_productos AS
SELECT nombre_fabricante, nombre_producto, precio
FROM fabricante_productos;
```

9. Elimina el fabricante Asus. ¿Es posible eliminarlo? Si no fue posible, ¿qué cambios debería realizar para que fuese posible borrarlo?

```
# Para eliminar correctamente el fabricante Asus desactivamos
el modo seguro de la base de datos, posteriormente eliminamos
todos los productos del fabricante, tras esto eliminamos al
fabricante de la tabla fabricantes y activamos el modo
seguro.
```

```
SET SQL_SAFE_UPDATES = 0;
DELETE FROM producto WHERE codigo_fabricante = (SELECT codigo
FROM fabricante WHERE nombre = 'Asus');
DELETE FROM fabricante WHERE nombre = "Asus";
SET SQL_SAFE_UPDATES = 1;
```



10. Elimina el fabricante Xiaomi. ¿Es posible eliminarlo? Si no fuese posible, ¿qué cambios deberías realizar para que fuese posible borrarlo?

```
# No existe ningún problema pues no hay entradas en la tabla
producto que contengan como codigo_fabricante el código
referente al fabricante Xiaomi
```

```
SET SQL_SAFE_UPDATES = 0;
DELETE FROM fabricante WHERE nombre = "Xiaomi";
SET SQL_SAFE_UPDATES = 1;
```

11. Actualiza el código del fabricante Lenovo y asígnale el valor 20. ¿Es posible actualizarlo? Si no fuese posible, ¿qué cambios debería realizar para que fuese posible actualizarlo?

```
SET SQL_SAFE_UPDATES = 0;
ALTER TABLE producto DROP FOREIGN KEY cod_fab;
```

	Bases de datos Tema 6		Práctica
	El lenguaje SQL II – Tratamiento de datos		1

```
UPDATE fabricante SET codigo = 20 WHERE nombre = "Lenovo";

SELECT * FROM producto WHERE codigo_fabricante NOT IN (SELECT
codigo FROM fabricante);

UPDATE producto SET codigo_fabricante = 20 WHERE
codigo_fabricante NOT IN (SELECT codigo FROM fabricante);

ALTER TABLE producto ADD CONSTRAINT cod_fab FOREIGN KEY
(codigo_fabricante) REFERENCES fabricante(codigo);

SET SQL_SAFE_UPDATES = 1;
```

- 12.** Actualizar el código del fabricante Huawei y asígnele el valor 30. ¿Es posible actualizarlo? Si no fuese posible, ¿qué cambios debería realizar para que fuese posible actualizarlo?

```
SET SQL_SAFE_UPDATES = 0;
UPDATE fabricante SET codigo = 30 WHERE nombre = "Huawei";
SET SQL_SAFE_UPDATES = 1;
```

- 13.** Actualizar el precio de todos los productos sumándole 5€ al precio actual.

```
SET SQL_SAFE_UPDATES = 0;
UPDATE producto SET precio = precio + 5;
SET SQL_SAFE_UPDATES = 1;
```

- 14.** Elimina todas las impresoras que tienen un precio menor de 200€.

```
SET SQL_SAFE_UPDATES = 0;
DELETE FROM producto WHERE (nombre LIKE "%Impresora%" AND
precio < 200);
SET SQL_SAFE_UPDATES = 1;
```