Exercici 1

Fent servir la base de dades de "botiga", crea una nova taula producto2:

```
CREATE TABLE producto2 (
   codigo INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   nombre VARCHAR(100) NOT NULL,
   precio DOUBLE NOT NULL,
   codigo_fabricante INT UNSIGNED NOT NULL,
   FOREIGN KEY (codigo_fabricante) REFERENCES fabricante(codigo)
);
```

1.Insereix a la taula producto2, les files de la taula producto.

```
INSERT INTO producto2 (codigo, nombre, precio, codigo_fabricante)
SELECT codigo, nombre, precio, codigo_fabricante FROM producto;
```

Per verificar si contenen la mateixa informacio:

```
SELECT * FROM producto;
SELECT * FROM producto2;
```

2. Actualitza el preu de producto2 dels fabricants "Asus" i "Lenovo", incrementa'ls un 6%.

```
UPDATE producto2 SET precio = precio * 1.06 WHERE codigo_fabricante IN (
SELECT codigo FROM fabricante WHERE nombre IN ('Asus', 'Lenovo') );
```

Per verificar si contenen diferent informació:

```
SELECT * FROM producto;
SELECT * FROM producto2;
```

3. Tornar a inserir a la taula producto2 les files de la taula producto. Funciona bé o dona error?

```
INSERT INTO producto2 (codigo, nombre, precio, codigo_fabricante)
SELECT codigo, nombre, precio, codigo_fabricante FROM producto;
```

Sí, dará un error de clave primaria duplicada, porque ya existen los mismos codigo en producto2.

Explicación: La columna codigo es una clave primaria en producto2, y no puede haber valores duplicados.

4. Canvia el nom dels productes, posa el prefixe "P2." al nom

Para que esto funcione he tenido que bajar el nivel de seguridad:

```
SET SQL_SAFE_UPDATES = 0;
```

Despues si que me dejaba actualizarlo

```
UPDATE producto2
SET nombre = CONCAT('P2.', nombre);
```

Comprobamos si funciona

```
SELECT * FROM producto;
SELECT * FROM producto2;
```

5. Esborra de la taula producto2 els productes dels fabricants "Asus" i "Lenovo"

```
DELETE FROM producto2
WHERE codigo_fabricante IN (
  SELECT codigo FROM fabricante WHERE nombre IN ('Asus', 'Lenovo')
);
```

Comprobamos si funciona:

```
SELECT * FROM producto;
SELECT * FROM producto2;
```

Exercici 2

Crea una nova taula fabricante2

```
CREATE TABLE fabricante2 (
  codigo INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL
);
```

1. Insereix a la taula fabricante2, les files de la taula fabricante.

```
INSERT INTO fabricante2 (codigo, nombre)
SELECT codigo, nombre FROM fabricante;
```

Comprobacion:

```
SELECT * FROM fabricante;
SELECT * FROM fabricante2;
```

2. Esborra de la taula fabricante2 les files corresponents als fabricants "Asus" i "Lenovo"

```
DELETE FROM fabricante2
WHERE nombre IN ('Asus', 'Lenovo');
```

Comprobacion

```
SELECT * FROM fabricante;
SELECT * FROM fabricante2;
```

3. Esborra de la taula fabricante les files corresponents als fabricants "Asus" i "Lenovo". Es poden esborrar? Dóna algun error? Explica per què?

```
DELETE FROM fabricante
WHERE nombre IN ('Asus', 'Lenovo');
```

Da un error debido a restricciones de clave foránea (FOREIGN KEY). Si existen productos en la tabla producto o producto2 que referencian a fabricante mediante la columna codigo_fabricante, MySQL no permitirá eliminar esos fabricantes porque rompería la integridad de los datos

4. Esborra de la taula fabricante2 aquells fabricants que no tenen productes ni a la taula producto ni a la taula producto2.

```
DELETE FROM fabricante2
WHERE codigo NOT IN (
   SELECT DISTINCT codigo_fabricante FROM producto
   UNION
   SELECT DISTINCT codigo_fabricante FROM producto2
);
```

Comprobacion

```
SELECT * FROM fabricante;
SELECT * FROM fabricante2;
```

5. Esborra de la taula fabricante aquells fabricants que no tenen productes ni a la taula producto ni a la taula producto2. Es poden esborrar? Dóna algun error? Explica per què?

```
DELETE FROM fabricante
WHERE codigo NOT IN (
  SELECT DISTINCT codigo_fabricante FROM producto
  UNION
  SELECT DISTINCT codigo_fabricante FROM producto2);
```

A mi no me ha dado error pero deberia de dar error si en la tabla producto o producto2 existe una restricción de clave foránea que referencia a fabricante, entonces no deberías poder eliminar un fabricante que tenga productos asociados en estas tablas.

Exercici 3

Es tracta de veure quins motiu poden provocar que al fer insercions o actualitzacions de files en aquestes taules es produeixen errors.

Escriu 5 sentències d'INSERT o UPDATE que donen error a l'hora d'executar-les. Explica per què.

1.

```
INSERT INTO producto2 (codigo, nombre, precio, codigo_fabricante)
VALUES (1, 'Producto1', 100.0, 1);
```

Error Code: 1062. Duplicate entry '1' for key 'producto2.PRIMARY' 0.000 sec

Si intentas insertar un producto con un codigo que ya existe en la tabla producto2, MySQL dará un error de clave primaria duplicada.

2.

```
INSERT INTO producto2 (codigo, nombre, precio, codigo_fabricante)
VALUES (101, 'Producto Nuevo', 200.0, 999);
```

Error Code: 1452. Cannot add or update a child row

Si intentas insertar un producto con un codigo_fabricante que no existe en la tabla fabricante, MySQL dará un error de clave foránea

3.

```
INSERT INTO producto2 (codigo, nombre, precio, codigo_fabricante)
VALUES (102, 'Producto Incompatible', 'Precio', 1);
```

Error Code: 1265. Data truncated for column 'precio' at row 1 0.000 sec

Si intentas insertar un valor no numérico en una columna que requiere un tipo de dato específico, como en el caso de intentar insertar un string en la columna precio, se producirá un error de tipo de datos.

4. .

```
UPDATE producto2
SET codigo_fabricante = 999
WHERE codigo = 1;
```

Error Code: 1452. Cannot add or update a child row

Si intentas actualizar el campo codigo_fabricante a un valor que no existe en la tabla fabricante en este caso con codigo = 999 como no existe se generará un error de violación de clave foránea.

5.

```
UPDATE producto2
SET precio = NULL
WHERE codigo = 1;
```

Error Code: 1048. Column 'precio' cannot be null 0.000 sec

Si intentas poner un valor NULL en una columna que está definida como NOT NULL, se producirá un error.