SQL de la B.D. TOROS

1. Crear en SQL las tablas con sus claves primarias:

```sql
CREATE TABLE toreros (
    dni INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(25),
    apodo VARCHAR(25),
    fecha_alternativa DATE,
    dni_padrino INT
);

CREATE TABLE corridas (
    cod_corrida INT PRIMARY KEY AUTO_INCREMENT,
    num_orden INT,
    nombre_feria VARCHAR(50),
    anyo_celebracion VARCHAR(4),
    precio float
);

CREATE TABLE actua (
    cod_actua INT PRIMARY KEY AUTO_INCREMENT,
    dni INT,
    cod_corrida INT,
    orejas INT,
    rabos INT,
    puerta_grande BIT
);
```

2. Modificar las tablas para introducir ahora sus claves ajenas:

```sql
ALTER TABLE toreros
ADD FOREIGN KEY (dni_padrino) REFERENCES toreros(dni);

ALTER TABLE actua
ADD FOREIGN KEY (dni) REFERENCES toreros(dni),
ADD FOREIGN KEY (cod_corrida) REFERENCES corridas(cod_corrida);
```

4. Añadir nueva columna en la tabla toreros:

```sql
ALTER TABLE toreros
ADD COLUMN ciudad_nacimiento VARCHAR(25);
```

5. Obtener las siguientes consultas:

```sql
-- 5.1. Todos los datos de la tabla toreros.

SELECT * FROM toreros;

-- 5.2. Nombre, apodo y ciudad de los toreros cuyo nombre contenga una ‘n’ y la ciudad contenga en la segunda letra una ‘a’.

SELECT nombre, apodo, ciudad_nacimiento
FROM toreros
WHERE nombre LIKE '%n%' AND ciudad_nacimiento LIKE '_a%';

-- 5.3. Código de la corrida y nombre de la feria cuyo precio esté entre 50 y 150 euros (ambos inclusive), ordenados por el nombre de la feria.

SELECT cod_corrida, nombre_feria
FROM corridas
WHERE precio BETWEEN 50 AND 150
ORDER BY nombre_feria;

-- 5.4. Nombre del torero junto con el dni del padrino cuya fecha de alternativa es hoy.

SELECT nombre, dni_padrino
FROM toreros
WHERE toreros.fecha_alternativa = CURRENT_DATE();

-- 5.5. Nombre del torero (sin duplicados) que aún no haya conseguido salir por la puerta grande. Hacerlo con reunión natural porque hay más de una tabla.

SELECT DISTINCT nombre, puerta_grande
FROM toreros, actua
WHERE actua.puerta_grande = 0
AND toreros.dni = actua.dni;

-- 5.6. Nombre de la feria, año de celebración y orejas que consiguió el torero “Pepe Chen”. Hacerlo con reunión natural porque hay más de una tabla.

SELECT nombre_feria, anyo_celebracion, orejas
FROM corridas, actua, toreros
WHERE toreros.nombre = 'Pepe Chen'
AND actua.dni = toreros.dni
AND corridas.cod_corrida = actua.cod_corrida;

-- 5.8. Fecha de alternativa más antigua.

SELECT MIN(fecha_alternativa)
FROM toreros
WHERE fecha_alternativa IS NOT NULL;

-- 5.9. Media de los precios de entradas.
```

```sql
SELECT AVG(precio)
FROM corridas;

-- 5.10. Los dni de los toreros, junto con el total del número de orejas
conseguidas.

SELECT dni, orejas
FROM actua
GROUP BY dni;

-- 5.11. Igual a la anterior, pero cuyo total de número de orejas conseguidas
superen las 20.

SELECT dni, SUM(orejas)
FROM actua, toreros
WHERE actua.dni = toreros.dni
GROUP BY actua.dni
HAVING SUM(orejas) > 20;

-- 5.12. Eliminar todas las actuaciones cuyas orejas sean mayores que 2 o
menores que cero.

DELETE FROM actua
WHERE orejas > 2 OR orejas < 0;

-- 5.13. Suponiendo que hemos exigido integridad referencial entre las
relaciones toreros y actúa, sin establecer borrado en cascada. Eliminar al
toreros cuyo dni es el 3 (se realizará en dos consultas 5.13.1 y 5.13.2)

DELETE FROM actua WHERE dni = 3;
DELETE FROM toreros WHERE dni = 3;

-- 5.14. Vamos a suponer que, de momento, se suspende la tradición taurina, por
lo que hay que eliminar toda la información. Realizar las consultas necesarias
para que se eliminen tanto las tablas como las relaciones.

DROP DATABASE toros_db;
```