

Consultas BD Gimnasio

```
CREATE DATABASE gimnasio;

USE gimnasio;

CREATE TABLE IF NOT EXISTS socios (
    cod_socio INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(25),
    ciudad VARCHAR(50),
    fecha_nacimiento DATE
);

CREATE TABLE IF NOT EXISTS clases (
    cod_clase INT AUTO_INCREMENT PRIMARY KEY,
    decripcion VARCHAR(120),
    dia DATE
);

CREATE TABLE IF NOT EXISTS alquiler (
    cod_alquiler INT AUTO_INCREMENT PRIMARY KEY,
    cod_socio INT,
    cod_clase INT,
    fecha_alquiler DATE,
    precio FLOAT,

    CONSTRAINT fk_alqsoc FOREIGN KEY (cod_socio) REFERENCES
socios(cod_socio),
    CONSTRAINT fk_alqcla FOREIGN KEY (cod_clase) REFERENCES
clases(cod_clase)
);

INSERT INTO socios (nombre, ciudad, fecha_nacimiento) VALUES
('Juan', 'Madrid', '1980-01-01'),
('María', 'Barcelona', '1985-02-15'),
('Pedro', 'Sevilla', '1990-03-30'),
('Lucía', 'Valencia', '1995-04-20'),
('Carlos', 'Bilbao', '2000-05-10');

INSERT INTO clases (decripcion, dia) VALUES
('Yoga', '2023-11-06'),
('Pilates', '2023-11-07'),
('Spinning', '2023-11-08'),
('Zumba', '2023-11-09'),
('Crossfit', '2023-11-10');

INSERT INTO alquiler (cod_socio, cod_clase, fecha_alquiler, precio) VALUES
(1, 1, '2023-11-06', 10.0),
(2, 2, '2023-11-07', 12.0),
```

```
(3, 3, '2023-11-08', 15.0),  
(4, 4, '2023-11-09', 20.0),  
(5, 5, '2023-11-10', 25.0);
```

5.1 Todos los datos de la tabla alquiler.

```
SELECT *  
FROM alquiler;
```

5.2 Nombre y ciudad de los socios cuyo nombre contenga una 'm' y la ciudad contenga en la segunda letra una 'a'.

```
SELECT nombre, ciudad  
FROM socios  
WHERE nombre LIKE '%m%'  
AND ciudad LIKE '_a%';
```

5.3 Código del alquiler y fecha del alquiler cuyo precio esté entre 10 y 30 euros, ordenados por el precio de forma ascendente.

```
SELECT cod_alquiler, fecha_alquiler  
FROM alquiler  
WHERE precio BETWEEN 10 AND 30  
ORDER BY precio;
```

5.4 Código de las clases que aún no tengan asignada la descripción.

```
SELECT cod_clase  
FROM clases  
WHERE descripcion IS NULL;
```

5.5 Nombre sin duplicados y teléfono de los socios que han alquilado una clase hoy.

```
SELECT DISTINCT nombre, telefono  
FROM socios, alquiler, clase  
WHERE dia = CURRENT_DATE()  
AND socios.cod_socio = alquiler.cod_socio  
AND clases.cod_clase = alquiler.cod_clase;
```

5.6 Nombre y fecha de nacimiento de los socios que han alquilado alguna clase de "spinning"

```
SELECT nombre, fecha_nacimiento
FROM socios, alquiler, clases
WHERE descripcion = 'spinning'
AND socios.cod_socio = alquiler.cod_socio
AND clases.cod_clase = alquiler.cod_clase;
```

5.7 Nombre del socio más mayor. Se hace en dos partes, con subconsulta.

```
SELECT nombre
FROM socios
WHERE edad = (
    SELECT MAX(edad)
    FROM socios
);
```

5.8 Total de las clases.

```
SELECT count(*)
FROM clases;
```

5.9 Media de los precios de alquiler.

```
SELECT AVG(precios)
FROM alquiler;
```

5.10 El código del socio junto con el total de los precios de las clases alquiladas.

```
SELECT cod_socio, SUM(precio)
FROM socios, alquiler
WHERE socios.cod_socio = alquiler.cod_socio;
```

5.11 Igual que la anterior, pero cuyo total de los precios supere los 100 euros.

```
SELECT cod_socio, SUM(precio)
FROM socios, alquiler
WHERE precio > 100
AND socios.cod_socio = alquiler.cod_socio;
```

5.12 Eliminar todos los socios anteriores al 1920.

```
DELETE
FROM socios
WHERE fecha_nacimiento < '1920-1-1';
```

5.13 Suponiendo que hemos exigido integridad referencial entre la relaciones socios y alquiler, sin establecer borrado en cascada. Elimina al socio con el número de código que tú quieras.

```
DELETE
FROM alquiler
WHERE cod_socio = 1;

DELETE
FROM socio
WHERE cod_socio = 1;
```

5.14 El Gimnasio ha sido traspasado a otro, ya no se necesita ninguna información sobre los socios, pues ellos tienen los suyos y con otros datos, por lo tanto pasamos a eliminar toda la información de los socios.

```
ALTER TABLE alquiler
DROP CONSTRAINT fk_alqsoc;

DROP TABLE socios;
```