

Respuestas –Gestión de un Centro de Entrenamiento Deportivo

Paso 1: Crear la base de datos

```
CREATE DATABASE centro_entrenamiento;  
USE centro_entrenamiento;
```

Paso 2: Crear las tablas

Tabla: **entrenadores**

```
CREATE TABLE entrenadores (  
    id_entrenador INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(50),  
    apellido VARCHAR(50),  
    especialidad VARCHAR(50),  
    activo BOOLEAN  
);
```

Tabla: **clientes**

```
CREATE TABLE clientes (  
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(50),  
    apellido VARCHAR(50),  
    edad INT,  
    objetivo VARCHAR(100)  
);
```

Tabla: **rutinas**

```
CREATE TABLE rutinas (  
    id_rutina INT AUTO_INCREMENT PRIMARY KEY,  
    descripcion TEXT,  
    nivel VARCHAR(20)  
);
```

Tabla: asignaciones

```
CREATE TABLE asignaciones (  
    id_asignacion INT AUTO_INCREMENT PRIMARY KEY,  
    id_cliente INT,  
    id_rutina INT,  
    id_entrenador INT,  
    fecha_inicio DATE,  
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),  
    FOREIGN KEY (id_rutina) REFERENCES rutinas(id_rutina),  
    FOREIGN KEY (id_entrenador) REFERENCES entrenadores(id_entrenador)  
);
```

Respuestas ejercicios:**1. Mostrar todos los clientes del sistema**

```
SELECT * FROM clientes;
```

2. Listar los entrenadores que están activos

```
SELECT * FROM entrenadores WHERE activo = TRUE;
```

3. Mostrar las rutinas con nivel "Principiante"

```
SELECT * FROM rutinas WHERE nivel = 'Principiante';
```

4. Consultar los clientes mayores de 25 años

```
SELECT * FROM clientes WHERE edad > 25;
```

5. Mostrar los nombres y apellidos de todos los entrenadores ordenados alfabéticamente por apellido

```
SELECT nombre, apellido FROM entrenadores ORDER BY apellido ASC;
```

6. Mostrar cuántos clientes hay por cada objetivo (usar **GROUP BY)**

```
SELECT objetivo, COUNT(*) AS cantidad FROM clientes GROUP BY objetivo;
```

7. Obtener la cantidad de entrenadores por especialidad

```
SELECT especialidad, COUNT(*) AS cantidad FROM entrenadores GROUP BY especialidad;
```

8. Mostrar la cantidad de rutinas que hay por nivel, pero solo si hay más de una (usar **HAVING)**

```
SELECT nivel, COUNT(*) AS cantidad FROM rutinas GROUP BY nivel HAVING COUNT(*) > 1;
```

9. Consultar los datos de los clientes junto a la rutina que tienen asignada (usar **JOIN)**

```
SELECT c.nombre, c.apellido, r.descripcion  
FROM clientes c  
JOIN asignaciones a ON c.id_cliente = a.id_cliente  
JOIN rutinas r ON a.id_rutina = r.id_rutina;
```

10. Mostrar el nombre del entrenador que tiene asignado a cada cliente (JOIN múltiple)

```
SELECT c.nombre AS cliente, e.nombre AS entrenador  
FROM asignaciones a  
JOIN clientes c ON a.id_cliente = c.id_cliente  
JOIN entrenadores e ON a.id_entrenador = e.id_entrenador;
```

11. Consultar los nombres de los clientes y sus rutinas donde el entrenador se llama "Lucía"

```
SELECT c.nombre AS cliente, r.descripcion AS rutina  
FROM asignaciones a  
JOIN clientes c ON a.id_cliente = c.id_cliente
```

```
JOIN rutinas r ON a.id_rutina = r.id_rutina
JOIN entrenadores e ON a.id_entrenador = e.id_entrenador
WHERE e.nombre = 'Lucía';
```

12. Listar todos los entrenadores que no tienen asignaciones

```
SELECT * FROM entrenadores
WHERE id_entrenador NOT IN (
    SELECT DISTINCT id_entrenador FROM asignaciones
);
```

13. Mostrar todos los clientes que tienen una rutina del nivel "Avanzado"

```
SELECT c.*
FROM clientes c
JOIN asignaciones a ON c.id_cliente = a.id_cliente
WHERE a.id_rutina IN (
    SELECT id_rutina FROM rutinas WHERE nivel = 'Avanzado'
);
```

14. Crear una vista llamada **vista_clientes_rutinas que muestre: nombre del cliente, rutina y fecha de inicio**

```
CREATE VIEW vista_clientes_rutinas AS
SELECT c.nombre, r.descripcion AS rutina, a.fecha_inicio
FROM clientes c
JOIN asignaciones a ON c.id_cliente = a.id_cliente
JOIN rutinas r ON a.id_rutina = r.id_rutina;
```

15. Actualizar el objetivo del cliente "Camila Vega" a "Rehabilitación avanzada"

```
UPDATE clientes
```

```
SET objetivo = 'Rehabilitación avanzada'  
WHERE nombre = 'Camila' AND apellido = 'Vega';
```

16. Eliminar la rutina "Yoga para la movilidad" del sistema

```
DELETE FROM rutinas  
WHERE descripcion = 'Yoga para la movilidad';
```

17. Insertar un nuevo entrenador llamado "Pedro Duarte", especializado en HIIT, activo

```
INSERT INTO entrenadores (nombre, apellido, especialidad, activo)  
VALUES ('Pedro', 'Duarte', 'HIIT', TRUE);
```

18. Mostrar todos los clientes que aún no tienen rutina asignada

```
SELECT * FROM clientes  
WHERE id_cliente NOT IN (  
    SELECT DISTINCT id_cliente FROM asignaciones  
);
```

19. Mostrar la edad promedio de los clientes

```
SELECT AVG(edad) AS edad_promedio FROM clientes;
```

20. Crear una subconsulta en FROM que devuelva la cantidad de clientes por objetivo y mostrar aquellos que tengan más de 1 cliente

```
SELECT objetivo, cantidad  
FROM (  
    SELECT objetivo, COUNT(*) AS cantidad  
    FROM clientes  
    GROUP BY objetivo  
) AS sub  
WHERE cantidad > 1;
```

