

**1. Explicación del proyecto**

El FC Barcelona Femenino pretende desarrollar una aplicación que permita almacenar y gestionar en una base de datos información acerca de las futbolistas que han militado en el club a lo largo de su historia. La idea gusta mucho en la Liga de Fútbol Profesional, y deciden hacerla extensiva al resto de clubes de primera y segunda división.

De esta forma tendremos que para cada futbolista se almacenaría la siguiente información: su nombre y apellido, su año de nacimiento, su nacionalidad y su NIF. De cada club de fútbol se guardaría su nombre oficial, el año de creación y el nombre de su estadio. Además, se almacenaría información de qué futbolistas militaron en que clubes en las diferentes campañas ligueras.

Para simplificar, supondremos que una futbolista dada sólo puede militar en un club a lo largo de una temporada.

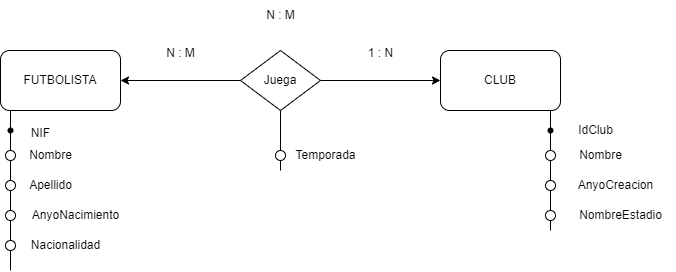
**2. Creación del modelo entidad-relación**

Dada la descripción del problema que se nos plantea, podemos extraer gracias a la información que se nos da dos tablas diferentes, una para las futbolistas y otra para los clubs donde juegan.

Ambas tablas son relacionadas entre sí mediante una relación N:M, ya que, una futbolista puede haber jugado en 1 o N clubes, y en un club pueden haber jugado N o M futbolistas, ya que para que el club pueda competir, necesita un mínimo de 11.

Al mantener una relación N:M, creamos una nueva tabla para la relación formada entre ambas entidades, cuyos atributos serán el NIF de la futbolista, el nombre del club en el que jugó y la temporada en la que militó en tal club.

El diagrama entidad-relación quedaría tal que así:



**3. Creación de las tablas**

Una vez tenemos claro cuales son las tablas que ocupan nuestra base de datos, vamos a crearlas. Empezaremos con la tabla “Futbolista”:

CREATE TABLE Futbolista (

    NIF VARCHAR(9) PRIMARY KEY,

    Nombre VARCHAR(25),

    Apellido VARCHAR(20),

    AnyoNacimiento YEAR NOT NULL,

    Nacionalidad VARCHAR(20) NOT NULL

);

* El NIF es del tipo VARCHAR, con longitud de 9 caracteres, ya que el formato exige que contenga una letra, por lo que no puede ser INT, y 9 es la longitud obligatoria para este campo.
* El nombre es del tipo VARCHAR, con una longitud de 25 caracteres para evitar problemas con posibles nombres compuestos y/o extranjeros que superen la longitud esperada.
* El apellido es del tipo VARCHAR, con una longitud de 20 caracteres para igualmente evitar problemas con apellidos extranjeros.
* El atributo AnyoNacimiento (sin “ñ” para evitar caracteres especiales) es del tipo YEAR, el cual recibe un numero entero de una longitud tanto mínima como máxima de 4 dígitos. Este campo no puede ser nunca nulo.
* La nacionalidad es del tipo VARCHAR, con una longitud de 20 caracteres, más que suficiente para recoger el nombre de cualquier nacionalidad.

Vamos a seguir con la creación de la tabla “Club”:

CREATE TABLE Club (

    IdClub INT PRIMARY KEY,

    Nombre VARCHAR(30) NOT NULL,

    AnyoCreacion YEAR NOT NULL,

    NombreEstadio VARCHAR(30) NOT NULL

);

* El IdClub es del tipo INT, y al ser clave primaria, no puede repetirse nunca.
* El nombre es del tipo VARCHAR(30) para evitar problemas de longitud con equipos con nombres formados por varias palabras (por ejemplo: Racing de Santander).
* El atributo AnyoCreacion (nuevamente sin “ñ” para evitar caracteres especiales) es del tipo YEAR, y nunca puede ser nulo.
* El atributo NombreEstadio es del tipo VARCHAR(30) para evitar nuevamente problemas de longitud con nombres compuestos por varias palabras (por ejemplo: Ramón Sánchez Pizjuán).

Una vez tenemos creadas las tablas para nuestras entidades, vamos a pasar a crear la tabla formada por la relación entre ellas, “juega”:

CREATE TABLE Juega (

    NIF\_futbolista VARCHAR(9),

    IdClub INT,

    Temporada VARCHAR(9),

    PRIMARY KEY (NIF\_futbolista, IdClub),

    FOREIGN KEY (NIF\_futbolista) REFERENCES Futbolista(NIF),

    FOREIGN KEY (IdClub) REFERENCES Club(IdClub)

);

* Los atributos NIF\_Futbolista (que hereda de Futbolista.NIF) y IdClub (que hereda de Club.IdClub) mantienen los tipos que tienen en las otras tablas. Actúan como claves foráneas hacia estas tablas, y juntas forman la clave primaria de esta tabla.
* El atributo Temporada es del tipo VARCHAR(9) ya que el formato en el que vamos a introducir este valor será año1/año2 (por ejemplo 2022/2023), siendo siempre una combinación de 9 caracteres.

**4. Modificaciones a las tablas**

**1.** Añade un atributo en el que se recoja si la jugadora llegó de los escalafones inferiores del FCB. Este atributo solo puede tomar los valores “S” ó “N”.

La primera modificación que se nos pide es añadir un nuevo atributo a la tabla “Futbolista”, que actúe como un booleano que indique si la futbolista ha pasado por las categorías inferiores del FC Barcelona.

ALTER TABLE Futbolista

ADD FilialFCB CHAR(1) NOT NULL CHECK (FilialFCB IN ('S', 'N'));

Con esta modificación añadimos un nuevo atributo llamado “FilialFCB” de tipo CHAR(1), y que únicamente puede contener los valores “S” y “N”, controlándolo gracias a CHECK.

**2.** Añade otro atributo que contenga el e-mail de la jugadora. El valor de este atributo debe ser único para cada jugadora y se debe comprobar que contiene la arroba (@) como parte de su validez.

La segunda restricción que nos piden es añadir un nuevo campo a la tabla Futbolista donde se registre el email de la jugadora. Este atributo debe ser único y se debe verificar que contenga la @.

ALTER TABLE Futbolista

ADD Email VARCHAR(50) UNIQUE,

ADD CONSTRAINT ComprobarEmail CHECK (Email LIKE '%@%');

Con esta sentencia añadimos el campo Email, de tipo VARCHAR(50) para evitar problemas de longitud, y con la cláusula UNIQUE para que nunca se repita. Seguidamente, añadimos un CONSTRAINT (comprobación) llamada ComprobarEmail que se encarga de que el email contenga una @.

**3.** Escribe la sentencia que eliminaría el atributo nacionalidad de la jugadora.

La tercera modificación que se nos pide es eliminar la columna Nacionalidad de la tabla Futbolista.

ALTER TABLE Futbolista

DROP COLUMN Nacionalidad;

Con la siguiente sentencia, eliminamos la columna Nacionalidad de la tabla Futbolista.

**4.** Escribe la sentencia que añadiría un atributo que contenga la capacidad de espectadores que tiene el estadio. Debe ser un número positivo no nulo y menor de 99.999, y por supuesto mayor de cero.

En esta modificación se nos pide añadir un nuevo campo que solicite la capacidad de espectadores del estadio. Debemos controlar que el numero sea válido y que este campo nunca sea nulo.

ALTER TABLE Club

ADD CapacidadEspectadores INT NOT NULL CHECK (CapacidadEspectadores > 0 AND CapacidadEspectadores < 99999);

Gracias al CHECK, podemos añadir operaciones aritméticas para controlar que el número que introduzcamos sea mayor que 0 y menor que 99999, y añadiendo la cláusula NOT NULL hacemos que el atributo nunca sea nulo.

**5.** Escribe la sentencia que cree un índice con el campo nombre y apellidos de las jugadoras.

En esta modificación se nos pide crear un índice que contenga el nombre y apellido de la futbolista, para así acelerar las búsquedas sobre la tabla.

CREATE INDEX IndiceNombreApellido ON Futbolista (Nombre, Apellido);

**6.** Escribe la sentencia que cree un índice con el campo año de creación del club.

Con esta modificación creamos un índice para acelerar las búsquedas sobre la tabla Club, a través del campo “AnyoCreacion”.

CREATE INDEX IndiceAnyoCreacion ON Club (AnyoCreacion);

**7.** Si por ejemplo has llamado a la tabla contiene a las jugadoras como “FUTBOLISTAS”, escribe la sentencia que renombraría esa tabla a “JUGADORAS”.

ALTER TABLE Futbolista

RENAME TO Jugadora;

Con esta sentencia, renombramos la tabla “Futbolista” a “Jugadora”.

**8.** Escribe la sentencia que eliminaría esta última tabla.

DROP TABLE Jugadora;

Con esta sentencia eliminamos la tabla que hemos renombrado como “Jugadora”.

**5. Funcionalidades**