

EJERCICIOS RESUELTOS TEMA 11

11.1. Crea una base de datos "ejercicio11", en la que guardaremos información sobre deportistas y disciplinas deportivas. Tendremos una tabla "DEPORTES" y una tabla "DEPORTISTAS", unidas por una relación M:M (cada deporte puede ser practicado por más de un deportista y cada deportista podría practicar más de un deporte). Además, existirá una tabla "PAISES". Para cada país guardaremos el nombre (por ejemplo, "España") y un código que actuará como clave primaria (por ejemplo, "ESP"). De cada deportista anotaremos código, nombre, apellidos. Para cada deporte, nos interesará código y nombre. La relación M:M se reflejará en una nueva tabla "PRACTICAR", cada uno de cuyos registros estará formado por el código de deporte y el código de deportista (ambos formarán la clave primaria -compuesta- de esta nueva tabla). La relación 1:M entre deportista y país se reflejará en que de cada deportista deberá se deberá anotar también el código del país al que representa.

¡OJO! HAY QUE AÑADIR UN CAMPO 'CÓDIGO DE DEPORTE' EN LA TABLA 'DEPORTISTAS'

```
using System;
using System.Data.SQLite;

namespace bbdd
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
            ..\\..\\..\\Ejercicio11.sqlite; " +
                "Version = 3; New = True; Compress = True");
            conexion.Open();

            string creacion = @"CREATE TABLE deportes (codigo varchar (5), nombre
            varchar (15));
                                CREATE TABLE deportistas (codigo varchar (5),
            nombre varchar (15),
                                apellidos varchar (20), pais varchar (15));
                                CREATE TABLE paises (nombre varchar(15), codigo
            varchar(5) PRIMARY KEY);
                                CREATE TABLE practicar (codDeporte varchar (5),
            codDeportista varchar (5))";
            SQLiteCommand cmd = new SQLiteCommand(creacion, conexion);
            cmd.ExecuteNonQuery();

            conexion.Close();
            Console.WriteLine("Tablas creadas con éxito.");
            Console.ReadLine();
        }
    }
}
```

11.2a. Añade los países:

- JAM, Jamaica
- ESP, España
- USA, Estados Unidos de América

- AUS, Australia
- RUS, Rusia

11.2b. Añade los deportistas:

- BOL, Usain, Bolt (Jamaica)
- POW, Asafa, Powell (Jamaica)
- CRA, Saúl, Craviotto (España)
- TAU, Diana, Taurasi (Estados Unidos)
- PHE, Michael, Phelps (Estados Unidos)
- MUR, Andy, Murray (UK)

11.2c. Añade los deportes:

- ATL, Atletismo
- REM, Remo
- BAL, Baloncesto
- NAT, Natación
- BAD, Badminton

11.2d. Y la relación Practicar:

- Usain Bolt, Velocidad
- Saúl Craviotto, Remo
- Diana Taurasi, Baloncesto
- Michael Phelps, Natación

```
using System;
using System.Data.SQLite;

namespace _11._2
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\\..\\..\\Ejercicio11.sqlite; " +
                "Version = 3; New = False; Compress = True");
            conexion.Open();

            string insercion = @"INSERT INTO paises VALUES ('Jamaica', 'JAM'),
('España', 'ESP'), ('Estados Unidos de America', 'USA'),
                ('Australia', 'AUS'), ('Rusia', 'RUS');
            INSERT INTO deportistas VALUES ('BOL', 'Usain',
'Bolt', 'Jamaica'),
                ('POW', 'Asafa', 'Powell', 'Jamaica'), ('CRA',
'Saúl', 'Craviotto', 'España'),
                ('TAU', 'Diana', 'Taurasi', 'Estados Unidos'),
('PHE', 'Michael', 'Phelps', 'Estados Unidos'),
                ('MUR', 'Andy', 'Murray', 'UK');
            INSERT INTO deportes VALUES ('ATL', 'Atletismo'),
('REM', 'Remo'), ('BAL', 'Baloncesto'),
                ('NAT', 'Natación'), ('BAD', 'Badminton');
            INSERT INTO practicar VALUES ('VEL', 'BOL'),
('REM', 'CRA'), ('BAL', 'TAU'), ('NAT', 'PHE')";
```

```

        SQLiteCommand cmd = new SQLiteCommand(insertion, conexion);
        cmd.ExecuteNonQuery();

        conexion.Close();
        Console.WriteLine("Datos insertados con éxito.");
        Console.ReadLine();
    }
}

```

11.3. Muestra los nombres y apellidos de todos los deportistas, junto al nombre del país al que pertenecen.

```

using System;
using System.Data.SQLite;

namespace _11._3
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
            ..\\..\\..\\Ejercicio11.sqlite; " +
                "Version = 3; New = False; Compress = True");
            conexion.Open();

            string mostrar = @"SELECT deportistas.nombre, apellidos,
            pais.nombre FROM deportistas
            INNER JOIN pais ON pais.codigo =
            deportistas.pais";
            SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
            SQLiteDataReader datos = cmd.ExecuteReader();

            while (datos.Read())
            {
                string nombre = Convert.ToString(datos[0]);
                string apellidos = Convert.ToString(datos[1]);
                string pais = Convert.ToString(datos[2]);
                Console.WriteLine("Nombre: {0}, Apellidos: {1}, País: {2}",
                nombre, apellidos, pais);
            }

            conexion.Close();
            Console.ReadLine();
        }
    }
}

```

11.4. Muestra los nombres y apellidos de todos los deportistas, junto al nombre del país al que pertenecen, en caso de que dicho país aparezca en la base de datos. También se deben mostrar los datos de los deportistas cuyo país no se haya introducido aún.

```

using System;
using System.Data.SQLite;

namespace _11._4
{

```

```

class Program
{
    static void Main()
    {
        SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\..\..\..\Ejercicio11.sqlite; " +
            "Version = 3; New = False; Compress = True");
        conexion.Open();

        string mostrar = @"SELECT deportistas.nombre, apellidos,
países.nombre FROM deportistas
LEFT OUTER JOIN países ON países.codigo =
deportistas.país";
        SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
        SQLiteDataReader datos = cmd.ExecuteReader();

        while (datos.Read())
        {
            string nombre = Convert.ToString(datos[0]);
            string apellidos = Convert.ToString(datos[1]);
            string pais = Convert.ToString(datos[2]);
            Console.WriteLine("Nombre: {0}, Apellidos: {1}, País: {2}",
nombre, apellidos, pais);
        }

        conexion.Close();
        Console.ReadLine();
    }
}

```

11.5. Muestra los nombres de todos los países, junto a los apellidos de los deportistas de ese país. Deben aparecer también los países de los que no conozcamos ningún deportista.

```

using System;
using System.Data.SQLite;

namespace _11._5
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\..\..\..\Ejercicio11.sqlite; " +
                "Version = 3; New = False; Compress = True");
            conexion.Open();

            string mostrar = "SELECT países.nombre, apellidos FROM países LEFT
OUTER JOIN deportistas ON países.codigo = deportistas.país";
            SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
            SQLiteDataReader datos = cmd.ExecuteReader();

            while (datos.Read())
            {
                string pais = Convert.ToString(datos[0]);
                string apellidos = Convert.ToString(datos[1]);
                Console.WriteLine("País: {0}, Apellidos: {1}", pais, apellidos);
            }

            conexion.Close();
        }
    }
}

```

```

        Console.ReadLine();
    }
}

```

11.6. Muestra los nombres y apellidos de todos los deportistas, junto al nombre del deporte que practican (sólo aquellos de los que tengamos constancia que realmente practican algún deporte).

```

using System;
using System.Data.SQLite;

namespace _11._6
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source = 
            ..\\..\\..\\Ejercicio11.sqlite; " +
                "Version = 3; New = False; Compress = True");
            conexion.Open();

            string mostrar = @"SELECT deportistas.nombre, deportistas.apellidos,
            deportes.nombre FROM deportistas
                                INNER JOIN practicar ON deportistas.codigo =
            practicar.codDeportista
                                INNER JOIN deportes ON practicar.codDeporte =
            deportes.codigo";
            SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
            SQLiteDataReader datos = cmd.ExecuteReader();

            while (datos.Read())
            {
                string nombre = Convert.ToString(datos[0]);
                string apellidos = Convert.ToString(datos[1]);
                string pais = Convert.ToString(datos[2]);
                Console.WriteLine("Nombre: {0}, Apellidos: {1}, Deporte: {2}",
                nombre, apellidos, pais);
            }

            conexion.Close();
            Console.ReadLine();
        }
    }
}

```

11.7. Muestra los nombres y apellidos de todos los deportistas, junto al nombre del deporte que practican (incluso los que no sepamos cuál es su deporte).

```

using System;
using System.Data.SQLite;

namespace _11._7
{
    class Program
    {
        static void Main()
        {

```

```

        SQLiteConnection conexion = new SQLiteConnection("Data Source =
        ..\\..\\..\\Ejercicio11.sqlite; " +
            "Version = 3; New = False; Compress = True");
        conexion.Open();

        string mostrar = @"SELECT deportistas.nombre, deportistas.apellidos,
        deportes.nombre FROM deportistas
                                LEFT OUTER JOIN practicar ON deportistas.codigo =
        practicar.codDeportista
                                LEFT OUTER JOIN deportes ON practicar.codDeporte =
        deportes.codigo";
        SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
        SQLiteDataReader datos = cmd.ExecuteReader();

        while (datos.Read())
        {
            string nombre = Convert.ToString(datos[0]);
            string apellidos = Convert.ToString(datos[1]);
            string deporte = Convert.ToString(datos[2]);
            Console.WriteLine("Nombre: {0}, Apellidos: {1}, Deporte: {2}",
        nombre, apellidos, deporte);
        }

        conexion.Close();
        Console.ReadLine();
    }
}
}

```

11.8. Muestra el nombre de todos los deportes, junto con los nombres y apellidos de los deportistas que lo practican (incluso si para algún deporte aún no hemos introducido ningún deportista).

RIGHT and FULL OUTER JOINS are not currently supported'

```

using System;
using System.Data.SQLite;

namespace _11._8
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
            ..\\..\\..\\Ejercicio11.sqlite; " +
                "Version = 3; New = False; Compress = True");
            conexion.Open();

            string mostrar = @"SELECT deportes.nombre, deportistas.nombre,
            deportistas.apellidos FROM deportes
                                LEFT OUTER JOIN practicar ON deportes.codigo =
            practicar.codDeporte
                                LEFT OUTER JOIN deportistas ON
            practicar.codDeportista = deportistas.codigo";
            SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
            SQLiteDataReader datos = cmd.ExecuteReader();

            while (datos.Read())
            {
                string deporte = Convert.ToString(datos[0]);

```

```

        string deportista = Convert.ToString(datos[1]);
        Console.WriteLine("Deporte: {0}, Deportista: {1}", deporte,
deportista);
    }
    conexion.Close();
    Console.ReadLine();
}
}
}

```

11.9. Unificar todos los ejercicios en la misma clase.

```

using System;
using System.Data.SQLite;

namespace _11._9
{
    class Program
    {
        static SQLiteConnection conexion;

        static void Main()
        {
            EstablecerConexion();
            CrearBBDD();
            Console.WriteLine("Tablas creadas con éxito. Pulsa Intro para
continuar...");
            Console.ReadLine();
            InsertarDatos();
            Console.WriteLine("Datos insertados con éxito. Pulsa Intro para
continuar...");
            Console.ReadLine();
            Console.WriteLine("Mostrar todos los nombres de los deportistas junto
a sus países de origen: ");
            PrimeraConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
            Console.WriteLine("Mostrar todos los nombres de los deportistas junto
a sus países de " +
                "origen (aunque no estén en la base de datos): ");
            SegundaConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
            Console.WriteLine("Mostrar todos los países de los deportistas junto
a sus apellidos: ");
            TerceraConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
            Console.WriteLine("Mostrar todos los deportistas junto a los deportes
que practican: ");
            CuartaConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
            Console.WriteLine("Mostrar todos los deportistas junto a los deportes
que " +
                "practican (incluso aunque no sepamos cuál es su deporte): ");
            QuintaConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
        }
    }
}

```

```

        Console.WriteLine("Mostrar todos los deportes junto a los deportistas
que los practican, incluso si no hay deportistas: ");
        SextaConsulta();
        Console.WriteLine("Pulsa Intro para acabar...");
        Console.ReadLine();
        CerrarBBDD();
    }

    static void EstablecerConexion()
    {
        conexion = new SQLiteConnection("Data Source =
..\..\..\..\Ejercicio11.sqlite; " +
        "Version = 3; New = True; Compress = True");
        conexion.Open();
    }

    static void CrearBBDD()
    {
        string creacion = @"CREATE TABLE deportes (codigo varchar (5), nombre
varchar (15));
                                CREATE TABLE deportistas (codigo varchar (5),
nombre varchar (15),
                                apellidos varchar (20), pais varchar (15));
                                CREATE TABLE paises (nombre varchar(15), codigo
varchar(5) PRIMARY KEY);
                                CREATE TABLE practicar (codDeporte varchar (5),
codDeportista varchar (5))";
        SQLiteCommand cmd = new SQLiteCommand(creacion, conexion);
        cmd.ExecuteNonQuery();
    }

    static void InsertarDatos()
    {
        string insercion = @"INSERT INTO paises VALUES ('Jamaica', 'JAM'),
('España', 'ESP'), ('Estados Unidos de America', 'USA'),
                                ('Australia', 'AUS'), ('Rusia', 'RUS');
                                INSERT INTO deportistas VALUES ('BOL', 'Usain',
'Bolt', 'JAM'),
                                ('POW', 'Asafa', 'Powell', 'JAM'), ('CRA',
'Saúl', 'Craviotto', 'ESP'),
                                ('TAU', 'Diana', 'Taurasi', 'USA'), ('PHE',
'Michael', 'Phelps', 'USA'),
                                ('MUR', 'Andy', 'Murray', 'UK');
                                INSERT INTO deportes VALUES ('ATL', 'Atletismo'),
('REM', 'Remo'), ('BAL', 'Baloncesto'),
                                ('NAT', 'Natación'), ('BAD', 'Badminton');
                                INSERT INTO practicar VALUES ('VEL', 'BOL'),
('REM', 'CRA'), ('BAL', 'TAU'), ('NAT', 'PHE')";
        SQLiteCommand cmd = new SQLiteCommand(insercion, conexion);
        cmd.ExecuteNonQuery();
    }

    static void PrimeraConsulta()
    {
        string mostrar = @"SELECT deportistas.nombre, apellidos,
paises.nombre FROM deportistas
                                INNER JOIN paises ON paises.codigo =
deportistas.pais";
        SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
        SQLiteDataReader datos = cmd.ExecuteReader();

        while (datos.Read())

```



```

        {
            string nombre = Convert.ToString(datos[0]);
            string apellidos = Convert.ToString(datos[1]);
            string pais = Convert.ToString(datos[2]);
            Console.WriteLine("Nombre: {0}, Apellidos: {1}, País: {2}",
nombre, apellidos, pais);
        }
    }

    static void SegundaConsulta()
    {
        string mostrar = @"SELECT deportistas.nombre, apellidos,
países.nombre FROM deportistas
LEFT OUTER JOIN países ON países.codigo =
deportistas.país";
        SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
        SQLiteDataReader datos = cmd.ExecuteReader();

        while (datos.Read())
        {
            string nombre = Convert.ToString(datos[0]);
            string apellidos = Convert.ToString(datos[1]);
            string pais = Convert.ToString(datos[2]);
            Console.WriteLine("Nombre: {0}, Apellidos: {1}, País: {2}",
nombre, apellidos, pais);
        }
    }

    static void TerceraConsulta()
    {
        string mostrar = @"SELECT países.nombre, apellidos FROM países
LEFT OUTER JOIN deportistas ON países.codigo =
deportistas.país";
        SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
        SQLiteDataReader datos = cmd.ExecuteReader();

        while (datos.Read())
        {
            string pais = Convert.ToString(datos[0]);
            string apellidos = Convert.ToString(datos[1]);
            Console.WriteLine("País: {0}, Apellidos: {1}", pais, apellidos);
        }
    }

    static void CuartaConsulta()
    {
        string mostrar = @"SELECT deportistas.nombre, deportistas.apellidos,
deportes.nombre FROM deportistas
INNER JOIN practicar ON deportistas.codigo =
practicar.codDeportista
INNER JOIN deportes ON practicar.codDeporte =
deportes.codigo";
        SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
        SQLiteDataReader datos = cmd.ExecuteReader();

        while (datos.Read())
        {
            string nombre = Convert.ToString(datos[0]);
            string apellidos = Convert.ToString(datos[1]);
            string pais = Convert.ToString(datos[2]);
            Console.WriteLine("Nombre: {0}, Apellidos: {1}, Deporte: {2}",
nombre, apellidos, pais);
        }
    }
}

```

```

    }
}

static void QuintaConsulta()
{
    string mostrar = @"SELECT deportistas.nombre, deportistas.apellidos,
deportes.nombre FROM deportistas
                                LEFT OUTER JOIN practicar ON deportistas.codigo =
practicar.codDeportista
                                LEFT OUTER JOIN deportes ON practicar.codDeporte =
deportes.codigo";
    SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
    SQLiteDataReader datos = cmd.ExecuteReader();

    while (datos.Read())
    {
        string nombre = Convert.ToString(datos[0]);
        string apellidos = Convert.ToString(datos[1]);
        string deporte = Convert.ToString(datos[2]);
        Console.WriteLine("Nombre: {0}, Apellidos: {1}, Deporte: {2}",
nombre, apellidos, deporte);
    }
}

static void SextaConsulta()
{
    string mostrar = @"SELECT deportes.nombre, deportistas.nombre,
deportistas.apellidos FROM deportes
                                LEFT OUTER JOIN practicar ON deportes.codigo =
practicar.codDeporte
                                LEFT OUTER JOIN deportistas ON
practicar.codDeportista = deportistas.codigo";
    SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
    SQLiteDataReader datos = cmd.ExecuteReader();

    while (datos.Read())
    {
        string deporte = Convert.ToString(datos[0]);
        string deportista = Convert.ToString(datos[1]);
        Console.WriteLine("Deporte: {0}, Deportista: {1}", deporte,
deportista);
    }
}

static void CerrarBBDD()
{
    conexion.Close();
}
}
}

```