

EJERCICIOS RESUELTOS BBDD TEMA 12

12.1. Partiendo de la base de datos "ejercicio11", que tenía información sobre deportistas y disciplinas deportivas: muestra en una misma consulta los nombres y apellidos de todos los deportistas, además de los nombres de las disciplinas deportivas.

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

namespace _12._1
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\\..\\..\\Ejercicio11.sqlite; " +
                "Version = 3; New = False; Compress = True");
            conexion.Open();

            string mostrar = @"SELECT nombre, apellidos FROM deportistas
                                union
                                SELECT nombre, ' ' FROM deportes";
            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]);
                Console.WriteLine("{0}, {1}", nombre, apellidos);
            }
            conexion.Close();
            Console.ReadLine();
        }
    }
}
```

12.2. 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. Deben aparecer todos los países y todos los deportistas, aunque alguno pudiera estar sin relacionar (exista algún deportista del que no conozcamos el país o algún país del que no tengamos anotados aún deportistas).

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

namespace _12._2
{
    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
                                INNER JOIN países ON países.codigo =
deportistas.país
                                union
                                SELECT deportistas.nombre, apellidos, '' FROM
deportistas WHERE país IS NULL
                                union
                                SELECT '', '', países.nombre FROM países
                                LEFT OUTER JOIN deportistas ON
países.codigo = deportistas.país WHERE deportistas.codigo IS NULL";
        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 país = Convert.ToString(datos[2]);
            Console.WriteLine("{0}, {1}, {2}", nombre, apellidos, país);
        }
        conexion.Close();
        Console.ReadLine();
    }
}

```

12.3. Muestra los nombres y apellidos de todos los deportistas, junto al nombre de la disciplina deportiva que practican. Deben aparecer todas las disciplinas y todos los deportistas, aunque alguno pudiera estar sin relacionar.

```

using System;
using System.Data.SQLite;

namespace _12._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,
deportes.nombre FROM deportistas
                                INNER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                                INNER JOIN deportes ON
deportes.codigo = practicar.codDeporte
                                union
                                SELECT deportistas.nombre, apellidos, '' FROM
deportistas
                                LEFT OUTER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                                WHERE practicar.codDeporte IS NULL
                                union
                                SELECT '', '', deportes.nombre FROM deportes

```

```

                                LEFT OUTER JOIN practicar ON
deportes.codigo = practicar.codDeporte WHERE practicar.codDeportista IS NULL";
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("{0}, {1}, {2}", nombre, apellidos, deporte);
}
conexion.Close();
Console.ReadLine();
}
}
}

```

12.4. Muestra los nombres y apellidos de todos los deportistas (en formato "Apellido, nombre"), junto al nombre de la disciplina deportiva que practican. Deben mostrarse todas las disciplinas y todos los deportistas, aunque alguno pudiera estar sin relacionar. Los datos deben aparecer ordenados según el apellido del deportista.

```

using System;
using System.Data.SQLite;

namespace _12._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 apellidos || ', ' || deportistas.nombre as
nombreJug, deportes.nombre FROM deportistas
                                INNER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                                INNER JOIN deportes ON deportes.codigo
= practicar.codDeporte
                                union
                                SELECT apellidos || ', ' || deportistas.nombre as
nombreJug, '' FROM deportistas
                                LEFT OUTER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                                WHERE practicar.codDeporte IS NULL
                                union
                                SELECT '', deportes.nombre FROM deportes
                                LEFT OUTER JOIN practicar ON
deportes.codigo = practicar.codDeporte WHERE practicar.codDeportista IS NULL";

            //Esta parte no es necesaria en SQLITE porque por
            //defecto ordena por la primera columna ascendentemente.
            //ORDER BY nombreJug ASC";
            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]);
            Console.WriteLine("{0}, {1}", nombre, apellidos);
        }
        conexion.Close();
        Console.ReadLine();
    }
}
}

```

12.5. Unifica todos los ejercicios en una misma clase.

```

using System;
using System.Data.SQLite;

namespace _12._5
{
    class Program
    {
        static SQLiteConnection conexion;

        static void Main()
        {
            AbrirBBDD();
            Console.WriteLine("Se mostrarán todos los deportistas y todos los deportes en una misma consulta: ");
            PrimeraConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
            Console.WriteLine("Se mostrarán todos los deportistas y el país al que pertenecen en una misma consulta: ");
            SegundaConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
            Console.WriteLine("Se mostrarán todos los deportistas y todos los deportes, aunque no haya relación: ");
            TerceraConsulta();
            Console.WriteLine("Pulsa Intro para continuar...");
            Console.ReadLine();
            Console.WriteLine("Se mostrarán todos los deportistas y todos los deportes, aunque no haya relación: ");
            CuartaConsulta();
            Console.WriteLine("Pulsa Intro para acabar...");
            Console.ReadLine();
            CerrarBBDD();
        }

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

        static void PrimeraConsulta()
        {
            string mostrar = @"SELECT nombre, apellidos FROM deportistas
                                union

```

```

        SELECT nombre, '' FROM deportes";

        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]);
            Console.WriteLine("{0}, {1}", nombre, apellidos);
        }
    }

    static void SegundaConsulta()
    {
        string mostrar = @"SELECT deportistas.nombre, apellidos,
países.nombre FROM deportistas
                                INNER JOIN países ON países.codigo =
deportistas.país
                                union
                                SELECT deportistas.nombre, apellidos, '' FROM
deportistas WHERE país IS NULL
                                union
                                SELECT '', '', países.nombre FROM países
                                LEFT OUTER JOIN deportistas ON
países.codigo = deportistas.país WHERE deportistas.codigo IS NULL";
        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 país = Convert.ToString(datos[2]);
            Console.WriteLine("{0}, {1}, {2}", nombre, apellidos, país);
        }
    }

    static void TerceraConsulta()
    {
        string mostrar = @"SELECT deportistas.nombre, apellidos,
deportes.nombre FROM deportistas
                                INNER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                                INNER JOIN deportes ON
deportes.codigo = practicar.codDeporte
                                union
                                SELECT deportistas.nombre, apellidos, '' FROM
deportistas
                                LEFT OUTER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                                WHERE practicar.codDeporte IS NULL
                                union
                                SELECT '', '', deportes.nombre FROM deportes
                                LEFT OUTER JOIN practicar ON
deportes.codigo = practicar.codDeporte WHERE practicar.codDeportista IS NULL";
        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("{0}, {1}, {2}", nombre, apellidos, deporte);
    }

    static void CuartaConsulta()
    {
        string mostrar = @"SELECT apellidos || ', ' || deportistas.nombre as
nombreJug, deportes.nombre FROM deportistas
                        INNER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                        INNER JOIN deportes ON deportes.codigo
= practicar.codDeporte
                        union
                        SELECT apellidos || ', ' || deportistas.nombre as
nombreJug, '' FROM deportistas
                        LEFT OUTER JOIN practicar ON
practicar.codDeportista = deportistas.codigo
                        WHERE practicar.codDeporte IS NULL
                        union
                        SELECT '', deportes.nombre FROM deportes
                        LEFT OUTER JOIN practicar ON
deportes.codigo = practicar.codDeporte WHERE practicar.codDeportista IS NULL";

        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]);
            Console.WriteLine("{0}, {1}", nombre, apellidos);
        }
    }

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

```