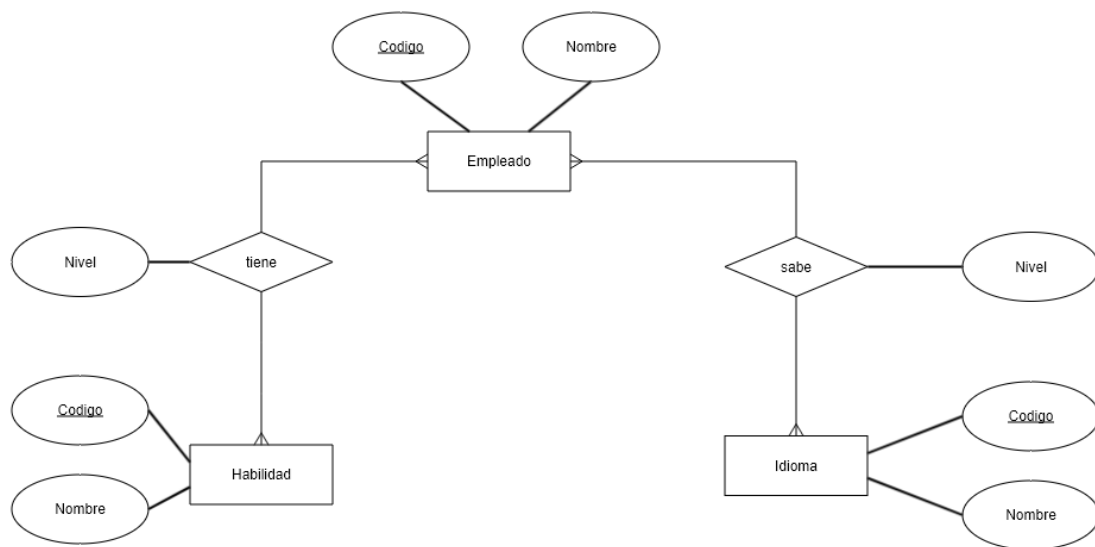


EJERCICIOS RESUELTOS BBDD TEMA 3

3.1. Deseamos informatizar una lista de empleados técnicos de nuestra empresa, que sean capaces de resolver problemas de nuestros clientes. Por eso, para cada empleado nos interesará guardar información sobre todas sus habilidades técnicas (por ejemplo, "bases de datos" o "programación") así como los idiomas que maneja con soltura (por ejemplo, "inglés" o "alemán"). Como puede haber varias personas que tengan una cierta habilidad técnica o que hablen un cierto idioma, usaremos tablas para esos datos y relaciones "muchos a muchos". Además, queremos valorar de 1 a 5 el nivel que cada empleado tiene con una habilidad técnica o con un idioma. Crea un diagrama Entidad-Relación que muestre cómo automatizar este sistema de información.



3.2. Convierte a tablas el sistema Entidad-Relación, dentro de una nueva base de datos llamada "ejercicio3".

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

namespace _3._2
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source = 
..\\..\\..\\Ejercicio3.sqlite; " +
                "Version = 3; New = True; Compress = True;");
            conexion.Open();

            string crear = @"CREATE TABLE Empleado (Nombre varchar (30), 
Codigo varchar (5) PRIMARY KEY);
CREATE TABLE Habilidad (Nombre varchar (30), 
Codigo varchar (5) PRIMARY KEY);
CREATE TABLE Idioma (Nombre varchar (30), 
Codigo varchar (5) PRIMARY KEY);";
```

```

        CREATE TABLE HabilidadDeEmpleado (Nivel numeric (2),
CodigoEmpleado varchar (5), CodigoHabilidad varchar (5) );
        CREATE TABLE IdiomaDeEmpleado (Nivel numeric (2),
CodigoEmpleado varchar (5), CodigoHabilidad varchar (5) );
        SQLiteCommand cmd = new SQLiteCommand(crear, conexion);
        cmd.ExecuteNonQuery();

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

```

3.3. Añade a los usuarios (y habilidades e idiomas) los siguientes datos:

- Aurora, con nivel de 5 estrellas en PHP, 4 estrellas en Javascript, 5 estrellas en diseño gráfico y 3 estrellas en idioma inglés.
- Adrián, con nivel de 4 estrellas en PHP, 4 estrellas en Javascript, 5 estrellas en montaje de equipos y 2 estrellas en idioma inglés.
- Enrique, con 5 estrellas en electrónica y 2 estrellas en idioma inglés.
- Gala, con 5 estrellas en inglés, 5 estrellas en francés y 5 estrellas en atención al cliente.

```

using System;
using System.Data.SQLite;

namespace _3._3
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\\..\\..\\Ejercicio3.sqlite; " +
                "Version = 3; New = False; Compress = True;");
            conexion.Open();

            string insertar = @"INSERT INTO Empleado VALUES ('Aurora', 'AUR'),
('Adrián', 'ADR'), ('Enrique', 'ENR'), ('Gala', 'GAL');
            INSERT INTO Habilidad VALUES ('PHP', 'PHP'),
('JavaScript', 'JS'), ('Diseño Gráfico', 'DG'),
                ('Montaje de Equipos', 'ME'), ('Electrónica',
'ELEC'), ('Atención al cliente', 'AC');
            INSERT INTO Idioma VALUES ('Inglés', 'ING'),
('Francés', 'FRA');
            INSERT INTO HabilidadDeEmpleado VALUES (5, 'AUR',
'PHP'), (4, 'AUR', 'JS'),
                (5, 'AUR', 'DG'), (4, 'ADR', 'PHP'), (4, 'ADR',
'JS'), (5, 'ADR', 'ME'),
                (5, 'ENR', 'ELEC'), (5, 'GAL', 'AC');
            INSERT INTO IdiomaDeEmpleado VALUES (3, 'AUR',
'ING'), (2, 'ADR', 'ING'),
                (2, 'ENR', 'ING'), (5, 'GAL', 'ING'), (5, 'GAL',
'FRA');

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

            conexion.Close();
        }
    }
}

```

```

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

```

3.4. Muestra el nombre de todas las personas que hablen francés.

```

using System;
using System.Data.SQLite;

namespace _3._4
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\\..\\..\\Ejercicio3.sqlite; " +
                "Version = 3; New = False; Compress = True;");
            conexion.Open();

            string mostrar = @"SELECT Nombre
                                FROM Empleado, IdiomaDeEmpleado
                                WHERE CodigoIdioma = 'FRA'
                                AND Empleado.Codigo =
IdiomaDeEmpleado.CodigoEmpleado";
            SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
            SQLiteDataReader datos = cmd.ExecuteReader();

            while (datos.Read())
            {
                string nombre = Convert.ToString(datos[0]);
                Console.WriteLine(nombre);
            }

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

```

3.5. Muestra los nombres de los empleados con conocimientos de diseño gráfico, ordenados del más experto (5 estrellas) al menos experto (1 estrella).

```

using System;
using System.Data.SQLite;

namespace _3._5
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\\..\\..\\Ejercicio3.sqlite; " +
                "Version = 3; New = False; Compress = True;");
            conexion.Open();

            string mostrar = @"SELECT Nombre

```

```

        FROM Empleado, HabilidadDeEmpleado
        WHERE CodigoHabilidad = 'DG'
        AND Empleado.Codigo =
HabilidadDeEmpleado.CodigoEmpleado
        ORDER BY HabilidadDeEmpleado.Nivel
DESC";

        SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
        SQLiteDataReader datos = cmd.ExecuteReader();

        while (datos.Read())
        {
            string nombre = Convert.ToString(datos[0]);
            Console.WriteLine(nombre);
        }

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

```

3.6. Muestra las habilidades de Adrián, ordenadas de aquella en la que es más experto a aquella en la que menos. Si dos habilidades coinciden, deberás ordenarlas alfabéticamente.

```

using System;
using System.Data.SQLite;

namespace _3._6
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\\..\\..\\Ejercicio3.sqlite; " +
                "Version = 3; New = False; Compress = True;");
            conexion.Open();

            string mostrar = @"SELECT Habilidad.Nombre
        FROM Empleado, HabilidadDeEmpleado, Habilidad
        WHERE Empleado.Nombre = 'Adrián'
        AND Empleado.Codigo =
HabilidadDeEmpleado.CodigoEmpleado
        AND
HabilidadDeEmpleado.CodigoHabilidad = Habilidad.Codigo
        ORDER BY
HabilidadDeEmpleado.Nivel DESC, Habilidad.Nombre";
            SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
            SQLiteDataReader datos = cmd.ExecuteReader();

            Console.WriteLine("Las habilidades de Adrián son: ");
            while (datos.Read())
            {
                string habilidades = Convert.ToString(datos[0]);
                Console.WriteLine(habilidades);
            }

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

```

```
}
```

3.7. Nos llama un cliente que sólo habla inglés y que quiere hacer una consulta técnica. Por eso, deberás obtener los nombres de los empleados con conocimientos de PHP y de inglés, ordenados de mayor a menor nivel de inglés, y, en caso de coincidir, de mayor a menor nivel de PHP.

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

namespace _3._7
{
    class Program
    {
        static void Main()
        {
            SQLiteConnection conexion = new SQLiteConnection("Data Source =
..\\..\\..\\Ejercicio3.sqlite; " +
                "Version = 3; New = False; Compress = True;");
            conexion.Open();

            string mostrar = @"SELECT Empleado.Nombre
                                FROM Empleado, IdiomaDeEmpleado,
HabilidadDeEmpleado
                                WHERE Empleado.Codigo =
IdiomaDeEmpleado.CodigoEmpleado
                                AND Empleado.Codigo =
HabilidadDeEmpleado.CodigoEmpleado
                                AND IdiomaDeEmpleado.CodigoIdioma =
'ING'
                                AND
HabilidadDeEmpleado.CodigoHabilidad = 'PHP'
                                ORDER BY
IdiomaDeEmpleado.Nivel DESC, HabilidadDeEmpleado.Nivel DESC ";
            SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
            SQLiteDataReader datos = cmd.ExecuteReader();

            while (datos.Read())
            {
                string nombre = Convert.ToString(datos[0]);
                Console.WriteLine(nombre);
            }

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

3.8. Unifica todos los ejercicios en la misma clase.

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

namespace _3._8
{
    class Program
    {
        static SQLiteConnection conexion;
```

```

static void Main()
{
    EstablecerConexionBBDD();
    CrearTablasBBDD();
    Console.WriteLine("Tablas creadas con éxito. Pulsa Intro para
continuar...");
    Console.ReadLine();
    InsertarDatosBBDD();
    Console.WriteLine("Datos insertados con éxito. Pulsa Intro para
continuar...");
    Console.ReadLine();
    Console.WriteLine("Muestra el nombre de los empleados que hablen
francés: ");
    PrimeraConsulta();
    Console.WriteLine("\nPulsa Intro para continuar...");
    Console.ReadLine();
    Console.WriteLine("Muestra el nombre de los empleados con
conocimiento de diseño gráfico, ordenados: ");
    SegundaConsulta();
    Console.WriteLine("\nPulsa Intro para continuar...");
    Console.ReadLine();
    Console.WriteLine("Las habilidades de Adrián son: ");
    TerceraConsulta();
    Console.WriteLine("\nPulsa Intro para continuar...");
    Console.ReadLine();
    Console.WriteLine("Mostrar los empleados con conocimientos de PHP y
de inglés, ordenados por nivel: ");
    CuartaConsulta();
    Console.WriteLine("\nPulsa Intro para acabar...");
    Console.ReadLine();
    CerrarBBDD();
}

static void EstablecerConexionBBDD()
{
    conexion = new SQLiteConnection("Data Source =
..\..\..\..\Ejercicio3.sqlite; " +
        "Version = 3; New = True; Compress = True;");
    conexion.Open();
}

static void CrearTablasBBDD()
{
    string crear = @"CREATE TABLE Empleado (Nombre varchar (30),Codigo
varchar (5) PRIMARY KEY);
CREATE TABLE Habilidad (Nombre varchar (30),Codigo
varchar (5) PRIMARY KEY);
CREATE TABLE Idioma (Nombre varchar (30),Codigo
varchar (5) PRIMARY KEY);
CREATE TABLE HabilidadDeEmpleado (Nivel numeric (2),
CodigoEmpleado varchar (5),CodigoHabilidad varchar (5) );
CREATE TABLE IdiomaDeEmpleado (Nivel numeric (2),
CodigoEmpleado varchar (5),CodigoIdioma varchar (5) );";
    SQLiteCommand cmd = new SQLiteCommand(crear, conexion);
    cmd.ExecuteNonQuery();
}

static void InsertarDatosBBDD()
{
    string insertar = @"INSERT INTO Empleado VALUES ('Aurora', 'AUR'),
('Adrián', 'ADR'), ('Enrique', 'ENR'), ('Gala', 'GAL');";
}

```

```

INSERT INTO Habilidad VALUES ('PHP', 'PHP'),
('JavaScript', 'JS'), ('Diseño Gráfico', 'DG'),
('Montaje de Equipos', 'ME'), ('Electrónica',
'ELEC'), ('Atención al cliente', 'AC');
INSERT INTO Idioma VALUES ('Inglés', 'ING'),
('Francés', 'FRA');
INSERT INTO HabilidadDeEmpleado VALUES (5, 'AUR',
(5, 'AUR', 'DG'), (4, 'ADR', 'PHP'), (4, 'ADR',
'JS'), (5, 'ADR', 'ME'),
(5, 'ENR', 'ELEC'), (5, 'GAL', 'AC');
INSERT INTO IdiomaDeEmpleado VALUES (3, 'AUR',
'ING'), (2, 'ADR', 'ING'),
(2, 'ENR', 'ING'), (5, 'GAL', 'ING'), (5, 'GAL',
'FRA')";

    SQLiteCommand cmd = new SQLiteCommand(insertar, conexion);
    cmd.ExecuteNonQuery();
}

static void PrimeraConsulta()
{
    string mostrar = @"SELECT Nombre
                        FROM Empleado, IdiomaDeEmpleado
                        WHERE CodigoIdioma = 'FRA'
                        AND Empleado.Codigo =
IdiomaDeEmpleado.CodigoEmpleado";
    SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
    SQLiteDataReader datos = cmd.ExecuteReader();

    while (datos.Read())
    {
        string nombre = Convert.ToString(datos[0]);
        Console.WriteLine(nombre);
    }
}

static void SegundaConsulta()
{
    string mostrar = @"SELECT Nombre
                        FROM Empleado, HabilidadDeEmpleado
                        WHERE CodigoHabilidad = 'DG'
                        AND Empleado.Codigo =
HabilidadDeEmpleado.CodigoEmpleado
                        ORDER BY HabilidadDeEmpleado.Nivel
DESC";
    SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
    SQLiteDataReader datos = cmd.ExecuteReader();

    while (datos.Read())
    {
        string nombre = Convert.ToString(datos[0]);
        Console.WriteLine(nombre);
    }
}

static void TerceraConsulta()
{
    string mostrar = @"SELECT Habilidad.Nombre
                        FROM Empleado, HabilidadDeEmpleado, Habilidad
                        WHERE Empleado.Nombre = 'Adrián'
                        AND Empleado.Codigo =
HabilidadDeEmpleado.CodigoEmpleado

```

```

                                AND
HabilidadDeEmpleado.CodigoHabilidad = Habilidad.Codigo
                                ORDER BY
HabilidadDeEmpleado.Nivel DESC, Habilidad.Nombre";
    SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
    SQLiteDataReader datos = cmd.ExecuteReader();

    while (datos.Read())
    {
        string habilidades = Convert.ToString(datos[0]);
        Console.WriteLine(habilidades);
    }
}

static void CuartaConsulta()
{
    string mostrar = @"SELECT Empleado.Nombre
                        FROM Empleado, IdiomaDeEmpleado,
HabilidadDeEmpleado
                        WHERE Empleado.Codigo =
IdiomaDeEmpleado.CodigoEmpleado
                        AND Empleado.Codigo =
HabilidadDeEmpleado.CodigoEmpleado
                        AND IdiomaDeEmpleado.CodigoIdioma =
'ING'
                        AND
HabilidadDeEmpleado.CodigoHabilidad = 'PHP'
                        ORDER BY
IdiomaDeEmpleado.Nivel DESC, HabilidadDeEmpleado.Nivel DESC ";
    SQLiteCommand cmd = new SQLiteCommand(mostrar, conexion);
    SQLiteDataReader datos = cmd.ExecuteReader();

    while (datos.Read())
    {
        string nombre = Convert.ToString(datos[0]);
        Console.WriteLine(nombre);
    }
}

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

```