

SQL Anywhere および ADO.NET Entity Framework

SQL Anywhere の Microsoft データ・アクセス・スタンダードに対応可能な優れた性能をベースとして、オブジェクト・リレーショナル・マッピング技術である ADO.NET 3.5 および ADO.NET Entity Framework に対するサポート機能が SQL Anywhere Ver.11 に追加されました。また、Visual Studio 2008 では、SQL Anywhere Explorer に対するサポート機能も装備され、Visual Studio 環境を維持しながら SQL Anywhere および Ultra Light アプリケーションの開発を行うことができるようになりました。

SQL Anywhere ADO.NET Entity Framework のチュートリアル

このチュートリアルでは、SQL Anywhere Ver.11 が ADO.NET EDM (Entity Data Model: エンティティ・データ・モデル) からデータを取り込む性能について実証します。使用するメソッドは、データ・ソース・オブジェクトに対する結合制御や、EntityClient プロバイダ、オブジェクト・サービス、エンティティに対する LINQ などです。

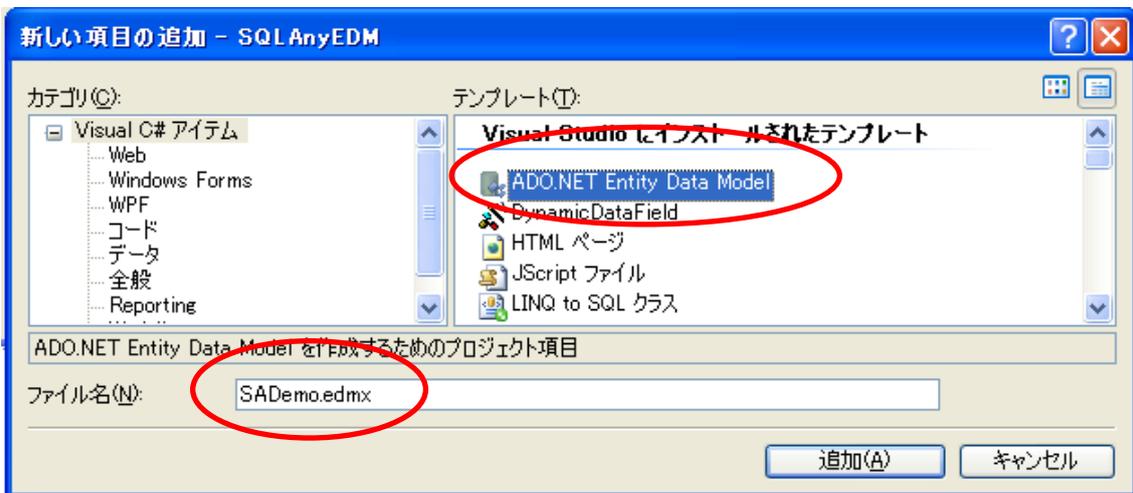
要求事項

- Visual Studio 2008 with Service Pack 1
- Microsoft .NET Framework 3.5 with Service Pack 1
- SQL Anywhere 11.0.1 または、以降のバージョン

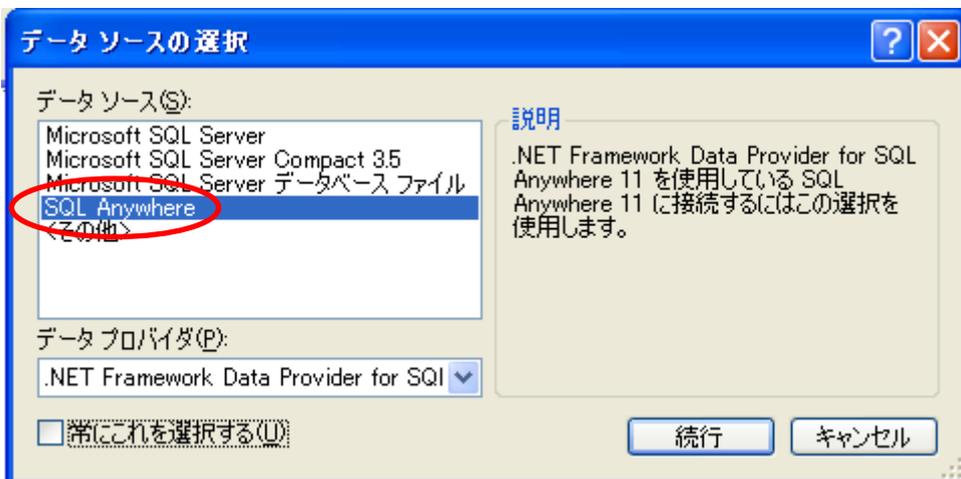
Visual Studio 用の SQL Anywhere 統合コンポーネントが正しくインストールされていることを確認してください。Visual Studio 2008 を起動して、[ツール] メニューで [Sybase Central] および [Interactive SQL] を探します。これらの項目が見つからない場合は、SQL Anywhere と Visual Studio を統合する必要があります。Visual Studio を終了してコマンド・プロンプトを開き、C:¥Program Files¥SQL Anywhere 11¥Assembly¥v2¥SetupVSPackage.exe を実行します。

SQL Anywhere Demo データベース用の EDM を作成

1. Visual Studio 2008 を起動して、新規で Visual C# **Windows フォーム アプリケーション**を作成します。プロジェクト名を *SQLAnyEDM* とします。
2. 新しい項目 **ADO.NET Entity Data Model** を追加し、このファイル名を *SADemo.edmx* とします。



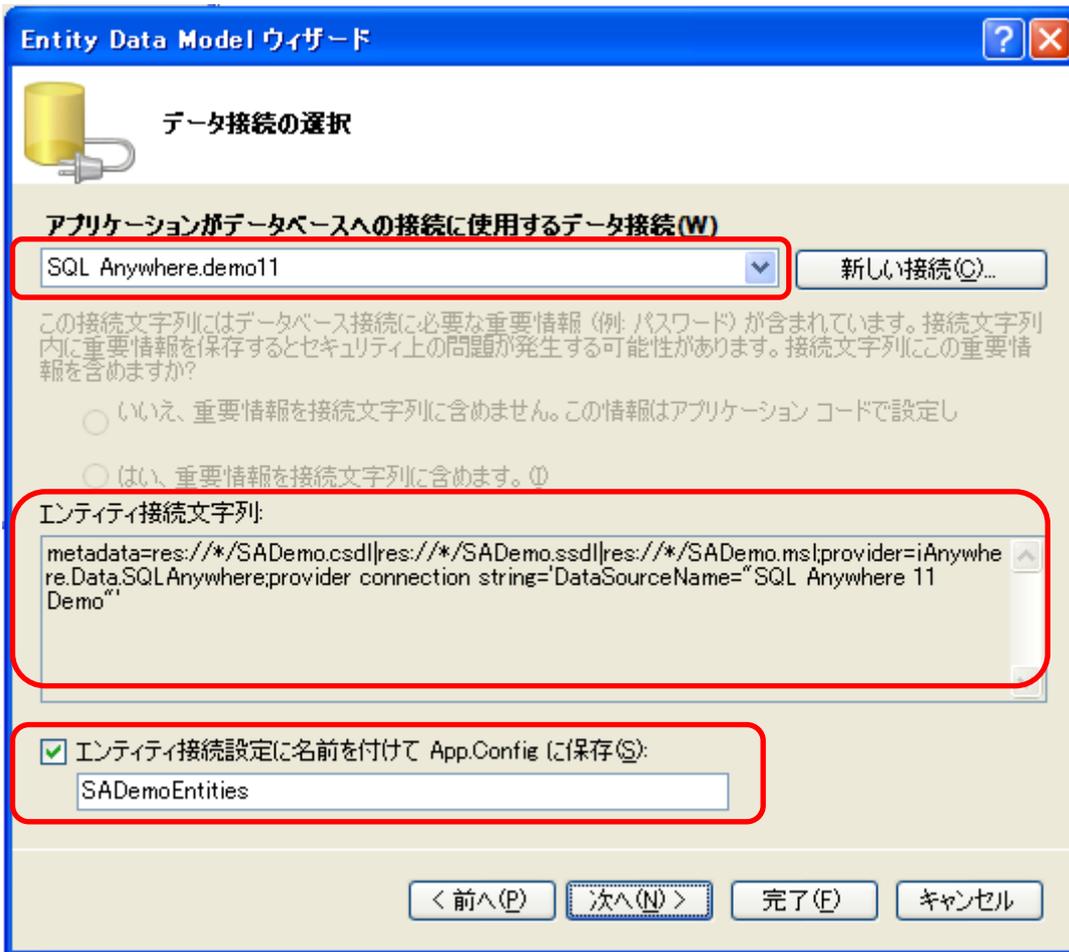
1. [Entity Data Model] ウィザードが表示されます。[データベースから作成] を選択し、[次へ] をクリックします。
2. 新しいデータ接続または SQL Anywhere を作成します。[新しい接続] ボタンをクリックします。
3. [接続のプロパティ] ダイアログで [変更] ボタンをクリックします。
4. リストから [SQL Anywhere] を選択し、[OK] をクリックします。



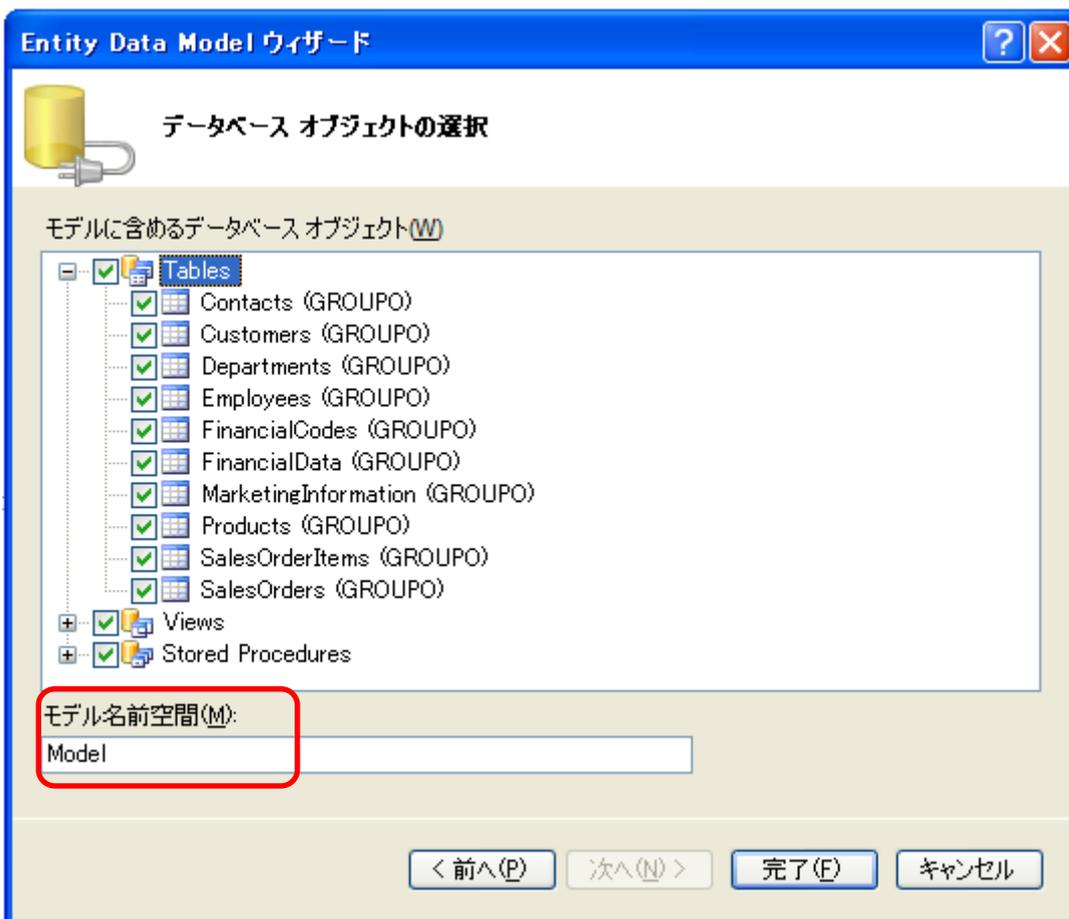
1. [ODBC Data Source name] を選択し、リストから [SQL Anywhere 11 Demo] を選択します。

必要な場合は、接続テストを行ってください。[OK] をクリックして、ダイアログを終了します。

1. [EDM] ウィザードに戻り、接続文字列で SQL Anywhere.NET データ・プロバイダが使用されていることを確認してください。接続の設定を [SADemoEntities] に変更し、[次へ] をクリックします。



1. すべてのデータベース・オブジェクトをモデルに含めます。モデルのネームスペースを [SADemoModel] に変更します。[完了] をクリックしてウィザードを終了します。

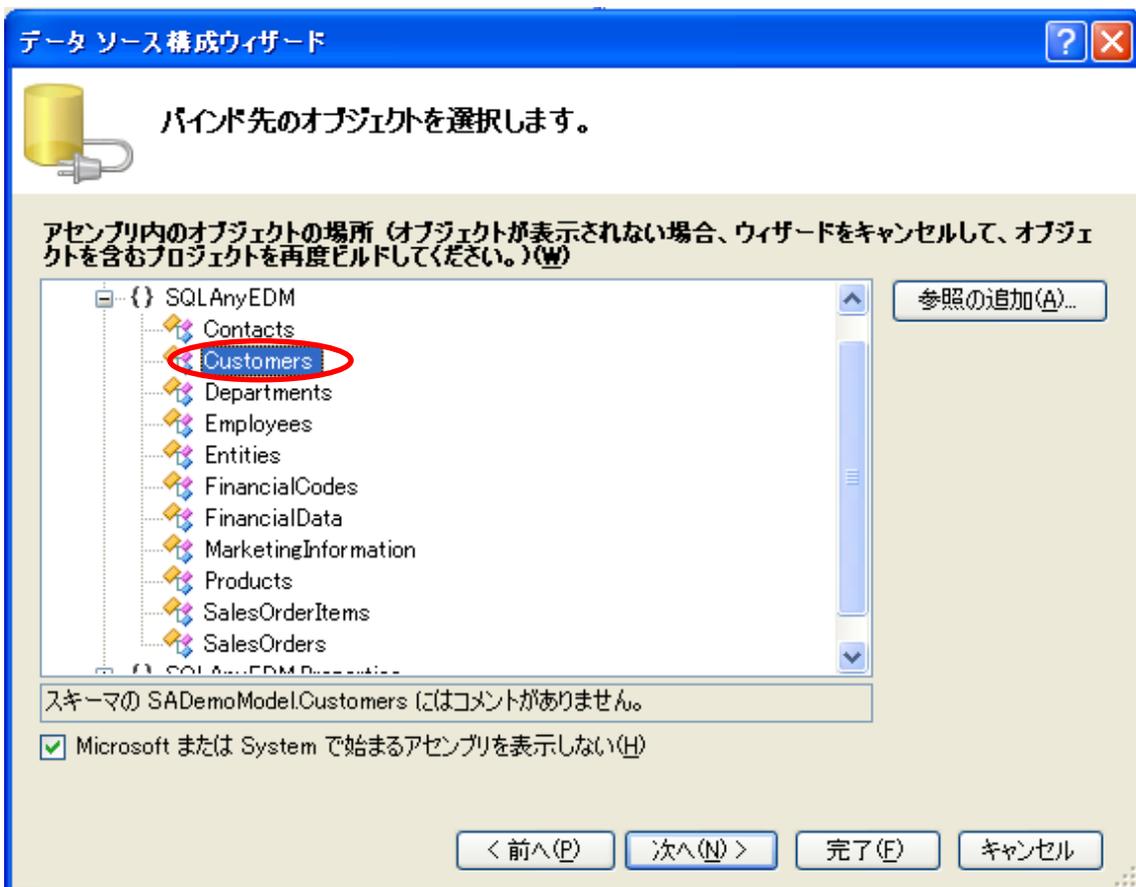


これで、SQL Anywhere Demo データベースの EDM が完成しました。ファイル [SADemo.edmx] をダブルクリックすると、このモデルが視覚的に表示されます。

SADemo EDM からデータ・ソースを作成

このチュートリアルでは、データ・グリッドを使用して顧客情報を表示します。このデータ・グリッドは、EDM から生成されたデータ・ソースに対して結合されています。

1. [データ] メニューから、[データソースの表示] を選択します。
2. [データソース] ウィンドウで、[新しいデータソースの追加] をクリックします。[データソース構成] ウィザードが表示されます。
3. [オブジェクト] を選択して、[次へ] をクリックします。
4. EDM を展開して [Customers] を選択し、[次へ] をクリックします。



1. **[完了]** をクリックして、ウィザードを終了します。

グリッド・ビューでデータベース・レコードを表示

データを取り込んで表示する方法の 1 つとして、データ・ソース・オブジェクトと、グリッド・ビューなどの制御を結合させる方法があります。

1. **[データソース]** ウィンドウから **[Customers]** オブジェクトをドラッグし、メイン・フォーム **[Form1]** にドロップします。これで **DataGridView** が作成され、**BindingNavigator** が制御して **[Customers]** オブジェクトに結合されます。
2. 必要な C# コードを追加して、データベースから顧客情報を取り込みます。**[Form1]** をダブルクリックして、コード・エディタを開きます。
3. フォーム用の **[Load]** イベントに、以下のコードを入力します。

```
// SADemoModel.SADemoEntities をベースとして、ObjectContext インスタンスを作成する  
var saEntities = new SQLAnyEDM.SADemoEntities();
```

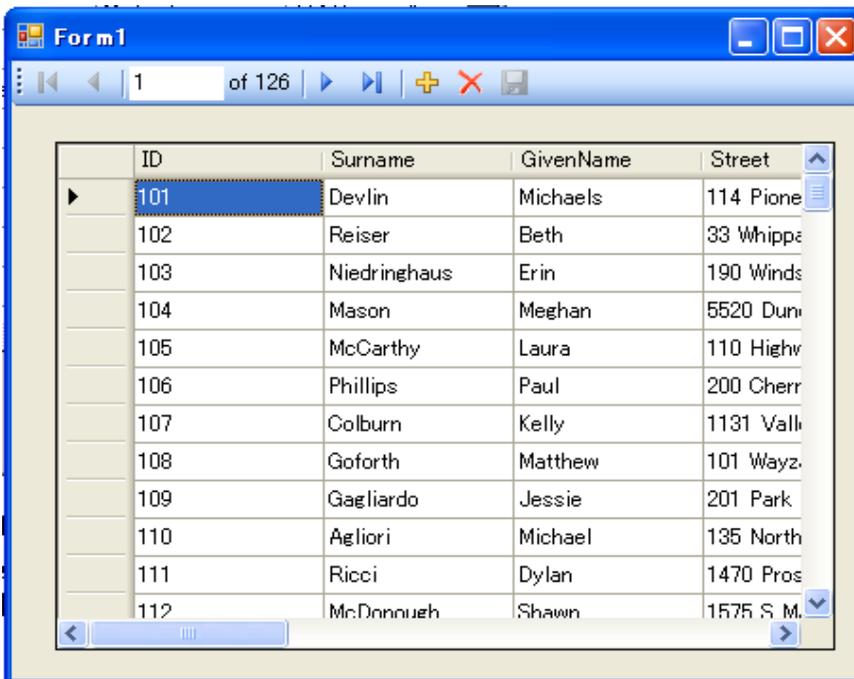
```
// エンティティから顧客リストを取り込む
```

```
var customers = saEntities.Customers;
```

```
// Bind the customers object to the data grid
```

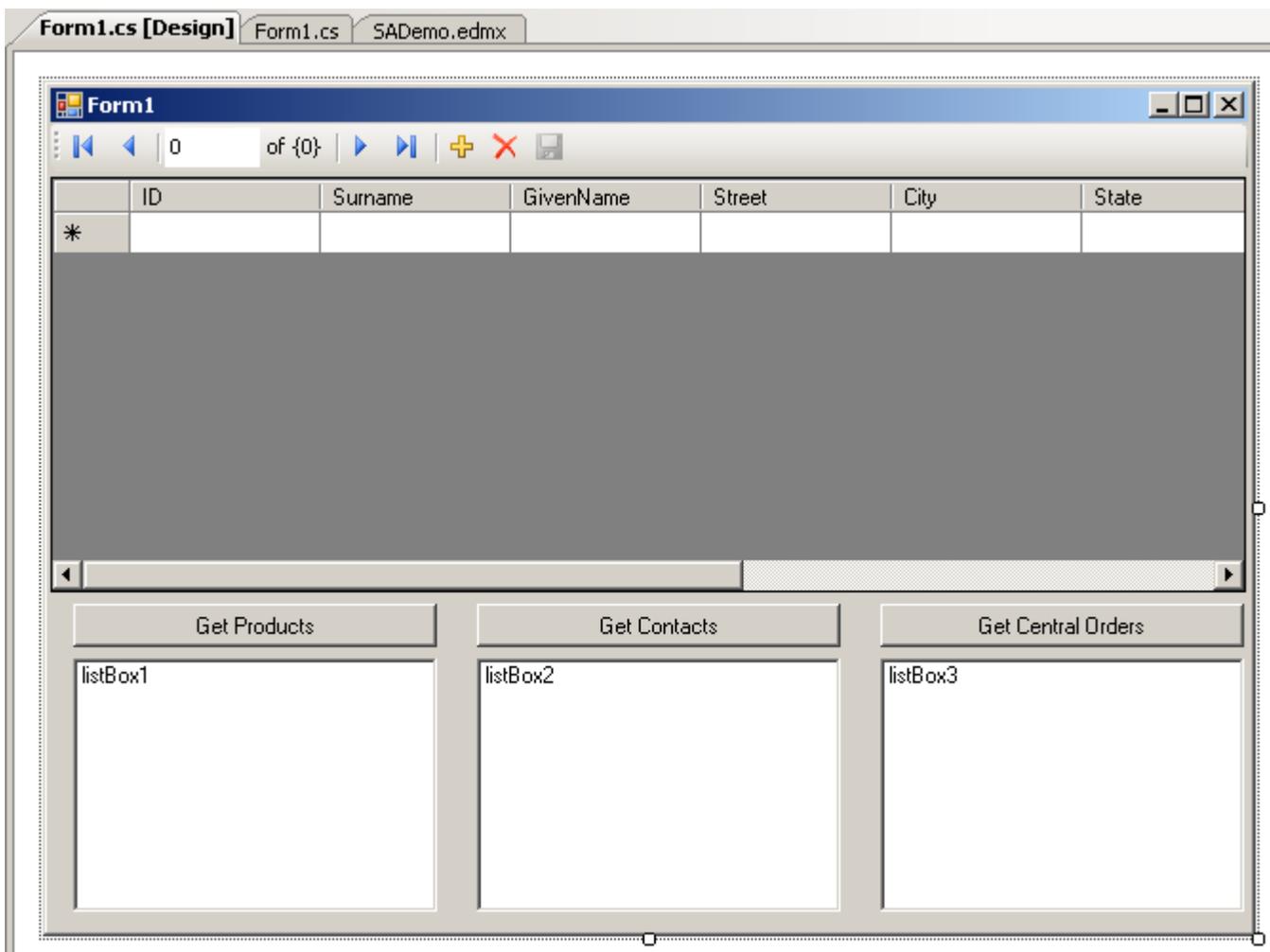
```
customersBindingSource.DataSource = customers;
```

4. [F5] を押してプロジェクトを実行します。この時、フォームに顧客情報が含まれていることを確認してください。



ID	Surname	GivenName	Street
101	Devlin	Michaels	114 Pione
102	Reiser	Beth	33 Whippe
103	Niedringhaus	Erin	190 Winds
104	Mason	Meghan	5520 Dum
105	McCarthy	Laura	110 Highv
106	Phillips	Paul	200 Cherr
107	Colburn	Kelly	1131 Vall
108	Goforth	Matthew	101 Wayz.
109	Gagliardo	Jessie	201 Park
110	Agliori	Michael	135 North
111	Ricci	Dylan	1470 Pros
112	McDonnueh	Shawn	1575 S. M.

1. アプリケーションを終了します。
2. さらに制御をいくつか追加して、他の情報も表示させます。コマンド・ボタンとリスト・ボックス・コントロールを追加すると、フォームが以下のように表示されます。



このチュートリアル・アプリケーション用のユーザ・インターフェースは、これで完成です。次は、SQL Anywhere データベースから追加情報を取り込むためのコマンド・ボタンにコードを追加する手順について説明します。

EntityClient プロバイダを使用したデータの取り込み

他にも、EntityClient プロバイダを使用してデータを取り込む方法があります。EDM では、Entity SQL を使用してクエリが実行されます。

1. **[Get Products]** ボタンをダブルクリックして、コード・エディタを開きます。
2. ボタンの *[Click]* イベントに、以下のコードを入力します。

```
var saEntities = new SQLAnyEDM.SADemoEntities();
```

```

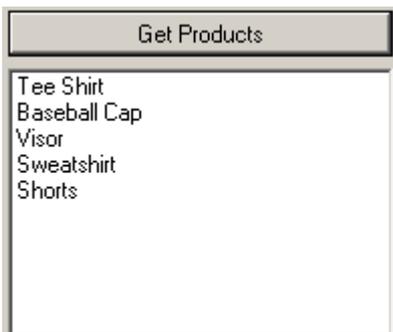
// EntityClient プロバイダおよび Entity SQL を使用して、EDM にクエリを実
// 行する
var saConn = new
System.Data.EntityClient.EntityConnection("Name=SADEMOEntities"
);
var saCmd = new System.Data.EntityClient.EntityCommand(
    @"select distinct p.Name
    from SAdemoEntities.Products as p", saConn);

saConn.Open();
var saReader =
    saCmd.ExecuteReader(System.Data.CommandBehavior.SequentialAccess);

// 結果セットをループし、製品名をリスト・ボックスに追加する
while (saReader.Read())
{
    listBox1.Items.Add(saReader.GetString(0));
} // while

```

- Entity SQL のクエリを確認してください。このクエリは、ユニークな製品リストをデータベースに返します。
- 製品を実行して **[Get Products]** ボタンをクリックし、情報を取り込みます。



- アプリケーションを終了します。

EntityClient プロバイダでは、demo EDM に対して単純にクエリを実行します。EntityFramework を使用しなければ、クエリが SQL Anywhere の接続オブジェクトに対して実行されます。

オブジェクト・サービスを使用したデータの取り込み

オブジェクト・サービスを使用してデータを取り込むことも可能です。この方法の場合も、Entity SQL を使用します。

1. **[Get Contacts]** ボタンをダブルクリックして、コード・エディタを開きます。
2. ボタンの *[Click]* イベントに、以下のコードを入力します。

```
var saEntities = new SQLAnyEDM.SADemoEntities();

// オブジェクト・サービスを使用して、EDM にクエリを実行する
var contacts =
saEntities.CreateQuery<System.Data.Common.DbDataRecord>(
    @"select c.GivenName, c.Surname
    from SADemoEntities.Contacts as c");

// 担当者の名前をリスト・ボックスに追加する
foreach (var record in contacts)
{
    listBox2.Items.Add(
        record.GetString(0) + " " + record.GetString(1));
} // foreach
```

3. 製品を実行して **[Get Products]** ボタンをクリックし、情報を取り込みます。



5. アプリケーションを終了します。

ここでは、オブジェクト・サービスと Entity SQL を使用して、demo EDM のクエリを実行しています。その後で結果セットをループし、必要に応じて、返されたデータを解析します。

エンティティの LINQ を使用したデータの取り込み

最後に、エンティティの LINQ を使用してデータを取り込む方法について説明します。

1. **[Get Central Orders]** ボタンをダブルクリックし、コード・エディタを開きます。
2. ボタンの *[Click]* イベントに、以下のコードを入力します。

```
var saEntities = new SQLAnyEDM.SADemoEntities();

// エンティティの LINQ を使用して、EDM にクエリを実行する
var orders = from o in saEntities.SalesOrders
              where o.Region == "Central"
              select o;

// 発注番号と日付をリスト・ボックスに追加する
foreach (var so in orders)
{
    listBox3.Items.Add(
        "Order #" + so.ID + " placed on " + so.OrderDate);
} // foreach
```

3. 製品を実行して、**[Get Central Orders]** ボタンをクリックし、情報を取り込みます。



5. アプリケーションを終了します。

エンティティの LINQ を使用して、オーダー・リストを取り込みます。Visual Studio の IntelliSense 機能のメリットを活用することで、demo EDM のさまざまなプロパティを掘り下げて表示できるため、クエリの構築プロセスを簡略化することができます。

結論

SQL Anywhere Ver.11 は、開発者の .NET プログラミングに関するスキルを活かしながら、強力なデータベース・アプリケーションを構築することができます。 .NET Entity Framework に対するサポート機能が追加されたことにより、新規または既存の SQL Anywhere データベースに基づいて、EDM を迅速に生成することが可能です。コントロールに対するデータ結合や、EntityClient プロバイダ、オブジェクト・サービス、エンティティの LINQ などといった数々の方法を使って、EDM から情報を取り込むことができます。