チュートリアル: SQL Anywhere を使った ASP.NET Web ページの作成

はじめに	1
必要なソフトウェア	1
概要	2
SQL Anywhere ASP.NET プロバイダのインストール	3
SQL Anywhere ASP.NET プロバイダのスキーマの表示	6
ASP.NET Web サイトの作成とエンティティ・データ・モデルの追加	9
データを表示するための Web コントロールの作成	14
SQL Anywhere ASP.NET Data Provider の構成	17
認証済みユーザに対する個人情報の表示	21
Dynamic Data 機能の有効化	28
SQL Anywhere ASP.NET プロバイダの削除	31
まとめとその他のリソース	32

改訂履歴

バージョン 1.1 - 2010 年 7 月

SQL Anywhere を使った ASP.NET Web ページの作成 バージョン 1.1 - 2009 年 3 月

i

はじめに

このホワイトペーパーでは、SQL Anywhere および Visual Studio 2010 を使ってデータベース駆動型の ASP.NET Web サイトを構築する方法を説明します。Web アプリケーションは、単純な企業ディレクトリです。パブリック・ペー ジには、一般的な従業員情報が表示され、パスワードで保護されるセキュア・ページには、詳細な情報が表示され ます。セキュア・ページのデータは、正しい認証情報を使ってログインしてからでないと表示されません。SQL Anywhere ASP.NET プロバイダの使用は、セキュリティ・メカニズムの実装方法として注目されています。各項で は、SQL Anywhere ASP.NET プロバイダのインストール、構成、および設定方法と、SQL Anywhere がアプリケーシ ョンのセキュリティ情報をどのようにデータベースに格納するかなどについて説明します。また、EntityDataSource オブジェクトをデータバインド・サーバ・コントロールにバインドする方法と、.NET Framework 4.0 の新しい Dynamic Data 機能をサンプル Web サイトに統合する方法についても説明します。

必要なソフトウェア

- SQL Anywhere 12 以上 以下の場所から SQL Anywhere Developer Edition を無料で入手できます。 http://www.ianywhere.jp/dl/dl_evl.html
- ・ Microsoft Visual Studio 2010 および .NET Framework 4.0
- オプションのソース・コード (C# および Visual Basic)

概要

このチュートリアルの内容は、以下のとおりです。

- SQL Anywhere ASP.NET プロバイダをインストールします。
- デモ・データベースに接続し、Sybase Central を使って SQL Anywhere ASP.NET プロバイダのスキーマを表示します。
- ASP.NET Web サイトを作成します。この項では Web サイトを作成し、SQL Anywhere データベース から Web サイトに EDM (Entity Data Model:エンティティ・データ・モデル)を追加します。
- GridView Web コントロールを作成し、EntityDataSource オブジェクトにバインドします。
 GridView コントロールは EDM からデータを取得し、パブリック・ページにそのデータを表示します。
- SQL Anywhere ASP.NET プロバイダを構成します。この項では、web.config ファイルにおける プロバイダの登録方法と ASP.NET 管理ツールの使用方法について説明します。
- 会員専用ページを作成します。このページの情報は、認証されたユーザのみアクセス 可能です。
- ASP.NET 4.0 の Dynamic Data 機能を使って、データの表示をフォーマットします。

SQL Anywhere ASP.NET プロバイダのインストール

SQL Anywhere ASP.NET プロバイダを使用すれば、SQL Anywhere データベースをバックエンド記憶領域として 使用するセキュリティ・メカニズムを構築できます。ユーザの機密情報の格納および管理のために、必要なスキ ーマを所定のデータベースに追加するためには、SQL Anywhere ASP.NET プロバイダをインストールする必要が あります。

SQL Anywhere には、以下の5種類のプロバイダが含まれています。

- Membership Provider:認証および承認サービスを提供します。
- Roles Provider:役割(ロール)の作成、役割へのユーザの追加、役割の削除に対応するメソッドを提供します。 グループへのユーザの割り当てやパーミッションの管理を行います。
- Profiles Provider : ユーザの基本設定など、ユーザ情報の読み取り、格納、および取得に対応するメソッドを提供します。
- Web Parts Personalization Provider : カスタマイズしたコンテンツや Web ページのレイアウトのロードと格納に対応するメソッドを提供します。
- Health Monitoring Provider : 配備済み Web アプリケーションのステータス監視に対応するメソッドを提供します。

SQL Anywhere ASP.NET プロバイダの詳細については、以下のオンライン・マニュアルを参照してください。

http://dcx.sybase.com/index.html#1201/ja/dbprogramming/aspdotnet-providers.html

SQL Anywhere には、必要なセキュリティ・スキーマをデータベースに追加するためのセットアップ・ウィザードも 用意されています。SQL Anywhere ASP.NET プロバイダは、新しいデータベースにも、既存のデータベースにも追 加できます。わかりやすくするために、以下の手順では SQL Anywhere デモ・データベース (http://dcx.sybase.com/index.html#1201/ja/saintro/fg-sademo.html) を使用します。

手順

1. Windows Explorer を使用して、SQL Anywhere がインストールされているインストール・フォルダを参照 し(以下はデフォルトのインストール・パス)、以下のフォルダを探します。

C:**¥**Program Files**¥**SQL Anywhere 12**¥**Assembly**¥**v2

2. SASetupAspNet.exe をダブルクリックし、[ASP.NET Security Schema Setup] ウィザードを実行します。また は、コマンド・ライン・プロンプトからでも SASetupAspNet.exe を実行できます。コマンド・ラインを使用 して SASetupAspNet.exe にアクセスする場合は、疑問符 (-?)引数を使用すると、データベースの構成に関する詳 細なヘルプが表示されます。

	ASP.NET Security Schema Setup
where	This wizard will guide you through installing and uninstalling ASP.NET security features on a SQL Anywhere server.
iAny	© <u>E</u> nglish C Japanese O <u>F</u> rench O <u>C</u> hinese O <u>G</u> erman
	Press Next to continue.
	< <u>B</u> ack. <u>N</u> ext> <u>Cancel</u>

- 3. 希望の言語を選択し、[Next] をクリックします。
- 4. [Connection String] に "dsn=SQL Anywhere 12 Demo" と入力し、[Next] をクリックします。ここで、データベース接続のテストも行えます。

Connect to the Database Specify a connection string to connect to the desired database				
Connection String:	dsn=SQL Anywhere 12 Demo			
	Test Connection			
You have the option to already owns, the data provider functionality. T	specify the SQL Anywhere database user that will own, or base resources (tables and stored procedures) that enable the his user is 'DBA' by default.			
Specify Custom Use	er DBA			
	< <u>B</u> ack <u>N</u> ext> <u>C</u> ancel			

5. 必要な機能を追加し(この場合はすべて)、[Next] をクリックします。すでにインストール済みの機能を削除することもできます。

SQL Anywhere	を使った	ASP.NE	Г Web	ページの作成	Ì
	バー	ージョン	1.1 -	2009年3月	

4

		<u> </u>
Choose Features Select features to add or remove		
	Add	Remove
Membership		
Roles		
Profile		
Web Parts Personalization		
Web Events (Health Monitoring)		
Checking "Add" next to features that are already installed v those features. By default, the data in tables associated will preserved whenever possible. You have the option to turn clean reinstall.	vill cause a re th those featu data preserva	einstall of ures will be ation off for a
🔽 Preserve Data		
< <u>B</u> ack	<u>N</u> ext>	<u>C</u> ancel

6. インストール内容を確認し、[Finish] をクリックします。

			
Finished Schema setup completed			
Details			
Engine: demo12 Database: demo User: DBA			<u> </u>
Installing feature: Common			
Installing feature: Membership			
Installing feature: Roles			
Installing feature: Profiles			
Installing feature: Personalization			
Installing feature: WebEvents			_
	< <u>B</u> ack	<u>F</u> inish	<u>C</u> ancel

SQL Anywhere ASP.NET プロバイダのスキーマの表示

前の項のウィザードにより、必要なテーブルおよびストアド・プロシージャがデモ・データベースに追加されました。 いずれも "aspnet_" というプレフィックスが付いています。この項では、変更する点を示しながら、ログインに関す る機密情報とアプリケーションのセキュリティ情報が、SQL Anywhere でどのように格納されるかについて説明しま す。

手順

 Sybase Central を開き、デモ・データベースに接続します。[Folders] ビューで [Tables] を展開し、ウィザードに よって作成されたテーブルを表示します。以下の図のように、"aspnet_"というプレフィックスが付いた新し いテーブルが、デモ・データベースに追加されています。これらのテーブルには、Web サイトで使用される セキュリティ・データが格納されます。



 テーブル aspnet_Membership をクリックすると、テーブルのカラムが表示されます。このテーブルには、 ユーザのメンバシップ情報が格納されます。たとえば、ユーザ ID (プライマリ・キー)、パスワード、セキ ュリティのための質問、最終ログイン日付など、Web サイトの認証に必要なすべての情報が格納されます。 デフォルトでは、データベースに格納された情報が確実に保護されるように、パスワード形式がハッシュさ れます。

	nsi Cor	nstraints Referencin	g Constra	iints Inde>	kes Text Indexes Trigg	ers Depen	ident View	s Data		
	PKey	Name	ID 🛦	Object ID	Data Type	Size	Scale	Cmp.	Nulls	Unique
1		ApplicationId	1	3523	uniqueidentifier					
2	•	UserId	2	3524	uniqueidentifier					
З		Password	3	3525	nvarchar	128				
4		PasswordFormat	4	3526	integer					
5		PasswordSalt	5	3527	nvarchar	128				
6		MobilePIN	6	3528	nvarchar	16			~	
7		Email	7	3529	nvarchar	256			~	
8		LoweredEmail	8	3530	nvarchar	256			~	
9		PasswordQuestion	9	3531	nvarchar	256			~	
10		PasswordAnswer	10	3532	nvarchar	128			~	
11		IsApproved	11	3533	bit			Γ		
12		IsLockedOut	12	3534	bit			Γ		
13		CreateDate	13	3535	datetime			Γ		
14		LastLoginDate	14	3536	datetime			Γ		
15		LastPasswordCha	15	3537	datetime			Γ		
16		LastLockoutDate	16	3538	datetime			Γ		
17		FailedPasswordAt	17	3539	integer			Г		
18		FailedPasswordAt	18	3540	datetime			Г		
19		FailedPasswordA	19	3541	integer			Γ		
20		FailedPasswordA	20	3542	datetime			Γ		
21		Comment	21	3543	long nyarchar			П	•	

3. [Constraints] タブをクリックします。ウィザードにより、カラムに対して適切なキー制約が設定され ていることがわかります。



[Folders] ビューで [Procedures and Functions] を展開します。ウィザードにより、各プロバイダで使用可能な操作のプロシージャが追加されています。aspnet_Membership_CreateUser プロシージャをクリックすると、SQL コードが表示されます。これは、データベースに新しいユーザ・レコードが挿入されたときに Membership Provider が使用するプロシージャです。

SQL Anywhere を使った ASP.NET Web ページの作成 Version 1.1 - March 2009

Sybase Central						
Eile Edit View Iools Connections Mode Help						
\$\$ ⇒ € 💟 🖻 ₩ 🚇 • 및 • ₽ • (← ► ♥ ► ₩ ₽ ▼ ₽ ▼ ₽ ▼ ₽ ▼ ■ × ▷ ○ ₩ □ ♥ ■					
Context: Sybase Central/SQL Anywhere 12/demo12/demo - DBA	/Procedures & Functions/aspnet_Membership_CreateUser (DBA)					
💥 Tools 👻 🎉 🛃 🖺 🏷 🏷 🕼 🕼 🗐						
Folders × = aspnet_Membership_CreateUser (DBA)						
Procedures & Functions	SQL Parameters					
aspnet_CheckSchemaVersion (DBA)	ALTER PROCEDURE "DBA"."aspnet_Membership_CreateUser"(
aspnet_Membership_ChangePasswordQu	IN @ApplicationName nvarchar(256),					
	IN @UserName nvarchar(256),					
aspnet_Membership_FindUsers (DBA)	IN @Password nvarchar(128),					
aspnet_Membership_GetAllUsers (DBA)	IN @PasswordSalt nvarchar(128),					
aspnet_Membership_GetNumberOfUsers	IN @Email nvarchar(256),					
aspnet_Membership_GetPassword (DBA)	IN @PasswordQuestion nvarchar(256),					
aspnet_Membership_GetPasswordWithFo	IN @PasswordAnswer nvarchar(128),					
	IN @IsApproved bit,					
aspnet_Membership_GetUserByName (DI	IN @CurrentTimeUtc datetime,					
aspnet_Membership_GetUserByUserId ([IN @UniqueEmail bit DEFAULT 0,					
aspnet_Membership_ResetPassword (DB	IN @PasswordFormat int DEFAULT 0,					
	OUT @UserId uniqueidentifier,					
aspnet_Membership_UnlockUser (DBA)	OUT @ReturnValue int)					
aspnet_Membership_UpdateUser (DBA)						
aspnet_Membership_UpdateUserInfo (Dt	ON EXCEPTION RESUME					
aspnet_Paths_CreatePath (DBA)	BEGIN					
aspnet_Personalization_DeleteAllState ([DECLARE @ApplicationId uniqueidentifier;					
aspnet_Personalization_FindState (DBA)	DECLARE @NewUserId uniqueidentifier;					
aspnet_Personalization_GetCountOfStat	DECLARE @NewUserCreated bit;					
aspnet_Personalization_GetPageSettings						
aspnet_Personalization_ResetPageSettir	CALL "DBA".aspnet_Applications_CreateApplication(@ApplicationN					

5. [Indexes] を展開します。パフォーマンスを向上するために、追加されたテーブル用のインデックス も自動的に作成されています。

Sybase Central						
File Frit View Tools Connections Mode Help						
		s:P				
Context: 👩 Sybase Central/SQL Anywhere 12/demo12/demo - DBA/I	ndexes					
% Tools マ 📐 蒹 🗗 🎁						
Folders ×	Indexes					
🗗 🗊 Indexes	indexes l					
- Baspnet_Applications - aspnet_Applications (DBA)		Object Name	Ohiesh Owness	Te days Trans	Unince	
🔤 aspnet_Membership - aspnet_Membership (DBA)	vame 🔺	Object Name	Object Owner	Index Type	Unique	
- 🔂 aspnet_Paths - aspnet_Paths (DBA)	aspnet_Applications	aspnet_Applications	DBA	Primary key index	Yes	
- 🚮 aspnet_PersonalizationAllUsers - aspnet_Persona	🚽 🔀 aspnet_PersonalizationAllUsers - aspnet_Persona 🛛 😼 aspnet_Membership aspnet_Membership DBA Primary key index Yes					
	🚽 🙀 aspnet_PersonalizationPerUser - aspnet_Persona 🛛 😼 aspnet_Paths aspnet_Paths DBA Primary key index Yes					
	🕂 🙀 aspnet_Profile - aspnet_Profile (DBA) 🛛 📲 aspnet_PersonalizationAllUsers aspnet_PersonalizationAll DBA Primary key index Yes					
	- 🔀 aspnet Roles - aspnet Roles (DBA) 🛛 🛛 🛃 aspnet_PersonalizationPerUser aspnet_PersonalizationPer DBA Primary key index Yes					
🗕 🙀 aspnet_SchemaVersions - aspnet_SchemaVersion 🛛 🙀 aspnet_Profile 🛛 aspnet_Profile 🛛 DBA Primary key index Yes				Yes		
aspnet_Noles DBA Primary key index Yes				Yes		
aspnet UsersInRoles - aspnet UsersInRoles (DB/	aspnet_SchemaVersions	aspnet_SchemaVersions	DBA	Primary key index	Yes	
aspnet WebEvent Events - aspnet WebEvent E	aspnet_Users	aspnet_Users	DBA	Primary key index	Yes	
ContactsKey - Contacts (GROUPO)	aspnet_UsersInRoles	aspnet_UsersInRoles	DBA	Primary key index	Yes	
Customers (GROUPO)	aspnet WebEvent Events	aspnet WebEvent Events	DBA	Primary key index	Yes	
		Contacts	GROUPO	Primary key index	Yes	
Employees/ey - Employees (GROUPO)	CustomersKev	Customers	GROUPO	Primary key index	Yes	
Einprovedskey - Einproveds (GROUPO)	DepartmentsKey	Departments	GROUPO	Primary key index	Yes	
FinancialDataKey, FinancialData (CROLIDO)		Employees	GROUPO	Primary key index	Yes	
Financial/Jackey - Mancial/Jaca (GROUPO) Financial/Jacakey - Mancial/Jaca (GROUPO) Financial/Jacakey - Mancial/Jacakey - Manc		EinancialCodes	GROUPO	Primary key index	Ves	
FK_Applicationud - aspnet_Membership (DBA)	FinancialDataKey	EinancialData	GROUPO	Primary key index	Ves	
rk_Applicationud - asphet_Paths (DBA)	EK ApplicationId	aconet Memberchin	DBA	Eoreign key index	No	
FK_Application1d - aspnet_Roles (DBA)	EK ApplicationId	asprict_nondership	DBA	Foreign Key Index	No	
E EV Applicational association associati association association association associati						

ASP.NET Web サイトの作成とエンティティ・データ・モデルの追加

Visual Studio で定義された任意の SQL Anywhere データベース・プロファイルを使用して、新しいエンティティ・デ ータ・モデル (EDM) を作成できます。以下の手順に従って、ASP.NET Web サイトを作成し、このサイトに SQL Anywhere デモ・データベースを EDM として追加します。

- 1. Visual Studio 2010 を起動します。
- 2. [File] > [New] > [Web Site] を選択します。
- 3. [New Web Site] ダイアログで [ASP.NET Web Site] を選択します。
- 4. 自分で選んだ既知の場所に、この Web サイトを 'Sample_asp.net' という名前で保存します。

5. Solution Explorer で Web サイト・プロジェクトを右クリックし、ポップアップ・メニューから [Add New Item] > [ADO.NET Entity Data Model] をクリックします。

- 6. [Name] フィールドに SADemo.edmx と入力します。[Add] をクリックします。
- 7. [Yes] をクリックすると、ADO.NET Entity Model がフォルダ App_Code に追加されます。

dd New Item - WindowsForms#	upplication 1		2
Installed Templates	Sort by: Default 🔳 💷 🔠	1	Search Installed Templates
Visual C# Items Code Date	Class	Visual C# Items	Type: Visual C# Items A project item for creating an ADO.NET Entity
General Web	C Interface	Visual C# Items	Data Model.
Windows Forms WPF	Windows Form	Visual ⊂# Items	
Reporting Worldlow	User Control	Visual C# Items	
Vnline Templates	Component Class	Visual C# Items	
	User Control (WPF)	Visual C# Items	
	About Box	Visual C# Items	
	ADO.NET Entity Data Model	Visual C# Items	
	ADO.NET EntityObject Generator	Visual C# Items	
	ADO.NET Self-Tracking Entity Generator	Visual C# Items	
	Application Configuration File	Visual C# Items	
	Application Manifest File	Visual C# Items	
	Assembly Information File	Visual C# Items	
	Bitmap File	Visual C# Items	
Salleme a	Class Diagram	Visual C# Items 💌	
parties and participations			

8. [Entity Data Model] ウィザードが表示されます。[Generate from database] を選択し、[Next] をクリックします。

Entity Data Model Wizard	? ×
What should the model contain?	
Generate from Empty model database	
Generates the model from a database. Classes are generated from the model when the project i compiled. This wizard also lets you specify the database connection and database objects to incl	s ude
in the model.	
< <u>Previous</u> <u>Next</u> <u>Finish</u> Canc	el

9. [New Connection] をクリックします。[Data Source] の横の [Change] ボタンをクリックします。[Data source] リス トで [SQL Anywhere] を選択し、[OK] をクリックします。

[Data source] リストに [SQL Anywhere] が表示されていない場合、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。統合コ ンポーネントをインストールするには、以下の手順を実行します。
1. Visual Studio 2010 を閉じます。
2. コマンド・プロンプトを開き、以下のディレクトリに移動します。 C:¥Program Files¥SQL Anywhere 12¥Assembly¥v2
3. 次に、以下のコマンドを実行します: SetupVSPackage.exe -i

Change Data Source	<u>? ×</u>
Data source: Microsoft SQL Server Microsoft SQL Server Compact 3.5 Microsoft SQL Server Database File SQL Anywhere <other> Data grovider: .NET Framework Data Provider for SK</other>	Description Use this selection to connect to SQL Anywhere 12 using the .NET Framework Data Provider for SQL Anywhere 12
Always use this selection	OK Cancel

10. [ODBC Data Source] の名前をクリックし、[SQL Anywhere 12 Demo] を選択します。[OK] をクリックします。

SQL Anywhere を使った ASP.NET Web ページの作成 バージョン 1.1 - 2009 年 3 月

10

dd Connection		<u>? ×</u>
Enter information to conne "Change" to choose a diffe	ct to the selected data source rent data source and/or provid	or click der.
Data <u>s</u> ource:		
SQL Anywhere (SQL Anyw	here 12)	hange
Data source		
C None		
ODBC <u>d</u> ata source:	SQL Anywhere 12 Demo	-
C Host name:		
Ser <u>v</u> er name:		
Data <u>b</u> ase name:		
-Login information		
● Supply user ID and pa	assword	
<u>U</u> ser ID:		
Password:		
O Use integrated login		
	[Ad <u>v</u> anced
Test Connection	ок	Cancel

11. エンティティ接続文字列の名前を "SAEntities" に変更し、[Next] をクリックします。

ntity Data Model Wizard	? ×
Choose Your Data Connection	
Which data connection should your application use to connect to the database?	
SQL Anywhere.demo12 New Connection	
This connection string appears to contain sensitive data (for example, a password) that is requir connect to the database. Storing sensitive data in the connection string can be a security risk. E you want to include this sensitive data in the connection string?	ed to o
$ \mathbb{C}$ No, gxclude sensitive data from the connection string. I will set it in my application code.	
${f C}$ Yes, include the sensitive data in the connection string,	
Entity connection string:	
Data.SQLAnywhere;provider connection string='DataSourceName="SQL Anywhere 12 Demo"	
Save entity connection settings in App.Config as:	
SAEntities	
< <u>P</u> revious <u>N</u> ext > ⊟nish Cano	el



 モデルに含めるデータベース・オブジェクトとしてすべてのデータベース・オブジェクトを選択し、[Finish] をクリックします。実際には、EDM はアプリケーションに必要なオブジェクトだけを含めます。このチュー トリアルでは、簡単にするために、すべてのオブジェクトを追加します。

Entity Data Model Wizard	? ×
Choose Your Database Objects	
Which database objects do you want to include in your model?	
W	
J ✓ Pluralize or gingularize generated object names	
Iinclude foreign key columns in the model Model Namespace:	
Model	
< Previous Next > Einish	Cancel

13. SADemo.edmx ファイルを開くと、Entity Designer にモデルが視覚的に表示されます。

SalesOrders エンティティを右クリックし、[Table Mapping] を選択すると、マッピングの詳細情報が表示 されます。以下の図は、データベース・スキーマに対してエンティティが適切にマッピングされている ことを示しています。

Mappir	ng Details - SalesOrders								
I	Column	Operator	Value / Property						
	⊡ Tables								
	🚊 🧕 Maps to SalesOrders								
	🗄 📲 Column Mappings								
	🧾 ID (int)	↔	🚰 ID (Int32)						
	🧾 CustomerID (int)	↔	*						
	🔲 OrderDate (datetime)	↔	🚰 OrderDate (DateTime)						
	🧾 FinancialCode (char)	↔	*						
	🔳 Region (char)	↔	🚰 Region (String)						
	🔤 SalesRepresentative (int)	↔	*						

14. Entity Designer を使用して、ASP.NET プロバイダによってデモ・データベースに追加されたテーブルを表示す ることもできます。Entity Designer で何もない部分を右クリックし、[Model Browser] をクリックすると、Model ネ ームスペース内のすべてのエンティティ型の概要が表示されます。



データを表示するための Web コントロールの作成

ASP.NET プロバイダと Web サイトで使用する Entity Model の設定が完了したので、デフォルト・ページに GridView コントロールを追加し、EntityDataSource にバインドします。このページは、Web サイトのパブリックな領域になります。このページでは、すべてのユーザが認証の必要なく情報を参照できます。

手順

- 1. Default.aspx を開いてデザイン・ビューに切り替えます。ツールボックスから GridView コントロールをドラ ッグし、デフォルトのボディ・コンテンツと置き換えます。
- 2. [GridView Tasks] メニューで、[Choose Data Source] ドロップダウン・リストを表示し、[<New data source...>] を選択します。

Home Profile MainContent (Custom) Welcome to Our Company Directory asp:GridView#GridView1 GridView Tasks Abc abc		SP.N	ET APP	LICATION	
Golumn0 Column1 Column2 abc abc abc	Home	Profile			
GridView#GridView	MainConter	nt (Custom)			
Column0 Column1 Column2 GridView Tasks abc abc abc Auto Format abc abc abc Edit Columns (None) abc abc abc Edit Columns (None) abc abc abc Edit Columns Edit Templates	Welcon asp:GridViev	ne to O w#GridView1	ur Compa	ny Directory	
abc abc abc	Column0	Column1	Column2	Grid¥iew Tasks	
abc abc abc abc abc abc abc abc abc abc	abc	abc	abc	Auto Format	(Nana)
abc abc abc abc abc abc dt Templates	abc	abc	abc	Edit Columns	(None)
abc abc Edit Templates	abc	abc	abc	Add New Column	<new data="" source=""></new>
	abc	abc	abc	Edit Templates	
		abc	abc		

3. [Entity] を選択し、[OK] をクリックします。

Data Source Co	onfiguration Wi	zard					? X
Cł	noose a Data	Source Ty	pe				
<u>W</u> here will th	ne application (get data fron	n?				
Access Database	International Sector	Entity	LINQ	object	Site Map	USAN KARAN	
Connect to an		Eromowork Ma					_
	ADO MET Endoy	Framework Mc					
Specify an <u>I</u> D f	or the data sour	te:					
EntityDataSou	rce1						
					ОК	Cano	el

4. [Configure Data Source] ウィザードが表示されます。[Named Connection] ドロップダウン・リストから [SAEntities] を選択し、[Next] をクリックします。

Configure Data Source - EntityDataSource1	? ×
Configure ObjectContext	
ConnectionString:	
Named Connection	
SAEntities	•
C Connection String	
DefaultContainerName:	
SAEntities	•
< <u>Pr</u> evious <u>Next</u> > <u>Finish</u>	<u>C</u> ancel

 [EntitySetName] ドロップダウン・リストから [Employees] を選択します。誰でも参照できるパブリック 情報として、[EmployeeID]、[ManagerID]、[Surname]、[GivenName]、[DepartmentID]、[State]、 [Country]、[Phone] を選択します。[Finish] をクリックします。

> SQL Anywhere を使った ASP.NET Web ページの作成 バージョン 1.1 - 2009 年 3 月

15

Configure Data Source - EntityData	Source1	<u>?</u> ×
Configure Data Sele	ction	
EntitySetName:		
Employees		-
Entity <u>T</u> ypeFilter:		
(None)		_
Select:		_
- Select All (Entity Value)	City	Salary
🖌 EmployeeID	🔽 State	🔄 🗌 StartD
✓ ManagerID	Country	🗌 Termir
Surname	PostalCode	BirthD
		🔄 Benefi
	Status	🔄 Benefi
🗖 Enable automatic inserts		
🗖 Enable automatic ypdates		
Enable automatic deletes		
	< Previous Next > Einish	

 [F5] を押してアプリケーションを実行/デバッグします。必要に応じて、Web.config ファイル内でデバッ グを有効にします(有効にするかどうかを尋ねるダイアログが表示される場合があります)。デフォルト・ ページに、デモ・データベースの Employee テーブルから選択したカラムが表示されます。この情報は、 このページにアクセスできるユーザであれば誰でも参照できます。ページを閉じてアプリケーションを終 了します。

2	Home P	age - Mozilla Firefo;	6						_ 🗆 ×
Đ	le <u>E</u> dit	<u>V</u> iew Hi <u>s</u> tory <u>B</u> o	okmarks <u>T</u> ool	s <u>H</u> elp					
(< >	- C 🗙 🎸	http:	//localhost:2359,	/Sample_asp.net/def	ault.asp	x	🖒 🔹 🛃 - Google	P
) Home	Page		+					-
									<u> </u>
	Mv	ASP NET		ΑΤΙΟΝ					
									_
	Home	About							
Г									
	WELC	OME TO OUR O	COMPANY	DIRECTOR	Y				
	Emplo	yeeID ManagerID	Surname	GivenName	DepartmentID	State	Country	Phone	
	102	501	Whitney	Fran	100	NY	USA	6175553985	
	105	501	Cobb	Matthew	100	UT	USA	6175553840	
	129	902	Chin	Philip	200	GA	USA	4045552341	
	148	1293	Jordan	Julie	300	AZ	USA	6175557835	
	160	501	Breault	Robert	100	PA	USA	6175553099	
	184	1576	Espinoza	Melissa	400	ME	USA	5085552319	
	191	703	Bertrand	Jeannette	500	AZ	USA	5085558138	
	195	902	Dill	Marc	200	PA	USA	6175552144	
	207	1576	Francis	Jane	400	FL	USA	5085559022	
	243	501	Shishov	Natasha	100	UT	USA	6175552755	
	247	501	Driscoll	Kurt	100	UT	USA	6175551234	
4									•

16

SQL Anywhere ASP.NET Data Provider の構成

次に、セキュア・ページを Web サイトに追加し、適切なセキュリティ設定を有効にします。また、SQL Anywhere ASP.NET プロバイダを使用するように、アプリケーションを構成する必要があります。以下の手順 は、SQL Anywhere ASP.NET プロバイダの登録方法と、ASP.NET 管理ツールを使ったユーザ情報の管理方法を 示しています。

 Solution Explorer でプロジェクトを右クリックし、[Add Reference] を選択します。[.NET] タブを選択し、 iAnywhere.Web.Security アセンブリへの参照を Web サイトに追加するために、リストから [iAnywhere.Web.Security] を選択します。

ltered to: .NET Framework 4			
Component Name 🔺	Version	Runtime	Pal 🔺
iAnywhere.QAnywhere.StandAloneClient	12.0.0.2483	v2.0.50727	C:N
iAnywhere.QAnywhere.WS	12.0.0.2483	v2.0.50727	- C; \
iAnywhere.VSIntegration.SQLAnywhere	12.0.0.24832	v2.0.50727	Cil
iAnywhere.Web.Security	12.0.0.2483	v2.0.50727	Ci
InfoPath.DomainControl	11.0.0.0	v1.1.4322	C()
InfoPath.DomainControl	11.0.0.0	v1.1.4322	C()
InfoPath.DomainControl	11.0.0.0	v2.0.50727	C()
ISymWrapper	4.0.0.0	v4.0.30319	C()
Microsoft.Build	4.0.0.0	v4.0.30319	C()
Microsoft.Build.Conversion.v4.0	4.0.0.0	v4.0.30319	C()
Microsoft.Build.CPPTasks.Common	4.0.0.0	v4.0.30319	C()
Microsoft.Build.CPPTasks.Itanium	4.0.0.0	v4.0.30319	Ců
Microsoft.Build.CPPTasks.Win32	4.0.0.0	v4.0.30319	C()
Microsoft.Build.CPPTasks.x64	4.0.0.0	v4.0.30319	Cù 🚽
	4000		للناري

2. web.config ファイルを開き、<connectionStrings>要素に接続文字列を追加します(ハイライト部分が変更箇所)。

```
<connectionStrings>
<!--接続文字列の登録-->
<add name="MyConnectionString"
connectionString="dsn=SQL Anywhere 12 Demo"
providerName="iAnywhere.Data.SQLAnywhere" />
<add name="SAEntities"
```

connectionString="metadata=res://*/App_Code.SADemo.csdl|res://*/App_Code.SADe
mo.ssdl|res://*/App_Code.SADemo.msl;provider=iAnywhere.Data.SQLAnywhere;provi der
connection string='UserID=dba;Password=sql;DataSourceName="SQL Anywhere 12
Demo"'"

```
providerName="System.Data.EntityClient" />
</connectionStrings>
```

3. プロバイダごとにエントリを追加し、<machineKey validation="SHA1">を <system.web>要素に追加しま す。SQL Anywhere ASP.NET プロバイダの名前をアプリケーションの "defaultProvider" 属性に追加し ます (ハイライト部分が変更箇所)。

```
<system.web>
  <machineKey validation="SHA1" />
   <compilation debug="true" strict="false" explicit="true" targetFramework="4.0">
      <assemblies>
         <add assembly="System.Security, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A" />
         <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
         <add assembly="System.Data.Entity.Design, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=B77A5C561934E089" />
      </assemblies>
      <buildProviders>
         <add extension=".edmx"
         type="System.Data.Entity.Design.AspNet.EntityDesignerBuildProvider" />
      </buildProviders>
   </compilation>
   <authentication mode="Forms">
      <forms loginUrl="~/Account/Login.aspx" timeout="2880" />
   </authentication>
   <membership defaultProvider="SAMembershipProvider">
      <providers>
         <clear />
         <add name="SAMembershipProvider"
              type="iAnywhere.Web.Security.SAMembershipProvider"
              connectionStringName="MyConnectionString"
              enablePasswordRetrieval="false" enablePasswordReset="true"
              requiresQuestionAndAnswer="false" requiresUniqueEmail="false"
              maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6"
              minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10"
              applicationName="/"
              passwordFormat="hashed" />
      </providers>
   </membership>
   <profile defaultProvider="SAProfileProvider">
      <providers>
         <clear />
         <add name="SAProfileProvider"</pre>
              type="iAnywhere.Web.Security.SAProfileProvider"
              connectionStringName="MyConnectionString" applicationName="/"
                 commandTimeout="30" />
      </providers>
   </profile>
   <roleManager enabled="true" defaultProvider="SARoleProvider">
      <providers>
         <clear />
         <add connectionStringName="MyConnectionString" applicationName="/"
```

	commandTi	neout="30" name="SARoleProvider"
	type="iAng	where.Web.Security.SARoleProvider" />
1</td <td>oleManager></td> <td></td>	oleManager>	
<td>em.web></td> <td></td>	em.web>	

SQL Anywhere ASP.NET プロバイダの構成方法の詳細については、以下の API のオンライン・マ ニュアルを参照してください。 http://dcx.sybase.com/index.html#1201/ja/dbprogramming/aspdotnet-providers.html

これで、Web Site Administration Tool を使用して新しいユーザを作成できます。このツールを使用するには、変更を保存してから、Visual Studio 2010 で [Website] メニューの [ASP.NET Configuration] を クリックします。[Provider Configuration] > [Select a different provider for each feature (advanced)] をクリックすると、SAMembershipProvider と SARoleProvider が使用されていることがわかります。

8	ASP.Net Web App	plication Ad	ministration - Mozi	lla Firefox				- O ×
Eile	<u>E</u> dit ⊻iew	Hi <u>s</u> tory (<u>B</u> ookmarks <u>T</u> ools	Help				
) > • C	🗙 🏠	http://loca	lhost:53637/asp.r	netwebadminfiles/providers/	'ManageProvi 🏠 🔹	Google	🔎 🔒
	ASP.Net Web	Applicatio	n Adminis 🔶					-
A	SP.net	Web Site	e Administratior	ı Tool			<u>How do I use t</u>	his tool?
ſ	Home) (Security	l l	Application	l l	Provider	
	Use this pag	e to selec o Provide	t a provider for	each feature.				
	• StiMemberg	hipProvider	e T					
	- Senembersi	hiperovider	L,	4				
	Role Provid	er						
	SARoleProvi	ider						
								Back
								_

5. [Security] タブには、ロールの有効化やユーザごとのアクセス・ルールの作成など、Web サ イトの管理タスクを実行するためのさまざまなツールが表示されます。

🕙 ASP.Net Web Application Administration - M	lozilla Firefox				<u>- 0 ×</u>
<u>File Edit View History Bookmarks To</u>	ols <u>H</u> elp				
🔇 🔊 - C 🗙 🏠 🗋 http://	localhost:53637/asp.netwe	badminfiles/security/secur	ity.aspx 🏠 🔹	Soogle Google	P 🔒
ASP.Net Web Application Adminis	÷				-
ASP. Web Site Administrat	tion Tool			How do I use this	<u>s tool?</u>
Home Securi	ity	Application	l l	Provider	
bu can use the Web Site Adminis up users and passwords (authenti controlling access to parts of you By default, user information is sto site. If you want to store user info provider. Use the security Setup Wizard to Click the links in the table to mana	cation), create roles (r application). red in a Microsoft SQL ormation in a different <u>configure security step</u> age the settings for you	Server Express datab database, use the Pr <u>b by step.</u> ur application.	create permiss ase in the Data ovider tab to se	a folder of your We	eb
Users	Roles		Access Rules		
Existing users: 0 <u>Create user</u> <u>Manage users</u> <u>Select authentication type</u>	Existing roles: 0 Disable Roles Create or Manag	<u>e roles</u>	<u>Create access</u> <u>Manage acces</u>	<u>: rules</u> s rule <u>s</u>	

6. [Security] > [Create user] をクリックすると、Web フォームを入力して新しいユーザ・アカウントを作成できます。

🕹 ASP.Net Web Application Administr	ation - Mozilla Firefox			<u> </u>
Eile Edit <u>V</u> iew History Bookma	rks <u>T</u> ools <u>H</u> elp			
🔇 🔊 • C 🗙 🏠 🗋	http://localhost:53637/a	sp.netwebadminfiles/security/users/addUser	☆ • <mark>왕</mark> • Google	P S
ASP.Net Web Application Adm	inis 🔶			-
				-
ASP.net Web Site Adm	inistration Tool			e this tool? 🕐
Home	Security	Application	Provider	
Create User Sign Up for Your	New Account	Roles No roles have been defined.	l	
User Name: Password: Confirm Password: E-mail:				
☑ Active User	_ Create User			Back

7. [ASP.NET Web Application Administration] ページを閉じます。

認証済みユーザに対する個人情報の表示

これで、ログインしている従業員のみ利用可能な、従業員のプロファイル情報が表示される2番目のページを作 成できます。簡単にするために、Employee テーブルにすでに情報が存在するユーザを登録します(ユーザ名として 名を使用)。組み込みの About.aspx ファイルの名前を変更し、個人情報を表示する従業員プロファイル・ページに なるようカスタマイズします。Visual Studio には、強化された ASP.NET Web サイト・テンプレートも用意され ています。このテンプレートには、基本的なメンバシップ機能がすでに構成済みの組み込みアカウント・ フォルダが含まれています。以下の手順では、これらのテンプレートを利用して、サンプル・アプリケー ションを設定します。

1. Solution Explorer で About.aspx ファイルを右クリックし、[Rename] をクリックします。ファイル名を Profile.aspx に変更します。対応するコード・ファイルも自動的に名前が変更されます。

 メニュー項目名がファイル名と一致するように、マスタ・ページのメニュー・バーを編集する必要があ ります。site.master ファイルを開いてデザイン・ビューに切り替えます。ナビゲーション・メニューを探 し、その横の [>]ボタンをクリックします。[Edit Menu Items] を選択し、Menu Item Editor で2番目のメニ ュー項目の URL を "~/Profile.aspx" に置き換え、Text プロパティを [Profile] に変更します。[OK] をクリ ックします。

Menu Item Editor	? :
Items:	Properties:
😫 🏗 🗙 🖢 🔶 🍜 🏦	
Home	Enabled True
····· Profile	ImageUrl
	NavigateUrl ~/Profile.aspx
	PopOutImageU
	Selectable True
	Selected False
	SeparatorImage
	Target
	Text Profile
	ToolTip
	Yalue The value of the menu item.
	OK Cancel

ソース・ビューに切り替えると、ナビゲーション・メニューのマークアップが以下のように表示されるはずです。

```
<asp:Menu ID="NavigationMenu"
runat="server"
CssClass="menu"
EnableViewState="false"
IncludeStyleBlock="false"
Orientation="Horizontal">
<Items>
<asp:MenuItem NavigateUrl="~/Default.aspx" Text="Home"/>
<asp:MenuItem NavigateUrl="~/Profile.aspx" Text="Profile"/>
</Items>
</asp:Menu>
```

 ログイン成功後に、このプロファイル・ページにユーザが自動的にリダイレクトされるように、デフォルト のログイン・テンプレートを編集します。Account/Login.aspx を開き、デザイン・ビューで Login コントロー ルを右クリックし、[Properties] をクリックします。DestinationPageURL プロパティに対して Profile.aspx を設 定します。

Properties	* ⊡ ×
LoginUser System.Web.UI.W	ebControls.Login 🔹
∄2↓ 🔲 🖋 🖂	
Accessibility	·
AccessKey	
TabIndex	0
Appearance	
FailureText	Your login attempt was not su
Font	
UserName	
Behavior	
ClientIDMode	Inherit
DestinationPageUrl	~/Profile.aspx
Enabled	True
EnableTheming	True
EnableViewState	False
FailureAction	Refresh
RememberMeSet	False
SkinID	
ToolTip	
ViewStateMode	Inherit —
Visible	True
VisibleWhenLoggedIn	True
🗉 Data	
(Expressions)	-
DestinationPageLirl	
The LIRL that the user is directed	d to upop successful login
	a to apont according in
EnableWiewState FailureAction RememberMeSet SkinID TooTrp ViewStateMode Visible VisibleWhenLoggedIn Data (Expressions) DestinationPageUrl The URL that the user is directe	False Refresh False Inherit True True to upon successful login.

4. Profile.aspx を開き、MainContent プレースホルダのデフォルト行を削除します。次に、Label コントロール、 LoginName コントロール、および DetailsView コントロールを MainContent プレースホルダにドラッグ・ア ンド・ドロップします。



SQL Anywhere を使った ASP.NET Web ページの作成 バージョン 1.1 - 2009 年 3 月

- 5. [Details View Tasks] メニューで、[Choose Data Source] ドロップダウン・リストを表示し、[<New data source...>] を選択します。パブリック・ページの場合と同様に、新しい Entity DataSource を追加します。
- 6. [Entity]を選択し、[OK]をクリックします。
- 7. [Named Connection] ドロップダウン・リストから [SAEntities] を選択し、[Next] をクリックします。
- 8. [EntitySetName] ドロップダウン・リストから [Employees] を選択します。今回は、テーブルのすべてのカ ラムを選択します。[Finish] をクリックします。

Configure Data Source - EntityDat	aSource1	? ×
Configure Data Sel	ection	
EntitySetName:		
Employees		
EntityTypeFilter:		
(None)		•
Select:		
Select All (Entity Value)	City	Salary
EmployeeID	State	StartD
ManagerID	Country	🗌 Termir
Surname	PostalCode	🗌 BirthD
GivenName	Phone	🗌 Benefi
DepartmentID	Status	🗌 Benefi
Street	SocialSecurityNumber	🗌 Benefi
•		F
Enable automatic inserts		
Imable automatic updates		
Enable automatic deletes		
	< Previous Next > Einish	<u>Cancel</u>

9. DetailsView コントロールの下に新しい EntityDataSource が表示されます。これを右クリックし、[Properties] を選択します。

- [Properties] メニューの横の [Where] ボタンをクリックして式エディタを開きます。[Add Parameter] を クリックし、[name] フィールドに "emp_name" と入力します。 [Parameter Source] ドロップダウ ン・リストから [Session] を選択し、[Session] フィールドに "uid" と入力します。[Where Expression] フ ィールドに "it.GivenName=@emp_name" と入力します。
- 11. [Show advanced properties] をクリックし、[DbType] ドロップダウン・リストから [String] を選択しま す。[OK] をクリックして式エディタを閉じます。

E	Rpression Edito	r generate the Where expressi	on based on	the provided parameters.	<u>?</u> ×
	Where Expressio	in:			
	it.GivenName =	@emp_name			×
	Param <u>e</u> ters:			Parameter <u>s</u> ource:	
	Name	Value	\$	Session	
	emp_name	Session("uid")		Drenevkies	
				Properties:	-
			\mathbf{x}	ConvertEmptyStringToNull DbType	Irue String
			_	DefaultValue	string <u>·</u>
				Direction	Input
				Name	emp_name
				SessionField	uid
				Size	0
				Туре	Empty
				DbType The database type of the param the Type property will be used in	eter. If this property is set to DbType.Object, stead.
		Add Parameter	1		
				Hide advanced properties	<u>QK</u> <u>Cancel</u>

24

```
12. Profile.aspx ページの Page_Load イベントに以下のコードを挿入します。
   [C#]
      //ユーザがログインしたかどうか確認
      if (User.Identity.IsAuthenticated)
      {
         //ユーザ名を取得してセッション変数として保存
         var user = User.Identity;
         Session["uid"] = user.Name;
         //ログイン情報を表示
         Label1.Text = "You are logged in as ";
      }
      else {
          Session["uid"] = "";
         Label1.Text = "Hello Guest! Please sign in to view your profile.";
      }
  [VB]
          ・ユーザがログインしたかどうか確認
          If User.Identity.IsAuthenticated Then
             ・ユーザ名を取得してセッション変数として保存
            Dim user As MembershipUser = Membership.GetUser()
            Session("uid") = user.UserName.ToString
            Label1.Text = "You are logged in as "
          Else
                     SQL Anywhere を使った ASP.NET Web ページの作成
                                 バージョン 1.1 - 2009 年 3 月
```

```
Session("uid") = ""
Label1.Text = "Hello Guest! Please sign in to view your profile." End
If
```

13. [F5] を押すとページが表示されます。ユーザがサインインしていないことを示すメッセージがラベル に表示されます。

🕗 Profile - Mozilla Firefox		
Eile Edit View History	<u>3</u> ookmarks <u>T</u> ools <u>H</u> elp	
🔇 🔊 - C 🗙 🔞	http://localhost:113! 🏠	· 🛐 · Google 🔎 📑 🤱
🗋 Profile	÷	-
		<u> </u>
WIT ASP.INLT	AFFLICATION	
Home Profile		
Hello Guest! Please sign ir	to view your profile.	
•		V

14. ページの右上隅の [Log in] をクリックします。[Register] をクリックし、ユーザ名として "Matthew" (レ コードがデモ・データベースにすでに存在する従業員) を入力します。[password] にパスワードを設定 します。[Create User] をクリックします。

😢 Register - Mozilla Firefox	_ 🗆 🗵
Eile Edit View History Bookmarks Iools Help	
🕜 💽 - C 🗙 🏠 📄 http://localhost:2359/Sample_asp.net/Account/Register.aspx?Re 🏠 🔹 🔀 - Goog	le 🔎
Register 🔅	
	<u> </u>
My ASP.NET Application	
Home Profile	
CREATE A NEW ACCOUNT	
Use the form below to create a new account.	
Passwords are required to be a minimum of 6 characters in length.	
A securit Information	
Account information	
User Name:	
Matthew	
E-mail:	
mat@abc.com	
Decement	
•••••••	
Coninni Password:	_
	-

15. ホーム・ページにはパブリック情報が表示され、プロファイル・ページの DetailsView

コントロールには、ユーザ Matthew の個人情報が、Employee テーブルのすべてのカラムを使って表示されます。

Profile - Mozilla Firefox		_	۵×
ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory	<u>B</u> ookmarks	Tools Help	
<>- C ×	🏠 🗋 h	ttp://localhost:2359/Sample_asp.net/Profile.a 🏠 👻 🔀 🗧 Google	P
Profile		*	-
Home Profile			
You are logged in as N	fatthew		
EmployeeID	105		
ManagerID	501		
Surname	Cobb		
GivenName	Matthew		
DepartmentID	100		
Street	7 Pleasant Street		
City	Grimsby		
State	UT		
Country	USA		
PostalCode	02154		
Phone	6175553840		
Status	A		
SocialSecurityNumber	052345739		
Salary	62000.000		
StartDate	01/01/1985 12:00:00 AM		
TerminationDate			
BirthDate	04/12/1960 12:00:00 AM		
BenefitHealthInsurance	V		
BenefitLifeInsurance	V		
BenefitDayCare	Γ		
Sex	м		
L	·]		ЪĽ

- 16. [Log out] をクリックし、再びページにアクセスします。ユーザが認証されていないため、情報が表示されなくなります。
- 17. Web ブラウザを閉じます。
- 18. Sybase Central に戻って [Tables] を展開し、テーブル aspnet_users を選択してデータを表示します。ユーザ "Matthew"の認証情報が格納された新しいローが作成されています。

Sybase Central						
Eile Edit View Tools Connections Mode Help						
	😰 🕶 🔀 🛣 🗡	K) (24 🗗				
Context: Sybase Central/SQL Anywhere 12/demo12/	2/demo - DBA/Tables/aspnet_Users (DB/	N)				•
% Tools 🔻 🔟 🐞 🕼 🛅 🖏 🛱 🔽						
Folders X	aspnet_Users (DBA)					
se Central	Columns Constraints Referencing	Constraints Inde	exes Text Indexes Tri	iggers Depende	nt Views Data	
QL Anywhere 12	ApplicationId UserId	UserName	LoweredUserName	MobileAlias	IsAnonymous	LastActivityDate
demoi2	1 CF723CD3-EBD6 6AEAEA37.	Matthew	matthew	(NULL)		0 2010-07-16 20:19:31.427
Tables		a posta se se se	Protection of	100-000 C		
aspiret_PersonalizationAllUsers (DBA) aspiret_PersonalizationAllUsers (DBA) aspiret_PersonalizationAllUsers (DBA) aspiret_PersonalizationAllUsers (DBA) aspiret_Roles (DBA) aspiret_Schematersions (DBA) Spriet_Users (DBA) aspiret_UsersInRoles (DBA) Contacts (GROUPO) Customers (GROUPO) Customers (GROUPO) FinancialData (GROUPO) FinancialData (GROUPO) SalesOrderItems (GROUPO) SalesOrderItems (GROUPO) SalesOrderItems (GROUPO)	4					

 テーブル aspnet_Membership を選択し、[Data] タブをクリックします。Matthew のメンバシップ情報を格納 するための新しいローが追加されています。Profiles Provider など、他の ASP.NET プロバイダを使用する場 合、Sybase Central を使用して対応するテーブルに格納されているデータを確認したり管理したりできます。



Dynamic Data 機能の有効化

データ駆動型 Web アプリケーションの場合、通常は、データベースから取得したデータの外観と動作をカスタマイズする必要があり ます。Dynamic Data 機能を使用しない場合、希望するフォーマットでデータが表示されるように、プログラマが SQL 文を操作する必要がありますが、これはうんざりする作業であるうえ、コードの読みやすさが低下します。 ASP.NET Dynamic Data は、実行時にデータ・エンティティの外観と動作を推測し、そこからユーザ・インタフェース の動作を導き出すことにより問題を解決します。ほんの少しの手順とわずかなコードの記述だけで、.NET 4.0 の Dynamic Data 機能を利用して、SQL Anywhere データベースに格納されているデータの表示をフォーマットできます。

手順

 単純な Web サイトを実行し、ユーザ Matthew としてログインします。StartDate および BirthDate カラムの データ・フォーマットが、希望するフォーマットで表示されていません。データベースには時間情報がない ため、12:00:00AM という時刻が自動的に生成されます。これら2つの日付が標準的な YYYY-MM-DD フォ ーマットで表示されるように、Employee クラスを修正します。

また、TerminationData フィールドは空ですが、これはデータベースにレコードが存在しないということです。 そこで、データが null の場合に表示されるデフォルト値を設定します。Web アプリケーションを閉じます。

2. Visual Studio に戻り、新しいクラスのファイルをフォルダ App_Code に追加し、この新しいファイルに Employee.cs または Employee.vb という名前を付けます。

3. 追加したファイルに以下のコード行を挿入します。

```
[C#]
using System.Web.DynamicData;
using System.ComponentModel.DataAnnotations;
```

```
namespace Mode1 {
```

}

} }

```
//Employee 部分クラスを拡張
[MetadataType(typeof(EmployeeMeta))]
public partial class Employee {
```

public class EmployeeMeta

//StartDate カラムを再フォーマット
[DisplayFormat(DataFormatString="{0:yyyy-MM-dd}")]
public object StartDate { get; set; }

[DisplayFormat(DataFormatString="{0:yyyy-MM-dd}")]
public object BirthDate { get; set; }

```
//TerminationDate カラムのデフォルト値を設定
[DisplayFormat(NullDisplayText="N/A")]
public object TerminationDate { get; set; }
```

```
28
```

[VB]

```
Imports Microsoft.VisualBasic
Imports System.ComponentModel.DataAnnotations
Imports System.Web.DynamicData
```

```
Namespace Model
```

```
<MetadataType(GetType(EmployeeMeta))> _
Partial Public Class Employee
End Class
```

Public Class EmployeeMeta

```
<DisplayFormat(DataFormatString:="{0:yyyy-MM-dd}")> _
    Public Property StartDate() As Object
        Get
        End Get
        Set(ByVal value As Object) End
        Set
    End Property
    <DisplayFormat(DataFormatString:="{0:yyyy-MM-dd}")> _
    Public Property BirthDate() As Object
        Get
        End Get
        Set(ByVal value As Object) End
        Set
    End Property
    <DisplayFormat(nulldisplaytext:="N/A")> _
    Public Property TerminationDate() As Object
        Get
        End Get
        Set(ByVal value As Object) End
        Set
    End Property
End Class
```

End Namespace

4. Open Profile.aspx and switch to Source view. Replace the markup for the GridView control and the EntityDataSource object with the following lines:

SQL Anywhere を使った ASP.NET Web ページの作成 バージョン 1.1 - 2009 年 3 月

29

ContextTypeName="Model.SAEntities" EntityTypeFilter="" Select=""> <whereParameters> <asp:SessionParameter DefaultValue="" Name="emp_name" SessionField="uid" DbType="String" /> </whereParameters> </asp:EntityDataSource>

これにより、DetailsView コントロールのバインド・フィールドが削除され、Dynamic Data がデータをフォーマットするようになります。

5. Profile.aspx ページの Page_Load イベントに以下の行を追加します。

[C#]

DetailsView1.EnableDynamicData(typeof(Model.Employee));

[VB]

DetailsView1.EnableDynamicData(GetType(Model.Employee))

6. ここで Web サイトを再び実行し、ユーザ Matthew としてログインします。StartDate、BirthData、および TerminationDate フィールドの表示フォーマットの変更が、Employee コード・ファイル内の注釈に従って行わ れています。

Profile - Mozilla Firefox		
e <u>E</u> dit ⊻iew Hi <u>s</u> tory	<u>B</u> ookmarks <u>T</u> ools	Help
	A bttp://	
		localitost.2559/Jampie_asp.net/Profile.2 💭 - 📑 🖓 Google 🔰
Profile		÷
My ASP.NE	T Applica	ATION
Home Profile		
You are logged in as N	Aatthew	
EmployeeID	105	
ManagerID	501	
Surname	Cobb	
GivenName	Matthew	
DepartmentID	100	
Street	7 Pleasant Street	
City	Grimsby	
State	UT	
Country	USA	
PostalCode	02154	
Phone	6175553840	
Status	А	
SocialSecurityNumber	052345739	
Salary	62000.000	
StartDate	1985-01-01	
TerminationDate	N/A	
BirthDate	1960-12-04	
BenefitHealthInsurance		
BenefitLifeInsurance	V	
	E	
BenefitDayCare	1	

7. Web ブラウザを閉じます。これでチュートリアルは終了です。

SQL Anywhere ASP.NET プロバイダの削除

デモ・データベースを元の状態に戻したい場合、SQL Anywhere ASP.NET プロバイダをアンインストールする必要 があります。

- 1. インストールの項で説明した [ASP.NET Security Schema Setup] ウィザードを起動し、デモ・データベース に接続します。
- 2. 削除するすべての機能を選択し、3番目の手順で [Preserve Data] の選択を解除して、ウィザードを完了します。
- 3. ウィザードにより、デモ・データベースから SQL Anywhere ASP.NET プロバイダのテーブルが削除されます。

まとめとその他のリソース

SQL Anywhere ASP.NET プロバイダと ASP.NET 4.0 の強化された Web サイト・テンプレートを利用することで、セ キュリティに対応した Web サイトを非常に簡単に構築できます。さらに、Visual Studio 2010 用の SQL Anywhere 統合 コンポーネントとその Entity Model のサポートは、EntityDataSources をデータバインド・サーバ・コントロールにバ インドすることにより、データ駆動型 Web アプリケーションの実装を可能にします。.NET 4.0 の新しい Dynamic Data 機能を使用すれば、マークアップや SQL クエリと格闘する必要なく、部分クラスを拡張するだけで、ビジネス・ ロジックを簡単に実装できます。

ホワイトペーパー、チュートリアル、サンプル・コードなどその他のリソースについては、以下のサイトの SQL Anywhere .NET Development Center を参照してください。

http://www.ianywhere.jp/developers/microsoft-net.html

