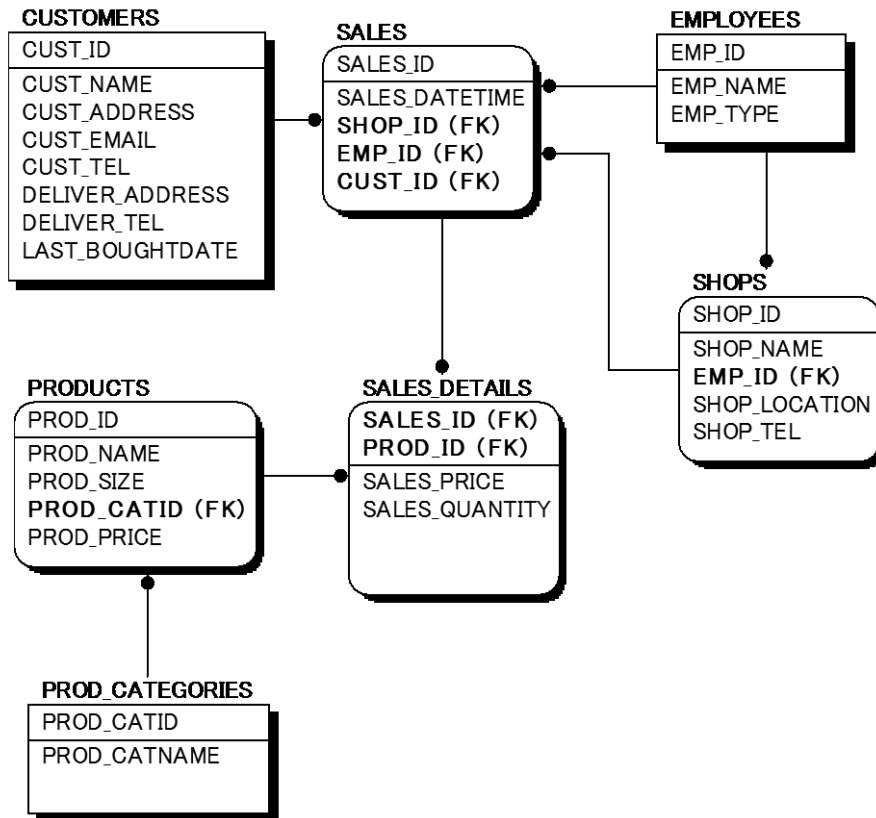


データベースの構成

本演習で用いるデータベースの構造は次のとおりです。



【CUSTOMERS】顧客データを格納

列名	説明
CUST_ID	顧客番号
CUST_NAME	顧客名
CUST_ADDRESS	顧客住所
CUST_EMAIL	顧客メールアドレス
CUST_TEL	顧客電話番号
DELIVER_ADDRESS	配送先住所
DELIVER_TEL	配送先電話番号
LAST_BOUGHTDATE	最終購入日

【EMPLOYEES】従業員データを格納

列名	説明
EMP_ID	従業員番号
EMP_NAME	従業員名
EMP_TYPE	従業員種別(S:正社員、A:アルバイト)

はじめに

【PRODUCTS】商品データを格納

列名	説明
PROD_ID	製品番号
PROD_SIZE	製品サイズ
PROD_CATID	製品カテゴリ番号
PROD_PRICE	製品単価

【PROD_CATEGORIES】商品分類データを格納

列名	説明
PROD_CATID	製品カテゴリ番号
PROD_CATNAME	製品カテゴリ名

【SHOPS】店舗データを格納

列名	説明
SHOP_ID	店舗番号
EMP_ID	店舗マネージャー番号
SHOP_LOCATION	店舗所在地
SHOP_TEL	店舗電話番号

【SALES】売上データを格納

列名	説明
SALES_ID	売上番号
SALES_DATETIME	売上日時
SHOP_ID	店舗番号
CUST_ID	顧客番号(店舗販売時は0)

【SALES_DETAILS】売上明細データを格納

列名	説明
SALES_ID	売上番号
PROD_ID	製品番号
SALES_PRICE	販売単価
SALES_QUANTITY	販売数量

練習1. 動的プロパティの設定と利用

「CoffeeShop.exe.config」ファイルに接続文字列を保存し、コードから読み込みます。

ステップ1. 接続文字列の設定

1. 「ソリューションエクスプローラー」で、プロジェクト名「CoffeeShop」を右クリックし、[プロパティ]を選択します。
2. 「設定」タブをクリックし、接続文字列として、以下の設定を追加します。

名前	CoffeeConStr
型	(接続文字列)
スコープ	アプリケーション
値	Data Source = データベースサーバー名; Initial Catalog = データベース名

	名前	型	スコープ	値
▶	coffeeDBCon...	(接続文字列)	アプリケーション	Data Source=instrutor;Initial Catalog=coffeeDB;Integrated Secur
	CoffeeConStr	(接続文字列)	アプリケーション	Data Source=Instrutor;Initial Catalog=CoffeeDB
*				

ステップ2. app.config ファイルの確認

1. 「ソリューションエクスプローラー」から、「app.config」ファイルをダブルクリックして開きます。
2. <connectionStrings>セクションに、接続文字列が以下のように設定されていることを確認します。

【VB】

1. <add name="CoffeeShop.My.MySettings.CoffeeConStr"...

【C#】

1. <add name="CoffeeShop.Properties.Settings.CoffeeConStr"...

ステップ3. 接続文字列の読み込み

1. 「ソリューションエクスプローラー」から、「Coffee.vb」または「Coffee.cs」ファイルをダブルクリックして開きます。
2. 以下のコードを確認します。

【VB】

1. ' 接続文字列を格納する変数
2. Public Shared coffeeConStr As String

【C#】

1. //接続文字列を格納する変数
2. public static string coffeeConStr;

ワンポイント:ここで宣言している「coffeeConStr」変数は、Login フォームで接続文字列を保存し、各フォームでデータベースにアクセスするために使用します。

3. 「System.Configuration」アセンブリを参照設定します。
4. 「Login」フォームの「btnLogin_Click」イベントハンドラーを表示します。
5. 「btnLogin_Click」イベントハンドラーの最後の行に、以下の条件を満たすコードを追加します。

- 構成ファイルから接続文字列を読み込み、「Coffee」クラスの静的メンバーである「coffeeConStr」変数に代入する。

【VB】

1. ' 構成ファイルから接続文字列を読み込みます
2. Coffee.coffeeConStr =
3. System.Configuration.ConfigurationManager.ConnectionStrings(
4. "CoffeeShop.My.MySettings.CoffeeConStr").ConnectionString

【C#】

1. //構成ファイルから接続文字列を読み込みます
2. Coffee.coffeeConStr =
3. System.Configuration.ConfigurationManager.ConnectionStrings[
4. "CoffeeShop.Properties.Settings.CoffeeConStr"].ConnectionString;

練習2. Windows 認証と SQL Server 認証

「Login」フォームを使用して、Windows 認証も SQL Server 認証も受け入れられるようにします。

ステップ1. 接続文字列の設定

1. 「Login」フォームをコードエディタで開きます。
2. 「Login」クラスの前頭に、以下の条件を満たすコードを追加します。

● SqlConnection オブジェクトを格納するための変数として、「cnCoffee」を宣言する。

【VB】

1. 'Connection オブジェクト変数の宣言
2. Private cnCoffee As SqlConnection

【C#】

1. //Connection オブジェクト変数の宣言
2. private SqlConnection cnCoffee;

3. 「Login」フォームの「btnLogin_Click」イベントハンドラーを表示します。
4. 「btnLogin_Click」イベントハンドラーの最後の行に、以下の条件を満たすコードを追加します。

- 「Windows 認証」オプションが選択されていた場合は、Windows 認証を利用し、「SQL Server 認証」オプションが選択されていた場合は、SQL Server 認証を利用する。
- SQL Server 認証が選択されていた場合には、接続文字列にユーザー名とパスワードも追加する。

第2章演習

- 接続文字列は、「Coffee.coffeeConStr」変数に追加する。
- Connection オブジェクト(cnCoffee)を生成した後、ConnectionString プロパティに接続文字列 (Coffee.coffeeConStr)を代入する。
- 接続をオープンし、成功したら Login フォームを隠す(Hide メソッド)。
- 接続に失敗した場合は、ラベル(lblMessage)に「ユーザー名またはパスワードが間違っています」というメッセージを表示する。
- 接続に成功しても、失敗しても接続はクローズする。

【VB】

```
1. 'Windows 認証か SQL Server 認証かで接続文字列を変更します
2. If optWindows.Checked = True Then
3.
4.     Coffee.coffeeConStr += ";Integrated Security=True"
5. Else
6.
7.     Coffee.coffeeConStr += ";User ID=" & txtUser.Text
8.     Coffee.coffeeConStr += ";Password=" & txtPassword.Text
9. End If
10. 'Connection オブジェクトを作成し、接続文字列をセットします
11. cnCoffee = New SqlConnection
12. cnCoffee.ConnectionString = Coffee.coffeeConStr
13.
14. '接続に成功したら、メインフォームを表示します
15. Try
16.
17.     cnCoffee.Open()
18.     Me.Hide()
19. Catch ex As SqlException
20.
21.     lblMessage.Text = "ユーザー名またはパスワードが間違っています"
22. Finally
23.
24.     cnCoffee.Close()
25. End Try
```

【C#】

```

1. //Windows 認証か SQL Server 認証かで接続文字列を変更します
2. if ( optWindows.Checked == true )
3. {
4.     Coffee.coffeeConStr += ";Integrated Security=True";
5. }else
6. {
7.     Coffee.coffeeConStr += ";User ID=" + txtUser.Text;
8.     Coffee.coffeeConStr += ";Password=" + txtPassword.Text;
9. }
10. //Connection オブジェクトを作成し、接続文字列をセットします
11. cnCoffee = new SqlConnection();
12. cnCoffee.ConnectionString = Coffee.coffeeConStr;
13.
14. //接続に成功したら、メインフォームを表示します
15. try
16. {
17.     cnCoffee.Open();
18.     this.Hide();
19. }catch ( SqlException ex )
20. {
21.     lblMessage.Text = "ユーザー名またはパスワードが間違っています";
22. }finally
23. {
24.     cnCoffee.Close();
25. }

```

ステップ2. スタートアップフォームの変更

1. VB のみ「ソリューションエクスプローラー」で、プロジェクト名「CoffeeShop」を右クリックし、[プロパティ]を選択します。
2. VB のみ「アプリケーション」タブをクリックし、「スタートアップオブジェクト」で「Sub Main」を選択します。
3. C#のみ「ソリューションエクスプローラー」で、「Program.cs」ファイルをダブルクリックして開き、以下のコードを探します。

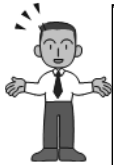
第2章演習

```
1. Application.Run(new Employees());
```

4. C#のみ手順3で探したコードを以下のように書き換えます。

```
1. Application.Run(Coffee.frmMain);
```

5. [デバッグ]メニューから[デバッグ開始]を選択し、アプリケーションを実行します。
6. 「ログイン」ウィンドウで「Windows 認証」を選択し、「ログイン」ボタンをクリックします。
- 「メイン」ウィンドウは表示されましたか？ それはなぜですか？



7. [デバッグ]メニューから[デバッグの停止]をクリックし、アプリケーションの実行を停止します。
8. [デバッグ]メニューから[デバッグ開始]を選択し、アプリケーションを実行します。
9. 「ログイン」ウィンドウで「SQL Server 認証」を選択し、「ユーザー名」として「データベースユーザー名」、「パスワード」として「データベースユーザーパスワード」を入力し、「ログイン」ボタンをクリックします。

「メイン」ウィンドウは表示されましたか？ それはなぜですか？



10. [デバッグ]メニューから[デバッグの停止]をクリックし、アプリケーションの実行を停止します。
11. [デバッグ]メニューから[デバッグ開始]を選択し、アプリケーションを実行します。
12. 「ログイン」ウィンドウで「SQL Server 認証」を選択し、「ユーザー名」として「TestUser」、「パスワード」として「P@ssw0rd」を入力し、「ログイン」ボタンをクリックします。

「メイン」ウィンドウは表示されましたか？ それはなぜですか？



13. [デバッグ]メニューから[デバッグの停止]をクリックし、アプリケーションの実行を停止します。

練習3. 選択クエリの実行

「Products」フォームの ComboBox コントロールに、商品一覧を表示します。



ステップ1. DataReader の利用

1. 「Products」フォームの「Products_Load」イベントハンドラーを表示し、以下の条件を満たすコードを入力します。

- Connection オブジェクト(cnCoffee)に、接続文字列(Coffee.coffeeConStr)を代入し、接続をオープンする。
- Command オブジェクトを、「cmSelProds」という名前で宣言、作成し、「PRODUCTS」テーブルのすべてのレコードを取得する SQL ステートメントを実行する。実行結果は DataReader オブジェクト(dr)で取得する。
- DataReader オブジェクト(dr)で読みこんだレコードの「prod_name」列の値を、ComboBox コントロール(cmbProductName)に追加する。この際、読み込んだレコードに「prod_size」列の値が存在する場合は、「prod_name」列の値と組み合わせて ComboBox コントロールに追加する。
- DataReader オブジェクト(dr)や Connection オブジェクト(cnCoffee)を Close する。

【VB】

1. ' 接続文字列をセットし、接続を開きます
2. cnCoffee.ConnectionString = Coffee.coffeeConStr
3. cnCoffee.Open()
4. ' Products テーブルを取得します
5. Dim cmSelProds As New SqlCommand
6. Dim dr As SqlDataReader
7. cmSelProds.Connection = cnCoffee
8. cmSelProds.CommandType = CommandType.Text

```

9. cmSelProds.CommandText = "SELECT * FROM PRODUCTS"
10. dr = cmSelProds.ExecuteReader
11. 'Products テーブルを読み込み、cmbProductName に追加します
12. Do While dr.Read
13.
14.     Dim prodName As String
15.     Dim prodSize As String
16.
17.     prodName = dr("prod_name").ToString
18. prodSize = dr("prod_size").ToString
19.     '複数のサイズがある商品の場合は、サイズもリストに含めます
20. If prodSize <> "" Then
21.
22.         cmbProductName.Items.Add(prodName & " " & prodSize)
23.     Else
24.
25.         cmbProductName.Items.Add(prodName)
26.     End If
27. Loop
28. dr.Close()
29. cnCoffee.Close()

```

【C#】

```

1. //接続文字列をセットし、接続を開きます
2. cnCoffee.ConnectionString = Coffee.coffeeConStr;
3. cnCoffee.Open();
4. //Products テーブルを取得します
5. SqlCommand cmSelProds = new SqlCommand();
6. SqlDataReader dr;
7. cmSelProds.Connection = cnCoffee;
8. cmSelProds.CommandType = CommandType.Text;
9. cmSelProds.CommandText = "SELECT * FROM PRODUCTS";
10. dr = cmSelProds.ExecuteReader();
11. //Products テーブルを読み込み、cmbProductName に追加します
12. while ( dr.Read() )
13. {
14.     string prodName;
15.     string prodSize;

```