

**Microsoft®**



Microsoft®  
**SQL Server® 2012**

## SQL Server 2012 自習書シリーズ No.7

---

Integration Services 入門

---

Published: 2008 年 3 月 18 日

SQL Server 2012 更新版: 2012 年 9 月 30 日

有限会社エスキューエル・クオリティ



この文章に含まれる情報は、公表の日付の時点での Microsoft Corporation の考え方を表しています。市場の変化に応える必要があるため、Microsoft は記載されている内容を約束していません。この文書の内容は印刷後も正しいとは保障できません。この文章は情報の提供のみを目的としています。

Microsoft、SQL Server、Visual Studio、Windows、Windows XP、Windows Server、Windows Vista は Microsoft Corporation の米国およびその他の国における登録商標です。

その他、記載されている会社名および製品名は、各社の商標または登録商標です。

© Copyright 2012 Microsoft Corporation. All rights reserved.

# 目次

---

STEP 1. Integration Services の 概要とインストール .....	4
1.1 Integration Services とは.....	5
1.2 インポート/エクスポート ウィザードと SSIS デザイナー .....	7
1.3 Integration Services のインストール .....	9
1.4 自習書を試す環境について .....	10
STEP 2. インポート/エクスポート ウィザード による単純なデータ転送.....	11
2.1 データベースの作成 .....	12
2.2 テキスト ファイルの取り込み .....	14
2.3 Access データベース (.accdb/.mdb) のインポート.....	23
2.4 Excel データ (.xlsx/.xls ファイル) の取り込み.....	34
2.5 保存したパッケージ (.dtsx) の実行と編集 .....	43
2.6 データのエクスポート .....	60
STEP 3. SSIS デザイナーの基本操作 .....	67
3.1 作成するパッケージの概要 .....	68
3.2 データベースの作成 .....	69
3.3 SSIS デザイナーの起動 .....	70
3.4 データ フロー タスクによるデータの転送.....	72
3.5 派生列コンポーネントによるデータ変換 .....	84
3.6 参照コンポーネントによる別テーブルのデータ取得 .....	92
3.7 データ ビューアーによる転送中のデータの表示 .....	103
STEP 4. パッケージの実行 .....	108
4.1 パッケージの実行 (dtexecUI) .....	109
4.2 コマンド プロンプトからのパッケージ実行 (dtexec) .....	112
4.3 パッケージの定期実行 (SQL Server Agent ジョブ) .....	114
4.4 Integration Services とデータ ウェアハウスの関係 ～ETL～ .....	119
4.5 BI システムとしての SQL Server .....	121

# STEP 1. Integration Services の 概要とインストール

---

この STEP では、Integration Services の概要とインストール方法、自習書を試すための環境などを説明します。

この STEP では、次のことを学習します。

- ✓ Integration Services の概要
- ✓ Integration Services のインストール方法
- ✓ 自習書を試す環境について



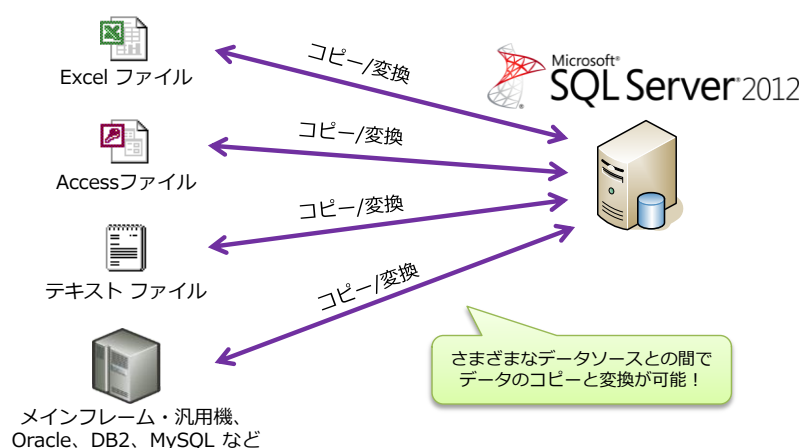
## 1.1 Integration Services とは

### ➡ Integration Services とは

**SQL Server Integration Services (SSIS : データ統合サービス)** は、SQL Server の標準機能として搭載されている、データの「コピー」や「変換」などが行える "**データ転送ツール**" です。SQL Server 2000 以前のバージョンでは、**DTS** (Data Transformation Services : データ変換サービス) と呼ばれていました。

Integration Services を利用すると、SQL Server 同士でのデータ転送はもちろん、Oracle や DB2、Microsoft Office Access、その他の ODBC 対応のデータベース、Microsoft Office Excel ファイル、可変長のテキスト ファイル (カンマ区切り、タブ区切り)、固定長のテキスト ファイルなど、さまざまなデータ ソースから SQL Server へデータを取り込んだり、それとは逆に SQL Server からデータを書き出したりすることができます。

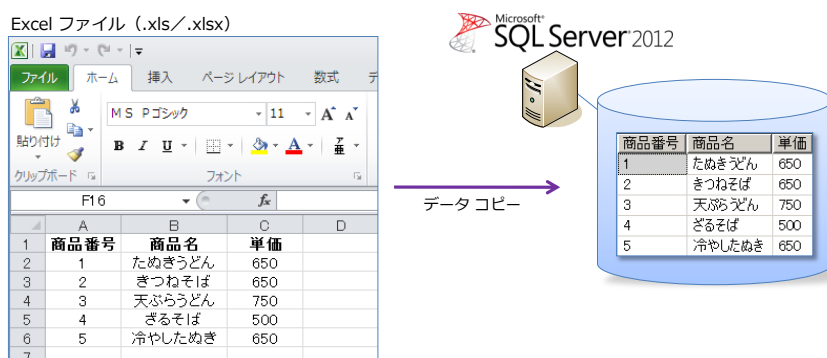
Integration Services は、データ転送・変換ツール



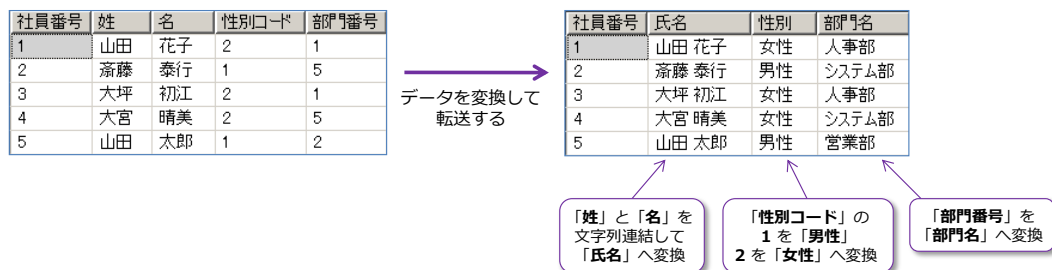
### ➡ Integration Services によるデータ転送・変換の例

Integration Services を利用すると、次のようなさまざまなデータ転送が行えます。

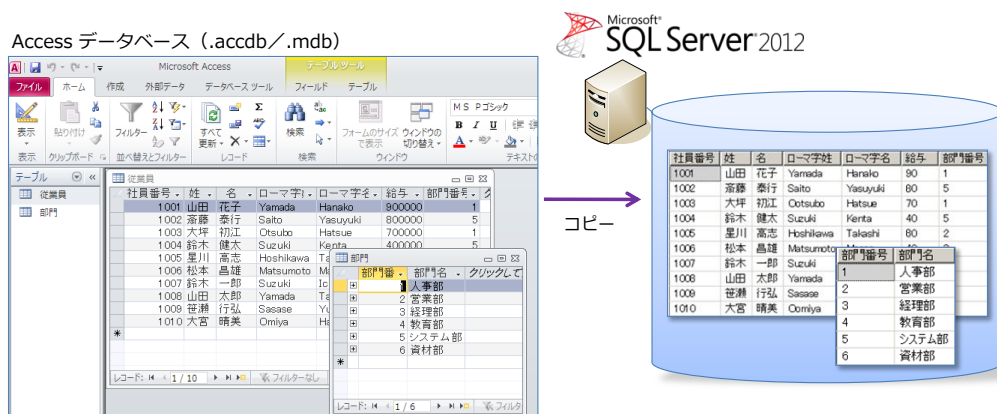
- 2 台の SQL Server の間でデータのコピーや変換を行う
- Excel データ (.xls/.xlsx ファイル) を SQL Server へ取り込む (インポートする)



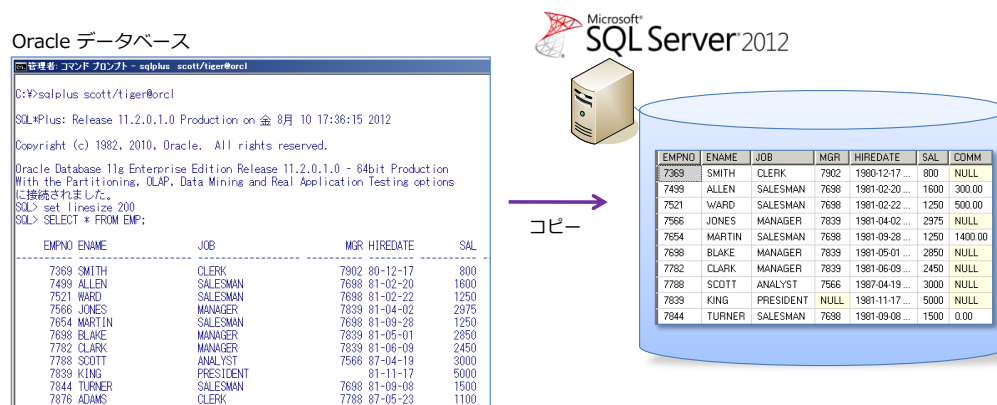
- データを変換して転送する



- Access データベース (.accdb/.mdb ファイル) を SQL Server へ取り込む



- Oracle データベースを SQL Server へ取り込む



- SQL Server のデータを Oracle へコピー／変換する
- SQL Server のデータをカンマ区切りのテキスト ファイルへ書き出す（エクスポートする）
- Oracle データベースを Excel ファイルへコピーする（SQL Server とは関係のないところでのデータ転送も可能）

このように Integration Services を利用すると、さまざまなデータベースまたはファイルとの間で SQL Server とのデータ転送が行えるので、色々な場面で役立ちます。特にデータを変換（データを加工しながら転送）できる機能は、非常に便利で、データ ウェアハウス(DWH:Data Warehouse)を構築する際には、欠かせないツールになります。Integration Services とデータ ウェアハウスとの関係については、STEP 4 で説明しています。

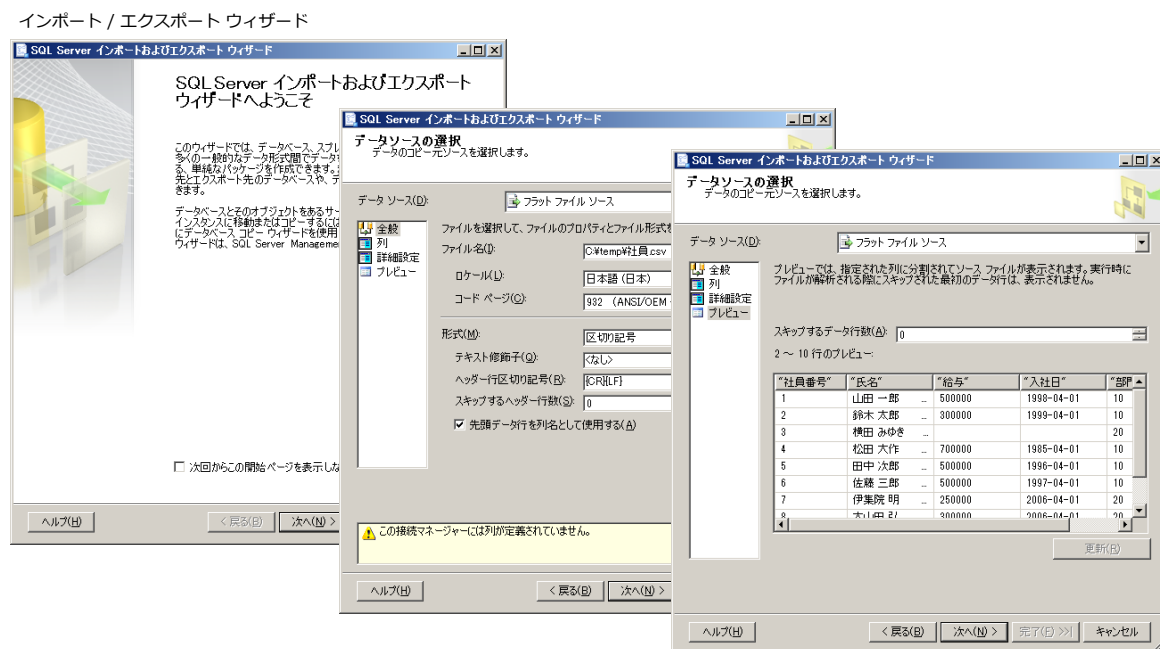
## 1.2 インポート/エクスポート ウィザードと SSIS デザイナー

### ➡ インポート/エクスポート ウィザードと SSIS デザイナー

Integration Services (SSIS) には、「SQL Server インポートおよびエクスポート ウィザード」(以下、インポート/エクスポート ウィザード)と「SSIS デザイナー」という 2つの機能があります。

### ➡ インポート/エクスポート ウィザード

インポート/エクスポート ウィザードは、[スタート] メニューや SQL Server Management Studio から起動できるツールで、ウィザード形式(対話形式)で指定された項目を入力、選択するだけで簡単にデータの転送(コピー)を行うことができます。

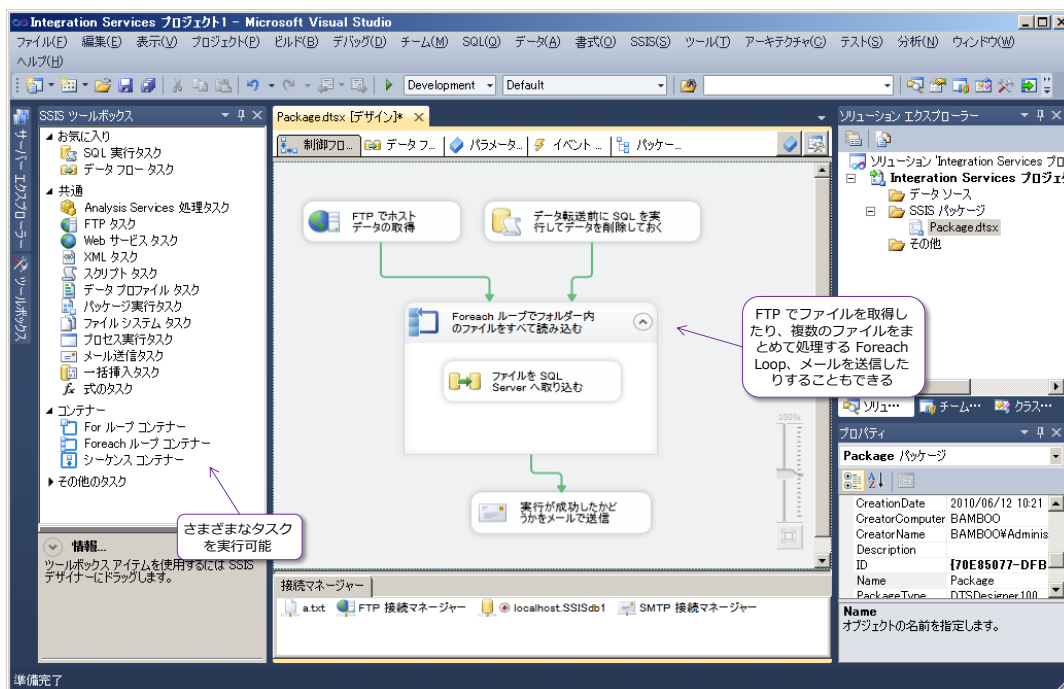


このウィザードの利用方法は、STEP2 で説明します。非常に簡単にデータ転送が行えるので、ぜひ試してみてください。

### ➡ SSIS デザイナー

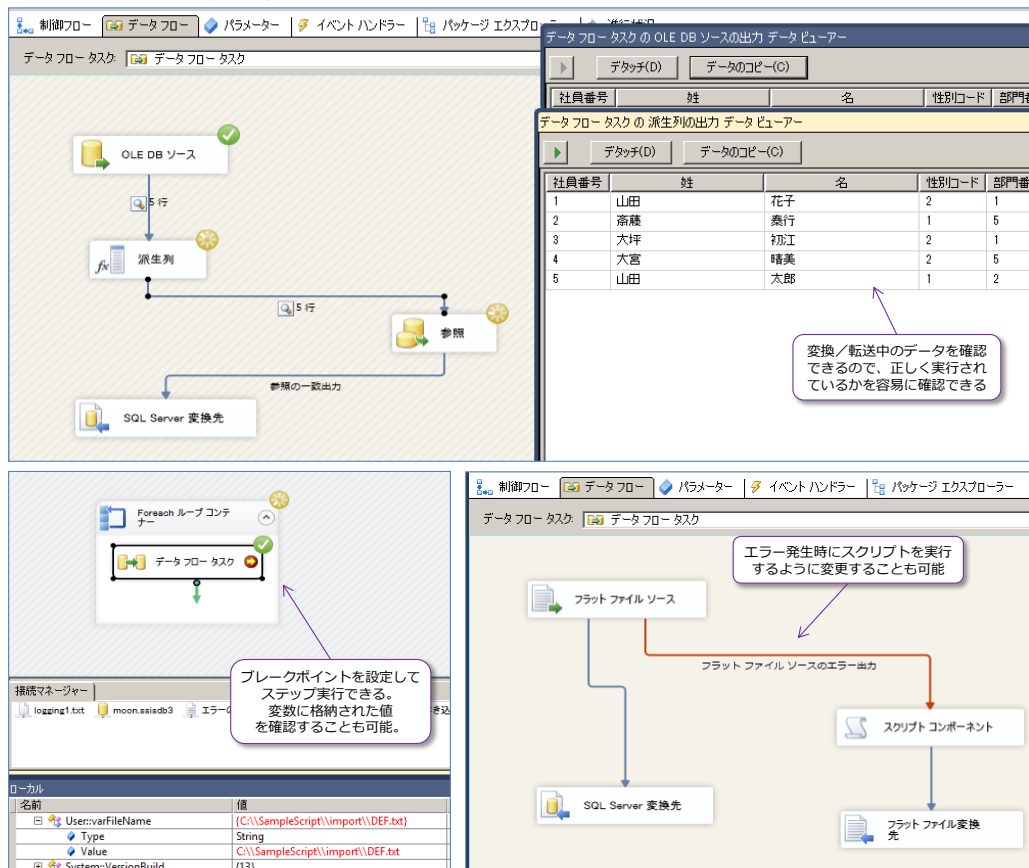
SSIS デザイナーは、中・上級者向けのツールで、SQL Server Data Tools (以前のバージョンの Business Intelligence Development Studio) の Integration Services プロジェクトから利用します。インポート/エクスポート ウィザードでは、単純なデータ転送しか行えませんが、SSIS デザイナーを利用すれば、データの転送に加えて、複雑な変換処理を追加したり、転送の前後に行いたい処理(FTP によるファイル取得やメール送信、SQL の実行など)を追加したりすることができます。

## SSIS デザイナーの利用例



また、SSIS デザイナーには、デバッグ機能も充実していて、データ転送中のデータを確認したり、ブレークポイントを設定して、ステップ実行(配置したコンポーネントを1つ1つ実行)をしたり、実行中の変数の値を確認したりすることも可能です。

### SSIS デザイナーはデバッグ機能も充実

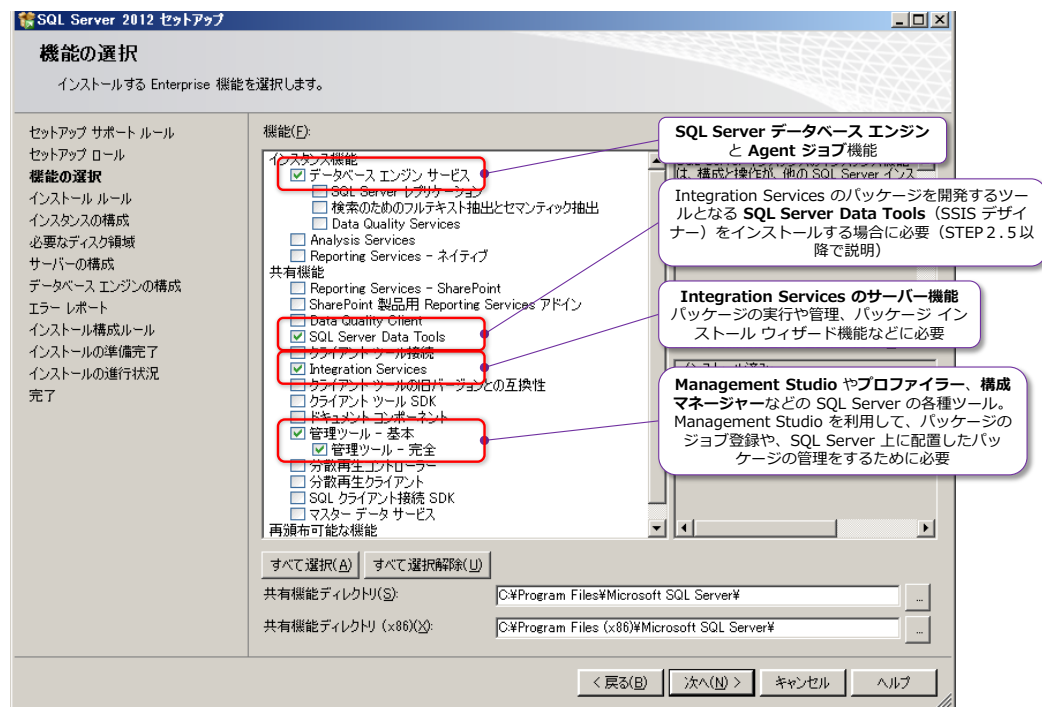


SSIS デザイナーの使用方法については、STEP 2.5 以降で説明しています。

## 1.3 Integration Services のインストール

### ➡ Integration Services のインストール

Integration Services のすべての機能を利用するには、SQL Server 2012 のインストール時の[機能の選択]画面で次のコンポーネントを選択します。



**データベース エンジン サービス**は、SQL Server 2012 のデータベース エンジンと SQL Server Agent サービスです。パッケージを定期実行したい場合には、Agent サービスのジョブ機能を利用します（STEP 4 で説明します）。

**SQL Server Data Tools**（以前のバージョンの Business Intelligence Development Studio）は、「**SSIS デザイナー**」を利用して、Integration Services のパッケージ（データ転送を設定したファイル）を開発するために必要なツールです。

**Integration Services** は、サーバー機能（パッケージの実行や管理、パッケージ インストール ウィザード機能など）を利用するために必要となるサービスです（詳しくは、本自習書シリーズの「**Integration Services 応用**」編で説明していますので、こちらもぜひご覧いただければと思います）。

**管理ツール**は、SQL Server の管理ツールである Management Studio やプロファイラー、構成マネージャーなどがインストールされ、Management Studio を利用して、パッケージのジョブ登録や、SQL Server 上に配置したパッケージを管理するために必要になります。

## 1.4 自習書を試す環境について

---

### ➡ 必要な環境

この自習書の手順を試すために必要な環境は次のとおりです。

- OS

Windows Server 2008 SP2 以降 または  
Windows Server 2008 R2 SP1 以降 または  
Windows Server 2012 または  
Windows Vista SP2 以降 または Windows 7 SP1 以降 または Windows 8

- ソフトウェア

SQL Server 2012

SQL Server 2012 のインストール時には、次のコンポーネントを選択して、インストールしておく必要があります（それぞれの役割については前のページを参照してください）。

- データベース エンジン サービス
- Integration Services
- SQL Server Data Tools
- 管理ツール

この自習書内での画面やテキストは、OS に Windows Server 2008 R2（x64）SP1、ソフトウェアに SQL Server 2012 Enterprise エディション（x64）を利用して記述しています。

### サンプル スクリプト

この自習書を試すには、サンプル スクリプトをダウンロードしておく必要があります。サンプル スクリプトには、各 STEP のインポートの実習で使用するファイル(CSV ファイルや SQL スクリプト) が含まれています。

## STEP 2. インポート/エクスポート ウィザード による単純なデータ転送

---

この STEP では、Management Studio から**インポート/エクスポート ウィザード**を利用して、テキスト ファイル（CSV ファイル）や Excel ファイル、Access データベースを SQL Server へ取り込んだり、それとは逆に SQL Server のデータをテキスト ファイルへ書き出したり、パッケージとして保存したりする方法について説明します。

この STEP では、次のことを学習します。

- ✓ テキスト ファイルの取り込み（インポート）
- ✓ Access データベース（.accdb/.mdb ファイル）の取り込み
- ✓ Excel データ（.xlsx/.xls ファイル）の取り込み
- ✓ パッケージの保存と実行、編集
- ✓ データの書き出し（エクスポート）

## 2.1 データベースの作成

### ➡ データの取り込み先となるデータベースの作成「ssisdb1」

この STEP では、インポート/エクスポート ウィザードを使用して、さまざまなデータを SQL Server 内へ取り込んでいくので、まずは、取り込み先（転送先）となる空のデータベースとして「ssisdb1」という名前のデータベースを作成しておきます。作成手順は、次のとおりです。

1. まずは、[スタート] メニューの [すべてのプログラム] から [Microsoft SQL Server 2012] の [SQL Server Management Studio] をクリックして、Management Studio を起動します。

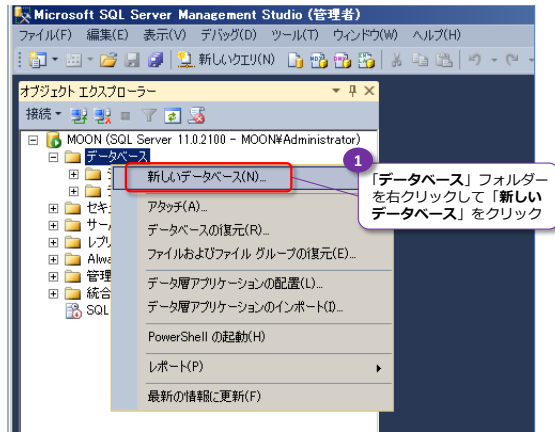


2. 起動後、[サーバーへの接続] ダイアログが表示されたら、[サーバー名] へ SQL Server の名前（画面は MOON）を入力し、[接続] ボタンをクリックします。

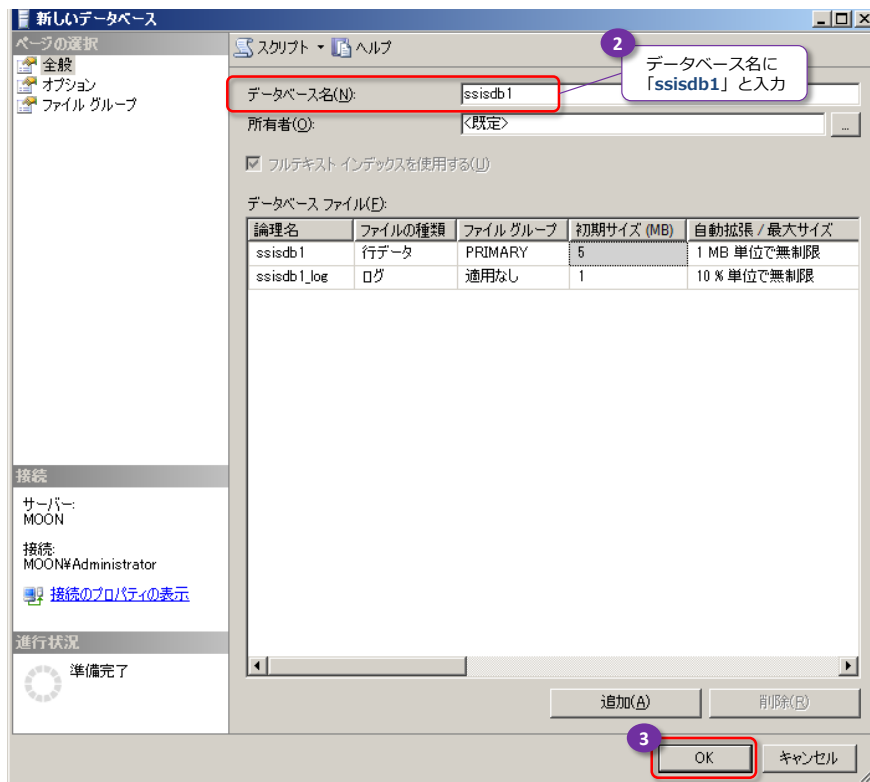


3. 接続完了後、次のようにオブジェクト エクスプローラーで [データベース] フォルダを右クリックして、[新しいデータベース] をクリックします。





4. 「新しいデータベース」ダイアログが表示されたら、[データベース名] へ「**ssisdb1**」と入力して、[OK] ボタンをクリックします。

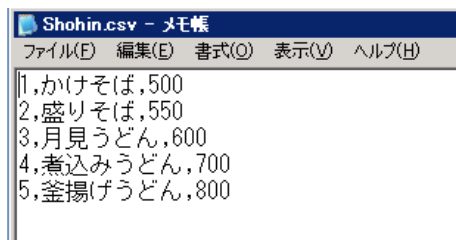


以上で、「**ssisdb1**」データベースの作成が完了です。

## 2.2 テキスト ファイルの取り込み

### ➡ 取り込むテキスト ファイル

ここでは、次の **CSV ファイル**（カンマ区切りのテキスト ファイル）を SQL Server へ取り込む（インポートする）手順を説明します。

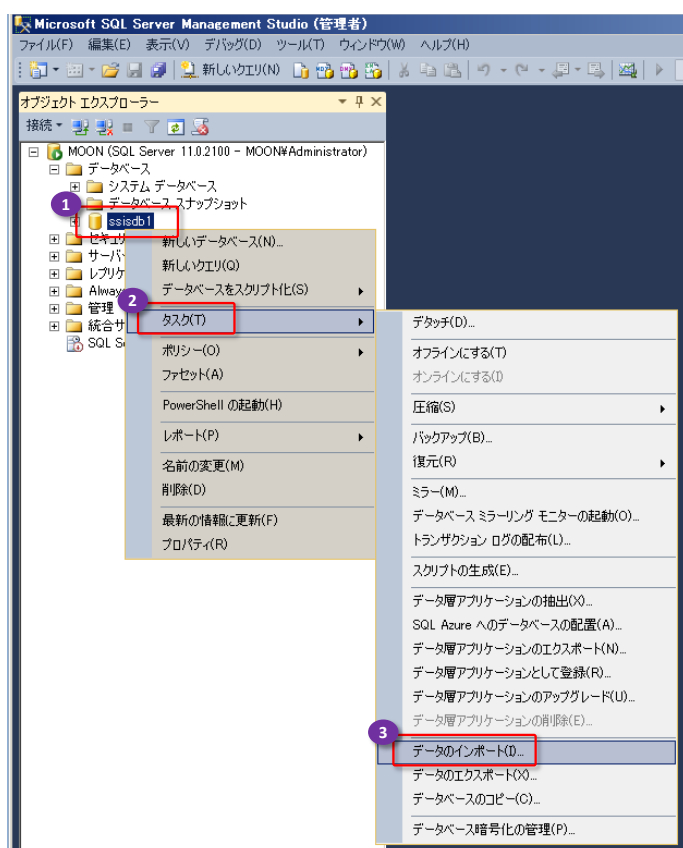


このファイルは、サンプル スクリプト内に「**Shohin.csv**」という名前で置いてあります。

### ➡ Let's Try

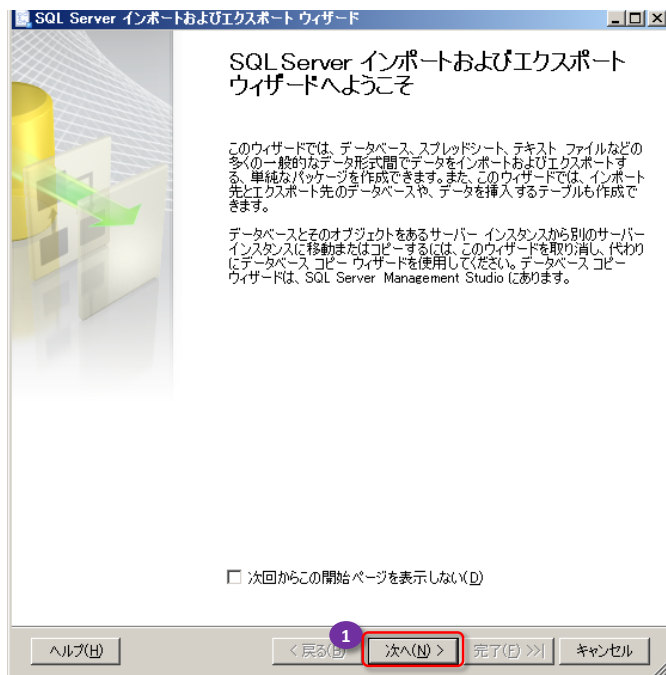
それでは、**インポート/エクスポート ウィザード**を利用して、データをインポートしてみましょう。

1. **インポート/エクスポート ウィザード**を利用するには、次のように Management Studio で取り込み先となるデータベース（今回は **ssisdb1**）を右クリックして、[タスク] メニューの [データのインポート] をクリックします。

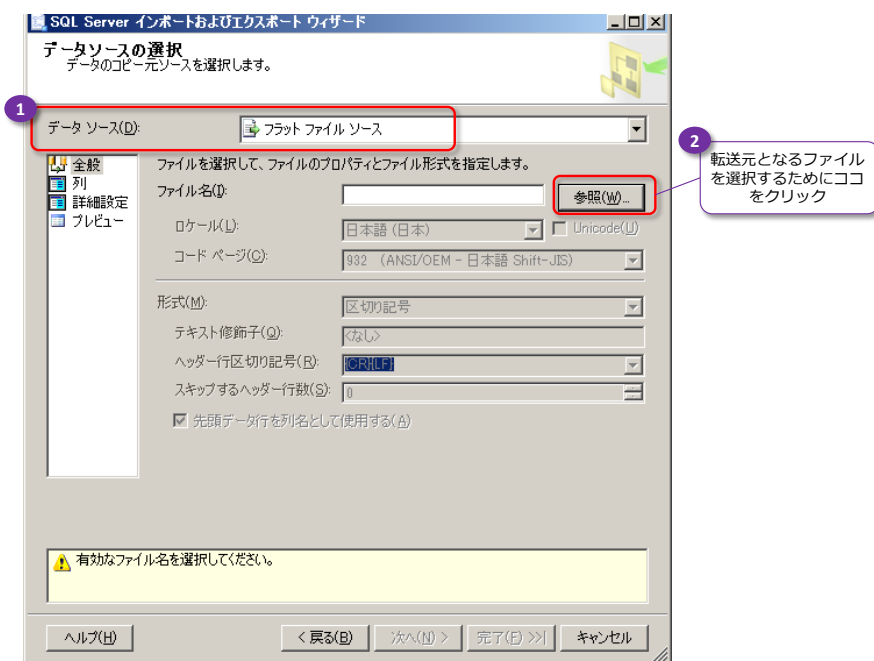


2. これにより、**インポート/エクスポート ウィザード**（SQL Server インポートおよびエクスポート

ート ウィザード) が起動するので、[次へ] ボタンをクリックします。

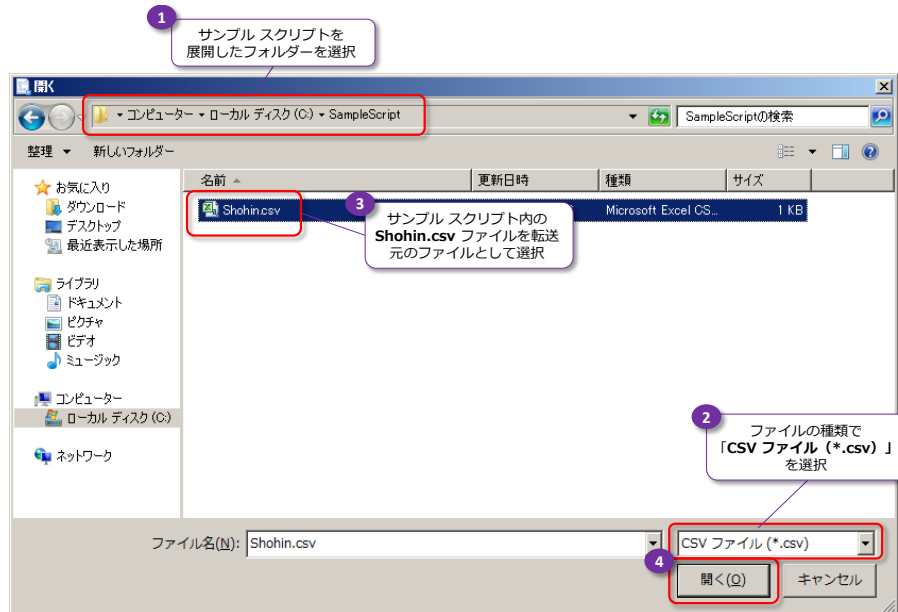


3. 次の[データ ソースの選択] ページでは、データの転送元を選択します。転送元がテキスト ファイルの場合は、[データ ソース] で「フラット ファイル ソース」を選択します。



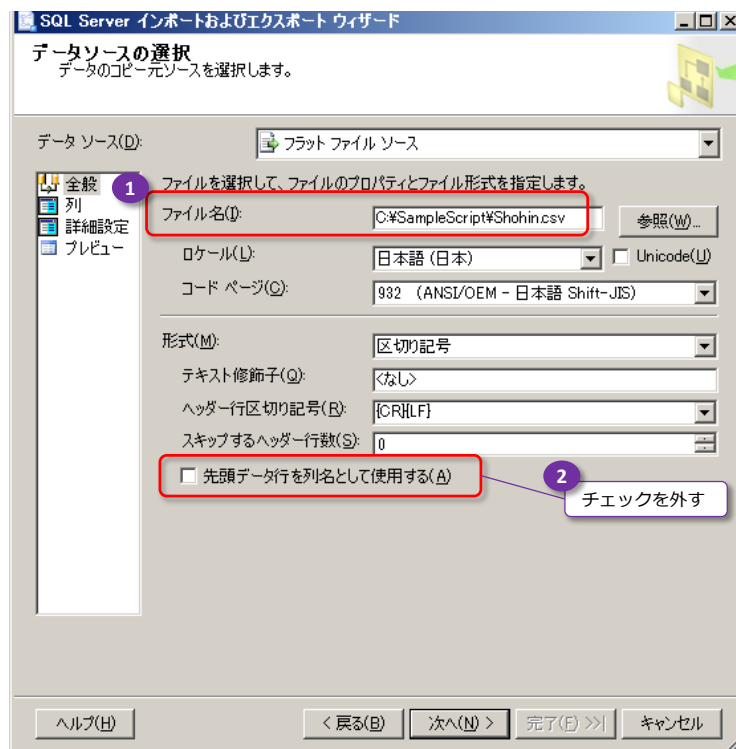
次に、[参照] ボタンをクリックして、転送元となるファイルを選択します。

4. 次のように[ファイルの場所]でサンプル スクリプトを展開したフォルダーを選択して、[ファイルの種類]で「CSV ファイル (\*.csv)」を選択します。

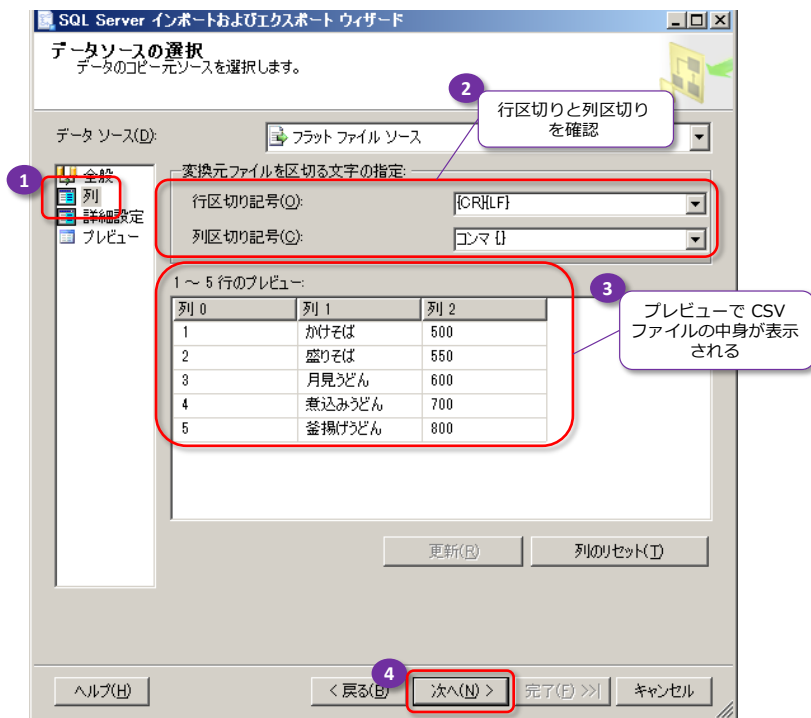


フォルダー内の CSV ファイルが表示されたら、「Shohin.csv」ファイルを選択して、[開く] ボタンをクリックします。

5. [データ ソースの選択] ページへ戻ったら、次のように [ファイル名] へ「Shohin.csv」へのパスが表示されているのを確認し、[先頭データ行を列名として使用する] のチェックを外します (Shohin.csv ファイルには、列名となるデータが先頭にはないので、チェックを外します)。



6. 続いて、次のように [列] ページをクリックして開きます。ここでは、テキスト ファイルのデータ形式を設定することができます。

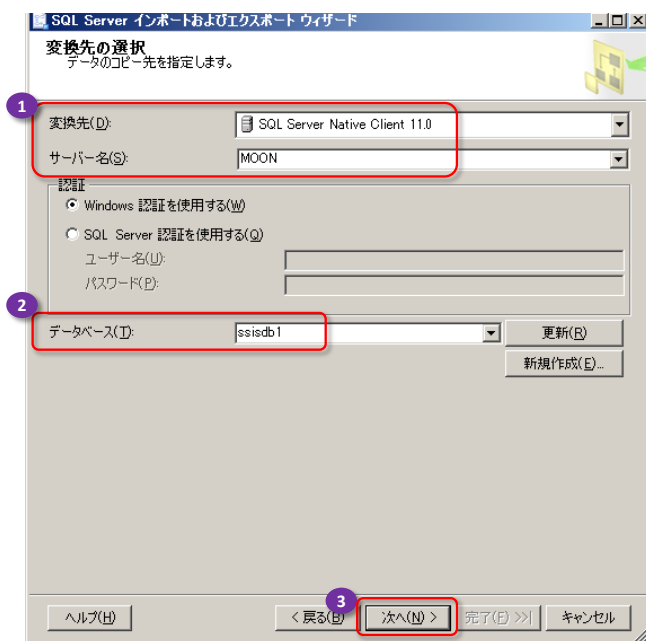


「行区切り記号」に "改行" を表す {CR}{LF} が選択され、「列区切り記号」に コンマ{,} が選択されていることを確認します。今回取り込むファイルは、CSV（カンマ区切り）ファイルなので、このままの設定で大丈夫です。

「1 ～ 5 行のプレビュー」には、転送元の CSV ファイルのデータが表示され、列名が自動的に「列 0」、「列 1」、「列 2」と命名されていることを確認できます。

確認後、「次へ」ボタンをクリックします。

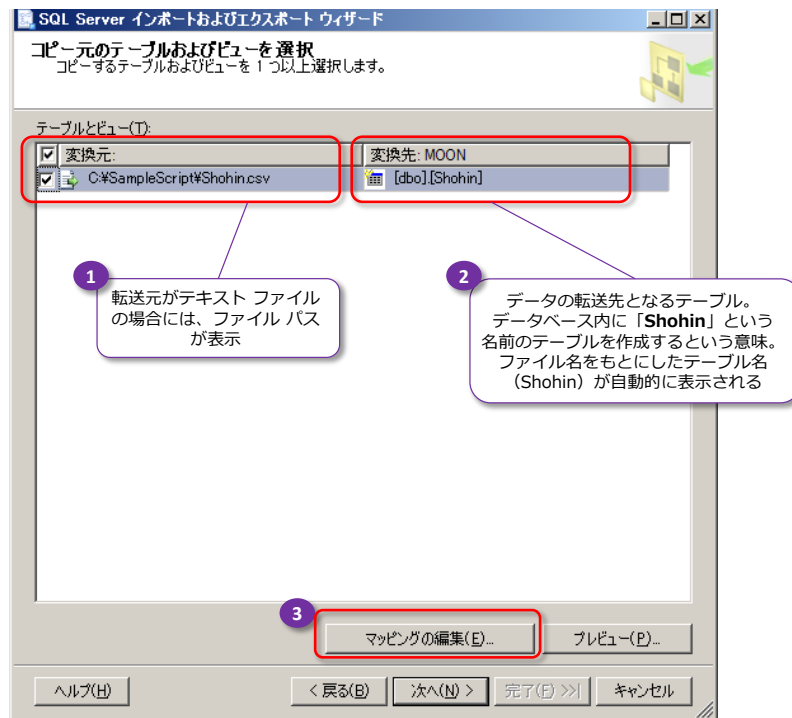
7. 次の「変換先の選択」ページでは、データの転送先（取り込み先）を指定します。



今回は、「ssisdb1」データベースを選択してウィザードを起動しているので、すでに「データベース」に「ssisdb1」が選択されていることを確認できます。また、「変換先」には「SQL

Server Native Client 11.0」が選択されていますが、これは SQL Server 2012 を転送先にするという意味です（SQL Server 2012 の内部的なバージョン番号は **11.X** なので、Native Client 11.0 が選択されています）。[サーバー名] には、SQL Server の名前が選択されていることも確認して、[次へ] ボタンをクリックします。

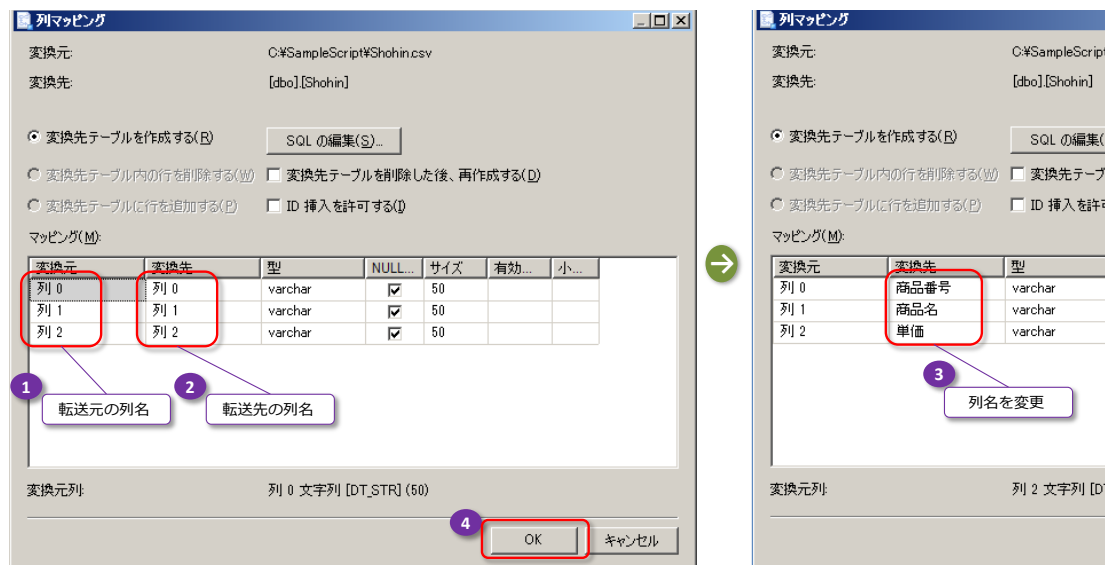
8. 次の[コピー元のテーブルおよびビューを選択] ページでは、転送元と転送先となるテーブルを指定します。今回のように転送元がテキスト ファイルの場合には、[変換元] ヘファイル パス（Shohin.csv ファイルへのパス）が表示されます。



[変換先] へは、[dbo].[Shohin] と表示されますが、これは「ssisdb1」データベース内へ「Shohin」という名前のテーブルを作成するという意味です（テーブルは、実際にデータが転送されるときに自動作成されます）。テーブル名は、ここで変更することもできますが、今回は **Shohin** という名前のままにしておきます。

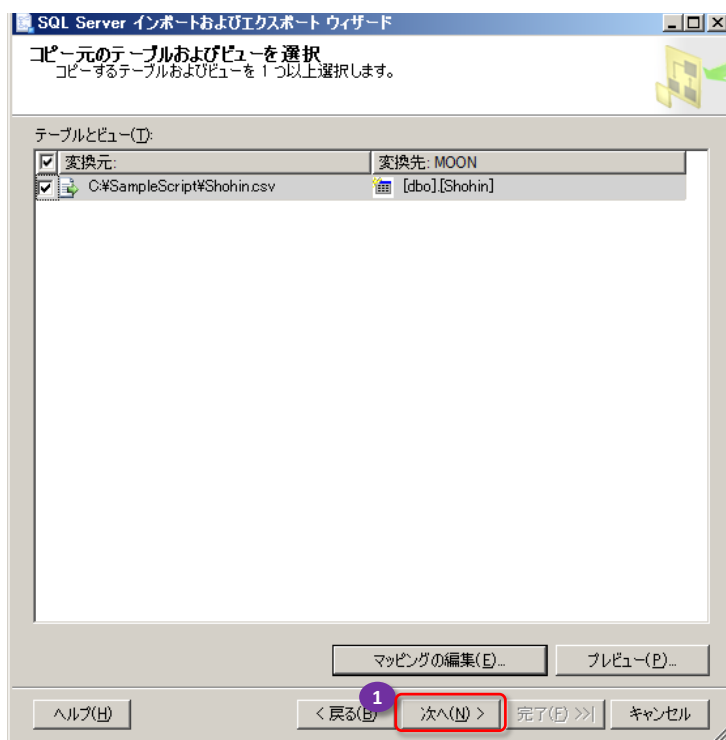
続いて、[マッピングの編集] ボタンをクリックします。

9. これにより、次のように[列マッピング] ダイアログが表示されて、作成されるテーブルの列名やデータ型などを変更できるようになります。

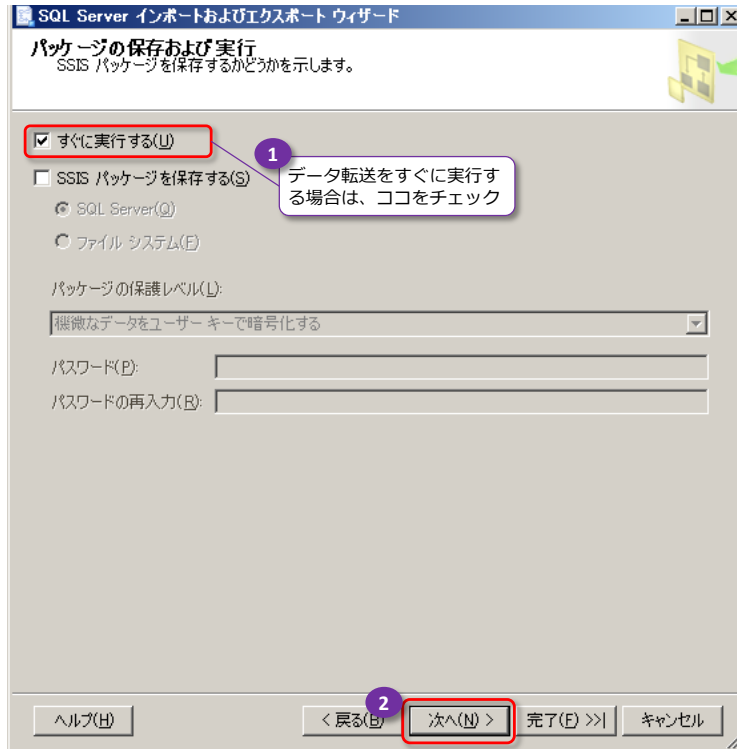


既定では、テキスト ファイルのデータは、**varchar** データ型として設定され、列名は、左側から順に「**列 0**」、「**列 1**」、「**列 2**」と設定されます。今回は、変換先の列名を分かりやすくするために、上の画面のように「**列 0**」を「**商品番号**」、「**列 1**」を「**商品名**」、「**列 2**」を「**単価**」へ変更しておきます。変更後、[OK] ボタンをクリックします。

10. [コピー元のテーブルおよびビューを選択] ページへ戻ったら、[次へ] ボタンをクリックします。

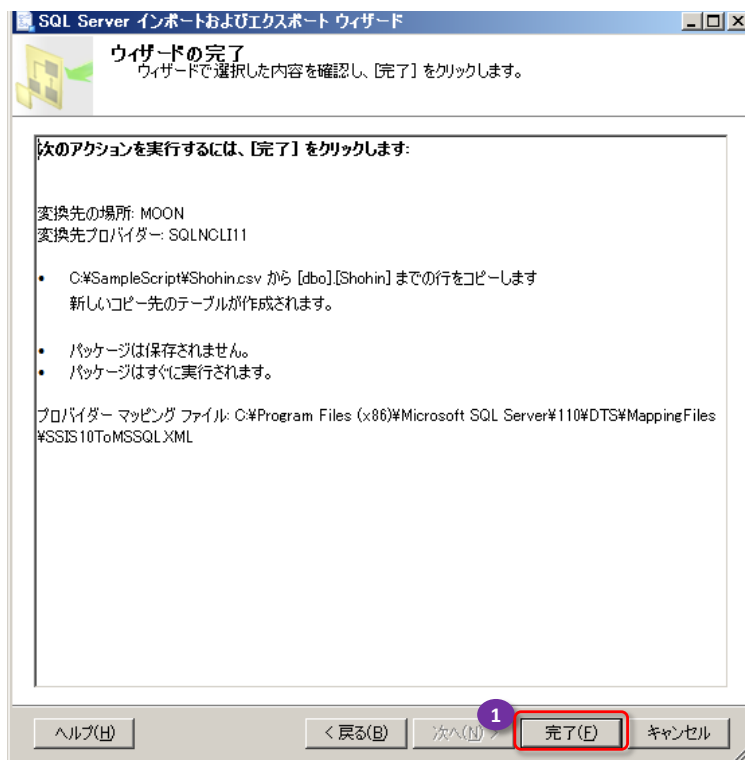


11. 次の[パッケージの保存および実行] ページでは、このデータ転送をすぐに実行するか、後から実行するために保存するかを選択します。



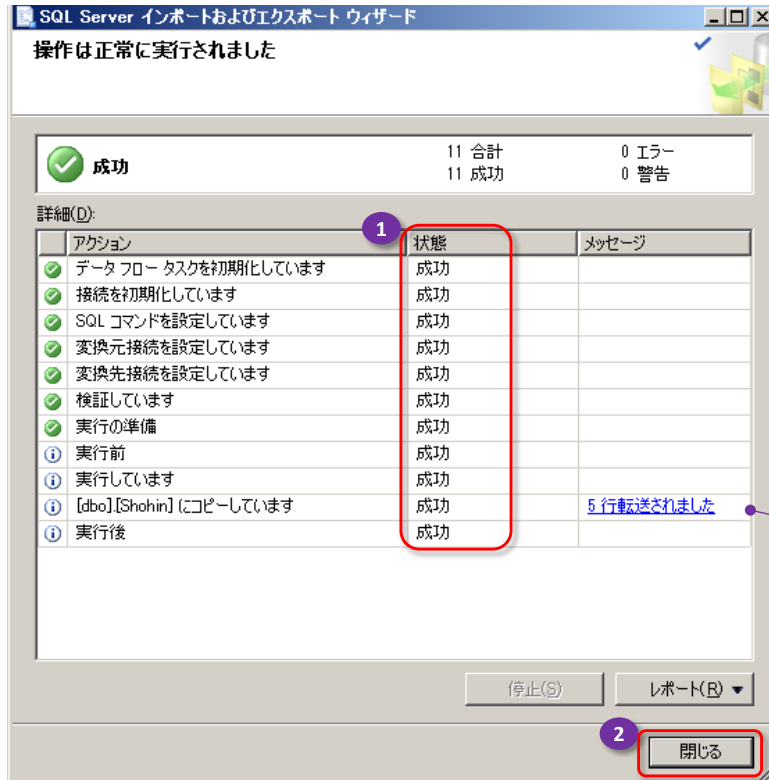
今回は、すぐにデータ転送を実行するので「**すぐに実行する**」が選択されていることを確認して、「**次へ**」ボタンをクリックします。

12. 次の「**ウィザードの完了**」ページでは、「**完了**」ボタンをクリックすると、データのインポートが開始されます。



13. 実行が完了すると、次のページが表示されます。

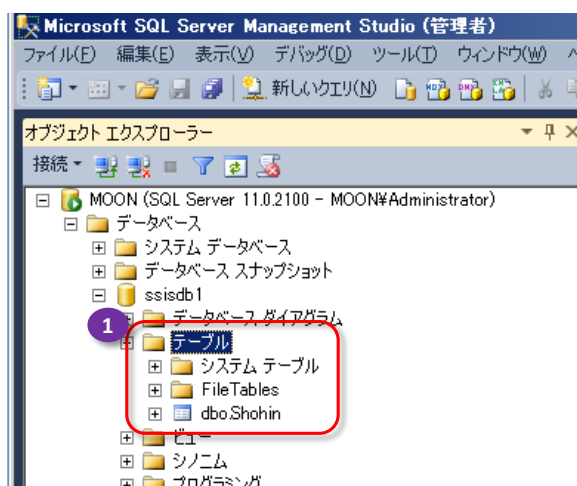




[状態] がすべて「成功」と表示されていることを確認して、[閉じる] ボタンをクリックします。

## ➡ 取り込んだデータの確認

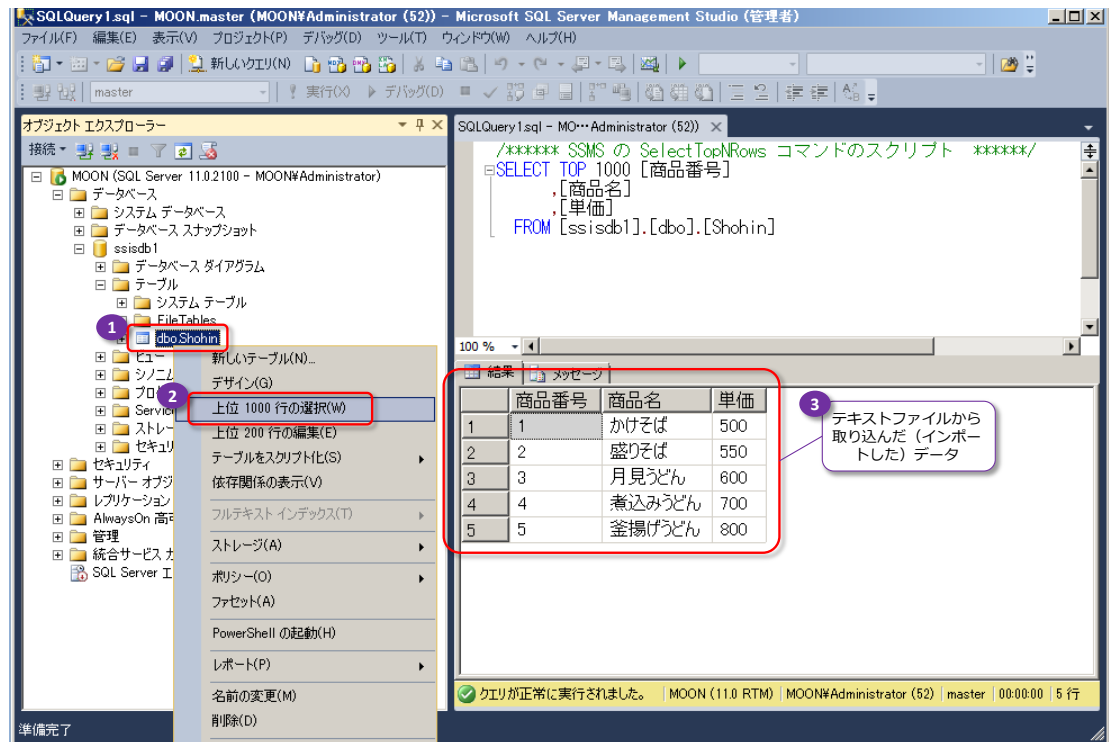
14. ウィザードでインポートしたデータを確認するには、次のように **Management Studio** で [ssisdb1] データベースの [テーブル] フォルダを展開します。



テーブルの一覧に、「Shohin」という名前のテーブルが作成されていることを確認できます(表示されない場合は、[テーブル] フォルダを右クリックして、[最新の情報に更新] をクリックします)。

15. 続いて、次のように [Shohin] テーブルを右クリックして [上位 1000 行の選択] をクリッ

くします。



これにより、**Shohin** テーブルのデータ (最初の 1000 件) が表示されます。表示されたデータが、取り込んだテキスト ファイル内のデータ (以下) と同じであることを確認します。

#### Shohin.csv ファイルのデータ

Shohin.csv - メモ帳	
ファイル(F)	編集(E) 書式(O) 表示(V) ヘルプ(H)
1,かけそば,500	
2,盛りそば,550	
3,月見うどん,600	
4,煮込みうどん,700	
5,釜揚げうどん,800	

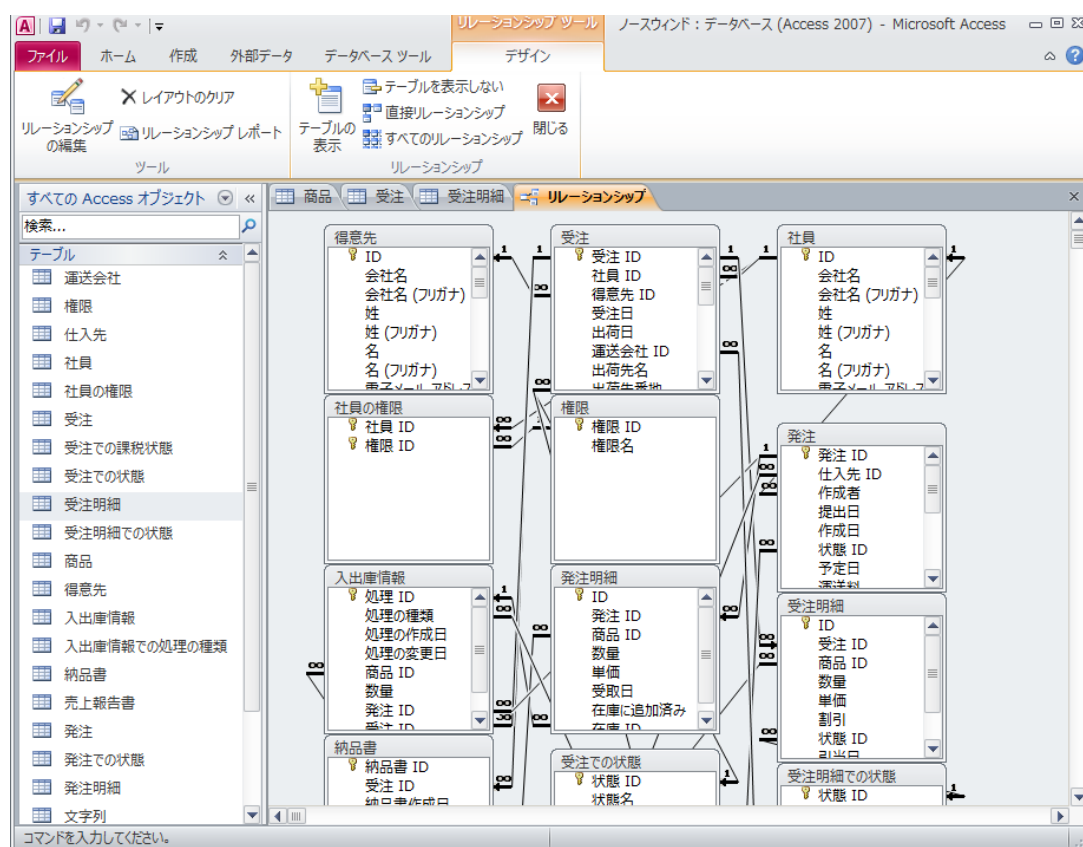
## 2.3 Access データベース (.accdb/.mdb) のインポート

### ➡ Access データベース (.accdb/.mdb) のインポート

Integration Services を利用すると、Access データベース (.accdb/.mdb ファイル) をインポートすることも簡単にできます。

### ➡ Let's Try

それでは、これを試してみましょう。ここでは、Microsoft Access 2010 に付属の「ノースウインド」サンプル データベースを SQL Server へインポートしてみましょう。このデータベースは、販売管理を題材としたサンプル データベースで、次のように商品に関する受注情報が格納されています。



以降の手順を試すには、このデータベース（ノースウインド.accdb）が必要になりますが、ほかの .accdb ファイルや Access 2003 以前のデータベース (.mdb ファイル) でもほとんど同じように試せるので、ノースウインド データベースがない場合は、任意の .accdb/.mdb ファイルで試してみてください。

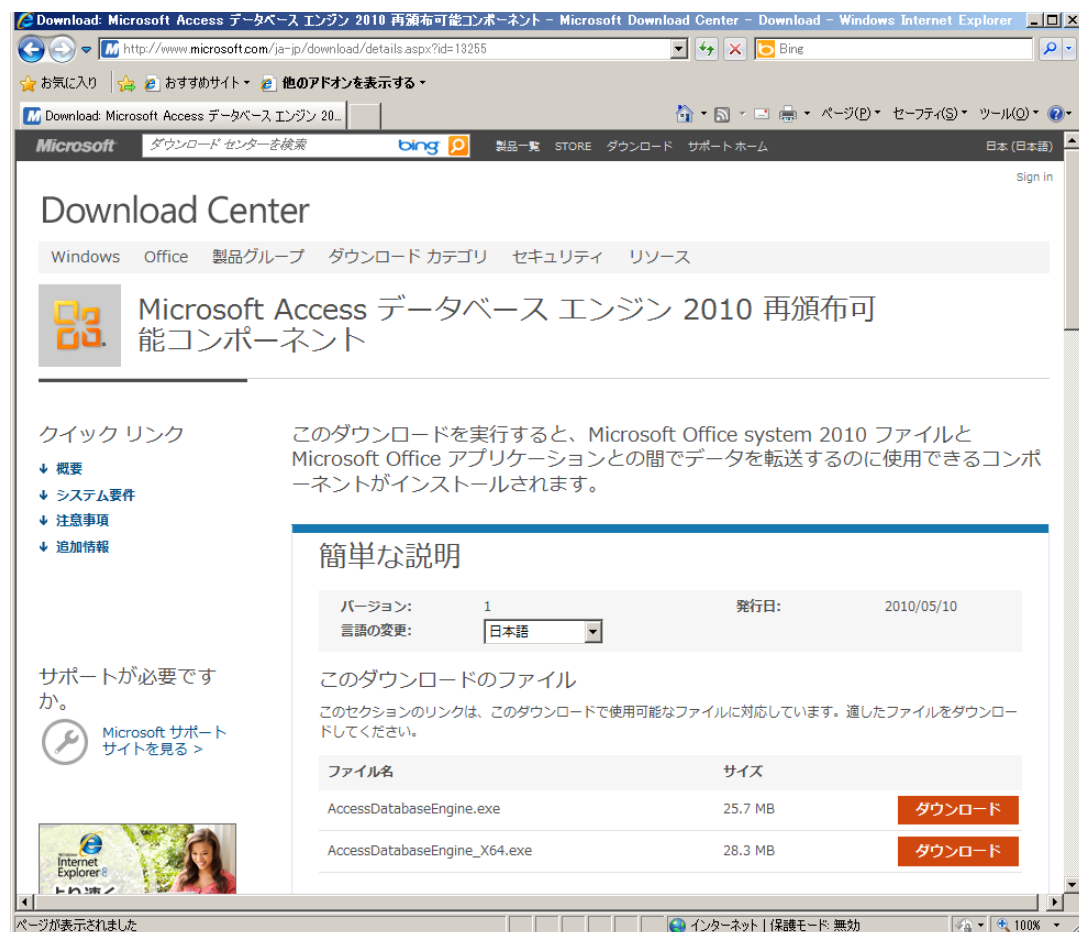
### ➡ Microsoft Access データベース エンジン 2010 再頒布可能コンポーネント

Integration Services を使って Access 2007/2010 のデータベース「.accdb」をインポート

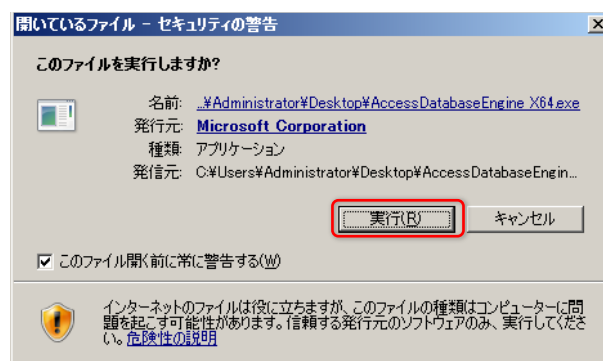
するには、ローカル マシン（インポート/エクスポート ウィザードを実行するマシン）に **Office 2010** をインストールしておくか、インストールしていない場合は、下記サイトから「**Access データベース エンジン 2010 再頒布可能コンポーネント**」をダウンロードして、インストールしておく必要があります。

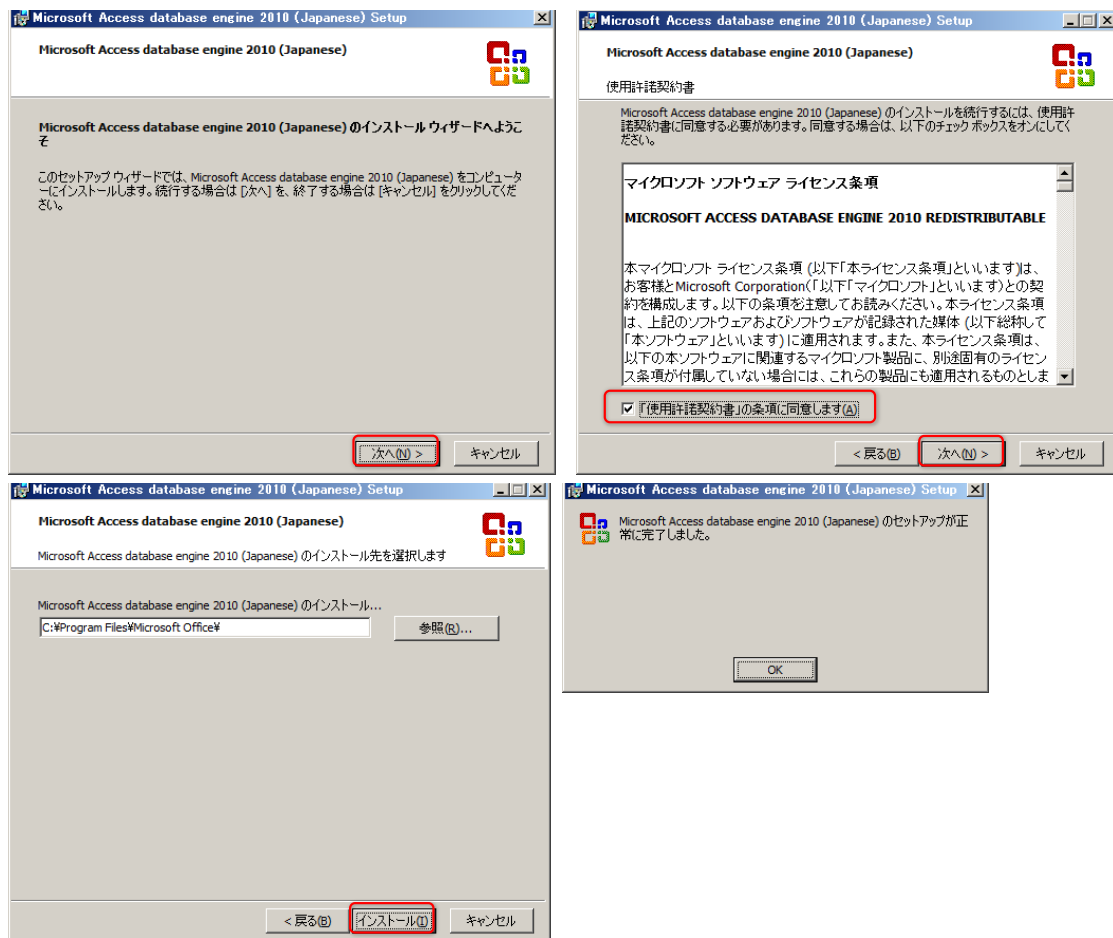
## Access データベース エンジン 2010 再頒布可能コンポーネント

<http://www.microsoft.com/ja-jp/download/details.aspx?id=13255>



32-bit 環境の場合は「**AccessDatabaseEngine.exe**」、X64（64-bit）環境の場合は「**AccessDatabaseEngine\_X64.exe**」ファイルをダウンロードしてインストールしておきます。ダウンロードしたファイルをダブル クリックすれば、次のようにインストーラーが起動して、インストールすることができます。

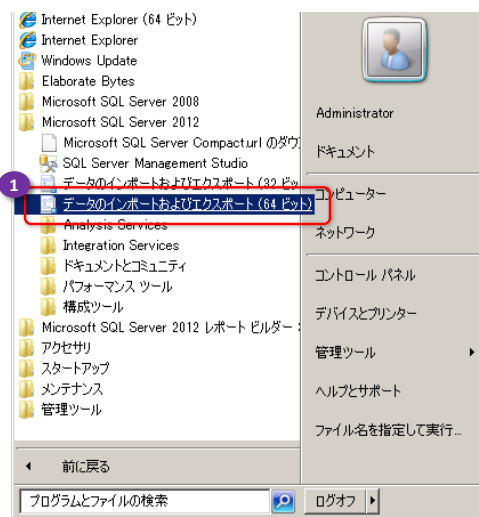




なお、Access 2003 以前のデータベース「.mdb」をインポートする場合には、上記のコンポーネントは不要です（Access 2007 以降のデータベース「.accdb」をインポートする場合に必要）。

## ➡ インポート／エクスポート ウィザードの起動

1. まずは、次のように [スタート] メニューの [Microsoft SQL Server 2012] から [データのインポートおよびエクスポート (32 ビット) または (64 ビット)] をクリックして、ウィザードを起動します（X64 環境の場合は、(64 ビット) メニューをクリックします）。



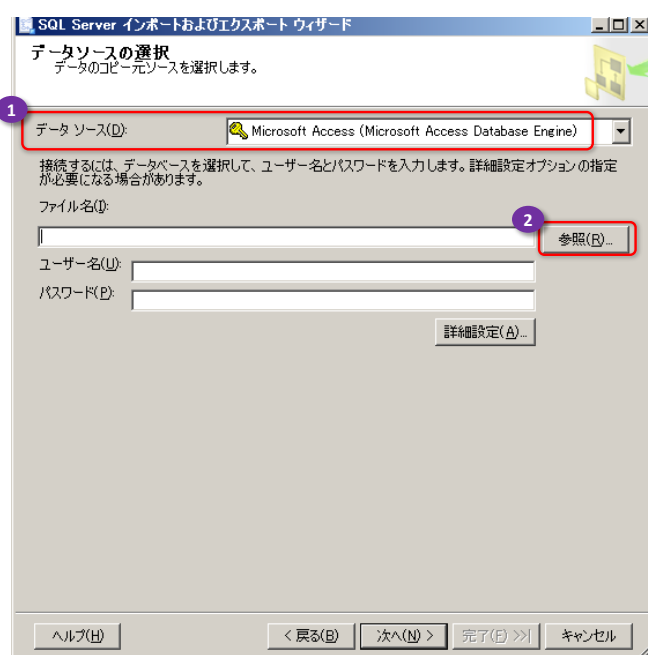
## Note : 32 ビット版と 64 ビット版の違い

Step 2.2 でテキスト ファイルを取り込んだときは、Management Studio からデータベースを右クリックして、**[タスク]** メニューの **[データのインポート]** からインポート/エクスポート ウィザードを起動しましたが、このように起動した場合は、32 ビット版のウィザードが起動します。64 ビット (X64) 環境の場合は、上の手順のように、**[スタート]** メニューに 32 ビット版と 64 ビット版の 2 種類のウィザードが登録されていて、**(64 ビット)** メニューを選択することで、64 ビット版のウィザードを起動することができます。Access 2010 の再頒布コンポーネント (64 ビット版) を利用するには、64 ビット版のウィザードを起動する必要があります。

2. インポート/エクスポート ウィザードが起動したら、**[次へ]** ボタンをクリックします。

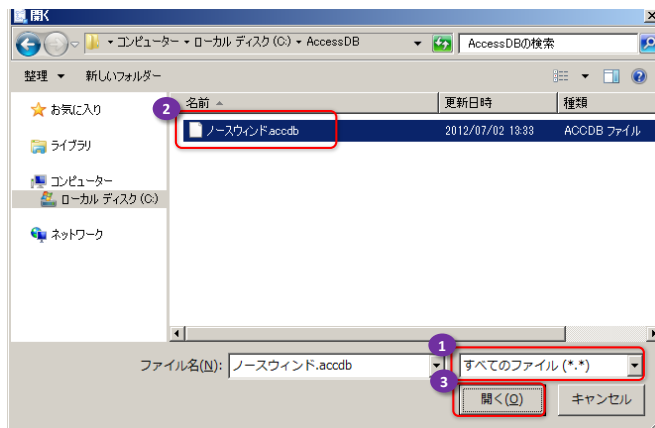


3. 次の **[データ ソースの選択]** ページでは、**[データ ソース]** (コピー元となるデータベース) に「**Microsoft Access (Microsoft Access Database Engine)**」を選択します。



X64 環境の場合に、**(64 ビット)** メニューをクリックせずに、**(32 ビット)** メニューをクリックしてウィザードを起動した場合は、このデータソースが表示されないので注意してください。データ ソースの選択後、**[ファイル名]** の **[参照]** ボタンをクリックします。

4. **[ファイルを開く]** ダイアログが表示されたら、**[すべてのファイル (\*.\*)]** を選択して、**ノースウィンド サンプル データベース (ノースウィンド.accdb)** を選択し、**[開く]** ボタンをクリックします。

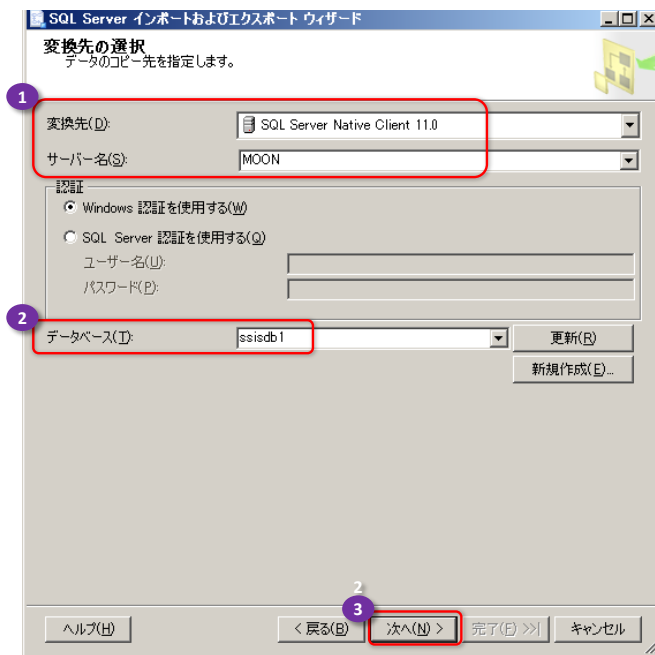


**[データ ソースの選択]** ページへ戻ったら、**[次へ]** ボタンをクリックします。

**Note : Access 2003 データベース (.mdb) をインポートしたい場合**

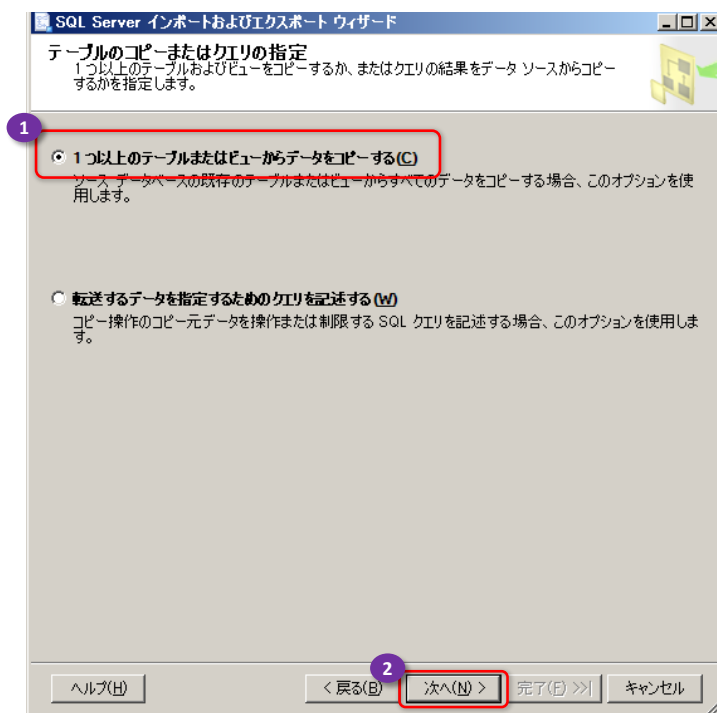
Access 2003 以前のデータベース (.mdb ファイル形式のデータベース) をインポートしたい場合にも、**[データソース]** に **「Microsoft Access (Microsoft Access Database Engine)」** を選択することが可能です。あるいは、**[データソース]** で **「Microsoft Access (Microsoft Jet Database Engine)」** (Jet エンジン) を選択してもインポートを行うことが可能です。ただし、Jet は 32 ビット版のみしか提供されていないので、X64 環境の場合は、**(32 ビット)** のメニューからインポート/エクスポート ウィザードを起動する必要があります。

5. 次の **[変換先の選択]** ページでは、**[変換先]** に **「SQL Server Native Client 11.0」** を選択して、**[サーバー名]** に SQL Server の名前 (画面は **MOON**) を入力します。



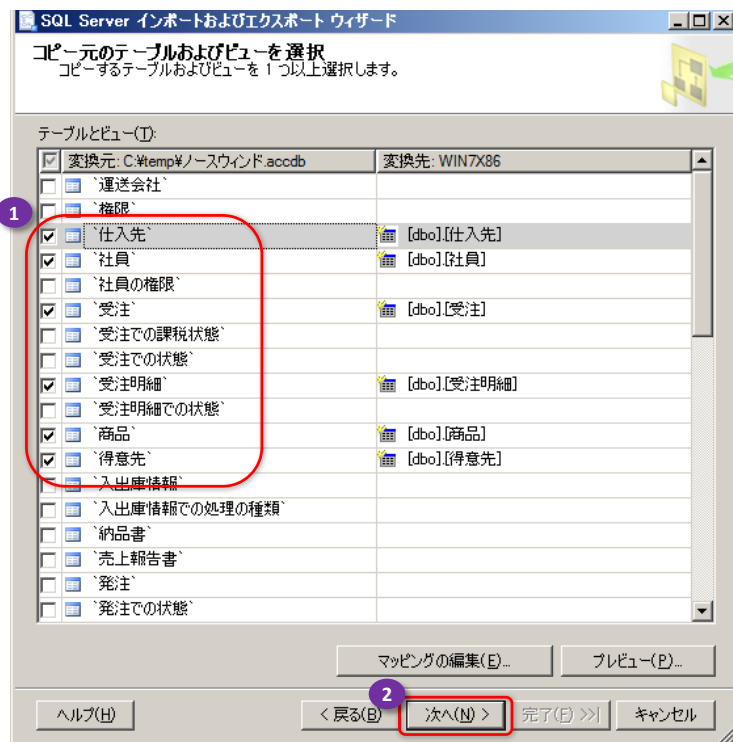
「データベース」で「ssisdb1」を選択して、[次へ] ボタンをクリックします。

6. 次の「テーブルのコピーまたはクエリの指定」ページでは、[1 つ以上のテーブルまたはビューからデータをコピーする] を選択して、[次へ] ボタンをクリックします。



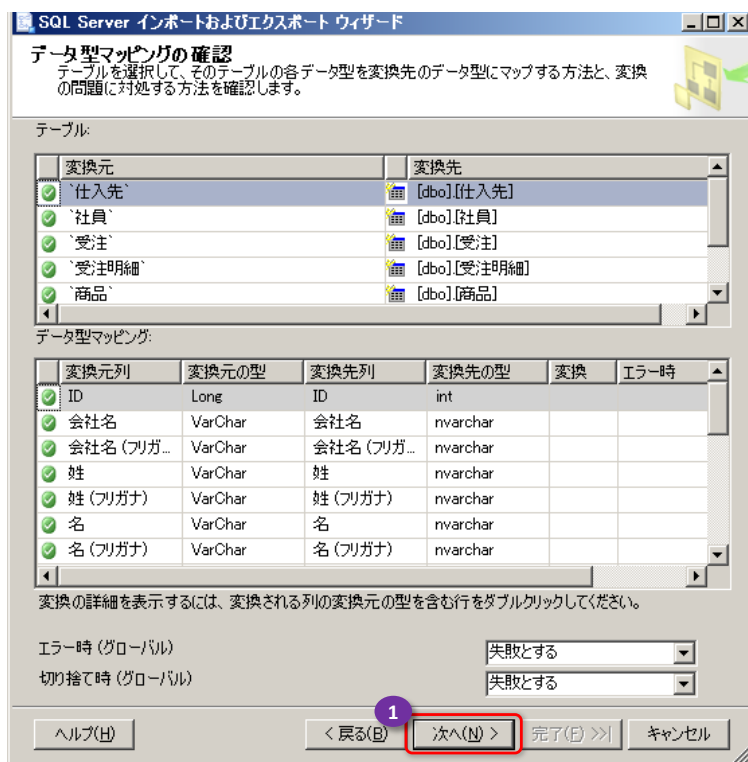
7. 次の「コピー元のテーブルおよびビューを選択」ページでは、Access データベース内のテーブルが一覧されます。





ここでは、「仕入先」や「社員」、「受注」、「受注明細」、「商品」、「得意先」など、任意のテーブルをチェックして、インポート対象に設定します。設定後、[次へ] ボタンをクリックします。

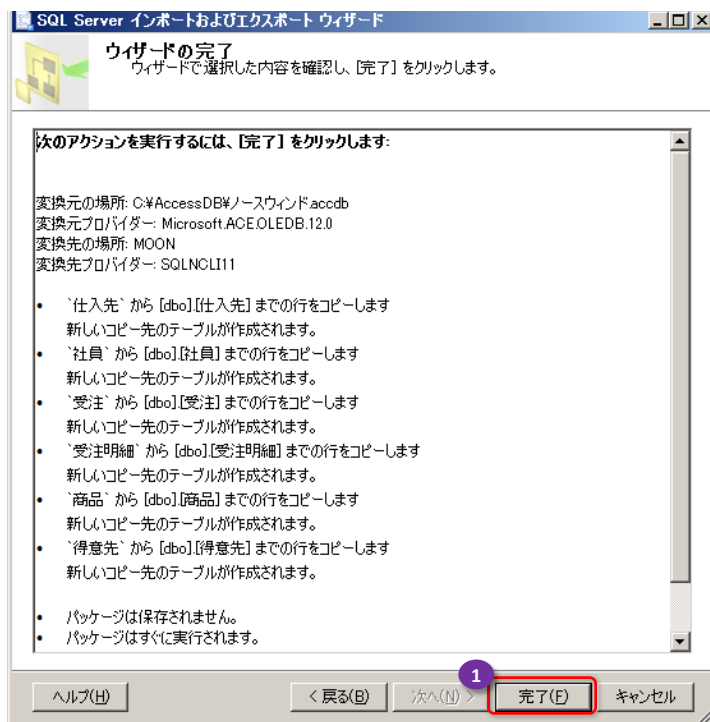
8. 次の「データ型マッピングの確認」ページでは、テーブル内の各列のマッピング（変換されるデータ型）を確認して、[次へ] ボタンをクリックします。



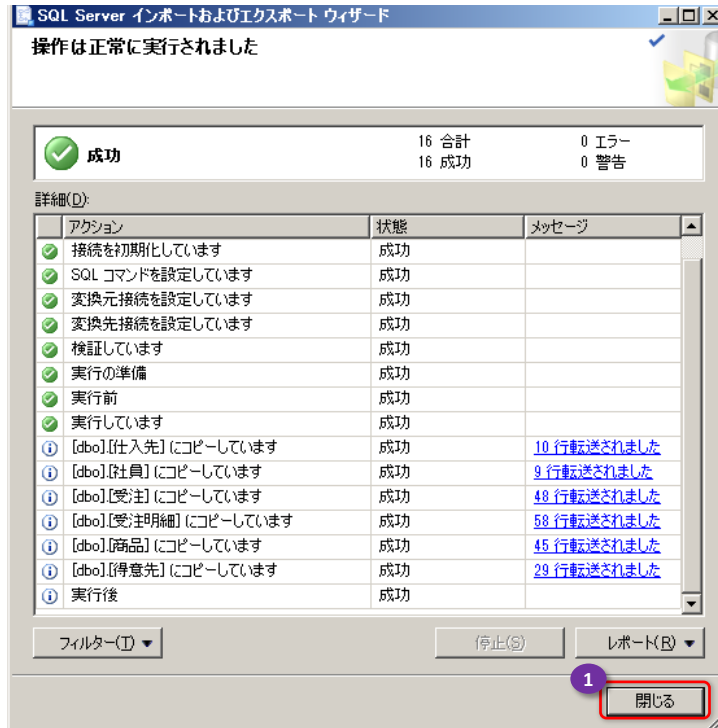
9. 次の「パッケージの保存および実行」ページでは、「すぐに実行する」を選択して、[次へ] ボタンをクリックします。



10. 最後の「ウィザードの完了」ページでは、[完了] ボタンをクリックすると、データのコピーが開始されます。



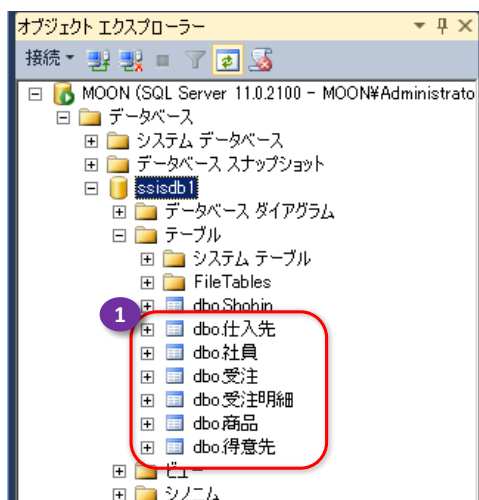
11. インポートが完了すると、次のように「操作は正常に実行されました」ページが表示されます。  
[状態] が「成功」と表示されていることを確認して、[閉じる] ボタンをクリックします。



以上で、Access データのインポートが完了です。

## ➡ インポートしたデータの確認

12. インポートしたデータを確認するには、**Management Studio** のオブジェクト エクスプローラーで「**ssisdb1**」データベースを展開して、[テーブル] フォルダを右クリックし、[最新の情報に更新] をクリックします。



インポートしたテーブル（仕入先や社員、受注、受注明細、商品、得意先など）が表示されることを確認できます。

13. 次に、「社員」テーブルや「商品」テーブルを右クリックして、[上位 1000 行の選択] をクリ

ックし、テーブルの中身を確認しておきましょう。

### 「社員」テーブルのデータの確認

SQLQuery2.sql - MOON.master (MOON Administrator (55)) - Microsoft SQL Server Management Studio (管理者)

オブジェクト エクスプローラー

MOON (SQL Server 11.0.2100 - MOON Administrator)

データベース

システム データベース

データベース スナップショット

ssisdb1

データベース ダイアグラム

テーブル

システム テーブル

FileTables

dbo.Shohin

dbo.仕入先

dbo.社員

新しいテーブル(N)...

デザイン(G)

上位 1000 行の選択(W)

上位 200 行の編集(E)

テーブルをスクリプト化(S)

依存関係の表示(V)

フルテキスト インデックス(T)

ストレージ(A)

ポリシー(O)

ファセット(A)

PowerShell の起動(H)

レポート(P)

名前の変更(M)

削除(D)

最新の情報に更新(F)

プロパティ(R)

SQLQuery2.sql - MO Administrator (55) x SQLQuery1.sql - MO Administrator (52)

\*\*\*\*\* SSMS の SelectTopRows コマンドのスクリプト \*\*\*\*\*

```
SELECT TOP 1000 [ID]  
,[会社名]  
,[会社名 (フリガナ)]  
,[姓]  
,[姓 (フリガナ)]  
,[名]  
,[名 (フリガナ)]  
,[電子メール アドレス]  
,[部署]  
,[会社電話番号]  
,[自宅電話番号]  
,[携帯電話番号]  
,[FAX 番号]  
,[番地]  
,[市区町村]
```

ID	会社名	会社名(フリガナ)	姓	姓(フリガナ)	名	名(フリガナ)
1	Northwind Traders	Northwind Traders	加藤	カトウ	泰江	ヤスエ
2	Northwind Traders	Northwind Traders	葛城	カツラギ	孝史	コウシ
3	Northwind Traders	Northwind Traders	川村	カフムラ	匡	タダシ
4	Northwind Traders	Northwind Traders	成宮	ナルミヤ	真紀	マキ
5	Northwind Traders	Northwind Traders	青木	アオキ	俊之	トシユキ
6	Northwind Traders	Northwind Traders	山本	ヤマモト	雅治	マサハル
7	Northwind Traders	Northwind Traders	松沢	マツザワ	誠一	セイイチ
8	Northwind Traders	Northwind Traders	小川	オガワ	さよ子	サヨコ
9	Northwind Traders	Northwind Traders	森上	モリウエ	偉久馬	イクマ

### 「商品」テーブルのデータの確認

SQLQuery3.sql - MOON.master (MOON Administrator (54)) - Microsoft SQL Server Management Studio (管理者)

オブジェクト エクスプローラー

MOON (SQL Server 11.0.2100 - MOON Administrator)

データベース

システム データベース

データベース スナップショット

ssisdb1

データベース ダイアグラム

テーブル

システム テーブル

FileTables

dbo.Shohin

dbo.仕入先

dbo.社員

dbo.受注

dbo.受注明細

dbo.商品

新しいテーブル(N)...

デザイン(G)

上位 1000 行の選択(W)

上位 200 行の編集(E)

テーブルをスクリプト化(S)

依存関係の表示(V)

フルテキスト インデックス(T)

ストレージ(A)

ポリシー(O)

ファセット(A)

PowerShell の起動(H)

レポート(P)

名前の変更(M)

削除(D)

最新の情報に更新(F)

プロパティ(R)

SQLQuery3.sql - MO Administrator (54) x SQLQuery2.sql - MO Administrator (55)

\*\*\*\*\* SSMS の SelectTopRows コマンドのスクリプト \*\*\*\*\*

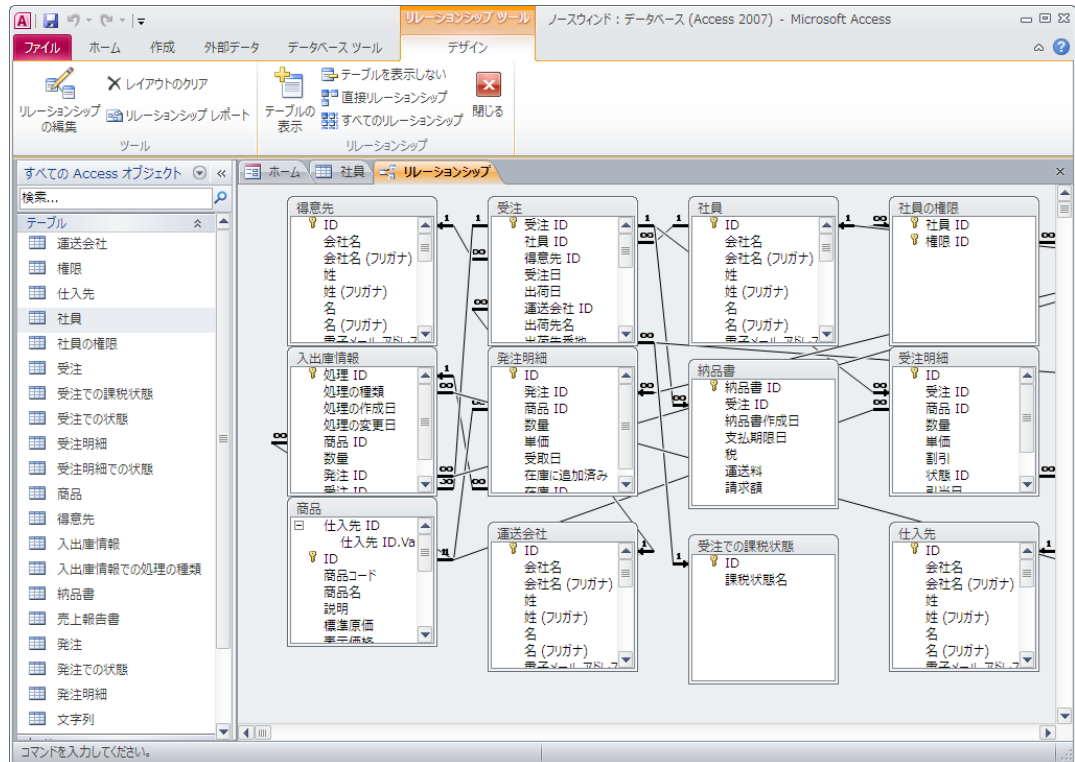
```
SELECT TOP 1000 [仕入先 ID]  
,[ID]  
,[商品コード]  
,[商品名]  
,[説明]  
,[標準原価]  
,[表示価格]  
,[発注点]  
,[目録点]  
,[梱包単位]  
,[生産中止]  
,[最少の再発注数量]  
,[商品区分]  
,[添付ファイル]  
FROM [ssisdb1].[dbo].[商品]
```

仕入先 ID	ID	商品コード	商品名	説明
4	1	NWTB-1	Northwind Traders チャイ	NULL
10	3	NWTCO-3	Northwind Traders シロップ	NULL
10	4	NWTCO-4	Northwind Traders ビリカラタバスコ	NULL
10	5	NWTO-5	Northwind Traders オリーブオイル	NULL
2,6	6	NWTJP-6	Northwind Traders ボイゼンバリージャム	NULL
2	7	NWTFN-7	Northwind Traders 乾燥ナシ	NULL
8	8	NWTS-8	Northwind Traders カレーソース	NULL
2,6	14	NWTFN-14	Northwind Traders ウォールナッツ	NULL
6	17	NWTCFV-17	Northwind Traders フルーツカクテル	NULL
1	19	NWTBGM-19	Northwind Traders チョコレートビスケットミックス	NULL

クエリが正常に実行されました。 MOON (11.0 RTM) | MOON Administrator (54) | master | 00:00:00 | 45 行

## Note : Access 側で設定された主キーやリレーションシップは転送されない

Integration Services では、Access データベース側で設定された主キーやリレーションシップなどの制約はコピーされません。Access 側で設定されていたリレーションシップは、次のとおりです。



Integration Services で取り込んだデータを Management Studio で確認すると、次のようになります。

社員テーブルの列の一覧。  
主キーと外部キーには  
鍵のアイコンが付くが、  
設定されていない

主キーと外部キーが一覧される場所。  
ここに存在しないことから主キー  
と外部キーが設定されていないこと  
を確認できる

制約が一覧される場所  
(ココも空)

【キー】フォルダーや【制約】フォルダーを参照しても、制約が作成されていないことから、制約が転送されていないことを確認できます。したがって、主キーやリレーションシップなどは、データのインポート後に手動で設定する必要があります。

## 2.4 Excel データ (.xlsx/.xls ファイル) の取り込み

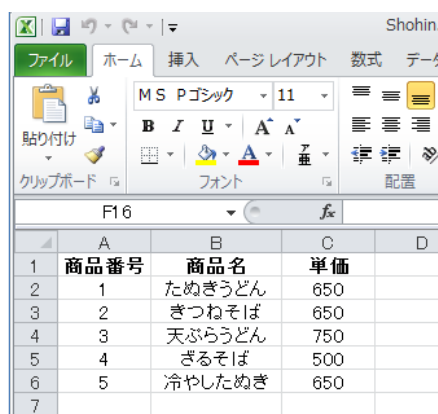
### ➡ Excel データ (.xlsx/.xls ファイル) の取り込み

次に、Excel 2010 のデータ (.xlsx/.xls ファイル) を取り込んでみましょう。Excel データをインポートする手順は、Access データベースをインポートする手順とほとんど同じで、違いは「**データソースの選択**」ページだけです。

また、Access データベースをインポートするときと同様、インポート/エクスポート ウィザードを実行するマシンに **Office 2010** をインストールしておくか、インストールしていない場合は「**Access データベース エンジン 2010 再頒布可能コンポーネント**」をダウンロードして、インストールしておく必要があります (**Excel 2007** 以降の「.xlsx」ファイルのインポートに必須)。

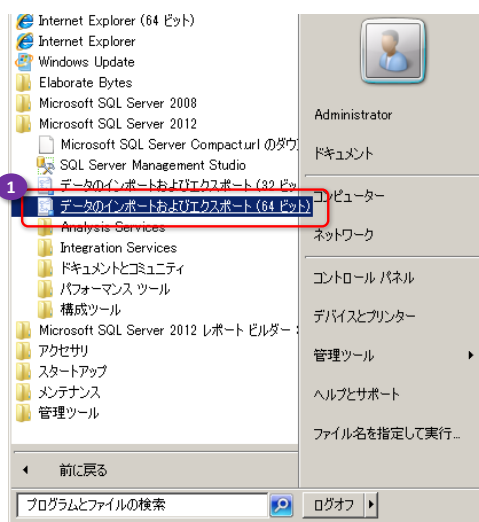
### ➡ Let's Try

それでは、これを試してみましょう。取り込むファイルは、サンプル スクリプト内の「**Shohin.xlsx**」という名前のファイル (以下) です。



	A	B	C	D
1	商品番号	商品名	単価	
2	1	たぬきうどん	650	
3	2	きつねそば	650	
4	3	天ぷらうどん	750	
5	4	ざるそば	500	
6	5	冷やしたぬき	650	
7				

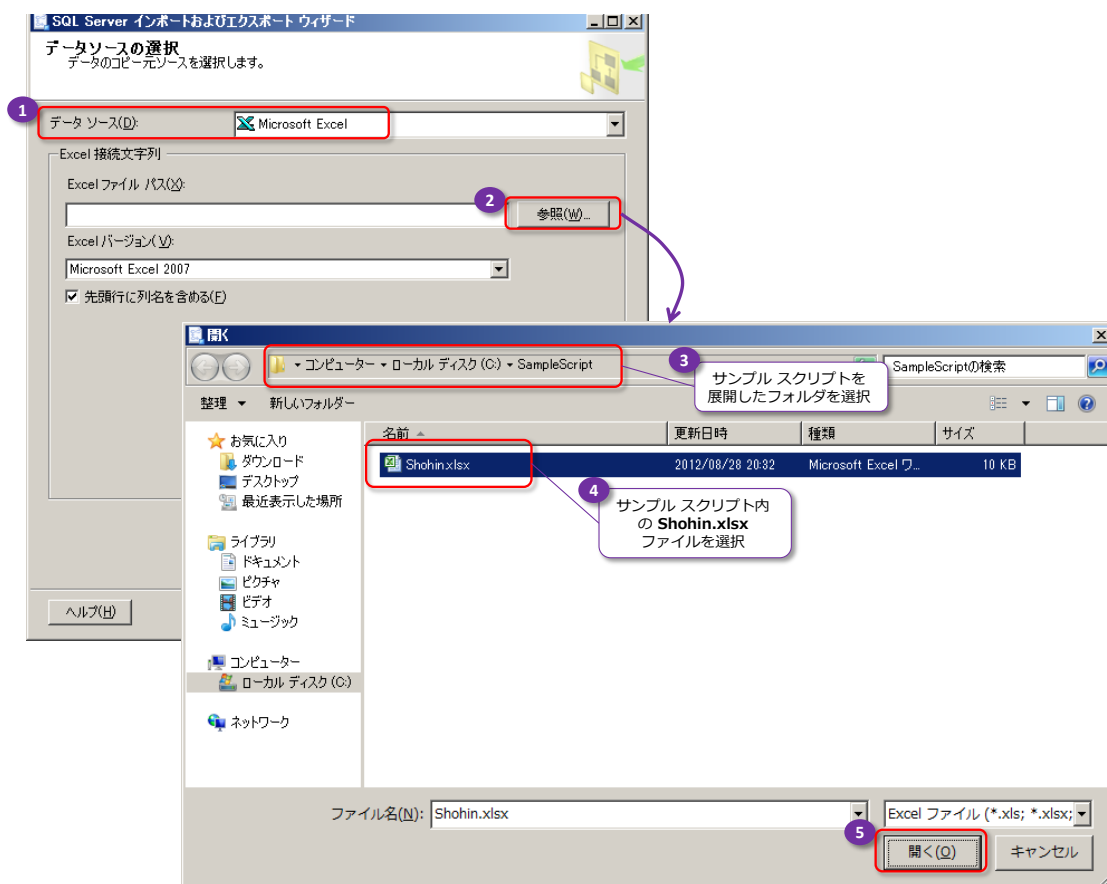
1. まずは、[スタート] メニューの [Microsoft SQL Server 2012] から [データのインポートおよびエクスポート (32 ビット) または (64 ビット)] をクリックして、インポート/エクスポート ウィザードを起動します (X64 環境の場合は、(64 ビット) メニューをクリック)。



2. インポート/エクスポート ウィザードが起動したら、[次へ] ボタンをクリックします。

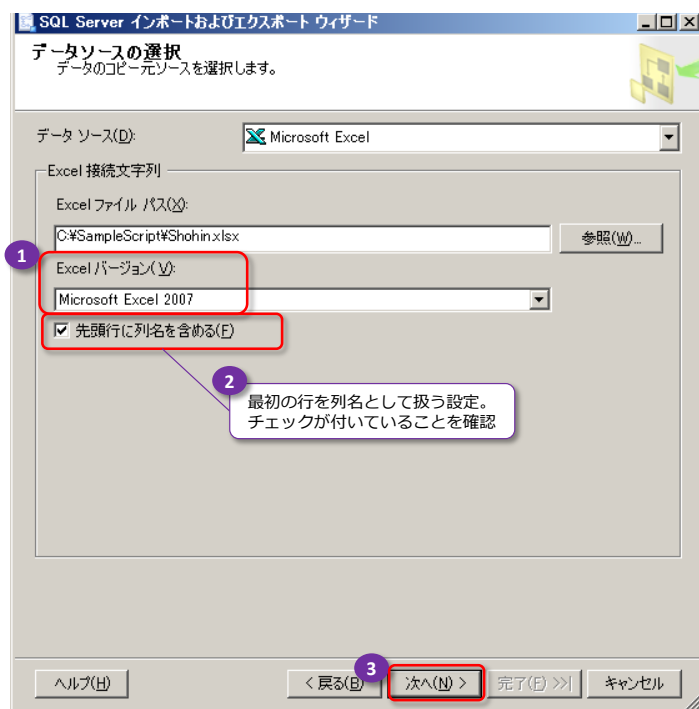


3. 次の [データ ソースの選択] ページでは、[データ ソース] で「Microsoft Excel」を選択して、[参照] ボタンをクリックします。

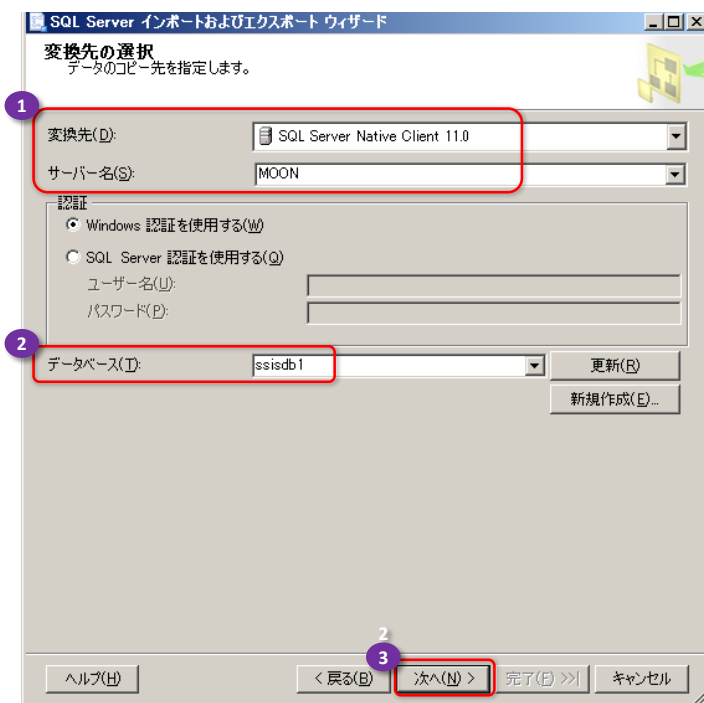


[開く] ダイアログが表示されたら、サンプル スクリプト内の「Shohin.xlsx」ファイルを選択して、[開く] ボタンをクリックします。

4. [データソースの選択]ページへ戻ったら、[Excel バージョン]で[Microsoft Excel 2007]を選択して、[先頭行に列名を含める]をチェックし、[次へ] ボタンをクリックします。



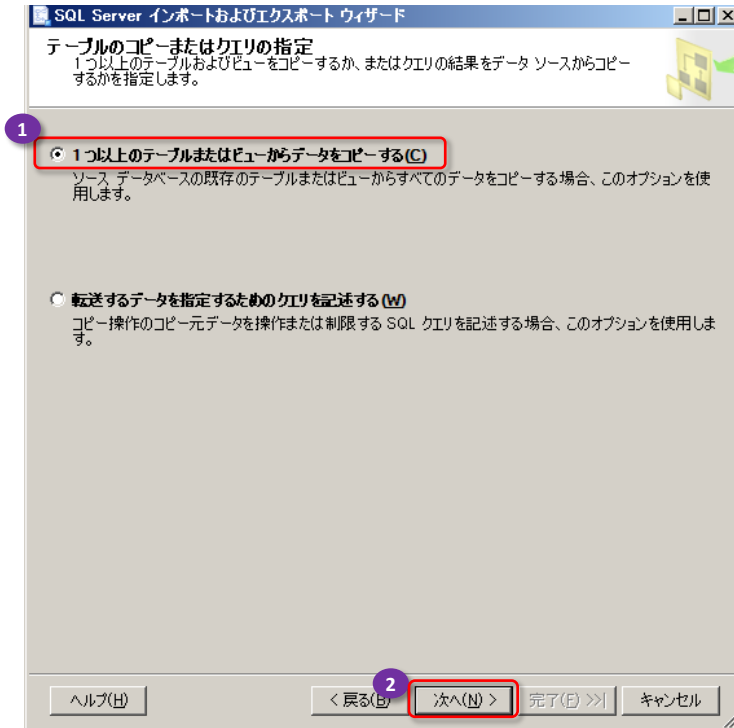
5. 次の[変換先の選択]ページでは、[変換先]に「SQL Server Native Client 11.0」を選択して、[サーバー名]に SQL Server の名前（画面は MOON）を入力します。



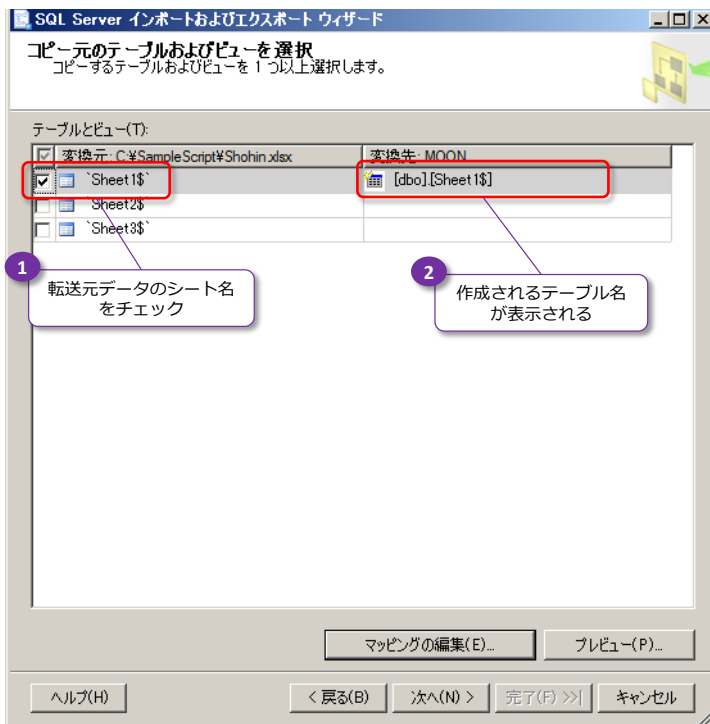
[データベース] で「ssisdb1」を選択して、[次へ] ボタンをクリックします。

6. 次の[テーブルのコピーまたはクエリの指定]ページでは、[1 つ以上のテーブルまたはビューからデータをコピーする]を選択して、[次へ] ボタンをクリックします。





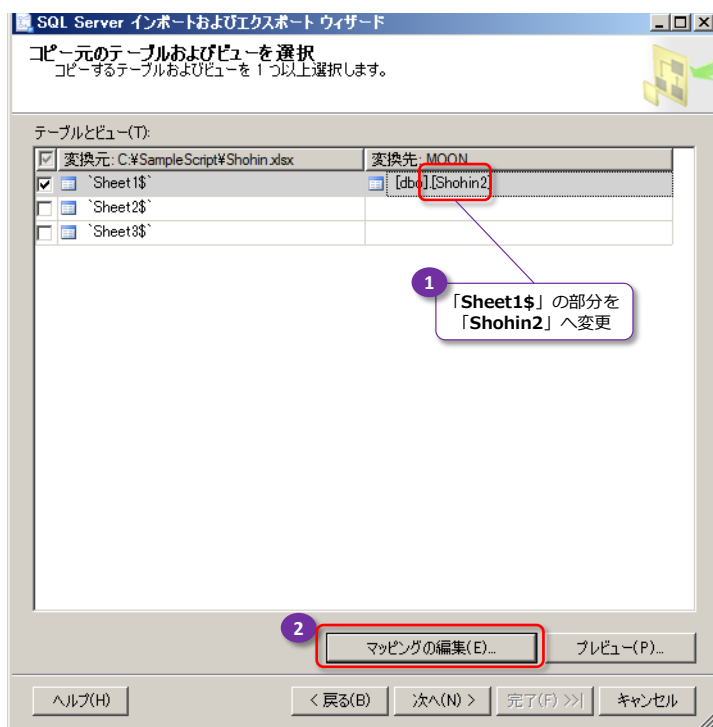
7. 次の「コピー元のテーブルおよびビューを選択」ページでは、Excel ファイル内のワークシートの名前に「\$」マークをつけたものが一覧されます。



今回のファイル (Shohin.xlsx) は、ワークシート名を変更していないので、既定の 3 つのワークシート名 (Sheet1、Sheet2、Sheet3) に「\$」マークを付けたものが一覧されています。今回取り込むデータは Sheet1 へ入れているので、「Sheet1\$」をチェックします。これにより、転送先のテーブル名 (変換先) が [dbo].[Sheet1\$] と表示されます。

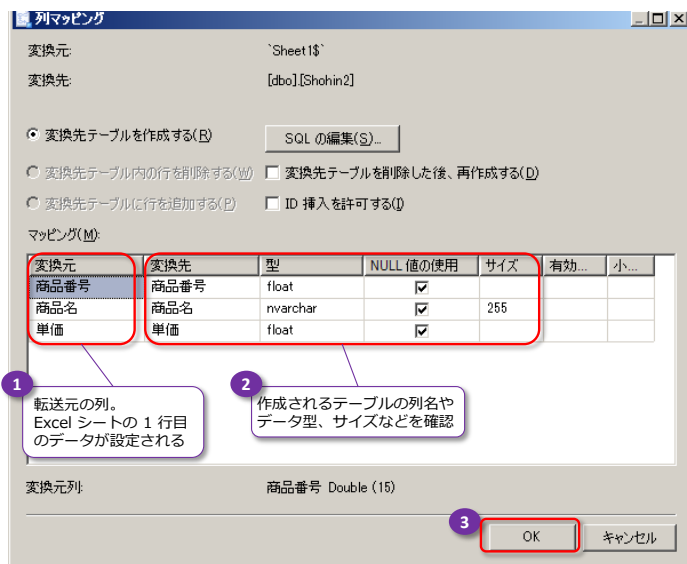
これは、「ssisdb1」というデータベースの中に「Sheet1\$」という名前のテーブルを作成す

るという意味です。これを、次のように「**Shohin2**」など任意の名前へと変更します。



続いて、[マッピングの編集] ボタンをクリックします。

8. [列マッピング] ダイアログが表示されたら、作成されるテーブルのデザインを確認します。

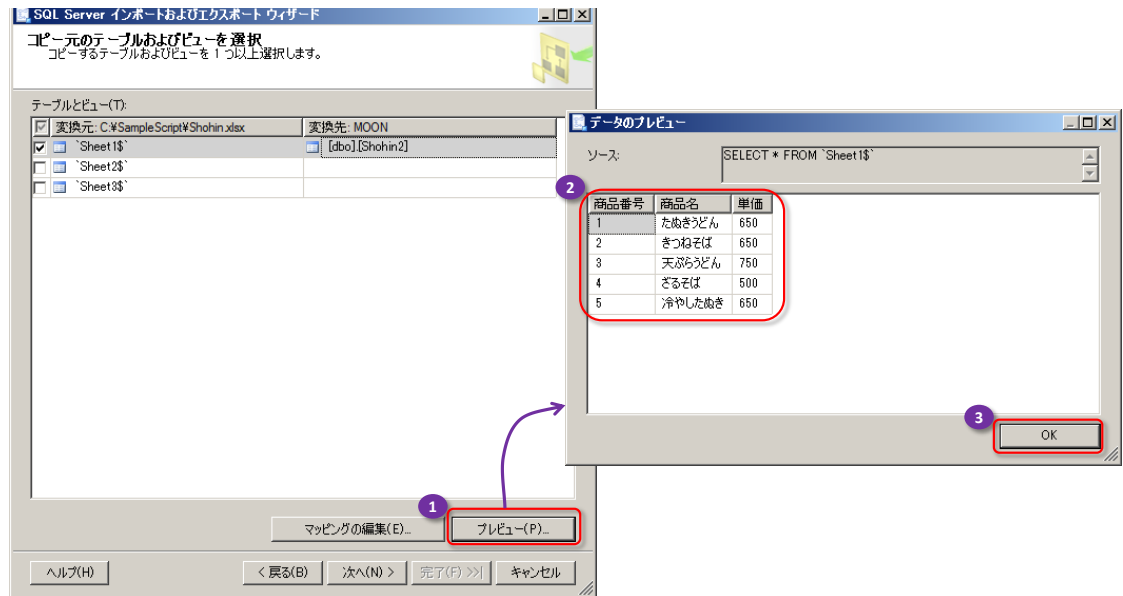


変換元（転送元）の列名には、Excel シートの 1 行目のデータ（商品番号、商品名、単価）が設定され、それをもとに変換先（転送先）に作成されるテーブルの列名が設定されていることを確認できます。

データ型は、既定では、Excel 内の数値データは float 型、文字データは nvarchar(255) 型として作成されます。列の名前やデータ型、サイズなどは、変更することもできますが、今回は何も変更せずに内容を確認するだけで、[OK] ボタンをクリックします。

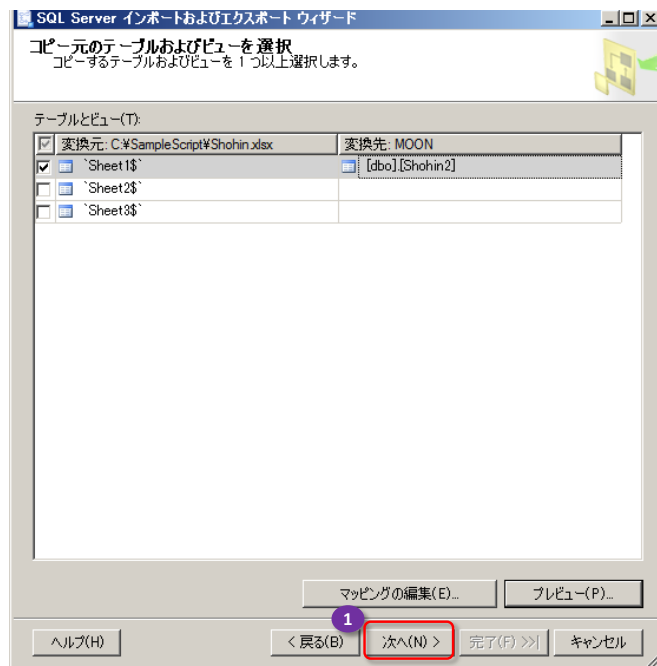
9. [コピー元のテーブルおよびビューを選択] ページへ戻ったら、[プレビュー] ボタンをクリック

クします。

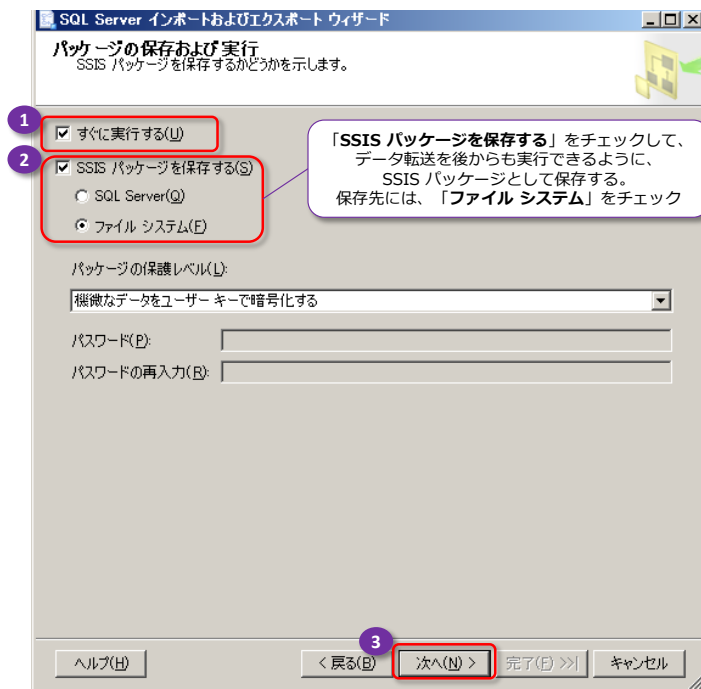


これにより、[データのプレビュー] ダイアログが開いて、転送元となる Excel データの内容を確認することができます。確認後、[OK] ボタンをクリックします。

10. [コピー元のテーブルおよびビューを選択] ページに戻ったら、[次へ] ボタンをクリックします。

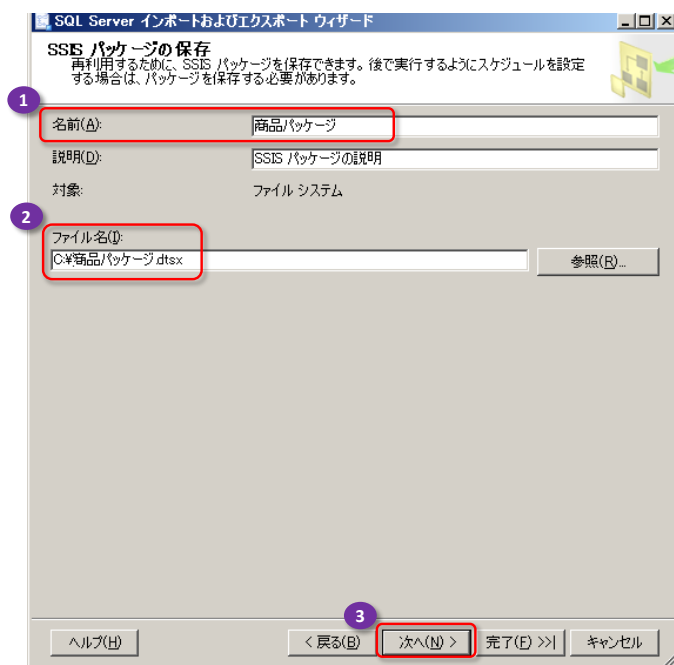


11. 次の [パッケージの保存および実行] ページでは、次のように [すぐに実行する] と [SSIS パッケージを保存する] の両方をチェックして、データ転送をすぐに実行し、かつ後からも実行できるように "SSIS パッケージ" として保存しておくようにします。

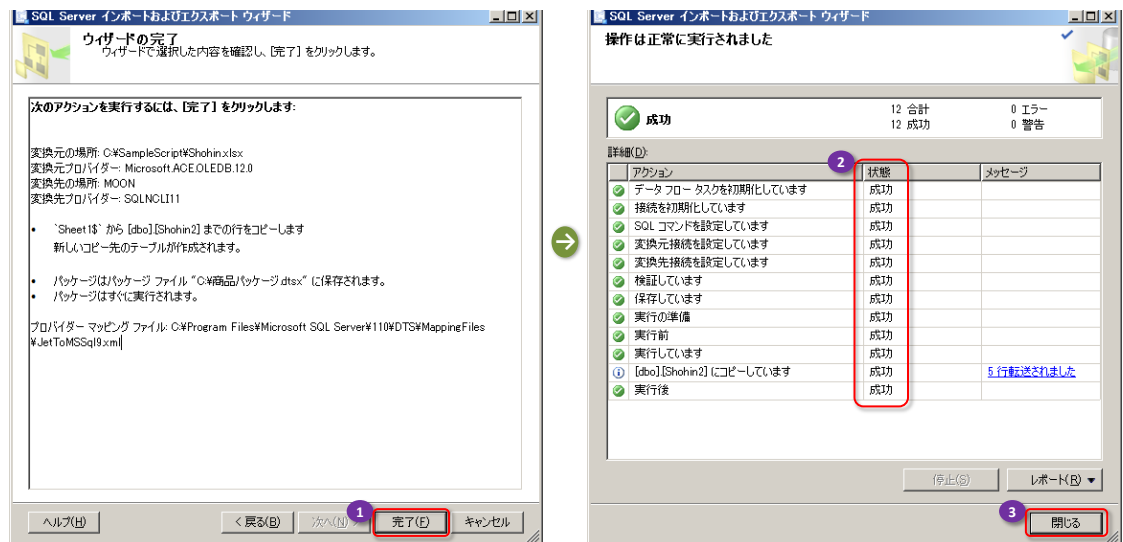


パッケージの保存先として[ファイル システム]を選択し、[次へ] ボタンをクリックします。

12. 次の[SSIS パッケージの保存] ページでは、保存するパッケージの名前と保存場所を指定します。今回は、次のように[名前] (パッケージ名) へ「商品パッケージ」と入力し、[ファイル名] のパスを「C:¥商品パッケージ.dtsx」へ変更し、C:¥ ドライブの直下へ保存されるようにします。変更後、[次へ] ボタンをクリックして、次へ進みます。



13. 最後の[ウィザードの完了] ページでは、[完了] ボタンをクリックすると、データのコピーが開始されます。

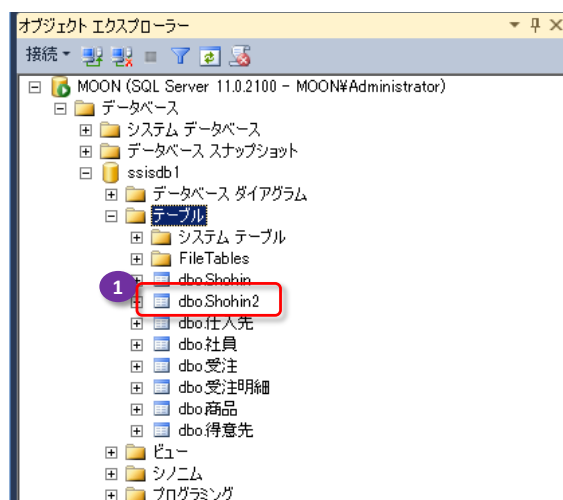


実行完了後、[状態] がすべて「成功」と表示されていることを確認して、[閉じる] ボタンをクリックします。

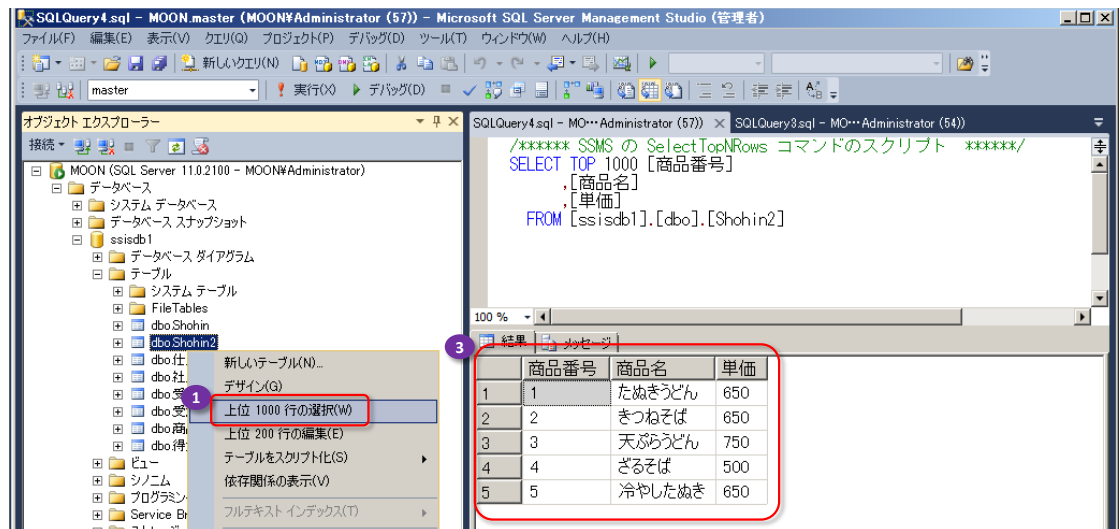
以上で、Excel データのインポートが完了です。

## ➡ 取り込んだデータの確認

- インポートしたデータを確認するには、Management Studio で「**ssisdb1**」データベースの [テーブル] フォルダを展開して、テーブルの一覧に「**Shohin2**」という名前のテーブルが表示されることを確認できます（表示されない場合は、[テーブル] フォルダを右クリックして、[最新の情報に更新] をクリックします）。



- 続いて、[Shohin2] テーブルを右クリックして、[上位 1000 行の選択] をクリックし、テーブルの中身を確認します。



商品テーブルのデータが、Excel ファイル内のデータ（以下）と同じであることを確認できます。

Shohin.xlsx ファイルのデータ

	A	B	C	D
1	商品番号	商品名	単価	
2	1	たぬきうどん	650	
3	2	きつねそば	650	
4	3	天ぷらうどん	750	
5	4	ざるそば	500	
6	5	冷やしたぬき	650	
7				

## 2.5 保存したパッケージ (.dtsx) の実行と編集

### ➡ 保存したパッケージ (.dtsx) の実行と編集

ここでは、前の STEP で保存したパッケージ「商品パッケージ.dtsx」を利用して、データの転送を再実行する手順を説明します。

### ➡ 64-bit (X64) マシンでパッケージを実行する場合

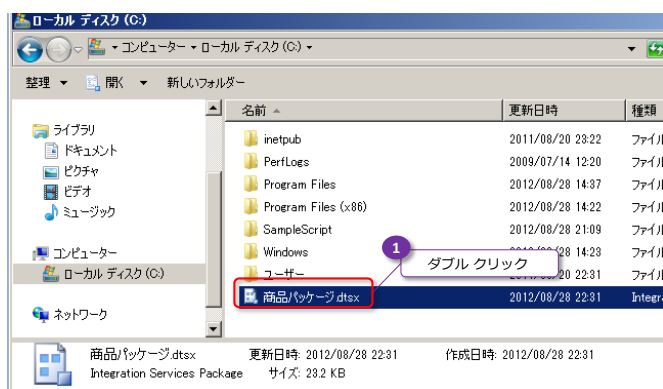
64-bit (X64) マシンで、保存したパッケージを実行するには、「**32-bit 版**」の「**Access データベース エンジン 2010 再頒布可能コンポーネント**」をインストールしておく必要があります（パッケージ内で **Access 2007** 以降の .accdb および **Excel 2007** 以降の .xlsx ファイルを扱う場合に、32-bit 版の再頒布コンポーネントが必要）。インポート/エクスポート ウィザードは 64-bit 版が提供されていますが、パッケージを実行するためのツールである「**パッケージ実行ユーティリティ**」(**dtexecUI.exe**) は 32-bit 版のみしか提供されていないため、32-bit 版の再頒布コンポーネントのインストールが必要になります（64-bit 版と 32-bit 版の両方をインストールすることはサポートされていないため、32-bit 版をインストールされたい場合、64-bit 版をあらかじめ削除してください）。

Access データベース エンジン 2010 再頒布可能コンポーネントのダウンロード URL  
<http://www.microsoft.com/ja-jp/download/details.aspx?id=13255>

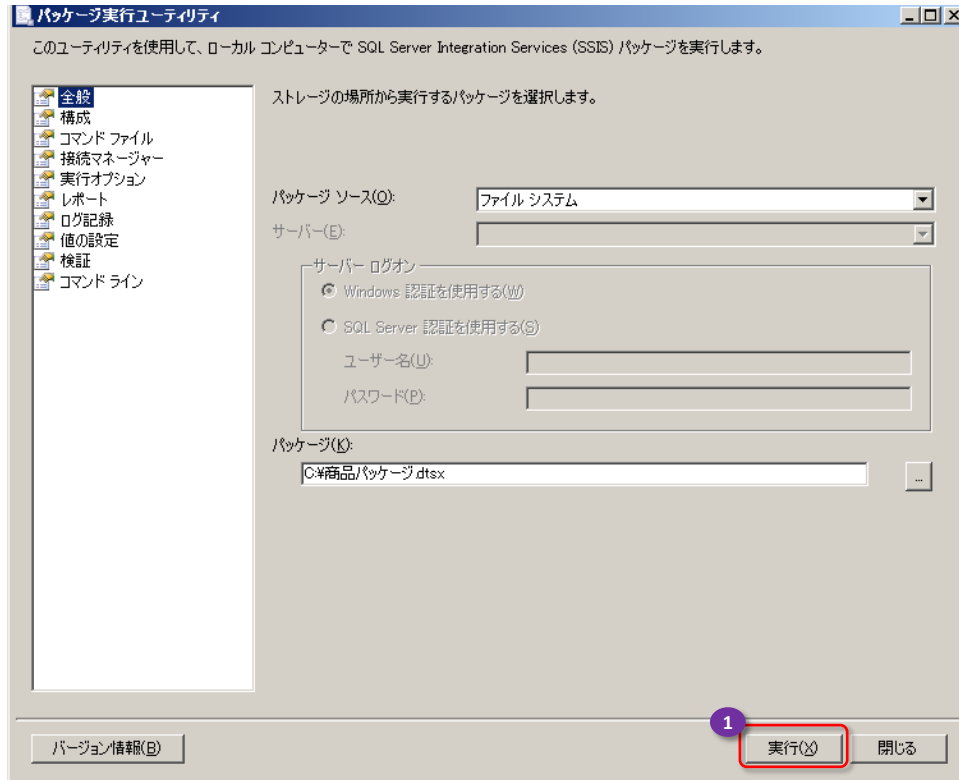
### ➡ 保存したパッケージの実行（パッケージ実行ユーティリティ）

再頒布コンポーネントのインストールが完了したら、次は、パッケージ実行ユーティリティ (**dtexecUI.exe**) を利用して、パッケージを実行してみましょう。

1. パッケージ実行ユーティリティを起動するには、**Windows エクスプローラー**から、パッケージ ファイル「**C:¥商品パッケージ.dtsx**」をダブル クリックします。

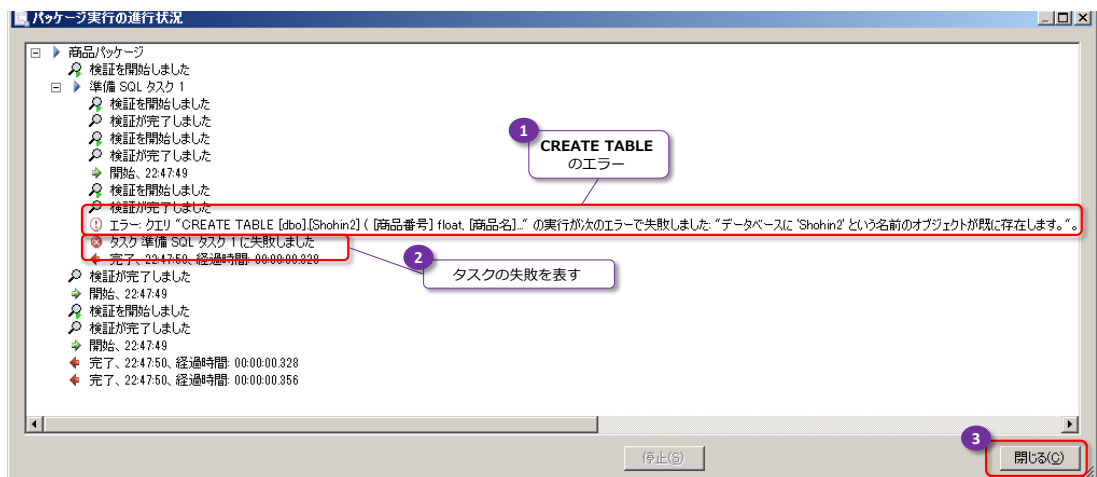


2. .dtsx ファイルをダブルクリックすると、次のように**パッケージ実行ユーティリティ**が起動します。



このダイアログでは、**[実行]**ボタンをクリックすると、パッケージを実行することができます。

3. パッケージの実行中は、次のように **「パッケージ実行の進行状況」** ダイアログが表示されますが、エラーが発生して、データ転送が失敗してしまいます。



「エラー」の行の記述をみると、「**CREATE TABLE [ssisdb1].[dbo].[Shohin2]**～」とあり、「**Shohin2**」テーブルを作成していることが原因でエラーとなっていることを確認できます。

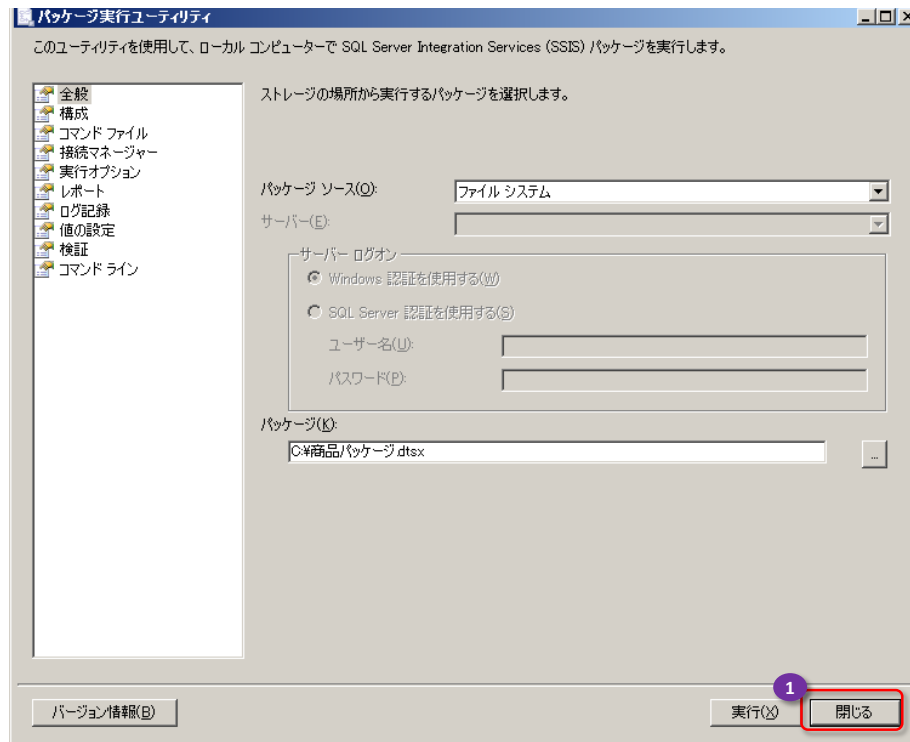
前の STEP で、ウィザードからデータ転送を実行したときは、「**Shohin2**」テーブルを作成することができましたが、2 回目以降のパッケージの実行時には、「Shohin2」テーブルが既に存在してしまっているので、エラーが発生してしまうのです。

確認後、**[閉じる]** ボタンをクリックして、ダイアログを閉じます。

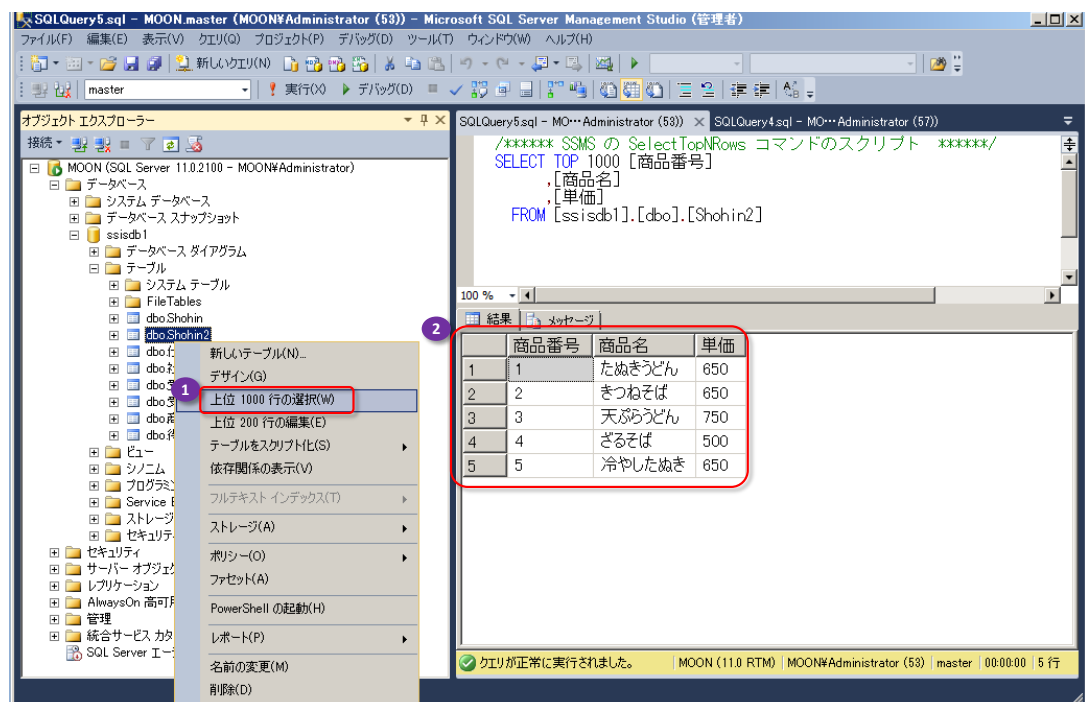
4. **「パッケージ実行ユーティリティ」** ダイアログへ戻ったら、**[閉じる]** ボタンをクリックして閉



じます。



- 次に、データ転送が失敗していることを Management Studio から念のため確認しておきましょう。「ssisdb1」データベースの「Shohin2」テーブルを右クリックして、「上位 1000 行の選択」をクリックします。



データが新しく追加されていないことから、データ転送が失敗していることを確認できます。

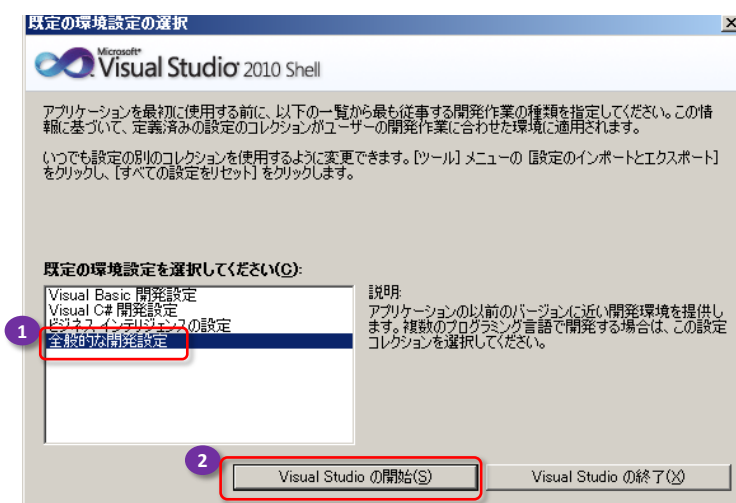
## ➡ 保存したパッケージの編集（SQL Server Data Tools）

次に、このパッケージのエラーを解消して、正しくデータ転送が行えるように、パッケージの編集を行ってみましょう。パッケージの編集には「**SQL Server Data Tools**」（以前のバージョンの Business Intelligence Development Studio）を利用します。

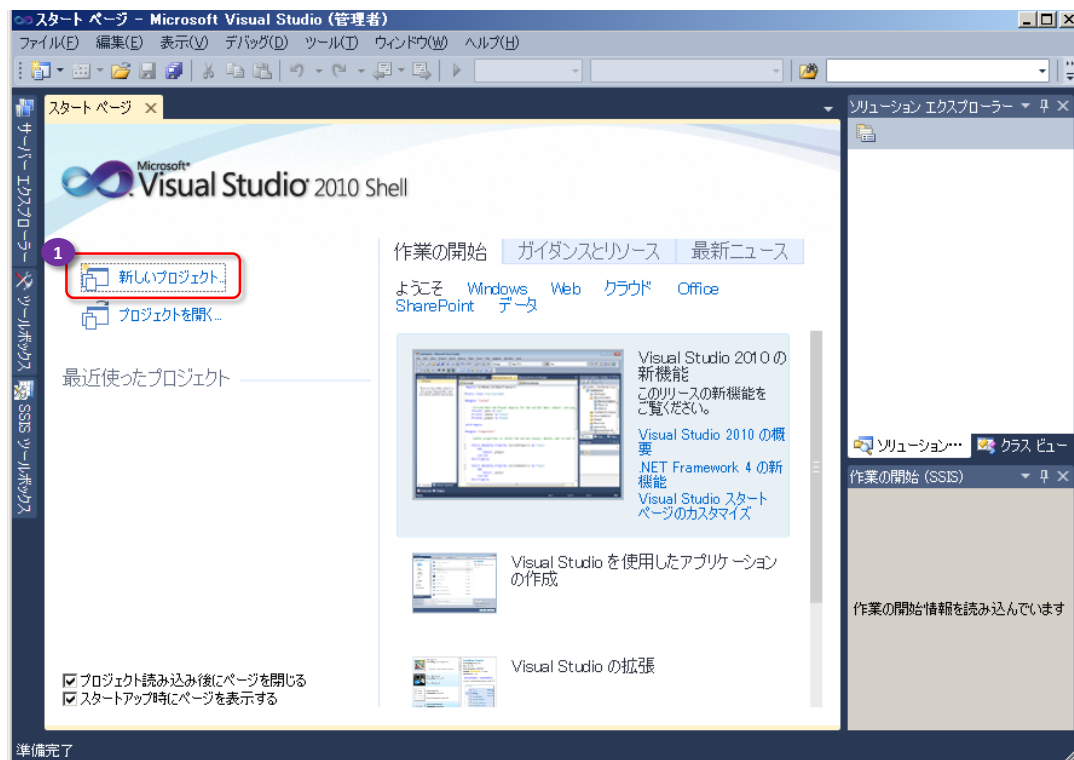
1. SQL Server Data Tools を起動するには、次のように【スタート】メニューの【すべてのプログラム】から【Microsoft SQL Server 2012】の【SQL Server Data Tools】をクリックします。



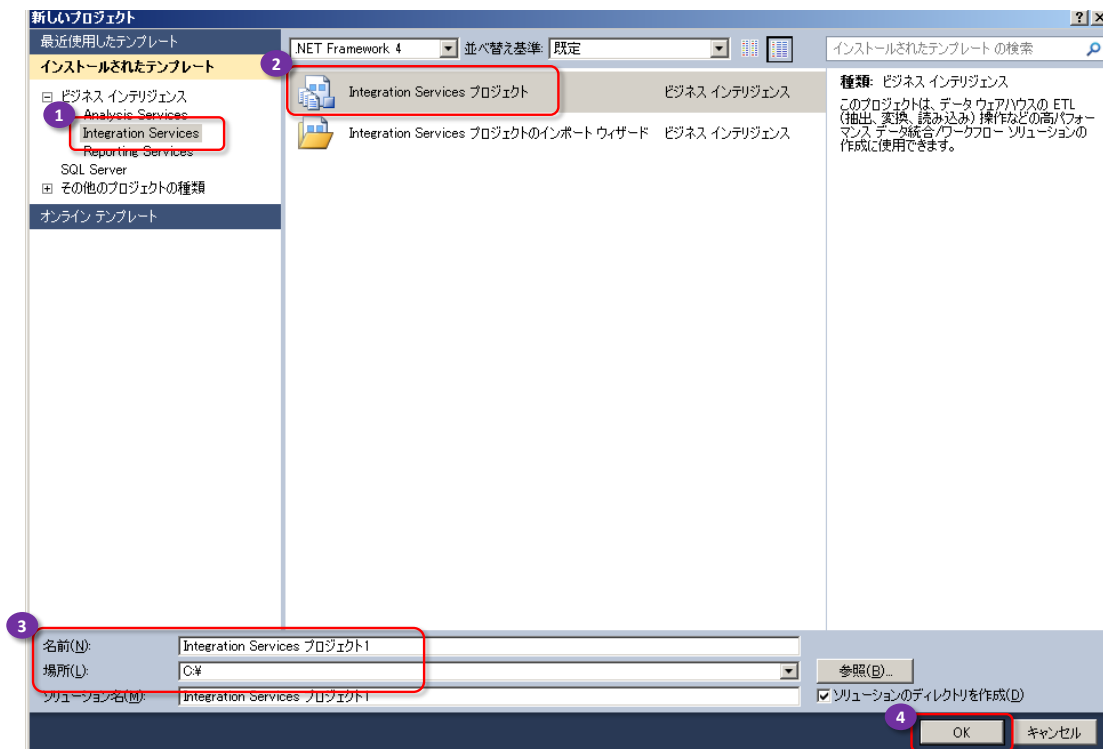
2. SQL Server Data Tools は、**Visual Studio 2010** に統合されたツールなので、Visual Studio 2010 が起動します。Visual Studio 2010 をはじめて起動する場合は、次のように【既定の環境設定の選択】ダイアログが表示されるので、任意の環境設定（画面は、**全般的な開発設定**）を選択して、【**Visual Studio の開始**】ボタンをクリックします。



3. Visual Studio 2010 が起動したら、【スタート ページ】の【新しいプロジェクト】をクリックして、新しいプロジェクトを作成します。



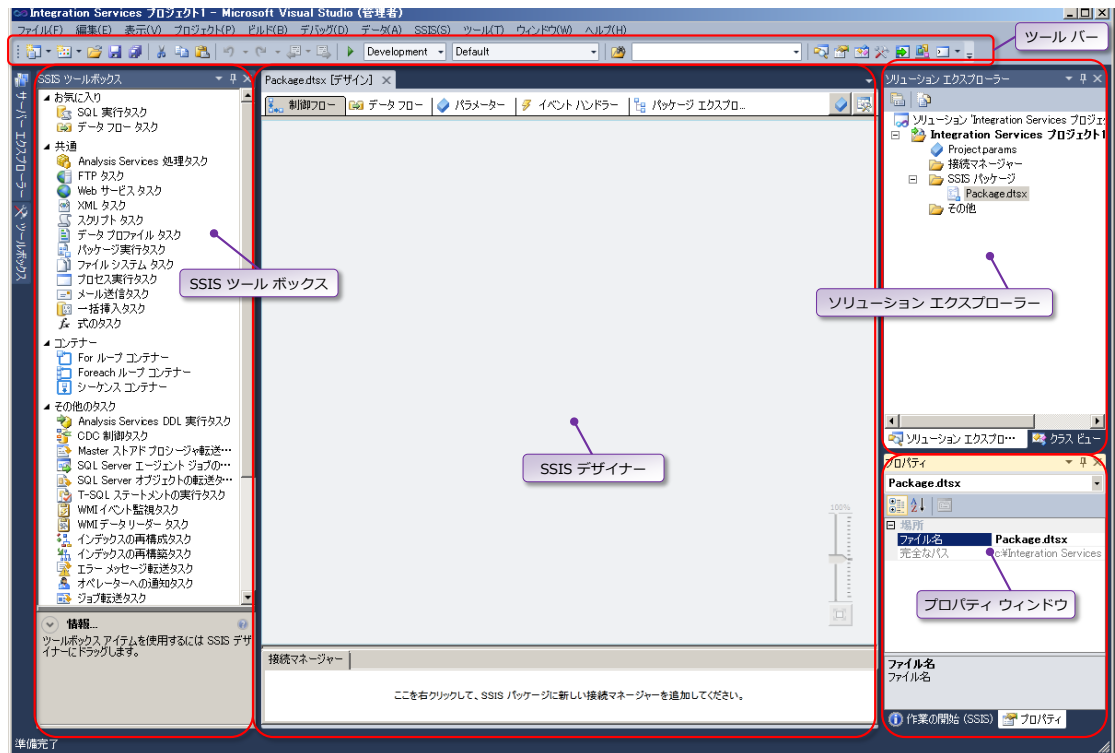
4. [新しいプロジェクト] ダイアログが表示されたら、次のように [インストールされたテンプレート] で [ビジネス インテリジェンス] の [Integration Services] を選択して、「Integration Services プロジェクト」を選択します。



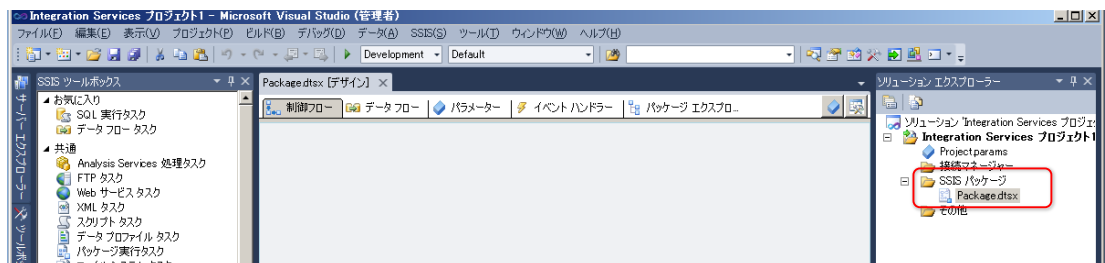
[名前] へ任意のプロジェクト名 (画面は **Integration Services プロジェクト 1**)、[場所] へ任意の保存場所 (画面は **C:\\$**) へ入力して、[OK] ボタンをクリックします。

5. これにより、Integration Services プロジェクト (**Integration Services プロジェクト 1**)

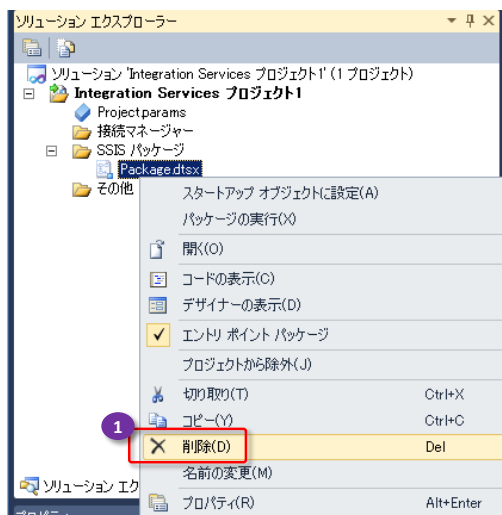
が作成されて、次のように画面が表示されます。



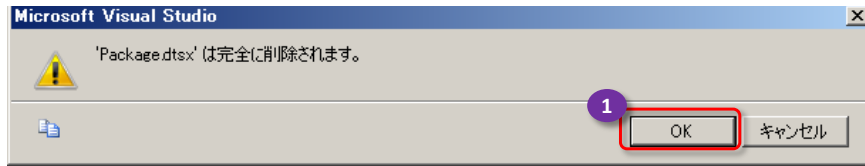
6. 次に、画面右側の「ソリューション エクスプローラー」で「SSIS パッケージ」フォルダーを展開すると、**Package.dtsx** という名前のパッケージが作成されていることを確認できます。



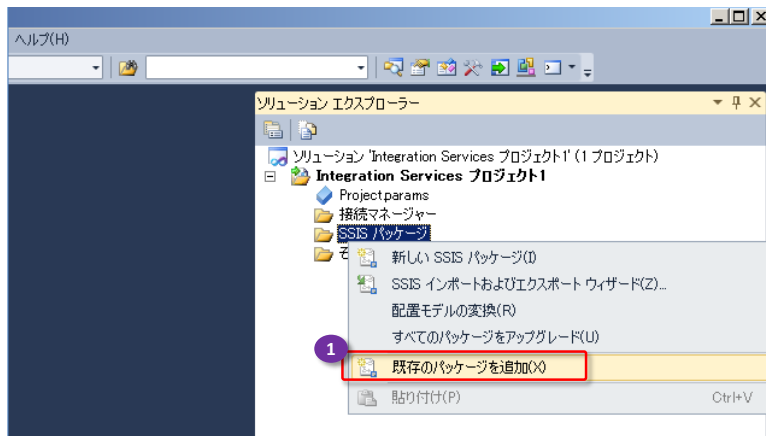
7. 今回は、このパッケージは利用しないので、次のように「**Package.dtsx**」を右クリックして、「**削除**」をクリックし、削除しておきます。



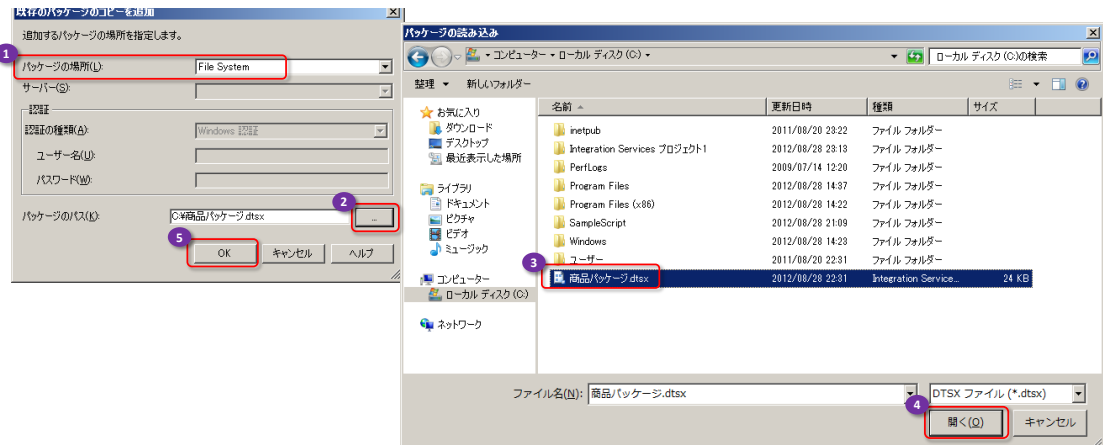
8. 削除を確認するダイアログが表示されるたら、[OK] ボタンをクリックします。



9. 次に、前の STEP で作成したパッケージ「商品パッケージ.dtsx」をプロジェクトへ追加します。次のようにソリューション エクスプローラーで [SSIS パッケージ] フォルダーを右クリックして、[既存のパッケージを追加] をクリックします。

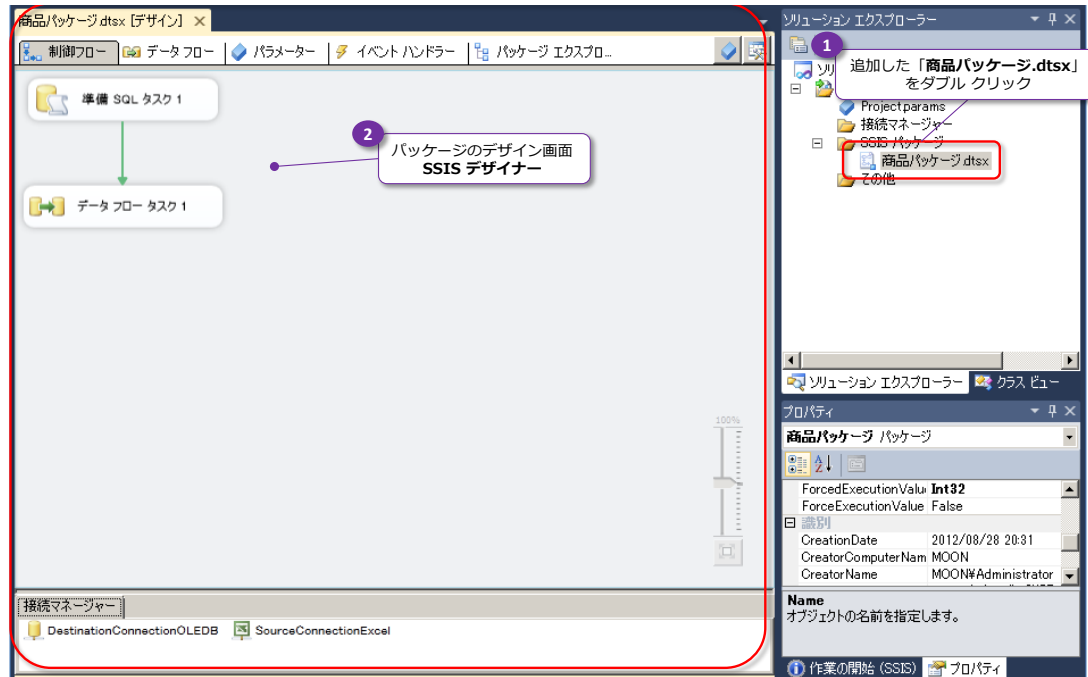


10. これにより、[既存のパッケージのコピーを追加] ダイアログが表示されるので、次のように [パッケージの場所] で [File System] を選択して、[パッケージのパス] で [...] ボタンをクリックします。



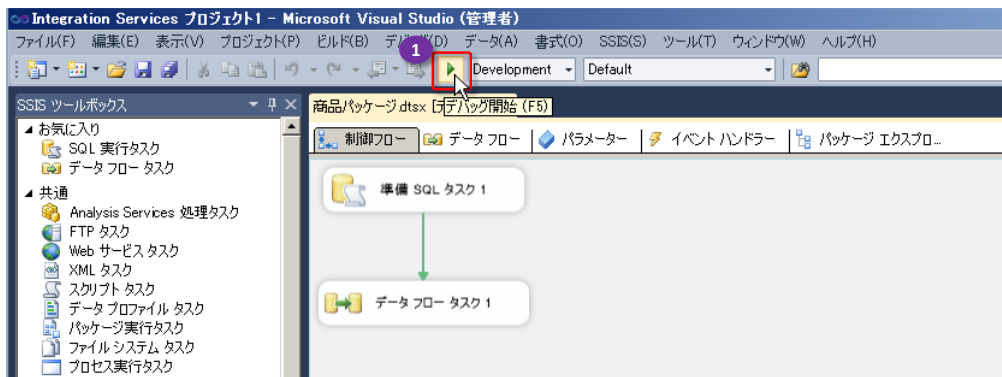
[パッケージの読み込み] ダイアログが表示されたら、「C:¥商品パッケージ.dtsx」ファイルを選択し、[開く] ボタンをクリックします。[既存のパッケージのコピーを追加] ダイアログへ戻ったら、[OK] ボタンをクリックして閉じます。

11. 次に、追加したパッケージ「商品パッケージ.dtsx」をダブル クリックします。

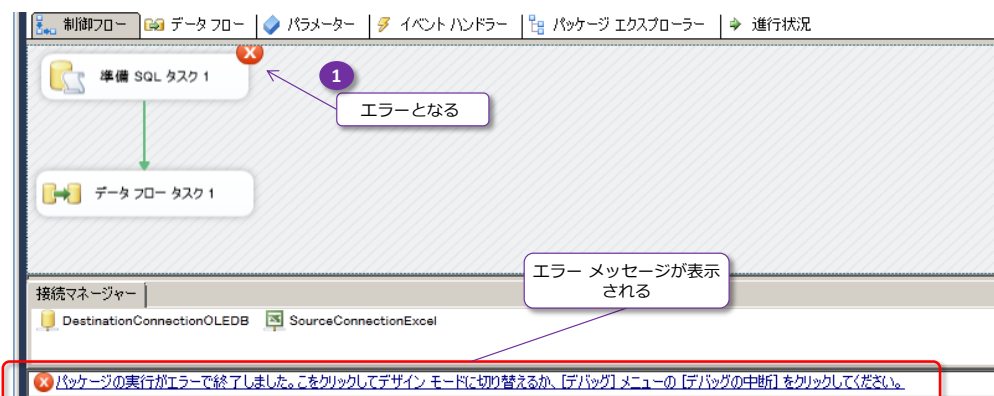


これにより、パッケージのデザイン画面（**SSIS デザイナー**）が表示されて、パッケージの編集ができるようになります。

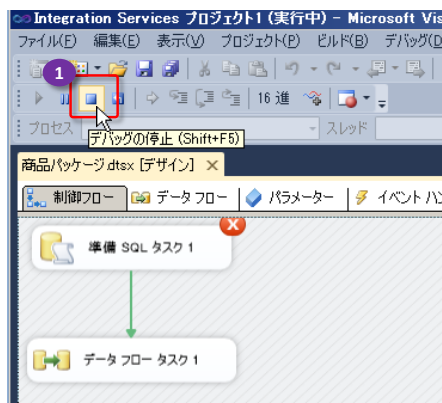
12. このデザイン画面からは、パッケージを実行することもできるので、ツールバーの「**デバッグ 開始**」ボタンをクリックして、パッケージを実行してみます。



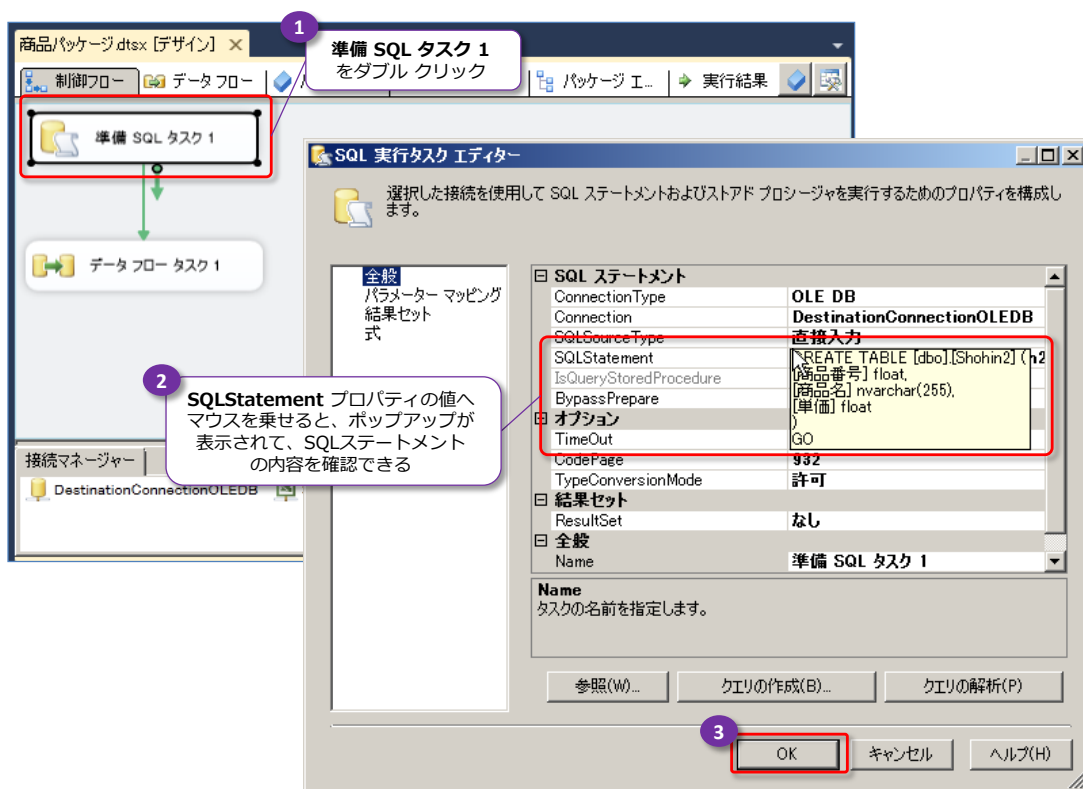
13. パッケージが実行されると、次のように「**準備 SQL タスク 1**」タスクがエラーとなって、実行が中断されます（デバッグ モードになります）。



14. ツールバーの「**デバッグの停止**」ボタンをクリックして、デバッグを終了します。



15. 次に、エラーとなったタスクの内容を確認するために、デザイン画面で「**準備 SQL タスク 1**」をダブル クリックします。



これにより、「**SQL 実行タスク エディター**」ダイアログが表示されるので、「**SQLStatement**」プロパティをクリックし、値（**CREATE TABLE ～**）へマウスを乗せて、ポップアップを表示すると、SQL ステートメントの内容を確認することができます。表示される SQL ステートメントは、「**CREATE TABLE [ssisdb1].[dbo].[Shohin2] …**」となっていて、**Shohin2** テーブルを作成するステートメントであることを確認できます。確認後、「**OK**」ボタンをクリックして、ダイアログを閉じます。

## ➡ タスクの無効化

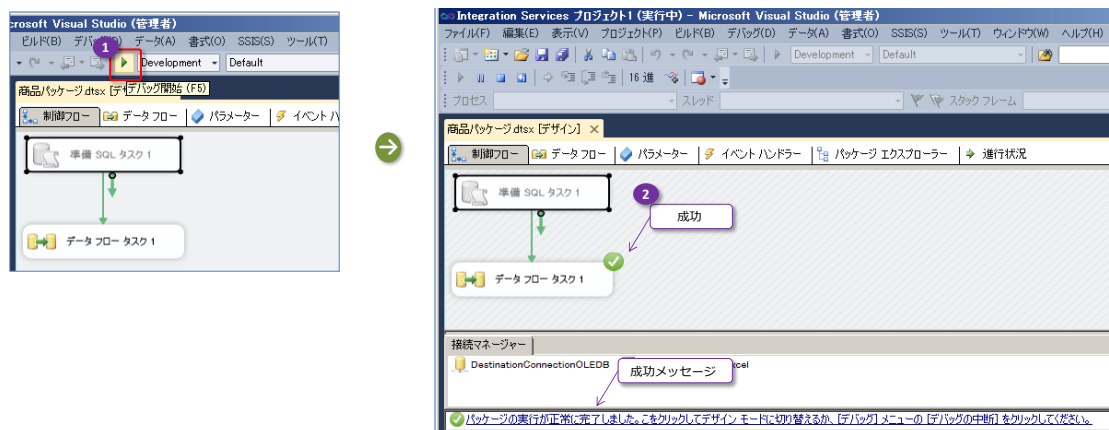
1. この「**準備 SQL タスク 1**」のエラーを回避するには、このタスクを無効化に設定すればよい

ので、次のように「準備 SQL タスク 1」を右クリックして「無効化」をクリックします。



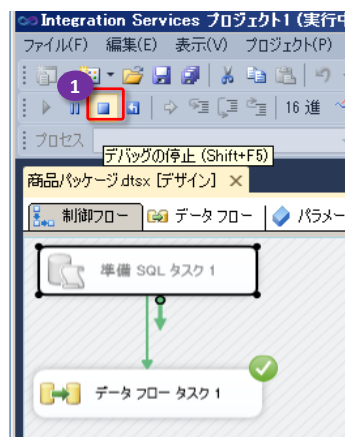
これにより、「準備 SQL タスク 1」が灰色に変わり、このタスクが実行されないようになります。

- 次に、ツールバーの「デバッグ開始」ボタンをクリックして、パッケージを実行（デバッグを開始）します。



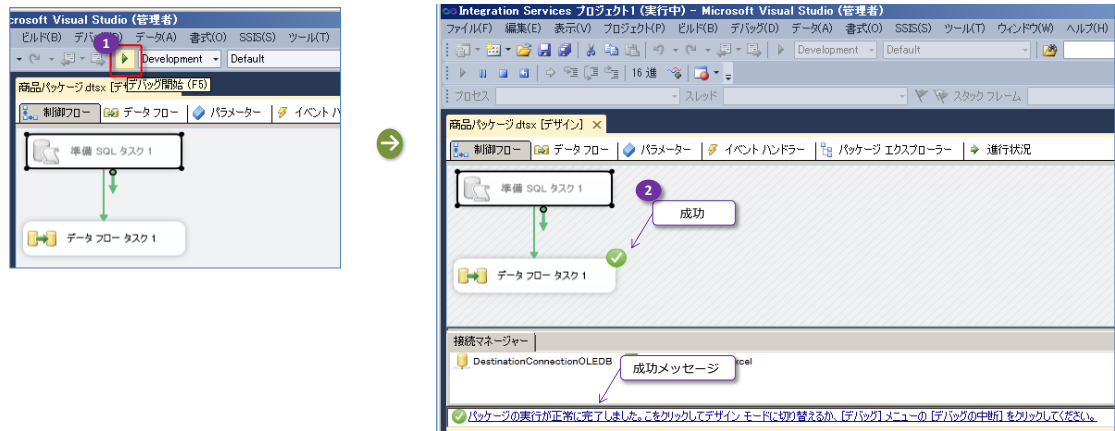
今度は、無効化した「準備 SQL タスク 1」は実行されずに、「データフロー タスク 1」（データ転送のためのタスク）に「緑のチェックマーク」が付いて、実行が成功したことを確認できます。

- 次に、ツールバーの「デバッグの停止」ボタンをクリックしてデバッグを終了します。

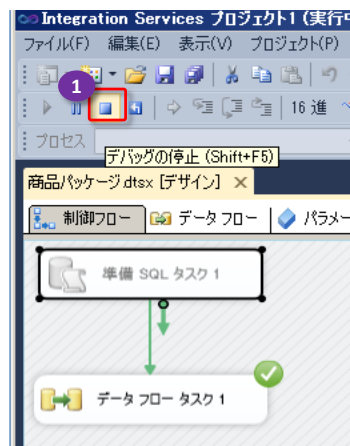




4. 続いて、もう一度、ツールバーの「デバッグ開始」ボタンをクリックして、パッケージを再実行し、エラーにならないことを確認しておきます。

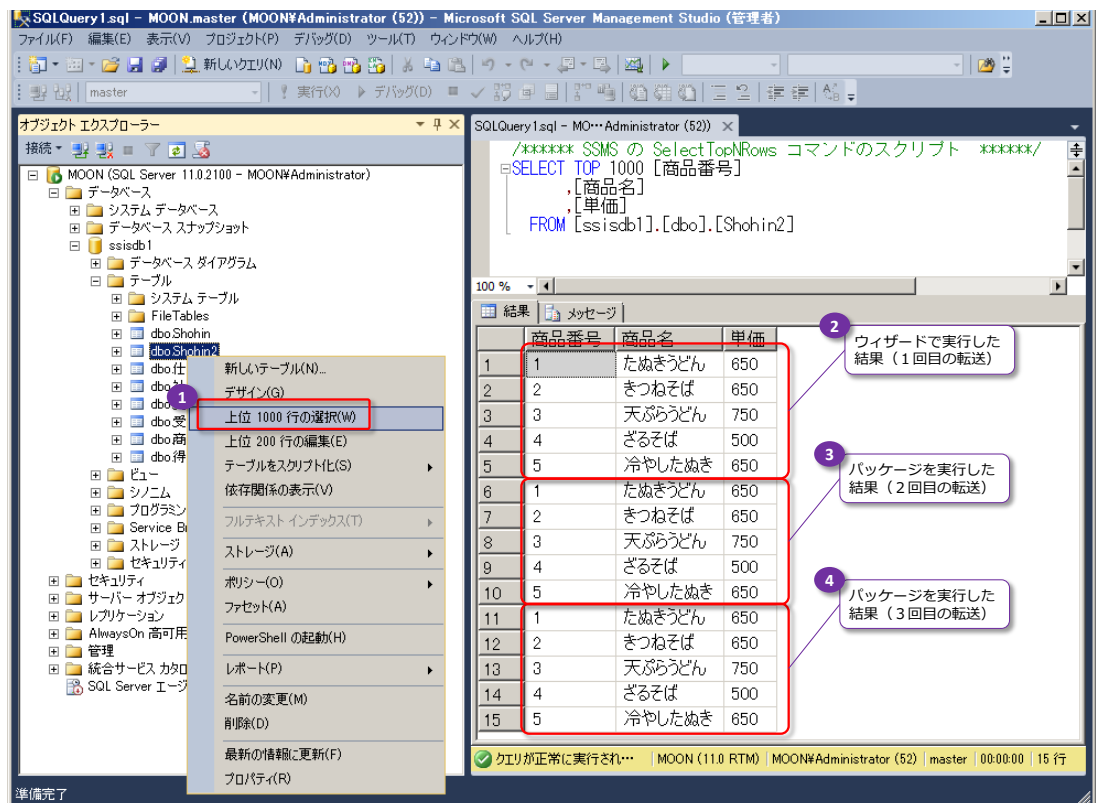


5. 実行後、ツールバーの「デバッグの停止」ボタンをクリックしてデバッグを終了します。



## ➡ 取り込んだデータの確認

6. 次に、パッケージの実行で取り込んだデータを確認しておきましょう。 Management Studio から、次のように **ssisdb1** データベース内の「**Shohin2**」テーブルを右クリックして「**上位 1000 行の選択**」をクリックします。



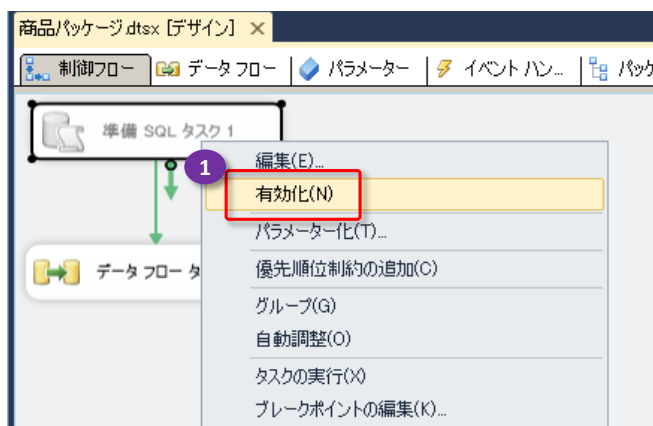
パッケージを実行した回数分だけ、データが追加されていることを確認できます。

このように、CREATE TABLE ステートメントが実行されないように変更すれば、データ転送を何度も実行できるようになります。ただし、このままでは、パッケージが実行されるたびにデータが上書きされずにどんどん追加されていってしまいます。

## ➡ データを削除してからデータ転送を行う（上書きする）

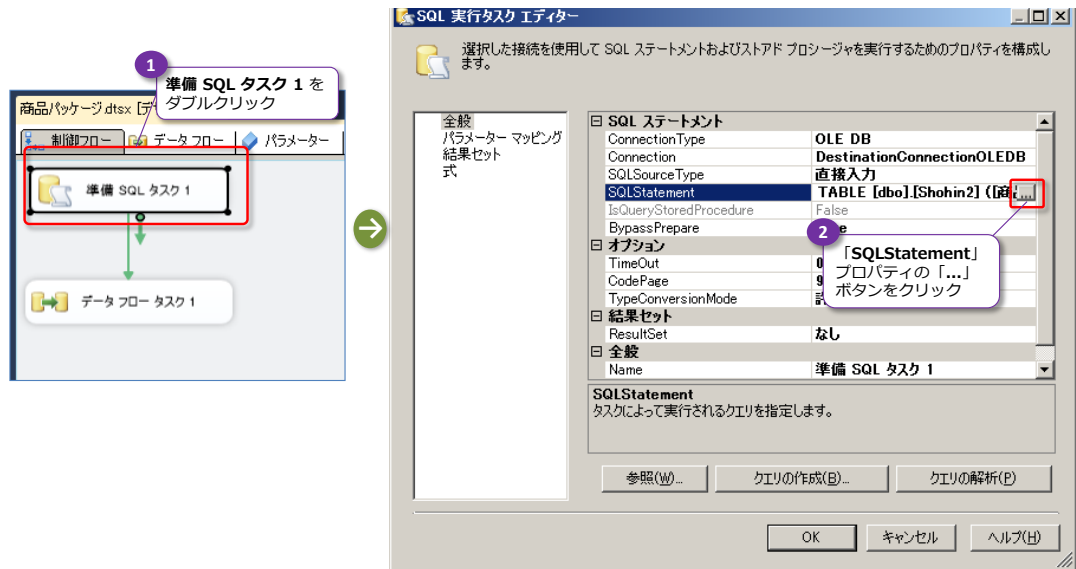
パッケージを実行するたびにデータが追加されていくのを防ぎ、新しいデータで上書きされるようにするには、次のようにパッケージを編集します

1. まずは、無効化していた「準備 SQL タスク 1」を有効に戻します。「準備 SQL タスク 1」を右クリックして「有効化(N)」をクリックします。



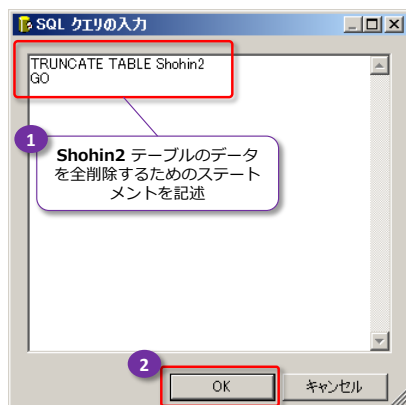
2. 次に、「準備 SQL タスク 1」の SQL ステートメントを変更します。「準備 SQL タスク 1」

をダブル クリックして、[SQL 実行タスク エディター]ダイアログを表示し、[SQLStatement] プロパティの「...」 ボタンをクリックします。



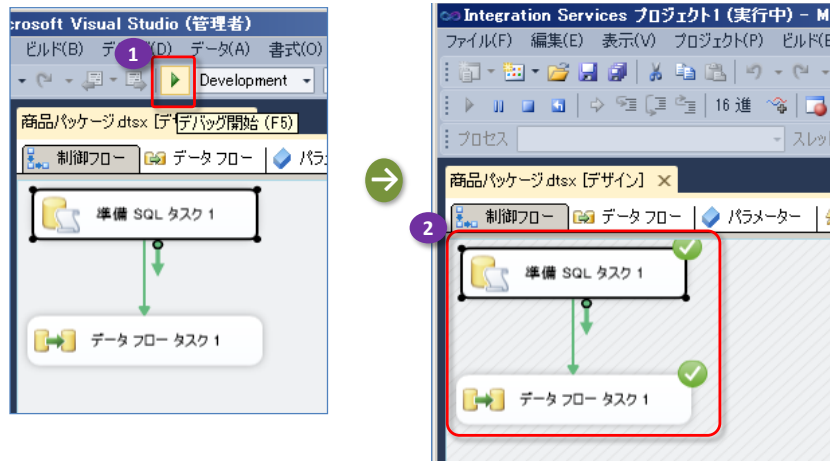
3. [SQL クエリの入力] ダイアログが表示されたら、SQL ステートメントを次のように変更します。

```
TRUNCATE TABLE Shohin2
go
```



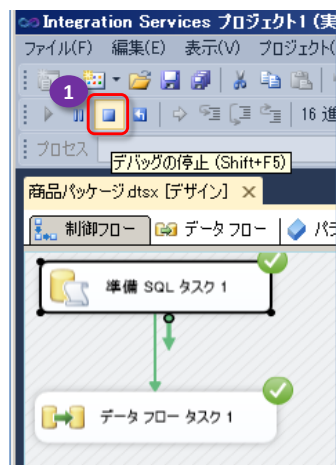
**TRUNCATE TABLE** ステートメントは、指定したテーブルのデータをすべて削除する効果があるので、この記述によって、**Shohin2** テーブルのデータをすべて削除できるようになります。ステートメントの記述後、[OK] ボタンをクリックしてダイアログを閉じます。

4. 次に、ツールバーの [デバッグ開始] ボタンをクリックして、パッケージを実行し、結果を確認します。



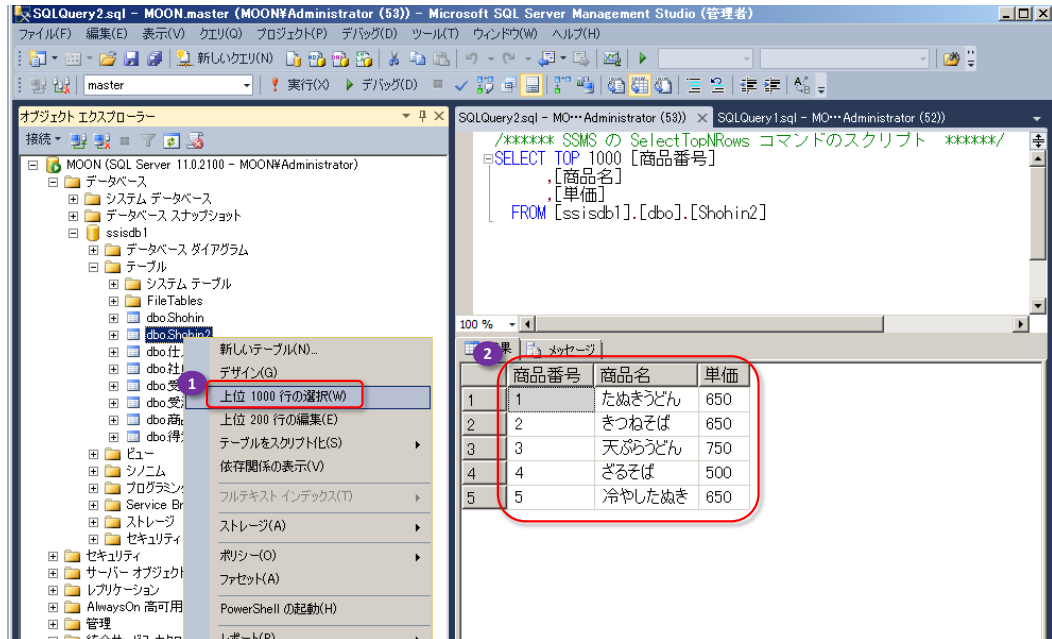
今度は、2つのタスクへ緑のチェックマークが付いて、パッケージの実行が成功したことを確認できます。

5. ツールバーの「**デバッグの停止**」ボタンをクリックして、デバッグを終了します。



## ➡ データが上書きされていることの確認

6. パッケージで取り込んだデータを確認するために、Management Studio から「**Shohin2**」テーブルを右クリックして「**上位 1000 行の選択**」をクリックします。



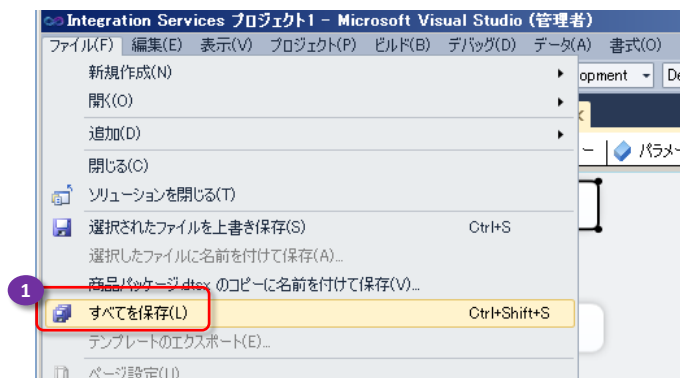
テーブルのデータは、すべて削除されてから、データ転送が行われているので、データが上書きされていることを確認できます。

このように、SQL Server Data Tools (SSIS デザイナー) を利用すると、単純なデータ転送だけでなく、任意の SQL ステートメントを実行できるようになります。また、SSIS デザイナーでは、"データの変換" を追加することもできるので、これについては STEP 3 で説明します。

## ➡ パッケージの保存

次に、編集したパッケージを保存しておきます。

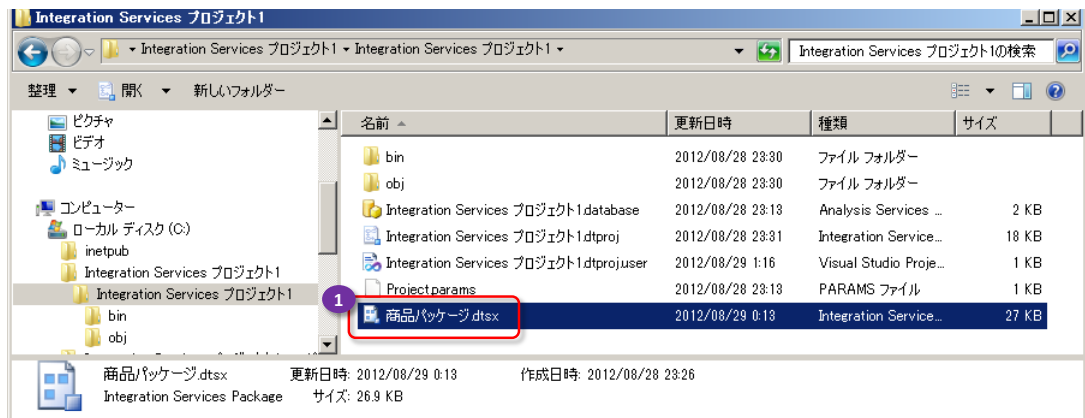
1. パッケージとプロジェクト全体を保存するには、次のように【ファイル】メニューから【すべてを保存】をクリックします。



## ➡ パッケージ実行ユーティリティからの実行

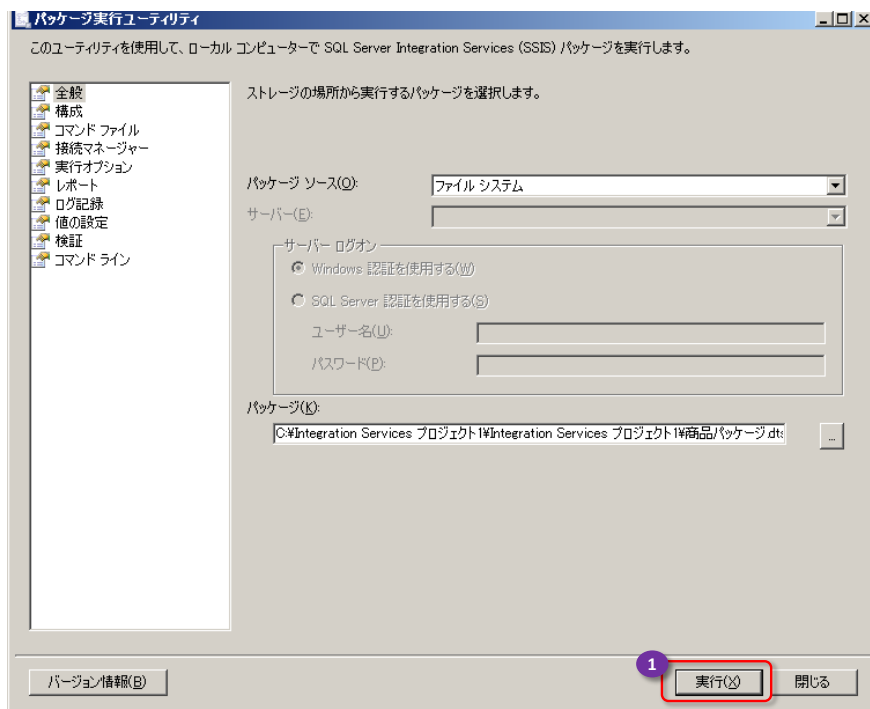
次に、SQL Server Data Tools (SSIS デザイナー) で編集したパッケージを **Windows エクスプローラー** から実行してみます。

1. SQL Server Data Tools で作成／編集したパッケージは、次のようにプロジェクト（C:\¥Integration Services プロジェクト 1¥Integration Services プロジェクト 1）フォルダーへ格納されています。

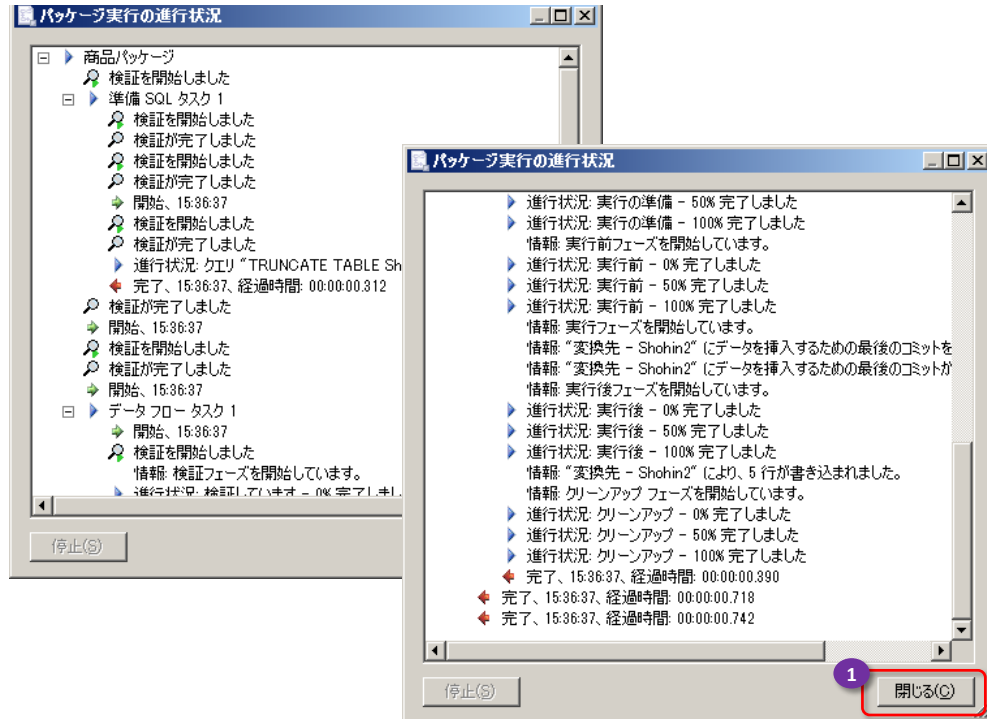


このフォルダー内のパッケージ ファイル（商品パッケージ.dtsx）をダブル クリックして、パッケージを実行してみましょう。

2. 「パッケージ実行ユーティリティ」ダイアログが表示されたら、[実行] ボタンをクリックしてパッケージを実行します。

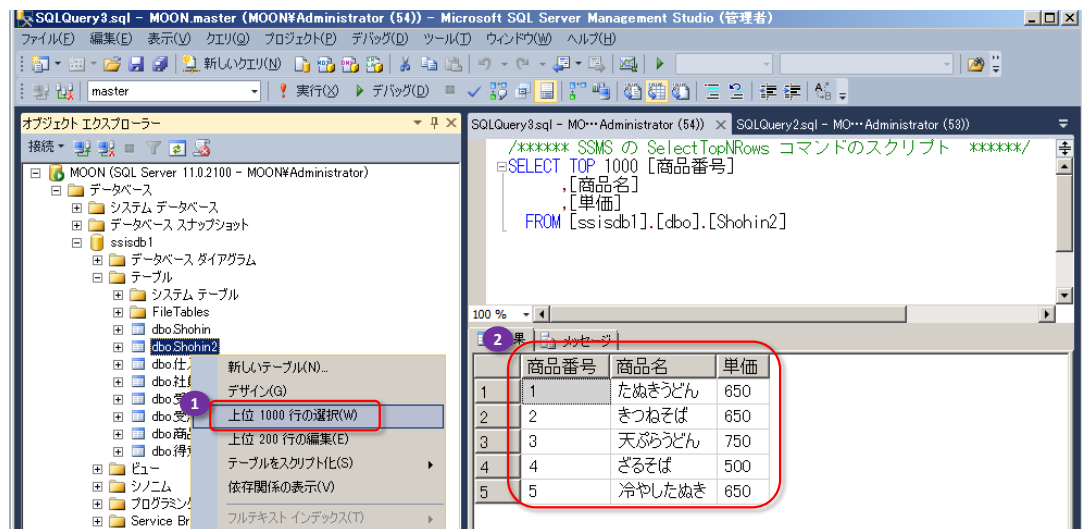


今度は、エラーが発生せずに、実行が正常に完了したことを確認できます。



完了後は、[閉じる] ボタンをクリックして、ダイアログを閉じます。

- 次に、パッケージで取り込んだデータを確認するために、Management Studio から「Shohin2」テーブルを右クリックして [上位 1000 行の選択] をクリックします。



データが上書きされて、正しくパッケージが実行されたことを確認できます。

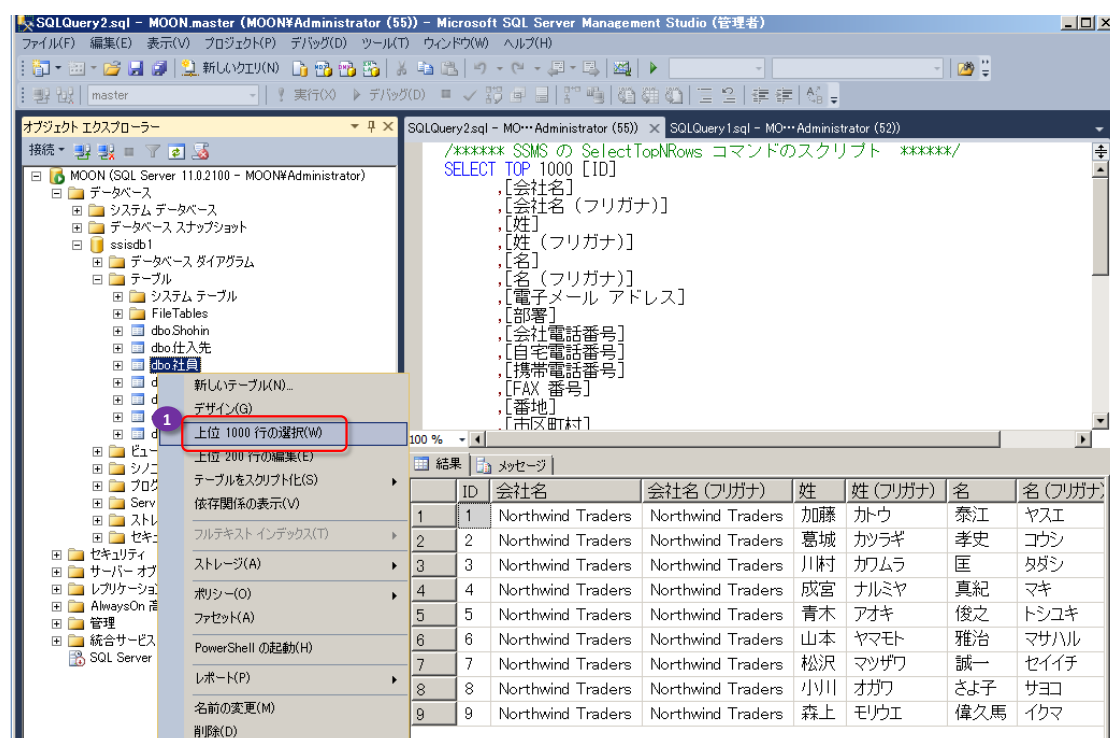
## 2.6 データのエクスポート

### ➡ データのエクスポート

ここでは、**STEP 2.3** で Access データベースから取り込んだ「社員」テーブルを、インポート／エクスポート ウィザードを利用して、「tsv」形式（タブ区切り）のテキスト ファイルへエクスポートする（書き出す）手順を説明します。

### ➡ エクスポートするデータ

エクスポートする社員テーブルは、次のとおりです。

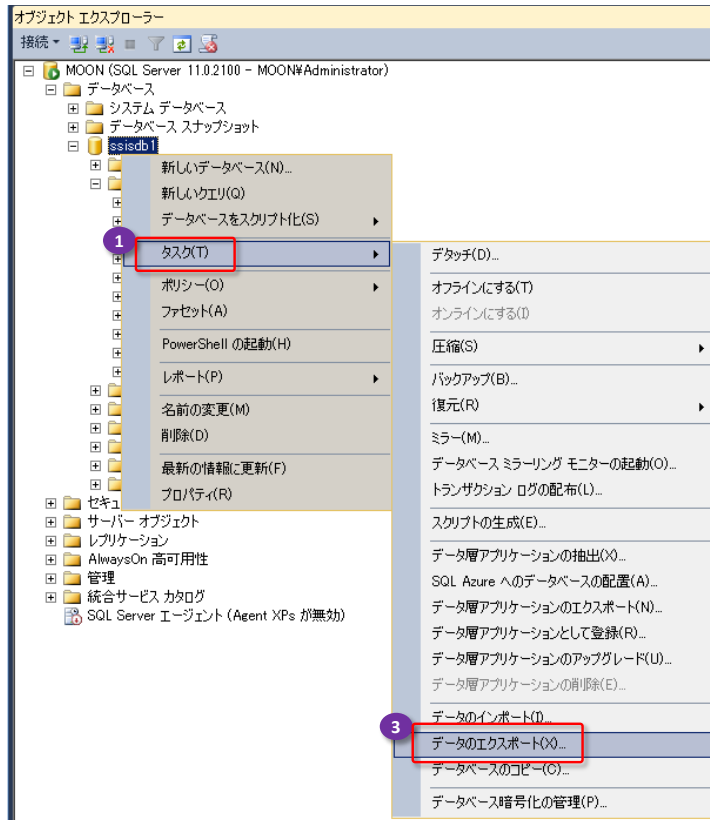


このテーブルの中から、「ID」、「姓」、「姓（フリガナ）」、「名」、「名（フリガナ）」、「電子メール アドレス」、「部署」という 7 つの列をエクスポートしてみます。

### ➡ 実行手順

1. インポート／エクスポート ウィザードを利用してデータをエクスポートするには、次のように Management Studio で転送元のデータベース（ssisdb1）を右クリックして、[タスク] メニューの「データのエクスポート」をクリックします。

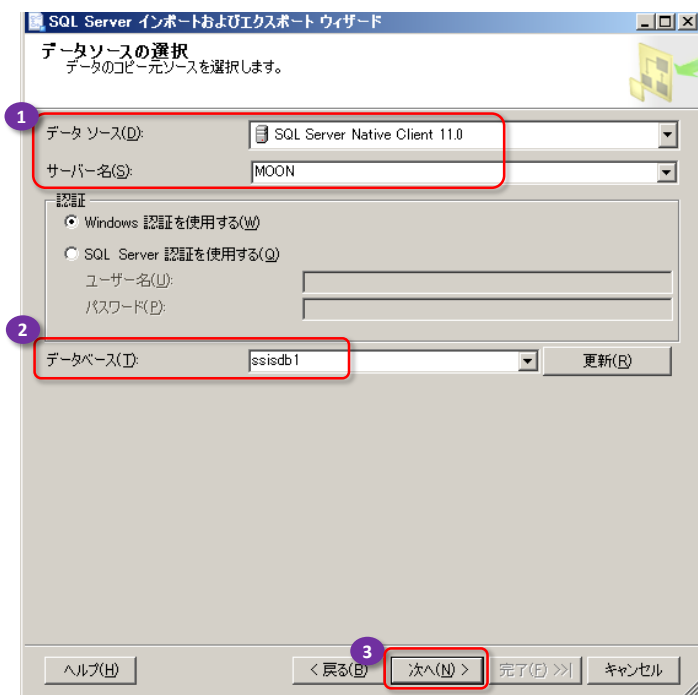




2. これにより、データのインポート時と同様、次のようにインポート／エクスポート ウィザードが起動するので、[次へ] ボタンをクリックします。

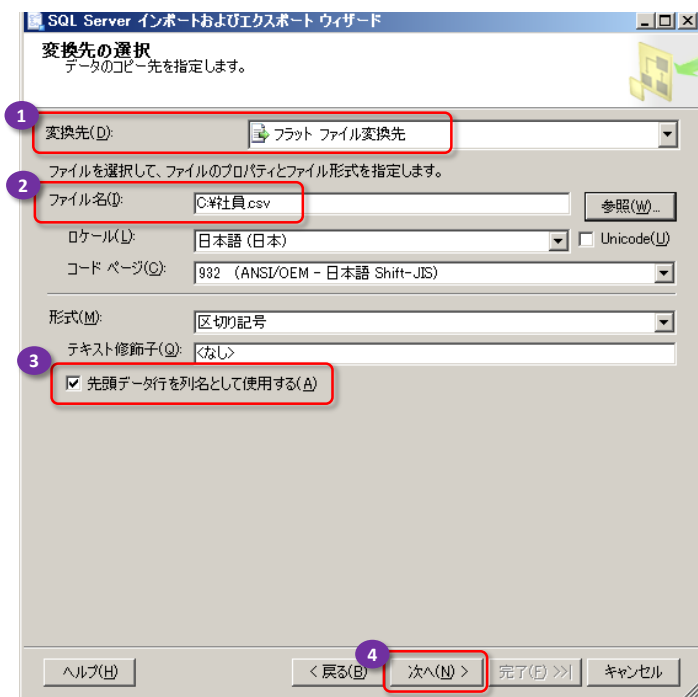


3. 次の[データ ソースの選択] ページでは、データの転送元（エクスポート元となるデータ）を指定します。



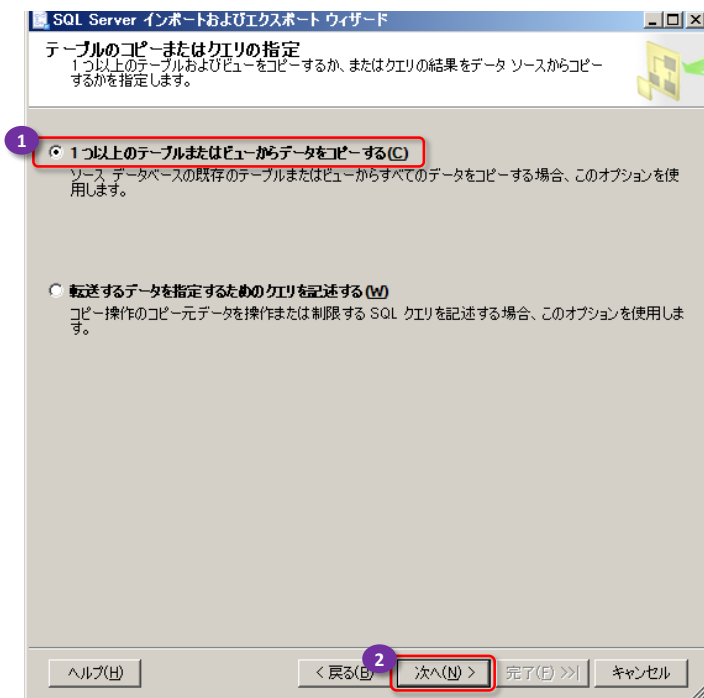
今回は、事前に「ssisdb1」データベースを選択してウィザードを起動しているので、[データベース] へ「ssisdb1」が選択されていることを確認して、[次へ] ボタンをクリックします。

4. 次の[変換先の選択] ページでは、データの転送先（エクスポート先）を指定します。

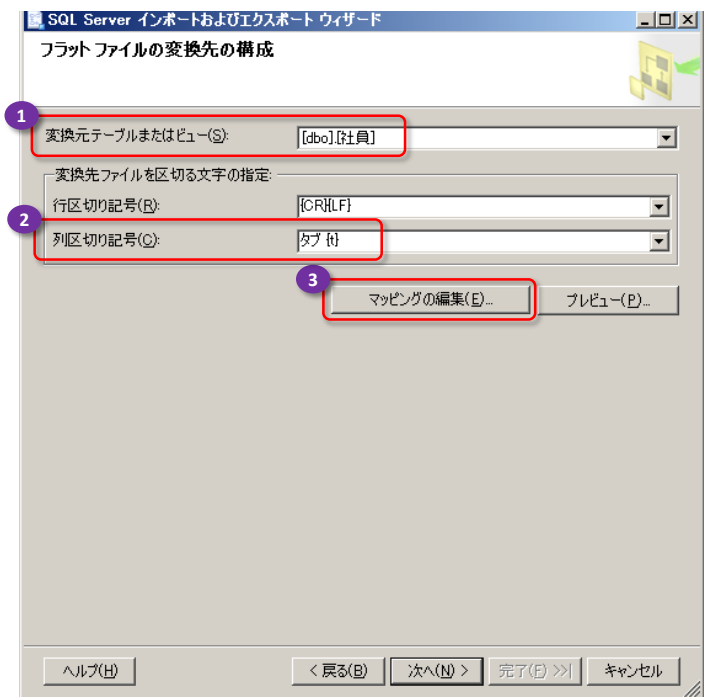


今回は、テキスト ファイルへデータを書き出すので、[変換先] で「フラット ファイル変換先」を選択します。[ファイル名] へは、エクスポート先となるファイル名を入力しますが、今回は「C:\¥社員.tsv」と入力します。また、[先頭データ行を列名として使用する] チェック ボックスをチェックして、ファイルの 1 行目には、列名が書き出されるようにします。設定後、[次へ] ボタンをクリックします。

5. 次の「**テーブルのコピーまたはクエリの指定**」ページでは、「**1 つ以上のテーブルまたはビューからデータをコピーする**」を選択して、「**次へ**」ボタンをクリックします。

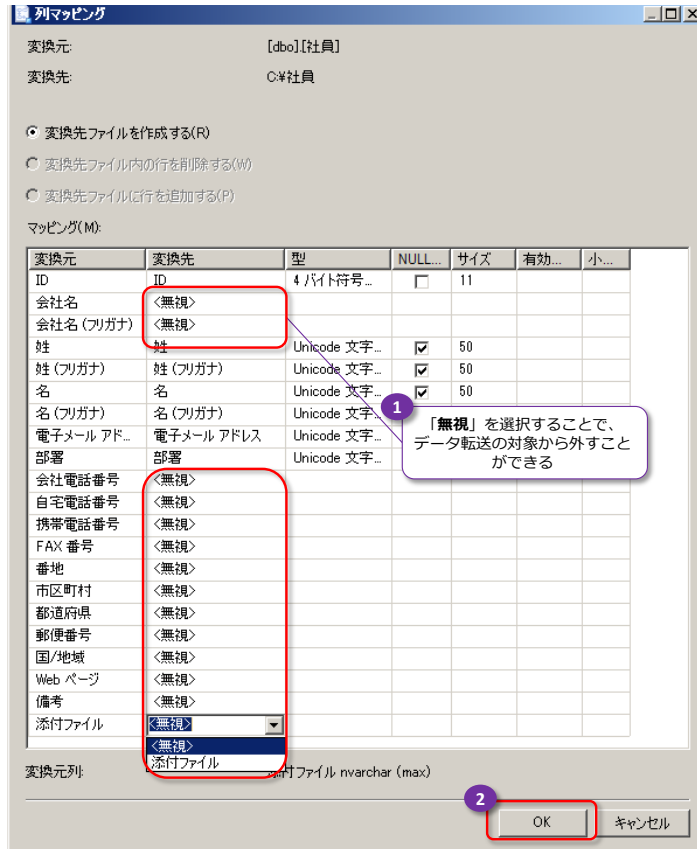


6. 次の「**フラット ファイルの変換先の構成**」ページでは、転送元となるテーブルを選択します。今回は、「**社員**」テーブルを選択します。



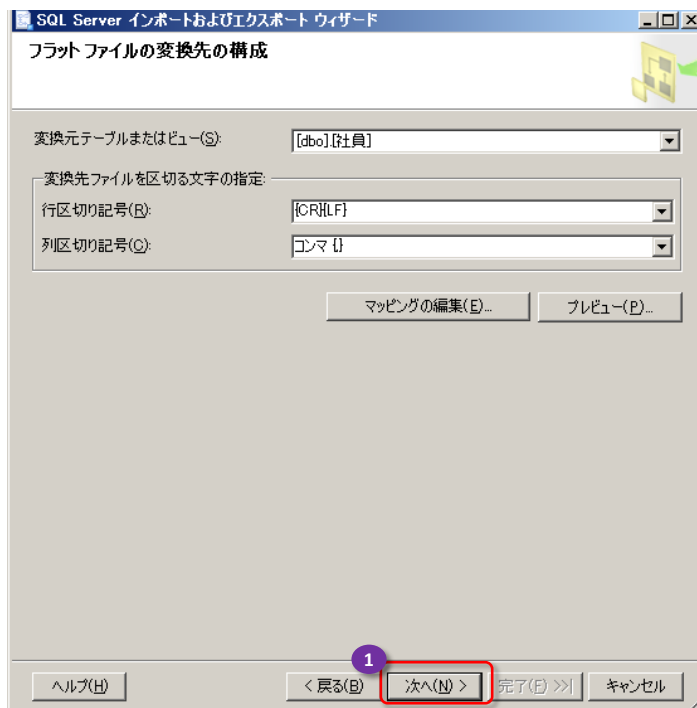
また、「**列区切り記号**」で「**タブ {t}**」を選択し、タブ区切りのテキスト ファイルとしてエクスポートするようにします。続いて、「**マッピングの編集**」ボタンをクリックします。

7. これにより、「**列マッピング**」ダイアログが表示されて、エクスポートするデータの列名や、データ転送の対象から外したい列の変更などを行えるようになります。

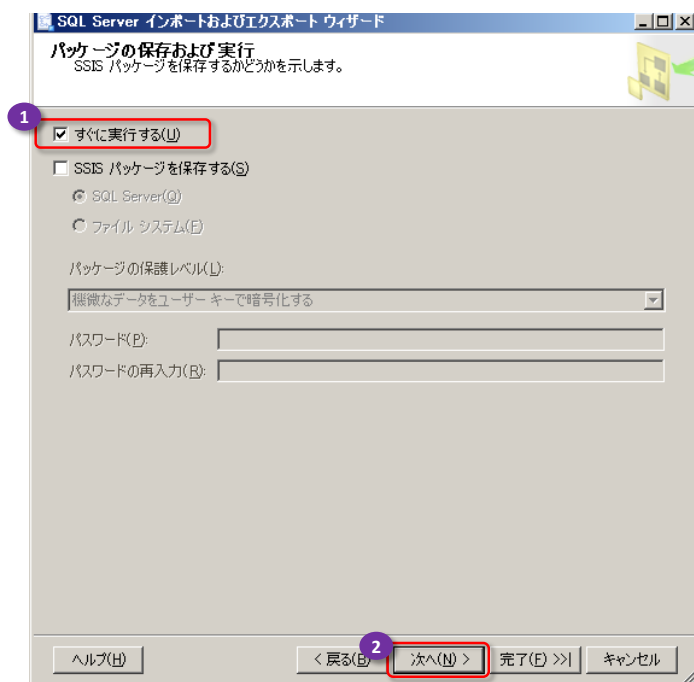


今回は、「ID」と「姓」、「姓 (フリガナ)」、「名」、「名 (フリガナ)」、「電子メール アドレス」、「部署」列のみをデータ転送するようにするので、それ以外の列（会社名や会社電話番号列など）に対して、[変換先] で「無視」を選択して、データ転送の対象から外すようにします。設定後、[OK] ボタンをクリックします。

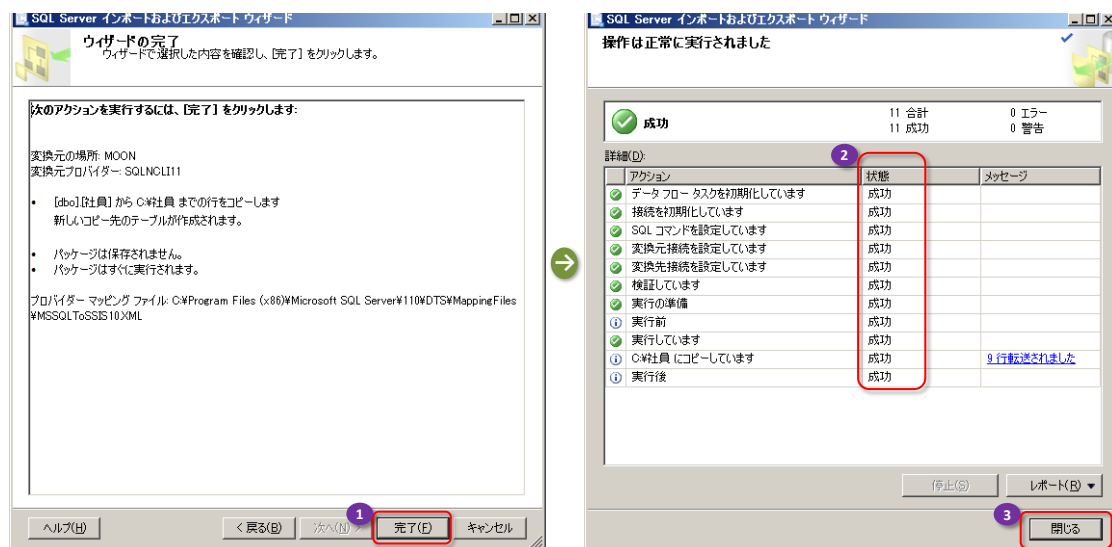
8. [フラット ファイルの変換先の構成] ページへ戻ったら、[次へ] ボタンをクリックします。



9. 次の「パッケージの保存および実行」ページでは、「**すぐに実行する**」を選択して、「**次へ**」ボタンをクリックします。



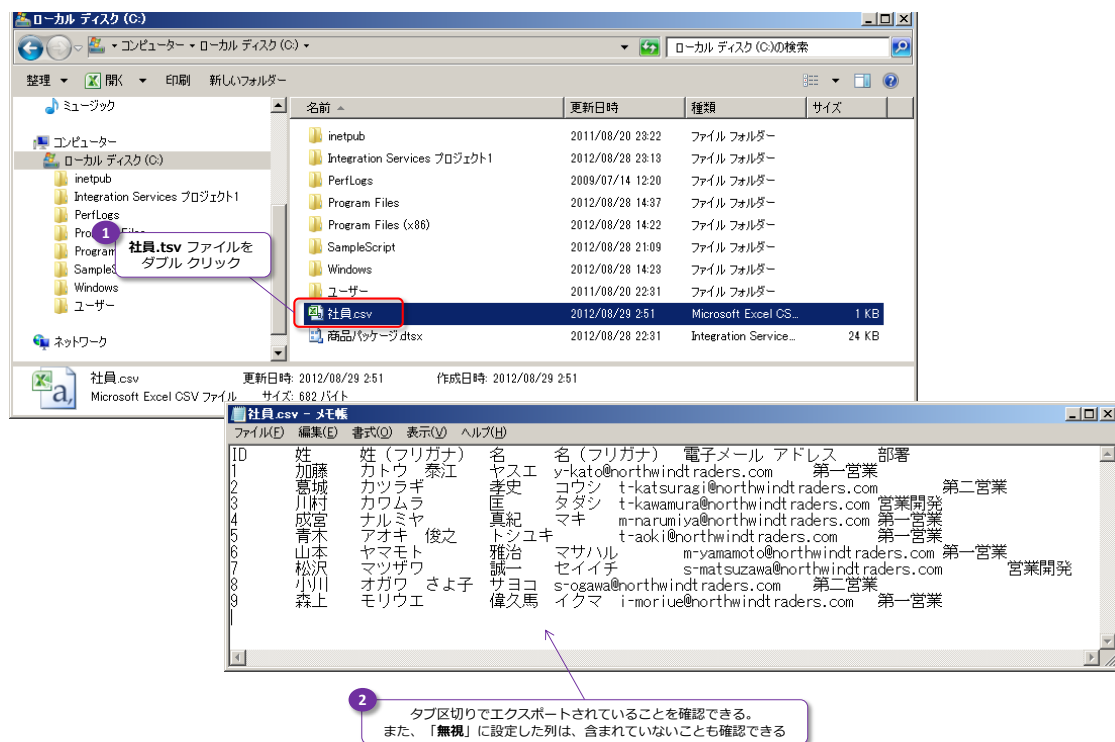
10. 最後の「ウィザードの完了」ページでは、「**完了**」ボタンをクリックするとデータのエクスポートが開始されます。



実行が完了すると、「**操作は正常に実行されました**」ページが表示されます。「**状態**」が「**成功**」と表示されていることを確認して、「**閉じる**」ボタンをクリックします。

## ➡ エクスポートしたデータの確認

11. ウィザードでエクスポートしたデータを確認するには、**Windows エクスプローラー**を起動して、「**C:¥社員.tsv**」ファイルをダブル クリックして開きます。



それぞれの列がタブ区切りで表示されて、「ID」と「姓」、「姓 (フリガナ)」、「名」、「名 (フリガナ)」、「電子メール アドレス」、「部署」列のみがエクスポートされていることを確認できます。

## STEP 3. SSIS デザイナーの基本操作

---

この STEP では、Integration Services (SSIS) の「**SSIS デザイナー**」を使用して、データの変換を伴うデータの転送を行い、SSIS デザイナーの基本操作を説明します。

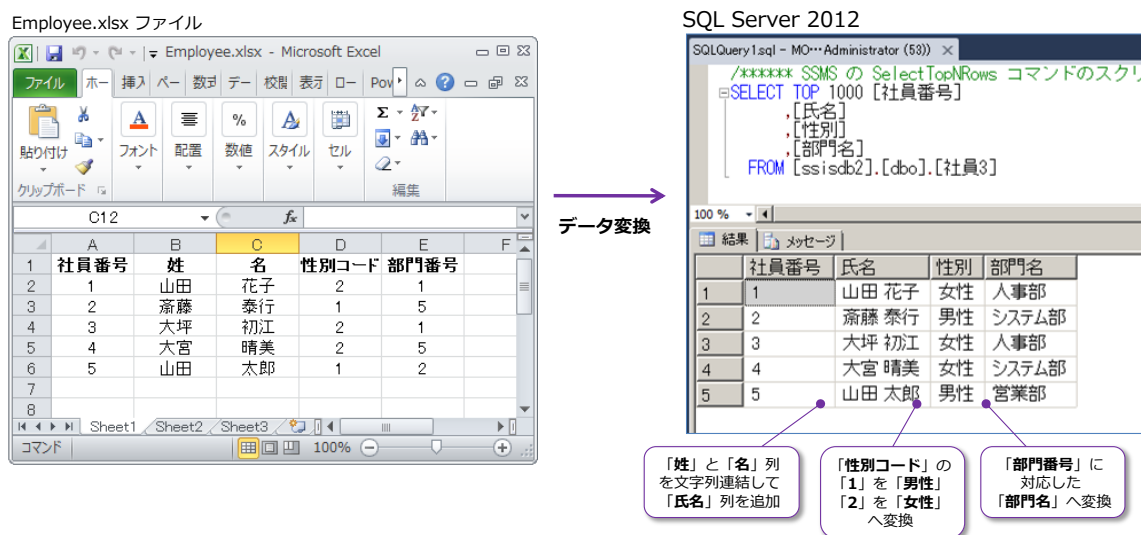
この STEP では、次のことを学習します。

- ✓ SSIS デザイナーの使用方法
- ✓ データ フロー タスクによるデータの転送
- ✓ 派生列コンポーネントによるデータ変換
- ✓ 参照コンポーネントによる別テーブルのデータ取得
- ✓ データ ビューアーによる転送中のデータの表示

## 3.1 作成するパッケージの概要

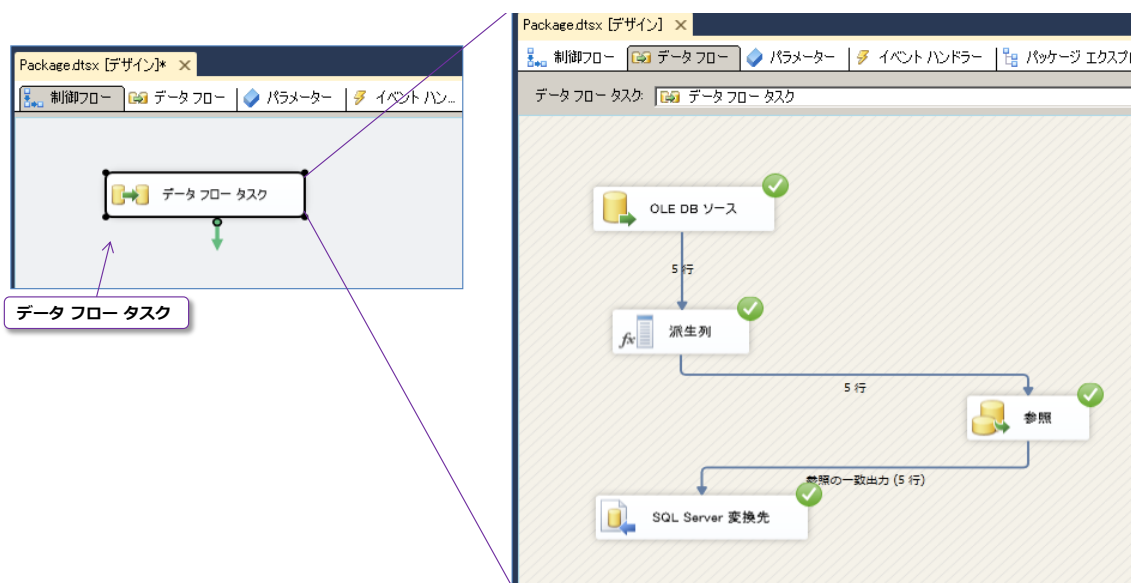
### ➡ データ変換の概要

この STEP では、SSIS デザイナーを利用して、次のようなデータ変換を行うパッケージ（Excel 2010 ファイルから SQL Server へのデータ転送時に変換処理を追加したパッケージ）を作成します。



### ➡ 作成するパッケージ (.dtsx)

作成するパッケージは、次のような構成になります。



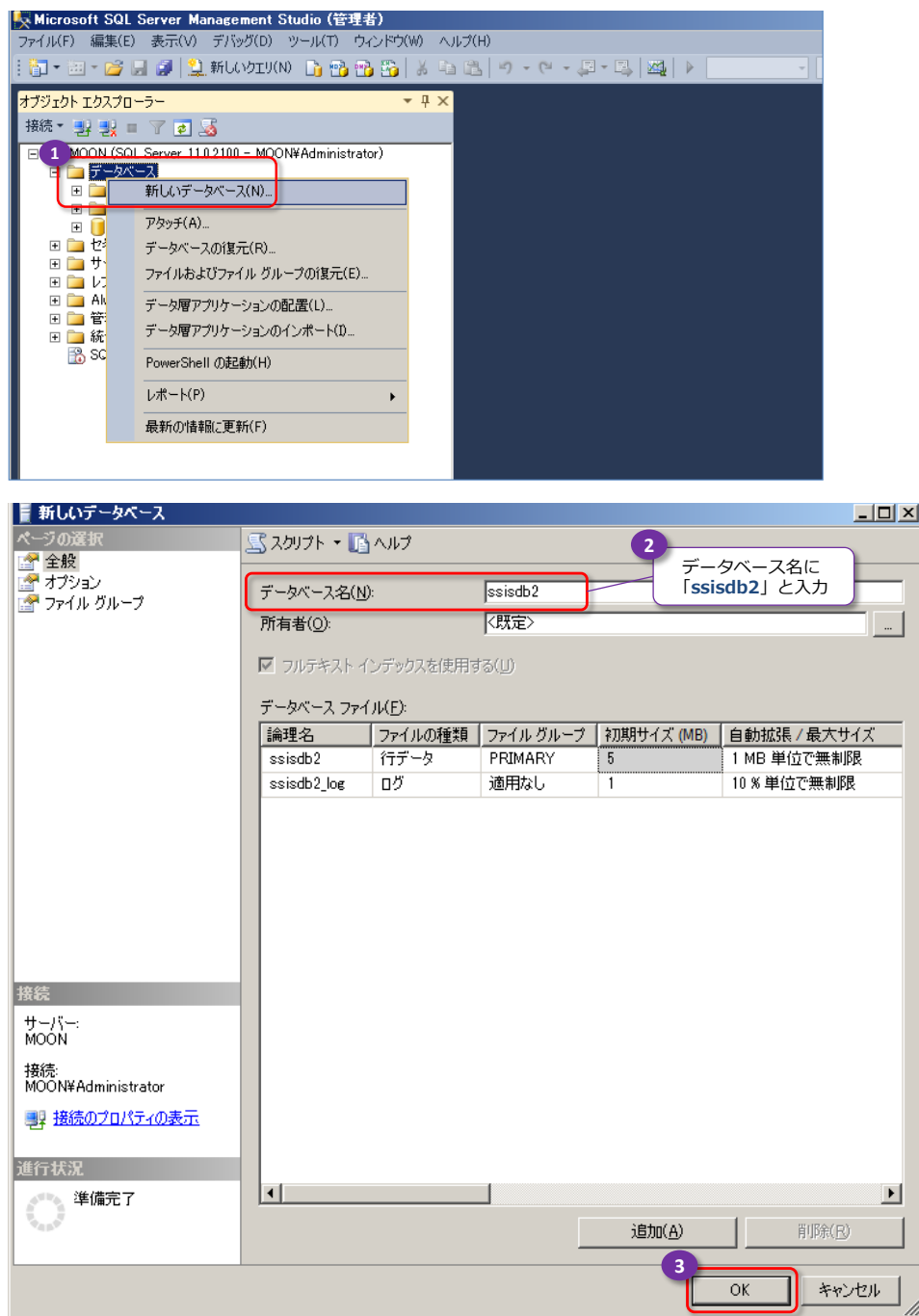


## 3.2 データベースの作成

### ➡ 転送先となるデータベースの作成「ssisdb2」

この STEP では、SSIS デザイナーを使用して、Excel データを SQL Server 内へ取り込んでいくので、まずは、取り込み先（転送先）となるデータベース「ssisdb2」を作成します。

次のように Management Studio で [データベース] フォルダを右クリックして、[新しいデータベース] をクリックし、「ssisdb2」という名前のデータベースを作成します。

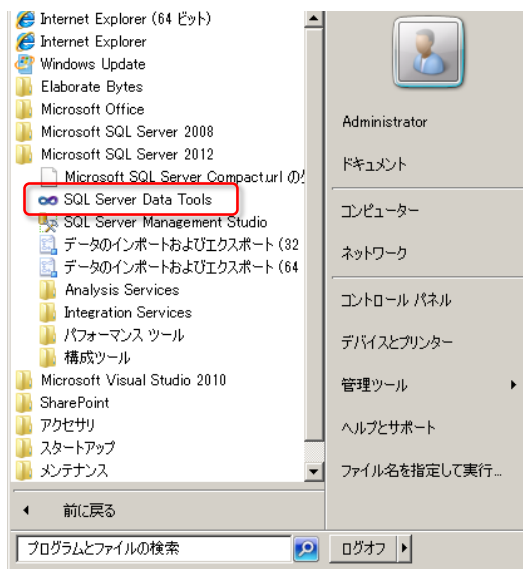


### 3.3 SSIS デザイナーの起動

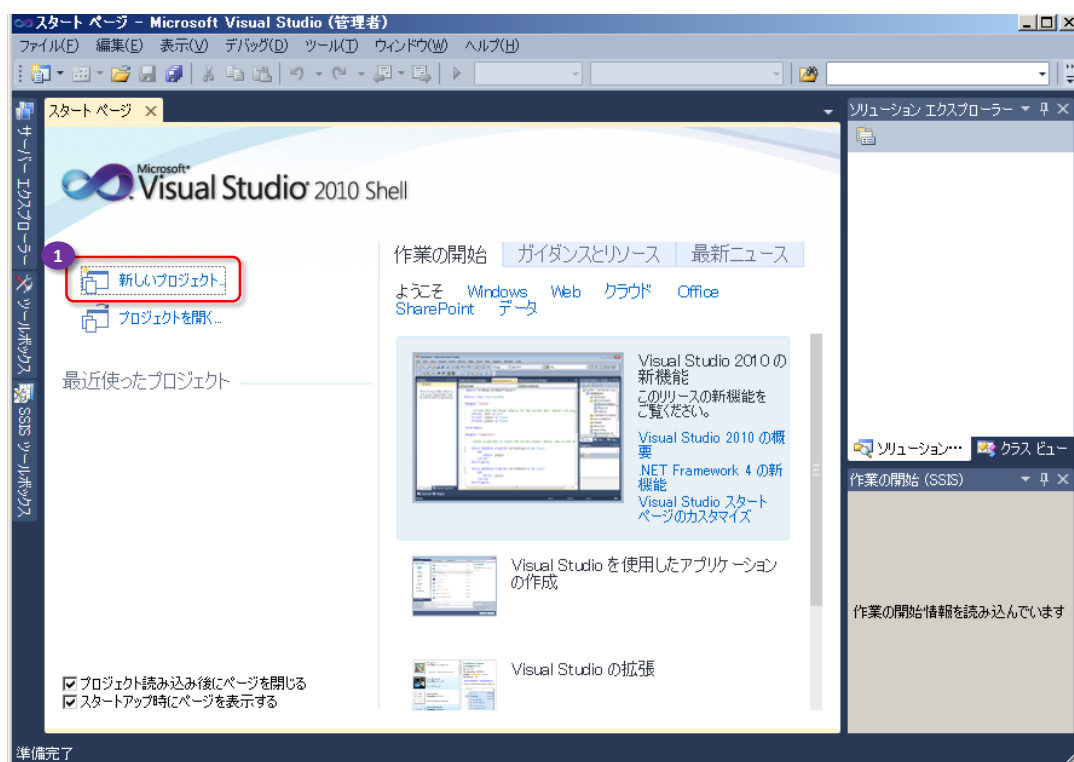
#### ➡ SSIS デザイナーの起動

次に、パッケージの作成ツール「**SSIS デザイナー**」を起動します。

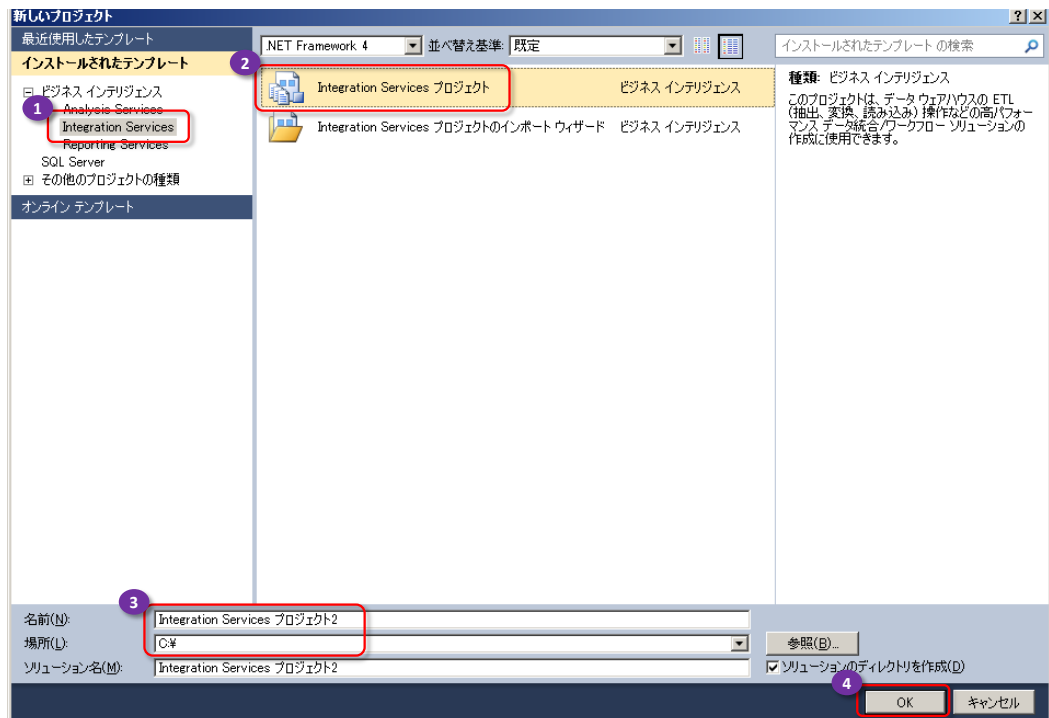
1. SSIS デザイナーを起動するには、[スタート]メニューの[すべてのプログラムから]、[Microsoft SQL Server 2012] の [SQL Server Data Tools] をクリックします。



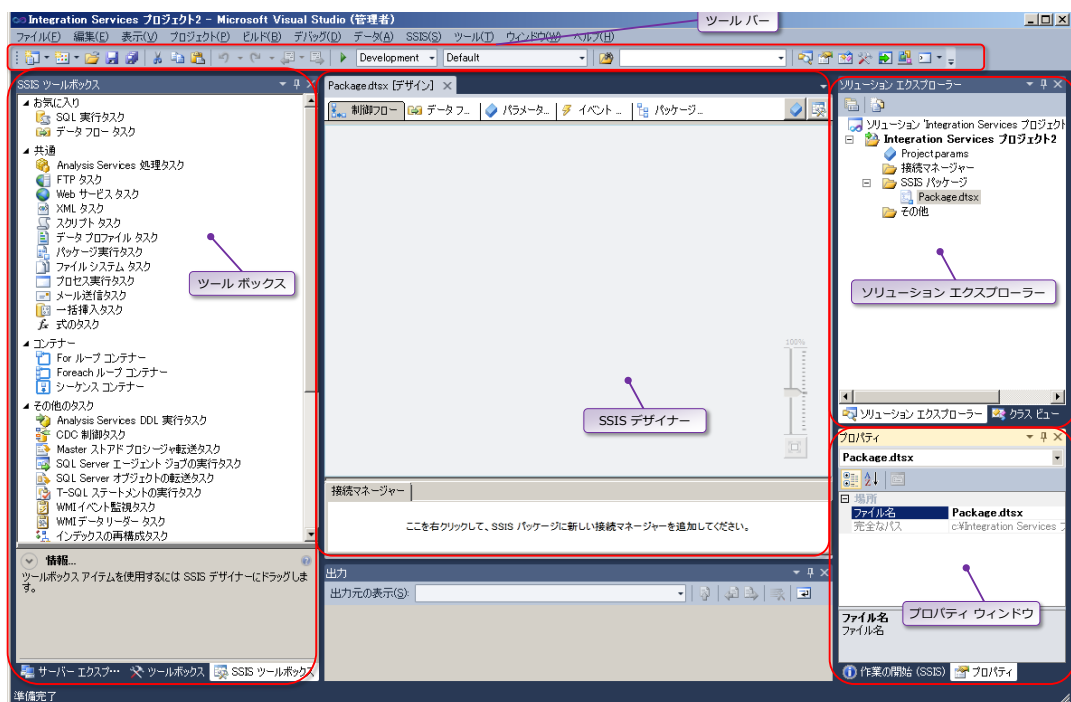
2. SQL Server Data Tools (Visual Studio 2010) が起動したら、[スタート ページ] の [新しいプロジェクト] をクリックして、新しいプロジェクトを作成します。



3. 「新しいプロジェクト」ダイアログでは、「インストールされたテンプレート」で「ビジネス インテリジェンス」の「Integration Services」を選択して、「Integration Services プロジェクト」を選択します。



「名前」へ任意のプロジェクト名（画面は **Integration Services プロジェクト 2**）、「場所」へ任意の保存場所（画面は **C:\**）へ設定して、「OK」ボタンをクリックします。これにより、Integration Services プロジェクトが作成されて、次のように画面が表示されます。

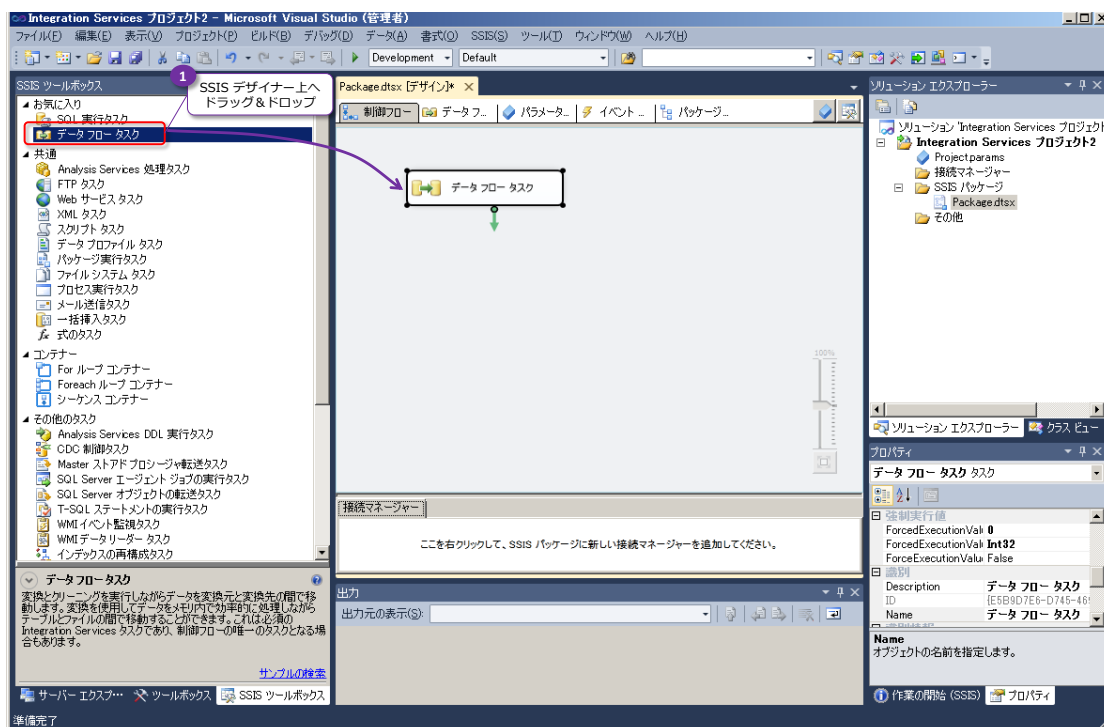


## 3.4 データ フロー タスクによるデータの転送

### ➡ データ フロー タスクによるデータの転送

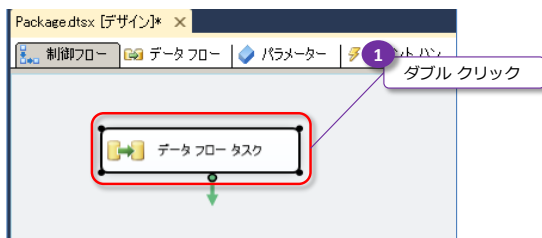
まずは、データ変換を伴わない、単純なデータ転送を行うパッケージを作成してみましょう。データ転送は、[データ フロー タスク] を利用して行うことができます

1. データ フロー タスクを追加するには、次のように SSIS ツールボックスから [お気に入り] カテゴリの [データ フロー タスク] を選択し、SSIS デザイナー上へドラッグ&ドロップします。

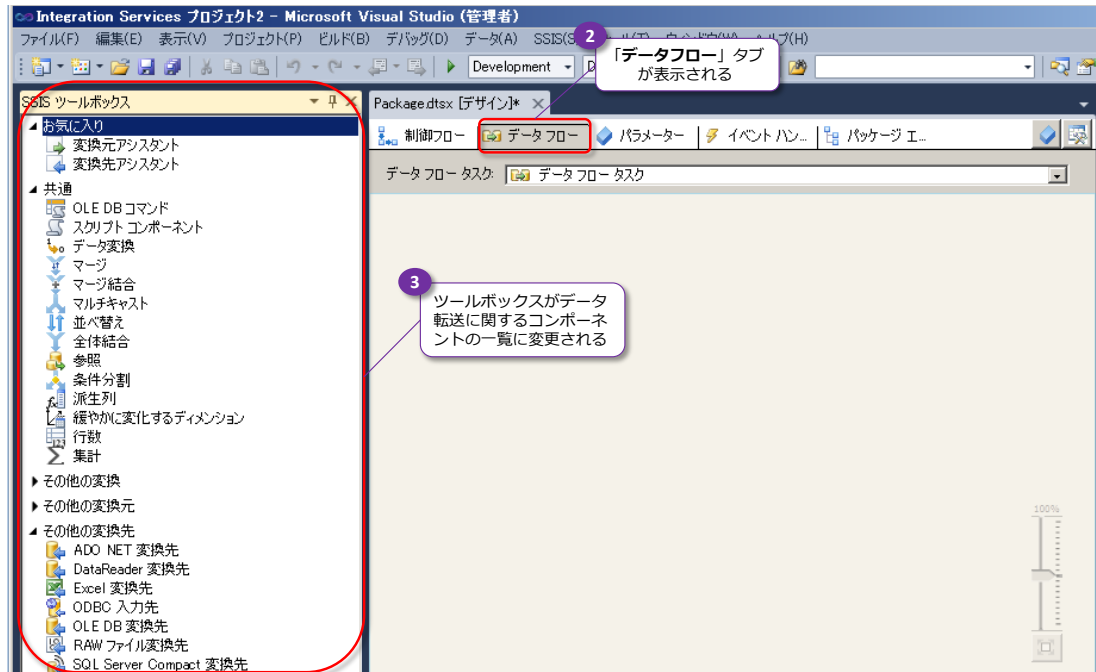


これにより、[データ フロー タスク] が、SSIS デザイナー上に配置されます。

2. 次に、データ転送の詳細（転送元や転送先など）を設定するために、次のように [データ フロー タスク] をダブル クリックします。

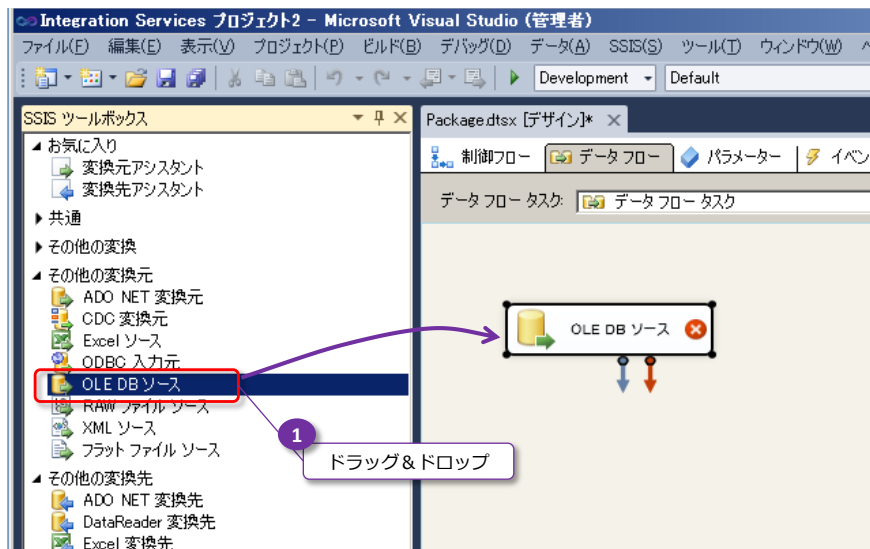


これにより、次のように [データ フロー] タブが表示されて、SSIS ツールボックスがデータ転送に関するコンポーネントの一覧へ変更されます。



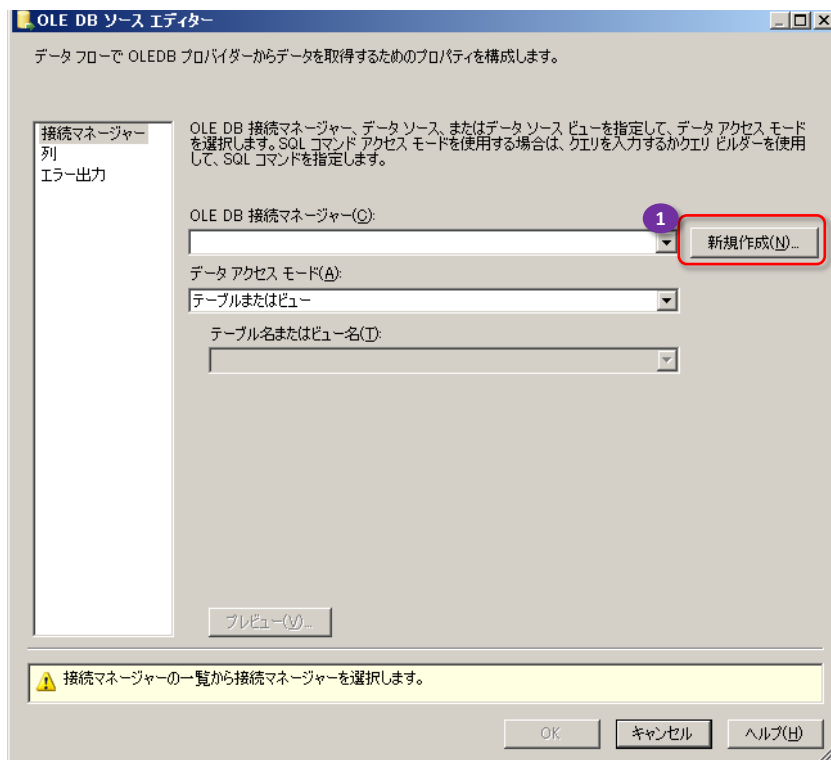
## ➡ 転送元データの指定

3. 転送元（ソース）データとして Excel ファイルを指定するには、次のように SSIS ツールボックスの **「その他の変換元」** カテゴリの中から、**「OLE DB ソース」** を SSIS デザイナー上へドラッグ&ドロップします。

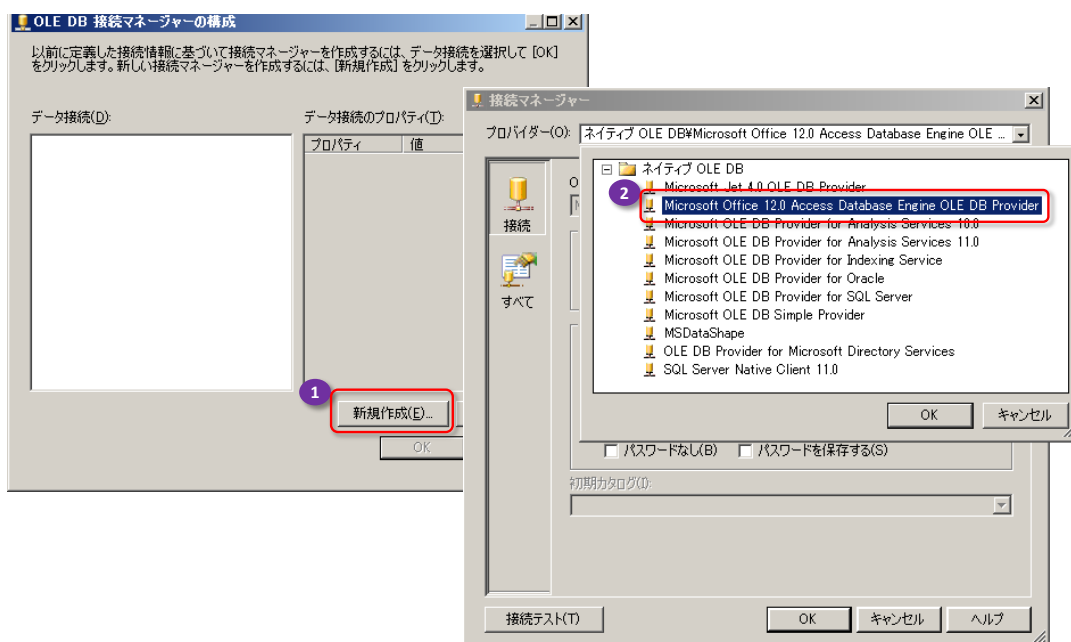


続いて、配置した **「OLE DB ソース」** をダブル クリックします。

4. これにより、**「OLE DB ソース エディター」** ダイアログが表示されるので、転送元の Excel ファイルを指定するために **「新規作成」** ボタンをクリックします。



5. 「OLE DB 接続マネージャーの構成」ダイアログが表示されたら、「新規作成」ボタンをクリックします。

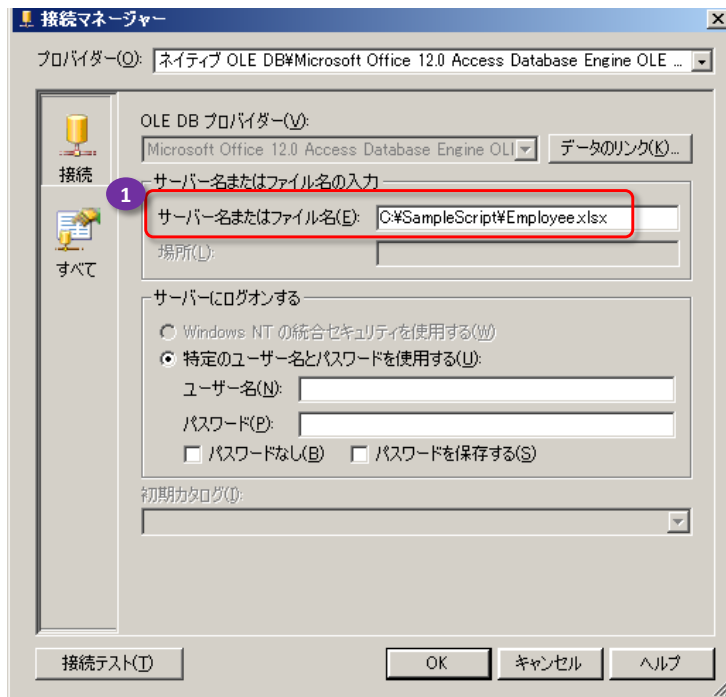


「接続マネージャー」ダイアログが表示されるので、「プロバイダー」で「Microsoft Office 12.0 Access Database Engine OLE DB Provider」を選択します。

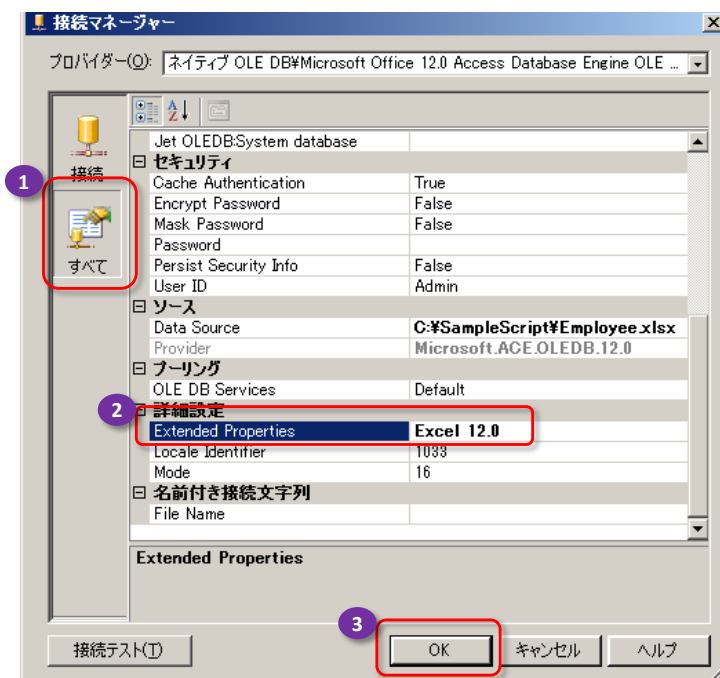
#### Note : 64-bit マシンの場合

64-bit (X64) マシンの場合は、前述の 32-bit 版の「Access データベース エンジン 2010 再頒布可能コンポーネント」をインストールしておかないと「Microsoft Office 12.0 Access Database Engine OLE DB Provider」を選択することができません。インストール手順については、Step 2.5 を参照してください。

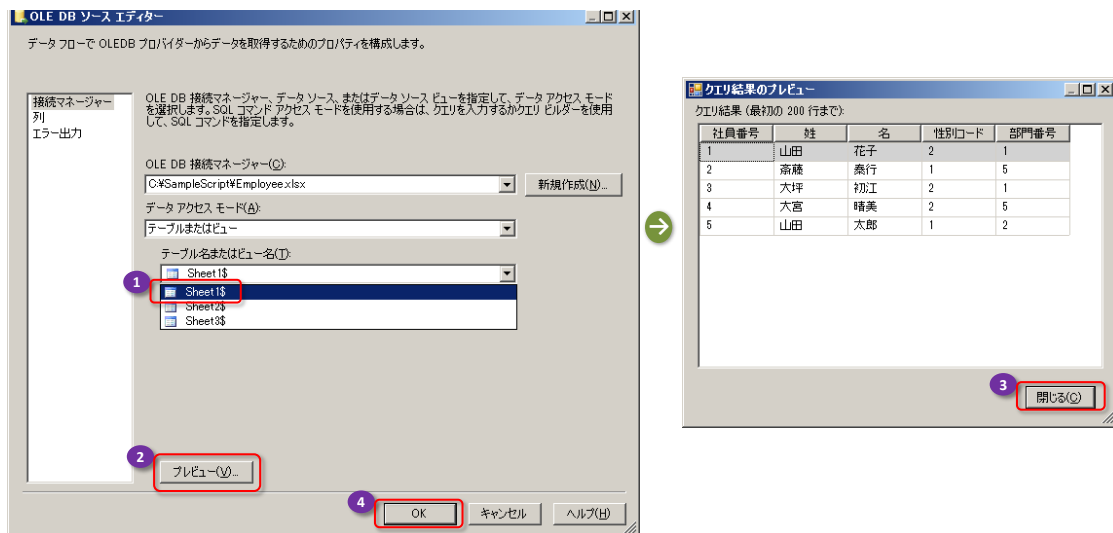
6. 続いて、[サーバー名またはファイル名] ヘサンプル スクリプト内の「Employee.xlsx」ファイルへのパスを入力します。



7. 次に、[すべて] ページをクリックして開き、[Extended Properties] ヘ「Excel 12.0」と入力し、[OK] ボタンをクリックします（Excel 12.0 は Excel 2010 の内部バージョン番号です）。



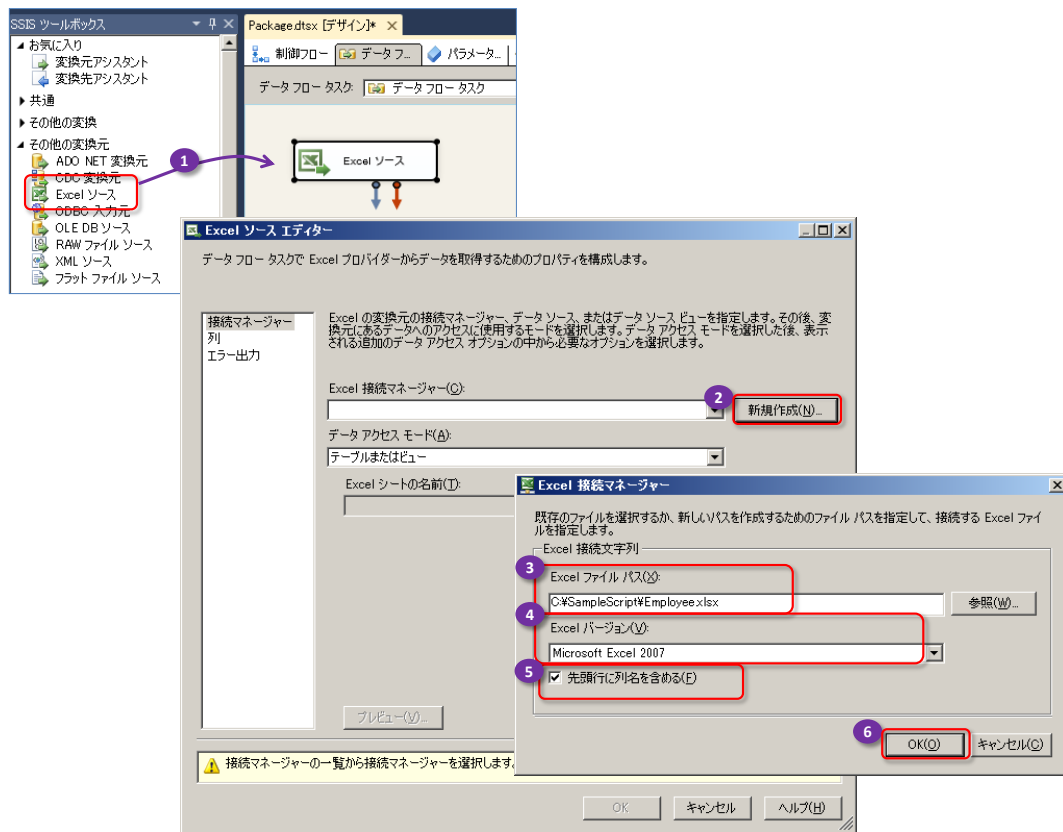
8. [OLE DB ソース エディター] ダイアログへ戻ったら、次のように [テーブル名またはビュー] で、転送元となるデータが格納されている Excel のワーク シートを選択します（ここで表示されるシート名は、実際のワーク シート名に「\$」マークをつけたものになります）。



今回のデータは、**Sheet1** へ入れているので、「**Sheet1\$**」を選択して、[**プレビュー**] ボタンをクリックします。これにより、[**クエリ結果のプレビュー**] ダイアログが表示されて、転送元のデータを確認することができます。確認後、[**閉じる**] ボタンをクリックして、ダイアログを閉じます。

#### Note : 「Excel ソース」コンポーネントを利用する場合

本文の手順では、[**OLE DB ソース**] を利用しましたが、[**Excel ソース**] コンポーネントを利用しても、Excel データを取り込むことができます。[**Excel ソース**] コンポーネントを利用する場合は、次のように操作します。

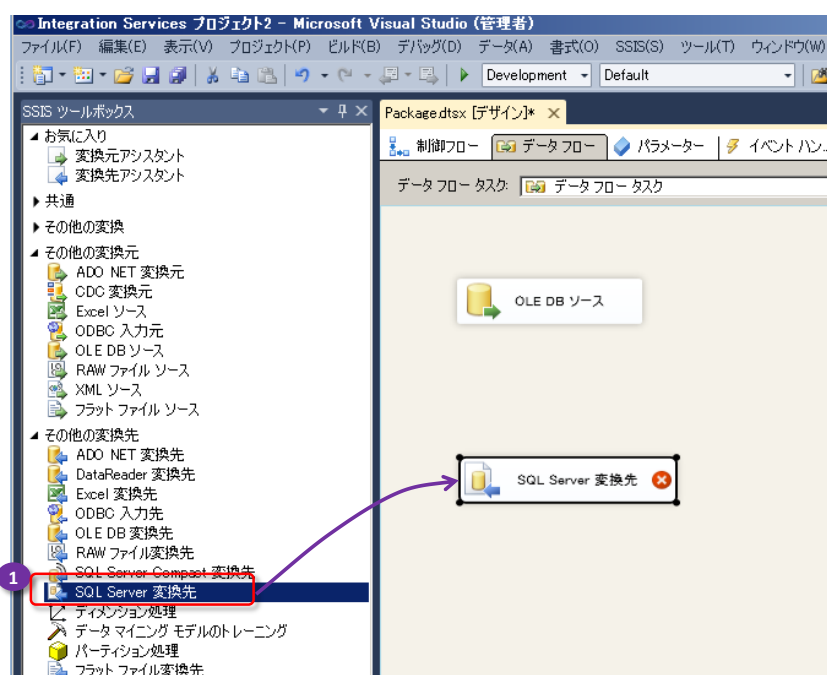




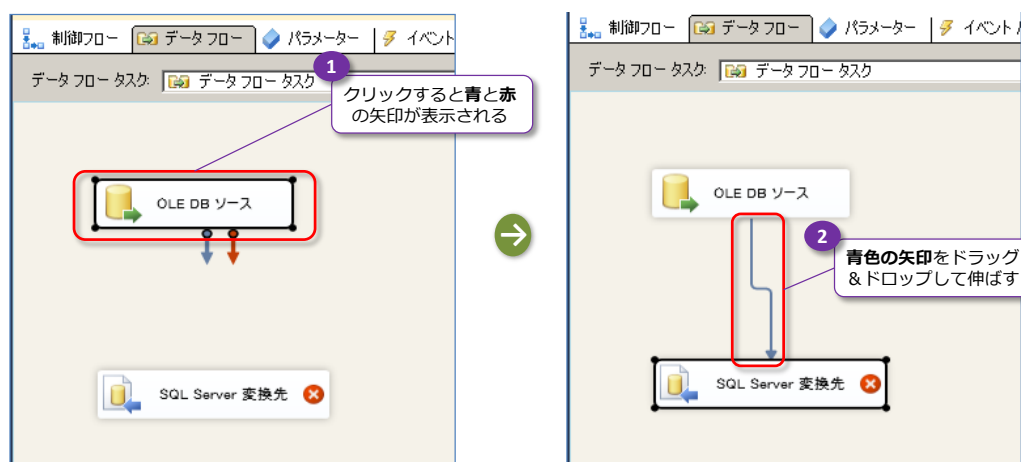
Steo 2.4 のインポート/エクスポート ウィザードで [データ ソースの選択] ページで「Microsoft Excel」を選択したときと同様の操作で、[Excel バージョン] で「Microsoft Excel 2007」を選択することで、Excel 2007 以降の .xlsx ファイルをインポートすることができます。

## ➡ 転送先データベースの指定

- 次に、転送先となる SQL Server 上のデータベースを指定するために、次のように SSIS ツールボックスの [その他の変換先] カテゴリから、[SQL Server 変換先] を SSIS デザイナー上へドラッグ&ドロップします。



- 次に、転送元と転送先を関連付けるために、SSIS デザイナー上にある [OLE DB ソース] をクリックして「青」と「赤」の矢印を表示し、「青の矢印」を [SQL Server 変換先] まで、ドラッグ&ドロップして伸ばします。

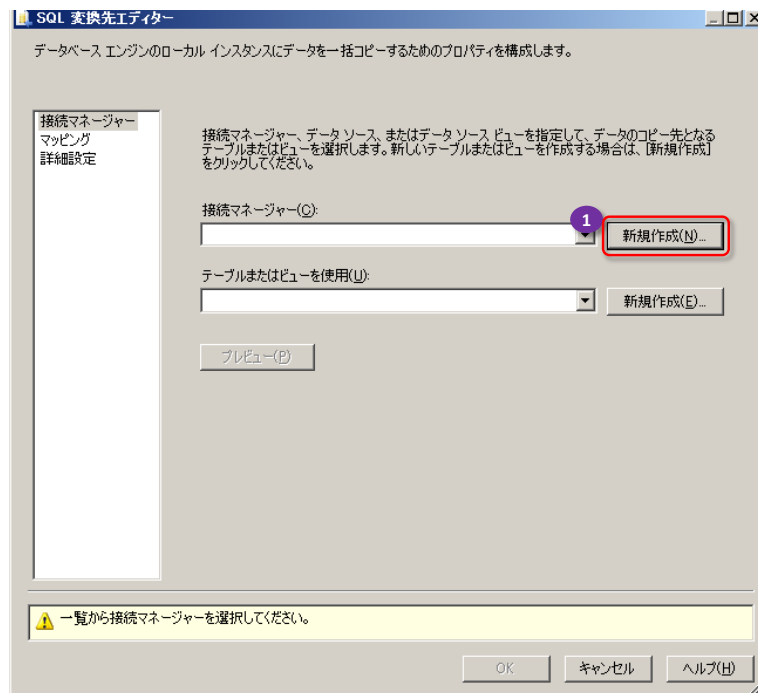


### Note : 「青」は成功時、「赤」は失敗（エラー）時の処理

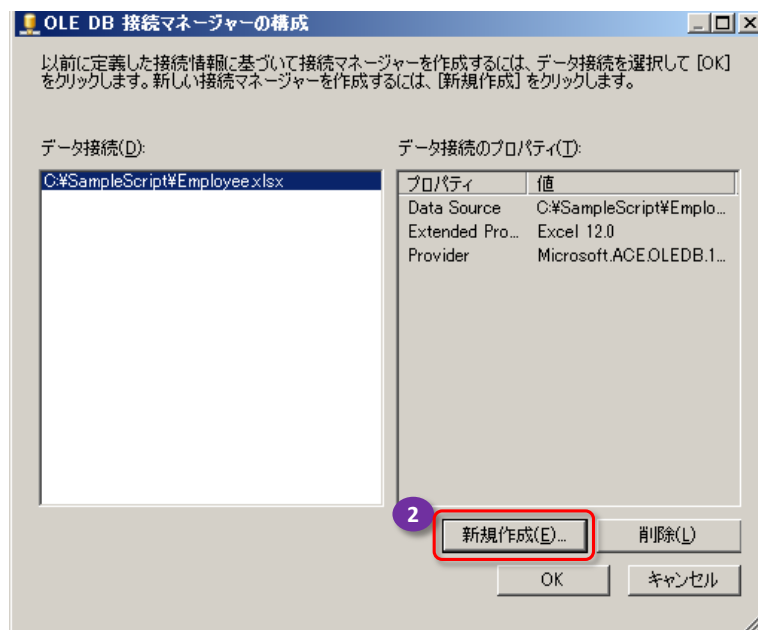
青の矢印は "成功時" の処理、赤の矢印は "失敗時"（エラー発生時）の処理の流れを意味します。エラー時の処理については、本自習書シリーズの「Integration Services 応用」編で詳しく説明しています。

11. 次に、転送先の詳細設定を行うために、[SQL Server 変換先] をダブルクリックします。

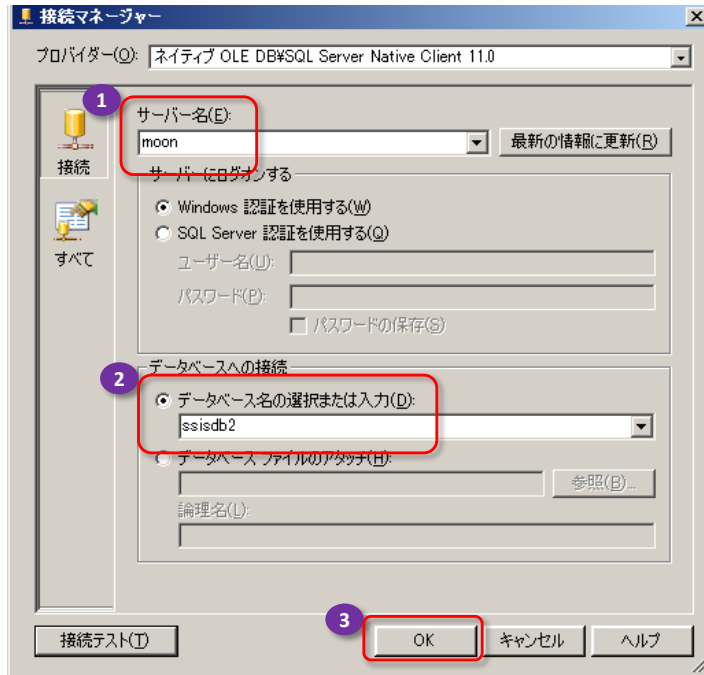
これにより、[SQL 変換先エディター] ダイアログが表示されるので、次のように [接続マネージャー] ページで [新規作成] ボタンをクリックします。



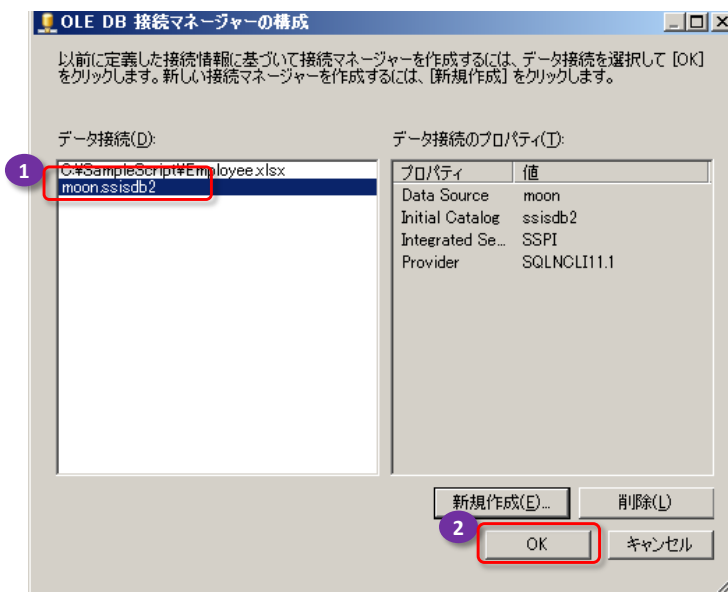
12. [OLE DB 接続マネージャーの構成] ダイアログが表示されたら、さらに [新規作成] ボタンをクリックします。



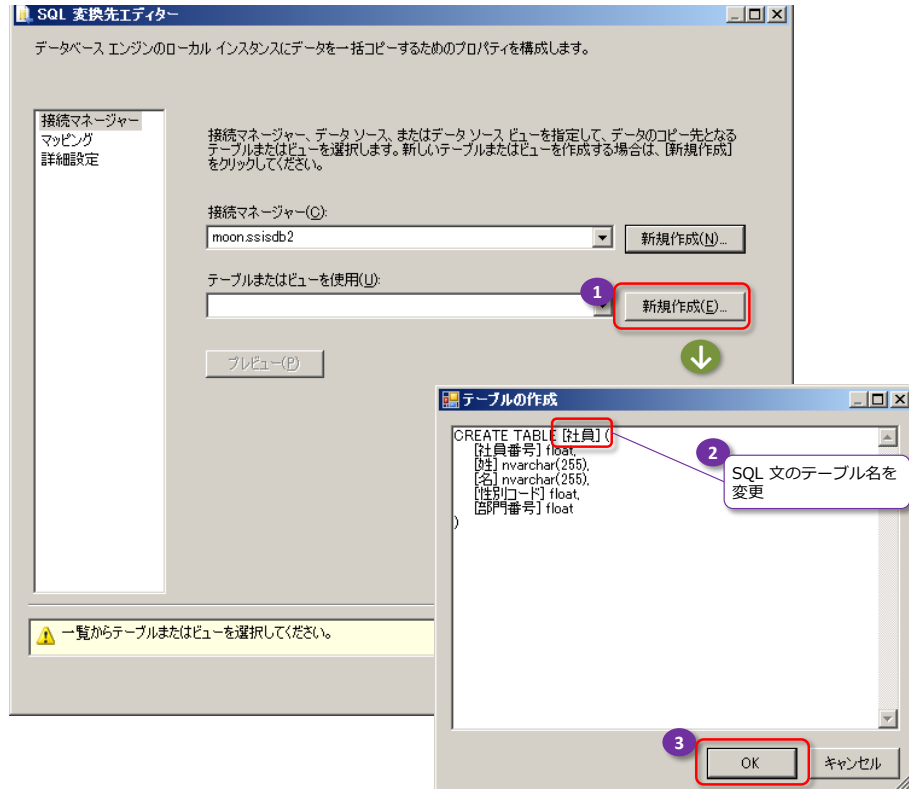
13. [接続マネージャー] ダイアログでは、次のように [サーバー名] へ転送先となる SQL Server の名前を入力して、[データベース名の選択または入力] で「ssisdb2」を選択し、[OK] ボタンをクリックします。



14. [OLE DB 接続マネージャーの構成] ダイアログへ戻ったら、[データ接続] で、「サーバー名.ssisdb2」が選択されていることを確認して、[OK] ボタンをクリックします。

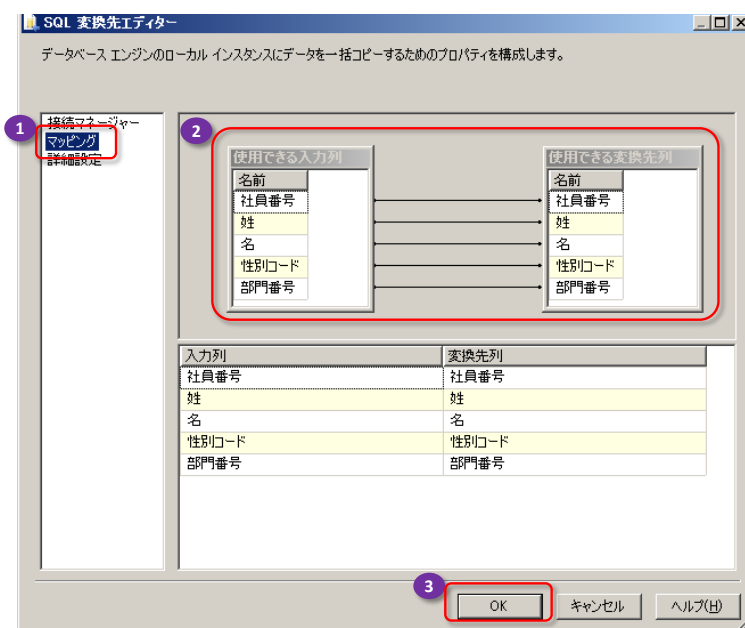


15. [SQL 変換先エディター] ダイアログへ戻ったら、[テーブルまたはビューを使用] で [新規作成] ボタンをクリックします。



これにより、「**テーブルの作成**」ダイアログが表示されて、転送元のデータをもとに、SQL Server 上へテーブルを作成できるようになります。ここで表示される **CREATE TABLE** ステートメントのテーブル名を「**社員**」へ変更して、「**OK**」ボタンをクリックします。これにより、「**社員**」テーブルを **ssisdb2** データベース内へ作成することができます（「**OK**」ボタンをクリックしたときに実際にテーブルが作成されます）。

16. 続いて、次のように「**マッピング**」ページをクリックすると、転送元のデータの各列と、転送先のデータの各列とのマッピング（対応）を設定することができます。

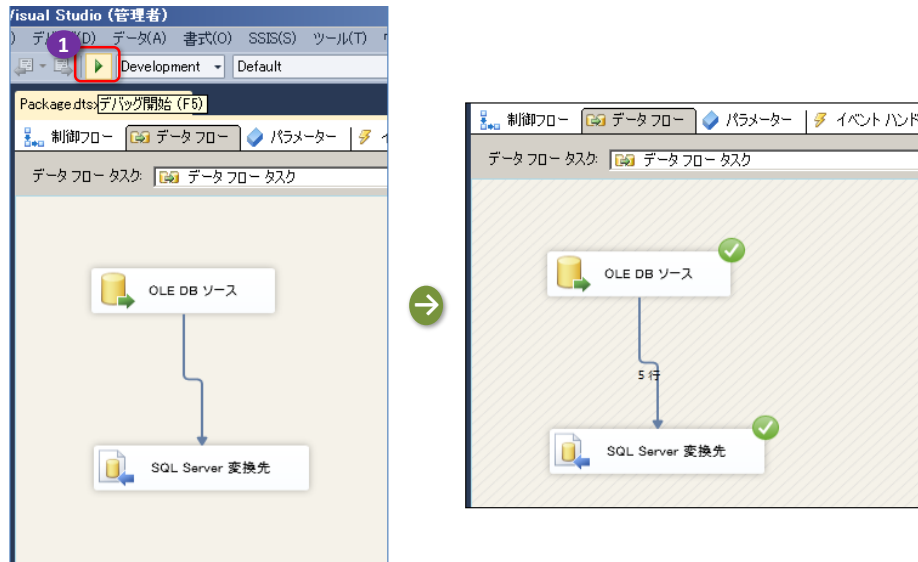


「**使用できる入力列**」と「**使用できる変換先列**」で、同じ名前の列が割り当てられていること

を確認して、[OK] ボタンをクリックします。

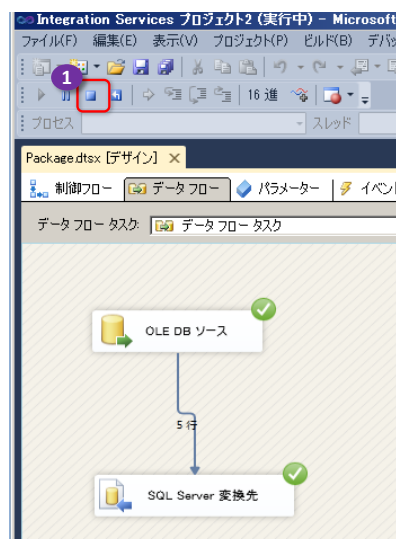
## ➡ データ転送の実行

17. ここまでの設定を確認するために、ツールバーの[デバッグ開始] ボタンをクリックして、データの転送を実行します。



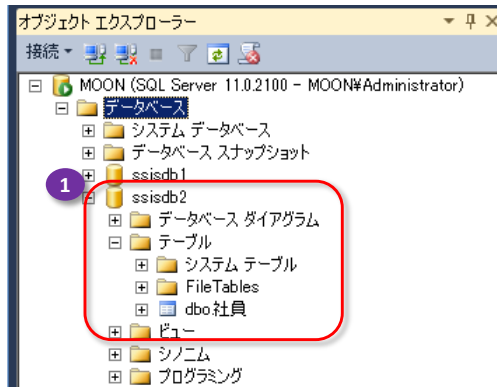
配置されているオブジェクトすべてに緑のチェックマークが付くと、データの転送が成功です。

18. 転送が成功したら、ツールバーの[デバッグの停止] ボタンをクリックして、デバッグを終了します。



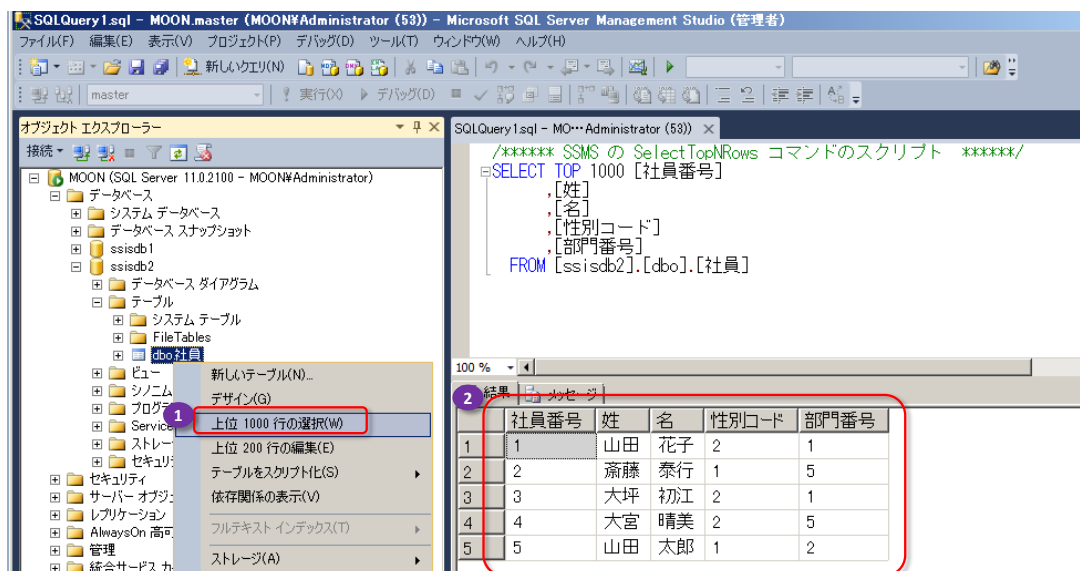
## ➡ 転送されたデータの確認

19. 転送されたデータを確認するには、Management Studio から「ssisdb2」データベースの[テーブル] フォルダを展開します。



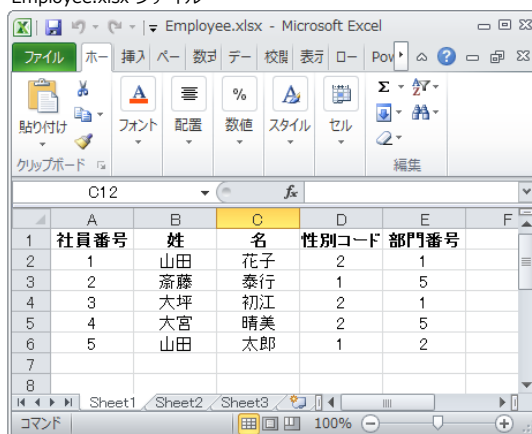
テーブルの一覧に「社員」という名前のテーブルが作成されていることを確認できます（表示されない場合は、[テーブル] フォルダを右クリックして、[最新の情報に更新] をクリックします）。

20. 続いて、[社員] テーブルを右クリックして、[上位 1000 行の選択] をクリックします。



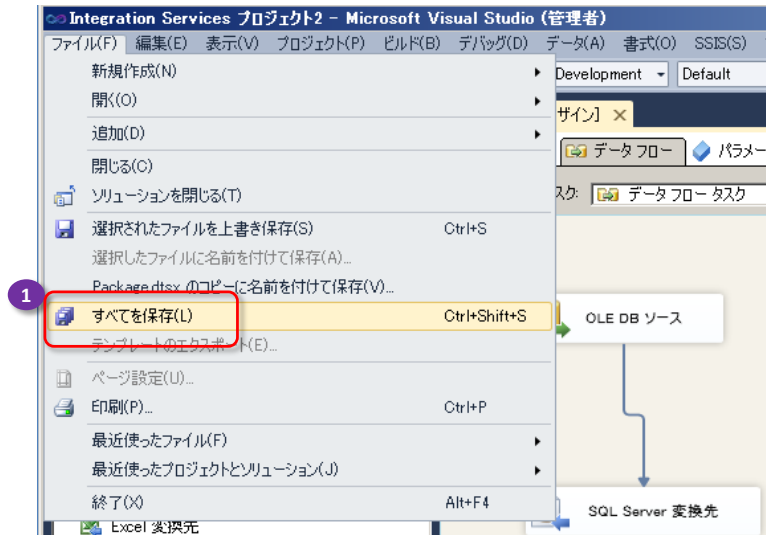
表示された社員テーブルのデータが、Excel ファイル内のデータ（以下）と同じであることを確認します。

Employee.xlsx ファイル



## ➡ プロジェクトの保存

21. ここまでの手順を保存するために、[ファイル] メニューの [すべてを保存] をクリックして、プロジェクトを保存します。

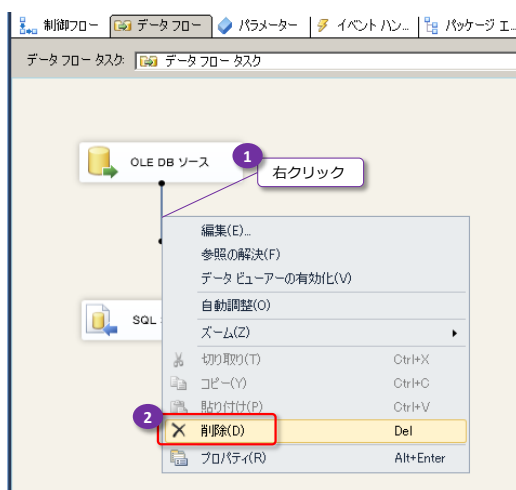


## 3.5 派生列コンポーネントによるデータ変換

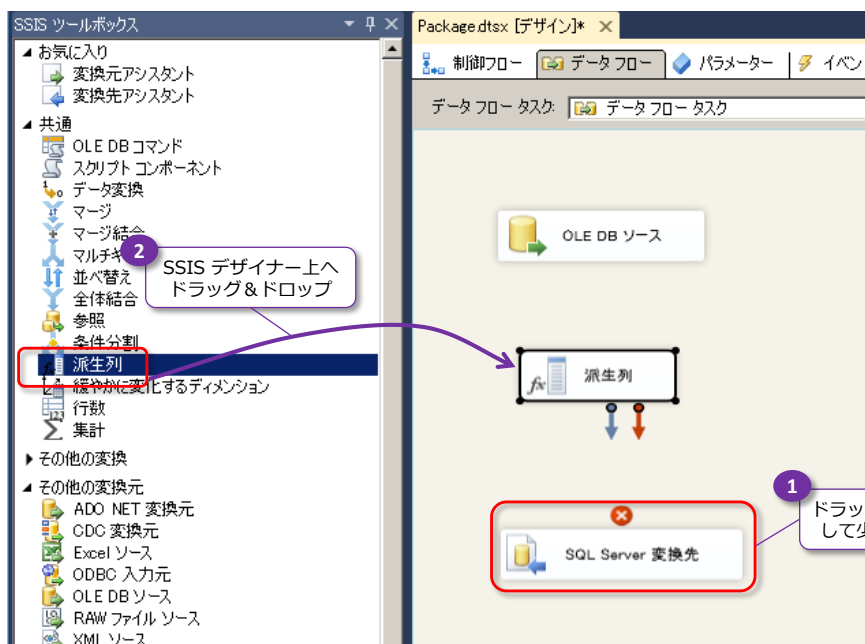
### ➡ 派生列コンポーネントによるデータ変換

次に、単純なデータ転送（コピー）だけでなく、データ変換を追加していきましょう。ここでは、「姓」と「名」列から「氏名」列へ変換し、「性別コード」列からは、「1」を「男性」へ、「2」を「女性」へと変換するような「性別」列を作成します。このようなデータ変換は、派生列コンポーネントを利用して簡単に行うことができます。

1. まずは、前の手順で設定した「OLE DB ソース」から、「SQL Server 変換先」へ向かって出ている「青色の矢印」を右クリックして、「削除」をクリックします。これにより、矢印が削除されます。



2. 次に、データ変換を行うためのコンポーネントとなる「派生列」を「OLE DB ソース」と「SQL Server 変換先」の間へ配置します。このコンポーネントは、SSIS ツールボックスの「共通」カテゴリにあります。

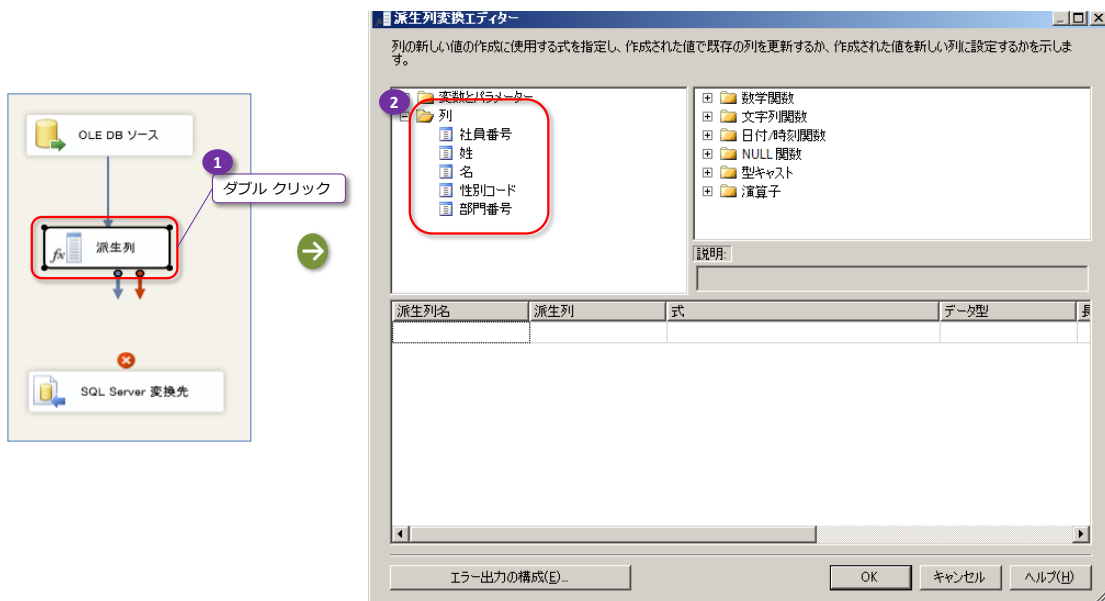




- 次に、転送元と派生列を関連付けるために、SSIS デザイナー上にある「OLE DB ソース」をクリックして "青色の矢印" を表示し、この矢印を「派生列」まで、ドラッグ&ドロップして伸ばします。

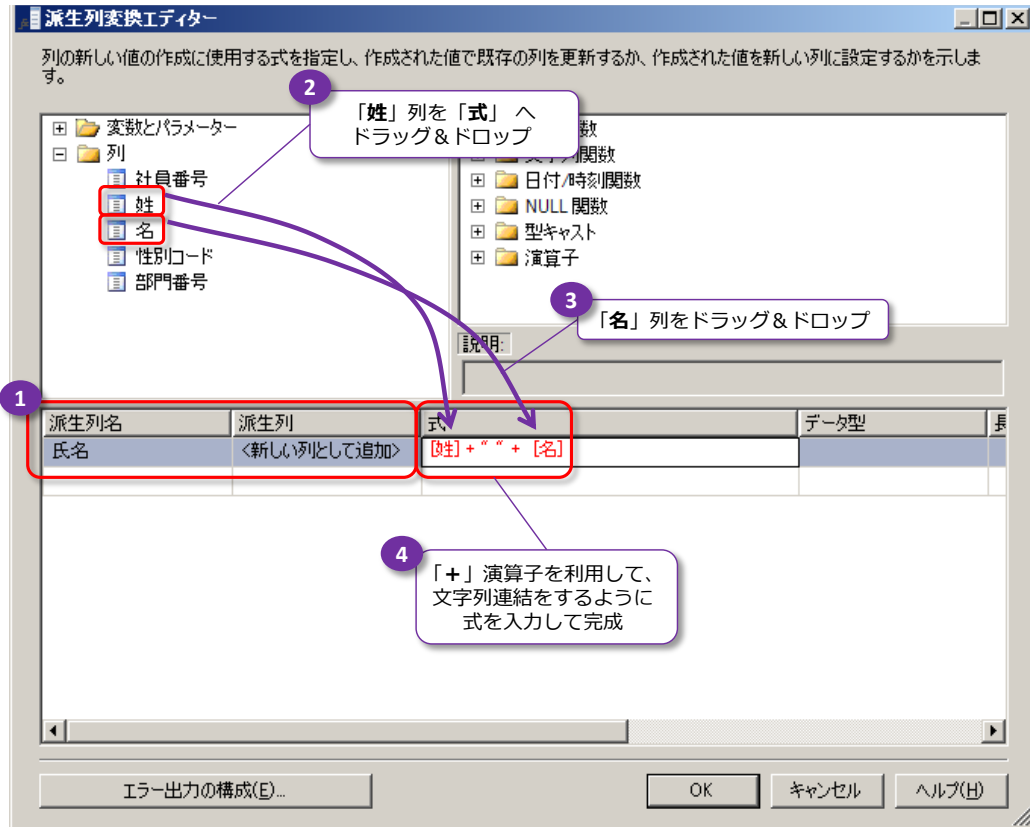


- 次に、データ変換を定義するために、次のように「派生列」コンポーネントをダブル クリックします。



これにより、「派生列変換エディター」ダイアログが表示されるので、「列」フォルダーを展開します。これで、転送元の列（社員番号や姓、名など）が一覧されるようになります。

- 今回は、「姓」列と「名」列をもとに「氏名」列を作成するので、次のように「派生列名」に「氏名」と入力して、「派生列」が「新しい列として追加」になっていることを確認します。



次に、「姓」と「名」列を文字列連結するために、[式] へ、「姓」と「名」列をドラッグ&ドロップして配置し、次のように式を入力します。

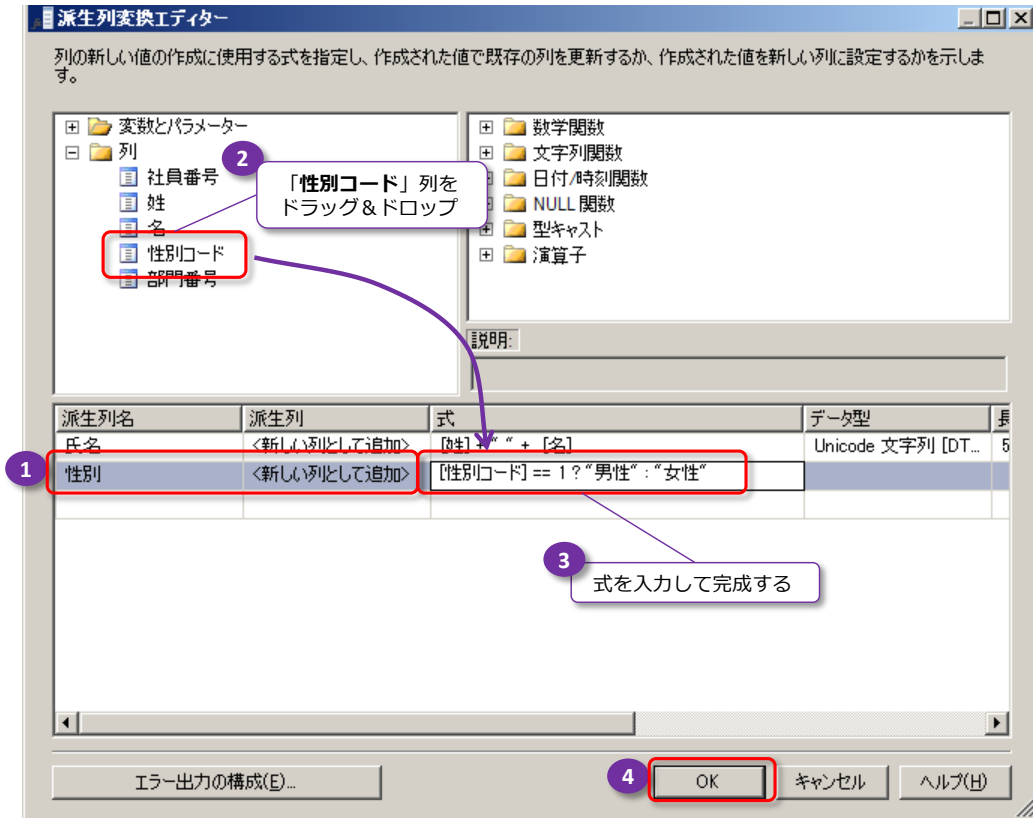
[姓] + " " + [名]

#### Note：文字列連結演算子の「+」

Integration Services では、文字列の連結に「+」演算子を利用し、文字列は「"」（二重引用符）で囲みます。したがって、上のように式を記述すると、「姓」列のデータに半角スペースを連結し、さらに「名」列を連結できるようになります。たとえば、姓が「山田」、名が「太郎」なら「山田 太郎」のように連結できます。

このように、派生列コンポーネントを利用すると、既存の列データを元に、新しい列（元データから派生した列）を作成できるようになります。

6. 続いて、「性別コード」列をもとに「性別」列を作成するために、次のように [派生列名] に「性別」と入力して、[派生列] が「新しい列として追加」になっていることを確認します。



次に「性別コード」が「1」の場合は「男性」、「2」の場合は「女性」になるように「性別」列を作成するために、「性別コード」列を[式]ヘドラッグ&ドロップして配置し、次のように式を入力します。

[性別コード] == 1 ? "男性" : "女性"

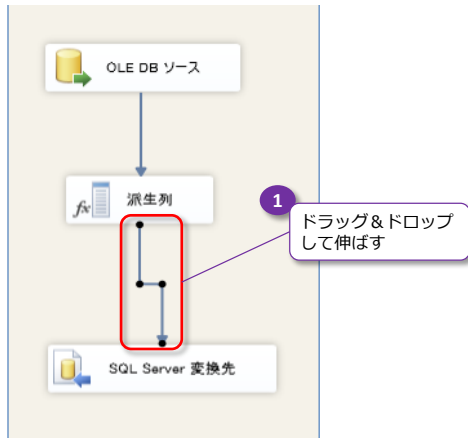
Integration Services では、条件分岐を行うための演算子として、「?」と「:」を利用して、次のように利用します。

条件式 ? 真の場合 : 偽の場合

この演算子は、Access や Reporting Services など利用できる「IIF」関数と同じ効果があります。また、式で利用している「==」演算子は、等価比較のための比較演算子です。したがって、上のように式を記述すると、性別コードが「1」に等しい場合は「男性」、そうでない場合は「女性」という文字列へ変換するという意味になります。

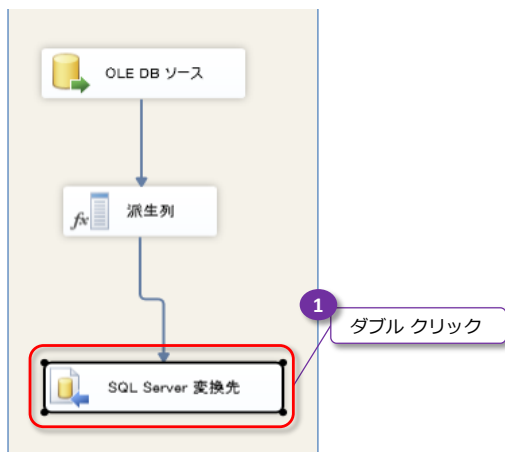
式を入力後、[OK] ボタンをクリックしてダイアログを閉じます。

- 次に、派生列として作成した「氏名」と「性別」列が SQL Server へ転送されるようにします。次のように[派生列]をクリックして "青色の矢印" を表示し、この矢印を[SQL Server 変換先]までドラッグ&ドロップして伸ばします。

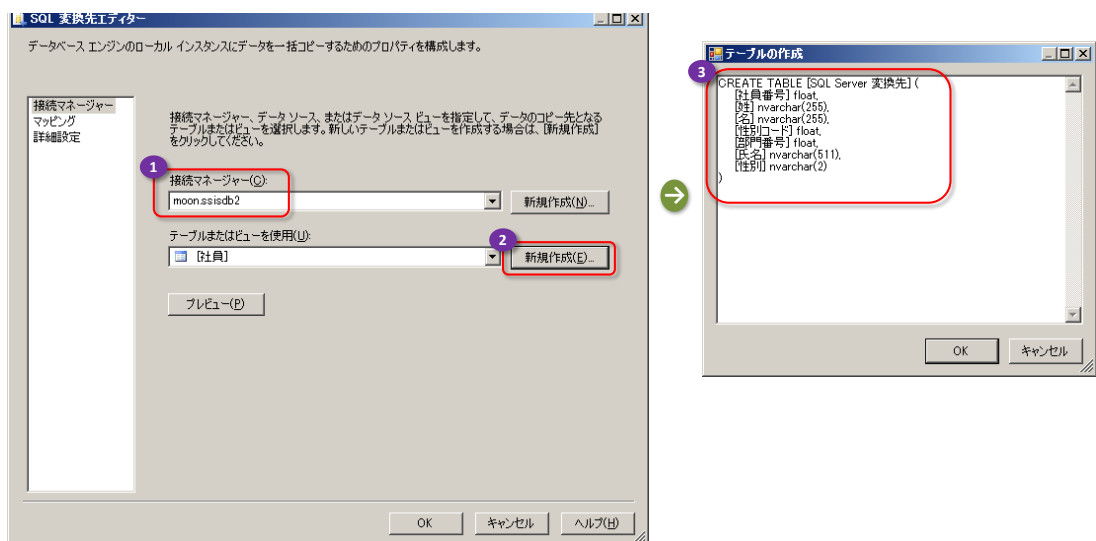


## ➡ 転送先のテーブルの変更

8. 次に、[SQL Server 変換先] をダブル クリックします。



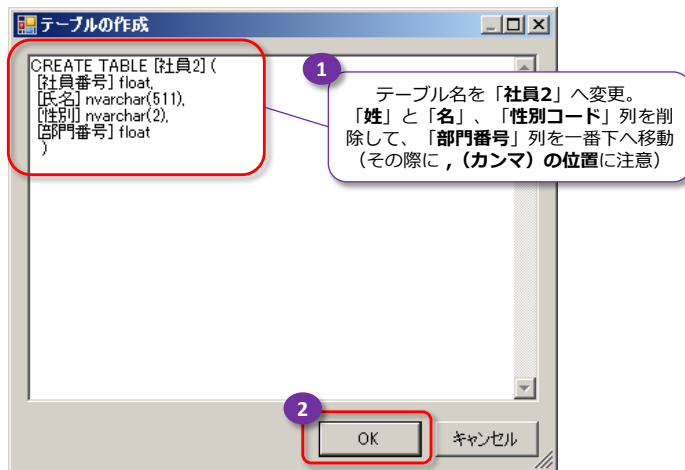
9. [SQL 変換先エディター] ダイアログが表示されたら、[接続マネージャー] で「サーバー名.ssisdb2」が選択されていることを確認します。続いて、[テーブルまたはビューを使用] で [新規作成] ボタンをクリックします。



これにより、[テーブルの作成] ダイアログが表示されて、転送元のデータと派生列コンポーネ

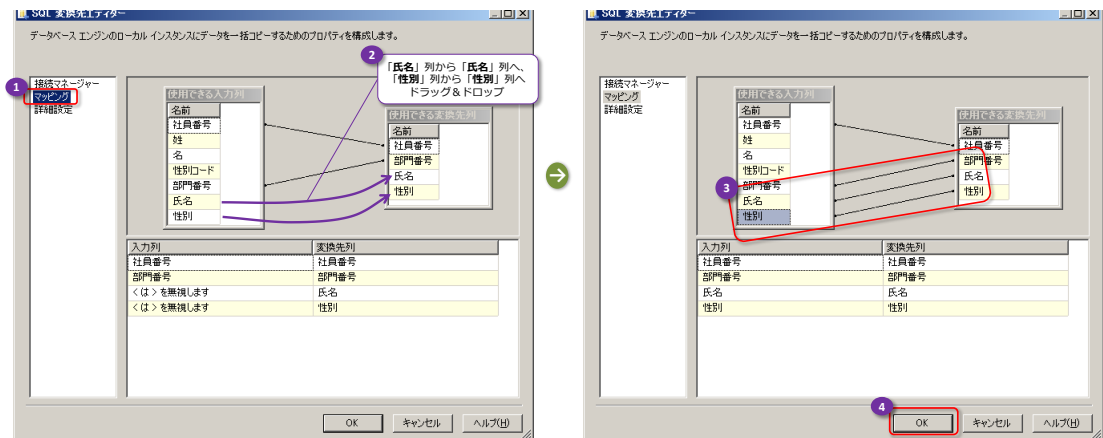
ントをもとに、**CREATE TABLE** ステートメントが自動生成されます。今回は、この SQL ステートメントを次のように変更します。

```
CREATE TABLE [社員2] (  
  [社員番号] float,  
  [氏名] nvarchar(511),  
  [性別] nvarchar(2),  
  [部門番号] float  
)
```



これにより、「社員2」テーブルを **ssisdb2** データベース内へ作成することができます（**[OK]** ボタンをクリックしたときに実際にテーブルが作成されます）。

10. 続いて、次のように **[マッピング]** ページをクリックすると、転送元のデータの各列と転送先のデータの各列とのマッピング（対応）を設定することができます。

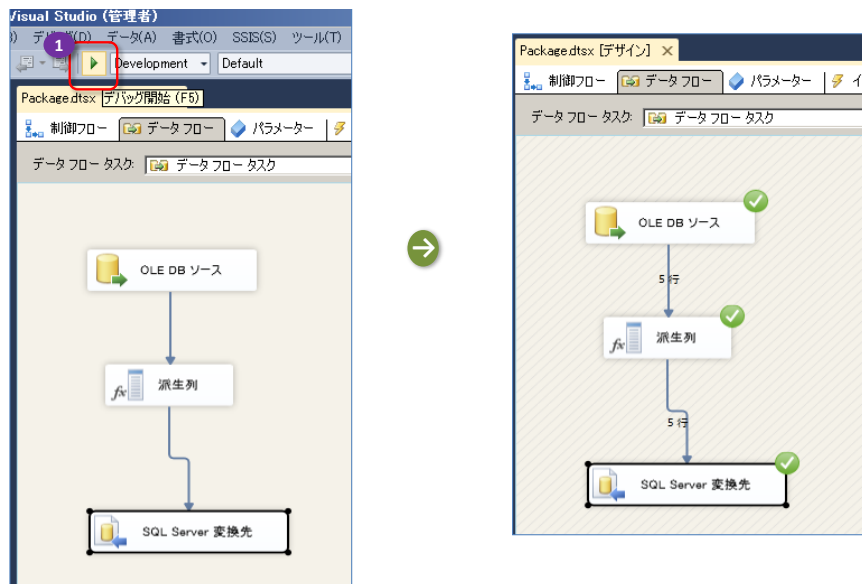


前の手順で作成した「氏名」列がテーブルへ転送されるように、**[使用できる入力列]**の「氏名」列から**[使用できる変換先列]**の「氏名」列へドラッグ&ドロップします。同様に、「性別」列に対しても同じ操作を行います。

これにより、**[使用できる入力列]** から**[使用できる変換先列]** へ 2 本の線（マッピング）が追加され、作成した「氏名」列と「性別」列を、「社員2」テーブルの「氏名」列と「性別」列へそれぞれ割り当てることができます。

## ➡ データ転送の実行

11. ここまでの設定を確認するために、ツールバーの【デバッグ開始】ボタンをクリックして、データ転送を実行します。

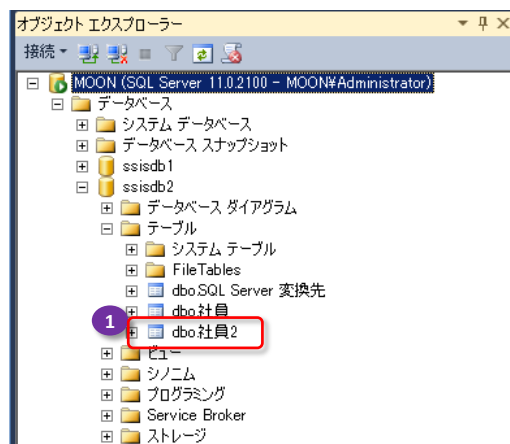


配置されているオブジェクトすべてに緑のチェックマークが付くと、データの転送が成功です。

転送が成功したら、ツールバーの【デバッグの停止】ボタンをクリックして、デバッグを終了します。

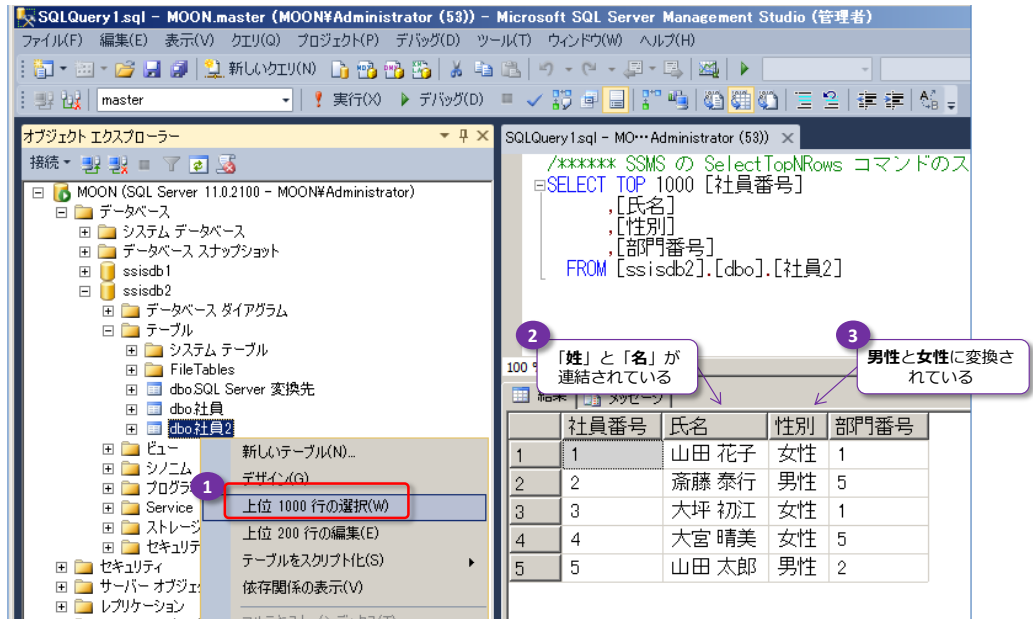
## ➡ 転送されたデータの確認

12. 転送されたデータを確認するには、Management Studio で、「ssisdb2」データベースの【テーブル】フォルダーを展開します。



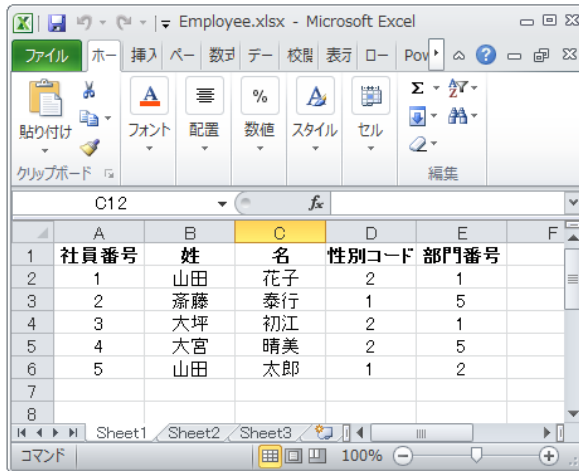
テーブルの一覧に「社員 2」という名前のテーブルが作成されていることを確認できます（表示されない場合は、【最新の情報に更新】をクリックします）。

13. 続いて、【社員 2】テーブルを右クリックして【上位 1000 行の選択】をクリックします。



表示されたデータが、Excel ファイル内のデータ（以下）をもとに変換され、「姓」と「名」列を連結して「氏名」列が、「性別コード」列が変換されて「性別」列が追加されていることを確認できます。

Employee.xlsx ファイル



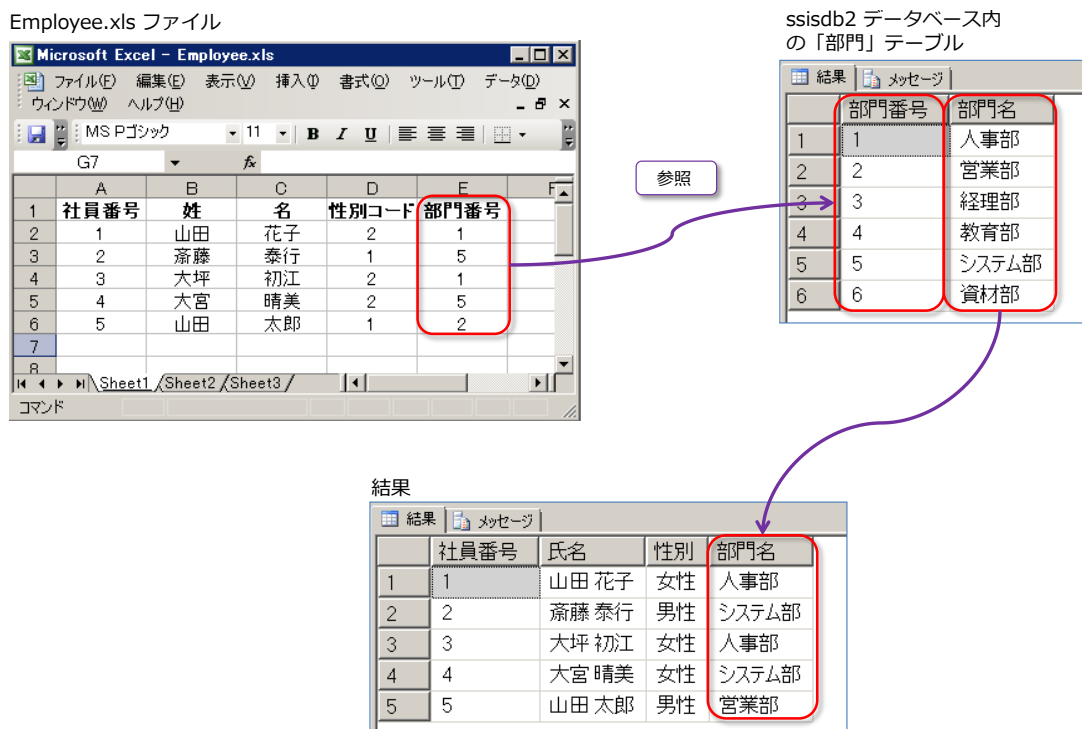
14. 次に、ここまでの手順を保存するために、[ファイル] メニューの [すべてを保存] をクリックして、プロジェクトを保存します。



## 3.6 参照コンポーネントによる別テーブルのデータ取得

### ➡ 参照コンポーネントによる別テーブルのデータ取得

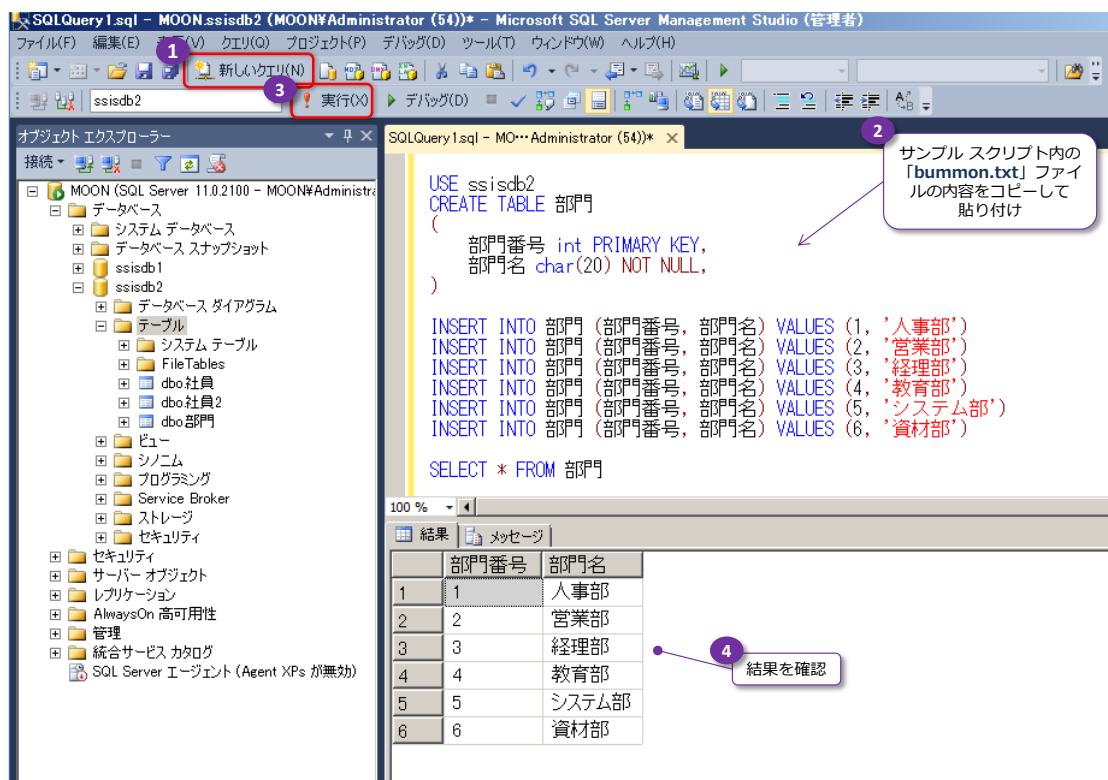
次に、社員データの「部門番号」から、「部門」テーブルにある「部門名」を取得（参照）してデータを転送するようにパッケージを変更します。他のテーブルのデータを参照するには、「参照」(Lookup)コンポーネントを利用します。



### ➡ 「部門」テーブルの準備

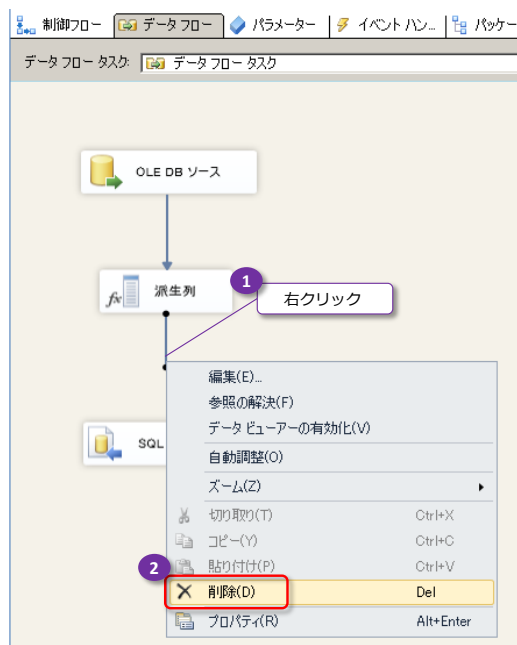
この Step を実施するには、サンプル スクリプト内にある「bumon.txt」ファイルを実行して、「ssisdb2」データベース内に「部門」テーブルを作成しておく必要があります。



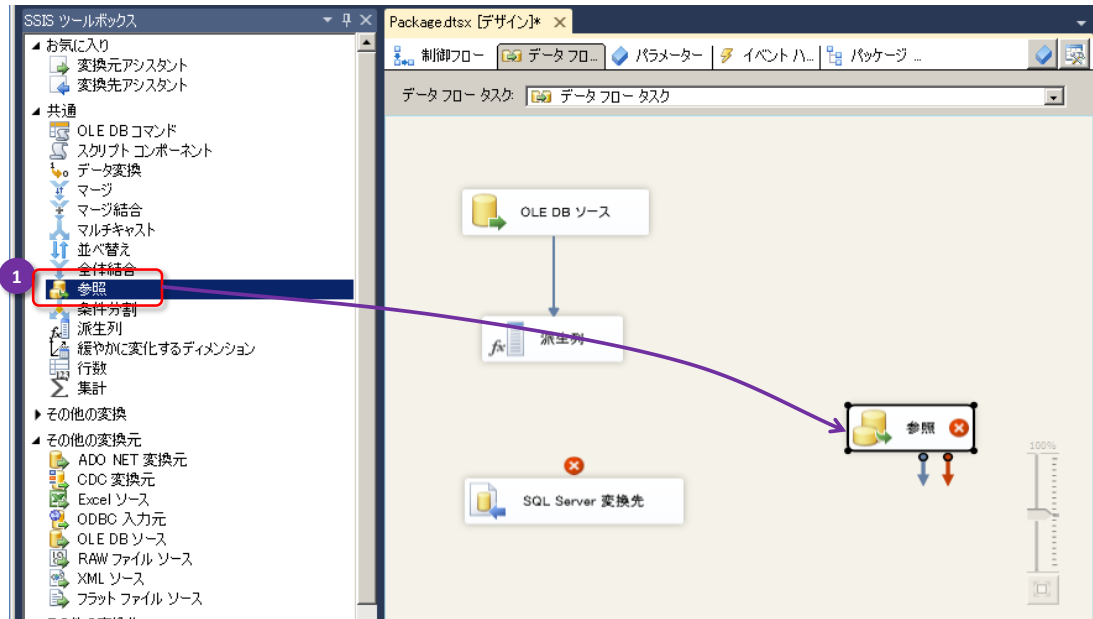


## ➡ 参照コンポーネントの追加

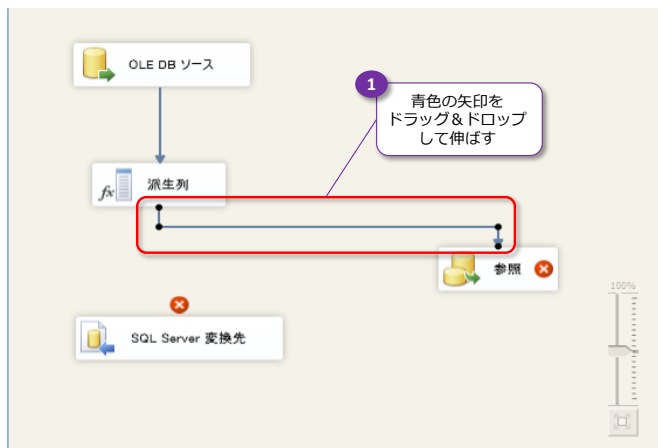
1. まずは、前の手順で設定した「派生列」から「SQL Server 変換先」へ向かって出ている "青色の矢印" を右クリックして、「削除」をクリックします。これにより矢印が削除されます。



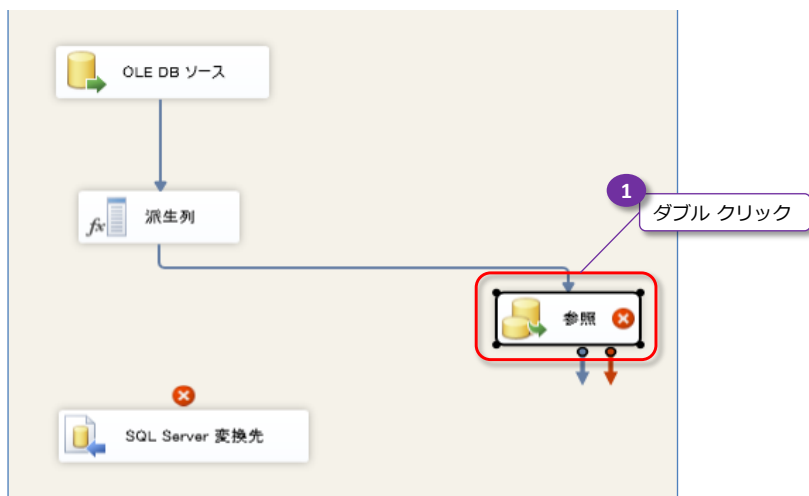
2. 次に、SSIS ツールボックスの「共通」カテゴリの中から、「参照」コンポーネントを SSIS デザイナー上へドラッグ&ドロップします。



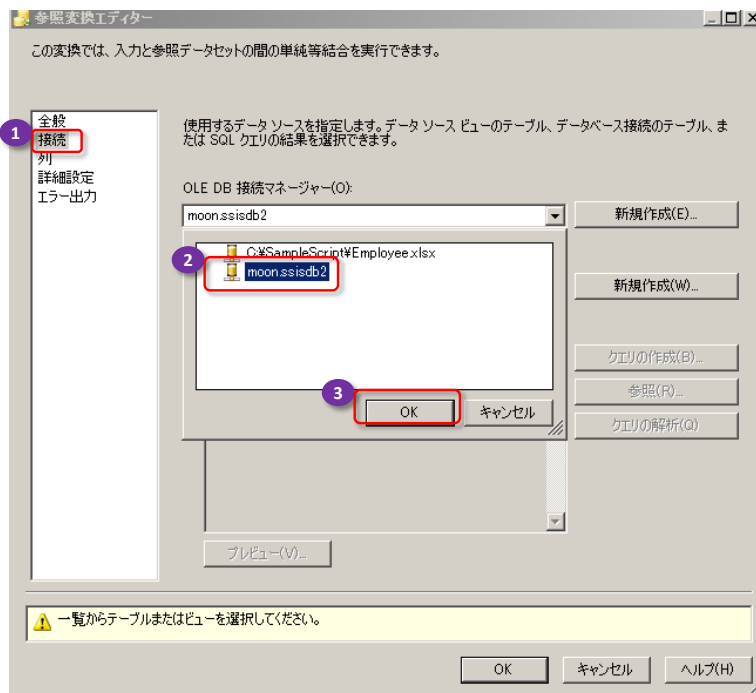
- 次に、SSIS デザイナー上にある「派生列」コンポーネントをクリックして "青色の矢印" を表示し、この矢印を「参照」コンポーネントまで、ドラッグ&ドロップして伸ばします。



- 次に、取得するデータを指定するために、「参照」コンポーネントをダブル クリックします。

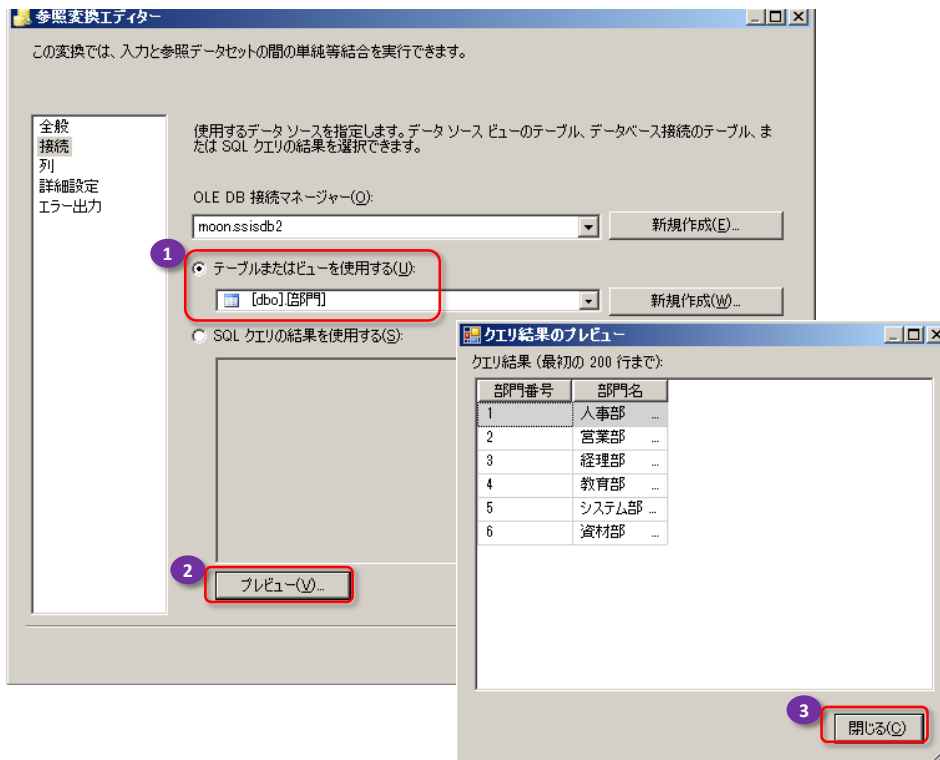


- 「参照変換エディター」ダイアログが表示されたら、「接続」ページをクリックして開きます。



[接続] ページが表示されたら、[OLE DB 接続マネージャー] で「サーバー名.ssisdb2」を選択して、[OK] ボタンをクリックします。

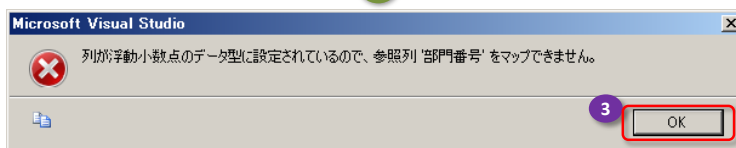
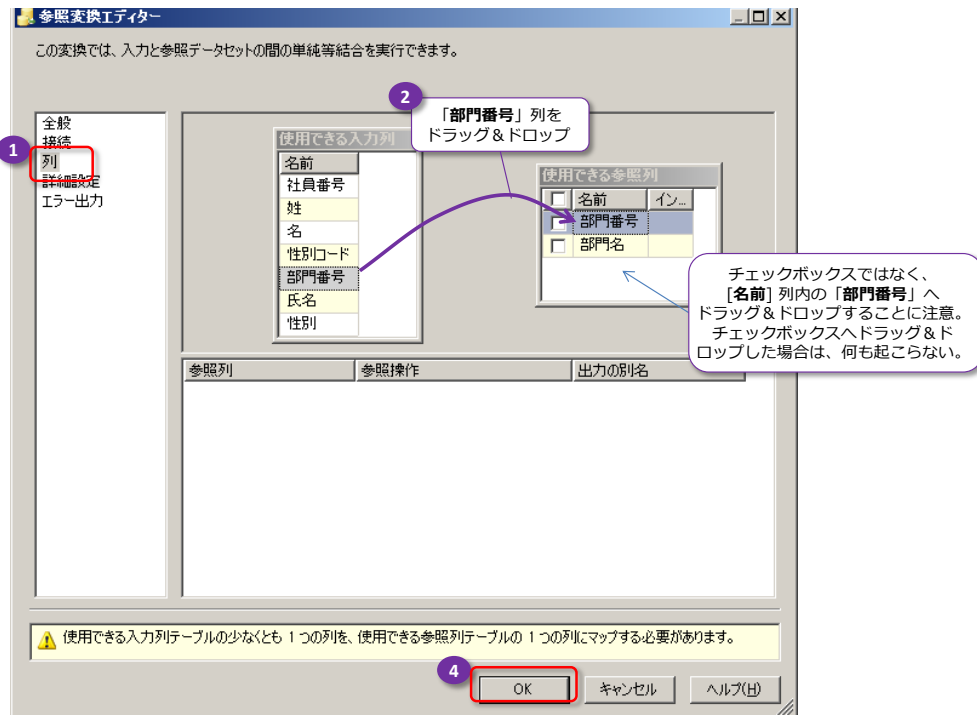
6. 続いて、[テーブルまたはビューを使用する] で参照先のテーブルとなる「部門」テーブルを選択します。



部門テーブルのデータを確認するために [プレビュー] ボタンをクリックします。確認後、[閉じる] ボタンをクリックします。

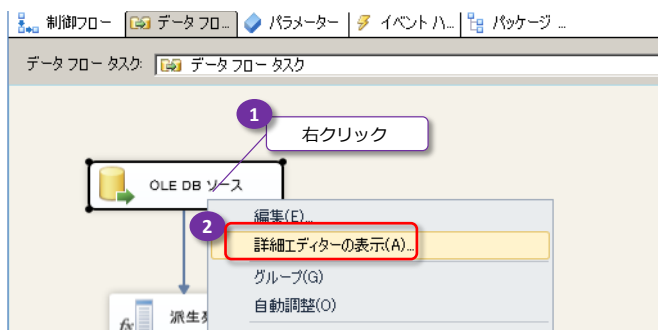
7. 続いて、[列] ページをクリックして、参照する列を設定します。今回は、[使用できる入力列]

の「部門番号」列の参照先が、[使用できる参照列]の「部門番号」列となるように、ドラッグ & ドロップします。



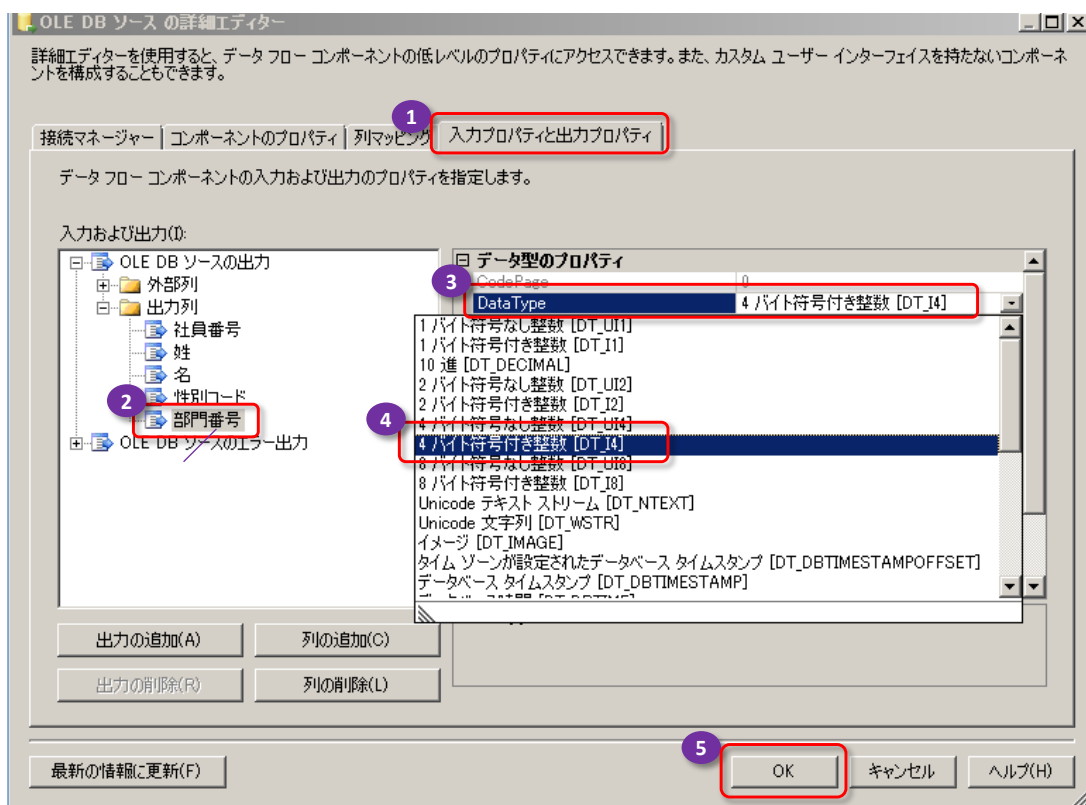
しかし、「データ型が浮動小数点 (float) 型のため、参照できない」という主旨のエラーメッセージが表示され、正しく設定することができません。参照コンポーネントは、データ型が一致しないと参照することができないためです (転送元の「部門番号」列は **float** 型で、参照先の「部門」テーブルの「部門番号」列は **int** 型で、データ型が一致していません)。そこで、ここでは、ひとまず [OK] ボタンをクリックして、エラーメッセージを閉じます。さらに、[接続] ページで [OK] ボタンをクリックして、[参照変換エディター] ダイアログを閉じます。

8. 転送元の「部門番号」列のデータ型を **float** 型から **int** 型へ変更するには、次のように [OLE DB ソース] を右クリックして、[詳細エディターの表示] をクリックします。



9. これにより、[OLE DB ソースの詳細エディター] ダイアログが表示されるので、次のように

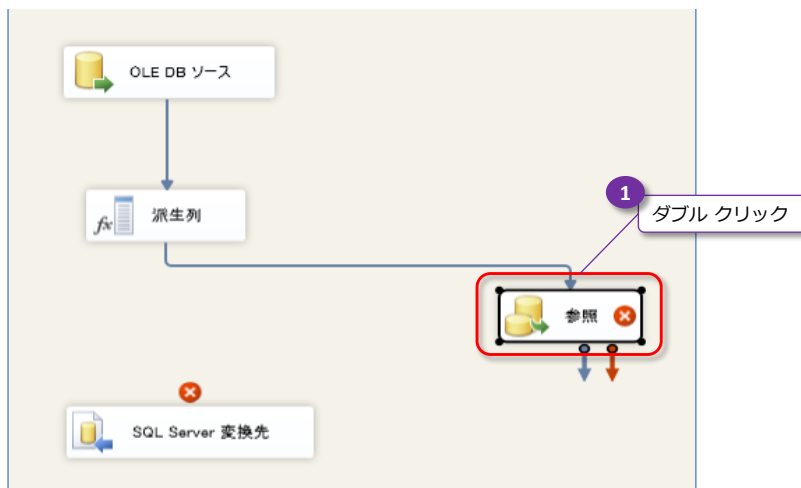
【入力プロパティと出力プロパティ】タブをクリックします。



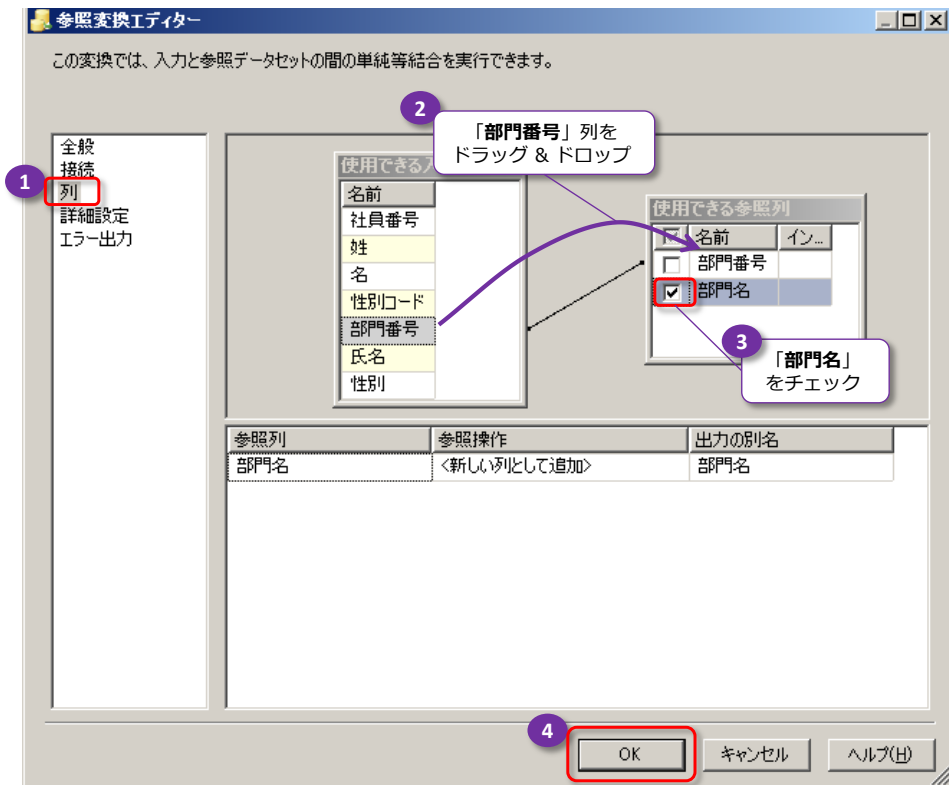
このダイアログでは、【OLE DB ソースの出力】の【出力列】を展開して、Excel ファイルから転送される列を一覧します。この一覧から「部門番号」列を選択して、表示されるプロパティの中から、【Data Type】（データ型）を探します。この値は、「倍精度浮動小数点数」（SQL Server での **float** 型に相当するデータ型）へ設定されているので、「4 バイト符号付き整数」（SQL Server での **int** 型に相当するデータ型）へ変更します。

データ型の変更後、【OK】ボタンをクリックして、ダイアログを閉じます。

- 次に、先ほど失敗した参照列の設定を再度行うために、【参照】コンポーネントをダブル クリックします。



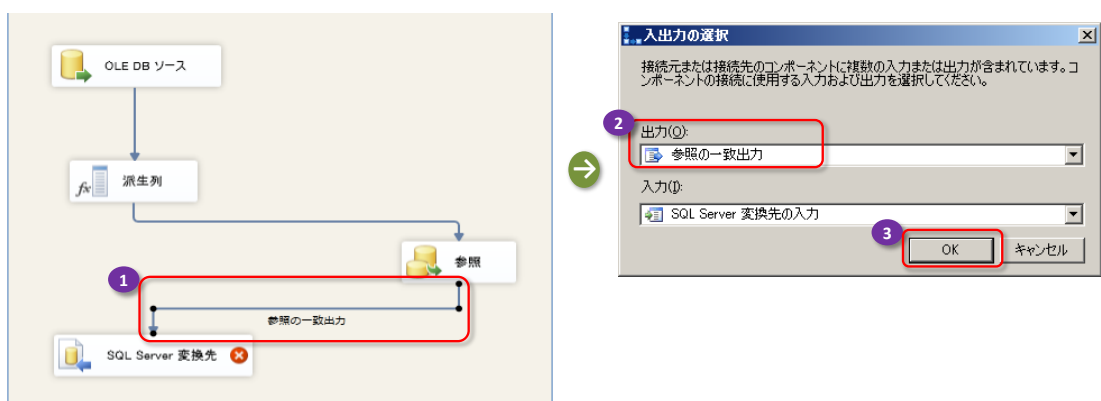
- 【参照変換エディター】ダイアログが表示されたら、【列】ページをクリックして開きます。



【使用できる入力列】の「部門番号」列を【使用できる参照列】の「部門番号」列へドラッグ＆ドロップすると、今度はエラーメッセージが表示されずに線が追加されます。続いて、取得したい列を指定するために、【使用できる参照列】の「部門名」列のチェック ボックスをチェックし、【OK】 ボタンをクリックします。

これで部門テーブルから部門名を参照（取得）できるようになります。

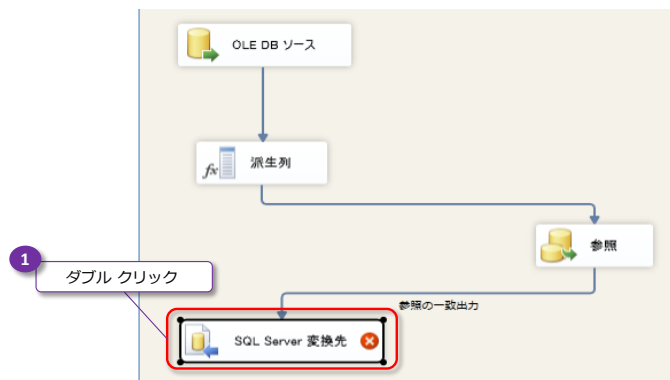
12. 次に、取得した部門名が SQL Server へ転送されるように、【参照】コンポーネントの青色の矢印を【SQL Server 変換先】までドラッグ＆ドロップして伸ばします。



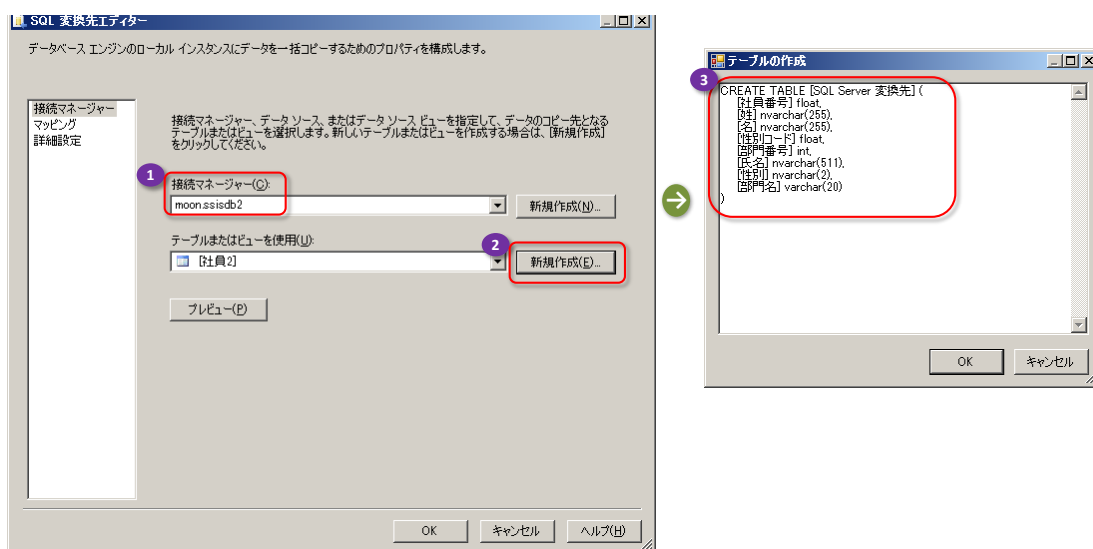
これにより、【入出力の選択】ダイアログが表示されるので、【出力】で「参照の一致出力」を選択して、【OK】 ボタンをクリックします。これで、【参照】コンポーネントで参照した値（参照元と参照先の部門番号が一致する部門名）を【SQL Server 変換先】へ渡せるようになります。

## ➡ 転送先のテーブルの変更

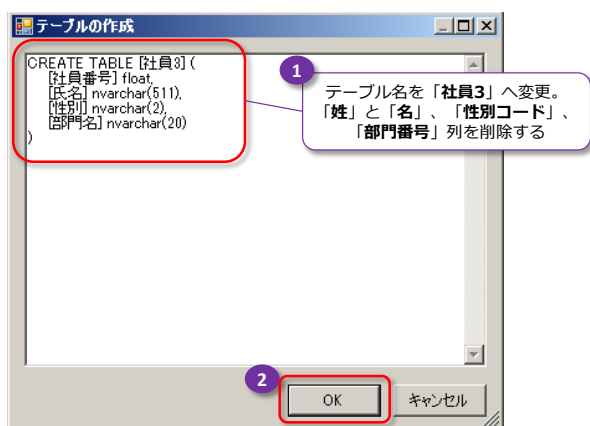
- 次に、転送先のテーブルを変更するために、[SQL Server 変換先] をダブル クリックして、[SQL 変換先エディター] ダイアログを開きます。



- [接続マネージャー] で「サーバー名.ssisdb2」が選択されていることを確認し、[テーブルまたはビューを使用] で [新規作成] ボタンをクリックします。

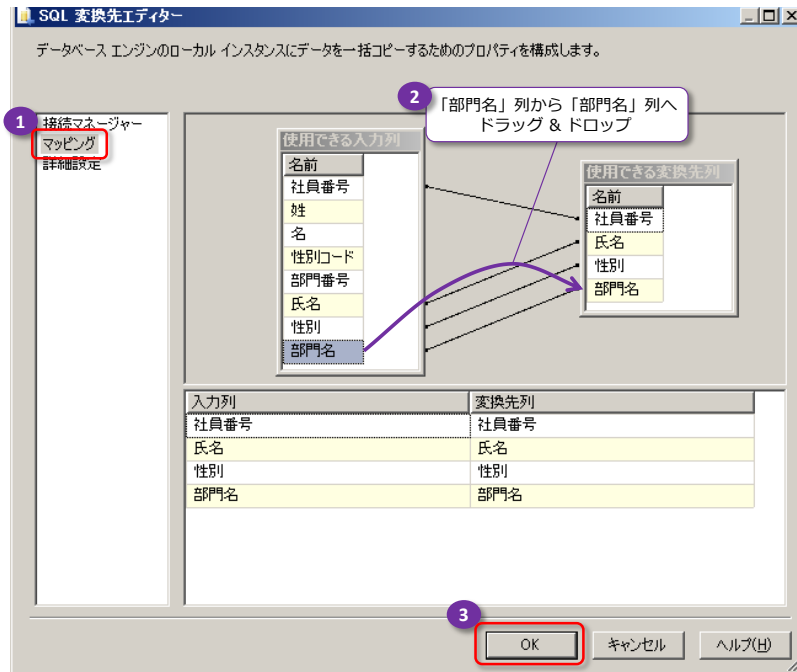


これにより、[テーブルの作成] ダイアログが表示されて、転送元のデータと派生列、参照コンポーネントで参照した列をもとに、**CREATE TABLE** ステートメントが自動生成されます。今回は、このステートメントを次のように変更します。



ステートメント変更後、[OK] ボタンをクリックすると、「社員 3」テーブルが作成されます。

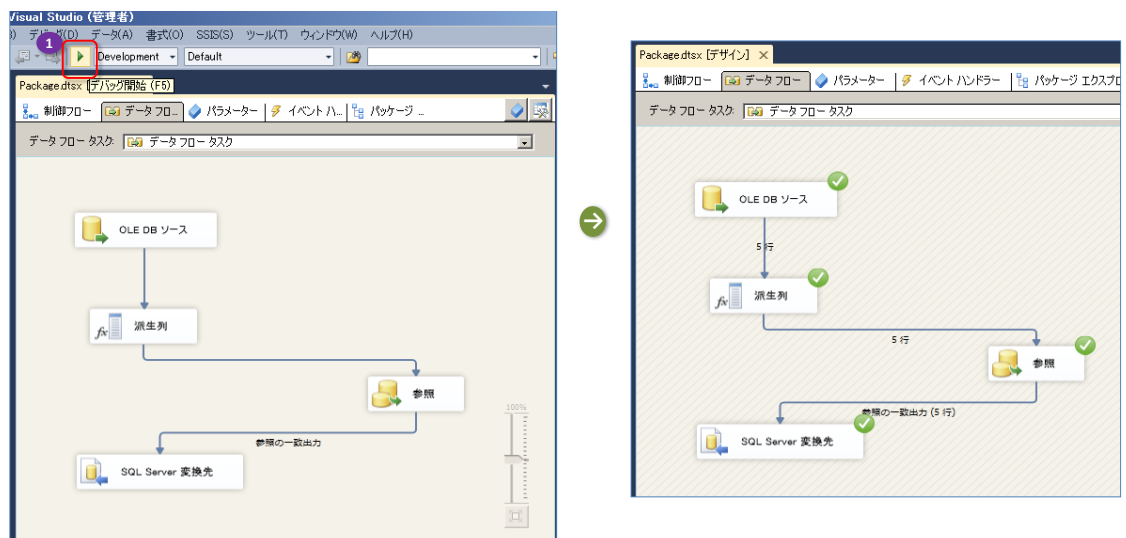
3. 「SQL 変換先エディター」ダイアログに戻ったら、次のように「マッピング」ページをクリックします。



【参照】コンポーネントで参照した「部門名」列がテーブルへ転送されるように、「使用できる入力列」の「部門名」列から「使用できる変換先列」の「部門名」列へドラッグ&ドロップします。線が追加されたことを確認して、[OK] ボタンをクリックします。

## ➡ データ転送の実行

4. ここまでの設定を確認するために、ツールバーの「デバッグ開始」ボタンをクリックして、パッケージを実行します。



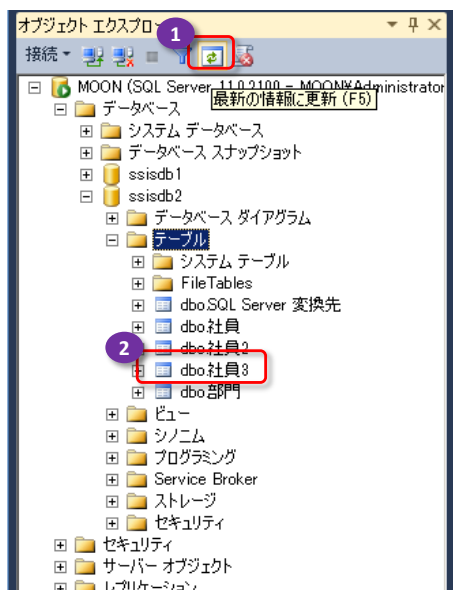
配置されているオブジェクトすべてに緑のチェックマークが付くと、データの転送が成功です。



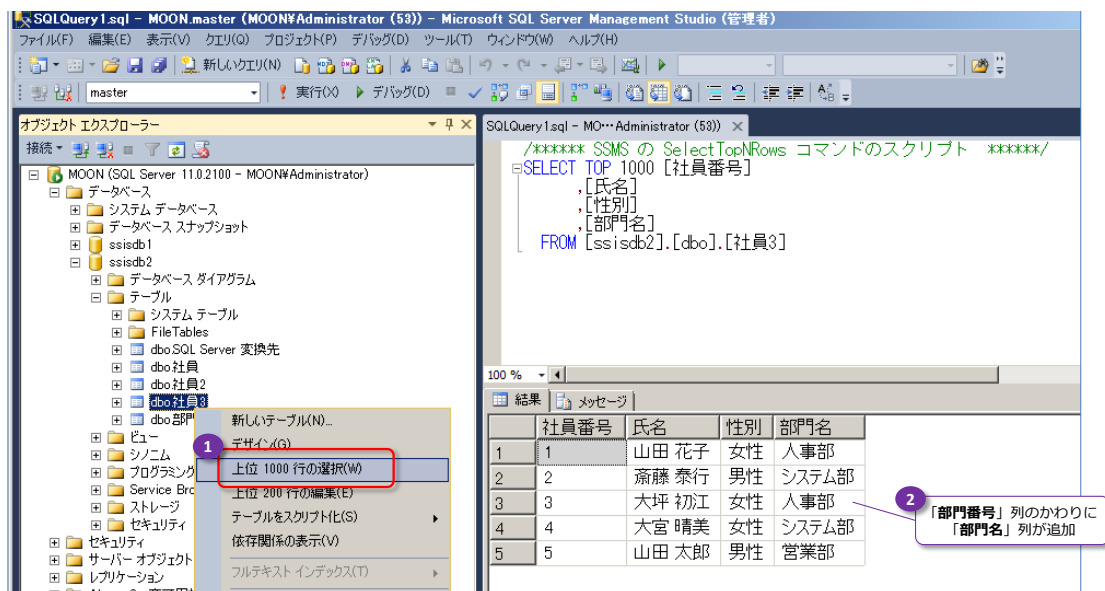
5. 転送が成功したら、ツールバーの「**デバッグの停止**」ボタンをクリックして、デバッグを終了します。

## ➡ 転送されたデータの確認

6. 転送されたデータを確認ために、Management Studio から、「ssisdb2」データベースの「テーブル」フォルダーに「社員3」という名前のテーブルが作成されていることを確認します（表示されない場合は、「最新の情報に更新」をクリックします）。



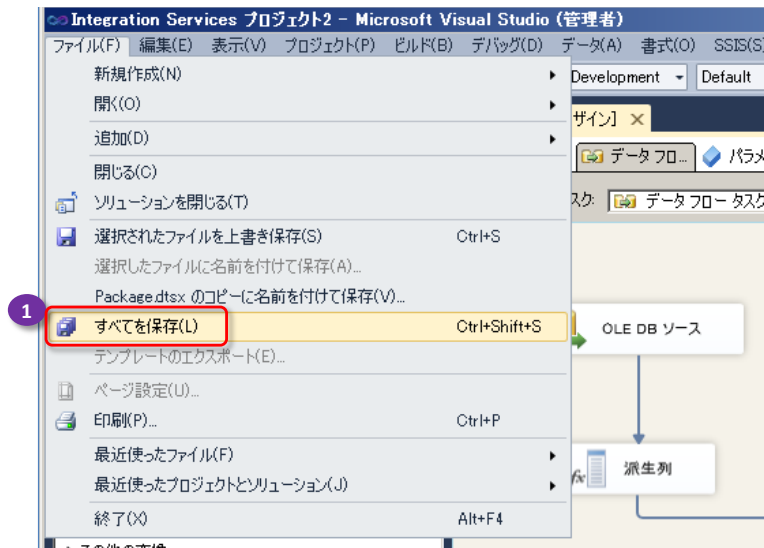
7. 続いて、「社員3」テーブルを右クリックして「上位 1000 行の選択」をクリックします。



「部門番号」列の代わりに「部門名」列が追加されていることを確認できます。このように参照コンポーネントを利用すると、他のテーブルを参照して、関連データを取得できるようになるので、大変便利です。

## ➡ プロジェクトの保存

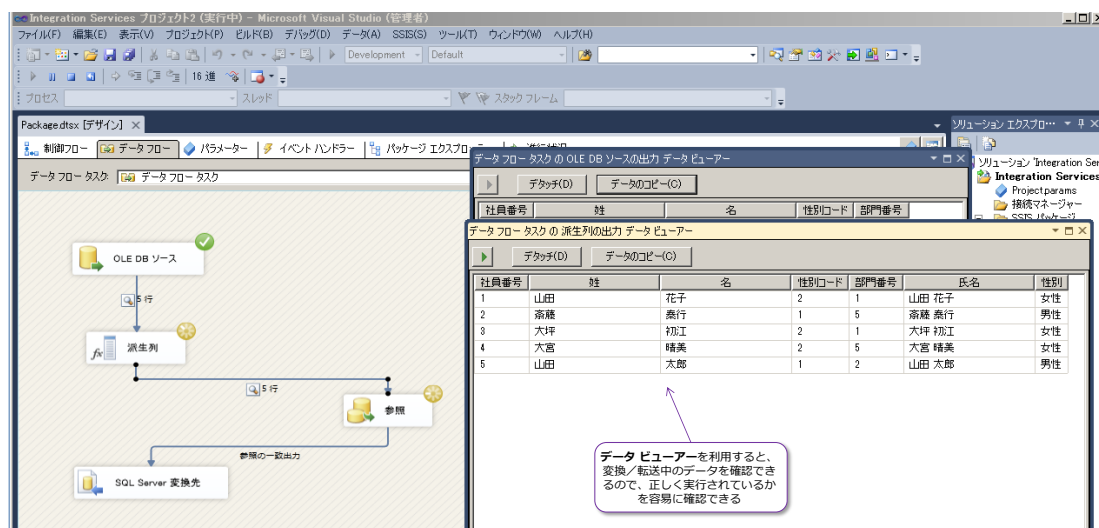
8. 最後に、ここまでの手順を保存するために、次のように【ファイル】メニューの【すべてを保存】をクリックして、プロジェクトを保存します。



## 3.7 データ ビューアーによる転送中のデータの表示

### ➡ データ ビューアーによる転送中のデータの表示

データの転送は、ここまで作成してきたように、矢印の順にそれぞれのコンポーネントを通して行われます。SSIS デザイナーでは、「データ ビューアー」機能を利用すると、コンポーネント間を転送されるデータを確認できるようになります。

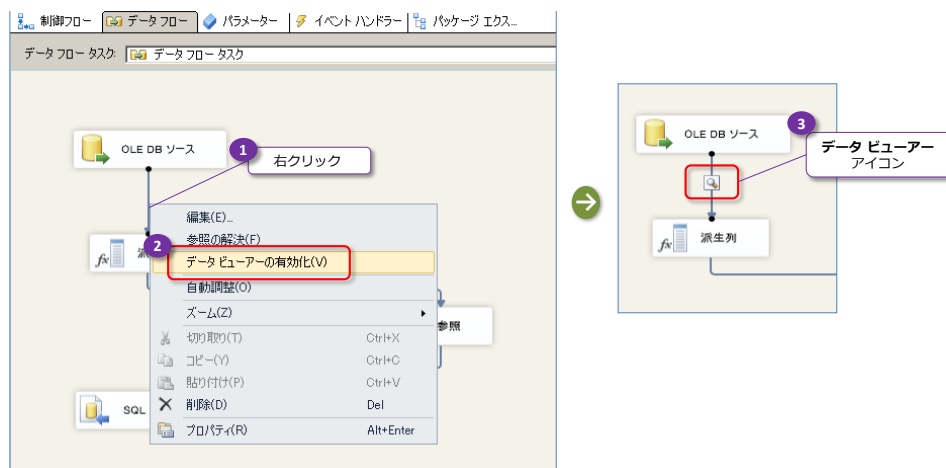


データ ビューアーは、派生列や参照コンポーネントによるデータ変換処理が正しく行われているかどうかの確認目的（デバッグ目的）として利用できるので、大変便利です。

### ➡ データ ビューアーの有効化

それでは、これを試してみましょう。ここでは、これまで作成してきた「**Integration Services プロジェクト 2**」プロジェクトのパッケージを引き続き利用します。

1. 最初に、[OLE DB ソース] から [派生列] へ転送されるデータを確認するためのデータ ビューアーを設定してみましょう。次のように SSIS デザイナー上の [OLE DB ソース] と [派生列] の間の矢印を右クリックして、[データ ビューアーの有効化] をクリックします。

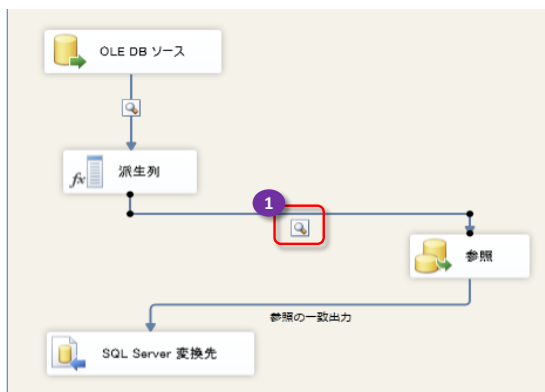


これにより、**データ ビューアー** アイコンが追加されることを確認できます。

2. 同様にして、[派生列] と [参照] コンポーネントとの間にもデータ ビューアーを設定しましょう。次のように SSIS デザイナー上の [派生列] と [参照] の間の矢印を右クリックして [**データ ビューアーの有効化**] をクリックします。



これにより、**データ ビューアー** アイコンが追加されることを確認します。

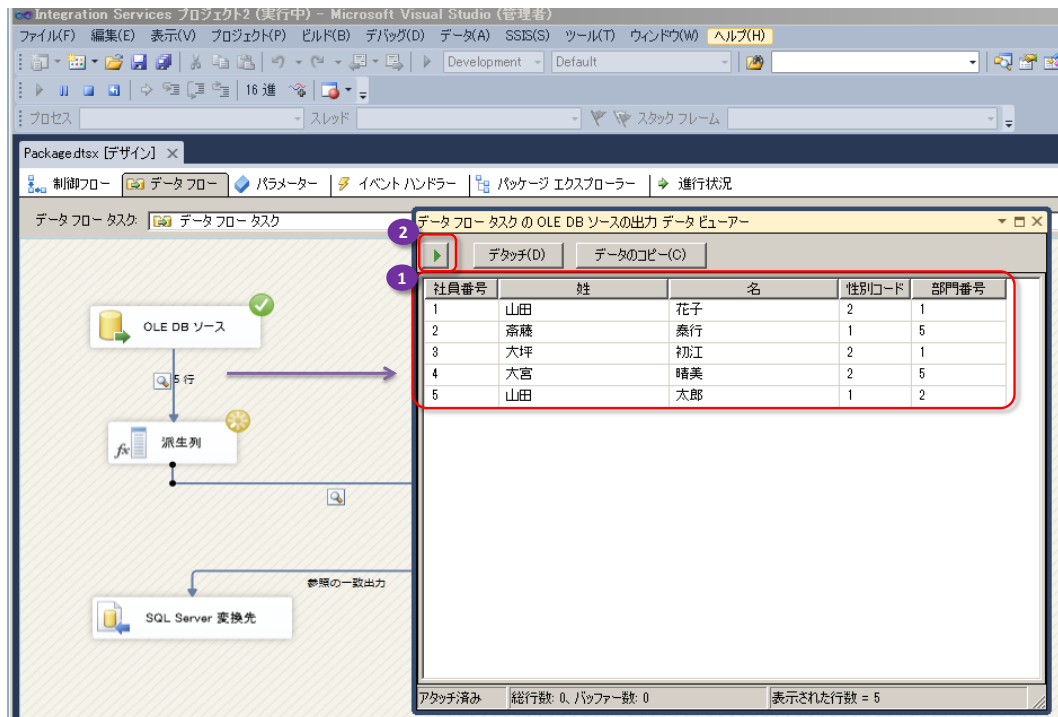


## ➡ 実行してデータ ビューアーの効果を確認


3. 次に、ツールバーの [**デバッグ開始**] ボタンをクリックして、パッケージを実行します。



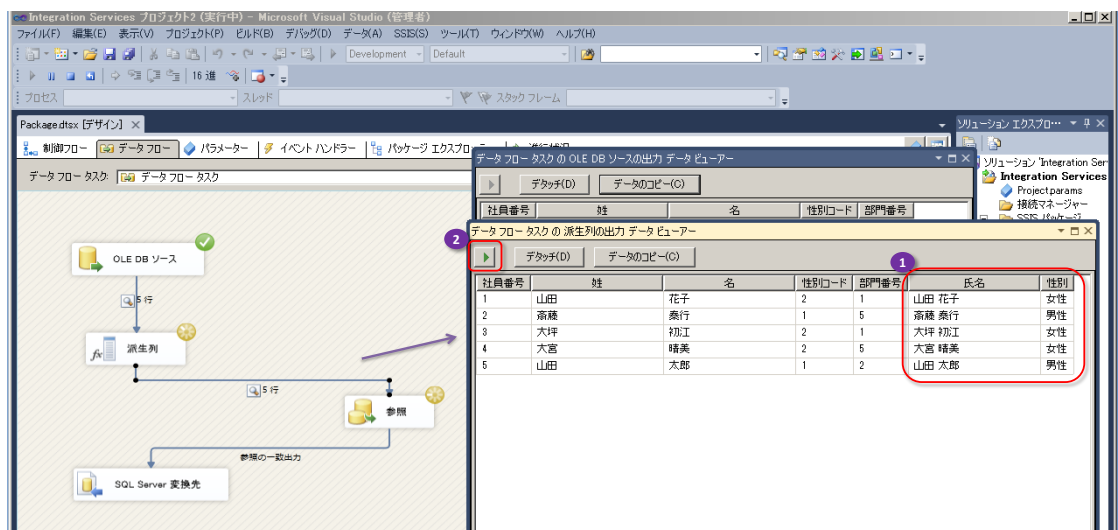
4. すると、[派生列] コンポーネントが実行中になってデータ転送がここで一時的に停止し、[データ ビューアー] ウィンドウが表示されます。




データ ビューアーでは、[OLE DB ソース] から[派生列] に転送されてきたデータを確認することができます。

このデータは、元の Excel データと同じであることを確認し、「」ボタンをクリックして、転送を再開します。

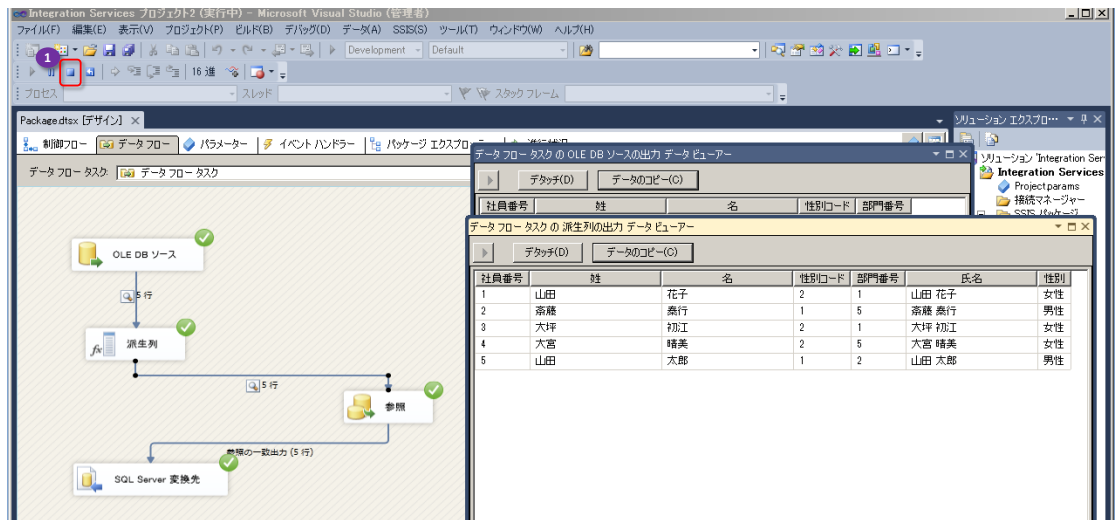
5. 今度は、[参照] コンポーネントが実行中になってデータ転送がここで再び一時的に停止し、次のように [データ ビューアー] ウィンドウが表示されます。



データ ビューアーでは、[派生列] から[参照] コンポーネントへ転送されてきたデータを確認することができます。

派生列（新しい列）として作成した「氏名」列と「性別」列が追加されていることを確認して、  
 ボタンをクリックし、転送を再開します。

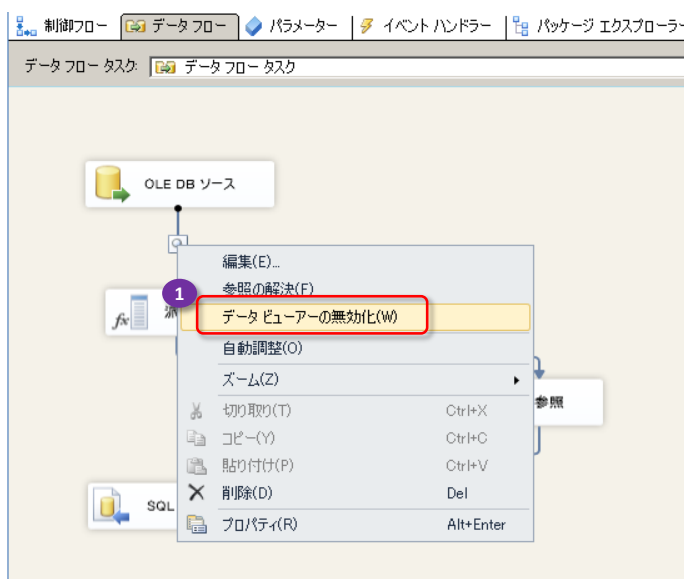
- すべてのコンポーネントに緑のチェックマークが付くと、データの転送が完了です。転送完了後、ツールバーの[デバッグの停止] ボタンをクリックして、デバッグを終了します。



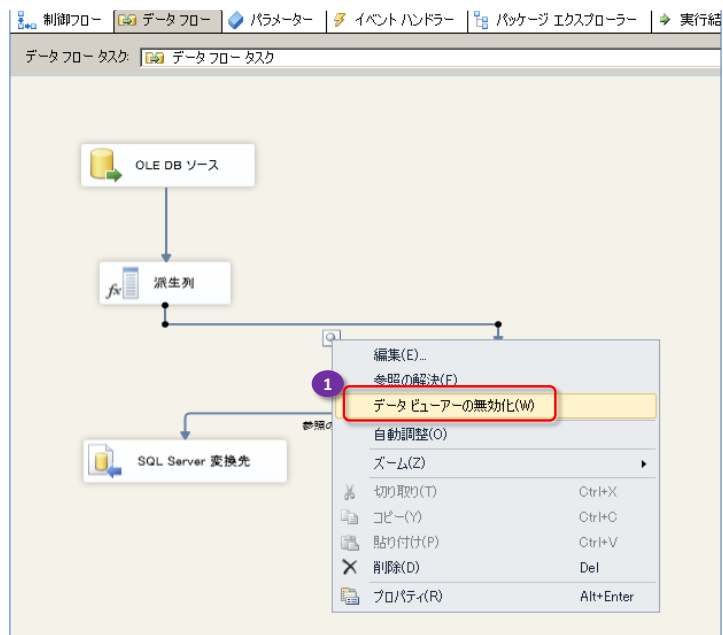
このように、データ ビューアーを利用すると、コンポーネント間を転送されるデータを確認することができ、派生列や参照コンポーネントなどのデータ変換処理が正しく行われているかどうかを確認する目的として利用できるので、大変便利です。

## ➡ データ ビューアーの無効化

- 設定したデータ ビューアーを無効化したい場合は、次のように青の矢印を右クリックして[データ ビューアーの無効化] をクリックします。



- もう 1 つのデータ ビューアーに対しても、同様の操作をして、無効化しておきます。



## STEP 4. パッケージの実行

---

この STEP では、保存したパッケージを実行するためのユーティリティとなる「**dtexecUI.exe**」と「**dtexec.exe**」や、パッケージを定期的に実行する方法を説明します。

この STEP では、次のことを学習します。

- ✓ パッケージの実行（dtexecUI.exe）
- ✓ コマンド プロンプトからの実行（dtexec.exe）
- ✓ パッケージの定期実行（SQL Server Agent ジョブへの登録）



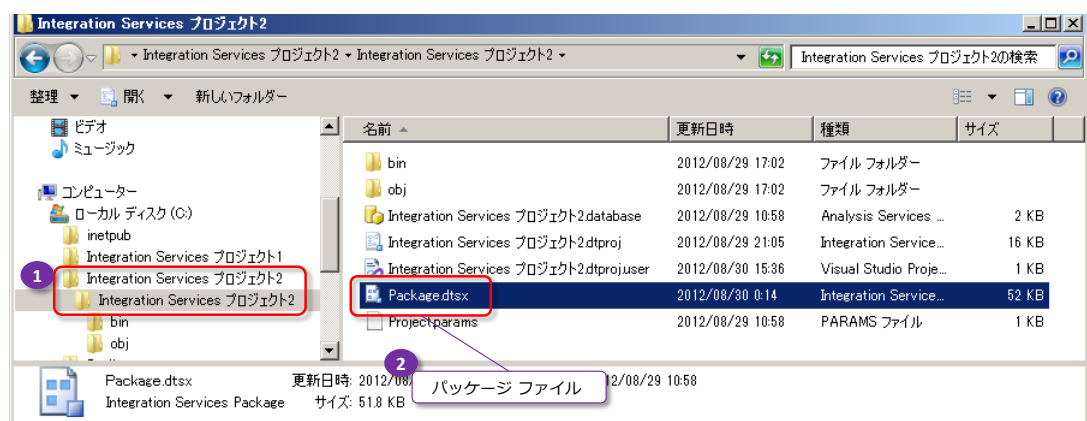
## 4.1 パッケージの実行 (dtexecUI)

### ➡ パッケージの実行

SSIS パッケージを実行するには、**dtexecUI.exe**（パッケージ実行ユーティリティ）を使用します。ここでは、STEP 3 で作成したパッケージ（**Integration Services プロジェクト 2** プロジェクト内にあるパッケージ）を実行してみましょう。

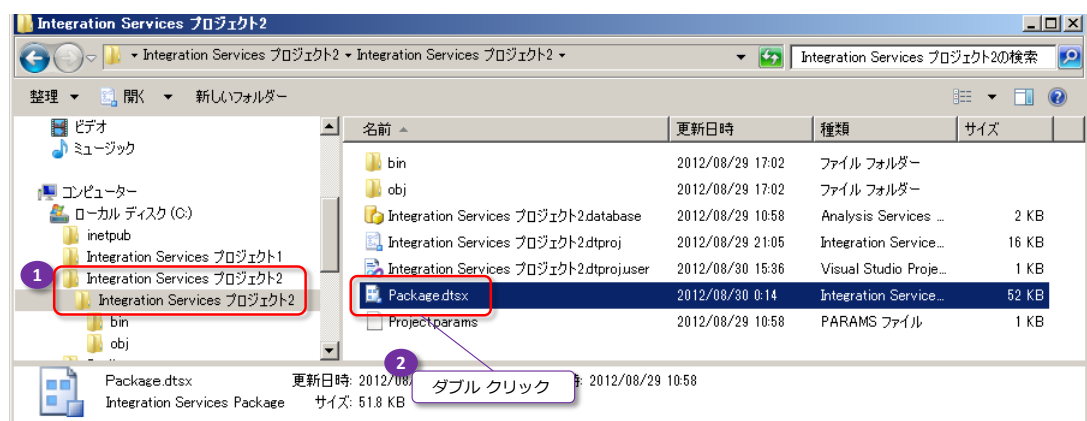
### ➡ パッケージの確認

1. まずは、STEP3 で作成したパッケージ ファイルを確認します。**Windows エクスプローラー**を起動して、「C:¥Integration Services プロジェクト 2¥Integration Services プロジェクト 2」フォルダーを展開します。この中にある「**Package.dtsx**」というファイルが、「**Integration Services プロジェクト 2**」プロジェクトで作成したパッケージ ファイルです。

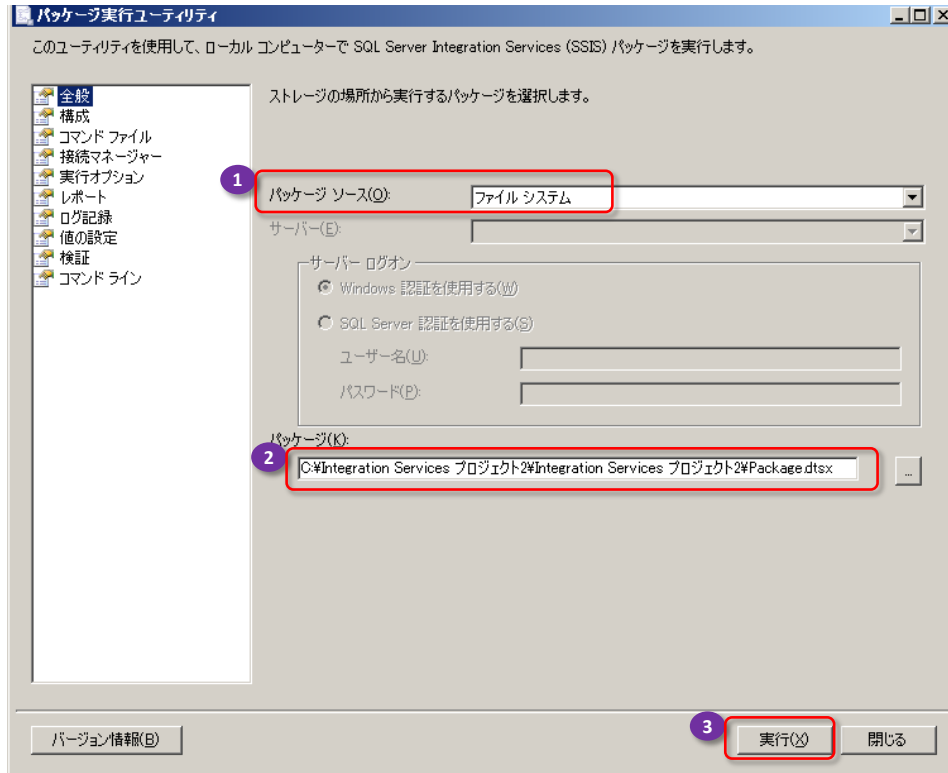


### ➡ dtexecUI の起動

2. 次に、パッケージ ファイル（**Package.dtsx**）をダブル クリックします。



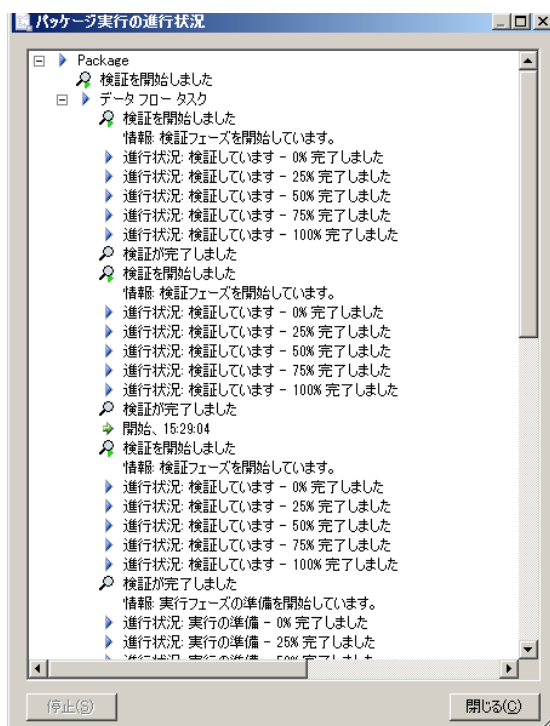
3. これにより、**パッケージ実行ユーティリティ (dtexecUI.exe)** が起動されて、次のように表示されます。

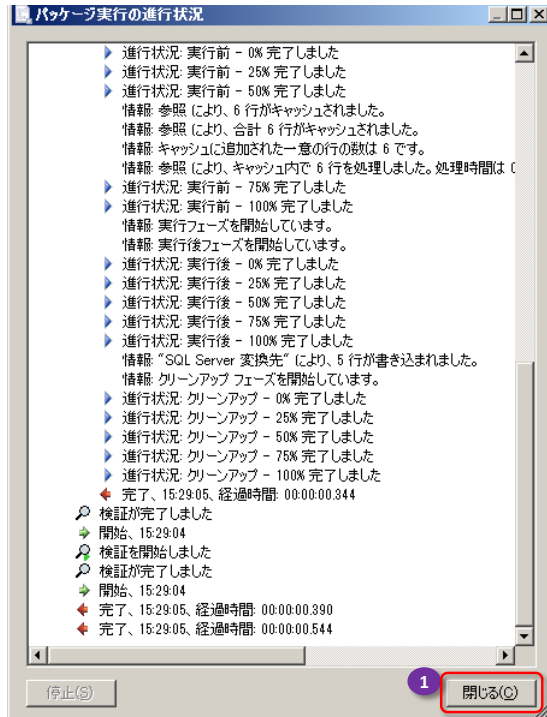


[パッケージ ソース] が「ファイル システム」になっていて、[パッケージ] がパッケージ ファイルのパス（今回は、**C:\Integration Services プロジェクト 2\Integration Services プロジェクト 2\Package.dtsx**）になっていることを確認します。

次に、[実行] ボタンをクリックして、パッケージの実行を開始します。

4. パッケージの実行が開始されると、次のように [パッケージ実行の進行状況] ダイアログが表示されて、進捗状況が表示されます。

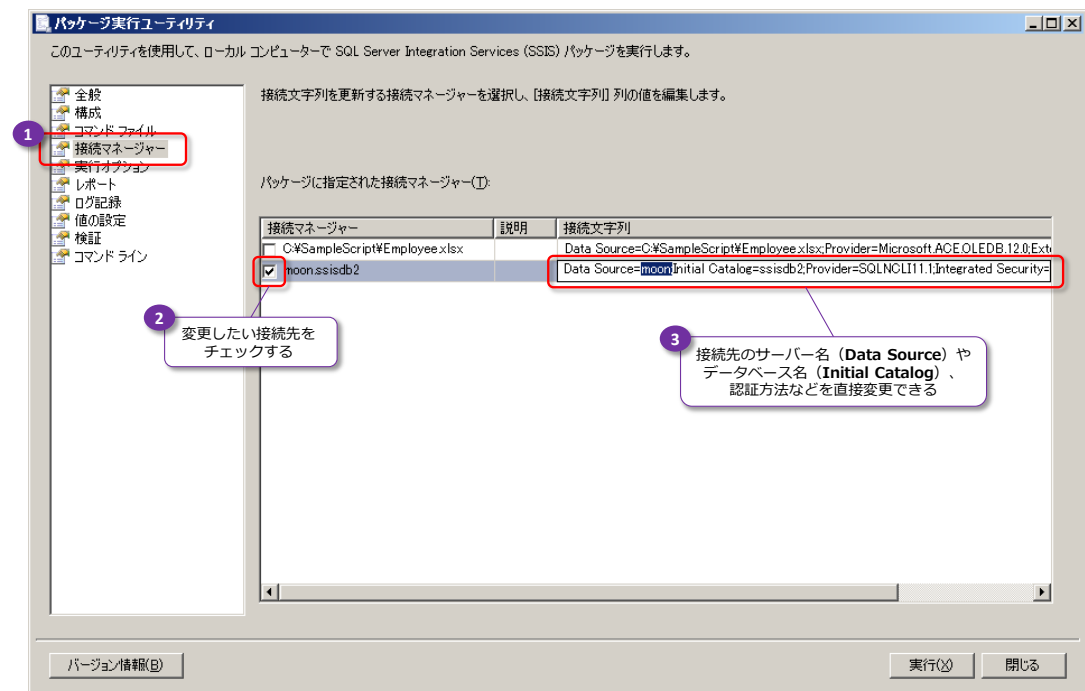




完了後、[閉じる] ボタンをクリックします。

#### Note : 実行時に接続先を変更することも可能

[パッケージ実行ユーティリティ] では、[接続マネージャー] ページを利用すると、パッケージの実行時に接続先 (SQL Server の名前やデータベース名、Excel ファイルの格納されているフォルダー パスなど) を変更することも可能です。



この機能は、開発機と本番機で SQL Server の名前が異なる場合や、データベース名が異なる場合に大変便利です。また、dtexecUI では、接続先だけでなく、パッケージ内に配置したコンポーネントのその他のプロパティについても可変にして、実行時に値を変更することができます。これは、「パラメーター化」機能を利用することで行えますが、これについては、本自習書シリーズの「Integration Services 応用」編で詳しく説明しています。

## 4.2 コマンド プロンプトからのパッケージ実行 (dtexec)

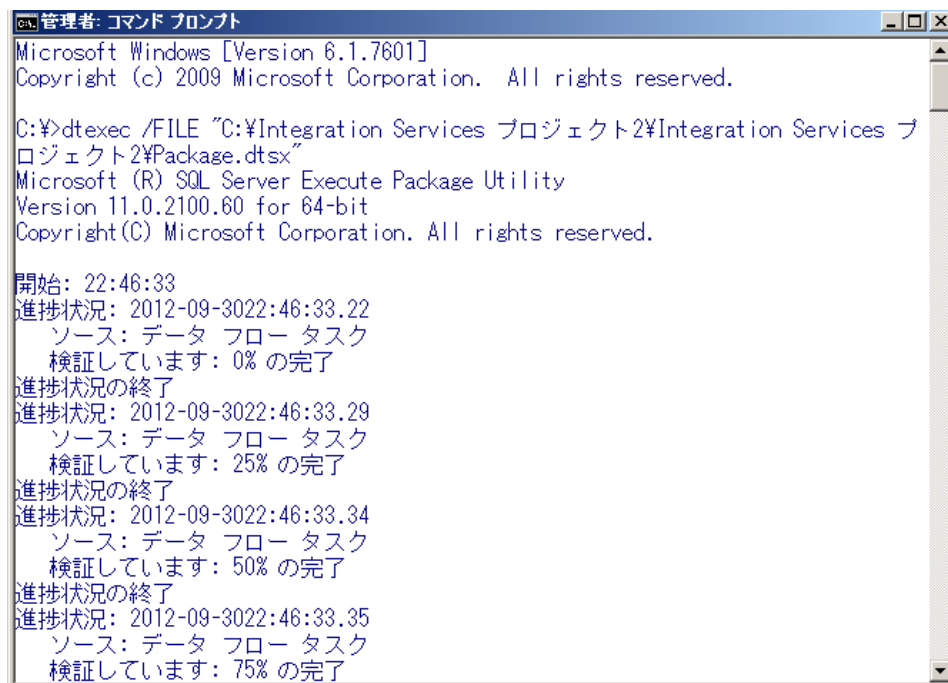
### ➡ コマンド プロンプトからのパッケージ実行 (dtexec)

SSIS パッケージは、「dtexec.exe」を利用すると、コマンドから実行することもできます。

1. **dtexec** を利用して、STEP 3 で作成したパッケージを実行するには、コマンド プロンプトから次のようにコマンドを入力し、[Enter] キーを押して実行します。

```
dtexec /FILE "C:\¥Integration Services プロジェクト 2¥Integration Services プロ  
ジェクト 2¥Package.dtsx"
```

これにより、次のようにパッケージの進捗状況が表示されます。

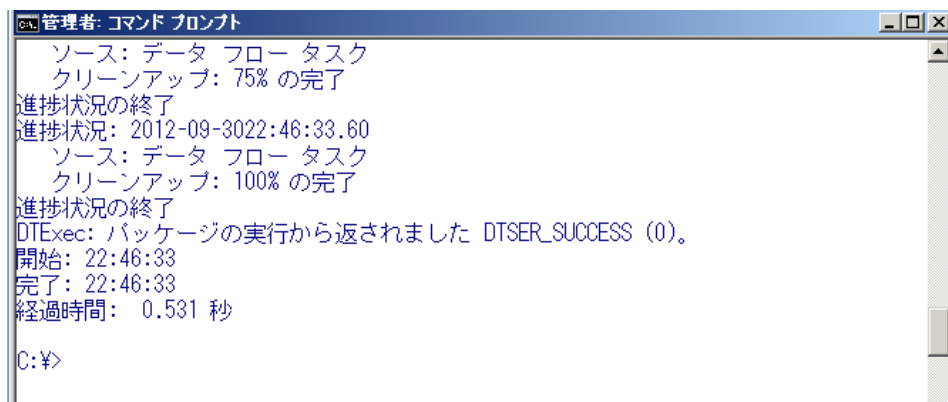


```
管理者: コマンド プロンプト
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\¥>dtexec /FILE "C:\¥Integration Services プロジェクト 2¥Integration Services プロ  
ジェクト 2¥Package.dtsx"
Microsoft (R) SQL Server Execute Package Utility
Version 11.0.2100.60 for 64-bit
Copyright (C) Microsoft Corporation. All rights reserved.

開始: 22:46:33
進捗状況: 2012-09-3022:46:33.22
  ソース: データ フロー タスク
  検証しています: 0% の完了
進捗状況の終了
進捗状況: 2012-09-3022:46:33.29
  ソース: データ フロー タスク
  検証しています: 25% の完了
進捗状況の終了
進捗状況: 2012-09-3022:46:33.34
  ソース: データ フロー タスク
  検証しています: 50% の完了
進捗状況の終了
進捗状況: 2012-09-3022:46:33.35
  ソース: データ フロー タスク
  検証しています: 75% の完了
```

パッケージの実行が完了すると、次のようにメッセージが表示されます。

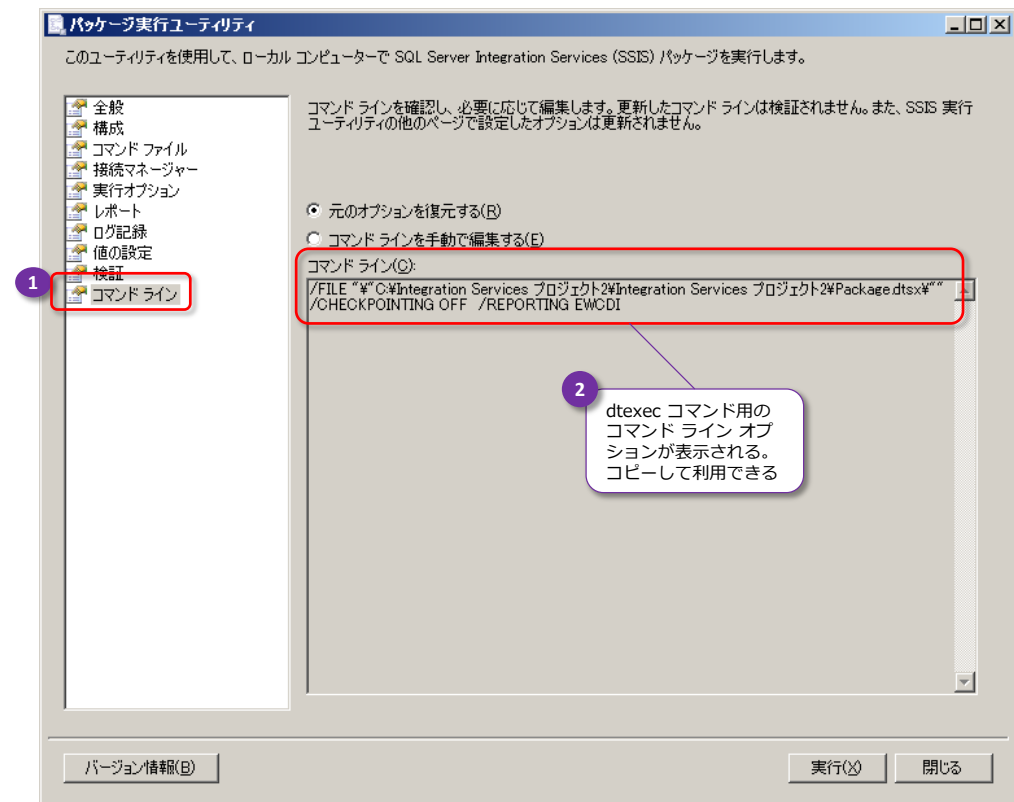


```
管理者: コマンド プロンプト
  ソース: データ フロー タスク
  クリーンアップ: 75% の完了
進捗状況の終了
進捗状況: 2012-09-3022:46:33.60
  ソース: データ フロー タスク
  クリーンアップ: 100% の完了
進捗状況の終了
DTExec: パッケージの実行から返されました DTExec_SUCCESS (0)。
開始: 22:46:33
完了: 22:46:33
経過時間: 0.531 秒

C:\¥>
```

## Note : dtexecUI を利用したコマンド ライン オプションの生成

dtexecUI.exe の [コマンド ライン] ページでは、GUI ベースで設定した実行時のオプションに対して、コマンド プロンプト (dtexec) から実行する際のコマンド ライン オプションを生成してくれる機能があるので、これをコピーして dtexec を実行すると便利です。



## 4.3 パッケージの定期実行（SQL Server Agent ジョブ）

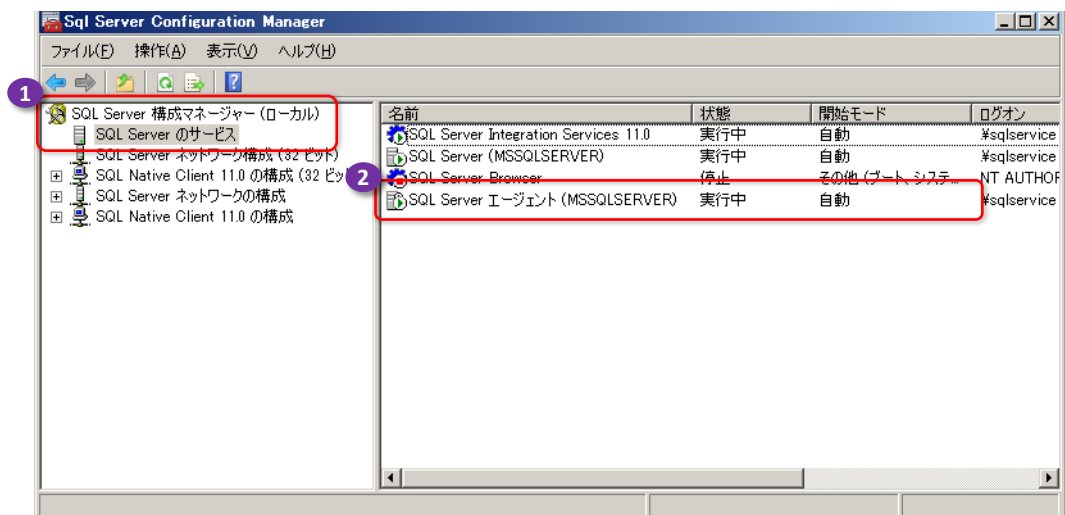
### ➡ パッケージの定期実行

SSIS パッケージを定期的に行きたい場合は、SQL Server Agent サービスの「ジョブ」機能を利用します。ここでは、STEP3 で作成したパッケージ（C:\¥Integration Services プロジェクト 2¥Integration Services プロジェクト 2¥Package.dtsx）を定期的に行する手順を例に説明します。

### ➡ SQL Server Agent サービスの起動

パッケージをジョブとして定期的に行させるには、事前に「SQL Server Agent」サービスを開始しておく必要があります。

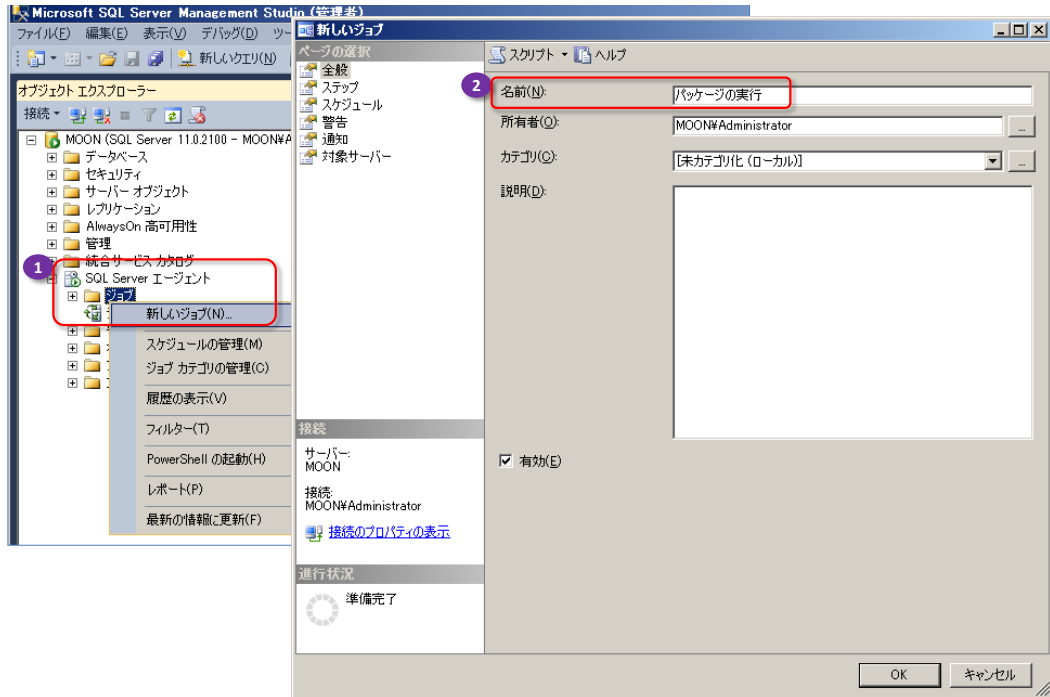
1. 「SQL Server Agent」サービスは、次のように「SQL Server 構成マネージャー」ツールを利用して開始しておきます。



「開始モード」を「自動」へ設定しておけば、次回 OS (Windows Server) が起動したときに、SQL Server Agent サービスも自動起動するようになります。

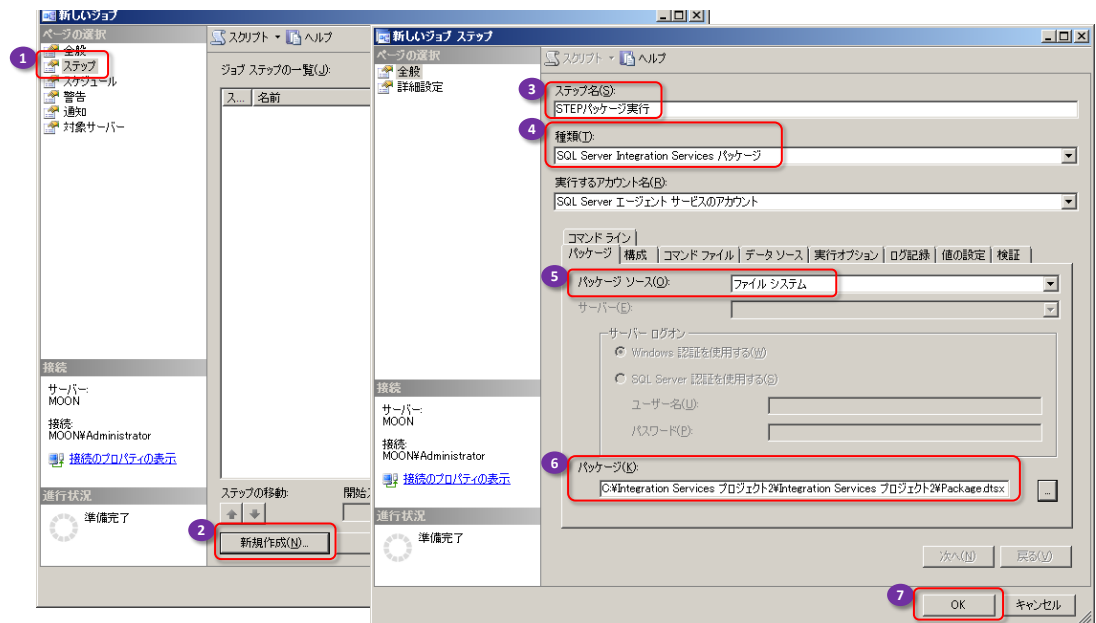
### ➡ SQL Server Agent のジョブとしての登録

2. パッケージを SQL Server Agent のジョブとして登録するには、次のようにオブジェクト エクスプローラーで「SQL Server エージェント」を展開して、「ジョブ」フォルダーを右クリックし、「新しいジョブ」をクリックします。



これにより、[新しいジョブ] ダイアログが表示されるので、[名前] へ任意の名前（**パッケージの実行**など）を入力します。

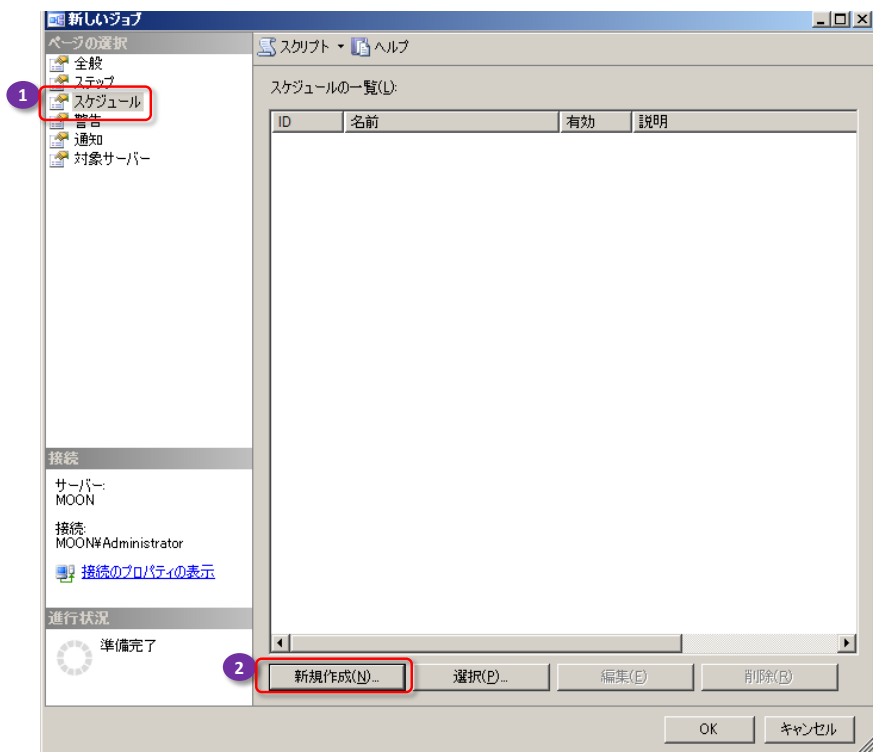
- 次に、[ステップ] ページをクリックして表示し、[新規作成] ボタンをクリックします。



これにより、[新しいジョブ ステップ] ダイアログが表示されるので、[ステップ名] へ任意の名前（**STEP パッケージ実行**など）と入力します。

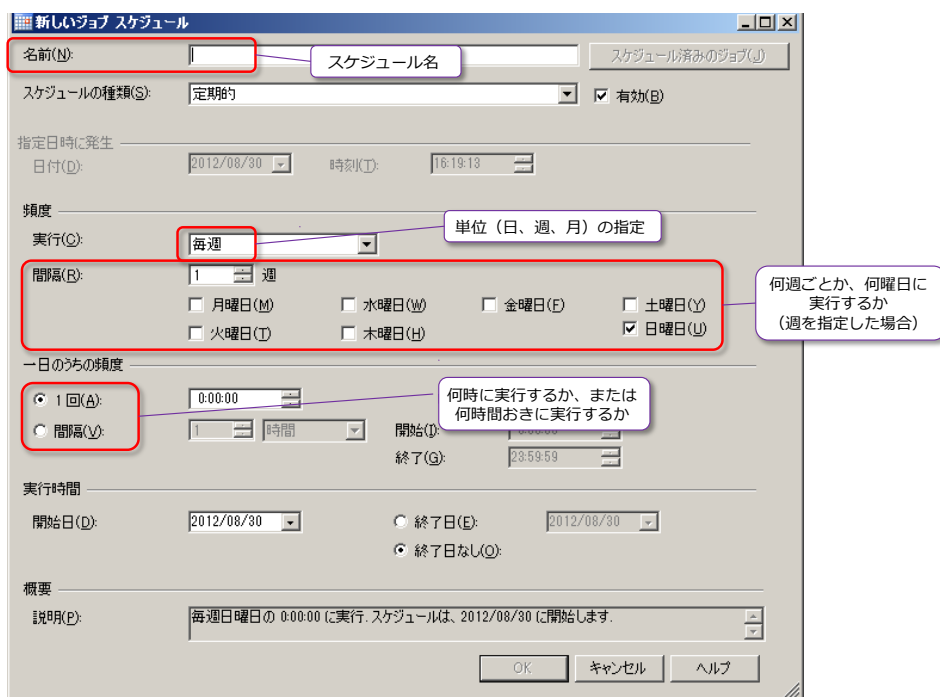
次に、パッケージをステップとして登録するために、[種類] で「**SQL Server Integration Services パッケージ**」を選択します。[パッケージ ソース] では、「**ファイル システム**」を選択し、[パッケージ]へ STEP 3 で作成したパッケージ ファイル「**C:\Integration Services プロジェクト2\Integration Services プロジェクト2\Package.dtsx**」を選択して、[OK] ボタンをクリックします。

4. 次に、ジョブのスケジュールを設定するために、次のように「スケジュール」ページをクリックして開きます。



新規のスケジュールを作成するために、「新規作成」ボタンをクリックします。

5. これにより、「新しいジョブ スケジュール」ダイアログが表示されて、スケジュールの設定が行えるようになります。



「名前」へ任意のスケジュール名を入力し、どのくらいの頻度で実行するかや、実行日時などを設定します。例えば、「毎日夜 2:00」に実行したい場合は、次のように設定します。



**新しいジョブ スケジュール**

名前(N): 毎日2時に実行 スケジュール済みのジョブ(J)

スケジュールの種類(S): 定期的 ☒ 有効(B)

指定日時に発生  
日付(D): 2012/08/30 時刻(T): 16:19:13

頻度

実行(O): 毎日

間隔(O): 1 日

一日のうちの頻度

☒ 1回(A): 2:00:00

☐ 間隔(V): 1 時間

開始(O): 0:00:00

終了(G): 23:59:59

実行時間

開始日(D): 2012/08/30 ☐ 終了日(E): 2012/08/30

☒ 終了日なし(O):

概要

説明(P): 毎日 2:00:00 に実行. スケジュールは、2012/08/30 に開始します.

1 **毎日 夜の 2:00**  
に実行したい場合は  
「実行」を「毎日」  
「間隔」を「1」日  
「1回」を「2:00:00」  
に設定する

2 **OK** キャンセル ヘルプ

6. 「新しいジョブ」ダイアログへ戻ったら、「OK」ボタンをクリックします。

**新しいジョブ**

ページの選択

- 全般
- ステップ
- スケジュール
- 警告
- 通知
- 対象サーバー

接続

サーバー:  
MOON

接続:  
MOON\Administrator

[接続のプロパティの表示](#)

進行状況

準備完了

スケジュールの一覧(L):

ID	名前	有効	説明
新規作成	毎日2時に実行	(はい)	毎日 2:00:00 に実行. スケジュールは、

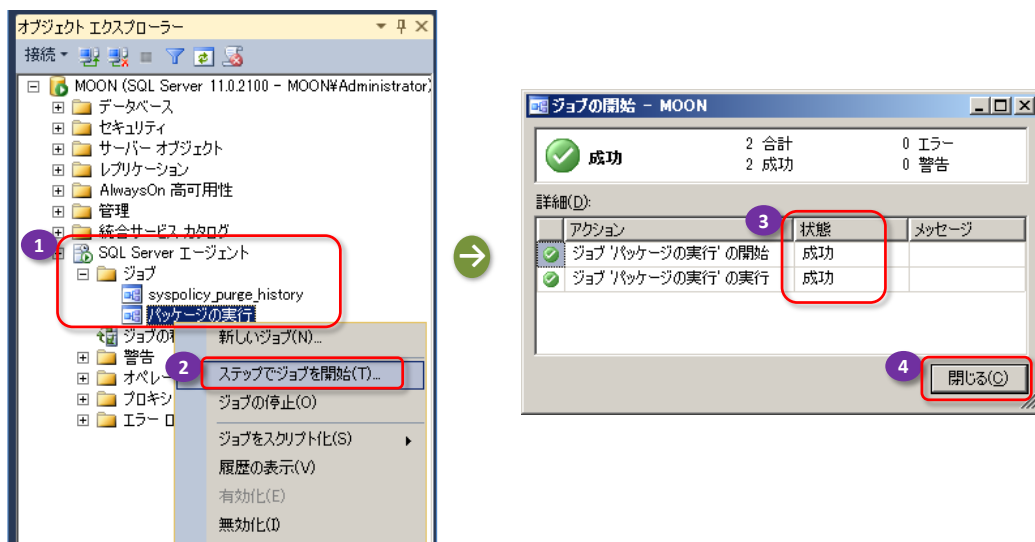
1 **OK** キャンセル

以上で、ジョブの作成が完了です。これで、**SQL Server Agent** サービスが起動している場合、設定したスケジュールの時刻にパッケージが自動実行されるようになります。

## ➡ ジョブの確認（手動実行）

次に、登録したパッケージのジョブが正常に実行されるかどうかを確認するために、ジョブを手動で実行してみましょう。

1. 登録したジョブを手動で実行するには、次のように「**ジョブ**」フォルダー内の登録したジョブ（**パッケージの実行**）を右クリックして、「**ステップでジョブを開始**」をクリックします。



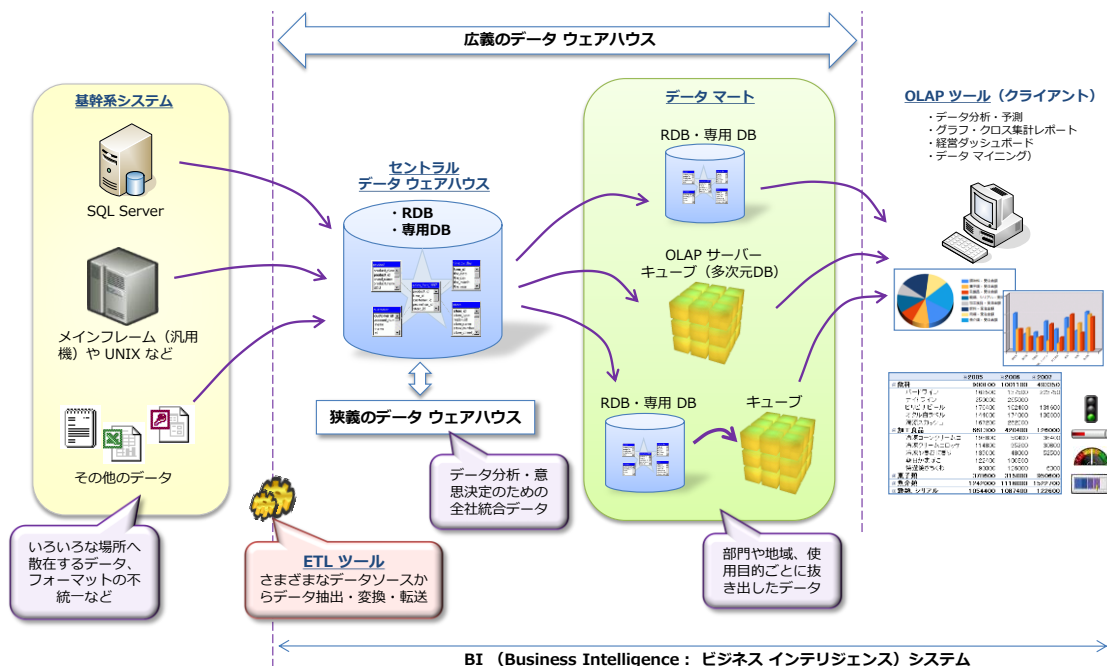
これにより、ジョブが開始されて、「**ジョブの開始**」ダイアログが表示されます。すべての「**状態**」が「**成功**」になれば、ジョブの実行が成功しています。結果を確認後、「**閉じる**」ボタンをクリックして、ダイアログを閉じます。

## 4.4 Integration Services とデータ ウェアハウスの関係 ～ETL～

### ➡ Integration Services とデータ ウェアハウスの関係

ここでは、Integration Services とデータ ウェアハウスとの関係について説明します。Integration Services は、データ ウェアハウスを構築する上で欠かせないツールとなります。

一般的なデータ ウェアハウスの構成は、次のようになります。



**データ ウェアハウス (DWH : Data Warehouse)** は、基幹業務系のシステムのデータを整理、統合、蓄積したもので、Warehouse は「倉庫」という意味です。蓄積したデータに対してさまざまな分析を行うことで、売上や利益を上げるための方針を立案したり、競合他社に対する施策を練ったりするために利用します。

### ➡ Integration Services は ETL ツール

データ ウェアハウスを構築する上で欠かせないツールとなるのが、Integration Services のように、さまざまなデータ ソースからデータを抽出して、フォーマットの統一（データ変換・加工）が行え、分析へ適した形へ変換／転送できるツールです。このようなツールは、**抽出、転送、ロード**が行えることから「**ETL**」(**Extraction, Transformation and Loading**) ツールと呼ばれています。このため、Integration Services は ETL ツールとも呼ばれています。

一般的には、データ ウェアハウスの構築作業のうち、ETL ツールによる作業が 60～80%も占めるとも言われており、非常に重要な役割を担っています。

**Note : データ ウェアハウスや BI の定義は、人によってさまざま**

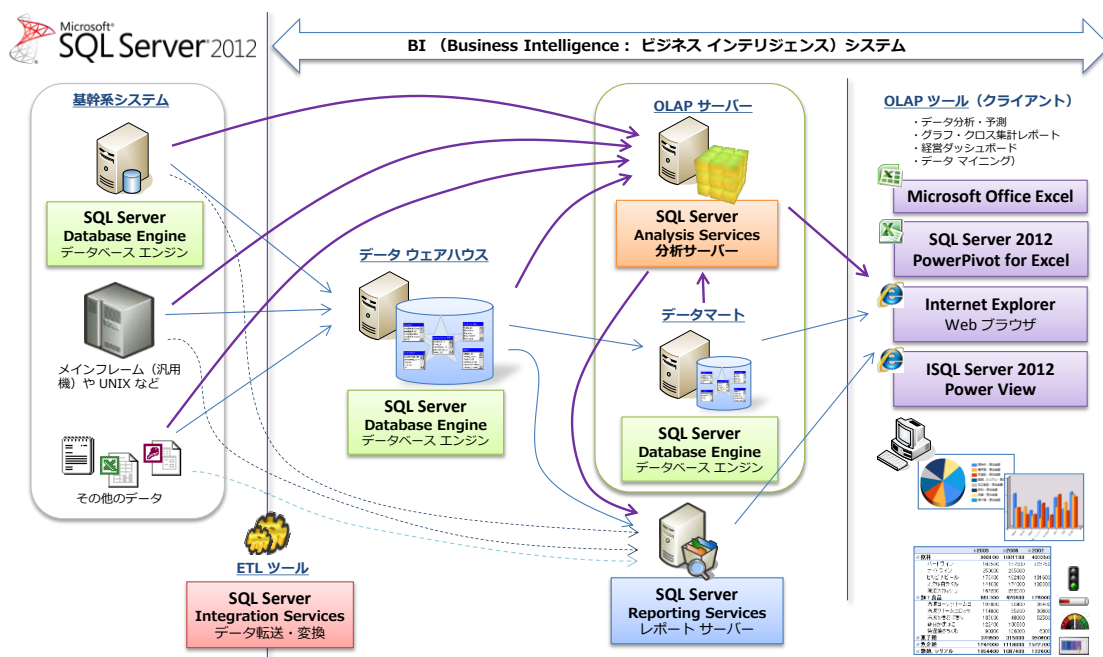
データ ウェアハウスの言葉の定義は、使う人によってさまざまで、狭い意味では「企業内に散在する業務システムのデータを抽出、変換、統合して蓄積したデータの集まり」(全社統合データ)を指し、「セントラル データ ウェアハ

ウス」と呼ばれています。これに対して、広い意味では「データの蓄積から活用までを含めた意思決定を支援するシステム全体」を指します（データマートや OLAP サーバーを含めて、あるいは人によっては OLAP ツールまでを含めてデータ ウェアハウスと呼ぶこともあります）。

最近では、「OLAP ツールを利用した高度なデータ分析までを含めたシステム全体」を指して、「BI」（Business Intelligence : ビジネス インテリジェンス）システムと呼ぶことが多くなり、データ ウェアハウスという言葉は、セントラル データ ウェアハウスやデータマートを指すことが多くなっています。なお、狭義での BI は、OLAP サーバーや OLAP ツールを指します。

## ➡ データ ウェアハウスと SQL Server の関係

SQL Server には、**Database Engine** や **Integration Services** だけでなく、**Reporting Services**（レポート サーバー）や **Analysis Services**（多次元分析・インメモリ分析サーバー）、**PowerPivot**（データ分析ツール）、**Power View**（より使いやすい分析ツール）など、いろいろな機能が提供されています。これらの SQL Server のサービスと、一般的なデータ ウェアハウスの構成を当てはめると、次のようになります。



SQL Server の **Database Engine Services**（RDB Engine : リレーショナル データベース エンジン）は、基幹系システムやセントラル データウェアハウス、データマートにおけるデータの格納先として、十二分に利用することができます。実際、SQL Server を利用したテラ バイト (TB) サイズのデータ ウェアハウスを構築している企業は多数あり、筆者もそういった企業のコンサルティング（物理設計や論理設計、BI システム設計など）を行った経験があり、パフォーマンス良く、安定稼働しているシステムを目の当たりにしてきています。

また、2 つ前のバージョンの SQL Server 2008 からは、データ ウェアハウス関連の機能（特にパフォーマンス）が大きく向上しており、より大規模かつミッション クリティカルなデータベース環境への対応を強化しています。

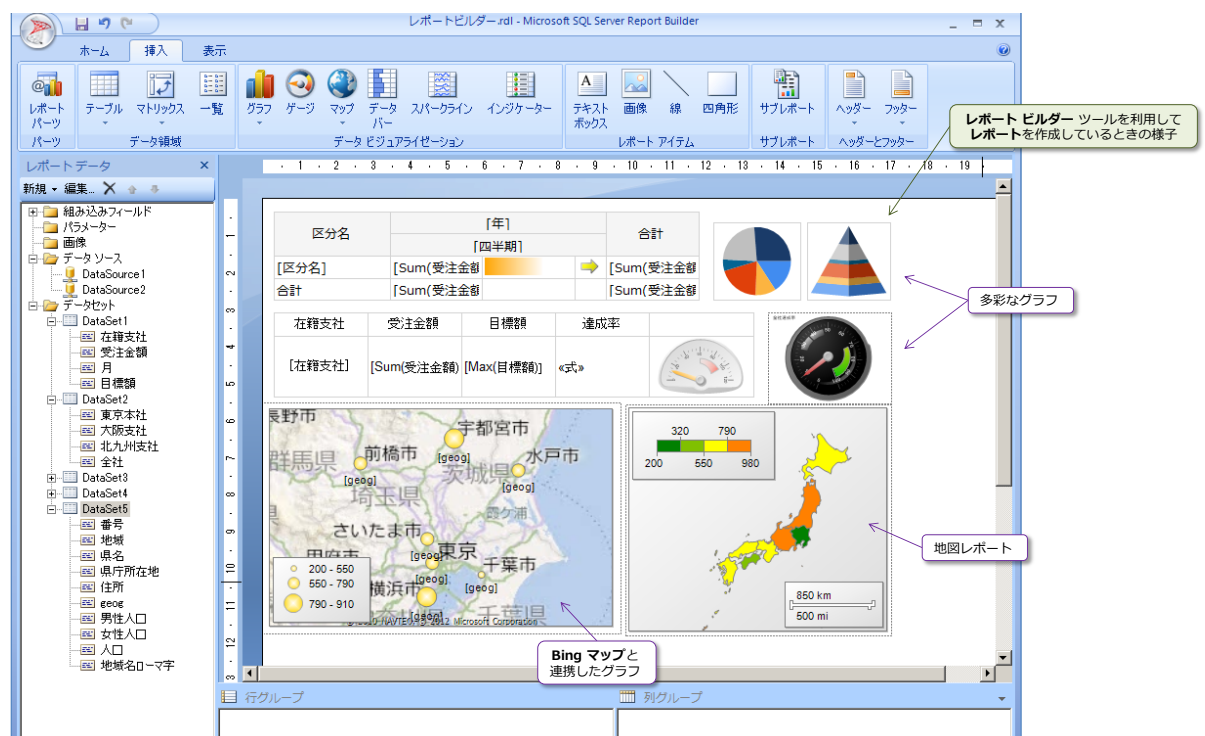
## 4.5 BI システムとしての SQL Server

### ➡ BI システムとしての SQL Server

SQL Server は、SQL Server 7.0 で**データ転送・変換機能**である DTS（現在の Integration Services）、**データ分析サーバー機能**である OLAP Services（現在の Analysis Services）を提供して以来、SQL Server 2000 から提供された**データマイニング機能**、アドオンとして提供された**レポートサーバー機能**である Reporting Services、SQL Server 2008 R2 から提供された**インメモリのデータ分析機能**の PowerPivot など、BI（ビジネス インテリジェンス）および DWH（データ ウェアハウス）まわりの機能がバージョンが上がることに飛躍的に進化しています。

弊社のお客様でも、業務システムは Oracle やメインフレーム、そこからデータを抽出してデータ分析レポートを作成する BI/DWH システムには SQL Server を採用している、というところが多いです。もちろん、業務システムに SQL Server、BI システムも SQL Server というお客様もいらっしゃいますし、弊社のコンサルティング サービス（BI/DWH システム構築支援サービス）へのお問い合わせもここ数年非常に増えています。

こうした BI システムとしての SQL Server の人気を支える理由の 1 つには、「**Reporting Services による強力なレポート機能**」があります。レポート作成ツールである「**レポートビルダー**」を利用すれば、次のようなレポートを簡単に作成することができます。



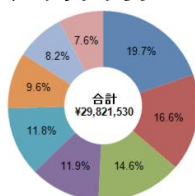
## レポート ビルダーで作成したレポート例

表形式のレポート（データバーやインジケーター、スパークラインも配置）

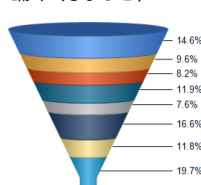
区分名	2005								合計	
	Q1		Q2		Q3		Q4			
飲料	¥355,900		¥513,600		¥411,900		¥486,000		¥1,767,400	
加工食品	¥244,700		¥264,300		¥285,000		¥359,800		¥1,153,800	
菓子類	¥330,400		¥174,400		¥181,800		¥217,400		¥904,000	
魚介類	¥485,200		¥506,600		¥671,200		¥437,800		¥2,100,800	
穀類、シリアル	¥407,600		¥282,100		¥444,800		¥448,300		¥1,582,800	
調味料	¥235,300		¥596,000		¥613,900		¥570,200		¥2,015,400	
肉類	¥545,800		¥247,200		¥163,700		¥123,500		¥1,080,200	
乳製品	¥297,000		¥193,800		¥227,900		¥276,600		¥995,300	
合計	¥2,901,900		¥2,778,000		¥3,000,200		¥2,919,600		¥11,599,700	

## Reporting Services による多彩なグラフ

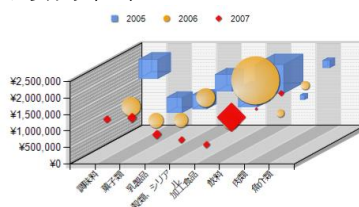
ドーナツ グラフ



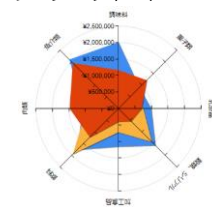
漏斗（じょうご）



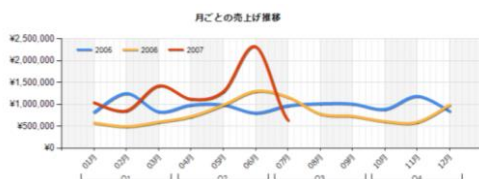
バブル チャート



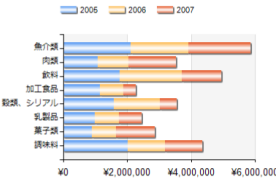
レーダー チャート



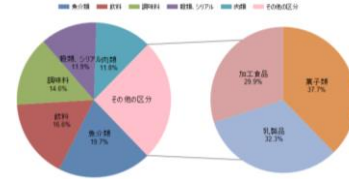
平滑折れ線



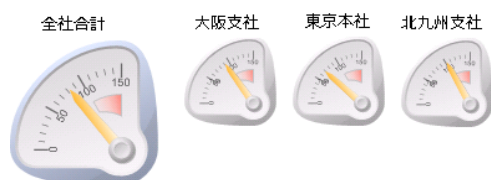
積み上げ横棒



補助円グラフ



## ゲージで目標達成率、傾向を表示



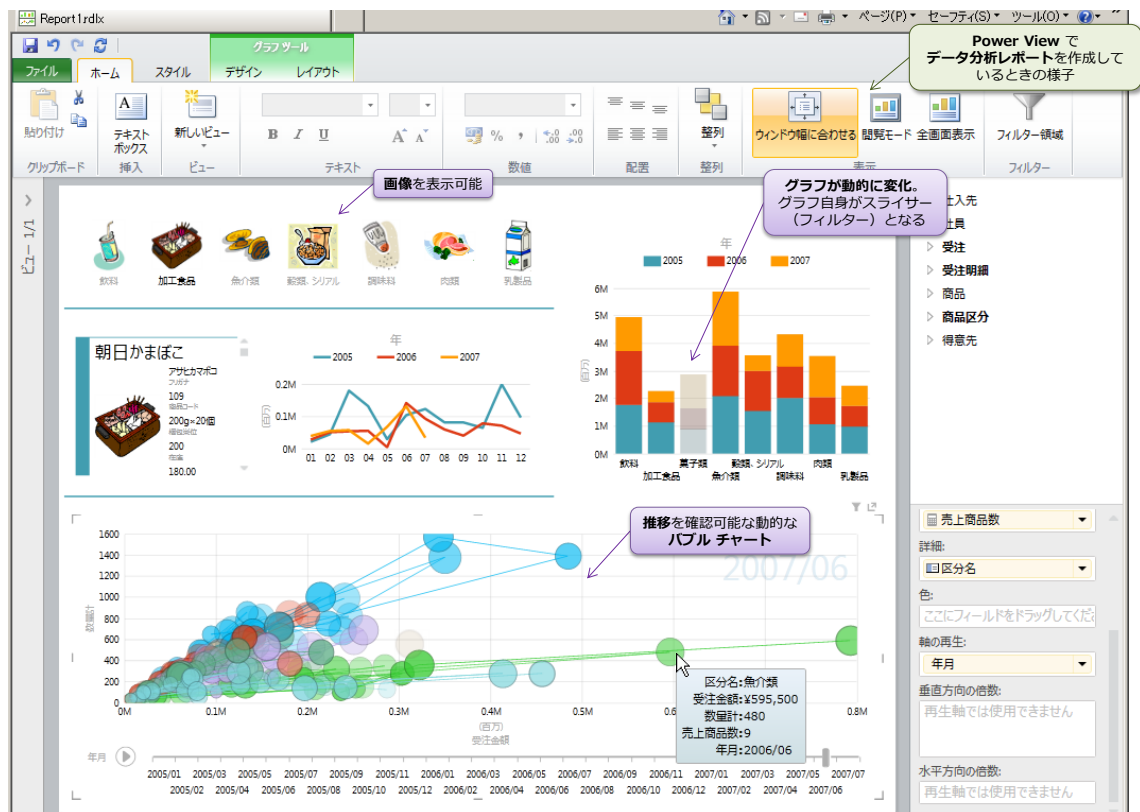
	売上金額	売上目標	目標達成率	傾向
大阪支社	¥13,737,180	¥15,000,000	91.6%	
東京本社	¥13,929,650	¥20,000,000	69.6%	
北九州支社	¥7,154,700	¥7,000,000	102.2%	
全社合計	¥34,821,530	¥42,000,000	82.9%	

表形式のレポートから、Excel でお馴染みのデータバーやインジケーター、スパークライン、各種グラフ（円、棒、円錐、バブル、散布図、レーダーチャート、ゲージなど）、地図レポートなどを作成することができます。売上分析レポートや予算管理、ABC 分析、スコアカード、商圏分析、電力利用状況の見える化など、さまざまなデータ分析レポートを容易に作成できるのが特徴です。

## ➡ Power View による動的レポート作成

SQL Server 2012 からは、**Power View** と呼ばれる、新しいデータ分析／レポートニング ツールが提供されました。Power View を利用すると、レポートビルダーや後述の PowerPivot for Excel ツールよりも、容易に見栄えの良いデータ分析レポートを作成することが可能です。次の画面は、Power View を利用してデータ分析レポートを作成している様子です。





Power View の特徴は、次のとおりです。

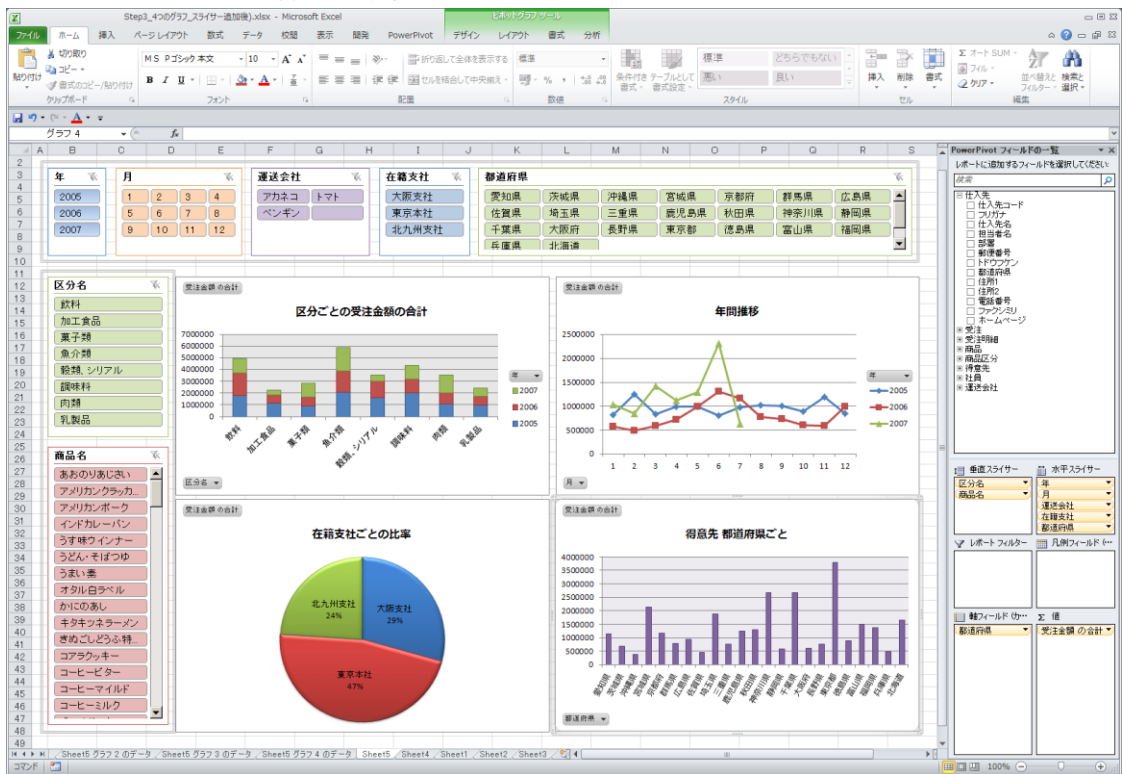
- 直感的な操作でレポートを作成できる
- レポート内に画像を簡単に配置できる
- グラフ同士を連動させることができる（グラフ自身がスライサーとなって、グラフ同士が選択された値によって動的に変化する）
- 再生可能なバブル チャートを作成できる（時間によって変化する値をアニメーションで確認可能。軌跡を表示できる）

## ➡ PowerPivot による高度なデータ分析

BI システムとしての SQL Server の人気を支えるもう 1 つの理由には、SQL Server 2008 R2 から提供された「**PowerPivot**」機能があります。PowerPivot には、クライアント側ツールの「**PowerPivot for Excel**」と SharePoint Server と連携できる「**PowerPivot for SharePoint**」の 2 種類があります。

PowerPivot（パワー ピボット）という名のとおり、従来の Excel のピボット テーブル機能を大きく進化させたようなツールで、これを利用すると、次のような高度なデータ分析レポートを簡単に作成することができます。

## PowerPivot for Excel による高度なデータ分析



また、**PowerPivot for SharePoint** を利用して、SharePoint Server 上へレポートを配置すれば、次のように共有することも可能です。

PowerPivot for SharePoint によるデータ分析レポートの共有（Web ブラウザーからレポートの参照が可能）





## ➡ ビッグデータ対応のインメモリ BI

PowerPivot は、すべてのデータをメモリ内へ配置して動作する「**インメモリ BI**」機能です。

「**xVelocity エンジン**」と呼ばれるエンジンを採用して、カラムベースでデータを格納し、高度な圧縮技術を利用しています。SQL Server 2012 からは、PowerPivot のサーバー版として、Analysis Services の「**テーブル モード**」(Tabular Mode : 表形式モード)が搭載され、**ビッグデータへの対応**(より大量のデータのサポート)、パーティショニング対応、行レベルのセキュリティ機能などが提供されました。

このように SQL Server の BI まわりの機能は、バージョンが上がるごとに大きく進化して、現場で "**使える**" 機能がどんどん増えています。今後ますます SQL Server の BI システムでの活用が増えていくのではないのでしょうか。

## ➡ おわりに

最後まで試された皆さま、いかがでしたでしょうか。SQL Server 2012 の Integration Services は、業務で大変役立つツールですので、ぜひ活用してみてくださいと思います。今回は、「**入門編**」ということで、基本操作のみをご紹介しましたが、Integration Services では、条件分割や文字マップ、ログ記録、イベント ハンドラー、チェックポイント、スクリプト コンポーネントによる複雑な変換処理とグローバル変数の利用、Integration Services サービスへの接続など、まだまだたくさんの機能が提供されています。これらの応用的な利用方法については、本自習書シリーズの「**Integration Services 応用**」編で説明していますので、こちらもぜひご覧いただければと思います。

## 執筆者プロフィール

**有限会社エスキューエル・クオリティ** (<http://www.sqlquality.com/>)

SQLQuality (エスキューエル・クオリティ) は、日本で唯一の **SQL Server 専門の独立系コンサルティング会社**です。過去のバージョンから最新バージョンまでの SQL Server を知りつくし、多数の実績と豊富な経験を持つ、OS や .NET にも詳しい **SQL Server の専門家 (キャリア 17 年以上) がすべての案件に対応します**。人気メニューの「**パフォーマンス チューニング サービス**」は、100%の成果を上げ、過去すべてのお客様環境で驚異的な性能向上を実現。チューニング スキルは**世界トップレベル**を自負、検索エンジンでは (英語情報を含めて) ヒットしないノウハウを多数保持。ここ数年は **BI/DWH システム構築支援**のご依頼が多い。

### 主なコンサルティング実績

- ▶ 大手映像制作会社の **BI システム構築支援** (会計/業務システムにおける予算管理/原価管理など)
- ▶ 大手流通系の **DWH/BI システム構築支援** (POS データ/在庫データ分析)  
大規模テラバイト級データ ウェアハウスの物理・論理設計支援および運用管理設計支援
- ▶ 大手アミューズメント企業の **BI システム構築支援** (人事システムにおける人材パフォーマンス管理)
- ▶ 外資系医療メーカーの Analysis Services による「**販売分析**」システムの構築支援 (売上/顧客データ分析)
- ▶ **9 TB** データベースの物理・論理設計支援 (パーティショニング対応など)
- ▶ ハードウェア リプレイス時の**ハードウェア選定** (最適なサーバー、ストレージの選定)、**高可用性環境**の構築
- ▶ **SQL Server 2000** (32 ビット) から **SQL Server 2008** (x64) への移行/アップグレード支援
- ▶ 複数台の SQL Server の **Hyper-V 仮想環境**への移行支援 (サーバー統合支援)
- ▶ **2 時間**かかっていた日中バッチ実行時間を、わずか **5 分**へ短縮 (**95.8%** の性能向上)
- ▶ ピーク時の CPU 利用率 **100%** のシステムを、わずか **10%** にまで軽減し、大幅性能向上
- ▶ 平均 **185.3ms** かかっていた処理を、わずか **39.2ms** へ短縮 (**78.8%** の性能向上)
- ▶ **Java 環境** (Tomcat, Seasar2, S2Dao) の SQL Server パフォーマンス チューニング etc

### コンサルティング時の作業例 (パフォーマンス チューニングの場合)

- ▶ アプリケーション コード (VB, C#, Java, ASP, VBScript, VBA) の解析/改修支援
- ▶ ストアド プロシージャ/ユーザー定義関数/トリガー (Transact-SQL) の解析/改修支援
- ▶ インデックス チューニング/SQL チューニング/ロック処理の見直し
- ▶ 現状のハードウェアで将来のアクセス増にどこまで耐えられるかを測定する高負荷テストの実施
- ▶ IIS ログの解析/アプリケーション ログ (log4net/log4j) の解析
- ▶ ボトルネック ハードウェアの発見/ボトルネック SQL の発見/ボトルネック アプリケーションの発見
- ▶ SQL Server の構成オプション/データベース設定の分析/使用状況 (CPU, メモリ, ディスク, Wait) 解析
- ▶ 定期メンテナンス支援 (インデックスの再構築/断片化解消のタイミングや断片化の事前防止策など) etc

## 松本美穂 (まつもと・みほ)

有限会社エスキューエル・クオリティ 代表取締役

Microsoft MVP for SQL Server (2004 年 4 月~)

経産省認定データベース スペシャリスト/MCDBA/MCSD for .NET/MCITP Database Administrator

SQL Server の日本における最初のバージョンである「SQL Server 4.21a」から SQL Server に携わり、現在、SQL Server を中心とするコンサルティングを行っている。得意分野はパフォーマンス チューニングと Reporting Services。コンサルティング業務の傍ら、講演や執筆も行い、マイクロソフト主催の最大イベント Tech・Ed などでもスピーカーとしても活躍中。SE や ITPro としての経験はもちろん、記名/無記名含めて多くの執筆実績も持ち、様々な角度から SQL Server に携わってきている。著書の『SQL Server 2000 でいってみよう』と『ASP.NET でいってみよう』(いずれも翔泳社刊)は、トップ セラー (前者は 28,500 部、後者は 16,500 部発行)。近刊に『SQL Server 2012 の教科書』(ソシム刊)がある。

## 松本崇博 (まつもと・たかひろ)

有限会社エスキューエル・クオリティ 取締役

Microsoft MVP for SQL Server (2004 年 4 月~)

経産省認定データベース スペシャリスト/MCDBA/MCSD for .NET/MCITP Database Administrator

SQL Server の BI システムとパフォーマンス チューニングを得意とするコンサルタント。過去には、約 3,000 本のストアド プロシージャのチューニングや、テラバイト級データベースの論理・物理設計、運用管理設計、高可用性設計、BI・DWH システム設計支援などを行う。アプリケーション開発 (ASP/ASP.NET, C#, VB 6.0, Java, Access VBA など) やシステム管理者 (IT Pro) 経験もあり、SQL Server だけでなく、アプリケーションや OS、Web サーバーを絡めた、総合的なコンサルティングが行えるのが強み。Analysis Services と Excel による BI システムも得意とする。マイクロソフト認定トレーナー時代の 1998 年度には、Microsoft CPLS トレーナー アワード (Trainer of the Year) を受賞。