



SQL Server から SQL Anywhere へのマイグレーション

A whitepaper from Sybase iAnywhere

このホワイトペーパーは、SQL Anywhere 12 をベースに書かれていますが、ほとんどの部分で過去のバージョンや将来のバージョンでも利用可能です。また、英語のホワイトペーパーをそのまま翻訳しているため、製品画面ショットなどは英語版ソフトウェアのもので、SQL Anywhere は、日本語にローカライズされています。

目次

A whitepaper from Sybase iAnywhere	1
目次	2
はじめに	3
必要なソフトウェア	3
概要	3
Northwind SQL Server データベース用の ODBC データソースの作成	4
新しい SQL Anywhere Database の作成	4
SQL Server から SQL Anywhere へのスキーマとデータの移行	6
移行したデータベースのスキーマとデータの確認.....	16
ビューの移行.....	18
データベースロジックの移行	25
ストアードプロシージャ	25
ユーザー定義関数とトリガー	31
適切な動作の確認と検証.....	31
まとめ.....	32

はじめに

このチュートリアルでは、Microsoft SQL Server データベースを SQL Anywhere データベースに移行する方法について、実例とともに説明します。スキーマとデータを移行するには、SQL Anywhere に用意されているデータベース移行ウィザードを使用します。データベースロジック（ストアドプロシージャ、ユーザー定義関数、およびトリガー）を移行するには、T-SQL コードに若干の修正を加えて、適切に動作するようにします。このチュートリアルでは、SQL Server のサンプルデータベース "Northwind" を新しい SQL Anywhere データベースに移行します。

必要なソフトウェア

- SQL Anywhere 12.0.1 以降
- Microsoft SQL Server (2008 R2 Express バージョンでテスト済みですが、他のバージョンでも有効です)
- **Northwind サンプルデータベース**

Northwind サンプルデータベースが実際の SQL Server インストールで適切にアタッチおよび構成されていることを確認してください。

SQL Server が混合モード認証で構成され、システム管理 "sa" アカウントが有効になっているという前提です。

このチュートリアルは、Windows オペレーティングシステム向けに作成しています。

概要

このチュートリアルで扱う範囲は、次のとおりです。

- Northwind SQL Server データベース用の ODBC DSN の作成
- Sybase Central による新しい SQL Anywhere データベースの作成
- データベース移行ウィザード による SQL Server データベースからのスキーマとデータの移行
 - SQL Server を参照するリモートデータベースサーバーの作成
 - 新しい SQL Anywhere データベースへのスキーマとデータの移行
- SQL Anywhere データベース内部へのストアドプロシージャとその他のデータベースロジックの生成を目的とする、SQL コードの修正

SQL Server には、AdventureWorks という高度なサンプルデータベースも用意されています。このデータベースの SQL Anywhere バージョン



ジョンは、[こちら](#)から入手できます。

Northwind SQL Server データベース用の ODBC データソースの作成

SQL Anywhere データ移行ツールは、SQL Server データベースへの ODBC 接続を必要とします。Sybase Central は ODBC データソースに基づいて「リモートデータベースサーバー」を作成し、SQL Anywhere 管理ツール (Sybase Central と Interactive SQL) による SQL Server データベースへのアクセスとクエリの実行を可能にします。この機能は、Remote Data Access と呼ばれています。

1. ODBC Administrator を開いて ([スタート] -> [すべてのプログラム] -> [SQL Anywhere 12] -> [Administration Tools] -> [ODBC Data Source Administrator])、**[Add]** をクリックします。
2. リストから [SQL Server] を選択して、**[Finish]** をクリックします。
3. [Microsoft SQL Server DSN Configuration wizard] が表示されます。次のデータベース接続オプションを入力します (ウィザードのページ間を移動するには **[Next]** または **[Back]** をクリックします)。
 - データソース名に *Northwind-SQLServer* を指定します。
 - 接続先の SQL Server インスタンスのサーバー名または IP アドレスを指定します。
 - 適切なクライアント構成 (TCP/IP、サーバー別名など) を指定します。
 - 適切なログイン資格情報 (ログイン ID とパスワード) を指定します。
 - デフォルトのデータベースを *Northwind* に変更します。
 - その他のオプションはすべて、デフォルト設定を使用します。
4. **[Finish]** をクリックし、ウィザードを終了します。**[OK]** をクリックし、[SQL Server Setup] ダイアログを閉じます。
5. **[OK]** をクリックし、ODBC Administrator を閉じます。

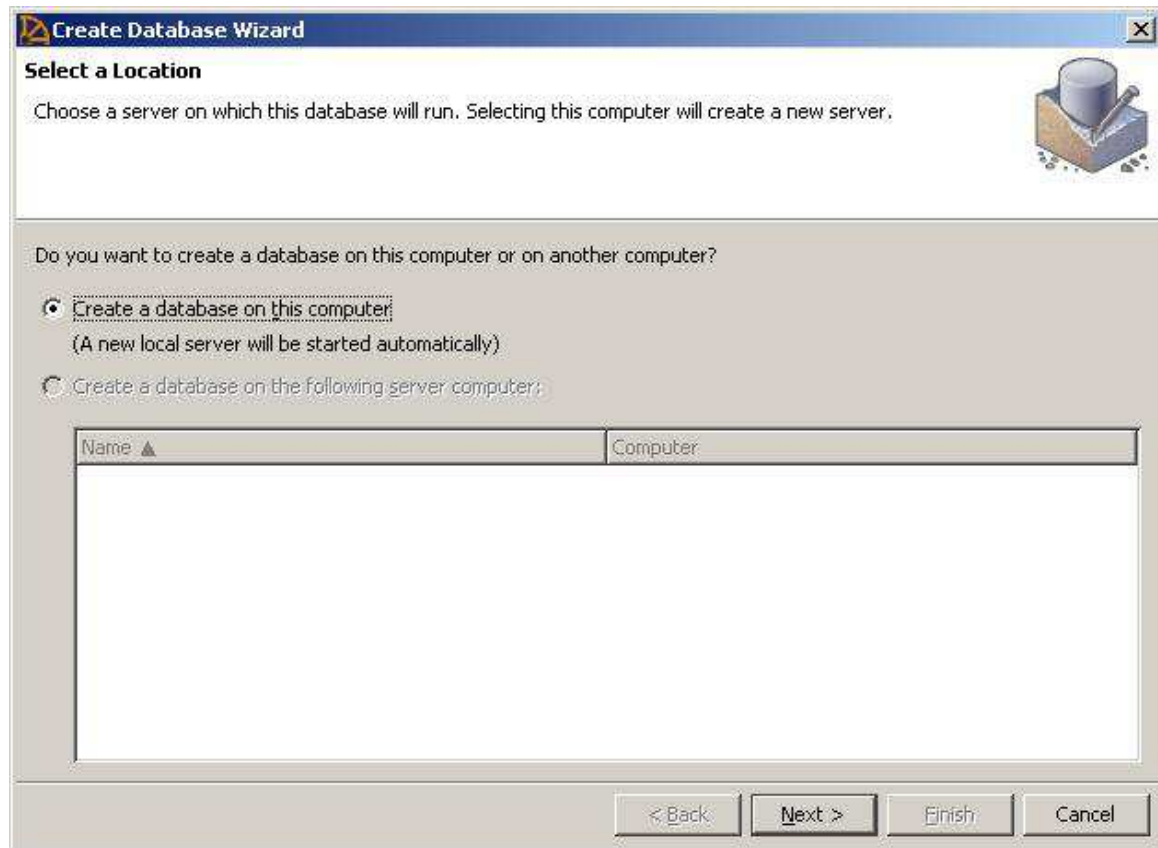
新しい SQL Anywhere データベースの作成

以降の操作はすべて、SQL Anywhere の管理ツール Sybase Central で実行します。

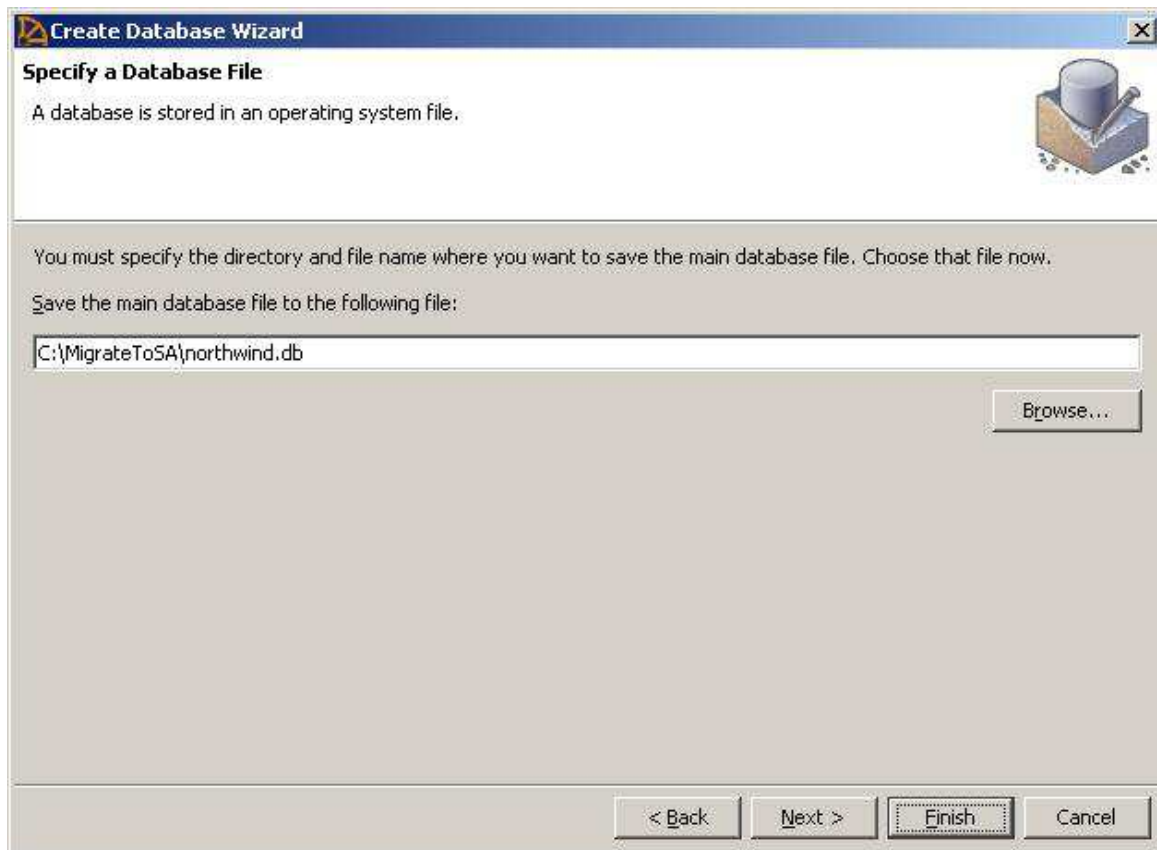
1. Sybase Central を起動します ([スタート] -> [すべてのプログラム] -> [SQL Anywhere 12] -> [Administration Tools] -> [Sybase Central])。[Tips] ダイアログおよび/または [Welcome] ダイアログを閉じます (表示された場合)。
2. [Tools] メニューから、[SQL Anywhere 12] -> [Create Database] を選択します。[Create Database Wizard] が表示されます。



3. **[Create a database on this computer]** を選択して、**[Next]** をクリックします。



4. データベースファイルを *northwind.db* という名前で保存します。



5. この時点で、新しいデータベースのデフォルト設定（ページサイズ、暗号化、大文字小文字の区別など）を変更できますが、このチュートリアルでは省略します。[Finish] をクリックし、データベースを作成します。
6. データベースが正常に作成されたら、[Close] をクリックし、[Create Database Wizard] を閉じます。

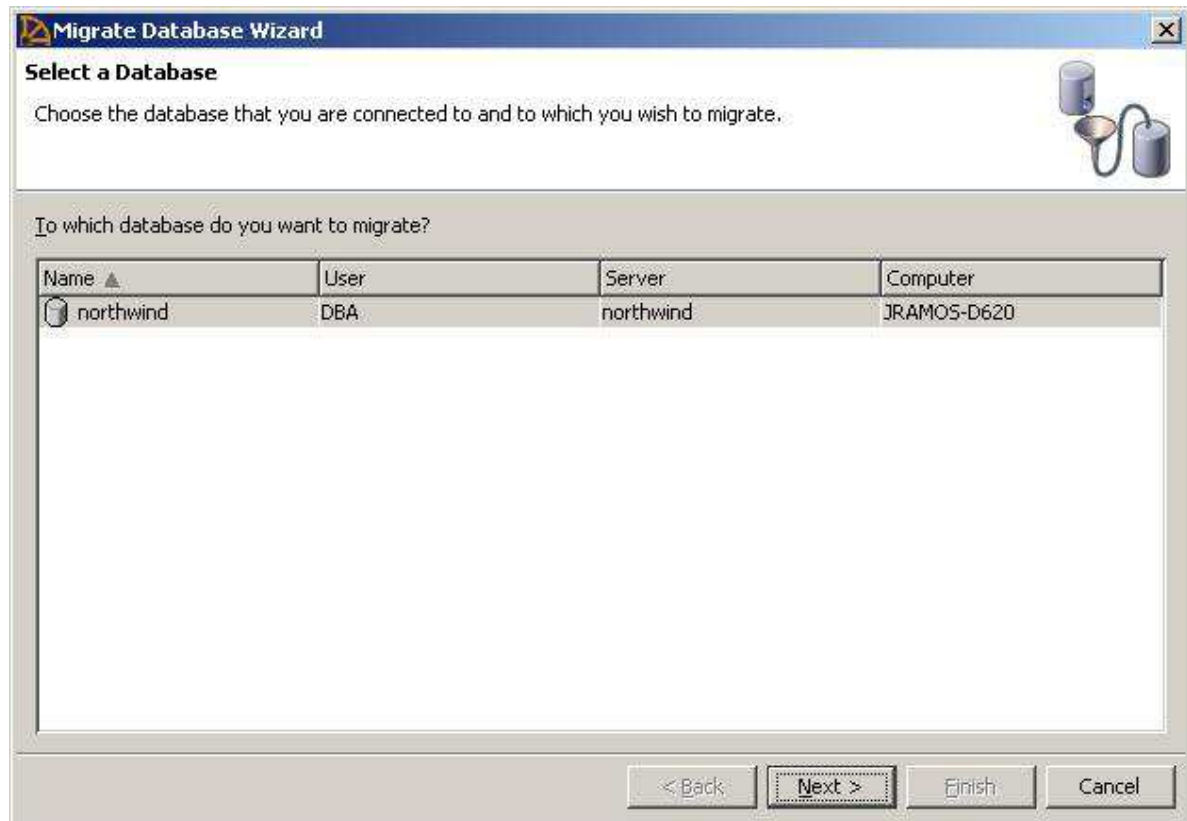
Sybase Central は、新しく作成した "northwind" SQL Anywhere データベースに自動的に接続します。その際には、デフォルトのユーザーアカウント "DBA" が使用されます。このデータベースは、初期状態では空です（スキーマやデータは存在しません）。

注： 別の方法として、dbinit コマンドラインユーティリティで新しい SQL Anywhere データベースを作成してから、Sybase Central を使用して、そのデータベースに接続するという方法もあります。

SQL Server から SQL Anywhere へのスキーマとデータの移行

SQL Server データベースから、新しく作成した SQL Anywhere データベースにスキーマとデータを移行するには、データベース移行ウィザードを使用します。このウィザードに従って、ソース（SQL Server）データベースのスキーマと適合するテーブル、インデックス、キーのリレーションシップをすべて作成します。その後、すべてのローをソースからターゲット（SQL Anywhere）データベースにコピーします。

1. Sybase Central で、[Tools] メニューから、[SQL Anywhere 12] -> [Migrate Database] を選択して、ウィザードを起動します。
2. "northwind" データベースを選択して、[Next] をクリックします。



3. 移行するデータベースオブジェクトは、"northwind" SQL Server データベースに格納されています。テキストボックスに *northwind* と入力して、[Create Remote Server Now] をクリックします。

Migrate Database Wizard

Select a Remote Server

Choose the remote server that you wish to migrate. You may create the remote server now, if needed.

From which remote server do you want to migrate?

Name	Server Type	Conn. Type	Connection Info.
------	-------------	------------	------------------

Create Remote Server Now... Properties...

Optionally, what is the name of the remote database from which you want to migrate?

northwind

< Back Next > Finish Cancel

4. リモートサーバー名として *NorthwindMSSQLRemoteServer* を入力して、[Next] をクリックします。

Create Remote Server Wizard

Welcome

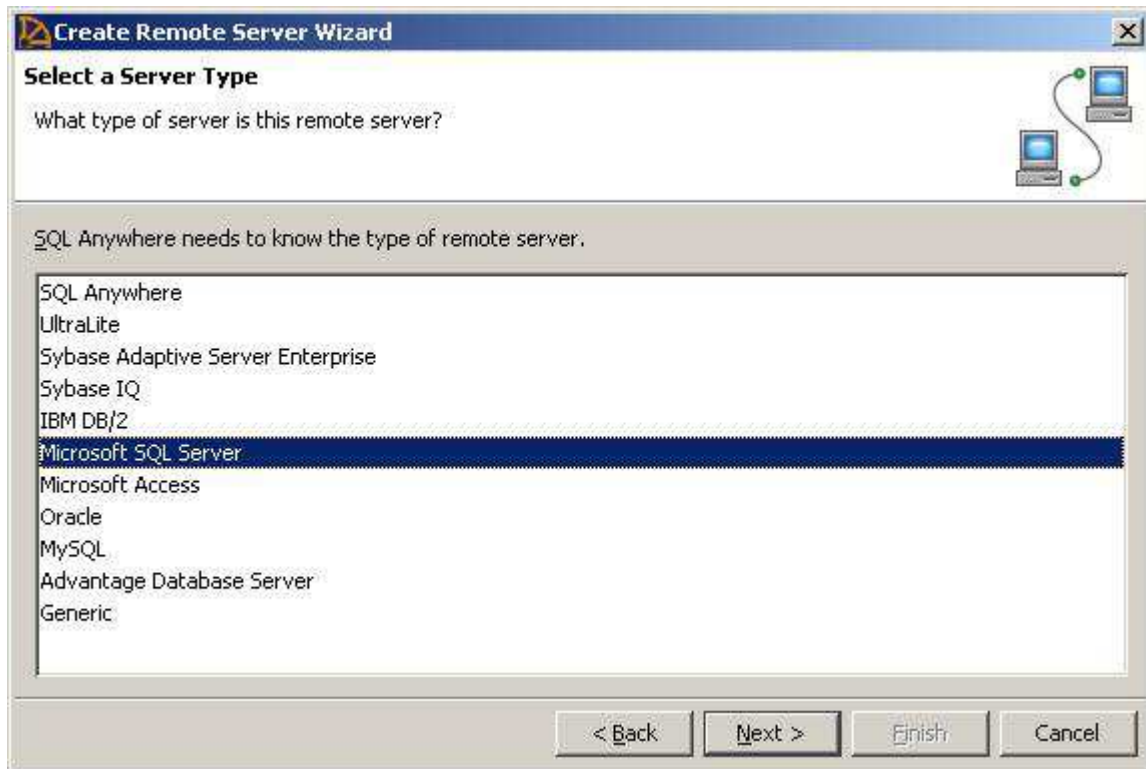
This wizard helps you create a new remote server. SQL Anywhere uses remote servers to give you access to data located on separate database servers.

What do you want to name the new remote server?

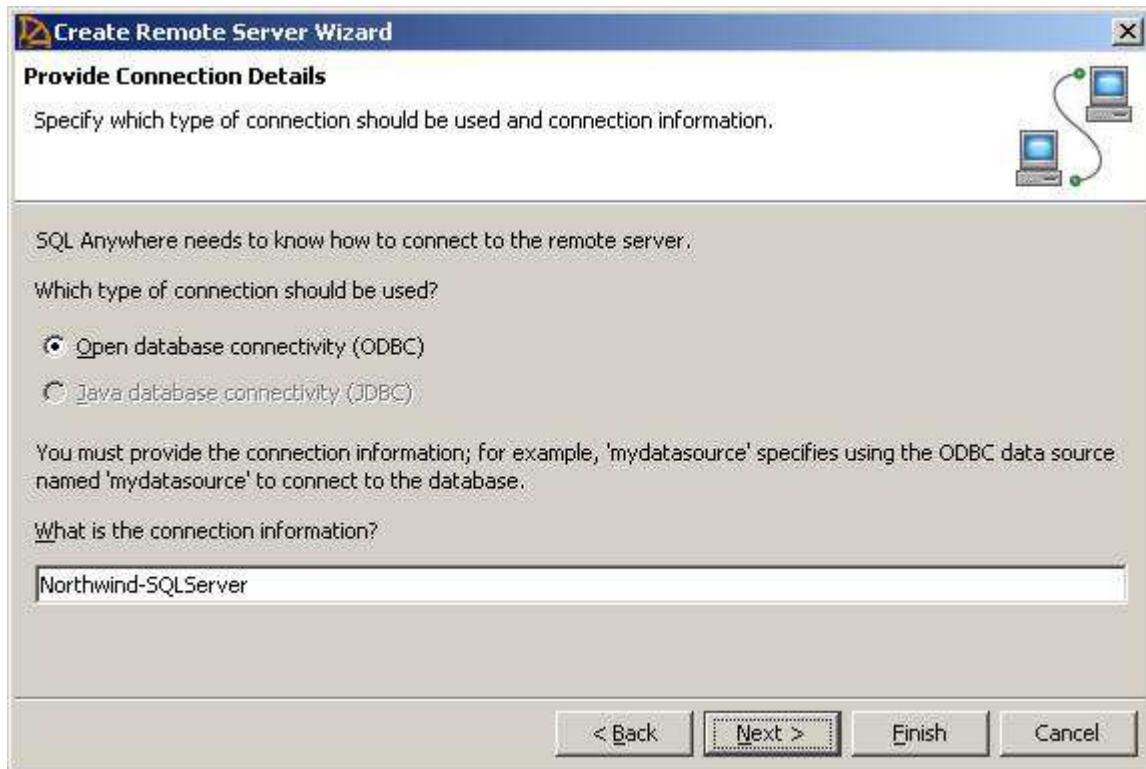
NorthwindMSSQLRemoteServer

< Back Next > Finish Cancel

5. リモートサーバータイプのリストから [Microsoft SQL Server] を選択して、**[Next]** をクリックします。



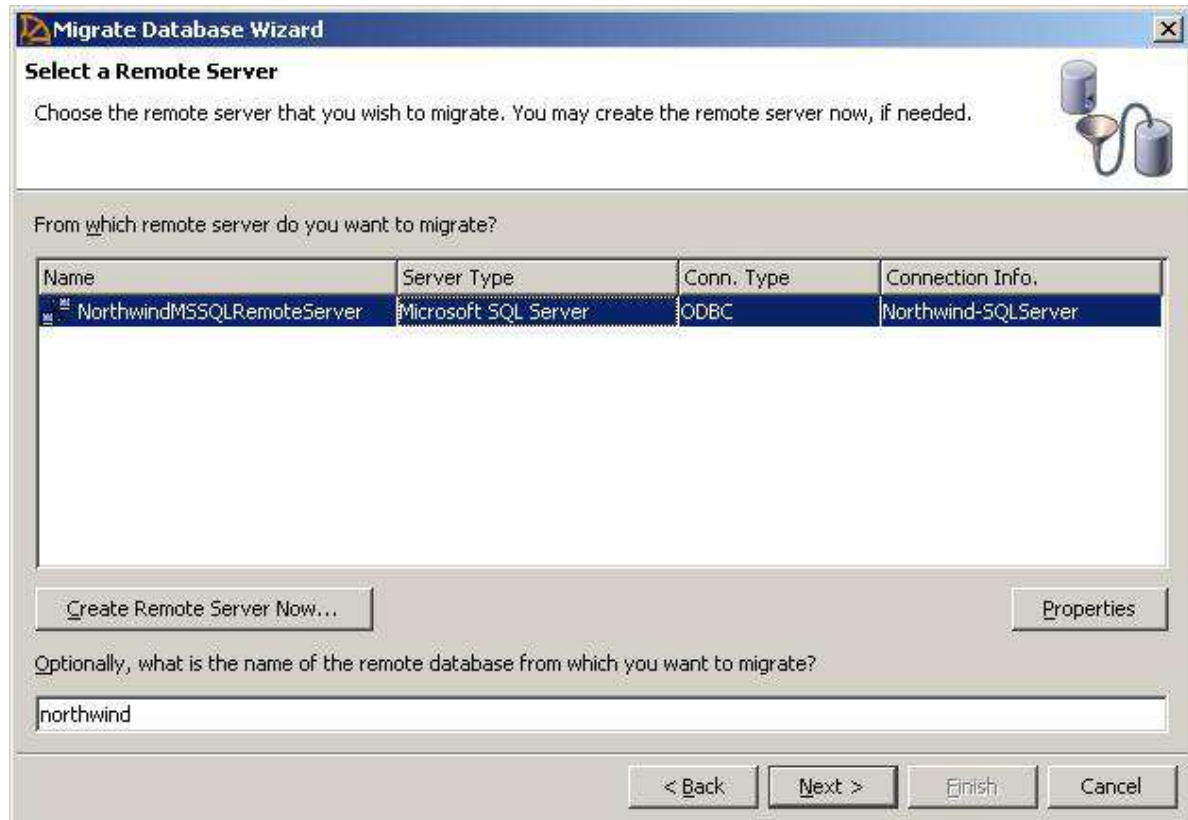
6. これで、ウィザードでは、ODBC を使用して SQL Server データベースに接続することになります。接続情報（先に作成した ODBC データソース）として *Northwind-SQLServer* を入力して、**[Next]** をクリックします。



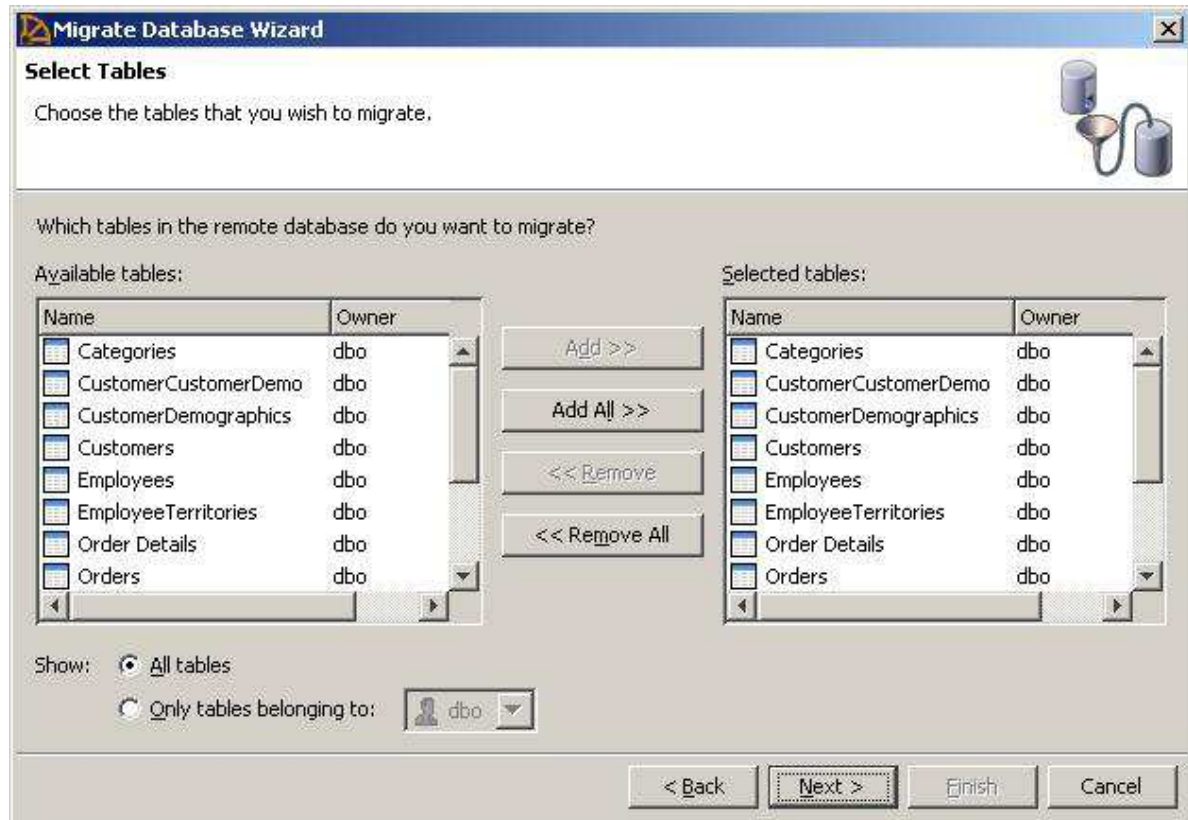
7. 選択すれば、リモートサーバーを読み取り専用データソースとして設定できます。今回の場合では、デフォルト設定のままにして、
[Next] をクリックします。
8. SQL Anywhere データベースへの接続は、ユーザー名 "DBA" が使用されます。リモートサーバー (SQL Server) では、このユーザーは認識されないため、外部ログインを作成して、(SQL Anywhere の) DBA ユーザーを、Northwind データベーススキーマを操作できる SQL Server ユーザー (今回の場合では、"sa") にマッピングさせる必要があります。**[Create an external login for the current user]** をオンにして、SQL Server のデータベースサーバーにログインするための適切な資格情報を入力します。**[Next]** をクリックします。

The screenshot shows a Windows-style dialog box titled "Create Remote Server Wizard". The current step is "Specify an External Login". The instructions state: "Specify the username and password that will be used to connect to the remote server." Below this, a note explains that the user is currently connected as 'DBA' and that an external login must be created if the remote server does not have a user with the same password. A checkbox labeled "Create an external login for the current user" is checked. There are three input fields: "Login name:" with the text "sa", "Password:" with masked characters "***", and "Confirm password:" also with "***". A "Test Connection" button is located below the password fields. At the bottom of the dialog are four buttons: "< Back", "Next >" (which is highlighted with a dashed border), "Finish", and "Cancel".

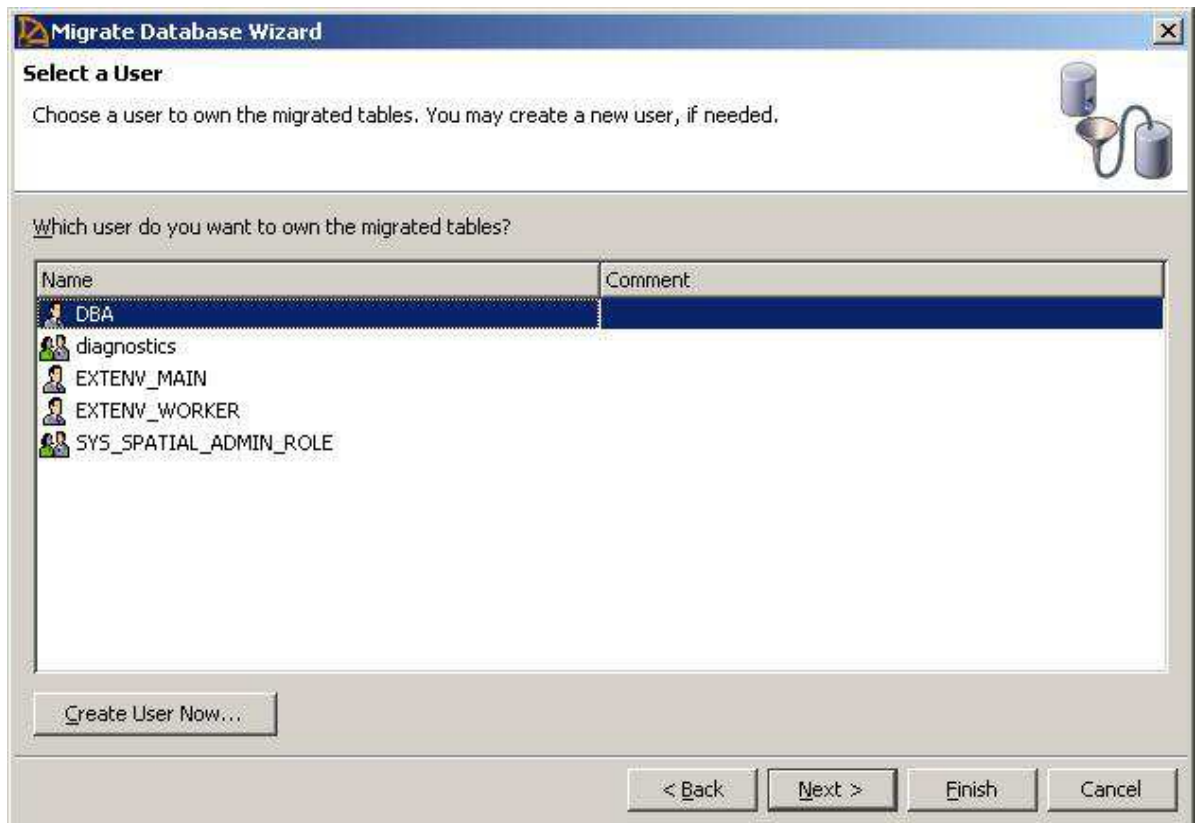
9. 概要ページが開き、リモートサーバーを作成するために SQL Anywhere で実行される SQL 文が表示されます。**[Finish]** をクリックし、リモートサーバーを作成します。
10. これで、データベース移行ウィザードに戻ります。新しく作成した "NorthwindMSSQLRemoteServer" を選択して、**[Next]** をクリックします。



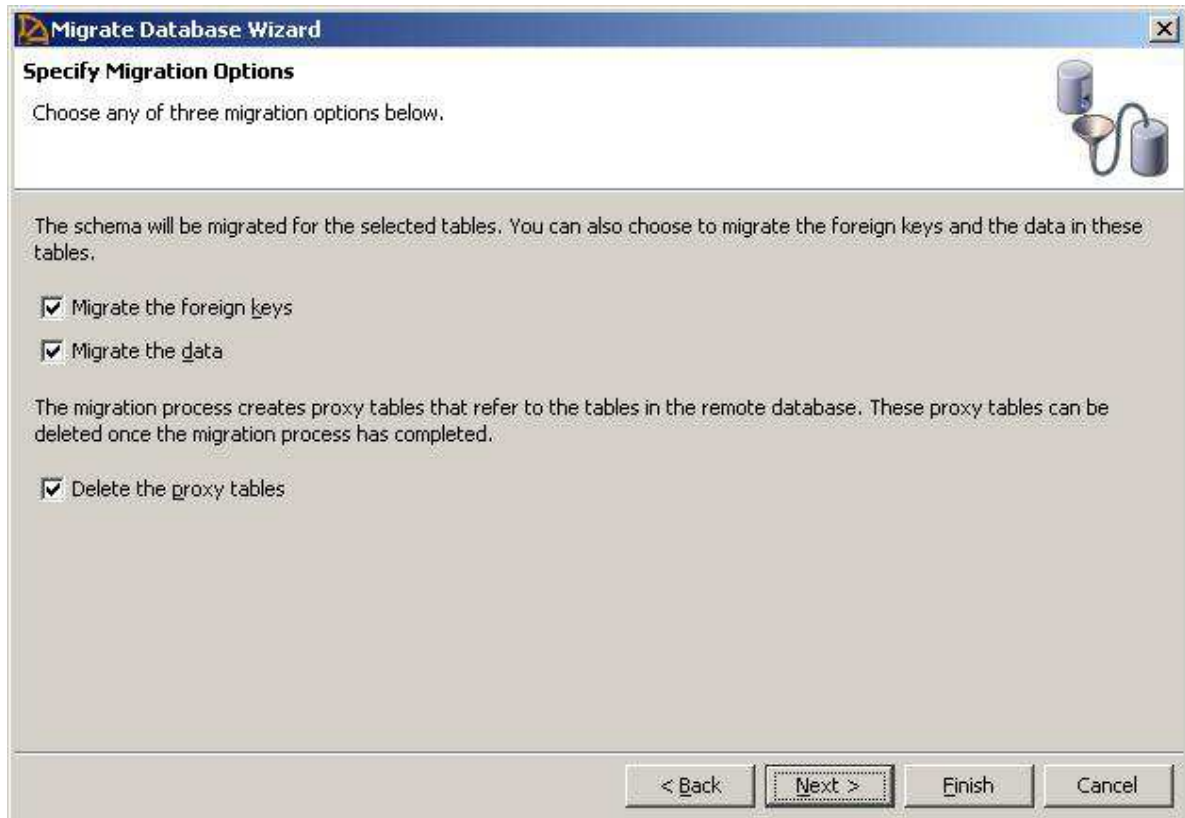
11. 左側のリストボックスに、Northwind SQL Server データベースのテーブルがすべて表示されます。テーブルをすべて移行するので、
[Add All]、[Next] の順にクリックします。



12. この時点で、選択すれば、"DBA" ユーザーまたは他のユーザーにテーブルを追加できます。わかりやすくするため、今回の場合では、テーブルを DBA ユーザーに追加します。DBA ユーザーを選択して、**[Next]** をクリックします。



13. [Specify Migration Options] ページが表示されます。外部キーおよび／またはデータを移行するオプションが用意されています。移行プロセスでは、プロキシテーブルを使用して、リモートデータベースサーバーのテーブルに対してクエリを実行します。オプションをすべてオンにしたままの状態、[Next] をクリックします。



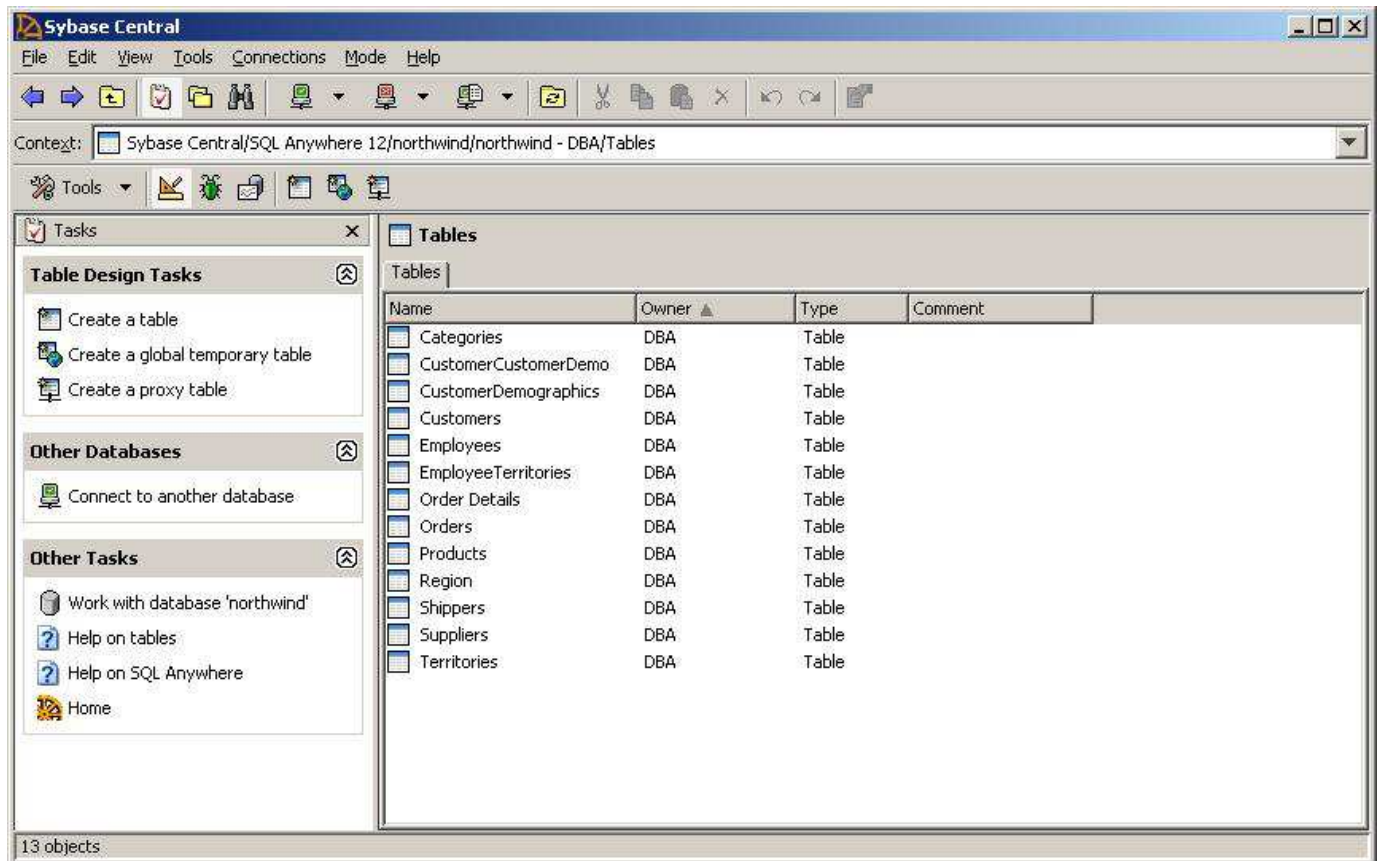
14. [Summary] ページが開き、スキーマとデータを移行するために実行される SQL 文が表示されます。各文はシステムストアプロシージャーで構成されており、データベース移行ウィザードによって実行されます。ただし、アプリケーション内で実行することも可能です。特定のテーブルやリモート外部キーのマッピングを移行する必要がある場合は、移行システムストアプロシージャーが役立ちます。



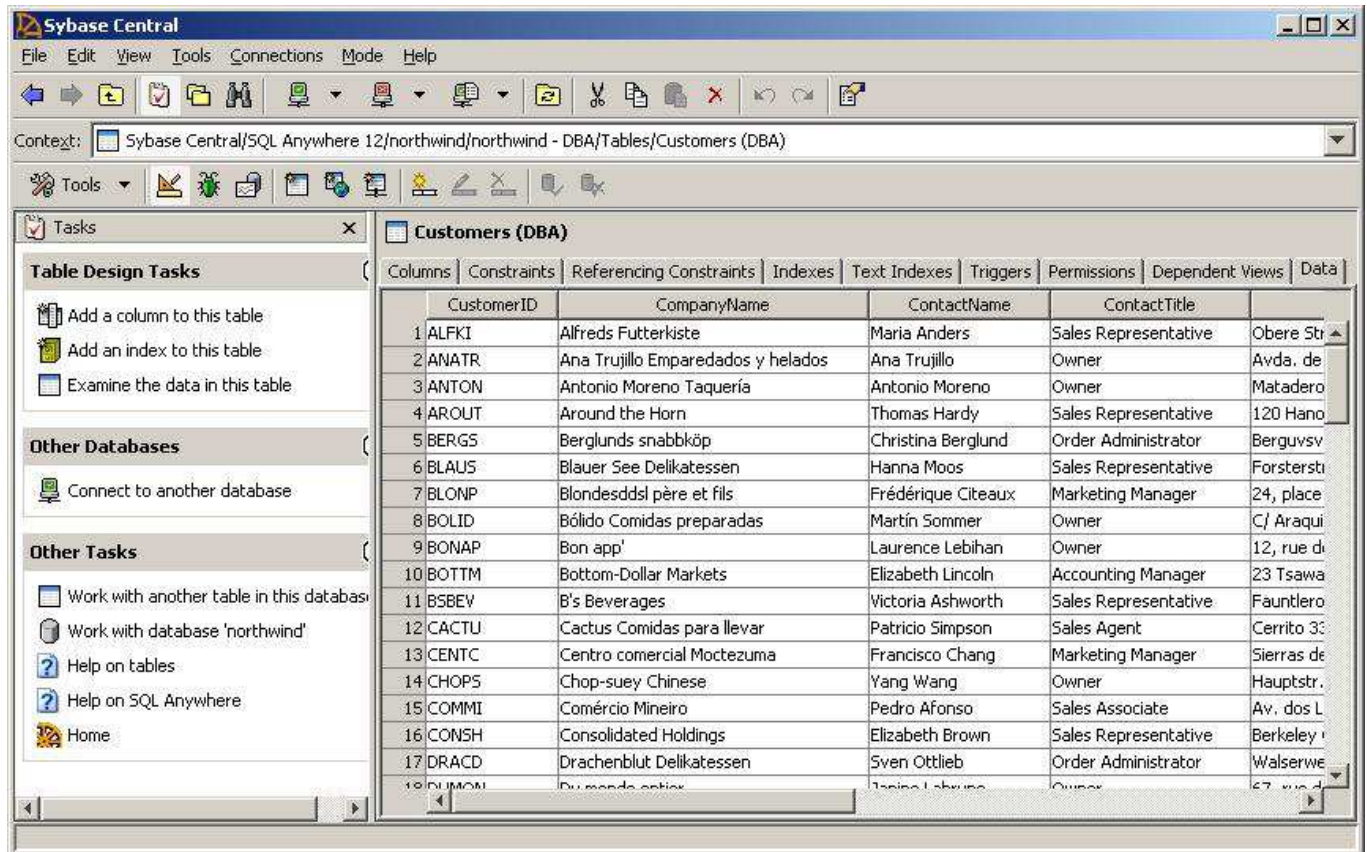
15. **[Finish]** をクリックし、移行を開始します。データベース移行ウィザード によって、Northwind SQL Server データベースから Northwind SQL Anywhere データベースにスキーマとデータが移行されます。処理が正常に完了したら、**[Close]** をクリックし、ウィザードを閉じます。

移行したデータベースのスキーマとデータの確認

1. Sybase Central で、右ペインの **[Tables]** をダブルクリックします。Northwind SQL Server データベースのテーブルはすべて、Northwind SQL Anywhere データベースに移行済みです。



2. テーブル [Customers] をダブルクリックし、カラム定義を表示して、SQL Server データベースのスキーマと適合していることを確認します。
3. それぞれ対応するタブで、制約、外部キー、インデックスなどを確認できます。[Data] タブをクリックし、ローを表示します。

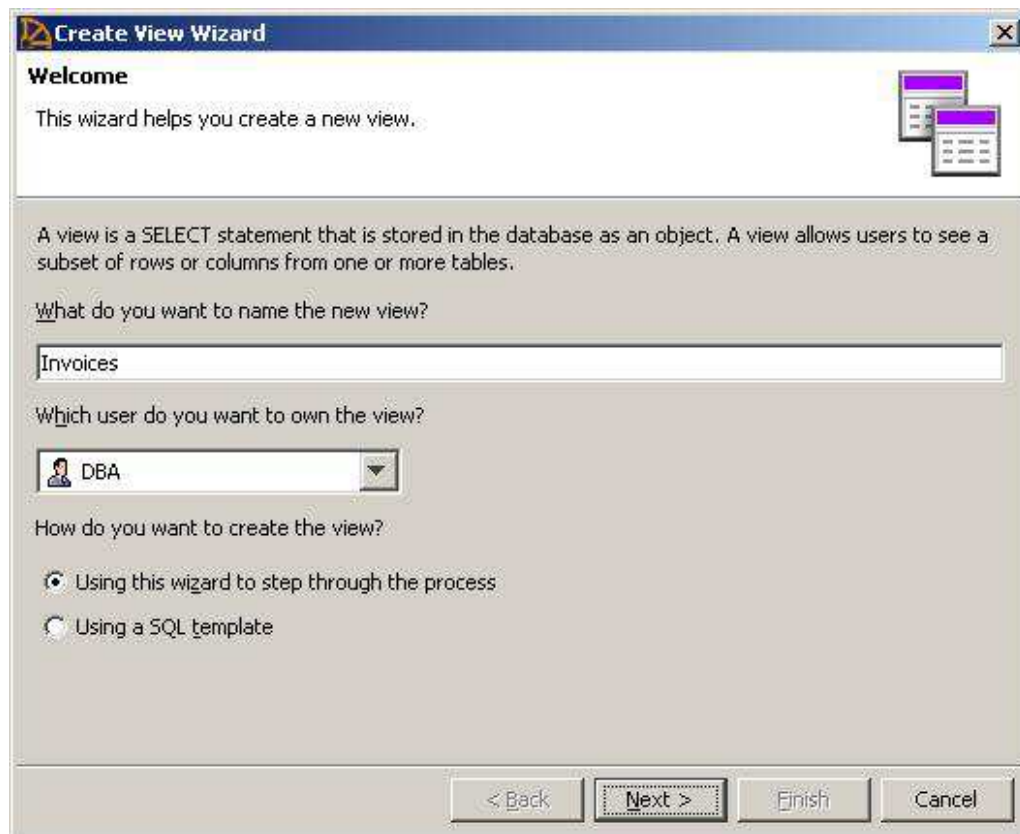


ユーザー、グループ、リモートサーバー、インデックスなど、データベースのその他のオブジェクトは、Sybase Central で確認できます。

ビューの移行

データベース移行ウィザードでは、SQL Server データベースに定義されているビューおよびマテリアライズドビューは移行されません。したがって、この移行手順は手動で実行する必要があります。幸い、SQL Server のビューの生成に使用するクエリは、SQL Anywhere の場合のクエリに若干の修正を加えるだけで流用できます。元の Northwind SQL Server データベースには、16 個のビューが定義されています。下記の SQL コードは、SQL Server の Northwind サンプルデータベースに用意されていた *instnwnd.sql* に修正を加えて流用したものです。SQL Server Management Studio を使用して、各ビューの SELECT 文を確認することもできます。

1. Sybase Central で、[View] メニューから [Folders] を選択して、[Folders] ビューに切り替えます。
2. 左ペインで、[Views] を右クリックし、ポップアップメニューから [New] -> [View] を選択します。
3. ビュー名として *Invoices* を入力して、[Next] をクリックします。



The image shows a 'Create View Wizard' dialog box with a blue title bar. The 'Welcome' section contains a description of a view and a small icon of two overlapping tables. The 'What do you want to name the new view?' section has a text box with 'Invoices' entered. The 'Which user do you want to own the view?' section has a dropdown menu with 'DBA' selected. The 'How do you want to create the view?' section has two radio buttons, with 'Using this wizard to step through the process' selected. At the bottom are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Create View Wizard

Welcome

This wizard helps you create a new view.

A view is a SELECT statement that is stored in the database as an object. A view allows users to see a subset of rows or columns from one or more tables.

What do you want to name the new view?

Invoices

Which user do you want to own the view?

DBA

How do you want to create the view?

☒ Using this wizard to step through the process

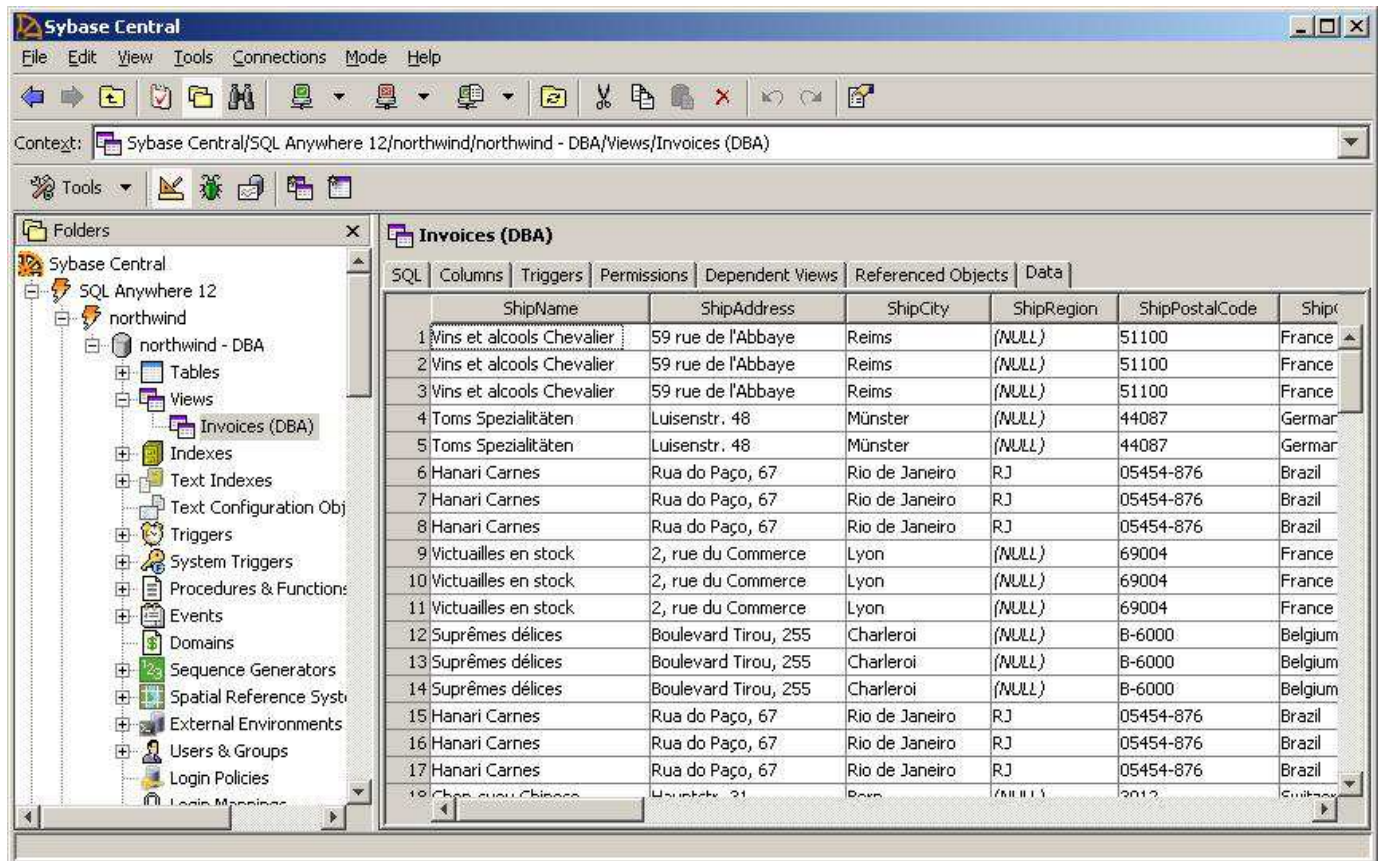
☐ Using a SQL template

< Back Next > Finish Cancel

4. 次の SQL 文を入力して、ビューを定義します。

```
SELECT Orders.ShipName, Orders.ShipAddress, Orders.ShipCity, Orders.ShipRegion, Orders.ShipPostalCode,
Orders.ShipCountry,
       Orders.CustomerID, Customers.CompanyName AS CustomerName, Customers.Address, Customers.City,
Customers.Region,
       Customers.PostalCode, Customers.Country, Employees.FirstName + ' ' + Employees.LastName AS Salesperson,
Orders.OrderID,
       Orders.OrderDate, Orders.RequiredDate, Orders.ShippedDate, Shippers.CompanyName AS ShipperName,
       "Order Details".ProductID, Products.ProductName, "Order Details".UnitPrice, "Order Details".Quantity, "Order
Details".Discount,
       CONVERT(money, ("Order Details".UnitPrice * "Order Details".Quantity) * (1 - "Order Details".Discount) / 100) * 100
AS ExtendedPrice,
       Orders.Freight
FROM Shippers INNER JOIN (
       Products INNER JOIN (
       (Employees INNER JOIN
       (Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
       ON Employees.EmployeeID = Orders.EmployeeID)
       INNER JOIN "Order Details" ON Orders.OrderID = "Order Details".OrderID)
       ON Products.ProductID = "Order Details".ProductID)
       ON Shippers.ShipperID = Orders.ShipVia
```

5. SQL Server と SQL Anywhere の SQL ダイアレクトには相違があることに注意してください。ただし、コード自体はほとんど同じです。
6. **[Finish]** をクリックし、ウィザードを終了します。SELECT 文に問題がある場合は、Sybase Central からエラーが生成されます。
7. 右ペインの **[Data]** タブをクリックし、ローを表示します。



上記の手順を繰り返して、残りの 15 個のビューを作成します。

Alphabetical list of products (製品名のアルファベット順リスト) ビューを作成する SQL 文 :

```
SELECT Products.ProductID,
       Products.ProductName,
       Products.SupplierID,
       Products.CategoryID,
       Products.QuantityPerUnit,
       Products.UnitPrice,
       Products.UnitsInStock,
       Products.UnitsOnOrder,
       Products.ReorderLevel,
       Products.Discontinued,
       Categories.CategoryName
FROM Categories INNER JOIN
     Products ON Categories.CategoryID = Products.CategoryID
WHERE (Products.Discontinued = 0)
```

Product Sales for 1997 (1997 年の製品別売上) ビューを作成する SQL 文 :

```
SELECT Categories.CategoryName,  
       Products.ProductName,  
       SUM(CONVERT(money, ("Order Details".UnitPrice * "Order Details".Quantity  
                        * (1 - "Order Details".Discount) / 100) * 100) AS ProductSales  
FROM (Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID)  
     INNER JOIN  
     (Orders INNER JOIN "Order Details" ON Orders.OrderID = "Order Details".OrderID)  
     ON Products.ProductID = "Order Details".ProductID  
WHERE (Orders.ShippedDate BETWEEN '19970101' AND '19971231')  
GROUP BY Categories.CategoryName, Products.ProductName
```

Category Sales for 1997 (1997 年のカテゴリ別売上) ビューを作成する SQL 文 :

```
SELECT CategoryName, SUM(ProductSales) AS CategorySales  
FROM "Product Sales for 1997"  
GROUP BY CategoryName
```

Current Product List (現在の製品リスト) ビューを作成する SQL 文 :

```
SELECT ProductID, ProductName  
FROM Products AS Product_List  
WHERE (Discontinued = 0)
```

Customer and Suppliers by City (地域別顧客およびサプライヤー) ビューを作成する SQL 文 :

```
SELECT City, CompanyName, ContactName, 'Customers' AS Relationship  
FROM Customers  
UNION  
SELECT City, CompanyName, ContactName, 'Suppliers'  
FROM Suppliers
```

Order Details Extended (注文明細) ビューを作成する SQL 文 :


```
SELECT "Order Details".OrderID,  
       "Order Details".ProductID,  
       Products.ProductName,  
       "Order Details".UnitPrice,  
       "Order Details".Quantity,  
       "Order Details".Discount,  
       CONVERT(money, ("Order Details".UnitPrice * "Order Details".Quantity) * (1 - "Order Details".Discount) / 100)  
         * 100 AS ExtendedPrice  
FROM Products INNER JOIN "Order Details" ON Products.ProductID = "Order Details".ProductID
```

Orders Qry (注文クエリ) ビューを作成する SQL 文 :

```
SELECT Orders.OrderID, Orders.CustomerID, Orders.EmployeeID, Orders.OrderDate, Orders.RequiredDate,  
Orders.ShippedDate,  
       Orders.ShipVia, Orders.Freight, Orders.ShipName, Orders.ShipAddress, Orders.ShipCity, Orders.ShipRegion,  
       Orders.ShipPostalCode, Orders.ShipCountry, Customers.CompanyName, Customers.Address, Customers.City,  
       Customers.Region, Customers.PostalCode, Customers.Country  
FROM Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID
```

Order Subtotals (注文小計) ビューを作成する SQL 文 :

```
SELECT OrderID, SUM(CONVERT(money, (UnitPrice * Quantity) * (1 - Discount) / 100) * 100) AS Subtotal  
FROM "Order Details"  
GROUP BY OrderID
```

Products Above Average Price (平均価格超過の製品) ビューを作成する SQL 文 :

```
SELECT ProductName, UnitPrice  
FROM Products  
WHERE (UnitPrice >  
       (SELECT AVG(UnitPrice) AS Expr1  
        FROM Products))
```

Products by Category (カテゴリ別製品) ビューを作成する SQL 文 :

```
SELECT Categories.CategoryName, Products.ProductName, Products.QuantityPerUnit, Products.UnitsInStock,  
       Products.Discontinued  
FROM Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID  
WHERE (Products.Discontinued <> 1)
```

Quarterly Orders (四半期単位の注文) ビューを作成する SQL 文 :

```
SELECT DISTINCT Customers.CustomerID, Customers.CompanyName, Customers.City, Customers.Country  
FROM Customers RIGHT OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
WHERE (Orders.OrderDate BETWEEN '19970101' AND '19971231')
```

Sales by Category (カテゴリ別売上) ビューを作成する SQL 文 :

```
SELECT Categories.CategoryID,  
       Categories.CategoryName,  
       Products.ProductName,  
       SUM("Order Details Extended".ExtendedPrice) AS ProductSales  
FROM Categories INNER JOIN  
       (Products INNER JOIN  
       (Orders INNER JOIN "Order Details Extended" ON Orders.OrderID = "Order Details Extended".OrderID)  
       ON Products.ProductID = "Order Details Extended".ProductID)  
       ON Categories.CategoryID = Products.CategoryID  
WHERE (Orders.OrderDate BETWEEN '19970101' AND '19971231')  
GROUP BY Categories.CategoryID, Categories.CategoryName, Products.ProductName
```

Sales Totals by Amount (数量別売上合計) ビューを作成する SQL 文 :

```
SELECT "Order Subtotals".Subtotal AS SaleAmount,  
       Orders.OrderID, Customers.CompanyName, Orders.ShippedDate  
FROM Customers INNER JOIN  
       (Orders INNER JOIN "Order Subtotals"  
       ON Orders.OrderID = "Order Subtotals".OrderID)  
       ON Customers.CustomerID = Orders.CustomerID  
WHERE ("Order Subtotals".Subtotal > 2500)  
       AND (Orders.ShippedDate BETWEEN '19970101' AND '19971231')
```

Summary of Sales by Quarter (四半期別売上集計) ビューを作成する SQL 文 :

```
SELECT Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal
FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID
WHERE (Orders.ShippedDate IS NOT NULL)
```

Summary of Sales by Year (年別売上集計) ビューを作成する SQL 文 :

```
SELECT Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal
FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID
WHERE (Orders.ShippedDate IS NOT NULL)
```

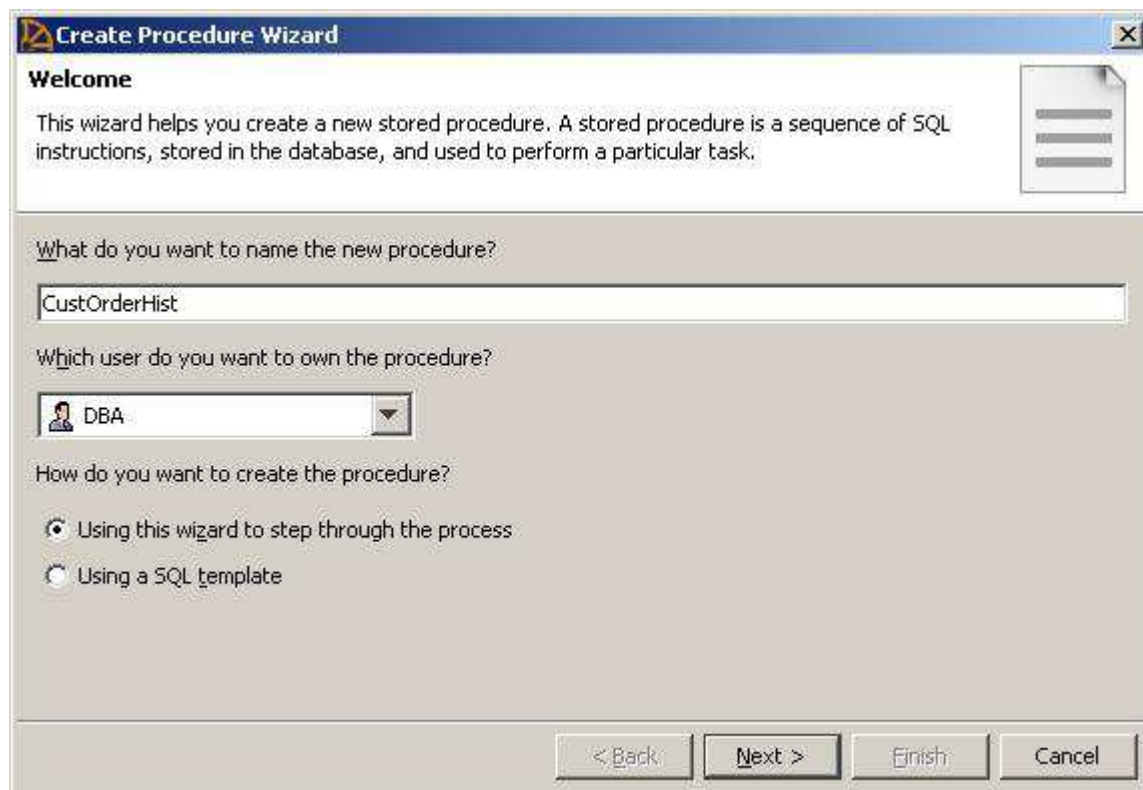
データベースロジックの移行

データベースごとに SQL ダイアレクトには相違があるので、ストアドプロシージャ、ユーザー定義関数、およびトリガーの完全な移行機能を備えたツールはありません。ただし、SQL Server と SQL Anywhere の SQL ダイアレクトには相違はありますが、共通する部分も数多くあります。インスタンスによっては、同じ SQL 文が両製品で適切に動作します。ただし通常、ビジネスロジックを想定どおりに実行するには、SQL 文に若干の修正を加える必要があります。下記の SQL コードは、SQL Server の Northwind サンプルデータベースに用意されていたファイル *instnwnd.sql* に修正を加えて流用したものです。

ストアドプロシージャ

元の Northwind SQL Server データベースには、7 個のストアドプロシージャが定義されています。このストアドプロシージャを今回の Northwind SQL Anywhere データベースに移行するには、若干の修正を加える必要があります。

1. Sybase Central の左ペインで、[Procedures and Function] を右クリックし、ポップアップメニューから [New] -> [Procedure] を選択します。
2. [Create Procedure Wizard] が表示されます。プロシージャ名として *CustOrderHist* を入力します。[Next] をクリックします。



The image shows a 'Create Procedure Wizard' dialog box. It has a title bar with a blue gradient and a close button. The main area is divided into sections. The first section is titled 'Welcome' and contains a paragraph explaining that the wizard helps create a new stored procedure. The second section asks 'What do you want to name the new procedure?' and has a text input field containing 'CustOrderHist'. The third section asks 'Which user do you want to own the procedure?' and has a dropdown menu with 'DBA' selected. The fourth section asks 'How do you want to create the procedure?' and has two radio buttons: 'Using this wizard to step through the process' (which is selected) and 'Using a SQL template'. At the bottom, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Create Procedure Wizard

Welcome

This wizard helps you create a new stored procedure. A stored procedure is a sequence of SQL instructions, stored in the database, and used to perform a particular task.

What do you want to name the new procedure?

CustOrderHist

Which user do you want to own the procedure?

DBA

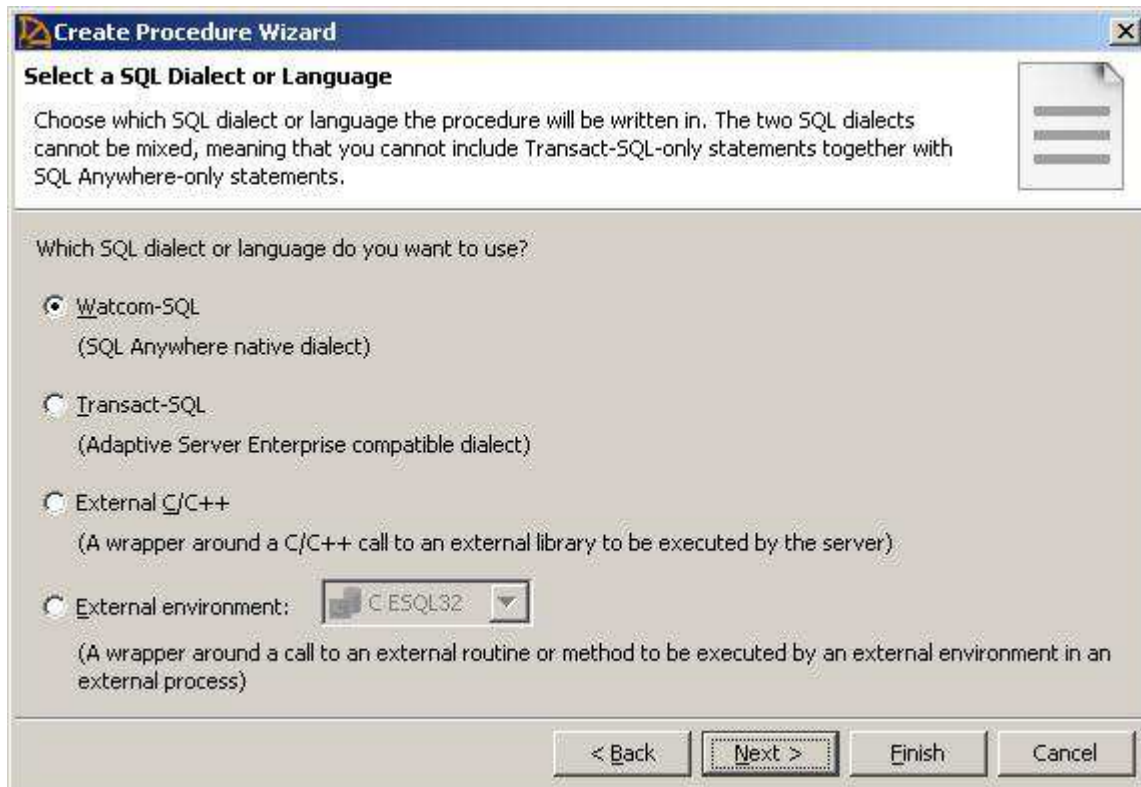
How do you want to create the procedure?

☒ Using this wizard to step through the process

☐ Using a SQL template

< Back Next > Finish Cancel

3. 別の SQL ダイアレクトや SQL 言語を使用して、プロシージャのコードを記述することもできます。**[Watcom-SQL]** を選択して、**[Next]** をクリックします。

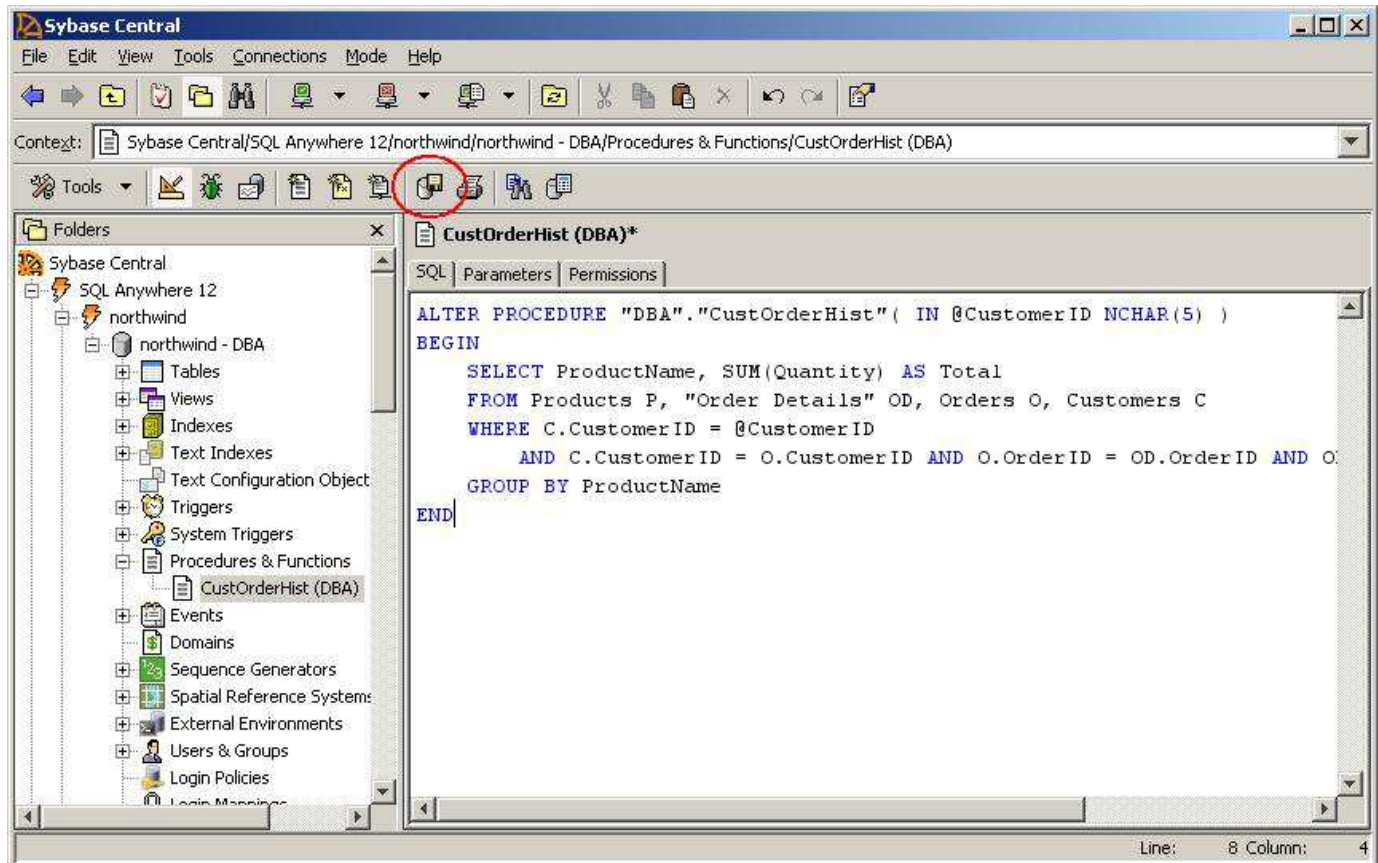


4. 今回の場合は、プロシージャーにコメントを入力する必要はありません。**[Finish]** をクリックし、ウィザードを終了します。

コードエディターが再表示されるので、プロシージャーの SQL コードを記述します。次のように *CustOrderHist* プロシージャーのコードを入力します。

```
ALTER PROCEDURE "DBA"."CustOrderHist"( IN @CustomerID NCHAR(5) )
BEGIN
    SELECT ProductName, SUM(Quantity) AS Total
    FROM Products P, "Order Details" OD, Orders O, Customers C
    WHERE C.CustomerID = @CustomerID
           AND C.CustomerID = O.CustomerID AND O.OrderID = OD.OrderID AND OD.ProductID = P.ProductID
    GROUP BY ProductName
END
```

5. ツールバーの **[Save]** ボタンをクリックするか、[File] メニューを使用して、プロシージャーを保存します。コード内にエラーがある場合は、Sybase Central から通知されます。



上記の手順を繰り返して、残りの 6 個のプロシーチャーを作成します。

CustOrdersDetail (顧客注文詳細) プロシーチャーを作成する SQL コード :

```
ALTER PROCEDURE "DBA"."CustOrdersDetail"( IN @OrderID INT )
BEGIN
    SELECT ProductName,
           ROUND(Od.UnitPrice, 2) AS UnitPrice,
           Quantity,
           CONVERT(INT, Discount * 100) AS Discount,
           ROUND(CONVERT(money, Quantity * (1 - Discount) * Od.UnitPrice), 2) AS ExtendedPrice
    FROM Products P, "Order Details" Od
    WHERE Od.ProductID = P.ProductID AND Od.OrderID = @OrderID
END
```

CustOrdersOrders (顧客別注文) プロシーチャーを作成する SQL コード :

```
ALTER PROCEDURE "DBA"."CustOrdersOrders"( IN @CustomerID NCHAR(5) )
BEGIN
    SELECT OrderID, OrderDate, RequiredDate, ShippedDate
    FROM Orders
    WHERE CustomerID = @CustomerID
    ORDER BY OrderID
END
```

Employee Sales by Country (国別従業員売上) プロシージャを作成する SQL コード :

```
ALTER PROCEDURE "DBA"."Employee Sales by Country"( IN @Beginning_Date DATETIME, IN @Ending_Date
DATETIME )
BEGIN
    SELECT Employees.Country,
           Employees.LastName,
           Employees.FirstName,
           Orders.ShippedDate,
           Orders.OrderID,
           "Order Subtotals".Subtotal AS SaleAmount
    FROM Employees INNER JOIN
           (Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID)
           ON Employees.EmployeeID = Orders.EmployeeID
    WHERE Orders.ShippedDate BETWEEN @Beginning_Date AND @Ending_Date
END
```

SalesByCategory (カテゴリ別売上) プロシージャを作成する SQL コード :


```
ALTER PROCEDURE "DBA"."SalesByCategory"( IN @CategoryName NVARCHAR(15), IN @OrdYear NVARCHAR(4)
DEFAULT '1998' )
BEGIN
    IF @OrdYear != '1996' AND @OrdYear != '1997' AND @OrdYear != '1998' THEN
        SET @OrdYear = '1998'
    ENDIF;
    SELECT ProductName,
           ROUND(SUM(CONVERT(DECIMAL(14,2), OD.Quantity * (1-OD.Discount) * OD.UnitPrice)), 0) AS
TotalPurchase
    FROM "Order Details" OD, Orders O, Products P, Categories C
    WHERE OD.OrderID = O.OrderID
           AND OD.ProductID = P.ProductID
           AND P.CategoryID = C.CategoryID
           AND C.CategoryName = @CategoryName
           AND SUBSTRING(CONVERT(NVARCHAR(22), O.OrderDate, 111), 1, 4) = @OrdYear
    GROUP BY ProductName
    ORDER BY ProductName
END
```

Sales by Year (年別売上) プロシージャを作成する SQL コード :

```
ALTER PROCEDURE "DBA"."Sales by Year"( IN @Beginning_Date DATETIME, IN @Ending_Date DATETIME )
BEGIN
    SELECT Orders.ShippedDate,
           Orders.OrderID,
           "Order Subtotals".Subtotal,
           DATENAME(yy,ShippedDate) AS Year
    FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID
    WHERE Orders.ShippedDate BETWEEN @Beginning_Date AND @Ending_Date
END
```

Ten Most Expensive Products (販売価格上位 10 製品) プロシージャを作成する SQL コード :

```
ALTER PROCEDURE "DBA"."Ten Most Expensive Products"()
BEGIN
    SELECT TOP 10 Products.ProductName AS TenMostExpensiveProducts, Products.UnitPrice
    FROM Products
    ORDER BY Products.UnitPrice DESC
END
```

ユーザー定義関数とトリガー

Northwind サンプルデータベースには、トリガーのユーザー定義関数は用意されていません。ただし、これに該当するオブジェクトを SQL Server から SQL Anywhere に変換するプロセスは、ストアドプロシージャを変換する場合と同じです。SQL Anywhere で関数およびトリガーを定義して呼び出す方法は SQL Server の場合と若干異なるので、その方法のセマンティックをよく理解しておく必要があります。

SQL Server から SQL Anywhere に関数とトリガーを変換する場合の例については、サンプルデータベース [AdventureWorks for SQL Anywhere](#) を参照してください。

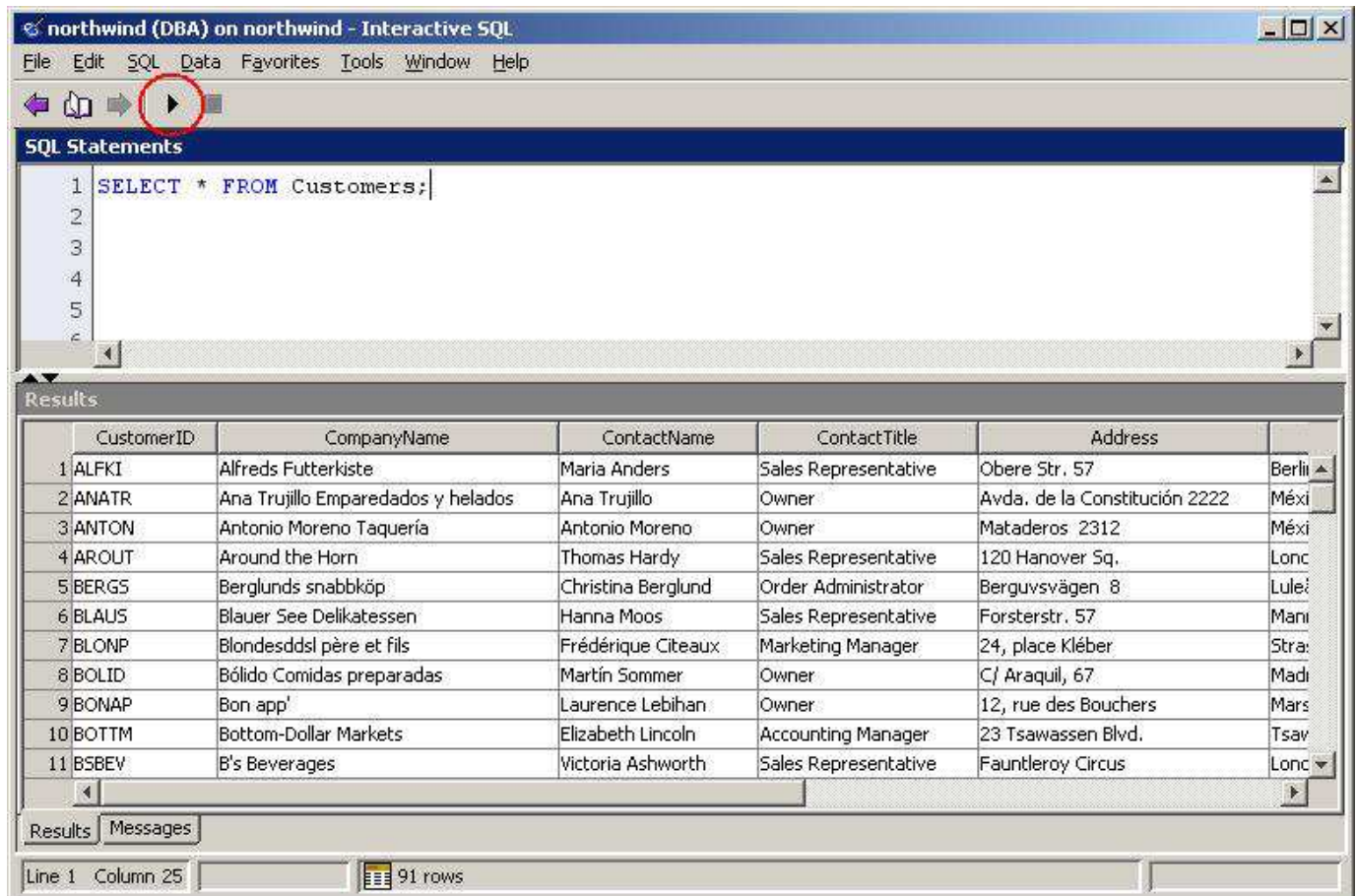
適切な動作の確認と検証

スキーマ、データ、およびロジックの移行が終了したら、新しい (SQL Anywhere) データベースが古い (SQL Server) データベースと同じように動作することを確認することが重要になります。場合によっては、アプリケーションの標準的なテスト手順を実行して、想定外の動作が発生しないことを確認する必要があります。最低でも、いくつかの SQL クエリを実行して、結果セットが適切であることを確認してください。

1. Sybase Central の左ペインで、[northwind - DBA] を右クリックし、ポップアップメニューから [Open Interactive SQL] を選択します。この操作によって、Interactive SQL (アドホッククエリを実行するグラフィカルツール) が起動して、Northwind SQL Anywhere データベースに自動的に接続します。
2. 次の SQL クエリを入力して、Customers テーブルの内容を取得します。

```
SELECT * FROM Customers;
```

3. [F5] を押すか、ツールバーの実行ボタンをクリックし、SQL クエリを実行します。下のパネルにクエリ結果が表示されます。



4. 結果セットが想定どおりであることを確認します。他のクエリをいくつか実行して、ビューとストアドプロシージャをテストします。次にいくつかの例を示します。

```
SELECT * FROM Invoices;
SELECT * FROM "Category Sales for 1997";
CALL CustOrdersOrders('LAUGB');
CALL "Employee Sales by Country"('1998-01-01', NOW());
CALL SalesByCategory('Beverages');
SELECT * FROM "Ten Most Expensive Products";
```

この時点で、前の手順で SQL Anywhere データベース内に作成した、SQL Server にマッピングしているリモートサーバー (NorthwindMSSQLRemoteServer) は不要になるので、Sybase Central から削除できます。

まとめ

SQL Anywhere には データベース移行ウィザード と一連のシステムストアドプロシージャが用意されており、SQL Server などのデータベースとの間でデータの迅速な移行を可能にします。テーブルオブジェクトの移行は、ほとんどの場合、ウィザードに従って手順を実行するだけで済み

ます。カスタムデータ型を使用している場合や、さらに柔軟な対応が必要な場合は、移行用のストアドプロシージャを使用すると、実行する必要がある操作に応じてより詳細な対応が可能になります。

SQL Anywhere と他のデータベースの SQL ダイレクトには相違があるので、ビューとデータベースロジックを移行する場合には、手順を手動で実行する方法が適しています。幸い、SQL コードには共通する部分が多く、移行プロセスは容易になっています。

スキーマとデータの移行が終了したら、新しい SQL Anywhere データベースを使用して、アプリケーションが適切に動作することを確認します。テスト手順を実行して、想定外の動作が発生しないことを確認します。