Adventure Works Cycles Business Scenarios

SQL Server 2008

Adventure Works Cycles, the fictitious company on which the AdventureWorks sample databases are based, is a large, multinational manufacturing company. The company manufactures and sells metal and composite bicycles to North American, European and Asian commercial markets. While its base operation is located in Bothell, Washington with 290 employees, several regional sales teams are located throughout their market base.

In 2000, Adventure Works Cycles bought a small manufacturing plant, Importadores Neptuno, located in Mexico. Importadores Neptuno manufactures several critical subcomponents for the Adventure Works Cycles product line. These subcomponents are shipped to the Bothell location for final product assembly. In 2001, Importadores Neptuno, became the sole manufacturer and distributor of the touring bicycle product group.

Coming off a successful fiscal year, Adventure Works Cycles is looking to broaden its market share by targeting their sales to their best customers, extending their product availability through an external Web site, and reducing their cost of sales through lower production costs.

AdventureWorks to Northwind Table Comparison

SQL Server 2008

Some tables in the AdventureWorks sample database are similar in structure and content to tables in the **Northwind** sample database. You can use the following table to convert queries that use **Northwind** to queries that use AdventureWorks by selecting columns from the appropriate AdventureWorks tables. For example, if a query references the **Orders** table in **Northwind**, a similar query can be written for AdventureWorks by using the **Sales.SalesOrderHeader** table. Notice that AdventureWorks uses schema names other than **dbo**. The schema names are included with the table names and must be specified when performing queries against the tables. For more information, see Schemas in AdventureWorks.

Northwind	AdventureWorks	Comments
Categories	**Production.ProductCategory**	See **ProductSubCategory** and **ProductModel** for a more detailed categorization.
Customers	**Sales.Customer** Join with **Sales.Individual** and **Sales.Store**	AdventureWorks contains two types of customers: stores (resellers) and individuals (consumers).
Customer Demographics	**Sales.Individual** **Sales.Store**	See the **Demographics** column (xml) in the **Sales.Individual** and **Sales.Store** tables.
Employees	**HumanResources.Employee** **Join with Person.Contact**	
Employee Territories	**Sales.SalesPerson**	**Sales.SalesPerson** maps sales representatives to sales territories. See also the **Sales.SalesTerritory** and **Employees** tables.
Orders	**Sales.SalesOrderHeader**	
Order Details	**Sales.SalesOrderDetail**	
Products	**Production.Product**	
Region	**Sales.SalesTerritory**	The **RegionDescription** column is the equivalent of the **Group** column in **Sales.SalesTerritory**.
Shippers	**Purchasing.ShipMethod**	
Suppliers	**Purchasing.Vendor**	
Territories	**Sales.SalesTerritory**	The **TerritoryDescription** column is the equivalent of the **Name** column in **Sales.SalesTerritory**.

AdventureWorks to pubs Table Comparison

SQL Server 2008

Some tables in the AdventureWorks sample database are similar in structure and content to tables in the **pubs** sample database. You can use the following table to convert queries that use **pubs** to queries that use AdventureWorks by selecting columns from the appropriate AdventureWorks tables. For example, if a query references the **discounts** table in **pubs**, a similar query can be written for AdventureWorks by using the **Sales.SpecialOffer** table. Notice that AdventureWorks uses schema names other than **dbo**. The schema names are included with the table names and must be specified when performing queries against the tables. For more information, see Schemas in AdventureWorks.

pubs	AdventureWorks	Comments
authors	**Purchasing.Vendor**	
discounts	**Sales.SpecialOffer**	
employee	**HumanResources.Employee**	
jobs	**HumanResources.Employee**	See the **Title** column in **Employee**.
pub_info	**Production.ProductPhoto** **Production.ProductDescription**	
publishers	**Sales.Store** **Person.Address** **Sales.CustomerAddress** **Person.CountryRegion** **Person.StateProvince**	The following query provides the equivalent information in the **publishers** table. USE AdventureWorks; GO SELECT S.CustomerID, S.Name AS Store, A.City, SP.Name AS State, CR.Name AS CountryRegion FROM Sales.Store AS S JOIN Sales.CustomerAddress CA ON CA.CustomerID = S.CustomerID JOIN Person.Address AS A ON A.AddressID = CA.AddressID JOIN Person.StateProvince AS SP ON SP.StateProvinceID = A.StateProvinceID

		JOIN Person.CountryRegion AS CR ON CR.CountryRegionCode = SP.CountryRegionCode GROUP BY S.CustomerID, S.Name, A.City, SP.Name, CR.Name ORDER BY S.CustomerID;
roysched	**Sales.SpecialOffer**	See the **MinQty** and **MaxQty** columns.
sales	**Sales.SalesOrderHeader** **Sales.SalesOrderDetail**	
stores	**Sales.Store**	
titleauthor	**Production.ProductVendor**	**titleauthor** is an associative table that maps authors to titles. **Production.ProductVendor** maps vendors to the products they sell to Adventure Works Cycles.
titles	**Production.Product**	

Schemas in AdventureWorks

SQL Server 2008

In the AdventureWorks sample OLTP database, objects such as tables, views, and procedures are contained in schemas. Schemas change the way in which these objects can be accessed. This topic provides a brief overview of schemas, describes how schemas are used in the AdventureWorks database, and provides methods and alternatives for accessing objects that are contained in schemas.

Schemas in AdventureWorks

In SQL Server 2005 and later, schemas are separate from users: As database principals, users own schemas, and objects are contained in the schemas. For more information, see User-Schema Separation.

The following table describes the schemas that are used in AdventureWorks and lists representative tables in each schema.

Schema	Contains objects related to	Examples
HumanResources	Employees of Adventure Works Cycles.	Employee Table Department Table
Person	Names and addresses of individual customers, vendors, and employees.	Contact Table Address Table StateProvince Table
Production	Products manufactured and sold by Adventure Works Cycles.	BillOfMaterials Table Product Table WorkOrder Table
Purchasing	Vendors from who parts and products are purchased.	PurchaseOrderDetail Table PurchaseOrderHeader Table Vendor Table
Sales	Customers and sales-related data.	Customer Table SalesOrderDetail Table SalesOrderHeader Table

Sales and Marketing Scenario

SQL Server 2008

Customer and sales-related information is a significant part of the AdventureWorks sample database. This topic provides details about the customers that are represented in the sample database, a schema of the major customer and sales tables and sample queries that demonstrate table relationships.

Customers Types

As a bicycle manufacturing company, Adventure Works Cycles has two types of customers:

- Individuals. These are consumers who buy products from the Adventure Works Cycles online store.
- Stores. These are retail or wholesale stores that buy products for resale from Adventure Works Cycles sales representatives.

The **Customer** table contains one record for each customer. The column **CustomerType** indicates whether the customer is an individual consumer (**CustomerType= 'I'**) or a store (**CustomerType= 'S'**). Data specific to these customer types is maintained in the **Individual** and **Store** tables, respectively.

Customer type	Major tables	Number of customers	Additional information
Individual	Person.Contact Sales.Customer Sales.Individual Sales.SalesOrderHeader Sales.SalesOrderDetail	18,484	Sales and demographic data have been trended for data mining scenarios. Demographic data (income, hobbies, number of cars, and so on) is stored as xml data in the **Demographics** column of the **Individual** table.
Store	Person.Contact Sales.Customer Sales.Store Sales.StoreContact Sales.SalesOrderHeader Sales.SalesOrderDetail	701	Data has been trended for Analysis Services scenarios. Stores are categorized by size: large, medium, and small. Demographic data stored as xml data. Store contacts are employees of the store who interact with Adventure Works Cycles sales representatives. For example, the store owner or purchasing manager would be typical contacts for Adventure Works Cycles salespeople.

You can use the following queries to view customer data and to become familiar with the customer-table relationships.

A. Viewing individual customers (consumers)

The following example returns the first and last name of each customer who is categorized as an individual consumer (CustomerType = 'I').

SQL

```
USE AdventureWorks;
GO
SELECT FirstName, LastName
FROM Person.Contact AS C
    JOIN Sales.Individual AS I
        ON C.ContactID = I.ContactID
    JOIN Sales.Customer AS Cu
        ON I.CustomerID = Cu.CustomerID
WHERE Cu.CustomerType = 'I'
ORDER BY LastName, FirstName ;
GO
```

B. Viewing individual customer address data

The following example lists the names and addresses of all individual customers.

SQL

```
USE AdventureWorks;
GO
SELECT I.CustomerID, C.FirstName, C.LastName, A.AddressLine1, A.City,
    SP.Name AS State, CR.Name AS CountryRegion
FROM Person.Contact AS C
    JOIN Sales.Individual AS I ON C.ContactID = I.ContactID
    JOIN Sales.CustomerAddress AS CA ON CA.CustomerID = I.CustomerID
    JOIN Person.Address AS A ON A.AddressID = CA.AddressID
    JOIN Person.StateProvince SP ON
        SP.StateProvinceID = A.StateProvinceID
    JOIN Person.CountryRegion CR ON
        CR.CountryRegionCode = SP.CountryRegionCode
ORDER BY I.CustomerID ;
GO
```

C. Viewing store customers, either retail or wholesale stores

The following example returns the name of each customer that is categorized as a store (CustomerType = 'S').

SQL

```
USE AdventureWorks;
GO
SELECT Name
FROM Sales.Store AS S
    JOIN Sales.Customer AS C
        ON S.CustomerID = C.CustomerID
WHERE C.CustomerType = N'S'
ORDER BY Name ;
GO
GO
```

D. Viewing store contacts by store

The following example returns the name of all store customers and the names and titles of store employees who authorized to purchase Adventure Works Cycles products on behalf of their company.

SQL

```
USE AdventureWorks;
GO
SELECT S.Name AS Store, C.FirstName, C.LastName, CT.Name AS Title
FROM Person.Contact AS C
    JOIN Sales.StoreContact AS SC ON C.ContactID = SC.ContactID
    JOIN Person.ContactType AS CT ON
        CT.ContactTypeID = SC.ContactTypeID
    JOIN Sales.Store AS S ON S.CustomerID = SC.CustomerID
ORDER BY S.Name ;
GO
```

E. Viewing sales by store

The following example lists store customers and their associated sales orders.

SQL

```
USE AdventureWorks;
GO
SELECT Name, SalesOrderNumber, OrderDate, TotalDue
FROM Sales.Store AS S
    JOIN Sales.SalesOrderHeader AS SO ON S.CustomerID = SO.CustomerID
ORDER BY Name, OrderDate ;
GO
```

F. Viewing stores by locations

The following example prints the store-customer name, city, state and country/region.

SQL

```
USE AdventureWorks;
GO
SELECT S.CustomerID, S.Name AS Store, A.City, SP.Name AS State, CR.Name
    AS CountryRegion
FROM Sales.Store AS S
    JOIN Sales.CustomerAddress AS CA ON CA.CustomerID = S.CustomerID
    JOIN Person.Address AS A ON A.AddressID = CA.AddressID
    JOIN Person.StateProvince SP ON
        SP.StateProvinceID = A.StateProvinceID
    JOIN Person.CountryRegion CR ON
        CR.CountryRegionCode = SP.CountryRegionCode
ORDER BY S.CustomerID ;
GO
GO
```

Purchasing and Vendor Scenario

SQL Server 2008

At Adventure Works Cycles, the purchasing department buys raw materials and parts used in the manufacture of Adventure Works Cycles bicycles. Adventure Works Cycles also purchases products for resale, such as bicycle apparel and bicycle add-ons like water bottles and pumps. The information about these products and the vendors from whom they are obtained is stored in the AdventureWorks sample database.

This topic provides details about the vendors represented in the sample database, a schema diagram of the major vendor-related tables and sample queries that demonstrate common table relationships.

Vendor and Purchasing Tables

The following table contains a brief description of the data stored in these tables.

Schema.Table	Contains this kind of content	Comments
Person.Address	Street address information for all customers. Customers may have more than one address. For example, a customer may have a billing address and a different address for shipping.	The associative table **VendorAddress** maps vendors to their addresses. The **Address** table also contains address information for Adventure Works Cycles employees and customers.
Person.Contact	Names of vendor employees with whom Adventure Works Cycles purchasing agents order products. A vendor may have more than one contact. For example, a sales agent and a sales manager. The Adventure Works Cycles purchasing agent may have the sales agent as a primary customer contact and	The associative table **VendorContact** maps contacts to vendors. The column **AdditionalContactInfo** contains data such as additional telephone numbers (cell telephone, fax, and so on) specific to the contact. The column is an xml data type. For more information, see About the Contact.AdditionalContactInfo xml Column.

	the sales manager as a secondary contact.	
Production.ProductVendor	Maps vendors to the products they supply. A product may be supplied by more than one vendor, and a vendor may supply more than one product.	
Purchasing.PurchaseOrderDetail	Details of the purchase order, such as products ordered, quantity, and unit price.	
Purchasing.PurchaseOrderHeader	Purchase order summary information, such as total due, order date, and order status.	The **PurchaseOrderHeader** and **PurchaseOrderDetail** tables together create a master-detail relationship.
Purchasing.ShipMethod	A lookup table that is used to maintain standard ways of shipping products.	The **ShipMethodID** column is included in the **PurchaseOrderHeader** table.
Purchasing.Vendor	Details about vendors, such as the vendor name and account number.	
Purchasing.VendorAddress	Links customers to address information in the **Address** table.	Addresses are categorized by type, such as billing, home, shipping, and so on). The **AddressTypeID** column maps to the **AddressType** table.
Purchasing.VendorContact	Street address information for all customers. Customers may have more than one address. For example, a customer may have a billing address and a different address for shipping.	This is an associative table. See the **Contact** and **Vendor** tables.

You can use the following queries to view purchasing and vendor data and to become familiar with the purchasing and vendor table relationships.

A. Viewing vendors by location

The following example lists the vendors and their address.

SQL
```
USE AdventureWorks;
GO
SELECT V.VendorID, V.Name AS Vendor, A.AddressLine1, A.AddressLine2, A.City,
SP.Name AS State, CR.Name AS Country
FROM Purchasing.Vendor AS V
    JOIN Purchasing.VendorAddress AS VA ON VA.VendorID = V.VendorID
    JOIN Person.Address AS A on A.AddressID = VA.AddressID
    JOIN Person.StateProvince AS SP on SP.StateProvinceID =
A.StateProvinceID
    JOIN Person.CountryRegion AS CR ON CR.CountryRegionCode =
SP.CountryRegionCode
GROUP BY V.VendorID, V.Name, A.AddressLine1, A.AddressLine2, A.City, SP.Name,
CR.Name
ORDER BY V.VendorID;
GO
```

B. Viewing products supplied by vendors

The following example lists the products that the vendors supply to Adventure Works Cycles.

SQL
```
USE AdventureWorks;
GO
SELECT P.ProductNumber, P.Name AS Product, V.Name AS Vendor, PV.LastReceiptCost
FROM Production.Product AS P
    JOIN Purchasing.ProductVendor AS PV ON P.ProductID = PV.ProductID
    JOIN Purchasing.Vendor AS V ON V.VendorID = PV.VendorID
ORDER BY P.Name ;
GO
```

C. Viewing vendor contacts by vendor

The following example lists vendor contacts. Vendor contacts are employees of the vendor with whom employees of the Adventure Works Cycles purchasing department interact to order parts and products.

SQL
```
GO
SELECT V.Name as Vendor, C.FirstName, C.LastName, CT.Name AS Title
FROM Person.Contact AS C
    JOIN Purchasing.VendorContact VC ON C.ContactID = VC.ContactID
    JOIN Person.ContactType CT ON CT.ContactTypeID = VC.ContactTypeID
    JOIN Purchasing.Vendor V ON V.VendorID = VC.VendorID
ORDER BY V.Name;
GO
```

D. Viewing purchases by vendor

The following example displays the vendors and their associated purchase orders.

SQL

```
USE AdventureWorks;
GO
SELECT V.Name AS Vendor, SUM(PH.TotalDue)AS [Total Purchase],
    AVG(PH.TotalDue)AS [Average Purchase], MIN(PH.TotalDue)
    AS [Minimum Purchase], MAX(PH.TotalDue)AS [Maximum Purchase]
FROM Purchasing.Vendor AS V
    JOIN Purchasing.PurchaseOrderHeader AS PH ON V.VendorID = PH.VendorID
GROUP BY V.Name
ORDER BY V.Name;
GO
```

Manufacturing Scenario

SQL Server 2008

This topic provides details about the Adventure Works Cycles manufacturing information that is represented in the AdventureWorks sample database, a list of manufacturing-related tables, and sample queries that demonstrate common table relationships.

Manufacturing Overview

In the AdventureWorks sample database, tables are provided that support the following typical manufacturing areas:

- Manufacturing processes:
  - Bill of materials: Lists the products that are used or contained in another product.
  - Work orders: Manufacturing orders by work center.
  - Locations: Defines the major manufacturing and inventory areas, such as frame forming, paint, subassembly, and so on.
  - Manufacturing and product assembly instructions by work center.
- Product inventory: The physical location of a product in the warehouse or manufacturing area, and the quantity available in that area.
- Engineering documentation: Technical specifications and maintenance documentation for bicycles or bicycle components.

Manufacturing Tables

The following table contains a brief description of the data that is stored in the manufacturing tables.

Schema.Table	Contains this kind of content	Comment
Production.BillOfMaterials	A list of all the components used to manufacture bicycles and bicycle subassemblies.	There is an intrinsic recursive relationship in the bill of material structure that indicates the relationship between a parent product and the components that make up that product. For example, if the parent product is a bicycle, the first-level component might be a wheel assembly. The wheel assembly has its own components, such as reflectors, rims, spokes, tires, and tire tubes. The **ProductAssemblyID** column represents the parent, or primary, product and **ComponentID**

		represents the child, or individual, parts used to build the parent assembly. The **BOM_Level** column indicates the level of the **ComponentID** relative to the **ProductAssemblyID**. In the previous example, the wheel assembly would have a **BOM_Level** of 1, the components of the wheel assembly would have a **BOM_Level** of 2, and so on.
Production.Document	Engineering specifications and other technical documentation.	The **DocumentSummary** column uses the varchar(max) data type. The **Document** column uses the varbinary(max) data type.
Production.Illustration	Bicycle manufacturing illustrations.	The illustrations are rendered in the manufacturing instructions that are contained in the **ProductModel** table. This column uses the xml data type.
Production.Location	A list of inventory and manufacturing areas within Adventure Works Cycles in which the products and parts are stored as inventory or built. For example, paint is stored in both the Paint Storage location in the warehouse and in the manufacturing work center, Paint Shop, where the bicycle frames are painted.	
Production.Product	Information about each product sold by Adventure Works Cycles or used to manufacture Adventure Works Cycles bicycles and bicycle components.	The **FinishedGoodsFlag** column indicates whether a product is sold. Products that are not sold are components of a product that is sold. For example, a bicycle would be sold, but the sheet of metal used to create the bicycle frame would not.
Production.ProductInventory	The inventory level of products by their location. See **Production.Location** previously mentioned.	
Production.ProductModel	The product models associated with products. For example, Mountain-100 or LL Touring	The **CatalogDescription** column contains additional product information by using the xml data

	Frame.	type. The **Instructions** column contains product manufacturing instructions by using the xml data type
Production.ScrapReason	A list of common reasons why bicycles or bicycles parts are rejected during the manufacturing process. For example, the scrap reason 'Paint failed' is used in the Paint work center to reject a bicycle frame for which the paint did not cure correctly.	The **WorkOrderRouting** table tracks the quantity scrapped and the reason for scrapping by product. Depending on the severity of the problem, the product must be fixed or replaced before the product can move to the next work center.
Production.WorkOrder	Defines the products and quantity that must be manufactured to meet current and forecasted sales.	
Production.WorkOrderRouting	The details for each work order. This includes the sequence of work centers the product travels through in the manufacturing or assembly process. For example, bicycle handlebars are manufactured in the Frame Forming work center. They are moved to the Frame Welding work center for additional work, and then moved to the Subassembly work center, where they are added to the bicycle frame.	

You can use the following queries to view manufacturing and product data and to become familiar with the manufacturing table relationships.

A. Viewing a multilevel bill-of-materials list for a parent product

The following example displays all the components that are used to create a specific parent product: ProductAssemblyID.

SQL

```
USE AdventureWorks;
GO
WITH Parts(AssemblyID, ComponentID, PerAssemblyQty, EndDate, ComponentLevel) AS
(
    SELECT b.ProductAssemblyID, b.ComponentID, b.PerAssemblyQty,
        b.EndDate, 0 AS ComponentLevel
    FROM Production.BillOfMaterials AS b
    WHERE b.ProductAssemblyID = 800
```

```
          AND b.EndDate IS NULL
    UNION ALL
    SELECT bom.ProductAssemblyID, bom.ComponentID, p.PerAssemblyQty,
        bom.EndDate, ComponentLevel + 1
    FROM Production.BillOfMaterials AS bom
        INNER JOIN Parts AS p
        ON bom.ProductAssemblyID = p.ComponentID
        AND bom.EndDate IS NULL
)
SELECT AssemblyID, ComponentID, Name, PerAssemblyQty, EndDate,
        ComponentLevel
FROM Parts AS p
    INNER JOIN Production.Product AS pr
    ON p.ComponentID = pr.ProductID
ORDER BY ComponentLevel, AssemblyID, ComponentID;
GO
```

B. Viewing Product Inventory

In the following example, the quantity that is available for each product is listed by its location in inventory. Products can be located in multiple locations.

SQL

```
USE AdventureWorks;
GO
SELECT P.Name AS Product, L.Name AS [Inventory Location],
    SUM(PI.Quantity)AS [Qty Available]
FROM Production.Product AS P
    JOIN Production.ProductInventory AS PI ON P.ProductID = PI.ProductID
    JOIN Production.Location AS L ON PI.LocationID = L.LocationID
GROUP BY P.Name, L.Name
ORDER BY P.Name ;
GO
```

C. Viewing work orders by product

In the following example, all work orders are listed for products in the subcategories Mountain Bike (1), Road Bike (2), and Touring Bike (3).

SQL

```
USE AdventureWorks;
GO
SELECT WorkOrderID, P.Name AS Product, OrderQty, DueDate
FROM Production.WorkOrder W
    JOIN Production.Product P ON W.ProductID = P.ProductID
WHERE P.ProductSubcategoryID IN (1, 2, 3)
ORDER BY P.Name, DueDate ;
GO
```