



The Next Generation of Networking : Learn about the business benefits of deploying Internet Protocol version 6 (IPv6)

[eWork Markets: IT Consulting](#)

[Marketing Optimization: \[x+1\]
Ad and Site Campaign
Software](#)

[Web Design and Development
To Maximize Your Online
Success](#)

[Professional Web
Development: Accr](#)

[Avocent: Computer Network
Connectivity Solution](#)

[Web Store Builders](#)

[Computer Repair and
Technical Assistance:
CompUSA Inc.](#)

[CrypKey: Security
Developers](#)

www.developer.com/db/article.php/3613301

[Back to Article](#)

Using Synonyms in SQL Server 2005

By [Mike Gunderloy](#)

June 14, 2006

If you've been developing with SQL Server for any length of time, you're undoubtedly used to referring to objects with four-part identifiers:

```
[[[server.][database.][schema_name].]object_name
```

As the square brackets show, different parts of this syntax can be omitted, as long as you supply enough to unambiguously identify what you're talking about. For example, all of these might refer to the same object:

```
Server1.AdventureWorks.Production.ProductCategory
AdventureWorks.Production.ProductCategory
AdventureWorks..ProductCategory
ProductCategory
```

For the most part, you can get by with three-part names, omitting the server name - unless you're dealing with objects on a linked server. By default, the context of all objects is the local database where your SQL statements are executing. But sometimes, to be precise, you have no choice but to use the four-part name (otherwise known as a *fully-qualified name*. Or do you? In SQL Server 2005, the story changes a bit.

Enter the Synonym

SQL Server 2005 introduces the concept of a *synonym*: a single-part name that can replace a two-, three-, or four-part name in many SQL statements. Using synonyms lets you cut down on typing (always a welcome advance for developers!) and can also provide an abstraction layer to protect you against changes in underlying objects. To understand how this works, let's start with the syntax for creating synonyms. Not surprisingly, this is the

`CREATE SYNONYM` statement:



```
CREATE SYNONYM [schema_name.]synonym_name FOR object_name
```

Here *object_name* is the name of a SQL Server object (specified fully enough to identify the object) and *synonym_name* is the new name that you want to assign to it. If you don't specify a schema for the synonym, then SQL Server uses the current user's default schema. The object doesn't need to exist when you create the synonym, because synonyms are late bound: SQL Server only checks the base object when you actually use the synonym.

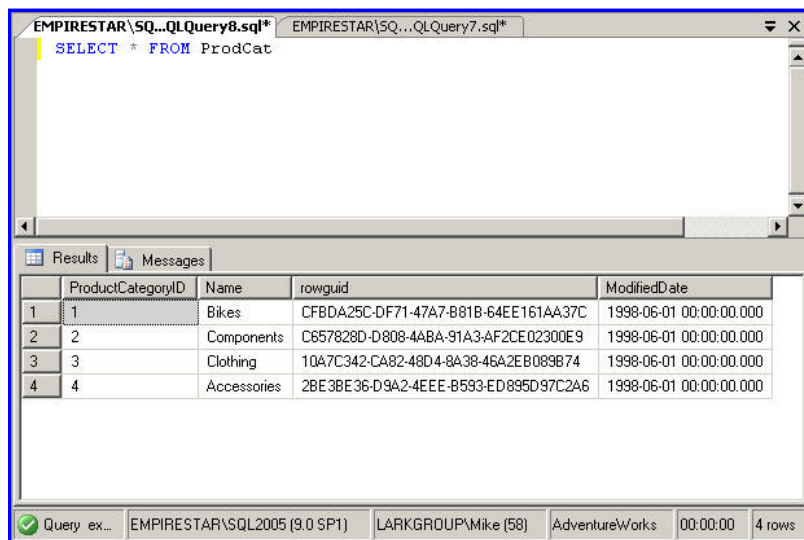
For example, in the AdventureWorks sample database you can create a synonym this way:

```
CREATE SYNONYM ProdCat  
FOR AdventureWorks.Production.ProductCategory
```

Having done this, you can now proceed to use the synonym in SQL statements. For example:

```
SELECT * FROM ProdCat
```

Figure 1 shows the results



	ProductCategoryID	Name	rowguid	ModifiedDate
1	1	Bikes	CFBDA25C-DF71-47A7-B81B-64EE161AA37C	1998-06-01 00:00:00.000
2	2	Components	C657828D-D808-4ABA-91A3-AF2CE02300E9	1998-06-01 00:00:00.000
3	3	Clothing	10A7C342-CA82-48D4-8A38-46A2EB089B74	1998-06-01 00:00:00.000
4	4	Accessories	2BE3BE36-D9A2-4EEE-B593-ED895D97C2A6	1998-06-01 00:00:00.000

[Click here for a larger image.](#)

When you're done with a synonym, you can get rid of it using the `DROP SYNONYM` statement, which has the syntax you expect:

```
DROP SYNONYM [schema_name.]synonym_name
```

A Few of the Details

Naturally, there are some limits to synonym use. For starters, you can't create a synonym for just anything. SQL Server 2005 contains more objects than ever before, but synonyms are limited to a core set of the most useful objects:

- Tables
- Views
- Stored procedures
- CLR stored procedures
- CLR functions
- Replication filter procedures
- Extended stored procedures
- SQL scalar, table-valued, and inline table-valued functions

There are also limits on where you can use synonyms. Generally speaking, you can use a synonym in these T-SQL statements:

- `SELECT`
- sub-selects
- `UPDATE`
- `INSERT`
- `DELETE`
- `EXECUTE`

If you think about it, that second list shouldn't be too surprising: it encompasses the most common statements where you'd use the objects from the first list. Note that you *cannot* reference a synonym in a DDL statement. If you want to use `ALTER TABLE` to change something about the `ProductCategory` table, for example, you need to work with the base table, not with the `ProdCat` synonym.

To create a synonym, you need `CREATE SYNONYM` permission. After a synonym has been created, it has its own associated `GRANT`, `DENY`, and `REVOKE` permissions.

Synonyms as an Abstraction Layer

Besides saving you typing, synonyms can also serve as an abstraction layer over unstable or frequently updated portions of your database. That's because the bindings between a synonym and its base object are only checked at runtime, not at design time.

To illustrate this, examine what happens if you redefine the `ProdCat` synonym:

```
DROP SYNONYM ProdCat
CREATE SYNONYM ProdCat
FOR AdventureWorks.Production.ProductCategory
SELECT * FROM ProdCat
DROP SYNONYM ProdCat
CREATE SYNONYM ProdCat
FOR AdventureWorks.Production.Culture
SELECT * FROM ProdCat
```

Note that you need to use `DROP` and `CREATE` whenever you want to change a synonym; there's no `ALTER SYNONYM` statement. Figure 2 shows the result of running this batch of statements.

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays two queries: 'EMPIRESTAR\SQ...QLQuery8.sql*' and 'EMPIRESTAR\SQ...QLQuery7.sql*'. The bottom pane shows the results of the second query, which is a 'SELECT * FROM ProdCat' statement. The results are displayed in two tables. The first table has columns 'ProductCategoryID', 'Name', 'rowguid', and 'ModifiedDate'. The second table has columns 'CultureID', 'Name', and 'ModifiedDate'.

	ProductCategoryID	Name	rowguid	ModifiedDate
1	1	Bikes	CFBDA25C-DF71-47A7-B81B-64EE161A437C	1998-06-01 00:00:00.000
2	2	Components	C657828D-D808-4ABA-91A3-AF2CE02300E9	1998-06-01 00:00:00.000
3	3	Clothing	10A7C342-CA82-48D4-8A38-46A2EB089B74	1998-06-01 00:00:00.000
4	4	Accessories	2BE3BE36-D9A2-4EEE-B593-ED895D97C2A6	1998-06-01 00:00:00.000

	CultureID	Name	ModifiedDate
1		Invariant Language (Invariant Country)	1998-06-01 00:00:00.000
2	ar	Arabic	1998-06-01 00:00:00.000
3	en	English	1998-06-01 00:00:00.000
4	es	Spanish	1998-06-01 00:00:00.000
5	fr	French	1998-06-01 00:00:00.000
6	he	Hebrew	1998-06-01 00:00:00.000
7	th	Thai	1998-06-01 00:00:00.000
8	zh-cht	Chinese	1998-06-01 00:00:00.000

At the bottom of the window, the status bar shows: Query e... EMPIRESTAR\SQL2005 (9.0 SP1) LARGROUP\Mike (55) AdventureWorks 00:00:00 12 rows

Note the very different results from the second execution of `SELECT * FROM ProdCat`, because the synonym was redefined to refer to a different table in the interim.

You can take advantage of this behavior in your own applications. For example, suppose you've got an application that needs to analyze customer data from a variety of different databases. Rather than linking and unlinking the base tables from the individual databases, or writing complex SQL that selects the tables to analyze, you could use synonyms. Define a synonym such as `DataTable` to refer to the four-part name of the data that you want to work with, and redefine it whenever you want to switch source data.



Of course, this late binding is a two-edged sword. Because SQL Server doesn't track whether a synonym points to anything, it also doesn't do any sort of schema binding. If you drop a synonym that's used in another statement, you won't find out that anything is wrong until you try to run that statement.

File Under "Nice to Have"

Could you develop your next SQL Server application without synonyms? Sure. They don't bring anything fundamentally new to the table; you could always use fully-qualified names anywhere that you could use a synonym. But sometimes it's nice to have these little amenities in the product, just because they make for less typing and an easier development experience. Perhaps synonyms will only save you ten minutes over the course of an entire application, but still, those little ten-minute savings add up. Let's hope Microsoft keeps delivering nice little improvements like this along with the big headline features.

About the Author

Mike Gunderloy is the author of over 20 books and numerous articles on development topics, and the Senior Technology Partner for [Adaptive Strategy](#), a Washington State consulting firm. When he's not writing code, Mike putters in the garden on his farm in eastern Washington state.

 <p>Compare prices and save on:</p>	Laptops <ul style="list-style-type: none"> Toshiba Satellite Powerbook G4 Sony Vaio VGN Dell Inspiron 6000 IBM ThinkPad 	PDA's <ul style="list-style-type: none"> Palm One Bluetooth HP Ipaq Dell Axim I-Mate 	Digital Cameras <ul style="list-style-type: none"> Olympus 740 Sony Cyber Shot Nikon Coolpix Canon S80 Panasonic DMC 	Desktop Computers <ul style="list-style-type: none"> Graphics Cards Sound Cards Memory Cards Motherboards Gaming Software 	<p>Comp</p> 

JupiterWeb networks:



Search JupiterWeb:

[Jupitermedia Corporation](#) has two divisions: [Jupiterimages](#) and [JupiterWeb](#)

[Jupitermedia Corporate Info](#)

Copyright 2007 Jupitermedia Corporation All Rights Reserved.
[Legal Notices](#), [Licensing](#), [Reprints](#), & [Permissions](#), [Privacy Policy](#).

[Web Hosting](#) | [Newsletters](#) | [Tech Jobs](#) | [Shopping](#) | [E-mail Offers](#)