

# Take advantage of SQL Server



Tony Rogerson, an independent SQL expert and MVP, talks to Kay Ewbank about his favourite bits of SQL Server 2008 and being part of the UK SQL User Group

**Q You're a UK SQL Server MVP and very involved with the UK SQL User Group. Who is this aimed at and what do you do?**

**A** The UK SQL User Group is aimed at anybody with an interest in SQL Server, from the core DBAs and developers, to people in smaller and medium-sized companies. At a higher level, we also have more strategic information that would be relevant to database architects. There are three main elements to what we do – the User Group, SQLBits community conference, and the SQLBlogcasts.com website. The User Group holds regular meetings throughout the country, so recent meetings have been held in Leeds, Edinburgh and Milton Keynes, for example.

**Q If a reader is interested in attending a User Group meeting, how can they find out more about what you've got planned?**

**A** Our websites at [www.SQLBits.com](http://www.SQLBits.com), [www.SQLServerFAQ.com](http://www.SQLServerFAQ.com), and [www.SQLBlogcasts.com](http://www.SQLBlogcasts.com) all have details.

**Q How do the websites fit with the User Group?**

**A** They're the online source of technical information. For example, at [SQLBlogcasts.com](http://SQLBlogcasts.com) you can view the postings of 24 active UK SQL Server experts who post regular blogs. The FAQ has answers to questions, as well as tutorials, articles and a lively forum

where you can post questions, while the SQLBits website has all the details of the SQLBits conference.

**Q What is SQLBits, and what will be the highlights of this year's conference?**

**A** SQLBits has been started by a group of individuals that are passionate about the SQL Server product suite. There is a breadth of knowledge in the SQL Community that will benefit everyone in the community. We want to spread that knowledge. We all work with the SQL community, some of us for many years, and have all been given the MVP award by Microsoft.

This year, the conference is being held on 13 September at the University of Hatfield. It's free to attend, and we have 20 sessions over four tracks given by leading SQL people. There are real-world technical sessions, and a SQL Server 2008 track, along with chalk-and-talk side sessions.

**Q What are your top three improvements to SQL Server 2008 and why are they important?**

**A** There are a lot of features that I really like, so I can't really limit them to just three. One of my favourites is the addition of the Query Governor, which lets you set an upper limit on the resources with which a query can run. We've never had the means to hive off the box hogs before, so one badly written query could cause problems for

everyone using the server. The Query Governor option now allows us to control badly behaved users to prevent this happening.

The improvements in the performance of database mirroring are another area that's really made a difference. The mirroring stream is now compressed

One of my favourites features is the Query Governor, which allows us to control badly behaved users

so data is transferred more rapidly, and you can make use of automatic page repair. This means that if SQL Server detects an error on the 'principal' database, it automatically goes to the mirror and brings it back. Data compression and encryption looks like it will be an absolute must, especially for enterprise level users.

From the viewpoint of the developer, the improvements to query performance look great. Some of these improvements come from the use of filtered indexes. For example, suppose you've got a 10 million row table, but only one per cent of the data is deemed active. You can set up a filtered index that only contains

those rows. This provides an improved performance, the index is quicker to create, and it takes less space. On a rather more detailed level, the introduction of the policy-based management framework means you can set policies for things such as naming conventions, or say that tables have to have clustered indexes, and apply that rule across the enterprise so your rules are enforced.

One thing I hated in SQL Server 2005 was Reporting Services, because I found the UI very non-intuitive. The improved version in SQL Server 2008 is so much better. Much of the GUI has been reworked, and it's no longer delivered through Visual Studio. Microsoft's purchase of Dundas has given rise to some really good charting options being added, and I'm really looking forward to using this in the field.

**Q The inclusion of the new development environments is well established now; which is your favourite SQL development language and why?**

**A** I have to admit, I wasn't a great fan of Management studio in SQL Server 2005, but the version in SQL Server 2008 seems a lot better. So far I've been testing it heavily, and I really like it. Some elements take a while to get used to, such as the AutoCompletion. This drove me mad to begin with, but now I've got to grips with how it works, it saves so much typing it's unbelievable. I do still prefer T/SQL over say C# or VB.NET because it fits so well with data processing tasks. It's so quick to just type a command or write a query when you know exactly what you want to do instead of writing what feels like a novel in C#.

**Q What do you hope to see in the next release of SQL Server?**

**A** There are two things that would be good to see. While the Resource Governor is great, one problem it

doesn't solve is that of procedure cache bloating. If you have lots of little queries, they all get cached in the procedure plan, and unless you parameterise all your queries, most never get reused. That means that the procedure cache grows, so you can end up in a situation where you make a gigabyte available for SQL Server, and end up with a procedure cache of 600MB. We need a way to control the size of the cache.

The other area I'd like to see improved is the handling of parallelism. It's easy to use parallelism to chop a query into groups of data, chop that data into streams, and gather the results together at the end. What I'd like to see added is support for operational parallelisation. If you have a query with multiple UNION operators, it would be possible to treat those UNIONS as separate queries to run in parallel. Instead, SQL Server does each UNION serially. Having the smarter parallelism would be a big improvement.

SQL Server is now a very mature product; I've been using it since version 4.21 in 1993 when, though not ground breaking in terms of features or scalability, it really started the process of opening up the market place and using

One thing I hated in SQL Server 2005 was Reporting Services, because I found the UI very non-intuitive. The version in SQL Server 2008 is so much better

a relational database in a cost-effective way. Every version since, I get closer to thinking that I, as a database professional, will be out of a job, but then the next version is released and you realise the product just gets bigger and bigger with more and more features that business can and wants to take advantage of. ■

## About Tony Rogerson

Tony has over 20 years of industrial experience, gained through his work within the IT industry. He is a highly respected and leading figure in the UK SQL Server Community, recognised through his work as founder and chairman of the highly successful UK SQL Server User Group and for his participation at Microsoft events as a SQL Server MVP.

For more on SQL Server, visit <http://technet.microsoft.com/en-gb/sqlserver>