Tech Notes

Ten Ways to Save an Hour a Day

Greg Keller, Embarcadero Technologies

January 2009

Corporate Headquarters
100 California Street, 12th Floor
San Francisco, California 94111

EMEA Headquarters
York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Asia-Pacific Headquarters
L7. 313 La Trobe Street
Melbourne VIC 3000
Australia

Introduction

For those in the data management community, including roles such as database administrators (DBA's), data architects and data stewards here, there has never been a more challenging period to effectively manage data assets within organizations. With data growing at mind bending speed and the demand for new and diverse applications further driving data growth, these professionals are seeking ways to be more productive and ultimately be free to focus on and meet their respective service level agreements (SLAs).

Data management professionals therefore need to automate as much as possible in addition to creating boiler plate like processes to their jobs. This article will outline ten helpful ideas for making your workflow more productive as a data management professional, identifying where appropriate tooling or other approaches may be implemented to raise productivity and help automate repetitive tasks.

1. Model-Driven Change Management

Modeling should not be characterized as a 'use only at the beginning' technology for solely creating new database applications. Integrating the right modeling tool into your existing database's change management lifecycle will provide effective time savings.

Select a modeling tool which facilitates clear articulation of required business changes with non-technical employees, but will also automatically calculate and generate syntactically correct database modification code allowing developers/DBAs to leverage. This process saves laborious and often error prone coding cycles measuring hours within a work day. Moreover it ensures predictability and repeatability by the operator and serves as an excellent resource for training newcomers to a database platform's required SQL alteration. This will quickly foster an understanding of the appropriate database vendor-specific approach to safely and successfully modify the structures and re-apply dependencies.

2. Abstract Database Vendor Complexity

Today, DBA's are required to manage massive infrastructures consisting of a variety of database platforms and thus must swivel their chairs continually from Oracle duties to SQL Server duties to IBM DB2 duties ... and so on. Frequently run operations such as back-ups, consistency checks (DBCCs), table alterations, index creation, security establishment, etc need to be learned in hours and mastered in days across all managed databases. Database software will come supplied with its own management tooling, but mastering those and their immense depths will inject a new set of problems obstructing the DBA from quickly learning the database itself. Look for tools which can manage lots of database types from one interface and allow the DBA to learn one common path to success. This will allow DBA's to repeat successes quickly and efficiently no matter what database they are 'thrown'.

3. Get Up On A WIKI

Data Architects and stewards receive countless and often redundant requests for reports on data ... e.g., where specific data is located, what type it is, the quality and state of the data, etc. This is most often used in data integration projects stemming from marketing and/or sales requests.

Data architects should be looking for ways to standardize an approach to find information quickly. This allows architects to move away from constant one-off requests and reports. Wiki's such as Microsoft SharePoint and Atlassian's Confluence are now staples of an organization and easily understood by anyone who can use an internet browser. Data stewards can leverage their tools (modeling or likely office productivity tools such as Microsoft Excel) to create and manage content and auto-generate this content to the Wiki. Consumers of the data are notified in real time of metadata changes through standard and familiar protocols such as RSS (e.g. like they receive their news updates) and allow sophisticated searching.

The productivity benefits here are related to developing a pattern of behavior data which stewards desire from their customers. This ultimately reduces the repetitive queries of those searching for similar data values/metadata and publishing this content in a way consumers will appreciate.

4. Reduce 'visual noise' when monitoring databases

"Eye candy" by way of dials, blinking lights and gauges sold plenty of database performance software in years past, but did little to abstract complexity and certainly didn't take into consideration the explosion of data we've seen in the last 3-4 years, tripling the number of databases born requiring management. Today, a typical DBA on average will personally manage 12-24 database instances (likely containing a multitude of 'databases') (1) with extreme cases measuring between 40-60 and more.

Tooling which will identify, for the DBA, what changed since the last time the database was working is key for success. Quick visual identifiers such as looking at configuration settings of a database install or structural changes to objects like indexes or procedures are often the culprit and easily restored to their optimal states, saving hours on the day looking for needles in a hay stack.

5. Offer pre-Fabricated Templates For Developers

Why should the wheel continually be re-invented? Data modelers should strive to create and publish a library of re-usable data structures to ensure standardization and reuse. Software development has been wildly successful with implementations such as the Gang of Four (GoF) UML patterns for software construction, and database development should be no different.

Architects can document and advertise common structures and associations/rules which developers continually need to re-create such as Bill of Materials, Address patterns or singletons such as a Customer table with all the required fields and data types. The time savings here is baked into the continual re-use and requires no downstream modification or extensive code to 'cleanse' or transform the data when it is re-purposed or moved via ETL jobs. Developers also will think less on these common structures and spend more time on critical aspects of the application.

6. Reflect on the Design to Mitigate Performance Woes

Poorly written SQL is not always the sole perpetrator of database performance problems. The design itself may be flawed. Low cost commercial tooling exists and should be evaluated which can scan the schema of a database to point out what should be fairly obvious contributors to performance degradation. Scanning for weak index design on tables as well as performing normalization checks to look at the number of joins a query will need to go through to access data are often overlooked on the original design, or are errantly changed (most frequently related to indexes) resulting in degraded performance. Applying tooling to perform these tasks automatically against the schema versus bolstering hardware or continually re-tuning code will provide major time saving benefits.

7. Virtualize

If you haven't done so already, run, don't walk to your PC and download one of a variety of open source or low-cost to free virtualization platforms such as VMWare Server or Citrix's XenServer. For those standardized on Microsoft Windows servers, its 2008 release will offer Hyper-V technology offering virtualization at its very core. For a DBA who is constantly trying to meet the demands of developers who need allocation to develop and test their code against databases, DBA's can migrate and build an entirely 'clean' version of the application on the OS the developers require while allowing a safe and quarantined environment for development. Moreover, the environment can be "re-set" by the DBA to return it to its original state at any time, which is incredibly effective for time savings. Virtualization is becoming increasingly popular as a platform to run production systems, so DBA's should make it a point to familiarize themselves with its benefits.

8. Database Snapshots

As compliance reporting becomes more pervasive and uptime requirements more critical, take advantage of readily-available commercial tools to periodically save off your performance and security settings, as well as structural information inside your most critical databases. This way, you can always look back and see what things looked like before, and understand how things have changed in an event where unwarranted access may become a reality.

9. Sniffing the Network

Carrying on the theme above, real time status of who is accessing data is an equally progressive trend in the data center for security and compliance needs. DBAs should look for tooling which will monitor data traffic to database instances, but in a way that will not degrade performance. In the past, turning logging on and then reviewing log files when needed was effective, but as user loads increased and response-time SLAs shrunk, logging became a detriment to a database's optimal performance. Tooling which will sniff and parse out packets of information traveling to the database to perform operations like queries by entering through specific ports should be evaluated as the most optimum for a databases performance.

10. Code Quality in Every Keystroke

Developers who are writing code for a variety of platforms require assistance to ensure their code is error free and optimized on the first try. The alternative unfortunately is writing code, inputting into production and then iterating by re-writing until it gets fixed. Look for tooling which will provide advice to the developer such as when to inject database specific optimization features (e.g. "use an Oracle HINT here ... ") or will parse existing names and structures to ensure appropriate connections and will also validate syntax.

About the Author

Greg Keller is a seasoned technologist, product manager, and product marketer with a marked passion for all things database-related. Greg brings more than 14 years of experience in commercial software product management and marketing to his role as Chief Evangelist for Embarcadero, where he is responsible for driving product strategy across the company's worldwide community. He previously led Embarcadero's product initiatives as Vice President of Product Management. Prior to joining Embarcadero, Greg served in key sales and product evangelism positions for the System Architect product line at Popkin Software and Systems (now IBM). Greg is a regularly featured panelist and speaker at data modeling and data management conferences, is on the editorial board for various industry publications, and is a regular contributor to numerous data management columns and discussion groups. He earned his bachelor's degree from Syracuse University's Maxwell School of Citizenship.

EMBARCADERO TECHNOLOGIES®

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.