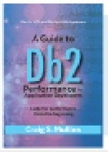


A Guide to Db2 Application Performance for Developers
By Craig S. Mullins
Order now!
A new book to help programmers write efficient Db2 code
Covers both Db2 for z/OS and LUW

ENTERPRISE Tech JOURNAL
THE MAGAZINE FOR IT STAFF IN THE WORLD'S LARGEST HYBRID DATA CENTERS

June/July 2014

Data Perspectives

Database Auditing on DB2 for z/OS

by Craig S. Mullins

amazon

DB2 Developer's Guide: A...
$77.41
Shop now

One of the more useful techniques to protect your company's data is database auditing, sometimes referred to as data access auditing. Database auditing is a facility for tracking the use of, and modifications made to, data, database resources and privileges.

Such a capability enables companies to produce an audit trail of information with regard to their database data. This audit trail can contain vast amounts of information, including what database objects were impacted, who performed the operations and when. A comprehensive audit trail of database operations, coupled with an analysis engine to identify and report on activity, lets auditors perform in-depth analysis of access and modification patterns against data in the database management system (DBMS).

Why would you need such a fine-grained audit trail? Only when

Perhaps the biggest traditional detriment to DB2 auditing is the potential for performance degradation. When many tables are being audited, particularly in a busy system, performance can suffer. Of course, trace overhead depends on many things, such as the actual amount of activity being traced, overall system activity and the number of trace records produced. Additionally, storing the audit information can become problematic when a massive number of events are captured.

But capturing the information is only part of the problem. You must then be able to access the audit trail and deliver it in a readable, useful format. Most DB2 performance monitoring tools provide audit reports, but they may not be in the best format for achieving your specific compliance goals. There are auditing tools for purchase that will read and create reports specifically for documenting compliance against governmental and industry regulations (such as HIPAA, PCI DSS and others).

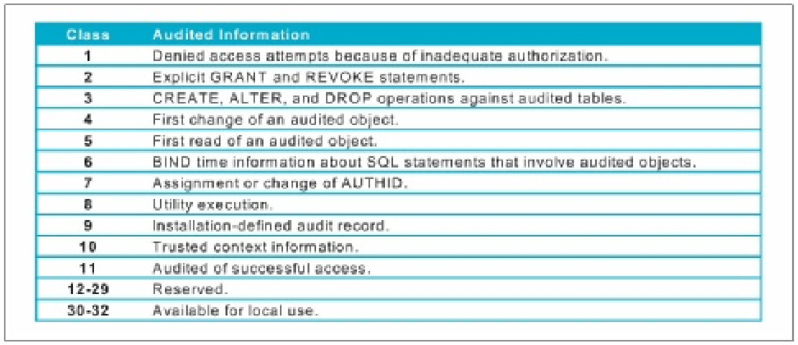
You might also consider auditing tools because they deliver armed with such details is it possible to comply with regulations, pass security audits and review the details of potential vulnerabilities. For example, many of the Payment Card Industry (PCI) Data Security Standard (DSS) requirements emphasize the importance of real-time monitoring and tracking of access to cardholder data, as well as continuous assessment of database security health status. In addition, the Health Insurance Portability and Accountability Act (HIPAA) directs healthcare providers to protect individuals' healthcare information, going so far as to state that the provider must be able to deliver a list of everyone who even so much as looked at their patients' information. Could you produce a list of everyone who looked at a specific row or set of rows in any database you manage?

From a DB2 for z/OS perspective, you can use the DB2 audit trace to monitor and track access to protected data. When you start an audit trace, specific activities are tracked, depending on the class of the trace, as shown in **Figure 1** (below).

You may have noted the terms "audited tables" and "audited objects." Simply starting a Class 3 audit trace, for example, won't start to track all data definition language (DDL) issued. The table must be audited, which means that AUDIT ALL must be specified for the table using either CREATE or ALTER. And starting with DB2 10, auditing is improved with fine-grained, policy-based auditing capabilities.

additional techniques for capturing audit details other than tracing. One such technique is to scan and parse transaction logs. By interpreting logs and identifying what data was changed, you can avoid traces. However, reads aren't captured on the logs; there are ways to disable logging that will cause modifications to be lost and it may prove difficult to retain logs over long periods.

Another data access auditing technique is proactively monitoring all activity at the subsystem level. This technique can guarantee that all requests are captured as they're made, but it can be problematic if a non-standard interface is used to capture the data. Another concern to keep in mind is that some proactive monitors sniff the network traffic for database activity. But this technique can miss activity, especially on a mainframe, because not every SQL request goes over the network. For example, it would miss CICS and IMS/TM transactions against DB2 for z/OS. Database auditing can be a crucial component of database security and compliance with government regulations. Be sure to study the auditing requirements of your organization and compare those needs against DB2's internal capabilities. Then look to augment these capabilities with additional tools to bolster the auditability of your DB2 databases.



Class	Audited Information
1	Denied access attempts because of inadequate authorization.
2	Explicit GRANT and REVOKE statements.
3	CREATE, ALTER, and DROP operations against audited tables.
4	First change of an audited object.
5	First read of an audited object.
6	BIND time information about SQL statements that involve audited objects.
7	Assignment or change of AUTHID.
8	Utility execution.
9	Installation-defined audit record.
10	Trusted context information.
11	Audited of successful access.
12-29	Reserved.
30-32	Available for local use.

Figure 1: DB2 for z/OS Audit Trace Classes

From zJournal, June/July 2014.

© 2014 Craig S. Mullins,

DB2PORTAL.com

© 2021 Mullins Consulting, Inc.   All Rights Reserved   Privacy Policy   Contact Us