Database Auditing Essentials

Tracking who did what to which data when

May 15, 2019

A webinar by Craig S. Mullins
www.mullinsconsutling.com

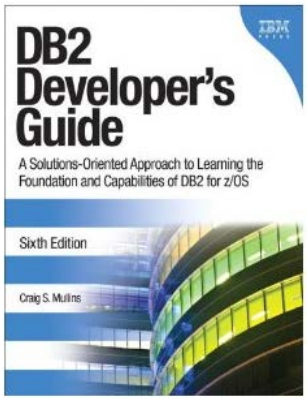
IDERA Geek Sync

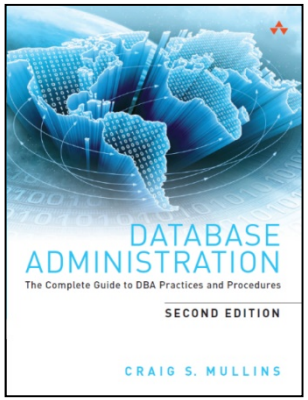
Author

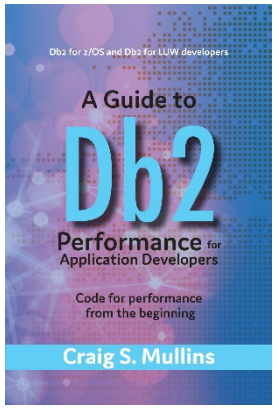
This presentation was prepared by:

Craig S. Mullins
President & Principal Consultant
IBM Champion for Analytics
IBM Gold Consultant

Mullins Consulting, Inc.
15 Coventry Ct
Sugar Land, TX 77479
Tel: 281-494-6153
E-mail: craig@craigsmullins.com
http://www.mullinsconsulting.com







Craig was named one of the Top 200 Thought Leaders in BigData & Analytics by AnalyticsWeek.

AnalyticsWeek

Agenda

Data Breach Issues

- Trends and considerations
- Cost of a data breach
- Your database is a target!

Dealing with Data Protection

- Government and Industry Regulations
- Compliance and Requirements

Database Auditing

- Who are the stakeholders
- Types of database auditing
- Database auditing methods

Data Breach Issues

Data Loss Facts and Trends

Data Breaches:
Still a Significant Threat to Your Data

Privacy Rights Clearinghouse

http://www.privacyrights.org/ar/ChronDataBreaches.htm

Since Feb 2005:

- There have been 9,094 data breaches impacting **over 11.5 billion total records** containing sensitive personal information were exposed due to data security breaches*
- That averages out to more than 12 data breaches per week
  - Starting with ChoicePoint: (Feb 15, 2005) – data on 165,000 customers breached
- In 2018 there were 828 public data breaches that impacts over 1.3 billion records
  - There were 39 public data breaches impacting almost a million records in just the first two months of 2019 alone

***** As of March 8, 2019, reported by Privacy Rights Clearinghouse

Frequency of Data Breaches

Source: 2018 Gemalto Breach Level Index
https://breachlevelindex.com/

Data Breach Statistics

Data breaches impact customer loyalty

- 64% of consumers are unlikely to do business with companies where their sensitive data was stolen

Companies bear the responsibility to protect customer data

- Governmental and industry regulations
- But also customers: 69% of whom believe companies are most responsible for protecting customer data

Customers do not believe that companies take this responsibility seriously

- 75% of consumers believe that companies do not take protection and security of their data seriously

Source: 2015 Gemalto Breach Level Index http://bit.ly/1PUCspY

The Average Cost of a Data Breach

$3.86M

The average cost of a data breach (in 2018) was up to $3.86 million

This is a 6.4% increase from 2017

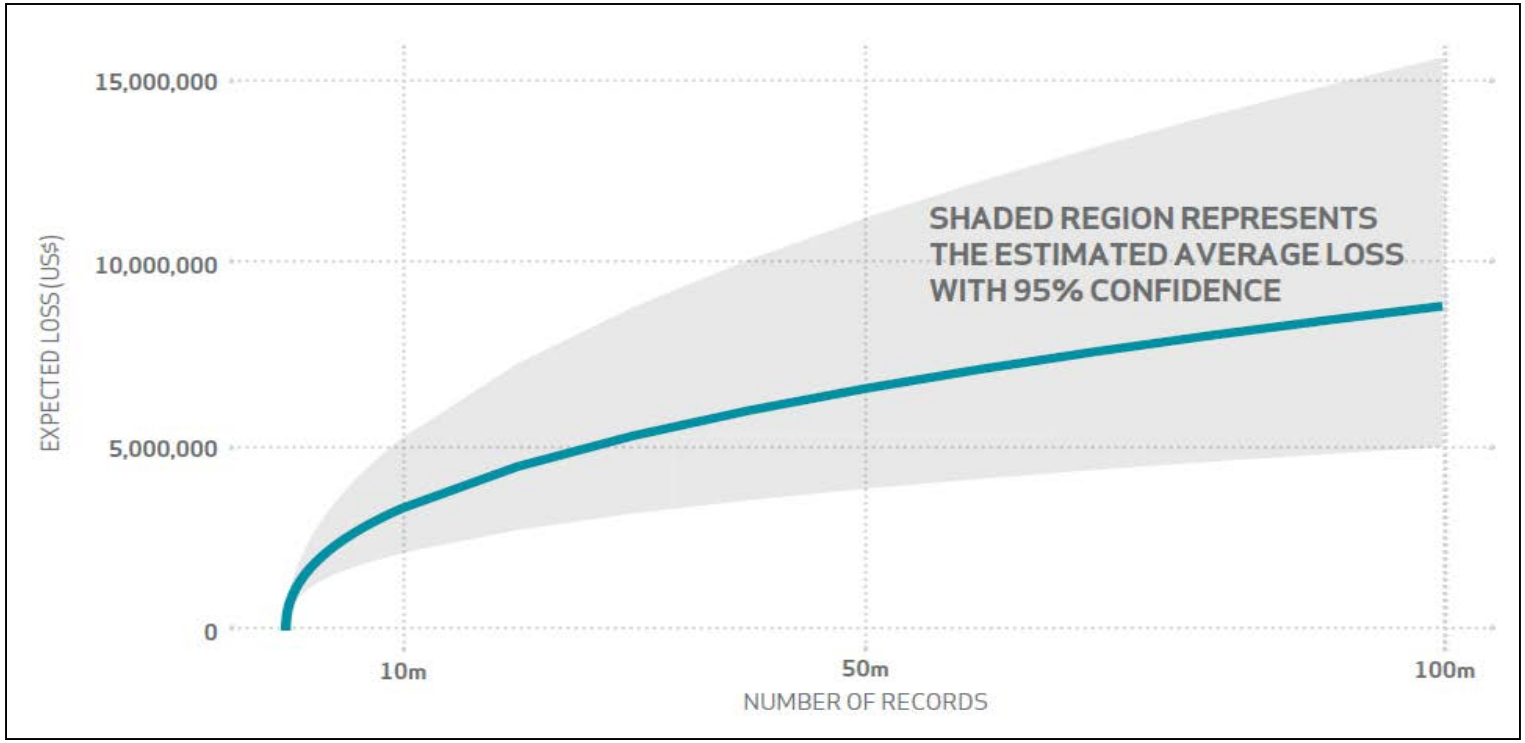
Factors that increase the cost of a breach

- **Regulatory Fines and Legal Judgments**
- **Legal Defense Costs**
- **Customer Notifications**
- **Credit Monitoring**
- **Forensic Analysis**
- **Reputational Losses**

Source: Ponemon Institute 2018 Cost of a Data Breach Study: United States

The cost of a data breach actually varies

- Average loss for a breach of 1,000 records:
  - Between $52,000 and $87,000
- Average loss for a breach affecting 10 million records:
  - Between $2.1 million and $5.2 million

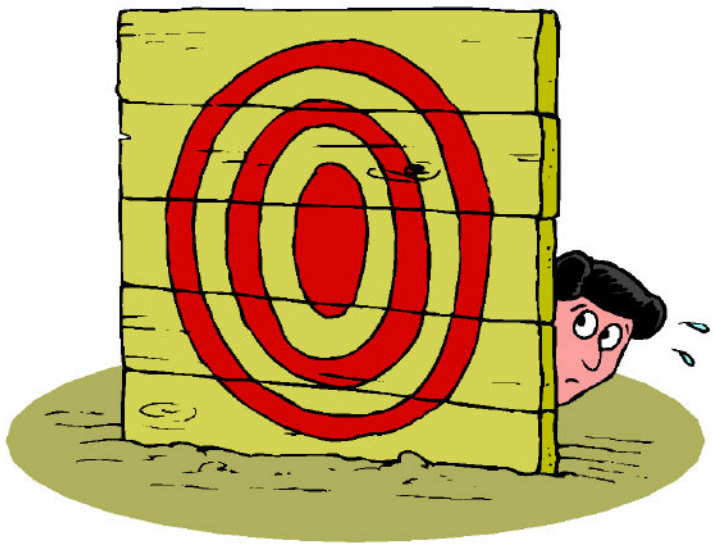


Source: 2015 Verizon Data Breach Investigation Report

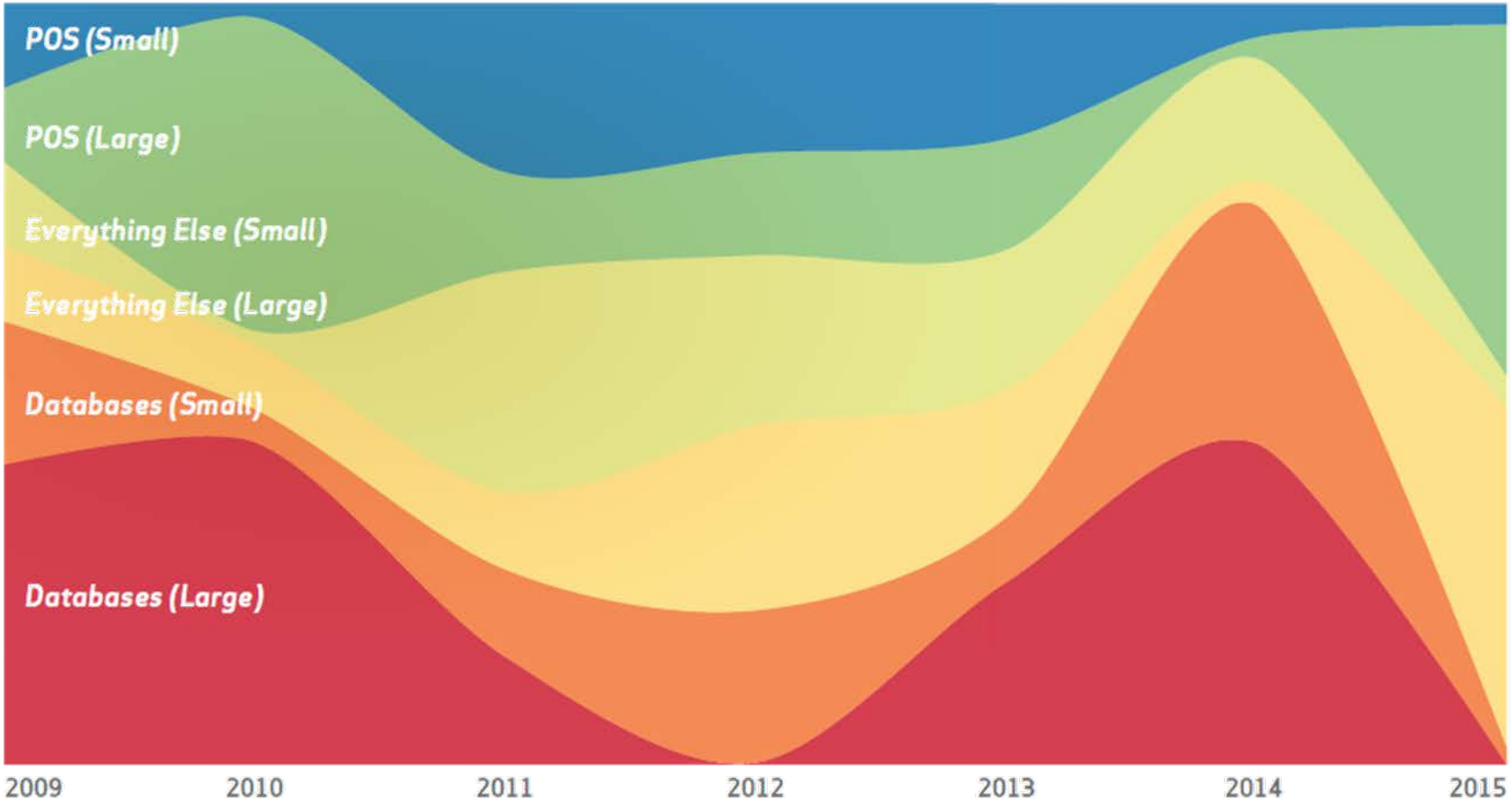
Your Database Systems are Targets for Attack

Enterprise database servers are a significant target of data breach attacks

- Because that is where the data is!
- Personally identifiable information (PII)
  - Such as SSN, address, etc.
- Personal financial data
  - Bank account/credit card information
- Healthcare data
- Etc.



Frequency of Database Attacks 2009-2015



Small company = less than 1000 employees

Source: 2015 Verizon Data Breach Investigation Report

Dealing With Data Protection Issues

Regulations and Governance

Sample Regulations Impacting Data Protection

Governance

1. Basel II
2. Sarbanes Oxley
3. Turnbull Report
4. OFAC
5. CMS ARS

Protect and control the process

Privacy

1. PCI DSS
2. HIPAA
3. CA SB 1386/AB 1950
4. GLBA
5. FCRA -- “Red Flag”
6. FISMA
7. GDPR

Protect the data

Guidance for Regulatory Compliance

It can be very complex to ensure compliance with one regulation, let alone multiple.

These resources can help to guide your compliance efforts:

- COBIT
  - https://cobitonline.isaca.org/
  - Control Objectives for Information and Related Technology
- Center for Internet Security (CIS)
  - https://www.cisecurity.org/
  - Organization dedicated to enhancing the cybersecurity readiness and response among public and private sector entities
- Department of Defense (DoD)
  - http://www.defense.gov/Resources/DoD-Information-Quality-Guidelines
  - Guidelines and procedures for information quality
- Security Technical Implementation Guide (STIG)
  - http://iase.disa.mil/stigs/Pages/index.aspx
  - Technical guidance for securely locking down computer systems & software that otherwise might be vulnerable to attacks
- Common Vulnerability Exposure (CVE)
  - https://cve.mitre.org/
  - Dictionary of publicly known information security vulnerabilities and exposures

Database Auditing by Regulation

Audit Requirement	SOX	PCI-DSS	GLB	HIPAA	Basel II	GDPR
Access to sensitive data (SELECT)		X	X	X	X	X
Modification of sensitive data (INSERT, UPDATE, DELETE)	X				X	X
Database changes/DDL (CREATE, ALTER, DROP)	X	X	X	X	X	X
Security authorizations/DCL (GRANT, REVOKE)	X	X	X	X	X	X
Security exceptions (e.g. failed logins, SQL errors)	X	X	X	X	X	X

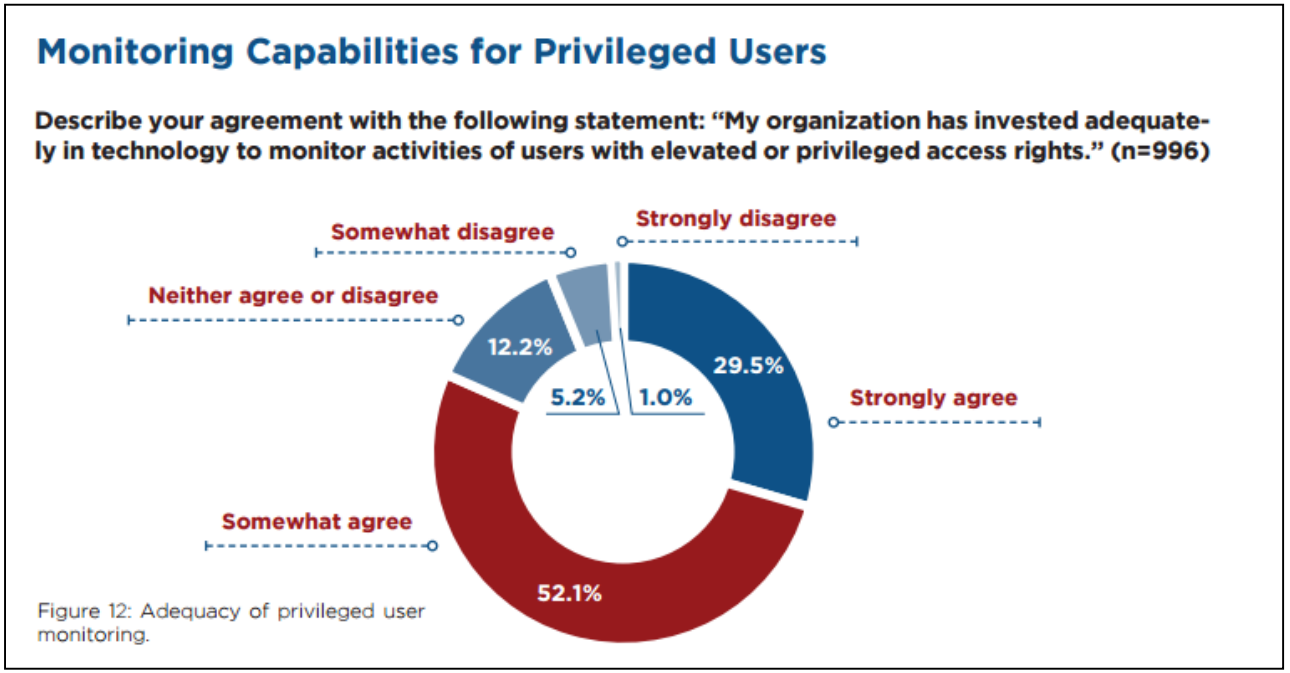
Privileged User Auditing

Monitoring privileged users is a significant aspect of compliance auditing

From a database perspective this includes:

- DBAs
- SYSADMs
- SECADMs

Not enough of this is being done…



Source: 2016 Cyberthreat Defense Report, CyberEdge Group

Database Auditing

What is it?

How is it done?

What is Database Auditing?

There are many names used for basically the same thing.

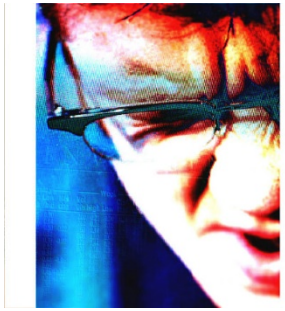
I'll call it database auditing, but you may also know it as:

- Data Access Auditing
- Data Monitoring
- Database Activity Monitoring (DAM)

My definition of Database Auditing:

The process of monitoring access to and modification of selected database objects and resources within operational databases and retaining a detailed record of the access where said record can be used to proactively trigger actions and can be retrieved and analyzed as needed.

Database Auditing Stakeholders & Requirements



SECURITY OPERATIONS

- ✓ Real-time policies
- ✓ Secure audit trail
- ✓ Data mining & forensics

COMPLIANCE AUDIT

- ✓ Business rqmts
- ✓ Separation of duties
- ✓ Best practices reports
- ✓ Automated controls

APPLICATION & DATABASE

- ✓ Minimal impact
- ✓ Change management
- ✓ Performance issues

Types of Database Auditing

Authorization Auditing

- **Who can do what.**

Access Auditing

- **Who did do what.**
- **Modifications: INSERT, UPDATE, DELETE**
- **Reads: SELECT**
- **DDL: CREATE, DROP, ALTER**
- **DCL: GRANT, REVOKE**
- **Utilities: Load, Unload, Export, Import, Copy…**

Replication Auditing

- **Who copied which data where.**

Don’t Forget About Non-Standard Access?

- File-level snooping as opposed to going through the DBMS interface
- System-level snooping and zapping (e.g. pointers)
  - Such as AMASPZAP on mainframes https://www.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.ieab100/iea3b1_AMASPZAP.htm
- Image copy backup data sets
- Unload data sets
- REORG data sets
- Database statistics
  - Some stats, like columns distribution statistics, contain values

General Database Auditing Requirements

- Selective
  - Must be rules-based to enable the capture of audit details only on the specific data that requires auditing.
  - Should be able to implement privileged user auditing, auditing by application, by database object, by type of command, by utility, etc.
- Comprehensive
  - Must be able to capture the complete scenario of auditable information.
- Non-invasive
  - Should be able to audit access to data without incurring expensive performance degradation.
- Be capable of answering:
  - Who accessed/modified the data?
  - When did it happen?
  - Using what computer program or client software?
  - From what location on the network?
  - What was the SQL query that accessed the data?
  - Was it successful; and if so, how many rows of data were retrieved?

Database Auditing Approaches

There are 6 common methods used to audit databases:

1. Audit within the DBMS (traces)
   - Must start performance trace
2. Audit using temporal capabilities
   - SYSTEM temporal tables to record all changed data
3. Audit using triggers *(or hand-coding)*
4. Audit from the database transaction log files
   - Modifications are on the log anyway so...
5. Audit over the network
   - Sometimes called network sniffing: captures SQL requests as they are sent over the network
6. Audit directly against the DBMS server control blocks
   - Sometimes called a “tap”

1. Native DBMS Audit

Most DBMSes offer an audit or trace capability that enables you to trace events or categories of events by UserID, object ownership, etc.

You can typically choose from multiple audit options such as:

- Categories of events
  - SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REVOKE, GRANT, etc.
- Privileged User Access
- Specific Tables, UserIDs, programs...

1. Native DBMS Audit

Potential Native Audit Issues:

- **Separation of duties** – auditing typically is turned on and off by DBAs (privileged users)
- **Overhead** – some audit traces can consume a significant amount of resources
  - Don't want to "dim the lights" to audit
- **Comprehensive capture** – may not capture everything that needs to be captured for compliance
  - First read/write in a UOR
  - Depends on the DBMS and the type of auditing

    There can be multiple types of native audit per DBMS
- **Audit trail access** – how do you access the audit trail; may require significant programming + support

2. SYSTEM Temporal Tables

A newer capability of several DBMS offerings is the ability to create temporal tables

A system-level temporal table can be used to audit changes (aka transaction time)

- System time tracks the insertion and modification history of the data.
- System Time is typically tracked using two tables.
  - One table contains the current data.
  - Another, history table, contains the non-current data.

2. Implementing System Time

- **Implementing auditing using SYSTEM time temporal tables…**
- Requires a 3 step process:
  1. Create the base table and include TIMESTAMP columns to track the the starting and ending points for the system time period.
  2. Create the history table with an identical structure, preferably by using a CREATE TABLE . . . LIKE statement.
  3. ALTER the current table to enable versioning thereby turning on the system temporal capability.

2. Temporal: Adding Data to our System Time "Table"

– After inserting these three rows →

```
INSERT INTO COURSE
  (COURSENO, TITLE, CREDITS, PRICE)
VALUES
  (500, 'INTRO TO COBOL', 2, 200.00);

INSERT INTO COURSE
  (COURSENO, TITLE, CREDITS, PRICE)
VALUES
  (600, 'INTRO TO JAVA', 2, 250.00);

INSERT INTO COURSE
  (COURSENO, TITLE, CREDITS, PRICE)
VALUES
  (650, 'ADVANCED JAVA', 3, 400.00);
```

– Our tables look like this...

COURSE table

COURSENO	TITLE	CREDITS	PRICE	SYS_START	SYS_END
500	INTRO TO COBOL	2	200.00	2012-01-10	9999-12-31
600	INTRO TO JAVA	2	250.00	2012-01-10	9999-12-31
650	ADVANCED JAVA	3	400.00	2012-01-10	9999-12-31

The table contains three rows

COURSE_HIST table

COURSENO	TITLE	CREDITS	PRICE	SYS_START	SYS_END

The table is empty

2. And then let's DELETE from our System Time "Table"

– After issuing these statements (UPDATE and DELETE) →

```
UPDATE COURSE
SET PRICE = 375.00
WHERE COURSENO = 650;

DELETE FROM COURSE
WHERE COURSENO = 600;
```

– Our tables now look like this...

COURSE table

COURSENO	TITLE	CREDITS	PRICE	SYS_START	SYS_END
500	INTRO TO COBOL	2	200.00	2012-01-10	9999-12-31
650	ADVANCED JAVA	3	375.00	2012-01-15	9999-12-31

The table contains two rows

COURSE_HIST table

COURSENO	TITLE	CREDITS	PRICE	SYS_START	SYS_END
650	ADVANCED JAVA	3	400.00	2012-01-10	2012-01-15
600	INTRO TO JAVA	2	250.00	2012-01-10	2012-02-05

The table contains two rows

2. System Time: Sample Queries (#1, #2)

AS OF
FROM
BETWEEN

- Simply retrieve the current info for Advanced Java course:

```
SELECT *
FROM   COURSE
WHERE  COURSENO = 650;
```

- Access info about Advanced Java course for January 16, 2012:

```
SELECT TITLE, CREDITS, PRICE
FROM   COURSE FOR SYSTEM_TIME AS OF TIMESTAMP('2012-01-16')
WHERE  COURSENO = 650;
```

  - Returns one row →

TITLE	CREDITS	PRICE
ADVANCED JAVA	3	375.00

2. Issues With Using Temporal to Audit

- **Only useful to track modifications** (U/I/D)
  - No tracking of access to data (read/SELECT)
- **No tracking of who made each change**
- **No way to audit privileged users**
- **Data management issues**
  - Large amount of “old” data
  - Needs to be purged – but within the specifications of the regulations being complied with!
  - Uses the database itself to store audit data
  - Separation of duties: a DBA can get past it
    - Disconnect the temporal connection
    - Delete data
    - Reconnect

3. Using Triggers

Similar to the temporal option, you can use database Triggers to write audit records

- Write a set of Triggers for the tables you wish to audit
  - One each for Insert, Update, and Delete
- Write the pertinent details – whatever you want – to audit tables you define
  - Or files if your DBMS allows triggers to write to files

3. Issues With Using Triggers to Audit

- **You have to write and maintain the Triggers**
  - This can be an exposure based on who has access to the trigger code
- **No tracking of access to data** (read/SELECT)
- **No way to audit privileged users**
- **Trigger performance can be poor**

- **Data management issues similar to temporal data**
  - e.g.) amount of data, purging, using the database to store audit data, separation of duties

3½. Hand-coding audit trails?

Sometimes developers add "audit columns" to tables, such as LAST_MODIFIED_DATE

- The idea here is for the program to auto-matically change the LAST_MODIFIED_DATE column in the programs whenever data is changed
- Every program?

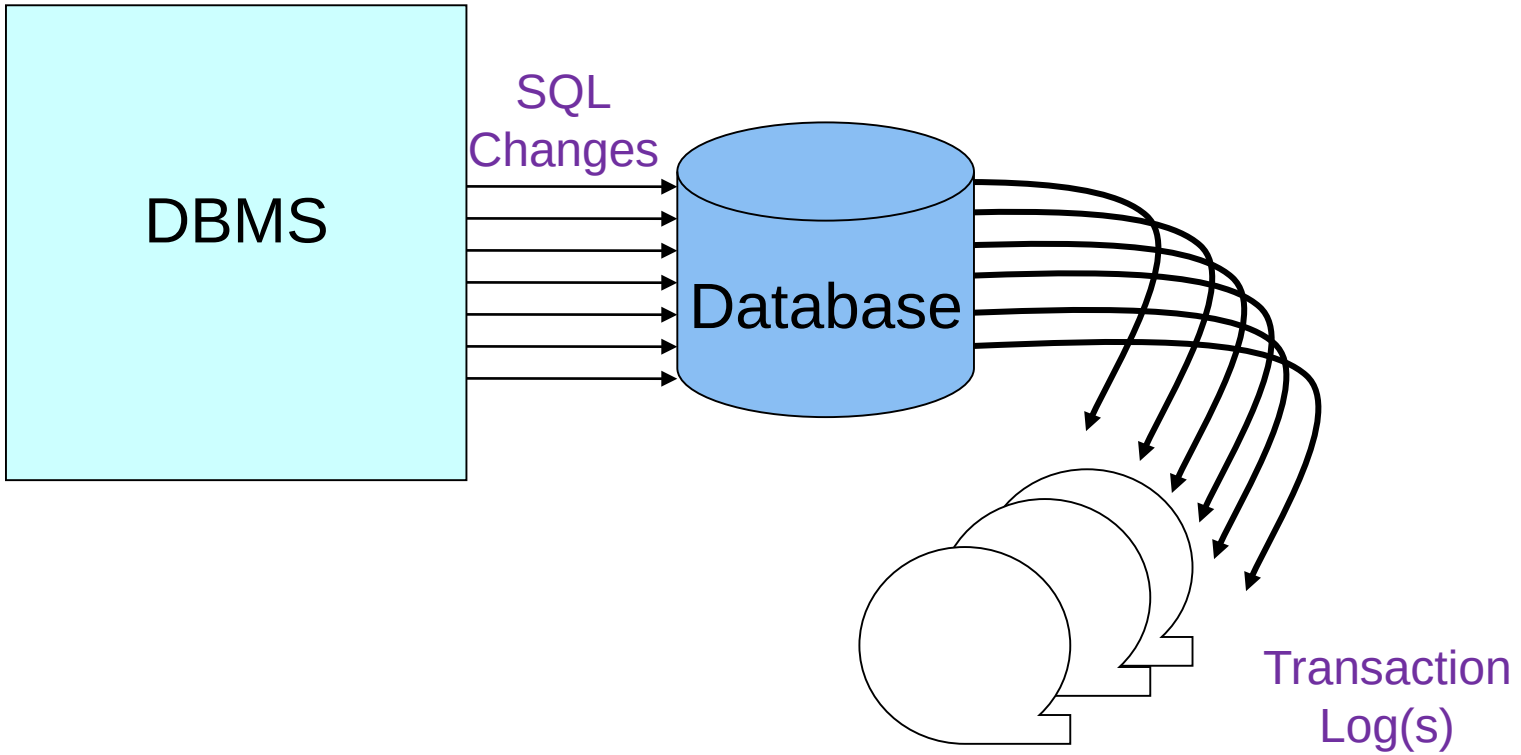
Auditors don't like this… it is a problem because:

- Audit trails should be kept outside of the database
  - If you delete the row you lose the audit data
- Can you guarantee that LAST_MODIFIED_DATE is accurate?
  - Couldn't someone have set it by accident (or nefariously)?

Not really a good idea

4. What About Using the Database Logs?

Database transaction log(s) capture ALL* changes made to data.

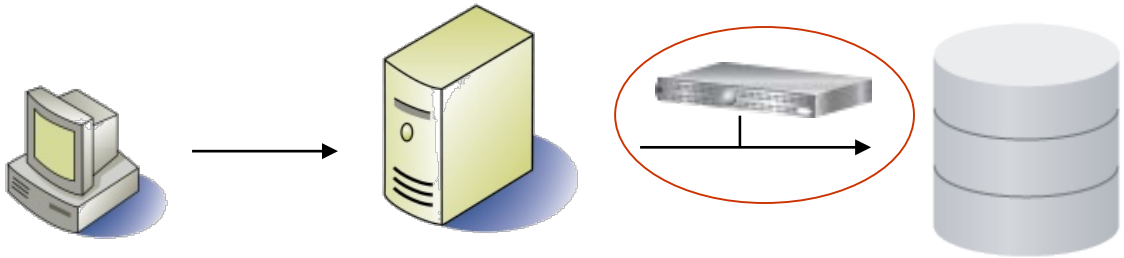


*Well, maybe not all changes, all the time.

4. Issues With Database Log Auditing & Analysis

- Log format is proprietary
- Volume can be an issue
- Easy access to online and archive logs?
  - But how long do you keep your archive logs?
- Dual usage of data could cause problems?
  - Recovery and protection
  - Audit
- Tracks database modifications, but what about reads?
  - Transaction logs do **not** record information about SELECT.
- And what about non-logged operations?
  - LOAD LOG NO, REORG LOG NO
  - Non-logged table spaces, LOBs
- Cannot invoke real-time actions using log-based auditing

5. Network Capture



Database auditing via network sniffing captures SQL requests as they go across the network.

- But not all requests go across the wire
  - DBA access directly on the server
  - Mainframe applications (CICS, IMS/TM, TSO, batch)
- Some third-party database auditing solutions use this approach

6. Tap Database Server Control Blocks

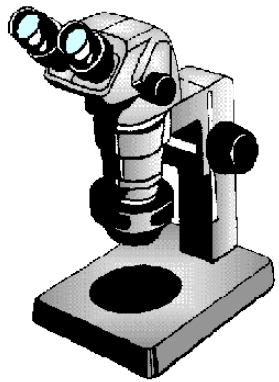
Audit database requests at the server

- Capture *all* SQL requests at the server
- All SQL access can be audited, not just those made over a network
- Retain all pertinent audited information

  Without relying on the DBMS
- No need to keep the active/archive log files
- No need to start a DBMS trace
- No need to modify the database schema
- Concerns?

  Requires purchasing additional ISV software

  Interfaces with DBMS internals



The Best Approach?

All have their merits depending upon the type of database applications that you need to audit

- Database auditing products may use one or multiples of these approaches and methods
  - Understand the methods used by any product you evaluate
- For example, a combination of native tracing and capturing information from database control blocks can make for a reasonable solution

Benefits of Database Auditing Products

- **Built-in, pre-defined support and reporting for compliance with regulations like GDPR, HIPAA, etc.**
- **Vendor maintenance and support for changing DBMS capabilities and regulatory compliance specifications**
- **Customizable policies and alerting**
- **Typically come with a dashboard to view compliance information, audit activities and issues**
  - Built-in support for specific regulations
- **Typically consume fewer resources than alternate approaches**

Contact Information

Craig S. Mullins

Mullins Consulting, Inc.
15 Coventry Ct
Sugar Land, TX 77479

E-mail: craig@craigsmullins.com

Web: www.mullinsconsulting.com

Data & Technology Today Blog

https://datatechnologytoday.wordpress.com/

DBA Corner Columns

http://www.mullinsconsulting.com/dba-corner.html

IDERA
An Idera, Inc. Company

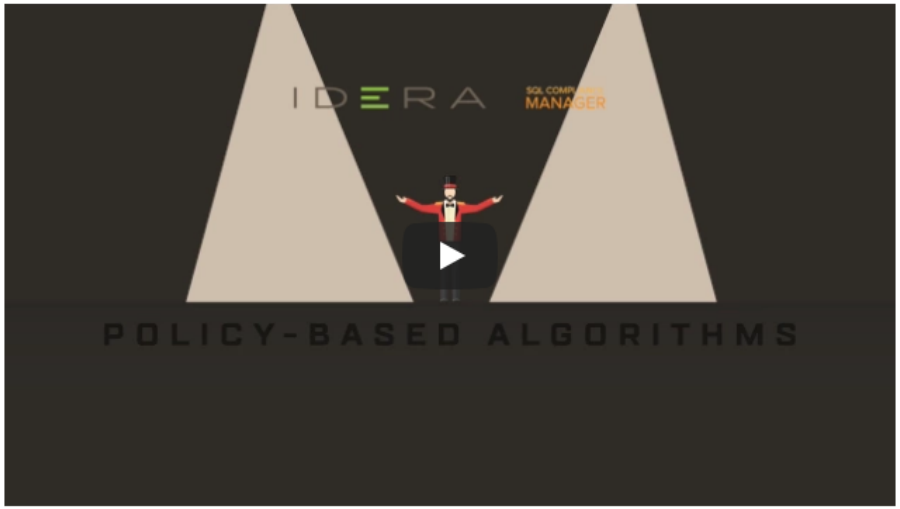
Products | Free Tools | Solutions | Resources | Community | Customer | BUY NOW

Improve Any SQL Server Audit

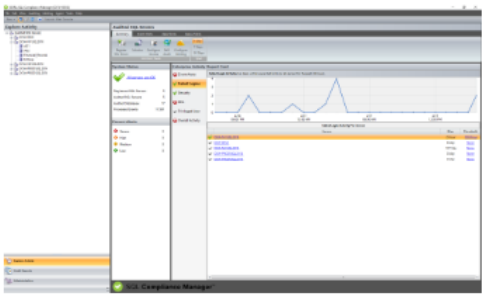
SQL Compliance Manager

Overview | Details | Tech Specs | Resources | Compare | Tour | Download

Start for FREE



No credit card required!
Fully functional for 14 days



- Audit sensitive data to see who did what, when, where, and how
- Monitor and alert on suspicious activity to detect and track problems
- Satisfy audits for multiple industry regulatory requirements
- Select from over 25 pre-defined compliance reports and create custom views
- Lightweight data collection agent minimizes server impact
- Web-based dashboard simplifies access from any browser

Buy Now

Includes first year of maintenance.
Volume discounts available.

Add to Cart

Need multiple licenses?

Save up to 45% with multi-license discount pricing.

Request a Quote

https://www.idera.com/productssolutions/sqlserver/sqlcompliancemanager