Communications of the IIMA

Volume 11 | Issue 1

Article 2

2011

The Causes, Security Issues, and Preventive Actions Associated with Data Integrity

Steve Hallman
Park University

Al Stahl
Park University

Vugar Ahmadov
Park University

Recommended Citation

Hallman, Steve; Stahl, Al; and Ahmadov, Vugar (2011) "The Causes, Security Issues, and Preventive Actions Associated with Data Integrity," *Communications of the IIMA*: Vol. 11: Iss. 1, Article 2.
Available at: http://scholarworks.lib.csusb.edu/ciima/vol11/iss1/2

The Causes, Security Issues, and Preventive Actions Associated with Data Integrity

Steve Hallman
Park University
USA
hallmanDBA@Yahoo.com

Al Stahl
Park University
USA
astahl_328@yahoo.com

Vugar Ahmadov
Park University
USA
vugar.ahmadov@park.edu

ABSTRACT

Data integrity is a major issue in today's world. Many organizations have faltered due to the nature of the date contained within their databases. Some organizations are failing as a result of inaccurate, modified, or incorrect data that is contained with organizational records. Still others are losing customers and clients, and yet others are the subject of malicious lawsuits based upon inaccurate decisions and subsequent actions derived from inaccurate, modified, or missing data. This article explores many of the sources of and issues relating to the entry, storage, and retrieval of data from databases. The article also examines several avenues for reducing data integrity problems, including the adequate training of users, who too often tend to be the inadvertent source of errors. The article was written to highlight several potential sources of errors, because data integrity is so critical for every organization.

INTRODUCTION

Databases represent powerful capabilities that are designed to track a host of different organizational records, which often include information regarding employees, transactions, finance, inventory, client history, supplier inventory, customer bases, and even internet interactions. Databases have become such an important aspect of organizational activity that without them, many organizations would cease to exist, and the veracity of others (including governments) would be greatly impacted. Why then, does so little attention appear to being given to ensuring the integrity of the data that is stored in a database?

Data integrity is so critical to any organization that maintains electronic records including: corporations, governmental agencies, non-profit organizations, service groups, medical practices and educational institutions. If the integrity of records is compromised, the impact on the organization could be horrific, resulting in financial records being exposed, the theft of customer or client identities, the exposure of strategic initiatives, loss of business, and even the malicious transfer of funds, all of which are potential outcomes, when an organization's database technologies are compromised.

In today's electronic world, data is so extensive and access to that data so important that it must be managed through technology, in order to provide the type of access that is needed in order for organizations to be responsive and competitive. However, the keys to data integrity include more than "technology watching technology." This article takes the position that users may be an untapped catalyst that when appropriately trained, encouraged, and acknowledged can make a major difference in data integrity.

Background

Data is defined in *Merriam-Webster's Collegiate Dictionary* ("Data," 2011) as "factual information (such as measurements and statistics) used as a basis for reasoning, discussion of calculation." The meaning of the word has expanded through technology to be more encompassing and is now inclusive of anything that can be stored in digital form, on a computer. Data is so plentiful that other terms like metadata (meaning data about data) have come into use in recent years.

There is evidence that data stored in databases have a significant rate of errors. Experts believe that between 1% and 10% of the data stored in databases [closer to the later] is incorrect, based on a survey conducted by Klein, Goodue, and Davis (1997), If such errors continue to go unspotted, then it likely will affect organizational outcomes and effectiveness. Some early-published research has implied that users of information systems (IS) tend to be ineffective in finding data errors. However, two relatively recent studies show that explicit error detection goals and incentives can modify user behaviour in error detection performance (Klein, et al., 1997). These findings provide an improved understanding of some of the conditions under which users can be more effective in detecting data errors.

The Problem

For the most part, *database integrity* is about trust. Can the users and businesses trust the data stored in their databases? Data integrity is a term that reflects the validity, veracity and reliability of the data, which is contained within a database.

> "Data should be managed where it resides in volume." That's the storage management mantra a colleague use to intone, every time the subject of enterprise backup arose. This sound philosophy has steered many network administrators, engineers and consultants . . . through successful storage management projects, over the years. (Rodgers & Rodgers, 1997, para. 1).

Unfortunately, data integrity has become a real issue as witnessed by the increasing number of instances of data thievery that seems to be reported in the news almost daily. Headlines

frequently seem to involve actions like the illegal transfer of funds from a bank, identity theft, and security breaches that compromise strategic organizational initiatives.
The issues representing data integrity are extensive and encompassing. Consider the plight of a network manager whose network consists of huge gigabyte (GB) or terabyte (TB) of data on Microsoft Windows Exchange/BackOffice servers spread across the corporate WAN, plus many GB on a Unix database server (Frey, 1997). Because of distributed Win-NT servers and relatively slow WAN links, this manager wisely decides to forego the idea of backing up all of the network data to a central Unix solution.

> Instead, humming the mantra loud and clear, he achieves success by backing up the Windows NT data with an NT-based solution (that can be managed in a distributed environment) and backing up the Unix data with either a Unix remote client agent or a native Unix solution (Rodgers & Rodgers, 1997, para. 1).

Key Issues Impacting Data Integrity

Data Authenticity: Many software companies have enabled most consumer firms to have large databases, where data is easily retrieved, updated or evaluated at nearly the same time or within a few seconds. These activities are crucial when every individual of the organization is in need of data. There is no allowance for wrong findings, which is where authenticity of the data comes into the picture. Imagine that a client's order from a computer company is 5400 PCs. The order is recorded as 4500 PCs (as a result of a typo), and the PCs need to be distributed to different sites within an exact timetable; there would be chaos! Database companies, such as Oracle and Counterparts, have devised software that incorporates a system, where data such as these, are checked to avoid miss typing and other similar errors.

Replication: A corporation also relies on databases to be updated constantly, preferably automatically. For example, an organization that has many offices will have many users accessing databases from the server online, as well as off-line. It is imperative for the decision-makers to have access to the most resonate data for meetings and to formulate appropriate business strategies. Thus, data replication is crucial and needs to be online and immediate.

Where data are constantly retrieved and updated, software formulators are incorporating some of these features. However, the challenge lies in updating online, especially in the case of multinational companies, where multiple time zones play an important role in updating data. Updating data works on an automatic time clock; however, it is a challenge for software to update data to the minute and not involve itself in the time lag of five hours or six hours.

Reliability: Computers running on electricity may shut down unexpectedly, and the result can be partially updated databases. When this happens, it becomes dangerous for the users, as they may have only half of the updated data. In this case, the database is currently not in a valid state, and rollback can be used to recover the database to a valid state by undoing the problem transaction (Pratt & Adamski, 2008). This is another reason that data integrity is so important. Databases today, are equipped with verification systems, which ask the user to analyze the changes and then save them, so that anyone who retrieves the data will have the correct version. This also ensures against miss feeding of data through typing.

Other Data Integrity Issues: Rabitti, Woelk, and Kim (1988) has clearly stated that there are at least three types of data integrity that must be designed into any database.

1. Key Integrity: Every table should have a primary key. "The primary key must be controlled so that no two records in the table have the same primary key value. In addition, the primary key for a record must never be allowed to have a null value. Otherwise that would defeat the purpose of the primary key to be a unique identifier" (Rabitti et al., 1988, pp. 231-250). If the database management system does not enforce these rules, other steps must be taken to ensure them.
2. Domain Integrity: "Appropriate control must be designed to ensure that no field takes on a value that is outside the range of legal values." (Rabitti, et al,, 1988, pp. 231-250). For example, if grade point average is defined to be a number between 0 in 4, controllers must be implemented to prevent negative numbers and numbers greater than 4. "For the foreseeable future the responsibility for data editing will continue to be shared between the application programs and the DBMS." (Rabitti et al., 1988, pp. 231-250).
3. Referential Integrity: "The architecture of relational database implements relations between the records in tables via foreign keys. The use of foreign keys increases the flexibility and scalability of any database, but it also increases the risk of integrity errors. This type of error exists when a foreign key value in one table has no matching primary key value in the related table" (Rabitti et al., 1988, pp. 231-250). For example, an invoice table usually includes a primary/foreign key, a customer number, to reference back to the matching customer number primary key in the customer's table.

One method that may assist in preventing many of these types of errors involves the deletion of customer records. When deleting customer records, automatically delete all involved records that have the same customer number.

Data Security Issues

Database security issues exist as long as database storing has begun. However, it has become a large topic in recent time as the areas using database storing has expanded dramatically. Viruses, hackers, worms, trojans, phishing, malware, and many other challenges threaten the security of databases. Many organizations today, tend to focus on using technologies for building, maintaining, and accessing databases while not devoting much focus to security (Britt, 2007). The ability for an attacker to gain illegal access to a database, and compromise precious data, places an organization at great risk. An attacker would have the ability to gain sensitive data such as credit card, social security numbers, passport IDs, or other personal identification, if one was to break into a database, many of which only have "light security measures."

The overall integrity of the data then becomes at risk. According to Mohamed, attackers can discover flaws in SQL databases and could "use higher privileged code with DBMS_SQL to perform, insert, update or delete records, and so change the data within the database directly" (2006, para. 6). In his example, Mohamed (2006, para. 6) states that:

> In a case where the data being inserted, it must not contain single quote marks, because while the higher privileged code checks for its presence, the attacker can "snarf" and replace data, so that it does contain a single quote mark, thereby causing an exception.

Many organizations are under pressure about their security as hackers find more sophisticated ways to find vulnerabilities to exploit, in order to gain valuable data from databases (Patterson, 2007). Currently, 83% of the organizations in the U.S. "believe they have made their data safer by installing or upgrading antivirus software, installing or upgrading a firewall, implementing intrusion detection/prevention technologies, and implementing vulnerability/patch management systems on their networks" as reported by Communications News (Patterson, 2007, p. 8). The truth is that no matter how sophisticated the protection becomes, so do the methods for breaking into databases and acquiring precious data from a database. In 2005, the number of security breaches reported reached 100 million and is still growing according to Privacy Rights Clearinghouse, who is a non-profit consumer information and advocacy program (Patterson, 2007).

Currently, many organizations use passwords to prevent unwanted individuals from gaining access to databases. This is a very common method of security, but is not always an effective one. Many passwords used, unfortunately are not effective, because many are too easy or too common for an intruder to discover (Britt, 2007). In order to have an effective password, it is recommended that a password be at least twelve characters long and include capital letters and numbers (2007). However, even though a more complex password can help to protect the system, it also creates a new problem for users, who, too often, seem to forget their passwords.

One way (which is recommended) of increasing database security is by adding security layers. In a layered security system, multiple security methods are taken instead of just one, to protect the system. Multiple methods can include having: firewalls, access controls, passwords, and encryptions, as well as various monitoring systems all of which would be simultaneously active (Britt, 2007). This approach is ideal for databases, which have large amounts of sensitive data to protect.

Databases are also vulnerable to eavesdropping due to the nature of decentralized communication used on the internet. "SSL (Secure Sockets Layer) certificates are used to ensure that a user can interact with a database in a secure fashion, even over insecure wireless networks" (Bickford, 2010, p. 2). As usage of online payments and storage of sensitive information increase, the database security becomes a major concern for businesses and individual customers. The companies that provide access to millions of records with bank accounts, credit card numbers, or social security numbers have very high responsibility to protect their databases. As sophisticated the software that the company uses, it is more difficult for the hackers to get access to their information.

Vizard (2007), who has done research on security threats, believes that a uniform approach to security needs to be created for the future. He believes "security tests need to be implemented before a customer or another business interacts online with an organization" (2007, para. 7). Though these measures may seem harsh, they may become ideal for organizations of the future.

Currently, there are database security tools that are being used today, in order to test the security on a network. Symantec has a security tool that is being used by hospitals in the Boston area, since 2006 for alpha testing. Though this tool does not stop unwanted intruders from entering the network, it monitors all activity on the network and can show an organization if suspicious activity is present. It can also alert an organization if someone is attacking a database and trying to alter or steal its information. Although this tool does not directly protect the database, it is still a huge improvement over previous database tools, which did not monitor network activity, nor did they test to see whether a database or its associated network is safe (Mesmer, 2006).

Database Complexities Impacting Data Integrity

Corporations and other organizations today are dependent on the authenticity of the data provided to them through their computers. Whether it is a multinational corporation working on a worldwide network, or a local company using a vast database to operate within the firm, each depends on valid, quickly accessible data to make crucial decisions. Thus, it is important to analyze and evaluate a system that is being incorporated into an organization with regards to its usage and its ability to process data.

Apart from recovering files that have been accidentally deleted, one of the main reasons an organization backs-up data is to safeguard against disasters. One disaster-recovery method requires that the hard drive be partitioned, formatted, and set up to reload the operating systems, prior to the actual recovery of the data. Another method recovers the partition and master boot record on the fly. For some, it is possible to gather the required device drivers on a floppy disk or tape, which may seem to allow for easier recovery. However, such an option usually does not actually result in the creation of a *bootable image*.

Choosing enterprise backup software increasingly hinges on add-ons such as database agents, application agents, image backup options, client agents, accelerator or interleaving client agents, redundant array of independent disks (RAID) options, open file agents, e-mail system agents, and antiviral integration; all of which can help create a backup operation.

Another particularly thorny problem for enterprise backup systems is that databases need to be up and running 24 hours a day, seven days a week. That is because many of the files associated with the database or application are open 24 hours a day, seven days a week. A similar problem arises when backing up e-mail or Groupware systems, most of which are databases in their own right. Most major database vendors have added software application programming interfaces (APIs) or hooks that place the database engine into a maintenance or backup mode, which facilitates successful backup of the database or database objects, while maintaining data integrity (Frey, 1997).

An Organizational Example

One example of the complexities related to data integrity that, in some way, is inherent in every organization is finance. Finance involves accurately maintaining massive amounts of data (ArkiData, 1999).

Totalling numbers, itemizing expenses, and producing detailed financial reports have traditionally been the tasks of corporate financial departments. Nevertheless, like many other business operations, the finance function is undergoing a significant change as organizations make better use of their internal resources to become even more competitive / service oriented.

Financial managers now must spend more time managing both financial and non-financial information that affects the future growth and competitiveness of their organizations. Issues such as market share analysis and business management are just two examples of areas that could affect growth of an organization, and where the integration of financial data and a financial perspective leads to better strategic decision-making. In many organizations today, business decisions that have significant financial implications are often made without a comprehensive understanding of their short-term and long-term financial impact on the organization. Often, such organizations continue to under-utilize their financial staff people by failing to leverage the valuable skills and experience in the analysis and disciplined thinking that they can offer in the competitive decision making environment.

More and more companies are now asking what else can be done, as financial professionals, from chief financial officers to department managers, play an increasingly critical role in strategic decision-making. Many finance professionals are finding this new role difficult because they don't always have easy access to the corporate information that they need. As a result, many finance departments are not integrating other business issues into their financial reporting.

One important way to address this problem and successfully increase the role of the Finance function is to free financial staff from manual data collecting. Such a move would provide easy access to financial data and minimize manual adjustments to the data-management maintenance functions that currently take up so much of their time. Accounting and other related processes are equally as important.

In general, studies show that “financial professionals spend 80 percent of their time collecting and managing data, and only 20 percent of their time studying and analyzing specific trends and opportunities that could help the business grow” (Klusek, n.d.). Finance can be a high-cost function, where it would not make much sense to spend large amounts of money to employ data clerks and data managers, when technology can do most of the tasks that they would undertake.

Some finance departments are using desktop productivity tools such as MS-Excel/Lotus Notes spreadsheets to generate the information required by decision-makers. The problem with this approach is that it involves manual re-keying of data from general ledger and other corporate databases. Not only is data integrity compromised (as a result of errors derived through retyping data), but people also spend much of their time on what tend to be administrative and clerical tasks, rather than providing their expertise in value-added analysis of important financial and non-financial information. Unfortunately, desktop generated spreadsheets also tend to be rather inflexible and difficult to manage, if there are major changes to the original data required, or if the assumptions used to generate the spreadsheet change. These sorts of modifications have to be manually undertaken, which can be very time-consuming and costly.

New enterprise-based technology solutions have overcome many of these problems by integrating financial and other corporate data into a single system or database. These programs are enabling organizations to maximize their efficiency in transaction processing, or other financial matters, and minimize the manual clerical tasks that historically have taken up so much of the finance department's time.

New business support software programs that utilize on-line transaction processing (OLTP) systems are able to automatically capture and process transactions quickly and accurately. Organizations can process data by time, location, client, category or other variables, depending on their specific organizational requirements.

Auditing: "Database auditing does not prevent security breaches, but it does provide a way to identify if breaches occurred" (Murray, 2010, p. 73).

Database auditing allows the company to track database access and all the activities. This can be used to identify what actions were performed, who made changes to data, and what data was changed. Common categories of database auditing include monitoring database access attempts, data control language (DCL) activities, data definition language (DDL) activities, and data manipulation language (DML) activities (Yang, 2009). Monitoring access attempts includes retaining information on successful and unsuccessful logon and logoff attempts. DCL audits record changes to user and role privileges, user additions, and user deletions. DDL audits record changes to the database schema such as changes to table structure or attribute data types. DML audits record changes to data. In addition, database errors should be monitored (Yang, 2009).

CONCLUSION

Integrity is the most potentially volatile part of the success of any database. A well-designed and maintained database (by the users, programmers, and management) can ensure key domain, and referential integrity. Accessibility and network security tools are very important aspects of data management activities. However, an understanding of viruses, malware, and other issues that threaten the integrity and security of databases, on the part of the users, may be a primary means of ensuring data integrity. The ability to plan and perform proactive manoeuvres is paramount and essential for success of today's businesses. Making all users aware of and skilled at implementing such manoeuvres will be a challenge for every organization.

There is strong evidence that human do seem able to detect errors under certain circumstances. With proper and modified behaviour (through goals and incentives), the employee can develop the ability to flag common errors. To make this type of integrated system work effectively, the following (Klein et al., 1997) should be considered:

1. An employee needs to know and be aware that it is part of his/her job to look for and flag suspicious data.
2. An employee should be aware of the different type of errors.
3. The attention of the employee in finding errors is important.
4. An incentive scheme needs to give rewards for finding errors.
5. Management needs to handle false alarms as fast as possible.
6. The organization needs to have a good training and hiring programs in place.

With the implementation of a good training program that can modify users' behaviours (through goals and incentives), he or she can develop the ability to find common errors. Users also can come to understand the causes and potential preventive actions for data integrity challenges. Organizations that use databases can benefit in multiple ways including: the reduction of data errors, employee increase job satisfaction, cost savings, while, at the same time making a positive impact on business outcomes—this is where "quality counts" (Lewis, 1997).

REFERENCES

ArkiData Corporation. (1999). ArkiData Corporation releases next generation of its data auditing and cleansing technology. *The On-Line Executive Journal for Data-Intensive Decision Support, 3*(40). Retrieved from http://www.tgc.com/dsstar/99/1005/101045.html

Bickford, W. (2010, July 9). *Database security issues*. Retrieved from http://www.helium.com/items/446543-database-security-issues

Britt, P. (2007). Tightening security in 2007. *Information Today, 24*(2). Retrieved February 28, 2007, from http://www.infotoday.com/it/feb07/index.shtml

Data. (2011). In *Merriam-Webster.com*. Retrieved from: http://www.merriam-webster.com/dictionary/data

Frey, A. (1997). DBMS backup agents: Because the data matters. Retrieved from http://www.networkcomputing.com/802/802r12.html

Klein, B. D., Goodue, D. L., & Davis, G. B. (1997). Can humans detect errors in data? Impact of base rates, incentives, and goals. *MIS Quarterly, 21*(2), 169-194.

Klusek, L. A. (n.d.). XBRL changes rinancial reporting. *AllBusiness Daily News*. Retrieved October 7, 2011, from: http://www.allbusiness.com/sales/customer-service-product-knowledge/4059101-1.html

Lewis, A. C. (1997). Data integrity. *Education Digest, 62*(7), 72.

Mesmer, E. (2006, January). Caregroup checks out Symantec database security tool. *Network World.* Retrieved March 6, 2007, from http://www.networkworld.com/news/2006/010906-caregroup.html

Mohamed, A. (2006, December). Oracle users warned of new threats to firms' data. *Computer Weekly*. Retrieved March 6, 2007, from http://business.highbeam.com/411267/article-1G1-155767027/oracle-users-warned-new-threat-firms-data

Murray, M. C. (2010). Database security: What students need to know. *Journal of Information Technology Education: Innovations in Practice, 9,* 61-77. Retrieved from http://jite.org/documents/Vol9/JITEv9IIPp061-077Murray804.pdf

Patterson, D. L. (2007). Data security still at risk. *Communications News 44*(2), 8. Retrieved March 6, 2007, from http://connection.ebscohost.com/c/articles/24036077/data-security-still-risk

Pratt, P., & Adamski, J. (2008). *Concept of database management*, (6th ed.). Boston, MA: Cengage Learning.

Rabitti, F., Woelk, D., & Kim W. (1988). A model of authorization for object-oriented and semantic databases. In J. W. Schmidt, S. Ceri, & M. Missikoff (Eds.), *Proceedings of the International Conference on Extending Database Technology* (pp. 231-250). Venice, Italy: Springer.

Rodgers, S. A., & Rodgers, T. (1997, March 07). Enterprise backup software that keeps your data afloat. *Network Computing*. Retrieved from http://www.networkcomputing.com/805/805f2.html

Vizard, M. (2007). Time to get tough on security threats. *Baseline*. Retrieved March 6, 2007, from http://www.baselinemag.com/c/a/Business-Intelligence/Time-to-Get-Tough-on-Security-Threats/

Yang, L. (2009). Teaching database security and auditing. In S. Fitzgerald, M. Guzdial, G. Lewandowski, & S. Wolfman (Eds.), *Proceeding of the 40th ACM Technological Symposium on Computer Science Education*, (pp. 241-245). Chattanooga, TN, USA. doi: 10.1145/1508865.1508954