Safe, Realistic Test Data: The Case for RowGen

White Paper

By Tom Jesionowski

Prime Data Consulting

March 2008

RowGen Version 2
Safe Test Data in Real Formats

Safe, Realistic Test Data -- White Paper

Contents

Introduction

Government regulations, the potential for devastating litigation and fines, along with the negative publicity that follows, are forever changing the way IT professionals should handle test and production data. In addition to traditional concerns such as deadlines, budgets, and untested conditions, technology managers must now address regulatory issues and prevent data breaches. The new regulations have, in essence, added requirements and deliverables to IT projects -- requirements that are not optional.

Combine these concerns with data volume growth, and new challenges begin to emerge related to the creation and management of test data. Privacy concerns from GLBA, HIPAA, et al now inhibit the use of production data as a practical source of test data. The size and scale of modern data processing environments make past methods of creating test data increasingly expensive, time-consuming, and risky. Besides, development teams have more valuable tasks to perform than populating test data tables, files, or reports.

These forces are driving the requirement for more cost-effective methods of high volume test data generation that provide referential integrity, contextual accuracy, and protection against data breaches.

Problem Statements

Rules for test data have changed

Creating test data used to be a simple matter of replicating the production data, performing any necessary edits, and starting to test. This was before privacy laws and the outsourcing of testing to external vendors. The proliferation of production data was not a concern requiring management attention.

In one approach, developers start with production data and wipe out the customer or patient identifying data, but many of these fields are used as keys to link and join the data. The lack of referential integrity renders the test data unsuitable for testing. A method is needed to create test data with field-level security, while keeping the test data structurally and contextually sound.

Local solutions are cumbersome and ineffective – exposure to added risk during development and testing

With the new requirement to build secure and structurally sound test data, project plans must be revisited. Developers, DBAs, and modelers must carve out time to team up and produce this data. The timeline gets pushed out, and promised deadlines are jeopardized. In addition, the following questions now arise: How to test the test data? What assurances exist that the test data is free from flaws that could adversely impact the project? These unknowns drive up the project risk and contribute to slip in the project schedule.

Problem Elaborations

Proliferation of sensitive data carries risk of uncontrolled disclosure

Security becomes another issue to manage when developers choose to copy production data to use as test data. This approach proliferates sensitive data in multiple, potentially uncontrolled locations. When it is used in the test environment, exposure of the data is expanded. In many documented instances, developers and testers moved data to local storage on notebooks, only to have the notebook stolen along with the sensitive data. News headlines routinely report on the data disclosures that resulted from use of this method.

Outsourcing of development and testing may require transport of sensitive data beyond company firewalls

While outsourcing promises to solve some problems, it may actually create new ones. For example, to get a vendor started, they must be provided with either the data model or actual production data. With the data model, they populate the tables with dummy data, but making it contextually correct and structurally sound requires a deep understanding of their customer's business. That level of understanding is problematic at many levels, including tight timelines, budgets, and goals.

If managers opt to give the vendor a copy of production data, they will need to resolve the myriad security questions. How is the data protected while in transit between the confines of a firewalled, secure environment and the vendor's environment? Once in the vendor's hands, can they provide audit controls to track access to the data, and did the project plan accommodate time and money for audit review?

Risk of not being adequately tested in a scaled environment - Need for test data simulation based on both size and structure

So far, we have considered the issues surrounding sensitive data proliferation and test data quality, yet the issue of scale remains. This cannot be satisfied by merely replicating rows. The data needs to be representative of the proposed production environment in size, structure, and distribution of the data. Doing so helps ensure the application or solution being developed will scale by testing against suitable volumes of data.

New requirements need new solutions

While many specifications address the new functionality desired from the project, very few adequately express all of these new business requirements. It falls on the development managers, program managers, and architects to see that these requirements are addressed and met in a timely and cost-effective fashion. RowGen can help them address all of these requirements in a single solution.

New requirements satisfied by RowGen:

- Eliminate or obscure sensitive data in the test environment
- Test against data sets that fully replicate production structures and distributions
- Test across a complete and relevant range of values
- Test against data sets that address anticipated increases in volume
- Preserve the relationships between database tables to simulate production
- Create initial test data where examples do not yet exist
- Allow test data creation external to, and in parallel with, application development

Past Solutions

As the problems associated with regulations and data expansion emerged, teams all over the globe began to address these problems locally. There were some very creative -- and some very costly -- solutions. As with all engineering problems, multiple solutions appear, but each will be lacking in one respect or another. Here are a few to consider:

Using production data – disclosure risk

As mentioned earlier, this solution was favored in the past, but now is often unacceptable due to the risk of data breaches. The best way to protect production data from disclosure is to keep it secure in the transaction or analytics environment, protected by virtual and physical layers of security. Whenever data is moved from behind these hardened security environments, the risk of a breach increases dramatically.

Creating obfuscated data - time consuming, low quality, higher risk of compounding errors

Many teams create scripts and other methods to obfuscate, anonymize, create ambiguity, or encrypt private data. The choices here are two-fold: have the developers build the method, or buy an application to do it for them. Using the developers for this work pulls them away from completing critical tasks on the primary software application, and shifts their focus to an area that may be outside their primary expertise. Project managers are faced with the question of how to test the test data. Project teams developing their own test data invite the possibility of error, adding risk to the project.

Adding a new single-purpose application to secure and unsecure data adds to project overhead and cuts into processing time. A better solution involves adding software like RowGen to eliminate field-level security risks while performing other high-value tasks simultaneously.

Simply stated, there are no methods currently available, other than RowGen, that easily and rapidly leverage existing data models and the business value embedded in the metadata.

Developing and testing with a smaller sample of data obscures problems that occur with larger data sets – scaling problems remain hidden

Many teams turn to simply working with smaller data sets. However, this does have pros and cons. On the upside, since test system performance will typically improve, testers can work through their test cases and run scripts at a faster pace.

The downside involves the hidden problems that will appear only when the data scales. For example, timeouts that are adequate for a 1/10th scale data set may fail when exposed to a fully populated data volume. There is no substitute for working with a test environment that approximates the production environment as closely as possible.

Introducing RowGen as a New Solution

One solution addresses multiple problems

RowGen was developed by Innovative Routines International (The CoSort Company) as the solution to these new and emerging business requirements. RowGen developers have addressed scale, referential integrity, data security, and test data realism within a single solution. Capable of reading a wide range of data sources, it is proving a challenge to find a data model that RowGen cannot leverage.

RowGen is able to build structurally and referentially correct test data for databases according to standard data models. This is achieved through technology integration with RapidACE, LLC (founded by VLDB and data integration expert Dan Linstedt). Through the RapidACE-RowGen graphical user interface (RA-RowGen GUI), both database table and flat file test data sets can be specified for RowGen execution. Through the GUI, RowGen can create test data for:

- Oracle
- Microsoft SQL Server
- Teradata
- DB2 UDB
- Sybase

RowGen also leverages the data movement and manipulation power of CoSort to generate high volumes of realistic test files that can also undergo transformations during creation.

Additional metadata support

Since test data that reflects the appearance of production data is more useful, it is important for the test data solution to use existing data layouts. In addition to using RapidACE technology to build data according to DB models, RowGen has converters for:

- COBOL Copybooks
- CSV file headers
- SQL*Loader Control files
- W3C ELF web logs

RowGen is also compatible with the Meta Integration Model Bridget (MIMB) from Meta Integration Technology (MITI), Incorporated. MIMB can automatically generate RowGen test data definitions from the metadata in any application that MIMB supports. This enables MIMB and RowGen users to more quickly and accurately build test data that matches the structure of their real data, enabling users to further leverage assets they already own.

Viewed collectively, RowGen's broad metadata support extends its use in generating test data for multiple generations of database, ETL, BI, and other data management tools. Conversely, Since RowGen extends the metadata content of existing models and data stores for the purpose of test data generation; it increases the value of metadata content in existing databases.

RowGen Application Scenarios - Case Studies

In the real-world scenarios outlined here, the benefits realized by RowGen are two-fold: cost controls and risk reduction. Cost controls are acquired through gains in the efficiency of operations and process repeatability. This is covered in the first scenario. The second and third scenarios show how RowGen helps to effectively contain test data risk by eliminating the need to replicate sensitive data content to support testing, whether done in-house or by an outsourced development team.

New applications being developed – data needed for user acceptance testing and benchmarking

In one scenario encountered by a RowGen user, a high volume of test data was needed to support a new application. In this instance, there simply was no production data to draw from for testing. With the large volume needed to test the boundaries of operations, the program manager estimated that the project would require roughly 120 hours of development time to build the test data set.

Using a conservative estimate of $100 per development hour, he was looking at an unplanned cost of $12,000 against the fixed-price project budget. This is on top of the delay incurred by reallocating a developer who was providing code for the primary project deliverables.

By bringing in RowGen, connecting to the model repository, setting up and running a script, the manager realized the following benefits:

- 114 hours development hours saved which equates to $11,400 in cost avoided
- Nearly three (3) weeks of project schedule reclaimed
- A repeatable process to create test data that will follow future data model revisions
- A method to generate new test data for future releases to the application being developed
- Error-free test data
- Reduced project risk
- High initial ROI for the RowGen application

Sensitive data needed for testing a GLBA / HIPAA - compliant system

A common practice among testers involves replication of the production environment for testing. This method more accurately mimics the production system for testing, and provides assurances that the test data will have referential integrity.

With changes in regulatory controls, this practice also carries additional resource burdens to safeguard the data from disclosure. The additional steps required to log access to test data, maintain an audit trail, plus the human and technical resources needed to conduct those safeguards, increase the costs associated with the delivery of new releases. This holds true whether the testing is done in-house, or outsourced.

RowGen provides a cost-effective alternative to the practice of production data replication. This is accomplished by providing data content that requires no additional security measures, yet possesses the referential integrity and volumes needed for complete testing of large-scale data applications.

In instances where testing is done in-house, no added controls are required in the test environment to protect sensitive data from a breach. And, there are no additional resources required to operate, and monitor the security software. Typically the staffing to monitor the test environment requires a partial resource, and in some cases, one full-time employee. The added benefit in this use-case is that the test platform now provides an environment to test security applications, while maintaining segregation of access.

When outsourced, the benefits of RowGen are initially realized with the negotiation of the outsourcing services contract. In the absence of the transfer of sensitive production data, any and all sections of the contract related to protecting private and sensitive product data can be simplified, thus simplifying the statement of work-reducing risks for both parties.

For the organization outsourcing the work, the benefit goes beyond reduced risk of data disclosure. Without the need for additional handling controls over the test data, the cost of the work can be reduced, further extending the cost savings associated with outsourcing.

Looking at the situation from the vendor's perspective, using RowGen creates a competitive advantage. The vendor can offer guarantees against data disclosure because no production data changes hands. They can complete the same testing tasks without the need to exchange production data and maintain an audit trail to satisfy regulators. Using RowGen, vendors can offer complete GLBA and HIPAA compliance in their testing environment.

Safe, Referentially Correct Data Needed for Integration Testing

While working on a systems integration project, a technology lifecycle consultancy called FlexITy Solutions required massive quantities of referentially correct test data. The project required the operation of a transaction database and the transmission of query results to validate the systems being installed. FlexITy's client, a financial institution, could not turn over any production data for testing due to regulatory and privacy concerns. FlexITy had to produce and incorporate test data that permitted queries against a representative database, and to transmit those results over a long-haul connection. RowGen was the logical fit for these requirements.

When the Solution Architect, Technical Lead, and Project Manager confronted the problem, they considered the same alternatives presented here. They estimated that 200 man-hours of development time would be required to build the scripts necessary to populate the tables -- work that would be hard to apply to future projects, given this is only a point solution. The time that could be saved using RowGen would provide a competitive advantage. Also, RowGen would be reusable for future projects. The decision was therefore made to move forward with RowGen.

The server used for the effort held four (4) CPUs and 8GB of RAM. The data model they decided upon contained 16 tables with referential integrity. When the RowGen job launched, the team was impressed when, after four hours, the entire suite of test data was generated. They now had 33 million rows of data spanning 20GB of storage. In just over 16 hours, they saved a potential 184 hours of development time.

The 184 man-hours that were saved were reallocated to other areas. From a financial perspective, those hours represented a significant savings for the project budget. Total savings, with the cost of RowGen factored in, were estimated at $20,000US, considering a fully loaded rate of $150US per hour. Whether applied to the project margin, or reallocated to bolster efforts in other areas, the savings represent a significant benefit.

Ancillary Functional Benefits

RowGen offers a compelling list of benefits that strengthen any business case proposal for the acquisition, purchase, and application of a test data solution.

Breadth, depth, and accuracy in test data generation.

An intuitive data definition syntax, high- performance generation engine, and facilitate the creation of many, large test data sets for application development. This allows other people to create test data independently of the application, which a) saves developer time, b) protects real data, and c) improves application quality because it does not rely on developers who may only create test data that shows their application working.

Bringing application validation to the testing process

While generating test data, RowGen can transform it to mirror the same data filtering and aggregation behavior occurring in the production application. By creating test data with built-in grouping results, customer applications running against the same test data can be validated for their grouping results. If the summary results match across the RowGen output and the application's output, then it is likely the user's application is correct and will also work correctly against production data.

Business Benefits

- Cuts development and testing time/costs
- Realistic volume and range testing = better quality control = lower risk management costs
- Well-tested, high-quality applications increase customer satisfaction/loyalty and decrease support costs
- Helps organizations to comply with data privacy policies, and addresses risks associated with production data
- Better resource utilization – more time spent developing and testing versus preparing data to develop and test

Financial Benefits

- ROI typically realized with the first project
- Resources that are saved can be applied to create more competitive project proposals.
- Reduction of risk associated with potential fines, litigation, and damage to brand image.

Feature Summary

- Generates tables with referential integrity
- Rich graphical user interface
- Reads existing data models using a collection of metadata bridges
- Produces either pristine data sets or data sets with intentional outliers for boundary testing
- Creates multiple targets to support:
    - ETL loads
    - Development
    - Outsourcing
- Operates with multiple RDBM systems including:
    - Oracle
    - Microsoft SQL Server
    - Teradata
    - DB2 UDB
    - Sybase
- Generate test data in sorted order – presorting speeds database loads
- Capable of building full volume test data with unprecedented speed to support scaled testing

Compatible Applications

CoSort

The RowGen control language was created with the same powerful parallel processing engine found in IRI's CoSort product. Job scripts and metadata are interchangeable between RowGen and CoSort. This saves time when moving between real data transformation / formatting and test data generation.

RapidACE

RapidACE is a highly advanced tool for the management and integration of disparate data models. RowGen is now embedded into the full, standalone RapidACE package for data model visualization and consolidation.

Fast Extract (FACT) for Oracle

IRI's FACT product unloads large Oracle tables in parallel, and also creates the data definition file (.DDF) format supported by both CoSort and RowGen. Applying FACT and RowGen together, users can create test data in the same format as the Oracle tables being extracted.

Meta Integration Model Bridge

Meta Integration Technology, Inc. (MITI)'s Meta Integration Model Bridge (MIMB) suite locates and converts the metadata between the most popular design, database, ETL, OLAP, and BI tools. One of the destination metadata formats that MIMB supports is the data definition file (.ddf) format that RowGen uses. Therefore, users of both MIMB and RowGen can automatically build RowGen-ready test data definitions according to the metadata in their existing applications.

Arranging a RowGen Demo

Additional information about RowGen is available at the IRI website; http://www.iri.com/rowgen. Free trials can be downloaded.

Contact Innovative Routines International (or its agents outside the USA) through its website at http://www.iri.com or call them at 1-321-777-8889.