HOW TO PROVISION THE RIGHT TEST DATA FOR BUSINESS EVENTS?

Sunder Nochilur
Sr. Consultant, Wipro Technologies

Table of Contents

Abstract

For any given organization to succeed, irrespective of its size, area of expertise and customer base, an awareness of the end user needs, and adapting to meet those is of utmost importance. In addition adapting to the changes promptly is equally important to achieve competitive advantage.

Test Data Management plays a key role in ensuring quality roll out of applications at the right time. As a service/practice, it has matured in most organizations. The predominant activities include sub-setting, masking, data refresh and synthetic creation standardized through Test Data Management (TDM), Extract Transform Load (ETL) or Database (DB) tools.

However, domain and platform specific test data provisioning still remains a challenge as this needs deep knowledge of the entire business processes. Data preparation for business events across domains is still in its infancy with organizations devising their own methods to address these challenges.
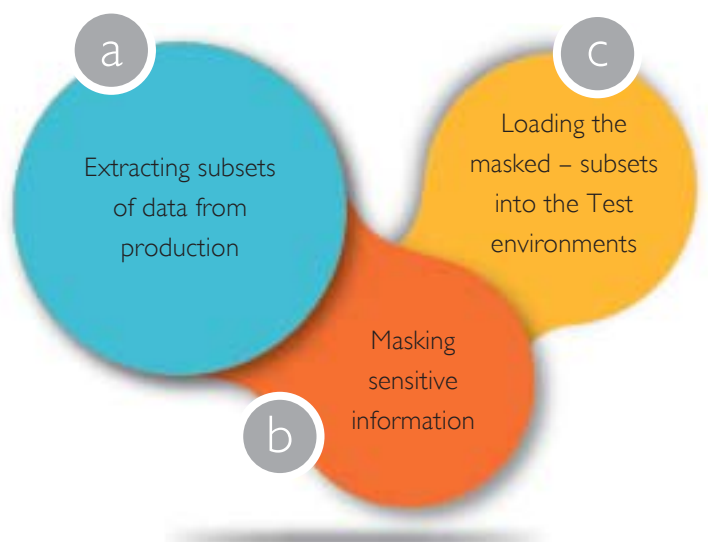
This paper summarizes the various approaches available for effectively preparing Test data sets for business events.

Introduction

In any organization, approximately 40% of a tester's time is spent on waiting for the right kind of test data. To add to it, 20% to 25% of defects found in UAT and production phase are directly or indirectly attributable to the quality of test data used.

Organizations have realized the impact of test data on the three pillars of Cost, Quality and Schedule. Hence, investments are being made in Centralizing Test Data as-a-Service (TDaaS) and automating test data provisioning through implementation of sophisticated Test Data Management (TDM) tools. These tools play a key role in:

a Extracting subsets of data from production

b Masking sensitive information

c Loading the masked – subsets into the Test environments

And all this needs to be achieved while retaining data integrity at an application level across applications.

Data integrity across applications is a critical requirement for testing end-to-end business scenarios. The TDM tools also have capabilities for generating synthetic data for new functionalities where data does not exist in production databases.
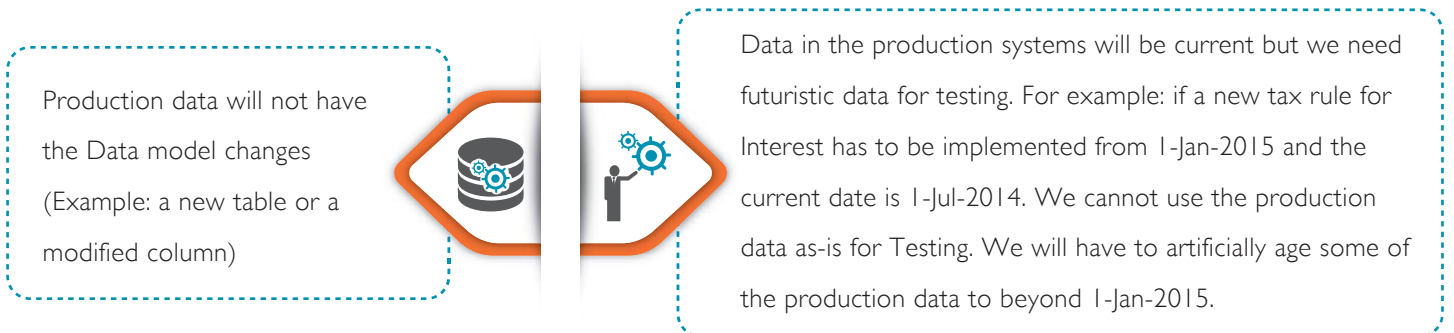
Preparing test data for business events like End of Day (EoD) process in the banking sector or holiday season in the retail domain are complex scenarios where out-of-the-box solutions don't exist. Various dimensions like volume of data, dependencies across applications and time sensitivity adds to the complexity of preparing data for business event testing.

The Challenge

Consider a scenario where the taxation rules for interest earned from bank deposits in US changes from 1-Jan-2015 and we are currently on 1-Jul-2014. In this case, the software changes may be developed in Sep-2014 and the modified programs may be implemented in production by 31-Oct-2014. However, the new tax calculation program will not be actually executed till say, the night of 31-Jan-2015. In order to test this scenario, we will have to simulate the data for 1-Jan-2015 to 31-Jan-2015 in the test environment.

A typical bank will have in excess of 20000 batch jobs for the month-end processing. Now, if a data model change impacting these batch jobs has to be implemented, it is imperative to test the entire month-end process. A copy of data from production cannot be used directly for testing as there will be at least two shortcomings:

Production data will not have the Data model changes (Example: a new table or a modified column)

Data in the production systems will be current but we need futuristic data for testing. For example: if a new tax rule for Interest has to be implemented from 1-Jan-2015 and the current date is 1-Jul-2014. We cannot use the production data as-is for Testing. We will have to artificially age some of the production data to beyond 1-Jan-2015.

Similarly, in the retail industry, let us take the example of a situation where a new piece of code needs to be implemented for calculating and applying discounts to purchases made between the discount period of 25-Nov-2014 and 5-Jan-2015.

This cannot be addressed completely by just advancing the date on the database and application servers. There will be many scenarios that will actually need data that spans before, during and after the discount period like:

1. Product is sold prior to 25-Nov-2014 but returned during discount period

2. Product is sold and returned in the discount period

3. Product is sold in the discount period and it is returned / exchanged after 5-Jan-2015

Hence, organizations have to carefully strategize their approach towards test data management, to ensure the quality of application and deliver excellent end user experience to safeguard their brand image.

Solution Approaches

Let us now focus on the approaches available for provisioning data for Business events like the ones mentioned above.
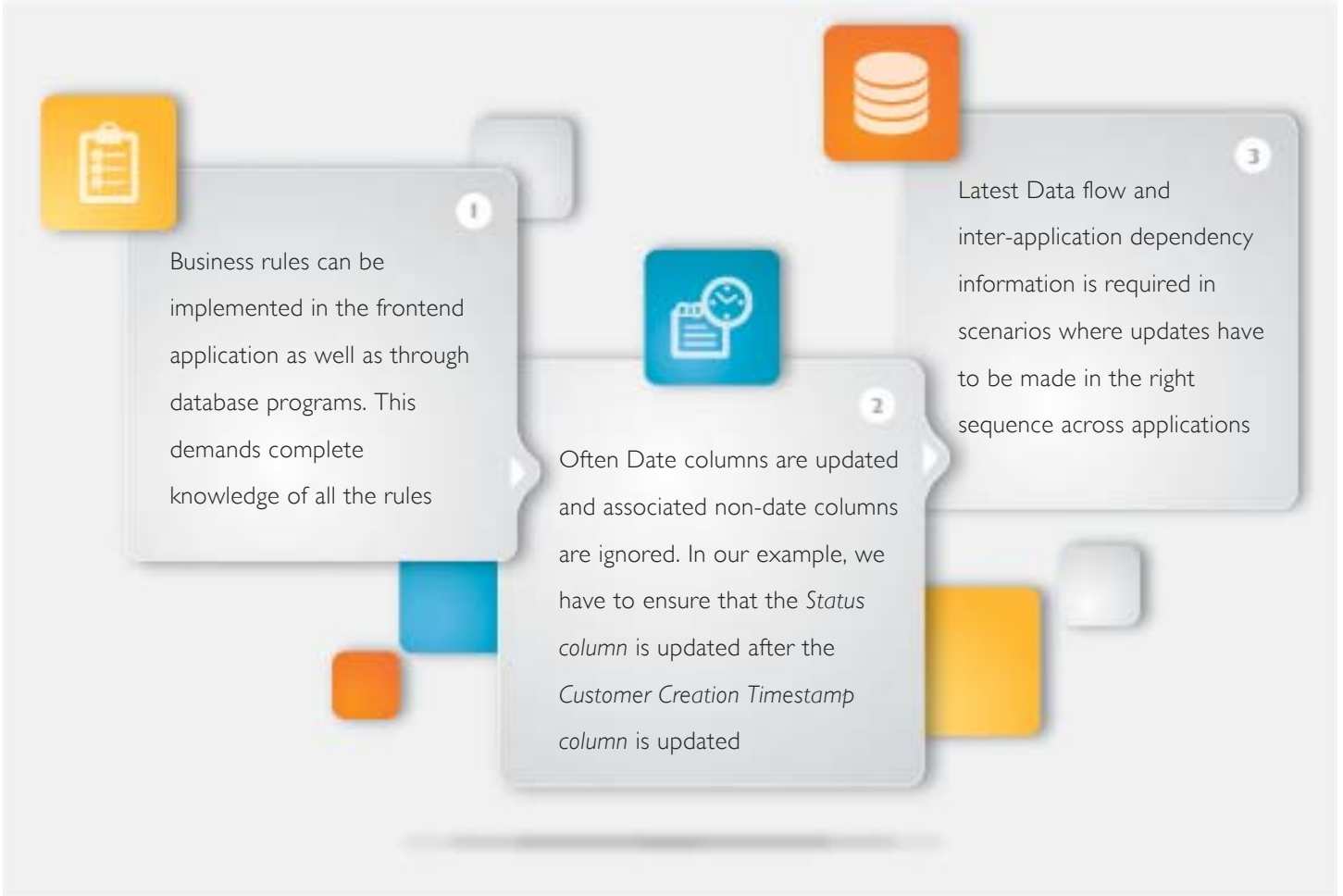
Selective Updates

Scenario 1: Suppose a telecom service provider gets a new customer request and the *Customer status* is initially "New" and the status changes to "Active" only after 48 hrs.

This is a typical scenario where a small volume of data is required for testing the scenarios. For this we can use employee selective updates where the table columns that are impacted are first identified and then updated with values for the records that will be used in that particular test cycle.

So, in this case we can create a new customer from the application frontend with "New" status and then wait for 48

hours for it to change it to "Active" status. Only after that can they further test cases associated with this customer. Alternatively we can also update the "Customer creation Timestamp" column for the new customer to 48 hrs in the past and then run the "Customer Status Update" program. This will set the customer status to "Active".

The latter approach is most commonly used by organization as it is quick and least disruptive. But the following factors need to be considered while employing this approach.

1. Business rules can be implemented in the frontend application as well as through database programs. This demands complete knowledge of all the rules

2. Often Date columns are updated and associated non-date columns are ignored. In our example, we have to ensure that the *Status column* is updated after the *Customer Creation Timestamp column* is updated

3. Latest Data flow and inter-application dependency information is required in scenarios where updates have to be made in the right sequence across applications

Mass Update

Scenario 2: An organization wants to simulate entire data in their Test environment from 31-Jan-2015 while current date is 1-Jul-2014.

For this, data backup from production is loaded into the test environment and all date columns are advanced by a preset number of days.

The process that would be needed to be followed will include:

1. Preparation of detailed Data flow diagram across applications

2. Extracting production data for the last 'n' months (n can be 1, 3, 6 or 12, based on the project need) for all the applications as per the data flow

3. As the current date in the scenario is 1-July-2014, we need to advance the dates by 7 calendar months to get data for 31-Dec-2015

4. Data has to be loaded for applications in the test environment in the same order in which it was extracted

5. Disable all Integrity constraints in the databases

6. Identify all date columns and update all date columns by 7 months using the add months function of the database

7. Identify all non-date dependent columns and update them if necessary

8. Enable all Integrity constraints in the database

    a. Resolve issues identified while enabling constraints

    b. Document the issues encountered and the solution for future use

9. Retain a backup of this prepared data before opening up the test environment to the Test teams
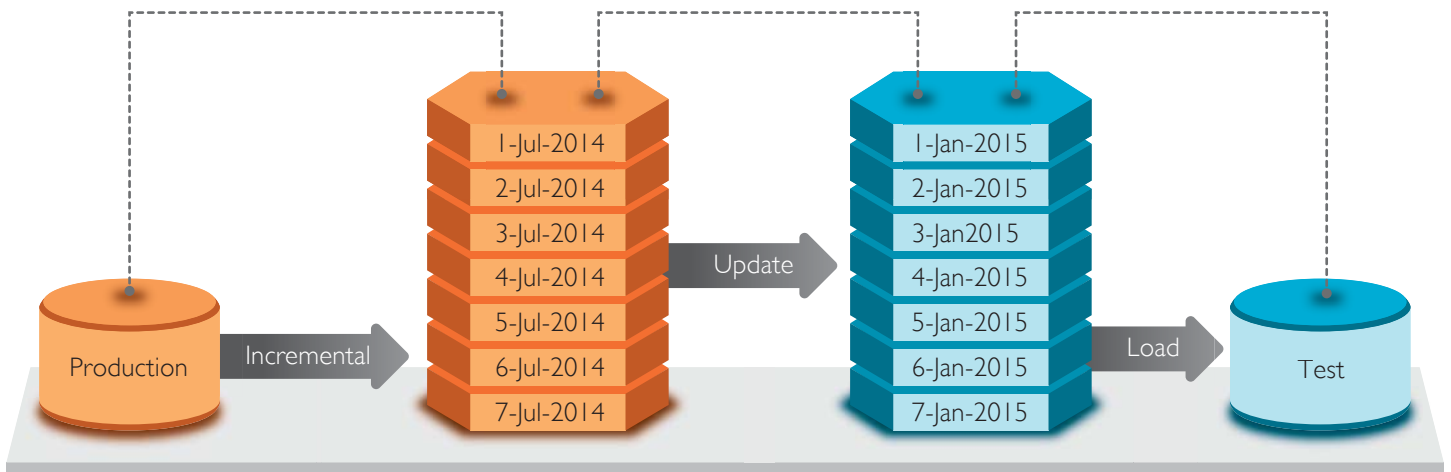
This approach should be treated like a software development project. The complete process has to be designed developed and tested. It has to be maintained on an ongoing basis to take care of:

- Changes to Data model in production

- Changes to Data flow within and across applications

- New / updated dependencies across column values

- New / updated integrity constraints

Mass Update with Look up

One of the drawbacks of the Mass Update approach is that when the data is updated by a fixed number of days, the date in production may map to a wrong day in the Test environment.

Scenario 3: 1-Jul-2014 is a Friday. If it is advanced by 7 months to 1-Jan-015 it will be a Thursday. However, 1-Jan-2015 is a holiday where 1-Jul-2014 is a working day. If it is desired that data of a weekday is mapped to a week day, that of a weekend is mapped to a weekend and data of a Holiday is mapped to a Holiday, a lookup table will have to be used. The lookup table will contain three columns at a minimum. The table below shows sample data for a look up table:

Source	Target	Type	Day
1-Jul-14	2-Jan-15	Weekday	Thursday
2-Jul-14	3-Jan-15	Weekday	Friday
3-Jul-14	4-Jan-15	Weekend	Saturday
4-Jul-14	5-Jan-15	Weekend	Sunday
5-Jul-14	6-Jan-15	Weekday	Monday
6-Jul-14	7-Jan-15	Weekday	Tuesday
7-Jul-14	8-Jan-15	Weekday	Wednesday
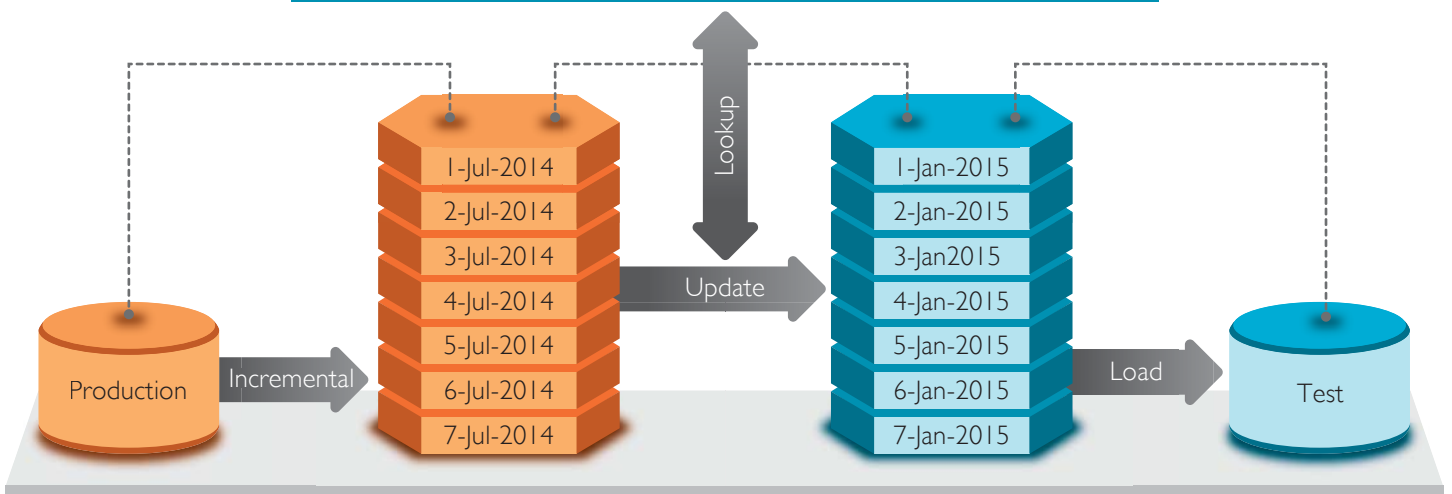
The steps involved in this process are very similar to the Mass update approach. Only the Update step will be different, wherein the lookup table will be used instead of a formula.
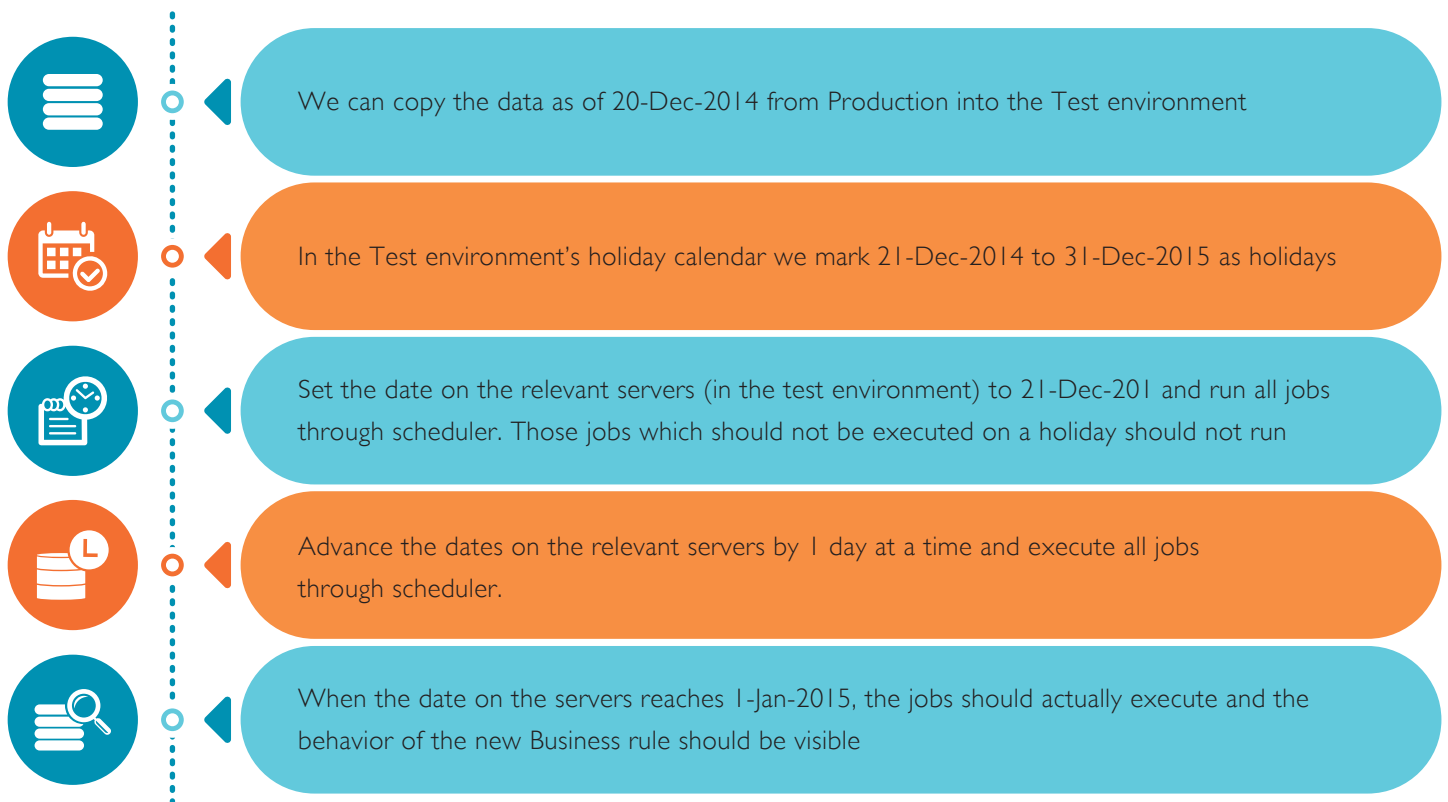
This approach gives the flexibility of generating data for specific dates based on the project needs. If we have to simulate a scenario where sales on an e-Commerce application shoots up by 400%. All we have to do is map 4 days in the Source to 1 day in the Target. Also, by using a single look up table across application, we can easily ensure data integrity is maintained.

Holiday Calendar

Many organizations maintain a Holiday calendar which can be accessed by one or more applications. This Holiday calendar is used by the scheduler to determine if a particular job needs to be run on that date or not. If this feature is available in an environment, the Holiday calendar can be used for ageing data sets.

Scenario 4: Suppose a new business rule comes into effect from 1-Jan-2015 and we need to test this business rule on 21-Dec-2014.

We can copy the data as of 20-Dec-2014 from Production into the Test environment

In the Test environment's holiday calendar we mark 21-Dec-2014 to 31-Dec-2015 as holidays

Set the date on the relevant servers (in the test environment) to 21-Dec-201 and run all jobs through scheduler. Those jobs which should not be executed on a holiday should not run

Advance the dates on the relevant servers by 1 day at a time and execute all jobs through scheduler.

When the date on the servers reaches 1-Jan-2015, the jobs should actually execute and the behavior of the new Business rule should be visible

This approach has a dependency on Holiday Calendar being a part of the application design. This is generally applicable in the Banking domain. This is effective if data has to be aged by a few days and might not work if data has to be aged by a month or more.
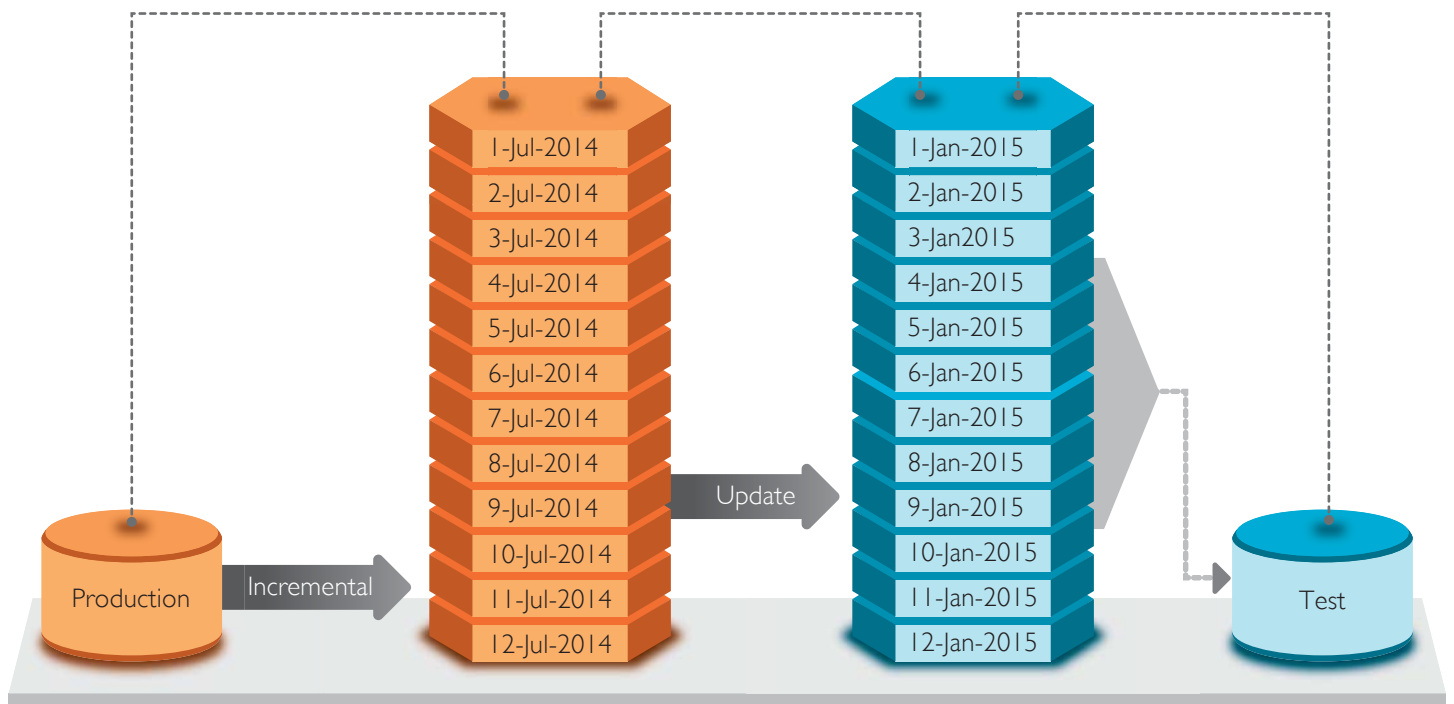
Incremental Data Sets

The Mass update and Mass Update with look up approaches require that a separate data set is created and maintained for each business cycle (daily, weekly, monthly and so on). Maintaining multiple data sets could be painstaking if there are too many cycles. And, if the data is not masked, there could be potential data security risks with multiple data sets. In the Incremental data sets approach, data is extracted at a lower granularity level from production and updated separately for each incremental set.

Example 1: In an e-Commerce environment, data can be extracted for every hour. Each hourly data set is updated (using the steps mentioned in the mass update approach) and data can be provisioned in the Test environment using combinations of hourly data sets.

Example 2: Suppose we need to prepare data for a month-end job in Jan-2015. We extract data from production on a daily basis from 1-Jul-2014 to 31-Jul-2014. Each day's data is updated as per our need (following steps mentioned in the Mass Update approach). All the 31 data sets will be loaded sequentially into the Test environment. Suppose we need to provision data for an end of week job, we can identify 7 days' worth of data from our data repository and load them.

Production	Incremental		Update			Test
	1-Jul-2014			1-Jan-2015		
	2-Jul-2014			2-Jan-2015		
	3-Jul-2014			3-Jan2015		
	4-Jul-2014			4-Jan-2015		
	5-Jul-2014			5-Jan-2015		
	6-Jul-2014			6-Jan-2015		
	7-Jul-2014			7-Jan-2015		
	8-Jul-2014			8-Jan-2015		
	9-Jul-2014			9-Jan-2015		
	10-Jul-2014			10-Jan-2015		
	11-Jul-2014			11-Jan-2015		
	12-Jul-2014			12-Jan-2015		

Conclusion

To sum up, while Test Data Management tools and processes have matured in the last few years, there are still many areas across domains where processes are still not standardized. Hence, Businesses are not deriving the maximum benefits that a robust TDM function can deliver. Multiple approaches are available to overcome this and the right one for a given scenario depends on the quality, quantity and quickness with which data is required. In some complex situations, a combination of these approaches can also be adopted to solve the Test data challenge. Implementing a standardized Test Data solution for business events can significantly reduce risk, improve quality and reduce cost for Businesses.

About the Author

Sunder Nochilur is a Sr. Consultant at Wipro Technologies. He has over 17 years of experience in Information Management and Business Intelligence. He has conceptualized, designed, developed, tested and implemented Data warehouses and BI applications for Automotive, Fast Moving Consumer Goods (FMCG), Insurance and Energy customers across geographies. He has defined and implemented end-to-end TDM processes at enterprise level for Media, Retail and Banking customers.

About Wipro Ltd.

Wipro Ltd. (NYSE:WIT) is a leading Information Technology, Consulting and Business Process Services company that delivers solutions to enable its clients do business better. Wipro delivers winning business outcomes through its deep industry experience and a 360 degree view of "Business through Technology" - helping clients create successful and adaptive businesses. A company recognized globally for its comprehensive portfolio of services, a practitioner's approach to delivering innovation, and an organization wide commitment to sustainability, Wipro has a workforce of over 140,000, serving clients in 175+ cities across 6 continents.

For more information, please visit www.wipro.com

WIPRO
Applying Thought
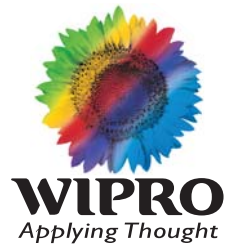
DO BUSINESS BETTER

CONSULTING | SYSTEM INTEGRATION | BUSINESS PROCESS SERVICES

WIPRO LIMITED, DODDAKANNELLI, SARJAPUR ROAD, BANGALORE - 560 035, INDIA. TEL : +91 (80) 2844 0011, FAX : +91 (80) 2844 0256, Email: info@wipro.com

North America Canada Brazil Mexico Argentina United Kingdom Germany France Switzerland Nordic Region Poland Austria Benelux Portugal Romania Africa Middle East India China Japan Philippines Singapore Malaysia South Korea Australia New Zealand