# GenRocket

# **GENROCKET CASE STUDY**

Data Refresh: Part of a Seamless Transition to Test Data Generation

### BACKGROUND

As QA departments introduce Test Data Generation (TDG) into their testing environment, they will need to support legacy Test Data Management (TDM) functions that are in place to manage their test data. One of these functions, data refresh, is necessary to keep test data current so it reflects any changes that have been made to the production database.

With GenRocket, data refresh becomes easy and automated with the help of data model driven test data generation. The approach is straight forward: Import the current data model for the given application and then generate (refresh) the data any time, even when there are changes to the data model.

# THE TECHNICAL CHALLENGE

The QA team at a major insurance company needed to insert new data into their test database as part of a data refresh process. They were looking to add synthetic test data into an existing table without changing the existing data. The test data is a MySQL table with 10 columns in which 7 columns already have data and 3 columns need to be populated with data using the GenRocket TDG platform. GenRocket can be configured to automate this function and refresh the test database with real-time synthetic test data.

## THE GENROCKET SOLUTION

3 new columns have to be added to the Address Table using GenRocket. The following ALTER statement can be used to add the columns:

1	ALTER	TABLE ADDRESS ADD `
2	(	
3		secondary_city VARCHAR2(20 BYTE) DEFAULT 'Tennessee',
4		secondary_state_code VARCHAR2(2 BYTE),
5		secondary_mobile_number VARCHAR2(15 BYTE)
6	);	

Then the test data is generated using GenRocket and populated in the 3 new columns without changing existing data in the Address Table. The following UPDATE statement can be used to populate the data to the columns.

1	UPDATE GRP.address set
2	<pre>secondary_city = 'Dover',</pre>
3	<pre>secondary_state_code = 'DE',</pre>
4	secondary_mobile_number = '(200) 954 7691'

### HOW IT WORKS



Domains can be created in GenRocket by using the "Import from DDL" option.

A new Domain is created and assigns the SQLUpdateReceiver as the Domain Receiver.

The SQLUpdateReceiver connects to a database and performs batched updates. The update statement is defined on the Receiver's data tab.

Attributes whose values will be used in the update statement are assigned a variable from var1 to var20.

The Alter statement can be added in the Data tab of the SQLUpdateReceiver as shown in the image below:

Name	Logging		Parameters	Data	Attributes Property Keys	
Add Receiver Add Receiver Attributes for SQLUpdateReceived Id	Ver Global Edit	2410	data	/	alter table GRP.ADDRESS add PREVIEWS_CITY varchar(20), add PREVIEWS_ST_CD varchar(2), add PREVIEWS_CELL_NBR varchar(15)	0
			Save			

Another Domain is created in GenRocket with the following Attributes for generating the data

Column Name	Domain	Generator Used
secondary_city	previewsCity	USStateCapitalGen
secondary_state_code	previewsStCd	USStateAbbrevGen
secondary_mobile_number	previewsCellNbr	FlexiblePhoneNumberGen

By using SQLUpdateReceiver again we can use the update statement to update the values.

Name	Logging	2010	Parameters	Data	Attributes Property Keys	
Add Receiver Attributes for SQLUpdateReceiver Id	Global Edit	/ 10	data	_	update GRP.address set secondary_city = 'var2', secondary_state_code = 'var3', secondary_mobile_number = 'var4' where address_number = var1;	0
previewsStCd						4
previewsCellNbr			Save			

As you can see, we are using var2 var3 and var4 and the condition is added. These variables are linked with the Attributes in the 'Attribute Property Keys` tab of the SQLUpdateReceiver.

		1010	Parameters	Data Attributes Property Keys		
SucupdateReceiver		2.10	Antonio			
			Autorio Grie Harr			
			S.No	*attribute 🚺	variable 🚺	order 🚺
			1	id	var1	1
			2	previewsCity	var2	1
AU30 PORCEIVER			3	previewsStCd	var3	1
Attributes for SQLUpdateRecei	Ver Global Edit		4	previewsCellNbr	var4	I
id						
previewsCity						
previewsStCd						

By using a Scenario Chain, we can run two Scenarios to Alter and Update the table simultaneously.

cenario Cl	ain: AddressChangeScenarioChain		Constant By Dropp States : care Manifest By Dropp B
Name:	AddressChangeScenarioChain	Description:	
		Save Download Copy Categorize Delete	
Name		Delay Secs After	
AlterAddre	ssScenario	0 /	01
AddressNe	wColScenario	01	01

#### Here's a screenshot of the table before populating the data

last_name	secondry	_name_tit	le   street_address	po_box	city	state_code	country	zip_code	date_created	last_updated	mobile_area	code   mobile_number	i secondry_middle_name	+
2		100	Miss	Maud			Berge		Miss	Miss		Maud		Berge
1 2	Miss	101	1 Washington Dr Mr	PO Box 1   Major	Olympia   Z	I WA	US     Guillot	98501	2019-03-22	2018-08-01		345   216-6341     Major	1.7	Guillot
	Mr		2 Adams Sq	PO Box 2	Nashville		US	37201	2019-02-05	2018-08-01		944   445-9192		
1 2	L Me	102	Mr   3 lefferson (t	Cody	S	1 ME	Pemberto	04330	Mr 1 2019-11-01	Mr   2018-08-01		Cody 567 1 663 3594 1	15	Pemberto
	1	103	Rev	Beverly	Y	1.7%	Fetzer	04330	l Rev	Rev		Beverly		Fetzer
	Rev		4 Madison St	PO Box 4	Honolulu		US	96801	2019-10-21	2018-08-01		812   117-3600		
1 3	Mrs	104	Mrs   5 Monroe Rd	Veronica   PO Box 5	Atlanta	I GA	Geier	38381	Mrs 2819-18-17	Mrs		Veronica 178   717-5244		Geier
		105	Rev	Denny	M		Brookshi	re	Rev	Rev		Denny	I N	Brookshi
f .	Rev		6 Adams Ave	PO Box 6	Tallahassee		US I	32301	2019-02-17	2018-08-01		581   353-9702		
1 3 n	Me	106	Mr   7 Jackson Blvd	Hong   PO Box 7	Jackson	1 MS	Pitkingt	on 39281	019-02-10	Mr   2018-08-01		Hong 883   211-4460	I K	Pitkingt
		107	Dr	Florentino			Brodeur		Dr	Dr		Florentino		Brodeur
	Dr		8 Van Buren Wy	PO Box B	Carson City	1 NV	US	89701	2019-09-09	2018-08-01		109   147-3797		1. Cabula
	Mr	168 1	Mr   9 Harrison Ln	PO Box 9	Oklahona City	/ 1 OK	i schun Lus I	73101	Mr   2819-87-86	2018-08-01		765   743-9849	i c	i schun
		109		Elton			Cauthen					Elton		Cauthen
	Dr		10 Tyler Cir	PO Box 10	Frankfort	I KY	US	40601	2019-04-05	2018-08-01		133   386-5591		E Malladan
1 4 5	Dr	110 1	11 Polk Pk	PO Box 11	Chevenne	1 WY		82001	2019-07-10	2018-08-01		187   710-6557	1.9	i vattadar
			Mr	Elisha			Newcomb		Mr	Mr		Elisha		Newcomb
	Mr		12 Taylor Pkwy	PO Box 12	Dover	I DE	US	19901	2019-03-24	2018-08-01		665   168-4159	1.4	1 0.41166
	Ms		13 Fillmore Dr	PO Box 13	Montgomery	1 AL		36101	2019-03-14	2018-08-01		479   757-1812	1 A.	i Mattin
			Rev	Chandra	l w		Lalibert		Rev	Rev		Chandra	l w	Lalibert
	Rev		14 Pierce Sq	PO Box 14	Carson City	1 NV	US	89701	2019-06-13	2018-08-01		838   120-7864	1.0	L Corcoran
	Mr	***	15 Buchanan Ct	PO Box 15	Denver		US I	80012	2019-03-21	2018-08-01		364   559-7503		i corcoran
			Mrs	Amy			Coronado		Mrs	Mrs		Amy		Coronado
1 6	Mrs	116 1	16 Lincoln St	PO Box 16	Boston	1 MA	Amerson	02108	2019-02-01	2018-08-01		135   753-9815     Berenice	Le	LAmerson
	l Mrs		17 Johnson Rd	PO Box 17	Washington	DC	US	20001	2019-06-23	2018-08-01		850   118-3145		
			Mrs	Cecile			Maki		Mrs	Mrs		Cecile		Maki
1 7	Mrs	118	18 Grant Ave	Fmerson	Baton Rouge	I LA	US     Avitia	10801	1 2019-02-04	1 2018-08-01		36/   5/5-5385     Emerson	1.6	Avitia
	i Mr		19 Hayes Blvd	PO Box 19	Boise		US I	83701	2019-06-11	2018-08-01		433   207-1724		
		119	Rev	Marylin	I R		Tolman		Rev	Rev		Marylin		Tolman
	Rev		20 Garfield Wy	PO Box 20	Des Moines	1 IA	i us i i	50301	2019-06-19	2018-08-01		821   648-3858		

divyasetia@htaylor: - My SQL Shell

#### Here's a screenshot of the table after running the GenRocket Scenario Chain

ysql> ysql> ysql>	sele	ct • from a	ddres	s limi	it 10;													
user_ ast_na ode	id   me   seco	address_nu secondry_n ndary_mobil	mber ane_t e_num	prim itle   ber	ary_name_prefix   street_address	primary_f   po_box	irst_name   p   city	rimary_mi   si	iddle_name tate_code	primary   country	_last_name   zip_code	primary_name_   date_created	_title   second   last_updated	ry_name_prefix   mobile_area_	secondry_first_name code   mobile_number	secondry_middl   secondary_city	e_name   secon	secondry dary_state
	2	Miss	100	Miss	: 1 Washington Dr	Maud   PO Box 1	V   Olympia	l w	A.	Berge   US	98501	Miss   2019-03-22	Miss   2018-08-01		Maud 345   216-6341	V   NULL	1 NULL	Berge
	2     NULL	Mr			2 Adams Sq	Major   PO Box 2	Z   Nashville	1 11	N	Guillot   US	37201	Mr   2019-02-05	Mr   2018-08-01		Major 944   445-9192	1 Z   NULL	NULL	Guillot
	2     NULL		182	Mr   	3 Jefferson Ct	Cody   PO Box 3	S   Augusta	1 M		Pembert   US	on   84338	Mr   2019-11-01	Mr   2018-08-01		Cody 507   663-3594	S   NULL	I NULL	Pemberto
	3     NULL	Rev		l Rev I	4 Madison St	Beverly   PO Box 4	Y   Honolulu		r	Fetzer   US	96801	Rev   2019-10-21	Rev   2018-08-01		Beverly 812   117-3600	I Y I NULL	I NULL	Fetzer
	3     NULL	Mrs	184	Mrs   	5 Monroe Rd	Veronica   PO Box 5	)   Atlanta	16	A.	Geier   US	30301	Mrs   2019-10-17	Mrs   2018-08-01		Veronica 178   717-5244	J   NULL	I NULL	Geier
	3   NULL	Rev		Rev   	6 Adams Ave	Denny   PO Box 6	M i   Tallahass	ee   Fi		Brooksh   US	ire   32301	Rev   2019-02-17	Rev   2018-08-01		Denny 581   353-9702	I M I NULL	I NULL	Brookshi
	3   NULL	Mr	106	Mr 	7 Jackson Blvd	Hong   PO Box 7	Jackson	M2		Pilking   US	ton   39201	Mr   2019-02-10	Mr   2018-08-01		Hong 863   211-4460	I K I NULL	NULL	Pilkingt
	NULL		107		8 Van Buren Wy	Florentin   PO Box 8	io   A     Carson Ci	ty  N		Brodeur   US	89701	Dr   2019-09-09	Dr   2018-08-01		Florentino 109   147-3797	I A I NULL	I NULL	Brodeur
	NULL	Mr	108		9 Harrison Ln	PO Box 9	)   Oklahoma	City   O		US	73101	MP   2019-07-06	Mr   2018-08-01		Willie 765   743-9849	I NULL	I NULL	Schuthen
	NULL		109	   	10 Tyler Cir	PO Box 1	8   Frankfort			US	40601	2019-04-05	2018-08-01		133   386-5591	( พบัน	NULL	- cauthen

rows in set (0.00 sec)

sql> 🛛

# IMPACT

By performing a data refresh with GenRocket's TDG platform, testers can keep their test database up to date as they combine real-time synthetic test data with their existing production test data.

Real-time synthetic test data brings a higher level of quality and efficiency to their test operation:



### TEST DATA SPEED

Provisioning test data on-demand at the rate of 1000's of rows/second



### TEST DATA QUALITY

Precise control of the data subset with patterned and conditioned data



### TEST DATA SECURITY

Eliminating personally identifiable information (PII) from the data



### COST AND SIMPLICITY

Reducing the dependency on costly and cumbersome TDM systems

With GenRocket, data refresh becomes a function that bridges the gap between the TDM model of managing and archiving a master copy of the production database and the TDG model in which test data is generated whenever it's needed and discarded when it's not. This allows the two worlds to gracefully coexist while the testing organization makes a transition to the more efficient approach of continuous testing with real-time synthetic test data.



If you would like to know more about GenRocket's Test Data Generation platform and our industry solutions, please visit our website at <u>www.genrocket.com</u>.