



# 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

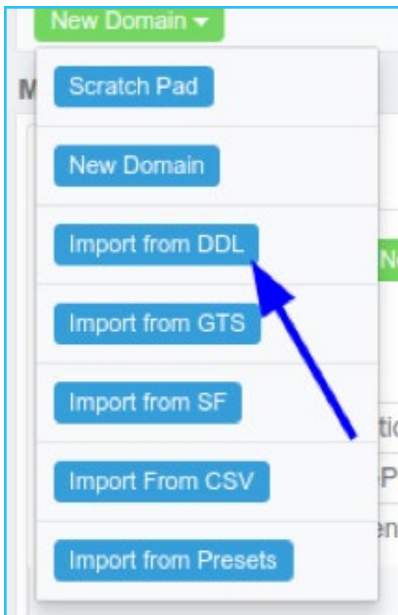
3 new columns have to be added to the Address Table using GenRocket. The following ALTER statement can be used to add the columns:

```
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   secondary_city = 'Dover',
3   secondary_state_code = 'DE',
4   secondary_mobile_number = '(200) 954 7691'
```

## HOW IT WORKS



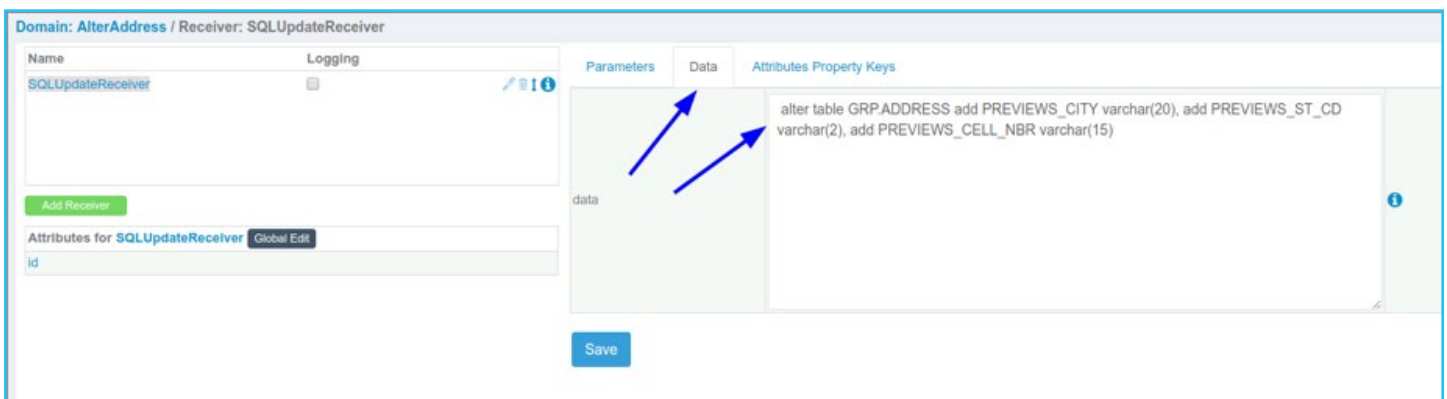
Domains can be created in GenRocket by using the “Import from DDL” option.

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:



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.

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.

S.No	*attribute	variable	order
1	id	var1	1
2	previewsCity	var2	1
3	previewsStCd	var3	1
4	previewsCellNbr	var4	1

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

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

```
mysql>
mysql> select * from address limit 20;
```

user_id	address_number	primary_name_prefix	primary_first_name	primary_middle_name	primary_last_name	primary_name_title	secondry_name_prefix	secondry_first_name	secondry_middle_name	secondry								
last_name	secondry_name_title	street_address	po_box	city	state_code	country	zip_code	date_created	last_updated	mobile_area_code	mobile_number							
2	Miss	100	Miss	1	Washington Dr	PO Box 1	Olympia	WA	US	98501	2019-03-22	2018-08-01	Maud	Miss	345	216-6341	V	Berge
2	Mr	101	Mr	2	Adams Sq	PO Box 2	Nashville	TN	US	37201	2019-02-05	2018-08-01	Major	Mr	944	445-9192	Z	Guillot
2	Mr	102	Mr	3	Jefferson Ct	PO Box 3	Augusta	ME	US	04330	2019-11-01	2018-08-01	Cody	Mr	507	663-3594	S	Pemberton
3	Rev	103	Rev	4	Madison St	PO Box 4	Honolulu	HI	US	96801	2019-10-21	2018-08-01	Beverly	Rev	812	117-3600	Y	Fetzer
3	Mrs	104	Mrs	5	Monroe Rd	PO Box 5	Atlanta	GA	US	30301	2019-10-17	2018-08-01	Veronica	Mrs	178	717-5244	J	Geier
3	Rev	105	Rev	6	Adams Ave	PO Box 6	Tallahassee	FL	US	32301	2019-02-17	2018-08-01	Denny	Rev	581	353-9702	M	Brookshire
3	Mr	106	Mr	7	Jackson Blvd	PO Box 7	Jackson	MS	US	39201	2019-02-10	2018-08-01	Hong	Mr	883	211-4460	K	Pilkington
4	Dr	107	Dr	8	Van Buren Wy	PO Box 8	Carson City	NV	US	89701	2019-09-09	2018-08-01	Florentino	Dr	109	147-3797	A	Brodeur
4	Mr	108	Mr	9	Harrison Ln	PO Box 9	Oklahoma City	OK	US	73101	2019-07-06	2018-08-01	Willie	Mr	765	743-9849	C	Schuh
4	Dr	109	Dr	10	Tyler Cir	PO Box 10	Frankfort	KY	US	40601	2019-04-05	2018-08-01	Elton	Dr	133	386-5591	K	Cauthen
4	Dr	110	Dr	11	Polk Pk	PO Box 11	Cheyenne	WY	US	82001	2019-07-10	2018-08-01	Stephaine	Dr	187	710-6557	S	Valladares
5	Mr	111	Mr	12	Taylor Pkwy	PO Box 12	Dover	DE	US	19901	2019-03-24	2018-08-01	Elisha	Mr	665	168-4159	V	Newcomb
5	Ms	112	Ms	13	Fillmore Dr	PO Box 13	Montgomery	AL	US	36101	2019-03-14	2018-08-01	Merry	Ms	479	757-1812	H	Ratliff
5	Rev	113	Rev	14	Pierce Sq	PO Box 14	Carson City	NV	US	89701	2019-06-13	2018-08-01	Chandra	Rev	838	120-7864	M	Laliberte
6	Mr	114	Mr	15	Buchanan Ct	PO Box 15	Denver	CO	US	80012	2019-03-21	2018-08-01	Dean	Mr	364	559-7503	R	Corcoran
6	Mrs	115	Mrs	16	Lincoln St	PO Box 16	Boston	MA	US	02108	2019-02-01	2018-08-01	Amy	Mrs	135	753-9815	Z	Coronado
6	Mrs	116	Mrs	17	Johnson Rd	PO Box 17	Washington	DC	US	20001	2019-06-23	2018-08-01	Berenice	Mrs	850	118-3145	C	Amerson
7	Mrs	117	Mrs	18	Grant Ave	PO Box 18	Baton Rouge	LA	US	70801	2019-02-04	2018-08-01	Cecile	Mrs	367	575-5385	O	Maki
7	Mr	118	Mr	19	Hayes Blvd	PO Box 19	Boise	ID	US	83701	2019-06-11	2018-08-01	Emerson	Mr	433	207-1724	C	Avitia
7	Rev	119	Rev	20	Garfield Wy	PO Box 20	Des Moines	IA	US	50301	2019-06-19	2018-08-01	Marylin	Rev	821	648-3858	R	Tolman

```
20 rows in set (0.00 sec)

mysql>
```

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

```
mysql>
mysql>
mysql> select * from address limit 10;
```

user_id	address_number	primary_name_prefix	primary_first_name	primary_middle_name	primary_last_name	primary_name_title	secondry_name_prefix	secondry_first_name	secondry_middle_name	secondry								
last_name	secondry_name_title	street_address	po_box	city	state_code	country	zip_code	date_created	last_updated	mobile_area_code	mobile_number							
2	Miss	100	Miss	1	Washington Dr	PO Box 1	Olympia	WA	US	98501	2019-03-22	2018-08-01	Maud	Miss	345	216-6341	V	Berge
2	Mr	101	Mr	2	Adams Sq	PO Box 2	Nashville	TN	US	37201	2019-02-05	2018-08-01	Major	Mr	944	445-9192	Z	Guillot
2	Mr	102	Mr	3	Jefferson Ct	PO Box 3	Augusta	ME	US	04330	2019-11-01	2018-08-01	Cody	Mr	507	663-3594	S	Pemberton
3	Rev	103	Rev	4	Madison St	PO Box 4	Honolulu	HI	US	96801	2019-10-21	2018-08-01	Beverly	Rev	812	117-3600	Y	Fetzer
3	Mrs	104	Mrs	5	Monroe Rd	PO Box 5	Atlanta	GA	US	30301	2019-10-17	2018-08-01	Veronica	Mrs	178	717-5244	J	Geier
3	Rev	105	Rev	6	Adams Ave	PO Box 6	Tallahassee	FL	US	32301	2019-02-17	2018-08-01	Denny	Rev	581	353-9702	M	Brookshire
3	Mr	106	Mr	7	Jackson Blvd	PO Box 7	Jackson	MS	US	39201	2019-02-10	2018-08-01	Hong	Mr	883	211-4460	K	Pilkington
4	Dr	107	Dr	8	Van Buren Wy	PO Box 8	Carson City	NV	US	89701	2019-09-09	2018-08-01	Florentino	Dr	109	147-3797	A	Brodeur
4	Mr	108	Mr	9	Harrison Ln	PO Box 9	Oklahoma City	OK	US	73101	2019-07-06	2018-08-01	Willie	Mr	765	743-9849	C	Schuh
4	Dr	109	Dr	10	Tyler Cir	PO Box 10	Frankfort	KY	US	40601	2019-04-05	2018-08-01	Elton	Dr	133	386-5591	K	Cauthen

```
10 rows in set (0.00 sec)

mysql>
```

# 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 [www.genrocket.com](http://www.genrocket.com).