# Developer Info:

Julius Lee  
eMail: [jwiradin@gmail.com](mailto:jwiradin@gmail.com)  
Mobile: 0470345166

# Shema creation

I used many-to-may on the address table just in case the front-end allow an option “same as home”. At this stage, one-to-may relationships for mail and phone is sufficient and many-to-may for for marketing/category , role and access role.

# Data generation

Name:

* For individuals, download sample names from several internet websites. Split up first name and last name. cross join first name and last name.
* For company, use street name and cross join with trades type (plumber, electrician, concrete, maintenance, etc).

Address:

* For street address, download street names from the electoral commission. Remove street type (st, rd, hwy, etc); Use random number as address number and cross join street name with street type to get a complete address line.
* For suburb, download the list of Australian suburbs. For this exercise, limit to only 5 suburbs per post code.
* Cross join the address line with suburb to complete the address.
* Primary addresses are generated randomly. For secondary addresses, the postcodes are limited to +/- 2 postcodes and must be in the same state.
* A customer is guaranteed to have a street address.

Phone number:

I only use land line numbers. The first 1/2 digits are from the real number format and the last 6 digits are random numbers. There is a check to ensure there is no phone number duplication.

Email address:

* For Individuals, first character of the first name append it with the last name and random number.
* For company, use ‘admin’ and append with company names without spaces and random number.

There is a check to ensure there is no email address duplication.

Marketing/Category and access role are randomly assigned to the customers. There is a check to make sure all customers are assigned.

# Stored procedure

The stored procedure is written without using dynamic SQL and cursor as requested and accepts all the parameters as specified.

I was not sure by request of using function to apply security and would like to find out the solution if mine was not the correct one.

From the query plan result I noticed that the query took a considerable resources on suburb joint. I changed the suburb index from non-clustered to clustered and reduced the query time (on my machine) from 32 seconds to 23 seconds, which is a 30% improvement.

# Testing

zz test schema.sql is a script to validate database validations.

Test case:

1. First name and last name or company name are required.
2. “First name and last name” and company name are mutually exclusive. i.e., only one element can have non null values
3. Customer can only have one address type.
4. Customer can have the same address for different address type.

zz test retrieval.sql is a script to validate usp\_customer\_retreiveList is working as expected.

Test case:

1. Validates number of returned rows by user role filter.
2. Validates number of returned rows by individual/organisation filter.
3. Validates number of returned rows by a state filter.
4. Validates number of returned rows by a name filter.
5. Validates number of returned rows by a suburb filter.
6. Validates number of returned rows by a marketing/category filter.
7. Validates paging function.
8. Validate combination of 5,6,7,8,9,10. Due to time constraint, not all combinations are tested.

# Test Result

Zz test schema.sql

validating customer check

--insert company name. Expected result success.

--insert company name. Success. 1 created

--insert first name and last name. Expected result success.

--insert first name and last name. Success. 1 created

--insert company name, first name and last name. Expected result failed

\*\*insert company name, first name and last name. Failed

--insert last name. Expected result failed

\*\*insert last name. failed

--insert first name. Expected result failed

\*\*insert first name. failed

validating - address

--Inserting a customerAddress expected result success

--Inserting a customeraddress success. 1 created

--Inserting a customerAddress with duplicate addressID but different addresstypeid expected result success

--Inserting a customerAddress with duplicate addressID but different addresstypeid. success. 1 created

--Inserting a customerAddress with duplicate addressTypeID expected result failed

\*\*Inserting a customeraddress with duplicate addressTypeID failed.

Zz test Retrieval.sql

Start extracting data for user 2 with no filter - 31 Jul 2015 11:51:26

Finish extracting data for user 2 with no filter - 31 Jul 2015 11:52:47 run: 81

Validating condition:number of rows returned for user 2 - 31 Jul 2015 11:52:47

Validation successful condition:number of rows returned for user 2 - 31 Jul 2015 11:52:47

Start extracting data for user 2 filter - individual only 31 Jul 2015 11:52:47

Finish extracting data for user 2 filter - individual only 31 Jul 2015 11:53:41 run: 54

Validating condition:number of rows returned for user 2 filter - individual only 31 Jul 2015 11:53:41

Validation successful condition:number of rows returned for user 2 - 31 Jul 2015 11:53:42

Start extracting data for user 2 filter - company only 31 Jul 2015 11:53:42

Finish extracting data for user 2 filter - company only 31 Jul 2015 11:54:17 run: 35

Validating condition:number of rows returned for user 2 filter - company only 31 Jul 2015 11:54:17

Validation successful condition:number of rows returned for user 2 filter - company only 31 Jul 2015 11:54:18

Start extracting data for user 2 filter - NSW only 31 Jul 2015 11:54:18

Finish extracting data for user 2 filter - NSW only 31 Jul 2015 11:54:44 run: 26

Validating condition:number of rows returned for user 2 filter - NSW only 31 Jul 2015 11:54:44

Validation successful condition:number of rows returned for user 2 filter - NSW only 31 Jul 2015 11:54:45

Start extracting data for user 2 filter - Name Alfred only 31 Jul 2015 11:54:45

Finish extracting data for user 2 filter - Name Alfred only 31 Jul 2015 11:54:52 run: 7

Validating condition:number of rows returned for user 2 filter - Name Alfred only 31 Jul 2015 11:54:52

Validation successful condition:number of rows returned for user 2 filter - Name Alfred only 31 Jul 2015 11:54:53

Start extracting data for user 2 filter - suburb Sydney only 31 Jul 2015 11:54:53

Finish extracting data for user 2 filter - suburb Sydney only 31 Jul 2015 11:54:58 run: 5

Validating condition:number of rows returned for user 2 filter - suburb Sydney only 31 Jul 2015 11:54:58

Validation successful condition:number of rows returned for user 2 filter - suburb Sydney only 31 Jul 2015 11:54:59

Start extracting data for user 2 filter - Security door, Security window only 31 Jul 2015 11:54:59

Finish extracting data for user 2 filter - Security door, Security window only 31 Jul 2015 11:56:12 run: 73

Validating condition:number of rows returned for user 2 filter - Security door, Security window only 31 Jul 2015 11:56:12

Validation successful condition:number of rows returned for user 2 filter - Security door, Security window only 31 Jul 2015 11:56:14

Start extracting data for user 2 filter - paging page 3/30 rows 31 Jul 2015 11:56:14

Finish extracting data for user 2 filter - paging page 3/30 rows 31 Jul 2015 11:56:45 run: 31

Validating condition:number of rows returned for user 2 filter - paging page 3/30 rows 31 Jul 2015 11:56:45

Validation successful condition:number of rows returned for user 2 filter - paging page 3/30 rows 31 Jul 2015 11:56:48

Start extracting data for user 2 filter - individual, name like smith and state = VIC paging page 3/10 rows 31 Jul 2015 11:56:48

Finish extracting data for user 2 filter - individual, name like smith and state = VIC paging page 3/10 rows 31 Jul 2015 11:56:50 run: 2

Validating condition:number of rows returned for user 2 filter - individual, name like smith and state = VIC paging page 3/10 rows 31 Jul 2015 11:56:50

Validation successful condition:number of rows returned for user 2 filter - individual, name like smith and state = VIC paging page 3/10 rows 31 Jul 2015 11:56:50

# Installation notes

List of sql files. Please execute them sequentially. Note there is no “use” database defined in the scripts. The function and stored procedure access are set to public. Scripts with zz prefix are testing scripts. All test messages are “print”ed.

* 01 - Create tables.sql
* 02 - Populate reference tables.sql
* 03 - Populate User table.sql
* 04 - Prepare to populate customer.sql
* 05 - Prepare to populate address.sql
* 06 - Populate customer.sql
* 07 - Populate address.sql
* 08 - Populate Phone Number.sql
* 09 - Populate Email Address.sql
* 10 - Populate Customer Category.sql
* 11 - Populate Access Role.sql
* 12 - dbo.ufn\_getRoles.sql
* 13 - dbo.ufn\_Split.sql
* 14 - dbo.usp\_customer\_retrieveList.sql
* zz test schema.sql
* zz test usp\_customer\_retrievelist.sql

Let me know if there is any issue jwiradin@gmail.com.