MSSQLTips.com
brought to you by Edgewood Solutions

Your daily source for SQL Server Tips

Generating SQL Server Test Data with Visual Studio 2010

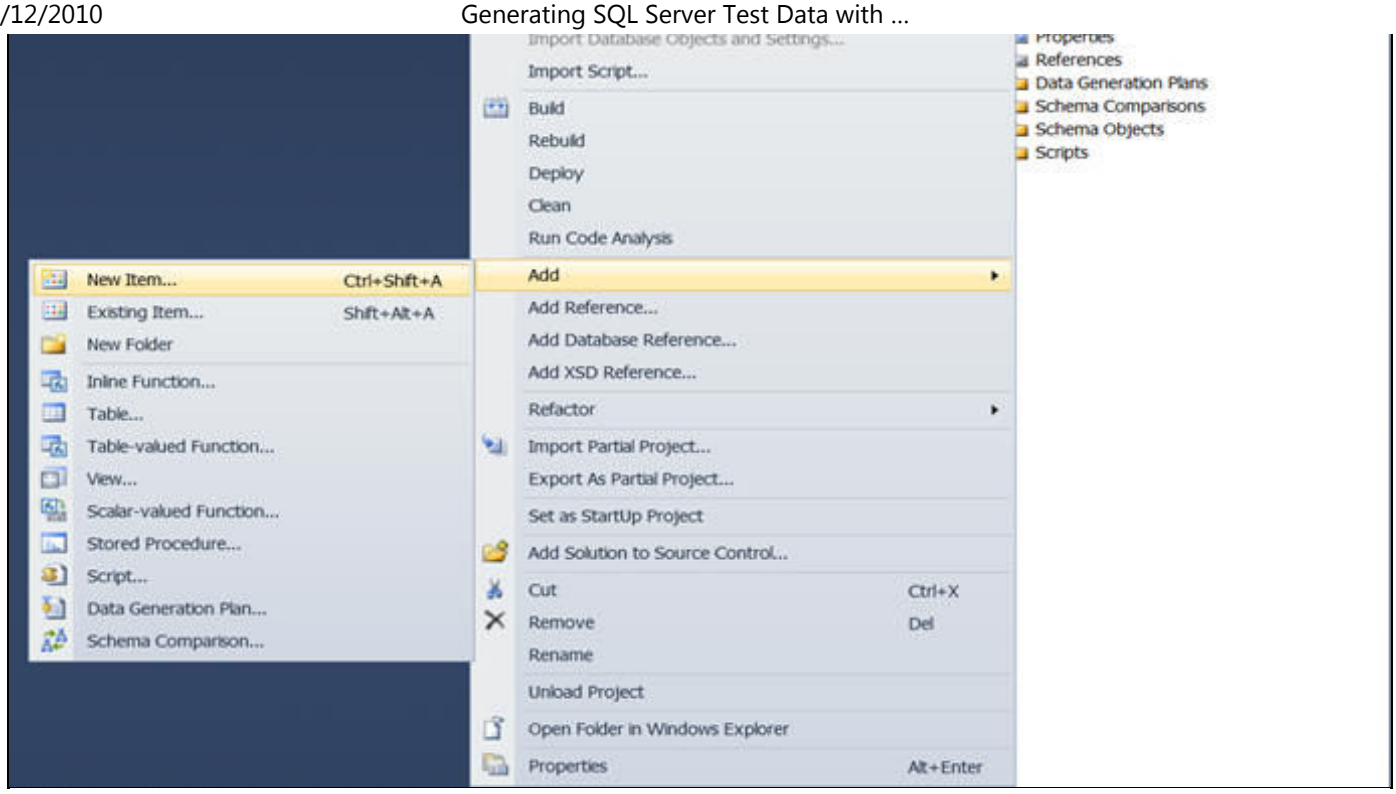
Written By: Arshad Ali -- 12/23/2010

Problem
As a database developer or tester sometimes you need to have production like data in your environment for your development or testing, but you cannot have the production data because of security and privacy issues. So how you can generate test data or replicate similar data as in production for your development or test environment?

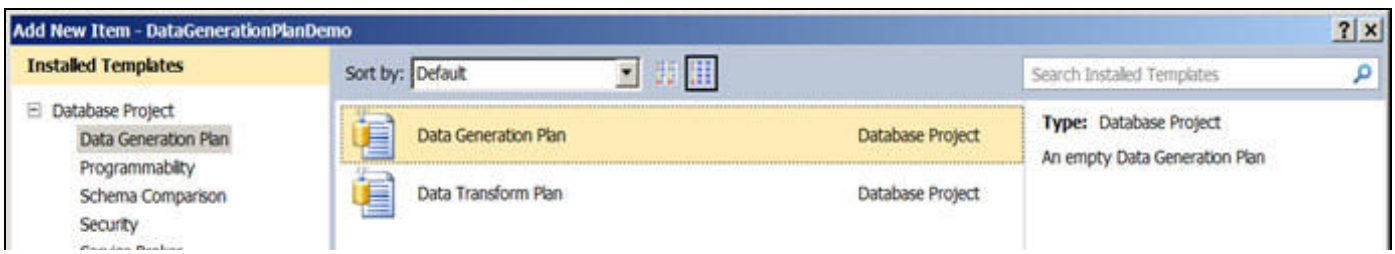
Solution
Visual Studio Database edition provides several features for database development and testing. One of them is generating random test data using Data Generation Plan. Data Generation Plan contains how you want your test data to be generated for your specific tables and columns. It uses several built in data generators which generate random data or generate data from other data sources as per your column data type. You can change properties of these data generators to define the range and format of the data being generated as well. If it does not suffice your need or you have different business rules which can't be satisfied by using the built-in data generators, you can even create your own custom data generators too.

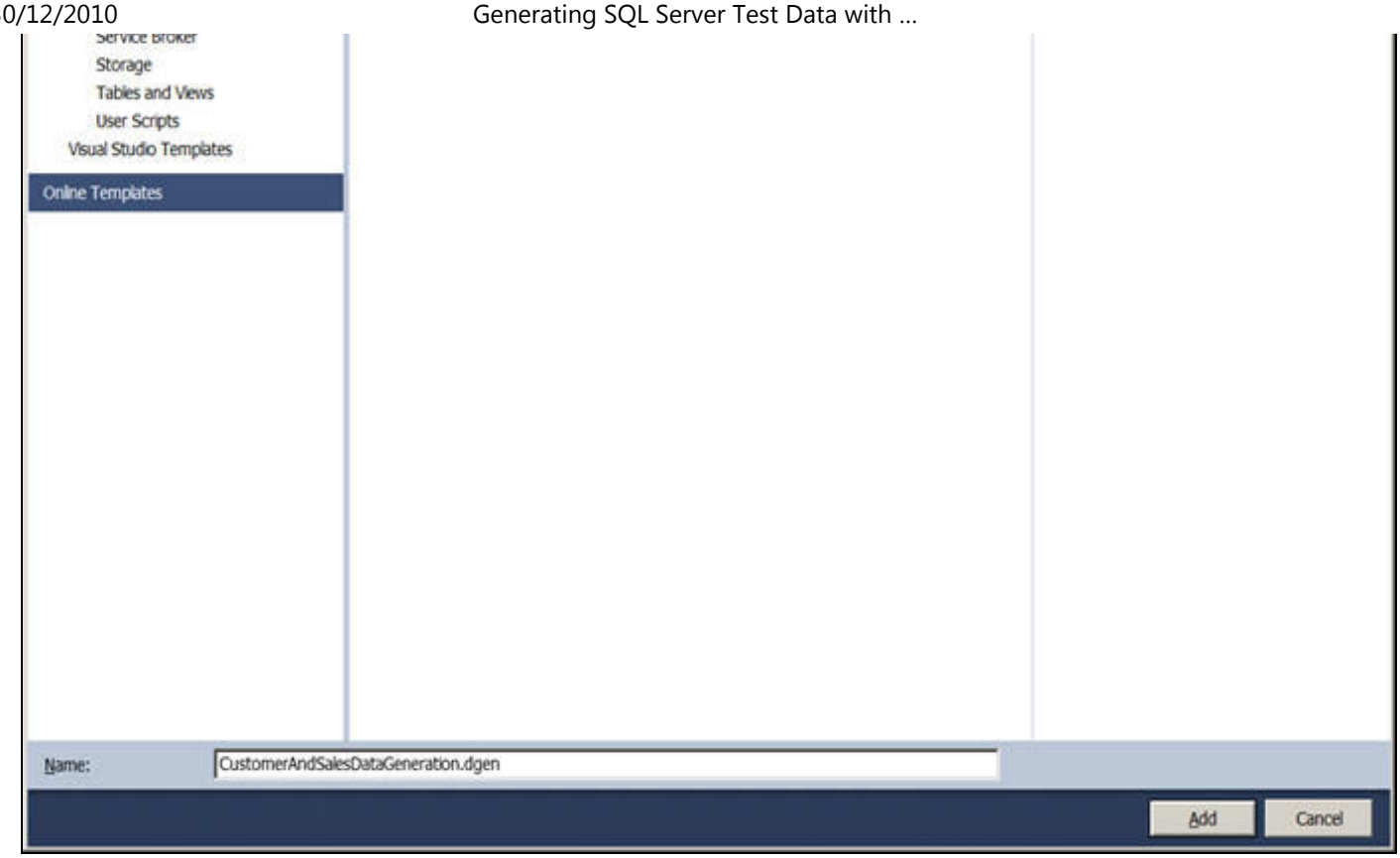
Note: In this demonstration I am going to show how you can generate test data in Visual Studio 2010 Ultimate edition although you can do the same with Visual Studio 2005/2008 Database edition too.

Launch Microsoft Visual Studio 2010 IDE (Integrated Development Studio) and open your database project (if you are not aware of database projects and want to know what is and how it works click here). Right click on your project node in the Solution Explorer, go to Add and then click on New Item as shown below:

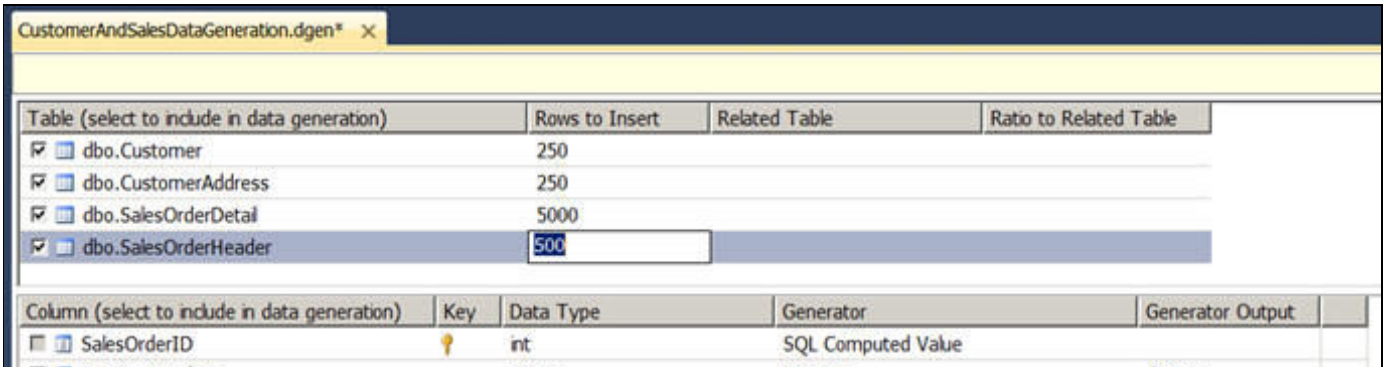


In Add New Item dialog box, select Data Generation Plan node in the left side under Database Project node and choose empty Data Generation Plan template from the detail section. Specify an appropriate name for your data generation plan as shown below and click on the Add button.



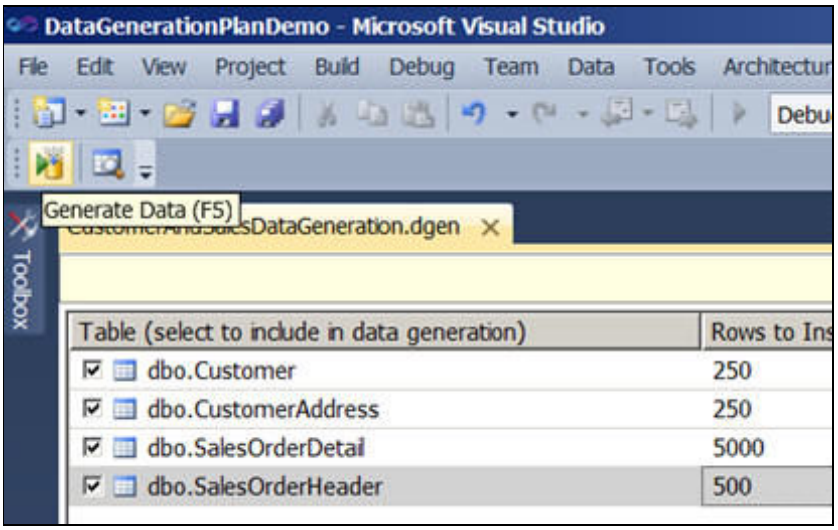


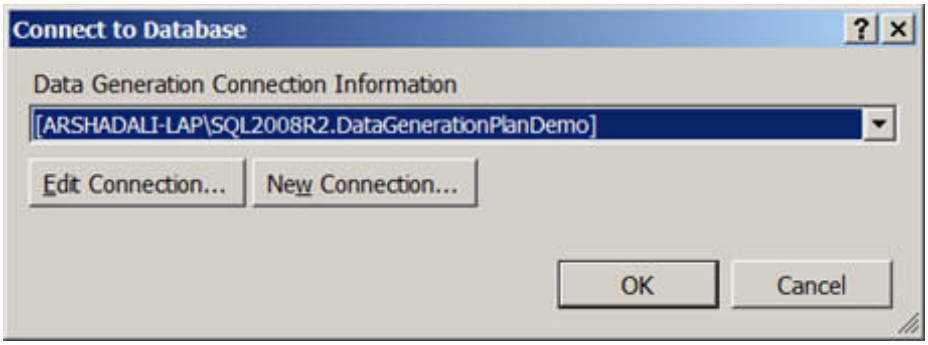
The Data Generation Plan will scan through the schema objects of your database project and will list all the tables that exist. By default all the tables are included in the plan for data generation. You can choose which tables you want to generate test data and how many records you want to generate for each table. Depending on the relationship defined the data generation plan shows related tables and you can specify a ratio for related tables for data generation as shown below.



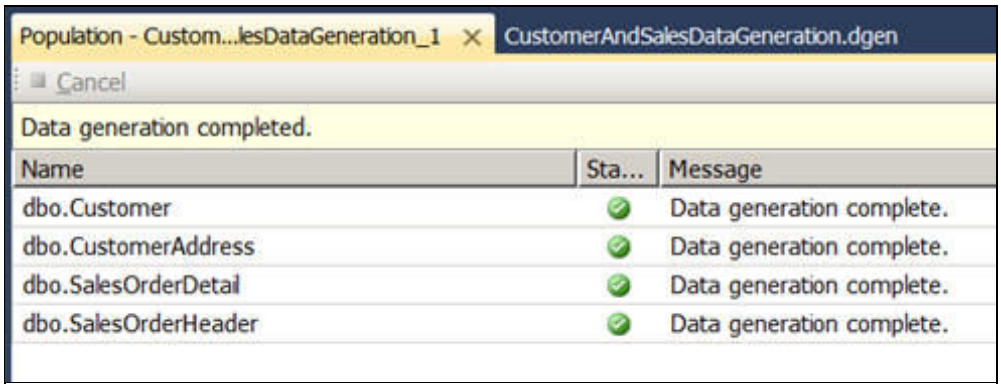
RevisionNumber	tinyint	TinyInt	Output
OrderDate	datetime	DateTime	Output
DueDate	datetime	DateTime	Output
ShipDate	datetime	DateTime	Output
Status	tinyint	TinyInt	Output
SalesOrderNumber	<unnamed>	SQL Computed Value	
CustomerID	int	Integer	Output
ShipToAddressID	int	Integer	Output
BillToAddressID	int	Integer	Output
ShipMethod	nvarchar (50)	String	Output
CreditCardApprovalCode	varchar (15)	String	Output
SubTotal	money	Money	Output
TaxAmt	money	Money	Output
Freight	money	Money	Output
TotalDue	<unnamed>	SQL Computed Value	
Comment	nvarchar (max)	String	Output
rowguid	uniqueidentifier	Guid	Output
ModifiedDate	datetime	DateTime	Output

To start generating data, click on the Generate Data icon in the toolbar or hit F5. This will launch the Connect to Database wizard, as shown below, where you need to specify the database details where you want to generate the test data.

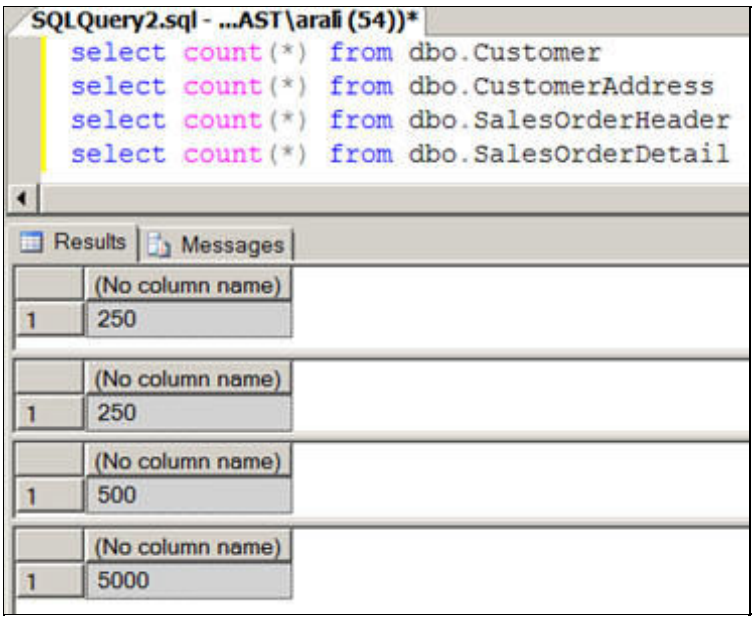




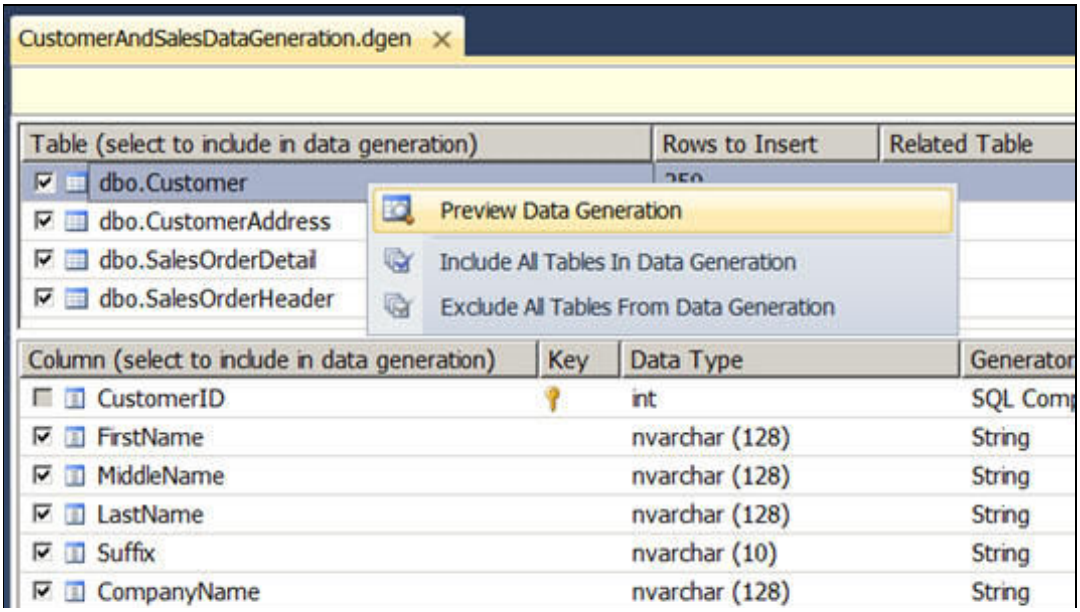
The moment you specify database connection and click on the OK button, the data generation plan starts generating data for each table selected in the plan and shows the status as follows. The wizard will also ask if you want to delete data from the target tables before data generation begins inserting rows.



To verify if the data has been generated, connect to your target database and query the record counts for those tables as shown below.



You don't need to generate test data to verify it in your target tables, but rather you can preview the data to be generated beforehand and generate test data only if it meets your requirement (this saves you from creating multiple iterations of test data). Right click on the table in the data generation plan screen and click on Preview Data Generation menu option as shown below and the test data will be shown in a preview screen as shown below.



SalesPerson	nvarchar (256)	String
EmailAddress	nvarchar (50)	String
PasswordHash	varchar (128)	String
PasswordSalt	varchar (10)	String
rowguid	uniqueidentifier	Guid
ModifiedDate	datetime	DateTime

Data Generation Preview - dbo.Customer

CustomerID	FirstName	MiddleName	LastName	Suffix	CompanyName	SalesPerson	EmailAddress	PasswordHash
	ebÕxøPúÒMá...	ebÕxøPúÒMá...	ebÕxøPúÒMá...	ebÕx	ebÕxøPúÒMá...	ebÕxøPúÒMá...	ebÕxøPúÒMá...	ebÕxøPúÒMá...
	EEçÃäáaÈUbv...	EEçÃäáaÈUbv...	EEçÃäáaÈUbv...	PúÒMánHHzQ	EEçÃäáaÈUbv...	NEIyKâIÇôjOI...	azH60UòY1ø...	EEçÃäáaÈUbv...
	BÚnKäZãPEM...	BÚnKäZãPEM...	BÚnKäZãPEM...	K4azH60UòY	BÚnKäZãPEM...	ÔRNÕzÈDÂE...	mCzIVGûÈáS...	BÚnKäZãPEM...
	âÐÍÀEgûÊÎpÂ...	âÐÍÀEgûÊÎpÂ...	âÐÍÀEgûÊÎpÂ...	øLDõgMpá	âÐÍÀEgûÊÎpÂ...	MIupxÓÍTëÕ...	ÇôjOI	âÐÍÀEgûÊÎpÂ...
	KÒ4Ójfõ¼Én...	KÒ4Ójfõ¼Én...	KÒ4Ójfõ¼Én...	1V	KÒ4Ójfõ¼Én...	ùÀ1èøÉ7xsCa...	OÙ4IZÒ4Rva...	KÒ4Ójfõ¼Én...
	ÙÑzwôØ0óÒ...	ÙÑzwôØ0óÒ...	ÙÑzwôØ0óÒ...	kLá6ÌUEE	ÙÑzwôØ0óÒ...	éwJcÑnGkóR...	ÚwKõLMñèÂE...	ÙÑzwôØ0óÒ...
	úuPO0ØÍMèë...	úuPO0ØÍMèë...	úuPO0ØÍMèë...	ÃãáaÈUb	úuPO0ØÍMèë...	ëÕdódKòÀHô...	U6HErGùÒ	úuPO0ØÍMèë...
	ñ3õsjÁÃÀtâU...	ñ3õsjÁÃÀtâU...	ñ3õsjÁÃÀtâU...	zuðm	ñ3õsjÁÃÀtâU...	6ðÊïcÂwÐkUÛ...	ÀÚû4QTzÉaâ...	ñ3õsjÁÃÀtâU...
	Nk4eÏSkÀaCë...	Nk4eÏSkÀaCë...	Nk4eÏSkÀaCë...	z	Nk4eÏSkÀaCë...	Í1Tð7ÚÍGeèÙ...	jfõ¼EnWBôÑ...	Nk4eÏSkÀaCë...
	éâÏrô5EAXgó...	éâÏrô5EAXgó...	éâÏrô5EAXgó...	V	éâÏrô5EAXgó...	pøÎmêùúLïâÙ...	akZáeá1dñzb...	éâÏrô5EAXgó...
	YøvâHHÈøAQ...	YøvâHHÈøAQ...	YøvâHHÈøAQ...	û	YøvâHHÈøAQ...	ç1BhvTTõÚ¦j...	P7XØòmh0ò4...	YøvâHHÈøAQ...
	MOuYÙÊslT0...	MOuYÙÊslT0...	MOuYÙÊslT0...	ÈáSØBÚn	MOuYÙÊslT0...	ÃañsàÕÛ6rw...	ÂÐòø7zoúuP...	MOuYÙÊslT0...
	aãäiIBnQuôÍb...	aãäiIBnQuôÍb...	aãäiIBnQuôÍb...	ã	aãäiIBnQuôÍb...	R3qcAiT0zçïé...	ÔÐbdkÙlEsU0...	aãäiIBnQuôÍb...
	â53ÊgcÇÕâi1...	â53ÊgcÇÕâi1...	â53ÊgcÇÕâi1...	ãÏ	â53ÊgcÇÕâi1...	XÇ2ùêÃúØhj...	5ÒzÊeãFVebè...	â53ÊgcÇÕâi1...
	ZIWHrhDäè...	ZIWHrhDäè...	ZIWHrhDäè...	EM	ZIWHrhDäè...	ëgÒwÑAÒRÓÍ...	ÈÓSIãÓgEréF...	ZIWHrhDäè...

There are couple of standard and built-in data generators provided (to generate random test data for different data types) which the data generation plan chooses automatically to use depending on the schema of your table or data type of your column. You can change different properties of these data generators associated with each column to define a range or format of data being generated for that column. If this default selection does not suit your requirement, you can change it to use a different data generator than the default. Not only this, you can even create you own custom data generator, to learn more about it [click here.]

CustomerAndSalesDataGeneration.dgen*

Table (select to include in data generation)	Rows to Insert	Related Table	Ratio to Related Table
dbo.Customer	250		
dbo.CustomerAddress	250		
dbo.SalesOrderDetail	5000		
dbo.SalesOrderHeader	500		

Column (select to include in data generation)	Key	Data Type	Generator	Generator Output
CustomerID	🔑	int	SQL Computed Value	
FirstName		nvarchar (128)	String	Output
MiddleName		nvarchar (128)		Output
LastName		nvarchar (128)		Output
Suffix		nvarchar (10)		Output
CompanyName		nvarchar (128)		Output
SalesPerson		nvarchar (256)		Output
EmailAddress		nvarchar (50)		Output
PasswordHash		varchar (128)		Output

Float
Integer
Real
Regular Expression
Sequential data bound generator
SmallInt
String
TinyInt

Also note, if you want test data to be generated by using SQL SELECT commands from a data source, you can use the Data Bound Generator.

Next Steps

- Review these additional tips
  - Populating a SQL Server Test Database with Random Dataa.
  - SQL Server Unit Testing with Visual Studio 2010
  - SQL Data Comparison with Visual Studio 2010
  - SQL Schema Comparison with Visual Studio 2010
  - How to: Create Data Generation Plans
  - Populating a SQL Server Test Database with Random Data