SQL Server 2012 Features

[Introduction](http://www.codeproject.com/Articles/526621/Top-exciting-features-of-SQL-Server-Part#Introduction)

[Feature number 1 (Revolution):- Column store indexes](http://www.codeproject.com/Articles/526621/Top-exciting-features-of-SQL-Server-Part#Feature_number_1_(Revolution):-_Column_store_indexes)  
[Feature number 2 (Evolution):- Sequence objects](http://www.codeproject.com/Articles/526621/Top-exciting-features-of-SQL-Server-Part#Feature_number_2_(Evolution):-_Sequence_objects)  
[Feature number 3 (Revolution):- Pagination](http://www.codeproject.com/Articles/526621/Top-exciting-features-of-SQL-Server-Part#Feature_number_3_(Revolution):-_Pagination)  
[Feature number 4 (Revolution):- Contained database](http://www.codeproject.com/Articles/526621/Top-exciting-features-of-SQL-Server-Part#Feature_number_4_(Revolution):-_Contained_database)  
[Feature number 5 (Evolution):- Error handling](http://www.codeproject.com/Articles/526621/Top-exciting-features-of-SQL-Server-Part#Feature_number_5_(Evolution):-_Error_handling)  
[Feature number 6 (Evolution):- User defined roles](http://www.codeproject.com/Articles/561797/Top-20-exciting-features-of-SQL-Server-2012-Part-2#Featurenumber6(Evolution):-Userdefinedroles)

[Feature number 7 (Evolution):- Windows server core support](http://www.codeproject.com/Articles/561797/Top-20-exciting-features-of-SQL-Server-2012-Part-2#Featurenumber7(Evolution):-Windowsservercoresupport)

[Feature number 8 (Revolution):- Tabular Model (SSAS)](http://www.codeproject.com/Articles/561797/Top-20-exciting-features-of-SQL-Server-2012-Part-2#Featurenumber8(Revolution):-TabularModel(SSAS))

[Feature number 9 (Revolution):- Power view](http://www.codeproject.com/Articles/561797/Top-20-exciting-features-of-SQL-Server-2012-Part-2#Featurenumber9(Revolution):-Powerview)

[Feature number 10 (Revolution):- DQS Data quality services](http://www.codeproject.com/Articles/561797/Top-20-exciting-features-of-SQL-Server-2012-Part-2#Featurenumber10(Revolution):-DQSDataqualityservices)

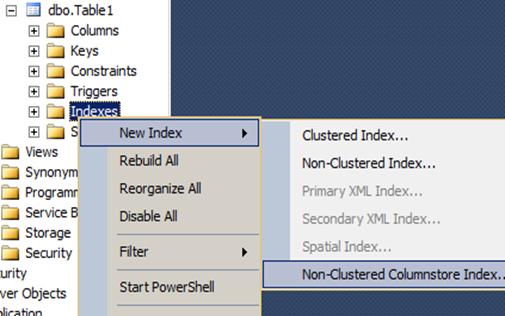
Introduction

SQL Server 2012 was released on April 2012 and it has started becoming favorite among professionals. Any new product comes from Microsoft the first thing I personally ask myself, is it worth to jump in?. Is it worth to spend customer’s hard earned money to get in to that product?. The way to assess the same is dividing the product features in to   “revolution” and “evolution”. “Revolution” means it’s completely a new thing while “evolution” means there was something already and it has been improvised.

In this article I will discuss 20 features of SQL Server 2012 which I personally like (we can agree to disagree on my list) and from these 20 features, 7 are revolution and 13 are evolution. This article is divided in to 4 parts, in every part we will discuss 5 features. So let’s start with the first 5 top features.

Feature number 1 (Revolution):- Column store indexes

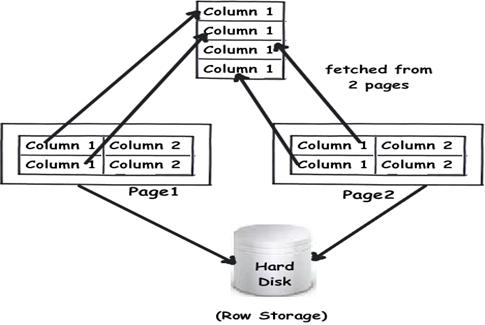
Column store indexes are unexpected and awesome feature. When I read this feature first time I was like, mouth wide open. You can get this feature when you right click on the indexes folder as “Non-Clustered Column store Index” , as shown in the below figure.



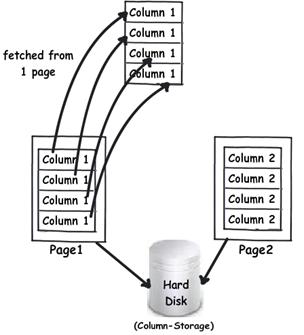
So let’s quickly understand what exactly it does. Now Relational database store data “row wise”. These rows are further stored in 8 KB page size.

For instance you can see in the below figure we have table with two columns “Column1” and “Column2”. You can see how the data is stored in two pages i.e. “page1” and “page2”. “Page1” has two rows and “page2” also has two rows. Now if you want to fetch only “column1”, you have to pull records from two pages i.e. “Page1” and “Page2”, see below for the visuals.

As we have to fetch data from two pages its bit performance intensive.



If somehow we can store data column wise we can avoid fetching data from multiple pages. That’s what column store indexes do. When you create a column store index it stores same column data in the same page. You can see from the below visuals, we now need to fetch “column1” data only from one page rather than querying multiple pages.



Feature number 2 (Evolution):- Sequence objects

This feature is good to have and I personally feel it just mimics Oracle’s sequence objects.  Looks like it’s just a good to have feeling, if Oracle has it why not SQL Server. A sequence object generates sequence of unique numeric values as per specifications. Many developers would have now got a thought, we have something similar like this called as “Identity” columns. But the big difference is sequence object is independent of a table while identity columns are attached to a table.

Below is a simple code to create a sequence object. You can see we have created a sequence object called as “MySeq” with the following specification:-

* Starts with value 1.
* Increments with value 1 Minimum value it should start is with zero.
* Maximum it will go to 100. No cycle defines that once it reaches 100 it will throw an error.
* If you want to restart it from 0 you should provide “cycle”.
* “cache 50”  specifies that till 50 the values are already incremented in to cache to reduce IO. If you specify “no cache” it will make input output on the disk.

create sequence MySeq as int

start with 1 -- Start with value 1

increment by 1-- Increment with value 1

minvalue 0 -- Minimum value to start is zero

maxvalue 100 -- Maximum it can go to 100

no cycle -- Do not go above 100

cache 50 -- Increment 50 values in memory rather than incrementing from

IO

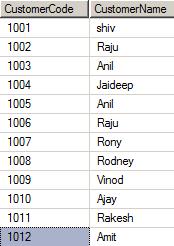
To increment the value we need to call the below select statement. This is one more big difference as compared to identity.In identity the values increment when rows are added here we need to make an explicit call.

SELECT NEXT VALUE FOR dbo.MySequence AS seq\_no;

Feature number 3 (Revolution):- Pagination

There are instances when you want to display large result sets to the end user. The best way to display large result set is to split them i.e.  apply pagination. So developers had their own hacky ways of achieving pagination using “top”, “row\_number” etc. But from SQL Server 2012 onwards we can do pagination by using “OFFSET” and “FETCH’ commands.

For instance let’s says we have the following customer table which has 12 records. We would like to split the records in to 6 and 6.



So doing pagination is a two-step process: -

* First mark the start of the row by using “OFFSET” command.
* Second specify how many rows you want to fetch by using “FETCH” command.

You can see in the below code snippet we have used “OFFSET” to mark the start of row from “0”position. A very important note order by clause is compulsory for “OFFSET” command.

select \* from

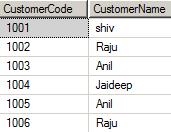
tblcustomer order by customercode

offset 0 rows – start from zero

In the below code snippet we have specified we want to fetch “6” rows from the start “0”position specified in the “OFFSET”.

fetch next 6 rows only

Now if you run the above SQL you should see 6 rows.



To fetch the next 6 rows just change your “OFFSET” position. You can see in the below code snippet I have modified the offset to 6. That means the row start position will from “6”.

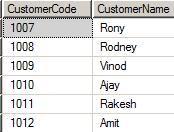
select \* from

tblcustomer order by customercode

offset 6 rows

fetch next 6 rows only

The above code snippet displays the next “6” records , below is how the output looks.



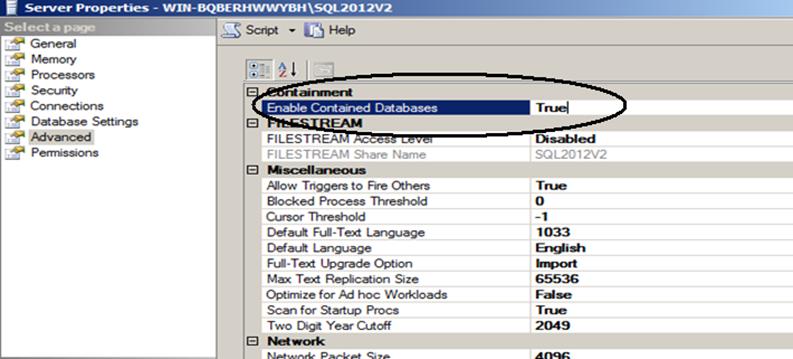
Feature number 4 (Revolution):- Contained database

This is a great feature for people who have to go through pain of SQL Server database migration again and again. One of the biggest pains in migrating databases is user accounts.  SQL Server user resides either in windows ADS or at SQL Server level as SQL Server users.  So when we migrate SQL Server database from one server to other server these users have to be recreated again. If you have lot’s of users you would need one dedicated person sitting creating one’s for you.

So one  of the requirements from easy migration perspective is  to create databases which are self-contained. In other words, can we have a database with meta-data information, security information etc with in the database itself. So that when we migrate the database, we migrate everything with it.  There’s where “Contained” database where introduced in SQL Server 2012.

Creating contained database is a 3 step process: -

**Step 1: -** First thing is to enable contained database at SQL Server instance level. You can do the same by right clicking on the SQL Server instance and setting  “Enabled Contained Database” to “true”.



You can achieve the same by using the below SQL statements as well.

sp\_configure 'show advanced options',1

GO

RECONFIGURE WITH OVERRIDE

GO

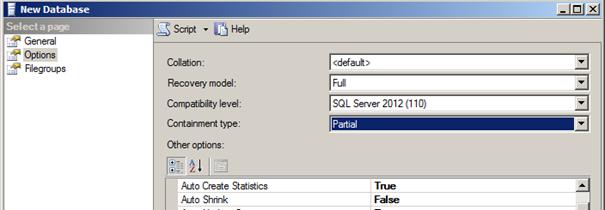
sp\_configure 'contained database authentication', 1

GO

RECONFIGURE WITH OVERRIDE

GO

**Step 2 -** The next step is to enable contained database at database level. So when create a new database set “Containment type” to partial as shown in the below figure.



You can also create database with “containment” set to “partial” using the below SQL code.

CREATE DATABASE [MyDb]

CONTAINMENT = PARTIAL

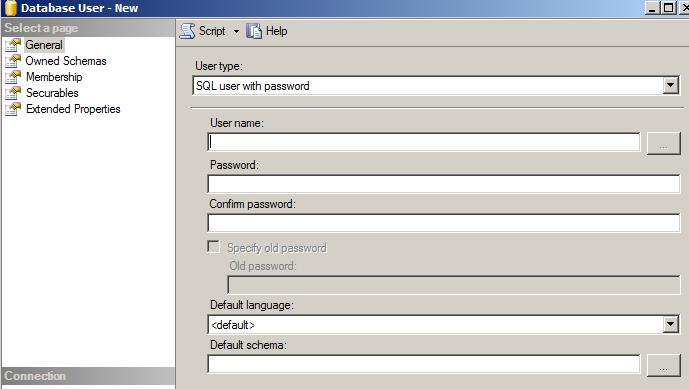
ON PRIMARY

( NAME = N'My', FILENAME = N'C:\My.mdf')

LOG ON

( NAME = N'My\_log', FILENAME =N'C:\My\_log.ldf')

**Step 3: -** The final thing now is to test if “contained” database fundamental is working or not. Now we want the user credentials to be part of the database , so we need to create user as “SQL User with password”.



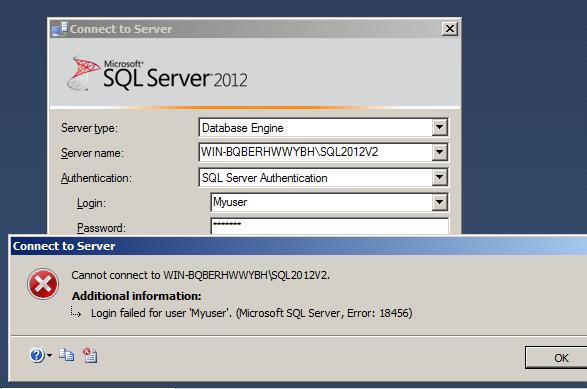
You can achieve the same by using the below script.

CREATE USER MyUser

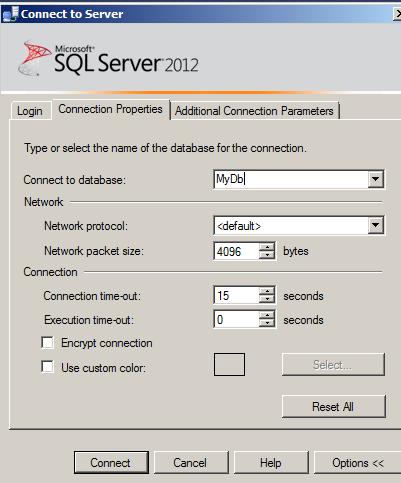
WITH PASSWORD = 'pass@123';

GO

Now if you try to login with the user created, you get an error as shown in the below figure. This proves that the user is not available at SQL Server level.



Now click on options and specify the database name in “connect to database” , you should be able to login , which proves that user is part of database and not SQL Server



Feature number 5 (Evolution):- Error handling

As a developer I am personally very comfortable with using “try/catch/throw” syntax structure for error handling in c# or vb.net. Thanks to SQL Server team in 2005 they brought in “try/catch” structure which is very much compatible the way I as a developer was doing error handling in c#.  It was nightmare handling error using “IF” conditions and “@error” code before SQL Server 2005. Below is a sample code which shows how “try/catch” code looks.

begin try

declare @n int = 0;

set @n = 1/0;

end try

begin catch

print('divide by zero');

RAISERROR ( ‘Divide by zero‘, 16, 1) ;

end catch

But what still is itching me in the above code is when it comes to propagating errors back to the client I was missing the “THROW” command.  We still need to  use “RAISEERROR” which does the job, but lacks lot of capabilities which “THROW” has. For example to throw user defined messages you need to make entry in to “sys.messages” table.

Below is how the code with “throw” looks like.

begin try

-- The code where error has occurred.

end try

begin catch

-- throw error to the client

Throw;

end catch

If you want to throw exception with a user defined message defined you can use the below code. No entry need in the “sys.messages” table.

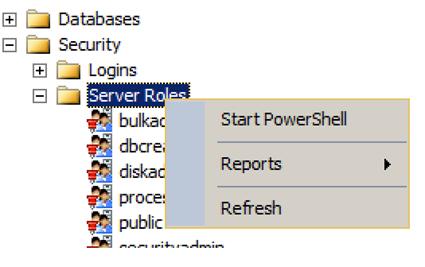
THROW 49903, 'User define exception.', 1

From SQL Server 2012 onwards use “Throw” rather than “raiseerror” , looking at the features of “throw” looks like sooner or later “raiseerror” will be deprecated . Below is a comparison table which explains the difference between “throw” vs “raiseerror”.

|  |  |  |
| --- | --- | --- |
|  | **Throw** | **RaiseError** |
| **User & system exception** | Can generate only user exception. | Can generate user and system exception. |
| **“Sys.Messages” table** | You can supply adhoc text does not need an entry in “Sys.Messages” table. | You need to make an entry in “Sys.Messages” table. |
| **Original exception.** | Original exception is propagated to the client. | Original exception is lost to the client. |

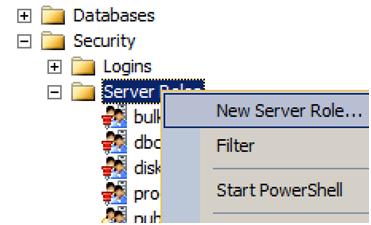
Feature number 6 (Evolution):- User defined roles

In SQL Server 2008 R2 we had the ability to create roles at database level. So you create customized roles at the database level and then assign them to users. But at the server level or instance level we did not have options of creating server roles. So if you right click on the “Server roles” you will not find any options for adding new server roles.

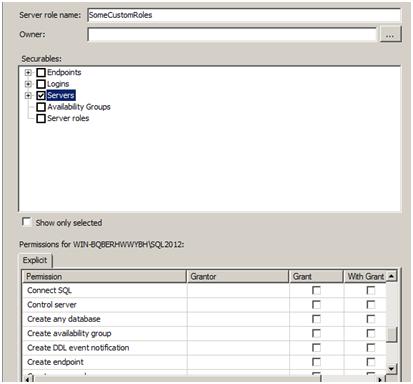


Now that’s a serious limitation. Let’s say you have two sets of database user one programmers and the other DBA’s. The programmers should be able to fire insert, update and delete queries while DBA’s should be able to create database, backup and do maintenance related activities. But DBA’s should not be able to fire insert, update and delete queries. But now because you have fixed roles the DBA’s get more access so they can even fire insert, update and delete queries. In simple words we need flexible roles.

In SQL Server 2012 you can create your own role and define customized permission for the role at a more granular level.

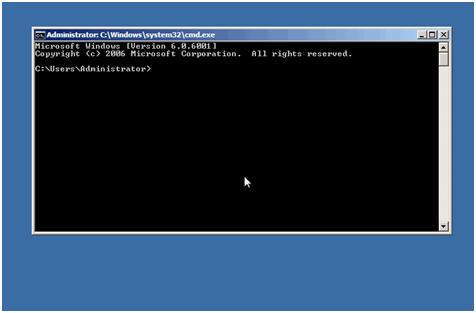


You can see in the below image how you can select permission at a finer level and create customized roles which can be later assigned to a user.



Feature number 7 (Evolution):- Windows server core support

This is a small evolution but an important one. Windows server core is one of the flavors of Windows operating system. It is a GUI less version of windows operating system. When you boot with windows core you would be surprised to get a simple DOS command line as shown in the figure as compared to start program files and crowded desktop short cuts. Because only necessary services are enabled, we have less memory consumption, simplified management as many features are not enabled and great stability. When we talk about SQL Server we would love to run it over an operating system with minimal feature enabled. So this is the most welcome feature and on production server using windows core is definitely the way to go.



Feature number 8 (Revolution):- Tabular Model (SSAS)

This is my personal top feature in SQL Server. Now the main goal of SSAS (SQL Server analysis service) is to do analysis, i.e. Convert data in to information. And SSAS achieves this by creating CUBES from data provided.

So the basic flow goes in 3 steps:

1. First data is brought to central database (data ware house) using SSIS package. The design of the data ware house system is normally in snow flake or star schema, so that we can create CUBE’s effectively.
2. Later analysis services runs over the data ware house to create CUBES to give multi-dimensional view of the data for better analysis.
3. We can then run different clients like EXCEL, SSRS etc to display data to different sections of users.



Can you guess one big potential problem with the above 3 step approach?. Give a PAUSE and think over it for a minute before you read ahead.

The biggest issue is simple business users CAN NOT CONTRIBUTE TO CUBES. I mean if I am a business user who would like to take data from a excel sheet, use my excel formula skills, derive conclusions and publish cubes, so how do I go about it?. My personal belief is that the best business analysis can only be done by business end users who actually do business on the field. They are the best people who understand things and can create CUBES which are more useful and logical.

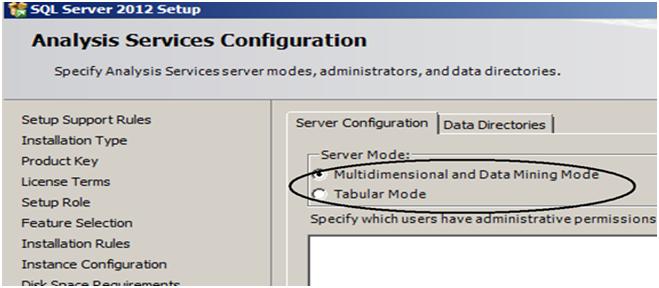
Also if you notice the previous steps its highly technical:-

* Can a simple business user create DB designs like snow flake / star schema?
* Can he use the complicated SSAS user interface to publish cubes?
* Does he have the knowledge of using SQL Server analysis capability?

**Note: -** We will change our vocabulary so that we are compatible with Microsoft vocabulary. We will term simple business users as personal users hence forth.

Now personal users work most of the time with EXCEL and if we really want to give analysis power to them, it should be inside excel itself. That’s what power pivot does. Power pivot is plugin which sits inside EXCEL and gives analytical capabilities to simple personal users to do analysis with data they have in EXCEL.

Now EXCEL data is in tabular format with rows and columns. So if you want publish this kind of analyzed data from EXCEL you need to have SSAS installed in tabular mode.



So now if you compare personal users with professional BI the workflow will be following:-

* **IMPORT**

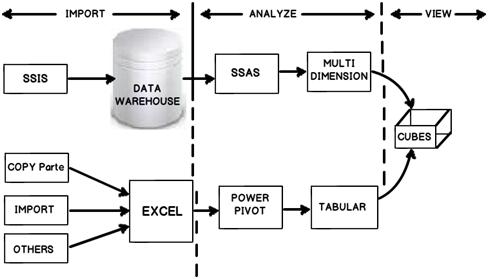
1. Professional BI personal will use SSIS, data flows, control flows etc.
2. Personal BI people can use import, copy past mechanism to get data in to EXCEL.

* **ANALYZE**

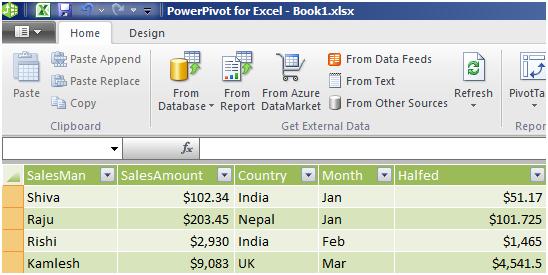
1. Professional BI person will uses SSAS , BI intelligence algorithm to do analysis. Once analysis is done they will publish in multi-dimension format.
2. Personal BI people will use power pivot and excel formulas to come to an analysis. Once analysis is done they will publish in tabular format.

* **VIEW**

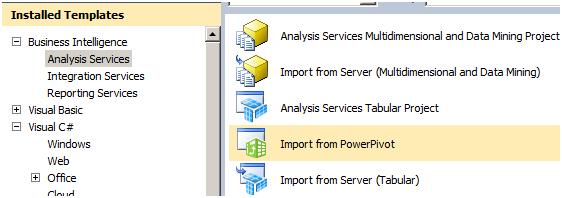
At the end of the day both personal BI and SSAS will publish in a CUBE format. So you can view the data from CUBE using SSRS , EXCEL or any other mechanism.

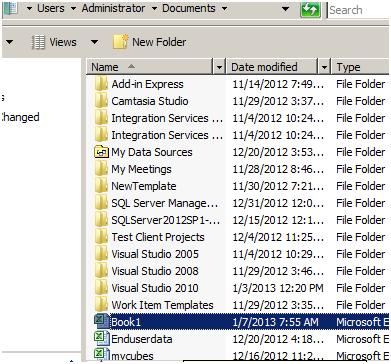


So the personal BI user can use power pivot to do analysis. He can then save the same as an simple EXCEL file.

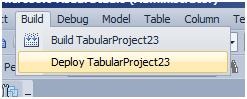


You can then select import from power pivot, go to power pivot EXCEL file and deploy the same in a tabular format.

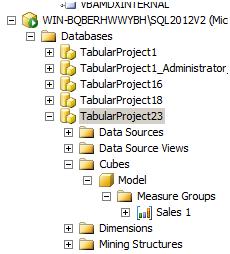




To publish the same to tabular you can click on Build – Deploy tabular project name.

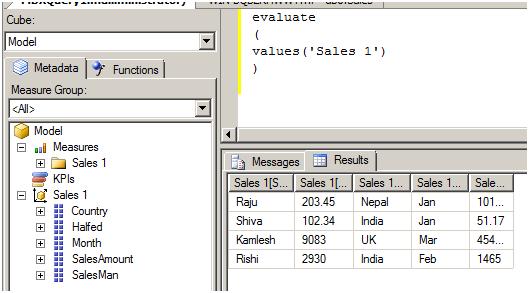


Once deployed you should see the CUBE deployed in SSAS as shown in the below figure.



Because the CUBE is created from tabular format we cannot use MDX to query the CUBE. No worries, a new simple query language have been introduced called as DAX (Data analysis expression). You can see in the below figure how I have queried the “Sales 1” cube. DAX query starts with evaluate keyword, brackets and then the cube name.

This article will not go in to DAX as our main concentration is SQL Server 2012 new features.

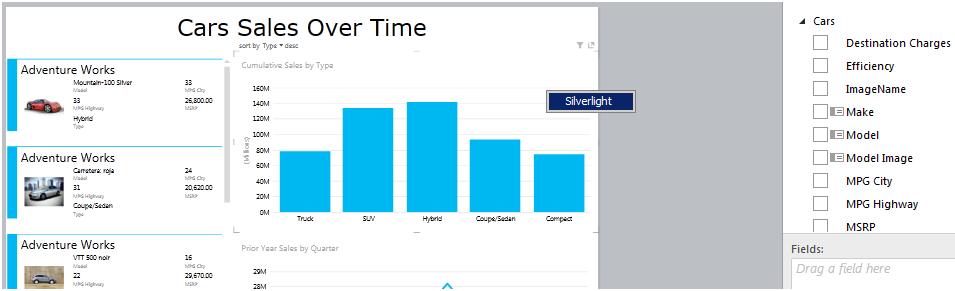


Feature number 9 (Revolution):- Power view

Every second project I have worked in my life always wanted a system where in end users can go and create their own custom reports. Even though we have a facility in SSRS for adhoc reporting it has huge limitations like you need to install something on the client, works only with windows operating system and internet explorer etc.

Power view is created for simple end user who would like to drag and drop and create their own report using ad-hoc ways. It’s a simple Silverlight plugin which gets downloaded and you get a screen something as shown below. End users can now drag and drop the fields from right hand side, create a report and publish it. Please note end users can not add fields that have to be added from SSRS or Power pivot.

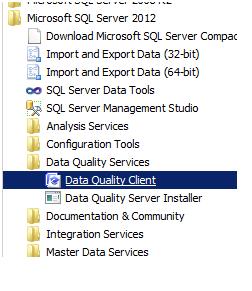
This feature would have been my top feature but due a serious limitation it is not. “Power view only works with SharePoint”….I am sure you are feeling hurt like me. Hope Microsoft makes this independent of share point.



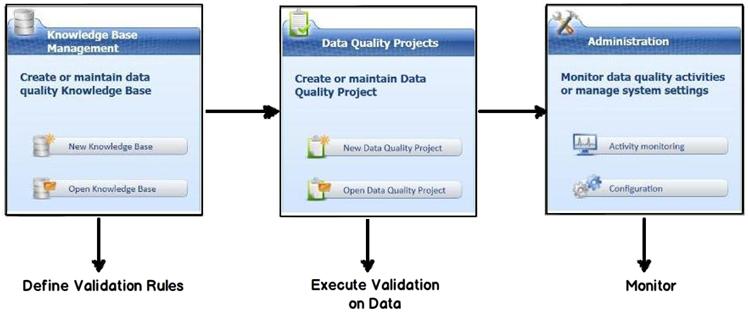
If we visualize properly you can understand what the end GOAL of Microsoft is to empower simple business users so that can do BI themselves. So a personal BI user cannot get data in EXCEL, do analysis by using Power pivot and finally create reports using the ad-hoc reporting tool power view.

Feature number 10 (Revolution):- DQS Data quality services

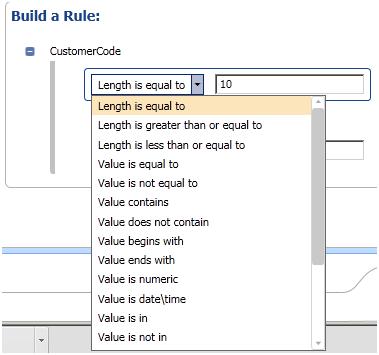
This feature really touched by heart. When we talk about business intelligence it’s all about DATA, DATA and DATA. One of the big problems with data is that it can come in crude and unpolished formats. For instance if someone has entered “IND” and you would like to change it to “India” so that data is in a proper format.DQS helps you build a knowledge base for your data and you can then use this knowledge base to do data cleaning. You can locate DQS as shown in the below image.



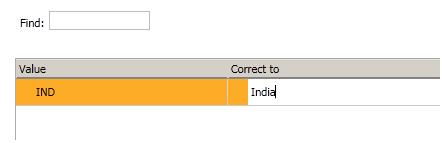
Once you open DQS you will find three sections as shown below Knowledge base, Data quality projects and Administration.



Knowledge base will help you define your validation rules. For instance you can see in the below figure how we are creating a validation called as “CustomerCode” and this validation checks if the data length is equal to 10.



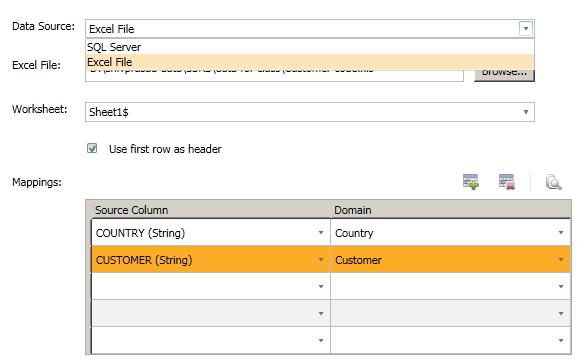
You can also define correction rules like as shown one below. If you find data as “IND” change it to “India”.



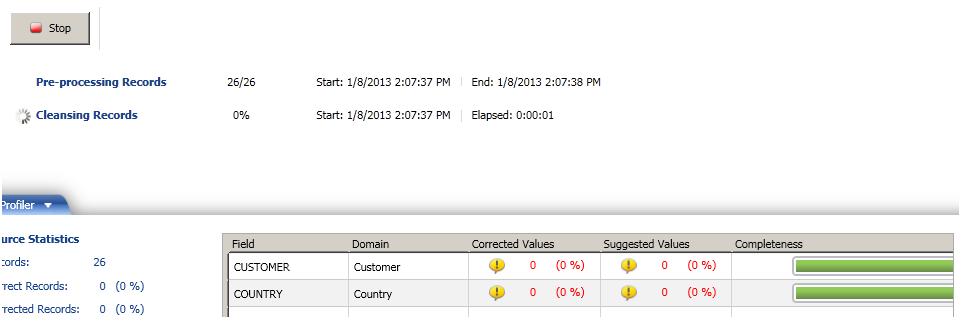
Once you have defined you knowledge, next step is to run this knowledge base over a data. So create a DQS project and apply the knowledge base which you had created as shown in the below figure.



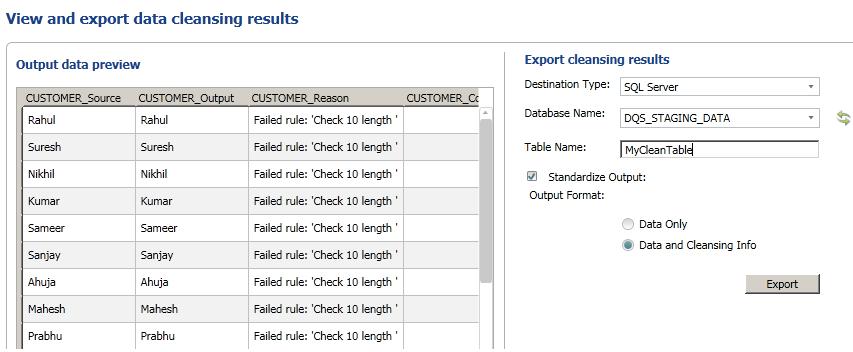
You can then define where the data can come from and also you can map which columns can have which validations. For instance you can see in the below screen for country and customer we have mapped different domains. Domains are nothing but validation rules.



Once done you can start the process and you would see a progress screen as shown below of corrected values and suggested values depending.



Finally you can export the cleaned data to SQL Server, Excel or CSV.



Next 5 features I will discuss about Always on, Trouble shooting in SSIS, Varying columns, SSIS CDC support and SSIS parameter support.

If you ever get a chance, do visit my site which has some awesome collection of [SQL Server Interview question and answer](http://www.questpond.com/) videos. I would also like to celebrate with everyone the 10000 subscriber mark we reached on our youtube channel which is dedicated for [C# and .NET interview questions](http://www.youtube.com/dnfvideo).