***How to solve common Data Quality Problems using Data Quality Services (Part 1)***

**[[http://www.solidq.com/_layouts/images/pdficon_small.gif](http://www.solidq.com/sqj/JournalDocuments/Business_Intelligence/How%20to%20solve%20common%20DQ%20Problems%20with%20DQS%20%28Part%201%29/Part%201%20How%20to%20solve%20common%20Data%20Quality%20Problems%20using%20Data%20Quality%20Services.pdf)Download a PDF of this Article](http://www.solidq.com/sqj/JournalDocuments/Business_Intelligence/How%20to%20solve%20common%20DQ%20Problems%20with%20DQS%20%28Part%201%29/Part%201%20How%20to%20solve%20common%20Data%20Quality%20Problems%20using%20Data%20Quality%20Services.pdf" \t "_blank)**

Poor data is bad for business. Some reasons to have data quality initiatives in a company are as follows:

1. 1. **[Projects](http://searchbusinessanalytics.techtarget.com/news/2240022618/Survey-Poor-data-quality-most-common-business-intelligence-problem)** can be delayed (or declared as failure) because business users saw “poor” data in Business Intelligence deployments. In fact, poor data quality has been reported as the single most BI problem hindering their deployments.
2. 2. Companies risk losing money from poor quality data. In a report "**[Data Quality and the Bottom Line](http://download.101com.com/pub/tdwi/Files/DQReport.pdf)**" published in 2002 by TDWI in as much as 10$ million per year.

To that end, in this paper, I present to you how you can use SQL Server 2012’s Data Quality Services to solve common data quality problems.

I’ll cover following topics in this two -part series:Part 1:    A. Data Standardization    B. Identifying and correcting unrealistic or invalid values    C. Validating and correcting records using Regular ExpressionsPart 2:    D. Validating and correcting value in a field based on value in another field(s) in the same row    E. Reduce data duplication    F. Cleaning records using

Before we begin, just want you to note that in real world, the requirements would not be as straight forward as I have discussed in the tutorial. This is so because the article is written with a goal of becoming a reference for IT professionals or Data stewards who are looking for examples that shows how to solve common problems using Data Quality Services. I have kept the demos and technical discussion at **level 200**.And I am assuming that you have knowledge about basic features and concepts of Data Quality services and in this article, I’ll show demo’s that extends your knowledge to reach to the intermediate level so that you are empowered to build solutions with Data quality services in your organization. And I encourage you to follow along as we discuss each technique, data files required for following the tutorials can be found **[here](http://www.solidq.com/sqj/PublishingImages/DQSMaterialsPart%201.zip" \o "Source files)**.

Also, you’ll need SQL Server 2012 along with installation of Data Quality Services to try out examples on your machine. If you’ve not installed Data Quality Services, you can find the instructions here:[**http://msdn.microsoft.com/en-us/library/gg492277.aspx**](http://msdn.microsoft.com/en-us/library/gg492277.aspx)Withthat, here are the solutions to the common problems that I listed before:

**Data Standardization**Scenario:

We’ve customer data coming in from two different source systems. In one system, the values in the Gender column has values Male & Female, while in the other system, the values in the Gender column has values M & F. We want to correct this data inconsistency issue and standardize the values in the Gender column.Solution:

Let’s see how SQL server 2012′s data quality services would help us solve this problem:

1. 1.  Open DQS client, Connect to DQS server

1. 2.  Under Create or Maintain Data Quality knowledge base, click on **New Knowledge Base**

1. 3.  Enter the name, Description

1. 4.  Under Select Activity, choose **Knowledge Discovery**

1. 5.  Select the Data Source as **Excel File**, Browse and open the **CustomerInformation.xml**file. Leave the worksheet value to the default. And make sure that the Use First row as header is checked.   
   Note: The Full Names in the data set that I shared are imported from the AdventureWorks Sample Database provided by Microsoft.

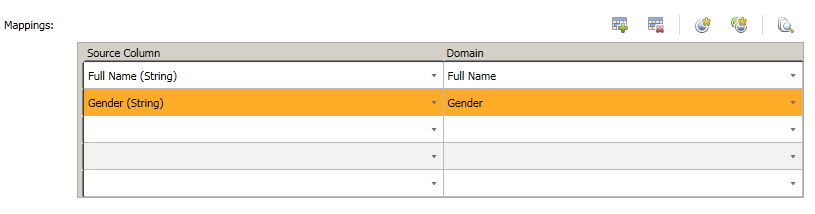
1. 6.  Select Full Name and Gender under **Source Column**

1. 7.  Create a domain with domain name Full Name and leave all the values to default

1. 8.  Map the Full Name under source column to domain that you created in previous step named Full Name

1. 9.  Create a domain with domain name Gender and leave all the values to default

1. 10. Map the Gender under source column to domain that you created in previous step named Gender

1. ****

1. **Figure 1: Mapping Source Columns to Domains**

1. 11.  Click on **Next**

1. 12.  Click on **Start**

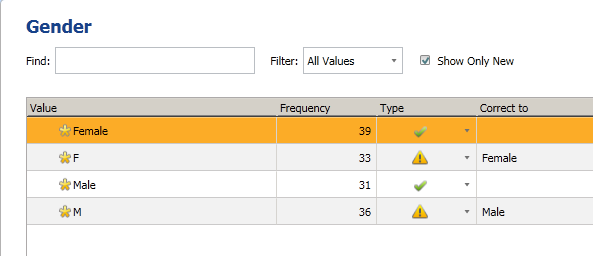
1. 13. ​ In the Profiler statistics displayed note that the Gender Column has 4 unique values while we were expecting 2 unqique values i.e. Male & Female. This the profiler statistics has inted to us that we should be further investigating this column.

1. 14.  Click on **Next**

1. 15.  Select the Domain **Gender**to see the values in the source column

1. 16.  Change the Type of value F and M to**invalid**

1. 17.  Correct the values F and M to Female and Male respectively

1.   
   Figure 2. Configuring Domain Values in Data Quality Services

1. 18. Click on **Finish** and publish the **Knowledge Base (KB)**

1. 19. We are going to use the KB that we created in previous steps in a Data Quality Project

1. 20.  Under Create or maintain Data Quality Project, click on **New Data Quality Project**

1. 21.  Enter the **Name** and select the **KB** that we created in previous steps

1. 22.  Under Select Activity, select **Cleansing**

1. 23.  Click Next.

1. 24.  Select Excel File as the Data Source and browse to the [**CustomerInformation.xls**](http://www.solidq.com/sqj/JournalDocuments/Business_Intelligence/How%20to%20solve%20common%20DQ%20Problems%20with%20DQS%20%28Part%201%29/CustomerInformation.xls) file and leave the worksheet value to Sheet1$

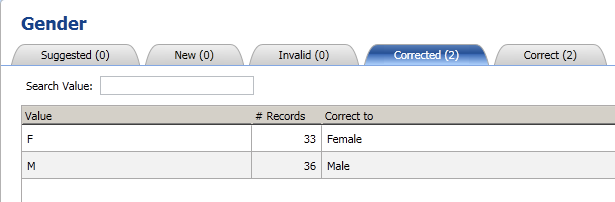
1. 25.  Map Source Column Full Name to Domain Named Full Name, and map Source Column named Gender to Domain named Gender, click next

1. 26.  Click on start

1. 27.  Once the Analysis of Data source has been completed successfully, click on next

1. 28.  Select the **Gender** Domain from the left column

1. 29.  Switch to the **corrected** tab, note that the values F and M were corrected to Female and Male respectively:

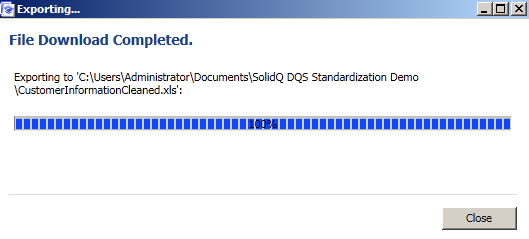
1.   
   Figure 3. Values F and M were successfully corrected to Female and Male

1. 30.  Click next

1. 31.  Under the Destination Type, select Excel; Browse to the Destination location and enter Excel File name (file will be created if it does not exist)

1. 32.  Select **data only** for output format and click on **Export**

1. 33.  You'll get a notification when the file download completes:

1.   
   Figure 4. File Download Completed

1. 34.  Now, , we want to use the KB that we had built earlier in a SQL Server Integration Services (SSIS) Project. This would help you understand the integration of Data Quality Services (DQS) integration with SSIS. For official resource on DQS cleansing Transformation in SSIS, read:[**http://msdn.microsoft.com/en-us/library/ee677619.aspx**](http://msdn.microsoft.com/en-us/library/ee677619.aspx)

1. 35.  Open SQL Server Data Tools (SSDT) and start a New **Integration Services Project**

1. 36.  Drag and drop the Data Flow Task to the Control Flow in the SSIS designer

1. 37.  Rename the Data Flow Task to **Receive and Standardize data using DQS**

1. 38.  Double click the Data Flow Task to switch to the **Data Flow** Tab

1. 39.  Drag the Excel Source Task to the Designer and configure it to connect to[**CustomerDataFromSourceSystem1**](http://www.solidq.com/sqj/JournalDocuments/Business_Intelligence/How%20to%20solve%20common%20DQ%20Problems%20with%20DQS%20%28Part%201%29/CustomerDataFromSourceSystem1.xls)**.**Select Sheet1$ as the Name of the Excel sheet.

1. 40.  Rename this Excel Source Task to CustomerDataFromSourceSystem1

1. 41.  Drag one more Excel Source Task to the Designer and configure it to connect to[**CustomerDataFromSourceSystem2**](http://www.solidq.com/sqj/JournalDocuments/Business_Intelligence/How%20to%20solve%20common%20DQ%20Problems%20with%20DQS%20%28Part%201%29/CustomerDataFromSourceSystem2.xls)**.**Select Sheet1$ as the Name of the Excel sheet.

1. 42.  Rename this Excel Source Task to CustomerDataFromSourceSystem2

1. 43.  Drag **Union All** Task to the designer and Rename it to **Combine Data From SourceSystem1 and SourceSytem2**

1. 44.  Drag the blue connector from the two excel source tasks to the Union all Task

1. 45.  Drag the **DQS Cleansing** task to the designer. Rename it to **Standardize Data using DQS**

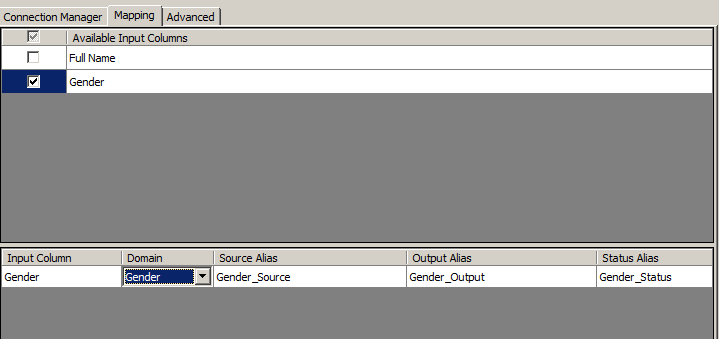
1. 46.  Right click the DQS Cleansing Task, click on Edit to open **DQS cleansing Transformation Editor**

1. 47.  Configure the Data Quality Connection Manager and Data Quality Knowledge Base. Select the knowledge Base that was created in Previous Steps

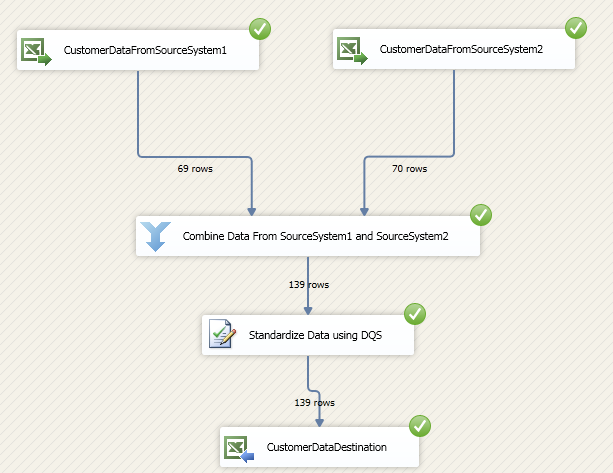
1. 48.  After successfully connecting to the DQS server and selecting the KB, you’ll see the available domains

1. 49.  Switch to the “mapping” Tab in the DQS cleansing Transformation Editor

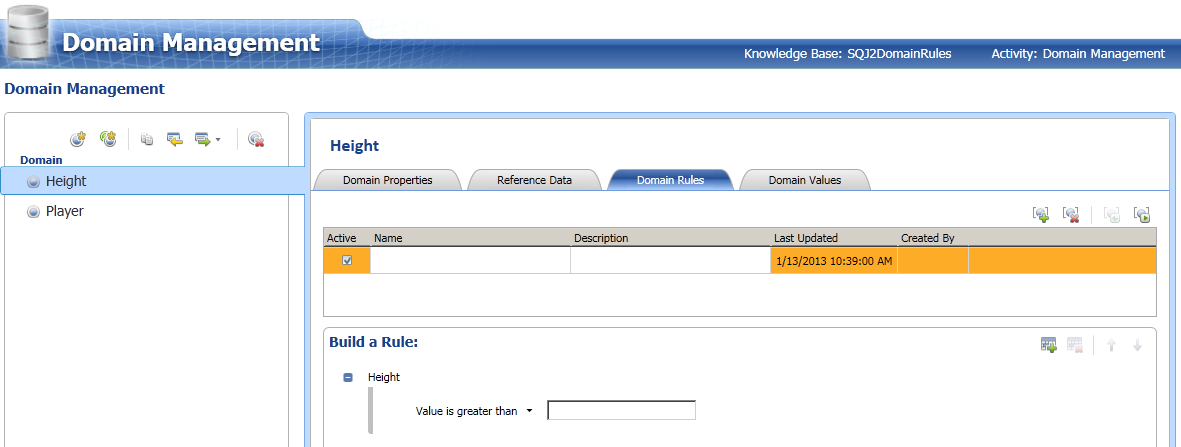
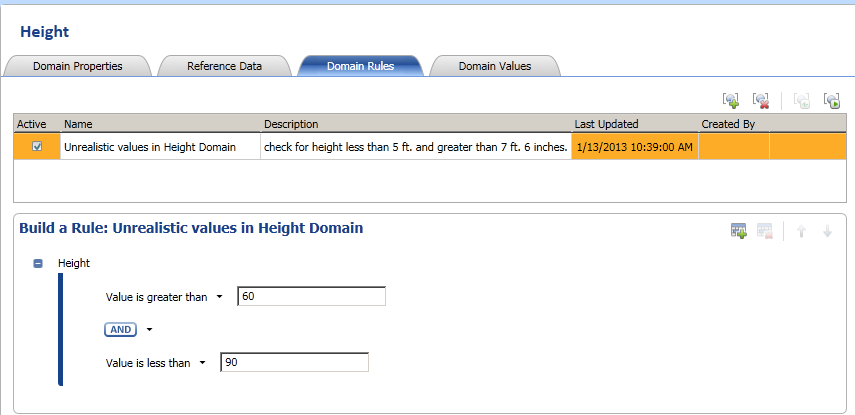
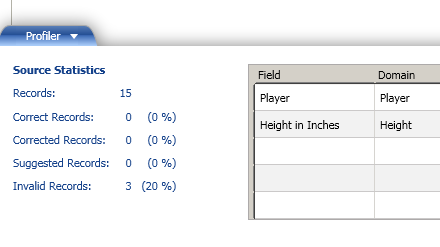
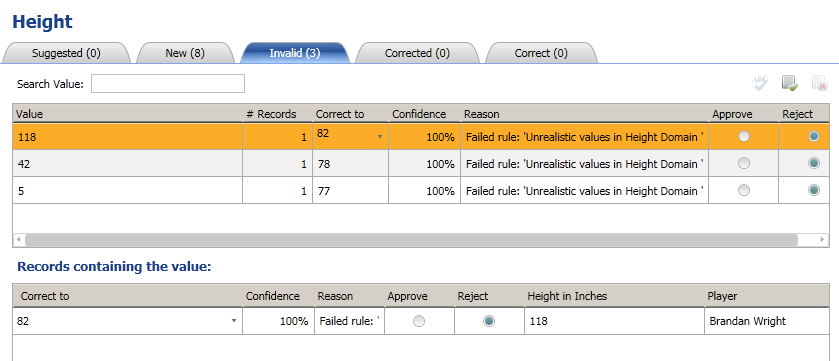
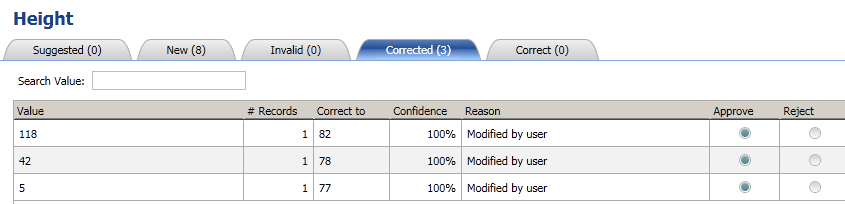
1. 50.  Check **Gender** from the Available Input Columns
2. 51.  Under the Domain select **Gender** for the Input Column Gender

1.   
   Figure 5. Mapping Domain with Input columns in DQS cleansing Transformation editor

1. 52.  Click OK to switch back to Designer
2. 53.  Drag **Excel Destionation** Task to the Designer and rename it to CustomerDataDestination
3. 54.  Drag the blue connector from **Standardize Data using DQS** to the **CustomerDataDestination**
4. 55.  Configure the Excel Destination Task. Connect it to an empty excel file that has two columns named Full Name and Gender. Then after establishing the connection to this file, map the Gender Column in the Destination File with the input column **Gender\_Output** and Map the Destination column Full Name with the Input column Full Name
5. 56.  Start Debugging (F5) to Test the package and see the results:

1.   
   Figure 6. Running SSIS project having a DQS cleansing Transformation
2. 57.  You can deploy the package after successful testing.
3. In this section, we saw how to standardize data that is coming from multiple source systems and take a step closer to achieving "single version of truth".

**Identifying and correcting unrealistic or invalid values**  
  
With Data quality services, Data Stewards can define rules that correct invalid values in the data. Let’s take an example; if you get a data set where a person’s height is 120 inches (10 ft.), you know that there’s probably something wrong about it. For such cases, DQS lets you define rules that detect such outliers. Also by defining such rules in DQS, you’ve automated the process of identifying and correcting invalid values from the data.   
  
With that, here’s an example to create a Domain Rule that flags height values that are not between 5 ft. and 7 ft. 6 inches. Here are the steps:

1. 1.  We will start with the domain management activity of the knowledge base but in real world scenario, typically knowledge discovery activity is performed before the domain management activity. In our example, we know the problem in our data and so we're directly jumping to the domain management activity
2. 2.  Open DQS client > connect to DQS server > click on **New Knowledge Base**
3. 3.  Enter the name of the knowledge base and select **Domain Management** as the activity
4. 4.  Create a domain name **Player** and leave all values to default
5. 5.  Create another domain name **Height** and change the Data Type to **Decimal**
6. 6.  Switch to the **Domain Rules** tab of the Height Domain
7. **Figure 7: Creating a Domain Rule in Data Quality Services**
8. 7.  Here, based on the business requirement, we are going to create a domain rule that checks for player heights that's less than 5ft and greater than 7ft 6 inches for review.
9. Name the domain rule: Unrealistic Values in Height Domain.
10. Description of the domain rule: check for height less than 5 ft. and greater than 7ft. 6 inches.
11. Note: 1 ft. = 12 inches.
12. Here is the domain rule:
13. **Height**
14. Value is **greater than** 60
15. **AND**
16. Value is **less than** 90
17. 
18. **Figure 8: Configuring a Domain Rule in Data Quality Services**
19. 8.  Click **Finish** and Publish the KB  
    9.  Click on New Data Quality Project
20. 10.  Enter name, select the knowledge Base created in previous steps and select **Cleansing** as the activity. Click Next
21. 11.  Choose the database to cleanse against the mapped domains
22. Select Excel File as the data source
23. Select **[Dallas Mavericks Player Rooster.xlsx](http://www.solidq.com/sqj/JournalDocuments/Business_Intelligence/How%20to%20solve%20common%20DQ%20Problems%20with%20DQS%20%28Part%201%29/DallasMavericksPlayerRoster.xls)** as the Excel File
24. Map Source Column named height in Inches to Height Domain
25. Map Source Column named Player to Player Domain
26. 12.  Click next
27. 13.  Click on **start**
28. 14.  Notice that DQS provides statistics via Profiler. Under source statistics, it says that 3 values are invalid:
29. **Figure 9: Profiler Statistics in Data Quality Services**
30. 15.  Click next
31. 16.  Select the Height Domain
32. 17.  Switch to the **Invalid** tab
33. 18.  You can enter the valid values in the **correct to** field
34.   
    **Figure 10. Viewing the Result of running a Domain Rule on set of records**
35. 19.  After entering the values in the **correct to** field check**Approve** for all three invalid values
36. 20.  Switch to **corrected** tab to see the changes that we just made
37. 

**Figure 11. Invalid records corrected to valid values**

1. 21.  Click next22.  Note that the column **Height\_in\_inches\_output** contains the cleaned records23.  Here you can choose to export the results to the desired destination

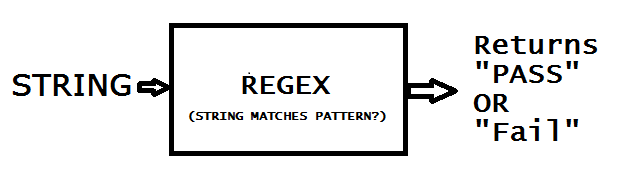
Note that you can re-use the knowledge base (KB) that you had created in above steps for other Data Quality Projects too.

**Validating and correcting records using Regular Expressions**

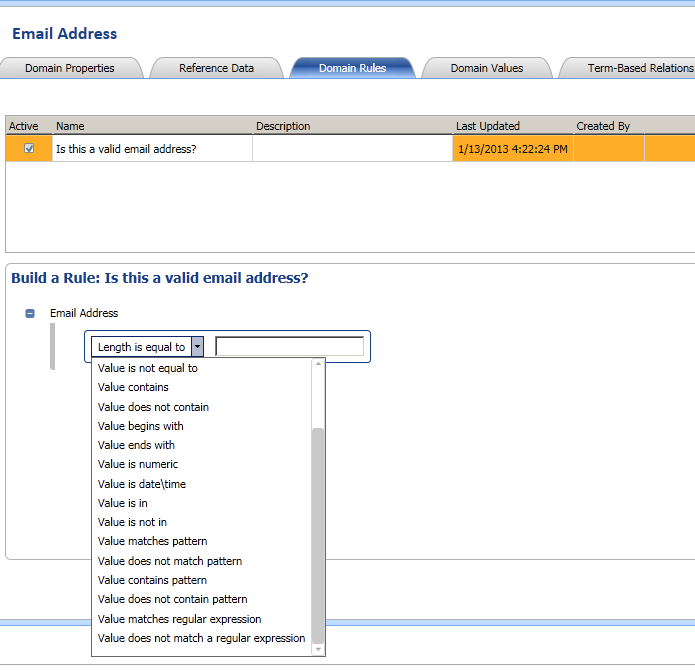
In the previous section, we used domain rules to check for unrealistic values by using the greater than and less than operators. But conditions in domain rules for other data types like string is not limited to greater than, less than, equal to; there are conditions like value begins with, value is date/time, value contains, value matches regular expression among other conditions. The official reference can be read [**here**](http://msdn.microsoft.com/en-us/library/hh510397.aspx); in this section we'll explore one of these conditions. We'll see how to clean records using regular expressions in a domain rule. For demo purposes, we'll perform preliminary cleansing activity on email records.

Before we begin, here's a primer on Regular Expressions:

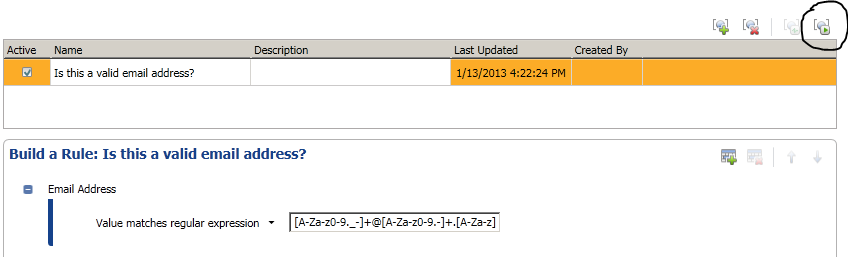
Regular expressions (also known as regex or regexp) is a language used to match strings to patterns. Here's a visual:

  
**Figure 11. Invalid records corrected to valid values**

So, if we defined a pattern that asked "is this the valid email address?" the regular expression would retunr PASS or FAIL. So let's try creatinga regular expression that performs a preliminary email address check:

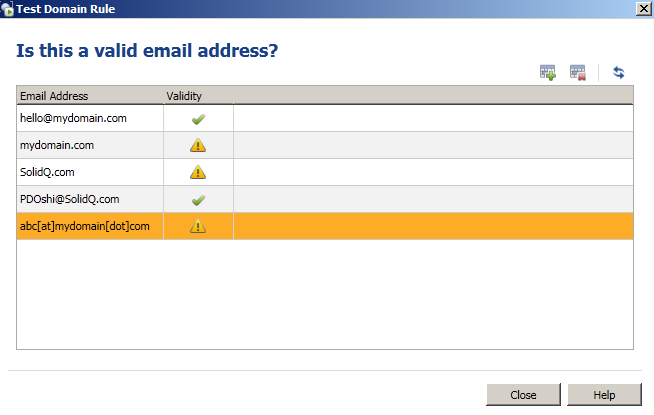
1. 1.  Open DQS client, connect to DQS Server, click on **New Knowledge Base**
2. 2.  Enter Name and select **Domain Management** from the activity
3. 3.  Create Domain, name it **Email Address** and select**String** as the Data Type
4. 4.  Switch to the **Domain Rules** tab
5. 5.  Add a new Domain Rule
6. 6.  Name the Domain Rule:**Is this a valid email address?**
7. 7.  Under the Build a Rule section, select **value matches regular Expression**
8. 

**Figure 13. Value matches regular expressions**

1. 8.  Use this regular expression for the purpose of the demo: [A-za-z0-9.\_-]+@[A-Za-z0-9.-]+.[A-Za-z]
2. Note that the result you get will be as good as your regular expression. After all, regular expressions are like "programs" and so the "Quality" of your regex depends on how well you write it.
3. 9.  You can test the domain rules while you are creating them. To do so, click on **Run the selected domain rule on test data**
4. 

**Figure 14. Run a Domain Rule on Test data**

10.  Click on Adds as a new testing term for the domain rule

1. 11.  Enter a few records to validate them against the domain rule, for this demo, I am going to add:
2. hello@mydomain.com
3. mydomain.com
4. solidq.com
5. PDoshi@SolidQ.com  
   Abc[at]mydomain[dot]com
6. 12.  After adding the terms, click on **Test the Domain Rule on all term**
7. **Figure 15. Is this a valid email address**

13.  Click on **Close**

1. 14.  Click on Finish and Publish the KB.
2. 15.  You can use this KB to validate email addresses in your data sets. We're not going to start a Data Quality Project for this section.
3. 16.  This was just a basic demo to show how you can use Regular Expressions to detect values that do not match a pattern. In real world scenario, for cleaning Email records, there are other options like using third-party data sets or writing an advanced regular expression. The RegEx I shared was just meant for demo but if you want to read more about Regular Expressions used to clean emails then I strongly recommend you to read **[email regular expressions](http://www.regular-expressions.info/email.html)**.

Note: We're going to see how to use third party reference datasets in part 2 of this series.

**Conclusion**

In this article, we saw how to solve some of the common data quality problems using SQL Server 2012 Data Quality Services. We saw:

1. 1.  How to standardize data
2. 2.  Learned how to use DQS cleansing transformation in SSIS
3. 3.  Learned to build domain values and domain rule

In next article, we will continue our journey and we'll learn about:

1. 1.  Validating and correcting value in a field based on value in another field(s) in the same row2.  Reduce data duplication
2. 3.  Cleaning records using reference data-set from Windows Azure Data market

**Resources**

[**Survey: Poor data quality most common Business Intelligence problem**](http://searchbusinessanalytics.techtarget.com/news/2240022618/Survey-Poor-data-quality-most-common-business-intelligence-problem)  
[**Data Quality and Bottom Line**](http://download.101com.com/pub/tdwi/Files/DQReport.pdf)  
[**Data Quality Services concepts**](http://msdn.microsoft.com/en-us/library/hh213015.aspx)  
[**Create Composite Domain**](http://msdn.microsoft.com/en-us/library/hh510414.aspx)  
[**Create a Cross-Domain Rule**](http://msdn.microsoft.com/en-us/library/hh510390.aspx)  
[**Using DQS: Cleansing complex data using composite domains**](http://social.technet.microsoft.com/wiki/contents/articles/13324.using-dqs-cleansing-complex-data-using-composite-domains.aspx)  
[**Official resource on DQS cleansing Transformation in SSIS**](http://msdn.microsoft.com/en-us/library/ee677619.aspx)  
[**Using DataMarket with SQL Server Data Quality Services**](https://datamarket.azure.com/help-using-marketplace-with-data-quality-services)  
[**Create a Domain Rule**](http://msdn.microsoft.com/en-us/library/hh510397.aspx)  
[**How to Find or Validate Email Address**](http://www.regular-expressions.info/email.html)  
  
 **About the Author**

|  |  |
| --- | --- |
| sqlint_pdoshi_LThumb.jpg | **Paras Doshi** ([**twitter**](http://www.twitter.com/paras_doshi) | [**blog**](http://www.parasdoshi.com/))  Paras is a Microsoft Community Contributor awardee and a Blogger. He is passionate about helping businesses extract insights from data. To that end, He has worked on projects that involved buidling various components of a Business-Intelligence /Analytics solution for clients. |

**About the Technical Reviewer**

|  |  |
| --- | --- |
| http://www.solidq.com/SiteCollectionImages/EmployeePhotos/sqlint_vsanchez_LThumb.png    ​ | ​  **Víctor Sánchez**([**twitter**](http://www.twitter.com/atharky) | [**blog**](http://bifase.blogspot.com.es/)) is an experienced Data Platform Architect working in the Business Intelligence division and collaborating in projects globally. He began working with ERP products in financial areas 10 years ago, focused on development with Microsoft technologies: Visual Basic and SQL Server 2000. He has been involved in data modelling, integration and analysis through all later versions of SQL (2005/2008/R2/2012). Prior join SolidQ he has been in charge of the area of BI managing development teams. He have strong technical skill with business focus and able to management. Along his career were taking knowledge of different industries like Banking, Transport, Energy and Public Administration. He has spoken at various events for clients, backed by Microsoft and designed courses about Integration Services, Reporting Services and Excel. Follow him also on the corporate blog [**http://blogs.solidq.com/BICorner**](http://blogs.solidq.com/BICorner). |