

**Oracle® Data Profiling and
Oracle Data Quality for Data
Integrator**

Sample Tutorial
10g Release 3 (10.1.3)

November 2007

Copyright © 2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table of Contents

Introduction to Oracle Data Quality Products	1
Oracle Data Quality Products	1
Tutorial Contents	1
Recommended Readings	1
Prepare for the Tutorial	3
Install Oracle Data Quality and Data Profiling	3
Configure the Metabase and the Connections	3
Setup the Data Files	6
Install the Postal Directories	6
Preload the Metabase	7
Oracle Data Profiling Tutorial	11
Investigate Data	11
Explore Relationships within Entities	13
Explore Existing Keys and Find Alternate Keys	13
Examine Dependencies	14
Explore Relationships between Entities (Joins)	14
Check Data Compliance	18
Apply Business Rules	20
Oracle Data Quality for Data Integrator Tutorial	22
Design a Name and Address Cleansing Project	22
Run the Quality Project in ODI	32
Going Further with Oracle Data Quality for Data Integrator	32

Introduction to Oracle Data Quality Products

Oracle Data Quality Products

Oracle Data Quality Products - **Oracle Data Profiling** and **Oracle Data Quality for Data Integrator** - extend the inline Data Quality features of **Oracle Data Integrator** to provide more advance data governance capabilities.

Oracle Data Profiling is a data investigation and quality monitoring tool. It allows business users to assess the quality of their data through metrics, to discover or infer rules based on this data, and to monitor the evolution of data quality over time.

Oracle Data Quality for Data Integrator is a comprehensive award-winning data quality platform that covers even the most complex data quality needs. Its powerful rule-based engine and its robust and scalable architecture places data quality and name & address cleansing at the heart of an enterprise data integration strategy.

Tutorial Contents

This tutorial guides you through a first project involving data profiling and data quality.

You will first Prepare for the Tutorial by configuring a new installation of the Oracle Data Quality products for running projects.

The Oracle Data Profiling Tutorial section will guide you through an investigation process on a set of files to detect data anomalies and inconsistencies, and create new business rules on this data.

Finally, the Oracle Data Quality for Data Integrator Tutorial section you show you how to create a data quality process cleanse a file containing incorrect and incomplete name and address records.

Recommended Readings

It is recommended that you read first the *Oracle Data Quality for Data Integrator - Getting Started Guide* to have an overview of the user interface, the key concepts and steps for data profiling and quality.

Prepare for the Tutorial

Install Oracle Data Quality and Data Profiling

Refer to the *Oracle Data Integrator Installation Guide* for installing Oracle Data Quality products as well as Oracle Data Integrator.

Configure the Metabase and the Connections

1. Make sure Oracle Data Quality and Data Profiling, as well as Oracle Data Integrator are installed and working.
2. Select **Start > All Programs > Oracle > Oracle Data Profiling and Quality > Metabase Manager** to Log in to the Metabase Manager as the Metabase Administrator (*madmin*)
3. Select **Tools > Add Metabase** from the menu
4. Add a metabase named *oracledq*, with the default pattern and a Public Cache Size of 10 Mb, and then click **OK**.

Add Metabase

Add or edit a metabase. The cache settings define how much server memory is used by this Metabase - the larger these values the better the performance.
Warning: Make sure that the total cache for all metabases does not exceed the available server memory or performance will decrease rapidly.

Name:

Default Pattern:

Public Cache Size (in Megabytes):

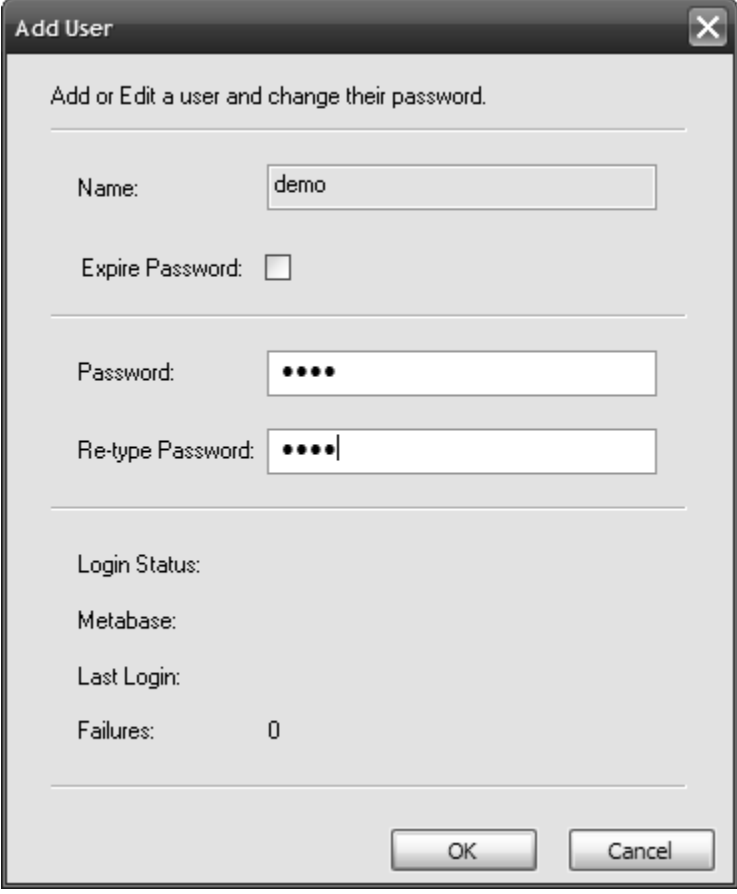
Created By:

Created Date:

Edited By:

Edited Date:

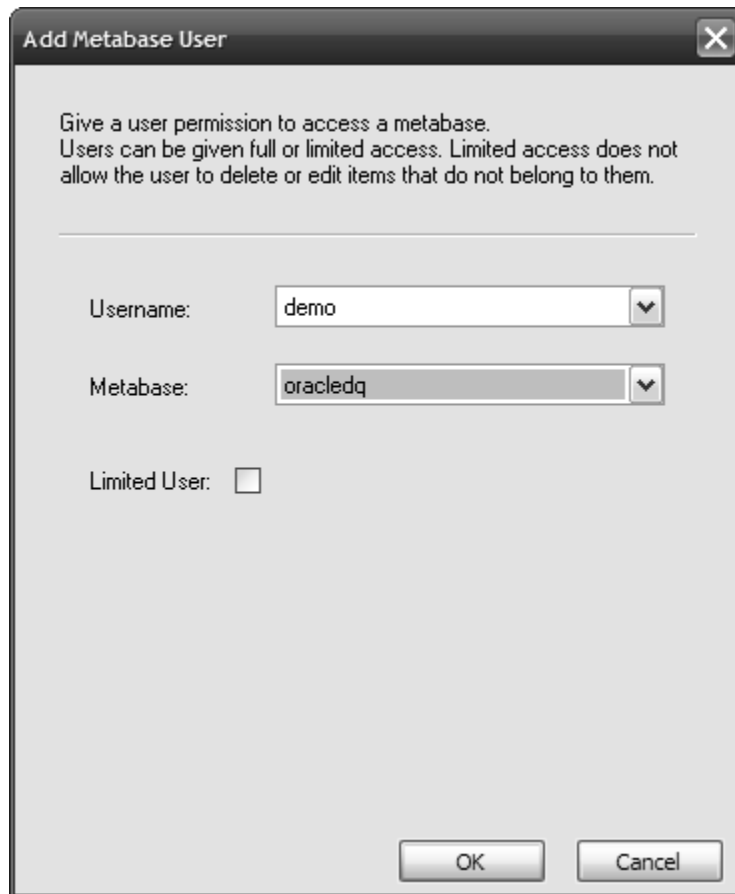
5. Select **Tools > Add User** from the menu
6. Add a User named *demo* with the password *demo*, as shown below, then click **OK**.



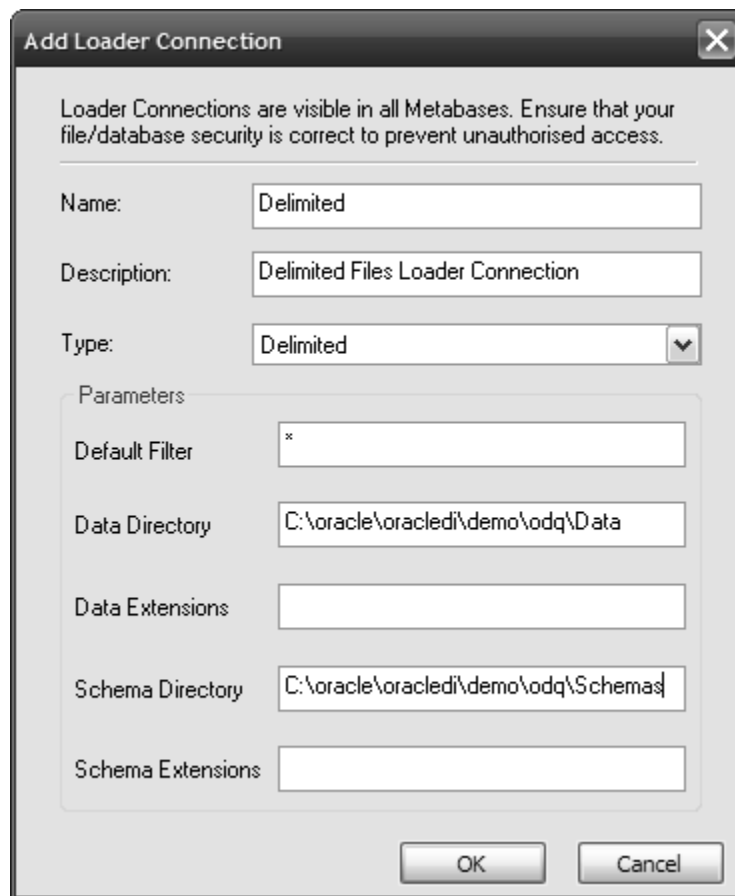
The image shows a 'Add User' dialog box with a title bar containing a close button (X). The dialog contains the following fields and controls:

- Instruction: 'Add or Edit a user and change their password.'
- Name: A text field containing 'demo'.
- Expire Password: A checkbox that is currently unchecked.
- Password: A text field with four dots (••••).
- Re-type Password: A text field with four dots (••••).
- Login Status: A label.
- Metabase: A label.
- Last Login: A label.
- Failures: A label followed by the value '0'.
- Buttons: 'OK' and 'Cancel' buttons at the bottom right.

7. Select **Tools > Add Metabase User** to add the *demo* user to the *oracledq* metabase, as shown below, and then click **OK**.



8. Select **Tools > Add Loader Connection**. Create a loader connection for delimited files as shown below.
- Type: Delimited
 - Default filter: *
 - Data directory: <ODI_Home>\demo\oracledq\Data
 - Schema directory: <ODI_Home>\demo\oracledq\Schemas



9. Close Metabase Manager.

Setup the Data Files

1. On your server, make sure the <ODI_Home>\demo\oracledq directory exists. If not create it.
2. Copy and unzip the file named *oracledq_sample_data.zip* to <ODI_Home>\demo\oracledq\.

Install the Postal Directories

1. Uncompress the *oracledq_sample_directory.zip* to a temporary directory.
2. Copy the content of this temporary directory into the Oracle Data Quality server directory, in the tables\postal_tables\ sub-directory. Overwrite existing files.

Preload the Metabase

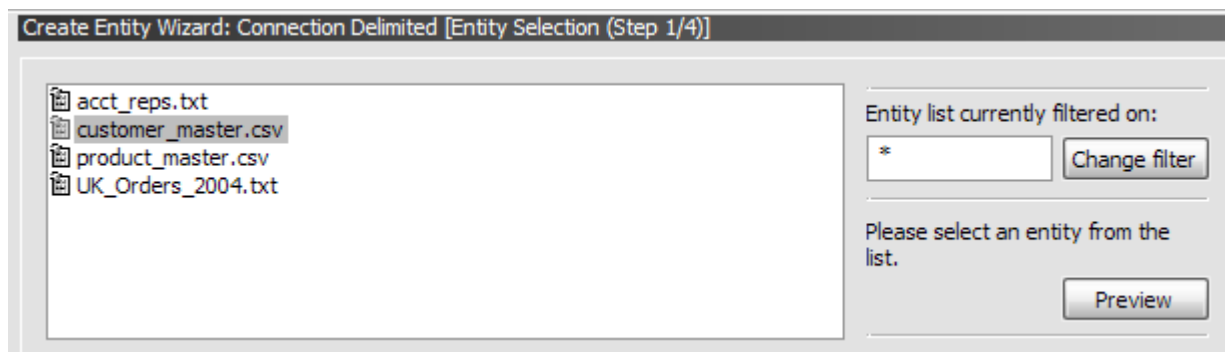
The Metabase contains both the description of the data structures as well as sample data to perform the Data Profiling operations and to design the Data Quality projects. The first step in the quality process is to preload the metabase.

In this sample, we will load the database with the flat files located in the `<ODI_Home>\demo\oracledq` directory, using the delimited and COBOL data loaders defined in the previous chapter. Each of these files has a specific format that we will define when creating entities corresponding to these files.

We first need to create an entity corresponding to the `customer_master.csv` source file with the following parameters:

Source File	File Info	Data Selection	Load Rows
customer_master.csv	delimiter: comma quote: none Names on first line CR/LF terminated: Y Character Encoding: ascii	Keep all data	All

1. Login to Oracle Data Quality client with the following:
 - Repository: primary
 - Metabase: oracledq
 - Username: demo
 - Password: demo
2. Select **Analysis > Create Entity** from the menu.
3. Select the *Delimited* Loader Connection, and then click **Next**.
4. Select the `customer_master.csv` file, and then click **Next**.



5. Set the file info as shown below.

Create Entity Wizard: Connection Delimited [Delimited Schema Settings (Step 2/4)]

Select the appropriate schema settings for this file

Characters: Delimiter: , <input type="button" value="Advanced"/> Quote: NONE <input type="button" value="Advanced"/>	Attribute Information: <input type="radio"/> No Information <input checked="" type="radio"/> Names on first line <input type="radio"/> DDL <input type="button" value="Select..."/>	Misc: Records are CR/LF terminated <input checked="" type="checkbox"/> Character Encoding: ascii <input type="button" value="v"/>
---	---	--

Use the listview column chooser to select your attributes


6. Select **All Rows**, click **Next**, and then **Finish** in the next window.

Create Entity Wizard: Connection Delimited [Confirm Settings (Step 4/4)]

Attributes:	names
Record terminator:	crlf
Encoding:	ascii
Sample settings:	ALL

7. Select **Run Now** in the Schedule job popup window.

Schedule Job

8. Click the background tasks icon in the toolbar () to view the list of running task and wait until the job is complete.

Note: Remember to use this icon to review job completion every time you will start a job with the **Schedule Job** window.

9. Repeat the operation to create Entities using the following information:

Source File	File Info	Data Selection	Load Rows
product_master.csv	delimiter: comma quote: double Names on first line CR/LF terminated: Y Character Encoding: ASCII	Keep all data	All
acct_reps.txt	delimiter: tab quote: double DDL: acct_reps.ddl CR/LF terminated: Y Character Encoding: ASCII	Keep all data	All

uk_orders_2004.txt	delimiter: tab quote: none Names on first line CR/LF terminated: Y Character Encoding: ASCII	Keep all data	All
--------------------	--	---------------	-----

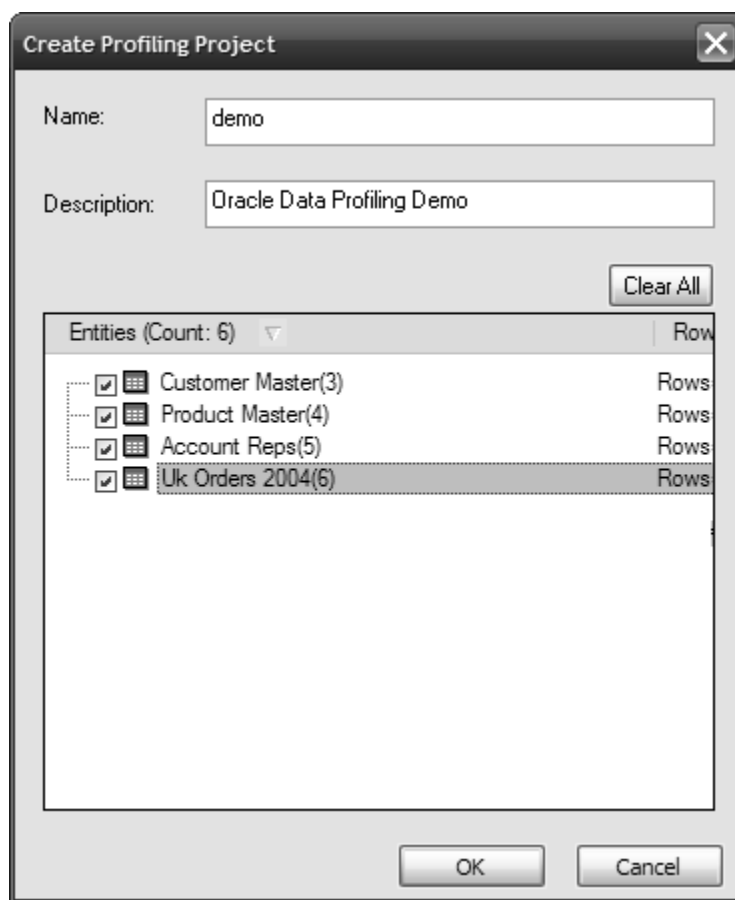
All entities are created for the sources, and loaded with the source data. We can start profiling this data.

Oracle Data Profiling Tutorial

Investigate Data

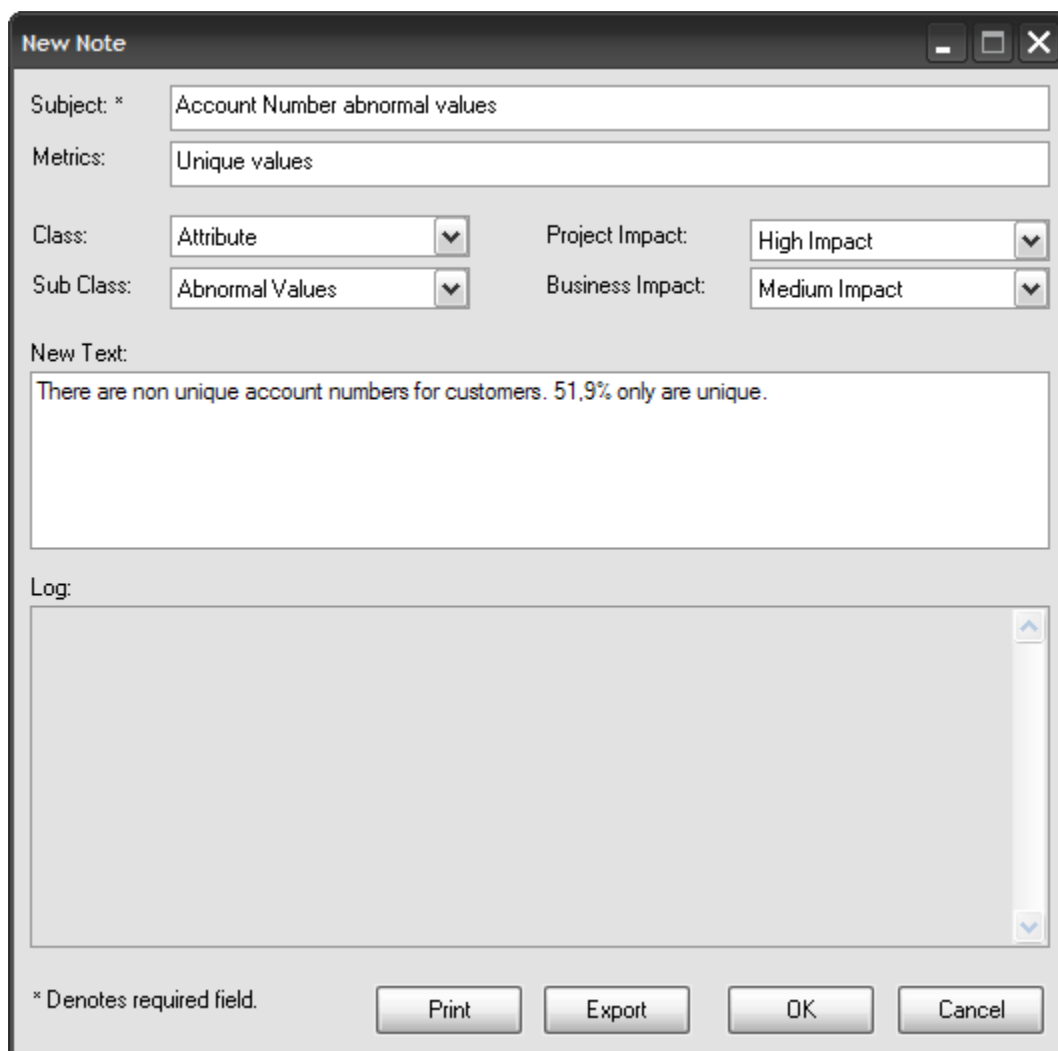
This first profiling step will simply create a project with the four entities previously created. We will then explore one of these entities.

1. Create a Discovery Project named **demo**
 - a. Select **Profiling** in the **Project Types**, right click and select **Create Project** in the popup menu.
 - b. Enter the project name and description, then select all entities as shown below, then click **OK**.



2. Explore Entity level Metadata
 - a. From the Discovery Project **demo**, expand **Customer Master** under the **Entities** folder
 - b. Explore its **Metadata** folder and look at structural metadata such as:
 - Row min len – double-click to see the distribution for the shortest row found, then double click the distribution value to view the list of smallest rows.
 - Row max len – Drill down as above to explore the longest rows.

- Source Type – type of the data source
 - Data Source – name of the data source
 - Load Sampling – sampling method (all rows)
 - Entity Type
 - Rows Loaded – double-click to see all rows for the Entity
3. Explore Attribute level Metadata
 - a. Under **Customer Master**, expand the **Attributes** folder and the Attribute **Account Number**
 - b. Examine Unique Values – notice that not 100% of the values are unique. Several customers exist with the same account number.
 - c. Double-click the Unique Values node to see duplicate values.
 - d. Drill down on a value (double click a row in the table on the right panel) to see rows where similar account numbers occurs
 4. Add a note describing the discovered quality issue
 - a. Right-click the **Account Number** attribute, right-click and select **Notes > Add...**
 - b. Enter the details for your note, as shown below, and then click **OK**.



New Note

Subject: * Account Number abnormal values

Metrics: Unique values

Class: Attribute ▼ Project Impact: High Impact ▼

Sub Class: Abnormal Values ▼ Business Impact: Medium Impact ▼

New Text:

There are non unique account numbers for customers. 51,9% only are unique.

Log:

* Denotes required field.

Print Export OK Cancel

5. Explore the different **Patterns** found for the **Phone** field.
 - a. Under **Customer Master**, expand the **Attributes** folder and the Attribute **Phone**
 - b. Double-click on **Patterns**
 - c. Drill down on pattern values with low frequencies.
 - d. Drill down to the row level to see the rows with a given pattern.

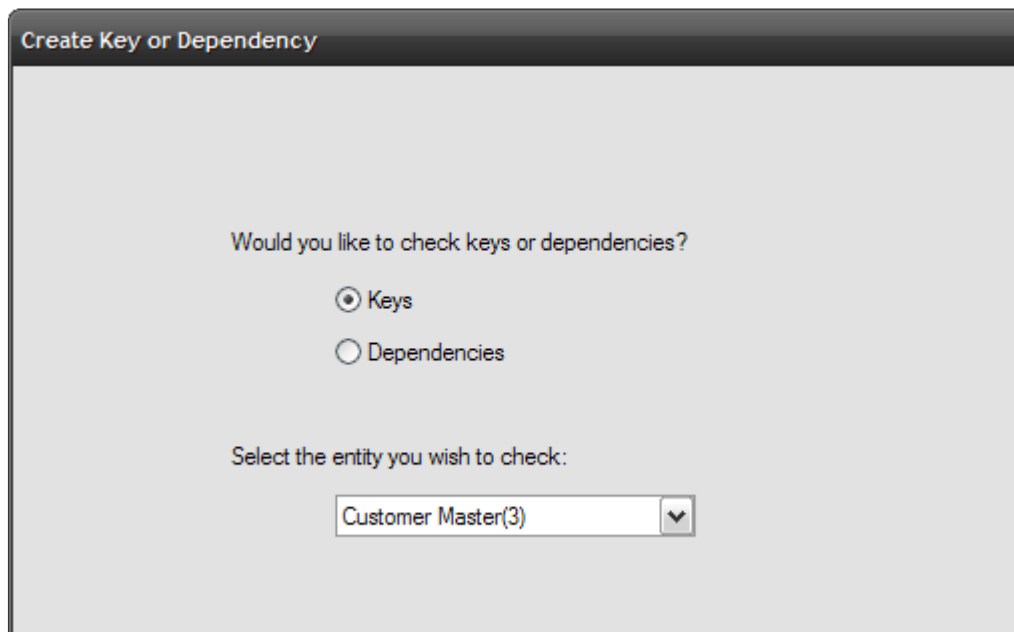
Explore Relationships within Entities

Oracle Data Profiling allows you to profile individual entities as well as relations between group of entities. In this step, we will investigate possible keys for the Customer Master source data, and examine the dependencies between the customer account number and its reference in the UK Orders file.

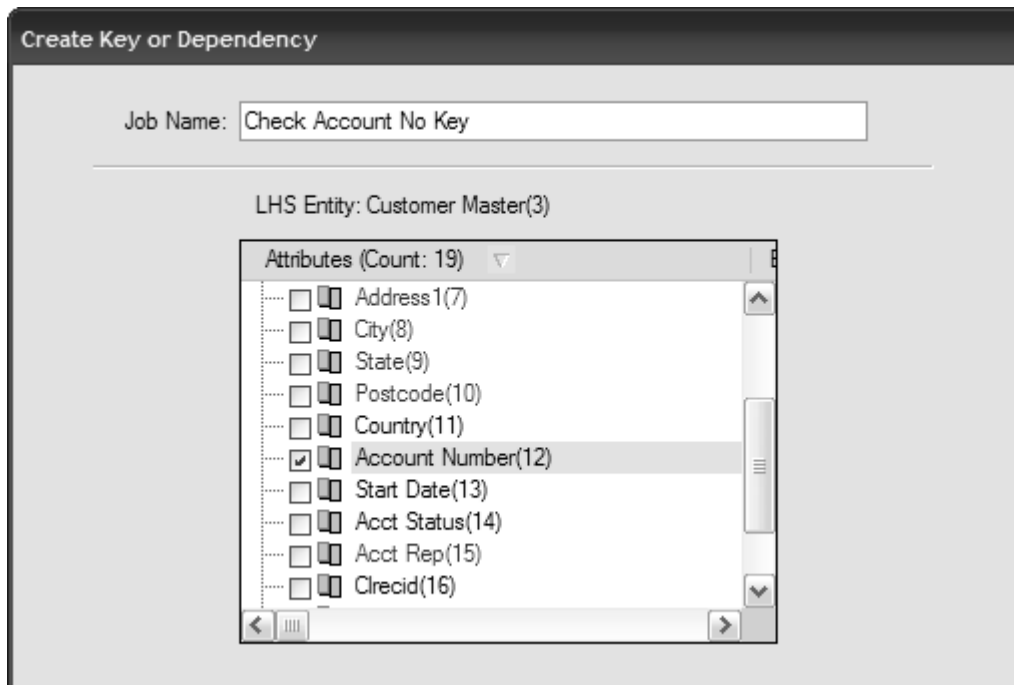
Explore Existing Keys and Find Alternate Keys

There is an implicit key defined on the Customer Master data source, the *Account Number*. We will now examine its validity as a key, and evaluate another column as a possible key;

1. Double-click **Keys** in the Metadata folder for **Customer Master**
2. **Account Number** should be a key field in this data, but is not showing in the list due to its low uniqueness.
3. Make **Account Number** a key through Create Key feature.
 - a. Select **Analysis > Create Key or dependency...** in the menu
 - b. Select the *Customer Master* entity in the list then click **Next**



- c. Name the Job "Check Account No Key", then select the *Account Number* attribute in the list and then click **Finish**.



- d. Click **Run Now** in the Schedule Job window.
4. Drill down to the duplicate values and rows with duplicate values.
 - a. Double-click **Keys** in the Metadata folder for **Customer Master**
 - b. Double-click the *Account Number* key in the table to drill down to the duplicate values
 - c. Double-click of the duplicate values to drill down to the rows with duplicate values.
5. Identify **Clrecid** as a good alternate key.
 - a. Double-click **Keys** in the Metadata folder for **Customer Master**
 - b. Double-click the *Clrecid* key in the table to drill down to the duplicate values
 - c. Double-click the only duplicate value to drill down to the 3 rows with duplicate values.

Examine Dependencies

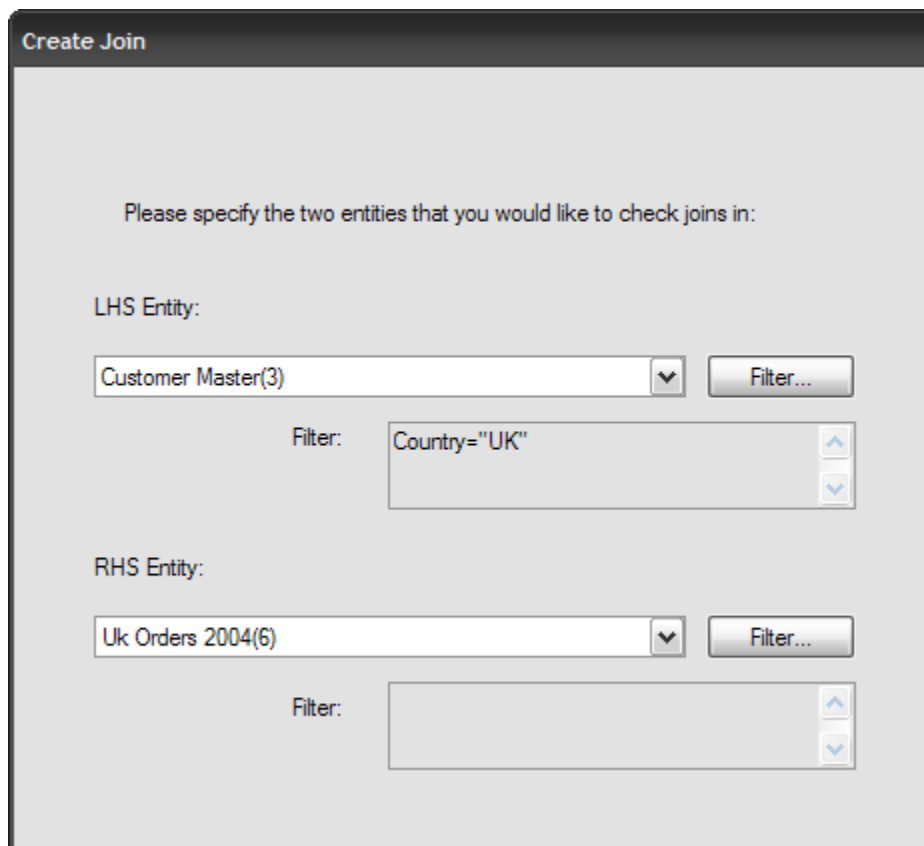
There is a discovered dependency in the Uk Orders 2004 table. An Order ID should have one and only one Account ID associated. We will examine now the potential conflicts on this dependency.

1. Double click the **Dependencies (Discovered)** node in the in the Metadata folder for *Uk Orders 2004*
2. Look at the dependency between **Order Id** and **Account Id**
3. Double click the row showing this dependency to drill down to see the two conflicts (several Accounts sharing one same Order)
4. Double click one row showing a conflict instance to drill down to the rows with the conflicts

Explore Relationships between Entities (Joins)

1. Create a join between Customer Master and UK Orders 2004
 - a. Select **Analysis > Create Join** in the menu.

- b. Select the *Customer Master* and *UK Orders 2004* entities and then apply filter to Customer Master (`Country = "UK"`). Click **Next**.



The image shows a 'Create Join' dialog box with a title bar. Inside, it prompts the user to specify two entities for a join. The 'LHS Entity' section has a dropdown menu showing 'Customer Master(3)' and a 'Filter...' button. Below this, a 'Filter:' label is followed by a text box containing 'Country="UK"' and two small arrow buttons. The 'RHS Entity' section has a dropdown menu showing 'Uk Orders 2004(6)' and a 'Filter...' button. Below this, a 'Filter:' label is followed by an empty text box and two small arrow buttons.

Create Join

Please specify the two entities that you would like to check joins in:

LHS Entity:

Customer Master(3) Filter...

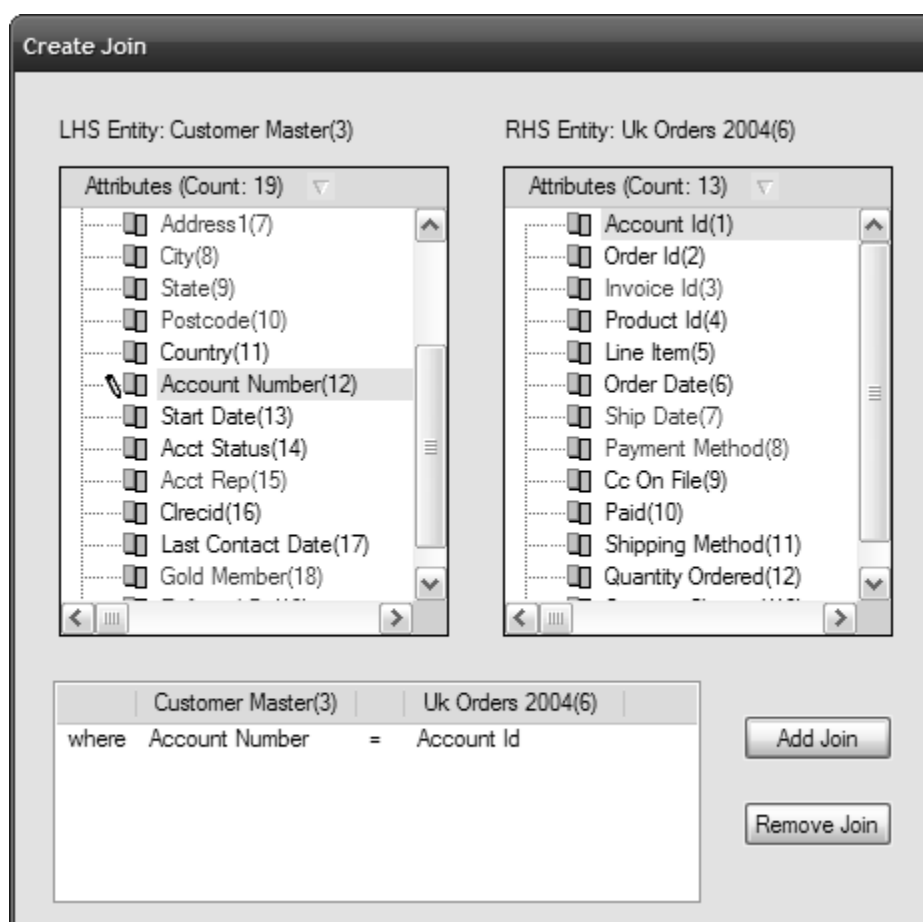
Filter: Country="UK" ↑ ↓

RHS Entity:

Uk Orders 2004(6) Filter...

Filter: ↑ ↓

- c. Join on **Account Number** and **Account Id**, by selecting these attributes under each entity, then clicking the **Add Join** button.



- d. Create the Join as shown below and then click **Finish**.

Create Join

Job Name:

This Join might create more than 755 joined rows. Do you want to create the join index with this number of rows? ☒

DSD Options

Cardinality:

Optionality:

Match Quality:

Documented No Match Actions

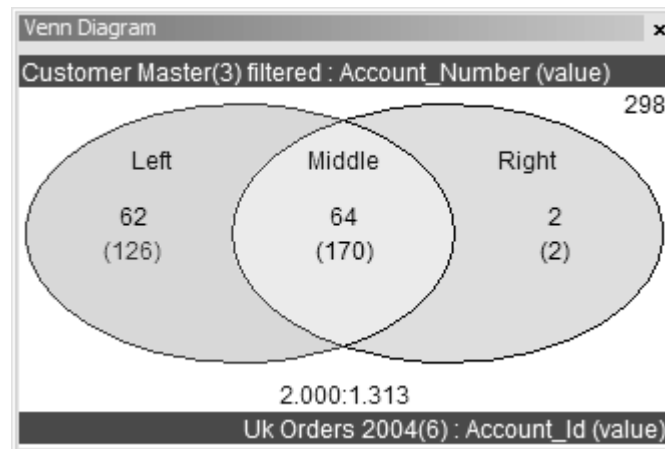
Left:

Right:

Join Result Segment:

Note: This job will create a permanent join by default.

- e. Click **Run Now** in the Schedule Job window.
- f. Expand the **Permanent Joins** node under the *demo* project.
- g. Select the new join to display its properties in the left panel.
 - Examine the number of matching and non-matching values.
- h. Right-Click one of the properties in the list and then select Venn Diagram



- i. Double click sections of the diagram to:
 - Drill down to customers without orders

- Drill down to orders that don't have an Account
 - Drill down to customers that have orders
2. Reproduce the previous steps to create a join between **UK Orders 2004** and **Product Master**
 - a. Join on **Product Id** and **Item Number**
 - b. View Venn diagram
 - Drill down to orders without products
 - Drill down to products that haven't been ordered
 - Drill down to ordered products
 3. Create a join between **Customer Master** and **Acct Reps**
 - a. Join on **Acct Rep** and **Rep Id**
 4. Right-click the **Permanent Join** node under the demo project, and then select **Entity Relationship Diagram**. The following diagram appears.



Check Data Compliance

In the profiling phase, you can check whether the data stored in the source files complies with a set of rules (based on patterns, values, data types, etc). These compliance checks allow you to evaluate the quality of each record.

1. Add the following compliance checks to attributes in **Uk Orders 2004**

Attribute	DSD to apply
Order Id	Pattern Check - Pattern allowed d5 Null check – no null values allowed Acceptable values between 30560 and 32000
Payment Method	Valid Values are CREDIT CARD, EFT, ACCOUNT and COD

- a. Select in the *Demo* Project the **Entities > Uk Orders 2004 > Attributes > Order ID** attribute, right-click, then select **Edit DSD**.
- b. Select the **Pattern Check** tab, enable the test and then enter the d5 pattern to match. Set the tolerance to 0% of rows.

DSD For Uk Orders 2004.Order Id

☒ Sum Check
 ☒ Schema Data Type Check
 ☒ Spaces Check
 ☒ Patterns Check
 ☒ Values Check
 ☒ Null Check

☒ **Passed** (Click [here](#) to disable this test.)
 Drill to [passing](#) or [failing](#) values.

Determine whether specific patterns exist.

Match	Not Match
d5	

 Set the tolerance to be % of

Passes on 98.529% of values.
 Passes on 98.837% on rows.

- c. Select the **Null Check** tab, enable the test, and make sure that no null row is allowed (0%).

DSD For Uk Orders 2004.Order Id

☒ Spaces Check
 ☒ Patterns Check
 ☒ Values Check
 ☒ Null Check
 ☒ Range Check

☒ **Passed** (Click [here](#) to disable this test.)

☒ The maximum permitted percentage of rows with nulls is %.
☐ The maximum permitted number of rows with nulls is .

- d. Select the **Range Check** tab, enable the test, and enter the range of values. Set the tolerance to 0% of rows.

DSD For Uk Orders 2004.Order Id

☒ Spaces Check
 ☒ Patterns Check
 ☒ Values Check
 ☒ Null Check
 ☒ Range Check
 ☒ Schema Length Check

☒ **Passed** (Click [here](#) to disable this test.)
 Drill to [passing](#) or [failing](#) values.

Ensure the values fall between and .
 Set the tolerance to be % of

Passes on 97.059% of values.
 Passes on 97.674% on rows.

- e. Select in the **Demo Project** the **Entities > Uk Orders 2004 > Attributes > Payment Method** attribute, right click, then select **Edit DSD**.
- f. Select the **Value Check** tab, enable the test and enter values to check as shown below:

DSD For Uk Orders 2004.Payment Method

☒ Schema Data Type Check
 ☒ Spaces Check
 ☒ Patterns Check
 ☒ Values Check
 ☐ Null Check
 ☐ Range Check

☒ **Passed** (Click [here](#) to disable this test.)
 Drill to [passing](#) or [failing](#) values.

Determine whether specific values exist.

Exists	Not Exists
CREDIT CARD EFT ACCOUNT	

Set the tolerance to be % of

 Passes on 80.000% of values.

 Passes on 76.744% on rows.

2. Re-analyze each Attribute
 - a. Select in the **Demo Project** the **Entities > Uk Orders 2004 > Attributes > Order ID** attribute, right click, then select **Re-Analyze Attribute DSD**.
 - b. Click **Run Now** in the Schedule Job popup window.
 - c. Repeat these steps for the *Payment Method* attribute.
3. To examine **Compliance %**, expand the **Order ID** and **Payment Method** nodes, and then click the **Compliance %** information. You can drill down to the different DSD Tests results.

Apply Business Rules

We want now to add a business rule to check the following business rule: “If something was shipped then an order should exist”.

1. Select in the **Demo Project** the **Entities > Uk Orders 2004 > Metadata > Business Rules**. Right-Click and select **Add Business Rules**.
2. Enter the business rules parameters as shown below. The code of the rule is :
`IF [Order Id]>0 THEN [Quantity Shipped]>0`

Entity Business Rule

Enabled ☒
 Name

Description

Set the threshold to be % of Rows.

 The test passed on 95.349% of rows.

IF [Order Id]>0 THEN [Quantity Shipped]>0

Choose expression elements from the lists below

Attributes	Account Id
Functions	Cc On File
Literals	Invoice Id
Operators	Line Item
	Order Date
	Order Id
	Paid
	Payment Method
	Product Id

Save Cancel

3. After saving the rule, you are prompted to check the rule. Click **OK** to check the run and **Run Now** to run the job immediately.
4. Double click the **Business Rules** node to list the business rules, and then select the *Order Shipped* business rule to drill down to the failing rows. These show
 - 3 empty shipments, where *Quantity Shipped=0* and *Orders Ids > 0*
 - One shipment with no order, where *Quantity Shipped>0* and *Order Id = 0*

Oracle Data Quality for Data Integrator Tutorial

Design a Name and Address Cleansing Project

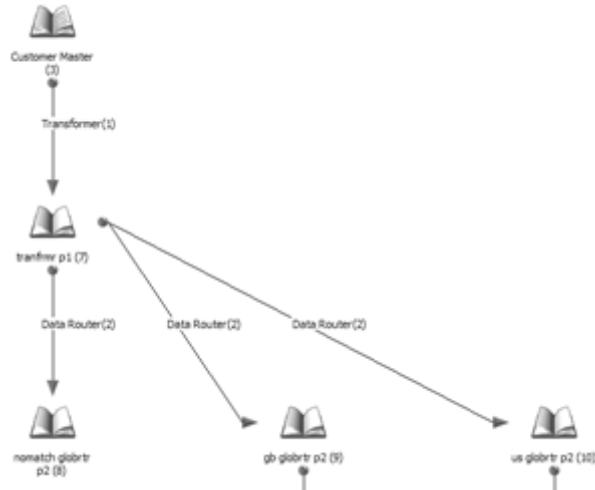
1. A data cleansing task is created in the form of a Quality project.
Create a Quality Project as follows.
 - a. Select **Quality** in the Project Types, right click and select **Create Project** in the popup menu.
 - b. Enter the project name and description, and then select **Name and Address Project**.
 - c. Select the *Customer Master* entity, and then click **Next**.

The screenshot shows a 'Create Quality Project' dialog box. It has a title bar with a close button. Inside, there are two text input fields: 'Name:' with the value 'customer master' and 'Description:' with the value 'cleanse names and addresses'. Below these are three radio button options: 'Name and Address Project' (which is selected), 'Business Data Project', and 'Empty Project'. Under the heading 'Entity selection', there is a list box containing four items: 'Customer Master(3)', 'Product Master(4)', 'Account Reps(5)', and 'Uk Orders 2004(6)'. The first item is highlighted. To the right of the list box are up, down, and list icons. At the bottom of the dialog are two buttons: 'Next' and 'Cancel'.

- d. Select *United States (us)* and *United Kingdom (gb)* for the **Countries** and then click **OK**.



- e. Click **Run Now** in the Schedule Job window.
- f. Double-Click the *customer master* project under the **Quality** node. The project diagrams opens. The project is being generated. Wait until all steps are generated.



In this diagram, the arrows correspond to processes of the data cleansing project, and the books icons to the intermediate entities for the cleansing project.

In this tutorial, we will review the processes of the data quality project, change and execute them step by step.

2. A **Transformer** process filters and performs basic transformations on input data. We use in our project a transformer to filter UK and US data and remove dashes and spaces from the phone numbers.
 - a. Double click the **Transformer** arrow to configure the **Transformer** process as follows:
 - b. To filter US and UK data, define the following row filter in **Input Settings**:

customer master - gb Transformer Configuration - Inputs Settings

Schema Editor Parser Inputs Input Settings Input Conditionals Output Settings Output Conditionals Preview Finish Cancel	Select Input: Customer_Master(3) ▼ Start Record: <input type="text"/> Maximum Records: <input type="text"/> Sample every nth row: <input type="text"/> Row Filter: <input type="text" value="Country='USA' or Country='UK'"/>
--	--

- c. To remove all dashes and spaces in the phone field, select the **Output Conditionals** option, then in the empty table, right click and select **Add > Attribute Scan** in the popup menu.
 - **Description of scan:** Phone: remove dashes and spaces
 - **Which Attribute would you like to scan:** Phone
 - **Choose alignment of the attribute:** Left Pack - this option removes all spaces in the value.
 - **Specify what the scan should look for:** Literal Value.
 - **Literal Value:** - (dash symbol)
 - Change all instances of the value to : " " (two double quotes)

The wizard steps are given below:

customer master - gb Step 1 - Attribute Scan Wizard

Description of scan: dashes and spaces

Which attribute would you like to scan: Phone

Choose alignment of the attribute: Left Pack

Specify what the scan should look for

☒ Literal Value: —

☐ Mask Value:

☐ Delimiters

Start Delimiter:

End Delimiter:

customer master - gb Step 2 - Attribute Scan Wizard

How much of the attribute should be scanned: Entire Attribute

In which direction should the attribute be scanned

☒ Left to Right

☐ Right to Left

Function to perform if Scan Value is found: Change

customer master - gb Step 3 - Change Attribute Scan Value

Enter new Value: ""

No. of occurrences to change: ☒ All ☐ 1 ☐ Other:

- d. In the same transformer, we will now define the relevant input fields that will be used by the country-specific address parsers. These fields can be overridden later in the country-specific transformers.

Under **Parser Inputs**, define your **Parser Inputs** as below

Parser Inputs (max 10 lines):

```

Bus Name
Address1
City
State
Postcode

```

Note: For this tutorial we only take into account Business Names, and not people from the companies.

- e. If you click the **Postcard...** button, you have a preview of the name and addresses fields that will be used for standardization.

- f. Click **Finish** to save the transformer parameters.
- g. Right-click the transformer arrow in the diagram, and then select **Execute...> Just This Process**.
- h. Once the process has finished, right-click the transformer arrow, then select **View...>Stats File**

The statistic file reports appears as below.

```

RECORD INPUT
Count      Statistic      Qualifier File
786        Records read   Customer_Master(1)
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E1/e1.dat
554        Records selected Customer_Master(1)
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E1/e1.dat
0          Records bypassed Customer_Master(1)
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E1/e1.dat
554        Records processed Customer_Master(1)
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E1/e1.dat

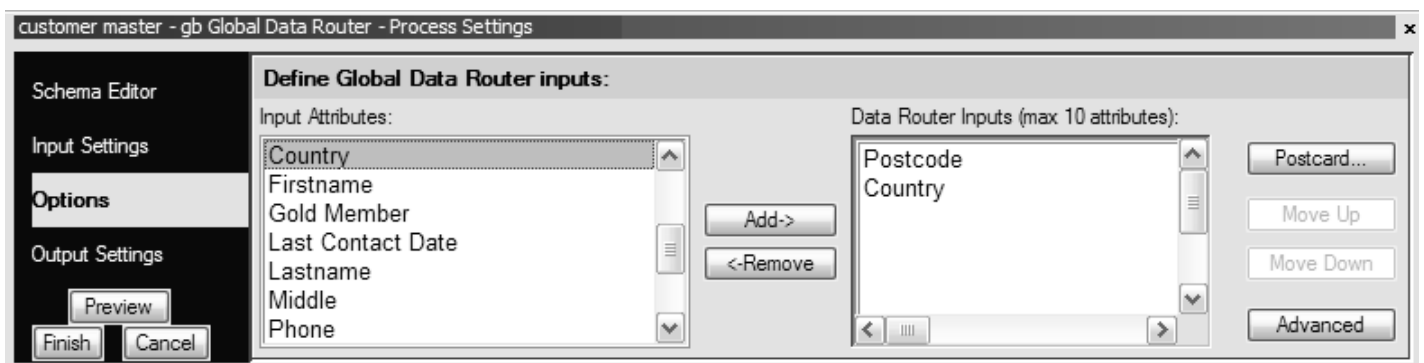
RECORD OUTPUT
Count      Statistic      Qualifier File
554        Records processed OUTPUT
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E24/e24.dat
554        Records selected OUTPUT
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E24/e24.dat
0          Records bypassed OUTPUT
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E24/e24.dat
554        Records written OUTPUT
           C:/oracle/oracledq/metabase_data/metabase/oracledq/./E24/e24.dat

FIELD SCANNING STATISTICS
Count      EntryID      Format      Funct      Scan Field
289        1          L          CH         PHONE

```

You can read the following statistics.

- In the *RECORD INPUT* section, *786 Records read* original input records. This results also in 554 records in the *RECORD OUTPUT* section.
 - In the *RECORD INPUT* section, *554 Records selected* after the filter.
 - In the *FIELD SCANNING STATISTICS*, *286 PHONE* fields scanned and transformed.
3. A **Global Data Router** separates the input records into separate entities depending on the country. As name and address processing is country-dependant, the router appears early in a data quality project.
 - a. Go back to the Quality project Diagram.
 - b. Double click the **Data Router** process.
 - c. Under **Options**, select **Postcode** and **Country** as **Data Router Inputs**.



- d. Click the **Advanced** button and next to **Country Code Attribute**, select Country from the list.

customer master - gb Global Data Router - Advanced Settings

Advanced Settings

Country Code Attribute: Country

Postcode Attribute:

Postcode Position:

Write every n records to log file:

Back

- e. Click **Back** then **Finish**.
- f. Right-click the **Data Router** arrow in the diagram, and then select **Execute...> Just This Process**.
- g. Once the process has finished, right-click the **Data Router** arrow, then select **View...>Stats File**.

These stats show 300 records for the USA and 254 for the UK (GB). Thanks to the filter in the transformer, there is no record with a NOMATCH qualifier.

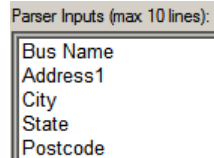
RECORD INPUT			
Count	Statistic	Qualifier	
554	Records read	tranfrmr_p1(2)	
554	Records selected	tranfrmr_p1(2)	
0	Records bypassed	tranfrmr_p1(2)	
554	Records processed	tranfrmr_p1(2)	

RECORD OUTPUT			
Count	Statistic	Qualifier	
0	Records processed	NOMATCH C:/Docu	
0	Records selected	NOMATCH C:/Docu	
0	Records bypassed	NOMATCH C:/Docu	
0	Records written	NOMATCH C:/Docu	
300	Records processed	US	C:/Docu
300	Records selected	US	C:/Docu
0	Records bypassed	US	C:/Docu
300	Records written	US	C:/Docu
254	Records processed	GB	C:/Docu
254	Records selected	GB	C:/Docu
0	Records bypassed	GB	C:/Docu
254	Records written	GB	C:/Docu

4. Now that the data flow is split per country, we can perform country specific transformations using **Country-Specific Transformers**. In this specific transformer, we will configure in the **Parsers Inputs** which fields in the input records are need to be examined when standardizing name and addresses. In our case, these fields are Bus Name (Business Name), Address1, City, State and Postcode (Zip Code).

Note: The Country Specific Transformer can be used to perform other type of transformations, such as the attribute scans we have used in the first transformer step.

- a. Double click the **us Transformer** to edit it.
- b. Under **Parser Inputs**, check that the **Parser Inputs** are defined as below. These are inherited from the values specified in the first Transformer.



Parser Inputs (max 10 lines):

Bus Name
Address1
City
State
Postcode

- c. If you click the **Postcard...** button, you have a preview of the name and addresses fields that will be used for standardization. Note that these are only US addresses.
- d. Click **Finish** to save the transformer parameters.
- e. Execute the **us Transformer** and examine the stats file. It should show that all 300 input records end up in the output records.

Note: For this tutorial, we will only focus on the US data, and delete all subsequent process steps involving UK data.

- f. Right-click the **gb Transformer**, then select **Delete Process... > This process and dependants**. Wait until all processes after the **gb globtr pXX** entity are deleted.
5. We have defined the fields useful for recognizing the name and addresses. These fields will be analyzed by a **Customer Data Parser** that will identify and parse name and address data. This parser uses country-specific rules for analyzing the addresses. It output original data plus recoded or standardized data. We will customize the data parser by indicating that the first line returned by the previous transformer step is a business line, and to indicate that we only want to have one business name per record.

- a. Double-click the **us Customer Data Parser** process to edit it.
- b. Select the **Options**, and then select *Business Name* for **Line 1**. Let all other lines as *Not Predefined*.
- c. Click **Finish** to apply your changes.
- d. Execute the **us Customer Data Parser** process.
- e. Right-click the **us cusparse pXX** entity displayed under the **us Customer Data Parser**, then select **Analyze** in the popup menu.
- f. In the window that appears, click **OK** to start the output entity analysis.
- g. Double-click in the Explorer the **Quality > customer master > Entities > us cusparse pXX > Attributes > PR_REV_GROUP > Unique Values** node.

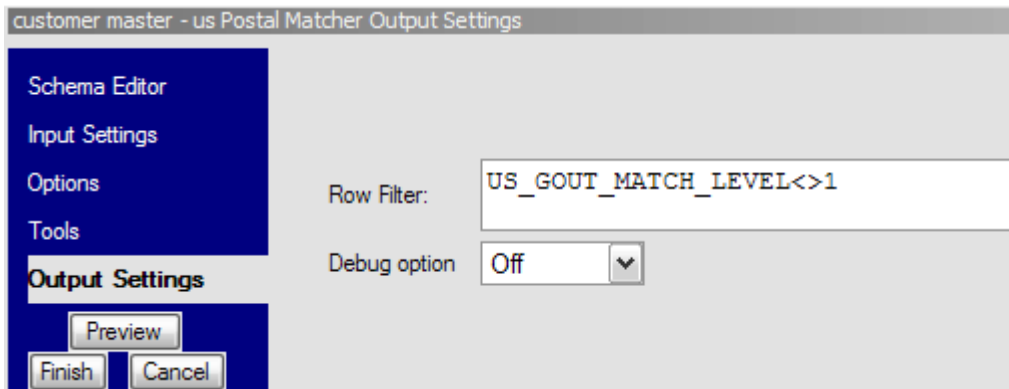
This value distribution shows the occurrence of the different data parser review group codes. For example, the 6 records with **PR_REV_GROUP=18** are those for which the city name is present but not recognized due to typos. You can drill-down and review the invalid values. See the on-line documentation for more information or the review codes and review group codes.

- h. You can also examine the Stats file for the **us Customer Data Parser** process (right-click then View... > Stats File) to have a detailed report on the parsing.
6. The **Sort for Postal Matcher** sorts the data in geographic order to improve the performances of the next step: the postal matcher.
- a. Execute the **us Sort for Postal Matcher** process.
7. The **Postal Matcher** enriches data by matching data with postal directory information.
- a. Execute the **us Postal Matcher** process.
 - b. Right-click the **us pmatch pXX** entity displayed under the **us Customer Data Parser**, then select **Analyze** in the popup menu.

- c. In the window that appears, click **OK** to start the output entity analysis.
- d. Double-click in the Explorer the **Quality > customer master > Entities > us pmatch pXX > Attributes > US_GOUT_MATCH_LEVEL > Unique Values** node. These values correspond to how accurately the record matched with the postal directory data. Drill down to the rows for each value and examine them.
 - 0: exact match
 - 1: No city found to match.
 - 2: Street name failure
 - 3: House number range failure
 - 4: Street component failure
 - 5: Multiple possible matches to directory.

Important Note: Most records end up in a “1: No city found to match.” state. This is because the sample postal directory only contains the information for New York City. Other cities are not recognized and the records cannot be enriched by the Postal Matcher. For a better readability of the results, we will filter the output of this process for the rest of the tutorial and ignore records outside New York.

- e. Double-click the **us Postal Matcher** process to edit it.
- f. Select the **Output Settings**, and then add a **Row Filter** as shown below.



- g. Click **Finish** to save these settings, then execute the **us Postal Matcher** process again. The new *Record Output* section in the statistics (right-click **View ... > Stats File**) for this process should not show *15 Records written*.
8. The **Window Key Generator** generates a composite key used in relationship linking. This key is constructed from elements in the input data. Records with similar Window Key are likely to be matching records. This key generation can be customized if needed.
 - a. Execute the **us Window Key Generator** process.
 - b. Right-click the **us winkey pXX** entity displayed under the **us Window Key Generator**, then select **Analyze** in the popup menu.
 - c. In the window that appears, click **OK** to start the entity analysis.
 - d. Double-click in the Explorer the **Quality > customer master > Entities > us winkey pXX > Attributes > WINDOW_KEY_01 > Unique Values** node.
 - e. Drill down to the unique values, then to the rows with matching WINDOW_KEY_01 values. These rows are likely to match (similar zip codes, business names, street address, person names, etc).
9. The **Sort for Linking** process sorts data for optimizing the relationship linker process.

- a. Execute the **us Sort for Linking** process.
10. The Relationship Linker process identifies records with a matching relationship or duplicate records. The output is categorized as success, fail, or suspicious based on the records similarity. A score is assigned to records.
 - a. Execute the **us Relationship Linker** process.
 - b. Right-click the **us rellink pXX** entity displayed under the **us Relationship Linker**, then select **Analyze** in the popup menu.
 - c. In the window that appears, click **OK** to start the entity analysis.
 - d. Double-click in the Explorer the **Quality > customer master > Entities > us rellink pXX > Attributes > LEV1_MATCHED > Unique Values** node.
 - e. These unique values refer to the unique business detected by the linker. If you click on one unique value, you see all rows representing the same business.
11. The **Commonizer** copies data across records linked by the relationship linker. It also selects a best of surviving record.
 - a. Execute the **us Commonizer** process.
12. The **Data Reconstructor** reconstructs data for each output of the commonizer. Configure **us Data Reconstructor** as follows:
 - a. Double-click the **us Data Reconstructor** to edit it.
 - b. Under **Schema Editor**, add the following to the Output Attributes list, by dragging them from the Attributes under **us common pXX** in the left of the panel at the end of the list of output attributes at the right of the panel.
 - **LEV2_SURVIVOR_FLAG**
 - c. Click **Finish** to close the **us Data Reconstructor**
 - d. Right-click the **us Data Reconstructor** then select **Apply Schema Changes** in the popup menu.
 - e. Click **OK** to apply the changes.
 - f. Execute **us Data Reconstructor** process.
 - g. Double click the output entity **us datarec pXX**, and examine the rows.
 - h. Right-click the header row of the Data Rows view, then select **Choose Columns**. Double click in the **Hidden fields** list the following columns to have them added to the **Displayed fields**:
 - **Lev2 Survivor Flag**
 - i. Click **OK** to validate. This new column appears in the table.
 - j. Review the output records.
 - Records flagged *Us Gout Match Level = 0* are those that have been enriched with the postal data, and those with *Us Gout Match Level = 2* are those that have not been correctly recognized and enriched.
 - Records flagged with *Lev2 Survivor Flag = 1* are the survivor records of the de-duplication process (containing the most comprehensive Business information).
 - The fields *Newaddr1*, *Newaddr2*, etc contain new address lines with comprehensive and cleansed addresses.
13. Export the project as a Batch Script. This process makes this project available for Oracle Data Integrator.
 - a. From the **Explorer** or **Project Workflow**, right-click the **demo** project and select **Export... > ODQ Batch Project > No data**.

- b. In the **Browse for Folder** window, select the
`<ODI_Home>\demo\oracledq\projects` folder for exporting the project.
- c. Click **OK**. A message indicates that the files are being copied. This creates a `oracledq` (named after the metabase) folder at the location that you specified. This folder contains a `projectN` sub-folder (where *N* is the project identifier in Oracle Data Quality). This project sub-folder contains the following folders among others:
 - **data**: This folder contain input and output data as well as temporary data files. As you specified No data for the export, this folder is empty for now.
 - **ddl**: This folder contains the entities metadata files (.DDX and .XML). These files described the data files. These files are prefixed with `eNN_`, where *NN* is the Entity ID. Customized reverse-engineering is used in Oracle Data Integrator to retrieve these entities file format in the form of datastores.
 - **scripts**: This folder contains the configuration file `config.txt`, and the batch script `runprojectN.cmd`
- d. You will need to make a few changes to various configuration files in order to point your newly exported batch program to the correct input and output data. The first change tells the data quality engine where the project has been exported. You typically need to perform this change when moving the export file from the design-time server to the run-time server.
 - In the **scripts** directory, open the **config.txt** and change the DATABASE parameter to "`<ODI_Home>\demo\oracledq\projects\oracledq`"
 - Also in the **scripts** directory, open the **runProjectN.cmd** file and change the TS_PROJECT parameter to
`<ODI_Home>\demo\oracledq\projects\oracledq\projectN`
 and uncomment the very last line of the file (remove the `::` character at the beginning of the line)
- e. The second change indicates to the data quality engine the location of the source and target files, as well as their format. The source file is referenced in the first transformer process settings file, and the target file is referenced in the last process settings.
 - In the **settings** directory, open the file named `eN_transfmr_pXX.stx` (where *N* is the internal ID of the entity corresponding to the `customer_master.csv` file) and change the following options in the XML structured file:


```

/CATEGORY/INPUT/PARAMETER/INPUT_SETTINGS/ARGUMENTS/ENTRY/DATA_FILE_NAME =
    <ODI_Home>\demo\oracledq\Data\customer_master.csv
/CATEGORY/INPUT/PARAMETER/INPUT_SETTINGS/ARGUMENTS/ENTRY/FILE_DELIMITER =
    ,
/CATEGORY/INPUT/PARAMETER/INPUT_SETTINGS/ARGUMENTS/ENTRY/START_RECORD =
    2
          
```
 - Also in the **settings** directory, open the file named that starts with `eN_` where *N* is the largest value in the directory and ends with `_xfmr.stx` and change the following options in the XML structured file:


```

/CATEGORY/OUTPUT/PARAMETER/OUTPUT_SETTINGS/ARGUMENTS/DATA_FILE_NAME =
    <ODI_Home>\demo\oracledq\Data\cleansed_customer_m
aster.csv
          
```

- f. Your cleansing job can now be invoked by running the
`<ODI_Home>\demo\oracledq\projects\oracledq\projects\oracledq\projectN\scripts\runProjectN.cmd`

Run the Quality Project in ODI

1. Open Oracle Data Integrator and connect to your repository. If you are starting with Oracle Data Integrator, use the demo environment.
2. Select the Project view, click the new project button to create a new project named *demo_quality*
3. Under this new project, open the **First Folder**, then in the **Packages** node, right-click and select **Insert Package**.
4. Enter *Quality Call* in the **Package Name** field, and then select the **Diagram** tab.
5. In the **Utilities** tools group in the toolbar, select the **OdiDataQuality** tool.
6. Click the diagram to add a step with this tool.
7. Set the following parameters for this tool:
 - a. **BATCH_FILE**: Name of the runprojectN.cmd file in the /script sub-directory of your project export directory. For example:
`<ODI_Home>\demo\oracledq\projects\oracledq\projectN\scripts\runprojectN.cmd`
2. Click the **Apply** button to save the package, then the **Execute** button to run it. The entire quality project runs.

Going Further with Oracle Data Quality for Data Integrator

Now that the project runs, it is possible to use the input and output files in regular Oracle Data Integrator interfaces, in order to:

- Load the input file using datastores from various sources.
- Re-integrate the cleansed output data into the sources.