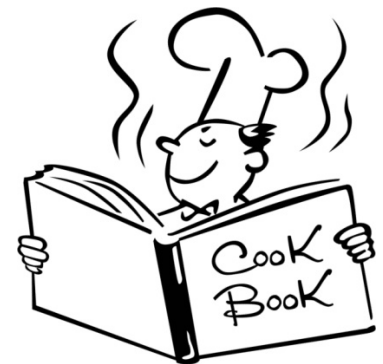


# Lead / Contact Cleansing 101

**OPEN**RISE

Cook Book Series



# Recipe Overview

## Rule 1

Clean R1 - Clean email

## Rule 2

Clean R2 - Clean URL

## Rule 3

Clean R3 - Normalize state + city

## Rule 4

Clean R4 - Infer city from ZIP

## Rule 5

Clean R5 - Infer state from ZIP

## Rule 6

Clean R6 - Infer country from state

## Rule 7

Clean R7 - Normalize phone

## Rule 8

Clean R8 - De-duplication

This is a basic recipe for cleaning contact data

- Fix bad email and URL
- Clean up and normalize telephone number and URL
- Normalize and fill in missing city, state, and country
- De-duplicate based on email



You will need the following raw data:

- Email and URL
- City, state, country, and postal code
- Phone number

# tips

- Add a rule by clicking on an existing rule  and  and +



- Put new data into a new data attribute so you can easily compare before vs. after and confirm the rule is doing what it is supposed to do
- Use the  button to create a new attribute
- Use the  button to add multiple actions within a rule
- Can't see the open reference data? Check the setting in your Data Catalog  
 Show open data sets
- Most rule templates can perform different tasks when combined with different reference data
- You can use reference data provided by Openprise or ones created by yourself

# Rule 1: Clean Email

Rule template \* [Need help picking a templates?](#)

Email address format fix

IF this happens

Input Data Sources

Marketo

**SELECT DATA**

Email Address contains ( \* )

THEN take these actions

Email clean up

Fix bad email addresses in the following attributes:

**ADD EMAIL** **ADD ATTRIBUTE**


Email attribute to fix: Email

Write fixed email to: Email\_Cleaned

Use suffixes listed in: Reference - Domain Suffixes

Suffix

## Reference Data



**Reference - Domain Suffixes**  
List of known domain suffixes on the Internet. This reference data is used by email and URL clean-up rules.

# Rule 2: Clean URL

Rule template \* Need help picking a templates?

URL format clean-up

IF this happens

Input Data Sources

Rule Output - Clean R1 - Clean email

SELECT DATA

Website contains ( \* )

THEN take these actions

URL clean up

clean-up bad URL addresses in the following attributes:

ADD URL    ADD ATTRIBUTE

URL attribute to clean-up: Website

Write fixed URL to: Website\_Clean

Use suffixes listed in: Reference - Domain Suffixes


Suffix

Include HTTPS:// in cleaned URL data

Convert Unicode to Punycode

Options!

## Reference Data



**Reference - Domain Suffixes**  
List of known domain suffixes on the Internet. This reference data is used by email and URL clean-up rules.

# Rule 3: Normalize State + Country

Rule template \* Need help picking a templates?

Simple replacement / normalization

IF this happens

Input Data Sources

Rule Output - Clean R2 - Clean URL

SELECT DATA

All new data

THEN take these actions

Set attribute value

Attribute to normalize / clean up :

State/Region

Write normalized value to

State/Region\_Cleaned

CREATE ATTRIBUTE

Normalized-alias mapping is stored in

Reference - States, Provinces, and

Normalized value is stored in

State Province Name

Alias is stored in

State Province Name Alias

Match method

Fuzzy match

Reference Data



Reference - States, Provinces, and Regions

List of states, provinces, and regions of various countries. Current coverage includes the US, Canada, UK, Germany, France, and Australia. This reference data set works with Data Rules to perform normalization and enrichment.

Reference Data



Reference - Countries

This data set contains the countries of the world with name, ISO alpha-2 and alpha-3 codes, calling code, and currency. This data set is designed to use with data rules to normalize and enrich data.

Set attribute value

Attribute to normalize / clean up :

Country

Write normalized value to

Country\_Cleaned

CREATE ATTRIBUTE

Normalized-alias mapping is stored in

Reference - Countries

Normalized value is stored in

Country Name

Alias is stored in

Country Name Alias

Match method

Fuzzy match

# Rule 4: Infer City From ZIP (US Only)

Rule template \* Need help picking a templates?  
Infer value

IF this happens

Input Data Sources  
Rule Output - Clean R3 - Normalize state + country

SELECT DATA


City has no value ( novalue )  
and ZIP contains ( \* )  
and Country\_Clean matches ( United States )

THEN take these actions

Inferred value

Use value from	ZIP
To infer a value for	City_Infer
	CREATE ATTRIBUTE
Inferred value mapping is stored in	Reference - US ZIP Codes
Inferred value is stored in	City
Look up values are stored in	ZIP Code
Match method	Begins with

## Reference Data



**Reference - US ZIP Codes**  
List of ZIP (Postal) Codes in the United States. This reference data works with Data Rules for normalization and enrichment.

Inferred value

Use value from	ZIP
To infer a value for	City
	CREATE ATTRIBUTE
Inferred value mapping is stored in	Reference - US ZIP Codes
Inferred value is stored in	City
Look up values are stored in	ZIP Code
Match method	Begins with

TIP: Populate both "City" and "City\_inferred" to easily identify which records have been updated and directly overwrite existing "City"

# Rule 5: Infer State From ZIP (US Only)

**Rule template** \* [Need help picking a templates?](#)

Infer value

**IF this happens**

**Input Data Sources**

Rule Output - Clean R4 - Infer city from ZIP

**SELECT DATA**

State\_Clean has no value ( novalue )

and ZIP contains ( \* )

and Country\_Clean contains ( states )

**THEN take these actions**

**Inferred value**

Use value from ZIP

To infer a value for State\_Clean

**CREATE ATTRIBUTE**


Inferred value mapping is stored in Reference - US ZIP Codes

Inferred value is stored in State Name

Look up values are stored in ZIP Code

Match method Begins with

## Reference Data



**Reference - US ZIP Codes**

List of ZIP (Postal) Codes in the Unites states. This reference data works with Data Rules for normalization and enrichment.



# Rule 6: Infer Country From State

Rule template \* [Need help picking a templates?](#)

Infer value

IF this happens

**Input Data Sources**

Rule Output - Clean R5 - Infer state from ZIP

**SELECT DATA**

Country\_Clean has no value ( novalue )

and State\_Clean contains ( \* )

THEN take these actions

**Inferred value**

Use value from: State\_Clean

To infer a value for: Country\_Clean

**CREATE ATTRIBUTE**


Inferred value mapping is stored in: Reference - States, Provinces, and Regions

Inferred value is stored in: Country Name

Look up values are stored in: State Province Name

Match method: Exact match

## Reference Data



**Reference - States, Provinces, and Regions**  
List of states, provinces, and regions of various countries. Current coverage includes the US, Canada, UK, Germany, France, and Australia. This reference data set works with Data Rules to perform normalization and enrichment.

# Rule 7: Normalize Phone Number

Rule template \* [Need help picking a templates?](#)

Phone number normalization

IF this happens

Input Data Sources

Rule Output - Clean R6 - Infer country from state

SELECT DATA

Phone Number contains ( \* )

THEN take these actions

Reformat phone number

Reformat the phone number found in

Phone Number

based on the country information in

Country\_Cleaned

Write the reformatted phone number to

Phone\_International

CREATE ATTRIBUTE

in

International +41 44 668 18 00

format

Country lookup requires ISO alpha-2 code

Use the ISO code mapping stored in

Reference - Countries

ISO alpha-2 code is stored in

ISO Alpha-2 Code

Country alias is stored in

Country Name Alias

## Reference Data



### Reference - Countries

This data set contains the countries of the world with name, ISO alpha-2 and alpha-3 codes, calling code, and currency. This data set is designed to use with data rules to normalize and enrich data.

# Rule 8: De-duplicate

**Rule template** \* [Need help picking a templates?](#)

De-duplication and merge

**IF this happens**

**Input Data Sources**

Rule Output - Clean R7 - Normalize phone

**SELECT DATA**

All new data

**THEN take these actions**

**De-duplication and merge**

Records with exactly the same values in the following attributes are considered duplicates:

Email\_Cleaned

The surviving record is the records with latest Last Modified Date

Non-surviving records should be merged into surviving records

TIP: Your final data output can be found in these Data Sources

**Output Data Sources**

Rule Output - Clean R8 - Dedupe  
and Rule Output - Clean R8 - Dedupe - non-surviving

# Recipe Review

## Rule 1

Clean R1 - Clean email

## Rule 2

Clean R2 - Clean URL

## Rule 3

Clean R3 - Normalize state + country

## Rule 4

Clean R4 - Infer city from ZIP

## Rule 5

Clean R5 - Infer state from ZIP

## Rule 6

Clean R6 - Infer country from state

## Rule 7

Clean R7 - Normalize phone

## Rule 8

Clean R8 - De-duplication

## Use the Search screen to find out:

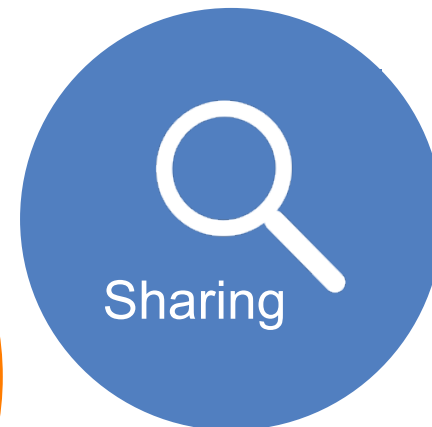
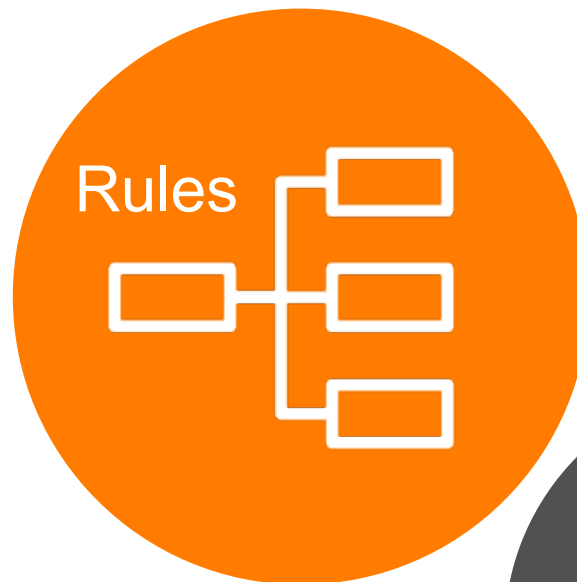
- How many emails and URLs were fixed?
- How many telephone numbers and URLs were normalized?
- How many cities, states, and countries were normalized and filled in?
- How many duplicate records were identified?

## Want to do more? Try the following on your own:

- Fill in missing company, state, and country using IP inferred data
- Normalize company name against customer and target account lists

# OPENPRISE

Data Automation For Business Users



[info@openprisetech.com](mailto:info@openprisetech.com)

Twitter: @openprisetech

[www.openprisetech.com](http://www.openprisetech.com)