

**Custom Matching Rule**

Oracle Watchlist Screening (OWS)

for

INTERNATIONAL REGISTRIES INC. (IRI)

Document Control Information

Document Edit History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Additions/Modifications | Prepared/Revised by |
| 1.0 | May 30, 2020 | Initial version | Senthil Anand |
| 1.1 | October 08, 2021 | Updated for 3 word name matching requirements | Yusuf Rajkotwala |
|  |  |  |  |

Document Review/Approval History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Name | Organization/Title | Comments |
|  |  |  |  |
|  |  |  |  |

Distribution of Final Document

The following people are designated recipients of the final version of this document:

|  |  |
| --- | --- |
| Name | Organization/Title |
|  |  |
|  |  |

**Table of Contents**

[1. Introduction 4](#_Toc84498980)

[2. Matching Requirement 4](#_Toc84498981)

[3. Logic for One-word name 6](#_Toc84498982)

[4. Logic for Two-word name 6](#_Toc84498983)

[5. Logic for Three-word name 6](#_Toc84498984)

[6. Logic for Three + word name 7](#_Toc84498985)

[7. Custom Matching Rules 7](#_Toc84498986)

[8. Custom Comparison 9](#_Toc84498987)

# 

# Introduction

INTERNATIONAL REGISTRIES (IRI) has Oracle Watchlist Screening (OWS) solution and there is a need to customize the matching rules for individual screening, IRI has SAN Watchlist Screening (Sanctions) which means, Oracle has less tolerance in matching and identify every possible match against sanctions list, so IRI wanted to optimize certain rules to bring less false positives from OWS. This document will explain how the matching logic is tuned to achieve the expected results.

# Matching Requirement

The names are split by geography wise

* Asian
* South Asian
* Western
* Middle Eastern
* Russian

and names were taken into consideration that has one word, two word, three word and three word + names

IRI’s expectation on the matches shall be as depicted below in the table

|  |  |  |  |
| --- | --- | --- | --- |
| **TYPE** | **PREFIX** | **SUFFIX** | **LOGIC** |
| ONE WORD |  |  | EXACT NAME |
| *IBRAHIM* | any 1st word | any last word | \*SOUNDEX\* |
|  | any 1st, 2nd word | any last word n, n-1 | \*\*SOUNDEX\*\* |
|  | any 1st, 2nd, 3rd word | any last word n, n-1,n-2 | \*\*\*SOUNDEX\*\*\* |
|  |  |  |  |
| TWO WORD |  |  | EXACT NAME |
| *CHEN HU* | 1st word | 2nd word | CHEN HU |
|  | 2nd word | 1st word | HU CHEN |
|  |  |  | 1st word \* 2nd word |
|  |  |  | 2nd word\*1st word |
|  |  |  | \*1st 2nd word |
|  |  |  | \*2nd 1st word |
|  |  |  | 1st word 2nd word \* |
|  |  |  |  |
| THREE WORD |  |  | EXACT NAME |
| *THOMAS JAMES JACKSON* | 1st word | 3rd word | THOMAS JACKSON |
|  | 1st word | 2nd word | THOMAS JAMES |
|  | 1st word | 3rd word | 1st word \*J\* 3rd word |
|  | 2nd word | 1st word | JAMES THOMAS |
|  | 3rd word | 1st word | JACKSON THOMAS |
|  |  |  |  |
| FOUR LETTER WORD |  |  | EXACT NAME |
| *CHRISTIAN DE LOS SANTOS* | 1st word | 4th word | CHRISTIAN SANTOS |
|  | 1st word | 3RD, 4TH WORD | CHRISTIAN LOS SANTOS |
|  | 1st word | 3RD, 4TH,5TH WORD | 1ST WORD \*D\* 2ND,3RD,4TH,5TH WORD |

# Logic for One-word name

For Example, if the name input in IBRAHIM

The matching will fetch all combinations of IBRAHIM which includes ABRAHAM, EBRAHIM, IBRAHIMI etc., Oracle considers 20 variations of Mohammad as one name example, MOHAMMAD will fetch MUHAMAD, MOHAMED, MD, MOHAMMED etc. and same applies for any name which sounds like the name

In this scenario IRI has customized to bring all IBRAHIM, EBRAHIM, IBRAHIMI (one letter words)

Two letter words which has IBRAHIM, EBRAHIM, IBRAHIMI

Three letter words which has IBRAHIM, EBRAHIM, IBRAHIMI

Three letter + words which has IBRAHIM, EBRAHIM, IBRAHIMI

This will also include all titles with the names DR, PROF, MR, MRS, REVERAND, SHEIKH etc

# Logic for Two-word name

For example, if the name input is CHEN HU

The matching will fetch all combinations of

* 1st word and 2nd word - CHEN HU
* 2nd and 1st word – HU CHEN
* \*,1st 2nd word (\* denotes any initials or names with 1st and 2nd word as middle and last names)

M CHEN HU, HU CHEN HU

* \*,2nd 1st word (\* denotes any initials or names with 2nd and 1st word as middle and last names)

HU HU CHEN, FANG HU CHEN

# Logic for Three-word name

For example, if the name input is THOMAS JAMES JACKSON

The matching will fetch all combinations of

* 1st word, 2nd word and 3rd word – THOMAS JAMES JACKSON
* 1ST word, 3rd word – THOMAS JACKSON
* 1st word, 2nd word – THOMAS JAMES
* 1ST word ,\*,3rd word – THOMAS J\* JACKSON (\* denotes any second name which has similarity of James as the second word, example THOMAS JR JACKSON, THOMAS JOHN JACKSON etc)
* 2nd word, 1st word – JAMES THOMAS
* 3rd word, 1st word – JACKSON THOMAS

# Logic for Three + word name

For example, if the name input in CHRISTIAN DE LOS SANTOS

The matching will fetch all combinations of

* 1st word, 2nd word, 3rd word, 4th word – CHRISTIAN DE LOS SANTOS
* 1st word, 4th word – CHRISTIAN SANTOS
* 1st word, 3rd word, 4th word – CHRISTIAN LOS SANTOS
* 1st word . \*, 2nd word, 3rd word, 4th word – (\* denotes any second name which has similarity of De as the second word, example CHRISTIAN DES LOS SANTOS etc.)

# Custom Matching Rules

|  |  |
| --- | --- |
| **Rule name** | **Function** |
| [I130OB] IRI All names in any order only | All names in the full name match (using a **Word Edit Distance** of 0) after nametoken standardization, in any order. A single typo (1 character edit) is allowed in each name token. |
| [I180OA] IRI First name and full name similar and sounds like only | The full name matches with a **Character** **Match Percentage** of 80% or above, aftername token standardization. At least one of the family name tokens, excluding initials, must match by a 4-character Metaphone key. |
| [I180OB] IRI First name and full name similar and sounds like only | The full name matches with a **Character** **Match Percentage** of 80% or above, aftername token standardization. At least one of the family name tokens, excluding initials, must match by a 4-character Metaphone key. |
| [I220OA] IRI Additional names typo tolerant only | All name tokens from the full name with fewest tokens must be present in the other full name. A character error tolerance of 20% is allowed (that is, one character edit every 5 characters). At least 2 name tokens must match with the same matching logic. If a name contains only one token it is not considered a match according to this rule.  NOTE: Word Match Count may return >1 if a single name matches twice in a longer name string. For example, ‘ABDUL’ matches ‘ABDUL ABDUL’ with a Word Match Count of 2. Matching is order sensitive. |
| [I220OB] IRI Additional names typo tolerant only | All name tokens from the full name with fewest tokens must be present in the other full name. A character error tolerance of 20% is allowed (that is, one character edit every 5 characters). At least 2 name tokens must match with the same matching logic. If a name contains only one token it is not considered a match according to this rule.  NOTE: Word Match Count may return >1 if a single name matches twice in a longer name string. For example, ‘ABDUL’ matches ‘ABDUL ABDUL’ with a Word Match Count of 2.  Matching is order sensitive. |
| [I220Oc] IRI Additional names typo tolerant only | All name tokens from the full name with fewest tokens must be present in the other full name. A character error tolerance of 20% is allowed (that is, one character edit every 5 characters). At least 2 name tokens must match with the same matching logic. If a name contains only one token it is not considered a match according to this rule.  NOTE: Word Match Count may return >1 if a single name matches twice in a longer name string. For example, ‘ABDUL’ matches ‘ABDUL ABDUL’ with a Word Match Count of 2.  Matching is order sensitive. |
| [I220OD] IRI Additional names typo tolerant only | All name tokens from the full name with fewest tokens must be present in the other full name. A character error tolerance of 20% is allowed (that is, one character edit every 5 characters). At least 2 name tokens must match with the same matching logic. If a name contains only one token it is not considered a match according to this rule.  NOTE: Word Match Count may return >1 if a single name matches twice in a longer name string. For example, ‘ABDUL’ matches ‘ABDUL ABDUL’ with a Word Match Count of 2.  Matching is order sensitive. |
| [I991A] IRI FN Single word name match rule | If the input has 1 word, check for the word match count to 1 |
| [I991B] IRI FN Single word name sounds like | If the input has 1 word, check for character edit distance of 1 word and names sound similar |
| [I991C] IRI Single word name metaphone | If the input has 1 word, and the metaphone is true |
| [I991D] IRI Single word name metaphone in common | If the input has 1 word and the metaphone in common is true |
| [I060OB] IRI Abbreviated standardized given name only | Given names match using a Starts With comparison, after name standardization using the Given Name Map. Family name matches exactly. |

# Custom Comparison

|  |  |
| --- | --- |
| Comparison Name | Function |
| IRI Family Name | Exact string match of the family name of the person |
| IRI Full name CED | The Character Edit Distance comparison compares two String/String Array values and determines how closely they match each other by calculating the minimum number of character edits (deletions, insertions and substitutions) needed to transform one value into the other |
| IRI full name WMC | The Word Match Count comparison enables matching of multi-word String/ String Array values that contain a number of common distinct words (separated by whitespace), regardless of the order in which they are found. |
| IRI Full name CMP | The Character Match Percentage comparison determines how closely two values (String, String Array) match each other by calculating the Character Edit Distance between two String values, and also taking into account the length of the longer or shorter of the two values, by character count. |
| IRI full name WED | The Word Edit Distance comparison determines how well multi-word String/ String Array values match each other by calculating the minimum number of word edits (word insertions, deletions and substitutions) required to transform one value to another. |
| IRI full name Metaphone | The Metaphone processor converts the values for a String attribute into a code which represents the phonetic pronunciation of the original string, using the Double Metaphone algorithm |
| IRI full name Metaphone in common | The Metaphone processor converts the values for a String attribute into a code which represents the phonetic pronunciation of the original string, using the Double Metaphone algorithm |
| IRI full name strong name | If the input full name has strong name which is 3 and 3+ word names |
| IRI full name has single word | If the input has one word of full name |