

ORACLE®

EDQ Matching Essentials

January, 2015

ORACLE

Copyright © 2014 Oracle and/or its affiliates. All rights reserved. |

What is Matching?

Matching allows you to identify records which may relate to a single entity for the purpose of running your business.

CU_NO	CU_ACCOUNT	TITLE	NAME	ADDRESS1	ADDRESS2	ADDRESS3	POSTCODE
10782	95-15134-SH	Mr	Victor CARSON	1A Spire Road, Glover Road Est East	Washington	NE37 3ES	
15906	98-21229-PB	Ms	J CARSON	Spire Road, Glover Estate East	WASHINGTON	Tyne & Wear	NE37 3ES

Should we treat these two customers as one?

COMPANY_NAME
DIRECT LINE GROUP SERVICES
DIRECT LINE GROUP SERVICES LIMITED
DIRECT LINE GROUP SERVICES LTD
DIRECT LINE FINANCIAL SERVICES

How many different companies are there here?

What Makes Matching Difficult?

Free text fields allow users to enter data in different formats and using different conventions

Source data is likely to be incomplete and/or incorrect.
Cannot assume that data will be in the "right" place

CU_NO	CU_ACCOUNT	TITLE	NAME	ADDRESS1	ADDRESS2	ADDRESS3	POSTCODE
10782	95-15134-SH	Mr	Victor CARSON	1A Spire Road, Glover Road Est East	Washington	NE37 3ES	
15906	98-21229-PB	Ms	J CARSON	Spire Road, Glover Estate East	WASHINGTON	Tyne & Wear	NE37 3ES

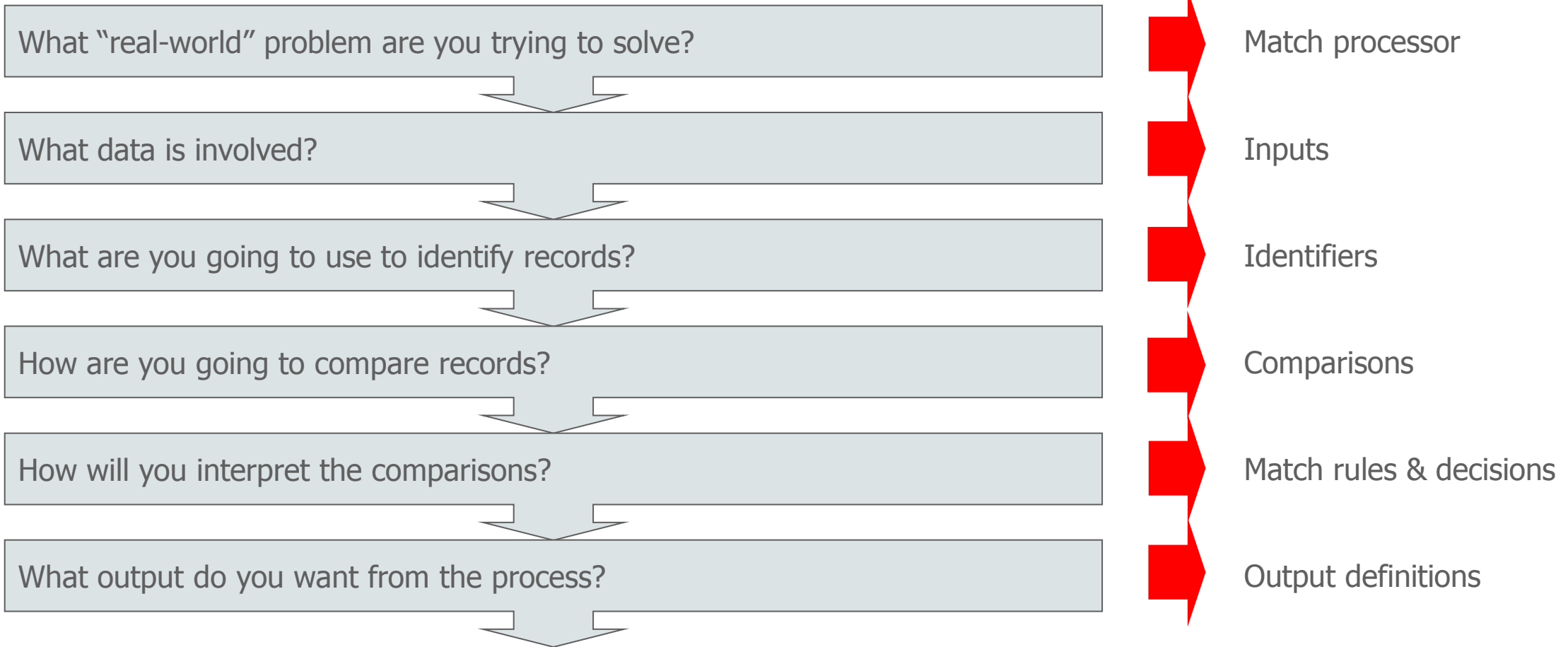
Matching configuration must capture and replicate the users' knowledge and experience together with the business rules surrounding the data.

Context is critical.
Knowledge that these 4 fields represent an address allows us to see they may be a match.

Matching Scenarios

- De-duplication
 - Find & remove duplicate entries in a system
- Consolidation
 - Combine a number of systems, eliminating duplicates and creating the “best” records
- Enhancement
 - Improve data by bringing in trusted reference data
- Linking
 - Establishing links between multiple data sets

The Matching Process



Matching in Oracle Enterprise Data Quality



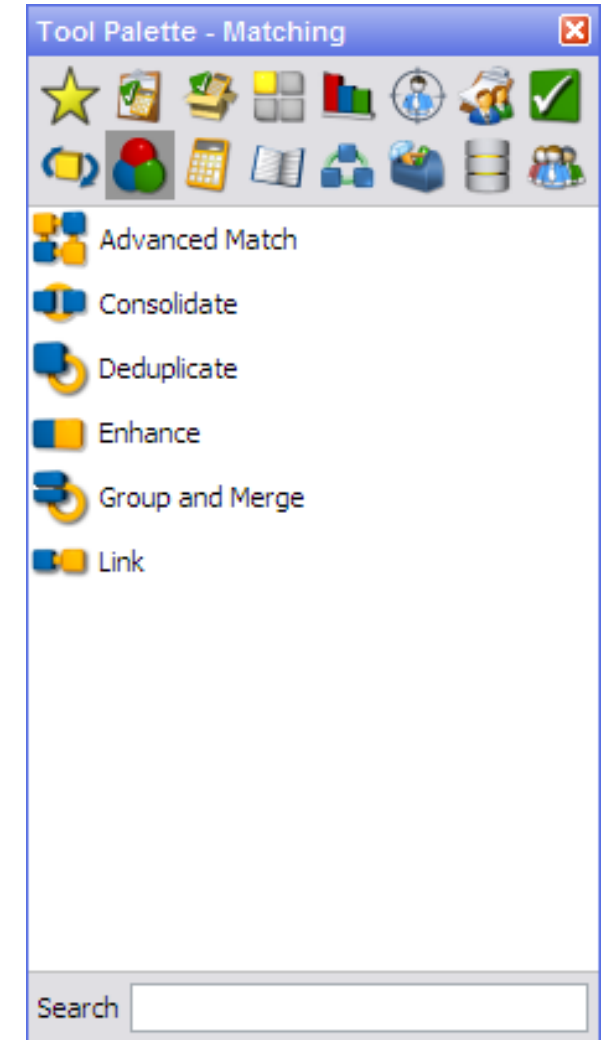
Before We Can Start Matching

- What problem are we trying to solve?
 - Do we have a business definition of a match?
 - What output do we want?
- What data can we use to find matches?
 - Profile each dataset to establish quality/reliability of identifying information
 - Names, Addresses, Part descriptions
 - What sort of variability is in the data?
 - Mis-fielding? Typos? Missing data?






The Matching Palette

Each processor solves a specific problem:

- **Advanced Match** gives you control over all the configuration options
- **Consolidate** matches and merges multiple datasets
- **Deduplicate** matches a single dataset
- **Enhance** adds information from one or more reference data sets
- **Group & Merge** gives a simple merge based on exact match only
- **Link** find matches between datasets but doesn't merge the data



The Matching Sub-processors

Icon	Sub-processor	Description
	Input	Select the attributes from the data streams included in the matching process.
	Identify	Create identifiers to use in matching, and map them to attributes.
	Cluster	Divide the data streams into clusters. See the Clustering concept guide .
	Match	Choose which comparisons to perform, and how to interpret them with match rules.
	Merge	Optionally, use rules to merge matching records, to create a 'best' set of output records

Identify and Cluster



Identifiers

- We need to create an 'Identifier' for each piece of information that we want to use in matching
- For example, within a system storing information about books, a book could be identified by:
 - A Primary Key (System identifier)
 - Its ISBN (Real-world identifier)
 - A combination of Title, Author and Publication date (Alternative identifier)

We map Attributes onto Identifiers

We map the attribute names from the 'MainDB' dataset onto the identifiers

And similarly for the 'Spreadsheet' dataset.

The screenshot shows the 'Identify' dialog box with the 'Identifier Mapping' section. It contains a table with three columns: 'Identifiers', 'MainDB', and 'Spreadsheet'. The table lists several attributes and their corresponding identifiers in both datasets. The 'address line 1' row is highlighted in blue. Below the table are 'Add' and 'Delete' buttons. At the bottom right are 'OK', 'Cancel', and 'Apply' buttons.

Identifiers	MainDB	Spreadsheet
surname	lname.LATEST	SURNAME.LATEST
forename	fname.LATEST	FORENAME.LATEST
postcode	pocode.LATEST	POSTCODE.LATEST
email	email.LATEST	EMAIL.LATEST
hometel	homePhone.LATEST	HOME_TELEPHONE.LATEST
address line 1	addr1.LATEST	ADDRESS_LINE_1.LATEST

Clustering – Why We Need It

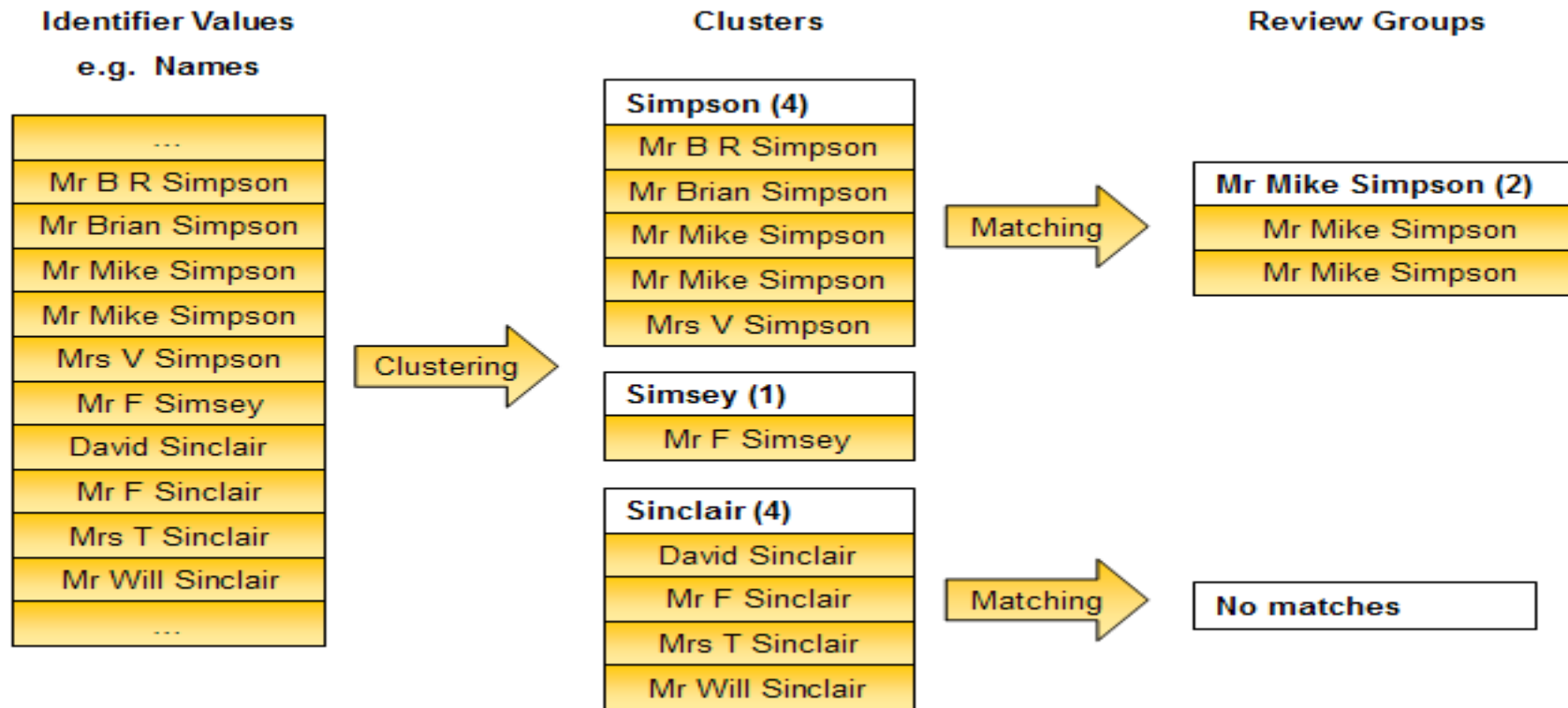
- Let's try the 'brute force' approach to de-duplicating 10 million records:
 - Start at record 1 and compare it with:
 - Record 2, Record 3, ... , Record 10 million
 - Now move onto record 2 and compare it with:
 - Record 3, Record 4, ... , Record 10 million
- This could take some time...

How Long Would It Really Take?

- The number of comparisons is about
 - Half of 10 million x 10 million, which is
 - 50,000,000,000,000!
- If a server can do 100,000 per second it will take
 - 500,000,000 seconds OR
 - 138889 hours OR
 - 15.85 years
- Which is rather too long to wait!
- So we need to work a bit smarter...

Clustering Avoids Unnecessary Work

- Only compare records with 'some similarity'

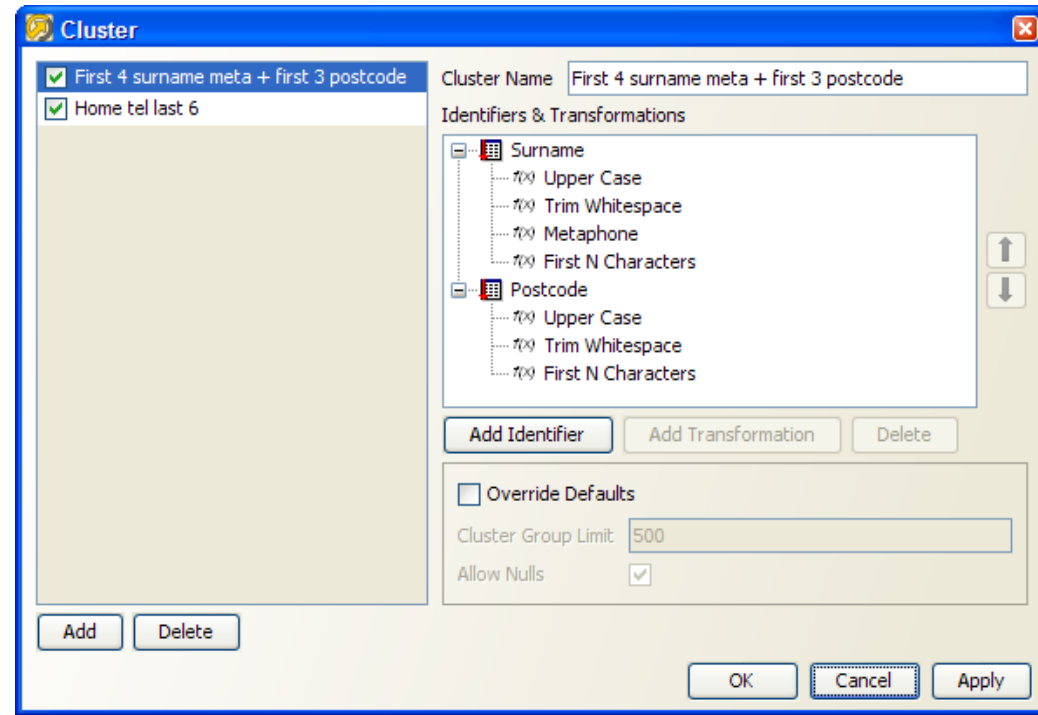


Simple Clustering Examples

- All people with the same Surname
 - What if some have a typo?
- All people with the same Postal code
 - What if the postcode has changed?
- All people with the same phone number
 - What format variability do we have?
 - Area codes, extension numbers etc.
- All people with the same date of birth
 - Do we have an accurate DOB for every record?

A More Complex Cluster

- Combines identifiers in a composite cluster
- Standardizes the identifiers to tolerate minor variation



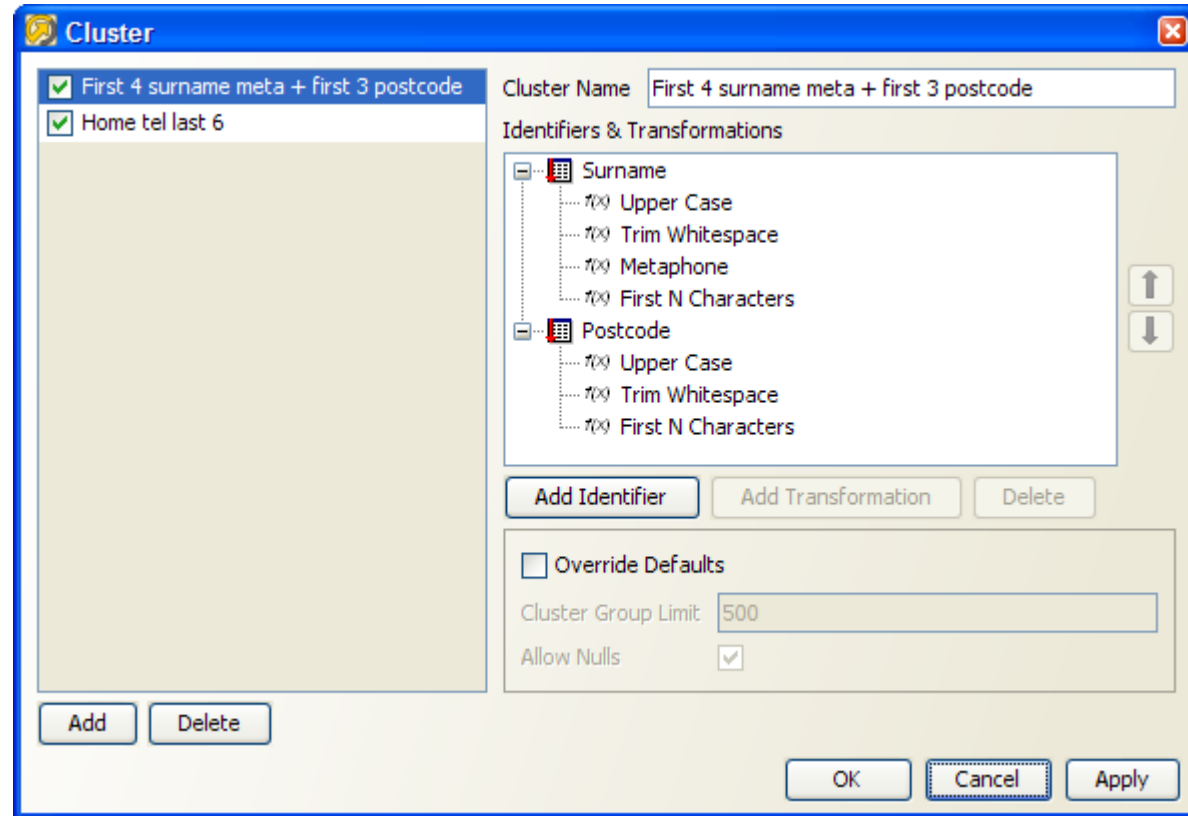
Surname	Postcode	Cluster key
Matthews	CB13 7AG	M0SCB1
Foster	CB4 1YG	FSTRCB4
JONES	SW11 3QB	JNSSW1
Jones	sw11 3qb	JNSSW1

A Good Clustering Strategy

- Tolerates errors and missing data
- Balances the trade-off between cluster group size (and therefore performance) and error tolerance
- Normally uses multiple clustering methods e.g.
 - Surname + DOB
 - Surname + Postcode
 - Postcode + DOB
 - If 1 of the 3 pieces of data is wrong/missing, we will still be OK
- May use more complex keys created by transforming the data before matching – see clustering in the EDQ Customer Data Services Pack or Oracle Watchlist Screening.

Configuring Clusters

- Add the Identifiers
- Add transformations to each identifier
- Transformations are applied in order
- Identifiers are concatenated to form a composite cluster



Run, Review & Refine Clusters

Results Browser - Match individuals

Cluster	Group size	CustDB.Customers
KLRKEC1	12	12
PLEC4	10	10
KLRKEC3	9	9
PRNEC1	8	8
KLEC4	8	8
TFSEC4	7	7
PNEC1	7	7
HSEC4	7	7
KMRNEC4	7	7
KLEC1	7	7
ATMSEC4	7	7
ANTREC4	7	7
ANTREC1	7	7
PKREC1	6	6
ATRTEC2	6	6
ATRTEC1	6	6
ATMSEC1	6	6
RSEC1	5	5

Input: CustDB.Customers Clusters: First 4 surname meta + first 3 postcode

Results Browser - Match individuals

Family_Parse	Title_Parse	Given_Parse	Middle_Parse	Address1	Address2
Clark	Mrs	Helen	M B	Church Vestibule	80 Leadenhall Street
Clark	Mrs	Helen	M B	Offices	122 Leadenhall Street
Clark	Mrs	Helen	M C	Physiotherapist	107 Leadenhall Street
Clark	Mrs	Helen	M B	Retail Unit	107 Leadenhall Street
Clark	Mrs	Helen	M B	Snack Bar	104 - 106 Leadenhall Str
Clarke	Mr	Gerald	S	Flat 1	25 Savage Gardens
Clarke	Mrs	Susan			35 Eastcheap
Clark	Mr	Louis	S D		8 Crosby Square
Clarke	Ms	P		Offices	73 Aldgate High Street

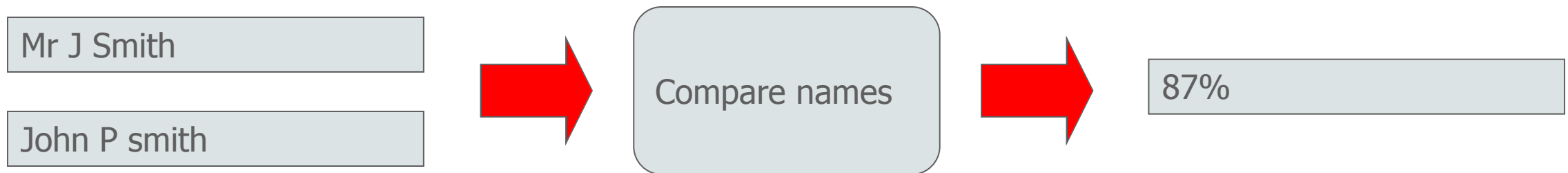
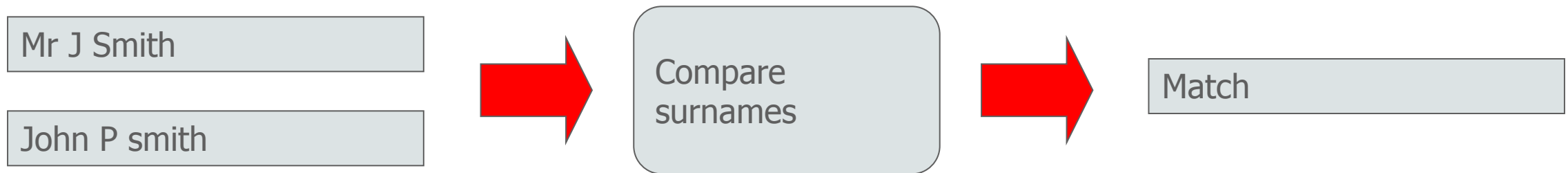
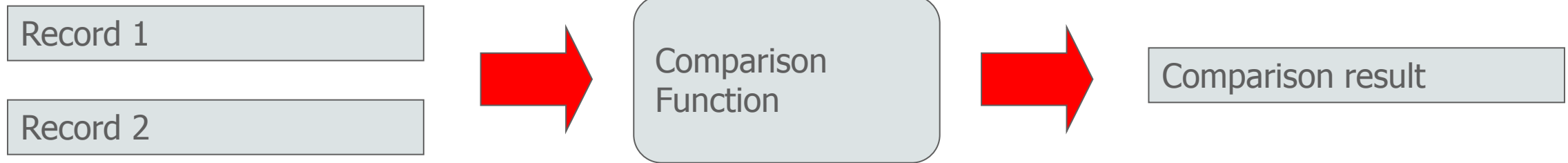
Reminder: Clustering => Performance

- Clustering dictates how much work we get to do in the matching process
- If you have modest volumes, you can afford for the clusters to be simple
- When you have big volumes and tough performance constraints, you need to configure clusters carefully
 - Use multiple clusters
 - Use composite clusters
- Set cluster limits correctly for your maximum data volumes

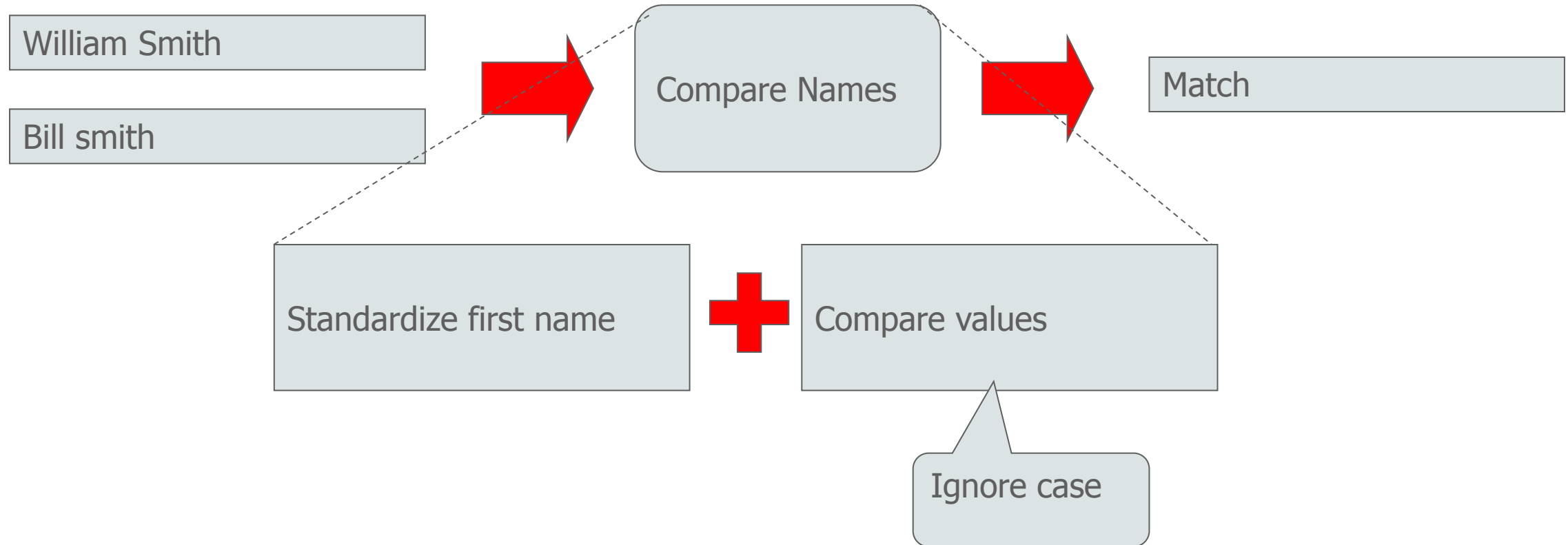
Comparisons and Match Rules



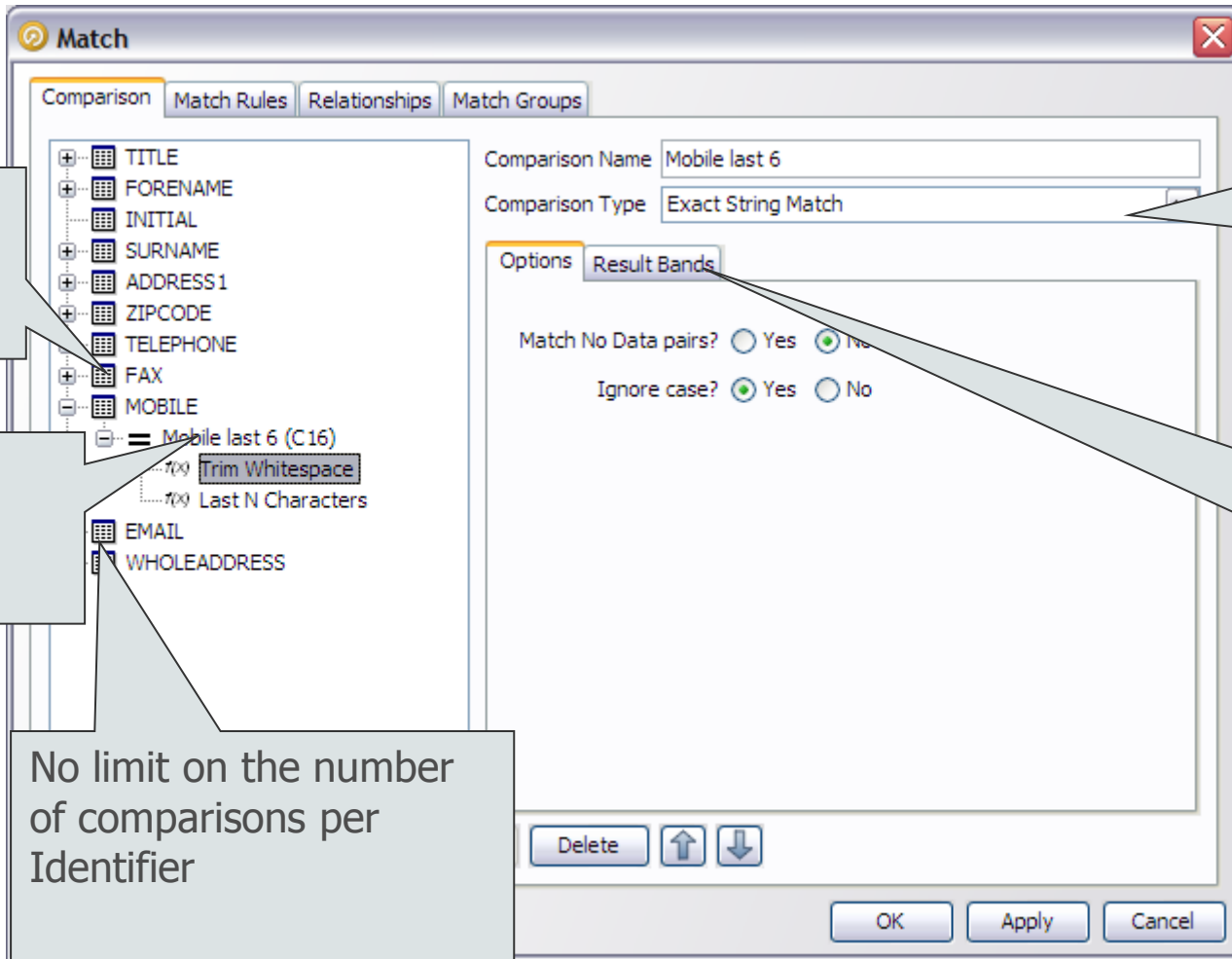
Comparisons



Comparisons Can Include Transforms



Configuring Comparisons



Comparisons apply to Identifiers












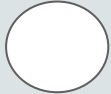
Transformations are applied before the comparison

No limit on the number of comparisons per Identifier

Select the comparison type (algorithm)

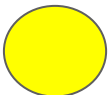
If the comparison gives a numeric answer rather than Yes/No, we define bands for the result

Results of Comparisons = Match Rules

Gender & Initial & Surname	Premise No. & Locality	Postcode	Decision
			Match
			Match
			No Match
			Review



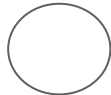
No Match



Close Match

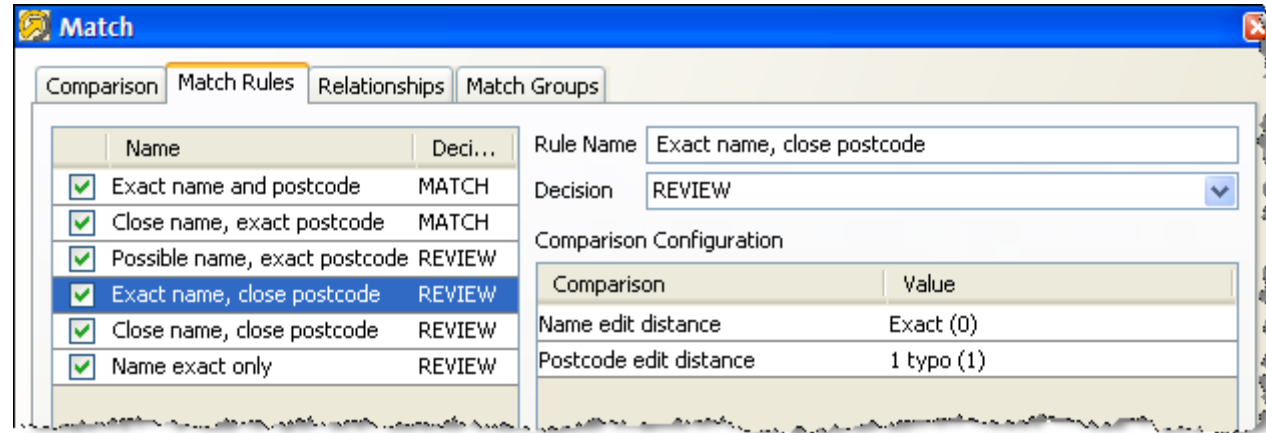


Equivalent



No Data

Configuring Match Rules



- A Match Rule is simply the combination of comparison results
- Rules are evaluated in order when a pair of records are processed and if one hits, we stop
- Rules can be 'negative' to eliminate pairs that are too different with a 'No Match' rule
- Rules can easily be turned on & off during the tuning process
- Each rule can have an optional Priority Score associated with it, normally used where matches are presented in strength order externally to EDQ

Date Matching Tips

- Profile the dates to establish how good they are
 - Date profiler will show unusual distributions
- Consider the source of the data
 - DDMMYY vs MMDDYY formats in text fields?
 - Be careful on translation from text to date
- Useful comparisons:
 - Date Transposition Match (DDMM vs MMDD)
 - Date Edit Distance – number of typos
 - Date Difference

People Matching Tips

- Name is a good starting point, but not as helpful as you would hope
 - Lots of John Smiths
- Date of Birth is very useful, but be sure to check the quality & completeness before you rely on it
- Other supporting information includes things like
 - Address – but people move house
 - Telephone numbers – esp. mobile number
 - E-mail
 - SSN/Passport no if you have them!

Address Matching Tips

- Understand how good your data is
 - Consistency of field use
 - Consistency of spelling
 - Has an address standardisation package been used?
- Some bits of address information are more identifying than others:
 - Building number
 - Apt number
 - Zipcode
- It's often useful to extract these with the parser to create additional attributes to match on
- Using a fuzzy match on a 'whole address' field can deliver great results for some applications

Hardware and Software Engineered to Work Together

ORACLE®