Written Logic for Each Attributes’ Golden Record Creation

One of the possible causes for the noticed duplicates >> customer “imprints” = like a trade name under which customers publish their works.

* Validated Duplicates :: customer wants to create different accounts for different (different imprints) genres/formats they publish – to be able to track sales separately [Ex: children’s book vs adults, eBooks vs audio] in which case, both the records are valid, and are used for tracking different sales
* Technical Issues :: customer created an account >> forgot password >> forgot password feature isn’t working >> new account >> new record}

BPM tables record order management specific records with production team. This can be used for market/ business expansion

* Doesn’t require address lines/banking/tax
* Just has order details and communication
* BPM has customer info of eBooks and Gantec :: maybe the only way to distinguish would be by looking at lead owner [ON A SIDE NOTE: WE ARENT GIVEN LEAD OWNER DETAILS IN THE TABLE ON OUR END]

Updating existing records, there are 2 ways followed:

* Usually, when updating existing records, they are done in CRM over BPM. WHY?
  + CRM is used for marketing/user experiences purposes, updates done here first to not disrupt those processes
  + Two CRMs = InfusionSoft CRM and .net CRM; the one we have is mostly likely InfusionSoft CRM
  + If customer placed an order via a phone call, this customer details would be created/updated in BPM as they would be processing the order
    - Also, if customer only ordered services and not distribution >> customer info will not crm but will be in bpm.
    - Otherwise, they can be in both

Note that some customers are authors, but they also work with a publishing company to publish books.

USUALLY MOSTLY, THE ADDRESS AND PHONE ARE UPDATED IN CRM. ONLY CASE OTHERWISE WOULD BE IF A CUSTOMER WAS ORDER WOULD BE UPDATED IN BPM. BEST WAY ~ MERGE CRM/BPM TABLES, TAKE MOST RECENT ONE?

For each of the source tables, customer\_id to email\_address ratio should ideally be a one-to-one relationship. Hence, when multiple customer\_ids are identified for a single email\_address, following are the predicted scenarios and the proposed plan of action to be taken.

First and Last Names:

* All duplicates with matching names
  + Take the “original”/first listed name
* All duplicates with different names
  + Take most occurring name
  + If Tie, take the “original”/first listed name unless otherwise instructed
* No names recorded
* If all names returned for duplicate records are the same, then take first listed name
* If names for all returned duplicate records are different
  + Take most occurring name
  + If Tie
    - Take name from most recently created record

Last Name:

* If all names returned for duplicate records are the same then take first listed name
* If names for all returned duplicate records are different
  + Take most occurring name
  + If Tie
    - Take name from most recently created record

Address:

* Uses the following fields:
  + Address Line 1
  + Address Line 2
  + City
  + State
  + Country
  + Zip
* Multiple addresses recorded between duplicates
  + Use an entire address of member and send to address validation API
  + eliminate addresses that get recommended other addresses (1 for yes or 0 for no) only compare between the addresses that return “0” in the recommendation column.
  + Use most occurring address
    - If tie, then use latest created date of the record for tie breaker.
* Part of the address mentioned
* No address mentioned

Phone Number:

* Irregular formatting
  + In data cleaning, align all numbers to similar format using regex
* All duplicates containing same phone numbers
  + Use the in first listed phone number
* Multiple phone numbers recorded between duplicates
  + Use the most occurring phone number
  + If tie, use most recently added number
* If all phone #s returned for duplicate records are the same then take first listed phone #
* If phone #s for all returned duplicate records are different
  + Take most occurring phone #s
  + If Tie
    - Take phone #s from most recently created record

Email address:

* Case sensitive database languages
  + Apply (.lower) to get unique email addresses

Publisher name: [We decided to keep the column? Ref table takes care of it.. no?]

* All duplicates with matching names
  + Take the first listed name
* All duplicates with different names
  + (all that are valid and ordered with the company will be included in the ref\_company\_name table)
* No names recorded
* Convert to publisher id column using reference table id in place
* If all names returned for duplicate records are the same, then take first listed name
* If names for all returned duplicate records are different
  + Take most occurring name
  + If Tie
    - Take name from most recently created record

Username:

* All duplicate records appear to have the same email address as their username so, validate and take the most occurring email

Password:

* Duplicates with same password/different passwords
  + Use most recently created password

Deactivated:

* Use most recent created date field value

Role\_id:

* Should always be 1 {filters applied during data cleaning}

Is\_publisher:

* Use most recent created date field value

Website: [How is this relevant?]

* Data validating step
  + Use filtering to only get website that end with “.xxx”
* All duplicates with matching website
  + Take the first listed website
* All duplicates with different websites
  + Take most occurring name
  + If Tie, take the most recent copyrighted website unless otherwise instructed
* No names recorded

Is\_partner:

* Use most recent created date field value

Lead owner:

* Do we need this?

Lead source:

* Do we need this?

Order type:

* Do we need this?

Created Date:

* timestamp

Updated date:

* timestamp