Canonicalization of XML Datasets

SHRASHTI SINGHAL
Data Curation- IS 531
Department of Information Science
The University of Illinois at Urbana Champaign

Abstract

The agency tracks complaints made by customers against financial institutions, like banks and lending companies. The agency has switched systems for managing its complaint data, and this requires a transfer of complaint data from the old system into the new system.

The job is to assess the quality of the data transfer, primarily to ensure the equivalence of the data in the old and new systems. This task requires canonicalizing two files.

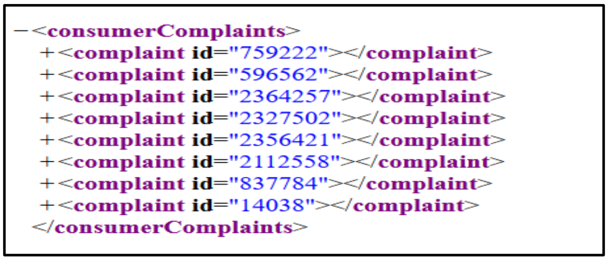
I. Original Database: File A

I.I Description

It is an XML file. The file doesn't contain any DTD or XSD information about the XML file. Line Feeds are used to format the file consistently. This XML file holds complaint information as a list of complaint elements. The file uses attributes to keep information including identifiers, dates, etc. Consumer Complaints File A is the original file with the agency.

Access the Actual File here : Input_Consumer_Com

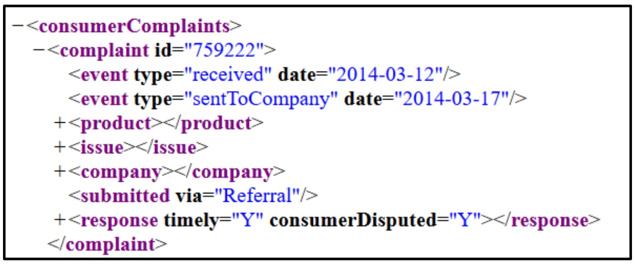
The length of this document is 10764 words and 215 lines of code written in UTF-8 as the encoding in XML format under version 1.0. The DOCTYPE of this file is for consumer complaints that the company handles. The file contains eight complaints. The parent element is consumerComplaint. Entire complaint record is written inside <consumerComplaints> tag. Followed by *complaint id* per complaint. Each record in the file has been uniquely organized and identified by the element complaint id.



```
−<consumerComplaints>
  +<complaint id="759222"></complaint>
  +<complaint id="596562"></complaint>
  +<complaint id="2364257"></complaint>
  +<complaint id="2327502"></complaint>
  +<complaint id="2356421"></complaint>
  +<complaint id="2112558"></complaint>
  +<complaint id="837784"></complaint>
  +<complaint id="14038"></complaint>
 </consumerComplaints>
```

Image 1: Structure of File A.xml

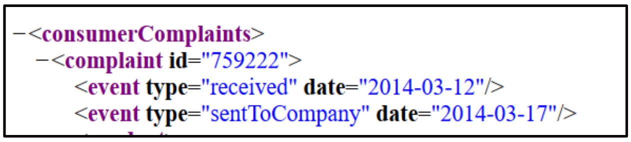
Each record has seven significant elements under the primary record, which are: *event, product, issue, consumerNarrative, company, submitted, response*



```
−<consumerComplaints>
  −<complaint id="759222">
     <event type="received" date="2014-03-12"/>
     <event type="sentToCompany" date="2014-03-17"/>
    +<product></product>
    +<issue></issue>
    +<company></company>
     <submitted via="Referral"/>
    +<response timely="Y" consumerDisputed="Y"></response>
   </complaint>
```

Image 2: Elements & Attributes of each complaint

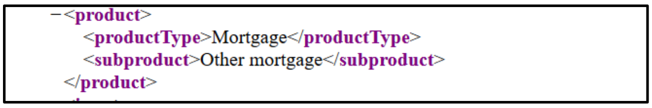
- *event:* Each complaint can have multiple event elements, the type of event is specified in a type attribute. The date attribute defines the date of the event.



```
−<consumerComplaints>
  −<complaint id="759222">
     <event type="received" date="2014-03-12"/>
     <event type="sentToCompany" date="2014-03-17"/>
```

Image 3: Element- event structure

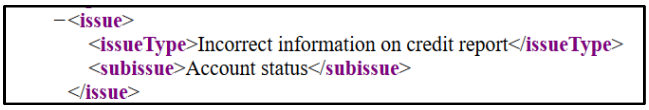
- *product:* Each complaint has a description of the product in a product element. The product element has no attributes for itself but has two child-elements: productType and subProduct.



```
  −<product>
     <productType>Mortgage</productType>
     <subproduct>Other mortgage</subproduct>
   </product>
```

Image 4:Element- product structure

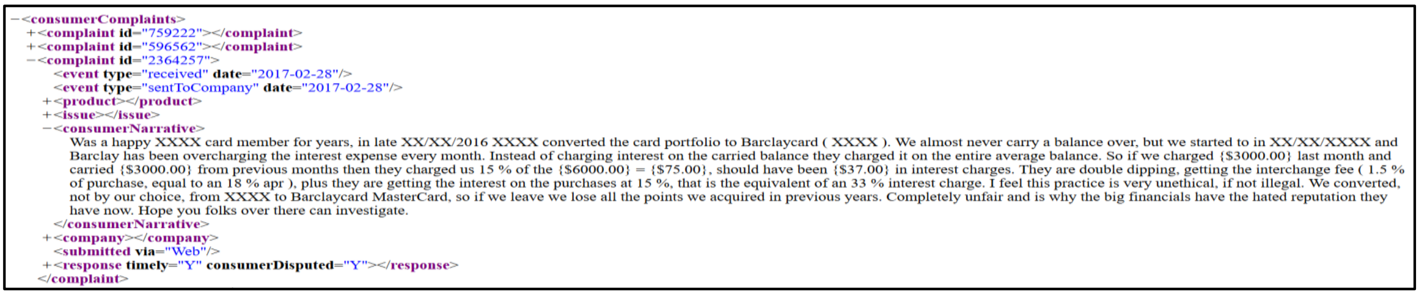
- *issue:* The issue element does not have its own attributes but has a child element called *issueType*. The issueType child element describes the problem. There is an optional child element, which *subissue*, it defines the general issue type.



```
  −<issue>
     <issueType>Incorrect information on credit report</issueType>
     <subissue>Account status</subissue>
   </issue>
```

Image 5: Element: issue structure

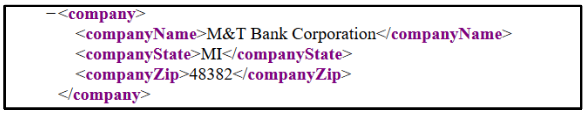
- *consumerNarrative:* It is an optional element. It contains the description of the issue by the consumer.



```
−<consumerComplaints>
  +<complaint id="759222"></complaint>
  +<complaint id="596562"></complaint>
  −<complaint id="2364257">
     <event type="received" date="2017-02-28"/>
     <event type="sentToCompany" date="2017-02-28"/>
    +<product></product>
    +<issue></issue>
    −<consumerNarrative>
       Was a happy XXXX card member for years, in late XX/XX/2016 XXXX converted the card portfolio to Barclaycard ( XXXX ). We almost never carry a balance over, but we started to in XX/XX/XXXX and
       Barclay has been overcharging the interest expense every month. Instead of charging interest on the carried balance they charged it on the entire average balance. So if we charged {$3000.00} last month and
       carried {$3000.00} from previous months then they charged us 15 % of the {$6000.00} = {$75.00}, should have been {$37.00} in interest charges. They are double dipping, getting the interchange fee ( 1.5 %
       of purchase, equal to an 18 % apr ), plus they are getting the interest on the purchases at 15 %, that is the equivalent of an 33 % interest charge. I feel this practice is very unethical, if not illegal. We converted,
       not by our choice, from XXXX to Barclaycard MasterCard, so if we leave we lose all the points we acquired in previous years. Completely unfair and is why the big financials have the hated reputation they
       have now. Hope you folks over there can investigate.
     </consumerNarrative>
    +<company></company>
     <submitted via="Web"/>
    +<response timely="Y" consumerDisputed="Y"></response>
   </complaint>
```

Image 6: Element-consumerNarrative structure

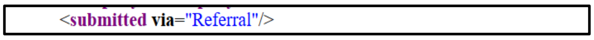
- *company*: The company element has information about the company raised the complaint. The company doesn't have its attributes but have three child-elements, companyName, companyState, companyZip.



```
-<company>
    <companyName>M&T Bank Corporation</companyName>
    <companyState>MI</companyState>
    <companyZip>48382</companyZip>
  </company>
```

Image 7: Element-company structure

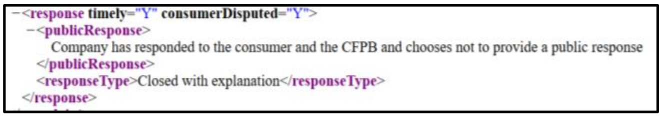
- *submitted:* The submitted element has an attribute via which describes how the complaint was submitted.



```
  <submitted via="Referral"/>
```

Image 8: Element-submitted structure

- *response:* The response element has two attributes timely and consumerDisputed that describe how the answer to the complaint was handled. These attributes values can be "Y" for Yes or "N" for NO. The response element has a child element, responseType which describes the kind of response given by the company for the issue.
  There is an optional child element publicResponse, which states if the public response has been given or not.



```
-<response timely="Y" consumerDisputed="Y">
  -<publicResponse>
      Company has responded to the consumer and the CFPB and chooses not to provide a public response
    </publicResponse>
    <responseType>Closed with explanation</responseType>
  </response>
```

Image 9: Element-response structure

I.II Checksum

A *checksum* is a small-sized datum derived from a block of digital data. The checksum of 2 different files can never be the same. The checksum of the file is calculated to check the equivalence of the databases in the canonicalization process. Checksum is calculated from http://onlinemd5.com/

There are variety of checksums available. For reference purposes, we will report 3 types of checksum, as below:

MD5: 637737835B3639596BF6DB0FA0FFF691

SHA1: BCD232CA1BB39998BF7374850BCD1013D347C960

SHA-256:

A3B46F0FCC94864280A21C66F8AA2FCE06D7FA04C69C3DDE24199BE16D7996BB

I.III Meta Data

Consumer_Complaints_FileA.xml	
Format	xml
Contents	This file contains complaints made by customers against financial institutions, like banks and lending companies. This file is used for storing this complaint records before changing the switching to the new database management system.
MD5 Checksum	637737835b3639596bf6db0fa0fff691
Encoding	UTF-8
Words	10764
Lines	215
Size	10,815 bytes
URL	https://github.com/ShrashtiSinghal/Data-Curation/blob/master/Assignment%204-%20Canonicalizing%20Data/3_Files/1%20Input_Consumer_Complaints_FileA.xml

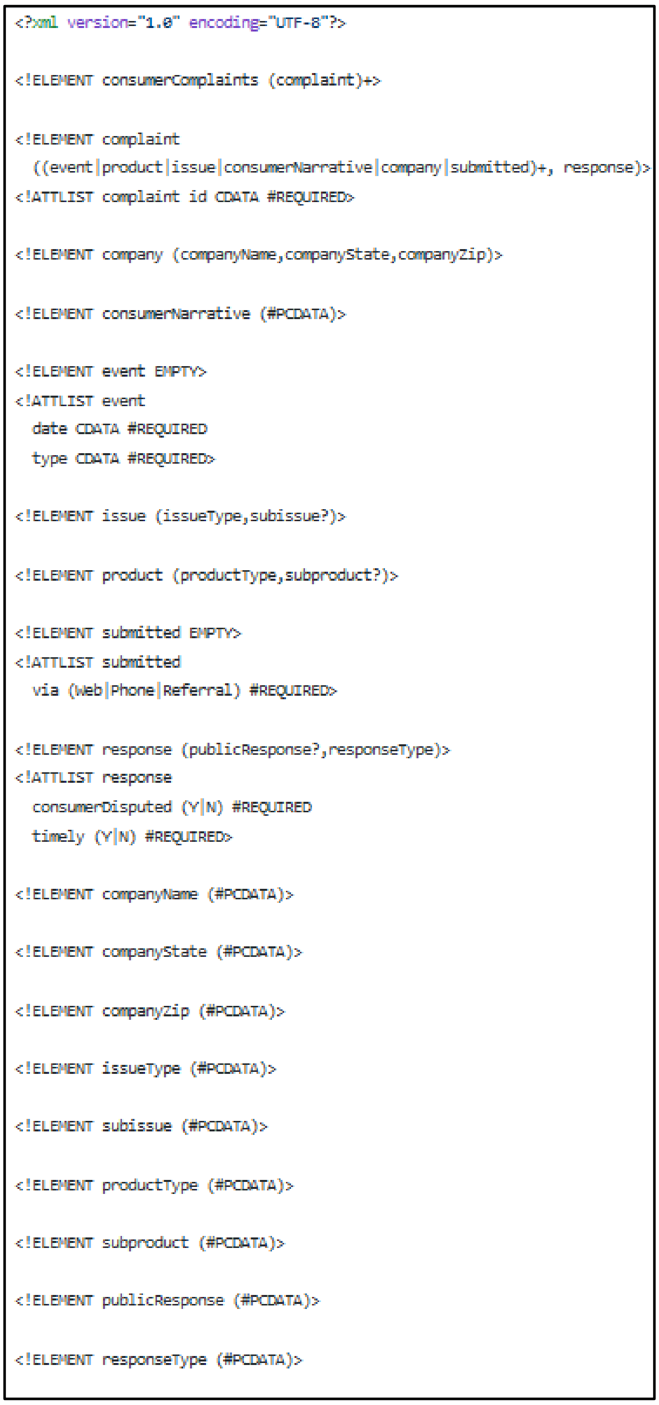
Table 1: Meta Data File A

I.IV DTD

A DTD is designed. All the constraints for the elements and attributes from XML are implemented in the DTD.

- The root element consumerComplaints can have more than one records
- Inside element complaint, its seven child elements are defined.
- consumerDisputed & timely attribute can have only either of 2 values N|Y
- subissue, subProduct and publicResponse are made optional.
- Via attribute can have only one of 3 values, web\phone\Referral.

Access the DTD here : 3 a_dtd.dtd



```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT consumerComplaints (complaint)+>

<!ELEMENT complaint
  ((event|product|issue|consumerNarrative|company|submitted)+, response)>
<!ATTLIST complaint id CDATA #REQUIRED>

<!ELEMENT company (companyName,companyState,companyZip)>

<!ELEMENT consumerNarrative (#PCDATA)>

<!ELEMENT event EMPTY>
<!ATTLIST event
  date CDATA #REQUIRED
  type CDATA #REQUIRED>

<!ELEMENT issue (issueType,subissue?)>

<!ELEMENT product (productType,subproduct?)>

<!ELEMENT submitted EMPTY>
<!ATTLIST submitted
  via (Web|Phone|Referral) #REQUIRED>

<!ELEMENT response (publicResponse?,responseType)>
<!ATTLIST response
  consumerDisputed (Y|N) #REQUIRED
  timely (Y|N) #REQUIRED>

<!ELEMENT companyName (#PCDATA)>

<!ELEMENT companyState (#PCDATA)>

<!ELEMENT companyZip (#PCDATA)>

<!ELEMENT issueType (#PCDATA)>

<!ELEMENT subissue (#PCDATA)>

<!ELEMENT productType (#PCDATA)>

<!ELEMENT subproduct (#PCDATA)>

<!ELEMENT publicResponse (#PCDATA)>

<!ELEMENT responseType (#PCDATA)>
```

Image 10: DTD for File A

II. Transformed Database: File B

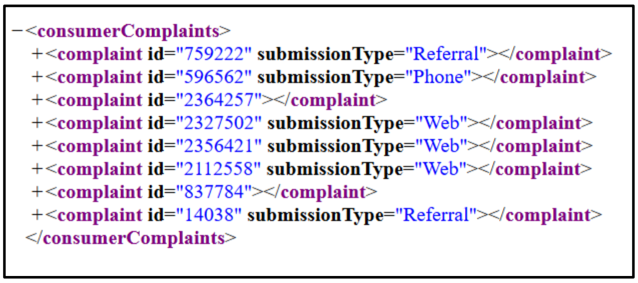
II.I Description

File B is the XML data in the new Data system after the agency switched the data system to manage the complaint records.

The new dataset can be found here: Input_Consumer_Com.

Individual complaint records in the file are identified by complaint id and submission type. Line Feeds do not systematically separate elements, and comments can be found within the XML file. The XML file contains a minimal DTD which consists of an entity definition. This XML file holds complaint information as a list of complaint elements. The file utilizes attributes to keep information including identifiers, submission type, dates, etc. The elements contain product information, company information, response, issue, and events.

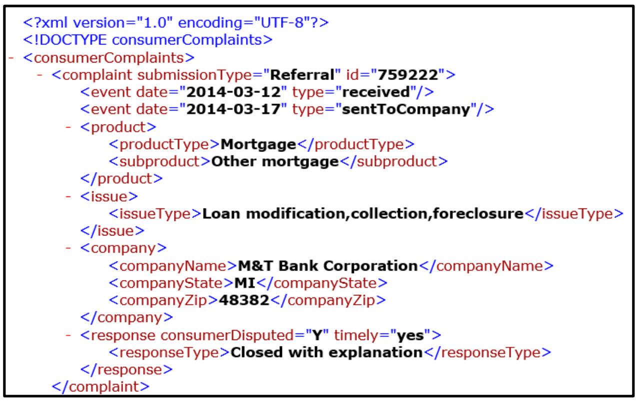
The length of this document is 9876 words and 117 lines of code written in UTF-8 as the encoding in XML format under version 1.0. The file contains eight complaints. The entire complaint record is written inside <consumerComplaints> tags. Each complaint is identified by complaint IDs and its submission type. Submission type isn't included in a few complaint records.



```
-<consumerComplaints>
   +<complaint id="759222" submissionType="Referral"></complaint>
   +<complaint id="596562" submissionType="Phone"></complaint>
   +<complaint id="2364257"></complaint>
   +<complaint id="2327502" submissionType="Web"></complaint>
   +<complaint id="2356421" submissionType="Web"></complaint>
   +<complaint id="2112558" submissionType="Web"></complaint>
   +<complaint id="837784"></complaint>
   +<complaint id="14038" submissionType="Referral"></complaint>
 </consumerComplaints>
```

Image 11: Structure of File B.xml

Each record has seven major elements under the main record, which are: *event, product, issue, consumerNarrative, company, submitted, response.* Here, the submitted element is unused.

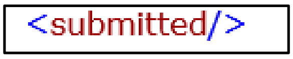


```
  <?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE consumerComplaints>
- <consumerComplaints>
  - <complaint submissionType="Referral" id="759222">
      <event date="2014-03-12" type="received"/>
      <event date="2014-03-17" type="sentToCompany"/>
    - <product>
        <productType>Mortgage</productType>
        <subproduct>Other mortgage</subproduct>
      </product>
    - <issue>
        <issueType>Loan modification,collection,foreclosure</issueType>
      </issue>
    - <company>
        <companyName>M&T Bank Corporation</companyName>
        <companyState>MI</companyState>
        <companyZip>48382</companyZip>
      </company>
    - <response consumerDisputed="Y" timely="yes">
        <responseType>Closed with explanation</responseType>
      </response>
    </complaint>
```

Image 12: Elements & Attributes of each complaint

- *event:* Each complaint can have multiple event elements, the type of event is specified in a type attribute. The date attribute defines the date of the event.

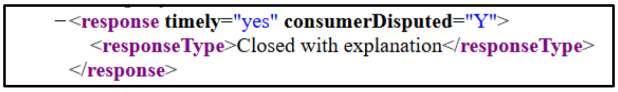
- *product:* Each complaint has a description of the product in a product element. The product element has no attributes for itself but has two child-elements: productType and subProduct.
- *issue:* The issue element does not have its attributes but has a child element called *issueType*. The issueType child element describes the issue. There is an optional child element, which *subissue*, it defines the general issue type.
- *consumerNarrative:* It is an optional element. It contains the description of the issue by the consumer.
- *company*: The company element has information about the company raised the complaint. The company doesn't have its attributes but have three child-elements, companyName, companyState, companyZip.
- *submitted:* The submitted element is mentioned in the file, but it is unused.



```
<submitted/>
```

Image 13: Element-submitted, empty

- *response:* The response element has two attributes timely and consumerDisputed that describe how the response to the complaint was handled. The attributes values for *consumerDisputed* can be "Y" for Yes or "N" for NO. While attribute values for *timely* can be "yes" for Yes or "no" for NO, the response element has a child element, responseType which describes the kind of response given by the company for the issue.
  There is an optional child element publicResponse, which states if the public response has been given or not.



```
-<response timely="yes" consumerDisputed="Y">
   <responseType>Closed with explanation</responseType>
 </response>
```

Image 14: Element response attribute values

II.II Checksum

Similar to File A.xml, we have calculated checksums for File B.xml

MSD5: C2FB08E9A52DC8CD4D7B0C195061C783

SHA1: 2A6912CD8F7DF5E3033BC64E7D3DE977E0AFA552

SHA-256:

94B406413C6C11EAA5826CA895A7038F0333F69B7233B4B2AE9403D9D3C582D1

II.III MetaData

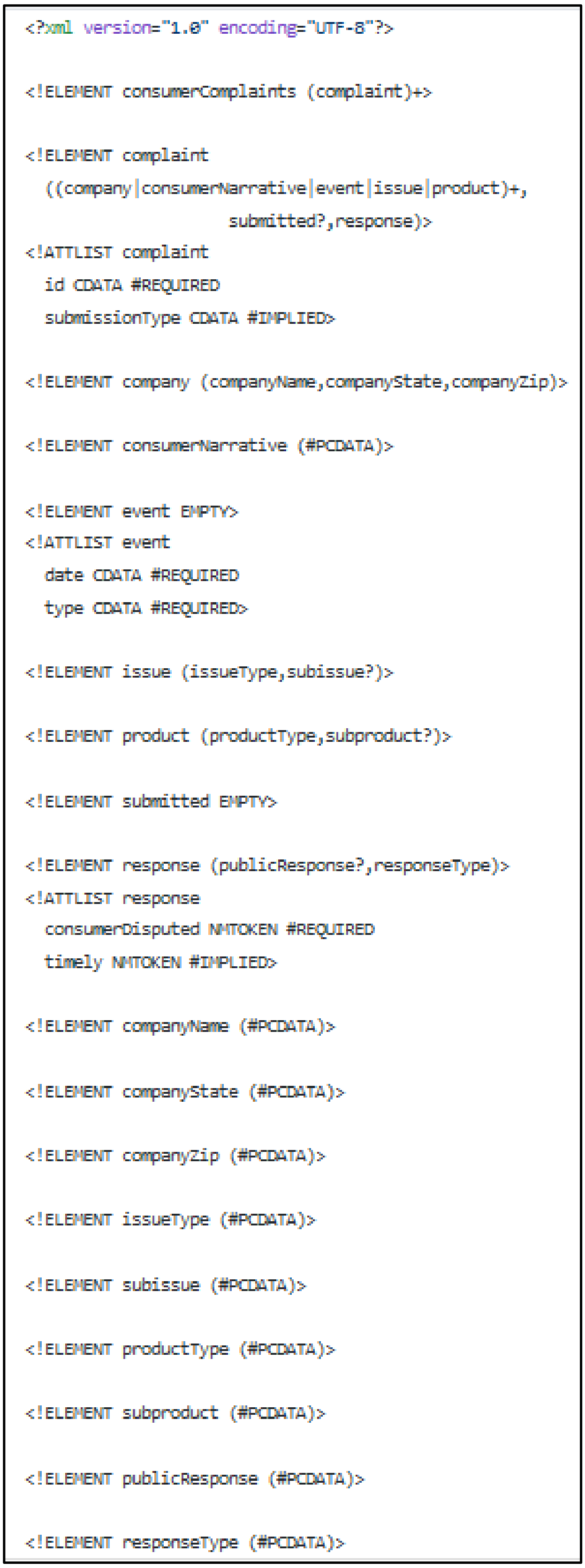
Consumer_Complaints_FileB.xml	
Format	xml
Contents	This file contains complaints made by customers against financial institutions, like banks and lending companies. This file is used by the new system.
MD5 Checksum	C2FB08E9A52DC8CD4D7B0C195061C783
Encoding	UTF-8
Words	9876
Lines	117
Size	9,876 bytes
URL	https://github.com/ShrashtiSinghal/Data-Curation/blob/master/Assignment%204-%20Canonicalizing%20Data/3_Files/2%20Input_Consumer_Complaints_FileB.xml

Table 2: Meta Data File B

II.IV DTD

A DTD is designed for File B. All the constraints for the elements and attributes from XML are implemented in the DTD.

Access DTD here : 4 b_dtd.dtd



```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT consumerComplaints (complaint)+>

<!ELEMENT complaint
  ((company|consumerNarrative|event|issue|product)+,
                  submitted?,response)>
<!ATTLIST complaint
  id CDATA #REQUIRED
  submissionType CDATA #IMPLIED>

<!ELEMENT company (companyName,companyState,companyZip)>

<!ELEMENT consumerNarrative (#PCDATA)>

<!ELEMENT event EMPTY>
<!ATTLIST event
  date CDATA #REQUIRED
  type CDATA #REQUIRED>

<!ELEMENT issue (issueType,subissue?)>

<!ELEMENT product (productType,subproduct?)>

<!ELEMENT submitted EMPTY>

<!ELEMENT response (publicResponse?,responseType)>
<!ATTLIST response
  consumerDisputed NMTOKEN #REQUIRED
  timely NMTOKEN #IMPLIED>

<!ELEMENT companyName (#PCDATA)>

<!ELEMENT companyState (#PCDATA)>

<!ELEMENT companyZip (#PCDATA)>

<!ELEMENT issueType (#PCDATA)>

<!ELEMENT subissue (#PCDATA)>

<!ELEMENT productType (#PCDATA)>

<!ELEMENT subproduct (#PCDATA)>

<!ELEMENT publicResponse (#PCDATA)>

<!ELEMENT responseType (#PCDATA)>
```

Image 15: DTD of File B.xml

III. Analysis

III.I Differences in 2 Databases

The checksum of two XMLs yields that the two files are different. Overall the details in the two files might look same, but there are differences in their representation styles, and usage of elements and attributes.

File A	File B
Elements & Attributes	
Each complaint record is identified by attribute id.	Each complaint record is identified by an ID and optional submitted type attributes
The submitted element is used with attributed via, to state the method by which complaint had been registered.	The submitted element is defined, but it is never used.
Via attribute had three kinds of values, phone/web/Referral	Via attribute is missing
The timely attribute has values "Y" and "N"	The timely attribute has values "yes" and "no"
submissionType attribute is missing.	submissionType attribute used to state the method by which complaint had been registered.
No comments found in the file.	There are comments in the file.
DOCTYPE Missing	DOCTYPE consumerComplaints
Formatting	
The file is consistently indented by TAB SPACE.	The file is indented inconsistently by spaces.
Child elements definition is in a new line.	Child elements in continuation with the parent element in the same line.
Attributes of event element are ordered as type first and date second.	Attributes of event elements are unordered. Type and date are randomly ordered inside the element event.
Attributes values don't have trailing or leading spaces.	Attributes values have trailing or leading spaces.

Table 3: Differences between datasets A & B

III.II Standard Practices Problems

There are general problems with both the databases File A and File B. The design and formatting of the two databases aren't consistent. To design a good, accurate, reproducible and healthy dataset, we will need to improve the overall design of the database, and standardize it to establish general formatting, syntax and quality standards.

Problem Type	Description	Example
Order of Attributes	We need to follow an order to arrange attributes inside the elements	FileA follows the arrangement of type first & date second in element-event, while FileB randomly interchange the order Same goes for timely and consumerDisputed attributes inside Element-response.
Elements Order	We need to follow an order to arrange the elements inside each complaint record	<complaint id="14038">, Element-company is the first element in this record, while in others company element is at 4th place
Carriage Return	All attributes values should be in one line. It makes the file more readable.	consumerNarrative attribute value, which is generally multi-line description, uses carriage return to go to the next line. Same goes for publicResponse attribute
Order of Tags	Opening and closing of tags should follow an order.	Generally, child Elements opens and closes in the same line. While parent elements open and close in separate lines. This rule hasn't been followed in case of consumerNarrative tags.

Table 4: General Standard Issues with Datasets A & B

IV. Canonicalization

To match the two databases for equivalence and to determine whether or not two XML files define the same data structure, we will need to canonicalize them.

Canonicalization is a technique for determining the representational identity and is a reasonable proxy for propositional identity.

As syntax of 2 datasets is the same, but still, there are differences in design, attributes, elements, formatting, encoding, printing conventions, etc.

IV. I Canonicalization Process Steps:

1. **Convert to a single character encoding and normalize line ends.**
   - Ensured that both the documents are encoded in UTF-8.
   - Line breaks are removed from attributes publicResponse & ConsumerNarrative, to arrange in 1 line.

2. **Remove all comments, tabs, non-significant spaces, etc.**
   - Comments are removed from File B.
   - All spaces separating elements and attributes are converted to Tabs.

3. **Propagate all attribute defaults indicated in the schema to the elements themselves**
   - All attributes are converted into Elements.- There are few problems with the attributes as attributes cannot contain multiple values, attributes are not easily expandable for future changes, attributes cannot describe structures (while child elements can), attributes are more difficult to manipulate by program code, and attribute values are not easy to test against a DTD.
   - Attributes type & date of Element event are converted to sendtocompany_date & recieved_date child Elements of the event element.
   - Via attribute of Element submitted is converted to element submittedVia Element. Removed submitted element from File A, removed via attribute from File A and submissionType attribute from File B.
   - The id attribute of Element complaint is converted to element ID.
   - consumerDisputed & timely attributes of element response are converted to the child elements.

4. **Put attribute/value pairs on elements in alpha order**
   - As all attributes are converted to either elements or child elements, all the parent elements are arranged in alphabetically ordered.
   - All the child elements inside the parent elements are also arranged in alphabetically ordered.

5. **Expand all character references**
   - Aliases entity are expanded in file B. Entity redaction is expanded to XXXX at all instances in File B.

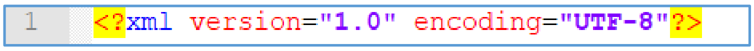
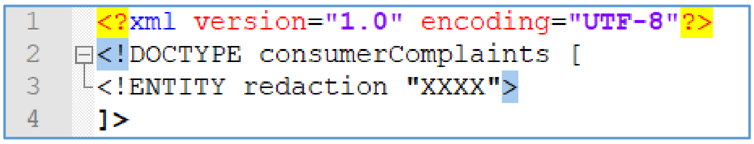
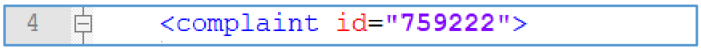
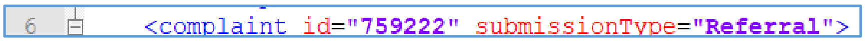
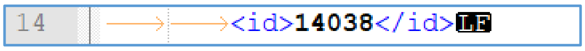
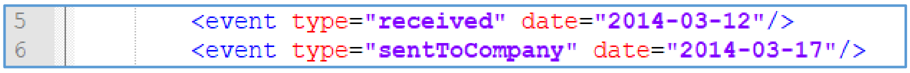
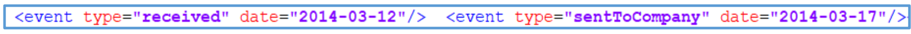
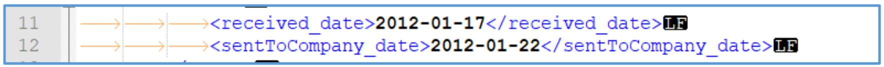
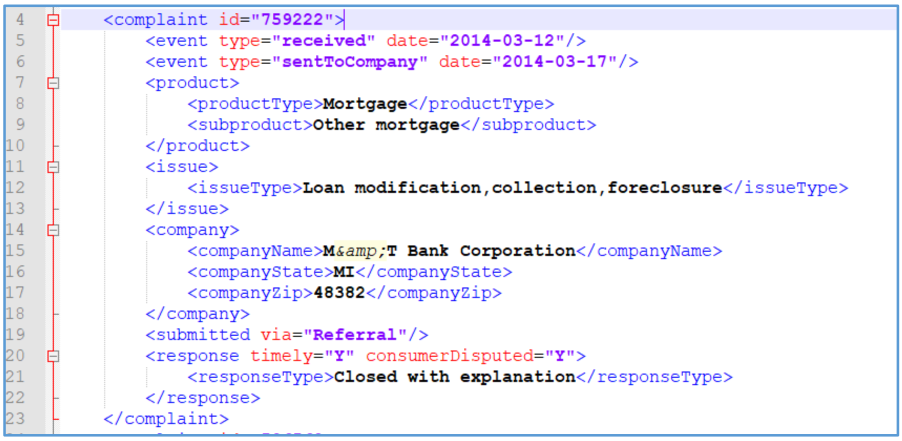
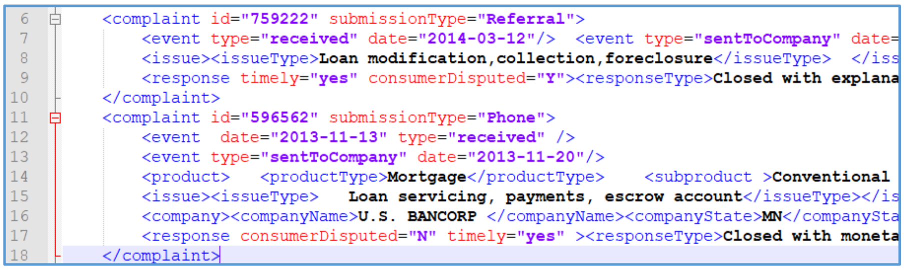
6. **Remove any internal schema or declarations.**
   - XML version and encoding declaration is removed from File A
   - XML version, XML encoding, DOCTYPE, Entity Aliases reference is removed from File B.

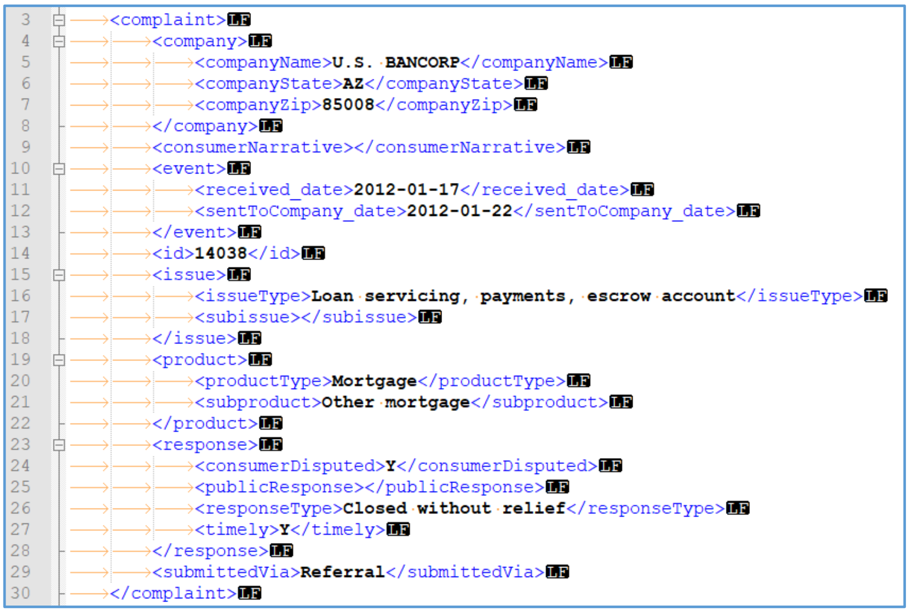
7. **Now test to see if character sequences are identical.**
   http://onlinemd5.com/
   is used to check the MD% checksum of the two canonicalized files.

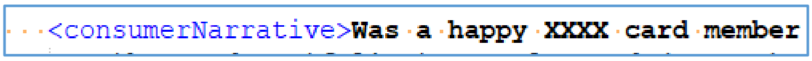
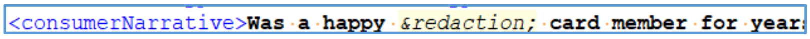
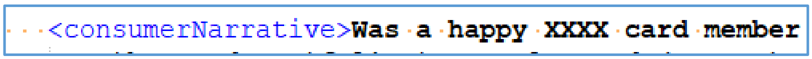
IV.II Canonicalization Script/Tools

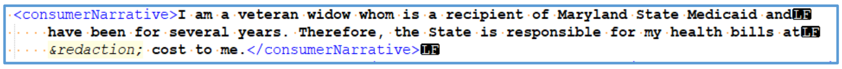
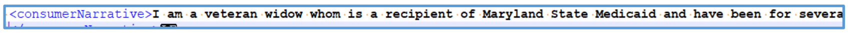
- Conversion of attributes to Elements is done manually.
- Notepad++, replace all is used to convert black spaces to TAB Spaces.
- Notepad++, view all characters, is used to display line breaks, tabs, and spaces. Line breaks are then deleted.
- Text-compare.com is used to compare the various portions of the files for alphabetical orderings, spacing, and deletion.

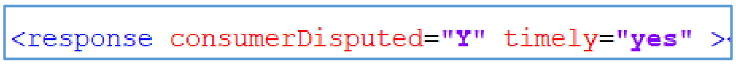
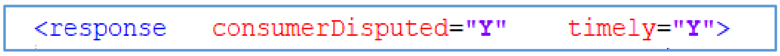
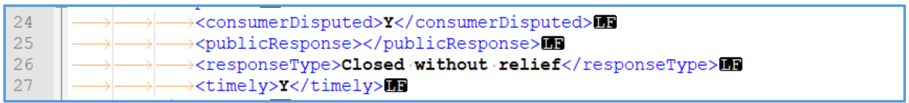
IV.III. Implementation of Canonicalization Steps

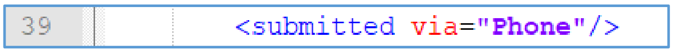
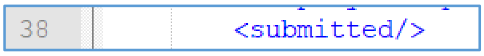
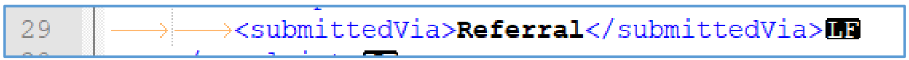
S.NO.	Description	Screenshots	Modification
1	XML Declaration of version & encoding DOCTYPE ENTITY	**File A**  <?xml version="1.0" encoding="UTF-8"?> **File B**  <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE consumerComplaints [<!ENTITY redaction "XXXX">]>	Removed all these headers
2	Each complaint record Identification	**File A**  <complaint id="759222"> **File B**  <complaint id="759222" submissionType="Referral"> **Canonicalized File A & B**  <id>14038</id>LF	Just keep the attribute ID as Element ID. Removed submissionType attribute
3	Order of attributes/ Elements	**File A**  <event type="received" date="2014-03-12"/> <event type="sentToCompany" date="2014-03-17"/> **File B**  <event type="received" date="2014-03-12"/> <event type="sentToCompany" date="2014-03-17"/> **Canonicalized File A & B**  <received_date>2012-01-17</received_date>LF <sentToCompany_date>2012-01-22</sentToCompany_date>LF	Lexicographic ordering of attributes. Now all attributes are elements. Order is 1. received_date 2. sentToCompany_date
4	Indentation Spacing Tags	**File A**  <complaint id="759222"> <event type="received" date="2014-03-12"/> <event type="sentToCompany" date="2014-03-17"/> <product> <productType>Mortgage</productType> <subproduct>Other mortgage</subproduct> </product> <issue> <issueType>Loan modification,collection,foreclosure</issueType> </issue> <company> <companyName>M&amp;T Bank Corporation</companyName> <companyState>MI</companyState> <companyZip>48382</companyZip> </company> <submitted via="Referral"/> <response timely="Y" consumerDisputed="Y"> <responseType>Closed with explanation</responseType> </response> </complaint> **File B**  <complaint id="759222" submissionType="Referral"> <event type="received" date="2014-03-12"/> <event type="sentToCompany" date= <issue><issueType>Loan modification,collection,foreclosure</issueType> </iss <response timely="yes" consumerDisputed="Y"><responseType>Closed with explana </complaint> <complaint id="596562" submissionType="Phone"> <event date="2013-11-13" type="received" /> <event type="sentToCompany" date="2013-11-20"/> <product> <productType>Mortgage</productType> <subproduct >Conventional <issue><issueType> Loan servicing, payments, escrow account</issueType></is <company><companyName>U.S. BANCORP </companyName><companyState>MN</companySta <response consumerDisputed="N" timely="yes" ><responseType>Closed with moneta </complaint>	XML will follow a hierarchical order. At each hierarchical order, the elements are tag indented. All attributes are converted into Elements, which lie in separate lines. No use of space bars. Parent elements, tags open and close in new line Child elements or Elements without any child, tags open and close in the same line

		Canonicalized File A & B
		
		`<complaint>`
		`<company>`
		`<companyName>U.S. BANCORP</companyName>`
		`<companyState>AZ</companyState>`
		`<companyZip>85008</companyZip>`
		`</company>`
		`<consumerNarrative></consumerNarrative>`
		`<event>`
		`<received_date>2012-01-17</received_date>`
		`<sentToCompany_date>2012-01-22</sentToCompany_date>`
		`</event>`
		`<id>14038</id>`
		`<issue>`
		`<issueType>Loan servicing, payments, escrow account</issueType>`
		`<subissue></subissue>`
		`</issue>`
		`<product>`
		`<productType>Mortgage</productType>`
		`<subproduct>Other mortgage</subproduct>`
		`</product>`
		`<response>`
		`<consumerDisputed>Y</consumerDisputed>`
		`<publicResponse></publicResponse>`
		`<responseType>Closed without relief</responseType>`
		`<timely>Y</timely>`
		`</response>`
		`<submittedVia>Referral</submittedVia>`
		`</complaint>`	

5	Entity aliases	**File A**
		
		`<consumerNarrative>Was a happy XXXX card member`
		File B
		
		`<consumerNarrative>Was a happy &redaction; card member for year:`
		Canonicalized File A & B
		
		`<consumerNarrative>Was a happy XXXX card member`
			Entity redaction is deleted.
			XXXX is kept

6	Line breaks	**File A & File B**
		
		`<consumerNarrative>I am a veteran widow whom is a recipient of Maryland State Medicaid and`
		`have been for several years. Therefore, the State is responsible for my health bills at`
		`&redaction; cost to me.</consumerNarrative>`
		Canonicalized File A & B
		
		`<consumerNarrative>I am a veteran widow whom is a recipient of Maryland State Medicaid and have been for severa`
			Line Breaks are removed from each attribute values.
			All attribute values are ensured to occupy just 1 line

7	Attribute Values	**File A**
		
		`<response consumerDisputed="Y" timely="yes" >`
		File B
		
		`<response consumerDisputed="Y" timely="Y">`
		Canonicalized File A & B
		
		`<consumerDisputed>Y</consumerDisputed>`
		`<publicResponse></publicResponse>`
		`<responseType>Closed without relief</responseType>`
		`<timely>Y</timely>`
			consumerDisputed & timely attributes , which are now Elements, values are fixed to either "Y" or "N"

8	Elements	**File A**
		
		`<submitted via="Phone"/>`
		File B
		
		`<submitted/>`
		Canonicalized File A & B
		
		`<submittedVia>Referral</submittedVia>`
			submitted Elements is retained as submittedVia Element.

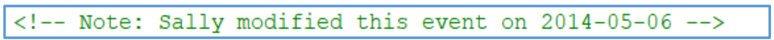
9	Comments	**File B**  `<!-- Note: Sally modified this event on 2014-05-06 -->`	All comments are deleted

Table 5: Implementation Steps of Canonicalization in Datasets A & B

IV.IV DTD Generation

- DTD is generated after Canonicalization of Files using http://xml.mherman.org/index.php/trang/generate
- After automatic generation of DTDs, few redundant attributes and declarations were removed.
- DTDs are embedded into canonicalized files.
- Canonicalized XMLs are validated against the internal DTD using the online validator, using http://xmlvalidator.new-studio.org/

Access all the physical files below:

	File A	File B
Canonicalized	Canonicalized A NO DTD.xml	Canonicalized B NO DTD.xml
Canonicalized with DTD	Canonicalized A with DTD.xml	Canonicalized B with DTD.xml
Canonicalized with DTD and verified	canonicalized A validated.pdf	canonicalized B validated.pdf

Table 6: Physical Files after Canonicalization

IV.V Checksum of Canonicalized Files

Even after the canonicalization steps, we noticed that the checksums of the two canonicalized files do not match.

S.No.	Canonicalized A	Canonicalized B
MD5	730931E27400E2AE0CA932C3B1D1D186	02256D7B91ABDE261F02D9BE9A4B9C1E
SHA1	15285E28F6A691D3CEB405F1718F4B7DCF8CF2C6	847152B1C2A72DF0C08D7F7DDA9EE990CAF6387C
SHA-256	0C6C606B90CB7F9E9F51AF5DDCF6BA89C4A6F111DC40F6F585BC495F1E3DD625	599395D4741D70C8000C2EB1D99F987BA844D3083F2C8A6106E659A1D06A8B21

Table 7: Checksums of Datasets A & B after Canonicalization

V. Differences in Datasets

The two datasets yield different checksums. The datasets are not equivalent.

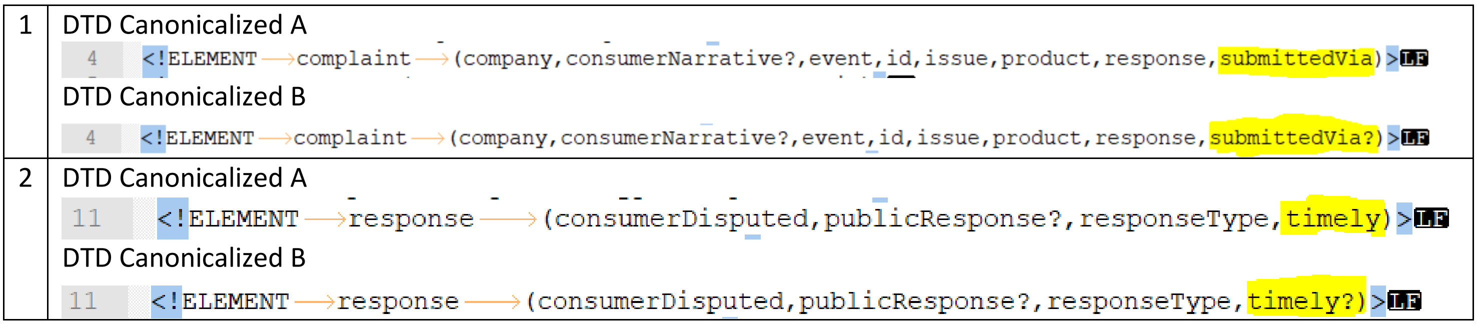
As the files are not equivalent after canonicalization, we would look into the differences in DTD to know about the differences in the datasets.

Difference in DTDs

DTD shows that elements timely and submittedVia are optional in Canonicalized File B while these elements are compulsory in canonicalized File A. This can happen only when the datasets have different values at some point in them. Therefore, at few places values of timely and submittedVia are missing in File B. The analysis had shown that there are 4 Data values missing in File B, which are present in File A. This indicates that the changing of the data management system lost values from the old database.

File A	File B
complaint id="2364257", submitted method by web.	complaint id="2364257", submitted method is missing.
complaint id="837784" submitted method by web	complaint id="837784", submitted method is missing.
Complaint id="837784", timely attribute value is "Y"	Complaint id="837784", timely attribute value missing.
complaint id="14038", timely attribute value is "Y"	complaint id="14038", timely attribute value missing.

Table 8: Difference in Data Values of Datasets A & B



1 DTD Canonicalized A

```
<!ELEMENT	complaint	(company,consumerNarrative?,event,id,issue,product,response,submittedVia)>LF
```

DTD Canonicalized B

```
<!ELEMENT	complaint	(company,consumerNarrative?,event,id,issue,product,response,submittedVia?)>LF
```

2 DTD Canonicalized A

```
<!ELEMENT	response	(consumerDisputed,publicResponse?,responseType,timely)>LF
```

DTD Canonicalized B

```
<!ELEMENT	response	(consumerDisputed,publicResponse?,responseType,timely?)>LF
```

Image 16: Difference in DTDs of canonicalized datasets A & B

V. Making Equivalent Files

- Add the four missing values in canonicalized file B v2.
- Generate DTD of File B
- Now Canonicalized file B generates the same DTD and Canonicalized File A.
- Check Checksum of the two canonicalized file with added data in File B.
- The checksum of both Canonicalized File A & B is same.

Checksum Type	Final Canonicalized File
MD5	730931E27400E2AE0CA932C3B1D1D186
SHA1	15285E28F6A691D3CEB405F1718F4B7DCF8CF2C6
SHA-256	0C6C606B90CB7F9E9F51AF5DDCF6BA89C4A6F111DC40F6F585BC495F1E3DD625

Table 9: Checksum of Final XML Dataset

VII. Conclusion

The job was to assess the quality of the data transfer from one dataset to another, primarily to ensure equivalence of the data in the old and new systems. For this process, we needed to implement canonicalization of both the datasets and perform checksums.

After performing canonicalization on the two datasets, we found the **datasets are not equivalent**.
Later, we found that the new dataset missed 4 data values. After fixing the missing values in the new dataset and performing checksums, we found that the two datasets are equivalent.

The goal of this process was to provide the data in a standard format that could be generically used.
The dataset values were systematically refined to ensure a consistent mapping strategy.
The final database, which is now canonicalized, is in standard form, readable, reusable form and contains all the vital data values.

Creation of this dataset, after getting the correct checksum for the provided two databases, we have addressed two issues.

- Canonicalized 2 datasets perfectly, to implement common standard among both.
- Ensured both the datasets, now contain the same data, and there is no missing data and no mismatch of data.

Final Canonicalized
Database with DTD.xn

VI. Questions

1. Describe your process for canonicalization (i.e., decisions, actions, representation selection, attribute issues, provenance decisions). Report the checksum values after canonicalization.

- DTD is created for both the datasets, A & B
- Individual checksums are calculated
- Checksums didn't match
- Canonicalization is performed, on File A & B
- DTDs are developed for both A & B
- Files are validated against the DTDs.
- Individual checksums are calculated for both the canonicalized files.
- Checksum still doesn't match
- DTDs are tested for data mismatch
- Few lines of data were missing in file B.
- Missing data was added to canonicalized File B.
- Checksums of canonicalized File A and canonicalized File b v2 are checked.
- Checksum Matches.
- We get our final database, which is canonicalized and contain all the data.

For detailed Canonicalization process, **click here**

2. How does the way data is represented impact reproducibility?

Reproducibility means the system or process supports the ability to reproduce results, ensuring scientific validity and reliability. This involves data collection, data management, analysis and documenting every process involved.

The data is reproducible because:
1. The process through which NEW data is created, is documented. Following the same process will yield the same result.
2. All data in OLD was accounted for in NEW, unharmed
3. No data in OLD was lost while transferring to NEW
4. No New elements were added in NEW that was not accounted for in the OLD document.

The canonicalization process will help the new data files with similar prepositional content to lead to same results, because the canonicalized representation of dataset take care of variants that might exist in new files, such as,

- order of elements
- attributes within elements
- space
- new lines
- tabs inside the file
- entities

Moreover, the validity of the contents of an XML can now be checked using the DTD which boosts the reliability of this representation.

There is a robust schema specification applied to the dataset. This standard schema can be imposed on new datasets through a code, which can easily compare equivalence, data mismatch, missing data and order of data. This will also allow us to reproduce data every time.

3. How may your canonicalization support the overarching goals of data curation (revisit objectives and activities of Week 1)?

Canonicalization has supported the below goals/activities of Data Curation:

- **Organization:** Organization is determination of an appropriate data model and schema. Various kinds of standards have been adopted such as XML, XMLT, and DTD. Final document schema is a canonicalized internal DTD and XML. Canonicalization process resulted in well-structured, formatted and standard model which helped in Organization of data.
- **Preservation:** Canonicalization process produced the data model which is easy to understand and reuse in future. This helped us achieve data curation goal of Preservation.
- **Discoverability:** Canonicalization standardized the elements order in XML. The ability to search for and locate relevant data easily is achieved by the help of canonicalized schema.

- **Integration:** Integration of data from different sources using different data models has been done by using canonicalized XMLs and DTDs. We could quickly find the data mismatch in 2 datasets after canonicalization, which we later added to the dataset, missing the data, to make them equivalent.
- **Reproducibility:** The canonicalization process converted the data set into a standard format. If same or new dataset is presented, and made to go through this canonicalization process, it will result in accurate and same results.

  Our integration process supports the ability to reproduce results, ensuring scientific validity and reliability. Data curation for reproducibility included documenting not only data collection and management but also documenting processing and analysis.
- **Provenance:** The canonicalized XML generated DTD, which ensured data constraints and kind of data types used for particular attributes. The entire process from input to output is documented to support identification of inputs, calculations, and actions are responsible for data values. The data set B, is derived from another, file A, through canonicalization, which helped in provenance.

This will allow efficient and reliable support to the analysis of data, and will also enable reuse over time. How these activities have been incorporated and have enhanced the database design is mentioned above in this paper.

4. Which additional curation activities would you recommend enhancing the data set for future discovery and use?

- **Security:** Data can be encrypted to ensure that data is secure from tampering or inappropriate access and distribution.
- **Modification:** Versioning of schema helps in the appropriate and accurate modification of files. Versioning also supports management, corrections, and updates of datasets.
- **Compliance:** This data can be verified against local, industrial and federal laws. It will ensure compliance of these complaint records.