Creating metadata out of thin air and managing large batch imports

Linda Newman

Digital Projects Coordinator

University of Cincinnati Libraries

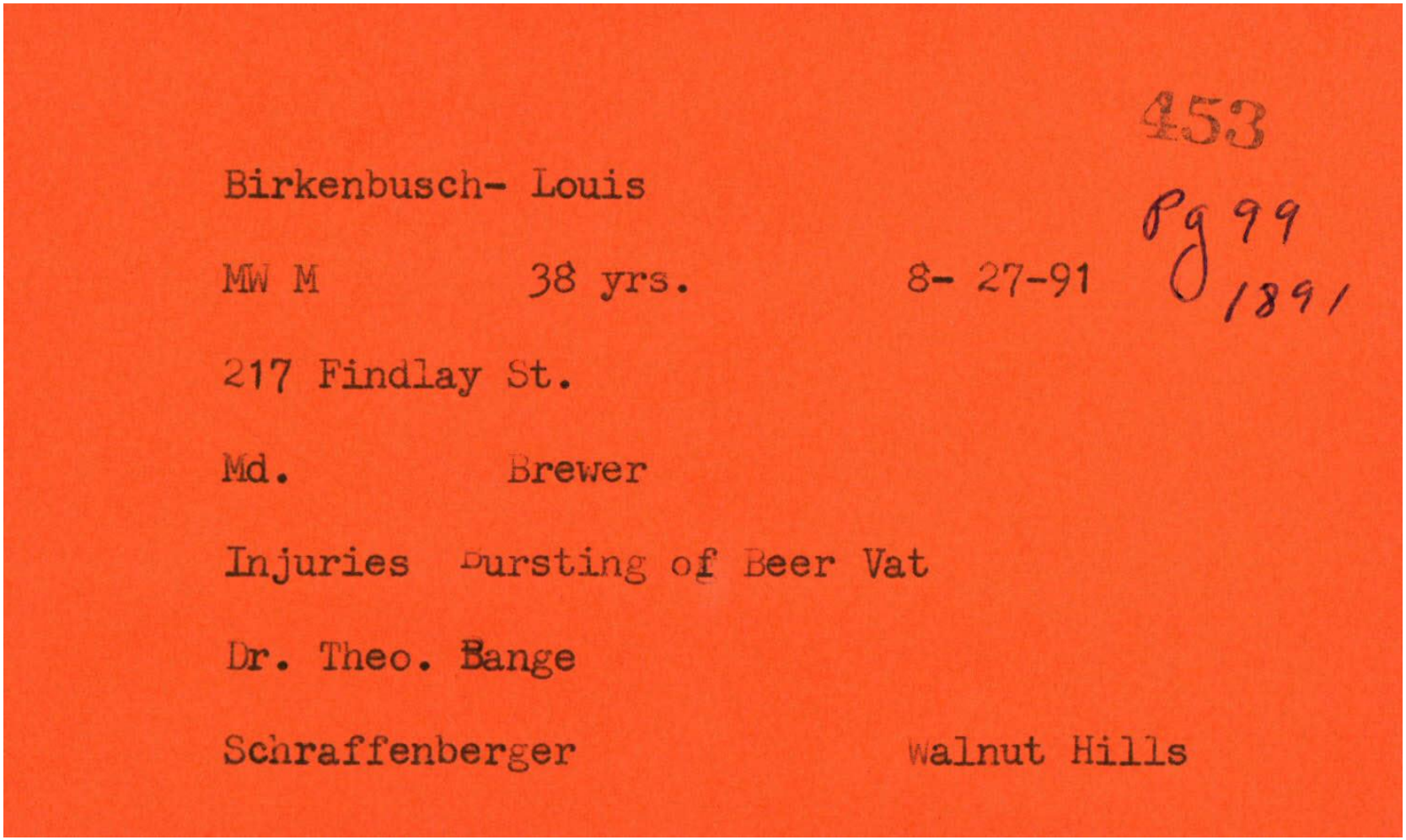
The Project

- The University of Cincinnati Libraries processed a large (over 500,000 items) collection of birth and death records from the city of Cincinnati from 1865-1912, and successfully created dublin_core.xml (“Simple Archive Format”) submission packages from spreadsheets with minimal information, using batch methods to create a 524,360 record DSpace community, in the OhioLINK Digital Resource Commons (DRC)

The Resource

http://drc.libraries.uc.edu/handle/2374.UC/2032

The source records



453

Pg 99

1891

Birkenbusch- Louis

MW M 38 yrs. 8- 27-91

217 Findlay St.

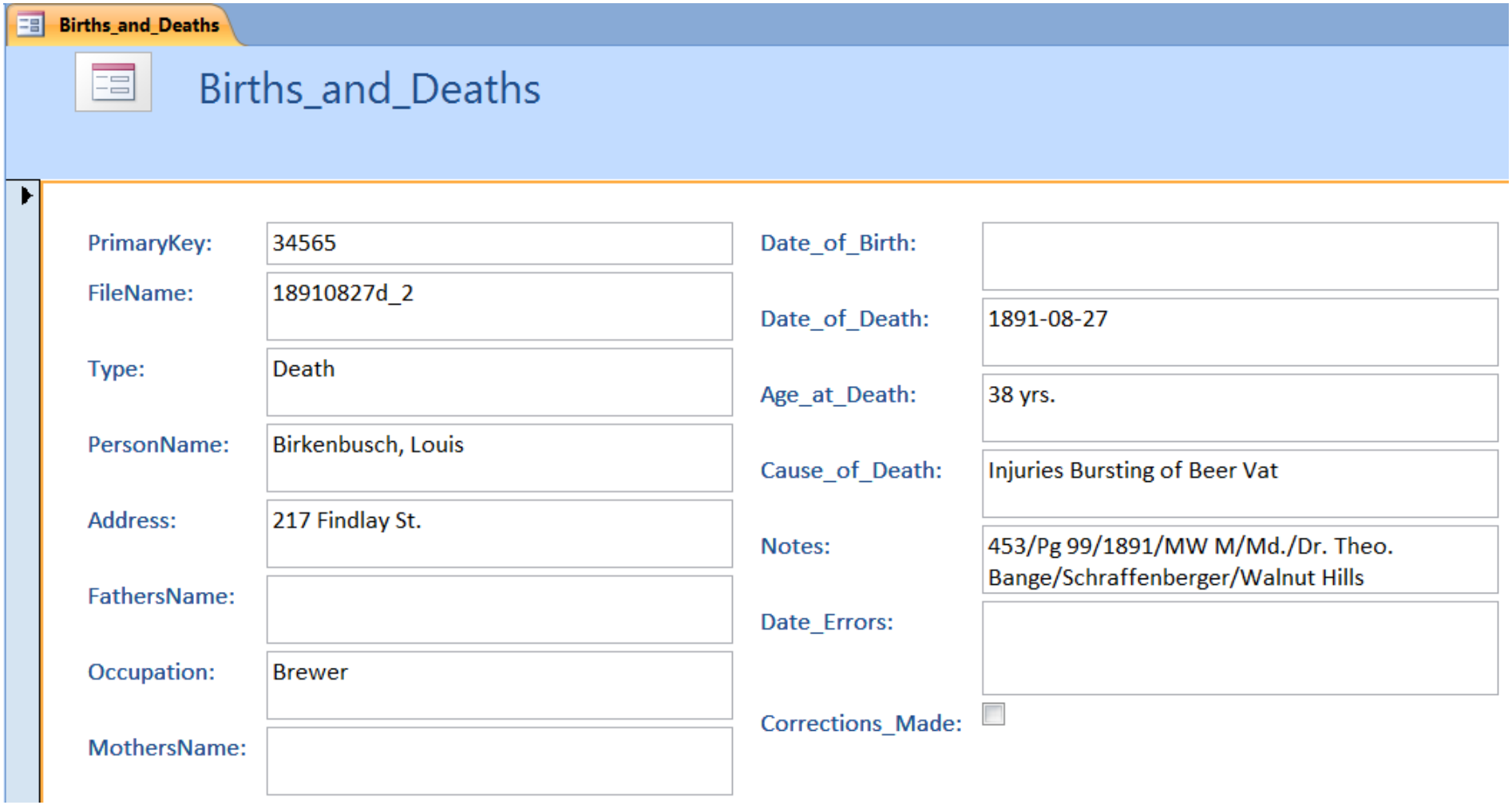
Md. Brewer

Injuries Bursting of Beer Vat

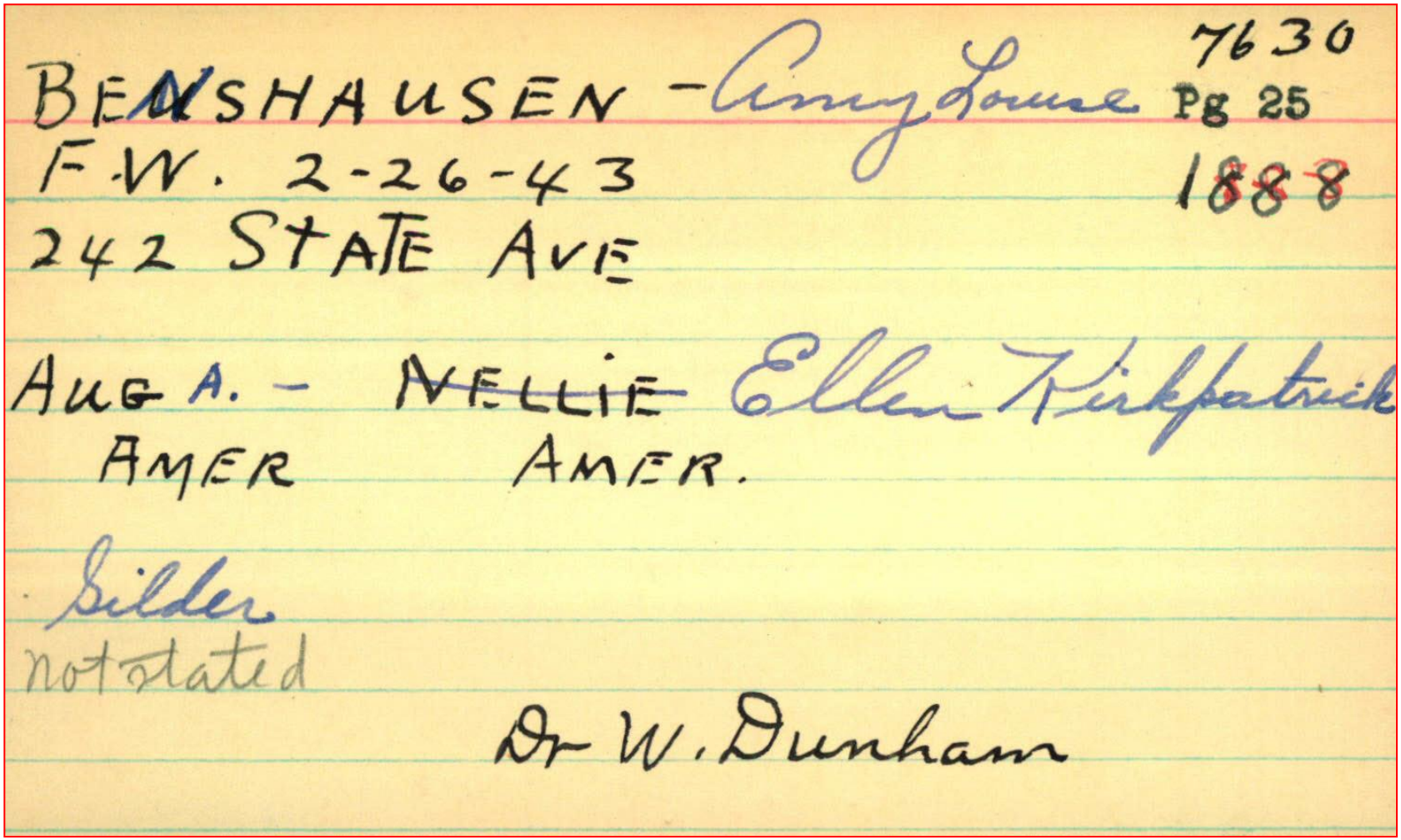
Dr. Theo. Bange

Schraffenberger Walnut Hills

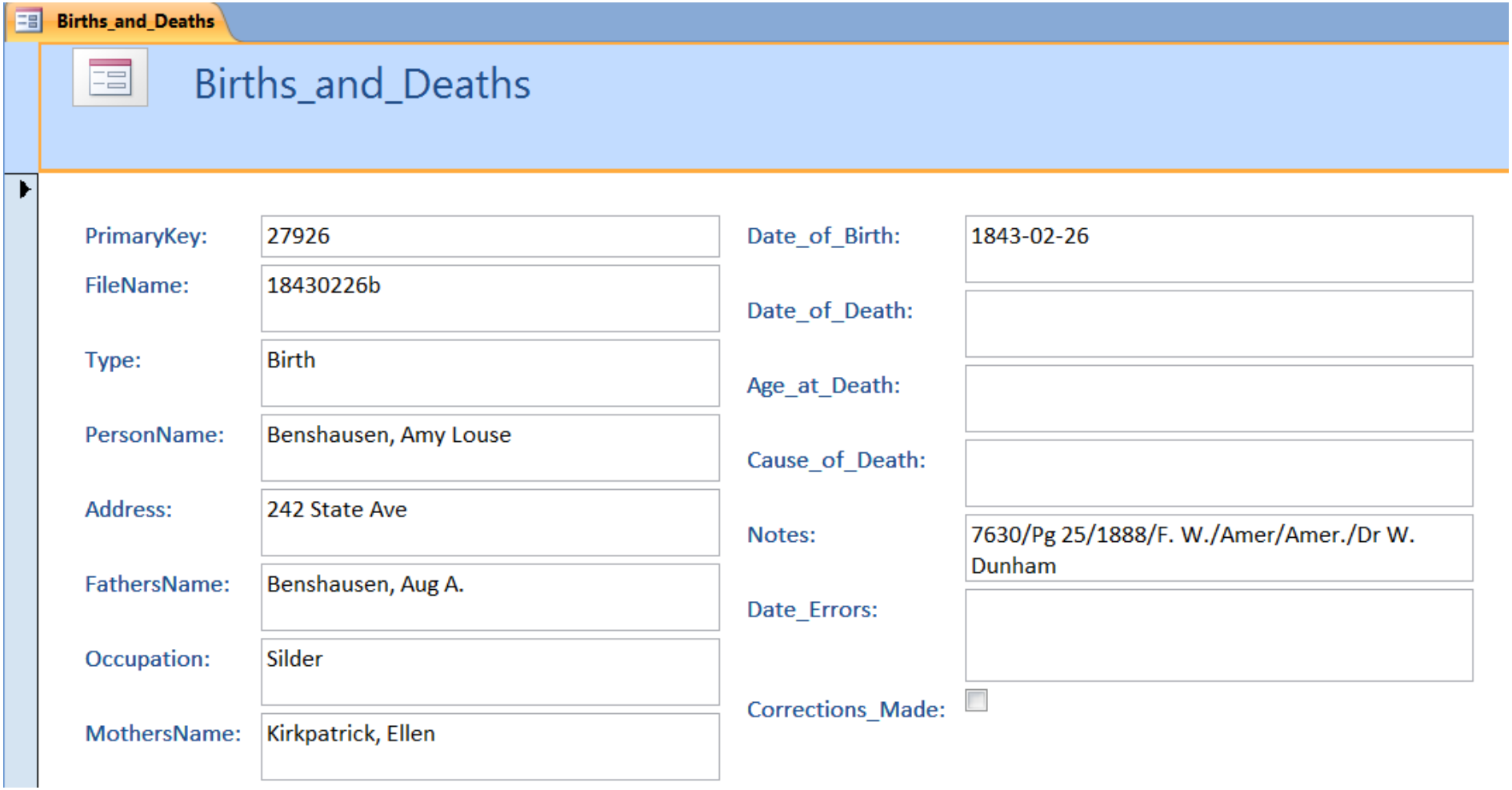
The data as entered by the digitization vendor



A typical birth record



The data as entered by the digitization vendor



SQL Update Query Example

Active Update Father for Occupation

```
UPDATE Active SET Active.subject_FathersName = [Active].[subject_FathersName]+" ("+[Active].[subject_Occupation]+")"
WHERE ((([Active].birth_or_death)="Birth") AND (([Active].subject_Occupation) Is Not Null));
```

VBA Scripting Example:

batch_importer-stugrescant-Z-cut.xlsm - Module1 (Code)

(General) | FindNoXML

```
If LenB(StrTemp) <> 0 Then 'file already exists -- delete it
   Kill oDirectory & "\" & StrTemp
End If


If LenB(SubDirStr) <> 0 Then
  SubDirArray = Split(SubDirStr, ";")
  For i1 = 0 To UBound(SubDirArray)
    Debug.Print SubDirArray(i1)
    foundFile = FindXML(oDirectory + "\" + SubDirArray(i1))
    If (foundFile = False) And (SubDirArray(i1) <> "") Then
      NoXMLTrue = True
      If BoolTemp = False Then
         StrTemp = Dir$(oDirectory & "\" & TextFileName, vbNormal)
         If LenB(StrTemp) = 0 Then 'file does not already exist
           myFile = oDirectory & "\" & TextFileName
           fnum = FreeFile()
           Open myFile For Output As fnum
         End If
         BoolTemp = True
      End If
      Print #fnum, SubDirArray(i1)
      Write #fnum,
```

dublin_core.xml:

dublin_core.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<dublin_core>
  <dcvalue element="subject" qualifier="none">Birkenbusch, Louis</dcvalue>
  <dcvalue element="title" qualifier="none">Birkenbusch, Louis (Death, 1891-08-27)</dcvalue>
  <dcvalue element="description" qualifier="none">Address: 217 Findlay St.</dcvalue>
  <dcvalue element="subject" qualifier="none">Occupation -- Brewer</dcvalue>
  <dcvalue element="date" qualifier="created">1891-08-27</dcvalue>
  <dcvalue element="coverage" qualifier="temporal">1891</dcvalue>
  <dcvalue element="coverage" qualifier="spatial">Cincinnati (Ohio)</dcvalue>
  <dcvalue element="date" qualifier="issued">1891-08-27</dcvalue>
  <dcvalue element="description" qualifier="none">Age at death: 38 yrs.</dcvalue>
  <dcvalue element="subject" qualifier="none">Cause of death -- Injuries Bursting of Beer Vat</dcvalue>
  <dcvalue element="description" qualifier="none">453/Pg 99/1891/MW M/Md./Dr. Theo. Bange/Schraffenberger/Walnut Hills</dcvalue>
  <dcvalue element="description" qualifier="notes">Original record filed in drawer labeled &amp;#039;BIRD-BLACKNER&amp;#039;.</dcvalue>
  <dcvalue element="type" qualifier="none">Image</dcvalue>
  <dcvalue element="publisher" qualifier="OLinstitution">University of Cincinnati</dcvalue>
  <dcvalue element="publisher" qualifier="OLrepository">University of Cincinnati. Archives and Rare Books Library</dcvalue>
  <dcvalue element="publisher" qualifier="digital">University of Cincinnati. University of Cincinnati Libraries</dcvalue>
  <dcvalue element="language" qualifier="iso">en_US</dcvalue>
  <dcvalue element="contributor" qualifier="author">Cincinnati (Ohio). Health Dept.</dcvalue>
  <dcvalue element="format" qualifier="medium">paper</dcvalue>
  <dcvalue element="format" qualifier="mimetype">image/jpeg</dcvalue>
  <dcvalue element="rights" qualifier="uri">http://drc.libraries.uc.edu/fairuse.html</dcvalue>
  <dcvalue element="relation" qualifier="ispartof">Cincinnati Birth and Death Records, 1865-1912</dcvalue>
  <dcvalue element="date" qualifier="digitized">2010-02-17</dcvalue>
  <dcvalue element="identifier" qualifier="other">34565 (File Order Number)</dcvalue>
 </dublin_core>
```

The entire excel macro written in VBA can be found at the Wiki for the OhioLINK DRC project. I also extended the macro to identify files as license files, archival masters or thumbnail images, when creating the contents manifest, if you can identify for the macro a consistent pattern in the file naming convention that you use. https://sites.google.com/a/ohiolink.edu/drmc/bulk-submission/bulk-submission----alternate-e

Initial results

- Each submission package was built initially with 5,000 records, and then gradually increased up to 83429 records in one batch load.
- Record loads starting taking significantly longer per record – an undesirable effect of an import program for 1.6.2 that 'prunes' the indexes after each record is loaded. "For any batch of size n, where n > 1, this is (n - 1) times more than is necessary," as uncovered by Tom De Mulder and others in 2010. See http://dspace.2283337.n4.nabble.com/DSJ-Created-DS-470-Batch-import-times-increase-drastically-as-repository-size-increases-patch-to-mitm-td3291379.html and http://tdm27.wordpress.com/2010/01/19/dspace-1-6-scalability-testing/.

More results

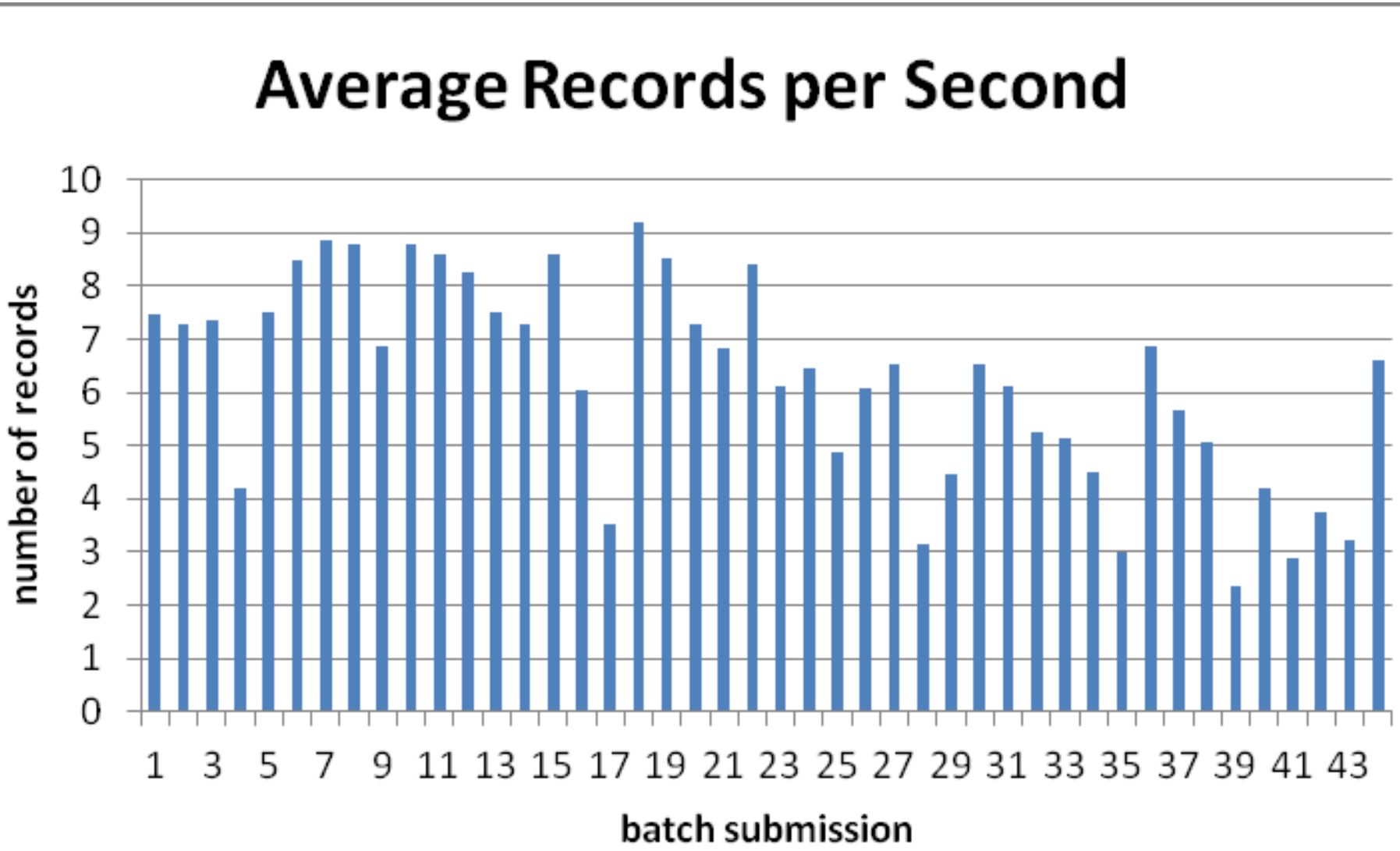
- After we crossed the 200,000 records threshold, the test system GUI began to perform very badly.
- And before we had determined a solution, the production system, now at 132,136 records for this collection, also started behaving very badly.
- Soon both systems failed to restart and to come online at all. This was the very result we had hoped to avoid.

More results…

- After several attempts to tweak PostGreSQL parameters and DSpace parameters, and to add memory, OhioLINK developers built a new DSpace 1.7.1 instance on new hardware on an Oracle, instead of PostGreSQL, back-end, and we were able to successfully load all 500,000+ records of this collection onto that platform.
- A rebuilt 1.6.2 test instance, still on PostGreSQL, contained 132,136 records from the previous production system, and a 3rd UC instance was also 1.6.2 and contained all other collections.
- OhioLINK developers concluded that both previous DSpace test and production systems (and test had been originally cloned from production) had suffered from the same file corruption that had intensified and compounded as our instances grew. They concluded that PostGreSQL was NOT the issue per se. However, we had better diagnostic tools with Oracle.

Denouement

- The export and re-import process used for collections previously built did not replicate the file corruption.
- The import program in DSpace 1.7.1 had been optimized to avoid the sequential indexing issue, and we were able to reload all records from the same submission packages originally uploaded to OhioLINK, in three days, getting us to an instance in DSpace 1.7.1 that was over 500,000 records at the end of April 2011. Record loads in DSpace 1.7.1 averaged a blinding 6 records per second.


Average Records per Second
number of records
0
1
2
3
4
5
6
7
8
9
10
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43
batch submission

Conclusion

- In sum, it took three months of struggling with the import process, and the system performance issues we were facing in DSpace 1.6.2 on PostGreSQL, to conclude that we absolutely had to migrate to the next version of DSpace and an Oracle back end, on new disk hardware -- once we did so, we finished the bulk of the load remarkably quickly.

Final Results

- We now had two production instances – one was over 500,000 records and running 1.7.1 on top of Oracle; one was much smaller and running 1.6.2 on PostGreSQL, and a test instance with over 137,000 records, also running 1.6.2 on PostGreSLQ. All three systems were performing well.
- But the separate production systems confused our users, and we did not have 'handles' or permanent record URI's, because we intended to merge these two DSpace platforms, creating one handle sequence.
- We began a merger project in May of 2012, and as of June 25, 2012, all collections are on one platform – now DSpace 1.8.2 on Oracle, with one handle sequence. We have also upgraded our test platform to 1.8.2 on Oracle.
- Other collections were exported and re-imported to the new platform, maintaining all handles. The Birth and Death records were re-loaded from the original submission packages, the second or third such load for all of these records, in a process that again took only a matter of days. The production University of Cincinnati DRC (http://drc.libraries.uc.edu) as of July 5, 2012, contains 526,825 records.

Sabin collection

- Our Albert B. Sabin collection has more recently been processed in a similar way. We outsourced the scanning processes and asked the vendor to create a spreadsheet with minimal metadata.
- We used SQL and VBA to transform this minimal metadata into a complete dublin_core.xml record.

Sample SQL Update Query

02 - Fix Initials Query-SubjectRecipient

```
UPDATE SourceTable SET SourceTable.SubjectRecipient = [SourceTable].[SubjectRecipient]+"."
WHERE (((StrComp(UCase(Right([SourceTable].SubjectRecipient,1)),Right([SourceTable].SubjectRecipient,1),0)=0)=True) And ((Right([SourceTable].SubjectRecipient,1))<>".") And
((IsNumeric(Right([SourceTable].SubjectRecipient,1)))=False));
```

Data from Digitization vendor

B2 'Correspondence, General

	B	C	D	E	F	G	H	I	J	K
1	Box Title	Folder Number	Folder Title	Directory Name	File Name	File Type	Scan Date	Author	Recipient	Date
2	Correspondence, (	2	B Virus -- 1955-58	bvirus_1955-58	bvirus_1955-58_0(	letter	2010/11/23	Spaar, F W	Sabin, Albert B. (A	1955/09/05

Resultant dublin_core.xml record

dublin_core.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<dublin_core>
  <dcvalue element="type" qualifier="none">letter</dcvalue>
  <dcvalue element="date" qualifier="digitized">2010-11-23</dcvalue>
  <dcvalue element="contributor" qualifier="author">Spaar, F. W.</dcvalue>
  <dcvalue element="subject" qualifier="none">Sabin, Albert B. (Albert Bruce), 1906-1993 -- Correspondence</dcvalue>
  <dcvalue element="date" qualifier="issued">1955-09-05</dcvalue>
  <dcvalue element="title" qualifier="none">B Virus -- 1955-58 -- Correspondence, General -- letter, 1955-09-05</dcvalue>
  <dcvalue element="type" qualifier="none">text</dcvalue>
  <dcvalue element="subject" qualifier="none">Spaar, F. W. -- Correspondence</dcvalue>
  <dcvalue element="date" qualifier="created">1955-09-05</dcvalue>
  <dcvalue element="coverage" qualifier="temporal">1955</dcvalue>
  <dcvalue element="description" qualifier="none">Letter from Spaar, F. W. to Sabin, Albert B. dated 1955-09-05.</dcvalue>
  <dcvalue element="description" qualifier="none">&lt;a href=&quot;http://digitalprojects.libraries.uc.edu/sabin/fairuse/&quot;&gt;Sabin Collection Fair Use Policy&
  <dcvalue element="language" qualifier="iso">en_US</dcvalue>
  <dcvalue element="relation" qualifier="ispartofseries">Sabin Archives. Correspondence, General. Box 01.  File 02 (B Virus -- 1955-58)</dcvalue>
  <dcvalue element="publisher" qualifier="digital">University of Cincinnati. University of Cincinnati Libraries</dcvalue>
  <dcvalue element="publisher" qualifier="OLrepository">University of Cincinnati. Hauck Center for the Albert B. Sabin Archives</dcvalue>
  <dcvalue element="publisher" qualifier="OLinstitution">University of Cincinnati</dcvalue>
  <dcvalue element="rights" qualifier="uri">http://digitalprojects.libraries.uc.edu/sabin/fairuse/</dcvalue>
  <dcvalue element="relation" qualifier="ispartof">The Albert B. Sabin Archives</dcvalue>
  <dcvalue element="format" qualifier="mimetype">application/pdf</dcvalue>
  <dcvalue element="format" qualifier="medium">paper</dcvalue>
 </dublin_core>
```

Human processing

- Approximately one-fourth of the Sabin Archive requires human review for content that could violate the privacy of medical information, and the collection also contains documents that once were classified by the military and require scrutiny to be sure they are no longer classified.
- The archivist reviews the letters and uses either Acrobat X or Photoshop to line through the sensitive information, rebuilds the PDF, and may redact metadata as well.
- The DSpace record is exported, and the revised PDF and xml file are uploaded to our digital projects Storage Access network, from where I rebuild the submission package, and re-import these records.

More processing

- The dspace_migrate shell script that is part of the DSpace install is used to manipulate the exported dublin_core.xml files. I found that I could run this script outside of the DSpace environment on our local (LAMP) server, so that the records didn't have to be moved until they were ready to be sent again to the OhioLINK server for re-loading.
- After re-building the submission packages, records are again loaded to test, and if everything checks out we notify OhioLINK staff that the second submission package can now be imported into our production system.

More processing still…

The dspace_migrate shell script (from 1.6.2) was deleting our handle fields, because of language codes that had been added to the field. I managed to extend the regular expression so that it did delete the handle field:

```
 | $SED -e 's/<dcvalue element=\"identifier\"
qualifier=\"uri\"
language=\".*\">http:\/\/hdl.*<\/dcvalue>//'  -e 's/[
^I]*$//' -e '/^$/ d'
```

Note that the second -e 's/ is to delete blanks or tabs remaining in the line, and the 3rd -e'/ deletes the blank line.

Inspiration for separating the /d into a separate regular expression found here:
http://www.grymoire.com/Unix/Sed.html#uh-30

Conclusions

- Although multi-stepped, these procedures work and allow us to build this archive with a level of review that is satisfying our university's legal counsel.
- DSpace has proven to be capable of serving large numbers of records with associated indices. Processing large record sets is achievable with a replicable set of procedures and methods, and the willingness to use scripting languages to manipulate large sets of data.
- A significant amount of disk storage is required to manipulate and stage large datasets. (Our current local Storage Area Network is was just almost doubled in size to 36 Terabytes, space we will use.)

My Job Title?

- At times I think my job title should be changed to ‘Chief Data Wrangler’, but for now Digital Projects Coordinator will still suffice.

Linda Newman
Digital Projects Coordinator
University of Cincinnati Libraries
Cincinnati, Ohio (United States) 45221-0033
Telephone: 523-556-1555
Email: Linda.Newman@uc.edu
http://digitalprojects.libraries.uc.edu
http://drc.libraries.uc.edu