

# **What Was Here? Prototyping a Map-centric Historic Document Database for King County Archives**

Tyler Graham and Tyler Buntain

University of Washington

MGIS 2017 Cohort

Geography 569 Capstone Project

## **Table of Contents**

<b>Recommended Course of Action</b>	<b>2</b>
<b>Introduction</b>	<b>4</b>
Social Ecological Systems	6
<b>Design and Methods</b>	<b>7</b>
Adding The PSAAP Toolbox	7
PSAAP Database Table Script	9
Import Data to Database Table Script	9
Processing Fields for Relationship Classes	11
Building Relationships	11
Web Map / Story Map	11
Adding Coded Values	11
<b>Results</b>	<b>13</b>
The Geodatabase	13
Tools	13
Archive Web Map Portal	13
<b>Discussion</b>	<b>14</b>
Goals vs Reality	14
Relationship Problems - Participating in Multiple Relationships at Once	14
Geocoding	14
Area Selection	14
<b>Business Case and Implementation Plan</b>	<b>14</b>
NEPA, 106	14
<b>Literature Cited</b>	<b>14</b>

## List of Figures

Figure 1: PSAAP Database Social Ecological Systems Table

Figure 2: Opening the Toolbox.

Figure 3: The PSAAP\_Tools successfully added to ArcToolbox.

Figure 4: Locating the Python scripts used by the PSAAP\_Tools

Figure 5: Navigate to the PSAAP Database in ArcCatalog.

Figure 6: The List of Domains (top table) and Coded Values of a selected domain (bottom table).

Figure 7: Scroll down to empty rows to add additional domain values and descriptions.

Figure 8: Enter the Code (left) and Description (right)

## Recommended Course of Action

There are several paths still available to take the Spatially aware PSAAP Database down depending on the sponsor's opinion of how the user interface meets their goals and idea of a user friendly interface.

Our first recommendation is that the database continues to be scoped and developed. We have a robust database underway, however it has not reached its final iteration to be considered a finished product. Additional database needs were still being identified during the last week of class and there was not enough time left in the quarter to begin to research or formulate a solution to them.

One question was about incorporating Getty Vocabulary into at least one of the database fields. We had a quick discussion with the sponsor about how this would be used in the database. Getty Vocabulary has millions of different standardized terms available for use (Getty Research Institute 2017). Getty Vocabulary Terms are available in a variety of formats; manual searches of the website, "XML, relational tables, APIs, and Linked OPEN Data" (Getty Research Institute 2017). This question is outside of our scope for this project, but one that does need an answer to if incorporating the Getty Vocabulary into the database is possible. If it is possible, it raises the second question of "How would it be done?"

The second question that still needs an answer in regards to database design, is how to incorporate historic documents that span multiple Township, Ranges, Sections, or Quarter Sections? The current solution was to choose an arbitrary number of iterations of the Township, Range, Section, Quarter Section fields (in this case, duplicating these fields 4 times). Each historic record that spanned more than one of these polygons would need to use the next iteration of the groups of fields. This would need to occur until all Township/Ranges were accounted for. If the database is implemented across King County, this would create a problem for long linear features, such as roads. The available number of Township/Range fields would simply not be enough. A potential solution to this problem was in the works, but problems with the Python script prevented its rollout as part of the database. The solution identified by our team was to append the Township/Range/Section/Quarter Section data to the Unique Record Identifier(URI) and then duplicate the record for each township/range/section/quarter section it crossed.

For example, if a record existed that spanned the Township/Range fields T1, R1, S1, QS1, T2, R2, and S2, the script would create 5 new records in the PSAAP table appending the URI for the new records in sequence with T1+R1, then T1+R1+S1, and then T1+R1+S1+QS1. These would be followed up with additional URIs for new records having T2+R2, then T2+R2+S2 appended to the name. This script would also need to write the later Township/Range fields (T2/R2 and up) to the first iteration of the Township/Range fields (T1/R1). The reason for this is because of a difference between how ArcMap handles relationship classes and how ArcGIS Online (AGOL) handles relationship classes. ArcMap lets you create multiple relationships between the same feature class and database table and you can find the records that relate for each version of the relationship. This allowed us to create four relationships between the PSAAP Database Table and the PLSS Township feature class using the PSAAP Database Table fields: Town\_Range1, Town\_Range2, Town\_Range3, and Town\_Range4. The problem we ran into was when we attempted to publish as a feature service to AGOL. It treated the PSAAP Database Table as non-existent when publishing. This resulted in no historic document metadata going up to AGOL, which rendered the database useless. We tracked the issue to the multiple relates, a feature class could

only have one relationship class assigned to it from one table. We eliminated the extra relationships, but this is a problem that needs a solution to handle records that cover large geographic areas.

After data was hosted to AGOL, problems were encountered with the ability to configure pop-ups to embed data from related table into the pop-ups. The source of this problem is the type of relationship class that was necessary to link multiple historic document attributes to one feature. It was necessary to use a 1 to M (One to Many) relationship. This lets multiple records in the PSAAP Database Table that share the same attribute value (such as multiple records attributed to one township/range) all point to and from the feature in a feature class. The authors' previous experience with AGOL has only been with 1 to 1 relationships, where only one record in a table relates to only 1 feature in a feature class. A 1 to 1 relationship can embed in the pop-up, but 1 to M relationships will only show the total number of records that relate to the feature and not actual records themselves. In other words with a 1 to 1 relationship the pop-up would show the Unique Record Identifier value in the Unique Record Identifier field since only one record is related. In a 1 to M relationship it show the total number or URI values that occur; so if there are 9 historic records that relate to the feature, the URI field displays '9' instead of each URI value. This opens up the need to revisit how the data is being hosted. It will be worth digging deeper into Open Source options, such as Carto and Mapbox to see how they work with 1 to M relationships.

We recommend exploring a custom web mapping application to develop a strong user interface that can leverage a related database. Examples of web mapping applications that allow users to search for records by location are USGS Earth Explorer ([earthexplorer.usgs.gov](http://earthexplorer.usgs.gov)) and USGS TopoView ([ngmdb.usgs.gov/topoview](http://ngmdb.usgs.gov/topoview)). TopoView uses Mapbox, an open source map hosting service and both websites use JavaScript to develop the interface, which is beyond the skillset of this Capstone group. We had hoped that an out of the box solution existed, but it appears that customization by web developers is going to be a necessity to produce a well-designed and functional product.

## Introduction

Our project sponsor, King County Archives, is part of the Puget Sound Area Archive Partnership (PSAAP). Their goal is to have the archives searchable by location on a user-friendly web map. Currently, PSAAP members maintain their own separate databases, which have different interfaces in how the records are searched. This creates a barrier that hinders the general public's ability to access historical records; people may not know about all the different archive websites, the different interfaces could be confusing, and text only searches may rely on information they do not have.

King County Archives are looking for a product that can link scanned historical documents to feature classes to perform a spatial records search. They were also hoping to have geocoding enabled for historic records. The desire was expressed to georeference all the map files and have them available as basemap layers in the web map.

As we started scoping the database, we interpreted this to mean that our database would be storing the images as well as the historic records metadata that would be used to create a spatially aware database. As database scoping progressed we learned that the hosted document scans will remain on the

individual archives servers, a centralized spatial database will point towards the scanned documents using a URL link.

Successful implementation of a centralized spatial database will have an impact beyond the general public’s ability to look up records about where grandpa used to live. Projects that receive federal funding or permitting are required to follow NEPA. This includes a review of historical records related to a project’s Area of Potential Effect (APE). Researchers will be able to visit one website which will have historic records related to their appropriate spatial locations. This will enable professional researchers to look at where their APE falls on a map and select the appropriate areas from the archive web map to find the documents that will help determine if there are potential cultural resources that need to be considered during the planning process.

Spatial and Temporal Scale	Environmental	Economic	Social
Above Focal Scale (+1) King County Future Implementation	Returns records covering large geographic areas, enabling compliance with NEPA Section 106 for historic records research	Online records contributed by multiple agencies will reduce the time travelling to archives, or learning different website interfaces. An intuitive website for a Partnership of Archives will save money for projects that must meet NEPA requirements by concentrating records into one website and making the database searchable by location rather than just keyword queries.	Large geographic area records could enable a broader understanding of local history that would now be accessible
Focal Scale (+0) City of Kenmore	Basemaps of historic maps in the focal scale can be used to help analyze areas for environmental permitting requirements.		Focusing on regional scale data allows for more public outreach for cities, unincorporated areas, neighborhoods, etc.
Below Focal Scale (-1) Quarter Section and Parcel features	If records indicate the potential for cultural resources (Historic Structures, Pre- and Post-Contact Archaeological sites, or any other resource that has the potential to meet national, regional, or local historic preservation standards) are discovered during the large area records search.	Rather than performing a cultural resources survey over an entire APE; targeted searches of records at the Quarter Section or Parcel level, allowing for intensive field surveys to be conducted in a more focused and efficient manner, resulting in a cost savings.	Records for individual parcels can be accessed by general public, requiring only the knowledge of where the parcel is on a map. They do not need to learn how to search archives by keyword queries. This will make archive records more accessible to the non-professional researcher.

Figure 1: PSAAP Database Social Ecological Systems Table

### *Social Ecological Systems*

No data analysis was performed as part of this database development, but that does not mean there were no Social, Economic, and Environment systems at work. Section 106 of the National Environmental Policy Act covers cultural resources when federal money or permitting is involved. Part of Section 106 is performing a records research for the project Area of Potential Effects (APE). The spans across all of the focal scales of the database; from large geographic areas above the focal scale to cover the APE for large projects. The focal scale of this map is at about the city scale (which is Kenmore, Wa for this project) enables APE document research for smaller projects. Then going to below the focal scale allows for targeted research in small areas at the quarter section, which will help design field survey methodology.

The environmental systems relates directly to economics; by centralizing archives into a spatially aware database this reduces the travel time researchers for trips to archives. This also streamlines the research process used online. Spatially aware records can be found simply by clicking on a map. The need to know keywords, neighborhoods, date ranges, parcels, townships, or other search keywords can now be shifted a role that serves to refine the search rather than the primary method of searching for records.

This ease with which records can be found spatially and in a centralized location will enable the general public to access historic records easier. Increased public awareness of historic records could lead to increased use and visitation of archive resources. There are positive and negative aspects of having easily accessible records. On the positive side, better access can lead to greater awareness and demand for more records, which could translate into additional funding to convert historic documents to digital format and add them to the PSAAP archive.

## **Design and Methods**

King County has a plethora of archival data. Thus, the scope of this project was reduced from all of King County down to just the Kenmore area. As the scope for this pilot project Kenmore proved to have enough data associated with it to carry out our models without associating so much data to make its processing impractical. Instead of maintaining the priority to georeference images en masse the scope was altered to focus on geodatabase creation. A well designed database will provide future archive contributors structure for additional data, and as a geodatabase it provides geospatial access to data instead of only keyword searches.

For the following Python scripts to function you need ArcGIS installed on the machine you will be using. The scripts use the arcpy Python module which comes as part of the ArcGIS software installation. The scripts also uses the Get Parameter As Text tool, which allows for user defined Parameters when the tool is run within ArcGIS.

### *Adding The PSAAP Toolbox*

1. Open ArcMap.

2. Select the Toolbox icon in the toolbar.

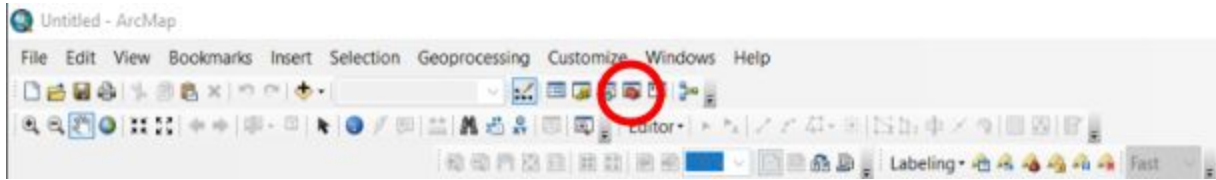


Figure 2: *Opening the Toolbox.*

3. The ArcToolbox should open up. Right click within the window and select 'Add Toolbox'.
4. Navigate to where you have stored the 'PSAADB' folder and open 'PSAAP\_Tools.tbx'.
5. The PSSAAP\_Tools toolbox should now be in the toolbox window.

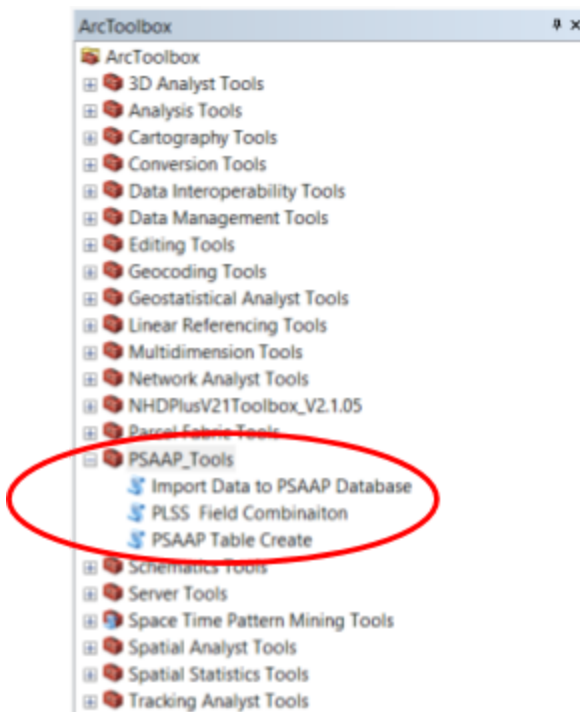


Figure 3: *The PSAAP\_Tools successfully added to ArcToolbox.*

6. Right-click on 'PSAAP Table Create' tool and select 'Properties'.



- In the 'Source' tab there is a browser bar. Either insert the file path for 'PSAAP\_db\_createV12' or navigate to it. You can find it in the 'Scripts' folder within 'PSAADB'.

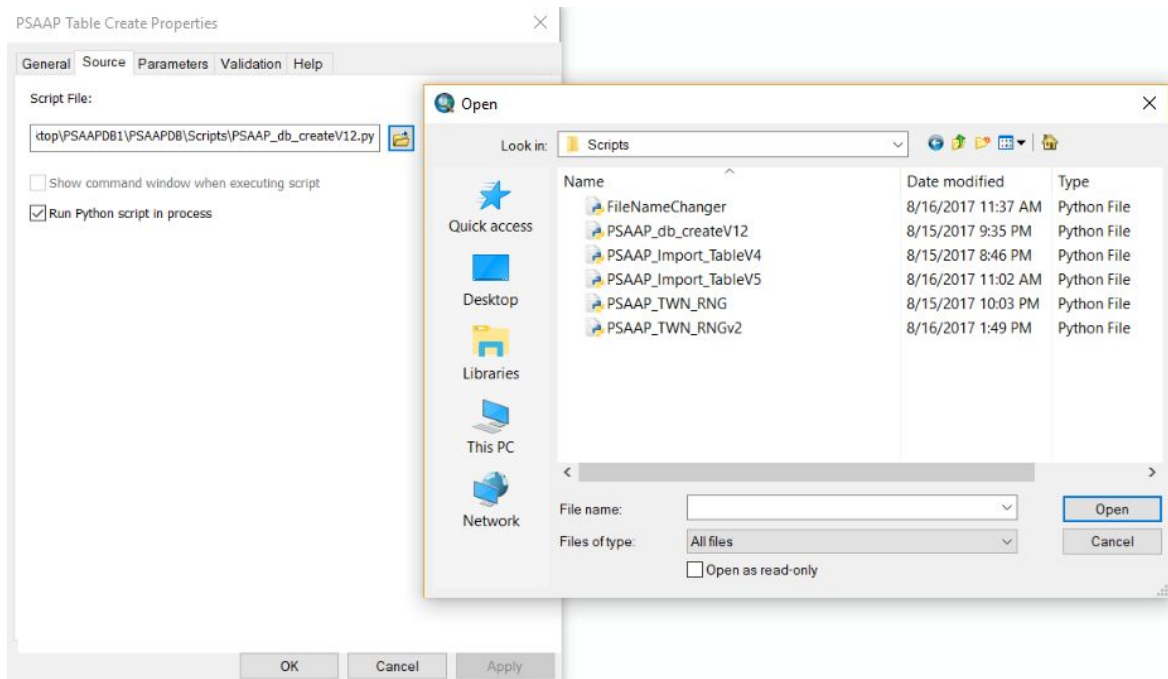


Figure 4: Locating the Python scripts used by the PSAAP\_Tools

- Save the ArcMap file as a template with the PSAAP tools waiting at the ready.
  - ➔ Repeat steps 5-8 for the tools 'Import Data to PSAAP Database' and 'PLSS Field Combination'

### PSAAP Database Table Script

**Note:** This tool requires an ArcMap Standard (ArcEditor) or Advanced (ArcInfo) license to function. It calls on ArcPy to execute database and table management tools.

A Python script was designed that creates a geodatabase table to store metadata spreadsheets from each archive partner. The initial setup imports arcpy from ArcMap, incorporates operation system rules, and pulls environment protocol from arcpy. There are two workspaces written into the script. Only one of these, 1.2.2., is meant for toolbox use within ArcMap. The other, 1.2.1., is for testing modifications to script versions outside of ArcMap. This method can be more efficient than choosing a file path as a parameter each time the script is run, but care must be taken to change it back and set the tool's script to the updated version.

The second section of the database creation script focuses on table creation, adding fields, and managing the parameters that define them. Variables are assigned to each component of the script. Because variables can represent file paths, lists, and other script components they are commonly implemented to reduce the amount of editing that accompanies modifications and can facilitate semantic uniformity.

In the interest of uniformity like fields were added together as a list whenever possible. On occasions where parameters such as field type or field length differ from like fields separate renditions of field management were executed. The fields are added to the table in an order consistent with the field format of archive metadata spreadsheets that predated this capstone project, but some reformatting was required to meet ArcMap standards.

In the third and final section of the database creation the script assigns domain variables and creates dictionaries for the finite values specific to some fields. The first version of the tool created subtypes for data form and genres. This resulted in each subtype overwriting its predecessor and the database only the final rendition existed by the time the script finished. We instead homogenized the script, opting to only create dictionaries for domains.

### *Import Data to Database Table Script*

**Note:** This tool requires an ArcMap Standard (ArcEditor) or Advanced (ArcInfo) license to function since the table will already have relationship classes created. There is no check for the license level in the script since the author's experience with government management of ArcGIS licenses uses a bank of licenses that are checked out as needed. The author's prior experience with integrating license checks with license banks has never produced a script that could accurately check the license level that was checked out. Therefore license checks were omitted to create a script that will run; it is up to the user to activate the appropriate ArcGIS license prior to running the script.

Once the database table script was developed, and run through the initial tests for functionality, a second python script was written to import record metadata from spreadsheets provided by King County Archives into the database. The early versions of this script had no capability to handle duplicate records. Each record from the Archives had a Unique Record Identifier number assigned to it. This was a combination of an identifier of the the PSAAP archive agency that provided the historic document, and an ascension number for the historic document.

Our vision of the workflow was that only new records were being output from the PSAAP archives, so duplicate handling was unnecessary. After consulting with King County Archives on this, we learned that the historic records metadata would need to be updated if more information about a record became available. These updated records would be included with new records as part of the spreadsheet to upload to the database. This meant that the script needed a way to not just insert new table attributes, but update existing ones at the same time.

This script function by the user setting the Workspace to the PSAAP geodatabase. They then set the spreadsheet to import into the geodatabase and then set the table in the database where the historic document metadata from the spreadsheet will be stored. The table must be a table that was created using the PSAAP Database Table script. Attempts to load it to different tables may result in errors due to number of fields, field types and attribute types (string, integer, decimal, or dates).

Once the script parameters are set in ArcGIS, the script runs a tool that imports the spreadsheet data into a temporary database table within the PSAAP Database. Two search cursors are then activated, one scans the temporary table and the other searches the PSAAP Database table for the Unique Record Identifier(URI) values in each table. The returned URI values are added to one of two lists; one list for

the temporary table URI values and another list for the PSAAP Database Table URI values. The lists are converted to sets and then intersected to find common values. The intersect URI values are duplicates and they are added to a new list. The scan cursors are closed.

As the tool continues to run it starts a new search cursor, which scans all the fields in the temporary table. If it finds a duplicate URI the script starts an update cursor and updates the existing record based on the data in the temporary table. If no duplicate URI is found the script starts an insert cursor and adds all the attributes to a new record in the PSAAP database table.

Three functions were also created in this script to process dates into a format that allows the update cursor and insert cursor to add them to the appropriate fields in a format compatible with the PSAAP Database table date fields. The Date Range field in the spreadsheet can hold both a start and end date for the time period a historic document belongs to. The first two functions separate the date range into a start date and an end date. The date range data must be formatted in one of two ways in the spreadsheet for these functions to work; Start Year - End Year without spaces between the hyphen (e.g. 1930-1945) or as a single year (e.g. 1933). In the event of a single year the function will use that year as both the start and end of the date range.

The third function processes the known date of the record. This function draws from the Original Record Date field in the spreadsheet, so dates in that field must follow a certain format to work. Dates in this field must be in m/d/yyyy format without leading zeros. This means that August 9, 1930 needs to be entered as 8/9/1930. If the date is entered as 08/9/1930 or 8/09/1930 it will return an error and stop the script. Also dates cannot be formatted with hyphens; a date format of 8-9-1930 will not work. Also it is important to make sure the Original Record Date column in the spreadsheet is formatted as text, otherwise it imports the date codes that Excel uses to tell the time; these are not usable values for the database.

Once the insert cursor and update cursor have finished adding or updating rows, the script deletes the temporary table from the database. The script is then complete and the PSAAP Database table will contain the new or updated records.

### *Processing Fields for Relationship Classes*

The third Python script takes the individual fields for the iterations of the Township, Range, Section, and Quarter Section fields and concatenates them into single strings that are used to relate to the King County PLSS (Public Land Survey System) feature classes for Township/Range, Section, and Quarter Section. For each iteration of the of the Township Range fields it populates 3 fields specifically created as part of the PSAAP Database Table tool to hold these values. Town\_Range fields holds the combined Township/Range fields as double type data. Town\_Range\_Sec fields hold the combined Township/Range/Section fields as double type data. Town\_Range\_Sec\_QSec fields hold the combined Township/Range/Section/Quarter Section fields as a string. This is because quarter sections are NE, NW, SE, SW and not number values. The fields are populated using an update cursor. These field types match the corresponding fields used to create the relationships in each of the PLSS feature classes.

## *Building Relationships*

Relationship classes were created in the database using ArcCatalog. The HistDocsData table of the PSAAP Database, which contains the metadata about the historic records, was related to the King County PLSS feature classes for Township/Range, Sections, and Quarter Sections. Initially there were four sets of relates to each PLSS feature class, one for each iteration of the arbitrary number of repeated PLSS fields that we had decided to use. This did not function in AGOL (more detail is provided in the Discussion subsection Relationship Problems) and we had to revert to using one relationship per feature class. Relationships were created using the following steps and if the database needs to be rebuilt, follow these steps to build a relationship:

1. Open the PSAAP Database in ArcCatalog,
2. Right click on the PLSS feature dataset.
3. Select **New** and from the submenu select **Relationship Class**.
4. In the new window that appears, name the relationship class. Make sure it has no spaces and is easily identifiable.
5. For the **Origin Table** select **HistDocsData**.
6. For the **Destination Table** expand the PLSS feature dataset and select **plss\_qtr**.
7. Click Next.
8. Set the relationship type to **Simple (peer to peer) relationship**.
9. Click Next.
10. In the next window leave the values at the default settings and click Next.
11. Set the cardinality for the relationship. Chose **1-M (one to many)**.
12. Click Next.
13. In the next window leave the setting as **No, I do not want to add attributes to this relationship class**.
14. Click Next.
15. In the next window you will set the Primary Key for the origin table. Since this is the Quarter Section feature class we relating to set this to the field **Town\_Range\_Sec\_Qsec1**. This tells the HistDocsData Table to use this field to create the relationship with the plss\_qtr feature class.
16. Set the Foreign Key field in the destination feature class. The field to select here is **TRSQ**.
17. Click Next.
18. The next window shows a summary of the relationship class. Hit **Finish** and the relationship will be established.
19. Repeat these steps to create new relationships with each of the plss\_sec and plss\_twn feature classes (Step 6). To create the relationship with the plss\_sec feature class, in the HistDocsData table select the field **Town\_Range\_Sec1** in Step 15 for the Primary Key field. For the Foreign Key field in Step 15 select the **TWNRNGSEC** field.

For the plss\_twn relationship, in Step 15, set the primary key to **Town\_Range1** and the foreign key to **TWNRNG**.

To create the relationship for the parcel\_address\_area feature class, in Step 6, expand the Parcel Feature dataset and select the **parcel\_address\_area** feature class. In Step 15, the Primary Key will be **Address2017** and the Foreign key will be **FULL\_ADDR**.

A Note on Address Relationships. The FULL\_ADDR field addresses are all capitalized. The same must be done to the Address2017 addresses for the relate to work.

### *Web Map / Story Map*

The web map that was produced used the PSAAP Database AGOL hosted feature service to add layers to the web map. The map also includes the 1936 King County Aerial imagery as a layer that can be enabled and disabled. The group was not able to get raster mosaics that we had created published as imagery services to include on the web map. The raster are the Pre-1940 Road Establishment map and the combined Kroll Land Use map pages from 1951 - 1959. These can be hosted by King County on their servers and then added to the web map in the future.

### *Adding Coded Values*

The ability to add more Coded Values to a domain is a workflow the project sponsor requested to have described to them. This will allow the sponsor to update Domains as more archives participate, more historic record formats are incorporated, or more townships and ranges are needed.

1. Open **ArcCatalog**.
2. Find the **PSAAP Database** (PSAAP\_Final.gdb in the example)

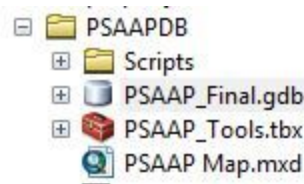
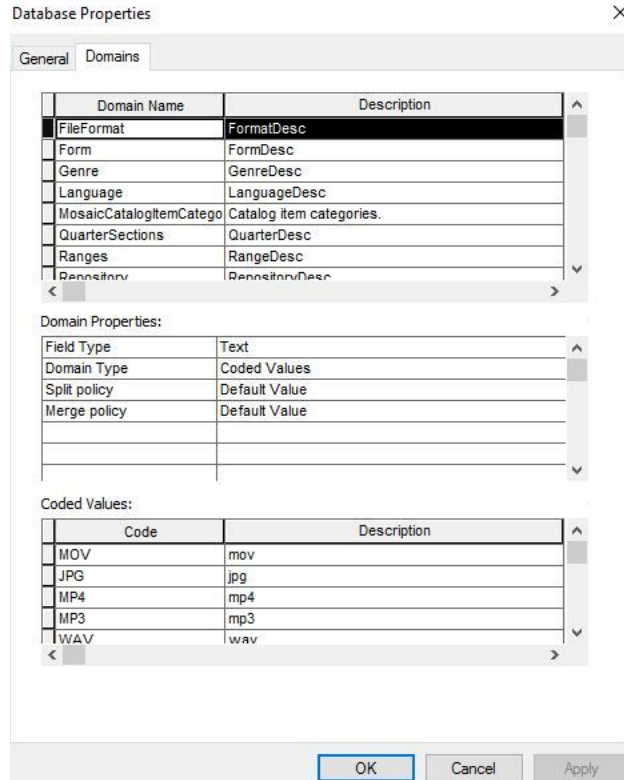


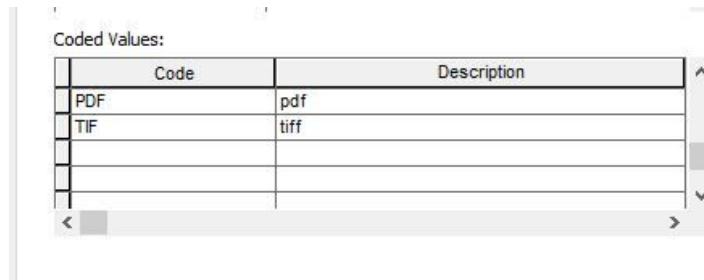
Figure 5: *Navigate to the PSAAP Database in ArcCatalog.*

3. Right click on it and select **Properties**.
4. Click on the **Domains** tab.
5. From the table at the top Select the Domain to edit by clicking on it until it highlights.



**Figure 6:** The List of Domains (top table) and Coded Values of a selected domain (bottom table).

- The **Coded Values** for the **Coded Value Domain**, that has been selected, displays on the bottom table. Scroll through the list until you find empty Code and Description fields.



**Figure 7:** Scroll down to empty rows to add additional domain values and descriptions.

- Type in the **Code**. This is the coded value that is stored in the Database Table. The length of the Code must fall within the length of the Field(s) it is assigned to. For Example if the FileFormat domain is assigned to a field that has a length of 3, you can enter a 3 character code, like GIF. You could not however, use a 4 character code like JPEG.
- In the example above, **GIF** is entered as the coded value and **gif** is entered as the description.

Coded Values:

Code	Description
PDF	pdf
TIF	tiff
GIF	gif

Figure 8: Enter the Code (left) and Description (right)

9. The description is not held to the field length limits that the code is, which means you could enter a more indepth description such as **Graphics Interchange Format**. While the coded value is what is stored in the database, the description is what displays when you open the attribute table and look at the field that uses the domain.
10. Click **OK** to make the changes to the domain.

## Results

### *The Geodatabase*

PSAAP now has a geodatabase tailored to their needs that has the potential serve as a powerful repository for historical data. Visitors to the King County Archives facility or webmap can access archives through a medium that combines geospatial capabilities with broadened search capabilities in comparison to the current search portal. The geodatabase contains documentation, PLSS parcel feature classes, the PSAAP tools, and relationship classes. Its 'HistDocsData' table contains a date field, but also fields for start and end dates that are short integers instead of date data types. The fields processed for relationship classes that contain combined town-range and town-range-section values are doubles, and the quarter sections are text. File size is the only attribute that is a long integer. All other attributes are text fields.

The geodatabase has four relationship classes that connect it the attribute tables of feature classes used to spatially query data. The parcel, quarter, section, and township feature classes all share relationship classes with 'HistDocsData'. Each of the feature classes offer increased resolution through a recursive relationship. Parcels generally smaller than quarter sections which are nested within sections that fill townships.

### *Tools*

Three tools using Python scripts were created to streamline database creation, data importing and relationship management. The first tool creates an empty table in a database that has the fields mapped out and ready to use to import data from spreadsheets. The second tool imports data from the spreadsheets to the database table. It is capable of finding duplicates based on the Unique Record Identifier field and the updating the existing record in the database rather than creating a duplicate. It then adds new data from the spreadsheet to the table. The last tool processes the historic document's plss data into single fields that are used to relate the data to the plss feature classes.

## *Archive Web Map Portal*

The web map published can be upscaled to serve as a portal to access and share archival data with the public

## **Discussion**

### *Goals vs Reality*

Going into this project our assumption was that we would be linking existing non-spatial database tables to feature classes via relates to create a 'new' database which would then be spatially enabled and able to be used in a web mapping application. We assumed that a database existed.

As we met with the sponsor started to realize that there was a difference in database dialects that we used. Our interpretation after some meetings was that there was no database management system (DBMS) being used by King County Archives and we approached our project with that in mind. It was not until the end of the project that we found out, by accident, that was not the case. They do use a DBMS to manage their records. This misunderstanding actually guided the bulk of our project and in the end resulted in a much more robust examination of how to utilize the data and develop the first steps of what can be developed into very powerful and useful product.

Our interpretation of the sponsor's wants was that all the documents showing any sort of landscape should be georeferenced. Some of these records were odd shapes, covered small areas which would be of limited use for a web map that covers a large area. Georeferencing is lower skill level GIS work, and wouldn't have been suitable for a Master's Capstone project. Plus, georeferencing takes a lot of time per record to complete. With the potential for hundreds of thousands of record attributes to be stored in this database, we determined that georeferencing was a not a viable solution. We informed our sponsor of this and explained what we wanted to do; create a relational database that would access records quickly on a map without the need to make each historic record spatially aware beforehand.

Carol Shenk, our main contact with the archives, had very limited GIS experience going in. It was clear to us from the start that we were the GIS experts, and that it was our job was to translate the needs and wants she had into a useable product. Her willingness to listen and be responsive helped us immensely in scoping and developing the database. We would not have been able to produce a database that is developed to the extent it is, without her expertise in historic records and feedback that she provided.

### *Relationship Problems - Participating in Multiple Relationships at Once*

A significant problem that was encountered was the difference between how an ArcMap Geodatabase and ArcGIS Online Hosted Feature Service handle relationship classes. Relationships allow for tables to be related to each other using a set of keys to match attribute rows. Database managers can edit attributes of a table that is related to a feature class without actually editing the feature class itself. Relationships also allow data to be connected where it does not make sense from an organizational standpoint to include data in a feature class that has purpose different from the data you want to include. An example here is the data we are working with; parcel data has a totally different function (managing land ownership) than historic records metadata. However we need to use the parcel data to access the historic records metadata in a spatial context. This is where the relationship comes into play.



No editing of parcel data needed to take place to accomplish this; only the Address fields needed to be related to then access the metadata in a map.

The historic records table had multiple relationships that needed to be established with it; one for each of the Township/Range, Section, and Quarter Section, and Parcel feature classes. The PLSS feature classes were originally intended to have four relationships for each feature class in order to allow one historic document to relate to multiple features. This functioned in ArcMap; documents that covered large areas, such as road profile records, would appear as a related record for each section they crossed.

When the database was then published to AGOL as a feature service, none of the relates functioned. The historic document metadata table appeared in the AGOL feature service, as did the PLSS feature classes, but the relationship was broken. Testing took place, and it was determined that the 4 relationship classes that were assigned from one table to one feature class was what caused the problem. The number of relationships was reduced to one relationship per feature class to the historic documents metadata table and then re-published. This time the relationship worked. Clicking on a feature in an AGOL web map would show that related records existed, and the related table could be opened at the bottom of the map and the documents could be accessed.

It was then realized more work would be necessary to adjust the database schema to allow one historic document to relate to multiple features in a feature service. One potential solution, that would need more development and input from the project sponsor, would be to append the data used in the relationship to the URI field, for each different relationship the record participated in, it would need to be duplicated in the database. No functioning test of this was ever achieved, and a real concern is that if thousands, or hundreds of thousands of records live in this database, it would cause some serious bloat in terms of server space the database would need.

A second relationship issue was then encountered directly in AGOL. The authors' previous experience with relationships in AGOL had been with a 1 to 1, one feature relates to only one attribute in a table. The relationship classes in the PSAAP database are 1 to many, which allows multiple table attributes to relate to one feature. Pop-ups were going to be an integral interface for users to view and access historic records. Unfortunately with AGOL feature services that use a 1 to many relationship, AGOL is unable to show the individual table attributes for each relate. This means that an out of the box solution for an easy to use interface was not feasible with AGOL. It also meant that using the database to directly create story maps about certain subjects or areas would be impossible for someone with limited GIS experience; they wouldn't be able to query the results by a subject and add them to a new map. They will instead need to process all the records offline and join them to a feature class prior to publishing.

However, the web map can still display related records, in the pop-up clicking 'Show Related Records' will open the table and filter the results based on the number of documents that are related. While this is far from the intuitive user interface that King County Archives and PSAAP wanted; it is a start. People familiar with databases will still be able to use the web map. The URL to records even shows as one of the first fields in the table, so while the public may find the other table fields useless, they can click the link and go directly to the historic document hosted on the archives' server.

## *Geocoding*

The project sponsor and project team had hopes of being able to add a historic geocoder as part of the database. This would have allowed a text search by address, or other geocodable data, such as the PLSS data. This was pushed lower on the priority list of deliverables once research showed the immense undertaking this could be, New York Public Library has a citizen science geocoder and georeferencer to process old maps. It is a mammoth undertaking well beyond the time available to this project.

As the project progressed and it was clear that developing a functioning database with associated tools would be the main focus, the geocoder idea was scrapped as part of this project. Developing a robust database would allow this project to serve as a useful stepping stone for further development of a web mapping application. While the group failed to create the polished and functional web map that the project sponsor desired; it did build the framework to allow others to pursue further development without starting entirely from scratch.

## **Business Case and Implementation Plan**

AGOL's limited relationship class capabilities affected the user interface of the online map data portal, but in ArcMap all of the relationships are fully functional for local access. A GIS portal integrated into a workstation on site at King County Archives could offer in depth use of digitized records. ArcGIS desktop licenses may be available at minimal cost through the county enterprise system.

Agencies and contractors are required to follow the National Environmental Policy Act (NEPA) when considering development. Section 106 of NEPA states that agency officials are responsible for and EIS and Record of Decision an Environmental Assessment, and a Finding of No Significant Impact to ensure that historic properties are protected (National Preservation Institute). The webmap archive portal and the geodatabase created to support it can streamline and enhance the workflow of those tasked with scoping and identifying areas of cultural significance.

## **Literature Cited**

Getty Research Institute. 2017. "Getty Vocabularies."  
<http://www.getty.edu/research/tools/vocabularies/>

National Preservation Institute. 2017. "NEPA and Section 106 of the National Historic Preservation Act."  
<https://www.npi.org/NEPA/sect106>