

PG Cert: Computing for Cultural Heritage  
Work-Based Project for Information Professionals  
Birkbeck, University of London

PIMMS:  
Developing a Pre-Ingest Metadata Management System at the British Library  
(Project Report)

Jessica Green

19 April 2020

## **Abstract**

In my role as Project Analyst at the British Library, I primarily assess and prepare selections of the BL's legacy digitised content for ingest into our future Digital Asset Management and Preservation System (DAMPS). Over a period of a few years, we aim to analyse at least 1 PB of digitised content (tiffs) stored across the BL network, as well on external media such as CDs and DVDs. Metadata about this digitised content is stored and gathered in a wide range of systems, including the BL catalogues and the 'analysis spreadsheets' I've been generating in my role. Our current methods for analysis are heavily dependent on Excel spreadsheets and manual copying and pasting of metadata out of these systems. I propose to develop a proof-of-concept for a relational database to aggregate select metadata from some of these systems and query across them for reporting and analysis. Outputs from this database will include a series of reports and spreadsheets for a range of colleagues around the BL as we work towards the shared goal of preparing digitised content for ingest and online publication. After the pilot, I would hope to begin the implementation phase to build the network model of this database with the support of senior management and BL Technology.

## **Introduction**

In August 2018, I began my role as Project Analyst in the British Library's Heritage Made Digital team (British Library, 2018). Our team of 12 staff is partially split into two further subteams: the 'live' workflow team who manage and streamline ongoing BL digitisation efforts and a smaller 'legacy' team tasked with analysing and preparing the library's vast amounts of 'legacy' digitised content for long-term preservation and, where possible, online publication. This includes ~1 PB of TIFF image files from past digitisation projects stored across the BL's network drives. As these images were produced and funded by different BL teams, partners, and projects over a 20+ year period, they are of variable quality and contain varying amounts of metadata stored across different systems. Before select digitised items can be ingested into the BL's new Digital Asset Management and Preservation System (DAMPS), there is a tremendous amount of analysis and preparation work that needs to be done on the files and their associated metadata. One of the most important tasks includes identifying duplicates and ensuring that the identified 'master' image set is suitable for DAMPS ingest. This includes checking that the file sets are complete and fit current BL file-naming, file-format, and digitisation standards. It is also necessary for us to liaise with curators and other staff throughout the process to ensure that we have compiled the accurate, up-to-date, and complete metadata needed for ingest and pre-ingest copyright assessment for each of these items.

We are not the first individuals hired to work on this analysis of BL 'legacy' digitised content and have attempted throughout our tenure to build on and learn from this earlier work. Previous members of the Digital Preservation and Digital Scholarship teams began analysing this material years before the Heritage Made Digital team was formed in 2018 – the first team at the BL centrally managing and streamlining a portfolio of digitisation and publication efforts across the institution. One of the major outputs of this past analysis work included the development of the Digital Asset Register (DAR) – a Microsoft

Access database for recording information about BL digitisation projects. This was paramount in associating a unique project number and name to every digitisation project across the BL, including for 'business as usual' digitisation efforts.

While this database has been useful for storing 'project-level' information, including for project funder, digitisation provider, and the locations of file-sets themselves, it does not include the more detailed information of each digital item needed for ingest. It was initially designed for recording of information at the start of a digitisation project, meaning that the database reflects more of the ambition for each digitisation project, and not the reality of the output. For instance, the project manager might have selected 100 items for digitisation, but in the end was only able to digitise 80 items due to unforeseen copyright or conservation concerns. In most cases, the DAR was not updated again at the end of a project, meaning that the record for that digitisation project will not be up-to-date. The original intention of the DAR remains an important one, but its limitations and the scale of necessary pre-analysis work has led to an in-house desire for a more detailed, item-level 'DAR' for 'legacy' digitised content. The urgency of this need was heightened when the current DAR could no longer be supported by the BL's technology team and plans were made to migrate it into SharePoint.

Over the past year and a half, I have been focusing on analysing 'legacy' digitised content from two major BL collection areas: Asia and African Collections (AAC) and the Ancient, Medieval and Early Modern Manuscripts (AMEMM) section of the Western Heritage Collections. I have been using mainly Excel spreadsheets to record and analyse a range of metadata about these digital files, the projects/departments they were digitised as part of, and the physical items they represent. I have been manually checking a range of data sources and systems, as well as liaising with BL staff to compile the metadata needed for our analysis and eventual ingest into DAMPS. I have also been working with our team's part-time Research Software Engineer to write scripts to generate additional metadata about these items, including periodical network folder audits and technical metadata about individual files in these folders. One of the major outputs of this effort was creating 'master analysis spreadsheets' for these two collection areas – each row of these spreadsheets represents one BL network folder containing a complete TIFF file set (or one fully digitised item). Each of the columns in the spreadsheet represents the fields needed to generate 'ingest spreadsheets' for DAMPS as well as some additional fields useful during our ongoing analysis work.

The aim of my project is to develop a proof-of-concept for a SQL database that will support the management, analysis, and preparation of metadata about the BL's 'legacy' digitised content in preparation for a selection to be ingested into DAMPS. To make this project more manageable, I am focusing on the two main curatorial areas for which I've already done a considerable amount of manual analysis work: AAC and AMEMM. While the scope of the project will not include analysis of all ~1 PB of digitised content on the BL networks, I am liaising with other teams to help ensure that the schema I develop is fit-for-purpose across a wide range of digitised content, regardless of the curatorial area or project for which it was digitised.

There are a number of technical and institutional challenges for both realising this proof-of-concept and for future possible implementation. One of the key challenges is securing institutional buy-in for this software solution and for the technology support needed to host it on the BL network and interact with other 'live' BL systems. There are ever-increasing demands on the time of the BL Technology team, as more parts of the library strive to develop digital

services. I have taken into consideration the finite resources available and the need to focus primarily, at present, on key priorities agreed at a corporate level. The fact that the Heritage Made Digital team itself only has funding through March 2023 makes this more challenging; any software solution of this scale would need some ongoing technological support and maintenance. I have taken these limitations under consideration throughout the development of this process, trying to develop it in such a way that is lightweight and in line with existing BL digital services and technology solutions.

Another major challenge for analysing this digitised content is that it is not static; the files themselves, as well as their associated metadata, are subject to periodic change, deletion, and creation. Files are stored on the BL network in folders where numerous staff have read/write access and can move or edit the files at anytime (intentionally and unintentionally). This has made it more difficult to make our analysis spreadsheets reflect the current location and metadata for the digitised items we are meant to be analysing. In addition to existing files being edited or moved, more files and folders are being created as new digitisation projects are undertaken; many of these digital files are being stored in the same network directories as these 'legacy' items. The 'workflow' part of our Heritage Made Digital team record metadata about each of these 'live' digitisation projects in a bespoke SharePoint installation, tracking each of the items as they move through the end-to-end digitisation workflow and working with other teams to edit the records as they go. Some of the 'legacy' items may become 'live' items when the curator or another member of staff decides to publish the individual item online or include it as part of a 'live' digitisation project. As a result, metadata about this item may end up in both my 'legacy' analysis spreadsheets and the 'live' SharePoint system; this 'live/legacy' issue has continued to grow in scale and is one of the challenges this software solution is meant to help resolve.

My proposed software solution is to design a model SQLite database and write SQL queries using sample metadata I've gathered about digitised AAC and AMEMM collections. I used Python scripts to import existing metadata about this digitised content from a range of data sources and systems, including BL catalogues and analysis spreadsheets. Meetings with a range of colleagues were necessary throughout the process to better understand BL systems, how they interacted with each other, and which data sources were likely to be most reliable and complete. I periodically tested the validity and usefulness of this database by running sample SQL queries and adjusted the schema when necessary.

I found it both helpful and rewarding to share my schema and sample SQL outputs with internal and external stakeholders for feedback throughout the project. As a result, I have a better understanding of the organisational impact of this project and how it could be helpful to a range of staff working towards the same goals: preserving the BL's digital collections and making them as digitally accessible as possible – both to staff and when possible, the general public. At any given time, there are multiple many teams and individuals using different tools to answer the same questions about our 'legacy' digitised content:

- How much storage do we need? How much will we need in one year?
- Has this specific book/manuscript/object been digitised? Where is it?
- What have we already published online and where? Are there gaps?

- How are people engaging with our digitised content once it's online?

If this database were to be implemented and given adequate resources for ongoing support, I believe it will save BL staff time in their daily work and produce more accurate reporting and analysis about our digitised collections for their long-term preservation and where possible, publication online.

### **Literature review**

Another approach to the task of preparing digitised content for ingest would be for the British Library to invest in an out-of-the-box software solution. When faced with the challenge (and opportunity) of promoting its image collection online, the Institution of Civil Engineers decided to develop a bespoke Microsoft SQL database rather than an existing commercial product. Like many organisations dependent on external suppliers for software, they had grown 'wary of relying on external support' and wanted to take advantage of staff skills and knowledge. They predicted that this would also garner more institutional buy-in and allow for more seamless branding with existing services. This SQL database became instrumental in preparing and assessing digitised content for online publication. Several thousand of their institution's ~10,000 images had already been scanned, with varying amounts of image metadata and a high level of duplication. Although the database helped them remove duplicates and identify missing metadata before ingest, they found it might have been better in hindsight to rescan the images rather than spending the time identifying this missing information. This experience of searching for images on this database and those of partner institutions, also proved the helpfulness of rich metadata in searching and retrieving digitised images (Morgan, 2013).

One of the major challenges of conducting assessment of digitised content is that the metadata necessary to do so is spread across multiple systems, with varying degrees of accuracy and completeness. In their 2014 paper '*A preliminary evaluation of HathiTrust metadata: assessing the sufficiency of legacy records*' the authors discuss how they approached a similar metadata challenge. Their goal was to consolidate and categorise metadata from multiple MARC records to get a more comprehensive picture of the collections they described. This was achieved by converting their MARC records in MARCXML and then into a bespoke MODS database schema using XSLT stylesheets and SQL insert statements. I have personally experienced that the most difficult and time-consuming aspect of analysing digitised content is manually searching for and combining metadata from different systems and data sources, including the multiple BL catalogues. Other colleagues have expressed to me their frustration with spending a much of their time doing 'data clean-up' or manually copying/pasting data from different sources into new, often complex spreadsheets as part of their daily work. A major goal of this project is to develop a software solution to help staff save time on repeated manual tasks, to spend more time using their skills and domain knowledge.

Another key challenge for this project was to identify the most important data sources and systems that contained metadata necessary for conducting our pre-ingest analysis. Throughout the process, I met with a range BL colleagues to better understand the history and use of these systems, how they interact with each other, and how to export the necessary metadata. Even more important than the underlying technology, was the understanding of the

human element behind them: how they were being used by staff currently and how they've been used in the past. By speaking with colleagues who use and add to these systems, as well as those who manage them, I gained a deeper understanding of the metadata within them and why it might not always be 'perfect'. This was even more obvious in the cases where these data sources were spreadsheets and more easily susceptible to inconsistencies over time. These conversations also helped me understand better how metadata about the same item might be redundant or conflicting across these systems, and in these cases, where the most authoritative metadata was likely to be.

In addition to identifying the different systems used to hold metadata, it was important to understand the different types of metadata (descriptive, technical, administrative) as well as the various types of data they describe (physical items, digitisation projects, digital files, folders of files, external media). I consulted several sources about these specific types of metadata both at the British Library, as well as how they are defined and implemented by the wider archival community. One of the primary sources consulted was the BL's 2019 Collection Metadata Strategy (Neil, 2019), supplemented with several discussions with the Metadata team and users of these systems across the Library, including curators.

One of the primary sources of 'descriptive' metadata at the British Library are the primary catalogues – 'Aleph' for printed materials and 'IAMS' for archives and manuscripts. These catalogues describe the physical items themselves, including title, publication date, and language. Printed materials such as books and maps are catalogued using the MARC standard used widely by libraries, especially in the US (Library of Congress, 2008). These records are catalogued and searched by BL staff using the Aleph integrated library system (Ex Libris, 2020) and accessed by users online using the Primo user interface available on the BL's primary 'Explore' catalogue (British Library, 2020). For describing archives and manuscripts, the BL uses the ISAD(G) cataloguing guidelines common to archives, especially in Europe (International Council on Archives, 2000). BL Staff primarily use a bespoke backend cataloguing system called IAMS to edit and search the database, while users can access it online via the BL's 'Explore Archives and Manuscripts' catalogue, also on the Primo user interface (Ex Libris Knowledge Center, 2020). While it is possible to manually search and retrieve records for specific items using both the back-end and front-end interfaces, I submitted 'Collections Metadata Requests' from the BL's Metadata team to gather select data from these catalogues in bulk. These requested returned Excel spreadsheets with all of the metadata held in the system for a specified range of items. Although this was out of scope for the project, I can later investigate using the catalogues' APIs to pull metadata directly from these systems.

The second type of metadata necessary for analysis and ingest preparation of 'legacy' digitised content is called 'preservation' or 'technical' metadata. This includes detailed information about the individual files produced as part of the digitisation process – aka the 'digital surrogates' for the physical items described in the catalogues. Examples of this include file size, file type, and checksums – a string of numbers and letters unique to every digital file. This type of metadata does not already exist for most of the digitised material I was analysing, so I trialled several tools that would allow me to generate this information over large amounts of data. The most used tool for generating this type of metadata in the archival community is called DROID and has both a command line and GUI (National Archives, 2017). In the end, I found that another tool, Siegfried, was more robust and well-suited for

analysis digitised content held different storage locations and worked with our Research Software Engineer, Harry Moss, to write scripts for Siegfried to run over ~70TB of digitised content spread across 15 network folders using 20 office PCs over two weeks. The combined output of this Siegfried report was a csv file of over a million rows, which we were unable to open using Excel for analysis. I liaised with Harry to use Python Pandas to run analysis over this data; he has saved his scripts and sample outputs onto his GitHub page for later reuse (Moss, 2020). I also reviewed past BL initiatives to gather and analyse technical metadata about our collections, most notably by the Preservation Team. This included the PICASA project – an action plan for the ‘Preservation of Ingested Collections: Assessments, Sampling & Action’ and for validation of digitised content before ingest into the BL’s existing Digital Library System (DLS) (Pennock, 2016).

The third type of metadata generated and analysed as part of this work is called ‘administrative’ metadata. This includes important contextual history including which projects items were digitised as part of, who digitised the items and who funded this digitisation. This information is necessary to include with the digital items when they are ingested into the current DLS or future DAMPS system. Most of this information is held in spreadsheets or word documents or still in the heads of those who worked on these projects themselves. As I have been capturing some of this information in my analysis spreadsheets, I better understand where this data is already held, including who in the Library is most likely to know this information for each project. I have also come to learn that other colleagues are currently working on similar analysis for their own collection areas, as they also want to have an understanding of what they’ve digitised and published online (and what should be prioritised going forward). I worked directly with a colleague in the Ancient, Medieval, and Early Modern Manuscripts team who has shared with me five different spreadsheets used to gather and collate this type of information, as well as descriptive metadata about these items from the catalogues. As part of this project, I have been liaising with him to understand how best to combine these outputs into something more useful and streamlined going forward – as well as how we can best share this data between ourselves throughout the process.

A large part of this project has been recognising the previous and ongoing efforts from colleagues across the Library in gathering and generating the metadata described above; I am proposing a solution that aims to directly build on their efforts. This includes the work already done by our ‘live’ subteam to generate SIPS (Submission Information Packages) for ‘legacy’ digitised content. These SIPS are used by heritage organisations around the world to store metadata necessary for ingesting and retrieving images in their digital repositories (International Association of Sound and Audiovisual Archives, 2020). Before we can ingest any digitised items into the DLS or DAMPS, we need to create pSIPS (pre-Submission Information Packages) as Excel files that include a range of descriptive, technical and administrative metadata. Rather than building a database to be another ‘metadata repository’, I propose developing a database to import and analyse metadata held in existing systems or spreadsheets. This solution would allow staff around the Library to continue to be the ‘owners’ of this metadata and systems while we do our necessary analysis on the most up-to-date information about these digitised items. The real benefit of storing this data temporarily in a SQL database is that it would allow us to analyse and combine this metadata much easier, as well as export a range of reports in the form of Excel spreadsheets to share them with staff across the Library.

## Proposed solution

Although I plan to continue working on this software solution after the completion of the Birkbeck course, I have outlined the steps I followed to for the scope of this project (see Appendix – Project Plan). These steps included exporting data sources, mapping metadata from data sources to target tables, developing a schema for the database, importing data sources as source tables into the database, copying data from source tables to target tables, querying data from both source and target tables, and generating reports based on these queries. I tested and employed a range of tools to do these things, including Notepad++, Jupyter Notebooks, SQLite, Github for storing and sharing files, and creately and Lucidchart for drafting my models and schema.

The first step involved better understanding our current processes for analysing ‘legacy’ digitised content and comparing our ‘as is’ situation with my proposed software solution. I used creately software to draw two different models included in the Appendix of this paper: one for our current processes and one for my proposed software solution. These proved to be especially useful for discussions with colleagues who were able to pinpoint parts of the model that were specific to their work and make connections to other elements. These models went through multiple iterations based on this feedback and I also hung them in our office to solicitate interest and feedback from colleagues who happened to walk by them. On the left are some of the data sources and systems that hold metadata about digitised content and on the right are reports created by analysing and combining this metadata. In the middle of both models are the processes that we employ, or I propose we employ, to connect these data sources to their desired outputs.

Our current processes are heavily dependent on Excel spreadsheets and manually copying/pasting data from multiple sources (see Appendix – Model of Current Analysis Processes). While it has been possible to do meaningful analysis and generate helpful reports on digitised content using these methods, this process has also been time-consuming and not well suited to collaborative working. Copying/pasting from systems and spreadsheets into our ‘analysis spreadsheets’ is not only time-consuming, but this type of manual work is prone to error and cannot be kept up-to-date as these data sources are ‘managed’ or updated by other staff or teams. This proposed solution accepts the primary location of this metadata is often in externally managed systems and likely to change at any time. Rather than manually copying this data into spreadsheets that are bound to become obsolete, this project model proposes an automated system for importing data from external systems into structured database tables that can be updated systematically.

The ‘as is’ model also makes clear where it has proved difficult or impossible to analyse metadata from different data sources, even with manual intervention. One difficulty in connected metadata held in these different sources is that they sometimes describe different aspects and levels of digitised content – these include metadata about digitisation projects, physical items like books and manuscripts, the digital file created through digitisation, the folder of files on the BL network, or external media and harddrives. It has proven sometimes difficult to compare data across these systems and files because of these differences and something that a relational database could help solve.



One example of this challenge relates to the Siegfried report described in the Literature Review, which included detailed technical metadata about more than a million digital files – each file represents an image of a digitised object, for instance a page from a book, one side of an item, or a painting. The csv file containing this metadata so large it was impossible to even open on our office computers, let alone try to filter or analysis the data held within it. An additional difficulty in analysing this metadata was the fact that it was recorded at a different level than in our ‘analysis spreadsheets’. In our analysis spreadsheet, each row represents a folder of digitised content, and in the Siegfried csv file, each row represented an individual file in each of these folders. It proved difficult to combine this metadata, although combining them would have been very helpful for our analysis work. In addition to working with our RSE to write scripts for analysing the metadata in this file as described above, he also wrote a script to combine the size of tiff files and calculate the number of tiff files in each folder to help make this link.

One of the benefits of this database solution is to make links like this between metadata held in various systems more attainable and repeatable. The proposed PIMMS model shows a more interconnected and sustainable solution; at the heart of it is a SQL database called the Pre-Ingest Metadata Management System (PIMMS) (see Appendix – Model of PIMMS software solution). As the ‘current’ model shows, there are already several data sources being analysed by the HMD team to produce reports for a range of BL colleagues. One of the primary goals of this project was to test more efficient models of connecting and analysing the data from these same sources, and generating similar reports, but in more substantive, repeatable, and accurate manner. Since the number of systems and amount of data held in them are vast, it was important to develop a flexible model that could also work at scale. This meant that I had to constantly bounce back and forth between looking at the ‘big picture’ and zooming into the different elements to make sure the solutions I was putting in place worked for a variety of metadata and existing systems.

While developing these process models, I began exporting and gathering source data from a range of systems and teams across the Library. For each of the data sources described above, I had to liaise with the system owners and users to get the most up-to-date versions and in some cases, export the data itself. This included making ‘collections metadata requests’ to the BL Metadata team, liaising with colleagues in Technology to export data from the Digital Asset Register, and learning how to do export select data from our HMD team’s live SharePoint site. For simplicity, I decided to only import .xlsx and .csv files and to keep the files as close to the original source as possible. I had to make a few conversions, including copying txt files into csv files and dividing Excel workbooks with multiple sheets, but could later investigate more robust solutions to minimise this manual file transformation before import.

Once I had gathered sample excel and csv files to import into my database, I created a Google sheet to track the mapping of my data sources to my target tables. This process allowed me to easily view all metadata fields from each of my data sources in one place. I copied the full list of columns from each data source by transposing them into a single row and mapped them to the relevant target tables. I also attempted to normalise the columns names in the target tables and identified when the same metadata was being gathered in different data sources. This Google Sheet became the basis for my data

dictionary table and I have also used it to retain a complete inventory of columns from data sources in case we want to introduce more fields or sources to the database in future.

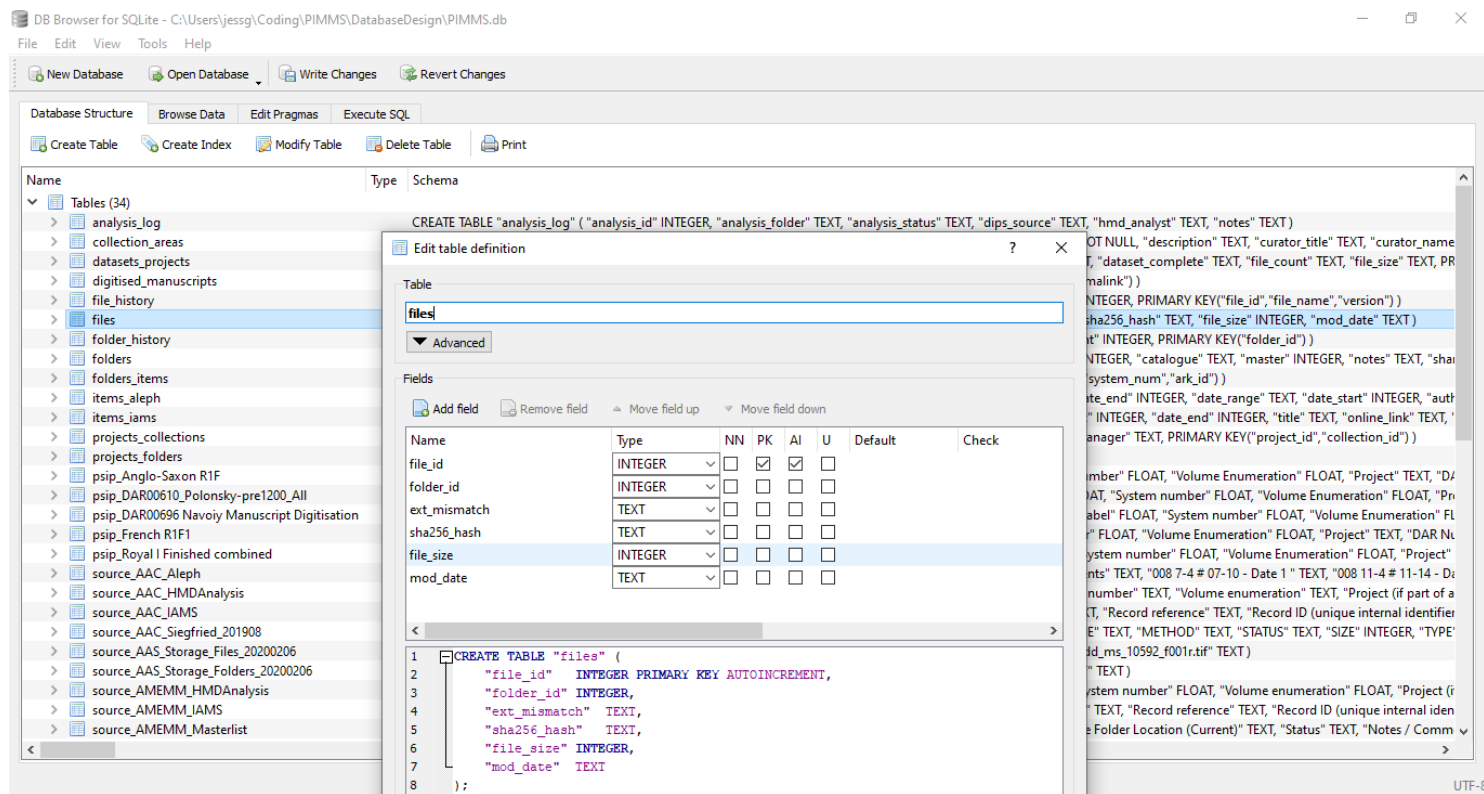
Data source	Source column	Target column	Transformation	Target table
Siegfried Report	FILE_PATH	file_name	Truncate to file name?	file_history
Siegfried Report	NAME	file_name	Use instead of FILE_PATH?	file_history
Siegfried Report	EXTENSION_MISMATCH	ext_mismatch	Change case: FALSE to False	files
Siegfried Report	SIZE	file_size	No	files
Siegfried Report	LAST_MODIFIED	mod_date	No	files
Siegfried Report	SHA256-HASH	sha256_hash	No	files

One of the most important and central aspects of my project was the development of the Entity Relationship Diagram (ERD) (see Appendix - PIMMS Entity Relationship Diagram (database schema)). This was also the visualisation that ended up soliciting the most productive and interesting feedback from colleagues. I tested different tools to design my schema, including pen/paper and creately (the same software I used for creating my process models), before finally settling on LucidChart. My schema went through at least 40 iterations; one of the later additions to the schema was organising it into 4 color-coded categories based on the types of analysis questions being asked:





1. Digital items (e.g. Where is digitised content stored on the BL networks? How much is there?)
2. Physical/Intellectual items (e.g. Which BL collection items have been digitised? Which have not?)
3. Administrative history (e.g. What digitisation projects have been undertaken at the BL? Who managed them?)
4. Publication and use (Which published items have people engaged with most on Twitter? Which ones not?)

Once I had a draft of both my schema and metadata mapping, I turned to development of the database itself. The decision on which SQL server to use turned out to be more difficult than I'd imagined. I investigated which SQL environments were already supported at the British Library, including PostgreSQL and Microsoft SQL server. I had also decided early in the project to primarily work on my personal laptop so as not to be dependent on BL technology for this early stage of development and for the freedom to work from home outside of working hours. This decision ended up being crucial with the unexpected transition to working from home, and on my personal laptop, during the Covid-19 outbreak.

I first tested PostgreSQL as I had wanted to build something that could be supported and integrated most easily into existing BL technology services. It was also free to install, which is the factor that made me rule out the use of Microsoft SQL server at this stage. After watching tutorials and trialing manually building my database tables in PGAdmin for PostgreSQL, I found the software itself a bit too complex and weighty for this early prototyping phase, especially considering the steep learning curve. I ultimately decided on using SQLite because it is free, lightweight and easy to start using with limited SQL experience. For possible future implementation at the BL, this PIMMS database could be mounted onto PostgreSQL or Microsoft SQL server quite easily. To start creating my actual database, I began manually creating my tables using DB Browser for SQLite as GUI editor. As I was still drafting my schema and column/table names, I wanted a solution that would make it easy to add, delete and modify tables while they remained in flux. I found DB Browser easier to use for this than PGAdmin as there were not as many features or ‘clicks’ necessary to perform these basic functions manually. In addition, it was much easier to export the database file as a .db file in SQLite and then simply email it to colleagues and my Birkbeck advisor for feedback.

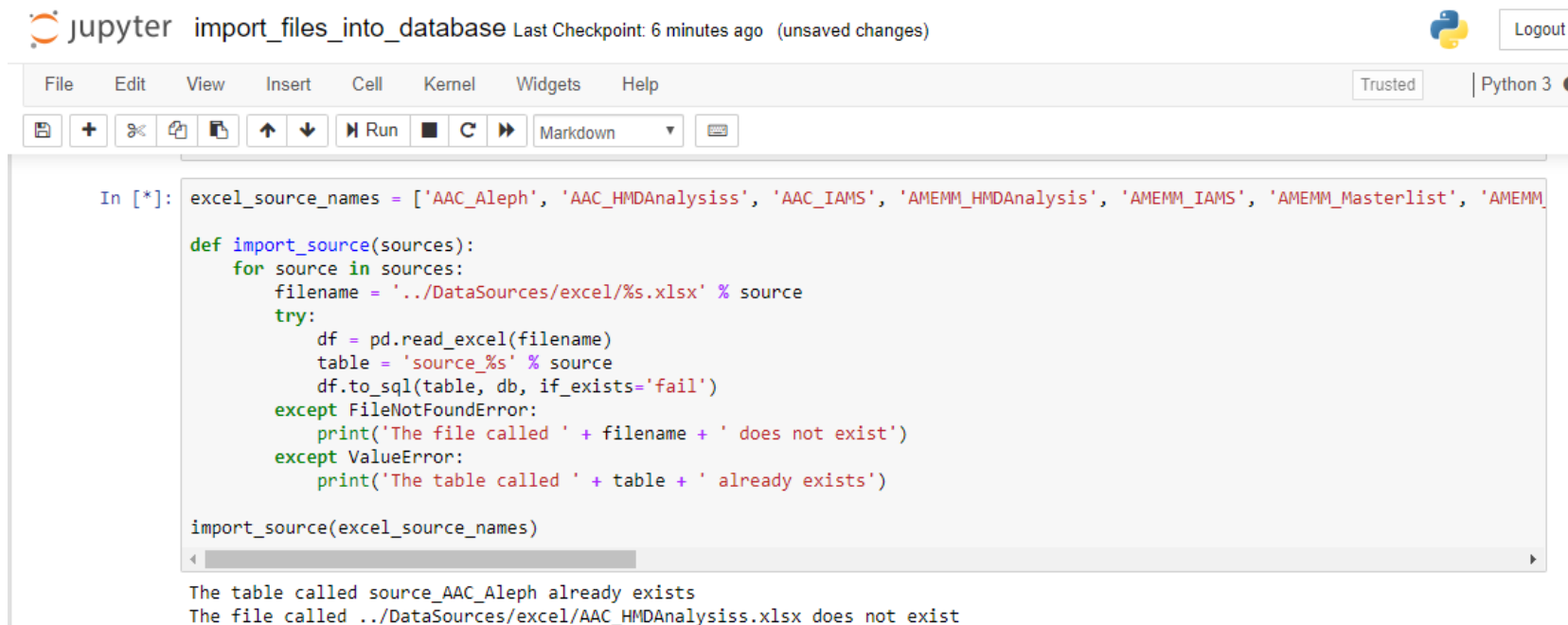


After creating my first database tables, I began manually entering sample data using the ‘Browse Data’ table in DB Browser for SQLite. Some of this data came directly from corresponding columns in my source tables, while others were entered from personal knowledge. Although I knew that I would later want to write a script to do the metadata input directly from data sources, I decided first to manually enter the data exactly as I wanted it to appear. This allowed me to easily see where the data would later need to be transformed without having to write code at this point to do so. This turned out to be especially time-saving as I ended up changing my schema several times and would have had to make many changes to any code I wrote to make these transformations.

Database Structure   Browse Data   Edit Pragmas   Execute SQL						
Table: files    						
	file_id	folder_id	ext_mismatch	sha256_hash	file_size	mod_date
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	False	50ba9310a26f3d72eef400eee3c0464c2b06ac8edbdec094c663b5f2976d8fda	95056216	2017-08-14T11:40:32+01:00
2	2	2	False	3d0a2024f12cc87a4af8ed95c82d545e5f16530c53a9251450acf97cd5b480a4	118488515	2018-02-23T10:14:02Z
3	3	3	False	b5d069dcc837f101bd676dea83b137f58d5306b0214630f65badd97d762343b	74938504	2015-04-02T12:04:26+01:00
4	4	4	True	ff602f7eeb24442c251f7f573e9465a9f8bb7277b2f577b6c65b0ca897892919	102065984	2015-03-12T17:23:02Z
5	5	5	False	d9993406b07d1b945c11a810ece996444566ae4ba53a949fd63b88d8b627a226	89940548	2015-01-26T15:05:48Z
6	6	6	False	1dc2e41c1fbc2dd8c7761afdd25b89723fcbf763ab8e9abeb18eb096aae0a197	117830433	2018-11-14T11:01:32Z
7	7	7	False	9321c0d387cc37699f7bb7ce0536b983e179a6f7dc2282859bb72e9266864177	42374907	2002-09-27T12:06:27+01:00
8	8	8	False	50ba9310a26f3d72eef400eee3c0464c2b06ac8edbdec094c663b5f2976d8fda	95056216	2017-08-14T11:40:32+01:00

Once I had my tables and some sample data in my single PIMMS database, I began writing Python scripts that would help me import and analyse the data itself. I experimented with different Python IDEs and code editors for this part of my project, including Anaconda, Notepad ++ and Jupyter Notebooks. I found Notepad ++ best for experimenting and drafting code, while Jupyter was best suited to running the code and adding informational text. Since I’ve only had a very limited amount of training in Python in the introductory module of this course, I found this the most difficult part of the project. I had to do a lot of research online, most notably on Stack Overflow, and attended several London Codebar sessions to work directly with experienced Python tutors who were able to help me troubleshoot and test my code. I also created a Github account to store and share my Jupyter notebooks as I worked on them (Green, 2020).

The first script I wrote was one to import all the data from my source Excel and CSV files into a series of new 'source' tables in my SQLite database. I analysed the different error messages that I was most likely to get and added them as exceptions to make these errors clear to any user of this script. For instance, if the filename is typed incorrectly or does not exist in the specified folder, it will print text specifying which filename does not exist. I chose to also print an error if that file had already been imported and a table had already been created for it. This allows the user to either manually delete the table to write over it. This script could be adapted in future if we find it more helpful to automatically delete the table before importing it if we want to continually update the database with new data imports.



The screenshot shows a Jupyter Notebook window titled 'import\_files\_into\_database'. The interface includes a top bar with the Jupyter logo, the notebook title, and a 'Logout' button. Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A toolbar contains icons for saving, adding new cells, undo, redo, and running code. The main area displays a code cell with the following Python code:

```
In [*]: excel_source_names = ['AAC_Aleph', 'AAC_HMDAnalysis', 'AAC_IAMS', 'AMEMM_HMDAnalysis', 'AMEMM_IAMS', 'AMEMM_Masterlist', 'AMEMM']

def import_source(sources):
    for source in sources:
        filename = '../DataSources/excel/%s.xlsx' % source
        try:
            df = pd.read_excel(filename)
            table = 'source_%s' % source
            df.to_sql(table, db, if_exists='fail')
        except FileNotFoundError:
            print('The file called ' + filename + ' does not exist')
        except ValueError:
            print('The table called ' + table + ' already exists')

import_source(excel_source_names)
```

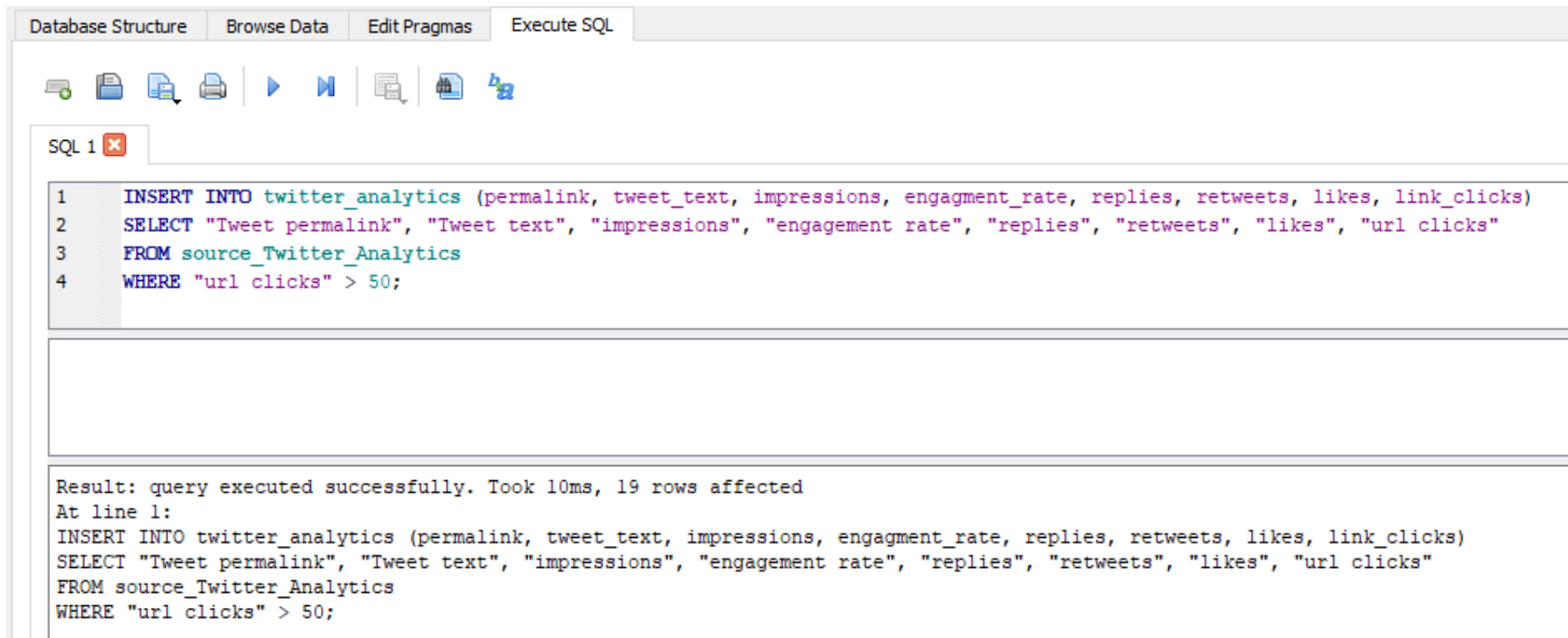
Below the code cell, the output is displayed:

```
The table called source_AAC_Aleph already exists
The file called ../DataSources/excel/AAC_HMDAnalysis.xlsx does not exist
```

While working on my database, I tried exporting and emailing the database to a few people for feedback, which caused me to rethink how much data I was importing into the database and how it was fundamentally structured. I experimented with putting all of my source tables and target tables in the same database, but once I started importing large source tables I saw how large this made my single file. After doing some research and learning that it was possible to query across two databases, I decided to split my database into two different databases: PIMMS.db and PIMMS\_sources.db. This worked well at first, but when I later came to do more queries across the two databases, I found it introduced a layer of complexity that outweighed the benefit of

the smaller file size. In the end I decided to combine the target tables and source tables into one database and include less source tables to keep the file size manageable and the database running smoothly.

After successfully employing my Python scripts to import data into the source tables, I wrote a SQL query to copy select data from source tables into the corresponding columns of the target tables according to my metadata mapping Google sheet. I tested this basic SQL query by running it over several source/target tables in the 'Execute SQL' tab of DB Browser for SQLite. This allowed me to see how long the query took to execute and if it was successful. I could then return to the 'Browse Data' tab to see the data in the table and make sure it was mapped and filtered as intended.



```
Database Structure | Browse Data | Edit Pragma | Execute SQL

SQL 1 ✕

1  INSERT INTO twitter_analytics (permalink, tweet_text, impressions, engagment_rate, replies, retweets, likes, link_clicks)
2  SELECT "Tweet permalink", "Tweet text", "impressions", "engagement rate", "replies", "retweets", "likes", "url clicks"
3  FROM source_Twitter_Analytics
4  WHERE "url clicks" > 50;

Result: query executed successfully. Took 10ms, 19 rows affected
At line 1:
INSERT INTO twitter_analytics (permalink, tweet_text, impressions, engagment_rate, replies, retweets, likes, link_clicks)
SELECT "Tweet permalink", "Tweet text", "impressions", "engagement rate", "replies", "retweets", "likes", "url clicks"
FROM source_Twitter_Analytics
WHERE "url clicks" > 50;
```

To make this easier to run in future, I adapted this SQL query into a Python function, replacing the table names and conditions with variables to be used where necessary. I also used the Google Sheet metadata mapping to generate a data dictionary table in my database. This allows me to generate lists of the up-to-date column names from both my source and target tables using the table names assigned to the variables in the Python function. Since the

column names in the sources themselves are subject to change, as well as the target tables/column names in my schema, this Python function and data dictionary table will help keep these correctly mapped.

Database Structure   Browse Data   Edit Pragmas   Execute SQL										
Table: source_data_dictionary		<div> <div>New Record</div> <div>Delete Record</div> </div>								
index	data_source	source_table	source_column	target_column	transformation	target_table	Data Type	Primary Key	Foreign Key	Descript
	sieg	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1 24	Siegfried Report	source_AAC_Siegfried_201908	FILE_PATH	file_name	Truncate to file name or keep full path?	file_history	TEXT	PK	NULL	NULL
2 25	Siegfried Report	source_AAC_Siegfried_201908	NAME	file_name	Use instead of FILE_PATH if truncating	file_history	TEXT	PK	NULL	NULL
3 28	Siegfried Report	source_AAC_Siegfried_201908	EXTENSION_MISMATCH	ext_mismatch	Change case, ie. FALSE to False	files	INT	NULL	NULL	NULL
4 29	Siegfried Report	source_AAC_Siegfried_201909	SIZE	file_size	No	files	INT	NULL	NULL	NULL
5 30	Siegfried Report	source_AAC_Siegfried_201910	LAST_MODIFIED	mod_date	No	files	DATE	NULL	NULL	NULL
6 31	Siegfried Report	source_AAC_Siegfried_201911	SHA256-HASH	sha256_hash	No	files	TEXT	NULL	NULL	NULL

## Analysis on the software

After building my PIMMS database prototype, I tried using it to reproduce reports created using existing methods and generating new reports by combining data in ways that were not previously possible. This included writing both SQL queries and Python scripts to analyse data from single tables as well as joining 2-3 tables to unlock the true potential of this software solution. Throughout the process, I exported results of my queries as csv files to share with colleagues and solicit feedback. I also saved some of these sample queries as ‘Views’ in my SQL database so the results could be easily retrieved and updated as the data in the database is updated.

Before even joining data from multiple tables, I found a lot of value in being able to more easily analyse data held within individual tables. In some cases, this data had been in large csv files or unwieldy databases making it difficult to query and analyse the data held within. Once I had imported the raw data from these systems into my database, I could run simple SQL queries and Python scripts to better understand the data and generate a range of helpful reports. For example, I used a SQL query to run several calculations necessary to our ongoing storage analysis using the Siegfried report previously analysed with Python Pandas (Moss, 2020). These calculations took less than a minute to run over a table with 1,328,610 rows and included calculations on the things most commonly asked when doing storage analysis on our BL networks:

- How many tiff files are there?
- What is the minimum, maximum, and average file size?

- What is the total file size?

1	SELECT
2	COUNT(ID) AS tif_count,
3	MIN(SIZE) AS min_tif_size_B,
4	MAX(SIZE) AS max_tif_size_B,
5	MAX(SIZE)/1024/1024 AS max_tif_size_MB,
6	AVG(SIZE)/1024/1024 AS avg_tiff_size_MB,
7	AVG(SIZE)/1024/1024/2012/1024 AS avg_tiff_size_TB,
8	SUM(SIZE)/1024/1024/1024 AS total_tiff_size_TB
9	FROM source_AAC_Siegfried_201908
10	WHERE EXT = 'tif';

	tif_count	min_tif_size_B	max_tif_size_B	max_tif_size_MB	avg_tiff_size_MB	avg_tiff_size_TB	total_tiff_size_TB
1	1251098	626	1867602612	1781	57.6770198090921	2.7994639491708e-05	68

Result: 1 rows returned in 28076ms  
At line 1:  
SELECT

```

COUNT(ID) AS tif_count,
MIN(SIZE) AS min_tif_size_B,
MAX(SIZE) AS max_tif_size_B,
MAX(SIZE)/1024/1024 AS max_tif_size_MB,
AVG(SIZE)/1024/1024 AS avg_tiff_size_MB,
AVG(SIZE)/1024/1024/2012/1024 AS avg_tiff_size_TB,
SUM(SIZE)/1024/1024/1024 AS total_tiff_size_TB
FROM source_AAC_Siegfried_201908
WHERE EXT = 'tif';

```

After writing and testing this SQL query in DB Browser for SQLite, I converted it to a Python script that can be run outside of the database and have the results displayed in a Pandas dataframe. Although I ran these queries specifically on data held one Siegfried report, these scripts could be easily applied to reports on other sets of data. One of the time-consuming elements of this work is generating the metadata necessary to do this analysis, as described in the Literature Review. In future, this Python script could be adapted to analyse the files on the network directly without having to separately generate the metadata. This script could also be adapted to analyse different file types or portions of the data, for example one specific folder. In addition, the Python maths library could be called to display these findings using visualisations like charts and graphs. By joining this table with the file\_history table and additional Siegfried reports, it will also be possible to calculate the rate of file growth— this is something that is of interest for storage planning and budgets.



```

In [1]: #Import the SQLite3 library and name it 'sql'
import sqlite3 as sql
#Import the Pandas library and name it 'pd'
import pandas as pd

#Assign an SQL query to meaningful variable
calculate_file_sizes = "SELECT COUNT(ID) AS tif_count, MIN(SIZE) AS min_tif_size_B, MAX(SIZE) AS max_tif_size_B, MAX(SIZE)/1024/1024 AS max_tif_size_MB"

def run_query(query):
    #Create the connection to the database (saved as PIMMS.db)
    con = sql.connect(r'../DatabaseDesign/PIMMS.db')
    #Create the dataframe from a SQL query defined above
    df = pd.read_sql_query(query, con)
    #Print the results of the query
    print(df)
    #Close the connection to the database
    con.close()

#Run the Python function using the SQL query in the query variable assigned above
run_query(calculate_file_sizes)

```

	tif_count	min_tif_size_B	max_tif_size_B	max_tif_size_MB \
0	1251098	626	1867602612	1781

	avg_tiff_size_MB	avg_tiff_size_TB	total_tiff_size_TB
0	57.67702	0.000028	68

I also tested the functionality of querying data held across multiple tables in the database using SQL joins, then exporting these results as csv files and saving them as 'Views' in the database (see Appendix – Screenshots of SQLite database). These sample queries included identifying select columns from 2-3 different tables and renaming certain column names in the output for clarity. I identified the matching elements in the different tables that held this data together – an often-frustrating process that made me reconsider the complexity of my schema. In defining my schema, I had made the choice to sometimes break data from the same source table into multiple target tables, although I attempted to keep this to a minimum. This was meant to limit the amount of redundant data across tables, although it also ended up making my joins more complex. One of the next steps of my project will involve revisiting this schema and deciding whether to simplify it to make the joins more straightforward. During this process, I also found myself querying the source tables more often than the target tables, as they held more data at this point in development. Although mapping this data into target tables will

make it easier to streamline column names and query across tables in future, the time it takes to do this mapping may not be worth the effort - it has proved fairly straightforward to query this 'raw' source data directly.

## Conclusions

I believe that this project shows that a relational database solution can help the British Library more efficiently and effectively conduct analysis of 'legacy' digitised content, as well as make even more connections between metadata sources in the process. The most rewarding feedback has come from colleagues involved in analysing and enhancing metadata about digitised content across the BL. Several of them have said this solution would save them and their teams months of work they would normally spend manually copying and pasting data from a range of data sources in preparation for things like copyright assessment and inventorying their digitised collections. Most staff are dependent on Excel spreadsheets for this work and often end up with more than one spreadsheet for analysing the same content. Over time, more colleagues work on the same spreadsheets which have the tendency to become messy and unwieldy. It also becomes difficult over time to determine which spreadsheets and which metadata is 'authoritative' and up-to-date. By simply combining the data held in these different spreadsheets, they can have a 'fresh' start and a solid foundation on which to continue their work. I believe that if we can implement this database solution on the BL network, we could save staff a lot of time and produce higher quality and more accurate analysis reports about their digitised content in the process.

Next steps for this project therefore involve sharing this prototype with more staff across the BL and investigating options for possible implementation. In addition to a shortened written report, I plan to present this solution in the form of a short video presentation – a screencast of me demonstrating my software solution with voiceover commentary. Based on feedback from colleagues, I plan to continue improving this software solution and testing more queries and reports useful to a range of staff around the Library. In time, I hope that I can gather enough institutional buy-in and support to implement this software solution at the British Library.

## References

British Library. 2018. *Heritage Made Digital*. [online] Available at: <<https://www.bl.uk/projects/heritage-made-digital>> [Accessed 6 March 2020].

British Library. 2020. *Explore The British Library*. [online] Available at: <[http://explore.bl.uk/primo\\_library/libweb/action/search.do?vid=BLVU1](http://explore.bl.uk/primo_library/libweb/action/search.do?vid=BLVU1)> [Accessed 19 April 2020].

Codebar. 2020. *Codebar*. [online] Available at: <<https://codebar.io/london>> [Accessed 6 February 2020].

Creately. 2020. *Chart, Diagram & Visual Canvas Software*. [online] Available at: <<https://creately.com/>> [Accessed 12 April 2020].

Datacarpentry.org. 2020. *Accessing Sqlite Databases Using Python And Pandas – Data Analysis And Visualization In Python For Ecologists*. [online] Available at: <<https://datacarpentry.org/python-ecology-lesson/09-working-with-sql/index.html>> [Accessed 1 March 2020].

Ex Libris. 2020. *Aleph Integrated Library System For Libraries | Ex Libris*. [online] Available at: <<https://www.exlibrisgroup.com/products/aleph-integrated-library-system/>> [Accessed 10 April 2020].

Ex Libris Knowledge Center. 2020. *New Primo User Interface*. [online] Available at: <[https://knowledge.exlibrisgroup.com/Primo/Product\\_Documentation/Primo/New\\_Primo\\_User\\_Interface](https://knowledge.exlibrisgroup.com/Primo/Product_Documentation/Primo/New_Primo_User_Interface)> [Accessed 19 April 2020].

Fenlon, K., Colleen Fallaw, Timothy Cole, and Myung-Ja Han. 2014. *A preliminary evaluation of HathiTrust metadata: assessing the sufficiency of legacy records*. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '14)*. IEEE Press, 317–320. Available at: <https://dl.acm.org/doi/abs/10.5555/2740769.2740824> (Accessed: 10 April 2020).

Green, J., 2020. *JessicaGreen/PIMMS*. [online] GitHub. Available at: <<https://github.com/JessicaGreen/PIMMS>> [Accessed 4 April 2020].

International Association of Sound and Audiovisual Archives. 2020. *6.2.1 Submission Information Package (SIP)*. [online] Available at: <<https://www.iasa-web.org/tc04/submission-information-package-sip>> [Accessed 19 April 2020].

International Council on Archives. 2000. *ISAD(G): General International Standard Archival Description*. [online] Available at: <[https://www.ica.org/sites/default/files/CBPS\\_2000\\_Guidelines\\_ISAD\(G\)\\_Second-edition\\_EN.pdf](https://www.ica.org/sites/default/files/CBPS_2000_Guidelines_ISAD(G)_Second-edition_EN.pdf)> [Accessed 19 April 2020].

Library of Congress. 2008. *MARC In XML*. [online] Available at: <<https://www.loc.gov/marc/marcxml.html>> [Accessed 7 April 2020].

Lucidchart. 2020. [online] Available at: <<https://www.lucidchart.com/>> [Accessed 19 April 2020].

Morgan, C. 2013. *Ebscohost Login*. [online] Search.ebscohost.com. Available at: <<http://search.ebscohost.com/login.aspx?direct=true&db=llf&AN=101801356&site=ehost-live>> [Accessed 9 April 2020].

Moss, H., 2020. *Harryjmooss/Hmdsiegfriedanalysis*. [online] GitHub. Available at: <<https://github.com/harryjmooss/HMDSiegfriedAnalysis>> [Accessed 4 April 2020].

National Archives. 2017. *DROID: User Guide*. [online] Available at: <<https://www.nationalarchives.gov.uk/documents/information-management/droid-user-guide.pdf>> [Accessed 11 February 2020].

Neil, W., 2019. *The British Library'S New Collection Metadata Strategy - Digital Scholarship Blog*. [online] Blogs.bl.uk. Available at: <<https://blogs.bl.uk/digital-scholarship/2019/04/the-british-librarys-new-collection-metadata-strategy.html>> [Accessed 12 February 2020].

Pennock, M., 2016. *Preservation Of Ingested Collections: Assessments, Sampling & Action Plans (PICASA)*. London: British Library.

stackoverflow.com. 2017. *Openpyxl Read Out Excel And Save Into Database*. [online] Stack Overflow. Available at: <<https://stackoverflow.com/questions/47432988/openpyxl-read-out-excel-and-save-into-database>> [Accessed 11 April 2020].

stackoverflow.com. 2018. *Mysql Select All Columns From One Table And Some From Another Table*. [online] Stack Overflow. Available at: <<https://stackoverflow.com/questions/3492904/mysql-select-all-columns-from-one-table-and-some-from-another-table>> [Accessed 3 April 2020].

SQLite Tutorial. 2020. *Sqlite INNER JOIN With Examples*. [online] Available at: <<https://www.sqlitetutorial.net/sqlite-inner-join/>> [Accessed 5 April 2020].

TechSpot. 2019. *Quickly Convert Between Storage Size Units: KB, MB, GB, TB & 512 Byte Blocks*. [online] Available at: <<https://www.techspot.com/news/68482-quickly-convert-between-storage-size-units-kb-mb.html>> [Accessed 18 April 2020].

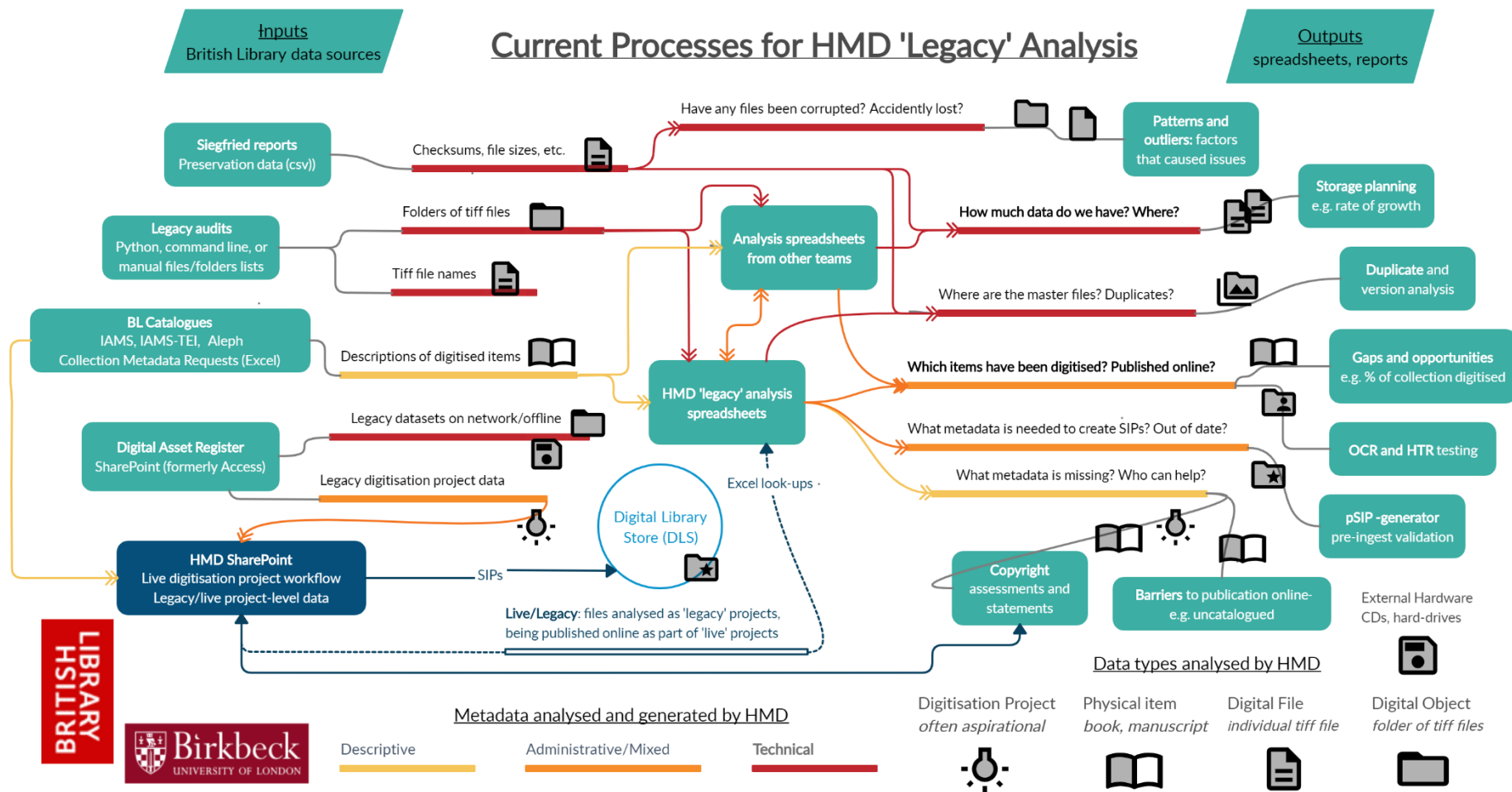
W3schools.com. 2020. *SQL INSERT INTO SELECT Statement*. [online] Available at: <[https://www.w3schools.com/sql/sql\\_insert\\_into\\_select.asp](https://www.w3schools.com/sql/sql_insert_into_select.asp)> [Accessed 13 April 2020].

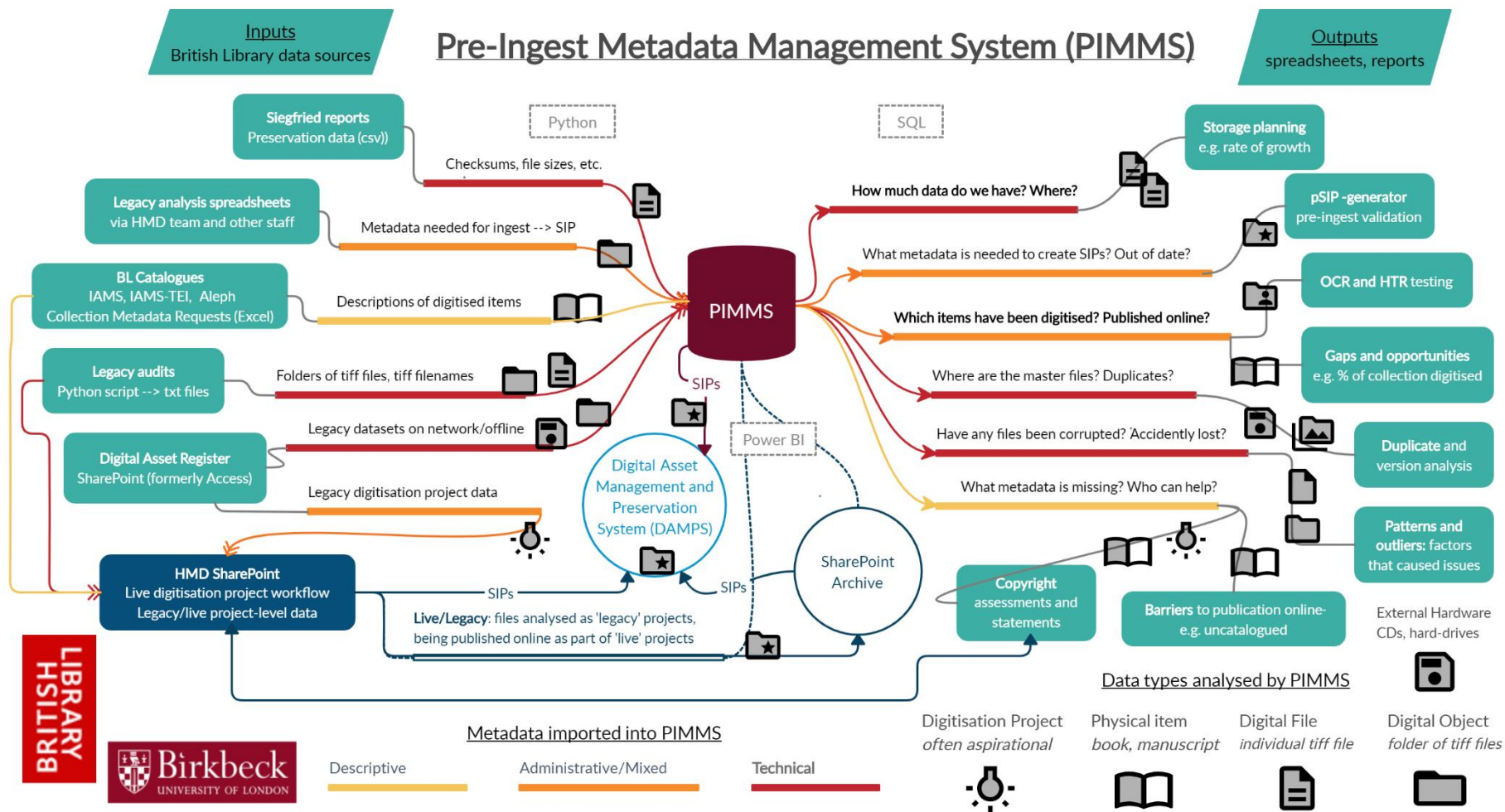
## Appendix

1. Project Plan
2. Model of current analysis processes
3. Model of PIMMS software solution
4. PIMMS Entity Relationship Diagram (database schema)
5. Screenshots of PIMMS database queries

PROJECT TASK		SKILL	RESOURCES/TOOLS
<b><u>Project Proposal</u></b>			
1	Discuss project ideas with BL supervisor	Communication	Sandra Tuppen
2	Write one-page 'pitch' for project	Communication	Microsoft Word
3	Circulate 'pitch' for feedback	Communication	BL Staff
4	Discuss project idea with BBK tutor	Communication	Martyn Harris
5	Write proposal (1500 words)	Communication	Microsoft Excel, Word
6	Complete proposal form	Communication	Moodle

<b><u>Data Sources</u></b>			
7	Export sample data from various BL systems (csv, Excel, txt files)	File formats	SharePoint, Access, Excel, BL catalogues
8	Write scripts to transform the data where necessary before import	Python, SQL	Jupyter Notebooks, Github
9	Manually insert sample data into model database (10 rows/items)	SQL	Jupyter Notebooks, SQLite
10	Import sample data into model database (100 rows/items)	Python, SQL	Jupyter Notebooks, SQLite
<b><u>Database Design</u></b>			
11	Make spreadsheet of all columns from data sources (task 7)	SQL	Google sheets
12	Create Entity Relationship Diagram (ERD)	SQL	Pen/paper, creately, office white board
13	Identify tables based on columns spreadsheet (task 12)	SQL	Pen/paper, creately, office white board
14	Identify and test primary and foreign keys	SQL	Pen/paper, creately, SQLite
15	Gather feedback on database design and potential uses	Communication	BL staff
16	Build tables and relationships in model database	SQL	SQLite
17	Finalise ERD: tables, keys, relationships, and attributes	SQL	creately, office white board
<b><u>Database Testing and Queries</u></b>			
18	Gather/export example outputs and reports	File formats	BL Staff
19	Gather feedback on potential queries and research questions	Communication	BL Staff
20	Write sample queries for analysis of metadata in database	Python, SQL	Jupyter Notebooks, SQLite
21	Run sample queries over first set of data (task 9)	Python, SQL	Jupyter Notebooks, SQLite
22	Test validity of results and adjust schema/scripts	Python, SQL	Jupyter Notebooks, SQLite
23	Run sample queries over second set of data (task 10)	Python, SQL	Jupyter Notebooks, SQLite
24	Test validity of results and adjust schema/scripts	Python, SQL	Jupyter Notebooks, SQLite
<b><u>PROJECT SUBMISSION</u></b>			
25	First draft of report (6000-8000 words)	Communication	Word, Google sheets, Jupyter, creately
26	Draw process diagrams: current and proposed processes	Communication	creately
27	Gather feedback on first draft	Communication	BL staff
28	Make final edits and corrections	Communication	Word, Excel, Jupyter, creately



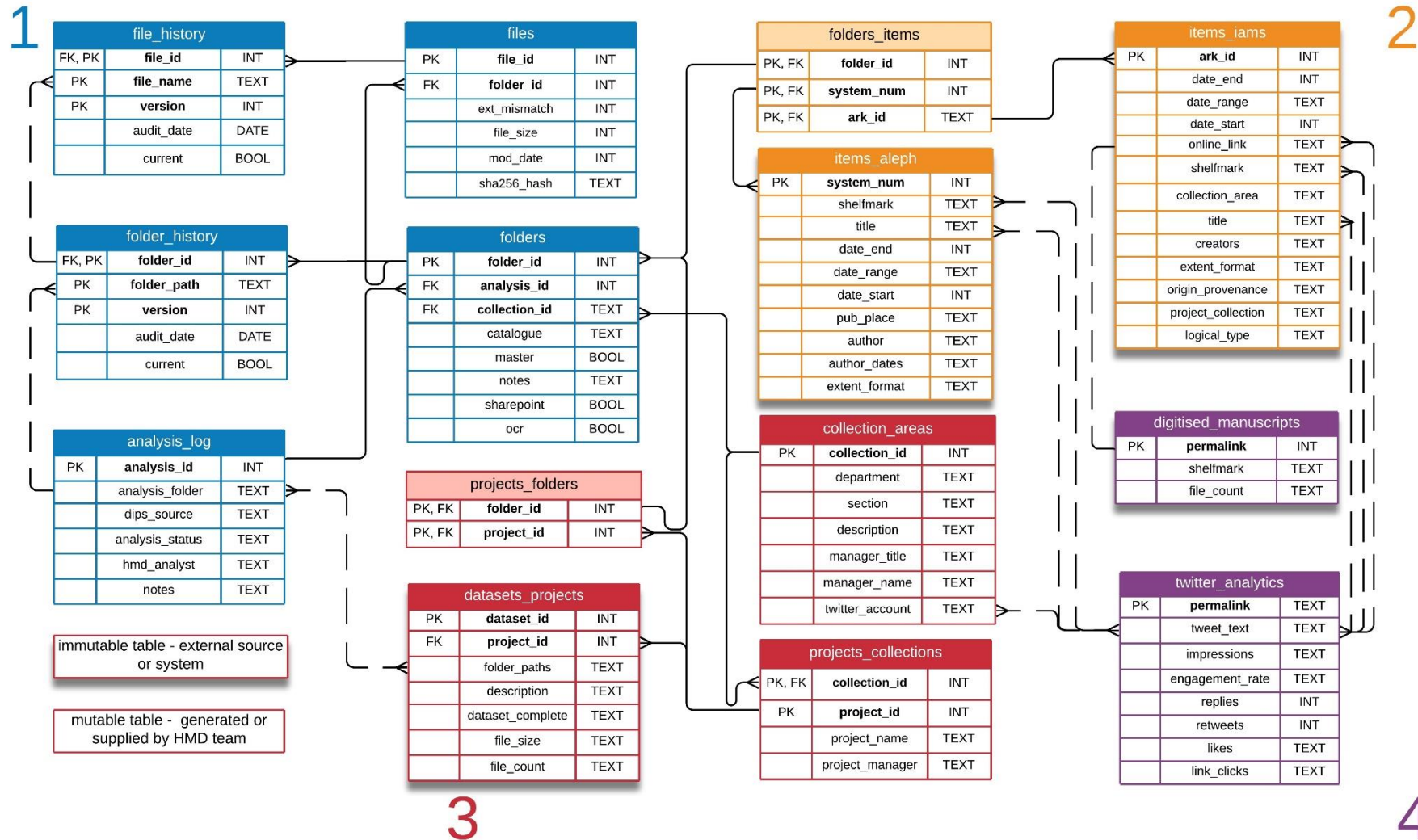




# Pre-Ingest Metadata Management System (PIMMS)

## Entity Relationship Diagram

Jessica Green, Heritage Made Digital, British Library  
Computing for Cultural Heritage, Birkbeck, University of London





## Exporting a CSV file for select data from two tables in the database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```

1 SELECT 'source_AMEMM_SelfAnalysis'.ms_shelfmark AS 'Shelfmark',
2 'source_AMEMM_SelfAnalysis'.pro_title AS 'Project Title',
3 'source_AMEMM_SelfAnalysis'.pro_details AS 'Project Details',
4 'source_AMEMM_SelfAnalysis'.ms_location AS 'Manuscript Location',
5 source_AMEMM_IAMS.*
6 FROM 'source_AMEMM_SelfAnalysis'
7 INNER JOIN source_AMEMM_IAMS ON 'source_AMEMM_SelfAnalysis'
8 WHERE source_AMEMM_IAMS.Languages LIKE '%$French%';

```

Export data as CSV

Column names in first line ☒

Field separator

Quote character

New line characters

Save Cancel

		Languages				Scope & content	Scale	Scale
1	his manu...	English, Old; French; Latin	ang; fr			This manuscript contains 3 separate items, boun...	NULL	NULL
2	his manu...	French	fre			Le Trésor des Histoires or Trésor de Sapience is ...	NULL	NULL
3	is manus...	Anglo-Norman; English, Old; French; Latin	ang; fr			Contents:.. ff. 2r-6r: Lunar and paschal tables; d...	NULL	NULL
4	his manu...	French, Old; Latin	fro; lat	Latin	Latn	Contents:.. ff. 1r-27v: The Winchcombe Chronicle...	NULL	NULL
5		French, Middle; Latin	frm; lat	Latin	Latn	This composite manuscript contains two parts th...	NULL	NULL
6	his manu...	Anglo-Norman; French; Latin	fre; lat; xno	Latin	Latn	This manuscript contains a variety of texts bound...	NULL	NULL
7	his manu...	French; French, Middle	fre; frm	Latin	Latn	Contents:.. Six texts on the Orient, including a set...	NULL	NULL
8	his manu...	English, Old; French; Latin	ang; fr; lat	Latin	Latn	This manuscript contains two separate collection...	NULL	NULL

AutoSave ☐ Off

French\_AMEMM\_items

Search

Jessica Green JG

File Home Insert Page Layout Formulas Data Review View Help

Share Comments

AF1

Languages

	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1	Title	Date range	Start date	End date	Calendar	Era	Creators	Extent & f	Access co	Access da	User cond	Languages	Language	Scripts	Script cod	Scope & c	Scale	Scale desi Pr
2	Bede, De	Mid-11th	1025	1199		CE	None	1 volume.	Restrictions to acces	Letter of i		English, Old; French; Latin	ang; fre; l	Latin	Latn	This manuscript contains 3 separa		
3	Le Tr	4th quarte	1475	1499		CE	None	A parchm	Restrictions to acces	Letter of i		French	fre	Latin	Latn	Le Tr		
4	Psalter (th	3rd quarte	1050	1199		CE	None	A parchm	Restrictions to acces	Letter of i		Anglo-Norman; English, Old; French	ang; fre; l	Latin	Latn	Contents:.. ff. 2r-6r: Lunar and pas		
5	Winchcom	1st half of	1100	1324		CE	Abbo of F	A parchm	Restrictions to acces	Letter of i		French, Old; Latin	fro; lat	Latin	Latn	Contents:.. ff. 1r-27v: The Winchco		
6	Cartulary	13th centu	1200	1399		CE	None	A parchm	Unrestricted			French, Middle; Latin	frm; lat	Latin	Latn	This composite manuscript contai		
7	Chronicle	c.1100-c.1	1100	1650		CE	None	Parchmen	Restrictions to acces	Letter of i		Anglo-Norman; French; Latin	fre; lat; x	Latin	Latn	This manuscript contains a variety		
8	A collecti	1400-1415	1400	1415		CE	None	A parchm	Restrictions to acces	Letter of i		French; French, Middle	fre; frm	Latin	Latn	Contents:.. Six texts on the Orient		

## Creating a view for select data from three tables in the database

Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1

```

1 SELECT folder_history.folder_path AS 'folderpath', items_aleph.*
2 FROM items_aleph
3 INNER JOIN folder_history ON folder_history.folder_id = folders_items.folder_id
4 INNER JOIN folders_items ON folders_items.system_num = items_aleph.system_num;

```

	folderpath	system_num	shelfmark	pub_place	title	date_end	date_range	date_start	author	author_dates	
1	AAS Storage\Korean (HT)\NLK Project\15026_d_...	17852506	15026.d.3	Korea	880-02 Pŏnyŏk myŏngŭijip / Pobun p'yŏn.	NULL	NULL	NULL	NULL	NULL	7 volumes
2	AAS Storage\Korean (HT)\NLK Project\15212_e_...	17846046	15212.e.8	Korea	880-02 Samgo illam myŏngsŏ sangnon / Pang Ky...	NULL	NULL	NULL	NULL	NULL	3 volumes

DB Browser f... ? X

Please specify the view name

Aleph\_items

OK Cancel

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: Aleph\_items

New Record Delete Record

	folder_path	system_num	shelfmark	pub_place	title	date_end	date_range	date_start	author	author_dates	extent_format
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	AAS Storage\...	17852506	15026.d.3	Korea	880-02 Pŏnyŏ...	NULL	NULL	NULL	NULL	NULL	7 volumes in ...
2	AAS Storage\...	17846046	15212.e.8	Korea	880-02 Samg...	NULL	NULL	NULL	NULL	NULL	3 volumes in ...