Albatross: Rolling on a Sea of Data

Annette Bailey
Virginia Tech University Libraries, afbailey@vt.edu

Tracy Gilmore
Virginia Tech University Libraries, tgilmore@vt.edu

Leslie O'Brien
Virginia Tech University Libraries, lobrien@vt.edu

Anthony D. Wright de Hernandez
Virginia Tech University Libraries, antwri@vt.edu

Annette Bailey, Tracy Gilmore, Leslie O'Brien, and Anthony D. Wright de Hernandez, "Albatross: Rolling on a Sea of Data" (2016). *Proceedings of the Charleston Library Conference.*
http://dx.doi.org/10.5703/1288284316438

Albatross: Rolling on a Sea of Data

Annette Bailey, Assistant Director, Electronic Resources and Emerging Technology Services, Virginia Tech University Libraries

Tracy Gilmore, Collections Assessment Librarian, Virginia Tech University Libraries

Leslie O'Brien, Director, Collections and Technical Services, Virginia Tech University Libraries

Anthony D. Wright de Hernandez, Resident Librarian, Virginia Tech University Libraries

Abstract

Big deals and journal package incentives are an increasing reality for academic libraries, yet the solutions for evaluating these package scenarios in a timely, cost-effective manner are few. The proliferation of these offers requires the examination of numerous and complex questions. There is a need to know the utilization and strength of a package, the inflation costs for various titles and packages, and the ability to identify cost trends. A team of librarians at Virginia Tech created a solution for addressing these concerns and for managing their journal data by designing and developing an in-house database. Albatross, named in reference to *The Rime of the Ancient Mariner*, is a database created to gather journal usage data and cost data in a central environment where the data can then be queried to use in return-on-investment analysis and journal package assessments.

Introduction

Over the last few years, a team at Virginia Tech University Libraries has been developing a database called Albatross. This database is intended to assist the Collections team in determining return on investment for our electronic journal subscriptions with an eye toward analyzing other electronic resources in the future.

There are several factors that drove the decision to build a database internally rather than using an existing third-party solution, including previous experiences with third-party systems, budgetary concerns, direct control of our own data, and adaptability.

Several third-party products were tried, and none provided a satisfactory level of accuracy and flexibility. For example, when SUSHI was tested with the library's Innovative Interfaces Inc. (III) Millennium system, it either didn't import certain datasets, or it didn't import them correctly. As a result of repeated unsatisfactory experiences with third-party products, it was decided the best solution was to develop something internally that would be tailored to the library's needs.

With regard to budget, the collections team strives to preserve as much of the collections budget as possible for content or products that can't easily be developed internally. There are many free or low-cost database support products available now that weren't available in the past, which effectively reduce the costs associated with developing and updating an in-house database. Many of the library's staff have acquired skills in scripting and writing queries that used to require outsourcing to Information Technology staff. The cost of file storage space has come down dramatically over the last decade, making local hosting of a database feasible.

Another reason for internal development of a solution is increased control over the data. The library receives many nonstandard reports—even reports that claim to be up to the latest COUNTER 4 standards don't always conform. By managing the data ingest process internally, the library can adapt to changes in COUNTER standards and ensure that no nonstandard data is ingested into the database. The database can grow to accommodate a variety of queries for different purchasing models, allowing the creation of "what if" scenarios for journal packages by comparing list prices to discounted prices or incorporating cost sharing with consortia and buying groups.

https://doi.org/10.5703/1288284316438

Background

Since 2002 (the inception of Project Counter), the library has had at least four different methods of compiling and evaluating electronic resources usage data. Vendor products, spreadsheets, and Microsoft Access databases were all tried. The most recent method is this effort to develop a database. The database is designed to pull together journal usage and cost data in a central environment where that data can be queried for return on investment analysis.

From 2002 to 2006, Virginia Tech University Libraries kept track of usage data on a spreadsheet called the "big ugly database," or BUD. BUD worked pretty well for tracking usage at the package level, but it didn't allow for analysis of the individual journal. It also didn't have any cost data, so cost per use and cost trends had to be calculated separately. Different metrics were all combined into one spreadsheet, and warning messages were added to remind the collections staff about anomalies or special calculation instructions. At its peak, BUD contained fewer than 200 lines of data. By 2008, Virginia Tech University Libraries was collecting over 150 different COUNTER reports with over 40,000 lines of data annually.

After BUD, some attempts to use SUSHI to import into our III electronic resources management module were made, but SUSHI couldn't handle reports that either didn't truly conform to the COUNTER standards or didn't make any attempt to do so. Other third-party solutions were also tried. The products we tried had issues with non-normalized data, which caused journal titles to be duplicated or omitted entirely from reports. These solutions also required that we do a lot of work to supply the third-party with our internal cost data to perform cost-per-use analysis. On top of all the other issues with these solutions, the subscription costs for them came out of the collections budget, which in the late 2000s was not very stable.
In 2010, a Microsoft Access database was created. This database was called Foster (a nod to the BUD and an inside reference to Virginia Tech football). Foster was a functional Access database that used COUNTER reports in combination with bibliographic data from the catalog. It allowed analysis of a journal's use across multiple platforms, cost-per-use by subject, and cost-per-title by subject. It allowed control over the raw data and creation of complex queries and data relationships.

For various technical and administrative reasons, Foster was discontinued after a few years of use. The library has been looking for a new solution since that time. Some solutions that have been considered include the business intelligence systems licensed by our university, and new products like LibInsight by Springshare. After carefully considering the options, the team chose to proceed with a locally developed relational database to manage the rising sea of electronic resources data.

Database Design

The first step in creating this new database was consideration of the data to be analyzed and what tools were best for conducting that analysis. The database design began with a review of the data points and development of an entity relationship diagram (ERD) to reflect how the data are all related (see Figure 1).

Primary development of the database was focused on electronic journals. Much of the ERD is arranged around how the data ultimately connect back to the journal and how to uniquely identify each journal over time through name and publisher changes or when more than one platform offers access. The main data points track information about the journal, package, publisher, platform, and order.

There were a couple of design challenges related to the nature of the data. The first major challenge presented by the data was how to represent renaming or rebranding. The journals, publishers, and platforms all have variability in their names over time. This issue isn't one of errors or different spellings of names in the reports; it is rather a reflection of the tendency for corporate entities to evolve and rebrand over time. The digital object identifier (DOI) was selected as a static identifier for the journal, since it is required by the current COUNTER standard and is the only journal data point the team identified that remains static and invariable, even if the journal's title changes.

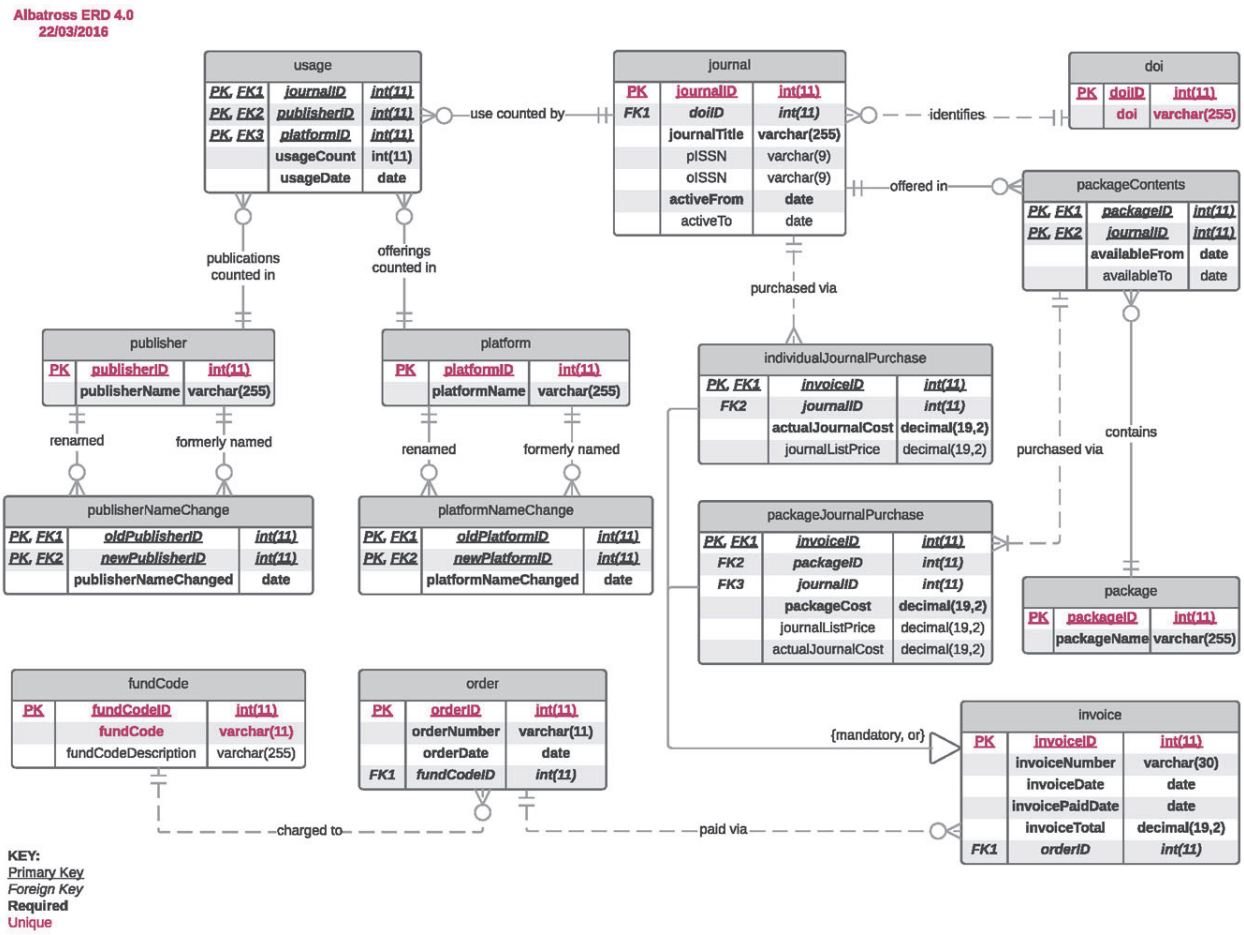

Figure 1. Entity relationship diagram at the time of database launch.

Another challenge was deciding how to reflect individual versus package purchases. The variable nature of how a journal is purchased is represented in the database as a decision point with subtables to enter the relevant data. The fact that cost can be per journal or per package makes calculating cost per use more complicated because there isn't always a neat individual price for each journal. To compensate for this lack, published list prices were added for use as part of the cost analysis.

The process of creating the ERD was made easier by using a paid software: Visual Paradigm. The team working on developing the database had some experience with database design but were not experts. Visual Paradigm was used to facilitate fast ERD development because it provides support for less experienced designers such as drop-down lists for data types, help with crow's-feet notation, and assistance correctly assigning primary and secondary keys.

After the initial ERD was created, it was migrated from Visual Paradigm into Lucidchart. Lucidchart was chosen for ongoing development work because it allows collaborative editing of diagrams at no charge for educational institutions, meaning there is no ongoing cost for this project. Since the ERD will be undergoing less frequent edits, a free option made more sense as a long-term ERD management software. Lucidchart is Web based and integrates with Google Drive, allowing everyone on the team to use it regardless of operating system. Lucidchart was not used for initial development since it does not provide the level of support and development assistance that Visual Paradigm does (i.e., there aren't any drop-downs or helps built in).

Data Collection and Cleaning

Most the data for the database comes from the COUNTER JR1 reports. These are reports of electronic journal usage following the standards set

forth by Project COUNTER, a nonprofit standards setting agency. An initial load of five years' worth of data was selected to allow for immediate analysis once the database was set up. Cleaning up five years of JR1 reports that had been previously analyzed required touching every single report from 2010 through 2015 to ensure that each report was consistently formatted. This meant moving year-to-date column totals on COUNTER 3 release reports from right to left, re-formatting non-COUNTER reports to look like COUNTER, and adding DOIs to all JR1 reports from before the DOI was a required field and many from after it was required.

About 70% of the reports on file did not have the DOIs required for entry into the database. For most, a simple look-up was all it took to add the DOI to the existing data. For journals where a DOI could not be located, an internal DOI schema was developed so that temporary identifiers could be assigned. All temporary DOIs begin with 10.9999, this is unique and easily identifiable as one created by the team. For journals with an associated ISSN, this identifier was used as the suffix (10.9999/2166-4072). However, usage data is often incomplete, so for those titles that had no ISSN, the journal title initials were used for the suffix (10.9999/abi). As registered DOIs become available for these items, the database will be updated to replace the temporary identifier with the real one. In addition to adding DOIs to the older reports, it was necessary to create a table that identified every title associated with a journal package or collection and assign them all specific identifiers and consistent DOIs.

After working to clean up the DOIs, it was then necessary to provide the associated list price for each of the journals for each year possible. The list prices were necessary because the purchase price data on file was associated with the package or collection and not the individual journal, and the goal was cost analysis at the individual journal level. List price information was obtained from published lists put out by vendors. Collection assessment team members gathered this information from as many sources as they could find. While it was not possible to gather all five years of back data, a sufficient amount was obtained to allow analysis once the database came online.

It was also necessary to include data on the orders and the prices paid in the database. This information was gathered by creating SQL queries against the III Sierra Integrated Library System currently in use. These SQL queries were created using the SierraDNA interface provided by III. Due to the back-end structure of Sierra, it was necessary to have someone familiar with SQL construct these queries, since figuring out how to connect two data points within Sierra can be a challenge. The final queries are designed so that a single filter can be changed to pull updated information from Sierra whenever desired.

The purpose of cleaning and creating consistent formatting for all data in our database was to enable both internal and external reporting of our journal usage over time. The data will be used to analyze our journal usage by platform, publisher, discipline, department, or college. It will also be used to provide multiple perspectives on usage and cost in reports to our liaisons and stakeholders in a clear, concise, and visually formatted way that contributes to a deeper understanding of how these resources are being used.

Looking Forward

Much of what has been accomplished so far has been the collection, cleaning, and managing of various COUNTER reports. Given the amount of work required to prepare five years of historic data for ingest into the database, there has been some concern over whether this will save time and improve reporting capabilities. To address these concerns, the team has been working on automating much of the data cleaning process. A series of Python scripts are under development that will automatically run SQL queries against Sierra to pull cost data, fill in missing DOIs on COUNTER reports, add new journals to the database while ensuring there is no duplication, and more. Many of these scripts use pattern analysis to determine the degree of similarity between data strings. If the computer thinks there is a match but can't be 100% certain, it will create output for human review. Using this scripting technique, much of the data cleaning process can be automated, and the team only needs to review those items about which the computer can't make a decision.

Once all initial data is loaded into the database, the most obvious next step is implementation and use of the database. Early testing of the database has allowed reporting that shows usage over time and cost-per-use by journal title. Once the database is

fully operational, there are plans to use the data within to consult specifically with subject liaisons and provide them with valuable insights and actionable data.

Beyond initial use of the journal database, future plans include expanding the number of data points included in the database. There is an increasing need to evaluate package scenarios. Big deals don't seem to be going away, and there are also evidence-based or usage-based package offers. It is important to know what titles in a package are being used, what the use of front file versus backfile articles is, and be able to calculate the inflation costs for different scenarios.

There is still some interest in using SUSHI to gather data for ingest. While this hasn't worked in the past, the new automated data cleaning processes in place may address the issues encountered previously. Even if SUSHI can't be incorporated, the team will be exploring ways to further automate the data gathering process to free up more employee hours for data analysis.

Finally, Virginia Tech is introducing performance-based budgeting in the near future, and the library has been asked to come up with metrics to demonstrate its success and value. The team working on Albatross will be developing new skills with data visualization to enable better presentation of the important information contained within the database.