Collaborative Data Management for Astronomy with the AstroDB Toolkit

The Need for a Collaborative Database Framework

We propose to fill a gap in the data management/sharing ecosystem for astronomers by providing a robust toolkit to empower astronomers to build databases of astronomical sources. In the "big data" era of large, long-standing missions such as TESS, Kepler, and JWST, astronomers find themselves managing increasingly large and unwieldy collections of target parameters, both observed and modeled. Currently, astronomers reinvent the wheel, spending time making technology choices, database design decisions, and web applications as opposed to being able to focus on the analysis and physical interpretation of the actual data. A plethora of different Astronomer-made "databases" exists, all with different tech stacks and underlying schemas. We propose to build an open-source tool that greatly lowers the technology burden on the astronomers and empowers them to make databases of astronomical sources using a common, interoperable framework.

The AstroDB Toolkit aims to serve the needs of individual astronomers and small-to-medium sized collaborations who need a data management system for hundreds to thousands of sources. The Toolkit will bridge the divide where a shared Google sheet is insufficient, but the dataset is either still living (e.g., follow-up observations are underway, new parameters being derived) or otherwise not appropriate for an institutional archive.

This project is particularly well aligned with several goals of TOPS and the HPOSS program. The AstroDB Toolkit will 1) increase access and discoverability of astronomical research data, 2) empower data sharing and support open science principles across astronomy disciplines, and 3) provide a tool that encourages community input and collaboration in astronomical data curation.

The AstroDB Toolkit

The technical choices that were made in the development of the AstroDB Toolkit followed several design requirements:

- Ability to represent complex data models in a simple, intuitive fashion,
- Support for datasets up to tens of thousands of astrophysical sources,
- Collaborative editing that enables community members to modify and maintain data holdings,
- Be usable privately for managing not-yet-public data,
- Interactively explore and visualize holdings via a website,
- Queryable locally via scripting without the need for an internet connection.

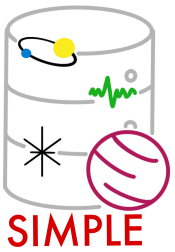
As described in the Technical Description, the AstroDB Toolkit meets all of these requirements. We have designed a tool that uses GitHub's features and its collaborative workflow. The Toolkit fuels open science by providing a common framework for databases, managed openly on GitHub. This shared framework makes databases easier for Astronomers to build and maximizes their interoperability, thus empowering open science and facilitating data sharing.

The AstroDB Toolkit consists of:

1. a template schema,
2. Python packages (`AstrodbKit`, `astrodb_utils`),
3. documentation and suggested workflows, and
4. a web application.

We are requesting support here to use the Toolkit to design **three new databases** covering three different astrophysics science domains: dwarf galaxies, tidal streams, and exoplanets. This design process will greatly expand the template schema, grow the functionality of the `astrodb_utils` package, and provide the scaffolding on which to build the much needed documentation. Improvements to the web application are beyond the scope of this one year proposal.

Proof of Concept: The SIMPLE Archive



The current poster-child of the AstroDB Toolkit is SIMPLE, the Substellar and IMaged PLanet Explorer Archive of Complex Objects (https://simple-bd-archive.org/). The SIMPLE database contains nearly 3,500 brown dwarfs with photometry, radial velocity, spectral types, and more. Holdings include data from NASA missions such as HST, JWST, and Spitzer in addition to ground-based data and modeled parameters.

Over the past few years, our team has built this database by ingesting new sources, adding spectra, refining the underlying schema, and updating our website to best meet our needs. The current AstroDB Toolkit was built based on lessons learned with this implementation. The Toolkit is still growing as we aim for a more generalized infrastructure. Our goal for this proposal is to work on these details, focusing on improvements to allow other users to straightforwardly develop their own databases.

Core Deliverables

To showcase the flexibility of the AstroDB Toolkit to the community and to expand the scope of the tool, we plan to use it to build three new databases. As part of the proposed project, we will focus on preparing the schema, or data model, behind these databases. These new databases will guide the growth of the template database and the AstroDB Toolkit itself.

- The first database example would focus on Dwarf Galaxies, in partnership with Collaborator #2. There are tens if not hundreds of galaxies in the Local Group, with a diverse range of applicable science problems. Many of these scientific applications depend on having as complete as possible a list of properties for these galaxies such as: total luminosity, half-light radius, and redshift/systemic velocity.

- The second database is one for Members of Tidal Streams, in partnership with Collaborator #3 and Collaborator #4. Stellar streams are disrupted dwarf galaxies and star clusters that are tidally stretched into thin, dynamically cold structures. We now know of over 100 streams orbiting around the Milky Way, which were discovered with a mix of different methods that rely on photometry, spectroscopy, or astrometry, or combinations of these. The data for stellar stream member stars are therefore heterogeneous and often not archived or stored in a consistent way.

- The third database is the Exoplanet Atmospheres database, primarily developed by Co-I #2. The past decade has seen an explosion in the number of atmospheres observed with ground-based, high-resolution spectrographs, now with more than 100 observations. For the detected species, the

reported properties may include detection significance, Doppler shift, gas abundance, and asymmetric errors on all quantities. We seek to create a database that holds this highly nonuniform information, collected from a wide array of literature sources, to enable probes of population-level trends.

Throughout the proposed project period, we will work closely with Collaborators #2, #3, and #4 to best identify how to represent their data and what science goals they need to achieve. This experience will also naturally help us refine our template schema, documentation, and manuals to make it more straightforward for other teams to create their own databases using the AstroDB Toolkit. All of these databases will both serve and enhance data from multiple NASA missions.

Technical Description

The AstroDB Toolkit facilitates the creation of intermediate-scale databases focused on typical astronomy use-cases. AstroDB relies on an object-model that naturally translates to astronomical sources, that is to say, the core table in the database can represent objects such as brown dwarfs or galaxies, while supporting tables represent *properties* of that object. Because the AstroDB Toolkit's core Python package, `AstrodbKit`, is a wrapper around the database package `SQLAlchemy`, it supports a large range of database architectures, including SQLite, PostgreSQL, MySQL, etc, while using language that is familiar to astronomers. `AstrodbKit` also supports cone searches and can output results as `astropy.table` objects.

Collaborative Workflow and Testing with GitHub

One of the key design requirements for an AstroDB Toolkit-powered database is support for **collaborative editing of the holdings using a GitHub workflow**. As a result, the Toolkit creates databases that are fundamentally a set of plain text JSON files that describe each object. When users make changes to the properties of an object, such as adding a new spectrum or updating a value for a radial velocity, these changes are human-readable as a simple diff between two JSON files and can be reviewed via pull requests. This JSON document store architecture allows for a community to maintain a database, review changes as they come in, and use automated tools to validate the database. The `Astrodbkit` package has tools to readily transform data between the document store and relational database to facilitate managing local, private data as well as external applications, such as a hosted website.

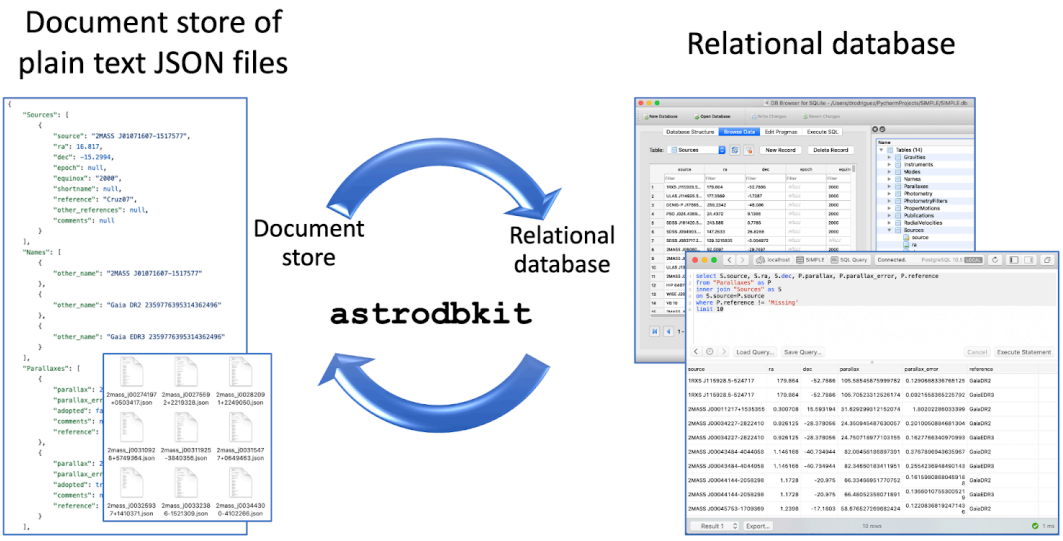


Figure 1: The `AstrodbKit` package readily transforms database contents to and from a JSON document store to a relational database, such as SQLite.

By exporting a database to a JSON document store, we can use git and GitHub to handle version control for our database as well as curate commits via pull requests.

An individual user may contain their own copy of any database. They may make changes in their local branch and push to their copy on GitHub. By issuing a pull request, they request their changes be adopted into the main branch of the database. Because the database is stored as individual JSON documents, reviewers can see exactly which objects have been updated and can comment on the changes if needed.

As part of the pull request process, automatic tests implemented via GitHub Actions can be run to verify the integrity of the database. This ensures no changes took place that break the functionality of the database and also include verification for the data that has been added.

Finally, when the pull request is accepted, additional automated tasks can be performed to regenerate the database and push it to external users of the database, such as a graphical user interface.

Spectra and non-tabular data

Spectra, images, and other non-tabular data are stored as pointers to cloud-hosted files. The files are hosted in a variety of places, including institutional repositories. However, we have found the best host to be Amazon Simple Storage Service (Amazon S3). In order to enable the database to fully function without an internet connection and/or to point to files not hosted in the cloud, we allow pointers to local files using an environment variable to indicate a local path.

AstrodbKit contains logic to understand which columns contain pointers to spectra and translates them into `specutils.Spectrum1D` objects. We propose here to provide support for lightcurves which would be translated into `lightkurve.lightcurve` objects. (`Lightkurve` is a NASA-package maintained by the TESS team at GSFC.)

Project Management

Governance

The AstroDB Toolkit project is quite small (<5 active developers) and does not yet require a formal governance structure. The current model is best described as a benevolent dictatorship with the PI currently serving as the "benevolent dictator for life" (BDFL). Weekly telecons, GitHub issues, and a Slack channel are all used to facilitate discussions and reach consensus.

Accessibility and Licenses

The AstroDB Toolkit project is being developed completely in the open. All repositories are on GitHub and licensed with a BSD-3 Clause license.

We have a dedicated Slack channel in the Astropy Slack workspace where interested developers and users are welcome to join. Participation and presentations at conferences are crucial for expanding the accessibility of the project to potentially interested developers and users.

Project Timeline

	Winter 2024	**Spring 2025**	**Summer 2025**	**Fall 2025**
New Database Designs	Gather requirements and understand the needs of users for the three new databases. Identify initial datasets for each new database.	Work iteratively with stakeholders and initial datasets to design initial conceptual schemas, including entities, attributes, and relationships.	Refine the conceptual schema based on feedback, further analysis, and additional datasets. Ingest sample datasets, and write tests.	Document the schema thoroughly, including data definitions, relationships, and constraints. Develop future plans for each database.
AstroDB Toolkit	Travel to Winter AAS to promote SIMPLE and the Toolkit and learn about user needs.	Expand functionality of the Toolkit to include lightcurves and images.	Add appropriate components to the Toolkit's template schema based on new databases.	Travel to ADASS to network with other technical, database, and data archive experts