Introduction to DSI

Project Lead: James Ahrens

Project Team Members: Divya Banesh, Hugh Greenberg, Pascal Grosset, Vedant Iyer, Jesus Pulido, David Schodt, Benjamin Sims, Terece Turton

LA-UR-25-29252

Presented by Jesus Pulido
pulido@lanl.gov
December 2025

Today’s agenda

- What DSI is and why should you use it
- Features
- User interface
- Examples and tutorial

What is DSI?

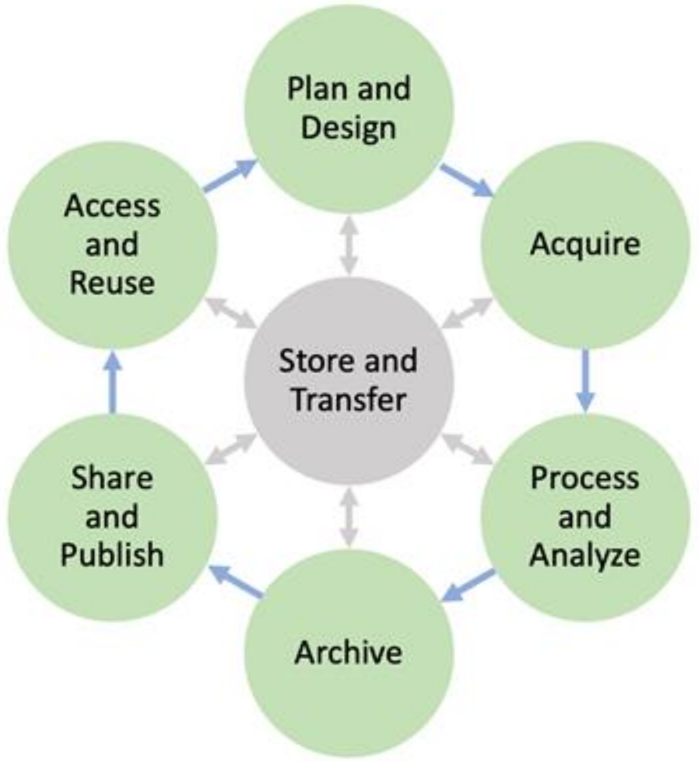
- A collection of simple and powerful tools for describing, storing, managing, moving, and querying your data and metadata seamlessly across locations and environments
  - Designed to scale and integrate with *data-intensive HPC simulation workflows*
  - Uses a *relational data model* to enable users to create, store and associate user defined *metadata and data* for later querying and processing (metadata = descriptive data about data)
  - Provides both command line and Python API interfaces
- Two ways we talk about DSI
  - An open source software package that can be freely downloaded and customized for many use cases (https://github.com/lanl/dsi)
  - Local instances of that software package with associated storage resources configured and available for use in LANL's collaborative (CE), yellow/restricted enclave (RE) and red networks

Why is DSI needed?

- Existing HPC data resources are not keeping up with evolving workflows
    - Data stored in files in hierarchical file systems with little metadata; difficult to share and maintain for future use
    - Users manually manage data movement across multiple storage resources with different and unpredictable deletion policies
    - Data storage and access commands are not uniform and change frequently
    - Increasingly complex and diverse data workflows (ensembles, UQ, AI/ML, etc.)
- ***DSI provides an abstraction layer that frees users to focus on the content of their data, not the details of how and where it is stored***
    - Supports interoperability and portability of data
    - Supports automation
    - Metadata and data access secured via POSIX group permissions

High-Level Goal: Support the Research Data Lifecycle

- Enable flexible, data-intensive scientific workflows that meet researcher and program needs
- **Support next-generation AI/ML-enabled data science workflows**
- **Facilitate seamless transitions from data-intensive/AI/ML research activity to long-term archiving and shared data repositories**
- Integrate with future institutional data management and preservation needs



Some basic data/DSI concepts

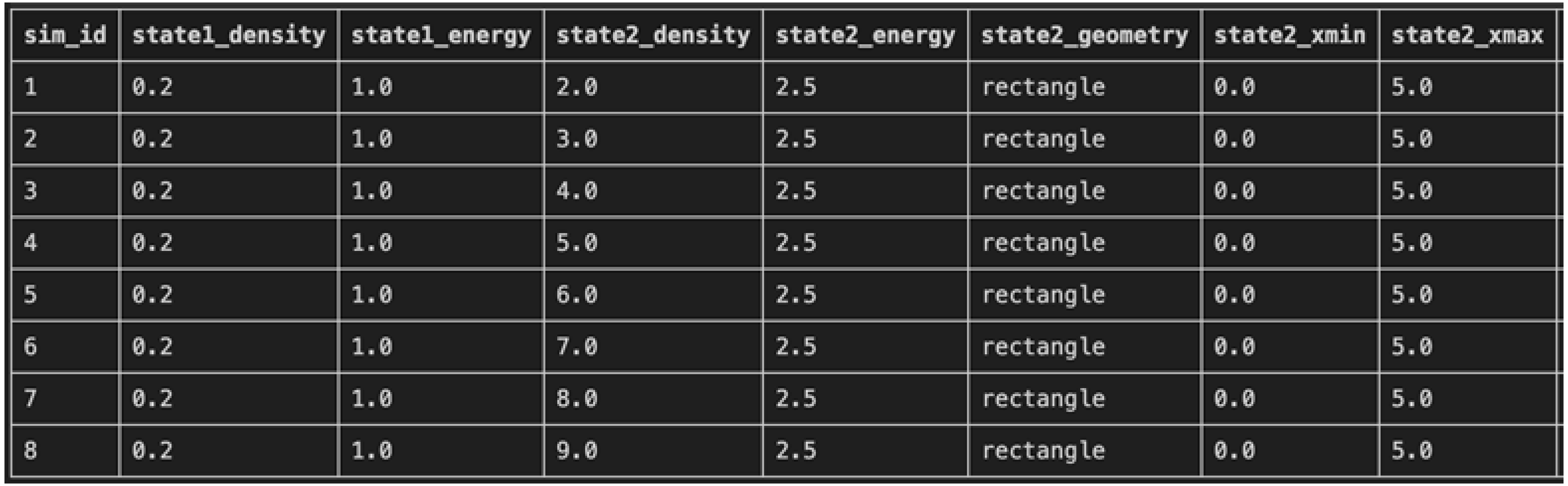
- **Data:** A grouping of values (numbers, strings, booleans, etc.)
- **Data set:** An organized body of data, usually with associated labels/column headings
  - E.g. the output of a series of runs of a particular model
- **Metadata:** Data that describes other data
  - Basic: Data labels/headings, info encoded in file names, etc.
  - Advanced: Creator, source, simulation/input deck parameters, environment parameters, annotations, etc.
  - Custom: Any parameter you want to use to organize your data!
- **Schema:** A standardized set of metadata & relationships defined for a particular use case
  - Specific term to relational database languages
  - Simple schema: Table and column headings (DSI-specific term)
  - Complex schema: Additional types of relationships between tables (DSI-specific term)
- **Collection:** Data object returned from a DSI action/query (DSI-specific term)
  - Implemented internally as a Pandas dataframe
- **Ensemble:** Data organized for parameter space analysis of multiple experimental/simulation runs (scientific analysis concept supported by DSI)

Data curation and metadata tiers

- Tier 1 (Descriptive) Metadata
  - User driven/captured metadata
  - Can be organized as a Data Card, i.e. Structured data description standards for consistency, discovery, and sharing
  - Used for database of databases/catalog of catalogs
- Tier 2 (”AI-Ready”) Metadata
  - Ensemble metadata (automatic)
  - Think of automatically parsing input decks, a design file, output files, etc.
  - Scripts can be used to create a DSI ‘Reader’, or to convert these files into an intermediate format such as CSV
- Tier 3 (System) Metadata
  - Data Governance metadata
  - Internal File Pointers from Ensemble **‘N’** to files related to that Ensemble **‘N’**
  - Filesystem metadata

Data curation and schema design

- We want users to start thinking about organizing their data
- Schema
  - Input decks / Design file

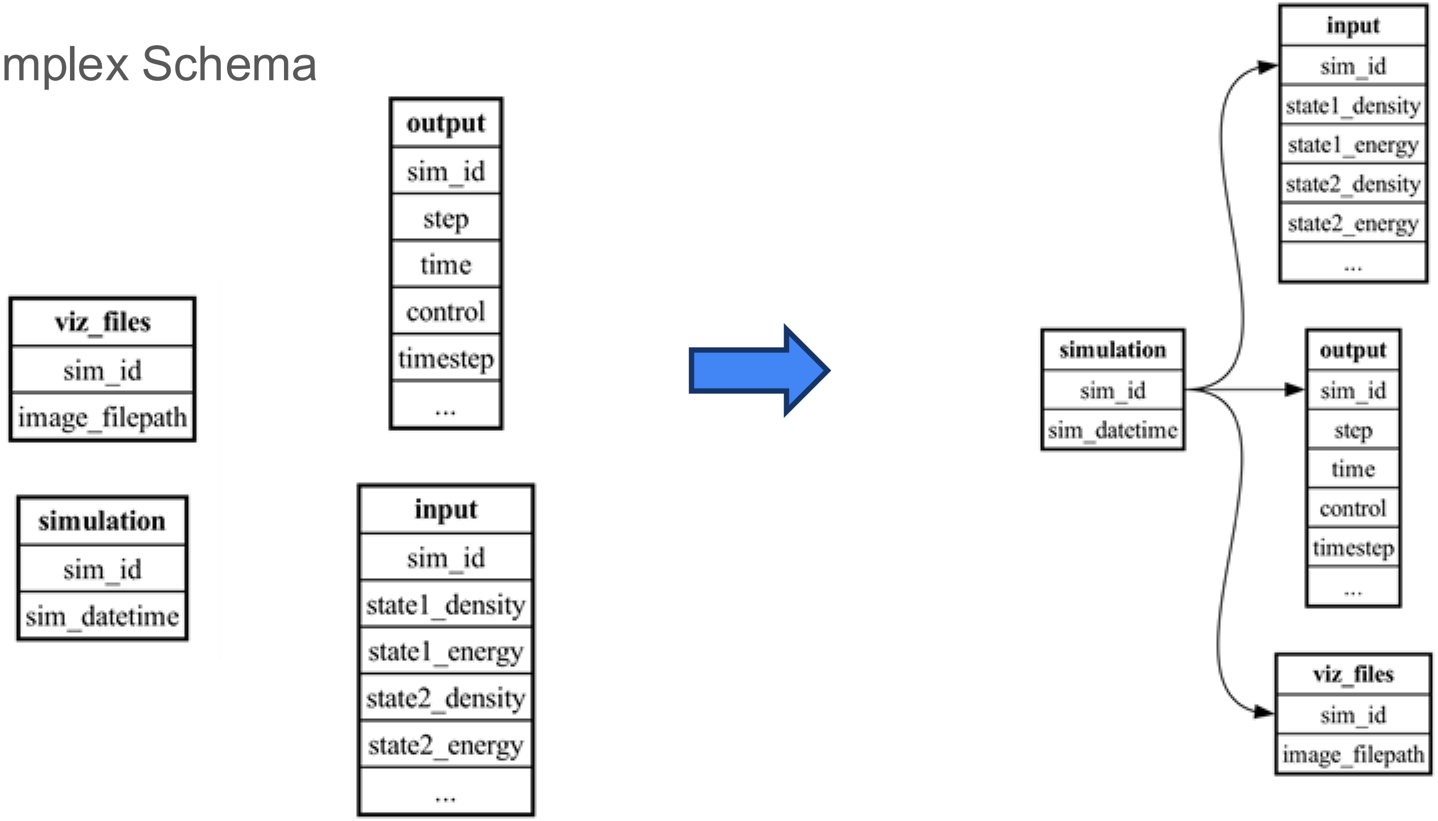


sim_id	state1_density	state1_energy	state2_density	state2_energy	state2_geometry	state2_xmin	state2_xmax
1	0.2	1.0	2.0	2.5	rectangle	0.0	5.0
2	0.2	1.0	3.0	2.5	rectangle	0.0	5.0
3	0.2	1.0	4.0	2.5	rectangle	0.0	5.0
4	0.2	1.0	5.0	2.5	rectangle	0.0	5.0
5	0.2	1.0	6.0	2.5	rectangle	0.0	5.0
6	0.2	1.0	7.0	2.5	rectangle	0.0	5.0
7	0.2	1.0	8.0	2.5	rectangle	0.0	5.0
8	0.2	1.0	9.0	2.5	rectangle	0.0	5.0

- For Ensembles, think of relating the ‘sim_id’ to the other metadata entries

Data curation and schema design

- Complex Schema



How does DSI work?

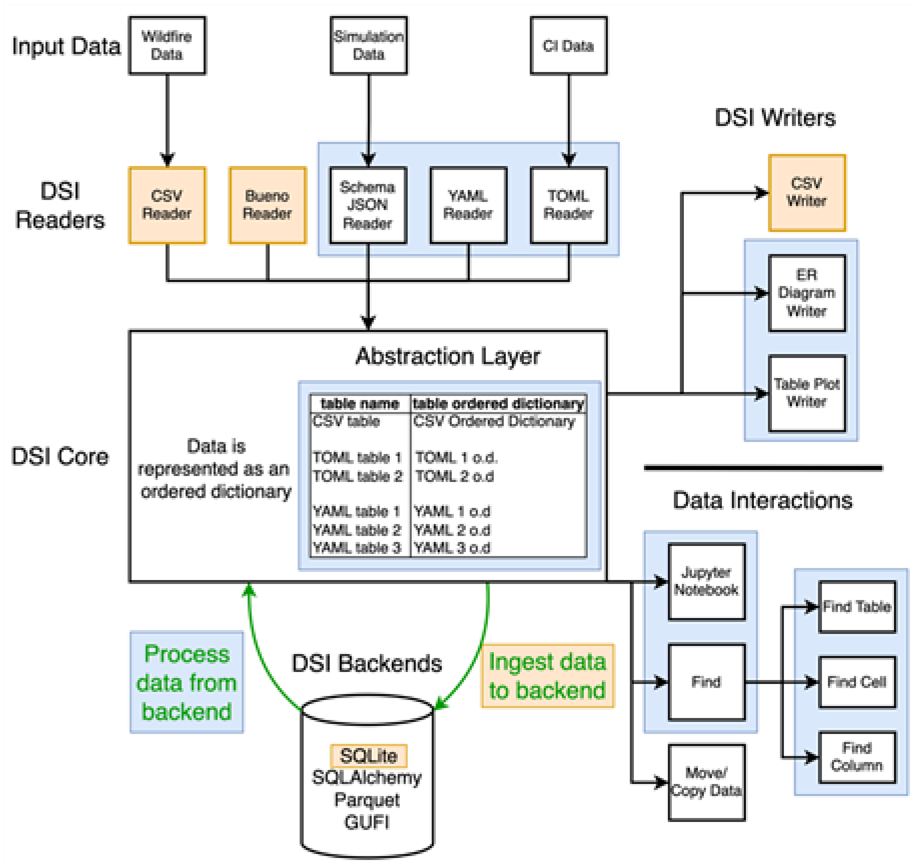
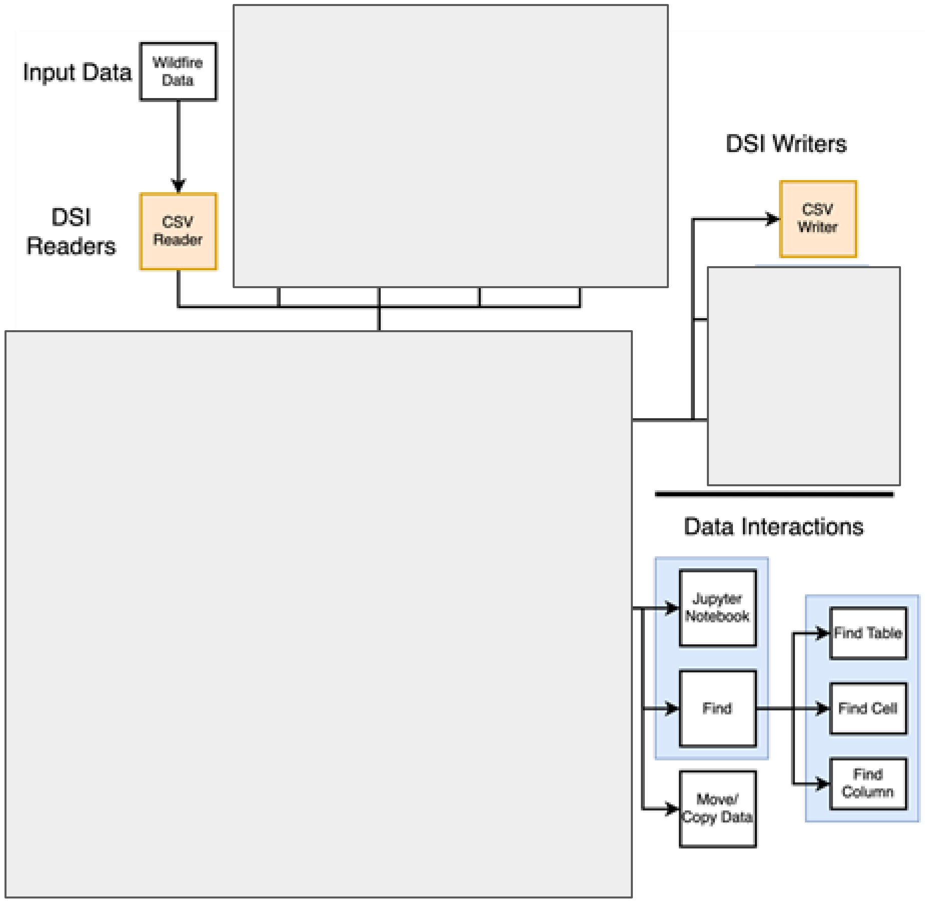


table name	table ordered dictionary
CSV table	CSV Ordered Dictionary
TOML table 1	TOML 1 o.d.
TOML table 2	TOML 2 o.d
YAML table 1	YAML 1 o.d
YAML table 2	YAML 2 o.d
YAML table 3	YAML 3 o.d

How does DSI work?



DSI Features

- Readers / Writers
  - Ingest many formats (csv, json, toml, yaml, ensemble, python collections)
- Data Storage Abstraction
  - Data can exist ‘anywhere’, let DSI handle the copy/move
- Find / Query / Update capabilities
  - No need to know backend languages like SQL
- Complex Schemas
- Data Cards
  - Dublin Core, Schema.org <Dataset>, Google’s “The Data Cards Playbook”, etc
- Viewers and Export
  - CLI, Cinema, pyCinema, Jupyter Notebook, Scikit-learn

DSI user interface: Python API

Python API

Data Science Infrastructure (DSI) Application Program Interface (API)	
DSI Action	Description
Read	Metadata and data from different data sources
Search/Find	Based on metadata search, can return as collection
Query	Metadata search using SQL, can return as a collection
Display / Summary	Summarizes statistics of collection metadata
Update	Commits new changes to the backend
Move / Get	Collections between file storage types
Write	For processing into specific formats

DSI user interface: CLI API

CLI API

Data Science Infrastructure (DSI) Application Program Interface (API)	
DSI Action	Description
Read	Metadata and data from different data sources
Search/Find	Based on metadata search, can return as collection
Query	Metadata search using SQL, can return as a collection
Display / Summary	Gives an overview on metadata read
Move / Get	Collections between file storage types
Write	Saving data in DSI to a permanent store
Draw / Plot Table	Exporting data into specific formats

DSI Resources

Open Github codebase

https://github.com/lanl/dsi

Documentation & Setup

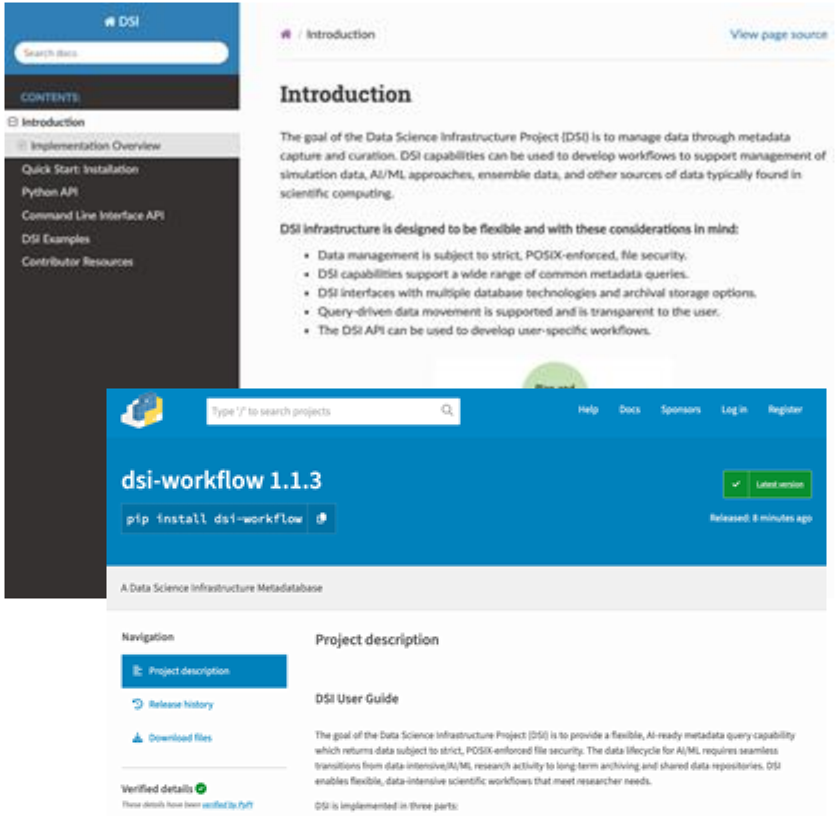
https://lanl.github.io/dsi/

PyPI

https://pypi.org/project/dsi-workflow/ (*)

Questions?

dsi-help@lanl.gov



Introduction

The goal of the Data Science Infrastructure Project (DSI) is to manage data through metadata capture and curation. DSI capabilities can be used to develop workflows to support management of simulation data, AI/ML approaches, ensemble data, and other sources of data typically found in scientific computing.

DSI infrastructure is designed to be flexible and with these considerations in mind:

- Data management is subject to strict, POSIX-enforced, file security.
- DSI capabilities support a wide range of common metadata queries.
- DSI interfaces with multiple database technologies and archival storage options.
- Query-driven data movement is supported and is transparent to the user.
- The DSI API can be used to develop user-specific workflows.

dsi-workflow 1.1.3

```
pip install dsi-workflow
```

A Data Science Infrastructure Metadatabase

Project description

DSI User Guide

The goal of the Data Science Infrastructure Project (DSI) is to provide a flexible, AI-ready metadata query capability which returns data subject to strict, POSIX-enforced file security. The data lifecycle for AI/ML requires seamless transitions from data intensive/AI/ML research activity to long-term archiving and shared data repositories. DSI enables flexible, data-intensive scientific workflows that meet researcher needs.

DSI is implemented in three parts:

*Variants of 'DSI' are taken on pypi, DSI is not specifically a 'workflow' tool

DSI Setup

1. Release Installation (pip)

```
python3 -m venv dsienv
source dsienv/bin/activate
pip install --upgrade pip
pip install dsi-workflow
```

1. Alpha Installation (git)

```
python3 -m venv dsienv
source dsienv/bin/activate
pip install --upgrade pip
git clone https://github.com/lanl/dsi.git
cd dsi
pip install .
```

DSI Setup

1. Release Installation (pip)

```
python3 -m venv dsienv
source dsienv/bin/activate
pip install --upgrade pip
pip install dsi-workflow
```

1. Alpha Installation (git)

```
python3 -m venv dsienv
source dsienv/bin/activate
pip install --upgrade pip
git clone https://github.com/lanl/dsi.git
cd dsi
pip install .
```

1. Load Module (LANL)
   WIP

```
module load dsi/1.2
```

DSI Tutorial Demo

- DSI Command Line Interface (CLI) API
- DSI Python API

DSI Unix CLI API

```
pulido@macbook dsi % cd examples/wildfire
pulido@macbook wildfire % dsi
DSI version 1.1.3

Enter "help" for usage hints.

dsi> read wildfiredata.csv
Loaded wildfiredata.csv into the table wildfiredata
Database now has 1 table

dsi> summary

Table: wildfiredata

column                       | type    | min    | max    | avg                 | std_dev
---------------------------------------------------------------------------------------------------
wind_speed                   | INTEGER | 2      | 12     | 6.000529380624669  | 3.6340690917175893
wdir                         | INTEGER | 175    | 270    | 219.9814716781366  | 34.640820171954076
smois                        | FLOAT   | 0.05   | 0.5    | 0.2709502382212822 | 0.1660589633626337
fuels                        | VARCHAR | None   | None   | None               | None
ignition                     | VARCHAR | None   | None   | None               | None
safe_unsafe_ignition_pattern | VARCHAR | None   | None   | None               | None
```

DSI CLI API

- Demo

DSI Python API Demo

Prerequisites:

Files for this tutorial: `git clone https://github.com/lanl/dsi.git`

- data and files for Jupyter Notebook:
  - `pip install jupyterlab`
  - `cd dsi/examples/`
  - Run: `jupyter lab`
    - Extra step on windows: `pip install notebook` then run: `python -m notebook`
  - Open dsi_diana_tutorial.ipynb inside your browser
- data and files for these slides:
  - cd dsi/examples/user/

DSI Python API

- Jupyter Notebook Demo

State of DSI

Current release. V1.1.3 (20251008)
Release Cadence - Quarterly

DSI Roadmap

1. Expanding remote backends (AWS*, DCStorage*, NSDS, Denodo, Granta, etc.)
2. LLM-assisted queries*
3. Training ML Models directly from data inside DSI*
4. Data Versioning*
5. Parallel Ingest (transactional database processing)
6. More Viewers

*dev branch

Questions?

dsi-help@lanl.gov