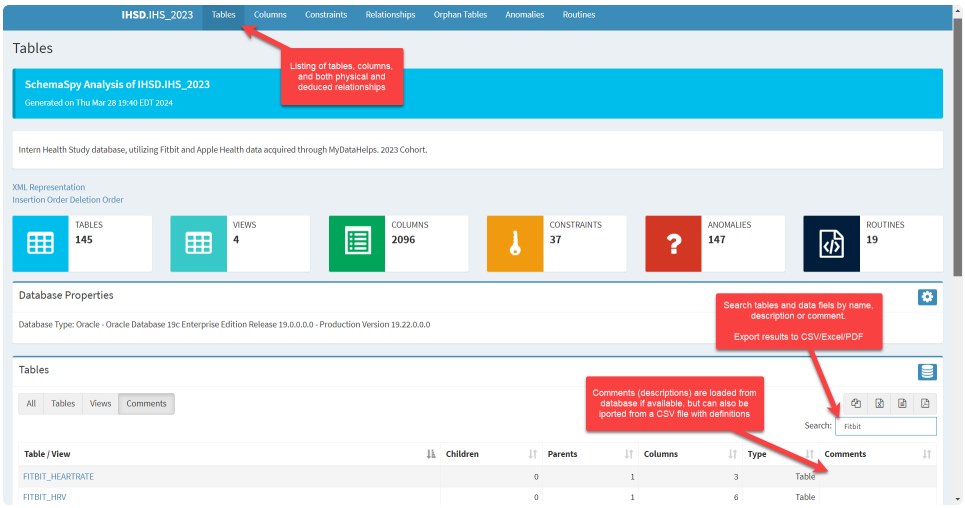
How to use SchemaSpy to document research datasets

Summary

Having a data dictionary - a document explaining the structure of data and its transformations - is crucial for study reproducibility and for meeting NIH data management & sharing requirements. However, creating data dictionaries can be time consuming and tedious work without the use of tools to automate as much of this as possible. This article shows how researchers can make use of an open source data dictionary tool called SchemaSpy to help create professional, easy-to-understand documentation for their datasets. Specific instructions are provided for Oracle, Microsoft SQL Server, CSV, SQLite, R, and Python + Pandas.



Example site: automation.depressioncenter.org/datadict

Steps

Video Tutorial

https://youtu.be/lxH5y187mCA

Getting SchemaSpy and Database Connectors

1. If using Windows, create the following directories:
   `c:\datadict`
   `c:\datadict\schemaspy`
   `c:\datadict\schemaspy\lib`
   `c:\datadict\html`
2. For Linux or Mac, create the same structure under your home directory (`~/datadict`)
3. Download the SchemaSpy Java JAR file from: https://schemaspy.org/
    1. Save schemaspy-<version>.jar to `datadict\schemaspy\schemaspy-<version>.jar`
4. Optionally, download and install the open source GraphViz library, which is needed for high-resolutio entity-relatioship diagrams. (You can also use the `-vizjs` parameter in the command line in later steps to use th built-in graphics library).
5. Ensure you have Java SE or Java SDK installed. Verify installation by opening a command prompt (shell) and typing:
   `java --version`
6. Download the appropriate database drivers (JDBC jar files) and save them to `datadict\schemaspy\lib`
    1. If connecting to SQLite in-memory databases, download the sqlite-xerial JDBC (***sqlite-jdbc-*.jar***)
    2. If connecting to Oracle databases, either download the Oracle JDBC package or install the Oracle Client for your operating system. Be sure to note where the files are extracted, and that they match your system architecture (32-bit or 64-bit).
        1. A common location for Oracle Client drivers in Windows is: *C:\Oracle\Product\19\client64\jdbc\lib*
        2. If you download the JDBC driver, extract all ****.jar*** files to `datadict\schemaspy\lib` .
    3. If connecting to Microsoft SQL Server databases, download and install the JDBC Driver for SQL Server.
        1. Make note of the installation path, and ensure the lib directory contains both ***mssql-jdbc-*.jar*** and ***mssql-jdbc_auth*.dll*** (for SSO and Integrated Security). Optionally, copy those files to `datadict\scheaspy\lib` .

4. If connecting to R datasets (.rds, .rdata, .rda), CSV files, or Pandas datasets:
    1. Download and install the Python csvs-to-sqlite package (`pip install csvs-to-sqlite`)
    2. Follow the instructions for downloading the SQLite driver in 6.1
7. Optional, advanced configuration: if connecting to other database types, or to create your own connectors (e.g. SharePoint Online), download the appropriate JDBC files and refer to the SchemaSpy documentation (see "Create your own db type - super advanced").
8. Additonal requirements:
    1. You must have an account with schema-level, admin-level or dbo-level access to the database
9. Once everything has been installed, verify that your driver is available and ready to use by SchemaSpy:
   ```
   java -Djava.library.path=c:\datadict\schemaspy\lib -jar "c:\datadict\schemaspy\schemaspy-6.2.4.jar" -dbhelp
   ```

Generate database documentation

Use the following examples to generate a data dictionary for your dataset. The output will be placed under `datadict\html` .

Oracle Databases

Edit and run this command:

```
 java -jar "C:\datadict\schemaspy\schemaspy-6.2.4.jar" -dp "C:\Oracle\Product\19\client64\jdbc\lib\ojdbc10.jar" -t ora -desc "<database description>" -o
"c:\datadict\html\<database>" -ahic -norows  -lq  -nologo -db <databse> -s <schema> -cat <schema> -host <server_hostname> -port 1521 -u <username> -p "<password>"
```

Microsoft SQL Server Databases

Create a custom properties file under **`datadict\schemaspy\mssql-custom.properties`** with the following contents:

```
 extends=mssql08
connectionSpec=jdbc:sqlserver://<hostOptionalPort>;databaseName=<db>;integratedSecurity=true;
```

Edit and run this command:

```
 java -Djava.library.path=c:\datadict\schemaspy\lib -jar "C:\datadict\schemaspy\schemaspy-6.2.4.jar" -dp "C:\datadict\schemaspy\lib\mssql-jdbc-11.2.3.jre11.jar" -t
mssql-ph -desc "<database description>" -o "c:\datadict\html\<database>" -ahic -norows  -lq  -nologo -db <database> -s "%" -cat <schema> -host <server_hostname> -port
1433 -connprops integratedSecurity\=true -sso -configFile "C:\datadict\schemaspy\mssql-custom.properties" -u <active_directory_username>
```

R or Pandas Datasets

1. The first option is to use an R or Python library to first "infer" the data types of your dataset (e.g. Pandas.DataFrame.infer_objects()), and then export directly to a SQLite database (c:\datadict\<dataset_name>.db)
2. The second option is to export your datasets to CSV, save them to `c:\datadict\<dataset_name>\csv` , and use the Python ***csvs-to-sqlite*** package to convert those CSVs to a SQLite database while inferring column data types automatically:
   ```
   pip install csvs-to-sqlite
   csvs-to-sqlite c:\datadict\<dataset_name>\csv c:\datadict\<dataset_name>.db
   ```
3. Once the datasets are in SQLite (.db file), edit and run this command to run SchemaSpy against it:

```
 java -Djava.library.path=c:\datadict\schemaspy\lib -jar "C:\datadict\schemaspy\schemaspy-6.2.4.jar" -dp "C:\datadict\schemaspy\lib\sqlite-jdbc-3.45.2.0.jar" -t
sqlite-xerial -desc "<dataset description>" -o "c:\datadict\html\<dataset_description>" -ahic  -norows  -lq  -nologo -db "c:\datadict\<dataset_name>.db" -all -sso
```

SQLite

Edit and run this command:

```
 java -Djava.library.path=c:\datadict\schemaspy\lib -jar "C:\datadict\schemaspy\schemaspy-6.2.4.jar" -dp "C:\datadict\schemaspy\lib\sqlite-jdbc-3.45.2.0.jar" -t
sqlite-xerial -desc "<database description>" -o "c:\datadict\html\<database>" -ahic  -norows  -lq  -nologo -db "c:\datadict\<database>.db" -all -sso
```

CSV

Follow steps 2 and 3 under R / Pandas.

Document Tables & Columns

1. Describing each table and column in your database allows others to understand what the data means and how to use it. This information is sometimes called "metadata." It is very important to provide a short description of each column, and if possible, the formulas used for calculating values or the steps taken to transform the underlying data. This allows future researchers to more easily validate or reproduce your study.



2. While SchemaSpy does not have the ability to edit table and column descriptions online, it does provide other options:
    1. ***Document table and columns at the database level*** (which may require working with a database administrator). SchemaSpy is able to read database comments and will display them on the documentation. This also allows anyone with direct access to the database to see definitions in their SQL development tools.

    2. Alternatively, ***create a "schema metadata" file*** and pass it in the commands in the previous section. Refer to the documentation on how to create such a file, and use it in the previous commands by adding `-meta <path_to_meta_file.xml>` .



Publish the documentation

1. The output of the previous commands will consist of one directory per database/schema, containing several HTML, CSS and JavaScript files, with an index.html as the entry point (home page) of each schema.
2. The files must be published to a web server. Due to the relative paths and file names used, they will not work if published to OneDrive, Dropbox, Google Drive or SharePoint.
3. U-M faculty, students and staff are eligible for free web hosting through ITS. This requries setting up an AFS file share to store the files, and a URL that wil resolve to it.
   See https://its.umich.edu/computing/web-mobile/web-hosting/afs-websites or https://its.umich.edu/computing/web-mobile/web-hosting/virtual-hosting for more information.
4. Once the web server is up and running, simply copy the contents of `datadict\html` to your public HTML folder (e.g. `~\www\html`)
    1. For example, we published to our AFS file share at: /afs/umich.edu/group/e/efdcautomation/Public/html/datadict/
    2. This gives us access via the following test web server URL: https://automation.depressioncenter.org/datadict/
5. It is recommended that you create an ***index.html*** file that links to each individual schema. For an example, see the code used in: https://automation.depressioncenter.org/datadict**/index.html**

```
<html>
  <head>
    <title>Data Dictionaries</title>
  </head>
  <body>
    <h1>Data Dictionaries</h1>
    <p>Listing of data dictionaries generated by SchemaSpy.</p>
    <ul>
      <li><a href="IHS_2023/index.html">IHS Study (2023 Cohort)</a></li>
      <li><a href="PROMPT/index.html">PROMPT Study</a></li>
    </ul>
  </body>
</html>
```





Notes

- Although SchemaSpy can be helpful for documenting datasets, we recommend using an enterprise-grade data dictionary tool whenever possible. This type of application normally requires institutional buy-in as part of a larger data governance and data stewardship program.

Resources

- SchemaSpy website and documentation.
- Code repository - DepressionCenter/MTC-Internal-Tools-and-Automation
- Pandas dataframe export documentation.
- CSV to SQL Lite converter - simonw/csvs-to-sqlite
- U-M Web Hosting Options Offered by ITS - https://its.umich.edu/computing/web-mobile/web-hosting
- National Institutes of Health - Final NIH Policy for Data Management and Sharing

About the Author



Gabriel Mongefranco is a Mobile Data Architect at the University of Michigan Eisenberg Family Depression Center. *Gabriel* has over a decade of experience in data analytics, dashboard design, automation, back end software development, database design, middleware and API architecture, and technical writing.

Details

Details

Article ID: 11889
Status: Approved
Revision Number: 9
Draft Status: None
Published: Yes **Public:** Yes

Owner
Gabriel Mongefranco

Created
Tue 4/2/24 10:02 AM by Gabriel Mongefranco

Modified
Thu 8/21/25 1:22 PM by Gabriel Mongefranco

Next Review Date
4/4/2025

Author(s)
Gabriel Mongefranco

Code Repository
https://michmed.org/Z7eRK