

# Database Administration

## Lab 01: Metadata Tools

Andrés Calderón, Ph.D.

February 2, 2026

## 1 Introduction

The main objective of this lab is to explore a database tool for automatically generating a Data Dictionary. [SchemaSpy](#) is a multiplatform DBA tool designed for various DBMS. It is a CLI program that connects to a database and extracts useful information about tables, attributes, and relationships. We will cover a basic guide and then explore other similar tools as independent work.

## 2 SchemaSpy Tutorial

1. First, we will download SchemaSpy v6.2.4 from [here](#). SchemaSpy is a Java application, so ensure that you have a stable JRE installed on your system. If Java is not present or if you want to update or change the current installation, follow the steps in [this](#) tutorial.
2. To perform this lab, we need a dummy database to analyze, so we will use the Motor Race Database (MRD), available [here](#), which contains data about Formula 1 races. An Entity-Relationship Diagram can be seen in Figure 1, and a PostgreSQL dump file of the database can be downloaded [here](#).

To upload the sample database, create a PostgreSQL database named `mrd` in a console while logged in as your Linux user:

```
$ createdb mrd
```

Connect to the sample database using the following command:

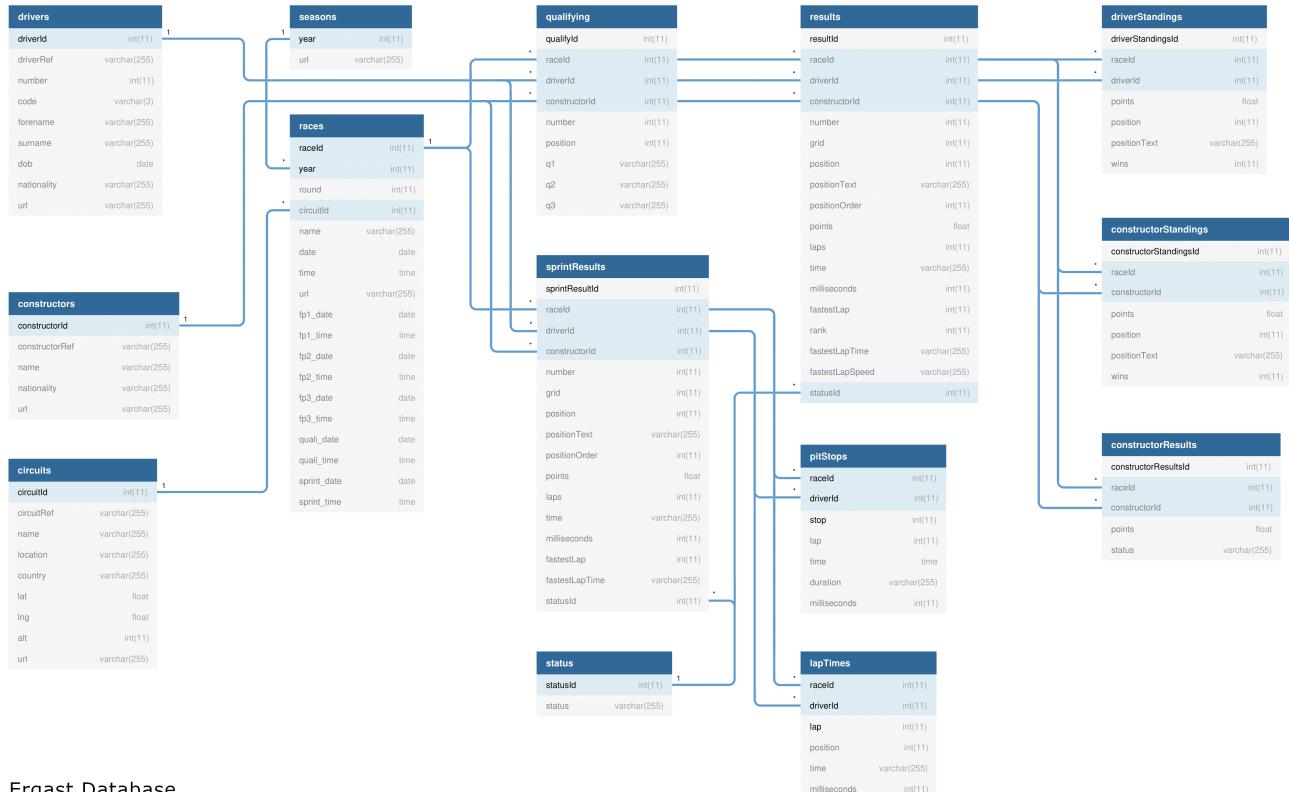
```
$ psql mrd
```

From the PostgreSQL prompt (once you are in the PostgreSQL environment), upload the dump file using the following command: `\i path_to_dump_file`

```
mrd=# \i dump_mrd.sql
```

You may need to adjust the full path of the file as necessary.

3. To connect to a particular DBMS, we need a Java DataBase Connector (JDBC) for that specific DBMS. Since we are working with PostgreSQL, we need to download the official PostgreSQL JDBC driver from <https://jdbc.postgresql.org/>.  
At the time of writing, the current version is 42.7.7, so you should download a file named `postgresql-42.7.7.jar`. Please place the driver in the same location as the SchemaSpy application.
4. Next, lets create a folder called `dd` in the same directory as `schemaspy` and the JDBC driver to store the results.



Ergast Database

Figure 1: The E-R Diagram of the MRD.

```
$ mkdir dd
```

- Now, we can run the SchemaSpy tool to generate a Data Dictionary from the available data. To do so, type the following command in a terminal console from the folder where the `schemaspy` executable and the JDBC driver reside:

```
$ java -jar schemaspy-6.2.4.jar -t pgsql11 -dp postgresql-42.7.7.jar -db mrd \
-host localhost -port 5432 -s f1db -u your_username -vizjs -o dd/
```

Remember that the \symbol represents a carriage return or line break (ENTER), allowing you to type the command across multiple lines. If you prefer, you can enter the full command on a single line, but in that case, remove the \.

The arguments that SchemaSpy receives are:

- t**: Specifies the type of DBMS to connect to. In our case, we use `pgsql11` because we are connecting to a PostgreSQL version greater than 11.0. It derives from `t=type`.
- dp**: Specifies the path to the JDBC driver for the target DBMS. It derives from `dp=driver path`.
- db**: Specifies the name of the database. It derives from `db=database`.
- host**: Specifies the host where the database resides.
- port**: Specifies the port where the host listens for connections.

Figure 2: Output of `schemaspy` over the MRD (in our case the PostgreSQL JDBC version should be 4.7.7).

- **-s**: Specifies the schema of the database. It derives from `s=schema`.
  - **-u**: Specifies the database user. It derives from `u=user`.
  - **-vizjs**: Specifies the library used for graphical representation of relationships. It derives from `Viz.js`,<sup>1</sup> a JavaScript implementation of the powerful DOT language and the GraphViz tool.
  - **-o**: Specifies the output path where the Data Dictionary will be stored. It derives from `o=output`.

If everything goes well, you will see an output similar to that in Figure 2. This process will generate a file named `index.html` in the `dd/` folder, which contains a webpage displaying the Data Dictionary for the MRD tables.

Go ahead and check it out [here](#)!

### 3 What is next?

We expect two things from you and your team as independent work:

- As you can see in the *Relationships* tab of the SchemaSpy Data Dictionary (DD) report for MRD, the relationships in the current database are implied. As stated in the report:

---

<sup>1</sup>Visit <https://viz-js.com/> and <https://graphviz.org/> for more info.

*“No “real” Foreign Key relationships were detected in the schema. Displayed relationships are implied by a column’s name/type/size matching another table’s primary key.”*

Your first task is to fix this issue by creating the appropriate integrity reference constraints for MRD. Specifically, all foreign keys must be defined according to the Entity-Relationship (E-R) Diagram shown in Figure 1.

2. You will read the following [document](#) on metadata tools overview. It contains information about additional commercial and open-source tools that support metadata creation.

Your task is to select one of these tools and create a similar tutorial, following the approach used with SchemaSpy, while analyzing the MRD data source. This should be done after updating the integrity reference constraints as required in the first task.

We expect you to submit a well-structured report in PDF format, along with a dump file (a plain text with .sql extension) of the fixed MRD, packaged in a ZIP file, by **February 16, 2025 before 2:00PM**, via the link that will be provided on Brightspace.

Happy Hacking 😊!