Creating Databases and Tables with SQL

In the previous reading, you learned how to use the Table Browser in Hue to create databases and tables through point-and-click actions. This method is sometimes convenient, but it's often better to use SQL commands to create databases and tables.

The SQL statements **CREATE DATABASE** and **CREATE TABLE** provide a more systematic way to create databases and tables. These commands give you greater power and flexibility, and they can be used in other Hive and Impala interfaces besides Hue. Using these commands, you also have the option to script and automate the creation of databases and tables, making these tasks more reproducible.

As you go through this reading, or after you've read through it once, use the VM to create some test databases and tables. You can drop the databases and tables when you're done.

To perform the steps described in this reading, you will need to use Hue on the VM. If you do not already have the VM installed and running, please follow the instructions in the reading *Downloading and Installing the VM* in the first week of this course. Then open the web browser on the VM and click the link for Hue in the bookmarks toolbar.

Creating a Database

Creating the database is actually as simple as step 2 below. Step 1 is just to get you to the place where you can run that step, and step 3 is verification of success.

1. First, get to the Impala query editor in the VM by clicking the **Query** button. Since Hive and Impala share the Hive metastore, either will work, but using the Impala query editor is easier. (If you use Hive, Impala will not immediately acknowledge the new database. You will learn more about this later in this Week's lessons. For now, use Impala rather than Hive.)
2. Enter and execute the command: **CREATE DATABASE test;** *Note:* This will create a directory in HDFS in the default location: **/user/hive/warehouse/test.db**.
3. Verify the creation was successful using the data source panel to the left of the query editor panel. Look in the Impala or Hive databases; you should see six databases (**default**, **fly**, **fun**, **test**, **toy**, and **wax**). If **test** does not appear, try refreshing the display using the refresh button (two curved arrows) at the top of this left panel. If you like, you can check the HDFS file structure to see that **/user/hive/warehouse/test.db** exists.
4. To drop your test database, enter and execute the command: **DROP DATABASE test;**

Creating a Table

This is just a quick introduction to the **CREATE TABLE** statement, giving the most basic structure. The next lesson will break down the **CREATE TABLE** statement and cover each of the clauses in it. Don't worry about the details yet.

Step 3 below is the actual creation step, which will be the same for any tool you use to enter SQL commands. Steps 1 and 2 are to help you navigate to the appropriate area in Hue, and step 4 is verification of success.

1. Click the **Query** button to access the Impala query editor.
2. Check the active database and be sure it's the one you want for your new table. For testing, put it in the **default** database; if necessary, select **default** as the active database. (See Figure 1 below.)
3. Enter and execute this command: **CREATE TABLE test (col1 INT, col2 STRING);**The table name comes after **CREATE TABLE**, and then a list of the column names with their data types. The command below creates a table named **test** with two columns, an integer column named **col1**and a string column named **col2**. (See *Notes* below.)
4. Verify the creation was successful using the data source panel to the left of the query editor panel. If the selected database is not Impala's **default** database, navigate to it and check that your table **test** is listed. You might need to refresh the display by clicking the refresh button (the two curved arrows). If you like, you can check the HDFS file structure to see that **/user/hive/warehouse/test**

exists.

5. To drop the table, enter and execute the command: **DROP TABLE test;** The table directory *and any data it might have held* will also be deleted. After dropping the table, verify that **/user/hive/warehouse/test** no longer exists.

CREATE TABLE test (col1 INT, col2 STRING);

Figure 1

Notes:

- The **CREATE TABLE** statement in Step 3 creates a subdirectory in the **/user/hive/warehouse/**directory on HDFS named **test**. If you put this in a different database, such as the **fun** database, the **test** directory will be a subdirectory of that database's directory (**/user/hive/warehouse/fun.db/** for the **fun** database).
- You also can qualify the table name with the database name: **CREATE TABLE default.test ...**. This will put the table named **test** into the database **default**, regardless of which database is active. (Try it!) You might find this a preferable method and skip step 2.