The STORED AS Clause

The data for each table is stored in files. The **ROW FORMAT** clause specifies the delimiters between column values in those files, but there are different file formats you can use. (You will learn more about these file formats in Week 3 of this course.) The **STORED AS** clause in the **CREATE TABLE** statement allows you to specify what file format you want a new table to use. To create a table that uses existing data files, you need to match the format of the file.

The syntax for this clause is simply:

STORED AS *filetype*

For example, a table creation statement might look like this:

```
CREATE TABLE jobs (id INT, title STRING, salary INT, posted TIMESTAMP)
  STORED AS TEXTFILE;
```

The default file format is **TEXTFILE**—simple text format, which is human readable—so in this example, the **STORED AS** clause is optional. Without it, Hive or Impala would still use **TEXTFILE** for the file format. However, other file formats (which will often look like gibberish if you try to view the files directly) must be specified explicitly.

Try It!

On the VM:

1. Log in to Hue and go to the Impala query editor.

2. Do the following to create a table, fill it with one row of data, and look at the resulting file on HDFS:

a. Execute the following **CREATE TABLE** statement:

```
CREATE TABLE jobs_txt
     (id INT, title STRING, salary INT, posted TIMESTAMP)
  STORED AS TEXTFILE;
```

b. Load one row of data by executing the following statement.

```
INSERT INTO jobs_txt
  VALUES (1,'Data Analyst',135000,'2016-12-21 15:52:03');
```

c. Use the File Browser or the data source panel (choosing the files icon rather than the database icon) and find the **/user/hive/warehouse/jobs_txt** directory. If you don't see the **jobs_txt** subdirectory, refresh the display by clicking the refresh button (two curved arrows). Find a file with a name that's just a string of letters and numbers, and click that file.

d. You can see the contents of the file in the main panel. Notice that you can clearly see each of the values you added to the table.

3. Now create another table using a different format, and see that the resulting file looks different:

a. Execute the following **CREATE TABLE** statement, which configures the table to store data in **PARQUET** format:

```
CREATE TABLE jobs_parquet
   (id INT, title STRING, salary INT, posted TIMESTAMP)
  STORED AS PARQUET;
```

b. Load one row of data by executing the following statement.

```
INSERT INTO jobs_parquet
    VALUES (1,'Data Analyst',135000,'2016-12-21 15:52:03');
```

c. Use the File Browser or the data source panel (choosing the files icon rather than the database icon) and find the **/user/hive/warehouse/jobs_parquet** directory. If you don't see the **jobs_parquet** subdirectory, refresh the display by clicking the refresh button (two curved arrows). Find a file with a name that's just a string of letters and numbers, and click that file. You'll get an error message that says Hue can't read the file.

d. Open a Terminal window. (You can do this by clicking the icon in the menu bar that looks like a computer.) Enter and run the following command, which will show the contents of the Parquet file. (Do not include the **$**; that's the prompt to indicate this is a command-line shell command, not a query.) Notice that the output includes a lot of non-ASCII characters, so you can't really read most of it.

```
$ hdfs dfs -cat /user/hive/warehouse/jobs_parquet/*
```

4. Drop both tables (**jobs_txt** and **jobs_parquet**) since you won't need either again.

5. Now try creating a table using data from an existing Parquet file. When you're done, keep this table, because you'll use it again later. (If you use the **EXTERNAL** keyword as directed below, then dropping the table will not delete the data, so you could drop it now and come back and recreate the table later if you wish.)

a. A Parquet version of the **investors** data is also stored in HDFS, at **/user/hive/warehouse/investors_parquet** (which will be the default location for a table named **default.investors_parquet**). Examine the file in the same way as you examined the jobs Parquet file: In the Terminal window, issue the command **hdfs dfs -cat /user/hive/warehouse/investors_parquet/investors.parq**. Again you'll see it's not really in human-readable format.

b. Now create the table from the query editor:

```
CREATE EXTERNAL TABLE default.investors_parquet
     (name STRING, amount INT, share DECIMAL(4,3))
    STORED AS PARQUET;
```

c. Use the data source panel or run a **SELECT *** query to verify that the contents of the new table are correct. (It should look identical to the other **investors** table you created in “The ROW FORMAT Clause” reading.)