Missing Values

NOTE: Although this reading is in the Cloud Storage lesson, it applies to working with data in HDFS as well.

When working with character string columns, it is important to remember that an *empty string*, also called a *zero-length string*, is *not* the same thing as a *missing value* (NULL).

However, when data is stored in delimited text files by processes external to Hive and Impala, it is common for missing values to be represented as empty strings. For example, your organization might have a data collection process in which first and last names are stored in a CSV file, and some individuals might leave their last names blank. So the rows of the resulting CSV file might look like this: fname,lname

Bert,

Big,Bird

Count,von Count

Ernie,

Two-Headed,Monster

Notice the trailing commas in the rows where individuals did not provide a last name.

If you load this CSV file into the storage directory for a Hive and Impala table, the absent values in the lname column will be interpreted as empty strings (''), *not* as missing values (NULL).

This might cause some confusion. For example, an analyst might use Hive or Impala run a query on this table, such as

```
SELECT * FROM names WHERE lname IS NOT NULL;
```

This query would return *all* the rows. This might defy the analyst's expectation that it would only return the rows for Big Bird and Count von Count (because the other rows appear to have no last names). But Hive and Impala do not treat these empty strings as NULL by default.

However, there is a table property you can set to make Hive and Impala interpret empty strings in table data files as missing values: the serialization.null.format property.

You can set this property in a CREATE TABLE by using this TBLPROPERTIES clause:

```
TBLPROPERTIES ('serialization.null.format' = '')
```

Notice how the value of this property is set to ''(an empty or zero-length string).

You can also set this property on an existing table by using an ALTER TABLE statement. For example:

```
ALTER TABLE names SET TBLPROPERTIES ('serialization.null.format' = '');
```

If you set this table property to an empty string (''), then Hive and Impala queries on that table will treat empty strings in character string columns as if they are true missing values (NULL).

The default value of this serialization.null.format is '\N'. This is a two-character-long literal string consisting of a backslash followed by the capital letter N; it is not a special character.

The serialization.null.formatproperty also determines how NULL values are represented in the table data files when data is *inserted* into the table using Hive or Impala. It affects how missing values are stored for columns of *all* data types (not only character string types).

For a particular table, you should decide whether missing values should be represented as \N or as an empty string in the files for that table. You should never mix different representations in the files for a single table.

As a best practice, we recommend processing text-based data to ensure that the values you want interpreted as NULL by Hive and Impala are represented as \N in the data files *before* you store the files in the table directory. But in situations where this is not practical, the method described above is an effective workaround.

Try It!

The above information may make more sense when you see it in action, so try the following steps using Hue.

1. To see an example of a data file that contains this value \N, look at the contents of the file /user/hive/warehouse/offices/offices.txt in HDFS. This is the file containing the data in the offices table in the default database. Notice the \N in the fourth row and third column of this file, which represents a missing value of state_province for the office located in Singapore. Query the offices table with Hive or Impala, and notice that this value appears as NULL in the query result.

For the rest of this section, you will create a table and see the effect of setting the serialization.null.format property.

2. Using either Hive or Impala, create a table named names by running the following CREATE TABLE statement:

```
CREATE TABLE names (fname STRING, lname STRING)

    ROW FORMAT DELIMITED

        FIELDS TERMINATED BY ',';
```

3. In the table's directory in HDFS (/user/hive/warehouse/names/), create a text file and store the following five lines in it:

```
Bert,

Big,Bird

Count,von Count
```

Ernie,

Two-Headed,Monster

You can do this with Hue's File Browser or with an hdfs dfs command.

4. If you're using Impala, run the following command to force Impala to refresh its file metadata for this table:

   REFRESH names;

5. Run the following SQL query on this table:

   SELECT * FROM names WHERE lname IS NOT NULL;

   Notice how the result set includes *all* the rows, including the rows in which lname is empty.

6. Now run the following statement to tell Hive and Impala to treat empty strings as missing values for this table:

   ALTER TABLE names SET TBLPROPERTIES ('serialization.null.format' = '');

   If you're using Impala, refresh the metadata for this table again.

7. Run the SQL query again to show rows for which lname IS NOT NULL. This time, notice how the rows with empty last names are *not* returned in the result set.

8. Finally, drop this names table, because you won't need it again.