CS 145 Introduction to Databases

Stanford University, Fall 2016

Bulk-Loading Data into SQLite Databases

Bulk-loading refers to the process of loading data specified in lines of a file directly into a database system, rather than executing a large series of `INSERT` statements. We first describe the data file format for bulk-loading into SQLite databases, then the loading process.

The Data File

The data file consists of a sequence of lines, each one specifying one tuple to be loaded into an existing table of the database. Each line lists values for the attributes of the table, in the order the attributes were declared when the table was created. Any string can be used as the separator between attribute values; by default, it is `|`.

As an example, suppose we are bulk-loading into a table `Student(ID,name)`, where `ID` is an integer and name is a string. If we specify `|` as our separator, our data file (e.g. `students.dat`) might look like:

```
123|Alice
456|Bob
789| Carol
```

If we load `students.dat` into the database, then the following tuples will be inserted into table `Student`:

```
(123,'Alice')
(456,'Bob')
(789,' Carol')
```

Warning: Notice that the third line of `students.dat` has a blank after separator `|`. This blank is not ignored by the loader, i.e., string ` Carol` is loaded with a leading blank, as demonstrated by the third tuple above. It is a common (and frustrating!) mistake to leave blanks before or after separators, and then wonder why string values are not matching as you expect.

Loading

Bulk-loading can be done via the SQLite special command `.import`, as follows:

```
.separator <separator>
.import <loadFile> <tableName>
```

For example, if we want to load a file `name.dat` into table `Name`, where attributes are comma-separated, we would execute:

```
.separator ,
.import name.dat Name
```

Keep in mind that you should use an attribute separator string that will never appear in your data, so separators aren't confused with data fields. Note that primary keys and other constraints on the table are enforced during bulk-loading; if any tuples violate these constraints, the load command will fail.

Loading NULL Values

Unfortunately, there is no way to specify `null` values in a load file such that SQLite will actually load a `null` value. Thus, loading data with `null` values must be done in a two-step process: first, load the data with some stand-in value for `null` (e.g. `''NULL''`). Then, execute an `UPDATE` statement on the table to replace instances of `''NULL''` with `null`.

For example, consider a table called name `Name` with attributes `first_name`, `middle_name`, and `last_name`, and suppose we want to bulk-load the names `Gottfried Wilhelm Leibniz` and `Lady Gaga`. Our `name.dat` load file would look like:

```
Gottfried|Wilhelm|Leibniz
Lady|NULL|Gaga
```

where `|` is our attribute separator. We can properly set the `null` values when we load our data into the `Name` table by executing:

```
.separator |
.import name.dat Name
update Name set middle_name = null where middle_name = 'NULL';
```

Escaping Quotation Marks

During the import, you may receive an error similar to:

```
Error: file.dat line 1: expected 4 columns of data but found 3
```

This means that SQLite believes your load file contains a different number of columns than the table you are attempting to load it into. Unfortunately, even if you have the correct number of columns, you may still receive this error if your data contains unterminated quotation marks.

Consider this line from a load file for a table with the schema `Item(id, name, location, description)`:

```
123|29" bike rims|California|Bike rims that are 29"
```

As there are unterminated quotation marks, SQLite will throw an error during the import. To prevent this, you need to perform both of the following:

1. Escape every instance of a double quote with another double quote.
2. Surround all strings with double quotes.

For example, you'd want to convert the above line in your load file to:

```
123|"29"" bike rims"|"California"|"Bike rims that are 29"""
```