Chapter 6

Stage 3: Data Load

6.1 GUERRILLA ANALYTICS WORKFLOW

Figure 16 shows the Guerrilla Analytics workflow. Data Load involves getting raw data from its storage location in the data folder on the file system into a Data Manipulation Environment (DME) so that it can be analyzed. This must be done in a way that is flexible and can cope with the variety of data types and DMEs the team may need to use. This must also be done while preserving data provenance by maintaining some link between data in the file system and the data as it is loaded into the DME.

6.1.1 Example Activities

There are several types of Data Load scenarios that you may have encountered.

- **A Relational Database Management System (RDBMS) extract:** A RDBMS system's data have been extracted into as many as several hundred text files, where each text file is an export of a data table in the source system. These text files must be loaded into the DME and the load must be checked to be complete and correct.
- **An unstructured extract:** A file share of thousands of scanned letters in PDF format has been made available for analysis. There is an intended meaning to the folder and subfolder structure in which these files are stored. For example, there may be a file year subfolder and within each file year subfolder there is a file month subfolder. Such a scenario is illustrated in Figure 17. You must run through this folder structure and load all files into a NoSQL document database (Sadalage and Fowler, 2012) while combining the loaded files with the "data" encoded in the subfolder locations and file names.
- **Semistructured data:** A customer has given you a spreadsheet workbook that contains 10 worksheets. Each worksheet has color-coded rows, where color is a Red–Amber–Green style indicator of importance. Some worksheets have hidden rows and columns that should not be used in any analysis. Some worksheets have embedded images that cannot be loaded. You must load this spreadsheet into the DME so it can be manipulated and integrated with other data.

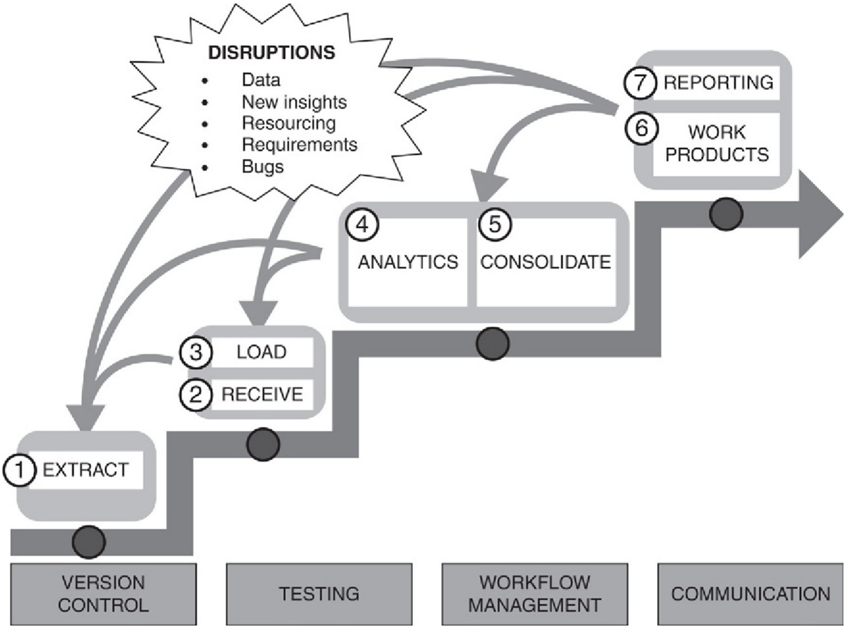

FIGURE 16 **The Guerrilla Analytics workflow**

- **Large files:** You have a single file of web log activity that is several gigabytes in size and is time-consuming to load. The file contains a corruption that is preventing a successful load. The size of the file means you cannot open it in a text editor to determine where the fault is located in the file so you can assess and repair it manually.

These examples are hopefully somewhat familiar to you. There are some common themes to these examples that illustrate the variety of challenges faced at Data Load in a Guerrilla Analytics environment.

- **Preparatory work:** Some amount of preparatory work may have to be done on the data so that it loads into the DME correctly.
- **Need to peek at files:** There will be files that you cannot open using conventional text editors because they are corrupt or too large to load into memory. A method is needed to deal with these scenarios.
- **Validating a Data Load:** There will be a requirement to verify that the data has loaded correctly from the file system into the DME.
- **Chunking:** A single file may have to be broken into several files before it is loaded (as with spreadsheets containing multiple sheets).

This variety of data and associated challenges is increasingly common and must be dealt with using a reproducible approach that preserves data provenance.

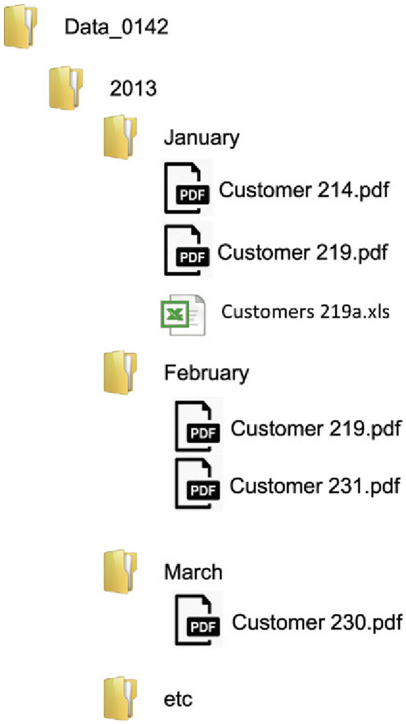

FIGURE 17 An unstructured data extract

6.2 PITFALLS AND RISKS

The Data Load stage has several pitfalls and risks to bear in mind.

- **Data corruption:** As discussed in the data extraction stage, data can be corrupted as it go from a file system or source system into your DME. Data records can be dropped and data values can be corrupted. How do you gain confidence that this has not happened?
- **Data preparation:** Very often, raw data files have to be modified before they can be successfully loaded into the DME. The case of graphics and text embedded in spreadsheets has already been mentioned as an example of this scenario. If these "preparatory" modifications to data are not carefully controlled, then you lose data provenance. How much modification is appropriate and what are the best approaches to making these modifications clear and reproducible?
- **Where did the data go?** Loss of the link between data on the file system and loaded data in the DME breaks data provenance. Without clear team guidelines, loaded data can be located anywhere in the DME. It then becomes difficult to know which raw data file was the source of a particular dataset. When investigating a problem with your data, the trail then goes cold at Data Load.

This chapter now describes some tips to mitigate these risks in a Guerrilla Analytics project.

6.3 PRACTICE TIP 13: MINIMIZE MODIFICATIONS TO DATA BEFORE LOAD

6.3.1 Guerrilla Analytics Environment

As already discussed, loading data is difficult because of the variety of data formats, inconsistencies, and large volumes that are time-consuming to process. Inevitably, a raw data file may have to be modified so that it loads successfully into the DME. In other scenarios, there are explicit modifications that you absolutely should make to a raw file before loading to facilitate the maintenance of data provenance.

Think about spreadsheets as an example file format that is often encountered in analytics work. Spreadsheets can have hidden rows and columns. They can have embedded charts, graphics, and text boxes. Their columns can be derived from functions rather than hard coded data values. They may have "data" embedded in color-coding and formatting of content. They can be divided into separate tabs that are linked to one another. Spreadsheets are a particularly difficult example. Even a plain text file may have to have its line endings or its encodings changed.

Some changes need to be made to these files before they are loaded. But where do you draw the line? If you are going to extract spreadsheet tabs into individual files, why not also add an extra calculated column to save having to do it later within the DME? The danger is, the more changes you make outside the DME with ad-hoc manual processes, the less traceable data provenance becomes.

6.3.2 Guerrilla Analytics Approach

Minimize the modifications done to data before it is loaded into the DME. Modifications outside the DME are more difficult to track and reproduce, so try to do the bare minimum necessary to get a file to load successfully into the DME.

6.3.3 Advantages

There are several immediate Guerrilla Analytics advantages to minimizing data modifications outside the DME.

- **Traceability on the file system:** There is minimal difference between the raw data as it was received from the provider and the raw data that has been loaded into the DME. This means less documentation is required for other team members to reproduce the data preparation, should a load have to be redone.

- **Traceability in the DME:** The data that is loaded into the DME is as similar as possible to the raw data on the file system. This means that you can use program code in the DME to report on raw data characteristics rather than having to go back out onto the file system. For example, imagine a spreadsheet file that has been loaded into the DME. Only 3 of the 15 spreadsheet data columns are required for analysis, but you have followed this practice tip and loaded all columns. Your loaded data now has all the same data fields as the raw spreadsheet. This means you can easily run queries and report records from the loaded spreadsheet that the customer recognizes and can understand. Should more data columns come into scope, you do not have to revisit the data load.
- **Reproducibility of data modifications:** Since all analyses based on the raw data are done using program code in the DME, it is easier to understand exactly what those analyses were, and reproduce them if necessary. Again spreadsheets are a particularly troublesome file format here. If modifications have been done to raw data in a spreadsheet, it is quite likely that these modifications involved cutting and pasting data, dragging formulae, or calculating derived data fields with spreadsheet functions. All of these modifications are difficult to reproduce without detailed documentation. Program code modifications, by contrast are more succinct, easier to reproduce, and can be version controlled.

6.4 PRACTICE TIP 14: DO DATA LOAD PREPARATIONS ON A COPY OF RAW DATA FILES

6.4.1 Guerrilla Analytics Environment

On occasion you will have to modify raw data files so they can be loaded into the DME. Even though modifications are minimal, they are still a change to the raw data the customer provided. Questions could be raised over data provenance and whether your "minimal" modifications corrupted the provided data.

6.4.2 Guerrilla Analytics Approach

Any modifications to raw data should be done to a copy of the raw data file.

6.4.3 Advantages

The advantages of this tip are two-fold in a Guerrilla Analytics environment.

- **Data provenance is preserved:** There are two files, raw and modified, on the file system. If there are concerns raised about data loss or data corruption, these files can be compared and investigated.
- **Errors can be corrected:** In the event that a load has to be rerun because a file was inadequately prepared, there is an unmodified copy of the raw data that can be used to start from scratch.

6.5 PRACTICE TIP 15: ADD IDENTIFIERS TO RAW DATA BEFORE LOADING

6.5.1 Guerrilla Analytics Environment

Some data manipulation languages such as SQL do not preserve the ordering of rows in a data file as it is loaded into a dataset. For example, row 1413 in a raw text file will not necessarily be row 1413 in the equivalent loaded dataset. This causes problems for data provenance when data does not have unique identifiers, as is often the case with logs and spreadsheets, for example. If you encounter an issue with some of the data, you have a difficulty in identifying this record of data to the provider who gave you the original file. The row 1413 they refer to can only be identified by comparing all the data fields in the row, and this is time-consuming.

6.5.2 Guerrilla Analytics Approach

Before loading any data file, add a unique row number to the file. For text files, command line tools such as SED and AWK (Dougherty and Robbins, 1997) can easily run through large text files adding a row number column at the start of every row. For spreadsheets, it is a simple modification to create a row number column.

6.5.3 Advantages

When every row has a unique row number, there can be no ambiguities around identifying and communicating about a row of data. The row that the customer sees in their source file or spreadsheet is the dataset record with the same row number in your DME.

6.6 PRACTICE TIP 16: PREFER ONE-TO-ONE DATA LOADS

6.6.1 Guerrilla Analytics Environment

You will sometimes receive data that is scattered across a number of files. Perhaps the data has been chunked into files of a million records each to facilitate loading of the original source file incrementally. Perhaps a spreadsheet with many tabs needs to be broken out into individual files. You then have a choice. Do you load a large number of individual files into the DME? Alternatively, do you append those files together on the file system and do a single load of the concatenated file into the DME? The former approach is potentially more time-consuming but is better for data provenance because it preserves a clear one-to-one mapping between file system and DME. The latter approach is quicker, but requires documentation of the modifications that were done on the file system.

6.6.2 Guerrilla Analytics Approach

As far as possible, always load a single raw data file into a corresponding single dataset in the DME. Avoid appending files on the file system before loading unless these modifications can be easily understood and repeated.

6.6.3 Advantages

The motivation for this tip leads back to data provenance. If each file on the file system has a corresponding dataset in the DME, it is much easier to trace where a particular dataset came from. If 10 files on the file system become a single dataset in the DME, it is harder to follow this data flow back to the file system. This wastes time for the Guerrilla Analyst when trying to track down a bug in a Data Load or confirm a data receipt from the customer.

6.7 PRACTICE TIP 17: PRESERVE THE RAW FILE NAME AND DATA UID

6.7.1 Guerrilla Analytics Environment

A data file has to land somewhere when it is loaded into the DME. That somewhere is a dataset that has a name and a location. If it is difficult to determine where a particular file was loaded into the DME, time is wasted trying to track down the data.

6.7.2 Guerrilla Analytics Approach

As discussed in the Data Receipt stage, all received data files should be given a UID to help traceability. These files should be loaded into a dataset with a name containing the data UID in addition to the raw file name.

6.7.3 Advantages

The advantages of this tip are as follows:

- **Ease of identification:** By looking at a dataset name in the DME, you can immediately identify the raw file location in the data folder by virtue of the data UID and file name.
- **Ease of communication:** By using a dataset's name in the DME, you can immediately refer to it in terms that the customer understands from the data file they delivered to you.

These advantages are illustrated in Figure 18 for the example of a relational database DME. Which of the two data files in the data folders on the left are easier to identify in the DME on the right? The file "Marketing_Statement.xls" stored under data UID 096 can be found in DME name space 096 in a dataset

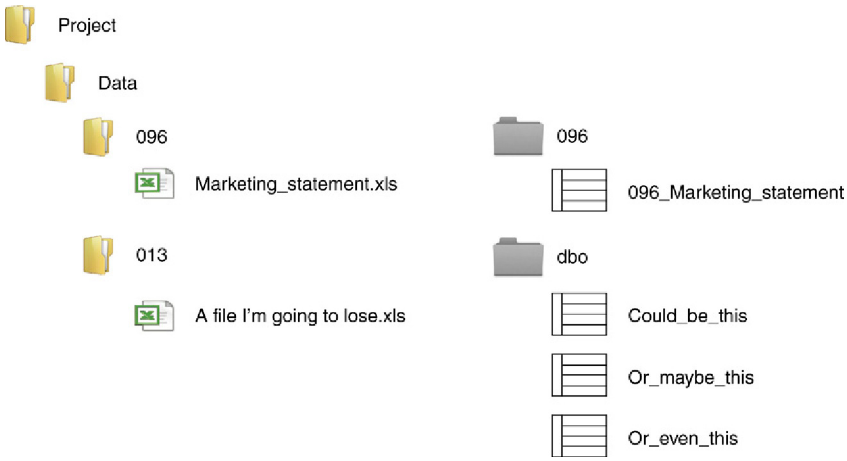

FIGURE 18 Preserve the raw file name and data UID

called "096_Marketing_Statement." The file under data UID 013 by contrast could be in any or none of the datasets scattered in the generic DME namespace "dbo."

A simple convention of preserving the raw file name and data UID makes communication and identification easier with no administrative overhead. This is critical for the Guerrilla Analyst who must stay up to speed with frequently changing data.

6.8 PRACTICE TIP 18: LOAD DATA AS PLAIN TEXT

6.8.1 Guerrilla Analytics Environment

It is possible in many DMEs to convert data into specific types as it is loaded. For example, you could specify that a numeric value gets loaded into a data field with a data type of integer. Alternatively, you could specify that the same numeric value gets loaded into a data field with a data type of floating point. With an unstructured document, you could enrich it as it is loaded, tagging entities such as people and business names. This changing of data at load should be avoided.

- **Brittle data loading:** Data type conversions at load time lead to a more brittle load process. There is nothing more frustrating than having a 1-hour load process fall over at its last data record because an unexpected value was encountered in the data. For example, a data loader encounters a text value when trying to load data into a numeric target field.
- **Less traceability:** Data Load utilities often give you less control of data type decisions. This makes load processes more difficult to trace and reproduce.

6.8.2 Guerrilla Analytics Approach

As far as possible, you should load all data in as generic and raw a format as possible. This usually means that all dates, numbers, currencies, etc. are loaded as text. Unstructured content is loaded as is without any enrichment. Conversion into more specific data types and enrichment can be done subsequently in program code.

6.8.3 Advantages

There are several advantages to loading data as plain text.

- **Robust loads:** Data Loads are less brittle because they use a simple data format that should work regardless of the contents of a data field. If numeric data is loaded into a text field, an occasional text value such as "Not applicable" will not cause the load to fail.
- **Better traceability:** In subsequent analytics workflow stages, loaded text data fields can be converted into more specific types using program code. This makes decisions on their type conversion clear and traceable.
- **Faster development times:** If a data conversion is incorrect, it can be quickly corrected in program code rather than having to undo and rerun a more time-consuming Data Load process.

Loading data as plain text speeds up the load and development process, so the Guerrilla Analyst can get on with the high value work of analyzing the data.

6.9 COMMON CHALLENGES

There is a lot to take in at the Data Load stage. Here are some common challenges encountered with teams and the possible resolutions when thinking about the Guerrilla Analytics principles.

6.9.1 Shouldn't My Data Preparation for Loading be Reproducible and Documented Too?

Ideally, yes it should. You have to keep in mind one of the overarching Guerrilla Analytics challenges, which is that of limited time. Your objective is always to preserve data provenance despite disruptions and constraints. The main occasion that would cause you to question your steps in preparation for Data Load is if you have discovered severely corrupted data that have to be extracted again or reprepared correctly. At this stage it matters little what the steps in the preparation were. All that matters is that something was broken when going from the raw file in your data folder to the prepared file in your data folder. Time to redo it!

6.9.2 If I'm Adding Metadata Such as Row Number Into My Loaded File, Should I also Add Receipt Date, Author, ... etc.?

The answer here is to think about what you most need for maintaining data provenance. Every piece of data has a UID and that UID is contained in the dataset name. The data log contains all the metadata you could possibly want. Include the metadata that will be useful in helping you preserve and report on data provenance. If there are likely to be questions around who provided various pieces of data, then perhaps it is helpful to make that answer available in the DME dataset. If these questions are not likely, then keep it simple and leave that information in the data log where it can be looked up with a small amount of effort. In general, including the data UID is enough to locate all the data tracking information you need.

6.9.3 Preserving these Crazy File Names is too Awkward. I Don't Want to Type Long Dataset Names When I Write Code

Unfortunately, losing track of the link between raw data files and loaded datasets is more time-consuming than typing an awkwardly named dataset. Most modern development environments can autocomplete dataset names. All things being equal, data provenance wins here.

6.10 WRAP UP

This chapter has discussed the Data Load stage of the Guerrilla Analytics workflow. Having read this chapter, you should now understand:

- The activities that take place during the Data Load stage of the workflow.
- The challenges of a Data Load.
- The common pitfalls and risks associated with data loading. These are:
  - Data loss through corruption and truncation.
  - Appropriate data preparation for load.
  - Location of loaded data in the DME.

  You should now have some useful practice tips to help you address the pitfalls of Data Load. Specifically, you know to:
  - Minimize data modifications in preparation for load into the DME.
  - Prepare a copy of raw data files so that you never modify source data.
  - Add identifiers to raw data records before load.
  - Prefer Data Loads that map one file to one dataset.
  - Preserve the raw file name and data UID in the loaded dataset name.
  - Load data as plain text.

You should also be able to counter the common challenges to this chapter's tips.