Data Science in the Database:
Using SQL for Data Preparation

D

Antonio Badia
University of Louisville, USA

INTRODUCTION

The analysis of data requires several steps that are usually denoted as the data life cycle. Typically, these steps include data discovery and upload, data exploration, data cleaning and wrangling, data analysis proper, and dissemination of results/further analysis (Badia, 2019). While much research focuses on the analysis itself, in the form of algorithms for Data Mining (DM) or Machine Learning (ML), popular accounts indicate that data scientists spend up to 80% of their time preparing the data for analysis, that is, in the exploration, cleaning and wrangling stages (Dasu & Johnson, 2003; Wickham, 2014). Hence, this part of the process requires proper attention and analysts need the proper tools to deal with the problems that arise at these stages.

Much data resides in databases; however, it is common to transfer such data to other environments like R (Wickham and Grolemund, 2017) or text files to be processed by programs in languages like Python (VanderPlas, 2016), since database usually do not provide tools for in-depth data analysis. In some cases, it can be beneficial to deal with data exploration and cleaning in the database itself. The objective of this article is to describe how data exploration, cleaning and preparation can be carried out in typical relational databases using the already existing capabilities of most SQL systems (Badia, 2019; Linoff, 2008; Trueblood, 2001). We show how to carry out basic data analysis and cleaning by providing examples of SQL commands over a simple table. We assume that the reader has basic knowledge about SQL, but not necessarily in-depth expertise. The references provided are excellent resources for the reader interested in learning more SQL.

BACKGROUND

Data must be prepared for analysis. Most DM) and ML algorithms make assumptions about the format and other properties of data (Dasu & Johnson, 2003). However, 'data in the wild' rarely conforms to such expectations. Discovering any problems or issues that could render data not ready for analysis and solving them is the goal of the data preparation process.

The process starts with Exploratory Data Analysis (EDA), where data is examined with simple, descriptive statistics in order to determine whether data problems exist. Most datasets often come with *dirty data* (Wickham, 2014); typical problems found at this stage include missing data, outliers, formatting problems and structure problems. Missing data refers to values that are absent. This can be a serious issue when the number of missing values is high (as a percent of all values) or when the values are not *missing at random* (that is, certain values are more likely than others to be missing), since this may introduce bias in the analysis. An outlier is a data point that is quite different from other data points and therefore could potentially be the result of an error in data gathering or storage. Some DM and ML algorithms can

DOI: 10.4018/978-1-7998-9220-5.ch069

be greatly affected by the presence of outliers; for instance, linear regression is particularly sensitive to this issue (Dasu & Johnson, 2003). Formatting problems refer to data values that are not encoded in the expected shape or arrangement. This includes issues like numbers that are not written out as numbers ("42"), dates that cannot be recognized as dates by the system ("Feb ten 2020"), and similar. This type of error is not infrequent, since computers expect values to be represented in certain ways, and any deviations may result in values being ignored or misinterpreted. Finally, structure problems arise because data is not structured in the way required by the data analysis algorithm to be used. In a typical dataset, there is a set of attributes that describe the data, and data elements are represented by a tuple or row of values, one for each attribute. However, some datasets do not come with this 'table-like' structure, while most DM and ML algorithms assume it (Wickham, 2014). All these are issues that must be detected and solved before analysis can proceed (Berthold et al., 2010).

FOCUS OF THE ARTICLE

In this section, we describe the typical activities that are carried out in data preparation; for each, we provide a short description, an example, and describe how it could be handled in a relational database using SQL. The first step is to load data into the database; the second one is to carry out some Exploratory Data Analysis (EDA) in order to discover characteristics of the dataset and any issues the data may have. At that point, *Data Cleaning* is attempted, in order to solve the issues and get data ready for analysis. Issues that are especially important are missing data, outliers, structural problems and duplication. We describe each step in a separate subsection.

Getting Data In and Out of the Database

Relational databases store data in units called *tables*. In fact, a database can be considered a collection of tables.

To create a table, the SQL language has a command, called (not surprisingly) CREATE TABLE. A very simple example of this command is

```
CREATE TABLE Employees (
  name char(64),
  age int,
  date-of-birth date,
  salary float)
```

Here, 'Employees' is the name of the table, and 'name', 'age', date-of-birth' and 'salary' are the names of the 4 attributes (typically called 'variables' in statistics and 'features' in Data Mining and Machine Learning) that make up the *schema* or structure of the table. Each attribute is given a data type: 'char(64)' means a string of up to 64 characters; 'int' denotes an integer, 'date' a data and 'float' a real-valued number. All database systems provide several data types to represent collections of values, including different types of numbers, strings, and temporal types (dates, times and timestamps).

If data is already in a file, it is possible to load the data into the database in one swoop. All database systems have some command which takes a file name and a table name and brings in the data from the file into the table -as far as the data in the file is compatible with the schema of the table. This command