Odlaw
Retroactive GDPR Compliance For Relational Databases

Jearson Alfajardo
Brown University

Connor Luckett
Brown University

1 Introduction

The requirements of the GDPR to report and/or delete a user's data upon request is a frustratingly difficult problem to solve. Doing this in the GDPR's timeframes adds further complications.

Simply put, the best way to comply with this requirement is to construct a compliant system from the beginning. Unfortunately, there are many already-deployed systems, and the process of retroactively complying is not always straightforward. Many applications are small in scale, and generating user data views may seem trivial. However, this process quickly becomes less intuitive for larger applications. Clearly, there is a need to investigate solutions for sthis problem.

We hope to leverage the fact that many of these systems utilize relational databases. Relational databases may have fallen out of favor for newer noSQL solutions, but many systems continue to rely on a relational back-end. Knowing the system in question is a relational database allows us to make assumptions about the data and how it fits together. By querying the schema of the database, we should be able to understand how tables relate to each other. We can use that knowledge to generate queries in an attempt to automate the user data export process. It will also be of use for finding and marking all user data for deletion if they wish to be forgotten.

2 Plan

We plan to show the GDPR encourages the use of good habits for database normalization and key specification. Our current algorithm will depend heavily on basic good habits (e.g. specifying primary and foreign keys, normalization, etc.). We intend to show that under certain conditions, the problem of generating a user's data is relatively simple for a well-constructed relational database. We also plan to incorporate features for alerting the database administrator (DBA) when the database was not well-designed (e.g. lack of foreign keys specified).

Our first goal is to generate queries in order to find all data related to a user. Later, we plan to refine our method to give DBAs opportunities to specify "secret" data columns (e.g. unique IDs and password data), which should *not* be returned when generating user data reports. Further, we will consider the automatic deletion process of a user's data. Although we do not expect to fully solve this problem, we believe our techniques can ease the frustration by providing a means to guarantee that *all* data possibly pertaining to the user will be deleted from the main database.

3 Techniques Applied

The material in our project is wholly guided by the GDPR's Chapter III (Rights of the Data Subject). Specifically, we consider Article 13 (Information to be provided where personal data are collected from the data subject) and Article 17 (Right to erasure). Having a full and proper understanding of the requirements will be of the utmost importance.

We plan to present an algorithm to automatically generate queries. We will need to describe and show the efficiency of our algorithm. We will also need to discuss our correctness and consider examples where our techniques fail. Thus, an introductory knowledge of algorithms should prove to be useful. Currently, we hope to implement it by way of a graph traversal. A graph will allow a database administrator to visually verify that the data properly relates to each other, and it can be used as evidence to show specifically what user data was deleted.

Our overall intent is to construct a working example of our algorithm with an application implemented in Python or Java. We hope to first implement our application for a popular back-end, such as MySQL or PostgreSQL. This project will rely heavily on the use of standard database connection libraries (JDBC or SQLAlchemy). In addition, we will likely need to utilize graph libraries (JGraphT, NetworkX) or possibly roll our own. An elementary knowledge of SQL querying and table structures will be essential. Definitions of normalization will be used to discuss a minimum level of compliance for Odlaw to function properly.