Purpose and consent enforcement in DBMS

Javam Machado[1], **Paulo Amora**[1], **Francisco D. B. S. Praciano**[1]

[1]LSBD/DC – Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brazil

{javam.machado,paulo.amora,daniel.praciano}@lsbd.ufc.br

Abstract

Nowadays, personal data is subject to laws and regulations that oblige data holders to ensure proper compliance with users' consent regarding how their data should be used. Existing tools in the DBMS, like RBAC/FGAC, can accomplish some level of control over data. However, they cannot model and correctly apply the required and desired restrictions, bringing this theme to a resurgence. In this tutorial, we explore a timeline of approaches to solve purpose-based access control and compare recent works over a common baseline to assert strong and weak points and suggest new research topics on this theme.

Introduction

Personal data must be responsibly handled by companies and businesses that collect and store this data. Specific legislation such as the Health Insurance Portability and Accountability Act (1996), which applies to health data and more broad legislation like Lei Geral de Proteção de Dados (LGPD) (2018), General Data Protection Regulation (GDPR) (2017), California Consumer Privacy Act (CCPA) (2018) pushed these companies to better care for this personal data, due to the risk of fines and regulatory sanctions. Academia has extensively studied this subject in works such as Hippocratic Databases [Agrawal et al. 2002], which describes a strawman architecture for dealing with personal data under the optics of HIPAA, although their principles apply in a broader fashion.

With the theme resurgence due to the recent regulations, other works like SchengenDB [Kraska et al. 2019] model a database architecture to comply with GDPR, Schwarzkopf et al. [Schwarzkopf et al. 2019] proposes a data storage model and a set of GDPR-compliant materialized views to answer queries ensuring rights, GDPRBench [Shastri et al. 2020] explores the impact of adding metadata to comply with GDPR articles. For purpose and consent, an attribute is added to the schema and checked during query processing and Machado et al. [Machado and Amora 2021] addresses techniques and approaches to deal with several rules imposed by these laws, which are quite extensive and deal with several aspects of data, some are right of access, right to be forgotten, right of consent. Although handling consent authorizations and appropriate data usage can be observed as a data access control problem, works have shown that roles and permissions are not the best tools to manage these entities adequately.

To better understand the problem, purpose can be defined as a finality for which the data shall be used. For instance, if I sign up for a social network, I could only provide my data to build my profile and accept that this data be used to be shown to other users. It is easy to see how this data can be misused, as the social network itself or third-party contractors may use it to derive information about my personal preferences or even drive

content to shape my opinions because this scenario happened in the Cambridge Analytica scandal. Therefore, applications and services must inform clearly and explicitly for what purposes the personal data will be used, and the user must consent to these purposes. Consent, then, can be defined as the previous, informed, explicit accordance to which purposes a user's data shall be used.

The tutorial delves into works that show the evolution of purpose and consent handling, including the most recent literature and presents their architectures. It discusses the strengths and weaknesses of each approach and concludes with a comparative experiment of three more recent works and a baseline to highlight how they fare over common ground. The tutorial concludes with some guidelines for future research.

Access control

Access control is a feature of many DBMSs, as specified in the SQL by the GRANT/REVOKE clauses. This feature, called Role-Based Access Control (RBAC), can restrict access to database objects like tables, rows, and attributes based on user or group identification. Another data access control mechanism is Views, which can expose only select data from different relations in the database. To represent this scenario, we selected two approaches that use these features to first attack the problem at hand.

Agrawal et al. [Agrawal et al. 2005] proposes using Fine Grained Access Control (FGAC) tools to implement purpose-based access. The work proposes attaching a Policy Translator, which will rewrite queries using the defined FGAC restrictions and other metadata inserted in the database to return only allowed results from the tables.

Byun and Li [Byun and Li 2008] propose a method to hierarchically organize purposes and evaluate whether a register should be returned through a simulation of release using concepts like Allowed Intended Purposes and Prohibited Intended Purposes. The work uses RBAC to model this approach and stores purposes inside a table, which is queried through a query modification algorithm that adds predicates to the executed query.

These works model purpose-based access control using existing features of DBMSs and add layers to ensure that these data are only returned to the permitted parties. However, the definition of consent is still unclear since these works only deal with hard concepts of accept/deny. Pappachan et al. [Pappachan et al. 2022] argue that RBAC/FGAC is not sufficient as a model because it allows for data discovery even if the intended data is filtered out. Konstantinidis et al. [Konstantinidis et al. 2021] proposes a collaborative policy that allows for better modeling of consent data, increasing the utility of retrieved data without violating consent restrictions. These works are better detailed below.

Purpose-based Access Control

With the new regulations, control over how data should be used became more refined, and the theme saw a revival. Now, the data owner must be given a set of choices for data purpose processing so he/she can explicitly allow data usage. Collected data with the corresponding authorizations are usually stored in database management systems that are responsible for retrieving and processing the data following the owner agreement.

Specific to Purpose-based Access Control (PBAC), Sieve [Pappachan et al. 2020] is a middleware that deals with the complexity of ever-scaling data and policies, using

an approach that greatly reduces the number of necessary checks to ensure consents are being complied with. It uses specific indexes and index guards to generate these indexes, and rewrite queries to force usage of these indexes. Purpose and consent data is stored in the database, along with other information that is used as predicates in query templates used by the middleware.

Konstantinidis et al. [Konstantinidis et al. 2021] provide formal constructs to model consent and purpose since constraints may be applied based on context; for example, there is no problem on a value being returned by itself, but it cannot be combined with another. The work provides Consent Constraints, which are used to select the most general query unifications (MGQU) and rewrite the queries using annotated data to ensure compliance with defined consents. Purpose and consent data are annotated in tuples and relations, these being used to calculate and create the MGQUs.

Pappachan et al. [Pappachan et al. 2022] argue that access control policies may reveal unintended information, allowing for data and knowledge leakage, which defeats the purpose of PBAC, since the querier could infer or even discover unreleased data due to data dependencies. This argument builds a stronger case for why RBAC and FGAC shouldn't be used to model PBAC by themselves. The work presents a technique called Full Deniability, which hides additional data if this data can be used to discover the values of a given cell or row in the DBMS.

Purpose Scan [Praciano et al. 2022] brings purpose verification and consent modeling inside the database. Defining specific operators to execute over personal data, this work prevents the querier from even knowing that they are querying personal or unrestricted data. The work models purpose and consent, purpose data is stored efficiently using filters, and purpose is enforced through special operators that replace the usual ones when the relation contains personal data. It also reduces data movement because only blocks with allowed purposes are retrieved to memory during the execution, adding an extra layer to prevent data leakage. To set Purpose Scan, the database administrator uses a SQL extension called ConSQL [Ítalo de Abreu et al. 2021], which allows the configuration of which tuples have assigned purposes through a syntax similar to how GRANT/REVOKE is defined, although the Purpose Scan modified DBMS executes different operations to ensure purpose definition and enforcement.

Experimental Evaluation

To conclude the tutorial, we present an experimental evaluation performed over four works: Sieve [Pappachan et al. 2020], Konstantinidis et. al. [Konstantinidis et al. 2021], Purpose Scan [Praciano et al. 2022] and as a reference, Hippocratic Databases [Agrawal et al. 2002]. Each of these works uses a different dataset for evaluation, so we made an effort to standardize all of them to use TPC-H, a widely known benchmark for analytical queries. We used three different metrics to compare them: storage space, throughput, and query plan modification. Given the different nature of each work, some inside the DBMS, some attached as a middleware, and others embedding metadata to data itself or the schema to achieve their goals, we found other metrics such as I/O measurements and result set completeness unfair to a given work.

Acknowledgements

The authors would like to thank Ítalo de Abreu for his contributions in developing the experimental comparison between the techniques. This research was partially funded by CNPq/Brazil under grant number 316729/2021-3.

References

Agrawal, R., Bird, P., Grandison, T., Kiernan, J., Logan, S., and Rjaibi, W. (2005). Extending relational database systems to automatically enforce privacy policies. In *ICDE*, pages 1013–1022. IEEE Computer Society.

Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. (2002). Hippocratic databases. In *VLDB*, pages 143–154. Morgan Kaufmann.

Byun, J. and Li, N. (2008). Purpose based access control for privacy protection in relational database systems. *VLDB J.*, 17(4):603–619.

Konstantinidis, G., Holt, J., and Chapman, A. (2021). Enabling personal consent in databases. *Proc. VLDB Endow.*, 15(2):375–387.

Kraska, T., Stonebraker, M., Brodie, M. L., Servan-Schreiber, S., and Weitzner, D. J. (2019). Schengendb: A data protection database proposal. In *Poly/DMAH@VLDB*, volume 11721 of *Lecture Notes in Computer Science*, pages 24–38. Springer.

Machado, J. C. and Amora, P. R. P. (2021). The impact of privacy regulations on DB systems. *J. Inf. Data Manag.*, 12(5).

Pappachan, P., Yus, R., Mehrotra, S., and Freytag, J. (2020). Sieve: A middleware approach to scalable access control for database management systems. *Proc. VLDB Endow.*, 13(11):2424–2437.

Pappachan, P., Zhang, S., He, X., and Mehrotra, S. (2022). Don't be a tattle-tale: Preventing leakages through data dependencies on access control protected data. *Proc. VLDB Endow.*, 15(11):2437–2449.

Praciano, F. D. B. S., Amora, P. R. P., Abreu, I. C., and Machado, J. C. (2022). Purpose scan: A purpose-aware access method. In *Poly/DMAH@VLDB*, volume 13814 of *Lecture Notes in Computer Science*, pages 24–36. Springer.

Schwarzkopf, M., Kohler, E., Kaashoek, M. F., and Morris, R. T. (2019). Position: GDPR compliance by construction. In *Poly/DMAH@VLDB*, volume 11721 of *Lecture Notes in Computer Science*, pages 39–53. Springer.

Shastri, S., Banakar, V., Wasserman, M., Kumar, A., and Chidambaram, V. (2020). Understanding and benchmarking the impact of GDPR on database systems. *Proc. VLDB Endow.*, 13(7):1064–1077.

Ítalo de Abreu, Praciano, F., Amora, P., and Machado, J. (2021). ConSQL: Consentimentos em SQL para o processamento de consultas orientado a propósitos. In *Anais Estendidos do XXXVI Simpósio Brasileiro de Bancos de Dados*, pages 8–14. SBC.