Differential Purpose Scan: An End-to-End Privacy-aware Access Method

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

Laboratório de Sistemas e Bancos de Dados (LSBD)
DC/UFC – CEP 60440-900 – Fortaleza – CE – Brazil

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

Abstract. *This work proposes a new operator, Purpose Scan (PS), which is introduced into the execution plan to ensure that data owners' consents for specific purposes are respected. It also introduces the Differential Purpose Scan, which incorporates differential privacy to prevent information leakage. Experiments show that PS improves performance compared to view-based approaches.*

Resumo. *Este trabalho propõe um novo operador, Purpose Scan (PS), que é introduzido no plano de execução para que os consentimentos dos donos dos dados para propósitos específicos sejam assegurados. Também é proposto o Differential Purpose Scan, que adiciona privacidade diferencial para evitar vazamentos de informação. Experimentos demonstram que o PS melhora o desempenho em relação a abordagens baseadas em visões.*

1. PhD Information

Level	PhD
Admission	2020.1
All Credits Completed	2020.2
Qualifying Exam Expected	2025.2
Thesis Proposal Defense Expected	2025.2
Thesis Defense Expected	2026.1

To date, the publications related to this work are: [Ítalo de Abreu et al. 2021], [Praciano et al. 2022], [Machado et al. 2024], [Amora et al. 2025].

2. Introduction

As data becomes increasingly important and data processing techniques advance, Database Management Systems (DBMS) are now utilized not only for storing data but also for managing personal data. However, this scenario has also raised privacy concerns, especially when this personal data is used indiscriminately to infer sensitive information about individuals. A key challenge is ensuring that the use of personal data respects users' informed consent, which means that this consent authorizes only specific applications to retrieve the personal data. This challenge introduces the concept of *purpose-aware access*, where the application must declare its purpose, and that purpose must be previously authorized by the user. While DBMSs already include access control techniques, they cannot apply restrictions based on specific purposes, as these techniques primarily focus on data security, not on enforcing purpose-specific restrictions based on user consent.

In this context, this work proposes an approach to incorporate *purpose-aware access* into DBMSs, ensuring effective user consent enforcement. A rule-based access control solution could, in theory, address this issue, but it presents significant limitations. Purpose-based access control aims to ensure that data is accessed only for explicitly authorized purposes, but modeling consent semantics can be complex [Ítalo de Abreu et al. 2021]. Furthermore, these models do not scale well with the increase in users or restrictions [Pappachan et al. 2020] and data remains vulnerable to leakage attacks [Pappachan et al. 2022], compromising sensitive information even with access control in place. This research proposes a solution that integrates the Purpose Scan operator to implement *purpose-aware access* and the Differential Purpose Scan, which enhances privacy and protects against data leakage attacks.

3. Research Problem

The problem of purpose-based access control has been extensively studied in the academic literature, with foundational works such as [Rizvi et al. 2004, Kabra et al. 2006, Agrawal et al. 2005, Byun and Li 2008]. More recently, supported by new data protection regulations, newer studies such as [Shastri et al. 2020, Pappachan et al. 2020, Deshpande 2021, Pappachan et al. 2022] have modeled purpose-based access control to ensure consent compliance. These works implement purpose-based access through query rewriting either at the SQL level, via stored procedures, or through middleware positioned between the DBMS and external applications. Using these approaches, it is possible to access data through usual access paths. For example, a scan operator applied directly to the relation retrieves all data; then, further operations are applied to ensure compliance with the consents. This either allows for an attacker to query the database directly, bypassing the interface, or obtain this data by using User Defined Functions (UDFs).

Thus, we argue that this unprotected gap can be filled if only a special operator can access relations containing personal data. Many of these solutions require that the query's purpose be explicitly declared within the query so that the solution can be rewritten in a valid SQL query, adding constraints to it. This brings another issue, which is leaving it open to possible unauthorized modifications. Ideally, a DBMS must be transparent in this regard, first, to guarantee that whoever queries the system remains ignorant of this additional layer within it. Second, to allow previous queries and procedures to remain unaltered, avoiding rework to use the new system. In short, the above scenario leads to the following first problem statement addressed in this PhD.

Problem Statement 1. *Is it possible to primitively enforce purpose constraints during data retrieval?*

To address this issue, we introduce the purpose-aware operator Purpose Scan for relational DBMSs, ensuring compliance with data protection legislation while operating transparently within query processing [Praciano et al. 2022]. Unlike existing solutions, our approach integrates purpose verification directly into data retrieval, reducing the risk of external data breaches. It requires no changes to query parsing, maintaining the same usage for SQL queries, and is modular, modifying only the operator in the query plan, making it compatible with other operators that rely on scans.

Although Purpose Scan provides an effective solution for enforcing user consent, it remains susceptible to information leakage through inference attacks, such as the one

demonstrated in [Pappachan et al. 2022]. To address, we tackle the following question:
Problem Statement 2. *Is it possible to enforce purpose constraints while providing formal guarantees against information leakage?*

To answer this problem, we propose Differential Purpose Scan (DPS), an extension of Purpose Scan that incorporates differential privacy techniques [Dwork 2006] to prevent information leakage through inference attacks. By doing so, DPS ensures not only the enforcement of user consent but also the preservation of user privacy.

4. Related Work

[Agrawal et al. 2005] propose using access control (AC) techniques to implement purpose-based access, introducing a Policy Translator that rewrites user queries based on ACs policies and additional metadata to ensure authorized results are returned. [Byun and Li 2008] suggest a hierarchical organization of purposes, determining whether a record should be returned by simulating its release based on concepts like Allowed and Prohibited Intended Purposes. Their approach uses access control techniques and stores purpose metadata in dedicated tables, enforcing constraints through query modification by appending purpose-based predicates.

[Kraska et al.] present SchengenDB, a database architecture designed to ensure compliance with the General Data Protection Regulation (GDPR). SchengenDB introduces tools to enforce purpose-based access control, manage data subject consent, and uphold the right to be forgotten. Additionally, SchengenDB proposes sandboxing applications to prevent unauthorized data leakage.

Sieve [Pappachan et al. 2020] is a middleware designed to handle the growing number of consent policies by minimizing the number of checks required to enforce compliance. It introduces index guards to efficiently filter data access, rewriting user queries to ensure these indexes are utilized. Purpose and consent metadata, along with other contextual information, are stored within the database and used as predicates in predefined query templates managed by the middleware.

[Konstantinidis et al. 2021] introduce formal constructs to model consent constraints that may depend on contextual combinations of data. For example, a value may be permissible in isolation but not when combined with others. The authors define Consent Constraints, which guide the selection of Most General Query Unifications (MGQUs) used to rewrite queries in compliance with consent restrictions. Purpose and consent metadata are annotated at both the tuple and relation levels, and these are leveraged to compute and enforce the MGQUs during query processing.

[Pappachan et al. 2022] highlight that traditional access control policies can unintentionally expose sensitive information through data dependencies, allowing users to infer unreleased data. They propose Full Deniability, a technique that suppresses data to prevent the disclosure of protected values. Their system builds on the middleware architecture from Sieve [Pappachan et al. 2020] to efficiently manage consent and purpose.

Finally, Table 1 presents the related work for comparison with both Purpose Scan [Praciano et al. 2022] and Differential Purpose Scan. The main distinction is that Purpose Scan is the only approach integrated directly into the DBMS core. At the same time, DPS is the only one that has incorporated differential privacy to prevent information leakage.

Table 1. Comparison of Related Work.

Work	IsMiddleware	Metadata	AC	Plan Change	Prevent Leakage
Agrawal et al., 2005	Yes	Additional Tables	Yes	Query Rewriting	No
Byun et al., 2008	N/A	Additional Tables and Table Schema	Yes	Query Rewriting	No
Kraska et al., 2019	No	Embedded	No	No	Yes
Pappachan et al., 2020	Yes	Additional Tables	No	Query Rewriting/Indexes	No
Konstantinidis et al., 2021	Yes	Embedded/Annotations	No	Query Rewriting	No
Pappachan et al., 2022	Yes	Additional Tables	No	Query Rewriting/Indexes	Yes
Praciano et al., 2022	No	Embedded	No	Added Operators	No
Differential Purpose Scan	No	Embedded	No	Added Operators	Yes

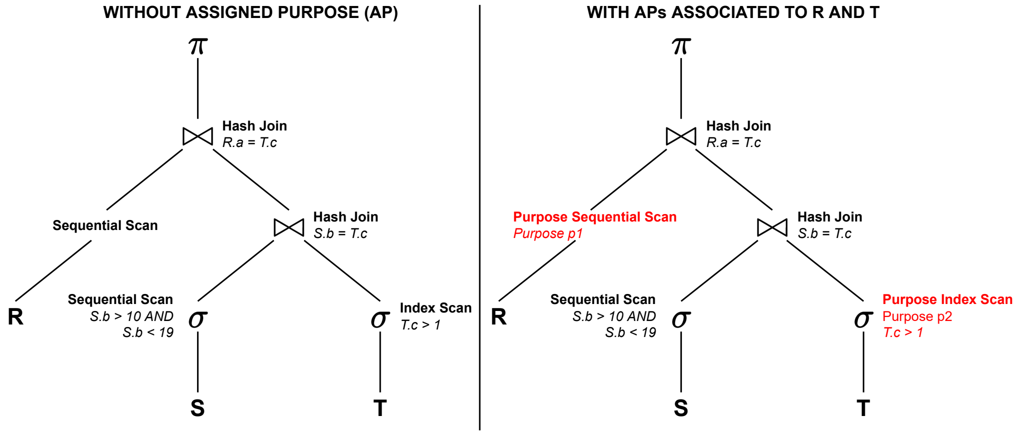

Figure 1. Two execution plans for the same query, with and without purposes.

5. Proposal

5.1. Purpose Scan

In this section, we briefly present our first contribution, Purpose Scan [Praciano et al. 2022]. To restrict the data used to generate the query response to only that data with consent, we added Purpose Scan to the list of available access methods. This new access method checks the permission of purposes associated with the data against query purposes before retrieving data. This prevents data not allowed by a specific user from being used during processing. In addition, it avoids the unnecessary cost of bringing it from disk to main memory.

During the evaluation of a query plan, Purpose Sequential Scan executes similarly to the full table scan, except that only the tuples that have consent for the purpose used in the query are retrieved from disk and passed on to the next operator. In other words, the tuples passed along have the respective purposes present in the query. On the other hand, the Purpose Index Scan performs similarly to the index scan, with the difference that, among the tuples selected by the index, only those that have the consent of the query's purpose will be retrieved and forwarded to the next operator. Consider there are three relations $R(a, b)$, $S(b, c)$, $T(c, d)$, and only relations R and T have sensitive data. Also consider the two execution plans shown in Figure 1 for the following query:

```
SELECT R.a FROM R, S, T WHERE R.a = T.c
AND S.b = T.c AND S.b > 10 AND S.b < 19
AND T.c > 1
```

On the left side of the Figure 1, we have a possible execution plan considering that the relations have no purposes. Hence, the access methods, such as sequential scan on R and S, are already known, while the index scan for T. On the right side, we show the execution plan for the query assuming that relation R is authorized for purpose $p1$ and T is authorized for $p2$. Thus, for these relations, the Purpose Sequential and Index Scan, respectively, are the chosen access methods to guarantee that only data that have $p1$ and $p2$ in relations R and T, respectively, are retrieved and forwarded to other operators. Finally, note that the method of accessing relation S remains the same as the previous one since that relation has no purpose, probably because its data is not sensitive.

5.2. Differential Purpose Scan

Now that Purpose Scan can enforce user consent, we present a motivating example illustrating the need to extend it to prevent information leakage. Suppose the table patient, Table 2, represents the result returned to José when executing the query `SELECT * FROM Patient`. Note that, since the `Result` column is protected, José only has access to the value in his own row. This column can be protected using *Purpose Scan*.

Table 2. Patient table example. Result column is protected.

ID	Name	Age	Diagnostic	Exam	Result	Treatment
1	João	35	Smoker with cancer	TCBD		Erlotinib
2	Maria	49	Diabetic	Blood glucose		Insulin
3	Morgana	18	Healthy	TCBD		None
4	José	22	Smoker with cancer	TCBD	Positive	Erlotinib

However, assume that the database has a *denial constraint*. If a patient is diagnosed as "Smoker with cancer" and their `Treatment` is "Erlotinib", then their `Result` must necessarily be "Positive". This implies that specific values in non-protected attributes may allow José to infer the protected value in the `Result` column, even though it is not directly visible, highlighting the need for an extended technique to prevent such inference-based leakage. We propose the Differential Purpose Scan, an extended version of the Purpose Scan that will incorporate differential privacy [Dwork 2006]. In this case, a differential privacy mechanism will be applied to the non-protected values to anonymize the data, thereby preventing the possibility of these inference attacks.

6. Preliminary Results

We now present the preliminary results obtained using Purpose Scan so far.

Setup. To build Purpose Scan, we used PostgreSQL version 9.6, adding it as an extension. Experiments were run in a Dell Power Edge machine with an Intel Xeon E5-2609 v3 1.9 GHz, 6 cores, running Ubuntu 18.04 and using a 7.2K RPM hard disk as storage technology. We used YCSB with factor $1,000$ as a benchmark through the OLTPBench. The benchmark was set up to run for 10 minutes on each run.

Effectiveness. In the first experiment, we observed how effective Purpose Scan is regarding the number of tuples retrieved to generate the final query result considering purpose-based access. In this scenario, the best outcome is that only allowed tuples are retrieved. Figure 2 shows the result obtained by the standard PostgreSQL to retrieve the tuples when varying the opt-in percentage between 10% to 100%. For 10% opt-in, we have that 10% of the retrieved tuples are allowed, while the other 90% are tuples that were retrieved unnecessarily. This situation occurs for both sequential scan and index scan. Note that up to 50% of opt-in, the amount of tuples retrieved and discarded is greater than that of retrieved and processed, which generates an unnecessary waste of resources. Obviously, for 100% opt-in, we have that PostgreSQL is total effective since all tuples are allowed.

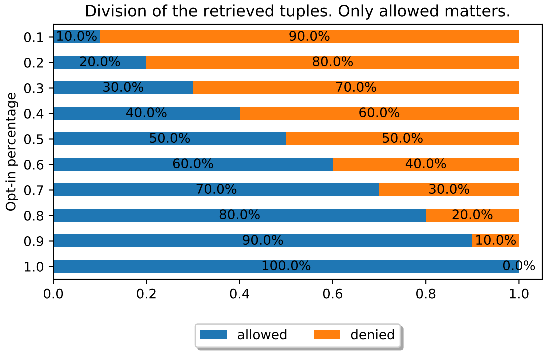

Figure 2. Stock PostgreSQL.

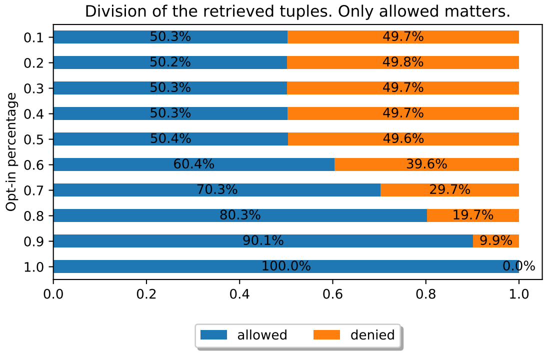

Figure 3. Purpose Scan.

Likewise, Figure 3 shows the result obtained by the Purpose Scan. To generate the result, we made the opt-in tuples spread across the pages with a probability of 50%, because when the opt-in tuples are clustered, Purpose Scan has full effectiveness, that is, only opt-in tuples are retrieved. Note that the Purpose Scan is more effective in retrieving tuples when the opt-in percentage varies between 0.1 and 0.5. Purpose Scan balances the number of tuples retrieved and discarded in these cases with those processed. Therefore, we can conclude that the more clustered the opt-in tuples are, the more effective the Purpose Sequential Scan will be. This same applies to Purpose Index Scan.

Efficiency. Now we compare how Purpose Scan fares against an alternative method of ensuring purpose-based access. To do this, we remove access to the relation in the baseline and create a view containing the same subset of tuples that the purpose-aware system returns, i.e., the result set is the same in Purpose Scan and the view. The terminals query the relation and filters in Purpose Scan and query only the view in the baseline. Table 3 presents the completed requests between Purpose Scan and the view-based approach. The results show that our strategy outperforms a more traditional view-based approach by more than 48%, showing that there is no trade-off between security and performance.

Table 3. Completed requests per terminal for each strategy

Technique	**Run 1**	**Run 2**	**Run 3**	**Average**
View-based	1202.0	1209.0	1196.0	1202.3
Purpose Scan	1783.0	1785.0	1790.0	1786.0

7. Conclusion and Next Steps

In this PhD work, we introduce Purpose Scan, a novel purpose-aware scan operator integrated into the query processing pipeline. It ensures that only consented personal data is retrieved. In this context, Purpose Scan outperforms a view-based access approach by almost 50% throughput, and has greater effectiveness in retrieving authorized tuples.

There is still work to be done to achieve purpose and privacy-aware query processing. As a next step, we plan to advance the development of Differential Purpose Scan by investigating how to prevent information leakage.

Acknowledgements

This research was partially funded by CAPES grant 88887.609129/2021 and CNPq/Brazil grant 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, Tokyo. IEEE Computer Society.

Amora, P., Praciano, F., and Machado, J. (2025). Purpose filter: A space-efficient purpose metadata storage. In *LNCS*, Lecture Notes in Computer Science.

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

Deshpande, A. (2021). Sypse: Privacy-first Data Management through Pseudonymization and Partitioning . In *CIDR*, pages 1–8, Online. www.cidrdb.org.

Dwork, C. (2006). Differential privacy. In *ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *LNCS*, pages 1–12. Springer.

Kabra, G., Ramamurthy, R., and Sudarshan, S. (2006). Redundancy and information leakage in fine-grained access control. In *ACM SIGMOD, Chicago, Illinois, USA, June 27-29*, pages 133–144. ACM.

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. SchengenDB: A data protection database proposal. In *VLDB 2019 Workshops, Los Angeles, CA, USA, August 30, 2019*, volume 11721 of *LNCS*.

Machado, J., Amora, P., and Praciano, F. (2024). Purpose and consent enforcement in dbms. In *SBBD*, pages 172–175. SBC.

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 *VLDB Workshops*, volume 13814 of *LNCS*.

Rizvi, S., Mendelzon, A. O., Sudarshan, S., and Roy, P. (2004). Extending query rewriting techniques for fine-grained access control. In *SIGMOD Conference*, pages 551–562, France. ACM.

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 *SBBD*. SBC.