Journal of Information and Data Management, 2026, 17:1, doi: 10.5753/jidm.2026.5859

Incorporating LGPD Requirements and Restrictions into Database Design

Patrícia Vieira [**Universidade Federal do Ceará** | ***patriciavieira@ufpi.edu.br***]
José Maria Monteiro [**Universidade Federal do Ceará** | ***jose.monteiro@lsbd.ufc.br***]
Javam Machado [**Universidade Federal do Ceará** | ***javam.machado@lsbd.ufc.br***]
Angelo Brayner [**Universidade Federal do Ceará** | ***brayner@dc.ufc.br***]

Computer Science Department, Universidade Federal do Ceará, Campus do Pici, Bloco 910, Pici, Fortaleza, CE, 60.440-900, Brazil.

Received: 07 April 2025 • **Published:** 13 March 2026

Abstract The Brazilian General Data Protection Law (LGPD) specifies how personal data processing, storage, and disposal should be conducted, conditioning it to the prior authorization of the data subject. On the other hand, current information systems rely heavily on personal data and, therefore, must comply with the LGPD. In this context, the database system becomes an even more critical component in software development, as it is responsible for storing, updating, and retrieving data. However, the methodologies and tools used for database design do not incorporate the requirements and constraints of the LGPD, making it difficult to ensure compliance between databases and current legislation. This article presents a methodology, called LGPDbyD, to incorporate the impositions and principles of the LGPD into the database design process. To achieve this, we extend the ER model, the Relational model, and the CREATE TABLE command. Additionally, we discuss how to model, design, and implement the concepts of purpose, consent, and personal data retention period. Finally, we extend the brModelo tool to provide support for the requirements and constraints of the LGPD. LGPDbyD aims to facilitate the processes of database design and auditing in compliance with the LGPD.

Keywords: LGPD, Data Protection, Database Design, Metadata

1 Introduction

The General Data Protection Law (Lei Geral de Proteção de Dados – LGPD), Law No. 13.709/18,[1] regulates how organizations may collect, process, store, and dispose of personal data, defined as information relating to an identified or identifiable natural person. The law aims to safeguard fundamental rights such as freedom and privacy. The LGPD represents a substantial shift in Brazil's data protection landscape, requiring comprehensive adjustments in organizational processes, documentation, and contractual arrangements. More importantly, it necessitates a cultural transformation in how companies handle personal data. The LGPD requires organizations to revise internal processes, update documentation and contractual instruments, and, most importantly, adopt a cultural shift in the way personal data is handled daily. The legislation introduces innovations not adequately addressed by previous sector-specific data protection laws in Brazil. Among its key contributions are a more precise definition of personal data, an explicit enumeration of the legal bases for data processing, specific provisions for the handling of publicly accessible data, the establishment of the National Data Protection Authority (ANPD), and the definition of applicable sanctions. Besides, the concepts of purpose and consent are fundamental to ensuring transparency and the informational self-determination right. Collectively, these innovations contribute to enhanced legal certainty for data subjects.

On the other hand, current Information Systems (ISs) are strongly based on the acquisition, storage, and processing of personal data. Logically, these systems need to comply with the LGPD. Thus, the LGPD has a major impact on the development of ISs, which must now treat personal data in the manner stipulated by the legislation, more formally, paying greater attention to the data life cycle, since this involves all operations performed on the information obtained by a company, from its collection to its proper destruction.

In this context, the database system (DBS) becomes an even more important component in software development since it is responsible for storing, updating, and retrieving data. More specifically, a database (DB), which represents a set of data and its interrelationships, must be compliant with the LGPD. Thus, for example, the result of an HIV test (which is used to diagnose an infection caused by the human immunodeficiency virus) should be stored in encrypted form.

One of the alternatives to ensure that a database complies with LGPD is to try to incorporate the requirements and restrictions imposed by the legislation into the database design process Kraska *et al*. [2019]; Schwarzkopf *et al*. [2019]; C. Machado and P. Amora [2021]; Wang *et al*. [2022]. However, database design is a complex activity involving four distinct phases: i) requirements analysis, ii) conceptual design, ii) logical design, and iv) physical design. In addition, these steps produce several artifacts, using different notations and often supported by different software tools. Furthermore, existing methodologies for database design do not incorporate the requirements and precepts brought by the LGPD.

[1] `https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm`

In our previous work da Silva Barros *et al*. [2024], we presented a strategy, called LGPDbyD, to incorporate the requirements and restrictions imposed by LGPD into database design. To this end, we adapted the ER model, the Relational model, and the CREATE TABLE command. In this paper, we have extended the LGPDbyD strategy to modeling, designing, and implementing the legal concepts of purpose, consent, and personal data retention period in database systems. Finally, we extended the brModelo tool to provide support for the methodology proposed in this paper. LGPDbyD aims to facilitate the processes of database design and auditing in compliance with LGPD.

2 Related Works

For educational purposes, this section is structured into two parts. First, we present studies that, in some way, aim to ensure compliance between information systems and the LGPD. Subsequently, we examine research efforts focused on extending or adapting database design methodologies for specific contexts.

2.1 Compliance between Information Systems and the LGPD

[Shastri *et al*., 2020], the authors seek to understand how SBDs can be affected by GDPR and provide a detailed analysis of typical DBS operations in light of GDPR (*General Data Protection Regulation*) requirements. In addition, the paper proposes a series of metrics to evaluate the performance of DBS in relation to the GDPR compliance, such as query execution time, resource consumption, and the effectiveness of anonymization and pseudonymization techniques. To validate the proposed approach, the authors conducted a series of experiments using different types of databases and workloads.

[Carauta Ribeiro and Dias Canedo, 2020], the authors described criteria and measures that guide the need for compliance with the LGPD in ICT (Information and Communication Technologies) processes at the University of Brasília (UnB). The research was applied to UnB's *software* systems Lachaud [2020]. Initially, the authors used the *Analytical Hierarchy Process* (AHP) method to perform a requirements analysis. Then, they applied the Preference Ranking Method for Enrichment Assessment (PROMETHEE) and the Multicriteria Decision Analysis (MCDA) process to enforce the standards according to the LGPD. In this work, the level of data protection, security risk, event severity, and data protection risk were defined as the main requirements for personal data security at UnB. Finally, the data protection risk criterion was established as the focus of UnB's LGPD implementation.

[Canedo *et al*., 2021], the authors discuss the process carried out to implement the LGPD in a Federal Public Administration Agency (FPAA) using the BPMN *(Business Process Model and Notation)* notation. They highlighted the need to define new roles and responsibilities within Brazilian public administration agencies at the federal level, since, with the new personal data protection legislation in force, each and every public body must adhere to this new Law.

[Araújo *et al*., 2021], the authors present a systematic mapping involving both the GDPR and the LGPD. Additionally, the authors propose a method, called LGPD4BP (LGPD for *Business Process*), to obtain the compliance of business processes in relation to the LGPD. The LGPD4BP consists of an evaluation questionnaire and a modeling method with a catalog of modeling patterns. The method was applied in a case study of the application college of the Federal University of Pernambuco (UFPE) and validated by a postgraduate class that applied the method and answered a questionnaire about the ease of use and completeness of the method. The results of the students' evaluations showed that the most difficult part was the modeling of the business process and not the components of the proposed method. The LGPD4BP method guides analysts to assess the compliance of business processes with the LGPD and guides analysts to model processes under this legislation. However, the LGPD4BP method does not specify who the actors, processing agents, and the certifying authority are; that is, it does not detail how the modeling was carried out.

The work proposed in [de Castro *et al*., 2022] aimed to develop a framework to support Information and Communication Technology (ICT) professionals in adapting companies to the requirements demanded by the LGPD. To achieve this purpose, a framework based on the BEST methodology (Business Engaged Security Transformation) was proposed. Besides, a survey was carried out to collect the perceptions of ICT practitioners in relation to adherence to LGPD adaptation actions by organizations. As a result, the authors identified a weakness in the privacy and information security management methodology implemented in organizations, which, in the future, may result in risks and damage to user information.

The study presented in [Rocha *et al*., 2023] aimed to identify the main obstacles that prevent software professionals from effectively ensuring compliance with the principles of the LGPD in their applications. Additionally, it proposes a reference guide designed to assist Information and Communication Technology (ICT) professionals in overcoming the challenges associated with implementing these principles. To this end, a survey was conducted with professionals working in various areas of software development to assess their level of compliance with the LGPD, based on their familiarity with implementation techniques within a hypothetical e-commerce scenario. The survey results revealed that all participants faced difficulties with at least one LGPD principle, with the majority of issues stemming from a lack of knowledge regarding appropriate implementation techniques. These findings highlight the need for software professionals to enhance their understanding of privacy-preserving techniques and LGPD-compliant practices—a need that the proposed reference guide seeks to address.

[Éllen Renner Ferrão *et al*., 2024], the authors proposed a comprehensive taxonomy of privacy requirements, drawing from the Brazilian General Data Protection Law (LGPD) and ISO/IEC 29100. This paper aims to assist software development teams in navigating the complexities of legal compliance. The taxonomy comprises 129 requirements, categorized into 10 distinct groups across 5 contexts.

2.2 Adaptations in Database Design

[Kamble, 2008] proposes a conceptual model for dealing with data that have multiple dimensions, offering a structure that allows the representation, organization, and analysis of these data, in addition to understanding how different dimensions relate to each other and how to extract information from these relationships. The work proposed in [Dani and Getta, 2005] presents a methodology and symbology for conceptual modeling focused on *Data Streams*, addressing the challenges of modeling in computing with continuous data flow, aiming to improve the efficiency and effectiveness of *Data Streams* processing in real time.

An extension to the Entity Relationship (ER) model is discussed in [Carvalho *et al*., 2023]. The authors present the ER+ conceptual model, which provides a more suitable framework for modeling distributed systems with multiple layers. Additionally, the authors discuss how this new extension deals with scalability and performance issues in distributed systems, with the inclusion of techniques to distribute the workload among the different nodes of the system, the optimization of communication between layers, and data consistency in a distributed environment.

[Khan *et al*., 2004] discusses the importance of integrating business requirements and constraints into database conceptual models, highlighting the importance of effective communication between database developers and business stakeholders, ensuring that business requirements are understood and correctly translated into the database model. In this way, organizations can develop systems that are more aligned with their specific needs, resulting in greater efficiency, flexibility, and user satisfaction.

[Sarkar and Athanassoulis, 2022], the authors present an extension for query languages that allows specifying data deletion policies. Thus, developers can define rules for automatic data deletion directly in an SQL command (usually in the INSERT clause) based on criteria such as the period the data remains stored. In this way, when inserting a certain "tuple", it is possible to define, for example, that it should be automatically removed after 5 years. This strategy is of fundamental importance in contexts where privacy and compliance with different laws are relevant concerns. In [de Abreu *et al*., 2021], the authors discuss how to ensure that query processing in databases respects user consent. The proposed solution is based on the development of extensions of the SQL language that aim to incorporate consent considerations during query processing, allowing developers to explicitly express in SQL commands the conditions under which data can be accessed and used.

The work presented in [Mok, 2024] proposed a MongoDB design methodology grounded in conceptual modeling. Given a query over a MongoDB database, it identifies a set of hierarchical data storage configurations—also referred to as scheme trees—derived from the conceptual model of the database, with the goal of minimizing query retrieval time. Scheme trees represent a logical design for the MongoDB database, which is subsequently implemented as MongoDB collections. These collections define the physical data layout within the database. To evaluate the proposed methodology, a COVID-19 dataset was employed as a case study.

3 The Proposed Strategy: LGPDbyD

This article proposes a strategy, called LGPDbyD, which aims to integrate the requirements and constraints established by the Brazilian General Data Protection Law (LGPD) into the database design process. To this end, the proposed strategy introduces minimal modifications to the concepts and notations traditionally employed in the conceptual, logical, and physical stages of database design. The central idea is to change the existing models as little as possible.

It is worth noting that, in general, database design is supported by CASE tools (Computer-Aided Software Engineering)[2]. The advantages offered by CASE tools are numerous, among which the following stand out: the ability to provide user feedback in response to changing requirements, increased productivity in software development, reduced development time, improved system quality, standardization, the potential to replace human resources in projects, and the capacity to address large and complex problems [Favero, 2019].

There is a wide range of commercial tools that assist in database design. Among the most well-known are ERWin and DBDesign. However, these tools provide support only for the logical and physical design stages, offering no assistance for conceptual modeling. In contrast, the brModelo tool[3] supports conceptual, logical, and physical database design. It is widely used in educational settings to teach and practice database design concepts. brModelo offers a user-friendly graphical interface that simplifies the creation and editing of models, making it a valuable tool for students, instructors, and information technology professionals alike [dos Santos Mello *et al*., 2021].

In this context, a second contribution of this article is an extension of the brModelo tool, called brModeloPD, which incorporates the notations of the LGPDbyD methodology, covering the conceptual, logical, and physical design stages. The brModeloPD tool is accessible via the web[4], and its source code is freely available online[5].

3.1 Conceptual Design

Initially, we propose an adaptation of the Entity-Relationship (ER) model, referred to as ER-PD, with the goal of enabling the conceptual design of databases in compliance with the Brazilian General Data Protection Law (LGPD). This extension allows the representation of key concepts defined by the LGPD, such as: personal data, types of data processing operations, and the data subject.

The **Personal Data Subject**, as specified in Article 5 of the LGPD, refers to the individual to whom the law aims to provide protection. Therefore, the **Personal Data Subject** is a central concept in the ER-PD model and is represented as a special type of entity set, which indicates the presence of personal attributes that must be given particular protection. The notation used to represent this specific type of entity set, called "Owner", is a **rectangle with dashed lines** (Figure 1).

[2] `https://www.iso.org/standard/43189.html`
[3] `https://docs.brmodeloweb.com/#/logical-model/README`
[4] http://34.29.8.6:9000/
[5] https://github.com/jmmfilho/lgpdbyd

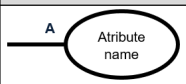
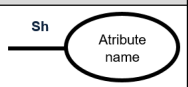
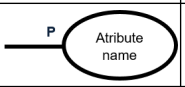
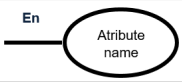
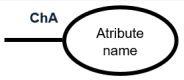
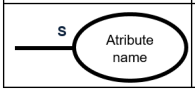
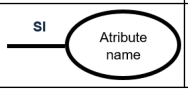
Symbol	Reprentation	Symbol	Reprentation	Symbol	Reprentation
	Owner Entity		Anonymized Attribute		Shared Attribute
	Personal Attribute		Encrypted Attribute		Child and Adolescent Attribute
	Sensitive Attribute		Identifier Attribute		Semi-Identifier Attribute

Figure 1. ER-PD Model Notation.

According to LGPD, some personal data are considered "sensitive" and must be treated specifically, as highlighted in Article 11. Also, according to the LGPD, one of the ways to treat sensitive data is through anonymization. Encryption is not mentioned at any time in the LGPD, but it is one of the alternatives commonly used to ensure data anonymization. To represent the fact that an attribute stores personal data, as well as the treatment that must be performed on this data, the ER-PD model proposes the use of 8 new types of attributes (Figure 1), the main ones being: "Personal" attribute (P), "Sensitive" attribute (S), "Anonymized" attribute (A) and "Encrypted" attribute (En). In addition, the ER-PD model introduces two further attribute types: "Identifier" attribute (I) and "Semi-Identifier" attribute (SI), to represent concepts commonly used in the data privacy domain. An **Identifier** attribute is one that can uniquely identify an individual—such as CPF (Brazilian tax ID), name, or email. A **Semi-Identifier** attribute, on the other hand, does not explicitly identify an individual on its own but can do so when combined with other attributes [Brito and Machado, 2017]. Examples include birth date and ZIP code. It is important to note that, in general, when the privacy of personal or sensitive attributes needs to be preserved, anonymization or encryption techniques are applied.

The LGPD establishes specific rules for the processing of data related to **Children and Adolescents**, in Article 14, and its paragraphs. To represent this characteristic, the ER-PD model adds a new type of attribute called "Child and Adolescent" (ChA). Finally, the data subject may authorize the sharing of their data, or portions thereof, with third parties. To represent this requirement, the ER-PD model proposes a new attribute type, the "Shared" (Sh) attribute. Figure 1 illustrates the notations introduced by the ER-PD model.

3.1.1 Running Example

Next, we will illustrate the use of the ER-PD model in the conceptual design of databases compliant with the LGPD. Initially, consider that a medical clinic wants to design a database to store information about patients and medical tests. Assume that a patient can perform zero or more tests and that a test can be performed by zero or more patients.

For patients, the following data should be stored: cod-patient, CPF, name, birthday, address, ethnicity, religion, sex, and gender. It is important to note that all these attributes are classified as personal data. Furthermore, assume that the clinic's "Data Controller" has defined that the attributes cod-patient and CPF must be encrypted, and that name and birthday must be anonymized. According to the "Data Controller", the attributes ethnicity, religion, sex, and gender are considered sensitive personal data, and therefore require special attention; however, no specific data processing technique has been defined for them. Additionally, the attribute address is considered both personal data and a semi-identifier, although no special processing has been specified in this case either. For medical tests (exams), the attributes to be stored include: cod-exam, description, and cost. None of these attributes are classified as personal data. Additionally, the relationship between patient and exam includes two attributes: date-exam and result. The "Data Controller" specified that the result attribute, which qualifies as personal data, must be encrypted, and that date-exam is a semi-identifier for which no specific processing was defined.

Figure 2 illustrates the conceptual schema produced during the conceptual design phase, using the ER-PD model, as modeled through the brModeloPD tool. Note that the entity set Patient is represented by a rectangle with dashed lines, indicating that it corresponds to a Personal Data Subject (Owner). Also observe that, next to each attribute name, the model indicates—between brackets—the type of data and the required processing technique, when applicable.

3.2 Logical Design

The next stage in the database design process is called logical design, and its purpose is to produce a logical schema for the database, using as input the conceptual schema developed during the conceptual design. The logical schema is expressed using a data model supported by a database management system, generically referred to as a logical model. The most commonly used logical model is the relational model, and the schema is typically represented by a **relational diagram (RD)**.

In this work, we propose an adaptation of the Relational model, called R-PD, to enable the logical design of databases in compliance with the LGPD. The R-PD model allows representing the key concepts defined by the LGPD, such as personal data, types of data processing operations, and the data subject. Additionally, the R-PD model supports the representation of attributes concerning children and adolescents.

Figure 3 illustrates the notation adopted in the R-PD model. Note that a relation (table) representing a personal data subject is symbolized by a rectangle with dotted lines. Moreover, next to each attribute name, enclosed in brackets,

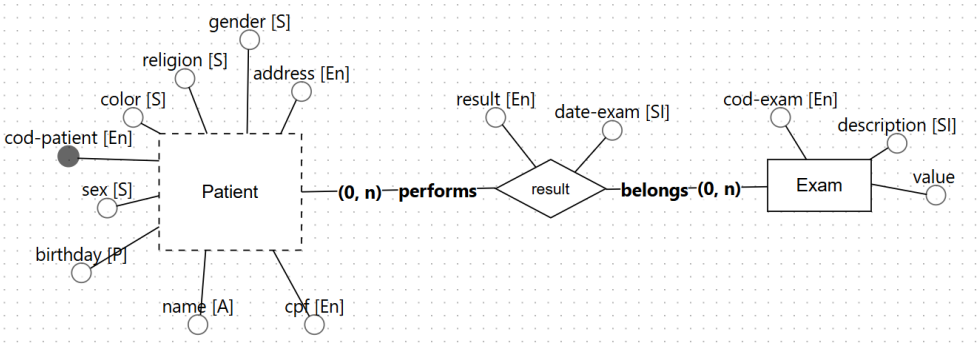


Figure 2. Conceptual Scheme Using the brModeloPD Tool.

the model highlights both the type of data and the processing technique to be applied.

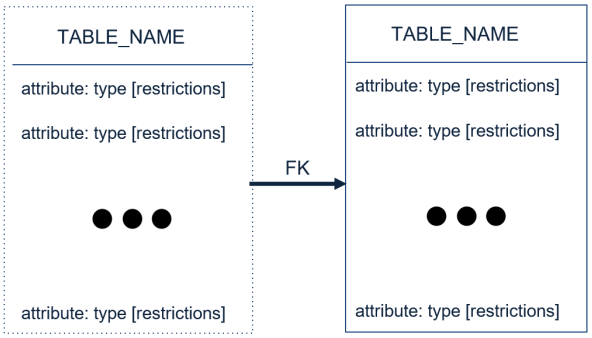


Figure 3. R-PD Model Notation.

3.2.1 Running Example

Next, we will illustrate the use of the R-PD model in the logical design of databases that comply with the LGPD. To do this, we will consider the same context described in the previous section, involving a medical testing clinic that wants to store information about patients and medical tests.

Initially, we will map the conceptual schema to the logical schema. In this process, each entity set present in the conceptual schema will be mapped to a relation (table) in the logical schema. Next, we map the attributes, specifying for each one its data type and associated constraints defining, for each attribute, its data type and its restrictions (e.g., *NOT NULL*, *UNIQUE*, *Primary Key - PK* and *Foreign Key - FK*). Subsequently, we map the LGPD-specific concepts for each attribute.

Table 1 presents the attributes of the "Patient" relation in tabular form, while Table 2 shows the attributes of the "Exam" relation. Note that the attribute "cod-patient" of the "Patient" relation is classified as both personal and sensitive data. For this reason, the value S is assigned to both the **P** (Personal) and **S** (Sensitive) columns. Furthermore, it was specified that the "patient-code" attribute should be encrypted, which justifies the use of the value **S** in the columns **A** (Anonymized) and **En** (Encrypted). Since **En** is a more restrictive type of data protection, his column is highlighted in bold red for emphasis.

The next step in translating the conceptual schema into the logical schema involves the mapping of "Relationship Sets". It is well established that any N:M "Relationship Set" is represented in the relational model by a new relation. Accordingly, a new relation named "Result" will be created to represent the "Result" "Relationship Set". Table 3 presents the attributes of the "Result" relation in tabular format. Note that the "cod-exam" attribute from the "Exam" relation does not constitute personal data. However, it will be encrypted since this attribute will be part of the primary key of the "Result" relation, which includes the result of a certain examination (test) performed by a specific patient. This result is classified as both personal and sensitive data and must therefore also be encrypted.

Figure 4 illustrates the logical scheme for the previously described running example, modeled using the brModeloPD tool. Note that the "Patient" relation is represented by a rectangle with dotted lines, as it is a "Data Subject" (e.g.,"Data Holder" or "Data Owner").

3.3 Physical Design

The final phase of the database design process is known as the physical design. This phase takes the logical schema, described using the R-PD model, as input and produces the physical schema of the database as output. The physical schema specifies the file organization methods and the internal storage structures to be used. The physical schema will be represented through a collection of DDL (*Data Definition Language*) commands.

In this paper, we propose an adaptation of the SQL CREATE TABLE command, called SQL-PD, to enable the physical design of databases in compliance with the LGPD. This adaptation allows the representation of the main concepts present in the LGPD based on metadata (comments in an SQL command). This metadata can be used for LGPD compliance audits. The grammar of the adapted CREATE TABLE command (SQL-PD) is shown in the Listing 1. Please note that 8 new restriction types have been added: PERSONAL, SENSITIVE, ANONYMIZED, ENCRYPTED, IDENTIFIER and SEMI-IDENTIFIER, SHARED, and CHILD AND ADOLESCENT.

Table 1. Tabular Representation of the Patient Relation.

Attribute	Type	NOT NULL	UNIQUE	PK	FK	P	S	A	En	Sh	ChA	I	SI
cod-patient	integer	Y	Y	Y	N	Y	Y	Y	**Y**	N	N	Y	N
cpf	varchar	Y	Y	N	N	Y	Y	Y	**Y**	N	N	S	N
name	varchar	Y	N	N	N	Y	Y	**Y**	N	N	**Y**	N	**Y**
address	varchar	N	N	N	N	Y	Y	**Y**	N	N	N	**Y**	N
birthday	date	N	N	N	N	Y	**Y**	N	N	N	N	Y	Y
sex	char	N	N	N	N	Y	**Y**	N	N	N	N	N	N
color	char	N	N	N	N	Y	**Y**	N	N	N	N	N	N
religion	varchar	N	N	N	N	Y	**Y**	N	N	N	N	N	N
gender	varchar	N	N	N	N	Y	Y	**Y**	N	N	N	N	N

Table 2. Tabular Representation of the Exam Relation.

Attribute	Type	NOT NULL	UNIQUE	PK	FK	P	S	A	En	Sh	ChA	I	SI
cod-exam	integer	Y	Y	Y	N	Y	Y	Y	**Y**	N	N	Y	N
description	varchar	Y	Y	N	N	Y	Y	Y	**Y**	N	N	S	N
value	varchar	Y	N	N	N	Y	Y	**Y**	N	N	**Y**	N	**Y**

Table 3. Tabular Representation of the Result Relation.

Attribute	Type	NOT NULL	UNIQUE	PK	FK	P	S	A	En	Sh	ChA	I	SI
cod-patient	integer	Y	Y	Y	Y	Y	Y	Y	**Y**	**Y**	N	**Y**	N
cod-exam	integer	Y	Y	Y	Y	N	N	Y	**Y**	N	N	**Y**	N
result	varchar	N	N	N	N	Y	Y	Y	**Y**	N	N	N	N
date-exam	date	N	N	Y	N	N	N	N	N	N	N	N	**Y**

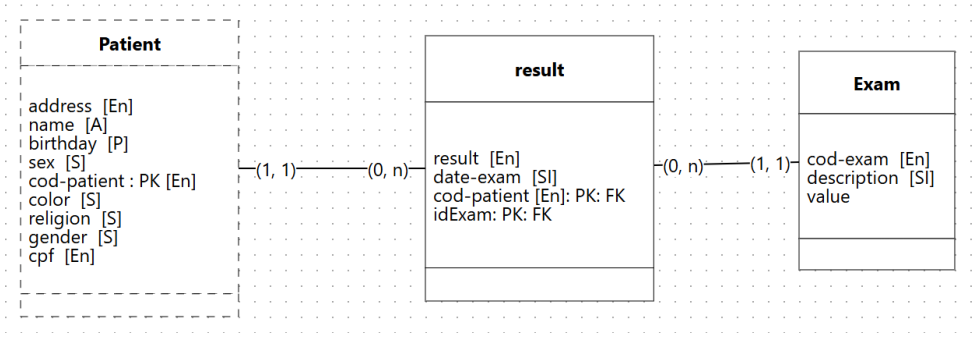


Figure 4. Logical Diagram Using the brModeloPD Tool.

Listing 1: Extended CREATE TABLE Command

```
CREATE TABLE table-name
        (column-name data-type [not null],
         [column-name data-type [not null] ... ],
         [CONSTRAINT name-restriction]
                UNIQUE column-name
                        | PRIMARY KEY(column-name {, column-name})
                        | FOREIGN KEY (column-name {, column-name})
                REFERENCES table-name
                        [ON DELETE CASCADE |
                                SET NULL | NO ACTION ],
                        [ON UPDATE CASCADE],
                        | CHECK (predicate)
                          | PERSONAL column-name
                          | SENSITIVE column-name
                          | ANONYMIZED column-name
                          | ENCRYPTED column-name
                          | SHARED column-name
                          | IDENTIFIER column-name
                          | SEMI IDENTIFIER column-name
)
```

3.3.1 Running Example

Next, we will illustrate the use of the SQL-PD extension in the physical design of LGPD-compliant databases. For this purpose, we will consider the same running example described in the previous section, involving a medical clinic that intends to store information about patients and exams. Listings 2, 3, and 4 present the CREATE TABLE commands generated by the brModeloPD tool for the "Patient", "Exam", and "Result" tables, respectively. Note that the LGPD-related constraints are embedded in the DDL statements through comments (following PostgreSQL syntax).

Listing 2: CREATE TABLE Patient Command (SQL-PD)

```
CREATE TABLE Patient
        (cod-patient integer NOT NULL,
        cpf char NOT NULL,
        name varchar NULL,
        address varchar NULL,
        birthday date NULL,
        sex char NULL,
        color char NULL,
        religion char NULL,
        gender varchar NULL,
        CONSTRAINT c1 PRIMARY KEY cod-patient
        /* , */
        /* CONSTRAINT c2 Encrypted cod-patient , */
        /* CONSTRAINT c3 Encrypted cpf , */
        /* CONSTRAINT c4 Sensitive  sex , */
        /* CONSTRAINT c5 Sensitive color , */
        /* CONSTRAINT c6 Sensitive religion , */
        /* CONSTRAINT c7 Sensitive gender , */
        /* CONSTRAINT c8 Sensitive birthday , */
        /* CONSTRAINT c9 Anonymized name, */
        /* CONSTRAINT c10 Anonymized address */
    )
```

Listing 3: CREATE TABLE Exam Command (SQL-PD)

```
CREATE TABLE Exam
       (cod-exam integer NOT NULL,
         description varchar,
         value numeric,
         CONSTRAINT c1 PRIMARY KEY cod-exam
         /* , */
         /* CONSTRAINT c2 Encrypted cod-exam */
    )
```

Listing 4: CREATE TABLE Command Result (SQL-PD)

```
CREATE TABLE Result
       (cod-patient integer NOT NULL,
         cod-exam integer NOT NULL,
         date-exam date,
         result varchar,
         CONSTRAINT c1 PRIMARY KEY cod-patient, cod-exam, date-exam
         CONSTRAINT c2 FOREING KEY cod-patient REFERENCES Patient (cod-patient),
         CONSTRAINT c3 FOREING KEY date-exam REFERENCES
         Exam (date-exam)
         /* , */
         /* CONSTRAINT c4 Encrypted cod-patient, */
         /* CONSTRAINT c5 Encrypted result */

    )
```

3.4 Modeling Purpose and Consent

The Brazilian General Data Protection Law (LGPD) establishes data ownership rights for individuals and delineates specific criteria governing data storage, processing, publication, and sharing. One of the fundamental pillars of data protection is the guarantee granted to the data subject of the right to determine the circumstances under which their personal data may be accessed, processed, or shared. This right is referred to as consent (or right of consent). According to the LGPD, consent is defined as an unambiguous manifestation whereby the data subject agrees to the processing of their personal data for a given purpose. Furthermore, the data subject is also entitled to determine the duration for which their data may be utilized. Thus, two new concepts must be modeled: purpose and consent Konstantinidis *et al*. [2021]. In order to represent the concept of consent, as defined by the LGPD, the LGPDByD strategy employs an extension of the SQL language.

3.4.1 Modeling the Concept of Purpose

To represent the concept of purpose, it is essentially necessary to define its name. Additionally, the description of a purpose can be enriched by means of a comment. Finally, it is assumed that hierarchical relationships may exist between purposes, whereby a given purpose *p2* may be a child (or inherit from) a higher-level purpose *p1*. Accordingly, if a given data subject consents to the use of their data for purpose *P1*, the data may also be accessed for purpose *P2*. However, the inverse does not hold true; that is, if a patient consents to the use of their data for purpose *P2*, this does not automatically imply that the data may be accessed for purpose *P1*. To illustrate the use of the purpose concept, consider the following purposes:

- *P1*: Diagnostic Support.
- *P2*: Triage Support in Patient Care.
- *P3*: Comparative Clinical Studies.

To enable the creation of purposes, the LGPDByD strategy proposes the introduction of a new SQL command called CREATE PURPOSE, whose syntax is presented in Listing 5:

Listing 5: CREATE PURPOSE Command Syntax

```
CREATE PURPOSE NOME [PARENT [=] NOME]
  [COMMENT [=] STRING];
```

Listing 6 illustrates the creation of purposes P1, P2, and p3, as previously mentioned. Table 4 presents the purposes metadata in tabular form.

Listing 6: Creating Purposes P1, P2 and P3

```
CREATE PURPOSE P1;
CREATE PURPOSE P2 PARENT P1;
CREATE PURPOSE P3 [COMMENT = Clinical studies to evaluate the use of
coconut water in the treatment of stress.];
```

3.4.2 Default Consent of Relations

The LGPDByD strategy proposes the use of a default consent for each relation, which can be of two distinct types: **prohibitive** or **permissive**. If the default consent for a given relation *r* is of the prohibitive type, any access to the data in that relation is, by default, denied and may only be granted if there is explicit consent authorizing its use. Conversely, if the default consent for a relation *r* is of the permissive type, its data may, by default, be accessed freely and will only be restricted if there exists a "consent" explicitly denying its use. The default consent is implemented in the LGPDByD strategy through two new types of constraints: **PROHIBITIVE CONSENT** and **PERMISSIVE CONSENT**.

Continuing with the running example under discussion, let us assume that we wish to specify the default consent for the *Patient* table as *prohibitive*, while the default consent for the *Result* table is to be *permissive*. The Listings 7 and 8 illustrate how to add the corresponding constraints to specify the default consent for the Patient and Result relations.

Listing 7: Default Consent of Patient Relation

```
ALTER TABLE Patient
      ADD CONSTRAINT CD1 PROHIBITIVE CONSENT
```

Listing 8: Default Consent of Exam Relation

```
ALTER TABLE Result
      ADD CONSTRAINT CD2 PERMISSIVE CONSENT
```

In this manner, the data in the *Patient* table may only be accessed if there is explicit consent authorizing its use. In contrast, the data in the *Result* table may be accessed freely by default and will only have their access restricted if there is explicit "consent" denying their use.

3.4.3 Modelling the Concept of Consent

Before presenting how the LGPDByD strategy models the concept of consent, let us consider an example of an instance of the *Patient* relation, as illustrated in Table 5.

Indeed, each patient has the right to determine the circumstances under which their data may or may not be accessed. To enable the creation of purposes, the LGPDByD strategy proposes the introduction of a new SQL command called CREATE CONSENT, whose syntax is presented in Listing 9.

Listing 9: CREATE CONSENT Command Syntax

```
CREATE CONSENT NOME
 AS (ALLOW/DENY) PURPOSE SET
 TO TABLE_LIST [(COLUM_LIST)] [WHERE P(X)];
```

To illustrate the creation and use of the proposed approach for consent management, consider that Amy consents to the use of all her personal data for purposes P1, P2, and P3. In this case, it is necessary to create a new consent record. The command for creating this consent is shown in Listing 10.

Table 4. Tabular Representation of the Purposes Metadata.

Name	Parent	Comment
P1	-	-
P2	P1	-
P3	-	Clinical studies to evaluate the use of coconut water in the treatment of stress

Table 5. Illustrative Instance of the Patient Relation.

cod-patient	cpf	name	address	birthday	sex	color	religion	gender
1	123.456.789-00	Amy	123 St	80/10/30	F	White	Christian	Female
2	234.567.890-01	Anna	234 St	81/09/30	F	Black	Muslim	Queer
3	456.789.012-34	Eva	456 St	82/08/30	F	Asian	Hindu	Female
4	678.901.234-56	Ben	678 St	83/07/30	M	White	Jewish	Male
5	891.234.567-89	Eric	891 St	84/06/30	M	Black	Atheist	Male

Listing 10: Creating Consent C1

```
CREATE CONSENT C1
 AS ALLOW PURPOSE P1, P2, P3
 TO Patient
 WHERE cod-patient = 1;
```

Additionally, consider that Anna consents to the use of all her data for purposes P1 and P2, but not for P3. In this case, it is necessary to create a new consent record. The command for creating this consent is shown in Listing 11.

Listing 11: Creating Consent C2

```
CREATE CONSENT C2
 AS ALLOW PURPOSE P1, P2
 TO Patient
 WHERE cod-patient = 2;
```

Furthermore, assume that Eva consents to the use of only the attributes name, birthday, sex, and color, exclusively for purpose P2. In this case, it is necessary to create a new consent record. The command for creating this consent is shown in Listing 12.

Listing 12: Creating Consent C3

```
CREATE CONSENT C3
 AS ALLOW PURPOSE P2
 TO Patient (name, birthday, sex, color)
 WHERE cod-patient = 3;
```

Now, consider that Ben consents to the use of the attributes name, birthday, sex, and color for purpose P2. He also consents to the use of the attributes name, birthday, and sex for purpose P1. In this case, two separate consent records are required to accurately represent Ben's consent preferences. The commands for creating these consents are shown in Listing 13 and Listing 14.

Listing 13: Creating Consent C4

```
CREATE CONSENT C4
AS ALLOW PURPOSE P2
TO Patient (name, birthday, sex, color)
WHERE cod-patient = 4;
```

Listing 14: Creating Consent C5

```
CREATE CONSENT C5
AS ALLOW PURPOSE P1
TO Patient (name, birthday, sex)
WHERE cod-patient = 4;
```

Finally, consider that Eric has not provided any explicit consent. In this case, no consent record needs to be created. Table 6 presents the consents metadata in tabular form.

3.4.4 The Concept of Role

As we can observe, the proposed strategy results in the creation of a large number of consent records. Therefore, to facilitate the management and use of these consents, we propose the concept of "Role", which is essentially a grouping of consents. A user who is granted permission to use a given Role **R** automatically gains access to all the consent records associated with it. To enable the creation of Roles, the LGPDByD strategy proposes the use of the GRANT CONSENT command. Suppose we want to create a "Role" **R1** that associates the previously created consent records C1, C2, C3, C4, and C5. Listing 15 shows the command to create **R1**. Table 7 presents the consents metadata in tabular form.

Listing 15: Creating Role R1

```
GRANT CONSENT C1, C2, C3, C4, C5 TO ROLE R1
```

Once the Roles have been created, the next step is to associate database system users with the previously defined Roles. The next step consists of associating the database system users with the previously defined Roles. To do this, the LGPDByD strategy proposes the use of the GRANT USER command. Suppose we want to associate the user **USER1** with the "Role" **R1**. Listing 16 shows the command to associate the user **USER1** with the "Role" **R1**. Table 8 presents the metadata used to associate users to roles.

Listing 16: Associating the user **USER1** with the Role **R1**

```
GRANT USER1 TO R1
```

3.4.5 Query Execution in the LGPDByD Framework

In the LGPDByD strategy, when requesting the execution of an SQL query, the database user must specify the purpose for which the query result will be used. In this way, every SQL query execution request will always be associated with one or more purposes.

To illustrate the SQL query execution process according to the LGPDByD strategy, suppose that the user USER1 has requested the execution of query Q1 for Purpose P1. The query *Q1* is illustrated in Listing 17.

Listing 17: SQL Query Q1

```
SELECT *
FROM Patient
```

Table 6. Tabular Representation of the Consents Metadata.

name	purpose set	table	columns	type	condition
C1	P1, P2, P3	Patient	All	Allow	WHERE cod-patient = 1
C2	P1, P2	Patient	All	Allow	WHERE cod-patient = 2
C3	P2	Patient	name, birthday, sex, color	Allow	WHERE cod-patient = 3
C4	P2	Patient	name, birthday, sex, color	Allow	WHERE cod-patient = 4
C5	P1	Patient	name, birthday, sex	Allow	WHERE cod-patient = 4

Table 7. Tabular Representation of the Roles Metadata.

Name	ConsentSet
R1	C1, C2, C3, C4, C5

Table 8. Metadata Used to Associate Users to Roles.

User	Role
User1	R1

Note that the query Q1 must be rewritten by the query processor in order to return only the data that has been authorized by patients for the purpose P1. The implementation of the query rewriting mechanism is beyond the scope of this work. However, for illustrative purposes, the rewritten query is shown in Listing 18, and the returned result is presented in Table 9.

Listing 18: Rewritten Query Q1 for the Purpose P1

```
SELECT *
FROM Patient
WHERE cod-patient = 1 OR cod-patient = 2
UNION
SELECT NULL, NULL, name, NULL, birthday, sex, NULL, NULL, NULL
FROM Patient
WHERE cod-patient = 4
```

Now, suppose that the user USER1 has requested the execution of query Q1 for Purpose P2. The query *Q1* is illustrated in Listing 17. The rewritten query is shown in Listing 19, and the returned result is presented in Table 10.

Listing 19: Rewritten Query Q1 for the Purpose P2

```
SELECT *
  FROM Patient
  WHERE cod-patient = 1 OR cod-patient = 2
UNION
  SELECT NULL, NULL, NAME, NULL, birthday, sex, NULL, NULL, NULL
  FROM Patient
  WHERE cod-patient = 4
```

Finally, suppose that the user USER1 has requested the execution of query Q1 for Purpose P3. The query *Q1* is illustrated in Listing 17. The rewritten query is shown in Listing 20, and the returned result is presented in Table 11.

Listing 20: Rewritten Query Q1 for the Purpose P3

```
SELECT *
FROM Patient
WHERE cod-patient = 1
```

We will now discuss a few examples regarding the use of the Results relation. It is important to recall that the default consent policy for the result relation is of the permissive type. Thus, its data may, by default, be accessed freely and will only be restricted if there exists an explicit consent denying its use.

To begin with, let us consider that Amy provides consent for the use of her medical test results for purposes *P1*, *P2*, and *P3*. In this case, given that the default consent for the Result relation is of the permissive type, no explicit consent record is required. Conversely, Anna authorizes the use of her test results for purposes *P1* and *P2*, but explicitly withholds consent for *P3*. Therefore, a consent record of type deny must be created, as shown in Listing 21.

Listing 21: Creating Consent C6

```
CREATE CONSENT C6
AS DENY PURPOSE P3
TO Result
WHERE code-patient = 2;
```

Now, assume that Eva provides consent for the use of her medical test data for purposes *P1* and *P2*, with the exception of the result attribute. However, she does not consent to the use of her data for purpose *P3*. In this case, in order to accurately represent Eva's consent preferences, two distinct consent records must be created. These consents are presented in Listings 22 and 23.

Listing 22: Creating Consent C7

```
CREATE CONSENT C7
  AS DENY PURPOSE P1, P2
  TO Result (result)
  WHERE cod-patient = 3;
```

Listing 23: Creating Consent C8

```
CREATE CONSENT C8
 AS DENY PURPOSE P3
 TO Result
 WHERE cod-patient = 3;
```

Furthermore, assume that Ben does not provide consent for the use of his medical test data for purposes *P1*, *P2*, or *P3*. In this case, a new consent record must be created. This consent is shown in Listing 24.

Listing 24: Creating Consent C9

```
CREATE CONSENT C9
 AS DENY PURPOSE P1, P2, P3
 TO Result
 WHERE cod-patient = 4;
```

Finally, consider that Eric provides consent for the use of his medical test results for purposes *P1*, *P2*, and *P3*. In this case, since the default consent model for the Result relation is *permissive*, no explicit consent record needs to be created. Table 12 presents the consents metadata updated and in tabular form.

Table 9. Output of the Rewritten Query Q1 for the Purpose P1.

cod-patient	cpf	name	address	birthday	sex	color	religion	gender
1	123.456.789-00	Amy	123 St	80/10/30	F	White	Christian	Female
2	234.567.890-01	Anna	234 St	81/09/30	F	Black	Muslim	Queer
NULL	NULL	Ben	NULL	83/07/30	M	NULL	NULL	NULL

Table 10. Output of the Rewritten Query Q1 for the Purpose P2.

cod-patient	cpf	name	address	birthday	sex	color	religion	gender
1	123.456.789-00	Amy	123 St	80/10/30	F	White	Christian	Female
2	234.567.890-01	Anna	234 St	81/09/30	F	Black	Muslim	Queer
NULL	NULL	Eva	NULL	82/08/30	F	Asian	NULL	NULL
NULL	NULL	Ben	NULL	83/07/30	M	White	NULL	NULL

Table 11. Output of the Rewritten Query Q1 for the Purpose P3.

cod-patient	cpf	name	address	birthday	sex	color	religion	gender
1	123.456.789-00	Amy	123 St	80/10/30	F	White	Christian	Female

Table 12. Tabular Representation of the Consents Metadata (Updated Version).

Name	Purpose Set	Table	Columns	Type	Condition
C1	P1, P2, P3	Patient	All	Allow	WHERE cod-patient = 1
C2	P1, P2	Patient	All	Allow	WHERE cod-patient = 2
C3	P2	Patient	name, birthday, sex, color	Allow	WHERE cod-patient = 3
C4	P2	Patient	name, birthday, sex, color	Allow	WHERE cod-patient = 4
C5	P1	Patient	name, birthday, sex	Allow	WHERE cod-patient = 4
C6	P3	Result	All	Deny	WHERE cod-patient = 2
C7	P1, P2	Result	Result	Deny	WHERE cod-patient = 3
C8	P3	Result	All	Deny	WHERE cod-patient = 3
C9	P1, P2, P3	Result	All	Deny	WHERE cod-patient = 4

Before discussing additional query execution examples, consider that we have modified Role R1 to include purposes C6, C7, C8, and C9. To achieve this, we executed the command shown in Listing 25. Note that USER1 had already been associated with Role R1, and therefore, no further association is required. Besides, let us consider an example of an instance of the *Result* relation, as illustrated in Table 13.

Listing 25: Updating Role R1

```
GRANT CONSENT C1, C2, C3, C4, C5, C6, C7, C9 TO ROLE R1
```

In order to illustrate the SQL query execution process according to the LGPDByD strategy, suppose that the user USER1 has requested the execution of query Q2 for Purpose P1. The query *Q2* is illustrated in Listing 26.

Listing 26: SQL Query Q2

```
SELECT *
FROM Result
```

Note that the query Q2 must be rewritten by the query processor in order to return only the data that has been authorized by patients for the purpose P1. The rewritten query is shown in Listing 27, and the returned result is presented in Table 14.

Listing 27: Rewritten Query Q2 for the Purpose P1

```
SELECT *
FROM Result
EXCEPT
SELECT *
FROM result
WHERE cod-patient = 4 OR cod-patient = 3
UNION
SELECT cod-patient, cod-exam, date-exam, NULL
FROM esult
WHERE cod-patient = 3
```

3.4.6 Modeling the Duration of Personal Data Storage

The Brazilian General Data Protection Law (Lei Geral de Proteção de Dados Pessoais – LGPD), Law No. 13.709/2018, does not establish a fixed and universal period for the storage of personal data. Instead, it sets forth guiding principles and conditions for data retention, particularly with respect to purpose, necessity, and adequacy. Nevertheless, according to the LGPD, personal data must be retained only for the period necessary to fulfill the purpose for which it was collected. The legal basis for this requirement is found in Article 15 of the LGPD, which states:

Art. 15. The processing of personal data shall be terminated in the following circumstances:

I. upon verification that the purpose has been achieved or that the data is no longer necessary or relevant to the achievement of the specific purpose;
II. upon the end of the processing period;
III. following communication from the data subject, including in the exercise of their right to withdraw consent;
IV. by determination of the national authority, in cases of violation of the LGPD.

In this context, it becomes necessary to model the period of time for which personal data should, in principle, be retained. To enable this, LGPDByD strategy introduces a new type of constraint, referred to as **DURATION**, which defines a list of possible storage durations for a given relation. The syntax of the **DURATION** constraint is shown in Listing 28.

Table 13. Illustrative Instance of the Result Relation.

cod-patient	cod-exam	date-exam	result
1	10	2025/01/01	Result 1
1	20	2025/02/02	Result 2
2	10	2025/03/03	Result 3
2	30	2025/04/04	Result 4
3	40	2025/05/05	Result 5
3	30	2025/06/06	Result 6
4	50	2025/07/07	Result 7
4	60	2025/08/08	Result 8

Table 14. Output of the Rewritten Query Q2 for the Purpose P1.

cod-patient	cod-exam	date-exam	result
1	10	2025/01/01	Result 1
1	20	2025/02/02	Result 2
2	10	2025/03/03	Result 3
2	30	2025/04/04	Result 4
3	40	2025/05/05	NULL
3	30	2025/06/06	NULL

Listing 28: DURATION Constraint Syntax

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT NAME_CONSTRAINT DURATION (d1 <desc1>, d2 <desc2>, ...);
```

Hypothetically, consider that patient data may be stored either for 30 days or for 60 days. Thus, there are two possible storage durations for the tuples in the Patient relation, generically referred to as ***t1*** and ***t2***. Listing 29 show the ALTER TABLE command used to add the DURATION constraint.

Listing 29: Adding the Duration Constraint in Patient Relation

```
ALTER TABLE Patient
ADD CONSTRAINT D1 DURATION (t1 '30 'days, t2 '60 'days);
```

Now, when inserting a new tuple into the Patient relation, it becomes necessary to also specify which of the predefined storage duration alternatives will be associated with that particular tuple. Consider, for example, the insertion of patient Amy into the Patient table. In this case, the **INSERT** command must indicate which of the previously defined storage duration options for the Patient table will be assigned to the tuple storing Amy's data. Suppose that Amy has authorized the storage of her data for only 30 days. In this case, Listing 30 shows the **INSERT** command to add Amy's data. Furthermore, assume that the remaining patients have chosen to authorize the storage of their data for a period of 60 days. Table 15 illustrates an instance of the Patient relation with the metadata necessary to manage the data retention.

Listing 30: Inserting Amy's Data into Patient Relation

```
INSERT INTO Patient
VALUES (1, ''123, ''Amy, '123 'St, ''80/10/30, ''F, ''White, '' Christian, ''Female)
WITH DURATION t1
```

It is worth noting that, in order to maintain compatibility with legacy systems that already use the INSERT command without the WITH DURATION clause, it is possible to implement an automatic mechanism, using triggers, for example, that whenever a tuple is inserted without the aforementioned clause, selects the most restrictive duration, i.e., the shortest retention period. Similarly, a less restrictive alternative could be selected, or even a default option. This decision could be configured by the Data Protection Officer (DPO). In any case, the duration constraint implies the need to store the insertion date. Additionally, a strategy should be implemented to handle the scenario in which the specified duration for a given tuple expires. Nevertheless, the design and implementation of such a strategy are beyond the scope of this article.

4 LGPD Compliance Support Tools

Based on the metadata generated during the application of the LGPDByD strategy, it is possible to identify which attributes are personal or non-personal, which are classified as sensitive, and which should be anonymized or encrypted, among other classifications. This opens the possibility for the development of tools capable of performing automated checks directly on the database tables, in order to verify whether the stored data is, in fact, consistent with what was defined during the database design. Moreover, such tools could also be developed to support LGPD compliance auditing activities.

4.1 Automatic Checking/Auditing Tool

The purpose of this tool is to leverage the metadata added during the database design to verify, for instance, whether the attributes defined as anonymized or encrypted are, in fact, stored in the database with the appropriate data processing applied as previously specified. After performing these checks, the tool could generate a compliance report. With this report in hand, the Data Controller or Data Protection Officer (DPO) would be able to request the necessary changes to ensure that the organization is fully compliant with the LGPD. Figure 5 illustrates a possible architecture for an automated auditing tool.

Table 15. Illustrative Instance of the Patient Relation with Duration Metadata.

cod-patient	cpf	name	address	birthday	sex	...	insert_date	duration
1	123.456.789-00	Amy	123 St	80/10/30	F	...	25/05/13	t1
2	234.567.890-01	Anna	234 St	81/09/30	F	...	25/05/14	t2
3	456.789.012-34	Eva	456 St	82/08/30	F	...	25/05/15	t2
4	678.901.234-56	Ben	678 St	83/07/30	M	...	25/05/16	t2
5	891.234.567-89	Eric	891 St	84/06/30	M	...	25/05/17	t2

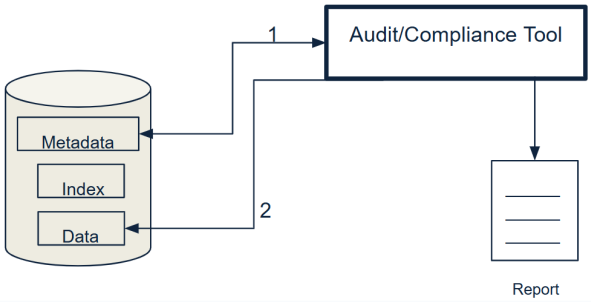


Figure 5. A Proposed Architecture for an Automated Auditing System.

4.2 Creating Packages/APIs for Developers

The purpose of this tool is to provide packages (*packages*), or APIs, independent of DBMSs, containing the implementation of different anonymization and encryption methods. With these packages, the programmer/developer can create *Triggers* and *Procedures* to implement, in the database system itself, the appropriate treatment, defined in the database project, for the attributes that represent personal data. For example, if a certain attribute needs to be stored in encrypted form, the developer selects, from the package provided, the most appropriate function and creates *Triggers* that will be executed during the insertion or update of this attribute. In this way, the proposed tool can contribute to reducing development time and compliance with the LGPD. Figure 6 presents a possible architectural model for an API intended to assist developers or database professionals.

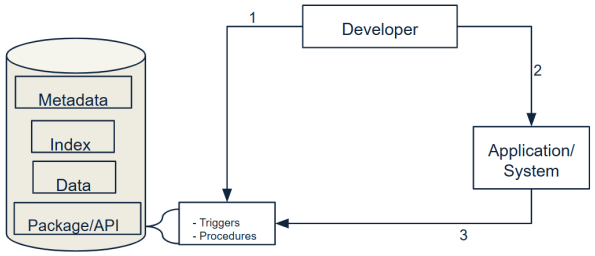


Figure 6. Architecture for an API Designed to Support Developers or Database Professionals.

5 Conclusions and Future Work

In this paper, we present a strategy, called LGPDbyD, to incorporate LGPD requirements and restrictions into database design. To this end, we added small adaptations to the ER model, the Relational model, and the CREATE TABLE command. Furthermore, we explore how modeling, designing, and implementing the legal concepts of purpose, consent, and personal data retention period in database systems. Finally, we extended the brModelo tool to provide support for the LGPDbyD methodology. The proposed methodology seeks to facilitate the processes of database design and auditing in compliance with the LGPD, in addition to encompassing the specifications outlined in Chapters I, II, III, VI, VII, and X of the law, since these are provisions that any companies, whether public or private, must meet to comply with the LGPD.

Although the LGPDbyD strategy represents a significant advancement in incorporating the requirements of the LGPD into the database design process, it is important to acknowledge its main limitations. First, the proposed approach requires some modifications to existing development tools and database management systems (DBMSs), which may pose a considerable barrier to adoption in organizations with consolidated legacy infrastructures. Furthermore, LGPDbyD relies heavily on the ability of database administrators to properly understand and apply the legal concepts of the LGPD during the modeling process, which may lead to inadequate or incomplete implementations in scenarios where appropriate legal expertise is lacking. The methodology also presents limitations regarding scalability and performance, since the automatic rewriting of queries for consent and purpose verification can introduce significant computational overhead in systems handling large volumes of data and complex queries. Finally, although the article mentions the need for automated auditing tools, the practical implementation of such compliance verification mechanisms remains as future work, leaving an important gap between the theoretical proposal and its effective application in production environments.

As future work, we intend to develop tools that assist in the design, implementation, and auditing of databases in compliance with the LGPD. Additionally, we will implement and evaluate a query rewrite mechanism to support the concepts of purpose, consent, and role proposed by LGPDByD, in order to ensure that a given piece of data is only used for the purpose specified by the Data Subject. Finally, we plan to provide support to manage the concept of duration time proposed by LGPDByD, making it possible, for example, to automatically remove data whenever the period of retention specified by the Data Subject is exceeded.

References

Araújo, E., Vilela, J., Silva, C., and Alves, C. (2021). Are my business process models compliant with lgpd? the lgpd4bp method to evaluate and to model lgpd aware business processes. In *XVII Brazilian Symposium on Information Systems*, pages 1–9. Sociedade Brasileira de Computação.

Brito, F. T. and Machado, J. C. (2017). Preservação de privacidade de dados: Fundamentos, técnicas e aplicações. *Jornadas de atualização em Informática*, pages 91–130.

C. Machado, J. and P. Amora, P. R. (2021). The impact of privacy regulations on db systems. *Journal of Information and Data Management*, 12(5). DOI: 10.5753/jidm.2021.1958.

Canedo, E. D., Cerqueira, A. J., Gravina, R. M., Ribeiro, V. C., Camoes, R., dos Reis, V. E., de Mendonça, F. L. L., and de Sousa Jr, R. T. (2021). Proposal of an implementation process for the brazilian general data protection law (lgpd). In *ICEIS (1)*, pages 19–30. Sociedade Brasileira de Computação.

Carauta Ribeiro, R. and Dias Canedo, E. (2020). Using mcda for selecting criteria of lgpd compliant personal data security. In *The 21st Annual International Conference on Digital Government Research*, pages 175–184.

Carvalho, G., Bernardino, J., Pereira, V., and Cabral, B. (2023). Er+: A conceptual model for distributed multilayer systems. *IEEE Access*, 11:62744–62757.

da Silva Barros, P. V., Monteiro, J. M., Brayner, A., and Machado, J. C. (2024). Incorporando os requisitos e as restrições da LGPD ao projeto de banco de dados. In Pires, C. E. S., Razente, H. L., and Ogasawara, E. S., editors, *Proceedings of the 39th Brazilian Symposium on Databases, SBBD 2024, Florianópolis, SC, Brazil, October 14-17, 2024*, pages 341–353. SBC. DOI: 10.5753/SBBD.2024.240791.

Dani, A. and Getta, J. (2005). Conceptual modelling of computations on data streams. *Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling*, 43.

de Abreu, I. C., Praciano, F. D., Amora, P. R., and Machado, J. C. (2021). Consql: Consentimentos em sql para o processamento de consultas orientado a propositos. In *SimpOsio Brasileiro de Banco de Dados (SBBD)*, pages 8–14. SBC.

de Castro, E. T. V., Silva, G. R. S., and Canedo, E. D. (2022). Ensuring privacy in the application of the brazilian general data protection law (lgpd). In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC '22, page 1228–1235, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3477314.3507023.

dos Santos Mello, R., Cândido, C. H., and Neto, M. B. S. (2021). brmodelo: An initiative for aiding database design. volume 12.

Favero, E. S. (2019). Um protótipo de referência para ferramentas case de modelagem em ambiente web. *Universidade Federal do Pampa; (2019); 105*.

Kamble, A. S. (2008). A conceptual model for multidimensional data. In *APCCM*, volume 8, pages 29–38.

Khan, K. M., Kapurubandara, M., and Chadha, U. (2004). Incorporating business requirements and constraints in database conceptual models. In *Proceedings of the first Asian-Pacific conference on Conceptual modelling-Volume 31*, pages 59–64.

Konstantinidis, G., Holt, J., and Chapman, A. (2021). Enabling personal consent in databases. *Proc. VLDB Endow.*, 15(2):375–387. DOI: 10.14778/3489496.3489516.

Kraska, T., Stonebraker, M., Brodie, M., Servan-Schreiber, S., and Weitzner, D. (2019). Schengendb: A data protection database proposal. In Gadepally, V., Mattson, T., Stonebraker, M., Wang, F., Luo, G., Laing, Y., and Dubovitskaya, A., editors, *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, pages 24–38, Cham. Springer International Publishing.

Lachaud, E. (2020). Iso/iec 27701 standard: Threats and opportunities for gdpr certification. *Eur. Data Prot. L. Rev.*, 6:194.

Mok, W. Y. (2024). A conceptual model based design methodology for mongodb databases. In *2024 7th International Conference on Information and Computer Technologies (ICICT)*, pages 151–159. DOI: 10.1109/ICICT62343.2024.00030.

Rocha, L. D., Silva, G. R. S., and Dias Canedo, E. (2023). Privacy compliance in software development: A guide to implementing the lgpd principles. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, SAC '23, page 1352–1361, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3555776.3577615.

Sarkar, S. and Athanassoulis, M. (2022). Query language support for timely data deletion. In *Proceedings of the 25th International Conference on Extending Database Technology*, volume 2.

Schwarzkopf, M., Kohler, E., Frans Kaashoek, M., and Morris, R. (2019). Position: Gdpr compliance by construction. In Gadepally, V., Mattson, T., Stonebraker, M., Wang, F., Luo, G., Laing, Y., and Dubovitskaya, A., editors, *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, pages 39–53, Cham. Springer International Publishing.

Shastri, S., Banakar, V., Wasserman, M., Kumar, A., and Chidambaram, V. (2020). Understanding and benchmarking the impact of gdpr on database systems. *Proceedings of the VLDB Endowment*, 13(7):1064–1077. DOI: 10.14778/3384345.3384354.

Wang, L., Near, J. P., Somani, N., Gao, P., Low, A., Dao, D., and Song, D. (2022). Data capsule: A new paradigm for automatic compliance with data privacy regulations. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, page 3–23, Berlin, Heidelberg. Springer-Verlag.

Éllen Renner Ferrão, S., Ramos Sousa Silva, G., Dias Canedo, E., and Freitas Mendes, F. (2024). Towards a taxonomy of privacy requirements based on the lgpd and iso/iec 29100. *Information and Software Technology*, 168:107396. DOI: https://doi.org/10.1016/j.infsof.2024.107396.