Access Control Policies for Relational Databases in Data Exchange Process

Adel Jbali(✉) and Salma Sassi

Economics and Management of Jendouba/VPNC Laboratory, Faculty of Law, University of Jendouba, Jendouba, Tunisia
adeljbali@mail.com, salma.sassi@fsjegj.rnu.tn

Abstract. Nowadays, many organizations rely on database systems as the key data management technology for a large variety of tasks. This wide use of such systems involved that security breaches and unauthorized disclosures threat those systems especially when the data is exchanged between several parts in a distributed system. Consequently, access control must adapt to this exchange process to maintain data privacy. In this paper, the challenge is to design an approach to deal with access control policies in a context of data exchange between relational databases. In fact, the main problem that we are dealing with is that given a set of policies attached to a source schema and a set of mapping rules to a target schema, the question is how the policies will pass from the source schema to the target schema and what are the policies that will be attached to the target schema to comply with the set of source policies. For that purpose, we propose in this paper our methodology called Policies-generation.

Keywords: Access control · Data exchange · Authorization view · Database security

1 Introduction

Databases may contain sensitive and critical government or enterprise information like medical document, personal health information or customer credit information cards. We focus in this paper on the security challenge that arises when this data is exchanged between distributed parts. Yet, although the importance of taking into consideration access control in a context of a data exchange, we have remarked during our bibliographic study the absence of works that tackles directly this issue. Data exchange is the process of taking data structured under a source schema and materializing an instance of a target schema that reflects as accurately as possible the source data [12]. Hence, the challenge in a such situation is to determine the way in which the access control policies will be translated from the source database to the target database, and the set of rules that will preserve the target policies to remain comply with source policies. Complying with the source policies means that a prohibited access at the source

D. Benslimane et al. (Eds.): DEXA 2017, Part I, LNCS 10438, pp. 264–271, 2017.
DOI: 10.1007/978-3-319-64468-4_20

level should also be prohibited at the target level to avoid policy violation. Thus, we propose a methodology that defines in the first time the set of access control policies attached to the source database and we give our reasons for the access control model choice, and in the second time we give our algorithm *Policies-generation* in which we specify how the access control policies are translated from the source to target database. In fact, our algorithm treats three steps: the first step is the policies filtering decision in which the algorithm determines the policies that will pass to the target database and the ignored policies. The second step is the policies modification decision in which our algorithm determines the set of policies that pass to the target database without modification and the policies that will be regenerated according to the mapping rules. In the last step, the algorithm generates the new policies respecting the mapping rules if it is possible. The remainder of the paper is organized as follows: Sect. 2 gives an overview of research effort related to our work. In Sect. 3 we describe our approach. We conclude in Sect. 4.

2 Background

In this section, we will discuss the different works which are related in any way to our problem. We highlight that, in actual database research literature, there are not significant contributions focusing on the relevant issue of accees control in relation with data exchange. This further confirms to us the novelty of the research we propose.

2.1 Logical Foundations of Relational Data Exchange

According to [12] data exchange is the problem of materializing an instance that adheres to a target schema, given an instance of a source schema and a specification of the relationship between the source schema and the target schema. This problem arises in many tasks requiring data to be transferred between independent applications that do not necessarily adhere to the same data format (or schema).

Definition 1. *A schema mapping is a triple* $M = (S, T, \sum)$*, where* S *and* T *are the disjoint source and target schema respectively, and* $\sum$ *is a finite set of sentences of some logical language over the schema* $S \cup T$ [1]*. We can define a schema mapping from a semantic view as a set of all pairs (I, J) where I is the source instance, J is the target instance and (I, J) must satisfies* $\sum$*. We said that the target instance J is a solution for I with respect to M if (I, J) satisfies* $\sum$*. The set of solution to I with respect to M is denoted by* $Sol_M(I)$*.*

– $\sum_{st}$ *consists of a set of source-to-target dependencies (stds), of the form:*

$$\forall\,(\phi(x) \longrightarrow \exists y \psi(x, y))$$

Where $\phi(x)$ *and* $\psi(x, y)$ *are conjunctions of atomic formulas in* S *and* T *respectively.*

– $\sum_t$ *the set of target dependencies. It represents the union between a set of equality generating dependencies (egds) of the form:* $\forall(\phi(x) \longrightarrow xi = xj)$ *for* xi, xj *variables in* x, *and a set of tuple generating dependencies of the form* $\forall(\phi(x) \longrightarrow \exists y\psi(x, y))$ *where* $\phi(x)$ *and* $\psi(x, y)$ *are conjunctions of atomic formulas in* T.

2.2 Access Control Model for Relational Databases with Authorization Views

Definition (Authorization View) 2. *A set of authorization views specifies what information a user is allowed to access. The user writes the query in terms of the database relations, and the system tests (by considering the authorization views) the query for validity by determining whether it can be evaluated over the database* [4].

The view based access control model was emerged to enforce access control at tables, columns and even cell level (fine granularity). This model can be enforced with the definition of a secure context for each request that encapsulate the information related to the query [4]. Another feature of this model is its ability to define permission based on the data content. Declarative languages like SQL query language, facilitated the development of this specification using the concept of view and query rewriting technique. According to [6] the author classify view based access control model in two categories. The first one is Truman Model which applies the query rewriting technique to the user query to provide only the answers that are authorized to the user. The second model is Non-Truman Model, in a such model a query rewriting technique is also executed, but the difference according to the previous model is in the interpretation of the query. Thereby, In our methodology, no query rewrite is performed also we mention the use of authorization views based on conjunctive queries (the clause where) to represent both the sets of source policies and target policies.

3 Access Control and Data Exchange: A View Based Approach

In this section we introduce our approach and we show through the Fig. 1 an overview about our proposal. In fact, this approach is split in four major step: The first step incarnates the process of data exchange. The second step talks about the adaptation of the View Based Access Control model to define the authorization views. The third step talks about the views transformation and we introduce in the last step the algorithm Policies-generation.

3.1 Specification of the Mapping Rules (Phase A)

As we had mentioned previously, our methodology aims to translate the authorization views from the source to target database in a context of data exchange

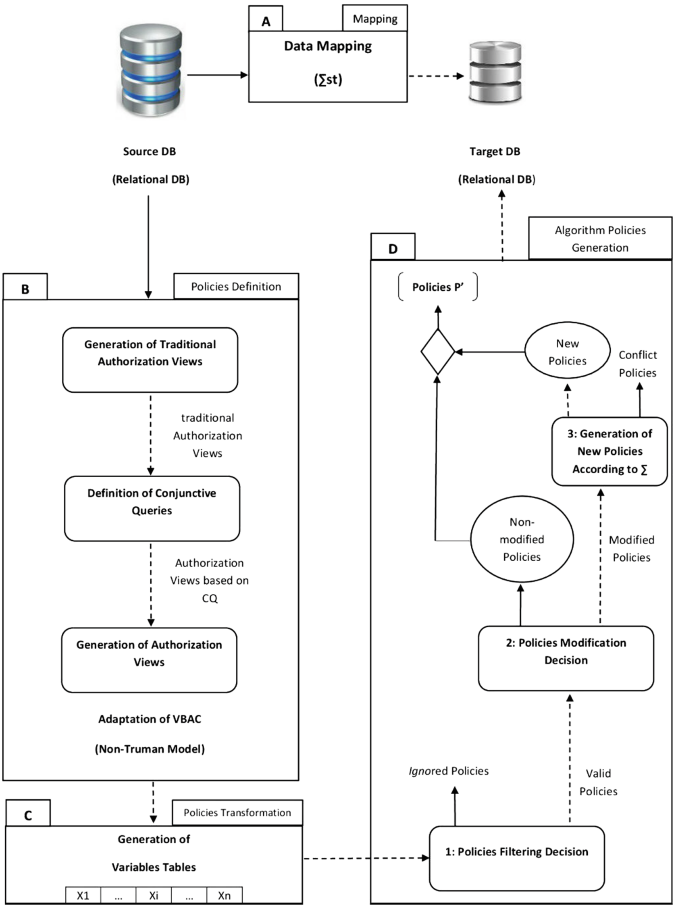

Fig. 1. Policies-generation: system architecture

respecting the mapping rules. The first type of tgds is a full tgds which there are no existentially quantified variables. It corresponds to a simple data migration from source-db to target-db:

$$\forall x \, (\phi(x) \longrightarrow \psi(x))$$

The second tgds treats the case of exchanging the attributes of the tables through an integration mapping (used in data integration system) [9]. In fact, for the set of distinguish attributes A_i that was translated from the source-db to the target-db, we associate a set of distinguish authorization views for these attributes and we show through this mapping how to generate a global authorization view that synthesizes all the source authorization views related to those attributes through the intersection of conjunctive queries.

3.2 Generation of the Source Policies and Extraction of the Attributes-Tables (Phase B-C)

Admitting the interesting features of the view based access control model and the expressive power of conjunctive queries language, the authorization views admits the following syntax:

CREATE AUTHORIZATION VIEW AS view-name
SELECT attributes FROM relation WHERE condition

Where the condition is a conjunction of atomic formulas over the attributes A_i. Subsequently, for every authorization view we extract a table containing the names of all attributes on which the authorization view is based. The attributes-tables represents the input of our algorithm policies-generation.

3.3 Algorithm Policies-Generation (Phase D)

Input

- Schema of the source-db and target-db
- The set of attributes-tables T_j: $T_j\,[i] = A_i$
- Mapping rules

Output

- Policies filtering decision
- Policies modification decision
- New generated policies

Algorithm Policies-Generation

First step: Policies filtering decision

Begin
1 $i \longleftarrow 0$
2 $solexistance \longleftarrow true$
3 $n = T_j.length$
4 $while\,((solexistance = true)\,\&\,(i < n))\,do$
5 $if\,(\phi\,(T_j\,[i]) \longleftarrow \emptyset)$
6 $solexistance \longleftarrow false$
7 $write(policy\ will\ be\ ignored)$
8 $else$
9 $i++$
10 $endif$
11 $endwhile$
12 $if\,(i = n)\,then$
13 $write("policy\ will\ pass\ to\ the\ target\ schema)$
14 $endif$
15 $Return$

This first step of our algorithm treats the problem of policy passing decision. Indeed, the algorithm run through the table T which represents the attributes A_i of the authorization view. Then for each attribute A_i in the table, it tests if this attribute admits a solution in the target schema, if it is not then the policy will be ignored since there is an attribute in the authorization view that had not been pass to the target schema. If we finish with a counter equal to the table length, it means that all the attributes of the authorization view had been passed to the target schema and the authorization view will also pass.

Second step: Policies modification decision

Begin
1 $i \longleftarrow 1$
2 $k \longleftarrow 1$
3 *for i from* 1 *to n do*
4 *if* $(\phi(T_j[i]) \longleftarrow \psi(Y_i)) \,\&\, (A_i = Y_i)$ *then*
5 $k++$
6 *endif*
7 *endfor*
8 *if* $(k = n)$ *then*
9 *write*("*policy will be pass to target schema without modification*)
10 *else*
11 *write*("*policy will be regenerated according to the mapping rules*)
12 *endif*
13 *Return*

In this step, our algorithm performed the following treatment. It compares the attribute A_i in the source schema with their solution Y_i in the target schema cell by cell. If it founds that all the attribute A_i in the source schema are identical to those in the target schema it avers that the authorization view will pass to the target schema without modification. Otherwise, the policy will be regenerated according to the mapping rules.

Third step: Generation of global view

Begin
1 $i \longleftarrow 1$
2 $k \longleftarrow 1$
3 $m \longleftarrow T_{j+1}.length$
4 $equal \longleftarrow true$
5 *while* $(k \leq m \;\&\; equal = true)$ *do*
6 *for i from* 1 *to n do*
7 *if* $(\phi(T_j[i]) = \phi(T_{j+1}[k]))$ *then*
8 *if* $(T_j[i]$ *intersect* $T_{j+1}[k] <> \emptyset)$ *then*

9 *write* ("*the global view is the intersection of the source attributes*")
10 *else*
11 *equal* $\longleftarrow$ *false*
12 *write* ("*conflict policies*")
13 *endif*
14 *else*
15 $k \longleftarrow k+1$
16 *endif*
17 *endwhile*
18 *if* $(k = m+1)$ *then*
19 *write* ("*the global view is the conjuction among all the source attributes*")
20 *endif*
21 *Return*

In this last step, our algorithm generates a global authorization view according to the mapping rules mention in Sect. 3.1. Hence, in this step we consider two source authorization views represented by two attributes-tables T_j and T_{j+1}, our algorithm run through the tables and if it detects that the authorization views admit common attributes images by the mapping rules then for every common attributes images it performs an intersection between those ones. If the two common attributes have common elements means that $\phi(T_j) \cap \phi(T_{j+1}) \neq \emptyset$, then the global authorization view is defined based on the new intersection and by adding attributes that are not in common. If it is not the case $(\phi(T_j) \cap \phi(T_{j+1}) = \emptyset)$ and the two attributes doesn't have any common elements then we have a conflict policies in this case and the generation of a global authorization view may conduct to a policy violation. In the case where the source authorization views don't admit common attributes then the global authorization view is defined by the conjunction of all attributes of the conjunctive queries.

4 Conclusion

In this work we have investigated an interesting problem in databases access control, we have focused on the access control policies in a context of data exchange between relational databases. We proposed an approach which exploits the view based access control model and the conjunctive query language to define the set of source policies called authorization views. Then by reasoning about the attributes, we have extracted an algorithm able to produce the set of target policies that should be attached to the target database respecting to the mapping rules in order to comply with source policies. As perspectives of our work, we aim to deal with the case where our algorithm ignores the policy, we will try to find an insight that assures the passage of the policy from the source schema to target schema with the minimum missing data selection. Another perspective is to make our system more flexible to deal with different data representation model like XML and RDF.

References

1. Fuxman, A., Kolaitis, P.G., Miller, R.J., Tan, W.C.: Peer data exchange. ACM Trans. Database Syst. (2006)
2. Ten Cate, B., Chiticariu, L., Kolaitis, P., Tan, W.: Laconic schema mappings: computing the core with SQL queries. Proc. VLDB Endow
3. Marnette, B., Mecca, G., Papotti, P.: Scalable data exchange with functional dependencies. Proc. VLDB Endow. (2010)
4. Bertino, E., Ghinita, G., Kamra, A.: Access control for databases: concepts and systems. Found. Trends Databases
5. Gertz, M., Jajodia, S.: Handbook of Database Security Applications and Trends (2007)
6. Haddad, M., Hacid, M.-S., Laurini, R.: Data integration in presence of authorization policies. In: 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom (2012)
7. Arenas, M., Barceló, P., Reutter, J.: Query Languages for Data Exchange: Beyond Unions of Conjunctive. Springer Science+Business Media, LLC (2010)
8. Sellami, M., Hacid, M.S., Gammoudi, M.M.: Inference control in data integration systems. In: Debruyne, C. (ed.) OTM 2015. LNCS, vol. 9415, pp. 285–302. Springer, Cham (2015). doi:10.1007/978-3-319-26148-5_17
9. Sellami, M., Gammoudi, M.M., Hacid, M.S.: Secure data integration: a formal concept analysis based approach. In: Decker, H., Lhotská, L., Link, S., Spies, M., Wagner, R.R. (eds.) DEXA 2014. LNCS, vol. 8645, pp. 326–333. Springer, Cham (2014). doi:10.1007/978-3-319-10085-2_30
10. Nait Bahloul, S.: Inference of security policies on materialized views. Report master 2 (2009). http://liris.cnrs.fr/snaitbah/wiki
11. Kolaitis, P.G.: Schema mappings and data exchange. In: Proceedings of the Twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (2012)
12. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. (2002)
13. Miller, R.J.: Retrospective on Clio: schema mapping and data exchange in practice. In: Proceedings of the 2007 International Workshop on Description Logics (2007)
14. Rizvi, S., Mendelzon, A, Sudarshan, S., Roy, P.: Extending query rewriting techniques for fine-grained access control. In: International Conference on Management of Data (2004)