УДК 004.657

УПРАВЛЕНИЕ ДОСТУПОМ К СТРОКАМ ТАБЛИЦЫ С ИСПОЛЬЗОВАНИЕМ ИЕРАРХИИ ПОЛНОМОЧИЙ

Михаил Коломыцев, Светлана Носок, Анастасия Мазуренко

Статья посвящена актуальной проблеме защите информации в базах данных. Прикладные программы доступа к базам данных в корпоративной информационной системе с целью обеспечения гибкости политики безопасности при доступе к данным требуют управления доступом через программирование механизма доступа на уровне строк таблиц БД (Row Level Security). Существующие подходы требуют создания дополнительных столбцов в таблицах и программных объектов, которые определяют механизмы фильтрации строк. В статье предлагается другой подход, когда правила предоставления полномочий вынесены в отдельную таблицу. Метод основывается на ограничении доступа к данным конкретных строк таблицы для операций чтения, модификации и удаления. Метод использует структурно-должностную иерархию пользователей, объекты базы данных и программные шаблоны операций управления доступом в различных СУБД. Предложенный метод реализован в виде специальной таблицы, триггеров, представлений и пользовательских функций для системы управления базами данных (СУБД) MS SQL Server.

Ключевые слова: *база данных, защита персональных данных, контроль доступа на уровне строк, триггер, представление, информационная система.*

Актуальность и постановка задачи разграничения доступа к данным.

Контроль доступа к информации, базирующийся на правах доступа пользователя, является фундаментальной частью большинства информационных систем. В некоторых случаях требуется управлять доступом на более низком уровне, чем путем простого предоставления, отзыва и отказа в праве доступа к данным. Например, список пациентов и их диагнозов может храниться в одной таблице. Каждый врач должен иметь доступ к информации только своих пациентов (разграничение на уровне записей). Похожие требования предъявляются к системам из многих областей, включая финансы, юриспруденцию, государственные и военные системы. Для предприятий, в ведении которых находится обработка персональных данных, выполнение требований такого рода продиктовано принятием закона о защите персональных данных.

Разграничение доступа на уровне строки используются для приложений, в которых данные хранятся в одной таблице. Современные СУБД как правило, обладают такой возможностью [1, 2, 3]. Для этого необходимо создавать дополнительные столбцы в таблицах и программные объекты, которые определяют механизмы фильтрации строк. Различные авторы предлагают свои подходы к решению этой задачи [4, 5, 6]. В статье предлагается подход, когда правила предоставления полномочий вынесены в отдельную таблицу. Такой подход называется data driven security.

Постановка задачи. Рассмотрим следующую ситуацию: организационная структура предприятия имеет иерархический вид (рис. 1):

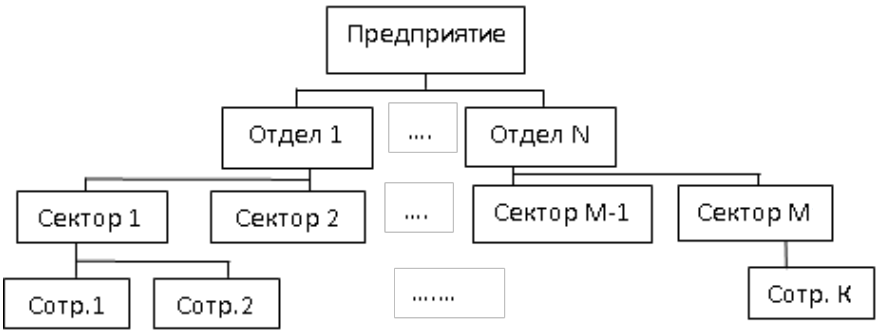

Рис. 1. Организационная структура предприятия.

Сотрудники предприятия имеют полномочия, соотвествующие их положению в иерархической структуре – сотрудники уровня отдела имеют доступ ко всей информации отдела и секторов их отдела, сотрудники уровня сектора имеет доступ только к информации данного сектора. Количество уровней иерархии может быть увеличено. Чем выше положение сотрудника в иерархии, тем больше информации ему доступно.

Сама информация располагается в отдельной таблице, назовем ее таблицей с данными. Каждая строка этой таблицы относятся к одному из секторов.

Целью работы является разработка метода управления доступом к строкам таблицы с учетом структурно-должностной иерархии пользователей.

Метод разграничения доступа к данным.

Предлагаемый метод разграничения доступа можно рассматривать как аналог разграничения доступа на уровне строк Row Level Security (RLS). Так же, как и RLS, метод реализуется с помощью набора скриптов и дополнительных объектов базы данных.

В описании предлагаемого метода используется MS SQL Server 2008 или выше. Само решение состоит из следующих этапов:

1. Создание специальной таблицы (таблица распределений полномочий), содержащей информацию о иерархической структуре предприятия и связывающая эту структуру с защищаемыми данными. В приведенных примерах реализации метода она называется AccessHierarchy. Таблица содержит такие атрибуты:

– атрибут HierarchyType содержит описание уровня иерархии. Значения этого поля таблицы могут быть такими: «Отдел» (DEPARTMENT), «Сектор» (OFFICE), «Пользователь» (USER), Документ (DOCUMENT) или название корня иерархии «Предприятие» (ROOT);

– атрибут Description носит вспомогательный характер и содержит краткий комментарий содержимого записи;

– атрибут UserLogin содержит название учетной записи пользователя Windows в форме DOMAIN\USERNAME. Если используется смешанный режим аутентификации SQL Server и Windows, атрибут может содержать имя входа пользователя. Это поле таблицы заполняется в тех строках, где атрибут HierarchyType содержит значение «USER»;

– атрибут DocumentsKey – это внешний ключ к таблице с защищаемыми данными. Это поле заполняется в тех строках таблицы, где поле HierarchyType содержит значение «DOCUMENT»;

– атрибут ParentHierarchyKey – это ключ рекурсивной связи таблицы AccessHierarchy, содержит ссылку на первичный ключ таблицы HierarchyKey. С помощью этого атрибута моделируется иерархическая структура предприятия.

2. Создание объектов базы данных (функции, представление, триггер), обеспечивающих разграничение прав доступа пользователей. Эти объекты, путем анализа содержимого таблицы AccessHierarchy, определяют доступные пользователю строки таблицы с данными. Для чтения данных используется представление (view), для модификации (в том числе и удаления строк таблицы) данных можно использовать триггер.

Рассмотрим пример реализации метода.Предварительно создадим таблицу, играющую роль таблицы с данными:

```
CREATE TABLE [dbo].[Documents](
    [Id] [int] NOT NULL,
    [Description] [nvarchar](Max) NULL,
 CONSTRAINT [PK_Documents] PRIMARY KEY CLUSTERED ([Id] ASC)
)
```

Для создания таблицы распределения полномочий используем следующий скрипт:

```
CREATE TABLE [dbo].[AccessHierarchy]
(
 [HierarchyKey] [int] IDENTITY(1,1) NOT NULL,
 [HierarchyType] [varchar](50) NOT NULL,
 [Description] [varchar](50) NOT NULL,
 [UserLogin] [varchar](50) NULL,
 [DocumentsKey] [int] NULL,
 [ParentHierarchyKey] [int] NULL
)
ALTER TABLE [dbo].[AccessHierarchy] ADD CONSTRAINT [PK_AccessHierarchy]
PRIMARY KEY CLUSTERED ([HierarchyKey] ASC)
ALTER TABLE [dbo].[AccessHierarchy] ADD CONSTRAINT [FK_AccessHierarchy]
FOREIGN KEY ([ParentHierarchyKey]) REFERENCES [dbo].[AccessHierarchy] ([HierarchyKey])
ALTER TABLE [dbo].[AccessHierarchy] ADD CONSTRAINT [FK_AccessHierarchy_Documents]
FOREIGN KEY ([DocumentsKey]) REFERENCES [dbo].[Documents] (Id)
```

Диаграмма таблиц представлена на рис. 2.

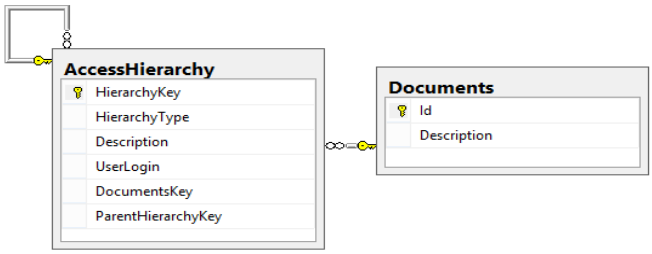

Рис. 2 ER-диаграмма таблиц базы данных.

На рис. 3 представлен пример заполнения таблицы AccessHierarchy.

HierarchyKey	HierarchyType	Description	UserLogin	DocumentsKey	ParentHierarchyKey
1	ROOT	ORGANIZATION HIERARCHY	NULL	NULL	NULL
2	Department	Отдел #1	NULL	NULL	1
3	Department	Отдел #2	NULL	NULL	1
4	Department	Отдел #3	NULL	NULL	1
5	OFFICE	Сектор 1	NULL	NULL	2
6	OFFICE	Сектор 2	NULL	NULL	2
7	OFFICE	Сектор 3	NULL	NULL	3
8	OFFICE	Сектор 4	NULL	NULL	3
9	OFFICE	Сектор 5	NULL	NULL	4
10	OFFICE	Сектор 6	NULL	NULL	4
11	USER	Пользователь 1	Test	NULL	2
12	USER	Пользователь 2	OUR_DOMAIN\user3	NULL	1
13	USER	Пользователь 3	OUR_DOMAIN\user3	NULL	10
14	DOCUMENT	Комментарий 1	NULL	1	12
15	DOCUMENT	Комментарий 2	NULL	2	13
16	DOCUMENT	Комментарий 3	NULL	3	12
17	DOCUMENT	Комментарий 4	NULL	4	12
18	DOCUMENT	Комментарий 5	NULL	5	11
19	DOCUMENT	Комментарий 6	NULL	6	11
20	DOCUMENT	Комментарий 7	NULL	7	13

Рис. 3 Пример заполнения таблицы распределения полномочий

Из рисунка видно, что Пользователь 1 имеет доступ к информации Отдела 1, Пользователь 2 имеет доступ на уровне корня иерархии (всего предприятия), Пользователю 3 доступна только информация Сектора 6.

Для получения списка доступных пользователю строк таблицы с данными создадим функцию DocsAccess():

```
CREATE FUNCTION
[dbo].[DocsAccess](@sys_usr char(30))
RETURNS TABLE
AS
RETURN
(
        WITH Documents_cte AS (
        SELECT parent.*
        FROM dbo.AccessHierarchy child
        JOIN dbo.AccessHierarchy parent ON
parent.HierarchyKey = child.ParentHierarchyKey
        WHERE child.UserLogin =  @sys_usr

        UNION ALL
        SELECT child.*
        FROM Documents_cte parent
        JOIN dbo.AccessHierarchy child ON
child.ParentHierarchyKey = parent.HierarchyKey
        )
        SELECT distinct DocumentsKey FROM
Documents_cte
)
```

Основные особенности функции:

– входным параметром функции является имя пользователя;

– в функции используется обобщенное табличное выражение (common table expression, CTE). Это объект, во многом сходный с представлением (view), но с важными отличиями. CTE не является объектом базы данных и его описание не хранится в схеме данных. Что особенно существенно, CTE допускает рекурсивный вызов. Именно эта особенность используется для отслеживания всех нижележащих ветвей иерархии;

– функция возвращает набор ссылок на строки таблицы данных, доступных из данной точки иерархии.

– Очевидно, пользователям должен быть запрещен прямой доступ к таблице с данными и разрешен только через представление. Пример такого представления для чтения данных:

```
CREATE VIEW [dbo].[AccessableDocuments]
as
SELECT ID, Description
```

```
FROM  dbo.Documents Docs
INNER JOIN (select *
from dbo.DocsAccess(SUSER_SNAME())) H
ON Docs.Id = H.DocumentsKey
```

В представлении использована системная функция SUSER_SNAME(), возвращающая имя текущего пользователя. Пример запроса для чтения данных:

```
SELECT * FROM dbo.Accessabledocuments
```

Id	Description
5	Описание документа 5
6	Описание документа 6

Рис. 4. Пример чтения данных для Пользователь 1.

Представление AccessableDocuments не является обновляемым. Для модификации таблицы с данными для этого представления можно создать триггер. Например, при вставке записи в таблицу с данными, триггер должен проанализировать атрибуты таблицы inserted и создать запись в базовой таблице (таблице с данными), а так же добавить запись в таблицу распределения полномочий, установив значение атрибута ParentHierarchyKey равным значений атрибута HierarchyKey пользователя – автора записи (непосредственная запись в эту таблицу запрещена).

Пример такого триггера:

```
CREATE TRIGGER InsteadTrigger on
[dbo].[AccessableDocuments]
INSTEAD OF INSERT
AS
IF @@ROWCOUNT=1
BEGIN
  DECLARE @DocId int
  INSERT INTO dbo.Documents
     SELECT Id, Description FROM inserted
  SELECT @DocId =  Id FROM inserted
  INSERT INTO dbo.AccessHierarchy
     VALUES('DOCUMENT','Комментарий','',
@DocId, dbo.UserKey(SUSER_SNAME()))
END
```

В триггере использована пользовательская функция UserKey(), которая возвращает значение атрибута HierarchyKey для текущего пользователя:

```
CREATE FUNCTION [dbo].[Us
erKey](@sys_usr char(30))
RETURNS int
AS
BEGIN
    DECLARE @i int;
    SELECT @i = T.ParentHierarchyKey
    FROM  dbo.AccessHierarchy T
    WHERE UserLogin = @sys_usr
    RETURN @i
END
```

Заключение

Предложенный авторами метод разграничения прав доступа использует подход, при которым одни данные управляют доступом к другим данным (data-driven). Другая особенность предложенного метода состоит в том, что он позволяет создать иерархию прав доступа, отображающую структурно-должностную иерархию пользователей. Сочетание этих особенностей позволяет упростить задачу администрирования базы данных, оперативно реагировать на изменение должностных обязанностей и прав доступа пользователей.

ЛИТЕРАТУРА

[1]. Предоставление разрешений уровня строки в SQL Server [Электронный ресурс] – Режим доступа: https:// msdn. microsoft. com / ru-ru/library/bb669076(v=vs.110).aspx.

[2]. Database Security Guide. 6 Access Control on Tables, Views, Synonyms, or Rows [Электронный ресурс] – Режим доступа: https://docs.oracle.com/cd/B19306_01/network.102/b14266/accessre.htm#CHDDGEJG.

[3]. Злыгостев А. Row-Level Security в РСУБД [Электронный ресурс] / Антон Злыгостев // RSDN Magazine: журнал для программистов. – 2004. – Режим доступа: http: / / rsdn. Ru / article / db / RowLevelSecurity.xml.

[4]. CRLS (Система управления доступом к данным) [Электронный ресурс] – Режим доступа: https://center-inform. Ru / upload / iblock / f9a / c626d1fc0985e11b23cc4f320c9ebeea.pdf.

[5]. Петухова Н. Метод обеспечения доступа к данным реляционных систем на уровне строк отношения [Электронный ресурс] / Наталья Петухова – Режим доступа: http: // www. tsi. lv / sites / default / files / editor / science / research_journal s /tr_tel/2003/v1/petuhova.pdf.

[6]. Хованец В. А. Адаптация информационных систем управления университетом требованиям закона о защите персональных данных [Электронный ресурс] / В. А. Хованец, П. В. Смолин. – 2010. – Режим доступа: http: // www. tusur. ru / filearchive/reports-magazine/2010-1/37-40.pdf.

REFERENCES

[1]. Granting row level permissions in SQL Server [Electronic resource] – Access mode: https: // msdn. microsoft.com/ru-ru/library / bb669076 (v = vs.110).aspx.

[2]. Database Security Guide. 6 Access Control on Tables, Views, Synonyms, or Rows [Electronic resource] – Access mode: https://docs.oracle. com / cd/B19306_01/network.102/b14266/accessre. htm #CHDDGEJG.

[3]. Zlygostev A. Row-Level Security in DDBMS [Electronic resource] / Anton Zlygostev // RSDN Magazine: magazine for programmers. – 2004. – Access mode: http://rsdn.ru/article/db/RowLevelSecurity.xml.

[4]. CRLS (Data access system) [Electronic resource] – Access mode: https://center-inform.ru/upload / iblock/f9a/c626d1fc0985e11b23cc4f320c9ebeea.pdf.

[5]. Petuhova N. Method of providing access to relational systems data on the row relation level [Electronic resource] / Natalya Petuhova – Access mode: http://www.tsi.lv/sites/default/files/editor/science /research_journals/tr_tel/2003/v1/petuhova.pdf.

[6]. Hovanets V. A. Adaptation of management information systems of university to the requirements of the personal data protection law [Electronic resource] / V. A. Hovanets, P. V. Smolin. – 2010. – Access mode: http://www.tusur.ru/filearchive/reports-magazine/2010-1/37-40.pdf

УПРАВЛІННЯ ДОСТУПОМ ДО РЯДКІВ ТАБЛИЦІ З ВИКОРИСТАННЯМ ІЄРАРХІЇ ПОВНОВАЖЕНЬ

Стаття присвячена актуальній проблемі захисту інформації в базах даних. Прикладні програми доступу до баз даних в корпоративній інформаційній системі з метою забезпечення гнучкості політики безпеки при доступі до даних потребують управління доступом через програмування механізму доступу на рівні рядків таблиць БД (Row Level Security). Існуючі підходи вимагають створення додаткових стовпців в таблицях і програмних об'єктів, які визначають механізми фільтрації рядків. У статті пропонується інший підхід, коли правила надання повноважень винесені в окрему таблицю. Метод ґрунтується на обмеженні доступу до даних конкретних рядків таблиці для операцій читання, модифікації і видалення. Метод використовує структурно-посадову ієрархію користувачів, об'єкти бази даних і програмні шаблони операцій управління доступом в різних СУБД. Запропонований метод реалізований у вигляді спеціальної таблиці, тригерів, представлень і користувацьких функцій для системи управління базами даних (СУБД) MS SQL Server. Метою роботи є розробка методу управління доступом до рядків таблиці з урахуванням структурно-посадовій ієрархії користувачів.

Ключові слова: база даних, захист персональних даних, контроль доступу на рівні рядків, тригер, представлення, інформаційна система.

ACCESS CONTROL TO TABLE ROWS USING HIERARCHY OF AUTHORITY

The article is devoted to the actual problem of the information protection in databases. Applications for databases access in the enterprise information system require access control by programming of access mechanism at the level of the database table rows (Row Level Security) to ensure the flexibility of security policy for data access. The existing approaches require the creation of additional columns in tables and program objects that define the mechanisms for rows filtering. The article proposes another approach where the rules of the granting permissions are in a separate table. The method is based on access restricting to data in specific rows in the table for reading, modifying and deleting. The method uses structural and job hierarchy of users, database objects and programming templates of operations for access control in different DBMS. The proposed method is implemented as special tables, triggers, views and user-defined functions for the database management system (DBMS) MS SQL Server. The goal is to develop a method for access control to table rows based on structural and job hierarchy of users.

Keywords: database, personal data protection, access control at the row level, trigger, view, information system.

Коломицев Михайло Володимирович, кандидат технічних наук, доцент Фізико-технічного інституту НТУУ «КПІ».
E-mail: box144a@ukr.net

Коломыцев Михаил Владимирович, кандидат технических наук, доцент Физико-технического института НТУУ «КПИ».

Kolomytsev Myhailo, candidate of technical sciences, associate professor of Physico- Technical Institute of the NTUU "KPI".

Носок Світлана Олександрівна, кандидат технічних наук, доцент Фізико-технічного інституту НТУУ «КПІ».
E-mail: svetlana@pti.kpi.net

Носок Светлана Александровна, кандидат технических наук, доцент Физико-технического института НТУУ «КПИ».

Nosok Svitlana, candidate of technical sciences, associate professor of Physico - Technical Institute of the NTUU "KPI".

Мазуренко Анастасія Євгенівна, студентка Фізико-технічного інституту НТУУ «КПІ».
E-mail: ks0610@mail.ru

Мазуренко Анастасия Евгениевна, студентка Физико-технического института НТУУ «КПИ».

Mazurenko Anastasia student of the Physico-Technical Institute of the NTUU "KPI".