## Экспертное заключение

Автор: Попов Сергей Андреевич, КБ-61СО

Программа: Приложение для настройки политики безопасности онлайн магазина

Выполнена в рамках курсовой (дипломной) работы: Реализация RLS (безопасность на уровне строк) в реляционных базах данных

Научный руководитель: Власова Ольга Владимировна

Аннотация:

Программа предназначена для гибкой настройки политики безопасности и может применяться в приложениях, где требуется часто менять правила ограничения доступа к данным базы данных.

Программа обеспечивает выполнение следующих функций:

* создание и распределение пользователей по группам
* создание предикатов в специальной текстовой форме
* создание политики безопасности и привязка к ней предикатов
* просмотр и редактирование данных базы данных
* назначение предикатов к таблицам и группам пользователей

Условия эксплуатации:

Для работы программы необходимо:

* операционая система – windows 7-10
* наличие установленных библитек фраемворка .NET не ниже версии 4.6
* наличие установленной среды node.js не ниже версии 7
* наличие установленного и запущеного SQL сервера не ниже версии 2016 года
* наличие устновленной программы для обозревания интернета
* наличие установленной программы visual studio не ниже версии 2015 года

Для работы с данной программой необходимо:

1. собрать release версию проекта находящегося в папке SqlParser
2. выполнить SQL скрипт в файле под названием – “db creating.sql” в папке “sql scripts”
3. выполнить SQL скрипт в файле под названием – “ policies creating.sql” в папке“sql scripts”
4. выполнить с помощью консоли windows команду “npm i” в папке “frontend”
5. выполнить с помощью консоли windows команду “npm run start” в папке “frontend”
6. открыть программу для обозревания интернета и перейти на адрес localhost:8080

*Должна, быть приведена полная информация об использовании программного средства, охватывающая:*

*a) подготовку к использованию;*

*b) входные команды и данные;*

*c) функции программного средства;*

*d) вывод результатов;*

*е) сообщения об ошибках и отказах, корректирующие и восстанавливающие действия.*

Данные:

a) Входные данные

*Должны быть описаны содержание, логическая структура и формат*

*входных данных, включая значение каждого элемента. Должна быть также описана любая зависимость данных от индивидуальных особенностей носителя данных.*

*Если требуются другое оборудование или программные средства для подготовки входных данных к программе, например, для создания файлов, требования к ним должны быть полностью определены.*

b) Вспомогательные данные

*Должны быть описаны любые вспомогательные данные, требуемые для работы программного средства, например словарь для программы обработки*

*текста.*

c) Выходные данные

*Должны быть описаны содержание и формат экранных изображений, печатных форм и любых других форм выходных данных. Должны быть приведены иллюстрации шаблона вывода.*

d) Промежуточные данные

*Для понимания пользователем работы программного средства, при необходимости, должны быть описаны содержание и формат любых промежуточных данных.*

е) Ограничения на данные

*Должны быть описаны любые ограничения на данные, например, предел по числу разрядов.*

f) Хранение данных

*Для понимания пользователем работы программного средства, при необходимости, должны быть описаны форматы файла и записей. Должно быть установлено, позволяют ли условия форматирования использовать файлы и записи других программных средств.*

*Должны быть описаны любые ограничения, например, максимальное число записей на файл, максимальное число файлов на диск.*

Среда разработки, используемые языки программирования, библиотеки, пакеты:

Среда разработки: .NET и node.js с клиентским javasript

Используемые языки программирование: C#, клиентский и серверный javascript

Библиотеки: библиотеки System и Microsoft среды .NET

Пакеты: Entity Framework, webpack, react, http, express, socket.io

### Объём программы: 1Мб

Используемые принципы и алгоритмы решения задач:

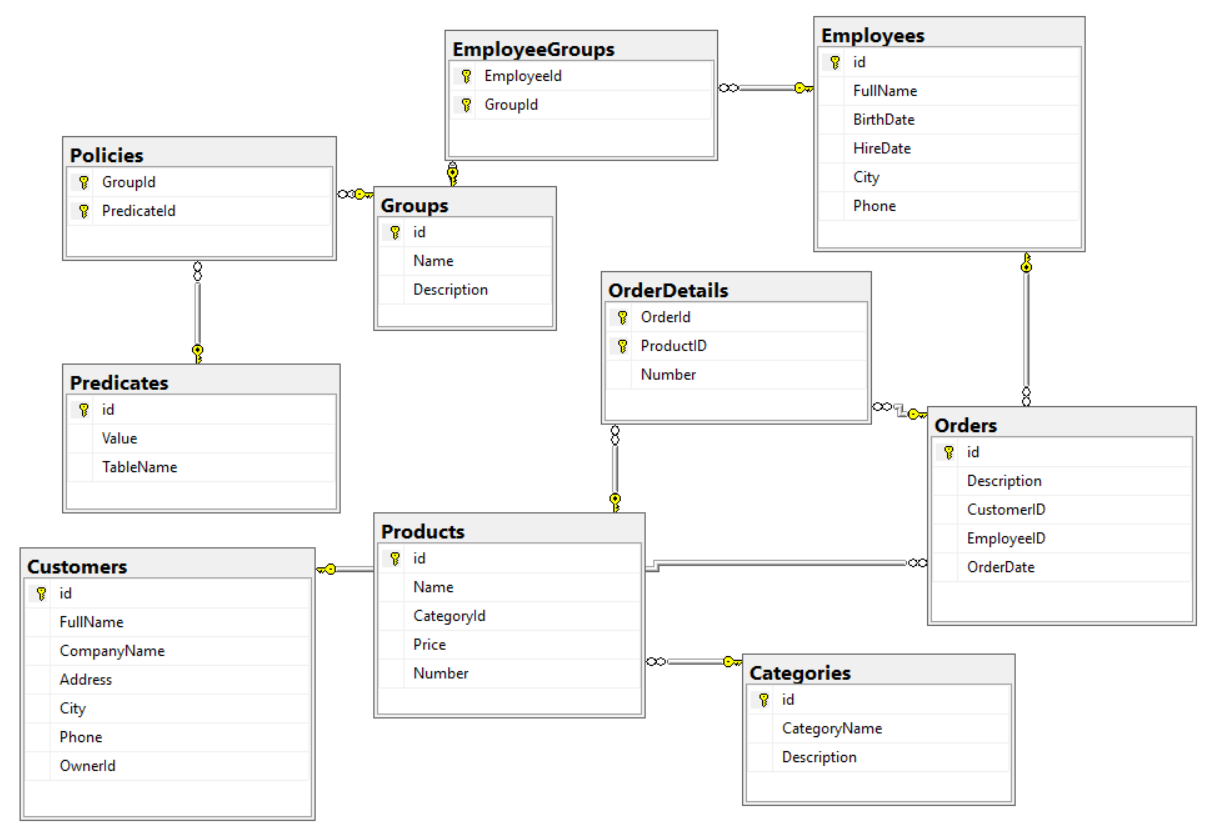
Для создания политики безопасности используется принцип построения дискретной политик безопасности. Для описания предикатов используется контекстно-свободная грамматика. Для вычисления предикатов используется алгоритм обхода синтаксического дерева в глубину. Для сканирования записи предиката используется рекурсивное проход по символам записи без возвратов.

Структура программного средства:

Программа состоит из

* api сервера, к которому обращается клиент
* клиента который обращается к api серверу за данными базы данных
* запущенного SQL сервера с базой данной
* библиотеки написанной на языке C# для разбора и исполнения предикатов

Схема базы данных:



Тестирование:

Пример:

Необходимо запустить данную программу и перейти в браузере на url – localhost:8080. Далее необходимо запустить SQL скрипт в файле “data.sql”для заполнения базы данных тестовыми данными. Далее вводя разные идентификаторы пользователей на странице localhost:8080/#login b и после нажатия кнопки с надписью “Sign In” можно увидеть что количество строк во вкладке с таблицей Orders разное, так как перейдя на вкладку Predicates можно увидеть предикаты наложенные на группы пользователей. Меняя идентификаторы пользователей на идентификаторы из групп, к которым прикреплены предикаты можно увидеть, что количество строк таблицы Orders меняется в зависимости от условия наложеного в предикате.

Дата Подпись автора:

Заключение комиссии:

Дата:

Члены комиссии: