

Особенности разработки информационной системы для предприятия

Е.А. Верещагина, Д.С. Колесникова, А.К. Рудниченко

Дальневосточный федеральный университет, Владивосток

Аннотация: В данной работе приведён функционал программного обеспечения, который рекомендуется предусматривать в случае разработки информационной системы для предприятия в целях экономии рабочего времени системных администраторов и денежных средств компании в будущем.

Ключевые слова: информационная система, предприятие, управление доступом, разграничение доступа, роли, обновление, логирование, Microsoft SQL Server, администрирование, системный администратор.

Информационная система – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств (Федеральный закон от 27.07.2006 N 149-ФЗ «Об информации, информационных технологиях и о защите информации»).

Разработка любого программного обеспечения – это использование целого комплекса широко известных для всех разработчиков операций, а также применение своих субъективных приёмов к той или иной ситуации. При проектировании информационной системы для предприятия необходимо учесть не только функционал конечного пользователя, но и многие другие аспекты, о которых будет вестись речь в данной статье [1, 2].

Разграничение доступа на основе ролей

Разграничение доступа (управление доступом) – определение полномочий (совокупности прав) субъекта для дальнейшего использования вычислительных ресурсов информационной системы [3].

Чаще всего в информационных системах механизм управления доступом основывается на использовании ролей, когда права доступа субъектов системы группируются в роли с учётом их функционала [4]. Конечным пользователям такой подход наиболее понятен, потому что права

доступа чётко ассоциируются с наименованием ролей (например, оператор, наблюдатель, администратор). Как правило, данный подход применяется в системах защиты систем управления базами данных (СУБД).

Таблица № 1

Пример разграничения доступа на основе ролей. Программа для учёта автомобильных пропусков.

1	Системный администратор	1. Управление пользователями 2. Управление справочниками (перечень организаций, марки и модели автомобилей) 3. Чтение базы данных (БД) 4. Добавление новых записей в БД 5. Изменение существующих записей в БД 6. Удаление записей из БД 7. Печать пропуска
2	Оператор	1. Чтение БД 2. Добавление новых записей в БД 3. Изменение существующих записей в БД 4. Удаление записей из БД 5. Печать пропуска
3	Редактор	1. Чтение БД 2. Изменение существующих записей в БД 3. Удаление записей из БД
4	Наблюдатель	1. Чтение БД

Механизм самостоятельного обновления

Самостоятельное обновление программного обеспечения – один из самых важных аспектов при разработке информационной системы для предприятия. Данный функционал в будущем сэкономит время как конечного пользователя, так и системного администратора.

Механизм самостоятельного обновления выглядит следующим образом:

1. Проверка версии программы на сервере. Сравнение её с текущей версией.

2. Получение ссылки на новую версию программного обеспечения.
3. Скачивание установщика.
4. Проверка контрольной суммы загруженного файла с файлом на сервере (проверка целостности файла обновления).
5. Прекращение работы основной программы.
6. Установка программного обеспечения из файла в автоматическом режиме так, чтобы вмешательство пользователя не требовалось.
7. Запуск новой версии программы.

Чаще всего механизм самостоятельного обновления выносят в отдельный исполняемый файл, скрипт или библиотеку, так как технически невозможно прекратить основной поток программы (закрыть её), а после этого запустить установщик [5].

Кроме этого, разработчики часто прибегают к разработке своего механизма установки программы без использования встроенной утилиты «msiexec» в операционной системе Microsoft Windows.

Логирование операций

При работе пользователей в информационной системе совершается много операций над различными данными: чтение, запись, обновление или удаление. Чем больше пользователей, тем больше совершается операций за единицу времени. Необдуманное или случайное действие человека в информационной системе может нанести ощутимый вред информации в базе данных. Например, изменение диагноза в медицинской карте, что может повлиять на дальнейший ход лечения пациента или удаление автомобильного пропуска из базы данных, что будет препятствовать доступу автомобиля на территорию организации. Узнать о факте внесённых изменений, а также найти виновника зачастую довольно сложно. Для этого разрабатываются модули логирования действий пользователей [6].

Логирование (журналирование) – форма автоматической записи операций о происходящих процессах в информационной системе в хронологическом порядке.

Лог (файл регистрации, протокол, или журнал) – файл с записями о событиях в хронологическом порядке, простейшее средство обеспечения журналирования. Может представляться как в виде файла регистрации, так и в виде базы данных.

Есть несколько подходов к логированию операций:

- отправка двойного SQL-запроса в базу данных из программного обеспечения;
- логирование в хранимой процедуре Microsoft SQL Server (MS SQL);
- использование дополнительной библиотеки.

В данной статье рассмотрены указанные методы логирования данных на примере Microsoft SQL Server.

Первый способ заключается в отправке дополнительного SQL-запроса вместе с запросом на запись, обновление или удаление данных. Для данного способа в базе данных MS SQL создаётся таблица (например, «logs»). В неё помещается информация обо всех изменениях в основной таблице. Пример простейшего логирования [7, 8]:

```
INSERT INTO table (id, name, age) VALUES (2, 'Петров Пётр Петрович', 56);  
INSERT INTO logs (datetime, idtable, username, type) VALUES ('2019-01-01  
14:30:00.000', 2, CURRENT_USER, 'INSERT');
```

Первым SQL-запросом добавляются данные в основную таблицу, а вторым – данные в таблицу, предназначенную для логирования. В данном случае оба запроса необходимо отправить из программного обеспечения,

которое находится на компьютере пользователя. В этом главный минус данного метода. Запрос, отвечающий за логирование, формируется и отправляется пользователем, а значит последний может на это повлиять (например, с помощью отладчика компьютерных программ).

Способ логирования в хранимой процедуре MS SQL более безопасен, так как пользователь не имеет доступа к самому запросу напрямую. Он заложен в хранимую процедуру MS SQL.

Хранимые процедуры представляют собой набор команд, состоящий из одного или нескольких операторов или функций SQL. Сохраняются в базе данных в откомпилированном виде. Хранимые процедуры – полноценный объект базы данных. Вследствие этого каждая хранимая процедура располагается в конкретной базе данных, где и выполняется. Пользователь может её запустить командой EXEC, если имеет на это право доступа.

Ранее написанные SQL-запросы представим в виде процедуры [9]:

```
CREATE PROCEDURE AddInfo(@id int, @name nvarchar(MAX), @age int) AS
BEGIN
    INSERT INTO table (id, name, age) VALUES (@id, @name, @age);
    INSERT INTO logs (SYSDATETIME(), @id, CURRENT_USER, 'INSERT');
END
```

В этом случае в программном обеспечении не передаётся целый запрос на сервер баз данных, а вызывается функция «AddInfo» с необходимыми аргументами:

```
EXEC AddInfo 2, 'Петров Пётр Петрович', 56;
```

Для того, чтобы не разрабатывать функционал логирования с нуля, можно использовать дополнительные библиотеки. Как правило, они

позволяют записывать действия в файл регистрации, а также отправлять в базу данных. Такое решение подходит ко многим реализациям, так как набор выходных данных в лог определяется разработчиком.

Такие библиотеки имеют ощутимый минус – сбор логируемых данных и отправка в базу данных происходит со стороны пользователя. Возможные проблемы аналогичны первому способу – отправкой двойного SQL-запроса в базу данных.

При обеспечении логирования операций в информационной системе на предприятии также важно предусмотреть функционал отката того или иного события. Для этого также в файл регистрации записывается старое значение из таблицы, чтобы потом при необходимости его импортировать в основную таблицу базы данных.

Панель администрирования

В процессе разработки информационной системы для предприятия необходимо учитывать, что конечным пользователем может выступать как среднестатистический оператор, так и системный администратор, который будет управлять ролями и пользователями в рамках разграничения доступа. Кроме этого, системный администратор может производить экспорт/импорт информации в базы данных, массовое редактирование, добавление информации в справочники (например, организации, марки и модели автомобилей и т. д.).

В связи с этим проектируется и разрабатывается отдельный раздел информационной системы – панель администрирования, к которой будет иметь доступ только тот субъект, который принадлежит роли «Администратор».

Любая информационная система должна быть гибкой и удобной для конечного пользователя. Например, системный администратор для добавления новой роли разграничения доступа пользуется имеющимся



функционалом в программе. В противном случае, если это не предусмотрено на этапе разработки заранее, разработчику придётся выполнять ту же операцию в исходном коде программы или базе данных вручную. Таким образом, более гибкое программное обеспечение, спроектированное с учётом представленных в данной статье аспектов, позволяет сэкономить денежные средства предприятия, если программист не числится в кадровом штате.

Литература

1. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд. Стандарт третьего поколения. СПб.: Питер, 2012. С. 539-542
2. Панкратов А.А., Анисимова Г.Б. Проектирование информационной системы оптимизации работы автостоянки. // Инженерный вестник Дона, 2018, №3 URL: ivdon.ru/ru/magazine/archive/n3y2018/5091.
3. Messaoud Benantar. Access Control Systems. New York: Springer Science+Business Media, 2006. pp. 26, 192-194.
4. Прохорова О.В. Информационная безопасность и защита информации. Самара: СГАСУ, 2014. С. 18-22.
5. Качин К., Гуерру Р., Родригес Л. Введение в надежное и безопасное распределенное программирование. М.: ДМК Пресс, 2016. С. 32-35.
6. Баранчиков А.И., Баранчиков П.А., Громов А.Ю., Ломтева О.А. Организация сетевого администрирования: Учебник. М.: КУРС: ИНФРА-М, 2017. С. 65-76.
7. Mike McQuillan, 2015. Introducing SQL Server. Apress, pp: 130-136.
8. Астахова И.Ф., Мельников В.М., Толстобров А.П., Фертиков В.В. и др. СУБД: язык SQL в примерах и задачах. М.: Физматлит, 2009. С. 71-76.

9. Создание хранимой процедуры. URL: docs.microsoft.com/ru-RU/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=aps-pdw-2016 (дата обращения: 22.01.2019).

10. Рудниченко А.К. Разработка клиент-серверной информационной системы учёта автомобильных пропусков с использованием информации из устаревшей базы данных. // Инженерный вестник Дона, 2018, №4 URL: ivdon.ru/ru/magazine/archive/n4y2018/5296.

References

1. Orlov S.A., Tsil'ker B.Ya. Tekhnologii razrabotki programmnoho obespecheniya [Software Development Technologies]: Uchebnik dlya vuzov. 4-e izd. Standart tret'ego pokoleniya. SPb.: Piter, 2012. pp. 539-542

2. Pankratov A.A., Anisimova G.B. Inženernyj vestnik Dona (Rus). 2018, №3. URL: ivdon.ru/ru/magazine/archive/n3y2018/5091.

3. Messaoud Benantar. Access Control Systems. New York: Springer Science+Business Media, 2006. pp. 26, 192-194.

4. Prokhorova O.V. Informatsionnaya bezopasnost' i zashchita informatsii [Information Security and Data Protection]. Samara: SGASU, 2014. pp. 18-22.

5. Kachin K., Guerru R., Rodrigues L. Vvedenie v nadezhnoe i bezopasnoe raspredelennoe programmirovaniye [Introduction to Reliable and Secure Distributed Programming]. M.: DMK Press, 2016. pp. 32-35.

6. Baranchikov A.I., Baranchikov P.A., Gromov A.Yu., Lomteva O.A. Organizatsiya setevogo administrirovaniya [Network administration organization]: Uchebnik. M.: KURS: INFRA-M, 2017. pp. 65-76.

7. Mike McQuillan, 2015. Introducing SQL Server. Apress, pp: 130-136.

8. Astakhova I.F., Mel'nikov V.M., Tolstobrov A.P., Fertikov V.V. i dr. SUBD: yazyk SQL v primerakh i zadachakh [DBMS: SQL language in examples and tasks]. M.: Fizmatlit, 2009. pp. 71-76.



9. Sozдание khranimoy protsedury [Creating a stored procedure]. URL: docs.microsoft.com/ru-RU/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=aps-pdw-2016 (data obrashcheniya: 22.01.2019)

10. Rudnichenko A.K. Inzhenernyj vestnik Dona (Rus), 2018, №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5296.