

Разработка клиент-серверной информационной системы учёта автомобильных пропусков с использованием информации из устаревшей базы данных

Е.А. Верецагина, А.К. Рудниченко

Дальневосточный федеральный университет, Владивосток

Аннотация: В данной работе приведено описание существующей в университете базы данных в Microsoft Excel, её проблемные аспекты, а также методы унификации информации в ней. Описана новая база данных в системе управления базами данных Microsoft SQL Server и миграция на неё. Спроектированы основные составляющие будущего прикладного программного обеспечения для работы с новой базой данных.

Ключевые слова: база данных, Microsoft SQL Server, Microsoft Excel, университет, автомобильный пропуск, реестр, унификация, миграция, C#, программное обеспечение.

Введение

В своей основе любая система управления базами данных (СУБД) – это компьютерная система ведения неких записей. Она позволяет сохранять информацию, обеспечивает доступ к ней, выполняет какие-то операции над данными. Многие организации при отсутствии соответствующего финансирования или в силу простоты использования ведут базы данных в офисном приложении Microsoft Excel. Как правило, обусловлено это тем, что Excel – это универсальный аналитический инструмент, который больше подходит для сложных расчетов, вычислений, сортировки и даже для сохранения структурированных данных [1]. У данной программы простой и понятный интерфейс, который не нужно создавать с нуля программисту.

Одним из главных недостатков Microsoft Excel является его зависимость от рабочего места. Для того, чтобы такой базой данных могли пользоваться несколько сотрудников, её необходимо передавать на другой компьютер и обратно обычным файлом. Некоторые организации решают этот вопрос с помощью облачных технологий, но в них тоже есть свои минусы. Например, отсутствие разграничения прав доступа к документу по

процессам (создание, обновление, добавление), полям (столбцам), таблицам и т. д.

Со временем в базах данных растёт количество записей, а их структура становится всё сложнее и сложнее. В связи с этим, использование Microsoft Excel уже не представляется возможным. Кроме этого, в Excel очень сложно унифицировать данные. В результате этого появляется разнородность данных в таблице, которую с течением времени сложнее исправить. Виной тому человеческий фактор, так как с базой данных могут работать разные сотрудники и допускать различные ошибки в формате данных.

При переходе на другую СУБД необходимо перенести существующую базу данных в новую СУБД (осуществить миграцию) [2]. Именно в данном случае актуальна проблема проверки данных на унифицированность, а также последующая их унификация при отрицательном результате тестирования.

После миграции на новую СУБД необходимо все последующие операции с базой данных производить только в новой СУБД, так как база данных пополняется записями каждый день. Работа в СУБД возможна только с разработанным программным обеспечением для работы с базой данных. Миграция будет проходить на последних этапах, перед пилотным запуском программного обеспечения.

В системе управления проектами Microsoft Project разработан план создания клиент-серверной информационной системы учёта автомобильных пропусков [3]. На рис.1 показана диаграмма Ганта, отображающая график работ по проекту и примерную общую длительность его реализации.

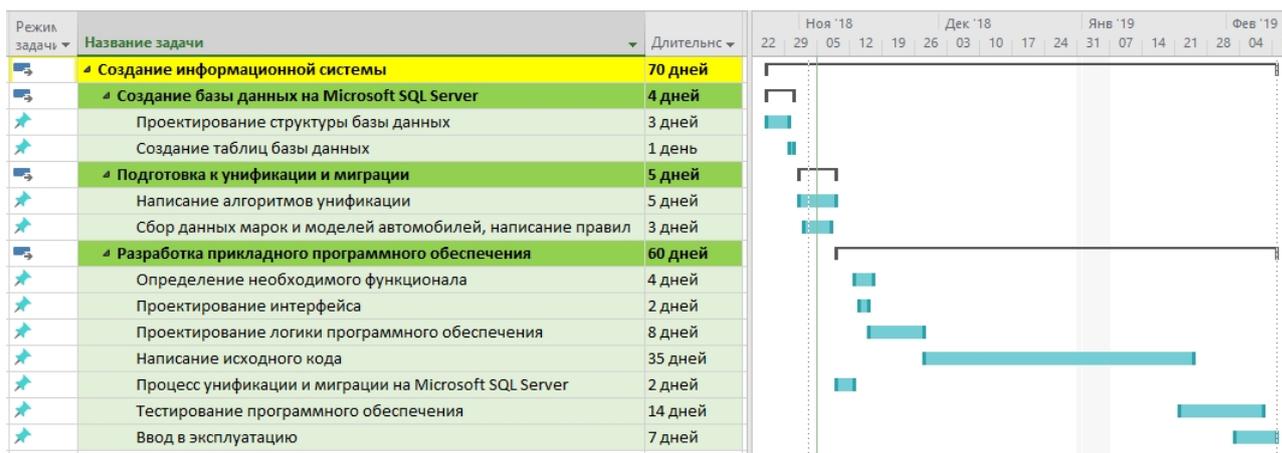


Рис. 1. – Диаграмма Ганта проекта по созданию информационной системы учёта автомобильных пропусков

Создание новой информационной системы, включающей в себя разработку прикладного программного обеспечения, обусловлено тем, что на рынке практически отсутствуют аналоги данных систем. Университет, как и любое другое крупное учреждение, имеет свою специфику, которую необходимо отразить в информационной системе. Необходимую гибкость имеют решения от компании 1С – они и составляют основную конкуренцию данной разработке.

На любом предприятии используются по-своему уникальные решения 1С, реализованные не только настройкой конфигурации, но и программированием на языке 1С, который встроен в платформу. Таким образом, возникает необходимость купить платформу 1С, которую затем следует доработать на специализированном языке программирования согласно специфике организации.

Платформа 1С не отвечает требованиям модульности программного обеспечения и зачастую имеет в себе лишний неотключаемый функционал, который может загружать систему. Для минорного обновления одной части программы, необходимо загружать и обновлять весь комплекс целиком, что занимает довольно большое количество времени.

Любая разработка, основанная на платформе 1С, имеет свои ограничения, а также плохо интегрируется с другими информационными системами. Собственную информационную систему в будущем планируется интегрировать с имеющимися сервисами университета, написанными собственными сотрудниками образовательного учреждения.

Таким образом, рассматривая информационную систему учёта автомобильных пропусков, при написании прикладного программного обеспечения с нуля будет затрачено меньшее количество ресурсов и получено больше возможностей на заключительном этапе разработки.

Описание существующей базы данных

База данных в формате Microsoft Excel находится в университете в профильном подразделении, которое занимается организацией дорожного движения на территории университета. Территория университета огорожена и на каждом въезде оборудован контрольно-пропускной пункт, а в самом университете организован пропускной режим.

Существующая база данных состоит из двух таблиц, в которых содержатся данные об автомобильных пропусках двух видов:

- постоянные автомобильные пропуска;
- временные автомобильные пропуска.

Поля (столбцы) таблиц разнятся в зависимости от типа пропуска. При этом каждая таблица имеет в себе по 9 полей.

Таблица № 1

Поля реестров автомобильных пропусков

Реестр постоянных пропусков	Реестр временных пропусков
1. Номер пропуска;	1. Номер пропуска;
2. Фамилия, имя, отчество (ФИО);	2. Период действия пропуска (две даты);
3. Марка и модель транспортного средства;	3. Фамилия, имя, отчество (ФИО);
	4. Марка и модель транспортного средства;

4. Государственный регистрационный знак (ГРЗ) транспортного средства;	средства;
5. Номер парковки;	5. Государственный регистрационный знак (ГРЗ) транспортного средства;
6. Символ «Р» (позволяет парковаться на всех парковках);	6. Номер парковки;
7. Подразделение сотрудника, студент или внешняя организация;	7. Подразделение сотрудника, студент или внешняя организация;
8. Контактный номер телефона;	8. Контактный номер телефона;
9. Примечание.	9. Примечание.

Исходя из того, что таблицы заполняются вручную с минимальной автоматизацией процесса, в ней присутствует неунифицированность вводимых данных. Уже после приведения таблиц к единому формату необходимо приступить к миграции на другую СУБД.

Основные несоответствия данных, подлежащие последующей унификации:

1. В поле «фамилия, имя, отчество» (ФИО) проставляется два пробела, например, между фамилией и именем;
2. Первая буква в написании фамилии, имени или отчества – строчная;
3. Государственный регистрационный знак (ГРЗ) вводился с использованием как русской раскладки клавиатуры, так и английской. Также есть случаи, когда номер региона отделяется либо через символ пробела, либо через символ «слеш» («/»).
4. Марка и модель автомобиля введены неправильным регистром (строчными буквами, или наоборот, заглавными).
5. Номера телефонов в таблице начинаются с «7», «8», «+7» или введены по формату «+7 (XXX) XXX-XX-XX».

6. В существующей таблице информация о том, что пропуск был сдан в бюро пропусков, помещалась в поле «Примечание». Планируется вывести в отдельное логическое поле «passed» (рис.2, 3).

7. Период действия пропуска, состоящий из двух дат, вводился в одну ячейку. Планируется разделить на два значения (рис.3).

8. В таблицах существуют пустые записи (строки), так как при удалении пропуска из базы данных строка не удалялась.

Фрагменты базы данных с неунифицированными данными для примера представлены в таблице 2. Оранжевым цветом выделены несоответствия данных, которые необходимо унифицировать.

Таблица № 2

Фрагменты таблицы с временными автомобильными пропусками

№ пропуска	Период действия	Фамилия Имя Отчество	Марка и модель ТС	Гос. рег. номер	Парковка	Подразделение, студент или организация	Контактный номер телефона	Примечание
4913	30.09.2017 31.12.2017	Бойко Анатолий Сергеевич	NISSAN BLUEBIRD	В 846 СА/25	П-2	Студент	79249283402	сдан
5634	14.10.2017 31.12.2018	Богдарова Кристина Николаевна	Mitsubishi Rvr	Н 364 НА 125	П-3	Отдел кадрового делопроизводства	+79243578421	
6298	11.12.2017 01.05.2018	ильченко Алексей Андреевич	Toyota Aqua	К 529 МС/125	П-1	Студент	79081196821	Сдан
7233	20.03.2018 31.12.2018	Титов Филипп Викторович	SUZUKI SWIFT	Е 195 ВС 125	П-3	Отдел закупок	89141274788	
8234	01.08.2018 31.08.2018	Иванова Дарья Анатольевна	Toyota Corolla	О 722 РН 125	П-2	ООО "Метео-С"	79249826766	сдан
8856	02.10.2018 31.12.2018	Бондарев Илья Вячеславович	toyota vitz	Р 135 ОХ 125	П-2	Студент	89840192748	
9110	14.10.2018 01.08.2019	Савченко алексей егорович	Honda Fit	Р 344 РР 125	П-1	Студент	79148472672	

Унификация данных

В качестве цели для миграции выбрана СУБД Microsoft SQL Server [4], так как практически вся инфраструктура университета построена на решениях от корпорации Microsoft. Особых требований к СУБД не требуется.

В Microsoft SQL Server создан каркас базы данных – пустые таблицы с настроенными полями. Реестры пропусков будут храниться также в двух таблицах: таблица с постоянными пропусками (рис.2) и таблица с временными пропусками (рис.3).

	Имя столбца	Тип данных	Разрешить ...
🔑	id	int	<input type="checkbox"/>
	number	text	<input type="checkbox"/>
	date	date	<input type="checkbox"/>
	fullname	text	<input type="checkbox"/>
	carmodel	int	<input type="checkbox"/>
	carnumber	text	<input type="checkbox"/>
	parking	text	<input type="checkbox"/>
	organization	text	<input type="checkbox"/>
	telephone	text	<input checked="" type="checkbox"/>
	passed	bit	<input type="checkbox"/>
	note	text	<input checked="" type="checkbox"/>

Рис. 2. – Конструктор таблицы с постоянными пропусками

🔑	id	int	<input type="checkbox"/>
	number	text	<input type="checkbox"/>
	date	date	<input type="checkbox"/>
	begindate	date	<input type="checkbox"/>
	enddate	date	<input type="checkbox"/>
	fullname	text	<input type="checkbox"/>
	carmodel	int	<input type="checkbox"/>
	carnumber	text	<input type="checkbox"/>
	parking	text	<input type="checkbox"/>
	organization	text	<input type="checkbox"/>
	telephone	text	<input checked="" type="checkbox"/>
	passed	bit	<input type="checkbox"/>
	note	text	<input checked="" type="checkbox"/>

Рис. 3. – Конструктор таблицы с временными пропусками

Для решения вопроса унификации данных к имеющимся таблицам будут применены алгоритмы, написанные на языке программирования C# в среде разработки Microsoft Visual Studio 2017. После унификации данные переносятся в базу данных Microsoft SQL Server посредством SQL-запросов [5, 6]. Диаграмма деятельности процесса унификации в общем случае представлена на рис.4 [7, 8].



Рис. 4. – Диаграмма деятельности общего алгоритма унификации
Унификация ФИО (рис.5). В поле «фамилия, имя, отчество» изредка проставляется два пробела, например, между фамилией и именем. Фамилия, имя или отчество написаны со строчной буквы. Унификация производится заменой символов по условию. Циклически проверяется регистр каждой первой буквы слова и, при необходимости, исправляется.



Рис. 5. – Диаграмма деятельности алгоритма унификации ФИО
Унификация ГРЗ (рис.6). ГРЗ вводился с использованием как русской раскладки клавиатуры, так и английской. Также есть случаи, когда номер

региона отделяется через символ пробела либо через символ «слеш» («/»). Унификация производится заменой символов по условию, а также расстановкой символов пробела по заданному шаблону государственного регистрационного номера для автомобилей в Российской Федерации.

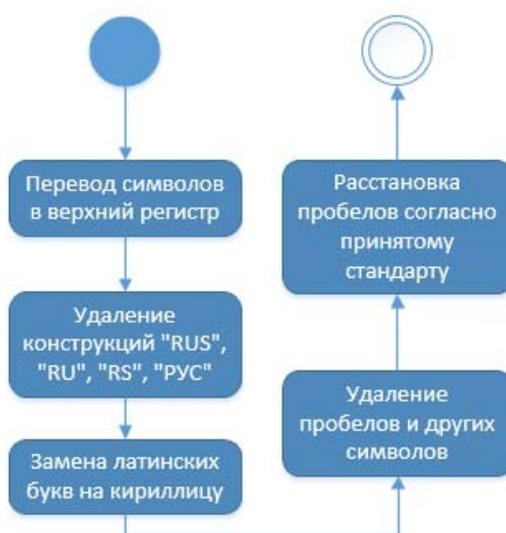


Рис. 6. – Диаграмма деятельности алгоритма унификации ГРЗ

Унификация марки и модели автомобиля (рис.7). Марка и модель автомобиля введены неправильным регистром (строчными буквами или, наоборот, заглавными). Унификация достигается методом сравнения введённых данных со специальными, заранее подготовленными списками (список марок автомобилей и список моделей автомобилей). При обнаруженном несоответствии строка изменяется согласно данным в списках. «Replace» - правила диктуют, на какие данные заменять определённые конструкции в строковой переменной.

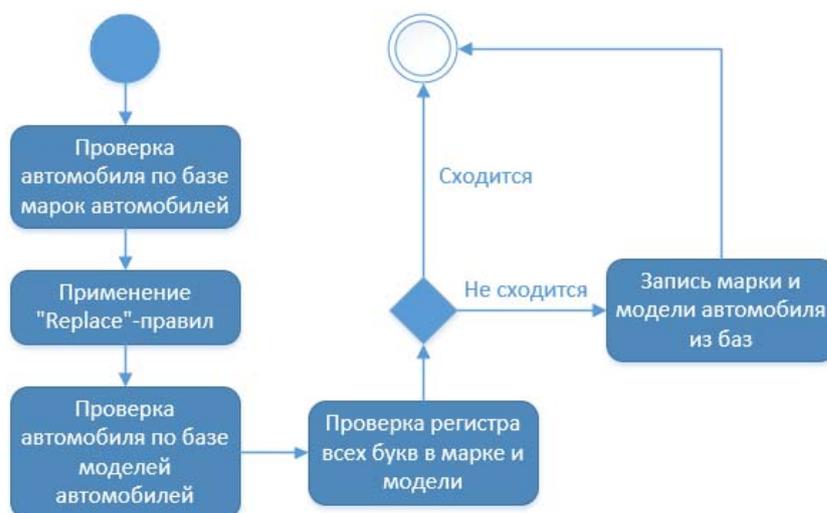


Рис. 7. – Диаграмма деятельности алгоритма унификации марки и модели автомобиля

Унификация номера телефона (рис.8). Номера телефона в таблице начинаются с «7», «8», «+7» или введены по формату «+7 (XXX) XXX-XX-XX». Унификация производится заменой символов по условию. Главным принципом является замена кода страны «+7» на «8». Впоследствии в прикладном программном обеспечении номер телефона можно будет выводить по любому шаблону.



Рис. 8. – Диаграмма деятельности алгоритма унификации номера телефона

Для унификации всех данных использовались три заранее созданных текстовых списка:

- «Brands» – список марок автомобилей с их идентификаторами;
- «Models» – список моделей автомобилей с идентификаторами марок автомобилей;
- «Rules» – список «Replace»-правил корректировки данных. Заполняется вручную. Например, «RAV 4» заменить на «RAV4», «Mark 2» заменить на «Mark II» и т.д.

Остальные проблемы унификации, перечисленные в данной работе, решены с помощью встроенных средств Microsoft Excel вручную до запуска основного алгоритма унификации.

Создание программного обеспечения

Соединение и работа с базой данных будет осуществляться с помощью SQL-запросов к СУБД из прикладного программного обеспечения,

написанного специально для данной базы данных на языке программирования C# в среде Microsoft Visual Studio 2017 [4,9].

Программа должна состоять из четырёх разделов:

1. Постоянные пропуска. Раздел будет содержать основную таблицу с постоянными пропусками. Возможно применять фильтры.
2. Временные пропуска. Раздел будет содержать основную таблицу с временными пропусками. Возможно применять фильтры.
3. Пропуска на изготовление. Раздел будет содержать только те пропуска, которые отмечены «На изготовление» (при условии, если изготовлением занимается отдельный сотрудник).
4. Чёрный список. В данном разделе будут помещаться те заявки, которые были аннулированы и занесены в чёрный список.

Предусматривается три уровня доступа:

1. Оператор базы данных. Имеет права на добавление, изменение, удаление любого вида пропусков в базе данных. Может отправлять пропуска на изготовление, а также добавлять в чёрный список их, или убирать из чёрного списка.
2. Изготовитель пропусков. Имеет права только на раздел «Изготовление пропусков». Может просмотреть пропуск и отметить его изготовленным и готовым к выдаче.
3. Сотрудник охраны. Имеет права на раздел «Чёрный список» только в режиме чтения.

Интерфейс программного обеспечения должен быть простой и дружелюбный. В главном окне размещены таблица и фильтр. Таблицы можно переключать с помощью закладок «Постоянные пропуска», «Временные пропуска», «На изготовлении», «Чёрный список». Отображение этих вкладок определяется правилами разграничения доступа. На данный момент интерфейс выглядит так, как показано на рис.9.

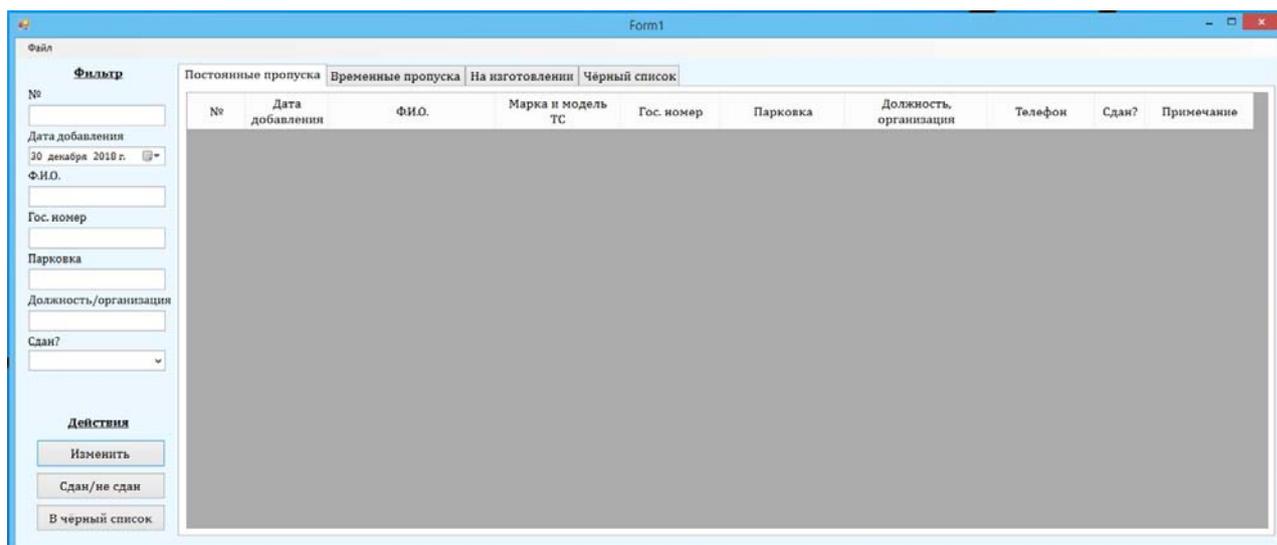


Рис. 9. – Интерфейс программного обеспечения

На главном экране приложения продублированы действия в виде кнопок «Изменить», «Сдан/не сдан», «В чёрный список». Данные действия можно выполнить как через нажатие правой кнопки мыши, так и через кнопки в форме. По мере разработки программы интерфейс будет меняться, но основная концепция «Фильтр, таблица» останется неизменной.

Таким образом, в данной статье приведены методы унификация данных для последующей миграции на новую СУБД через встроенные средства Microsoft SQL Server. Методом миграции на многофункциональную СУБД решена проблема человеческого фактора при заполнении таблиц. Спроектированы основные составляющие будущего программного обеспечения для работы с новой базой данных.

Литература

1. Paul Cornell, 2007. Excel as Your Database. Apress, pp: 127-129.
2. Импорт данных из Excel в SQL Server или базу данных Azure. // Microsoft Docs. URL: docs.microsoft.com/ru-ru/sql/relational-databases/import-export/import-data-from-excel-to-sql?view=sql-server-2017 (дата обращения: 05.11.2018).

3. Кудрявцев Е.М. Методы сетевого планирования и управления проектом. М.: ДМК Пресс, 2005. С. 110-116.

4. Земцов А.Н., Болгов Н.В., Божко С.Н. Многокритериальный выбор оптимальной системы управления базы данных с помощью метода анализа иерархий. // Инженерный вестник Дона. 2014. № 2. URL: ivdon.ru/magazine/archive/n2y2014/2360

5. Тарасов С.В. СУБД для программиста. Базы данных изнутри. М.: Солон-Пресс, 2015. С. 9-14, 83-87.

6. 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.

7. Мартин Фаулер. UML Основы. 3 изд. СПб.: Символ-Плюс, 2005. С. 139-150.

8. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2 изд. М.: ДМК Пресс, 2008. С. 112-118.

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

10. 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

References

1. Paul Cornell, 2007. Excel as Your Database. Apress, pp: 127-129.

2. Import dannykh iz Excel v SQL Server ili bazu dannykh Azure. // Microsoft Docs. URL: docs.microsoft.com/ru-ru/sql/relational-databases/import-export/import-data-from-excel-to-sql?view=sql-server-2017 (data obrashcheniya: 05.11.2018).

3. Kudryavtsev E.M. Metody setevogo planirovaniya i upravleniya proektom. [Methods of network planning and project management] M.: DMK Press, 2005. pp. 110-116.



4. Zemtsov A.N., Bolgov N.V., Bozhko S.N. Inzenernyj vestnik Dona (Rus). 2014. № 2. URL: ivdon.ru/magazine/archive/n2y2014/2360

5. Tarasov S.V. SUBD dlya programmista. Bazy dannykh iznutri. [DBMS for the programmer. Databases inside] M.: Solon-Press, 2015. pp. 9-14, 83-87.

6. 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.

7. Martin Fowler. UML Osnovy. [UML Basics] 3 izd. SPb.: Simvol-Plyus, 2005. pp. 139-150.

8. Buch G., Rambo D., Yakobson I. Yazyk UML. Rukovodstvo pol'zovatelya. [UML language. User's manual.] 2 izd. M.: DMK Press, 2008. pp. 112-118.

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

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